

# Mensa-App

Applikation für Mitteilung und kollektiven Austausch von  
Speiseplaninformationen in universitären Gastronomieeinrichtungen

## Pflichtenheft

Praxis der Softwareentwicklung  
Sommersemester 2023

Alexander Albers, Peer Booken, Elena Häußler,  
Alexander Kutschera, Jonatan Ziegler

26. Mai 2023

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>3</b>
<b>2 Produkteinsatz</b>	<b>4</b>
2.1 Zielgruppe . . . . .	4
2.2 Betriebsbedingungen . . . . .	4
<b>3 Zielbestimmung</b>	<b>5</b>
3.1 Musskriterien . . . . .	5
3.2 Kannkriterien . . . . .	5
3.3 Abgrenzungskriterien . . . . .	5
<b>4 Produktumgebung</b>	<b>6</b>
4.1 App . . . . .	6
4.2 Server . . . . .	6
4.3 Datenbank . . . . .	7
4.4 API . . . . .	7
4.5 Mensa-Webseite . . . . .	7
4.6 Bildhoster . . . . .	7
<b>5 Produktübersicht</b>	<b>8</b>
<b>6 Produktfunktionen</b>	<b>10</b>
6.1 Speiseplanansicht . . . . .	12
6.2 Detailansicht . . . . .	18
6.3 Bilder . . . . .	22
6.4 Filterfunktion . . . . .	26
6.5 Favoriten . . . . .	29
6.6 Einstellungen . . . . .	30
6.7 Benachrichtigungen . . . . .	32
6.8 Server . . . . .	34
<b>7 Nichtfunktionale Anforderungen</b>	<b>39</b>
<b>8 Produktdaten</b>	<b>40</b>
8.1 App . . . . .	40
8.2 Server und Datenbank . . . . .	40
8.3 Nutzerdaten . . . . .	41
8.4 Export-Daten . . . . .	41

<b>9 Globale Testfälle</b>	<b>42</b>
9.1 Funktionstests . . . . .	42
9.2 Sequenztests . . . . .	48
<b>10 Benutzerschnittstelle</b>	<b>52</b>
10.1 Ansichten und Dialoge . . . . .	52
10.2 Fehlermeldungen . . . . .	57
10.3 Entwürfe grafische Benutzerschnittstelle . . . . .	58
<b>11 Glossar</b>	<b>65</b>

## Abbildungsverzeichnis

1 Produktumgebung . . . . .	6
2 Usecase Diagramm des Nutzers . . . . .	8
3 Usecase Diagramm der serverseitigen Features . . . . .	9
4 UI-Übersicht . . . . .	52
5 Speiseplanansicht Galerie . . . . .	58
6 Speiseplanansicht Liste . . . . .	59
7 Favoritenansicht . . . . .	59
8 Einstellungsansicht . . . . .	60
9 Detailansicht . . . . .	60
10 Galeriedialog . . . . .	61
11 Filterdialog . . . . .	61
12 Kalenderdialog . . . . .	62
13 Mensadialog . . . . .	62
14 Bewertungsdialog . . . . .	62
15 Hochladedialog . . . . .	62
16 Meldedialog . . . . .	62
17 Geschlossene Mensa . . . . .	63
18 Keine Filterergebnisse . . . . .	63
19 Keine Daten . . . . .	64

Alle in diesem Dokument angegebenen Web-Links wurden zuletzt am auf dem Deckblatt angegebenen Datum abgerufen.

### 1 Einleitung

Für viele Angehörige der Hochschulen in Karlsruhe und Pforzheim ist es von großer Bedeutung, sich über die aktuellen Tagesgerichte in der Mensa zu informieren. Möchte man diese schon vorab in Erfahrung bringen und sich nicht im Foyer der Mensa zu den dort angebrachten Bildschirmen durchdrängen, bietet die Webseite des Studierendenwerks einen Online-Speiseplan. Allerdings besteht insbesondere auf mobilen Geräten Verbesserungsbedarf hinsichtlich dessen Übersichtlichkeit. Zwar existieren vereinzelte Apps, die das Bedürfnis einer einfachen Speiseplanauskunft abdecken, jedoch weisen diese erhebliche Schwächen auf, beispielsweise werden nur Preise für Studierende angezeigt, es gibt keine Bilder der angebotenen Gerichte oder man kann keine Bewertungen abgeben.

Aufgrund dieses Zustands wird im Rahmen dieses Projekts eine Android-App entwickelt, die die Speisepläne der Menschen des Studierendenwerks Karlsruhe anzeigen kann.

Die App soll dabei die Betrachtung des Speiseplans möglichst angenehm und einfach machen, gleichzeitig aber möglichst viele Informationen darbieten. Um die App genauer auf die Bedürfnisse und Wünsche der Zielgruppe zuzuschneiden, wurde eine Umfrage durchgeführt, bei der die Studierenden und Mitarbeitende ihre Wünsche auflisten, Vorschläge einreichen und Wertungen abgeben konnten. Die Ergebnisse sind in die Erstellung dieses Pflichtenhefts eingeflossen.

Im Folgenden werden die Funktionen und Spezifikationen der App genauer beschrieben.

## 2 Produkteinsatz

### 2.1 Zielgruppe

Die Zielgruppe umfasst alle Personen, die in den Mensen des Studierendenwerks Karlsruhe ihr Mittagessen zu sich nehmen. Dazu gehören Studierende, Mitarbeitende, Schüler und Gäste.

Das Studierendenwerk Karlsruhe betreibt dabei folgende Mensen:

- Mensa am Adenauerring
- Menseria Schloss Gottesau
- Mensa Moltke
- Menseria Moltkestraße
- Menseria Erzbergerstraße
- Mensa Tiefenbronner Straße
- Menseria Holzgartenstraße

### 2.2 Betriebsbedingungen

Die Mensen des Studierendenwerks haben je mindestens eine Linie, an denen meist ein oder zwei Gerichte angeboten werden. An manchen Linien werden zusätzlich Beilagen angeboten, die je nach Linie einzeln bepreist werden oder schon im Preis des Gerichts inbegriffen sind. Dabei gibt es für jedes Gericht unterschiedliche Preise für Studierende, Mitarbeitende, Schüler und Gäste. Im Folgenden bezeichnet Preisklasse je die Kategorien der Preise für Studierende, Mitarbeitende, Schüler und Gäste.

Ein Gericht bzw. eine Beilage hat immer eine Gerichtsbezeichnung, also einen Namen. Die meisten Gerichte und Beilagen der Mensa enthalten Allergene oder Zusatzstoffe, die das Studierendenwerk angeben muss.

Außerdem gibt das Studierendenwerk an, welchen Gerichtstyp ein Gericht hat. Dabei wird unterschieden, ob ein Gericht Rindfleisch, Schweinefleisch oder Fisch enthält oder ob es vegetarisch oder vegan ist. Es werden allerdings nicht alle Gerichte vom Studierendenwerks einer Kategorie zugeordnet, weshalb hier die Kategorie „Keine Angabe“ eingefügt wurde, die sowohl vegane und vegetarische Gerichte, als auch Gerichte mit Fleisch, enthalten kann.

Für jede Mensa gibt es jeden Tag einen Speiseplan, der die Gerichte umfasst, die es an den verschiedenen Linien gibt. Die dazugehörigen Daten werden im Folgenden als Speiseplandaten bezeichnet. In den Mensen gibt es die einzelnen Gerichte in teils regelmäßigen und teils unregelmäßigen Abständen. Um dies zu vermitteln, ist das letztes Angebotsdatum das Datum, an dem das Gericht zum letztem Mal angeboten wurde. Außerdem bezeichnet die Häufigkeit wie oft ein Gericht in drei Monaten in einer Mensa angeboten wurde. Daraus lassen sich verschiedene Häufigkeitsklassen ableiten, die besagen, ob ein Gericht neu ist, es das Gericht selten gibt oder ob es das Gericht täglich gibt. Nicht alle Gerichte fallen in eine der Häufigkeitsklassen.

### 3 ZIELBESTIMMUNG

---

## 3 Zielbestimmung

Ziel dieses Projekts ist es, eine Android-App (im folgenden App) zu entwickeln, mithilfe derer man sich über den Speiseplan eines bestimmten Tages informieren kann. Dabei werden im Folgenden Kriterien für diese App definiert, die in drei Kategorien unterteilt werden:

- **Musskriterien** bezeichnen Anforderungen an die App, die zwingend erfüllt werden müssen.
- **Kannkriterien** bezeichnen Anforderungen an die App, die im Zuge des Projekts umgesetzt werden können, aber nicht müssen.
- **Abgrenzungskriterien** bezeichnen denkbare Funktionalitäten einer App, die explizit nicht umgesetzt werden.

### 3.1 Musskriterien

[MK01] Gerichte anzeigen

- zu einem bestimmten Tag bis zu einem Monat im Voraus
- für alle Menschen des Studierendenwerk Karlsruhe
- inklusive ihrer Allergene und Zusatzstoffe

[MK02] Preise für die verschiedenen Preisklassen anzeigen

[MK03] Gerichte nach Gerichtstyp und Allergenen filtern

[MK04] Bilder zu Gerichten verlinken und anzeigen

[MK05] Von Nutzern verlinkte Bilder melden

### 3.2 Kannkriterien

[KK01] Lightmode / Darkmode der App

[KK02] Gerichte favorisieren

[KK03] Gerichte mit ein bis fünf Sternen bewerten

[KK04] Bilder zu Gerichten als hilfreich oder nicht hilfreich bewerten

[KK05] Häufigkeit der Gerichte anzeigen

[KK06] Benachrichtigungen für favorisierte Gerichte

[KK07] Favoriten und Einstellungen im- und exportieren

[KK08] Gerichte nach Preis, Bewertung, Favoriten und Häufigkeit filtern

[KK09] Gerichte nach Linie, Bewertung, Preis und Häufigkeit sortieren

### 3.3 Abgrenzungskriterien

[AK01] Accountstrukturen

[AK02] Kommentare zu Gerichten

[AK03] Anzeigen von Gerichten der Campus-Nord-Mensa

## 4 Produktumgebung

Die Anwendung wird im Client-Server-Modell mit einem Fat-Client entwickelt. Mithilfe des Clients können Nutzer mit dem Produkt interagieren. Der Server dient zur Bereitstellung und Synchronisierung der Daten, welche von den Clients benötigt werden.

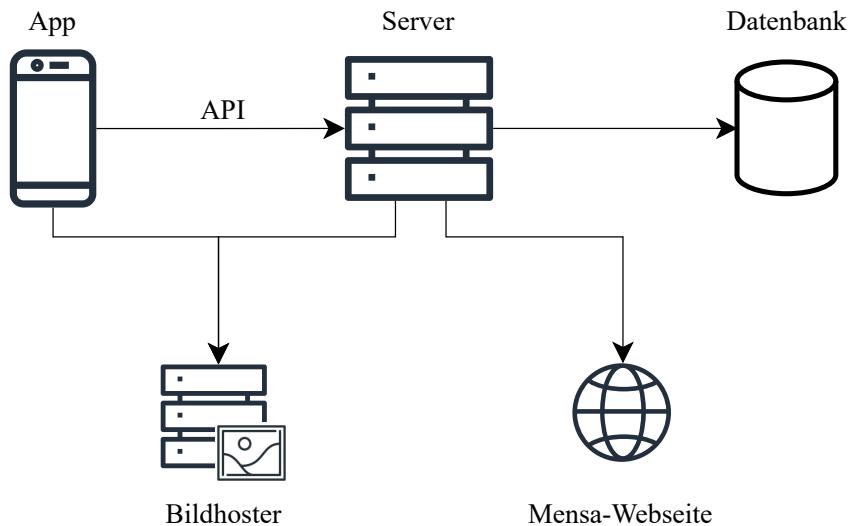


Abbildung 1: Produktumgebung

### 4.1 App

Als Client dient eine App, welche mithilfe des Flutter-Framework<sup>1</sup> Version 3.10+ mit primärer Zielplattform Android entwickelt wird. Die App muss auf gängigen Android-Geräten mit einer Android SDK Version von mindestens 19 lauffähig sein.

### 4.2 Server

Der Server wird in Rust<sup>2</sup> Edition 2021 implementiert und soll als Docker<sup>3</sup>-Container auf der Docker Engine 23 lauffähig sein. Damit ein ordnungsgemäßer Betrieb garantiert werden kann, muss ein Server die folgenden Anforderungen erfüllen:

- mindestens vier CPU-Kerne der AMD Ryzen™ Produktfamilie<sup>4</sup> oder einer Vergleichbaren mit einer Taktrate von mindestens 3 GHz
- mindestens 2 GB Arbeitsspeicher
- mindestens 20 GB freier Festplattenspeicher
- eine Verbindung zum Internet mit mindestens 100 MBit/s Bandbreite in beide Richtungen

---

<sup>1</sup><https://flutter.dev/>

<sup>2</sup><https://www.rust-lang.org/>

<sup>3</sup><https://www.docker.com/>

<sup>4</sup><https://www.amd.com/en/products/specifications/processors>

### 4.3 Datenbank

Als Datenbank wird Postgresql<sup>5</sup> Version 15 verwendet. Ein ordnungsgemäßer Betrieb geht davon aus, dass die Datenbank auf derselben physischen Ressource wie der Server in Betrieb genommen wird.

### 4.4 API

Zur Kommunikation zwischen App und Server wird eine GraphQL<sup>6</sup>-API verwendet.

Für den ordnungsgemäßen Betrieb muss eine Transportverschlüsselung verwendet werden, welche jedoch nicht vom Produkt bereitgestellt wird, sondern durch z.B. einen Reverse-Proxy vorgeschaltet werden muss. Des Weiteren muss der Client sich bei einer API-Anfrage mithilfe eines API-Schlüssels authentifizieren, sofern diese Anfrage eine Änderung von Daten in der Datenbank zur Folge hat. Somit soll ein Missbrauch solcher Anfragen erschwert werden.

### 4.5 Mensa-Webseite

Zum Erlangen der Speiseplandaten soll der Server die Webseite des Studierendenwerk Karlsruhe<sup>7</sup> auslesen.

### 4.6 Bildhoster

Um das Verwalten der Bilder von Nutzern zu vereinfachen, werden diese auf einen externen Bildhoster wie z.B. Flickr<sup>8</sup> ausgelagert. Der Nutzer muss dort seine Bilder selbstständig hochladen und kann diese in der App lediglich verlinken. Der Server überprüft dabei, ob ein vom Nutzer bereitgestellter Link auf ein valides Bild verweist. Dabei wird unter anderem die Lizenz des Bildes überprüft. Der Client lädt Bilder direkt vom Bildhoster.

---

<sup>5</sup><https://www.postgresql.org/>

<sup>6</sup><https://graphql.org/>

<sup>7</sup><https://www.sw-ka.de/de/hochschulgastronomie/speiseplan/>

<sup>8</sup><https://www.flickr.com/>

## 5 Produktübersicht



Abbildung 2: Ein Usecase Diagramm, welches die Interaktionen des Nutzers mit dem Produkt beschreibt. Die Akteure Server und Client sind aus Gründen der Übersichtlichkeit nicht dargestellt. Für den Client gilt, dass dieser genauso mit den Usecases interagiert, wie der Nutzer. Bei den Usecases, die mit dem Server interagieren, wurde ein \* an den Namen geschrieben. Falls nicht explizit anders angegeben, stellen alle Pfeile mit gestrichelter Linie „extends“ Beziehungen dar.

## 5 PRODUKTÜBERSICHT

---

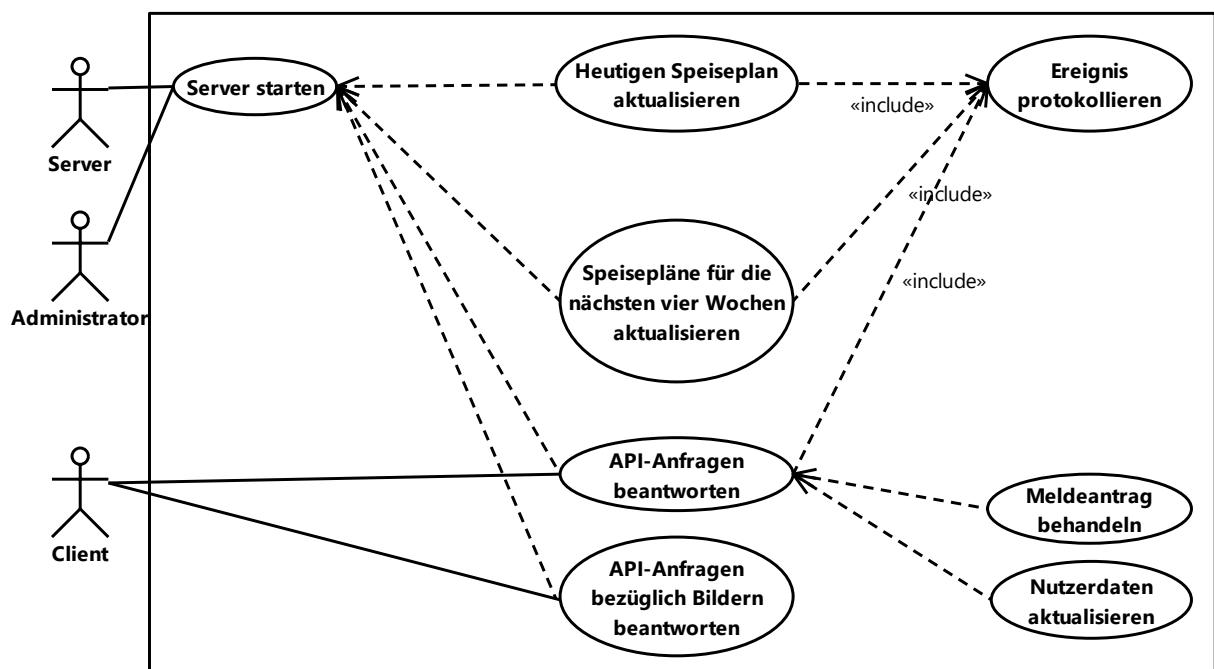


Abbildung 3: Ein Usecase Diagramm, welches die Interaktionen zwischen dem Server und dem Client beschreibt. Falls nicht explizit anders angegeben, stellen alle Pfeile mit gestrichelter Linie „extends“ Beziehungen dar.

