3

Spécification des exigences

3.1 Introduction

Purpose

The purpose of this document is to describe the requirement specifications for the project « Mini-Hearthstone » for software engineering students.

The intended audience of this specification includes the prospective developers of the tool, as well as the technical assessment personnel.

Document Conventions

None so far.

Intended Audience and Reading Suggestions

Project Scope

The software system to be produced is a simplified version of the Hearthstone online game, which will be referred to as « Mini-Hearthstone » thorough this document.

The Mini-Hearthstonesystem will allow players from different locations to confront each-other in short and intensive games.

References

1. IEEE Standard 830-1993 : IEEE Recommended Practice for Software Requirements Specifications

Overview

The rest of this document contains an overall description of the Mini-Hearthstonesoftware system (section 3.2), the specific functional requirements (section 3.4), and the non-functional requirements for the system 3.5.

3.2 Overall Description

Product Perspective

Hearthstone is a card game where two players confront each-other. The Mini-Hearthstonesoftware should allow players that are connected to the Internet to use their connected devices to play. Thus, « Mini-Hearthstone » is an online, electronic version of the card game.

While the system is distributed and organized in different components, players should perceive it as a single piece of software. Figure 31 presents the overall architecture of the software. Players interact with a Web Client, which uses the HTTP protocol to communicate with (at most) one Game Server. Servers use TCP/IP to communicate with a Database Management Server, which stores all software data.

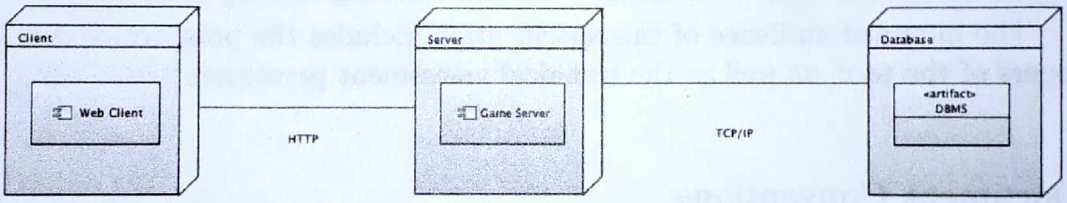

FIG. 31 : UML Deployment Diagram

Product Functions

The Mini-Hearthstonesoftware must provide two main functions :

1. Game creation : allowing two players to discover each other and start a game.
2. Game play : allowing players to actually play Mini-Hearthstoneuntil the victory of one of them.

User Classes and Characteristics

The Mini-Hearthstonesoftware has only one class of user : players. Players may have different levels : beginners, intermediate, or expert. However, independently from their level, players should use the same user interface to play against each other.

Operating Environment

The Mini-Hearthstonesoftware should operate on any popular and recent operating system : Linux, Windows, or MacOS. The Web Client should operate on any recent web browser : Firefox, Chrome, Safari, or Edge.

Design and Implementation Constraints

1. The Game Server must be developed in Java (version 1.8), using the Spring Framework[1].

2. The Client must be developed in TypeScript (version 3.1), using the Angular Framework[2].

3. All software artifacts must use a building too : Maven or Groovy for Java, npm for TypeScript.

4. Dynamic tests must use JUnit (version > 5.0) or its equivalent for other languages.

User Documentation

No user documentation is required for the first version of the software.

Assumptions and Dependencies

None until now.

3.3 External Interface Requirements

User Interfaces

Hardware Interfaces

The software does not interact directly with any hardware device.

Software Interfaces

The client part of the software must operate on web browsers, whereas the server part must interact with a database through the Java Persistence API (JPA).

1. `https://spring.io`
2. `https://angular.io`

Communications Interfaces

Communications between the client and the game server must use both, RESTful and Websockets Web services.

3.4 System Features

Game initialization

The Mini-Hearthstonesoftware must allow the setup of a game with two players and automatically prepare and distribute cards.

Description and Priority

See Chapter 2 (domain analysis) for further details.

Stimulus/Response Sequences

Functional Requirements

Game play

The Mini-Hearthstonesoftware must allow two players to play against each other until a winer is settled. See Chapter 2 (domain analysis) for further details.

3.5 Other Nonfunctional Requirements

Performance Requirements

1. The game must be *playable*, meaning that users must have fast feedback for their actions and delays due to communications/connection problems must be correctly held.

2. The Web Client must be able to execute on a personal computer with 4GB of RAM.

Safety Requirements

Security Requirements

Software Quality Attributes

Extensibility

The software must be extensible, it must be easy for developers to add new cards and heroes to the game.

Maintainability

1. The software must be readable and easy to maintain.
2. The Java source must respect Google's guidelines : `https://google-styleguide.googlecode.com/svn/trunk/javaguide.html`

Business Rules

3.6 Other Requirements

Appendix A : Glossary

Appendix B : Analysis Models

See Chapter 2 (domain analysis) for further details.

Appendix C : To Be Determined List