OBJECT-ORIENTED SOFTWARE ENGINEERING

DEBATE IT

ANALYSIS REPORT

GROUP 1B

21301294 **YAĞIZ GANİ**

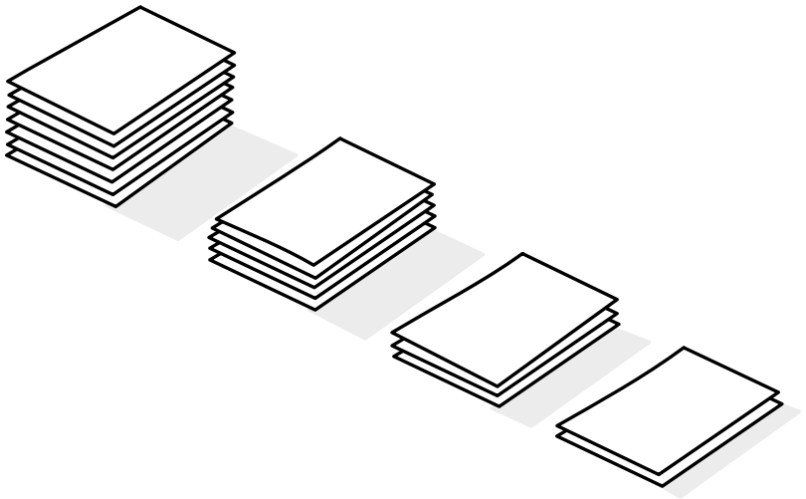
21502938 **ÇAĞATAY SEL**

21200846 **AHMET SARIGÜNEY**

21501109 **YASİN BALCANCI**

SECTION 01

03.04.2018

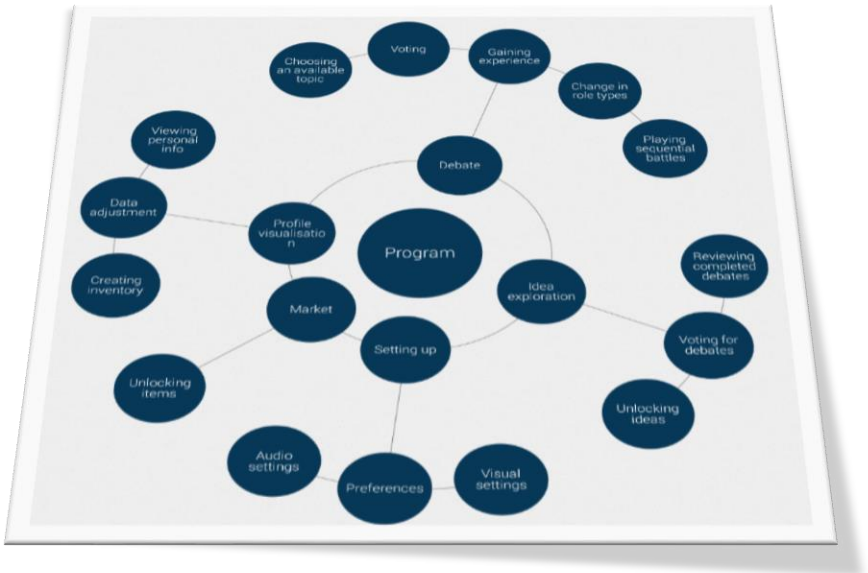


INTRODUCTION

- **Overview**

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.



- **Gameplay**

In order to play a debate, user browses current debate lobbies and joins an available lobby. When the maximum number of players, which is 4, in a lobby is reached, a random idea will be generated by the server and the debate will start. After the generation of the idea, players will be asked to choose a side as an opposing or supporting side of the idea. When side selection ends, if there is at least one opposing and supporting player debate will start and one player from the each side will be selected as debaters and the rest will be spectators. After a successful side selection, battle will start. If all players choose opposing or supporting side, a new idea will be generated until there is at least one opposing and supporting side of the idea.

A battle starts with an introductory stage that the selected two players mount their initial arguments with respect to given idea. Debaters will type their arguments and send it to other players. Arguments send by a player will be displayed as a text in other players' phones. After the first stage, an opposition stage comes and counter-arguments to the first arguments are mounted. Then both players answer to the counter-arguments that were mounted by their opponent at the answers stage. Lastly, an epilog stage takes place in order to conclude the debate. Players present their last and concluding arguments in this stage. After the conclusion stage, players who were selected as spectators vote the debaters. After the voting stage, battle will generate a new idea and the debates will go on.

Players will have limited amount of time for each stage. However, if both players finish before the time is up, next stage will start immediately. If one of the debaters leave during before the conclusion stage is finished, debate will be closed and the remaining debater will be treated as he/she got 2 votes from the spectators. If a spectator leaves during battle, debate will continue and other users will be able to join ongoing debates as spectator.

REQUIREMENTS SPECIFICATION

In this section, functional and non-functional requirements of our system will be introduced. While determining these requirements, we tried to cover all possible user actions. The functional requirements aim to enrich user experience by providing more functionality to application. Also, we tried to identify non-functional requirements to provide multiplayer game experience to its users.

- **Functional requirements**

  - Topic categorization

    Ideas that are being generated by system will gather under 5 different categories. These categories are health, economy, philosophy, history and education. These categories are fixed.

  - Lobby browser

    There will be multiple lobbies and debates being played at the same time in which users can choose to join.

  - Choosing side

    User will be able to choose either opposing side or supporting side of an idea. Choosing a side is crucial to determine the opposing and supporting debaters of an idea so that a debate can start.

