

# **Projektdokumentation**

## **1. Release**

**13.05.19 – 09.06.19**

**Universität Kassel  
Software Engineering I SS19  
Gruppe G  
PO: Tobias Klipp, SM: Omar Sood**



# Inhalt

1	Einleitung .....	7
1.1	Das Projektteam.....	7
1.2	Rahmenbedingungen .....	7
1.2.1	Projektorganisation.....	7
1.2.2	Projektziel.....	8
2	Eingesetzte Technologien.....	8
2.1	Dokumentation .....	8
2.1.1	Jira .....	8
2.2	Serverkommunikation .....	8
2.2.1	Funktionsweise REST-Endpunkt .....	8
2.2.2	Funktionsweise von Websockets.....	9
2.3	Entwicklungswerkzeuge .....	10
2.3.1	Java 11 .....	10
2.3.2	Spring Framework.....	11
2.4	Die Benutzeroberfläche .....	12
2.4.1	JavaFX.....	12
2.5	Sonstige .....	12
3	Erster Sprint.....	12
3.1	Vorbesprechung .....	12
3.2	Dokumentation .....	12
3.3	Kundengespräch .....	13
3.4	Erstes Teamtreffen .....	13
3.5	Ziel des Sprints .....	13
3.6	Planung .....	13
3.7	Domain-Stories .....	14
3.7.1	Login des Benutzers .....	14
3.7.2	Beschreibung.....	15
3.7.3	Erstellung von Benutzer .....	16
3.8	Beitritt in die Lobby .....	17
3.8.1	MockUp.....	17
3.8.2	Beschreibung.....	17
3.8.3	Domain Story .....	18
3.8.4	Spiel beitreten.....	19
3.8.5	Erstellen von Spiel .....	20
3.8.6	Abmeldung aus Lobby .....	22
3.8.7	Abmeldung aus Spiel .....	23
3.8.8	Chatten .....	23

3.9	Tasks .....	25
3.9.1	Öffnen des Clients .....	25
3.9.2	Anzeige des Logins .....	26
3.9.3	Login vom Benutzer - Erfolg .....	26
3.9.4	Login vom Benutzer - Fehlschlag .....	27
3.9.5	Anlegen von Benutzer – Erfolgsfall .....	28
3.9.6	Anlegen von Benutzer - Fehlschlag .....	28
3.9.7	Anzeige von Lobby .....	29
3.9.8	Wechseln der Sprache - Optional .....	30
3.9.9	Spiel beitreten.....	30
3.9.10	Spiel erstellen – Erfolgsfall .....	31
3.9.11	Spiel erstellen - Fehlschlag .....	32
3.9.12	Spiel verlassen.....	33
3.9.13	Chatten – Teilnahme am Allgemeinen Chat.....	33
3.9.14	Chatten – Privater Chat .....	34
3.10	Zusammenfassung der ersten Woche .....	35
3.10.1	Burndown-Chart.....	35
3.10.2	Tasks SOLL vs IST .....	36
3.10.3	Aufgetretene Probleme.....	36
3.10.4	Besprechung der ersten Woche .....	36
3.10.5	Chat .....	36
3.10.6	Login .....	36
3.10.7	Registrierung .....	37
3.10.8	Absprachen für die kommende Woche.....	37
3.11	Zusammenfassung der zweiten Woche.....	37
3.11.1	Rücksprache mit dem Kunden zu Feature-Vorschlägen, 20.05.2019.....	37
3.11.2	Absprache am Mittwoch, den 22.05.19.....	38
3.11.3	.....	38
3.11.4	Burndown-Chart.....	39
3.11.5	Kumuliertes Flussdiagramm .....	39
3.11.6	Tasks SOLL vs IST .....	40
3.11.7	Aufgetretene Probleme.....	40
3.11.8	Teamtreffen Freitag 24.05.19 .....	40
4	Zweiter Sprint.....	42
4.1	Ziele des Sprints .....	42
4.2	Planung .....	42
4.3	Tasks .....	42
4.3.1	Core – Anzeigen von Lobby .....	42

4.3.2	Core – Schließen des Clients .....	43
4.3.3	Core – Spiel beitreten .....	43
4.3.4	Core – Spiel verlassen .....	44
4.3.5	Appearance – General .....	44
4.3.6	Appearance – Lobby .....	45
4.3.7	Appearance – Chat .....	45
4.3.8	Appearance – Login .....	45
4.3.9	Optional – Wechsel der Sprache .....	46
4.3.10	Optional – Chat Features: Zitate .....	46
4.3.11	Änderung - Überarbeitung der Fehlermeldungen im Login-Screen .....	47
4.3.12	Appearance - Animierter Titel im Login-Bildschirm .....	47
4.3.13	Appearance - Spielname in Fensterüberschrift .....	48
4.3.14	Appearance - Icon in Spielfenster austauschen .....	48
4.3.15	Optional - Zeitstempel vor Chat-Nachrichten .....	48
4.3.16	Optional – Ladesymbol im Login .....	49
4.4	Zusammenfassung der ersten Woche .....	49
4.4.1	Absprache zum Chat, 26.05.2019 .....	49
4.4.2	Absprache über den Aufbau des Ingame-Fensters, 26.05.2019 .....	49
4.4.3	Feature-Anfrage, 26.05.2019 .....	50
4.4.4	Tasks SOLL vs IST .....	50
4.4.5	Burndown-Chart .....	51
4.4.6	Kumuliertes Flussdiagramm .....	51
4.5	Teamtreffen 28.05.2019 .....	52
4.5.1	Organisatorische Probleme .....	52
4.5.2	Entwicklungsprobleme .....	52
4.5.3	Absprache über Produkt-Beinamen, am 31.05.2019 .....	52
4.5.4	Absprache über Lobby, am 31.05.2019 .....	53
4.5.5	Absprache über die Darstellung des Chats, am 01.06.2019 .....	53
4.6	Zusammenfassung der zweiten Woche .....	54
4.6.1	Tasks SOLL vs IST .....	54
4.6.2	Kumuliertes Flussdiagramm .....	55
4.6.3	Absprache über Darstellung des Login-Screens, 06.06.2019 .....	55
4.6.4	Absprache über Darstellung des Chats, 07.06.2019 .....	56
4.6.5	Absprache über Sortier-Funktion der Spiel-Liste in der Lobby, am 07.06.2019 .....	56
4.6.6	Absprache über Wechsel des Hintergrundbilds im Login und in der Lobby, am 07.06.2019 .....	56
4.6.7	Teamtreffen, Freitag 07.06.2019 .....	56
4.7	Die Client-Software .....	57

4.7.1	Der Login .....	57
4.7.2	Die Lobby.....	58
4.7.3	Der Ingame-Bildschirm.....	59
5	Abbildungsverzeichnis .....	61
6	Tabellenverzeichnis .....	61
7	Literaturverzeichnis.....	62

# 1 Einleitung

Dieses Dokument dient der Dokumentation der Projektarbeit innerhalb des Scheins Softwaretechnik 1 im Bachelor-Studiengang Informatik zum Sommersemester 2019.

## 1.1 Das Projektteam

Das Projektteam der Gruppe G, welches die Anforderungen des Betreuer-Personals, nachfolgend Kunde genannt, erfüllen muss, setzt sich aus den folgenden Personen zusammen:

- Georg Siebert
- Jan Müller
- Juri Lozowoj
- Keanu Stückrad
- Tobias Klipp
- Omar Sood

Das Team arbeitet nach der Scrum Methodik, einer Variante der agilen Softwareentwicklung, die auf kurzweilige Entwicklungsetappen mit iterativ definierten Zielsetzungen basiert. (1) Dabei gilt insbesondere das Paradigma der Kommunikation-über-Dokumentation und der stetigen Absprache und Orientierung an Kundenwünsche statt fixer Vorgaben. (2)

Die Aufteilung des Teams erfolgt dabei in 4 Entwickler, einen Scrum-Master (SM) und einen Product Owner (PO).

Der Scrum-Master stellt dabei sicher, dass die Methoden korrekt eingehalten und das Team möglichst unbeeinträchtigt durch den organisatorischen Rahmen arbeiten kann. Darüber hinaus unterstützt er das Team darin, Kundeninteressen in Softwareinteressen zu übersetzen. Der Product Owner vertritt die Kundeninteressen gegenüber dem Entwicklerteam dient als Schnittstelle zwischen dem Kunden und Team. (1)

Die Besetzung der Posten SM und PO erfolgt dabei für jeweils vier Wochen und wird im Laufe des Projekts gewechselt, sodass jede der genannten Personen einmal die Rolle des PO oder SM's übernimmt.

## 1.2 Rahmenbedingungen

### 1.2.1 Projektorganisation

Die Projektarbeit erfolgt innerhalb des Sommer-Semesters 2019. Der Ablauf wird dabei in vier Abschnitte von jeweils vier Wochen Dauer unterteilt. Jeder dieser monatlichen Blöcke stellt einen Release dar, welcher wiederum in zwei Sprints mit jeweils zwei Wochen Dauer aufgeteilt wird.

Die gestellten Anforderungen des Kunden werden durch den PO in so genannte User-Stories unterteilt, welche in Tasks festgehalten werden. Diese Tasks werden durch den PO im Backlog priorisiert und nachfolgend vom Projektteam gewichtet.

Die Aufgabe des Scrum-Masters ist es, die gegebenen Aufgaben in lösbarer Sub-Tasks zu unterteilen, welche im Rahmen der Projektarbeit vom Entwicklerteam umgesetzt werden.

### **1.2.2 Projektziel**

Ziel des Übungsprojektes ist es eine voll funktionsfähige Client-Software zu schreiben, welche mit dem durch den Kunden bereitgestellten Server interagiert.

Während der Bearbeitung müssen die Anforderungen des Kunden berücksichtigt werden. Die Anforderungen werden zum einen durch ein bereitgestelltes Dokument vorgegeben und zum anderen durch die vorbereiteten MockUps des Projektteams ergänzt und ausgeweitet.

Zusätzlich zu der Entwicklung ist eine Dokumentation des Projektes anzufertigen. Diese soll den Ablauf des Projektes zusammenfassen und dem Kunden somit einen nachträglichen Überblick über die geleistete Projektarbeit ermöglichen.

Am Ende des Releases muss eine Präsentation vorgetragen werden, in der die entwickelte Software vorgestellt wird. Weiterhin sind die eingesetzten Technologien darzulegen und eventuelle Abweichungen von der Planung zu begründen.

## **2 Eingesetzte Technologien**

Nachfolgend werden die Technologien erläutert, welche für den Betrieb des Clients verwendet werden.

### **2.1 Dokumentation**

#### **2.1.1 Jira**

Jira ist eine Vorgangserfassungssystem des Unternehmen Atlassian. Jira bietet eine Jira Software Komponente, die Projekt-Teams darin unterstützt nach Paradigmen der agilen Softwareentwicklung zu arbeiten. Dabei bietet Sie Unterstützung für Kanban sowie für Scrum-Projekte. Alle Schritte der Projektentwicklung werden innerhalb der Software dokumentiert und verwaltet. (3)

Jira bietet die Möglichkeit die vorhandenen Daten anhand von Diagrammen zu visualisieren, um somit detailliertere Auswertungen des Geschehens zu ermöglichen.

### **2.2 Serverkommunikation**

Die Kommunikation mit dem Server erfolgt über zwei Schnittstellen. Einen REST-Endpunkt und einen Websocket. Der REST-Endpunkt regelt sowohl den Login-Vorgang, wie auch die Registrierung. Weiterhin lassen sich Informationen über die aktuell angemeldeten Spieler und die aktiven Spiele über diese Schnittstelle erfragen.

Über den Websocket werden Chat-Nachrichten versendet und empfangen.

#### **2.2.1 Funktionsweise REST-Endpunkt**

REST beschreibt einen Programmierparadigma zur Entwicklung und Nutzung von Webservices basierend auf dem synchronen, Client-initiierten Austausch von http Nachrichten. (4) Einfach ausgedrückt kann man sich vorstellen, dass ein Programm, welches über REST kommuniziert, bei Bedarf eine Seite im Internet öffnen, sich den Text abspeichern und auf diesem Stand arbeitet.



Die Verbindung mit dem REST-Endpunkt wird aktiv vom Client initiiert, der darauf eine Antwort des Servers erhält, wie man es vom aktiven Aufrufen einer Website kennt. Es gibt üblicherweise öffentliche Endpunkte, deren Daten immer abgefragt werden können, sowie gesicherte Endpunkte, deren Daten erst nach Authentifizierung abgerufen werden können.

Im Fall des Projekts muss zur Authentifizierung ein Login eines vorher registrierten Nutzers gesendet werden. In der Antwort des Servers liegt ein Authentifizierungsschlüssel vor, welcher für nachfolgende Anfragen verwendet wird. Der Authentifizierungsschlüssel gewährleistet dabei, dass kein unbefugter Nutzer auf Spiel-Informationen zugreifen kann.

Die Kommunikation wird dabei synchron durchgeführt. Auf eine Anfrage des Clients erfolgt eine Antwort des Servers. Es kann also festgestellt werden, wenn eine Antwort auf eine bestimmte Anfrage nicht eingetroffen ist. Dadurch ist es möglich eine entsprechende Fehlerbehandlung durchzuführen und den Benutzer auf eine fehlgeschlagene Verbindung hinzuweisen. Es ist hingegen nicht einfach möglich mitzubekommen, wenn Änderungen auf dem Server stattfinden, ohne aktiv nach diesen zu fragen.

REST APIs werden daher insbesondere zum Laden initialer oder langlebiger Daten genutzt wie z.B. der anfänglichen Userliste beim Verbinden des Clients.

### **2.2.2 Funktionsweise von Websockets**

Bei dem WebSocketprotokoll handelt es sich im Gegensatz zu dem REST-Endpunkt um ein Kommunikationsprotokoll zur Realisierung von Duplex-Verbindungen. Das heißt, beide Teilnehmer der Verbindung können jederzeit Übertragungen an den anderen initiieren. (5)

Im Gegensatz zum REST Prinzip kann man sich Websockets so vorstellen, dass eine Person eine andere anruft, um sich auszutauschen, und beide am Telefonhörer bleiben und bei Bedarf sprechen, wenn sie etwas mitteilen wollen, solange bis einer von beiden Gesprächspartnern auflegt. Wie bei einem Telefonat muss nicht jede Aussage sofort eine Entgegnung bewirken.

Um sicherzustellen, dass nicht eine Person verbunden bleibt, obwohl die andere gar nicht mehr aktiv am Gespräch teilnimmt, ist es üblich, dass Teilnehmer regelmäßig Lebenszeichen senden, um den anderen zu informieren, dass noch Interesse am Offenhalten der Verbindung besteht.

Der große Vorteil liegt daher genau darin, dass beide Teilnehmer, insbesondere also auch der Server, aktiv ohne Anfrage Daten an den anderen Senden können, sobald diese zur Verfügung stehen. Dadurch ist es anders als bei REST APIs möglich, aktiv Serveränderungen an den Client zu pushen und diesen zu aktualisieren, ohne dass dieser selbst nach Änderungen fragen muss.

Websockets eignen sich insbesondere zum Austausch von Daten, die sich potentiell häufig aktualisieren können, und deren Änderung nicht deterministisch ist, die aber in Quasi-Echtzeit zur Verfügung stehen müssen. Dies ist zum Beispiel beim Chatten der Fall, wo es keine klare Taktung von neuen Nachrichten gibt, der User diese aber sofort sehen möchte.

## 2.3 Entwicklungswerkzeuge

### 2.3.1 Java 11

#### 2.3.1.1 Beschreibung

Zur Entwicklung des Spiele-Clients wird Java in Version 11 verwendet.

Java ist eine objektorientierte Programmiersprache und eingetragene Marke von Oracle, erschienen 1995 und seit März 2019 in Version 12 verfügbar. Java besteht einerseits aus der Sprachdefinition, den Standardbibliotheken und dem Compiler, der den Quellcode in maschinell ausführbare Instruktionen, Byte-Code, übersetzt. Diese sind im Entwicklungspaket (JDK) enthalten. Darüber hinaus zeichnet sich Java insbesondere durch eine Laufzeitumgebung (JRE) aus. Diese muss wie andere Programme auch vom Nutzer, der Java-Software verwenden will, vorinstalliert werden. Die Verwendung einer Java-Laufzeitumgebung statt der direkten Übersetzung in Maschineninstruktionen ermöglicht insbesondere eine höhere plattformunabhängigkeit. (6)

Oracle, das Unternehmen hinter Java, hat in den letzten Jahren begonnen hat, ihren Releasezyklus und ihr Lizenzmodell zu überarbeiten. Der Long-Term-Support-Rythmus für die Laufzeitumgebung wurde verkürzt. Das heißt, dass in kürzeren Zeitspannen aufgehört wird, Aktualisierungen für veraltete Versionen auszuspielen. Dies soll forcieren, dass Standardnutzer ihre Java-Laufzeitumgebung aktuell halten. Im Zuge dessen wurde auch die Verfügbarkeit älterer Java Versionen eingeschränkt.

Dies soll es ermöglichen, schneller neue Features an Nutzer auszurollen und weniger Versionen parallel verwalten zu müssen.

Dieser Entscheidung folgend, wurde sich dazu entschieden das Projekt mit Java 11 zu realisieren.

Zusätzlich gibt es einige neue Sprachfeatures, die die Entwicklung erleichtern.

Der Garbage Collector, der unbenötigten Speicher aufräumt, wurde verbessert, und führt seltener zu kurzzeitigem Einfrieren der Java Programme. Außerdem wurde die sogenannte Stream-API zum Arbeiten auf Datenlisten weiter verbessert.

Darüber hinaus erlaubt die Verwendung von Java 11 eine breitere Auswahl an OpenSource-Projekten, die die Entwicklung unterstützen können.

#### 2.3.1.2 Lizenz

Bis Java 8 war die Verwendung von Java unter Oracle gebührenfrei. Beginnend mit Java 9 wurde das Lizenzmodell geändert. Für den Privatanwender und die Entwicklung bleibt Java von Oracle kostenlos. Für den Produktiveinsatz von Java-Software in Unternehmen, das heißt also das Ausführen von Java Programmen in der Laufzeitumgebung, fallen Lizenzgebühren an, sollte man die Oracle Java Laufzeitumgebung nutzen.

Allerdings ist es nicht nötig, die Oracle Java Laufzeitumgebung zu nutzen. Alternativ kann z.B. auch auf das OpenJDK Projekt zurückgegriffen werden, welches eine weiterhin kostenlose OpenSource Variante der JDKs zur Verfügung stellt und sich sehr eng am Release-Zyklus von Oracle orientiert. In der OpenJDK Variante muss nur die Verwendung einiger besonderen Enterprise-Features für große Unternehmen verzichtet werden. (7) (8)

## **2.3.2 Spring Framework**

### **2.3.2.1 Beschreibung**

Das Spring Framework ist quelloffene Software, welche die Entwicklung von Java-Applikationen, insbesondere JavaEE Anwendungen, anhand von verschiedenen Prinzipien vereinfachen soll. Die Funktionalitäten sind dabei in so genannten Modulen gekapselt, welche bei Bedarf eingebunden werden können. (9)

Der Hauptgrund für die Wahl von Spring ist die Verfügbarkeit eines Containers, in Spring Application Context genannt, welcher den Lebenszyklus von Java-Objekten verwaltet und die für das Programm benötigten Services zur Verfügung stellt.

Dieser Container erleichtert deutlich das Programmierparadigma der Dependency Injection oder auch Inversion of Control.

Diese Programmierparadigma beschreibt das Vorgehen Teilaufgaben einer Softwarekomponente an andere Komponenten auszulagern und diese Abhängigkeiten von außen in einem Konfigurationsschritt zu setzen. Die Alternative ist, die Abhängigkeiten in der Komponente selbst zu definieren, was allerdings bedeutet, dass es schwieriger wird Abhängigkeiten auszutauschen oder anzupassen. Dies wird dann insbesondere im Rahmen von Tests negativ spürbar, da weniger Kontrolle über die Rahmenbedingungen einer Komponente möglich ist.

