Quiz

Spezifikation der Softwareanforderungen

Version 3.0

Inhaltsverzeichnis

1. Vorwort 4

1.1 Ziel 4

1.2 Geltungsbereich 4

1.3 Definitionen, Akronyme, und Abkürzungen 4

1.4 Referenzen 4

2. Anforderungsbeschreibung 5

2.1 Use-Case Modell 5

3. Spezielle Anforderungen 6

3.1 Use-Case Beschreibung (nur ausgewählte charakteristische Use Cases) 6

3.1.1 Konzeptioneller Aufbau des User Interface 6

3.1.2 Paket: Neues Spiel 9

3.1.3 Paket: Spiel-Informationen 11

3.1.4 Paket: Spiel 12

3.2 Zusätzliche Anforderungen 19

3.2.1 Fehlerauswertung 19

3.2.2 Benutzerbarkeit 19

3.2.3 Leistung 19

3.3 Das Quiz Spiel als Tabelle 21

Spezifikation der Softwareanforderungen

# Vorwort

## Ziel

Ziel dieses Dokumentes ist es, funktionale und qualitative nicht funktionale Anforderungen für die Software Quiz darzustellen und zu beschreiben.

Das Dokument beschreibt zunächst alle möglichen Spielsituationen und die dazu gehörigen Anwendungsfälle in einem Diagramm. Weiterhin werden alle Anwendungsfälle einzeln textuell beschrieben, sodass genaue Informationen zu den jeweiligen Spielfällen gefunden werden können. Auch enthält dieses Dokument zusätzliche qualitative nicht funktionale Anforderungen.

## Geltungsbereich

PC Software für das Spiel Quiz ohne Android Smartphone Anbindung und ohne Server Anbindung.

## Definitionen, Akronyme, und Abkürzungen

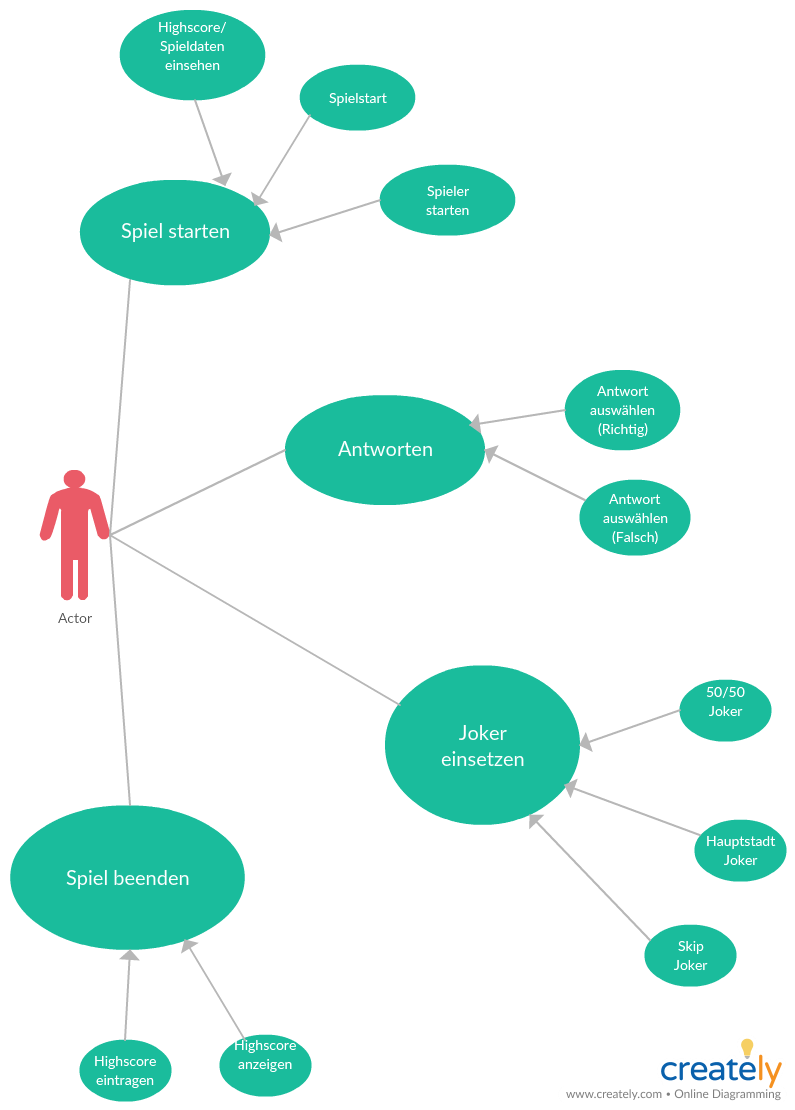
Siehe Glossar.

