

# IT-Projekt

---

## Tichu – System Specifications



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

# IT-Projekt

---

## Projektdokumentation

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

Datum: 20. Dezember 2019

Status: ☐ In Bearbeitung ☒ Abgeschlossen

Originaldatei: SystemSpecifications\_TeamPlatin.docx

---

## Inhaltsverzeichnis

<b>1</b>	<b><u>Einleitung .....</u></b>	<b><u>4</u></b>
<b>2</b>	<b><u>Systemarchitektur.....</u></b>	<b><u>5</u></b>
2.1	Aufbau .....	5
<b>3</b>	<b><u>Klassendiagramme .....</u></b>	<b><u>6</u></b>
<b>4</b>	<b><u>Netzwerkkommunikation.....</u></b>	<b><u>8</u></b>
4.1	Kommunikation für einen Spielzug .....	8
4.2	Kommunikation für das Spielen einer Bombe .....	9
4.3	Kommunikation für die Funktion Passen .....	11
<b>5</b>	<b><u>Organisatorisches .....</u></b>	<b><u>12</u></b>
5.1	Gruppenrichtlinien .....	12
5.2	Aufteilung.....	12
<b>6</b>	<b><u>Änderungen während des Projekts .....</u></b>	<b><u>14</u></b>
6.1	Klassendiagramm Ist-Zustand.....	15
6.2	Anpassungen an der Aufteilung .....	17
6.3	Diverse Änderungen .....	17
<b>7</b>	<b><u>Abbildungsverzeichnis.....</u></b>	<b><u>18</u></b>

## 1 Einleitung

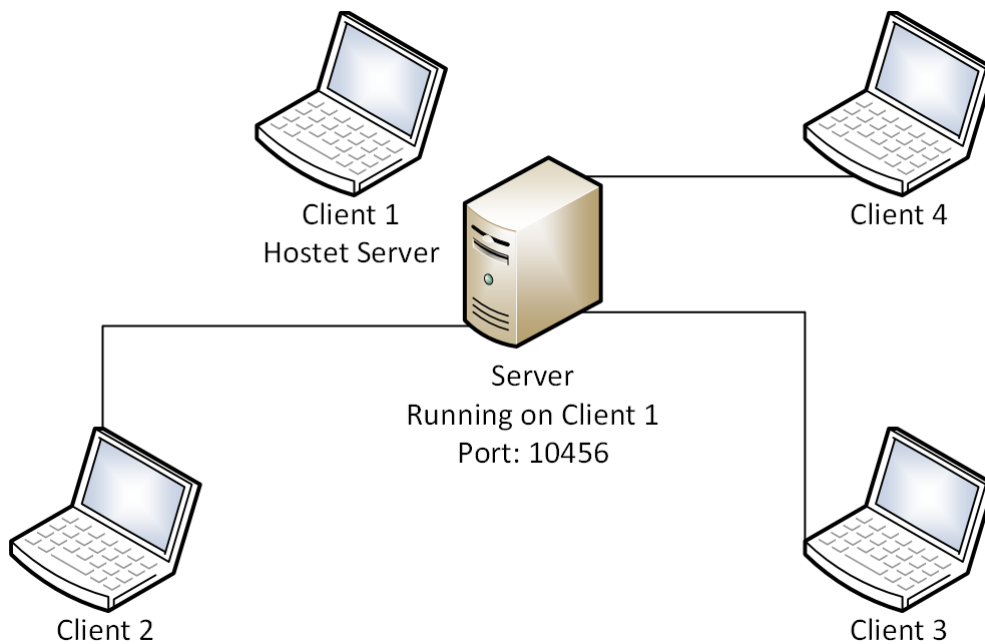
In diesem Dokument werden die Systemspezifikationen für das mit Java entwickelte Spiel Tichu definiert. Dazu werden UML-Klassendiagramme sowie diverse Visio Diagramme zur Veranschaulichung benutzt. Das Dokument wurde nach dem Motto «Konzept und Übersicht vor Vollständigkeit» erstellt.

Änderungen welche während des Projekts vorgenommen wurden, sind im Kapitel 6 dargestellt.

## 2 Systemarchitektur

Das Spiel soll als Client- Server Anwendung implementiert werden. Ein Client übernimmt dabei jeweils die Rolle als Server. Das heisst, er ist Spieler und hostet gleichzeitig die Serveranwendung.

