|  |  |  |  |
| --- | --- | --- | --- |
| **Übersicht** | | | |
| **Projekt** | **Online Snake** | | |
| **Projektleiter** | **Marcel Steinle** | **Datum** | **09.03.2024** |

Inhalt

[1. Zieldefinition (als SMART Ziele) 1](#_Toc1091876301)

[2. Anwendungsfälle im Überblick 2](#_Toc507624862)

[3. Anwendungsfälle im Detail 3](#_Toc2084209827)

[3.1. Anwendungsfälle Game-Page 4](#_Toc1055022245)

[3.2. Anwendungsfälle Home-Page 6](#_Toc1117242957)

[4. Grundlagen der Qualitätsanforderungen 10](#_Toc1390702257)

[5. Anforderungen im Überblick 12](#_Toc788645764)

[6. Qualitätsanforderungen im Detail 14](#_Toc269905233)

[6.1. Funktionalität 15](#_Toc915464739)

[6.2. Zuverlässigkeit 15](#_Toc1581050912)

[6.3. Effizienz 15](#_Toc924960562)

[6.4. Benutzbarkeit 15](#_Toc182650742)

[6.5. Übertragbarkeit 15](#_Toc756381018)

[6.6. Wartbarkeit 16](#_Toc1666213521)

[7. Übersicht der relevanten Daten 16](#_Toc1118223147)

[8. Glossar 16](#_Toc1569274727)

[9. Ausfüllhilfe 16](#_Toc609456506)

[9.1. Anwendungsfälle 17](#_Toc1700756269)

[9.2. Anforderungen 17](#_Toc1390889926)

# Zieldefinition (als SMART Ziele)

**(Alle)**

***S****pecific (spezifisch): Ziele müssen eindeutig und verständlich sein.*  
Als Gruppe wollen wir ein Multiplayer Snake erstellen, das auf einer Internetseite Spielbar ist. Jede Schlange kann Früchte essen, um größer zu werden. Wenn eine Schlange in sich selbst oder in eine andere Schlange schleicht, dann stirbt sie. Verschiedene Spielmodi bestimmen die Gewinnbedingung. Spielgruppen (bzw. Lobbys) können von Spielern erstellt werden, sie können privat oder öffentlich sein.

***M****easurable (messbar): Ziele müssen messbar sein*  
Das Spiel soll alleine spielbar sein, der Spieler gewinnt, wenn er die maximale Größe erreicht hat (Wenn er so groß wie das Spielfeld ist). Der Einzelspieler und der Multiplayer soll über der Startseite erreichbar sein. Für Multiplayer soll man einem Spiel zufällig beitreten können oder man tritt einer (von einem Spieler erstellten) Lobby über einen Code bei oder man erstellt eine Lobby selbst. Bei der Lobby Erstellung kann man die Raumgröße einstellen. Im Multiplayer gibt es mehrere „Äpfel“, die jeder Spieler essen kann. Schleicht eine Schlange in eine andere, so stirbt diese. Der im Spiel gestorbene Spieler schaut dann zu, bis er wieder auf Spielen klickt, dann wird er wieder zufällig auf der Karte platziert. Im Multiplayer kann man nicht gewinnen, aber es wird eine Tabelle angezeigt, die den Rekord anzeigt, wer die größte Schlange war und mit welcher Länge.

***A****ppropriate /* ***A****tractive (angemessen/wünschenswert): Ziele müssen den Aufwand rechtfertigen.*  
Unsere Ambitionen sind eine gute Note zu erreichen und dabei etwas spielbares/lustiges für den Zeitvertreib mit Freunden oder alleine zu kreieren.

***R****ealistic (realistisch): Ziele müssen mit den gegebenen Mitteln erreicht werden können.*  
Ein Einzelspieler Snake mit JavaScript zu programmieren leicht, dafür reichen ca. 80 Zeilen code. Es gibt viele ähnliche Snake Spiele im Web, die sogar komplizierter sind (siehe slither.io). In unserem Spiel soll man sich nur horizontal und vertikal bewegen können, was eben vereinfacht ist.

***T****ime-based (zeitbasiert): Ziele müssen in absehbarer Zeit erreicht werden können.*  
Die einzelne Ziele sind in der angegebenen Zeit erreichbar. Zu viert können wir uns die Teilziele aufteilen.

