



AG Computergrafik & HCI
apl. Prof. Dr. Achim Ebert

SEP/MP 2019

Bohnanza

Pflichtenheft

5. Mai 2019

Gruppe 12

Lukas Bayer
Nils Fitting
Marco Tücks
Alexander Wilhelm
Lukas Tuchtenhagen

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Projekttreiber	3
1.1 Projektziel	3
1.2 Stakeholders	3
1.3 Aktuelle Lage	4
2 Projektbeschränkungen	5
2.1 Beschränkungen	5
2.2 Glossar	6
2.3 Relevante Fakten und Annahmen	7
3 Funktionale Anforderungen	8
3.1 Systemfunktionen	8
3.2 Beschreibungen der Anwendungsfälle	9
3.3 Produktdaten	22
4 Nicht-funktionale Anforderungen	23
4.1 Softwarearchitektur	23
4.2 Benutzerfreundlichkeit	23
4.3 Leistungsanforderungen	24
4.4 Anforderungen an Einsatzkontext	24
4.5 Anforderungen an Wartung und Unterstützung	25
4.6 Sicherheitsanforderungen	26
4.7 Prüfungsbezogene Anforderungen	27
4.8 Kulturelle und politische Anforderungen	27
4.9 Rechtliche und standardsbezogene Anforderungen	28
4.10 Bedienoberfläche	29
5 Systemtestfälle	36
6 Warteraum	44

Kapitel 1

Projekttreiber

1.1 Projektziel

Im Rahmen des Software-Entwicklungs-Projekts/Modellierungspraktikums 2019 soll ein einfach zu bedienendes Client-Server-System zum Spielen von Bohnanza über ein Netzwerk implementiert werden. Die Benutzeroberfläche soll intuitiv bedienbar sein.

1.2 Stakeholders

/SH10/ **Name:** Spieler

Beschreibung: Menschliche Spieler.

Ziele/Aufgaben: Das Spiel zu spielen.

/SH20/ **Name:** Eltern

Beschreibung: Eltern minderjähriger Spieler.

Ziele/Aufgaben: Um die Spieler zu kümmern, indem Eltern Spielzeit begrenzen wollen und zugriff auf sensible Inhalte begrenzen.

/SH30/ **Name:** Gesetzgeber

Beschreibung: Das Amt für Jugend und Familie.

Ziele/Aufgaben: Die Rechte der Spieler zu schützen und zu gewähren, indem er Gesetze erstellt.

/SH40/ **Name:** Investoren (nur für Beispielszwecken)

Beschreibung: Parteien, die das Finanzmittel für die Entwicklung des Systems bereitstellen.

Ziele/Aufgaben: Gewinn zu ermitteln, indem das System an Endverbraucher verkauft wird.

/SH50/ **Name:** Betreuer

Beschreibung: HiWis, die SEP/MP Projektgruppen betreuen.

Ziele/Aufgaben: Das Entwicklungsprozess zu betreuen, zu überwachen und teilweise zu steuern als auch die Arbeit der Projektgruppen abzunehmen sowie den Studenten im Prozess Hilfe zur Verfügung zu stellen.

/SH60/ **Name:** Prof. Dr. Achim Ebert

Beschreibung: Betreuer des Gesamtprojekts

Ziele/Aufgaben: Als zentraler Ansprechpartner betreut er in Zusammenarbeit mit den HiWis das Projekt und bewertet es schlussendlich.

1.3 Aktuelle Lage

Aktuell wird Bohnanza als Familien-Kartenspiel verwendet und besitzt (für seine Originalvariante) keine technisch realisierte Anwendung. Es sind diverse Erweiterungen, wie auch Spin-Offs des Titels erschienen, auf die in diesem Projekt aus zeitlichen Gründen nicht weiter eingegangen wird. Für uns ist es wichtig, eine solide Alternative zum klassischen Kartenspiel zu entwerfen, die dem Original durch angemessene Musik, Grafik und Interaktion neue Farbe verleiht.

Im weiteren Verlauf sind weder Erweiterungen noch Online-Varianten geplant.

Kapitel 2

Projektbeschränkungen

2.1 Beschränkungen

/LB10/ **Name:** Selbstlehrende Bots

Beschreibung: Keine Selbstlehrfunktion von Bots wird implementiert.

Motivation: Die Funktionalität ist zu aufwändig zu implementieren und passt deshalb nicht in das Zeitbudget.

Erfüllungskriterium: Intelligenzalgorithmus von Bots ist so vorprogrammiert, dass sie Entscheidungen nur anhand des vorprogrammierten Wissens sowie des aktuellen Spielstands treffen, ohne dabei frühere Spiele zu berücksichtigen.

/LB20/ **Name:** Anwendungsbereich

Beschreibung: Das System ist ausschließlich für den privaten Bereich ausgelegt.

Motivation: Im Laufe des Projekts muss die Software lediglich privat getestet und vorgeführt werden.

Erfüllungskriterium: Das Programm kann fehlerfrei vorgeführt werden.

/LB30/ **Name:** Implementierungssprache

Beschreibung: Für die Implementierung ist ausschließlich Java 8 oder höher zu verwenden.

Motivation: Das optimiert die Betreuung von SEP/MP und koordiniert die Mitarbeit.

Erfüllungskriterium: Die Software wurde einheitlich mittels Java 8 implementiert.

/LB40/ **Name:** GUI-Framework

Beschreibung: Die GUI ist mit JavaFX zu realisieren.

Motivation: Das optimiert die Betreuung von SEP/MP und koordiniert die Mitarbeit.

Erfüllungskriterium: Fenster und ihre Funktionen sind für die Nutzer intuitiv gestaltet und somit von von Spieler einfach zu bedienen.

/LB50/ **Name:** Gitlab

Beschreibung: Für die Entwicklung ist das vorgegebene GitLab-Repository zu verwenden.

Motivation: Das optimiert die Betreuung von SEP/MP und koordiniert die Mitarbeit.

Erfüllungskriterium: Die Struktur und Interaktion zwischen den Softwareabschnitten wurde bereits frühzeitig in der Architektur festgehalten.

2.2 Glossar

Deutsch	Englisch	Bedeutung
Bot	bot	Spieler, dessen Spielaktionen vom Computer entschieden und durchgeführt werden
Kekse	Cookies	Offiziell keine gültige Maßnahme zur Bestechung der HiWis
Lobby	lobby	Virtueller Raum zum Betreten eines Spielraums
Spiel (Regelwerk)	game	Bohnanza
Spieler	player	Teilnehmer am Spielgeschehen
Spielraum	game room	Virtueller Raum, in dem ein Spiel stattfindet
Zug	turn	Zustand in dem ein Spieler eine Spielaktion ausführen muss
Spin-off	spinoff	Eigenständiges Spiel mit grundlegender Abwandlung des originalen Spielprinzips
Bohne	bean	Bezeichnung für die in verschiedene Kategorien eingeteilten Spielkarten
Taler	thaler	Eine Währung im Spiel
Bohnenfeld	field	Ablage, auf der Bohnen im Spielverlauf angebaut werden

Deutsch	Englisch	Bedeutung
Hand	hand	Karten, die der Spieler in der Hand hält
Bohnenhandel	bean trading	Möglichkeit der Spieler innerhalb bestimmter Spielphasen Karten miteinander zu tauschen
Bohnenschutzregel	bean protection policy	Eine Regel, die das Abbauen von Bohnen vom Feld einschränkt

2.3 Relevante Fakten und Annahmen

Wichtige gekannte Fakten und getroffene Annahmen, die sich auf das Projekt direkt oder indirekt beziehen und dadurch auf die zukünftige Implementierungsentscheidungen Effekt haben können.

