**Software Requirements Specification (SRS) for Othello Project**

*Baseline version 0.1*

*Issued on: November 19, 2014*

Issued by: Game, Inc.

Issued for: Game Company

# Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | November 9, 2014 | Faradila Assagaf | First SRS |
| 1.0 | November 10, 2014 | Faradila Assagaf | Final version of SRS |
| 1.0 | November 17, 2014 | Faradila Assagaf | Final version of SRS revised |
| 2.0 | November 20, 2014 | Faradila Assagaf | Result |

# Document Approval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date** | **Signature** |
| Faradila | Project Manager |  |  |
| Joan Felix | Instructor |  |  |

# Table of Contents

[Change History 1](#_Toc404289848)

[Document Approval 1](#_Toc404289849)

[Table of Contents 2](#_Toc404289850)

[List of Figures 4](#_Toc404289851)

[List of Tables 5](#_Toc404289852)

[INRODUCTION 6](#_Toc404289853)

[1.1. Purpose 6](#_Toc404289854)

[1.2. Scope 6](#_Toc404289855)

[1.3. Definitions, Acronyms, and Abbreviations 6](#_Toc404289856)

[1.4. References 7](#_Toc404289857)

[1.5. Overview 7](#_Toc404289858)

[GENERAL DESCRIPTION 8](#_Toc404289859)

[2.1. Product Perspective 8](#_Toc404289860)

[2.2. Product Functions 8](#_Toc404289861)

[2.3. User Characteristics 8](#_Toc404289862)

[2.4. General Constraints 9](#_Toc404289863)

[2.5. Assumptions and Dependencies 9](#_Toc404289864)

[SPECIFIC REQUIREMENTS 10](#_Toc404289865)

[3.1. External Interface Requirements 10](#_Toc404289866)

[3.2. Functional Requirements 10](#_Toc404289867)

[3.3. Use Cases 10](#_Toc404289868)

[3.3.1. Use Case Login 10](#_Toc404289869)

[3.3.2. Use Case Play vs AI 10](#_Toc404289870)

[3.3.3. Use Case Play Game 11](#_Toc404289871)

[3.3.4. Use Case View Ranking 11](#_Toc404289872)

[3.3.5. Use Case Manage Game 11](#_Toc404289873)

[3.3.6. Use Case Update Score 12](#_Toc404289874)

[3.3.7. Use Case Diagram 12](#_Toc404289875)

[3.4. Classes/Objects 13](#_Toc404289876)

[3.5. Non-Functional Requirements 13](#_Toc404289877)

[3.6. Design Constraints 13](#_Toc404289878)

[ANALYSIS MODELS 14](#_Toc404289879)

[4.1. Sequence Diagrams 14](#_Toc404289880)

[4.1.1. Sequence Diagram Login 14](#_Toc404289881)

[4.1.2. Sequence Diagram Play vs AI 14](#_Toc404289882)

[4.1.3. Sequence Diagram Play Game 15](#_Toc404289883)

[4.1.4. Sequence Diagram View Ranking 15](#_Toc404289884)

[4.1.5. Sequence Diagram Manage Game 16](#_Toc404289885)

[4.1.6. Sequence Diagram Update Score 16](#_Toc404289886)

[4.2. State-Transition Diagrams (STD) 17](#_Toc404289887)

[4.2.1. State Transition Diagram Login 17](#_Toc404289888)

[4.2.2. State Transition Diagram Play vs AI 17](#_Toc404289889)

[4.2.3. State Transition Diagram Play Game 18](#_Toc404289890)

[4.2.4. State Transition Diagram View Ranking 18](#_Toc404289891)

[4.2.5. State Transition Diagram Manage Game 19](#_Toc404289892)

[4.2.6. State Transition Diagram Update Score 19](#_Toc404289893)

[CHANGE MANAGEMENT PROCESS 20](#_Toc404289894)

[A. Appendices 21](#_Toc404289895)

[A.1. Appendix 1 21](#_Toc404289896)

# List of Figures

Figure 1…………………………………………………………………………………………..10

Figure 2…………………………………………………………………………………………..10

Figure 3…………………………………………………………………………………………..11

Figure 4…………………………………………………………………………………………..11

Figure 5…………………………………………………………………………………………..11

Figure 6…………………………………………………………………………………………..12

Figure 7…………………………………………………………………………………………..12

Figure 8…………………………………………………………………………………………..13

Figure 9…………………………………………………………….……………………………14

Figure 10…………………………………………………………….…………………………..14

Figure 11…………………………………………………………….…………………………..15

Figure 12…………………………………………………………….…………………………..15

Figure 13…………………………………………………………….…………………………..16

Figure 14…………………………………………………………….…………………………..16

Figure 15…………………………………………………………….…………………………..17

Figure 16…………………………………………………………….…………………………..17

Figure 17…………………………………………………………….…………………………..18

Figure 18…………………………………………………………….…………………………..18

Figure 19…………………………………………………………….…………………………..19

Figure 20…………………………………………………………….…………………………..19

# List of Tables

Table 1………………………………………………………………...…………………………..7

Table 2………………………………………………………………….………………………..20

Table 3………………………………………………………………….………………………..21

Chapter 1

# INRODUCTION

## Purpose

The purpose of this document is to give a detailed description of the “Othello Project”. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## Scope

“Othello Project” is a strategy game played by two players: Black and White. It is played on an 8x8 board (usually Green in color), called the Othello board. The two players place 64 discs, each of which is black on one side and white on the other. For example, each player begins with 32 discs but these do not belong to him and if his opponent runs out of discs, he is obliged to give him some. A disc is black if the black side is visible and white if the white face is on top.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| Player | A person taking part in Othello game |
| Othello board | Place to put discs |
| Admin/Administrator | System administrator who is given specific permission for managing and  controlling the system |
| Artificial Intelligence (AI) | Computer player |
| Stable discs | A condition where it is impossible to flip a disc placed in a corner since it can never be flanked between two opponent’s discs |
| Mobility | A condition when a player should avoid playing to squares adjacent to a corner |
| Frontiers | Each move is played to an empty square adjacent to an opponent’s discs |
| Parity | If neither player passes his turn during the game, there will be an even number of empty squares whenever black moves, and an odd number of empty squares whenever white moves |
| Minimax | A technique that yields discovery of the optimal solution to a two player game of perfect information, given enough time and space |

Table 1

## References

[1] Lazard E. Othello: the rules of the game. France. 1993. Available URL from:

http://radagast.se/othello/Help/strategy.html

## Overview

The remainder of this document includes four chapters and appendix. Next chapter will discuss about general description including product perspective, product function, user characteristics, general constraints and assumption. In chapter 3, there will be specific requirements with external interface requirement, functional and non-functional requirements, followed by use cases and the diagrams. Chapter 4 discuss about analysis models with sequence and state transition diagrams. Final chapter will be about change management process.

Chapter 2

# GENERAL DESCRIPTION

## 2.1. Product Perspective

This system will consist of two parts: one with AI (artificial Intelligence) and one without AI. In Othello with AI, All AI’s try to maximize their own positions using the minimax way of looking ahead and always try to minimize the maximal damage the opponent can make. It looks ahead for all possible moves a given number of turns ahead (this is determined by the “difficulty” setting). There is also incorporated a concept called “fuzziness” that lets the AI glance to the side for the second most likely outcome and the third etc. The fuzziness factor determines how much these possible outcomes will matter for the AI. These two factors are set for all different Ais. It’s also possible to determine the start position of the board. Othello without AI can be played by two players in one device**.**

## 2.2. Product Functions

The basic goal of an AI player is to consider the possible moves from the current game state, evaluate the position resulting from each move, and choose the one that appears best. One major component of an AI player is the static evaluation function, which heuristically estimates the value of a position without exploring moves. This value indicates which player has the advantage and how large that advantage is. A second major component is the search algorithm, which more accurately evaluates a state by looking ahead at potential moves. This value indicates which player has the advantage and how large that advantage is. A second major component is the search algorithm, which more accurately evaluates a state by looking ahead at potential moves.

