OBJECT-ORIENTED SOFTWARE ENGINEERING

DEBATE IT

ANALYSIS REPORT

GROUP 1B

21301294 **YAĞIZ GANİ**

21502938 **ÇAĞATAY SEL**

21200846 **AHMET SARIGÜNEY**

21501109 **YASİN BALCANCI**

SECTION 01

17.02.2018

Table of Contents

1. Introduction

Exchange of ideas is a prevalent activity. By virtue of human nature, people usually make inherent arguments to take up any position. Thus we as a group broadly considered that such concept like bouncing the ideas off each other can be widely approved in a cyber world. It is also considered that mobile phones rather computers are suitable for this concept. Our project shaped by these considerations and emerged as an interactive mobile application based on the ground of a literal debate.

As all we know that debate is a formal discussion on a systematic pattern in which opposite ideas are asserted by two different sides. It generally ends with voting in order to expose which side is stronger in terms of advocacy and content. Basically, we aimed to transform literal debate into an entertaining and worthwhile activity. Just as in real debate events, some principles like taking sides, mounting arguments, racing against time and voting form the boundaries of "Debate it". However, we designed some complementary features that users can additionally utilize.

2. *Overview*

"Debate it" enables its users to defend their opinions about various topics such as science, health, technology etc. and each topic contains a fair amount of controversial ideas. Some major purposes of our project can be specified as:

- driving people to pondering,
- improving defending skills based on ideas,
- providing pleasant time,
- and creating a competition environment through experiences.

2.1. Gameplay

Debate episodes take place in four player rooms and they are carried out by only two players of four. A random idea is generated under current topic and initially it is asked for players to defend or oppose. At this point, there is a base case to start with generated idea that both sides have to have at least one player. Otherwise, any opposition can't be achieved in order to debate and a new idea comes till the base case has reached. Then system randomly selects two players from each side. The targeted gameplay algorithm doesn't let any players to be inactive more than two consecutive debates.

A battle starts with an introductory stage that the selected two players mount the first arguments. Both arguments are introduced in the presence of all players. Following that, an opposition stage [1] comes and counter-arguments to the first arguments are mounted. Then both players answer to the counter-arguments at third stage. Lastly, an epilog stage takes place in order to conclude the debate. It has also arguments same as the previous stages. After all, each performed stage within its arguments is displayed on screen. Players have only one and a half minutes to complete the stages, however, if both players finish before the time is up, stage is automatically finished.

3. Requirement Specification

3.1. Functional Requirements

3.1.1 Topic Categorization

The debate application contains different types of topics for arguments and for votes of the users. These topics will be health, economy, philosophy, history and education. Arguments or statements under these topics will be projected through the user of the application.
Also, each topic will indicate 10 ideas or subtopics. The ideas will be presented to the users and each user will choose sides whether they agree or don't agree with the idea. The number of arguments or ideas can be increased by the time according to the size of the players.

3.1.2 Lobby Browser

Each player will choose a lobby first. The application will show the current rooms to the players. Also the application will give the current number of the players on the room, in order to inform them. The players will determine which room to enter according to that information. If the room is empty, there is no meaning to enter and wait for so long so the player should choose a lobby with 3 players and only 1 player remaining to begin the game. When 4 players enter a lobby the game begins.

3.1.3 Battle Arena with 4 Players

Each game will indicate a battle arena composed of 4 players.

3.1.4 Choose sides

Votes play an essential role in this game. So choosing sides is crucial in order the game to begin. If there is only 1 opinion about the given idea, then another idea will come because the debate can't begin if everyone thinks the same thing. For example; the idea is shown on the screen and everyone agreed with the idea. The debate won't start and another idea will be shown on the screen. On the next idea there are people who agrees and at least 1 disagreed person. The debate will begin then.

3.1.5 Game course with 4 stages

There are 4 stages in this game. These are opening, counter, answer and final stages. Detailed information about the stages and action is given on the overview part.

3.1.6 Experience Gain & Unlock Customizable Items Ability

Each player will gain experience points after each battle and they will be able to unlock some items or gifts with these xp' s. (xp = experience point) These items can be advanced avatars, titles, etc. The players will have the ability to see the finished battles and they can still vote the ideas came before. So the players of the past matches can still gain xp' s even if the battle is over. There will be a pop-up window showing the top 10 battles or something near 10. So the players who have just entered the application will be able to see the results of the battles and they will be able to see the statistics of these battles. Also, they are able to vote the players or ideas they like on the shown battle.

3.1.7 Ability to send expression during battle

The players are capable of sending some expressions like smiling face or upset face etc. during the gameplay.

3.1.8 Avatar

Each user profile will indicate an avatar. Some of the avatars will be free to use but some of them will be locked. A player must gain enough xp's to unlock an advanced avatar. There will be a market option on the main menu. The user is able to unlock items from there.

3.1.9 Battle Screen

When the players enter a lobby and the lobby gets full, the game screen will open and there will be a green or some colored frame. Around the frame, there will be 4 players' avatars shown. The idea will be projected on the middle of the frame and there will be voting icons which will express agree and disagree.

3.2. Non-Functional Requirements

3.2.1 Network

The game will use a network connection to provide the users to play each other, using Internet.

