

Abnahmetest für Team 1

Bela Pfahl, Daniela Rainer, Daniel Rainer,
Joshua Ocker, Laurin Wischounig

Datum: 20.05.2021

Inhaltsverzeichnis

1. Zusammenfassung	2
2. Funktionalität	2
3. Performanz, Fehlertoleranz und Stabilität	3
4. Datenschutz	4
5. Usability	4
6. Testdrehbuch	5
7. Weitere Auffälligkeiten	6
7.1 JUnit Tests	6
7.2 Konzeptbeschreibung	6
7.3 Projektdateien	6
7.4 Docker	7
8. Tabellarische Aufstellung festgestellter Mängel	7

unsinn

wird geändert

optional

1. Zusammenfassung

In der aktuellen Version des Systems ist bereits der größte Teil der vorgegebenen Funktionen der Aufgabenstellung implementiert. Die Funktionalität der Benutzerverwaltung und Statistik, die Würfelanbindung sowie die Spielelogik sind vollständig implementiert und können verwendet werden. Das System erweckt jedoch aktuell insgesamt einen unfertigen Eindruck.

Neben einigen Unausgereiftheiten in der UI ist es leider nur schwer möglich, überhaupt ein Spiel zu starten und damit die Spielelogik zu testen. Bei der Verwendung der Applikation treten immer wieder Fehler auf, die nur durch erneutes Laden der Applikation behoben werden können. Kleinere Tätigkeiten wie zB. das Löschen von Topics sind aufwändig und können durch verschiedene Quality-Of-Life improvements stark erleichtert werden. Die Anpassung der UI auf Geräte verschiedener Größen ist grundsätzlich gegeben, sollte jedoch ebenfalls durch Finetuning flüssiger gestaltet werden.

Die Spielelogik und damit der Spielablauf nach dem Start läuft weitgehend gut und fehlerfrei – die Begriffe, Punkte und der Timer werden den verschiedenen Nutzern korrekt angezeigt und das Design ist insgesamt stimmig. Das Backend bzw. UI ist zur Zeit jedoch darauf ausgelegt, dass zu jedem Zeitpunkt nur ein einziges Spiel gleichzeitig im Gang ist: **Es ist dem User nicht möglich, auszuwählen welchem Spiel er beitreten will.** Desweiteren geht der RaspberryPi aktuell davon aus, dass das Backend am Pi selbst deployed wird. Im Projekt fehlen zum jetzigen Zeitpunkt Sicherheitsmaßnahmen wie Autorisierungen praktisch gänzlich: So ist es möglich, Daten fremder Benutzer auszulesen und Würfelergebnisse an fremde Spiele abzusenden. Hier sollte durch entsprechende Maßnahmen entgegengewirkt werden.

Positiv zu Erwähnen ist der Grundansatz, möglichst viele Seiten im Frontend durch eine Hilfsfunktion besser verständlich zu machen. Auf dieser Basis können im weiteren Projektverlauf echte Hilfsseiten entstehen, die die Benutzung des Systems vereinfachen. Auch sehr loblich ist die umfangreiche und vielseitige Dokumentation des Projektes und des Setups, die durch mehrere Dokumente mitgeliefert wurde.

Die sehr umfangreiche Test-Suite mit zahlreichen Junit-Tests stellt ein gutes Fundament für eine hohe Test-Coverage dar. Es müssen hierfür jedoch viele Tests entkommentiert, repariert und ausgebaut werden. Alles in allem ist zu sagen, dass das Projekt im jetzigen Stand die grundsätzlichen Vorgaben der Projektbeschreibung erfüllt und fast vollumfänglich verwendbar ist. Es muss jedoch noch einiges an Arbeit in die Usability, die Fehlerbehandlung und Code-Qualität sowie Testabdeckung gesteckt werden. Es wäre nun angebracht, sich auf die Fertigstellung der bestehenden Features zu konzentrieren, bevor weitere Funktionalitäten implementiert werden.

2. Funktionalität

Die Software besitzt bereits zum jetzigen Zeitpunkt einen Großteil der versprochenen Funktionen. **Es fehlen noch Spielestatistiken mit Informationen zu anderen Spielern und vergangenen Spielen**, wie sie auf Seite 23 der Konzeptbeschreibung zu finden sind. Diese finden sich zwar auf verschiedensten Seiten, eine große Übersicht ist jedoch nicht vorhanden. **Ebenso fehlen einsehbare Spielerprofile** und eine entsprechende Spielersuchfunktion.