  - Game course with four stages

    There will be four stages in debates. These are opening, counter, answer and final stages. Detailed information about the stages and action is given on the overview part.

  - Earning points

    Users will earn points after each battle and as their past debates keep getting votes. User will use these points in the market.

- Ability to purchase customizable items

  Users will also be able to purchase some items or gifts with their points. These items can be new avatars, titles, frames or expressions.

- Sending expressions during battle

  Players will have the opportunity to send expressions like smiling face or upset face etc. during battles in order to express their feelings in a entertaining way.

- Avatar selection

  Each user profile will have an avatar. Some of the avatars will be free to use but some of them will be locked. A player must purchase new avatars in order to use them. There will be market option on the main menu. The user is able to purchase items from there.

- Battle screen

  When players enter the lobby and the lobby gets full, the game screen will open with a thematic background depending on the category. User's selected avatar, tile and frame will be displayed in this screen.

- Viewing past debates

  Users will be able to view past debates that were played by other users or debates that they have played. User will also be able to vote the debaters of past debates.

- Replaying past debates

  Users are able to replay finished debate according to the order that the original debate was played.

- Register & Login

  Users are required to have an account and be logged in with it in order to access the main page of the application and perform activities. That is why they are able to login to their pre-registered accounts with their usernames and passwords.

- Inventory display & item selection

  As the users purchase new items from market, their items will be stored in their inventory. Users will be able to choose which items they want to use from their inventory.

- Sound effects

  There will be sound effects through user actions in order to create more attractiveness.

- 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 background.

- **Non-functional requirements**

  - Network

    The application will need network connection to server. The delay caused by the connection between clients and server should not be more than 1 second.

  - Response time of server

    Server will respond to requests coming from client in less than 10 seconds.

- Operating system

Since the system will be a mobile application, it will be supported on the devices which has Android operating system KitKat or higher.

- Data storage

A MySQL will be deployed on a machine that server is connected to. This machine will store user information's and past debates. Database should be able to respond to server requests in less than 0.5 seconds.

- Multiple client connections

Our server machine will enable at least four players to be connected at the same time.

- UI and graphics

Default views are supplied by Google, inc to all android developers will be used to construct UI. These image files will be taken from google pictures will be royalty-free.

- Battery consumption

Battery consumption will support at least one hour usage on %70 of android devices which have put on market after 2018 January 1st

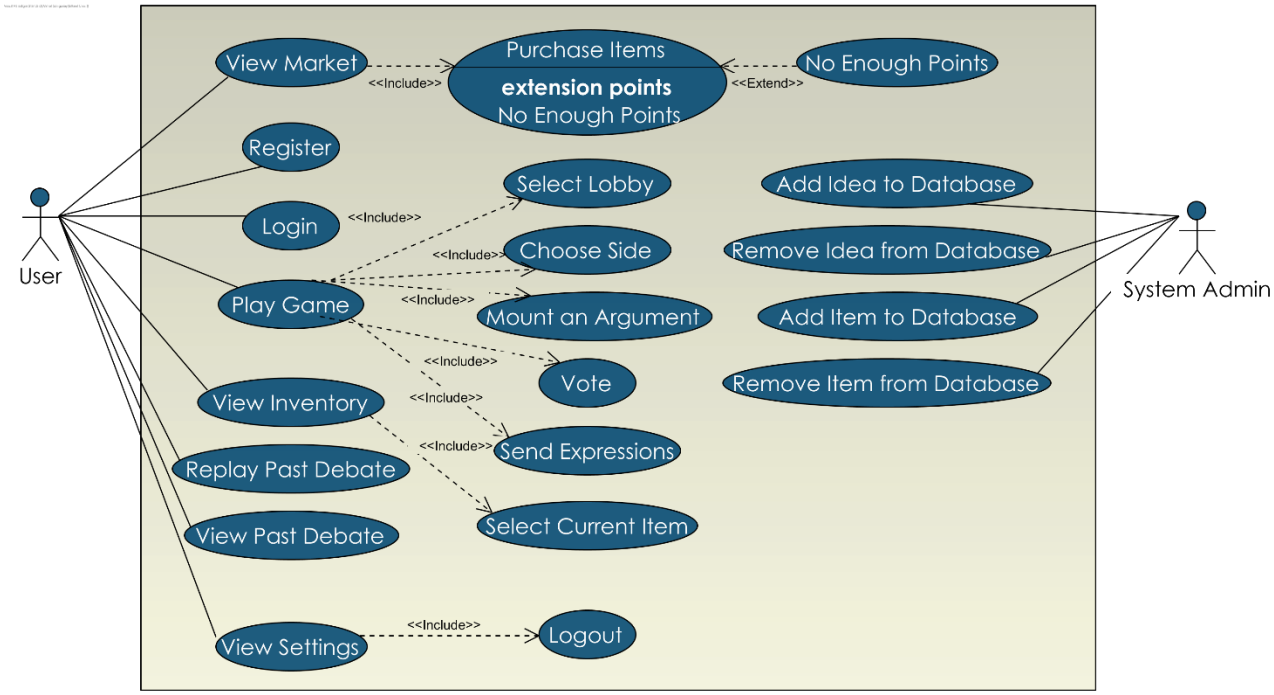
- Idea variety

There will be five different categories and ten different ideas under each of the category.

SYSTEM MODEL

- **Use case model**

Under modelling stage, we firstly asked some questions with "what" in order to determine the use cases of "Debate it". As it is seen that two different characters as "user" and "system admin" can utilize these specific use cases.



