Testdrehbuch und Protokoll

Team: SpeziRangers/NR.3

Mitglied 1: (Martin Beyer, 11909749)

Mitglied 2: (Nicolas Marte, 11909113)

Mitglied 3: (Islam Mechtijev, 11910366)

Mitglied 4: (Martin Neuner, 11917314)

Mitglied 5: (Clemens Prosser, 11907449)

Proseminargruppe: 6

Datum: 13.05.2021

Inhaltsverzeichnis

1. Ausführung der Applikation

> *Hier finden Sie die erforderlichen Vorbereitungsschritte zur Ausführung der in diesem Dokument angeführten Tests.*

1.1 Setup Applikation

Setup

Prerequisite (Using docker, this requirements are not needed):

- NodeJS >=v11.0.0
- v11.0.0 <= Java <= v15.0.0

Follow the steps below:

- Clone the git repository
- `cd` into the directory

Setup using docker

- Create a production build using `mvn -DskipTests package` (We skip the tests as the database is setup using docker)
- Build the docker image using `docker build -t timeguess-webapp .`
- Start the container using `docker compose up --build webapp` (in this case, the app will be mounted at port `8080` and the database at `8081`)

Manual setup

- Install the dependencies using `mvn clean install`
- Edit the database configuration in `/backend/src/main/resources/application.properties`. A valid configuration could be:

```
spring.datasource.url=jdbc:mysql://localhost:3306/database?serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=root
```

Hint: Don't forget to adjust the Test application properties as well

- Run `mvn spring-boot:run`

All steps are done. After launching the application, it should be up and running at port `8080` (http://localhost:8080/).

Hint:

If your main focus is to edit the frontend, consider starting a second npm server using `npm run serve -- --port=8081` - this enables hot reloading. The page (with hot reloading enabled) is available at http://localhost:8081/.

1.2 Setup RaspberryPi / Würfel

Setup

Prerequisite

- wifi connection
- Java = v1.8.0 (make sure JAVA_HOME is set)
- to compile make sure you also have the JDK installed. This isn't the default yet with java 1.8.0
- maven
- tinyb
- a running backend application with no port rules blocking connection

For easy setUp instructions follow the README.md of the skeleton-bleclient.

The 'tinyb.jar' built on your system needs to be in this directory: /cube/lib/.

There is a 'tinyb.jar' already included, but in case of errors, make sure to use your own.

Building

To build the cube-project go into the directory `/cube/` and install the maven project:

```
mvn clean install
```

If you want to open the project in an IDE like IntelliJ, you might need to include this module. (Choose the cube/pom.xml and right click -> add module)

Executing

The program is a spring boot application. So simply run:

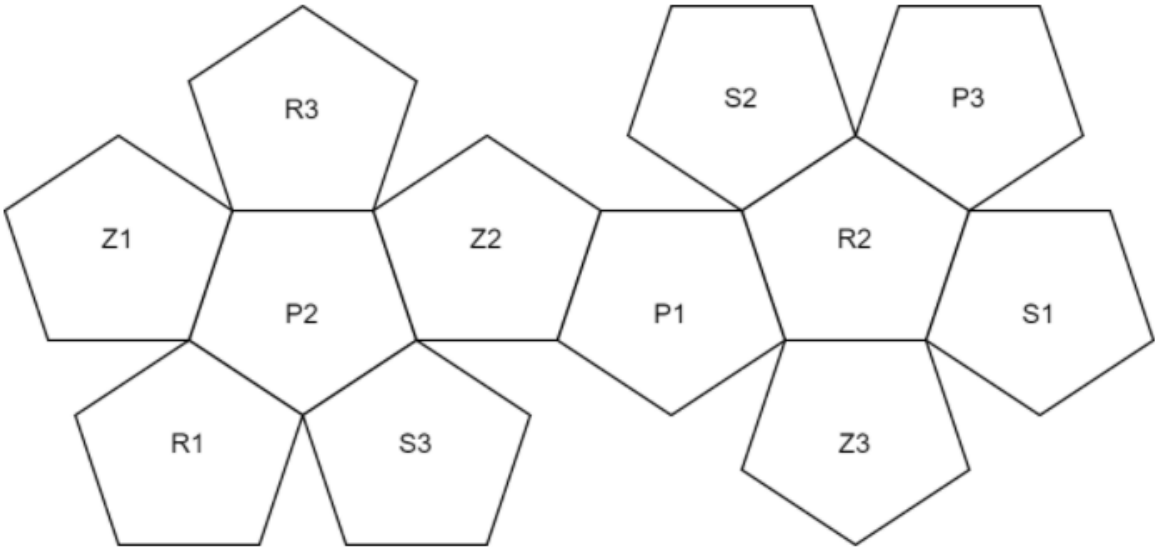
```
mvn spring-boot:run
```

Calibrating the cube:

The program guides you through the calibration. However, it is neccesarry to know what facet equals to what Label.

