SEP Projektmappe

**Projektmappe des Projektes**

<Kicktipp 2.0>

**Dokumentation des Projektes**

Gruppe G:

<Alexander Grebennikov>

<Tom Schneider>

<Felix Farke>

Inhalt

[Projektbeschreibung 4](#_Toc115257124)

[Zyklus I 5](#_Toc115257125)

[Spezifikationsplanung](#_Toc115257126)…………………………………………………………………………………………………………………….5

User Stories…………………………………………………………………………………………………………………………………….6

[Papierprototypen…………………………………………………………………………………………………………………………](#_Toc115257128)..9

Klassendiagramm………………………………………………………………………………………………………………………….11

Komponentendiagramm……………………………………………………………………………………………………………….12

Routing…………………………………………………………………………………………………………………………………………13

Verhaltensdiagramme…………………………………………………………………………………………………………………..14

[Zyklus II 11](#_Toc115257133)

[Spezifikationsplanung 11](#_Toc115257134)

[User-Stories 11](#_Toc115257135)

[Papierprototypen 12](#_Toc115257136)

[Strukturdiagramme (Komponenten- und Klassendiagramme) 12](#_Toc115257137)

[Verhaltensdiagramme (Kommunikationsdiagramme) 12](#_Toc115257138)

[Funktionalitätsplanung 13](#_Toc115257139)

[Modultests 14](#_Toc115257140)

[Systemtests 14](#_Toc115257141)

[Zyklus III 15](#_Toc115257142)

[Spezifikationsplanung 15](#_Toc115257143)

[User-Stories 15](#_Toc115257144)

[Papierprototypen 16](#_Toc115257145)

[Strukturdiagramme (Komponenten- und Klassendiagramme) 16](#_Toc115257146)

[Verhaltensdiagramme (Kommunikationsdiagramme) 16](#_Toc115257147)

[Funktionalitätsplanung 17](#_Toc115257148)

[Modultests 18](#_Toc115257149)

[Systemtests 18](#_Toc115257150)

[Nutzerhandbuch 19](#_Toc115257151)

[Technische Anforderungen 19](#_Toc115257152)

[Installationsanleitung 19](#_Toc115257153)

[Bedienungsanleitung 19](#_Toc115257154)

# Projektbeschreibung

**Zyklus 1**

***(Allgemeines):***

Im ersten Zyklus soll zunächst die Grundstruktur des Systems implementiert werden. Dem System soll eine Service-Architektur (bestehend aus mindestens einem Frontend-Service und einem BackendService) zu Grunde liegen. Sämtliche Daten des Systems sollen persistent gespeichert werden.

***(Registrierung eines Nutzers):***

Nutzer sollen die Möglichkeit haben sich zu registrieren und somit ein Profil zu erstellen. Ein Nutzerprofil soll aus dem Vor- und Nachnamen des Nutzers, einer gültigen E-Mailadresse, dem Geburtsdatum des Nutzers, einem Passwort und einem optionalen Profilbild bestehen. Ein Nutzer soll die Möglichkeit haben sich im System mittels seines Profils einzuloggen. Der Login-Vorgang soll durch eine Zwei-Faktor-Authentifizierung realisiert werden, bei der, neben dem Passwort, ein zufällig generierter Sicherheitscode eingegeben werden muss, der per E-Mail an den Nutzer versendet wird. Für die Abnahmen: Es soll einen „super“ Sicherheitscode existieren, der ebenfalls akzeptiert wird.

***(Registrierung der Systemadministratoren):***

Nutzer sollen die Möglichkeit haben, sich als Systemadministrator zu registrieren und somit ihr Profil zu erstellen. Diese Systemadministrator-Profile sollen aus dem Vor- und Nachnamen, einer gültigen EMailadresse und einem Passwort bestehen. Ein Nutzer soll die Möglichkeit haben sich im System mittels seines Systemadministrator-Profils einzuloggen. Der Login-Vorgang soll durch eine ZweiFaktor-Authentifizierung realisiert werden, bei der, neben dem Passwort, ein zufällig generierter Sicherheitscode eingegeben werden muss, der per E-Mail an den Nutzer versendet wird. Für die Abnahmen: Es soll ein „super“ Sicherheitscode existieren, der ebenfalls akzeptiert wird.