Durch die Philosophie der Inversion of Control ist es möglich, Einfluss auf die Abhängigkeiten einer Softwarekomponente zu nehmen. (10)

Je mehr man so den Quellcode in einzelne Komponenten zerlegt, umso mehr bedeutet dies allerdings auch, dass man sehr viele Abhängigkeiten zusammensetzen muss, um die gewünschte Funktionalität sicher zu stellen.

Solcher Code wird insbesondere sehr schnell repetitiv, weshalb es Projekte gibt, die die Aufgabe des korrekten Zusammenbauens benötigter Softwarekomponenten auf Basis einfacher Konfigurationsanweisungen übernehmen. Eine solche ist der Application Context von Spring.

Alternative OpenSource Bibliotheken, die diese Aufgabe übernehmen, sind Dagger (11) und Google Guice (12), welche beide von Google verwaltet werden.

Diese Alternativen boten allerdings nach der subjektiven Einschätzung des Teams einen schwereren Zugang. Durch die weitere Verbreitung von Spring sind auch Erklärungen und Hilfen zahlreich verfügbar.

Darüber hinaus liefert Spring eine kuratierte Quelle von weiteren Abhängigkeiten zur Lösung von Standardproblemen, insbesondere im Rahmen der Serverkommunikation, welche ebenfalls entsprechend weitere Verbreitung und damit zusammenhängend umfangreiche Dokumentation vorweisen. (13)

Die beiden primären Gründe für die Auswahl des Frameworks sind die Aktualität der vorhandenen Bibliotheken und die Verwendung von „Dependency Injection“. Diese ermöglicht es Objekte zentral zu verwalten, sodass diese nicht in festen Abhängigkeiten definiert werden müssen. Dies soll für eine saubere Struktur des Codes sorgen.

### **2.3.2.2 Lizenz**

Das Spring-Framework ist unter der Apache-Lizenz veröffentlicht und darf somit kostenfrei für kommerzielle Zwecke genutzt werden. (14)

## 2.4 Die Benutzeroberfläche

Die Benutzeroberfläche, nachfolgend UI (User Interface), genannt stellt die Schnittstelle zwischen Benutzer und Rechner dar. Über Sie werden alle Aktionen mit der Software gesteuert und Informationen an den Nutzer übermittelt.

Für die Programmierung der UI findet JavaFX Verwendung.

### 2.4.1 JavaFX

#### 2.4.1.1 Beschreibung

JavaFX ist ein Framework zur Programmierung von graphischen Oberflächen, welches durch das Unternehmen Oracle entwickelt wurde.

Anwendungen, welche mit JavaFx geschrieben werden, durchlaufen einen Lebenszyklus. Dieser beginnt mit dem Aufruf der Anwendung. Dabei wird das Java-Programm gestartet und eine so genannte Stage (Bühne) erstellt. Diese Stage stellt das Fenster dar, in dem die weiteren grafischen Elemente der Anwendung angezeigt werden. In Anlehnung an eine Theateraufführung existieren weiterhin so genannte Scenes (Szenen). Eine Stage kann eine oder mehrere Scenes enthalten, wobei die Scenes die eigentlichen grafischen Elemente beinhalten. Die Zuordnung erfolgt dabei mittels einer Baumdatenstruktur. An der Basis befindet sich das Szenenobjekt, welches hierarchisch gegliederte Kind-Knoten zugewiesen bekommt.

Wird die Anwendung geschlossen, so werden entsprechende Funktionalitäten aufgerufen, welche für einen ordnungsgemäßen Schließvorgang der Anwendung sorgen.

#### 2.4.1.2 Lizenz

JavaFX steht unter der GPL Lizenz und darf somit für die Programmierung von kommerziellen Programmen verwendet werden. (15)

## 2.5 Sonstige

Darüber hinaus verwendete Bibliotheken zur Erleichterung der Entwicklung können im Projektarchiv eingesehen werden.

## 3 Erster Sprint

### 3.1 Vorbesprechung

Die Vorbesprechung des Projekts fand am 07.05.2019 statt. Dort wurden die folgenden Abläufe der anstehenden Projektarbeit abgestimmt.

### 3.2 Dokumentation

Die Dokumentation soll in zweifacher Ausführung erfolgen. Zum einen soll der aktive Arbeitsablauf mit dem Projektmanagement Programm Jira erfasst werden, zum anderen ist eine schriftlich ausgearbeitete Dokumentation des Projektfortgangs anzulegen. Diese muss im Anschluss an die Projektarbeit ausgehändigt werden.

Das Projekt wird in 4 Releases mit jeweils 2 Sprints, welche einen Zeitraum von jeweils zwei Wochen umfassen, eingeteilt.

### **3.3 Kundengespräch**

Das initiale Kundengespräch wurde am 10.05.19 abgehalten. Dort wurden die Mock-Ups präsentiert. Der Kunde legte insbesondere auf die Struktur der Lobby wert, welche nicht aus zu vielen einzelnen Teilbereichen aufgebaut sein soll. Weiterhin wurde der Wunsch geäußert zusätzlich zu den normalen Nachrichten weitere Funktionalitäten, wie Spieleinladungen und Emotes, in den Client integriert zu bekommen.

### **3.4 Erstes Teamtreffen**

Im ersten Teamtreffen, welches ebenfalls am 10.05.19 stattfand, wurde der Ablauf des Projektstarts abgestimmt. Insbesondere wurden dabei die Storypunkte für die anstehenden Aufgaben ausgepokert und den Team-Mitgliedern zur Bearbeitung zugewiesen.

### **3.5 Ziel des Sprints**

Im ersten Sprint werden die Basisfunktionalitäten des Clients implementiert. Dazu gehören die Anmeldung und Registrierung am Server, der Chat-Mechanismus, sowie das Anzeigen und Navigieren innerhalb der Lobby.

Der Kunde soll nach dem ersten Sprint in der Lage sein, einen Eindruck über das Layout der UI zu erhalten und die Bedienung der Software zu beurteilen.

### **3.6 Planung**

Der erste Sprint erfolgt im Zeitraum vom 13.05.19 bis zum 24.05.19.

Zur Bearbeitung der Anforderungen stehen 5 Programmierer zur Verfügung.

Die Tasks des Sprints werden in Core-Features und Optionale-Features unterteilt. Die Core-Funktionalitäten entsprechen den Vorgaben des Kunden, während die optionalen Komponenten die erforderlichen Features erweitern, bzw. das Portfolio ergänzen.

Ein Story-Point entspricht einer vollständigen Zeitstunde geplanter Werkszeit.

Task	Story-Points	Bearbeiter
Öffnen des Clients	1	Jan Müller
Anzeige des Logins	2	Jan Müller
Login vom Benutzer - Erfolg	3	Keanu Stückrad
Login vom Benutzer - Fehlschlag	2	Keanu Stückrad
Anlegen von Benutzer - Erfolg	3	Juri Lozowoj
Anlegen von Benutzer - Fehlschlag	3	Juri Lozowoj
Anzeige von Lobby	4	Georg Siebert
Optional: Wechseln der Sprache	3	Georg Siebert
Spiel beitreten	7	Christoph Hanauer
Spiel erstellen - Erfolg	3	Juri Lozowoj
Spiel erstellen - Fehlschlag	1	Juri Lozowoj
Spiel verlassen / Schließen des Clients	3	Keanu Stückrad
Chatten	7	Jan Müller
$\Sigma$	42	

Tabelle 1: Planung des ersten Sprints

## 3.7 Domain-Stories

### 3.7.1 Login des Benutzers

#### 3.7.1.1 MockUp



Abbildung 1: Mockup - Login von Benutzer

## 3.7.2 Beschreibung

Der Benutzer soll sich über eine Login-Maske am Spiel anmelden können. Hierzu muss er sich mit dem Benutzernamen und einem Passwort authentifizieren.

Im Login-Screen ist es möglich zwischen den Sprachen Deutsch und Englisch zu wechseln.

### 3.7.2.1 Domain Story

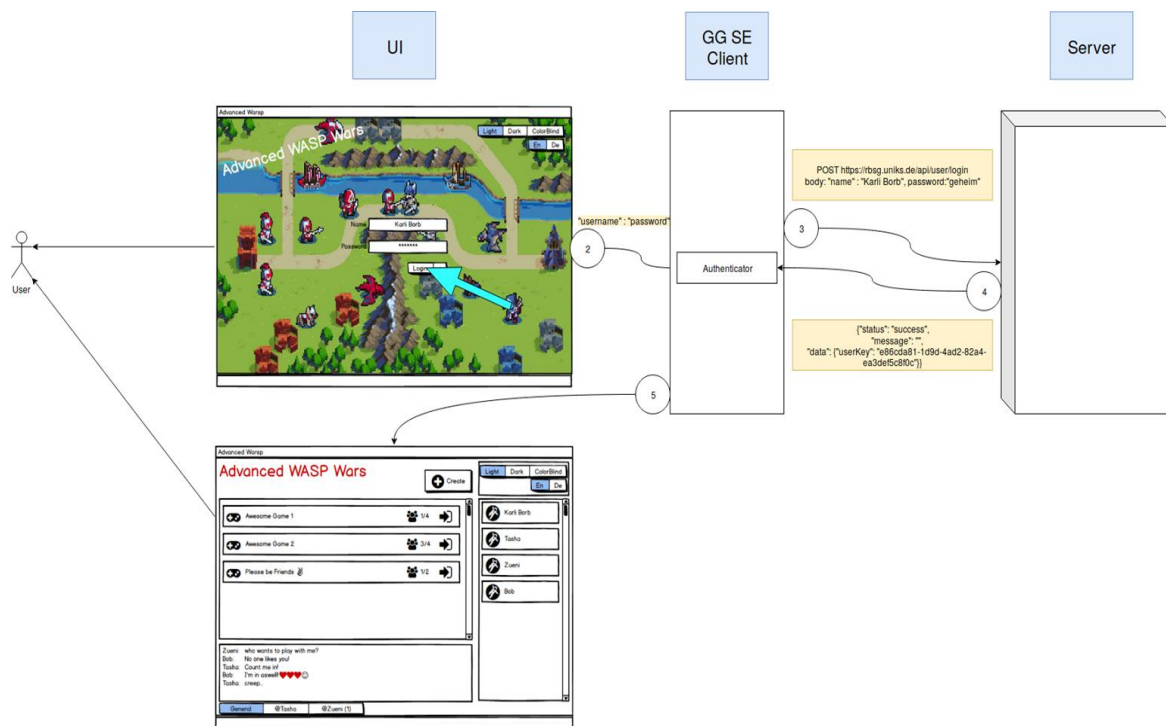


Abbildung 2: Domain Story - Login

Der Benutzer Karli Bob besitzt bereits einen Account für das RBSG System. Er möchte sich einloggen.

1.) Karli Bob gibt seinen Usernamen und sein Passwort in die vorgesehene Maske ein und klickt auf Login.

2.) Der Client sendet eine REST-Anfrage an den RBSG-Server, um den Benutzer zu authentifizieren.

3.) Der RBSG-Server antwortet mit einer entsprechenden Erfolgsmeldung und einem user key, den sich die Applikation für alle weiteren Interaktionen der Sitzung mit dem Server speichert.

4.) Der Client aktualisiert das UI und leitet den Spieler in die Lobby weiter.

### 3.7.3 Erstellung von Benutzer

#### 3.7.3.1 Beschreibung

Der Benutzer soll im Login-Bildschirm ein neues Spieler-Konto erstellen können.

#### 3.7.3.2 Domain Story

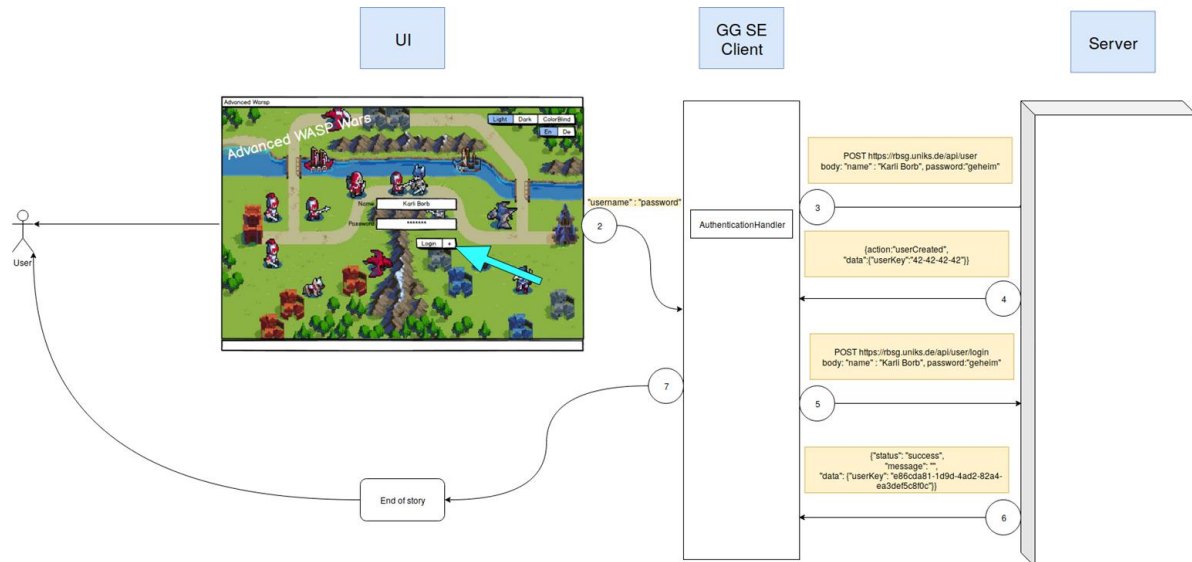


Abbildung 3: Domain Story - Erstellung von Benutzer

Karli Borb besitzt keinen Account für das RBSG System und möchte sich registrieren.

- 1) Karli Borb sieht den Login Screen, er gibt einen Benutzernamen und ein Passwort an und klickt auf +, um sich neu beim RBSG-Projekt zu registrieren.
- 2) Der Client sendet eine REST-Anfrage an den RBSG-Server, um den neuen Benutzer zu erstellen.
- 3) Der RBSG-Server antwortet auf die Anfrage mit einer Erfolgsmeldung.
- 4) Der Client sendet nachfolgend automatisch eine weitere Anfrage, um den nun registrierten Benutzer am Server anzumelden.
- 5) Die folgenden Schritte entsprechen dem Prozess wie in der User Story Anmelden beschrieben.



## 3.8 Beitritt in die Lobby

### 3.8.1 MockUp

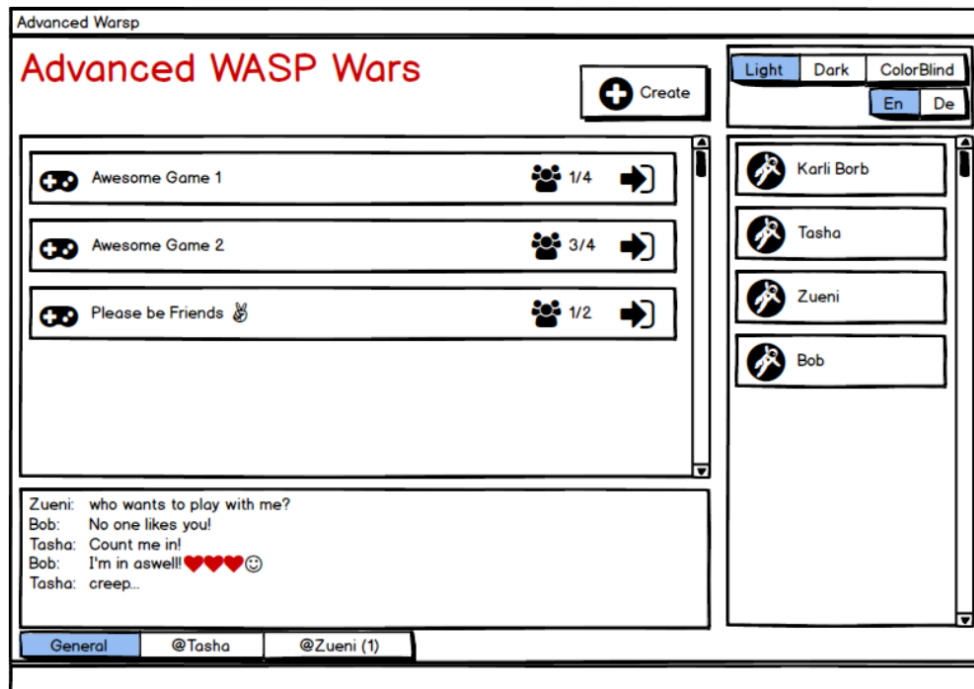


Abbildung 4: Die Lobby

### 3.8.2 Beschreibung

Der Benutzer soll nach dem erfolgreichen Anmeldevorgang in eine Lobby weitergeleitet werden. Innerhalb der Lobby soll es möglich sein ein Spiel zu erstellen, die angemeldeten Spieler, den Chat und die Liste der laufenden Spiele einzusehen.

### 3.8.3 Domain Story

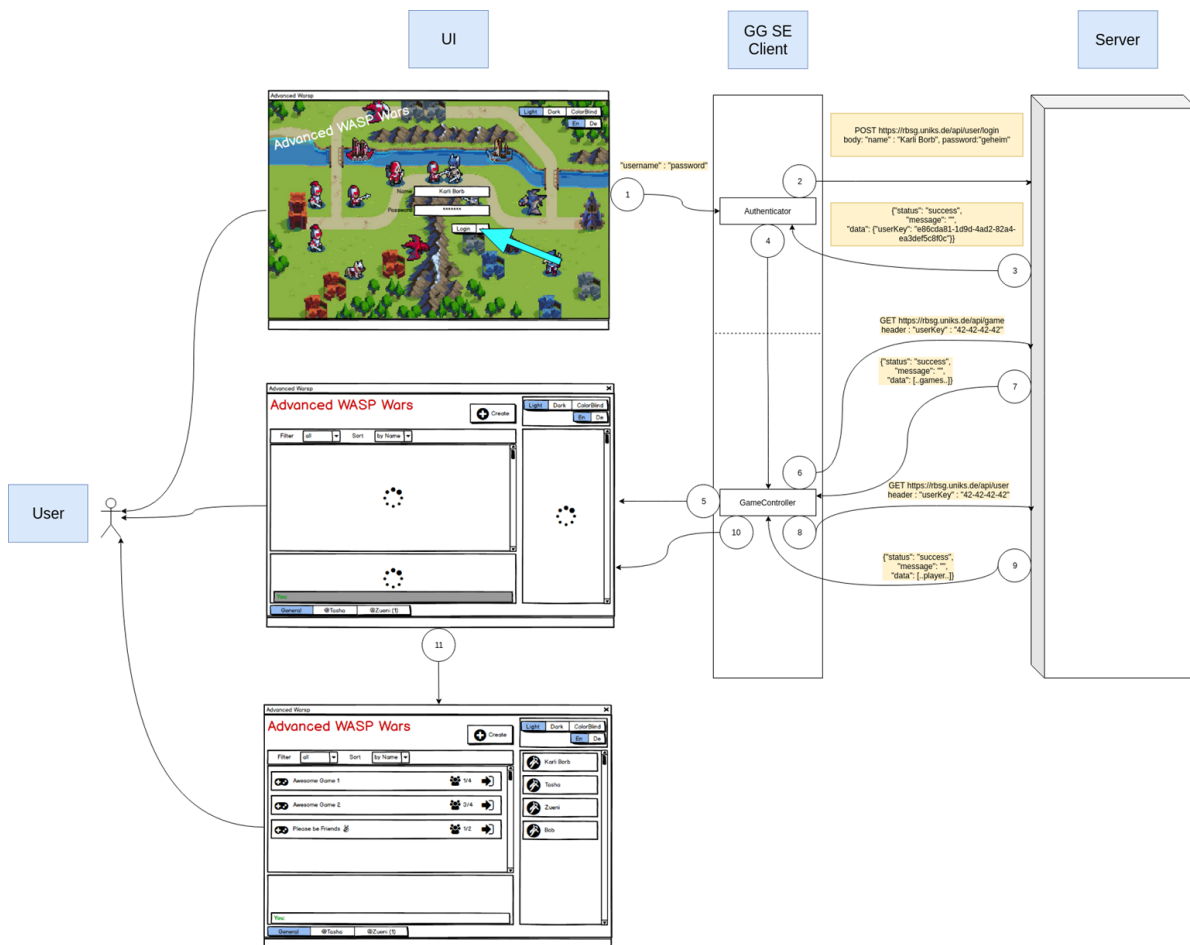


Abbildung 5: Domain Story - Anzeige der Lobby

Karli Borb ist ein registrierter Benutzer, der sich einloggt um in die Lobby zu gelangen.