## 2.3. User Characteristics

There are three types of users that interact with the system: Player, AI and Administrators. Each of these three types of users has different use of the system so each of them has their own requirements. During each turn, player can only make legal moves or pass. If a player places a disc in a square (call it square X), and there is another square (call it square Y) on a row, column, or diagonal going through X, and all squares on the line between X and Y are filled with discs of the opponent’s color, then the player is committing a legal move. After making a legal move, all discs between squares Xand Y flip their color, for example the player making the legal move acquires the discs of the opponent. It is important to mention that a legal move can constitute disc flipping on more than one row, column, or diagonal. AI player is to consider the possible moves from the current game state, evaluate the position resulting from each move, and choose the one that appears best, while administrator control all the system.

## 2.4. General Constraints

Both AI and player are constrained by the system where the users can’t be arbitrarily leading the game. Besides, users also not able to determines the path of the game.

## 2.5. Assumptions and Dependencies

One assumption about the product is that it will always be used on computer or device that has enough performance.

Chapter 3

# SPECIFIC REQUIREMENTS

## 3.1. External Interface Requirements

It provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

## 3.2. Functional Requirements

Functional requirements including user login, player can choose to play against Artificial Intelligence (AI), players can view their ranking, admin can manage the game and updating score, also manage the user interface.

## 3.3. Use Cases

### 3.3.1. Use Case Login

|  |  |
| --- | --- |
| Use Case Name | Login |
| Use Case ID | OP 001 |
| Actor | Player |
| Description | Player has to login in order to start the game. If the player failed to login, so he can’t start the game, but if he succeed, then he can start to play. |

Figure 1

### 3.3.2. Use Case Play vs AI

|  |  |
| --- | --- |
| Use Case Name | Play vs AI |
| Use Case ID | OP 002 |
| Actor | Player |
| Description | Player chooses to play against AI. This is the condition where the player and AI will play the game continuously until one of them wins the game. |

Figure 2

### 3.3.3. Use Case Play Game

|  |  |
| --- | --- |
| Use Case Name | Play Game |
| Use Case ID | OP 003 |
| Actor | Player |
| Description | Player starts the game after login. |

Figure 3

### 3.3.4. Use Case View Ranking

|  |  |
| --- | --- |
| Use Case Name | View ranking |
| Use Case ID | OP 004 |
| Actor | Player |
| Description | Player sees the rank while and after playing the game. During the game, player can see his score in order to know how many scores left to win the game. Player also can see the score of AI or the other player. |

Figure 4

### 3.3.5. Use Case Manage Game

|  |  |
| --- | --- |
| Use Case Name | Manage Game |
| Use Case ID | OP 005 |
| Actor | Admin |
| Description | Admin can manage all the systems of the game. Admin has the authority to manage the rules and strategies of the game. Besides, admin also has authority in order to build the discs, the board, user interface etc. |

Figure 5

### 3.3.6. Use Case Update Score

|  |  |
| --- | --- |
| Use Case Name | Update Score |
| Use Case ID | OP 006 |
| Actor | Admin |
| Description | Admin can manage the score of the game. When the player starts the game, the score will be zero. After that when the player or AI in the next move, the score will automatically change, either increased or decreased (based on how many discs each player has). The scores will affect the chance to win for both players. |

Figure 6

### 3.3.7. Use Case Diagram

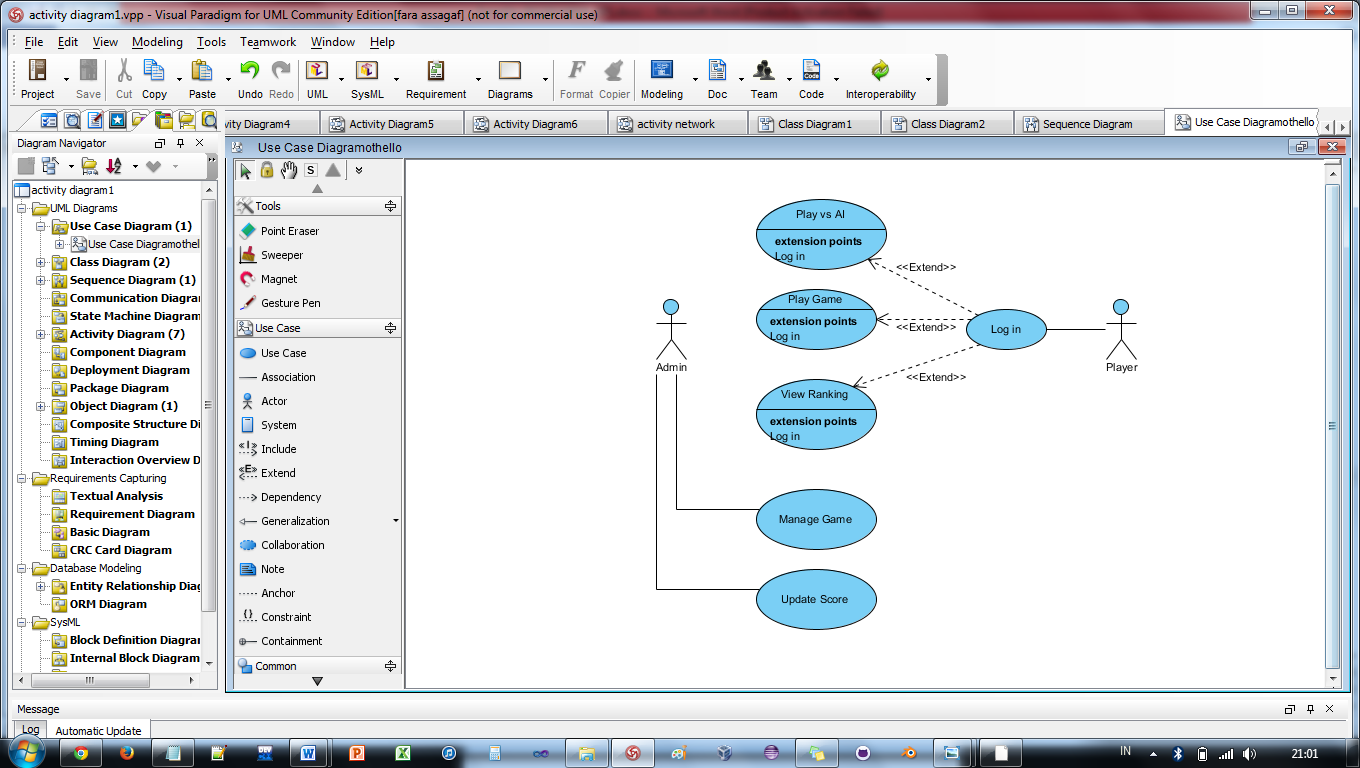
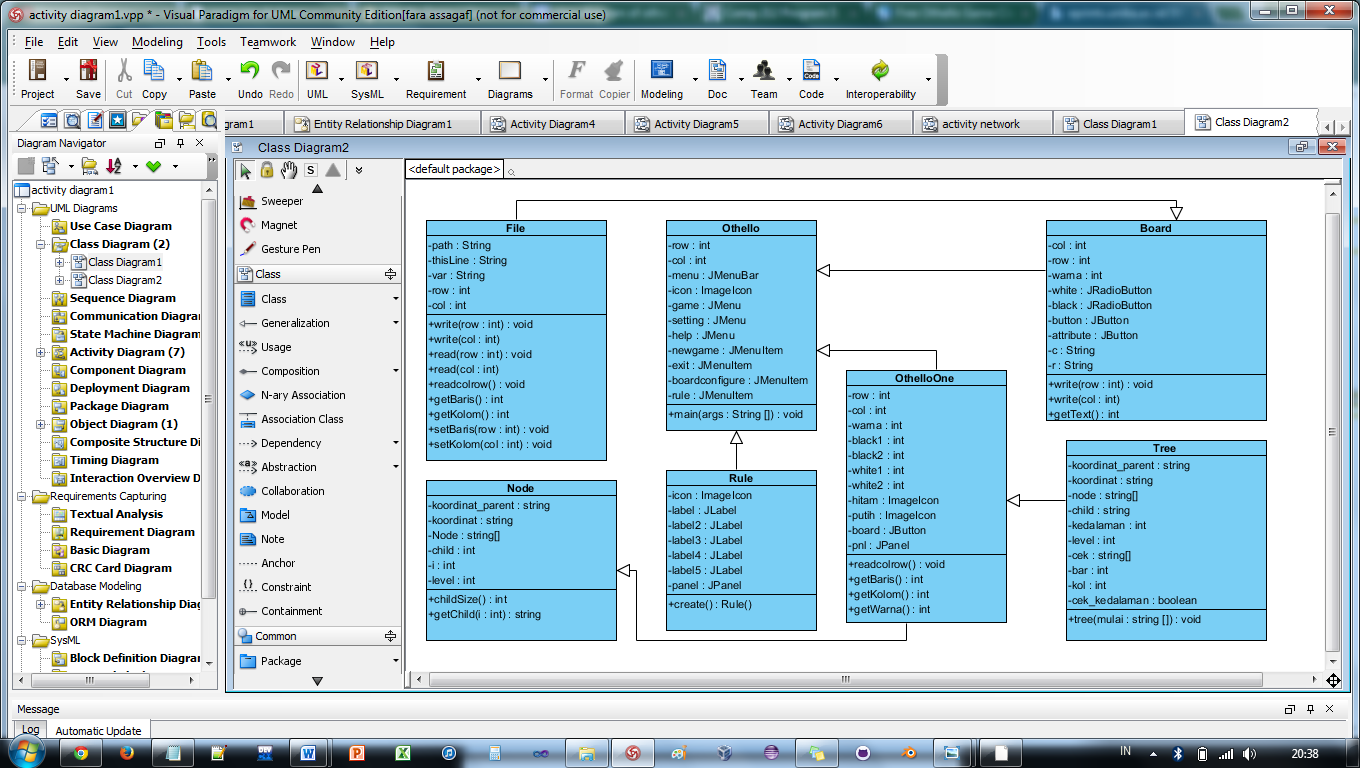


