



# **Studienprojekt im Rahmen der Vorlesung Mobile Computing**



(Icon mit Auflösung für mobile Geräte)

## **HAW Mensa**

### **Konzeption und Realisierung einer Android-basierten Applikation für den Mensa Speiseplan**

Dozent: Prof. Dr. Abdelmajid Khelil

Autor: Daniel Lackmann

Matrikelnummer: 1022839

Landshut, den 05.11.18

## **Inhaltsverzeichnis**

<b>Einführung</b>	<b>3</b>
Beschreibung und Vision	3
Zielgruppe	3
<b>Planung und Konzeption</b>	<b>4</b>

Funktionale Anforderungen	4
Menüleiste	5
Auswählen eines Tages	5
Auswählen eines Gerichtes	6
Gericht Detailansicht	6
Filtern von Gerichten	7
Einstellungen	7
Über die Applikation	7
Hilfe/Legende	8
<b>Backend</b>	<b>9</b>
<b>Styleguide</b>	<b>10</b>
Typography	10
Iconography	11
Colors	11
Theme 1: (Example can change during development)	11
Theme 2: (Example change during development)	12
<b>Release und Planung</b>	<b>13</b>
<b>Implementierung</b>	<b>14</b>
Activities	15
Data access object	15
<b>Test</b>	<b>16</b>
Stufe 2 - Smoke-Tests	16

# Einführung

## Beschreibung und Vision

Bist du gerade in der Nähe der öffentlichen Mensa, und hast noch nichts gegessen und würdest gerne wissen, was zur Auswahl steht?

Oder willst du nur noch eine Kleinigkeit von den Beilagen?

Oder willst du einfach gerade wissen, welche Desserts gerade auf dem Plan stehen?

Die mobile Applikation „HAW Mensa“ ermöglicht dem Benutzer auf die schnelle eine einfache und anschauliche Ansicht der aktuellen Speisekarte zu erhalten und beantwortet dem Benutzer diese Fragen.

Mit der App lassen sich die Gerichte für die aktuelle als auch nächste Wochen anzeigen.

In einer strukturierten Ansicht kann der gewünschte Tag ausgewählt werden.

Die Speisen eines Tages können in verschiedenen Kategorien angezeigt werden und der Endnutzer kann nach seinen Präferenzen filtern.

Du willst nach dem Essen deine Meinung über die Speise äußern?

Auch dafür hat die App die Antwort, denn sie lässt dich für jede Speise eine Bewertung abgeben.

## Zielgruppe

Die Zielgruppe von „HAW Mensa“ beschränkt sich zum größten Teil auf Gäste der öffentlichen Mensa.

Das Alter der Zielgruppe spielt keine ausschlaggebende Rolle, da auch das Alter der Besucher der Mensa eine große Altersspanne haben.

Unabhängig von persönlichen Eigenschaften liegen besonders die Benutzer im Fokus, die bestimmte Präferenzen bei der Auswahl ihrer Speisen haben.

Eine weitere Abgrenzung der Zielgruppe ergibt sich aus den bestehenden technischen Systemvoraussetzungen.

Der potentielle Nutzer benötigt ein Smartphone, der vorgegebenen Android-Plattform, das mindestens mit der Version API 14: Android 4.0 (IceCreamSandwich) ausgestattet sein muss.

Rund 100% Prozent der aktuell genutzten Android-Geräte erfüllen dieses Kriterium.

