



AUGMENTED MULTI-PLAYER CARD GAME

Pflichtenheft zur Bachelor-Thesis

Studiengang:	Bsc Informatik
Autor:	Samuel Grimm
Betreuer:	Prof. Urs Künzler
Experte:	Dr. Federico Flueckiger
Datum:	20.01.2022

1	EINLEITUNG.....	4
2	SPIELBESCHRIEB.....	4
2.1	Kurzbeschrieb	4
2.2	Nebenrollen	5
2.3	Waffen.....	6
2.4	Aktionskarten.....	7
2.5	Weitere Gegenstandskarten	9
2.6	Lizenz.....	10
3	ARCHITEKTURBESCHREIBUNG/SCHNITTSTELLEN	10
3.1	Systemübersicht	10
3.2	Software-Architektur	11
4	ANFORDERUNGEN	12
4.1	Systemumfang	12
4.2	Akteure	13
4.3	Systemfunktionalität	14
4.4	Funktionale Anforderungen	15
4.5	Nicht funktionale Anforderungen/Qualitätsanforderungen	20
4.6	Einschränkungen/Randbedingungen/Abgrenzung	24
5	SPIELFLUSS	24
5.1	Fluss zwischen den einzelnen Screens (generell)	24
5.2	Fluss zwischen den einzelnen Screens während des Spieles	25
6	ABNAHMETESTS	27
6.1	Test der nichtfunktionalen Anforderungen.....	28
7	PROJEKTMANAGEMENT	28
7.1	Projektorganisation	28
7.2	Stakeholder und deren Aufgaben	28

7.3	Meilensteine	28
7.4	Gantt-Diagramm	29
7.5	Ausblick	29
8	ANHANG	30
9	INDEX	31
10	LITERATURVERZEICHNIS	31
11	ABBILDUNGSVERZEICHNIS	31

1 Einleitung

Ein neues, komplexes Kartenspiel kennenzulernen oder jemandem beizubringen, ist manchmal ziemlich schwer und kann frustrierend enden, weil Gesagtes oft nicht auf Anhieb klar ist und es viele Fragen gibt, die nicht immer gestellt werden können.

In einer solchen Situation wäre es manchmal praktisch, wenn man als Spieler zu jeder Karte angezeigt bekommt, wie und ob diese gespielt werden kann.

Hier kommt das Augmented Card Game «ins Spiel». Es führt alle Anfänger unkompliziert durch das Spiel hindurch und nimmt z.B. die Arbeit des (Lebens)Punkte-Zählens ab. Ausserdem soll es möglich sein, mit den Liebsten «am gleichen Tisch» zu spielen, auch wenn jemand nicht physisch am gleichen Ort ist.

Um das Spielgeschehen möglichst authentisch zu machen, sitzen (fast) alle am gleichen Tisch und können mit den virtuellen Karten auf dem Tisch interagieren.

In diesem «Augmented Card Game» soll es möglich sein, das Spiel «Bang!» zu spielen. Dieses Projekt soll ein Prototyp von diesem «Augmented Card Game» liefern.

Dieses Dokument ist das Pflichtenheft zu diesem Projekt. Zuerst wird das Kartenspiel grundsätzlich erklärt, dann wie es architektonisch geplant ist, welche Anforderungen das System erfüllen sollte, wie der Spielfluss aussehen wird und wie das System getestet werden soll.

2 Spielbeschreibung

2.1 Kurzbeschreibung

Das Spiel «Bang!» von daVinci Games und Abacusspiele ist ein rundenbasiertes Kartenspiel, das für 4 bis 7 Spieler geeignet ist. Jeder Spieler hat eine der folgenden Rollen:



Abbildung 1: Rollen des Spieles «Bang».

Die einzig bekannte Rolle ist diejenige des Sherifs, alle anderen sind unbekannt bis zu dessen Tod. Das Ziel des Sherifs und der Hilfssheriffs ist es, alle Banditen und den Gesetzlosen auszuschalten. Die Banditen wiederum wollen den Sheriff ausschalten. Der Gesetzlose ist auf keiner der beiden Seiten, er möchte der letzte Überlebende sein. Ein Spieler wurde eliminiert, wenn er/sie keine Lebenspunkte mehr hat.

Das Spielende ist erreicht, wenn eine der folgenden Situationen eintrifft:

Bedingung	Gewonnen hat
▪ Der Sherif wurde eliminiert	▪ Wenn nur noch der Gesetzlose lebt, dann er, ▪ ansonsten die Banditen
▪ Alle Banditen und der Gesetzlose wurden eliminiert	▪ Der Sherif und alle Hilfssheriffs

Das Spiel ist rundenbasiert. Konkret bedeutet das, dass jeweils nur ein Spieler an der Reihe ist und seinen Spielzug ausführen kann. Der Spielzug ist in drei Phasen unterteilt:

1. Phase: Der Spieler zieht 2 Karten vom Nachziehstapel.
2. Phase: Beliebig viele Karten spielen.
3. Phase: Überzählige Karten abspielen: Am Ende des Spielzuges darf der Spieler maximal so viele Karten auf der Hand haben wie er/sie Lebenspunkte hat.

Nach dem Spielzug kommt der nächste Spieler links an die Reihe.

2.2 Nebenrollen

Alle Spieler haben eine zweite Rolle. Am Anfang des Spieles werden jedem Spieler 2 Charakterkarten zugeteilt. Nur eine davon kann der Spieler ansehen. Ist er/sie mit der aufgedeckten Charakterkarte nicht zufrieden, so deckt er/sie die zweite Charakterkarte auf und die erste Charakterkarte gilt somit nicht mehr für ihn/sie. Ansonsten gilt die als erstes aufgedeckte Charakterkarte.

Die Nebenrollen/Charakterkarten bestimmen, wie viele Lebenspunkte ein Spieler maximal haben kann. Diese sind mit der Anzahl Patronen rechts auf der Karte gekennzeichnet. Der Sherif hat einen Lebenspunkt mehr.



Abbildung 2: Charakterkarten, nicht vollständig.

Zusätzlich haben alle Charaktere eine Spezialfunktion, die auf der Karte beschrieben ist.

2.3 Waffen

Waffen-Karten werden während des Spielzuges ausgespielt. Alle Spieler dürfen maximal 1 Waffe vor sich spielen. Möchte der Spieler eine andere Waffe, so kann er/sie während des Spielzuges eine bereits vor sich gespielte Karte wegwerfen.

Die Waffen bestimmen die Reichweite, die der Spieler hat. Ohne Waffe hat der Spieler eine Reichweite von 1. Dies bedeutet, dass er/sie nur die Spieler, die direkt neben ihm/ihr sind, mit der Karte «Bang!», treffen kann.

Eine spezielle Waffe ist die Volcanic: Hat ein Spieler diese Karte vor sich gespielt, so kann er beliebig viele «Bang!-Karten pro Spielzug spielen, ansonsten ist nur 1 «Bang!» pro Spielzug erlaubt.

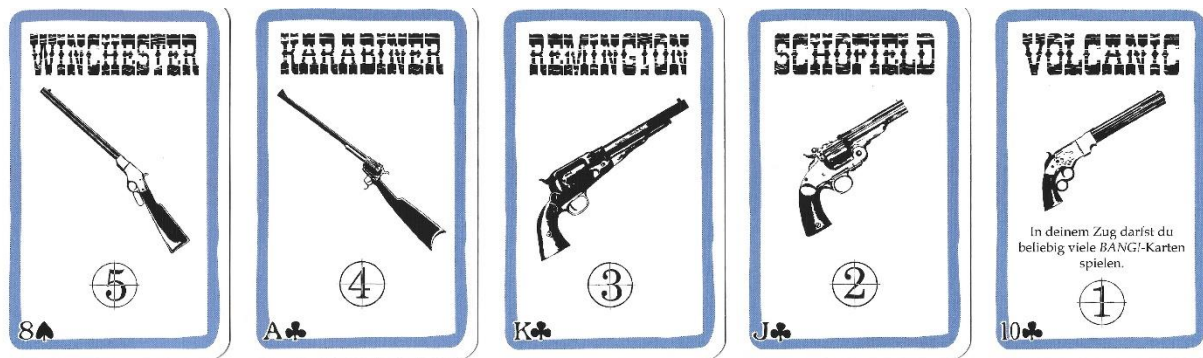


Abbildung 3: Waffenkarten

2.4 Aktionskarten

Die wichtigste Karte im Spiel ist die «Bang»-Karte. Diese Karte kann ein Spieler während seines Spielzuges gegen einen anderen Spieler spielen, der in seiner Reichweite liegt. Der angegriffene Spieler darf gegen eine «Bang»-Karte mit einem «Fehlschuss» reagieren (also von der Hand abwerfen). Tut er dies nicht, so verliert der angegriffene Spieler 1 Lebenspunkt.