# Anwendungsfälle im Überblick

H = HomePage, G = GamePage; 0 = höchste Priorität, 5 = niedrigste Priorität

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Titel | Beschreibung | Prio | Kritikalität |
| AF-G-01 Alle | Spielen des Einzelspielerspiels | Der Spieler steuert die Richtung, in die seine Schlange “schleicht”. Der Spieler Essen verschlingen, indem er über das Essen schleicht, woraufhin die Schlange länger wird. Wenn die Schlange in sich selbst oder z.B. gegen eine Wand schleicht, dann stirbt sie (Ihr Körper bleibt stehen). Dann auf den Befehl des Spielers wieder das Spiel neu geladen werden, die Schlange wird irgendwo auf der Karte platziert, mit der Startlänge. | 1 | Kritisch |
| AF-G-02 Alle | Spielen des Mehrspielerspiels | Genau wie beim Einzelspieler AF-G-01 aber mit anderen Spielern. Wobei beim Tod einer Schlange ihr Körper verschwindet. | 2 | Kritisch |
| AF-G-03 Alle | Verlassen eines Spiels | Das Spiel muss die Möglichkeit haben, ein Spiel (Solo oder Multi) zu verlassen, um auf das Hauptmenü zurückzukehren. (Ohne einfach die Seite zu schließen) | 1 | Wichtig |
| AF-H-01 Alle | Starten eines Solospiels | Über einen Startknopf auf der Homepage soll der Spieler ein Solospiel mit verschiedenen Konfigurationen starten können. | 1 | Kritisch |
| AF-H-02 Alle | Erstellen eines Multiplayer Raums | Auf der Homepage soll ein „Raum erstellen“ Knopf existieren, über die der Spieler eine UI angezeigt bekommt, auf der er konfigurieren kann (z.B. ein privater Raum mit Passwort, Anzahl der Äpfel) und dann das Spiel starten kann. | 2 | Kritisch |
| AF-H-03 Alle | Beitreten eines Multiplayer Raums über einen Raum Code | Auf der Homepage soll eine Box existieren, in der man den Code eines Raumes eingeben kann, um somit gezielt einem bestimmten Raum beizutreten. | 2 | Kritisch |
| AF-H-04 Alle | Automatisches Suchen und Beitreten eines Raums. | Der Spieler kann, ohne einen Code einzugeben, automatisch, nach eingeben seines Namens, ein Spiel beitreten -> Global Queue | 4 | Unwichtig |

# Anwendungsfälle im Detail

## 3.1. Anwendungsfälle Game-Page

**(Alexander)**

|  |  |
| --- | --- |
| ID | AF-G-01 |
| Titel | Spielen des Einzelspielerspiels |
| Beschreibung | Ablauf des Spiels und Spielstarts |
| Akteure | Der Spieler, der Client, der lokale Server |
| Vorbedingungen  *(Was muss erfüllt sein, bevor der Use Case beginnt?)* | Auf der Homepage muss die Einzelspielertaste gedrückt werden, bei Bedarf kann dann auch das Spiel konfiguriert werden. Die Logik und Berechnungen für das Spiel übernimmt dann der Client selbst. |
| Nachbedingungen  *(Was muss erfüllt sein, wenn der UseCase erfolgreich beendet ist?)* | Wenn das Spiel beendet wird, soll der Spieler zurück zur Homepage geführt werden. |
| Ablauf/Main Path *(Der erwartete oder gewöhnliche Weg des Use Case)* | Während des Spiels soll der Spieler die Richtung steuern können, in die seine Schlange “schleicht”. Der Spieler kann über Essen schleichen um es zu verschlingen, dadurch wird die Schlange länger. Wenn die Schlange in sich selbst oder z.B. gegen eine Wand schleicht, dann stirbt sie (der Bildschirm bleibt stehen, das Spiel/die Runde ist vorbei). Wenn der Spieler weiterspielen möchte, muss die Schlange wieder irgendwo auf der Karte platziert werden, mit der Startlänge. Das Essen wird neu zufällig auf der Karte verteilt, alle Werte werden zurückgesetzt und das Spiel fängt wieder an zu laufen. |
| Erweiterung/ Alternative Path  *(Ablauf für definierte Ausnahmen)* |  |
| Priorität / Kritikalität | 0 |