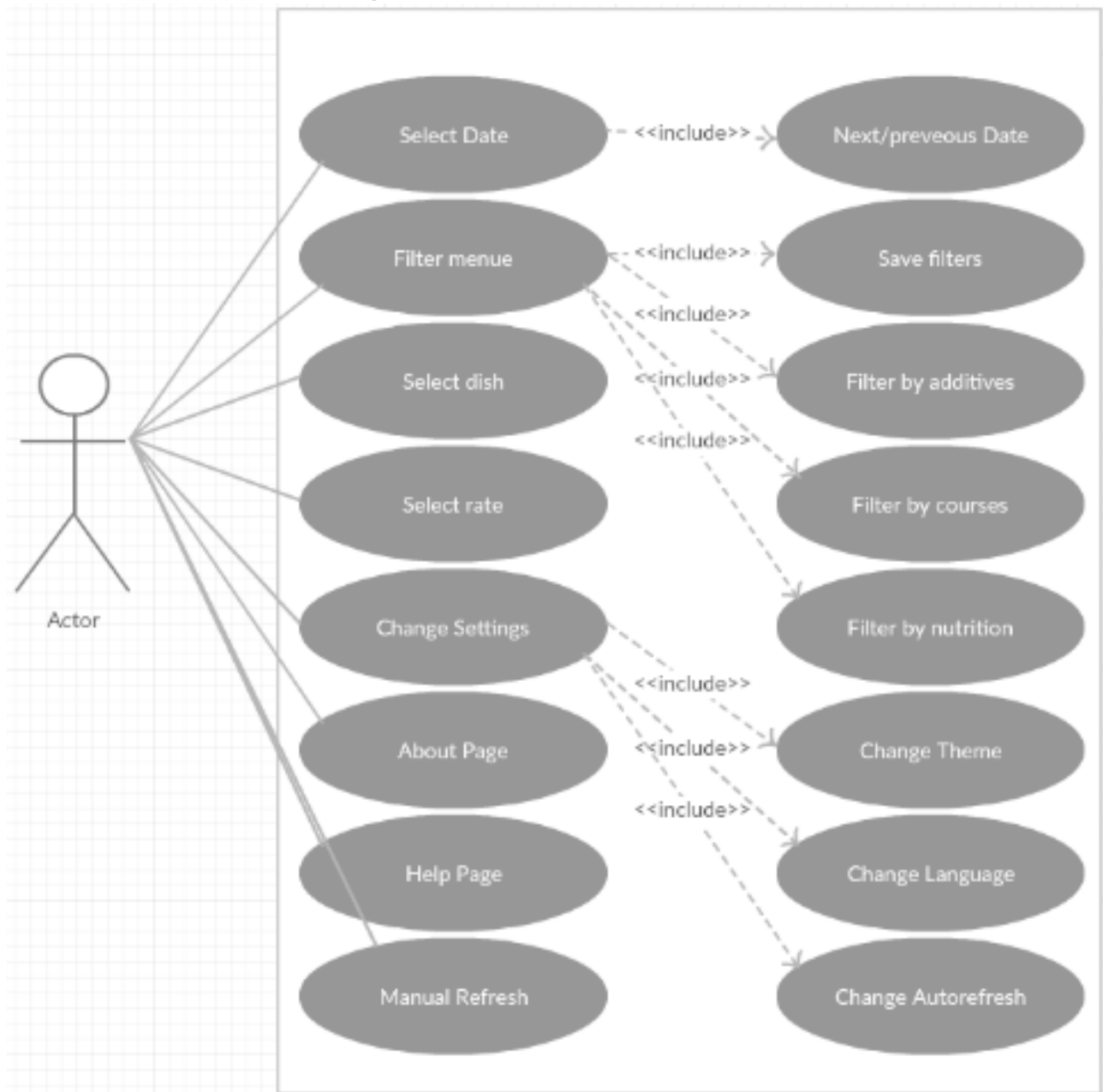
Aufgrund

von hohen veranschlagten Aufwänden zur Realisierung der Abwärtskompatibilität werden ältere Versionen vorerst nicht unterstützt.

# Planung und Konzeption

Mockups können von der Implementierung abweichen.

## Funktionale Anforderungen

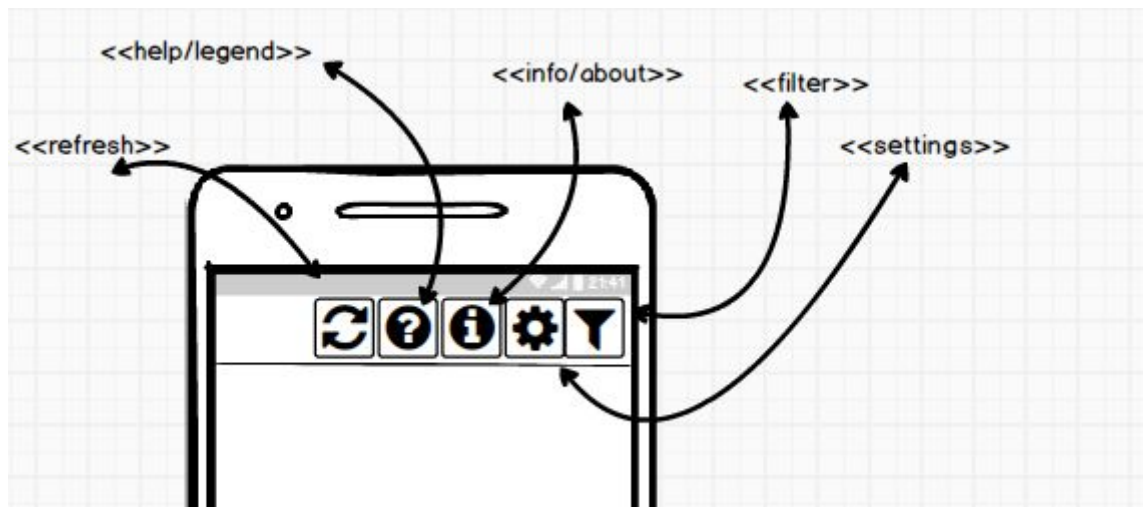


Das oben dargestellte Use Case Diagramm liefert eine grobe Übersicht über die Funktionen von „HAW Mensa“. Die Anwendungsfälle werden im Folgenden Näher Beschrieben.