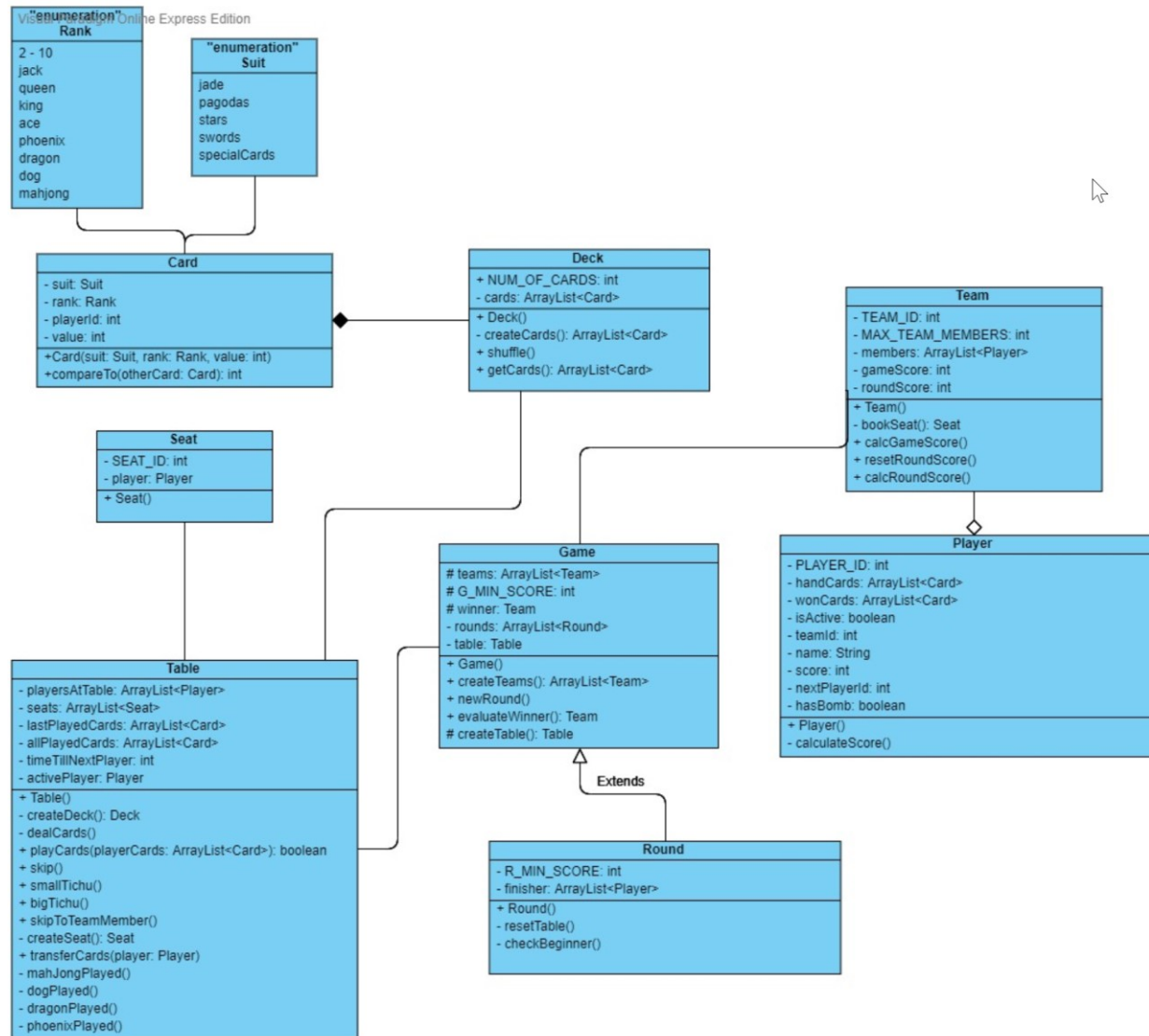
### 2.1 Aufbau

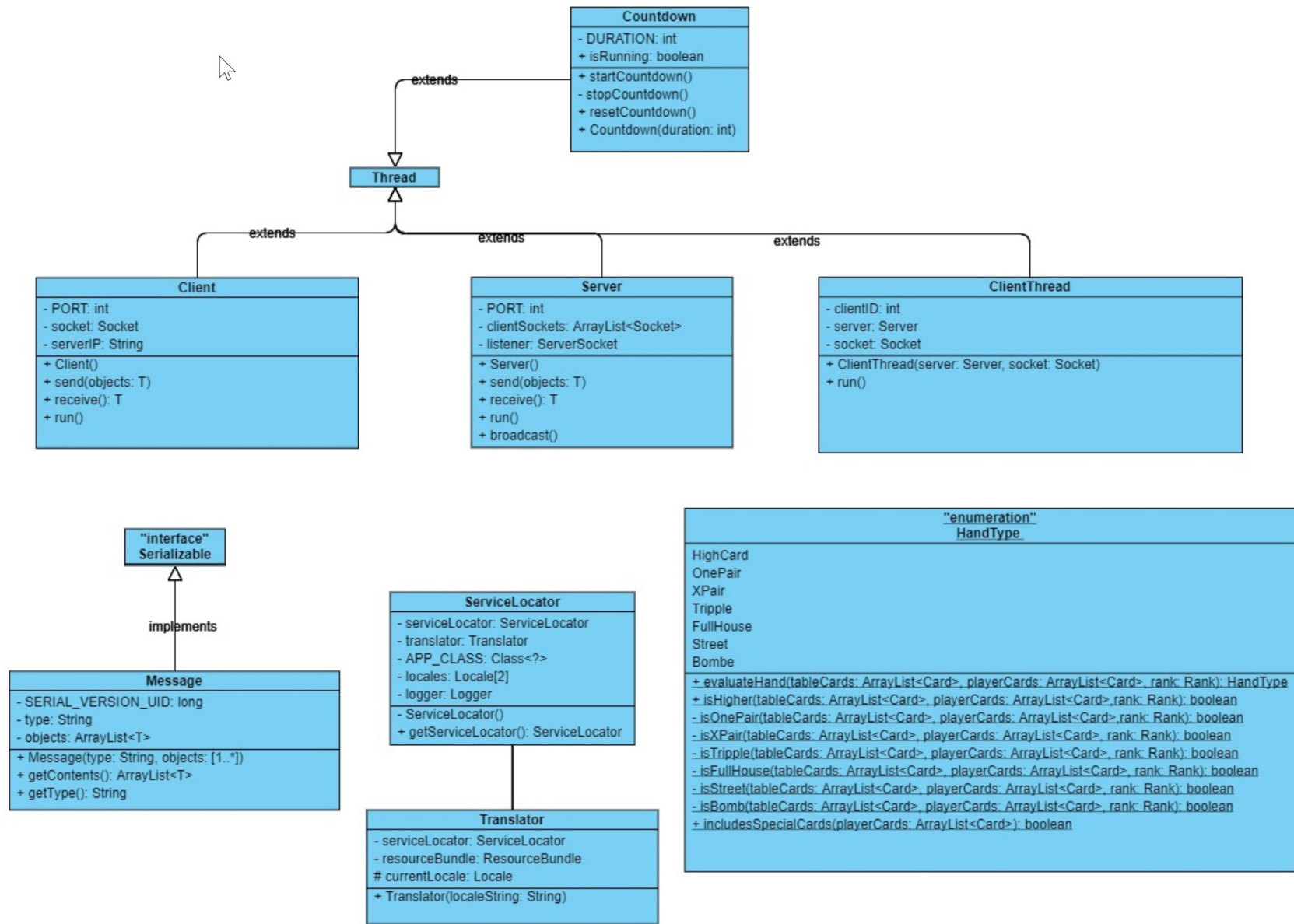


*Abbildung 1 Client Server aufbau*

Der Server läuft auf dem Port 10456 auf einem der Clients. Der Server wird gestartet, indem der Benutzer beim starten des Programms die Checkbox «Ich bin der Server» anwählt.

### 3 Klassendiagramme





## 4 Netzwerkkommunikation

Die Netzwerkkommunikation wird anhand mehrerer Visiodiagramme dargestellt. Es werden nur die wichtigsten Kommunikationsvorgänge dargestellt.

