CS251: Software Engineering I

Software Requirements Specification For Bingo

Prepared by
Team Name: A2M2

CS251: Phase 1 – A2M2
Project: Bingo (Multiplayer Distributed Game)

Software Requirements Specifications

Versions History

Date	Version	Description	Author(s)
1/9/2014	1.0	Initial version	A2M2
8/9/2014	1.1	Final version	A2M2

Approved by:

Eng. Mohammed Arnaoty

CS251: Phase 1 – A2M2
Project: Bingo (Multiplayer Distributed Game)

Software Requirements Specifications

Contents

Team

CS251: Phase 1 – A2M2
Project: Bingo (Multiplayer Distributed Game)

Software Requirements Specifications

ID	Name	Email	Mobile
20120363	Mahmoud Mohamed Gamal	mandolove200@gmail.com	01100841259
20120256	Amr Abo El-Aila	aboelala.amr@gmail.com	01117908593
20120352	Mahmoud Hamdy Sayed	ma7moud_7amdy18m@yahoo.com	01114044592
20120260	Amr Sameh Rafat	amr4159@gmail.com	01113288609

Document Purpose and Audience

- **This document is the Software Requirements Specification (SRS) for Bingo Game. It is designed and written for stakeholders: professors and project developers. Its purpose is to describe the functional.**

Introduction

Overview

- http://en.wikipedia.org/wiki/Bingo_(U.S.)

Software Purpose

- **The goal of the project is to implement a board game known as Bingo. The game should be simple and fast. The final product is going to be integrated in the Main GUI application. It will allow players to play against others.**

Software Scope

Software Requirements Specifications

- **It will be for smartphones which work with Android anyone has a mobile like this he can use our application.**
- **It is a multiplayer game and one of the users will be the server host and he can also play with other users.**
- **The scope of the LOGIC component is to simulate a Bingo game between players maintaining the game state and by enforcing the rules of the game.**
- **The GUI component is the mean of interaction between the user and the game itself, and its scope covers the following aspects: visualizing the state of the game, letting the user make decisions and actions during the game and allowing the user to start a new game, or exit from existing game.**
- **As soon as the players enter the game's graphical user interface (GUI), they will be able to see a set of controls, each of them representing different functionalities of the game. The main window will also display: the square board, the player's scores, a button to make the announcements "Ready" and a button for leaving the game.**

Definitions, acronyms, and abbreviations

- **Acronyms:**
  - **SRS:** the Software Requirements Specification is a document that describes the functionalities of the system to be implemented.
- **Abbreviations:**
  - **B:** Bingo
  - **G:** GUI

Glossary

- **Player**: a person that is currently playing an instance of the *Bingo* game.
- **User:** a person that is using the application but has not yet become a player.
- **GUI:** a component of the *Bingo* Game that is responsible for displaying all the relevant information to the player and receiving his/her actions.

Requirements

Software Requirements Specifications

Functional Requirements

- **Create server:**
  - User or player shall create the server first and after connect to the server and other players do,

  He will be able to start the game by press on the button ("start") and then he will be able to fill the board of the game.
- **Join server:**
  - After creating the server the other players will join the existing server and then they has joined and by pressing the button ("start") then he will fill the board of the game.
- **Set the board:**
  - User or player will fill the initial board of the game so as to be able to start the game and every player will have only one minute to fill the board and if the time pass over him and he didn't fill it then the board will be filled randomly by the game.
- **Starting the game:**
  - The game will start after the board was filled completely and all cells are filled by all the users then the game shall start.
- **Get number:**
  - Here when the user pick a number in his board in his turn, this will found the cell in his board that contains the number and then cancel it by changing the color of the cell.
- **Update cell:**
  - This function will have the turn to search for the cell in all boards of the game that contain that number that the last user picked and cancel it by change the color of the cell.
- **Check row:**
  - this function's responsibility is to check in all boards of the users of the game and see if anyone have complete the row ("all the row cells have been colored") .if any of them did then the function should return true .
- **Check column:**
  - This function's responsibility is to check in all boards of the users of the game and see if anyone have complete the column ("all the column cells have been colored"). if any of them did then the function should return true .
- **Check diagonal:**

Software Requirements Specifications

- This function's responsibility is to check in all boards of the users of the game and see if anyone have complete the diagonal line ("all the diagonal line's cells have been colored"). If any of them did then the function should return true.

