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: Florin Barbisch

Valid-Experte: Daniel Mosimann

Hauptexperte: Daniel Mosimann

Nebenexperte: -

Projektleiter: Florin Barbisch

Entwickler: Florin Barbisch

Tester: Florin Barbisch

# Inhaltsverzeichnis

Inhaltsverzeichnis 2

Abbildungsverzeichnis 7

Tabellenverzeichnis 9

1 Einleitung 11

1.1 Inhalt und Zweck des Dokuments 11

Teil 1: Ablauf Organisation und Umfeld 12

2 Aufgabestellung 12

2.1 Ausgangslage 12

2.2 Detaillierte Aufgabenstellung 12

2.3 Mittel und Methoden 12

2.4 Vorkennnisse 12

2.5 Vorarbeiten 13

2.6 Neue Lerninhalte 13

2.7 Arbeiten in den letzten 6 Monaten 13

3 Projektmethodik 13

3.1 Projektmethode 13

3.2 Szenario 13

3.3 Phasen 13

3.4 Module 14

3.5 Aufgaben 14

3.6 Projektorganisation 15

4 Zeitplanung 16

4.1 Zeitplan 16

5 Organisation der IPA 17

5.1 Arbeitsplatz 17

5.2 Datensicherung 17

5.3 Wiederherstellung 18

6 Firmenstandards 19

7 Arbeitsjournal 20

7.1 KW 24: 08.06.2020 20

7.2 KW 25: 15.06.2020 21

7.3 KW 26: 22.06.2020 21

7.4 KW 27: 29.06.2020 22

7.5 KW 33: 10.08.2020 22

7.6 KW 34: 17.08.2020 23

7.7 KW 35: 24.08.2020 23

7.8 KW 36: 31.08.2020 24

7.9 KW 37: 07.09.2020 24

7.10 KW 38: 14.09.2020 25

7.11 KW 39: 21.09.2020 26

7.12 KW 43: 19.10.2020 27

7.13 KW 44: 26.10.2020 27

7.14 KW 45: 02.11.2020 27

7.15 KW 46: 09.11.2020 28

7.16 KW 47: 16.11.2020 28

7.17 KW 48: 23.11.2020 29

7.18 KW 49: 30.11.2020 29

7.19 KW 50: 07.12.2020 30

7.20 KW 51: 14.12.2020 30

7.21 KW 01 04.01.2021 31

7.22 KW 02 11.01.2021 32

7.23 KW 03 18.01.2021 33

Teil 2: Projektdokumentation 34

8 Kurzfassung 34

8.1 Ausgangssituation 34

8.2 Umsetzung 34

8.3 Ergebnis 34

9 Initialisierung 35

9.1 Analyse 35

9.1.1 Studie; IST-Zustand 35

9.1.2 Projektziele 39

9.1.3 Anforderungen 39

9.2 Varianten 39

9.2.1 Variante 1 39

9.2.2 Variante 2 40

9.2.3 Entscheidungsmatrix 40

9.2.4 Gewählte Variante 40

10 Konzept 40

10.1 Architekturdiagramm 40

10.2 Use Cases 41

10.2.1 Einzelspieler 41

10.2.2 Mehrspieler 43

10.2.3 Login 46

10.3 UI-Mockups 48

10.3.1 Allgemein 48

10.3.1 Einzelspieler 49

10.3.1 Mehrspieler 51

10.4 Testkonzept 54

10.4.1 Während dem Entwickeln 54

10.4.2 Testziele 54

10.4.3 Teststrategie 54

10.4.4 Aufbau eines Tests 55

10.4.5 Testrahmen 56

10.4.6 Testinfrastruktur 56

10.5 Testfälle 57

10.5.1 Testfall T01 57

10.5.2 Testfall T02 58

10.5.3 Testfall T03 59

10.5.4 Testfall T04 60

10.5.5 Testfall T05 61

10.5.6 Testfall T06 62

10.5.7 Testfall T07 63

10.5.8 Testfall T08 64

10.5.9 Testfall T09 64

10.5.10 Testfall T10 65

10.5.11 Testfall T11 66

10.6 Informationssicherheit und Datenschutz (ISDS) 67

11 Realisierung 68

11.1 System realisieren 68

11.1.1 REST 68

11.1.2 Logik des Computerspieler 70

11.1.3 Socket.io 71

11.1.4 Logik aus Komponenten auslagern 73

11.2 GUI 74

11.2.1 Allgemein 75

11.2.2 Einzelspieler 76

11.2.3 Mehrspieler 80

11.3 Testprotokoll 84

11.3.1 Testfall T01 84

11.3.2 Testfall T02 85

11.3.3 Testfall T03 86

11.3.4 Testfall T04 87

11.3.5 Testfall T05 88

11.3.6 Testfall T06 89

11.3.7 Testfall T07 90

11.3.8 Testfall T08 91

11.3.9 Testfall T09 92

11.3.10 Testfall T10 93

11.3.11 Testfall T11 94

12 Reflexion 95

13 Schlusswort 95

14 Literatur- und Quellverzeichnis 96

15 Glossar 97

16 Anhang 98

# Abbildungsverzeichnis

Abbildung 1: Hermes-Phasen 13

Abbildung 2: Hermes-Module 14

Abbildung 3: Zeitplan 16

Abbildung 4: Arbeitsplatz 17

Abbildung 5: Daten wiederherstellen mit git checkout 18

Abbildung 6: Homescreen der alten Website 35

Abbildung 7: Während eines Einzelspiels auf der alten Webseite 36

Abbildung 8: Suche nach vergangenen Einzelspielen auf der alten Webseite 37

Abbildung 9: Suche nach Mehrspielerspielen auf der alten Webseite 38

Abbildung 10: Architekturdiagramm 40

Abbildung 11: Use-Cases Einzelspieler 41

Abbildung 12: Use-Cases Mehrspieler 43

Abbildung 13: Use-Cases Login 46

Abbildung 14: UI-Mockup Startseite eingeloggt 48

Abbildung 15: UI-Mockup Sign up 48

Abbildung 16: UI-Mockup Sign in 49

Abbildung 17: UI-Mockup Einzelspielerspiel erstellen 49

Abbildung 18: UI-Mockup Einzelspielerspiel spielen 49

Abbildung 19: UI-Mockup Aktuelle Einzelspielerspiele 50

Abbildung 20: UI-Mockup Einzelspieler Ergebnis 50

Abbildung 21: UI-Mockup Vergangene Einzelspielerspiele 51

Abbildung 22: UI-Mockup Mehrspielerspiel erstellen 51

Abbildung 23: UI-Mockup Mehrspielerspiel spielen 51

Abbildung 24: UI-Mockup Aktuelle Mehrspielerspiele 52

Abbildung 25: UI-Mockup Mehrspielerspiel Ergebnis 52

Abbildung 26: UI-Mockup Vergangene Mehrspielerspiele 53

Abbildung 27: Testmethode Blackbox-Testing 54

Abbildung 28: Berechtigungen des GitHub-Repository 67

Abbildung 29: GET-Ressource in expressJS 68

Abbildung 30: Umwandeln eines Parameters zu einem Objekt 68

Abbildung 31: Spielzug vom Controller aus ausführen 69

Abbildung 32: Logik von Spielzug ausführen 69

Abbildung 33: Ablauf des Computerspielers 70

Abbildung 34: Socket.io-Server an expressJS-Server anhängen 71

Abbildung 35: Event über socket.io emittieren 71

Abbildung 36: Verarbeiten eines socket.io Events 72

Abbildung 37: Aufruf von FETCH\_MULTIPLAYER\_GAMES 73

Abbildung 38: Action FETCH\_MULTIPLAYER\_GAMES 73

Abbildung 39: Finales-GUI Startseite eingeloggt 75

Abbildung 40: Finales-GUI Sign up 76

Abbildung 41: Finales-GUI Sign in 76

Abbildung 42: Finales-GUI Einzelspieler erstellen 76

Abbildung 43: Finales-GUI Einzelspieler spielen 77

Abbildung 44: Finales-GUI Aktuelle Einzelspielerspiele 77

Abbildung 45: Finales-GUI Einzelspielerspiel Ergebnis 78

Abbildung 46: Finales-GUI Vergangene Einzelspielerspiele 79

Abbildung 47: Finales-GUI Mehrspielerspiel erstellen 80

Abbildung 48: Finales-GUI Mehrspielerspiel erstellen 80

Abbildung 49: Finales-GUI Aktuelle Mehrspielerspiele 81

Abbildung 50: Finales-GUI Mehrspielerspiel Ergebnis 82

Abbildung 51: Finales-GUI Vergangene Mehrspielerspiele 83

# Tabellenverzeichnis

Tabelle 1: Hermes-Aufgaben 14

Tabelle 2: Projektrollen 15

Tabelle 3: Verwendete CASE-Tools 19

Tabelle 4: Arbeitsjournaleintrag KW24 20

Tabelle 5: Arbeitsjournaleintrag KW25 21

Tabelle 6: Arbeitsjournaleintrag KW26 21

Tabelle 7: Arbeitsjournaleintrag KW27 22

Tabelle 8: Arbeitsjournaleintrag KW33 22

Tabelle 9: Arbeitsjournaleintrag KW34 23

Tabelle 10: Arbeitsjournaleintrag KW35 23

Tabelle 11: Arbeitsjournaleintrag KW36 24

Tabelle 12: Arbeitsjournaleintrag KW37 24

Tabelle 13: Arbeitsjournaleintrag KW38 25

Tabelle 14: Arbeitsjournaleintrag KW39 26

Tabelle 15: Arbeitsjournaleintrag KW43 27

Tabelle 16: Arbeitsjournaleintrag KW44 27

Tabelle 17: Arbeitsjournaleintrag KW45 27

Tabelle 18: Arbeitsjournaleintrag KW46 28

Tabelle 19: Arbeitsjournaleintrag KW47 28

Tabelle 20: Arbeitsjournaleintrag KW48 29

Tabelle 21: Arbeitsjournaleintrag KW49 29

Tabelle 22: Arbeitsjournaleintrag KW50 30

Tabelle 23: Arbeitsjournaleintrag KW51 30

Tabelle 24: Arbeitsjournaleintrag KW01 31

Tabelle 25: Arbeitsjournaleintrag KW02 32

Tabelle 26: Arbeitsjournaleintrag KW03 33

Tabelle 27: Projektziele 39

Tabelle 28: Entscheidungsmatrix 40

Tabelle 29: Use Case UC01 41

Tabelle 30: Use Case UC02 42

Tabelle 31: Use Case UC03 42

Tabelle 32: Use Case UC04 42

Tabelle 33: Use Case UC05 43

Tabelle 34: Use Case UC06 44

Tabelle 35: Use Case UC07 44

Tabelle 36: Use Case UC08 44

Tabelle 37: Use Case UC09 45

Tabelle 38: Use Case UC10 45

Tabelle 39: Use Case UC11 46

Tabelle 40: Use Case UC12 46

Tabelle 41: Use Case UC13 47

Tabelle 42: Muss-Funktionen 55

Tabelle 43: Vorlage Testfall 55

Tabelle 44: Vorlage Testfall T01 57

Tabelle 45: Vorlage Testfall T02 58

Tabelle 46: Vorlage Testfall T03 59

Tabelle 47: Vorlage Testfall T04 60

Tabelle 48: Vorlage Testfall T05 61

Tabelle 49: Vorlage Testfall T06 62

Tabelle 50: Vorlage Testfall T07 63

Tabelle 51: Vorlage Testfall T08 64

Tabelle 52: Vorlage Testfall T09 64

Tabelle 53: Vorlage Testfall T10 65

Tabelle 54: Vorlage Testfall T11 66

Tabelle 55: Schritte des Computerspielers 71

Tabelle 56: Ergebnisse erster Testdurchlauf 84

Tabelle 57: Erste Durchführung Testfall T01 84

Tabelle 58: Erste Durchführung Testfall T02 85

Tabelle 59: Erste Durchführung Testfall T03 86

Tabelle 60: Erste Durchführung Testfall T04 87

Tabelle 61: Erste Durchführung Testfall T05 88

Tabelle 62: Erste Durchführung Testfall T06 89

Tabelle 63: Erste Durchführung Testfall T07 90

Tabelle 64: Erste Durchführung Testfall T08 91

Tabelle 65: Erste Durchführung Testfall T09 92

Tabelle 66: Erste Durchführung Testfall T10 93

Tabelle 67: Erste Durchführung Testfall T11 94

Tabelle 68: Glossar 97

Tabelle 69: Codezeilen-Übersicht 98

# Einleitung

## Inhalt und Zweck des Dokuments

Dieses Dokument dient der Dokumentation der Arbeiten und Ergebnisse, welche der Kandidat während der IPA erarbeitet.

Zu gleich dient dieses Dokument auch der Bewertung und der Rekonstruktion/Nachvollziehbarkeit der getätigten Arbeiten.

Das Dokument ist in zwei Teilen unterteilt. Der erste Teil beinhaltet alles Organisatorische. Dies ist alles, was nicht nach der definierten Projektmethode umgesetzt wird. Dieser Teil ist nach den organisatorischen Inhalten unterteilt.

Im zweiten Teil ist das eigentliche Projekt dokumentiert, welches nach der Projektmethode umgesetzt wird. Dieser Teil ist nach den Projektphasen unterteilt. In diesen Projektphasen wird weiter nach Thema unterteilt.

Am Ende des Dokuments sind noch Themen, welche die ganze Arbeit betreffen aber nicht in den organisatorischen Teil passen. Dies sind namentlich: Reflexion, Schlusswort, Literatur- und Quellenverzeichnis, Glossar und Anhänge (wie zum Beispiel Code-Listings).

# 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», 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.

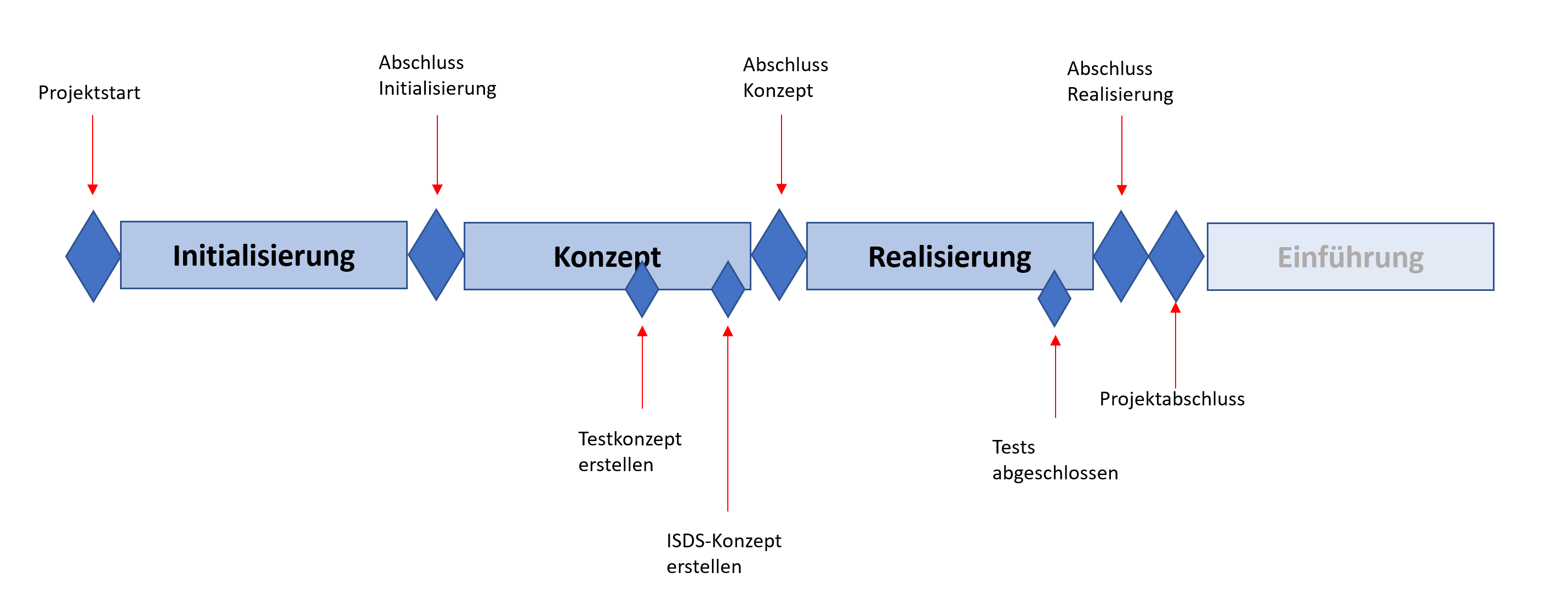


Abbildung : Hermes-Phasen

## Module

Auch hier fällt die Einführung wieder weg.