Es darf immer nur der Spieler senden, bei welchem das Attribut isActive auf true gesetzt ist.

### 4.1 Kommunikation für einen Spielzug

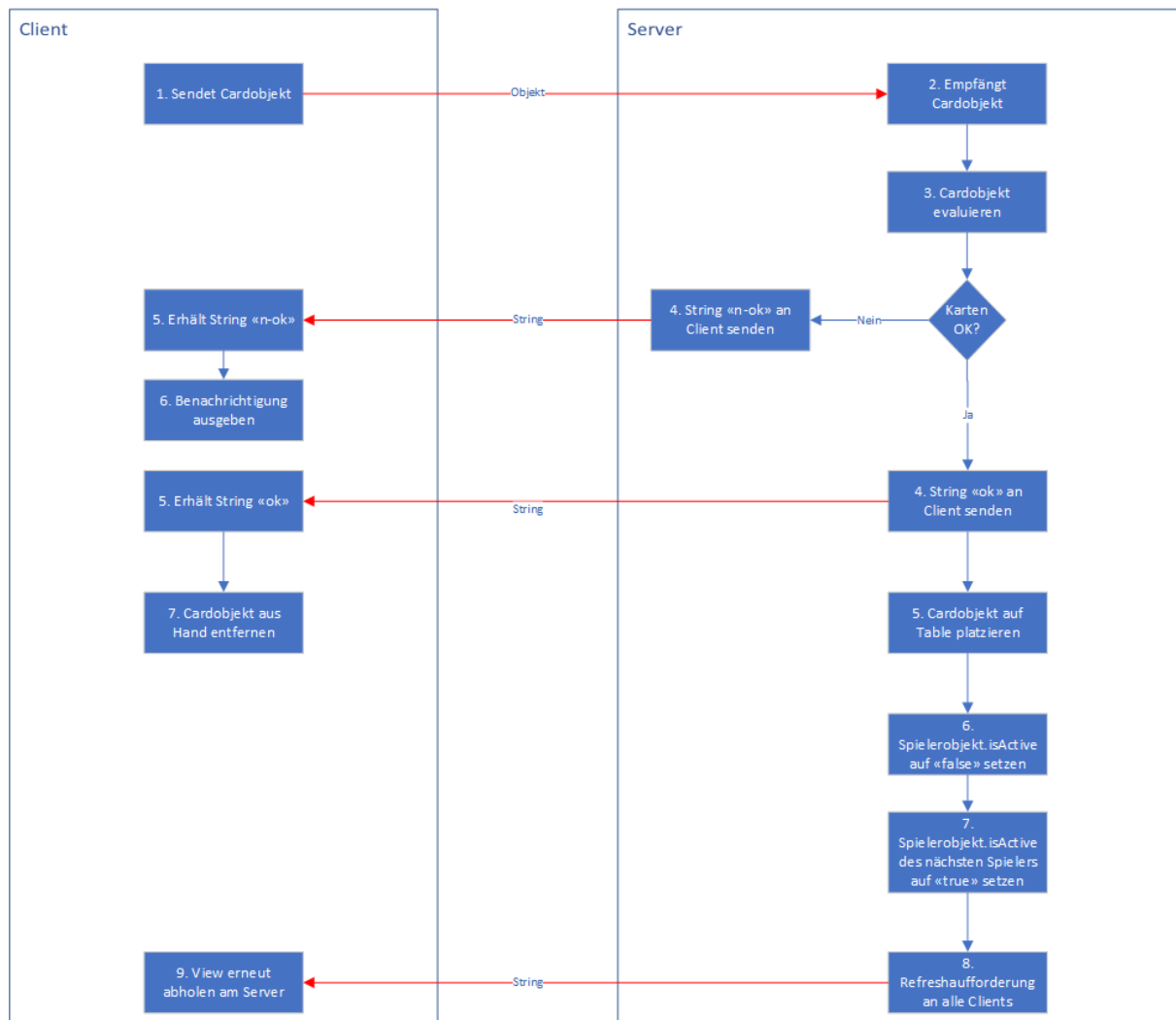


Abbildung 2 Kommunikation Spielzug

Der Client sendet das Cardobjekt welches die Karte, die der Spieler spielen möchte zum Server. Der Server empfängt das Cardobjekt und überprüft, ob diese Karte gemäss den Spielregeln gespielt werden darf.

