

Projektdokumentation

Projekt im 7. Semester, Angewandte Informatik

Projektthema:

Einsatz von Informationstechnik in Gastronomiebetrieben
Entwicklung der Applikation E-Bar



Projektgruppe:

Jens Klinker
Daniel Prusseit
Jens Rekow
Stephan Ziep

Betreuende Lehrkraft:

Prof. Jürgen Sieck

Projektzeitraum:

Oktober 2003 bis Februar 2004

Inhaltsverzeichnis:

Inhaltsverzeichnis:	2
1. Einführung, Thema	3
1.1. Über dieses Projekt	3
2. Ideen und Techniken	3
2. Ideen und Techniken	4
2.1. Istanalyse	4
2.2. Anforderungen an eine IT-Lösung	6
2.3. Ideen zu eingesetzter Hardware	7
2.3.1. Ethernet	7
2.3.2. Wireless LAN	8
2.3.3. Bluetooth	9
2.3.4. PDA	10
2.4. Mögliche Techniken	13
2.4.1. HTML	13
2.4.2. PHP	14
2.4.6. Java	17
2.4.3. Servlets	18
2.4.4. JSP	19
2.4.5. Javascript	20
2.4.7. MySQL	20
2.5. Zielstellung des Projekts	23
2.6. Ziel- und Kundengruppen	24
2.7. geplante Funktionalitäten	25
2.8. Ausbaustufen	26
2.9. Aufgabenverteilung	26
3. Design	28
3.1. Use Cases (Diagramm von Edward Datel)	28
3.2. beteiligte Akteure	28
3.4. n-Tier	28
3.5. Systemarchitektur	29
3.6. Datenbankstruktur	31
3.7. Anforderungen an die Oberfläche	33
3.8. Abbildung der Geschäftsprozesse	34
3.9. Objektorientierung in PHP?	36
3.10. Klassen der Anwendungslogik	36
4. Implementierung	38
4.1. Auswahl der Techniken	38
4.2. Hardware und Installation	38
4.4. Implementierung der Oberfläche	39
4.4.1. Beschränkungen der Endgeräte	39
4.4.2. Umsetzung (Generatoren etc.)	39
4.5. Sicherheit	42
4.5.1. Session-Management	42
4.5.3. WLAN Sicherheit	44
4.6. Installation	47
4.7. Das System erweitern	48
5. Schlussbetrachtung	52
5.1. mögliche Erweiterungen	52
5.2. Fazit	52

1. Einführung, Thema

1.1. Über dieses Projekt

Dieses Projekt ist ein Ergebnis des Projektsemesters des Studiengangs Angewandte Informatik an der FHTW Berlin. In Vorbereitung auf das Erstellen der Diplomarbeit werden im 7. Semester des Studiengangs verschiedene Projektthemen angeboten, die in einer Gruppe zu bearbeiten sind.

Ziele des Projektes sind unter, das Vorgehen beim Bearbeiten eines großen Projektes zu erlernen bzw. zu verbessern, das effiziente Zusammenarbeiten in einer Gruppe zu üben und zum späteren Erstellen der Abschlussarbeit Erfahrungen zu sammeln.

Wie das Thema "Einsatz von Informationstechnik in Gastronomiebetrieben" suggeriert, sollte im Verlauf des Projektes untersucht werden, inwieweit sich all die modernen Entwicklungen, Techniken und Medien wie Internet, PDA, MP3, Java und so weiter und so fort, die heutzutage in aller Munde sind, in einem Internetcafé, einem Szenelokal oder auch der "Kneipe an der Ecke" einsetzen lassen.

Das Gaststättengewerbe ist wohl eines der ältesten Gewerbe überhaupt. Und noch heute sieht man in vielen Restaurants Techniken aus der Anfangszeit der Gastronomie im Einsatz. Egal ob Getränkestrichliste auf dem Bierdeckel oder die nur für Kryptographie-Experten nachvollziehbare handschriftliche Rechnung – vom Einsatz moderner Technik fehlt vielerorts noch jede Spur. Wir schicken Mails in Sekunden um den Globus, lassen uns von unseren Autos sagen wo wir hinfahren können und sind mit mobilen Geräten die längst nicht mehr die Bezeichnung "Telefon" verdienen "always online", aber nach dem Essen fuchteln wir eine Ewigkeit nervös herum, winken mit der Brieftasche und machen allerlei Verrenkungen um die Aufmerksamkeit des Kellners zu bekommen, damit wir endlich die Rechnung begleichen dürfen.

**Lässt sich das nicht mit Technik unterstützen?
Geht das nicht einfacher???**



2. Ideen und Techniken

2.1. Istanalyse

Prototypischer Ablauf vor Projektbeginn

Standard: ohne Einsatz von Informationstechnik

Ein Gast betritt eine Speisegaststätte und nimmt an einem freien Tisch Platz. Der Ober sieht den Gast und reicht ihm die Speise- und Getränkekarte. Der Gast bestellt sofort ein Getränk. Der Ober notiert den Getränkewunsch zusammen mit der Tischnummer auf einem Schreibblock, geht zum Tresen und zapft das Getränk. Mit dem Getränk kehrt er zum Tisch zurück. Der Gast bestellt die gewünschte Speise. Der Ober notiert den Wunsch auf dem bereits mit der Tischnummer beschriebenen Zettel.

Hierauf geht der Ober zum Kücheneingang und teilt dem Koch den Speisewunsch des Gastes mit. Der Koch bereitet die Speise zu und stellt sie bereit. Durch ein Signal erkennt der Ober, dass die Speise bereit ist, holt sie ab und bringt sie dem Gast.



Nach Verzehr der Speise signalisiert der Gast dem Ober, dass er die Rechnung begleichen möchte.

Der Ober kommt mit einem Portemonnaie zum Tisch des Gastes, addiert das Verzehrte auf dem zum Tisch gehörenden Zettel und teilt die Summe dem Gast mit. Dieser bezahlt bar.

Nach dem Bezahlvorgang trägt der Ober den getätigten Umsatz in ein Buch ein. Am Ende des Geschäftstages werden die Umsätze addiert.

Variante: Einsatz einer (elektronischen) Registrierkasse

Bei Einsatz einer elektronischen, auf gastronomische Belange ausgelegten Registrierkasse gibt der Ober die weiterhin handschriftlich aufgenommene Bestellung am Tresen in die Kasse ein und lässt einen Beleg für die Küche drucken. Anhand dieses Beleges bereitet der Koch die Speisen.

Wünscht der Kunde zu bezahlen, wird nach Eingabe eines Schlüssels automatisch die Gesamtrechnung ausgedruckt. Die Ausgabe der Tages-, Wochen- und weiterer Bilanzen erfolgt auf Anforderung automatisch.

Variante: halbautomatische, drahtlose Aufnahme der Bestellungen

Ein weit verbreitetes System bedient sich bereits drahtloser Kommunikation zur Aufnahme und Übermittlung der Bestellungen.

Der Ober nimmt die Bestellung des Gastes auf und gibt diese in ein mitgeführtes drahtloses Erfassungssystem ein. Dies geschieht anhand Eingabe von Schlüsseln oder durch Einlesen eines Barcodes. Die Bestellung wird hiernach dem Koch übermittelt und in der zentralen Kasse gespeichert.

Das proprietäre, in der Anschaffung teure und herstellerabhängige System ist in Deutschland verbreitet in Eiscafés und einigen so genannten Szenekneipen im Einsatz.

Variante: SB-Restaurant

In so genannten Selbstbedienungsrestaurants gibt der Kunde seine Speise- und Getränkewünsche direkt an einer Theke auf bzw. stellt sich das Essen aus Auslagen zusammen und zapft die Getränke an einem Automaten. An einer zentralen Kasse werden die Produkte bezahlt.

Alternativ erfolgt zuerst eine Bezahlung der gastronomischen Wünsche verbunden mit der Ausgabe von Coupons. Diese werden an der Getränkeausgabe bzw. nach Fertigstellen der Gerichte eingelöst.

Bewertung des prototypischen Ablaufes ohne Einsatz von IT

den Geschäftsablauf positiv beeinflussende Aspekte

Der typische Ablauf in einem gastronomischen Betrieb ohne Informationstechnologie gestützte

Bestellaufnahme und -weiterleitung basiert vor allem auf persönlichem Kontakt zwischen Kunde und Ober sowie zwischen Ober und Koch.

Bei der Bestellung ermöglicht dies dem Gast schnelle Rückfragen und schafft eine persönliche Atmosphäre. Dem Ober ist es möglich, auf persönliche Wünsche des Gastes einzugehen sowie bei der Wahl des geeigneten Produktes behilflich zu sein.



negative Aspekte

Der erwähnte persönliche Kontakt kann in bestimmten Fällen hinderlich für die Abwicklung der Geschäfte sein.

Beispielsweise ist der Gast zum Aufgeben seiner gastronomischen Wünsche, zur Signalisierung des Zahlungswunsches sowie für Rückfragen jeglicher Art stets an die Anwesenheit des Obers gebunden. Wird der Ober nicht selbständig auf den Gast aufmerksam, ist der Gast gezwungen, den Ober auf sich aufmerksam zu machen.

Gelingt dies nicht sofort, führt dies häufig zu Unmut, zum Verwerfen des ursprünglichen Wunsches oder gar zur zukünftigen Wahl eines anderen Lokals. Des Weiteren ist die Registrierung des Kundenwunsches mit Hilfe eines Bestellzettels ein Unsicherheitsfaktor. Äußert der Kunde etwa unverständlich seine Bestellung oder geschieht die Notation nicht gewissenhaft, kann es zu Missverständnissen kommen. Weiterhin können diese Missverständnisse bei der Übermittlung des Auftrages an den Koch zustande kommen, sofern diese mündlich oder mit Hand geschriebenen Zetteln erfolgt.

Ein weiterer wichtiger Faktor ist die Kapazität des eingesetzten Personals. Ist das Verhältnis Anzahl der Kunden zu Anzahl der Mitarbeiter sehr hoch, kommt es häufig zu Wartezeiten und/oder Unregelmäßigkeiten. In kurzer Zeit tritt dann eine große Menge von Bestellungen, Zahlungs- oder anderen Wünschen auf, die alle persönlich von einem Angestellten bearbeitet werden müssen. Im schlimmsten Fall wird die Reihenfolge der Abarbeitung der Wünsche nach dem Zeitpunkt



des Eintreffens nicht eingehalten, sondern durch sekundäre Faktoren wie Entfernung des Tisches von der Theke, Auffälligkeit des Kunden o.ä. bestimmt.

Betreffend des Angebotes des gastronomischen Betriebes zeichnet sich der Nichteinsatz von IT dadurch aus, dass Speise- und Getränkekarten dem Gast in gedruckter Form ausgehändigt werden. Aus wirtschaftlichen Gründen ist die Lebensdauer einer solchen Karte hoch zu halten. Somit sind kurzfristige Änderungen des Angebotes nur über Umwege möglich, beispielsweise durch Beilage von Tageskarten oder handschriftliches Ändern in der Karte. Auch zeitlich beschränkte Angebote wie eine Happy Hour oder gar direkt durch die Bestellungen beeinflusste Angebote sind nur umständlich und für den Kunden teilweise schwer erfassbar oder gar nicht möglich.

Zusammenfassung

Die Geschäftsabläufe in Gastronomiebetrieben sind, mit Ausnahme von SB-Restaurants durchgehend an den Einsatz von Personal gebunden. Ein Ober dient gleichzeitig als Kommunikationsmittel zwischen Gast und Koch und ist Ansprechpartner für jegliche Wünsche.

Die Erstellung des Angebotes, die Aufnahme der Bestellungen sowie die Buchhaltung geschieht entweder gänzlich ohne den Einsatz von Informationstechnologie oder durch teilweise Unterstützung durch elektronische Registrierkassen, standardisierte Office-Anwendungen oder proprietäre, in der Anschaffung teure Erfassungssysteme.

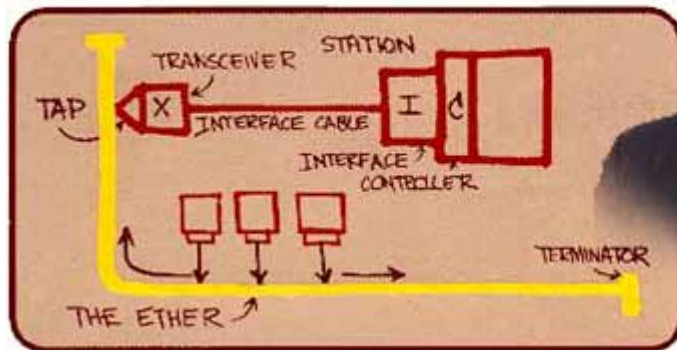
2.2. Anforderungen an eine IT-Lösung

- Das vorhandene Personal ist effektiver zu nutzen. Bestellungen und Wünsche sind automatisiert aufzunehmen und direkt elektronisch an die zuständige Person (Koch, Barkeeper,...) weiterzuleiten. Der Einsatz des Obers ist zu reduzieren auf die schnelle Auslieferung der fertigen Produkte sowie die Entgegennahme der Zahlungen, sofern nicht gleichzeitig automatisierte, vom Kunden zu bedienende Zahlungssysteme zum Einsatz kommen.
- Die Angebote des Gastronomiebetriebes sind unter Nutzung multimedialer Möglichkeiten dem Kunden in ansprechender Form zu präsentieren. Die Zusammenstellung des Angebotes erfolgt flexibel.
- Das Angebot soll bei Bedarf auch extern abgerufen werden, beispielsweise über die Webseite des Gastronomiebetriebes.
- Dem Kunden ist es zu ermöglichen, mit anderen Gästen oder den Angestellten elektronisch zu kommunizieren bzw. Inhalte aus dem Internet abzurufen. Auch erhält der Kunde auf Wunsch elektronisch Informationen zum Status seiner Bestellung.
- Informationen für den Inhaber des Betriebes, beispielsweise statistische Daten über Kundenbestellungen und für die Buchhaltung relevante Informationen sind automatisch zu erfassen und aufzubereiten.
- Die Wirtschaftlichkeit, der Umsatz und die Zufriedenheit der Kunden sind durch Verwirklichung o.g. Zielstellungen zu steigern.

2.3. Ideen zu eingesetzter Hardware

2.3.1. Ethernet

Am 22. Mai 1973 -- wurde das Ethernet von Robert M. Metcalfe in einem Memo zum ersten Mal erwähnt. Am 31. März 1975 meldete er es unter US-Patent-Nr. 4063220 an und am 13. Dezember 1977 wurde der Netzwerkstandard schließlich patentiert.



Zum Medientausch konzipiert, stieß der neue Standard eine Entwicklung an, der viele heute standardisierte und selbstverständliche Errungenschaften zu verdanken sind.

Mit der am Xerox Palo Alto Research Center entwickelten proprietären Lösung hat jedoch das heutige Ethernet außer dem Namen nicht mehr viel gemeinsam.

Xerox tat sich mit Intel und DEC zusammen, um das System weiter voranzubringen. Lizenzen an andere Hersteller wurden beinahe kostenlos vergeben, so dass sehr schnell eine große Menge an PC-Karten und sonstiger Vernetzungstechnik für Ethernet von zahlreichen Herstellern auf den Markt kamen. Diese Vielfalt führte zu einer sehr sauberen Implementierung des Standards, da sich kein Hersteller leisten konnte, dass seine Komponenten mit denen anderer Hersteller Probleme machen.

Im Laufe der Zeit wurden einige Ergänzungen des Ethernet-Standards eingeführt, insbesondere was Verkabelung und Geschwindigkeit angeht. Diese Ergänzungen werden durch ein einheitliches Namensschema gekennzeichnet: Zuerst kommt die Bandbreite in Mbit/s, dann das Wort "Base" (steht für Basisbandübertragung. Es gab auch Überlegungen für eine breitbandige Übertragung, die durch "Broad" gekennzeichnet wurde, sich aber nicht durchsetzen konnte.) und anschließend die Maximallänge in 100m bei Koaxverkabelung bzw. ein Buchstabenkürzel, das die Verkabelungsart angibt. Das ursprüngliche Ethernet, später als "Thickwire" bezeichnet, heißt nach diesem Schema "10Base5". Es hat heute keine Bedeutung mehr.

Mit einem dünneren und damit billigeren Koaxialkabel gab es dann das "Cheapernet" oder "Thinwire" genannte 10Base2. Es erreichte sehr große Verbreitung und ist bis heute noch recht häufig anzutreffen, allerdings mit stark rückläufiger Tendenz.



Heute wird fast ausschließlich eine strukturierte Verkabelung mit zwei bzw. vier verdrehten Leitungspaaren (UTP-Kabel) verwendet, die es zunächst nur als 10BaseT gab.

Mittlerweile beherrschen die meisten Netzwerkkarten zusätzlich 100BaseT, meist als "Fast Ethernet" bezeichnet.

2.3.2. Wireless LAN

Mit der Bezeichnung Wireless LAN (WLAN) wird eine ganze Protokollfamilie 802.11 der IEEE zusammengefasst, die unterschiedliche drahtlose Übertragungstechnologien für den Aufbau eines LANs spezifiziert. Die Spezifikation wurde nach sieben Jahren Entwicklungszeit 1997 zum ersten Mal von der IEEE veröffentlicht und hat seit dem zahlreiche Erweiterungen und Modifikationen erfahren.



