

**Pflichtenheft**

Plan4BA



Autoren Maximilian Schädlich

Robert Beier

Eric Mäuer

Seminargruppe CS16-1

Version 1.0

Auftraggeber Staatliche Studienakademie Leipzig

Studienrichtung Informatik

Herr Prof. Dr.-Ing. Christian Heller

Schönauer Straße 113 a

04207 Leipzig

Leipzig,

Inhaltsverzeichnis

[0 Über dieses Dokument 4](#_Toc532463189)

[1 Zielbestimmung 5](#_Toc532463190)

[1.1 Musskriterien 5](#_Toc532463191)

[1.2 Wunschkriterien 6](#_Toc532463192)

[1.3 Abgrenzungskriterien 6](#_Toc532463193)

[2 Produktübersicht 7](#_Toc532463194)

[2.1 Produkteinsatz 7](#_Toc532463195)

[2.1.1 Anwendungsbereiche 7](#_Toc532463196)

[2.1.2 Zielgruppe 7](#_Toc532463197)

[2.1.3 Betriebsbedingungen 7](#_Toc532463198)

[2.2 Produktfunktionen 7](#_Toc532463199)

[2.2.1 Funktionale Anforderung 7](#_Toc532463200)

[2.2.2 Nichtfunktionale Anforderungen 8](#_Toc532463201)

[2.3 Produktumgebung 8](#_Toc532463202)

[2.3.1 Software 9](#_Toc532463203)

[2.3.2 Hardware 9](#_Toc532463204)

[2.3.3 Orgware 9](#_Toc532463205)

[2.4 Produktdaten 10](#_Toc532463206)

[2.5 Produktbeschreibung 11](#_Toc532463207)

[2.6 Benutzeroberfläche 12](#_Toc532463208)

[3 Entwicklungsumgebung 15](#_Toc532463209)

[3.1.1 Software 15](#_Toc532463210)

[3.1.2 Hardware 15](#_Toc532463211)

[3.1.3 Orgware 15](#_Toc532463212)

[4 Qualitätsanforderungen 16](#_Toc532463213)

[5 Testfälle und Testszenarien 17](#_Toc532463214)

[6 Verzeichnisse 18](#_Toc532463215)

[6.1 Versionsverzeichnis 18](#_Toc532463216)

[6.2 Abbildungsverzeichnis 18](#_Toc532463217)

[6.3 Tabellenverzeichnis 18](#_Toc532463218)

# Über dieses Dokument

Im Rahmen des Moduls „Softwareprojekt“ soll ein eigenständiges Softwareprojekt durchgeführt werden. Dieses Dokument stellt in diesem Zusammenhang das Pflichtenheft dar. Dieses ist die Vertragsgrundlage zwischen Auftragnehmer und -geber. In diesem Fall ist der Auftraggeber der Modulverantwortlich in Person von Prof. Dr.-Ing. Christian Heller und die Auftragnehmer sind die Studenten Maximilian Schädlich, Robert Beier und Eric Mäuer.

Der erste Abschnitt dieses Dokuments befasst sich mit den Zielen dieses Projekts. Dementsprechend wird dargestellt, was im Zuge dieses Projekts unbedingt oder eventuell umgesetzt werden soll und was nicht. Darauffolgend soll näher auf das Produkt, welches als Resultat entstehen soll, eingegangen werden. Dazu gehört unter anderem die genauere Betrachtung des Produkteinsatzes und demzufolge dem Problembereich. Außerdem werden die Funktionalitäten, sowie die Daten, welche das System verwaltet, genauer betrachtet, bevor zum Schluss des Abschnitts die Idee hinter der Benutzeroberfläche mit ersten Entwürfen beschrieben wird. Anschließend wird näher auf die Werkzeuge eingegangen, die zur Entwicklung des Produkts verwendet werden. Die letzten beiden Abschnitte befassen sich dann damit, welche Qualitätsansprüche an das fertige Produkt gestellt werden und wie durch Tests gezeigt werden kann, ob es diesen genügt.

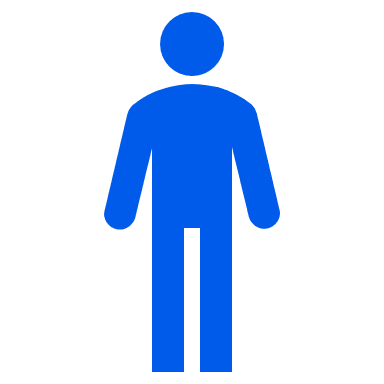
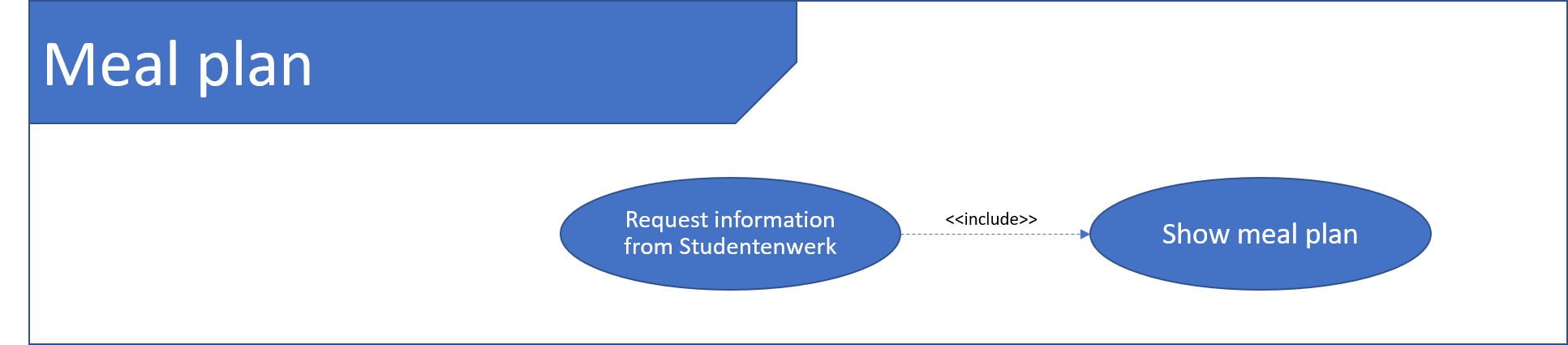
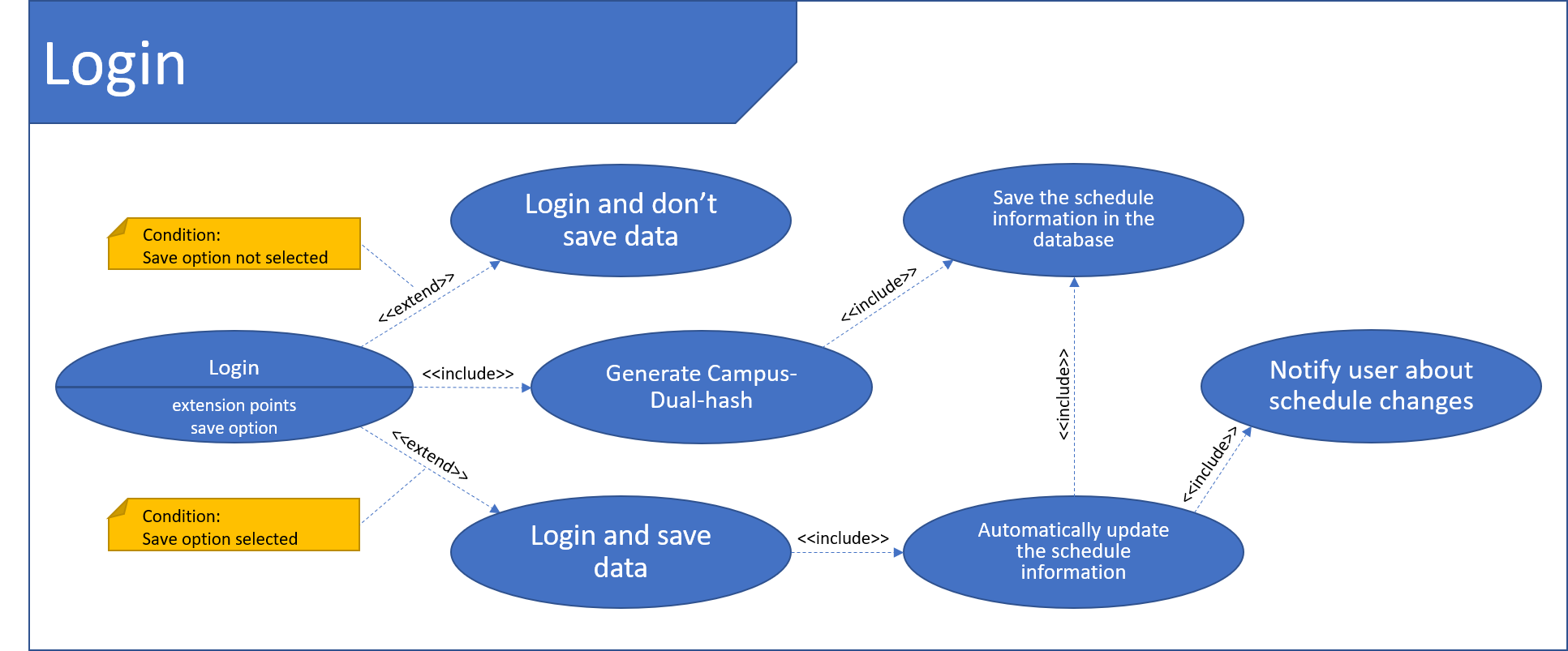
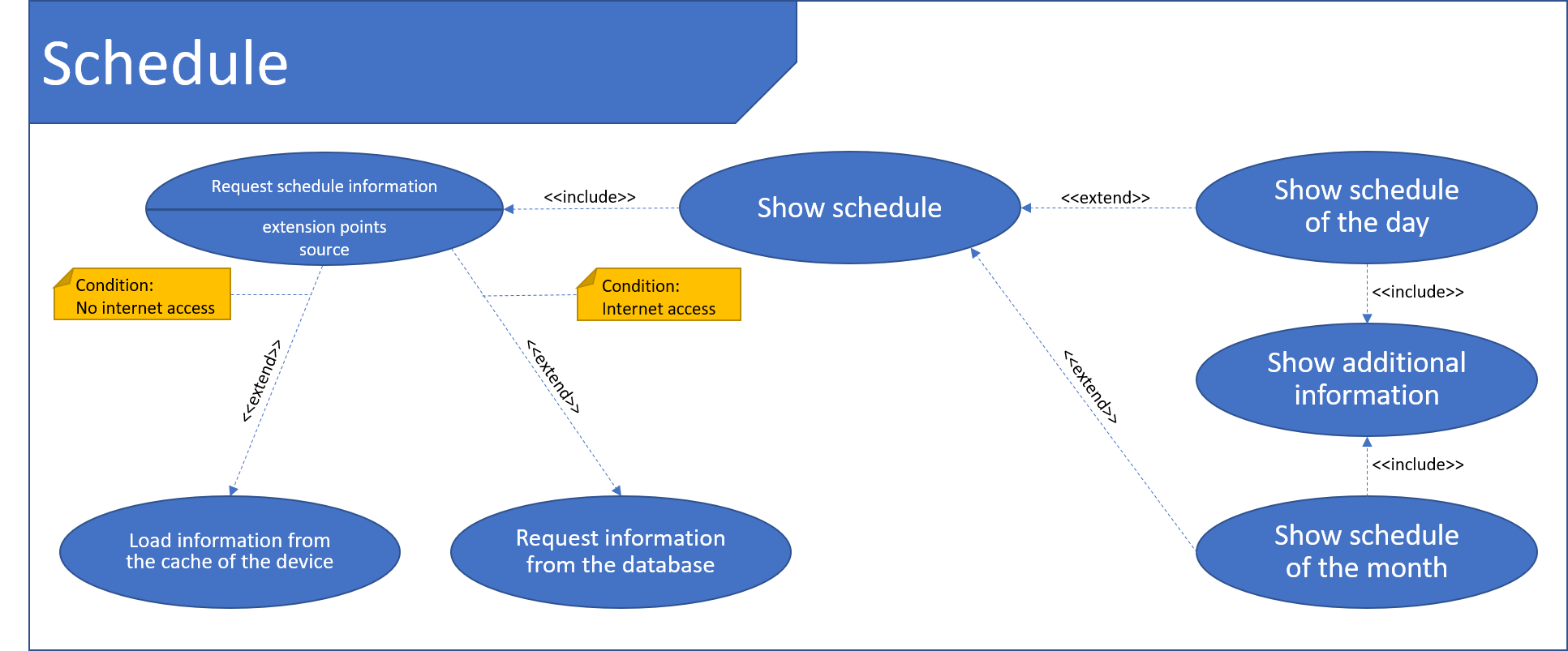
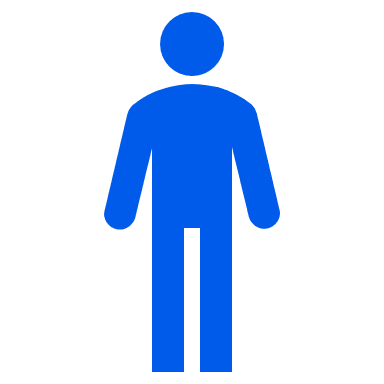
# Zielbestimmung