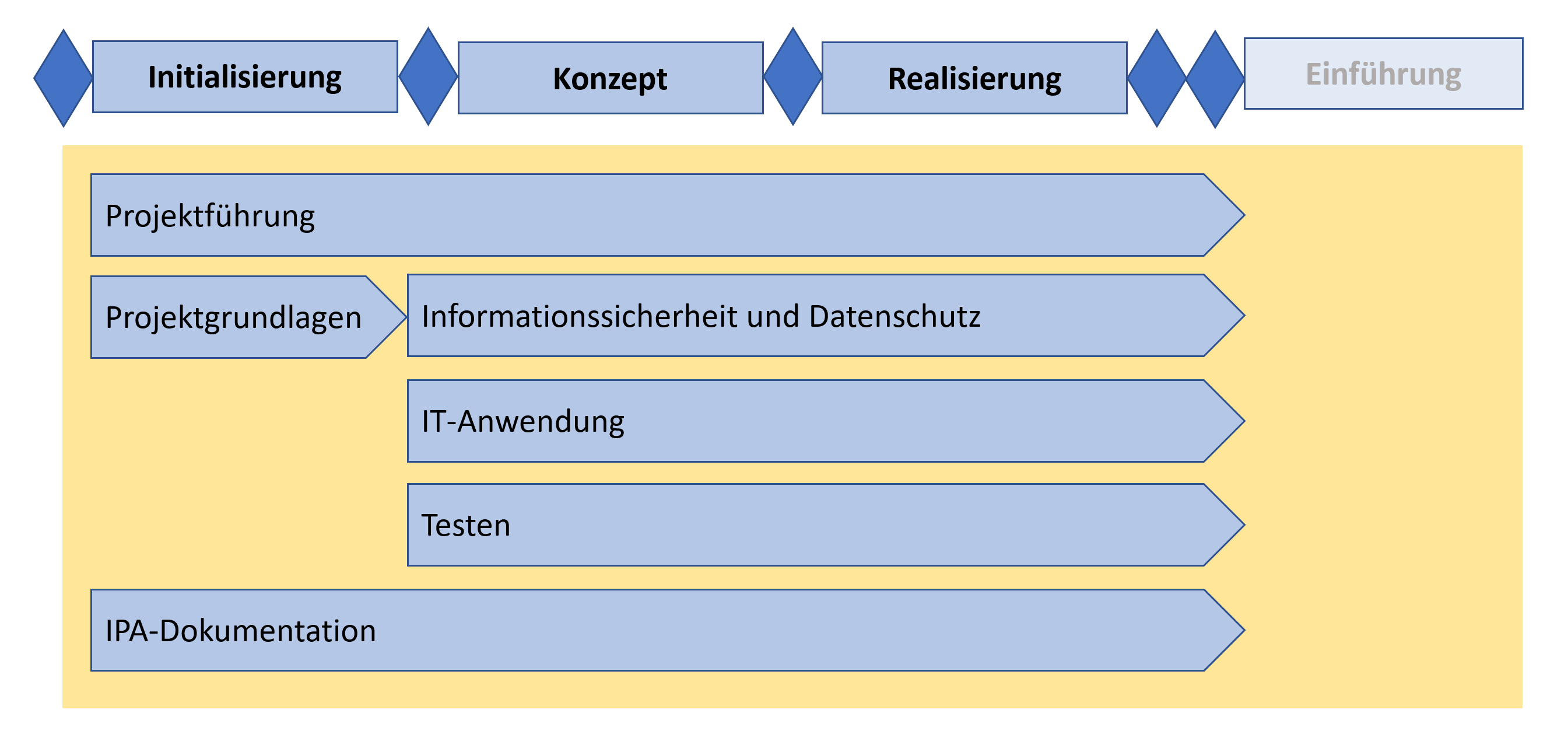


Abbildung : Hermes-Module

## 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

**Auftraggeber:**

Florin Barbisch

**Qualität- & Sicherheitsmanager**

Valid-Experte: Daniel Mosimann  
Hauptexperte: Daniel Mosimann  
2. Experte: -

**Projektleiter:**

Florin Barbisch

**Entwickler**

Florin Barbisch

**Tester**

Florin Barbisch

Projektrollen

|  |  |  |
| --- | --- | --- |
| Rolle | Belegt durch | Beschreibung |
| Auftraggeber | Florin Barbisch, Lernender | Schreibt die detaillierte Aufgabenstellung. |
| Valid-Experte | Daniel Mosimann, Lehrer | Gibt die Aufgabenstellung frei. |
| Hauptexperte | Daniel Mosimann, Lehrer | Korrigiert die Arbeit. Entscheidet bei Problemen über Massnahmen. |
| 2. Experte | Nicht belegt | Unterstützung des Hauptexperten. |
| Projektleiter | Florin Barbisch, Lernender | Führt, steuert und kontrolliert das Projekt. |
| Entwickler | Florin Barbisch, Lernender | Entwickelt die Applikation. |
| Tester | Florin Barbisch, Lernender | Führt Testfälle nach dem Testkonzept aus. |

Tabelle : Projektrollen

# Zeitplanung

## Zeitplan

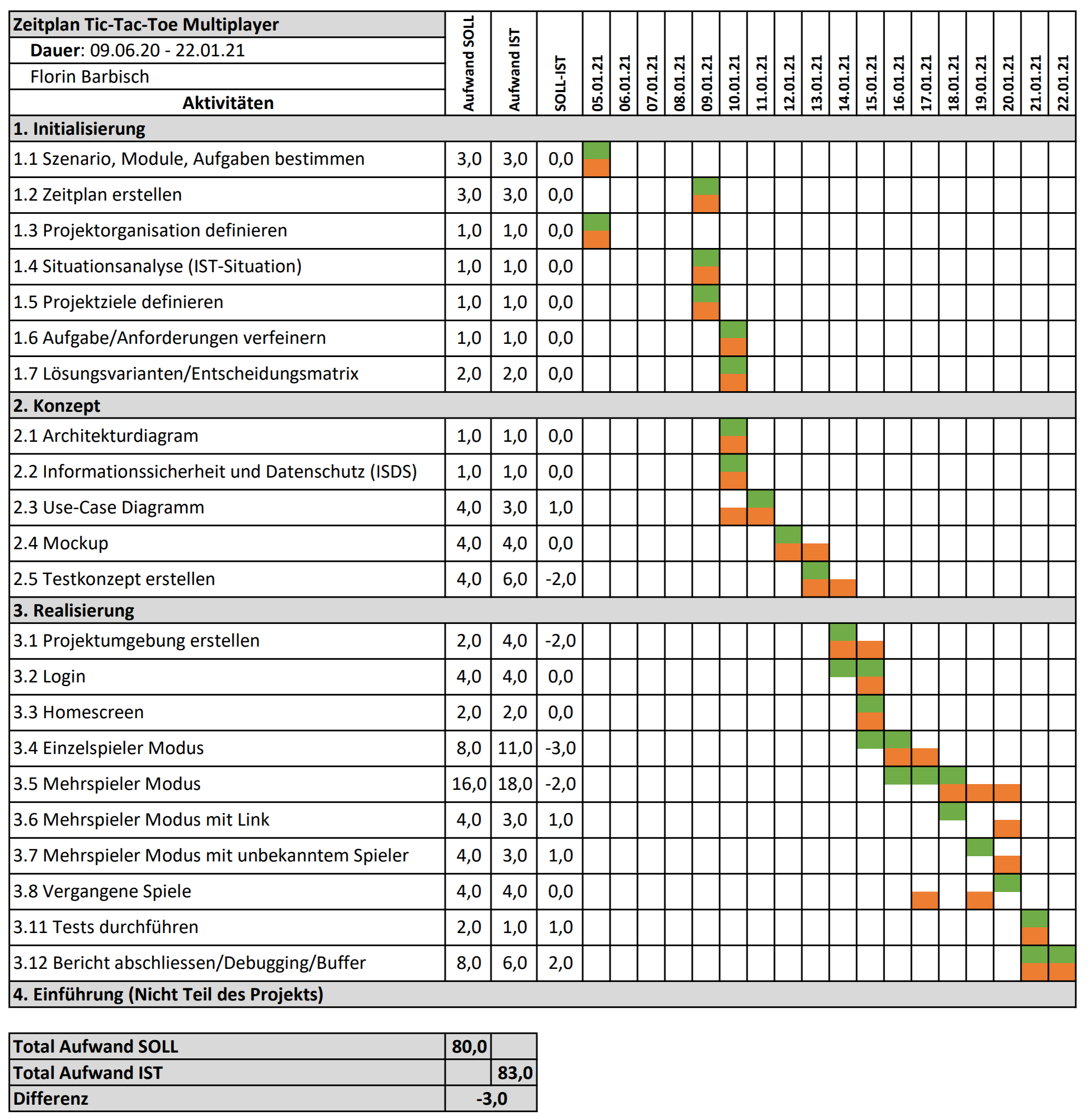


Abbildung : Zeitplan

# 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 dieses GitHub-Repository hochgeladen: <https://github.com/florinbarbisch/tic-tac-toe-multiplayer>

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

## Wiederherstellung

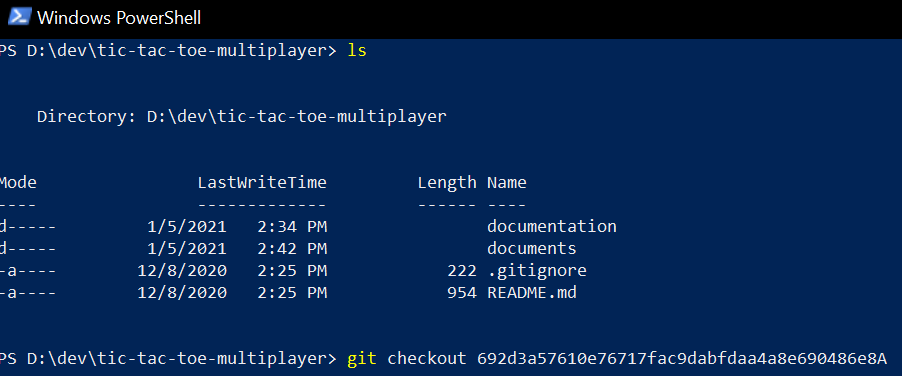


Abbildung : Daten wiederherstellen mit git checkout

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:

|  |  |
| --- | --- |
| Tool | Beschreibung |
| GitKraken[[1]](#footnote-1) | Git Client. |
| GitHub[[2]](#footnote-2) | Anbieter von Git-Versionierungsverwaltung für Softwareprojekte. |
| Webstorm[[3]](#footnote-3) | IDE welche sich auf Webentwicklung spezialisiert hat. |
| Visual Studio Code[[4]](#footnote-4) | IDE welche für alle möglichen Programmiersprachen verwendet werden kann. |
| draw.io[[5]](#footnote-5) | Webseite um alle Art von Skizzen/Diagrammen zu erstellen. |
| Bootstrapstudio[[6]](#footnote-6) | Wireframing-Tool zum entwickeln von UI-Mockups. Dabei wird direkt Bootstrap verwendet. So sehen die Mockups genau gleich aus wie das fertige GUI. |
| Carbon[[7]](#footnote-7) | Webseite, um Code-Snippets zu generieren und zu formatieren. |
| Zotero[[8]](#footnote-8) | Quellenverwaltungstool mit Integration im Browser und in Word. Ein Quellenverzeichnis kann auch automatisch generiert werden. |

Tabelle : Verwendete CASE-Tools

# 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 | | | |

Tabelle : Arbeitsjournaleintrag KW24

## 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 zu dem 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. | | | |

Tabelle : Arbeitsjournaleintrag KW25

## 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. | | | |

Tabelle : Arbeitsjournaleintrag KW26

## 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 | | | |

Tabelle : Arbeitsjournaleintrag KW27

## 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. | | | |

Tabelle : Arbeitsjournaleintrag KW33

## 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. | | | |

Tabelle : Arbeitsjournaleintrag KW34

## KW 35: 24.08.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| ***Total*** | | ***0*** | ***0*** |
| **Tagesablauf** | | | |
| Heute hatte ich Rekrutierung und war nicht im Unterricht anwesend. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
|  | | | |

Tabelle : Arbeitsjournaleintrag KW35

## 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, E-Mail 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. | | | |

Tabelle : Arbeitsjournaleintrag KW36

## 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. | | | |

Tabelle : Arbeitsjournaleintrag KW37

## 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. | | | |

Tabelle : Arbeitsjournaleintrag KW38

## 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. | | | |

Tabelle : Arbeitsjournaleintrag KW39

## 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. | | | |

Tabelle : Arbeitsjournaleintrag KW43

## 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  Es gibt sehr viele Vorgaben und diese sind sehr unübersichtlich strukturiert. | | | |

Tabelle : Arbeitsjournaleintrag KW44

## KW 45: 02.11.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | | |

Tabelle : Arbeitsjournaleintrag KW45

## KW 46: 09.11.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **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. | | | |

Tabelle : Arbeitsjournaleintrag KW46

## KW 47: 16.11.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| VPN/RDP konfigurieren | Initialisierung | 1.5 | 1 |
| ***Total*** | | ***1.5*** | ***1*** |
| **Tagesablauf** | | | |
| Damit ich die Software auf dem GIBS Rechner nicht jedes Mal neu installieren muss, wenn die Schule die PCs neu aufsetzt, habe ich bei mir zu Hause ein VPN und RDP eingerichtet. Ich kann mich somit in der Schule auf meinen privaten Rechner verbinden. Schade ist nur, dass dies nicht auch auf meinen Laptop über das Schul-WLAN geht. Denn ich kann keine VPN Verbindung im Schul-WLAN aufbauen. Was das bringt/soll ist mir unklar. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Meine Erkenntnisse von heute  Ich kann jetzt unabhängig davon, was die Schule anstellt, auch in der Schule arbeiten. | | | |

Tabelle : Arbeitsjournaleintrag KW47

## KW 48: 23.11.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Bewertungskriterien studieren | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Ich habe mir heute die Bewertungskriterien dieses Moduls angesehen. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Meine Erkenntnisse von heute  Mir ist klar geworden, dass der Code der Anwendung, respektive die Anwendung selbst, niemanden so richtig interessiert oder bewertet wird. Es wird vielmehr die Dokumentation bewertet. Und was nicht dokumentiert ist, kann nicht bewertet werden. | | | |

Tabelle : Arbeitsjournaleintrag KW48

## KW 49: 30.11.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| IPA Vorstellung von Tschumi | Initialisierung | 1 | 1 |
| Austausch über die IPA mit meinen Kollegen | Initialisierung | 0 | 1 |
| ***Total*** | | ***1*** | ***2*** |
| **Tagesablauf** | | | |
| Heute hat uns Dominik Tschumi die IPA vorgestellt, und wir hatten die Möglichkeit Fragen zu stellen. Danach habe ich mich noch mit meinen Kollegen über die IPA ausgetauscht, da diese vom Betrieb her schon eine Test-IPA hatten und ich nicht. | | | |
| **Wissensbeschaffung** | | | |
| Tschumi  Schulkollegen | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Meine Erkenntnisse von heute  Ich habe einen Überblick über die IPA erhalten und weiss jetzt wie dieses Ablaufen wird. | | | |

Tabelle : Arbeitsjournaleintrag KW49

## KW 50: 07.12.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Austausch über die IPA mit meinen Kollegen | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Ich habe mich heute noch einmal mit Kollegen über ihre Erfahrungen mit Test-IPAs ausgetauscht. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Meine Erkenntnisse von heute  Ich weiss jetzt mehr, wie die IPA in der Praxis Ablaufen wird. | | | |

Tabelle : Arbeitsjournaleintrag KW50

## KW 51: 14.12.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Austausch über die IPA mit meinen Kollegen | Initialisierung | 1.5 | 1.5 |
| ***Total*** | | ***1.5*** | ***1.5*** |
| **Tagesablauf** | | | |
| Ich habe mich heute, gleich wie letztes und vorletztes Mal, noch einmal mit Kollegen über ihre Erfahrungen mit Test-IPAs ausgetauscht. | | | |
| **Wissensbeschaffung** | | | |
|  | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Meine Erkenntnisse von heute  Ich weiss jetzt mehr, wie die IPA in der Praxis Ablaufen wird. | | | |

Tabelle : Arbeitsjournaleintrag KW51

## KW 01 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 | 1 | 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. | | | |

