

# Weiterentwicklung vom Gästebuch

## Die Dokumentation

## Inhalt

Inhalt .....	2
Einleitung und Ziele.....	3
Zielpublikum dieses Dokuments.....	3
Stakeholder für dieses Projekt .....	3
Ausgangslage .....	4
Ist-Situation .....	4
Vorhandenes Datenmodell.....	4
Anforderungen .....	5
Funktionale Anforderungen .....	5
Nichtfunktionale Anforderungen .....	6
Lösungsstrategie .....	7
Mockups.....	7
Statistiken.....	8
Profil.....	9
Profilangaben ändern.....	9
Komponenten .....	10
Neues Datenmodell .....	11
Migration .....	11
Dateistruktur .....	12
Testing .....	13
Testplan .....	13
Testfälle .....	13
Arbeitsjournale .....	19
Tabellenverzeichnis .....	22
Abbildungsverzeichnis.....	22

## Einleitung und Ziele

In diesem Projekt wird eine vorhandene Web-Applikation (Gästebuch vom Modul 133) weiterentwickelt, um fortgeschrittene Techniken wie AJAX und JavaScript anzuwenden.

Als Basis gilt das bereits vorhandene Datenmodell, das zusammen mit dem Design des Frontends aus dem alten Projekt übernommen wird. Das Datenmodell wird wo nötig erweitert oder angepasst, um den neuen Funktionalitäten zu entsprechen, während View und Controller zusammen mit einem Service vollständig neu aufgebaut werden.

## Zielpublikum dieses Dokuments

Um den Inhalt dieses Dokuments zu verstehen, muss die lesende Person in Kenntnis über das Vorgängerprojekt vom Modul 133 sein.

Dieses Dokument richtet sich an IT-Fachleute mit Verständnis im Bereich Web-Entwicklung mit HTML, CSS, PHP und JavaScript.

## Stakeholder für dieses Projekt

*Tabelle 1: Stakeholder*

Stakeholder	Rolle und Erwartung
Manuel Sollberger	<b>Projektleiter</b>  Aus Sicht der Projektleitung wird eine saubere Weiterentwicklung der Applikation erwartet, um die erweiterten Technologien wie AJAX und JavaScript zum Einsatz zu bringen.
Stefan Sollberger	<b>Auftraggeber und Endanwender</b>  Vom Auftraggeber werden einige Verbesserungen und Erweiterungen der Applikation gefordert. Auf diese wird im Abschnitt <i>Anforderungen</i> eingegangen.

## Ausgangslage

Im Modul 133 wurde eine PHP-basierte Applikation zur Erfassung von Besuchen erstellt. Dieses System wird bereits aktiv genutzt, was bei der Datenmigration zu beachten ist.

Das Projekt wird durchgeführt, da mit der aktuellen Lösung einige gewünschte Funktionen nicht optimal implementiert wurden oder noch gar nicht existieren.

## Ist-Situation

Die aktuelle Applikation wurde nach MVC aufgebaut, weswegen diese drei Komponenten vorhanden sind:

- Model: Datenbankstruktur und serverseitige Helper-Klassen
- View: HTML-basierte Präsentation, die von PHP-Skripts zusammengebaut wird
- Controller: PHP-Skripte, die Formulareingaben entgegennehmen und verarbeiten

Da in der existierenden Applikation nur auf PHP gesetzt wurde, gibt es keine clientseitigen Skripts. Alle Aktionen erfordern eine Serveranfrage.

Die Ordnerstruktur der Applikation entspricht den MVC-Komponenten.

## Vorhandenes Datenmodell

Das Datenmodell aus dem ursprünglichen Datenmodell sieht folgendermassen aus:

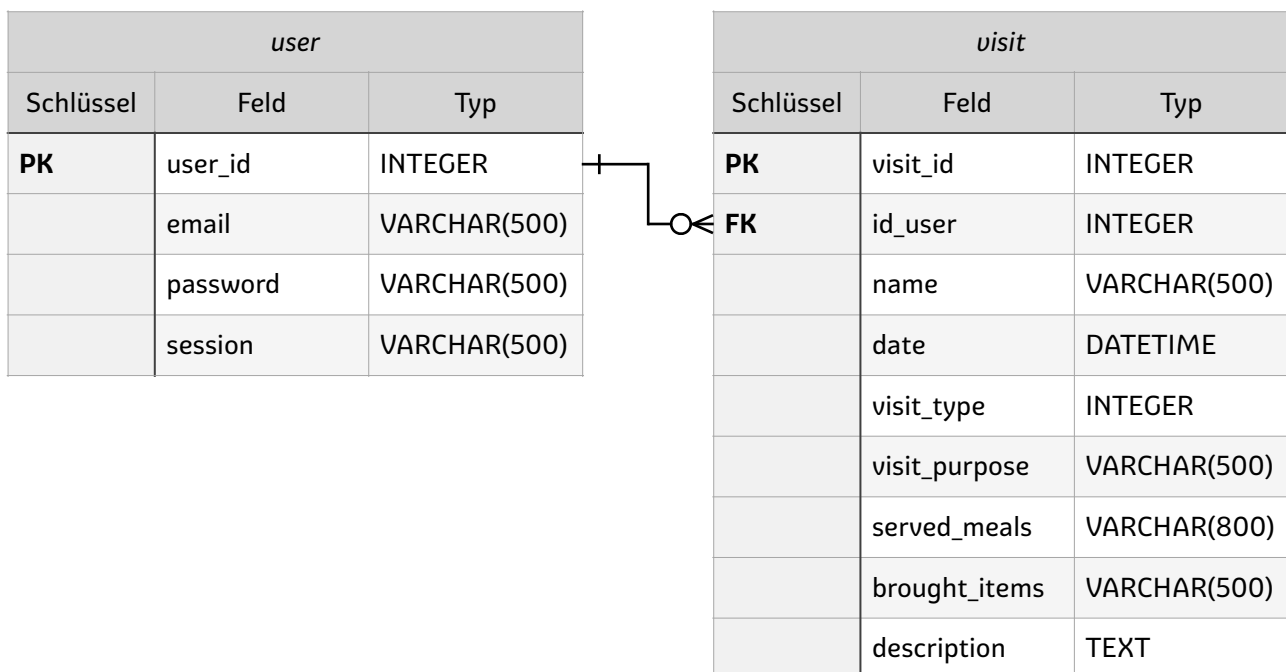


Abbildung 1: Vorhandenes Datenmodell

## Anforderungen

Vom Auftraggeber wurden diese Änderungen gefordert:

