IPA Bericht

Multiplayer Tic-Tac-Toe

Autor: Florin Barbisch

Datum: Freitag, 22. Januar 2021

Projekt: Multiplayer Tic-Tac-Toe

Projektstart: Mittwoch, 09. Juni 2020

Projektende: Freitag, 22. Januar 2021

Auftraggeber: Daniel Mosimann

# Inhaltsverzeichnis

Inhaltsverzeichnis 2

Abbildungsverzeichnis 5

Tabellenverzeichnis 6

1 Einleitung 6

1.1 Inhalt und Zweck des Dokuments 6

1.2 Abkürzungen und Definitionen 7

Teil 1: Ablauf Organisation und Umfeld 7

2 Aufgabestellung 7

2.1 Ausgangslage 7

2.2 Detaillierte Aufgabenstellung 7

2.3 Mittel und Methoden 8

2.4 Vorkennnisse 8

2.5 Vorarbeiten 8

2.6 Neue Lerninhalte 8

2.7 Arbeiten in den letzten 6 Monaten 8

3 Projektmethodik 8

3.1 Projektmethode 8

3.2 Szenario 9

3.3 Phasen 9

3.4 Module 10

3.5 Aufgaben 10

3.6 Projektorganisation 11

3.6.1 Projektorganisation 11

3.6.2 Projektrollen 11

4 Zeitplanung 11

4.1 Zeitplan 11

5 Organisation der IPA 12

5.1 Arbeitsplatz 12

5.2 Datensicherung 12

5.3 Wiederherstellung 13

6 Firmenstandards 13

7 Arbeitsjournal 14

7.1 KW 24: 08.06.2020 14

7.2 KW 25: 15.06.2020 15

7.3 KW 26: 22.06.2020 15

7.4 KW 27: 29.06.2020 16

7.5 KW 33: 10.08.2020 16

7.6 KW 34: 17.08.2020 16

7.7 KW 35: 24.08.2020 17

7.8 KW 36: 31.08.2020 17

7.9 KW 37: 07.09.2020 18

7.10 KW 38: 14.09.2020 18

7.11 KW 39: 21.09.2020 19

7.12 KW 43: 19.10.2020 20

7.13 KW 44: 26.10.2020 20

7.14 KW 1 04.01.2021 21

7.15 KW 2 11.01.2021 22

7.16 KW 3 18.01.2021 23

Teil 2: Projektdokumentation 23

8 Kurzfassung 23

9 Initialisierung 23

9.1 Analyse 23

9.1.1 Studie; IST-Zustand 23

9.1.2 Projektziele 27

9.1.3 Anforderungen 27

9.2 Varianten 27

9.2.1 Variante 1 27

9.2.2 Variante 2 27

9.2.3 Entscheidungsmatrix 28

9.2.4 Gewählte Variante 28

10 Konzept 28

10.1 Architekturdiagramm 28

10.2 Use Cases 29

10.2.1 Einzelspieler 29

10.2.1.1 UC01 29

10.2.1.2 UC02 30

10.2.1.3 UC03 30

10.2.1.4 UC04 30

10.2.1.5 UC05 31

10.2.2 Mehrspieler 31

10.2.2.1 UC06 32

10.2.2.2 UC07 32

10.2.2.3 UC08 32

10.2.2.4 UC09 33

10.2.2.5 UC10 33

10.2.3 Login 34

10.2.3.1 UC11 34

10.2.3.2 UC12 34

10.2.3.3 UC13 35

10.3 UI-Mockups 35

10.3.1 Allgemein 35

10.3.1.1 Startseite ausgeloggt 35

10.3.1.2 Startseite eingeloggt 36

10.3.1.3 Sign up 36

10.3.1.4 Sign in 37

10.3.1 Einzelspieler 37

10.3.1.1 Einzelspielerspiel erstellen 37

10.3.1.2 Einzelspielerspiel spielen 37

10.3.1.3 Aktuelle Einzelspielerspiele 38

10.3.1.4 Einzelspielerspiel Ergebnis 38

10.3.1.5 Vergangene Einzelspielerspiele 39

10.3.1 Mehrspieler 39

10.3.1.1 Mehrspielerspiel erstellen 39

10.3.1.2 Mehrspielerspiel Link-Einladung 39

10.3.1.3 Mehrspielerspiel spielen 40

10.3.1.4 Aktuelle Mehrspielerspielspiele 40

10.3.1.5 Mehrspielerspiel Ergebnis 41

10.3.1.6 Vergangene Mehrspielerspiele 41

10.4 Testkonzept 42

10.4.1 Während dem Entwickeln 42

10.4.2 Testziele 42

10.4.3 Teststrategie 42

10.4.4 Aufbau eines Tests 43

10.4.5 Testrahmen 44

10.4.6 Testinfrastruktur 44

10.4.7 Testfälle 45

10.4.7.1 Testfall T01 45

10.4.7.2 Testfall T02 45

10.4.7.3 Testfall T03 46

10.4.7.4 Testfall T04 46

10.4.7.5 Testfall T05 47

10.4.7.6 Testfall T06 48

10.4.7.7 Testfall T07 49

10.4.7.8 Testfall T08 49

10.4.7.9 Testfall T09 50

10.4.7.10 Testfall T10 50

10.4.7.11 Testfall T11 51

10.5 Informationssicherheit und Datenschutz (ISDS) 52

11 Realisierung 53

11.1 System realisieren 53

11.2 Testprotokoll 53

11.2.1 Testfall T-001 53

11.2.2 Testfall T-002 53

12 Reflexion 54

13 Schlusswort 54

14 Literatur- und Quellverzeichnis 54

15 Glossar 56

16 Anhang 56

# Abbildungsverzeichnis

Abbildung 1: Hermes-Phasen 9

Abbildung 2: Hermes-Module 10

Abbildung 3: Arbeitsplatz 12

Abbildung 4: Daten wiederherstellen mit git checkout 13

Abbildung 5: Homescreen der alten Website 24

Abbildung 6: Während eines Einzelspiels auf der alten Webseite 24

Abbildung 7: Suche nach vergangenen Einzelspielen auf der alten Webseite 25

Abbildung 8: Suche nach Mehrspielerspielen auf der alten Webseite 26

Abbildung 9: Architekturdiagramm 28

Abbildung 10: Use-Cases Einzelspieler 29

Abbildung 11: Use-Cases Mehrspieler 31

Abbildung 12: Use-Cases Login 34

Abbildung 13: UI-Mockup Startseite ausgeloggt 35

Abbildung 14: UI-Mockup Startseite eingeloggt 36

Abbildung 15: UI-Mockup Sign up 36

Abbildung 16: UI-Mockup Sign in 37

Abbildung 17: UI-Mockup Einzelspielerspiel erstellen 37

Abbildung 18: UI-Mockup Einzelspielerspiel spielen 37

Abbildung 19: UI-Mockup Aktuelle Einzelspielerspiele 38

Abbildung 20: UI-Mockup Einzelspieler Ergebnis 38

Abbildung 21: UI-Mockup Vergangene Einzelspielerspiele 39

Abbildung 22: UI-Mockup Mehrspielerspiel erstellen 39

Abbildung 23: UI-Mockup Mehrspielerspiel Link-Einladung 39

Abbildung 24: UI-Mockup Mehrspielerspiel spielen 40

Abbildung 25: UI-Mockup Aktuelle Mehrspielerspiele 40

Abbildung 26: UI-Mockup Mehrspielerspiel Ergebnis 41

Abbildung 27: UI-Mockup Vergangene Mehrspielerspiele 41

Abbildung 28: Testmethode Blackbox-Testing 42

# Tabellenverzeichnis

Tabelle 1: Abkürzungen 7

Tabelle 2: Hermes-Aufgaben 10

Tabelle 3: Projektrollen 11

Tabelle 4: Projektziele 27

Tabelle 5: Entscheidungsmatrix 28

Tabelle 6: Muss-Funktionen 43

Tabelle 7: Vorlage Testfall 43

Tabelle 8: Vorlage Testfall T01 45

Tabelle 9: Vorlage Testfall T02 45

Tabelle 10: Vorlage Testfall T03 46

Tabelle 11: Vorlage Testfall T04 46

Tabelle 12: Vorlage Testfall T05 47

Tabelle 13: Vorlage Testfall T06 48

Tabelle 14: Vorlage Testfall T07 49

Tabelle 15: Vorlage Testfall T08 49

Tabelle 16: Vorlage Testfall T09 50

Tabelle 17: Vorlage Testfall T10 50

Tabelle 18: Vorlage Testfall T11 51

Tabelle 19: Glossar 56

# Einleitung

## Inhalt und Zweck des Dokuments

TODO

## Abkürzungen und Definitionen

| **Abkürzung** | **Bedeutung** |
| --- | --- |
| IPA | Individuelle praktische Arbeit |
| QV | Qualifikationsverfahren |
| HW | Hardware |
| SW | Software |
| TODO |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabelle : Abkürzungen

# Teil 1: Ablauf Organisation und Umfeld

# Aufgabestellung

## Ausgangslage

Ich habe im Modul 307 im ÜK eine Webseite erstellt, in der man Tic-Tac-Toe spielen kann. Diese Webseite verfügt über einen Einzelspieler- und einen lokalen Mehrspieler-Modus. Des Weiteren können auf der Webseite die vergangenen Spiele angesehen werden.

Die Webseite hat einige Nachteile:

* Der Mehrspieler-Modus funktioniert nicht auf zwei unterschiedlichen Rechnern.
* Das Backend und das Frontend ist nicht getrennt.
* Die Identität eines Benutzers kann in keiner Weise überprüft werden.
* Die Webseite ist zwar mit responsiven Technologien entwickelt worden, sieht aber auf dem Smartphone nicht gut aus.

Die Webseite eignet sich deshalb nicht, um zusammen aus weiter Distanz Tic-Tac-Toe gegeneinander zu spielen.

## Detaillierte Aufgabenstellung

Eine Webanwendung erstellen, in der ein Spieler Tic-Tac-Toe gegen den Computer oder einen anderen Spieler in ferner Distanz spielen kann.

Grundsätzliches:

* Der Spieler kann sich mit einem Nicknamen und einem Passwort registrieren und anmelden.
* Der Spieler angemeldete Spieler kann seine vergangenen Spiele ansehen.
* Der Spieler angemeldete Spieler kann seine offenen Mehrspieler-Spiele ansehen.

Einzelspieler-Modus:

* Der Benutzer kann zwischen mehreren Schwierigkeitsgraden auswählen.
* Der Computer führt seine Züge dem Schwierigkeitsgrad entsprechend gut aus.

Mehrspieler-Modus:

* Der Spieler A kann ein Spiel erstellen. Dazu kann er einen zweiten Spieler direkt einladen, einen Link an einen zweiten Spieler schicken oder mit einem unbekannten Spieler spielen.
* Die Spieler können, ohne die Seite neu zu laden, gegeneinander spielen.
* Ein Spieler kann das Spiel auch verlassen und später zurückkommen, um weiter zu spielen.
* Wenn das Spiel fertig ist, können sich die Spieler revanchieren. Dabei beginnt der Spieler der bei der letzten Runde als zweiter begonnen hat.

## Mittel und Methoden

* Projektmethodik Hermes
* vueJS (Frontend)
* REST mit nodeJS (Backend)
* MongoDB (Datenbankserver).

## Vorkennnisse

* Entwicklung einer Tic-Tac-Toe Webseite mit PHP und MySQL.
* Entwicklung von mehreren VueJS Applikationen.
* Entwicklung einer REST-API mit PHP.

## Vorarbeiten

Überprüfen der Entwicklungsumgebung auf dem persönlichen PC/Notebook. Git Repository einrichten.

## Neue Lerninhalte

REST-Backend mit NodeJS und MongoDB als Datenbank Server. Hermes als Projektmethodik.

## Arbeiten in den letzten 6 Monaten

Zwei VueJS Applikation (mit-)entwickelt, wobei eine über ein REST-Backend mit PHP verfügte. In diesen VueJS-Applikationen ging es mehrheitlich darum Daten zu verwalten/anzuzeigen.

# Projektmethodik

## Projektmethode

Als Projektmethode verwende ich «Hermes 5.1 IPA» [1], da dies von Daniel Mosimann vorgeschrieben wurde. Die Projektmethode «Hermes 5.1 IPA» ist eine stark vereinfachte Version für die Abschlussprüfung IPA ICT-Lernende Kanton Bern der offiziellen Form von «Hermes 5.1».

Deklaration der gewählten Projektmethodik. Angabe Quelle und Version. Kurzer Überblick.

Für den IPA-Test ist Hermes gesetzt! Deshalb auch die nachfolgenden Unterkapitel…

## Szenario

Da es sich bei meinem Projekt um eine Software handelt, welche ich selbst entwickle, habe ich das Szenario «IT-Individualanwendung» genommen.

## Phasen

Es gibt bei der IPA keine Phase Einführung, deshalb wurde diese hier ausgegraut dargestellt. Es ist auch unklar, ob diese Webseite jemals für die Öffentlichkeit verfügbar sein wird.