Tabelle : Arbeitsjournaleintrag KW01

## KW 02 11.01.2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Use-Case Diagramme | Konzept | 3 | 2 |
| Mockup | Konzept | 4 | 4 |
| Testkonzept erstellen | Konzept | 4 | 6 |
| Projektumgebung erstellen | Realisierung | 2 | 4 |
| Login | Realisierung | 4 | 4 |
| Homescreen | Realisierung | 2 | 2 |
| Einzelspieler Modus | Realisierung | 8 | 11 |
| Vergangene Spiele | Realisierung | 0 | 2 |
| Mehrspieler Modus | Realisierung | 6 | 0 |
| ***Total*** | | ***33*** | ***35*** |
| **Tagesablauf** | | | |
| Ich habe diese Woche die Konzeptphase beendet. Beim Testkonzept habe ich ein bisschen länger gebraucht, da ich noch nie ein Testkonzept erstellt habe musste ich mich zuerst informieren.  Ich habe beim Erstellen der Projektumgebung länger gebraucht als geplant. Danach bin ich aber gut vorangekommen mit dem Erstellen des Homescreen und der Login. Ein bisschen mehr Zeit brauchte ich beim Erstellen des Einzelspieler Modus, vor allem beim Implementieren der Logik des Computers. Ich habe auch schon den Teil «Vergangene Spiele» für den Einzelspielermodus implementiert. Mit dem Mehrspieler Modus konnte ich nicht wie geplant anfangen. | | | |
| **Wissensbeschaffung** | | | |
| <https://www.hettwer-beratung.de/konzepte/testkonzept/>  <https://expressjs.com/en/4x/api.html>  <https://nodejs.org/en/docs/>  <https://vuex.vuejs.org/>  <https://mongoosejs.com/docs/guide.html> | | | |
| **Hilfestellungen** | | | |
| - | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich hatte nach dem Abschliessen der Konzeptphase ein gutes Gefühl, dass ich vorbereitet bin, für die Realisierung. Gut ist auch, dass ich trotz Zeitverzögerungen mit dem Einzelspielermodus fertig wurde.  Was lief weniger gut?  Es gab diverse Zeitverzögerungen, mit welchen ich nicht gerechnet habe. Ich konnte deshalb nicht mit dem Mehrspielermodus anfangen.  Meine Erkenntnisse von heute  Ich habe diese Woche gemerkt, dass ein Konzept zwar im Moment, in dem man es erstellt nicht wirklich hilfreich ist, aber während der Umsetzung hilfreich ist, da alles schon genau definiert worden ist. | | | |

Tabelle : Arbeitsjournaleintrag KW02

## KW 03 18.01.2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Tätigkeiten** | **Projektphase** | **Zeit SOLL** | **Zeit IST** |
| Mehrspieler Modus | Realisierung | 16 | 18 |
| Mehrspieler Modus mit Link | Realisierung | 4 | 3 |
| Mehrspieler Modus mit unbekanntem Spieler | Realisierung | 4 | 3 |
| Vergangene Spiele | Realisierung | 2 | 2 |
| Tests durchführen | Realisierung | 2 | 1 |
| Bericht abschliessen | Realisierung | 8 | 6 |
| ***Total*** | | ***36*** | ***33*** |
| **Tagesablauf** | | | |
| Diese Woche war die letzte Woche, also Endspurt. Ich habe mit dem Mehrspieler Modus angefangen. Dabei habe ich zwei Stunden mehr gebraucht als geplant. Ich habe mit dem Implementieren von socket.io viel Zeit verloren, weil ich zuerst nicht begriffen habe, dass ich CORS selbst implementieren muss. Diese verlorene Zeit, habe ich aber bei den nachfolgenden Tasks «Mehrspieler Modus mit Link» und «Mehrspieler Modus mit unbekanntem Spieler» wieder aufgeholt. Diese Tasks gingen schneller als gedacht, da ich mittlerweile mit NodeJS mehr Erfahrungen habe. Die Tests durchführen ging auch schneller als gedacht, da kein einziger fehlschlug. Als letztes habe ich noch Zeit aufgewendet, um den Bericht abzuschliessen, ich musste noch verschiedene Sachen machen, wie z.B. Zeitplan, Kurzfassung, Schlusswort, Quellen, Tabellenbeschriftungen etc. | | | |
| **Wissensbeschaffung** | | | |
| <https://expressjs.com/en/4x/api.html>  <https://nodejs.org/en/docs/>  <https://socket.io/get-started/chat/>  <https://socket.io/docs/v3/handling-cors/> | | | |
| **Hilfestellungen** | | | |
|  | | | |
| **Reflexion** | | | |
| Was lief gut?  Ich konnte das ganze Mehrspieler Modul in der gegebenen Zeit umsetzen. Auch die Tests sind gut gelaufen, liegt wahrscheinlich daran, dass ich nach dem Implementieren immer kurz die implementierte Logik teste.  Was lief weniger gut?  Ich hatte viel Zeit verloren mit dem Identifizieren des Problems, dass ich bei socket.io CORS explizit selbst einstellen muss.  Meine Erkenntnisse von heute  Diese Woche habe ich das Projekt mit einem guten Gefühl abgeschlossen. Ich war froh, dass ich mir Zeit genommen habe, und die Dokumentation durchgegangen bin. Ich musste dies auch nicht mit einem Stressgefühl machen, da ich genügend Zeit dafür eingeplant hatte. | | | |

Tabelle : Arbeitsjournaleintrag KW03

# Teil 2: Projektdokumentation

# Kurzfassung

## Ausgangssituation

Die Tic-Tac-Toe-Webseite ist mit PHP umgesetzt. Diese Technologie und die Art wie die Webseite aufgebaut ist, sind nicht mehr zeitgemäss. Das hat zur Folge, dass immer wenige Personen diese Webseite warten können. So ist es sinnvoll, die Applikation lieber früher als zu spät zu ersetzen.

Zwar verfügt die Webseite über einen Mehrspielermodus, aber dieser funktioniert nur lokal. Dies ist insbesondere in der Zeit einer Pandemie ein Nachteil. Weiters soll die Identität der Spieler sichergestellt werden können.

Diese drei Punkte sollen in der IPA behoben werden.

## Umsetzung

Die bestehende Webseite wurde durch eine neue Webseite ersetzt. Diese wurde mit Javascript geschrieben. Dazu wurden zwei weit verbreitete und unterstütze Technologien (NodeJS und VueJS) verwendet. Die HTTP-Schnittstelle wurde nach dem aktuellen Standard REST umgesetzt.

Der Mehrspielermodus funktioniert jetzt auf zwei unterschiedlichen Rechnern. Ohne dass der Benutzer die Webseite aktualisieren muss.

Die Identität des Spielers wird über ein Login sichergestellt. Dieses Login verläuft über den Standard JWT ab.

## Ergebnis

Die Webseite wurde durch eine moderne und flexiblere Technologie abgelöst. Die zusätzlichen Funktionen (Mehrspieler auf zwei Computer und Benutzeridentifizierung) wurden mit zeitgemässen Technologien implementiert und funktionieren einwandfrei.

Das Projekt wurde, im Rahmen des Zeitbudget von 85 Stunden, abgeschlossen. Es sind alle definierten Tests erfolgreich.

