



Department of Computer Engineering

Bilkent University

# CS-319 Project

*Your Story: An RPG Simulation Portal*

## Analysis Report

Ali GÜNEŞ, Cevat Barış YILMAZ, Erin AVLLAZAGAJ, Ertuğrul AKAY

Course Instructor: Uğur DOĞRUSÖZ

October 29, 2016

This report is submitted to the Github in partial fulfillment of the requirements of the Object Oriented Software Engineering Project, course CS319

# Contents

- [1 Introduction](#)
- [2 Overview](#)
  - [2.2 Functional Requirements](#)
  - [2.3 Non-functional Requirements](#)
  - [2.4 Pseudo-Functional Requirements](#)
- [3 Use Cases](#)
  - [3.1 Use-Case Diagram](#)
- [4 Object and Class Model](#)
- [5 Dynamic Models](#)
  - [5.1 Play Game Sequence Diagrams](#)
    - [5.1.1 Login Sequence Diagram](#)
    - [5.1.2 Create Lobby Sequence Diagram](#)
    - [5.1.3 Start Game Sequence Diagram](#)
    - [5.1.4 Play Game Sequence Diagram](#)
  - [5.2 View and Update Profile Sequence Diagram](#)
  - [5.3 Lobby State Diagram](#)
  - [5.4 AccessHandler State Diagram](#)
  - [5.5 VotingHandler State Diagram](#)
  - [5.6 Profile State Diagram](#)
  - [5.7 Seat State Diagram](#)
  - [5.8 Activity Diagram](#)
- [6 User Interface](#)
  - [6.1 Login UI](#)
  - [6.2 Home Page](#)
  - [6.3 Lobby](#)
  - [6.4 Profile](#)
  - [6.5 In-Game UI](#)
  - [6.6 Lobby Creation UI](#)
- [7 References](#)

# Analysis Report

## 1. Introduction

Our group's project decision is creating a game, which is called "Your Story". We decided to develop a game which has completely different gameplay from the other small games. "Your Story" is a text-based multiplayer role-playing game, which will be implemented in Java. The game will be designed for desktop computers. There will be different scenarios and different kinds of characters in the game, so you can select the story and the character you want to play and maintain the game by texting the other players.

The purpose of the game is having fun by chatting with the other players in order to develop a story and acting like a character you always wanted to be. The players will get some points for each game they play. The points and the stories that player participated before will be shown on the player's profile. The players can also read the stories played by the other players to have fun.

The detailed information about the content of the game will be presented in the overview section. This report contains detailed gameplay information and the use cases of the project in the upcoming sections. The first following section is the overview where we describe all the details the game has. This section is then followed by requirements also divided as functional, non functional and pseudo functional accordingly. Right after this will be the use case diagram followed by a detailed description of all the use cases.

## 2. Overview

"Your Story" is a fast, easy and creative alternative portal for online text-based RPGs. The desktop client will let the players login anonymously, create a personal game profile, group up with other players and chat with them simultaneously to develop a story.

One of the essential features of YS is called "Lobbies". Lobby is basically a game room where people can join before entering the game, just

an intermediate step to make sure the required number of players is met before the game starts. Within the available selection of lobbies that are currently waiting for players, the user can choose to participate in any of them displayed in the main screen as a list or create their own lobby. The lobbies in the list will have displayed the number of players currently in, the total number of players, the environment and the player who created the lobby.

If the player decides to make his own lobby, once created, he/she will be prompted to choose a timeline, environment, a background story, and maximum number of players (this will be a number of the range from 2 to 10, both ends included). After all the selection is completed the lobby is created and is automatically viewed to the lobby list for other users to join, in meantime the creator of the lobby will be sent to character selection screen to choose the character possibly before other players arrive. This screen will contain all the possible characters (4 initially) which all have a part in the story and also the storyline written so that all the newly joined participants can read before they select the champion and agree to stay in the lobby until the game starts. In the character selection screen the players can also chat with the other people in the lobby to make sure they agree in any issue beforehand or even get to know each other.

If player decides to participate in one of the lobbies he can single-click and enter the lobby. Right after he/she enters, he/she has a time limit of 2 minutes to read the storyline of that lobby and chose a character from the possible (initially) 4 selections of character or has the last chance to leave, before getting stuck in the game. After 2 minutes the player will be assigned a random character and is destined to wait in the lobby for the quota to be filled or for people to vote for the game to start. Then, depending on their selections, players will act like that character throughout the rest of the story. The game starts if the lobby has filled its total capacity, or the players voted in favor of starting the game. The game starts with a description of timeline, place, characters, and occurring events which are specific to that room. With the explanation of events that are happening at the start of the game, players will have to choose what to do about this event, and how to improve / finish it.

The voting is fairly easy. The players who have chosen their character have the right to cast a voting to start the game every 10 minutes, starting 10 minutes after selecting a character for the rest of their game. The only players who can then vote are the ones who have already chosen their

character before the voting was casted. The game will then start if more than 50% of the players participating in the vote agree to start the game. The ones who still haven't chosen will be kicked out of the lobby.

After this point, players will develop their story as they wish in the chat section of the game, that will appear once the game starts. At the beginning of the game the users will have pre-written for the second time the timeline, environment and the story as a reminder. It will also have another window that will show to the player who is online and who is not. Through messaging, players can read others' messages and send replies once at a time (a player can't send two messages simultaneously, this is to prevent spam traffic to our server and annoying other payers).

Once the game has started, it can continue even if the players want or need to disconnect so that the players don't have to chat for hours until the story is finished. When they are finished with their story (which means they are satisfied with the "finalé" they created), or want to end the game by will, any player can trigger a voting system. The players who aren't online have a 2 day period from the day the voting has started to cast their vote otherwise the decision will be taken by the ones who voted, by completely disregarding the rest of the players. What is special about our decision making system is that the players' votes won't weigh equally for all the players. The players' votes will weigh over the contributions they gave to development of the story.

The algorithm will check the number of messages the players have sent. The user with more messages will have have a stronger vote. This will prevent the AFK players and trollers to have the same vote weight as a serious player. Once the game ends, a copy will be saved into players' profile pages. Their story will be automatically published into "Recently Finished Games" section and will let the other users read it. Also the players of the game will be rewarded some scored depending on how many messages they wrote.

The database will create connection to the clients through and will send data like: voting started, current votes, new chat messages coming up or new lobbies created. These will keep the client updated all the time(every one second to be exact). If the client decides to get information about a player's profile or wants to read a finished story then the database server will respond with the appropriate data. Likewise, the database will get data from the client like chat messages, profile, vote etc. Then the database will automatically update all the clients involved in that communication, so that

everything is done in real-time and there is no time loss between chat messages. The database will also keep all the valuable information like username, password, chat messages, profiles, keys, lobbies in a MySQL database. Our database server will always be connected to the database and will be doing all the needed insertions, deletions, updates or selections according to the clients' demands. The connection will be kept open until the user tries to log out then the database will not accept any data once the connection is closed. The main login will create the connection to database and will maintain it throughout the whole game.

As explained above, the players will play in a setting which is exclusive to the lobby. Settings are prepared to ignite the players' creative side, give them a kickstart as we want them to start role playing immediately, and not waste their time with preparation discussions. In a future version of the game, there might be more than ten different storylines with interesting events that the players will have to solve. Since creating the lore takes serious time and afford, the sample settings below only presents the current discussions and possible settings that our group discussed together.

#### **-Sample Story Settings-**

**Timeline:** Middle Ages

**Environment:** Medieval Europe

**Background Story:** A war is rising again between kingdoms!

**Timeline:** Year 2100

**Environment:** Planet Mars

**Background Story:** Mars inhabitants fight against invader from Earth

**Timeline:** July 1974

**Environment:** France

**Background Story:** A the French revolution against their corrupt government broke off!

**Timeline:** Year 2500

**Environment:** Earth

**Background Story:** Aliens have taken control of Earth. Humans are now slaves!

#### **-4 Sample Characters-**

- Alien
- Assassin

- Knight
- Soldier

## 2.1. Functional Requirements

- Users should be able to group up with other players in lobbies

Lobbies are heart of the game. Player will find his companions for the next game in lobbies and they will be bounded until the game ends. All players either can find a lobby that has an empty seat or create an entirely new one.