## 6 Produktfunktionen

### 6.1 Speiseplanansicht

- [F101] App starten
- [F102] Speiseplanansicht anzeigen
- [F103] Speiseplan aktualisieren
- [F104] Mensa wechseln
- [F105] Tag ändern
- [F106] Ansicht von Galerie zu Liste wechseln
- [F107] Ansicht von Liste zu Galerie wechseln
- [F108] Vorherigen Tag anzeigen
- [F109] Nächsten Tag anzeigen

### 6.2 Detailansicht

- [F201] Detailansicht für Gerichte anzeigen
- [F202] Detailansicht schließen
- [F203] Bewertung abgeben
- [F204] Bewertung anpassen
- [F205] Bild hochladen
- [F206] Allergene und Zusatzstoffe einblenden
- [F207] Allergene und Zusatzstoffe ausblenden

### 6.3 Bilder

- [F301] Galeriedialog öffnen
- [F302] Galeriedialog verlassen
- [F303] Bild melden
- [F304] Bilder nach rechts durchwechseln
- [F305] Bilder nach links durchwechseln
- [F306] Aufwertung zu einem Bild hinzufügen
- [F307] Aufwertung eines Bildes entfernen
- [F308] Abwertung zu einem Bild hinzufügen
- [F309] Abwertung eines Bildes entfernen

### 6.4 Filterfunktion

- [F401] Filterdialog öffnen
- [F402] Filterdialog schließen
- [F403] Filteroptionen auswählen
- [F404] Filterkonfiguration speichern
- [F405] Filter zurücksetzen
- [F406] Filter aktivieren
- [F407] Filter deaktivieren

### 6.5 Favoriten

- [F501] Favoritenansicht anzeigen
- [F502] Gericht favorisieren

## 6 PRODUKTFUNKTIONEN

---

[F503] Gericht defavorisieren

### 6.6 Einstellungen

[F601] Einstellungsansicht anzeigen

[F602] Farbschema einstellen

[F603] Preisklasse einstellen

[F604] Einstellungen und Favoriten exportieren

[F605] Einstellungen und Favoriten importieren

### 6.7 Benachrichtigungen

[F701] Benachrichtigungen aktivieren

[F702] Benachrichtigungen deaktivieren

[F703] Benachrichtigungszeitpunkt wählen

### 6.8 Server

[F801] Server starten

[F802] Heutigen Speiseplan aktualisieren

[F803] Speisepläne für die nächsten vier Wochen aktualisieren

[F804] API-Anfragen beantworten

[F805] API-Anfragen bezüglich Bildern beantworten

[F806] Nutzerdaten aktualisieren

[F807] Meldeantrag behandeln

[F808] Ereignis protokollieren

## 6.1 Speiseplanansicht

### [F101] App starten

**Kurzbeschreibung:** Die Gerichte sollen nach dem Starten der App angezeigt werden.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App ist nicht gestartet.

**Nachbedingung Erfolg:** Die Gerichte werden angezeigt.

**Nachbedingung Fehlschlag:** Es wird eine Fehlermeldung angezeigt.

**Auslösendes Ergebnis:** Starten der App

**Beschreibung:**

1. Die Speiseplandaten für den aktuellen Tag werden vom Server angefragt.
2. Die Benutzeroberfläche zeigt die Speiseplanansicht von der zuletzt genutzten Mensa in der zuletzt genutzten Form (Liste oder Galerie). Dabei finden die Filter, falls aktiv, Anwendung.
3. Falls keine Daten für den Tag vorhanden sind, so wird eine Fehlermeldung angezeigt (vgl. [UI14]).
4. Die App wartet, empfängt die aktuellen Speiseplandaten vom Server und speichert diese.
  - (a) Falls die Verbindung zum Server fehlschlägt, so werden den alten Daten, falls vorhanden, weiterverwendet und eine temporäre Fehlermeldung wird angezeigt.
  - (b) Falls die Verbindung zum Server fehlschlägt und keine alten Daten vorhanden sind, so wird eine Fehlermeldung (vgl. [UI14]) angezeigt.
5. Die Benutzeroberfläche wird aktualisiert.
  - (a) Wird die App zum ersten Mal gestartet, so werden die Daten als Galerie angezeigt.
  - (b) Existieren auch auf dem Server keine Daten für den Tag, so wird angezeigt, dass die Mensa geschlossen hat(vgl. [UI12]).

### [F102] Speiseplanansicht anzeigen

**Kurzbeschreibung:** Die Speiseplanansicht wird angezeigt.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt die Favoritenansicht oder die Einstellungsansicht an.

**Nachbedingung Erfolg:** Die Speiseplanansicht wird angezeigt.

**Nachbedingung Fehlschlag:** Es wird eine Fehlermeldung angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Speiseplanansicht-Schaltfläche

**Beschreibung:**

1. Die Benutzeroberfläche zeigt die Speiseplanansicht von der zuletzt genutzten Mensa in der zuletzt genutzten Form (Liste oder Galerie). Dabei finden die Filter, falls aktiv, Anwendung.
2. Falls keine Daten für den Tag vorhanden sind, so wird eine Fehlermeldung angezeigt (vgl. [UI14]).

**[F103] Speiseplan aktualisieren**

**Kurzbeschreibung:** Der Speiseplan wird aktualisiert.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt die Speiseplanansicht an.

**Nachbedingung Erfolg:** Der Speiseplan wird aktualisiert oder ein entsprechender Hinweis wird angezeigt.

**Nachbedingung Fehlschlag:** Eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Nach unten wischen oder tippen auf die „Erneut versuchen“-Schaltfläche, falls keine aktuellen Daten verfügbar sind.

**Beschreibung:**

1. Eine Ladeanimation wird angezeigt.
2. Die aktuellen Speiseplandaten werden beim Server angefragt.
3. Die App wartet, empfängt die aktuellen Speiseplandaten vom Server und speichert diese.
  - (a) Falls die Verbindung zum Server fehlschlägt, so wird eine temporäre Fehlermeldung angezeigt.
4. Die App aktualisiert die Benutzeroberfläche für den gewählten Tag. Dabei finden die Filter, falls aktiv, Anwendung.
  - (a) Existieren auch auf dem Server keine Daten und der Tag liegt weniger als vier Wochen in der Zukunft, so wird angezeigt, dass die Mensa geschlossen hat.
  - (b) Existieren auf dem Server keine Daten und der Tag liegt mehr als vier Wochen in der Zukunft, so wird angezeigt, dass keine Daten verfügbar sind.

**[F104] Mensa wechseln**

**Kurzbeschreibung:** Die Mensa, deren Speisplan angezeigt wird, wird gewechselt.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt die Speiseplanansicht an.

**Nachbedingung Erfolg:** Die App zeigt den Speiseplan der neu gewählten Mensa an.

**Nachbedingung Fehlschlag:** Die App zeigt eine Fehlermeldung an.

**Auslösendes Ergebnis:** Tippen auf die aktuell ausgewählte Mensa

**Beschreibung:**

1. Dropdown-Menü mit allen Mensen wird angezeigt (vgl. Mensadialog).
2. Der Nutzer tippt auf andere Mensa.
3. Die aktuellen Speiseplandaten der neu ausgewählten Mensa werden beim Server angefragt.
4. Die Benutzeroberfläche zeigt die Speiseplanansicht in der zuletzt genutzten Form (Liste oder Galerie für den aktuellen Tag an), falls dazu lokale Daten existieren. Die angezeigten Gerichte werden entsprechend der aktivierten Filter gefiltert.
5. Die App wartet auf die Daten des Servers und speichert sie.
  - (a) Falls die Verbindung zum Server fehlschlägt, werden die alten Daten, falls vorhanden, weiterverwendet.
  - (b) Falls die Verbindung zum Server fehlschlägt und keine alten Daten vorhanden sind, wird eine Fehlermeldung angezeigt.
6. Die Benutzeroberfläche wird aktualisiert.
  - (a) Existieren auch auf dem Server keine Daten, so wird angezeigt, dass die Mensa geschlossen hat.

**[F105] Tag ändern**

**Kurzbeschreibung:** Der Tag, für den der Speiseplan angezeigt wird, wird geändert.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt die Speiseplanansicht an.

**Nachbedingung Erfolg:** Der Speiseplan wird für den neu ausgewählten Tag in der zuletzt genutzten Form (Galerie oder Liste) dargestellt.

**Nachbedingung Fehlschlag:** Die App zeigt eine Fehlermeldung an.

**Auslösendes Ergebnis:** Tippen auf das Datum

**Beschreibung:**

1. Die Benutzeroberfläche zeigt einen Kalender (vgl. Kalenderdialog) an.
2. Die App wartet bis der Nutzer den Tag ausgewählt hat.
  - (a) Wählt der Nutzer keinen Tag aus, sondern schließt die Ansicht, so wird der Tag nicht geändert.
3. Die App fragt die Speiseplandaten des ausgewählten Tages beim Server an.
4. Die App aktualisiert die Benutzeroberfläche und zeigt die Speiseplandaten des ausgewählten Tages an, falls dazu lokale Daten existieren. Dabei finden die Filter, falls aktiv, Anwendung.
5. Die App wartet auf die Daten vom Server und speichert sie lokal.
  - (a) Falls die Verbindung zum Server fehlschlägt, so werden die alten Daten, falls vorhanden, weiterverwendet.
  - (b) Falls die Verbindung zum Server fehlschlägt und keine alten Daten vorhanden sind, so wird eine Fehlermeldung angezeigt.
6. Die App aktualisiert die Benutzeroberfläche.
  - (a) Ist der ausgewählte Tag mehr als einen Monat im Voraus, so wird angezeigt, dass noch keine Daten verfügbar sind.
  - (b) Existieren auch auf dem Server keine Daten, so wird angezeigt, dass die Mensa geschlossen hat.

**[F106] Ansicht von Galerie zu Liste wechseln**

**Kurzbeschreibung:** Die Darstellungsform wird von Galerie zu Liste gewechselt.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Speiseplanansicht als Galerie an.

**Nachbedingung Erfolg:** Die App zeigt die Speiseplanansicht als Liste an.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf den Listen-Schaltfläche

**Beschreibung:**

1. Die Benutzeroberfläche wird als Liste aktualisiert.

**[F107] Ansicht von Liste zu Galerie wechseln**

**Kurzbeschreibung:** Die Darstellungsform wird von Liste zu Galerie gewechselt.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Speiseplanansicht als Liste an.

**Nachbedingung Erfolg:** Die App zeigt die Speiseplanansicht als Galerie an.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf den Galerie-Schaltfläche

**Beschreibung:**

1. Die Benutzeroberfläche wird als Galerie aktualisiert.

**[F108] Vorherigen Tag anzeigen**

**Kurzbeschreibung:** Der Tag wird auf den Tag vor dem derzeit angezeigten Tag geändert.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** -

**Nachbedingung Erfolg:** Der Speiseplan wird für den vorherigen Tag in der zuletzt genutzten Form (Galerie oder Liste) dargestellt.

**Nachbedingung Fehlschlag:** Es wird eine Fehlermeldung angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Schaltfläche links neben dem Datum

**Beschreibung:**

1. Die App fragt die Speiseplandaten beim Server an.
2. Die App aktualisiert die Benutzeroberfläche und zeigt die Speiseplandaten des vorherigen Tages an, falls dazu lokale Daten existieren. Dabei finden die Filter, falls aktiv, Anwendung.
  - (a) Falls die Verbindung zum Server fehlschlägt, so werden alten Daten, falls vorhanden, weiterverwendet.
  - (b) Falls die Verbindung zum Server fehlschlägt und keine alten Daten vorhanden sind, so wird eine Fehlermeldung angezeigt.
3. Die App wartet auf die Daten vom Server und speichert sie lokal.
  - (a) Falls die Verbindung zum Server fehlschlägt, so werden alten Daten, falls vorhanden, weiterverwendet.
  - (b) Falls die Verbindung zum Server fehlschlägt und keine alten Daten vorhanden sind, so wird eine Fehlermeldung angezeigt.
4. Die App aktualisiert die Benutzeroberfläche.
  - (a) Ist der Tag mehr als einen Monat im Voraus, so wird angezeigt, dass noch keine Daten verfügbar sind.
  - (b) Existieren auch auf dem Server keine Daten, so wird angezeigt, dass die Mensa geschlossen hat.

**[F109] Nächsten Tag anzeigen**

**Kurzbeschreibung:** Der Tag wird auf den Tag nach dem derzeit angezeigten Tag geändert.

**Anforderungen:** [MK01], [MK02], [MK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** -

**Nachbedingung Erfolg:** Der Speiseplan wird für den nächsten Tag in der zuletzt genutzten Form (Galerie oder Liste) dargestellt.

**Nachbedingung Fehlschlag:** Es wird eine Fehlermeldung angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Schaltfläche rechts neben dem Datum

**Beschreibung:**

1. Die App fragt die Speiseplandaten beim Server an.
2. Die App aktualisiert die Benutzeroberfläche und zeigt die Speiseplandaten des nächsten Tages an, falls dazu lokale Daten existieren. Dabei finden die Filter, falls aktiv, Anwendung.
  - (a) Falls die Verbindung zum Server fehlschlägt, so werden alten Daten, falls vorhanden, weiterverwendet.
  - (b) Falls die Verbindung zum Server fehlschlägt und keine alten Daten vorhanden sind, so wird eine Fehlermeldung angezeigt.
3. Die App wartet auf die Daten vom Server und speichert sie lokal.
  - (a) Falls die Verbindung zum Server fehlschlägt, so werden alten Daten, falls vorhanden, weiterverwendet.
  - (b) Falls die Verbindung zum Server fehlschlägt und keine alten Daten vorhanden sind, so wird eine Fehlermeldung angezeigt.
4. Die App aktualisiert die Benutzeroberfläche.
  - (a) Ist der Tag mehr als einen Monat im Voraus, so wird angezeigt, dass noch keine Daten verfügbar sind.
  - (b) Existieren auch auf dem Server keine Daten für den Tag, so wird angezeigt, dass die Mensa geschlossen hat.

## 6.2 Detailansicht

### [F201] Detailansicht für Gerichte anzeigen

**Kurzbeschreibung:** Die Detailansicht und weitere Informationen zum Gericht anzeigen.

**Anforderungen:** [MK01], [MK02]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Speiseplanansicht oder die Favoritenansicht an.

**Nachbedingung Erfolg:** Detailansicht wird angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf Gericht in Speiseplanansicht, Tippen auf Favorit in Favoritenansicht

**Beschreibung:**

1. Die Detailansicht wird mit lokal gespeicherten Daten angezeigt.

### [F202] Detailansicht schließen

**Kurzbeschreibung:** Die Detailansicht wird geschlossen.

**Anforderungen:** [MK01], [MK02]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Detailansicht an.

**Nachbedingung Erfolg:** Detailansicht wird geschlossen.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer tippt auf das Symbol zum Schließen der Detailansicht.

**Beschreibung:**

1. Die Detailansicht hat sich geschlossen und die zeigt die Speiseplanansicht oder die Favoritenansicht an, je nachdem aus welcher Ansicht in die Detailansicht navigiert wurde.

**[F203] Bewertung abgeben**

**Kurzbeschreibung:** Eine Bewertung zu einem spezifischen Gericht wird abgegeben.

**Anforderungen:** [KK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt die Detailansicht eines Gerichts an.

**Nachbedingung Erfolg:** Die Bewertung wird abgegeben und ist in der Detailansicht sichtbar.

**Nachbedingung Fehlschlag:** Die Bewertung ist nicht in der Detailansicht sichtbar und es wird eine temporäre Fehlermeldung angezeigt.