Die ursprüngliche Spezifikation IEEE 802.11 umfasst ein Zugriffsprotokoll und drei alternative physikalische Übertragungstechnologien, davon zwei im 2,4 GHz (sog. ISM-Band, Industrial, Scientific, Medical) Funkwellen-Bereich und ein Infrarot-Verfahren. Das lizenzfreie ISM-Band ist für die Funkübertragung unter Kostenaspekten besonders interessant und lässt sich weltweit zur Übertragung von Daten nutzen. Die Verwendung des ISM-Bandes ist allerdings in einigen Ländern (darunter insbesondere Frankreich, Spanien und Japan) auf bestimmte Teilfrequenzen eingeschränkt. Zusätzlich arbeiten auch weitere Verfahren wie Bluetooth in diesem Band, so dass es zu Störungen bei der gleichzeitigen Nutzung des Frequenzbereichs kommen kann. Letztlich führen auch andere Haushaltsgeräte (z.B. Microwellenherde) zur Störung der Übertragungsraten.

Die ursprüngliche Spezifikation IEEE 802.11 sieht Übertragungsraten von 1-2 MBit/s vor. Mit Geräten zum Standard IEEE 802.11b werden mittlerweile theoretische Übertragungsraten von 11 MBit/s erreicht. Im gleichen Band erreicht eine neuere



Spezifikation des Protokolls (IEEE 802.11g) bis zu 54 MBit/s. Die typische Übertragungreichweite für diese Technologie liegt innerhalb von Gebäuden bei ungefähr 30-100 Metern mit einer Sendeleistung von 100 mW. Für Anwendungen im Freien sind mit der Nutzung von Richtantennen bis zu 2km Reichweite z.B. zur Verbindung von Teilnetzen möglich. Sämtliche Betriebsarten in diesem Band sind abwärtskompatibel zu Ihren Vorgängern und entsprechende Geräte lassen sich teilweise per Firmware-Update auf die jeweils nächste Systemversion aktualisieren.

Die Geräte zur Spezifikation IEEE 802.11a bzw. IEEE 802.11h verwenden ein anderes Übertragungsband (5.15-5.35 + 5.72-5.82 GHz) und ermöglichen ebenfalls Übertragungsraten von bis zu 54 MBit/s. Das Protokoll IEEE 802.11h ergänzt das IEEE 802.11a Protokoll um automatische Frequenzwahl (DFS) und Sendeleistungssteuerung (TPC). Im 5 GHz Band stehen zur Übertragung 18 nicht überlappende Sendekanäle zur Verfügung (für IEEE 802.11 b/g nur sind es nur drei). Die zukünftige Spezifikation IEEE 802.11 SuperG soll sogar Übertragungsraten bis zu 108 MBit/s erlauben.

Die Verwendung des 5 GHz Frequenzbereichs ist mit weniger Störungen und daher potenziell höheren effektiven Übertragungsraten verbunden, allerdings lassen sich Geräte dieses Standards nicht mit Geräten im 2,4 GHz-Band kombinieren. Die Reichweite zur Übertragung in diesem Bereich liegt für IEEE 802.11a Geräte bei etwa 20-30% der für IEEE 802.11b üblichen Reichweiten. In Deutschland ist hier die Sendeleistung auf max. 60 mW beschränkt. Für WLANs nach der Spezifikation IEEE 802.11h sind zu IEEE 802.11b vergleichbare Sendeleistungen und damit verbundene Reichweiten spezifiziert.

Mit folgenden Problemen waren die Entwickler des 802.11-Standards konfrontiert:

Störungen:

Diese wurden durch die Spreizbandtechnik weitgehend unschädlich gemacht.

Unbeabsichtigtes Mithören:

Wenn sich die Reichweitenbereiche zweier WLANs überschneiden, können Datenpakete aus dem einen ins andere übertragen werden. Dies lässt sich verhindern, indem man in den Basisstationen die Mac-Adressen der eigenen Endgeräte einträgt und Pakete mit anderen Adressen verwirft. Zudem kann bei den eigenen Geräten eine Kennung (BSSID, Basic Service Set Identifier) eingetragen werden, durch die die Kommunikation auf Basisstationen mit gleicher BSSID eingeschränkt wird.

Absichtliches Mithören:

Lauschangriffe von außen sind schon deswegen schwierig, weil durch die Spreizbandtechnik auf jeder Frequenz nur mit sehr geringer Energie gesendet wird. Zudem müsste der Lauscher bei FHSS den Frequenzsprüngen folgen können bzw. bei DSSS die verwendete Bitfolge kennen. Zusätzlich können die Pakete optional verschlüsselt werden. Das Verfahren dazu heißt WEP (Wired Equivalence Privacy) und arbeitet mit 40, 64 oder 128 Bit langen Schlüsseln.

2.3.3. Bluetooth

Bluetooth ist ein Standard für die In-House-Kommunikation mittels Funk. Der Bluetooth-Standard wurde 1998 von Ericsson, IBM, Intel, Nokia und Toshiba entwickelt und hat das Ziel, die Kurzstrecken-Kommunikation zwischen Endgeräten wie Notebooks, Organizational PDAs und Handys zu unterstützen.

Als Standard ist eine Sendeleistung von 1 mW (0 dBm) definiert, wodurch die Entfernungen für die Kommunikation auf 10 Meter begrenzt ist. Dem Verfahren nach arbeitet Bluetooth in Anlehnung an 802.11 im ISM-Band, mit einem schnellen



Frequenzsprung-Verfahren. Die Kanalbreite der 79 Kanäle beträgt 1 MHz, wobei die Frequenz bis zu 1.600 mal in der Sekunde zwischen 79 Frequenzen springt (FHSS). Um die

Nachrichten empfangen zu können, müssen die Empfänger auf die Sprungfolge des Senders synchronisiert sein.

2.3.4. PDA

Definition

Ein PDA ist ein tragbarer Rechner in Notizbuchformat. PDAs bestehen meist aus einer Anzeige, die den Großteil der Oberfläche des Geräts einnimmt. Bei den meisten PDA werden die Eingaben mittels eines Stift-ähnlichen Stabes direkt auf der druckempfindlichen Anzeige gemacht. Dazu haben die meisten PDA eine Software zur Handschriftenerkennung eingebaut. Bei einigen PDA ist offen oder unter einer Klappe oder Schieber eine kleine Tastatur eingebaut.

Geschichte

Die ersten, noch relativ großen PDA wurden in den 1990er Jahren unter der Bezeichnung Newton von der US-amerikanischen Firma Apple hergestellt.

Mitte der 1990er Jahre kam die Firma US-Robotics (zwischenzeitlich 3COM, dann Palm Inc., jetzt palmOne) mit kleinen und sehr erfolgreichen Geräten unter den Namen Pilot (später Palm) heraus.

Ebenfalls in den 1990er Jahren wurden von der britischen Firma Psion eine Reihe von recht leistungsfähigen PDAs mit Tastatur entwickelt. Das erste Modell Organizer I entstand bereits 1984. Mit dem Organizer II und dem Microcomputer Serie 3 gelang dann der Durchbruch. Die Psion-Modelle erfreuen sich jedoch auch nach dem Ausstieg

Psions im Jahr 1999 aus dem Endkundenmarkt wegen des ausgereiften, logischen Betriebssystems weiterhin ungebrochener Beliebtheit. Die Industriemodelle netBook und netpad sowie Workabout werden von Psion-Teklogix weiterhin angeboten.

Verwendungszweck

Die wichtigsten Anwendungen sind Adressbuch, Terminplaner, Notizblock, Aufgabenplaner, Taschenrechner und Spiele. In den meisten aktuellen Geräten sind viele weitere Anwendungen eingebaut. Häufig kann man Software aus dem Internet nachladen und per Kabel (USB oder seriell) oder kabellos per Infrarot- bzw. Bluetoothschnittstelle in den PDA überspielen.



Eine dieser Anwendungen ist die Wikipedia selbst, die es mittlerweile sogar schon in zwei Versionen für den Palm und andere PDAs gibt.

Ein großer Vorteil aktueller PDAs ist, dass man sie problemlos mit dem PC abgleichen kann (HotSync). Dabei können z.B. E-Mails und Adressen (z.B. aus Outlook, Notes etc) in den PDA übertragen werden.

Typische Anschluss- bzw. Erweiterungsmöglichkeiten eines PDA

Dank einer Vielzahl an Anschluss- und Erweiterungsmöglichkeiten sind PDAs zu universellen Wegbegleitern in den unterschiedlichsten Situationen geworden: Man kann sie zum Musik hören und Fotos schießen verwenden. Als Fußgänger oder im Auto stehen sie als Navigationshilfe zur Seite. Die Vorteile durch das Zusammenführen mehrerer Geräte sind im allgemeinen: Weniger Gewicht, kompaktere Geräte und eine einzelne Stromversorgung. Darüber hinaus geringere Redundanz beim Datenabgleich. So ist es beispielsweise mit einem SmartPhone (Mobiltelefon und PDA in einem) nicht mehr notwendig zwischen drei Datenspeichern (PDA, Mobiltelefon und PC) einen Abgleich z.B. bei einer Telephonnummernänderung mitzuziehen sondern nur noch zwischen zwei Geräten. Die Nachteile sind wie bei allen Multifunktionsgeräten: Anfälliger für Störungen, fällt ein Gerät aus (Accu leer) so sind zumeist alle Geräte davon betroffen.

Literaturfreunde werden sich angesichts der Speicherkarten darüber freuen, dass man Krieg und Frieden bequem in die Westentasche stecken kann und mit Hintergrundbeleuchtung auch bei wenig vorteilhaften Lichtverhältnissen in Bus, Schlafwagen oder Bett lesen kann. Ein hochauflösendes, kontrastreiches Farbdisplay ist hierfür Voraussetzung.

Erweiterungsmöglichkeiten bzw. Hybrid-Geräte:

Speicherkarten (CF=Compact Flash, SM=Smart Media u.v.a.m) zur Datensicherung und zur bedarfsweisen Kapazitätserweiterung bzw. zum schnelleren Datentransfer zwischen PC und PDA zum Beispiel für Multimedia-Daten.

Navigationsgehilfen (GPS) z.B. GPS Springboard Modul für den Handspring Visor oder Garmin iQue360

SmartPhones (=Mobiltelefon und PDA zusammengefasst in einem Gerät) z.B. VisorPhone oder Nokia Communicator

MP3-Spieler-Foto-PDA (Mobiltelefon, PDA und Kamera) Sony CLIÉ PEG-NX73V

MP3-Foto-Smartphone (Mobiltelefon, MP3 Player, PDA und Kamera) z.B. Sony Ericsson T900



Anschlüsse

Infrarot - zum "beamen" (=optischen Übertragen mittels unsichtbaren Infrarot, Vergleichbar mit einer Fernseher Fernsteuerung) von Digitalen Visitenkarten, Notizen oder ToDo's an andere Besitzer eines PDA's die sich in unmittelbarer Nähe befinden) So kann man z.B. auch Krieg und Frieden einem interessierten Reisekollegen binnen einiger Minuten beamen.

Bluetooth - zur drahtlosen Übertragung (Funk) zwischen zwei Bluetooth- Geräten.
WLAN (=Wireless LAN) zum einwählen in das Internet via lokalen Access Points (AP), zum Beispiel am Flughafen, an Bahnhöfen und manchen Caféhäusern.

Betriebssysteme

Das EPOC-Betriebssystem bildet die Basis für Symbian OS, welches in Smartphones wie Nokia 9210/9290, 3650, 7650 und Sony Ericsson P800 zum Einsatz kommt.

Ein weiteres Betriebssystem für PDA ist OPIE das im Moment (August 2003) für Sharp Zaurus, Siemens Simpad und HP iPAQ in mehreren Sprachen kostenlos zur Verfügung steht.

Viele PDA-Typen werden heute unter der Bezeichnung **Pocket PC** vertrieben, die allesamt mit dem Betriebssystem Microsoft Windows für Pocket PC ausgestattet sind und sich nahezu nahtlos in die Windows Familie integrieren.

Ein solcher Pocket PC, der **Compaq iPAQ**, kam für die Projektstudie zum Einsatz. Neben Wireless LAN und Bluetooth besitzt er, so sagt es zumindest die Werbung, viele praktische Funktionen, die man erst beim Benutzen entdeckt.



2.4. Mögliche Techniken

2.4.1. HTML

HTML bedeutet HyperText Markup Language. Es handelt sich dabei um eine Sprache, die mit Hilfe von SGML (Standard Generalized Markup Language) definiert wird. SGML ist als ISO-Norm 8879 festgeschrieben.

HTML ist eine sogenannte Auszeichnungssprache (Markup Language). Sie hat die Aufgabe, die logischen Bestandteile eines Dokuments zu beschreiben. Als Auszeichnungssprache enthält HTML daher Befehle zum Markieren typischer Elemente eines Dokuments, wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen.

Das Beschreibungsschema von HTML geht von einer hierarchischen Gliederung aus. HTML beschreibt Dokumente. Dokumente haben globale Eigenschaften wie zum Beispiel einen Titel oder eine Hintergrundfarbe. Der eigentliche Inhalt besteht aus Elementen, zum Beispiel einer Überschrift 1. Ordnung. Einige dieser Elemente haben wiederum Unterelemente. So enthält ein Textabsatz zum Beispiel eine als fett markierte Textstelle, eine Aufzählungsliste besteht aus einzelnen Listenelementen, und eine Tabelle gliedert sich in einzelne Tabellenzellen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >
<html >
<head>
  <title>Ein Dokument</title>
</head>
<body>
  <h1>Ich bin ein HTML-Dokument</h1>
</body>
</html >
```

Ein einfaches HTML-Dokument

Eine der wichtigsten Eigenschaften von HTML ist die Möglichkeit, Verweise zu definieren. Verweise ("Hyperlinks") können zu anderen Stellen im eigenen Projekt führen, aber auch zu beliebigen anderen Adressen im World Wide Web und sogar zu Internet-Adressen, die nicht Teil des WWW sind.

Durch diese einfache Grundeigenschaft eröffnet HTML völlig neue Welten. Das Bewegen zwischen räumlich weit entfernten Rechnern wird bei modernen grafischen WWW-Browsern auf einen Mausklick reduziert. In Ihren eigenen HTML-Dateien können Sie Verweise notieren und dadurch inhaltliche Verknüpfungen zwischen Ihren eigenen Inhalten und denen anderer Anbieter herstellen. Auf dieser Grundidee beruht letztlich das gesamte World Wide Web, und dieser Grundidee verdankt es seinen Namen.

Im Zeitalter der Kommerzialisierung des Internet sind natürlich auch die Verweise zu einem kommerziellen Gegenstand geworden. Anklickbare Werbe-Grafiken ("Banner") auf häufig besuchten Seiten führen zu Anbietern, die für die Platzierung der Banner Miete bezahlen. Auch das sind Verweise. Glücklicherweise gibt es daneben aber weiterhin genügend "herkömmliche" Verweise im WWW, die einfach nur die Grundidee des Web verfolgen und zur weltweiten Vernetzung von Information beitragen wollen.

HTML ist ein sogenanntes Klartext-Format. HTML-Dateien können Sie mit jedem beliebigen Texteditor bearbeiten, der Daten als reine Textdateien abspeichern kann. Es gibt also keine bestimmte Software, die man zum Erstellen von HTML-Dateien benötigt. Zwar gibt es längst mächtige Programme, die auf das Editieren von HTML spezialisiert sind, doch das ändert nichts an der entscheidenden Eigenschaft: HTML ist nicht an irgendein bestimmtes, kommerzielles Software-Produkt gebunden. Diese wichtige, vielleicht sogar wichtigste Eigenschaft von HTML sollten Sie immer im Auge behalten, wenn man Ihnen erzählen will, daß Web Publishing nur mit bestimmten Software-Produkten möglich sei.

Die Klartext-Befehle von HTML sind für Maschinen und Menschen gedacht. Wer keine sinnlosen Vorurteile gegenüber sichtbaren Befehlen am Bildschirm hat, wird in HTML eine überraschend einfache Befehlssprache finden. Die Sprache ist Englisch, doch da die Anzahl der Befehle begrenzt ist, ist es auch ohne tiefere Kenntnisse der englischen Sprache möglich, sich in HTML hineinzudenken.

HTML ist als Auszeichnungssprache zum Erstellen von WWW-Seiten gedacht - eigentlich. HTML-Dateien funktionieren aber nicht nur im WWW. Es ist kein Problem, eine HTML-Datei lokal auf jedem Rechner mit einem WWW-Browser zu öffnen. HTML-Dateien sind deshalb auch ideal geeignet für lokale Dokumentationen, für CD-ROM-Oberflächen, für Readme-Dateien usw. Mit HTML und seinen unmittelbaren Ergänzungssprachen CSS und JavaScript, die ebenfalls lokal funktionieren, können Sie auch anspruchsvolle Projekte realisieren, die nicht für den Einsatz im WWW gedacht sind. Egal ob Sie Ihr Tagebuch fürs nächste Jahrtausend fit machen möchten, ob Sie bei der nächsten Version Ihrer Software eine HTML-basierte Online-Hilfe begeben wollen, oder ob Sie eine informative CD produzieren wollen - HTML ist längst das verbreitetste Dateiformat der Welt. Ihre HTML-Dateien laufen auf jedem Rechner, auf dem ein WWW-Browser installiert ist - und ein Rechner, auf dem kein WWW-Browser verfügbar ist, darf mittlerweile bei aller Rücksicht als ein "veralteter Rechner" bezeichnet werden.

(Quelle: selfHTML)

2.4.2. PHP

PHP (Akronym für "PHP: Hypertext Preprocessor") ist eine weit verbreitete und für den allgemeinen Gebrauch bestimmte Open Source Skriptsprache, welche speziell für die Webprogrammierung geeignet ist, und in HTML eingebettet werden kann.

Hört sich einfach an, aber was heißt es genau? Ein Beispiel:

```
<html >
  <head>
    <title>Beispiel </title>
  </head>
  <body>

    <?php
      echo "Hallo, ich bin ein PHP-Skript!";
    ?>

  </body>
</html >
```

ein einfaches PHP-Skript

Beachten Sie den Unterschied zu einem Skript, welches in anderen Sprachen wie Perl oder C geschrieben wurde -- anstatt ein Programm mit vielen Anweisungen zur Ausgabe von HTML zu schreiben, schreibt man einen HTML-Code mit einigen eingebetteten Anweisungen, um etwas auszuführen (z.B. um - wie oben - Text auszugeben). Der PHP-Code steht zwischen speziellen Anfangs- und Schlusstags, mit denen man in den PHP-Modus und zurück wechseln kann.

Was PHP von clientseitigen Sprachen wie Javascript unterscheidet, ist dass der Code auf dem Server ausgeführt wird. Sollten Sie ein Skript wie das obige auf ihrem Server ausführen, würde der Besucher nur das Ergebnis empfangen, ohne die Möglichkeit herauszufinden, wie der zugrundeliegende Code aussieht. Sie können ihren Webserver auch anweisen, alle ihre HTML-Dateien mit PHP zu parsen, denn dann gibt es wirklich nichts, das dem Benutzer sagt, was sie in petto haben.

Das Beste an der Verwendung von PHP ist, dass es für Neulinge extrem simpel ist, aber auch einen riesigen Funktionsumfang für den professionellen Programmierer bietet. Scheuen Sie sich nicht, die lange Liste der PHP-Funktionen zu lesen. Sie können einsteigen, und binnen weniger Stunden bereits mit dem Schreiben von einfachen Skripten beginnen.

Was kann PHP?

Alles. PHP ist hauptsächlich auf serverseitige Skripte fokussiert, weshalb Sie alles tun können, was auch ein anderes CGI Programm kann, wie z.B. Formulardaten sammeln, dynamische Inhalte für Websites generieren oder Cookies senden und empfangen. Aber PHP kann noch viel mehr.

Es gibt drei Hauptgebiete, in denen PHP Skripte genutzt werden.

Serverseitige Skripte. Dies ist das traditionelle und auch Hauptfeld von PHP. Sie benötigen dazu drei Dinge. Den PHP Parser (CGI oder Server-Modul), einen Webserver und einen Webbrowser. Sie müssen den Webserver, verbunden mit einer PHP Installation laufen lassen. Sie können sich die Ausgabe der PHP Programme über den Server mit einem Webbrowser ansehen. Für weitere Informationen lesen Sie bitte im Abschnitt Installation weiter.



Skripte auf der Kommandozeile. Sie können auch PHP Skripte schreiben, um sie ohne einen Server oder einen Browser laufen zu lassen. Hierfür benötigen Sie nur den PHP Parser. Diese Art der Verwendung ist für die regelmäßig auszuführende Skripte mittels cron (unter *nix oder Linux) oder dem Task Scheduler (unter Windows). Diese Skripte können auch für einfache Aufgaben zur Verarbeitung von Text verwendet werden. Weitere Informationen dazu finden Sie im Abschnitt Using PHP from the command line.

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 wahrscheinlich anderen. PHP unterstützt auch die meisten der heute gebräuchlichen Webserver. Dies umfasst Apache, Microsoft Internet Information Server, Personal Web Server, Netscape und iPlanet Server, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd, und viele andere. Für den Großteil der Server

bietet PHP ein eigenes Modul, für die anderen, welche den CGI Standard unterstützen, kann PHP als CGI Prozessor arbeiten.

So haben Sie Freiheit, PHP auf dem Betriebssystem und dem Webserver Ihrer Wahl laufen zu lassen. Weiters können Sie je nach Vorliebe prozedural, objektorientiert, oder gemischt programmieren. Auch wenn in der derzeitigen Version von PHP nicht jedes Standard OOP Feature realisiert ist, sind viele Bibliotheken und große Applikationen (inklusive der PEAR Bibliothek) exklusiv unter Verwendung von OOP Code geschrieben worden.

Mit PHP sind Sie nicht auf die Ausgabe von HTML beschränkt. Seine Fähigkeiten umfassen auch das dynamische Generieren von Bildern, PDF Dateien und Flash Animationen (mittels libswf und Ming). Sie können auch leicht jede Art von Text, wie XHTML oder irgendeine andere XML Datei ausgeben. PHP kann diese Dateien automatisch generieren und im Dateisystem speichern, anstatt diese nur auszugeben bzw. auszudrucken, und formt auch serverseitigen Zwischenspeicher Ihrer dynamischen Inhalte.

Vielleicht die größte und bemerkenswerteste Stärke von PHP ist seine Unterstützung für eine breite Masse von Datenbanken. Eine datenbankgestützte Website zu erstellen ist unglaublich einfach. Die folgenden Datenbanken werden zur Zeit unterstützt:

Adabas D	Ingres	Oracle (OCI7 und OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (nur Lesezugriff)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

PHP besitzt auch die Erweiterung DBX zur Datenbankabstraktion, welche Ihnen die transparente Verwendung irgendeiner von dieser Erweiterung unterstützten Datenbank erlaubt. Weiterhin unterstützt PHP ODBC, den Open Database Connection Standard, mit welchem Sie sich zu jeder anderen diesen Weltstandard unterstützenden Datenbank verbinden können.

PHP unterstützt auch die Kommunikation mit anderen Services, welche Protokolle wie LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (auf Windows) und unzählige andere unterstützen. Sie können auch einfache Netzwerk Sockets öffnen, und unter Verwendung irgendeines Protokolls kommunizieren. PHP unterstützt auch WDDX (Web Distributed Data Exchange) zum Austausch komplexer Daten zwischen Programmiersprachen. Da wir gerade vom Zusammenwirken sprechen: PHP bietet auch Unterstützung zur Instantiierung von Java Objekten, und deren transparente Verwendung als PHP Objekte. Sie können auch unsere CORBA Erweiterung verwenden, um auf entfernte Objekte zuzugreifen.

PHP verfügt über äußerst hilfreiche Textverarbeitungsfunktionen, von den regulären Ausdrücken (POSIX erweitert oder Perl) bis zum Parsen von XML Dokumenten. Für den Zugriff und das Parsen von XML Dokumenten unterstützen wir die Standards

SAX und DOM. Sie können unsere XSLT Erweiterung verwenden, um XML Dokumente zu transformieren.

Wenn Sie PHP im Bereich des E-Commerce nutzen, werden Sie Funktionen für Cybercash Payment, CyberMUT, VeriSign Payflow Pro und CCVS für Ihre Onlineprogramme zur Zahlungsabwicklung zu schätzen lernen.

PHP besitzt auch viele andere interessante Erweiterungen, wie mnoGoSearch für Suchmaschinen, die IRC Gateway Funktionen, viele Komprimierungswerkzeuge (gzip, bzip2), Kalenderumrechnung, Übersetzung...

(Auszug aus "Das PHP Tutorial")

2.4.6. Java

Java ist nicht nur eine Insel sondern eine plattformunabhängige Programmiersprache, die ursprünglich von der Firma Sun Microsystems, entwickelt wurde. Java wird in allen IT-Bereichen von Handel, Industrie und Verwaltung eingesetzt und ist für unzählige Betriebssysteme und Plattformen, vom mobilen Telefon bis hin zur Echtzeit-Großrechneranlage kostenlos verfügbar.



Java ist...

einfach & klein

Java ist für Einsteiger und Profis gleichermaßen geeignet. Einfache Fenster-Anwendungen, Browser-Applets oder komplexe Client-/Server- und Datenbank-Module lassen sich dank der klaren Spracheigenschaften und dem gegenüber C++ verbesserten objektorientierten Konzepts schnell umsetzen.

Für mehrschichtige Client-/Server-Anwendungen wird die "Java 2 Enterprise Edition" angeboten. Hierin finden sich auch Java Servlets und Java Server Pages für die Erstellung interaktiver Webseiten.

objekt-orientiert & verteilt

"Objekt-orientiert" ist sicherlich eines der meist-verschlunzten Wörter ("Jetzt zugreifen: objekt-orientierte Programme aus der Tube!"), das ändert jedoch nichts daran, dass es sich, aus guten Gründen, in der Software-Entwicklung durchgesetzt hat.

Wie der Name schon sagt, konzentriert man sich bei der objekt-orientierten Programmierung auf die Erstellung sog. "Objekte". Grob gesagt besteht ein Objekt aus Daten und Funktionen (sog. "Methoden"), die diese Daten verändern. Im Idealfall erzeugt man mit objekt-orientierter Programmierung wiederverwendbare Objekte ("Klassen"). Bei fortschreitender Programmierung kann man dann auf einen immer größer werdenden Pool an schon bestehenden Objekten zurückgreifen ("Klassenbibliotheken"). Auf entfernt liegende Objekte können Java-Programme sehr einfach über das Internet zugreifen.

kompiliert & schnell

Java-Programme werden im Gegensatz zu vielen anderen Compiler-Sprachen nicht direkt in eine für einen "echten Prozessor" verständliche Menge an Befehlen übersetzt, sondern zunächst in den sog. "Java Bytecode" umgewandelt.

Dieser Bytecode ist dennoch sehr maschinennah. Erst auf der Zielmaschine wird er in einer Java Laufzeit-Umgebung ausgeführt (Java Runtime Environment, JRE).

Da bei Java für gewöhnlich zweimal kompiliert wird (einmal beim Entwickler und einmal beim Nutzer), wird dieses Prinzip Just-In-Time Compiler (JIT) genannt. Dies hat neben den genannten Vorteilen natürlich auch den Nachteil, dass Java-Programme länger zum Starten benötigen, da während der Initialisierung der Maschinen-Code erzeugt wird. Auch hier stehen jedoch für die unterschiedlichen Anwendungszwecke ebenso unterschiedliche Laufzeitumgebungen bereit (one size fits all gibt's nicht). J2EE mit der sog. "HotSpot Performance Engine" erlaubt den gemischten Betrieb aus interpretiertem und kompiliertem Code: Zur Laufzeit prüft die Umgebung die kritischen Stellen des Codes (die sog. "HotSpots") und führt darauf weitere Optimierungen durch, so daß das Programm "schneller wird, je länger es anläuft."

robust & sicher

Die Kombination Compiler UND Laufzeitumgebung hat für Java unter einem anderen Gesichtspunkt große Vorteile: Einerseits kann der "Vor"-Compiler auf Typenfehler und dergleichen prüfen (Java ist eine typenstrenge Sprache), andererseits kann die Java-Runtime Umgebung Fehler während der Laufzeit abfangen und behandeln - ein kompiliertes Programm würde schlimmstenfalls abstürzen.

plattformunabhängig & portierbar

Hier wird ein Traum der Software-Entwickler wahr:

Jeder kennt das Problem mit den verschiedenen Rechner- und Betriebssystemen. Auf dem einen läuft's, auf dem anderen nicht und für Software-Entwickler ist das alles wirklich ein Graus! (Man stelle sich vor, es gäbe 10 verschiedene Video-Systeme..!) Auch hier greift das Compiler/Interpreter-Konzept: Der Compiler übersetzt das Programm in den "Bytecode", der völlig unabhängig von irgendeinem bestehenden Prozessor ist, auf dem das Programm später vielleicht mal läuft.

Ein- und dasselbe Java-Programm läuft ohne Änderung auf allen unterstützten Systemen!!

"multithread"-fähig & dynamisch

In einem modernen Programm passieren viele Dinge gleichzeitig. Diese Eigenschaft nennt man "multithreading". Java unterstützt den Entwickler mit verschiedenen Funktionen beim Entwurf solcher parallel-verarbeitender Programme, so daß er sich auf seine eigentliche Aufgabe konzentrieren kann.

2.4.3. Servlets

Servlets stellen Java Klassen dar, die Anforderungen nach Web Seiten verarbeiten können. Das Schreiben dieser Klassen ist sehr technisch und nahe am Medium. Die Entwicklung von Web Seiten mit Servlets gleicht daher der Programmierung mit Assembler. Der Programmierer hat sich um fast alles selbst zu kümmern wie der Code in Beispiel 1 zeigt. Näher an der Technik geht es nur mit CGI Skripten.

Servlets sind für die Realisierung technischer Aufgaben das Mittel der Wahl.

Typische Aufgaben für Servlets sind:

Kontrolle von Uploads und Downloads

Sicherheitsüberprüfung, Protokollierung

Lastverteilung

Das Erstellen von komplexen Web Anwendungen rein mit Servlets ist in vielen Fällen zu aufwändig und zu teuer.

Ein kleines Beispiel zur Annahme einer http-Bestellanfrage würde wie folgt aussehen.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class CafeServlet extends HttpServlet {

    public void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        String bestellung = req.getParameter("Bestellung");

        PrintWriter out = res.getWriter();

        out.println("<head><body>");
        out.println("<h2>Hier ist Ihr " +bestellung+ ". <h2>");
        out.println("</body></head>");
    }
}
```

ein einfaches Servlet zur Bestellannahme

2.4.4. JSP

Mit JavaServer Pages lassen sich schnell und einfach kleinere Web Anwendungen erstellen. Die Server Page ist eine um Java Code angereicherte HTML Seite. Der Entwickler kommt mit Ihnen schnell zu einem Ergebnis.

Mit JSP realisierte Web Anwendungen werden mit der Zeit unübersichtlich und schwer zu pflegen. In den Server Pages sind wie aus Beispiel 2 ersichtlich HTML und Java Code oft bunt gemischt. Abhilfe schafft die Verwendung von Helper Klassen und Taglibs. Eine Taglib ist eine Bibliothek mit Tags, die dem JSP Entwickler zur Verfügung stehen.

Eine weitere Trennung von Markup und Code kann durch die Verwendung von Server Pages in Verbindung mit einem Model View Controller Framework wie Apache Struts erzielt werden.

```
<html >
<body>
<% for( int i = 0; i < 10; i++)
  { %>
    <%=i %><br >
  <% } %>
</body>
</html >
```

JSP-Code in HTML

2.4.5. Javascript

JavaScript ist kein direkter Bestandteil von HTML, sondern eine eigene Programmiersprache. Diese Sprache wurde jedoch eigens zu dem Zweck geschaffen, HTML-Autoren ein Werkzeug in die Hand zu geben, mit dessen Hilfe sich WWW-Seiten optimieren lassen.

JavaScript-Programme werden wahlweise direkt in der HTML-Datei oder in separaten Dateien notiert. Sie werden nicht - wie etwa Java-Programme - kompiliert, sondern als Quelltext zur Laufzeit interpretiert, also ähnlich wie Batchdateien bzw. Shellscripts. Dazu besitzen moderne WWW-Browser wie Netscape oder Microsoft Internet Explorer entsprechende Interpreter-Software.

Auch bei mobilen Anwendungen hat Javascript Einzug gehalten. So besitzen etliche Smartphones, PDAs und andere mobile Endgeräte mit Browser-Funktionalität einen Interpreter für Javascript. Der Umfang der Implementierungen und teilweise auch die Resultate von gleichen Anweisungen auf verschiedenen Geräten unterscheiden sich jedoch beträchtlich.

2.4.7. MySQL

Der Datenbankserver MySQL ist die populärste Open-Source-Datenbank der Welt. Durch seine Architektur ist er äußerst schnell und er kann einfach eingerichtet werden. Die extensive Wiederverwendung von Code innerhalb der Software und ein minimalistisches Herangehen an die Erstellung von funktionsreichen Eigenschaften haben ein Datenbankmanagement-System hervorgebracht, das in Sachen Geschwindigkeit, Kompaktheit, Stabilität und einfachem Einsatz unschlagbar ist. Die eindeutige Trennung des Serverkerns von der Storage Engine ermöglicht die Wahl zwischen strikter Transaktionskontrolle oder sehr schnellem, transaktionslosen Plattenzugriff - je nachdem was gerade benötigt wird.

Der Datenbankserver MySQL ist kostenlos als GNU General Public License (GPL) verfügbar. Es sind auch kommerzielle nicht-GPL-Lizenzen für Anwender erhältlich, die nicht durch die GPL-Bestimmungen eingeschränkt sein wollen.



Der Datenbankserver ist in vier verschiedenen Varianten verfügbar:

MySQL Standard enthält die Standard Storage Engines von MySQL sowie die InnoDB Storage Engine. InnoDB ist eine transaktionssichere ACID-konforme Storage Engine mit Commit, Rollback, Absturz-Wiederherstellung und der

Möglichkeit, Datensätze auf Zeilenebene zu sperren. Diese Variante ist für Anwender, die die Hochleistungs-Datenbank MySQL mit voller Transaktionsunterstützung nutzen möchten. MySQL Standard wird unter der GPL lizenziert.

MySQL Pro ist die Variante mit kommerzieller Lizenz des MySQL Standard Datenbank Servers mit den gleichen Eigenschaften, einschließlich InnoDB-Unterstützung.

MySQL Max ist für Anwender, die frühzeitig Zugang zu neuen Funktionen und Eigenschaften haben möchten. Diese Variante beinhaltet die Standard Storage Engines, die InnoDB Storage Engine und andere Extras wie die Berkley Database (BDB) Storage Engine und SSL Transport-Layer-Verschlüsselung. MySQL Max unterstützt auch die Aufteilung von Tabellen auf verschiedene Dateien, was nützlich ist, wenn die Datei sonst größer wird als das Betriebssystem zulässt. In zukünftigen Releases wird MySQL Max mehr technisch führende Merkmale enthalten.

MySQL Classic enthält nur die Standard MySQL Storage Engines und unterscheidet sich von MySQL Pro und MySQL Standard dadurch, dass die InnoDB Storage Engine nicht enthalten ist. MySQL Classic ist nur als kommerzielle Lizenz erhältlich.

Funktionen:

ANSI SQL Syntax-Unterstützung

Der Datenbankserver MySQL unterstützt eine umfassende Untermenge der ANSI SQL 99 Syntax sowie einige Erweiterungen wie den REPLACE-Befehl und die LIMIT-Klausel für SELECT und DELETE. Außerdem wird die Syntax von anderen Datenbanksystemen unterstützt, um die Portierung von Anwendungen zu vereinfachen.