- Einzelne Angabe von Besuchsteilnehmern (analog zum Empfängerfeld beim Schreiben einer E-Mail)
- Aufteilung der servierten Gerichte in einzelne Felder pro Mahlzeit
  - Benutzerdefinierte Auswahl an Mahlzeiten
- Filterung der Besuchseinträge nach Person
- Optimierung der Statistiken
  - Klick auf eine Person in einer Statistik wechselt zur Übersicht mit dem entsprechenden Filter aktiv

Zusätzlich zu diesen geforderten funktionalen Anforderungen gilt es nun mit AJAX und JavaScript zu arbeiten.

Es folgt eine komplette Auflistung aller funktionalen und nichtfunktionalen Anforderungen an die Applikation (bisherige und neue zusammen).

## Funktionale Anforderungen

*Tabelle 2: Funktionale Anforderungen / User-Stories*

Story	Anforderung
<b>US1</b>	Als Benutzer möchte ich mich mit meiner E-Mail-Adresse und einem Passwort registrieren können, um Zugang zur Besuchserfassung zu erhalten.
<b>US2</b>	Als Benutzer möchte ich mich an- und abmelden können, um auf meine erfassten Besuche zugreifen zu können.
<b>US3</b>	Als angemeldeter Benutzer möchte ich nach 30 Minuten Inaktivität automatisch abgemeldet werden, damit meine Sitzung nicht länger gültig ist als nötig.
<b>US4</b>	Als angemeldeter Benutzer möchte ich meine Anmeldeinformationen ändern können.
<b>US5</b>	Als angemeldeter Benutzer möchte ich Besuchseinträge erstellen, bearbeiten und löschen können, um ein sauberes Gästebuch meiner Besuche zu erhalten.
<b>US6</b>	Als angemeldeter Benutzer möchte ich eine Liste der Besuche pro Person einsehen können, um zu sehen, wen ich am meisten sehe.
<b>US7</b>	Als angemeldeter Benutzer möchte ich sehen können, welche Personen ich am längsten nicht mehr gesehen habe, um den Kontakt aufrecht zu erhalten.
<b>US8</b>	Als angemeldeter Benutzer möchte ich beim Erfassen eines Besuchs auswählen können, welche Mahlzeiten serviert wurden. Pro Mahlzeit soll ich ein Textfeld erhalten, in welchem ich das servierte Gericht beschreiben kann.
<b>US9</b>	Als angemeldeter Benutzer möchte ich meine Besuchsliste nach Person filtern können, um alle Besuche mit einer bestimmten Person einsehen zu können.

## Nichtfunktionale Anforderungen

*Tabelle 3: Nichtfunktionale Anforderungen*

Nummer	Anforderung
<b>NFA1</b>	Die serverseitige Applikation wird mit PHP umgesetzt.
<b>NFA2</b>	Die clientseitige Applikation wird mit JavaScript umgesetzt.
<b>NFA3</b>	Zur Datenübertragung zum Server wird mit AJAX gearbeitet.
<b>NFA4</b>	Die Benutzer- und Besuchsdaten werden in einer MySQL-Datenbank gespeichert.
<b>NFA5</b>	Zur Authentifizierung wird das Standard-PHP-Session-Handling verwendet.
<b>NFA6</b>	Wiederkehrende Eingaben werden mit einer <datalist> mit den bereits vorhandenen Werten vereinfacht.
<b>NFA7</b>	Die Filterung der Besuchsliste soll clientseitig und ohne Neuladen der Seite erfolgen.
<b>NFA8</b>	Die Datenübertragung zwischen Client und Server erfolgt im JSON-Format.

## Lösungsstrategie

Es werden im Wesentlichen zwei Applikationen umgesetzt, nämlich jeweils eine auf dem Client und auf dem Server. Diese beiden Applikationen kommunizieren untereinander initial via Seitenaufruf und zum Datenaustausch via AJAX. Dabei wird zum Datenaustausch das JSON-Format gewählt.

Als Schnittstelle zwischen der Client- und Server-Applikation wird ein Service gebaut. Dieser bietet die notwendigen Funktionen zur Datenhandhabung.

Es wird nach wie vor mit einer MySQL-Datenbank gearbeitet, um die Daten serverseitig abzuspeichern.

## Mockups

Es folgen einige Abbildungen zum geplanten Aufbau des Frontends. Es gibt an mehreren Stellen Abweichungen zur Struktur der bestehenden Applikation.

Die grösste Neuerung ist die Einführung von Modal-Ansichten an mehreren Stellen. Dies verbessert den User-Flow.