- Users should be able to communicate with other players

Communication with other players is one of the key elements of having fun. While the game itself is based on chatting, players will also have opportunity to chat while waiting for game to start in lobbies.

- Users should have a profile page and be able to update it as she/he desires

Profile pages will contain a profile picture and a description written by their owner, so that players be able to tell something about themselves to other players easily.

- Users should be able to choose a character for each game

Characters will be the main point of how players will develop their story. Different characters for every game will make game more interesting and replayable for longer time. Once player enters a lobby, he will be able to select his own character from the list of characters for that specific scenario.

- Users should be able to play the game by developing a story with other users according to their characters and pre-defined game conditions

The game is based on developing a story with other players. According to predefined scenario user will be able to evolve a story according to their characters and the contribution of other players.

- Users should be able to vote for decisions about the current lobby they are in

Decisions about the lobby will be taken by a voting system. All players will be able to decide together for the decisions like kicking a player, starting or ending the game.

- Users should be able to choose the game/lobby he wants to participate

Making friends and playing with them is usually funnier than playing with random people. Choosing the lobby will let players to play with their favorite companions or friends that he made throughout the game.

- User should be able to continue her/his unfinished previous games

Developing a good story usually takes longer than one session. By being able to continue games later will make sure players to have enjoyment of a longer story that they can contribute anytime they want.

- Users should be able to see informations about other players and view their profiles

Users will be able to check other players' profile pages and have more information about them easily. Also they will be able to see the previous games that played by the owner of that profile.

- Users should be able to read finished games

Of course stories are written to be read by other people or even by its authors. Players will be able to read any story that written and published in the game so that no good story will be shelved and forgotten.

- The whole system should be able to update in realtime. This means that the moment the client starts and the user logs in the system should build up a reliable connection with the client and keep it updated on any notification or even any kind of chatting box the player uses, or even the score.
- The database should handle multiple connections and update all the clients accordingly. No message between two people can be sent to another random person.
- The database shouldn't accept changes in scores of the player when the client is reconnected to avoid abuse, only chat messages the user typed will be sent as a block message if the user types many messages, any other change will be ignored. Also the profile picture can be sent together with the bio.



- The scores shouldn't be saved in the local machine since the user can tweak it, so it must be saved in the database.

## 2.2. Non-Functional Requirements

- The database should have high performance and efficiency to refresh quickly all the client it is connected to without any user experiencing a lag.
- The interface should be as understandable as possible so that the user doesn't have to go through any tutorial on how to use it. And it should also send the user to their destination with as few clicks as possible.
- The software needs to have many classes to be able to handle complexity so that it can be easier to maintain and extend its features later on, like adding more stories or characters.

## 2.3. Pseudo-Functional Requirements

- No game engine should be used.
- The database server must work with SQL.
- No other language than PHP, Java, C/C++, Javascript, Python shall be used.

## 3. Use Cases

### Use case #1: View Profile

**Participating actors:** Player

**Entry condition:** Player has already logged in.

**Exit condition:** Player clicked out to go to main menu.

**Main flow of events:**

1. Player selects to view any player.
2. Player views description, profile image and list of finished games of the selected player.
3. Player exits the profile.

**Alternative flow of events:**

- Player lost his internet connection. (display appropriate message and go to step 3)

### **Use case #2: Update Profile**

**Participating actors:** Player

**Entry condition:** Player has already logged in.

**Exit condition:** The player has successfully updated his profile or did not save changes.

**Main flow of events:**

1. Uploads a profile picture and / or edits his description.
2. Updates his changes.

**Alternative flow of events:**

- Player uploads a not supported image format. (display appropriate message and go to Step 1)
- Player enters an unsupported character (display appropriate message and go to Step 1)
- Player does not save his changes. (display appropriate message)

### **Use case #3: Play a Story**

**Participating actors:** Player

**Entry condition:** Player has already logged in.

**Exit condition:** Player is either kicked from the lobby, game did not start, player left the lobby.

**Main flow of events:**

1. Player chooses an available lobby from the list, or creates one by himself.
2. If the player creates a lobby, system will assign a random story to the lobby.
3. In either way, player chooses a character from a list that is given by the system.
4. Players who have chosen their character either vote for starting the game or wait for more people to join.

5. The voting occurs automatically every 5 minutes. When the voting is passed with the majority pro, the game starts.
6. The system represents the description of settings and characters.
7. Players develop the story by messaging alternatively.
8. Players choose to end the game at any time by voting. When the voting is passed with the majority pro, the game ends.
9. The system saves the story into the players' profiles.

**Alternative flow of events:**

- Player does not choose his character within 2 minutes, he is kicked from the lobby. (Go to step 1.)
- Player disturbs the flow of the game, after the game starts. Other players vote to kick him. (Go to step 1 if voting is passed)
- Player loses his internet connection. (If the game did not start, go to step 1. Else, go to step 7. )
- Player leaves the game before it is finished. (Player can reconnect to the same game any time.)

**Use case #4: Read a Finished Game**

**Participating actors:** Player

**Entry condition:** Player has already logged in.

**Exit condition:** Player finished reading.

**Main flow of events:**

1. Player finds a finished story from the list of "Recently Finished Stories" or from any player's profile.
2. Player reads the story.
3. Player exits.

**Alternative flow of events:**

- Player lost his internet connection. (display appropriate message and go to step 3 )

### 3.1. Use Case Diagram

Visual Paradigm Standard Edition(Bilkent Univ.)

