CS2013/CS3013

Project Plan

TCD ADAPT - Erwan Moreau

Group 20

Andrew Mc Donald
Conor Mac Amhlaoibh
Divine Mbunga
Ethan Monkhouse
Evan Mc Croary
Luke Mc Grath

1 Project Goals and Objectives

1.1 Background

The Turing Game is a mobile app envisioned by our client: Erwan Moreau of TCD ADAPT. It is an educational game in which a player must determine whether they are interacting with a human or an artificial intelligence (AI) after engaging them in a short conversation. The project has been split up into three parts: client-side, server-side and AI. Our group is handling the client-side aspect of the project.

1.2 Objectives

The purpose of the overall project is to introduce the concepts of AI and machine learning to average people in an interesting and educational manner. Seeing as our group is working on the client facing section of the application our primary objectives are to (1) make the user interface (UI) easily navigable and simple to understand and (2) sufficiently 'gamify' the user experience in order to make the app interesting and enjoyable.

1.3 Goals

Our project will deliver an Android application that will communicate with the application programming interface (API) developed by the group handling the server-side aspect of the project. The application will allow the end user to engage in conversations with another user or an AI and subsequently guess the 'type' (user or AI) of the other conversant. The end user should also be able to view a leaderboard containing their ranking compared to other users.

2 Project Scope

2.1 Project Deliverables

Upon the completion of the project we hope to deliver a completed Android application that is fully integrated with the server-side API. The application should authenticate the user via Google before allowing them to start a conversation and should ensure that no private data is stored as per the GDPR. The codebase of our application should be well-documented and easily extensible.

We also hope to deliver clear and thorough documentation of the API as well as instructions on integration. Alongside the documentation we will deliver a number of reports and specifications: the requirements document, this project plan, the software design specifications, the development plan and the management report. Each member of our group will also submit an individual reflective essay.

2.2 Project Boundaries

2.2.1 In Scope

- Login to the app using a Google account

- Start a conversation in a particular game mode
- At the end of a conversation receive a user's guess (human or AI)
- Flagging mechanism available from within a conversation
- Send API requests for messages, guesses, etc. to the server
- Parse JSON data received from the server
- Receive feedback from users in the form of a survey
- Display user ranking in a leaderboard
- Show a 'typing bubble' when a conversant is typing and simulate a typing bubble for the AI

2.2.2 Out of Scope

- The user interface should be minimal but aesthetic
- Detailed API documentation should be written
- API requests should be as efficient as possible
- API requests should be authenticated and secure
- User data should be GDPR compliant

2.3 Product Backlog

ID	Story	Priority	Sprint	Status
1	As an unauthenticated user I should be able to authenticate myself with my Google account.	Requirement	1	Completed
2	As a user I want to type a message so I can communicate with the other player	Requirement	1	Completed
3	As a user I want to see my message so I can see what I sent to the other player.	Requirement	1	Completed
4	As a user I want to see the other player's message so I can decide whether they are a human or AI player.	Requirement	1	Completed
5	As a user I want an AI button so I can choose that I was playing against an AI player.	Requirement	1	Completed
6	As a user I want a human button so I can choose that I was playing against a human player.	Requirement	2	In Progress
7	As a user I want to see a message when the other player is ready so I know when the game is about to start.	Feature	2	In Progress
8	As a user I want to see the leaderboard in order of score so I can quickly identify my position in comparison to everyone else.	Feature	2	In Progress
9	As a user I want to see if I was correct or	Requirement	2	In Progress

	incorrect so I know I well I played the game.			
10	As a user I want to be able to flag a conversation so I can report inappropriate behaviour.	Requirement	2	In Progress
11	As a user I want to see a message when the game is finding the other player so I know the game is working.	Feature	2	In Progress
12	As a system I want to connect to the score information of all the users so I can create a leaderboard.	Feature	2	In Progress
13	As a user I want to see my overall guess percentage or score so I know how well I am playing the game over all of my attempts.	Feature	3	Not Started
14	As a user I want to be able to fill out a survey so I can let the developers know how I feel about the game.	Requirement	3	Not Started
15	As a user I want to see the other player is typing so I know they are replying to my message.	Feature	3	Not Started
16	As a system I want to save received messages so I can analyse conversations to improve the game.	Feature	3	Not Started
17	As a user I want to save sent messages so I can analyse conversations to improve the game.	Feature	3	Not Started

Our requirements document and software design specification are included in Appendices 1 and 2, respectively.

3 Project Approach

3.1 Scrum Sprints

Using the Scrum Framework, we have broken our project down into epics, features and user stories. Our Product Backlog includes the user stories which are categorized under features and epics. Our product owner Connor prioritises certain user stories that need to be completed in our next sprint. We make sure everyone is clear of what each user story requires from a development standpoint before we begin our relative estimation and assigning user story points to already prioritised user stories. We calculate the user story points in estimation taking into account complexity and effort required for a developer to fully complete the work needed in the user story. Following the scrum guide, we use the Fibonacci sequence to assign a number to the complexity and effort of a user story. We agree on a number by discussing our estimates until we reach an acceptable level of consensus. If a user story has an estimate that is over 13 then we cannot bring that user story into the sprint until it is broken down into more manageable and simpler user stories. Once we reach what we feel is an achievable story points total with the combined points from each user story we have prioritised then we break down the user story into tasks to simplify and clarify the development process before a developer starts working on it. This helps us avoid or address obvious obstacles before allowing the developers to choose which user story they want to work on.