➢ View 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.

- Purchase 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 points to purchase.

- Register

  Participating actors: User

  Entry condition: -

  Exit condition: Registering in with a valid and not pre-registered account.

  Main flow of events:

  → User types a non pre-registered username, password, confirm them and taps to "Register" button.

- Login

  Participating actors: User

  Entry condition: -

  Exit condition: Logging in with a valid account

  Main flow of events:

  → User types the pre-registered username and password and taps to "Login" button.

- Play game

Participating actors: User

Entry condition: -

Exit condition: User finishes the debate or exits the existing debate battle.

Main flow of events:

→ User selects a lobby. Game starts when the number of players is four.

→ The idea is displayed by the system.

→ User says "yes" or "no" according to idea.

→ One user that says "yes" and one that says "no" are choosen by the system

→ Each choosen user express four arguments in periods of debate stages.

→ At the end of the debate, four users in that battle choose "yes or "no" for the current idea what system displayed according to the arguments expressed during debate.

→ Users get back to the main menu.

- Select lobby

Participating actors: User

Entry condition: User selects "Play game" from the menu.

Exit condition: User selects a lobby or exits this section.

Main flow of events:

→ The system displays available lobbies.

- Choose side

  Participating actors: User

  Entry condition: User is in the game.

  Exit condition: -

  Main flow of events:

  → User says "yes or "no" according to the displayed idea.

- Mount an argument

  Participating actors: User

  Entry condition: User is in the "Play game" section and one of the debating players.

  Exit condition: The time to send argument is up.

  Main flow of events:

  → User types and sends an argument in the time he is allowed to.

- Vote

  Participating actors: User

  Entry condition: User is in the "Play game" section and the debate is over.

  Exit condition: -

  Main flow of events:

  → User votes "yes" or "no" for the idea displayed, according to the arguments expressed in the debate.

- Send expression

  Participating actors: User

  Entry condition: User is in the debate battle.

  Exit condition: -

  Main flow of events:

  → User chooses an expression to send

- Edit profile

  Participating actors: User

  Entry condition: -

  Exit condition: User chooses some other section from the menu.

  Main flow of events:

  → User chooses an item that he wants it to be the new current Title, Frame or Avatar.

- View inventory

  Participating actors: User

  Entry condition: User is in the "Profile" section.

  Exit condition: Choosing another section from the menu.

  Main flow of events:

  → User taps the "Inventory" button.

  → The owned items are displayed.

  → User goes back.

- Replay past debate

  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.

- View past debates

  Participating actors: User

  Entry condition: User selects "Browse" from the menu.

  Exit condition: User exits this section.

  Main flow of events:

  →User chooses "Browse" in the menu.

  → User selects a finished debate from the displayed ones.

  → User votes these debates as "yes" or "no" for the idea displayed, according to the arguments expressed in the debate.

- View settings

  Participating actors: User

  Entry condition: User selects "Settings" from the menu.

  Exit condition: User chooses some other section from the menu.

  Main flow of events:

  → Remember me, logout and sound on/off settings are displayed.

  → User can change settings or logout.

  → User exits this section by choosing another one if it is not logged out.

- Logout

  Participating actors: User

  Entry condition: User is in the "Settings" section.

  Exit condition: -

  Main flow of events:

  → User taps "Logout" button and goes back to "Login screen.

  according to the arguments expressed in the debate.

- Add idea to database

  Participating actors: System admin

  Entry condition: -

  Exit condition: -

  Main flow of events:

  → System admin adds a new idea to database that will be used in the games later.

- Remove idea from database

  Participating actors: System admin

  Entry condition: -

  Exit condition: -

  Main flow of events:

  → System admin removes an idea from database.

- Add item to database

  Participating actors: System admin

  Entry condition: -

  Exit condition: -

  Main flow of events:

  → System admin adds a new item to database that will be used in the games later.

- Remove item from database

  Participating actors: System admin

  Entry condition: -

  Exit condition: -

  Main flow of events:

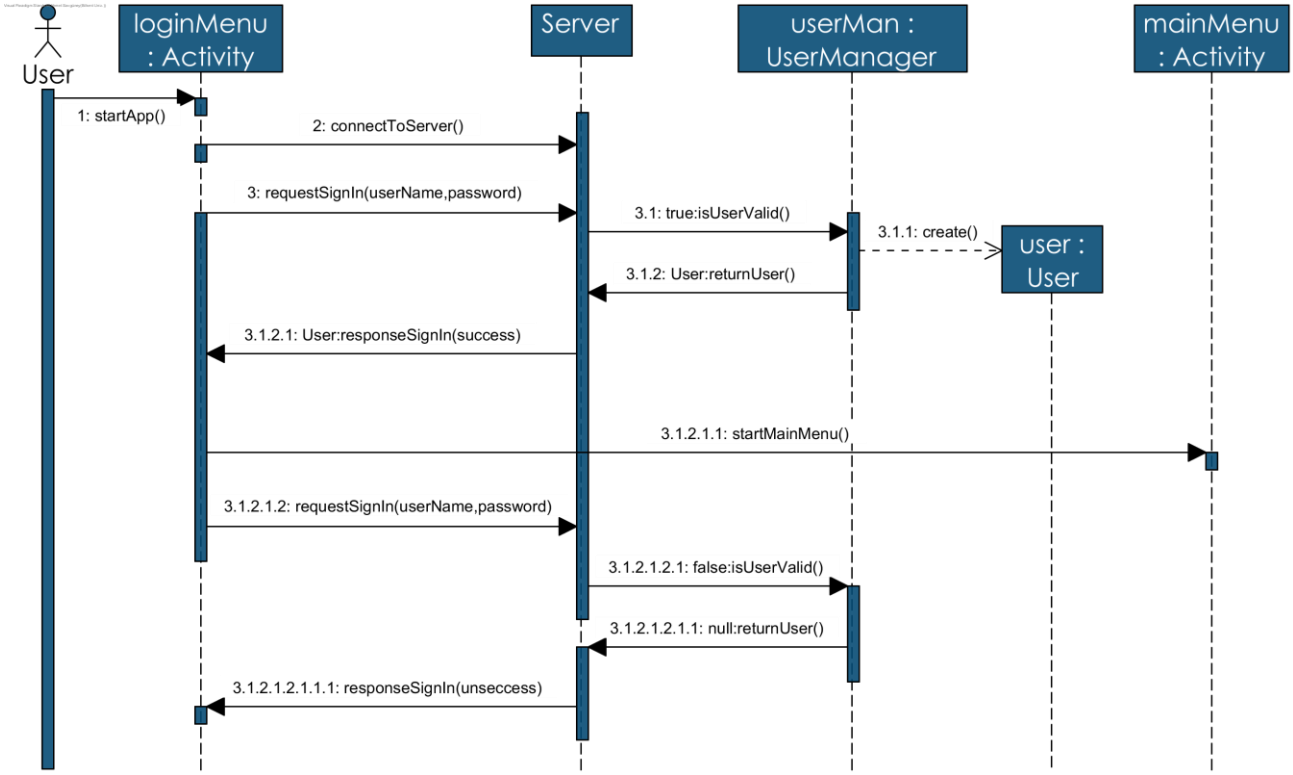
  → System admin removes an item from database.

- **Dynamic models**

This section includes the dynmaic models of "Debate it". It represents some certain scenarios that users may experience.

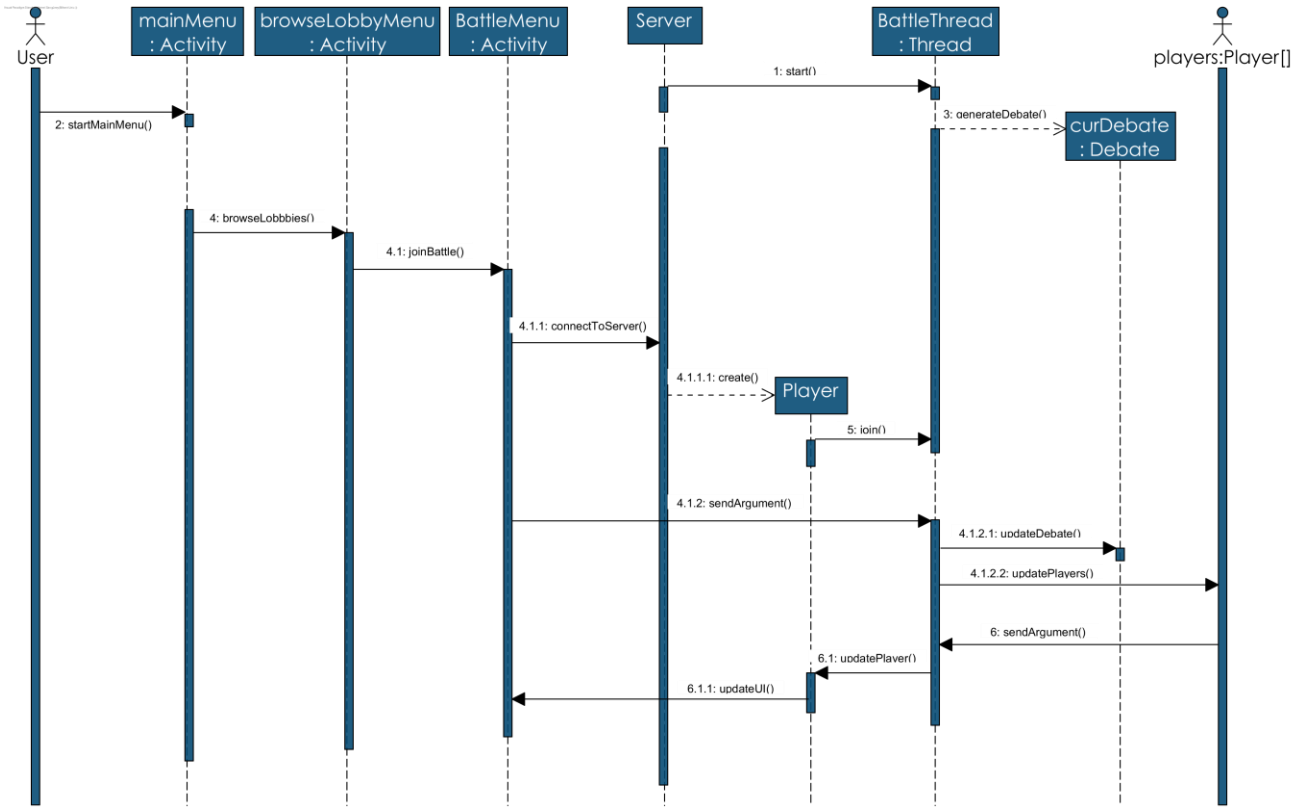
➢ Activity diagram of login activity

In every application launch, it is expected from users to login existing accounts. A "remember password" option will be hold in order to make shorten the login session.



