

Aus dem Institut für Tierzucht und Haustiergenetik der
Georg-August-Universität Göttingen

**Entwicklung von Systemen
der mobilen Informationstechnologie
zwecks Zugriffs auf Datenbestände aus der Nutztierzucht
am Beispiel von Daten aus der Traberzucht**

Dissertation
zur Erlangung des Doktorgrades
der Fakultät für Agrarwissenschaften
der Georg-August-Universität Göttingen

vorgelegt von
Hanefi Yağmur Akgün
geboren in Ankara

Göttingen, im September 2005

D7

Referent: Prof. Dr. E. Bruns
Korreferent: Prof. Dr. W. Lücke

Tag der mündlichen Prüfung: 18. November 2005

Sevgili Aileme...

Inhaltsverzeichnis

I. Verzeichnis der Tabellen.....	I
II. Verzeichnis der Abbildungen.....	II
III. Verzeichnis der Abkürzungen.....	III
1. Einleitung.....	1
2. Ausgangssituation.....	3
2.1. Wirtschaft Im Internet: Wiedererfindung einer alten Wirtschaftsstruktur	3
2.2. Sicherheit.....	6
2.2.1. Definition der Sicherheit	6
2.2.2. Sicherheit im Netzwerk und Internet.....	8
Angriffsmotive	10
Angriffsarten	11
<i>Denial of Service (Dos)</i>	11
<i>Identitätsdiebstahl</i>	12
<i>Man-In-The-Middle-Angriff</i>	12
<i>Insider Angriffe</i>	13
Sicherheitsmaßnahmen.....	13
<i>Verschlüsselungstechnologien</i>	13
<i>Netzwerktechnische Maßnahmen</i>	14
<i>Maßnahmen gegen Social-Engineering</i>	14
2.3. Mobile Internetzugänge.....	14
2.3.1. Entwicklung der mobilen Internetzugänge.....	14
2.3.2. Mobiltelefon vs. Computer.....	15
Mobilität und Bedienungsergonomie	16
Penetranz im Alltagsleben	17
Kosten	17
Vermarktungsmodell.....	17
Organisation der GSM-Netzbetreiber	18
Netzwerkumgebung.....	18
2.4. Wettgeschäft im Internet.....	20
2.4.1. Produktdifferenzierung beim Pferderennenwetten im Internet	21
2.4.2. Mobiltelefone als ein Medium für Pferderennenwetten	22
2.5. Datenverkehr im Trabrennwettgeschäft	23
3. Material und Methoden	26
3.1. Datenstrukturen und Daten	26
3.2. Technische Ausstattung	32
3.2.1. Server-Hardware	32
Arbeitsspeicher	32
Hauptprozessor (CPU)	33
Festplatten	33
Netzwerk.....	34
3.2.2. Dauerbetrieb.....	34
3.2.3. Mobile Internetzugänge	35
3.3. Software	36
3.3.1. Betriebssystem	37
3.3.2. Webserver	37
3.3.3. Datenbanken	38
3.4. Programmiersprachen	40

3.4.1. PHP: Hypertext Preprocessor (PHP).....	40
Grundlagen der PHP	41
Funktionen und Serverside-Includes im PHP	45
Sitzungsmanagement und Cookies im PHP	48
Datenbankverbindung und -Abfragen beim PHP	50
3.4.2. Structured Query Language (SQL).....	56
4. Darstellung der Ergebnisse	62
4.1. Das Programm	62
4.1.1. Komponenten	62
Datenbankschnittstelle	63
Authentifizierung und Sitzungsmanagement	64
Datenaufbereitung	66
WAP2-Schnittstelle	67
Zeitkontrolle	68
4.1.2. Überblick der einzelnen PHP-Skripten des Programms	68
4.1.3. Funktionsweise	74
4.1.4. Installation	78
Einstellungen des Apache HTTP-Servers	78
Konfiguration des Programms	79
4.1.5. Sicherheit.....	79
Sicherheit auf der Benutzerebene	80
Sicherheit bei der Datenübertragung	80
Sicherheit im Serverbereich	81
Praktische Sicherheitshinweise zum Programm	81
5. Diskussion	83
5.1. Das Programm	84
5.2. Programmiersprachen	85
5.3. Datenbanken	87
5.4. Weitere Entwicklungsmöglichkeiten des Programms	87
5.5. Mobile Internetzugänge	88
5.6. Kosten der Programmentwicklung	89
5.7. Mögliche Kosten eines Produktiven Systems	89
5.8. Realisierbarkeit eines Produktiven Systems	90
5.9. Implementierungsmöglichkeiten des Programms in anderen Bereichen der Landwirtschaft	90
6. Zusammenfassung	93
7. Summary	95
8. Literaturverzeichnis	97
8.1. Zitierte Publikationen	97
8.2. Zitierte Internetverweise	100
9. Anhang	101
9.1. Abbildungen	101

I. Verzeichnis der Tabellen

Tabelle 1: TGP – Renntage.....	27
Tabelle 2: ASP – Rennen.....	27
Tabelle 3: STP – Starter.....	28
Tabelle 4: TLP – Leistungsdaten.....	28
Tabelle 5: TRP – Traberstamm.....	29
Tabelle 6: SLP – Schlüssel.....	30
Tabelle 7: KUNDE – Kundendaten.....	30
Tabelle 8: WETTE – Wettscheine.....	30
Tabelle 9: Datenmengen in der Datenbank.....	31
Tabelle 10: Die wichtigsten SQL-Befehle.....	56
Tabelle 11: Die Tabelle für die SQL-Beispiele.....	57
Tabelle 12: Liste der Operatoren in SQL.....	59
Tabelle 13: Entscheidungsprozess zur Benutzerauthentifizierung.....	65

II. Verzeichnis der Abbildungen

Abbildung 1: Graphischer Verlauf des Dot-Com-Booms und -Crashes in NASDAQ ...	5
Abbildung 2: Schematische Darstellung einer Datenübertragung über das Internet .	9
Abbildung 3: Ausgabe eines PHP-Skriptes; erster Schritt.....	42
Abbildung 4: Ausgabe eines PHP-Skriptes; zweiter Schritt	42
Abbildung 5: Ausgabe eines PHP-Skriptes; dritter Schritt.....	43
Abbildung 6: Ausgabe eines PHP-Skriptes; vierter Schritt.....	43
Abbildung 7: Ausgabe eines PHP-Skriptes; fünfter Schritt.....	44
Abbildung 8: Ausgabe des PHP-Skriptes im Browserfenster.....	45
Abbildung 9: Ausgabe des PHP-Skriptes im Browserfenster.....	47
Abbildung 10: Ausgabe des PHP-Skriptes im Browserfenster.....	50
Abbildung 11: Ausgabe eines PHP-Skriptes im Browserfenster	55
Abbildung 12: Logische Komponenten des Programms nach Funktionsbereichen .	62
Abbildung 13: Darstellung der Kontrollebenen des Programms	63
Abbildung 14: Flussdiagramm des Programms	77
Abbildung 15: Startseite des Programms.....	101
Abbildung 16: Seite zur Suche nach Traberinformationen.....	101
Abbildung 17: Seite zur Anmeldung.....	102
Abbildung 18: Begrüßungsseite nach erfolgreicher Anmeldung	102
Abbildung 19: Darstellung der Rennveranstaltungen an einem Datum	102
Abbildung 20: Darstellungen der Rennen in einem Renntag an einem Rennbahn	103
Abbildung 21: Darstellung der Starter eines Rennens	103
Abbildung 22: Darstellung des Wettscheins für ein Rennen	104
Abbildung 23: Ein ausgefüllter Wettschein.....	104
Abbildung 24: Darstellung eines zu spielenden Wettscheins, zur Kontrolle für den Benutzer	105
Abbildung 25: Darstellung des Ergebnisses einer erfolgreichen Wettannahme	105
Abbildung 26: Darstellung der Informationen eines Trabers.....	106
Abbildung 27: Darstellung der Formdaten eines Trabers.....	106
Abbildung 28: Seite zur Darstellung der Kundeninformationen.....	107
Abbildung 29: Seite zur Passwortänderung	107

III. Verzeichnis der Abkürzungen

ACK	A cknowledgement
ANSI	A merican N ational S tandards Institute
ASP	A ctive S erver P ages
CGI	C ommon G ate I nterface
CPU	C entral P rocessing U nit
CTS	C lear to S end
EDV	E lektronische D atenverarbeitung
GNU	G NU's N ot U nix
GPL	G NU G eneral P ublic L icence
HLR	H ome L ocation R egister
HTML	H ypertext M arkup L anguage
HTTP	H ypertext T ransfer P rotocol
IEEE	The Institute of E lectrical and E lectronics E ngineers
IMSI	I nternational M obile S tation Identifier
ISO	I nternational S tandards O rganisation
ISP	I nternet S ervice P rovider
IT	I nformationstechnologie
JDBC	J ava D atabase C onnection Standard
JSP	J ava S erver P ages
JVM	J ava V irtual M achine
LAMP	L inux A pache M ySQL P HP
LAN	L ocal A rea N etwork
NASDAQ	N ational A ssociation of S ecurities D ealers A utomated Q otations
NEMAX	N euer M arkt A ktien Index
ODBC	O pen D atabase C onnection Standard
PDA	P ersonal D igital A ssistant
PDF	A dobe P ortable D ocument F ormat
PHP	P HP: H ypertext P reprocessor
RDBMS	R elational D atabase M anagement S ystem
RTS	R equest to S end

SQL	Structured Query Language
TCO	Total Cost of Ownership
UMTS	Universal Mobile Telephone System
UPS	United Parcel Service
USV	Unterbrechungsfreie Stormversorgung
WAN	Wide Area Network
WAP / WAP2	Wireless Application Protocol / version 2
WLAN	Wireless Local Area Network
WML	Wireless Markup Language
XHTML	Extended Hypertext Markup Language
XML	Extended Markup Language
bzw.	beziehungsweise
ca.	circa
ggf.	gegebenfalls
vs.	versus

Sinnesgleichen Namen im Arbeit:

Apache HTTP Server, Apache Server, Apache Web Server, Apache
MySQL RDBMS, MySQL Server, MySQL Datenbank Server, MySQL

1. Einleitung

“All men by nature desire knowledge.”

Aristotle, Metaphysics

Die Erfindung der Schrift kann als Ergebnis des Bedarfs an Erfassung von Wissen und Informationen zur Weitergabe an die nächsten Generationen betrachtet werden. Die historischen Steintafeln, eine der ersten Datenträger zur Erfassung von Informationen, können die heutige Gesellschaft davon überzeugen, dass die Strategie zur Datenerhebung erfolgreich war. Mit Erfindung der Druckkunst konnten die erfassten Informationen kostengünstig vervielfältigt werden, ein Entwicklung die zur heutigen Informationsgesellschaft geführt hat. Heute bewegen sich die Informationen in elektronischer Form in Sekunden um die ganze Welt und die Gefahr eines Informationsverlustes besteht nicht mehr, da alle Informationen ohne entscheidenden Kostenaufwand vervielfältigt werden können.

Die raschen Entwicklungen in der Datenerfassung, -bearbeitung und der Kommunikationstechnik, insbesondere das Internet, bringen große Veränderungen im Alltagsleben. Der Bedarf an aktuellen Informationen kann längst nicht mehr mit den älteren Informationsaustauschformen wie z.B. Büchern, Zeitschriften und Datenträgersammlungen gedeckt werden. Diese Informationsübertragungsformen auf bedruckten oder elektronischen Medien werden durch Netzwerkübertragungen ersetzt. Datenbankbasierte Webseiten erfüllen diesen Bedarf in der Kombination mit automatisierten Datenerfassungssystemen und fortschrittlichen Plausibilitätskontrollen.

Neben dem Bedarf an schneller Übertragung der aktuellen Daten entstand in den letzten Jahren der zusätzliche Wunsch nach Mobilität. Die Erfüllung dieser Wünsche liegt in den technischen Entwicklungen der mobilen Internetzugänge, dessen Funktion aus der Bereitstellung der gewöhnlichen Internetdienste über einen kabellosen und ortsunabhängigen Internetzugang besteht.

Die Datenerfassung ist, wie in allen Branchen der Wirtschaft, auch in der Landwirtschaft ein unverzichtbarer Bestandteil. Die effektive Nutzung der erfassten Daten in der Landwirtschaft kann entscheidende Auswirkungen auf die Wirtschaftlichkeit eines Betriebes haben, da sich die Optimierung der Produktions-

und Vermarktungsabläufe auf die Auswertung der erfassten Daten beziehen. Schlechte Abrufbarkeit sowie mangelhafte Datenorganisation sind die hauptsächlichen Ursachen des Effektivitätsproblems. Daten, durch dessen Erfassung und Speicherung Kosten entstehen, sind nutzlos, wenn sie nicht leicht abruf- und verwendbar sind.

Das Ziel dieser Arbeit ist die Suche nach Einsatzmöglichkeiten von mobilen Internetzugängen zur effektiven Nutzung landwirtschaftlicher Datenbestände. Hierfür soll ein Programm konzipiert werden, dessen Funktionalität aus der Aufnahme und Darstellung der landwirtschaftlichen Daten über mobile Internetzugänge besteht. Der Datenbestand zu diesem Programm wurde vom Hauptverband des Traber-Zucht und -Rennens e.V. (HVT) zur Verfügung gestellt und beinhaltet Daten über Trabrennpferde und -sport in Deutschland. Als praktisches Anwendungsbeispiel ist vorgesehen, dass das Programm die Darstellung der aktuellen Informationen über Traber und Trabrennsportveranstaltungen sowie die Möglichkeit der Abgabe eines Wettscheins für ein aktuelles Rennen enthalten soll.

2. Ausgangssituation

2.1. Wirtschaft im Internet: Wiedererfindung einer alten Wirtschaftsstruktur

Die gesamte Systematik der klassischen Vermarktung ist über die Kommunikation in eine Richtung geregelt: die Werbung. Die Firmen machen Werbung, um ihre Produkte bei den Kunden bekannt zu machen. Die Reaktionen der Kunden auf diese Werbungen kommen über einen indirekten Weg zurück, über Verkaufszahlen. Am Anfang wurde der Handel im Internet in diese Schablone gezwungen. Das galt nicht nur für die Kunden im Internet, sondern auch für die ersten Inhaltsherausgeber und Vermarkter, die ihre Geschäfte ins Internet getragen haben: Internet war eine Art Werbeplakat für Firmen, eine billige Methode zur Vermarktung über elektronische Form der Briefe oder ein Schaufenster, das auch nachts beleuchtet war. Im Laufe der Zeit hat sich das Internet zu einer altbekannten Wirtschaftsstruktur gewandelt, zu einem Marktplatz. Ein Marktplatz ist nicht eine Straße mit Schaufenstern und Werbeplakaten, sondern eine Art soziales Netz, wo die Verkäufer und die Kunden miteinander aktiv kommunizieren und handeln. Die Kunden unterhalten sich untereinander über die Produkte, das Verhalten der Verkäufer und die Preise. Die Verkäufer stehen in Konkurrenz zueinander. Die Rolle der Kunden und Verkäufer ist im ständigen Wechsel, wobei jeder Verkäufer auch Kunde eines anderen Verkäufers sein kann (HESSLER, 2000). Der Handel im Internet heute passt sich sehr gut in die Rolle eines Marktplatzes ein und bietet noch weit mehr: Dieser Marktplatz ist weltweit groß, ständig offen und was die Vielfalt der Produkte angeht, kann keine andere Wirtschaftsstruktur ihn übertreffen. In diesem Markt gibt es keine Raritäten aus einem fernen Land, da es kein „fernes Land“ mehr gibt.¹ Mit Unterstützung der Warenlieferungs- und Geldtransferfirmen wie UPS und Paypal kennt die Internetwirtschaft keine Grenzen, auch wenn die Produzenten einer Ware sich dieses gewünscht hätten².

¹ Dies ist eine sehr optimistische Sicht zu diesem Thema. Das Internet ist in vielen Ländern nicht genügend verbreitet, um sagen zu können, dass diese Länder auch ein Teil dieses Marktes sind.

² Eine Sendung aus Südkorea erreicht mit normalem Kargoservice Europa in zwei Wochen. Während der Vorbereitung dieses Dokuments spielen tausende von Playstation-Fans in Europa seit neun Monaten mit ihrer neuen Playstation Portable (PSP), die sie über das Internet aus Südkorea bestellt haben, wobei die Markteinführung des PSP in Europa laut Hersteller Sony erst am 1. September 2005 ist.

Wie ist das Internet so schnell in das Alltagsleben eingedrungen? Die Menschen haben das Internet nicht als eine neue Technologie, sondern als eine Erweiterung der bisherigen Kommunikationsarten wahrgenommen (CASTELLS, 2000). Diese Wahrnehmung hat dem Internet geholfen, in dem es die Anpassungsdauer deutlich verringert hat. Menschen neigen im Alltag zu einer Kommunikationsgeschwindigkeit, die sich der Sprache angleicht. Das ist der Grund, warum Menschen lieber fernsehen als Zeitung lesen oder lieber reden als schreiben. Durch die Verbreitung der Breitband-Internetverbindung in die Haushalte wurden die letzten technischen Begrenzungen aufgehoben, die auf dem Weg zu einer reibungslosen schnellen Kommunikation lagen. Heute kann man alle gewöhnlichen Kommunikationsmedien durch das Internet ersetzen. Telefonieren über das Internet ist schon längst möglich, man kann es auch durch Videokonferenz bereichern. Empfang der Fernseh- und Radiosendungen über das Internet sind noch in der Entwicklungsphase, aber es wird nicht mehr lange dauern, bis ein Netzwerkkabel den Kabelanschluss ersetzt.

Diese Veränderungen werden von der Wirtschaft stark unterstützt, da die Kosten der Geschäftsabwicklung über das Internet im Vergleich zu den herkömmlichen Abwicklungen deutlich geringer sind. In einem Internetgeschäft muss kein Verkaufspersonal arbeiten, es gibt keine physische Präsenz des Ladens; die Miet-, Steuer- und Instandhaltungskosten fallen ebenfalls weg.

Bis die Internetwirtschaft den heutigen Stand erreicht hatte, wurde sie für eine Weile von der Wirtschaftswelt als eine „nebenbei Aktivität“ gesehen. Grund dafür war, dass die größten dot.com-Firmen bis Mitte der 90'er nur rote Zahlen schrieben. Ab 1996 begann die Wirtschaft im Internet zu florieren. Grund dafür war nicht nur der wirtschaftliche Vorteil der Vermarktung von Waren über das Internet gegenüber der klassischen Vermarktung, sondern auch die Spekulationen der Börsenwelt mit Anlagen der Internetfirmen. Zwischen dem letzten Quartal des Jahres 1998 und dem Ende des ersten Quartals des Jahres 2000 haben viele Investoren auf der Suche nach schnellem Profit mit Penny-Stock-Aktien der neu gegründeten Internet- und Technologiefirmen stark spekuliert (BUCKMAN, 2000). Durch diese Spekulationen entstand eine Spekulationsblase³, die sich in der Technologiebörse NASDAQ in den USA sowie auch im NEMAX in Deutschland bemerkbar machte

³ Eine Spekulationsblase bezeichnet einen überkaufte Markt, in der Regel im Börsen-, Rohstoff- oder im Immobilienbereich, der sich von der realen Wirtschaftsentwicklung völlig abkoppelt. Getrieben von Berichten über raschen und mühelosen Reichtum springen immer mehr Anleger auf den Zug der Spekulation. Die Preise scheinen unbegrenzt immer weiter zu steigen, bis sich dann irgendwann die Erkenntnis des realen Wertes des jeweiligen Spekulationsobjektes durchsetzt und Käufer zu höheren Preisen plötzlich ausbleiben. (Quelle: Wikipedia [WIK])

und als „Dot-Com-Boom“ oder „Dot-Com-Blase“ bezeichnet wurde. Der NASDAQ-Index erreichte am 10. März 2000 mit 5048,62 Punkten seinen Höchststand und signalisierte den Anfang vom Ende des Dot-Com-Booms. In Deutschland war die Spekulationsblase eng mit dem Börsengang der 200 Millionen staatseigenen Deutsche-Telekom-Aktien verbunden, dessen Wert erst über 100 Euro erreichte und dann bis auf 8 Euro gefallen war.

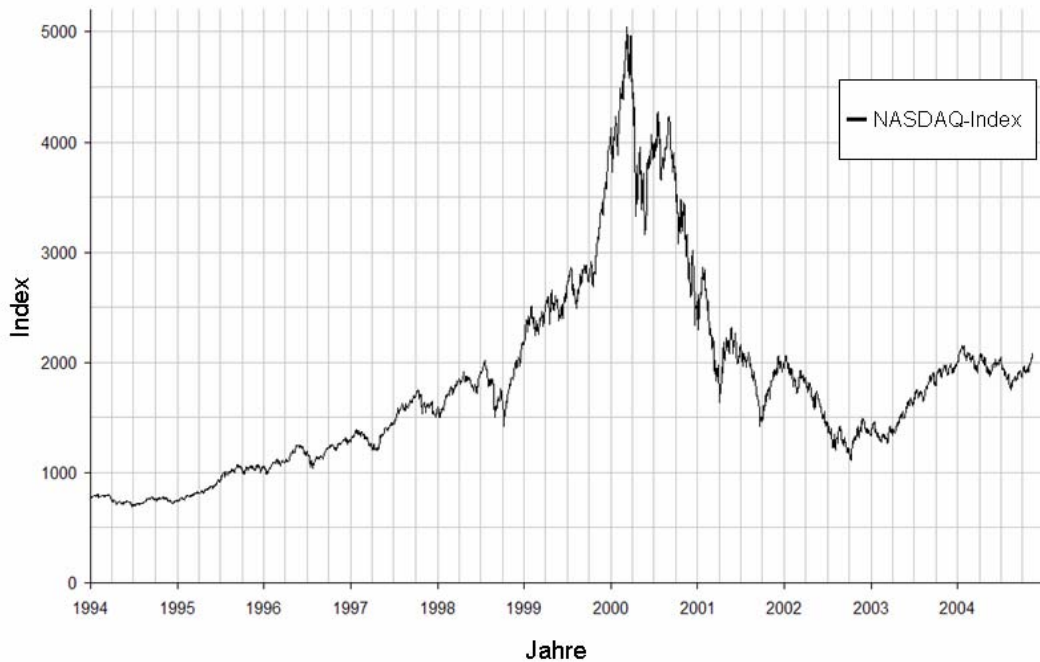


Abbildung 1: Graphischer Verlauf des Dot-Com-Booms und -Crashes in NASDAQ (Quelle: NASDAQ)

Im Anschluss an den Dot-Com-Boom ereignete sich der Börsencrash, genannt „Dot-Com-Crash“, wobei viele Internetfirmen, die keine Infrastruktur zur Unterstützung ihrer Vermarktungsstrategie und Lieferkanäle hatten und überwiegend im Internet existent waren, durch Verlust der finanziellen Unterstützung Ihrer Investoren Konkurs anmelden mussten. Für eine Weile wurde das Geschäftsleben im Internet von den übrig gebliebenen Dot-Com-Firmen dominiert, die durch den Börsencrash einen größeren Spielraum bekamen, da die potentielle Konkurrenz nicht mehr am Markt war. Das führte zu einem monopolähnlichen Status dieser Firmen. Gute Beispiele dafür sind die Firmen Ebay und Amazon. Obwohl der Zugang in diesen Markt nicht begrenzt ist, wagen sich keine der großen Investoren gegen diese Firmen anzutreten, somit bleiben diese Firmen zurzeit konkurrenzlos. Die Vermarktung der allgemeinzugänglichen Waren im Internet bleibt dadurch unter

Kontrolle der etablierten Konzerne. Die Neuzugänge haben bessere Chancen bei der Vermarktung von speziellen Gütern und Dienstleistungen.

2.2. Sicherheit

2.2.1. Definition der Sicherheit

Sicherheit wird von der Deutschen Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE als „einen Zustand, der frei von unvermeidbaren Risiken der Beeinträchtigung ist oder als gefahrenfrei angesehen wird“ definiert (DKE 2004).

Schneier erklärt die Grundregeln zur Definition der Sicherheit wie folgt (SCHNEIER, 2003):

1. Sicherheit beschäftigt sich mit Verhinderung. Ein Sicherheitssystem besteht aus Maßnahmen zur Verhinderung der Ereignisse mit nachteiligen Konsequenzen.
2. Sicherheit richtet sich an Aktionen mit Absicht. Das deutet auf eine wichtige Unterscheidung hin: Maßnahmen gegen Aktionen mit Absicht werden von Sicherheit (engl. security) umfasst, hingegen Maßnahmen gegen unabsichtliche Aktionen unter Schutz (engl. safety) eingeordnet.
3. Diese absichtlichen Aktionen sind unbefugt. Das bedeutet nicht unbedingt, dass diese Aktionen auch illegal sind.
4. Sicherheit erfordert das Konzept eines Angreifers, der diese absichtlichen unbefugten Aktionen durchführt. Der Begriff „Angreifer“ ist nicht pejorativ sondern werteneutral gemeint und macht keine Annahmen über Moralität oder Legalität des Angriffs.
5. Die absichtlichen unbefugten Aktionen eines Angreifers werden Angriff genannt. Ein Angriff ist ein spezifischer Weg um zu versuchen, die Sicherheit oder eine Komponente eines Systems zu durchbrechen.
6. Das Ziel eines Angriffs wird Wertsache (engl. assets) genannt.
7. Sicherheit kann auch als die Mechanismen angesehen werden, die ein gewisses Maß an Schutz anbieten. Diese werden Gegenmaßnahmen

genannt und umfassen alles (Türschlösser, ID-Karten, Mauern, Alarmanlagen, Firewall-Software, ...), das zum Schutz einer Wertsache dienen könnte. Ein Sicherheitssystem besteht aus einer Serie von Gegenmaßnahmen, die zum effektiven Schutz einer Wertsache erforderlich sind.

Für eine effektive Sicherheit sollte die Bedrohung der Wertsache und das Risiko analysiert werden. Bedrohung definiert eine mögliche Form des Angriffs (z.B. beim Schwimmen von einem Haifisch gebissen zu werden) und das Risiko definiert die Wahrscheinlichkeit der Realisierung dieser Bedrohung und dessen Effekt auf die Wertsache im Fall einer Realisierung. Einige Bedrohungen haben sehr große Effekte im Fall der Realisierung (z.B. Inbesitznahme einer Nuklearwaffe durch Terroristen), sodass man zu sehr hohen Sicherheitsmaßnahmen greifen sollte, obwohl die Wahrscheinlichkeit der Realisierung der Bedrohung ziemlich gering ist. Dagegen sind die Nachwirkungen anderer Bedrohungen (z.B. Autodiebstahl) ziemlich gering; trotz der relativ hohen Wahrscheinlichkeit zur Realisierung dieser Bedrohung genügen sehr moderate Sicherheitsmaßnahmen (z.B. Abschließen des Autos).

Der Faktor Mensch ist entscheidend bei der Analyse der Bedrohungen, Schätzung der Risiken und Einleiten der Sicherheitsmaßnahmen. Die menschliche Natur verhindert meistens eine objektive Bewertung des Risikos. Die Risiken für ungewöhnliche Bedrohungen (z.B. Umkommen bei einem Flugzeugabsturz) werden übertrieben, hingegen die Risiken für gewöhnliche Bedrohungen (z.B. Umkommen bei einem Verkehrsunfall) werden unterschätzt. Eine andere Ursache der Fehleinschätzung des Risikos liegt an der subjektiven Wahrnehmung des Menschen: Wenn der Mensch das Gefühl der Kontrolle gegenüber einer Bedrohung hat (z.B. beim Autofahren), wird das Risiko unterschätzt, andererseits aber wird bei der Empfindung von Hilflosigkeit das Risiko überschätzt.

Es gibt keine absolute Sicherheit in der realen Welt. Man kann nur von relativer Sicherheit reden, dessen Effektivität von der Erkennung und Bewertung der Bedrohungen und dem Einsatz von vorbeugenden Sicherheitsmaßnahmen abhängt. Das Ziel von Sicherheitsmaßnahmen ist nicht die Abschaffung der Bedrohung, sondern die Reduzierung des Risikos zu einer vertretbaren Stufe. Alle Sicherheitsmaßnahmen sind mit Kosten und Kompromissen verbunden. Die Kosten der Sicherheit nehmen mit weiteren Sicherheitsmaßnahmen zu, wobei der marginale Gewinn an Sicherheit abnimmt. Abgesehen von den Kosten sind die

Kompromisse, die man für die Realisierung der Sicherheit treffen soll, entscheidend bei der Auswahl an Sicherheitsmaßnahmen. Diese Kompromisse hängen von der Bereitschaft der Menschen ab, wie viel sie von ihrer Privatsphäre, Freiheit und ihrem Komfort der alltäglichen Gewohnheiten für die Sicherheit abzugeben bereit wären. Diese Bereitschaft ist mit der Bedrohung und dem Risiko und dessen Wahrnehmung durch den Menschen verbunden: Fast jede Person wird problemlos zustimmen, vor einem Flug durchsucht zu werden, da das Risiko eines Bombenangriffs vom Gefühl her höher ist. Dieselbe Prozedur könnte man bei einer ICE-Zugfahrt nicht durchführen, obwohl die Bedrohung nicht weniger real und das Risiko (da die Anzahl der Reiseteilnehmer höher als bei einem Flugzeug ist) höher ist. In terrorbedrohten Ländern (z.B. Türkei und Israel) ist eine Durchsuchung beim Eintritt in ein Einkaufszentrum nahezu selbstverständlich, weil die Menschen dazu bereit sind, einen Teil ihrer Privatsphäre gegen eine durch diese Prozedur gewonnene zusätzliche Sicherheit einzutauschen.

Sicherheitslösungen bestehen aus einer Serie von Prozeduren die parallel oder in Reihe durchgeführt werden. Das führt dazu, dass die gesamte Sicherheit nur so stark wie das schwächste Glied dieser Prozedurenkette ist.

2.2.2. Sicherheit im Netzwerk und Internet

Die Sicherheit der EDV-Systeme kann in drei Gruppen gegliedert werden:

- Physikalischer Schutz der Hardware
- Schutz der Software
- Schutz der Informationen

Abgesehen von dem Schutz der speziellen EDV-Hardware (z.B. militärische EDV-Hardware) kann die Sicherheit der Hardware ähnlich wie die Sicherheit von anderen Wertsachen gesehen werden. Man kann Hardware mit gewöhnlichen Sicherheitsmaßnahmen wie Mauern, Schlösser, Videoüberwachung und Alarmanlagen schützen. Die eigentlichen Ziele der Angriffe im EDV-Bereich betreffen die Software und Informationen, wobei die Software auch meistens nur für einen Zugang zu den Informationen angegriffen wird.