## Referenzen

* Vision
* Glossar

# Anforderungsbeschreibung

## Use-Case Modell



Die nachfolgenden Spielregeln erläutern den Zusammenhang der Use Cases :

* Zunächst kann man ein Spiel starten. Dies beinhaltet, dass man die Spielmodus auswählt
* Danach kommt der wichtigste Use Case - Frage beantworten.
* Während der Spieler eine Frage beantwortet kann man das Spiel beenden
* Weiterhin kann der Spieler auch einen Joker benutzen
* Nachdem das Spiel dann beendet wurde wird dem Spieler die Wertung für dieses Spiel angezeigt.  
  Wenn ein Spieler ein neue Höchstleistung erzielt hat wird dem Spieler hier auch die Möglichkeit gegeben einen Namen in die Bestenliste einzutragen.

# Spezielle Anforderungen

## Use-Case Beschreibung (nur ausgewählte charakteristische Use Cases)

### Konzeptioneller Aufbau des User Interface (gilt für alle Use Cases)

Hier ist die grafische Oberfläche, die dem Spieler zur Verfügung stehen soll grob skizziert. Der Aufbau der GUI kann sich allerdings je nach technischen Restriktionen noch von dem Hier eingetragenen unterscheiden.

### Paket: Spiel starten

*Highscores/Spieldaten einsehen*

|  |  |
| --- | --- |
| Name des Use-Case | Highscores/Spieldaten einsehen |
| Nummer | 1 |
| Paket | 1 |
| Kurzbeschreibung | Namen und Highscores werden angezeigt |
| Beteiligte Akteure | Alle Spieler |
| Vorbedingung | Start des Programms |
| Nachbedingung | Spieldaten werden angezeigt, Paket: Spiel starten |
| Detaillierter Ablauf | Durch das Drücken des Highscores-Button wird die Liste mit Namen und Punkte sortiert angezeigt. |
| Priorität | 2 |

#### *Spiel konfigurieren (= Spiel starten)*

|  |  |
| --- | --- |
| Name des Use-Case | Spiel konfigurieren |
| Nummer | 1 |
| Paket | 1 (Neues Spiel) |
| Kurzbeschreibung | Ein neues Spiel beginnen und Spieldaten eingeben (Modus, Schwierigkeitsgrad etc.) |
| Beteiligte Akteure | Alle Spieler |
| Vorbedingungen | Start des Programms |
| Nachbedingungen | Eingegebene Daten in Code setzen, Auswahl des Codes für die entsprechenden Optionen. |
| Detaillierter Ablauf | Sofort nach dem Starten des Spiels kann der Spieler die Optionen (Modus, Schwierigkeit etc. einstellen. Nachdem der Spieler das Spiel konfiguriert hat, kann er durch das starten. |
| Priorität | 1 |

#### *Spieler starten*

|  |  |
| --- | --- |
| Name des Use-Case | Spieler starten |
| Nummer | 2 |
| Paket | 1 |
| Kurzbeschreibung | Spiel wird mit den eingestellten Konfiguration gestartet |
| Beteiligte Akteure | Alle Spieler |
| Vorbedingungen | Man hat ein neues Spiel gestartet |
| Nachbedingungen | Paket: Antwort auswählen, Paket: Einsatz von Joker |
| Detaillierter Ablauf | Durch das drücken des Start Buttons wird das Spiel gestartet |
| Priorität | 1 |