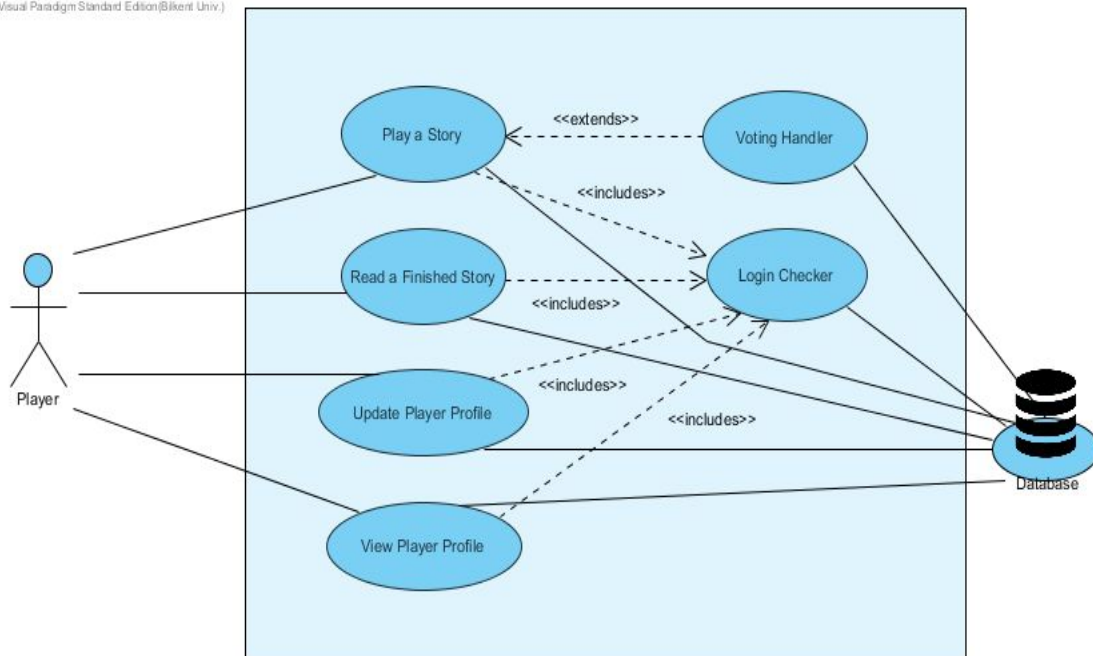
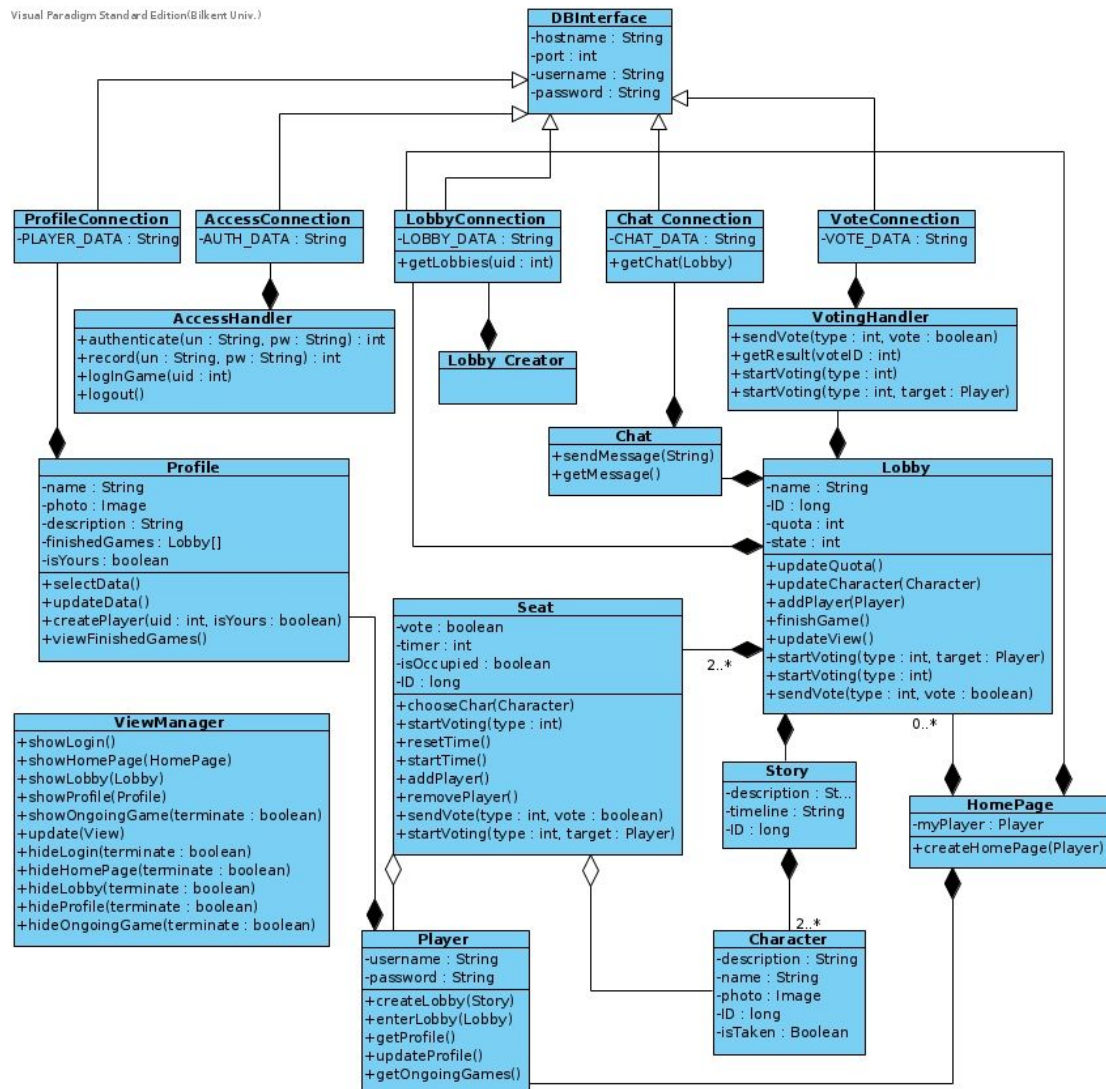


Figure 1: Use Case Diagram of "Your Story"

## 4. Object and Class Model

Visual Paradigm Stand and Edition (Gilkert Univ.)



## 5. Dynamic Model

### 5.1. Play Game Sequence Diagrams

### 5.1.1. Login Sequence Diagram

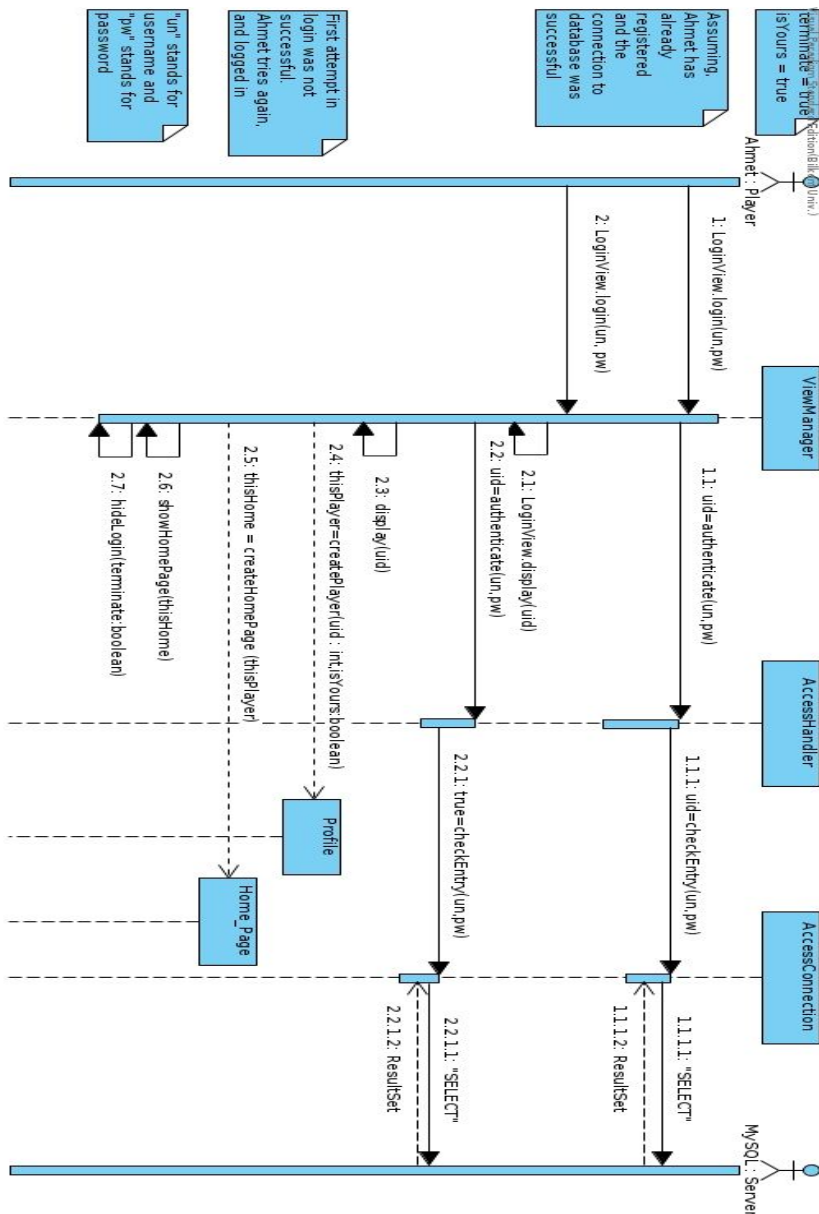


Figure 2: "Login" sequence diagram of "Your Story"

Scenario: Ahmet is already registered in the system. He wants to login to play. First try he entered wrong credentials and the system rejected him. Second time he fixed the mistake and he successfully logged in, where he now sees the main page from where he can do anything.

Design: The ViewManager at first will display the login view where Ahmet should enter the username and password. Upon clicking "Login" the ViewManager will call AccessHandler class which will get the username and password of Ahmet and send them to the AccessConnection class which handles the connection to the right table in the database specified at "PLAYER\_DATA" constant and sends the right query to verify if user entry

exists. After that the connection returns the -1 as uid since the username and password combination was not found in the table so the ViewManager displays a message according to the uid(in this case an error). The second try MySQL server gives a number > 0 Ahmet is finally logged in. ViewManager will destroy completely the login page and will create a Profile according to the uid and run the HomePage for that Profile.

