SEP Projektmappe

**Projektmappe des Projektes**

SEP-Tippspiel

**Dokumentation des Projektes**

Gruppe F:

Max Armin Eschenbach

Florian Heinen

Fabian Müller

Can Kalafat

Danial Imani Shakibaei

**Hinweis**

An vielen Stellen findet Ihr im Dokument folgendes Kästchen:

*Dies ist eine Hilfestellung.*

Diese Kästen dienen dazu, Euch kurze Informationen über Ziele und Inhalte der jeweiligen Abschnitte zu geben. Auch die Beispiele und Templates dienen dazu, euch bei der Dokumentation eures Projektes zu unterstützen. **Sowohl die Kästchen als auch die Beispiele und Templates sind spätestens zur finalen Abgabe der Projektmappe vollständig zu entfernen.** Betrachtet dieses Dokument bitte nicht als Aufgabe, die man von oben nach unten abarbeiten soll; es soll vielmehr als durchgängige Dokumentation eurer Projektarbeit dienen und fortlaufend erweitert bzw. angepasst werden, sodass am Ende des SEPs der Entwicklungsprozess Eurer Software vollständig dokumentiert ist.

Das SEP-Team wünscht Euch  
**Viel Erfolg**

Inhalt

[Projektbeschreibung 4](#_Toc117616648)

[Zyklus I 5](#_Toc117616649)

[Spezifikationsplanung 5](#_Toc117616650)

[User-Stories 7](#_Toc117616651)

[Papierprototypen 10](#_Toc117616652)

[Strukturdiagramme (Komponenten- und Klassendiagramme) 13](#_Toc117616653)

[Verhaltensdiagramme (Kommunikationsdiagramme) 14](#_Toc117616654)

[Funktionalitätsplanung 16](#_Toc117616655)

[Systemtests 17](#_Toc117616656)

[Zyklus II 19](#_Toc117616657)

[Spezifikationsplanung 19](#_Toc117616658)

[User-Stories 19](#_Toc117616659)

[Papierprototypen 20](#_Toc117616660)

[Strukturdiagramme (Komponenten- und Klassendiagramme) 20](#_Toc117616661)

[Verhaltensdiagramme (Kommunikationsdiagramme) 20](#_Toc117616662)

[Funktionalitätsplanung 21](#_Toc117616663)

[Modultests 22](#_Toc117616664)

[Systemtests 22](#_Toc117616665)

[Zyklus III 23](#_Toc117616666)

[Spezifikationsplanung 23](#_Toc117616667)

[User-Stories 23](#_Toc117616668)

[Papierprototypen 24](#_Toc117616669)

[Strukturdiagramme (Komponenten- und Klassendiagramme) 24](#_Toc117616670)

[Verhaltensdiagramme (Kommunikationsdiagramme) 24](#_Toc117616671)

[Funktionalitätsplanung 25](#_Toc117616672)

[Modultests 26](#_Toc117616673)

[Systemtests 26](#_Toc117616674)

[Nutzerhandbuch 27](#_Toc117616675)

[Technische Anforderungen 27](#_Toc117616676)

[Installationsanleitung 27](#_Toc117616677)

[Bedienungsanleitung 27](#_Toc117616678)

# Projektbeschreibung

In diesem Abschnitt soll die Projektbeschreibung abgedruckt werden, die ihr als Aufgabenbeschreibung von eurem Betreuer erhalten habt. Sie dient als initiales Anforderungsdokument für eure Spezifikationsaktivitäten.

