

# IT-Projekt

---

Tichu



Autoren:	Thomas Hottinger, Sandro Zimmermann, Pascal Marbach, Fabio Schüpfer
Klasse	WIVZ_2.51
Status:	<input type="checkbox"/> In Bearbeitung <input checked="" type="checkbox"/> Abgeschlossen
Dozent:	Lukas Frey, Prof. Dr. Richards Bradley

# Tichu

---

## Projektdokumentation

Autoren: Thomas Hottinger (TH), Sandro Zimmermann (SZ), Pascal Marbach (PM), Fabio Schüpfer (FS)

Datum: 20. Dezember 2019

Status: ☐ In Bearbeitung ☒ Abgeschlossen

Originaldatei: Anforderungsanalyse\_TeamPlatin.docx

---

Leistungsbezüger: Lukas Frey, Richards Bradley

Leistungserbringer: FHNW Olten

Gesamtvorhaben: **Die Anforderungsanalyse sowie Umsetzung des Spiels Tichu in einer Client-Server Java Anwendung.**

---

Verteiler:

Fabio Schüpfer	<a href="mailto:fabio.schuepfer@students.fhnw.ch">fabio.schuepfer@students.fhnw.ch</a>
Thomas Hottinger	<a href="mailto:thomas.hottinger@students.fhnw.ch">thomas.hottinger@students.fhnw.ch</a>
Sandro Zimmermann	<a href="mailto:sandro.zimmermann@students.fhnw.ch">sandro.zimmermann@students.fhnw.ch</a>
Pascal Marbach	<a href="mailto:pascal.marbach@students.fhnw.ch">pascal.marbach@students.fhnw.ch</a>
Lukas Frey	<a href="mailto:lukas.frey@fhnw.ch">lukas.frey@fhnw.ch</a>
Richards Bradley	<a href="mailto:richards.bradley@fhnw.ch">richards.bradley@fhnw.ch</a>

## Inhaltsverzeichnis

<b>1</b>	<b><u>EINLEITUNG</u></b>	<b>4</b>
1.1	ANLASS	4
<b>2</b>	<b><u>ERMITTELTE ANFORDERUNGEN</u></b>	<b>5</b>
2.1	BASISANFORDERUNGEN	5
2.2	OPTIONALE ANFORDERUNGEN	5
2.3	SONSTIGE ANFORDERUNGEN	5
2.4	USE CASES	6
<b>3</b>	<b><u>MOCKUPS</u></b>	<b>18</b>
3.1	MOCKUP WELCOME SCREEN	18
3.2	MOCKUP WAIT FOR CLIENTS	19
3.3	MOCKUP INGAME	20
3.4	MOCKUP END OF THE GAME	21
<b>4</b>	<b><u>TESTPLAN</u></b>	<b>22</b>
4.1	KOMPONENTENTESTS	22
4.2	INTEGRATIONSTESTS	22
4.3	FUNKTIONSTESTS	23
<b>5</b>	<b><u>EFFEKTIV UMGESETZTE ANFORDERUNGEN</u></b>	<b>29</b>
5.1	BASISANFORDERUNGEN	29
5.2	OPTIONALE ANFORDERUNGEN	29
5.3	SONSTIGE ANFORDERUNGEN	29
<b>6</b>	<b><u>ABBILDUNGSVERZEICHNIS</u></b>	<b>30</b>

# 1 Einleitung

## 1.1 Anlass

Dieses Dokument definiert sämtliche Anforderungen sowie Testfälle für das Spiel Tichu, welches im Rahmen des IT-Projekts entwickelt werden soll.

Ausserdem wird mithilfe dieses Dokuments der Projektrahmen abgegrenzt. Schnittstellen werden beansprucht, in diesem Dokument allerdings nicht weiter spezifiziert.

Ziel dieser Projektarbeit ist es, dass das Team Platin anhand dieses Dokuments das Spiel Tichu entwickeln kann.

## 2 Ermittelte Anforderungen

In diesem Kapitel werden die ermittelten Anforderungen aufgezeigt.

### 2.1 Basisanforderungen

Client und Server starten, Client- und Serververbindung aufbauen, Karten mischen, Karten austeilen, Spieler mit Mah Jong Karte beginnt, Karten ausspielen, Kartenkombination erkennen, Spieler kann passen, Stechen ermitteln, Rundengewinner ermitteln, Gewinnerteam ermitteln

### 2.2 Optionale Anforderungen

Voller Umfang von Mah Jong implementieren, Drache implementieren, Hund implementieren, Phönix implementieren, Bomben implementieren, Kleines Tichu ausrufen, Grosses Tichu ausrufen, Bilingual (Englisch, Deutsch), Datenbankbindung, Teamspieler miteinander Chatten, Lobby einbinden, Neustartoptionen, Karten vor Spielbeginn austauschen,

### 2.3 Sonstige Anforderungen

Spielrichtung nach rechts, Teammitglieder sind sich gegenüber, Regelwerk anzeigen, Kartenhilfe anzeigen, Soundeffekte abspielen