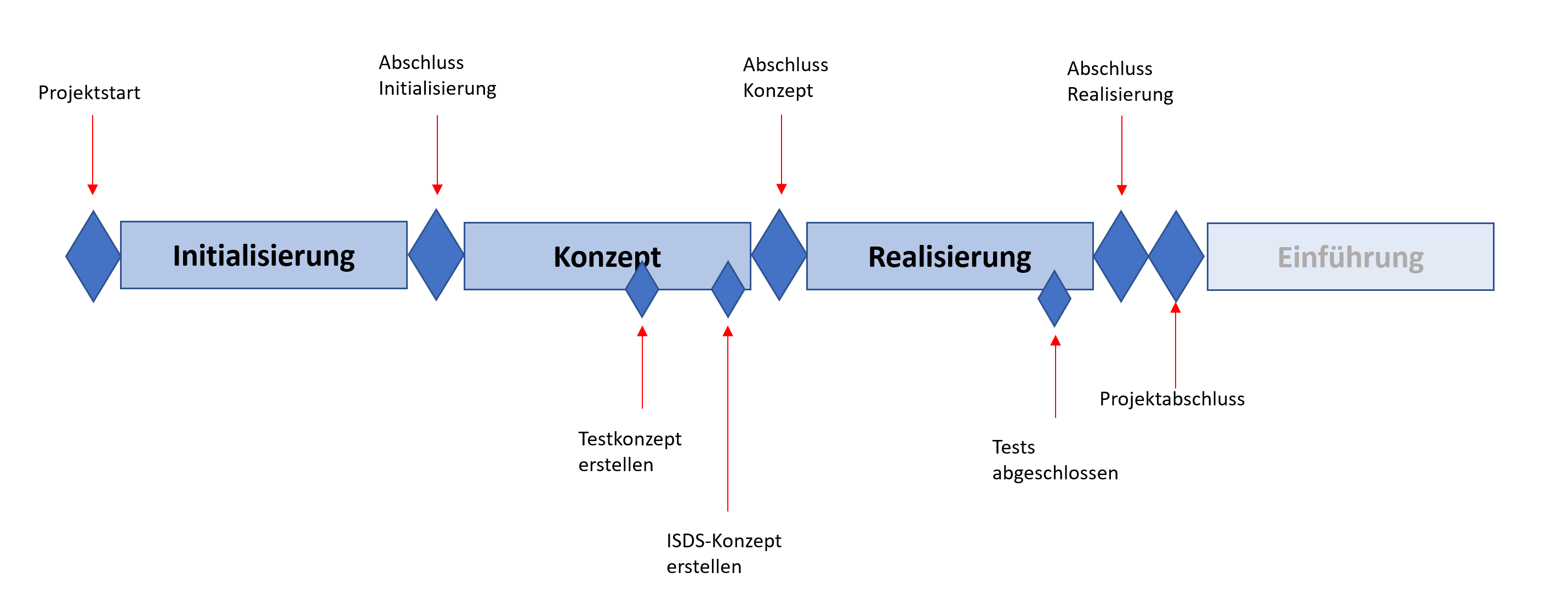


Abbildung : Hermes-Phasen

## Module

Auch hier fällt die Einführung wieder weg.

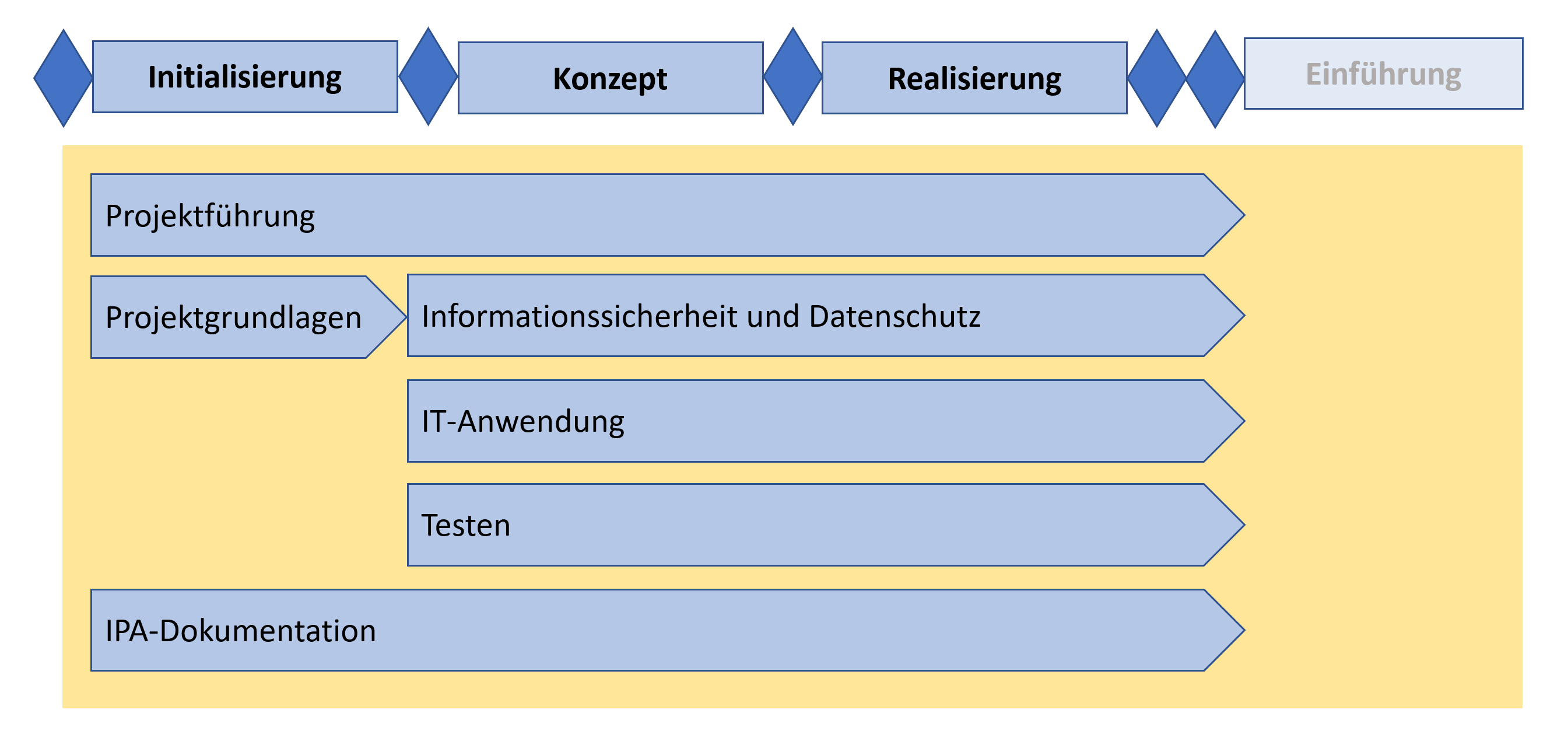


Abbildung : Hermes-Module

## Aufgaben

Auch hier fällt die Einführung wieder weg.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Initialisierung | Konzept | Realisierung |
| Projektführung | Projektauftrag erarbeiten | Projekt führen und kontrollieren  Projekt steuern  Probleme behandeln und Erfahrungen nutzen  Risiken managen  Qualitätssicherung führen | |
| Projekt-grundlagen | Dokumente Studieren |  |  |
| Informations-sicherheit und Datenschutz |  | ISDS-Konzept erstellen | ISDS-Konzept umsetzen |
| IT-Anwendung |  | Applikationskonzept erarbeiten | Applikation realisieren |
| Testen |  | Testkonzept erarbeiten | Testinfrastruktur realisieren  Tests durchführen |
| IPA-Dokumentation | Initialisierung dokumentieren | Konzept dokumentieren | Realisierung dokumentieren |

Tabelle : Hermes-Aufgaben

## Projektorganisation

### Projektorganisation

**Auftraggeber:**

Daniel Mosimann

**Projektleiter:**

Florin Barbisch

### Projektrollen

|  |  |
| --- | --- |
| **Auftraggeber:** | Daniel Mosimann, Lehrer |
| **Projektleiter:** | Florin Barbisch, Lernender |

Tabelle : Projektrollen

# Zeitplanung

## Zeitplan

TODO

Zeitplan mit SOLL-/IST-Vergleich. Es wurde eine absolute Zeitachse definiert (Datum). Die Zeitachse hat eine vernünftige Auflösung (2- oder 4-Stundenblöcke).

Siehe Muster-Zeitplan!

# Organisation der IPA

## Arbeitsplatz



Abbildung : Arbeitsplatz

Wie auf dem Bild zu sehen ist, verfüge ich über die nötige Infrastruktur, um diese IPA durchzuführen.

## Datensicherung

Alle Daten und Dokumente werden regelmässig auf GitHub hochgeladen: <https://github.com/florinbarbisch/tic-tac-toe-multiplayer>

Damit ist nicht nur eine hohe Verfügbarkeit garantiert sondern auch eine Versionierung sichergestellt.

## Wiederherstellung

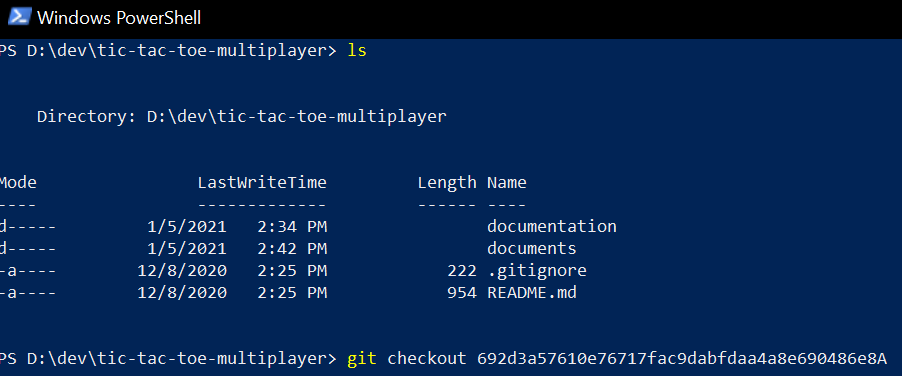


Abbildung : Daten wiederherstellen mit git checkout

Eine Wiederherstellung der Daten erfolgt mit Git über ein Check-out des jeweiligen Commits.

# Firmenstandards

Verwendete Vorlagen:

* Dokumentationsvorlage HERMES IPA von Daniel Mosimann
* Vorlage Zeitplan von Daniel Mosimann

Verwendete Arbeitsmethode: HERMES

Verwendete CASE-Tools:

* Git
  + GitKraken
  + GitHub
* Visual Studio Code
* StarUML
* bootstrapstudio.io

# Arbeitsjournal

## KW 24: 08.06.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Teams Anruf mit Daniel Mosimann | Initialisierung | 2 | 2 |
| IPA Berichtvorlage ausfüllen | Initialisierung | 1 | 1 |
| ***Total*** | | ***3*** | ***3*** |
| **Tagesablauf** | | | |
| Heute Morgen hatten wir einen Anruf mit Daniel Mosimann. Er hat das Modul 306 erklärt und uns den Projektauftrag für das Modul abgegeben. | | | |
| **Wissensbeschaffung** | | | |
| Microsoft Teams | | | |
| **Hilfestellungen** | | | |
| - | | | |
| **Reflexion** | | | |
| Was lief gut?  Das erste Ausfüllen des IPA-Berichts verlief gut.  Was lief weniger gut?  Die Teams Sitzung war lang, das hat viel Energie gekostet.  Meine Erkenntnisse von heute | | | |

## KW 25: 15.06.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Teams Anruf mit Daniel Mosimann | Initialisierung | 0.5 | 0.5 |
| Repository erstellen | Initialisierung | 1 | 1 |
| Themensuche / Gedanken zur Aufgabenstellung | Initialisierung | 1 | 1 |
| ***Total*** | | ***2.5*** | ***2.5*** |
| **Tagesablauf** | | | |
| Heute konnten wir allfällige Fragen an Daniel Mosimann stellen. Daniel Mosimann hat uns auch die Bewertungskriterien für diesen Auftrag erklärt/abgegeben. Danach habe ich das Projekt Repository erstellt und an Daniel Mosimann gesendet. Ich habe mir bereits Gedanken zur Thema/Produkt der Aufgabenstellung gemacht. | | | |
| **Wissensbeschaffung** | | | |
| Microsoft Teams | | | |
| **Hilfestellungen** | | | |
| Beispiel Aufgabenstellung | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich hatte keine Fragen und konnte das Repository erfolgreich erstellen. Ich habe mir erste Gedanken zur Aufgabenstellung gemacht.  Was lief weniger gut?  Ich konnte mich nicht direkt für eine Aufgabe entscheiden.  Meine Erkenntnisse von heute  Themensuche ist schwer. Nichtsdestotrotz will ich nicht die Standard Aufgabenstellung umsetzen. | | | |

## KW 26: 22.06.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Aufgabenstellung | Initialisierung | 4 | 4 |
| ***Total*** | | ***4*** | ***4*** |
| **Tagesablauf** | | | |
| Ich habe nur an der Aufgabenstellung gearbeitet und diese dann abgegeben. | | | |
| **Wissensbeschaffung** | | | |
| Eigene TicTacToe Applikation in PHP. | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich konnte die Aufgabenstellung gut lösen, da sie ein bestehendes Problem löst welches ich gut kenne.  Was lief weniger gut?  -  Meine Erkenntnisse von heute  Auch wenn man ein Problem schon kennt ist es nicht einfach eine Aufgabenstellung darüber zu schreiben. | | | |

## KW 27: 29.06.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| ***Total*** | | ***0*** | ***0*** |
| **Tagesablauf** | | | |
| Heute habe ich nichts gemacht, da ich letzte Woche viel Zeit in die Aufgabenstellung investiert habe. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Was lief weniger gut?  Meine Erkenntnisse von heute | | | |