## Besuchsliste

Besuche	Statistiken	Profil
<div><b>Besuche</b></div> <div><div>Besuch erfassen</div><div>Personen: <div>Alle ▼</div></div></div> <div><div><b>Besuch von Horst Henkelschwümkler</b> zum Nacht</div><div>2021-08-01</div><div><i>Servierte Gerichte:</i> <b>Vorspeise:</b> Chueche <b>Hauptspeise:</b> Gratin <b>Dessert:</b> Glacé <i>Mitgebracht:</i> Luschtigi sache</div></div> <div><div><b>Besuch von Horst Wuppmann und Monika Juvet</b> zum Nacht</div><div>2021-08-01</div><div><i>Servierte Gerichte:</i> <b>Apéro:</b> Chueche <b>Hauptspeise:</b> Gratin <b>Dessert:</b> Glacé <i>Mitgebracht:</i></div></div>		

Abbildung 2: Mockup von der Besuchsliste

## Besuchserfassung/-bearbeitung

**Besuche** Statistiken Profil

Besuch erfassen/bearbeiten X

Datum  
12.03.2021

Anlass  
Weihnachtsessen

Servierte Gerichte  
Apéro ▼ Chips X  
Hauptspeise ▼ Rindsfilet X

+  
Mitgebrachtes  
Nätts züüg

Sonstiges

Speichern Löschen ...

Abbildung 3: Mockup von der Besuchserfassung

## Statistiken

**Besuche** **Statistiken** Profil

Besuche pro Person ▼	
Hans	12 Einträge
Letztes Treffen: 2021-09-01	Anzeigen >
Schorsch	8 Einträge
Letztes Treffen: 2021-06-21	Anzeigen >
Fritz	2 Einträge
Letztes Treffen: 2021-12-20	Anzeigen >

Abbildung 4: Mockup von den Statistiken



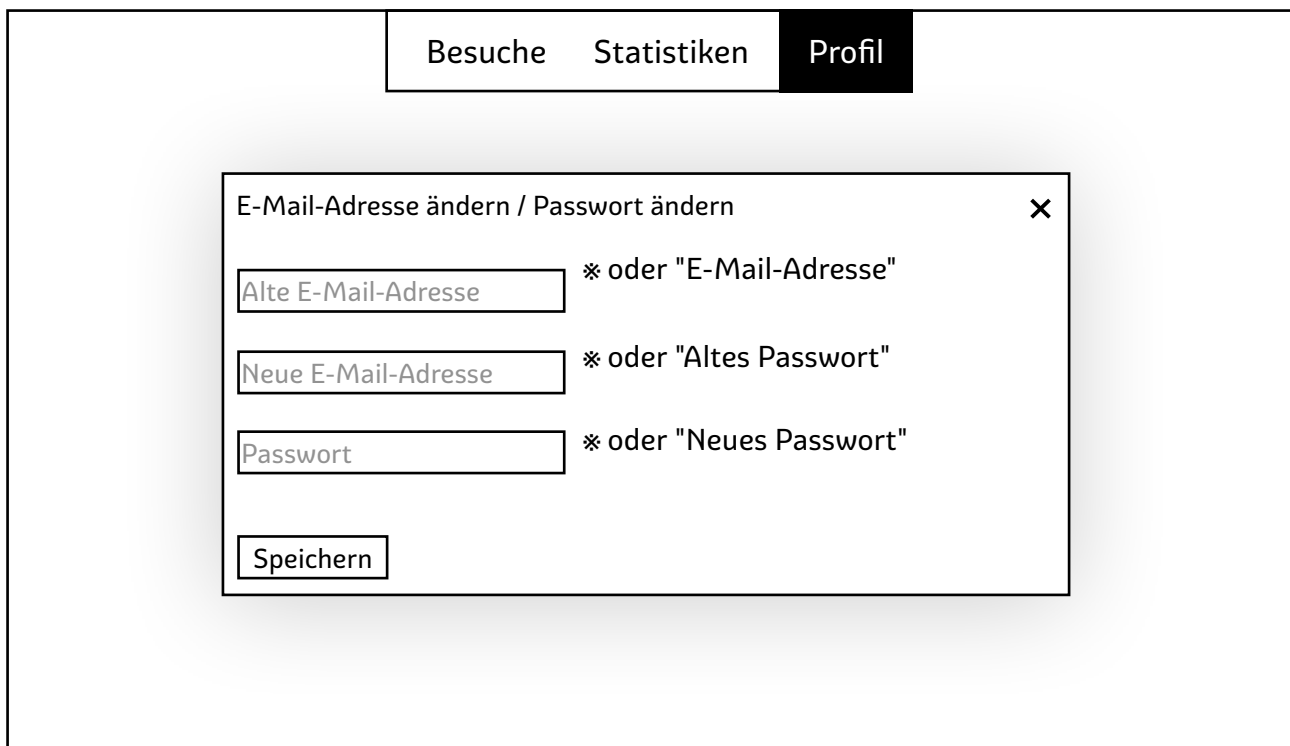
## Profil



The mockup shows a web interface with a top navigation bar containing three buttons: 'Besuche', 'Statistiken', and 'Profil'. The 'Profil' button is highlighted with a black background. Below the navigation bar, there is a central box titled 'Mein Profil'. Inside this box, the email address 'horst@henkelschwuenkler.de' is displayed. Below the email address, there are three buttons: 'E-Mail-Adresse ändern', 'Passwort ändern', and 'Abmelden'.

Abbildung 5: Mockup von der Profilansicht

## Profilangaben ändern



The mockup shows a web interface with a top navigation bar containing three buttons: 'Besuche', 'Statistiken', and 'Profil'. The 'Profil' button is highlighted with a black background. Below the navigation bar, there is a modal dialog box titled 'E-Mail-Adresse ändern / Passwort ändern' with a close button (X) in the top right corner. Inside the dialog, there are three input fields: 'Alte E-Mail-Adresse', 'Neue E-Mail-Adresse', and 'Passwort'. To the right of each input field is a label: '✖ oder "E-Mail-Adresse"', '✖ oder "Altes Passwort"', and '✖ oder "Neues Passwort"'. At the bottom of the dialog, there is a 'Speichern' button.