Die Sicherheit von Software und Information sind eng mit einander verbunden, da die Information die Materie ist, die von Software bearbeitet bzw. produziert wird. Trotzdem sollte man die Software und die Information differenziert betrachten:

Software ist ein für EDV-Zwecke spezialisierter Mechanismus zur Bearbeitung der Information und da sie für ihre Funktionalität Hardware benötigt, befindet sie sich eigentlich nur in elektronischer Form. Information wird von Informationswissenschaftlern als „(geglückte) Übertragung von Wissen oder Wissen in Aktion“ (HAMMWÖHNER et al., 2004) definiert. Daraus kann man ableiten, dass die Information unabhängig von der Form ist und es verändert ihre Natur nicht, egal ob sie sich in elektronischer oder in einer anderen Form befindet. Die Information ist wertvoll für die Beteiligten des Wissenstransfers und die Sicherheit dieser Information ist mit ihrem ökonomischen Wert verbunden (NOLL, 1994), da der Umfang der Sicherheitsmaßnahmen durch dem ökonomischen Wert der Information begrenzt ist.

Computernetzwerke sind für die Übertragung von Informationen geschaffen. Das Internet, sowie die modernen Netzwerke funktionieren nach dem End-to-End-Prinzip, wobei jede Datenübertragung zwischen zwei Endpunkten geschieht. Zwischen diesen Endpunkten kann es zahlreiche Zwischenpunkte (z.B. Gateways, Router und Firewalls) geben, über die die Datenpakete übertragen werden (GARFINKEL, et al., 2003). Folgende Abbildung stellt eine typische Datenübertragung über das Internet schematisch dar:

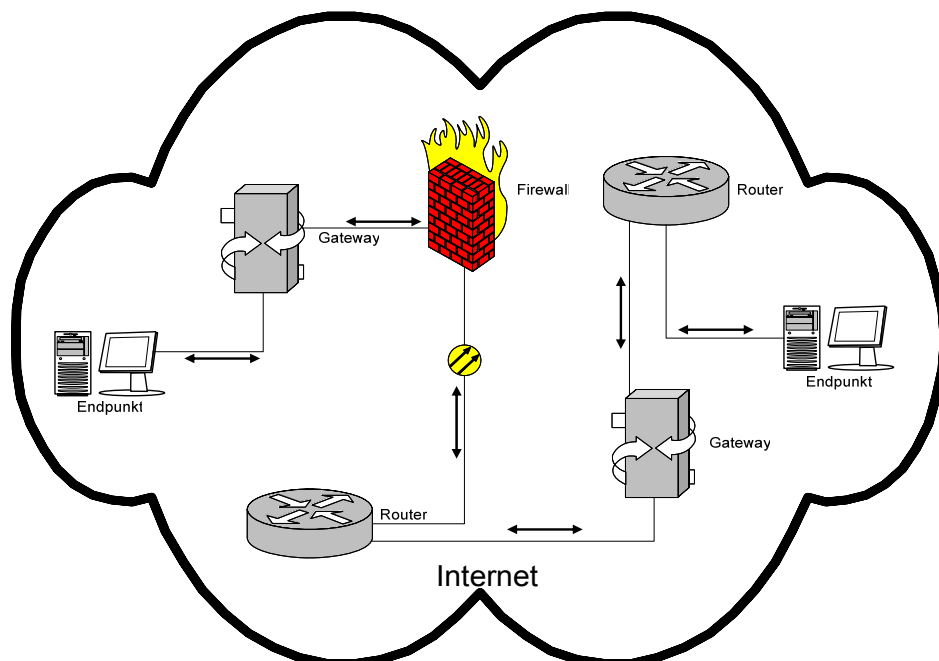


Abbildung 2: Schematische Darstellung einer Datenübertragung über das Internet

Die über das Netzwerk zu übertragenden Informationen werden erst auf kleine Datenpakete verteilt und dann über das Netzwerk verschickt. An der

Empfängerseite werden diese Pakete geöffnet und die aufgeteilten Informationen wieder zusammengefügt (BRADEN, 1989). Die Datenpakete beinhalten zusätzliche Informationen über Herkunftsadresse und Zieladresse des Pakets, wie bei einem Postpaket. Sie werden auch wie ein Postpaket geliefert: Jeder Zwischenpunkt sendet das Paket an einen weiteren Zwischenpunkt, der näher an der Zieladresse liegt, wobei keine der Zwischenpunkte die gesamte Zieladresse erkennt und sich mit dem Inhalt des Datenpakets beschäftigt. Diese Netzwerkarchitektur ist mit dem Ziel der Einfachheit entstanden und bietet keine Sicherheit, da die zu übertragenden Datenpakete von allen Zwischenpunkten gelesen, gespeichert und sogar manipuliert werden könnten.

Die Informationssicherheit im EDV-Bereich beschäftigt sich mit dem Schutz der Daten und Datenübertragungen. Man kann die möglichen Bedrohungen in drei Gruppen gliedern (SCHNEIER, 1999; 2000a):

1. Angriffe zur Störung der Funktionalität eines Datensystems oder zur Zerstörung des Datensystems
2. Angriffe zur Erlangung der Informationen oder übertragbaren Wertsachen
3. Angriffe zur Manipulation eines Datensystems oder von Informationen

Angriffsmotive

Neben der Analyse der Bedrohungen, ist es auch sinnvoll zu verstehen, was für Motive die Angreifer haben:

1. Publizität: Das Verlangen nach Publizität hat eine lange Geschichte in der realen Welt und es findet sich in der digitalen Welt wieder, seit dem die Angriffe auf Computersysteme das Interesse der Medien erweckt haben. Der Angreifer hat bei einer Publizitätsattacke keine kommerziellen Absichten, was zählt ist berühmt zu werden. Diese Motivation kann zu sehr destruktiven Angriffen führen, da das Interesse der Medien eigentlich mit der Größe des angerichteten Schadens verbunden ist.

2. Spionage: Diese Motivation kann viele Gründe haben, von einfacher Neugier bis zur militärischen Spionage⁴ und im Gegensatz zu den Publizitätsattacken möchte der Angreifer möglichst unbemerkt bleiben.
3. Kommerziell: Wie die häufigsten Verbrechen in der realen Welt zur illegalen Bereicherung begangen werden, spiegelt sich die Motivation genauso im Internet. Der Angreifer hat die Absicht, durch seine Aktionen kommerzielle Vorteile zu erlangen.

Häufig bilden die kommerziellen Zwecke und die Spionage ein gemeinsames Motiv, da die Informationen, die durch Spionage erbeutet wurden, in meisten Fällen für kommerzielle Zwecke verwendet werden.

Angriffsarten

Es gibt zahlreiche Angriffsarten gegen EDV-Systeme und Datenübertragungen, wobei die Auswahl einer Angriffsart hauptsächlich vom Angriffsmotiv abhängt. Z.B. würde ein Angreifer mit der Motivation zur Datenspionage die Funktionalität des Zielobjekts nicht stören wollen, womit ein Denial of Service (DoS) Angriff für diesen Zweck nicht in Frage käme.

Die wichtigsten Angriffsarten gegen EDV-Systeme und Datenübertragungen sind:

- Denial of Service (DoS)
- Identitätsdiebstahl (eng. Identity Theft)
- Man-In-The-Middle-Angriff
- Insider Angriffe

Denial of Service (DoS)

Diese Art von Angriff hat keine kommerziellen Zwecke, er ist nur für destruktive Zwecke da und das Motiv des Angriffs ist Publizität. Das Ziel dieses Angriffs ist die Störung der Funktionalität eines Systems und wobei die Schwächen des Betriebssystems oder Netzwerkprotokolls ausgenutzt werden. In den meisten Fällen sind die Angreifer nicht die Erfinder der Schwächen des Systems, sondern Personen, die kein vertieftes Wissen über Computersysteme haben. Die Erfinder sind meistens Sicherheitsexperten, Akademiker oder Computerfirmen, die die

⁴ Ein gutes Beispiel hierfür ist die IT-Spionage-Konkurrenz zwischen Nord- und Südkorea (McWILLIANS, 2003)

gefundenen Schwächen publizieren. Nur sehr gute Programmierer können mit Hilfe dieser Publikationen Tools programmieren, die diese Schwächen ausnutzen. Diese Tools werden hauptsächlich dafür programmiert, um gefundene Sicherheitslücken zu testen und stehen meistens unter Open Source Lizenzen im Internet zur Verfügung. Die Angreifer modifizieren diese Software für einen Angriff auf EDV-Systeme. Da diese modifizierten Tools möglicherweise im Internet zu finden sind, kann jede Person ohne besonderes Computerwissen solche Angriffe ausführen. Eine besondere Form der DoS-Attacken ist das Distributed-DoS (DDoS), bei dem mehrere Angreifer gleichzeitig ein System angreifen.

Identitätsdiebstahl

Identitätsdiebstahl kann in vielen Formen vorkommen, z.B. das Kopieren der Kreditkarteninformationen an einem Geldautomaten kann genauso wie die unbefugte Übernahme der IP-Adresse eines Rechners unter Identitätsdiebstahl eingeordnet werden. Das Grundprinzip bei dieser Angriffsart ist, dass der Angreifer sich für jemand anderen ausgibt, in dem er die Daten von dieser Person verwendet. Dieser Angriff muss nicht unbedingt zwischen Menschen passieren, auch ein Rechner kann in einer Netzwerkumgebung einen anderen Rechner imitieren (SCHNEIER, 2005a). Ein passendes Beispiel hierfür ist der heutzutage populäre E-Mail-Betrug, wobei die Personen per E-Mail aufgefordert werden, ihre Kontoinformationen auf der Webseite ihrer Bank zu aktualisieren. Die Personen, die auf die Verweise dieser E-Mail klicken, werden auf eine fremde Webseite geleitet, die die Webseite der Bank vortäuscht. Die Informationen, die auf dieser Webseite eingegeben werden, werden von den Angreifern gesammelt. Das Motiv des Identitätsdiebstahls ist meistens kommerziell, wobei es auch für Spionagezwecke eingesetzt werden kann.

Man-In-The-Middle-Angriff

Bei dieser Angriffsform befindet sich der Angreifer im Netzwerk zwischen zwei Kommunikationspartnern und hat dabei mit seinem System die Kontrolle über den Datenverkehr zwischen diesen Systemen und kann die Informationen nach Belieben einsehen und sogar manipulieren. Man-In-The-Middle-Angriffe sind schwer zu erkennen, da die Datenübertragung nicht gefährdet wird.

Insider Angriffe

Von Natur aus erwartet der Mensch die Gefahr immer von Außen. Aus diesem Grund verschließen die Menschen immer ihre Haustür, besitzen Alarmanlagen und Versicherungen. In einem EDV-System sind die meisten Sicherheitsmaßnahmen auch gegen die Gefahren von Außen gerichtet. Es kann aber auch vorkommen, dass das System von einem Insider (z.B. ein verwirrter Mitarbeiter) angegriffen wird. Diese Attacken werden kaum erwartet und die sind auch sehr schwer zu identifizieren, wenn der Angreifer vertiefte Kenntnisse vom System hat (SCHNEIER, 2005b; KEENEY et al., 2005).

Die gemeinsame Eigenschaft dieser Angriffsarten ist, dass sie die Schwachpunkte der Sicherheit ausnutzen. Bei einem DoS-Angriff werden die Schwächen des Betriebssystems und der Software ausgenutzt. Der Identitätsdiebstahl greift meistens den Faktor Mensch an, hingegen die Man-in-the-Middle-Angriff die Schwächen der Netzwerktopologie zu Gebrauch macht. Ein Insider-Angriff ist schwer vorzubeugen, wenn die innerbetrieblichen Sicherheitsmaßnahmen nicht effektiv eingesetzt werden.

Sicherheitsmaßnahmen

Der Einsatz von Firewallsystemen, ständige Überwachung der Netzwerkaktivitäten und regelmäßige Aktualisierungen der Softwareprodukte stellen die wichtigsten Maßnahmen dar, die man gegen diese Angriffe ergreifen kann. Folgende Punkte können zur Verbesserung der Sicherheit dienen:

- Verschlüsselungstechnologien
- Netzwerktechnische Sicherheitsmaßnahmen
- Maßnahmen gegen Social-Engineering

Verschlüsselungstechnologien

Die Verschlüsselungstechnologien dienen zum Schutz der Datenübertragung, da die übertragenen Daten, auch wenn sie mitgeschnitten werden, ohne den passenden Schlüssel nicht gelesen werden können. Der Einsatz einer unabhängigen Schlüsselautorität kann durchaus eine Lösung gegen Man-in-the-Middle-Angriffe sein, da sonst über das Netzwerk mitgeteilte Schlüssel auch von den Angreifern gelesen werden könnten.

Netzwerktechnische Maßnahmen

Viele Angriffsmethoden nutzen die Einfachheit des Internet Protokolls aus, wobei für die Datenübertragungen kaum begrenzende Regelungen vorgesehen sind (ALBERTS und DOROFEE, 2002). Mit netzwerktechnischen Maßnahmen wie Paket-Filter und Firewalls kann man den Datenverkehr im Netzwerk regeln und die gesamten Netzwerkaktivitäten überwachen.

Maßnahmen gegen Social-Engineering

Meinungsumfragen im Internet zeigen, dass viele Menschen dazu bereit wären, ihre Computerpasswörter gegen Schokolade einzutauschen (WAGNER, 2004). Social-Engineering ist die Kunst, die Menschen ohne Gewaltanwendung zu manipulieren, um an geheime Informationen zu gelangen oder die Menschen dazu zu bringen, einige Aktivitäten auszuführen, die für den Angreifer von Vorteil sind (MITNICK und SIMON, 2002). Man kann die höchstmögliche Sicherheitsstufe mit technischen Sicherheitsmaßnahmen erreichen, dennoch besteht die Gefahr, dass die Menschen, die diese Systeme verwenden, von Angreifern manipuliert werden können. Gegen Social-Engineering gibt es keine perfekten Lösungen, wobei die Sicherheitsschulung der Personen,⁵ die möglicherweise angegriffen werden können, viel zur Sicherheit beitragen kann (MITNICK und SIMON, 2005; SCHNEIER, 2000b).

2.3. Mobile Internetzugänge

2.3.1. Entwicklung der mobilen Internetzugänge

Bis vor ein paar Jahren konnte man einen Internetzugang relativ einfach beschreiben, da es sich hauptsächlich um einen PC handelte, der mit Hilfe einer Kommunikationshardware (Modem, ISDN-Adapter, Netzwerkkarte) über ein ISP (Internet Service Provider) den Zugang ins Internet hatte. Heute kann man zur Darstellung von Informationen aus dem Internet nicht mehr davon ausgehen, dass der Internetzugang auf der Benutzerseite über einen vollwertigen Computer erfolgt. Es gibt viele Wege für eine Verbindung ins Internet, bei denen man keine PCs benötigt, darunter auch mobile Internetzugänge. Mit diesem Begriff verbindet man alle tragbaren Geräte, unabhängig von Ihren primären Funktionen, die über eine Internetverbindung verfügen und dem Benutzer die Möglichkeit geben, diese Internetverbindung für Webnavigation und E-Mail-Funktionalität zu verwenden

⁵ z.B. Kundenservice Mitarbeiter sind sehr häufig Ziel der Social-Engineering-Methoden

(TUROWSKI und POUSTTCHI, 2004). In der Praxis versteht man unter dem Begriff „mobile Internetzugänge“ Mobiltelefone ab der zweiten Generation und PDAs mit Internetverbindung.

Durch die Einführung der dritten Generation Mobiltelefone in den letzten Jahren hat sich der Markt für mobile Internetzugänge stark entwickelt. Heutzutage ist es kaum möglich ein Mobiltelefon ohne Internetfunktionalität auf dem Markt zu finden. Die Verwendungsrichtung dieser Internetzugänge scheint sich je nach Altersklasse der Personen zu ändern; die Jugendlichen verwenden es um Bilder und Musik herunterzuladen, die Erwachsenen setzen es hauptsächlich als Informationsquelle für Wetterberichte, Staumeldungen und als Suchmaschine ein. Im Vergleich zu dem klassischen Internetzugang mit Hilfe eines Computers setzen sich mobile Internetzugänge schneller im Alltagsleben durch, da man hier keine speziellen Computerkenntnisse benötigt. Zusätzlich wird unter Jugendlichen der Besitz eines Mobiltelefons mit Internetfunktionalität als Statussymbol angesehen (HÖFLICH, 2001).

Es ist absehbar, dass in naher Zukunft die Mehrheit der Internetverbindungen über mobile Geräte erfolgen wird, wobei es sich überwiegend um Mobiltelefone handeln wird. Nach Schätzungen der International Telecommunication Union gab es im Jahr 2000 7.382.603.000 Mobiltelefonbenutzer, dessen Zahl im Jahr 2004 17.481.054.000 erreichte, womit 27,75% der Weltbevölkerung im Besitz eines Mobiltelefons sind. Die Penetranz der Mobiltelefone liegt bei 60,97% in den USA, 86,42% in Deutschland und 71,50% in Kontinentaleuropa. Hingegen liegt die geschätzte Anzahl der Internetbenutzer im Jahr 2004 bei 8.739.371.000, wobei die Penetranz der PCs weltweit bei nur 12,97% (USA: 74,06%, Deutschland: 56,10%, Europa: 29,24%) liegt [ITU]. Die Zahlen zeigen, dass die Mobiltelefone ein großes Potenzial für den Einsatz als Internetzugang haben.

2.3.2. Mobiltelefon vs. Computer

Mobiltelefone bieten viele technische, ergonomische und wirtschaftliche Vorteile gegenüber konventionellen Internetzugängen, obwohl sie nicht als Ersatz der klassischen Internetzugänge gesehen werden können, sondern eher als koexistierende Lösungen im Informationsalltag. Diese Vorteile lassen sich in folgende Themenbereiche gliedern:

- Mobilität und Bedienungsergonomie

- Penetranz im Alltagsleben
- Kosten
- Vermarktungsmodell
- Organisation der GSM-Netzbetreiber
- Netzwerkkumgebung

Mobilität und Bedienungsergonomie

Mobilität und einfache Bedienung sind wichtige Vorteile des Mobiltelefons im Vergleich zum Laptop und PDA. Aus ergonomischer Sicht können die Mobiltelefone als benachteiligt angesehen werden, da sie im Vergleich zu den herkömmlichen Internetzugängen mit kleinem Bildschirm und rudimentärer Tastatur ausgerüstet sind, wobei diese Eigenschaften aus der Sicht der Mobilität bestimmt werden. Diese Nachteile können sich in naher Zukunft ändern, z.B. mit Einsatz der faltbaren Bildschirme (KANEKO et al., 1996) und alternativer Eingabemethoden (KÖLSCH und TURK, 2002).

Obwohl es immer einen konstanten Markt für PCs und Laptops mit einer konventionellen Internetverbindung geben wird, sehen die Chancen für Mobiltelefone auch sehr gut aus, da die Mobilität und die einfache Bedienung eines Mobiltelefons von keinem Laptop und keinem PDA übertroffen werden kann. Ein Mobiltelefon bleibt im Normalfall ständig an und muss nicht gebootet werden, um zu funktionieren.

Desweiteren bieten Mobiltelefone spontane Verwendungsmöglichkeiten, die die Laptops nicht anbieten können. Zum Beispiel kann man mit einem Mobiltelefon die Wettervorhersage abrufen, während man auf den Aufzug wartet, oder sich über das Ergebnis eines Fußballspiels informieren, wenn man in der Warteschlange im Supermarkt steht. Solche spontanen Nutzungen würden mit einem Laptop nicht so bequem und unkompliziert möglich sein. Zur Verwendung eines üblichen Mobiltelefons benötigt man nur eine Hand, hingegen muss beim PDA das Gerät mit der einen Hand gehalten und mit der anderen betätigt werden. Besonders unter Jugendlichen ist zu beobachten, dass die Texteingabe mit Hilfe des T-9-Wörterbuchsystems schneller als eine vollwertige Tastatur gemacht werden kann.

Penetranz im Alltagsleben

Wie die statistischen Schätzungen der International Telecommunication Union zeigen, haben die Mobiltelefone im Vergleich zu den PCs einen deutlich höheren Anteil am Markt. Rein zahlentechnisch gesehen sind die Mobiltelefone schon jetzt die dominierenden Internetzugänge, wobei leider die meisten Mobiltelefonbesitzer die Datenübertragungseigenschaften dieser Geräte nicht verwenden. Die Tendenz der technologischen Entwicklungen zeigt, dass die Nutzung von Datenübertragungseigenschaften der Mobiltelefone in der nahen Zukunft zunehmen wird, da sie ständig mit neueren Multimedia- und EDV-Funktionen ausgestattet werden. Diese Tendenz wird von GSM-Netzbetreibern stark unterstützt. Der Grund dieser Unterstützung liegt an der ständig abnehmenden Gewinnspanne der normalen Telefongespräche über das GSM-Netz aufgrund der starken Konkurrenz zwischen den Netzbetreibern. Der abnehmende Profit könnte mit den Erträgen aus den Datenübertragungsdiensten gedeckt werden. Ein gutes Beispiel hierfür sind die Dienste der Firma NTT DoCoMo aus Japan, die ihren Abonnenten Zugriff auf Videos, Musik und Informationen wie Straßenkarten, Börseninfos, Gelbe Seiten und Wetterberichte sowie Reservierungsmöglichkeiten für Hotels, Restaurants und Events mit einer hohen Bandbreite anbietet [NTT].

Kosten

Die Mobiltelefone werden heute nicht als besondere Produkte betrachtet, sondern in dem Vermarktungssegment der Konsumentenelektronik eingeordnet, welches die PC-Produzenten seit über zwanzig Jahren nicht schaffen konnten. Eine Nebenwirkung der hohen Penetranz ist die Senkung der Preise für die Mobiltelefone und auch für die Dienstleistungen. Mobiltelefone werden in sehr großen Stückzahlen produziert und haben eine mittlere Lebensspanne von zwei Jahren, die durch das biennale Vertragsmodell geprägt wird.

Vermarktungsmodell

Die GSM-Netzbetreiber verkaufen die Mobiltelefone mit großen Preisnachlässen, teilweise sogar kostenlos im Rahmen eines zweijährigen Abonnementvertrags, bei dem meistens eine monatliche Nutzungsgebühr oder ein Mindestumsatz festgesetzt wird. Durch diese Gebühren werden die subventionierten Mobiltelefone finanziert. Die Kunden bekommen jedes zweite Jahr ein aktuelles Mobiltelefon zu Sonderkonditionen, womit sich der Vertrag um zwei Jahre verlängert. Somit kann

man sicher sein, dass die überwiegende Mehrheit der Mobiltelefone im Besitz der GSM-Kunden über aktuelle Technologien verfügen. Hieraus ergibt sich, dass die Durchsetzungsdauer neuer Technologien in der Mobiltelefonlandschaft ungefähr zwei Jahre beträgt. Eine derartige Aktualität der technischen Eigenschaften kann im PC und PDA Bereich nicht erwartet werden, da die Neukäufe dieser Produkte nicht derart begünstigt werden.

Ein ähnliches Finanzierungssystem wie bei den Mobiltelefonen wurde in der Vergangenheit für PCs und Palm PDAs von den Firmen AOL und 3com mit dem Zweck der Vermarktung einer Internetverbindung probiert, aber scheiterte aufgrund mangelnden Kundeninteresses.

Organisation der GSM-Netzbetreiber

Die Eintrittskosten in den GSM-Markt sind deutlich höher als die für den ISP-Markt, da die Investoren für die Betriebszulassung eines Mobilfunknetzes hohe Lizenzgebühren zahlen und mit hohen Hardwareinvestitionen rechnen müssen [BNA]. Das führt zu einer oligopolen Marktsituation, in der eine kleine Anzahl von Firmen mit vergleichbaren Kostenfaktoren operiert. Zudem werden diese Firmen zur Zusammenarbeit gezwungen, indem sie ihre Mobilfunkhardware für Roaming⁶ der Kunden anderer Netzbetreiber zur Verfügung stellen müssen. Dadurch kann der Kunde ohne es zu merken überall mit demselben Mobiltelefon telefonieren und erhält nur eine Rechnung von seinem eigenen Netzbetreiber. Ein zusätzlicher Vorteil des Roamings ist die Einhaltung der Emissionsschutzgrenzen für Mobilfunkstrahlung [BFS], da alle Netzbetreiber dieselbe Antenneninstallation verwenden können. Eine ähnliche Zusammenarbeit in dem ISP-Markt ist bisher nicht gelungen und es gibt keine etablierte Lösung für einen globalen Internetzugang.

Netzwerkumgebung

Das Internet wurde unter dem Gesichtspunkt der Einfachheit konzipiert, da der Zweck die Funktionsfähigkeit in einem Atomkrieg sein sollte (DENNING, 1989). Deshalb bietet es ganz einfache Netzwerkfunktionalitäten, wobei alle Kommunikationsaktivitäten, die man heute über das Internet durchführen kann, auf diese grundsätzlichen Netzwerkfunktionen aufbauen. Dagegen haben

⁶ Roaming bezeichnet die Nutzung eines Mobilfunkgerätes in einem anderen Netzwerk als dem Heimatnetzwerk.

Mobilfunknetze keine hohen Ambitionen wie das Überleben eines Atomkriegs, sondern sie wurden dazu konzipiert, möglichst viele Funktionalitäten anzubieten. Ein großes Problem des Internets ist es, dass die IP-Adressen nur netzwerktopologisch identifizierbar sind. Es gibt keine Möglichkeiten einen Rechner durch seine IP-Adresse zu identifizieren und genau zu lokalisieren. Dagegen sind die IP-Adressen bei Mobiltelefonen immer mit der IMSI-Nummer (International Mobile Station Identifier) verbunden und dadurch sind die Mobiltelefone unabhängig von der Netzwerktopologie identifizierbar. Diese Identifizierung ist mit der HLR-Liste (Home Location Register) verbunden, die die Informationen über den Ort des Mobiltelefons speichert. Somit können die Mobiltelefone nicht nur identifiziert, sondern auch dynamisch lokalisiert werden. Die zahlreichen Datenkanäle des Mobilfunknetzes können als ein weiterer Vorteil angesehen werden, wobei einige Kanäle nur für die Kontrollsignale der Datenübertragung (RTS, CTS, ACK) und andere für Daten- und Sprachübertragung eingesetzt werden. Dieses führt zur besseren Kontrolle des Datenverkehrs im Vergleich zu den Wireless-LAN-Datenübertragung (IEEE 801.11 Protokoll), bei der der gesamte Datenverkehr über einen Kanal läuft (BAIOCCHI et al., 2004). Darüber hinaus sind die Frequenzbereiche der GSM-Netze reserviert und somit gibt es weniger Interferenzen von anderen Systemen, die ein Störsignal erzeugen können⁷. Die Bandbreite der einzelnen Datenübertragungskanäle pro Mobilfunkgerät ist auch reserviert, dadurch wird sichergestellt, dass die gesamte Bandbreite nicht von anderen Datenübertragungen besetzt werden kann, was häufig bei Wireless-LAN-Netzwerken vorkommt (BHARGAVAN et al., 1994). Die Unterstützung von IPv6 (Internet Protokoll Version 6) ist im IP-Protokoll der neueren Mobilfunknetze (4. Generation) implementiert, im Gegensatz zum Internet, bei dem das Update zu IPv6 seit 15 Jahren nicht durchgesetzt werden konnte (GAZIS et al., 2003).

⁷ Ein gutes Beispiel dafür ist, dass man die Funktionalität eines Wireless-LAN-Netzwerks mit einem schlecht abgeschirmten Mikrowellenofen stark stören kann.

2.4. Wettgeschäft im Internet

Parallel zur Verbreitung des Internets in Privathaushalte nahm die Anzahl der Glücksspielanbieter über das Internet deutlich zu. Neben den Casinos, die nur im Internet existieren und ihre Türen für die ganze Welt⁸ öffnen, wurden auch Internetwettannahmestellen eröffnet, die unter anderem auch Pferdewetten anbieten.

In den letzten Jahren nimmt der Anteil der Internetwetten an den gesamten Wetten ständig zu, wobei der Anteil der Wetten, die auf der Rennbahn gespielt werden, abnehmen. Die möglichen Ursachen dieser Entwicklung liegen vermutlich an der Benutzerfreundlichkeit der Internetwetten. Internetwetten bieten folgende Vorteile für den Kunden:

- Der Wettschein ist in den meisten Fällen interaktiv. Das ermöglicht dem Kunden, noch während des Ausfüllens des Wettscheins sich über die Kosten und mögliche Tippfehler zu informieren.
- Die Darstellung eines Wettscheins auf einem Bildschirm wirkt meistens übersichtlicher als ein Wettschein in Papierform. Darüber hinaus kann der Internetwettanbieter den Wettschein in verschiedenen einfacheren Formaten darstellen und der Kunde hat die Wahl zwischen mehreren Formaten.
- Die Zahlungsmethoden sind vielfältiger als auf einer Rennbahn.
- Man hat ständig Zugriff auf aktuelle Quoten und andere Renninformationen.
- Man kann sich immer und überall über die Rennveranstaltungen informieren und Wettscheine abgeben.
- Es müssen keine Scheine aufbewahrt werden, um Gewinne einzulösen.

Bei diesen Vorteilen ist es nicht überraschend, dass die Kunden Internetwetten präferieren. Dazu kommen noch neue Kunden, die keine Pferderennenkenner, sondern nur an Glücksspielen interessiert sind.

Die Internetbuchmacher, die als Erweiterung der klassischen Buchmacher fungieren, leiten alle Wettscheine an den Totalisator weiter, den alle Buchmacher verwenden. Deswegen kann die Internetwette nicht als eine Ursache der Wettumsatzverluste angesehen werden.

⁸ Die Türen sind auch für die Länder, wo Casinos verboten sind, geöffnet. Einige Staaten versuchen diese Webseiten durch Netzwerkfilter (z.B. „Die Chinesische Mauer“ von China) zu sperren, anscheinend vergeblich.

Betrachtet man das Wettgeschäft über das Internet als Vermarktung einer Dienstleistung, kann es sehr profitabel sein, da viele Kostenfaktoren einer klassischen Wettannahmestelle wegfallen und man durch diesen Internetservice eine breitere Kundschaft ansprechen kann. Andererseits ist die Konkurrenz zwischen Internetbuchmacher deutlich höher als die zwischen Wettannahmestellen. Alle Internetbuchmacher sprechen dieselbe Gruppe von Kunden an und diese Kunden können ohne große Mühe von einem Anbieter zum anderen wechseln.

2.4.1. Produktdifferenzierung beim Pferderennwetten im Internet

Es stellt sich die Frage, wie ein Internetbuchmacher Marktvorteile gegenüber den Konkurrenten bei der Vermarktung einer Dienstleistung, dessen Rahmenbedingungen für alle gleich sind (DIREKTORIUM und HVT, 2004), erzielen kann. Diese Rahmenbedingungen sind:

1. Es wird (innerhalb eines Landes) immer für die gleichen Rennveranstaltungen gewettet.
2. Die Wetteinsätze werden bei demselben Totalisator gesammelt und dadurch werden bei allen Anbietern gleiche Quoten erzielt.
3. Die Wettarten und die Einsatzstufen für die Wetten sind gesetzlich vorgeschrieben.