### Paket: Antworten

#### *Antwort auswählen (Richtig)*

|  |  |
| --- | --- |
| Name des Use-Case | Antwort auswählen (Richtig) |
| Nummer | 1 |
| Paket | 2 |
| Kurzbeschreibung | Der Spieler wählt die richtige Antwort |
| Beteiligte Akteure | Spieler |
| Vorbedingungen | Spiel gestartet |
| Nachbedingungen | Neue Frage wird gestellt, Paket: Antworten |
| Detaillierter Ablauf | Spieler klickt den Button mit der richtigen Antwort |
| Priorität | 1 |

*Antwort auswählen (Falsch)*

|  |  |
| --- | --- |
| Name des Use-Case | Antwort auswählen (Falsch) |
| Nummer | 1 |
| Paket | 2 |
| Kurzbeschreibung | Spieler wählt die falsche Antwort aus |
| Beteiligte Akteure | Spieler |
| Vorbedingung | Spiel wird gestartet |
| Nachbedingung | Spiel ist beendet. Paket: Spiel beendet |
| Detaillierter Ablauf | Durch das drücken eines Button mit der falschen Antwort wird das Spiel sofort beendet. |
| Priorität | 1 |

### Paket: Einsatz von Jokern

#### *Spiel-Joker auswählen*

|  |  |
| --- | --- |
| Name des Use-Case | Spielzug durchführen |
| Nummer | 1 |
| Paket | 3 |
| Kurzbeschreibung | Der Spieler betätigt ein Joker |
| Beteiligte Akteure | Spieler |
| Vorbedingungen | Spiel wurde bereits gestartet |
| Nachbedingungen | 50/50 Joker, Hauptstadt-Joker, Skip-Joker |
| Detaillierter Ablauf | Der Spieler kann während eines Spielzugs einen von drei möglichen Jokern auswählen. |
| Priorität | 1 |

*50/50 Joker*

|  |  |
| --- | --- |
| Name des Use-Case | 50/50 Joker |
| Nummer | 2 |
| Paket | 3 |
| Kurzbeschreibung | Durch das Betätigen des Jokers werden 2 Antwortmöglichkeiten aussortiert |
| Beteiligte Akteure | Spieler |
| Vorbedingung | Spiel wurde gestartet, Joker muss zur Verfügung stehen |
| Nachbedingung | Zwei Antwortmöglichkeiten sollten zur Verfügung stehen, 1 Sekunde weniger zum Antworten für die nächste Frage. Paket: Antwort auswählen. |
| Detaillierter Ablauf | Durch das Betätigen des Jokers werden zwei der vier Antwortmöglichkeiten aussortiert und es stehen nurnoch zwei Antwortmöglichkeit zur Verfügung. Eines davon muss die richtige Antwortmöglichkeit sein. Der Zeitraum für das Antworten wird um 1 Sekunde reduziert. |
| Priorität | 1 |

*Hauptstadt-Joker*

|  |  |
| --- | --- |
| Name des Use-Case | Hauptstadt-Joker |
| Nummer | 2 |
| Paket | 3 |
| Kurzbeschreibung | Durch das Betätigen des Jokers wird dem Spieler die Hauptstadt angezeigt |
| Beteiligte Akteure | Spieler |
| Vorbedingung | Spiel wurde gestartet, Joker muss zur Verfügung stehen |
| Nachbedingung | Hauptstadt wird angezeigt, 1 Sekunde weniger zum Antworten für die nächste Frage. Paket: Antwort auswählen. |
| Detaillierter Ablauf | Durch das Betätigen des Jokers wird die Hauptstadt der richtigen Antwort angezeigt. Der Zeitraum für das Antworten wird um 1 Sekunde reduziert. |
| Priorität | 1 |