**Auslösendes Ergebnis:** Tippen auf die „Bewerten“-Schaltfläche

**Beschreibung:**

1. Der Bewertungsdialog öffnet sich.
2. Die Bewertung wird vom Nutzer eingegeben
3. Der Nutzer bestätigt die Eingabe.
4. Die Bewertung wird an den Server geschickt.
  - (a) Falls die Verbindung zum Server fehlschlägt, so wird eine temporäre Fehlermeldung angezeigt.
5. Die Benutzeroberfläche und somit auch die Bewertungen werden aktualisiert.

**[F204] Bewertung anpassen**

**Kurzbeschreibung:** Eine Bewertung zu einem spezifischen Gericht wird geändert.

**Anforderungen:** [KK03]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt die Detailansicht eines Gerichts an.

**Nachbedingung Erfolg:** Die Bewertung wird geändert und ist in der Detailansicht sichtbar.

**Nachbedingung Fehlschlag:** Die Bewertung wird nicht geändert.

**Auslösendes Ergebnis:** Tippen auf die „Anpassen“-Schaltfläche

**Beschreibung:**

1. Ein Dialog öffnet sich.
2. Die Bewertung wird nach Bedarf angepasst.
3. Der Nutzer bestätigt die Eingabe.
4. Die Bewertung wird an den Server geschickt.
  - (a) Falls die Verbindung zum Server fehlschlägt, so wird eine temporäre Fehlermeldung angezeigt.
5. Die Benutzeroberfläche und somit auch die Bewertungen werden aktualisiert.

**[F205] Bild hochladen**

**Kurzbeschreibung:** Der Nutzer kann ein Bild in die App einbinden.

**Anforderungen:** [MK04]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt die Detailansicht an.

**Nachbedingung Erfolg:** Eine temporäre Erfolgsmeldung wird angezeigt.

**Nachbedingung Fehlschlag:** Eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Tippen auf den Hochladen-Schaltfläche

**Beschreibung:**

1. Der Hochladedialog wird geöffnet, in dem der Nutzer ein Link zum Bild des jeweiligen Hosters einträgt.
2. Die App wartet auf die Bestätigung des Nutzers.
3. Die App sendet den Link zum Bild an den Server.
  - (a) Bei erfolgreicher Übermittlung und Validierung des Links, wird eine temporäre Erfolgsmeldung angezeigt.
  - (b) Ist die Übermittlung des Links nicht erfolgreich oder die Validierung des Links schlägt fehl, so wird eine entsprechende temporäre Fehlermeldung angezeigt.

**[F206] Allergene und Zusatzstoffe einblenden**

**Kurzbeschreibung:** Die Allergene und Zusatzstoffe werden eingeblendet.

**Anforderungen:** [MK01]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Detailansicht eines Gerichts an und die Allergene und Zusatzstoffe des Gerichts oder einer Beilage sind ausgeblendet.

**Nachbedingung Erfolg:** Die Allergene und Zusatzstoffe des Gerichts oder der Beilage werden angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf den Namen des Gerichts oder der Beilage

**Beschreibung:**

1. Die Allergene und Zusatzstoffe werden angezeigt.
2. Enthält ein Gericht oder der Beilage keine Allergene oder Zusatzstoffe, so wird ein entsprechender Text angezeigt.

**[F207] Allergene und Zusatzstoffe ausblenden**

**Kurzbeschreibung:** Die Allergene und Zusatzstoffe werden ausgeblendet.

**Anforderungen:** [MK01]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Detailansicht eines Gerichts an und die Allergene und Zusatzstoffe des Gerichts oder einer Beilage sind eingeblendet.

**Nachbedingung Erfolg:** Die Allergene und Zusatzstoffe des Gerichts oder der Beilage werden ausgeblendet.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf den Namen des Gerichts oder der Beilage

**Beschreibung:**

1. Die Allergene und Zusatzstoffe des Gerichts oder der Beilage, auf deren Namen getippt wurde, werden ausgeblendet.

### 6.3 Bilder

#### [F301] Galeriedialog öffnen

**Kurzbeschreibung:** Das am höchsten bewertete Bild des Gerichts wird im Galeriedialog angezeigt.

**Anforderungen:** [MK04]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Detailansicht eines Gerichts an.

**Nachbedingung Erfolg:** Das Bild wird in dem Galeriedialog angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf das Bild

**Beschreibung:**

1. Das in der Detailansicht angezeigte Bild wird auch im Galeriedialog angezeigt.
  - (a) Ist kein Bild vorhanden, so wird der Galeriedialog nicht geöffnet.

#### [F302] Galeriedialog verlassen

**Kurzbeschreibung:** Der Galeriedialog wird verlassen.

**Anforderungen:** [MK04]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt den Galeriedialog an.

**Nachbedingung Erfolg:** Die Detailansicht des Gerichts wird angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf die Schaltfläche, die aus dem Dialog hinaus führt

**Beschreibung:**

1. Der Galeriedialog wird geschlossen.

**[F303] Bild melden**

**Kurzbeschreibung:** Ein Bild wird gemeldet.

**Anforderungen:** [MK05]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt den Galeriedialog an und der Nutzer hat das Bild noch nicht gemeldet.

**Nachbedingung Erfolg:** Eine temporäre Erfolgsmeldung wird angezeigt und das Bild wird nicht mehr angezeigt oder nichts verändert sich, sollte der Nutzer den Vorgang abbrechen.

**Nachbedingung Fehlschlag:** Eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Melden-Schaltfläche

**Beschreibung:**

1. Ein Bestätigungsdialog wird angezeigt.
2. Der Nutzer wählt eine der vorgegebenen Kategorien (vgl. Meldedialog) aus.
3. Der Nutzer bestätigt die Meldung.
  - (a) Der Nutzer bricht durch Tippen auf die „Abbruch“-Schaltfläche des Bestätigungsdialogs den Vorgang ab.
4. Die vom Nutzer ausgewählte Kategorie wird an den Server übermittelt.
  - (a) Schlägt die Übertragung der Meldung an den Server fehl, so wird eine temporäre Fehlermeldung angezeigt und der Vorgang abgebrochen.
5. Das gemeldete Bild wird ausgeblendet und eine temporäre Erfolgsmeldung angezeigt.

**[F304] Bilder nach rechts durchwechseln**

**Kurzbeschreibung:** Das nächst höher bewertete Bild wird angezeigt.

**Anforderungen:** [MK04]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt den Galeriedialog an.

**Nachbedingung Erfolg:** Ein anderes Bild wird angezeigt.

**Nachbedingung Fehlschlag:** Dasselbe Bild wird angezeigt.

**Auslösendes Ergebnis:** Von links nach rechts wischen

**Beschreibung:**

1. Anzeigen des nächst höher gerankten Bildes.
  - (a) Falls es kein höher geranktes Bild gibt, so kann nicht weiter gewischt werden.

**[F305] Bilder nach links durchwechseln**

**Kurzbeschreibung:** Das nächst niedriger bewertete Bild wird angezeigt

**Anforderungen:** [MK04]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt den Galeriedialog an.

**Nachbedingung Erfolg:** Ein anderes Bild wird angezeigt.

**Nachbedingung Fehlschlag:** Dasselbe Bild wird angezeigt.

**Auslösendes Ergebnis:** Von rechts nach links wischen

**Beschreibung:**

1. Anzeigen des nächst niedriger gerankten Bildes.
  - (a) Ist kein niedriger geranktes Bild vorhanden, so kann nicht weiter gewischt werden.

**[F306] Aufwertung zu einem Bild hinzufügen**

**Kurzbeschreibung:** Ein Bild wird als hilfreich markiert.

**Anforderungen:** [KK04]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt den Galeriedialog an und das Bild ist nicht als hilfreich markiert.

**Nachbedingung Erfolg:** Die Aufwertung, also die Markierung eines Bildes als hilfreich, wird durch Veränderung der Schaltfläche angezeigt.

**Nachbedingung Fehlschlag:** Die Aufwertung wird nicht angezeigt und eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Aufwerten-Schaltfläche

**Beschreibung:**

1. Die Aufwertung wird dem Server übermittelt.
  - (a) Existiert eine Abwertung von demselben Nutzer für dasselbe Bild, so wird diese gelöscht. (vgl. [F309])
  - (b) Schlägt die Übertragung der Aufwertung fehl, so wird die Aufwertung nicht angezeigt und der Vorgang schlägt fehl.

**[F307] Aufwertung eines Bildes entfernen**

**Kurzbeschreibung:** Eine Aufwertung eines Bildes wird entfernt.

**Anforderungen:** [KK04]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt den Galeriedialog an und das Bild ist als hilfreich markiert.

**Nachbedingung Erfolg:** Es wird keine Aufwertung angezeigt.

**Nachbedingung Fehlschlag:** Die Aufwertung und eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Aufwerten-Schaltfläche

**Beschreibung:**

1. Die Rücknahme der Aufwertung wird dem Server übermittelt.
  - (a) Schlägt die Übertragung der Rücknahme der Aufwertung fehl, so wird die Aufwertung weiterhin angezeigt und der Vorgang schlägt fehl.

**[F308] Abwertung zu einem Bild hinzufügen**

**Kurzbeschreibung:** Ein Bild wird als nicht-hilfreich markiert.

**Anforderungen:** [KK04]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt den Galeriedialog an und das Bild ist nicht als nicht-hilfreich markiert.

**Nachbedingung Erfolg:** Die Abwertung wird angezeigt.

**Nachbedingung Fehlschlag:** Die Abwertung wird nicht angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Abwerten-Schaltfläche

**Beschreibung:**

1. Die Abwertung wird dem Server übermittelt.
  - (a) Existiert eine Aufwertung von demselben Nutzer für dasselbe Bild, so wird diese gelöscht. (vgl. [F307])
  - (b) Schlägt die Übertragung der Abwertung fehl, so wird die Abwertung nicht angezeigt und der Vorgang schlägt fehl.

**[F309] Abwertung eines Bildes entfernen**

**Kurzbeschreibung:** Eine Abwertung eines Bildes wird entfernt.

**Anforderungen:** [KK04]

**Akteure:** Nutzer, Client, Server

**Vorbedingung:** Die App zeigt den Galeriedialog an und das Bild ist als nicht-hilfreich markiert.

**Nachbedingung Erfolg:** Es wird keine Abwertung angezeigt.

**Nachbedingung Fehlschlag:** Die Abwertung und eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Abwerten-Schaltfläche

**Beschreibung:**

1. Die Rücknahme der Abwertung wird dem Server übermittelt.
  - (a) Schlägt die Übertragung der Rücknahme der Abwertung fehl, so wird die Abwertung weiterhin angezeigt und der Vorgang schlägt fehl.

## 6.4 Filterfunktion

### [F401] Filterdialog öffnen

**Kurzbeschreibung:** Der Filterdialog wird geöffnet.

**Anforderungen:** [MK03], [KK08], [KK09]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Speiseplanansicht an.

**Nachbedingung Erfolg:** Der Filterdialog wird angezeigt und zeigt alle möglichen Filteroptionen (vgl. Filterdialog) an.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer tippt auf das Filtersymbol zum Öffnen des Filterdialogs.

**Beschreibung:**

1. Die lokale Filterkonfiguration wird geladen.
2. Der Filterdialog wird geöffnet.

### [F402] Filterdialog schließen

**Kurzbeschreibung:** Der Filterdialog wird geschlossen ohne geänderte Filteroptionen zu speichern

**Anforderungen:** [MK03], [KK08], [KK09]

**Akteure:** Nutzer, Client

**Vorbedingung:** Der Filterdialog ist geöffnet

**Nachbedingung Erfolg:** Die Speiseplanansicht wird angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer tippt auf das Symbol zum Schließen des Filterdialogs.

**Beschreibung:**

1. Der Filterdialog wird geschlossen und die Speiseplanansicht wird angezeigt.

**[F403] Filteroptionen auswählen**

**Kurzbeschreibung:** Verschiedene Optionen zum Filtern können ausgewählt werden.

**Anforderungen:** [MK03], [KK08], [KK09]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt den Filterdialog an.

**Nachbedingung Erfolg:** Die ausgewählten Filteroptionen werden im Filterdialog angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Der Nutzer ändert Filteroptionen

**Beschreibung:**

1. Die Konfiguration ist beim ersten Öffnen mit Standardwerten gefüllt.
2. Der Nutzer sieht Sektionen, in welchen er folgende Filtermöglichkeiten hat:
  - Gerichtstyp
  - An- und Abwählen einzelner Allergene
  - Einstellen eines Preisintervalls
  - Einstellen eines Bewertungsintervalls
  - Wählen einer der Häufigkeitsklassen „selten“ oder „neu“
  - Filtern nach Favoriten
3. Der Nutzer kann einer der folgenden Sortierungen je auf- oder absteigend wählen:
  - nach Linie
  - nach Preis
  - nach Bewertung
  - nach Häufigkeit

**[F404] Filterkonfiguration speichern**

**Kurzbeschreibung:** Die Änderungen an der Filterkonfiguration werden gespeichert.

**Anforderungen:** [MK03], [KK08], [KK09]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt den Filterdialog an.

**Nachbedingung Erfolg:** Die Speiseplanansicht wird angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer tippt auf die „Speichern“-Schaltfläche

**Beschreibung:**

1. Die Konfiguration des Filters wird gespeichert.
2. Der Filterdialog wird geschlossen und die Speiseplanansicht wird nach den aktualisierten Filtern gefiltert angezeigt.

**[F405] Filter zurücksetzen**

**Kurzbeschreibung:** Zurücksetzung der Konfiguration des Filters

**Anforderungen:** [MK03], [KK08], [KK09]

**Akteure:** Nutzer, Client

**Vorbedingung:** Der Filterdialog wird angezeigt

**Nachbedingung Erfolg:** Alle Filteroptionen sind auf ihren Standardwert zurückgesetzt. Jedes Gericht wird bei dieser Filterkonfiguration angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer tippt auf die „Zurücksetzen“-Schaltfläche im Filterdialog

**Beschreibung:**

1. Die Filteroptionen werden in der Benutzeroberfläche auf die Standardwerte aktualisiert, sodass kein Gericht heraus gefiltert wird.

**[F406] Filter aktivieren**

**Kurzbeschreibung:** Der Filter wird per Schaltfläche aktiviert.

**Anforderungen:** [MK03], [KK08], [KK09]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die Speiseplanansicht wird angezeigt und der Filter ist deaktiviert.

**Nachbedingung Erfolg:** Der Filter wird auf die Speiseplanansicht angewendet.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer hält das Filtersymbol gedrückt

**Beschreibung:**

1. Das Filtersymbol verändert sich.
2. Die Speiseplanansicht wird aktualisiert und nur die gefilterten Gerichte angezeigt.
  - (a) Existieren keine Gerichte für die gewählten Filtereinstellungen, so wird eine entsprechende Nachricht in der Ansicht angezeigt ([UI13]).

**[F407] Filter deaktivieren**

**Kurzbeschreibung:** Der Filter wird per Schaltfläche deaktiviert.

**Anforderungen:** [MK03], [KK08], [KK09]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die Speiseplanansicht wird angezeigt und der Filter ist aktiviert.

**Nachbedingung Erfolg:** Der Filter wird deaktiviert und die Speiseplanansicht zeigt alle Gerichte der ausgewählten Mensa für den jeweiligen Tag.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer hält das Filtersymbol gedrückt

**Beschreibung:**

1. Das Filtersymbol verändert sich.
2. Die Speiseplanansicht wird aktualisiert, ohne, dass die Filter angewendet werden.

## 6.5 Favoriten

### [F501] Favoritenansicht anzeigen

**Kurzbeschreibung:** Die Favoritenansicht wird angezeigt.

**Anforderungen:** [KK02]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Speiseplanansicht oder die Einstellungsansicht an.

**Nachbedingung Erfolg:** Die favorisierten Gerichte werden in der Favoritenansicht angezeigt.

**Nachbedingung Fehlschlag:** Eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Tippen auf die Favoriten-Schaltfläche in der Navigationsleiste

**Beschreibung:**

1. Die App fragt die Daten der vorhanden Favoriten beim Server an.
2. Die App empfängt die Daten vom Server.
  - (a) Schlägt die Verbindung zum Server fehl, so wird eine temporäre Fehlermeldung angezeigt.
3. Die App zeigt die Favoritenansicht an, die eine Liste der favorisierten Gerichte enthält, falls der Nutzer Gerichte favorisiert hat.

### [F502] Gericht favorisieren

**Kurzbeschreibung:** Ein Gericht zu den Favoriten hinzufügen.

**Anforderungen:** [KK02]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Detailansicht eines Gerichtes an und das Gericht ist kein Favorit.