## KW 33: 10.08.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Dokumente studieren | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Heute konnten wir uns nicht im Office 365 einloggen und nicht auf alle wichtigen Dokumente zugriefen. Ich konnte Schlussendlich die wichtigsten Dokumente auf dem Handy anschauen. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich konnte auf dem Handy die wichtigsten Dokumente studieren.  Was lief weniger gut?  Ich konnte mich nicht an der Schule anmelden aufgrund eines Netzwerkfehlers. Deshalb hatte ich kein Zugriff auf wichtige Informationen und der Tag verlief nicht wirklich produktiv.  Meine Erkenntnisse von heute  Wenn das Internet nicht richtig funktioniert wird das Arbeiten erschwert. | | | |

## KW 34: 17.08.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Alle referenzierten Dokumente in das Repository kopieren | Initialisierung | 0.5 | 0.5 |
| Dokumente studieren | Initialisierung | 1 | 1 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Heute habe ich die referenzierten Dokumente in das Repository kopiert. Des Weiteren habe ich die referenzierten Dokumente studiert. | | | |
| **Wissensbeschaffung** | | | |
| Referenzierten Dokumente aus OneNote | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Es ging alles gut  Was lief weniger gut?  Nichts.  Meine Erkenntnisse von heute  Viele Dokumente ohne grossen Überblick. | | | |

## KW 35: 24.08.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| ***Total*** | | ***0*** | ***0*** |
| **Tagesablauf** | | | |
| Heute hatte ich Rekrutierung. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Was lief weniger gut?  Meine Erkenntnisse von heute | | | |

## KW 36: 31.08.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Arbeitsjournal führen | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Heute habe ich das Arbeitsjournal nachgeführt. | | | |
| **Wissensbeschaffung** | | | |
| Teams Nachrichten, Email Verlauf, | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich konnte mich noch sehr gut daran erinnern was ich früher gemacht habe und das Dokumentieren.  Was lief weniger gut?  Meine Erkenntnisse von heute  Ich hätte früher anfangen sollen mit Arbeitsjournal nachführen. | | | |

## KW 37: 07.09.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Arbeitsjournal führen | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Heute habe ich das Arbeitsjournal nachgeführt. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich konnte mich noch sehr gut daran erinnern was ich früher gemacht habe und das Dokumentieren.  Was lief weniger gut?  Meine Erkenntnisse von heute  Ich hätte früher anfangen sollen mit Arbeitsjournal nachführen. | | | |

## KW 38: 14.09.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Erklärung Hermes durch Mosimann | Initialisierung | 0.5 | 0.5 |
| PC-Aufsetzen, Software installieren | Initialisierung | 0.0 | 0.5 |
| Arbeitsjournal führen | Initialisierung | 0.5 | 0.5 |
| Projektmethodik in der Dokumentation beschreiben | Initialisierung | 0.5 | 0.5 |
| ***Total*** | | ***1.5*** | ***2*** |
| **Tagesablauf** | | | |
| Am Anfang der Lektion hat Mosimann Hermes erklärt. Danach musste ich Software welche ich zum Arbeiten brauche neu installieren da die Schule alles gelöscht hat. Danach habe ich das Arbeitsjournal aktualisiert. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich habe Hermes besser verstanden. Und Hermes in meiner Dokumentation ergänzt  Was lief weniger gut?  Ich musste die Software welche ich zum Arbeiten brauche neuinstallieren.  Meine Erkenntnisse von heute  Hermes hat für die IPA keine Vorteile/Nachteile gegenüber IPERKA. | | | |

## KW 39: 21.09.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| PC-Aufsetzen, Software installieren | Initialisierung | 0.0 | 0.5 |
| Zeitplan Vorlage erstellen | Initialisierung | 1 | 1 |
| ***Total*** | | ***1*** | ***1.5*** |
| **Tagesablauf** | | | |
| Heute musste ich wieder die Software welche ich zum Arbeiten brauche neu installieren da die Schule wieder alles gelöscht hat. Danach habe ich eine Vorlage für den Zeitplan erstellt. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich bin mit der Vorlage für den Zeitplan fertig geworden.  Was lief weniger gut?  Ich musste die Software welche ich zum Arbeiten brauche schon wieder neuinstallieren.  Meine Erkenntnisse von heute  Es wäre vielleicht besser, wenn ich eine RDP-Verbindung zu meinem PC zu Hause mache, um dort zu arbeiten. Es braucht definitiv weniger Zeit einen VPN-Client zu installieren und eine RDP Verbindung aufzubauen als Passwortmanager und einen Gitclient zu installieren, dann den Git-Client einzurichten und am Ende noch das Repository zu klonen. | | | |

## KW 43: 19.10.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Überblick verschaffen über die Vorgaben/Hermes | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Heute habe ich mir einen Überblick über die Vorgaben und Hermes verschafft. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Meine Erkenntnisse von heute  Es gibt sehr viele Vorgaben und diese sind sehr unübersichtlich strukturiert. | | | |

## KW 44: 26.10.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Studieren der Vorgaben/Hermes | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Heute habe ich die Vorgaben der IPA und Hermes angeschaut. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Meine Erkenntnisse von heute  Gleiche Erkenntnis wie letzte Woche. | | | |

TODO maybe 45 bis 51 auch eintragen? lol

## KW 1 04.01.2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Projektplanung erstellen | Initialisierung | 3 | 3 |
| Zeitplan erstellen | Initialisierung | 3 | 3 |
| Projektorganisation definieren | Initialisierung | 1 | 1 |
| Situationsanalyse (IST-Situation) | Initialisierung | 1 | 1 |
| Projektziele definieren | Initialisierung | 1 | 1 |
| Aufgabe/Anforderungen verfeinern | Initialisierung | 1 | 1 |
| Lösungsvarianten/Entscheidungsmatrix | Initialisierung | 2 | 2 |
| Architekturdiagram | Konzept | 1 | 1 |
| Informationssicherheit und Datenschutz (ISDS) | Konzept | 1 | 1 |
| Use-Case Diagramm | Konzept | 4 | 1 |
| ***Total*** | | ***15*** | ***15*** |
| **Tagesablauf** | | | |
| Ich habe diese Woche die Projektplanung und Projektorganisation erstellt, damit sind die Kapitel 3, 5 und 6 fertig. Des Weiteren habe ich den Zeitplan erstellt. Beim Zeitplan habe ich mich entschieden Tageseinheiten zu nehmen, statt den ursprünglichen Wocheneinheiten, da ich mit dem Projekt erst diese Woche begonnen habe. Um die Situationsanalyse (IST-Situation) zu machen, musste ich zuerst das alte System wieder installieren was etwas Zeit brauchte aber dennoch im Rahmen meiner Schätzung lag. Die Projektziele zu definieren verlief hingegen ohne Hürden. Die Entscheidungsmatrix hat volle 2 Stunden gedauert, weil ich mich zuerst einlesen musste, was Socket-Communication ist. Mit zwei Tasks Architekturdiagram und ISDS konnte ich ohne weiter Probleme erledigen. Ich habe bereits mit den Use-Case Diagrammen angefangen, weil ich noch Zeit hatte. | | | |
| **Wissensbeschaffung** | | | |
| Alte Tic-Tac-Toe Anwendung. | | | |
| **Hilfestellungen** | | | |
| Vorlage Zeitplan von Daniel Mosimann. | | | |
| **Reflexion** | | | |
| Was lief gut?  Es lief mehrheitlich gut.  Was lief weniger gut?  Ich musste XAMPP wieder installieren, um die alte Webseite zu installieren, da XAMPP in einem unbrauchbaren Zustand war. Damit habe ich nicht gerechnet.  Meine Erkenntnisse von heute  Ich weiss nun an welchen Tagen ich wieviel arbeiten muss, um diese Arbeit abzuschliessen. | | | |

## KW 2 11.01.2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Use-Case Diagramme | Konzept | 3 | 2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| ***Total*** | | ***8*** | ***8*** |
| **Tagesablauf** | | | |
| Mongodb installieren,  repos aufsetzen  backend realworld template | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Was lief weniger gut?  Meine Erkenntnisse von heute | | | |

## KW 3 18.01.2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| ***Total*** | | ***8*** | ***8*** |
| **Tagesablauf** | | | |
|  | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Was lief weniger gut?  Meine Erkenntnisse von heute | | | |

# Teil 2: Projektdokumentation

# Kurzfassung

TODO: Zusammenfassung der IPA-Resultate auf max. einer A4 Seite (eine Art Management Summary  Zusammenfassung für die Geschäftsleitung). Die Kurzfassung enthält die Abschnitte: Kurze Ausgangssituation - Umsetzung - Ergebnis.

# Initialisierung

## Analyse

### Studie; IST-Zustand

In der Applikation kann man momentan Tic-Tac-Toe gegen den Computer oder gegen einen anderen Spieler spielen, dies ist leider nur auf dem gleichen Rechner möglich. Beim Singleplayer kann der Benutzer zwischen drei Schwierigkeitsgraden auswählen: EASY, MEDIUM, IMPOSSIBLE. Der Spieler gibt vor dem Spiel immer sein Name ein. Des Weiteren kann nach vergangenen Spielen gesucht werden (in Merhspielerspielen nach NameX und NameO und in Einzelspielerspielen nach Name und Schwierigkeitsgrad).