We use a Kanban board to keep track of who is working on what user story and avoid duplication of work. Our Kanban includes our Product backlog, Sprint backlog, in progress and Done. During a sprint our developers can only work on and assign themselves to user stories that are in the Sprint backlog.

Each of our sprints consists of the Product owner and scrum master prioritising and preparing user stories, a relative estimation meeting to assign user story points, sprint planning to choose what we can take into the sprint and break down chosen user stories, we carry out the sprint communicating through our group chat and using the kanban board properly through trello, we have a sprint review and a sprint retrospective to look at what we can improve and what we feel we should continue to do going into our next sprint.

Epic	**Feature**	**User Story**	**US Points**
Game mode	User message	As a user I want to **type a message** so I can communicate with the other player.	3
		As a user I want to **see my message** so I can see what I sent to the other player.	5
		As a user I want a **send button** so I can send my message to the other player.	3

		As a system I want to **save sent messages** so I can analyse conversations to improve the game.	8
	Bot Message	As a user I want to **see the other player is typing** so I know they are replying to my message.	5
		As a user I want to **see the other player's message** so I can decide whether they are a human or AI player.	5
		As a system I want to **save received messages** so I can analyse conversations to improve the game.	8
	Guess Option	As a user I want an **AI button** so I can choose that I was playing against an AI player.	5
		As a user I want a **human button** so I can choose that I was playing against a human player.	5
	Result Message	As a user I want to **see if I was correct or incorrect** so I know I well I played the game.	8
		As a user I want to **see my overall guess percentage or score** so I know	8

		how well I am playing the game over all of my attempts.	
	Find Player	As a user I want to **see a message when the other player is ready** so I know when the game is about to start.	13
		As a user I want to **see a message when the game is finding the other player** so I know the game is working.	8
Leaderboard	Display and update leaderboard	As a system I want to c**onnect to the score information of all the users** so I can create a leaderboard.	13
		As a user I want to **see other users and their scores** so I can compare myself against them.	5
		As a user I want to **see the leaderboard in order of score** so I can quickly identify my position in comparison to everyone else.	5

4 Project Organisation

4.1 Staff

Team Member Name	**Past Experience**	**Technical Skills**
Conor Mac Amhlaoibh	Software Engineering Project 2019, Programming Project 2018	Python, C#, Java
Evan McCroary	Software Engineering Project 2019, Programming Project 2018	Web development, Java, C, SQL, Python
Divine Mbunga	Software Engineering Project 2019, Programming Project 2018	Web Development, Java, Python
Luke McGrath	Programming Project 2019	Java, C
Andrew McDonald	Programming Project 2019	C, Java
Ethan Monkhouse	Programming Project 2019	Web Design, Java, C

4.2 Staff Chart

Team Member Name	**Role**
Conor Mac Amhlaoibh ***[Product Owner]***	Backend Manager
Evan Mc Croary ***[Scrum Master]***	Frontend Manager
Divine Mbunga	UX/I Manager
Ethan Monkhouse	UX/I Developer
Luke McGrath	Frontend Developer
Andrew Mc Donald	Backend Developer

5 Risk Analysis

5.1 Risk Analysis

Risk Element	Impact (1 to 5)	Likelihood (1 to 5)	Risk Factor (I * L)
Not complying to GDPR.	5	3	15
Failure to meet deadlines.	4	4	16
Miscommunication with other teams involved in the project.	3	4	12
Miscommunication with the client.	5	2	10
Conflict within the team.	4	2	8
Failure to complete the functional requirements.	5	2	10

5.2 Risk Mitigation

Risk	Measures to Reduce Risk
Not complying to GDPR.	Ensuring that the user has the ability to delete data associated with them.
Miscommunication with other teams involved in the project.	Uploading important documents to the repository, keeping everyone up to date with major changes and following a standard.
Miscommunication with the client.	Meeting when anything needs to be discussed and being clear with any issues or concerns that may arise.
Conflict within the team	Have regular meetings discussing any issues or concerns anyone has and how the team can move forward.
Failure to complete the functional requirements.	Focus on completing the functional requirements before moving onto the non-functional requirements.

6 Project Controls

Factor	**Control Method**
Scope	The scope of this project is to develop a downloadable app that employs the turing game, in a way that’s easy to use and accessible to all users in an attempt to familiarise people with AI who wouldn’t usually be immersed in the tech world. We discussed with the client as to what he considered primary and secondary requirements. Knowing this, we were able to work on what the client was after first, taking out the possibility of a mad dash toward the end due to poor time management.
Quality	The quality will be ensured by our pair system. Managers work closely with their paired developer to ensure the code produced is clear, concise and most importantly, functional. Where the code fails to meet these objectives the manager can step in and offer help where needed.
Schedule	We have set up weekly group meetings and outline our goals for the following week there. This guarantees the timing of the project is constant and well managed. Another benefit to this schedule is the constant public accountability of each person, making sure everyone pulls their weight in the team. Additionally the group meets with a demonstrator who is very helpful not least because of his experience in the field. Keeping strict to the schedule has been invaluable to the overall efficiency of the team.