# 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 Namen und Schwierigkeitsgrad).

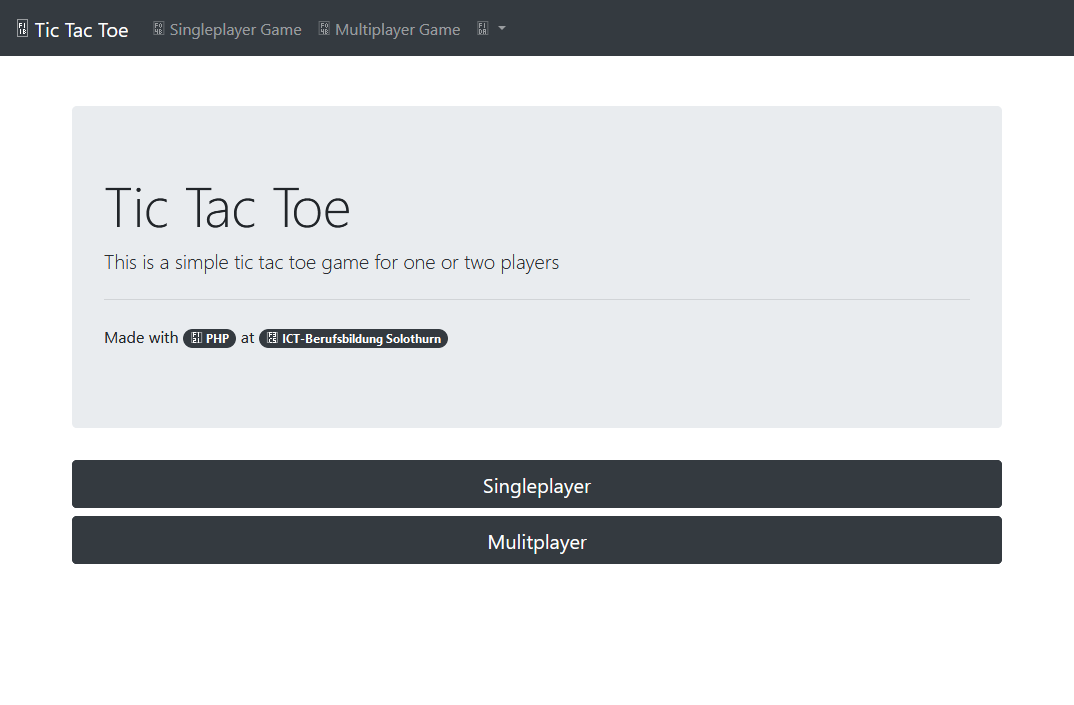


Abbildung : Homescreen der alten Website

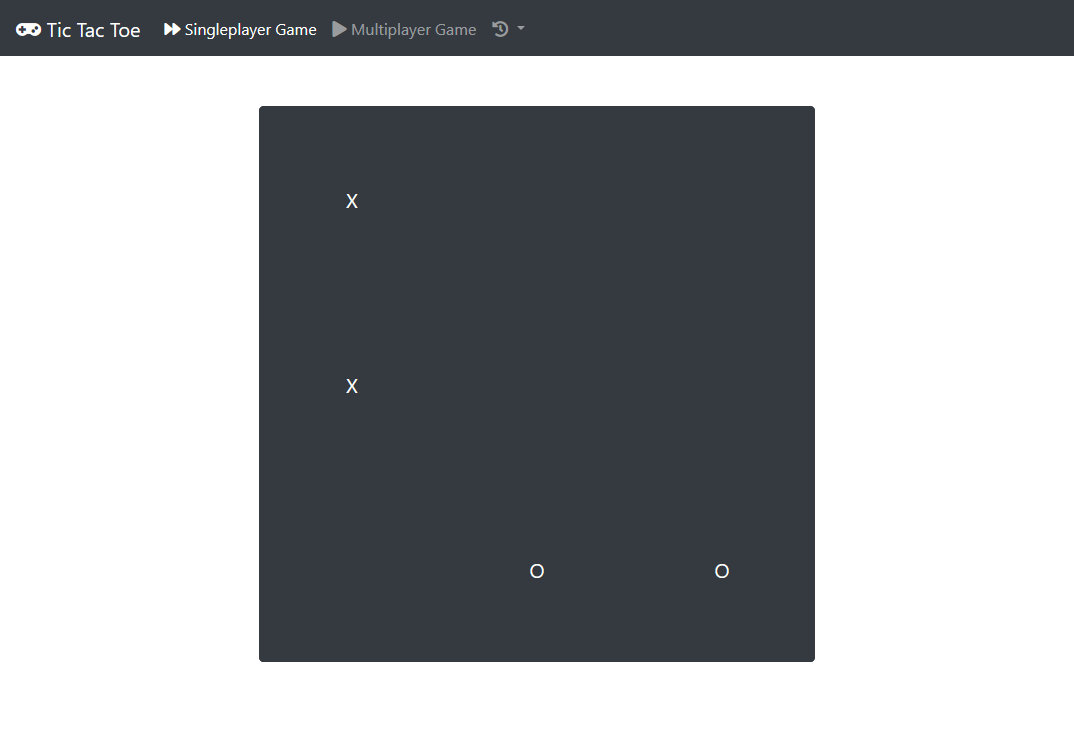


Abbildung : Während eines Einzelspiels auf der alten Webseite

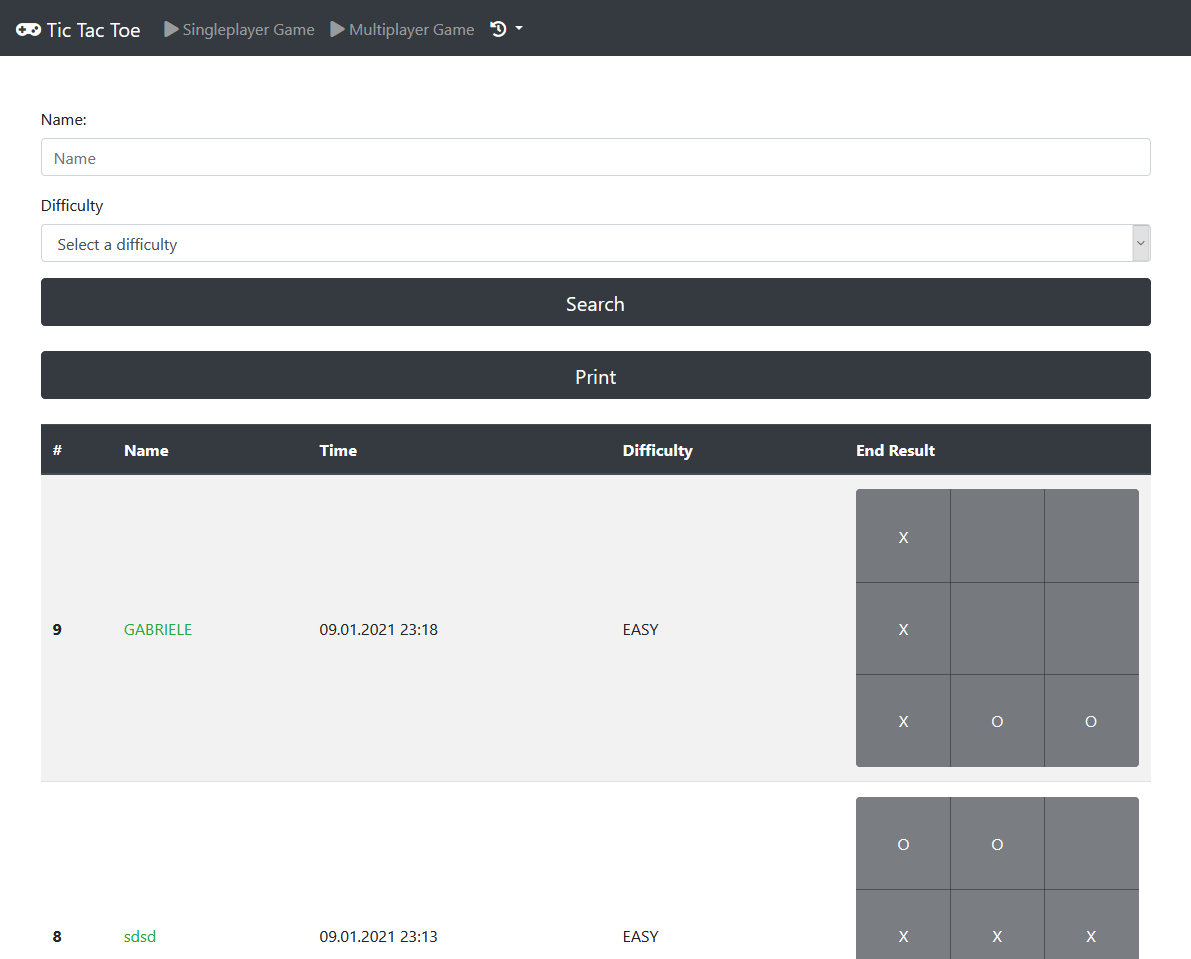


Abbildung : Suche nach vergangenen Einzelspielen auf der alten Webseite

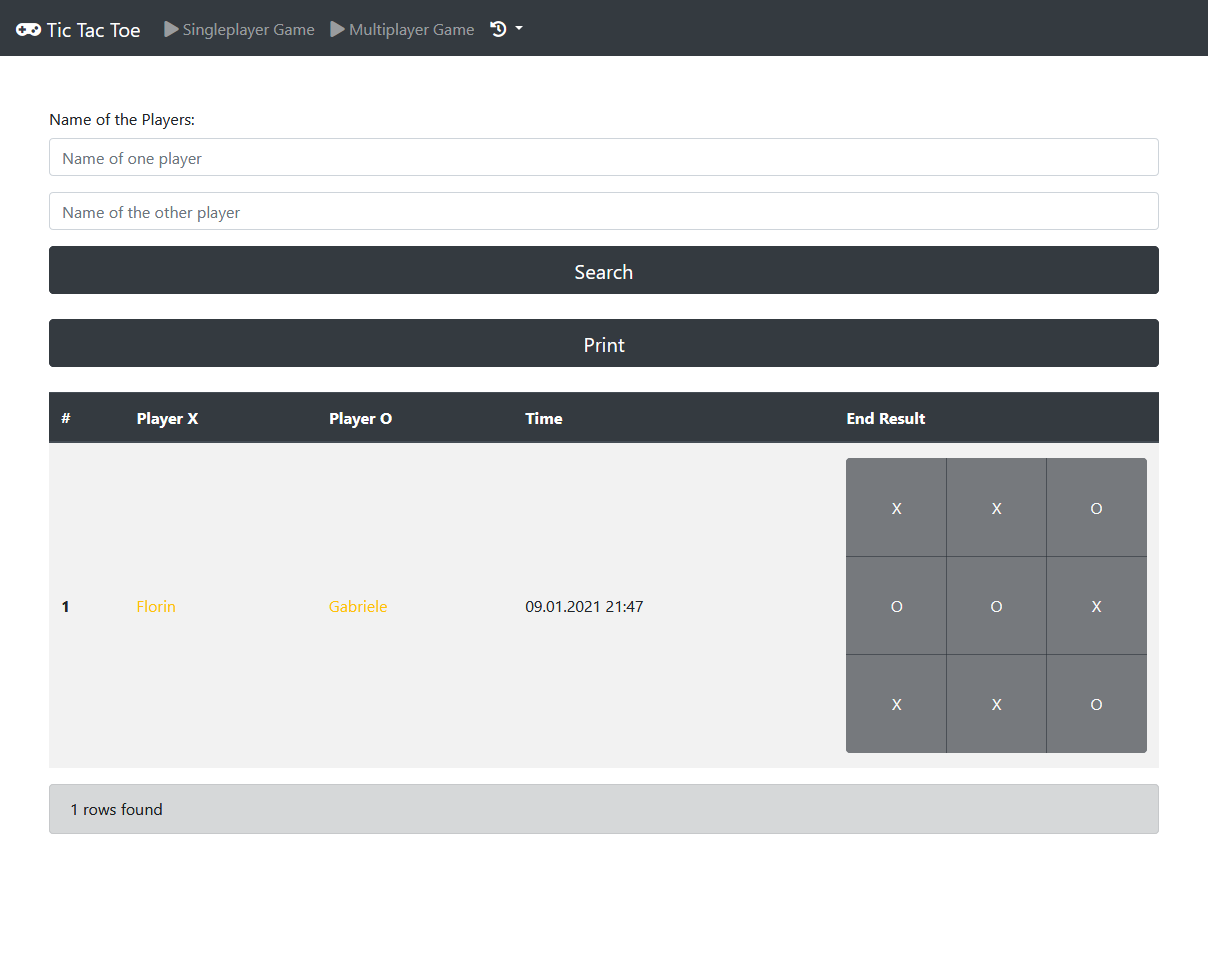


Abbildung : Suche nach Mehrspielerspielen auf der alten Webseite

### 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[[9]](#footnote-9) umgesetzt
* Das Frontend ist mit VueJS umgesetzt
* Als Datenbank wird MongoDB[[10]](#footnote-10) 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 (Socket-Kommunikation[[11]](#footnote-11)).

### Variante 1

Als erste Variante habe ich mir WebSocket[[12]](#footnote-12) angeschaut. WebSocket ist das Protokoll welche es erlaubt eine bidirektionale Verbindung zwischen dem Server und dem Client. Dies erfolgt ohne grossen Overhead. 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[[13]](#footnote-13). 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

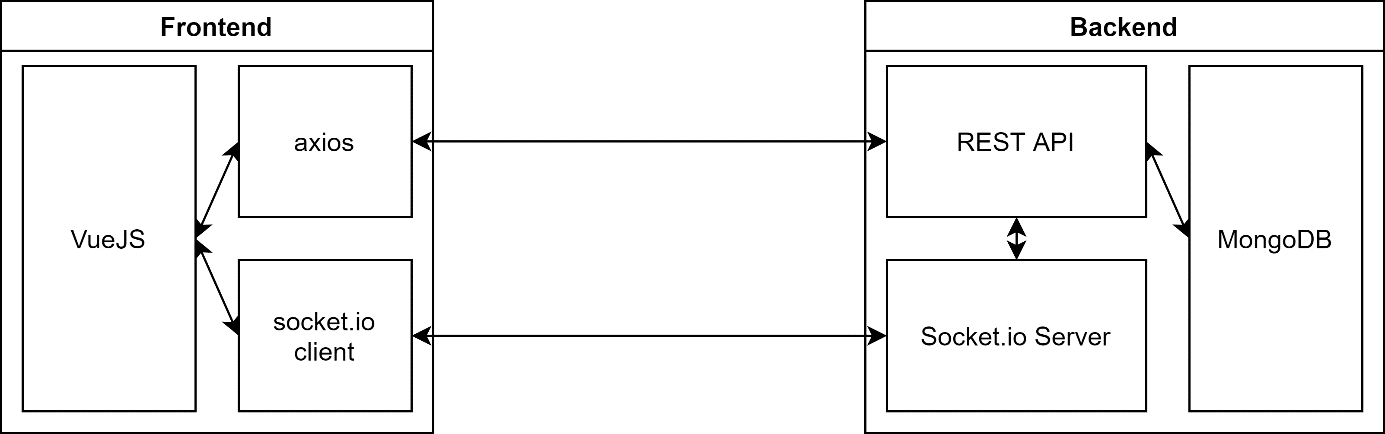


Abbildung : Architekturdiagramm

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

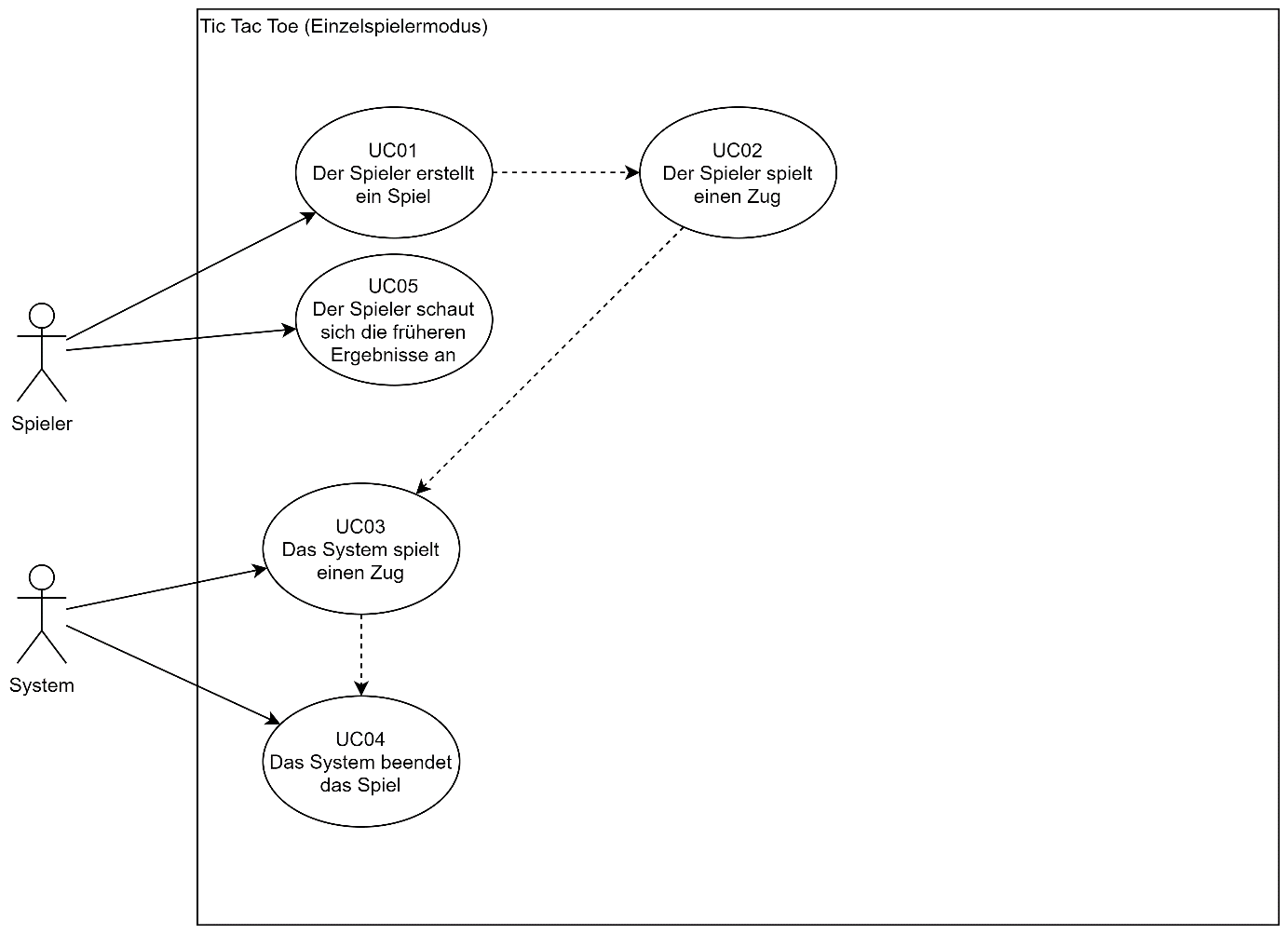


Abbildung : Use-Cases Einzelspieler

#### 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 |

Tabelle : Use Case 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: | - |

Tabelle : Use Case UC02

#### 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: |  |

Tabelle : Use Case UC03

#### 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: |  |

Tabelle : Use Case UC04

#### 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: |  |

Tabelle : Use Case UC05

### Mehrspieler

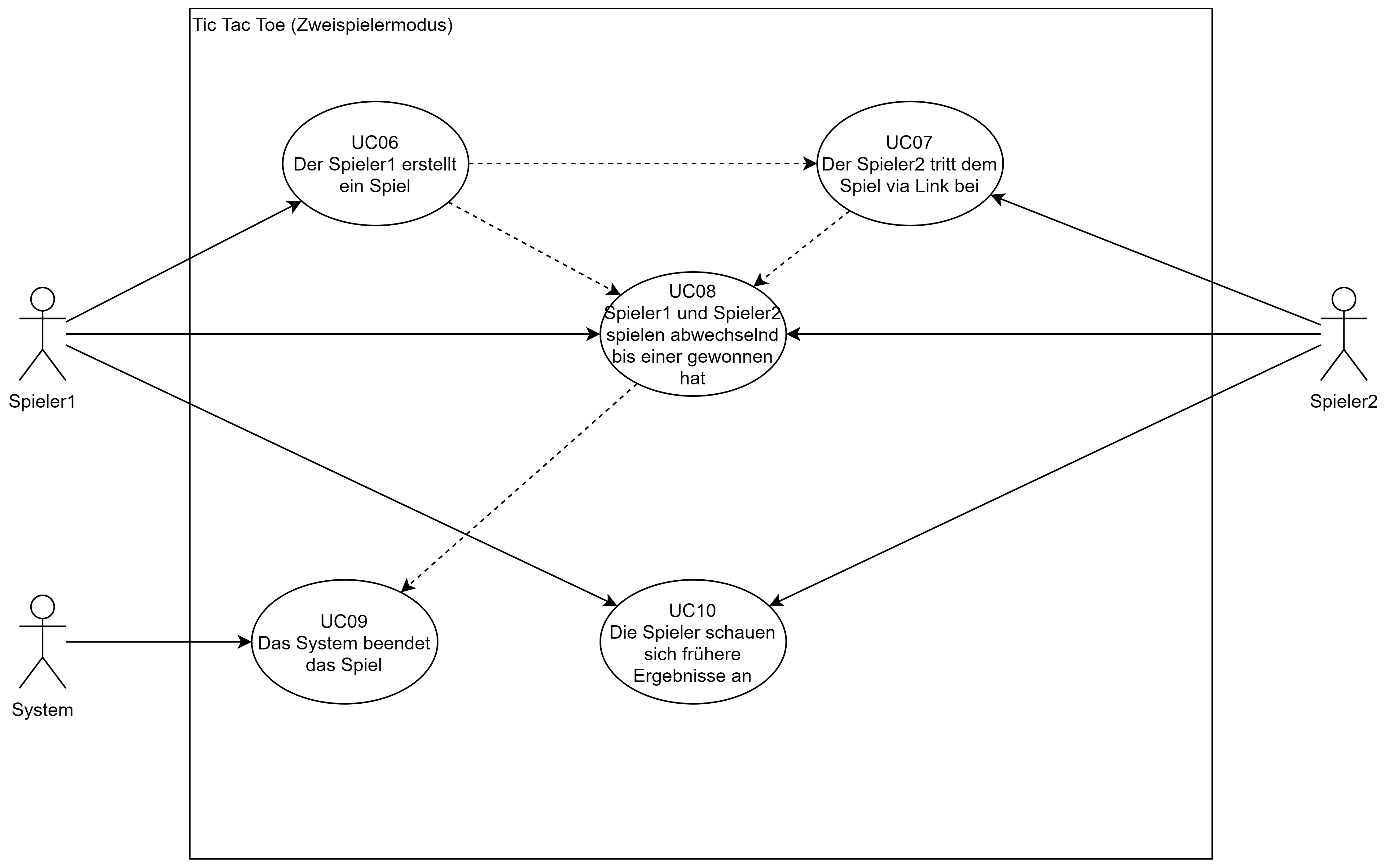


Abbildung : Use-Cases Mehrspieler

#### 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: |  |

Tabelle : Use Case UC06

#### UC07

|  |  |
| --- | --- |
| ö | 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: |  |