- 1) Siehe in der Domain Story Login
- 2) Siehe in der Domain Story Login
- 3) Siehe in der Domain Story Login
- 4) Übergang zur Lobby
- 5) Der Client aktualisiert das UI und bringt den angemeldeten Benutzer in die Lobby.
- 6) Der Client sendet eine REST Anfrage an den RBSG Server, um eine Liste aller aktuellen Spiele abzufragen. Dabei überträgt er den userKey des angemeldeten Spielers.
- 7) Der RBSG Server sendet dem Client eine Liste aller aktuellen Spiele.

8) Der Client sendet eine REST Anfrage an den RBSG Server, um eine Liste aller aktuell angemeldeten Spieler abzufragen. Dabei überträgt er den userKey des angemeldeten Spielers.

9) Der RBSG Server sendet eine Liste aller angemeldeten Spieler.

10) Der Client aktualisiert das User Interface mit den aktuellen Spielen und angemeldeten Benutzern.

11) Der Client aktiviert den Chat.

### 3.8.4 Spiel beitreten

#### 3.8.4.1 Beschreibung

Der Spieler kann in der Lobby eine Liste der laufenden Spiele einsehen. Durch eine Interaktion mit der UI kann er einem Spiel beitreten.

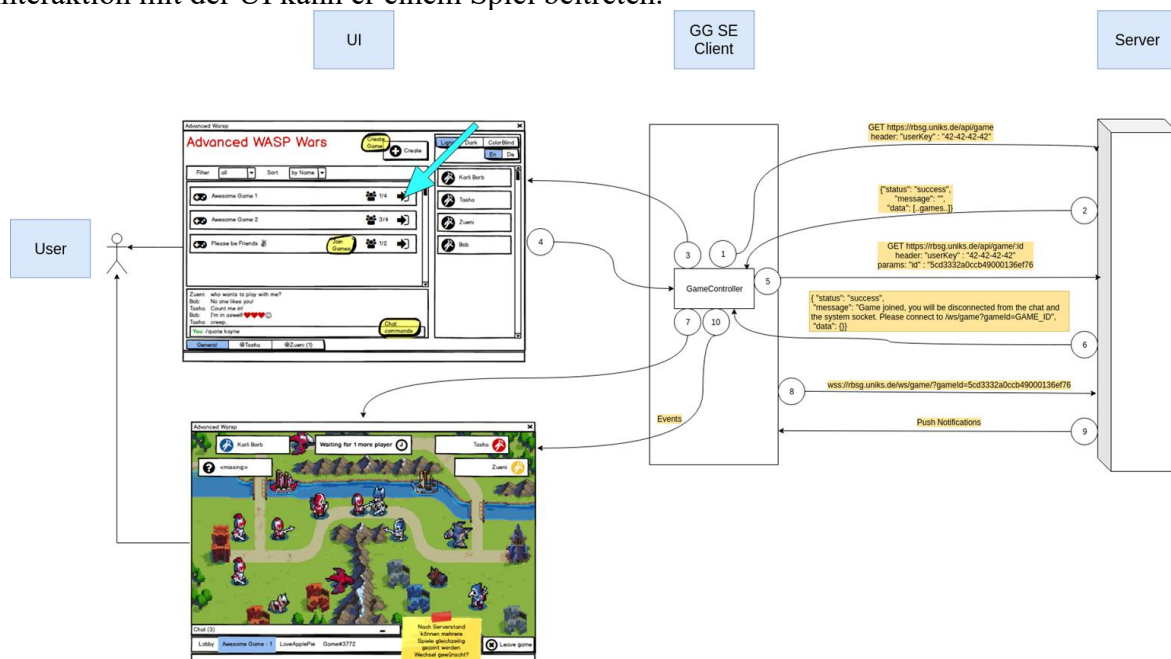


Abbildung 6: Domain Story - Beitritt zu offenem Spiel

#### 3.8.4.2

#### 3.8.4.3 Domain Story

Karli Borb ist ein angemeldeter Benutzer. Er möchte einem offenen Spiel beitreten.

- 1) Der Client stellt eine Anfrage nach den offenen Spielen an den Server.
- 2) Der Server antwortet mit einer Liste der laufenden Spiele.
- 3) Der Client aktualisiert das UI mit den neuen Informationen.
- 4) Karli Borb wählt eines der offenen Spiele aus der Liste aus.

- 5) Der Client sendet eine REST-Anfrage an den RBSG-Server, um den angemeldeten Benutzer dem ausgewählten Spiel beitreten zu lassen.
- 6) Der Server quittiert dies mit einer Erfolgsmeldung.
- 7) Der Client leitet den Benutzer zur Spielansicht weiter.
- 8) Der Client verbindet sich über einen Websocket mit dem RBSG-Server, um nachfolgend Events über das gewählte Spiel zu empfangen.
- 9) Der Server übermittelt erste Spielinformationen an den Client.
- 10) Der Client aktualisiert das UI mit dem aktuellen Spielzustand.

### 3.8.5 Erstellen von Spiel

#### 3.8.5.1 Mockup

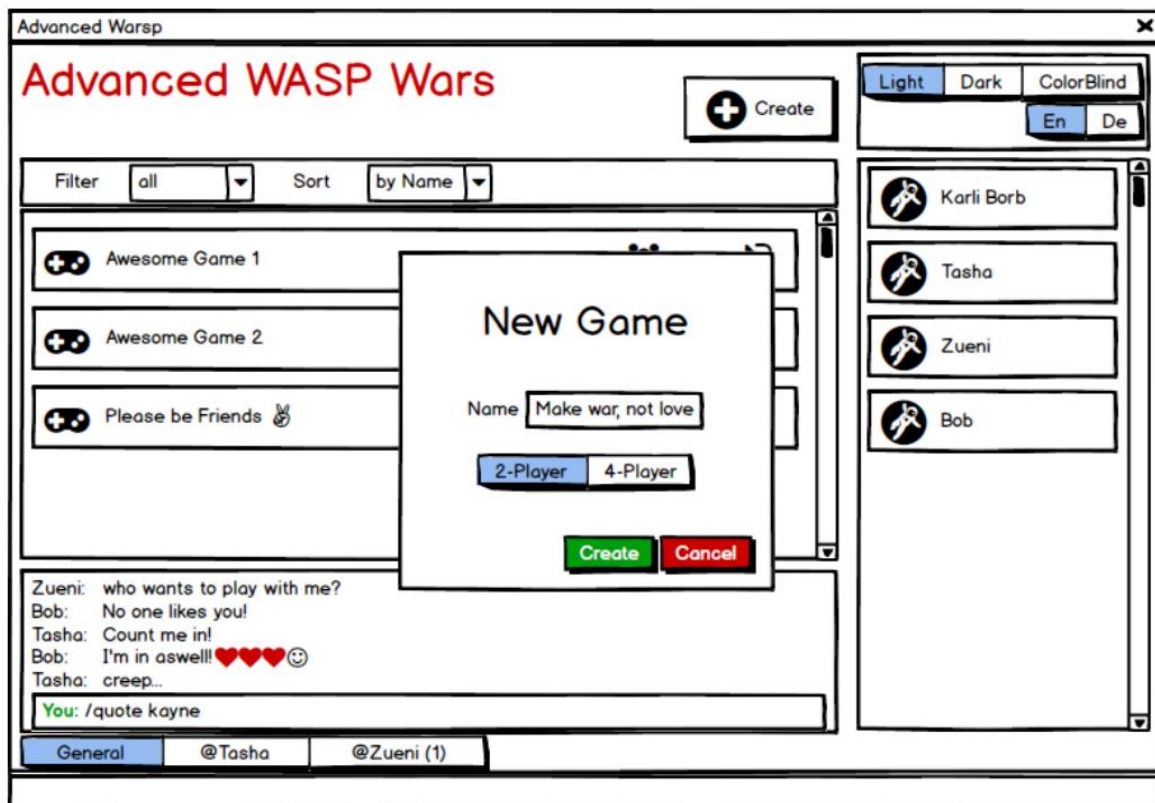


Abbildung 7: Mockup - Erstellen von Spiel

#### 3.8.5.2 Beschreibung

Der Benutzer soll innerhalb der Lobby neue Spiele erstellen können. Zur Erstellung eines Spiels werden der Name und die Anzahl der Spieler abgefragt. Die Eingaben können nachfolgend entweder bestätigt, oder verworfen werden.

### 3.8.5.3 Domain Story

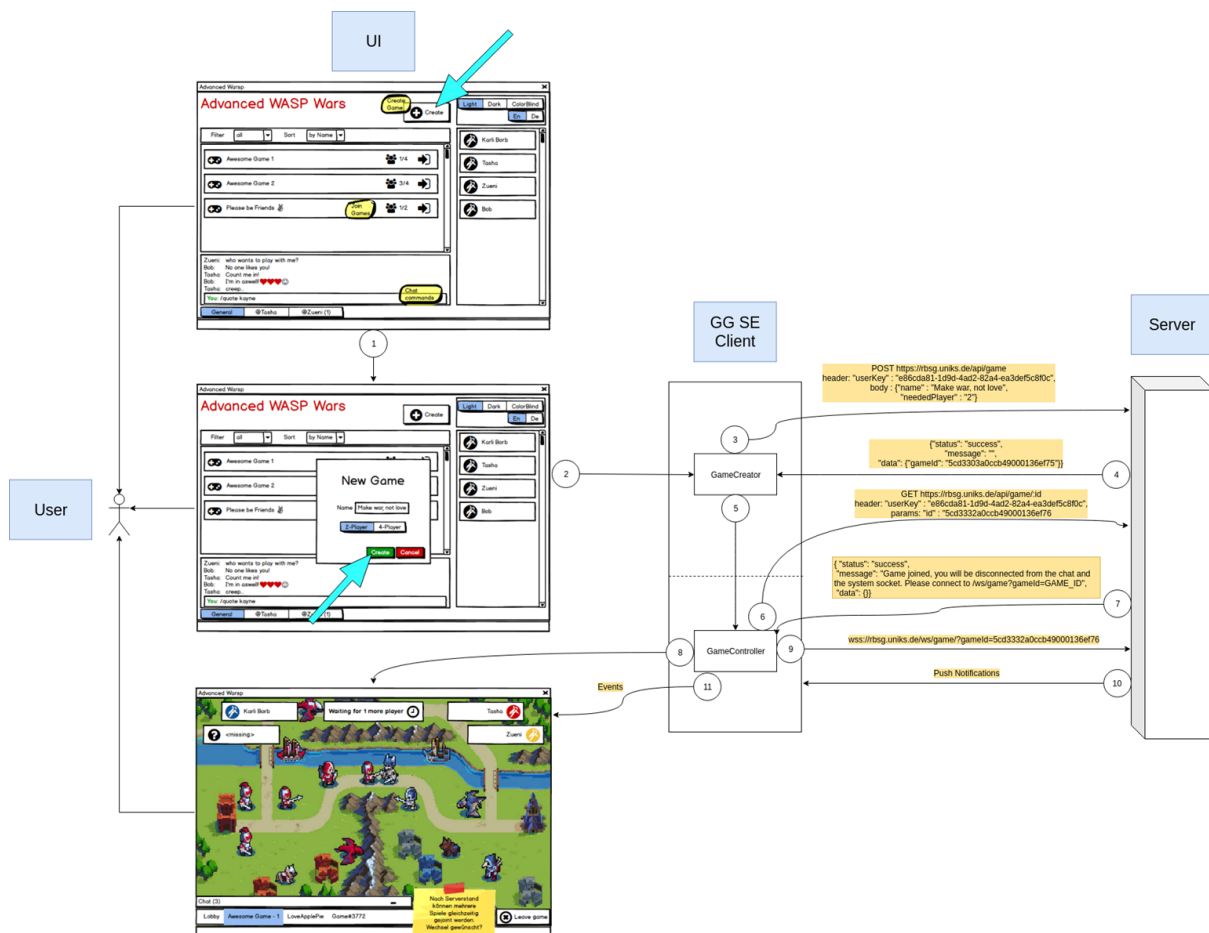


Abbildung 8: Domain Story - Erstellen von Spiel

Karli Borb ist ein angemeldeter Benutzer. Er möchte ein Spiel erstellen.

- 1) Karli Borb bestätigt den Button zum Erstellen eines neuen Spiels, trägt den gewünschten Namen und die Spieleranzahl an und bestätigt dies.
- 2) Der GameCreator wird beauftragt, ein neues Spiel zu erzeugen mit den angegebenen Daten.
- 3) Der GameCreator sendet eine REST Anfrage zum Erstellen eines neuen Spiels an den RBSG-Server im Namen des angemeldeten Benutzers.
- 4) Der Server quittiert dies mit einer Erfolgsmeldung und der Spiel-ID.
- 5) Der GameCreator stellt eine erfolgreiche Erstellung fest und leitet automatisch das Beitreten des neu erstellten Spiels ein.
- 6) Der weitere Ablauf entspricht der Domain Story „Spiel beitreten“

## 3.8.6 Abmeldung aus Lobby

### 3.8.6.1 Beschreibung

Ein angemeldeter Spieler soll beim Verlassen der Lobby vom Spiel-Server abgemeldet werden.

### 3.8.6.2 Domain Story

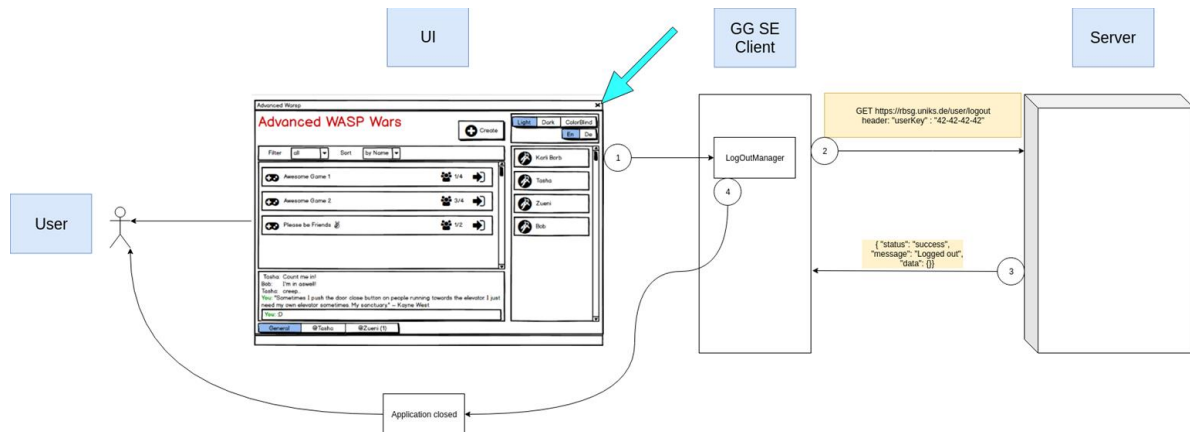


Abbildung 9: Domain Story - Abmeldung in der Lobby

Der Benutzer befindet sich in der Lobby.

- 1) Der Benutzer betätigt den „Schließen-Knopf“
- 2) Der Client sendet eine Anfrage an den Spiel-Server
- 3) Der Server bestätigt die Abmeldung
- 4) Der Client schließt die Anwendung

## 3.8.7 Abmeldung aus Spiel

### 3.8.7.1 Beschreibung

Die Abmeldung aus einem laufenden Spiel erfolgt nahezu analog zur Abmeldung aus der Lobby.

### 3.8.7.2 Domain Story

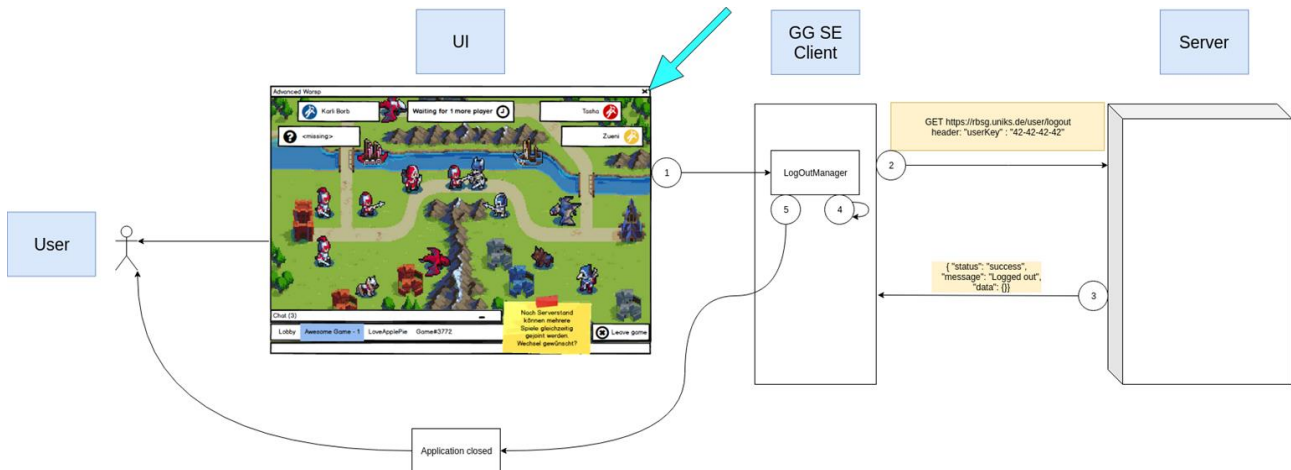


Abbildung 10: Domain Story - Abmeldung aus dem Spiel

Der Benutzer befindet sich in einem laufenden Spiel.

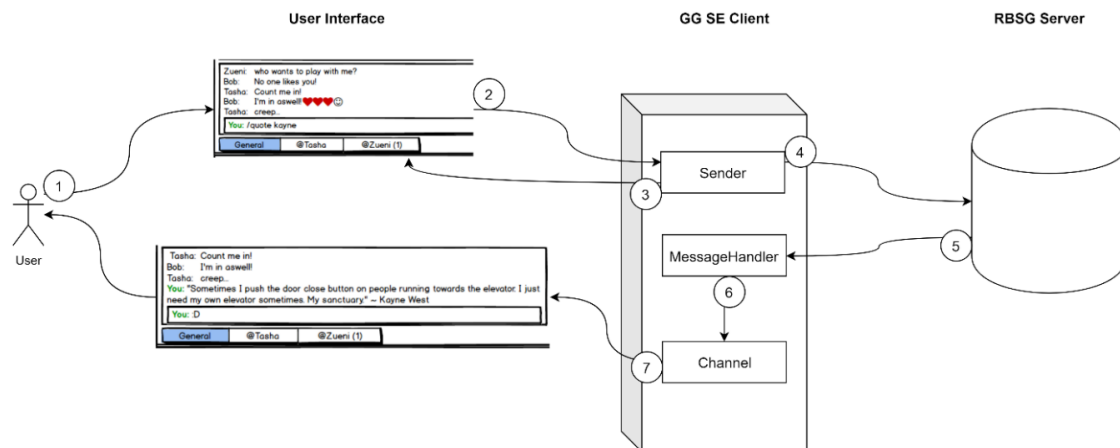
- 1) Analog zu Abmeldung aus Lobby
- 2) Analog zu Abmeldung aus Lobby
- 3) Analog zu Abmeldung aus Lobby
- 4) Der Client schließt die geöffneten Verbindungen zum Server(Sockets)
- 5) Der Client schließt die Applikation

## 3.8.8 Chatten

### 3.8.8.1 Beschreibung

Der Benutzer soll innerhalb der Lobby an einem öffentlichen Chat teilnehmen können. Jeder angemeldete Spieler kann den öffentlichen Chat einsehen.