- /FA10/ Name:** Keine Fortentwicklung des App nach SEP/MP.
Beschreibung: Nach Ende des SEP/MP wird das Projekt nicht weiterentwickelt.
Motivation: Das Entwicklungsteam hat keinen Bock darauf.
- /FA20/ Name:** Keine Lizenzen für Spielartefakte.
Beschreibung: Weder TU Kaiserslautern noch die Spielwerk + Freizeit GmbH gewahren dem Entwicklungsteam die Rechte für die Spielartefakte.
Motivation: Rechtliche Vorsorge.
- /FA30/ Name:** Keine bekannte Nachteile von Verwendung von Spielartefakten.
Beschreibung: Es ist nicht bekannt, dass die SEP/MP-Teilnehmer der letzten Jahre irgendwelche rechtliche Probleme dadurch gehabt haben, dass sie die Spielartefakten von Spielwerk + Freizeit GmbH im Rahmen ihrer SEP/MP eingesetzt haben.
Motivation: Rechtliche Vorsorge.

Kapitel 3

Funktionale Anforderungen

3.1 Systemfunktionen

/LF10/ **Name:** Spielverwaltung

Beschreibung: Das System verwaltet das von mehreren Spielern geteilte Spiel in einem Spielraum. Das Spiel erfolgt nach den Spielregeln.

/LF20/ **Name:** Zugriffsverwaltung

Beschreibung: Das System verwaltet den Zugang zum Spiel anhand Benutzerdaten. Spieler können sich registrieren, anmelden, abmelden sowie ihre Kontos löschen.

/LF30/ **Name:** Verwaltung der Spielräume

Beschreibung: Das System verwaltet die Erstellung, Änderung und Löschung der Spielräume.

/LF40/ **Name:** Bestenliste

Beschreibung: Die Anzahl der gewonnenen Spiele aller Spieler anzeigen.

/LF50/ **Name:** Intelligente Bots

Beschreibung: Diese Bots ermöglichen erfahreneren Spielern, gegen einen vergleichbar starken Gegner anzutreten.

/LF60/ **Name:** Chat

Beschreibung: Ein Dialogfenster ermöglicht den Spielern, ihre Angebote in der Tauschphase zu übermitteln und abzuschließen.

3.2 Beschreibungen der Anwendungsfälle

/UC10/ **Name:** Spieler registrieren.

Ziel: Spieler kann Bohnanza spielen.

Akteure: Spieler.

Vorbedingungen: Bohnanza wurde erfolgreich installiert und gestartet.

Eingabedaten: Zugriffsdaten bestehend aus Benutzername, Passwort und Passwort-Wiederholung

Beschreibung:

1. Der Spieler installiert und öffnet Bohnanza.
2. Der Spieler öffnet den Registrieren-Dialog.
3. Der Spieler gibt seine Daten ein und sendet das Formular ab.
4. Das System prüft die Gleichheit von Passwort und Passwort-Wiederholung.
5. Das System prüft auf bereits vorhandene Spieler des gleichen Namens.
6. Das System legt einen neuen Eintrag in der Datenbank an.
7. Der System bestätigt die erfolgreiche Registrierung und leitet den Spieler in die Lobby weiter.

Ausnahmen:

Passwort und Passwort-Wiederholung stimmen nicht überein:

Das System zeigt eine Fehlermeldung an und fordert den Benutzer auf, das Passwort erneut einzugeben.

Benutzername bereits vorhanden: Das System zeigt eine Fehlermeldung an und fordert den Benutzer auf, einen anderen Benutzernamen zu wählen.

Ergebnisse und Outputdaten: Neuer Eintrag (Spieler) in der Datenbank. Spieler befindet sich in der Lobby.

Systemfunktionen /LF20/.

/UC20/ **Name:** Spieler anmelden.

Ziel: Spieler meldet sich im System an.

Akteure: Spieler.

Vorbedingungen Spieler ist im Vorraum.

Eingabedaten: Zugriffsdaten.

Beschreibung:

1. Spieler sendet das Formular ab.
2. Das System prüft die Gültigkeit von Zugangsdaten und bewegt den Spieler in Lobby.

Ausnahmen:

Passwort oder Benutzername ist falsch: Das System zeigt eine Fehlermeldung an, anstatt des Schrittes 2.

Ergebnisse und Outputdaten: Spieler ist im Lobby und sieht die Bestenliste.

Systemfunktionen: /LF20/.

/UC30/ **Name:** Spieler löschen.

Ziel: Spieler entfernt seine Daten aus dem System.

Akteure: Spieler.

Vorbedingungen Spieler ist im Vorraum.

Eingabedaten: Passwort.

Beschreibung:

1. Spieler sendet das Formular ab.
2. Das System prüft die Richtigkeit vom Passwort und fragt Spieler noch ein mal, ob er sich wirklich aus dem System entfernen möchte.
3. Spieler bestätigt seine Intention.
4. Das System entfernt alle Daten des Spielers aus der Datenbank und bewegt Spieler in den Vorraum.

Ausnahmen:

Passwort ist falsch: Das System zeigt eine Fehlermeldung an, anstatt des Schrittes 2.

Keine Löschung erwünscht: Anstatt des Schrittes 4, schließt das System das Dialog.

Ergebnisse und Outputdaten: Spieler ist im Vorraum, Spielerkonto wurde gelöscht.

Systemfunktionen: /LF20/.

/UC40/ **Name:** Spieler abmelden

Ziel: Spieler meldet sich vom System ab.

Akteure: Spieler.

Vorbedingungen Spieler ist im Vorraum.

Eingabedaten: Betätigung des Buttons.

Beschreibung:

1. Spieler betätigt den Abmeldebutton.
2. Das System fragt, ob die Abmeldung wirklich erfolgen soll.
3. Spieler bestätigt seine Intention.
4. Das System meldet den Spieler vom Spiel ab.

Ausnahmen:

Keine

Ergebnisse und Outputdaten: Spieler ist vom System abgemeldet und Programm wird geschlossen.

Systemfunktionen: /LF20/.

/UC50/ **Name:** Neuen Spielraum erstellen.

Ziel: Der Spieler erstellt ein neuen Spielraum in der Lobby

Akteure: Spieler.

Vorbedingungen Spieler ist in der Lobby.

Eingabedaten: Name des Spielraumes, Anzahl der maximalen Spieler (3-5).

Beschreibung:

1. Spieler betätigt "Spiel erstellen"-Button.
2. Das System fordert den Benutzer auf einen Spielnamen und die Anzahl der maximalen Spieler einzugeben.
3. Spieler bestätigt seine Intention.
4. Das System prüft, ob dieser Spielname bereits vorhanden ist.
5. Das System erstellt einen neuen Spielraum in der Lobby.

Ausnahmen:

Spielname bereits vorhanden: Das System zeigt eine Fehlermeldung an und fordert den Spieler auf, einen neuen Spielnamen einzugeben.

Ergebnisse und Outputdaten: Neues Spiel in der Lobby ersichtlich. Spieler ist dem Spiel als "Spielersteller" zugeordnet.

Systemfunktionen: /LF30/.

/UC60/ **Name:** Vorhandenen Spielraum ändern.

Ziel: Der Spielraumersteller ändert die Einstellungen des Spielraumes.

Akteure: Spieler (Spielersteller).

Vorbedingungen Spieler ist im Spielraum.

Eingabedaten: Name des Spielraumes, Anzahl der maximalen Spieler (3-5), Anzahl der Bots (1-4).

Beschreibung:

1. Spieler betätigt "Ändern"-Button.
2. Das System fordert den Benutzer auf den Spielnamen, die Anzahl der maximalen Spieler und die Anzahl der Bots zu ändern.
3. Spieler bestätigt seine Intention.
4. Das System prüft, ob dieser Spielname bereits vorhanden ist.
5. Das System prüft, ob die Anzahl der zulässigen maximalen Spieler durch die Änderungen und durch das Hinzufügen der Bots überschritten wurde.
5. Das System ändert die Daten des Spielraumes.

Ausnahmen:

Spielname bereits vorhanden: Das System zeigt eine Fehlermeldung an und fordert den Spieler auf, einen anderen Spielnamen einzugeben.

Anzahl der maximalen Spieler überschritten: Das System zeigt eine Fehlermeldung an und fordert den Spieler auf, Bots zu entfernen oder die Anzahl der maximalen Spieler zu erhöhen.

Ergebnisse und Outputdaten: Daten des Spielraumes aktualisiert.

Systemfunktionen: /LF30/.

/UC70/ **Name:** Spielraum löschen.

Ziel: Der Spielraumersteller löscht den Spielraum.