**Nachbedingung Erfolg:** Das Gericht ist zu den Favoriten hinzugefügt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf die Favoriten-Schaltfläche eines nicht-favorisierten Gerichtes

**Beschreibung:**

1. Das Gericht wird zu den Favoriten hinzugefügt.

### [F503] Gericht defavorisieren

**Kurzbeschreibung:** Ein Gericht von den Favoriten entfernen.

**Anforderungen:** [KK02]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Detailansicht eines Gerichtes an und das Gericht ist favorisiert.

**Nachbedingung Erfolg:** Das Gericht wird von den Favoriten entfernt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf die Favoriten-Schaltfläche eines favorisierten Gerichtes

**Beschreibung:**

1. Das Gericht wird aus den Favoriten entfernt.

## 6.6 Einstellungen

### [F601] Einstellungsansicht anzeigen

**Kurzbeschreibung:** Die Einstellungsansicht mit verschiedenen Einstellungsmöglichkeiten wird geöffnet.

**Anforderungen:** [MK02], [KK01], [KK06], [KK07]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Speiseplanansicht oder die Favoritenansicht an.

**Nachbedingung Erfolg:** Die App zeigt die Einstellungsansicht an.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf das Einstellungssymbol in der Navigationsleiste

**Beschreibung:**

1. Die Einstellungsansicht wird angezeigt.

### [F602] Farbschema einstellen

**Kurzbeschreibung:** Das Farbschema wird verändert.

**Anforderungen:** [KK01]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Einstellungsansicht an.

**Nachbedingung Erfolg:** Die App erscheint im gewählten Farbschema.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf Farbschema

**Beschreibung:**

1. Das ausgewählte Farbschema wird gespeichert.
2. Die App wird im neuen Farbschema angezeigt.

### [F603] Preisklasse einstellen

**Kurzbeschreibung:** Die Preisklasse wird verändert.

**Anforderungen:** [MK02]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Einstellungsansicht an.

**Nachbedingung Erfolg:** Bei Gerichten werden die Preise der neu ausgewählten Preisklasse angezeigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Nutzer wählt eine Preisklasse

**Beschreibung:**

1. Die Preisklasse wird gespeichert.

**[F604] Einstellungen und Favoriten exportieren**

**Kurzbeschreibung:** Die Einstellungen und Favoriten werden als Datei exportiert.

**Anforderungen:** [KK07]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Einstellungsansicht an.

**Nachbedingung Erfolg:** Es wurde eine Datei mit den Einstellungen und Favoriten des Nutzers erstellt.

**Nachbedingung Fehlschlag:** Es wurde keine Datei erstellt und eine temporäre Fehlermeldung wird angezeigt.

**Auslösendes Ergebnis:** Der Nutzer betätigt Schaltfläche „Exportieren“

**Beschreibung:**

1. Ein Systemdialog zum Wählen eines Speicherortes für die Export-Datei wird geöffnet.
2. Die Export-Datei mit allen Export-Daten wird erzeugt und am gewünschten Speicherort abgelegt. Falls keine Datei erstellt werden kann schlägt der Export fehl und es wird eine temporäre Fehlermeldung angezeigt.

**[F605] Einstellungen und Favoriten importieren**

**Kurzbeschreibung:** Zuvor exportierte Einstellungen und Favoriten werden importiert.

**Anforderungen:** [KK07]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Einstellungsansicht an und eine Export-Datei liegt vor.

**Nachbedingung Erfolg:** Die Einstellungen und Favoriten wurden durch jene aus der Export-Datei überschrieben.

**Nachbedingung Fehlschlag:** Die Einstellungen und Favoriten wurden nicht geändert.

**Auslösendes Ergebnis:** Der Nutzer betätigt Schaltfläche „Daten importieren“

**Beschreibung:**

1. Ein Systemdialog zum Wählen einer Datei wird geöffnet.
2. Die gewählte Datei wird gelesen und falls sie sich in einem korrektem Format befindet, werden die Einstellungen und Favoriten durch jene aus der Datei überschrieben. Andernfalls schlägt der Import fehl, die bestehenden Einstellungen und Favoriten werden nicht verändert und es wird eine temporäre Fehlermeldung angezeigt.

## 6.7 Benachrichtigungen

### [F701] Benachrichtigungen aktivieren

**Kurzbeschreibung:** Die Benachrichtigungen für favorisierte Gerichte werden aktiviert.

**Anforderungen:** [KK06]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App zeigt die Einstellungsansicht an und die Benachrichtigungen sind deaktiviert

**Nachbedingung Erfolg:** Der Nutzer wird zu gegebener Uhrzeit über favorisierte Gerichte benachrichtigt.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf den Schalter zum aktivieren der Benachrichtigungen

**Beschreibung:**

1. Der Nutzer erhält zu festgelegter Uhrzeit eine Benachrichtigung pro favorisiertem Gericht, das aktuell bei seiner (zum Zeitpunkt der Benachrichtigung) zuletzt ausgewählten Mensa angeboten wird. Diese Benachrichtigung beinhaltet:

- Gerichtsbezeichnung
- Linie
- Bild

### [F702] Benachrichtigungen deaktivieren

**Kurzbeschreibung:** Nutzer deaktiviert Benachrichtigungen.

**Anforderungen:** [KK06]

**Akteure:** Nutzer, Client

**Vorbedingung:** Benachrichtigungen sind aktiviert und die App zeigt die Einstellungsansicht an.

**Nachbedingung Erfolg:** Nutzer erhält keine Benachrichtigungen mehr.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf den Schalter zum deaktivieren der Benachrichtigungen

**Beschreibung:**

1. Der Nutzer erhält keine Benachrichtigungen mehr.

**[F703] Benachrichtigungszeitpunkt wählen**

**Kurzbeschreibung:** Der Nutzer wählt den Benachrichtigungszeitpunkt.

**Anforderungen:** [KK06]

**Akteure:** Nutzer, Client

**Vorbedingung:** Die App befindet sich in der Einstellungsansicht und Benachrichtigungen sind aktiviert.

**Nachbedingung Erfolg:** Der Benachrichtigungszeitpunkt wird aktualisiert.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Tippen auf die Schaltfläche für den Benachrichtigungszeitpunkt

**Beschreibung:**

1. Der Nutzer wählt eine Uhrzeit.
2. Die Uhrzeit wird als Benachrichtigungszeitpunkt gespeichert.
3. Der Nutzer erhält zukünftig Benachrichtigungen zur gewählten Uhrzeit.

## 6.8 Server

### [F801] Server starten

**Kurzbeschreibung:** Server wird gestartet und initialisiert sich.

**Anforderungen:** [MK01], [MK02], [MK03], [MK04], [MK05], [KK02], [KK03], [KK04], [KK05], [KK06]

**Akteure:** Administrator, Server

**Vorbedingung:** -

**Nachbedingung Erfolg:** Server ist gestartet und erfolgreicher Start wurde protokolliert.

**Nachbedingung Fehlschlag:** Server ist nicht gestartet und Fehlermeldungen wurden protokolliert.

**Auslösendes Ergebnis:** Administrator startet Server

**Beschreibung:**

1. Server liest Einstellungen aus Umgebungsvariablen. Diese beinhalten:

- Verbindungsinformationen zur Datenbank
- Protokollierungsstufe
- Verbindungsinformationen zu einem Mailserver für Administratorbenachrichtigungen
- Administrator E-Mail-Adresse
- URL zur Mensa-Webseite
- Uhrzeiten für Aktualisierungsabfragen und Komplettabfragen der Mensa-Webseite
- Meldeschranke

2. Server verbindet sich mit Datenbank.

3. Server macht sich für API-Anfragen bereit.

4. Server protokolliert einen erfolgreichen Start

Falls einer der Schritte 1, 2 oder 3 nicht erfolgreich war, protokolliert der Server dies und bricht den Startvorgang ab.

**[F802] Heutigen Speiseplan aktualisieren**

**Kurzbeschreibung:** Server fragt Mensa-Webseite ab und aktualisiert heutige Speisepläne.

**Anforderungen:** [MK01], [MK02], [MK03], [KK02], [KK05], [KK06]

**Akteure:** Server

**Vorbedingung:** Der Server ist gestartet.

**Nachbedingung Erfolg:** Die Datenbank wurde mithilfe der neuen Speisepläne aktualisiert.

**Nachbedingung Fehlschlag:** Die Datenbank wurde nicht aktualisiert und eine Fehlermeldung wurde protokolliert.

**Auslösendes Ergebnis:** Die zuvor festgelegte Uhrzeit für eine Aktualisierungsabfrage wurde an einem der Wochentage Montag bis Freitag erreicht.

**Beschreibung:**

1. Der Server protokolliert den Start der Speiseplanaktualisierung.
2. Der Server lädt die Mensa-Webseite.
3. Der Server parst die Mensa-Webseite und extrahiert die Speiseplandaten für den aktuellen Tag.
4. Der Server verarbeitet die Speiseplandaten vor und konvertiert diese in ein für die Datenbank geeignetes Format.
5. Der Server aktualisiert die Datenbank mit den neuen Speiseplandaten.
6. Der Server protokolliert den erfolgreichen Abschluss der Speiseplanaktualisierung und die hierfür benötigte Zeit.

Falls einer der Schritte 2, 3, 4 oder 5 nicht erfolgreich war, protokolliert der Server dies und bricht den Aktualisierungsvorgang ab.

**[F803] Speisepläne für die nächsten vier Wochen aktualisieren**

**Kurzbeschreibung:** Server fragt Mensa-Webseite ab und aktualisiert Speisepläne der nächsten vier Wochen.

**Anforderungen:** [MK01], [MK02], [MK03], [KK02], [KK05], [KK06]

**Akteure:** Server

**Vorbedingung:** Der Server ist gestartet.

**Nachbedingung Erfolg:** Die Datenbank wurde mit den neuen Speiseplänen aktualisiert.

**Nachbedingung Fehlschlag:** Die Datenbank wird nicht aktualisiert und eine Fehlermeldung wird protokolliert.

**Auslösendes Ergebnis:** Die zuvor festgelegte Uhrzeit für eine Komplettabfrage wurde an einem beliebigen Wochentag erreicht.

**Beschreibung:**

Vorgehen äquivalent zu [F802], jedoch wird der Vorgang für die Speiseplandaten der nächsten vier Wochen beginnend mit der aktuellen Woche durchgeführt.

**[F804] API-Anfragen beantworten**

**Kurzbeschreibung:** Der Server beantwortet API-Anfragen mit Daten aus der Datenbank.

**Anforderungen:** [MK01], [MK02], [MK03], [KK02], [KK05], [KK06]

**Akteure:** Client, Server

**Vorbedingung:** Der Server ist gestartet.

**Nachbedingung Erfolg:** Die Anfrage wurde mit den gewünschten Daten beantwortet und protokolliert.

**Nachbedingung Fehlschlag:** Die Anfrage wurde mit einer Fehlermeldung beantwortet.

**Auslösendes Ergebnis:** Server erhält eine API-Anfrage

**Beschreibung:**

1. Der Server erhält eine API-Anfrage und protokolliert diese.
2. Der Server fragt die gewünschte Daten in der Datenbank ab oder berechnet diese aus anderen Daten in der Datenbank.
3. Der Server beantwortet Anfrage mit den gewünschten Daten.

Falls Schritt 2 nicht erfolgreich war protokolliert der Server dies und beantwortet die Anfrage mit einer Fehlermeldung.

**[F805] API-Anfragen bezüglich Bildern beantworten**

**Kurzbeschreibung:** Der Server beantwortet API-Anfragen bezüglich Bildern mit Referenzen zum Bildhoster.

**Anforderungen:** [MK04]

**Akteure:** Client, Server

**Vorbedingung:** Der Server ist gestartet.

**Nachbedingung Erfolg:** Die Anfrage wurde mit den gewünschten Daten beantwortet und protokolliert.

**Nachbedingung Fehlschlag:** Die Anfrage wird ohne Bilder beantwortet.

**Auslösendes Ergebnis:** Server erhält eine API-Anfrage

**Beschreibung:**

1. Der Server erhält eine API-Anfrage bezüglich Bildern und protokolliert diese.
2. Der Server berechnet den Bildrang für alle Bilder des gewünschten Gerichts aus der Anzahl und Art der Bewertungen, dem Einstelldatum des Bildes und der Anzahl an Meldeanträgen.
3. Der Server beachtet die Bilder in absteigender Reihenfolge des Bildrangs.
4. Der Server fragt beim Bildhoster die Bilder an, um zu überprüfen, ob diese dort immer noch vorhanden sind.
  - (a) Ist dies nicht der Fall, so wird dies protokolliert. Außerdem wird die Referenz zu diesem Bild aus der Datenbank entfernt und dieses im weiteren Vorgehen nicht beachtet.
5. Der Server beantwortet die Anfrage mit Referenzen zu den Bildern.

**[F806] Nutzerdaten aktualisieren**

**Kurzbeschreibung:** Der Server aktualisiert Nutzerdaten wie Bewertungen oder verlinkte Bilder.

**Anforderungen:** [MK04], [KK03], [KK04]

**Akteure:** Client, Server

**Vorbedingung:** Der Server ist gestartet.

**Nachbedingung Erfolg:** Die Nutzerdaten wurden in der Datenbank aktualisiert und der Vorgang wurde protokolliert.

**Nachbedingung Fehlschlag:** Die Nutzerdaten wurden nicht aktualisiert und Fehler wurden protokolliert.

**Auslösendes Ergebnis:** Server erhält eine API-Anfrage auf Nutzerdatenänderung

**Beschreibung:**

1. Der Server erhält eine API-Anfrage für Änderung der Nutzerdaten und protokolliert diese.
2. Der Server validiert den API-Schlüssel der Anfrage, siehe 4.4 API.
3. Falls es sich bei den Nutzerdaten um ein verlinktes Bild handelt, so wird überprüft ob ein valider Link angegeben wurde. Ist dies nicht der Fall, so wird die API-Anfrage mit einer Fehlermeldung beantwortet.
4. Der Server aktualisiert Nutzerdaten in der Datenbank.
5. Der Server bestätigt dem Client die erfolgreiche Aktualisierungsabfrage.

Falls einer der Schritte 2 oder 3 nicht erfolgreich war protokolliert der Server dies und beantwortet die Anfrage mit einer Fehlermeldung.

**[F807] Meldeantrag behandeln**

**Kurzbeschreibung:** Der Server speichert einen Meldeantrag und leitet diesen an einen Administrator weiter.

**Anforderungen:** [MK05]

**Akteure:** Server

**Vorbedingung:** Der Server ist gestartet.

**Nachbedingung Erfolg:** Der Administrator wurde kontaktiert.

**Nachbedingung Fehlschlag:** Der Administrator wurde nicht kontaktiert.

**Auslösendes Ergebnis:** Server erhält API-Anfrage für Meldeantrag

**Beschreibung:**

1. Der Server erhält eine API-Anfrage für einen Meldeantrag und protokolliert diese.
2. Der Server validiert den API-Schlüssel der Anfrage, siehe 4.4 API.
3. Der Server trägt einen Meldeantrag in die Datenbank ein.
4. Der Server berechnet das Meldemaß aus der Anzahl und Art der Bewertungen und der Anzahl der Meldeanträge.
5. Der Server überprüft, ob das Meldemaß für das Bild die Meldeschranke überschreitet. Ist dies der Fall, so wird das Bild in Zukunft nicht mehr an Clients versendet und der Vorgang protokolliert.
6. Der Server kontaktiert den Administrator per E-Mail und berichtet über den Meldeantrag und eine gegebenenfalls vorgenommene Reaktion. Der Administrator kann ein Bild manuell aus der Datenbank entfernen.
7. Der Server bestätigt dem Client den bearbeiteten Meldeantrag und teilt mit, ob das Bild entfernt wurde.

Falls einer der Schritte 2, 3 oder 4 nicht erfolgreich war, protokolliert der Server dies und beantwortet die Anfrage mit einer Fehlermeldung.

Falls Schritt 5 nicht erfolgreich war protokolliert der Server dies.

**[F808] Ereignis protokollieren**

**Kurzbeschreibung:** Der Server protokolliert ein Ereignis wie eine erhaltene Anfragen oder einen Fehler.

**Anforderungen:** [NF06]

**Akteure:** Server

**Vorbedingung:** -

**Nachbedingung Erfolg:** Ein Ereignis oder ein Fehler wurde protokolliert.

**Nachbedingung Fehlschlag:** -

**Auslösendes Ergebnis:** Ereignis oder Fehler tritt ein, wie in anderen Funktionen beschrieben

**Beschreibung:**

1. Der Server protokolliert ein Ereignis oder einen Fehler, falls diese von der aktuell eingestellten Protokollierungsstufe zugelassen werden.

# 7 Nichtfunktionale Anforderungen

Neben den zuvor spezifizierten funktionalen Anforderungen muss das Produkt auch einige nichtfunktionale Anforderungen erfüllen, die nicht eine einzelne Funktion des Produkts betreffen, sondern das Produkt als Ganzes und dessen Benutzbarkeit beschreiben.