Tabelle : Use Case UC07

#### 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. |

Tabelle : Use Case UC08

#### 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 |

Tabelle : Use Case UC09

#### 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: |  |

Tabelle : Use Case UC10

### Login

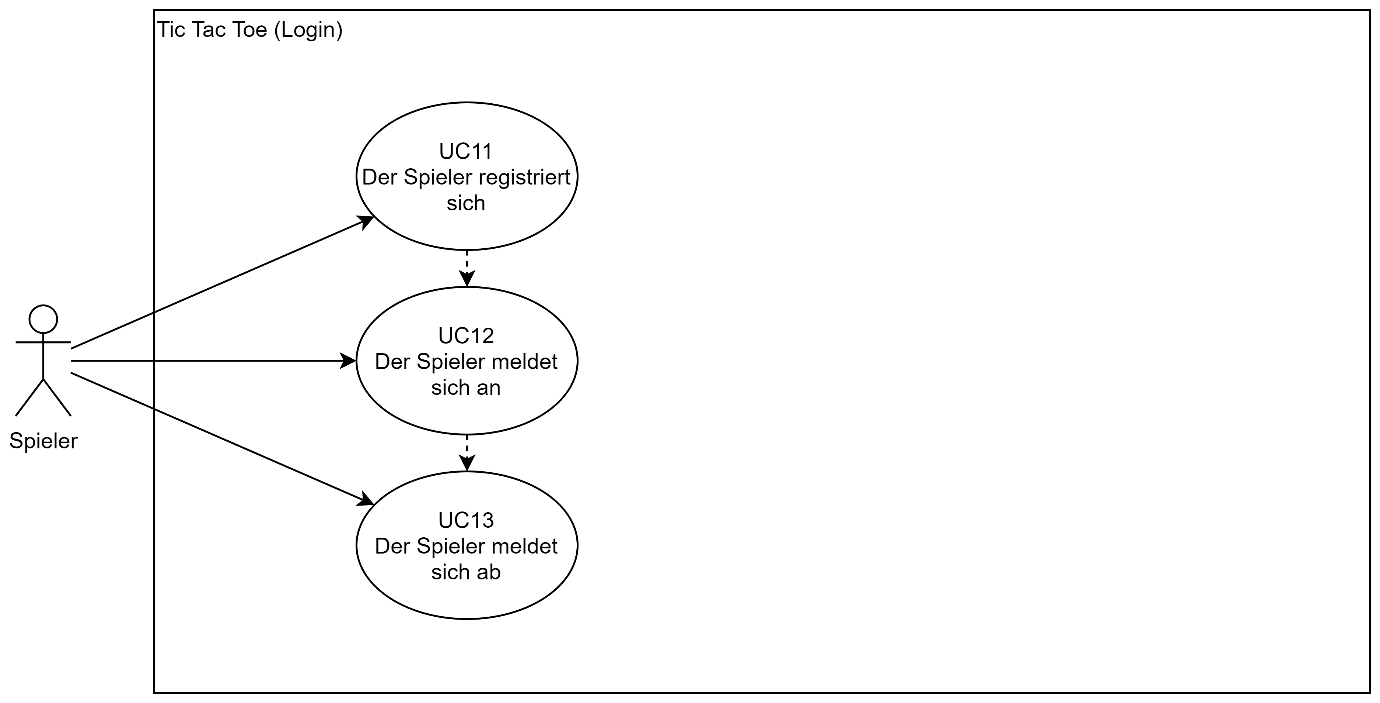


Abbildung : Use-Cases Login

#### 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. |

Tabelle : Use Case UC11

#### 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: |  |

Tabelle : Use Case UC12

#### 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: |  |

Tabelle : Use Case UC13

## UI-Mockups

### Allgemein

#### Startseite eingeloggt

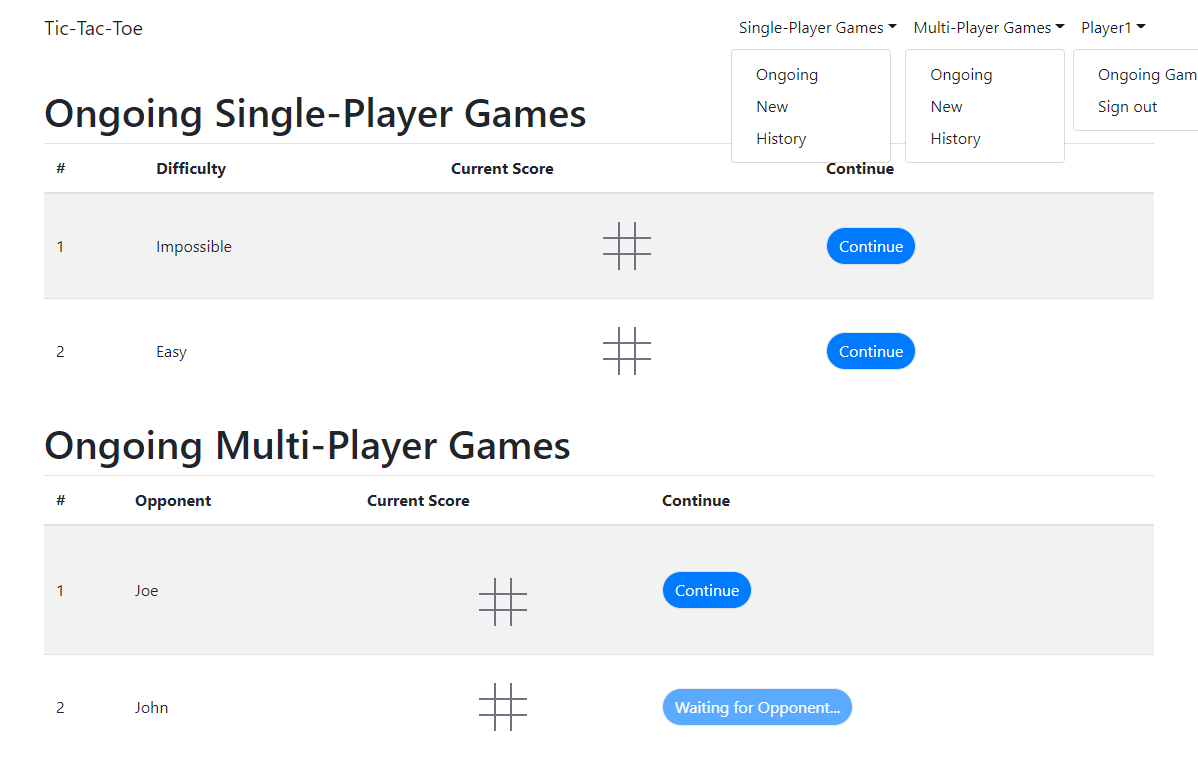


Abbildung : UI-Mockup Startseite eingeloggt

#### Sign up

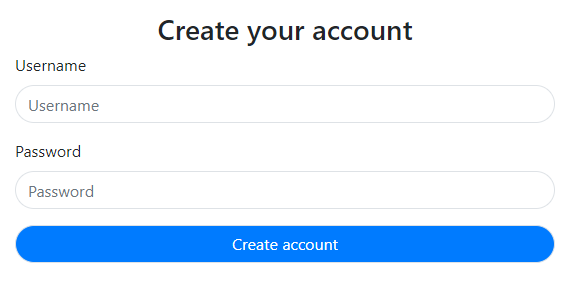


Abbildung 15: UI-Mockup Sign up

#### Sign in



Abbildung : UI-Mockup Sign in

### Einzelspieler

#### Einzelspielerspiel erstellen

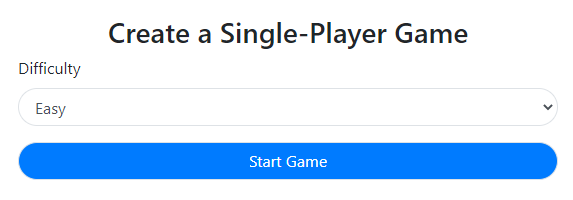


Abbildung : UI-Mockup Einzelspielerspiel erstellen

#### Einzelspielerspiel spielen

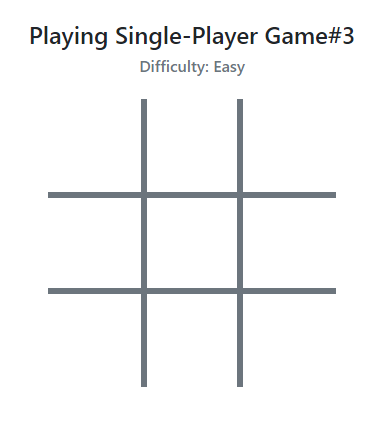


Abbildung : UI-Mockup Einzelspielerspiel spielen

#### Aktuelle Einzelspielerspiele

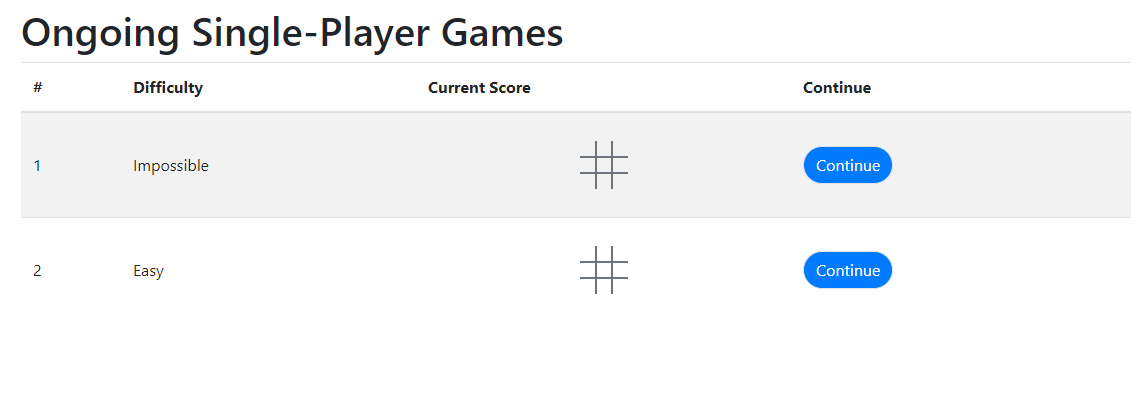


Abbildung : UI-Mockup Aktuelle Einzelspielerspiele

#### Einzelspielerspiel Ergebnis

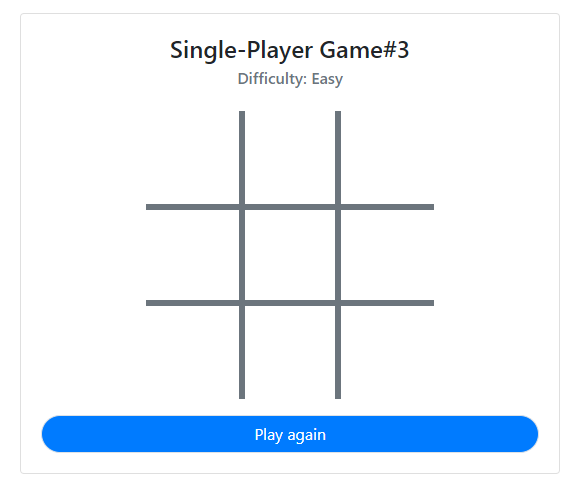


Abbildung : UI-Mockup Einzelspieler Ergebnis

#### Vergangene Einzelspielerspiele

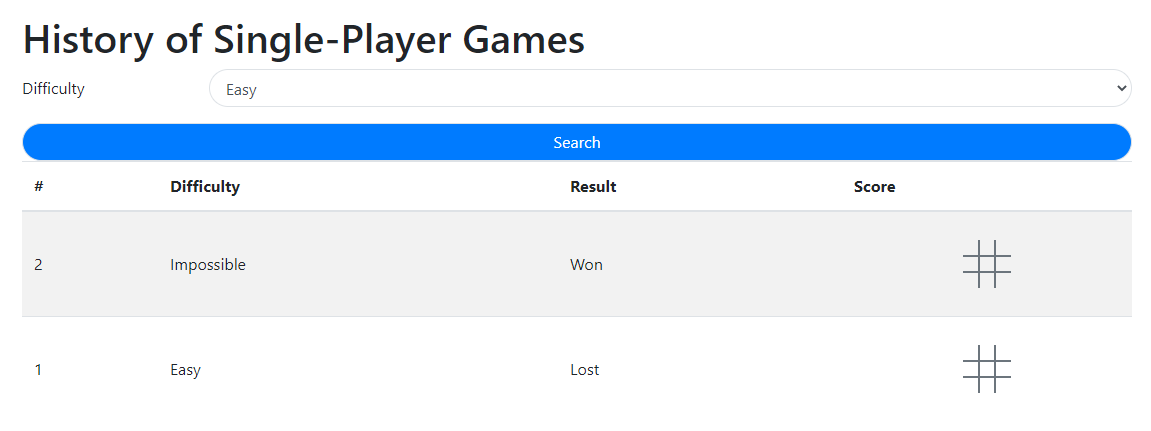


Abbildung : UI-Mockup Vergangene Einzelspielerspiele

### Mehrspieler

#### Mehrspielerspiel erstellen

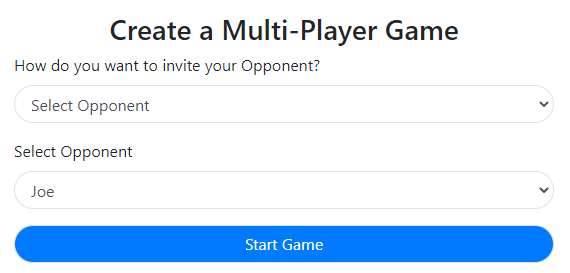


Abbildung : UI-Mockup Mehrspielerspiel erstellen

#### Mehrspielerspiel spielen

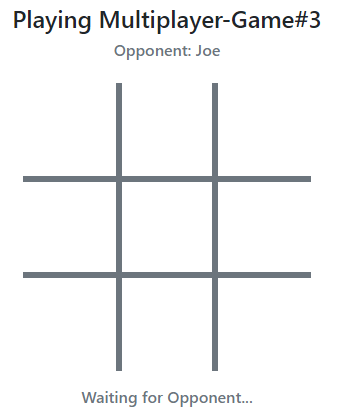


Abbildung : UI-Mockup Mehrspielerspiel spielen

#### Aktuelle Mehrspielerspielspiele

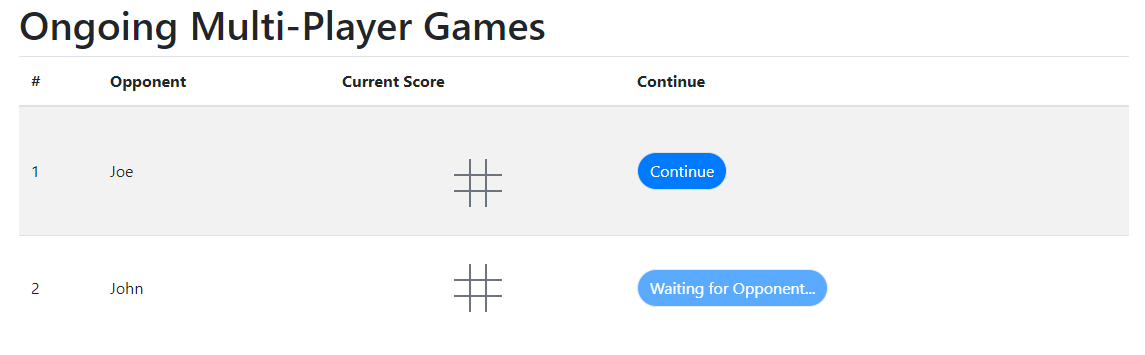


Abbildung : UI-Mockup Aktuelle Mehrspielerspiele

#### Mehrspielerspiel Ergebnis

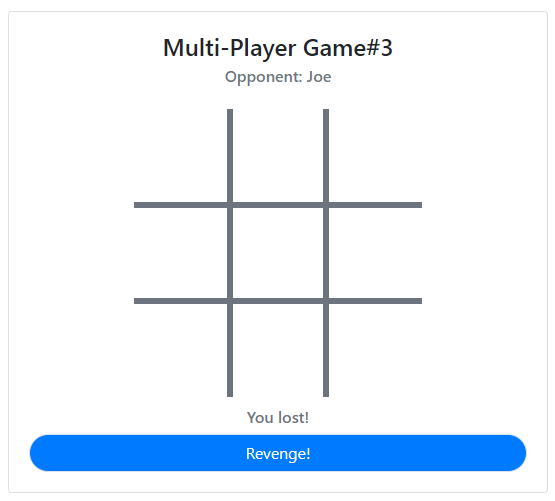


Abbildung : UI-Mockup Mehrspielerspiel Ergebnis

#### Vergangene Mehrspielerspiele

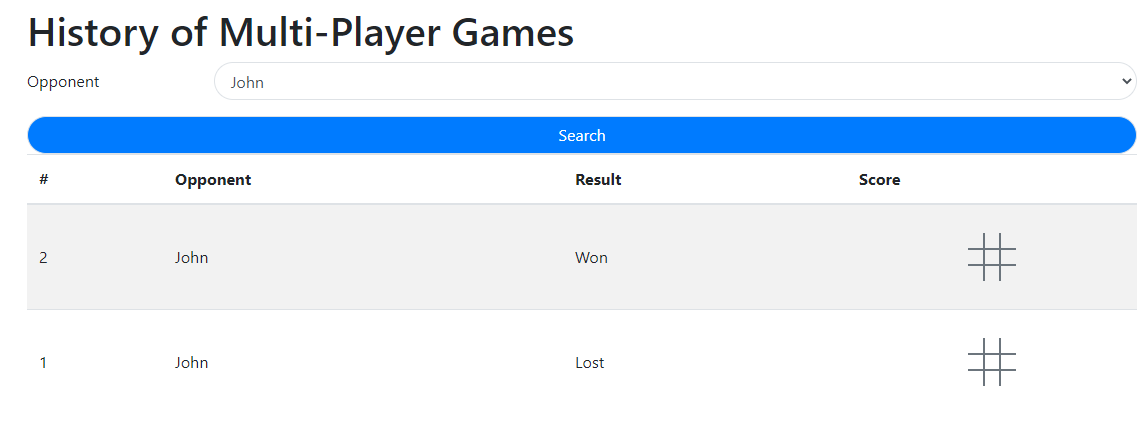


Abbildung : UI-Mockup Vergangene Mehrspielerspiele

## 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[[14]](#footnote-14)). 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[[15]](#footnote-15) und Blackbox-Testing[[16]](#footnote-16) 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 42: 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 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 «Sign in»-Seite 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. |
| **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. |
| **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.