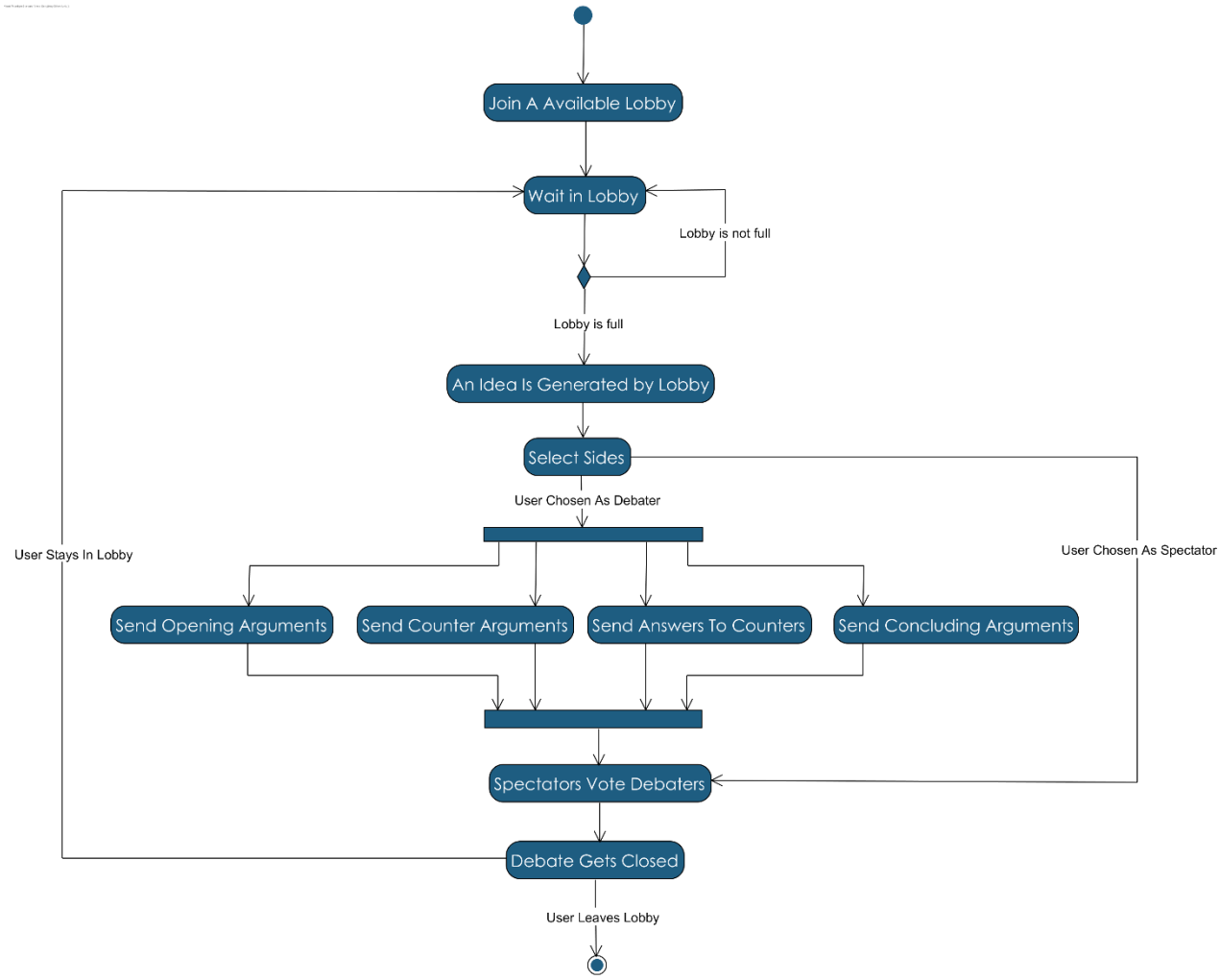
➢ Sequence diagram of game play

At first, users tap on the browse lobbies button in main menu. This action initializes the LobbyManager which lists the possible lobbies that user can join. Users can also select and join a lobby from the possible available rooms. When any room has enough players to play a debate battle, the game initializes.