This picture shows the label of each facet but from the outside.

It is recommended to use the stickers for easy labeling. Alternatively, you can write the labels onto the cubes facet.



- At first the connection to the cube is checked. On our cube this fails about 70% of the time. A fix for this is to reset the battery. As the program shows you.
- Once a connection is established, the calibration starts.
- If the calibration (+ plus connection configuration) was done once before, the configuration is stored and loaded automatically. To redo the configuration, type 'redo'
- For the calibration, you are asked to place the cube in such a manner, that the corresponding facet shows up. (It does not matter, if you always place the cube with the corresponding facet to the table, but keep in mind to always use facet up or facet down)
- With the cube placed for the current calibration step, simply type 'y' into the console of the application.
- If you want to skip the calibration and don't care about correct facets, type 'skip'.
- You are now asked for a 'PiName' this is the unique identifier of your minicomputer (and thus cube). The default is simply 'piName'.
- Now please enter the Ip-Address of your backend
- In the next step you are asked for the port number of your backend application
- You are now connected to the backend. Use the frontend to interact with the cube. It should be visible at the room creation.

1.3 How to mock the cube

We implemented a feature, which allows everyone to use / simulate the cube, since not every member of the team has a physical cube at home.

How to do it (Using e.g. Postman):

- Send a POST to **localhost:8080/api/rooms/{room_id}/connect_pi**
  Here, you need an admin Token as authentication!
  Body:

```
{
    "piName": "testPi"
}
```

  Where “testPi” is just the name you want to set for the pi (ANY String). Now the Pi is successfully connected (mocked).

- Send a POST to **localhost:8080/api/rooms/{room_id}/mockCubeUpdate**
  Here, you also need an admin Token as authentication!
  No body needed.

Now, every time you need to “Roll the dice”, you can simply mock it!

2. Testdaten

Hier finden Sie die zur Verfügung gestellten Testdaten. Diese sind bereits in der Datenbank hinterlegt.

2.1 Benutzer

Zu Testbeginn sind folgende Nutzer eingerichtet:

Benutzername	Passwort	Rolle
administ	passwd	Administrator
manager	passwd	Spieleverwalter
susiK	passwd	Benutzer
hansP	passwd	Benutzer
johannG	passwd	Benutzer
jamesJB	passwd	Benutzer
lisaRitt	passwd	Benutzer
makkiF	passwd	Benutzer

2.2 Themengebiete / Begriffe

Zu Testbeginn sind folgende Themengebiete inkl. Begriffe vorhanden:

Geographie	Asien, Afrika, Nordamerika, Südamerika, Antarktis, Europa, Australien, Österreich, Wien, Deutschland, Berlin, Belgien, Brüssel, Dänemark, Kopenhagen, Italien, Rom, Polen, Warschau, Spanien, Madrid, Türkei, Ankara, Vereinigtes Königreich, London, China, USA, Japan, Mexiko, Mittelmeer
Geschichte	Steinzeit, Antike, Mittelalter, Dreißigjähriger Krieg, Romantik, Französische Revolution, Erster Weltkrieg, Weimarer Republik, Nachkriegszeit, Zweiter Weltkrieg, Kalter Krieg, Pearl Harbor, Römisches Reich, Arabischer Frühling, Kubakrise, Berliner Mauer, Terroranschläge am 11. September 2001, NATO, Cäsar, Alexander der Große, DDR, UDSSR, Gladiatorenkampf, Donald Trump, Industrialisierung, Karl Marx , Ost-West-Konflikt, Saddam Hussein, Muammar al-Gaddafi, Mao Zedong, Benito Mussolini, Josef Stalin
Filme	Titanic, Ghostbusters, Batman, Shrek, Pulp Fiction, Fight Club, The Matrix, Kevin allein zu Haus, Toy Story, Inception, König der Löwen, Charlie und die Schokoladenfabrik, Iron Man, Spiderman, Django, Kung-Fu Panda, X-Men, Life of Pi, Herr der Ringe, Forrest Gump, Der Terminator, Harry Potter, Sherlock Holmes, Alice im Wunderland, Sie nannten ihn Mücke, Der Da Vinci Code, Robinson Crusoe, Twilight, Star Wars, Interstellar

3. Testprotokoll

Dieser Abschnitt sollte zur Testdurchführung ausgefüllt werden

Testdatum: _____ (wann wurde getestet? -> Zeitraum)

Tester: _____ (wer hat getestet)

Testumgebung: _____ (z.B. Anwendung lokal auf eigenem Rechner)

Weitere Anmerkungen: _____

4. Testfälle

Die nachfolgenden Testfälle sollten getestet werden.