Falls ja, wird das Objekt auf dem Spieltisch platziert und der Client erhält die Statusmeldung «ok» woraufhin er das Cardobjekt aus der Hand des Spielers entfernt.

Falls nein, erhält der Client die Statusmeldung «n-ok» worauf hin er eine Benachrichtigung an den Spieler ausgibt.



## 4.2 Kommunikation für das Spielen einer Bombe

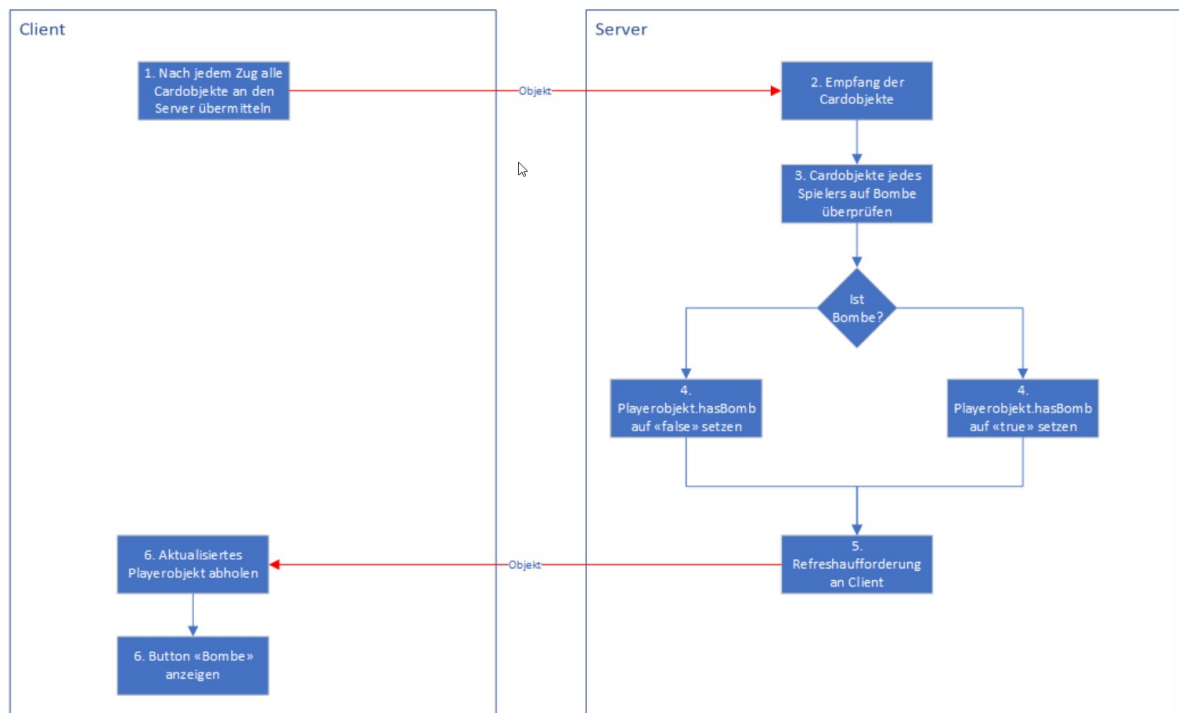


Abbildung 3 Kommunikation Bombe Teil 1

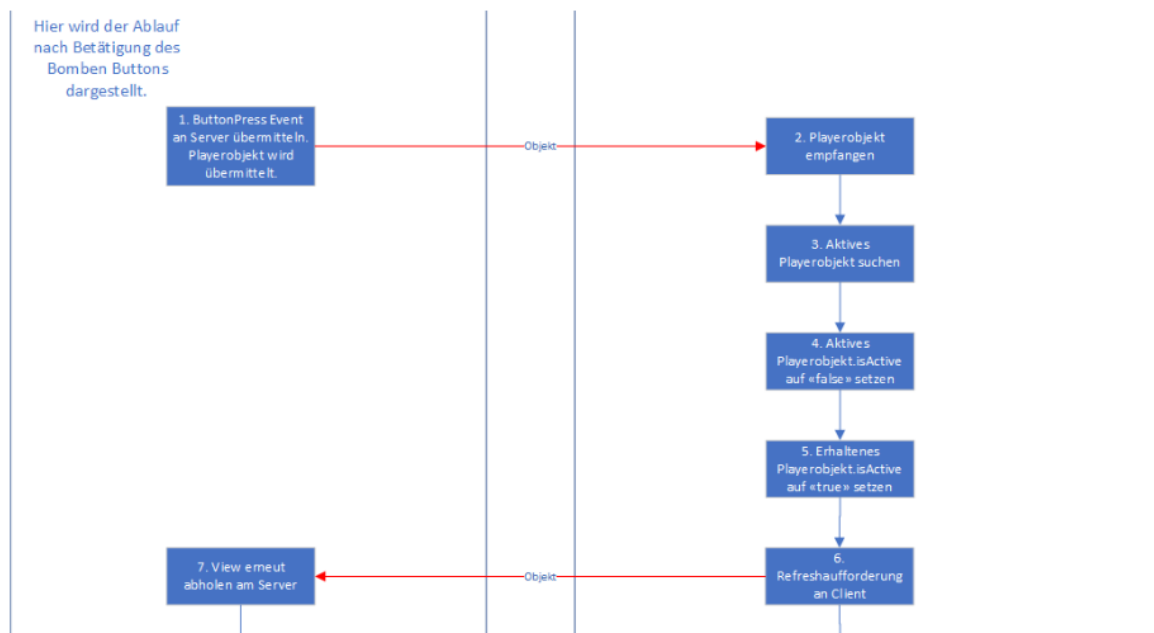


Abbildung 4 Kommunikation Bombe Teil 2

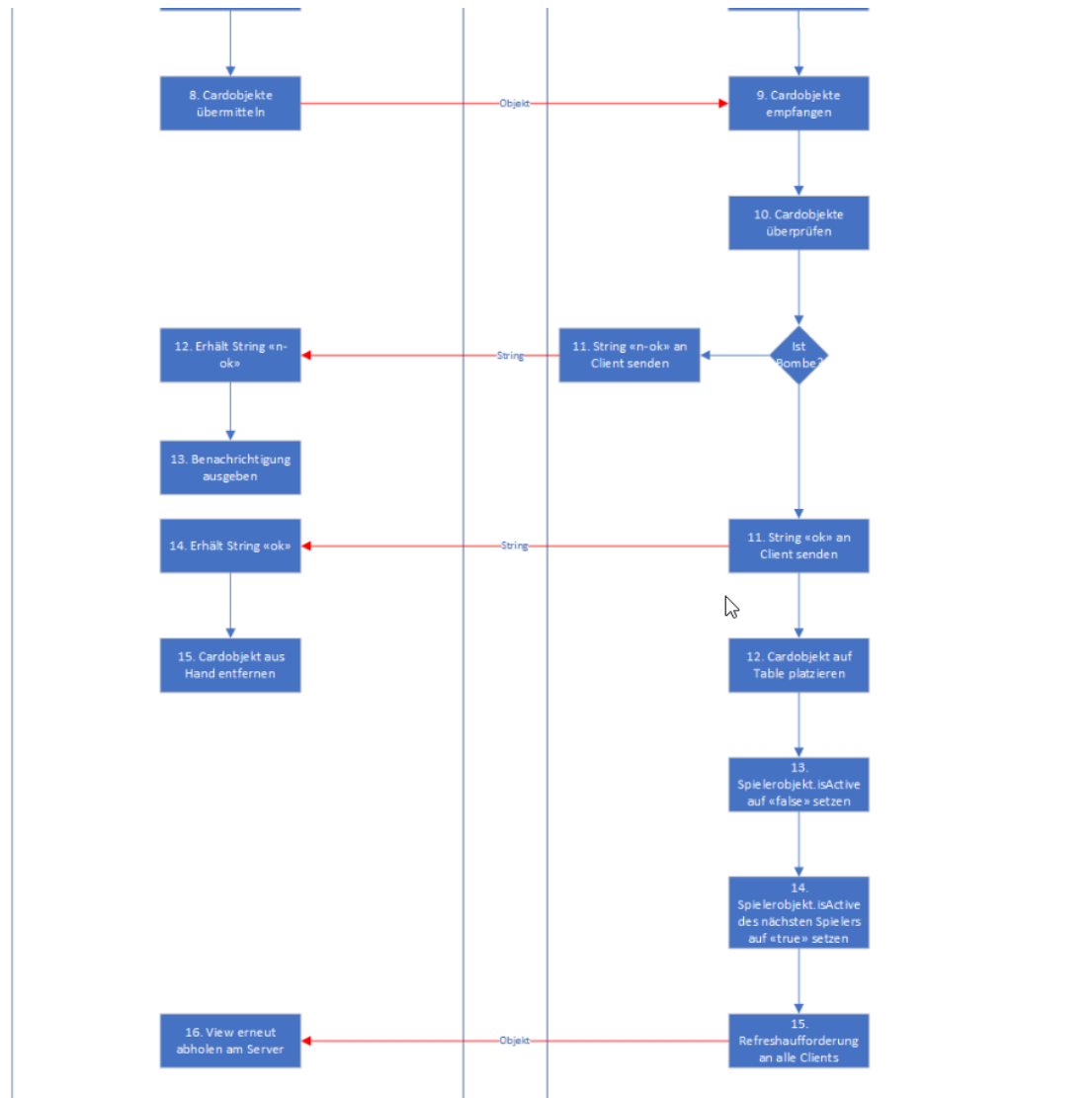


Abbildung 5 Kommunikation Bombe Teil 3

Der Client sendet alle Karten des Spielers nach jedem Zug an den Server. Der Server überprüft die «Cardobjekte» jedes Spielers auf eine Bombe. Falls der Spieler eine Bombe hat, wird das im Playerobjekt das Attribut «hasBomb» auf «true» gesetzt. Daraufhin wird dem Spieler ein Button «Bombe» angezeigt, mit welchem er die Bombe werfen kann.

Klickt der Spieler den Button «Bombe» an, wird das Spiel direkt zu ihm transferiert und der vorherige Spieler in seinem Spielzug unterbrochen. Er hat nun die Möglichkeit die Bombe auszuspielen. (Ablauf wie normaler Spielzug). Es wird überprüft, dass er nur die Bombe ausspielen kann und nicht eine andere Kombination.