***(Ligen erstellen):***

Systemadministratoren sollen verschiedene Ligen anlegen können. Eine Liga besteht aus einem Namen, einem Spielplan (bestehend aus mehreren Spieltagen) und einem optionalen Liga-Bild. Ein Spieltag besteht aus mehreren einzelnen Spielen. In jedem Spiel treten zwei unterschiedliche Mannschaften (eine Heimmannschaft und eine Auswärtsmannschaft) gegeneinander an. Ein Systemadministrator soll die Möglichkeit haben den Spielplan mit Hilfe einer CSV-Datei1 einzulesen. Alle diese Liga-Daten (Liga, Spieltage, Spiele) sollen im Nachhinein für jeden Nutzer einsehbar sein. Ein Systemadministrator soll Liga-Daten manuell ändern können. 1 Als

***(Systemdatum ändern):***

Der Systemadministrator soll die Möglichkeit haben das Datum, nach dem sich das System richtet zu ändern. Das Systemdatum definiert welche Spiele aus dem Spielplan noch anstehen und welche schon beendet wurden.

Grundlage für die Spielpläne und die Ergebnisse sollen historische Daten genutzt werden: <https://github.com/footballcsv/deutschland>

# **Zyklus I**

## **Spezifikationsplanung**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** | Registrierung der Nutzer |  |  |  |
| 1.1 | User Story Nutzerregistrierung | User Story | Felix | Fertig |
| 1.2 | Papierprototyp Nutzerregistrierung | Papierprototyp | Felix | Fertig |
| **2.** | Login der Nutzer |  |  |  |
| 2.1 | User Story Login | User Story | Alex | Fertig |
| 2.2 | Login Fenster | Papierprototyp | Alex | Fertig |
| **3.** | Registrierung Admins |  |  |  |
| 3.1 | User Story Admin Registrierung | User Story | Felix | Fertig |
| 3.2 | Registrierungs-Fenster | Papierprototyp | Felix | Fertig |
| 3.3 | User Story Admin Login | User Story | Felix | Fertig |
| **4.** | Architektur |  |  |  |
| 4.1 | Klassendiagramm Backend | Klassendiagramm | Alex | Fertig |
| 4.2 | Komponentendiagramm Frontend | Komponentendiagramm | Alex | Fertig |
| 4.3 | Routing Frontend | Komponentendiagramm | Alex | Fertig |
| 4.4 | Kommunikationsdiagramm  Login | Kommunikationsdiagramm | Alex | Fertig |
| 4.5 | Kommunikationsdiagramm  Admin | Kommunikationsdiagramm | Alex | Fertig |
| 5 | Ligen erstellen |  |  |  |
| 5.1 | User Stories Liga erstellen | User Story | Tom | Fertig |
| 5.2 | User Story Liga bearbeiten | User Story | Tom | Fertig |
| 5.3 | Papierprototyp Liga erstellen | Papierprototyp | Tom | Fertig |
| 5.4 | Papierprototyp Liga bearbeiten | Papierprototyp | Tom | Fertig |
| 5.5 | User Stories CVS-Datei einlesen | User Story | Tom | Fertig |
| 5.6 | Papierprototyp CVS-Datei  Einlesen | Papierprototyp | Tom | Fertig |
| 5.7 | User Story Spiele manuell einfügen | User Story | Alex | Fertig |
| 5.8 | Papierprototyp Spiele manuell einfügen | Papierprototyp | Alex | Fertig |
| 6 | Systemdatum ändern |  |  |  |
| 6.1 | User Story Systemdatum ändern | User Story | Tom | Fertig |
| 6.2 | Papierprototyp Systemdatum ändern | Papierprototyp | Tom | Fertig |
| **7.** | Nutzer Funktionen |  |  |  |
| 7.1 | User Story Liga anzeigen | User Story | Alex | Fertig |
| 7.2 | Papierprototyp Liga anzeigen | Papierprototyp | Alex | Fertig |