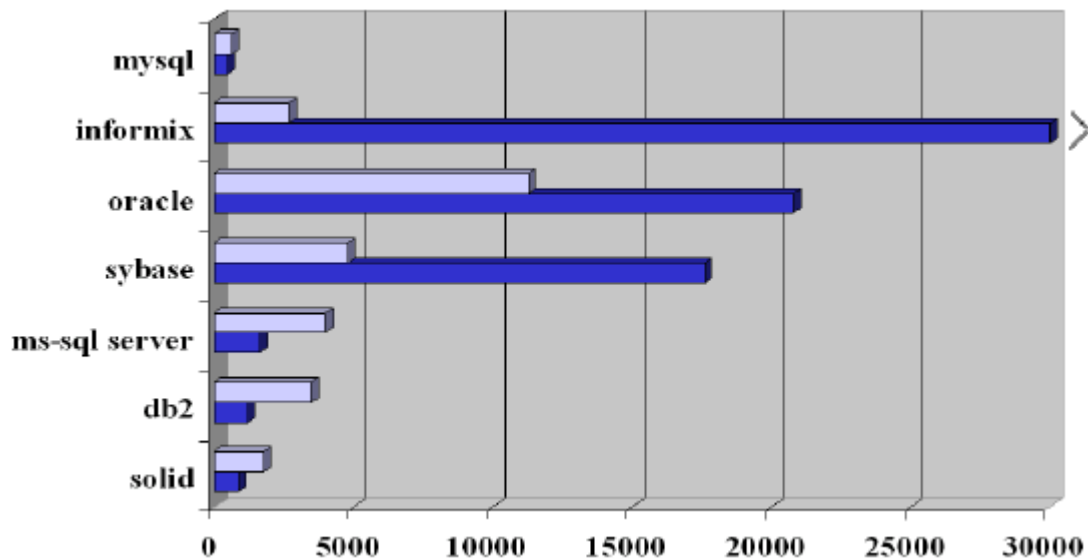
Mit der Version 4.0 decken wir einen noch größeren Bereich der ANSI SQL 99 Syntax ab, indem wir auch den UNION-Befehl unterstützen.
Plattformübergreifende Unterstützung

Wir stellen für viele Betriebssysteme optimierte Binärdateien zur Verfügung. Hierzu zählen Linux, Microsoft Windows, FreeBSD, Sun Solaris, IBM's AIX, Mac OS X, HP-UX, AIX, QNX, Novell NetWare, SCO OpenUnix, SGI Irix, und Dec OSF.

Sie können einen MySQL Datenbankserver von allen gängigen Betriebssystemen aus verbinden. Dabei können Sie fast jede Programmiersprache mit unserer 'thread safe' Client-Bibliothek oder einer der Produkte unserer Datenbanktreiber der Connector-Familie verwenden.

Unabhängige Storage Engines

Die einzigartigen, unabhängigen Storage Engines des Datenbankservers MySQL erlauben Ihnen die Wahl der Datenbank Storage Engine, die am besten für Ihre besonderen Bedürfnisse geeignet ist. Wenn Sie Sperrung auf Datensatzebene und Transaktionsunterstützung benötigen, können Sie die InnoDB Storage Engine verwenden. Wenn Ihre Anwendungen keine Transaktionen benötigt, können Sie die MyISAM Storage Engine verwenden um höchstmögliche Leistung zu erhalten.



DBMS Performance im Vergleich: Zeitdauer in Millisekunden für folgende Nutzeraufträge:
2 Mio. Tupel lesen (dunkel), 350.000 Tupel einfügen (hell)

Transaktionen

Mit Verwendung der InnoDB oder Berkley DB (DBD) Storage Engine unterstützt der Datenbankserver MySQL Transaktionen. Die InnoDB Storage Engine unterstützt außerdem Fremdschlüsseldefinitionen.

Flexible Sicherheitssysteme, einschließlich SSL-Unterstützung

Der Datenbankserver MySQL enthält ein erweitertes Berechtigungs- und Sicherheitssystem, einschließlich der Unterstützung von SSL Transport-Layer-Verschlüsselung. Mit Version 4.0 erlaubt Ihnen das Sicherheitssystem auch die Beschränkung von Serverressourcen auf Benutzerebene.

Query Caching

Version 4.0 des Servers enthält einen neuen Query Cache, der die Leistung von häufig verwendeten Abfragen deutlich erhöht, ohne eine besondere Programmierung zu benötigen. Im typischen Anwendungsfällen kann die Leistung um 200% erhöht werden.

Replikation

Sie können Datenbankreplikation nutzen, um erhöhte Stabilität und Geschwindigkeit zu erhalten. Mehrere "Slave"-Server replizieren dabei einen einzelnen "Master"-Server.

Volltextindizierung und Suche

Volltextindizes erlauben Ihnen die Suche nach bestimmten Wörtern und Sätzen in Feldern, die beliebigen Text enthalten. Dabei wird Ihnen auch die Relevanz der Ergebnisse angezeigt. In Version 4.0 haben wir die Volltextsuche um genaue Satzvergleiche und Boolesche Operatoren erweitert. Dies ermöglicht Ihnen eine noch feinere Steuerung Ihrer Suchergebnisse.

Eingebettete Datenbankbibliothek

Mit Verwendung der eingebetteten Datenbankbibliothek (libmysqld) können Sie die volle Leistungsfähigkeit des Datenbankservers MySQL in Anwendungen und elektronische Geräte einbinden. Der Anwender wird die darunterliegende Datenbank

gar nicht wahrnehmen. Die eingebettete Datenbank MySQL ist ideal für den Einsatz hinter den Kulissen von Internetanwendungen, öffentlichen Shopsystemen, schlüsselfertigen Kombinationen von Hardware/Software-Einheiten, hochleistungsfähigen Internetservern, selbständigen Datenbanken auf CD-ROMs und anderen Möglichkeiten, die nur darauf warten, von Ihnen entdeckt zu werden.

(Quelle: MySQL Handbuch)

2.5. Zielstellung des Projekts

Die in einem gastronomischen Betrieb auftretenden Abläufe basieren zu einem Großteil auf dem Einsatz von Personal. Die typische Abfolge Auswählen-Bestellen-Verzehren-Bezahlen erfordert bei jedem Teilprozess die Signalisierung des zuständigen Kellners und dessen Anwesenheit. Die interne konventionelle Kommunikation zwischen Servierpersonal, Küche und Tresen sowie die fehlenden Möglichkeiten zum Einsatz multimedialer Techniken stellen weitere Nachteile im Tagesgeschäft dar. (siehe Istanalyse)

Dieses Projekt befasst sich mit dem Einsatz von Informationstechnik in gastronomischen Betrieben. Es wird untersucht, wie die mittlerweile sehr günstig verfügbaren technischen Möglichkeiten in einem Restaurant, einer Bar u.a. eingesetzt werden können.

Hierbei geht unsere Idee nicht nur so weit, das Personal entsprechend technisch auszurüsten und dessen Kommunikation über elektronische Wege ablaufen zu lassen. Es soll vielmehr auch dem Gast ermöglicht werden, mit an den Tischen fest installierten Endgeräten (oder in einer zweiten Ausbaustufe mit mitgebrachten PDAs oder Notebooks) auf das Angebot des Gastronomiebetriebes zuzugreifen, direkt Bestellungen an die Küche zu senden, die Wartezeit mit Chatten oder Surfen im Internet zu überbrücken sowie bei Wünschen wie dem nach Begleichen der Rechnung direkt mit dem Personal in Verbindung zu treten.

Ideen zu Anforderungen

Art des gastronomischen Betriebes

Das zu erstellende System soll in einer Vielzahl von gastronomischen Betrieben einsetzbar sein. Es soll keine Rolle spielen ob es sich um ein Spezialitätenrestaurant, eine Eckkneipe oder ein Schnellrestaurant mit Selbstbedienung handelt. Die Zugangsmöglichkeiten für das Personal und die Gäste sind entsprechend flexibel zu halten.



Ideen zur eingesetzten Technik

Die Menge der eingesetzten Technik ist gering, um eine preiswerte Realisierung des Systems zu ermöglichen.

Grundsätzlich erhält der gastronomische Betrieb einen Hauptrechner, der die Dienste des Betriebes bereitstellt. Je mindestens ein Rechner für den Kellner und das Küchenpersonal stellen auf betrieblicher Seite den Zugang zum System dar, wobei in einfachen Varianten auch o.g. Hauptrechner als Zugangspunkt verwendet werden kann. Jeder Tisch bzw. Tresen erhält hiernach einen Stift bedienten Kleinrechner (PDA), über den die Gäste Bestellungen aufgeben oder kommunizieren.

Um in bestehenden Gastronomiebetrieben auf umständliche Verlegung einer konventionellen Netzwerktopologie verzichten zu können, kommt drahtloses Netzwerk (Wireless LAN, WLAN) zum Einsatz. Datenverschlüsselung sichert diesen Kommunikationsweg.

Schlussendlich dient ein zentraler Datenbankrechner zur Haltung des Angebotes, der Bestellungen und der betriebswirtschaftlichen Daten. Diese Datenbank kann für einen Einzelbetrieb oder, an zentralem Standpunkt stehend und mit permanentem Netzzugang ausgestattet, für eine Kette von Betrieben eingesetzt werden. Aufgrund der Leistungsfähigkeit und der Datenbankstruktur kann beinahe jegliche Anzahl von Filialen realisiert werden.

Personal

Die Anforderungen an das Personal werden möglichst gering gehalten. Auch ohne tief greifende Kenntnisse im Bereich der Computertechnik oder der Bedienung von Software ist eine schnelle und sichere Bedienung des Systems zu ermöglichen. Auch wird berücksichtigt, dass die Abläufe in Gastronomiebetrieben, insbesondere in Spitzenlastzeiten, es erfordern, dass das System sehr schnell und effizient bedient werden kann. Dies betrifft nicht nur die Schnittstelle zum Personal sondern auch die Erscheinungsform für den Kunden.

Prototypischer Ablauf

Der bereits in der Istanalyse beschriebene prototypische Ablauf sieht mit Einsatz unserer Lösung wie folgt aus:

Ein Gast betritt das Restaurant und setzt sich an einen freien Tisch. Der am Tisch installierte PDA begrüßt ihn und bietet Einblick in die Speise- und Getränkekarte. Der Gast wählt aus dem Angebot ein Getränk und bestellt es durch einen Tipp mit dem Stift. Er möchte vorerst nichts Weiteres bestellen und schließt die Bestellung mit einer Bestätigung ab. Nun wird die Bestellung in der zentralen Datenbank als "bestellt" abgelegt. Die für die Bereitung des Getränkes zuständige Person („Mann hinter dem Tresen“) bekommt auf seinem Terminal eine Meldung über die Bestellung und stellt das Getränk bereit. Nachdem er dies bestätigt hat, bekommt der entsprechende Kellner eine Meldung, an welchen Tisch er das bereit stehende Getränk liefern muss und bringt es dem Gast. Analog hierzu bestellt der Gast nach einiger Zeit ein Essen, welches dem Koch signalisiert wird. Es wird gekocht, bereit gestellt, dem Kellner signalisiert und ausgeliefert. Letztendlich ist der Gast zufrieden gestellt, er zeigt mit einem Klick auf den PDA an, dass er bezahlen möchte. Der entsprechende Kellner erhält hierüber eine Meldung, drückt die Rechnung über die Verzehrungen und nimmt wie aus herkömmlichen Gastronomiebetrieben bekannt die Bezahlung entgegen.



2.6. Ziel- und Kundengruppen

Das zu erstellende System richtet sich an eine möglichst große Zahl von Kunden. Der Einsatz des Systems soll insbesondere nicht davon abhängen, welche Art des gastronomischen Betriebes vorliegt, welche Technik bisher sich bisher im Einsatz

befindet, welche Kundengruppe vorwiegend den Betrieb besucht oder welche informationstechnische Vorbildung das eingesetzte Personal hat. Da das zu erstellende System intensiv aktuelle Technologien und Medien wie z.B. PDAs, Internet, MP3 nutzt, besteht der Einsatzbereich der Software insbesondere in folgenden Gastronomiebetrieben:

- Internetcafés, da dort ohnehin informationstechnische Infrastruktur vorhanden ist und die Gäste durchweg mit der Bedienung eines Web-Systems vertraut sind
- Bars und Restaurants mit vorwiegend jungem Publikum, d.h. Cocktailbars, so genannte Szene- und In-Kneipen, Themencafés und –bars
- Einrichtungen, in denen besonderen Wert auf Effizienz und kurze Wartezeiten bei der Bedienung gelegt wird, insbesondere Schnellrestaurants, Einrichtungen mit überwiegend touristischer Kundschaft oder auch moderne Firmenkantinen

Grundsätzlich lässt sich dieses System jedoch in jeglicher Art von Gastronomiebetrieb einsetzen und flexibel an die Anforderungen und die Corporate Identity des Betriebes anpassen.

2.7. geplante Funktionalitäten

für den Gast/Kunden:

- Speisekarte einsehen
- Getränke, Speisen bestellen
- Rechnung anfordern und bezahlen
- Nachrichten, Fragen an Gäste bzw. Personal verschicken
- Nutzen von Internetdiensten wie z.B. Hotelreservierung, Fahrplan, News,...
- Radio hören am Platz mittels Kopfhörer
- Musikwünsche an Jukebox in Gaststätte Jukebox senden
- Spiele auf dem PDA spielen

für den Kellner:

- Getränkebestellungen entgegennehmen
- Rückmeldung über Ausschank
- Nachrichten, Fragen an Gäste schicken
- fertig gestellte Speisen und Getränke ausliefern
- Bezahlwünsche entgegennehmen
- Rechnungen ausstellen und kassieren
- Sonderangebote in Angebot einpflegen
- Ablösung, Personalwechsel

für den Koch:

- Speisenwünsche entgegennehmen
- Rückmeldung über Zubereitungsstatus geben
- Nachrichten, Fragen an Gäste schicken
- fertig zubereitete Speisen an Kellner melden
- Ablösung, Personalwechsel
- Speiseangebot beeinflussen (z.B. Produktbestand)

Verwaltungs- /Managementaufgaben:

- Speisekarte erstellen und verwalten
- Sonderangebote einfügen
- Personal und Nutzerrechte verwalten
- Statistiken und Bilanzen

2.8. Ausbaustufen

In der ersten Ausbaustufe, die zum Projektende vorliegt, ist der in einem Gastronomiebetrieb auftretende prototypische Ablauf vollständig abgebildet. Alle Funktionen rund um das Bestellen, Zubereiten, Servieren und Bezahlen von Produkten sind sowohl auf Kunden- als auch auf Personalseite realisiert. Weiterhin wurde in der ersten Ausbaustufe eine möglichst große Anzahl von (teils multimedialen) Features realisiert, die über das bloße Bestellen von Speisen und Getränken hinausgeht. Dazu gehört das Abrufen von (vordefinierten) Inhalten aus dem Internet wie z.B. Fahrplanauskünfte, Hotelinformationen, Nachrichten oder Fernsehprogramme, die Realisierung einer MP3-Jukebox, die vom Gast mit Musikwünschen beeinflussbar ist, die Verfügbarmachung von Radiosendern am Platz des Gastes sowie das Versenden von Messages zwischen Gast und Gast bzw. Gast und Personal.

Auf der Seite des Managements wurden die wichtigsten Verwaltungsaufgaben umgesetzt. Dazu gehört das Editieren des Angebots der Gaststätte, das Erzeugen einer einfachen Umsatzstatistik sowie das Verwalten der Jukebox.

In einer zweiten Ausbaustufe werden sämtliche Managementfunktionen realisiert, so z.B. das Erstellen von Sonderangeboten, die Ablösung von Personal und differenzierte Statistiken. Weiterhin ist die Einbindung von auf Flash basierenden Spielen in das Kundeninterface vorgesehen.

In einer dritten Ausbaustufe wird schließlich die Dynamik der Geschäftsprozesse verfeinert, so dass beispielsweise Warenbestände die Angebotskarte beeinflussen ("Eisbein is aus"). Speise- und Getränkeangebot oder auch Tischbelegungen und Anzahl freier Plätze werden über das Internet oder externe Displays kommuniziert.

In einer weiteren Ausbaustufe ist eine Anbindung an betriebswirtschaftliche bzw. Warenwirtschaftssysteme geplant.