Während des Spielzuges dürfen beliebig viele «Bier»-Karte gespielt werden. Pro solche Karte erhält dieser Spieler einen Lebenspunkt dazu. Die maximale Lebenspunkte dürfen aber nicht überschritten werden. Das Einzigartige dieser Karte ist, dass diese auch ausserhalb des Spielzuges von der Hand gespielt werden darf, wenn derjenige Spieler von einem Schuss getroffen wird.



Abbildung 4: Die wichtigsten Aktionskarten des Spieles

Die Symbole werden auf der Hilfskarte gut zusammengefasst:



Abbildung 5: Zusammenfassung der Symbole auf den Karten

Neben dem «Bang» und «Bien» gibt es weitere Möglichkeiten, anderen Lebenspunkte abzuziehen oder hinzuzufügen.

Der «Catling» zieht allen anderen Spielern 1 Lebenspunkt ab, die diesen Angriff nicht mit einem «Fehlschuss» abwehren. Der «Saloon» macht genau das Gegenteil: Alle Spieler erhalten einen Lebenspunkt zurück.

Wird der «Indianer» gespielt, so verlieren alle anderen Spieler einen Lebenspunkt, die nicht eine «Bang»-Karte spielen können.

Ein «Duell» wird hingegen gegen einen Spieler gespielt, den man auswählen kann. Anschliessend werfen der angegriffene Spieler und der Spieler, der an der Reihe ist, abwechselungsweise eine «Bang»-Karte ab. Derjenige, der zuerst keine «Bang»-Karte mehr spielt, verliert einen Lebenspunkt.



Abbildung 6: Karten, mit welchen andere Lebenspunkte verlieren oder bekommen.

Mit einem «Wells Fargo» oder einer «Postkutsche» können 2 bzw. 3 Karten vom Nachziehstapel gezogen werden. Bei einem «Warenhaus» werden so viele Karten aufgedeckt, wie noch Spieler im Spiel sind. Anschliessen können alle Spieler beginnend beim Spieler, der an der Reihe ist, weiter in Spielrichtung eine Karte davon auf die Hand nehmen.

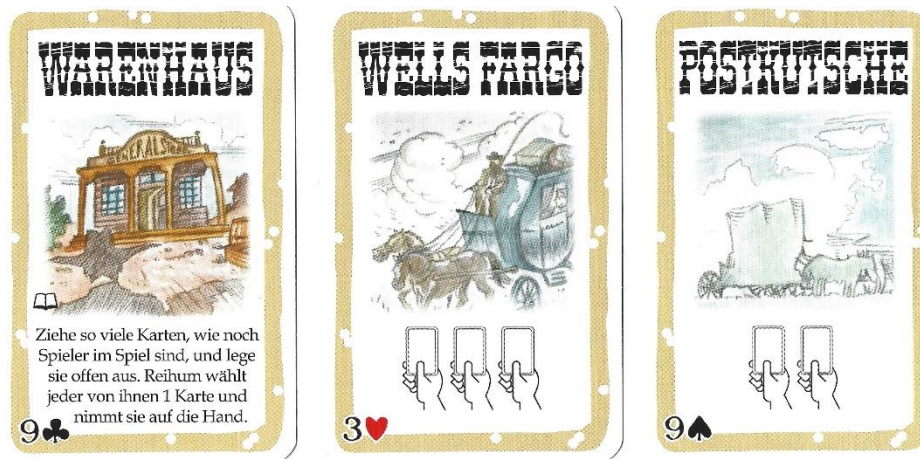


Abbildung 7: Karten, die bewirken, dass Karten gezogen werden können.

Mit dem «Cat Balou» kann der Spieler einen beliebigen anderen Spieler dazu zwingen, eine Handkarte abzuwerfen. Mit der «Panik»-Karte hingegen kann der Spieler eine beliebige Karte von einem anderen Spieler im Abstand 1 aufnehmen, die entweder vor dem anderen Spieler ausgespielt ist oder sich in der Hand des Spielers befindet.

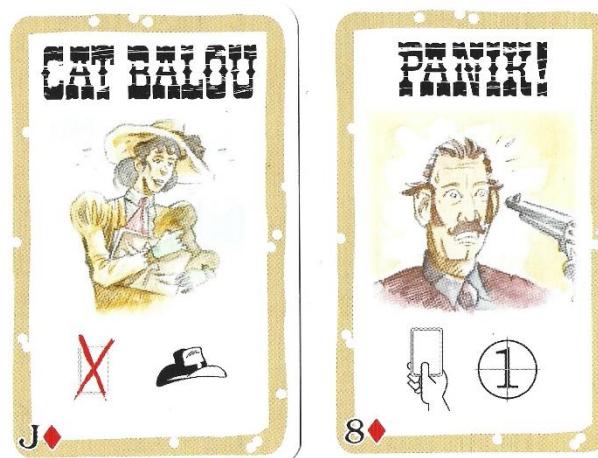


Abbildung 8: Karten, die eine Karte einem anderen Spieler entfernen.

2.5 Weitere Gegenstandskarten

So wie die Waffen werden diese blauen Karten vor einem Spieler gespielt.

Das «Dynamit» wird zuerst vor sich selbst abgespielt. Befindet sich diese Karte am Anfang des Spielzuges vor einen, muss man am Anfang des Spielzuges zuerst eine Karte aufziehen und wegwerfen. War unten links auf dieser Karte eine Schaufel, verliert dieser Spieler 3 Lebenspunkte. Ansonsten wandert diese Karte zum Spieler links von ihm/ihr.

Das «Gefängnis» kann vor einen beliebigen Spieler gespielt werden, der nicht Sheriff ist. Hat der Spieler am Anfang seines Zuges das Gefängnis vor sich, muss er zuerst eine Karte vom Stapel ziehen. Hatte diese Karte unten links ein Herz abgebildet, so kann er das Gefängnis wegwerfen und normal seinen Zug starten. Ansonsten kann er auch das Gefängnis wegwerfen, muss aber seinen Spielzug passen.

Mit dem «Zielfernrohr» oder «Mustang» sehen Spieler einen mit einem um 1 grösseren bzw. kleineren Abstand, was bedeutet, dass sich die Reichweite entsprechend um 1 verändert.

Das «Fass» kann der Spieler vor sich selbst spielen. Dies bewirkt, dass immer, wenn auf ihn geschossen wird, er eine Karte vom Stapel aufdecken kann. Falls unten links auf dieser Karte ein Herz war, so zählt diese Karte als «Fehlschuss». Das «Fass» bleibt aber bestehen nach dem Schuss.



Abbildung 9: weitere blaue Karten, die vor Spieler gespielt werden.

2.6 Lizenz

Das Spiel wurde von daVinci Games und Abacusspiele herausgegeben und ist urheberrechtlich geschützt. Für diese Arbeit sind keine Lizenzen ausgestellt worden. Aus diesem Grund darf diese Bachelor-Thesis nicht kommerziell verwendet werden.

3 Architekturbeschreibung/Schnittstellen

3.1 Systemübersicht

Aus Hardware-Sicht ist das System so geplant:

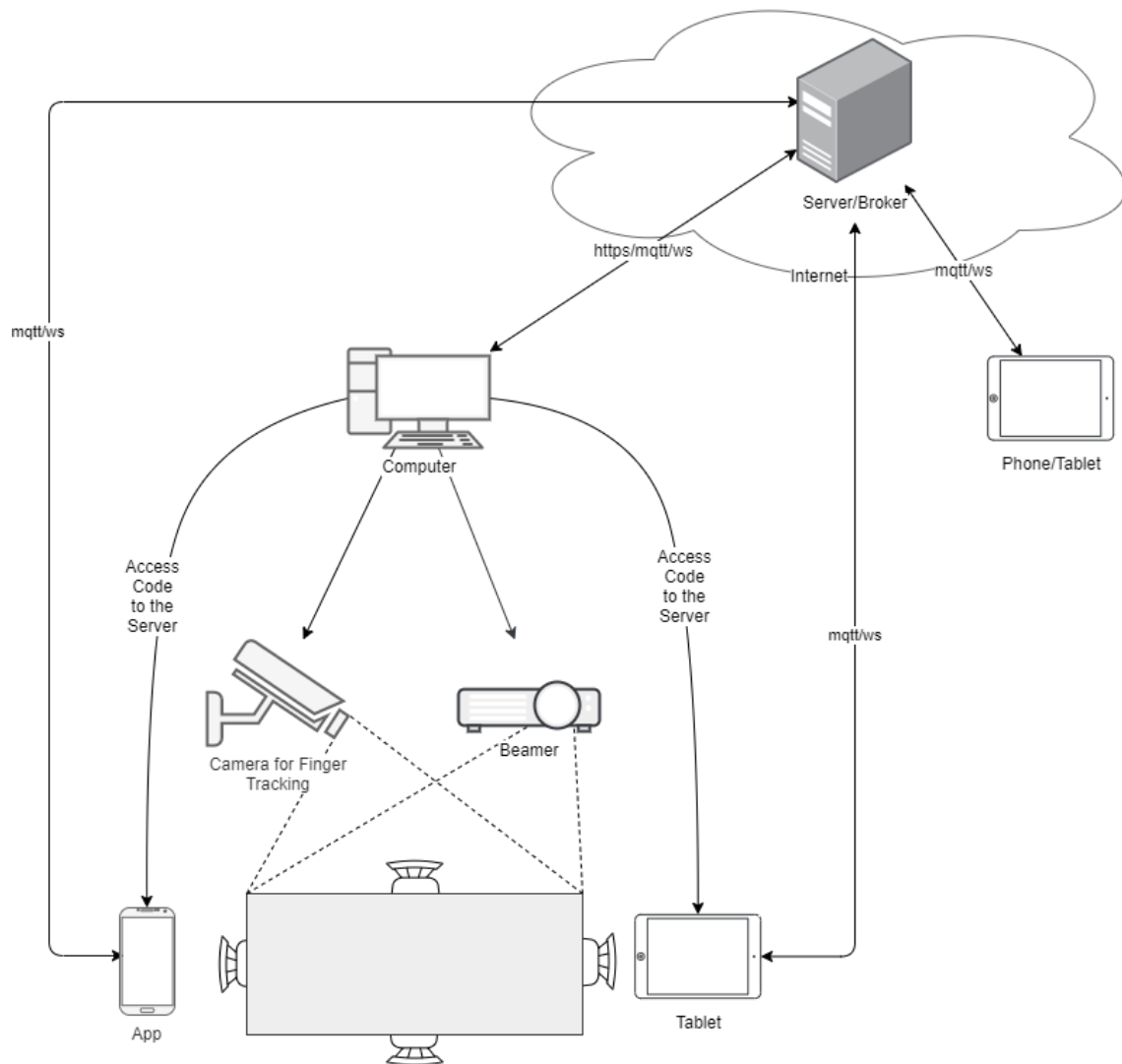


Abbildung 10: Hardware-Ansicht des Systems

Bis zu 4 Spieler sitzen an einem Spieltisch. Alle Spieler haben ihre Karten auf ihrem Smartphone oder Tablet. Auf dem Tisch vor ihnen wird das aktuelle Spielgeschehen mit einem Beamer projiziert. Wird von einem physisch am Tisch sitzenden Spieler eine Karte auf dem Spieltisch ausgewählt, so soll dies mit Handbewegung möglich sein. Hierfür kommt eine Kamera zum Einsatz, die die Handbewegungen der Spieler trackt.

Der Beamer und die Kamera für das Hand-Tracking sind mit einem Computer verbunden, auf dem das Spiel läuft.

Alle Tablets und Smartphones mit der App dieses Systems sind via einen Server indirekt mit diesem Computer verbunden. Damit sich die Apps korrekt verbinden können, wird ein Code (QR-Code oder ein Text-/Zahlencode) auf den Tisch gebeamt, der dann eingegeben bzw. gescannt werden kann. Dieser Code soll auch von solchen verwendet werden können, die nicht physisch am Tisch sitzen können.

3.2 Software-Architektur

Software-technisch ist folgende Architektur geplant:

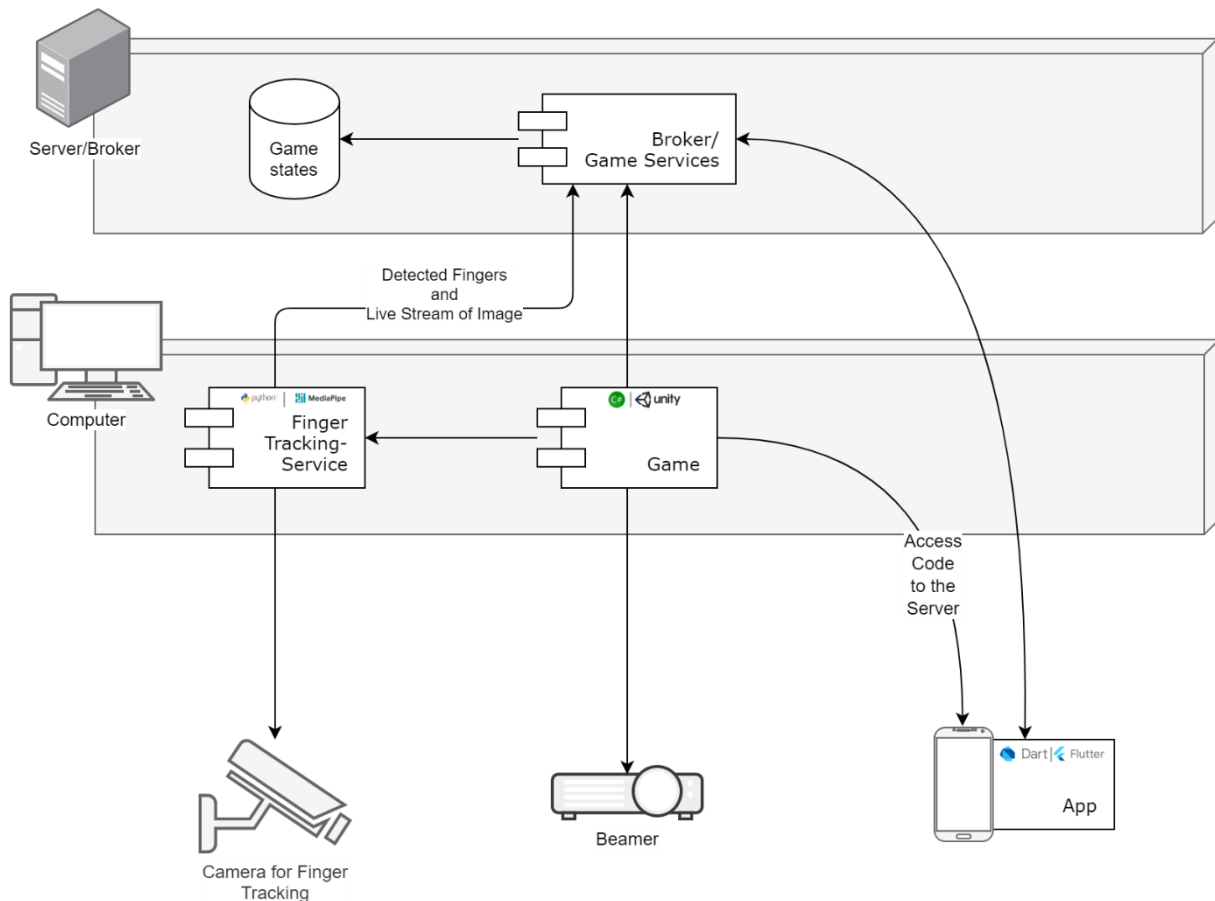


Abbildung 11: Software-Architektur des Systems

Das eigentliche Game läuft auf einem Computer, das mit der Kamera und dem Beamer verbunden ist. Das Game bekommt die Handtracking-Daten von einem «Finger Tracking»-Service. Das Game soll ausserdem diesen Service konfigurieren können. Des Weiteren kann das Game auch festlegen, ob das Live-Bild der Kamera auf den Server gestreamt werden soll, damit dieses Bild dann auf der App abrufbar ist. Das Game ist zusätzlich für die Spiellogik und das projizierte Bild verantwortlich.

Den aktuellen Spielstand muss auf einen Server aktualisiert werden, damit in der App der aktuelle Spielstand angezeigt werden kann.

In der Smartphone- und Tablet-App müssen die Handkarten des Spielers angezeigt werden. Des Weiteren wird angestrebt, dass alle Spielhandlungen auch mit der App durchgeführt werden können. Dies bedeutet, dass das projizierte Bild in einer ähnlichen Weise auch auf der App ersichtlich sein muss, und dass die Aktionen, die mit dem 4.1. Finger-Tracking getätigt werden, auch mit der App ausführbar sein müssen.

4 Anforderungen

4.1 Systemumfang

Muss-Kriterien

- Das Spiel wird mit Beamer auf Tisch projizieren
- Der Spielstand wird via einen Server aktuell gehalten, damit die App den aktuellen Spielstand anzeigt.
- In der App müssen die Handkarten angezeigt werden und spielbar sein.

- Alle Aktionen, die mit Hand-Tracking möglich sein sollen, müssen auch mit der App ausführbar sein (z.B., wenn die Kamera nicht ordnungsgemäss funktioniert).
- Der verwendete Server gibt nur auf authentifizierten Anfragen Informationen zum Spiel heraus (anonyme Authentifizierung kann reichen).
- Ein minimales Gameplay (wie in den Anforderungen definiert) muss spielbar sein.
- Die Sitzpositionen am Tisch müssen definiert werden können.

Soll-Kriterien

- 4.1.2
- Die Spieler, die physikalisch vor dem Tisch sitzen, sollen bestimmte Handlungen mittels Handtracking mit Kamera auf dem Tisch direkt ausführen können.
 - Der Hand-Tracking-Service soll konfigurierbar sein (z.B. Bildausschnitt definieren, Streamen des Bildes an-/ausschalten)
 - Die App soll Hilfestellungen zu jeder Karte zur Verfügung stellen.