# Zyklus I

## Spezifikationsplanung

Jedes Artefakt, das im Rahmen des SEP erstellt wird, muss mit dem Namen genau einer Verantwortlichen/eines Verantwortlichen versehen werden. Das bedeutet, dass jede User Story, jeder Papierprototyp, jedes Komponentendiagramm, jedes Klassendiagramm, jedes Kommunikationsdiagramm und sämtliche Tests mit dem Namen der/des Verantwortlichen versehen und hier in der untenstehenden Tabelle entsprechend eingetragen werden muss. Natürlich kann die Gruppe gemeinsam an einem Artefakt arbeiten, als Verantwortlicher sollte aber genau eine Person eingetragen werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** | Registrierung der Nutzer |  |  |  |
| 1.1 | Registrierung eines Nutzers | User Story | Danial Imani Shakibaei | Fertig |
| 1.2 | Login eines Nutzers | User Story | Danial Imani Shakibaei | Fertig |
| 1.3 | 2FA Authentifizierung | User Story | Danial Imani Shakibaei | Fertig |
| 1.4 | Login Fenster | Papierprototyp | Danial Imani Shakibaei | Fertig |
| 1.5 | 2FA Fenster | Papierprototyp | Fabian Müller | Fertig |
| 1.6 | Registrierung eines Nutzers | Papierprototyp | Fabian Müller | Fertig |
| 1.7 | Frontend Registrierung & Login | Klassendiagramm | Florian Heinen | Fertig |
| 1.8 | Registrierung & Login | Verhaltens-diagramm | Florian Heinen | Fertig |
| **2.** | Registrierung der Systemadministratoren |  |  |  |
| 2.1 | Registrierung der Systemadministratoren | User Story | Danial Imani Shakibaei | Fertig |
| 2.2 | 2FA Authentifizierung | User Story | Danial Imani Shakibaei | Fertig |
| 2.3 | Admin Registrierung | Papierprototyp | Fabian Müller | Fertig |
| 2.4 | Frontend Registrierung & Login | Klassendiagramm | Florian Heinen | Fertig |
| 2.5 | Registrierung & Login | Verhaltens-diagramm | Florian Heinen | Fertig |
| **3.** | Ligen erstellen |  |  |  |
| 3.1 | Ligen erstellen | User Story | Max Armin Eschenbach | Fertig |
| 3.2 | Spielplan einlesen | User Story | Danial Imani Shakibaei | Fertig |
| 3.3 | Liga Daten manuell ändern | User Story | Max Armin Eschenbach | Fertig |
| 3.4 | Fenster zum Ändern/Anlegen einer Liga | Papierprototyp | Fabian Müller | Fertig |
| 3.5 | Fenster zum Einsehen der Ligadaten | Papierprototyp | Fabian Müller | Fertig |
| 3.6 | Frontend Liga erstellen | Klassendiagramm | Florian Heinen | Fertig |
| 3.7 | Liga Erstellen | Verhaltens-diagramm | Florian Heinen | Fertig |
| **4.** | Systemdatum ändern |  |  |  |
| 4.1 | Systemdatum ändern | User Story | Max Armin Eschenbach | Fertig |
| 4.2 | Übersichtsseite Admin | Papierprototyp | Fabian Müller | Fertig |
| 4.3 | Systemdatum ändern | Papierprototyp | Fabian Müller | Fertig |
| 4.4 | Fronten Systemdatum ändern | Klassendiagramm | Florian Heinen | Fertig |
| 4.5 | Systemdatum ändern | Verhaltens-diagramm | Florian Heinen | Fertig |
| **5.** | Backend Architektur |  |  |  |
| 5.1 | Komponentendiagramm | Komponentendiagramm | Can Kalafat | In Bearbeitung |
| 5.2 | Klassendiagramm | Klassendiagramm | Max Armin Eschenbach | Fertig |
| **6.** | Frontend Architektur |  |  |  |
| 6.1 | Komponentendiagramm | Komponentendiagramm | Can Kalafat | In Bearbeitung |

## User-Stories

**Template:**

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone:  Als <Rolle> möchte ich <Ziel> [, um/sodass <Nutzen>]  (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** | <Einschätzung der Zeit, die benötigt wird, um die Userstory zu implementieren> |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung z.B. hoch, mittel niedrig> |
| **Autor** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |

1. *Registrierung der Nutzer*

|  |  |
| --- | --- |
| **User Story-ID** | 1.1 |
| **User Story-Beschreibung** | Als Nutzer möchte ich die Möglichkeit haben ein Profil bestehend aus Vor- und Nachnamen, E-Mail, Geburtsdatum, Passwort und optionalem Profilbild zu erstellen, um meine Daten dauerhaft zu speichern. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Danial Imani Shakibaei |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 1.2 |
| **User Story-Beschreibung** | Als Nutzer möchte ich mich in mein Profil einloggen können, um an einer Tipprunde teilnehmen zu können. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Danial Imani Shakibaei |
| **Abhängigkeiten zu anderen User Stories** | 1.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 1.3 |
| **User Story-Beschreibung** | Als Nutzer möchte ich, dass der Login durch eine 2FA Authentifizierung per E-Mail realisiert wird, um eine zusätzliche Sicherheit zu gewährleisten. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Danial Imani Shakibaei |
| **Abhängigkeiten zu anderen User Stories** | 1.2 |

1. Registrierung der Systemadministratoren

|  |  |
| --- | --- |
| **User Story-ID** | 2.1 |
| **User Story-Beschreibung** | Als Nutzer möchte ich die Möglichkeit haben mich mit Vor- und Nachnamen, E-Mail und Passwort als Systemadministrator zu registrieren, um Liga-Funktionen realisieren zu können. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Danial Imani Shakibaei |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 2.2 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich die Möglichkeit haben mich mittels meines Administrator-Profils einzuloggen. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Danial Imani Shakibaei |
| **Abhängigkeiten zu anderen User Stories** | 2.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 2.3 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich die Möglichkeit haben, dass der Login durch eine 2FA-Authentifizierung realisiert wird, um zusätzliche Sicherheit zu gewährleisten. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Danial Imani Shakibaei |
| **Abhängigkeiten zu anderen User Stories** | 2.2 |
| 1. Ligen erstellen |  |
| **User Story-ID** | 3.1 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich verschiedene Ligen anlegen können, wobei eine Liga aus Namen, Spielplan und optionalem Liga-Bild bestehen soll. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Max Armin Eschenbach |
| **Abhängigkeiten zu anderen User Stories** |  |

|  |  |
| --- | --- |
| **User Story-ID** | 3.2 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich die Möglichkeit haben den Spielplan aus einer CSV-Datei einzulesen, um auch auf ältere Ligen Zugriff zu haben. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Danial Imani Shakibaei |
| **Abhängigkeiten zu anderen User Stories** | 3.1 |

|  |  |
| --- | --- |
| **User Story-ID** | 3.3 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich die Möglichkeit haben, die Liga Daten manuell zu ändern, um Tests durchführen zu können und Daten anzupassen. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Max Armin Eschenbach |
| **Abhängigkeiten zu anderen User Stories** | 3.1 |

1. Systemdatum ändern

|  |  |
| --- | --- |
| **User Story-ID** | 4.1 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich das Datum, nach dem sich das System richtet, ändern können, um die nächste Tipprunde einzuleiten |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Max Armin Eschenbach |
| **Abhängigkeiten zu anderen User Stories** | 3.1, 3.3 |

Papierprototypen

1. Registrierung der Nutzer



Papierprototyp .4: Login Fenster





Papierprototyp 1.5: 2FA Fenster

Papierprototyp 4.3: Übersichtsseite User

Papierprototyp 1.6: Registrierung eines Nutzers

1.  Registrierung der Systemadministratoren



Papierprototyp 4.1: Übersichtsseite Admin

Papierprototyp 2.3: Admin Registrierung

1. Ligen erstellen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Papierprototyp 3.4: Fernster zum Ändern/Anlegen einer Liga