## 2.4 Use Cases

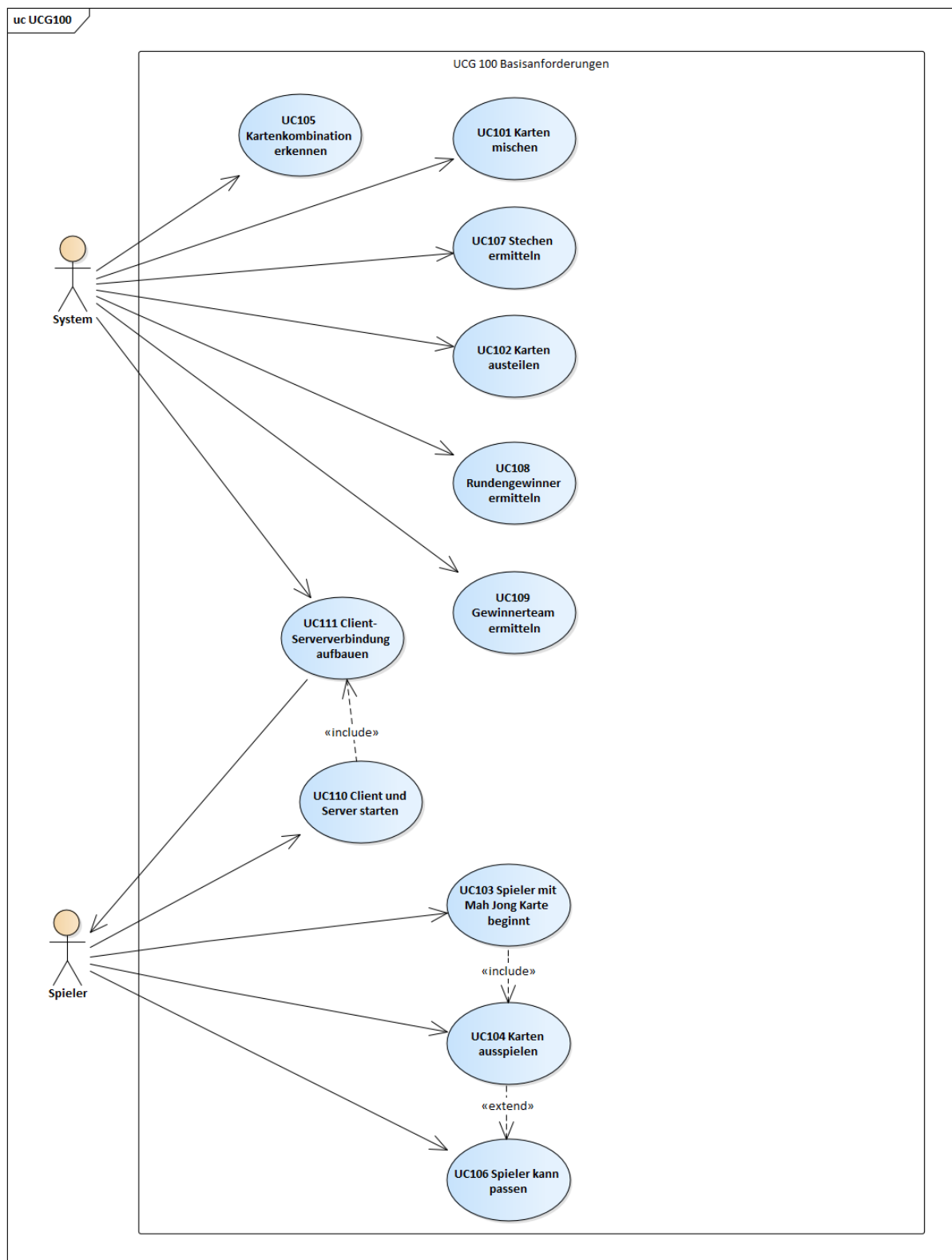


Abbildung 1 Modellerte Basisanforderungen

<b>Use Case ID</b>	<b>101</b>	
<b>Name</b>	Karten mischen	
<b>Beschreibung</b>	Die Karten sollen zu Beginn jeder Runde gemischt werden. Es sind 56 Spielkarten vorhanden.	
<b>Verantwortlicher</b>	Spieler	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Ein Spiel beginnt.	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Programm wurde gestartet.</li> <li>- Spiel wurde gestartet.</li> </ul>	
<b>Eingaben</b>	-	
<b>Ablauf</b>	<b>Standardablauf</b> 1. Spiel startet.	<b>Alternativablauf</b>
	2. Alle Spieler sind am Tisch.	
	3. Karten werden gemischt.	
<b>Ausgaben</b>	-	
<b>Optionale Anforderungen</b>		
<b>Nachbedingungen</b>	Karten sind gemischt.	