Kann-Kriterien (optional)

- 4.1.3
- Damit die Spieler ganz einfach dem Spiel beitreten können, könnte ein QR-Code projiziert werden, der mit der App gescannt wird.
 - Bild-Übertragung der Smartphone- oder Tablet-Frontkamera durch die App und evtl. sogar Projektion mit Beamer auf den Spieltisch, damit diejenigen am Spieltisch die abwesenden sehen können und umgekehrt (nicht geplant im Rahmen der Bachelor-Thesis)
 - Tutorial-Modus, damit die Spieler das Spiel einfacher lernen können.

4.2 Akteure

Folgende Akteure sind für das Endprodukt relevant:

Stakeholder	Relevanz	Haltung	Ziele und Interessen
Administrator/Besitzer des Systems	Hoch	Positiv	Ein einfach konfigurierbares System, das bei Gebrauch verfügbar ist. Er/sie möchte mit Freunden und Bekannten das Kartenspiel «Bang» augmented spielen und Spass haben. Er möchte nicht viel Zeit mit Konfigurieren verbringen müssen.
Spieler am Tisch	Hoch	Positiv, verwirrt, interessiert oder skeptisch	Diese Spieler möchten schnell in das Spiel eintauchen können. Es wäre eine Zumutung, dass diese Spieler zuerst einen Benutzer-Account mit Verifizierung machen müssten oder vor dem Spiel Änderungen am Smartphone/Tablet (wie z.B. sich im gleichen WLAN befinden wie der Game-Computer) oder an der App machen müsste. Er/sie soll vor und während dem Spiel so wenig wie möglich falsch machen können. Möglicherweise kennt dieser Spieler das Spiel nicht. Deswegen soll das Spiel/System so selbsterklären wie möglich für ihn/sie sein.

Spieler, der nicht physisch am Tisch sein kann	Mittel	Positiv, verwirrt	<p>Gleiche Ziele wie der Spieler am Tisch.</p> <p>Im Gegensatz zu den anderen Spielern kann er/sie nicht direkt mit den anderen Spielern kommunizieren. Deswegen muss dieser Spieler so wenig Fehler machen können wie möglich.</p> <p>Möglicherweise (empfohlen) ist er/sie mit den anderen Spielern via Call (Skype, Discord, MS Teams, Zoom, etc.) verbunden. Vielleicht wünscht er sich, dass sich die Spieler gegenseitig sehen könnten. Aus diesem Grund kann es Sinn machen, das Webcam-Bild via App zu übertragen oder sogar sein eigenes Bild mit dem Beamer zu projizieren.</p>
Betreuer	Hoch	Positiv	Fordert ein gut funktionierendes und dokumentiertes System.

4.3 Systemfunktionalität

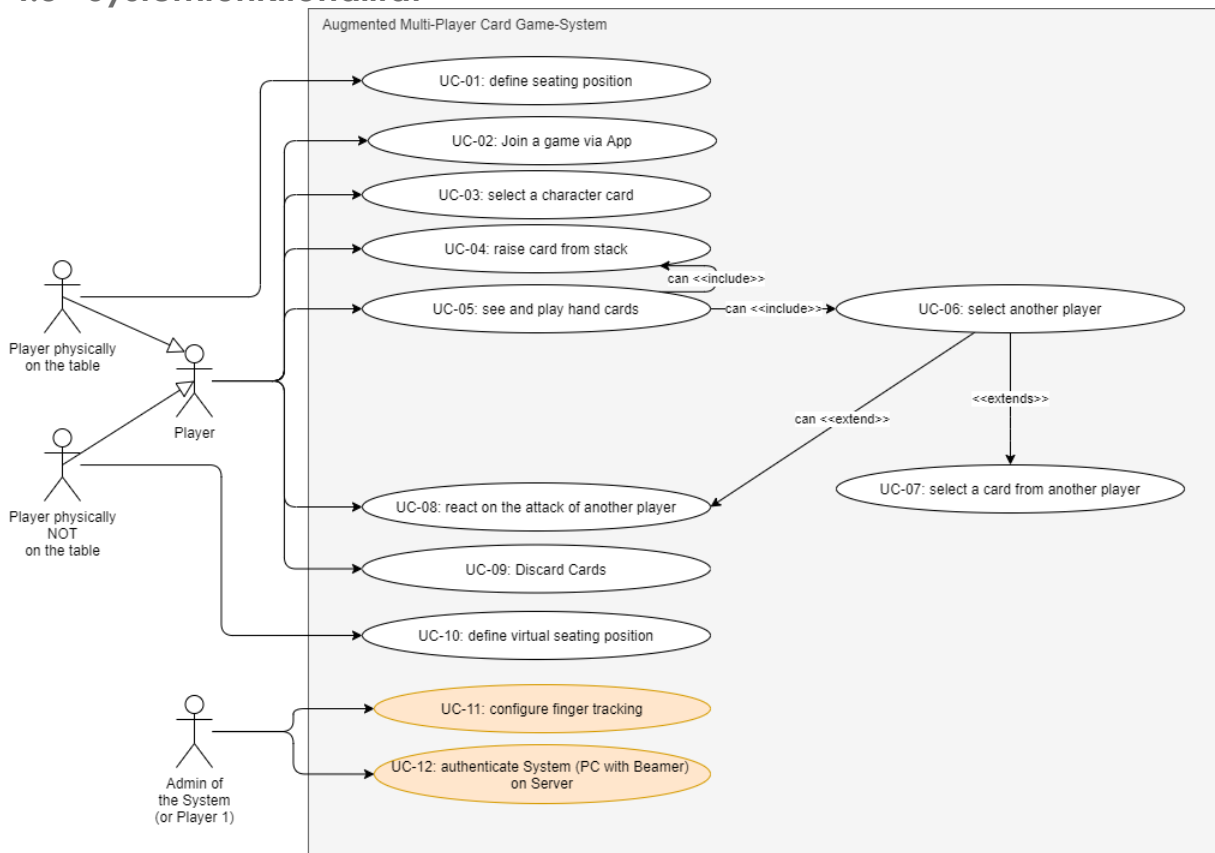


Abbildung 12: Anwendungsfälle des Systems

4.4 Funktionale Anforderungen

Minimales Gameplay

4.4.1.1 FA-01 – distribute roles

Ziel	Das System weist den Spielern eine der folgenden Rollen zu: Sheriff, Hilfssherif, Bandit oder Outlaw. Alle Spieler sehen ihre Rolle auf ihrem Smartphone/Tablet.		
Use Case	(keiner, da dies automatisch passiert)		
Begründung	Im Spiel müssen alle Teilnehmenden eine Rolle haben, damit diese ihren Auftrag erfüllen können.		
Priorität	MUSS	Version	1.0

4.4.1.2 FA-02 show who's the sheriff

Ziel	Auf dem projizierten Bild ist ersichtlich, welcher Spieler der Sheriff ist.		
Use Case	(keiner, da dies automatisch passiert)		
Begründung	Die Rolle des Sheriffs muss bekannt sein.		
Priorität	MUSS	Version	1.0

4.4.1.3 FA-03 distribute and select character card

Ziel	Jedem Spieler werden zwei zufällige Charakterkarten verteilt. Nur eine davon wird dem Spieler sichtbar gezeigt. Von beiden kann jeder Spieler eine davon auswählen. Die ausgewählte Charakterkarte wird bei jedem Spieler sichtbar für die anderen angezeigt.		
Use Case	UC-03: select a character card		
Begründung	Alle Spieler müssen eine Zusatzrolle haben, die bestimmt, wie viele Lebenspunkte sie maximal haben können.		
Priorität	MUSS	Version	1.0

4.4.1.4 FA-04 receive life points in the beginning of the game

Ziel	Am Anfang des Spieles erhalten alle Spieler so viele Lebenspunkte, wie es ihre Zusatzrolle zulässt. Der Sheriff erhält einen mehr. Die Anzahl Punkte werden auf dem projizierten Bild angezeigt.		
Use Case	UC-03: select a character card		
Begründung	Die Lebenspunkte sind ein zentrales Element des Spieles. Alle Spieler müssen sehen können, welcher Spieler wie viele Lebenspunkte noch besitzen.		
Priorität	MUSS	Version	1.0

4.4.1.5 FA-05 automatically raise cards in the beginning of the game

Ziel	Allen Spielern werden automatisch so viele Handkarten verteilt, wie Lebenspunkte sie besitzen.		
Use Case	(keiner, da dies automatisch passiert)		
Begründung	Vor dem ersten Spielzug müssen alle Spieler bereits Karten auf der Hand haben.		
Priorität	MUSS	Version	1.0

4.4.1.6 FA-06 raise cards in the beginning of the player's move

Ziel	Am Anfang jedes Spielzuges kann (und muss) der Spieler, der an der Reihe ist, 2 Karten aufziehen.		
Use Case	UC-04: raise card from stack		
Begründung	Vorgabe des Spieles.		
Priorität	MUSS	Version	1.0