Die Aufgabenstellung ist dementsprechend auch fast vollständig erfüllt, wobei weitere kleinere Features, wie zum Beispiel das **ändern von Passwörtern** oder das **Registrieren auf der Startseite**, eine Bereicherung wären. Weiters sollten **Spieler auch die Möglichkeit bekommen ein Spiel zu starten**.

Vergessen wurden aber **Benachrichtigungen an User, die zu Spielen eingeladen wurden**. Außerdem war der **Json Import eigentlich dafür bestimmt Wörter zu importieren, und nicht ganze Topics**.

Skripts zur Konfiguration und Ausführung des RaspberryPi-clients sind zwar vorhanden, funktionieren aber kaum. Pis identifizieren sich über eine hard-coded IP und gehen davon aus, dass das Backend

auf dem Pi selbst läuft. Es ist somit zur Zeit nur möglich ein nicht-lokales Backend zu verwenden, indem man selbst die erstellte Konfigurationsdatei bearbeitet. Die hard-coded IP führt offensichtlich zu Problemen, sobald mehr als 1 Pi verwendet wird.

3. Performanz, Fehlertoleranz und Stabilität

Die Performance der Anwendung wirkt insgesamt absolut adequat. Beim Testen der Anwendung sind zu keinem Zeitpunkt langsame Reaktionszeiten oder andere merkbare Verzögerungen aufgetreten. Auch der Start der Software verläuft in erwartbarer Zeit. Die Performance stellt im aktuellen Zustand also absolut kein Problem dar.

Es ist jedoch leider wirklich nicht schwer, auf fehlerhafte Zustände im System zu stoßen. Im folgenden werden von diesen Fehler- und Problemfällen einige gefundene exemplarisch aufgelistet, um eine Basis für eine Verbesserung dieser zu bieten.

(a) In der UI werden Null- und Leer-String-Eingaben nicht ausreichen behandelt: Es erscheinen falsche und/oder unpassende Fehlermeldungen und die Länge der Inputwerte wird nicht überprüft. Beispiele für die schlechte Null-Behandlung sind:

- ✓ ■ Json Import: "Successful: null was uploaded" **leere / falsche Dateien**
- ✓ ■ Topic erstellen: "Topic already exists" **bei fehlendem Namen**
- ✓ ■ Terms: "" kann als Term verwendet werden
- ✓ ■ User erstellen ist ohne Passwort möglich
- ✓ ■ Spiel ohne Namen kann erstellt werden

Auch bei anderem Fehlverhalten erhält der Benutzer unpassende und ungenaue Nachrichten, was auch im Kapitel 5, Usability, beschrieben wird.

(b) Das Löschen von Objekten funktioniert weitgehend fehlerfrei. Einzig **beim Löschen eines Users ist dieser immer noch bei "Chat - Who's online?" in der Lobby gelistet.**

(c) Sehr fehleranfällig ist leider die Seite "game_room/join.xhtml". Selbst wenn man nicht bewusst etwas falsch machen will kommt man immer wieder in Zustände, in denen man weder weiß, wie man sie verursacht hat, noch wie man wieder heraus kommt. Meistens hilft nur ein Neustart. Aufgetreten sind beispielsweise die verschiedensten Fehler:

- ✓ ■ "no records available"
- "3/2" bzw "4/2 players assigned."
- Assignen nicht mehr möglich
- ✓ ■ Spiel starten nicht möglich
- ✓ ■ Erstellt ein neu erstellter Admin ein Spiel, findet er nur diejenigen Spieler die manuell erstellt wurden, aber nicht die Anderen

Choose your Team Players

Team Name:

4 of 2 players assigned.

Player Name	available
admin	assigned
user1	yes
user2	assigned
elvis	yes
topPlayer	yes
mrNice	yes
dangerdan	assigned
dangerdumm	yes

Join Game

Choose your Team Players

Team Name:

1 of 2 players assigned.

Player Name	available
test1	assigned
test2	yes

Join Game

Startseite eines Spiels mit verschiedenen fehlerhaften Zuständen.

(d) Das Backend startet nicht auf der aktuellen Maven-version 3.8.1 (siehe <https://forum.primefaces.org/viewtopic.php?t=66660>). Der Docker-container startet ohne Probleme.

(e) Das Konfigurationsskript für den RaspberryPi erstellt eine fehlerhafte Konfiguration, falls der Pi mehr als ein Networkinterface hat. Dies führt schlussendlich zu einem Runtimeerror, da das Programm die IP nicht parsen kann.