Unter diesen Bedingungen kann man kaum eine Produktdifferenzierung realisieren, die das Produkt anders als die Produkte der Konkurrenten darstellt. Man kann auch den Preis des Produkts nicht ändern, da er vorgeschrieben ist. Unter diesen Umständen kann der Anbieter eigentlich nur durch die Differenzierung der Servicequalität Marktvorteile erzielen. Es kommt somit nicht mehr darauf an, was man verkauft, sondern wie man es verkauft. Um die Qualität der Dienstleistung zu erhöhen könnte man folgende Maßnahmen einleiten:

1. Erhöhung der Erreichbarkeit der Dienstleistung, durch Anbieten anderer Zugangsmöglichkeiten außer dem Web-Browser zu dem Service. Diese können von mobilen Internetzugängen wie Mobiltelefone oder PDAs bis zu automatisierten Kiosksystemen variieren.
2. Bereicherung des Informationsangebots mit Hilfe einer stabilen und fehlerfreien Informationsbasis.

3. Erhöhung der Benutzerfreundlichkeit mit gutem Design, einem effektiven Hilfesystem und einem übersichtlichen Kontoführungssystem für Kunden.

Mit der Erhöhung der Erreichbarkeit kann man weitere Kunden gewinnen, z.B. die Personen, die ungern an einem PC sitzen oder keinen Computer haben. Es kann auch dazu kommen, dass die Kunden, die den Service normalerweise mit dem Web-Browser verwenden, diese neuen Möglichkeiten in Anspruch nehmen und dadurch den Service häufiger verwenden. Ein reiches Informationsangebot gibt den Kunden ein größeres Selbstvertrauen: Menschen spielen Glücksspiele häufiger, wenn sie denken, dass sie es gut verstehen. Mit verbesserter Benutzerfreundlichkeit können neue Kunden gewonnen werden und man verliert auch weniger Kunden an die Konkurrenz; übersichtliche und stabile Informationssysteme führen zu einer Gewöhnung der Kunden an das System, und erhöhen somit die Kundentreue.

Das im Rahmen dieser Arbeit konzipierte Programm beschäftigt sich hauptsächlich mit der Verbesserung der Erreichbarkeit des Trabrennwettgeschäfts durch Bereitstellung von zusätzlichen Informationen und die Möglichkeit der Wettabgabe über mobile Internetzugänge. Um dieses Ziel zu erreichen stellt das Programm die Renn- und Traberinformationen über Mobiltelefone bereit und bietet des Weiteren die Möglichkeit an, Wettscheine zu den aktuellen Rennen über Mobiltelefone abzugeben.

2.4.2. Mobiltelefone als ein Medium für Pferderennenwetten

Das Internet ist wie geschaffen für die Vermarktung von Dienstleistungen, die von Computern erledigt werden, da es dabei keine Verzögerungen der Transaktionen durch Lieferzeiten entstehen. Auf der Seite des Dienstleisters entsteht kein hoher Anspruch an Arbeitskräfte und dadurch werden die laufenden Kosten des Dienstleisters stark gesenkt. Angewandt auf das Wettgeschäft würde die Möglichkeit einer Wettannahme durch einen Internetservice die Kosten pro Wettschein im Vergleich zu einer klassischen Wettannahmestelle enorm reduzieren. Dadurch kann man das Wettgeschäft profitabler machen und mit Hilfe der verbesserten Erreichbarkeit durch das Internet neue Kunden gewinnen.

Erreichbarkeit im Sinne von Abrufbarkeit ist ein wichtiger Faktor für die Dienstleistungen im Internet. Ohne hohe Erreichbarkeit hat eine Dienstleistung im Internet kaum Wert. Die generelle Vorstellung der Menschen, dass man für eine Internetverbindung unbedingt einen PC benötigt, ist ein großes Handicap für die

Erreichbarkeit. Die Ablehnung der PCs im Privatleben erschwert dem Internet die Verbreitung in Privathaushalte und verringert dadurch die Erreichbarkeit der über das Internet zu vermarktenden Dienstleistungen.

Unter diesen Umständen sollte ein Dienstleister, der seine Erreichbarkeit erhöhen will, nicht nur PC-Benutzer mit vollwertigen Webbrowsern als Zielpublikum ansehen, sondern auch die stark anwachsende Benutzergruppe mit mobilen Internetzugängen berücksichtigen.

Im Wettgeschäft gibt es noch weitere Kundengruppen, die ein solches Angebot für Mobiltelefone begrüßen würden:

- Personen, die keine Zeit haben, um an eine Wettannahmestelle zu kommen.
- Personen, die sich unabhängig von Ort und Zeit über Rennen informieren wollen.
- Personen, die Interesse am Wettspiel haben, aber PCs zu kompliziert finden.
- Personen, die sich auf der Rennbahn über Teilnehmer eines Rennens detailliert informieren wollen.
- Die Pferdebesitzer und -züchter, die Informationen ihrer Pferde überall verfügbar haben wollen.

Die Ist-Situation zeigt, dass es noch unerfüllte Anforderungen der Pferdewettkunden gibt, die man mit effektiven Vermarktungsstrategien in Markterlöse verwandeln könnte.

2.5. Datenverkehr im Trabrennwettgeschäft

Trotz der hohen Anzahl der Wettanbieter für Trabrennen im Internet, ist die Informationsinfrastruktur, die den Wettanbieter mit den aktuellen Informationen unterstützen sollte, noch relativ primitiv. Die aktuellen Renninformationen, die auf der Rennbahn per Hand erhoben werden, gelangen per Fernkopierer (Fax) an die Buchmacher. Bis diese Daten in der offiziellen Datenbank des HVTs eingetragen werden, können ein paar Tage vergehen. Neben dieser Problematik der Zeiteffizienz kann man die Genauigkeit der über diesem Wege übertragenen Daten nicht garantieren, da es keine standardisierten Kontrollprozeduren für die Datenerhebung gibt [HVT].

Die Eigentumsrechte der erhobenen Informationen liegen beim jeweiligen Rennverein und daraus resultierend auch beim HVT. Das macht den HVT zum

offiziellen Erfasser und Herausgeber der Daten und nur diese Daten, die vom HVT akzeptiert wurden, gelten als offiziell. Der HVT verwaltet Renn-, Herkunfts- und Formdaten der Traber in einem geschlossenen Datenbanksystem, das auf der *IBM Lotus Domino Server*-Software basiert. Da diese Datenbank von anderen Systemen nicht abfragbar ist, kann es von den Buchmachern für die Zusammenstellung der Form- oder Herkunftsdaten nicht genutzt werden. Aufgrund der zeitverzögerten Aktualisierung der Daten ist sie als Informationsquelle für die aktuellen Renninformationen ungeeignet. Die Buchmacher, die diese Informationen aus der HVT-Datenbank beziehen wollen, kopieren die entsprechenden Webseiteninhalte, die auf der Basis der HVT-Datenbank generiert werden. Obwohl der HVT bisher keine Aktionen dagegen unternommen hat, ist es rechtlich klar, dass diese Firmen, durch Kopieren der Informationen die Urheberrechte des HVTs verletzen, es sei denn es gibt Abmachungen zwischen HVT und Buchmacher.

Dieser Sachverhalt lässt sich auch als Datensmog bezeichnen. Auf einer Seite sind die offiziellen Informationen, die schwer zu erreichen, zu verwenden und nicht unbedingt aktuell sind. Auf der anderen Seite sind die kleinen Datenbanken der Buchmacher, deren Inhalte nicht komplett und möglicherweise fehlerhaft sind. Die Kunden, die über das Internet wetten, lesen mit hoher Wahrscheinlichkeit diese auf den fehlerhaften Daten basierenden Informationen, da sie auf der gleichen Webseite mit den Wettabgabemöglichkeiten dargestellt werden. Nur ein geringer Teil der Kunden würde sich noch bemühen, die offiziellen Informationen von der HVT-Homepage parallel zur Web-Seite des Wettanbieters aufzurufen. Im Endeffekt werden die gleichen Daten mehrfach von verschiedenen Wettanbietern erhoben, bearbeitet und gespeichert. Das führt zu Integritätsproblemen der dargestellten Informationen und könnte die Kunden verwirren und möglicherweise Verluste bei den Wettumsätzen zuzufolge haben. Hieraus wird deutlich, dass der Datensmog dem gesamten Wettgeschäft schadet.

Ein Lösungsansatz dieser Problematik könnte folgendermaßen aussehen: Erstens sollte man den Datenverkehr produktiv organisieren. Die vor Ort erhobenen Daten sollten zuerst an den HVT weitergeleitet und dann von der HVT-Datenbank an die Buchmacher verteilt werden. Durch die Umstellung der Methoden zur Datenerhebung auf der Rennbahn auf elektronische Basis könnten die gesammelten Daten zeitnah über das Internet an den HVT gesendet werden. Die HVT-Datenbank sollte für eine ständige Aktualisierung und für Abfragen von Drittpersonen umgestellt bzw. umstrukturiert werden. Die Buchmacher sollten ihre

Serversysteme und Web-Applikationen zur Abfrage der HVT-Datenbank umstellen. Die durch diese Systemumstellung gesparte Arbeitszeit könnte die Kosten der Umstellung bzw. des Betriebs des Systems im Laufe der Zeit decken. Das Wettgeschäft wird davon profitieren, da man mit einheitlichen Informationen aus dem Trabrennsport mehr Kundenvertrauen gewinnen könnte.

3. Material und Methoden

3.1. Datenstrukturen und Daten

Die in dieser Arbeit verwendeten Daten stammen von der Traberdatenbank des HVTs. Die Datenstruktur der Projektdatenbank wurde wie die Datenstruktur der HVT-Datenbank gestaltet, damit eine automatisierte Übertragung der Daten von der HVT-Datenbank zur Projektdatenbank in der Zukunft leichter ermöglicht werden kann. Die Datenbank beinhaltet sechs Tabellen, die von der HVT-Datenbank stammen. Zusätzlich dazu wurden zwei Tabellen in derselben Datenbank erstellt, die die Kundeninformationen und Wettscheine beinhalten. Bei der Erstellung der Tabellen in MySQL-Server wurden die Längen- und Datentypangaben der HVT-Datenbank nicht übernommen. Somit wurde ein flexibleres Arbeitsumfeld für die Programmierung ermöglicht und es entstanden weniger Kompatibilitätsprobleme beim Datenimport. Die Kontrollfunktionen über den Dateninhalt werden nicht in der Datenbankstufe, sondern auf der Programmierungsebene erstellt, da hier bessere Kontrollebenen für wahrscheinliche Fehler aufgebaut werden können. Die Datenbankstruktur ist relational organisiert und jede Tabelle beinhaltet Dateneinträge, die mit anderen Dateneinträgen in anderen Tabellen durch einzigartige Schlüsselfelder in Verbindung stehen. Diese Tabellen sind:

- Renntage (Tabelle tgp)
- Rennen (Tabelle asp)
- Starterangaben (Tabelle stp)
- Leistungsdaten (Tabelle tlp)
- Traberstamm (Tabelle trp)
- Schlüsselwerte (Tabelle slp)
- Kundendaten (Tabelle kunde)
- Wettscheine (Tabelle wette)

Die Datenstrukturen dieser Tabellen werden in den folgenden Tabellen dargestellt:

Tabelle 1: TGP - Renntage

Name	Art	Info
tgrvid	Schlüssel	Kürzel der Rennbahn Schlüssel: 410
tgdat		Datum des Renntags
tgrvbez		Kurzbezeichnung des Rennvereins
tgbez1		Renntagsbezeichnung, freier Text
tgbez2		Renntagsbezeichnung, freier Text
tgrntr		Renntag Nummer
tgbahnzu	Schlüssel	Bahnzustand Schlüssel: 100
tgstartzt		Uhrzeit Start des Renntags HH:MM
tgwetter		Text zum Wetter
tgrenuma		Außenumsatz
tgrenumb		Bahnumsatz
tgrenumg		Gesamtumsatz
tgtemp		Temperatur
tgtempbez		Bezeichnung für Temperatur; Wertebereich: C° F°
tglufeu		Luftfeuchtigkeit
tgstatusa		Status des Renntags; Wertebereich: abgesagt genehmigt Renntag abgeschlossen
tgwae		Währungskürzel; Wertebereich: EUR

Tabelle 2: ASP – Rennen

Name	Art	Info
asdat		Datum des Renntags
asrvid	Schlüssel	Kürzel der Rennbahn Schlüssel: 470
asausnr		Ausschreibungs-Nr.
asrenabt		Abteilungs-Nr. des Rennens
asrennr		Nummer des Rennens am Renntag
asartren	Schlüssel	Art des Rennens Schlüssel: 230
asbahnkat		Bahnkategorie; Wertebereich: A B C
asgeldpre		Dotierung
aspreiauf		Rennpreisaufteilung
asdist		Distanz
asstartzt		Startzeit des Rennens (HH:MM)
asstrtar	Schlüssel	Startart Schlüssel: 470
asasbt		Ausschreibungskriterien
asrennam1		Name des Rennens 1. Zeile
asrennam2		Name des Rennens 2. Zeile
asrenabt		Rennabteilung
asstsuvon		Startsumme von
asstsubis		Startsumme bis
asnensl		Nennungsschluß von
asnenslb		Nennungsschluß bis

asstasl		Starterangabe von
asstaslb		Starterangabe bis
asaltsvon		Alter der Stuten von
asaltsbis		Alter der Stuten bis
asalthvon		Alter der Hengste von
asalthbis		Alter der Hengste bis
asrekvon		Rekordzeit von
asrekbis		Rekordzeit bis
astrags		Trabergeschlecht; Wertebereich: H H/W S S/H/W S/H
asherk		Herkunft
asricht		Rennrichtung
asrenart		Text für Art des Rennens
asfahrei	F/R	Kennzeichen Fahren oder Reiten
asstatus		Status

Tabelle 3: STP - Starter

Name	Art	Info
stdat		Datum des Renntags
strvid	Schlüssel	Kürzel der Rennbahn Schlüssel: Rennbahnen
stausnr		Ausschreibungs-Nr.
stabt		Abteilungs-Nr. des Rennens
stident		Traber-Identnummer
stbesnam		Name des Besitzers *
stfahnam		Name des Fahrers *
ststkennz		Wertebereich: neu genannt Starter nicht startberechtigt gemindert Nichtstarter gestrichen freiwillig zurückgetreten gesperrt

Tabelle 4: TLP - Leistungsdaten

Name	Art	Info
tlident		Traber-Identifikation
tldat		Datum des Rennens
tlindex		Index des Rennens
tlort		Kürzel der Rennbahn Schlüssel: Rennbahnen
tlaustra		Kennzeichen Inland/Ausland A=Ausland I=Inland
tlname		Trabername
tlartren		Art des Rennens Schlüssel: Kennzeichen Art des Rennens
tlplatz		Platzierung
tldisqtx		Disqualifikation Schlüssel: Art der Disqualifikation Ermittlung über Platzierung (>20<50)
tlgzeit		Gesamtzeit
tldzeit		KM-Zeit
tldist		Distanz
tlstrart		Startart Schlüssel: Startart
tlbahnzu		Bahnzustand Schlüssel: Bahnzustand
tlstartnr		Start-Nr.

tlgewsum		Gewinnsumme
tlzueprae		Züchterprämie-Betrag
tlzuchtfo		Zuchtfond-Betrag
tlwae		Währung der Gewinnsummen Wertebereich: EUR
tlfahrnam		Name des Fahrers *
tlbesname		Name des Besitzers *
tlzuenam		Name des Züchters *
tltrainam		Name des Trainers *
tlequote		Eventualquote
tleinlauf		Namen der 3 erstplatzierten Pferde
tleerstqua		Kennzeichen Qualifikation Q = Erstqualifikation W = Wiederqualifikation leer = kein Qualif.-Rennen
tlalter		Alter des Trabers
tlges		Geschlecht des Trabers Wertebereich: H/W/S
tlfarbe		Farbe des Trabers

Tabelle 5: TRP - Traberstamm

Name	Art	Info
trident		Traber Identifikation
trname		Name des Trabers
trtrastat		Status eines Trabers 00 Unbekannter Status 01 Inländer A 02 Inländer B 03 Ausgeführt A 04 Ausgeführt B 05 Eingeführt A 06 Eingeführt B 07 Tod 08 Ausgeschieden 09 keine Einfuhrgenehmigung 10 Fohlenmeldung 11 Fohlenaufnahme 12 Einfuhr abgelaufen 13 Unbekannter Staus
trzuena		Züchter-Kurzname *
trbesna		Besitzer-Kurzname *
trnameur		ursprünglicher Name
trnation		Nationalität (aktuelle Länderzugehörigkeit)
trgebdat		Geburtsdatum
trtoddat		Todesdatum
trmutter		Traberid Mutter
trvater		Traberid Vater
trgeschl		Geschlecht
trfarbea		Farbe aktuell Schlüssel: Farbenverzeichnis
trqualzt		Zeit der 1. Qualifikation
trgewsum		Gewinnsumme

trgewsumw		Gewinnsumme Währung
trrekzeit		Km-Zeit bei Rekord
trrekart		Rekordart Q=Qualifikationsrekord
trrekges		Geschlecht bei Rekord
trrekdat		Rekordzeit Datum
trrekland		Land
trrekort		Rekordzeit Ort Schlüssel: Rennbahnen
trrekdist		Rekordzeit Distanz
trrekstrt		Rekordzeit Startart Schlüssel: Startart
trrekazei		Rekordzeit Autostart
trrekaart		Rekordart Autostart Q=Qualifikationsrekord
trrekages		Geschlecht bei Rekord Autostart
trrekadat		Rekordzeit Datum Autostart
trrekanlan		Land Rekordzeit Autostart
trrekaort		Rekordzeit Ort Autostart Schlüssel: Rennbahnen
trrekadis		Rekordzeit Distanz Autostart

Tabelle 6: SLP - Schlüssel

Name	Art	Info
sschg		Schlüsselgruppe
sdaten		Schlüssel Kurztext
Sschl1		Schlüssel Langtext

Tabelle 7: KUNDE - Kundendaten

Name	Art	Info
Id	Schlüssel	Benutzer ID
Anrede		Person Anrede
Vorname		Person Vorname
Nachname		Person Nachname
Benutzername		Person Benutzername
Passwort	MD5-Signatur	MD5-Signatur des Passworts mit 2 Byte seed
Guthaben		Guthaben
Aktiv		Aktivitätsschlüssel Wertebereich: 0: Inaktiver Benutzer 1: Aktiver Benutzer

Tabelle 8: WETTE – Wettscheine

Name	Art	Info
Id	Schlüssel	Wettschein ID
UserId		Benutzer ID
Rennr		Rennnummer
Datum		Datum des Renntags
ortcode		Ort des Rennens
ausschreibung		Ausschreibungsnummer

abteilung		Abteilungsnummer des Rennens
wettart		Art der Wette Wertebereich: S: Sieg SP: Sieg/Platz P: Platz 2: Zweier 3: Dreier
Cost		Kosten des Wettscheins
wettline		Wettstring
Status		Status des Wettscheins

Während der Erstellung dieser Dokumentation hatte die Datenbank folgende Datenmengen:

Tabelle 9: Datenmengen in der Datenbank

Name der Tabelle	Anzahl Dateneinträge	Mittelwert Datenlänge	Gesamte Datenmenge
Asp	3747	248	929.888
Kunde	2	68	136
Slp	523	32	17.112
Stp	29432	85	2.502.900
Tgp	289	147	42.712
Tlp	18207	236	4.313.456
Trp	12712	234	2.975.584
Wette	12	88	1.056

Die Gesamtgröße der Datenbank beträgt 10 MB.

3.2. Technische Ausstattung

3.2.1. Server-Hardware

Das Programm wurde auf mehreren Rechnern betrieben und funktionierte problemlos. Als Plattform für den Betrieb des Programms waren Linux oder andere Unix Varianten vorgesehen, somit kann man erwarten, dass jeder Rechner, der für Server-Betrieb eines Linux Distribution oder Unix Variante einsetzbar ist, fähig sein würde, das Programm durchzuführen. Die Datenbank des Projektes wurde auf einem SUSE 9.1 Rechner mit einem 700 Mhz AMD Duron CPU und 512 MB Arbeitsspeicher gehalten. Das Programm selbst wurde von einem sehr langsamen Rechner (133 Mhz Pentium mit 32 MB Arbeitsspeicher) bis zu einem Linux Cluster (GWDG Webserver Cluster, unbekannte Anzahl von Rechnern) installiert und getestet. Dabei wurden keine Funktionsunterschiede festgestellt, der Hauptunterschied lag in der Reaktionszeit.

Leider liegt uns keine genaue Zeitmessung für die getesteten Plattformen vor, weil eine genaue Messung der Seitendarstellung über das Internet wegen der Netzwerktopologie fast unmöglich ist.

Bei den Messungen der CPU Auslastung (durch Führung eines Shellscripts) von Apache Server (wobei PHP als ein CGI Programm läuft) auf verschiedenen Linux Rechnern konnte kein bedeutender Auslastungsunterschied durch Ausführung des Programms festgestellt werden. Diese Messung konnte wegen der Benutzereinschränkungen nicht bei dem GWDG-Cluster geführt werden. Das bedeutet, dass bei der Durchführung dieser Diagnose keine bemerkenswerte Auslastung durch mehrfaches Aufrufen des Programms realisiert werden konnte. Bei einem realen Serverbetrieb kann man davon ausgehen, dass mehrere Benutzer sich gleichzeitig anmelden und gleichzeitige Datenbankabfragen tätigen werden. Für einen produktiven Linux-Serverbetrieb kann man die Hardware folgendermaßen optimieren:

Arbeitsspeicher

Arbeitsspeicher ist die wichtigste Komponente eines Linux Servers, mit dem man das beste Preis/Leistungsverhältnis erzielt. Als Praxiswert sollte man pro ein MFLOP CPU-Leistung ein MB oder einen höheren Arbeitsspeicher anrechnen.

Hauptprozessor (CPU)

Die mittlere Prozessorgeschwindigkeit am Markt erhöht sich jedes Jahr um das Zweifache, aber mit abnehmender Tendenz. Die technische Leistungsgrenze bei den Silikon-Aluminium basierenden Chips ist schon erreicht worden und in ein paar Jahren wird man auch die Grenze für Silikon-Kupfer basierte Chips erreichen. Während Prozessorproduzenten bei Desktop-Systemen mehr Geschwindigkeit durch Verbesserung der Sekundärkomponenten wie L1 Cache erzielen, geht im Serverbereich die Optimierung der Prozessoren bei der Command-set und Parallelberechnungen. Frühere Mehrprozessorenserver-Architektur wird mit mehrkernigen Prozessoren ersetzt. Damit wird das größte Problem der Mehrprozessorensysteme, die Geschwindigkeitsbegrenzungen der Kommunikation zwischen Prozessoren, gelöst. Ein weiterer Vorteil der Verwendung von mehrkernigen Prozessoren ist der geringe Platzbedarf und dementsprechend ein geringerer Kühlungs- und Energiebedarf.

Bei einem Linux System nimmt der Prozessor die zweite Stelle bei einem Optimierungsvorgang an. Man sollte einen Prozessor verwenden, der im Geschwindigkeits-Preis Optimum liegt.

Festplatten

Die Auswahl an Festplatten und Festplattencontroller ist sehr wichtig bei ständiger Festplattenaktivität; da sie ein paar Millisekunden Unterschied bei den Datenbankabfragen ausmachen kann. Niedrigere random-seek-timing und ausreichend Zwischenspeicherung kann für die Leistung eines Datenbankservers entscheidend sein. Ein anderer Aspekt bei der Festplattenauswahl ist die Datensicherheit. Bei RAID-Konfigurationen kann man gleichzeitig mehrere gespiegelte Kopien einer Festplatte betreiben, wobei eine der Spiegelplatten im Fall einer Störung automatisch die Rolle der Hauptfestplatte übernimmt. Diese ersetzen nicht den Bedarf nach einem sicheren und automatisierten Backupsystem, bei dem die Daten auf einem weniger umweltempfindlichen Medium für Langzeitspeicherung sicher untergebracht werden: Die Rolle eines RAID-Systems ist die Sicherstellung einer momentanen Datensicherheit während des Betriebs. Früher konnte man RAID-Systeme nur mit teuren SCSI-Controllerkarten und dementsprechend mit SCSI-Festplatten konfigurieren. Heutzutage kann mit IDE- oder SATA-Festplatten und relativ günstigen ATAPI-Controllerkarten ein RAID-System aufgebaut werden. Dabei sollte man nicht vergessen, dass SCSI-Systeme dennoch mit höheren

technologischen Eigenschaften wie Job-Caching, Fehlerüberprüfung und Load-Balancing immer noch die bessere Wahl für Serversysteme sind.

Netzwerk

Bis vor ein paar Jahren gab es noch keine große Auswahl an Netzwerkkarten. Mit den technischen Neuerungen muss nicht mehr irgendeine 100 Mbit Netzwerkkarte gewählt werden, es gibt weitaus mehr Auswahlmöglichkeiten. Die Kommunikationsgeschwindigkeit eines Rechners ist hauptsächlich durch die langsamste Hardwarekomponente in der Konfiguration begrenzt, welches meistens die Festplatte ist. 100 Mbit Netzwerkkarten (eine gute Verkabelung sollte nicht unterschätzt werden) sind völlig ausreichend wenn es um Transfer von Daten, die sich auf der Festplatte befinden, geht. Die Netzwerkkarte kann einen Flaschenhals darstellen, z.B. bei Serverkonfigurationen, bei denen nur Daten, die sich auf dem Arbeitsspeicher befinden, weitergeleitet werden, oder wenn zwei oder mehrere Serversysteme in einer Art Cluster betrieben werden. Um diese Begrenzung zu umgehen, kann man Gigabit-Ethernet Karten verwenden, die bei neuerer Serverhardware besonders für Zwecke des Clustering vorinstalliert sind.

3.2.2. Dauerbetrieb

Dauerbetrieb ist das grundlegende Ziel eines Serverbetriebes. Serversysteme müssen 24 Stunden am Tag fehlerfrei laufen und sollen mit dementsprechender Hardware- und Netzwerkinfrastruktur unterstützt werden. Man kann mögliche Ausfallursachen voraussehen und vorbeugende Maßnahmen einleiten, die an erster Stelle die Wahrscheinlichkeit eines Ausfalls verringern und im Fall eines Ausfalles möglichst schnell darauf reagieren.

Die Maßnahmen, mit denen man die Wahrscheinlichkeit eines hardwarebedingten Ausfalles verringern könnte:

- Verwendung von stabiler und getesteter Hardware
- Unterstützung des Servers mit Voltage-Regulatoren und USV
- Regulierung der Umweltfaktoren (Temperatur, Luftfeuchtigkeit) in den Räumlichkeiten des Servers
- Regelmäßige Wartung der Serverhardware
- Regelmäßige Erneuerung der Hardware mit beweglichen mechanischen Teilen, wie Festplatten, Lüfter.

Die Maßnahmen, mit dem man im Fall eines Ausfalles schnell reagieren kann:

- Regelmäßiges Backup
- Verwendung von RAID Festplattenkonfigurationen
- Verwendung eines High-Availability Server Konfiguration mit Master-Slave Clustering Methode oder Verwendung eines Beowulf Cluster Konfiguration (wie bei Google) wobei Ausfall der einzelnen Systemen nicht von Bedeutung sind.

Obwohl der Ausfall eines kompletten Serverbetriebs mit Clustern, Backup-Servern und automatisierter Überwachung sehr unwahrscheinlich ist, kann es im Fall einer Naturkatastrophe von Vorteil sein, eine Offsite Replikation der Datenbanken und Konfigurationsdaten zu haben.

3.2.3. Mobile Internetzugänge

Das in diesem Projekt konzipierte Programm richtet sich an Mobiltelefone der dritten Generation mit WAP2-Unterstützung und an PDAs mit Web-Browser Funktionalität. Am Anfang des Projekts wurde das Programm an die damals aktuelle i-mode-Technologie gerichtet. Im Laufe der Zeit ging der Mobiltelefonmarkt, besonders im Smartphone-Bereich, in die Richtung einer XML-kompatiblen neueren Version von WAP, genannt WAP2. WAP2 unterstützt XHTML, ein Subset von HTML für mobile Internet Zugänge anstatt die von WAP unterstützen WML. XHTML hat weitaus bessere Fähigkeiten als die WML. Somit wurde die i-mode-Unterstützung überflüssig. WAP2 bietet bessere Möglichkeiten der Datendarstellung und unterstützt viele Eigenschaften wie Darstellung der Tabellen, JPEG-Bildformat, Formulare und Cookies. Eine weitere Fähigkeit der WAP2 ist die Unterstützung von HTTP-Header-Direktiven, mit denen man grundsätzliche Fähigkeiten eines Web-Browsers wie zum Beispiel automatisches Seitenaktualisierungen, Feststellung des Ablaufdatums für einzelne Seiten, automatisierte Weiterleitung auf andere Adressen und einer Authentifizierung durch HTTP-Header. Während der Entwicklung des Programms wurde das Programm mit den Mobiltelefonen Siemens CX65, S65 und Nokia 6600 getestet.

Das Programm kann aufgrund des Konzepts an technische Weiterentwicklungen angepasst werden. Ein Vorteil dieser Erweiterbarkeit zeigt sich bei der Anpassung der Ausgabeschnittstellen. Die frühere i-mode-Ausgabeschnittstelle des Programms wurde im Laufe des Projektes für die WAP2-Ausgabe neu programmiert.

3.3. Software

Die im Rahmen des Projektes verwendeten Softwareprodukte können in zwei Gruppen geteilt werden. Zum einen die Software, die bei der Entwicklung des Programms verwendet wurden, und zum anderen die Software, die für den Testbetrieb verwendet wurden.

Bei der Entwicklungsphase des Programms wurden folgende Softwareprodukte zur Erstellung des Programms verwendet:

- **XAMPP:** XAMPP ist ein zusammengestelltes Paket aus Apache HTTP Server, MySQL RDBMS und PHP, welches den Betrieb eines kompletten Webservers mit vollfunktionsfähiger Datenbank und einem Applikation Backend auf einem Windows Rechner ermöglicht. XAMPP ist laut des Herstellers [AFO] eine (netzwerktechnisch gesehen) unsichere Testkonfiguration und sollte nicht als ein komplettes Serverpaket gesehen oder betrieben werden. XAMPP ist unter GPL zu beziehen.
- **PHP Expert Editor:** Für die Verfassung des Quellcodes des Programms ist ein einfacher Texteditor wie z.B. Pico relativ ungeeignet. Um den Überblick zu behalten und Fehlerkontrollen durchführen zu können ist ein PHP-spezifischer Programmeditor erforderlich. Zu diesem Zwecke wurde das Programm „PHP Expert Editor“ von der Firma Ankord [ANK] eingesetzt.
- **UltraEdit32:** UltraEdit32 ist ein sehr fortschrittlicher Texteditor mit eingebauter Skriptsprache für Makrofunktionalität und wurde für Fehlerkontrollen der CSV-Dateien vor dem Import in die MySQL-Datenbanken eingesetzt.
- **Navicat MySQL Studio:** Diese Software bietet ein graphisches Interface für MySQL RDBMS und wurde für den Import der CSV Dateien, bzw. zur Replikation der Datenbanken eingesetzt [NAV].

Außerdem wurden übliche Programme wie Total Commander zum Datentransfer und SSH Client für SSH Verbindungen mit Linux-Servern verwendet.