- [NF01] **Leistung:** Bis zu 1000 Nutzer müssen im ordnungsgemäßen Betrieb (vgl. Produktumgebung) gleichzeitig durch die App navigieren können und infolgedessen Anfragen an den Server senden, welche beantwortet werden.
- [NF02] **Sicherheit:** API-Anfragen an den Server, welche Änderungen in der Datenbank auslösen, müssen durch einen API-Schlüssel abgesichert sein (siehe 4.4 API).
- [NF03] **Benutzbarkeit:** Die Nutzerführung der Anwendung muss intuitiv gestaltet sein. Dadurch soll eine Navigation durch die App ohne externe Anweisungen möglich sein.
- [NF04] **Praktikabilität:** Alle im Pflichtenheft spezifizierten Ansichten müssen innerhalb von maximal fünf Interaktionen nach Starten der App erreichbar sein.
- [NF05] **Zuverlässigkeit:** Die App kommuniziert eigene Fehler durch Fehlermeldungen an den Nutzer und wird dabei aufgrund von unbehandelten Fehlern nicht vom Betriebssystem beendet, soweit dies nicht auf Probleme des Geräts, wie z.B. die Überlastung des Arbeitsspeichers zurückzuführen ist.
- [NF06] **Administrierbarkeit:** Der Server muss Ereignisse und Fehler in unterschiedlichen Protokollierungsstufen protokollieren.
- [NF07] **Übertragbarkeit:** Die App muss auf Android-Geräten mit einer SDK Version von mindestens 19 lauffähig sein.
- [NF08] **Transparenz:** Der Quellcode des Produkts muss unter der MIT-Lizenz<sup>9</sup> veröffentlicht werden.
- [NF09] **Datenschutz:** Das Produkt muss sich an die Datenschutzgrundverordnung (DSGVO) halten. Dies ist insbesondere für die Bewertungen und Verlinkung von Bildern relevant, welche serverseitig gespeichert werden. Außerdem muss dem Nutzer eine Anlaufstelle mitgeteilt werden, um alle seine gespeicherten personenbezogenen Daten abfragen oder löschen zu können.
- [NF10] **Barrierefreiheit:** Die App muss die vom Flutter-Framework empfohlenen Richtlinien zur Barrierefreiheit<sup>10</sup> einhalten.
- [NF11] **Sprache:** Die Nutzeroberfläche muss auf Deutsch gestaltet sein.
- [NF12] **Übersetzbarkeit:** Die Nutzeroberfläche der App muss ohne Änderungen im Programmcode in andere natürliche Sprachen übersetzt werden können.
- [NF13] **Änderbarkeit:** Das Produkt muss modular aufgebaut sein, um zukünftige Erweiterungen zu ermöglichen. Dafür sind die SOLID-Designprinzipien einzuhalten.
- [NF14] **Portierbarkeit:** Die App muss einfach auf weitere Zielplattformen wie iOS oder das Web übertragbar sein. Konkret muss der Aufwand hierfür geringer als der einer Neuentwicklung für diese Plattformen sein.

---

<sup>9</sup><https://mit-license.org/>

<sup>10</sup><https://docs.flutter.dev/accessibility-and-localization/accessibility>

## 8 Produktdaten

Um alle gewünschten Funktionen umzusetzen, müssen einige Daten persistent gespeichert werden.

### 8.1 App

In der App werden folgende Daten gespeichert:

[DC01] Client-Identifikator zur Identifikation des Clients

[DC02] Gespeicherte Liste der favorisierten Gerichte

[DC03] Lokale Einstellungen, einschließlich Filtereinstellungen und zuletzt gewählter Mensa, zuletzt gewählte Form der Speiseplanansicht, Preisklasse, Benachrichtigungseinstellungen und dem Farbschema

[DC04] Zwischengespeicherte Speisepläne aller Menschen für die nächste Woche. Dabei werden die Speisepläne für vergangene Tage aus dem Speicher entfernt. Pro Gericht wird gespeichert:

- Gerichtsidentifikator
- Gerichtsbezeichnung
- Gerichtstyp
- Bewertung
- Linie
- Preis
- Häufigkeit
- Letztes Angebotsdatum
- Beilagen
- Allergene und Zusatzstoffe
- Häufigkeitsklasse

### 8.2 Server und Datenbank

In der Datenbank wird serverseitig gespeichert:

[DS01] Speisepläne aller Menschen für die nächsten vier Wochen. Speisepläne für vergangene Tage werden dabei nicht aus dem Speicher entfernt und sind entsprechend weiterhin abrufbar. Pro Gericht wird gespeichert:

- Gerichtsidentifikator
- Gerichtsbezeichnung
- Gerichtstyp
- Linie
- Preis zu verschiedenen Preisklassen
- Allergene und Zusatzstoffe
- Einzelne Nutzerbewertungen
- Referenzen zu von Nutzern verlinkten Bildern

[DS02] Meldeanträge zu Bildern

[DS03] API-Schlüssel

#### 8.3 Nutzerdaten

Folgende vom Nutzer erzeugten Daten werden serverseitig gespeichert:

[DU01] Bewertung eines Gerichts

[DU02] Link zu Bild

[DU03] Bewertung eines Bildes

[DU04] Meldeantrag für ein Bild

#### 8.4 Export-Daten

Folgende Daten können im Client exportiert werden:

[DE01] Farbschema

[DE02] Preisklasse

[DE03] Filtereinstellungen

[DE04] Liste aller favorisierten Gerichte

[DE05] Client-Identifikator

# 9 Globale Testfälle

## 9.1 Funktionstests

Zur Gewährleistung der Musskriterien werden Testfälle definiert, die deren Funktionalität überprüfen. Ein Funktionstest ist erfolgreich, wenn das erwartete Ergebnis eintrifft. Bei allen Funktionstests gehen wir, falls nicht explizit erwähnt, von einer funktionierenden Serververbindung aus. Falls nicht anders angegeben, ist die Mensa und der Tag so gewählt, dass die Mensa an dem Tag geöffnet ist und Gerichte anbietet. Folgende fundamentale Funktionalitäten sind zu testen:

### [TF01X] App starten

**Betroffene Funktionen:** [F101]

**Ausführung:** Die App wird gestartet

**Varianten:**

[TF011] **Vorbedingung:** Die App wird gestartet

**Erwartung:** Die App zeigt die Speiseplanansicht und der Speiseplan wird aktualisiert.

### [TF02X] Server starten

**Betroffene Funktionen:** [F801]

**Ausführung:** Der Administrator startet den Server

**Varianten:**

[TF021] **Vorbedingung:** Umgebungsvariablen wurden nicht definiert

[TF022] **Vorbedingung:** Die Datenbank-Initialisierung schlägt fehl

[TF023] **Vorbedingung:** Der Webserver konnte nicht gestartet werden

**Erwartung:** Der Server konnte nicht gestartet werden und die jeweiligen Fehler werden protokolliert

[TF024] **Vorbedingung:** Der Server startet ohne Probleme (Keiner der obigen Fälle trifft zu)

**Erwartung:** Der Server protokolliert einen erfolgreichen Start

### [TF03X] Speiseplanansicht anzeigen

**Betroffene Funktionen:** [F102]

**Ausführung:** Der Nutzer befindet sich in der Favoriten-, oder Einstellungsansicht

**Varianten:**

[TF031] **Vorbedingung:** Die App zeigt die Einstellungsansicht an

[TF032] **Vorbedingung:** Die App zeigt die Favoritenansicht an

**Erwartung:** Die Speiseplanansicht wird angezeigt

**[TF04X] Speiseplan aktualisieren****Betroffene Funktionen:** [F103], [F802], [F804]**Ausführung:** Der Nutzer wischt den Speiseplan nach unten**Varianten:****[TF041] Vorbedingung:** Die App hat eine Verbindung zum Server**Erwartung:** Die App zeigt die Speiseplanansicht an. Es werden alle Gerichte des ausgewählten Tages angezeigt**[TF042] Vorbedingung:** Die App hat keine Serververbindung und die App hat lokale Daten für den Tag gespeichert**Erwartung:** Die App zeigt die Speiseplanansicht an. Es werden alle Gerichte des ausgewählten Tages angezeigt**[TF043] Vorbedingung:** Die App hat keine Serververbindung und die App hat keine lokalen Daten für den Tag gespeichert**Erwartung:** Es wird eine Fehlermeldung angezeigt**[TF05X] Detailansicht anzeigen****Betroffene Funktionen:** [F201]**Ausführung:** Der Nutzer tippt auf ein Gericht**Varianten:****[TF051] Vorbedingung:** Die App zeigt die Speiseplanansicht an**[TF052] Vorbedingung:** Die App zeigt die Favoritenansicht an**Erwartung:** Die Detailansicht wird angezeigt**[TF06X] Einstellungen anzeigen****Betroffene Funktionen:** [F601]**Ausführung:** Der Nutzer tippt auf das Einstellungssymbol in der Navigationsleiste**Varianten:****[TF061] Vorbedingung:** Die App zeigt die Speiseplanansicht an**[TF062] Vorbedingung:** Die App zeigt die Favoritenansicht an**Erwartung:** Die Einstellungsansicht wird angezeigt**[TF07X] Preisklasse einstellen****Betroffene Funktionen:** [F603]**Ausführung:** Der Nutzer wählt eine Preisklasse**Varianten:****[TF071] Vorbedingung:** Die App zeigt die Einstellungsansicht an**Erwartung:** Die Preise der Gerichte entsprechen der ausgewählten Preisklasse

**[TF08X] Mensa wechseln****Betroffene Funktionen:** [F104]**Ausführung:**

- Der Nutzer tippt auf die aktuell ausgewählte Mensa
- Der Nutzer wählt eine andere Mensa aus

**Varianten:****[TF081] Vorbedingung:** Die App zeigt die Speiseplanansicht an**Erwartung:** Der Speiseplan der anderen Mensa wird angezeigt**[TF09X] Tag ändern****Betroffene Funktionen:** [F105], [F803]**Ausführung:**

- Der Nutzer tippt auf das Datum
- Der Nutzer wählt einen Tag aus

**Varianten:****[TF091] Vorbedingung:** Die App zeigt die Speiseplanansicht an**Erwartung:** Der Speiseplan für den neu ausgewählten Tag wird angezeigt**[TF10X] Filterdialog öffnen****Betroffene Funktionen:** [F401], [F402]**Ausführung:** Der Nutzer tippt auf das Filtersymbol**Varianten:****[TF101] Vorbedingung:** Die App zeigt die Speiseplanansicht an und es wurde noch kein Filter konfiguriert**Erwartung:** Der Filterdialog wird angezeigt und alle Standardwerte sind ausgewählt**[TF102] Vorbedingung:** Die App zeigt die Speiseplanansicht an und es wurde ein Filter konfiguriert**Erwartung:** Der Filterdialog wird angezeigt mit den zuletzt eingestellten Filtermöglichkeiten**[TF11X] Filterdialog schließen****Betroffene Funktionen:** [F401], [F402]**Ausführung:** Der Nutzer tippt auf die „Schließen“-Schaltfläche**Varianten:****[TF111] Vorbedingung:** Der Filterdialog ist geöffnet**Erwartung:** Der Filterdialog wird geschlossen und die App zeigt die Speiseplanansicht

**[TF12X] Filteroptionen auswählen****Betroffene Funktionen:** [F403]**Ausführung:** Der Nutzer wählt Filteroptionen aus**Varianten:**

- [TF121] **Vorbedingung:** Es wird nach vegetarischen Gerichten gefiltert  
**Erwartung:** Alle Gerichte, die angezeigt werden, sind vegetarisch
- [TF122] **Vorbedingung:** Es wird nach Gerichten mit Schweinefleisch gefiltert  
**Erwartung:** Alle Gerichte, die angezeigt werden, enthalten Schweinefleisch
- [TF123] **Vorbedingung:** Es wird nach Gerichten gefiltert, die keine Cashewnüsse enthalten  
**Erwartung:** Alle Gerichte, die angezeigt werden, enthalten keine Cashewnüsse
- [TF124] **Vorbedingung:** Es wird nach Gerichten gefiltert, die unter 5€ kosten  
**Erwartung:** Alle Gerichte, die angezeigt werden, kosten weniger als 5€
- [TF125] **Vorbedingung:** Es wird nach Gerichten gefiltert, die eine Bewertung von mindestens vier Sternen haben  
**Erwartung:** Alle Gerichte, die angezeigt werden, haben eine Bewertung von mindestens vier Sternen
- [TF126] **Vorbedingung:** Es wird nach favorisierten Gerichten gefiltert  
**Erwartung:** Alle Gerichte, die angezeigt werden, sind favorisiert
- [TF127] **Vorbedingung:** Es wird nach neuen Gerichten gefiltert  
**Erwartung:** Alle Gerichte, die angezeigt werden, gab es noch nicht

**[TF13X] Filter speichern****Betroffene Funktionen:** [F404]**Ausführung:** Der Nutzer tippt auf die „Speichern“-Schaltfläche**Varianten:**

- [TF131] **Vorbedingung:** Keine Filtermöglichkeit hat sich geändert
- [TF132] **Vorbedingung:** Eine Filtermöglichkeit hat sich geändert  
**Erwartung:** Die Filterkonfiguration wird lokal gespeichert und der Filterdialog schließt sich

**[TF14X] Filter zurücksetzen****Betroffene Funktionen:** [F405]**Ausführung:** Der Nutzer tippt auf die „Zurücksetzen“-Schaltfläche**Varianten:**

- [TF141] **Vorbedingung:** Filter wird zurückgesetzt  
**Erwartung:** Alle Filtermöglichkeiten werden auf ihre Standardwerte zurückgesetzt

**[TF15X] Filter de-/aktivieren****Betroffene Funktionen:** [F406], [F407]**Ausführung:** Der Nutzer hält das Filtersymbol gedrückt**Varianten:****[TF151] Vorbedingung:** Der Filter ist aktiviert**Erwartung:** Der Filter ist deaktiviert und alle Gerichte des ausgewählten Tages der jeweiligen Mensa werden angezeigt**[TF152] Vorbedingung:** Der Filter ist deaktiviert**Erwartung:** Der Filter ist aktiviert. Alle Gerichte die dem Filter entsprechen werden angezeigt**[TF16X] Galeriedialog ein-/ausblenden****Betroffene Funktionen:** [F301], [F302], [F806]**Ausführung:**

- Galeriedialog ist ausgeblendet: Der Nutzer tippt auf das Bild in der Detailansicht
- Galeriedialog ist eingeblendet: Der Nutzer tippt auf die „Schließen“-Schaltfläche

**Varianten:****[TF161] Vorbedingung:** Der Galeriedialog ist eingeblendet**Erwartung:** Die App zeigt die Detailansicht des jeweiligen Gerichts**[TF162] Vorbedingung:** Der Galeriedialog ist ausgeblendet und zu dem ausgewählten Gericht wurde ein Bild hochgeladen**Erwartung:** Der Galeriedialog öffnet sich und zeigt das Bild der Detailansicht**[TF163] Vorbedingung:** Der Galeriedialog ist ausgeblendet und zu dem ausgewählten Gericht wurde noch kein Bild hochgeladen**Erwartung:** Der Galeriedialog lässt sich nicht öffnen**[TF17X] Bild melden****Betroffene Funktionen:** [F303], [F807]**Ausführung:**

- Der Nutzer tippt auf die „Melden“-Schaltfläche
- Der Nutzer gibt einen Grund zum Melden an
- Der Nutzer bestätigt das Melden

**Varianten:****[TF171] Vorbedingung:** Der Nutzer meldet Bild wie oben beschrieben mit einer Verbindung zum Server**Erwartung:** Das Bild wird dem Nutzer nicht mehr angezeigt und eine temporäre Erfolgsmeldung wird angezeigt**[TF172] Vorbedingung:** Der Nutzer meldet Bild wie oben beschrieben ohne eine Verbindung zum Server**Erwartung:** Eine temporäre Fehlermeldung wird angezeigt**[TF173] Vorbedingung:** Der Nutzer bricht das Melden ab**Erwartung:** Es passiert nichts

**[TF18X] Bild hochladen****Betroffene Funktionen:** [F204], [F805]**Ausführung:**

- Der Nutzer tippt auf die „Hochladen“-Schaltfläche.
- Der Nutzer fügt einen Link zu dem Bild ein.
- Der Nutzer bestätigt das Einbinden des Bildes.

**Varianten:**

**[TF181]** **Vorbedingung:** Die App hat eine Verbindung zum Server und der eingefügte Link ist valide  
**Erwartung:** Das Bild wird eingebunden und angezeigt

**[TF182]** **Vorbedingung:** Die App hat eine Verbindung zum Server und der eingefügte Link ist nicht valide

**[TF183]** **Vorbedingung:** Die App hat keine Verbindung zum Server  
**Erwartung:** Eine temporäre Fehlermeldung wird angezeigt

## 9.2 Sequenztests

Um die Funktionalität des Produkts sicherstellen zu können, müssen Funktionen nicht nur im Einzelnen, sondern auch in Kombinationen und in verschiedenen Reihenfolgen getestet werden. Dadurch ergeben sich folgende grundlegende Funktionsabfolgen, die zu überprüfen sind. Ein Sequenztest ist erfolgreich, wenn er die zu erwartenden Ergebnisse liefert. Bedingungen und Voraussetzungen sind durch die verlinkten Funktionalitäten gegeben oder im Ablauf erwähnt. Grundsätzlich wird von einer funktionierenden Verbindung zum Server ausgegangen, wenn nicht anderweitig spezifiziert.