Die Fehlerbehandlung bei Absturz des Programms ist gegeben: Der Client schafft es den Abbruch des Programms mittels Ctrl+C problemlos zu handhaben und kann sich danach wieder ohne Probleme mit dem TimeFlip verbinden.

Das Backend verarbeitet die HTTP-posts vom Würfel fehlerfrei.

4. Datenschutz

✓ Die Sicherheit für die Würfel API ist von der Idee her gut umgesetzt. Das einzige Problem besteht darin, dass die IPs der Würfel frei einsehbar sind und ausreichen um an den API key eines Würfels zu kommen. In der Folge können die Würfel nach Belieben imitiert werden.

Im Frontend werden den NutzerInnen nur die ihren Berechtigungen entsprechend Inhalte angezeigt. Allerdings sind ansonsten (mit Ausnahme des Ladens von Topics) absolut keine Zugriffs-Einschränkungen zu finden. Der Datenschutz ist damit leider nicht zur Genüge gewährleistet, hier müssen noch entsprechende Lösungen gefunden werden (am einfachsten mit @PreAuthorize Annotations an passenden Stellen).

5. Usability

Die Software weist nicht die beste Benutzerfreundlichkeit auf, mit ein bisschen Einarbeitung ist es aber schaffbar sich zurecht zu finden. Gemeinsam mit der Fehlerbehandlung stellt die Usability den Großteil der Schwierigkeiten mit dem aktuellen Projektstand dar. Hier sind wiederum einige besonders schwerwiegende Probleme exemplarisch aufgelistet.

(a) Das Starten eines Spiels ist nicht sehr intuitiv. Die Daten in der "Available games" Lobby suggerieren, dass es laufende Spiele gibt. So ist zum Beispiel user2 im Spiel mit ID 1, kann diesem aber nicht beitreten. Ein neues Spiel kann nur von einem Admin oder Game Manger gestartet werden und da keine Einladungen verschickt werden kann nun jeder Benutzer dem Spiel beitreten.

(b) Auch die Teambildung ist eingeschränkt, da jeder Benutzer der dem Spiel beitrifft, ein neues Team

bildet. In diesem Prozess kommt es immer wieder zu diversen Fehlern, die das Zuweisen von Spielern unmöglich machen. Manchmal ist es, aufgrund von angeblich vollen Teams, nicht möglich, einem neu erstelltem Spiel beizutreten. Der Spielbeitritt ist insgesamt sehr fehleranfällig und schwer zu verstehen (siehe auch Abbildungen unter 3.).

✓ (c) Die Verarbeitung von Fehlern im Frontend ist aktuell unzureichend. Fehlermeldungen sind nicht sonderlich detailliert und vermitteln manchmal einen falschen Informationsstand: Ein Beispiel hierfür ist der Upload von Topics über ein Json File: Man erhält immer eine Meldung, dass etwas hochgeladen wurde, auch wenn man nichts oder eine falsche Datei hochlädt. Die Meldung, dass eine Topic erstellt wurde erscheint bei einem problemlosen Ablauf. Wenn man jedoch nichts davon weiß, ist man der Meinung dass das hochgeladene Dokument richtig verarbeitet wurde und die Topic erstellt werden konnte. Die Fehlermeldungen sollten differenzierter und verständlicher gestaltet werden.

(d) Das Erstellen und Löschen von Topics ist sehr aufwändig gestaltet: Für das Erstellen eines jeden Wortes muss der Dialog geöffnet, das Wort eingegeben, und bestätigt werden. Beim Löschen müssen dann alle Wörter einzeln entfernt werden.

✓ (e) Bei Fehlern, die während des Erstellens eines neuen Users im User Hub auftreten, wird der Dialog zur Bearbeitung immer wieder frühzeitig geschlossen.

✓ (f) Der Unterschied zwischen den Seiten Manager Hub und Available Games geht nicht deutlich hervor. Es ist nicht ersichtlich wie es möglich sein kann, dass Spiele keine Teams haben. **Manager Hub: Beschreibung im header hinzufügen (siehe Games tab)**

(g) Als BenutzerIn ist es wirklich schwierig nachzuvollziehen, warum bestimmte Fehler oder Systemabläufe auftreten und wie man diese umgehen kann. Die eingebaute Hilfefunktion ist ein grundsätzlich nützliches Feature, welches durch den Ausbau der Hinweise selbst noch hilfreicher werden kann.