## **User-Stories**

|  |  |
| --- | --- |
| **User Story-ID** | 1.1 |
| **User Story-Beschreibung** | Als Nutzer möchte ich, dass es möglich ist ein Profil zu registrieren, mit meinem Vornamen, Nachnamen, Emailadresse, Geburtsdatum, Passwort und optional mit einem Profilbild, damit meine Mitspieler auch gewisse Informationen über mich haben und meine Daten im Backend gespeichert werden. |
| **Geschätzter Realisierungsaufwand** | 1 Tage |
| **Priorität** | Hoch |
| **Autor** | Felix Farke |
| **Abhängigkeiten zu anderen User Stories** | 2.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 2.1 |
| **User Story-Beschreibung** | Als Nutzer möchte ich, dass ich in der Lage bin mich mit meinem Profil einzuloggen, damit ich die Liga angezeigt kriegen kann und meine Tipps abgeben/ändern kann. |
| **Geschätzter Realisierungsaufwand** | 1 Tage |
| **Priorität** | Hoch |
| **Autor** | Felix Farke |
| **Abhängigkeiten zu anderen User Stories** | 1.1, 3.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 3.1 |
| **User Story-Beschreibung** | Als Nutzer möchte ich, dass ich ein Systemadministratorprofil erstellen kann, damit ich die Rechte habe Liegen erstellen/bearbeiten zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tage |
| **Priorität** | Mittel |
| **Autor** | Felix Farke |
| **Abhängigkeiten zu anderen User Stories** | 2.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 3.3 |
| **User Story-Beschreibung** | Als Admin möchte ich, dass es mir möglich ist mich mit meinem Profil einzuloggen, um bestehende Ligen weiterhin verwalten zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tage |
| **Priorität** | Mittel |
| **Autor** | Felix Farke |
| **Abhängigkeiten zu anderen User Stories** | 2.1, 3.1. |

|  |  |
| --- | --- |
| **User Story-ID** | 5.1 |
| **User Story-Beschreibung** | Als Admin möchte ich, dass es mir möglich ist eine Liga mit einem Namen, einem Spielplan und optional mit einem Bild zu erstellen, um mir und meinen Mitspielern ein Spielerlebnis beschaffen zu können. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | Mittle |
| **Autor** | Tom Schneider |
| **Abhängigkeiten zu anderen User Stories** | 2.1, 3.1, 3.3 |

|  |  |
| --- | --- |
| **User Story-ID** | 5.2 |
| **User Story-Beschreibung** | Als Admin möchte ich, dass es mir möglich ist eine Liga und ihre Attribute zu bearbeiten, um Änderungen währen der laufenden Liga vornehmen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Mittle |
| **Autor** | Tom Schneider |
| **Abhängigkeiten zu anderen User Stories** | 2.1, 3.1, 3.3, 5.1 |

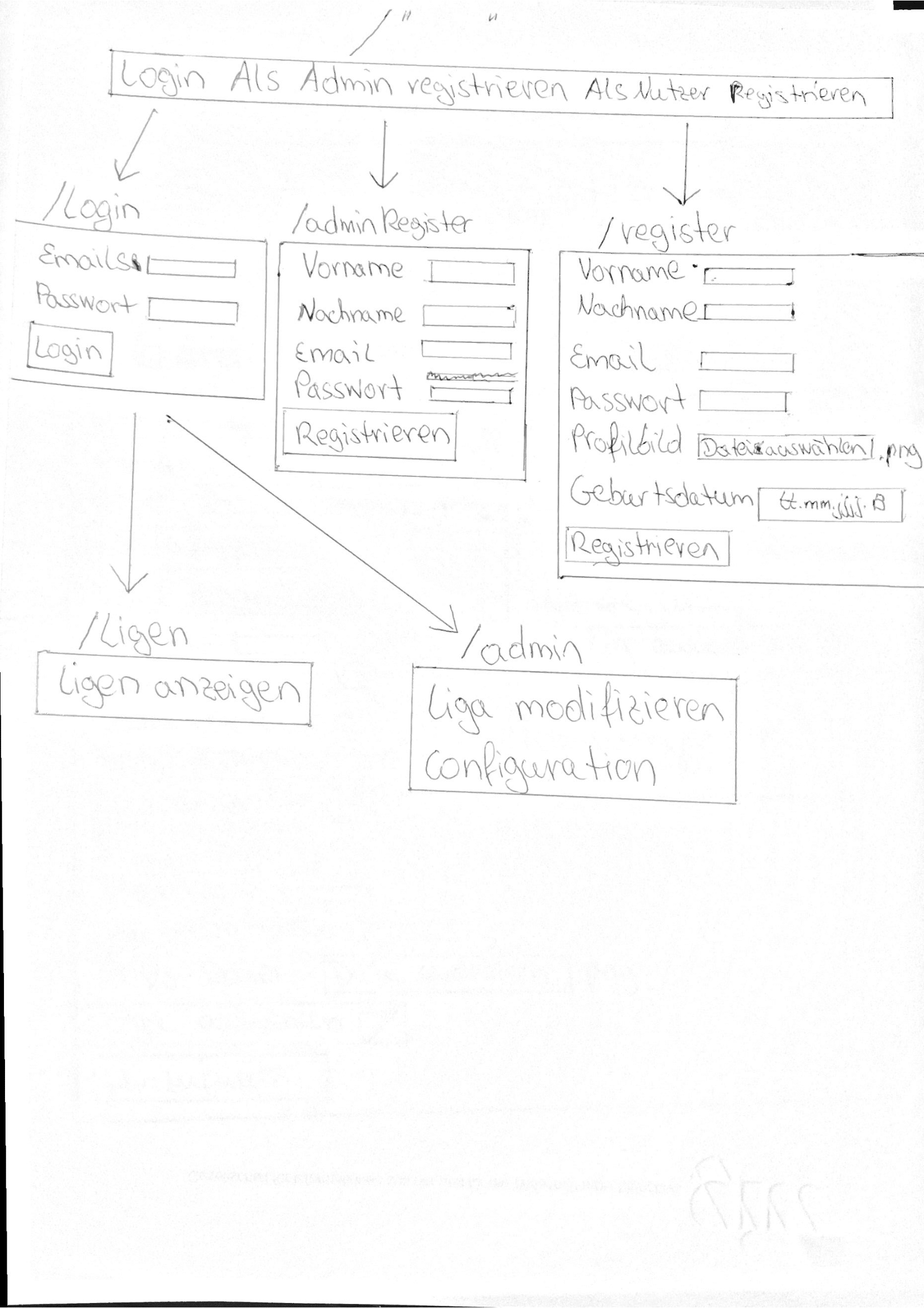
|  |  |
| --- | --- |
| **User Story-ID** | 5.5 |
| **User Story-Beschreibung** | Als Admin möchte ich, dass es mir möglich ist Liga Daten mithilfe einer CSV-Datei einzulesen, um nicht alle per Hand manuell einlesen zu müssen. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor** | Tom Schneider |
| **Abhängigkeiten zu anderen User Stories** | 2.1, 3.1, 3.3 |

