MDW Group 7

Shihan Zhiwei ALI

User Requirement Specification

(URS)

# 

# Document Approval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Document Approval 1](#_Toc403592069)

[1. Introduction 2](#_Toc403592070)

[2. Functional Requirement 3](#_Toc403592071)

[3. Use cases 4](#_Toc403592072)

[*1.* *Add guess* 4](#_Toc403592073)

[*2.* *Register* 5](#_Toc403592074)

[*3.* *Start a game* 6](#_Toc403592075)

[4 Non-Functional requirement 6](#_Toc403592076)

# Introduction

This section introduces the requirement specification document for MDW project(group 7). It provides the purpose and scope of the system. Any definitions and references are listed in this section as well as an overview of the remaining requirements specification document.

* 1. **Purpose**

This requirements specification document describes the functions and requirements specified for the multi-player Bulls and cows application (Master mind). It will help our audience (our Tutor and group member) understand what this project is about and its functionality.

* 1. **Definitions, Acronyms, and Abbreviations**

This section will provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the URS.

URS – User Requirement Specification

UI - User interface

MMS – Main success scenario

Ext - extension

W7- Windows 7

WCF - Windows Communication Foundation

# Functional Requirement

**MASTER LIST**

This section contains the listing of all requirements for the web application

1. User can register to be a client to our game
2. User can Login in to the game lobby by typing username and password
3. User can chat with other players in the lobby
4. User can invite other player to play a game
   * A game can be played only for 2 players
5. A player can add guess
6. The other player can enter the number that he want to guess
7. The system should check if the digits that user types are different.
8. The system must compare the value and position of each of the four digits by the user with the position and value of each of the digits of the hidden numbers.

The following MOSCOW table shows the priority level for the implementation of the web application functionalities.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Functional Requirements | Must | Should | Could | Would be better |
| 1 | User can register to be a client to our game | X |  |  |  |
| 2 | User can Login in to the game lobby by typing username and password | X |  |  |  |
| 3 | User can chat with other players in the lobby |  | X |  |  |
| 4 | User can invite other player to play a game |  | X |  |  |
| 5 | A player can add guess | X |  |  |  |
| 6 | The other player can enter the number that he want to guess | X |  |  |  |
| 7 | The system should check if the digits that user types are different. | X |  |  |  |
| 8 | The system must compare the value and position of each of the four digits by the user with the position and value of each of the digits of the hidden numbers. | X |  |  |  |

Table 1. MOSCOW

# Use cases

## Add guess

Goal: To Add a new guess.

Actors: User

Precondition: Program is open.

Post-condition: Desired guess is added

MSS:

1. User types the desirable’s four different digits that compounds a guess.

2. System checks whether the guess is valid.

3. System compares the guess with the hidden numbers and prints on the list box in the same row the guess with the respective feedback:

A checkmark means that the user guess has one digit of the hidden numbers in the right position.

A dot means that the user guess has one digit of the hidden numbers, but in the wrong position.

The above feedback will repeat as needed, and if there is one or more digits from the guess that are not in the hidden numbers then there is no feedback.

4. Use case ends.

Ext:

2.1. If the four digits are not different, system prints a message to the user indicating that different numbers should be typed and user case ends without adding the guess.

## Register

Goal: To create a new user account.

Actors: User

Precondition: Program is open.

Post-condition: a new user account is created.

MSS:

1. User chooses Add Account from the starting form

2. Users types his/her information.

3. System checks if the information is valid.

4. Account is added and use case ends.

Ext:

3.1 If the information is not valid, system shows the message “please, enter all information required”, go to step 2 MSS.

3.2 If the Account Name already exist, system shows the message “The Account xxxx ha’s been created, please, enter another Account Name”, go to step 2 MSS.

## Start a game

Goal: To start game.   
Actors: User

Precondition: Program.

Post-condition: Game is started.

MSS:

1. User logged in.

2. User choose one player and send game invite

3. Invitation accept game start

4. Use case ends.

Ext:

3.1 User chooses button decline, go to step 2 MSS.

# Non-Functional requirement

1. System cannot let user guess character different than digits. System can only let user guess numbers.
2. System cannot provide more or less than four random digits number.
3. System cannot let user determine the hidden number.
4. System cannot work without Internet.
5. System cannot determine the time of game.
6. System cannot let user who have no account to play the game.
7. System cannot record the number of the user’s winning game.
8. System cannot store the message of the guessing and chatting history after user exit the form.