<b>Use Case ID</b>	<b>102</b>	
<b>Name</b>	Karten austeilen	
<b>Beschreibung</b>	Jeder Spieler muss zu Beginn einer Runde 14 Karten erhalten.	
<b>Verantwortlicher</b>	System	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Eine neue Runde wird gestartet.	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Karten wurden gemischt (UC101)</li> </ul>	
<b>Eingaben</b>	-	
<b>Ablauf</b>	<b>Standardablauf</b> 1. Jeder Spieler erhält 14 Karten.	<b>Alternativablauf</b>
<b>Ausgaben</b>	<ul style="list-style-type: none"> <li>- Information «Karten ausgeteilt» (Schritt 2)</li> </ul>	
<b>Optionale Anforderungen</b>	<ul style="list-style-type: none"> <li>- Grosses Tichu ausrufen</li> <li>- Kleines Tichu ausrufen</li> <li>- Karten vor Spielbeginn austauschen</li> </ul>	
<b>Nachbedingungen</b>	Karten sind ausgeteilt.	



<b>Use Case ID</b>	<b>103</b>	
<b>Name</b>	Spieler mit Mah Jong Karte beginnt.	
<b>Beschreibung</b>	Der Spieler, der die Mah Jong Karte besitzt, beginnt das Spiel. Er muss die Mah Jong Karte nicht als erstes Auspielen.	
<b>Verantwortlicher</b>	System	
<b>Beteiligte</b>	Spieler	
<b>Auslöser</b>	<ul style="list-style-type: none"> <li>- Karten wurden ausgeteilt.</li> </ul>	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Karten austeilen (UC102).</li> </ul>	
<b>Eingaben</b>	-	
<b>Ablauf</b>	<b>Standardablauf</b> 1. Spieler mit Mah Jong Karte startet mit UC104.	<b>Alternativablauf</b>
<b>Ausgaben</b>	<ul style="list-style-type: none"> <li>- Meldung «Spieler XY beginnt!»</li> </ul>	
<b>Optionale Anforderungen</b>	-	
<b>Nachbedingungen</b>	-	

<b>Use Case ID</b>	<b>104</b>	
<b>Name</b>	Karten ausspielen	
<b>Beschreibung</b>	Die Spieler können Karten anhand der Spielregeln ausspielen.	
<b>Verantwortlicher</b>	Spieler	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Ein Spieler ist am Zug.	
<b>Vorbedingungen</b>	- Karten austeilen (UC102)	
<b>Eingaben</b>	- Karten die gespielt werden sollen.	
<b>Ablauf</b>	<b>Standardablauf</b>	<b>Alternativablauf</b>
	1. Spieler wählt eine oder mehrere Karten aus.	
	2. Spieler bestätigt seine Kartenkombination.	
	3. Kartenkombination ist gültig.	3a1. Kartenkombination entspricht nicht den Spielregeln. 3a2. Meldung, dass andere Karten gewählt werden müssen.
	4. Karten werden in die Mitte gelegt. Nächster Spieler ist an der Reihe.	
<b>Ausgaben</b>	- Meldung «Kartenauswahl nicht gültig» (Schritt 3a2).	
<b>Optionale Anforderungen</b>	-	
<b>Nachbedingungen</b>	Karten wurden ausgespielt.	

<b>Use Case ID</b>	<b>105</b>	
<b>Name</b>	Kartenkombinationen erkennen	
<b>Beschreibung</b>	Das System erkennt die nach Spielregeln gültigen Kartenkombinationen – ohne Spezialkarten und Bomben.	
<b>Verantwortlicher</b>	System	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	<ul style="list-style-type: none"> <li>- Spieler bestätigt seine Kartenkombination.</li> <li>- Wenn gestochen wird.</li> </ul>	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Karten austeilen (UC102)</li> </ul>	
<b>Eingaben</b>	<ul style="list-style-type: none"> <li>- Karten die gespielt werden sollen.</li> <li>- Karten die ausgespielt wurden.</li> </ul>	
<b>Ablauf</b>	<b>Standardablauf</b> 1. Erkennt Einzelkarten, Erkennt Paare, Erkennt aufeinanderfolgende Paare, Erkennt Tripple, Erkennt Full House, Erkennt Reihe von mindestens Länge 5	<b>Alternativablauf</b>
<b>Ausgaben</b>	<ul style="list-style-type: none"> <li>- Meldung «Kartenauswahl nicht gültig» (Schritt 3a2).</li> </ul>	
<b>Optionale Anforderungen</b>	<ul style="list-style-type: none"> <li>- Drache implementieren</li> <li>- Hund implementieren</li> <li>- Phönix implementieren</li> <li>- Bomben implementieren</li> </ul>	
<b>Nachbedingungen</b>	Karten wurden ausgespielt.	