## Menüleiste

Mit Hilfe der Menüleiste kann der Benutzer

- Den Speiseplan neu laden
- Die Info bzw die Legende anzeigen
- Infos "Über die Applikation" anzeigen / about page
- Zu den Einstellungen Gelangen
- Filter anzeigen und einstellen

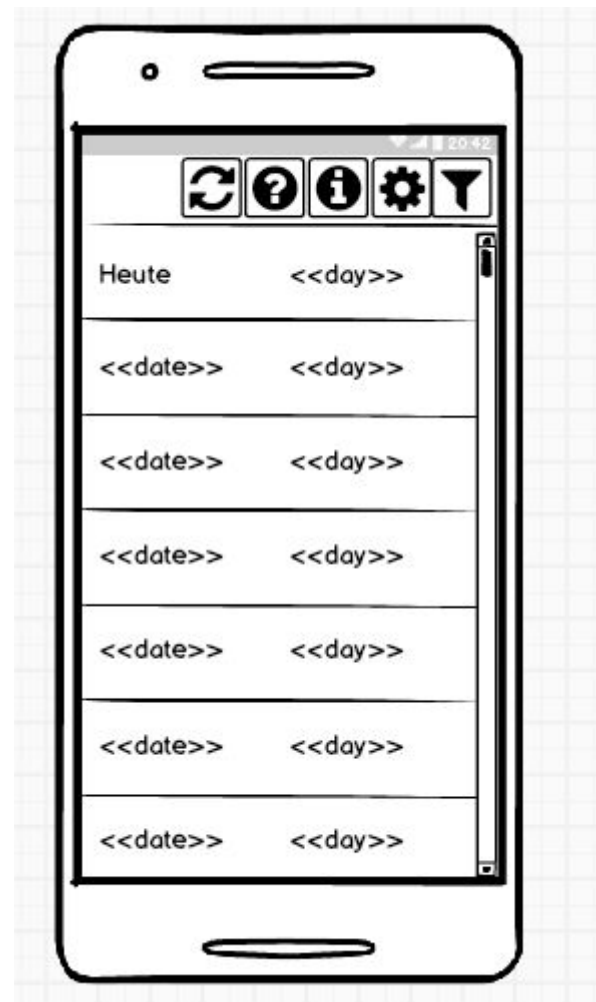


## Auswählen eines Tages

Ein Benutzer Kann über eine Tagesansicht einen Tag auswählen für den die Tageskarte angezeigt wird.

Die Tagesansicht zeigt die Tage vom Aktuellen Datum bis zu 14 tage später an.

In einer Scrollbar können die Tage durchsucht werden.



## Auswählen eines Gerichtes

Bei der Ansicht für einen Tag werden dem Benutzer die entsprechenden Gerichte aufgelistet.

Im wird zum Gericht Name der Entsprechende Preis (für seine Rolle) angezeigt

- Studierende
- Bedienstete
- Gäste

In dieser Ansicht kann der benutzer auf den nächsten beziehungsweise vorherigen Tag wechseln.

Die Gerichte werden in "Gänge" aufgeteilt:

- Hauptgericht
- Dessert
- Salat
- Beilagen
- Suppen

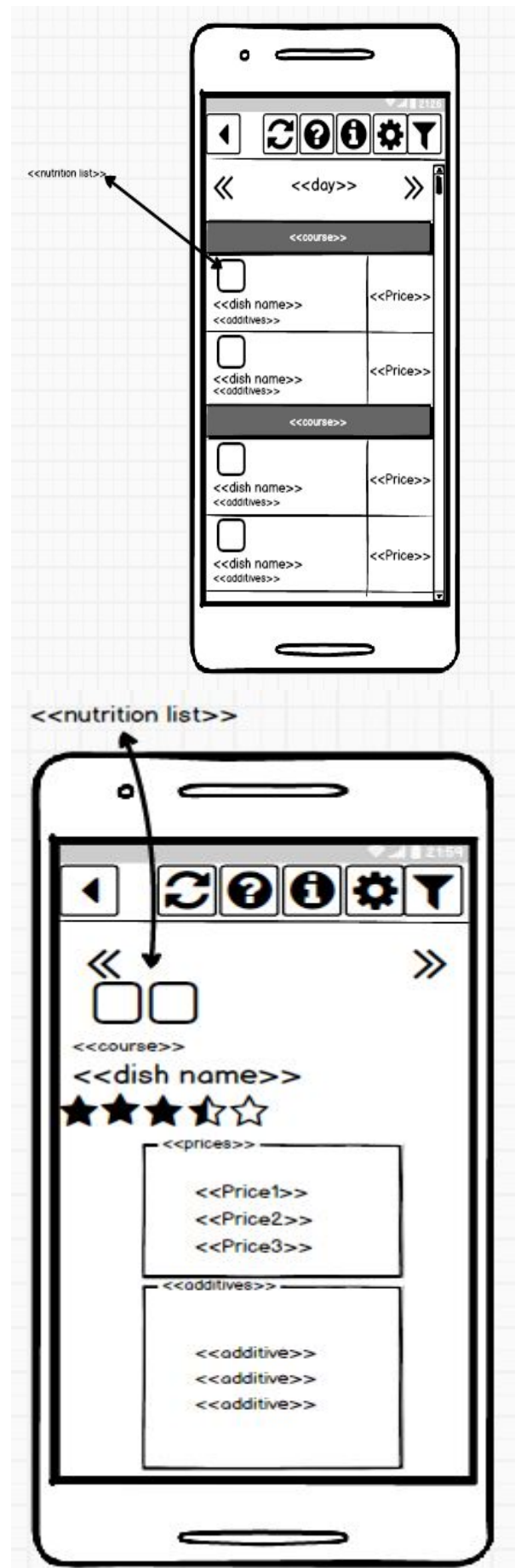
Außerdem wird zu jedem Gerichte die Kategorie angezeigt als auch die Zusatzstoffe und beinhaltenden Allergene in abgekürzter Form aufgelistet (Siehe: Info/Legende).