3.2.2 Operating System

Since the system will be a mobile application, it will be supported on the devices which has Android operating system. The system requirements will be as following:

- Android Operating System Software versions.
- An Intel Pentium 4 processor or later
- 1 Mbps or better Internet connection

3.2.3 Database

A database is required in order to store the players' personal data and required to store game records, statistics, results, avatars, experience points, etc.

3.2.4 Music

There can be music on the background of the game in order to make the game more attractive for the users.

3.2.5 Background Theme

For different types of topics, there will be different types of background images during the battles. For example; if the idea category is philosophy, the background image will turn to ancient Greek theme. If the idea is within the health category, a medicine image will be shown on the backhand.

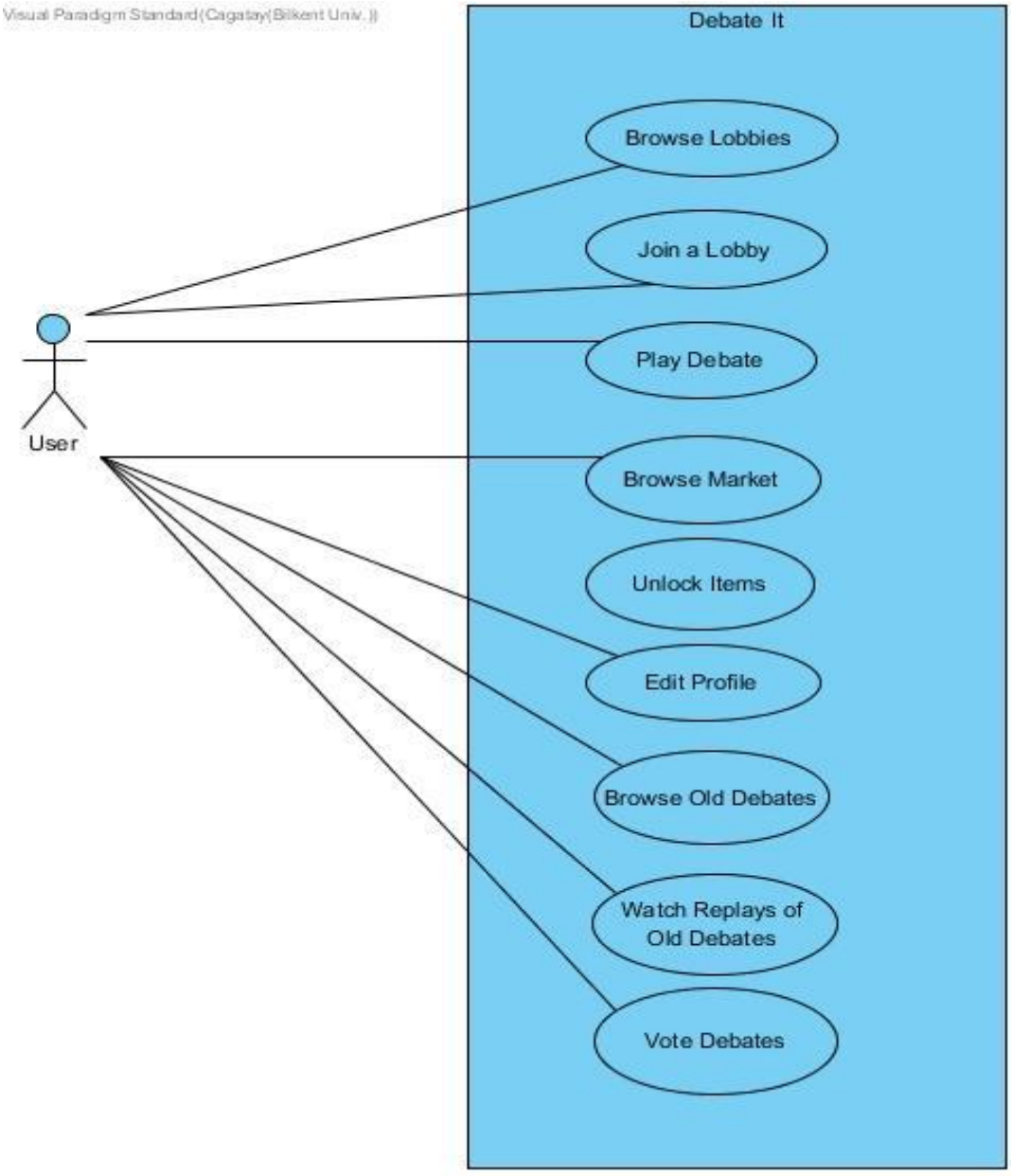
3.2.6 Performance

Since the game will use network connection, there might be some unexpected problems like performance issues, bugs, freezing, etc. in the application. The project members will try to minimize these possible issues by considering this possibility during the coding stage of the project. Also most of the mobile devices are able to handle the “DebateIt” application in terms of hardware. Most Android devices have the appropriate processors and CPU’s for these mobile games.

4. System Model

Models were drawn based on the guidelines in the course book[2].

4.1. Use Case Model



4.1.1 Browse Lobbies

Participating Actors: *User*

Entry Condition: *User selects "Play Game" from the menu*

Exit Condition:

-User selects a lobby or,

-User exits this section.

Main Flow of Events:

-The system displays available lobbies.