Figure 7

## 3.4. Classes/Objects

Figure 8

This system has seven classes. Each class have specific job. The main class of this system is Othello. File class is used for save number of last board which played. Rule class to show rule of the Othello game. Board class is used for board configuration. Tree class is used for create Tree. OthelloOne class is used for press the Othello piece and how to decide other player's next move. Node class is used for wight to the board in column and row.

## 3.5. Non-Functional Requirements

Non-Functional requirements include Usability such as the speed of use. There is also help option that can be used when player confuse to start the game. The other one is Reliability with error handling.

## 3.6. Design Constraints

This project also has Implementation and design constraints such as programming language (system shall be written in Java) and Operating system (system shall be operate in all windows version).

Chapter 4

# ANALYSIS MODELS

## 4.1. Sequence Diagrams

### 4.1.1. Sequence Diagram Login

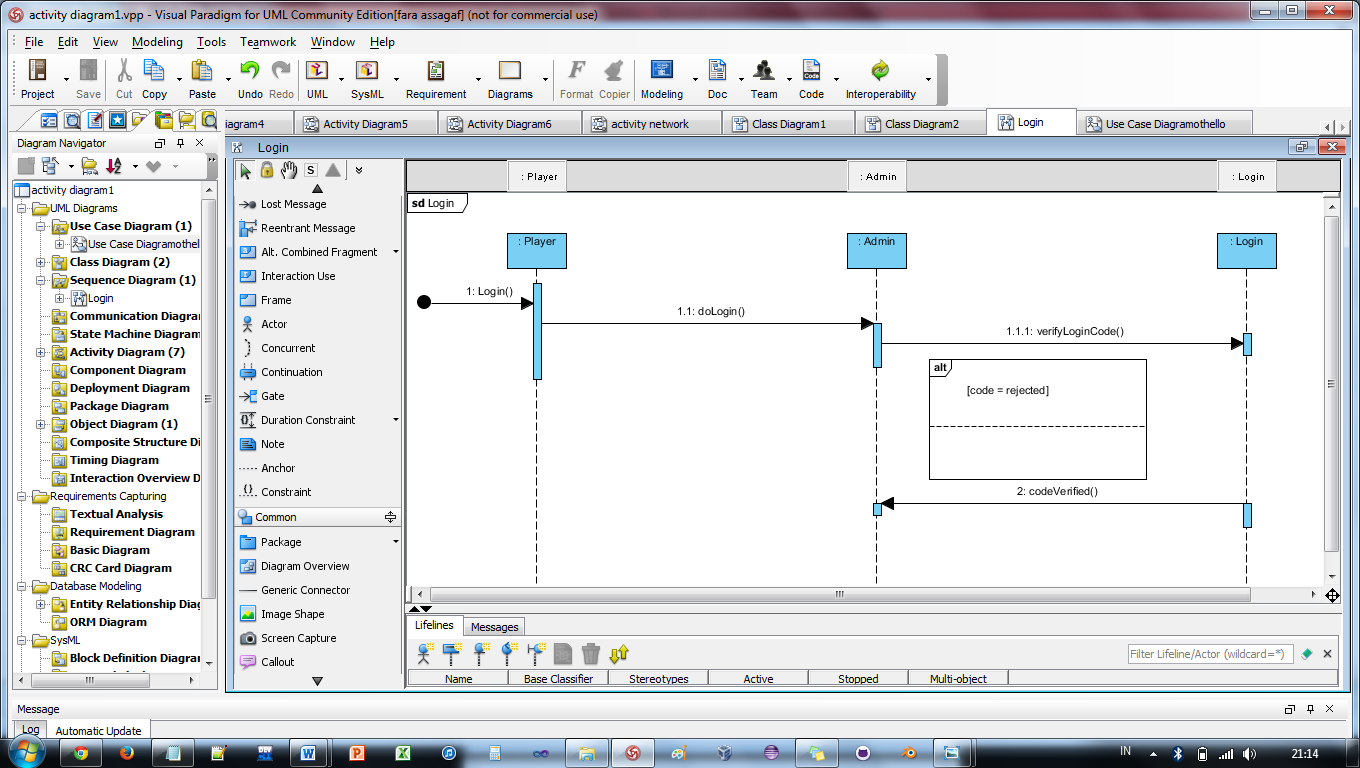


Figure 9

### 4.1.2. Sequence Diagram Play vs AI

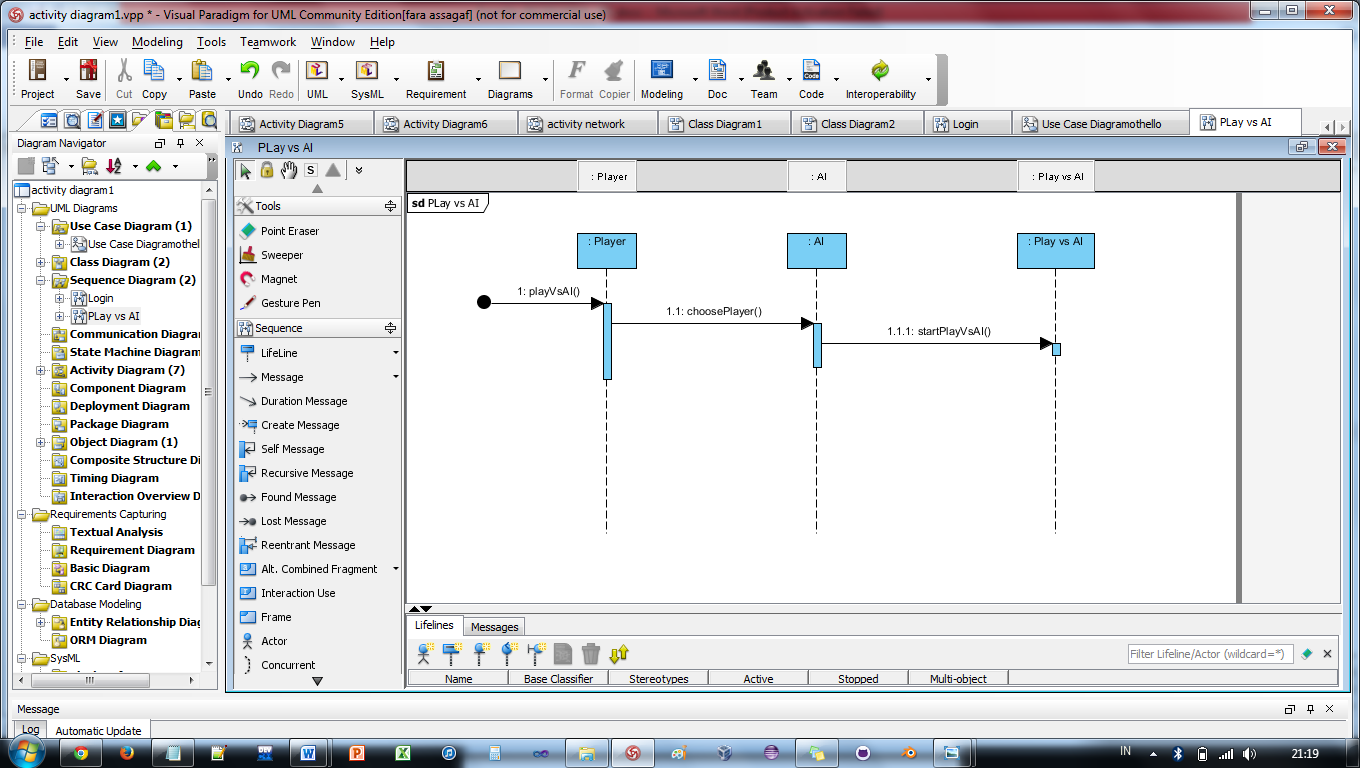


Figure 10

### 4.1.3. Sequence Diagram Play Game

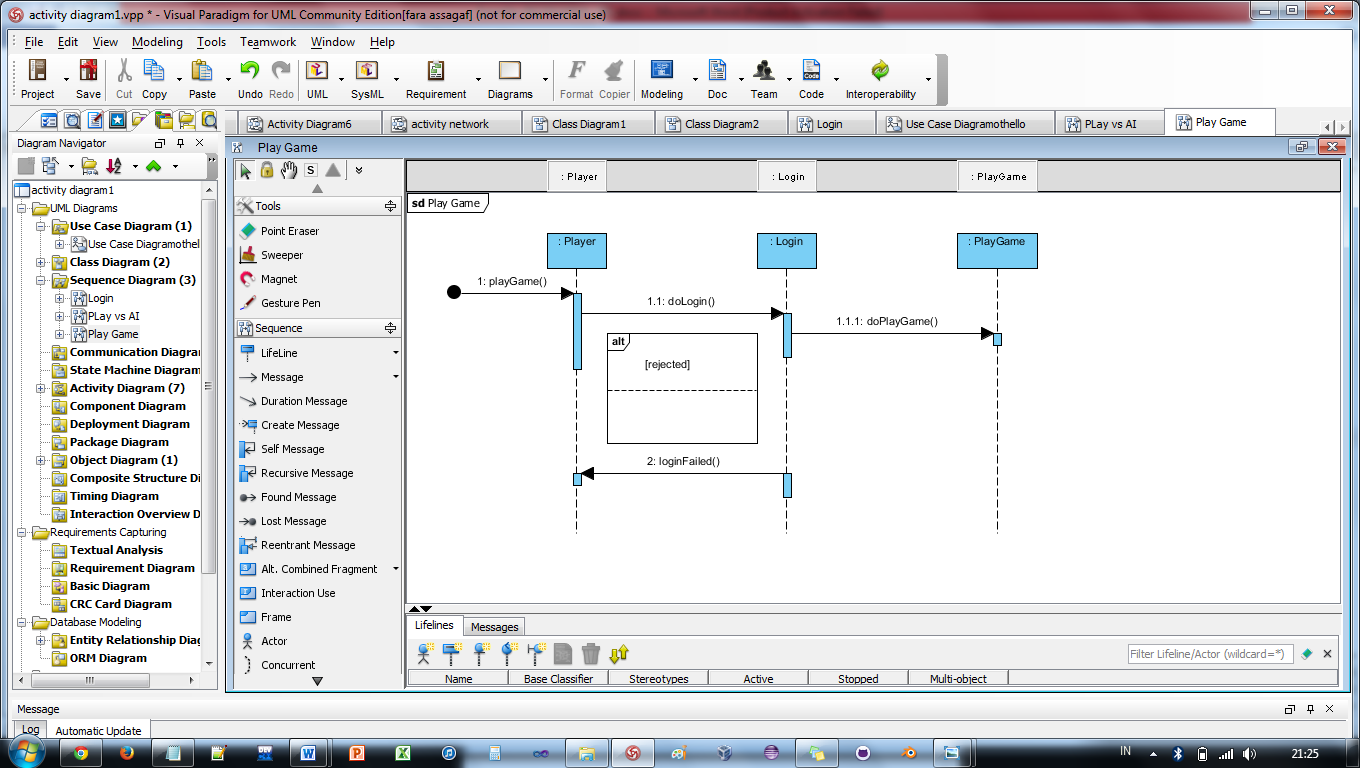


Figure 11

### 4.1.4. Sequence Diagram View Ranking

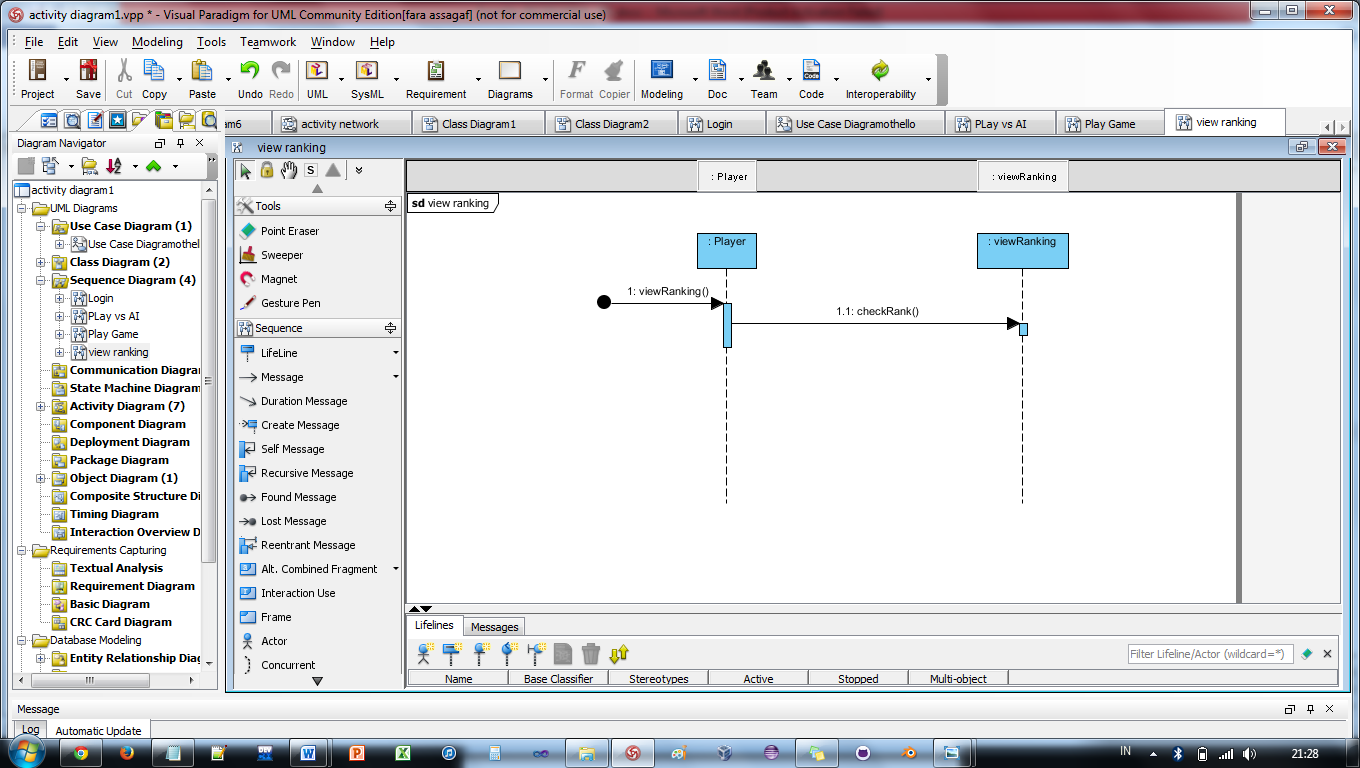


Figure 12

### 4.1.5. Sequence Diagram Manage Game

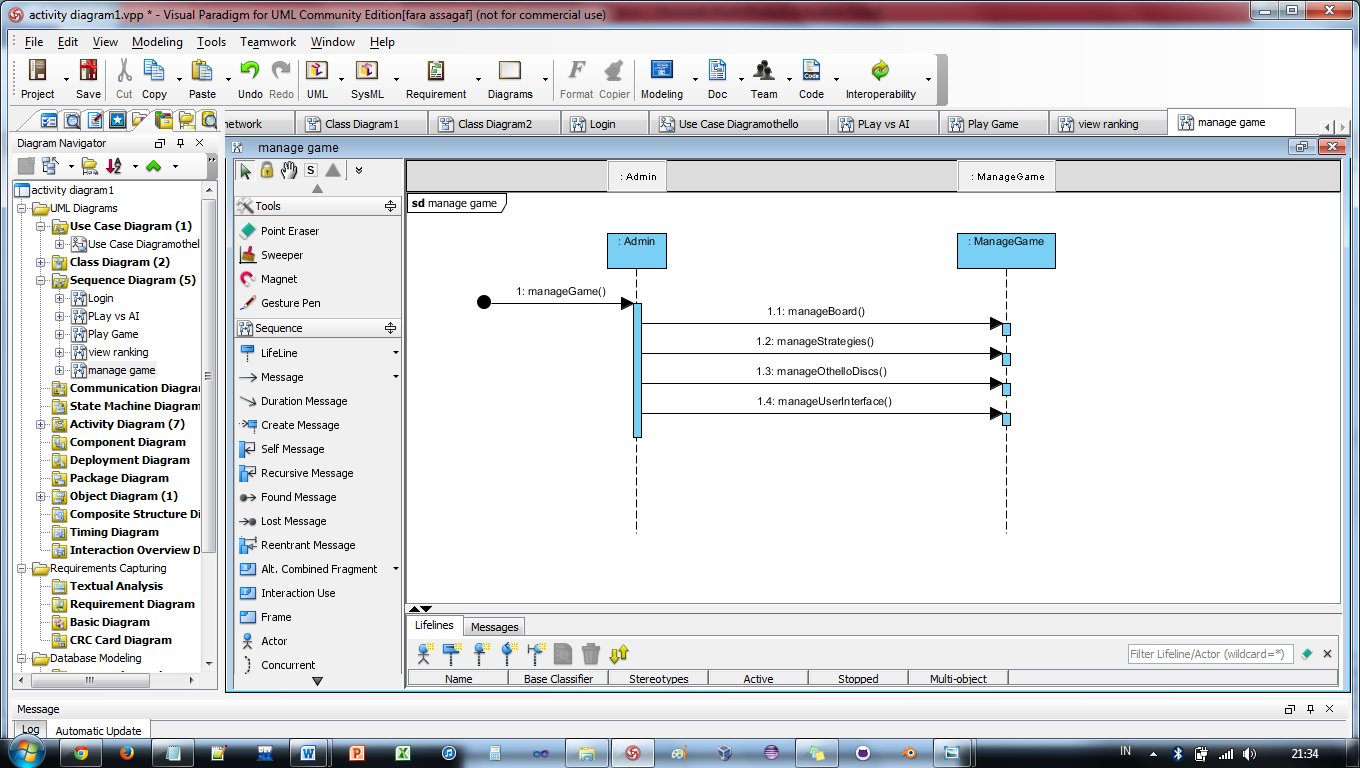