Papierprototyp 3.5: Fenster zum Einsehen der Ligadaten

1. Systemdatum ändern



Papierprototyp 4.2: Übersichtsseite Admin

Papierprototyp 4.3: Systemdatum ändern

## Strukturdiagramme (Komponenten- und Klassendiagramme)

Im SEP soll die statische Struktur des Systems mittels Komponenten- und Klassendiagramme modelliert werden. Ein Komponenten- und Klassendiagramme dienen der grafischen Darstellung von Komponenten/Klassen, Schnittstellen und deren Beziehungen. Die Diagrammtypen helfen dabei, Quellcode und Implementierungsarbeiten zu strukturieren, bevor diese starten und ermöglicht eine Aufteilung der Programmieraufgaben.

### Klassendiagramm

#### Backend



#### Frontend



## Komponentendiagramm:









## Verhaltensdiagramme (Kommunikationsdiagramme)

Im SEP soll das dynamische Verhalten des Systems mittels Kommunikationsidagramme modelliert werden. Ein Kommunikationsdiagram ermöglicht die grafische Darstellung des Nachrichtenaustausches zwischen Systemobjekten. Systemobjekte können Komponenten im Komponentendiagramm und Klassen im Klassendiagramm sein. Kommunikationsdiagramme zielen darauf ab, die Zusammenarbeit der Systemobjekte darzustellen



## 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 | Verantwortlich | Abhänigg mit | User-Stories | Quellcode-Referenz | Status |
| 1. | Registrierung |  |  |  |  |  |
| 1.1 | Registrierung Admin |  |  | 2.1 |  |  |
| 1.1.1 | RegistrierungsService | Max Eschenbach |  | 2.1 |  | Erledigt |
| 1.1.2 | Registrierungsview | Fabian Müller |  | 2.1 |  | Erledigt |
| 1.1.3 | Datenbankmodell Admin | Max Eschenbach |  | 2.1 |  | Erledigt |
| 1.2 | Registrierung User |  |  | 1.1 |  |  |
| 1.2.1 | RegistrierungsService | Max Eschenbach |  | 1.1 |  | Erledigt |
| 1.2.2 | Registrierungsview | Fabian Müller |  | 1.1 |  | Erledigt |
| 1.2.3 | Datenbankmodell User | Max Eschenbach |  | 1.1 |  | Erledigt |
| 2. | Login |  |  |  |  |  |
| 2.1 | Loginview | Fabian Müller |  | 2.2, 1.2 |  | Erledigt |
| 2.2 | 2FA Abgleichview | Florian Heinen |  | 2.3, 1.3 |  | Erledigt |
| 2.3 | Validierung Login-Daten | Can Kalafat |  | 2.2, 1.2 |  | Erledigt |
| 2.4 | Mail-Sender | Can Kalafat |  | 2.3, 1.3 |  | Erledigt |
| 3. | Liga |  |  |  |  |  |
| 3.1.1 | Liga erstellen View | Florian Heinen |  | 3.1 |  | Erledigt |
| 3.1.2 | Spielplanview | Fabian Müller |  | 3.1 |  | Erledigt |
| 3.2 | CSV – Reader-Backend | Florian Heinen |  | 3.2 |  | Erledigt |
| 3.2.1 | CSV -Reader-Frontend | Max Eschenbach |  | 3.2 |  | Erledigt |
| 3.3.1 | Systemdatum ändern | Danial Imani Shakibaei |  | 4.1 |  | Erledigt |
| 3.3.2 | Systemdatum ändern View | Fabian Müller |  | 4.1 |  | Erledigt |
| 3.4.1 | LigaDaten ändern | Max Eschenbach |  | 3.3 |  | Erledigt |
| 3.4.2 | LigaDaten ändern View | Fabian Müller |  | 3.3 |  | Erledigt |
| 4. | Dashboard |  |  |  |  |  |
| 4.1 | AdminView | Fabian Müller |  | 4.1 |  | Erledigt |
| 4.2 | UserView | Fabian Müller |  | 4.1 |  | Erledigt |
| 5. | Docker | Danial Imani Shaikbae |  |  |  | Erledigt |
| 6. | Systemtests | Can Kalafat |  |  |  | Erledigt |