Während des Testbetriebs des Programms wurden mehrere Linux-Systeme mit verschiedenen Hard- und Softwarekonfigurationen verwendet. Im Folgenden wird ein genereller Überblick auf diese Softwarekonfigurationen gegeben, wobei eine Unterteilung in drei Kategorien (Betriebssystem, Webserver und Datenbanken) vorgenommen wird.

3.3.1. Betriebssystem

Das im Rahmen dieses Projektes entwickelte Programm ist plattformunabhängig, d.h. es läuft unter allen aktuellen netzwerktauglichen Betriebssystemen, die PHP-Parser unterstützen. Während der Entwicklung wurde das Programm hauptsächlich unter Linux betrieben. Für den späteren produktiven Serverbetrieb wird empfohlen, dass das Programm auch unter Linux oder eine Unix-Variante betrieben werden sollte, obwohl es auch unter Windows Betriebssysteme problemlos laufen würde. Linux ist ein freies Betriebssystem. Mehrere TCO-Analysen haben gezeigt, dass Linux im Serverbetrieb in den meisten Fällen weniger kostet als vergleichbare Windows-Systeme (ROBERT FRANCES GROUP, 2005).

Auch unter Sicherheitsaspekten ist Linux die bessere Wahl. Bei einer Neuinstallation ist Linux von Anfang an netzwerktechnisch gesehen sicher (BAUER, 2002). Im Vergleich dauert die Verseuchung eines neu installierten Windows Rechners im Netzwerk ca. acht Minuten.

3.3.2. Webserver

Das Programm wurde unter Apache HTTP Server [APC] (Versionen 1.3 und 2.0) erfolgreich getestet. Das Programm hat keine webserverspezifischen Komponenten, deswegen ist es theoretisch möglich, das Programm auch unter anderer Webserver-Software zu betreiben.

Apache HTTP Server ist einer der am meisten benutzten Webserver-Software im Internet. Netcraft.com [NCC], ein unabhängiges Unternehmen aus England, das hauptsächlich an dem Thema Netzwerksicherheit arbeitet, führt jeden Monat eine automatisierte Abfrage durch, um festzustellen, wie viele aktive Webseiten es gibt und welche Server Programme auf diesen Servern laufen. Nach Angaben der Netcraft.com gab es im August 2005 22.195.711 aktive Apache HTTP Server, die auf die Abfrage geantwortet haben. Das sind 69,97% aller Webserver. Es gibt viele

Gründe, warum der Apache HTTP Server mit großem Vorsprung den Markt anführt. Einige davon sind (LAURIE und LAURIE, 2002):

- Apache HTTP Server ist Open Source Software und kostenfrei
- Fast alle Linux Distributionen beinhalten Apache HTTP Server
- Sehr leicht zu installieren
- Gut dokumentiert
- Viele Konfigurationsmöglichkeiten
- Sehr stabile, fehlerfreie Funktion nach dem Motto „Installieren & Vergessen“
- Benutzt die Systemressourcen sparsam; Hardware Anforderungen sind sehr niedrig
- Unterstützung von CGI und native Schnittstellen für Perl, PHP, Python, ASP, JSP, XML und viele andere Programmiersprachen

3.3.3. Datenbanken

Die in diesem Projekt verwendeten Datenbanken wurden in der MySQL RDBMS erstellt. Das Programm verwendet spezifische SQL-Verbindungsfunktionen und Abfragen für MySQL-Verbindung, damit ist die Funktionalität des Programms abhängig von einem funktionsfähigen MySQL-Server. Andererseits kann man das Programm mit kleinen Veränderungen umkonfigurieren, damit es andere Datenbankserver ansprechen kann.

MySQL ist eine der populärsten Open Source SQL-Datenbanken, die von der Firma MySQL AB [MAB] zur Verfügung gestellt wird. MySQL AB ist ein kommerzielles Unternehmen, dessen Geschäft darin besteht, Serviceleistungen rund um die MySQL-Datenbank zur Verfügung zu stellen. MySQL ist kostenfrei, solange man den Bedingungen der GPL [GPL] zustimmt. In den Fällen, in denen man den GPL Bedingungen nicht zustimmen kann (z.B. Integrierung der MySQL in eine kommerzielle Software Lösung), kann die entsprechende Lizenz von der Firma MySQL AB erworben werden.

Leichte Installation und Konfiguration, vertrauenswürdige Sicherheit, hohe Geschwindigkeit und eingebaute Netzwerkzugriffsmöglichkeiten machen den MySQL zu einer beliebten Lösung zur Erstellung der datenbankunterstützten Webseiten und für Zurverfügungstellung der Datenbanken über das Internet.

MySQL ist ein Client-Server-System, das aus einem multi-thread SQL-Server besteht, der unterschiedliche Back-Ende, verschiedene Client-Programme, Verwaltungswerkzeuge und viele Programmschnittstellen unterstützt.

Die folgende Aufzählung beschreibt einige wichtige Eigenschaften von MySQL:

- Unterstützung der Mehr-Prozessor-Rechner
- Schnittstellen für C, C++, Eiffel, Java, Perl, PHP, Python und Tcl
- Läuft auf vielen verschiedenen Plattformen und Betriebssystemen:
 - AIX 4.x
 - Amiga
 - BSDI 2.x, 3.x und 4.x
 - DEC Unix 4.x
 - FreeBSD 2.x, 3.x und 4.x
 - HP-UX 10.20, 11.x
 - Linux (Alle Distributionen)
 - Mac OS X Server
 - NetBSD 1.3/1.4 Intel und NetBSD 1.3 Alpha
 - OpenBSD
 - OS/2 Warp 3, 4
 - SGI Irix
 - Solaris 2.5 und höher
 - SunOS 4.x
 - Caldera (SCO) OpenServer und UnixWare 7.x
 - Tru64 Unix
 - Windows 9x, ME, NT, 2000, XP, 2003
- Unterstützung vielen Spaltentypen
- Volle Unterstützung der SQL Standards
- Zugriffsmöglichkeit auf die Tabellen aus unterschiedlichen Datenbanken in ein und derselben SQL-Anfrage
- ODBC Unterstützung für Windows. Man kann z.B. MS Access benutzen, um sich mit einem MySQL-Server zu verbinden.
- Datensätze fester und variabler Länge
- Volle Unterstützung für mehrere unterschiedliche Zeichensätze
- Vom gewählten Zeichensatz abhängige Sortierung
- Fehlermeldungen in verschiedenen Sprachen

- Verbindungsmöglichkeiten über TCP/IP Sockets, Unix Sockets (Unix) oder Named Pipes (NT)
- Kann große Datenbanken verwalten. Nach Angaben der MySQL AB wurden Datenbanken getestet, die 50 Millionen Datensätze haben und es gibt Benutzer, die MySQL mit 60.000 Tabellen und etwa 5 Milliarden Zeilen benutzen.
- Ein System von Zugriffsberechtigungen und Passwörtern, das sehr flexibel und sicher ist, und das Host-Basierende Verifizierung erlaubt. Passwörter sind sicher, weil jeder Passwort-Verkehr verschlüsselt wird, wenn man sich mit einem Server verbindet.

MySQL ist im Internet weit verbreitet und wird von vielen großen Unternehmen wie z.B. Alcatel, Apple, BBC, Cisco, Compaq, Ericsson und auch von staatlichen Behörden wie der NASA verwendet.

3.4. Programmiersprachen

3.4.1. PHP: Hypertext Preprocessor (PHP)

Obwohl das Programm in der gesamten Entwicklung für ein LAMP-System (**L**inux **A**pache **M**ySQL **P**HP) ausgedacht war, ist PHP die einzige Komponente, die das Programm unbedingt benötigt um zu funktionieren. PHP ist eine weit verbreitete und für den allgemeinen Gebrauch bestimmte Open Source Skriptsprache, welche speziell für die Web-Programmierung geeignet ist und in HTML eingebettet werden kann. PHP besteht aus zwei Teilen. Der erste Teil ist die Skriptsprache, mit der ein Programmierer seine Skripten schreibt. Der zweite Teil ist der Parser, ein Programm, das die vom Programmierer geschriebenen Skripte ausführt. In der PHP-Architektur spricht man nicht von einem Compiler, weil die geschriebenen Programme immer in Form einer ASCII-Datei bleiben und der Parser niemals ein selbst lauffähiges binäres Programm erzeugt. Diese Teile sind in den meisten Fällen kostenfrei zu benutzen, sogar für die Entwicklung von kommerziellen Programmen. In Ausnahmefällen sollte die PHP-Lizenz in Anspruch genommen werden.

Es gibt drei Hauptanwendungsgebiete, in denen PHP-Skripte eingesetzt werden.

1. Serverseitige Skripte. Dies ist traditionell auch das Hauptfeld von PHP, wie in dieser Arbeit. Man benötigt dazu drei Komponenten: Den PHP-Parser (CGI oder

Server-Modul), einen Webserver und einen Webbrowser. Man muss den Webserver, verbunden mit einer PHP-Installation laufen lassen. Die Ausgabe der PHP Programme kann man sich über den Server mit einem Webbrowser ansehen.

2. Skripte auf der Kommandozeile.
3. Schreiben clientseitiger GUI Applikationen.

PHP kann auf allen gängigen Betriebssystemen verwendet werden, inkl. Linux, vielen Unix-Varianten (inkl. HP-UX, Solaris und OpenBSD), Microsoft Windows, Mac OS X, RISC OS, und vielen anderen. PHP unterstützt auch die meisten der heute gebräuchlichen Webserver, wie Apache, Microsoft Internet Information Server, Personal Webserver, Netscape und iPlanet Server, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd.

In dieser Arbeit wurde PHP hauptsächlich dazu benutzt, um HTML Ausgaben zu generieren. Andererseits ist die Ausgabemöglichkeit des PHP nicht nur auf Textdateien beschränkt. Mit PHP können auch Bilder, Graphiken, PDF-Dateien und Flash Animationen dynamisch generiert werden.

PHP unterstützt die Kommunikation mit anderen Services, welche Protokolle wie LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (auf Windows) unterstützen.

Grundlagen der PHP

Ein PHP-Skript beginnt immer mit „<?php“ oder kurzgeschrieben mit „<?“ und endet immer mit „?>“. Alle Anweisungen enden mit einem Semikolon-Zeichen. Wenn am Anfang einer Zeile „//“ steht, dann wird diese Zeile vom PHP-Parser einfach übersprungen.

Ein ganz einfaches PHP-Skript sieht wie folgt aus:

```
<?php
// Dies ist ein PHP-Skript, aber PHP-Parser sieht diese Zeile nicht!
echo ("Ich habe ein Pferd.");
?>
```

Das Skript oben zeigt den Text „Ich habe ein Pferd.“ als Ausgabe:

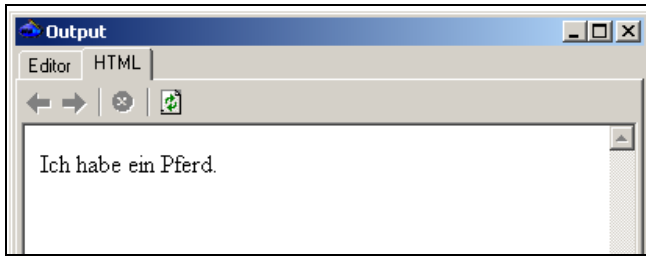


Abbildung 3: Ausgabe eines PHP-Skriptes; erster Schritt

Man kann beliebig viele Variablen in einem PHP-Skript benutzen und die Werte dieser Variablen nach Bedarf ändern:

```
<?php
// Folgende Zeile beschreibt eine Variable:
$Anzahl = 1;
//Folgende Zeile addiert die $Anzahl Variable mit 2:
$Anzahl = $Anzahl + 2;
//Folgende Zeile gibt das Ergebnis „Ich habe 3 Pferde.“:
echo ("Ich habe " . $Anzahl . " Pferde.");
?>
```

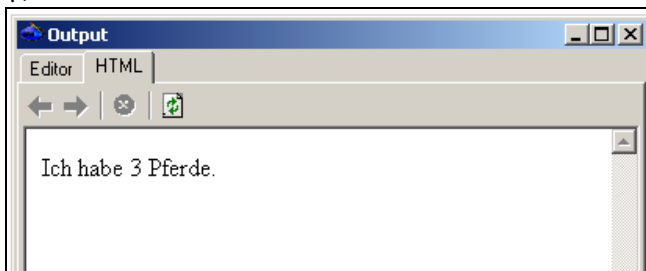


Abbildung 4: Ausgabe eines PHP-Skriptes; zweiter Schritt

Es gibt viele Methoden um Schleifen zu bilden. Eine der beliebtesten ist die WHILE-Funktion:

```
<?php
$Anzahl = 1;
// Folgende Zeile öffnet eine Schleife:
while ($Anzahl <= 5) {
// Folgende Zeile macht eine Ausgabe:
echo ("Ich habe " . $Anzahl . " Pferde. ");
//Folgende Zeile addiert die $Anzahl Variable mit 1:
$Anzahl = $Anzahl + 1;
//Folgende Zeile schließt die Schleife:
};
?>
```

Das Ergebnis des Skriptes soll wie folgt aussehen:

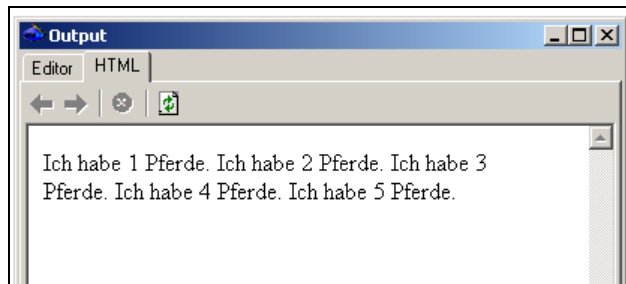


Abbildung 5: Ausgabe eines PHP-Skriptes; dritter Schritt

Es gibt viele Möglichkeiten Bedingungsausdrücke zu benutzen. Eine einfache und häufig benutzte Funktionsgruppe dafür ist die IF-ELSE Gruppe:

```
<?php
$Anzahl = 1;
while ($Anzahl <= 5){
//Folgende Zeile beschreibt eine Bedingung um folgende Anweisung zu
führen:
if ($Anzahl == 1) { $Name = " Pferd. ";}
// ELSE-Funktion beschreibt die Anweisung zu laufen, wenn die mit
der IF-Funktion
// beschriebene Bedingung nicht erfüllt wird:
else {$Name = " Pferde. ";};
// Folgende Zeile macht eine Ausgabe:
echo ("Ich habe " . $Anzahl . $Name);
$Anzahl = $Anzahl + 1;
};
?>
```

Die Ausgabe dieses Skriptes soll wie folgt aussehen:

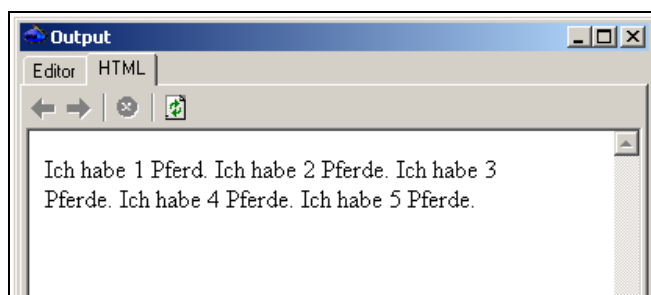


Abbildung 6: Ausgabe eines PHP-Skriptes; vierter Schritt

Die Variablen müssen bei der PHP nicht im Voraus definiert werden. Eine Variable entsteht einfach, wenn sie im Programm genannt wird. Mit Hilfe der Funktion „isset();“ kann kontrolliert werden, ob eine Variable schon existiert. Eine Variable wird gelöscht, in dem man die Variable mit der Funktion „unset();“ auflöst. Das ist besonders bei arbeitsspeicherintensiven Applikationen nötig, da man durch Auflösen der Variablen dafür benötigten Arbeitsspeicher räumt. Die Variablentypen werden bei PHP dynamisch kontrolliert. Beim folgenden Skript ist der Typ der Variablen \$Anzahl auf „INT“ (Integer) gestellt, weil am Anfang des Skriptes dieser

Variablen ein Zahlenwert zugewiesen wurde. Der Typ der Variablen wird automatisch auf „TEXT“ gestellt, wenn man einen alphanumerischen Wert zuweist:

```
<?php
$Anzahl = 1;
while ($Anzahl <= 5){
$n = $Anzahl;
if ($Anzahl == 1) { $Name = "Pferd. ";}
else {$Name = "Pferde. "};
//Folgende Zeilen geben neue Werte zu den Variable $Anzahl:
if ($Anzahl == 1){$Anzahl = "ein";};
if ($Anzahl == 2){$Anzahl = "zwei";};
if ($Anzahl == 3){$Anzahl = "drei";};
if ($Anzahl == 4){$Anzahl = "vier";};
if ($Anzahl == 5){$Anzahl = "fünf";};
// Folgende Zeile macht eine Ausgabe:
echo ("Ich habe " . $Anzahl . $Name);
//Folgende Zeile gibt $Anzahl seinen ehemaligen Wert zurück:
$Anzahl = $n;
$Anzahl = $Anzahl + 1;
};
?>
```

Die Ausgabe dieses Skriptes soll wie folgt aussehen:

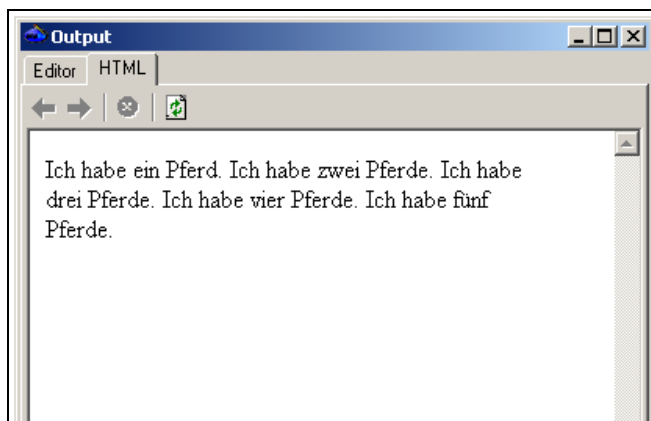


Abbildung 7: Ausgabe eines PHP-Skriptes; fünfter Schritt

Was bei diesem Skript fehlt, ist die Ordnung der Ausgabe. Angenommen, dass dieses Skript als HTML-Ausgabe programmiert würde, fehlen noch HTML-Bausteine. Hier werden diese fehlenden Elemente eingebaut:

```
<HTML>
<HEAD>
</HEAD>
<BODY>

<?php
$Anzahl = 1;
while ($Anzahl <= 5){
$n = $Anzahl;
if ($Anzahl == 1) { $Name = "Pferd. ";}
else {$Name = "Pferde. "};
if ($Anzahl == 1){$Anzahl = "ein";};
if ($Anzahl == 2){$Anzahl = "zwei";};
if ($Anzahl == 3){$Anzahl = "drei";};
if ($Anzahl == 4){$Anzahl = "vier";};
```

```
if ($Anzahl == 5){$Anzahl = "fünf";};  
echo ("<P>Ich habe " . $Anzahl . "&nbsp;" . $Name. "</P>");  
$Anzahl = $n;  
$Anzahl = $Anzahl + 1;  
};  
?>
```

```
<BODY>  
</HTML>
```

Das Ergebnis dieses Skriptes wird in einem Webbrowser wie folgt aussehen:

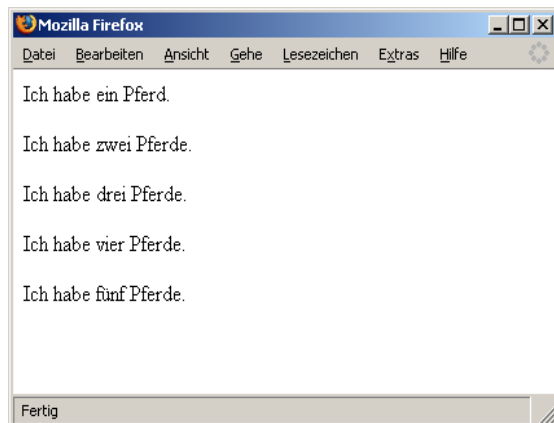


Abbildung 8: Ausgabe des PHP-Skriptes im Browserfenster

Der Quellcode dieser Webseite wird wie folgt aussehen:

```
<HTML>  
<HEAD>  
</HEAD>  
<BODY>  
  
<P>Ich habe ei n&nbsp;Pferd. </P>  
<P>Ich habe zwei &nbsp;Pferde. </P>  
<P>Ich habe drei &nbsp;Pferde. </P>  
<P>Ich habe vi er&nbsp;Pferde. </P>  
<P>Ich habe fünf&nbsp;Pferde. </P>  
  
<BODY>  
</HTML>
```

Wie oben, bei der Ausgabe eines PHP-Skriptes, kann man nicht erkennen, dass es sich um eine automatisch generierte Ausgabe handelt, weil die Ausgabe eine ganz normale HTML-Datei ist.

Funktionen und Serverside-Includes im PHP

PHP wird meistens dazu verwendet, um häufig wiederholte Arbeitsvorgänge zu erledigen. Auch innerhalb eines PHP-Skriptes kann es Codes geben, die häufig wiederholt werden müssen. Um den gleichen Code nicht wiederholt schreiben zu müssen, kann man die häufig benutzten Teile des Codes als eine

„Funktion“ definieren und diese Funktion immer aufrufen, wo sie gebraucht wird.

Das oben beschriebene Skript würde mit einer Funktionsdefinition so aussehen:

```
<HTML>
<HEAD>
</HEAD>
<BODY>

<?php
//Hier wird die Funktion Zahl_Name definiert:
function Zahl_Name(){
if ($Anzahl == 1){echo ("ei n");};
if ($Anzahl == 2){ echo ("zwei");};
if ($Anzahl == 3){echo ("drei");};
if ($Anzahl == 4){echo ("vi er");};
if ($Anzahl == 5){echo ("fünf");};
};
$Anzahl = 1;
while ($Anzahl <= 5){
if ($Anzahl == 1) { $Name = "Pferd.";}
else {$Name = "Pferde."};
//Hier wird die Funktion Zahl_Name aufgerufen:
$Zahl = Zahl_Name();
echo ("<P>Ich habe " . $Zahl . "&nbsp;". $Name. "</P>");
$Anzahl = $Anzahl + 1;
};
?>

<BODY>
</HTML>
```

Dieses Skript ergibt das gleiche Ergebnis. Man kann sich die Frage stellen, wozu diese Funktion gut ist, da das Skript nicht kürzer geworden ist. Dieses Skript ist nicht kürzer geworden, weil die genannte Funktion nur einmal eingesetzt wurde. Es kommt aber in der Praxis sehr häufig vor, dass man eine Funktion oftmals in einem Skript verwendet. Meistens werden alle Funktionen in separaten Dateien gespeichert und werden von vielen Skripten von diesen separaten Dateien als Komponenten abgerufen. Das spart dem Programmierer viel Zeit bei der Programmierung und hilft bei der Fehlerbeseitigung, weil man immer die Komponenten benutzt, die schon häufig verwendet und geprüft wurden. Aufrufen der anderen Dateien wird Serverside-Includes genannt, da die Dateien an der Serverseite in andere Dateien eingebettet werden. Folgendes Skript ist eine verfeinerte Version der Funktion des obigen Beispiels als eine separate Datei:

```
<?php
function Zahl_Name($var) {
if ($var == 0){$text= "kein";}
if ($var == 1){$text= "ein";}
elseif ($var == 2){ $text= "zwei ";}
elseif ($var == 3){$text= "drei ";}
elseif ($var == 4){$text= "vier ";}
elseif ($var == 5){$text= "fünf ";}
elseif ($var > 5){$text= "mehr als fünf ";}
return $text;
unset ($text);
```



```
unset($var);  
};  
?>
```

Angenommen dass diese Funktion unter dem Namen „zahl_text.php“ gespeichert wurde, könnte man diese Funktion jetzt als eine Serverside-Include aufrufen:

```
<HTML>  
<HEAD>  
</HEAD>  
<BODY>  
  
<?php  
//Hier wird die Datei zahl_text.php aufgerufen:  
include ('zahl_text.php');  
$Anzahl = 0;  
while ($Anzahl <= 7){  
if ($Anzahl <= 1) { $Name = "Pferd. "; }  
else { $Name = "Pferde. "; }  
//Hier wird die Funktion Zahl_Name aufgerufen:  
$Zahl_text = Zahl_Name($Anzahl);  
echo ("<P>Ich habe " . $Zahl_text . "&nbsp;". $Name. "</P>");  
$Anzahl = $Anzahl + 1;  
};  
?>  
  
<BODY>  
</HTML>
```

Dieses Skript würde im Browserfenster wie folgt aussehen:

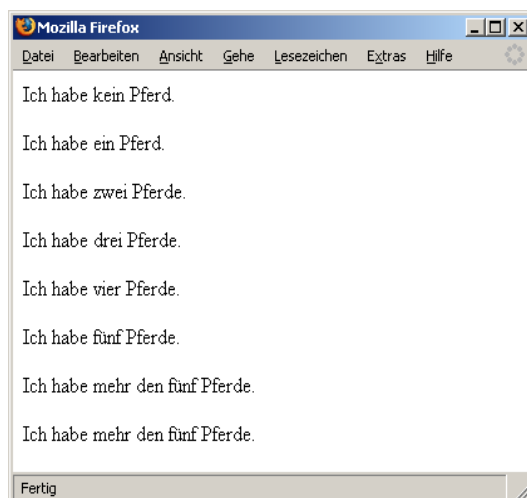


Abbildung 9: Ausgabe des PHP-Skriptes im Browserfenster

Man könnte anstatt des Kommandos „include();“ auch „require();“ verwenden. Der Unterschied zwischen beiden Kommandos liegt darin, dass das „include();“ Kommando die Datei erst aufruft, wenn es durchgeführt wird. Das Kommando „require();“ dagegen, ruft die angegebene Datei immer auf, auch wenn der Abschnitt wo dieser Befehl sich befindet nicht ausgeführt wird. Deshalb sollte man „require();“ nur verwenden, wenn man unbedingt sicher sein will, dass die Serverside-Include aufgerufen wird, auch wenn es nicht benötigt wird. Dabei sollte man darauf achten, dass jeder zusätzliche Zugriff auf die Festplatte und jede eingelesene Zeile von einer externen Datei ein paar CPU-Zyklen benötigt und dadurch das Programm verlangsamt. Während der Leistungsoptimierungsphase der Programmierung ist es sinnvoll, genau zu entscheiden, mit welcher Methode die Einbettung der externen Dateien erfolgen soll.

Sitzungsmanagement und Cookies im PHP

Eine Sitzung (engl. Session) bezeichnet in der EDV eine stehende Verbindung eines Clients mit einem Server, wobei die Daten, die durch die Aktivitäten während einer Sitzung entstanden sind, Sitzungsdaten genannt werden. Bei der HTTP-Datenübertragung gibt es keine konstanten Verbindungen zwischen dem Benutzer und dem Server (THE INTERNET SOCIETY, 1999), was die Zuordnung der Sitzungsdaten erschwert. Sitzungsdaten werden in Dateien namens Cookies server- oder clientseitig gespeichert und sind oft für komplexe Transaktionen nötig. PHP bietet eine fortschrittliche Funktionalität für das Sitzungsmanagement mit ganz einfachen Befehlsstrukturen. Eine Sitzung wird mit dem Kommando „session_start();“ gestartet und mit dem Kommando „session_destroy();“ beendet. Aufgrund der Struktur des HTTP-Protokolls kann die Funktion „session_start();“ nur verwendet werden, bevor das Skript eine Ausgabe an den HTTP-Port produziert hat. Diese Einschränkung kann mit der Verwendung des Funktionspaars „ob_start(); und ob_end_flush();“ überwunden werden, wobei der Server die gesamte Ausgabe zwischenspeichert und dann erst an den Browser sendet. Andererseits sollte man dadurch ein Verlust an Geschwindigkeit des Programms erwarten.

Wenn man eine Sitzung startet, wird durch eine Zufallsfunktion eine theoretisch einmalige Identifikationsnummer (Sitzungsnummer) generiert, die zur Kennzeichnung der Sitzung dient. Diese Nummer wird in eine Cookie-Datei auf dem Rechner des Benutzers gespeichert. Beim PHP beinhalten die Cookies nur die Sitzungsnummer, solange man mit dem Befehl „setcookie();“ keine zusätzlichen

Informationen hinzufügt. Alle anderen Informationen zur Sitzung werden serverseitig gespeichert, in so genannten Serverside-Cookies. Der Benutzer wird mit Hilfe der eindeutigen Sitzungsnummer, die auf seinem Rechner gespeichert ist, den serverseitigen Daten zugeordnet. Falls man keine Cookies auf der Benutzerseite speichern kann, wird die Sitzungsnummer in dem Adressstring bei jedem Schritt der Navigation an den Server übermittelt. Bei den Sitzungen mit Cookies kann man dieselbe Sitzung auch noch nach mehreren Tagen wieder erwecken, wobei die Sitzungen ohne Cookies nach dem Schließen des Browserfensters verloren gehen.

Die Sitzungsdaten werden in Variablen (sog. Sitzungsvariablen) gehalten, die in der Serverside-Cookie-Datei gespeichert werden. Die Sitzungsvariablen sind bis auf ein paar Ausnahmen Elemente der Globalvariablen `$_SESSION`. Folgendes Beispiel zeigt, wie Werte in Sitzungsvariablen gespeichert werden:

```
<?php
session_start();
$_SESSION['Benutzer_Anrede'] = "Herr";
$_SESSION['Benutzer_Vorname'] = "Max";
$_SESSION['Benutzer_Name'] = "Mustermann";
?>
```

In diesem Skript werden in den Sitzungsvariablen „Benutzer_Anrede“, „Benutzer_Vorname“ und „Benutzer_Name“ die Werte „Herr“, „Max“ und „Mustermann“ gespeichert. Diese Werte kann man nachher in einem anderen Skript wie folgt aufrufen:

```
<?php
session_start();
$Anrede = $_SESSION['Benutzer_Anrede'];
$Vorname = $_SESSION['Benutzer_Vorname'];
$Name = $_SESSION['Benutzer_Name'];
echo ("Willkommen $Anrede $Vorname $Name");
?>
```

Das Ergebnis des zweiten Skripts wird wie in der folgenden Abbildung aussehen, wenn es nach dem ersten Skript aufgerufen wird.