4.1.2 Join a Lobby

Participating Actors: *User*

Entry Condition: *User is in the "Play Game" section.*

Exit Condition: *User chooses a lobby.*

Main Flow of Events:

-User selects a lobby from the displayed ones.

4.1.3 Play Debate

Participating Actors: *User*

Entry Condition: *User selects a lobby.*

Exit Condition:

-User finishes the debate or

-User exits the game

Main Flow of Events:

-The idea is displayed by the system.

-User says "Yes" or "No" according to the idea.

-One user that says "Yes" and one that says "No" are chosen by the system

-Each chosen user expresses 4 arguments in periods of 1 minute.

-At the end of the debate, 4 users in that lobby choose "Yes" or "No" to the idea that system displayed according to the arguments expressed during debate.

-Users get back to the main menu.

4.1.4 Browse Market

Participating Actors: *User*

Entry Condition: *User selects "Market" from the menu*

Exit Condition: *User chooses some other section from the menu.*

Main Flow of Events:

-Items that are locked and unlocked are displayed.

-User chooses some other section from the menu.

4.1.5 Unlock Items

Participating Actors: *User*

Entry Condition: *User is in the "Market" section*

Exit Condition: -

Main Flow of Events:

-User selects an item that he has enough EXP to unlock.

4.1.6 Edit Profile

Participating Actors: *User*

Entry Condition: *User is in the "Profile" section.*

Exit Condition: -

Main Flow of Events:

-User taps the "Change" button that is on his avatar.

-User chooses a new avatar.

4.1.7 Browse Old Debates

Participating Actors: *User*

Entry Condition: *User selects "Browse" from the menu*

Exit Condition: *User exits this section.*

Main Flow of Events:

-User choses "Browse" in the menu,

-User selects a finished debate from the displayed ones,

-User votes "Yes" or "No" for the idea displayed, according to the arguments expressed in the debate.

-User goes to main menu.

4.1.8 Watch Replays of Debates

Participating Actors: *User*

Entry Condition: *User is in the "Browse" section.*

Exit Condition: -

Main Flow of Events:

-User selects a finished debate from the displayed ones and selects "Replay",

-The system shows the arguments in the debate one by one as it is live,

-User votes "Yes" or "No" for the idea displayed, according to the arguments expressed in the debate.

4.1.9 Vote Debates

Participating Actors: *User*

Entry Condition: *User is in the "Browse Old Debates" section.*

Exit Condition: -

Main Flow of Events:

-User selects a finished debate from the displayed ones and selects "View",

-User votes "Yes" or "No" for the idea displayed, according to the arguments expressed in the debate.

4.2. Dynamic Models

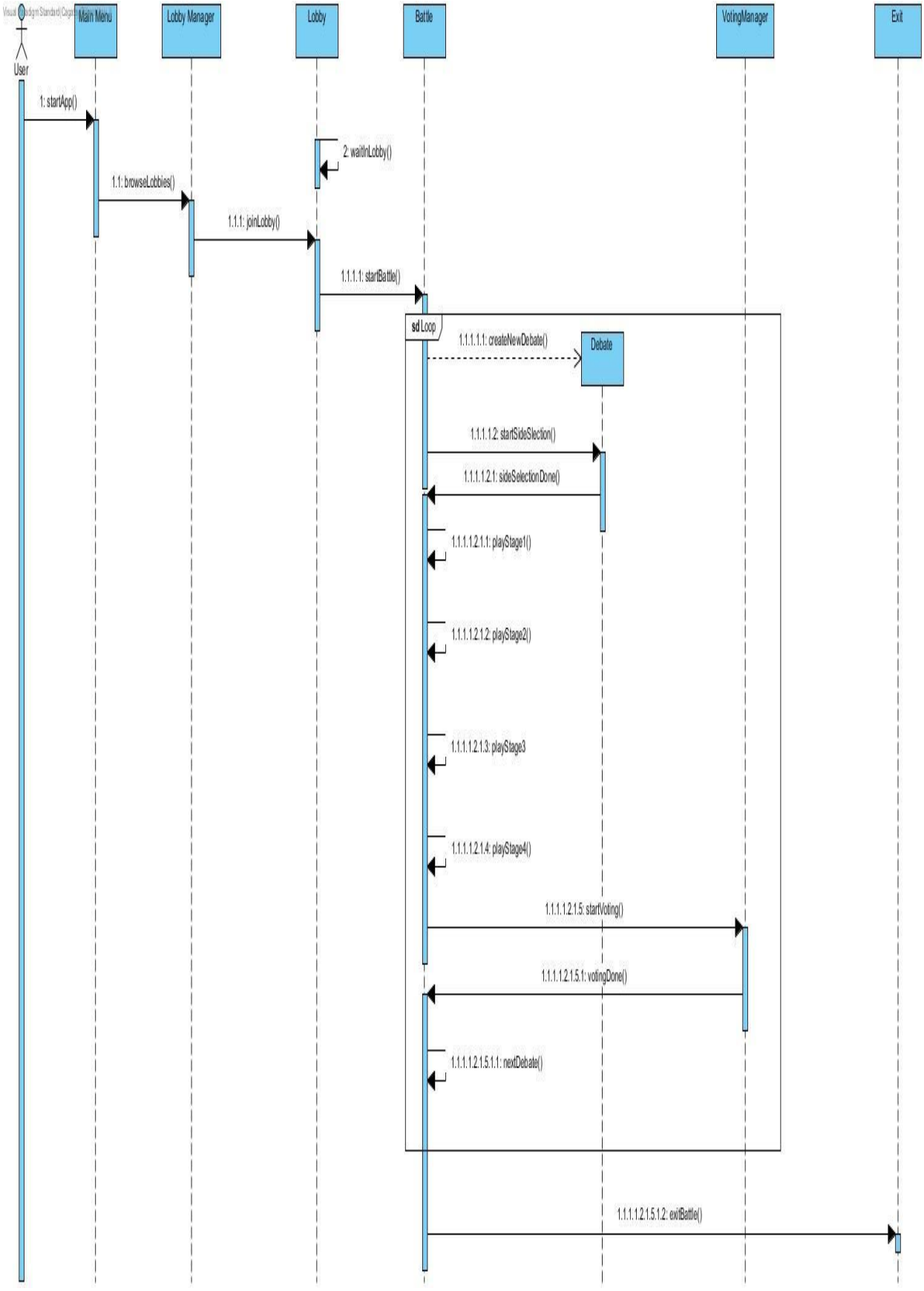
This section includes to dynamic models which represents certain scenarios that user may experience while using Debate It.