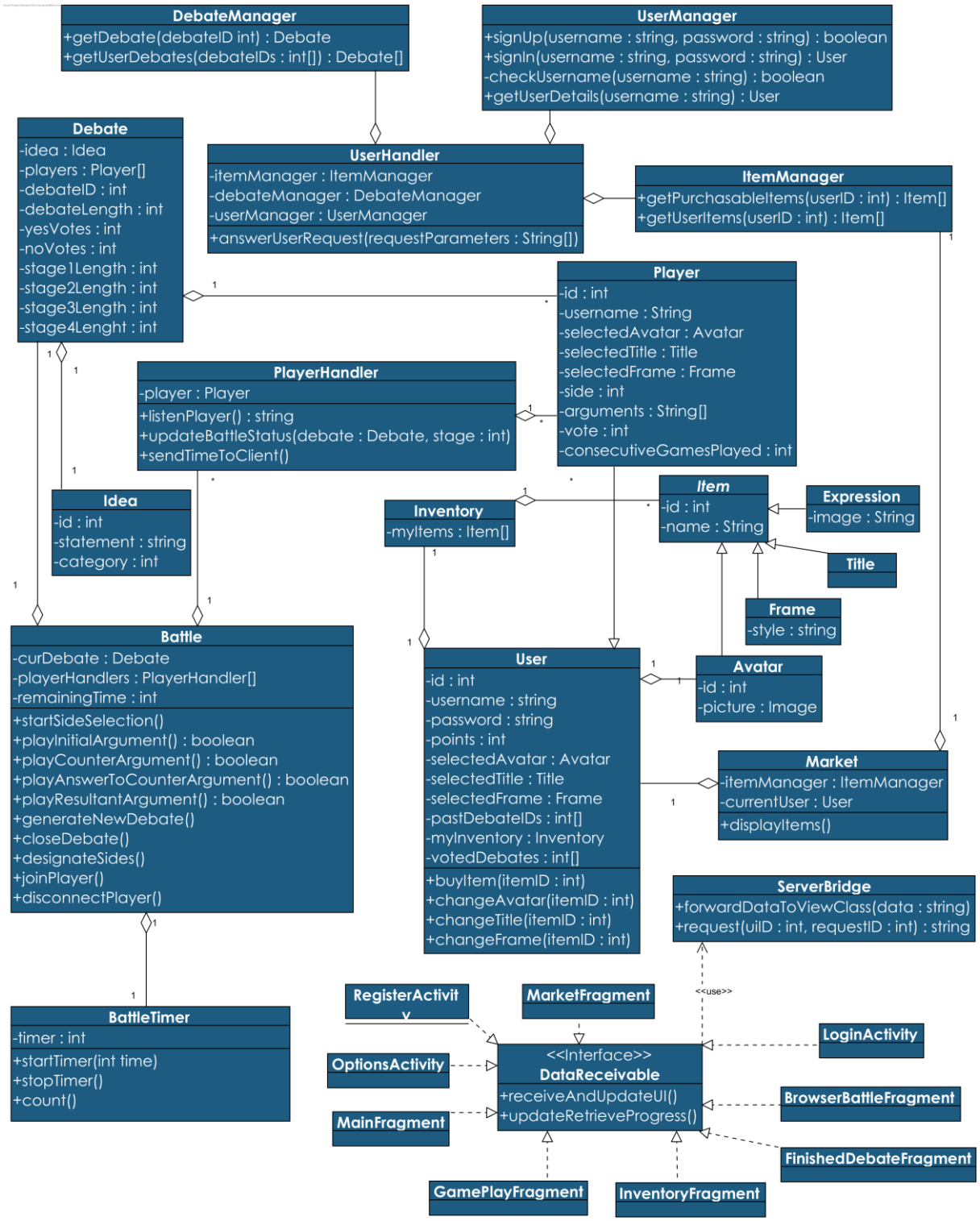


➢ Activity diagram of game play

Below model displays the flow of events when a user decides to play a debate.



o Class diagram



- **Battle**

  Battle class handles the dynamic structure of the game. It is responsible for the course of events in debate battles.

- **BattleTimer**

  BattleTimer class serves to manage time issues of battle as is evident from it's name.

- **Debate**

  Debate class holds the static structure of the game which includes the user information, arguments, idea and the result of voting. This class is used in order to store and restore past debates.

- **User**

  User class holds the user related information.

- **UserHandler**

  ItemManager class is responsible for retrieving the items from database.

- **Player**

  Player class is a different part of user class. It is used in order to represent the users as a player in debate battles.

- **PlayerHandler**

  ItemManager class is responsible for retrieving the items from database.

- **Idea**

  Idea class represents the controversy for debates. It is provided for each debates at the begining of battles.

- **Inventory**

  This class handles the items of a user.

- **Item**

  Item class is a parent class of purchasable entities.

- **Avatar, expression, title and frame**

  These classes represent the purchasable entities.

- **Market**

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

- **ItemManager**

  ItemManager class is used in order to perform data operations such as store and retrieve to existing items through database management.

- **DebateManager**

  DebateManager class is responsible for the data operations to debates.

- **UserManager**

  UserManager class performs user related operations through the database system.

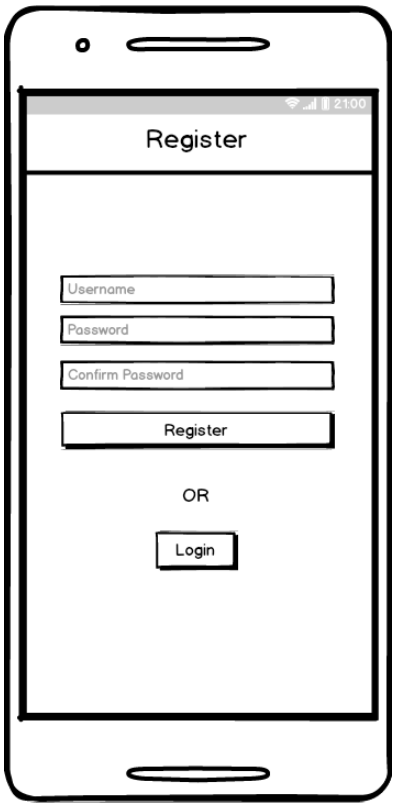
- **ServerBridge**

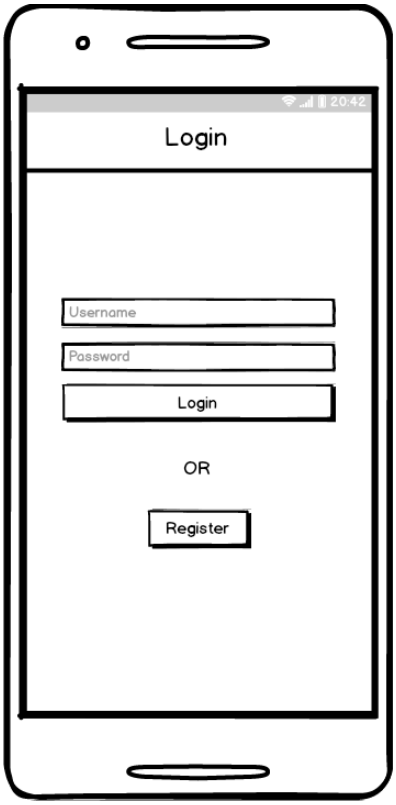
  This class act as a bridge between server and its clients.