Abbildung 6: Kombiniertes Mockup von der Profilbearbeitung

## Komponenten

Das folgende Komponentendiagramm erklärt die Aufteilung der neuen Applikation(en) in die einzelnen Komponenten.

Die User-Story beginnt mit dem Benutzer, der die Webseite aufruft. Der Leitfaden führt von Anfang an zur Server-Applikation, da der Browser selbst nicht Teil der Client-Applikation ist.

Erst durch die vom PageBuilder in die Seite geschriebenen Skripts wird die Client-Applikation gestartet, welche dann wiederum Daten vom Server via Service bezieht.

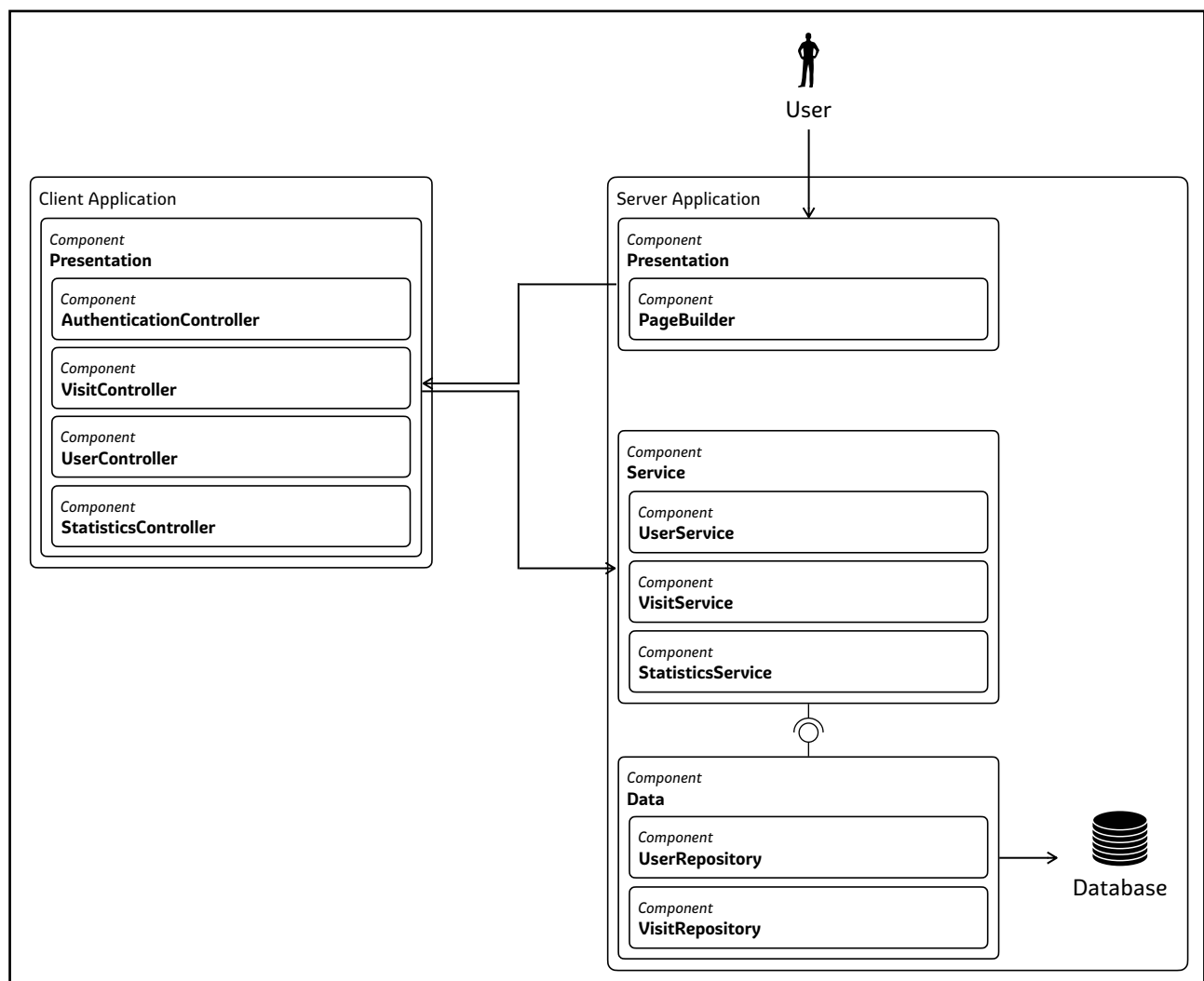


Abbildung 7: Komponentendiagramm

## Neues Datenmodell

Hier wird das angepasste Datenmodell mithilfe eines ERD definiert. Rot markierte und durchgestrichene Felder wurden entfernt und grün hinterlegte Elemente wurden neu hinzugefügt.