Das Ziel dieses Projekts ist die Neuentwicklung einer Anwendung, die es den Studenten der BA Sachsen ermöglicht ihren Stundenplan schnell und übersichtlich einsehen zu können. Die Anwendung soll dabei möglichst Plattformunabhängig sein, so dass der Stundenplan auf jedem Gerät eingesehen werden kann.

## Musskriterien

* Benutzer
  + Anmeldung mit Campus-Dual-Anmeldedaten
  + Löschung der eigenen Campus-Dual-Anmeldedaten aus dem System
  + Einsehen des eigenen Stundenplans
  + Übersicht mit Speiseplan der Mensa
  + Zugriff auf eine Übersicht mit Verlinkungen zu wichtigen Dokumenten und Webseiten
  + Ändern der Sprache
  + Manuelles Aktualisieren des Stundenplans
* Stundenplan
  + Anzeige eines Tages oder Monats
  + Zeitlich geordnete Ansicht der Module
  + Module mit Zusatzinformationen, wie Dozenten und Raum
  + Anzeigen von Prüfungen
* Administrator
  + Wartung und Sicherung der Datenbank
  + Konfigurieren der Betriebsparameter
* Backend
  + Tägliche Aktualisierung der Stundenplan-Daten
  + Speichern der Campus-Dual-Daten, sowie dem zugehörigen Hash-Wert
  + Verschlüsselte Ablage der Campus-Dual-Daten und des zugehörigen Hash-Werts
  + Nutzer bei Änderungen des Stundenplans benachrichtigen

In Abbildung 1‑1 wird erkennbar, dass das System grundsätzlich zwei Akteure kennt, zum einen den Benutzer und den Administrator. In diesem Use-Case-Diagramm wird noch einmal veranschaulicht, welche Anwendungsfälle das System bietet und von wem diese genutzt werden.



Benutzer

Administrator



Abbildung 1‑1 Musskriterien in Form von Use-Cases

## Wunschkriterien

* Erstellung individueller Termine durch den Nutzer
* Einladung anderer Nutzer zu einem individuell erstellten Termin (z.B. für Lerngruppen)
* Verlinkung des Materialverzeichnis eines Dozenten zu dem jeweiligen Modul
* Freigabe des Stundenplans für nicht angemeldete Nutzer
* Notifikation für vorhandene Prüfungsanmeldungen

## Abgrenzungskriterien

* Auflistung der Teilnehmer eines Moduls
* Nutzung aller sonstigen Campus-Dual-Funktionalitäten abseits des Kalenders (Anmeldung zu Prüfungen, Noten, etc.)
* Manuelles Ändern von Kalenderdaten

# Produktübersicht

## Produkteinsatz

### Anwendungsbereiche

Studenten der BA Sachsen können den Stundenplan derzeit nur über das Campus-Dual ansehen, was Angesicht der Wartezeiten dieses Portals sehr unkomfortabel ist. Zusätzlich verfügt das Campus-Dual nur über eine sporadisch umgesetzte Mobileansicht, welche die Daten nur teilweise oder gar nicht anzeigt. Andere Möglichkeiten, wie die Stundenplan-App, existieren zwar, aber diese zeigen nicht den aktuellen Stand des Stundenplans an und bieten somit keine Alternative.

Die genannten Probleme sollen mit Hilfe von Plan4BA gelöst werden. Damit soll es Studenten möglich sein den Stundenplan jederzeit schnell und komfortabel einsehen zu können.

### Zielgruppe

Plan4BA ist für alle Studenten gedacht, die den Kalender des Campus-Dual als Stundenplan nutzen. Damit ist die Voraussetzung für die Nutzung, dass Studenten gültige Anmeldedaten für das Campus-Dual besitzen.

### Betriebsbedingungen

Die Betriebsbedingungen dieses Systems sollen sich im Allgemeinen nicht von anderen Webanwendungen oder Internetdiensten unterscheiden. Das System soll den Nutzern jeder Zeit zur Verfügung stehen und soll dabei wartungsfrei arbeiten. Die Aktualisierung des Stundenplans soll einmal täglich automatisch stattfinden. Die Sicherung der Datenbank im Gegensatz dazu soll nicht automatisch geschehen, sondern muss vom Administrator durchgeführt werden.