### [TS01] Gewöhnliche Abfolge bei Gerichtsauswahl

**Betroffene Funktionen:** [F101], [F102], [F301], [F302]

**Sonstige Komponenten:** [UI01], [UI04]

**Ablauf:**

1. Die App wird gestartet [TF01X]
2. Der Nutzer kann den Speiseplan einsehen
3. Ein Gericht auswählen
4. Der Nutzer kann die Informationen zu einem Gericht anschauen
5. Zurück zur Speiseplanansicht navigieren
6. Die letzten drei Schritte beliebig oft wiederholen
7. Die App zeigt die Speiseplanansicht an

### [TS02] Filter auf Speiseplanansicht anwenden

**Betroffene Funktionen:** [F101], [F102], [F401], [F403], [F404]

**Sonstige Komponenten:** [UI01], [UI06]

**Ablauf:**

1. Die App wird gestartet [TF01X]
2. Der Nutzer kann den Speiseplan einsehen
3. Filterdialog öffnen [TF10X]
4. Wählen von beliebiger Filterkombination [TF12X]
5. Übernehmen der Filterkonfiguration [TF13X]
6. Der Nutzer kann den gefilterten Speiseplan einsehen
7. Es werden nur Gerichte angezeigt die dem Filter entsprechen

**[TS03] Einstellen des Filters wird abgebrochen****Betroffene Funktionen:** [F101], [F102], [F401], [F402], [F403]**Sonstige Komponenten:** [UI01], [UI06]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. Filterdialog öffnen [TF10X]
4. Auswahl einer beliebigen Filterkombination [TF12X]
5. Verlassen des Filterdialogs
6. *Keine Änderung in der Speiseplanansicht kann festgestellt werden*

**[TS04] Filter deaktivieren und aktivieren****Betroffene Funktionen:** [F101], [F102], [F406], [F407]**Sonstige Komponenten:** [UI01]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. *Filter auf Speiseplanansicht anwenden [TS02]*
4. Filter deaktivieren [TF15X]
5. *Alle Gerichte des ausgewählten Tages der jeweiligen Mensa werden angezeigt*
6. Filter aktivieren [TF15X]
7. *Es werden nur Gerichte angezeigt, die dem Filter entsprechen*

**[TS05] Lokale Daten, Preisklasse und Filter bleiben gespeichert****Betroffene Funktionen:** [F101], [F102], [F601], [F603]**Sonstige Komponenten:** [UI01], [UI03]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. *Filter auf Speiseplanansicht anwenden [TS02]*
4. Wechseln in die Einstellungsansicht [TF06X]
5. Ändern der Preisklasse [TF07X]
6. App schließen
7. App öffnen [TF01X]
8. *Die Lokalen Daten, die Preisklasse und die Filterkonfiguration sind dieselben wie vor dem Schließen der App*

**[TS06] Der Nutzer meldet ein Bild****Betroffene Funktionen:** [F101], [F102], [F201], [F301], [F303]**Sonstige Komponenten:** [UI01], [UI04], [UI05], [UI11]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. Ein Gericht auswählen [TF05X]
4. *Es wird die Detailansicht mit eingebundenem Bild angezeigt*
5. Bild auswählen [TF16X]
6. *Der Galeriedialog öffnet sich*
7. „Melde“-Schaltfläche betätigen [TF17X]
8. *Der Meldedialog öffnet sich*
9. Kategorie auswählen
10. Meldedialog bestätigen
11. *Das Bild wird ausgeblendet*
12. *Der Administrator bekommt eine Mail zugesendet*

**[TS07] Der Nutzer verlinkt ein Bild zu einem Gericht****Betroffene Funktionen:** [F101], [F102], [F201], [F205]**Sonstige Komponenten:** [UI01], [UI04], [UI10]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. Ein Gericht auswählen [TF05X]
4. Auf das „Hochladen“-Symbol tippen [TF18X]
5. *Ein Dialog öffnet sich*
6. Link zum Bild eines Bildhosters einfügen
7. Dialog wird bestätigt
8. *Server validiert die Eingabe*

**[TS08] Mensa hat zum ausgewähltem Tag keine Gerichte****Betroffene Funktionen:** [F101], [F102], [F105]**Sonstige Komponenten:** [UI01], [UI07]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. Tag ändern auf einen Tag, an dem die ausgewählte Mensa geschlossen ist [TF091]
4. *Es wird angezeigt, dass die Mensa geschlossen ist*

**[TS09] Speiseplan aktualisieren ohne Serververbindung****Betroffene Funktionen:** [F101], [F102], [F103], [F201]**Sonstige Komponenten:** [UI01], [UI04]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. Internetzugang auf dem Gerät deaktivieren
4. Speiseplan aktualisieren [TF04X]
5. *Speiseplan wird mit lokalen Daten geladen*
6. *Der Nutzer kann den Speiseplan einsehen*
7. Ein Gericht auswählen [TF05X]
8. *Die Detailansicht öffnet sich mit lokalen Daten*

**[TS10] Filter schließt alle Gerichte aus****Betroffene Funktionen:** [F101], [F102], [F401], [F403], [F404]**Sonstige Komponenten:** [UI01], [UI06], [UI13]**Ablauf:**

1. Die App wird gestartet [TF01X]
2. *Der Nutzer kann den Speiseplan einsehen*
3. Filterdialog öffnen [TF10X]
4. Filter anpassen, sodass kein Gericht die Filterkriterien erfüllt [TF120]
5. Filter speichern [TF13X]
6. *Es werden keine Gerichte angezeigt, da alle durch den Filter ausgeschlossen sind*

## 10 Benutzerschnittstelle

Die grafische Benutzeroberfläche soll möglichst intuitiv und ohne Anleitungen zu bedienen sein. Da es einen Dark- und einen Lightmode gibt, existieren für jede Ansicht und jeden Dialog zwei Entwürfe und Farbschemata.

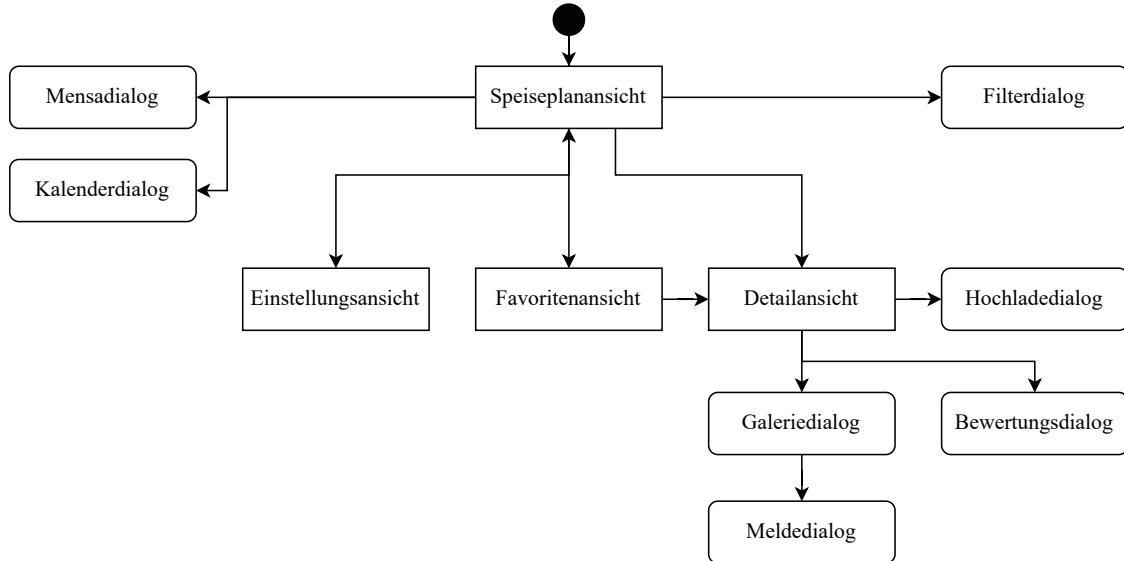


Abbildung 4: UI-Übersicht

Durch das Tippen auf die Symbole auf der Navigationsleiste lässt sich zwischen der Speiseplanansicht, der Favoritenansicht und der Einstellungsansicht wechseln. Die Navigationsleiste befindet sich am unteren Rand des Bildschirms.

Temporäre Fehlermeldungen werden als Banner am unteren Rand des Bildschirms angezeigt. Die temporären Fehlermeldungen verschwinden nach einer kurzen Zeit.

### 10.1 Ansichten und Dialoge

Es wird hier zwischen Ansichten und Dialogen unterschieden. Ansichten beschreiben eine Oberfläche, welche eigenständig Inhalte und Funktionen anbietet. Ein Dialog ist eine Oberfläche, welche eine Erweiterung zu einer oder mehreren Ansichten darstellt und zusätzliche Funktionen anbietet. Ein Dialog wird über der entsprechenden Ansicht eingeblendet und kann dabei entweder den gesamten Bildschirmbereich oder nur einen Ausschnitt des Bildschirmes einnehmen. Hierbei wird dann die Ansicht im Hintergrund entsprechend abgedunkelt, um den Dialog hervorzuheben.

#### [UI01] Speiseplanansicht

Die Speiseplanansicht enthält eine Übersicht über die verschiedenen Gerichte, die an einem Tag angeboten werden. Dafür gibt es zwei Darstellungsformen, um die verschiedenen Gerichte anzuzeigen:

- Speiseplanansicht Galerie (vgl. Abbildung 5)
- Speiseplanansicht Liste (vgl. Abbildung 6)

Durch eine Schaltfläche lässt sich zwischen den beiden Darstellungsformen wechseln. Standardmäßig wird die Speiseplanansicht als Galerie ausgewählt.

Für die ausgewählte Mensa, deren Speiseplan gerade sichtbar ist, werden die Gerichte der verschiedenen Linien angezeigt, die es dort gibt. Dabei werden pro Gericht jeweils folgende Informationen angezeigt:

- Gerichtsbezeichnung
- Gerichtstyp
- Preis
- Linie
- Bild, falls kein Bild für das Gericht vorhanden ist, dann wird ein Standard-Bild angezeigt
- Bewertung
- ob ein Gericht ein Favorit ist
- Beilagen (nur in Galerie-Darstellung)

In folgenden Sonderfällen wird nicht der Speiseplan, sondern verschiedene Nachrichten angezeigt:

[UI12] Geschlossene Mensa

[UI13] Keine Filterergebnisse

[UI14] Keine Daten

Von der Speiseplanansicht kann man zu folgenden Ansichten und Dialogen navigieren:

[UI02] Favoritenansicht

[UI03] Einstellungsansicht

[UI04] Detailansicht

[UI06] Filterdialog

[UI07] Kalenderdialog

[UI08] Mensadialog

### **[UI02] Favoritenansicht**

Die Favoritenansicht (vgl. Abbildung 7) enthält eine Übersicht über alle favorisierten Gerichte in Form einer Liste. Dabei werden folgende Informationen angezeigt:

- Gerichtsbezeichnung
- Gerichtstyp
- Preis
- Bild
- Bewertung

Von der Favoritenansicht kann man zu folgenden Ansichten navigieren:

[UI01] Speiseplanansicht

[UI03] Einstellungsansicht

[UI04] Detailansicht

### **[UI03] Einstellungsansicht**

Die Einstellungsansicht (vgl. Abbildung 8) enthält unterschiedliche Einstellungsmöglichkeiten und Informationen. Dazu gehören:

- Farbschema mit den Optionen:

- Lightmode
- Darkmode
- systemgesteuerter Modus
- Preisklasse mit den Optionen:
  - Studierende
  - Mitarbeitende
  - Schüler
  - Gäste
- Benachrichtigungseinstellungen mit den Einstellungen
  - Aktivieren/Deaktivieren
  - Benachrichtigungszeitpunkt
- In- und Exportmöglichkeiten
- Softwareinformationen
  - Version der Software
  - Lizenz der Software
  - Link zum Quellcode der Software
  - Link zu einer Kontaktmöglichkeit zum Anbieter der Software
  - Link zu Informationen zum Datenschutz

Von der Einstellungsansicht kann man zu folgenden Ansichten und Dialogen navigieren:

[UI01] Speiseplanansicht

[UI02] Favoritenansicht

#### **[UI04] Detailansicht**

Die Detailansicht (vgl. Abbildung 9) zeigt genauere Informationen zu dem ausgewählten Gericht an. Dazu gehören:

- Gerichtsbezeichnung
- Gerichtstyp
- Linie
- Beilagen
- Bild, falls kein Bild für das Gericht vorhanden ist, dann wird ein Standard-Bild angezeigt
- Preis
- Allergene und Zusatzstoffe, falls eingeblendet
- Häufigkeit
- letztes Angebotsdatum
- Durchschnittsbewertung
- eigene Bewertung (falls vorhanden)
- ob ein Gericht ein Favorit ist

Von der Detailansicht kann man zu folgenden Ansichten und Dialogen navigieren:

[UI01] Speiseplanansicht, falls von der Speiseplanansicht aus aufgerufen

[UI02] Favoritenansicht, falls von der Favoritenansicht aus aufgerufen

[UI05] Galeriedialog

[UI09] Bewertungsdialog

[UI10] Hochladedialog

### **[UI05] Galeriedialog**

Der Galeriedialog (vgl. Abbildung 10) zeigt die Bilder eines Gerichts im Vollbildmodus an, wobei immer nur ein Bild gleichzeitig angezeigt wird. Die Reihenfolge, in der die Bilder angezeigt werden, ergibt sich aus dem Bildrang. Dabei gibt es mehrere Schaltflächen:

- Ansicht schließen
- Aufwerten
- Abwerten
- Bild melden

Durch Wischen kann man durch die Bilder wechseln.

Von dem Galeriedialog lassen sich folgende Ansichten und Dialoge erreichen:

[UI04] Detailansicht

[UI11] Meldedialog

### **[UI06] Filterdialog**

Der Filterdialog (vgl. Abbildung 11) ist ein Dialog, über den verschiedene Filtereinstellungen und Sortierungen genutzt werden können. Dazu gehört das Filtern nach:

- Gerichtstyp mit den exklusiven Optionen:
  1. alles mit den (nicht-exklusiven) Wahlmöglichkeiten:
    - Rind
    - Schwein
    - Fisch
  2. vegetarisch
  3. vegan

Bei jeder der drei Optionen besteht die Möglichkeit, Gerichte „ohne Angabe“ einzuschließen.

- Allergene
- Preis
- Bewertung
- Favoriten
- Häufigkeitsklassen

Und das Sortieren nach (je auf- und absteigend):

- Linie
- Preis
- Bewertung
- Seltenheit

Vom Filterdialog lässt sich nur die Speiseplanansicht erreichen.

**[UI07] Kalenderdialog**

Der Kalenderdialog (vgl. Abbildung 12) ist ein Dialog, der aus einem Kalender besteht, auf dem ein Tag ausgewählt werden kann. Mit Schaltflächen kann direkt zum nächsten oder vorherigen Tag navigiert werden. Vom Kalenderdialog lässt sich nur die Speiseplanansicht erreichen.

**[UI08] Mensadialog**

Der Mensadialog (vgl. Abbildung 13) ist ein Dialog, der alle verfügbaren Menschen des Studierendenwerks Karlsruhe anzeigt. Aus diesen kann man eine Mensa auswählen, deren Speiseplan angezeigt wird. Vom Mensadialog lässt sich nur die Speiseplanansicht erreichen.

**[UI09] Bewertungsdialog**

Der Bewertungsdialog (vgl. Abbildung 14) ist ein Dialog, durch den der Nutzer eine Bewertung von ein bis fünf Sternen abgeben kann. Durch eine Bestätigung durch den Nutzer wird diese dann dem Server übermittelt. Vom Bewertungsdialog lässt sich nur die Detailansicht erreichen.

**[UI10] Hochladedialog**

Der Hochladedialog (vgl. Abbildung 15) ist ein Dialog, der einen Link in Empfang nimmt, der zu einem Bild führt. Durch eine Bestätigung durch den Nutzer wird der Link dann dem Server übermittelt. Außerdem enthält er eine Anleitung für den Nutzer, wie er ein Bild hochladen kann, und einige rechtliche Hinweise. Vom Hochladedialog lässt sich nur die Detailansicht erreichen.