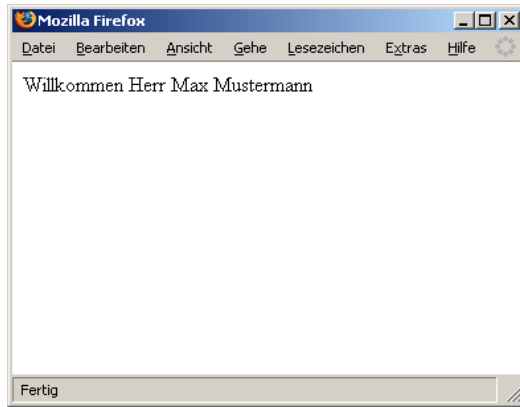


Abbildung 10: Ausgabe des PHP-Skriptes im Browserfenster

Datenbankverbindung und -abfragen beim PHP

Eine Stärke von PHP ist seine Unterstützung für eine breite Masse von Datenbanken. Die folgenden Datenbanken werden zurzeit unterstützt:

- Adabas D
- dBase
- Empress
- FilePro (nur Lesezugriff)
- FrontBase
- Hyperwave
- IBM DB2
- Informix
- Ingres
- InterBase
- mSQL
- MS SQL
- MySQL
- ODBC
- Oracle
- Ovrimos
- PostgreSQL
- Solid
- Sybase
- Unix dbm

Weiterhin unterstützt PHP ODBC und JDBC mit denen man die PHP-Skripten zu jeder anderen Datenbank verbinden kann, die diese Protokolle unterstützen. Um

eine Datenbankabfrage ausführen zu können, muss erst eine Verbindung mit dem Datenbankserver aufgebaut werden. Folgender Skriptausschnitt zeigt, wie eine Verbindung mit dem MySQL-Server aufgebaut wird:

```
<?php
$verbindung = @mysql_connect('host', 'benutzername', 'password');
if (!$verbindung) {
echo ("Datenbankverbindung fehlgeschlagen!");
exit();
}
else {
echo ("Datenbankverbindung erfolgreich!");
}
?>
```

Die Verbindung mit der Datenbankserver wird durch die Funktion „mysql_connect()“ aufgebaut. Diese Funktion bekommt als Parameter die IP- oder Domain-Adresse des Datenbankservers ('host'), und das Benutzername-Passwort-Paar für den Zugriff auf den Server ('benutzername', 'password'). Das @-Zeichen im obigen Quellcode verhindert die Ausgabe von Fehlermeldungen an die Benutzer, wenn ein Fehler beim Verbindungsaufbau auftritt. Die Verbindungsdaten werden in der Variablen \$Verbindung gespeichert und danach wird mit Hilfe der if-else-Funktion überprüft, ob eine erfolgreiche Verbindung vorliegt. Bei einer erfolgreichen Verbindung wird dieselbe Variable als eindeutige Identifikationsreferenz bei den Datenbankabfragen verwendet wird, damit PHP einordnen kann, auf welche Datenbankverbindung sich die Abfrage bezieht. Im Fall eines Fehlers wird bei diesem Code keine Fehlermeldung produziert, da sie mit @-Zeichen unterdrückt wird. Die Fehlermeldungen können aber auch nach der Durchführung der Datenbankbefehle (Datenbankverbindung, Datenbankauswahl, sowie SQL-Abfragen und andere Datenoperationen) mit Hilfe der Funktion „mysql_error()“ abgefragt werden. Dadurch ergeben sich bessere Möglichkeiten die Fehler zu kontrollieren und das Programm für Reaktionen bei wahrscheinlichen Fehlern vorzubereiten.

Ein Datenbankserver kann mehrere Datenbanken beinhalten und gleichzeitig zur Verfügung stellen. Deswegen sollte man nach dem Verbindungsaufbau die Datenbank auswählen, mit der man arbeiten will. Dieses wird bei einem MySQL-Server mit Hilfe der Funktion „mysql_select_db()“ realisiert:

```
<?php
$verbindung = @mysql_connect('host', 'benutzername', 'password');
if (!$verbindung) {
$fehlermeldung = mysql_error($verbindung);
echo ("Datenbankverbindung fehlgeschlagen! Die Meldung zu diesem Fehler lautet: $fehlermeldung");
exit();
}
else {
echo ("Datenbankverbindung erfolgreich!");
$datenbankauswahl = @mysql_select_db('datenbank', $verbindung);
if (!$datenbankauswahl) {
```

```
echo ("Fehler bei der Datenbankauswahl, die produzierte
Fehl ermel dung lautet: ". mysql _error($verbi ndung)); }
else { echo("Datenbankauswahl erfol grei ch"); }
};
?>
```

Nach einer erfolgreichen Datenbankauswahl können die SQL-Abfragen durchgeführt werden. Die SQL-Abfragen werden durch die Funktion „mysql_query()“ an den Datenbankserver gerichtet und die Antwort der Datenbank wird in einer Variablen in Form eines Arrays gespeichert.

Die Beispiele dieses Kapitels wurden mit den folgenden Annahmen gebildet:

IP-Adresse der MySQL-Server: 192.168.1.1
 Benutzername/Passwort für Zugriff: benutzer/passwort
 Name der Datenbank: PFERDE
 Name der Tabelle: PFERDINFO
 Datenfelder in der Tabelle:

Field	Type	Null	Key	Default	Extra
ID	bigint(20) unsigned	YES	PRI	NULL	auto_increment
PFERD	varchar(16)	YES		NULL	
PFERDNAME	varchar(26)	YES		NULL	
GESCHLECHT	varchar(13)	YES		NULL	
VERBAND	varchar(10)	YES		NULL	

5 rows in set (0.00 sec)

Daten in der Tabelle:

ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
1001	DE 309090083138	Donna	W	Trak.
1533	DE 331180082388	Donner	M	Hann.
1535	DE 331180082597	Donner Ja	M	Hann.
1536	DE 331180082664	Donnerkel I	M	Hann.
3309	DE 331310635591	Donnerpri nz	M	Hann.
3310	DE 331310637694	Donna Kl ara	W	Hann.
3911	DE 331311801182	Donna Mari e	W	Hann.
3921	DE 331311812177	Donnerwetter	M	Hann.
5190	DE 331316013254	Donnerfri ede	W	Hann.

9 rows in set (0.02 sec)

Eine ganz einfache Datenbankabfrage würde die Auswahl des gesamten Inhalts einer Tabelle sein, wie im folgenden SQL-Befehl:

```
SELECT * FROM PFERDE.PFERDEI NFO;
```

Dieser SQL-Befehl könnte wie folgt in einem PHP-Skript durchgeführt werden:

```
$sql abfrage = "SELECT * FROM PFERDE.PFERDEI NFO;";
$datenbankabfrage = mysql _query ($sql abfrage, $verbi ndung);
```

Die Ergebnisse dieser Abfrage werden in der Variablen `$datenbankabfrage` im Form eines Arrays gespeichert. Es beinhaltet nicht nur die Daten, sondern auch zusätzliche Informationen wie Anzahl der Ergebnisse, Dauer der Abfrage und Datengröße der Ergebnisse. Um die tatsächlichen Daten von diesem Array herauszuziehen, verwendet man eine der beiden Funktionen „`mysql_fetch_row()`“ oder `mysql_fetch_array()`“. Der Unterschied zwischen beiden Funktionen ist, dass die Ergebnisse bei der Funktion „`mysql_fetch_row()`“ als ein String und bei der „`mysql_fetch_array()`“ als ein Array gespeichert werden. Da die Array-Methode verschiedene Vorteile sowie bessere Geschwindigkeit gegenüber String-Methode aufweist, wird es meistens bevorzugt. Folgendes PHP-Skript-Fragment zeigt, wie diese Funktion eingesetzt wird:

```
$data = mysql_fetch_array($datenbankabfrage);
```

Die auf der Array-Variablen `$data` gespeicherten Daten können mit den Feldnamen aus der Tabelle z.B. `$data[PFERD]` aufgerufen werden. Man kann den Arraypointer mit Hilfe von Arrayfunktionen (z.B. `array_push()`;, `next()`;, `pos()`; und `prev()`;) positionieren und Daten auslesen. Ein einfacherer Weg zum Auslesen der Daten ist die Platzierung der Funktion „`mysql_fetch_array()`“ in einer WHILE-Schleife wie in dem folgenden Skript-Fragment:

```
while ($data = mysql_fetch_array($datenbankabfrage)){
    $PFERD = $data[PFERD]; };
```

Die ausgelesenen Daten können dann innerhalb oder außerhalb⁹ dieser WHILE-Schleife zur Erstellung einer Ausgabe oder zur Berechnungen sowie zur Erstellung weiterer SQL-Abfragen verwendet werden.

Folgendes PHP-Skript stellt den gesamten Inhalt der Tabelle auf einer HTML-Seite dar:

```
<html >
<head>
</head>
<body>
<?php

//Datenbankverbindung
$verbindung = @mysql_connect(' 192.168.1.1' , 'benutzer' , 'password' );
if (!$verbindung) {
    $fehlermeldung = mysql_error($verbindung);
    echo ("Datenbankverbindung fehlgeschlagen! Die Meldung zu diesem Fehler lautet: $fehlermeldung");
    exit(); };
```

⁹ Mit Hilfe der variablen Variablen (engl. Variable variables).

```

else {
echo ("Datenbankverbindung erfolgreich! <br>");
$datenbank_auswahl = @mysql_select_db(' PFERDE' , $verbindung);
if (!$datenbank_auswahl) {
echo ("Fehler bei der Datenbankauswahl, die produzierte
Fehlermeldung lautet: ". mysql_error($verbindung));}
else { echo("Datenbankauswahl erfolgreich! <br>");

//Datenbankabfrage
$sql_abfrage = "SELECT * FROM PFERDE.PFERDEINFO";
$datenbankabfrage = mysql_query ($sql_abfrage,$verbindung);

//Anzahl der Ergebnisse
$anzahl_ergebnisse = $mysql_num_rows($datenbankabfrage);
if ($anzahl_ergebnisse == 0) { echo ("Es gibt leider kein
Ergebnis!"); exit();}
elseif ($anzahl_ergebnisse == 1) { echo ("Es gibt nur ein
Ergebnis. <br>");}
else { echo ("Es gibt $anzahl_ergebnisse Ergebnisse. <br>");}

//Auslesen der Ergebnisse
while ($data = mysql_fetch_array($datenbankabfrage)){

$ID = $data[ID];
$PFERD = $data[PFERD];
$PFERDNAME = $data[PFERDNAME];
$GESCHLECHT = $data[GESCHLECHT];
$VERBAND = $data[VERBAND];

//Inhaltliche Anpassungen der Ergebnisse
if ($GESCHLECHT == "M") { $GESCHLECHT = "Hengst";}
elseif ($GESCHLECHT == "W") { $GESCHLECHT = "Stute";}
else { $GESCHLECHT = "Unbekannt";};

if ($VERBAND == "Hann.") { $VERBAND = "Hannoveraner Verband";}
elseif ($VERBAND == "Trak.") { $VERBAND = "Trakehner Verband";}
else { $VERBAND = "Unbekannter Verband";};

//Darstellung der Ergebnisse
echo ("<P>Name des Pferdes: $PFERDNAME<br>Lebensnummer:
$PFERD<br>Geschlecht: $GESCHLECHT<br>Verband: $VERBAND</P>");
};};};
?>
</body>
</html >

```

Dieses Skript produziert folgende HTML-Datei:

```

<html >
<head>
</head>
<body>
Datenbankverbindung erfolgreich! <br>
Datenbankauswahl erfolgreich! <br>
Es gibt 9 Ergebnisse. <br>
<P>Name des Pferdes: Donna<br>Lebensnummer: DE
309090083138<br>Geschlecht: Stute<br>Verband: Trakehner Verband</P>
<P>Name des Pferdes: Donner Ja<br>Lebensnummer: DE
331180082388<br>Geschlecht: Hengst<br>Verband: Hannoveraner
Verband</P>
<P>Name des Pferdes: Donner Ja<br>Lebensnummer: DE
331180082597<br>Geschlecht: Hengst<br>Verband: Hannoveraner
Verband</P>
<P>Name des Pferdes: Donnerkeil<br>Lebensnummer: DE
331180082664<br>Geschlecht: Hengst<br>Verband: Hannoveraner
Verband</P>

```



```
<P>Name des Pferdes: Donnerprinz<br>Lebensnummer: DE  
331310635591<br>Geschlecht: Hengst<br>Verband: Hannoveraner  
Verband</P>  
<P>Name des Pferdes: Donna Klara<br>Lebensnummer: DE  
331310637694<br>Geschlecht: Stute<br>Verband: Hannoveraner  
Verband</P>  
<P>Name des Pferdes: Donna Marie<br>Lebensnummer: DE  
331311801182<br>Geschlecht: Stute<br>Verband: Hannoveraner  
Verband</P>  
<P>Name des Pferdes: Donnerwetter<br>Lebensnummer: DE  
331311812177<br>Geschlecht: Hengst<br>Verband: Hannoveraner  
Verband</P>  
<P>Name des Pferdes: Donnerfriede<br>Lebensnummer: DE  
331316013254<br>Geschlecht: Stute<br>Verband: Hannoveraner  
Verband</P>  
</body>  
</html >
```

Das Ergebnis des PHP-Skripts sieht im Browserfenster wie in der folgenden Abbildung aus:

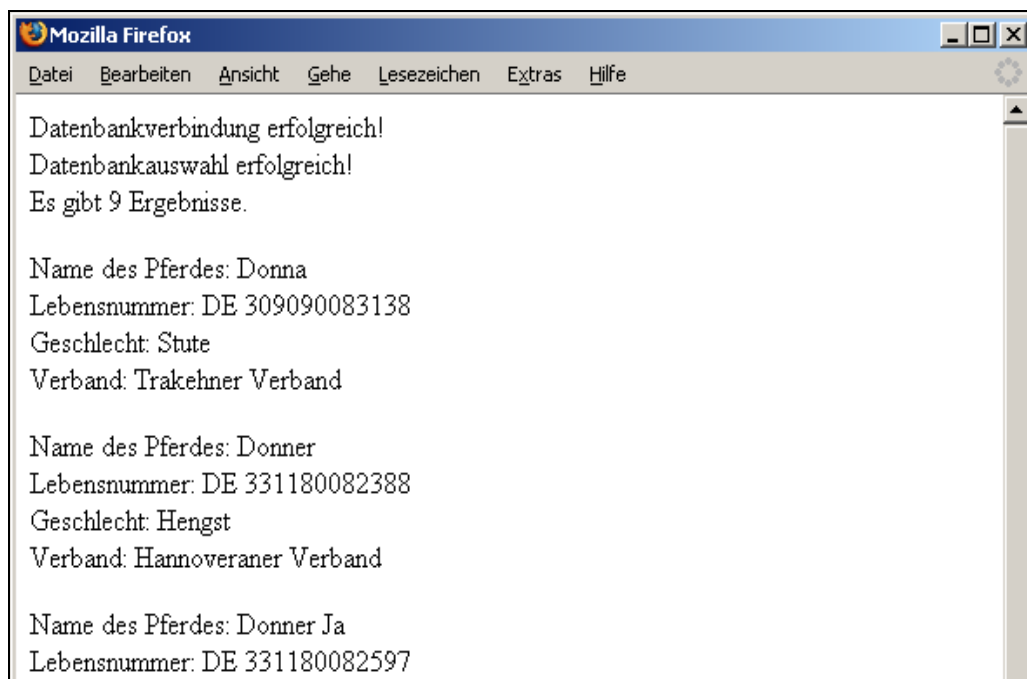


Abbildung 11: Ausgabe eines PHP-Skripts im Browserfenster

PHP ist eine sehr komplexe und umfangreiche Programmiersprache, die zum größten Teil ausführlich dokumentiert ist. Die in diesem Kapitel vorgestellten Beispiele zeigen nur einen kleinen Ausschnitt der Funktionen von PHP und sollten dazu dienen, ein grundsätzliches Verständnis für das in dieser Arbeit konzipierte Programm aufzuzeigen. Für vertiefte Informationen zur PHP-Programmierung sollte das offizielle PHP-Handbuch zu Rate gezogen werden [PHP].

3.4.2. Structured Query Language (SQL)

SQL wurde Ende der 70'er Jahre in den IBM Forschungslaboratorien in San José (Kalifornien) von R.F. Boyce und D.D. Chamberlain im Rahmen des System-R Projektes entwickelt. Es ging darin um die Erforschung des praktischen Ansatzes der Relationalen Datenbankmodelle, die im Jahr 1970 von E.F. Codd von IBM entworfen wurden. SQL, früher als SEQUEL (Structured English Query Language) bezeichnet, ist eine nichtprozedurale Programmiersprache, mit der man speziell nur an Datenbanken arbeiten kann. Die Einführung der ersten kommerziellen RDBMS von Oracle Corp. im Jahr 1979, ebnete für die SQL Sprache den Weg zur Industrienorm. 1982 bildete ANSI (American National Standards Institute) ein Komitee zur Normierung der SQL Sprache. Der erste SQL-Standard wurde im Jahr 1986 festgelegt und als SQL-86 bezeichnet. Er wurde auch als SQL-87 bekannt, weil er Anfang 1987 publiziert wurde. Die ISO (International Standards Organisation) übernahm diesen Standard und entwickelte ihn weiter. Im Laufe der Jahre wurden immer mehr Eigenschaften hinzugefügt und unter neuen Standardnamen publiziert. Der aktuellste Stand ist als SQL-99 (ANSI) und SQL-3 (ISO) bekannt und wurde im Jahr 1999 publiziert. Viele RDBMS Produkte weichen von diesen Normen ab. Die Abweichungen der Systeme von den ISO-Normen lassen sich generell in zwei Richtungen gliedern:

1. Funktionen werden nur teilweise implementiert oder sind nicht in der standardisierten Form vorhanden.
2. Es wird der Funktionsumfang in inkompatibler Weise erweitert, um sich von der Konkurrenz abzugrenzen bzw. einen Marktvorteil zu erreichen.

Trotz aller Differenzen zwischen RDBM Systemen ist die Basis der SQL Befehle und deren Anwendung fast ähnlich. Jede SQL Anweisung beginnt mit einem Befehl und endet mit einem Semikolon-Zeichen. Einen Überblick über die wichtigsten Befehle in SQL gibt die Tabelle 10.

Tabelle 10: Die wichtigsten SQL-Befehle

SQL-Befehl	Erklärung
CREATE	Definition von Basistabellen
ALTER	Tabellendefinition ändern
GRANT	Autorisierung (Vergabe von Zugriffsrechten)
SELECT	Zentrale Anweisung zur Datenauswahl
INSERT	Hinzufügen von Zeilen
DELETE	Löschen von Zeilen
UPDATE	Verändern von Zeilen durch Änderung von Spalteninhalten

Für die folgenden Beispiele wurde angenommen, dass die Datenbank PFERDE und die Tabelle PFERDINFO heißt. Im folgenden eine Tabelle für weitere Erklärungen:

Tabelle 11: Die Tabelle für die SQL-Beispiele

		DATENFELDER				
		ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
DATEN		1001	DE 309090083138	Donna	W	Trak.
		1533	DE 331180082388	Donner	M	Hann.
		1535	DE 331180082597	Donner Ja	M	Hann.
		1536	DE 331180082664	Donnerkeil	M	Hann.
		3309	DE 331310635591	Donnerprinz	M	Hann.
		3310	DE 331310637694	Donna Klara	W	Hann.
		3911	DE 331311801182	Donna Marie	W	Hann.
		3921	DE 331311812177	Donnerwetter	M	Hann.
		5190	DE 331316013254	Donnerfriede	W	Hann.

Um eine Abfrage zu gestalten, braucht man den Namen der Datenbank, den Namen der Tabelle und die Namen der Datenfelder. Die Namen der Datenfelder können auch abgefragt werden:

>SHOW FIELDS FROM PFERDE.PFERDINFO;

Ergibt das Ergebnis:

Field	Type	Null	Key	Default	Extra
ID	bigint(20) unsigned		PRI	NULL	auto_increment
PFERD	varchar(16)	YES		NULL	
PFERDNAME	varchar(26)	YES		NULL	
GESCHLECHT	varchar(13)	YES		NULL	
VERBAND	varchar(10)	YES		NULL	

5 rows in set (0.00 sec)

An diesem Ergebnis kann man sehen, welche Eigenschaften die Datenfelder in der Tabelle haben. Mit Hilfe des SELECT-Befehls kann man Daten wählen, die zu den in der Anweisung vorgegebenen Bedingungen passen:

SELECT * FROM PFERDE.PFERDINFO;

Ergibt den ganzen Inhalt der Tabelle:

ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
1001	DE 309090083138	Donna	W	Trak.
1533	DE 331180082388	Donner	M	Hann.
1535	DE 331180082597	Donner Ja	M	Hann.
1536	DE 331180082664	Donnerkeil	M	Hann.
3309	DE 331310635591	Donnerprinz	M	Hann.
3310	DE 331310637694	Donna Klara	W	Hann.
3911	DE 331311801182	Donna Marie	W	Hann.
3921	DE 331311812177	Donnerwetter	M	Hann.
5190	DE 331316013254	Donnerfriede	W	Hann.

9 rows in set (0.02 sec)

Die letzte Zeile „9 rows in set (0.02 sec)“ zeigt die Anzahl der gewählten Daten und die Zeit, die zur Durchführung dieser Anweisung gebraucht wurde. Der Stern nach dem SELECT-Befehl entspricht den Namen der Datenfelder, die ausgewählt werden. Mit einem Stern wählt man alle Datenfelder. Wenn man nur bestimmte Datenfelder auswählen will, dann müssen nach dem SELECT-Befehl die Namen dieser Datenfelder eingesetzt werden:

```
SELECT ID, PFERD, PFERDNAME FROM PFERDE.PFERDINFO;
```

Ergibt den ganzen Inhalt der Tabelle nur mit den gewählten Datenfeldern:

ID	PFERD	PFERDNAME
1001	DE 309090083138	Donna
1533	DE 331180082388	Donner
1535	DE 331180082597	Donner Ja
1536	DE 331180082664	Donnerkei I
3309	DE 331310635591	Donnerpri nz
3310	DE 331310637694	Donna Kl ara
3911	DE 331311801182	Donna Marie
3921	DE 331311812177	Donnerwetter
5190	DE 331316013254	Donnerfri ede

9 rows in set (0.02 sec)

Man benutzt den SELECT-Befehl meistens, um eine bestimmte Gruppe der Daten aus dem gesamten Dateninhalt zu selektieren. Das erfolgt mit Eingabe von Bedingungen, zu denen die ausgesuchten Daten passen müssen. Diese Bedingungen werden mit Hilfe des WHERE-Befehls gestellt. In der folgenden Anweisung wird nach den weiblichen Pferden gesucht:

```
SELECT * FROM PFERDE.PFERDINFO WHERE GESCHLECHT = 'W' ;
```

Ergibt die Daten, bei denen das Datenfeld Geschlecht den Wert “W” hat:

ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
1001	DE 309090083138	Donna	W	Trak.
3310	DE 331310637694	Donna Kl ara	W	Hann.
3911	DE 331311801182	Donna Marie	W	Hann.
5190	DE 331316013254	Donnerfri ede	W	Hann.

4 rows in set (0.01 sec)

Das Gleichheitszeichen “=” in den Anweisungen nennt man einen Operator. Operatoren sind mathematisch und logisch vordefinierte Bedingungen. Bei der Definition von Bedingungen können die Operatoren in der Tabelle 6 benutzt werden. Außer Operatoren gibt es zwei Joker-Zeichen, die man in den Bedingungen benutzen kann. Das erste ist das Prozentzeichen „%“, das wie ein Sternzeichen bei DOS wirkt. Man kann mit dem „%“ beliebig viele (oder auch keine) Zifferstellen ersetzen. Das zweite ist ein Unterstrich „_“. Dieser funktioniert wie ein Fragezeichen unter DOS. Mit dem „_“ kann man nur eine Zifferstelle ersetzen. Diese Joker-Zeichen funktionieren nur mit den Operatoren LIKE und NOT LIKE, die zur

Mustererkennung dienen. Diese Operatoren sind besonders wichtig, weil in einem großen Teil der Suchoperationen versucht wird, mit Hilfe der teilweise gegebenen Suchwörter die entsprechenden Daten zu finden. Obwohl diese Operatoren in dem SQL-99 Standard als „Groß-/Kleinschreibung Abhängig“ definiert sind, weichen viele Datenbanksysteme von dieser Regelung mit dem Ziel ab, mehrere Treffer in einer Mustererkennungssuche zu erzielen.

Tabelle 12: Liste der Operatoren in SQL

Operator	Name	Erklärung
=	Gleich	Kontrolliert, ob zwei Werte gleich sind
>	Größer	Kontrolliert, ob ein Wert größer als der andere ist
<	Kleiner	Kontrolliert, ob ein Wert kleiner als der andere ist
>=	Größer oder Gleich	Kontrolliert, ob ein Wert größer oder gleich zu dem anderen Wert ist
=<	Kleiner oder Gleich	Kontrolliert, ob ein Wert kleiner oder gleich zu dem anderen Wert ist
!= oder <>	Nicht gleich	Kontrolliert, ob zwei Werte nicht gleich sind
IS NOT NULL	Adress hat einen Wert	Kontrolliert, ob das Feld einen Wert hat
IS NULL	Adress hat keinen Wert	Kontrolliert, ob das Feld keinen Wert hat
BETWEEN	Zwischen	Kontrolliert, ob der Wert gleich oder größer als das Minimum und gleich oder kleiner als Maximum ist
IN	Innerhalb Menge	Kontrolliert, ob der Wert zu einer Menge gehört
NOT IN	Ausserhalb Menge	Kontrolliert, ob der Wert nicht zu einer Menge gehört
LIKE	Muster Gleichung	Kontrolliert, ob zwei Werte ähnlich sind
NOT LIKE	Muster Gleichung	Kontrolliert, ob zwei Werte nicht ähnlich sind
AND und OR	Verbindung Ausdruck	Verbindet mehrere Bedingungen

Eine Anweisung wie `SELECT * FROM PFERDE.PFERDINFO WHERE ID = "1%";` wird kein Ergebnis produzieren, weil in den Datenfeldern genau nach "1%" gesucht wird. Aber wenn man die gleiche Anweisung mit dem LIKE-Operator benutzt, bekommt man Ergebnisse:

```
SELECT * FROM PFERDE.PFERDINFO WHERE ID LIKE "1%";
```

Ergibt ein Ergebnis mit den IDs, die am Anfang eine 1 und dann weitere beliebige Ziffern haben:

ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
1001	DE 309090083138	Donna	W	Trak.
1533	DE 331180082388	Donner	M	Hann.
1535	DE 331180082597	Donner Ja	M	Hann.
1536	DE 331180082664	Donnerkei l	M	Hann.

4 rows in set (0.02 sec)

Man kann die Joker-Zeichen auch miteinander kombiniert benutzen:

```
SELECT * FROM PFERDE.PFERDINFO WHERE PFERD LIKE "%9_";
```

Ergibt das Ergebnis:

ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
1535	DE 331180082597	Donner Ja	M	Hann.
3309	DE 331310635591	Donnerpri nz	M	Hann.
3310	DE 331310637694	Donna Kl ara	W	Hann.

3 rows in set (0.01 sec)

Mehrere Bedingungen können mit Hilfe der Verbindungsausdrücke „AND“ und „OR“ miteinander verbunden werden. Die Bedingungen, die mit „AND“ verbunden sind, ergeben nur dann ein Ergebnis, wenn alle Bedingungen erfüllt sind. Die Bedingungen, die mit „OR“ verbunden sind, geben schon dann ein Ergebnis, wenn mindestens eine der Bedingungen erfüllt ist. Wenn es mehr als zwei Bedingungen gibt, werden die Bedingungen von links nach rechts gelesen und bewertet. Dabei sollte die Reihenfolge der Bedingungen und deren Verbindungen beachtet werden. Im nachstehenden Beispiel ergeben die folgenden Befehle nicht das gleiche Ergebnis:

```
1- SELECT * FROM PFERDE.PFERDINFO WHERE ID LIKE "1%" OR ID LIKE "%3%" AND GESCHLECHT ="W";
```

Ergibt 6 Treffer:

ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
1001	DE 309090083138	Donna	W	Trak.
1533	DE 331180082388	Donner	M	Hann.
1535	DE 331180082597	Donner Ja	M	Hann.
1536	DE 331180082664	Donnerkei l	M	Hann.
3310	DE 331310637694	Donna Kl ara	W	Hann.
3911	DE 331311801182	Donna Mari e	W	Hann.

6 rows in set (0.02 sec)

```
2- SELECT * FROM PFERDE.PFERDINFO WHERE ID LIKE "1%" AND ID LIKE "%3%" OR GESCHLECHT ="W";
```

Ergibt 7 Treffer:

ID	PFERD	PFERDNAME	GESCHLECHT	VERBAND
1001	DE 309090083138	Donna	W	Trak.
1533	DE 331180082388	Donner	M	Hann.
1535	DE 331180082597	Donner Ja	M	Hann.
1536	DE 331180082664	Donnerkei l	M	Hann.
3310	DE 331310637694	Donna Kl ara	W	Hann.
3911	DE 331311801182	Donna Mari e	W	Hann.
5190	DE 331316013254	Donnerfri ede	W	Hann.

8 rows in set (0.02 sec)

Bei dem ersten Befehl wird erwartet, dass die erste Ziffer von ID den Wert „1“ hat oder es irgendwo in der ID den Wert „3“ gibt und dass das Geschlecht den Wert „W“ hat. Deswegen kommen alle Pferde, deren ID mit 1 beginnt und nur die Pferde, die irgendwo in der ID den Wert 3 haben und dennoch weiblich sind als Treffer in die Auswahl. Bei dem zweiten Befehl wird erwartet, dass die ID des Pferdes mit 1 beginnt und es irgendwo in der ID noch eine 3 gibt oder unabhängig von der ID, dass das Pferd in dem Geschlechtsfeld den Wert „W“ hat. Somit kommen die Pferde mit den IDs 1001, 3310, 3911 und 5190 zur Auswahl, unabhängig von deren ID-Nummern.

In dieser Arbeit wurden die SQL-Anweisungen immer in PHP-Skripten eingebettet. PHP ermöglicht dem Programmierer, die SQL-Anweisungen so zu geben, wie man sie auch direkt an einer Datenbankkonsole eingibt. Eine typische Einbettung einer SQL-Anweisung ins PHP sieht so aus:

```
$frage_rennen= ("SELECT DISTINCT * FROM asp WHERE asp.asdat LIKE
'".$datum."%' and asp.asrvld='".$sortcode.'" and asrennr > 0 ORDER BY
(asrennr+0);", $dbcnx);
```

Mit dieser Anweisung werden die Ergebnisse in der Variablen \$frage_rennen als ein Array¹⁰ gespeichert. Die Variable \$dbcnx enthält die Verbindungsinformationen über den Datenbankserver, an den die SQL-Anweisung gerichtet wird.

¹⁰ Ein Array ist eine aus Zeilen und Spalten bestehende Matrix aus Daten.

4. Darstellung der Ergebnisse

4.1. Das Programm

Das Programm übernimmt die Rolle einer Schnittstelle, die die Kommunikation zwischen Benutzer und Datenbank ermöglicht und regelt. Während dieses Kommunikationsvorganges sichert das Programm, dass erstens der Benutzer eine für das System bekannte Persönlichkeit ist, zweitens die zu übertragenden Informationen in sich logisch sind und drittens diese Informationen an der Benutzerseite korrekt dargestellt werden. Diese Funktionalitäten werden durch die Zusammenarbeit von verschiedenen logischen Komponenten und Kontrollstrukturen des Programms erfüllt.