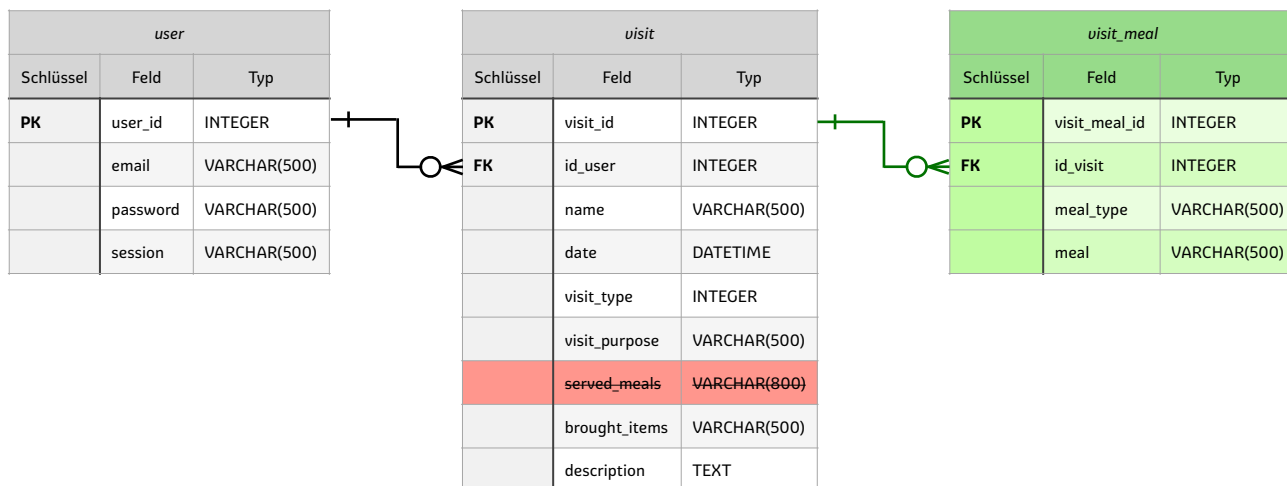


Abbildung 8: Neues Datenmodell

Damit die servierten Gerichte dynamisch erfasst werden können, wird diese Angabe nun in eine separate Tabelle verschoben.

Die Namen der Personen eines Besuchs werden immer kommasepariert im Namensfeld erfasst.

## Migration

Da die bestehende Applikation aktiv genutzt wird, muss eine sichere und geplante Datenmigration erfolgen. Sicherheitshalber wird während der Migration der Zugriff auf die Web-Applikation verweigert, bis alle Daten im neuen Format in der Datenbank vorhanden sind.

Konkret handelt es sich hierbei um die Angabe der servierten Gerichte bei einem Besuch. Diese Angabe war bisher ein einfaches Textfeld und wird nun in einzelne Einträge aufgespalten.

Da es sich bei der Angabe der Gerichte um ein Textfeld handelt, sind die Werte inkonsistent. Nicht immer wurde von den Benutzern eine klare und einheitliche Struktur gewählt, um die verschiedenen Gerichte zu erfassen. Deswegen können die Werte nicht automatisiert auf mehrere Einträge aufgeteilt werden.

Stattdessen muss die Aufteilung von Hand durch den Benutzer erfolgen. Bis dahin wird der gesamte Inhalt des alten Feldes als einziger Eintrag in die Zusatztable übertragen. Die Art der Mahlzeit wird leer gelassen. Der Benutzer kann sich jederzeit die Zeit nehmen, um die Werte in separate Einträge aufzuteilen.

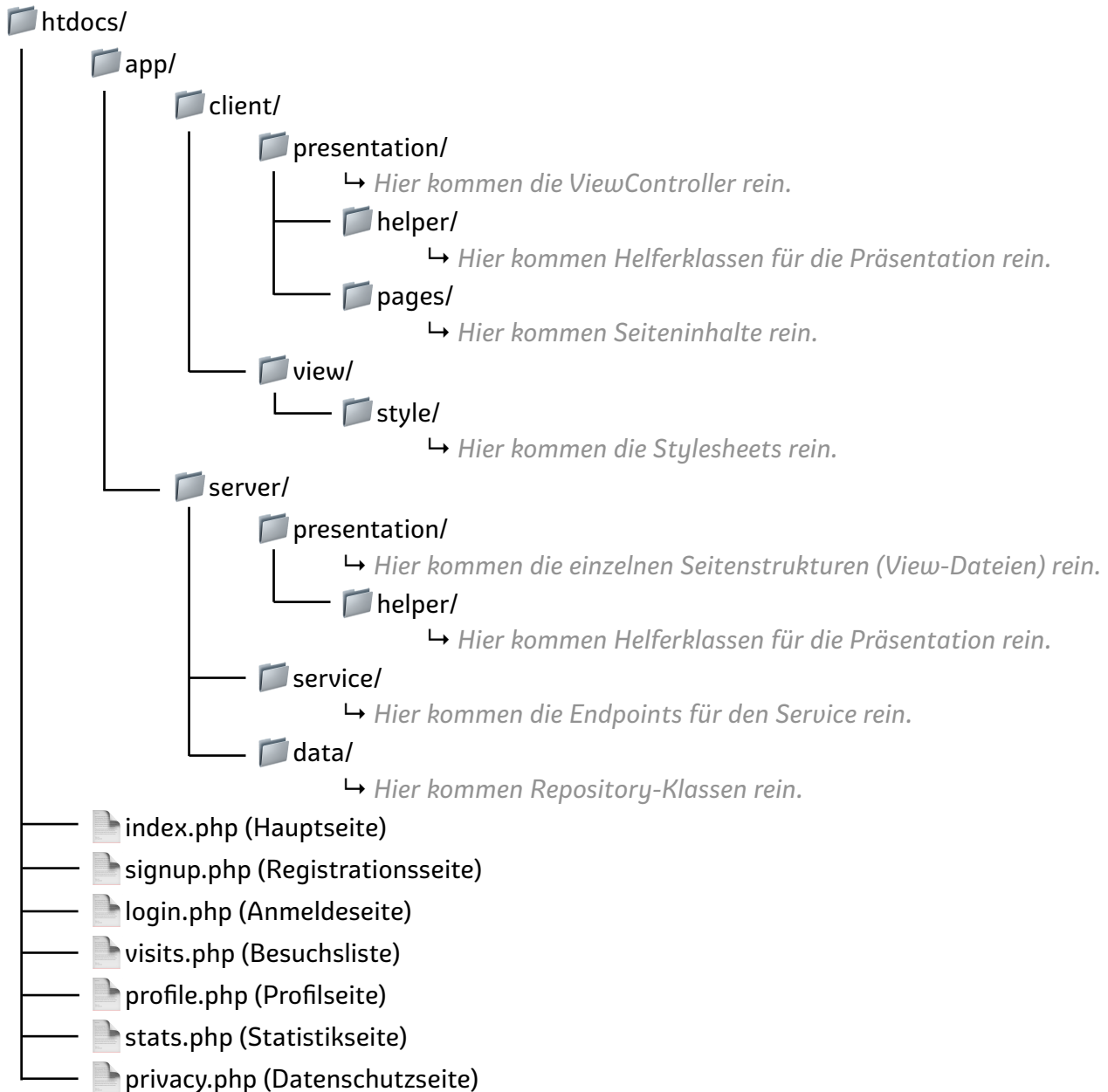
Die Namen der Besuchenden/Besuchten von bisherigen Einträgen werden nicht verändert, obwohl nun eine strikte kommaseparierte Liste im Feld erwartet wird. Bei existierenden Besuchen, bei denen mehrere Personen nicht kommasepariert erfasst wurden, beispielsweise mit einem + oder &, wird die Personengruppe als "eine Person" behandelt. Der Benutzer kann diese Angaben später nach belieben bereinigen und aufteilen.