✓ (h) Des Weiteren sollte man sich in der Oberfläche auf eine einheitliche Sprache einigen. Eine Mischung von Deutsch und Englisch ist hier nicht angebracht. Dies sollte jedoch keinen größeren Aufwand darstellen, da die Unklarheiten nur an einigen Stellen existieren. **Topic Hub: Tabellenüberschriften**

Derzeit ist eine Verwendung der Software ohne Einarbeitung zwar möglich, frustriert aber zu sehr. Es ist sicherlich möglich, die Usability durch die Anpassung der UI und der Fehlerbehandlung deutlich zu verbessern. Erste Schritte könnten hier Folgendes darstellen, (i) die Hilfs-Dialoge mit detaillierteren Erklärungen zu füllen, (ii) Fehlerhandling zu vereinheitlichen und informativer zu gestalten und (iii) häufige Abläufe zu vereinfachen, wie etwa Topics mit beinhalteten Wörtern löschen, Spiele flüssig erstellen und das Beitreten von Spielen verständlicher gestalten.

Die Readme's zur Einarbeitung sind grundsätzlich hilfreich. Sie enthalten jedoch mehrmals hard-coded IPs und Pfade von einem System der Entwickler, die in der Endversion entfernt werden sollten. Der Start des Clients mittels Skript funktioniert gut und ist sehr angenehm.

6. Testdrehbuch

Das Testdrehbuch enthält viele Testfälle, die alle Usecases der Konzeptbeschreibung abdecken und die einzelnen Nutzeraktionen nachvollziehbar beschreiben. Auch die Gliederung und Nummerierung des Dokuments ist übersichtlich und einleuchtend.

Es wäre interessant, die Angabe, dass keine Abweichungen aufgetreten sind, mit den zahlreich erstellten JUnit Tests zu belegen. Weiters gehen die Testfälle stark von einem plangemäßen Ablauf aus, fehlerhafte Prozesse werden nicht behandelt. Im Laufe des Abnahmetests konnten wir uns davon überzeugen, dass die Testfälle großteils keine Fehler erzeugen, man aber bei einer geringen Abweichung von der Aktionsbeschreibung schnell auf verschiedenste Probleme trifft. Dennoch waren sie eine Hilfe bei der Einarbeitung.

Während die Testfälle einen soliden Eindruck hinterließen und nur mehr eine geringfügige Überarbeitung

benötigt wird, ist das Übrige noch ausbaufähig. Die Vorlage wurde teils nicht angepasst, Kommentare und Anweisungen für die einzelnen Kapitel wurden nicht entfernt und es gibt keine selbst geschriebenen Beschreibungen. Als Beispiel wäre hier die Kapitel 2 "Testprotokoll" zu nennen, das nicht ausgefüllt wurde, oder Seite 2, auf der noch Michael Breu mit seinen Initialen Beispielen eingetragen ist.

7. Weitere Auffälligkeiten

Im Weiteren führen wir hier einige Punkte auf, die von Interesse seien dürften.

7.1 JUnit Tests

Es ist beachtlich, wie viele JUnit Tests erstellt wurden. Dennoch liegt die Testabdeckung bei sehr geringen Werten. Wenn man die Datei "index.html" öffnet, die bei "mvn jacoco:report" erstellt wird und sich im Ordner "target\site\jacoco" befindet, sieht man, dass nur 25% der Instructions und 12% der Branches abgedeckt sind. Der Report zeigt, dass viele Klassen nur unzureichend getestet wurden, was auch daran liegen mag, dass sehr viele Tests aus verschiedensten Gründen fehlschlagen, darunter fehlende @PreAuthorize, NullPointerExceptions, aber auch abweichende Ergebnisse. Wenn man "mvn test" ausführt, erhalten wir Tests run: 146, Failures: 22, Errors: 36, Skipped: 0.

Uns ist diesbezüglich jedoch nicht bewusst, ob wir die Tests korrekt durchführen, oder hier ein anderes Setup als das zu testende Team verwenden. Auffällig ist, dass Testmethoden oft nur eine auskommentierte @Test Annotation haben, was die schlechte Testabdeckung vermutlich erklärt.

Positiv zu betonen sind definitiv die Frontend Controller Tests mit Mockito, welche eine weitere Testabdeckung ermöglichen. Diese stellen eine sehr gute Basis für den Test weiterer Edge cases dar.