### 5.1.2. Read Finished Story Sequence Diagram

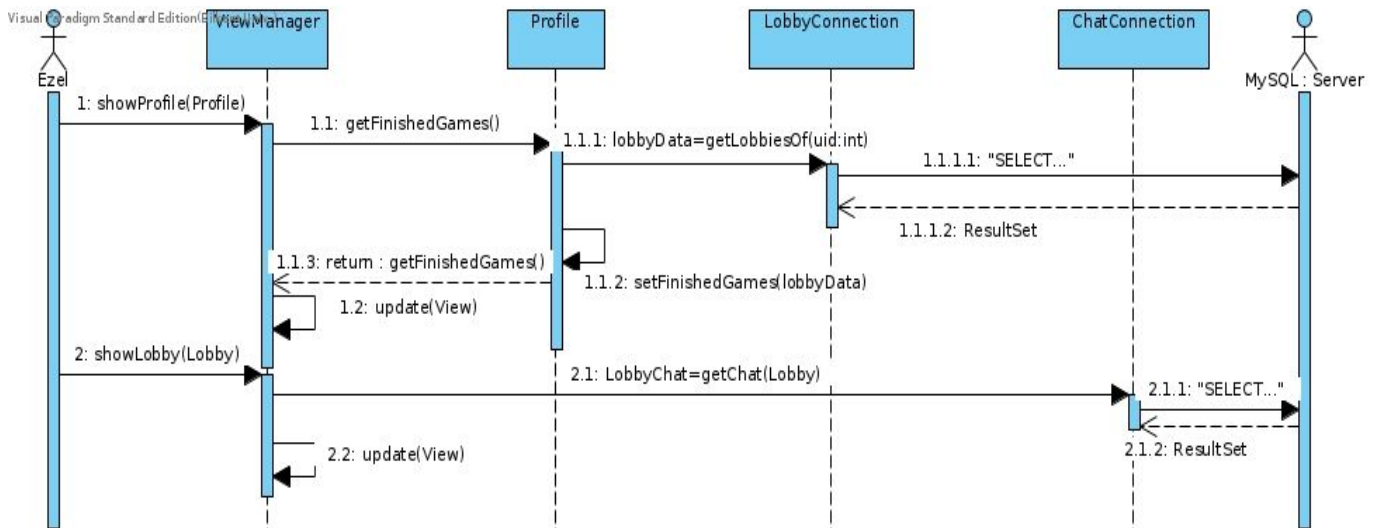


Figure 3: "Read Finished Story" state diagram of "Your Story"

Scenario: After logging in, Ezel decides to view his finished games in his profile. He enters his profile and opens the list of games that he is finished.

Design: This sequence starts with creation of player's profile. View manager gets the related information from server. In profile, player can open his/her old games by clicking a button. When the button is clicked, the profile gets the finished lobby information from the database. Then, the finished lobbies are listed in his profile.

### 5.1.3. Create Lobby Sequence Diagram

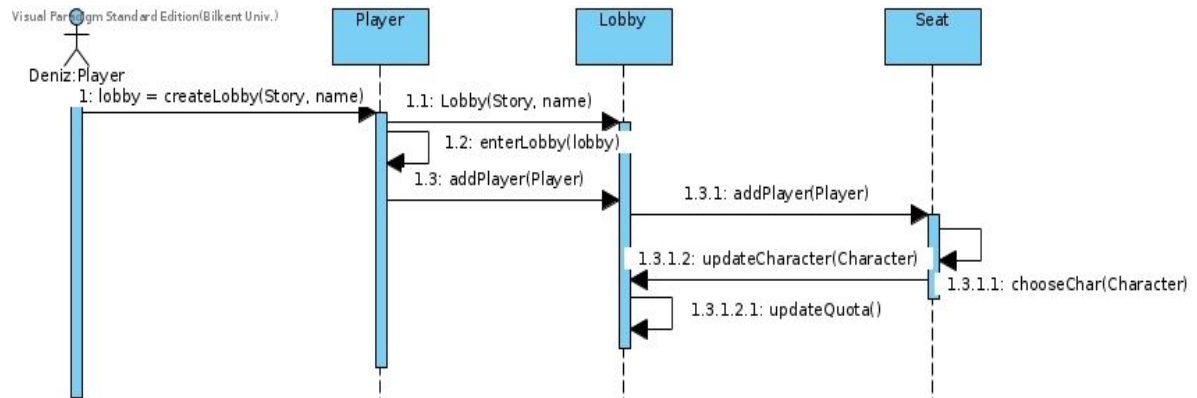


Figure 4: "Create Lobby" sequence diagram of "Your Story"

Scenario: Deniz decides a name and story for her lobby and creates it. Then she chooses a character for herself before other players start to join.

Design: After create lobby action comes from user, player class creates a lobby object and adds itself to it. Then lobby attaches player to a seat in which player chooses a character for himself. Then seat returns the selected character to lobby and lobby sets that character as selected.

### 5.1.4. Start Game Sequence Diagram

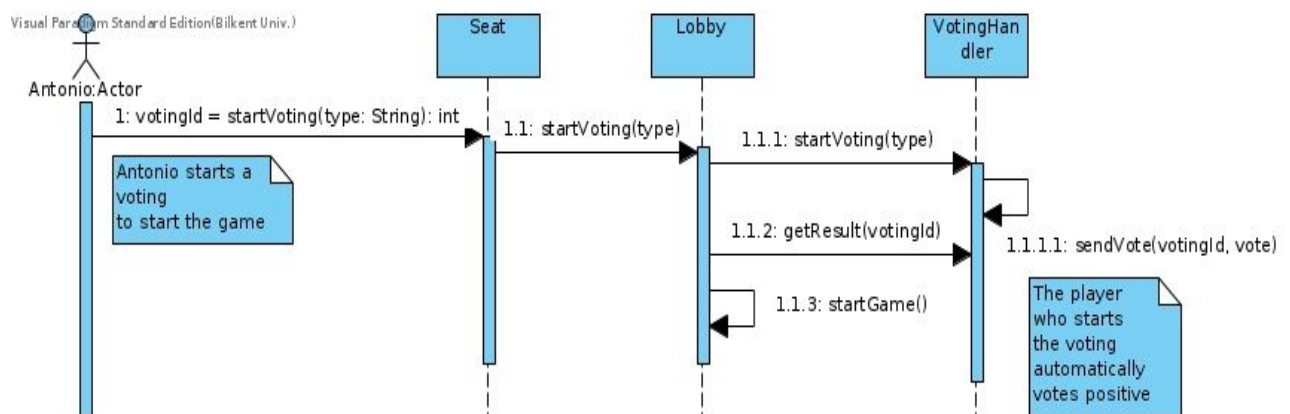


Figure 5: "Start Game" sequence diagram of "Your Story"



Scenario: When lobby reaches its quota Antonio starts a voting to start to game, after the voting time ends, start to game choice wins and game starts.

Design: After start voting action comes from user, user's seat tells lobby to start a voting to start the game, then lobby informs voting handler that a new voting is starting voting handler starts the voting and the starter of the voting automatically sends a positive vote. After voting ends, lobby gets the result from voting handler then starts the game.

### 5.1.5. Play Game Sequence Diagram

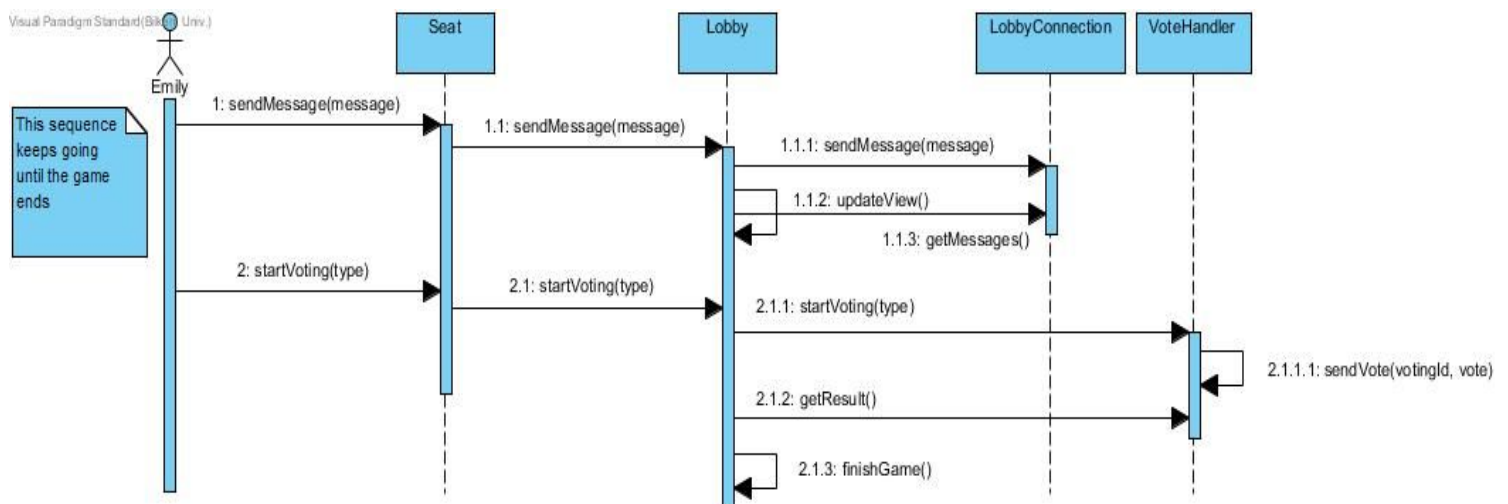


Figure 6: "Profile" state diagram of "Your Story"

Scenario: Emily sends messages as her character according to other players' messages. This keeps going until game comes to an end. Then Emily starts a voting to end the game. After collecting other votes, voting passes and game ends.