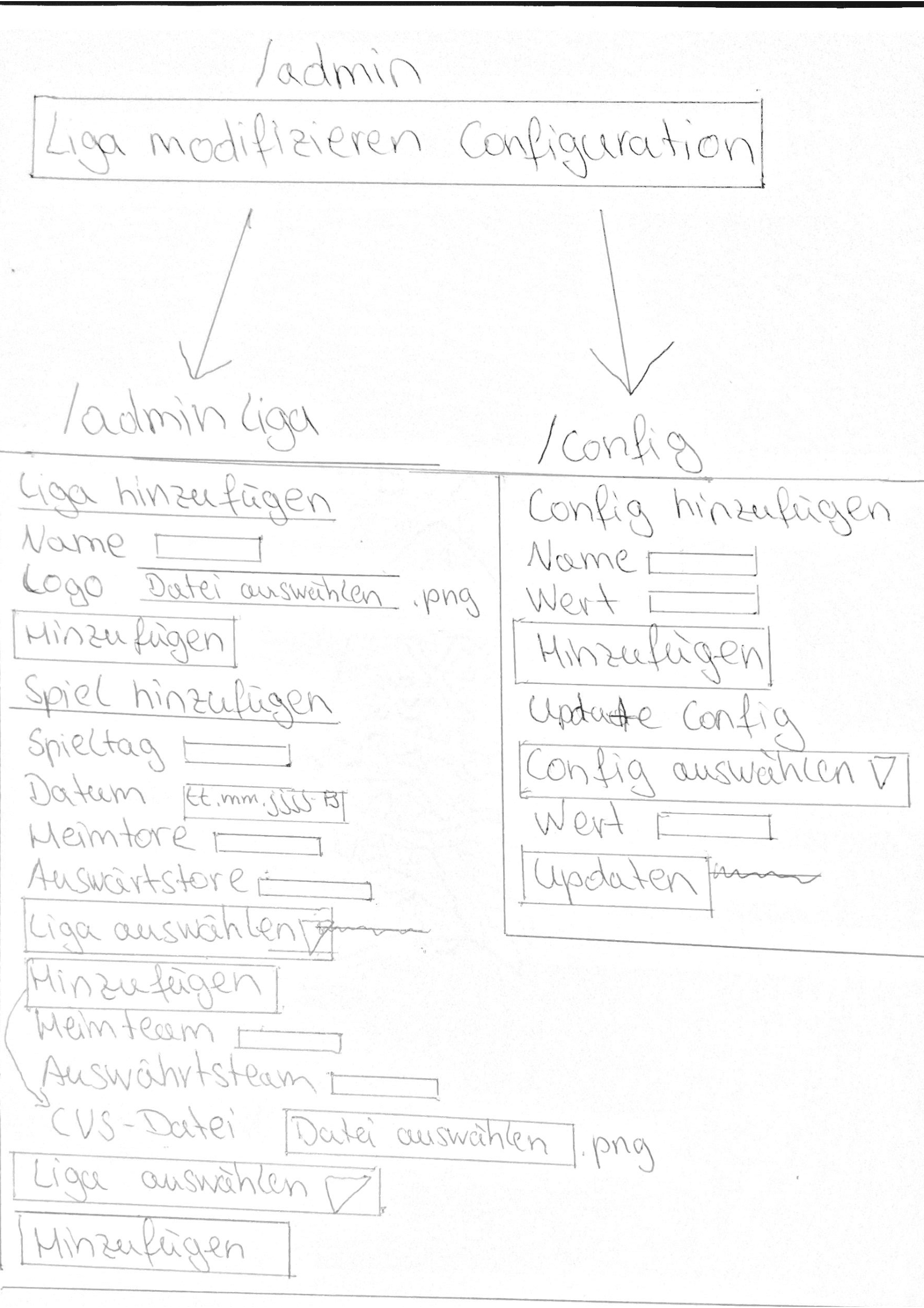
|  |  |
| --- | --- |
| **User Story-ID** | 5.7 |
| **User Story-Beschreibung** | Als Admin möchte ich, dass es mir möglich ist Spiele manuell hinzufügen zu können, um eine 2. Option Eingabemöglichkeit zu haben als Backup oder für einzelne Spiele. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor** | Alex Grebennikov |
| **Abhängigkeiten zu anderen User Stories** | 2.1, 3.1, 3.3 |