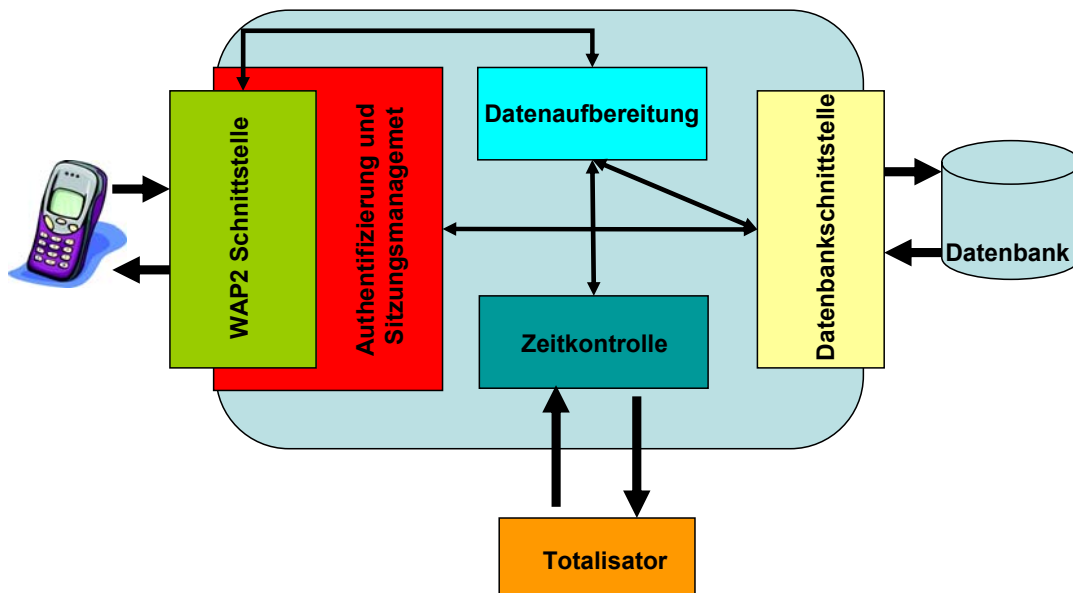


Abbildung 12: Logische Komponenten des Programms nach Funktionsbereichen

4.1.1. Komponenten

Das Programm hat folgende logische Komponenten, aufgeteilt nach Funktionsbereichen:

- Datenbankschnittstelle
- Authentifizierung und Sitzungsmanagement
- Datenaufbereitung
- WAP2 Schnittstelle
- Zeitkontrolle

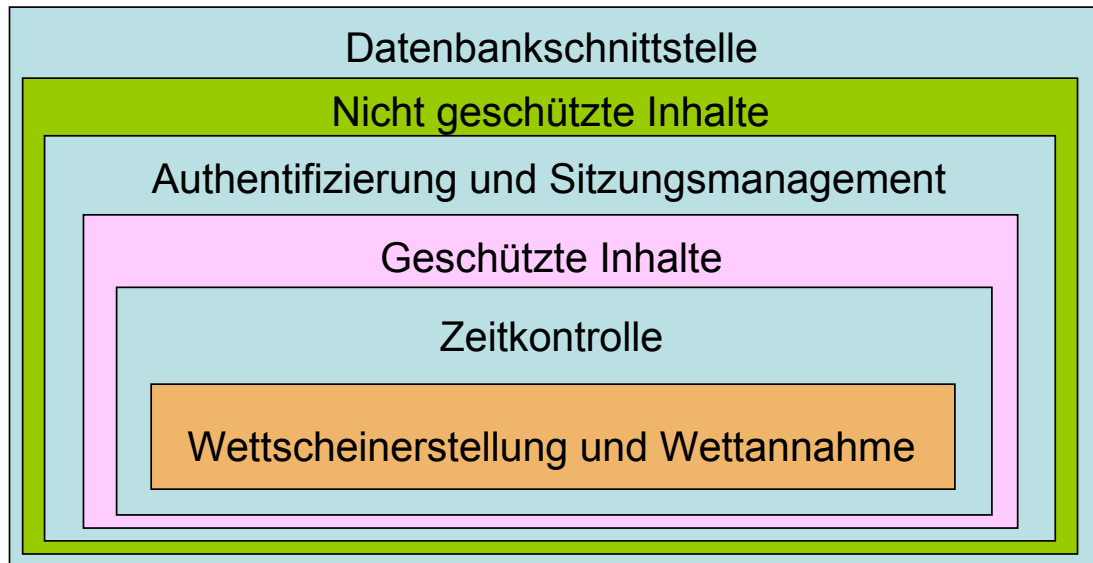


Abbildung 13: Darstellung der Kontrollebenen des Programms

Datenbankschnittstelle

Die Datenbankschnittstelle ist die Kommunikations- und Kontrollstruktur des Programms für Datenbankverbindungen und -abfragen. Sie bildet die äußerste Kontrollebene des Programms und sichert die erfolgreiche Durchführung der Datenbankabfragen. Diese ist die Grundvoraussetzung für das Funktionieren des Programms, da beinahe der gesamte Inhalt des Systems datenbankbezogen ist. Bei dieser Arbeit wurde die Datenbankschnittstelle für MySQL Datenbankverbindungen programmiert. Diese Schnittstelle kann aber ohne großen Aufwand für andere RDBMS-Systeme umgesetzt werden. Für die Datenbankverbindung sorgt das folgende PHP-Skript „datenbank.php“:

```
<?php
$dbcnx = @mysql_connect("host", "benutzername", "passwort");
if (!$dbcnx) {$verbindung="0"; $errno="1"; $page="err";}
else {
$select_db = @mysql_select_db("hvt", $dbcnx);
if (!$select_db) {$verbindung="0";}
else { $verbindung="1";};
};
?>
```

Dieses Skript bildet eine Datenbankverbindung über das Kommando „@mysql_connect(„host“, „benutzername“, „passwort“);“ mit der Adresse „host“ mit dem Benutzernamen „benutzername“ und dem Passwort „passwort“. Die Informationen über diese Verbindung werden in der Variablen \$dbcnx gespeichert. Wenn die Verbindung erfolgt, wird durch den Befehl „@mysql_select_db

(`"hvt", $dbcnx`); " die Datenbank „hvt“ gewählt. Sind diese zwei Schritte erfolgreich, bekommt die Variable `$verbindung` den Wert „1“, dies bedeutet, dass die Datenbankverbindung erfolgreich ist und für Abfragen bereit steht. Scheitert einer dieser Schritte, bekommt die Variable `$verbindung` den Wert „0“ und das Programm wird an die Fehlermeldungskontrolle weitergeleitet. Bei diesem Projekt handelt es sich nur um eine Datenbank mit mehreren Tabellen auf einem einzigen Host, deswegen kann man alle Datenbankabfragen mit der Variablen `$dbcnx` durchführen.

PHP bietet zwei Möglichkeiten der Datenbankverbindung: eine Kurzzeitverbindung und eine Permanentverbindung. Bei diesem Projekt wurde die Kurzzeitverbindung verwendet. Diese Verbindung wird getrennt, wenn nach einer definierten Zeiteinheit¹¹ keine Datenbankabfrage durchgeführt wird, oder die Verbindung mit dem Befehl `„mysql_close();“` getrennt wird. Bei dem Programm wird das Skript `„datenbank.php“` bei jedem Schritt aufgerufen. Besteht die Verbindung mit der Datenbank noch, wird sie automatisch weitergeführt, wenn sie getrennt ist, wird sie neu aufgebaut. Nach dem Aufruf von `„datenbank.php“` wird der Wert der Variablen `$verbindung` kontrolliert. Somit wird an jeder Stelle sichergestellt, dass die Datenbankverbindung problemlos funktioniert.

Um den Datenbankverbindung auf andere RDBMS umzustellen, sollte man die Befehle `„mysql_connect();“` und `„mysql_select_db();“` mit den Befehlen, die der Zieldatenbank entsprechen, ersetzen.

Authentifizierung und Sitzungsmanagement

Das Programm verfügt über zwei Bereiche: Einen nicht geschützten anonymen Bereich für Informationsangebote und einen durch Authentifizierung geschützten Bereich für Renn-, Wett- und Traberinformationen (Formen).

Innerhalb des nicht geschützten Bereichs können die Benutzer einige Informationen über den HVT (Kontaktadressen, Telefonnummern etc.) bekommen und mit Hilfe einer Suchmaschine nach Informationen über Pferde in der Traberstammdatenbank suchen.

¹¹ Einstellbar in der PHP Konfiguration

Der geschützte Bereich beinhaltet Informationen über aktuelle Rennen und bietet die Möglichkeit der Abgabe von Wettscheinen. Zusätzlich bietet dieser Bereich detaillierte Informationen über Traber und deren Formen an.

Die Authentifizierungs- und Sitzungsmanagementkomponente bildet die zweite Ebene der Kontrollstrukturen nach der Datenbankschnittstelle. Die Authentifizierung funktioniert mit einem Benutzername-Passwort-Paar für jeden Benutzer. Bei der Erstellung eines Benutzers für das Programm werden der Benutzername und das Passwort in die Tabelle „kunde“ gespeichert, wobei das Passwort nicht tatsächlich gespeichert wird, sondern nur die MD5-Signatur des Passwortes. MD5 ist ein Einwegberechnungsinstrument, mit dem man eine Signatur für eine Eingabe berechnen kann. Dieselbe Eingabe ergibt immer dieselbe Signatur und es gibt keine Möglichkeit die Eingabe von einer Signatur zu generieren¹². Damit hat selbst der Datenbankbetreiber keine Information über das Passwort des Benutzers, was eine sinnvolle Methode des internen Datenschutzes ist.

Meldet sich ein Benutzer durch die Anmeldefunktion an, wird die Angabe des Benutzers mit den Daten in der Datenbank verglichen. Nur wenn das Benutzername-Passwortsignatur-Paar mit dem in der Datenbank übereinstimmt, wird der Benutzer als authentifiziert angesehen. Der Entscheidungsprozess des Programms wird in der folgenden Tabelle dargestellt:

Tabelle 13: Entscheidungsprozess zur Benutzerauthentifizierung

Benutzername	Passwortsignatur	Authentifizierung
Richtig	Falsch	Nicht erfolgreich
Falsch	Irrelevant wenn kein passender Benutzername in der Datenbank gefunden wird.	Nicht erfolgreich
Richtig	Leer	Nicht erfolgreich
Leer	Irrelevant wenn kein passender Benutzername in der Datenbank gefunden wird.	Nicht erfolgreich
Richtig	Richtig	Erfolgreich

Nachdem der Benutzer sich erfolgreich eingeloggt hat, wird eine Sitzung gestartet¹³. Dabei wird an der Clientseite ein anonymes Cookie gespeichert, das nur ein einmaliges String als Sitzungsidentitätsnummer beinhaltet. Am Ende der Sitzung oder bei Abschaltung des Browsers wird dieses Cookie automatisch gelöscht. Bei

¹² Die einzige Methode um ein MD5-Signatur zu brechen wird „Brute-Force Attack“ genannt und funktioniert durch testen von allen möglichen Eingaben.

¹³ Die Sitzung ist in der Tat von Anfang an gestartet, aber die Funktionalität wird erst verwendet, wenn der Benutzer sich erfolgreich anmeldet.

jedem Neuaufwurf der Seite wird diese Identitätsnummer an den Server weitergeleitet, damit der Server die Sitzung erkennt. Alle Informationen zu Sitzung und Benutzer werden an der Serverseite mit einem serverseitigen Cookie gespeichert. Dieses Cookie wird mit der gleichen Identitätsnummer versehen, damit der Server jedes Mal die passenden Informationen zu der Sitzung finden kann. Serverseitige Cookies können nach Beendigung der Sitzung behalten werden, um die Kundengewohnheiten zu analysieren. Die Erhaltung des Serverseitigen Cookies ist keine Funktion des Programms, sondern hängt von den Einstellungen des PHP-Parsers ab. Bei dieser Arbeit wurde die PHP-Konfiguration mit normalen Voreinstellungen verwendet, d.h. serverseitige Cookies werden ähnlich wie die Clientseitigen Cookies gelöscht, wenn die Sitzung beendet oder das Browserfenster geschlossen wird. Der Benutzer kann die Sitzung beenden, in dem er auf den Link „Abmelden“ klickt.

Sitzungsmanagement kann in verschiedenen Richtungen eingesetzt werden. Bei dieser Arbeit wurde es für zwei Aufgaben verwendet. Die erste Aufgabe besteht darin, eine plattformunabhängige Benutzerauthentifizierung zu konzipieren, die zweite Aufgabe ist es, den ständigen Austausch der wiederholten Informationen zwischen Klient und Server zu verringern, in dem man diese Informationen durch Sitzungsvariablen an der Serverseite hält. Somit wird es ermöglicht, große Mengen von Informationen zu bearbeiten, ohne an die technischen Grenzen der mobilen Klienten zu geraten und außerdem die relativ hohen GSM-Datenübertragungskosten zu reduzieren.

Datenaufbereitung

Die Datenaufbereitungskomponente stellt alle Inhalte, die an den Benutzer gesendet werden, durch Durchführung mehrerer Datenbankabfragen und Integritätskontrolle der Daten her. Alle vom Benutzer angegebenen Daten werden auch von dieser Komponente bearbeitet. Folgende Funktionen liegen im Aufgabenbereich dieser Komponente:

- Erstellung, Durchführung und Bearbeitung der Ergebnisse der Datenbankabfragen
- Auflisten der Renntage, der Rennen an einem Renntag, der Starter eines Rennens
- Erstellung der Wettscheine für die aktuellen Rennen
- Bearbeitung, Kontrolle und Annahme der gespielten Wettscheine

- Erstellung der Informationsseiten über die Traber, dessen Eltern und Formen
- Darstellung der Benutzerinformationen und der gespielten Wettscheine

Diese Komponente kann auch als Brücke zwischen der WAP2-Schnittstelle und der Datenbankschnittstelle gesehen werden, die in zwei Richtungen funktioniert. In die eine Richtung wird der von der Datenbankschnittstelle gelesene Inhalt bearbeitet, und an die WAP2-Schnittstelle weitergeleitet. In die andere Richtung werden die Eingaben, die durch die WAP2-Schnittstelle gelesen werden, bearbeitet und als SQL-Abfragen an die Datenbankschnittstelle weitergeleitet. Diese Komponente arbeitet mit allen Kontrollebenen des Programms zusammen, in dem sie bei der Darstellung der Inhalte stufenweise auf die Kontrollebenen zurückgreift.

Diese Ebenen sind:

- 1- Bei den Datenbankabfragen wird kontrolliert, ob die Datenbankverbindung funktionsfähig ist.
- 2- Bei der Darstellung der geschützten Inhalte wird kontrolliert, ob der Benutzer angemeldet ist.
- 3- Bei der Darstellung der Wettscheine und der Annahme der gespielten Wettscheine wird mit Hilfe der Zeitkontrolle überprüft, ob der Totalisator für die Wettannahme noch offen ist.

WAP2-Schnittstelle

Die WAP2-Schnittstelle ist verantwortlich für die Kommunikation des Programms mit dem Benutzer eines mobilen Internetzugangs. Diese Komponente erfüllt folgende Aufgaben:

1. Darstellung, der von der Datenaufbereitungskomponente zur Verfügung gestellten Daten, auf beliebige WAP2-kompatible mobile Internetzugänge.
2. Empfang und Weiterleitung der vom Benutzer angegebenen Daten an die Datenaufbereitungskomponente.

Die Komponente hat zwei statische Teile, die das Skelett einer XHTML-Seite darstellen. Mit dem Skript „head.php“ wird die Standardkopfzeile für eine XHTML-Seite an den Browser übertragen, der die Seite startet, und mit dem Skript „foot.php“ wird die Seite abgeschlossen. Der Inhalt, der sich innerhalb dieser Seite

darstellen sollte, wird von der Datenaufbereitungskomponente erstellt. Bei den Skripten, die eine Datenausgabe erstellen, arbeiten diese Komponenten zusammen, deswegen kann man sagen, dass die Trennung eher logisch als programmtechnisch ist.

Zeitkontrolle

Die Zeitkontrolle stellt die innerste Kontrollschicht des Programms dar und ist für die Kommunikation mit dem Kontrollsystem des Totalisators¹⁴ zuständig. Durch diese Verbindung prüft das Programm, ob der Totalisator für die von Benutzer gewählten Rennen noch Wetten akzeptiert. Wenn der Totalisator eine negative Antwort gibt, wird kein Wettschein gezeigt und/oder angenommen. Diese Kontrolle wird durch das Skript „zeitkontrolle.php“ durchgeführt. Dieses Skript bekommt Informationen über Datum, Ort und Nummer des Rennens als Parameter und baut eine Verbindung mit dem Totalisator auf, um diese Daten nachzufragen. Der Totalisator liefert ein String als Antwort, mit Hilfe dieses Strings entscheidet das Programm, ob der Totalisator offen oder geschlossen ist. Diese Komponente wird dreimal während der Durchführung des Programms eingesetzt¹⁵; erstens vor der Darstellung eines Wettscheins, zweitens vor der Annahme des Wettscheins und zuletzt während der Abgabe des Wettscheins.

4.1.2. Überblick der einzelnen PHP-Skripten des Programms

Body.php: Stellt den Rahmen des Inhalts eines XHTML-Dokuments. Ruft einzelne PHP-Skripte nach Wahl des Benutzers auf.

Datenbank.php: Baut die Datenbankverbindung auf, und dient als Kontrollebene für Datenbankabfragen.

Datum.php: Erstellt und zeigt eine Liste der Renntage an einem Datum, der vom Skript „datumform.php“ übergeben wird. Die gezeigten Renntage verweisen auf das Skript „renntag.php“.

Datumform.php: Formular für die Auswahl eines Datums. Dieses Skript wird durch den Verweis „Rennen am...“ aufgerufen.

¹⁴ Es verbindet sich bei dem Prototyp eigentlich nicht mit dem Totalisator, sondern mit dem Webserver der Internet-Wettfirma Trago (www.trago.de)

¹⁵ Diese Kontrollebene ist bei dem Prototyp deaktiviert, da die Beispieldaten von gelaufenen Rennen stammen.

Datumgen.php: Generiert die Datumsinformationen basierend auf der Systemuhr oder auf ein vorgegebenes Datum. Ergebnisse werden im Programm für die Generierung der Verweise „Heutige Rennen“ und „Rennen in einer Woche“ verwendet.

Dreier.php: Funktion zur Wettscheinüberprüfung und Kostenberechnung der Dreierwette. Wird vom Skript „wettecheck.php“ aufgerufen.

Err.php: Kontroll- und Meldfunktion für Fehler.

Foot.php: Skript zum Schließen der XHTML-Seiten. Es wird am Ende jeder gesendeten Seite angefügt. Produziert je nach Seitenname die Verweise, die am Ende einer Seite gezeigt werden.

Formen.php: Führt Datenbankabfragen durch und liefert Ergebnisse über die Formen eines abgefragten Trabers. Wird über das Skript „traber.php“ durch ein Verweis aufgerufen.

Guthaben_check.php: Kontrollfunktion zur Überprüfung der Deckung des Guthabens eines Benutzers von Kosten eines berechneten Wettscheins. Wird von den Skripten „wettschein.php“ und „wetterec.php“ aufgerufen.

Head.php: Startskript für XHTML-Seiten, wird am Anfang jeder Seite gesendet. Der Empfänger erkennt somit den Inhalt als eine browserkompatible XHTML-Seite.

Heute.php: Erstellt und zeigt eine Liste der Renntage am selben Tag. Gezeigte Renntage verweisen auf das Skript „renntag.php“.

Hilfe.php: Standardseite von Hilfeinformationen für die Verwendung des Programms. Der Inhalt sollte vom Betreiber zur Verfügung gestellt werden.

Index.php: Startskript für das Programm. Konfigurationsparameter für PHP-Parser sollen in dieser Datei eingefügt werden. Ansonsten beinhaltet dieses Skript keine Informationen, ruft vorerst nur folgende Skripte auf:

- sitzung.php
- datenbank.php

Wenn die Verbindung mit der Datenbank erfolgt, dann werden folgende Skripte aufgerufen:

- head.php
- body.php
- foot.php

Wenn es Fehler bei der Datenbankverbindung oder beim Sitzungsstart gibt, wird das Skript „err.php“ aufgerufen.

Index2.php: Begrüßungsskript nach einer erfolgreichen Anmeldung und das Startskript für den geschützten Datenbereich. Beinhaltet Informationen über Anrede, Name und Guthaben des Benutzers und zeigt Verweise auf folgende Skripte:

- „heute.php“ (Heutige Rennen)
- „nwoche.php“ (Rennen in einer Woche)
- „datum.php“ (Rennen am ...)

Wird bei einer erfolgreichen Anmeldung vom Skript „body.php“ aufgerufen.

Kontakt.php: Standardseite über Kontaktinformationen zum HVT. Der Inhalt sollte vom Betreiber zur Verfügung gestellt werden.

Kundeninfo.php: Zeigt die Informationen über einen Kunden; Name, Guthaben, gespielte Wettscheine und Status dieser Wettscheine. Bietet einen Verweis auf das Skript „password.php“ um das Passwort zu ändern.

Login.php: Formular zur Anmeldung eines Benutzers für den Zugang in den geschützten Bereich.

Logout.php: Skript zur Abmeldung eines angemeldeten Benutzers. Dieses Skript beendet die Sitzung und löscht die erstellten Cookies. Bietet einen Verweis auf das Skript „index.php“.

Neu.php: Standardseite der aktuellen Informationen vom HVT. Der Inhalt sollte vom Betreiber zur Verfügung gestellt werden.

Nwoche.php: Erstellt und zeigt eine Liste der Renntage der nächsten sieben Tage. Gezeigte Renntage verweisen auf das Skript „renntag.php“.

Passwort.php: Wird über das Skript „kundeninfo.php“ durch ein Verweis aufgerufen und beinhaltet ein Formular und eine Funktion zur Änderung des Benutzerpassworts. Das eingegebene Passwort wird auf eine minimale Länge kontrolliert. Wenn es

längenmäßig passend ist, wird eine MD5-Signatur von diesem Passwort berechnet und es wird in der Datenbank gespeichert.

Platz.php: Funktion zur Wettscheinüberprüfung und Kostenberechnung der Platzwette. Wird vom Skript „wettecheck.php“ aufgerufen.

Renntag.php: Wird durch die Verweise der Ausgaben von den Skripten „heute.php“, „nwoche.php“ oder „datum.php“ aufgerufen. Erstellt eine Liste der einzelnen Rennen an einem Renntag, die auf das Skript „starter.php“ für Starterinformationen verweist.

Sieg.php: Funktion zur Wettscheinüberprüfung und Kostenberechnung der Siegwette. Wird vom Skript „wettecheck.php“ aufgerufen.

Siegplatz.php: Funktion zur Wettscheinüberprüfung und Kostenberechnung der Sieg-Platzwette. Wird vom Skript „wettecheck.php“ aufgerufen.

Sitzung.php: Die Funktion und die Kontrollebene zur Authentifizierung der Benutzer und der Verwaltung von Sitzung. Wird von dem Skript „index.php“ aufgerufen und wird bei jedem Seitenaufruf wiederholt. Die tatsächliche Funktionalität wird erst nach einer erfolgreichen Anmeldung, durch Erstellung eines anonymen Cookies bei dem Klienten, aktiv. Zusätzlich fungiert es bei der Erstellung und Bewertung der Wettscheine als Datenträger zwischen einzelnen Seiten mit Hilfe von Sitzungsvariablen.

Starter.php: Wird durch die Verweise der Ausgaben von dem Skript „renntag.php“ aufgerufen. Stellt eine Liste der einzelnen Starter mit Startnummer, Name des Trabers und Name des Fahrers bzw. Reiters für die gewählten Rennen dar, die auf das Skript „traber.php“ für Traberinformationen verweisen. Zusätzlich wird ein Button zur Abgabe einer Wette dargestellt, der auf das Skript „wettschein.php“ verweist.

Traber.php: Wird durch die Verweise der Ausgaben von dem Skript „starter.php“ aufgerufen. Sucht nach den Informationen des ausgewählten Trabers und deren Eltern in der Datenbank und stellt diese Informationen in Form eines Berichtes dar. Falls weitere Informationen über die Eltern gefunden werden, werden Verweise für diese erstellt. Zusätzlich dazu wird ein Verweis auf das Skript „formen.php“ erstellt, der nach den Formen des Trabers sucht.

Tsuche.php: Die Suchfunktion nach Pferde durch ein Teil des Namens oder Lebensnummer im nichtgeschützten Bereich. Stellt die Informationen (ohne Formen) des Trabers dar.

Wettecheck.php: Kontrolliert die vom Skript „wettschein.php“ gelieferten Wettscheininformationen auf Plausibilität und Gültigkeit. Ruft je nach Art der Wette eine der Skripten „dreier.php“, „platz.php“, „sieg.php“, „siegplatz.php“ oder „zweier.php“ auf, um den wettartentsprechenden Kontrollen durchzuführen und die Wettkosten zu berechnen. In folgenden Situationen wird eine Fehlermeldung generiert:

- Der gespielte Wettschein ist ungültig, fehlerhaft gespielt.
- Das Guthaben der Benutzer deckt die Wettscheinkosten nicht.
- Der Totalisator meldet die Wettannahme als beendet, oder antwortet nicht (Verbindungsfehler).

Um die Kontrolle bei dem Totalisator durchzuführen wird das Skript „zeitkontrolle.php“ aufgerufen. Für die Kontrolle der Benutzerguthaben wird das Skript „guthaben_check.php“ aufgerufen. Wenn der gespielte Wettschein diese Kontrollebenen durchläuft, wird die gespielte Wette in Textform dargestellt und es wird durch ein Verweis ermöglicht, die Wette endgültig abzugeben. Dieser Verweis leitet alle nötigen Informationen zur Wettabgabe an das Skript „wetterec.php“ weiter.

Wetterec.php: Dieses wird vom Skript „wettecheck.php“ nach der Berechnung der Wettscheinkosten und der Gültigkeitskontrolle der Wette, sowie der Zustimmung des Benutzers zur Abgabe der Wette, aufgerufen. Es wird ein zweites Mal kontrolliert, ob der Benutzer über genügend Guthaben verfügt und ob der Totalisator Wettscheine für dieses Rennen noch akzeptiert. Nach diesen Kontrollen wird der Wettschein in der Datenbank gespeichert und die Kosten der Wette werden vom Guthaben des Benutzers abgezogen. Der Status des Wettscheins wird auf „unbekannt“ gesetzt. Diese Zustandseinstufung ist für ein produktives System gedacht worden, wobei der Totalisator den Wettschein auch nach der Abgabe ablehnen kann. Bei einem produktiven Einsatz des Programms sollte eine weitere Kommunikationsebene den Wettschein an den Totalisator liefern und dementsprechend eine Bestätigung für die Gültigkeit der Wette seitens des Totalisators bekommen. Der Zustand des Wettscheins sollte erst danach in „gültig“ geändert werden.

Wettschein.php: Erstellt ein Formular in Form eines Wettscheins für die vom Skript „starter.php“ angegebenen Rennen. Vor der Erstellung des Formulars wird durch das Aufrufen des Skriptes „zeitkontrolle.php“ überprüft, ob der Totalisator offen für die angegebenen Rennen ist. Das Formular besteht aus vier Spalten, die für erster, zweiter, dritter Platz und für Kombinationswette stehen. Die Anzahl der Zeilen variiert je nach Anzahl der Starter. Die Starternummern werden für jedes Pferd einzeln abgerufen, d.h. wenn ein Pferd vor dem Rennen aussteigt, dann wird seine Starternummer nicht auf dem Wettschein gezeigt. Dadurch kann es vorkommen, dass die Starternummern nicht aufeinander folgen. Der Benutzer kann in diesem Formular folgendes wählen:

- Die Art der Wette (Sieg, Platz, Sieg/Platz, Zweier, Dreier),
- Den Einsatz (ab 0,50 € bis 500 €)
- Die einzelnen Pferde und ggf. die Platzierungen der Pferde

Das Formular ist aufgrund der Begrenzungen in den WAP2-Protokollen nicht Interaktiv, d.h. es kann während der Eingabe die Fehler nicht feststellen. Erst nach der Abgabe des Wettscheins werden die Angaben durch das Skript „wettecheck.php“ überprüft.

Zweier.php: Funktion zur Wettscheinüberprüfung und Kostenberechnung der Zweierwette. Wird vom Skript „wettecheck.php“ aufgerufen.

Zeitkontrolle.php: Diese Funktion baut eine Verbindung mit dem Webserver der Internet-Wettfirma Trago auf und kontrolliert, ob der Totalisator noch offen für die gewählten Rennen ist. Es wird von den Skripten „wettschein.php“, „wettecheck.php“ und „wetterec.php“ aufgerufen. Es bekommt Informationen über Datum, Ort und Nummer des angegebenen Rennens als Parameter. Mit diesen Daten wird ein String erstellt, der durch eine HTTP-Verbindung an ein Javascript auf dem Trago-Server gesendet wird. Als Antwort wird eine Ausgabe als String zurückgelesen, wobei ein Teil dieses Strings den Zustand des Totalisators enthält. Somit kann das Programm entscheiden, ob der Totalisator offen oder geschlossen ist. Dieses Skript ist in dem Prototyp deaktiviert, da die Beispieldaten von bereits gelaufenen Rennen stammen.

4.1.3. Funktionsweise

Das Programm empfängt den Benutzer durch das Skript „index.php“, das erst eine Datenbankverbindung aufbaut und dann eine Sitzung startet. Wenn es bei diesen Schritten einen Fehler gibt, wird eine Fehlermeldung erzeugt. Wenn die Datenbankverbindung und der Sitzungsstart erfolgreich waren, werden die Skripten „head.php“, „body.php“ und „foot.php“ aufgerufen.

- „head.php“ sendet die Standard XHTML Seiteninformationen und startet somit eine XHTML-Seite.
- „body.php“ ruft immer durch das Programm gewählte Inhaltsseiten auf. Wenn es keine Auswahl gibt, wird das Skript „index2.php“ aufgerufen, die je nach Anmeldungsstatus der Benutzer folgende Verweise zeigt:
 - Nicht angemeldet:
 - Aktuelles (neu.php)
 - Anmelden (login.php)
 - Angemeldet:
 - Heutige Rennen (heute.php)
 - Rennen in einer Woche (nwoche.php)
 - Rennen am ... (datum.php)
- „foot.php“ sendet je nach Anmeldestatus der Benutzer folgende Verweise und beendet die Seite:
 - Nicht angemeldet:
 - Anmelden (login.php)
 - Suche nach Traber (tsuche.php)
 - Hilfe (hilfe.php)
 - Kontakt (kontakt.php)
 - Angemeldet:
 - Suche nach Traber (tsuche.php)
 - Hilfe (hilfe.php)
 - Kontakt (kontakt.php)
 - Kundeninformationen (kundeninfo.php)
 - Abmelden (logout.php)

Somit kann ein nicht angemeldeter Benutzer folgende Seiten anwählen:

- Die statischen Seiten über Kontakt und Hilfe

- Über „Suche nach Traber“ zu einer Suchmaske, mit der man nach Informationen über Pferde suchen kann
- Über „Anmelden“ auf das Formular zum Einloggen.