Design: As seat is the representer of the player in that lobby, seat class is used to send messages lobby. Lobby class tells LobbyConnection to add the message to database. Then by also collecting other players' messages from LobbyConnection lobby updates it view.

## 5.2. View and Update Profile Sequence Diagram

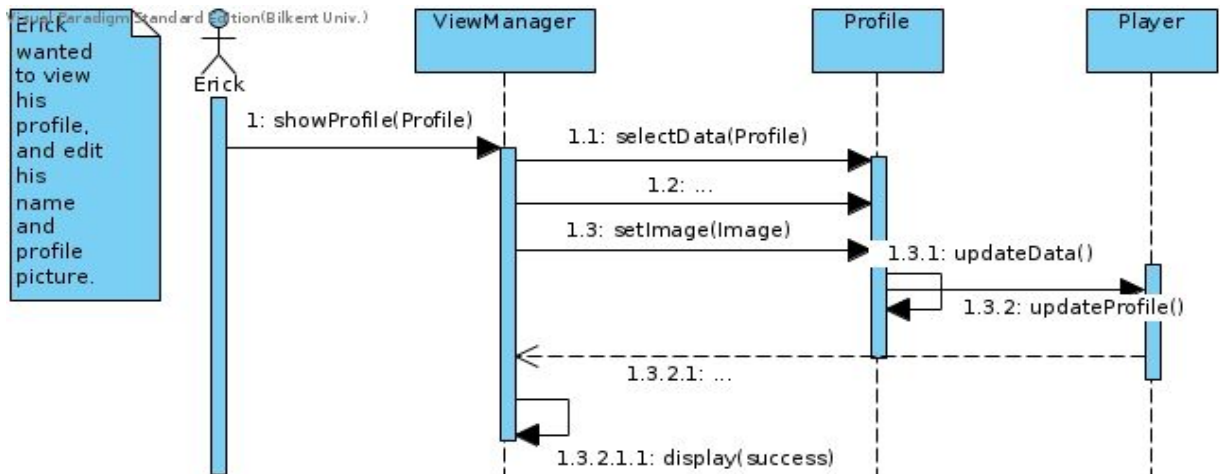


Figure 7: "View and Update Profile" sequence diagram of "Your Story"

Scenario: Erick wants to change his nickname and upload a new image for his profile. Thus, he enters his profile, and change his name as he desires. He chooses an image and uploads to the database. At the end he saves his changes and get a success message.

Design: When the player wants to view a profile, viewManager pulls the player's information and checks if the chosen profile is his or not. If the player wants to look somebody else's profile, system does not allow editing. In this sample it is Erick's profile, so he is able to set a new name and upload another picture.

## 5.3. Lobby State Diagram

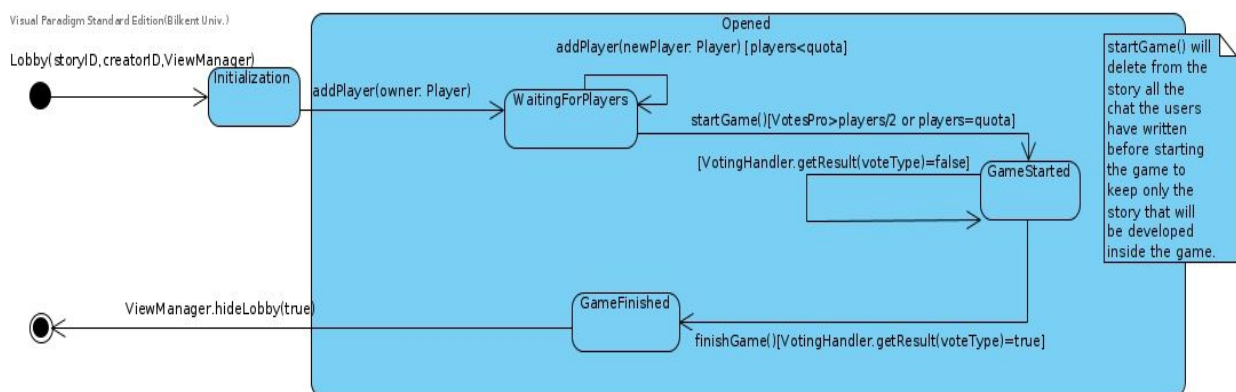


Figure 8: "Lobby" state diagram of "Your Story"

The lobby has exactly 3 states it can be in. The first state right after the initialization it the waiting state where the lobby is supposed to wait until the player quota is filled up or until the current

participants decide to cast a vote in order to start the game. At this state the users inside the lobby can use the chatbox to get to know each other or even make any decision before the game starts. If the voting is successful or if the quota is fulfilled the game will start. The first thing that the game will do is delete all the chat messages so that the story can start. When it starts then the users can exit the lobby and connect to it at any time to give his contribution. The third step is when the players vote to end the game and it was successful. In this state the lobby will save all the chat in the server and will return back to home page to let the user join a new game.

#### 5.4. AccessHandler State Diagram

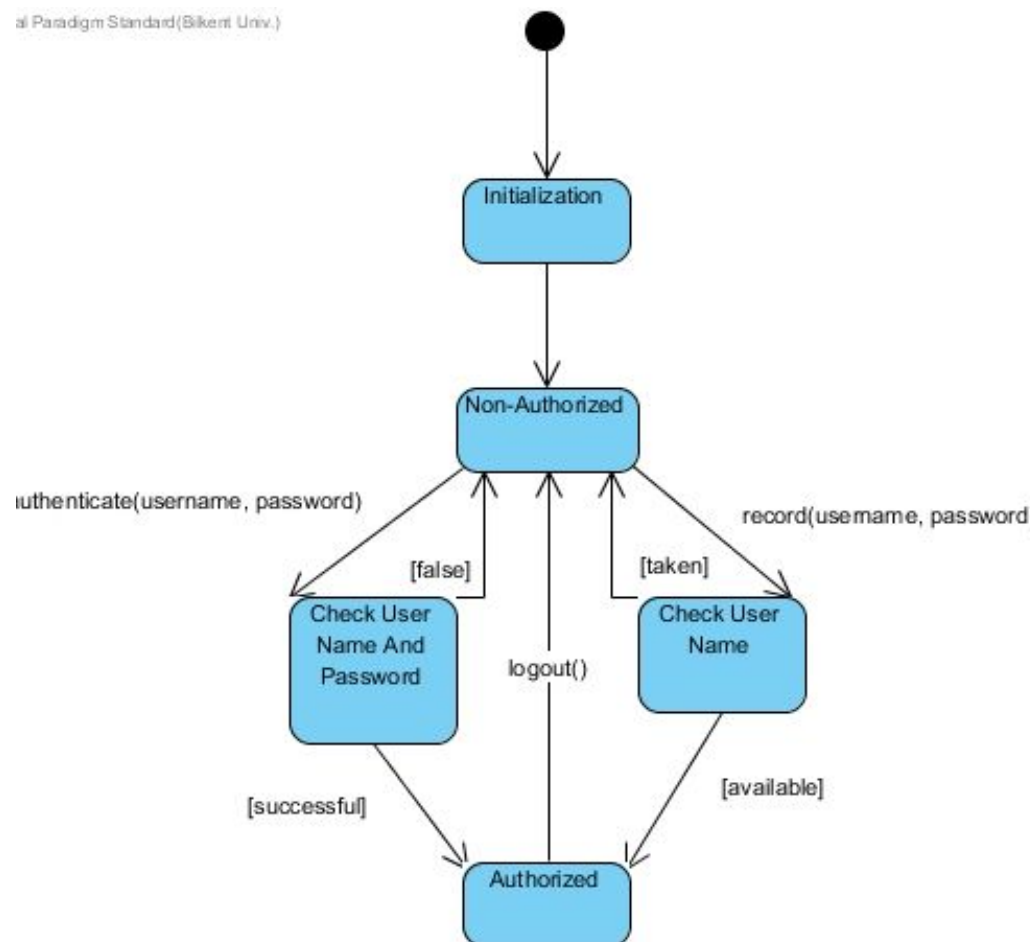


Figure 9: "AccessHandler" state diagram of "Your Story"

In the beginning user starts in non-authorized state. Then by logging with correct user data or registering with an available username makes user authorized. Then the user can logout at any time or just close the game where the handler is destroyed.