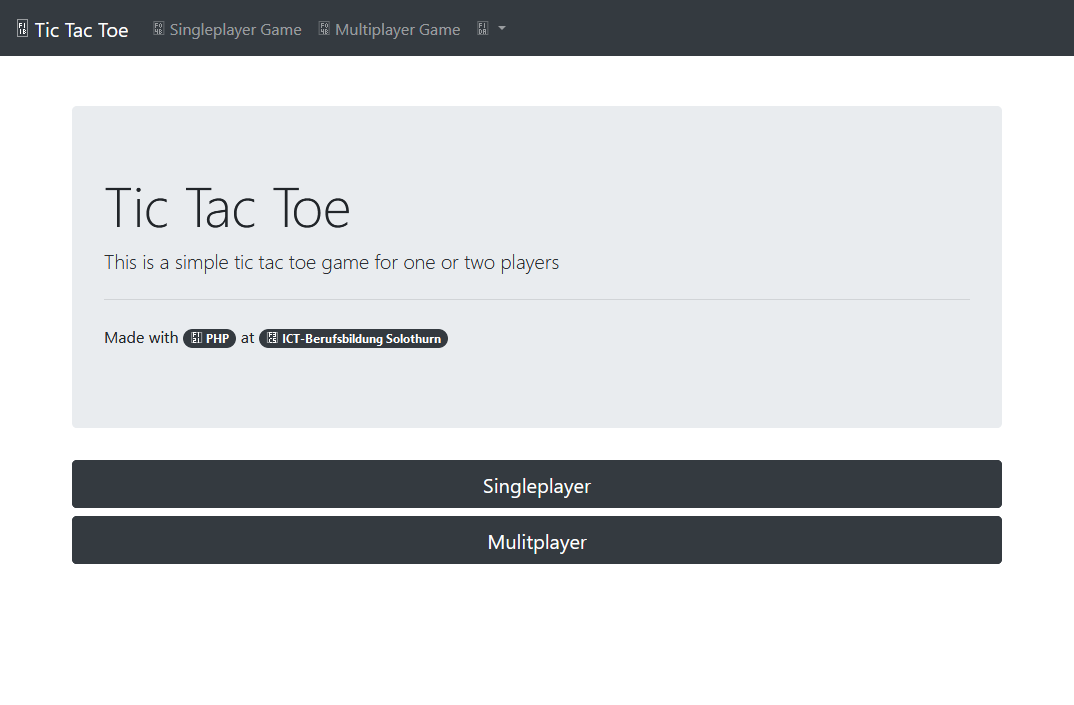


Abbildung : Homescreen der alten Website

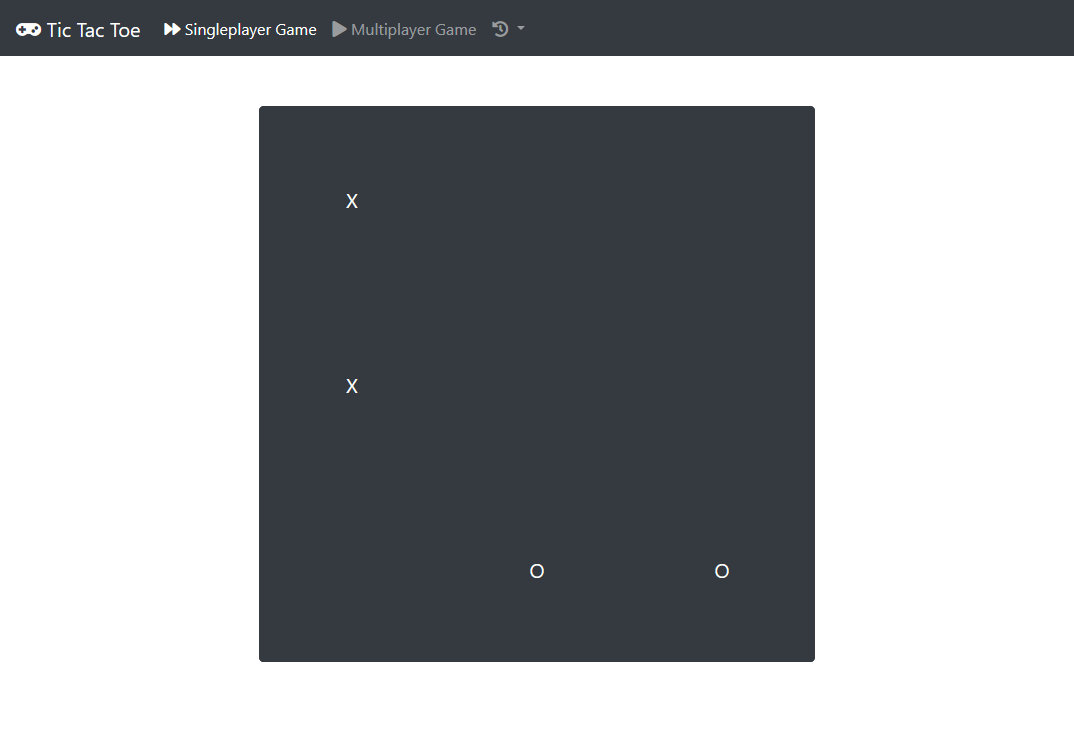


Abbildung : Während eines Einzelspiels auf der alten Webseite

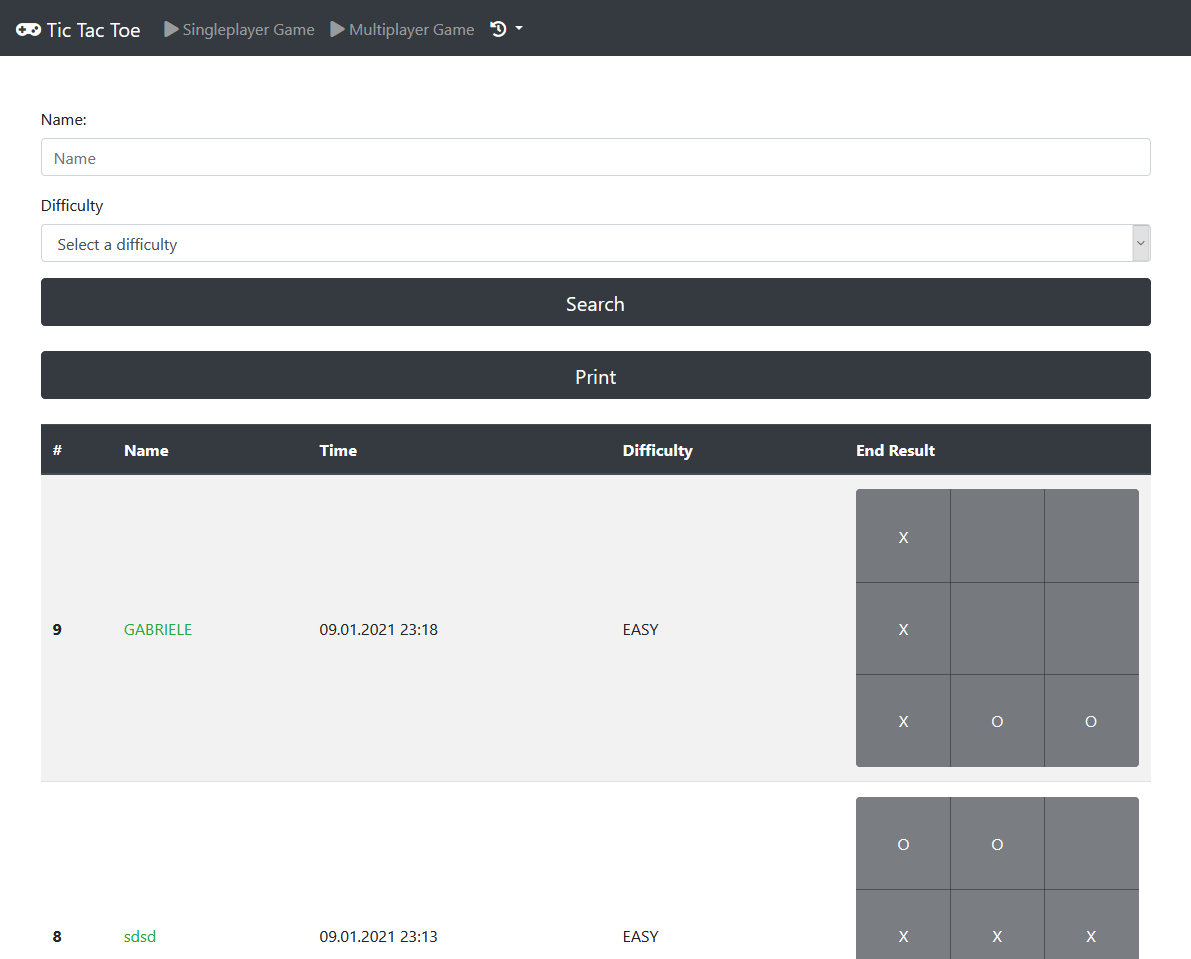


Abbildung : Suche nach vergangenen Einzelspielen auf der alten Webseite

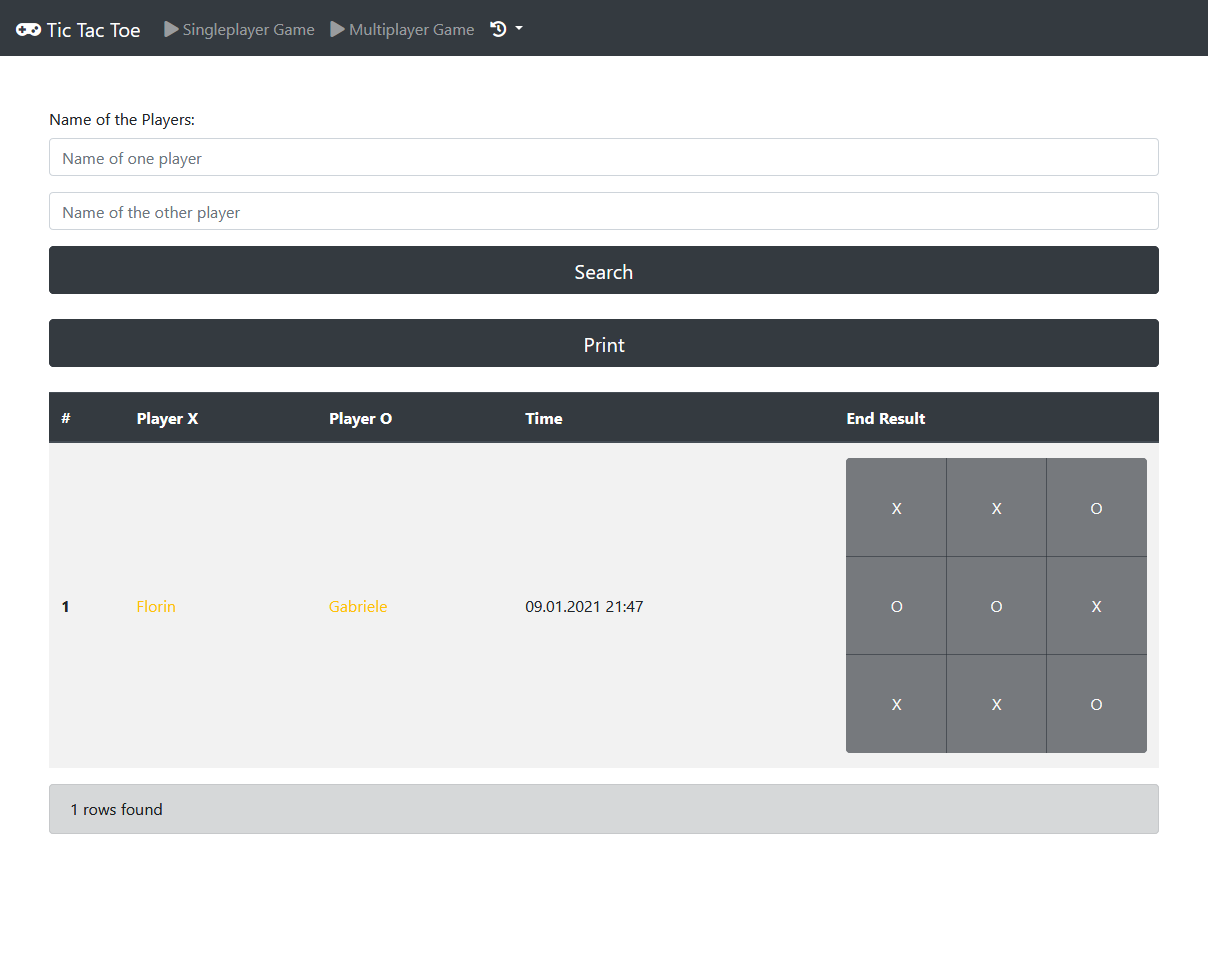


Abbildung : Suche nach Mehrspielerspielen auf der alten Webseite

### Projektziele

|  |  |
| --- | --- |
| Ziel | Beschreibung |
| Ablösen der alten Applikation | Die neue Applikation soll in etwa die gleichen, respektive mehr, Funktionalitäten haben wie die alte Applikation. Funktionen können weggelassen werden, wenn diese einen geringen Mehrwert bieten. |
| Frontend & Backend ist getrennt | Das Frontend wird vom Backend getrennt, sodass ohne Anpassungen am Backend ein neues Frontend entwickelt werden kann. |
| Mehrspieler auf 2 Rechnern | Der Mehrspielermodus muss nicht mit dem gleichen Rechner gespielt werde. Die Spieler können sich auf unterschiedlichen Orten der Welt befinden und dennoch miteinander spielen. |
| Intuitive und einfache Bedienung | Die Applikation soll so gestaltet werden, dass für den Benutzer wenig, bis keine Erklärung nötig ist. Des Weiteren soll der Benutzer nicht unnötige Klicks oder dergleichen machen müssen. |
| Spieler einladen ist einfach | Es ist einfach einen Spieler zu einem Mehrspielerspiel einzuladen. |

Tabelle : Projektziele

### Anforderungen

* Das Backend ist mit REST umgesetzt
* Das Frontend ist mit VueJS umgesetzt
* Als Datenbank wird MongoDB verwendet
* Die Benutzer können gegen einen Computer oder gegeneinander Tic-Tac-Toe spielen
* Frontend und Backend sind nicht von der genauen Implementation abhängig
* Einen Spieler einladen ist einfach
* Die Bedienung ist ergonomisch

## Varianten

Damit der Benutzer sofort mitbekommt, dass der andere Spieler einen Zug gespielt hat, muss eine Library verwendet werden welche dies ermöglicht (real-time, bidirektionale Kommunikation).

### Variante 1

Als erste Variante habe ich mir WebSocket angeschaut. WebSocket ist das Protokoll welche es erlaubt eine bidirektionale Verbindung zwischen dem Server und dem Client. Dies erfolgt ohne grossen Overhead (TODO https://de.wikipedia.org/wiki/WebSocket). Diese kann man über JavaScript sehr einfach ansteuern. Leider muss man das Backend aber selbst implementieren was als einen grossen Aufwand erscheint.

### Variante 2

Als zweite Variante habe ich Socket.io (TODO Quelle). Diese abstrahiert WebSocket und kommt gleich mit einem NodeJS Server, was für mein Projekt super passt.

### Entscheidungsmatrix

|  |  |  |
| --- | --- | --- |
| Kriterium | Bewerung (0-5) | |
| Websocket | Socket.io |
| Backend einfach zu implementieren | 1 | 4 |
| Frontend einfach zu implementieren | 4 | 5 |
| Total | **5** | **9** |

Tabelle : Entscheidungsmatrix

### Gewählte Variante

Socket.io schneidet deutlich besser ab, vor allem deshalb, weil ein Backend bereits implementiert ist und dies nur noch richtig angesteuert werden muss.

# Konzept

## Architekturdiagramm

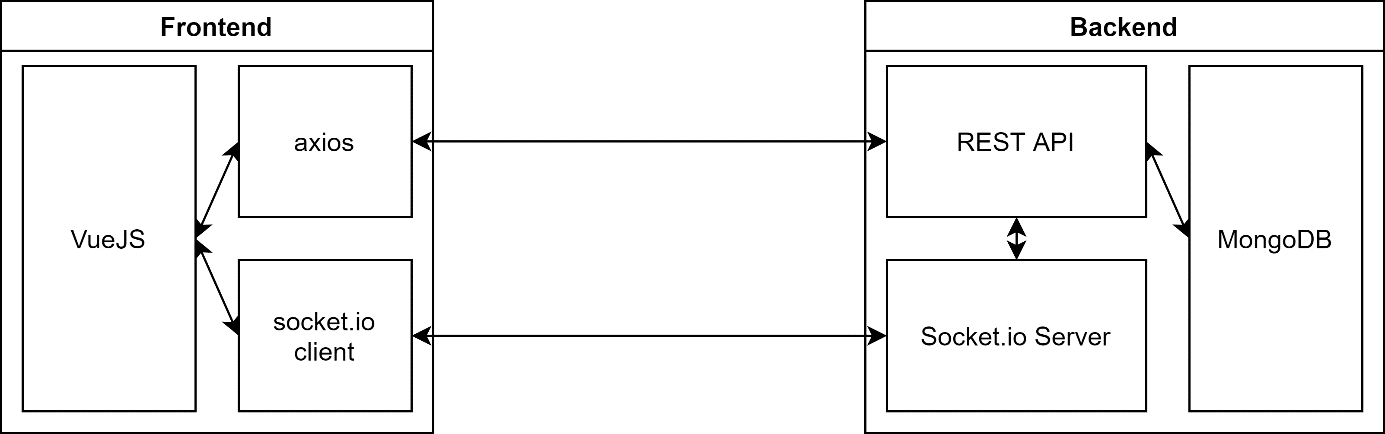


Abbildung : Architekturdiagramm

Ich habe mich bei der Software Architektur dazu entschieden, dass der Socket.io Server mit der REST API kommuniziert und nicht direkt mit der Datenbank. Das heisst, dass ein Spielzug der REST API gemeldet werden, und diese dann dies in der DB anpasst und dem socket.io Server mitteilt. Damit kann die Applikation theoretisch auch ohne Socket.io funktionieren.

