Software Design Description Template

Author: Lodewijk Klassen & Job Nieling

Version: 1 (20-03-2024)

Inhoud

[1 Introduction 1](#_Toc161831578)

[1.1 Algemene Beschrijving 1](#_Toc161831579)

[1.2 Doel van dit document 1](#_Toc161831580)

[1.3 Definitions, acronyms, and abbreviations 1](#_Toc161831581)

[2 Architectural Overview 1](#_Toc161831582)

[3 Detailed Design Description 1](#_Toc161831583)

[3.1 Deployment Diagram 2](#_Toc161831584)

[3.1.1 Design Decisions related to deployment 2](#_Toc161831585)

[3.2 Design Sub-System A 2](#_Toc161831586)

[3.2.1 Design Class Diagram 2](#_Toc161831587)

[3.2.2 Sequence Diagrams 2](#_Toc161831588)

[3.2.3 Activity and State Diagrams 2](#_Toc161831589)

[3.2.4 Ontwerpbeslissingen voor het subsysteem 3](#_Toc161831590)

[3.3 Ontwerp Subsysteem B (en zo verder) 3](#_Toc161831591)

[3.4 Databasedesign 3](#_Toc161831592)

[3.4.1 Design decisions related to the database 4](#_Toc161831593)

# Introduction

## Algemene Beschrijving

Parola is een quiz-applicatie ontwikkeld door MEC Games, bedoeld om beschikbaar te zijn op verschillende gangbare apparaten. Het doel van de software is om gebruikers een leuke en educatieve quizervaring te bieden. De applicatie streeft ernaar om quizzen van verschillende categorieën aan te bieden, waarbij spelers punten kunnen verdienen door vragen correct te beantwoorden en woorden te vormen.

## Doel van dit document

Het doel van dit document is om een gedetailleerde beschrijving te geven van het ontwerp van de Parola-software, inclusief architectonisch overzicht, implementatiebeslissingen, en ontwerpbeslissingen voor elk subsysteem. Dit document zal dienen als een handleiding voor ontwikkelaars die betrokken zijn bij het bouwen van de Parola-applicatie.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| Term | Beschrijving |
|  |  |
|  |  |

# Architectural Overview

*<Geef een overzicht op hoog niveau van het architectonisch ontwerp, bijvoorbeeld aan de hand van een architectuurschets. Zorg ervoor dat ten minste alle sub-systemen en links naar externe systemen worden weergegeven. De schets kan informeel zijn. Het gebruik van UML is niet vereist.>*

# Detailed Design Description

*<Deze sectie bevat gedetailleerde ontwerpdocumentatie van alle softwarecomponenten. De inhoud van deze sectie groeit iteratief tijdens de sprints. Aan het einde van elke sprint moeten de getoonde diagrammen consistent zijn.>*

## Deployment Diagram

*<Bied een UML-implementatiediagram waarin alle fysieke en virtuele knooppunten worden weergegeven die in het systeem worden gebruikt. Het diagram moet ook alle implementatie-artefacten bevatten die in het systeem worden gebruikt, zoals JAR- of WAR-bestanden, of webartefacten.>*

### Design Decisions related to deployment

*<Beschrijf alle ontwerpbeslissingen die worden gemanifesteerd in het implementatiediagram. Bijvoorbeeld de keuze van besturingssystemen, protocollen, distributie van componenten over sub-systemen en dergelijke.>*

## Design Sub-System A

<Geef een sectie voor elk subsysteem. Deze secties worden iteratief toegevoegd en verfijnd tijdens de sprints. Voorbeelden van subsystemen zijn Persistent Storage, Business Tier, Web Application, Webservice API. De onderstaande subsecties kunnen worden uitgebreid als je denkt dat dit nuttig is voor het beschrijven van het softwareontwerp. De onderstaande subsecties zijn alleen vereist voor objectgeoriënteerde subsystemen. Gebruik andere middelen om niet-OO-subsysteem te beschrijven (bijvoorbeeld Javascript-modules).>

### Design Class Diagram

<Object-oriented sub-systems should be described using a class diagram. If classes or interfaces are used across sub-systems, make sure you mention this in the description of the class diagrams. If your system entails layers, make sure you indicate this in the class diagram, e.g. by means of packages. For each class diagram, make sure you also mention the deployment artifact (from the deployment diagram) it is part of.>

### Sequence Diagrams

<Provide sequence diagrams for major object interactions within the sub-system. It is ok if sequence diagrams cross sub-system boundaries. Make sure you explain this in the description of the diagram. Sequence diagrams must be consistent with the class diagrams described above. Also, if sequence diagrams cover interaction with users, make sure the diagrams are consistent with SDDs you may have documented as part of the SRS.>

### Activity and State Diagrams

<Deze sectie is optioneel. Als het nuttig is, biedt dan activiteiten- en/of toestandsdiagrammen om complexe workflows en systeemstaattransities te beschrijven.>

Afbeelding met tekst, schermopname, diagram, Parallel

Automatisch gegenereerde beschrijving

### Ontwerpbeslissingen voor het subsysteem

*<Beschrijf alle ontwerpbeslissingen die zijn genomen voor het subsysteem. Geef ten minste beslissingsbeschrijvingen voor alle frameworks, bibliotheken en andere gebruikte technologieën. Andere beslissingen kunnen betrekking hebben op softwarepatronen, systeemstructuur, aangepaste principes of dergelijke.>*

## Ontwerp Subsysteem B (en zo verder)

…

## Databasedesign

We gebruiken voor dit project geen database. Hierdoor is deze niet van toepassing.

### Design decisions related to the database

Aangezien voor dit project alles “hard coded” is. Zal deze niet van toepassing zijn. Alles is gemaakt door middel van seeds en hard coded vragen, antwoorden etc.