4.2.1. Sequence Diagrams

4.2.1.1. Playing A Debate

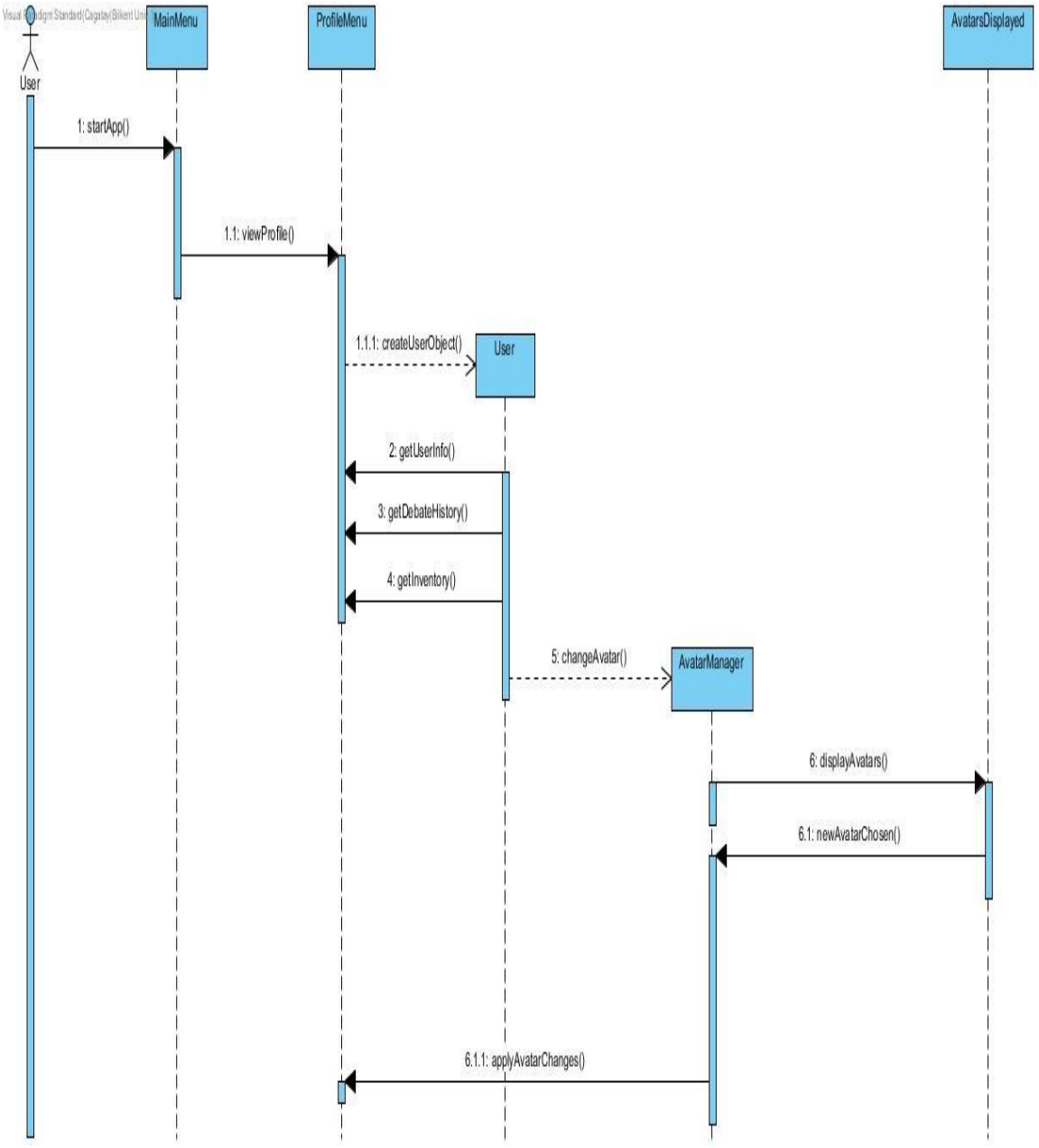
Scenario:

User wants to play our game. So the user taps on the browse lobbies option in the main menu. This action initializes the LobbyManager which lists the possible lobbies that user can join. User can select a lobby and select join lobby option to join this lobby. Users will wait in lobby until the necessary number of players joined lobby. When lobby has enough players, it starts the battle. Players will be given an idea which they can either agree or disagree. Based on players' choices 2 players will be selected for the debate. Following the end of debate, players vote for the winner. When the voting is finished, battle continues with another debate until players exit the battle.