### 3.8.8.2 Domain Story



**Abbildung 11: Domain Story - Chatten**

Benutzer1 ist ein angemeldeter Benutzer der Applikation. Benutzer1 möchte eine Nachricht an alle anderen Benutzer im Chat senden.

- 1) Benutzer1 gibt den einen Text in das Chatfenster für öffentliche Nachrichten ein und bestätigt dies mit Enter.
- 2) Das Chateingabefenster blockiert und leitet die Chatnachricht weiter an den Sender des Clients.
- 3) Der Sender sendet die Nachricht mittels Websocket an den RBSG Server.
- 4) Der Sender informiert das User Interface über die Verarbeitung, welches für neue Nachrichten freigegeben wird.
- 5) Der RBSG Server informiert den MessageHandler über eine neu eingegangene Nachricht.
- 6) Der MessageHandler wählt den öffentlichen Chat aus und aktualisiert das Chatfenster mit der neuen Nachricht.
- 7) Benutzer1 und alle anderen Chatteilnehmer sehen die neue Chatnachricht.



## 3.9 Tasks

### 3.9.1 Öffnen des Clients

#### 3.9.1.1 Anforderungen

Der Client muss durch eine ausführbare .jar Datei durch den Benutzer geöffnet werden können.

#### 3.9.1.2 Zeit SOLL

Die geplante Zeit beträgt 0,5 Stunden.

#### 3.9.1.3 Mock-Up



Abbildung 12: Mockup - Login Bildschirm

#### 3.9.1.4

#### 3.9.1.5 User-Story

Der Benutzer öffnet den Client über eine Desktop-Verknüpfung. Es wird ein Fenster geöffnet, welches das Login-Fenster mit der Eingabemaske zeigt.

#### 3.9.1.6 Position innerhalb der Domain-Story

Der Task steht in Verbindung mit den Tasks Login und Registrierung. Beide Nachfolger sind abhängig von der Fertigstellung dieser Aufgabe.

#### 3.9.1.7 Zeit IST

Die Abarbeitungszeit des Vorgangs betrug 2 Stunden und 20 Minuten.

#### 3.9.1.8 Abweichungen

Die Abweichungen entstanden durch die Einarbeitungszeit des Entwicklers in JavaFX und dem Umsetzen des Programms innerhalb einer ausführbaren .jar Datei.

## **3.9.2 Anzeige des Logins**

### **3.9.2.1 Anforderungen**

Der Benutzer muss nach dem Öffnen des Clients eine Login-Maske angezeigt bekommen, an der er sich anmelden, oder registrieren kann.

### **3.9.2.2 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurde 1 Stunde geplant.

### **3.9.2.3 User-Story**

Der Benutzer hat die Anwendung geöffnet. Er sieht die Login-Maske. Diese besteht aus textuellen Eingabefeldern für den Namen und für das Passwort. Weiterhin werden unter dem rechten Ende des Passwort-Feldes ein Login Feld und ein Nutzer-erstellen Feld (Wird als "+" dargestellt) angezeigt.

### **3.9.2.4 Position innerhalb der Domain-Story**

Der Task steht in Verbindung mit der Implementierung des Login- und Registrierungsmechanismus. Beide Aufgaben sind direkt abhängig von der Implementierung der Login-Maske.

### **3.9.2.5 Abweichungen**

Die Abweichungen entstanden durch die benötigte Einarbeitungszeit des Entwicklers in JavaFX.

## **3.9.3 Login vom Benutzer - Erfolg**

### **3.9.3.1 Anforderungen**

Der Benutzer muss sich über die zur Verfügung gestellte Anmeldemaske am Server authentifizieren können. Nach der Anmeldung muss eine Lobby für den weiteren Ablauf bereitgestellt werden.

### **3.9.3.2 Zeit SOLL**

Für die Implementierung des Login-Vorgangs wurden 1,5 Stunden eingeplant.

### **3.9.3.3 User-Story**

Der Benutzer befindet sich im Login-Bildschirm. Er gibt seine Benutzerdaten (Name:Passwort) in die Anmeldemaske ein.

Nach der Eingabe betätigt er den "Login"-Knopf. Er wird bei erfolgreicher Rückmeldung des Servers in die Lobby weitergeleitet.

### **3.9.3.4 Position innerhalb der Domain-Story**

Der Login Vorgang ist ein zentraler Bestandteil im Ablauf der User-Aktionen. Alle Client-Mechanismen, welche einen aktiven user-key benötigen sind direkt abhängig von diesem Task.

### **3.9.3.5 Zeit IST**

Die Bearbeitung des Vorgangs hat insgesamt 3 Stunden und 31 Minuten in Anspruch genommen und wurde in der ersten Woche des ersten Sprints fertiggestellt.

### **3.9.3.6 Abweichungen**

Der Task wurde im ersten Sprint nicht abgeschlossen. Weiterhin wurde der Task durch die Einarbeitungszeit für die Serverkommunikation verzögert.

## **3.9.4 Login vom Benutzer - Fehlschlag**

### **3.9.4.1 Anforderungen**

Bei einem Fehlschlag im Authentifizierungsablauf muss eine Fehlerbehandlung zur Verfügung stehen.

### **3.9.4.2 Zeit SOLL**

Für die Implementierung der Fehlerbehandlung wurden 2 Stunden eingeplant.

### **3.9.4.3 User-Story**

Der Benutzer befindet sich im Login-Bildschirm. Er hat seine Benutzerdaten eingetippt und auf den Login Knopf gedrückt. Der Login schlägt fehl.

1.) Wegen falschen Benutzerdaten:

Ein Pop-Up Fenster öffnet sich und es wird die Fehlermeldung "Fehler: Falsche Anmeldedaten" angezeigt.

2.) Wegen Verbindungsproblemen:

Nach 10 Sekunden ohne Rückmeldung vom Server wird die Fehlermeldung "Fehler: Es konnte keine Verbindung zum Server hergestellt werden." angezeigt.

### **3.9.4.4 Position innerhalb der Domain-Story**

Der Login Vorgang ist ein zentraler Bestandteil im Ablauf der User-Aktionen. Alle Client-Mechanismen, welche einen aktiven user-key benötigen sind direkt abhängig von diesem Task.

### **3.9.4.5 Zeit IST**

Die Bearbeitung des Vorgangs hat zum aktuellen Zeitpunkt 4 Stunden und 30 Minuten in Anspruch genommen.

In der zweiten Woche des ersten Sprints wurden 5 Stunden und 30 Minuten für die Bearbeitung in Anspruch genommen.

Insgesamt wurde die Aufgabe bis zu ihrem Abschluss 10 Stunden bearbeitet.

### **3.9.4.6 Abweichungen**

Die Abweichungen sind mit der Einarbeitungszeit des Programmierers in die Serverkommunikation und der Implementierung eines allgemeinen Umgangs mit Fehlern begründet.

### 3.9.5 Anlegen von Benutzer – Erfolgsfall

#### 3.9.5.1 MockUp



Abbildung 13: Mockup - Benutzerdaten

#### 3.9.5.2

#### 3.9.5.3 Anforderungen

Ein nicht registrierter Benutzer muss sich über den Client an dem RBSG System registrieren können.

#### 3.9.5.4 Zeit SOLL

Für die Bearbeitung des Vorgangs wurden 1 Stunde und 30 Minuten eingeplant.

#### 3.9.5.5 User-Story

Der Benutzer befindet sich im Login-Bildschirm. Die gewünschten Login-Daten wurden in die dafür vorgesehenen Felder eingetragen. Durch das betätigen des Registrierungs-Knopfes werden die erfassten Informationen an der Server weitergeleitet.

Bei einer erfolgreichen Registrierung wird der Benutzer in die Lobby weitergeleitet.

#### 3.9.5.6 Position innerhalb der Domain-Story

Das Anlegen des Benutzers ist direkt abhängig von der Anzeige des Lobby-Bildschirms und der Kommunikation mit dem Server.

#### 3.9.5.7 Zeit IST

Die Bearbeitungszeit der Aufgabe betrug 1 Stunde.

#### 3.9.5.8 Abweichungen

Der Vorgang wurde durch die Implementierung von Tests und dem Einarbeiten in die Funktionsweise der Serverkommunikation verzögert.

### 3.9.6 Anlegen von Benutzer - Fehlschlag

#### 3.9.6.1 Anforderungen

Ein nicht registrierter Benutzer muss sich über den Client an dem RBSG System registrieren können. Bei einem Fehlschlag der Registrierung muss eine entsprechende Fehlerbehandlung zur Verfügung stehen.

#### 3.9.6.2 Zeit SOLL

Für die Bearbeitung des Vorgangs wurden 1 Stunde und 30 Minuten eingeplant.

### 3.9.6.3 User-Story

Der Benutzer befindet sich im Login-Bildschirm.

Er hat seine Anmeldedaten eingegeben.

Der Benutzer betätigt den Registrierungsknopf.

Bei erfolgloser Registrierung (der Nutzer existiert bereits) wird dem Benutzer ein Pop-Up Fenster mit der Nachricht "Fehler: Der Nutzer existiert bereits." angezeigt.

Gibt der Server nach 10 Sekunden keine Rückmeldung, so wird die Fehlermeldung: "Fehler: Keine Verbindung zum Server möglich." in einem Pop-Up Fenster angezeigt,

### 3.9.6.4 Position innerhalb der Domain-Story

Für den Login-Vorgang ist eine funktionierende Fehlerbehandlung unerlässlich. Der Benutzer muss über vorhandene Eingabe und Verbindungsfehler aufgeklärt werden. Insbesondere der Beitritt zur Lobby und die Teilnahme an Spielen hängt von einem fehlerfreien Login-Vorgang ab.

### 3.9.6.5 Abweichungen

Der Task wurde in der eingeplanten Zeit abgeschlossen.

## 3.9.7 Anzeige von Lobby

### 3.9.7.1 MockUp

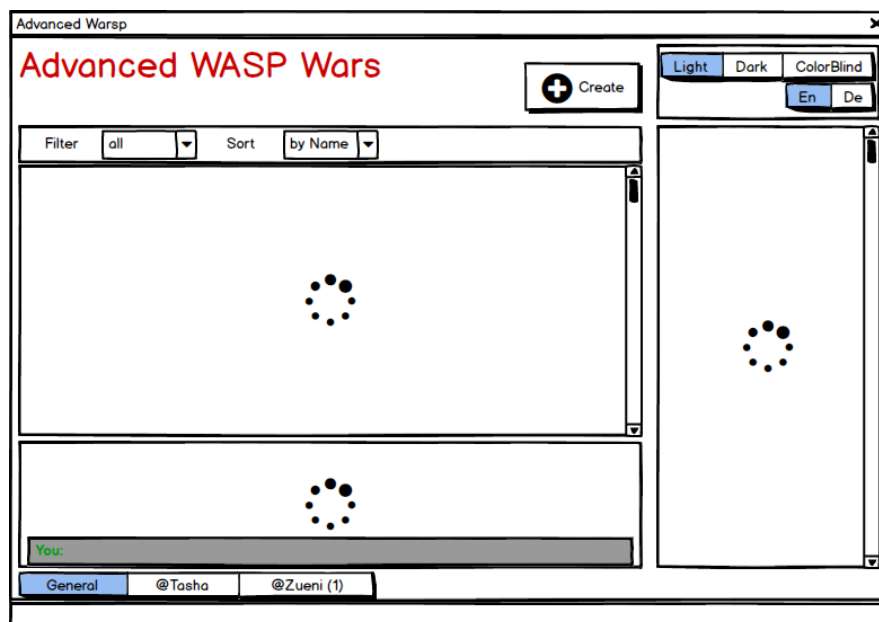


Abbildung 14: Mockup - Lobby

### 3.9.7.2 Anforderungen

Nach dem der Login erfolgt ist muss der Nutzer in eine Lobby weitergeleitet werden.

Innerhalb der Lobby ist eine Übersicht über die laufenden Spiele, die angemeldeten Spieler und den Chat erforderlich.

### **3.9.7.3 Zeit SOLL**

Für die Bearbeitung wurden 4 Stunden eingeplant.

### **3.9.7.4 User-Story**

Der Benutzer hat sich erfolgreich eingeloggt. Die Lobby wird angezeigt. Solange keine Daten vom Server empfangen werden, werden die entsprechenden Felder mit einem "Lade"-Symbol markiert.

Wurden die Daten erhalten, so werden die Felder aktualisiert.

### **3.9.7.5 Position innerhalb der Domain-Story**

Die Lobby ist der zentrale Bestandteil des Clients. Alle Aufgaben, die den Chat und das Spiel betreffen sind direkt abhängig von dieser Aufgabe.

### **3.9.7.6 Abweichungen**

Der Task wurde im ersten Sprint nicht abgeschlossen. Die Verzögerung der Bearbeitung wurde durch die Einarbeitungszeit in JavaFX, die Serverkommunikation, Jackson und WebSockets verzögert. Zusätzlich wurde der Aufwand für das Schreiben von Tests in der Planung unterschätzt.

## **3.9.8 Wechseln der Sprache - Optional**

### **3.9.8.1 Anforderungen**

Der Benutzer kann zwischen der deutschen und englischen Sprache wechseln.

### **3.9.8.2 Zeit SOLL**

Für die Bearbeitung wurden 3 Stunden eingeplant.

### **3.9.8.3 User-Story**

Der Benutzer befindet sich im Login-Bildschirm, oder der Lobby. In der rechten oberen Ecke wird eine Auswahl für den Wechsel der UI-Sprache angezeigt. Zur Auswahl stehen Englisch und Deutsch.

Wählt der Benutzer eine Sprache über einen der vorhandenen Knöpfe aus, wird das Feld der ausgewählten Sprache farblich hervorgehoben und der vorhandene Text im Login-Screen, oder der Lobby angepasst.

### **3.9.8.4 Position innerhalb der Domain-Story**

Das Wechseln der Sprache ist ein optionales Feature. Es steht jedoch in direkter Abhängigkeit zu jedem vorhandenen Textfeld und muss im Laufe des Projekts fortwährend weitergeführt werden.

### **3.9.8.5 Abweichungen**

Der Task wurde im ersten Sprint nicht bearbeitet.

## **3.9.9 Spiel beitreten**

### **3.9.9.1 Anforderungen**

Der Benutzer soll aus der Lobby heraus einem offenen Spiel beitreten können.

### 3.9.9.2 Zeit SOLL

Für die Bearbeitung der Aufgabe wurden 7 Stunden eingeplant.

### 3.9.9.3 User-Story

Der Benutzer befindet sich in der Lobby.

Er sieht die Liste der offenen Spiele.

Der Benutzer klickt auf das gewünschte Spiel.

Es öffnet sich ein Pop-Up Fenster mit dem Text: "Wollen Sie dem Spiel beitreten?" und der Auswahl "Ok", "Abbrechen".

Wurde "Ok" gewählt, so wird der Benutzer zum Spiel-Bildschirm weitergeleitet.

### 3.9.9.4 Position innerhalb der Domain-Story

Alle Funktionen, die innerhalb des Spiels vorhanden sind, sind direkt abhängig von dem Beitritt zum Spiel.

### 3.9.9.5 Abweichungen

Der Vorgang konnte in der zweiten Woche des ersten Sprints nicht abgeschlossen werden, da eine Abhängigkeit zum Lobby-Task besteht, welcher ebenfalls noch in Bearbeitung war. Er wird deshalb in den zweiten Sprint übernommen.

## 3.9.10 Spiel erstellen – Erfolgsfall

### 3.9.10.1 MockUp

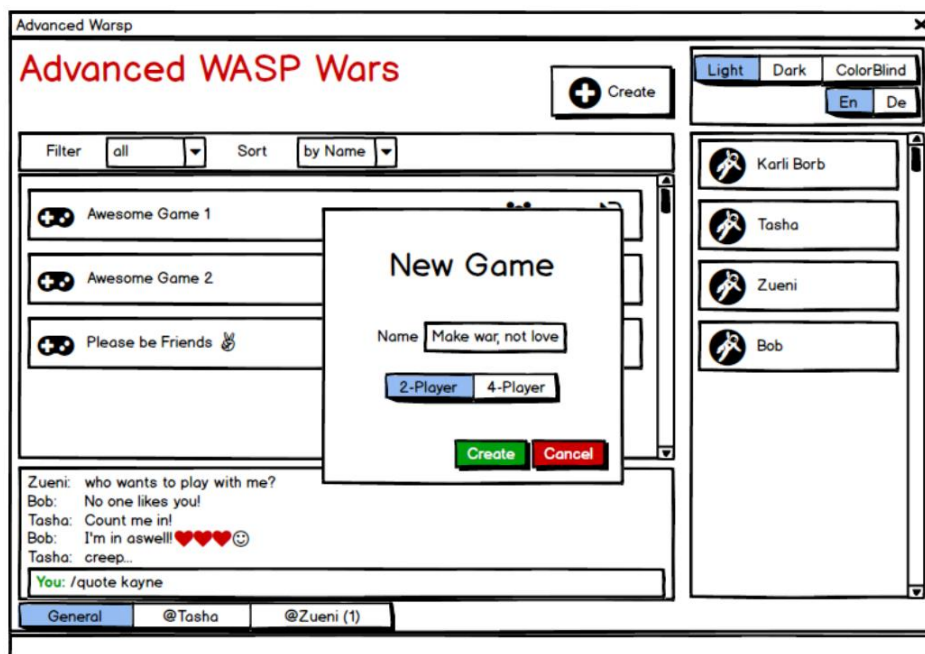


Abbildung 15: Mockup - Spiel erstellen

### 3.9.10.2 Anforderungen

Der Benutzer muss in der Lage sein innerhalb der Lobby ein neues Spiel zu erstellen.

### **3.9.10.3 Zeit SOLL**

Für die Bearbeitung des Vorgangs wurden 3 Stunden eingeplant.

### **3.9.10.4 User-Story**

Der Benutzer befindet sich in der Lobby. Er betätigt den "Spiel erstellen" Knopf.

Es öffnet sich das "Spiel erstellen" Pop-Up Feld.

Dort wird das Feld "Spielname" und die Auswahl "2-Spieler" und "4-Spieler" angezeigt.

(Vorsicht: Die Anforderung ist nicht ganz klar. Evtl. müssen im nächsten Release n-Spieler unterstützt werden)

Hat der Benutzer die Informationen eingetragen und bestätigt dies mit dem Knopf "Spiel erstellen", so wird er zum Spielbildschirm weitergeleitet.

### **3.9.10.5 Position innerhalb der Domain-Story**

Der Registrierungsvorgang ist unerlässlich für die Bedienung des Clients. Ohne ein valides Benutzerkonto kann der Nutzer die Funktionalitäten des Clients nicht benutzen.

### **3.9.10.6 Abweichungen**

Der Task wurde durch das Schreiben von Tests und der Überarbeitung des Designs verzögert.

## **3.9.11 Spiel erstellen - Fehlschlag**

### **3.9.11.1 Anforderungen**

Tritt bei der Erstellung eines Spiels ein Fehler auf, so muss dieser dem Benutzer kenntlich gemacht werden. Weiterhin muss ihm die Möglichkeit weiterer Interaktionen geboten werden, um den Programmablauf fortzusetzen.

### **3.9.11.2 Zeit SOLL**

Für die Bearbeitung des Vorgangs wurde 1 Stunde eingeplant.