Dies ist die SQL-Abfrage, die zur Migration der Datenbankwerte verwendet wird:

```
INSERT INTO visit_meal(id_visit, meal_type, meal) SELECT visit_id AS  
id_visit, "" AS meal_type, REPLACE(served_meals, "\r\n", ", ") AS meal  
FROM visit;
```

## Dateistruktur

Die Dateistruktur von der neuen Applikation folgt ziemlich genau dem Muster des Komponentendiagramms.



# Testing

## Testplan

Die Applikation wird mit macOS-, iOS- und Android-Geräten (daher Macs, iPhones und Smartphones verschiedener Non-Apple-Anbieter) getestet, da dies den verwendeten Geräten der Benutzer entspricht.

Zum Testen werden die Browser Safari und Chrome verwendet.

Aufgrund der verschiedenen Gerätegrößen ist die Prüfung der Ausnahmslosen Responsibilität ein zentrales Thema beim Testen der Applikation.

Zum Testen werden eigene Testdaten verwendet. Wenn die Applikation als einsatzreif angesehen wird, kann der Auftraggeber die neue Version mit seinen bestehenden Daten unter Aufsicht des Entwicklers im Praxiseinsatz testen.

## Testfälle

Hier werden Testfälle für alle User-Stories und ein paar zusätzliche Fälle definiert.

*Tabelle T1: Testfall US1*

Testfall zur User-Story 1: Registration	
<b>Durchführungsschritte</b>  <ol style="list-style-type: none"><li>1. Registrationsseite aufrufen</li><li>2. E-Mail-Adresse und Passwort eingeben</li><li>3. Registrationsknopf drücken</li></ol>	
<b>Erwartetes Ergebnis</b>  Ein neues Benutzerkonto wurde erstellt. Der Benutzer ist auch gleich angemeldet und wird auf seine Besuchliste geleitet.	<b>Tatsächliches Ergebnis</b>  Ein 500er-Fehler wird zurückgegeben.
<b>Getroffene Massnahmen</b>  Die Spalte "session" in der Benutzertabelle musste vorher noch <i>nullable</i> gemacht werden.	

*Tabelle T2: Testfall US2*

Testfall zur User-Story 2: An- und abmeldung
<b>Durchführungsschritte</b>  <ol style="list-style-type: none"><li>1. Anmeldeseite aufrufen, ohne angemeldet zu sein</li><li>2. E-Mail-Adresse und Passwort eingeben</li><li>3. Anmeldeknopf drücken</li><li>4. Profilseite aufrufen</li><li>5. Abmeldeknopf drücken</li></ol>

Testfall zur User-Story 2: An- und abmeldung	
<b>Erwartetes Ergebnis</b>  Der Benutzer wird nach Schritt 3 angemeldet und auf die Besuchliste geleitet. Nach dem Drücken des Abmeldeknopfes wird er wieder zur Anmeldeseite geleitet und die Sitzung ist nicht mehr gültig.	<b>Tatsächliches Ergebnis</b>  Die Anmeldung schlägt jedes Mal fehl.
<b>Getroffene Massnahmen</b>  Was das Problem am Ende gelöst hat, weiss ich nicht. Ich habe viel rumprobiert und offenbar gab es ein Problem mit der password_verify()-Funktion.	

*Tabelle T3: Testfall US3*

Testfall zur User-Story 3: Automatische Abmeldung	
<b>Durchführungsschritte</b>  1. Erfolgreich anmelden 2. 30 Minuten vergehen lassen, ohne eine Aktion durchzuführen 3. Login-pflichtige Anfrage an den Server schicken	
<b>Erwartetes Ergebnis</b>  Der Benutzer wird auf die Anmeldeseite geleitet, weil die Sitzung abgelaufen ist.	<b>Tatsächliches Ergebnis</b>
<b>Getroffene Massnahmen</b>	

*Tabelle T4: Testfall US4*

Testfall zur User-Story 4: Anmeldeinformationen ändern	
<b>Durchführungsschritte</b>  1. Erfolgreich anmelden 2. Profilseite aufrufen 3. "E-Mail-Adresse ändern" drücken 4. Alte und neue E-Mail-Adresse sowie das aktuelle Passwort eingeben 5. Das Formular abschicken 6. "Passwort ändern" drücken 7. Aktuelle E-Mail-Adresse sowie das alte und neue Passwort angeben 8. Das Formular abschicken	

Testfall zur User-Story 4: Anmeldeinformationen ändern	
<b>Erwartetes Ergebnis</b>	<b>Tatsächliches Ergebnis</b>
Nach Schritt 5 ist die neue E-Mail-Adresse gültig. Nach Schritt 8 gilt zur Anmeldung das neue Passwort. Das verwenden der alten Angaben ist nicht mehr möglich.	Wie erwartet.
<b>Getroffene Massnahmen</b>	
Keine.	