## 5.5. VotingHandler State Diagram

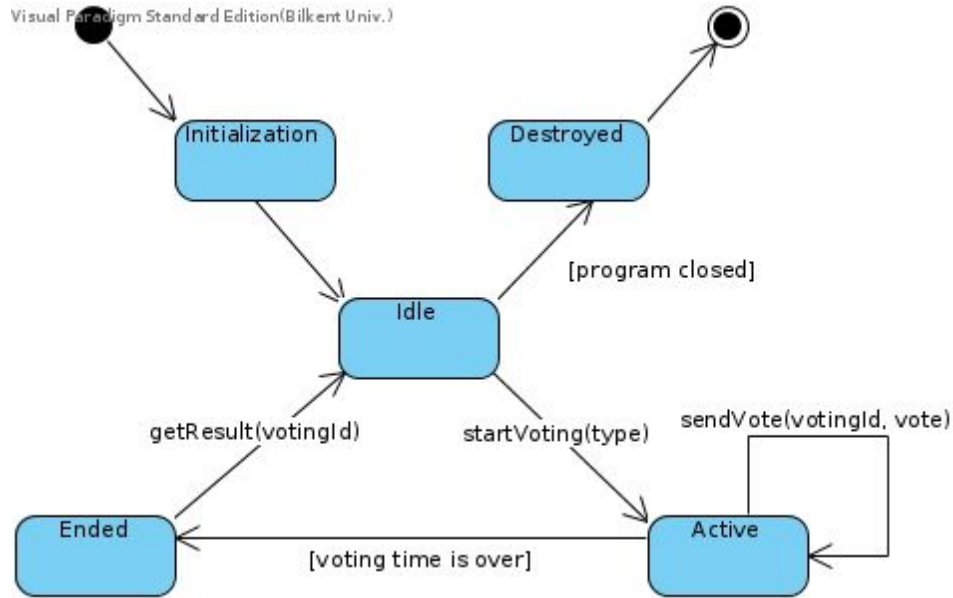


Figure 10: "VotingHandler" state diagram of "Your Story"

Default state of VotingHandler is 'Idle', until a voting starts or it is destroyed. As soon as it receives a 'start voting' method from Lobby, it switches to active state where it will wait until end of the voting time. In the meantime player can send his/her vote to the system. After timer ends, VotingHandler gets the result from database, which will be send to lobby object, then get backs to 'idle' state again.

## 5.6. Profile State Diagram

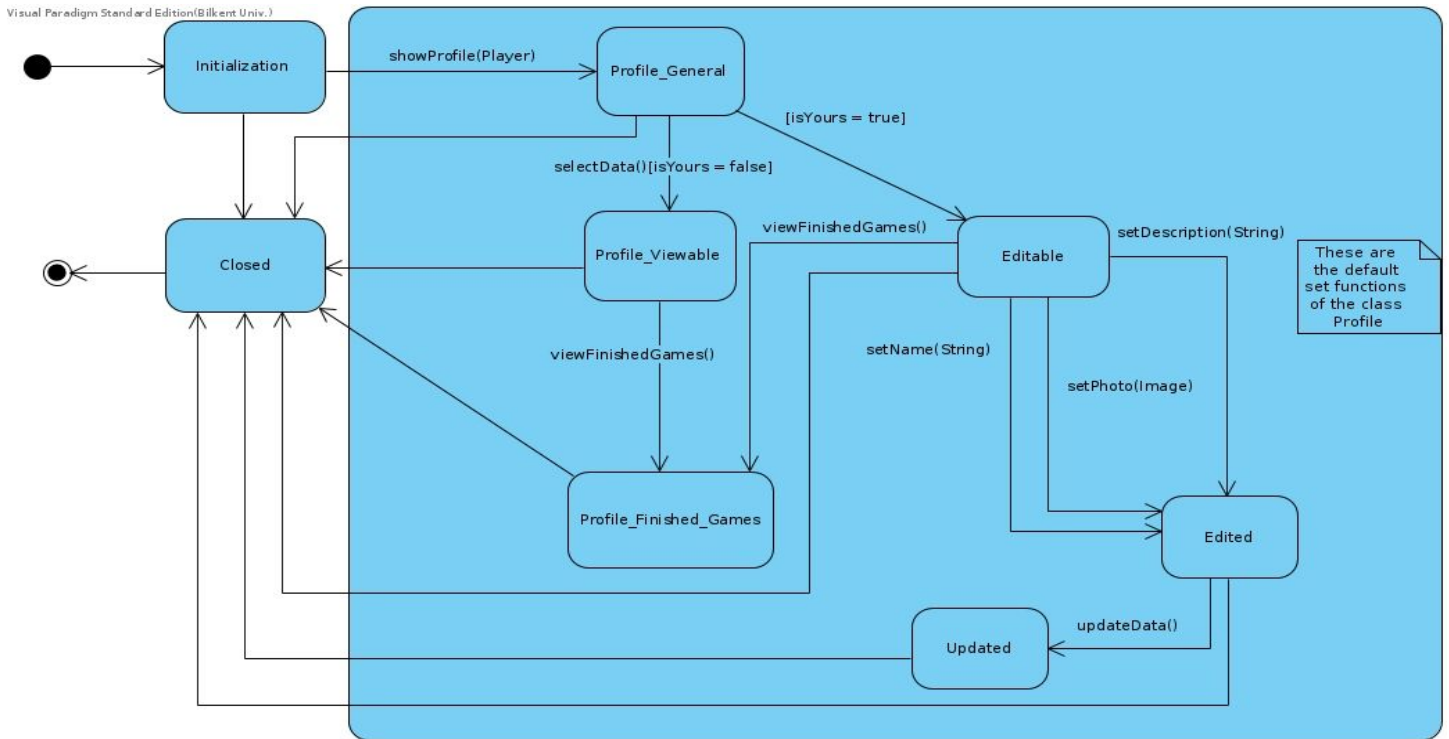


Figure 11: "Profile" state diagram of "Your Story"

Profile has different states depending on what the player wants to do. At first, the initial state is active when it is created. Then profile goes into general state after it obtains the related information of player. If this profile is player's profile, then it is "Editable" by this player. If it is not, profile is only viewable by this player. In each state, player can see the finished games of this profile in the Finished\_Games state. If the player wants to edit his information, he can do it in the "Editable" state. After he/she edits, profile enters the "edited" state, which he/she need to save the new information by updating it in the database. If this step fails, the new information will not be saved in any other state. After saving it, profile goes into "Updated" state, which display a success message and informs the user that his/her changes are complete. After this, player can exit the profile.