### **3.9.11.3 User-Story**

Der Benutzer befindet sich in der Lobby und hat den "Spiel erstellen" Knopf betätigt.

Das "Spiel erstellen" Pop-Up Fenster wird angezeigt. Nach der Bestätigung der Eingabe müssen folgende Fehler abgefangen werden:

1.) Keine Verbindung zum Server möglich

Wird nach der Bestätigung des Spiels innerhalb von 10 Sekunden keine Rückmeldung erhalten, so öffnet sich ein Pop-Up Fenster mit der Fehlermeldung "Fehler: Keine Verbindung zum Server möglich." im Vordergrund. Der Benutzer kann das Fenster mit "Ok" verlassen. Danach wird erneut die Lobby angezeigt.

2.) Abbruch durch den Spieler

Drückt der Benutzer den Knopf "Abbrechen" so wird das "Spiel erstellen" Fenster geschlossen und der Benutzer bekommt erneut die Lobby angezeigt.

3.) Fehlender Informationen



Wurde eines der Felder nicht korrekt ausgefüllt, so öffnet sich nach der Betätigung des "Spiel erstellen" Knopfes ein Pop-Up Fenster mit der Nachricht "Fehler: Fehlerhafte Eingabeinformationen".

#### **3.9.11.4 Abweichungen**

Der Task wurde erst in der zweiten Woche des ersten Sprints abgeschlossen. Die Verzögerung ist durch die unterschätzte Arbeitszeit zu begründen.

### **3.9.12 Spiel verlassen**

#### **3.9.12.1 Anforderungen**

Der Benutzer soll ein laufendes Spiel verlassen können.

#### **3.9.12.2 Zeit SOLL**

Für die Bearbeitung des Vorgangs wurden 3 Stunden eingeplant.

#### **3.9.12.3 User-Story**

Der Benutzer befindet sich in einem laufenden Spiel.

Er möchte Das Spiel verlassen.

Er betätigt den Spiel-verlassen Knopf in der oberen rechten Ecke des Bildschirms.

Ein Pop-Up Fenster mit der Nachricht "Spiel verlassen?" und den Optionen "Ok" und "Abbrechen" erscheint.

Nach der Bestätigung wird der Benutzer in die Lobby weitergeleitet.

#### **3.9.12.4 Zeit IST**

Für die Bearbeitung des Tasks wurde im ersten Sprint 1 Stunde verwendet.

#### **3.9.12.5 Abweichungen**

Der Task wurde im ersten Sprint weder angefangen, noch fertiggestellt.

### **3.9.13 Chatten – Teilnahme am Allgemeinen Chat**

#### **3.9.13.1 MockUp**

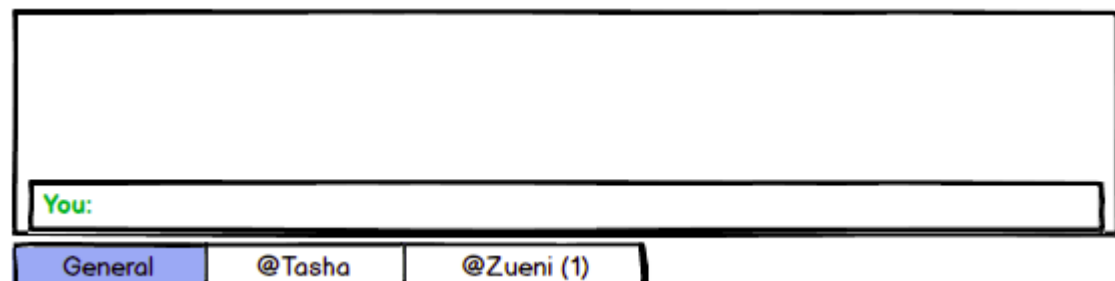


Abbildung 16: Mockup - Chat

#### **3.9.13.2 Anforderungen**

Der Benutzer des Clients muss mittels des Chats in die Lage versetzt werden mit anderen Spielern in Kontakt zu treten.

### **3.9.13.3 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurden 7 Stunden eingeplant.

### **3.9.13.4 User-Story**

Der Benutzer befindet sich in der Lobby.

Im unteren linken Bereich der Lobby befindet sich das Chatfenster. Der Chat-Bereich besteht aus einem Allgemeinen-Chat und mehreren privaten Chats. Im allgemeinen Chat werden alle Nachrichten angezeigt, die seit dem Betreten der Lobby versendet wurden.

### **3.9.13.5 Position innerhalb der Domain-Story**

Der Chat stellt die zentrale Schnittstelle zwischen den Spielern dar. Ohne Ihn ist keine soziale Interaktion möglich.

### **3.9.13.6 Zeit IST**

Für die Bearbeitung wurden 7 Stunden geplant und 14 Stunden benötigt. Dies entspricht einer Verzögerung von 7 Stunden. Die Aufgabe wurde in der zweiten Woche des ersten Sprints abgeschlossen.

### **3.9.13.7 Abweichungen**

Die Story Points wurden am 14. Mai nachgetragen, dadurch entstand ein Ausschlag im Burndown Chart.

Die Verzögerung der Bearbeitung ist durch die unterschätzte Zeit in der Planung zu begründen.

## **3.9.14 Chatten – Privater Chat**

### **3.9.14.1 Anforderungen**

Der Benutzer des Clients muss mittels des Chats in die Lage versetzt werden mit anderen Spielern innerhalb eines privaten Chats in Kontakt zu treten.

### **3.9.14.2 Zeit SOLL**

-

### **3.9.14.3 User-Story**

Der Benutzer befindet sich in der Lobby.

Er sieht die Liste der angemeldeten Spieler.

Der Benutzer öffnet den Chat und tippt "/w <Benutzername>" ein.

In der Chatleiste öffnet sich ein neuer Reiter für den neuen Cha

### **3.9.14.4 Position innerhalb der Domain-Story**

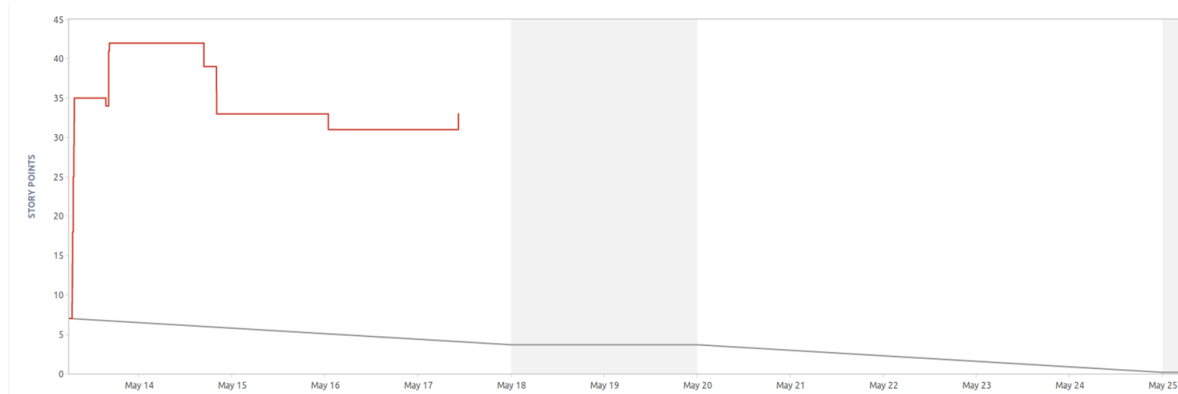
Der Chat stellt die zentrale Schnittstelle zwischen den Spielern dar. Ohne Ihn ist keine soziale Interaktion möglich.

### 3.9.14.5 Abweichungen

Der Task wurde fälschlicherweise bereits im Rahmen des Chat-Task abgearbeitet.

## 3.10 Zusammenfassung der ersten Woche

### 3.10.1 Burndown-Chart



**Abbildung 17: Burndown Chart - 1. Woche 1. Sprint**

In dem Burndown-Chart der ersten Woche des ersten Sprints erkennt man, dass die Aufgaben langsamer als geplant bewältigt und zwei Tasks zur Laufzeit verändert wurden.

Der Ausschlag im Diagramm am 14. Mai entstand durch einen Planungsfehler. Der Task „Chatten“ war zwar zu Beginn vorhanden des Sprints vorhanden, hatte jedoch keine geschätzten Story-Points. Durch das Nachtragen der genannten Punkte änderte sich die Gesamtanzahl der Schätzungen und damit der Verlauf des Graphen.

Der zweite geänderte Task „Anzeige des Login“ wurde am 14. Mai geschlossen, musste aber aufgrund von Änderungswünschen am 17. Mai wiedereröffnet werden.

Insgesamt wurden in der ersten Woche 8 Story-Points und damit 8 Arbeitsstunden abgearbeitet.

Die weiteren Verzögerungen wurden vor allem durch die Fehlschätzung der benötigten Zeit und Schwierigkeiten bei der Bearbeitung hervorgerufen. Weiterhin entstand ein großer Zeitbedarf durch die benötigte Einarbeitungszeit der Programmierer und dem Schreiben von Tests. Beide Punkte wurden in der Schätzung nicht ausreichend berücksichtigt.

### 3.10.2 Tasks SOLL vs IST

Die nachfolgende Tabelle stellt die geplante Zeit der tatsächlich geleisteten Zeit gegenüber.

Task	Zeit SOLL	Zeit IST	Fertig
Öffnen des Clients	1	2,2	x
Anzeige des Logins	2	2,5	x
Login vom Benutzer - Erfolg	3	3,5	-
Login vom Benutzer - Fehlschlag	2	1	-
Anlegen von Benutzer - Erfolg	3	10	x
Anlegen von Benutzer - Fehlschlag	3	-	-
Anzeige von Lobby	4	-	-
Optional : Wechseln der Sprache	3	-	-
Spiel beitreten	7	-	-
Spiel erstellen - Erfolg	3	-	-
Spiel erstellen - Fehlschlag	1	-	-
Spiel verlassen / Schließen des Clients	3	-	-
Chatten	7	-	-
<b>Σ Stunden</b>	<b>42(11)</b>	<b>19,2</b>	
<b>Offene Stunden</b>	<b>31</b>		

Tabelle 2: Zwischenstand der 1. Woche 1. Sprint

### 3.10.3 Aufgetretene Probleme

Innerhalb der ersten Projektwoche konnte ein Entwickler die Arbeit nicht fortsetzen. Dadurch verzögerte sich der Task „Spiel beitreten“ und musste von einem anderen Mitarbeiter übernommen werden.

### 3.10.4 Besprechung der ersten Woche

#### 3.10.4.1 Erstellung der Lobby

Der Websocket Client wurde programmiert. Die Entwicklung wurde durch die Einarbeitungszeit verzögert.

### 3.10.5 Chat

Die Basisfunktionen des Chats wurden implementiert. Die Entwicklung wurde durch die Einarbeitungszeit verzögert.

### 3.10.6 Login

Die vorhandenen Textfelder müssen sichtbar gemacht werden. Der Szenenwechsel zwischen Login und Lobby konnte nicht implementiert werden, da der Lobby Task geblockt war. Zur Fortführung der Aufgabe muss eine weitere Einarbeitung in JavaFX erfolgen.

### **3.10.7 Registrierung**

Die Registrierung wurde implementiert. Um wiederholende Anfragen an den Server zu vermeiden, wurde damit begonnen ein Server-Mock zu schreiben, welcher die Antworten des echten Servers simuliert. Die zur Verfügung stehende Login-Methode wurde noch nicht in den Registrierungs-Code aufgenommen.

### **3.10.8 Absprachen für die kommende Woche**

Nach der Besprechung der ersten Woche wurden folgende Verbesserungsvorschläge und Anmerkungen für die zweite Woche notiert:

- a) Notebooks und Präsentation müssen für das Teamtreffen vorbereitet werden.
- b) Bei der Fragerunde während des Teamtreffens sollen keine „Was hast du gemacht?“ Fragen gestellt werden. Vielmehr soll von den Entwicklern frei erzählt werden, welche Probleme aufgetreten sind, wie diese gelöst wurden und welche Verbesserungsvorschläge es für die Arbeit anderer Mitarbeiter gibt.
- c) Die Betreuer sind nur passive Beobachter des Teamtreffens und sollen nicht aktiv eingebunden werden.
- d) Um optionale Features nachzufragen soll eine Mail an den Kunden versendet werden.
- e) Das zweite Release muss am Montag zum neuen Release-Beginn vom Team selbst erstellt werden. Der neue Release kann nur erstellt werden, sobald der alte Release abgeschlossen ist.
- f) Es soll eine Absprache mit dem Kunden getroffen werden, um das Hintergrundbild abzustimmen.
- g) In die Dokumentation soll aufgenommen werden, welche Technologien eingesetzt werden, und wie sichergestellt wird, dass die Anwendung korrekt arbeitet?
- h) Die Tasks im Jira sollen priorisiert werden.
- i) MockUps und Server Responses sollen an die Tasks angehängt werden.
- j) Voraussetzungen für die Software sollen in die README aufgenommen werden.
- k) Wenn Java 11 nicht installiert wurde und auf die .jar Datei gedrückt wird, wird die Client Software nicht ausgeführt.

## **3.11 Zusammenfassung der zweiten Woche**

### **3.11.1 Rücksprache mit dem Kunden zu Feature-Vorschlägen, 20.05.2019**

Nachdem verschiedene Feature-Vorschläge gesammelt wurden, wurden dies am 19.05.2019 dem Kunden vorgestellt.

Die vorgeschlagenen Features waren:

- (1) Wechsel zwischen den Sprachen Deutsch und Englisch
- (2) Spieleinladungen, welche aus Chat-Nachrichten abgeleitet werden
- (3) Hintergrundmusik
- (4) Einen Actionlog-Channel im Spiel
- (5) Zitate, welche über einen Chat-Befehl generiert werden können

Alle genannten Features wurden genehmigt. Weiterhin wurde ein Vorschlag zum Einbinden der Unsplash API vom Kunden getätigt, diese wird jedoch nicht eingebunden, da dort keine Bilder zur Verfügung stehen, die im Kontext des Spiels zu sehen sind.

### **3.11.2 Absprache am Mittwoch, den 22.05.19**

In diesem Termin sollte die Zusammenführung der verschiedenen Komponenten abgesprochen werden. Als Resultat des Termins wurde eine Überarbeitung der UI eingeplant, welche nachfolgend mit dem Kunden abgestimmt werden sollte.

Die Überarbeitung enthält dabei folgende Änderungsvorschläge:

- 1) Einführung eines einheitlichen Dark-Themes über alle Teilabschnitte des Clients
- 2) Entfernen des Hintergrundbilds und Ersetzung durch einen dunklen Hintergrund entsprechend dem Dark-Theme
- 3) Einfügen von Spiel-Titel und Beinamen im Login-Screen, wobei der Beiname animiert sein soll.
- 4) Einfärbung von Chat: Jeder Benutzername im Allgemeinen-Chat soll in einer eigenen Farbe dargestellt werden.
- 5) Hinweis auf neue Nachrichten im Chat, durch farbliche Markierung des Tabs
- 6) Bereitstellung eines einheitlichen Erscheinungsbilds durch Erstellung von Vorlage.

### 3.11.3 Burndown-Chart

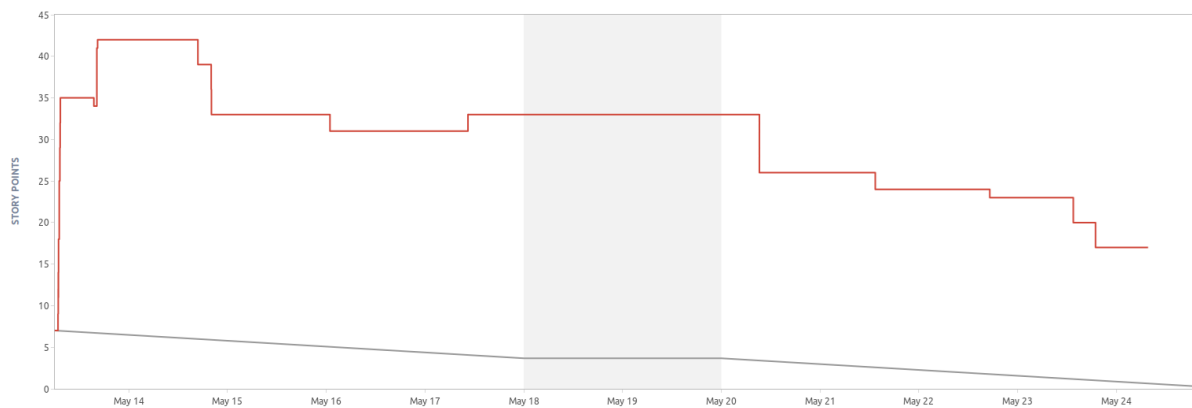


Abbildung 18: Burndown Chart - 2. Woche 1. Sprint

In dieser Woche wurden 16 Story Points abgearbeitet. Dies ist ein Zuwachs von 8 Punkten im Vergleich zur ersten Woche. Weiterhin offen sind 17 Story-Points, welche in den 2. Sprint übernommen werden müssen.

Im Gegensatz zur ersten Woche ist eine durchgehende Abnahme von Tasks zu erkennen. Dies ist durch die Erfahrungen aus der ersten Woche zu begründen.

Die verbliebenen Aufgaben konnten aufgrund von längeren Einarbeitungs- und Bearbeitungszeiten, in Kombination mit vorhandenen Abhängigkeiten untereinander nicht wie geplant abgeschlossen werden.

### 3.11.4 Kumuliertes Flussdiagramm

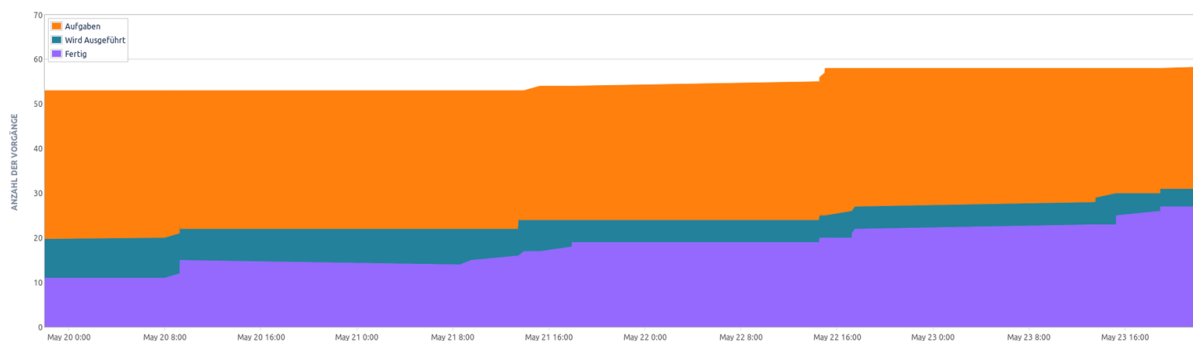


Abbildung 19: Flussdiagramm - 2. Woche 1. Sprint

Aus dem Diagramm lässt sich entnehmen, dass die vergebenen Tasks zum größten Teil bearbeitet wurden und nur ein kleiner Teil der zugewiesenen Arbeit noch in Bearbeitung ist. Weiterhin ist ersichtlich, dass ca. 50% der geplanten Tasks fertiggestellt wurden und somit eine glatte Teilung zwischen den Sprints stattfindet.