4.4.1.7 FA-07 select and play hand cards

Ziel	Während des eigenen Spielzuges kann der Spieler auf seinem Gerät spielbare Karten auswählen und spielen.		
Use Case	UC-05: see and play hand cards		
Begründung	Alle Spieler müssen Aktionen ausführen können.		
Priorität	MUSS	Version	1.0

4.4.1.8 FA-08 discard hand cards

Ziel	Der Spieler, der an der Reihe ist, kann beliebig viele Karten auf seiner Hand wegwerfen.		
Use Case	UC-09: discard cards		
Begründung	Alle Spieler dürfen am Ende ihres Spielzuges nur so viele Karten auf der Hand haben, wie Lebenspunkte sie besitzen.		
Priorität	MUSS	Version	1.0

4.4.1.9 FA-09 max amount of hand cards

Ziel	Der Spieler, der an der Reihe ist, kann seinen Spielzug erst beenden, sobald er/sie maximal so viele Handkarten wie Lebenspunkte besitzt.		
Use Case	UC-09: discard cards		
Begründung	Alle Spieler dürfen am Ende ihres Spielzuges maximal so viele Handkarten besitzen, wie sie Lebenspunkte haben.		
Priorität	MUSS	Version	1.0

4.4.1.10 FA-10 play weapons

Ziel	Während des Spielzuges kann der Spieler Waffenkarten spielen. Diese Karte wird dann vor dem Spieler angezeigt. Die Waffenkarte erhöht die Reichweite des Spielers entsprechend.		
Use Case	UC-05: see and play hand cards		
Begründung	Waffen werden benötigt, um weiter entfernte Spieler erreichen zu können.		
Priorität	MUSS	Version	1.0

4.4.1.11 FA-11 play «Bang!»

Ziel	Spielt der Spieler eine «Bang!»-Karte, so startet ein Modus, in welchem er/sie einen anderen Spieler auswählt. Kann dieser ausgewählte Spieler nicht mit einem Fehlschuss den Angriff abwehren, verliert dieser einen Lebenspunkt.		
Use Cases	<ul style="list-style-type: none"> ▪ UC-05: see and play hand cards 		

	<ul style="list-style-type: none"> ▪ UC-06: select another player ▪ UC-08: react on the attack of another player 		
Begründung	Elementare Karte des Spieles.		
Priorität	MUSS	Version	1.0

4.4.1.12 FA-12 react with «Fehlschuss» on «Bang» or «Catling»

Ziel	Alle Spieler, auf welche mit einer «Bang»- oder «Catling»-Karte geschossen wurde, haben die Möglichkeit, diesen Schuss mit einer «Fehlschuss»-Karte abzuwehren. Hierfür wird ein Modus gestartet, in welchem der/die im Beschuss stehende(n) Spieler aufgefordert werden, entweder eine «Fehlschuss»-Karte zu spielen oder ein Lebenspunkt abzugeben.		
Use Case	UC-08: react on the attack of another player		
Begründung	Alle Spieler sollten die Möglichkeit haben, auf eine «Bang»- oder «Catling»-Karte zu reagieren		
Priorität	MUSS	Version	1.0

4.4.1.13 FA-13 automatically discard played cards

Ziel	Jede Karte, die gespielt wird, wird sichtbar auf den Stapel der gespielten Karte abgelegt. Diese Karte befindet sich ab diesem Zeitpunkt nicht mehr auf der Hand des Spielers.		
Use Case	(keiner, da dies automatisch passiert)		
Begründung	Gespielte Karten sollen nicht mehrmals von einem Spieler gespielt werden können.		
Priorität	MUSS	Version	1.0

4.4.1.14 FA-14 die

Ziel	Sobald ein Spieler keine Lebenspunkte mehr besitzt, kann dieser nicht mehr mitspielen. Des Weiteren wird auf dem projizierten Bild sowie auf seinem Gerät angezeigt, dass dieser Spieler ausgeschieden ist.		
Use Case	(keiner, da dies automatisch passiert)		
Begründung	Vorgabe des Spiels.		
Priorität	MUSS	Version	1.0

4.4.1.15 FA-15 check if and who has won the game

Ziel	Immer, wenn ein Spieler im Spiel stirbt, wird evaluiert, ob das Spiel bereits beendet ist. Dies ist der Fall, wenn entweder <ul style="list-style-type: none"> • Der Sheriff getötet wurde. Dann haben die Banditen gewonnen (und der Outlaw, falls er noch lebt.) • Oder alle Banditen und Outlaws getötet wurden. In diesem Fall haben der Sheriff und seine Hilfssherifen gewonnen. Es wird angezeigt, wer gewonnen hat.		
Use Case	(keiner, da dies automatisch passiert)		
Begründung	Das Spiel muss ein Ende mit Gewinnern und Verlierern finden können.		
Priorität	MUSS	Version	1.0

Erweitertes Gameplay

4.4.2.1 FA-16 play «Been»

Ziel	Wird eine «Been»-Karte während des Spielzuges gespielt, so erhält der Spieler 1 Lebenspunkt. Besitzt der Spieler bereits die maximale Anzahl an Lebenspunkten, so hat diese Karte keine Funktion.		
Use Case	UC-05: see and play hand cards		
Begründung	Wichtige Karte des Spieles		
Priorität	SOLL	Version	1.0

4.4.2.2 FA-17 play «Catling»

Ziel	Wird während des Spielzuges ein «Catling» gespielt, so wird automatisch auf alle Spieler geschossen. Alle Spieler haben die Gelegenheit, einen «Fehlschuss» zu spielen, ansonsten verlieren sie einen Lebenspunkt.		
Use Case	UC-05: see and play hand cards		
Begründung	Wichtige Karte des Spieles		
Priorität	SOLL	Version	1.0

4.4.2.3 FA-18 play «Postkutsche» and raise two cards

Ziel	Wird eine «Postkutsche» während des Spielzuges gespielt, so kann der Spieler 2 Karten vom Stapel ziehen.		
Use Case	<ul style="list-style-type: none"> ▪ UC-05: see and play hand cards ▪ UC-04: raise card from stack 		
Begründung	Wichtige Karte des Spieles		
Priorität	SOLL	Version	1.0

4.4.2.4 FA-19 play «Indianen»

Ziel	Wird die «Indianen»-Karte während des Spielzuges gespielt, so wird ein Modus gestartet, in welchem alle anderen Spieler eine «Bang»-Karte spielen können. Alle Spieler, die keine «Bang»-Karte spielen konnten, verlieren einen Lebenspunkt.		
Use Case	<ul style="list-style-type: none"> ▪ UC-05: see and play hand cards ▪ UC-08: react on the attack of another player 		
Begründung	Elementare Karte des Spieles.		
Priorität	SOLL	Version	1.0

4.4.2.5 FA-20 play «Duel»

Ziel	Wird eine «Duell»-Karte während des Spielzuges gespielt, wird ein Modus gestartet, in welchem der aktuelle Spieler einen anderen Spieler auswählen kann. Dies wiederum führt zu einem Modus, in welchem der aktuelle und der ausgewählte Spieler abwechselungsweise eine «Bang»-Karte von der Hand spielen können. Der ausgewählte Spieler muss beginnen. Derjenige Spieler, der als erstes keine «Bang»-Karte mehr spielt, verliert einen Lebenspunkt.		
Use Cases	<ul style="list-style-type: none"> ▪ UC-05: see and play hand cards 		

	<ul style="list-style-type: none"> ▪ UC-06: select another player ▪ UC-08: react on the attack of another player 		
Begründung	Elementare Karte des Spieles.		
Priorität	SOLL	Version	1.0

- Catling, Saloon, Warenhaus, Wells Fargo/Postkutsche, Cat Balou, Panik, (Dynamit, Gefängnis, Zielfernrohr, Mustang, Fass)

4.4.2.6 FA-21 Character card with specific function

Ziel	Die Funktionen der Charakterkarte werden angewendet.		
Use Case	UC-03: select a character card		
Begründung	Die Funktionen der Zusatzrollen bringen mehr Abwechslung ins Gameplay.		
Priorität	OPTIONAL	Version	1.0

Allgemeine Anforderungen an das System

4.4.3.1 FA-22 login/register playing table on the server

Ziel	Mit dem Computer, der das Spiel projizieren soll, kann sich der Besitzer beim Server anmelden und registrieren.		
Use Case	UC-12: authenticate System (PC with Beamer)		
Begründung	Nur angemeldete Personen sollen Zugriff auf die Daten des Systems haben.		
Priorität	SOLL	Version	1.0

4.4.3.2 FA-23 join game via app

Ziel	Die Spieler können mittels der App einem Spiel beitreten. Hierfür soll der Spieler in der App einen Code eingeben/einscannen, damit er sich automatisch direkt mit dem richtigen Spieltisch verbinden kann.		
Use Case	UC-02: Join a game via App		
Begründung	Die Spieler müssen sich via Smartphone oder Tablet mit dem System verbinden können.		
Priorität	MUSS	Version	1.0

