Konzeptbeschreibung

**Team: <TEAM-NAME/g6t1.>**

Mitglied 1: (Vor- und Nachname, Matrikelnummer)

Mitglied 2: (Vor- und Nachname, Matrikelnummer)

Mitglied 3: (Vor- und Nachname, Matrikelnummer)

Mitglied 4: (Vor- und Nachname, Matrikelnummer)

Mitglied 5: (Vor- und Nachname, Matrikelnummer)

Mitglied 6: (Vor- und Nachname, Matrikelnummer)

**Proseminargruppe: <PS-GRUPPEN-NR.>**

**Datum: 18.03.2021**

# Systemüberblick

Our software is a IoT and web-based trivia game for two or more teams and is played with a TimeFlip. This is a 12-sided smart dice, which has different activities, points and times on each side.   
The game is played through a web application, into which the users have to login. Everything the players need to do or know can be found inside this app. Before a game a user can form teams inside the app, select a topic and once everything is set up, start the game.

Once a game has started the web-application guides the teams through the game.  
When it’s a teams turn to play, a player has to roll the TimeFlip, a player has to switch to a different device, where the app shows the terms his teammates have to guess. The playing team gets a clock on their screen.

Opponent teams have to input into the app, whether the term was guessed correctly or not and depending on this input, the elapsed time and the given points from the TimeFlip, the Application calculates the points for the team.

Throughout the game, players always have the opportunity to check statistics of past games inside de in-game interface.

There is also an administration site to this web application. A game-manager can create new topics and terms inside these topics. Further game-curators have an overview of all currently running games.

Administrators get the possibility to manage, edit and create games.