## Produktfunktionen

### Funktionale Anforderung

**/F100/ Benutzerverwaltung**

/F110/ Anmeldung des Nutzers mit Hilfe seiner Campus-Dual-Anmeldedaten

/F120/ Generierung des Campus-Dual-Hashwerts aus den eingegeben Anmeldedaten

/F130/ Extraktion der Nutzerdaten aus Campus-Dual

/F140/ Anmeldung mit Speicherung der Campus-Dual-Anmeldedaten in der Datenbank

/F150/ Anmeldung ohne Speicherung der Campus-Dual-Anmeldedaten

/F160/ Generierung eines JSON-Web-Tokens (RFC 7519) zur Authentifizierung eines Geräts

/F170/ Löschen der gespeicherten Campus-Dual-Daten aus der Datenbank

/F180/ Hinzufügen von Anmeldedaten

/F190/ Ändern der Anzeigesprache

**/F200/ Stundenplan**

/F210/ Laden der Kalenderdaten aus dem Campus-Dual

/F220/ Speicherung der Kalenderdaten in der Datenbank

/F230/ Anzeige des Stundenplans

/F240/ Anzeige von Vorlesungsinformationen (Raum, Zeit und Dozent)

/F250/ Manuelle Aktualisierung der Stundenplandaten

/F260/ Stundenplandaten externen Kalenderanwendungen zur Verfügung stellen

/F270/ Prüfungstermine anzeigen

**/F300/ Mensaplan**

/F310/ Abfragen der Mensadaten über die Studentenwerk-API

/F320/ Anzeige der Mensadaten

**/F400/ Backend**

/F410/ Automatische Aktualisierung der Stundenplandaten mit Hilfe der gespeicherten Nutzerdaten

/F420/ Benachrichtigung des Nutzers bei Stundenplanänderungen

### Nichtfunktionale Anforderungen

* /F410/ soll einmal täglich in der Nacht stattfinden
* Anzahl der Anfragen an das Campus-Dual geringhalten
* Nutzer deutlich machen, dass Wartezeiten durch Anfragen an das Campus-Dual entstehen
* /F230/ Anzeige mit verschieden Perspektiven (täglich, monatlich)
* /F320/ Anzeige der Mensadaten nach Gerichten unterteilt und vegetarische/ vegane Gerichte markieren
* Nach erster Anmeldung sind immer Stundenplandaten ohne Wartezeiten vorhanden
* /F210/ Benachrichtigungen über Stundenplanaktlisierung
* Darstellung siehe 2.6 Benutzeroberfläche
* Zuordnung einer individuellen Farbe zu jedem Modul im Stundenplan

## Produktumgebung

Das System soll nahezu keinen Nutzer auf Grund von Inkompatibilitäten ausschließen. Deshalb soll das System in Form einer Webanwendung möglichst plattformunabhängig arbeiten, so dass es sowohl auf mobilen Geräten, als auch auf Desktopcomputern verwendet werden kann. Trotz dessen müssen die folgenden Anforderungen hinsichtlich Software (Abschnitt 2.3.1), Hardware (Abschnitt 2.3.2) und Orgware (Abschnitt 2.3.3) erfüllt werden, damit das System funktionsfähig ist.

### Software

* Client
  + Webbrowser (textbasierte oder offline Browser ausgeschlossen)
* Server
  + Linux oder Windows mit Hyper-V Unterstützung
  + Containervirtualsierungssoftware Docker mit docker-compose
  + PostgreSQL – oder MySQL - Datenbank

### Hardware

* Client
  + Internetfähiges Gerät
* Server
  + Internetfähiger Server

### Orgware

Als Orgware, also als sonstige Voraussetzungen, muss eine permanente Internetverbindung auf der Server-Seite gewährleistet werden können. Außerdem muss der Administrator den Dienst starten und die damit verbunden die Betriebsparameter konfigurieren.

## Produktdaten

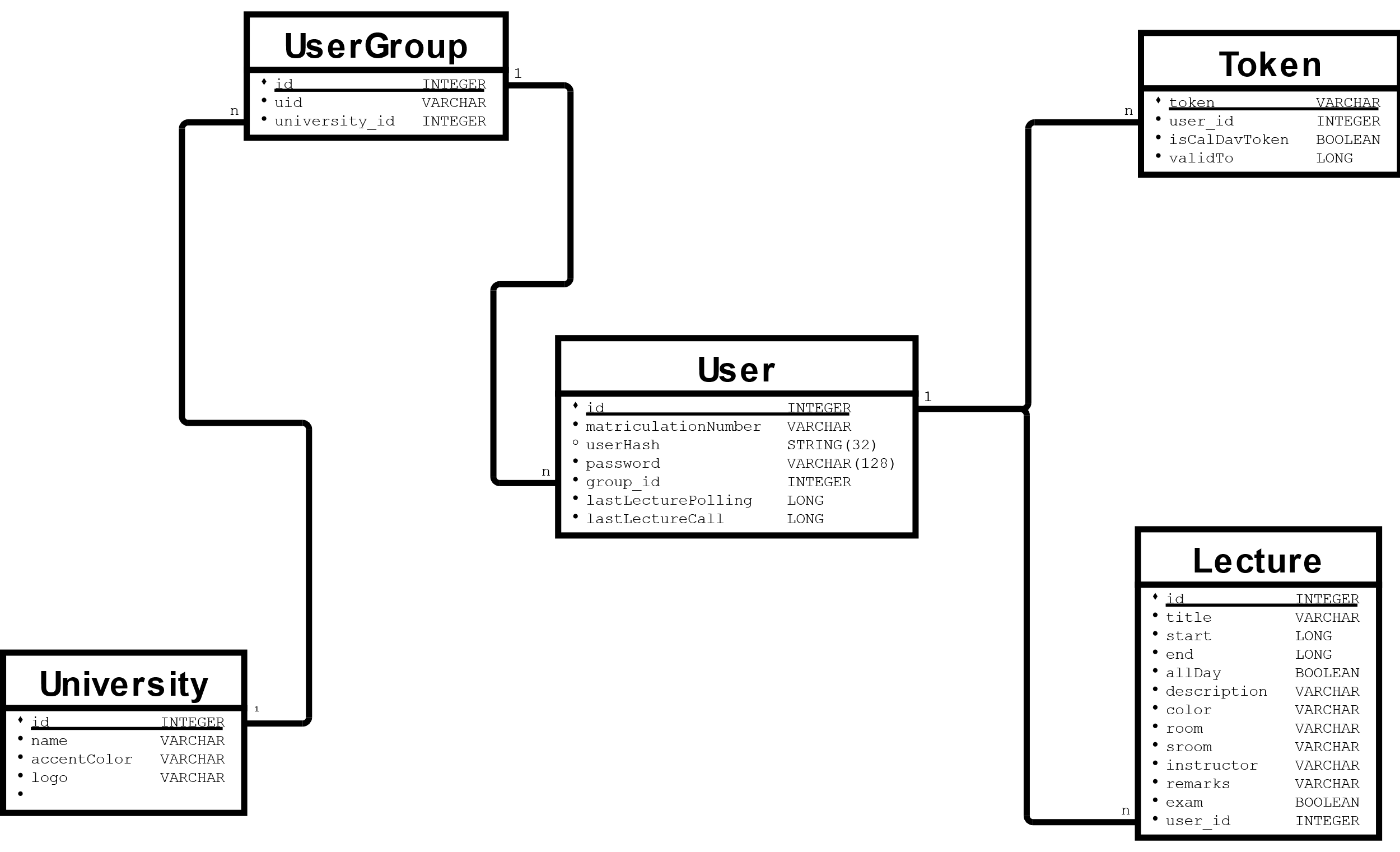


Abbildung 2‑1 Datenbankmodell in Form eines Entity-Relationship-Diagramms

**/D100/ Benutzerdaten**

/D110/ Matrikelnummer

/D120/ Campus-Dual-Hashwert

/D130/ Passwort

/D140/ Kurs

/D150/ Universität

**/D200/ Universitätsdaten**

/D210/ Name

/D220/ Logo

/D230/ Farbschema

**/D300/ Vorlesungsdaten**

/D310/ Bezeichnung

/D320/ Beginn

/D330/ Ende

/D340/ Ganztägig

/D350/ Beschreibung

/D360/ Raum

/D370/ Dozent

/D380/ Prüfung

## Produktbeschreibung

Der grundsätzliche Ablauf nach der Anmeldung eines Nutzers wird in der Abbildung 2‑2 gezeigt. Dabei werden speziell die Funktionen, die im Backend stattfinden, gezeigt.