## 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.** | | |

### Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 18.11.2022 | | |
| **Tester** | Gruppe G | | |
| **SW-Version** | e31c802e | | |
| **Vorbedin-gung(en)** | Login-Seite aufgerufen und User in Datenbank | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt die E-Mail die eingespeicherte E-Mail-Adresse auf der Tastatur ein. | Das System zeigt die E-Mail auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das entsprechende Passwort auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Login“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **√** |
| **Nachbe-dingung(en)** | Der Nutzer sieht die 2FA-Seite und hat eine E-Mail mit dem Code erhalten. | | **√** |
| **Testurteil** | **Test bestanden.** | | |

### 2FA

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 18.11.2022 | | |
| **Tester** | Gruppe G | | |
| **SW-Version** | e31c802e | | |
| **Vorbedin-gung(en)** | Nutzer ist auf 2FA-Seite | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Code, der per Mail gesendet wurde, ein | Das System zeigt den Code auf dem Display an. | **√** |
| 2 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **√** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet. Der Nutzer sieht die Dashboardansicht. | | **√** |
| **Testurteil** | **Test bestanden.** | | |

### Erstellen einer Liga

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 18.11.2022 | | |
| **Tester** | Gruppe G | | |
| **SW-Version** | e31c802e | | |
| **Vorbedin-gung(en)** | Der angemeldete Nutzer ist ADMIN und auf der „Liga Erstellen“-Ansicht | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt einen Liganamen ein. | Das System zeigt den Liganamen auf dem Display an. | **√** |
| 2 | Der Benutzer lädt eine CSV-Datei hoch. | Das System zeigt die eingegebene Datei an. | **√** |
| 3 | Der Benutzer lädt optional ein Profilbild hoch. | Das System zeigt die eingegebene Datei an. | **√** |
| 4 | Der Benutzer speichert die eingegebenen Daten via den vorgesehenen Button. | Das System speichert die eingegeben Daten. | **√** |
| **Nachbe-dingung(en)** | Die Ligadaten sind im System gespeichert und können angezeigt werden. | | **√** |
| **Testurteil** | **Test bestanden.** | | |