**(Alexander)**

|  |  |
| --- | --- |
| ID | AF-G-02 |
| Titel | Spielen in einem Multispielerspiel |
| Beschreibung | Ablauf von beitreten/erstellen und Spielen eines Mehrspielerspiels. |
| Akteure | Der Spieler, der Client, der Server |
| Vorbedingungen  *(Was muss erfüllt sein, bevor der Use Case beginnt?)* | Man kann einem bereits existierenden Raum durch die Eingabe des Raumcodes auf der Homepage beitreten. Alternativ kann auch ein eigenes Mehrspielerspiel erstellt werden (Der Client, der das macht, ist dann der Raumbesitzer), bei Bedarf kann dieses Spiel dann auch konfiguriert werden, sodass man z.B. einem Raum nur mit einem Passwort beitreten kann, oder eine spezifische Kartengröße. Nach der Raumerstellung wird eine Anfrage auf dem Main Server gemacht, sodass der Main Server einen Raum erstellt und die Rolle für die Logik Berechnung übernimmt. |
| Nachbedingungen  *(Was muss erfüllt sein, wenn der UseCase erfolgreich beendet ist?)* | Wenn der Raumbesitzer das Spiel verlässt oder die Verbindung abbricht, muss einem anderen Spieler die Besitzerrolle zugewiesen werden. Wenn keine Spieler mehr im Raum sind, muss der Server den Raum schließen. |
| Ablauf/Main Path *(Der erwartete oder gewöhnliche Weg des Use Case)* | Der Spielablauf ist derselbe wie beim Einzelspieler, jedoch befinden sich in diesem Fall auch andere Schlangen (andere Spieler) auf der Karte. Eine Schlange stirbt auch, wenn sie in eine andere Schlange schleicht. Technisch gesehen übernimmt jetzt der Main Server die Logik, d.h. dass jeder Client nur die aktuellen Daten abfragt und diese dann verwendet, um die Spielkarte zu rendern. |
| Erweiterung/ Alternative Path  *(Ablauf für definierte Ausnahmen)* |  |
| Priorität / Kritikalität | 1 |

**(Alexander)**

|  |  |
| --- | --- |
| ID | AF-G-03 |
| Titel | Verlassen eines Spiels |
| Beschreibung | Der Spieler muss eine Möglichkeit haben um in das Hauptmenü zurückzukommen, ohne die Website neuladen zu müssen. |
| Akteure | Der Spieler, der Client |
| Vorbedingungen  *(Was muss erfüllt sein, bevor der Use Case beginnt?)* | - Sodass der Spieler ein Spiel verlassen kann, muss er sich in einem Solo- oder Mehrspielerspiel befinden. |
| Nachbedingungen  *(Was muss erfüllt sein, wenn der UseCase erfolgreich beendet ist?)* | - Nach dem Verlassen eines Spiels muss der Spieler sich wieder im Hauptmenü befinden. |
| Ablauf/Main Path *(Der erwartete oder gewöhnliche Weg des Use Case)* | - Über das Drücken einer Optionen Taste auf dem Bildschirm oder durch das Drücken der ESC Taste soll ein Spielemenü geöffnet werden können, auf dem man unter anderem das Spiel verlassen kann. |
| Erweiterung/ Alternative Path  *(Ablauf für definierte Ausnahmen)* | Es kann auch sein, dass der Spieler einfach die Webseite schließt, wenn der Spieler sich in einem Mehrspielerspiel befand, muss der Server das erkennen und die Spielerinstanz aus dem Raum entfernen. |
| Priorität / Kritikalität | 0 / Kritisch |

## 3.2. Anwendungsfälle Home-Page

**(Jan)**

|  |  |
| --- | --- |
| ID | AF-H-01 |
| Titel | Starten eines Solospiels |
| Beschreibung | Über einen Startknopf auf der Homepage soll der Spieler ein Solospiel mit verschiedenen Konfigurationen starten können. |
| Akteure | Der Nutzer und der Client |
| Vorbedingungen  *(Was muss erfüllt sein, bevor der Use Case beginnt?)* | Der Nutzer ist über eine Anfrage an den Client, der Homepage beigetreten. Auf der Homepage kann man einen optionalen Namen hinzufügen und dann über den Button eine Anfrage an den Client schicken. |
| Nachbedingungen  *(Was muss erfüllt sein, wenn der UseCase erfolgreich beendet ist?)* | Die Homepage ist geschlossen und der Nutzer befindet sich auf der Gamepage mit den voreingestellten Einstellungen. |
| Ablauf/Main Path *(Der erwartete oder gewöhnliche Weg des Use Case)* | Der Nutzer sendet eine Anfrage an den Client, durch das Drücken eines Knopfes. Über diese Anfrage wird der Nutzer mit seinen Voreinstellungen auf die Gamepage weitergeleitet. |
| Erweiterung/ Alternative Path  *(Ablauf für definierte Ausnahmen)* |  |
| Priorität / Kritikalität | 1 / Kritisch |