User
Main Menu
Lobby Manager
Lobby
Battle
VotingManager
Exit
1: startApp()
2: waitInLobby()
1.1: browseLobbies()
1.1.1: joinLobby()
1.1.1.1: startBattle()
sd Loop
1.1.1.1.1: createNewDebate()
Debate
1.1.1.1.2: startSideSlection()
1.1.1.1.2.1: sideSelectionDone()
1.1.1.1.2.1.1: playStage1()
1.1.1.1.2.1.2: playStage2()
1.1.1.1.2.1.3: playStage3
1.1.1.1.2.1.4: playStage4()
1.1.1.1.2.1.5: startVoting()
1.1.1.1.2.1.5.1: votingDone()
1.1.1.1.2.1.5.1.1: nextDebate()
1.1.1.1.2.1.5.1.2: exitBattle()

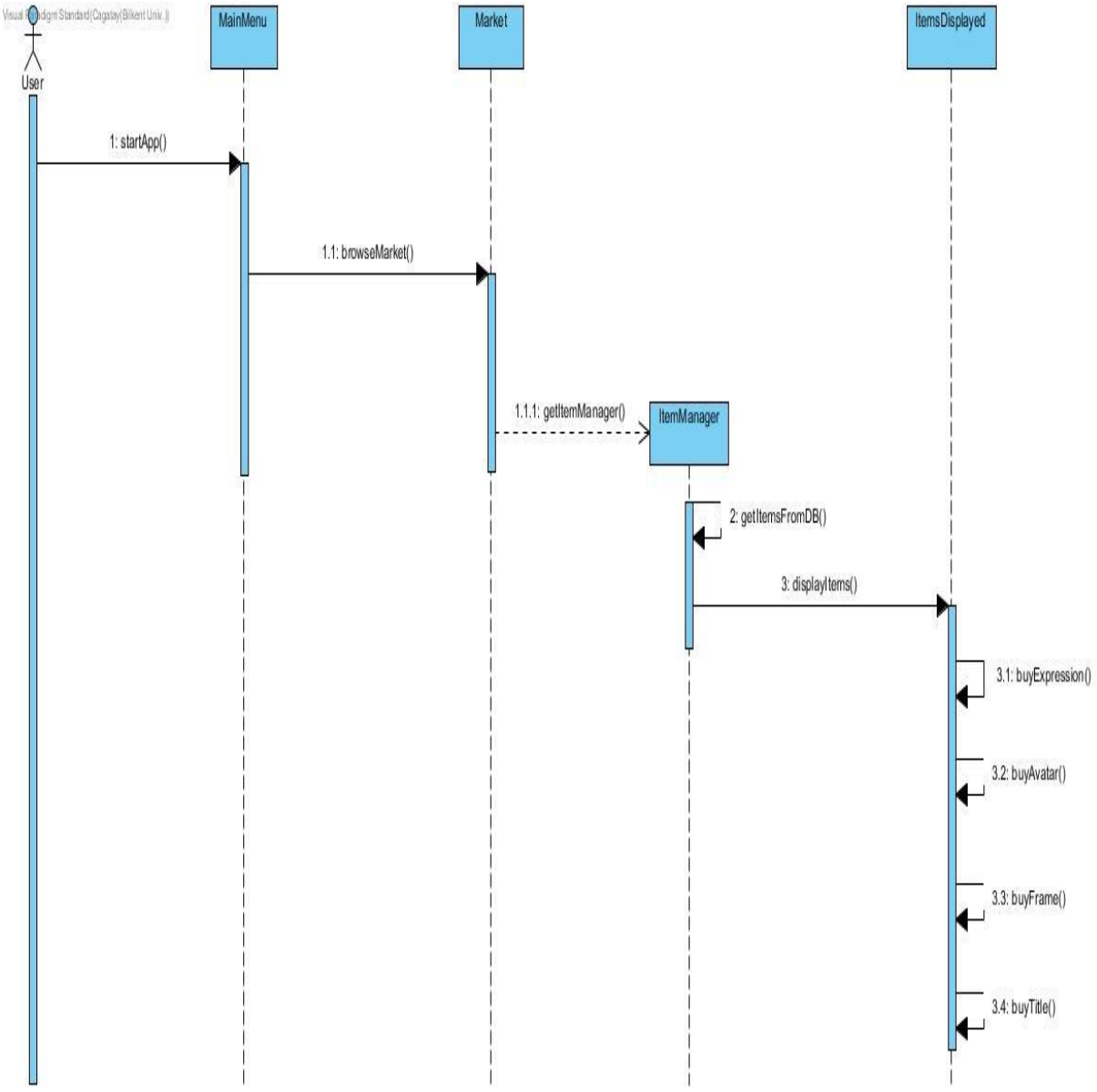
4.2.1.2 View Profile

Scenario: User wants to see his/her profile page where he/she can see nickname, avatar, XP, inventory and the past debates. He/she taps on view profile option in the main menu. A user objects gets created based on the id of the user. This object interacts with database and gets the user information. User can choose to change his/her avatar. Avatar manager gets the avatars that the user has and displays them on the screen. User can choose a new avatar or cancel the process and go back to profile page.