<b>Use Case ID</b>	<b>106</b>	
<b>Name</b>	Spieler kann passen	
<b>Beschreibung</b>	Der Spieler kann passen und somit einen Spielzug überspringen.	
<b>Verantwortlicher</b>	Spieler	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Spieler klickt den Passen Button an.	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Karten mischen (UC101)</li> <li>- Karten austeilen (UC102)</li> </ul>	
<b>Eingaben</b>	<ul style="list-style-type: none"> <li>- Button Passen wird gedrückt.</li> </ul>	
<b>Ablauf</b>	<b>Standardablauf</b>	<b>Alternativablauf</b>
	1. Der Spieler klickt den Button passen an.	
	2. Der Zug des Spielers ist beendet, ohne eine Karte zu spielen.	
	3. Der nächste Spieler ist am Zug.	
<b>Ausgaben</b>	<ul style="list-style-type: none"> <li>- Meldung «Spieler XY ist am Zug» (Schritt 3)</li> </ul>	
<b>Optionale Anforderungen</b>	-	
<b>Nachbedingungen</b>	-	

<b>Use Case ID</b>	<b>107</b>	
<b>Name</b>	Stechen ermitteln	
<b>Beschreibung</b>	Sobald alle Spieler passen müssen, sticht der Spieler, welcher als letztes die höchste Karte abgelegt hat und erhält die Karten in der Mitte.	
<b>Verantwortlicher</b>	System	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Alle Spieler mussten passen, bis auf den Spieler, der am Zug ist.	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Karten austeilen (UC102)</li> </ul>	
<b>Eingaben</b>	<ul style="list-style-type: none"> <li>- ID des letzten Spielers, welcher nicht gepasst hat.</li> </ul>	
<b>Ablauf</b>	<b>Standardablauf</b> 1. ID des gewinnenden Spielers erkennen.	<b>Alternativablauf</b>
	2. Karten in der Mitte dem Spieler der als letztes Karten gelegt hat zuweisen.	
<b>Ausgaben</b>	<ul style="list-style-type: none"> <li>- Meldung «Spieler XY sticht!» (Schritt 2).</li> </ul>	
<b>Optionale Anforderungen</b>	<ul style="list-style-type: none"> <li>- Drache implementieren</li> <li>- Hund implementieren</li> <li>- Phönix implementieren</li> <li>- Bomben implementieren</li> </ul>	
<b>Nachbedingungen</b>	Karten wurden ausgespielt.	

<b>Use Case ID</b>	<b>108</b>	
<b>Name</b>	Rundengewinner ermitteln	
<b>Beschreibung</b>	Sobald nur noch 1 Spieler Karten auf der Hand hat, muss der Rundengewinner ermittelt werden. Wenn beide Spieler eines Teams als 1ter und 2ter die Runde abschliessen, erhalten Sie ohne Gewinnberechnung 200 Punkte.	
<b>Verantwortlicher</b>	System	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Nur noch 1 Spieler hat Karten auf der Hand. Beide Spieler eines Teams haben nacheinander keine Karten mehr auf der Hand.	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Karten mischen (UC101)</li> <li>- Karten austeilen (UC102)</li> <li>- Karten ausspielen (UC104)</li> <li>- Stechen ermitteln (UC107)</li> </ul>	
<b>Eingaben</b>	<ul style="list-style-type: none"> <li>- Gestochene Karten der Spieler</li> <li>- Handkarten des Verlierers</li> </ul>	
<b>Ablauf</b>	<b>Standardablauf</b> 1. Die Handkarten des Verlierers gehen an das Gegnerteam.	<b>Alternativablauf</b> 1a1. Beide Spieler eines Teams schliessen die Runde als 1er und 2er ab. 1a2. Das Team erhält ohne Punkteberechnung 200 Punkte auf ihr Konto gutgeschrieben.
	2. Die Stiche des Verlierers gehen an den Rundengewinner	
	3. Punkte der Teams werden berechnet und auf ihr Punktekonto gutgeschrieben.	
	4. Runde abgeschlossen.	
<b>Ausgaben</b>	<ul style="list-style-type: none"> <li>- Meldung «Team XY hat einen Doppelsieg. 200 Punkte» (Schritt 1a2).</li> <li>- Meldung «Anzahl Punkte Team X: Anzahl Punkte Team Y:» (Schritt 3).</li> <li>- Meldung «Runde abgeschlossen» (Schritt 4)</li> </ul>	
<b>Optionale Anforderungen</b>	<ul style="list-style-type: none"> <li>- Drache implementieren</li> <li>- Phönix implementieren</li> </ul>	

<b>Nachbedingungen</b>	Runde abgeschlossen	
<b>Use Case ID</b>	109	
<b>Name</b>	Gewinnerteam ermitteln	
<b>Beschreibung</b>	Das Spiel ist gewonnen, sobald ein Team in n Runden mehr als 1000 Punkte erreicht hat. Erreichen beide Teams in derselben Runde 1000 Punkte, gewinnt das Team mit mehr Punkten.	
<b>Verantwortlicher</b>	System	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Ein Team erreicht nach Rundenende mindestens 1000 Punkte.	
<b>Vorbedingungen</b>	- Rundengewinner ermitteln (UC108)	
<b>Eingaben</b>	- Punktekonto der Teams	
<b>Ablauf</b>	<b>Standardablauf</b>	<b>Alternativablauf</b>
	1. Ein Team erreicht nach Rundenende 1000 Punkte.	1a1. Beide Teams erreichen nach Rundenende 1000 Punkte. 1a2. Das Team mit der höheren Punktezahl gewinnt.
	2. Das Team gewinnt.	
	3. Das Spiel ist beendet.	
<b>Ausgaben</b>	- Meldung «Team XY gewinnt» (Schritt 3)	
<b>Optionale Anforderungen</b>	- Neustartoption	
<b>Nachbedingungen</b>	Spiel beendet.	