## Gericht Detailansicht

In der Detaillierten Ansicht eines Gerichtes wird dem Benutzer nochmals der Name des Gerichtes angezeigt, aus welchem Gang das Gericht ist, in welcher Kategorie es und welche Zusatzstoffe und Allergene enthalten sind (nicht abgekürzte form).

Außerdem werden die Preise für alle Rollen Dargestellt.

Zusätzlich kann man ein Gericht bewerten, indem man Sterne vergeben kann.



## **Filtern von Gerichten**

Der Nutzer kann seine Gerichte Inhaltsstoffen wie wie Fisch, Rind etz. filtern.

Außerdem können nach vegetarisch, oder veganen Speisen gefiltert werden.

Oder auch die Auswahl von ausschließlich biologisch erzeugten Rohstoffen ist möglich.

Die einzelnen Filteroptionen werden ODER verknüpft, sodass z.B. vegetarisch als auch Gerichte mit Fisch selektiert werden können (aber keine anderen)

Bei keinem aktiven Filter werden alle Gerichte Selektiert.

Der gesetzte Filter soll gespeichert werden.

## **Einstellungen**

Für die Auswahl der Farbe sind verschiedene Arten vorstellbar.

Das Thema der Applikation könnte individuell angepasst werden. Der Benutzer kann verschiedene Einstellungen vornehmen, wie z.B. das Ändern der Sprache.

## **Über die Applikation**

Hier können zusätzliche Informationen über die Applikation, Lizenzen und den Autor gefunden werden.

## Backend

### stwno.de

Um die Daten für den Mensa speiseplan zu erhalten bietet das Studentenwerk Niederbayern/Oberpfalz die möglichkeit an eine CSV datei zu laden.

Unter: <https://www.stwno.de/infomax/daten-extern/csv/HS-LA/xxx.csv?t=1544084603>

Hierbei ist t= ein timestamp um das richtige Jahr zu erhalten und xxx die entsprechende Kalenderwoche

Aufbau der CSV datei:

	A	B	C	D	E	F	G	H	I	J
1	datum	tag	warengruppe	name	kennz	preis	stud	bed	gast	
2	03.12.2018	Mo	Suppe	Champignonsuppe (3,G,I)	V	0,70 / 0,90 / 1,40	0,70	0,90	1,40	
3	03.12.2018	Mo	HG1	Indisches Gemüsecurry (3,I)	VG	2,00 / 2,80 / 3,60	2,00	2,80	3,60	
4	03.12.2018	Mo	HG2	Köttbullar mit Rahmsoße, Preiselbeeren und Kartoffelpüree (3,A,AA,C,G,I)	S,R	2,80 / 3,40 / 4,40	2,80	3,40	4,40	
5	03.12.2018	Mo	HG3	Marillenknödel mit Vanillesauce (A,AA,C,G)	V	3,10 / 3,90 / 4,60	3,10	3,90	4,60	
6	03.12.2018	Mo	HG4	Schnitzel Wiener Art mit Kartoffelsalat (AA,C,J,I,3)	S	3,10 / 3,90 / 4,60	3,10	3,90	4,60	
7	03.12.2018	Mo	HG5	Bio Rinderrahmgulasch mit Bio-Nudeln (AA,G)	R,BL	4,70 / 5,70 / 6,20	4,70	5,70	6,20	
8	03.12.2018	Mo	B1	Kaisergemüse (3,I)	VG	0,80 / 1,00 / 1,50	0,80	1,00	1,50	
9	03.12.2018	Mo	B2	Bio-Basmatireis	B,VG	0,80 / 1,00 / 1,50	0,80	1,00	1,50	
10	03.12.2018	Mo	B3	Salatmix	VG	0,90 / 1,10 / 1,60	0,90	1,10	1,60	
11	03.12.2018	Mo	B4	Waffelkartoffeln (8,A,AA,I)	VG	1,00 / 1,10 / 1,60	1,00	1,10	1,60	
12	03.12.2018	Mo	B5	Kartoffelsalat (J,3,I)	VG	1,00 / 1,10 / 1,60	1,00	1,10	1,60	
13	03.12.2018	Mo	B6	Salat Buffet (1,3,6,9,16,C,G,J,L)	V	3,00 / 3,50 / 4,20	3,00	3,50	4,20	
14	03.12.2018	Mo	N1	Beerenquark (G)	V,A	0,80 / 1,00 / 1,50	0,80	1,00	1,50	
15	03.12.2018	Mo	N2	Mandelpudding (1,G,H,HA)	V	0,80 / 1,00 / 1,50	0,80	1,00	1,50	
16	03.12.2018	Mo	N3	Mousse Erdbeer-Rhabarber (G)	V	1,10 / 1,40 / 1,60	1,10	1,40	1,60	

### openmensa.org

Alternativ können die Daten von der openmensa.org api erhalten werden.

unter <https://doc.openmensa.org/api/v2/> lässt sich die dokumentation dieser anzeigen.