**(Jan)**

|  |  |
| --- | --- |
| ID | AF-H-02 |
| Titel | Erstellen eines Multiplayerspiels |
| Beschreibung | Auf der Homepage soll ein „Raum erstellen“ Knopf existieren, über die der Spieler eine UI angezeigt bekommt, auf der er konfigurieren kann (z.B. ein privater Raum mit Passwort, Anzahl der Äpfel) und dann das Spiel starten kann. |
| Akteure | Der Nutzer und der Client |
| Vorbedingungen  *(Was muss erfüllt sein, bevor der Use Case beginnt?)* | Der Nutzer ist über eine Anfrage an den Client, der Homepage beigetreten. Auf der Homepage fügt man einen Namen hinzu und schickt über den Button eine Anfrage an den Client. |
| Nachbedingungen  *(Was muss erfüllt sein, wenn der UseCase erfolgreich beendet ist?)* | Die Homepage ist geschlossen und der Nutzer befindet sich auf der Gamepage mit den voreingestellten Einstellungen. |
| Ablauf/Main Path *(Der erwartete oder gewöhnliche Weg des Use Case)* | Der Nutzer sendet durch Betätigung des Knopfes eine Anfrage an den Client. Dieser fragt nach Auswahl von gewissen Pflichtfeldern. Zum Beispiel nach dem Namen oder des Passwortes und so weiter. Danach leitet der Client den Nutzer auf den zugehörigen Server weiter. |
| Erweiterung/ Alternative Path  *(Ablauf für definierte Ausnahmen)* |  |
| Priorität / Kritikalität | 2 / Kritisch |

**(Hermann)**

|  |  |
| --- | --- |
| ID | AF-H-03 |
| Titel | Beitreten eines Multiplayer Raums über einen Raum Code |
| Beschreibung | Auf der Homepage soll eine Box existieren, in der man den Code eines Raumes eingeben kann, um somit gezielt einem bestimmten Raum beizutreten. |
| Akteure | Der Nutzer, der Client, der Code-Host |
| Vorbedingungen  *(Was muss erfüllt sein, bevor der Use Case beginnt?)* | Der Nutzer muss über einen Client, auf die HomePage zugreifen können. Der Nutzer kann einen optionalen Namen eingeben und über Betätigen der “Multiplayer” Taste soll ein Fenster erscheinen, in dem ein Raum-Code eingegeben werden kann. |
| Nachbedingungen  *(Was muss erfüllt sein, wenn der UseCase erfolgreich beendet ist?)* | Der Host kann den Client starten und beide Spieler gelangen auf die GamePage und können das Spiel beginnen. |
| Ablauf/Main Path *(Der erwartete oder gewöhnliche Weg des Use Case)* | Der Nutzer muss einen bereits-vorhandenen Code, der von einem Host erstellt wurde, eingeben, um in den Raum vom Host zu gelangen. Über eine “Beitreten” Taste gelangt der Nutzer in den Raum vom Host. |
| Erweiterung/ Alternative Path  *(Ablauf für definierte Ausnahmen)* |  |
| Priorität / Kritikalität | 2 / Kritisch |

**(Hermann)**

|  |  |
| --- | --- |
| ID | AF-H-04 |
| Titel | Automatisches Suchen und Beitreten eines Raums |
| Beschreibung | Der Spieler kann, ohne einen Code einzugeben, automatisch, nach eingeben seines Namens, ein Spiel beitreten -> Global Queue |
| Akteure | Der Nutzer, der Client |
| Vorbedingungen  *(Was muss erfüllt sein, bevor der Use Case beginnt?)* | Der Nutzer muss über einen Client, auf die HomePage zugreifen können. Der Nutzer kann einen optionalen Namen eingeben und über Betätigen der “Multiplayer” Taste soll ein Fenster erscheinen, in dem der Nutzer die Optionen hat, das Spiel über einen Raum-Code zu betreten oder er gibt nichts ein und wird in ein bereitsvorhandenes Spiel geworfen. |
| Nachbedingungen  *(Was muss erfüllt sein, wenn der UseCase erfolgreich beendet ist?)* | Die Homepage wird verlassen und der Nutzer gelangt in die GamePage, in der anderen Nutzer bereits das Spiel spielen oder noch beitreten können. |
| Ablauf/Main Path *(Der erwartete oder gewöhnliche Weg des Use Case)* | Wenn der Nutzer auf die “Beitreten” Taste klickt, dann wird er automatisch, über den Client, in ein Raum geworfen, in dem das Spiel bereits gestartet hat. |
| Erweiterung/ Alternative Path  *(Ablauf für definierte Ausnahmen)* |  |
| Priorität / Kritikalität | 4 / Unwichtig |