*Tabelle T5: Testfall US5*

Testfall zur User-Story 5: Besuche erfassen, bearbeiten und löschen	
<b>Durchführungsschritte</b>	
<ol style="list-style-type: none"> <li>1. Besuchliste aufrufen</li> <li>2. "Besuch erfassen" drücken</li> <li>3. Besuchsangaben eingeben</li> <li>4. "Speichern" drücken</li> <li>5. Besuch in der Liste anwählen</li> <li>6. Eine oder mehrere Angaben ändern</li> <li>7. "Speichern" drücken</li> <li>8. Besuch in der Liste anwählen</li> <li>9. "Löschen ..." drücken</li> <li>10. Bestätigungsdialog bestätigen</li> </ol>	
<b>Erwartetes Ergebnis</b>	<b>Tatsächliches Ergebnis</b>
Nach Schritt 4 ist der neu erfasste Besuch in der Besuchliste auffindbar. Nach Schritt 7 wurden die geänderten Angaben übernommen. Nach Schritt 10 wurde der Eintrag gelöscht und wird nicht mehr angezeigt.	Es wird noch kein Bestätigungsdialog vor dem Löschen eines Eintrags angezeigt.
<b>Getroffene Massnahmen</b>	
Bestätigungsdialog beim Löschen eines Eintrags eingebaut.	

*Tabelle T6: Testfall US6*

Testfall zur User-Story 6: Besuche pro Person einsehen
<b>Durchführungsschritte</b>
<ol style="list-style-type: none"> <li>1. Statistikseite aufrufen</li> <li>2. Statistik "Besuche pro Person" auswählen</li> </ol>

Testfall zur User-Story 6: Besuche pro Person einsehen	
<b>Erwartetes Ergebnis</b>  Es wird eine Liste aller Personen angezeigt. Daneben ist die Anzahl der Einträge mit dieser Person zu sehen. Die Liste ist sortiert nach der Anzahl Besuchen pro Person.	<b>Tatsächliches Ergebnis</b>  Wie erwartet.
<b>Getroffene Massnahmen</b>  Keine.	

*Tabelle T7: Testfall US7*

Testfall zur User-Story 7: Am längsten nicht mehr gesehene Personen einsehen	
<b>Durchführungsschritte</b>  1. Statistikseite aufrufen 2. Statistik "Lange nicht mehr gesehen" auswählen	
<b>Erwartetes Ergebnis</b>  Es werden alle Personen aus allen Besuchseinträgen aufgelistet, zusammen mit der Zeit in Tagen, seit denen diese Person nicht mehr an einem Besuch beteiligt war.	<b>Tatsächliches Ergebnis</b>  Wie erwartet.
<b>Getroffene Massnahmen</b>  Keine.	

*Tabelle T8: Testfall US8*

Testfall zur User-Story 8: Verschiedene Mahlzeiten hinzufügen	
<b>Durchführungsschritte</b>  1. Besuchserfassung aufrufen 2. "Mahlzeit hinzufügen" drücken 3. Art der Mahlzeit und serviertes Gericht eingeben 4. Schritte 1 bis 3 beliebig oft mit unterschiedlichen Mahlzeiten wiederholen	
<b>Erwartetes Ergebnis</b>  Es werden untereinander Mahlzeiten und die dazu servierten Gerichte erfasst, zum Beispiel "Apéro", "Vorspeise" und "Hauptspeise".	<b>Tatsächliches Ergebnis</b>  Wie erwartet.
<b>Getroffene Massnahmen</b>  Keine.	



*Tabelle T9: Testfall US9*

Testfall zur User-Story 9: Besuchliste filtern	
<b>Durchführungsschritte</b>  1. Besuchliste aufrufen 2. Im Feld "Personen" eine Person auswählen	
<b>Erwartetes Ergebnis</b>  Alle Besuchseinträge, bei denen die ausgewählte Person nicht beteiligt war, werden ausgeblendet.	<b>Tatsächliches Ergebnis</b>  Wie erwartet.
<b>Getroffene Massnahmen</b>  Keine.	

*Tabelle T10: Testfall S1*

Sondertestfall 1: SQL-Injection	
<b>Durchführungsschritte</b>  1. In einem beliebigen Eingabefeld ein SQL-Code-Segment eingeben 2. Das Formular abschicken	
<b>Erwartetes Ergebnis</b>  In Textfeldern eingegebene SQL-Anfragen werden nie ungeprüft an die Datenbank geschickt.	<b>Tatsächliches Ergebnis</b>  Wie erwartet.
<b>Getroffene Massnahmen</b>  Keine.	

*Tabelle T11: Testfall S2*

Sondertestfall 2: XSS	
<b>Durchführungsschritte</b>  1. In einem beliebigen Eingabefeld HTML eingeben 2. Die Daten speichern und anzeigen lassen	
<b>Erwartetes Ergebnis</b>  HTML wird aus den Werten rausgefiltert.	<b>Tatsächliches Ergebnis</b>  Bei den Mahlzeiten wird HTML tatsächlich auf die Seite ausgegeben.
<b>Getroffene Massnahmen</b>  Ich habe strip_tags() auf die Felder der Mahlzeiten angewendet.	