- **Update board:**
  - This will make update of the entire board by checking (rows, columns, diagonal lines). Then cancel the number of characters that everyone passed ("equal the number of trues in ever board") and see characters that have been crossed or deleted in the word (***BINGO***).

- **Check winner:**
  - Here this function will see at all the players how many characters have been canceled and if anyone has passed 5 characters of the word **BINGO** and he will be declared as the winner.
- **Exchange turns:**
  - After the first player play. this will change the turn to the next player and the sort will be as joined which is ,the play will begin with the server player ("he will take the first turn") then the first host to the server will take the second turn and so on….
- **Show winner:**
  - This function will print on the screen a message for all user to tell the winner ("you won"), and tell the use losers ("you lose").

Non Functional Requirements

- **Bingo is a multiplayer android game with a maximum 4 players.**
- **Usability**
  - ***Simple to play:*** the game rules is very easy and the player should know how to a play Bingo by attached instructions.
- **Reliability**
  - ***Crash safe***: The Bingo application should be crash safe in 90% of its runtime.
- **Performance**
  - The loading time of the Bingo application will be smaller than 20 seconds. All other response times must be within 10 seconds.
- **Supportability**
  - **Score:** the player score contains how many times he wins and losses and in which level.

Software Requirements Specifications

- **Coins**: The player enter the game with ex. 100 coins if he wins he take 10 coins from each player in the game.

- **Implementation**
  - ***Programming language***: Bingo must be implemented in Java and android.
- **Interface**
  - ***Simple user interface***: The user interface of Bingo should be understandable to the player on the first view. The user interface is based on a main window, which includes a button bar to start game, a player instruction and player score.

System Models

Scenarios

- **Starting Bingo**
  - The user starts the Bingo application by clicking on the installed Bingo file. The Bingo window opens with the main menu. The main menu contains the playing button, score button, instruction button and exit button. By clicking the playing button a new window will appear asking the user if he wants to be the server by creating new server or he will join existing server. If he choose to create server he will enter the number of players and waiting them to join him in his server, Else if he choose to join existing server a menu with the names of servers existing in his domain. The user will chose the server and wait the other players to join, once all players joined the server. The game is ready to play.

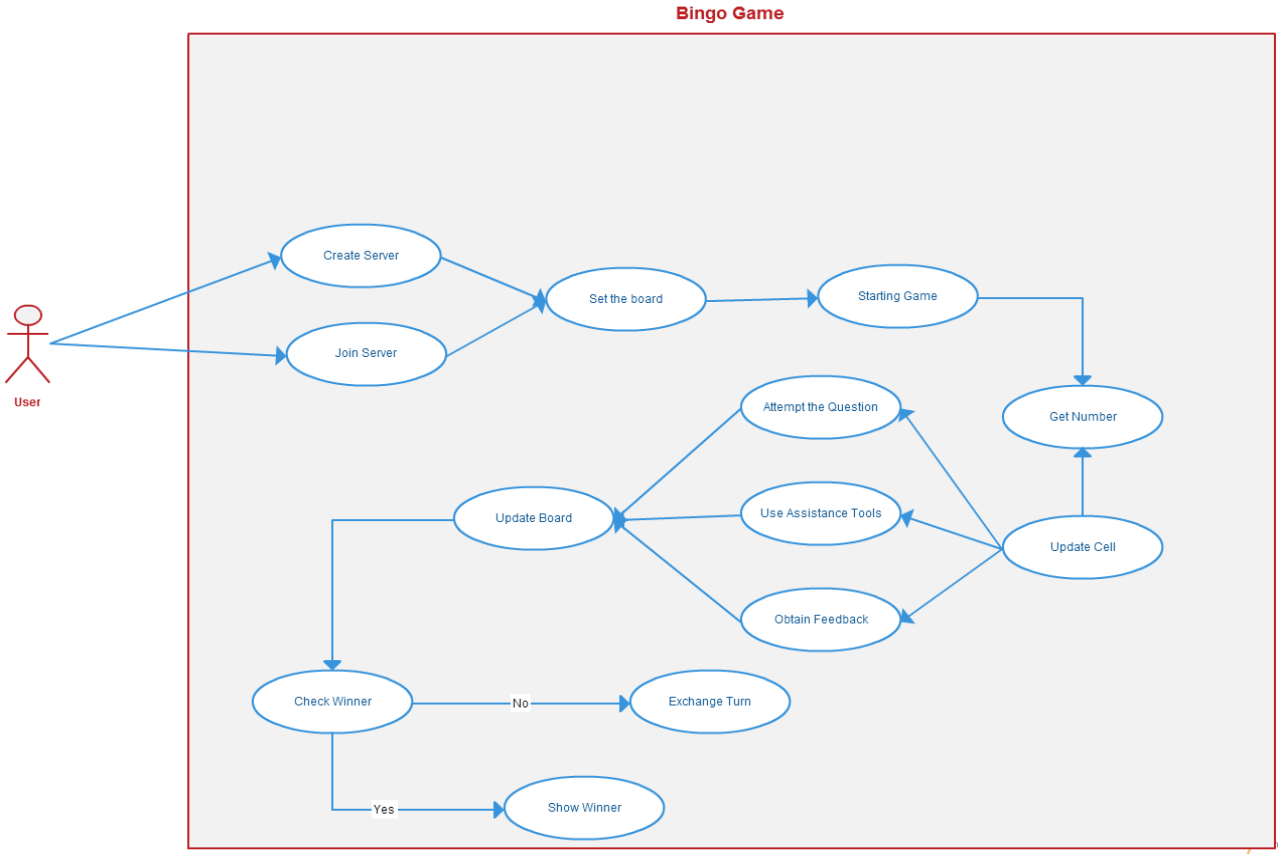
- **Playing a Bingo game**
  - After starting the Bingo application, all players start to distribute the 25 numbers in different cells. The user board will contain a label to enter the cell number he will add the next number to it, located under the board. Each number of the 25 numbers will be added in the cell the user will choose. After allocating the 25 number, the roles will be set in terms of which player entered the server first. The player with the first role will choose number this number will be appended in other players' boards. After ending a row, column or diagonal. A letter of the Bingo word will be shaded by a line