# Grundlagen der Qualitätsanforderungen **(Alexander)**

**Funktionalität (functionality)**

( ) ( ) ( ) ( ) (X)

Vollkommen egal sehr wichtig

Tauglichkeit (suitability)

( ) ( ) ( ) (X) ( )

Genauigkeit (accuracy)

( ) ( ) ( ) ( ) (X)

Interoperabilität (interoperability)

( ) ( ) (X) ( ) ( )

Regelkonformität (compliance)

( ) ( ) (X) ( ) ( )

Sicherheit (security)

( ) ( ) ( ) (X) ( )

**Zuverlässigkeit (reliability)**

( ) ( ) (X) ( ) ( )

Vollkommen egal sehr wichtig

Reifegrad (Maturity)

( ) ( ) ( ) (X) ( )

Fehlertoleranz (Fault tolerance)

( ) ( ) (X) ( ) ( )

Wiederherstellbarkeit (Recoverability)

( ) (X) ( ) ( ) ( )

**Effizienz (efficiency)**

( ) ( ) ( ) ( ) (X)

Vollkommen egal sehr wichtig

(Reaktions-)Geschwindigkeit (Time behavior)

( ) ( ) ( ) ( ) (X)

Ressourcenbedarf (Resource behavior)

( ) ( ) ( ) (X) ( )

**Benutzbarkeit (usability)**

( ) ( ) ( ) (X) ( )

Vollkommen egal sehr wichtig

Verstehbarkeit (Understandability)

( ) ( ) ( ) (X) ( )

Lernfähigkeit (Learnability)

( ) (X) ( ) ( ) ()

Bedienbarkeit (Operability)

( ) ( ) ( ) ( ) (X)

**Übertragbarkeit (portability)**

( ) ( ) ( ) (X) ( )

Vollkommen egal sehr wichtig

Anpassungsfähigkeit (adaptability)

( ) (X) ( ) ( ) ( )

Installierbarkeit (installability)

( ) (X) ( ) ( ) ( )

Konformität (conformance)

( ) (X) ( ) ( ) ( )

Austauschbarkeit (replaceability)

( ) ( ) (X) ( ) ( )

**Wartbarkeit (maintainability)**

( ) ( ) ( ) ( ) (X)

Vollkommen egal sehr wichtig

Analysierbarkeit (analyzability)

( ) ( ) ( ) ( ) (X)

Veränderlichkeit (changeability)

( ) ( ) ( ) ( ) (X)

Stabilität (stability)

( ) ( ) ( ) ( ) (X)

Testbarkeit (testability)

( ) ( ) (X) ( ) ( )

# Anforderungen im Überblick

FA = Funktionale Anforderung, NA = Nicht-funktionale Anforderung  
H = HomePage, G = GamePage, B = Backend