Die hier beschriebenen Testfälle decken die in der Konzeptbeschreibung angeführten Use Cases vollumfänglich ab. Weitere Testfälle wurden zur Überprüfung allgemeiner funktionaler Anforderungen ergänzt.

Abweichungen von den erwarteten Ergebniszustände sollten entsprechend der nachfolgenden Einstufungen dokumentiert / klassifiziert werden.

- **OK**: Keine Abweichungen gefunden.
- **Kosmetische Abweichungen**: Kleinere Layout Probleme: z.B. Zeilenumbrüche im Text ungeschickt, Texte für Buttons zu lang, usw.
- **Mittlere Abweichungen**: Die Funktionalität ist grundsätzlich vorhanden, kann aber nur eingeschränkt benutzt werden: z.B. einige erwartete Einträge in einer Dropdownliste fehlen, Datenänderungen sind erst nach Schließen und wieder Öffnen eines Dialoges sichtbar, usw.
- **Große Abweichungen**: Die Funktionalität ist nicht benutzbar, z.B. Aktionsbuttons zeigen keine Reaktion, Daten werden nicht korrekt in die Datenbank geschrieben, usw.
- **System unbenutzbar**: Die Durchführung dieses Tests hinterlässt das System in einem unbenutzbaren Zustand, z.B. System stürzt ab. Datenbank wird inkonsistent, Daten werden (ungeplant) gelöscht.

4.1 Testfälle Registrierung / Login / Logout

Use Case: Registrierung (Erfolg)

<u>Ausgangszustand</u>: Der Nutzer besitzt keinen Account

Aktion:

1. Der Nutzer klickt auf den Button „Anmelden“
2. Der Nutzer klickt auf „Sie besitzen noch keinen Account?“
3. Der Nutzer füllt das Formular aus (gewünschter Benutzername (UNIQUE), Passwort etc.)

Erwarteter Ergebniszustand:

- Der Nutzer hat einen Account erstellt
- Der Nutzer sieht die Login-Seite

Abweichungen: _____

Use Case: Registrierung (Misserfolg)

Ausgangszustand: Der Nutzer besitzt keinen Account

Aktion:

1. Der Nutzer klickt auf den Button „Einloggen“
2. Der Nutzer klickt auf „Registrieren”
3. Der Nutzer füllt das Formular aus und wählt als Username „administ“

Erwarteter Ergebniszustand:

- Der Nutzer erhält eine Mitteilung, welche ihn über das aufgetretene Problem informiert

Abweichungen: _____

Use Case: Anmeldung (Erfolg)

Ausgangszustand: Der Nutzer besitzt einen Account

Aktion:

1. Der Nutzer klickt auf den Button „Einloggen“
2. Der Nutzer trägt seinen Nutzernamen und das zugehörige Passwort ein
3. Der Nutzer klickt auf den Button „Einloggen“ im Formular

Erwarteter Ergebniszustand:

- Der Nutzer ist erfolgreich angemeldet
- Der Nutzer sieht die Startseite (Spielübersicht)

Abweichungen: _____

Use Case: Anmeldung (Misserfolg)

<u>Ausgangszustand</u>: Der Nutzer besitzt einen Account

Aktion:

1. Der Nutzer klickt auf den Button „Anmelden“
2. Der Nutzer trägt falsche Anmeldedaten ein
3. Der Nutzer klickt auf „anmelden“

Erwarteter Ergebniszustand:

- Der Nutzer erhält eine Mitteilung, welche Ihn über das aufgetretene Problem informiert

Abweichungen: _____

Use Case: Abmeldung (Erfolg)

<u>Ausgangszustand</u>: Der Nutzer ist angemeldet

Aktion:

1. Der Nutzer klickt auf das Profilbild
2. Der Nutzer klickt auf „Logout“

Erwarteter Ergebniszustand:

- Der Nutzer erhält eine Mitteilung, welche Ihn über das Ausloggen informiert
- Der Nutzer ist nicht mehr angemeldet

Abweichungen: _____

4.2 Testfälle Themengebiete / Begriffe

Use Case: Neues Themengebiet erstellen (Erfolg)

Ausgangszustand: Ein Nutzer mit Administratorberechtigungen ist angemeldet

Aktion:

1. Der Nutzer klickt auf „Dashboard“
2. Der Nutzer klickt auf die Schaltfläche „Themengebiete“
3. Der Nutzer klickt auf den „+“ Button
4. Der Nutzer gibt die gewünschten Informationen des Themengebiets ein

Erwarteter Ergebniszustand:

- Der Nutzer erhält eine Bestätigung
- Das Themengebiet wurde hinzugefügt
- Das Themengebiet wird in der Liste der Themengebiete angezeigt

Abweichungen: _____

Use Case: Neuen Begriff erstellen (Erfolg)

Ausgangszustand: Ein Nutzer mit Administratorberechtigungen ist angemeldet

Aktion:

