Software Requirements Specification (SRS) for Halma in Java Project

*Baseline version 0.1*

*Issued on : November 21, 2014*

Issued by : Kelvin, Inc.

Issued for : Java Client

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | November 21, 2014 | Kelvin | initial version |
|  |  |  |  |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following :

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date** | **Signature** |
| Kelvin | Project Manager |  |  |
| SomeoneFromClient | SomeRole |  |  |
|  |  |  |  |

**Table of Contents**

Contents

[Change History i](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665221)

[Document Approval i](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665222)

[Table of Contents ii](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665223)

[List of Figures iv](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665224)

[Chapter 1 1](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665225)

[INRODUCTION 1](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665226)

[1.1 Purpose 1](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665227)

[1.2 Scope 1](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665228)

[1.3 Definitions, Acronyms, and Abbreviations 1](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665229)

[1.4 References 1](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665230)

[1.5 Overview 2](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665231)

[Chapter 2 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665232)

[GENERAL DESCRIPTION 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665233)

[2.1 Product Perspective 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665234)

[2.2 Product Functions 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665235)

[2.3 User Characteristics 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665236)

[2.4 General Constraints 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665237)

[· Smart car simulation program will start automatically after the user execute the smart car simulation program. 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665238)

[2.5 Assumptions and Dependencies 3](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665239)

[Chapter 3 4](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665240)

[SPECIFIC REQUIREMENTS 4](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665241)

[3.1 External Interface Requirements 4](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665242)

[3.1.1 User interface 4](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665243)

[3.1.2. Hardware interfaces 4](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665244)

[3.1.3. Software interfaces 4](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665245)

[3.2 Functional Requirements 4](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665246)

[3.3 Use Cases 5](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665247)

[3.4 Classes/Objects 6](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665248)

[3.5 Non-Functional Requirements 6](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665249)

[3.6 Design Constraints 7](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665250)

[Chapter 4 8](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665251)

[ANALYSIS MODELS 8](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665252)

[4.1 Sequence Diagrams 8](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665253)

[4.1.1 Run Program 8](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665254)

[4.1.2 User Feedback 8](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665255)

[4.1.3 Maintain Program 9](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665256)

[4.1.4 Client Decision 9](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665257)

[4.2 State Diagrams 10](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665258)

[Chapter 5 11](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665259)

[CHANGE MANAGEMENT PROCESS 11](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665260)

[5.1 Process 11](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665261)

[5.2 Actor 11](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665262)

[5.3 Approval 11](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665263)

[Appendices 12](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665264)

[Appendix 1 12](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665265)

[Appendix 2 12](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665266)

[Appendix 3 12](file:///C:\Users\Kelvin\Downloads\SRS%20Template%20Final%20Cut.docx#_Toc403665267)

**List of Figures**

**Use Case.**….……………………..........................……………………………............................5

**Class Diagram**..…………………………………………………………………………………..6

**Sequence Diagram 1**…..……………………..........................……………………………........8

**Sequence Diagram 2**…..………………………………………………………………………....8

**Sequence Diagram 3**…...……………………..........................…………………………….......9

**Sequence Diagram 4**...…………………………………………………………………………...9

**State Diagram**...…………………………………………………………………………………10

**Process Diagram**…...……………………………………………………………………………

**Chapter 1**

**INRODUCTION**

*The content of this document is supposed to give overview of the Software Requirement Specification (SRS) that you write. Remember that by reading this document only, a software engineer can obtain enough information to design and implement software products.*

* 1. **Purpose**

This project consist of a game named halma game that can played with human or computer and it can show the high score of the game.

* 1. **Scope**

The purpose of this project is to analyze the requirements of, design, implement, and maintain the software for AnotherJava Client, which is designated specifically to help in monitoring Halma according to the requirements specified by the client.

Any activity directly relates to the purpose is considered to be in scope, otherwise it is considered to be out of scope. For instance, hardware failure is out of scope of this project.

The objectives of the project are mentioned as follows :

* Complete the project by the due date
* Complete the project within the budget
* Provide all deliverables identified in section 1.1.3 by the due date
* Meet all the requirements that are mentioned in the SRS
  1. **Definitions, Acronyms, and Abbreviations**

Here is the list of all assumptions that are made :

* This project is a component of a larger project
* This project will deliver only the software components of the larger project
* The hardware is not included in this project, and is handled as a different project. Hardware is assumed to be ready at least by the end of this project.
  1. **References**
* Meyer, Bertrand. Object-Oriented Software Construction 2nd Edition. California : ISE. Inc, 1997.
* Farrel, Joyce. Java Programmming 6th Edition. Boston : Cengage Learning, 2012.
  1. **Overview**

The first section of the SRS introducing what are our project that is a SmartCar application, Second section of the SRS discussing about our project description, Third section of the SRS discussing specific requirement about our project. Fourth section of the SRS discussing about our Analysis model and the last section discussing about change management process.