Akteure: Spieler (Spielersteller).

Vorbedingungen Spieler ist im Spielraum.

Eingabedaten: Bestätigungs-Button.

Beschreibung:

1. Spieler betätigt "Löschen"-Button.
2. Das System fragt nach, ob der Spielraum wirklich gelöscht werden soll.
3. Spieler bestätigt seine Intention.
4. Das System löscht den Spielraum.
5. Das System leitet den Spielersteller und weitere Spieler in die

Lobby zurück.

5. Das System meldet "Spielraum gelöscht".

Ausnahmen:

keine

Ergebnisse und Outputdaten: Spielraum gelöscht.

Systemfunktionen: /LF30/.

/UC80/ **Name:** Spielraum beitreten.

Ziel: Der Spieler tritt einem Spielraum bei.

Akteure: Spieler.

Vorbedingungen Spieler ist in der Lobby.

Eingabedaten: Bestätigungs-Button.

Beschreibung:

1. Spieler betätigt "Beitreten"-Button.
2. Das System prüft, ob die maximale Anzahl der Spieler überschritten wird.
3. Das System fügt den Spieler dem Spielraum hinzu.
4. Das System meldet den anderen Teilnehmern "Neuer Spieler beigetreten".

Ausnahmen:

Spielraum voll: Das System meldet dem Spieler: Spielraum bereits voll, bitte ein anderes Spiel auswählen".

Ergebnisse und Outputdaten: Spielraum gelöscht.

Systemfunktionen: /LF30/.

/UC90/ **Name:** Spielraum verlassen.

Ziel: Der Spieler verlässt den Spielraum

Akteure: Spieler.

Vorbedingungen Spieler ist im Spielraum.

Eingabedaten: Bestätigungs-Button.

Beschreibung:

1. Der Spieler betätigt "Verlassen"-Button.
2. Das System fordert Bestätigung.
3. Der Spieler bestätigt seine Intention.
4. Das System entfernt den Spieler aus dem Spielraum und leitet

ihn in die Lobby zurück.

4. Das System meldet den anderen Teilnehmern "Spieler hat Spielraum verlassen".

Ausnahmen:

Spielerersteller verlässt den Spielraum: Das System meldet dem Spieler: "Bitte Spielraum zuerst löschen".

Ergebnisse und Outputdaten: Spielraum verlassen. Spieler in Lobby.

Systemfunktionen: /LF30/.

/UC100/ **Name:** Spiel starten.

Ziel: Der Spielraumersteller startet das Spiel.

Akteure: Spieler (Spielersteller).

Vorbedingungen Spieler ist im Spielraum. Die Anzahl der Spieler inklusive Bots im Spielraum beträgt mindestens 3.

Eingabedaten: Bestätigungs-Button.

Beschreibung:

1. Spieler betätigt "Starten"-Button.
2. Das System fragt nach, ob das Spiel gestartet werden soll.
3. Spieler bestätigt seine Intention.
4. Das System entfernt das Spiel aus dem Spielraum.
5. Das System startet ein Spiel.
6. Das System bestimmt zufällig einen Startspieler.
7. Das System vergibt jedem Spieler fünf Handkarten.
8. Das System zeigt eine Meldung "Spiel beginnt".
9. Das System fordert den Startspieler dazu auf, den Zug zu beginnen.

Ausnahmen:

Nicht genügend Spieler vorhanden: Das System zeigt eine Meldung: "Nicht genügend Spieler zum Starten des Spiels vorhanden".

Ergebnisse und Outputdaten: Spiel gestartet. Die Spieler befinden sich in der Spieloberfläche.

Systemfunktionen: /LF30/.

/UC110/ **Name:** Spiel verlassen.

Ziel: Der Spieler verlässt das Spiel.

Akteure: Spieler.

Vorbedingungen Spieler ist im Spiel.

Eingabedaten: Bestätigungs-Button.

Beschreibung:

1. Der Spieler betätigt "Verlassen"-Button.
2. Das System fordert Bestätigung.
3. Der Spieler bestätigt seine Intention.
4. Das System entfernt den Spieler aus dem Spiel und leitet ihn in die Lobby zurück.
5. Das System meldet den anderen Teilnehmern "Spieler hat das Spiel verlassen".
6. Das System prüft, ob sich weniger als drei Spieler im Spielraum befinden und fügt ggf. einen Bot hinzu.
7. Das System prüft, ob der Startspieler das Spiel verlassen hat und bestimmt einen Startspieler links vom ehemaligen Startspieler.

Ausnahmen:

Spielerersteller verlässt den Spielraum: Das System meldet dem Spieler: "Spiel wird geschlossen, wenn Sie das Spiel verlassen - denken Sie über Ihre Entscheidung nach".

Ergebnisse und Outputdaten: Spiel verlassen. Spieler befindet sich in Lobby.

Systemfunktionen: /LF10/.

/UC120/ **Name:** Phase 1.

Ziel: Phase 1 durchlaufen.

Akteure: Spieler.

Vorbedingungen Spiel frisch gestartet oder Phase 4 beendet, Handkarten vorhanden.

Eingabedaten: Bestätigungs-Button, Handkarten in richtiger Reihenfolge.

Beschreibung:

1. System prüft, ob menschlicher Spieler oder Bot an der Reihe ist und ruft ggf. /UC200/ auf.
2. Spieler baut Handkarten an.
3. Der Spieler beendet die Phase.
4. Das System leitet Phase 2 ein.

Ausnahmen:

Kein freies Anbaufeld vorhanden: Das System zeigt eine Meldung: "Kein freies Anbaufeld vorhanden, Felder abernten (/UC130/)"

Ergebnisse und Outputdaten: Handkarten angebaut. Phase 1 beendet.

Systemfunktionen: /LF10/.

/UC130/ **Name:** Bohnenkarten abernten.

Ziel: Der Spieler baut Bohnenkarten ab.

Akteure: Spieler.

Vorbedingungen keine

Eingabedaten: Bestätigungs-Button.

Beschreibung:

1. Spieler betätigt "Ernten"-Button.
2. Das System fragt nach, ob die Bohnen geerntet werden sollen.
3. Spieler bestätigt seine Intention.
4. Das System prüft "Bohnenschutzregel".
4. Das System entfernt die Karten vom Ablagefeld
5. Das System legt Karten in der Anzahl der erhaltenen Taler auf das Talerfeld.
6. Das System legt die restlichen Karten auf den Ablagestapel.

Ausnahmen:

Bohnenschutzregel verletzt: Das System meldet "Bohnenschutzregel verletzt, Aktion nicht möglich".

Ergebnisse und Outputdaten: Bohnenfeld leer, Taler des Spielers ggf. erhöht.

Systemfunktionen: /LF30/.

/UC140/ **Name:** Phase 2.

Ziel: Phase 2 durchlaufen.

Akteure: Spieler.

Vorbedingungen Phase 1 beendet.

Eingabedaten: Bestätigungs-Button

Beschreibung:

1. Das System legt zwei Karten vom Nachziehstapel daneben und prüft /UC210/.
2. Das System fordert den Spieler auf die Karten zu nehmen oder

zu handeln (/UC150/)

3. Das System wartet ggf. /UC150/ ab.
4. Das System legt die übrigen Karten neben dem Nachziehstapel quer neben das Anbaufeld.
5. Der aktive Spieler beendet die Phase.
6. Das System leitet Phase 3 ein.

Ausnahmen:

keine

Ergebnisse und Outputdaten: Neue Karten liegen neben dem Bohnenfeld. Phase 2 beendet.

Systemfunktionen: /LF10/.

/UC150/ **Name:** Handel.

Ziel: Handel abschließen.

Akteure: Spieler.

Vorbedingungen Phase 2 läuft.

Eingabedaten: Bestätigungs-Button, Bohnenkarten

Beschreibung:

1. Der aktive Spieler wählt einen anderen Spieler zum Handeln aus.
2. Die Spieler bestätigen den Handel.
3. Das System tauschen die Karten aus.
4. Der aktive Spieler beendet den Handel.
5. Das System legt die neu erworbenen Karten neben das Bohnenfeld.
6. Der aktive Spieler wiederholt ggf. Schritt 1-4.
7. Der Spieler beendet seine Handelsphase.