# Zyklus II

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

## Spezifikationsplanung

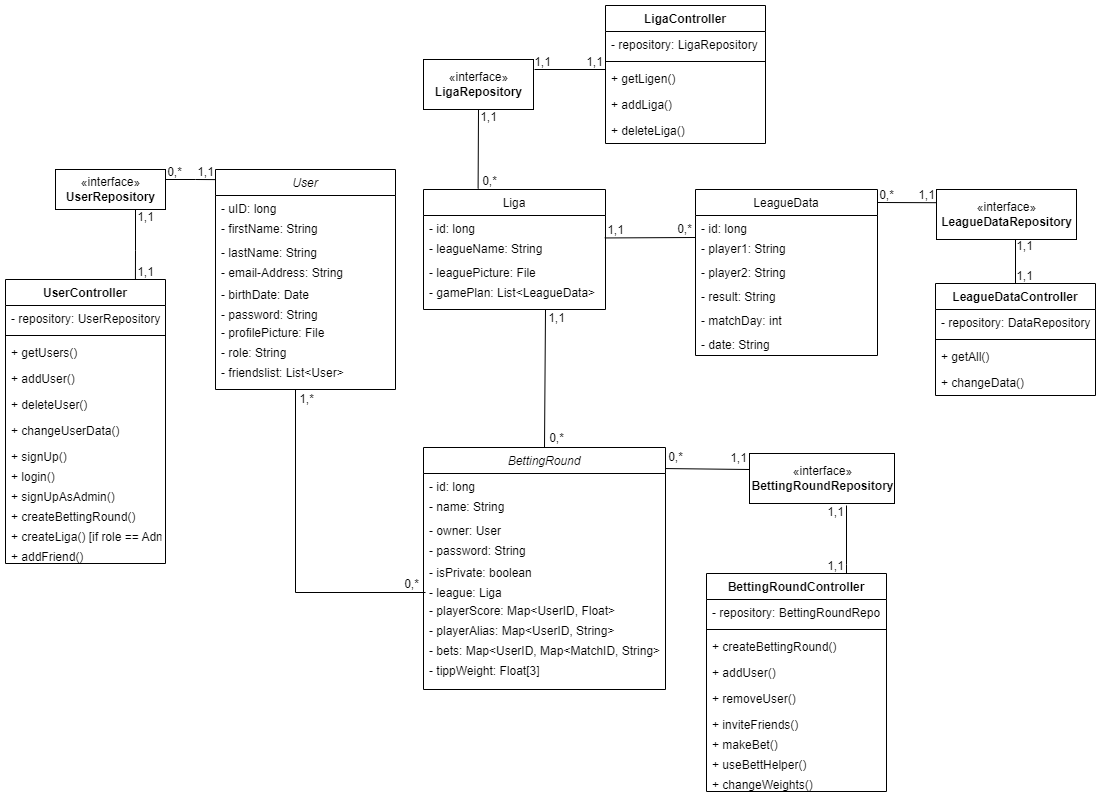
## User-Stories

Template:

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

Papierprototypen

## Strukturdiagramme (Komponenten- und Klassendiagramme)

Backend Klassendiagramm

## Verhaltensdiagramme (Kommunikationsdiagramme)

## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Funktionalität** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Tipprunde erstellen** |  |  |  |  |  |
| 1.1 | Tipprunde erstellen View |  |  |  |  |  |
| 1.2 | Tipprunde erstellen Service |  |  |  |  |  |
| 1.3 | Tipprunde Bewertung |  |  |  |  |  |
| **2.** | **Tipprunde beitreten** |  |  |  |  |  |
| 2.1 | Tipprunde View |  |  |  |  |  |
| 2.1.1 | Tipprunde beitreten |  |  |  |  |  |
| 2.1.2 | Tippnamen Wählen? |  |  |  |  |  |
| 2.1.3 | Tipps abgeben |  |  |  |  |  |
| 2.1.4 | Tipptabelle |  |  |  |  |  |
| 2.1.5 | Tipps Übernehmen |  |  |  |  |  |
| 2.2 | Tipprunde beitreten Service |  |  |  |  |  |
| 2.2.1 | Tipprunde nach Name Filtern |  |  |  |  |  |
| 2.2.2 | Tipprunde beitreten |  |  |  |  |  |
| 2.2.3 | Beitreten über Mail |  |  |  |  |  |
| 2.2.4 | Nutzer wählt Tippnamen |  |  |  |  |  |
| 2.2.5 | Nutzer kann Ergebnisse tippen |  |  |  |  |  |
| 2.2.6 | Tipptabelle einsehen |  |  |  |  |  |
| 2.2.7 | Tipps übernehmen aus anderer Tipprunde |  |  |  |  |  |
| **3.** | **Soziales** |  |  |  |  |  |
| 3.1 | Soziales View |  |  |  |  |  |
| 3.1.1 | Suchfunktion |  |  |  |  |  |
| 3.1.2 | Freundesliste |  |  |  |  |  |
| 3.2 | Soziales Service |  |  |  |  |  |
| 3.2.1 | Suchfunktion |  |  |  |  |  |
| 3.2.2 | Freundschaftsanfrage |  |  |  |  |  |
| 3.2.3 | Freundesliste |  |  |  |  |  |
| 3.2.4 | Freundesliste bearbeiten |  |  |  |  |  |
| 3.2.5 | Tipps teilen |  |  |  |  |  |
| **4.** | **Tipphilfe** |  |  |  |  |  |
| 4.1 | Tipphilfe View |  |  |  |  |  |
| 4.2 | Tipphilfe Service |  |  |  |  |  |
| 4.2.1 | Algorithmus |  |  |  |  |  |
| **5.** | **Tipprundenübergreifender Vergleich** |  |  |  |  |  |
| **5.1** | **Top View** |  |  |  |  |  |
| 5.1.1 | Top 3 Tipper |  |  |  |  |  |
| 5.1.2 | Top 3 Mannschaften |  |  |  |  |  |
| **5.2** | **Top Service** |  |  |  |  |  |
| 5.2.1 | Top 3 Tipper |  |  |  |  |  |
| 5.2.2 | Top 3 Mannschaften |  |  |  |  |  |