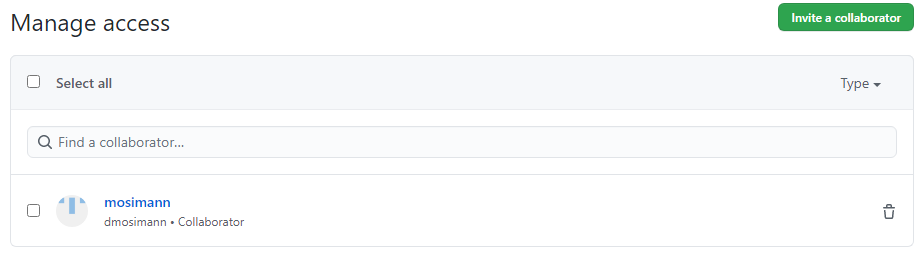


Abbildung : Berechtigungen des GitHub-Repository

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

# Realisierung

## System realisieren

### REST

Das Backend wurde, wie nach Vorgabe, nach REST implementiert. Da der Server mit NodeJS programmiert wurde, hat sich der Router von expressJS[[17]](#footnote-17) am besten angeboten.



Abbildung 29: GET-Ressource in expressJS

In der Abbildung oben ist eine GET-Ressource dargestellt, mit der ein Einzelspielerspiel geholt werden kann. Mit auth.required wird sichergestellt, dass der Benutzer authentifiziert ist.

Der Parameter :game wird in ein Objekt umgewandelt, bevor der Code in der Ressource ausgeführt wird. Dies hat zum Vorteil, dass die Attribute und Methoden des MongoDB-Schemas direkt verwendet werden können. Die Methode, welche das Objekt lädt, sieht wie folgt aus:



Abbildung : Umwandeln eines Parameters zu einem Objekt

Um die Ressourcen besser zu gruppieren wurden Controller verwendet. Im System gibt es folgende drei Controller:

* users (/api/users)
* singleplayergames (/api/singleplayergames)
* multiplayergames (/api/multiplayergames)

In den Controllern ist jeweils nicht viel Logik implementiert (wie zum Beispiel überprüfen, wer der Gewinner ist, oder das Objekt in JSON umwandeln). Diese Logik wird auf das Objekt/Schema ausgelagert.

Dies sieht dann wie folgt im Controller aus:

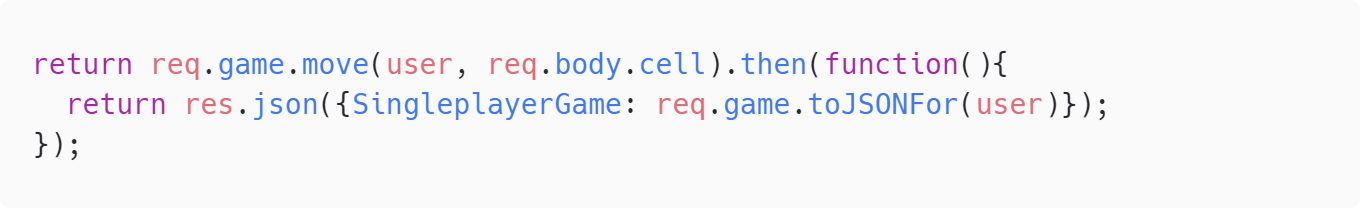


Abbildung : Spielzug vom Controller aus ausführen

Die Methode Singleplayer.move(user, cell) wird dann im Schema implementiert:



Abbildung : Logik von Spielzug ausführen

### Logik des Computerspieler

Das System erhält eine Funktion, in der man gegen einen Computer spielen kann. Hier musste eine Logik implementiert werden, welche die drei Schwierigkeitsgrade («Impossible», «Medium» und «Easy») berücksichtigt.

Es wurde deshalb eine Reihe von verschiedenen Schritten definiert, welche nacheinander versucht werden auszuführen.

Dies läuft folgendermassen ab:

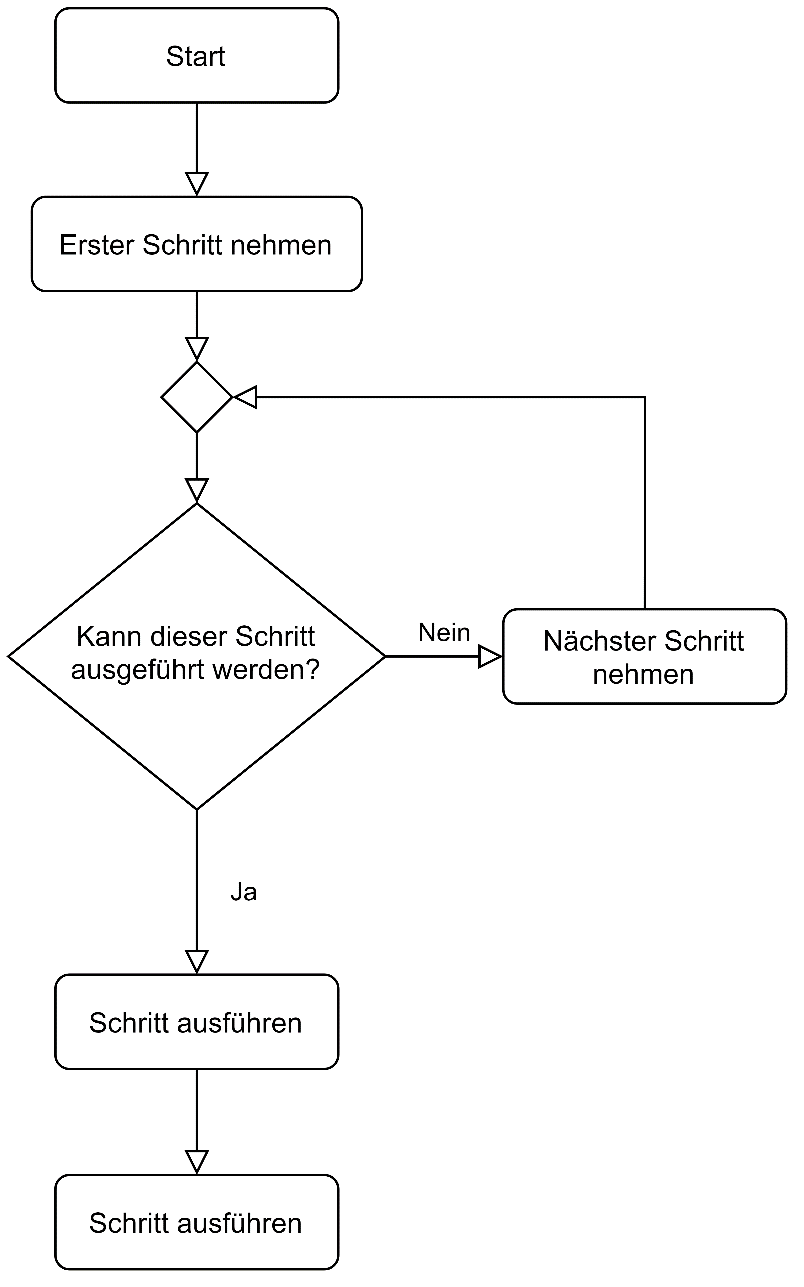


Abbildung : Ablauf des Computerspielers

In folgender Tabelle ist ersichtlich welche Schritte bei welchem Schwierigkeitsgrad ausgeführt werden (Der Computer ist ‘O’ und der Spieler ‘X’):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reihen-folge | Schritt | Impossible | Medium | Easy |
| 1. | Sind zwei ‘O’s in einer Reihe => die Reihe mit einem ‘O’ füllen | X |  |  |
| 2. | Sind zwei ‘X’s in einer Reihe => die Reihe mit einem ‘O’ blockieren | X |  |  |
| 3. | Ist in einer Reihe ein ‘O’ => ein weiteres ‘O’ in diese Reihe setzen | X | X |  |
| 4. | Ist die Mitte frei => ein ‘O’ in die Mitte setzen | X | X | X |
| 5. | Ist in einer Ecke ein ‘X’ und die gegenüberliegende Ecke frei => in die gegenüberliegende Ecke ein ‘X’ setzen | X | X | X |
| 6. | Ist eine Ecke frei => in die Ecke ein ‘X’ setzen | X | X | X |
| 8 | Ein ‘O’ an eine freie Seite setzen. | X | X | X |

Tabelle : Schritte des Computerspielers

Diese Reihenfolge stellt sicher, dass der Spieler im Schwierigkeitsgrad «Impossible» nicht gewinnen kann.

### Socket.io

Im System kann Tic-Tac-Toe gegeneinander gespielt werden. Dies kann auf zwei verschiedenen Rechnern passieren. Damit die Spieler die Webseite nicht alle 5 Sekunden neu laden müssen, werden Updates via Socket-Kommunikation versendet. Als Library wird dazu socket.io verwendet.

Der Webserver kann einfach an den expressJS-Server angehängt werden:



Abbildung : Socket.io-Server an expressJS-Server anhängen

Da das Backend und Frontend nicht auf dem gleichen Port laufen, muss via CORS Anfragen des Frontend erlaubt werden.

Ein Event kann dann in der REST-Ressource wie folgt emittiert werden:



Abbildung : Event über socket.io emittieren

Im Frontend wird der Event abgefangen und verarbeitet. Ist das Spiel fertig, wird der Benutzer zum Endergebnis weitergeleitet, sonst wird das Spielbrett aktualisiert.

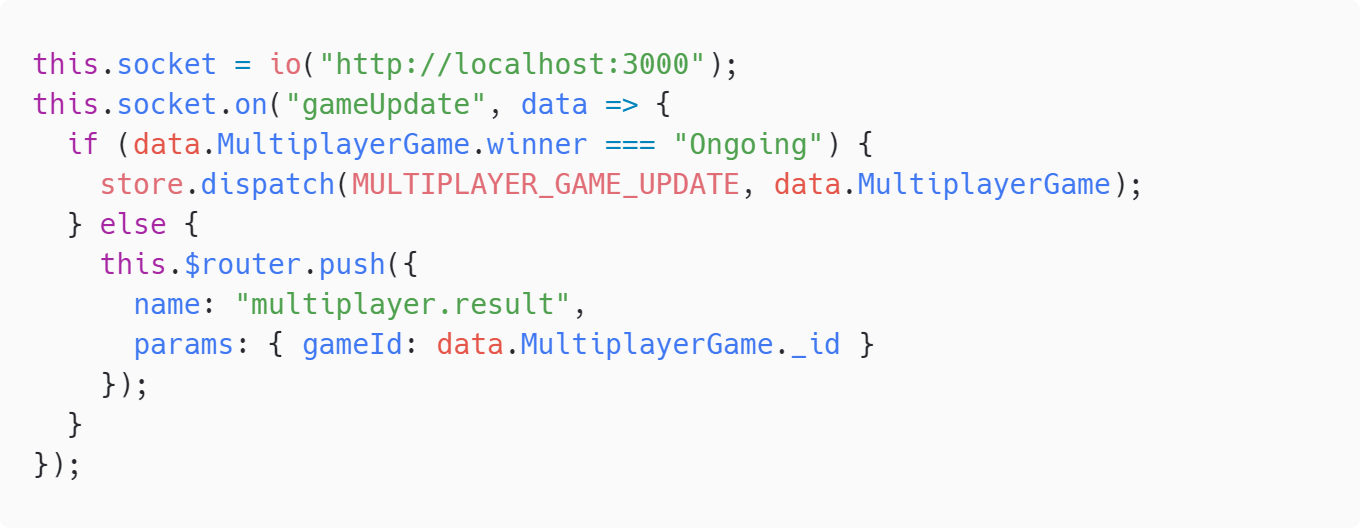


Abbildung : Verarbeiten eines socket.io Events

### Logik aus Komponenten auslagern

Um die axios-Logik, welche mit der REST-Schnittstelle spricht aus den Vue Komponenten auszulagern, wurde Vuex verwendet. So werden in den Komponenten jeweils nur Actions aufgerufen, welche dann den mit der API sprechen und den State von Vuex verändern. Der State von Vuex ist reaktiv und aktualisiert automatisch Eigenschaften in der Komponente.

Der Aufruf einer Action aus der Komponente sieht folgendermassen aus:

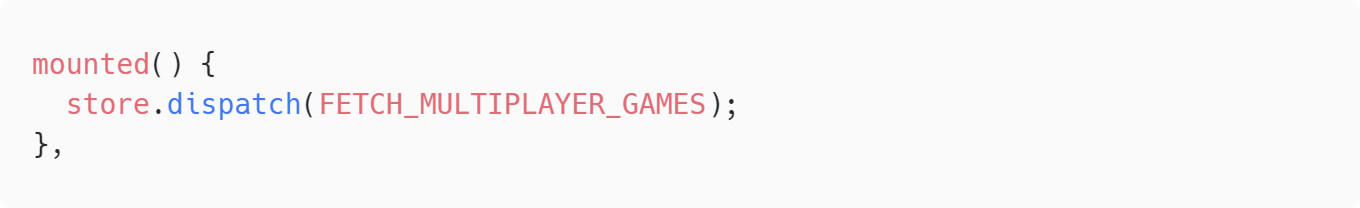


Abbildung : Aufruf von FETCH\_MULTIPLAYER\_GAMES

In der Action wird der MultiplayerGameService, welcher die Request an die REST-Schnittstelle macht, aufgerufen und dann der State über eine Mutation mit dem Ergebnis aktualisiert:



Abbildung 38: Action FETCH\_MULTIPLAYER\_GAMES

## GUI

Hier wird die finale Version der GUIs aus der Konzeptphase dargestellt.

Wenn es Unterschiede gibt, werden diese beschrieben.

