Software Requirements Specification Template

Author: Uwe van Heesch

Version: 1 (21/09/16)

1 Introduction 1

1.1 Overall Description 1

1.2 User Classes and Characteristics 1

1.3 Operating Environment 1

1.4 Design and Implementation Constraints 2

1.5 Product Functions 2

*2* *Domain Model* 2

*3* Use-case Descriptions 2

3.1 Use case 1 2

3.1.1 Fully-dressed use case description 2

3.1.2 System Sequence Diagram (optional) 2

3.1.3 Operation Contracts (optional) 2

3.2 Use case 2 ( and so on) 2

4 Other functional requirements (optional) 3

5 Non-functional Requirements 3

5.1 Performance Efficiency 3

5.2 Security 3

5.3 Reliability (and so on) 3

6 User interface sketches (optional) 3

# Introduction

## Overall Description

De opdracht **Parola** is een quiz-spel progragemeren voor MEC Games Met het oog op de huidige populariteit van pubquizzes en de behoefte aan een robuuste, flexibele en multi-device ondersteunde applicatie, heeft MEC Games besloten om Parola te ontwikkelen. Deze applicatie wordt ontworpen om beschikbaar te zijn op de meest gangbare devices, waardoor gebruikers op elk gewenst moment en vanaf elke locatie kunnen deelnemen aan deze quizzes.

Met Parola beoogt MEC Games de volgende voordelen en doelstellingen te bereiken:

- Breed Toegankelijk: Door de applicatie beschikbaar te maken op de meest gangbare devices, kunnen meer mensen deelnemen en genieten van de quizzen die door de applicatie worden aangeboden.

- Flexibiliteit: Door de applicatie te ontwerpen volgens de richtlijnen van Larman en met een nadruk op objectgeoriënteerde analyse, zal Parola flexibel en aanpasbaar zijn, wat toekomstige uitbreidingen en aanpassingen vergemakkelijkt.

- Doel: De kernambitie van Parola is om een platform te bieden waarop pubquizzes kunnen worden gespeeld.

Het doel van dit Software Requirement Specification (SRS) document is om de eindgebruikers en het domein van de Parola-applicatie in kaart te brengen en de functionaliteiten van de eindapplicatie zo vroeg mogelijk vast te leggen. Dit zal gebeuren via een gedetailleerd usecase-diagram, gevolgd door uitgebreide beschrijvingen van de usecases. Elk van deze usecases zal worden verduidelijkt met een system sequence diagram om de flow van de operaties te illustreren. Het overkoepelende doel van dit SRS-document is om een helder en volledig beeld te schetsen van de functionaliteiten die Parola zal bevatten en hoe gebruikers met de applicatie zullen interacteren. Bestemd voor ontwikkelaars, testers, projectmanagers en andere belanghebbenden, biedt dit SRS-document inzichten en views om alle stakeholders een grondig begrip van de onderliggende architectuur en functionaliteiten van Parola te geven.

## 1.2. Definitions, acronyms, and abbreviations

Het doel van Definitions, acronyms, and abbreviations is om een ​​duidelijke en consistente terminologie aan te geven binnen de casus, wat verwarring en misverstanden voorkomt.

| **Term** | **Definition** |
| --- | --- |
| UC | Use Case |
| RDB | Relational database |
| NFR | Non-functional Requirements |
| FR | Functional Requirements |

1. Tabel 1: Definitions, acronyms, and abbreviations

## User Classes and Characteristics

| **Actor** | **Omschrijving** |
| --- | --- |
| Speler | Een individuele gebruiker die deelneemt aan de quiz. |
| Medewerker van MEC Games | Een medewerker (of administrator) die verantwoordelijk is voor het beheren van de vragenlijst binnen de Parola-applicatie |
| Externe Software Library | Een dat door de Parola-applicatie wordt gebruikt om het eind ingevoerde woord te controlleren. |
| Betaalsysteem | Een extern systeem waarmee gebruikers credits kunnen kopen voor het quiz-spel |

1. Tabel 2: User Classes and Characteristics

## Operating Environment

Deze quizapplicatie is bedoeld voor gebruik op een standaard pc of laptop. Parola werkt op Windows, MacOS en Linux. Parola werkt soepel. De enige vereiste is dat Java, met name versie 17 of nieuwer.

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: hardware (e.g. specific mobile platforms), specific technologies, tools, and databases to be used; interfaces to other applications; programming language required; or communications protocols>*

## Product Functions

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. In most cases, this section will primarily contain a use case diagram and brief use case descriptions >*

# *Domain Model*

<Provide a diagram showing important real-situation conceptual classes in the application domain. Do NOT include software classes. Describe each of the conceptual classes in a glossary.>

# Use-case Descriptions

*<In this section, each use-case is described in detail, optionally accompanied by a system sequence diagram (SSD) and operation contracts. Make sure that the use case descriptions are consistent with the domain model and the use case diagram from Section 1.5>*

## Use case 1

<Don’t really say “Use case 1.” State the use-case name instead.>

### Fully-dressed use case description

*<Provide a fully-dressed use-case description in the format you know from the OOAD course>*

### System Sequence Diagram (optional)

*<In case the use-case entails complex scenarios, you may decide to create a system sequence diagram showing events generated by external actors, the order of events and inter-system events. All systems are treated as a black box>*

### Operation Contracts (optional)

*<If the use case contains complex manipulations of domain objects, you may decide to specify operation contracts for all system operations included in the use case/ SSD.>*

## Use case 2 ( and so on)

…

# Other functional requirements (optional)

*<Use this section to describe functional requirements that cannot be expressed in the shape of use cases, for instance because they do not concern goal-oriented interactions of an actor with the system.>*

|  |  |
| --- | --- |
| Code | Description |
| FR1 | Het system moet gebruik maken van een externe software library om het ingevoerde woord te controleren. |

# Non-functional Requirements

*<Describe non-functional requirements in this section. Please refer to existing software quality models like ISO\_IEC\_IEEE\_25010 or FURPS+.>*

|  |  |  |
| --- | --- | --- |
| Code | Description | Type |
| NFR1 | Tijdens een quiz, moet het systeem binnen 0.5 seconden de volgende vraag tonen vanaf het moment dat de speler de huidige vraag heeft beantwoord. | Performance |
| NFR2 | Het overstappen naar een alternatief – al bestaand – systeem voor puntentelling, moet niet langer duren dan 10 minuten. | Usability |
| NFR3 | Het systeem moet de mogelijkheid voor andere talen ondersteunen | Supportability |