7.2 Konzeptbeschreibung

Die Konzeptbeschreibung ist schlüssig und gut strukturiert. Sie ist verständlich und hilft bei der Einarbeitung und Wartung des Systems. Als Orientierungspunkt für die Suche nach verschiedenen Codestellen ist sie sehr hilfreich. Einige Fehler, die noch vorhanden sind, werden hier aufgelistet.

(a) Im Usecases Diagramm sind die Vererbungen falsch eingezeichnet: Erstellt man in der Webapp einen User, der beispielsweise nur Admin ist, kann dieser trotzdem ein Spiel spielen und auf die Topic Hub zugreifen.

(b) Es wäre sinnvoll, dem Spieler mehr Befugnisse zu erteilen, besonders das Starten eines Spieles sollte nicht nur Spielerverwalten vorbehalten sein.

(c) Das Klassendiagramm ist übersichtlich, jedoch stimmen die Kardinalitäten nicht immer mit der Umsetzung überein (ein Topic muss 30 Begriffe haben, ein Spiel kann mehrere Topics haben).

(d) Besonders gut gelungen ist Kapitel 4, die Softwarearchitektur, wobei eine kurze Beschreibung bei der Verteilungssicht noch fehlt.

(e) Der GUI-Prototyp ist durchwegs gelungen, verspricht aber Features, die momentan noch nicht vorhanden sind.

7.3 Projektdateien

Im zip Archiv, das wir erhalten haben, befinden sich Dateien, die nicht in git landen sollten: odt lock Dateien, target/ und .idea/. Diese Dateien/Ordner sollten mittels entsprechenden Einträgen in .gitignore von der Versionskontrolle ausgenommen werden.

7.4 Docker

Die Application mit Docker zu starten funktioniert grundsätzlich gut. Allerdings wäre es es angenehm wenn eine `docker-compose.yaml` Datei mitgeliefert werden würde, damit der Start mit einem einzigen Command durchführbar ist. Das erleichtert die Bedienung und kann vermeiden, dass das System Müll hinterlässt, den man manuell entfernen muss. Die Dockerfile könnte so geschrieben werden, dass Maven die benötigten Dependencies im Image speichert, was vermeiden würde, dass sie beim Starten jedes Mal neu heruntergeladen werden.

8. Tabellarische Aufstellung festgestellter Mängel

In der nachfolgenden Tabelle funktioniert die Klassifizierung nach dem folgenden Schema: <Thema>: <Schwere>. Bei der Schwere steht 1 für leichter, 2 für mittlerer und 3 für grober Mangel.

Klassifizierung	Beschreibung
UI: 1 ✓	Die Anzeige als wer und mit welchen Rollen man angemeldet ist, wirkt fehl am Platz. Auch bleibt sie beim scrollen nicht in der unteren rechten Ecke. Eine bessere Platzierung wäre beispielsweise in der oberen Leiste.
UI: 1 ✓	Wenn die linke Menüleiste eingeklappt ist wird der Text von "Join Game" weiterhin angezeigt.
UI: 2	Mischung aus Englischem und Deutschem.
UI: 1 ✓	Auf verschiedenen Seiten überlappt der Seiteninhalt beim scrollen die Kopfleiste.
UI: 1	Wenn ein User beispielsweise Topics bearbeitet oder löscht wird das Hub für andere User nicht automatisch auf den neuesten Stand gebracht.
III: 2 ✓ Lobby als Startseite	Die Startseite besteht immer noch aus Text über Spring und dem HelloWorld-Bean. Sie muss folglich dringend neu designed werden.
UI: 2	Das graphische Design ist nicht sehr stimmig. Die Tabellen und Dialoge haben immer noch das Standard Primefaces Design, die Spieleseite passt optisch nicht zum Rest.
UI: 2	Die GUI ist nicht für kleinere Endgeräte (unter 14 Zoll) oder Tiling-window-manager geeignet.
UI: 3	Alle Fehlermeldungen müssen überarbeitet werden. Viele sind ungenau und unpassend.
Null Werte: 3	Siehe Auflistung in Kapitel 3.
Clean code: 1 ✓	Teilweise gibt es auskommentierte Methoden und Annotationen (z.B. <code>UserService:117-157</code>).
Clean code: 2	Immer wieder werden in If-Bedingungen sehr lange und unübersichtliche Statements verwendet. Diese zu verstehen ist sehr aufwändig und macht den Code schlecht lesbar (Siehe z.B. <code>GamePlaySocketController:98</code> , <code>GameStartController:46-47</code>). Hier sollten Variablen verwendet werden.
Clean code: 1	<code>PrintIns</code> , die für das Debugging verwendet wurden, sollten entfernt werden.
Clean code: 2	Javadoc fehlt an manchen Stellen (besonders die Beschreibungen der Klassen), das Format (Großschreibung am Anfang der Beschreibung sowie Punkt am Ende) ist sehr uneinheitlich und man trifft immer wieder auf Rechtschreibfehler. <code>@returns</code> und <code>@params</code> wurden sehr selten beschrieben oder fehlen.
Code organisation: 2	Mangelnde Package-Struktur innerhalb der Klassen-Kategorien (So z.B. <code>.ui.controllers</code>). Hier wären Unterordner angebracht.
Topic JSON: 2 ✓	Wenn man eine falsch formatierte Datei hochlädt wird keine Fehlermeldung angezeigt sondern lediglich eine Meldung, dass die Datei erfolgreich hochgeladen wurde.
Topic/Game: 2	Topics können zwar nicht gelöscht werden wenn sie in einem Spiel sind, jedoch kann man problemlos alle Begriffe löschen.