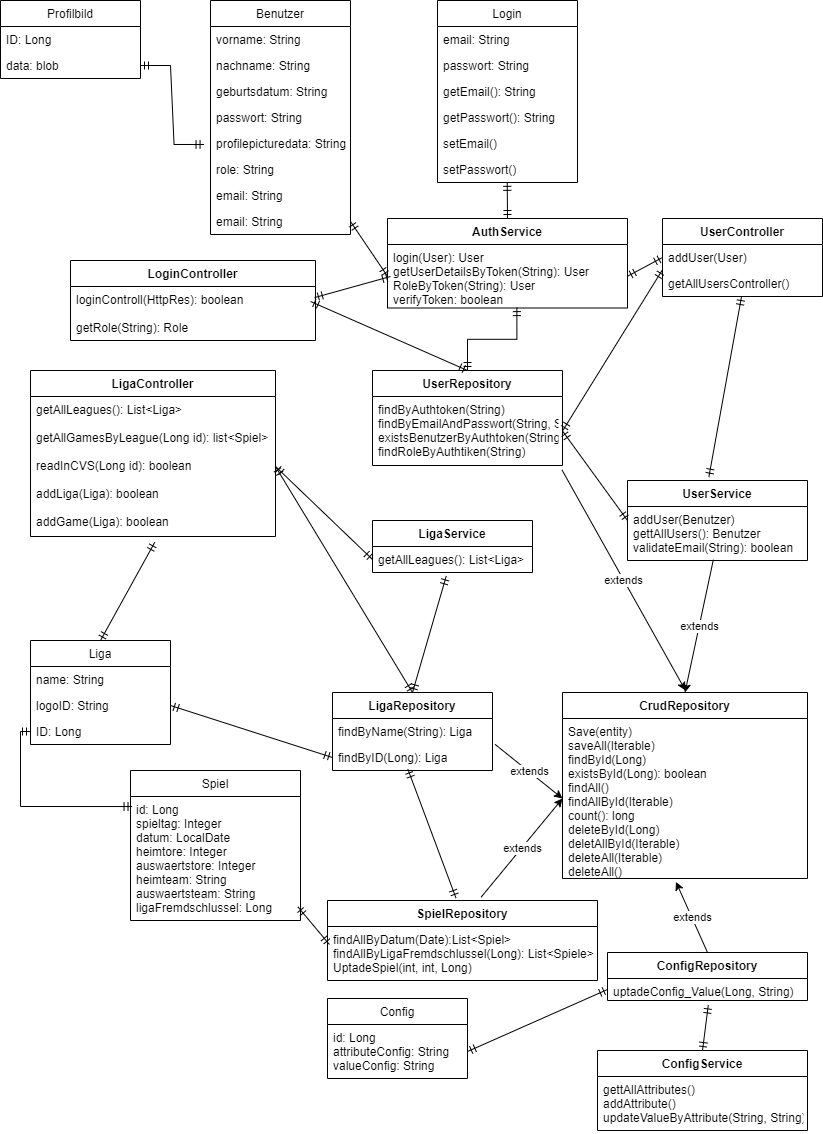
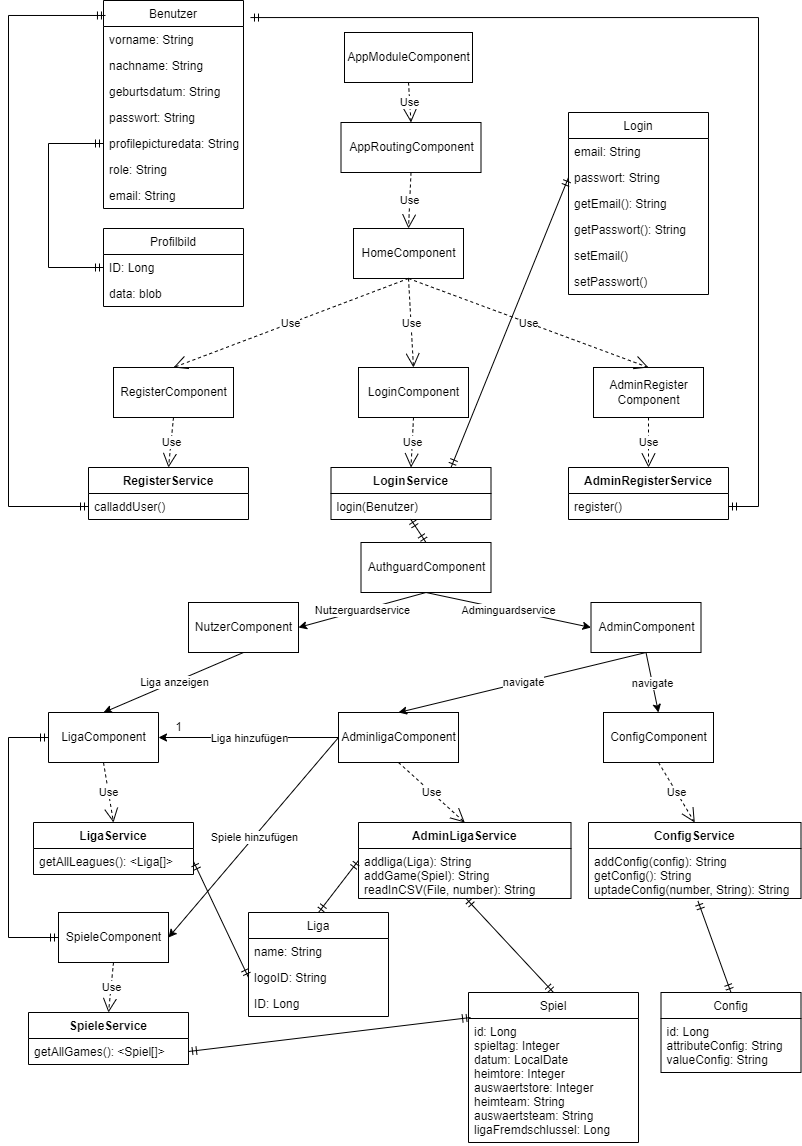
|  |  |
| --- | --- |
| **User Story-ID** | 6.1 |
| **User Story-Beschreibung** | Als Admin möchte ich, dass es mir möglich ist das Systemdatum nachdem sich das System richtet zu ändern, damit man entscheiden kann auf welche Spieletag man tippen kann. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Niedrig |
| **Autor** | Tom Schneider |
| **Abhängigkeiten zu anderen User Stories** | 2.1, 3.1, 3.3 |