### Allgemein

#### Startseite eingeloggt

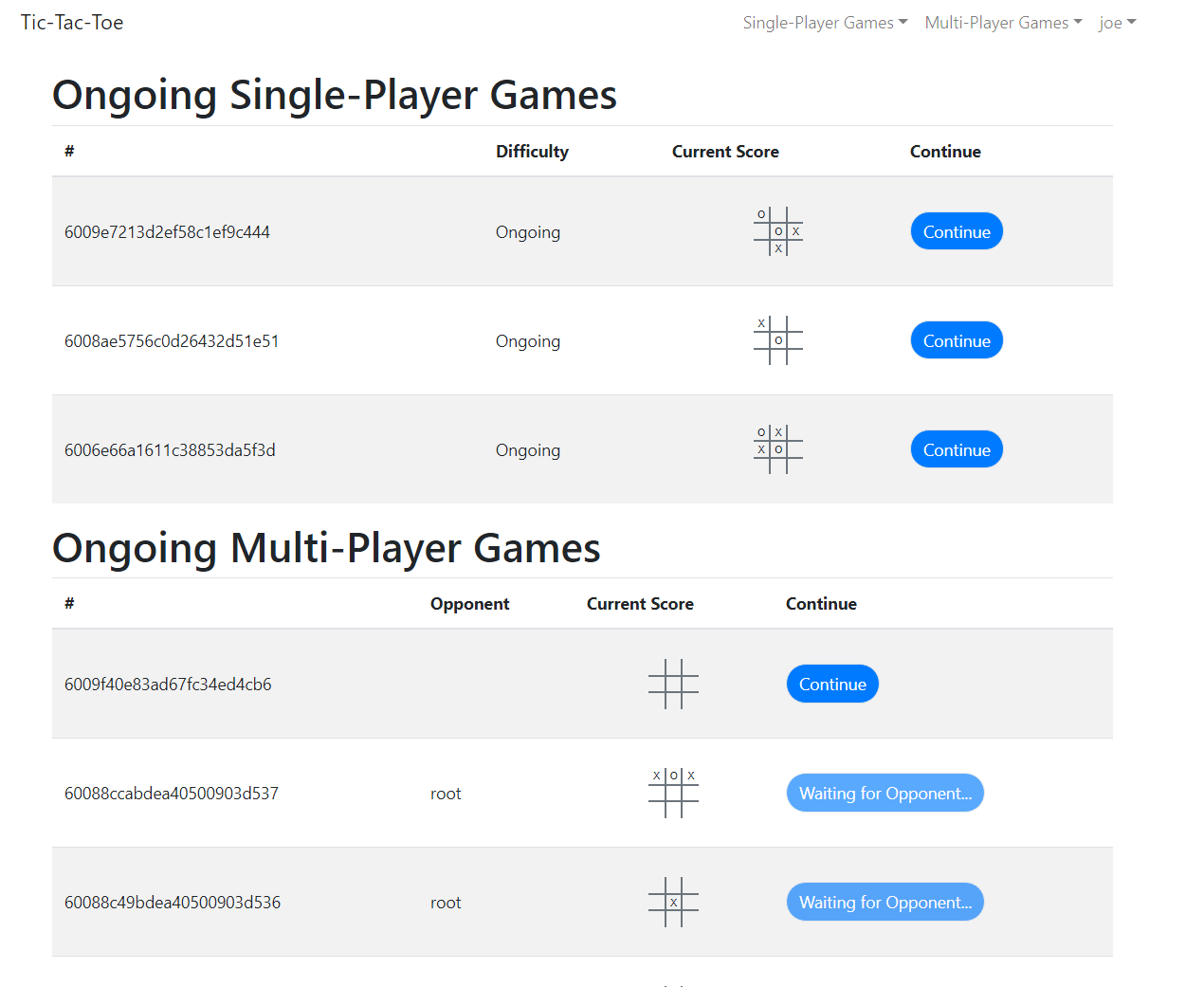


Abbildung : Finales-GUI Startseite eingeloggt

#### Sign up

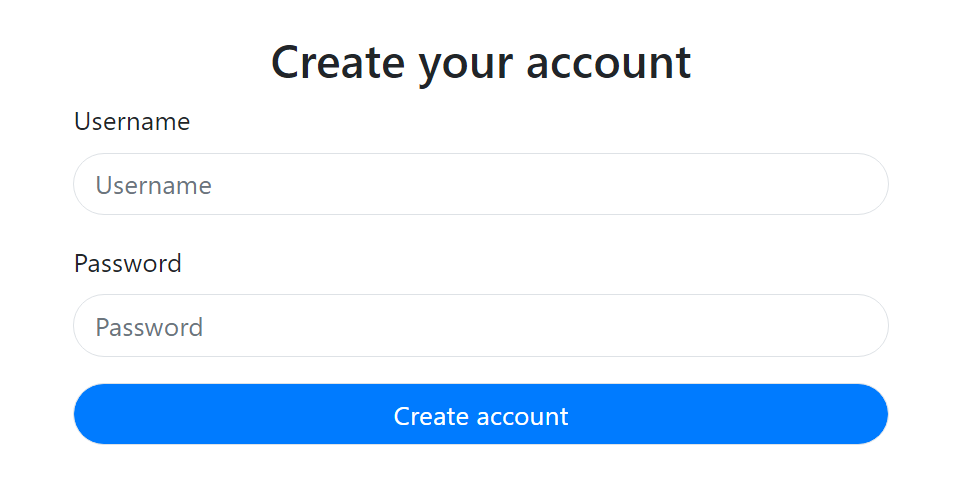


Abbildung : Finales-GUI Sign up

#### Sign in



Abbildung : Finales-GUI Sign in

### Einzelspieler

#### Einzelspielerspiel erstellen

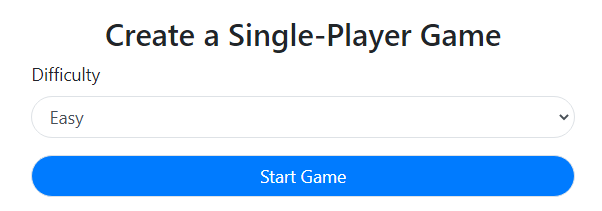


Abbildung : Finales-GUI Einzelspieler erstellen

#### Einzelspielerspiel spielen

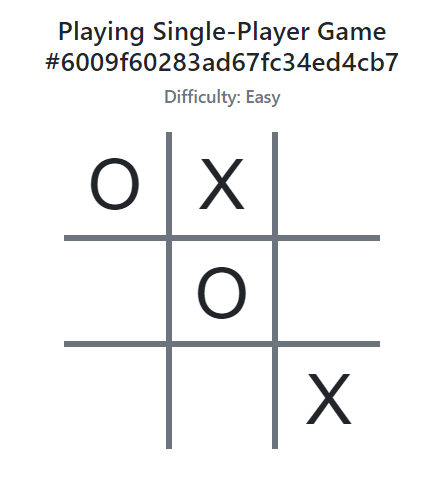


Abbildung : Finales-GUI Einzelspieler spielen

#### Aktuelle Einzelspielerspiele

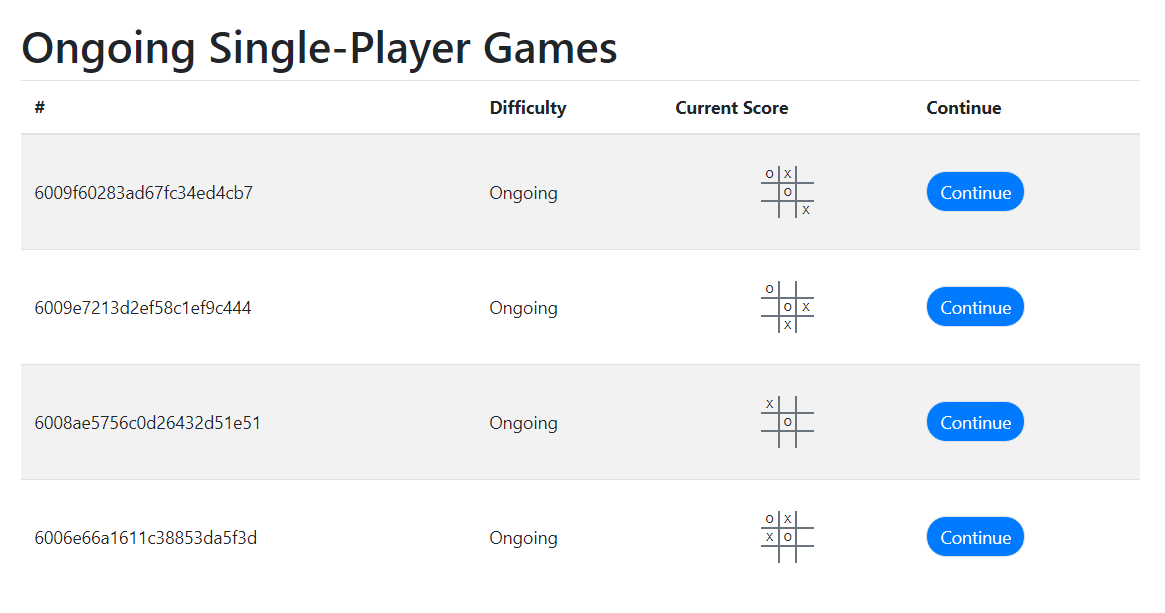


Abbildung : Finales-GUI Aktuelle Einzelspielerspiele

#### Einzelspielerspiel Ergebnis

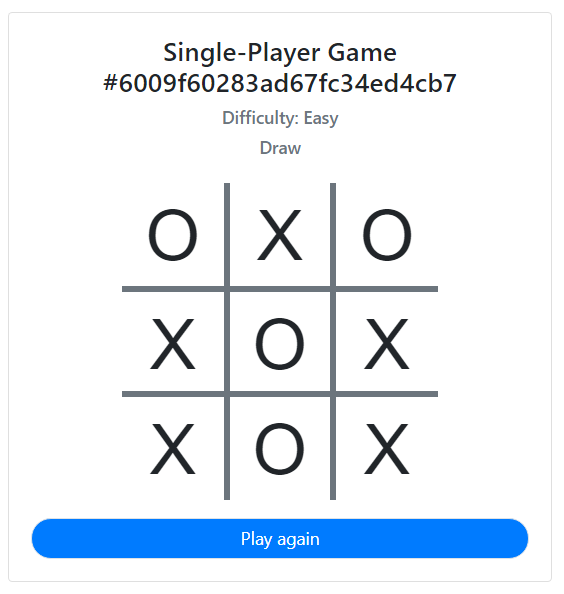


Abbildung : Finales-GUI Einzelspielerspiel Ergebnis

Hier wurde noch unter dem Schwierigkeitsgrad das Endergebnis dargestellt.

#### Vergangene Einzelspielerspiele

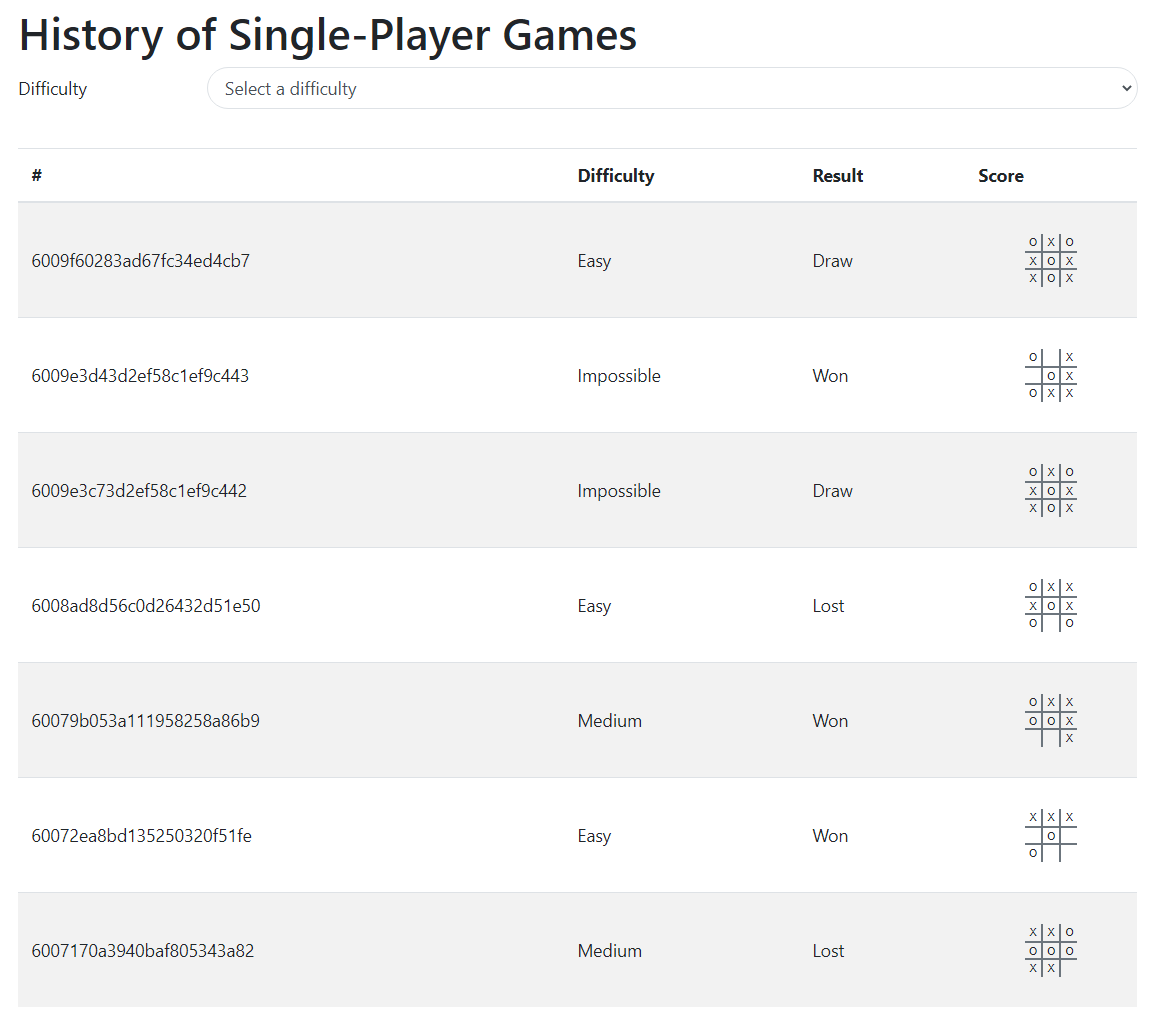


Abbildung : Finales-GUI Vergangene Einzelspielerspiele

Hier wurde der Suchbutton entfernt. Die Tabelle aktualisiert sich, sobald der Schwierigkeitsgrad verändert wird. Ein Suchbutton ist deshalb nur unnötig.