## Use Cases

### Einzelspieler

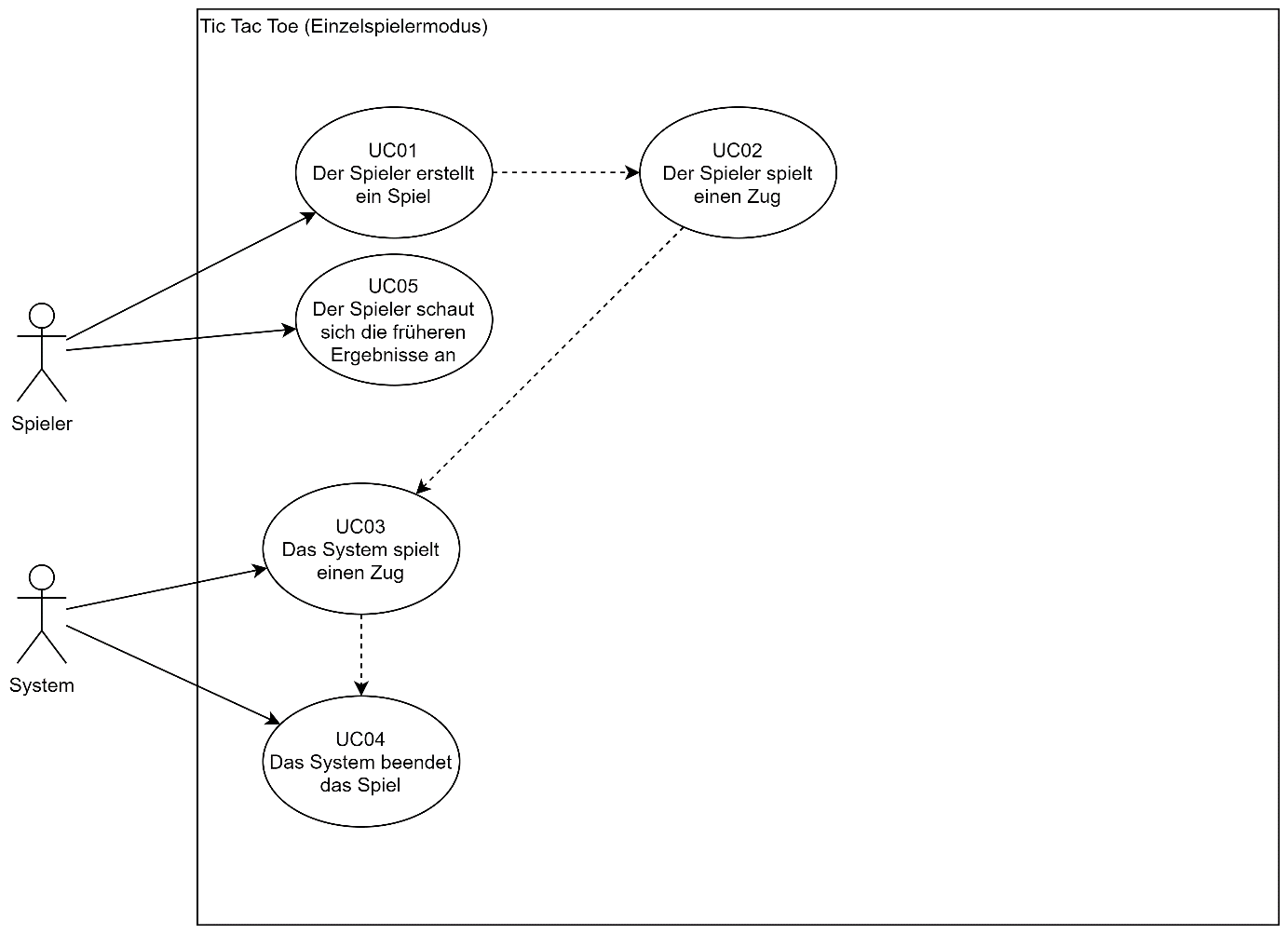


Abbildung : Use-Cases Einzelspieler

#### UC01

|  |  |
| --- | --- |
| Nummer: | UC01 |
| Name: | Der Spieler erstellt ein Spiel |
| Kurzbeschreibung: | - Der Spieler wählt einen Schwierigkeitsgrad aus  - Der Spieler startet das Spiel |
| Akteure: | Spieler |
| Auslöser: | Button: Einzelspielerspiel erstellen |
| Vorbedingung: | Der Spieler ist angemeldet |
| Nachbedingung: | Dem Spieler wird das Spiel angezeigt und er kann einen ersten Zug machen |
| Geschäftsregeln: | Alle Eingaben werden überprüft und validiert |
| Abgrenzung / Notizen: | Der Spieler ist das Kreuz |

TODO Tabelle 3: UC01

#### UC02

|  |  |
| --- | --- |
| Nummer: | UC02 |
| Name: | Der Spieler spielt ein Zug |
| Kurzbeschreibung: | Der Spieler setzt sein Kreuz auf ein leeres Feld |
| Akteure: | Spieler, System |
| Auslöser: | Starten des Spiels |
| Vorbedingung: | Der Spieler hat das Spiel gestartet |
| Nachbedingung: | Der Zug wird an den Server gesendet |
| Geschäftsregeln: | Es wird validiert, ob der Spieler überhaupt das Kreuz an diesem Ort setzen darf |
| Abgrenzung / Notizen: | - |

#### UC03

|  |  |
| --- | --- |
| Nummer: | UC03 |
| Name: | Das System spielt einen Zug |
| Kurzbeschreibung: | - Das System validiert den Zug des Spielers.  - Das System überprüft, ob das Spiel fertig ist.  - Das System führt selbst einen Zug aus.  - Das System überprüft, ob das Spiel fertig ist. |
| Akteure: | System |
| Auslöser: | Der Spieler hat einen Zug gespielt |
| Vorbedingung: | Der Spieler hat mindestens einen Zug gespielt |
| Nachbedingung: | Ist das Spiel fertig: Dem Spieler wird das Ergebnis präsentiert, sonst kommt der Spieler wieder zum Zug. |
| Geschäftsregeln: | Es wird überprüft, ob das Spiel fertig ist. |
| Abgrenzung / Notizen: |  |

#### UC04

|  |  |
| --- | --- |
| Nummer: | UC04 |
| Name: | Spiel beenden |
| Kurzbeschreibung: | - Das System speichert das Spiel in die Datenbank  - Das System zeigt dem Spieler das Ergebnis an |
| Akteure: | System |
| Auslöser: | In einer horizontalen, vertikalen oder diagonalen Linie sind 3 Kreuze oder 3 Kreise. |
| Vorbedingung: | Der Spieler oder das System hat gewonnen |
| Nachbedingung: | Dem Spieler wird das Ergebnis angezeigt |
| Geschäftsregeln: |  |
| Abgrenzung / Notizen: |  |

#### UC05

|  |  |
| --- | --- |
| Nummer: | UC05 |
| Name: | Frühere Ergebnisse anschauen |
| Kurzbeschreibung: | Der Spieler kann frühere Ergebnisse anschauen, dabei kann nach dem Schwierigkeitsgrad gefiltert werden.  Die neusten Ergebnisse sind zuoberst. |
| Akteure: | Spieler |
| Auslöser: | Klick auf Button «History» |
| Vorbedingung: | Der Spieler ist angemeldet |
| Nachbedingung: | Dem Spieler wird eine Liste mit den früheren Ergebnissen angezeigt |
| Geschäftsregeln: |  |
| Abgrenzung / Notizen: |  |

### Mehrspieler

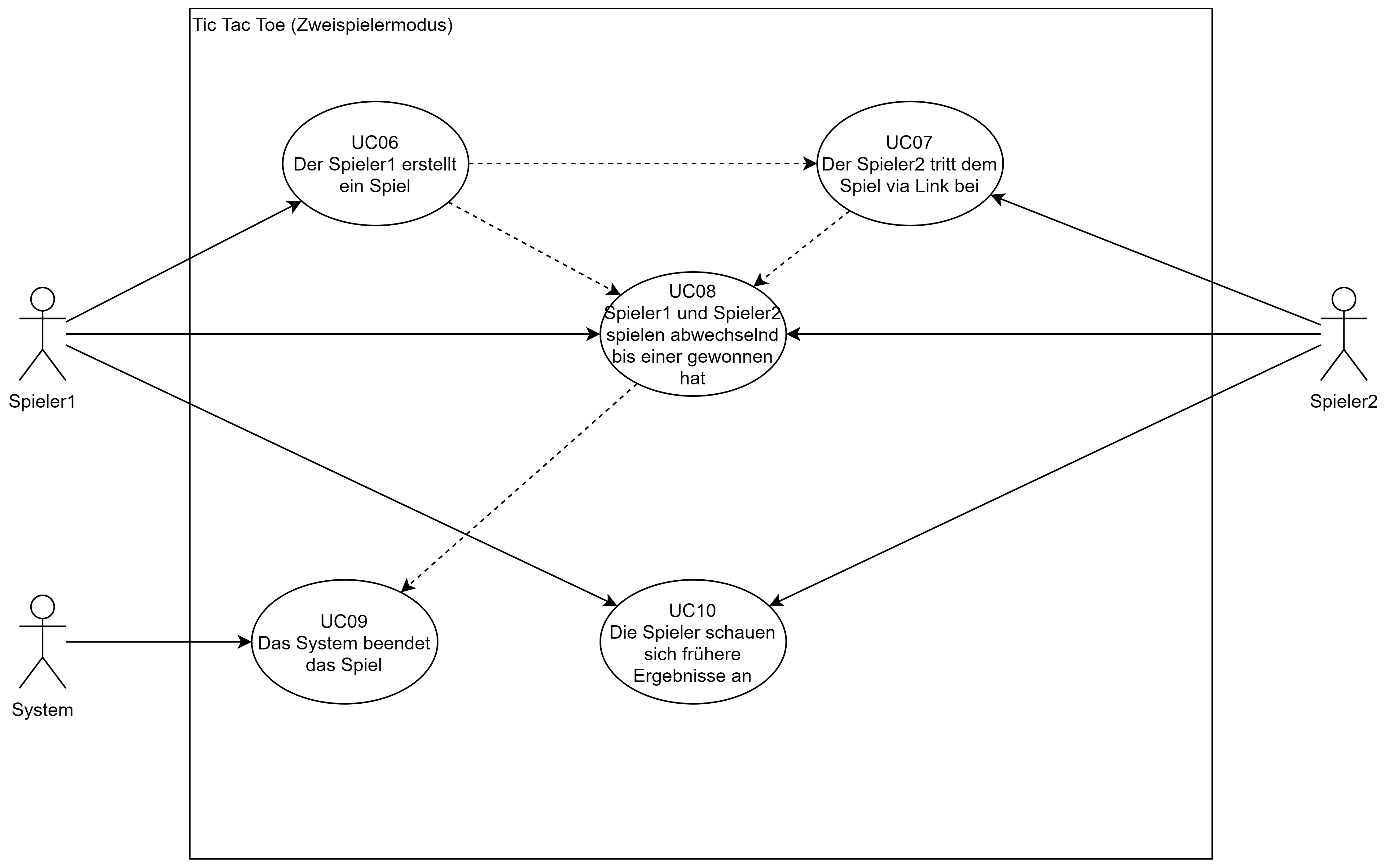


Abbildung : Use-Cases Mehrspieler

#### UC06

|  |  |
| --- | --- |
| Nummer: | UC06 |
| Name: | Der Spieler1 erstellt ein Spiel |
| Kurzbeschreibung: | Der 1. Spieler erstellt das Spiel, dabei hat er drei Optionen:   1. Er spielt gegen jemanden Zufälliges 2. Er wählt einen Gegner aus 3. Er lädt einen Gegner via Link ein |
| Akteure: | Spieler1 |
| Auslöser: | Klick auf Button «Multiplayer» |
| Vorbedingung: | Der Spieler ist angemeldet |
| Nachbedingung: |  |
| Geschäftsregeln: |  |
| Abgrenzung / Notizen: |  |

#### UC07

|  |  |
| --- | --- |
| Nummer: | UC07 |
| Name: | Der Spieler2 tritt dem Spiel via Link bei |
| Kurzbeschreibung: | Der Spieler2 klickt auf einen Link und wird dem Spiel zugewiesen. |
| Akteure: | Spieler2 |
| Auslöser: | Klick auf Link |
| Vorbedingung: | - Der Spieler ist angemeldet  - Der Link wurde nicht schon verwendet |
| Nachbedingung: | Das Spiel kann beginnen. |
| Geschäftsregeln: | Der Spieler kann nicht gegen sich selbst antreten. |
| Abgrenzung / Notizen: |  |

#### UC08

|  |  |
| --- | --- |
| Nummer: | UC08 |
| Name: | Spieler1 und Spieler2 spielen abwechselnd bis einer gewonnen hat |
| Kurzbeschreibung: | Die Spieler spielen abwechslungsweise einen Zug. Das System überprüft nach jedem Zug, ob ein Spieler das Spiel gewonnen hat. |
| Akteure: | 1. Spieler, 2. Spieler, System |
| Auslöser: | Beginn des Spiels |
| Vorbedingung: | Das Spiel wurde gestartet |
| Nachbedingung: | Der andere Spieler kann spielen, oder das Spiel ist fertig |
| Geschäftsregeln: |  |
| Abgrenzung / Notizen: | Der Spieler bekommt live-updates, ob der andere Spieler gespielt hat, ohne das Fenster aktualisieren zu müssen. |

#### UC09

|  |  |
| --- | --- |
| Nummer: | UC09 |
| Name: | Spiel beenden |
| Kurzbeschreibung: | Das System speichert das Spiel in die Datenbank  Den Spielern wird das Endergebnis angezeigt |
| Akteure: | System |
| Auslöser: | In einer horizontalen, vertikalen oder diagonalen Linie sind 3 Kreuze oder 3 Kreise. |
| Vorbedingung: | Ein Spieler hat gewonnen |
| Nachbedingung: |  |
| Geschäftsregeln: |  |
| Abgrenzung / Notizen: | Die Spieler können mit einem Klick noch ein Spiel spielen |