Ausnahmen:

Handel abgebrochen: Das System "Handel abgebrochen", wenn einer der Spieler den Handel nicht bestätigt oder das Handelfenster schließt.

Ergebnisse und Outputdaten: Handelskarten neben dem Bohnenfeld. Handelsphase abgeschlossen.

Systemfunktionen: /LF10/.

/UC160/ **Name:** Phase 3.

Ziel: Phase 3 durchlaufen.

Akteure: Spieler.

Vorbedingungen Phase 2 ist abgeschlossen.

Eingabedaten: Bestätigungs-Button

Beschreibung:

1. Das System prüft, ob freie Felder zum Anbauen der Bohnenkarten vorhanden sind.
2. Alle Spieler bauen die Bohnenkarten, erworben durch den Handel ([/UC150/](#)) oder vom Nachziehstapel (aktiver Spieler, [/UC140/](#)) an.
3. Das System beendet die Phase, sobald alle Spieler ihre Karten angebaut haben.

Ausnahmen:

Kein freies Anbaufeld: Das System meldet "kein freies Anbaufeld vorhanden".

Ergebnisse und Outputdaten: Erworbene Bohnenkarten aus Phase 2 wurden angebaut.

Systemfunktionen: [/LF10/](#).

[/UC170/](#) **Name:** Phase 4.

Ziel: Phase 4 durchlaufen.

Akteure: Spieler.

Vorbedingungen Phase 3 ist abgeschlossen.

Eingabedaten: Bestätigungs-Button

Beschreibung:

1. Das System prüft [/UC210/](#) und gibt dem aktiven Spieler drei neue Handkarten in entsprechender Reihenfolge.
2. Das System beendet Phase 4.
3. Das System übergibt Spielrunde dem nächsten Spieler und beginnt dort mit Phase 1.

Ausnahmen:

keine

Ergebnisse und Outputdaten: Drei neue Handkarten erworben. Aktiver Spieler gewechselt.

Systemfunktionen: [/LF10/](#).

[/UC180/](#) **Name:** Bestenliste.

Ziel: Das System zeigt eine Bestenliste an und hält diese aktuell.

Akteure: System.

Vorbedingungen Es existieren Einträge in der Datenbank.

Eingabedaten: Einträge bisher gespielter Spiele.

Beschreibung:

1. Das System stellt eine Anfrage an die Datenbank.
2. Das System zeigt die von jedem Spieler die Anzahl der gewonnenen Spiele an.

Ausnahmen:

Keine Einträge: Sind keine Einträge vorhanden, so zeigt das System "Keine Einträge vorhanden".

Ergebnisse und Outputdaten: Bestenliste.

Systemfunktionen: /LF40/.

/UC190/ **Name:** Chatverwaltung.

Ziel: Das System zeigt im Spielraum, in der Lobby und im Spiel ein Chatfenster.

Akteure: System, Spieler.

Vorbedingungen Es existiert ein Spiel oder ein Spielraum.

Eingabedaten: Chateingaben der Nutzer.

Beschreibung:

1. Das System zeigt einen Chat an.
2. Spieler können Chatnachrichten absenden.
3. Das System zeigt die Chatnachricht zusammen mit dem Spielernamen an.

Ausnahmen:

Keine Einträge: Ist das Chatfenster leer, zeigt das System die Meldung "Keine Chatnachrichten vorhanden".

Ergebnisse und Outputdaten: Chat mit aktuellen Chatnachrichten.

Systemfunktionen: /LF60/.

/UC200/ **Name:** Botverwaltung.

Ziel: Das System steuert einen Bot.

Akteure: System.

Vorbedingungen Es existiert ein Spiel oder ein Spielraum.

Eingabedaten: Chateingaben der Nutzer.

Beschreibung:

1. Das System steuert einen Bot nach den Phasen des Spiels: /UC120/, /UC140/, /UC160/ und /UC170/ und nach vorgegebenen Schwierigkeitsstufen /LF50/.

Ausnahmen:

keine

Ergebnisse und Outputdaten: Aktionen des Bots entsprechend der Phasen des Spiels und der Schwierigkeitsstufe des Bots.

Systemfunktionen: ??.

/UC210/ **Name:** Spielverwaltung.

Ziel: Das System verwaltet die Phasen eines Spiels.

Akteure: System.

Vorbedingungen Das Spiel wurde aus dem Spielraum heraus gestartet.

Eingabedaten: Aktion des Spielers.

Beschreibung:

1. Das System prüft, ob Nachziehstapel leer ist. 2. Das System erneuert bei Bedarf den Nachziehstapel mithilfe des Ablegestapels. 3. Das System prüft, wie oft der Ziehstapel erneuert wurde. Bei dreimaligem Leeren beendet das System nach Ablauf der Phasen 2-3 (Trigger in Phase 2 /UC140/) oder sofort (Trigger in Phase 4 /UC170/) das Spiel. 4. Das System meldet "Nach Phase 2 und 3 wird das Spiel beendet oder SSpil wird beendet". 5. Das System ruft usecase "Spiel beenden (/UC220/)" auf.

Ausnahmen:

keine

Ergebnisse und Outputdaten: Nächste Phase des Spiels oder Beendigung des Spiels.

Systemfunktionen: /LF30/.

/UC220/ **Name:** Spiel beenden.

Ziel: Das System beendet das Spiel.

Akteure: System.

Vorbedingungen Der Ziehstapel wurde dreimal neu gemischt, die letzten Phasen der Runde sind beendet.

Eingabedaten: keine

Beschreibung:

1. Das System bestimmt den Gewinner des Spiels.
2. Das System meldet den Spielern die Reihenfolge ihrer Platzierung.
3. Das System aktualisiert die Daten des Gewinners in der Datenbank.
4. Das System beendet das Spiel und leitet die Spieler in die Lobby weiter.

Ausnahmen:

keine

Ergebnisse und Outputdaten: Spieler befinden sich in der Lobby. "Gewonnene Spiele" des Gewinners in der Datenbank aktualisiert.

Systemfunktionen: /LF10/.

3.3 Produktdaten

Hier sollen die Daten genannt werden, die im System verwendet werden.

- /LD10/ Name:** Benutzername*¹
Fachliche Beschreibung: Benutzername des Spielers
Relevante Systemfunktionen: /LF10/, /LF20/
- /LD20/ Name:** Passwort*
Fachliche Beschreibung: Passwort des Spielers
Relevante Systemfunktionen: /LF20/
- /LD30/ Name:** Gewonnene Spiele*
Fachliche Beschreibung: Anzahl der gewonnenen Spiele eines Spielers
Relevante Systemfunktionen: /LF40/
- /LD40/ Name:** Spielzeit*
Fachliche Beschreibung: Spielzeit eines Spielers, damit Eltern die Spielzeit ihrer Kinder einschränken können.
Relevante Systemfunktionen: /LF20/
- /LD50/ Name:** Taler
Fachliche Beschreibung: Taler eines Spielers
Relevante Systemfunktionen: ??
- /LD60/ Name:** Chat
Fachliche Beschreibung: Chatnachrichten innerhalb eines Chats.
Relevante Systemfunktionen: ??
- /LD70/ Name:** Spielkarten
Fachliche Beschreibung: Anzahl der ausgegebenen Spielkarten.
Relevant für intelligente Bots.
Relevante Systemfunktionen: ??

¹ “*” bedeutet hier, dass die Daten in der Datenbank zu speichern sind

Kapitel 4

Nicht-funktionale Anforderungen

4.1 Softwarearchitektur

/NF10/ **Name:** Client-Server Anwendung

Beschreibung: Das verteilte Spiele-System ermöglicht das gemeinsame Spielen von verschiedenen Rechnern aus.

Motivation: Aufgabestellung v. SEP/MP.

Erfüllungskriterium: Das fertige System besteht aus Client- und Server-Teilen.

/NF20/ **Name:** Plattformunabhängigkeit

Beschreibung: Es soll sich um eine plattformunabhängige Anwendung handeln. Zumindest Windows- und Linuxsysteme sind zu unterstützen.