<b>Use Case ID</b>	<b>110</b>	
<b>Name</b>	Client und Server starten	
<b>Beschreibung</b>	Der Benutzer startet den Client und gibt dabei an ob er den Server hosten möchte.	
<b>Verantwortlicher</b>	Spieler	
<b>Beteiligte</b>	-	
<b>Auslöser</b>	Ein Benutzer möchte eine Runde Tichu spielen.	
<b>Vorbedingungen</b>	-	
<b>Eingaben</b>	- Button zum Start des Servers	
<b>Ablauf</b>	<b>Standardablauf</b>	<b>Alternativablauf</b>
	1. Fenster öffnet sich, der Benutzer kann angeben, dass er den Server hosten möchte.	1a1. Benutzer will den Server nicht starten. 1a2. UC111 wird gestartet.
	2. Server startet auf Port 10456.	
	3. UC111 wird gestartet.	
	4. Client wird gestartet.	
<b>Ausgaben</b>	-	
<b>Optionale Anforderungen</b>	<ul style="list-style-type: none"> <li>- Der Benutzer muss sich am Server Anmelden.</li> <li>- Der Benutzer kann sich am Server registrieren.</li> <li>- Bilingual (Englisch, Deutsch)</li> </ul>	
<b>Nachbedingungen</b>	<ul style="list-style-type: none"> <li>- Client und Server sind gestartet.</li> <li>- Client ist gestartet.</li> </ul>	



<b>Use Case ID</b>	<b>111</b>	
<b>Name</b>	Client – Serververbindung aufbauen	
<b>Beschreibung</b>	Der Benutzer hat beim Programmstart die Möglichkeit die IP-Adresse des Servers einzustellen, worauf hin eine Netzwerkverbindung aufgebaut wird.	
<b>Verantwortlicher</b>	System	
<b>Beteiligte</b>	Spieler	
<b>Auslöser</b>	<ul style="list-style-type: none"> <li>- Client und Server starten (UC110)</li> </ul>	
<b>Vorbedingungen</b>	<ul style="list-style-type: none"> <li>- Client und Server starten (UC110)</li> </ul>	
<b>Eingaben</b>	<ul style="list-style-type: none"> <li>- IP-Adresse des Servers</li> </ul>	
<b>Ablauf</b>	<b>Standardablauf</b>	<b>Alternativablauf</b>
	1. Fenster für Eingabe der IP öffnet sich.	
	2. Benutzer gibt IP-Adresse des Servers ein.	2a1. Benutzer gibt falsche IP Adresse ein. 2a2. Meldung erscheint, dass Verbindungsaufbau nicht möglich ist.
	3. Verbindung wird aufgebaut.	
<b>Ausgaben</b>	<ul style="list-style-type: none"> <li>- Meldung «Verbindungsaufbau nicht möglich» (Schritt 2a1)</li> </ul>	
<b>Optionale Anforderungen</b>	<ul style="list-style-type: none"> <li>- Nach dem Verbindungsaufbau wird dem Spieler eine Lobby angezeigt</li> <li>- Der Benutzer muss sich am Server Anmelden.</li> <li>- Der Benutzer kann sich am Server registrieren.</li> <li>- Datenbankanbindung</li> <li>- Chatfunktion der Spieler</li> </ul>	
<b>Nachbedingungen</b>	Client und Serververbindung hergestellt.	

## 3 Mockups

### 3.1 Mockup Welcome Screen

Beim Welcome Screen ist die CheckBox «Ich bin Server», standardmässig ausgewählt. Somit ist die Local-Host IP direkt auf 127.0.0.1 gesetzt. Falls man Client ist, muss man die IP-Adresse des Servers selbst eintragen. Die Sprachausgabe ist bilingual und es kann zwischen Deutsch und Englisch gewählt werden. Standardmässig ist die Sprache auf Deutsch gesetzt.