Nach der Anmeldung hat der Benutzer die Möglichkeit, durch den Verweis „Kundeninfo“ seine persönlichen Daten anzusehen und darunter durch den Verweis „Passwort ändern“ sein Passwort zu ändern.

Ein angemeldeter Benutzer kann zu Renninformationen drei Verweise folgen, diese sind:

- Heutige Rennen
- Rennen in einer Woche
- Rennen am ...

Der erste Verweis zeigt eine Liste der Renntage am selben Tag, die zweite Verweis erstellt eine Liste der Renntage an den folgenden sieben Tagen und durch den dritten Verweis gelangt man an eine Eingabemaske, in der ein Datum gewählt werden kann, um Renntage an diesem Datum zu sehen. Alle drei Skripten erstellen eine Liste der Renntage in einem Zeitraum, wobei jedes Element dieser Liste auf das Skript „renntag.php“ verweist. Wenn der Benutzer eine der Verweise auswählt, startet das Skript „renntag.php“, welches die einzelnen Rennen an dem gewählten Renntag zeigt. Durch Wahl einer dieser Rennen wird das Skript „starter.php“ aufgerufen, dessen Aufgabe darin besteht, die an dem gewählten Rennen teilnehmenden Traber aufzulisten. An diesem Punkt kann der Benutzer in zwei Richtungen navigieren:

Zum einen kann man auf einen der aufgelisteten Traber klicken und dadurch das Skript „traber.php“ aufrufen, das die Informationen über den gewählten Traber darstellt. Darüber hinaus kann man noch das Skript „formen.php“ aufrufen, das die Formen des Trabers zeigt.

Zum anderen kann man auf das Button „Wetten!“ klicken und dadurch das Skript „wettschein.php“ aufrufen, welches ein Wettscheinformular darstellt. Mit Aufrufen dieses Skripts wird gleichzeitig¹⁶ das Skript „zeitkontrolle.php“ aufgerufen, das den Status des Totalisators kontrolliert. Dieses Skript wird bei jedem Schritt bis zur endgültigen Abgabe des Wettscheins aufgerufen. Auf der vom Skript „wettschein.php“ dargestellten Seite kann man die Art der Wette, den Einsatz und

¹⁶ Mit dem Wort „gleichzeitig“ ist nicht ein Parallelbetrieb gemeint.

den zu wettenden Traber, ggf. Platzierungen wählen. Mit Klicken des Buttons „Wetten!“ wird die Kontrollfunktion zum Wetteinsatz aufgerufen, in dem das Skript „wettecheck.php“ mit Hilfe der den wettartentsprechenden Skripten (sieg.php, platz.php, siegplatz.php, zweier.php oder dreier.php) die Wettabgabe auf Gültigkeit überprüft und die Kosten des Wettscheins berechnet. Diese Kosten werden mit Hilfe des Skripts „guthaben_check.php“ mit dem Guthaben der Benutzer verglichen, um festzustellen, ob der Benutzer genügend Deckung für die Abgabe des Wettscheins hat. Nach diesen Kontrollen wird der Benutzer über die Kosten und den Inhalt seines Wettscheins informiert. Durch das abschließende Klicken auf den Button „Wetten!“ wird die Wette abgegeben und der Benutzer erhält eine Bestätigung. Danach kann der Benutzer wieder auf die Startseite zurückkehren oder einem der Verweise, die von dem Skript „foot.php“ dargestellt werden, folgen.

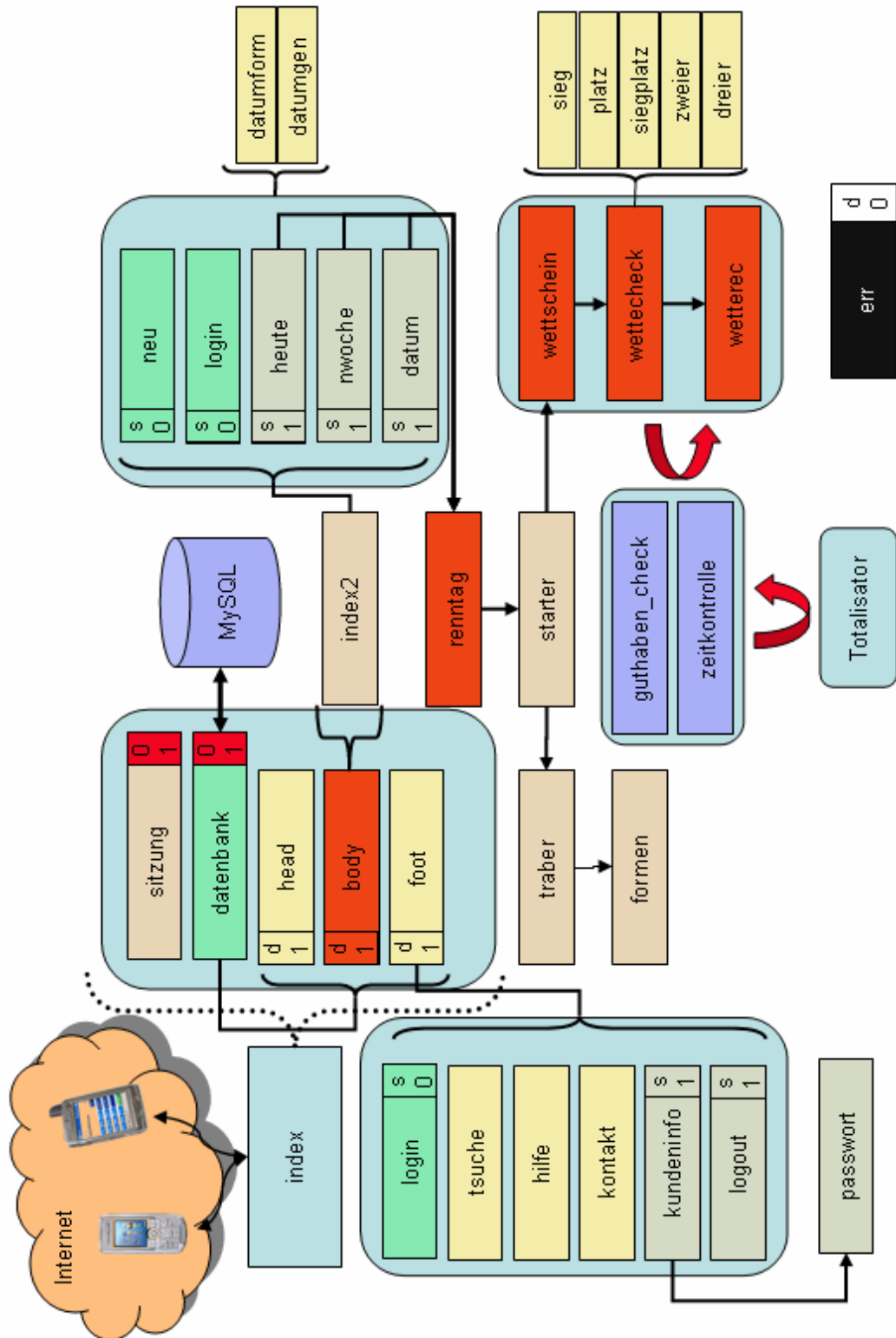


Abbildung 14: Flussdiagramm des Programms

4.1.4. Installation

Das Programm wurde so konzipiert, dass es auf einem PHP betreibenden Webserver installiert werden kann. Die Skripten beinhalten keine festen Verlinkungen, damit das Programm über einen relativen Pfad laufen kann. Unter einem LAMP-System sollte das Programm in folgender Weise installiert werden:

Einstellungen des Apache HTTP-Servers

Auf einem Apache Server, der zusammen mit dem PHP-Parser installiert ist, ist die Ausführung von PHP-Dateien voreingestellt. Bei einigen Linux Distributionen kann es vorkommen, dass der .php Datentyp in der Apache-Konfiguration nicht registriert ist. Dieses kann in der Apache Konfigurationsdatei folgendermaßen geändert werden:

- 1- Die Konfigurationsdatei „httpd.conf“ befindet sich bei den meisten Distributionen unter dem Verzeichnis „/etc/httpd/“ bei Apache 1.x Versionen oder unter dem „/etc/apache2/“ bei Apache 2.x Versionen.
- 2- In dieser Datei folgenden Ausdruck finden:

```
DirectoryIndex index.html index.htm  
Und wie folgt ändern:
```

```
DirectoryIndex index.html index.php index.htm  
Die Datei speichern und Apache neu starten.
```

Falls das Programm als ein Unterverzeichnis innerhalb der Homepage wie zum Beispiel „http://www.muster.de/handywette“ installiert werden soll, dann sollte man unter dem Stammverzeichnis des Webservers das Verzeichnis „handywette“ erstellen und die Programmdateien unter dieses Verzeichnis kopieren. Um sicherzustellen, dass der Webserver Zugriffsrechte für das Verzeichnis und die Daten hat, sollten die Datenrechte neu vergeben werden:

Angenommen dass die Dateien sich unter „/www/htdocs/handywette“ befinden und Apache mit dem Benutzernamen „httpd“ läuft;

```
chown -R httpd.root /www/htdocs/handywette  
chmod -R 644 /www/htdocs/handywette
```

Wenn man das Programm auf einem Virtuellen Host installieren will, wie zum Beispiel „http://handywette.muster.de/“ sollte man den gleichen Vorgang wie oben durchführen und dann die Virtualhost-Einstellungen bei der Apache

Konfigurationsdatei ändern. Dieses wird bei den Apache 1.x Versionen in der „httpd.conf“, ab der Apache Version 2.0 in der „vhost.conf“ Datei (meistens unter „/etc/apache2/vhosts.d/“) durchgeführt. Man sollte folgende Zeilen in der Datei hinzufügen:

Angenommen dass die IP Adresse des Servers „192.168.1.1“, HTTP Port „80“ ist und die Dateien sich unter „/www/htdocs/handywette“ befinden;

```
<VirtualHost 192.168.1.1:80>
DocumentRoot /www/htdocs/handywette
ServerName handywette.muster.de
</VirtualHost>
```

Alle Änderungen zwischen den <VirtualHost> ... </VirtualHost> Markierungen wirken nur auf das Verzeichnis „/www/htdocs/handywette“. Dadurch kann man viele zusätzliche Einstellungen nur für diesen Virtual Host vornehmen.

Konfiguration des Programms

In dem Programm soll nur das Skript „datenbank.php“ für Datenbankverbindung umkonfiguriert werden. In der Zeile „\$dbcnx = @mysql_connect ("host", "benutzername", "passwort");“ muss die IP-Adresse oder Domainname des MySQL Servers anstatt „host“ eingegeben werden. Die Benutzername und Passwort für Zugriff auf den MySQL Server sollen ebenfalls mit dem tatsächlichen Benutzername und Passwort ersetzt werden.

Mit diesen Konfigurationsumstellungen wird die Installation und Konfiguration des Programms erfolgreich abgeschlossen.

4.1.5. Sicherheit

Die Sicherheit des Programms kann in drei Segmente unterteilt werden. Das sind, Sicherheit auf der Benutzerebene, Sicherheit bei der Datenübertragung und Sicherheit im Serverbereich. Der Umfang der Sicherheitsrisiken und der Wert der zu schützenden Informationen sind in jedem Segment unterschiedlich. Deswegen sind auch die Sicherheitsmaßnahmen, die man ergreifen kann, sehr unterschiedlich. Man kann niemals von totaler Sicherheit ausgehen, aber man kann die zu schützenden Informationen bis zu einer Stufe sichern, wobei eine Attacke, die die Sicherheitsmaßnahmen übergehen könnte, mehr als den Wert der auszuspionierenden Informationen kostet. Weiterhin sollte beachtet werden, dass der negative Effekt der Sicherheitsmaßnahmen auf die Bedienungsfreundlichkeit

sich im angemessenen Verhältnis zum Wert der zu schützenden Information befindet. Man kann die Sicherheitseffizienz der einzelnen Segmente unter dieser Sichtweise betrachten.

Sicherheit auf der Benutzerebene

Das Programm funktioniert mit einem Benutzername-Passwort-Paar, das der Benutzer bei der Anmeldung über die Tastatur vom Handy eintippen muss. Die Mindestlänge des Passworts ist kurz gehalten (3 alphanumerische Zeichen) um die Eingabe zu erleichtern. Um die Sicherheitseffizienz zu erhöhen, könnte man als erstes die Mindestlänge des Passworts erhöhen. Beim Einloggen könnte man ab einer bestimmten Zahl von Fehlversuchen den Eingang für einen Zeitraum sperren, oder vom Benutzer eine andere Information, wie z.B. Antwort auf eine Geheimfrage, verlangen.

Es gibt viele Methoden, um die Benutzername-Passwort-Paar-Authentifizierung sicherer zu machen. Eine Methode, die hauptsächlich bei dem Telefon-Banking eingesetzt wird, könnte auch verwendet werden. Hier wird ein ganzer Satz als Passwort benutzt, bei der Anmeldung wird ein zufälliges Wort dieses Satzes nachgefragt. Der Einsatz von mehrstufigen Passwörtern, bei dem der Benutzer für einen Wetteinsatz, der mehr als eine vorbestimmte Menge kostet, nach einem zweiten (und auch einem längeren) Passwort gefragt wird, könnte ebenfalls eine Alternative darstellen. Diese Methode erschwert es den Attackierenden, das Guthaben schnell einzusetzen. Weitere Möglichkeiten, wie z.B. Transaktionsnummern für jede Wettabgabe nachzufragen, Überprüfung der Telefon-ID oder automatisierter Rückruf zur Bestätigung eines Wettscheins, sind ebenfalls durchführbar, verkomplizieren aber die Benutzung, und könnten somit die Benutzer abschrecken.

Sicherheit bei der Datenübertragung

Die Datenübertragung läuft unter Kontrolle des Telefonanbieters und kann von keinem der Beteiligten beeinflusst werden. WAP2-Protokoll bietet zurzeit keine großen Möglichkeiten der Verschlüsselung des Datentransfers. Viele Verschlüsselungsmethoden für die dritte Generation der Mobilien Netzwerke sind zurzeit in der Entwicklungsphase. Ein Problem dabei ist, dass die Prozessorkapazitäten der auf dem Markt befindlichen Handys nicht ausreichen, um eine zufrieden stellende Verschlüsselung mit gewöhnlichen Algorithmen

durchzuführen. Man kann erwarten, dass diese Begrenzungen mit der Zeit durch weitere technische Entwicklungen nicht mehr von Bedeutung sein werden und die Datenübertragungen auch effizient verschlüsselt werden können. Bis dahin ist es beruhigend zu wissen, dass die technische Ausstattung, um eine GSM-Transaktion abzuhören, mit einem hohen Kostenaufwand verbunden ist.

Sicherheit im Serverbereich

Der Serverbereich ist das wichtigste Segment im gesamten Sicherheitskonzept. Wenn ein Attackierender das Benutzernamen-Passwort-Paar eines Benutzers hat, kann er damit Schaden anrichten, der sich in Grenzen hält. Aber wenn der Attackierende die Kontrolle des Servers übernimmt, kann er das System zum Absturz bringen, die Daten von Benutzern ändern, Änderungen in den Konfigurationen und in dem Programm vornehmen und im schlimmsten Fall kann er nicht bemerkt werden. Einsatz von Firewall-Systemen, eine ständige Überwachung der Netzwerkaktivitäten und regelmäßige Aktualisierungen der Softwareprodukte sind die hauptsächlichen Maßnahmen, die generell gegen Attacken über das Internet ergriffen werden können. Gegen Attacken, die auf die kommerzielle Struktur des Programms zielen, kann man sich neben guten technischen Sicherheitsmaßnahmen auch mit einem stabilen Kontoführungssystem mit doppelten Einträgen verteidigen. Das System muss nicht nur gegen äußere Angriffe geschützt werden, sondern auch gegen Angriffe von Insidern. Gegen Insider-Attacken kann man nicht viel unternehmen. Es können aber auch hier Vorkehrungen getroffen werden, die diese Attacken gar nicht erst zustande kommen lassen. Es kann geregelt werden, dass keiner der Administratoren alleine große Änderungen am System vornehmen kann und die Personen, die keine Administrationsrechte haben, auch keine Gelegenheit bekommen, das System zu erreichen. Dies wäre die einzige Art der Attacke, die auch physikalisch werden kann. Zur Abwehr könnte man abgestufte Sicherheitsbereiche in den Serverräumen einrichten und Videoüberwachung einsetzen.

Praktische Sicherheitshinweise zum Programm

Das Programm kann in zwei Sicherheitsbereiche aufgeteilt werden. Der erste Bereich ist der Datenbankserver, der auf einem separaten Rechner betrieben werden sollte. Der zweite Bereich ist der Webserver, auf dem das Programm läuft. Der Webserver sollte hinter einer Firewall platziert werden. Alle unnötigen Services und Zugriffsmöglichkeiten durch das Internet und Subnetz an den Server sollen

abgeschaltet sein, wobei der einzige Zugriff über das Internet durch den HTTP-Port erfolgen sollte. Die Verbindung zwischen Datenbankserver und Webserver kann durch eine IPless-Bridge [BSD] aufgebaut werden, damit kein Rechner, außer dem Webserver, den Datenbankserver erreichen kann.

5. Diskussion

Am Anfang dieses Projekts gab es große technische Begrenzungen bei der Datenübertragung von mobilen Internetzugängen, bei denen die Darstellung eines Wettscheins auf einem Handybildschirm kaum vorstellbar war. Bei der Programmierung des Prototyps wurden mehrmals diese technischen Grenzen erreicht. Einige dieser Begrenzungen haben sich durch Entwicklungen in der Datenübertragungstechnik für mobile Internetzugänge aufgelöst und einige wurden durch programmiertechnisches Geschick umgangen. Es gibt aber trotzdem noch folgende Einschränkungen:

- Es ist immer noch keine praktische Umsetzungsmöglichkeit zur Videoübertragung vorhanden, obwohl diese laut Medienberichten seit ein paar Jahren möglich ist.¹⁷
- Die Verschlüsselungstechnik der Datenübertragung von Mobiltelefonen ist immer noch nicht auf einem Niveau, bei dem man sicher sein kann, dass die Datenübertragung nicht entschlüsselt wird.
- Die Datenübertragungsraten sind immer noch zu niedrig im Vergleich zu anderen Internetverbindungen. UMTS hat durch mehrere Verzögerungen bei der Markteinführung das interessierte Publikum verloren und kann sich wahrscheinlich auch in den kommenden Jahren nicht stark durchsetzen und wird ein Nischenprodukt in Großstädten bleiben¹⁸.

Trotz all dieser nicht realisierten Erwartungen an die Technik, wurde in dieser Arbeit das Ziel erreicht: Das als Prototyp konzipierte Programm ist ein Beispiel für die Einsatzmöglichkeit von mobilen Internetzugängen zur effektiven Nutzung der landwirtschaftlichen Datenbestände, in dem es die Funktionalitäten wie Aufnahme und Darstellung dieser Daten demonstriert. Das Programm stellt die Informationen zum Trabrennsport auf einem Mobiltelefon dar und ermöglicht eine fiktive Wettannahme für ein aktuelles Rennen, was ein gutes Beispiel für komplexe Datenaufnahmen gibt.

¹⁷ Die Medien verwechseln ganz häufig was auf den Messen wie IFA oder Cebit vorgestellt wird und was sich bei gegebener Marktsituation realisieren lässt.

¹⁸ Wird der UMTS-Standard sich als eine veraltete Technik noch erweitern können, wohingegen in Japan schon Gigabit-Geschwindigkeiten der 4. Generation Mobiltelefone im Testlauf sind?

5.1. Das Programm

Das Programm ist serverbasiert und es befindet sich auf dem Server, wo es mit Hilfe der Apache HTTP Server (Webserver) und PHP (Parser) betrieben wird.

Das Programm ist in der Lage gleichzeitig mehrere Verbindungen zu beantworten, wobei aufgrund von messtechnischen Schwierigkeiten keine genauen Angaben über die Kapazität des Programms in Verhältnis zu den Hardwarekosten gemacht werden können.

Die am Anfang der Arbeit vorgestellten Ziele zu den Funktionen des Programms sind erreicht worden. Das Programm bietet folgende Funktionalitäten in den vom HVT vorgegebenen Rahmenbedingungen:

- Ein Benutzerauthentifizierungssystem und Sitzungsmanagement zur Bereitstellung von geschützten Transaktionen
- Eine Suchfunktion zur Suche von Pferdeinformationen
- Darstellung der aktuellen Rennveranstaltungen, einzelne Rennen an einem Renntag, sowie die Starter eines Rennens
- Angebot über vertiefte Informationen der Starter eines Rennens (Herkunfts- und Formdaten)
- Darstellung eines Wettscheins mit verschiedenen Wettarten (Siegwette, Platzwette, Sieg/Platzwette, Zweierwette und Dreierwette) zur Abgabe eines Wettscheins
- Annahme eines gespielten Wettscheins
- Mehrschichtige Kontrollfunktionen zur Wettscheinannahme:
 - Kontrolle der Gültigkeit eines Wettscheins
 - Kontrolle der Deckung der Wettscheinkosten
 - Kontrolle über den Stand des Totalisators
- Erweiterbarkeit durch offene Programmkomponenten und Verfügbarkeit des Quellcodes

Der Prototyp ist innerhalb der vom HVT vorgegebenen Rahmenbedingungen funktionsfähig, wobei einige Teile, die bei einem produktiven System benötigt werden, fehlen. Hierbei handelt es sich um folgende Komponenten:

- Eine echte Verbindung zum Totalisator: Stattdessen wird eine Verbindung zum Webserver der Firma Trago (www.trago.de) aufgebaut und nach dem Status der Wettannahme gefragt.
- Ein reales Bezahlungssystem: Für ein produktives System ist ein automatisiertes Bezahlungssystem mit verschiedenen Zahlungsmöglichkeiten (z.B. Lastschrift, Kreditkarten, Paypal) nötig.
- Eine reale Kontoführung: Es wurde ein einfaches Guthabensystem in den Prototyp eingebaut. Das Programm liest die Guthaben der Benutzer aus einem Datenfeld. Nach einer Wettabgabe werden die Kosten des Wertscheins von dem Wert dieses Datenfeldes abgezogen. Für ein produktives System ist ein stabiles und übersichtliches Kontoführungssystem mit doppelten Einträgen angebracht.
- Ein Administrationsinterface zur Kundenbetreuung: Es gibt beim HVT kein Personal für die Kundenbetreuung und es konnte während des Projekts nicht festgestellt werden, welche Anforderungen zur Verwaltung und Betreuung der Kunden beim Betrieb eines produktiven Systems gestellt würden.
- Eine aktuelle Datenbank für Rennveranstaltungen: Das Programm arbeitet mit alten Datensätzen, die zum größten Teil aus dem Jahr 2004 stammen. Dies wird durch eine Umstellung im Skript „datumgen.php“ ermöglicht, wo anstatt des aktuellen Datums ein vorgegebenes Datum eingesetzt wird. Bei einem produktiven System sollte die Datenbank ständig mit aktuellen Renninformationen gespeist werden, der derzeitige Datenbestand des HVTs sowie die Organisation des Datenverkehrs im Trabrennwettgeschäft sind jedoch dazu nicht geeignet.
- Fehlende Informationen: Die Inhaltsseiten des Programms Hilfe, Kontaktinformationen und Aktuelles haben im Prototyp keinen Inhalt.

5.2. Programmiersprachen

Die Auswahl der Programmiersprachen basierte auf früheren Erfahrungen des Verfassers. Zur Programmierung wurde die Skriptsprache PHP verwendet, zusätzlich wurde zur Abfrage der Datenbanken die Datenbanksprache SQL, eingebettet in die PHP-Skripten, eingesetzt.

Die Auswahl der SQL-Sprache stand nicht zur Diskussion, da es keine andere standardisierte Sprache zur Abfrage von Datenbanken gibt. Man verwendet entweder die datenbankeigene Programmiersprache, z.B. PL/SQL bei Oracle und

opfert dadurch die Migrationsmöglichkeiten auf andere Datenbanksysteme oder benutzt SQL und bleibt somit datenbankneutral bei der Programmierung.

Es gibt viele Gründe für die Auswahl von PHP als Hauptprogrammiersprache, die wichtigsten davon sind:

- Plattformunabhängigkeit
- Eingebaute Unterstützung verschiedener Datenbanksysteme
- Kurze Entwicklungsdauer durch kürzeren Quellcode im Endprodukt bei gleicher Funktionalität im Vergleich zu anderen Sprachen

Anstatt PHP hätten Perl [PRL] oder Java [JVA] eingesetzt werden können, da beide vom Funktionsumfang vergleichbar mit PHP sind. Der Quellcode des Endprodukts ist bei den PHP-Skripten übersichtlicher als bei Perl, dieses erleichtert eine weitere Entwicklung durch Drittpersonen. Ein weiterer Vorteil von PHP ist die leichtere Erlernbarkeit im Vergleich zu Perl und Java. Auch bei der Datenbankunterstützung stellt PHP eine bessere Alternative als Perl dar. Java hat sehr hohe Hardware Anforderungen im Vergleich zu PHP. Die JVM (Java Virtual Machine) muss permanent laufen und besetzt dadurch ständig Systemressourcen, auch wenn keine Java-Programme aktiv sind. Dagegen bleibt PHP nicht resident im Arbeitsspeicher und wird nur bei Bedarf vom Webserver aufgerufen. Dies führt zu großen Leistungsunterschieden zu Gunsten von PHP bei vergleichbarer Hardwareumgebung.

In einem produktiven System könnte man bei der Programmierung der Kontrollebenen eine high-level ¹⁹ Programmiersprache, z.B. LISP [GRA04] verwenden und die Kontrollfunktionen als mathematische Algorithmen definieren. Das würde sicherlich die Geschwindigkeit des Programms erhöhen und auch zu dem Schutz der verwendeten Kontrollfunktionen dienen. Dies kann jederzeit umgesetzt werden, da PHP es ermöglicht, die in andere Programmiersprachen erstellten Funktionen aufzurufen.

Der Betrieb des Programms erfolgt serverseitig, dadurch werden keine Teile des Programms an den Klienten (empfangende Rechner oder mobiler Internetzugang) übertragen. Der Benutzer kann im Gegensatz zu Javascript-Programmen keine Teile des Programms herunterladen. Der Benutzer bekommt bei jedem Aufruf des Programms als Ausgabe eine reine Textdatei, die aus XHTML-Elementen besteht

¹⁹ High-level bezeichnet den technischen Abstand der Programmiersprache von dem Maschinencode.

und keine Fragmente des Quellcodes beinhaltet. Somit kann keiner von der Benutzerseite sich über die Funktionsweise des Programms informieren, was gleichzeitig auch als eine gute Sicherheitsmaßnahme dient. Der Server, als Arbeitsumgebung des Programms, kann ständig unter Kontrolle gehalten werden, womit die Funktionalität des Programms gesichert wird. Das ist ein klarer Vorteil gegenüber Javascript, bei dem man sich auf den Funktionsumfang der Klienten verlassen muss. In der Zukunft können die mobilen Klienten eine zuverlässigere Funktionsumgebung für Javascript-Programme anbieten, z.B. durch die Einführung der AJAX-Technologie (Asynchronous Javascript and XML) im mobilen Bereich.

5.3. Datenbanken

Der Prototyp ist vom Konzept her datenbankunabhängig, wobei es in der programmiertechnischen Realität mit der MySQL-Datenbank (MySQL RDBMS) arbeitet. MySQL wurde wegen der einfacheren Bedienung und problemlosen Übertragbarkeit der Datenbanken gewählt, da das eigentliche Ziel der Test des Prototyps unter verschiedenen Serverkonfigurationen war. Es gibt viele Firmen, dessen gesamte Präsenz im Internet auf MySQL-Datenbanken basiert. Andererseits gibt es in der Open-Source-Welt ein fortschrittlicheres Datenbanksystem, das sehr ausgereifte Eigenschaften anbietet und somit auf dieselbe Stufe wie die kommerziellen Datenbanksysteme (Oracle, IBM DB/2) platziert werden kann: *PostgreSQL*. PostgreSQL ist als ein Forschungsprototyp im Jahr 1986 in der Berkeley Universität, Kalifornien entwickelt worden. Es wird heute von der *PostgreSQL Global Development Group* weiterentwickelt. Diese Gruppe besteht aus Privatpersonen und Firmen, die die Entwicklung der PostgreSQL als ein Open Source Programm mit einem global verteilten Modell weiterführen.

Beim Aufbau eines produktiven Systems wäre es sicherlich ein Versuch wert, PostgreSQL anstatt MySQL zu verwenden. Dabei könnten die eingebauten Replikationsfunktionen des PostgreSQL eine bessere Datensicherheit in einem Clusterumfeld gewährleisten.

5.4. Weitere Entwicklungsmöglichkeiten des Programms

Der Prototyp ist von der Konzeption her offen für weitere Entwicklungen. Mit der modularen Programmierstruktur kann man jeder Zeit neue Funktionalitäten hinzufügen ohne den restlichen Funktionsumfang des Programms zu beeinträchtigen.

Das Programm ist direkt übertragbar auf den Galopprennsport der Vollblüter. Als praktische Anwendungen der neueren Funktionalitäten könnte man das Publizieren der Photofinish Bilder, sowie Fotos von einzelnen Pferden ohne großen Aufwand realisieren. Es wäre vorstellbar, den Pferdezüchtern und –besitzern die Möglichkeit zu geben, Fotos und Daten der eigenen Pferde hochzuladen. Mit Hilfe der zukünftigen Hochgeschwindigkeitsverbindungen kann man die aktuellen Videos von den Rennen anbieten.

In der Reitpferdezucht kann man mobile Internetzugänge immer dort effektiv nutzen, wo keine PC-Internetverbindung existiert. Als Beispiele kann man sich vorstellen, dass aktuelle Zuchtinformationen von Elterntieren abgerufen werden sollten

- bei der Vorauswahl von Auktionspferden oder Hengsten
- bei der Vorauswahl von Zuchtstuten
- bei Schauveranstaltungen
- bei Körungen und Zuchtstutenbewertungen
- bei Feldprüfungen der Jungpferde

Im Reitpferdesport wäre die Anwendung darin zu sehen, dass per Mobiltelefon aktuelle Zuchtinformationen über das Turnierpferd selbst bzw. dessen Eltern auf dem Turnier abgerufen werden können. Diese Informationen müssen nicht kostenfrei zur Verfügung gestellt werden, sondern können zur Refinanzierung von Züchtungsaufgaben dienen.

Die Einsatzmöglichkeiten der mobilen Internetzugänge im Pferdesport sind sehr vielseitig. Die Grundvoraussetzung dafür ist der Aufbau einer stabilen und aktuellen Dateninfrastruktur zur Unterstützung dieser Anwendungen.

5.5. Mobile Internetzugänge

Zu Beginn des Projekts waren i-mode Mobiltelefone als Zielplattform vorgesehen. Im Laufe des Projekts gab es starke Veränderungen auf dem Markt und die i-mode Geräte verloren ihre technischen Vorteile gegenüber neueren WAP2-Geräten. Deshalb wurde die Kommunikationsschnittstelle des Programms für die WAP2-Mobiltelefone modifiziert.