|  |  |
| --- | --- |
| **User Story-ID** | 7.1 |
| **User Story-Beschreibung** | Als jeder Nutzer möchte ich, dass es mir möglich ist die Ligen angezeigt zu bekommen, damit man die volle Information hat zum tippen. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Mittel |
| **Autor** | Alex Grebennikov |
| **Abhängigkeiten zu anderen User Stories** | 2.1, |

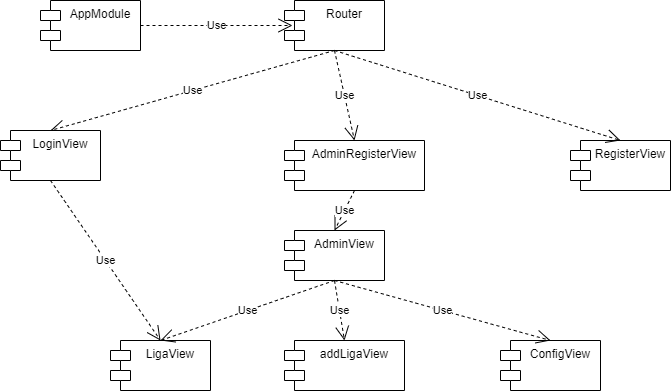
**Papierprototypen**





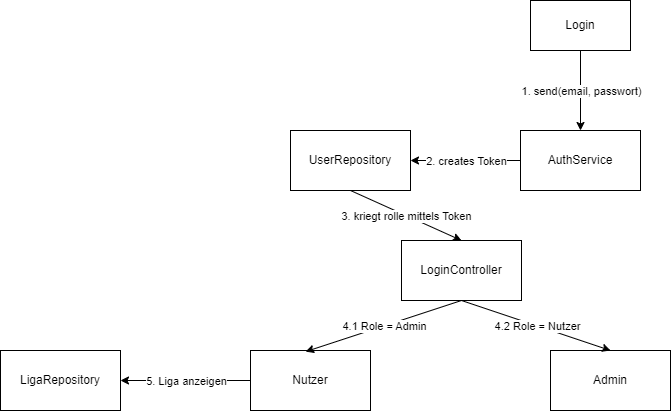
**Klassendiagramm Backend****Komponentendiagramm Frontend**

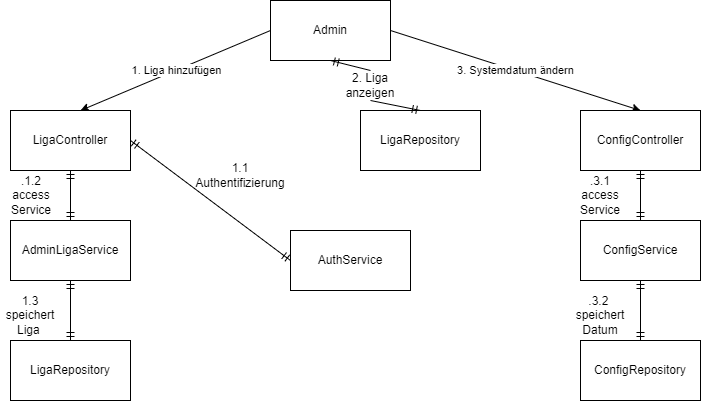
**Routing Frontend**



## 

## **Verhaltensdiagramme (Kommunikationsdiagramme):**





## Funktionalitätsplanung

Anhand der Funktionalitätsplanung werdet Ihr geprüft. Diese sollte dementsprechend **immer** aktuell gehalten werden und pro Funktionalität **nur einen** Verantwortlichen enthalten. Als Quellcodereferenz solltet Ihr immer euer Package, eure Klasse und die dazugehörige Methode angeben. ggf. könnt Ihr auch Zeilenangaben machen.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Funktionalität** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Registrierung von User/Admins** |  |  |  |  |  |
| 1.1 | Registrierungs Component | Alex | 2.1 | 1.1 |  | Fertig |
| 1.1.1 | Registrierungs Controller | Felix | 1.1.2 |  |  | Fertig |
| 1.1.2 | Registierungsservice | Felix | 2.3. |  |  | Fertig |
| 1.1.3 | Registrierungsview | Alex |  |  |  | Fertig |
| 1.2.1 | AdminRegistrierungsservice | Alex | 2.3 | 3.1 |  | Fertig |
| 1.2.2 | AdminRegistrierung View | Alex |  |  |  | Fertig |
| 2. | Login von User/Admin |  |  |  |  |  |
| 2.1 | Login Component | Alex | 2.2 | 2.1 |  | Fertig |
| 2.1.1 | Login Controller | Felix | 2.1.2 |  |  | Fertig |