## 4.3 Kommunikation für die Funktion Passen

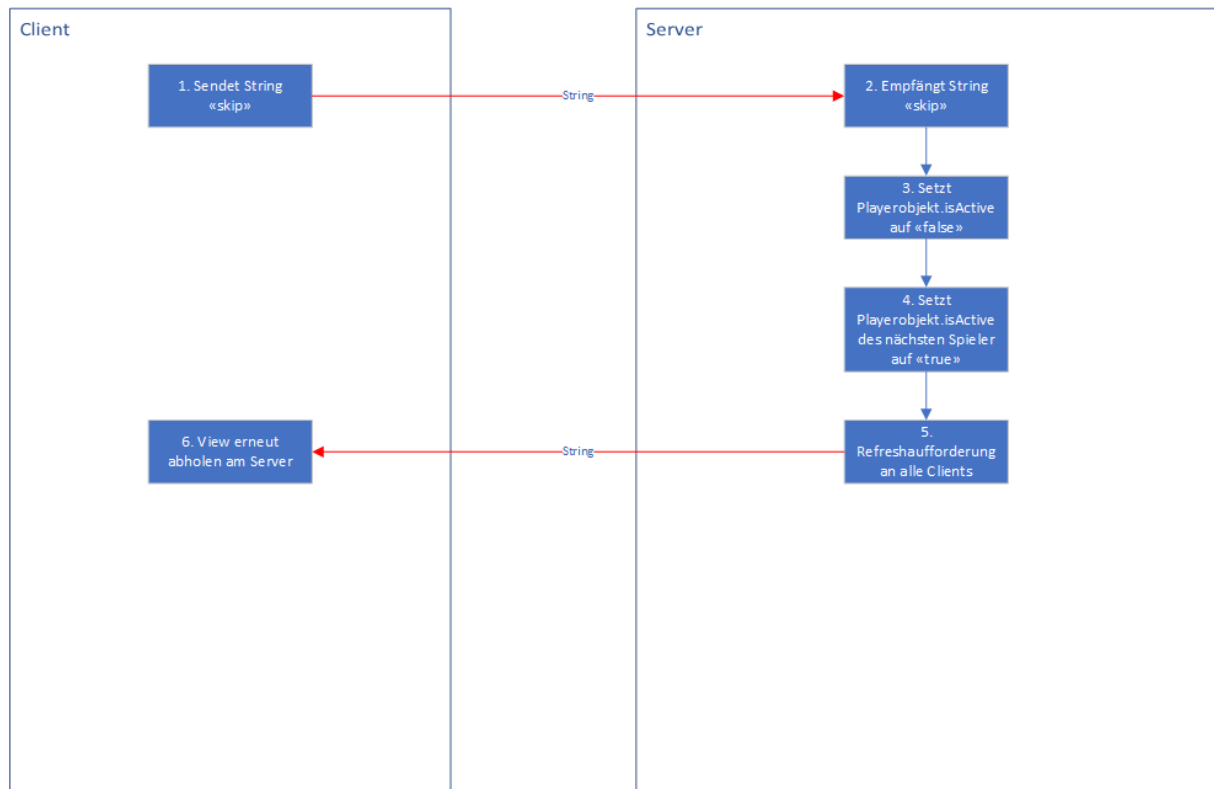


Abbildung 6 Kommunikation Passen

Der Client sendet einen String mit «skip» an den Server. Der Server empfängt den String und setzt bei dem Client, der die Nachricht gesendet hat, den Status von «.isActive» auf «false». Danach wird der Status von «.isActive», vom Client der als nächstes am Zug ist auf «true» gesetzt. Ist beides erledigt sendet der Server eine Refreshaufforderung an alle Clients im Netzwerk. Diese löst bei den Clients das erneute abholen der View vom Server aus, damit wird bei allen Clients das aktuelle Spielgeschehen angezeigt.

## 5 Organisatorisches

### 5.1 Gruppenrichtlinien

1. Einführen von neuen Variablen wird im Team besprochen.
2. Einführen von neuen Methoden wird im Team besprochen.
3. Einführen von neuen Klassen wird im Team besprochen.
4. Jede Methode muss kommentiert sein.
5. Jeder Kommentar enthält: Ersteller, Funktion der Methode.
6. Nach jedem grösseren Entwicklungsschritt testet jeder, ob sein Code zum bestehenden Code passt.
7. Git wird nach jedem Entwicklungsschritt aktualisiert. (Backup!)
8. Kein Code aus Internet / alten Programmen kopieren.
9. Code wird strukturiert dargestellt: (lieber zu viel platz brauchen als zu wenig)
  - a. Klasse
    - i. Methode
      1. Logik
0. Testen ohne System.out.println -> nur mit Debugger. Falls doch mit System.out nach **jedem** Test sofort wieder entfernen.

### 5.2 Aufteilung

Das Grundgerüst des GUI wird in Gruppenarbeit entwickelt. Für die Spezifikationen und detaillierten Darstellungen ist jeder in der Gruppe selbst verantwortlich.

Der Controller wird von jedem individuell gepflegt.

Die Konstruktoren der Klassen werden von allen Entwicklern nach Bedarf editiert.