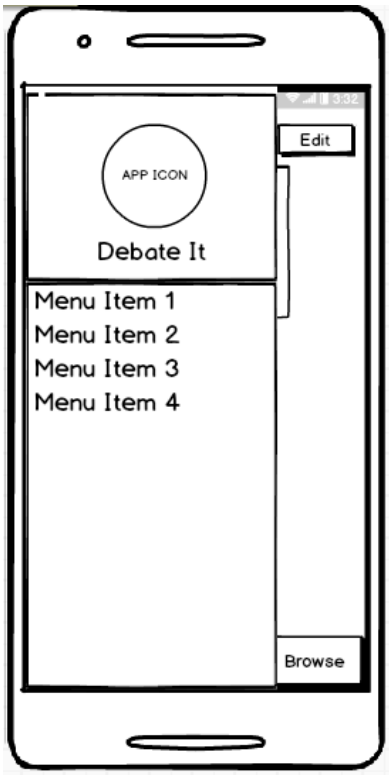
- **DataReceivable**

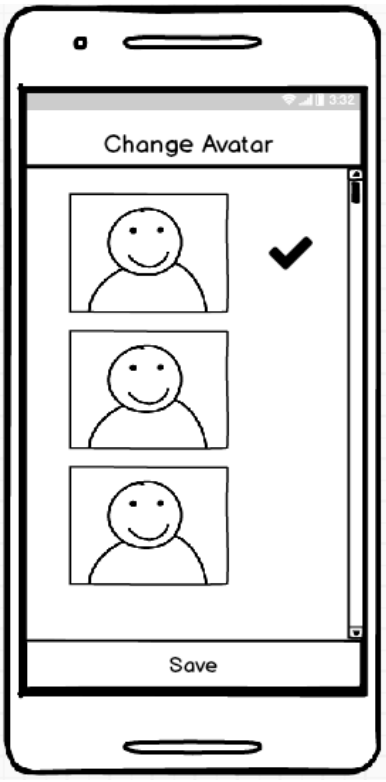
  ItemManager class is responsible for retrieving the items from database.