| 2.1.2 | Login Service | Alex | 2.1.1 |  |  | Fertig |
| 2.1.3 | Login View | Alex | 3.1 |  |  | Fertig |
| 2.2 | Authguard | Felix |  | 3.3 |  | Fertig |
| 2.2.1 | Auth Service | Felix | 2.3.2 |  |  | Fertig |
| 2.3 | User Component | Felix |  |  |  | Fertig |
| 2.3.1 | User Controller | Felix | 2.3.3 |  |  | Fertig |
| 2.3.2 | User Repository | Felix | 1.; 3. |  |  | Fertig |
| 2.3.3 | User Service | Felix | 2.3.2 |  |  | Fertig |
| 2.3.4 | User Model | Felix |  |  |  | Fertig |
| 2.4. | Admin Component | Felix |  |  |  | Fertig |
| 2.4.1 | Admin Service | Felix |  |  |  | Fertig |
| 2.4.2 | Admin View | Alex |  |  |  | Fertig |
| 3. | Nutzer Ansicht |  |  |  |  |  |
| 3.1 | Liga Component | Tom |  | 7.1 |  | Fertig |
| 3.1.1 | Liga Controller | Tom | 3.1.3 |  |  | Fertig |
| 3.1.2 | Liga Repository | Tom |  |  |  | Fertig |
| 3.1.3 | Liga Service | Tom | 3.1.2 |  |  | Fertig |
| 3.1.4 | Liga Model | Tom |  |  |  | Fertig |
| 4. | Admin Ansicht |  |  |  |  |  |
| 4.1 | AdminLiga Component | Felix | 1.2.1 | 5.1; 5,2 |  | Fertig |
| 4.1.1 | AdminLiga Service | Felix | 3.1 |  |  | Fertig |
| 4.1.2 | CVS-Datei einlesen | Tom | 3.1 | 5.5 |  | Fertig |
| 4.1.3 | AdminLiga View | Felix |  |  |  | Fertig |
| 4.2 | Spiele Component | Felix | 1.2.1 | 5.7 |  | Fertig |
| 4.2.1 | Spiele Repository | Felix | 3.1 |  |  | Fertig |
| 4.2.2 | Spiele Service | Felix | 4.2.3 |  |  | Fertig |
| 4.3 | Config Component | Tom | 3.1; 1.2.1 | 6.1 |  | Fertig |
| 4.3.1 | Config Controller | Tom | 4.3.3 |  |  | Fertig |
| 4.3.2 | Config Repository | Tom |  |  |  | Fertig |
| 4.3.3 | Config Service | Tom | 4.3.2 |  |  | Fertig |
| 4.3.4 | Config View | Tom |  |  |  | Fertig |
| 5. | App Routing | Felix |  |  |  | Fertig |