**Chapter 2**

**GENERAL DESCRIPTION**

1. **Product Perspective**

The Client and users will be given and the client and user can access the program and can see the halma game.

1. **Product Functions**

This halma game can be played human with human or human with computer and can shown it’s high score and save function

1. **User Characteristics**

General Users / Guest: They will be in a position to permit access to the users in the Internet and acknowledge their account status.

Administrators: They are the core users and are able to add new users to the system and permit them to access the Internet resources. They can also view in real time what a user is performing right now. They can also get the overall report of the user sessions.

1. **General Constraints**

* Halma game program will start automatically after the user execute the smart car simulation program.
* Only Administrator and group members can control and change the smart car simulation program.

1. **Assumptions and Dependencies**

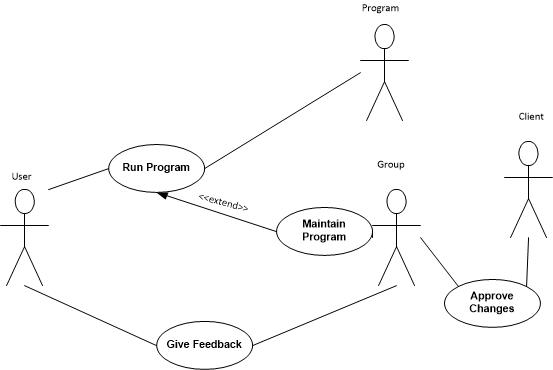
The aim of the game is to be the first to player to move all pieces across the board and into opposing corner, within area delineated by the heavy line.  Should one or more of the squares in the target area contain a piece belonging to another player, this does not prevent a player from winning.  In this case, the game is won when all the available points within the area are occupied. A toss of a coin decides who starts. Players take turns to move a single piece of their own colour. The piece may either be simply moved into an adjacent square OR it may make one or more hops over other pieces. Where a hopping move is made, each hop must be over an adjacent piece and into the vacant square directly beyond it. A hop may be over any coloured piece including the player's own and can proceed in any one of the eight directions. After each hop, the player may either finish or, if possible and desired, continue by hopping over another piece. It does occasionally happen that a player is able to move a piece all the way from the starting corner across the board and into the opposite corner in one turn! Pieces are never removed from the board. It is permitted to move a piece into any square on the board including squares in areas belonging to other players but, once a piece has reached the opposite area, it may not be moved out of the area but it may move within the area.

**Chapter 3**

**SPECIFIC REQUIREMENTS**

1. **External Interface Requirements**

Simulator Screen: The screen show how the simulation works (point, board).

1. **Use Cases**
2. **Classes/Objects**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field |  | HalmaBoard |  | HalmaButton |  | HalmaView |
| fieldType = int |  | getWidth = int |  | HalmaButton = public |  | board = HalmaBoard |
| homeBaseFromPlayer = int |  | getHeight = int |  | getField = field |  | frame = Jframe |
| endBaseFrom Player = int |  | getField = field |  | setField = void |  | LabelInfo = JLabel |
| xPosition = int |  | setField = void |  |  |  | moveStatus = int |
| yPosition = int |  | createInitialField = void |  |  |  | displayGetPlayerName = void |
| isAsleep = bool |  | move = void |  |  |  | displayAbout = void |
| setAsleep = void |  | forceUpdate = void |  |  |  | update = void |
| setFieldType = void |  | isGameOver = bool |  |  |  | rewardwinner = void |
| setHomeBaseFrom Player = void |  | showField = void |  | HalmaMove |  | getMoveStatus = int |
| setgetEndBaseFromPlayer = int |  | isMoveValid = bool |  | rating = int |  | setMoveStatus = void |
| setxPosition = void |  | checkSingleJump = bool |  | fromXPosition = int |  |  |
| setyPosition = void |  | checkMultiJump = bool |  | fromYPosition = int |  | HalmaViewMenu |
|  |  | getLastFreeField = field |  | toXPosition = int |  | menuBar = JMenuBar |
| HalmaController |  | findDOubleSubpath = bool |  | toYPosition = int |  | menu = Jmenu |
|  |  | checkLoop = bool |  | getFromXPosition = int |  | getNull = void |
|  |  | isFieldVisited = bool |  | setFromXPosition = void |  |  |
| actionPerformed = void |  | isDirectionValid = bool |  | getFromYPosition = int |  | Move |
| setupPlayer = void |  | getRating = int |  | setFromYPosition = void |  |  |
|  |  | setRating = void |  | getToXPosition = int |  |  |
| HalmaSetting |  |  |  | setToXPosition = void |  |  |
|  |  | Player |  | getToYPosition = int |  |  |
|  |  | name = string |  | setToYPosition = void |  |  |
|  |  | color = string |  | toString = string |  |  |
|  |  | id = int |  | getRating = int |  |  |
|  |  | toString = string |  | setRating = void |  |  |
|  |  | player = int |  |  |  |  |
|  |  | getId = int |  |  |  |  |
|  |  | setId = void |  |  |  |  |
|  |  | getName = string |  |  |  |  |
|  |  | setName = void |  |  |  |  |
|  |  | getColor = string |  |  |  |  |
|  |  | setColor = void |  |  |  |  |