#### UC10

|  |  |
| --- | --- |
| Nummer: | UC10 |
| Name: | Frühere Ergebnisse anschauen |
| Kurzbeschreibung: | Der Spieler kann frühere Ergebnisse anschauen, dabei kann nach den Gegnern gefiltert werden.  Die neusten Ergebnisse sind zuoberst. |
| Akteure: | Spieler |
| Auslöser: | Klick auf Button «History» |
| Vorbedingung: | Der Spieler ist angemeldet |
| Nachbedingung: | Dem Spieler wird eine Liste mit den früheren Ergebnissen angezeigt |
| Geschäftsregeln: |  |
| Abgrenzung / Notizen: |  |

### Login

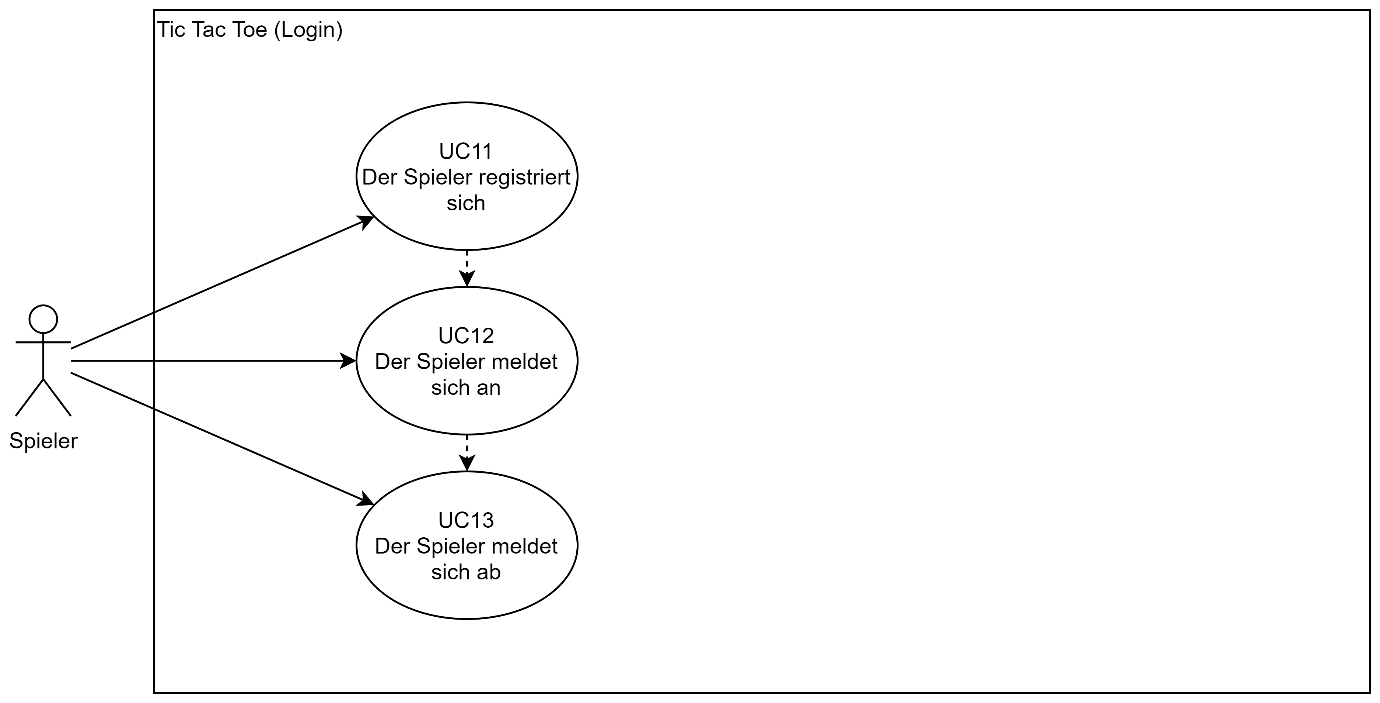


Abbildung : Use-Cases Login

#### UC11

|  |  |
| --- | --- |
| Nummer: | UC11 |
| Name: | Der Spieler registriert sich |
| Kurzbeschreibung: | Der Spieler gibt ein Benutzername und ein Passwort an.  Das Passwort muss eine Mindestlänge haben.  Das System überprüft, ob der Benutzername schon vergeben wurde. |
| Akteure: | Spieler, System |
| Auslöser: | Klick auf «Sign up» |
| Vorbedingung: | Der Spieler ist nicht angemeldet. |
| Nachbedingung: | Der Spieler kann sich mit dem Benutzernamen und Passwort anmelden |
| Geschäftsregeln: | Das Passwort wird nicht Klartext abgespeichert. |
| Abgrenzung / Notizen: | Der Benutzer wird nicht automatisch angemeldet. |

#### UC12

|  |  |
| --- | --- |
| Nummer: | UC12 |
| Name: | Der Spieler meldet sich an |
| Kurzbeschreibung: | Der Spieler meldet sich mit Benutzernamen und Passwort an. Das System überprüft, ob das Passwort und der Benutzername stimmen |
| Akteure: | Spieler, System |
| Auslöser: | Klick auf «Sign in» |
| Vorbedingung: | Der Spieler ist nicht angemeldet und hat bereits ein Login erstellt. |
| Nachbedingung: | Der Spieler ist angemeldet |
| Geschäftsregeln: | Das Passwort wird, zur Überprüfung, nicht an den Client gesendet, |
| Abgrenzung / Notizen: |  |

#### UC13

|  |  |
| --- | --- |
| Nummer: | UC13 |
| Name: | Der Spieler meldet sich ab |
| Kurzbeschreibung: | Der Spieler meldet sich ab und hat damit keinen Zugriff mehr auf die bisherigen Spiele. |
| Akteure: | Spieler, System |
| Auslöser: | Klick auf «Sign out» |
| Vorbedingung: | Der Spieler ist angemeldet |
| Nachbedingung: | Der Spieler ist nicht mehr angemeldet |
| Geschäftsregeln: |  |
| Abgrenzung / Notizen: |  |