## 5.7. Seat State Diagram

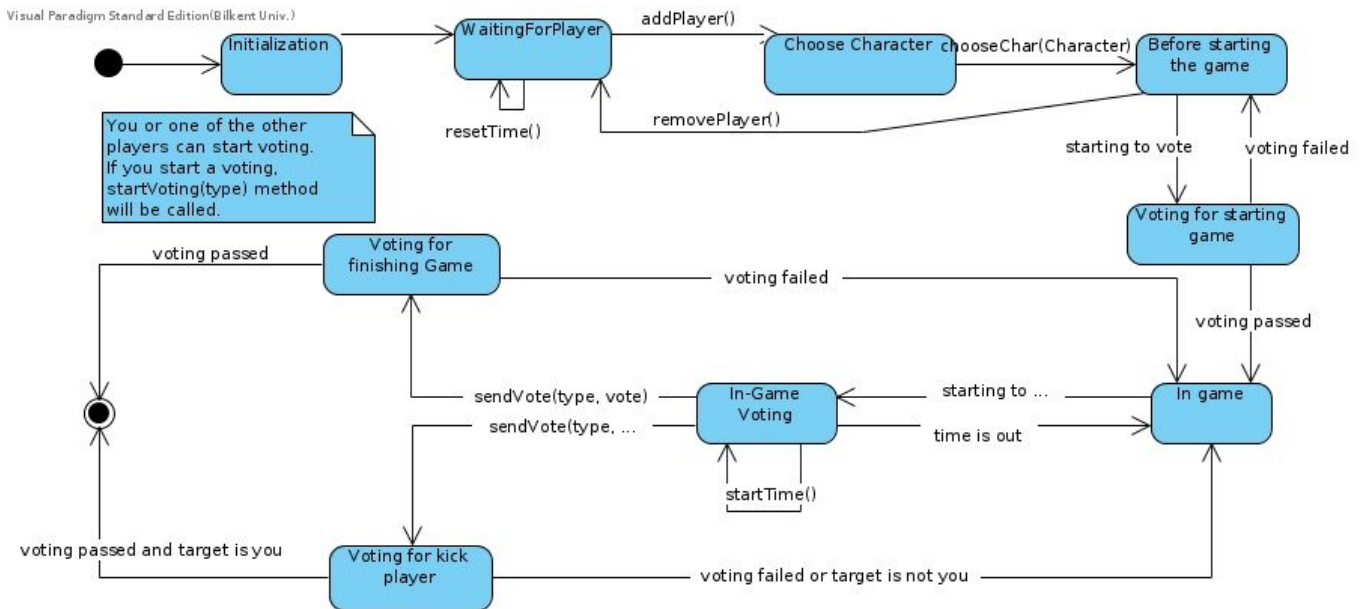


Figure 12: "Profile" state diagram of "Your Story"

The seat can be found in many different states the first state is when the Lobby is created where it is initialized. Then it is in "Choose Character" after the player is added. In between those two states is the waiting state where nothing happens it just resets the timer if the user decides to leave and waits for another user to be assigned to. Before starting the game the seat can vote to start the game. Then in-game state is where the Seat can vote to kick a Player or end the game. Seat ends when the game is finished or the Player in that seat is kicked out.

## 5.8. Activity Diagram

Visual Paradigm Standard Edition (Bilkent Univ.)

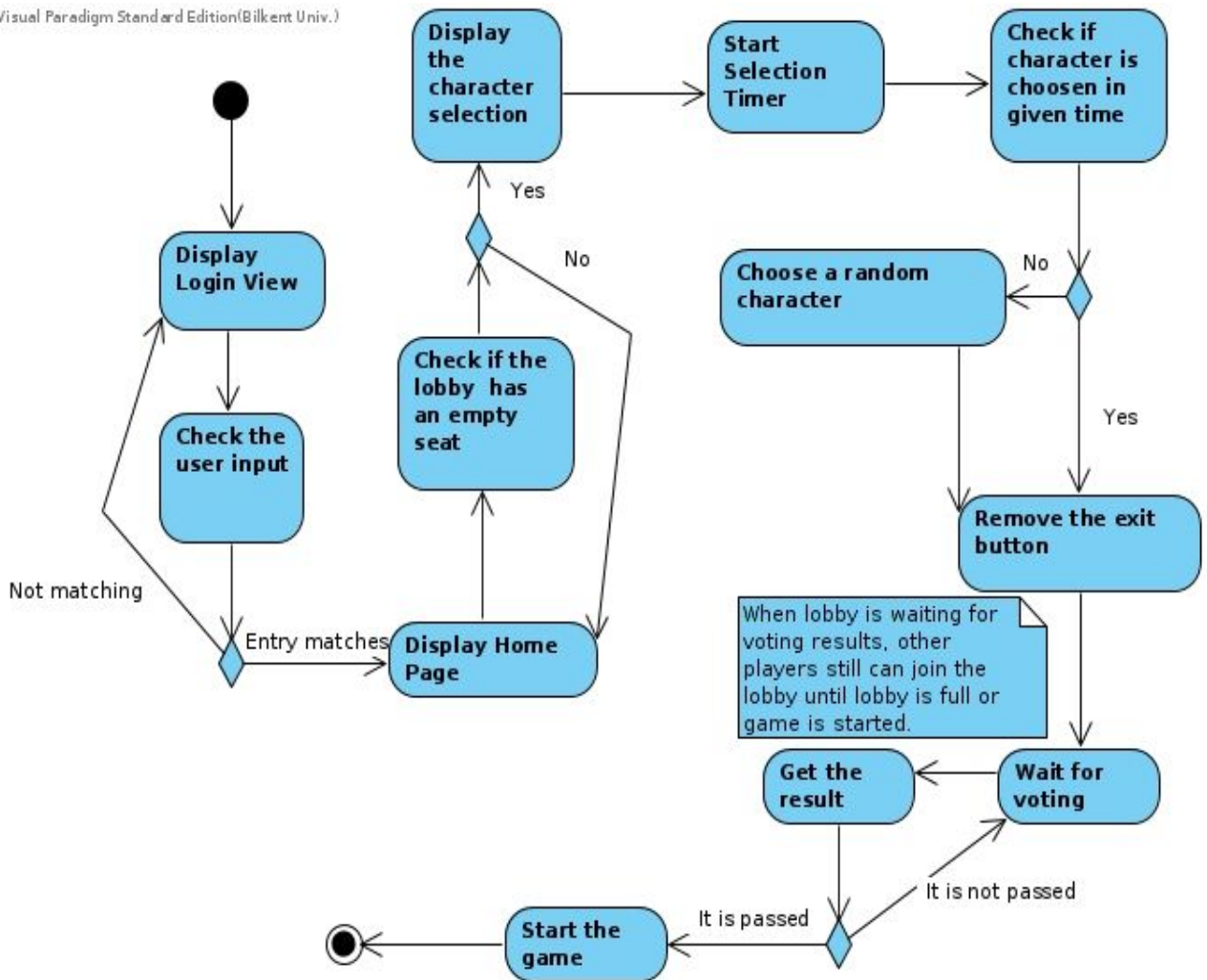


Figure 13: Activity diagram of "Your Story"

The activity diagram is displaying what a user would do in order to start playing a game. He/she logs in and joins a lobby. Then within the selection timer the user can choose any of the characters available and play or just leave. If time is up the user is assigned a random character from the available ones. The diagram describes that the user didn't leave before the time was up so he/she had 2 options. After this the user should wait for the voting to start the game to pass or as described if lobby suddenly is full the game will start ignoring the voting results.