Klasse	Methode	Entwickler
Player	all	Fabio
Team	all	Pascal
Seat	all	Pascal
Game	all	Thomas
Round	all	Thomas
Deck	all	Sandro
Card	all	Sandro
Suit	all	Sandro
Rank	all	Sandro

ServiceLocator	all	Fabio
Translator	all	Fabio
Client	all	Pascal / Fabio
Server	all	Fabio
ClientThread	all	Fabio
Message	all	Pascal / Fabio
HandType	all	Sandro / Thomas
Table	createDeck()	Thomas
	dealCards()	Thomas
	skip()	Sandro
	smallTichu()	Pascal
	bigTichu()	Pascal
	skipToTeamMember()	Sandro
	createSeat()	Pascal
	transferCards()	Fabio
	mahJongPlayed()	Thomas
	dogPlayed()	Sandro
	dragonPlayed()	Fabio
	phoenixPlayed()	Pascal
	playCards()	Fabio / Thomas
Countdown	startCountdown()	Sandro
	stopCountdown()	Sandro
	resetCountdown()	Sandro

## 6 Änderungen während des Projekts

In diesem Kapitel werden die Änderungen, welche an den SystemSpecifications während dem Projekt gemacht wurden, dargestellt.

## 6.1 Klassendiagramm Ist-Zustand

Visual Paradigm Online Diagrams Express Edition

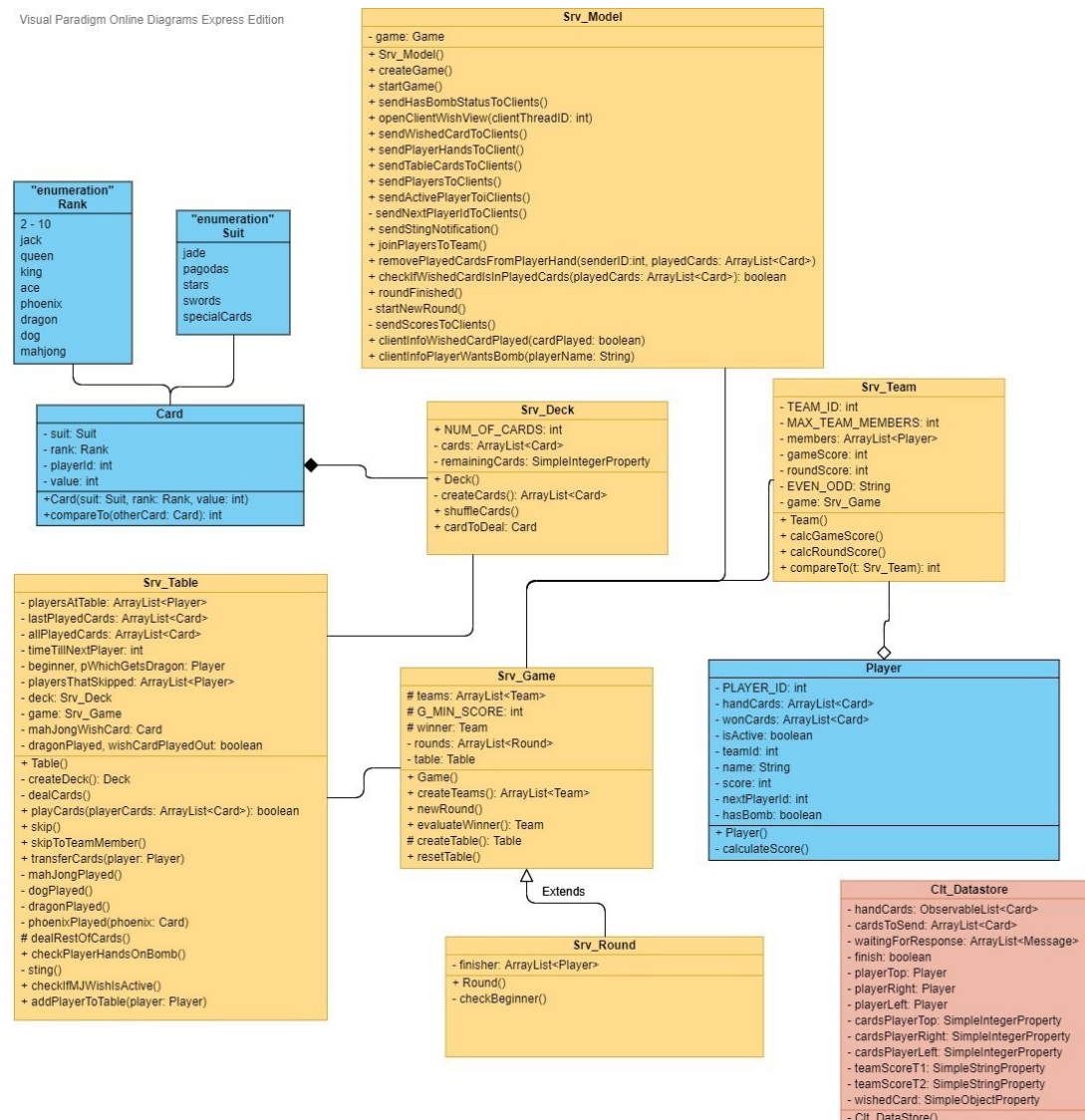


Abbildung 7 Ist-Zustand des Klassendiagramms Teil 1

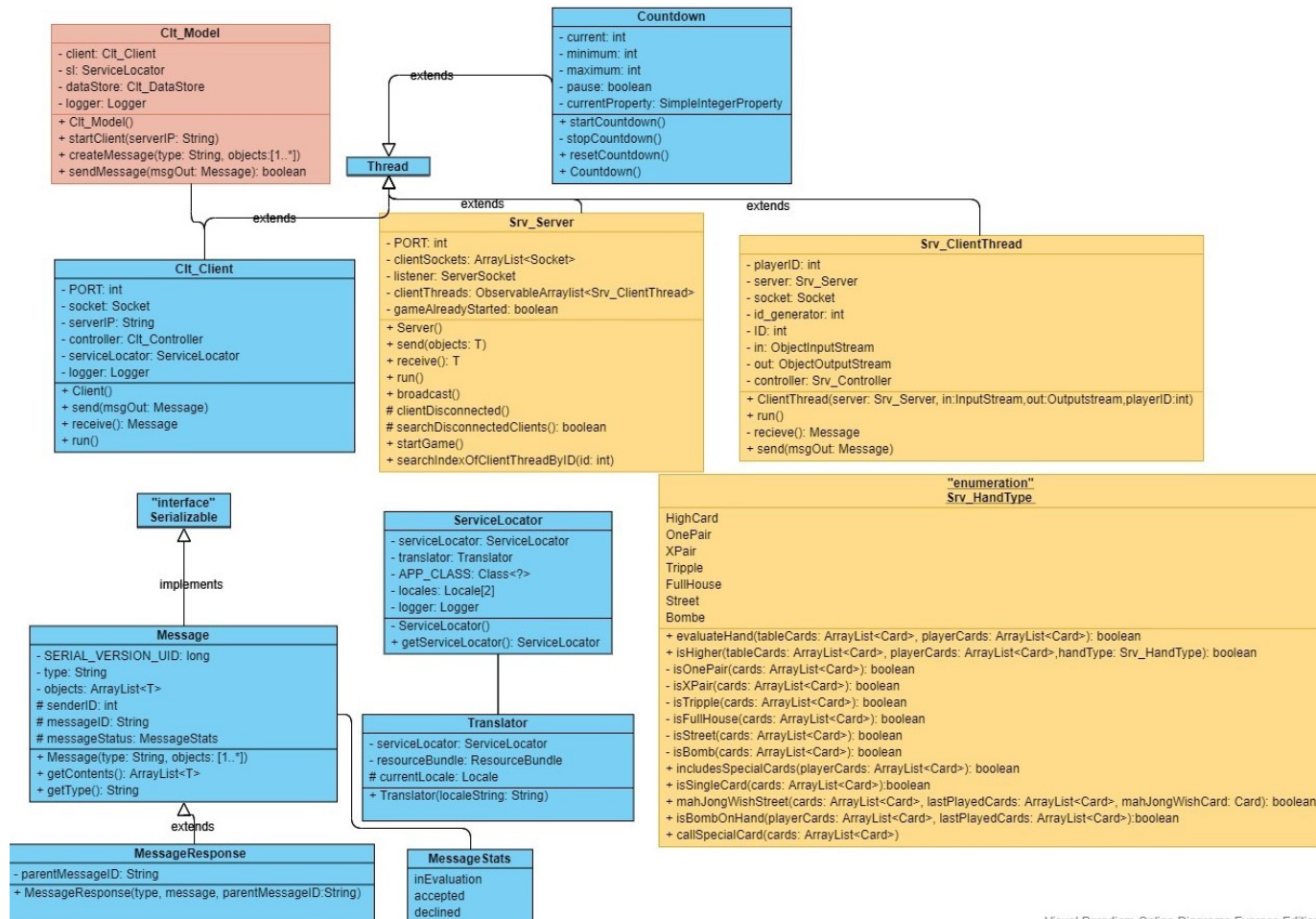


Abbildung 8 Ist-Zustand des Klassendiagramms Teil 2

Visual Paradigm Online Diagrams Express Edition



## 6.2 Anpassungen an der Aufteilung

Die definitive Aufteilung und Umsetzung einzelner Methoden und Klassen kann direkt im Code in den Kommentaren anhand der `@author`-Annotation erkannt werden.