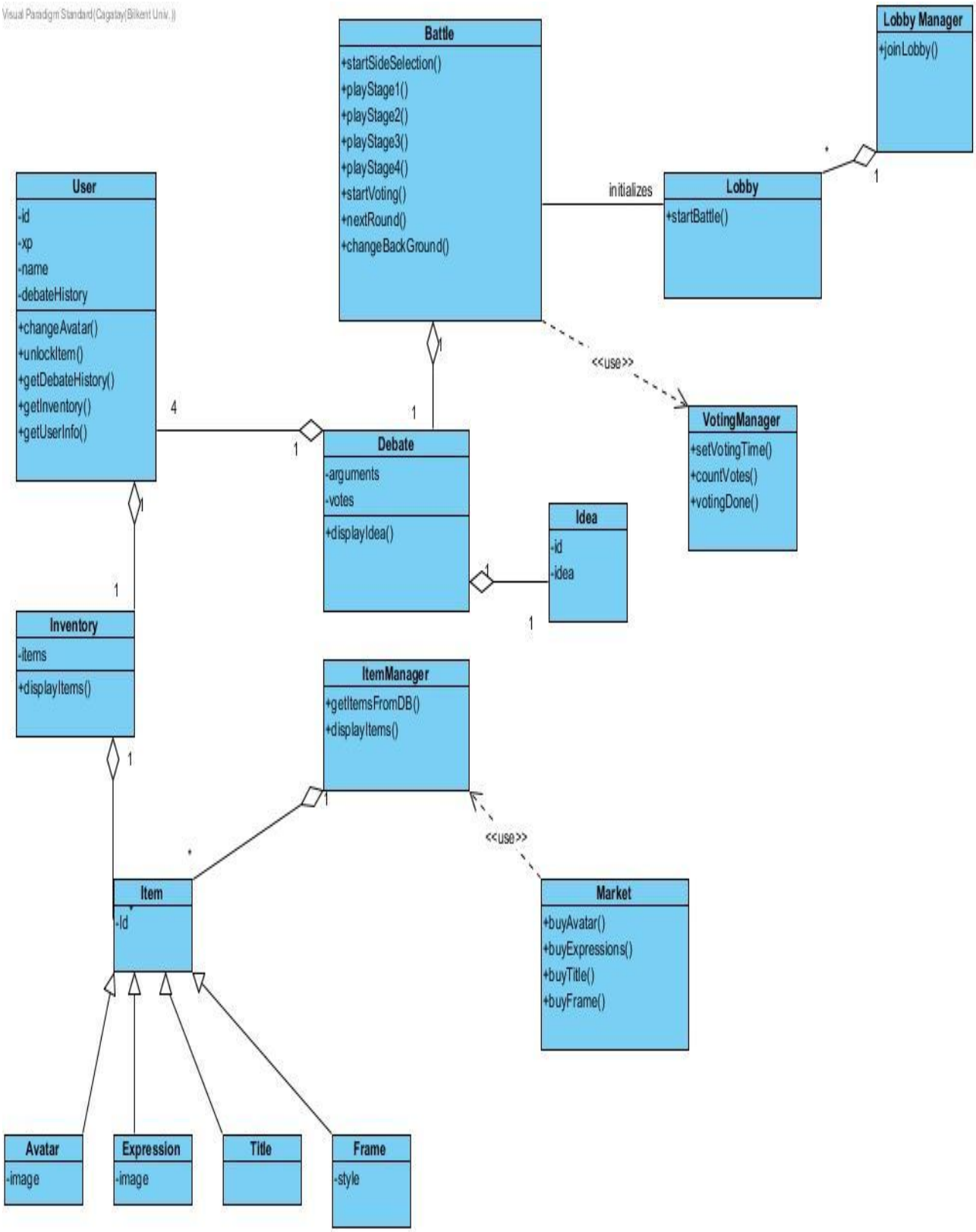


4.2.1.3 Browse Market

Scenario: User wants to browse market where he/she can unlock various items based on his XP level. These items are cosmetic items which changes the appearance of the user during debates. User navigates to market from main menu. An item manager gets created. Item manager gets the items that are available for user from the database. User can choose to unlock new expressions, avatars, frames or titles in this menu.



4.3. Object and Class Model



The above figure shows the initial structure of the classes that is planned for the first iteration.

Battle class will handle the dynamic structure of the game. It will be responsible for handling the stage transition. It will use the VotingManager for handling the voting at the end of each debate.

Debate class will hold the static structure of the game which includes the user information, arguments, idea, and the result of voting. This class will be used to store and restore old debates.

User class holds the user related information.

Inventory class handles the items of a user.

Item class is a parent class for the unlockable items.

Avatar, Expression, Title, Frame classes represents the unlockable items.

ItemManager class is responsible for retrieving the items from database.

Market class is used to enable user to unlock new items.