Motivation: Aufgabenstellung v. SEP/MP.

Erfüllungskriterium: In den Präsentationen kann der aktuelle Stand bzw. das fertige Produkt jederzeit vorgeführt werden.

4.2 Benutzerfreundlichkeit

/NF30/ **Name:** Benutzeralter

Beschreibung: Das System ist für Benutzer geeignet, die älter als 5 Jahre sind.

Motivation: Jüngere Benutzer sind unfähig das Spiel zu spielen.

Erfüllungskriterium: In AGB steht ein entsprechendes Hinweis.

/NF40/ **Name:** Technische Fähigkeiten

Beschreibung: Besondere technische Fähigkeiten sind von der Benutzern nicht zu erwarten.

Motivation: Auch die Menschen, die kaum etwas von Bedienung bzw. Programmierung von Rechnern verstehen, sollen fähig sein, das System zu verwenden.

Erfüllungskriterium: Jeder Benutzer kann die verschiedenen Funktionen intuitiv bedienen und spielen.

4.3 Leistungsanforderungen

/NF50/ **Name:** Antwortzeit

Beschreibung: Maximale Antwortzeit für alle Systemprozesse.

Motivation: Das System muss immer brauchbar sein.

Erfüllungskriterium: Das System antwortet auf Benutzerhandlungen nie später als in 10 Sekunden.

4.4 Anforderungen an Einsatzkontext

Anforderungen an physische Umgebung

/NF60/ **Name:** Lauffähigkeit an SCI-Rechnern

Beschreibung: Das Produkt muss auf einem eigenem Geräte lauffähig sein, welches zur Präsentation am Ende des SEPs genutzt werden muss. Falls keine eigenen Rechner vorhanden, stehen auch die SCI-Terminals zur Verfügung.

Motivation: Optimierung von Betreuung und Abnahme des SEP/MP

Erfüllungskriterium: Das fertige Produkt kann ohne Probleme an den vorhandenen Rechnern vorgeführt werden.

Absatz- sowie Installationsbezogene Anforderungen

/NF70/ **Name:** Installationsanleitung

Beschreibung: Falls die Installation nicht lediglich das Öffnen einer Datei voraussetzt, muss der genaue Installations- und Startvorgang schriftlich für Benutzer zur Verfügung gestellt.

Motivation: Spezifikation

Erfüllungskriterium: Eventuell auftretende Probleme bei der Installation können von jedem Nutzer anhand der Anleitung umgangen werden.

Anforderungen an Versionierung

/NF80/ **Name:** Keine weitere Versionen

Beschreibung: Nach Version 1.0 keine weitere Entwicklung ist vorgesehen.

Motivation: Das ist nur SEP/MP, keine Geschäftsprojekt, siehe [/FA10/](#)

Erfüllungskriterium: Es werden keine Updates hinzugefügt.

4.5 Anforderungen an Wartung und Unterstützung

Wartungsanforderungen

/NF90/ **Name:** Wartung

Beschreibung: Es ist keine Wartung von Seiten der Programmierer oder Dritter vorgesehen.

Motivation: Es sind weder Updates noch eine weitere Verwendung des Programms nach der Abnahme vorgesehen.

Erfüllungskriterium: Das Programm wird nach der Abschlusspräsentation nicht weiterentwickelt.

/NF100/ **Name:** Dokumentation

Beschreibung: Der Quellcode muss ausführlich dokumentiert werden.

Motivation: HiWis und der betreuende Professor sollen Entwicklungsschritte schneller nachvollziehen können.

Erfüllungskriterium: JavaDoc

/NF110/ **Name:** Testen

Beschreibung: Der Quellcode außer GUI muss gut getestet werden.

Motivation: Fehler sollen frühzeitig erkannt und ausgebessert werden.

Erfüllungskriterium: Von Unit-Tests muss mindestens 70% des Quellcodes bedeckt werden. GUI-Klassen sind aus der Anforderung ausgenommen.

Anforderungen an technische und fachliche Unterstützung

/NF120/ **Name:** Beispiel

Beschreibung: Keine technische und fachliche Unterstützung des Systems ist geplant.

Motivation: Siehe [/FA10/](#).

Erfüllungskriterium: Nicht anwendbar.

Anforderungen an technische Kompatibilität

/NF130/ **Name:** Technische Kompatibilität

Beschreibung: Das Programm interagiert korrekt mit aller vorhergesehenen Soft- und Hardware.

Motivation: Das Programm wird nicht von verschiedenen Kompatibilitätsfaktoren eingeschränkt.

Erfüllungskriterium: Das Programm kann in den vorgegebenen Entwicklungsumgebungen eingesetzt werden.

4.6 Sicherheitsanforderungen

Zugang

/NF140/ **Name:** Zugang

Beschreibung: Es wird nur Spielern Zugang gewährt, die ein registriertes Konto besitzen und sich darüber anmelden.

Motivation: Spieler und ihre Aktionen sollen auf ein Konto zurückverfolgbar sein.

Erfüllungskriterium: Niemand kann die Software ohne gültigen Account nutzen.

Integrität

/NF150/ **Name:** Modifikation des Spiels

Beschreibung: Spiele können nicht von anderen Spielern verändert werden.

Motivation: Schummeln und Ärgern anderer Mitspieler ist unerwünscht.

Erfüllungskriterium: Es wird bei der Implementierung auf gängige Sicherheitsverfahren geachtet. Die Spieler haben keine Möglichkeit, über die Programmoberfläche in den Quellcode des Spiels einzugreifen.

Datenschutz/Privatsphäre

/NF160/ **Name:** Datenschutz/Privatsphäre

Beschreibung: Private Spielerdaten sollen nicht von Dritten eingesehen oder für diese nutzbar gemacht werden.

Motivation: Der Nutzer hat ein Recht darauf, dass die genutzte Software Datenschutz gewährleistet.

Erfüllungskriterium: Es wird kein Rechtsbruch durch die Nutzung unserer Software auftreten.

Virenschutz

/NF170/ **Name:** Virenschutz

Beschreibung: Es ist kein explizierter Virenschutz vorgesehen.

Motivation: Die Software läuft lediglich über ein lokales Netzwerk, wodurch keine Malware von außen Einfluss nehmen kann.

Erfüllungskriterium: Weder innerhalb unserer Software noch in anderen Programmen lassen sich Viren finden, die durch Sicherheitslücken in unserer Implementierung entstanden sind.

4.7 Prüfungsbezogene Anforderungen

Anforderungen, die sich auf die Prüfung/Audit vom System von SEP/MP-Tutoren oder von weiteren Instanzen beziehen.

/NF180/ **Name:** Formate der Systemdokumentation

Beschreibung: Systemdokumentation muss in 2 Formen geführt werden (wenn anwendbar): Die Ausgangsdateien (L^AT_EX, Dateien von Diagrammensoftware, von Grafiksoftware usw.) und PDFs.

Motivation: Optimierung der SEP/MP-Betreuung.

Erfüllungskriterium: Siehe Beschreibung.

4.8 Kulturelle und politische Anforderungen

/NF190/ **Name:** Systemsprache

Beschreibung: Die Interfacesprache ist Deutsch.

Motivation: Synchronisation des Verständnisses von Teammitgliedern mit unterschiedlichen kulturellen Hintergründen.

Erfüllungskriterium: Es tauchen keine weiteren Sprachen (Anglizismen ausgenommen) in der Software auf.

4.9 Rechtliche und standartsbezogene Anforderungen

/NF200/ **Name:** Nicht rechtliche Anforderungen

Beschreibung: Es gibt keine relevanten rechtliche Anforderungen bekannt.

Motivation: Siehe [/FA10/](#).

Erfüllungskriterium: Nicht anwendbar.

4.10 Bedienoberfläche

/GUI10/ **Name:** Anmeldung

Beschreibung: Interface für Anmeldung

Relevante Systemfunktionen: [/LF20/](#)

Abbildungen: [4.1](#)

/GUI20/ **Name:** Registrierung

Beschreibung: Interface zum Registrieren

Relevante Systemfunktionen: [/LF20/](#)

Abbildungen: [4.2](#)

/GUI30/ **Name:** Lobby

Beschreibung: Ansicht der Lobby

Relevante Systemfunktionen: [/LF30/](#)

Abbildungen: [4.3](#)

/GUI40/ **Name:** Spielraum erstellen

Beschreibung: Ansicht, um einen Spielraum zu erstellen.