*.*

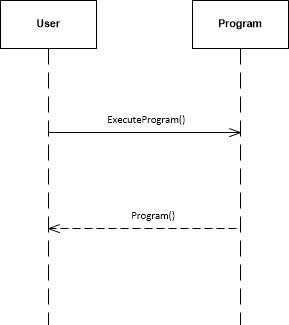
**Chapter 4**

**ANALYSIS MODELS**

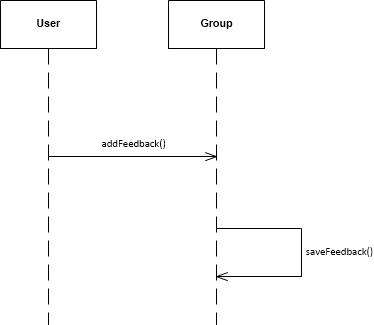
1. **Sequence Diagrams**

## 4.1 Sequence Diagrams

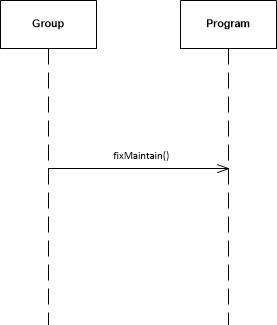
## 4.1.1 Run Program



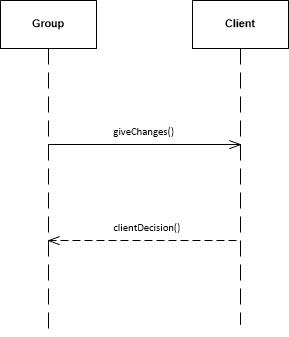
## 4.1.2 User Feedback



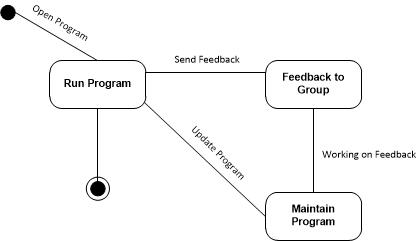
## 4.1.3 Maintain Program



## 4.1.4 Client Decision



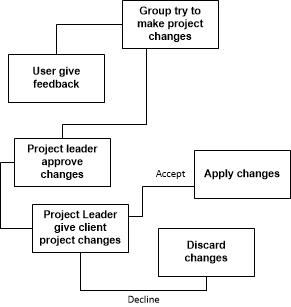
## 4.2 State Diagrams



**Chapter 5**

**CHANGE MANAGEMENT PROCESS**

## 5.1 Process



## 5.2 Actor

The one who responsible to change the SmartCar program are only group member.

## 5.3 Approval

Project changes will be approve by project leader first if the changes are error-free and sufficient, then the project leader will give the changes to the client. If the client approve than the project can change. If the client approve than the changes will be discarded

# Appendices

## Appendix 1

Farrel, Joyce. Java Programmming 6th Edition. Boston : Cengage Learning, 2012.

## Appendix 3

Hurby, P. (2013). Download Visio Stencil and Template for UML 2.5. Retrieved November 13, 2014, from http://www.softwarestencils.com/uml/

1. **Appendices**

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

1. **Appendix 1**
2. **Appendix 2**