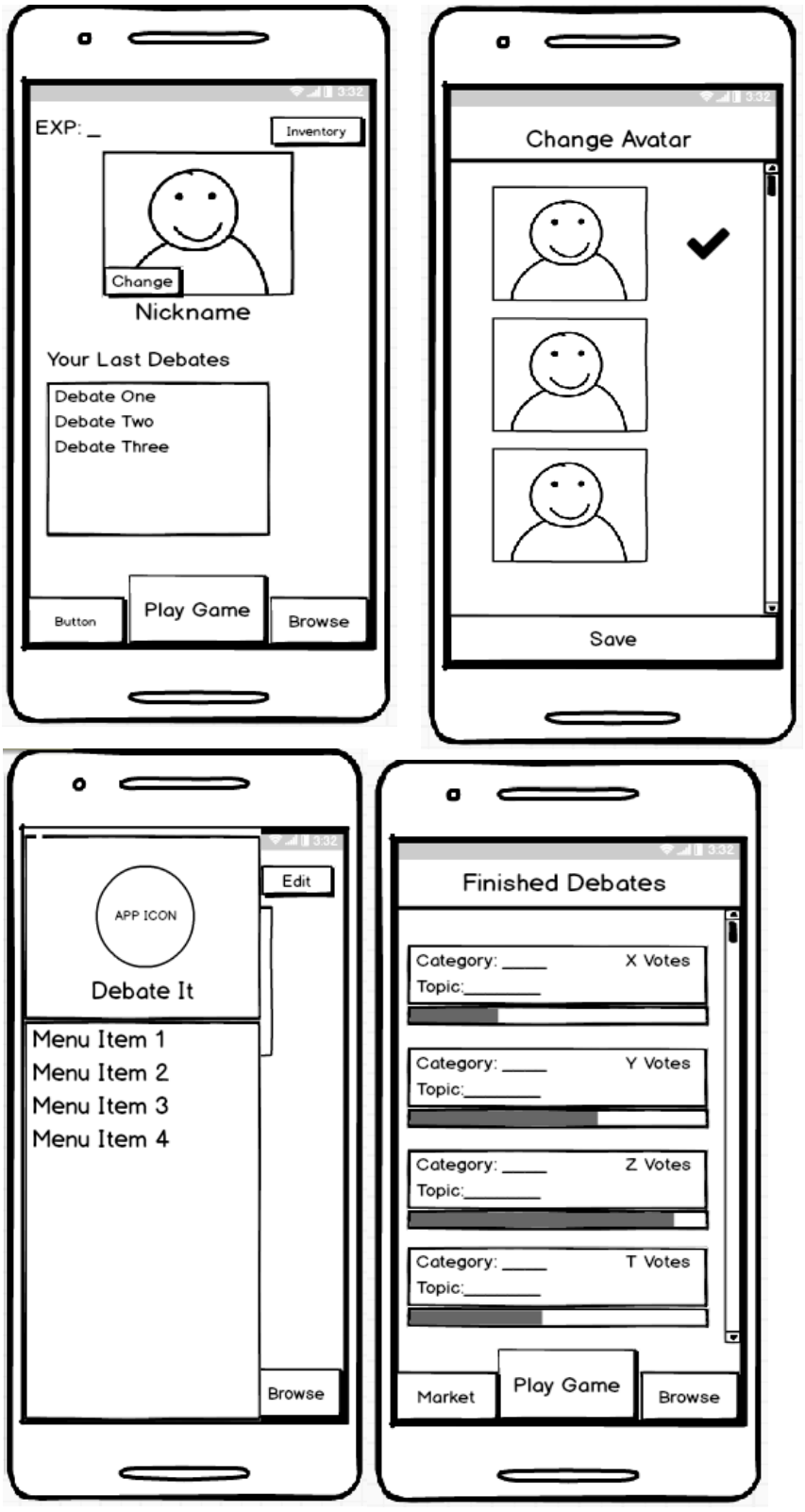
Idea class represents the controversial idea which will be provided to each debate.

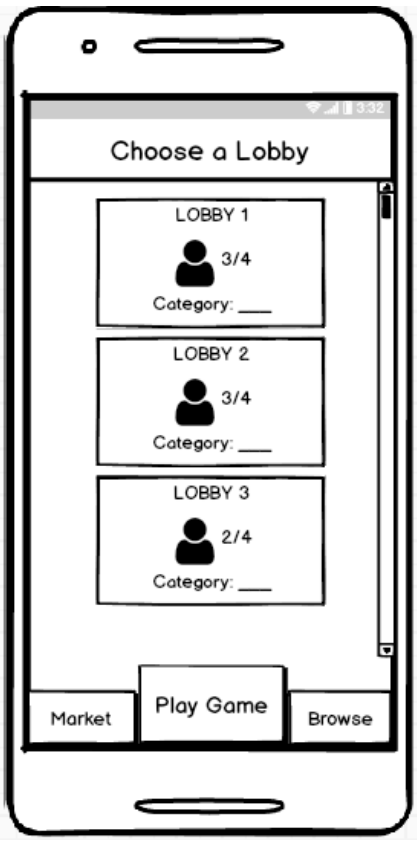
VotingManager class handles the voting periods.