1. Der Nutzer klickt auf „Dashboard“
2. Der Nutzer klickt auf die Schaltfläche „Begriffe“
3. Der Nutzer klickt auf den „+“ Button
4. Der Nutzer gibt die gewünschten Informationen des Begriffs ein

Erwarteter Ergebniszustand:

- Der Nutzer erhält eine Bestätigung
- Der Begriff wurde hinzugefügt
- Der Begriff wird in der Liste der Begriffe angezeigt

Abweichungen: _____

Use Case: Begriffe importieren

<u>Ausgangszustand</u>: Ein Nutzer mit Administratorberechtigungen / Spieleverwalterberechtigungen ist eingeloggt und befindet sich im Administratordashboard

1. Der Nutzer klickt auf den Button „Begriffe-importer“
2. Der Nutzer wählt das gewünschte Themengebiet aus und wählt eine lokale Datei mit Begriffen aus (.json) → Sie finden eine Beispiel Importdatei im Anhang!

Erwarteter Ergebniszustand:

- Die Begriffe wurden erfolgreich importiert
- Die importierten Begriffe sind nun dem gewählten Themengebiet zugeordnet

Abweichungen: _____

4.3 Testfälle Spielraum

Use Case: Spielraum erstellen (Erfolg)

<u>Ausgangszustand</u>: Der Nutzer ist eingeloggt

Aktion:

1. Der Nutzer klickt auf den Button „Spielraum erstellen“

Erwarteter Ergebniszustand:

- Der Nutzer befindet sich in einem Menü, in welchem er die gewünschten Einstellungen für das Spiel vornehmen kann

Abweichungen: _____

Use Case: Spielraum beitreten (Erfolg)

<u>Ausgangszustand</u>: Der Nutzer ist eingeloggt

Aktion:

1. Der Nutzer wählt den gewünschten Spielraum per Mausklick aus

Erwarteter Ergebniszustand:

- Der Nutzer ist dem gewünschten Spielraum beigetreten
- Der Nutzer sieht die Mitglieder des Spielraums

Abweichungen: _____

Use Case: Spielraum verlassen (Erfolg)

<u>Ausgangszustand</u>: Der Nutzer befindet sich in einem Spielraum

Aktion:

1. Der Nutzer klickt auf den Button „Raum verlassen“

Erwarteter Ergebniszustand:

- Der Nutzer hat den Raum verlassen

Abweichungen: _____

Use Case: Spiel starten (Erfolg)

<u>Ausgangszustand</u>: Der Nutzer ist Host (Ersteller) eines Spielraums

Aktion:

1. Der Nutzer klickt auf „Spiel starten“

Erwarteter Ergebniszustand:

- Alle Teilnehmer des Spielraums befinden sich nun im Spiel
- Das Spiel startet

Abweichungen: _____

Use Case: Team erstellen

<u>Ausgangszustand</u>: Der Nutzer befindet sich in einem Spielraum

Aktion:

1. Der Nutzer klickt auf „Team erstellen“

Erwarteter Ergebniszustand:

- Das Team ist nun zur Auswahl verfügbar

Abweichungen: _____

Use Case: Team auswählen

<u>Ausgangszustand</u>: Der Nutzer befindet sich in einem Spielraum

Aktion:

1. Der Nutzer wählt das gewünschte Team aus

Erwarteter Ergebniszustand:

- Der Nutzer ist nun Teil des ausgewählten Teams

Abweichungen: _____

Use Case: Virtueller Nutzer hinzufügen

<u>Ausgangszustand</u>: Ein Nutzer befindet sich im Spielraum

Aktion:

1. Ein Nutzer klickt auf den Button „Virtuellen Nutzer hinzufügen“
2. Der Nutzer wählt einen Namen für den Virtuellen Benutzer

Erwarteter Ergebniszustand:

- Ein Virtueller Nutzer wurde erstellt und befindet sich im Spielraum

Abweichungen: _____

Use Case: Themengebiet auswählen

<u>Ausgangszustand</u>: Der Nutzer befindet sich in der Spielraum-erstellen Ansicht und ist Host

Aktion:

1. Der Nutzer klickt auf den Button „Themengebiet auswählen“
2. Der Nutzer wählt eines der aufgelisteten Themengebieten aus

Erwarteter Ergebniszustand:

- Das Themengebiet wurde erfolgreich ausgewählt

Abweichungen: _____

5. Anhang

5.1 Glossar

Siehe

(→ Git: g6t3/documentation/glossary_mapping.md)

5.2 Referenzierte Dokumente

- Konzeptbeschreibung
  (→ Git: g6t3/documentation/1-Konzeptbeschreibung_Team_g6t3.docx)

- Begriffe-import Datei
  (→ *Git: g6t3/documentation/topic_import.json*)