

**CS251 – M-Chess Game**

**Project**

**Software Requirements Specifications**

|  |  |
| --- | --- |
| **Staff Memebers** | **Email** |
| **Dr Mohamed El Ramly** | **m.elramly@fci-cu.edu.eg** |
| Eng Catherine Bedrossian | c.bedrossian@fci-cu.edu.eg |
| Eng Sara Tarek Ali Othman | s.tarek@fci-cu.edu.eg |
| Eng Mohamed Samir | m.samir@fci-cu.edu.eg |
| Eng Marwa Nabil | m.nabil@fci-cu.edu.eg |
| Eng Yomna Magdy Mohamed | yomna@fci-cu.edu.eg |
| Eng Desoky Abd El-qawy | d.abdelqawy@fci-cu.edu.eg |
| Eng Omar Khaled Ali Ragab | o.khaled@fci-cu.edu.eg |
| Eng Mohamed Elarnaoty | m.elarnaoty@fci-cu.edu.eg |
| Eng Mostafa Saad | m.saad@fci-cu.edu.eg |

Contents

[Team](#_Toc402452669) 2

[Document Purpose and Audience](#_Toc402452670) 2

[Introduction 3](#_Toc402452671)

[Software Purpose 3](#_Toc402452672)

[Software Scope](#_Toc402452673) 3

[Definitions, acronyms, and abbreviations](#_Toc402452673) 3

[Requirements 4](#_Toc402452675)

[Functional Requirements 4](#_Toc402452676)

Non-[Functional Requirements 4](#_Toc402452676)

[System Models 5](#_Toc402452675)

Use case Model [5](#_Toc402452676)

Use case tables [6](#_Toc402452676)

[Graphical user interface 10](#_Toc402452675)

[Ownership Report 10](#_Toc402452675)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20120160 | Khaled Abd-ElRahman Yehia | Assassins\_doom@yahoo.com | 01099366891 |
| 20120069 | Ahmed Hzzaa’ Al-sai’d | Hzzaaahmed@gamil.com | 01111992307 |
| 20120335 | Mohamed Maher Ibrahem | Maherm221@yahoo.com | 01220463743 |
| 20120261 | Amr Adel Mohammed Khalefa | amradel525@gmail.com | 01098997611 |

# 

# Document Purpose and Audience

* Document purpose is to decalcify software requirements of multiplayer chess game.
* Audiences of this document are: Clients, users, Software manager, software developers, Graphics designers and Software analysts.

# Introduction

## Software Purpose

* Software purpose is to produce a multiplayer chess game named “M-Chess” that players can play through a standalone devices or machines

## Software Scope

* This game is a multiplayer chess game will contain a simple 2-D graphical user interface containing the normal chess game component with normal chess match (no special game modes).

## Definitions, acronyms, and abbreviations

* M-Chess: The project name refers to "Multiplayer chess".
* Chess: It’s a simple turn based mine-game simulate a real medieval battlefield with its main elements like (Soldiers, Knights, Castle towers …).  
  Reference: <http://en.wikipedia.org/wiki/Chess>

# Requirements

## Functional Requirements

1. Before any other functional requirements, all players must have the whole files of the game and its updates to be able to join the server.
2. The much must begin between 2 players, no more no less.
3. The game must be created by the host, Game server is responsible of initiating a new game and 2 players must be log in the game (the host and other player) to start the match by: the player sends request to the server then the server creates the game then the server tell the player to invite another player, the invited player has to open the game or the invitation operation won’t be succeeded.
4. Player controls pieces by using mouse, he has to first choose the piece he want to move then the targeted place he wants to move his chosen piece to, if the targeted place is suitable for the chosen piece normal move, the server allows the move, updates the interface of player and sends the update to the other player to update his interface too.
5. If one of players pause the game the server pauses the other player interface too so the other player can’t move any piece until the player who pauses the game resumes again.
6. The player can’t play two or more moves in his turn and he has to wait his turn.
7. There will be a chat box in the interface of the game so players can communicate while playing.
8. History of moves function: its job is to keep record of every player moves during the game, stores it and display it in the graphical user interface.
9. Match time function: contains a counter of seconds witch calculate the time elapsed in the match and display it in the graphical user interface.
10. Time to move function: contains a countdown counter displayed on the screen which tell the user how much time left to make his move.