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Titel | Beschreibung | Prio | Kritikalität |
| FA-G-01  **Hermann** | Steuern der Schlange | Die Schlange sollte mit den Pfeiltasten und mit WASD gesteuert werden können. | 1 | Kritisch |
| FA-G-02 | Erreichbarkeit des Servers | Ein für alle Clients erreichbarer Server muss existieren, der die Logik für die Mehrspielerräume übernimmt. | 2 | Kritisch |
| FA-G-03 | Verlassen eines Spiels | Ein Spieler muss ein Solo- oder Multi Spiel verlassen können, ohne die Seite schließen zu müssen. | 1 | Kritisch |
| FA-G-04 | Neuzuweiseung des Raumbesitzers | Wenn der Besitzer eines Raums den Raum verlässt, muss einen anderen Spieler die Rolle des Raumbesitzers zugeweisen werden. | 2 | Unwichtig |
| FA-H-02 | Funktionalität der Knöpfe | Vorallem auf der Homepage befinden sich Buttons, sodass ein Spieler ein Spiel starten kann etc. | 0 | Kritisch |
| FA-GH-01 | Github Page | Ein Client muss auf einer Website das Online Snake laden und spielen können. | 1 | Unwichtig |
| NA-H-01 | Gutaussehende und leicht verständliche UI | Die UI soll entsprechend unserer Fantasie einer Thematik entsprechen und intuitiv bedienbar sein. | 3 | Unwichtig |
| NA-H-02 | Lokale Bestenliste | Auf der Homepage sollen die lokalen Bestenlisten sichtbar sein (Rekordzahlen vom Singleplayer). | 4 | Unwichtig |
| NA-GH-01  **Alexander** | Sternen Parallaxe (Hintergrund) | Der Homepage und Spiel Hintergrund soll eine Sternenparallaxe beinhalten. Diese sorgt für eine Illusion des Unendlichen. | 3 | Unwichtig |
| NA-GH-02 | Light und Dark Mode | Eine helle und eine dunkle Thematik sollen existieren. | 3 | Unwichtig |
| NA-G-01  Jan | Schlangenname | Der Spieler-Name soll über der Spieler-Schlange angezeigt werden | 3 | Unwichtig |
| NA-G-02 | Abgerundete Schlangensegmente | Die Ecken der Schlange sollen abgerundet werden. | 3 | Unwichtig |
| NA-G-03  Jan | Menü im Spiel | Ein Optionen Menü, in dem man die Lautstärke und die Farbe der Schlange einstellen sowie das Spiel verlassen kann. | 2 | Unwichtig |
| NA-G-04 | Skalierbares Spielfeld | Die Kamera soll zoomen können. | 3 | Unwichtig |
| NA-G-05  **Alexander** | Spielkamera (frei) | Im Spiel soll die Kamera zwei Modis haben. Im freien Modus soll die Kamera das ganze Spielfeld zeigen. | 1 | Kritisch |
| NA-G-06  **Alexander** | Spielkamera (fixiert) | Im Spiel soll die Kamera zwei Modis haben. Im fixierten Modus ist die Kamera auf den Kopf der Schlange fixiert, sodass sich der Schlangenkopf immer in der Mitte des Bildschirms befindet. | 3 | Unwichtig |
| NA-G-07 | Bestenliste innerhalb eines Raumes (Mehrspieler) | In einem Mehrspieler Raum soll aufgezeichnet werden, wer die größte Schlange war und mit welcher Länge. Dazu soll die aktuelle Bestenliste angezeigt werden. | 3 | Unwichtig |
| NA-G-08  **Alexander** | Einzelspieler konfigurationen | Bevor ein Spieler ein Einzelspiel startet, soll der Spieler die Möglichkeit haben, das Spiel zu konfigurieren. Alle Konfigurationen werden hier aufgelistet: -Karten breite -Karten höhe  -Anzahl an Essen -Ob die Wand tödlich ist (Kann die Schlange durch wie Wand laufen? Dann wird die Schlange wieder auf der anderen Seite wieder rauskommen) | 2 |  |
| NA-G-09  Jan | Spielmodi | Verschiedene auswählbare Spielmodi | 3 |  |
| NA-GB-01  Jan | Ping | Möglichst niedriger Ping durch Optimierung | 1 | Wichtig |

# Qualitätsanforderungen im Detail

Zusätzliche Informationen zu unterschiedlichen Qualitätsanforderungen, unter anderem durch Definition und Bewertung relevanter Qualitätsmerkmale.

**(Marcel)**

## 6.1. Funktionalität

Unser Projekt soll funktional sein, was bedeutet, dass alle Kernfunktionen des Spiels (zum Bespiel Einzelspieler- und Mehrspielermodus, Steuerung der Schlange, Erzeugen von Nahrung, Kollisionserkennung) vollständig und fehlerfrei implementiert sein müssen. Tauglichkeit wird angestrebt, indem die Anwendung sowohl für Gelegenheitsspieler als auch für erfahrene Spieler ansprechend ist. Genauigkeit der Anwendung ist besonders bei der Kollisionserkennung und Punktevergabe wichtig, um ein faires Spielen zu gewährleisten. Interoperabilität wird geschaffen, indem die Anwendung in verschiedenen Webbrowsern und auf verschiedenen Desktop-Betriebssystemen getestet und angepasst wird. Die Einhaltung der Regelkonformität bedeutet, dass alle relevanten Webstandards und Datenschutzbestimmungen eingehalten werden sollen. Schließlich ist Sicherheit von Bedeutung, weil fehlerhafte Implementationen Sicherheitslücken darstellen und die Nutzer beziehungsweise deren Hardware gefährden.

