Cairo University Faculty of Computers and Information



**CS352 – Software Engineering II**

**Summer 2015**

**Dr. Mohammad El-Ramly**

Contents

[Team 3](#_Toc425629374)

[Introduction 3](#_Toc425629375)

[Software Purpose 3](#_Toc425629376)

[Software Scope 3](#_Toc425629377)

[Requirements 4](#_Toc425629378)

[Functional Requirements 4](#_Toc425629379)

[System Models 5](#_Toc425629380)

[Use Case Model 5](#_Toc425629381)

[Use Case Tables 6](#_Toc425629382)

[Ownership Report 15](#_Toc425629383)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20110462 | Mohammed Ahmad Hassan Ahmad | [Mohammed\_ahmad\_2011@yahoo.com](mailto:Mohammed_ahmad_2011@yahoo.com) | 01117381134 |
| 20110328 | Mohammed Mahmoud Abd-Elfadeel | [Mohamed.abdelfadeel148@gmail.com](mailto:Mohamed.abdelfadeel148@gmail.com) | 01220624748 |
| 20110119 | Amgad Fady Seed |  |  |
|  |  |  |  |

# Introduction

## Software Purpose

* Implement a Connect 4 game server using Google App Engine (GAE) technology and REST services. And implement the client application that is able to communicate with the server.
* Connect 4  is a two-player [connection game](https://en.wikipedia.org/wiki/Connection_game) allow players to play with each other and choose which player Wants to play with.
* Online game.

## Software Scope

* Connect 4 game uses GAE to run on server and build it as a web application.

**\*\* How to play:**

* There are two players each one have color and then take turns dropping colored discs from the top into a seven-column, six-row vertically suspended grid. The pieces fall straight down, occupying the next available space within the column.
* Components:  
  \* Two players.  
  \* Game view (colored discs, seven-column, six-row vertically suspended grid).  
  \* Game connected on the server.

# Requirements

## Functional Requirements

1. **upload application(for developer):**

* Build and run applications on Google’s infrastructure.
* Using App Engine’s Java runtime environment.
* App Engine's java runtime environment is JRE7 (Jdk1.7)

1. **players connected:  
   \*\* login:**

* Players can connect to the server and login in by name.   
  **\*\* new game:**
* Player create new game. A new game is given an ID**.  
  \*\* available game** **and request to join:**
* A player can see available games and request to join one.  
  **\*\* accept game:**
* A game owner (creator) can accept the invitation.

1. **Game view:**

* Colored discs, seven-column and six-row vertically suspended grid.
* Option for each player to choose the color of the discs.
* Scoreboard to show the result between two players and show the time spends in game.
* Button allows players start now game and end the game or match.

1. **Interaction(playing game, win, lose, draw):**

* When an invitation is accepted, the game starts and the players take turns in playing till one them wins or there is a draw.
* Players put pieces on a vertical board till one of them manages to have 4 consecutive pieces in a row, column or diagonally.

1. **check:**

* Check if the players put pieces in available spaces.
* The pieces fall straight down, occupying the next available space within the column.
* Check if have 4 consecutive pieces in a row, column or diagonally.

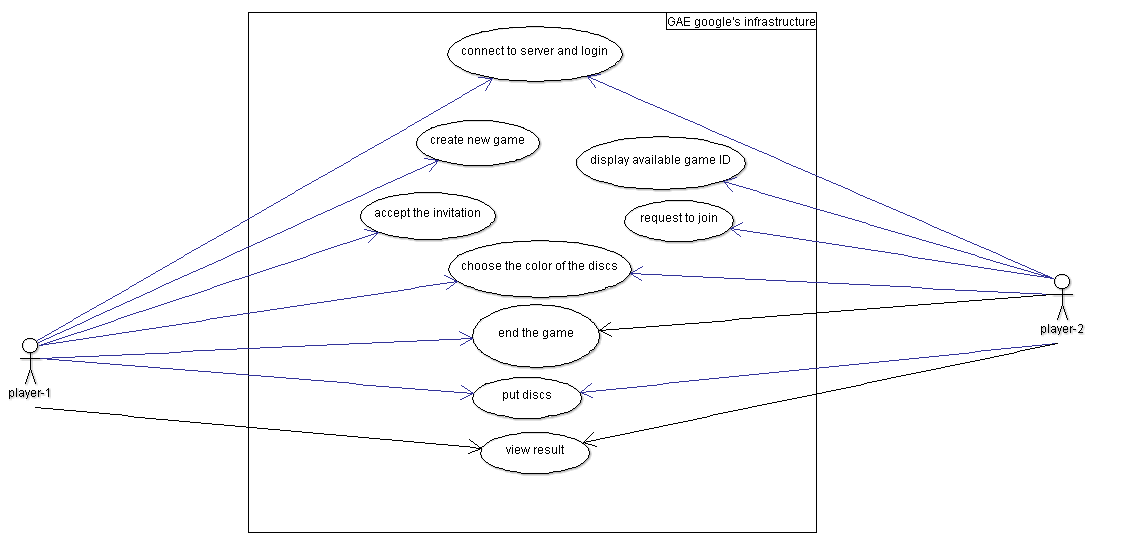
1. **Control the server:**   
   **\*\* limit of games at same time:**

* The server can manage up to x games at the same time (say 20 or 30).  
  **\*\* active game and terminate:**
* If a game is inactive for 30 minutes, the server terminates it.

