Cairo University Faculty of Computers and Information



**CS352 – Software Engineering II**

**Summer 2015**

**Dr. Mohammad El-Ramly**

Contents

[Instructions [To be removed] 3](#_Toc425629373)

[Team 3](#_Toc425629374)

[Introduction 3](#_Toc425629375)

[Software Purpose 3](#_Toc425629376)

[Software Scope 3](#_Toc425629377)

[Requirements 3](#_Toc425629378)

[Functional Requirements 3](#_Toc425629379)

[System Models 4](#_Toc425629380)

[Use Case Model 4](#_Toc425629381)

[Use Case Tables 4](#_Toc425629382)

[Ownership Report 5](#_Toc425629383)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20120372 | Marwa Mostafa Ahmed | Meromarwa\_1994@yahoo.com | 01121969630 |

# Introduction

## Software Purpose

**Connect 4 is game server for 2 players can play it.**

## Software Scope

Connect 4 can be played by any 2 players but the server can take up to 20 games at the same time.

There are major features in our game:

* Players have accounts and have their data and save their games with their scores.
* Players put pieces on a vertical board till one of them manages to win.
* A player can see available games and request to join one of them.
* The server can manage up to *20* games at the same time.
* If a game is inactive for 30 minutes, the server terminates it.

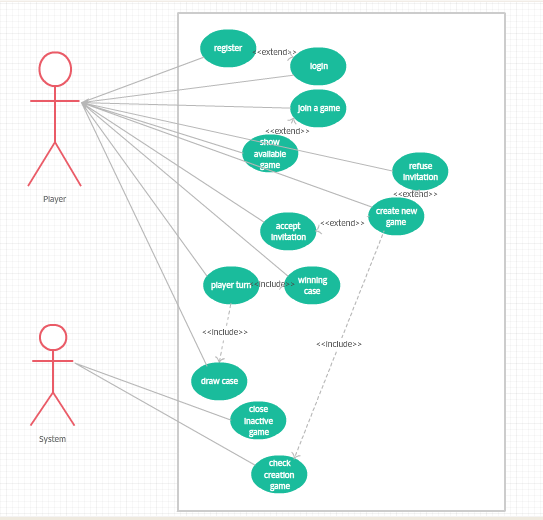
# Requirements

## Functional Requirements

1. Register  
   player will register for his account with his mail, name, password and gender and data will be saved in database for later usage.
2. Login  
   user write his name and password, system checks for the user in database and if data is right then login.
3. Create new game  
   a player can create new game and wait another player to join the game , each new game has a unique ID.
4. Show available games  
   use can see available games that were created to choose to join one.
5. Join a game  
   player chooses one of the available games to join and creator can accept his invitation or refused.
6. Accept invitation  
   game creator accept the invitation of the player to start the game.
7. Refuse invitation  
   game creator refuse the invitation of the player.
8. Player turn   
   player choose the column to put a piece on a vertical board till one of the players manages to have 4 consecutive pieces in a row, column or diagonally.
9. Winning case  
   check after each turn when the player has won by checking if a player managed to have 4 consecutive pieces in a row, column or diagonally.
10. Draw case  
     check after each turn if draw case has happened.
11. Check creation game  
    when user creates a game and the server has more than 20 games then the game will be refused.
12. Close inactive games  
    check if a game hasn’t started foe 30 min then it closed.

# System Models

## Use Case Model

****

## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_1 | |
| Use Case Name: | Register | |
| Actors: | Player | |
| Pre-conditions: |  | |
| Post-conditions: | Login | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter name, Password, email. |  |
|  | 2- System save data in database. |
|  |  |
|  |  |
|  |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_2 | |
| Use Case Name: | Create new game | |
| Actors: | Player | |
| Pre-conditions: | User logged in. | |
| Post-conditions: | Wait for someone to join the game. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User create new game. |  |
|  | 2- System check if there are less than 20 games available then if true it adds game to available games. |
|  |  |
|  |  |
|  |  |
| Exceptions: | System has more than 20 games. | |
| Includes: |  | |
| Notes and Issues: |  | |
|  |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_3 | |
| Use Case Name: | Show available games | |
| Actors: | Player | |
| Pre-conditions: | User logged in. | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- User check available games. |  |
|  | 2- System show available games in database. |
|  |  |
|  |  |
|  |  |
| Exceptions: | There is no available games. | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_4 | |
| Use Case Name: | Join a game | |
| Actors: | Player | |
| Pre-conditions: | User logged in. | |
| Post-conditions: | Wait for game creator to accept invitation. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User choose a game from available games. |  |
|  | 2- System send invitation to the creator. |
|  |  |
|  |  |
|  |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_5 | |
| Use Case Name: | Accept invitation | |
| Actors: | Player | |
| Pre-conditions: | A player wants to play a game that he created. | |
| Post-conditions: | Game starts. | |
| Flow of events: | **User Action** | **System Action** |
| 1- game creator accept invitation. |  |
|  | 2- System starts new game. |
|  |  |
|  |  |
|  |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_6 | |
| Use Case Name: | Refuse invitation | |
| Actors: | Player | |
| Pre-conditions: | A player wants to play a game that he created. | |
| Post-conditions: | the invitation is cancelled. | |
| Flow of events: | **User Action** | **System Action** |
| 1- game creator refuse invitation. |  |
|  | 2- System cancel invitation and game stay available. |
|  |  |
|  |  |
|  |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_7 | |
| Use Case Name: | Player turn | |
| Actors: | Player | |
| Pre-conditions: | User logged in and the game is started. | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- User choose columns to put piece. |  |
|  | 2- System add that piece and check the winning case and draw case |
|  |  |
|  |  |
|  |  |
| Exceptions: | There are no place for the pieces. | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_8 | |
| Use Case Name: | Draw case | |
| Actors: | Player | |
| Pre-conditions: | User logged in and the game is started. | |
| Post-conditions: | The game ended. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1- System check for draw and if true then end game. |
|  |  |
|  |  |
|  |  |
|  |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_9 | |
| Use Case Name: | Winning case | |
| Actors: | Player | |
| Pre-conditions: | User logged in and game is started. | |
| Post-conditions: | The Game ended. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1- System check for draw and if true then end game. |
|  |  |
|  |  |
|  |  |
|  |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | SN\_10 | |
| Use Case Name: | Close inactive game | |
| Actors: | System | |
| Pre-conditions: | The game is available for 30 minutes or more and didn’t start | |
| Post-conditions: | The game is deleted from available games | |
| Flow of events: | **User Action** | **System Action** |
| 1. Game notify the system if it’s open more than 30 minutes |  |
|  | 1. System closes the game. |
|  |  |
|  |  |
|  |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Software purpose, Software scope, Functional requirements, Use case model and use cases table | *Marwa Mostafa* |