## UI-Mockups

### Allgemein

#### Startseite eingeloggt

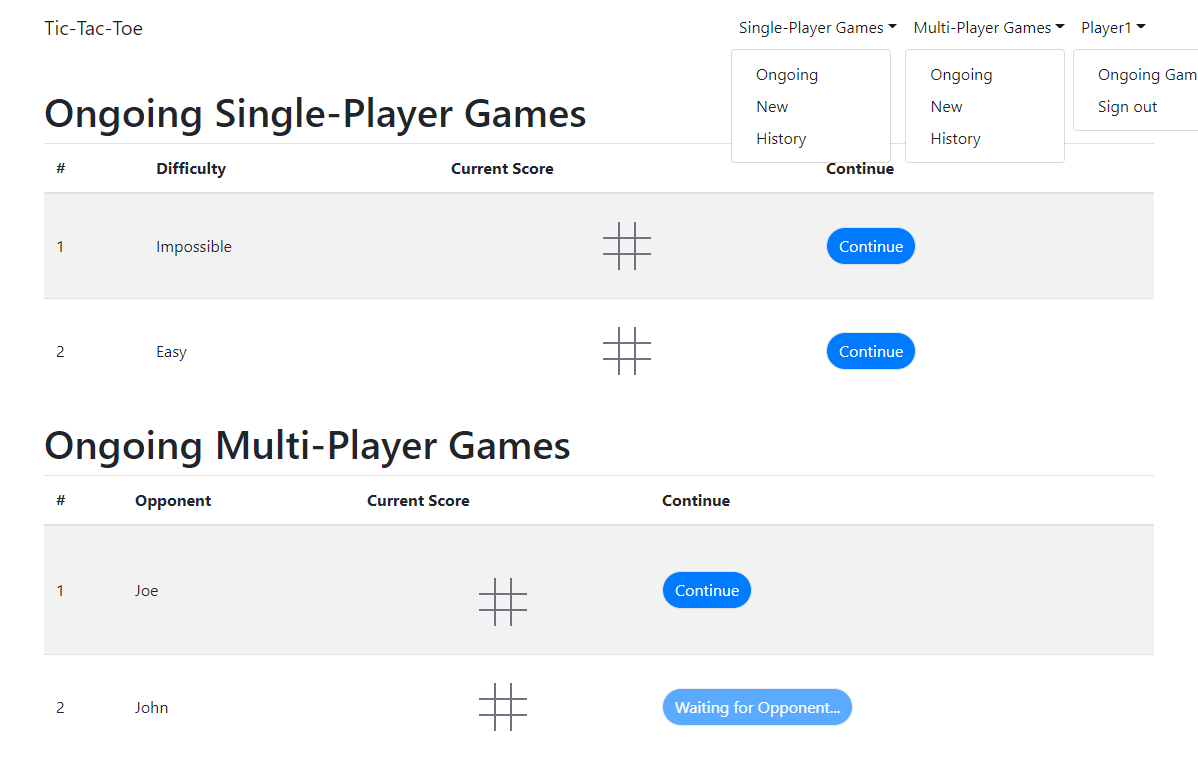


Abbildung : UI-Mockup Startseite eingeloggt

#### Sign up

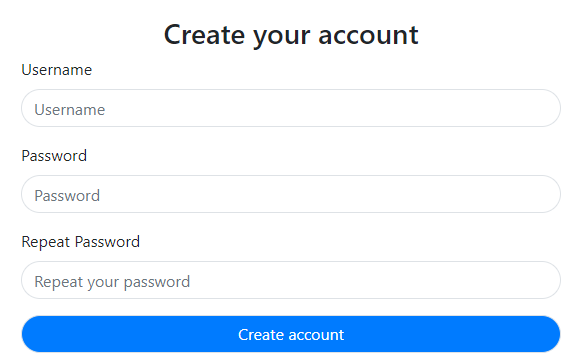


Abbildung 15: UI-Mockup Sign up

#### Sign in



Abbildung : UI-Mockup Sign in

### Einzelspieler

#### Einzelspielerspiel erstellen

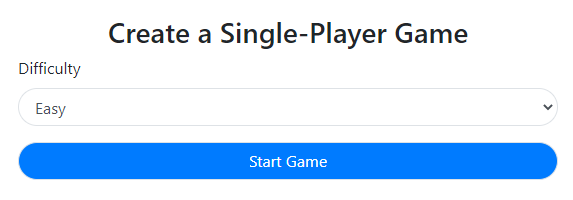


Abbildung : UI-Mockup Einzelspielerspiel erstellen

#### Einzelspielerspiel spielen

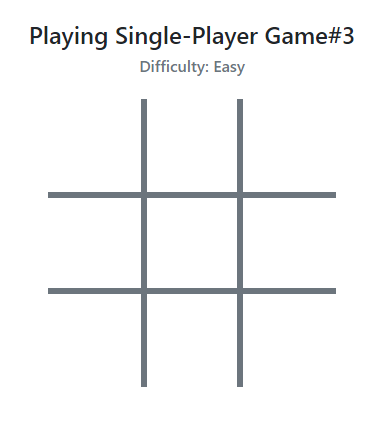


Abbildung : UI-Mockup Einzelspielerspiel spielen

#### Aktuelle Einzelspielerspiele

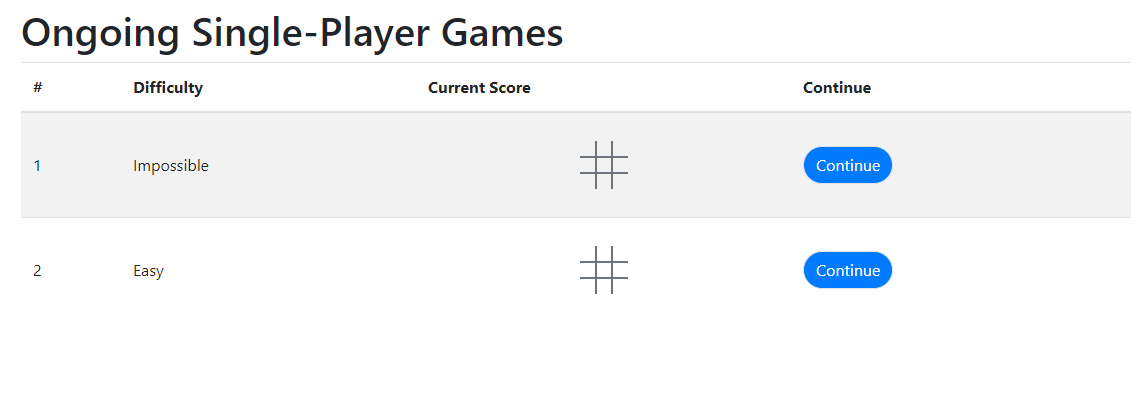


Abbildung : UI-Mockup Aktuelle Einzelspielerspiele

#### Einzelspielerspiel Ergebnis

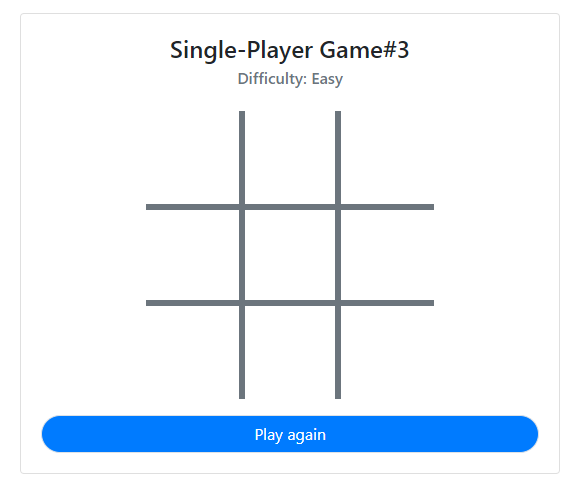


Abbildung : UI-Mockup Einzelspieler Ergebnis

#### Vergangene Einzelspielerspiele

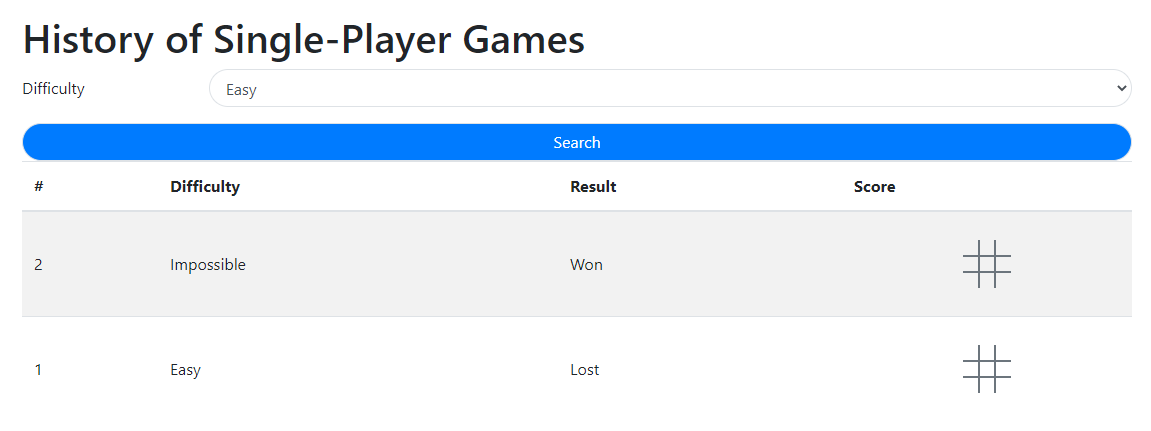


Abbildung : UI-Mockup Vergangene Einzelspielerspiele

### Mehrspieler

#### Mehrspielerspiel erstellen

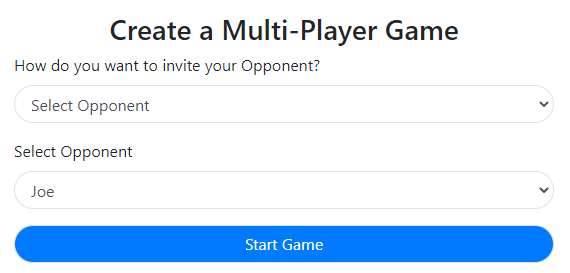


Abbildung : UI-Mockup Mehrspielerspiel erstellen

#### Mehrspielerspiel Link-Einladung

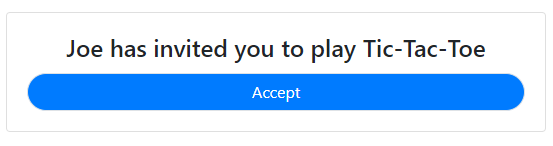


Abbildung : UI-Mockup Mehrspielerspiel Link-Einladung

#### Mehrspielerspiel spielen

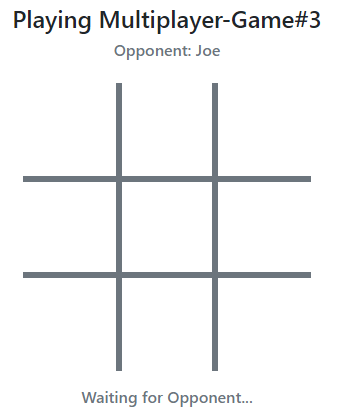


Abbildung : UI-Mockup Mehrspielerspiel spielen

#### Aktuelle Mehrspielerspielspiele

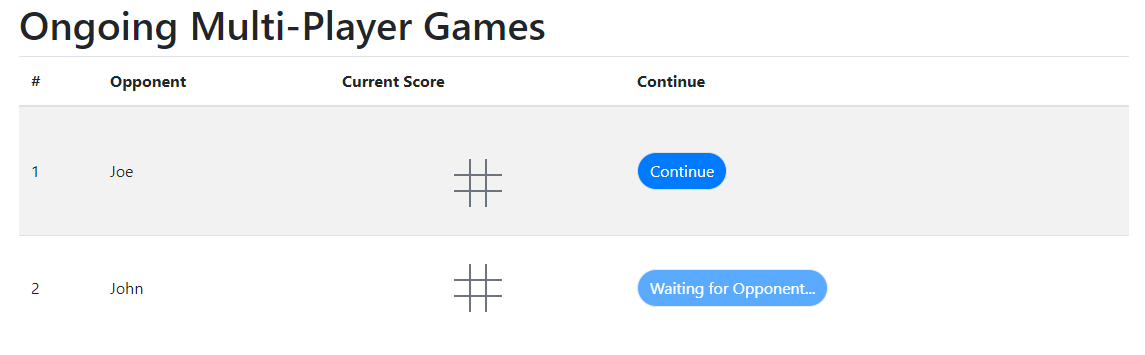


Abbildung : UI-Mockup Aktuelle Mehrspielerspiele

#### Mehrspielerspiel Ergebnis

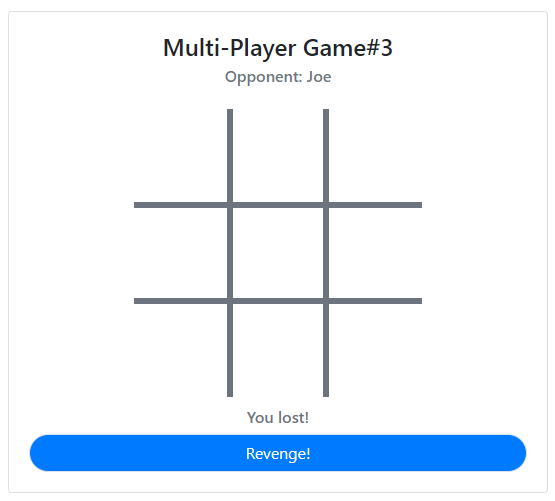


Abbildung : UI-Mockup Mehrspielerspiel Ergebnis

#### Vergangene Mehrspielerspiele

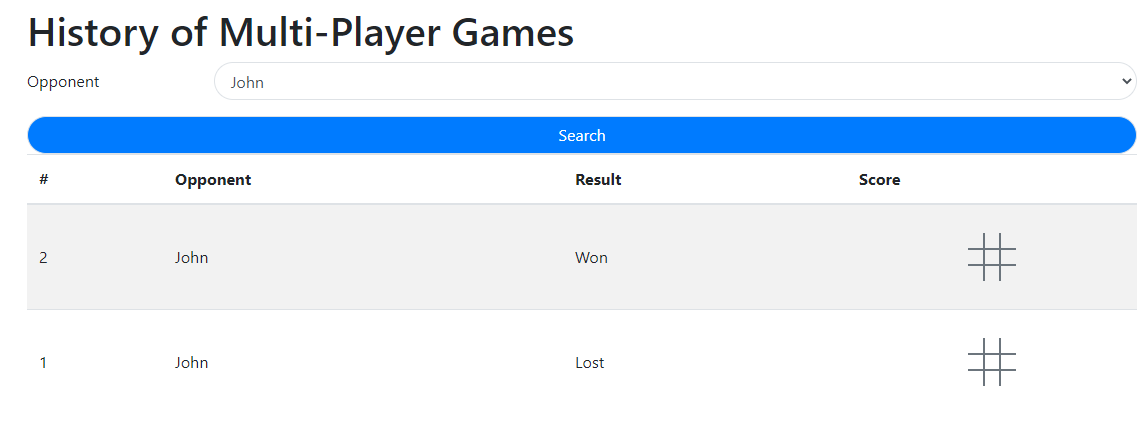


Abbildung : UI-Mockup Vergangene Mehrspielerspiele

## Testkonzept

### Während dem Entwickeln

Ich teste, während dem Entwickeln, jeden neu geschriebenen Code immer kurz, ob er so funktioniert wie ich mir das vorgestellt habe (Whitebox-Testing). Diese Minitests sind nicht systematisch, doch helfen meistens kleiner Fehler früh zu erkennen. Dies führt dazu das der Feedback-Loop kürzer ist und Fehler nicht erst beim Implementieren einer aufbauenden Funktion auftreten. Denn dann brauche ich wieder Zeit, um mich reinzudenken, den Code zu verstehen, zu überprüfen, ob das wirklich so sein soll oder nicht.

Es wird im Testkonzept nicht weiter auf diese Minitests eingegangen. Diese werden auch nirgends festgehalten oder niedergeschrieben. Sonst würde das Nutzen/Aufwand-Verhältnis nicht mehr stimmen.

### Testziele

Testen trägt wesentlich zur Qualitätssicherung der Applikation bei. Wenn man mehr auf Benutzerseite testet, stellt man die Qualität der Benutzererfahrung sicher, weniger aber die Qualität des API/Codes. Die Qualitätssicherung des API/Codes ist Ressourcen (Personen/Zeit) intensiver, da der Entwickler oft seine selbst geschriebenen Fehler nicht sieht.

Deshalb hat das Testkonzept vor allem zum Ziel die Qualität der Benutzererfahrung sicherzustellen.

### Teststrategie

Ich habe mich für Systemtests und Blackbox-Testing entschieden. Diese Art von Testing ist mir aus dem Lehrbetrieb und der Schule bereits gut bekannt.

Beim Blackbox-Testing hat man immer eine Eingabe und eine Ausgabe. Es ist des Weiteren kein Wissen über den Code notwendig.



Abbildung : Testmethode Blackbox-Testing

Diese beiden Methoden haben auch den Vorteil, dass sich aufgrund der klaren und vorhandenen Anforderungen leichter Tests schreiben lassen.

Getestet werden alle Use Cases, bei welchen der Spieler beteiligt ist. Es wird pro Use-Case mindestens ein Testfall definiert. Bei den Testfällen werden die nicht funktionalen Anforderungen auch berücksichtigt.

Bei der Reihenfolge wird nach Hardest-First verfahren. Das heisst, dass die wichtigsten Funktionen zuerst getestet werden. Hardest-First ist aber auch situationsbedingt, sodass beim Re-Testing, Funktionen, welche geflickt worden sind oder vielleicht nicht mehr gehen, zuerst getestet werden müssten.

Um die Seite nutzen zu können sind gewisse Funktionen ein Muss, diese sind auch gleich die Funktionen, welche beim ersten Testdurchlauf am wichtigsten sind, respektive zuerst getestet werden. Diese sind in der nachfolgenden Tabelle definiert.

|  |
| --- |
| Funktion |
| Registrieren |
| Anmelden |
| Mehrspielerspiel spielen |
| Einzelspielerspiel spielen |

Tabelle : Muss-Funktionen

Da Hermes im Kontext der IPA eher einem Wasserfallmodell gleicht und somit nach dem Testen ein Fixen des Codes keinen Sinn macht, wird nur ein Testdurchlauf gemacht. Ausnahme ist, wenn ich noch Zeitreserve habe. Dann werde ich die fehlgeschlagenen Tests zuerst beheben und einen weiteren Testdurchlauf starten.

### Aufbau eines Tests

Damit die Tests Systematisch sind, sind die Tests immer gleich aufgebaut. Und zwar wie folgt:

|  |  |
| --- | --- |
| **ID** | *T01 (eindeutige Bezeichnung)* |
| **Testobjekt** | *Use-Case oder Name Z.B: «UC01» oder «vergangenes Einzelspielerspiel suchen»* |
| **Beschreibung** | *Beschreibung was der Testfall abdeckt* |
| **Testvoraussetzung** | *Beschreiben was erfüllt sein muss, um den Test durchzuführen. Z.B: Benutzer muss eingeloggt sein.* |
| **Testschritte** | *Anleitung was gemacht werden muss, um den Test durchzuführen.*  *Dabei muss die Anleitung auch von einer anderen Person durchgeführt werden können.* |
| **Erwartetes Ergebnis** | *Das Resultat welches vom Testautor erwartet wird.* |
| **Tester** | *Testperson, im Fall der IPA immer ich.* |
| **Datum** | *Datum der Testdurchführung* |
| **Ergebnis** | *Erfüllt oder Fehlerklasse* |
| **Fehlerbeschreibung** | *Beschreibung des Fehlers, wenn eine Fehlerklasse vorhanden ist* |

Tabelle : Vorlage Testfall

### Testrahmen

Damit mit dem Testen begonnen werden kann, müssen mindestens alle Funktionen aus «Tabelle 6: Muss-Funktionen» implementiert sein. Im Idealfall sollen aber alle Funktionen implementiert sein.

Das Testen ist beendet, wenn entweder alle Tests funktionieren oder es zeitliche Gründe gibt.

Ich unterscheide zwischen drei Fehlerklassen:

* Schwerwiegende Fehler, welche die Applikation teilweise oder ganz unbrauchbar machen.
* Kritische Fehler, welche den Nutzen der Applikation vermindern aber nicht unbrauchbar machen.
* Geringe Fehler, welche zwar die Qualität der Applikation vermindern, aber keine oder nur einen geringen Einfluss auf die Qualität des Produkts haben.

Das Testen wird abgebrochen, wenn bei mindestens 3 Testfällen die Vorbedingung nicht erfüllt war.