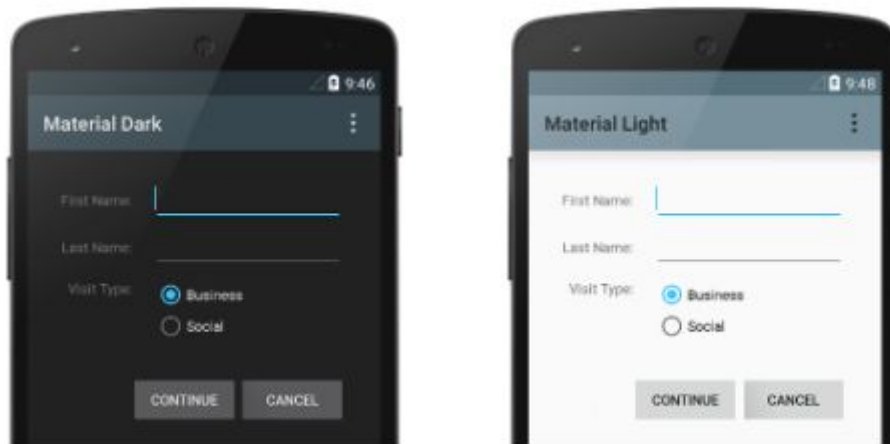
Tatsächlich stehen mehr funktionalitäten als beschrieben zur verfügung.

mit dem parameter ?starts=date kann eine ganze liste von tagen selektiert werden (die nächsten 5 tage inklusive des angegebenen Tages).



## Styleguide

Die App soll in (mindestens) 2 verschiedenen Designs implementiert werden.  
Hiebei soll ein Dunkles und ein Helles Theme erstellt werden.



Die Typography, Iconography und Farbdefinition ist festgelegt und wird im Folgenden Beschrieben: (wird während der Entwicklung festgelegt)

### Typography

Font:

LetterSpacing : 0

Attribute name	Default style
Font/FontFamily	Roboto
Letterspacing	0
textAppearanceHeadline1	Light 96sp
textAppearanceHeadline2	Light 60sp
textAppearanceHeadline3	Regular 48sp
textAppearanceHeadline4	Regular 34sp
textAppearanceHeadline5	Regular 24sp
textAppearanceHeadline6	Regular 20sp
textAppearanceSubtitle1	Regular 16sp
textAppearanceSubtitle2	Medium 14sp
textAppearanceBody1	Regular 16sp
textAppearanceBody2	Regular 14sp
textAppearanceCaption	Regular 12sp
textAppearanceButton	Medium all caps 14sp
textAppearanceOverline	Medium all caps 12sp

## Iconography



## Colors

Für die Applikation sind (mindestens) 2 verschiedene Themes geplant. Hierbei sind die die verschiedene Hintergrund und Vordergrundfarben fest definiert

### Theme 1: (Example can change during development)

Attribute Name	Description	Default Value
colorPrimary	The color displayed most frequently across your app's screens and components.	#6200EE
colorPrimaryVariant	A tonal variation of the primary color.	#3700B3
colorOnPrimary	A color that passes accessibility guidelines for text/iconography when drawn on top of the primary color.	#FFFFFF
colorSecondary	The secondary branding color for the app, usually an accented complement to the primary branding color.	#03DAC6
colorSecondaryVariant	A tonal variation of the secondary color.	#018786
colorOnSecondary	A color that passes accessibility guidelines for text/iconography when drawn on top of the secondary color.	#000000

### Theme 2: (Example change during development)

Attribute Name	Description	Default Value
colorPrimary	The color displayed most frequently across your app's screens and components.	#6200EE

colorPrimaryVariant	A tonal variation of the primary color.	#3700B3
colorOnPrimary	A color that passes accessibility guidelines for text/iconography when drawn on top of the primary color.	#FFFFFF
colorSecondary	The secondary branding color for the app, usually an accented complement to the primary branding color.	#03DAC6
colorSecondaryVariant	A tonal variation of the secondary color.	#018786
colorOnSecondary	A color that passes accessibility guidelines for text/iconography when drawn on top of the secondary color.	#000000