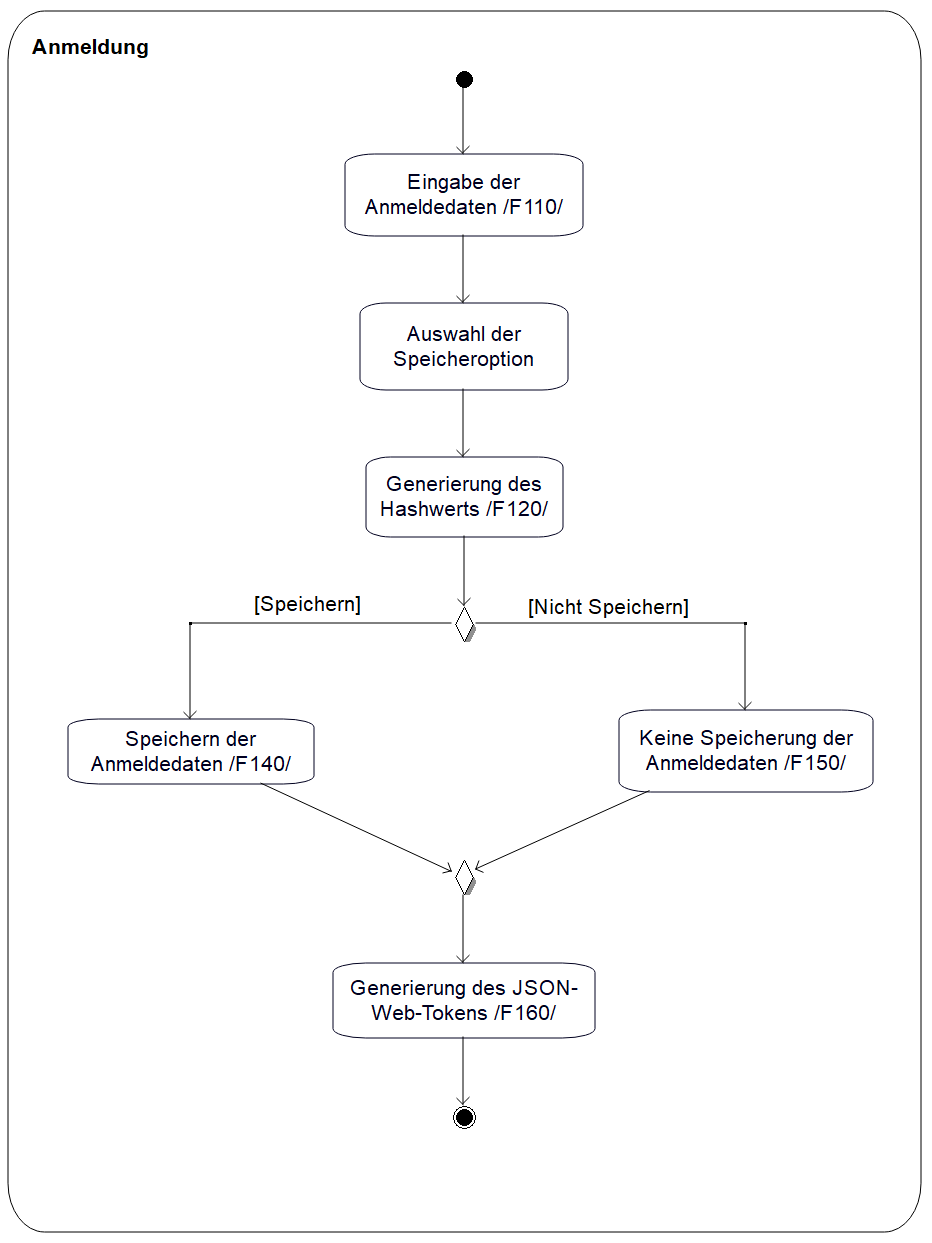
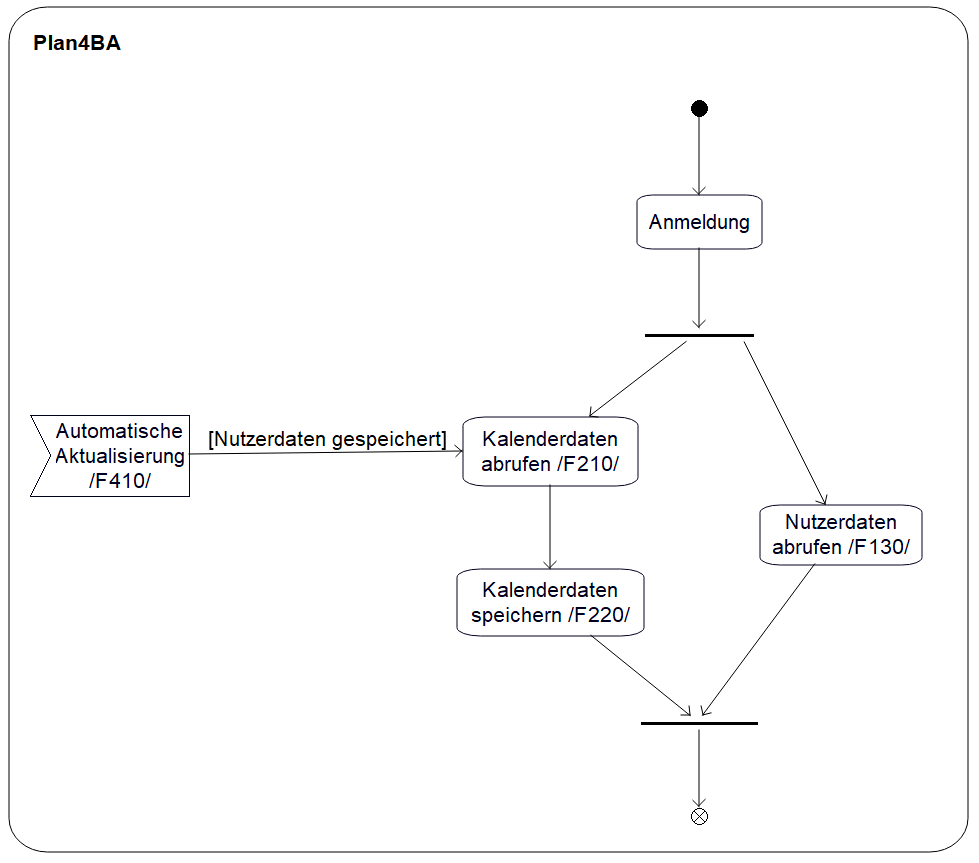