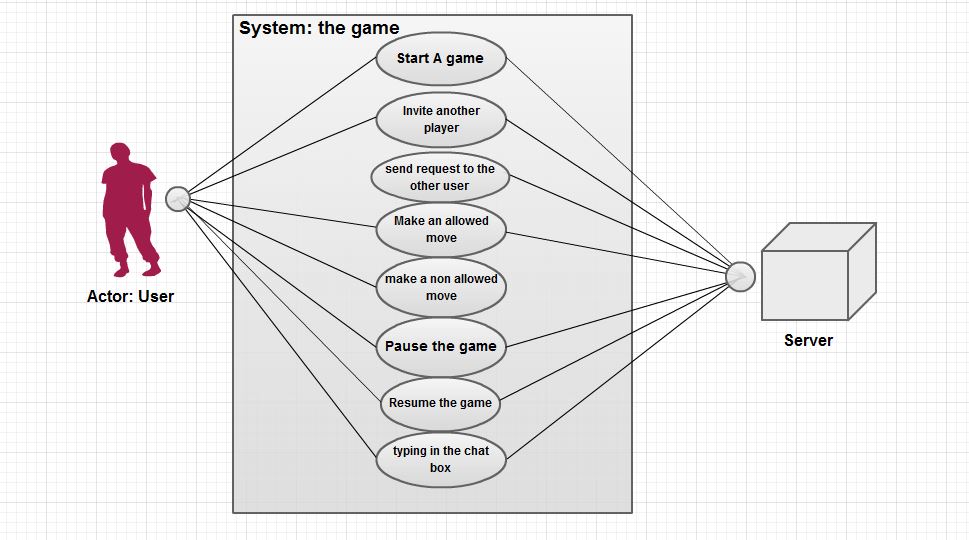
## Non Functional Requirements

* The game has to be simple with 2D Graphical interface so any user can play it (no need for High-End machines).
* The response of the game must be fast and smooth (the board of chess must be updated after a new move in 1 second or less).
* There is a 1 minute for every turn so the player has to take a move through this time or he will lose the match.
* The game must have a main menu that contains  
  "New Game": to create a new match.  
  "Records": shows the history of player (wins, loses, fastest win and its time).  
  "Exit": to exit game.
* The chat texting must be simple, fast and support Arabic and English languages.

# 

# System Models

## Use Case Model



## 

## Use Case Tables

Requirement No. 1:

|  |  |  |
| --- | --- | --- |
| Actors: | User | |
| Pre-conditions: | Launching game | |
| Post-conditions: | System checking game files and its version. | |
| Flow of events: | **User Action** | **System Action** |
| 1- User removed some files from the game or his game has some damaged files |  |
|  | 2- System sends message to user: some game files are not founded or corrupted, try to fix the game or re install it |
| 3- User fixed the game but his version is not up to date |  |
|  | 4- System sends message to user: your game is not up to date, Do you want to download and install the latest update now (yes or no)? |
| 5- User choose (yes) |  |
|  | 6- The system download the new update from the server, apply it then start the game. |
| Exceptions: | -Event 5: If user chooses (No) the game must close. | |
| Includes: |  | |
| Notes and Issues: |  | |

Requirement No.3

|  |  |  |
| --- | --- | --- |
| Actors: | Host user, other User | |
| Pre-conditions: | Choosing "New Game" from the menu | |
| Post-conditions: | Sending a game request to the server. | |
| Flow of events: | **User Action** | **System Action** |
| 1- Host user choose new game |  |
|  | 2- System sends request to the server to initialize a new game |
|  | 3- The server asks the Host user to invite another player by sending his IP address to it. |
|  |  |
| 4- The user types and sends the other User IP to the server. |  |
|  | 5- the server checks the other user IP and checks if the owner of the IP has the game and the game is opened or not |
|  | 6- the system sends an invitation to the other user with a message: "Host user name" invites you to play a match with him (accept, ignore) |
| 7- The other user accept the invitation |  |
|  | 8- The System starts the match. |
| Exceptions: | -Event 5: if the system didn't find the IP owner or the owner of the IP didn't open the game, the operation of invitation will fail with a message to Host user: "the IP is not right or the other player doesn't have the game"  -Event 7: if the other player ignored the invitation, the system sends a message to the Host player: "the other player ignored your invitation" | |
| Includes: |  | |
| Notes and Issues: |  | |

Requirement No. 4

|  |  |  |
| --- | --- | --- |
| Actors: | User | |
| Pre-conditions: | The game has been started | |
| Post-conditions: | The system processing players moves | |
| Flow of events: | **User Action** | **System Action** |
| 1- User choose a soldier as a unit he want to move it (source) |  |
| 2- user choose the place he wants to send the chosen soldier to (destination) |  |
|  | 3-System takes the info of the source and destination. |
|  | 4- System analyses the info and see if the destination is suitable for the source characteristic or not. |
|  | 5-if it suitable, system apply the move and update the board in front of every player with the new situation. |
| Exceptions: | -Event 4: if the move is not suitable for the character in the source position, the system sends message to the user: "invalid move" -Event 4: if the destination position has a one of enemy pieces (except enemy's king), it dies and the system has to kick it out of the board then the user piece will stand in destination position instead of the dead piece | |
| Includes: |  | |
| Notes and Issues: |  | |