2.9. Aufgabenverteilung

(die "Hauptaufgaben" sind jeweils kursiv hervorgehoben)

Jens Klinker:

generelle Systemarchitektur

Nutzeraufträge und logische Abläufe im System

Hauptrealisierung der Anwendungslogik (PHP)

Realisierung der Oberfläche für Rechnung, Bezahlung

Mitarbeit an weiteren Details der Oberfläche PDA

Daniel Prusseit:

Istanalyse und Projektziele

Installation und Sicherung des Wireless LAN

Einrichten des PDA

Mitarbeit bei Realisierung der Anwendungslogik (PHP)

Internetinhalte und Radio auf dem PDA

Mitarbeit an einzelnen Aspekten der Oberfläche PDA

Verwaltung und Pflege der Dokumentation

Jens Rekow:

Installation und Administration der Server

Datenbankstruktur und –realisierung

Datenbankadministrator

Aufsetzen eines MP3-Servers

Funktionalität Jukebox auf PDA und Server

Mitarbeit an Oberfläche PDA und Server

Stephan Ziep:

Installation und Administration der Server

Vorbetrachtungen zu Oberflächen

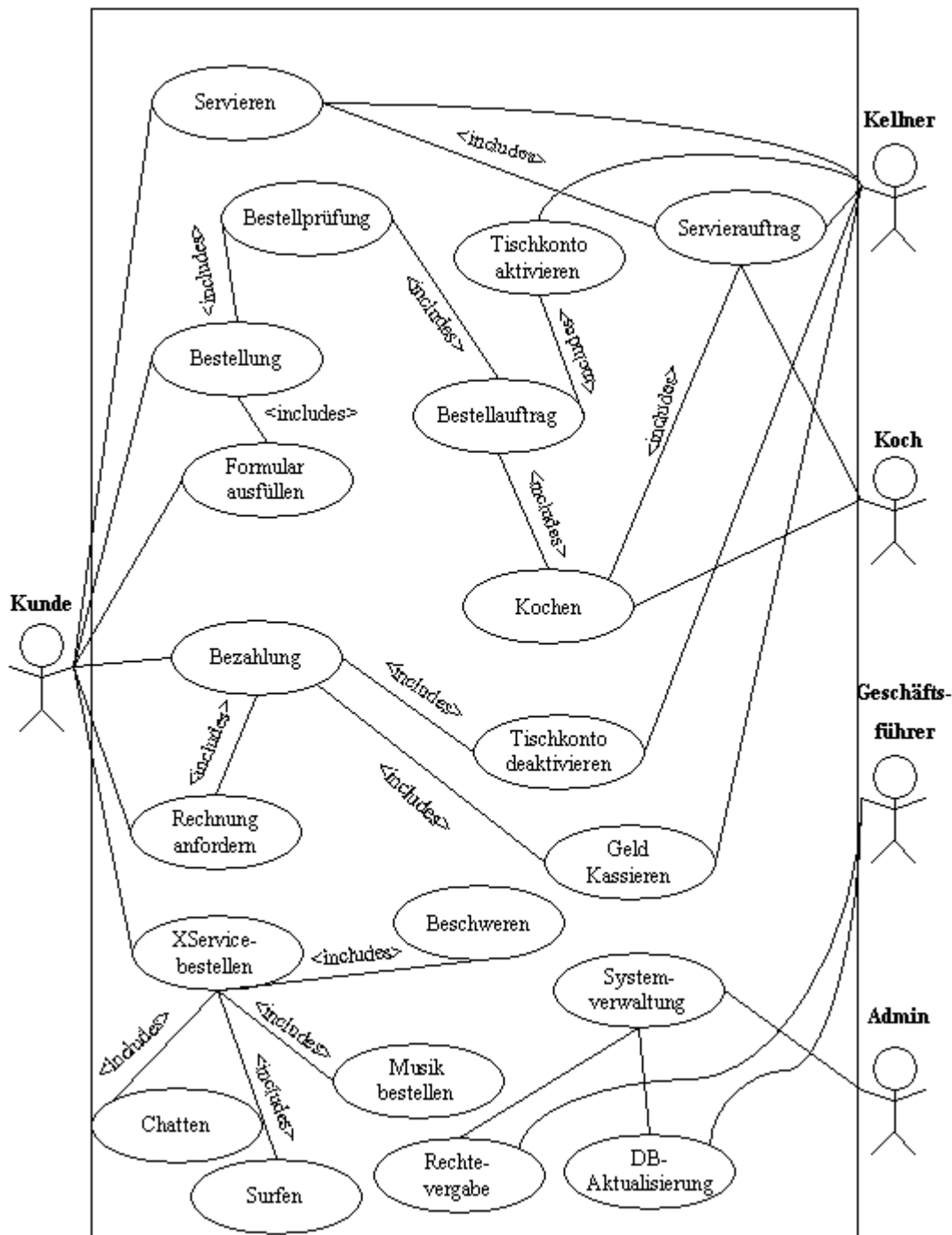
Konzept der Oberfläche

Realisierung der Oberfläche PDA (PHP)

Realisierung des Personal Interfaces

3. Design

3.1. Use Cases (Diagramm von Edward Date1)

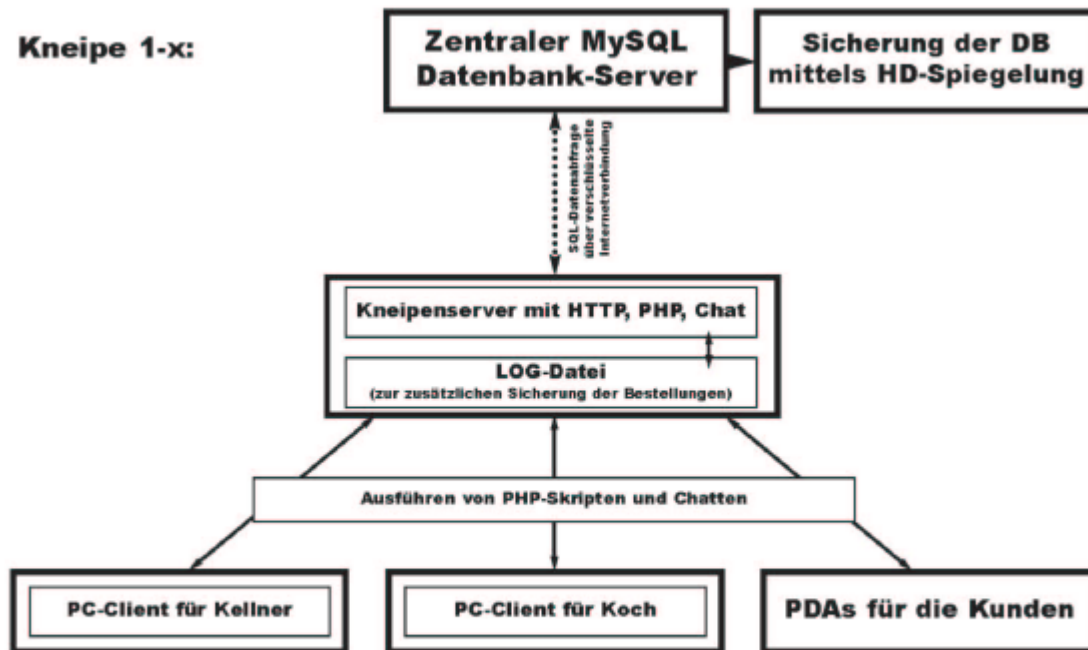


3.2. beteiligte Akteure

3.4. n-Tier

Wir haben uns für eine 3-Tier-Applikation entschieden. Dadurch haben wir eine saubere Trennung von zentraler Datenbank, Applikationsserver und GUI. Die GUI soll plattformunabhängig auf allen Systemen laufen. Weiteres hierzu folgt unter dem Aspekt der Systemarchitektur.

3.5. Systemarchitektur



1. Zentraler Datenbankserver:

Es soll für alle Kneipen einen zentralen SQL-Datenbankserver geben, auf dem alle Speisen und Getränke gespeichert werden. Unter der Berücksichtigung von Kostengesichtspunkten und der schnell geplanten Umsetzung des Projektes erscheint uns bei der Größe des Projektes MySQL als sinnvollste SQL-Datenbank. Der Zugriff auf die Datenbank soll über eine gesicherte Internetverbindung erfolgen. Dadurch lässt sich die Gefahr von Angriffen aus dem Internet minimieren. Damit die wichtigen DB-Daten gesichert werden, soll die Festplatte gespiegelt werden.

2. Kneipenserver mit PHP, HTTP, Chat:

- Der Zugriff auf die Datenbank soll durch einen einfachen Applikationsserver erfolgen. PHP ist hierfür die kostengünstigste und am einfachsten handhabbare Variante. Es muss eine zuverlässige Internetverbindung in jeder Kneipe vorhanden sein, da sonst kein Zugriff auf die zentrale Datenbank möglich ist.
- Das GUI lässt sich relativ flexibel mit HTML erstellen und ist dadurch auch automatisch plattformunabhängig, weil es mit jedem beliebigen Web-Browser angezeigt werden kann. Hierfür ist ein HTTP-Server notwendig
- Für die Chat-Funktion bietet sich aber an, einen einfachen HTML-, Java-, oder RMI-Chat zu verwenden.

3. LOG-Datei

Falls die Verbindung zur zentralen Datenbank im laufenden Geschäftsbetrieb abbrechen sollte, lässt sich nur schwer nachvollziehen, welcher Kunde was bestellt hat. Aus diesem Grund sollen alle Geschäftsprozesse zusätzlich in einer LOG-Datei protokolliert werden.

4. Client für Kellner:

Der Kellner muss einen PC besitzen, mit dem er seine seine Geschäftsprozesse abwickeln kann (z.B. bestellte Speisen liefern, Kundenwünsche und Bezahlungen entgegennehmen etc.)

5. Client für Koch:

Der Koch muss einen PC besitzen, mit dem er seine seine Geschäftsprozesse abwickeln kann (z.B. bestellte Speisen zubereiten, Speisekarte bearbeiten, Kellner informieren etc.)

6. PDAs für Kunden:

Für die Kunden soll es in jeder Kneipe fest vorinstallierte PDAs geben. Damit lassen sich Bestellungen aufgeben, im Internet surfen, mit anderen Menschen in der Kneipe Chatten etc.

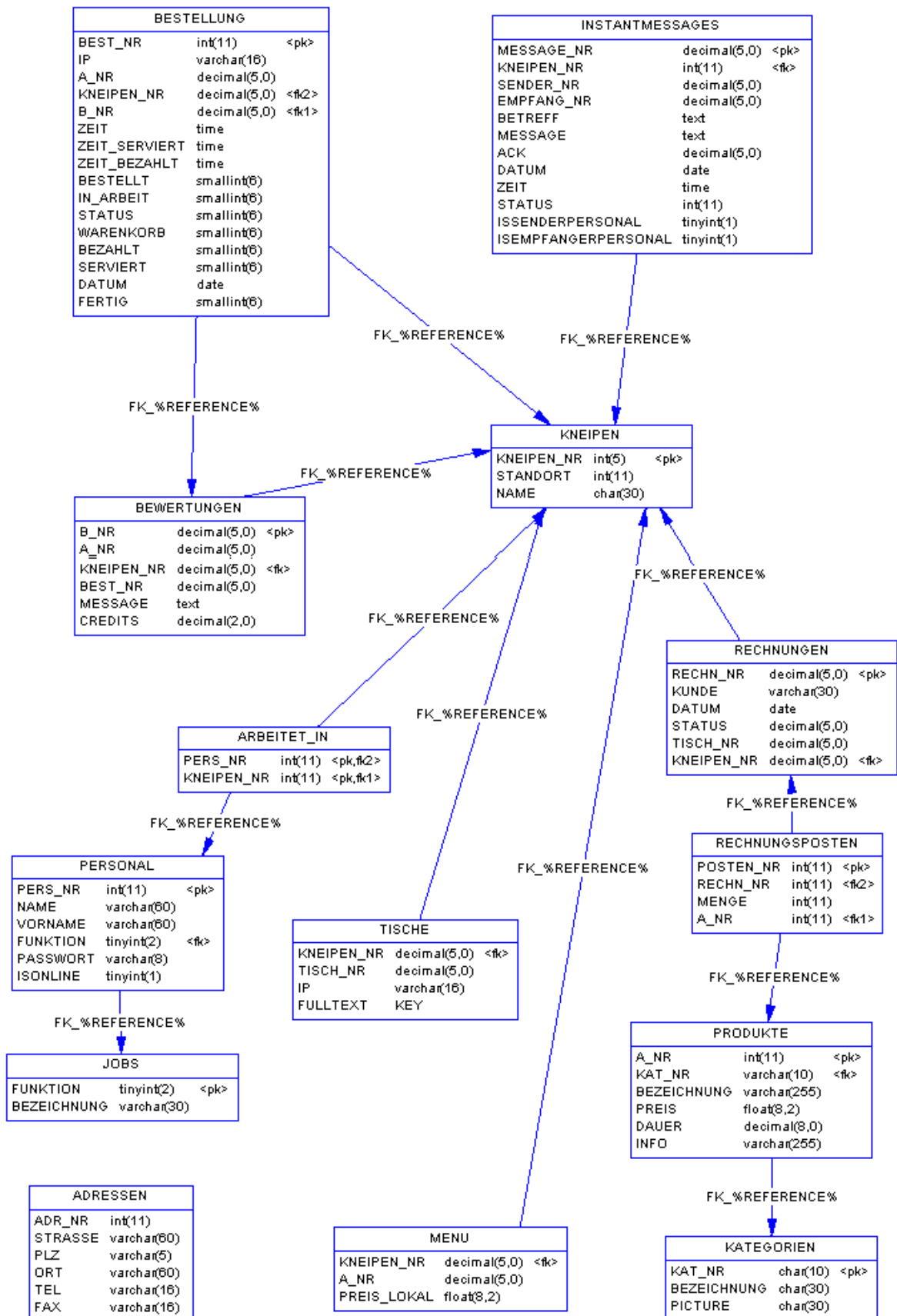
7. Hardwarebedarf:

Es wird ein Linux-Server für die zentrale Datenbank benötigt.

Pro Kneipe:

- 1 PC als Kneipenserver (PHP, HTTP und Chat), LOG-Datei
- 1 PC als Client für Kellner
- 1 PC als Client für Koch
- Pro Kneipentisch werden 2 PDA benötigt
- Alle Rechner arbeiten mit Wireless LAN, es werden also Karten für die PCs (3 je Kneipe) in den Kneipen benötigt und 1 Access Point (je Kneipe).

3.6. Datenbankstruktur



Die Entitäten **BESTELLUNG**, **KNEIPEN**, **RECHNUNGEN**, **RECHNUNGSPOSTEN**, **PRODUKTE** und **MENU** bilden die primären Geschäftsdaten in der Datenbank ab. **PRODUKTE** enthält sämtliche Artikel, die in den jeweiligen Filialen des Gastronomiebetriebes angeboten werden können. Dazu gehören neben Speisen und Getränken auch angebotene Extras, wie zum Beispiel Musiktitel, Radioprogramme und Informationsangebote im Internet.

Neben Bezeichnung und Zusatzinfos des Produktes ist hier auch der globale Preis definiert, welcher relevant ist, wenn die Filiale keinen lokalen Preis festsetzt. Ein lokaler Preis kann beispielsweise ein Sonderangebot oder auch ein nachfragebedingter höherer Preis sein.

Lokale Preise festlegen erfolgt mittels der Tabelle **MENU**. Diese spezifiziert das spezielle Angebot einer einzelnen Filiale, eine Teilmenge aus der Produkt-Entität, wobei ein hier gesetzter Preis den globalen Preis überschreibt.

Das Kernstück der Geschäftsdaten ist die Tabelle **BESTELLUNG**. Jedes bestellte Produkt erscheint hier als ein Tupel, welches über die Attribute 'Kneipen_Nr' und 'IP' die Zuordnung zur Kneipen und zum Tisch gewährleistet. Eine gleichzeitig bestellte höhere Anzahl des selben Produktes wird als ein Tupel abgebildet, die bestellte Anzahl gibt das Attribut 'Warenkorb' an. Im Verlauf der Bearbeitung der Bestellung kann nun ein Teil oder auch die gesamte bestellte Anzahl einem neuen Status, beispielsweise nach dem Absenden des Warenkorbes nach 'bestellt' oder nach dem Servieren als 'serviert', zugeführt werden. Die Attribute 'bestellt', 'in_Arbeit', 'fertig', 'serviert' und 'bezahlt' ermöglichen die Abbildung des "Lebenslaufes" eines Produktes in der Datenbank.

Durch diese Informationen, welcher Teil einer Bestellung welchen Status hat, ist sichergestellt, dass auch größere Bestellungen eines Produktes in Teilen vom Koch fertiggestellt, vom Kellner serviert oder vom Kunden bezahlt werden können.

Um die Rechnungslegung für den Kunden möglichst flexibel gestalten zu können, ist diese durch zwei weitere Entitäten realisiert. Pro Bezahlwunsch gibt es eine separate Rechnung und somit den Eintrag eines Tupels in die Tabelle **RECHNUNGEN** mit Rechnungsnummer, Kundennamen, Datum, Tisch- und Kneipennummer. Die eigentlichen Bestandteile der Rechnung setzt sich aus Posten der Tabelle **RECHNUNGSPOSTEN** zusammen. Jeder Posten stellt einen Artikel in einer bestimmten Menge dar und ist einer Rechnung zugeordnet.

Von zentraler Bedeutung für die Struktur der Anwendung ist die Tabelle **KATEGORIEN**, da hier alle Knoten, sprich Funktionen der Anwendungsoberfläche abgelegt sind, aus denen sich somit auch die baumartige Navigationsstruktur der Oberfläche herleitet. Mittels einer speziellen Notation aus Ziffern, welche den Knoten in einer Ebene angeben, und Underscores, die in die nächst-tiefere Ebene des Baumes führen, wird diese Baumstruktur in der Datenbank abgebildet. Der Eintrag '1_2_4' bezeichnet demnach den vierten Knoten in der dritten Ebene. Dahinter verbirgt sich in diesem Fall die Rubrik 'Spiele' im Gast-Interface.

Alle Nachrichten des Messaging Systems werden ebenso in der Datenbank unter **INSTANTMESSAGES** gehalten. Da die Gäste nicht nur untereinander sondern auch mit dem Personal kommunizieren können, erforderte die Realisierung der internen Adressierung der Textmitteilungen etwas mehr Aufwand. Kommunizieren Gästen untereinander, so bilden die jeweiligen Tischnummern die Sender- und Empfängeradressen. Sollte auch das Personal, also Kellner oder Koch, mit einbezogen werden, so kommen die Attribute 'issenderpersonal' beziehungsweise 'isempfängerpersonal' zur Wirkung. Sind diese Flags auf 'true' gesetzt, so symbolisieren die Sender- und Empfängeradressen keine Tischnummern, sondern

werden als Personalnummern interpretiert und sorgen somit für die eindeutige Zuordnung.

Die übrigen Tabellen, wie beispielsweise **ADRESSEN**, **JOBS**, **PERSONAL** und **ARBEITET_IN** dienen vorrangig administrativen Zwecken. Die Ablage auch dieser Daten in der Datenbank unterstützt die Verwaltung des Gastronomiebetriebs und führen zu größerer Flexibilität bei der Applikation und Skalierung der Anwendung.