Abbildung 2‑2 Ablauf der Funktionen

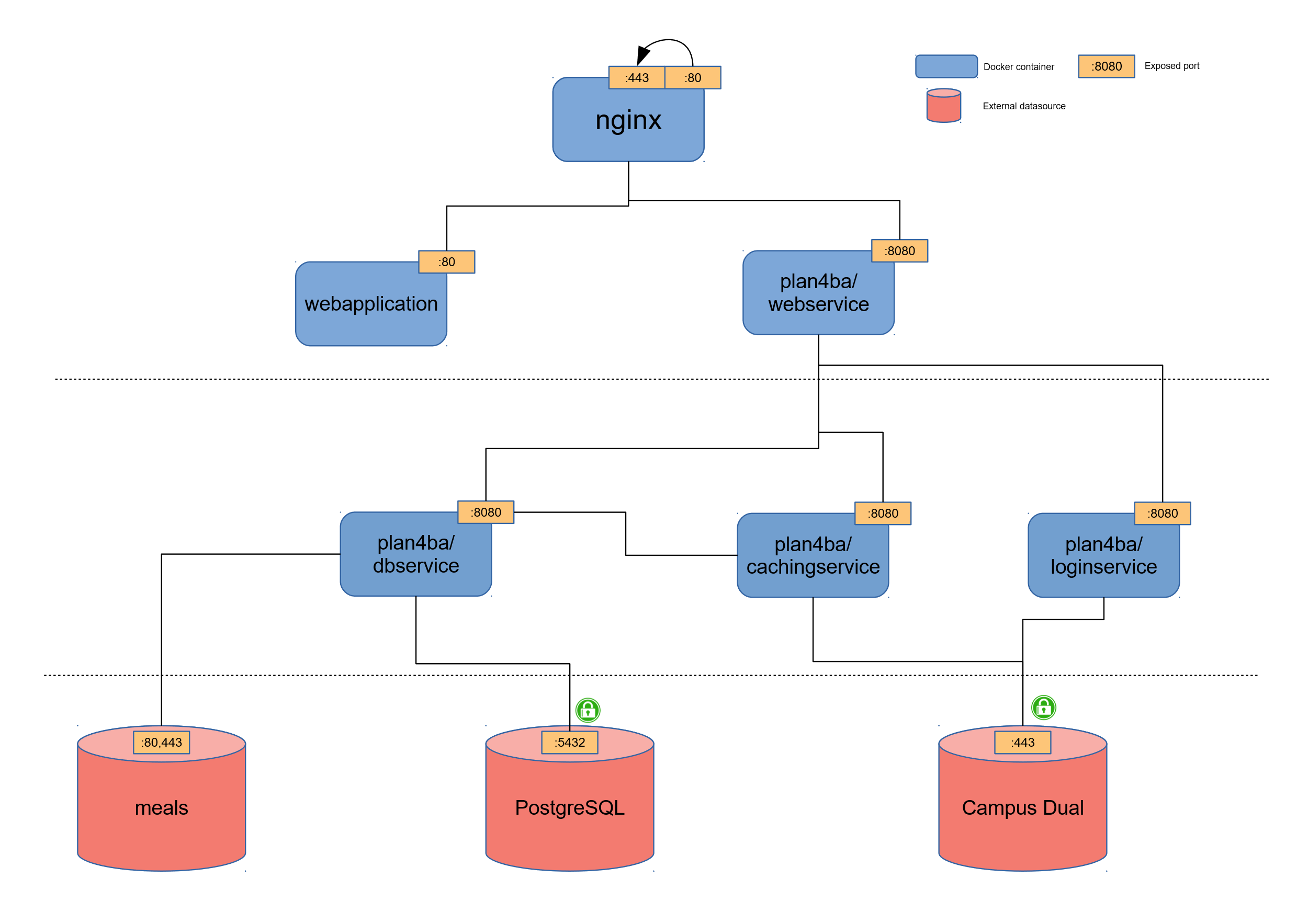
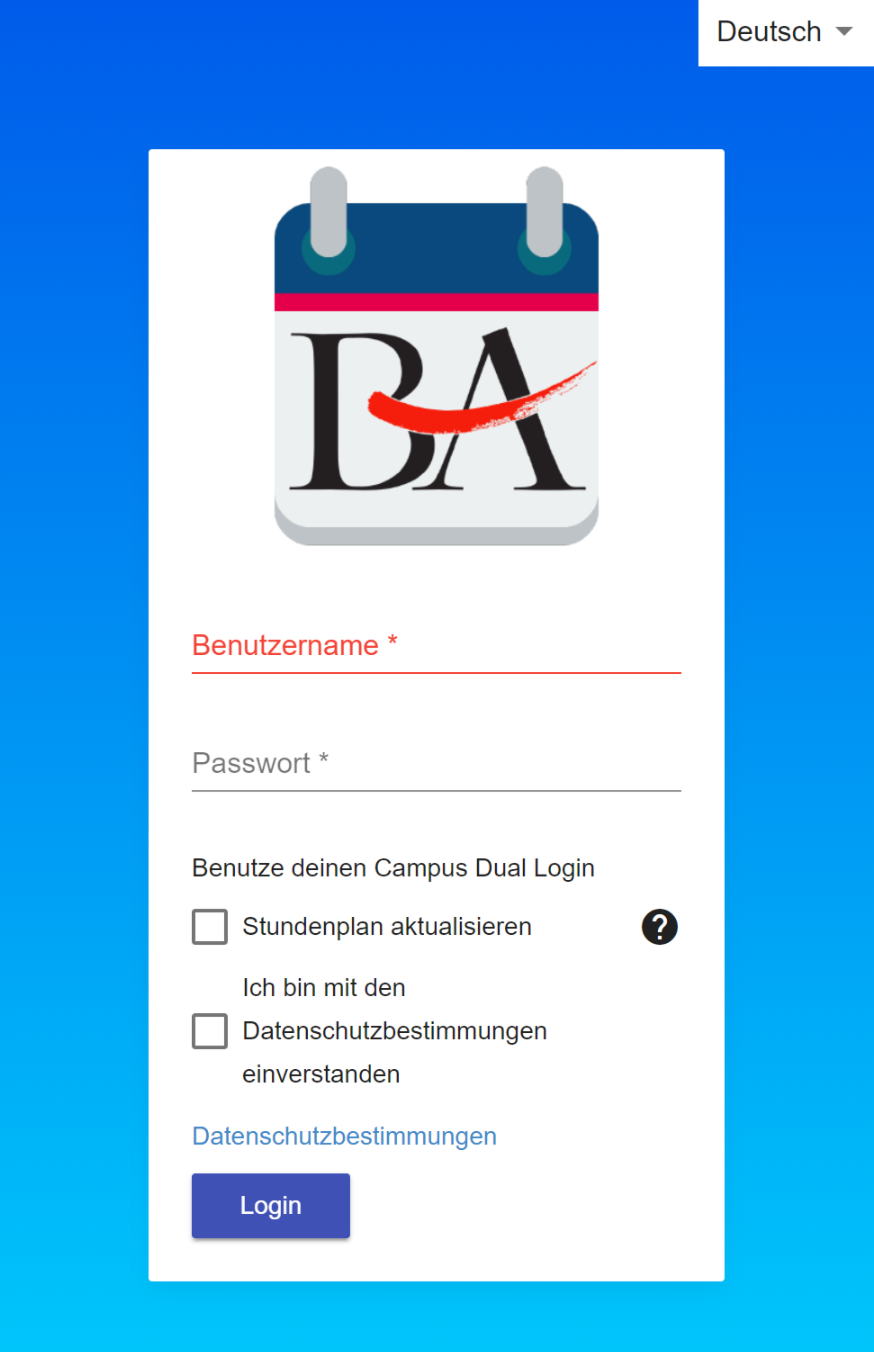
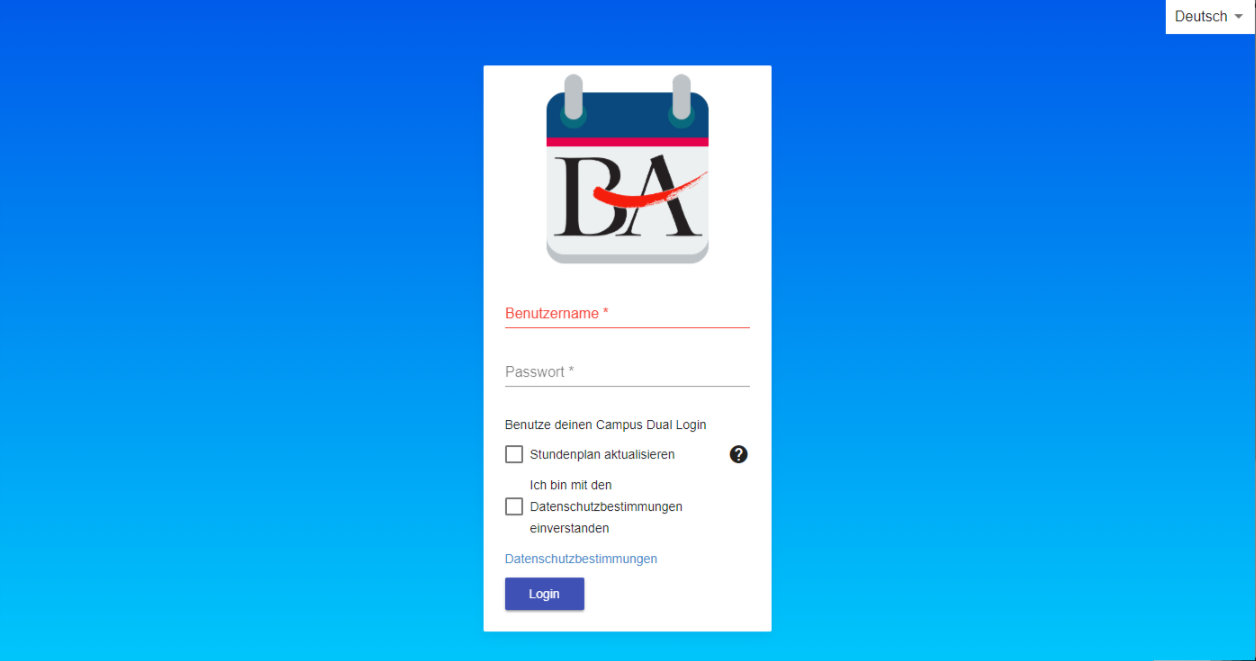


Abbildung 2‑3 Architektur der Docker Container

## Benutzeroberfläche

Ruft ein Nutzer die Applikation das erste Mal auf, so wird als erstes die Anmeldeseite erscheinen (Abbildung 2‑4). Auf dieser Seite hat er die Möglichkeit sich mit seinen Campus-Dual-Anmeldedaten zu registrieren (/F110/). Dem Nutzer stehen bei der Anmeldung zwei Optionen zur Verfügung, zum einen kann er seine Daten in der Datenbank speichern lassen (/F140/) oder er meldet sich an, ohne die Daten speichern zu lassen (/F150/). Durch Bestätigung des Anmelde-Knopfes werden anschließend die Funktionen /F120/, /F130/ und /F160/ aufgerufen. Bei der Anmeldung kann es zu Verzögerungen kommen, die durch die Wartezeiten auf das Campus-Dual verursacht werden.