Alle Methoden welche keine `@author`-Annotation aufweisen, sind durch alle Teammitglieder zu gleichen Teilen entwickelt worden.

## 6.3 Diverse Änderungen

1. Der Server wird aufgrund von Performanceproblemen nicht mehr durch den Client gestartet. Er muss unabhängig, direkt in der Entwicklungsumgebung gestartet werden.
2. Der Server benachrichtigt den Client nicht, dass er neue Daten abholen soll, sondern schickt ihm in der Antwort direkt die aktualisierten Daten zurück. Bezieht sich auf Kapitel 4.
3. Im Kapitel 4.2 werden nicht nach jedem Zug alle Karten jedes Spielers vom Client zum Server übermittelt. Der Server bezieht die Karten der Spieler direkt von der Table.
4. Im Kapitel 4.2, der Spieler muss die Bombe ausspielen, wenn er das Spiel mithilfe des Buttons zu sich geholt hat. Er kann diesen Vorgang nicht abbrechen. Sobald das Spiel bei ihm ist, wählt er die Karten aus und klickt auf Ausspielen.

## 7 Abbildungsverzeichnis

Abbildung 1 Client Server aufbau .....	5
Abbildung 2 Kommunikation Spielzug .....	8
Abbildung 3 Kommunikation Bombe Teil 1 .....	9
Abbildung 4 Kommunikation Bombe Teil 2 .....	9
Abbildung 5 Kommunikation Bombe Teil 3 .....	10
Abbildung 6 Kommunikation Passen .....	11
Abbildung 7 Ist-Zustand des Klassendiagramms Teil 1 .....	15
Abbildung 8 Ist-Zustand des Klassendiagramms Teil 2 .....	16