## 6.2. Zuverlässigkeit

Ziel ist es, eine zuverlässige Anwendung zu schaffen, die stabil läuft und selten abstürzt. Durch Tests und darauffolgende Updates wird die Software etappenmäßig ausgereift. Die Anwendung soll fehlertolerant sein, indem sie unerwartete Eingaben oder Fehler abfängt und angemessen darauf reagiert.

## 6.3. Effizienz

Unsere Software soll effizient sein, indem es ressourcenschonend arbeitet und keine unnötigen Systemkapazitäten der Spieler verbraucht. Um Frustration zu vermeiden, müssen kurze Ladezeiten, möglichst wenig Netzwerkverzögerung (“Ping”) und eine flüssige Spielmechanik gewährleistet werden. Der Ressourcenbedarf soll minimiert werden, um eine breite Nutzerbasis zu erreichen, auch auf älteren, leistungsschwächeren Geräten.

## 6.4. Benutzbarkeit

Benutzbarkeit wird durch eine intuitive und benutzerfreundliche Benutzeroberfläche geschaffen, die Spielregeln und die Bedienung sollen leicht verständlich und an das klassische Spiel “Snake” angelehnt werden, damit neue Spieler das Spielprinzip schnell erfassen können. Die Bedienung sollte reaktionsschnell sein, mit Unterstützung für verschiedene Eingabemethoden, beispielsweise durch eine separate Maus- und Tastatur-Steuerung.

## 6.5. Übertragbarkeit

Das Projekt soll übertragbar sein, indem es auf den verschiedenen Plattformen gleichermaßen durch einen Webbrowser erreicht werden kann. Installierbarkeit ist somit nicht relevant da keine Installation nötig ist, sondern direkt im Browser das Spiel gespielt werden kann. Einzelne Programmkomponenten sollten der Austauschbarkeit wegen modular sein.

## 6.6. Wartbarkeit

Der Code wird wartbar gestaltet, indem er gut strukturiert und dokumentiert wird. Analysierbarkeit wird durch Protokollierungs- und “Debug”-Funktionen gewährleistet, um Fehler und Leistungsprobleme zu identifizieren. Der Code soll veränderlich sein, um zukünftige Änderungen und neue Funktionen zu ermöglichen. Testbarkeit ist durch einfaches Aufsetzen des Spiels sichergestellt.

# Übersicht der relevanten Daten

Welche Daten, Dateien, Datenbanken, Tabellen und Felder sind relevant?

**(Alexander)**

Das Online Snake Repository ([Link zu GitHub](https://github.com/SnakeOnlineOrganization/SnakeOnline)) beinhaltet alles, was der Client braucht, um spielen zu können. Der Server-seitige Code ist in einem anderen Privaten Repository.

# Glossar

**(Marcel)**

**Anwendungsfall:** Sammlung aller Szenarien zur Erreichung eines Ziels aus Sicht des Akteurs.

**Funktionale Anforderung:** Eine funktionale Anforderung ist eine Anforderung bezüglich eines Ergebnisses oder eines Verhaltens, das von einer Funktion eines Systems bereitgestellt werden soll.

**Nicht-Funktionale Anforderung:**Nichtfunktionale Anforderungen beschreiben Aspekte, die typischerweise mehrere oder alle funktionalen Anforderungen betreffen bzw. Überschneiden.

**Client:** Der “Client” bezeichnet die Anwendung, die auf dem Gerät des Nutzers ausgeführt wird und dazu dient, eine Verbindung zum Server aufzubauen und die Benutzerschnittstelle anzuzeigen, z.B. für das Spiel und die Interaktion.

**Server:** Der “Server” ist die zentrale Anwendung, die die Geschäftslogik bereitstellt, Anfragen vom Client verarbeitet und die konsistente Verwaltung der Spieldaten sicherstellt.

**Repository:** Das “Repository” beschreibt ein zentrales Verzeichnis, in dem der Quellcode, Konfigurationsdateien und andere relevante Dateien eines Projekts versioniert und für die Zusammenarbeit bereitgestellt werden.

**Frontend:** Bezeichnet die Komponente des Systems, die für die Darstellung der Benutzeroberfläche und die Interaktion mit dem Nutzer verantwortlich ist.

**Backend:** Das “Backend” ist die Server-Anwendung, die auf einem entfernten, nicht durch einen Nutzer erreichbaren Computer läuft und ausschließlich zur Verwaltung der Spiel-Daten, z.B. Bestenlisten und den Mehrspieler-Räumen dient.