3.7. Anforderungen an die Oberfläche

Die Anforderungen an das User Interface unterscheiden sich für die Zielgruppen Gast und Personal.

Gast

Für den Gast steht einfache Bedienbarkeit an erster Stelle. Nicht jeder ist es gewohnt, mit einem so kleinen PDA Bildschirm umzugehen. Die Navigation soll so einfach wie auf einer normalen Webseite funktionieren. Damit kann heutzutage fast jeder intuitiv umgehen, und es ist relativ einfach zu erstellen.

Damit sich der Benutzer zurecht findet auf den verschiedenen Seiten des Systems, muss es Bereiche geben, die sich nicht verändern. Diese statischen Bereiche enthalten die Navigation und einige Zusatzinformationen (Anzahl neue Nachrichten,...). Er gibt auch Auskunft darüber, in welcher Navigationsebene sich der Benutzer gerade befindet. Um die Performance des PDA und die Netzwerkbandbreite nicht unnötig zu belasten, wurden zudem auf unnötige Grafiken verzichtet. Wo es vertretbar war, wurden jedoch Symbole benutzt, um die Navigation zu erleichtern. Z.B steht vor dem Menüpunkt „Getränke“ ein Symbol mit einem Getränk. So weiß der Gast, was ihn hinter den verschiedenen Menüpunkten erwartet.



Hauptmenü Gastzugang auf PDA

Die Navigation ist dem kleinen PDA Bildschirm angepasst. Man bekommt die komplexe Navigationshierarchie in kleinen Häppchen vorgesetzt. Der Gast hat alles immer auf einen Blick. Überdies entspricht auch die hierarchische Ordnung des Menus, der menschlichen Neigung zur Kategorisierung, was wiederum ein vereinfachtes Zurechtfinden zur Folge hat.

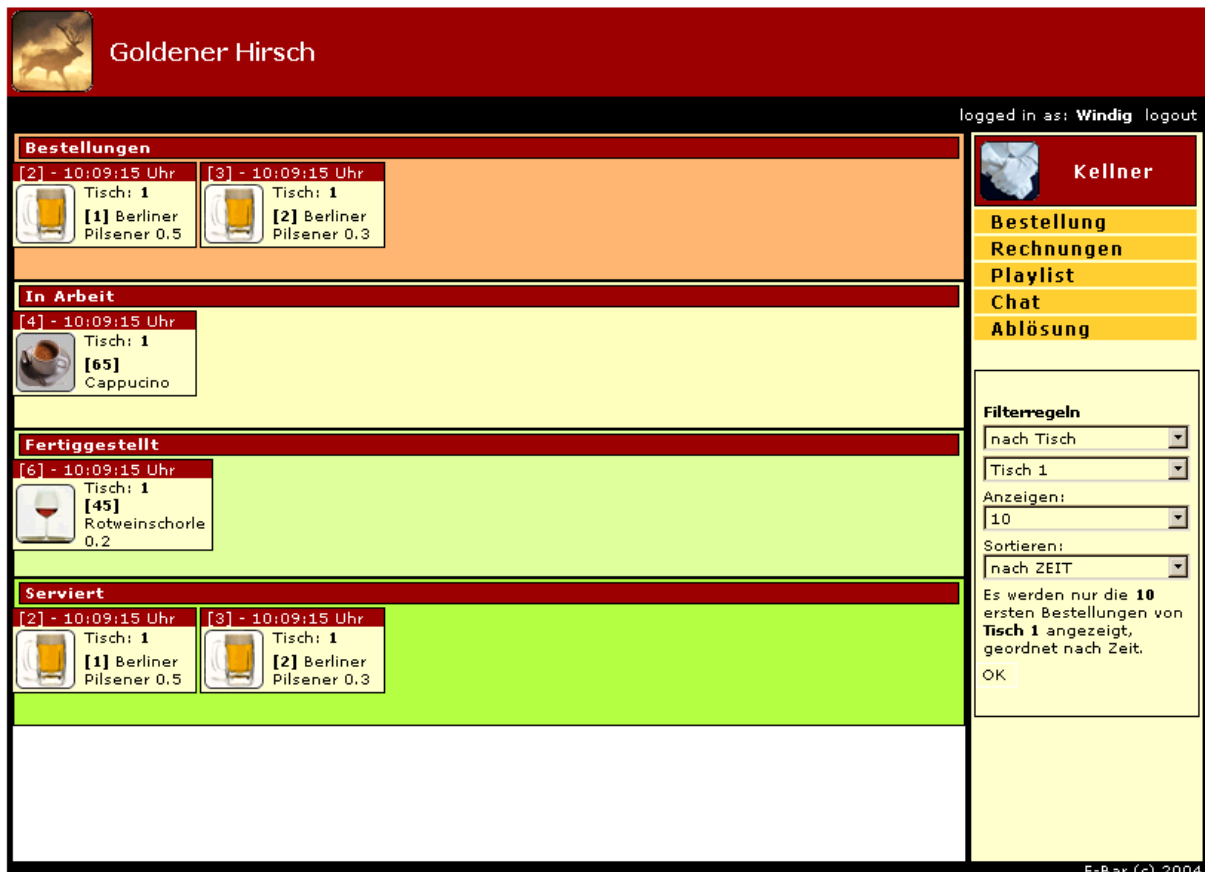
(Getränke->Softdrinks->Cola)

Personal

An das Personal können wiederum andere Anforderungen und Voraussetzungen gestellt werden. Hier kommt es nicht auf eine intuitive Benutzerführung und eine flache Lernkurve an, sondern um effiziente Bedienung des Systems. Aus diesem Grund wird auch die komplette Menuleiste auf einen Blick dargestellt. Das Personal muss sich also nicht erst bis zum dritten Unterpunkt durchkämpfen, sondern kann mit

einem Klick auf dem gewünschten Menüpunkt sein. Häufig genutzte Menüpunkte befinden sich weiter oben.

Das User Interface sieht bei jedem Mitarbeiter gleich aus, lediglich die Menüpunkte unterscheiden sich je nach Aufgabenbereich. So bekommt jeder nur das zu sehen, was er für die unmittelbare Ausführung seiner Aufgaben braucht. Z.B der Kellner sieht auf seinen Bestellungen keine Speisen, bis sie vom Koch als „Fertig zum Servieren“ gekennzeichnet hat. Im Gegenzug erscheinen beim Koch keine Getränke-Bestellungen auf dem Display.



Nutzeroberfläche des Kellners – Status von Getränkebestellungen

Auch die Art der Bedienung ist dem Arbeitsumfeld angepasst. So kann der Koch seine Bestellungen per Drag & Drop annehmen, da sich ein Touchscreen in der Küche besser macht als der fummelige Umgang mit Maus und Tastatur.

3.8. Abbildung der Geschäftsprozesse

Im einzelnen müssen die konkreten Kundenwünsche und Geschäftsprozesse in Klassen abgebildet werden. Wir haben dies anhand der Rechnungen einmal exemplarisch dargestellt. Eine Übersicht der entstandenen Klassen findet sich unter 3.10.

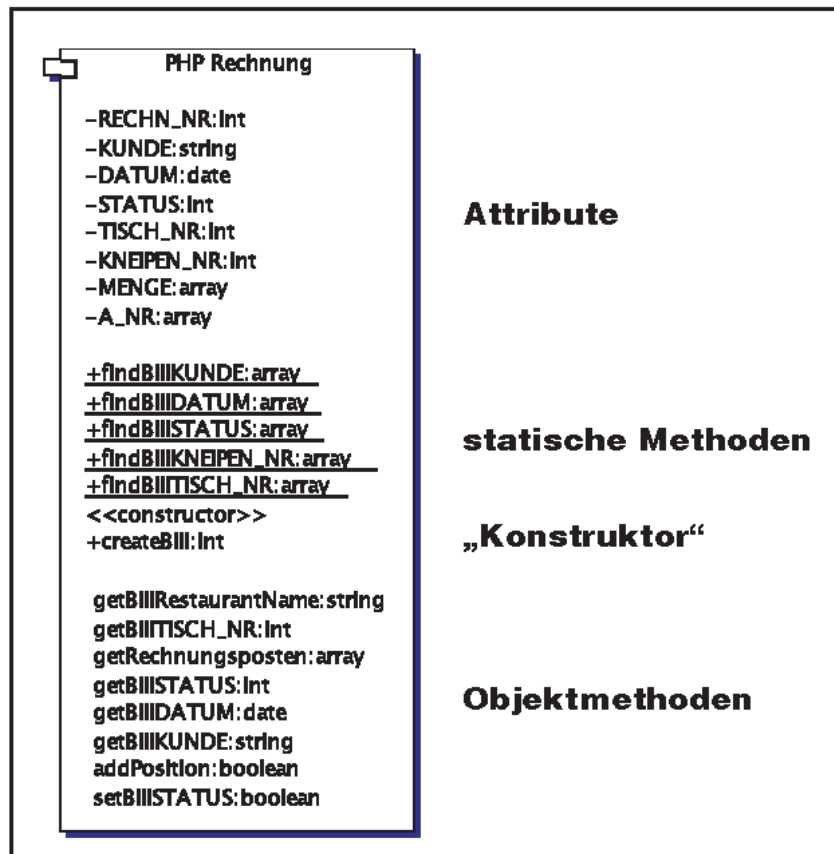
Zuerst überlegen wir uns, über welche Attribute eine Rechnung verfügen soll:

- eindeutige Rechnungsnummer
- Kundenbezeichnung
- Rechnungsdatum
- Status

- Kneipennummer
- Tischnummer
- mehrere Rechnungsposten mit je einer Artikelnummer und einer Menge.

Diese Attribute haben wir auch in unsere Datenbank in den Tabellen **RECHNUNGEN** und **RECHNUNGSPOSTEN** übernommen.

Beispiel-Klassendiagramm zur Klasse **RECHNUNGEN**:



Es muss die Möglichkeit geben, diese Attribute z.B. für ein Rechnungsformular abzufragen. Dafür gibt es folgende Methoden:

```

function getBillDATUM ($RECHN_NR)
function getBillKUNDE ($RECHN_NR)
function getBillRestaurantName ($RECHN_NR)
function getBillTISCH_NR ($RECHN_NR)
function getRechnungsposten ($RECHN_NR)
function getBillSTATUS ($STATUS)

```

Welche Geschäftsprozesse sollen mit der Klasse Rechnungen abgebildet werden?

1. Kunde fordert Rechnung an:
 - Erstellen der Rechnung: **function createBill**
 - Hinzufügen von einzelnen Positionen zur Rechnung: **function addPosition**
2. Der Kellner bestätigt Rechnung und druckt diese aus:
 - Der Rechnungsstatus muss auf 1 gesetzt werden: **function setBillSTATUS**

3. Es sollen Rechnungen nach bestimmten Kriterien gesucht werden:

```
function findBillKUNDE ($KUNDE)
function findBillDATUM ($DATUM)
function findBillSTATUS ($STATUS)
function findBillKNEIPEN_NR ($KNEIPEN_NR)
function findBillTISCH_NR ($KNEIPEN_NR, $TISCH_NR)
```

3.9. Objektorientierung in PHP?

Objekte sind in PHP nicht persistent, sondern existieren nur bis zur Bearbeitung einer Benutzeranfrage. Es gibt aber die Möglichkeit, Objekte persistent zu halten: PHP bietet eine Funktionalität, mit der Variablen während einer gesamten Session persistent gehalten werden können. Dabei wird die Variable auf dem PHP-Server gespeichert und über Cookies dem Client zugeordnet. Statt Variablen können so auch Objekte gespeichert werden. Näheres zu Sessions in PHP in Punkt 4.5.1.

Das Problem ist jedoch, dass die Objekte nach jeder Änderung wieder in die Sessionvariable gespeichert werden müssen und mit jeder Benutzerinteraktion muss die Sessionvariable wieder neu abgefragt werden.

Da die Daten ohnehin in einer zentralen Datenbank gespeichert sind, haben wir uns für ein anderes Konzept entschieden. Wir arbeiten formell mit statischen Klassen, simulieren aber Objekte durch den Primärschlüssel der entsprechenden Tabelle:

- Statische Methoden (im Modell die unterstrichenen) haben wir direkt aus dem Klassenmodell in unsere PHP-Klassen übernommen.
- Pro Klasse gibt es einen Art Konstruktor, der ein Tupel in der Datenbanktabelle erzeugt und den Primärschlüssel zurück gibt (beim Beispiel Rechnungen: **createBill**). Dieser Primärschlüssel dient für uns in PHP als Verweis auf das simulierte Objekt.
- Objektmethoden aus unserem Klassenmodell werden von uns in PHP wie statische Methoden aufgerufen. Durch die Übergabe des Primärschlüssels der entsprechenden Tabelle an die statische Methode kann mit Hilfe der Datenbank auf alle Attribute z.B. einer Rechnung zugegriffen oder diese manipuliert werden. Damit ähneln diese Methoden sehr stark den Objektmethoden aus dem Klassenmodell.

Die Attribute der einzelnen Rechnung z.B. Rechnungsdatum werden deshalb bei jedem Methodenaufruf aus der Datenbank geholt. Als Identifikator einer einzelnen Rechnung dient der Primärschlüssel der Tabelle RECHNUNGEN (RECHN_NR).

3.10. Klassen der Anwendungslogik

Wir haben uns bereits in 3.4 für die Entwicklung einer 3-Tier-Applikation entschieden. Im folgenden wird der Entwurf der Klassenstrukturen für die Anwendungslogik beschrieben. Die Anwendungslogik soll mit folgenden Klassen verwirklicht werden:

- *Bestellung*

Mit dieser Klasse soll der gesamte Bestellvorgang abgewickelt werden (Vom Warenkorb bis zur Bezahlung).

- Jobs

Hierbei handelt es sich um eine Wrapper-Klasse für die Tabelle JOBS, um die einzelnen JOB-Arten zu verwalten.

- Messages

Hierbei handelt es sich um eine Wrapper-Klasse für die Tabelle Messages. Mit dieser Klasse soll die Chat/Instantmessage-Funktionalität in der Datenbank realisiert werden.

- Rechnung

Wenn ein Artikel bezahlt wird, wird eine Rechnung erstellt. Die Vorgänge des Rechnungsschreibens und des Verwaltens von Rechnungen sollen mit dieser Klasse erfolgen.

- Personal

Hierbei handelt es sich um Wrapper-Klassen für die Tabellen PERSONAL und ARBEITET_IN. Mit dieser Klasse soll das Personal und die Zuordnung Personal-Kneipe verwaltet werden.

- Produkte

Hierbei handelt es sich um eine Wrapper-Klasse für die Tabelle Produkte. Mit ihr sollen die Produkte verwaltet werden.

- DBClass

Diese Klasse regelt den Zugriff auf die Datenbank. Sie erstellt automatisch eine persistente Verbindung mit der Datenbank "kneipedb" und führt SQL-Anweisungen aus.



4. Implementierung

4.1. Auswahl der Techniken

Erzeugen von HTML-Seiten, Verwenden von JavaScript:

Auf den im Projekt verwendeten mobilen Endgeräten (PDA) ist standardmäßig ein Browser installiert, welcher HTML-Seiten anzeigt und im gewissen Umfang JavaScript verarbeitet (siehe auch 4.4.1.). Viele weitere Geräte (Handys, Smartphones,...) sind heutzutage in der Lage, HTML-Code zu interpretieren und darzustellen.

Um die Anwendung flexibel und portierbar zu gestalten, fiel die Entscheidung darauf, HTML-Seiten zu generieren. Interaktionen wie z.B. Formulareingaben und Wertprüfungen werden dabei von JavaScript unterstützt.

Programmieren mit PHP

Wie bereits in 2.4.2. beschrieben, ist PHP auf der einen Seite für Neulinge schnell zu erlernen, bietet auf der anderen Seite jedoch einen riesigen Funktionsumfang auch für professionelle Einsatzzwecke. Gründe wie freie Verfügbarkeit, einfache Datenbankbindung und Bekanntheit führten dazu, dass wir uns zur Verwendung von PHP entschieden haben.

Datenbank mit MySQL:

Aus verschiedenen Gründen fiel die Entscheidung bezüglich der permanenten Datenhaltung auf eine MySQL Datenbank. Obwohl MySQL in der aktuellsten Version den Tabellentyp "InnoDB" zur Verfügung stellt, welcher ein hohes Maß an referentieller Integrität ermöglicht, wurde in diesem Fall eher Wert auf ein schlankes, stabiles und schnelles Datenbankkonzept Wert gelegt.

Somit kam der MySQL-Standard-Tabellentyp 'MyISAM' zum Einsatz. Die Logik für Fremdschlüssel beziehungsweise referentielle Abhängigkeiten bei delete- und update-Operationen verlagern sich demzufolge in die Programmierung der Anwendung.

4.2. Hardware und Installation

Der Anwendungsserver ist ein Intel Pentium II Rechner mit 400 Mhz Taktfrequenz, 64 Megabyte Arbeitsspeicher und einer Festplattenkapazität von 6 Gigabyte.

Als Betriebssystem kommt Suse Linux 8.2 zum Einsatz. Als frei verfügbares System bedeutet dies eine enorme Kostenersparnis und bietet darüber hinaus ein höheres Maß an Betriebssicherheit als ein vergleichbares Microsoft Windows System.

Des Weiteren beinhaltet das Suse Linux 8.2 Paket zusätzlich benötigte Dienste, wie den Apache Webserver, ein PHP-Modul und einen MySQL Datenbankserver.

Der Apache Webserver ist in der Version 1.3.27 mit dem PHP - Modul der Version 4.3.1 installiert. Um ein einfaches Übermitteln von Anwendungsdaten zur Laufzeit zu ermöglichen, wurde vom Installationsstandard abweichend im PHP-Modul die Option 'register_globals' aktiviert.