### Mehrspieler

#### Mehrspielerspiel erstellen

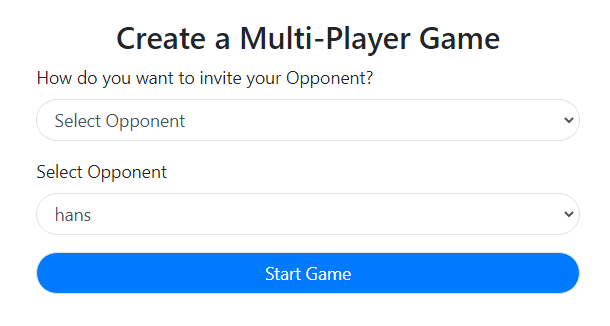


Abbildung : Finales-GUI Mehrspielerspiel erstellen

#### Mehrspielerspiel spielen

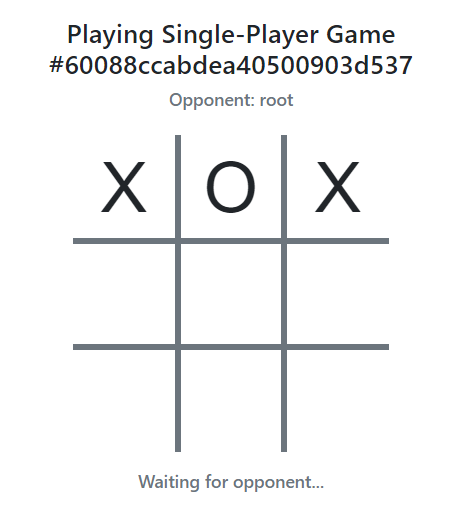


Abbildung : Finales-GUI Mehrspielerspiel erstellen

#### Aktuelle Mehrspielerspielspiele

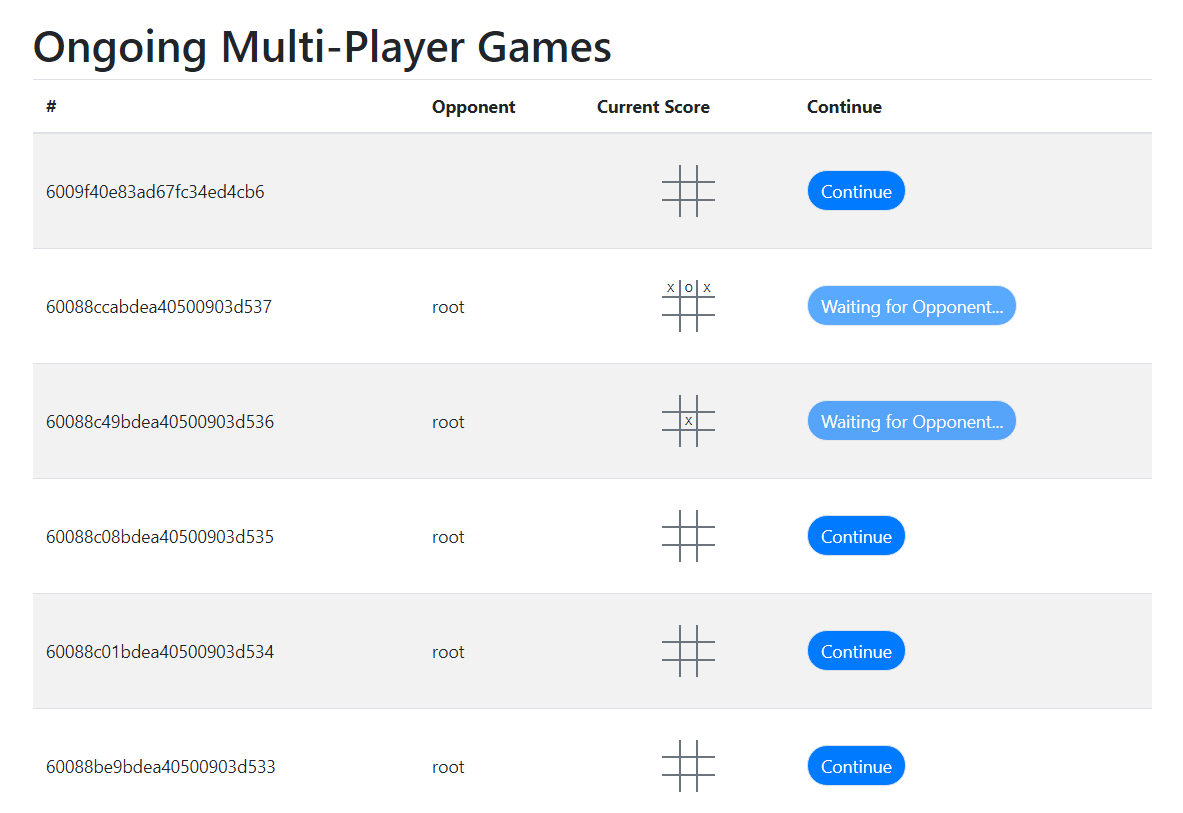


Abbildung : Finales-GUI Aktuelle Mehrspielerspiele

#### Mehrspielerspiel Ergebnis

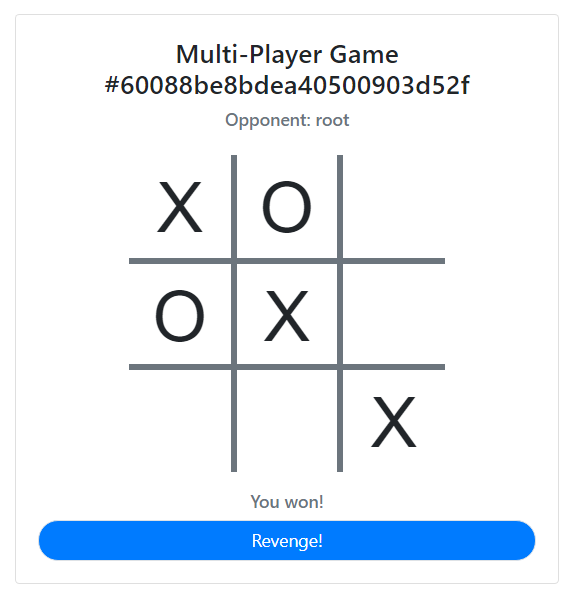


Abbildung : Finales-GUI Mehrspielerspiel Ergebnis

#### Vergangene Mehrspielerspiele

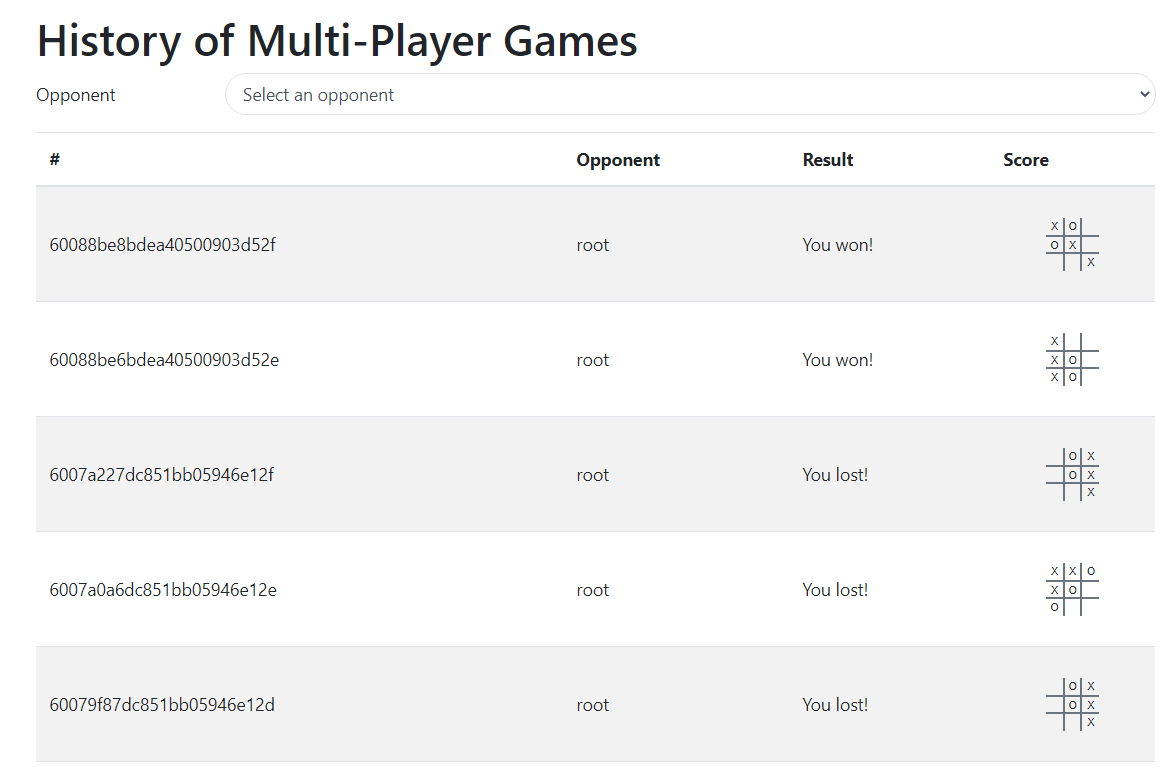


Abbildung : Finales-GUI Vergangene Mehrspielerspiele

Hier wurde auch der Suchbutton entfernt. Die Tabelle aktualisiert sich, sobald das Gegnersuchfeld verändert wird. Ein Suchbutton ist deshalb nicht nötig.

## Testprotokoll

In diesem Kapitel wurden die, im Konzept definierten, Testfälle durchgeführt. Es waren beim ersten Durchlauf alle Tests erfolgreich.

|  |  |
| --- | --- |
| Testfall | Ergebnis |
| T01 | Erfüllt |
| T02 | Erfüllt |
| T03 | Erfüllt |
| T04 | Erfüllt |
| T05 | Erfüllt |
| T06 | Erfüllt |
| T07 | Erfüllt |
| T08 | Erfüllt |
| T09 | Erfüllt |
| T10 | Erfüllt |
| T11 | Erfüllt |

Tabelle : Ergebnisse erster Testdurchlauf

### 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 klickt auf «Create account» |
| **Erwartetes Ergebnis** | Der Benutzer ist nun registriert. |
| **Tester** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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 «Sign in»-Seite weitergeleitet. Er hat keinen Zugriff mehr auf irgendwelche Spiele. |
| **Tester** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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. |
| **Tester** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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. |
| **Tester** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung 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** | Florin Barbisch |
| **Datum** | 21.01.2021 |
| **Ergebnis** | Erfüllt |
| **Fehlerbeschreibung** | - |

Tabelle : Erste Durchführung Testfall T11

# Reflexion

Das Projekt ist gut verlaufen. Ich habe mich an den Verlauf von Hermes gehalten und hatte so für die Realisierung klare Vorgaben. Auch zu diesem Erfolg hat die gute und klar formulierte Aufgabenstellung beigetragen.

Ich würde nächstes Mal aber mehr Tests machen, auch wenn dies mehr Zeit beansprucht. Ich konnte in den Tests nicht alle möglichen Parameter abdecken und es gibt deshalb nicht ein 100%-Abbild des Erfolgs. Nichtsdestotrotz habe ich mich an das Testkonzept gehalten und für jeden Use-Case einen Test. Damit ist der grösste Teil des Projekts abgedeckt.

Auch kritisch sehe ich die Vorgabe, das Arbeitsjournal in Wochen, statt in Tage aufzuteilen. Auch wenn es, im Sinne der Zeitspanne dieses Projekts (etwa ein Semester), keinen Sinn macht jeden Tag zu erfassen, macht es aufgrund der tatsächlichen Einteilung der Zeit (Letze 18 Tage vor Abgabe) Sinn. Jetzt sind gewisse Arbeitsjournaleinträge sehr mager während wiederum die letzten drei sehr gross sind.

Die 1,5 Stunden, welche unter der Woche in der Schule zur Verfügung stehen, können mässig zum Arbeiten genutzt werden. Vor allem, weil man sich jede Woche in das Thema wieder hereindenken muss und auch manchmal den PC wieder neu einrichten muss. Gegen letzteres habe ich aber Massnahmen ergriffen, und ein VPN und RDP eingerichtet. So kann ich mich vom Schul-PC aus auf meinen privaten Computer verbinden und auch die dort installierte Software brauchen.

Dazu kommt noch meine persönliche Präferenz, dass ich lieber mehrere Stunden an einem Projekt arbeitet und nicht für so kurze Zeit.

Ich habe schon diverse Projekte umgesetzt, bei denen ich mir die Zeit mehr auf den Ganzen Zeitraum verteilt habe. Das ich das in diesem Projekt nicht gemacht habe, liegt wahrscheinlich an zwei Tatsachen: Einerseits ist dies eine der letzten Arbeiten in der Schule und der Notenschnitt lässt sich nur noch unten korrigieren. Dazu brauche ich aber in diesem Semester eine Eins und im nächsten eine 1,5 (oder umgekehrt). Anderseits musste ich letzten Januar Abschied von meinem Vater nehmen, was immer noch Auswirkungen in Form einer gewissen Antriebslosigkeit hat.

# Schlusswort

Auch wenn es manchmal ein bisschen frustrierend war, wenn wieder etwas nicht funktioniert hat, war es im Grossen und Ganzen eine spannende und herausfordernde Projektarbeit. Ich während dem Entwickeln nicht nur bekanntes angewandt, sondern habe auch neue Technologien kennen gelernt. Ich habe in dieser Projektarbeit auch viel über Hermes und die IPA gelernt und denke, dass mir dies für die richtige IPA helfen wird.

Unklar ist noch, ob die Applikation jemals wirklich gebraucht wird. Ich denke damit die Applikation einen Anklang bei vielen Benutzer findet und auch zum Gamen gebraucht wird, müsste man noch ein Reizsystem einbauen. Denn wer kurz Tic-Tac-Toe spielen will, macht das entweder auf Papier oder über eine Suchmaschine wie Google[[18]](#footnote-18).

# Literatur- und Quellverzeichnis

[1] «Free Git GUI for Windows, Mac, Linux | GitKraken», *GitKraken.com*. https://www.gitkraken.com/ (zugegriffen Jan. 22, 2021).

[2] «Build software better, together», *GitHub*. https://github.com (zugegriffen Jan. 22, 2021).

[3] «WebStorm: The Smartest JavaScript IDE by JetBrains», *JetBrains*. https://www.jetbrains.com/webstorm/ (zugegriffen Jan. 22, 2021).

[4] «Visual Studio Code - Code Editing. Redefined». https://code.visualstudio.com/ (zugegriffen Jan. 22, 2021).

[5] «diagrams.net». https://app.diagrams.net/ (zugegriffen Jan. 22, 2021).

[6] «Bootstrap Studio - The Revolutionary Web Design Tool». http://bootstrapstudio.io/ (zugegriffen Jan. 22, 2021).

[7] «Carbon». https://carbon.now.sh/ (zugegriffen Jan. 22, 2021).

[8] «Zotero | Your personal research assistant». https://www.zotero.org/ (zugegriffen Jan. 22, 2021).

[9] «Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST)». https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\_arch\_style.htm (zugegriffen Jan. 22, 2021).

[10] «The most popular database for modern apps», *MongoDB*. https://www.mongodb.com/2 (zugegriffen Jan. 22, 2021).

[11] «Network socket», *Wikipedia*. Dez. 27, 2020, Zugegriffen: Jan. 22, 2021. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=Network\_socket&oldid=996623472.

[12] «The WebSocket API (WebSockets) - Web APIs | MDN». https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\_API (zugegriffen Jan. 22, 2021).

[13] D. Arrachequesne, «Socket.IO», *Socket.IO*, Jan. 22, 2021. https://socket.io/index.html (zugegriffen Jan. 22, 2021).

[14] «White-box testing», *Wikipedia*. Jan. 12, 2021, Zugegriffen: Jan. 22, 2021. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=White-box\_testing&oldid=999915567.

[15] «Softwaretest», *Wikipedia*. Jan. 18, 2021, Zugegriffen: Jan. 22, 2021. [Online]. Verfügbar unter: https://de.wikipedia.org/w/index.php?title=Softwaretest&oldid=207797419.

[16] «Black-box testing», *Wikipedia*. Jan. 12, 2021, Zugegriffen: Jan. 22, 2021. [Online]. Verfügbar unter: https://en.wikipedia.org/w/index.php?title=Black-box\_testing&oldid=999915211.

[17] «Express - Node.js web application framework». https://expressjs.com/ (zugegriffen Jan. 22, 2021).

[18] «tic tac toe - Google Search». https://www.google.com/search?q=tic+tac+toe (zugegriffen Jan. 22, 2021).

[19] AlDanial, *AlDanial/cloc*. 2021.

# Glossar

|  |  |
| --- | --- |
| Begriff | Bedeutung |
| API | Eine Computerschnittstelle, die Interaktionen  zwischen mehreren Software-Komponenten  regelt. |
| CORS | Cross-origin resource sharing. Damit wird kontrolliert, von welchen Servern Daten geladen werden können. Dies macht Sinn, wenn nicht alle Daten einer Webseite auf dem gleichen Server liegen. |
| expressJS | JavaScript Webserver welcher auf NodeJS läuft. Es ist sozusagen das Standard-Framework, wenn man einen Webserver in NodeJS implementieren will. |
| Git | Populäre Open-Source Versionierungssoftware. Git erlaubt es auch, dass mehrere Personen am gleichen Projekt gleichzeitig arbeiten. |
| JWT | JSON Web Token über das Sicherheitsansprüche (z.B. Zugriff auf das Admin-Panel oder Login auf einer Webseite) sicher übertragen werden können. |
| JSON | JavaScript Object Notation (kurz JSON) ist ein Format Daten zu übertragen. Das Format ist einfach aus der Software anzusteuern und kann gleichzeitig vom Benutzer gut gelesen werden. |
| MongoDB | Datenbank welche Dokument-orientiert ist. Diese Datenbanken sind nur semi-organisiert. |
| NodeJS | Plattformunabhängiges JavaScript Runtime-Environment welches nicht im Browser läuft. Damit können Applikationen oder Server in JavaScript geschrieben werden. |
| REST | Representational State Transfer (kurz REST) ist ein Paradigma der Softwarearchitektur von Webservices. Es zielt auf die |
| Router | expressJS-Komponenten, welche zum organisieren HTTP Anfragen dient. Dies hilft eine Ordnerstruktur zu pflegen, welche ähnlich wie die URL ist. |
| Schema | Struktur einer Datenbanktabelle oder eines Datenbankdokument. |
| Socket | Kommunikationsendpunkt welcher zum Kommunizieren von zwei Computer verwendet wird. Die Kommunikation erfolgt meistens bidirektional. |
| Tic-Tac-Toe | Tic-Tac-Toe ist ein klassisches, einfaches Zweipersonen-Strategiespiel. Das Spiel besteht aus einem 3x3 Spielbrett. Ein Spieler ist ‘O’ und der andere ‘X’. Sie setzen abwechslungsweise ihr Zeichen auf ein leeres Feld. Wer als erster 3 seiner Zeichen in einer Zeile, Spalte oder Diagonalen hat, hat gewonnen. Ist das Spielbrett voll ist Unentschieden. |
| VueJS | VueJS ist ein Model View Viewmodel JavaScript-Framework mit dem das Benutzerinterface einer Webseite erstellt werden kann und reaktiv mit Daten verbunden werden kann. |

Tabelle : Glossar

# Anhang

Im Anhang sind die Code Listings zu finden. Da diese nicht abgegeben werden sollen, ist hier eine Übersicht über die Anzahl der Codezeilen im Projekt. Diese wurde mit cloc[[19]](#footnote-19) erstellt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprache | Dateien | Leere Zeilen | Kommentarzeilen | Codezeilen |
| JavaScript | 28 | 210 | 106 | 1561 |
| VueJS | 15 | 41 | 4 | 1332 |
| XML | 8 | 0 | 0 | 229 |
| JSON | 2 | 0 | 0 | 102 |
| CSS | 1 | 13 | 0 | 77 |
| HTML | 1 | 0 | 1 | 16 |
| Total | 55 | 264 | 111 | 3317 |

Tabelle : Codezeilen-Übersicht

1. [1] «Free Git GUI for Windows, Mac, Linux | GitKraken». [↑](#footnote-ref-1)
2. [2] «Build software better, together». [↑](#footnote-ref-2)
3. [3] «WebStorm: The Smartest JavaScript IDE by JetBrains». [↑](#footnote-ref-3)
4. [4] «Visual Studio Code - Code Editing. Redefined». [↑](#footnote-ref-4)
5. [5] «diagrams.net». [↑](#footnote-ref-5)
6. [6] «Bootstrap Studio - The Revolutionary Web Design Tool». [↑](#footnote-ref-6)
7. [7] «Carbon». [↑](#footnote-ref-7)
8. [8] «Zotero | Your personal research assistant». [↑](#footnote-ref-8)
9. [9] «Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST)». [↑](#footnote-ref-9)
10. [10] «The most popular database for modern apps». [↑](#footnote-ref-10)
11. [11] «Network socket», *Wikipedia* [↑](#footnote-ref-11)
12. [12] «The WebSocket API (WebSockets) - Web APIs | MDN». [↑](#footnote-ref-12)
13. [13] «Socket.IO». [↑](#footnote-ref-13)
14. [14] «White-box testing», *Wikipedia* [↑](#footnote-ref-14)
15. [15] «Softwaretest», *Wikipedia* [↑](#footnote-ref-15)
16. [16] «Black-box testing», *Wikipedia* [↑](#footnote-ref-16)
17. [17] «Express - Node.js web application framework». [↑](#footnote-ref-17)
18. [18] «tic tac toe - Google Search». [↑](#footnote-ref-18)
19. [19] *AlDanial/cloc*. [↑](#footnote-ref-19)