# 

# System Models

## Use Case Model



## 

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Connected to server and login | |
| Actor: | User | |
| Pre-conditions: | have internet access | |
| Post-conditions: | start login | |
| Flow of events: | **User Action** | **System Action** |
| 1- User load the server. |  |
|  | 2- System opens login page. |
| 3- User enters his name. |  |
|  | 4- System loads page with create a game and display all available games buttons. |
|  |  |
| Exceptions: | Server cannot load because there is more than 30 game at the same time. | |
| Includes: | User can login with name only. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | Crate new game | |
| Actor: | User (player1) | |
| Pre-conditions: | Player1 logins and creates a new game. | |
| Post-conditions: | A new game created and given an ID and be available. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User creates new game |  |
|  | 2- System creates new game and gives it an ID. |
|  | 3- System makes this game available to other users. |
| Exceptions: | Server cannot load because there is more than 30 game at the same time. | |
| Includes: |  | |
| Notes and Issues: | Player1 is a user who creates the new game. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Display available game ID | |
| Actor: | User (player-2) | |
| Pre-conditions: | Players-1 create games and system given it an ID | |
| Post-conditions: | Player-2 displays available games and chooses one. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Player1 clicks on display all available games button. |  |
|  | 2- System displays all available games ID. |
| Exceptions: | There is no available game. | |
| Includes: |  | |
| Notes and Issues: | Players1 is users who create the new game and players2 who need to display available game. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Request to join | |
| Actor: | User (player-2) | |
| Pre-conditions: | Player2 displays available games | |
| Post-conditions: | The system send invitation to player1 | |
| Flow of events: | **User Action** | **System Action** |
| 1- User request to join an available game. |  |
|  | 2- System sends invitation to player1. |
| 3- Player2 waits for player1 acceptance. |  |
| Exceptions: | System didn’t send invitation. | |
| Includes: |  | |
| Notes and Issues: | Player2 who requests to join a game.  Player1 who accepts player2 request. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 5 | |
| Use Case Name: | accept the player2 invitation | |
| Actor: | User (player1) | |
| Pre-conditions: | System sends the invitation to player1 | |
| Post-conditions: | System starts the game | |
| Flow of events: | **User Action** | **System Action** |
|  | 1. System sends invitation to player1. |
| 1. Player1 get invitation from player2. |  |
| 1. Player1 accepts invitation. |  |
|  | 1. System deletes game with ID from available games. |
|  | 1. System starts the game. |
| Exceptions: | System didn’t send invitation.  Player1 didn’t accept the invitation. | |
| Includes: | Player1 accepts the invitation | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 6 | |
| Use Case Name: | Choose disc color | |
| Actor: | Player1,2 | |
| Pre-conditions: | System starts the game and display available colors. | |
| Post-conditions: | System view game board to allow players to put discs | |
| Flow of events: | **User Action** | **System Action** |
|  | 1. System starts the game and display available colors. |
| 1. Player1,2 choose their color of discs. |  |
|  | 1. System view game board to allow players put discs. |
| Exceptions: |  | |
| Includes: | Player1 chooses his color first. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 7 | |
| Use Case Name: | Put discs | |
| Actor: | Player1, 2 | |
| Pre-conditions: | System views game board to allow players to put pieces | |
| Post-conditions: | System ends the game and computes results | |
| Flow of events: | **User Action** | **System Action** |
|  | 1. System view game board. |
| 1. Player1, 2 put discs and take turns. |  |
|  | 1. System ends the game and compute results. |
| Exceptions: | Player 1 or 2 didn’t put discs. | |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 8 | |
| Use Case Name: | View result | |
| Actor: | System, players | |
| Pre-conditions: | System ends the game. | |
| Post-conditions: | System view result and the winner. | |
| Flow of events: | **User Action** | **System Action** |
| 1. Players take turns. |  |
|  | 1. System ends the game |
|  | 1. System computes result and displays it. |
|  | 1. Players can click view result button to display result again. |  |
|  |  | 1. System computes result and displays it. |
| Exceptions: |  | |
| Includes: | Result includes the winner. | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 9 | |
| Use Case Name: | Player1 or 2 end the game. | |
| Actor: | Player1, 2 | |
| Pre-conditions: | System display result. | |
| Post-conditions: | System loads page with create a game and display all available games buttons. | |
| Flow of events: | **User Action** | **System Action** |
|  | 1. System display game result. |
| 1. Player 1 or 2 ends the game. |  |
|  | 1. System deletes game ID. |
|  | 1. System loads page with create a game and display all available games buttons. |
| Exceptions: | Player 1 or 2 didn’t end the game. | |
| Includes: | Player 1 and 2 ended the game. | |
| Notes and Issues: |  | |

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| * Software purpose * Software scope * Functional requirements * Use case tables 1,2 | *MOHAMED AHMAD HASSAN* |
| * Use case model * Use case tables 7,8,9 | *MOHAMED MAHMOUD ABDELFADEEL* |
| * Functional requirements * Use case tables 3,4,5,6 | AMGAD FADY |