### 3.11.5 Tasks SOLL vs IST

Task	Zeit SOLL	Zeit IST	Fertig
Core - Öffnen des Clients	1	2,33	x
Core - Anzeige des Logins	2	2,5	x
Core - Login vom Benutzer - Erfolg	3	3,5	-
Core - Login vom Benutzer - Fehlschlag	2	10	x
Core - Anlegen von Benutzer - Erfolg	3	10	x
Core - Anlegen von Benutzer - Fehlschlag	3	2,133	x
Core - Anzeige von Lobby	4	14,5	-
Optional - Wechseln der Sprache	3	-	-
Core - Spiel beitreten	7	0,5	-
Core - Spiel erstellen - Erfolg	3	12,72	x
Core - Spiel erstellen - Fehlschlag	1	1,5	x
Core -Spiel verlassen / Schließen des Clients	3	-	-
Core - Chatten	7	14	x
<b>Σ Stunden</b>	<b>42</b>	<b>73,68</b>	
<b>Offene Stunden</b>	<b>4</b>		

**Tabelle 3: Zwischenstand 2. Woche 1. Sprint**

Der Tabelle ist zu entnehmen, dass 6 Tasks nicht abgeschlossen wurden. An zwei der offenen Aufgaben wurde bereits gearbeitet, die vorhandene Zeit reichte jedoch nicht aus, um eine Fertigstellung zu erwirken.

### 3.11.6 Aufgetretene Probleme

Durch die naive Schätzung von Story-Points zu Beginn des ersten Sprints wurden zu viele Tasks mit zu kurzer Bearbeitungszeit vergeben. Dadurch konnten nicht alle Tasks in der geplanten Zeit fertiggestellt werden.

### 3.11.7 Teamtreffen Freitag 24.05.19

Das nächste Teamtreffen findet am Dienstag, den 28.05.19 um 14 Uhr statt.

Frage an Betreuer: User Stories mit optischen Anpassungen? Task oder User-Stories? Mail an Sepp, damit das Feature freigeschaltet wird, Story points auf Tasks zu verteilen.

Wochen Resümee: Was für Probleme sind aufgetreten? Was für Fortschritt gab es.

#### 3.11.7.1 Chat:

Der Server schickt Fehlermeldungen. Die Chat-Software muss deshalb angepasst werden, um eine Behandlung der Fehlermeldungen zur Verfügung zu stellen. Weiterhin wurde das Format der Servernachrichten verändert.

Da der Allgemeine-Chat serverseitig Fehlermeldungen lieferte, muss ein Bugreport an den Kunden geschrieben werden.



Der private Chat funktionierte hingegen ohne weitere Probleme. Bei einem privaten Chat mit einem unangemeldeten Nutzer versendete der Server eine Fehlermeldung: „User xy is not online“.

Die gewünschten ChatCommands wurden implementiert. Zusätzlich wurde ein /leave Kommando für den privaten Chat implementiert.

#### **3.11.7.2 Lobby:**

Es gab weiterhin Probleme mit der Chat-Integration. Zur Lösung des Problems soll im Folgenden eine Lobby-Klasse angelegt werden, die beim Übergang in den Lobby-Bildschirm erstellt wird.

Diese soll als reine Datenklasse fungieren.

#### **3.11.7.3 Ingame-Bildschirm:**

Die benötigten Controller und Builder wurden fertiggestellt. Die FXML-Datei für den Spiel-Bildschirm muss noch erstellt werden.

#### **3.11.7.4 Spiel erstellen:**

Für den Task der Spielerstellung wurde der alte Branch des ehemaligen Mitarbeiters nicht übernommen. Durch die vollständige Neuimplementierung wurde zusätzliche Zeit benötigt.

#### **3.11.7.5 Allgemein**

Die vorhandene Ordnerstruktur soll überarbeitet werden und nach Funktion gekapselt werden. Der LoginController besaß zum aktuellen Zeitpunkt zu viele Aufgaben/Funktionalitäten. Diese sollen aufgeteilt und wo möglich in andere Klassen übernommen werden.

## 4 Zweiter Sprint

### 4.1 Ziele des Sprints

### 4.2 Planung

Task	Story-Points	Bearbeiter
Core – Anzeigen von Lobby	4	Georg Siebert
Core – Schließen des Clients	13	Jan Müller
Core – Spiel beitreten	7	Keanu Stückrad
Core – Spiel verlassen	3	Keanu Stückrad
Appearance – General	8	Juri Lozowoj
Appearance – Lobby	13	Georg Siebert
Appearance - Chat	13	Jan Müller
Appearance – Login	2	Juri Lozowoj
Optional – Wechsel der Sprache	3	Georg Siebert
Optional – Chat Features: Zitate	3	Juri Lozowoj
Änderung - Überarbeitung der Fehlermeldungen im Login-Screen	2	Juri Lozowo
Appearance - Animierter Titel im Login-Bildschirm	8	Keanu Stückrad
Appearance - Spielname in Fensterüberschrift	0	Juri Lozowoj
Appearance - Icon in Spielfenster austauschen	1	Juri Lozowoj
$\Sigma$	80	

**Tabelle 4: Planung des zweiten Sprints**

### 4.3 Tasks

#### 4.3.1 Core – Anzeigen von Lobby

Der Task wurde aus dem ersten Sprint übernommen. Vgl. dazu den Eintrag 2.9.7

##### 4.3.1.1 Anforderungen

Vgl. Eintrag 2.9.7.2

##### 4.3.1.2 Zeit SOLL

Für die Zeit wurde eine Bearbeitungszeit von 4 Stunden eingeplant. Im ersten Sprint bereits 16 Stunden und 30 Minuten Arbeitszeit eingesetzt.

##### 4.3.1.3 User-Story

Vgl. Eintrag 2.9.7.4

#### **4.3.1.4 Position innerhalb der Domain-Story**

Vgl. Eintrag 2.9.7.5

#### **4.3.1.5 Abweichungen**

Die Aufgabe sollte bereits im ersten Sprint abgearbeitet sein. Für die Bearbeitung wurde wesentlich mehr Zeit benötigt, als ursprünglich geplant war. Dies lag vor allem an der Implementierung des neuen Designs und dem Schreiben von Tests.

### **4.3.2 Core – Schließen des Clients**

#### **4.3.2.1 Anforderungen**

Der Benutzer muss den Client schließen können, um das Programm zu beenden.

#### **4.3.2.2 Zeit SOLL**

Für die Bearbeitung des Tasks wurden 13 Stunden eingeplant.

#### **4.3.2.3 User-Story**

Der Benutzer sieht den geöffneten Client. Er betätigt den "Schließen" Knopf, welcher in der rechten oberen Ecke des UI als Kreuz dargestellt wird. Nach der Betätigung erscheint ein Pop-Up Menü im Vordergrund des Startbildschirms mit der Nachricht: "Wollen Sie die Anwendung wirklich schließen?". Weiterhin werden die Auswahlmöglichkeiten "Ok" und "Abbrechen" angezeigt. Wird auf den "Ok" Knopf gedrückt, so schließt sich die Anwendung, wird der "Abbrechen" Knopf betätigt, so wird erneut der Startbildschirm angezeigt.

#### **4.3.2.4 Position innerhalb der Domain-Story**

Das Schließen des Clients sorgt für die ordnungsgemäße Abmeldung des Spielers am Server und gibt belegte Ressourcen, sowie Sockets frei.

### **4.3.3 Core – Spiel beitreten**

#### **4.3.3.1 Anforderungen**

Der Benutzer muss innerhalb der Lobby einem laufenden Spiel beitreten können.

#### **4.3.3.2 Zeit SOLL**

Für die Bearbeitung des Tasks wurden 7 Stunden eingeplant.

#### **4.3.3.3 User-Story**

Der Benutzer befindet sich in der Lobby.

Er sieht die Liste der offenen Spiele.

Der Benutzer klickt auf das gewünschte Spiel.

Es öffnet sich ein Pop-Up Fenster mit dem Text: "Wollen Sie dem Spiel beitreten?" und der Auswahl "Ok", "Abbrechen".

Wurde "Ok" gewählt, so wird er zum Spiel-Bildschirm weitergeleitet.

#### **4.3.3.4 Position innerhalb der Domain-Story**

Die Spiel-beitreten Funktion ist neben der Spiel-erstellen Funktion, die einzige Möglichkeit aus der Lobby in den Spiel-Bildschirm zu wechseln.

#### **4.3.3.5 Abweichungen**

Der Task konnte nicht in der geplanten Zeit abgeschlossen werden. Dies lag an der Fehlschätzung zu Beginn des Sprints.

### **4.3.4 Core – Spiel verlassen**

Der Task wurde aus dem ersten Sprint übernommen, vgl. 2.9.12

#### **4.3.4.1 Anforderungen**

Vgl. 2.9.12.1

#### **4.3.4.2 Zeit SOLL**

Vgl. 2.9.12.2

#### **4.3.4.3 User-Story**

Vgl. 2.9.12.3

#### **4.3.4.4 Position innerhalb der Domain-Story**

Vgl. 2.9.12.4

#### **4.3.4.5 Abweichungen**

Der Task konnte nicht in der geplanten Zeit fertiggestellt werden.

### **4.3.5 Appearance – General**

#### **4.3.5.1 Anforderungen**

Der Client soll in einem einheitlichen Gestaltungs-Schema dargestellt werden.

#### **4.3.5.2 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurden 8 Stunden eingeplant.

#### **4.3.5.3 User-Story**

Das Erscheinungsbild des Logins, der Lobby und des Spiels ist einheitlich. Es gibt ein gleiches Farbschema.

Knöpfe, Listenelemente und Textfelder werden in schwarz gehalten. Die Schriftart und die Rahmen von Elementen werden in weiß dargestellt.

Die Schriftgrößen sind in allen Bereichen des Spiels genormt und in einem CSS-Stylesheet festgehalten.

#### **4.3.5.4 Abweichungen**

Der Task konnte in der geplanten Zeit abgeschlossen werden.

## **4.3.6 Appearance – Lobby**

### **4.3.6.1 Anforderungen**

Die Lobby als zentraler Bestandteil des Clients soll sich am einheitlichen Gestaltungs-Schema orientieren und muss wichtige Informationen für den Spieler klar hervorheben, sowie Funktionalitäten möglichst einfach zugänglich machen.

### **4.3.6.2 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurden 13 Stunden eingeplant.

### **4.3.6.3 User-Story**

Der Benutzer befindet sich in der Lobby.

Die Lobby erscheint im selben Stil wie die Eingabemaske im Login.

Die Schriftart ist weiß, die schwarzen Listenelemente heben sich durch ihre weißen Rahmen vom Hintergrund ab.

### **4.3.6.4 Abweichungen**

Der Task konnte in der geplanten Zeit abgeschlossen werden.

## **4.3.7 Appearance – Chat**

### **4.3.7.1 Anforderungen**

Der Chat als zentrales Medium der sozialen Interaktion muss Nachrichten gut lesbar darstellen und unterschiedliche Gespräche und Sender voneinander unterscheidbar machen.

### **4.3.7.2 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurden 13 Stunden eingeplant.

### **4.3.7.3 User-Story**

Der Nutzer sieht den Chat.

Im Allgemeinen Chat wird jeder Benutzername farblich hinterlegt und ist einmalig vergeben. Die Schriftfarbe der Chat-Nachrichten sind weiß.

Der Hintergrund des Chats hebt sich farblich vom dunklen Hintergrund hervor.

### **4.3.7.4 Abweichungen**

Der Zeit wurde schneller als in der Planung vorgesehen fertiggestellt.

## **4.3.8 Appearance – Login**

### **4.3.8.1 Anforderungen**

Die Gestaltung des Login-Bildschirms muss sich dem einheitlichen Design des Clients anpassen.

### **4.3.8.2 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurden 2 Stunden eingeplant.

#### **4.3.8.3 User-Story**

Der Benutzer befindet sich im Login-Bildschirm.

Die Eingabemaske befindet sich in einer Box, welche in einem dunklen Farbschema dargestellt wird. (Hintergrund schwarz, Eingabefelder weiß).

Im Hintergrund des Login-Screens befindet sich das Splash-Art, welches eine Spiel-Szene darstellt.

#### **4.3.8.4 Abweichungen**

Der Task wurde nicht in der geplanten Zeit fertiggestellt. Der Entwickler benötigte für die Implementierung mehr Zeit als vorgesehen war.

### **4.3.9 Optional – Wechsel der Sprache**

Der Task wurde aus dem ersten Sprint übernommen vgl. 2.9.8

#### **4.3.9.1 Anforderungen**

Vgl. 2.9.8.1

#### **4.3.9.2 Zeit SOLL**

Vgl. 2.9.8.2

#### **4.3.9.3 User-Story**

Vgl. 2.9.8.3

#### **4.3.9.4 Position innerhalb der Domain-Story**

Vgl. 2.9.8.4

#### **4.3.9.5 Abweichungen**

Der Task wurde nicht in der geplanten Zeit fertiggestellt. Dies lag an der Einarbeitungszeit in die Software-Bibliothek für die Internationalisierung.

### **4.3.10 Optional – Chat Features: Zitate**

#### **4.3.10.1 Anforderungen**

Durch die Eingabe eines Befehls werden Chuck-Norris Zitate im Chat generiert.

#### **4.3.10.2 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurden 3 Stunden eingeplant.

#### **4.3.10.3 User-Story**

Der Benutzer befindet sich in der Lobby oder im Spiel.

Er öffnet den Chat und gibt das Kommando "/chuckMe" ein.

Es erscheint ein Chuck Norris Witz.

#### **4.3.10.4 Abweichungen**

Die Bearbeitung des Tasks wurde durch die Einarbeitungszeit in die Verarbeitung von JSON-Objekten verzögert.

### **4.3.11 Änderung - Überarbeitung der Fehlermeldungen im Login-Screen**

#### **4.3.11.1 Anforderungen**

Die Pop-Up Fehlermeldungen im Login, sollen durch Status-Meldungen über der Eingabemaske ersetzt werden.

#### **4.3.11.2 Zeit SOLL**

Für die Bearbeitung wurden 2 Stunden eingeplant.

#### **4.3.11.3 User-Story**

Der Benutzer befindet sich im Login-Bildschirm.

Solange keine Fehlermeldung aufgetreten ist, ist das Feld "unsichtbar".

Tritt eine Fehlermeldung auf, so wird diese angezeigt, bis eine andere Aktion ausgeführt wird.

#### **4.3.11.4 Abweichungen**

Die Bearbeitungszeit des Task wurde in minimalem Umfang überschritten.

### **4.3.12 Appearance - Animierter Titel im Login-Bildschirm**

#### **4.3.12.1 Anforderungen**

Im Login-Bildschirm sollen Titel und Beiname des Spiels erscheinen. Letzterer soll animiert sein.

#### **4.3.12.2 Zeit SOLL**

Für die Bearbeitung der Aufgabe wurden 8 Stunden eingeplant.

#### **4.3.12.3 User-Story**

Der Benutzer befindet sich im Login-Bildschirm.

Im Login-Screen soll der Titel des Spiels angezeigt werden.

Dieser besteht aus zwei Teilen:

- 1.) dem Spiel-Titel "RBSG"
- 2.) dem Beinamen "Enhanced Wars"

#### **4.3.12.4 Abweichungen**

Der Task wurde schneller als vorgesehen abgearbeitet.

### **4.3.13 Appearance - Spielname in Fensterüberschrift**

#### **4.3.13.1 Anforderungen**

Im Rahmen des Spielfensters soll der Spielname dargestellt werden.

#### **4.3.13.2 Zeit SOLL**

Für die Bearbeitung des Tasks wurden weniger als 5 Minuten eingeplant.

#### **4.3.13.3 User-Story**

Der Benutzer hat den Client geöffnet.

Der Spielname "RBSG - Advanced Wars" wird in der Leiste des Spiels (oben in der Mitte) dargestellt werden.

#### **4.3.13.4 Abweichungen**

Der Task wurde durch die Veränderung des Designs verzögert.

### **4.3.14 Appearance - Icon in Spielfenster austauschen**

#### **4.3.14.1 Anforderungen**

Im Rahmen des Spielfensters soll das Spiel-Icon dargestellt werden.

#### **4.3.14.2 Zeit SOLL**

Für die Bearbeitung des Tasks wurde 1 Stunde eingeplant.

#### **4.3.14.3 User-Story**

Der Benutzer hat den Client geöffnet.

In der oberen linken Ecke des Fensters befindet sich das Spiel-Icon.

#### **4.3.14.4 Abweichungen**

Der Task wurde in der vorgesehenen Zeit abgearbeitet.

### **4.3.15 Optional - Zeitstempel vor Chat-Nachrichten**

#### **4.3.15.1 Anforderungen**

Im Chat soll zu jeder Nachricht ein Zeitstempel angezeigt werden, welcher den Zeitpunkt der Nachricht des Erhalts der Nachricht beschreibt.

#### **4.3.15.2 Zeit SOLL**

Für die Bearbeitung des Tasks wurde 1 Stunde eingeplant.

#### **4.3.15.3 User-Story**

Der Benutzer befindet sich in der Lobby.

Im Chat werden Zeitstempel vor den Nachrichten angezeigt.

Format:



[hh:mm:ss] <Username> : Nachricht

#### **4.3.15.4 Abweichungen**

Der Task wurde in der zweiten Woche des zweiten Sprints hinzugefügt.

Der Task wurde in der vorgesehenen Zeit abgearbeitet.

### **4.3.16 Optional – Ladesymbol im Login**

#### **4.3.16.1 Anforderungen**

Loggt sich der Benutzer ein, so soll solange ein Ladesymbol angezeigt werden, bis eine Nachricht des Servers eintrifft.

#### **4.3.16.2 Zeit SOLL**

Für die Bearbeitung des Tasks wurde 5 Stunden eingeplant.

#### **4.3.16.3 User-Story**

Der Benutzer befindet sich im Login-Bildschirm.

Er gibt seine Benutzerdaten ein und drückt auf Login/Registrieren.

Es erscheint ein Ladesymbol im Status-Fenster.

#### **4.3.16.4 Abweichungen**

Der Task wurde in der zweiten Woche des zweiten Sprints hinzugefügt.

Der Task konnte schneller als geplant abgearbeitet werden.

## **4.4 Zusammenfassung der ersten Woche**

### **4.4.1 Absprache zum Chat, 26.05.2019**

Ziel der Absprache war die Klärung des Chat-Verhaltens bei Nachrichten an nicht-angemeldete Benutzer.

Als Ergebnis wurde folgendes festgehalten:

Wird eine Nachricht an einen nicht bestehenden, oder nicht angemeldeten Nutzer verschickt, so öffnet sich ein Fenster, in dem die versendete Nachricht und eine Fehlermeldung erscheinen. In diesem neuen Tab kann der Sender aber nicht schreiben.

### **4.4.2 Absprache über den Aufbau des Ingame-Fensters, 26.05.2019**

Ziel der Absprache war eine Klärung der implementierten Proportionen der Elemente im Ingame-Bildschirm.

Als Ergebnis der Absprache wurde folgendes festgehalten:

Der Chat soll über die vollständige Breite des Fensters skalieren. Zusätzlich sollen die Felder für die Spielernamen und die Status-Meldungen nicht direkt am oberen Ende des Fensters anliegen, sondern auf ca. 90 % der Fensterhöhe anliegen.

#### 4.4.3 Feature-Anfrage, 26.05.2019

In einer Mail an den Kunden wurden die folgenden Features erfragt:

- (a) Fehlermeldungen, während des Login-Screens, oder der Registrierung in einem Textfeld über der Eingabemaske
- (b) Anzeigen von Spiel-Titel im Fensterrahmen
- (c) Anzeigen von Spiel-Icon im Fensterrahmen
- (d) Erweiterung der Login-UI durch Dark-Theme mit Titel "RBSG" und animiertem Beinamen

Bis zum Ende des 1. Releases ist keine Antwort eingetroffen.

#### 4.4.4 Tasks SOLL vs IST

Die folgende Tabelle ist eine Gegenüberstellung der geplanten und der tatsächlich geleisteten Arbeit innerhalb der ersten Woche des zweiten Sprints.