4.4.3.3 FA-24 enter user nickname

Ziel	In der App soll es eine Möglichkeit geben, dass jeder Spieler einen Nickname eingeben kann, damit dieser z.B. auf dem Spieltisch auch angezeigt werden kann.		
Use Case	(noch nicht vorhanden)		
Begründung	Es kann praktisch sein, wenn man als Spieler auf dem Tablet oder Tisch sieht, wer an einer bestimmten Position sitzt.		
Priorität	KANN	Version	1.0

4.4.3.4 FA-25 select camera for finger tracking

Ziel	Wenn Fingertracking eingesetzt wird, soll der Admin/Spieler 1 die Möglichkeit haben, die richtige Kamera hierfür einzusetzen.		
Use Case	UC-11: configure finger tracking		
Begründung	Die meisten Computer haben mehrere Kameras. Deswegen soll es möglich sein, die richtige Kamera auszuwählen.		
Priorität	SOLL	Version	1.0

4.4.3.5 FA-26 crop the playing field for more accurate finger tracking position

Ziel	Wenn Fingertracking eingesetzt wird, soll der Admin/Spieler 1 die Möglichkeit haben, mit der App anzugeben, wo auf der Kamera sich das projizierte Spielfeld befindet. Hiermit soll die Genauigkeit der errechneten Finger-Positionen verbessert werden.		
Use Case	UC-11: configure finger tracking		
Begründung	Meistens ist die Kamera nicht 100%ig senkrecht auf den Spieltisch gerichtet. U.a. deswegen kommt es zu kleinen Verzerrungen. Diese sollen hiermit ein bisschen korrigiert werden.		
Priorität	KANN	Version	1.0

4.4.3.6 FA-27 define seating position

Ziel	In der App müssen alle Spieler die Möglichkeit haben, ihre Sitzposition zu bestimmen.		
Use Case	<ul style="list-style-type: none"> ▪ UC-01: define seating position ▪ UC-10: define virtual seating position 		
Begründung	Damit die Karten beim am richtigen Ort projiziert und die Finger der richtigen Person zugeordnet werden können, muss das System wissen, wo sich welche Person befindet.		
Priorität	MUSS	Version	1.0

4.5.1 4.5 Nicht funktionale Anforderungen/Qualitätsanforderungen

Qualitätsbestimmungen

Die folgende Tabelle soll ungefähr bestimmen, welche allgemeinen Qualitätsbestimmungen wie relevant für das zu entwickelnde System sein sollen. Bei der Planung und Implementierung sollen Merkmale mit hoher Relevanz eine grössere Relevanz spielen.

Produktqualität	Sehr hoch	Hoch	Normal	Nicht relevant
Funktionalität				
Angemessenheit			X	
Richtigkeit			X	
Interoperabilität			X	
Ordnungsmässigkeit			X	
Sicherheit			X	
Zuverlässigkeit				
Reife			X	
Fehlertoleranz		X		

Wiederherstellbarkeit		X
Benutzbarkeit		
Verständlichkeit	X	
Erlernbarkeit	X	
Bedienbarkeit	X	
Effizienz		
Zeitverhalten		X
Verbrauchsverhalten		X
Änderbarkeit		
Analysierbarkeit		X
Modifizierbarkeit		X
Prüfbarkeit		X
Übertragbarkeit		
Anpassbarkeit		X
Installierbarkeit		X
Konformität		X
Austauschbarkeit		X

Tabelle 1: Qualitätsbestimmungen für das zu entwickelnde System.

Systemumfeld

4.5.2 Um sich das Systemumfeld besser vorstellen zu können, ist ein Blick auf Abbildung 11: Software-Architektur des Systems empfehlenswert.

4.5.2.1 Hardware

Grundsätzlich läuft das System auf drei verschiedenen Computer-Typen:

- Die Spiel-Logik ('Game') sowie der Hand-Tracking-Service soll auf einem normalen Personal-Computer/Laptop (Windows 11) laufen. Für das Handtracking ist eine (USB-)Kamera mit dem Computer verbunden. Das Game (bzw. das Spielfeld) wird auf dem Computer gerendert und mittels angeschlossenen Beamer auf einen Tisch projiziert.
- Die Smartphone- und Tablet-App, die unter anderem die Handkarten den Spielern anzeigt, soll auf Windows 11-Tablets und Android-Smartphones und -Tablets laufen. Um einen QR-Code scannen zu können, wird von den Geräten eine festverbaute und von Android erkannte Kamera vorausgesetzt. Für das Windows-Tablet wird dies nicht nötig sein, weil dort wahrscheinlich auf diese Funktion verzichtet wird (QR-Bibliotheken für Flutter unterstützen nur Android und iOS).
- Die Server werden voraussichtlich in verschiedenen Clouds sein.

4.5.2.2 Software

Bei den einzelnen Komponenten kommen folgende Software zum Einsatz:

KOMPONENTE	HARDWARE	SOFTWARE
GAME	Personal-Computer	Das Spiel soll mit Unity und der Programmiersprache C# entwickelt werden.
FINGER-TRACKING-SERVICE	Personal-Computer	Um das Spiel mit Handbewegungen auf dem Tisch steuern zu können, sollen die Hände bzw. die Finger mit einer KI-

		<p>basierten Bilderkennung erkannt werden. Aufgrund Erfahrungen wird dies mit der open-source und cross-platform Machine-Learning-Lösung MediaPipe von Google [1] versucht. MediaPipe unterstützt verschiedene Programmiersprachen. Für dieses Projekt kämen nur Python oder C++ in Frage. Aufgrund von mehr Hilfestellung wird für dieses Projekt dieser Service wahrscheinlich in Python geschrieben.</p>
APP	Smartphone/Tablet	<p>Auf Grund meiner eigenen Erfahrung kämen zwei verschiedene Frameworks für die App in Frage: Flutter (mit der Programmiersprache Dart) oder Unity (mit C#). Vor dem Start soll kurz evaluiert werden, welches Framework sich besser für die App eignet.</p>
GAME SERVICES	Cloud	<p>Die Game Services beinhalten Dienste, die für die Spiellogik relevant sind, und nicht bereits durch die Game States (DB) oder den Authentication-Service abgedeckt sind. Aus Erfahrung wären hier folgende Server möglich: NodeJS, ASP.net (C#), oder Flask (Python).</p> <p>Grundsätzlich muss dieser Service folgendes unterstützen:</p> <ul style="list-style-type: none"> - Streaming oder Polling des Kamera-Bildes auf die App (optional) - Der Service soll dem Gerät Update schicken können. Daher soll eine bidirektionale Verbindung möglich sein (z.B. WebSockets).
GAME STATES	Cloud (DB)	<p>Der Spielstatus soll in einer Datenbank zwischengespeichert werden, damit Spiel und App darauf Zugriff haben. Wahrscheinlich wird diese Datenbank eine NoSQL Firebase Realtime DB (dokument-orientiert) werden, was sich aber noch ändern darf.</p>
AUTHENTICATION	Cloud	<p>Alle Daten von einem bestimmten Spiel soll nur von Berechtigten (also Mitspieler oder dem Computer für den Spieltisch) gelesen oder bearbeitet werden können. Hierfür wird ein Token-basierter Dienst verwendet. Wahrscheinlich kommt Firebase Authentication zum</p>

Einsatz, weil ich damit bereits Erfahrungen machen konnte.

4.5.2.3 Schnittstellen

Serverseitige Daten

Folgende Daten werden von den Benutzern langfristig auf einem Server gespeichert:

4.5.3

ID	Dauer	Bezeichnung	Konkrete Daten
DA-L-01	langfristig	Login-Informationen	<ul style="list-style-type: none"> ▪ Benutzername/-ID ▪ Passwort (verschlüsselt) oder Token

Kurzfristig – also während des Spiels – müssen sicher folgende Daten auf einem Server zwischengespeichert werden:

ID	Dauer	Bezeichnung	Konkrete Daten
DA-02	kurzfristig	Spieler in einem Spiel	<ul style="list-style-type: none"> ▪ ID des Benutzers ▪ ID der Spiel-Instanz
DA-03	kurzfristig	Karten	<ul style="list-style-type: none"> ▪ Alle Rollen aller Spieler ▪ Karten jedes Spielers auf der Hand und vor sich ▪ Letzte Karte(n) auf dem Wegwerf-Stapel ▪ Lebenspunkte aller Spieler
DA-04	kurzfristig	Aktueller Spielmodus	<ul style="list-style-type: none"> ▪ Welche Spielphase gerade läuft (also z.B. Anmeldephase für Smartphone, Rollenverteilung oder reguläres rundenbasiertes Spiel) ▪ Welcher Spieler an der Reihe ist ▪ Ob gerade ein spezieller Modus (wie «Duell» oder «Indianen») gestartet wurde und wer bereits wie reagiert hat. ▪ Welche(r) Spieler involviert sind
DA-05	kurzfristig	Kamera-Bild	<ul style="list-style-type: none"> ▪ Bild der Kamera
DA-06	kurzfristig	Erkannte Finger	<ul style="list-style-type: none"> ▪ Erkannte Finger für Anzeige innerhalb der App

4.6 Einschränkungen/Randbedingungen/Abgrenzung

Abgrenzung

Diese Arbeit soll primär einen Prototypen für ein Augmented Card Game als Endprodukt liefern. Es soll lediglich das Spiel «Bang!» unterstützen, andere Spiele sind im Rahmen dieser Arbeit nicht geplant.

