Fact Racing

Software Requirements Specification

For <Subsystem or Feature>

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20.10.2016 | 1.0 | Initial version | Michael Buchwald |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Overall Description 5

3. Specific Requirements 6

3.1 Functionality 6

3.1.1 Creating a game session 6

3.1.2 Joining a game session 6

3.1.3 Playing the game 6

3.2 Usability 6

3.2.1 Using a web browser 6

3.2.2 Familiarity with board games 6

3.2.3 Language 6

3.3 Reliability 6

3.3.1 Server uptime 6

3.3.2 MTBF 6

3.3.3 MTTR 6

3.4 Performance 6

3.4.1 Response time 6

3.4.2 Capacity 6

3.5 Supportability 6

3.5.1 Language support 6

3.6 Design Constraints 7

3.6.1 Languages 7

3.6.2 Spring Framework 7

3.7 On-line User Documentation and Help System Requirements 7

3.8 Purchased Components 7

3.9 Interfaces 7

3.9.1 User Interfaces 7

3.9.2 Hardware Interfaces 7

3.9.3 Software Interfaces 7

3.9.4 Communications Interfaces 7

3.10 Licensing Requirements 7

3.11 Legal, Copyright, and Other Notices 7

3.12 Applicable Standards 7

4. Supporting Information 8

4.1 Appendices 8

Software Requirements Specification

# Introduction

## Purpose

The purpose of this SRS is to document all specifications for “Fact Racing”. It is a web-based board game application developed with the Java programming language. An explanation for how the game will be designed will be given in upcoming chapters. This includes important characteristics like user synchronization and stability.

## Scope

This SRS applies to the whole application. It consists of a website that allows users to freely create game sessions that other users can join with a generated link. The game itself is a board game with a playing field that players can move across. The goal is to answer questions on certain fields to gain points or access to shortcuts. A player wins after they reached a certain amount of points or there are no questions left.

## Definitions, Acronyms, and Abbreviations

N/A

## References

Github: <https://github.com/Kahmul/TINF15B4MCJI>

Blog: <https://gamescobold.wordpress.com>

## Overview

The following chapters of this SRS will be addressing the design of the game, our demands and the technical realization.

# Overall Description

A web-based board game written in the Java programming language using the Spring Framework.

Players can create a room to start a game and other people can join it. No account required, the creator of the session is able to invite other people by providing the link.

The game consists of a 6×10 playing field and between 2-4 players moving on the board. A dice throw (1-6) determines how many fields a player can move at a time. Fields can have a question attached to them that the player currently on the field has to answer. Answering questions correctly awards you points that you need in order to win the game as well as allowing you to sometimes take shortcuts.

**Game Rules**

– Everybody starts on the same starting field.  
– Starting player is decided at random.  
– Each player has to throw a dice each turn unless they have been forced to skip a turn the round before.  
– If a player lands on a field with a question he has to answer it. The field determines whether it is an easy or a hard question.  
– Answering a question correctly awards the player with points depending on the difficulty of the question. Any shortcuts associated with the field are also allowed to be taken by the player.  
– Answering the question incorrectly results in a penalty for the player in question. This can vary between things like having to skip a turn the next round or receiving a point penalty.  
– Each time a player reaches the starting field again they are awarded points.  
– The game ends after a player has reached a certain amount of points set before the start of the game or when one deck of cards is empty at which point the player with the highest score wins.

# Specific Requirements

## Functionality

### Creating a game session

The user is able to create his own game session on the website. There he will be able to communicate with other users that have joined the game via chat. Users can be invited via a link or can randomly be assigned to a game if they wish.

### Joining a game session

Joining a game can be done either by being given a link to a specific session by the user who created it or by choosing to be assigned to a random currently open game session.

### Playing the game

Players that have joined a game session will end up playing together once the game is started. Users will act in turns throughout the game according to the rules described in chapter 2.

## Usability

### Using a web browser

The user is expected to be familiar with standard website design like buttons, chat, etc. and knows how to use them.

### Familiarity with board games

The user should have knowledge about standard board game components like moving across a playing field, throwing a dice and so on.

### Language

The language will be English. The user is expected to be familiar with it.

## Reliability

### Server uptime

Server uptime should be ensured about 80% of the time.

### MTBF

Should be as high as possible, but may be expected to be around 5 days.

### MTTR

Should be as low as possible, a few days at most.

## Performance

### Response time

The response time should be low, doesn’t have to be incredibly fast though, 1-2s are enough even while in-game.

### Capacity

Number of users at the same time should be good enough for about 20 games, so a 100 users at a time should be manageable.

## Supportability

### Language support

The programming language used will be Java, which will make sure that the application will be able to run on any operating system.

## Design Constraints

### Languages

The language use for the implementation will be Java.

### Spring Framework

The Spring MVC architiecture has to be used.

## On-line User Documentation and Help System Requirements

The design of the website will be intuitive to use. There will be a game explanation available though.

## Purchased Components

N/A

## Interfaces

### User Interfaces

Not yet determined.

### Hardware Interfaces

N/A

### Software Interfaces

N/A

### Communications Interfaces

N/A

## Licensing Requirements

Not yet determined.

## Legal, Copyright, and Other Notices

N/A

## Applicable Standards

Not yet determined.

# Supporting Information

## Appendices

Visit our github for more information (<https://github.com/Kahmul/TINF15B4MCJI>) and our blog for the current status of the project (<https://gamescobold.wordpress.com> ).

## Overall Use-Case-Diagram