*Tabelle T12: Testfall S3*

<b>Sondertestfall 3: Besuch einer anderen Person verändern</b>	
<b>Durchführungsschritte</b>  1. Einen Besuch zur Bearbeitung aufrufen 2. Die ID des zu bearbeitenden Besuchs manipulieren 3. Den Besuch speichern	
<b>Erwartetes Ergebnis</b>  Das Speichern schlägt fehl, da nur auf Besuche des angemeldeten Benutzers zugegriffen werden kann.	<b>Tatsächliches Ergebnis</b>  Wie erwartet.
<b>Getroffene Massnahmen</b>  Keine.	

## Arbeitsjournale

*Tabelle A1: Arbeitsjournal vom Tag 1*

Tag 1	
Tätigkeit	Arbeitszeit (h)
Dokumentationsstruktur aufbauen	1
Zielsetzung, Anforderungen und Lösungsstrategie beschreiben	3.5
Neues Datenmodell gestalten	0.8
Testplan schreiben	1.4
<b>Reflexion</b>  Das war heute ein sehr dokumentationsintensiver Tag. Doch das ist nichts schlechtes: So kann ich morgen direkt mit der Umsetzung beginnen. Ich bin froh, konnte ich die Planung innerhalb von einem Tag fertigstellen.	

*Tabelle A2: Arbeitsjournal vom Tag 2*

Tag 2	
Tätigkeit	Arbeitszeit (h)
Dokumentation durchlesen und optimieren	1.2
Dateistruktur definieren und erstellen	0.4
Umsetzung des Frontends beginnen	1
<b>Reflexion</b>  Heute habe ich weniger geschafft als erhofft. Immerhin habe ich nun die Grundlage, um die Applikation endgültig umzusetzen. Die Login-Seite ist aufgebaut und gilt in der Architektur als Basis für die kommenden Seiten.	

*Tabelle A3: Arbeitsjournal vom Tag 3*

Tag 3	
Tätigkeit	Arbeitszeit (h)
Service aufbauen	2
Registration, Anmeldung und E-Mail-Änderung umsetzen	1.9
<b>Reflexion</b>  Heute war ein durchschnittlicher Tag. Ich konnte grosse Teile der Benutzerverwaltung umsetzen. Einzig die Funktion zur Passwortänderung und die Abmeldung fehlen. Wenn diese umgesetzt sind, kann ich mit der Umsetzung der Besuchserfassung beginnen.	

*Tabelle A4: Arbeitsjournal vom Tag 4*

Tag 4	
Tätigkeit	Arbeitszeit (h)
Benutzerverwaltung fertigstellen	2.7
Besuchsliste bauen	1.5
<b>Reflexion</b>  Heute konnte ich endlich die Benutzerverwaltung inklusive Passwortänderung fertigstellen. Daraufhin begann ich mit der Umsetzung der Besuchsliste. Da kam ich einigermaßen voran, jedoch fehlt noch vieles.	

*Tabelle A5: Arbeitsjournal vom Tag 5*

Tag 5	
Tätigkeit	Arbeitszeit (h)
Besuchsverwaltung fertigstellen	4.5
Datenbank umbauen	0.5
Statistiken implementieren	1.5
<b>Reflexion</b>  Was für ein Tag! So viel habe ich glaube ich noch nie an so einem Projekt gearbeitet. Dies blieb aber nicht unbelohnt: Nun ist das Projekt fast fertig. Nur noch die Migration auf das Produktivsystem fehlt. Darum werde ich mich morgen kümmern.	

*Tabelle A6: Arbeitsjournal vom Tag 6*

Tag 6	
Tätigkeit	Arbeitszeit (h)
Letzte Optimierungen durchführen	1.3
Migration auf dem Produktivsystem durchführen	0.5
Testen und verbessern	1.5
<b>Reflexion</b>  So! Damit ist das Projekt abgeschlossen. Ich konnte heute die Testphase abschliessen und anschliessend die Migration durchführen. Beim Testen kamen mehrere Probleme ans Licht, die ich sofort beheben konnte. Nun steht die neue Version des Gästebuchs im Produktiveinsatz.	

## Tabellenverzeichnis

*Tabelle E1: Tabellenverzeichnis*

<b>Tabelle</b>	<b>Bezeichnung</b>
<b>1</b>	Stakeholder
<b>2</b>	Funktionale Anforderungen / User-Stories
<b>3</b>	Nichtfunktionale Anforderungen
<b>T1</b>	Testfall US1
<b>T2</b>	Testfall US2
<b>T3</b>	Testfall US3
<b>T4</b>	Testfall US4
<b>T5</b>	Testfall US5
<b>T6</b>	Testfall US6
<b>T7</b>	Testfall US7
<b>T8</b>	Testfall US8
<b>T9</b>	Testfall US9
<b>T10</b>	Testfall S1
<b>T11</b>	Testfall S2
<b>T12</b>	Testfall S3
<b>A1-A6</b>	Arbeitsjournale
<b>E1</b>	Tabellenverzeichnis
<b>E2</b>	Abbildungsverzeichnis

## Abbildungsverzeichnis

*Tabelle E2: Abbildungsverzeichnis*

<b>Abbildung</b>	<b>Bezeichnung</b>
<b>1</b>	Vorhandenes Datenmodell
<b>2</b>	Mockup von der Besucherliste
<b>3</b>	Mockup von der Besuchserfassung
<b>4</b>	Mockup von der Statistiken

Abbildung	Bezeichnung
5	Mockup von der Profilansicht
6	Kombiniertes Mockup von der Profilbearbeitung
7	Komponentendiagramm
8	Neues Datenmodell