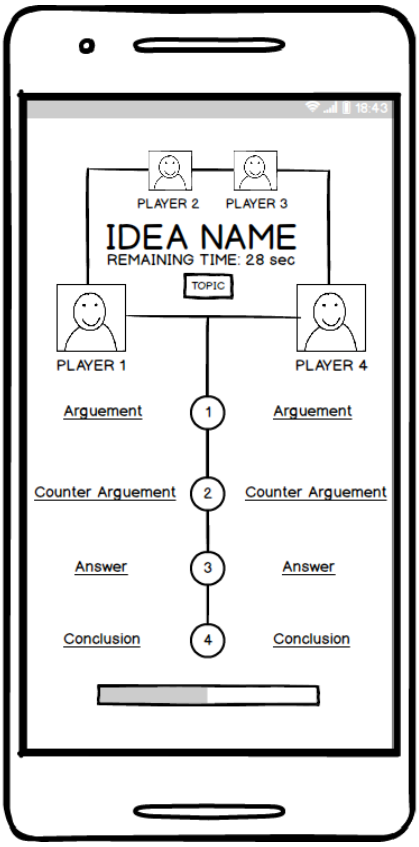
LobbyManager handles the available lobbies which user can select.

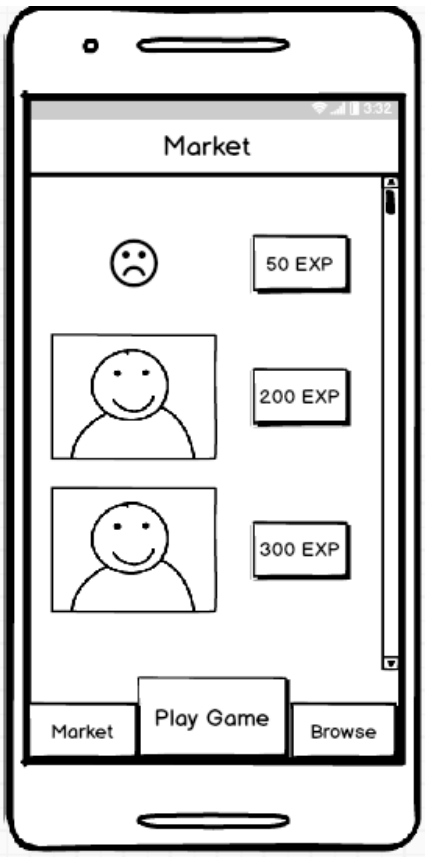
Lobby class represent the lobby which users will join to start battle.

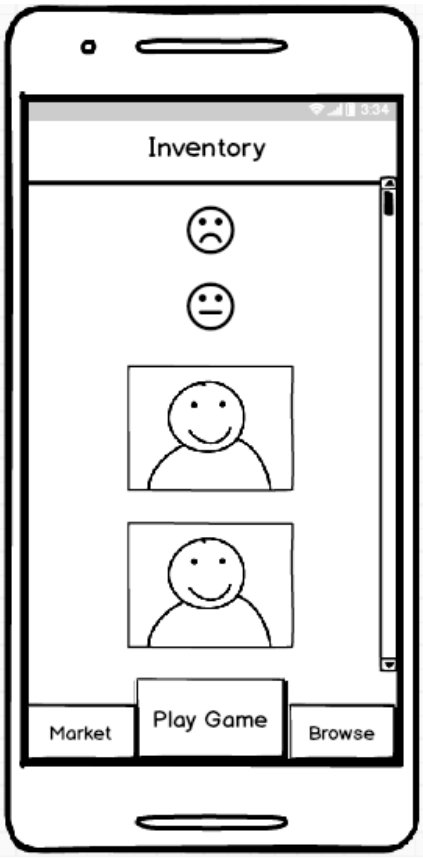
4.4. Mockups


EXP: _
Inventory
Change
Nickname
Your Last Debates
Debate One
Debate Two
Debate Three
Button
Play Game
Browse
Change Avatar
Save
APP ICON
Edit
Debate It
Menu Item 1
Menu Item 2
Menu Item 3
Menu Item 4
Browse
Finished Debates
Category: ____ X Votes
Topic:_____
Category: ____ Y Votes
Topic:_____
Category: ____ Z Votes
Topic:_____
Category: ____ T Votes
Topic:_____
Market
Play Game
Browse


Choose a Lobby
LOBBY 1
3/4
Category: ___
LOBBY 2
3/4
Category: ___
LOBBY 3
2/4
Category: ___
Market
Play Game
Browse


PLAYER 2
PLAYER 3
IDEA NAME
REMAINING TIME: 28 sec
TOPIC
PLAYER 1
PLAYER 4
Arguement
1
Arguement
Counter Arguement
2
Counter Arguement
Answer
3
Answer
Conclusion
4
Conclusion


Market
50 EXP
200 EXP
300 EXP
Market
Play Game
Browse


Inventory
Market
Play Game
Browse

5. Glossary

Idea: A controversial statement which will divide users to 2 sides.

Avatar: A small image which will be shown in the debate screen with the nickname of user.

Frame: A border that will be around user's nickname.

Title: A title which will be shown before user's nickname in his profile and debate screen.

Expression: A certain expression such as "WOW" or "GG" which users will be able to use during debates.

6. References

[1]"The Premier Online Debate Website | Debate.org", Debate.org, 2018. [Online]. Available: http://www.debate.org/. [Accessed: 17- Feb- 2018].

[2]B. Bruegge and A. Dutoit, *Object-oriented software engineering*. Harlow, Essex: Pearson, 2014.