Relevante Systemfunktionen: [/LF30/](#)

Abbildungen: [4.4](#)

/GUI50/ **Name:** Spielraum

Beschreibung: Ansicht auf einen Spielraum.

Relevante Systemfunktionen: [/LF30/](#)

Abbildungen: [4.5](#)

/GUI60/ **Name:** Spielübersicht

Beschreibung: Ansicht auf das Spielfeld.

Relevante Systemfunktionen: [/LF10/](#)

Abbildungen: [4.6](#)

/GUI70/ **Name:** Handelfenster

Beschreibung: Ansicht des Handelfensters.

Relevante Systemfunktionen: [/LF10/](#)

Abbildungen: [4.7](#)

/GUI80/ **Name:** Gewinnerübersicht



← Spiel beenden Anmeldung Info

BOHNANZA®

Spielername:

Passwort:

Registrieren Anmelden

Abbildung 4.1: Fenster zur Anmeldung

Beschreibung: Übersicht des Spielergebnisses.

Relevante Systemfunktionen: [/LF10/](#)

Abbildungen: [4.8](#)

/GUI90/ Name: Gesamtübersicht

Beschreibung: Übersicht über alle Spielfenster.

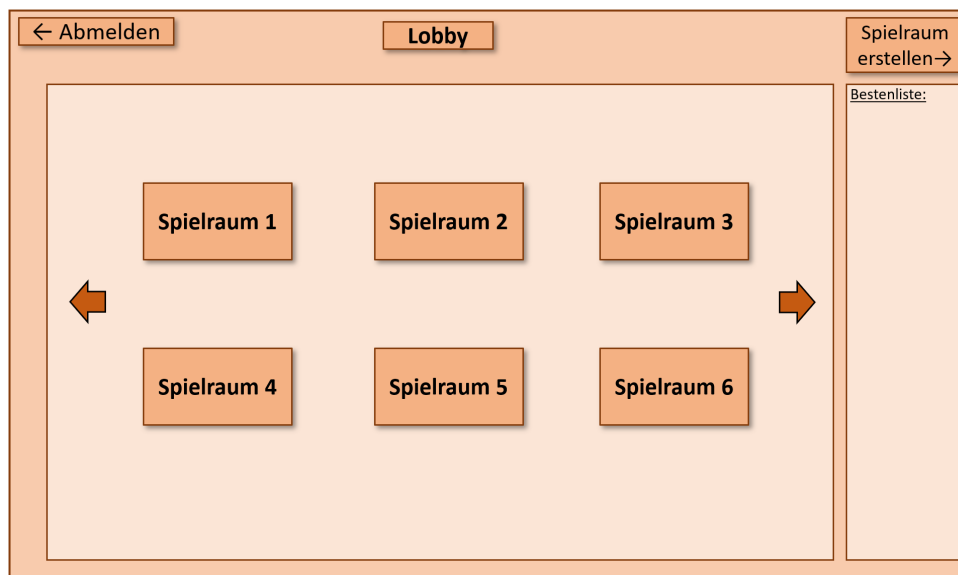
Relevante Systemfunktionen: [/LF10/](#)

Abbildungen: [4.9](#)



The diagram shows a registration window with an orange background. At the top center is a button labeled "Registrierung". Below it are three input fields: "Spielername:", "Passwort:", and "Passwort wiederholen:". At the bottom left is a button labeled "← zurück" and at the bottom right is a button labeled "Registrieren".

Abbildung 4.2: Fenster zum Registrieren



The diagram shows a lobby view with an orange background. At the top left is a button labeled "← Abmelden". At the top center is a button labeled "Lobby". At the top right is a button labeled "Spielraum erstellen→". Below these buttons is a large central area containing six buttons labeled "Spielraum 1", "Spielraum 2", "Spielraum 3", "Spielraum 4", "Spielraum 5", and "Spielraum 6" arranged in two rows of three. To the left of this central area is a left-pointing arrow, and to the right is a right-pointing arrow. On the far right is a vertical panel labeled "Bestenliste:".

Abbildung 4.3: Ansicht der Lobby

Spiel erstellen

Spielname:

Anzahl Spieler:

Anzahl Bots:

← zurück Erstellen

Abbildung 4.4: Fenster zum Spielerstellen

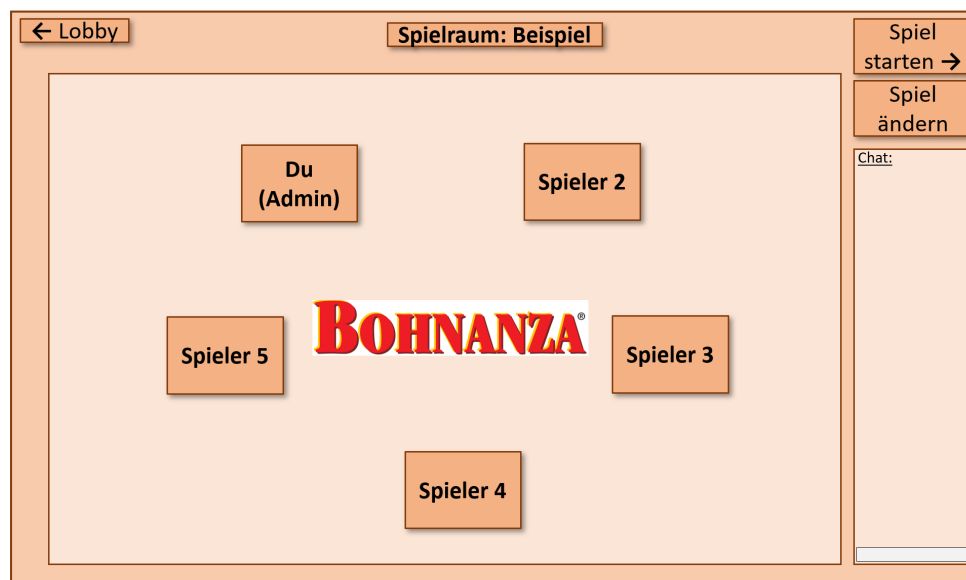


Abbildung 4.5: Ansicht auf einen Spielraum

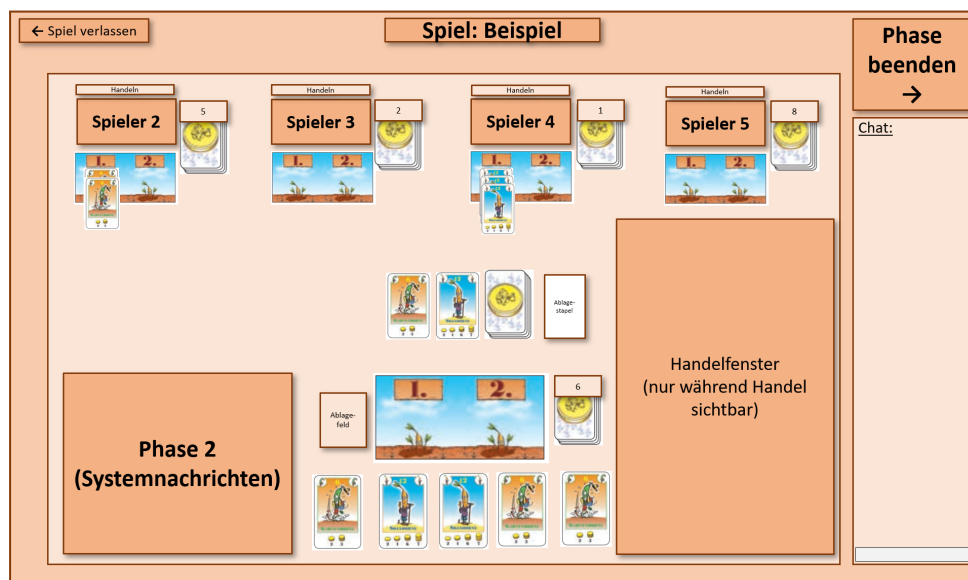


Abbildung 4.6: Ansicht auf das Spielfeld



Abbildung 4.7: Ansicht des Handelfensters

Gewinner: Spieler 2		
Platzierung	Spieler	Punkte
1.	Spieler 2	13
2.	Spieler 5	11
3.	Spieler 1	10
4.	Spieler 3	9
5.	Spieler 4	7