*Skip-Joker*

|  |  |
| --- | --- |
| Name des Use-Case | Skip-Joker |
| Nummer | 2 |
| Paket | 3 |
| Kurzbeschreibung | Durch das Betätigen des Jokers wird die Frage übersprün |
| Beteiligte Akteure | Spieler |
| Vorbedingung | Spiel wurde gestartet, Joker muss zur Verfügung stehen |
| Nachbedingung | Nächste Frage wird gestellt, 1 Sekunde weniger zum Antworten für die nächste Frage. Paket: Antwort auswählen. |
| Detaillierter Ablauf | Durch das Betätigen des Jokers wird die Frage durch die nächste ersetzt. Der Zeitraum für das Antworten wird um 1 Sekunde reduziert. |
| Priorität | 1 |

#### *8. Paket: Spiel beendet*

#### *Highscore anzeigen*

|  |  |
| --- | --- |
| Name des Use-Case | Highscore |
| Nummer | 1 |
| Paket | 4 |
| Kurzbeschreibung | Spieler sammelt Punkte. Höchstpunktzahl wird als Highscore gespeichert |
| Beteiligte Akteure | Spieler |
| Vorbedingungen | Spiel ist beendet, |
| Nachbedingungen | Highscore eintragen, Paket: Neues Spiel |
| Detaillierter Ablauf | Spiel wurde beendet in dem der Spieler alle Fragen beantwortet hat oder er eine Frage falsch beantwortet hat. Dem Spieler werden daraufhin seine erreichten Punkte gezeigt. Wenn es eine neue höchstpunktzahl ist wird sie als neuer Highscore gespeichert. |
| Priorität | 2 |

#### 

*Highscore eintragen*

|  |  |
| --- | --- |
| Name des Use-Case | Highscore eintragen |
| Nummer | 2 |
| Paket | 4 |
| Kurzbeschreibung | Namen kann mit den erzielten Punkten eingetragen werden |
| Beteiligte Akteure | Spieler |
| Vorbedingung | Spiel ist beendet, Highscores werden angezeigt |
| Nachbedingung | Namen und Punkte werden eingetragen, danach Paket: Spiel starten |
| Detaillierter Ablauf | Man kann den Namen eintragen und nach dem Betätigen wird der eingegebene Name und die erreichten Punktzahl den Highscores einsortiert |
| Priorität | 2 |

## Zusätzliche Anforderungen

### Fehlerauswertung

Alle Systemfehler werden protokolliert. Fatale Systemfehler sollen zu einem ordnungsgemäßen Herunterfahren des Systems führen. Die Fehlermeldungen enthalten eine Beschreibung des Fehlers, einen Fehlercode, einen Daten- und einen Zeitstempel.

### Benutzerbarkeit

Das Programm soll einfach zu bedienen, leicht verständlich und übersichtlich gestaltet sein, so dass Kinder ab 3 Jahren es ohne großen Aufwand erlernen können.

### Leistung

Das System muss das wechseln von einer Frage zu einer anderen in weniger als 1/Bildrate des Zielsystems Sekunden erreichen können. (hier setzen wir die Bildrate des Zielsystems auf 60 hz)

Eine eventuelle zwischengeschaltete Animation sollte ein festlegbare Zeit haben und es ermöglichen, dass der Zeitpunkt ab dem der Spieler die vollständige Frage sieht bestimmbar ist.

Zudem dar die Animation nicht langsam einblenden sondern muss die Flagge sofort einblenden.

#### *Verwendung von bekannten Technologien*

Für die Umsetzung wird Java 1.8 verwendet.

Die JavaFX Bibliothek wird für die Umsetzung der GUI verwendet.

Die Softwareentwicklung muss konform mit dem Rational Unified Process (RUP) sein.

Highscores sollen über eine relationale Datenbank gespeichert werden.

XML- Dateien sollen die Daten über die Länder beinhalten.

#### *Programmierregeln*

Das Programm wird konform zu den Regeln der Java-Programmierung (Java Code Conventions) umgesetzt.

#### *Wiederverwendbarkeit*

Die Architektur soll so gestaltet werden, dass ein modulares System mit austauschbaren Komponenten entsteht.

### Hardware Anforderungen

* 4 GB RAM und 1 GB Disk Space.
* Microsoft Windows 7.