Diese Konfiguration des Linux Projektserver erfüllt alle gestellten Anforderungen, welche im wesentlichen aus Datenübertragung mittels HTTP resultieren.



Da ausnahmslos Texte sowie sehr kleine Bilder übertragen werden, bleibt die Datenrate eher gering und nur die Anzahl der Anfragen sorgt für eine Belastung des Servers. Der Betrieb mehrerer Filialen mit einer gewöhnlichen Anzahl von Tischen sollte jedoch auch mit einem derart dimensionierten Anwendungsserver reibungslos erfolgen.

Den separaten Datenrechner unterscheidet nur die installierte MySQL Datenbank vom Anwendungsserver. Ansonsten ist die Hard- und Softwarekonfiguration identisch.

Bei der Wahl des Datenbankmanagement-Systems standen neben den Kostengesichtspunkten vor allem auch die Vorzüge von MySQL im Vordergrund. Dazu zählt weite Verbreitung, Schnelligkeit, Stabilität sowie unkomplizierte Einrichtung und Wartung.

Obwohl das aktuelle Production Release 4.0 schon den MySQL-Tabellentyp "InnoDB" anbietet, welcher in puncto referentieller Integrität bis hin zum Transaktionsmanagement eine weitaus größere Funktionalität besitzt, fiel die Wahl auf den MySQL-Server der Version 3.23. Dies ist Gründen der Kompatibilität geschuldet, da fast alle Anbieter von Webspace mit Datenbankunterstützung noch mit den älteren, etablierten Versionen, vor allem 3.22 und 3.23, arbeiten. Somit werden die Einsatzmöglichkeiten der E-Bar von dieser Seite her nicht eingeschränkt.

4.4. Implementierung der Oberfläche

4.4.1. Beschränkungen der Endgeräte

Die auf den eingesetzten PDA vom Typ Hewlett Packard ipaq vorhandenen Browser unterliegen etlichen Beschränkungen, die sich auf die Umsetzung des Oberflächenkonzepts auswirken. Beschränkungen sind unter anderem:

- keine Cascading Style Sheets,
- keine animierten GIF
- Refresh (automatisches Neuladen) von Seiten nur eingeschränkt möglich
- keine Layer
- JavaScript nur zum Teil möglich, z.B. kein Auffrischen von Formulardaten

4.4.2. Umsetzung (Generatoren etc.)

Bei der Umsetzung der Oberfläche hatte Priorität, dass das System ein Maximum an Flexibilität und Erweiterbarkeit bietet. Dazu gehören unter anderem die flexible Anpassung und Erweiterung von Menüpunkten und der Funktionalität. Um dies zu gewährleisten, wurde die Oberfläche als Modulsystem angelegt.

Bestandteile

Basiert auf 2 Bestandteilen:

- MenuNode
 - o Darstellung des Menüpunktes
 - o Darstellung des statischen Bereichs
 - o Verantwortlich für Look&Feel der Seiten
 - o Ermöglicht hierarchische Navigation durch Baumstruktur
 - o MenuNode Informationen kommen aus Datenbank (Kategorien)
- PageGenerator

- Liefert den Inhalt der Seiten
- Darstellung des Dynamischen Bereichs
- Grafische Darstellung und Anwendungslogik im Generator getrennt
- Wird an einen MenuNode angehängt.
- abstrakte Klasse
- Stellt die Schnittstelle für die Implementierung eines PageGenerators dar

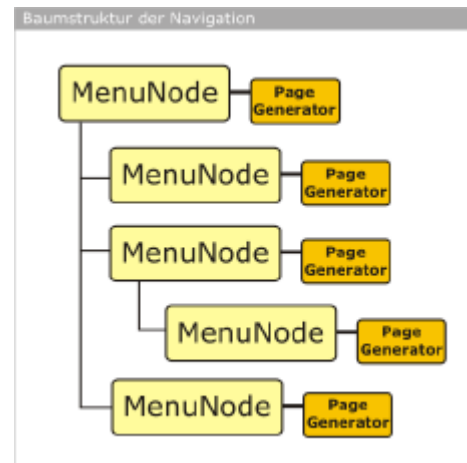
Ablauf

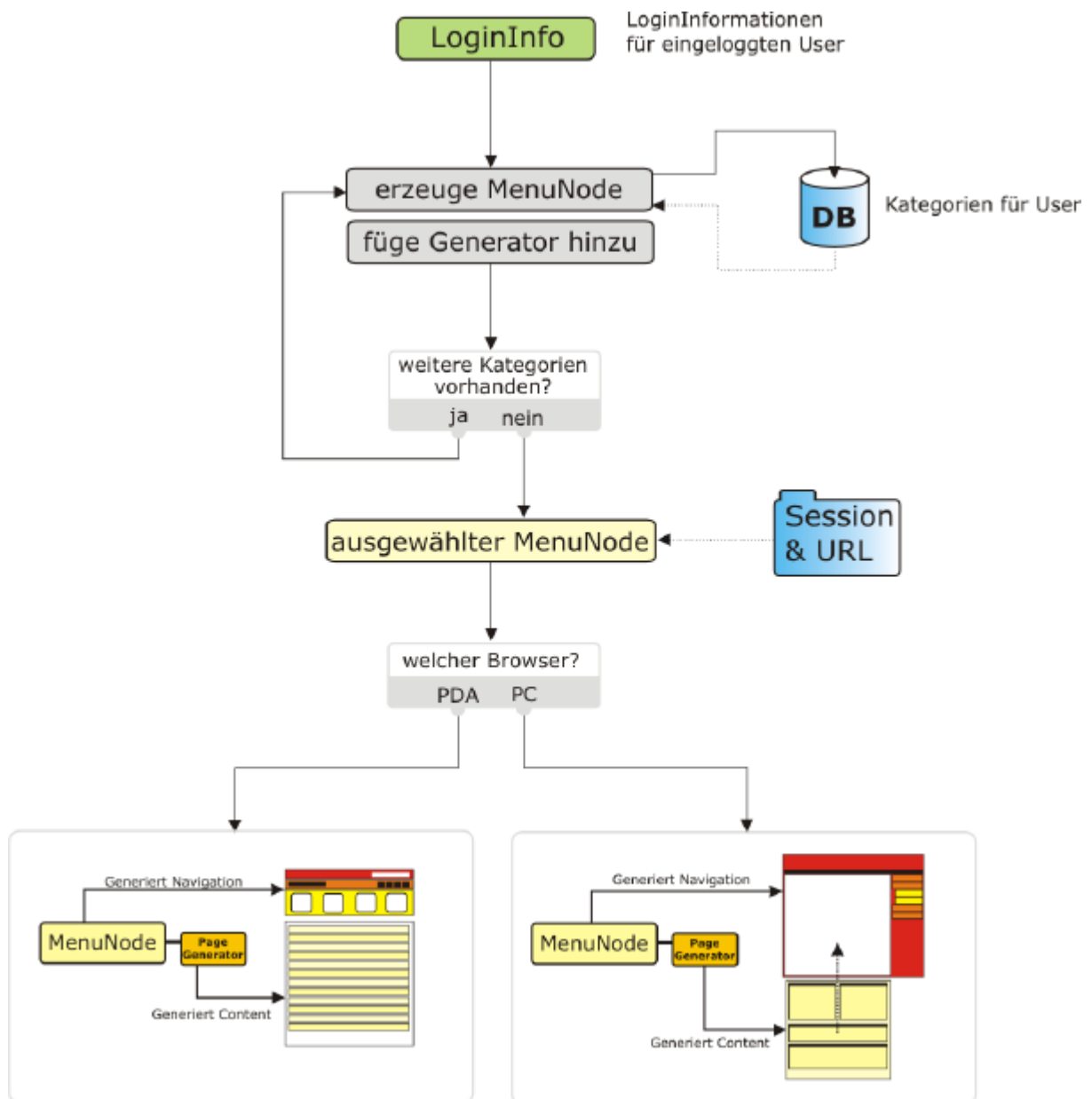
Das Personal muss sich am System anmelden. Diese Informationen werden in der Datenbank gespeichert, und nach dem Login, wird ein Session persistentes Objekt der Klasse Login angelegt, was die Daten des Benutzers dauerhaft zur Verfügung stellt.

Da die Gäste keine Login-Daten benötigen um das System zu benutzen, bekommen die Gast – PDAs eine feste Kneipennummer, IP, und Tischnummer zugewiesen. Auf diese Weise ist es möglich Bestellungen etc. genau einem Tisch zuzuordnen.

Es werden nun alle Menüpunkte für den Benutzertyp (Kellner, Koch, ...) von der Datenbank erfragt, und MenuNodes generiert. Gleichzeitig wird entschieden, für welchen Menüpunkt, welcher PageGenerator erzeugt wird. Dieser wird an den MenuNode angehängt.

Ist dies abgeschlossen, wird der aktuell gewählte Menüpunkt aus dem URL String ermittelt, und je nach Browsertyp, die Funktionen zum Erzeugen des Menüs aufgerufen. Für den PC heißt diese Funktion „drawMenuNormal“, für den PDA heißt sie „drawMenu“.





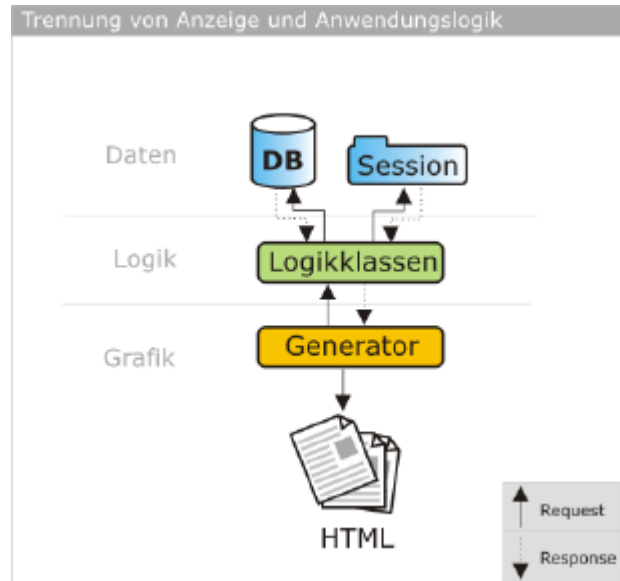
In den Funktionen zum Zeichnen des Menus, wird auf dem angehängten PageGenerator die Funktion zum Darstellen des dynamischen Bereichs aufgerufen. Das geschieht wieder je nach Browsertyp unterschiedlich.

Mit diesem System ist es sehr leicht möglich, durch Ableitung der Klasse MenuNode und Überschreibung der „DrawMenu“-Funktionen, dem System ein völlig anderes Aussehen zu verpassen.

Generatoren

Die Aufgabe des PageGenerators ist es den Code der Darstellung von der Anwendungslogik zu trennen. Dazu bedient sich er sich vor allem der Logikklassen, die die eigentliche Arbeit ausführen, und formatiert deren Rückgaben.

Alle Generatoren erben von der abstrakten Klasse PageGenerator, die Schnittstellen für den Browserabhängigen Aufruf der Generatoren zur Ausgabe definiert.



Um für eine neue Funktionalität des Systems einen Generator zu erstellen, sind lediglich folgende Schritte notwendig.

1. Die Klasse muss von PageGenerator erben und die Funktion „generatePage“ überschreiben.
2. Diese Klasse muss in der Datei „global_includes.php“ eingebunden werden.
3. Wenn ein neuer Menüpunkt hinzukommen soll, muss in die Tabelle „KATEGORIEN“ ein entsprechender Eintrag gemacht werden.
4. In der Startseite (index_gast.php oder index_service.php) muss definiert werden an welchen MenuNode dieser neue Generator angehängt werden soll.

4.5. Sicherheit

4.5.1. Session-Management

Die Anwendung macht es erforderlich, dass Benutzerdaten gehalten werden. Zum einen spielen hierbei Sicherheitsaspekte eine entscheidende Rolle, da die verschiedenen Benutzertypen ganz unterschiedliche Zugriffsrechte besitzen. Während Gäste ausschließlich auf das Angebot zugreifen können, hat das Personal auch die Möglichkeit das Angebot zu modifizieren. Darüber hinaus gibt es auch beim Personal unterschiedliche Funktionalitäten, so hat der Koch sich vorrangig mit der Abarbeitung von Bestellungen zu beschäftigen, während es dem Kellner auch gestattet ist, Rechnungen zu erstellen und die Playlist der Jukebox zu bearbeiten. Mit dem Login auf der Startseite beginnt die Differenzierung zwischen unterschiedlichen Benutzern. Hier starten dann dementsprechend auch die jeweiligen Sessions.

PHP unterstützt ab der Version 4 ein solches Session Management, wobei in erster Linie Cookies verwendet werden.

Der Befehl `session_start()` erstellt, falls noch keine erzeugt wurde, eine neue Session. Dabei wird eine Session ID generiert und ein Set-Cookie Header-Feld in die Response eingefügt. Dieses Cookie, namens PHPSESSID, speichert die auf dem Server erzeugte Session ID auf dem Client.

Für den Fall, dass im Browser Cookies deaktiviert sein sollten, wird alternativ dazu übergegangen die Daten über die URL zu übermitteln.

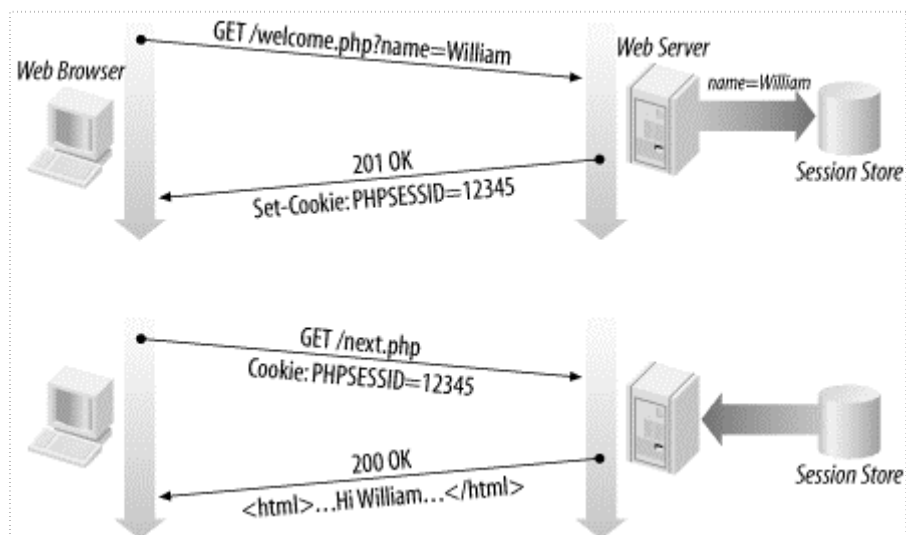
Die Session ID besteht üblicherweise aus einem Zufalls-String aus 32 Hexadezimal-Zahlen und kann über die PHP-Variablen `$HTTP_COOKIE_VARS` beziehungsweise `$PHPSESSID` abgerufen werden.

Beim erstmaligen Aufrufen von `session_start()` wird eine Sitzungsdatei angelegt, welche Sitzungsvariablen speichert. Mit dem Befehl `session_register()` kann eine solche Sitzungsvariable angelegt werden, für den Fall, dass noch keine Sitzung existiert, wird dabei automatisch `session_start()` aufgerufen.

```
// Benutzernamen "Hans" in den Sitzungsvariablen speichern
session_start();
session_register("benutzer");
$benutzer = "Hans";
session_unregister("benutzer");
session_destroy();
```

Beispiel: Speichern einer Sessionvariable

Der Befehl `session_unregister()` entfernt Variablen wieder aus dem Sitzungsspeicher und `session_destroy()` beendet eine Sitzung, sinnvoll für das Abmelden eines Benutzers.



In unserem Fall wird die Sitzung bereits beim Aufruf der ersten Seite des Managerinterfaces gestartet.

Beim folgenden Login durch einen Benutzer werden die Daten aus der Datenbank abgerufen und in der Login-Klasse gespeichert. Eine Instanz dieser Klasse wiederum wird in der Session als Sitzungsvariable abgelegt.

Somit ist es jederzeit möglich über die Variablen der Sitzung auf eine Instanz der Login-Klasse zuzugreifen, den Benutzer zu verifizieren und die Bildschirmausgabe dementsprechend anzupassen.

Mit dem Logout wird `session_destroy()` aufgerufen und die Sitzungsdatei gelöscht. Somit sind auch die Sitzungsvariablen nicht mehr verfügbar.

4.5.3. WLAN Sicherheit

Die Verwendung von drahtlosen Netzwerken in der Unternehmenskommunikation ist ohne weitere Vorkehrungen unsicher, da jedermann mittels einer Zugangskarte und Software die übertragenen Daten "mithören" kann. Deswegen erfordert es diverse Verfahrensweisen, um die Übertragung insbesondere von Geschäftsdaten vor unbefugtem Zugriff zu sichern.



Verfahren 1: Adresse für den Access Point

Mit dem Service Set Identifier (SSID) bestimmt der Administrator, auf welche Access Points Notebooks oder PCs zugreifen können. Der SSID benennt einen oder eine Gruppe von Access Points (AP). Damit der Anwender drahtlos auf das Netz zugreifen kann, müssen auf den Notebooks die entsprechenden SSIDs hinterlegt sein. Das SSID funktioniert wie ein einfaches Passwort, wobei der Name des Access Points vom Endgerät übermittelt wird.

Jeder AP ist mit einem SSID programmiert, der mit einem konkreten Wireless LAN korrespondiert. Übermittelt das Notebook nicht die korrekte Adresse des Access Points, so erhält der Anwender keinen Zugriff auf die Daten.