- **Winning a Bingo game**
  - During the playing of a Bingo game, the Bingo word for all players will be checked once one of the players ended all the letters a dialog window with the text 'Congratulation, you win!' appears for the player. The board will be cleared and be ready for new game.

Software Requirements Specifications

- **Losing a Bingo game**
  - During the playing of a Bingo game, if one of the others players ended his letters first the game will stop.
- **Actors**
  - The only actor in the Bingo game will be the players. The players, will distribute the numbers in the empty cells and starting the game and choose numbers till one of the players end the letters of Bingo.

Use Case Model



Software Requirements Specifications

Use Case Tables

Use Case Name :	Create Server	
Actor :	Player	
Pre-conditions :	The player choose to start new game	
Post-conditions :	Other players join the created servers	
Flow of events :	Player Action	System Action
	1. The Player press create server 2. The player will choose the number of the players will join the server and play with him	3. Server will be created with the name of the domain name 4. New window appear showing the players who entered the server

Software Requirements Specifications

Use Case Name :	Join Server	
Actor :	Player	
Pre-conditions :	Other player created a server	
Post-conditions :	Starting the game	
Flow of events :	Player Action	System Action
	1. The Player press join server 3.The player will choose the server name	2.Server will show the player the server he can join 4.once all players joined the server the game will start
Exception :	No servers to join	

.

Use Case Name :	Show Score	
Actor :	Player	
Pre-conditions :	The Player played the game before	
Post-conditions :	The player score will be shown	
Flow of events :	Player Action	System Action
	1. The Player press Score button	2. Server will show the player statistics.

Software Requirements Specifications

Use Case Name :	Show Instruction	
Actor :	Player	
Pre-conditions :	The Player Installed the game	
Post-conditions :	The game instruction will be shown	
Flow of events :	Player Action	System Action
	1. The Player press Instruction button	2. System will show the game instructions.

Use Case Name :	Exit game	
Actor :	Player	
Pre-conditions :	The Player entered the game	
Post-conditions :	The game is ended	
Flow of events :	Player Action	System Action
	1. The Player press Exit button	2. System will exit the game.

Software Requirements Specifications

Ownership Report

Item	Owners
Document Purpose and Audience , Introduction , Glossary , System Models (Use Case Diagram)	*Mahmoud Mohamed Gamal*
Functional Requirements	*Amr Sameh Rafaat*
Non Functional Requirements	*Mahmoyd Hamdy Sayed*
System Models (Scenarios , Use Case Tables)	*Amr Abo El-Aila*