**WebSocket:** Ein Kommunikationsprotokoll, das eine bidirektionale, persistente Verbindung zwischen Client und Server ermöglicht, oft genutzt für Echtzeit-Anwendungen wie Online-Spiele.

**Lobby:** Ein virtueller Raum, in dem Spieler vor dem Start eines Spiels zusammenkommen können, um Einstellungen vorzunehmen oder Mitspieler auszuwählen.

**Session:** Eine zeitlich begrenzte Instanz eines Spiels oder einer Verbindung, die spezifisch für einen Nutzer oder eine Nutzergruppe gilt.

**Leaderboard:** Eine Bestenliste, die die Spieler anhand ihrer Punktzahl, Erfolge oder anderer Kriterien rangiert.

**Ping:** Die Latenzzeit zwischen einer Anfrage des Clients und der Antwort des Servers, üblicherweise in Millisekunden gemessen, wichtig für reaktionsschnelle Spiele.

**HomePage:** Die “HomePage” beschreibt das Hauptmenü der Web-App und damit auch die Startseite der Client-Anwendung.

**GamePage:** Die “GamePage” ist die Seite, auf der das Spiel angezeigt werden soll sobald es gestartet wurde und dient als eindeutige Abgrenzung von der Startseite.

# Ausfüllhilfe

**Zieldefinition**  
Zieldefinition wie sie im Projektantrag zu finden ist. Bei Revisionen anzupassen.

## 9.1. Anwendungsfälle

**ID**  
Systemische und eindeutige Nummerierung der Anwendungsfälle.

**Titel**  
Beschreibender Name des Anwendungsfalls für die kurze Bezugsname im Dokument.

**Beschreibung**  
Kurze Beschreibung des Anwendungsfalls in einem oder zwei Sätzen.

**Akteure**  
Personen, Systeme oder Organisatorische Einheiten die in diesem Anwendungsfall aktiv sind.

**Vorbedingungen**  
Zustand aller relevanten Systeme zu Beginn des Anwendungsfalls.

**Nachbedingungen**  
Zustand aller relevanten Systeme beim Abschluss des Anwendungsfalls im gewünschten Ablauf.

**Ablauf/Main Path**   
Der erwartete oder gewöhnliche Verlauf des Anwendungsfalls

**Erweiterung/Alternative Path**  
Ablauf für definierte Abweichungen vom Ablauf

**Priorität**  
Position in der Umsetzungsreihenfolge der Anwendungsfälle.

**Kritikalität**  
Eigenschaft, ob ein Anwendungsfall den Projekterfolg verhindern kann.

## 9.2. Anforderungen

**ID**  
Systemtische und eindeutige Nummerierung der Anwendungsfälle inklusive ihres Typs.

**Titel**   
Beschreibender Name des Anwendungsfalls für die kurze Bezugsname im Dokument.

**Beschreibung**  
Kurze Beschreibung der Anforderung entsprechend der Satzschablone.

**Priorität**  
Position in der Umsetzungsreihenfolge der Anforderungen.

**Kritikalität**  
Eigenschaft, ob eine Anforderung den Projekterfolg verhindern kann.

# Anmerkung

**Dieser Bereich ist NICHT Teil der offiziellen Anforderungsdokumentation.**

*“Für die Bewertung der Aufgabe ist es zwingend notwendig, dass ihr die Ausarbeitung einzelner Anwendungsfälle und Anforderungen gleich in der Gruppe verteilt und in den Übersichten vermerkt, welche Person was geschrieben hat.” - übernommen aus Moodle-Arbeitsauftrag “Anforderungsdokumentation”*

Alle Inhalte des Dokuments sind in Gruppen-Arbeit ausgearbeitet worden, deshalb ist die Abgrenzung, welcher Gruppenteilnehmer was geschrieben und wer die Einträge nochmals abgeändert hat, schwierig. Deshalb haben manche Gruppenmitglieder weniger Einträge, weil sie z.B. mehr Anforderungs-Beschreibungen überarbeitet haben, als neue Anforderungen zu erstellen. Die Arbeitsteilung ist zudem problematisch, da unterschiedliche Erfahrung im Gebiet von Design, Programmieren oder Ähnlichem zu unterschiedlichem Arbeitstempo führen, und deshalb ein größerer Anteil der Arbeit durch erfahrenere Mitglieder erledigt wird.

**Marcel**  
**i.A. Online Snake Team**