Abbildung 2 Mockup Welcome Screen - Ich bin Server

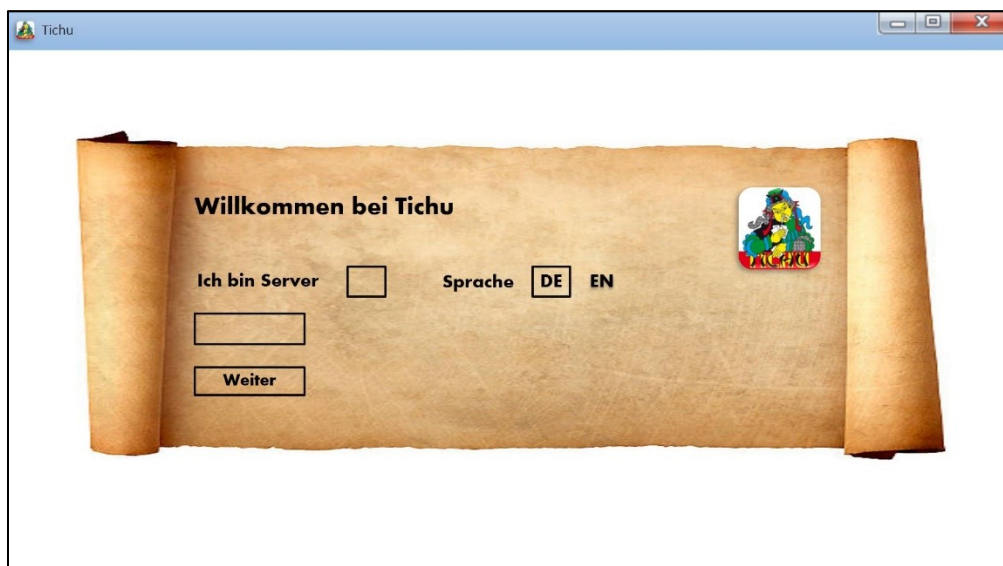


Abbildung 3 Mockup Welcome Screen - Ich bin Client

### 3.2 Mockup Wait for Clients

Sobald man sich dafür entschieden hat, Server oder Client zu sein, muss man warten, bis der Tisch voll ist. Sobald dies geschehen ist, kommt man zum Ingame



Abbildung 4 Mockup Wait for Clients Ansicht

### 3.3 Mockup Ingame

In der Ingame-Ansicht werden jeweils die 2 Gegner, der Mitspieler und die eigenen Handkarten angezeigt. Die Nummer auf der Karte vor den Spielern am Tisch, zeigt an wie viele Karten diese noch auf der Hand haben. Oben rechts wird der aktuelle Punktestand der Teams angezeigt – das Team das mehr Punkte hat, hat den gelben Stern vor dem Teamnamen.

Der Spieler, der an der Reihe ist, erhält die Reisschüssel und sogleich beginnt der 30-Sekunden Countdown. Mit dem Button ausspielen werden die ausgewählten Karten ausgespielt und mit dem Button Passen wird die Runde ausgesetzt. Sobald ein Spieler die Möglichkeit hat eine Bombe zu spielen, wird der Button Bomben freigeschaltet (wechselt von der Farbe Grau auf Rot). Danach kann der Spieler die gewünschte Bomben-Kombination auswählen und ausspielen.