Der Prototyp wurde mit Mobiltelefonen der Firmen Siemens und Nokia erfolgreich getestet. Obwohl der Entwicklungsstand des Programms zurzeit die technischen

Grenzen der Mobiltelefone im Markt erreicht hat, könnte man weitere technische Eigenschaften einbauen, die allerdings mit dem Risiko von Kompatibilitätsproblemen verbunden sein könnten.

5.6. Kosten der Programmentwicklung

Das Programm wurde auf einem Linux Betriebssystem, mit der PHP Programmiersprache in Verbindung mit dem MySQL-Datenbanksystem und Apache-Webserver entwickelt. All diese Softwareprodukte sind unter einem der Open Source Lizenzen²⁰ zu bekommen und sind somit kostenlos. Durch die Verwendung der Open Source Programme entstanden keine Kosten für die Software. Da die Hardwareanforderungen des Programms nicht besonders anspruchsvoll sind, wurden hauptsächlich handelsübliche Rechner für den Testbetrieb verwendet. Das Cluster-System der GWDG wurde ebenfalls kostenlos für den Testbetrieb eingesetzt. Die einzigen Kosten bei der Entwicklung des Programms entstanden durch die tatsächliche Programmierarbeit.

5.7. Mögliche Kosten eines produktiven Systems

Ein produktives System kann auch mit einem Linux-Betriebssystem, wie in der Entwicklungsphase des Programms, betrieben werden. Die restlichen Softwareprodukte für den Betrieb des Programms sind auch mit einem der Open Source Lizenzen kostenlos zu verwenden. Derzeit ist es nicht mehr nötig, in eine eigene Serverhardware zu investieren, da man mit den Mietangeboten der Serverhousing-Firmen ein gutes Angebot vorfindet. Die Mietkosten eines dedizierten Linux-Servers inklusive der Internetverbindung beträgt heute 200 € bis 500 € pro Monat, je nach Art des Kundensupports und der automatisierten Backup-Möglichkeiten.

Für den produktiven Betrieb sind Änderungen im Programm notwendig, diese Programmierarbeit ist auch mit Kosten verbunden.

Nennenswerte Kosten werden wahrscheinlich erst durch die Einstellung des Personals zur Kundenbetreuung entstehen, nach dem das Programm mit kommerziellen Komponenten wie Wettannahme in Betrieb genommen wird.

²⁰ Es gibt verschiedene Lizenzmodelle für Open Source Software

5.8. Realisierbarkeit eines produktiven Systems

Für einen kommerziellen Einsatz des Programms gibt es einige technische und infrastrukturelle Voraussetzungen. Die technischen Voraussetzungen sind eng mit der Organisation des Datenverkehrs im Trabrennwettgeschäft verbunden. Das Programm muss mit aktuellsten Daten gespeist werden, besonders wenn es um die Wettannahme geht. Dies ist mit dem heutigen zeitversetzten Datenverkehr zwischen den Rennvereinen und dem HVT schwer zu realisieren.

Unter infrastrukturellen Voraussetzungen versteht man die Investitionen, die getätigt werden müssten, bis der erste Wettschein angenommen werden kann. Ein Buchmachergeschäft ist nicht nur mit einem System zur Wettannahme zu realisieren, sondern es werden weitere unterstützende Strukturen benötigt, z.B. Kundenbetreuung und Kontoführung. Zurzeit wäre es nicht realistisch zu denken, dass die laufenden Kosten und Investitionskosten eines produktiven Systems durch die Wettumsätze über Mobiltelefone gedeckt werden können, trotzdem sollte eigentlich ein Buchmachergeschäft über alle möglichen Kommunikationswege Wettannahmen machen können. Deshalb sollte dieses Programm eher als Teil einer gesamten Palette von Wettabgabemöglichkeiten angesehen werden. Es wäre unter diesen Umständen sinnvoll, das Programm an die zurzeit im Internet aktiven Buchmacher zu lizenzieren. Der Prototyp ist zurzeit das einzig funktionsfähige Programm, das über einem mobilen Internetzugang Wettannahmen durchführen kann, somit hat er derzeit einen hohen Verkaufswert.

5.9. Implementierungsmöglichkeiten des Programms in anderen Bereichen der Landwirtschaft

Das in dieser Arbeit entwickelte Programm nutzt die Daten aus der Traberzucht als ein Beispiel für jegliche Datenbestände aus der Nutztierzucht, wodurch die Übertragbarkeit auf andere Gebiete der landwirtschaftlichen Produktionsprozesse ohne großen Aufwand möglich ist. Die Client-Server-Architektur ermöglicht die Durchführung sehr komplexer Rechenprozesse auf der Serverseite, die auf einem gewöhnlichen PC zu zeitaufwendig wären. Unter diesen komplexen Rechenprozessen könnte es welche geben, die die Einzelpersonen aus Datenschutzgründen nicht durchführen dürften, da für diese Zwecke benötigte Datenbanken sich im Besitz dritter Personen befinden. Beispiele für solche Fälle

sind die Berechnungen (Zuchtwertschätzung, Anpaarungsplanung) bei denen die benötigten Daten der Population nicht vom Landwirt direkt genutzt werden dürfen.

Das Programm kann mit Hilfe von zusätzlichen Ausstattungen (Tieridentifikationshardware, Ultraschallgerät, elektronische Waage) die Erfassung der Produktions- und Leistungsdaten automatisieren und die Speicherung der erfassten Daten in einer zentralen oder lokalen Datenbank ermöglichen.

Ein möglicher Einsatz des Programms in der Rinderzucht könnte zur Anpaarungsplanung dienen. Das Programm könnte zur Suche eines optimal passenden Bullen für eine Färse (bzw. Kuh), dessen lineare Beurteilung über einen mobilen Internetzugang vom Landwirt vor Ort eingegeben wird, eingesetzt werden, wobei die eingegebenen Informationen serverseitig ausgewertet und hieraus eine Liste der optimal passenden Bullen übermittelt wird. Als weitere Dienstleistung könnte das Programm dem Landwirt die Möglichkeit anbieten, das Sperma des ausgewählten Bullen gleich zu bestellen.

Für die Anpaarungsplanung gibt es sicher auch in der Pferdezucht eine Nutzungsmöglichkeit; z.B. bei Körungen oder Hengstvorführungen mit dem Ziel der passenden Auswahl von Hengsten für die eigenen Stuten. Auch hier könnte vor Ort durch Eingabe wichtiger Qualitätskriterien der Stute der passende Hengst als Anpaarungspartner gesucht werden und auf der Hengstvorführung noch einmal in Augenschein genommen werden.

In der Schweinezucht könnte man das Programm in der Kombination mit zusätzlichen Datenerfassungsgeräten zur Eigenleistungsprüfung im Feld von Jungsauen und Endprodukten einsetzen. An der Serverseite des Programms kann anschließend eine Zuchtwertschätzung durchgeführt werden. Der Landwirt bekommt die Ergebnisse in kurzer Zeit. Ein derartiger Einsatz des Programms könnte deutliche Ersparnisse des Arbeits- und Zeitaufwands bringen.

Eine weitere Einsatzmöglichkeit des Programms könnte in kleinen Schweinezuchtbetrieben sein, die noch keinen Sauenplaner verwenden. Diese Betriebe könnten Ihre Wurfmeldedaten über mobile Internetzugänge an die Datenbank des Zuchtverbands mitteilen. Dadurch könnte der Datenfluss auf den Papierwege ersetzt werden, welches wiederum zu einer Arbeitsentlastung für den Landwirt führt.

Das Programm könnte auch zur Überprüfung der Tieridentifikation beim Transport von Tieren eingesetzt werden, für den Fall, dass die notwendigen Identifikationspapiere beim Transport fehlen.

6. Zusammenfassung

Ziel der vorliegenden Arbeit ist es, Einsatzmöglichkeiten von mobilen Internetzugängen zur effektiven Nutzung landwirtschaftlicher Datenbestände zu untersuchen.

Zu diesem Zweck wurde ein Programm entwickelt, das ein Beispiel für die Einsatzmöglichkeit von mobilen Internetzugängen zur effektiven Nutzung der landwirtschaftlichen Datenbestände darstellt, in dem es die Funktionalitäten wie Aufnahme und Darstellung dieser Daten demonstriert. Bei der Entwicklung des Programms wurden die Datenbestände des Trabrennsports des Hauptverbandes für Traber-Zucht und -Rennen e.V. (HVT) als Beispieldaten verwendet. Das Programm stellt die Informationen zum Trabrennsport auf einem Mobiltelefon dar und ermöglicht eine fiktive Wettannahme für ein aktuelles Rennen, was ein gutes Beispiel für komplexe Datenaufnahmen gibt.

Die Programmierung wurde mit dem Programmiersprachen *PHP: Hypertext Preprocessor* (PHP) und *Structured Query Language* (SQL) in Kombination mit der MySQL Datenbank durchgeführt. Mit der datenbank- und plattformunabhängigen objektorientierten Programmierung kann das Programm auf beliebigen Betriebssystemen mit allen SQL-fähigen Datenbanken betrieben werden, wobei die Beispielapplikation in der Arbeit mit einer *MySQL-Datenbank* in Verbindung mit *Apache-Webserver* und *Linux-Betriebssystem* läuft.

Das entwickelte Programm ist fähig die Datenbestände mit Hilfe von SQL-Abfragen, die durch die Interaktion mit dem Benutzer erstellt werden, abzufragen. Die Ergebnisse dieser Abfragen werden für Darstellung an mobilen Internetzugängen, insbesondere WAP2-Mobiltelefone mit kleinen Bildschirm und eingeschränkten Datenübertragungskapazitäten formatiert und nach einer Serie von Fehlerkontrollen an den mobilen Internetzugang weitergeleitet. Dieser Datenverkehr wird durch Kontrollstrukturen zur Überprüfung der Benutzeridentität und Integrität der Datenübertragung überwacht.

Das Programm bietet verschiedene Anwendungen für Benutzer. Es befindet sich eine Suchfunktion zur Abfrage der Herkunftsdaten, sowie der Leistungsdaten der Pferde. Die Trabrennveranstaltungen können nach Datum ausgewählt werden, wobei man die einzelnen Rennen und die Starter an diesen Rennen auch sehen

kann. Zusätzlich besteht das Angebot Wettscheine für aktuelle Rennen auszufüllen und abzugeben, wobei das Programm mit einer eingebauten Kontrollfunktion den Stand des Totalisators für die gewählten Rennen an einem externen Serversystem über das Internet abfragt.

Die Ergebnisse dieser Arbeit verdeutlichen, dass der Einsatz der mobilen Internetzugänge zur Verbesserung der effektiven Nutzung von landwirtschaftlichen Datenbeständen beitragen kann, in dem er den Bedarf an aktuellen Informationen unabhängig vom Ort und von einer ständigen Internetverbindung erfüllt.

Weitere Anwendungen für das Programm sind in vielen Bereichen der Nutztierproduktion möglich, wie z.B. Anpaarungsplanung in der Pferde- und Rinderzucht, automatisierte Datenerfassung in der Schweinezucht, sowie bei landwirtschaftlichen Aktivitäten, bei denen eine konventionelle Internetverbindung nicht verfügbar ist.

7. Summary

The purpose of the present work is to examine the application possibilities of mobile Internet access devices for effective use over agricultural data supplies.

For this purpose, a program was developed which shows an example of the application possibility of mobile Internet access devices to the effective use of the agricultural data supplies in which the functionalities like admission and representation of these data are demonstrated. In the development stage of the program, racing data of trotters supplied by the *Hauptverband für Traber-Zucht und -Rennen e.V. (HVT)* have been used as an example database. The developed program shows information about trotter racing on a mobile phone and allows fictional entries of bets which gives a good example of complicated data admissions.

The programming has been made with the computer language PHP: Hypertext Preprocessor (PHP) and Structured Query Language (SQL) in combination with the MySQL Database. With the object-oriented, platform and database independent programming architecture, the program is able to run on any modern operating system and with all database systems capable of SQL-queries. In our example, application runs with a MySQL database in connection with an Apache web server and a Linux operating system.

The developed program is capable of querying the data supplies with the help of the SQL queries which are created *on-the-fly* through the interaction with the user. The results of these queries are formatted for representation in mobile Internet access devices, in particular WAP2 mobile phones with small screens and limited data transfer capacities, and after a series of error checks, are passed on to the mobile Internet access devices. This data traffic is supervised by control structures regarding the examination of the user identity and integrity of the data transfer.

The program offers different applications for users. It has a built-in search module for the query of the pedigree data, as well as the performance data of the horses. The trotting race events can be selected by date and one can also view the starts and the participating trotters in these races. In addition, there exists an offer to bet for these races. To check the eligibility of approving a bet, the program has a function which questions the state of the totalisator for the selected race on an external server system located in the Internet.

The results of this work make clear that the application of the mobile Internet access devices to the improvement of the effective use of agricultural data supplies can contribute in a way which it fulfils the demand for actual information, independent of place and a constant Internet connection.

There are many possible applications for the program in livestock production, e.g. mating management in horse and cattle breeding, automated data acquisition in the pig breeding, as well as agricultural activities where a conventional Internet connection is not available.

8. Literaturverzeichnis

8.1. Zitierte Publikationen

ALBERTS, C., DOROFEE, A. (2002): Managing Information Security Risks, Addison-Wesley Professional, ISBN: 0321118863

BAUER, M.D. (2002): Building Secure Servers with Linux, O'Reilly Media, ISBN: 0596002173

BHARGAVAN, V., DEMERS, A., SHENKER, S., MACAW (1994): A Media Access Protocol for Wireless LANs. Proceedings of ACM SIGCOMM '94, London, September 1994.

BRADEN, R. (1989): Requirements for Internet Hosts - Communication Layers, Network Working Group, Internet Engineering Task Force (<http://www.ietf.org/rfc/rfc1122.txt>)

BAIOCCHI, A., TODINI, A., VALLETA A. (2004): Why a Multichannel Protocol can boost IEEE 802.11 Performance, proc. MSWiM'04 p. 143-148, October 46, 2004, Venezia, Italy.

BUCKMAN, R. (2000): Who Caused the Dot-Com Crash? The Wall Street Journal, 5. März 2000

CASTELLS, M. (2000): The Rise of the Network Society, Blackwell, ISBN: 0631221409

DENNING, P.J. (1989): The ARPANET after Twenty Years, Research Institute for Advanced Computer Science, NASA Ames Research Center, RIACS Technical Report TR-89.38, 20. September 1989

DKE (2004): Normenreihe DIN EN 61508 (VDE 0803): Funktionale Sicherheit - Sicherheitssysteme (E/E/PES), Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

GAZIS, V., HOUSSOS, N., ALONISTIOTI, N. (2003): Reconfiguration Management in 4G Mobile Environments: Requirements, Process and Architecture, Software Defined Radio Forum Technical Conference

GRAHAM, P. (2004): Hackers & Painters Big Ideas from the Computer Age, O'Reilly Books, ISBN: 0596006624

GARFINKEL, S., SPAFFORD, G., SCHWARTZ, A. (2003): Practical UNIX & Internet Security, O'Reilly Media, 3rd. Print, ISBN: 0596003234

HESSLER, A. (2000): Internet-Marktplatz: Wie sich die Beziehung Kunde - Lieferant ändert, in: Absatzwirtschaft, 43, S. 38-41.

HÖFLICH, J.R. (2001): Das Handy als „persönliches Medium“ Zur Aneignung des Short Message Service (SMS) durch Jugendliche, kommunikation@gesellschaft, Jg. 2, 2001, Beitrag 1

HAMMWÖHNER, R., RITTBERGER, M., SEMAR, W. (2004): Wissen in Aktion : der Primat der Pragmatik als Motto der Konstanzer Informationswissenschaft / Festschrift für Rainer Kuhlen, UVK-Verl.-Ges., ISBN: 3896697048

KEENEY, M., KOWALSKY, E., CAPELLI, D., MOORE, A., SHIMEAL, T., ROGERS, S. (2005): Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors, National Threat Assessment, U.S Secret Service and CERT Coordination Center/SEI

KÖLSCH, M., TURK, M. (2002): Keyboards without keyboards: a survey of virtual keyboards, Workshop on Sensing and Input for Media-centric Systems, Santa Barbara, CA, 20-21. Juni 2002

KANEKO, Y. YAMAGUCHI, M., MATSUYA, H., TSUKADA, T. (1996): Foldable-display systems as a standard platform for multimedia use, IEEE Transactions on Consumer Electronics, Volume 42, Issue 1, 1996

LAURIE, B., LAURIE, P. (2002): Apache, The Definitive Guide., O'Reilly Media, 3rd. Print, ISBN: 0596002033

MITNICK, K., SIMON, L. (2002): The Art of Deception, Wiley Publishing, ISBN: 0471237124

MITNICK, K., SIMON, L. (2005): The Art of Intrusion, Wiley Publishing, ISBN: 0764569597

McWILLIAMS, B. (2003): North Korea's School for Hackers, Wired News, 2. Juni 2003

NOLL, G.R. (1994): The Economics of Information: A User's Guide, The Knowledge Economy / The Nature of Information in the 21st Century, Annual Review of the Institute for Information Studies, 1993-1994, p. 25-52, ISBN: 0898431476

ROBERT FRANCES GROUP (2005): TCO for Application Servers: Comparing Linux with Windows and Solaris (<http://www.ibm.com/linux>)

SCHNEIER, B. (1999): Attack Trees, Dr. Dobb's Journal, Dezember 1999

SCHNEIER, B. (2000): Secrets and Lies, Copernicus Books, ISBN: 0387026207

SCHNEIER, B. (2000): The Process of Security, Information Security Magazine, April 2000

SCHNEIER, B. (2003): Beyond Fear, Copernicus Books, ISBN: 0387026207

SCHNEIER, B. (2005): Schneier on Security, Mitigating Identity Theft, <http://www.schneier.com>, 15. April 2005

SCHNEIER, B. (2005): Schneier on Security, Insider Threats, <http://www.schneier.com>, 18. Mai 2005

TUROWSKI, K., POUSTTCHI, K. (2004): Mobile Commerce Grundlagen und Techniken, Springer Verlag, ISBN: 2540005358

DIREKTORIUM und HVT (2004): Vorschriften für den Wettbewerb, Direktorium für Vollblutzucht und Rennen e.V., Hauptverband für Traber-Zucht und -Rennen e.V., 1. Oktober 2004

THE INTERNET SOCIETY (1999): RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1, <http://www.w3.org>

WAGNER, M. (2004): Will Trade Passwords For Chocolate, Security Pipeline (<http://www.securitypipeline.com>), 19. April 2004

8.2. Zitierte Internetverweise

- [AFO] <http://www.apachefriends.org>
- [ANK] <http://www.ankord.com>
- [APC] <http://httpd.apache.org>
- [BFS] <http://www.bfs.de>
- [BNA] <http://www.bundesnetzagentur.de>
- [BSD] <http://www.openbsd.org/faq/pf>
- [GPL] <http://www.gnu.de/gpl-ger.html>
- [HVT] <http://www.hvt.de>
- [ITU] <http://www.itu.int/ITU-D/ict/statistics>
- [JVA] <http://java.sun.com>
- [MAB] <http://www.mysql.com>
- [NAV] <http://www.navicat.com>
- [NCC] <http://www.netcraft.com>
- [NTT] <http://www.nttdocomo.com>
- [PHP] <http://de.php.net/manual/de>
- [PRL] <http://www.perl.org>
- [WIK] <http://www.wikipedia.org>

9. Anhang

9.1. Abbildungen



Abbildung 15: Startseite des Programms



Abbildung 16: Seite zur Suche nach Traberinformationen



Abbildung 17: Seite zur Anmeldung



Abbildung 19: Darstellung der Rennveranstaltungen an einem Datum



Abbildung 18: Begrüßungsseite nach erfolgreicher Anmeldung

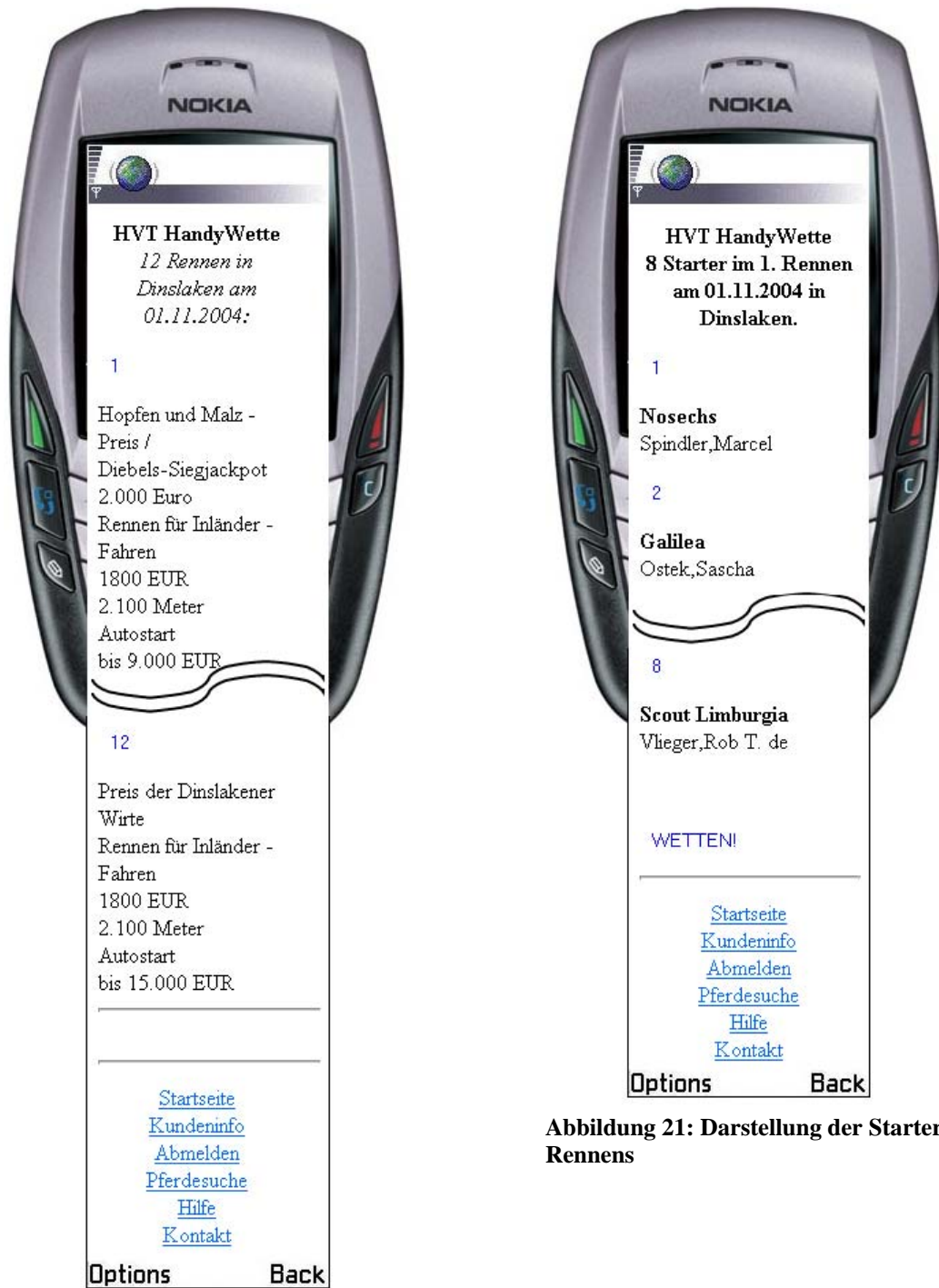


Abbildung 21: Darstellung der Starter eines Rennens

Abbildung 20: Darstellungen der Rennen in einem Renntag an einem Rennbahn



Abbildung 22: Darstellung des Wettscheins für ein Rennen



Abbildung 23: Ein ausgefüllter Wettschein



Abbildung 24: Darstellung eines zu spielenden Wettscheins, zur Kontrolle für den Benutzer



Abbildung 25: Darstellung des Ergebnisses einer erfolgreichen Wettannahme



Abbildung 26: Darstellung der Informationen eines Trabers



Abbildung 27: Darstellung der Formdaten eines Trabers

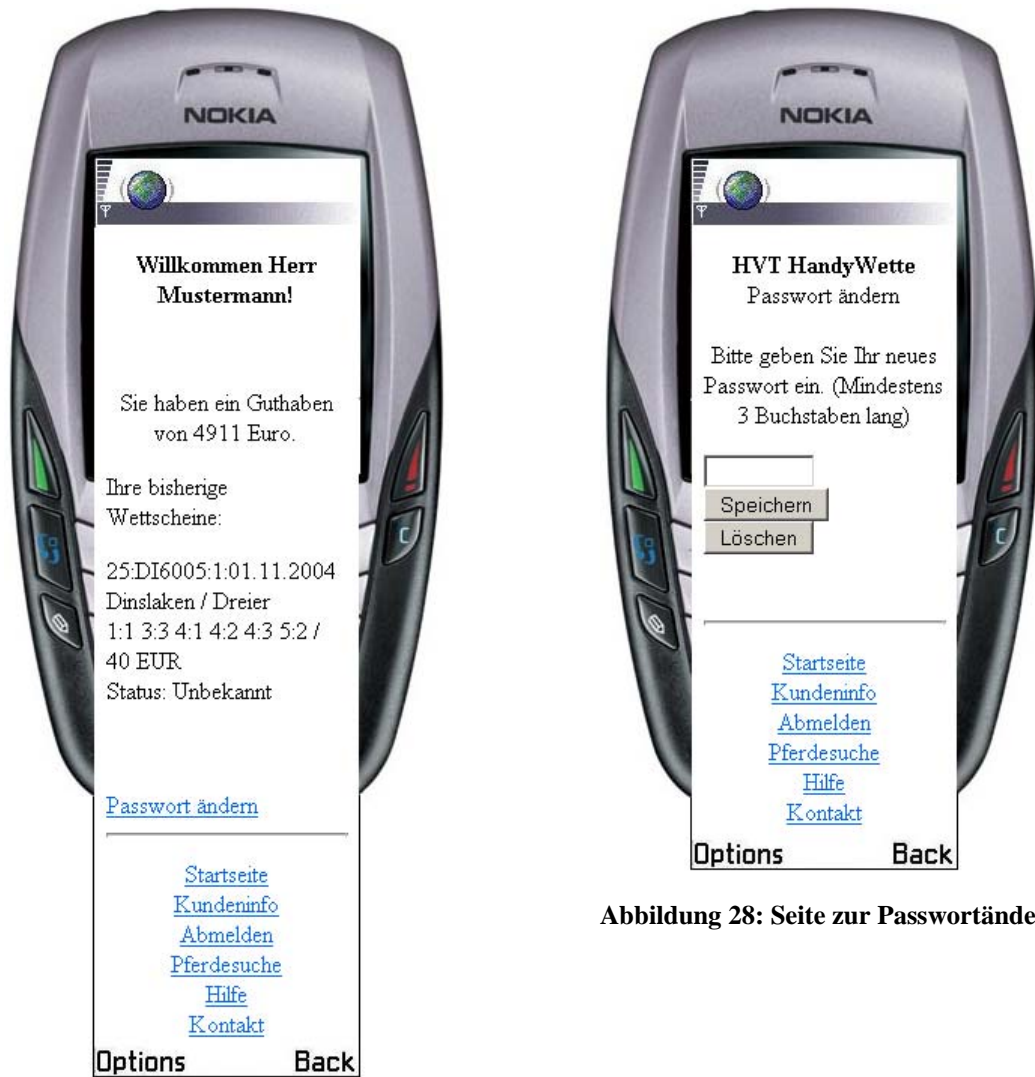


Abbildung 28: Seite zur Passwortänderung

Abbildung 29: Seite zur Darstellung der Kundeninformationen

Danksagung

Mein besonderer Dank richtet sich an Herrn Prof. Dr. Erich Bruns, für die Überlassung des Dissertationsthemas und für die ununterbrochene Unterstützung meiner Ambitionen im Bereichen der Agrarinformatik und Informationstechnologie. Herr Bruns war für mich nicht nur der ideale Betreuer, sondern auch ein Vorbild als Forscher und Mensch.

Für die Übernahme des Korreferats möchte ich mich an Herrn Prof. Dr. W. Lücke bedanken. Herrn Dr. D. von Hörsten bedanke ich für die freundliche Mitbetreuung meiner Arbeit.

Dem Hauptverband für Traber-Zucht und -Rennen e. V. (HVT) möchte ich in Personen von Herrn Dr. Willms, Herrn Dr. Peters und Herrn Höfinghoff danken. Der HVT hat die vorliegende Arbeit durch die Bereitstellung des nötigen Wissens sowie die finanzielle Förderung ermöglicht.

Ohne die Korrekturen und Kritik meines Kollegen und guten Freundes Herrn Buchhard Möllers würde diese Arbeit schwer zu verstehen sein, wofür ich ihm sehr dankbar bin.

Bei meiner Kollegin und Büronachbarin Frau Mabinti Busche möchte ich mich für ein nettes Beisammensein bedanken, ohne sie wären diese zwei Jahren ziemlich langweilig gewesen.

Ohne ständige Unterstützung, Vertrauen und Liebe meiner Familie wäre Erfolg ein Fremdwort für mich. Daher möchte ich mich bei meinem Vater Muhammet, meiner Mutter Fatma Tülin und meinen Schwestern Gonca İrem und Meltem Hilâl für Alles bedanken und widme diese Arbeit ihnen. Ich liebe euch.

Auch wenn ich während meiner Promotion nicht viel mit Tieren zu tun gehabt habe, gibt es ein Paar Freunde aus der Tierwelt, dessen Namen hier erwähnt werden sollten. Der Irishterrier Basco hat die Neugier eines Forschers. Der Wellensittich Kısmet kann über fünfzig Wörter, aber weiß auch wann er schweigen soll. Der Perserkater Mestan ist das charismatischste Tier, dem ich je begegnet bin. Die Westfälische Stute De Bona ist bisher das einzige Pferd, das ich ohne Angst reiten konnte.

Hanefi Yağmur Akgün
Göttingen, Dezember 2005

... istemek ve başarmak; müthiş bir ikili ...

Lebenslauf

Name	Hanefi Yağmur Akgün
Geburtstag und –ort	08.08.1978 in Ankara
Staatsangehörigkeit	türkisch
Familienstand	ledig
Eltern	Muhammet Akgün, Agraringenieur Fatma Tülin Akgün (geb. Yozgat), Lehrerin
Schule	
1984 – 1989	Grundschule Aydinlikevler, Ankara
1989 – 1996	Ankara Anadolu Lisesi (Gymnasium) Abteilung deutsche Sprache
Studium	
1996 – 2000	B. Sc. Studium der Agrarwissenschaften an der Universität Ankara, Fakultät für Agrarwissenschaften, Studienrichtung Tierproduktion
2000 – 2003	M. Sc. Studium der Agrarwissenschaften an der Georg-August-Universität Göttingen, Fakultät für Agrarwissenschaften, Studienrichtung Tierproduktion
Berufstätigkeit	
2003 – 2005	Anfertigung der vorliegenden Dissertation am Institut für Tierzucht und Haustiergenetik der Georg-August-Universität Göttingen