**[UI11] Meldedialog**

Der Meldedialog (vgl. Abbildung 16) ist ein Dialog, durch den der Nutzer das Melden eines Bildes bestätigen muss, bevor das Melden dem Server übermittelt wird. Außerdem muss der Nutzer einen Grund auswählen, warum er das Bild meldet. Dabei hat er folgende Auswahlmöglichkeiten:

- Anstößig
- Werbung
- kein Gericht
- falsches Gericht
- Verletzt meine Rechte
- Sonstiges

Vom Meldedialog lässt sich nur der Galeriedialog erreichen.

## 10.2 Fehlermeldungen

### [UI12] Geschlossene Mensa

Diese Fehlermeldung wird angezeigt, wenn das Studierendenwerk für einen Tag, der weniger als vier Wochen in der Zukunft liegt, keine Daten liefert.

Die Anzeige wird anstelle von Speiseplanansicht angezeigt. (vgl. Abbildung 17)

### [UI13] Keine Filterergebnisse

Diese Fehlermeldung wird angezeigt, wenn durch den aktiven Filter alle Gerichte des Tages ausgeschlossen werden.

Durch das Tippen auf eine Schaltfläche lässt sich der Filter temporär deaktivieren.

Die Anzeige wird anstelle von Speiseplanansicht angezeigt. (vgl. Abbildung 18)

### [UI14] Keine Daten

Die Fehlermeldung wird angezeigt, wenn für den ausgewählten Tag keine Daten vorhanden sind, das heißt der Tag mehr als vier Wochen in der Zukunft liegt oder keine Verbindung zum Server aufgebaut werden kann und es keine Daten gibt, die auf dem Client zwischengespeichert sind.

Durch das Tippen auf eine Schaltfläche wird erneut versucht eine Verbindung zum Server aufzubauen.

Die Anzeige wird anstelle von Speiseplanansicht angezeigt. (vgl. Abbildung 19)

### 10.3 Entwürfe grafische Benutzerschnittstelle

Diese Entwürfe stellen ausschließlich Skizzen der Benutzerschnittstelle dar und müssen nicht so implementiert werden, sondern dienen der Veranschaulichung der Funktionalitäten.

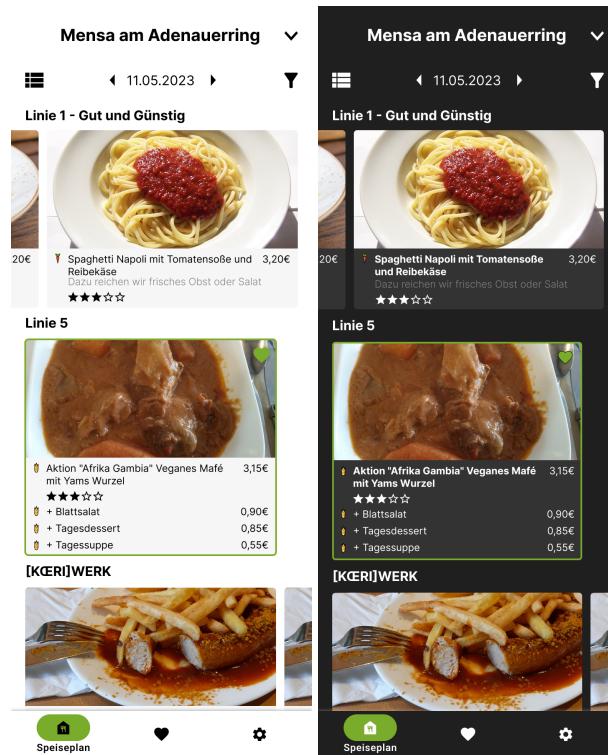


Abbildung 5: Speiseplanansicht Galerie (Lightmode/Darkmode)

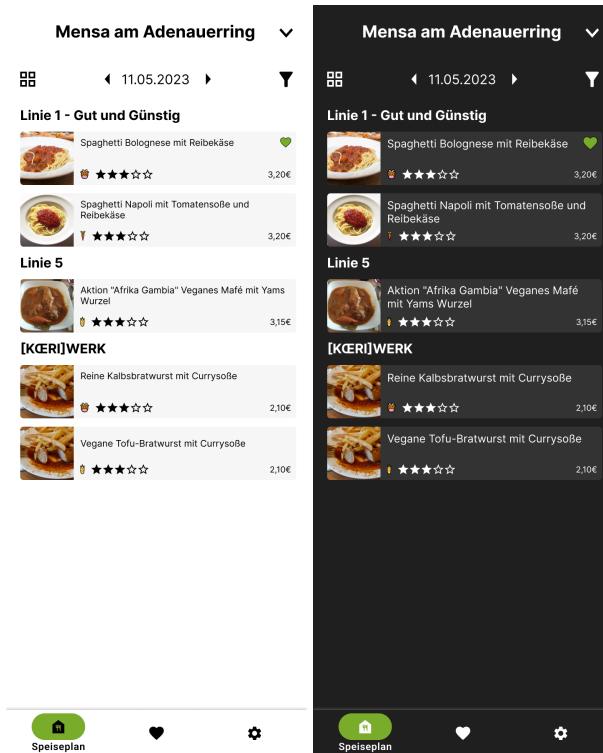


Abbildung 6: Speiseplanansicht Liste (Lightmode/Darkmode)

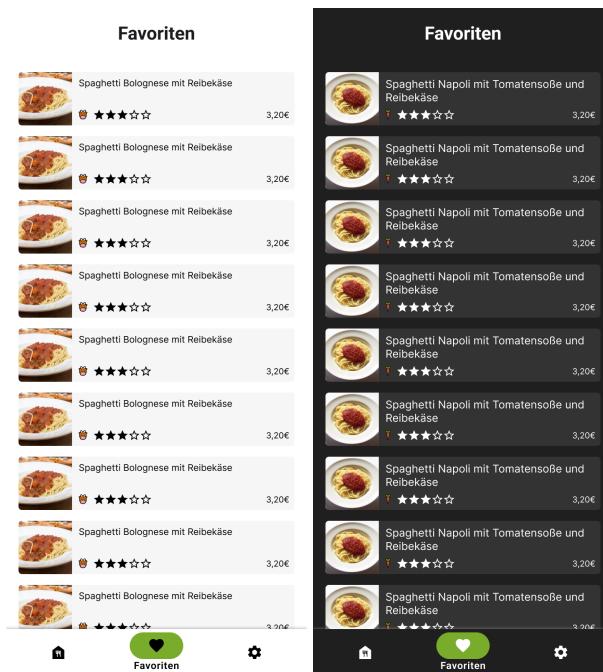


Abbildung 7: Favoritenansicht (Lightmode/Darkmode)

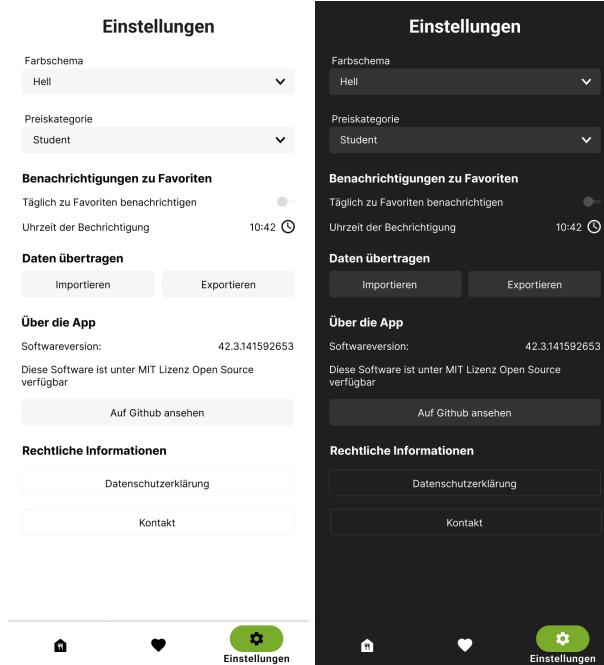


Abbildung 8: Einstellungsansicht (Lightmode/Darkmode)

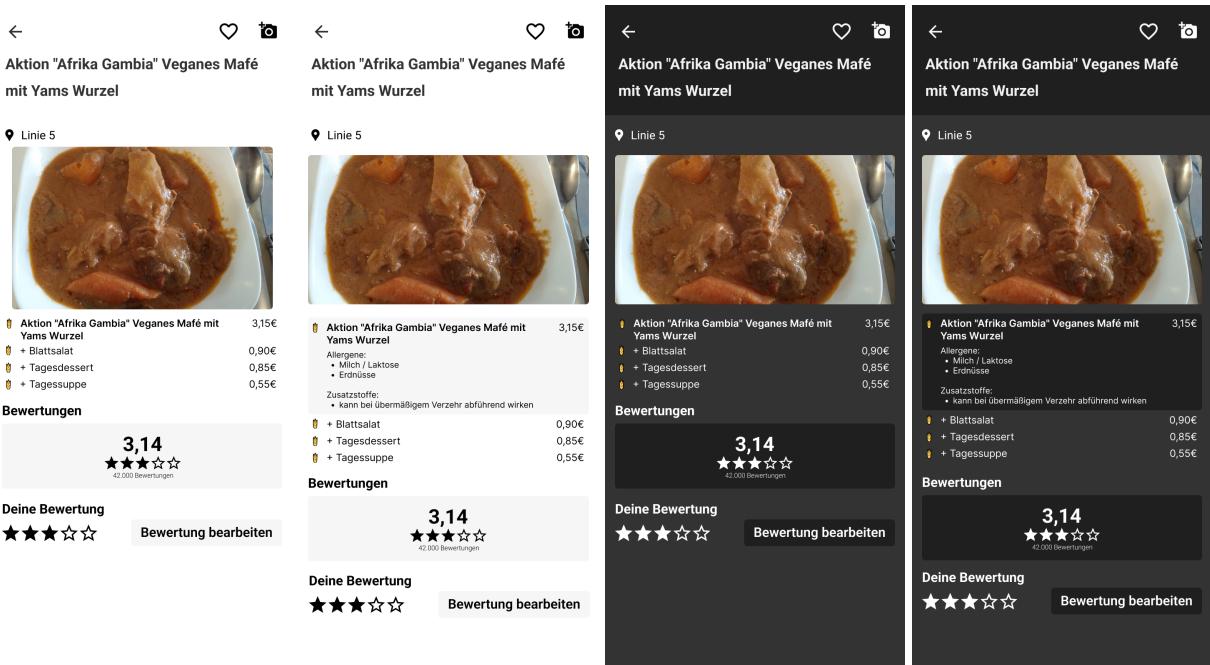


Abbildung 9: Detailansicht (Lightmode/Darkmode jeweils in Normaler Ansicht sowie mit Allergenen zu einem Gericht eingebettet)

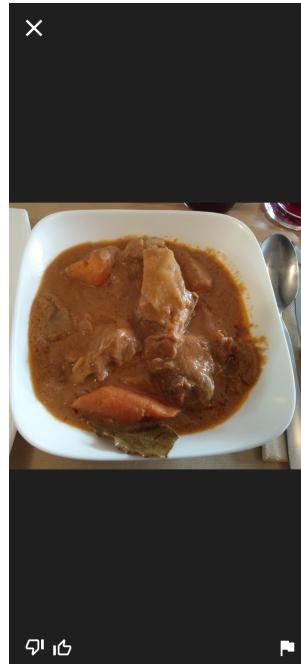


Abbildung 10: Galeriedialog für Lightmode und Darkmode

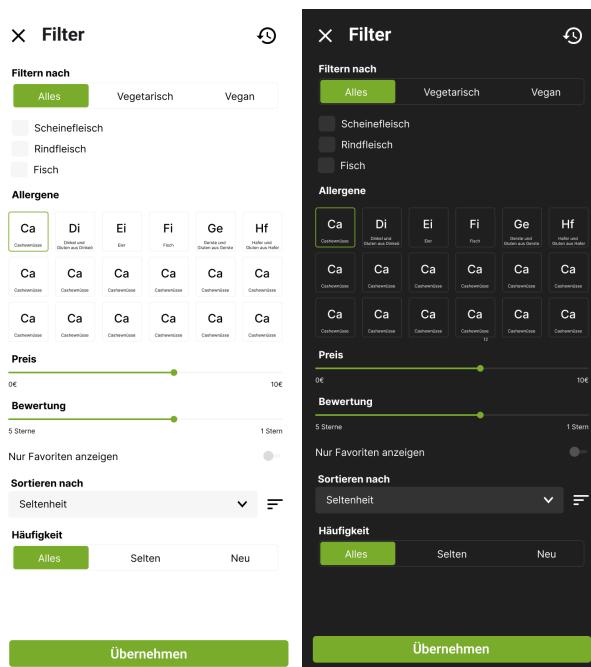


Abbildung 11: Filterdialog (Lightmode/Darkmode)

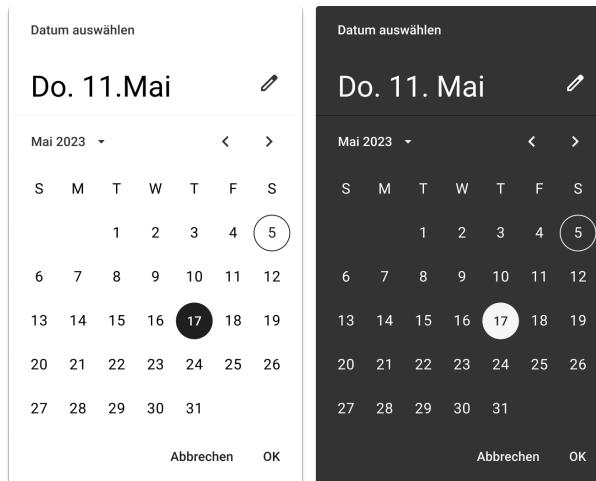


Abbildung 12: Kalenderdialog (Lightmode/Darkmode)

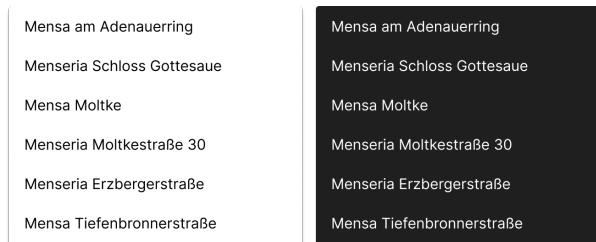


Abbildung 13: Mensadialog (Lightmode/Darkmode)



Abbildung 14: Bewertungsdialog (Lightmode/Darkmode)

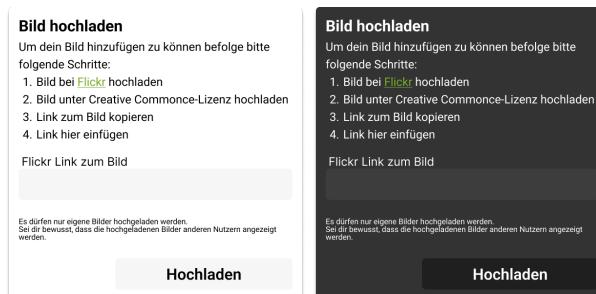


Abbildung 15: Hochladedialog (Lightmode/Darkmode)

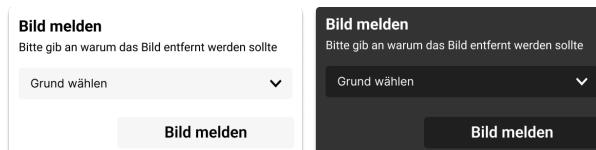


Abbildung 16: Meldedialog (Lightmode/Darkmode)

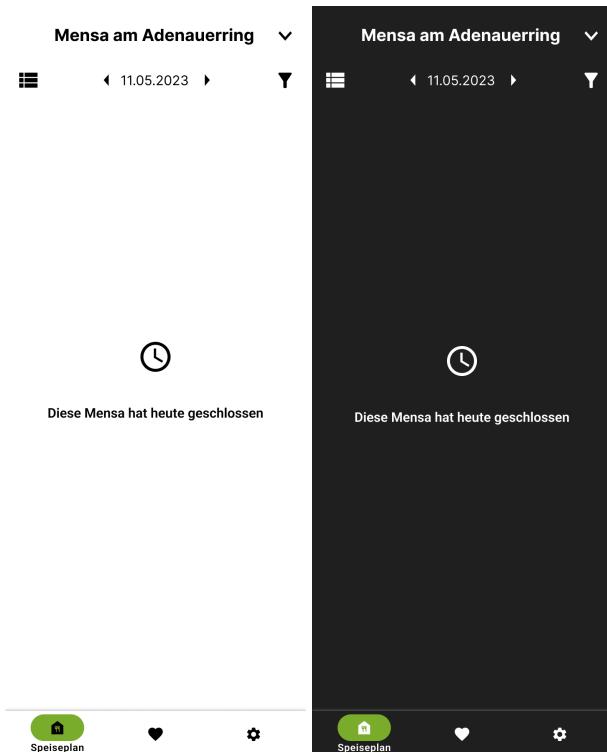


Abbildung 17: Geschlossene Mensa (Lightmode/Darkmode)