## 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.

* Docker
* Browser

## Installationsanleitung

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

1. Die .tar Dateien & das docker-compose file in ein beliebiges Verzeichnis kopieren
2. Im gleichen Verzeichnis die Eingabeaufforderung öffnen und die folgenden commands eingeben:

docker load --input tippspiel.tar

docker load --input client.tar

docker load --input postgres.tar

docker compose up

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte System zu bedienen ist.

Zunächst muss im Browser „localhost:4200“ eingegeben werden in der URL.

Login/Registrierung:

Hier kann man sich entweder via den vorgesehenen Eingabefeldern mithilfe von E-Mail und Passwort anmelden oder alternativ als „Admin“- oder „Basic“-User mit den entsprechenden Daten registrieren. Bei einem Login wird man auf die 2FA-Seite verlinkt, wobei der hierfür vorgesehene Code via Mail verschickt wurde (oder man nutzt den Mastercode „SecureLogin“). Hiernach und nach dem Registrierungsfeld wird man auf das rollenbasierte Dashboard verlinkt.

Dashboard:

* Systemdatum ändern: Klickt man auf diesen Button, kann man das aktuelle Datum, nach dem sich das System richtet, umstellen.
* Liga erstellen: Klickt man auf diesen Button, gelangt man zu der Seite, auf der man eine neue Liga erstellen kann.
* Ligenübersicht: Klickt man auf diesen Button, werden einem alle aktuellen Ligen angezeigt und entsprechende Ligadaten.
* Ausloggen: Klickt man auf diesen Button, wird der User ausgeloggt und man gelangt wieder zur Loginseite.

Liga erstellen:

Um eine Liga zu erstellen, muss das Input Feld für einen Liganamen ausgefüllt werden und eine CSV-Datei ausgewählt werden. Optional kann der Liga noch ein Bild in Form einer png/jpg Datei hinzugefügt werden. Durch einen Klick auf den Button "Liga erstellen" wird die entsprechende Liga dem System hinzugefügt. Durch einen Klick auf den "Zurück" Button wird man auf das Admin-Dashboard zurückgeleitet.

Ligadaten bearbeiten:

Ist man als „Admin“ eingespeichert und geht vom Dashboard aus auf „Ligenübersicht“, so kann man auf eine bestimmte Liga drücken, sodass die Ligadaten angezeigt werden. Für jede einzelnen Eintrag kann man nun rechts neben den Informationen beliebige Daten verändern. Um die geänderten Ligadaten anzeigen zu lassen, muss oben auf der Seite nochmals der Button „Ligadaten anzeigen“ benutzt werden.

Systemdatum ändern:

Um das Systemdatum zu ändern, muss das Input Feld angeklickt werden und ein Datum ausgewählt werden. Zur Änderung des Systemdatums muss folgend noch der Button "Speichern" benutzt werden. Das aktuelle Systemdatum ist hierbei oben auf der Seite einzusehen. Mit einem Klick auf den Button "Zurück" gelangt man wieder zurück zum Admin-Dashboard.