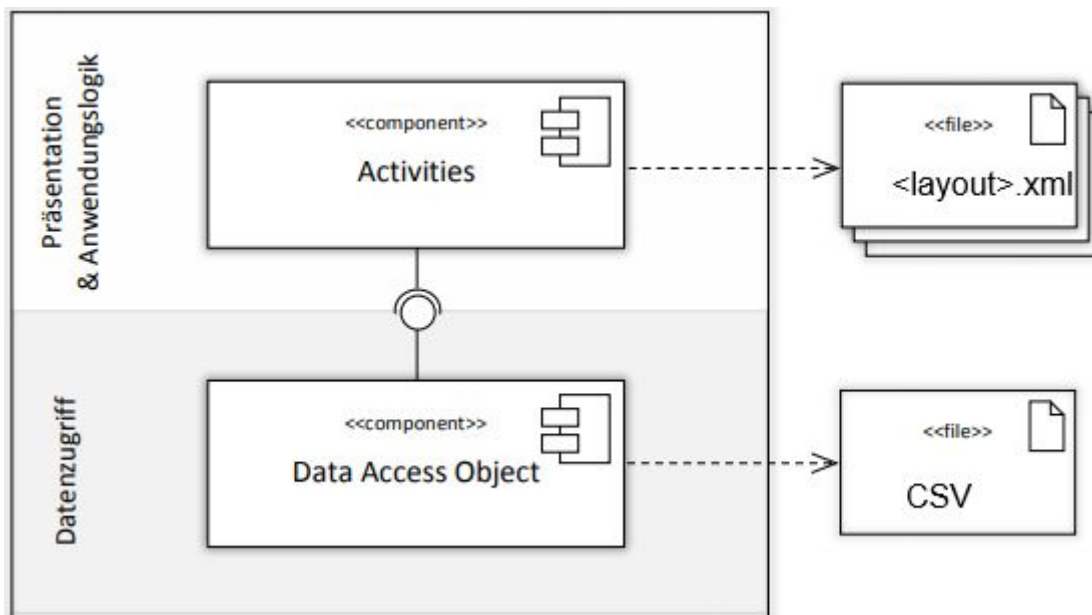
## Release und Planung

Die folgende Tabelle zeigt die Zuordnung der zuvor beschriebenen Funktionen und die dafür notwendigen Tätigkeiten zu den geplanten Releases. Die Entwicklung von JUnit-Tests ist in der Tabelle nicht explizit aufgeführt, da sie parallel zur Entwicklung vorgenommen wurde.

Release	Funktion mit den wesentlichen Tätigkeiten
0.1	<ul style="list-style-type: none"><li>• Einrichtung der Entwicklungsumgebung</li><li>• Anlegen der Projektstruktur</li><li>• Bereitstellung einer ausführbaren Version</li></ul>
0.2	<ul style="list-style-type: none"><li>• Parser für CSV datei</li></ul>
0.3	<ul style="list-style-type: none"><li>• Navigationsmenü</li><li>• Filtern von Gerichten</li><li>• Filtern von Gerichten mit Parser integrieren</li></ul>
0.4	<ul style="list-style-type: none"><li>• Wochenansicht mit "highlights" für heute (Auswahl eines Tages)</li></ul>
0.5	<ul style="list-style-type: none"><li>• Tagesansicht (Auswahl eines Gerichtes)</li></ul>
0.6	<ul style="list-style-type: none"><li>• Gericht Ansicht (Gericht Detailansicht)</li></ul>
0.7	<ul style="list-style-type: none"><li>• Designfragen klären</li><li>• Themes implementieren</li></ul>
1.0	<ul style="list-style-type: none"><li>• Einstellungen</li><li>• About page</li><li>• Help page</li></ul>
1.x	<ul style="list-style-type: none"><li>• Optionale Funktionen</li><li>• Bugfixing</li><li>• Refactoring</li><li>• Optimizing</li><li>• Themes erweitern</li><li>• Sprachen erweitern</li></ul>

## Implementierung

Nach Abschluss der Anforderungsanalyse und Entwurf der Oberflächen wurde der strukturelle Aufbau der mobilen Applikation erörtert. Wie in der folgenden Abbildung zu sehen ist, wurde die Applikation in zwei grundlegende Schichten unterteilt.



Die Präsentations- und Anwendungslogik Schicht soll sich dabei um die Erzeugung der Oberflächenkomponenten kümmern, die Benutzeroberfläche darstellen, die Benutzeraktionen entgegennehmen und die jeweilige Funktion ausführen.

Bei der Realisierung dieser Schicht wurde, entsprechend der empfohlenen Vorgehensweise der Android Developer-Webseite, auf eine konsequente Trennung von Ressourcen und Programmlogik geachtet.

Die Definition und Konfiguration der Oberflächenkomponenten, deren Layout sowie die Definition von Zeichenketten, Farben und Größenangaben wurden daher durch den deklarativen Ansatz über XML-Dateien vorgenommen. Neben den XML-Dateien wird diese Schicht hauptsächlich durch Activity-Klassen realisiert, auf die im folgenden Unterkapitel genauer eingegangen wird.

Zur Erledigung der meisten Funktionen von der App benötigen die Activities Zugriff auf Persistenz Operationen. Die Datenzugriffsschicht bietet daher Schnittstellen zur Speicherung und Bereitstellung der Domänenobjekte an. Eine genauere Beschreibung der Domänenobjekte folgt nach dem Activity-Unterkapitel. Für die Umsetzung der Datenzugriffsschicht ist das Entwurfsmuster der Data Access Objects (DAO) vorgesehen. Dieses sorgt einerseits für eine Entkopplung der eigentlichen Datenspeicherung von der Anwendung. Andererseits abstrahiert das Muster den Zugriff auf die zugrundeliegende Datenhaltungskomponente der Anwendung. Auf die konkrete Umsetzung wird im letzten Unterkapitel näher eingegangen.