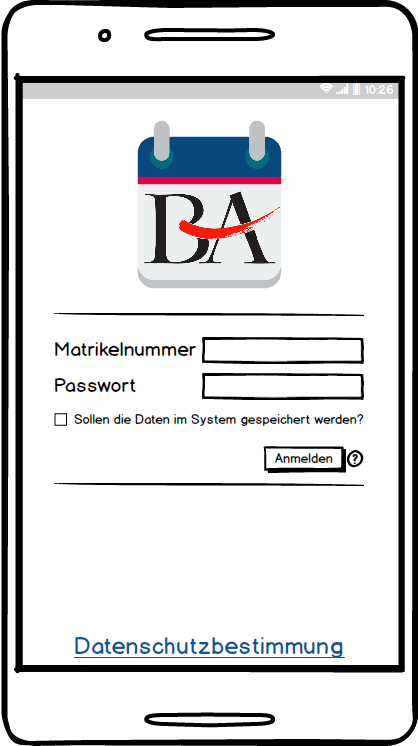
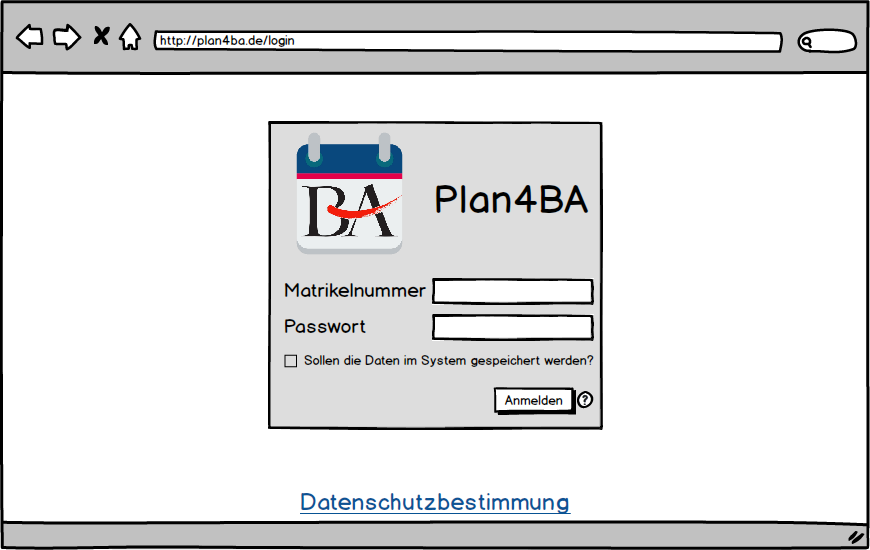


Abbildung 2‑4 Entwurf der verschiedenen Anmeldeseiten

Nachdem der Nutzer sich erfolgreich angemeldet hat und alle nötigen Daten abgefragt wurden (/F210/ und /F310/), wird dem Nutzer seine Startseite (Abbildung 2‑5) angezeigt werden. Auf dieser werden ihm kompakt alle Informationen des aktuellen Tages angezeigt. Dementsprechend wird der Stundenplan und Speiseplan dieses Tages angezeigt (/F230/, /F240/ und /F320/). Durch einen Klick auf einen dieser beiden Bereiche gelangt der Nutzer direkt zu den ausführlicheren Ansichten, wo er auch die Informationen anderer Tage einsehen kann.