Die Testfälle werden von mir geschrieben, dies hat leider einen Nachteil, weil ich auch der Entwickler bin. Dies ist aber in einer IPA nicht anders möglich.

Die Testfälle wird der Tester (ich) gegen Ende der Test-IPA durchführen. Voraussichtlich ist die der 21.01.2021.

### Testinfrastruktur

Ich werde die Tests auf meinem privaten Rechner durchführen, dieser hat folgende Spezifikationen:

* OS: Windows 10 Pro (Version 1909, OS build 18363.1316, 64-bit)
* Browser: Google Chrome (Version 87.0.4280.141 (Official Build) (64-bit))
* CPU: Intel® Core™ i7-8700K
* RAM: 16 GB
* GPU: Nvidia GeForce GTX 1080 Ti

Aufgrund des Hardest-First-Verfahren kann ich die Daten, welche ich während dem Testen generiere zum Testen der weiteren Tests verwenden (sofern diese nicht Fehlschlagen).

Ich verwende zum Testen, nebst dem Browser, keine Hilfsmittel.

### Testfälle

#### Testfall T01

|  |  |
| --- | --- |
| **ID** | *T01* |
| **Testobjekt** | UC11 |
| **Beschreibung** | Der Benutzer registriert sich |
| **Testvoraussetzung** | Der Benutzer ist nicht angemeldet und es existiert kein Konto mit dem Namen John. |
| **Testschritte** | 1. Der Benutzer klickt auf «Sign up» 2. Der Benutzer gibt im Feld «Username» «John» ein 3. Der Benutzer gibt im Feld «Password» «tictac123» ein 4. Der Benutzer gibt im Feld «Username» «tictac123» ein 5. Der Benutzer klickt auf «Create account» |
| **Erwartetes Ergebnis** | Der Benutzer ist nun registriert. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T01

#### Testfall T02

|  |  |
| --- | --- |
| **ID** | *T02* |
| **Testobjekt** | UC12 |
| **Beschreibung** | Der Benutzer meldet sich an |
| **Testvoraussetzung** | Der Benutzer ist nicht angemeldet und der Testfall T01 war erfolgreich |
| **Testschritte** | 1. Der Benutzer klickt auf «Sign in» 2. Der Benutzer gibt im Feld «Username» «John» ein 3. Der Benutzer gibt im Feld «Password» «tictac123» ein 4. Der Benutzer klickt auf «Sign in» |
| **Erwartetes Ergebnis** | Der Benutzer ist nun angemeldet. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T02

#### Testfall T03

|  |  |
| --- | --- |
| **ID** | *T03* |
| **Testobjekt** | UC13 |
| **Beschreibung** | Der Benutzer meldet sich ab |
| **Testvoraussetzung** | Der Benutzer muss eingeloggt sein. |
| **Testschritte** | 1. Der Benutzer öffnet das Account Menü oben rechts 2. Der Benutzer klickt auf «Sign out» |
| **Erwartetes Ergebnis** | Der Benutzer ist abgemeldet und wird auf die Startseite weitergeleitet. Er hat keinen Zugriff mehr auf irgendwelche Spiele. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T03

#### Testfall T04

|  |  |
| --- | --- |
| **ID** | *T04* |
| **Testobjekt** | UC06 |
| **Beschreibung** | Mehrspieler spiel erstellen und gegen zufälligen Spieler spielen. |
| **Testvoraussetzung** | Es müssen zwei Accounts vorhanden sein. |
| **Testschritte** | 1. Im ersten Account anmelden 2. Klick auf «Multi-Player Games» 3. Klick auf «New» 4. Bei «How do you want to invite your Opponent?» «Random Opponent» auswählen 5. Klick auf «Start game» 6. Ausloggen 7. Mit dem zweiten Account anmelden 8. Klick auf «Multi-Player Games» 9. Klick auf «New» 10. Bei «How do you want to invite your Opponent?» «Random Opponent» auswählen 11. Klick auf «Start game» |
| **Erwartetes Ergebnis** | Der zweite Account ist im Spiel und kann den ersten Zug machen. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T04

#### Testfall T05

|  |  |
| --- | --- |
| **ID** | *T05* |
| **Testobjekt** | UC06 |
| **Beschreibung** | Mehrspieler spiel erstellen und gegen bestimmten Spieler spielen. |
| **Testvoraussetzung** | Es müssen zwei Accounts vorhanden sein. |
| **Testschritte** | 1. Im ersten Account anmelden 2. Klick auf «Multi-Player Games» 3. Klick auf «New» 4. Bei «How do you want to invite your Opponent?» «Select Opponent» auswählen 5. Bei «Select Opponent» den zweiten Account auswählen 6. Klick auf «Start game» |
| **Erwartetes Ergebnis** | Der erste Account ist im Spiel und kann den ersten Zug machen. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T05

#### Testfall T06

|  |  |
| --- | --- |
| **ID** | *T06* |
| **Testobjekt** | UC06/UC7 |
| **Beschreibung** | Mehrspieler spiel erstellen und Spieler mit Link einladen. |
| **Testvoraussetzung** | Es müssen zwei Accounts vorhanden sein. |
| **Testschritte** | 1. Im ersten Account anmelden 2. Klick auf «Multi-Player Games» 3. Klick auf «New» 4. Bei «How do you want to invite your Opponent?» «Invite with Link» auswählen 5. Klick auf «Start game» 6. Erhaltener Link kopieren 7. Ausloggen 8. Mit dem zweiten Account anmelden 9. Den kopierten Link öffnen |
| **Erwartetes Ergebnis** | Der zweite Account ist im Spiel und kann den ersten Zug machen. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T06

#### Testfall T07

|  |  |
| --- | --- |
| **ID** | *T07* |
| **Testobjekt** | UC08 |
| **Beschreibung** | Zwei Benutzer spielen, bis einer gewonnen hat oder das Spielbrett voll ist. |
| **Testvoraussetzung** | Es müssen zwei Accounts vorhanden sein.  Ein Spiel zwischen diesen Accounts muss schon gestartet sein. |
| **Testschritte** | 1. Der Benutzer hat ein normales Fenster und ein Inkognito Fenster offen. 2. Der Benutzer meldet sich mit jeweils einem Account in einem Fenster an 3. Der Benutzer spielt für beide Accounts, indem er nur Felder anklickt, auf welchen nicht bereits ein X oder ein O ist. 4. Dies wiederholt der Benutzer, bis in einer Zeile, einer Spalte oder einer Diagonalen nur Xs oder nur Os sind. Oder das Spielbrett voll ist. |
| **Erwartetes Ergebnis** | Beiden Accounts wird das Ergebnis angezeigt. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T07

#### Testfall T08

|  |  |
| --- | --- |
| **ID** | *T08* |
| **Testobjekt** | UC10 |
| **Beschreibung** | Frühere Mehrspielerspiele anschauen |
| **Testvoraussetzung** | Ein Spiel zwischen zwei Accounts muss soeben abgeschlossen worden sein. |
| **Testschritte** | 1. Der Benutzer meldet sich mit dem ersten Account an. 2. Klickt auf «Multi-Player Games» 3. Klickt auf «History» 4. Beim Feld «Opponent» wählt er den zweiten Account aus |
| **Erwartetes Ergebnis** | Der Benutzer sieht nun das soeben gespielte Spiel. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T08

#### Testfall T09

|  |  |
| --- | --- |
| **ID** | *T09* |
| **Testobjekt** | UC01 |
| **Beschreibung** | Einzelspielerspiel erstellen |
| **Testvoraussetzung** | Der Benutzer ist mit einem Account angemeldet |
| **Testschritte** | 1. Klick auf «Single-Player Games» 2. Klick auf «New» 3. Bei «Difficulty» «Easy» auswählen 4. Klick auf «Start game» |
| **Erwartetes Ergebnis** | Dem Spieler wird das Spiel angezeigt und er kann einen ersten Zug machen. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T09

#### Testfall T10

|  |  |
| --- | --- |
| **ID** | *T10* |
| **Testobjekt** | UC02/UC03 |
| **Beschreibung** | Einzelspieler Spiel spielen |
| **Testvoraussetzung** | Der Benutzer hat ein Einzelspielerspiel gestartet. |
| **Testschritte** | 1. Der Benutzer setzt klickt auf ein leeres Feld 2. Der Benutzer wartet, bis das System seinen Zug gespielt hat 3. Dies wiederholt der Benutzer, bis in einer Zeile, einer Spalte oder einer Diagonalen nur Xs oder nur Os sind. Oder das Spielbrett voll ist. |
| **Erwartetes Ergebnis** | Dem Benutzer wird das Endergebnis angezeigt. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T10

#### Testfall T11

|  |  |
| --- | --- |
| **ID** | *T11* |
| **Testobjekt** | UC05 |
| **Beschreibung** | Frühere Einzelspielerspiele anschauen |
| **Testvoraussetzung** | Ein Einzelspielerspiele mit dem Schwierigkeitsgrad «Einfach» muss soeben abgeschlossen worden sein. Und der Benutzer ist angemeldet. |
| **Testschritte** | 1. Klickt auf «Single-Player Games» 2. Klickt auf «History» 3. Beim Feld «Difficulty» wählt er «Einfach» aus |
| **Erwartetes Ergebnis** | Der Benutzer sieht nun das soeben gespielte Spiel. |
| **Tester** |  |
| **Datum** |  |
| **Ergebnis** |  |
| **Fehlerbeschreibung** |  |

Tabelle : Vorlage Testfall T11

## Informationssicherheit und Datenschutz (ISDS)

Während der IPA verwende ich keine vertraulichen Daten. Die Daten, welche auf der Datenbank sind, sind nur intern zugänglich. Um die Daten von der Webseite abzurufen, muss man sich anmelden.

Der Spieler kann auch nicht Daten von anderen Spielern ansehen. Es sei denn, es ist ein Mehrspieler spiel, bei welchem er beteiligt war.

Die IPA wird lokal auf dem Rechner und auf den Servern von GitHub gespeichert. Auf das GitHub Projekt habe nur ich und mein Auftraggeber Zugriff. Auf den lokalen Rechner hat niemand ausser mir Zugang. Ich sperre auch mein Rechner, wenn ich meinen Arbeitsplatz verlasse.

Damit sind alle Daten (Datenbank und IPA-Dokumentation) immer passwortgeschützt und vor Unbefugten sicher.

# Realisierung

## System realisieren

Backend template: <https://github.com/gothinkster/node-express-realworld-example-app>

Während der IPA arbeiten die Kandidaten rund 40h-50h an der Umsetzung. Um den Lesern der Dokumentation einen guten Überblick und ein Verständnis der getätigten Arbeit zu vermitteln, ist es erforderlich, dass aufgezeigt wird, wie das Produkt/System technisch umgesetzt wurde. Dazu müssen Abweichungen zum Konzept und die Umsetzung von relevanten Inhalten (Konfigurationen; Spezielle Methoden/Klassen; o.ä.) aufgezeigt werden.

Beispiele:

* Es wird ein Mockup-Konzept erarbeitet. In der Umsetzung soll das fertige UI ersichtlich sein. inkl. Beschreibung der Abweichungen
* Bei einem Serverkonzept muss die Umsetzung (Installationsschritte) erkennbar sein Bsp. in Form von Printscreens mit dazugehörigen Beschreibungen.
* Ein konzeptionelles Klassendiagramm im Entwurf verlangt ein effektives Klassendiagramm in der Umsetzung mit Beschreibung der Abweichungen.

## Testprotokoll

Es empfiehlt sich 1 zu 1 Testfallset auf dem Konzept zu entnehmen und dieses durch eine weitere Tabelle "Testdurchführung und Testergebnis" zu ergänzen.

### Testfall T-001

|  |  |
| --- | --- |
| **ID / Bezeichnung** | *T-001* |
| **Beschreibung** |  |
| **Testvoraussetzung** |  |
| **Testschritte** |  |
| **Erwartetes Ergebnis** |  |

| **Tester (Testperson)** |  |
| --- | --- |
| **Datum Testdurchführung** |  |
| **Fehlerklasse (Testergebnis)** |  |
| **Fehlerbeschreibung** |  |

### Testfall T-002

|  |  |
| --- | --- |
| **ID / Bezeichnung** | *T-002* |
| **Beschreibung** |  |
| **Testvoraussetzung** |  |
| **Testschritte** |  |
| **Erwartetes Ergebnis** |  |

| **Tester (Testperson)** | Florin Barbisch |
| --- | --- |
| **Datum Testdurchführung** |  |
| **Fehlerklasse (Testergebnis)** |  |
| **Fehlerbeschreibung** |  |

# Reflexion

Projekt revuepassieren lassen. Was ist gut gelaufen? Was kann man besser machen?

# Schlusswort

Persönliches Fazit.

Offene Punkte

Zukünftige Punkte

# Literatur- und Quellverzeichnis

|  |  |
| --- | --- |
| [1] | A. Müller, «Hermes 5.1 IPA Factsheet,» 2016. |

Internet Quelle:

Name des Autors (falls erkennbar), "Titel der Seite", Webadresse, Datum des letzten Zugriffs

Beispiel Internet-Quellenangabe:

"Koala", http://de.wikipedia.org/wiki/Koala, 22.03.2008

Buch Quelle:

Erklärung Name des Autors: Titel. Verlag, Jahr, Seite, auf der der zitierte Text steht.

Beispiel Buch-Quellenangabe:

Ernst Walter Bauer: Humanbiologie. Cornelsen, 2006, S. 50.

# Glossar

Alphabetisch sortiertes Glossar!

|  |  |
| --- | --- |
| **Begriff** | **Bedeutung** |
| Tic-Tac-Toe | TODO |
| API | TODO |
| REST | TODO |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabelle : Glossar

# Anhang

TODO