Task	Zeit Soll	Zeit Ist	Fertig
Core – Anzeigen von Lobby	4	19,15	x
Core – Schließen des Clients	13	6,5	x
Core – Spiel beitreten	7	5,5	-
Core – Spiel verlassen	3	3	-
Appearance – General	8	-	-
Appearance – Lobby	13	7	-
Appearance - Chat	13	4,5	x
Appearance – Login	2	-	-
Optional – Wechsel der Sprache	3	1,5	-
Optional – Chat Features: Zitate	3	3	x
Änderung - Überarbeitung der Fehlermeldungen im Login-Screen	2	2,016	-
Appearance - Animierter Titel im Login-Bildschirm	8	4	x
Appearance - Spielname in Fensterüberschrift	0	0,083	x
Appearance - Icon in Spielfenster austauschen	1	0,25	x
<b>Σ</b>	<b>80</b>	<b>56,5</b>	
<b>Offene Stunden</b>			

**Tabelle 5: Zwischenstand 1. Woche 2. Sprint**

In der ersten Woche des zweiten Sprints konnten 7 von 14 Tasks abgeschlossen werden. Bei den abgearbeiteten Tasks wurden in 4 Fällen weniger Zeit für die Bearbeitung verbraucht als ursprünglich geplant war.

Dies lag vor allem daran, dass das Problem schneller behoben und Einarbeitungszeiten kürzer ausfielen, als in den Teamtreffen angedacht wurde.

In einem der sieben Fälle wurde mehr Zeit benötigt als vorgesehen war. Bei diesem Vorgang handelt es sich um die Erstellung der Lobby.

Durch die umfangreichen Änderungen am Design, welche durch den Neuaufbau der UI entstanden, mussten bereits getätigte Arbeiten angepasst und verbessert werden. Ebenfalls traten einige Abhängigkeiten zu anderen Aufgaben auf, welche innerhalb der Planung nur teilweise bedacht worden waren.

Insgesamt konnten durch die schnellere Bearbeitung der Aufgaben in der ersten Woche des zweiten Sprints 0,75 Stunden vom Budget eingespart werden.

#### 4.4.5 Burndown-Chart

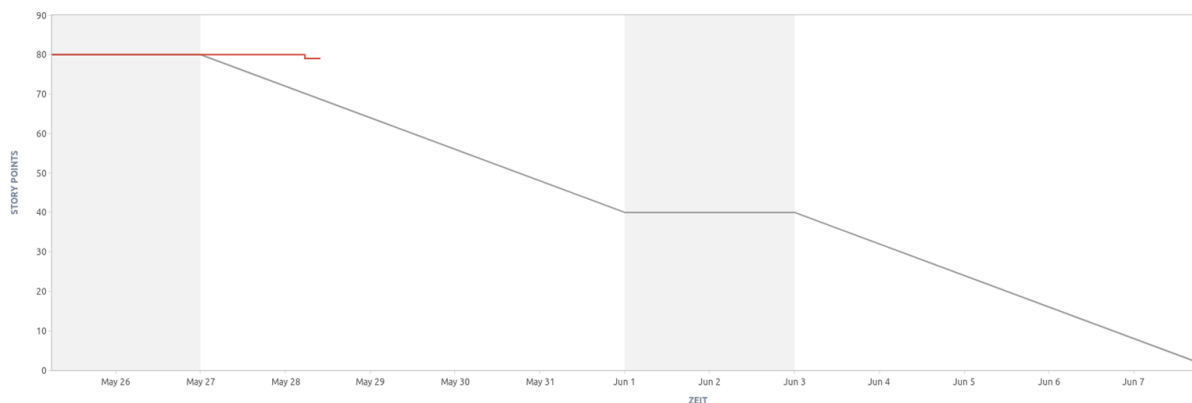


Abbildung 20: Burndown Chart - 1. Woche 2. Sprint

Im Burndown-Chart der ersten Woche des zweiten Sprints sieht man keine großen Abnahmen von Story-Points. Dies hat verschiedene Gründe.

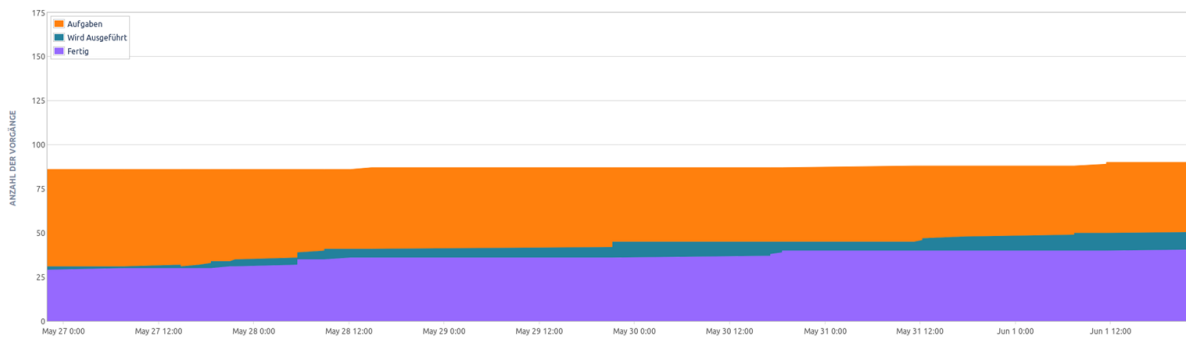
Bedingt durch die Größe der einzelnen Tasks und dem Credo „Keine Fertigstellung von Tasks vor vollständiger Anpassung in die UI und Abnahme durch den PO“ wurden Tasks nicht als beendet gekennzeichnet, obwohl diese den gestellten Ansprüchen zu großen Teilen bereits entsprachen.

Die fehlende Abnahme ist also irreführend in dem Sinne, als dass sie den Eindruck vermittelt es wäre keine Arbeit geleistet worden.

Weiterhin gab es durch die Abhängigkeiten der verschiedenen Aufgaben Wartezeiten, welche sich auf die Arbeit und den folgenden Abschluss auswirkten.

Unabhängig von der graphischen Darstellung der Arbeit wurden, wie der Arbeitszeitentabelle zu entnehmen ist, 50% der geforderten Aufgaben erfolgreich abgeschlossen und 45 Minuten an geplanter Arbeitszeit eingespart.

#### 4.4.6 Kumuliertes Flussdiagramm



**Abbildung 21: Kumuliertes Flussdiagramm 1. Woche 2. Sprint**

Dem Flussdiagramm lässt sich ebenfalls entnehmen, dass wenige der zugeteilten Tasks als abgeschlossen gekennzeichnet wurden.

## **4.5 Teamtreffen 28.05.2019**

### **4.5.1 Organisatorische Probleme**

Im Teamtreffen wurden keine organisatorischen Probleme festgestellt.

### **4.5.2 Entwicklungsprobleme**

#### **4.5.2.1 Schließen des Clients**

Folgende Fragen sollen in der kommenden Woche geklärt werden:

- a) Wie soll der Client richtig geschlossen werden?
- b) Soll das Client Modell verändert, oder nur die Funktionen angepasst werden?

#### **4.5.2.2 Anzeige der Lobby**

Die UI muss noch weiter mit dem Product Owner abgesprochen werden.

#### **4.5.2.3 Anzeige des Ingame-Bildschirms**

Nachdem die Anforderungen des zweiten Releases veröffentlicht wurden, wurde festgestellt, dass das Ingame UI Ticket sich mit Anforderungen an das zweite Release überschneidet.

Weiterhin wurde vermerkt, dass die vorgegebene Schriftart für den Beinamen des Titels im Login-Bildschirm nicht funktioniert.

### **4.5.3 Absprache über Produkt-Beinamen, am 31.05.2019**

Ziel der Absprache war es einen neuen Beinamen für den Produktnamen „RBSG“ zu finden. Da bis zu diesem Zeitpunkt keine offizielle Rückmeldung erfolgt war, wurden Vorschläge gesammelt. Schlussendlich wurde sich für den Titel „Enhanced Wars“ entschieden, da der bisherige Beiname „Advanced Wars“ markenrechtlich geschützt ist.

#### 4.5.4 Absprache über Lobby, am 31.05.2019

Aufgrund der geplanten Änderungen des Erscheinungsbilds waren einige Fragen aufgetreten. Zu klären war, ob die bisherige Umsetzung des Entwicklers den Vorstellungen des Product-Owners entsprachen.

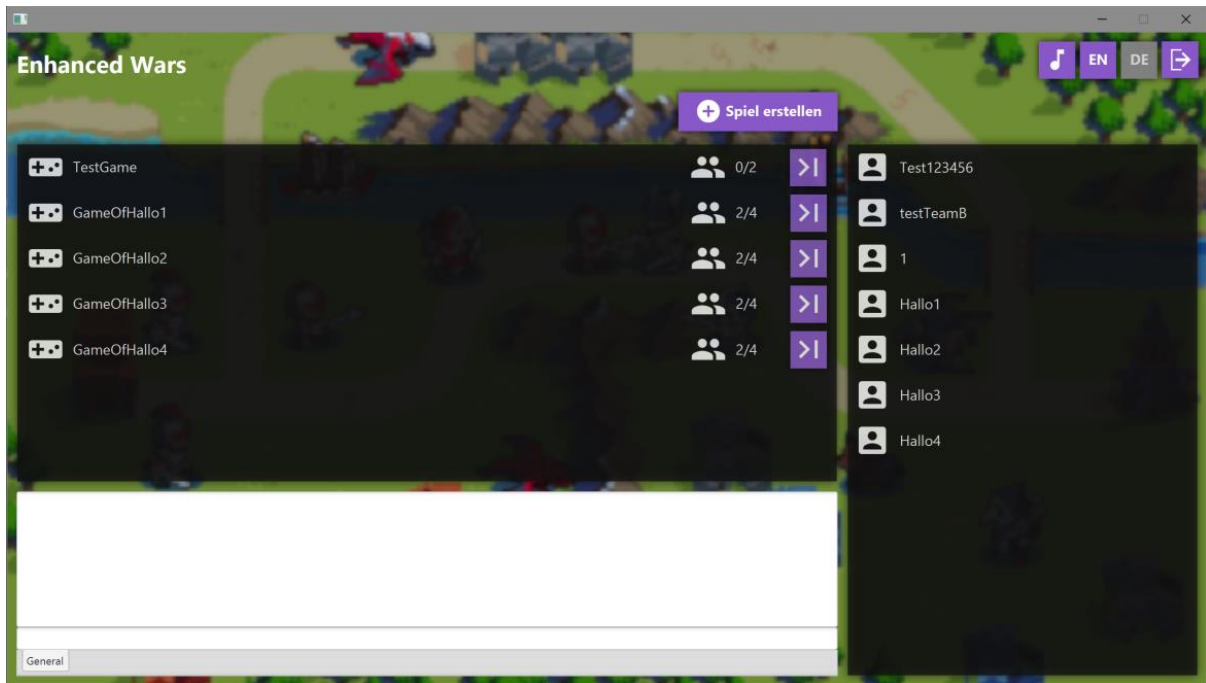


Abbildung 22: Überarbeitetes Design der Lobby

Resultat der Absprache waren:

- a) Abgrenzung der Listenelemente untereinander durch weiße Rahmen.
- b) Verschiebung des Spiel-erstellen Knopfes auf die Unterseite der Spielliste.

#### 4.5.5 Absprache über die Darstellung des Chats, am 01.06.2019

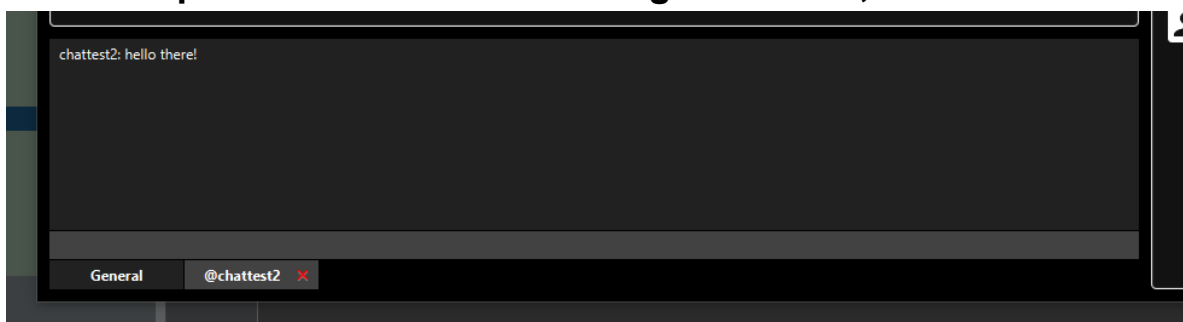


Abbildung 23: Überarbeiteter Chat

Ziel der Absprache war ein Feedback zum überarbeiteten Design des Chats.

- a) Anforderung, dass der Chat mit einem weißen Rahmen umschlossen werden soll.
- b) Akzeptanz der sonstigen Änderungen

## 4.6 Zusammenfassung der zweiten Woche

### 4.6.1 Tasks SOLL vs IST

Task	Zeit Soll	Zeit Ist	Fertig
Core – Anzeigen von Lobby	4	19,15	x
Core – Schließen des Clients	13	8	x
Core – Spiel beitreten	7	8,66	x
Core – Spiel verlassen	3	5	x
Appearance – General	8	8	x
Appearance – Lobby	13	13	x
Appearance - Chat	13	4,5	x
Appearance – Login	2	2,75	x
Optional – Wechsel der Sprache	3	3,75	x
Optional – Chat Features: Zitate	3	6,5	x
Änderung - Überarbeitung der Fehlermeldungen im Login-Screen	2	2,0167	x
Appearance - Animierter Titel im Login-Bildschirm	8	4	x
Appearance - Spielname in Fensterüberschrift	0	0,833	x
Appearance - Icon in Spielfenster austauschen	1	0,25	x
Optional - Zeitstempel vor Chat-Nachrichten	1	0,1667	x
Optional - Ladesymbol im Login	5	3	x
<b>Σ</b>	<b>80</b>	<b>86,41</b>	
<b>Offene Stunden</b>			

**Tabelle 6: Zwischenstand 2. Woche 2. Sprint**

In der zweiten Woche des zweiten Sprints konnten alle verbleibenden Tasks abgearbeitet werden.

Zusätzlich zu den Tasks aus dem 1. Sprint wurden die Tasks „Zeitstempel von Chat-Nachrichten und Ladesymbol im Login hinzugenommen. Diese beiden Aufgaben wurden in der geplanten Zeit abgearbeitet.

Insgesamt trat eine Verzögerung von 5,41 Stunden auf. Der Anteil der der Verzögerungen aus der zweiten Woche betrug dabei 4,327 Stunden.

## 4.6.2 Kumuliertes Flussdiagramm

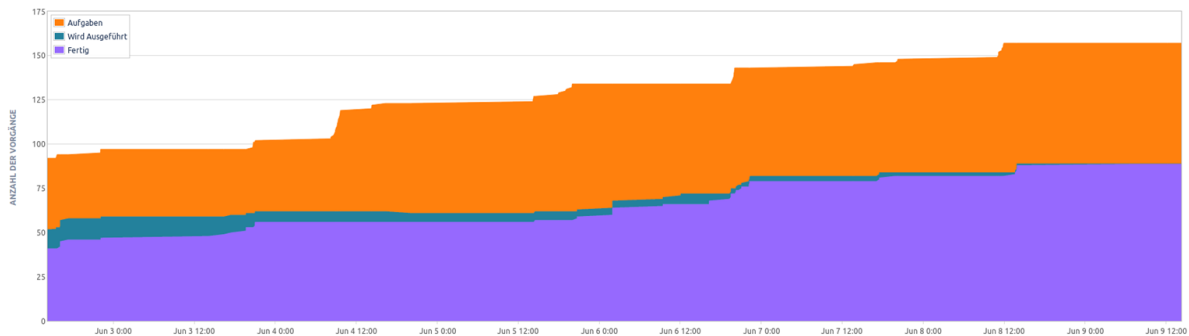


Abbildung 24: Flussdiagramm - 2. Woche 2. Sprint

Dem Flussdiagramm ist zu entnehmen, dass in der zweiten Woche des zweiten Sprints nahezu alle zugeteilten Aufgaben abgeschlossen werden konnten. Einzig der Task „Spiel verlassen“ mit einem Umfang von 4 Story Points ist zum Ende der zweiten Woche des zweiten Sprints noch offen.

Der Zuwachs der Aufgaben ist durch die Vorbereitung des neuen Releases zu begründen. In der betrachteten Woche wurden die neuen Anforderungen veröffentlicht und im Jira-System erfasst.

## 4.6.3 Absprache über Darstellung des Login-Screens, 06.06.2019

Ziel der Absprache war die Klärung der Darstellung der Eingabemaske im Login-Fenster.

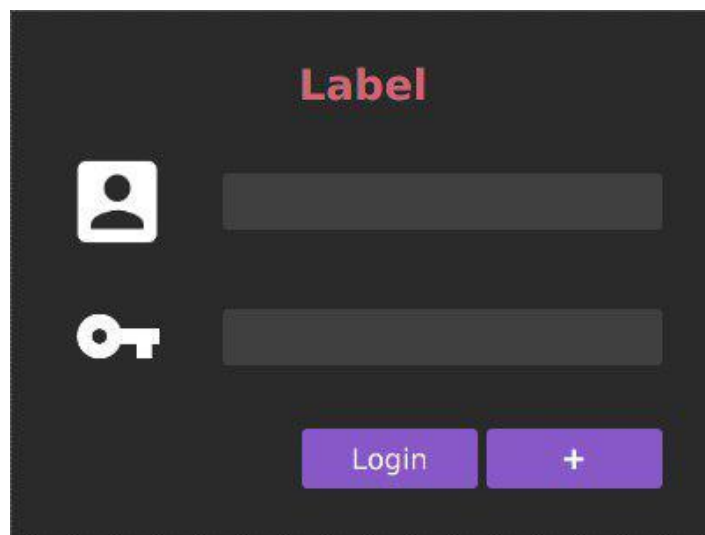


Abbildung 25: Überarbeitetes Erscheinungsbild der Login-Eingabemaske

Ergebnis der Absprache waren:

- a) Eine Vergrößerung der Schrift in den Eingabefeldern, der Fehler-Anzeige und den Knöpfen
- b) Der Abstand zwischen den Knöpfen und den Eingabefeldern

#### **4.6.4 Absprache über Darstellung des Chats, 07.06.2019**

Ziel der Absprache war die Klärung des Chat-Designs.

Die Ergebnisse der Absprache waren:

- a) Entfernen der weißen Umrandung des Chats
- b) Farbliche Markierung des ausgewählten Chat-Fensters

#### **4.6.5 Absprache über Sortier-Funktion der Spiel-Liste in der Lobby, am 07.06.2019**

Aufgrund der knappen restlichen Zeit und der offenen Verbesserungs- und Abschlussarbeiten wurde beschlossen, die Sortierfunktion für die Spielliste nicht im ersten Release zu implementieren.

#### **4.6.6 Absprache über Wechsel des Hintergrundbilds im Login und in der Lobby, am 07.06.2019**

Aufgrund der fehlenden Informationen bezüglich der Verwendung von Platzhalter-Bildern mit Copyright, wurde beschlossen das zum genannten Zeitpunkt verwendete Bild gegen ein Bild ohne Copyright auszutauschen.

Der Hintergrund des Login-Screens und der Lobby unterscheidet sich aus diesem Grund von den Mockups und den Bildern aus der Dokumentation.

#### **4.6.7 Teamtreffen, Freitag 07.06.2019**

##### ***4.6.7.1 Probleme mit der Organisation***

Während der Arbeit in der zweiten Woche des zweiten Sprints entstand eine Feedbackschleife mit dem PO. Dadurch wurde die Abarbeitung des Redesigns verzögert.

##### ***4.6.7.2 Probleme bei der Entwicklung:***

Bei Verlassen der Ingame-Lobby konnte kein neues Spiel erstellt und die Listen nicht aktualisiert werden. Dies schien auf ein Problem mit den WebSockets hinzudeuten.