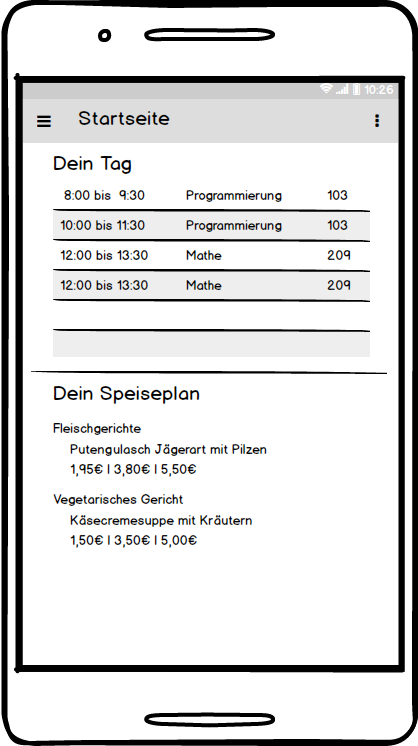
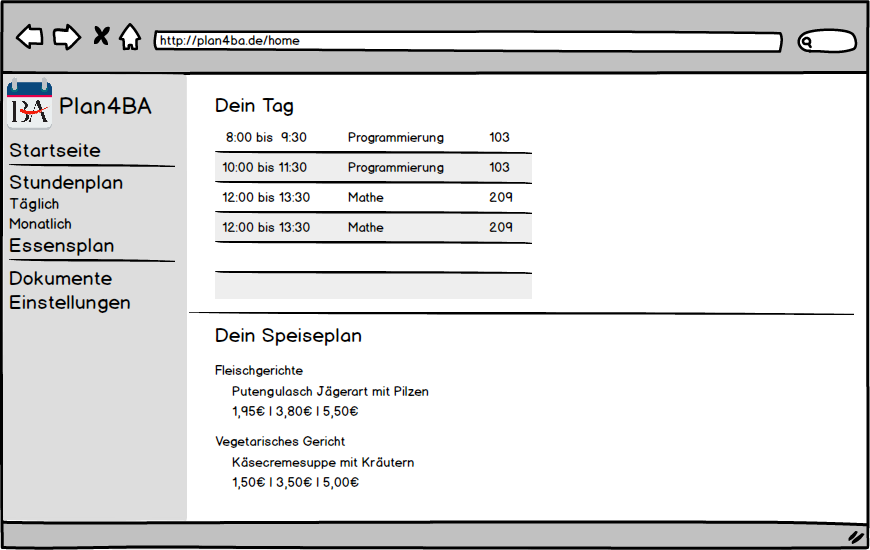
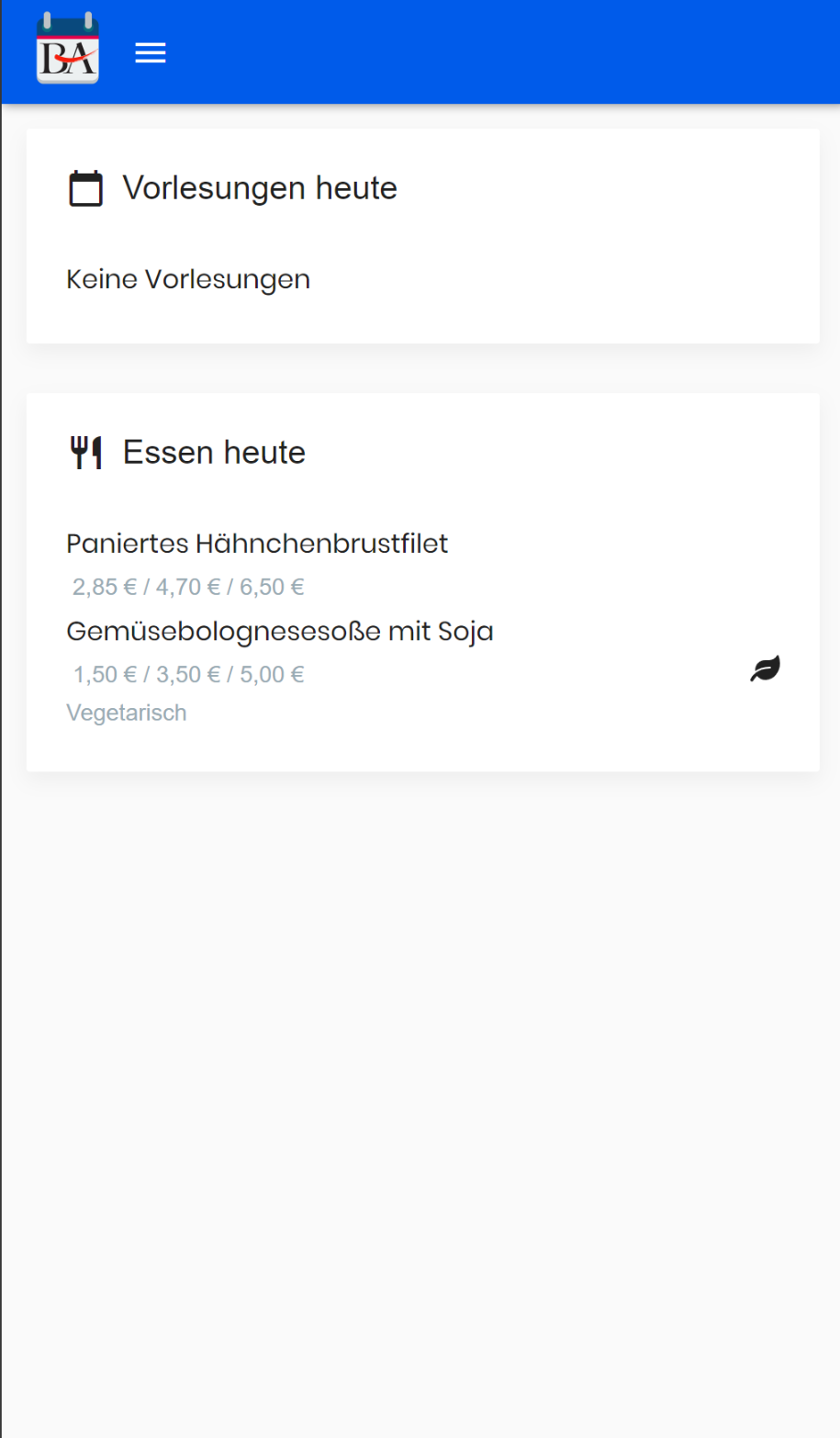
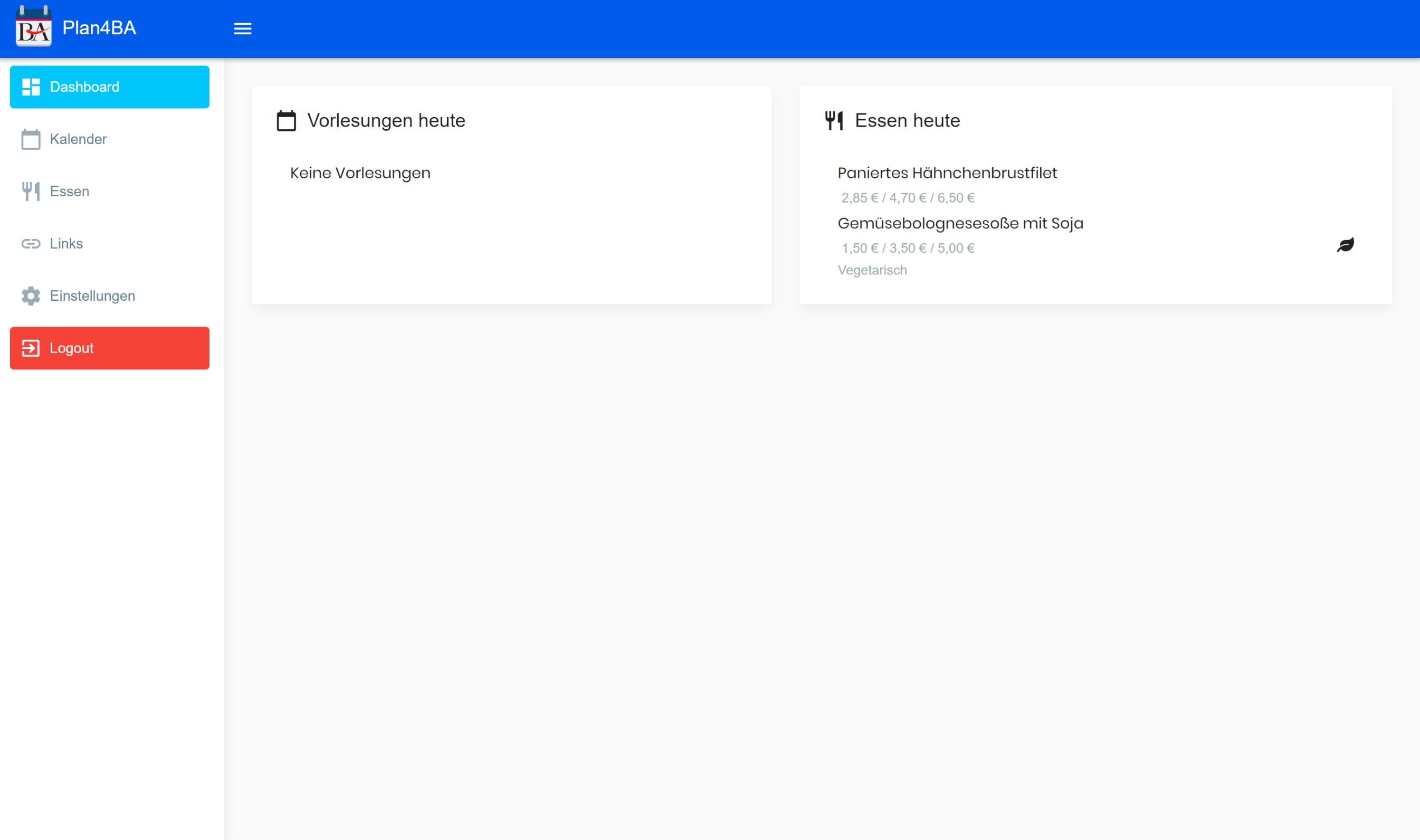


Abbildung 2‑5 Entwurf der verschiedenen Startseiten

Die wichtigste Navigationsmöglichkeit der Anwendung ist das Menü. Bei der Desktop-Anwendung wird das Menü, auf Grund der horizontalen Orientierung, dauerhaft angezeigt (Abbildung 2‑5). Im Gegensatz dazu muss das Menü bei der mobilen Variante über das Symbol in der oberen linken Ecke aufgerufen werden (Abbildung 2‑6). Das Menü bietet die Möglichkeit auf die Startseite zurück zu kehren, den Stundenplan in einer täglichen oder monatlichen Ansicht zu öffnen und den Essensplan anzuzeigen. Zusätzlich können über dieses die Einstellungen aufgerufen werden, wo der Nutzer zum Beispiel seine Daten löschen lassen kann (/F170/). Eine weitere Ansicht, die Verlinkungen zu wichtigen Seiten und Dokumenten bietet, kann über den Menüpunkt „Dokumente“ erreicht werden.