## 6. User Interface

### 6.1. Login UI

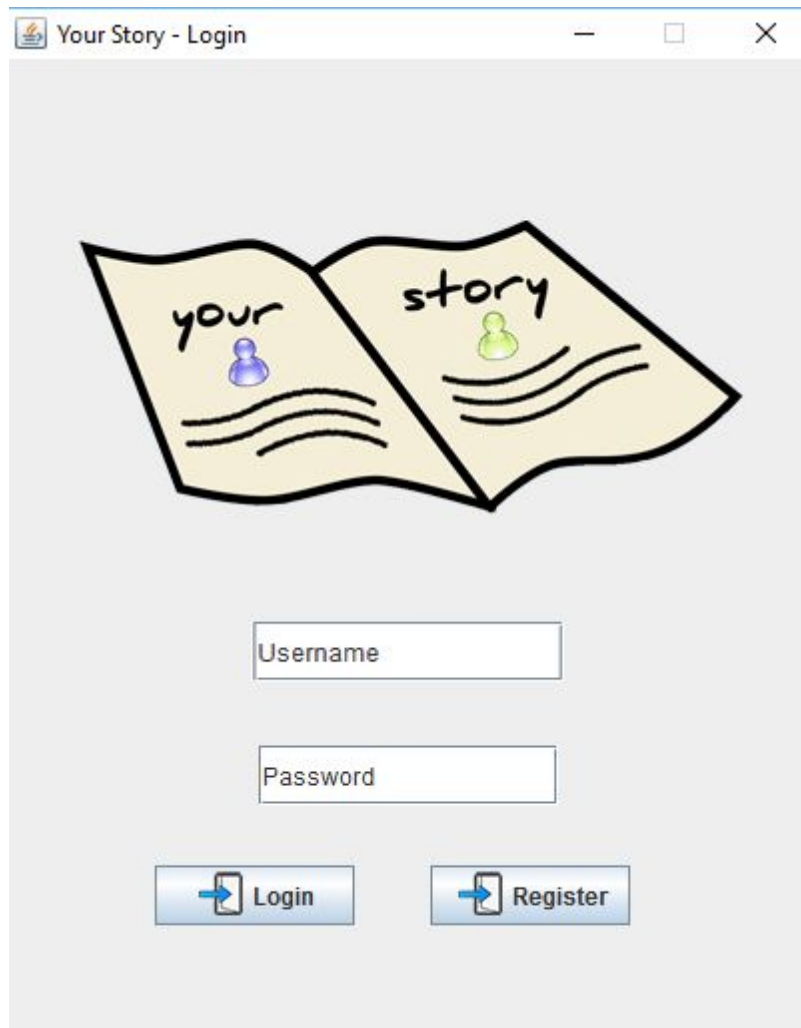


Figure 14: Login UI of "Your Story"



## 6.2. Home Page

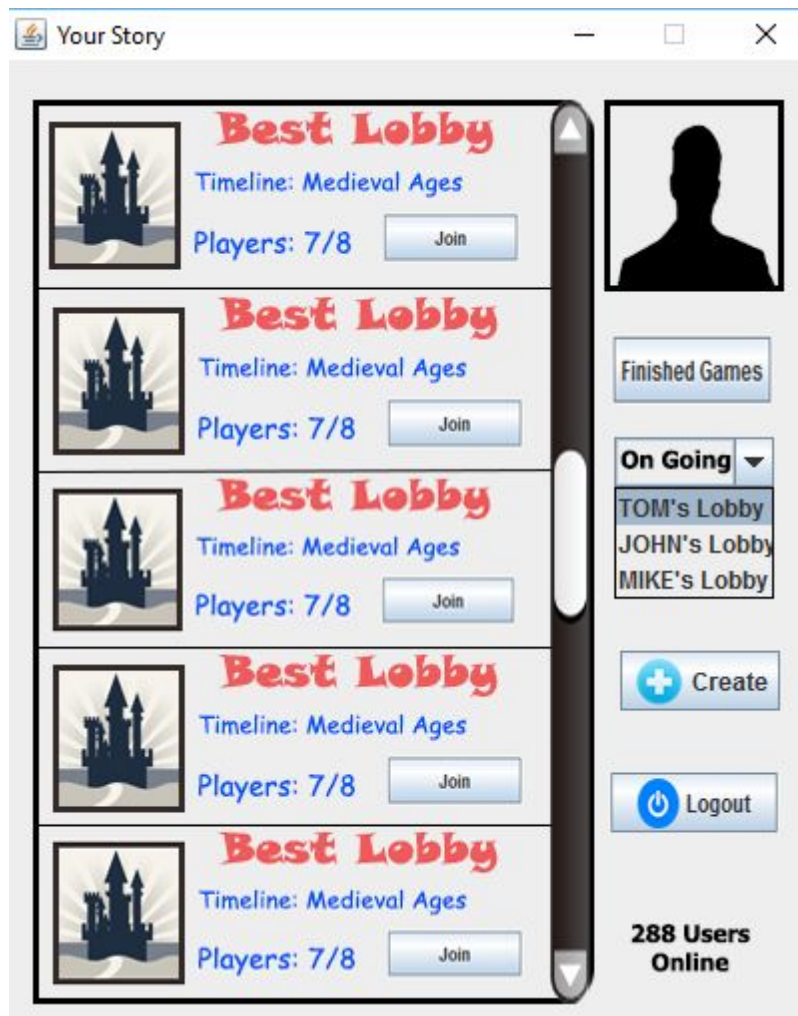


Figure 15: Home Page UI of "Your Story"

### 6.3. Lobby



Figure 16: Lobby UI of "Your Story"

## 6.4. Profile



Figure 17: Profile UI of "Your Story"

## 6.5. In-Game UI

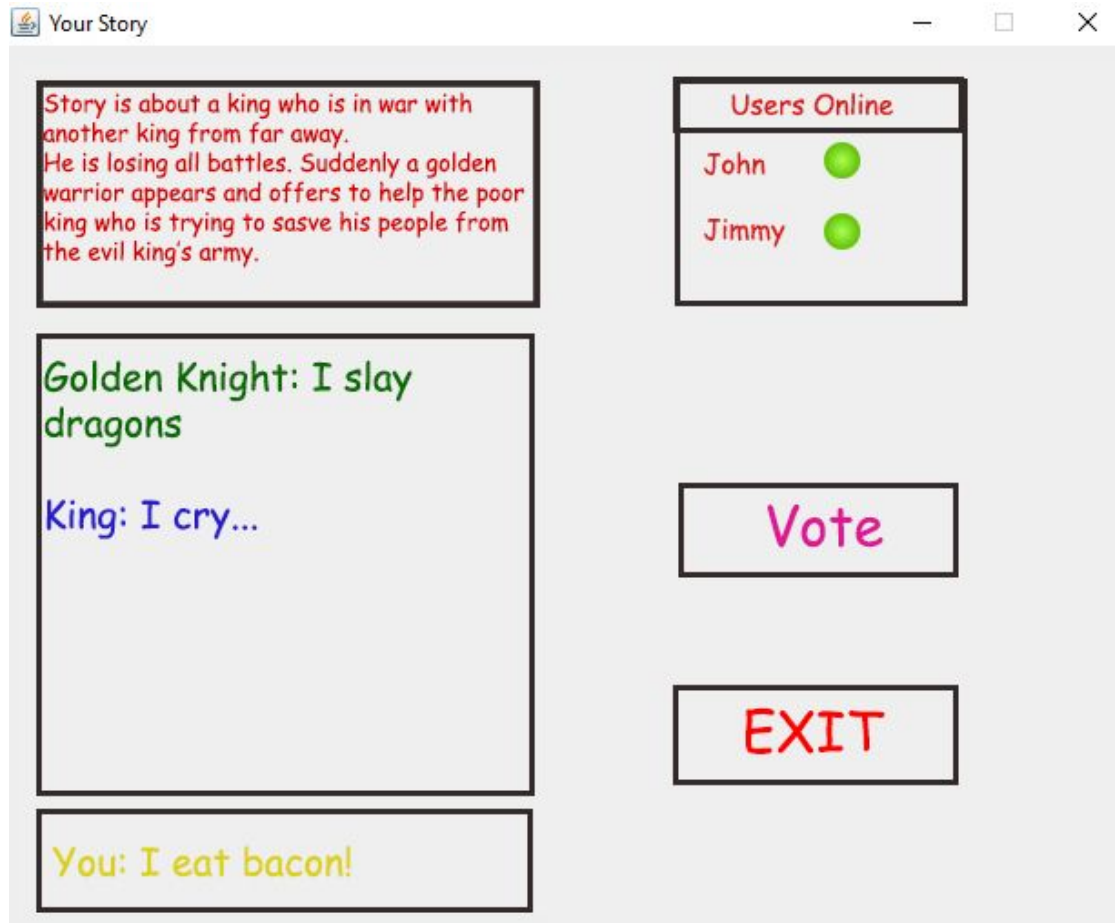


Figure 18: In-Game UI of "Your Story"

## 6.6. Lobby Creation UI



Figure 19: Lobby Creation UI of "Your Story"

## 7. References

- [1] Knight picture: <a href='http://www.freepik.com/free-vector/coloured-knight-design\_951845.htm'>Designed by Freepik</a>
- [2] King picture: <a href='http://www.freepik.com/free-vector/coloured-king-logo-design\_905188.htm'>Designed by Freepik</a>
- [3] Castle: <a href='http://www.freepik.com/free-vector/castle-silhouette-on-a-cliff-background\_685264.htm'>Designed by Freepik</a>
- [4] Profile Avatars: <a href='http://www.freepik.com/free-vector/avatar-silhouettes\_893143.htm'>Designed by Freepik</a>