## **Activities**

Im Paket `de.haw_landshut.lackmann.haw_mensa` sind die Activities enthalten, die jeweils einer Bildschirmansicht der mobilen Applikation entsprechen. Eine im Vorfeld durchgeführte Analyse zeigte eine Reihe von Gemeinsamkeiten zwischen den geplanten Activities auf, die beim Entwurf der Klassenstruktur in Form gemeinsamer Basisklassen berücksichtigt wurden.

Alle Activities teilen sich elemente der Navigations bzw Menüleiste.

## **Data access object**

Die geplante Datenzugriffsschicht bietet, wie aus dem Namen hervorgeht, Zugriff auf Daten die Für den Speiseplan erforderlich sind. Da die Verwendeten Daten relativ überschaubar sind und im CSV format über einen GET request zur verfügung stehen, werden Die daten für die Applikation nicht langfristig sondern nur im Cash gespeichert.

Das Data access object kümmert sich darum die daten aktuell zu erhalten und um das parsen der CSV datei und stellt simple Methoden bereit um die Daten zu erhalten.

## Test

Zur Sicherstellung einer niedrigen Fehlerrate ist ein dreistufiges Testvorgehen geplant. Auf erster Stufe sind White-Box-Tests mit Hilfe von JUnit und dem Android Test Framework vorgesehen. Diese sollen nicht nur zur Verifikation einer gerade entwickelten oder geänderten Funktion dienen, sondern vor allem als Regressionstest eingesetzt werden. Bevor der Quellcode in die Versionsverwaltung überführt wird, soll ein sogenannter Smoke-Test in manueller Form durchgeführt werden, der die zweite Stufe repräsentiert. Hierbei soll mit geringem Aufwand sichergestellt werden, dass sich die grundlegenden Funktionen der App korrekt verhalten und die Navigation zwischen den wichtigsten Oberflächen gemäß der Spezifikation abläuft.

### Smoke-Tests

Nach jeder abgeschlossenen Quellcode Änderung soll ein Smoke-Test in manueller Form durchgeführt werden, dessen Inhalte unabhängig von dem Anwendungsbereich der Änderung sind. Die Durchführungszeit des Tests beträgt für geübte Benutzer etwa 4 Minuten. Der Test soll dabei die folgende Aktionen enthalten und wird als erfolgreich gewertet, wenn stets das beschriebene Soll-Verhalten des Systems eintritt:

	Aktion	Soll Verhalten
1	Installation der App	App lässt sich fehlerfrei installieren.
2	Neuinstallation der App App lässt sich fehlerfrei installieren	App lässt sich fehlerfrei re-installieren.
3	Deinstallation der App	App lässt sich ohne rückstände deinstallieren.
4	Tag auswählen	Tag lässt sich auswählen und man landet in der Gerichtsliste für den entsprechenden Tag.
5	Gericht auswählen	Gericht lässt sich auswählen und man landet in der Detaillierten ansicht für das Gericht.
6	navigieren zur Information page	Man kann über das Navigationsmenü zur Informations page gelangen.
6	navigieren zu Filter	Man kann über das Navigationsmenü zum Filter gelangen.
6	navigieren zu einstellungen	Man kann über das Navigationsmenü zu den Einstellungen gelangen.
7	refresh	Man kann über das Navigationsmenü die daten erfrischen.
8	Filter festlegen	Filter lassen sich festlegen und ändern die angezeigten daten.
9	nächsten Tag auswählen	In der Gerichteliste lässt sich auf den nächsten Tag

		schalten.
10	vorherigen Tag auswählen	In der Gerichteliste lässt sich auf den vorherigen Tag schalten.
11	vorheriges Gericht auswählen	in der detaillierten gerichte ansicht lässt sich auf das vorherige Gericht schalten.
12	App läuft in Systemsprache	Beim ändern der Systemsprache ändert sich die Sprache der Applikation
13	Theme ändern	Das Theme lässt sich in den Einstellungen auf Dark und Light ändern
14	Auswahl zwischen Backend	Das Backend lässt sich zwischen CSV Lösung von stwno.de und REST Lösung von openmensa.org wechseln