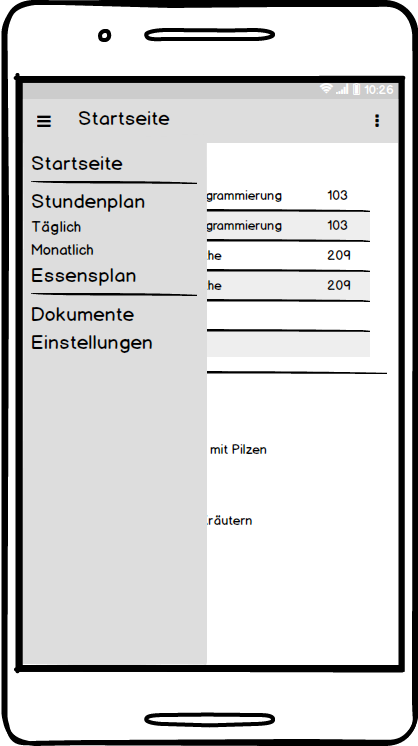
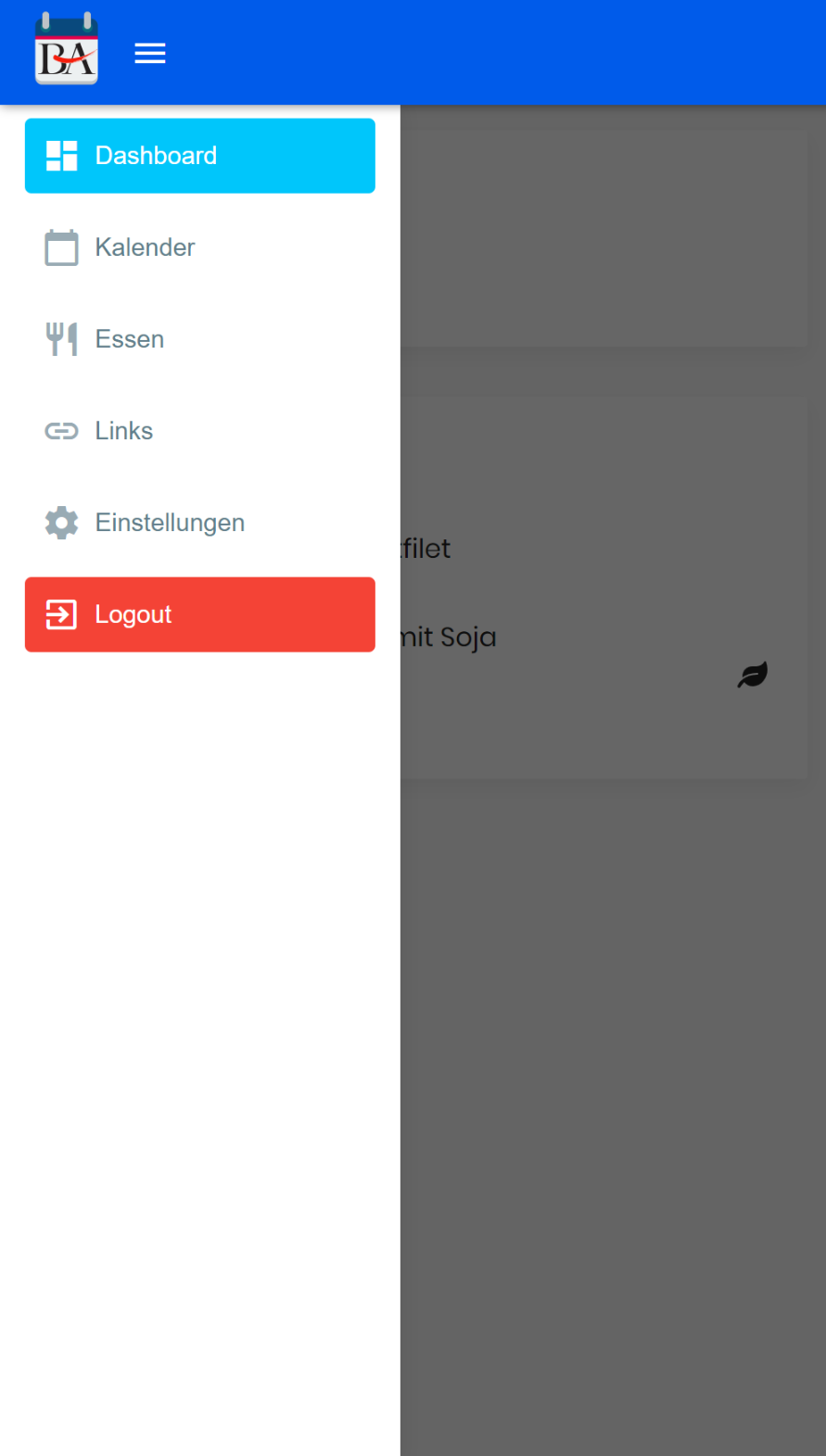


Abbildung 2‑6 Entwurf der Menüführung

# Entwicklungsumgebung

### Software

* IDEA IntelliJ von Jetbrains
* Docker
* Angular CLI
* Nativescript CLI
* Android SDK
* XCode
* OpenJDK 10/11, Kotlin-Kompiler > 1.2.70(auch 1.3 bei Release möglich)
* PostgreSQL
* Balsamiq Mockups 3

### Hardware

* App-Entwicklung
  + Gerät mit MacOS
  + Gerät mit IOS
  + Gerät mit Android
* Backend-Entwicklung
  + Beliebiger Computer mit einem gängigen Betriebssystem (Windows, Linux, MacOS)

### Orgware

* Kommunikation
  + Instant-Messaging-Dienst
* Versionsverwaltung und Canban
  + Github
* Dokumentation
  + UML
  + Entity Relationship Diagramm
  + OpenAPI Specification

# Qualitätsanforderungen

In der folgenden Tabelle 4‑1 wird gezeigt welche Qualitätsanforderungen an das System gestellt werden. Bei den Qualitätsanforderungen wurde sich an der ISO 9126 orientiert, wobei nur die wichtigsten Punkte zur Bewertung ausgewählt wurden.

*Funktionalität:* Die Hauptfunktionen sollen zuverlässig und wie erwartet arbeiten.

*Zuverlässigkeit:* Die Anwendung soll immer zur Verfügung stehen und Wartezeiten durch Fehler sollen vermieden werden.

*Benutzbarkeit:* Die Bedienung der Software soll sowohl mit Maus und Tastatur, als auch mit einem Touchscreen bedienbar sein. Außerdem soll die Bedienung intuitiv und einfach gestaltet werden.

*Effizienz:* Das Programm soll deutlich geringere Wartezeiten als das Campus-Dual aufweisen und in kurzer Zeit den Stundenplan anzeigen.

*Übertragbarkeit:* Die Anwendung soll sowohl auf Desktop-Computern mit unterschiedlichen Betriebssystemen, als auch mit mobilen Geräten verwendet werden können.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Sehr wichtig | Wichtig | Neutral | Weniger wichtig | unwichtig |
| **Funktionalität** | | | | | |
| Richtigkeit |  |  | X |  |  |
| Sicherheit |  | X |  |  |  |
| **Zuverlässigkeit** | | | | | |
| Reife |  |  | X |  |  |
| Wiederherstellbarkeit |  |  |  | X |  |
| **Benutzbarkeit** | | | | | |
| Verständlichkeit |  |  | X |  |  |
| Erlernbarkeit |  |  | X |  |  |
| Bedienbarkeit | X |  |  |  |  |
| **Effizienz** | | | | | |
| Zeitverhalten |  | X |  |  |  |
| **Übertragbarkeit** | | | | | |
| Anpassbarkeit |  |  | X |  |  |
| Portierbarkeit/ Kompatibilität |  | X |  |  |  |

Tabelle 4‑1 Qualitätsanforderungen

# Testfälle und Testszenarien

/T100/ Anmeldung mit validen Campus-Dual-Anmeldedaten

/T200/ Beta-Phase der Anwendung (Kommilitonen nutzen die Anwendung)

* + - Erfolgreich, wenn die Tester mit der Nutzererfahrung zufrieden sind

/T300/ Nach Löschung der Nutzerdaten ist eine Neuanmeldung notwendig

/T400/ Speiseplan ist identisch zu der Webseite

/T500/ Stundenplan ist identisch zu Campus-Dual-Kalender

/T600/ Abfrage der Stundenplandaten startet einmal täglich automatisch und reagiert auf Änderungen

/T610/ Es gibt eine Benachrichtigung über Änderungen

# Verzeichnisse

## Versionsverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Status | Kommentar |
| 0.1 | 10.10.2018 | Entwurf | Entwurf des Pflichtenhefts |
| 0.2 | 11.10.2018 | Entwurf | Festlegung der Gliederung |
| 0.3 | 11.10.2018 | Entwurf | Ausarbeitung der Zielbestimmung |
| 0.4 | 14.10.2018 | Entwurf | Festlegung der Qualitätskriterien |
| 0.5 | 15.10.2018 | Entwurf | Ergänzung der Entwicklungsumgebung |
| 0.6 | 16.10.2018 | Entwurf | Ausarbeitung von Testfällen und der Produktübersicht |
| 0.7 | 17.10.2018 | Entwurf | Ergänzung von Entwürfen der Benutzeroberfläche |
| 0.8 | 17.10.2018 | Entwurf | Ergänzung von UML-Diagrammen |
| 0.9 | 18.10.2018 | Entwurf | Hinzufügen von Docker-Architektur |
| 1.0 | 19.10.2018 | Fertiger Entwurf | Letzte Korrekturen und Anpassungen |
| 1.1 | 12.11.2018 | Fertiger Entwurf | Hinzufügen von Muss- und Wunschkriterien |
| 1.2 | 15.11.2018 | Fertiger Entwurf | Ergänzen der Funktionalen Anforderungen |
| 1.4 | 13.12.2018 | Finaler Entwurf | Aktualisierung der Abbildungen |

## Abbildungsverzeichnis

[Abbildung 1‑1 Musskriterien in Form von Use-Cases 5](#_Toc532462882)

[Abbildung 2‑1 Datenbankmodell in Form eines Entity-Relationship-Diagramms 9](#_Toc532462883)

[Abbildung 2‑2 Ablauf der Funktionen 10](#_Toc532462884)

[Abbildung 2‑3 Architektur der Docker Container 11](#_Toc532462885)

[Abbildung 2‑4 Entwurf der verschiedenen Anmeldeseiten 12](https://d.docs.live.net/7d1f0f105e207898/Dokumente/BA/Semester%205/Software_Project/SE-Projekt.docx#_Toc532462886)

[Abbildung 2‑5 Entwurf der verschiedenen Startseiten 12](https://d.docs.live.net/7d1f0f105e207898/Dokumente/BA/Semester%205/Software_Project/SE-Projekt.docx#_Toc532462887)

[Abbildung 2‑6 Entwurf der Menüführung 13](#_Toc532462888)

## Tabellenverzeichnis

[Tabelle 4‑1 Qualitätsanforderungen 15](#_Toc532462889)