Einschränkungen

- 4.6.1 - Aus Platzgründen und wegen dem Fingertracking können maximal 4 Personen physisch am Spieltisch spielen. Weitere Spieler können nicht am Spieltisch sitzen.
- Es können maximal 7 Spieler mitspielen (Limitation des Spieles «Bang!»)
- 4.6.2 - Die optionale QR-Code-Scannen-Funktion wird unter Windows wahrscheinlich nicht implementiert, weil für Flutter diese Funktion nur unter Android und iOS verfügbar ist.

5 Spielfluss

5.1 Fluss zwischen den einzelnen Screens (generell)

Im folgenden Flussdiagramm sind die notwendigen Screens des Games (ohne App) als abgerundetes Viereck ersichtlich:

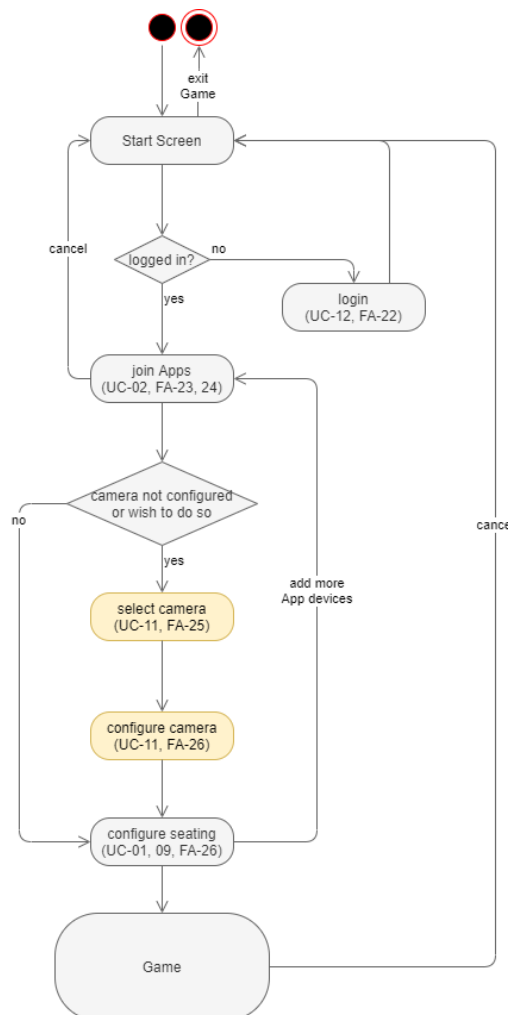


Abbildung 13: Fluss der einzelnen Screens (ohne spezifisches Spiel)

Das Spiel (welches auf dem Computer läuft, welcher mit dem Beamer verbunden ist) startet immer im Start Screen. Sollte der Computer nicht beim Server eingeloggt sein, so erscheint eine Aufforderung zum Einloggen oder Registrieren.

Anschliessend erscheint ein Screen, der es erlaubt, Smartphones und Tablets via App sich diesem Spieltisch anzuschliessen. Dies geschieht, indem in der App ein Code eingegeben/eingescannt wird, der vom Computer stammt. Die erste App ist der Admin.

Wenn mindestens eine App mit dem Game verbunden ist, kann fortgeführt werden. Anschliessend wird ein Setup durchlaufen, um die Kamera zuerst auszuwählen und anschliessend zu konfigurieren. Mit Konfigurieren ist gemeint, dass eingestellt werden kann, welcher Ausschnitt des Kamerabildes der Spielfläche entspricht. Diese Konfigurationen werden in der App vorgenommen.

Anschliessend könnte das Spiel gestartet werden. Sind noch zu wenige Spieler anwesend, müssen sich noch welche anmelden mit der App. Wenn alle Spieler in der App bekannt gegeben haben, dass sie bereit sind, wird das Spiel gestartet.

5.2 Fluss zwischen den einzelnen Screens während des Spieles

Nachdem das Game gestartet wurde, wird folgender Spielfluss durchlaufen. Die einzelnen abgerundeten Rechtecke entsprechen verschiedene Screens am Computer (also auf dem Spieltisch):

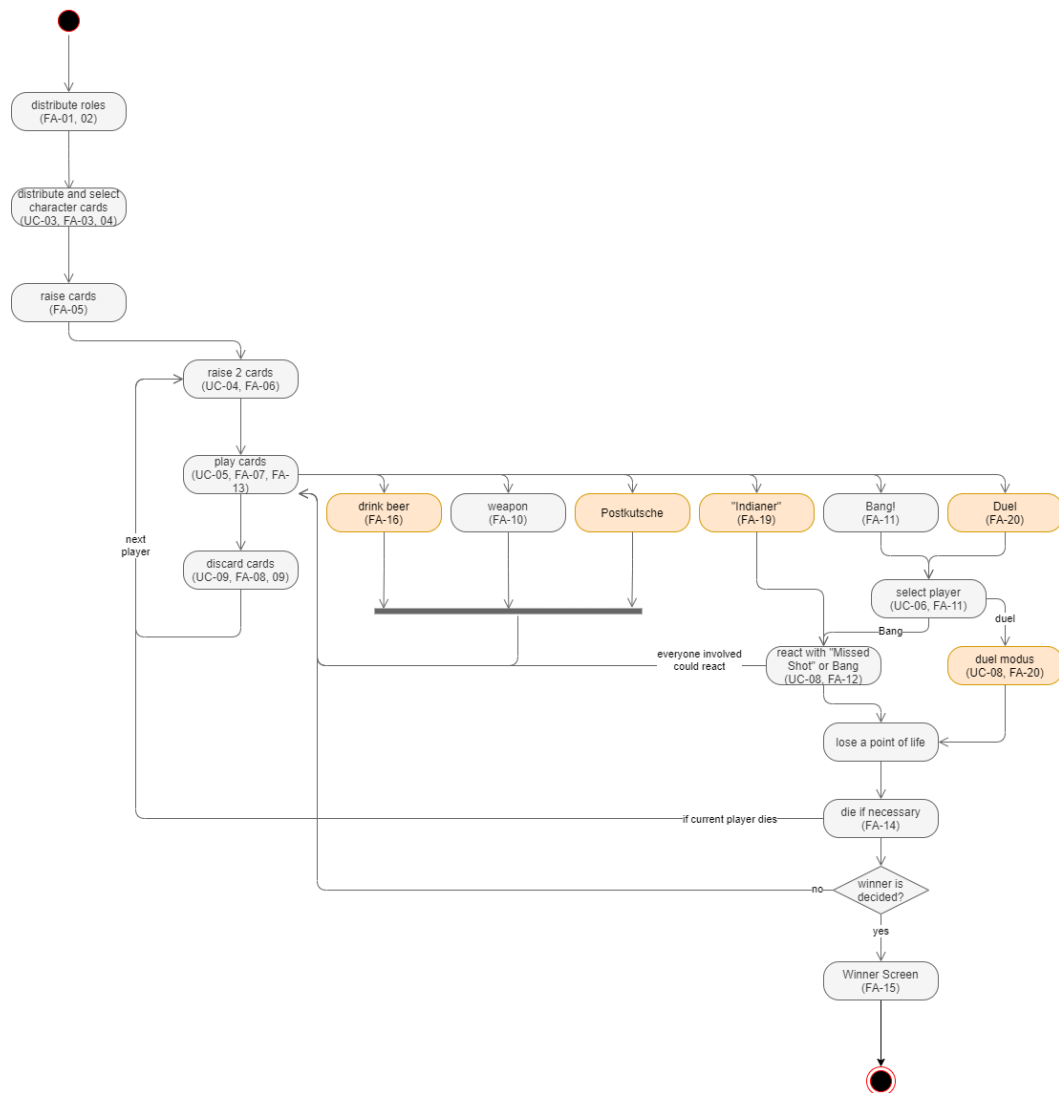


Abbildung 14: Die notwendigen Screens für die minimale Version des Bang!-Spieles

1. Zuerst werden die Rollen verteilt und in der App angezeigt. Der Sherif wird offen auf dem Spieltisch angezeigt.
2. Anschliessend werden allen Spielern zwei zufällige Charakterkarten ausgespielt, wobei nur die erste sichtbar ist. Alle Spieler können in der App auswählen, ob sie die aufgedeckte Charakterkarte haben möchten oder die andere. Möchte der Spieler die zweitere, so wird diese aufgedeckt und die nicht gewollte Karte wird weggeworfen. Die gewählte Charakterkarte ist für alle sichtbar. Entsprechend den Anzahl Maximal-Lebenspunkte werden die Lebenspunkte verteilt, wobei der Sherif ein Maximal-Lebenspunkt mehr erhält.
3. Automatisch werden allen Spielern so viele Handkarten verteilt, wie sie Lebenspunkte besitzen. Die Lebenspunkte werden bei allen Spielern angezeigt auf dem Spieltisch.
4. Anschliessend beginnt der Spieler, der an der Reihe ist, seinen Spielzug, indem er/sie zwei Karten vom Stapel zieht. Dies geschieht, in dem er/sie auf den Stapel klickt (in der App) oder mit der Hand den Stapel berührt (auf dem Spieltisch).
5. Anschliessend kann der Spieler in der App eine Karte auswählen, die er/sie spielen möchte. Diese gespielte Karte wird auch kurz auf dem Spieltisch angezeigt.