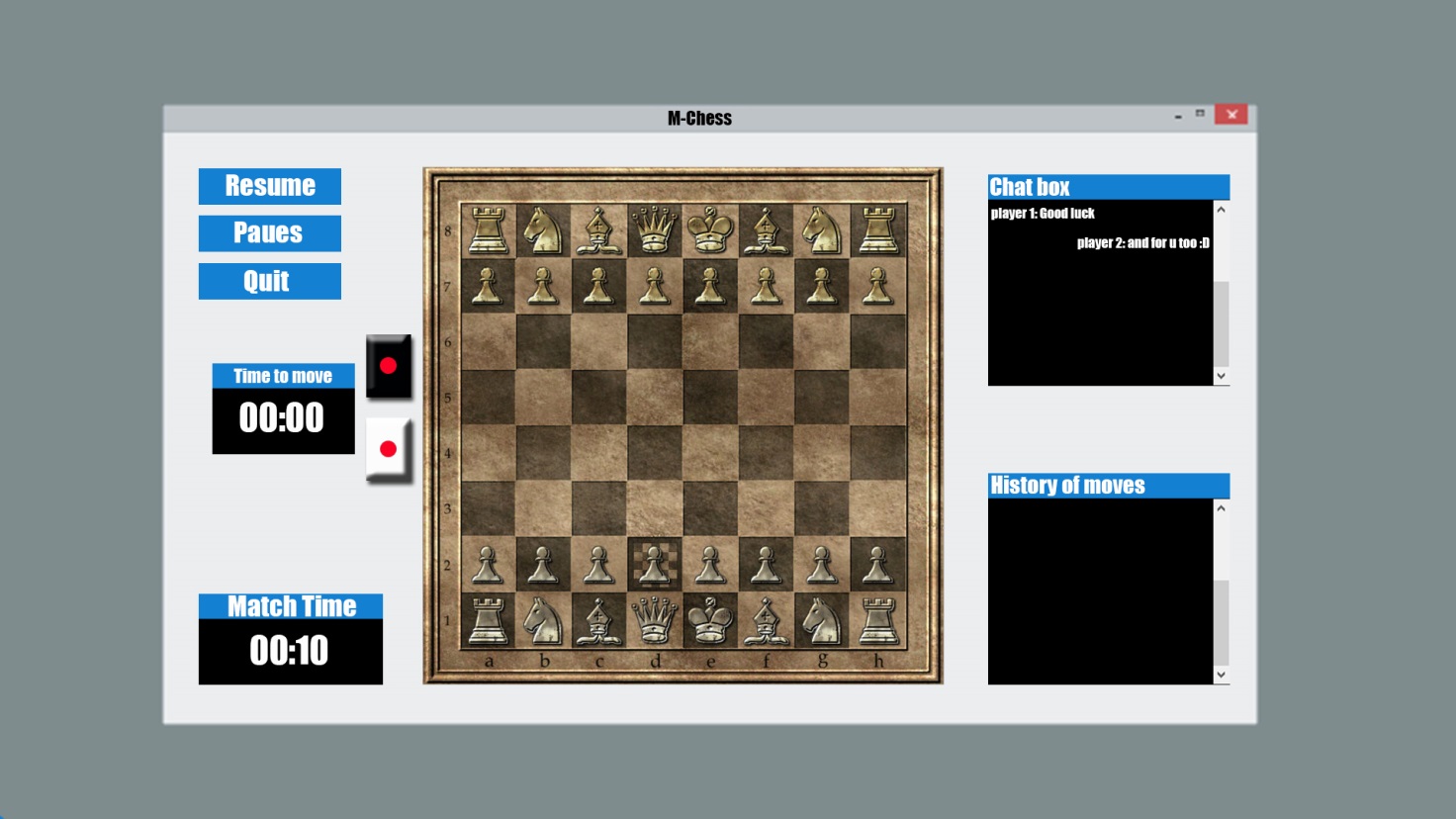
Requirement No.5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Actor | Pre Action | Post Action | output |
| #1 | User | User pressed on "pause" while playing | The system send Pausing request to the server for applying it with the two players | The game pauses and the board will freeze until the user resume again |

Requirement No.6:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Actor | Pre Action | Post Action | output |
| #1 | User | Player spend his turn by a move and try to make another move | The system see if this move is allowed to the user or not | System sends message to user: "invalid move" |

# Graphical user interface



# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Khaled Abd-ElRahman Yehia | Functional Requirements , Use Case Tables , Graphical User Interface |
| Ahmed Hzzaa’ Al-sai’d | Non Functional Requirements , Use Case model |
| Mohamed Maher Ibrahem | Document Purpose and Audience , Definitions, acronyms, and abbreviations |
| Amr Adel Mohammed Khalefa | Software Purpose , Software Scope |