##### ***4.6.7.3 Zusammenfassung***

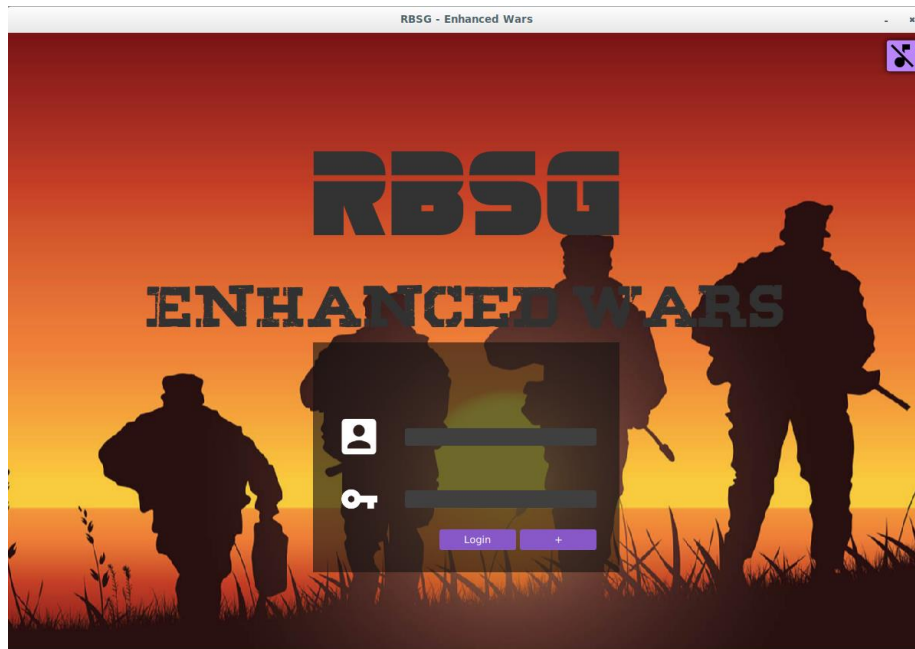
- a) Die Hintergrundmusik konnte mittels JavaFX-Media Bibliothek angebunden werden. Die zu verwendende Musik war nicht ganz klar.
- b) Die Tests wurden durch das Löschen von Abhängigkeiten entschlackt.
- c) Die Internationalisierung der Client-Software erfolgte mittels Rincled Bibliothek.
- d) Die Chuck Norris Api wurde als optionales Feature eingefügt.
- e) Absprache am 07.06.2019



## 4.7 Die Client-Software

Nachfolgend wird die Implementierung der Client-Software zum Ende des 1. Releases dargelegt.

### 4.7.1 Der Login



**Abbildung 26: Login - Bildschirm**

Der Login-Bildschirm beinhaltet den Titel und Beinamen des Spiels, die Eingabemaske für die Benutzerdaten und den Kontrollknopf für die Hintergrundmusik.

Die Eingabemaske besteht aus dem Feld für den Benutzernamen, gekennzeichnet durch das „Benutzer-“ Symbol, dem Feld für das Passwort, welches durch einen Schlüssel gekennzeichnet ist und den beiden Knöpfen für den Login „Login“ und die Registrierung „+“.

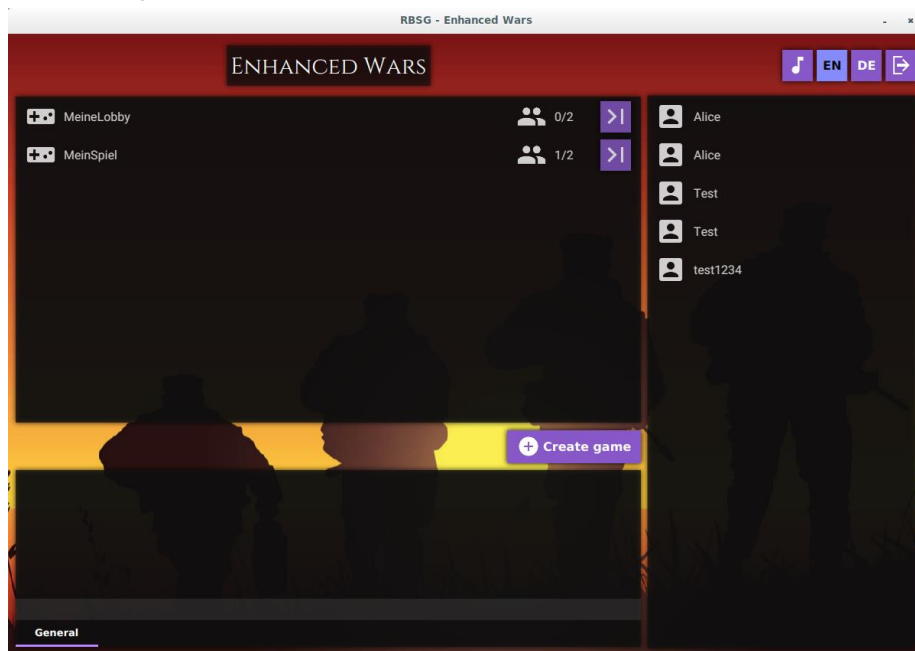
Durch das Betätigen des Musik-Knopfes in der oberen rechten Ecke des Bildschirms kann die Hintergrundmusik an-, oder abgeschaltet werden.

Durch ein Betätigen des Login-, bzw. des Registrierungsknopfes erfolgt die Weiterleitung in die Lobby.

Das eingesetzte Splash-Art „Navy Seal Training Silhouette“ wurde von Vecteezy.com bezogen, einer Content-Plattform für freie Vektorgraphiken. Es ist für die kostenfreie Verwendung lizenziert. Dabei ist das Bild eingeschränkt auf die Verwendung in Produkten mit maximal 50 Verkäufe oder Marketingzwecken mit maximal 10.000 View. Zusätzlich ist eine Nennung der Plattform Vecteezy.com erforderlich. (16)

Das Splash-Art dient zum aktuellen Zeitpunkt lediglich als Platzhalter und wird bei erster Gelegenheit durch eine originale Spielszene ausgetauscht.

## 4.7.2 Die Lobby



**Abbildung 27: Lobby - Bildschirm**

Die Lobby besteht aus fünf Komponenten. Der Liste aller offenen Spiele(1), der Liste aller angemeldeten Spieler(2), dem Chat(3), der Menü-Leiste(4) und dem „Spiel erstellen“-Knopf(5).

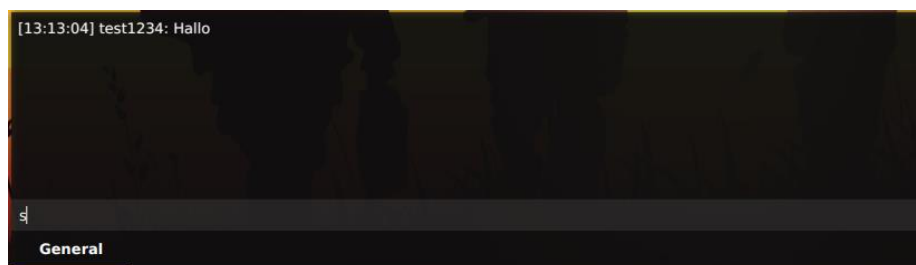
Die Liste der offenen Spiele zeigt alle zum aktuellen Zeitpunkt verfügbaren Spiele an. Ein Spiel besteht dabei aus dem Spielnamen, der Anzahl der beigetretenen Spieler und dem „Beitreten“-Knopf.

Durch ein betätigen des Beitreten-Knopfes wird der Benutzer in den Ingame-Bildschirm weitergeleitet.

Die Spielerliste zeigt alle zum aktuellen Zeitpunkt angemeldeten Benutzer an. Durch einen Rechtsklick auf einen Benutzernamen wird ein privater Chat mit diesem Spieler geöffnet.

Loggt ein Spieler sich aus, so wird sein Eintrag aus der Liste entfernt.

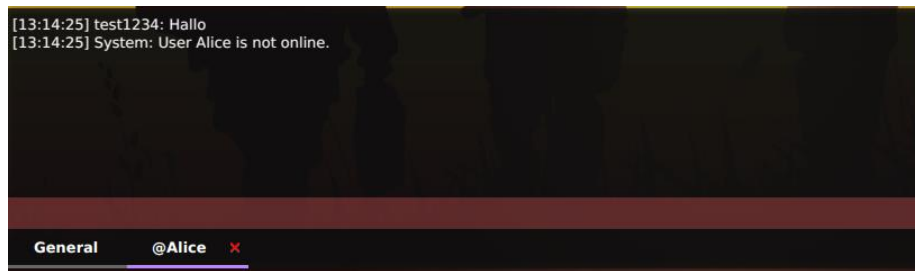
Der Chat stellt die zentrale Komponente der sozialen Interaktion zwischen den Spielern dar. Er besteht aus zwei Teilen. Einem Fenster für den Allgemeinen-Chat und mehreren Fenstern für private Unterhaltungen.



**Abbildung 28: Allgemeiner Chat**

Für jeden Eintrag im Chat wird ein Zeitstempel hinterlegt. Dadurch kann der Benutzer nachvollziehen, zu welchem Zeitpunkt die Nachricht eingetroffen ist.

Ein privater Chat kann entweder über einen Rechtsklick auf ein Element in der Spielerliste erfolgen, oder aber durch das Chat-Kommando „/w „Benutzername“ “ eröffnet werden.

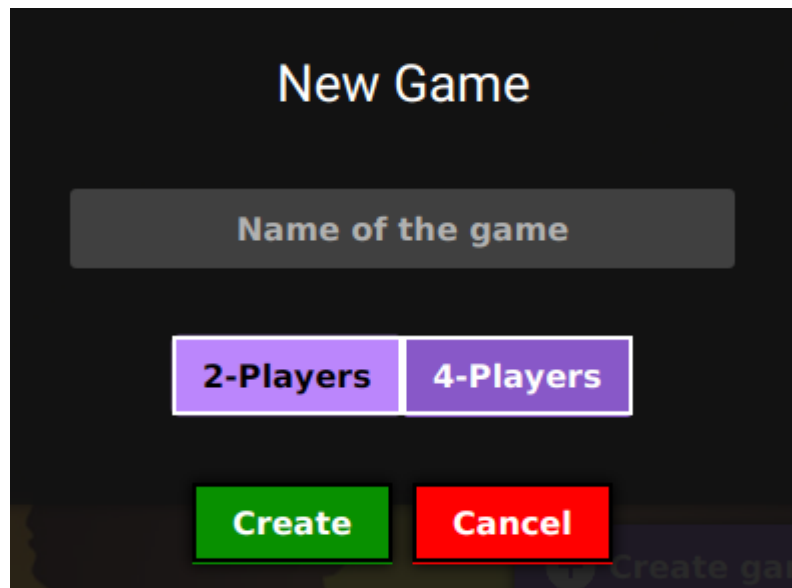


**Abbildung 29: Privater Chat**

Ist ein Spieler, mit dem ein privater Chat eröffnet wurde nicht online, so wird eine entsprechende Meldung im Chat-Fenster angezeigt.

Jeder private Chat kann mit einem Betätigen des Kreuz-Knopfes beendet werden.

Durch das Betätigen des „Spiel erstellen“-Knopfes öffnet sich ein neues Fenster.

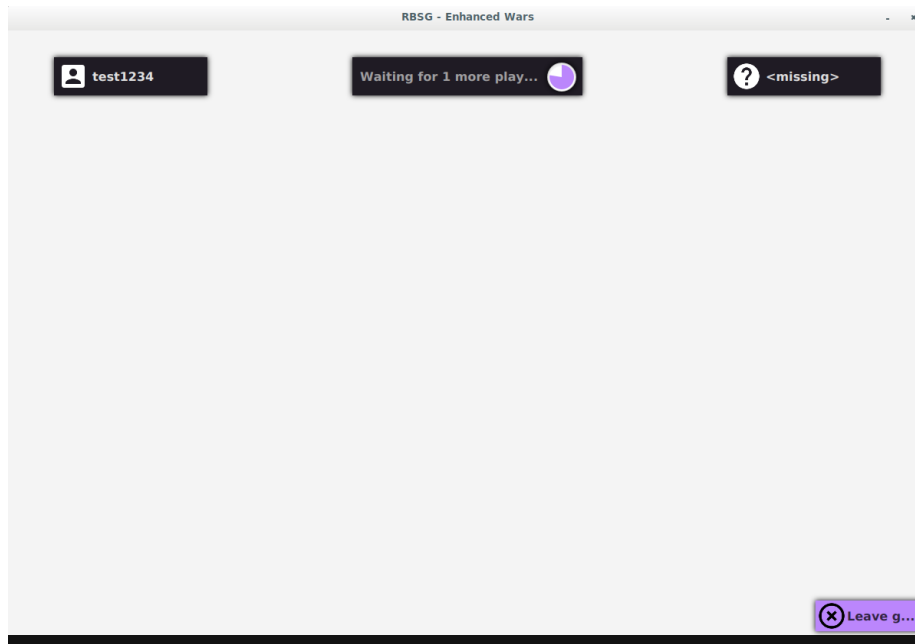


**Abbildung 30: Spiel erstellen**

In diesem Fenster werden das Eingabefeld für den Spielnamen, die Auswahl der Spieleranzahl, der „Erstellen“-Knopf und der Abbrechen-Knopf angezeigt.

### **4.7.3 Der Ingame-Bildschirm**

Der Ingame-Bildschirm kann über zwei Wege erreicht werden. Zum einen durch das Erstellen eines Spiels und zum anderen, dem Beitritt eines bereits geöffneten Spiels über die Spiel-Liste.



**Abbildung 31: Ingame - Bildschirm**

Der Ingame-Bildschirm besteht zum aktuellen Zeitpunkt aus dem Spieler-Feld(1), der Statusanzeige(2) und dem „Spiel verlassen“-Knopf(3).

Das Spielerfeld zeigt den Namen des Spielers an. Zu einem späteren Zeitpunkt soll dort zusätzlich ein den Spieler auszeichnendes Symbol hinterlegt werden.

Die Statusanzeige stellt Systemnachrichten dar. Zu den Systemnachrichten gehört zum aktuellen Zeitpunkt die Meldung, das auf zusätzliche Spieler gewartet wird.

Durch das Betätigen des „Spiel verlassen“ Knopfes wird der Benutzer zurück in die Lobby weitergeleitet.

## 5 Abbildungsverzeichnis

Abbildung 1: Mockup - Login von Benutzer .....	14
Abbildung 2: Domain Story - Login .....	15
Abbildung 3: Domain Story - Erstellung von Benutzer .....	16
Abbildung 4: Die Lobby .....	17
Abbildung 5: Domain Story - Anzeige der Lobby .....	18
Abbildung 6: Domain Story - Beitritt zu offenem Spiel .....	19
Abbildung 7: Mockup - Erstellen von Spiel.....	20
Abbildung 8: Domain Story - Erstellen von Spiel .....	21
Abbildung 9: Domain Story - Abmeldung in der Lobby .....	22
Abbildung 10: Domain Story - Abmeldung aus dem Spiel.....	23
Abbildung 11: Domain Story - Chatten.....	24
Abbildung 12: Mockup - Login Bildschirm.....	25
Abbildung 13: Mockup - Benutzerdaten .....	28
Abbildung 14: Mockup - Lobby.....	29
Abbildung 15: Mockup - Spiel erstellen .....	31
Abbildung 16: Mockup - Chat .....	33
Abbildung 17: Burndown Chart - 1. Woche 1. Sprint.....	35
Abbildung 18: Burndown Chart - 2. Woche 1. Sprint.....	39
Abbildung 19: Flussdiagramm - 2. Woche 1. Sprint .....	39
Abbildung 20: Burndown Chart - 1. Woche 2. Sprint.....	51
Abbildung 21: Kumuliertes Flussdiagramm 1. Woche 2. Sprint.....	52
Abbildung 22: Überarbeitetes Design der Lobby.....	53
Abbildung 23: Überarbeiteter Chat .....	53
Abbildung 24: Flussdiagramm - 2. Woche 2. Sprint .....	55
Abbildung 25: Überarbeitetes Erscheinungsbild der Login-Eingabemaske .....	55
Abbildung 26: Login - Bildschirm .....	57
Abbildung 27: Lobby - Bildschirm .....	58
Abbildung 28: Allgemeiner Chat .....	58
Abbildung 29: Privater Chat .....	59
Abbildung 30: Spiel erstellen.....	59
Abbildung 31: Ingame - Bildschirm.....	60

## 6 Tabellenverzeichnis

Tabelle 1: Planung des ersten Sprints.....	14
Tabelle 2: Zwischenstand der 1. Woche 1. Sprint .....	36
Tabelle 3: Zwischenstand 2. Woche 1. Sprint .....	40
Tabelle 4: Planung des zweiten Sprints .....	42
Tabelle 5: Zwischenstand 1. Woche 2. Sprint .....	50
Tabelle 6: Zwischenstand 2. Woche 2. Sprint .....	54

## 7 Literaturverzeichnis

1. **Wikipedia - Scrum\_(software\_development)**. en.wikipedia.org. [Online] [Zitat vom: 13. 06 2019.] [https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)).
2. **al., Kent Beck et.** agilemanifesto.org/. [Online] [Zitat vom: 13. 06 2019.] <https://agilemanifesto.org/>.
3. **Atlassian Corporation Plc.** <https://www.atlassian.com/>. [Online] 12. 06 2019. <https://www.atlassian.com/software/jira/features>.
4. **Wikipedia - REST**. en.wikipedia.org. [Online] 06 12, 2019. [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer).
5. **Wikipedia - Websocket**. en.wikipedia.org. [Online] [Zitat vom: 12. 06 2019.] <https://en.wikipedia.org/wiki/WebSocket>.
6. **Wikipedia - Java\_(Programmiersprache) In.** de.wikipedia.org. [Online] 12. 06 2019. [https://de.wikipedia.org/wiki/Java\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/Java_(Programmiersprache)).
7. **Oracle Corporation.** www.oracle.com. [Online] [Zitat vom: 12. 06 2019.] <https://www.oracle.com/technetwork/java/javase/overview/oracle-jdk-faqs.html>.
8. **Paraschiv, Eugen.** www.baeldung.com. [Online] [Zitat vom: 12. 06 2019.] <https://www.baeldung.com/oracle-jdk-vs-openjdk>.
9. **Wikipedia - Spring Framework**. en.wikipedia.org. [Online] [Zitat vom: 12. 06 2019.] [https://en.wikipedia.org/wiki/Spring\\_Framework](https://en.wikipedia.org/wiki/Spring_Framework).
10. **Pivotal Software.** docs.spring.io. [Online] [Zitat vom: 12. 06 2019.] <https://docs.spring.io/spring/docs/3.2.x/spring-framework-reference/html/beans.html#beans-introduction>.
11. **Google LLC.** dagger.dev. [Online] [Zitat vom: 12. 06 2019.] <https://dagger.dev/>.
12. **Wikipedia- Google Guice**. en.wikipedia.org. [Online] [Cited: 06 12, 2019.] [https://en.wikipedia.org/wiki/Google\\_Guice](https://en.wikipedia.org/wiki/Google_Guice).
13. **Mikusa, Daniel.** community.pivotal.io. [Online] [Zitat vom: 12. 06 2019.] <https://community.pivotal.io/s/question/0D50e000055cYcTCAU/how-are-the-spring-boot-dependencies-curated>.
14. **github - spring-projects/spring-boot**. github.com. [Online] [Zitat vom: 13. 06 2019.] <https://github.com/spring-projects/spring-boot/blob/master/LICENSE.txt>.
15. **github - javafxports/openjdk-jfx**. github.com. [Online] [Zitat vom: 13. 06 2019.] <https://github.com/javafxports/openjdk-jfx/blob/develop/LICENSE>.
16. **Vecteezy - Navy Seal Training Silhouette Free Vector**. www.vecteezy.com. [Online] [Zitat vom: 13. 06 2019.] <https://www.vecteezy.com/vector-art/165405-navy-seal-training-silhouette-free-vector>.