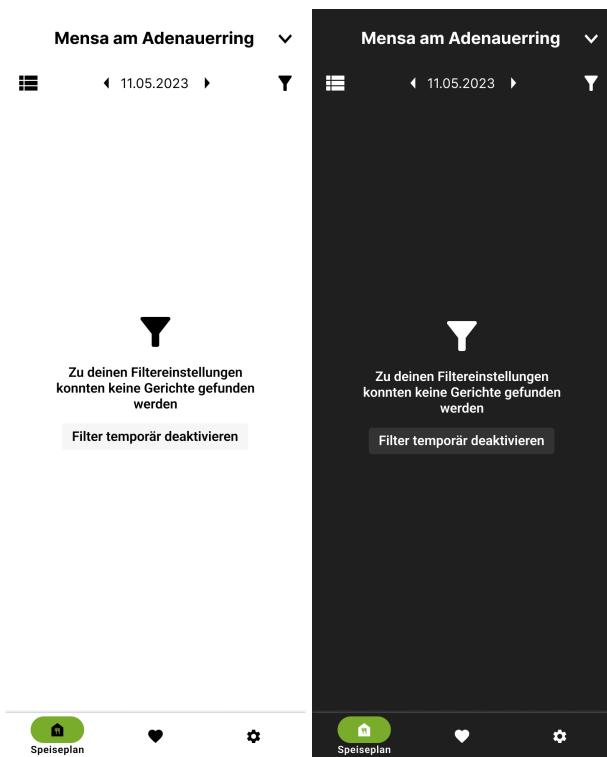


Abbildung 18: Keine Filterergebnisse (Lightmode/Darkmode)

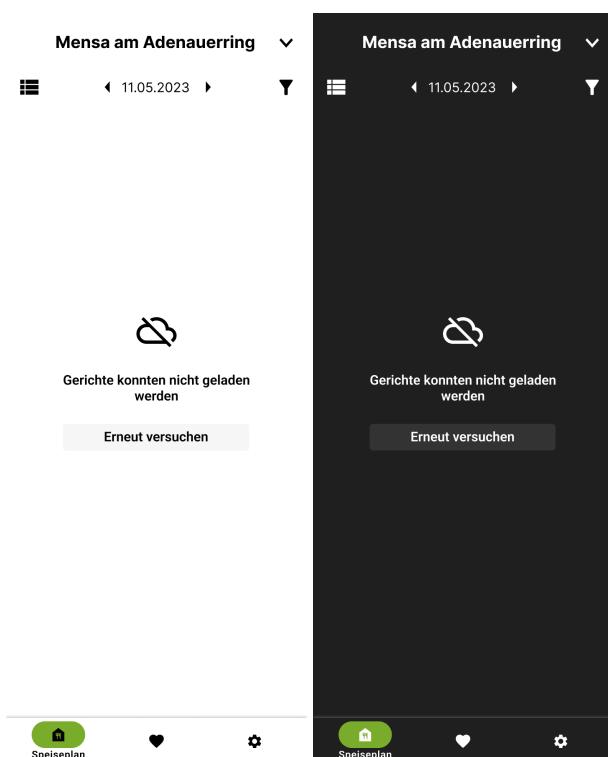


Abbildung 19: Keine Daten (Lightmode/Darkmode)

# 11 Glossar

**Abwertung** Markierung eines Bildes als nicht hilfreich.

**Accountstrukturen** Konzept für Verwaltung von Nutzern, bei dem jeder Nutzer eigene Zugangsinformationen erhält, mithilfe deren er sich identifizieren kann.

**Administrator** Person, welche für den Betrieb des Servers zuständig ist.

**Aktualisierungsabfrage** Speiseplandaten für den aktuellen Tag werden von der Mensa-Webseite abgefragt.

**Allergene** Die in einem Gericht enthaltenen allergenen Stoffe.

**AMD Ryzen** Familie an Prozessoren der Firma „Advanced Micro Devices“.

**Android** Betriebssystem für Smartphones

**Android SDK** Android Software-Development-Kit. Sammlung an Werkzeugen zur Entwicklung von Android-Apps.

**Ansicht** Oberfläche, welche eigenständig Inhalte und Funktionen anbietet.

**API** Application-Programm-Interface. Definierte Schnittstelle, über welche unterschiedliche Programme kommunizieren können.

**API-Schlüssel** Eine Art Passwort, mithilfe welcher sich ein Programm authentifizieren kann.

**App** Client zum Anzeigen der Gerichte und ausführen anderer Funktionen.

**Arbeitsspeicher** Bestandteil eines Computers der für die Zwischenspeicherung von Daten zuständig ist.

**Aufwertung** Markierung eines Bildes als hilfreich.

**Authentifizierung** Verfahren, das die Identität eines Nutzers oder Clients sicherstellt.

**Bandbreite** Maximalmenge an Daten welche in einer festen Zeiteinheit über eine Leitung transportiert werden kann.

**Banner** Kleine Textnachricht, die am unteren Rand des Bildschirms auftaucht, um dem Nutzer etwas mitzuteilen. Wird auch „Snackbar“ genannt.

**Beilage** Zugabe zu einem Gericht.

**Benachrichtigung** Nachricht, die auf das Smartphone des Nutzers gesendet wird.

**Benachrichtigungszeitpunkt** Uhrzeit, zu welcher Benachrichtigungen für Gerichte versendet werden.

**Benutzeroberfläche** Zeigt dem Nutzer Informationen an und bietet Bedienelemente.

**Betriebssystem** Software, die Grundfunktionen für die Benutzung eines Gerätes zur Verfügung stellt.

**Bewertung** Durchschnittliche Bewertung des Gerichts.

**Bildhoster** Dienstleister, der Nutzern erlaubt, Bilder im Internet zugänglich zu machen.

**Bildrang** Indikator für die Beliebtheit eines Bildes. Wird wird aus Anzahl der Aufwertungen bzw. Anzahl der Abwertungen eines Bildes, des Alters eines Bildes und der Anzahl der Meldeanträgen berechnet. Je größer der Bildrang, desto früher wird dieses angezeigt.

**Campus-Nord-Mensa** Mensa auf dem Campus Nord des KITs, die nicht vom Studierendenwerk Karlsruhe betrieben wird.

**Client** Rolle für ein Gerät, welches Informationen von einer zentralen Stelle, einem Server, abfragt.

**Client-Identifikator** Information, mithilfe welcher ein Client eindeutig identifiziert werden kann.

**Client-Server-Modell** Bezeichnung für den Aufbau einer Anwendung, welche aus einem Client und einem Server besteht.

**Computer** Gerät, welches mit Rechenoperationen beauftragt werden kann und diese automatisiert und selbstständig vornimmt.

**CPU-Kern** Bestandteil eines Prozessors der eine gleichzeitige Berechnung ausführen kann.

**Darkmode** Modus, in welcher die Benutzeroberfläche in dunklen Farben dargestellt wird.

**Datenbank** Anwendung, welche die persistente Speicherung von Daten übernimmt und verwaltet.

**Dialog** Oberfläche, welche eine Erweiterung zu einer oder mehreren Ansichten darstellt und zusätzliche Funktionen anbietet.

**Docker** Standardisiertes Format für Computerprogramme, welches die Installation und den Betrieb erleichtert.

**Docker Engine** Software, welche Docker-Container ausführt.

**Dropdown-Menü** Liste an Wahlmöglichkeiten, welche erst nach betätigen einer Schaltfläche erscheint.

**DSGVO** Datenschutzgrundverordnung. Verordnung der Europäischen Union, welche die Verarbeitung von personenbezogenen Daten reguliert.

**Export** Vorgang bei welchem ein Programm gespeicherte Daten in einem definierten Format ausgibt, sodass der Nutzer diese an anderer Stelle von einem Programm einlesen lassen kann (Import).

**Export-Datei** Datei, die exportierte Einstellungen und Favoriten enthält.

**Export-Daten** Daten, welche aus dem Client in eine Export-Datei exportiert werden können.

**Farbschema** Auswahl an Farben, in welchen die Benutzeroberfläche dargestellt wird. Möglich sind Lightmode, Darkmode oder die automatische Wahl einer der beiden Optionen abhängig von den Einstellungen im Betriebssystem.

**Fat-Client** Client, auf welchem ein großer Teil der Funktionalität implementiert ist und nur die nötigsten Daten beim Server abfragt.

**Fehlermeldung** Ein Text, der angezeigt wird, wenn ein Fehler aufgetreten ist.

**Festplattenspeicher** Gerät, welches Daten in großen Mengen persistent abspeichern kann.

**Filterkonfiguration** Eine Menge von Einschränkungen der Eigenschaften der Gerichte, die angezeigt werden.

**Flickr** Ein Bildhoster, der die Einbindung seiner Bilder in andere Anwendungen erlaubt.

**Flutter-Framework** Sammlung von Softwarebibliotheken zu Entwicklung von plattformunabhängigen Anwendungen.

**funktionale Anforderungen** Anforderungen an ein Produkt, die die genaue Funktionsweise einzelner Funktionen und Abläufe beschreiben.

**Gericht** Portionierte, zubereitete Nahrung.

**Gerichtsbezeichnung** Name des Gerichts.

**Gerichtsidentifikator** Synthetisches Merkmal, das ein Gericht eindeutig Identifiziert.

**Gerichtstyp** Kategorisierung eines Gerichts in „vegetarisch“, „vegan“, „Schweinefleisch“, „Rindfleisch“, „Fisch“ oder „keine Angabe“.

**GraphQL** Spezifikation für eine API.

**Häufigkeit** Die Anzahl, wie oft ein Gericht in einem Zeitintervall von drei Monaten angeboten wurde.

**Häufigkeitsklasse** Unterteilung der Gerichte in neue Gerichte, Gerichte, die es selten (z.B. ein mal in drei Monaten) gibt und Gerichte, die es täglich gibt, in Abhängigkeit ihrer Häufigkeit.

**Import** Siehe Export.

**iOS** Betriebssystem für Smartphones.

**KIT** Karlsruher Institut für Technologie. Universität in Baden-Würtemberg.

**Komplettabfrage** Speiseplandaten für die nächsten vier Wochen werden von der Mensa-Webseite abgefragt.

**Ladeanimation** Eine Animation, die angezeigt wird, während die App auf den Abschluss eines Vorgangs wartet.

**Letztes Angebotsdatum** Das Datum an dem ein Gericht zum letzten Mal in einer bestimmten Mensa angeboten wurde.

**Lightmode** Modus, in welcher die Benutzeroberfläche in hellen Farben dargestellt wird.

**Linie** Bezeichnet eine Ausgabe einer Mensa, an welcher verschiedene Gerichte angeboten werden. Es existieren auch Linien an welchen mehrere (meist zwei) Gerichte angeboten werden. Oft werden auch ein Hauptgericht und eine oder mehrere Beilagen angeboten.

**Mailserver** Server, der für das Versenden und Zustellen von E-Mails zuständig ist.

**Meldeantrag** Antrag eines Nutzers, ein Bild aufgrund unpassenden Inhalts zu entfernen.

**Meldemaß** Zahlenwert für ein Bild, der die Dringlichkeit zur Entfernung dieses beschreibt. Er berechnet sich aus der Anzahl der Meldeanträgen und der Beliebtheit eines Bildes.

**Meldeschranke** Untere Schranke an das Meldemaß, damit ein Bild automatisch entfernt wird.

**Mensa** Von anderen Menschen räumlich getrennte Verpflegungsstätte, bestehend aus einer oder mehreren Linien, an welchen Gerichte angeboten werden.

**Mensa-Webseite** Webseite des Studierendenwerk Karlsruhe, von welcher die aktuellen Speisepläne gelesen werden.

**MIT-Lizenz** Lizenz für Software, welche die Verwendung und Modifizierung dieser nahezu uneingeschränkt erlaubt.

**modular** Adjektiv, welches ein System beschreibt, bei welchem einzelne Bestandteile einfach ausgetauscht werden können.

**Navigationsleiste** Abschnitt in einer Benutzeroberfläche, der die Navigation zwischen Speiseplanansicht, Favoritenansicht und Einstellungsansicht ermöglicht.

**nichtfunktionale Anforderungen** Anforderungen an ein Produkt, die anstatt der Funktionsweise die Qualität und Rahmenbedingungen beschreiben.

**ordnungsgemäßer Betrieb** Betriebszustand des Produkts, in welchem die Erfüllung der Anforderungen garantiert wird.

**parsen** Verb, das den Vorgang des maschinellen Auslesens von Informationen aus einem Text beschreibt.

**persistent** Adjektiv, welches eine Speicherung von Daten beschreibt, die über lange Zeiträume auch ohne Energieversorgung gewährleistet ist.

**personenbezogene Daten** Daten, welche einen direkten Bezug zu einer Person besitzen, wie z.B. ein Geburtsdatum oder eine Lieblingsfarbe.

**Postgresql** Beispiel für eine Software, welche eine Datenbank verwaltet.

**Preis** Preis des Gerichts (Abhängig von der gewählten Preisklasse).

**Preisklasse** Klasse an Personen, welche den selben Preis für ein Gericht zahlen müssen: Studierender, Mitarbeitender, Schüler oder Gast.

**Produkt** Bezeichnet die von diesem Pflichtenheft spezifizierten Komponenten, mit welchen die spezifizierten Leistungen erbracht werden sollen. Beinhaltet insbesondere den Server und die App.

**Programmcode** Folge an Anweisungen, die den Ablauf und das Verhalten eines Computerprogramms definieren. Bildet die Grundlage für eine Software.

**Programmiersprache** Formal definierte Sprache, mit welcher Computer mit Berechnungen beauftragt werden.

**Projekt** Zeitlich begrenzte Unternehmung mit klar definiertem Ziel. Hier ist damit die Erstellung des Produkts gemeint.

**Protokoll** Liste von Ereignissen und Fehlern, die vom Server festgehalten (protokolliert) wurden.

**Protokollierungsstufe** Wichtigkeit einer Protokollmeldung, nach welcher im Protokoll gefiltert werden kann.

**Prozessor** Bestandteil eines Computers, der für Berechnungen zuständig ist.

**Reverse-Proxy** Server, der vor einen anderen Server geschaltet wird und einkommende Anfragen umleitet. Dieser kann dabei Anfragen auf mehrere Server je nach gewünschtem Ziel weiterleiten oder sich um eine Transportverschlüsselung kümmern.

**Rust** Programmiersprache, welche für den Server verwendet wird.

**Schaltfläche** Bedienelement, das auf Berührung reagiert.

**Server** Dient zur Bereitstellung und Synchronisierung der von den Clients benötigten Daten.

**serverseitig** Adjektiv, welches beschreibt, dass etwas auf der Seite des Servers passiert.

**Software** Computerprogramm

**Softwarebibliothek** Programmcode welcher von Dritten bereitgestellt wird und eine bestimmte Funktionalität abdeckt. Dieser kann von einem anderen Programmcode wiederverwendet werden

**SOLID-Designprinzipien** Grundprinzipien der Softwareentwicklung, welche eine gute Strukturierung und Erweiterbarkeit einer Software sicherstellen sollen.

**Sortierung** Bestimmt die Reihenfolge einer Liste an Elementen anhand von Eigenschaften der jeweiligen Elemente.

**Speiseplan** Menge aller Gerichte, die in einer Mensa an einem Tag angeboten werden.

**Speiseplandaten** Die zu einem Speiseplan gehörenden Daten.

**Standard-Bild** Ein Bild, das als Platzhalter für andere Bilder angezeigt wird, wenn kein Bild von einem Gericht vorhanden ist.

**Standardwert** Vorgegebener Wert für eine Option in der Filterkonfiguration. Dieser lässt alle Gerichte zu.

**Studierendenwerk Karlsruhe** Betreiber der Menschen an den Hochschulen und Universitäten in Karlsruhe und Pforzheim.

**Systemdialog** Vom Betriebssystem bereitgestellter Dialog, z.B. zum Wählen einer Datei.

**systemgesteuerter Modus** Modus einer Benutzeroberfläche, die ihr Farbschema anhand von Einstellungen des Betriebssystems automatisch wählt.

**Taktrate** Frequenz, in welcher ein CPU-Kern einzelne Berechnungsschritte ausführt.

**Transportverschlüsselung** Verfahren zur Verbergen des Inhalts einer Nachricht vor Dritten.

**Umgebungsvariable** Möglichkeit, einem Programm beim Start Informationen zu übergeben.

**Usecase** Ein Anwendungsfall, der eine mögliche Interaktion mit dem Produkt beschreibt.

**Web** Zielplattform, auf welcher Anwendungen nicht manuell installiert werden müssen, sondern direkt von einem Internet-Browser aus aufgerufen werden können.

**Zielplattform** Softwaresystem, für welches eine Anwendung ausgelegt ist. Beispiele: Android, iOS, Web

**Zusatzstoffe** Stoffe, die zu einem Gericht hinzugefügt werden, um dessen Eigenschaften zu beeinflussen. Diese müssen zu einem Gericht explizit angegeben werden.