Figure 13

### 4.1.6. Sequence Diagram Update Score

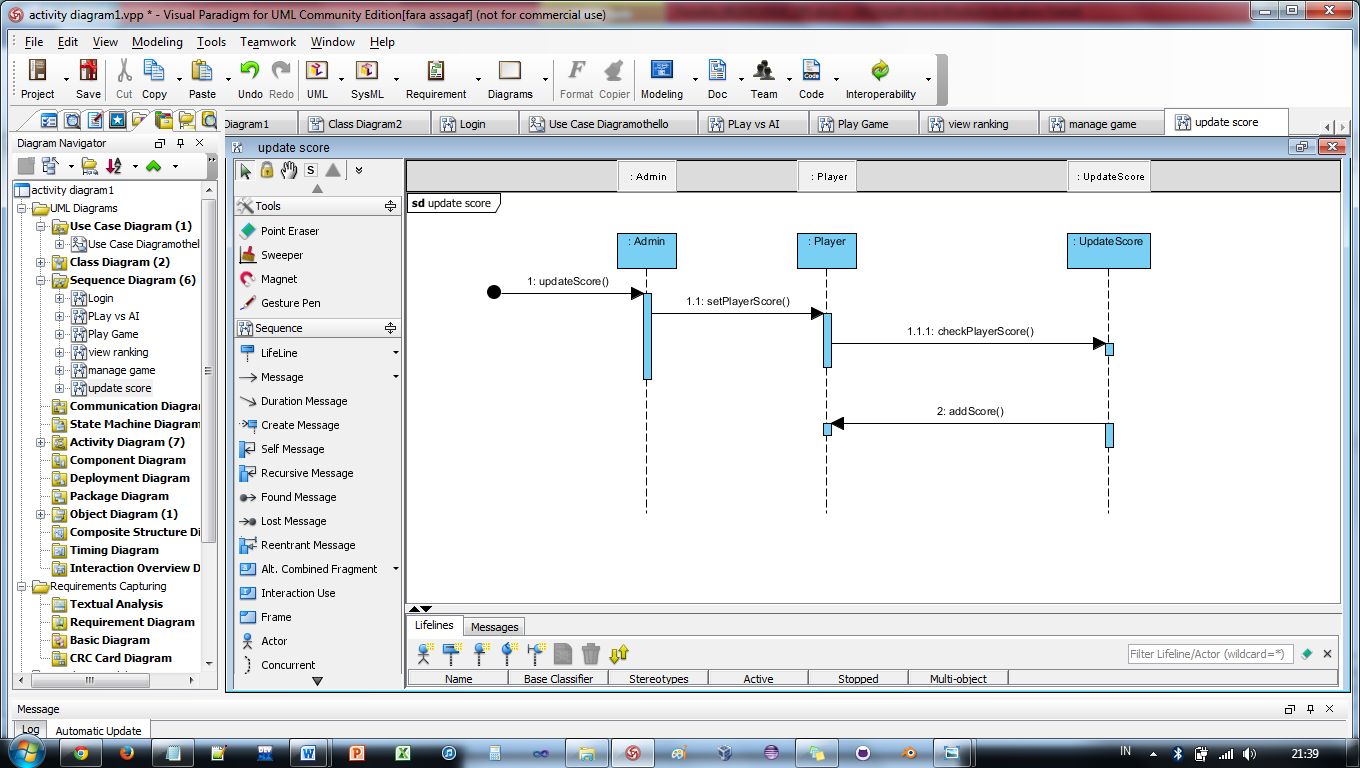


Figure 14

## 4.2. State-Transition Diagrams (STD)

### 4.2.1. State Transition Diagram Login

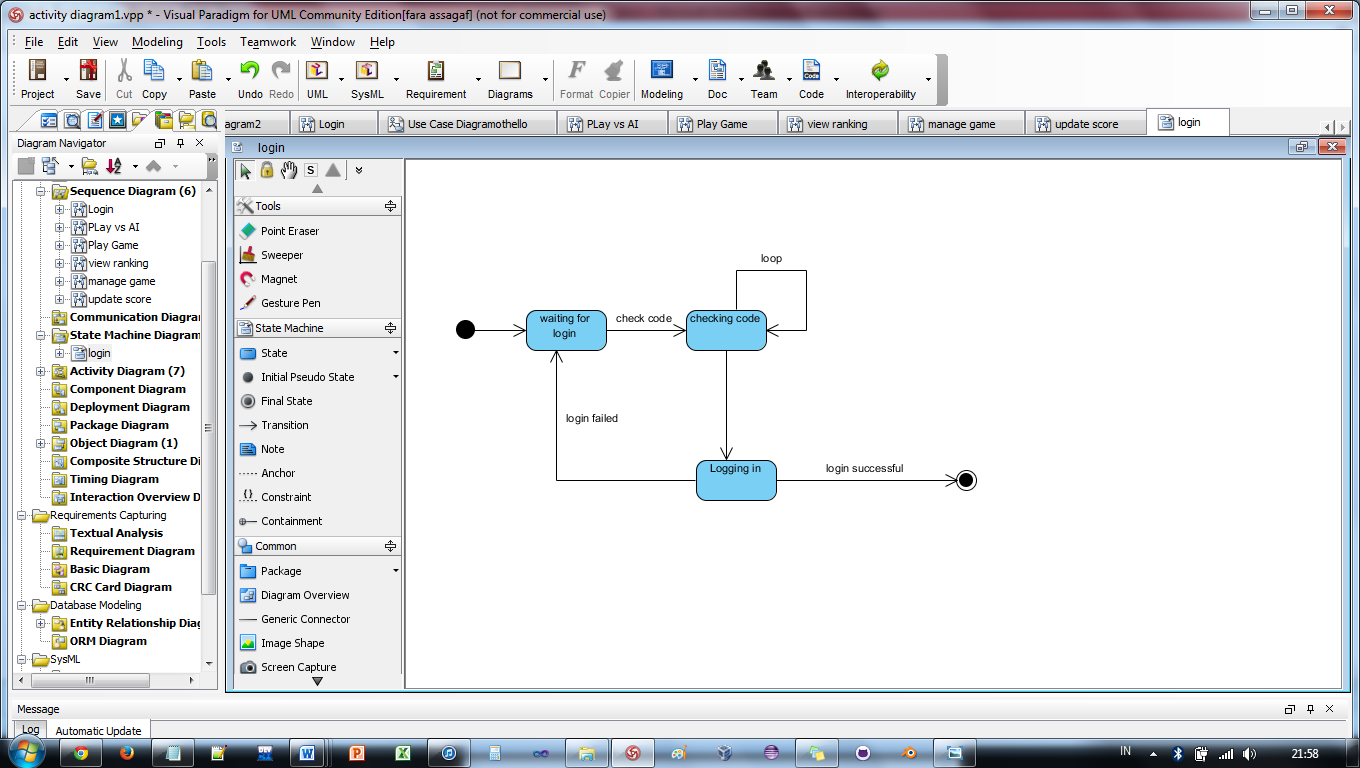


Figure 15

### 4.2.2. State Transition Diagram Play vs AI

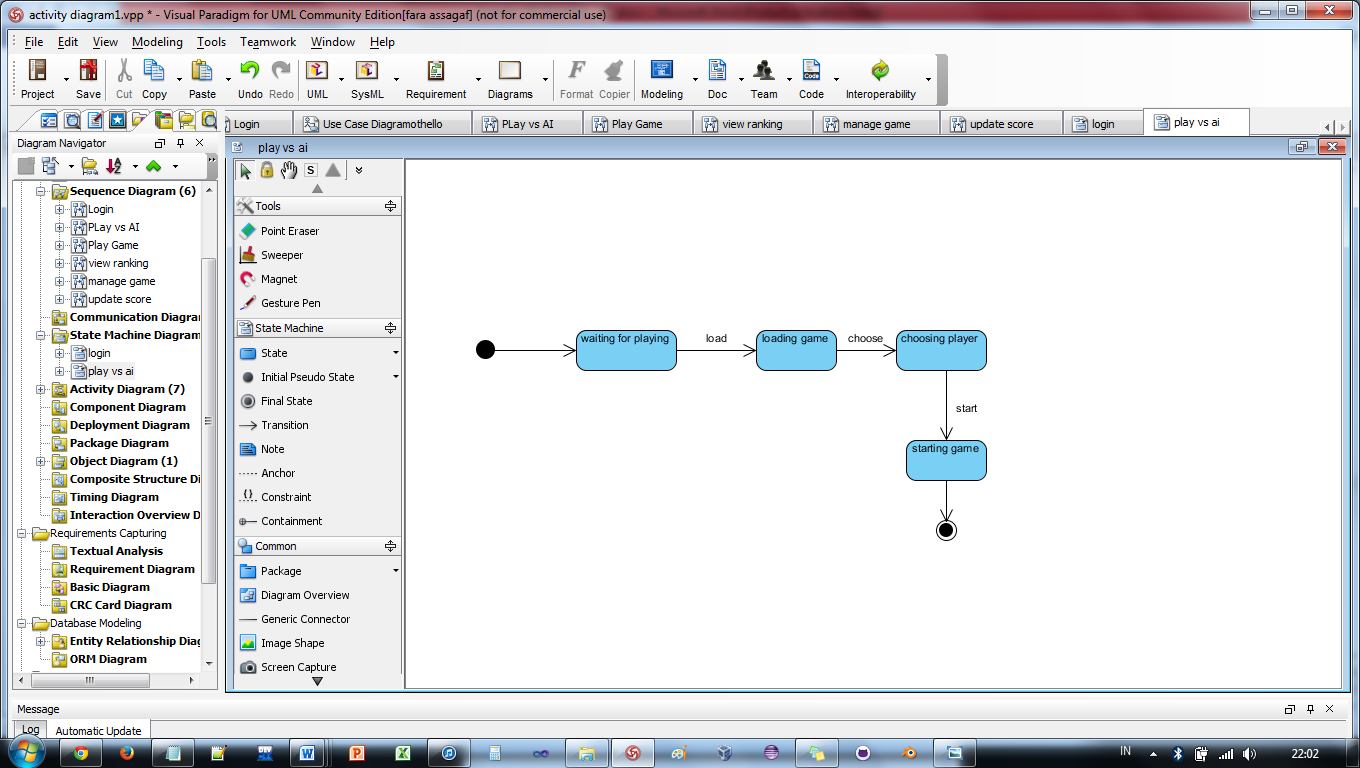


Figure 16

### 4.2.3. State Transition Diagram Play Game

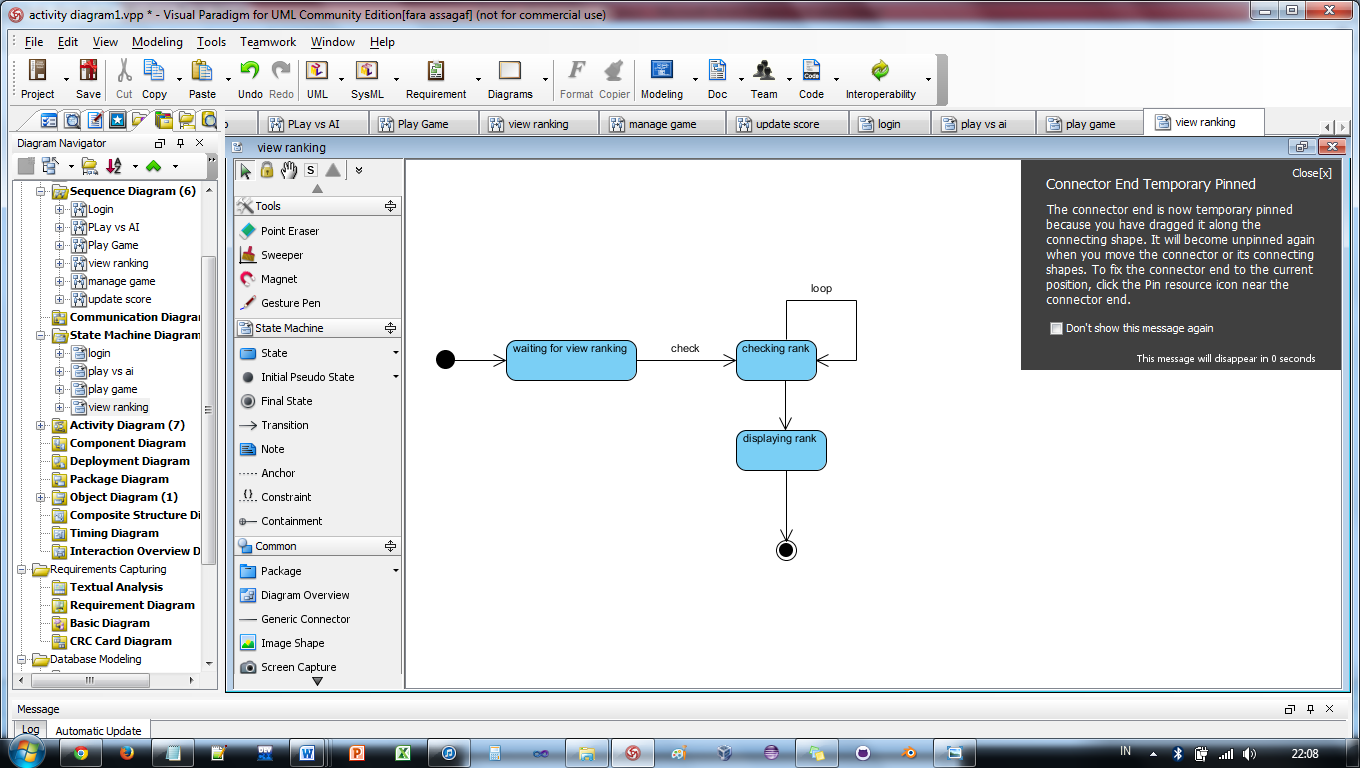


Figure 17

