DD-v0.1

Francesco Spangaro - Tosetti Luca - Francesco Riccardi

07 January 2024

POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to provide an exhausting and implementative description of the platform that will be implemented (CKB platform). In particular the document is focused on the description of the architectural styles and decisions that will be adopted, the modules that compose the platform and their interfaces. The document will contains also several details regarding the deployment choices, the runtime view of the core functionalites of the platform that will be used in it. The document contains some mockups of the user interface design. The document also covers the implementation, integration and testing processes required to implement correctly the CKB platform.

1.2 Scope

CodeKataBattle (CKB) is a platform which aims to give to Educators an easy-to-use experience, and let them propose homework and/or lessons in a new and fresh way. The main goal of the platform is to give the Students the possibility to improve and acquire new software developing skills by particpating to several battles in as many tournaments. The platform let Educators of the Students to create such tournaments and battles within them in order to challenge the Students to upload the best possible solution to the battle's problem. That solution will be then automatically evaluated by the platform which will give it a score, and eventually even by the Educator who created the battle, and will be associated to it a proper score. The platform also allow Educators to add several recognition badges for the work done by the students. This badges can be personalized by the Educators themselves.

From the architectural point of view we have decided to adopt a 3-Tier Client-Server architecture combined with a MSA server side, in addition to a MVC software architectural choice.

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

Term	Definition
3-Tier Architecture	→ A 3-Tier architecture in the field of informatic systems, very simply a software and hardware architecture in which a running application/platform is divided in three different modules or also called "layers" which are: Presentation Layer, Logic Layer, Data Layer.
Presentation Layer	→ The top layer of the 3-Tier architecture. It takes care of the interaction between the user and the application.
Logic Layer	→ The middle layer of the 3-Tier architecture. It takes care of implementing the real application logic that allows to the application/platform to actually work.
Data Layer	→ The bottom layer of the 3-Tier architecture. It takes care of all the data generated by the users or the application, and with which the application itself has to interact. The interactions can include queries, updates, deletions, ...
Microservice architecture	→ An architectural style that consist in the creation of an application/platform as a suite of small services, each one handling a part of the business logic of the application and that comunicates with each other throught lightweight protocols (such as HTTP).
Model-View-Controller	→ An architectural pattern used to develop the software logic of an application. This pattern consist in dividing the application in three different parts: View, Model and Controller.

Term	Definition
View	→ Part of the MVC pattern which takes care of the visualization of the data contained in the model and the interaction with the user.
Controller	→ Part of the MVC pattern that receives commands from the user, and execute them by modifying the View and/or Model parts.
Model	→ Part of the MVC pattern that gives to the Controller part, the methods to access the application's data and to modify them.

1.3.2 Acronyms

Acronym	Meaning
MSA	→ MicroServices Architecture
MVC	→ Model-View-Controller
RASD	→ Requirement Analysis and Specification Document
	→
	→
	→

1.3.3 Abbreviations

Abbreviation	Meaning
e.g.	→ Exempli gratia, latin phrase meaning "for example".
	→
	→
	→

Abbreviation	Meaning
	→

1.4 Revision history

- **Placeholder data**: version 1.0

1.5 Reference documents

UML official specification → `https://www.omg.org/spec/UML`

Sequence diagrams specification → `https://www.uml-diagrams.org/sequence-diagrams.html`

1.6 Document structure

- ***Section 1: Introduction***
  This section offers a brief description of the problem and the platform/application that will be developed in order to resolve it. It describes the major purpose of this document, a brief very brief recap of the domain described in detail in the RASD document. In addition, in this section are inserted definitions, acronyms and abbreviations used in the document, including its revision history and refereced documents or web pages.
- ***Section 2: Architectural Design***
- ***Section 3: User interface design***
  This section describes the user interface design of the platform. It contains several mockups of the interface that the Educators and Students will find. The presented mockups refers to the client-side experience.
- ***Section 4: Requirements traceability***
- ***Section 5: Implementation, Integration & Test plan***
- ***Section 6: Effort spent***
  This section contains all the information about the time spent by each group member in order to complete this document and its division by each section of the document.

2 Architectural Design

2.1 Overview: High-level components and interactions

2.2 Component view

2.3 Deployment view

2.4 Component interfaces

2.5 Runtime view

2.6 Selected architectural styles and patterns

2.7 Other design decisions

3 User Interface Design

4 Requirements traceability

5 Implementation, Integration & Test plan

6 Effort Spent

7 References