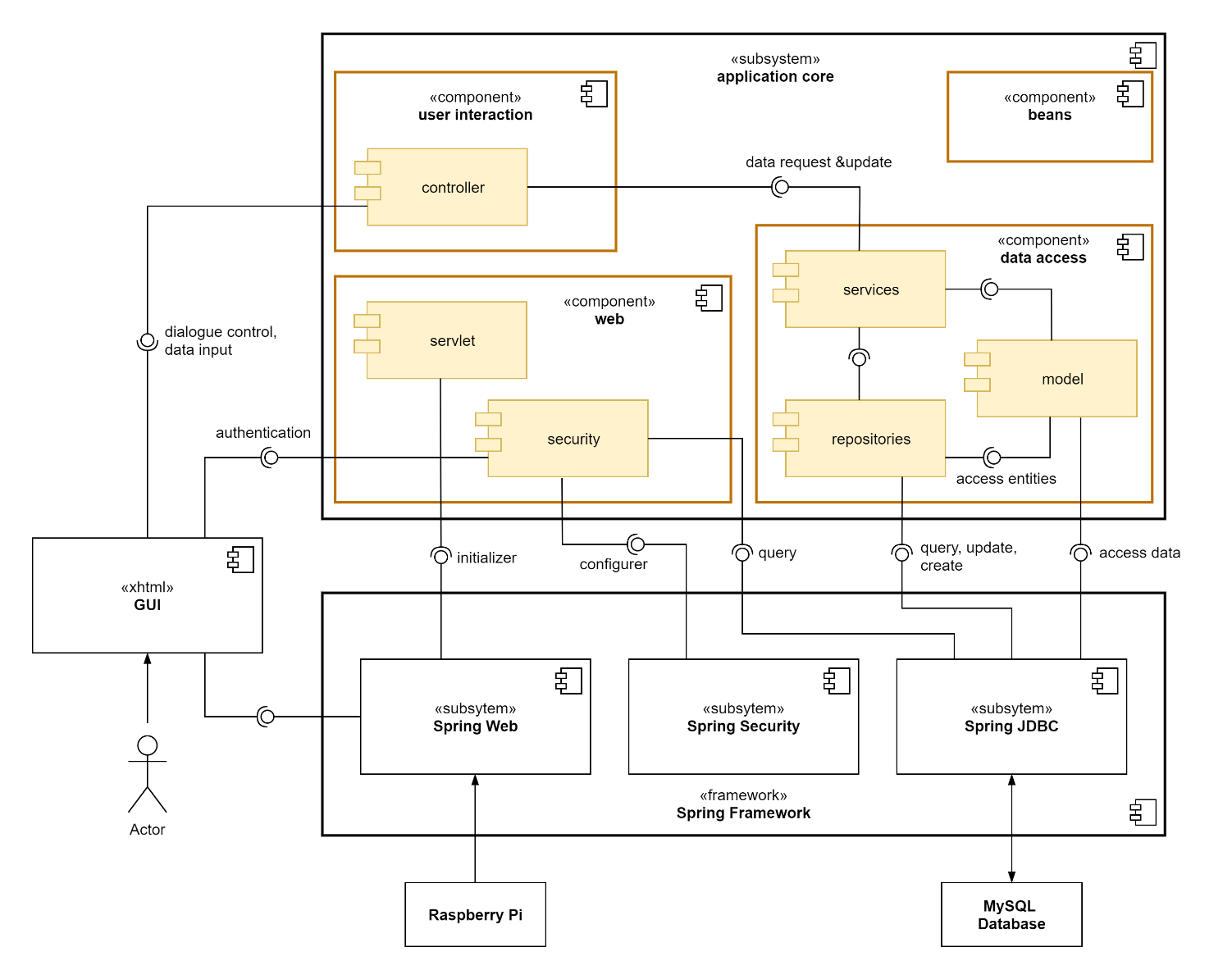
# Use Cases

*Kapitel 2 gibt einen Überblick über die Akteure und deren Anforderungen an das System. Hierzu soll ein Use Case Diagramm erstellt werden und mit entsprechenden textuellen Beschreibungen der Use Cases (Vorbedingung, Basisablauf, Alternativen, Involvierte Klassen, vgl. Vorlesungsunterlagen) ergänzt werden. Orientieren Sie sich bei diesem Kapitel an den Vorlesungsfolien und dem in der Vorlesung besprochenen Beispiel (Terminverwaltung).*

# Klassendiagramm

*Kapitel 3 soll ein fachliches Klassendiagramm mit entsprechender textueller Erklärung der Klassen und deren Aufgaben beinhalten. Orientieren Sie sich an dem in der Vorlesung besprochenen Beispiel-Klassendiagramm (Terminverwaltung).*

# SW-Architektur



The system can be divided into the spring framework it is built upon, the graphical user interface, which is realised via XHTML, and the core of the application, which holds the actual implementation of the business logic. The application core is further broken down in the following table.

|  |  |
| --- | --- |
| component | role |
| data access | retrieves, creates and updates the persisted data from the MySQL database via spring’s JDBC interface |
| ▸ model | models the persisted data as Java objects (entities) |
| ▸ repositories | used to query, create, delete and update data in the database |
| ▸ services | supplies methods to other components for all operations that need to access or modify persisted data, using the repositories and model modules |
| web | initiates and configures the web application |
| ▸ servlet | initiates the web application |
| ▸ security | manages authentication and error handling when somebody wants to access the web application  needs to query user data from the database and receives user input from the user interface |
| user interaction | interface between GUI and the data model |
| ▸ controller | responsible for dialogue control of the GUI, implement the business logic initiated by the user input, retrieve the data to display from the database and modify the persisted data by communicating with the services-module |
| beans | contains independently usable components that implement additional features |

# GUI Prototyp

*Kapitel 5 soll die Kernfunktionalitäten anhand eines GUI Prototyps darstellen. Erstellen Sie dazu einfache GUI Mockups und beschreiben Sie diese. Handschriftliches Zeichnen ist ausdrücklich erlaubt, achten Sie auf Lesbarkeit und eine angemessene Qualität der eingebundenen Scans.*

# Projektplan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Milestone | Edited by | App. Time | Deadline |
| 1 | Konzeptbeschreibung  - Systemüberblick  - Use Cases  - Klassendiagramm  - SW-Architektur  - GUI Prototyp  - Projektplan | All | 50:00 | 18.03.2021 |
| 2 | Programming Issue Distribution | All | 04:00 | 23.03.2021 |
| 3 | Model | Flaminia/Angela | 35:00 | 25.03.2021 |
|  | Database | Sebastian |  |  |
| 4 | Basic Frontend  -login  -admin menu  -manager menu | Ismail | 10:00 | 25.03.2021 |
| 5 | Raspberry Pi Setup | Michael/Max | 15:00 | 28.03.2021 |
| 6 | TimeFlip Setup | Michael/Sebastian | 15:00 | 03.04.2021 |
| 7 | REST BLE Communication | Max/Sebastian | 30:00 |  |
| 8 | Advanced Backend Functionality Controllers, Services, Repository | All | 80:00 | 13.04.2021 |
| 9 | Advanced Frontend  -lobby  -special prompts  -special error handling  -special stats view | Angela/Flaminia /Max | 50:00 | 20.04 |
| 10 | Data Management  -create game statistics  -create personal statistics  -create terms | Ismail/Michael | 20:00 | 27.04 |
| 11 | Bugfix | All | 50:00 | 27.04 |
| 12 | Stable and working system | All |  | 06.05.2021 |
| 13 | Project Results for Acceptance Test | All |  | 13.05.2021 |
| 14 | Documentation Acceptance Tests | All |  | 20.05.2021 |
| 15 | Testing | All | 30:00 |  |
| 16 | Final project results | All | 10:00 | 18.06.2021 |
| 17 | Final presentation | All | 08:00 | 21.06.2021 |