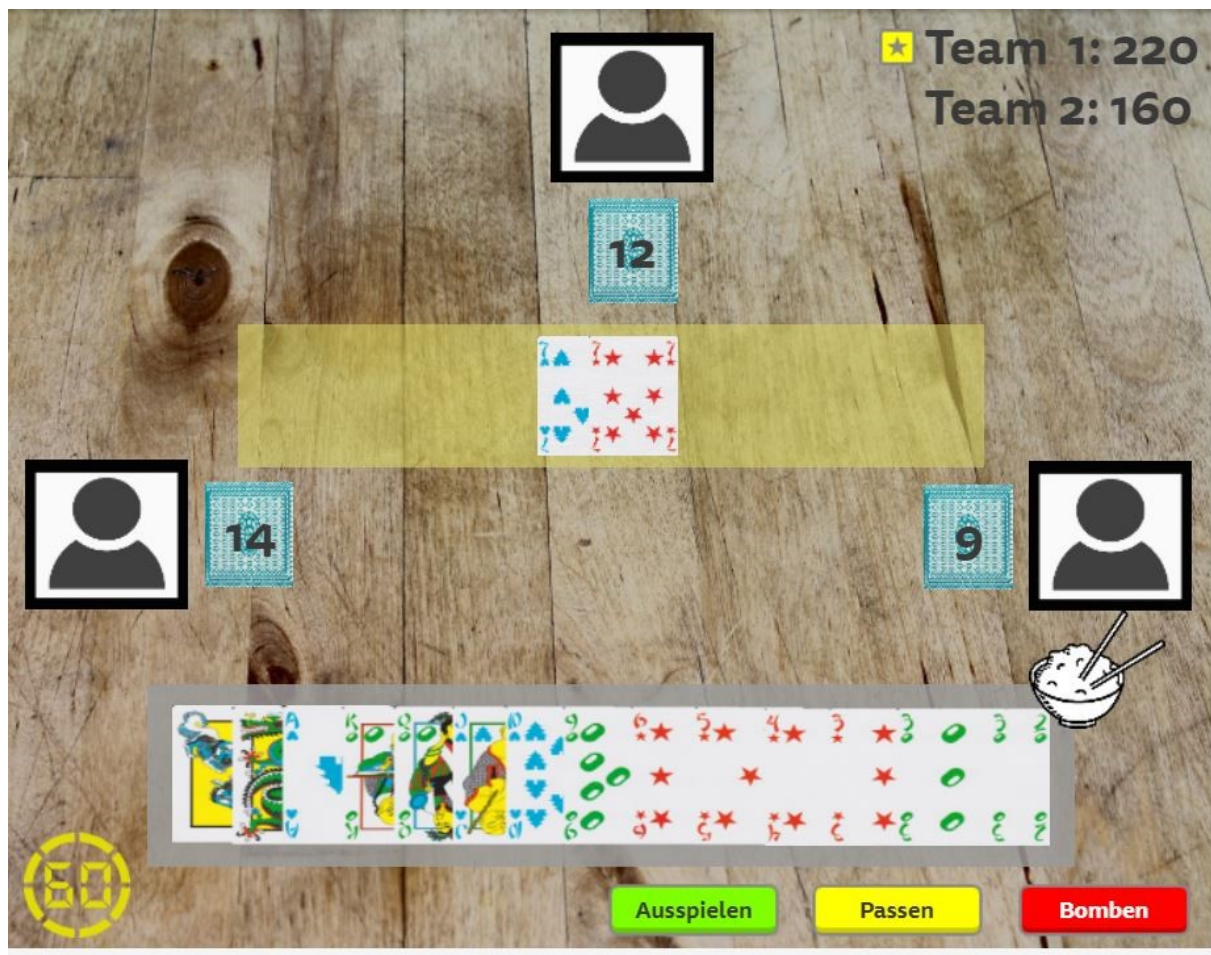


Abbildung 4 Mockup Ingame Ansicht



### 3.4 Mockup end of the game

Am Ende des Spiels wird der Punktestand angezeigt und es wird nochmals erwähnt, ob man gewonnen oder verloren hat. Danach kann ausgewählt werden, ob man nochmals spielen möchte oder nicht. Falls ja, kommt man zurück zum Welcome Screen, falls nein, wird das Fenster geschlossen und das Spiel wird beendet.

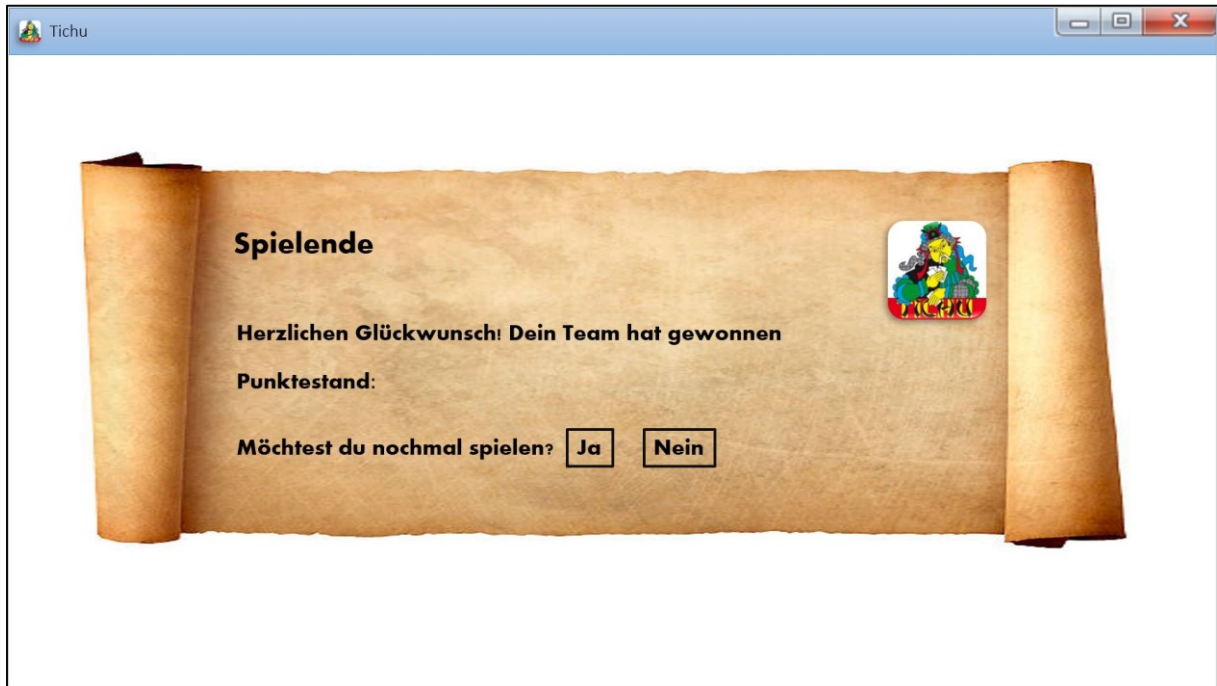


Abbildung 6 Mockup End of the Game Ansicht

## 4 Testplan

Mithilfe von Testplänen wird beurteilt, ob die Umgesetzten Anforderungen der gewünschten Qualität entsprechen. Alle Eingeegebenen Daten müssen getestet und überprüft werden.

Die Testpläne werden in die drei Rubriken unterteilt: Komponententests, Integrationstests und Funktionstests.

### 4.1 Komponententests

Die Komponententests können in dieser Projektphase noch nicht definiert werden, da die Komponenten erst in der nächsten Projektphase definiert werden.