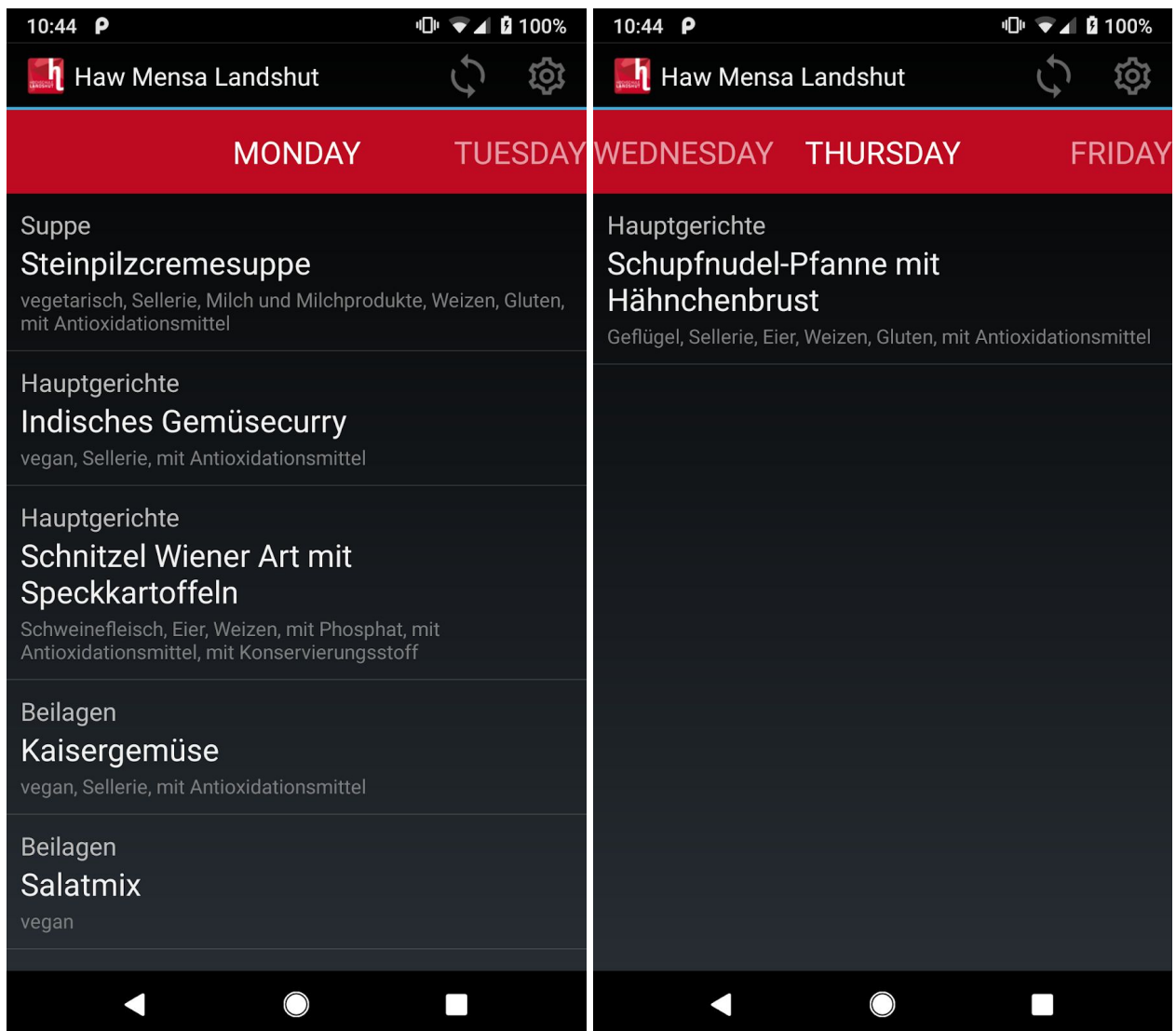
# Geplante Features

## Auswahl verschiedener Mensen


Durch den umstieg vom CSV basierten backend auf auf das REST backend Openmensa.org/api/v2 können durch integration des Quellcodes mehrer verschiedene Mensen ausgewählt werden.



Die SCV basierte lösung müsste dann weggeschmissen werden.

## Screenshots



10:44 P

Haw Mensa Landshut



MONDAY

TUESDAY

Suppe

Steinpilzcremesuppe

vegetarisch, Sellerie, Milch und Milchprodukte, Weizen, Gluten, mit Antioxidationsmittel

Hauptgerichte

Indisches Gemüsecurry

vegan, Sellerie, mit Antioxidationsmittel

Hauptgerichte

Schnitzel Wiener Art mit Speckkartoffeln

Schweinefleisch, Eier, Weizen, mit Phosphat, mit Antioxidationsmittel, mit Konservierungsstoff

Beilagen


Kaisergemüse

vegan, Sellerie, mit Antioxidationsmittel


Beilagen



Salatmix

vegan



22:44 P

Haw Mensa Landshut



MONTAG

DIENSTAG

Suppe

Steinpilzcremesuppe

vegetarisch, Sellerie, Milch und Milchprodukte, Weizen, Gluten, mit Antioxidationsmittel

Hauptgerichte

Indisches Gemüsecurry

vegan, Sellerie, mit Antioxidationsmittel

Hauptgerichte

Schnitzel Wiener Art mit Speckkartoffeln

Schweinefleisch, Eier, Weizen, mit Phosphat, mit Antioxidationsmittel, mit Konservierungsstoff

Beilagen


Kaisergemüse

vegan, Sellerie, mit Antioxidationsmittel

Beilagen

Salatmix

vegan



22:45 P

100 %

## Einstellungen

### FILTER

---

Sollte beinhalten

### DATENQUELLE

---

API Server

<http://openmensa.org/api/v2/>

### DESIGN

---

Theme

Light Theme

### ÜBER

---

Lizenz

GNU General Public License, Version 3

---