Danach wird die aktuelle Karte ausgeführt und evtl. kartenspezifische Screens angezeigt.

- a. Bei einem «Bang» oder «Duell» wird zuerst ein Screen angezeigt, der es erlaubt, einen anderen Spieler auszuwählen. In der App kann man einen Spieler anklicken. Auf dem Spieltisch genügt es, wenn man die Hand genügend lange vor einem Spieler hält. Visuell soll dann ein Feld vor dem ausgewählten Spieler aufleuchten, um zu zeigen, dass gerade dieser Spieler ausgewählt wird. Diese Aktion soll in der App bestätigt oder abgelehnt werden können.
 - b. Wurde eine «Duell»-Karte gespielt, so wird ein entsprechender Screen angezeigt, der abwechselungsweise den angegriffenen Spieler und den Spieler, der an der Reihe ist, auffordert, eine «Bang»-Karte zu spielen. Auf dem Tablet sind bei beiden Spielern die Handkarten zu sehen, wobei alle nicht «Bang»-Karten deaktiviert sind. Diese Spieler können dann entweder eine «Bang»-Karte spielen oder «aufgeben». Weil es in diesem Modus einen Verlierer geben muss, wird ein Spieler einen Lebenspunkt verlieren. Hierfür wird zu einem entsprechenden Screen gewechselt, in der das angezeigt wird.
 - c. Bei einem «Bang» (nach Spielerauswahl) oder «Indianen» wird ein Screen gezeigt, der Anzeigt, welcher Angriff ausgeführt wurde. In der App werden bei allen beteiligten Spielern alle nicht «Bang»- bzw. «Fehlschuss»-Karten deaktiviert. Wie beim «Duell» gibt es auch die Möglichkeit zu passen in der App. Bei denjenigen Spielern, die getroffen wurden, wird ein Lebenspunktabzug angezeigt. Dann kann der Spieler, der an der Reihe ist, weitere Karten spielen.
6. Am Ende des Spielzuges darf der Spieler maximal so viele Karten auf der Hand haben, wie er/sie Lebenspunkte besitzt. Hierfür kann er/sie in der App Karten auswählen, die er/sie wegwerfen möchte. Die entsprechenden Karten landen auf dem Tisch auf den Wegwerf-Stapel.
 7. Wird ein Spieler eliminiert, so wird rasch ein Screen mit einer entsprechenden Meldung angezeigt und anschliessen steht beim eliminierten Spieler ein entsprechender Text.
 8. Sobald das Spiel entschieden ist, wird zu einem entsprechenden Screen gewechselt. In diesem Screen wird bekanntgegeben, wer gewonnen hat. Die entsprechenden Spieler werden auf dem Spieltisch aufgeleuchtet. In der App erscheint auch eine entsprechende Meldung.

6 Abnahmetests

Die spezifischen Tests sind noch nicht Teil des Pflichtenheftes. Vor der Abgabe des Systems wird mindestens 1 Testfall pro Use Case definiert.

Die Tests sollen mindestens folgende Beschreibungen enthalten:

- Vor- und Nachbedingung
- Das erwartete Resultat und das zu vermeidende Resultat
- Ablauf des Tests

Die App soll auf Windows sowie auf einem Android-Smartphone und -Tablet getestet werden.

6.1 Test der nichtfunktionalen Anforderungen

Um herauszufinden, ob die App sowie der Spieltisch einfach zu bedienen ist, soll ein Test mit mindestens 2 Freiwilligen gemacht werden, um zu überprüfen, ob die Anwendungen selbsterklärend und einfach zu bedienen sind. Der genaue Testfall hierzu wird später definiert.

7 Projektmanagement

7.1 Projektorganisation

Für die Thesis kommt keine spezielle Projektmanagement-Methode wie Kanban oder Scrum zum Einsatz. Am Anfang der Thesis wird ein Pflichtenheft definiert und die entsprechenden Features werden dann Woche für Woche implementiert. In jeder Woche ist ein kurzes Austausch-Meeting zwischen dem Betreuer und dem Studierenden geplant, wo die aktuelle und folgende Arbeit besprochen wird.

7.2 Stakeholder und deren Aufgaben

Prof. Urs Künzler (Betreuer)	Er ist die Haupt-Ansprechsperson für den Studierenden und bewertet die Bachelor-Thesis.
Dr. Federico Flueckiger (Experte)	Neben dem Betreuer bewertet auch der Experte die Endarbeit.
Samuel Grimm (Studierender)	Ist verantwortlich für das Projektmanagement der Thesis und für das Umsetzen des Projektes.

7.3 Meilensteine

Die Meilensteine bestehen aus Paketen von Anforderungen. Folgende stehen zur Diskussion:

Name des Meilensteins	Erklärung	Anforderungen
Login-Prozess	Im Game für den Spieltisch kann sich der Benutzer anmelden und anschliessend können sich App-Benutzer damit verbinden. Das Game zeigt die verbundenen Benutzer an.	<ul style="list-style-type: none"> FA-22 login/register playing table on the server FA-23 join game via app FA-24 enter user nickname
Automatische Aktionen beim Spielstart	Die automatischen Aktionen am Spielstart werden automatisch durchlaufen und angezeigt.	<ul style="list-style-type: none"> FA-01 distribute roles FA-02 show who's the sheriff FA-03 distribute and select character card FA-04 receive life points in the beginning of the game

Grundlegende Funktionen bei einem Spielzug	Die Spieler können ganz normal einen Spielzug mit Karten legen usw. ausführen. Die Karten haben aber noch keine Wirkung.	<ul style="list-style-type: none"> ▪ FA-05 automatically raise cards in the beginning of the game ▪ FA-06 raise cards in the beginning of the player's move ▪ FA-07 select and play hand cards ▪ FA-08 discard hand cards ▪ FA-09 max amount of hand cards ▪ FA-13 automatically discard played cards
Die wichtigsten Karten	Die wichtigsten Karten aus dem Spiel können gespielt werden.	<ul style="list-style-type: none"> ▪ FA-10 play weapons ▪ FA-11 play «Bang!» ▪ FA-12 react with «Fehlschuss» on «Bang» or «Catling»
Möglichkeit zum Gewinnen	Es ist möglich zu sterben und zu gewinnen.	<ul style="list-style-type: none"> ▪ FA-14 die ▪ FA-15 check if and who has won the game
Konfigurierbares Fingertracking	Das Fingertracking kann mittels App konfiguriert werden.	<ul style="list-style-type: none"> ▪ FA-25 select camera for finger tracking ▪ FA-26 crop the playing field for more accurate finger tracking position ▪ FA-27 define seating position
Erweitertes Gameplay	Weitere Karten können gespielt werden.	<ul style="list-style-type: none"> ▪ FA-16 play «Been» ▪ FA-17 play «Catling» ▪ FA-18 play «Postkutsche» and raise two cards ▪ FA-19 play «Indianen» ▪ FA-20 play «Duel»

7.4 Gantt-Diagramm

7.5 Ausblick

Was sind die nächsten Schritte für die Bachelor-Thesis

8 Anhang

9 Index

No index entries found.

10 Literaturverzeichnis

[1] Google LLC, „MediaPipe,“ Google LLC, 2020. [Online]. Available: <https://mediapipe.dev/>. [Zugriff am 13 10 2021].

11 Abbildungsverzeichnis

Abbildung 1: Rollen des Spieles «Bang!».	4
Abbildung 2: Charakterkarten, nicht vollständig.	6
Abbildung 3: Waffenkarten.....	7
Abbildung 4: Die wichtigsten Aktionskarten des Spieles.....	7
Abbildung 5: Zusammenfassung der Symbole auf den Karten	8
Abbildung 6: Karten, mit welchen andere Lebenspunkte verlieren oder bekommen.	8
Abbildung 7: Karten, die bewirken, dass Karten gezogen werden können.	9
Abbildung 8: Karten, die eine Karte einem anderen Spieler entfernen.....	9
Abbildung 9: weitere blaue Karten, die vor Spieler gespielt werden.	10
Abbildung 10: Hardware-Ansicht des Systems	11
Abbildung 11: Software-Architektur des Systems	12
Abbildung 12: Anwendungsfälle des Systems	14
Abbildung 13: Fluss der einzelnen Screens (ohne spezifisches Spiel)	24
Abbildung 14: Die notwendigen Screens für die minimale Version des Bang!-Spieles	26