### 4.2 Integrationstests

Die Integrationstests werden laufend während des Projekts durchgeführt. Das heisst, der Entwickler testet nach einem «grösseren» Entwicklungsschritt ob sein neu geschriebener Code mit dem Gesamtcode funktioniert.

## 4.3 Funktionstests

ID	TC100 Programm startet
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Das Programm wird ohne Fehlermeldung gestartet.
Tatsächliches Ergebnis	Das Programm wird ohne Fehlermeldung gestartet.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC101 Server startet
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Der Server startet bei dem Spieler, der das Häkchen gesetzt hat.
Tatsächliches Ergebnis	Der Server startet fehlerfrei, jedoch unabhängig vom Client. (Aufgrund von Performancegründen wurde davon abgesehen den Server aus dem Client heraus zu starten.)
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC102 Client startet
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Der Client startet und die Verbindung zum Server wird aufgebaut.
Tatsächliches Ergebnis	Der Client startet und die Verbindung zum Server wird aufgebaut.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC103 Karten werden gemischt
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Die Karten werden gemischt.
Tatsächliches Ergebnis	Die Karten werden gemischt
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen



ID	TC104 Karten austeilen
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Jeder Spieler erhält 14 Karten.
Tatsächliches Ergebnis	Jeder Spieler erhält 14 Karten. Nach den ersten 8 Karten wird für 30 Sekunden gewartet, damit die Spieler ein grosses Tichu ansagen können.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC105 Spieler mit Mah Jong Karte beginnt
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Nur der Spieler mit der Mah Jong Karte kann das Spiel beginnen.
Tatsächliches Ergebnis	Nur der Spieler mit der Mah Jong Karte kann das Spiel beginnen.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC106 Karten ausspielen
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Die Spieler können die Karten anhand der Spielregeln ausspielen.
Tatsächliches Ergebnis	Die Spieler können die Karten anhand der Spielregeln ausspielen.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC107 Kartenkombinationen erkennen
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Alle Kartenkombinationen werden vom Server korrekt erkannt.
Tatsächliches Ergebnis	Alle Kartenkombinationen werden vom Server korrekt erkannt.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC108 Spieler kann passen
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Der nächste Spieler ist am Zug wenn ein Spieler passt.
Tatsächliches Ergebnis	Der nächste Spieler ist am Zug wenn ein Spieler passt.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC109 Stechen ermitteln
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Der Spieler der die letzte, höchste Karte gespielt hat erhält die Karten in der Mitte.
Tatsächliches Ergebnis	Der Spieler der die letzte, höchste Karte gespielt hat erhält die Karten in der Mitte.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	
Ergebnis neue Prüfung	
Status	Abgeschlossen

ID	TC110 Rundengewinner ermitteln
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Wenn nur noch ein Spieler Karten hat, wird der Rundengewinner gemäss den Regeln errechnet, und die Punkte dem Gewinnerteam gutgeschrieben.
Tatsächliches Ergebnis	Wenn nur noch ein Spieler Karten hat, wird der Rundengewinner gemäss den Regeln errechnet, und die Punkte dem Gewinnerteam gutgeschrieben.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	-
Ergebnis neue Prüfung	-
Status	Abgeschlossen

ID	TC111 Gewinnerteam ermitteln
Getestet von	Gesamte Gruppe
Datum Prüfung	17.12.2019
Erwartetes Ergebnis	Wenn ein Team mehr als 1000 Punkte erreicht, ist das Spiel beendet und das Team gewinnt. Oder wenn beide Teams mehr als 1000 Punkte erreichen, gewinnt das Team mit mehr Punkten.
Tatsächliches Ergebnis	Wenn ein Team mehr als 1000 Punkte erreicht, ist das Spiel beendet und das Team gewinnt. Oder wenn beide Teams mehr als 1000 Punkte erreichen, gewinnt das Team mit mehr Punkten.
Ergebnis	Erfolgreich
Massnahmen Problembehandlung	
Ergebnis neue Prüfung	
Status	Abgeschlossen

## 5 Effektiv umgesetzte Anforderungen

### 5.1 Basisanforderungen

Es wurden alle Basisanforderungen erfolgreich umgesetzt.

### 5.2 Optionale Anforderungen

Die folgenden optionalen Anforderungen konnten ausserdem umgesetzt werden:

- Voller Umfang von Mah Jong implementieren
- Drache implementieren
- Hund implementieren
- Phönix implementieren
- Bomben implementieren
- Kleines Tichu ausrufen
- Grosses Tichu ausrufen
- Bilingual (Englisch, Deutsch)

### 5.3 Sonstige Anforderungen

Die folgenden sonstigen Anforderungen wurden umgesetzt:

- Spielrichtung nach rechts
- Teammitglieder sind sich gegenüber
- Soundeffekte abspielen

## 6 Abbildungsverzeichnis

Abbildung 1 Modellerte Basisanforderungen .....	6
Abbildung 2 Mockup Welcome Screen - Ich bin Server.....	18
Abbildung 3 Mockup Welcome Screen - Ich bin Client .....	18
Abbildung 4 Mockup Ingame Ansicht .....	20