### 4.2.4. State Transition Diagram View Ranking

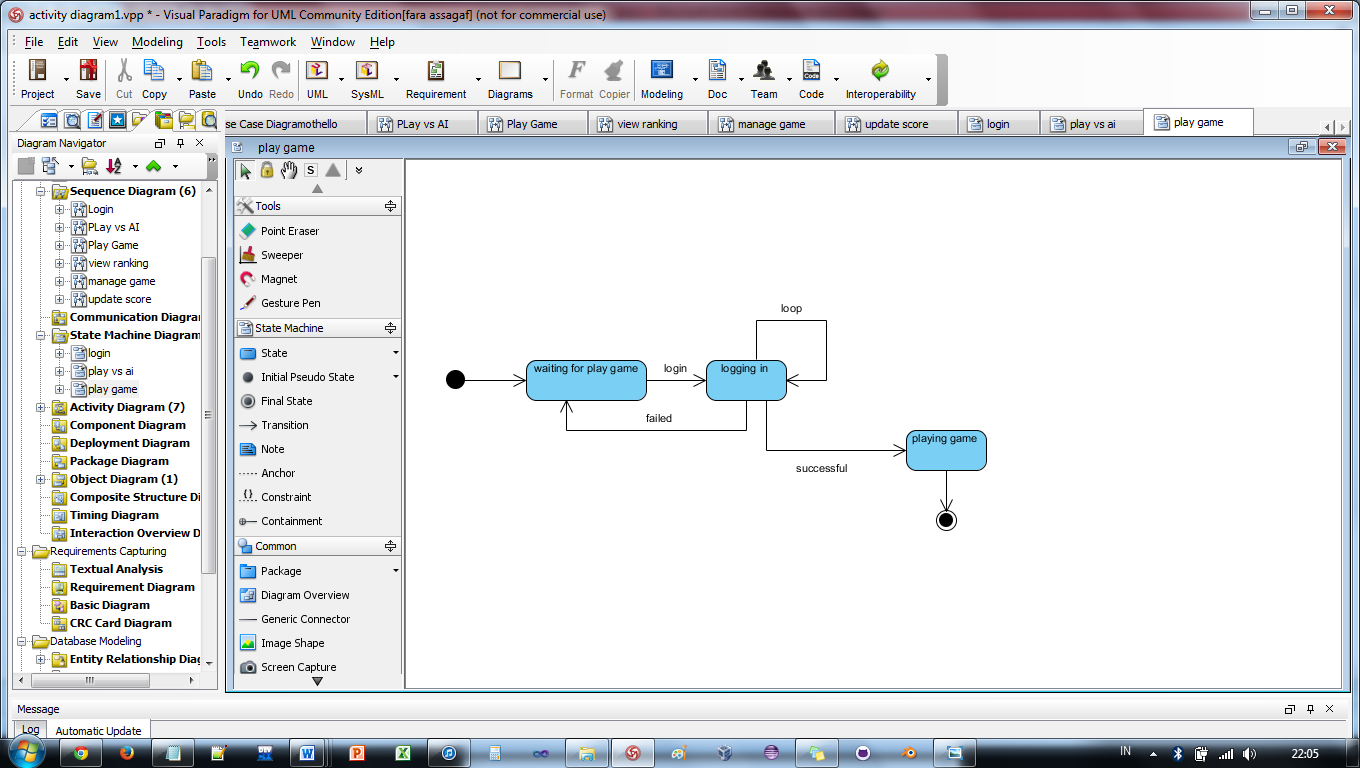


Figure 18

### 4.2.5. State Transition Diagram Manage Game

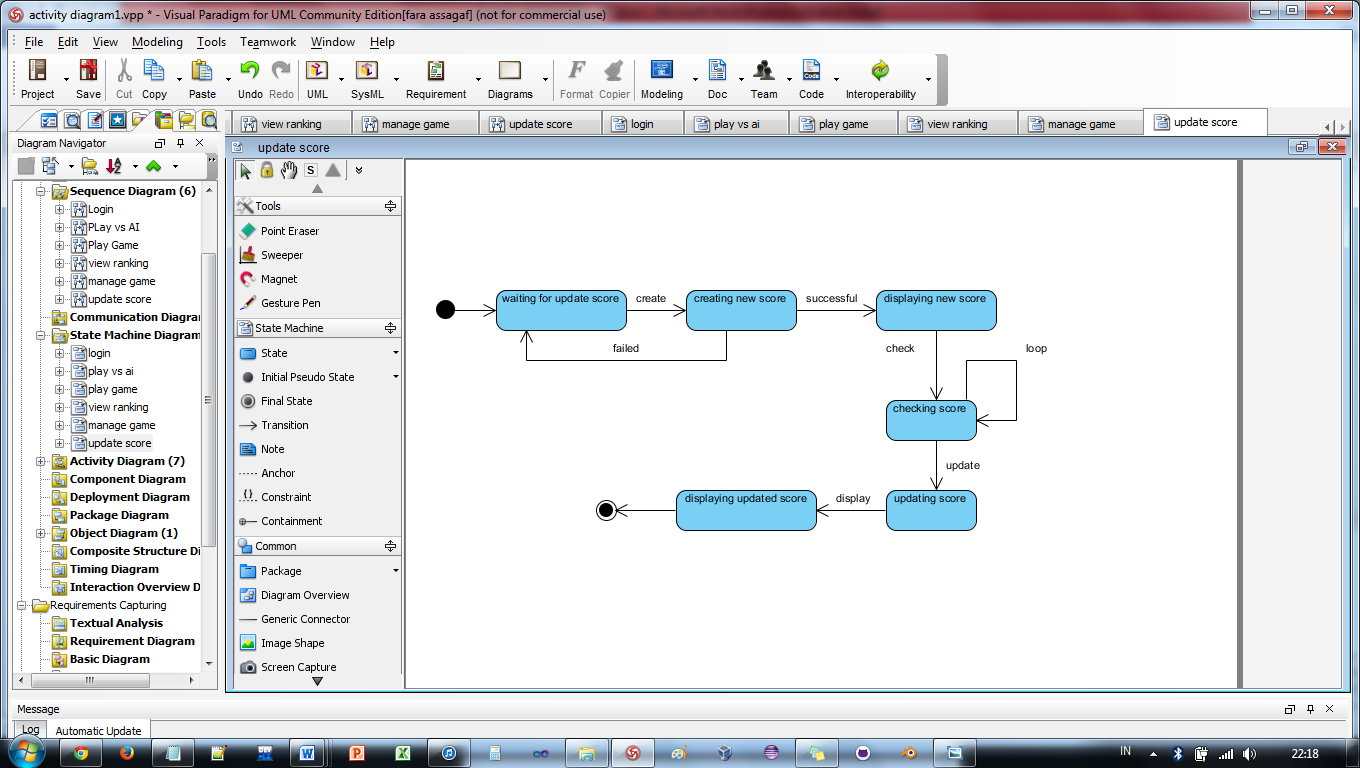


Figure 19

### 4.2.6. State Transition Diagram Update Score

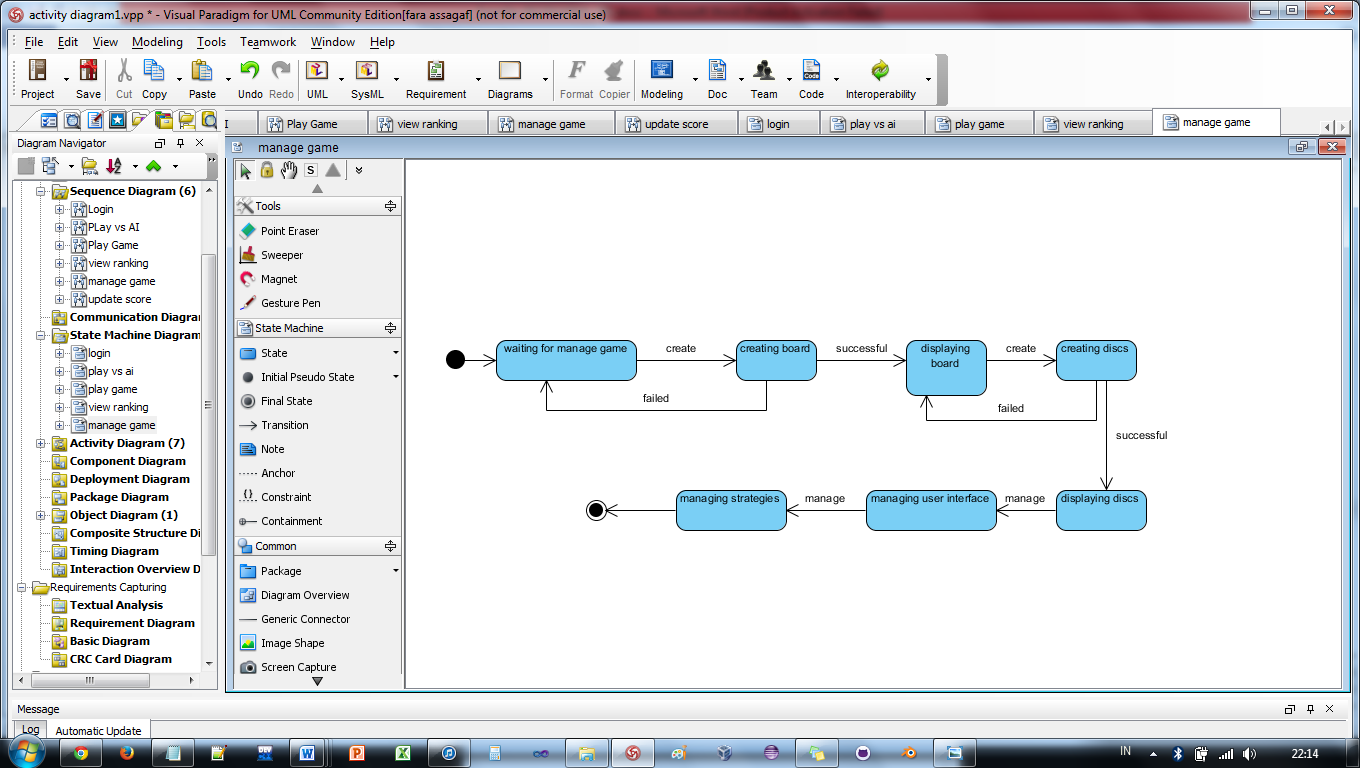


Figure 20

Chapter 5

# CHANGE MANAGEMENT PROCESS

|  |  |
| --- | --- |
| Project | Othello |
| Req. No. | 1 |
| Change Specification | System support for devices |
| On Schedule | Nil |
| On Effort | 10 person-days |
| Status | Will be incorporated in new devices |
| Impact Analysis | Schedule estimation, consistency checking, risk analysis |

Table 2

# Appendices

## A.1. Appendix 1

|  |  |  |
| --- | --- | --- |
| Requirement ID | Title | Requirement type |
| OP 001 | Login | Function |
| OP 002 | Play vs AI | Function |
| OP 003 | Play Game | Function |
| OP 004 | View Ranking | Function |
| OP 005 | Manage Game | Function |
| OP 006 | Update Score | Function |

Table 3