## Systemtests

Systemtests sind Tests des Gesamtsystems gegen die Anforderungen nach erfolgreicher Integration. Eingaben und Sollverhalten werden dabei aus der Anforderungs-spezifikation abgeleitet.

Die Systemtests werden von Eurer Parallelgruppe spezifiziert und durchgeführt, daher ist dieser Bereich von den Mitgliedern der Parallelgruppe auszufüllen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **X** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **X** |
| **Testurteil** | **Test nicht bestanden.** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **√** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **√** |
| **Testurteil** | **Test bestanden.** | | |

# Zyklus II

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
|  |  |  |  |  |
| … |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |
| … |  |  |  |  |

## Spezifikationsplanung

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** |  |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor** |  |
| **Abhängigkeiten zu anderen User Stories** |  |

Papierprototypen

## Strukturdiagramme (Komponenten- und Klassendiagramme)

## Verhaltensdiagramme (Kommunikationsdiagramme)

## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Funktionalität** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** |  |  |  |  |  |  |
| 1.1 |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |
| 1.2.1 | …. |  |  |  |  |  |
| **2.** |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |
| … |  |  |  |  |  |  |

## Modultests

Modultests sind Komponententests. Diese werden in der Softwareentwicklung angewendet, um die funktionalen Einzelteile (Units) von Computerprogrammen zu testen.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Getestete Funktionalität** | **Quellcode Referenz** | **Status** |
|  |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | | |
| **Tester** |  | | |
| **SW-Version** |  | | |
| **Vorbedin-gung(en)** |  | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| **Nachbe-dingung(en)** |  | |  |
| **Testurteil** |  | | |

# Zyklus III

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
|  |  |  |  |  |
| … |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |
| … |  |  |  |  |

## Spezifikationsplanung

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** |  |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor** |  |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** |  |

Papierprototypen

## Strukturdiagramme (Komponenten- und Klassendiagramme)

## Verhaltensdiagramme (Kommunikationsdiagramme)

## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Funktionalität** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** |  |  |  |  |  |  |
| 1.1 |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |
| 1.2.1 | …. |  |  |  |  |  |
| **2.** |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |
| … |  |  |  |  |  |  |

## Modultests

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Getestete Funktionalität** | **Quellcode Referenz** | **Status** |
|  |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | | |
| **Tester** |  | | |
| **SW-Version** |  | | |
| **Vorbedin-gung(en)** |  | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| **Nachbe-dingung(en)** |  | |  |
| **Testurteil** |  | | |

# Nutzerhandbuch

## Technische Anforderungen

Technische Mindestanforderungen, welche das System benötigt, um wie gewünscht bedienbar zu sein.

## Installationsanleitung

Genaue Erläuterung, wie das entwickelte System vollkommen funktionsfähig auf einem Rechner in Betrieb genommen werden kann.

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte System zu bedienen ist.