Abbildung 4.8: Ansicht des Gewinnerfensters

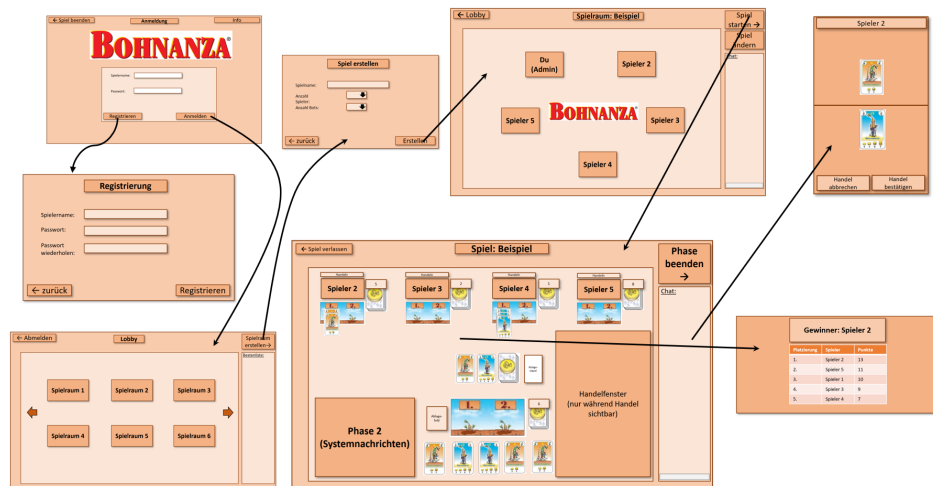


Abbildung 4.9: Übersicht über alle Spielfenster

Kapitel 5

Systemtestfälle

Hier sollen verschiedene Szenarien beschrieben werden, mithilfe deren Sie später Systemtests ausführen, sowie die erwartete Ergebnisse.

/TF10/ Name: Spieler anmelden.

Motivation: Testet, ob die Anmeldung in das System korrekt funktioniert.

Szenarien:

1. *Zugriffsdaten sind vorhanden und richtig*
⇒ Spieler wird in Lobby bewegt.
2. *Benutzername ist registriert, Passwort ist falsch*
⇒ Fehlermeldung wird angezeigt.
3. *Benutzername ist nicht registriert*
⇒ Fehlermeldung wird angezeigt.

Relevante Systemfunktionen: /LF20/

Relevante Use Cases: /UC20/

/TF20/ Name: Spieler registrieren

Motivation: Testet, ob die Registrierung in das System korrekt funktioniert.

Szenarien:

1. *Benutzername ist nicht vergeben und Passwort/Passwort-Wiederholung stimmen überein*
⇒ Spieler wird registriert und in Lobby bewegt.
2. *Benutzername ist vergeben*
⇒ Fehlermeldung wird angezeigt.
3. *Passwort und Passwort-Wiederholung stimmen nicht überein*
⇒ Fehlermeldung wird angezeigt.

Relevante Systemfunktionen: /LF20/

Relevante Use Cases: /UC10/

/TF30/ **Name:** Spieler löschen.

Motivation: Testet, ob sich der Spieler aus dem System löschen kann.

Scenarien:

1. *Passwort ist richtig*
⇒ Spieler wird gelöscht und zum Anmeldebildschirm geleitet.
2. *Passwort ist falsch*
⇒ Fehlermeldung wird angezeigt.

Relevante Systemfunktionen: /LF20/

Relevante Use Cases: /UC30/

/TF40/ **Name:** Spieler abmelden.

Motivation: Testet, ob die Abmeldung vom System korrekt funktioniert.

Scenarien:

1. *Button erfolgreich betätigt*
⇒ Spieler wird aus der Lobby bewegt.

Relevante Systemfunktionen: /LF20/

Relevante Use Cases: /UC40/

/TF50/ **Name:** Spielraum erstellen.

Motivation: Testet, ob ein Spielraum erstellt werden kann.

Scenarien:

1. *Eingabeoptionen sind richtig*
⇒ Spieler wird in den Spielraum bewegt und Spielraum wird für andere Spieler sichtbar.
2. *Eingabeoptionen sind ungültig*
⇒ Fehlermeldung wird angezeigt.
3. *Spielraumname bereits vorhanden*
⇒ Fehlermeldung wird angezeigt.

Relevante Systemfunktionen: /LF30/

Relevante Use Cases: /UC50/

/TF60/ **Name:** Spielraum ändern.

Motivation: Testet, ob der Spielraum geändert werden kann.

Scenarien:

1. *Änderung liegen in den vorgegebenen Bedingungen*
⇒ Spielraum wird geändert.
2. *Änderung liegen nicht in den vorgegebenen Bedingungen (Anzahl Spieler/Bots)*
⇒ Fehlermeldung wird angezeigt.
3. *Spielraumname bereits vergeben*
⇒ Fehlermeldung wird angezeigt.

Relevante Systemfunktionen: /LF30/

Relevante Use Cases: /UC60/

/TF70/ **Name:** Spielraum löschen.

Motivation: Testet, ob ein Spielraum gelöscht werden kann.

Scenarien:

1. *Löschung wurde bestätigt*
⇒ Spieler werden in Lobby bewegt und Spielraum steht in der Lobby nicht mehr zur Verfügung.
2. *Löschung wurde nicht bestätigt*
⇒ Spielraum bleibt bestehen.

Relevante Systemfunktionen: /LF30/

Relevante Use Cases: /UC70/

/TF80/ **Name:** Spielraum beitreten.

Motivation: Testet, ob ein Spielraum betreten werden kann.

Scenarien:

1. *Button wurde betätigt*
⇒ Spieler wird in den Spielraum bewegt.

Relevante Systemfunktionen: /LF30/

Relevante Use Cases: /UC80/

/TF90/ **Name:** Spielraum verlassen.

Motivation: Testet, ob ein Spielraum verlassen werden kann.

Scenarien:

1. *Button wurde betätigt*
⇒ Spieler wird in die Lobby bewegt.
2. *Spielraumersteller möchte Spielraum verlassen*
⇒ System meldet "Bitte Spielraum zuerst löschen".

Relevante Systemfunktionen: /LF30/

Relevante Use Cases: /UC90/

/TF100/ Name: Spiel starten.

Motivation: Testet, ob ein Spiel gestartet werden kann.

Scenarien:

1. *Start wurde bestätigt und Vorbedingungen sind erfüllt*
⇒ Spiel wird gestartet, Spielraum wird aus der Lobby entfernt, Startspieler wird bestimmt, Handkarten werden ausgegeben.
2. *Start wurde nicht bestätigt*
⇒ Spielraum bleibt bestehen.
3. *Start wurde bestätigt und Vorbedingungen sind nicht erfüllt*
⇒ Fehlermeldung wird angezeigt.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC100/

/TF110/ Name: Spiel verlassen.

Motivation: Testet, ob ein Spiel verlassen werden kann.

Scenarien:

1. *Verlassen wurde bestätigt*
⇒ Spieler wird aus dem Spiel entfernt, Bots werden ggf. hinzugefügt, Startmarke wird ggf. neu zugewiesen.
2. *Verlassen wurde nicht bestätigt*
⇒ Spieler bleibt im Spiel.
3. *Spielersteller (Host) möchte Spiel verlassen*
⇒ System meldet "Spiel wird geschlossen, wenn Sie das Spiel verlassen - denken Sie über Ihre Entscheidung nach"

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC110/

/TF120/ Name: Bestenliste anzeigen.

Motivation: Testet, ob die Bestenliste ordnungsgemäß angezeigt wird.