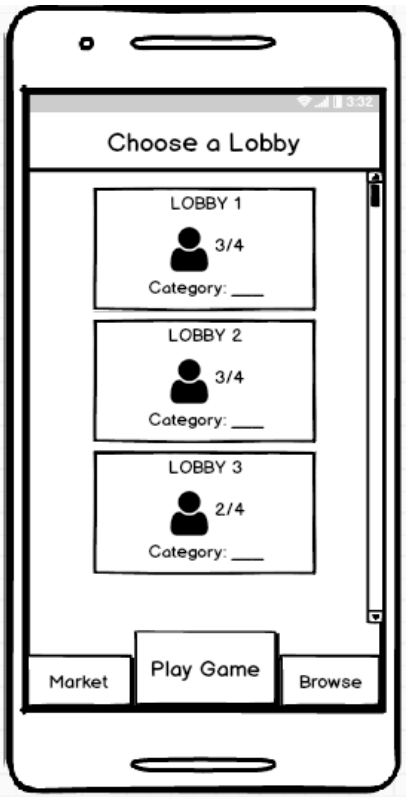
- **Mockups**

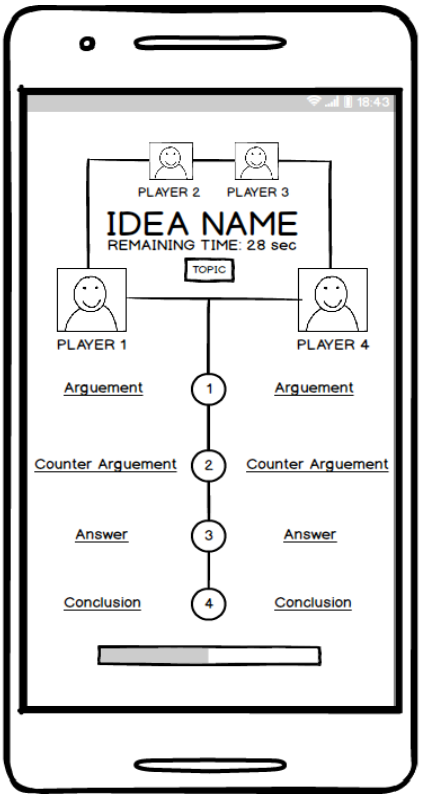

Register
Username
Password
Confirm Password
Register
OR
Login

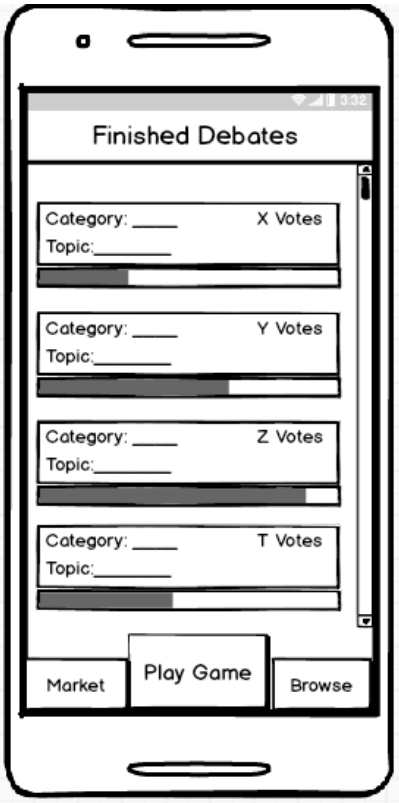

Login
Username
Password
Login
OR
Register


APP ICON
Edit
Debate It
Menu Item 1
Menu Item 2
Menu Item 3
Menu Item 4
Browse

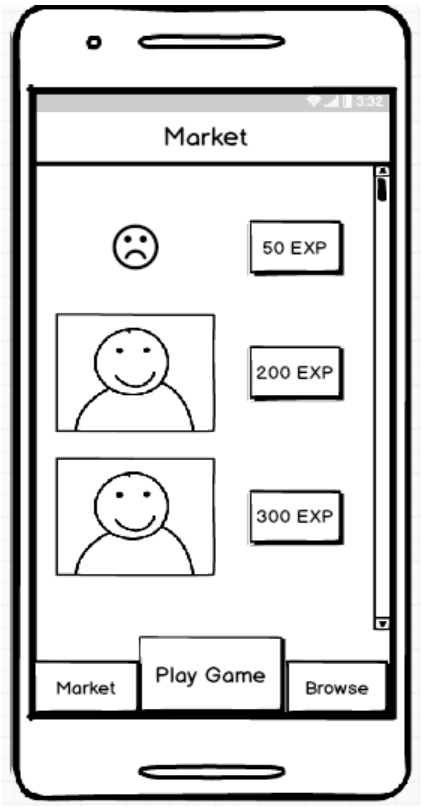

Change Avatar
Save

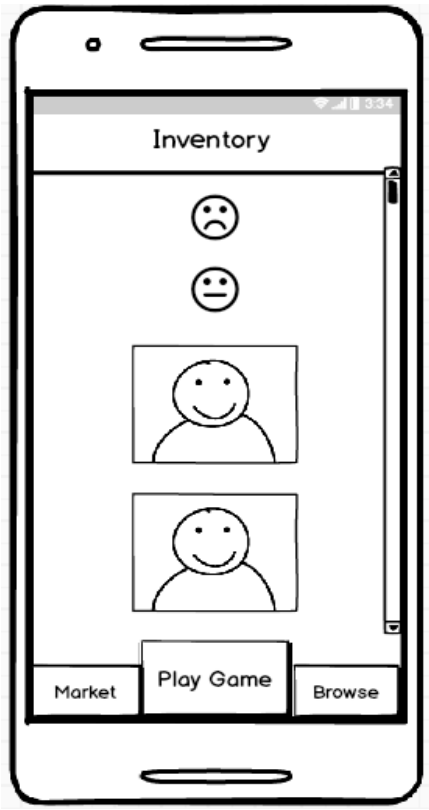

Choose a Lobby
LOBBY 1
3/4
Category: ___
LOBBY 2
3/4
Category: ___
LOBBY 3
2/4
Category: ___
Market
Play Game
Browse

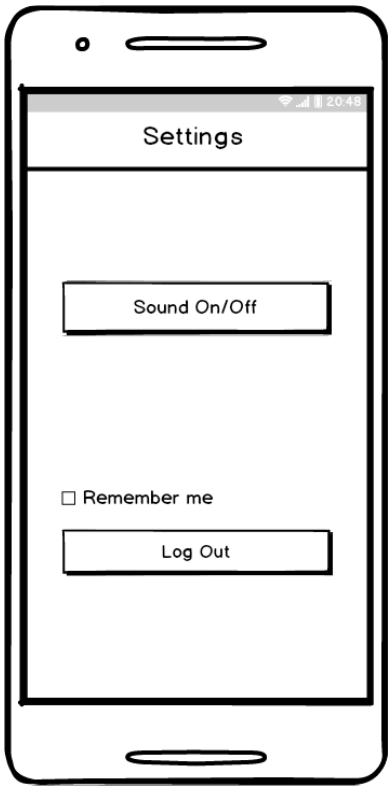

PLAYER 2
PLAYER 3
IDEA NAME
REMAINING TIME: 28 sec
TOPIC
PLAYER 1
PLAYER 4
Arguement
1
Arguement
Counter Arguement
2
Counter Arguement
Answer
3
Answer
Conclusion
4
Conclusion


Finished Debates
Category: ____
X Votes
Topic:_____
Category: ____
Y Votes
Topic:_____
Category: ____
Z Votes
Topic:_____
Category: ____
T Votes
Topic:_____
Market
Play Game
Browse


EXP: _
Inventory
Change
Nickname
Your Last Debates
Debate One
Debate Two
Debate Three
Button
Play Game
Browse


Market
50 EXP
200 EXP
300 EXP
Market
Play Game
Browse


Inventory
Market
Play Game
Browse


Settings
Sound On/Off
Remember me
Log Out

CONCLUSION

- **Glossary**

Idea: A controversial statement which will divide users into two sides.

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

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

Title: It will 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 debate battles.

- **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.