Delete Cascade: 2		In Fällen wie beispielsweise den Topics müssen child-entities gelöscht werden um die entity selbst löschen zu können. Hier wären delete cascades angebrachter, sprich das Löschen eines topics sollte möglich sein ohne jeden Begriff einzeln löschen zu müssen.
Term Deletion: 2		Wenn ein User eine Topic löscht während dies von einem anderen User bearbeitet wird, tritt bei Letzterem eine unbehandelte Exception auf wenn dieser versucht einen Begriff zu erstellen.
User Deletion: 2		Wenn ein User gelöscht wird, während dieser von einem anderen User bearbeitet wird, tritt bei Letzterem eine unbehandelte Exception auf. Wenn der bearbeitende User dann auf speichern klickt, wird der gelöschte user wieder in der Datenbank abgelegt.
User Deletion: 2	✓	Fehlermeldung dass der User nicht gelöscht werden kann, weil er keine Rolle besitzt.
User Deletion: 2		Gelöschte Spieler scheinen immer noch im Unterpunkt Chat "Who's online?" in der Lobby auf.
User Creation: 1		Bei Fehlern beim Erstellen von Benutzern wird der Dialog stets geschlossen und muss neu geöffnet werden.
User Creation: 3		Wenn man einen neuen Benutzer erstellt und dieser ein Spiel erstellt, kann er niemanden zuweisen, weil es keine Einträge gibt (es erscheinen nur User die man erstellt hat).
Architektur: 2		In Services werden teils Controller verwendet (zB. GameService:74-76. Dies bricht die Schichtenarchitektur und sollte vermieden werden.
API: 1		Die Methode um ein Würfelergebnis an das System zu schicken sollte POST nicht PATCH sein.
API: 1		In der Klasse welche für die POSTs verwendet wird ist ein nie benutztes Attribut "val".
Game: 3	✓	Wenn man mit Usern, die nicht in einem Spiel sind, (z.B. user5) auf "Join Game" klickt tritt eine unbehandelte NullPointerException auf und man wird weder redirected, noch bekommt man eine Fehlermeldung.
Game: 1		Es ist möglich einen Begriff als nicht erraten zu kennzeichnen bevor die Runde durch den abgelaufenen Timer oder den Würfel gestoppt wurde.
Game: 3	prüfen	Es können nicht mehrere Spiel gleichzeitig gestartet werden
Game: 3		BenutzerInnen die auf "Join Game" klicken kommen immer in das gleiche Game, und müssen dafür nicht eingeladen werden.
Game, 1		Die Scoreliste während des Spiels und im Endbildschirm ist nicht nach Punkten der Teams sortiert. Dies wäre vorteilhaft und wenig aufwändig.
Game, 1		Textnachrichten während des Spiels sind nicht auf die aktuelle Rolle (Ratend/Nicht ratend) des Teams personalisiert. So wird zB. die Überschrift "Term:" auch angezeigt, wenn kein Wort zu sehen ist.
Game: 3		Es werden keine Benachrichtigungen / Spieleeinladungen verschickt.
Testdrehbuch: 1		Der Ausgangszustand von "TC 10.2 zu Use-Case Stopprunde" ist auf englisch.
Konzeptbeschreibung: 1		Vererbungen im Usecases Diagramm stimmen mit Umsetzung nicht überein.
Konzeptbeschreibung: 1		Min-Max Notation im Klassendiagramm stimmt mit Umsetzung nicht überein.