Scenarien:

1. *Einträge in der Datenbank vorhanden*
⇒ Bestenliste zeigt Spieler mit den meisten Wins in absteigender Reihenfolge.
2. *Keine Einträge in der Datenbank vorhanden*
⇒ Bestenliste zeigt "Keine Einträge vorhanden"

Relevante Systemfunktionen: /LF40/

Relevante Use Cases: /UC180/

/TF130/ **Name:** Chatten.

Motivation: Testet, ob Chatnachrichten ordnungsgemäß angezeigt werden.

Scenarien:

1. *Chat wird initialisiert*
⇒ Spieler sehen ein Chatfenster.
2. *Chatnachricht wird abgeschickt*
⇒ Spieler sehen Chatnachricht des Verfassers.

Relevante Systemfunktionen: /LF60/

Relevante Use Cases: /UC190/

/TF140/ **Name:** Bot.

Motivation: Testet, ob Bot die einzelnen Spielphasen durchläuft.

Scenarien:

1. *Bot durchläuft Phasen des Spiels*
⇒ Einzelne Phasen werden erfolgreich abgeschlossen.
2. *Bot durchläuft Phasen des Spiels nicht*
⇒ Fehlermeldung, wenn Phasen nicht abgeschlossen werden.

Relevante Systemfunktionen: ??

Relevante Use Cases: /UC200/

/TF150/ **Name:** Spielrunden durchlaufen.

Motivation: Testet, ob leer ist und Nachziehstapel dreimal geleert wurde.

Scenarien:

1. *Nachziehstapel nicht leer*
⇒ Keine Aktion des Systems.
2. *Nachziehstapel ist leer, jedoch nicht dreimal geleert*
⇒ Ablegestapel wird gemischt und als Nachziehstapel deklariert.
3. *Nachziehstapel ist leer und dreimal geleert*
⇒ System leitet letzte Spielphasen ein und meldet dies den Spielern. Verweis auf /UC220/ wird angezeigt.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC210/

/TF160/ **Name:** Spiel beenden.

Motivation: Testet, ob ein Spiel ordnungsgemäß beendet wird.

Scenarien:

1. *System hat Funktion "Spielbeenden" aufgerufen*
⇒ Gewinner wird bekanntgegeben, Spieler werden in Lobby geleitet, Datenbankeintrag des Gewinners wird aktualisiert.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC220/

/TF170/ **Name:** Phase 1.

Motivation: Testet, ob Phase 1 korrekt durchlaufen wird.

Scenarien:

1. *Spieler baut Handkarten ab*
⇒ Handkarten erscheinen auf dem Bohnenfeld.
2. *Kein freies Bohnenfeld vorhanden*
⇒ Spieler erhält Meldung des Systems.
3. *Spieler beendet Phase 1.*
⇒ System leitet Phase 2 ein.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC120/

/TF180/ **Name:** Bohnenkarte abernten.

Motivation: Testet, ob Bohnenkarten erfolgreich abgeerntet werden.

Scenarien:

1. *Spieler baut Bohnenkarten ab*
⇒ Spieler erhält Taler im Wert der abgeernteten Bohnenkarten, System legt übrige Karten auf den Ablagestapel.
2. *Bohnenschutzregel verletzt*
⇒ Spieler erhält Meldung des Systems.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC130/

/TF190/ **Name:** Phase 2.

Motivation: Testet, ob Phase 2 korrekt durchlaufen wird.

Scenarien:

1. *System hat Prüfung von /UC210/ erfolgreich abgeschlossen*
⇒ System legt zwei Karten vom Nachziehstapel daneben.
2. *System hat Prüfung von /UC210/ nicht erfolgreich abgeschlossen*
⇒ Verweis auf ??.
3. *Spieler betätigt Handel*
⇒ System leitet Handelphase ein /TF200/.

4. *Spieler handelt nicht oder handel beendet*
⇒ System legt ürbige Karten neben Anbaufeld.
5. *Spieler beendet Phase 2*
⇒ System leitet Phase 3 ein.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC140/

/TF200/ **Name:** Handel.

Motivation: Testet, ob Handel erfolgreich durchlaufen wird.

Szenarien:

1. *Spieler wählt anderen Spieler zum Handeln aus*
⇒ Das System zeigt beiden Spieler ein Handelfenster.
2. *Spieler schließt Handelfenster*
⇒ Das System meldet "Handel abgebrochen".
3. *Spieler beenden Handel*
⇒ Das System legt die neu erworbenen Bohnenkarten neben das Anbaufeld.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC140/

/TF210/ **Name:** Phase 3.

Motivation: Testet, ob Phase 3 erfolgreich durchlaufen wird.

Szenarien:

1. *Spieler wählen die Karten des Anbaustapels und bauen diese an.*
⇒ Das System verschiebt die Karten auf die entsprechenden Anbaufelder.
2. *Spieler besitzen keine freien Felder*
⇒ Das System meldet "Kein freies Anbaufeld vorhanden."
3. *Alle Spieler haben die Bohnenkarten aus Phase 2 angebaut*
⇒ Das System leitet Phase 4 ein.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC160/

/TF220/ **Name:** Phase 4.

Motivation: Testet, ob Phase 4 erfolgreich durchlaufen wird.

Szenarien:

1. *System hat Prüfung von /UC210/ erfolgreich abgeschlossen*
⇒ System weist dem Spieler drei neue Handkarten in entsprechender Reihenfolge zu.

2. *System hat Prüfung von /UC210/ nicht erfolgreich abgeschlossen*
⇒ Verweis auf ??.
3. *System hat drei Handkarten zugewiesen*
⇒ Das System übergibt die Rolle des aktiven Spielers dem nächsten Spieler und beginnt bei Phase 1.

Relevante Systemfunktionen: /LF10/

Relevante Use Cases: /UC170/

Kapitel 6

Warteraum

Hier werden Anforderungen spezifiziert die den sogenannten “Warteraum” darstellen. Hier gehören alle Anforderungen, die “Wünschkriterien” sind, das heißt, sie sind zwar erwünscht, aber werden nur dann in aktuelle Anforderungen übernommen, wenn dafür genügend Zeitbudget vorhanden ist und werden am wahrscheinlichsten in der Zukunft und nicht jetzt implementiert (oder in den kommenden Sprints beim SCRUM-Prozessmodell).

/WR10/ Name: Hintergrundmusik

Beschreibung: Für die Spieler soll ein Auswahl zur Verfügung stehen, was für Hintergrundmusik sie beim Spielen in der Anwendung hören.

Motivation: Höhere Zufriedenheit der Benutzer

Erfüllungskriterium: Spieler können an jedem Zeitpunkt außer Vorraum aus einem Auswahl eine Musik auswählen oder die Musik ausschalten.

/WR20/ Name: Stats der Spieler

Beschreibung: In der Bestenliste soll es möglich sein, eine erweiterte Ansicht mit gewonnenen und verlorenen Spielen des Spielers zu sehen, sowie dessen höchst gewonnenes Spiel.

Motivation: Höhere Zufriedenheit und mehr Ehrgeiz der Benutzer

Erfüllungskriterium: Spieler können auf die Namen in der Bestenliste klicken.

/WR30/ Name: Freundesliste

Beschreibung: Es soll möglich sein, eine Freunde hinzuzufügen und zu sehen, ob diese online sind.

Motivation: Höhere Zufriedenheit der Benutzer und mehr soziale Interaktion

Erfüllungskriterium: Spieler können andere Spieler durch einen Button zur Freundesliste hinzufügen oder bestimmte Spielernamen suchen.

/WR40/ Name: Bots

Beschreibung: Es sollen mehr als zwei Schwierigkeitsstufen für Bots hinzugefügt werden.

Motivation: Höhere Zufriedenheit der Benutzer

Erfüllungskriterium: Spieler können aus mehreren Schwierigkeitsstufen für Bots wählen.

/WR50/ Name: Tutorial

Beschreibung: Beim ersten Login werden alle Menüpunkte durch Popups bzw. Tooltips erläutert.

Motivation: Höhere Zufriedenheit der Benutzer und leichter Einstieg ins Spiel

Erfüllungskriterium: Spieler erhalten beim ersten Login eine schrittweise Einführung in die Menüführung.