Schwächen:

Eine Sicherheitslücke entsteht, wenn der Access Point so konfiguriert ist dass er seine SSIDs denjenigen Notebooks, die sich anmelden wollen, per Broadcastverfahren mitteilt.

Ein weiteres Risiko besteht darin, dass die Anwender ihre Systeme selbst konfigurieren und anderen Personen die SSIDs möglicherweise mitteilen.

Verfahren 2: MAC Adressen filtern

Mit der Mac-Adresse lassen sich die Endgeräte identifizieren. Anhand der MAC-Adresse (Media Access Control) bestimmt der Administrator im Unternehmen, welche Endgeräte Daten über einen Access Point senden oder empfangen können. Um die Sicherheit in einem drahtlosen Netz zu erhöhen, lässt sich das System mit einer Liste von MAC-Adressen derjenigen Endgeräte programmieren, denen Zugriff aufs LAN gestattet ist.

Versucht nun ein Client, dessen Adresse nicht in der Liste enthalten ist, Zugang zu dem Access Point zu erlangen, wird er automatisch abgewiesen. Die Liste der Adressen muss der Administrator in jeden Access Point manuell eingeben und aktualisieren.

Schwächen:

Hoher Verwaltungsaufwand aufgrund der Adresseingabe in die Access Points, womit die Ausbaufähigkeit eines WLANs-Netzes begrenzt wird.

Verfahren 3: Code für die Datenpakete

Die WEP-Verschlüsselung (Wired Equivalent Privacy) soll die Kommunikation innerhalb eines WLANs (Wireless Local Area Network) vor Lauschangriffen schützen. Damit Access Point und Notebook miteinander Daten ver- und entschlüsseln können, benutzen sie einen identischen Code. Das Verschlüsselungssystem codiert Datenpakete bis zu 256bit. Die Codierung dient als Zugangskontrolle: Einem Notebook wird der Zugriff auf einen AP verweigert, wenn die Schlüssel der beiden Komponenten nicht übereinstimmen.

Schwächen:

Der 802.11-WLAN-Standard sieht kein Protokoll für das Key Management vor, sodass alle Schlüssel in einem Netz manuell administriert werden müssen.

Die WEP-Sicherheit ist zudem in Ad-hoc Peer-toPeer 802.11-Netzen, die keinen Access-Point benötigen, nicht verfügbar.

Profihacker konnten bereits die WEP-Verschlüsselung kurzer Schlüssel bei „Abhören“ von ausreichenden Datenmengen knacken.

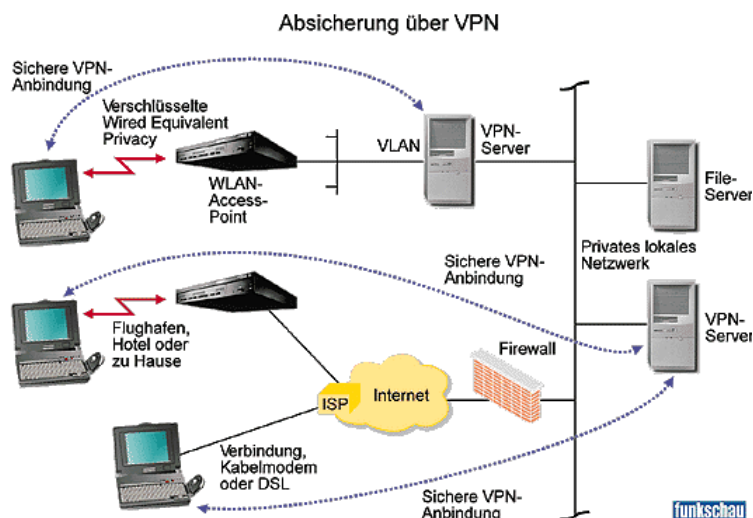
Bei Verlust eines Endgeräts muss der Administrator zudem bei allen anderen Geräten und APs den Schlüssel ändern.



Verfahren 4: VPN im WLAN

Ursprünglich bietet das VPN (Virtual Private Network) einen sicheren "Tunnel" durch das weltweite öffentliche Netz, seine Sicherheitsverfahren lassen sich aber auch auf ein lokales drahtloses Netzwerk anwenden.

Über das VPN sind beispielsweise Filialen mit ihren Mutterkonzernen über einen geschützten virtuellen Pfad verbunden. In Verbindung mit zentralisierten Authentifizierungslösungen wie RADIUS-Servern (Remote Authentication Dial-In User Service) kommen dabei verschiedene Tunneling-Protokolle wie PPTP (Point to Point Tunneling Protocol) oder L2TP (Layer 2 Tunneling Protocol) zum Einsatz.



Roaming zwischen Basisstationen unterstützt wird. WEP-Keys werden dynamisch generiert; es besteht keine Notwendigkeit, diese manuell auf Clients oder APs zu pflegen.

Ziel von 802.1X ist die Authentifizierung und Übermittlung von WEP-Schlüsseln. Die Spezifikation bietet jedoch keine allgemeine VPN-Lösung für einen sicheren Zugriff aus der Ferne. Für einen Remote-Access über öffentliche Access Points von einem Home-Office, einem Flughafen oder Hotel ist nach wie vor ein virtuelles privates Netzwerk notwendig.

(Quelle: „funkschau“ August 2001)

Im Rahmen des Projektes wurde, um generell einen unbefugten Zugriff auf unseren Access Point und die Manipulation unserer Daten zu verhindern, ein WEP Schlüssel der Länge 128bit verwendet. Durch die geringe übertragene Datenmenge ist ein Cracken dieses Schlüssels sehr zeitaufwendig und nicht ohne weiteres möglich. Weiterhin haben wir verschiedene weitere Möglichkeiten, wie beispielsweise MAC Filtering oder den Einsatz eines VPN auf Machbarkeit und Aufwand zur Umsetzung untersucht. VPN werden beispielsweise vom verwendeten Endgerät HP ipaq unterstützt.

4.6. Installation

Die Installation des Systems gestaltet sich denkbar einfach und ist binnen weniger Minuten erledigt. Voraussetzung ist die ordnungsgemäß durchgeführte Einrichtung der Server (Betriebssystem, Rechtevergabe, Netzwerk), der Endgeräte (PDA) sowie des Datenbankbetriebssystems MySQL und der Skriptsprache PHP. Beachten: In der php.ini ist der Wert "register_globals=On" zu setzen.

Sind diese Voraussetzungen erfüllt, wird das System vom Administrator mit folgenden Schritten installiert:

1. Archiv entpacken – Das gelieferte Archiv, welches die Ebar Anwendung enthält, ist auf den Gastronomieservern in ein neues Verzeichnis unterhalb des Apache Dokumentenverzeichnisses (document root) zu entpacken.
2. Datenbank erstellen – An einer MySQL Eingabeaufforderung auf dem Datenbankserver mit CREATE DATABASE eine neue Datenbank erstellen und einen Nutzer mit Zugriffsrechten auf die Datenbank anlegen.
3. Rohdatenbank einspielen – Mit der Datenbank verbinden und die mitgelieferte Rohdatenbank install.sql an der MySQL Eingabeaufforderung mittels des Befehls \. <Pfad>/install.sql einspielen.

4. Datenbank im System melden – Mit Hilfe eines Editors ist auf dem Gastronomieserver die Datei /classes/logic/DBClass.cls.php zu öffnen. An den entsprechenden Stellen sind die Zugangsdaten der in 2. angelegten Datenbank einzutragen:

```
function getConnection()
{
    $connection=$_SESSION['connection'];
    if (!$connection)
    {
        $Host = "localhost"; // Datenbankserver (IP/Hostname)
        $User = "hirsch";    // Username des DB- Nutzers
        $Pass = "kneipe";    // Passwort
        $DB   = "hirschdb";  // Datenbankname
        $connection=@ mysql_pconnect ($Host, $User, $Pass);
        mysql_select_db ($DB, $connection);
        $_SESSION['connection']=$connection;
    }
    return ($connection);
}
```

DBClass.cls.php – Datenbankverbindung einrichten

5. System installiert! – Wenn alles geklappt hat, meldet sich das System bei Aufruf der index.php im Anwendungsverzeichnis. Nun kann man im Management-Interface die Filialdaten eingeben, Speisekarten erstellen, Personal verwalten usw. Anschließend ist das System einsatzbereit.

4.7. Das System erweitern

Durch seine flexible Architektur ist das Ebar System sehr einfach durch zusätzliche Funktionalität erweiterbar. Im folgenden wird die Erweiterung des Systems anhand der Erstellung einer neuen Funktionalität für den Gast verdeutlicht.

Beispiel: Das System soll durch eine Funktion erweitert werden, die es dem Gast ermöglicht, seine eigenen Emails durch Eingabe von Adresse und Passwort abzufragen und auf Nachrichten zu reagieren.

Schritt 1: neuen Menüpunkt erstellen

In der Datenbank sind in der Tabelle **KATEGORIEN** alle Menüpunkte des Systems enthalten. Anhand der **KAT_NR** wird die Hierarchie abgebildet. Das zu erweiternde Menü "Extras" hat die Kategorie "1_2", die bereits existierenden Unterpunkte haben die Schlüssel "1_2_1" bis "1_2_4".

Man erstellt einen neuen Menüpunkt:

```
insert into KATEGORIEN values ("1_2_5", "Email", "email.gif");
```

Schritt 2: Bild erstellen

Mittels einer Grafiksoftware erstellt man ein Icon für die Funktionalität. Das Icon muss in zwei Größen vorliegen. Die Icons im Standardsystem haben die Größe 55 mal 55 Pixel bzw. 40 mal 40 Pixel. Die beiden erstellten Bilder legt man schließlich im Verzeichnis **/images/icons/** unter den Namen email.gif bzw. email_small.gif ab.

Schritt 3: Funktionalität in Index einfügen

In der **index_gast.php** wird sichergestellt, dass bei Aufruf des neuen Menüpunktes auch die hierzu gehörende Seite generiert wird. Für die neue Funktion ist entsprechend eine IF-Verzweigung einzubauen:

```
...
//der Emailfunktion den Email-Generator zuweisen
else if( substr($id, 0, 5) == "1_2_5")
{
    //eigene URL ermitteln
    $myurl = $PHP_SELF . "?selection=" . $selection;

    //... und dem Generator mitgeben
    $pg = new Gen_Email("Emailgenerator", $myurl);
    ${$name}->addPageGenerator( $pg );
}

```

Im Index der Gastoberfläche die neue Funktion verdrahten

Wird nun die neue Funktion gewählt, wird eine Instanz des Email-Pagegenerators erstellt, welcher Methoden besitzt um die dazugehörige HTML-Seite zu generieren.

Schritt 4: Funktionalität implementieren

Im Verzeichnis **/classes/logic/** wird die Kern-Funktionalität des Systems gehalten. Für die neu zu erstellende Funktion "Emails lesen" wird nun eine neue Klasse, **Email.cls.php** erstellt. Hierin entstehen alle für den späteren Einsatz wichtigen Methoden wie beispielsweise das Verbinden mit einem Server, das Abholen einer Mail etc.

```
<?
// Klasse zum Senden und Empfangen von Mails mittels POP3
class Email
{
    //Konstruktor
    function Email()
    {
    }

    //Abrufen der Mailliste von einem POP3-Server
    function getMailList($Server, $User, $Pass)
    {
        //hier kommt irgend ein Connect und so weiter...
        return $MailList
    }

    //Abrufen einer Mail von einem POP3-Server
    function getMessage($Server, $User, $Pass, $MessageID)
    {
        //eine einzelne Mail wird von einem Server geholt...
        return $Content
    }

    //einfaches Senden einer Mail
    function sendMail($Server, $User, $Pass, $To, $Subject, $Text)
    {
        //Mail wird irgendwie gesendet
        return $SendStatus
    }
}
?>
```

neue Klasse für die Email-Funktionalität

Schritt 5: Pagegenerator implementieren

Pagegeneratoren, zu finden in **/classes/generators/**, sind für die Darstellung der Seiten auf Clientseite zuständig. Für Gastseiten existiert eine abstrakte Klasse `Gen_Page`, von der man nun den neuen Generator für die Email-Funktionalität ableitet. Hierbei implementiert man vor allem die Methode `generatePage`, sie wird aufgerufen, wenn der Inhalt darzustellen ist.

```
<?
// Klasse Emailgenerator zum Senden und Lesen von Mail
class Gen_Email extends Gen_Page
{
    //Konstruktor
    function Gen_Email( $name, $myurl )
    {
        //eigener Name
        $this->Name = $name;

        //meine URL, falls spaeter gebraucht
        $this->myurl = $myurl;
    }

    function generatePage()
    {
        // was sollen wir darstellen?
        $action = $_GET['action'];
        if ( $action == "maillist" ) printMailList();
    }

    // gibt die Mailliste aus
    function printMailList()
    {
        // Serverdaten holen
        $Server = $_GET['server'];
        $User = $_GET['user'];
        $Pass = $_GET['pass'];

        // Mails vom Server holen
        $MailList = Email::getMailList($Server, $User, $Pass);

        // Liste darstellen
        echo "<table width=240>" ... to be continued
    }
}
```

neuer Pagegenerator für Email-Darstellung auf PDA

Die Darstellung erfolgt beispielsweise mit Formularen zum Eingeben der Serverdaten, Tabellen zur Auswahl der zu empfangenden Mail etc. Um unterschiedliche Funktionen in einem Generator darzustellen ist denkbar, die Seite sich selbst wieder aufrufen zu lassen und in einem POST bzw. GET Parameter die darzustellende Funktion, beispielsweise `action=read`, `action=send`, `action=input`, mitzuteilen. Eine IF-Verzweigung könnte dann aus `generatePage` zu weiteren Methoden innerhalb der Klasse verweisen, die wiederum die Methoden der Logikklasse `Email.cls.php` aufrufen und das Ergebnis darstellen.

6. neue Dateien in Includes eintragen

In der Daten **global_includes.php** werden die Logikklasse und der Generator eingetragen und stehen ab sofort dem System zur Verfügung.

```
include('classes/logic/Email.cls.php');
include('classes/generators/Gen_Email.cls.php');
```

Mit diesen sechs Schritten ist eine schnelle Erweiterung des Systems möglich. Auch eine Umstellung der Menüstruktur oder die Wiederverwendung von Funktionalitäten ist problemlos realisierbar durch entsprechende Updates der Datenbank bzw. Ableiten der Klassen.

Ebenso das Implementieren eines völlig neuen User Interfaces, beispielsweise einer Leuchtreklame im Fenster des Gastronomiebetriebes, ist denkbar einfach, da die vorhandenen Logikklassen bereits alle nötigen Funktionen beispielsweise zum Anzeigen von Produktinformationen zur Verfügung stellen.

5. Schlussbetrachtung

5.1. mögliche Erweiterungen

Durch die offene Architektur des Systems lassen sich neue Funktionalitäten sehr schnell und einfach realisieren. Viele Anwendungsgebiete in Gastronomiebetrieben und denkbare Features sind bereits durch die vorliegende Version erfasst, doch gibt es viele Ideen zu möglichen Erweiterungen. Einige seien im Folgenden erwähnt:



- Anbindung an betriebswirtschaftliche Anwendungen, d.h. Übergabe der Daten an Buchhaltungssysteme, Warenwirtschaft, Einkauf etc.
- Nutzung von Endgeräten des Kunden, d.h. Bestellen mit mitgebrachten PDA bzw. Notebook
- automatisierter Onlineauftritt der Gastronomiebetriebe
- Kommunizieren von Tischbelegungen oder Anzahl freier Plätze über Internet oder externe Displays
- Nutzung elektronischer Zahlungssysteme wie z.B. Kreditkarten
- Nutzerkonten für Gäste zum Verwalten von "Lieblingsgerichten" oder zusammengefasste Zahlung zu bestimmten Terminen am Monatsende o.ä.
- Einbinden populärer Instant Messenger und Chatsysteme
- verteilte Datenbanklösungen mit Datenabgleich

5.2. Fazit

Abschließend lässt sich sagen, dass das Vorhaben, ein multimediales Komplettsystem für Gastronomiebetriebe zu entwickeln, welches alle geschäftlichen Abläufe und Akteure integriert, zu einem ansprechenden Ergebnis geführt hat. Obwohl bereits die Analyse ein breites Spektrum an denkbaren Features offenbarte, entdeckten wir beim Programmieren immer wieder neue Möglichkeiten, das System noch umfassender und praktischer zu gestalten.

Herausforderungen bestanden unter anderem in der Zusammenarbeit innerhalb der Projektgruppe, da noch nicht jeder vor diesem Projekt die selben Erfahrungen in einem größeren Team gemacht hat. Auch bereitete es anfangs Schwierigkeiten, Aufgabenschwerpunkte zu finden und die Projektmitglieder entsprechend ihrer Vorkenntnisse und Ambitionen einzusetzen. Erschwerend kam noch hinzu, dass eines der Projektmitglieder in einem späten Stadium leider das Projekt verlassen hat und viele Zuständigkeiten neu verteilt werden mussten.

Auch auf technischer Seite standen wir zeitweise diversen Schwierigkeiten gegenüber. Beispielsweise traten immer wieder Fälle auf, in denen der PDA generierte Seiten nicht wie erwartet darstellte oder nicht vollständig zu (bspw. Javascript-) Standards kompatibel war, oder verschiedene Browser unsere komplexen Layer-Strukturen des Management-Interfaces nicht zufriedenstellend interpretierten.

Letztendlich ist es uns jedoch gelungen, ein homogenes und ansprechendes System zu realisieren, welches ohne Einschränkungen einsatzbereit ist. Es ist jedoch flexibel genug, um es in beinahe jegliche Richtung mit geringstem Aufwand zu erweitern. Selbst die bereits genannten und in kurzer Zeit realisierbaren Ausbaustufen geben noch längst keinen endgültigen Stand an.