7 Communication

7.1 Client Communication

We hope to maintain clear and consistent communication with the client throughout the entire development of the application. Until now we have maintained great communication with the client. Frequent meetings with the client and communication via email have allowed us to form a great understanding of what the client has in mind for the application. The client has been very helpful so far.

7.2 Project Team Meetings

Team meetings take place every week where we discuss things like our progress and our goals. Setting goals for the following week and reviewing our goals from the previous week allows us to keep track of where we stand in terms of our overall progress with the project. Outside of meetings we communicate via Messenger not only within our own group but also with the two other groups involved in the Turing Game for integration purposes.

Appendix 1: Requirements Document

1. Introduction

1.1 Overview - Purpose of the System

In this project three groups will be working on building the Turning Game. The Turing test is popular in the domain of AI. It is a game where a person asks written questions, in the similar to a messaging app, to two entities, player A and player B. These entities will reply through messages. The goal is for the person to guess who between player A and player B is a human and which is the AI. The Turing test game is considered a foundational description of what could be defined as artificial intelligence. If a human cannot recognise the difference between a conversation with a human or with a machine, then this machine shows some form of intelligence.

1.2 Scope

For the Turing Game application, our scope is to design and implement the game modes and user interface of the overall system, focusing on the interface of a sign in component, the dialogue system, a scoring system and a way for the player to flag any inappropriate content from another player or the AI.

1.3 Objective and Success Criteria

The main objective of this project following completion is to have built upon the previous years project, and implemented the features which have been agreed with the client, to a standard that all developers, managers, the client and all other parties involved agree is of a satisfactory level. Our specific objectives *to have achieved by the the completion of this project are the implementation and testing of the Turing Game's user interface, game modes, and extra features agreed with the client previously*.

Below outlines the key metrics of which the team completing the following project will follow in order to assess the progress and quality of the project, as well as the overall success, throughout the duration of the project.

1. User Involvement - ***Agreed by team***
2. Support from Managers - ***Agreed by team***
3. Clear Requirements Documentation - ***Client briefed***
4. Satisfactory Planning - ***Assessed by team***
5. Realistic Expectations - ***Client briefed***
6. Project Milestones - ***Confirmed with client***
7. Competent Team Members - ***Assessed by team***
8. Ownership of Work Completed - ***Agreed by team***
9. Clear Vision of Objectives - ***Agreed by team***
10. Hard-working, Focused Team Members - ***Agreed by team***

2. Current System

The current system was designed by last year's group using Android Studio. The system features an interface which features three buttons. The user can initiate a game and begin conversation by pressing the start button. There are also buttons which bring the use to a contact section or to settings. After starting a game the user can message back and forth with the bot/human and proceed to make a guess. The system then displays the result.

3. Proposed System

3.1 Overview

The proposed system will be an improved version of last year's system. The system will essentially be an interface for the Turing App that includes a sign-in feature, dialogue with a human/bot, a scoring system and a system that allows the user to flag inappropriate content. Building on last year's project, the system will also include requirements discussed with the client such as game modes and an option to receive feedback from the user, in the form of a survey, for example.

3.2 Functional Requirements

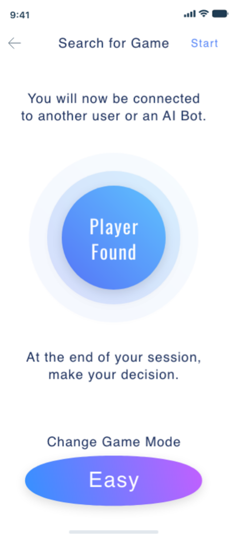
- Login to the app using a Google account
- Start a conversation in a particular game mode
- At the end of a conversation receive a user's guess (were they speaking to a bot or a human)
- Flagging mechanism within a conversation
- Send API requests for messages, guesses, etc. to the server
- Parse JSON data received from the server
- Receive feedback from users in the form of a survey
- Display user rankings in a leaderboard
- Show 'typing bubble' when a conversant is typing

3.3 Non-functional Requirements

- The user interface should be minimal but aesthetic
- Detailed API documentation should be written
- API requests should be as efficient as possible
- User data should be GDPR compliant

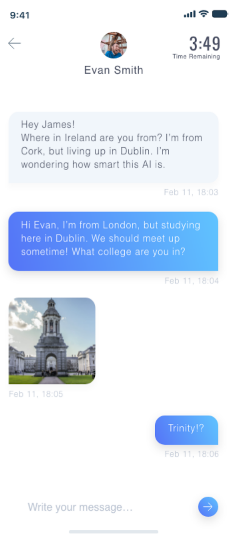
3.4 System Prototype

3.4.2 User Interface Mockups



You will now be connected to another user or an AI Bot.

At the end of your session, make your decision.



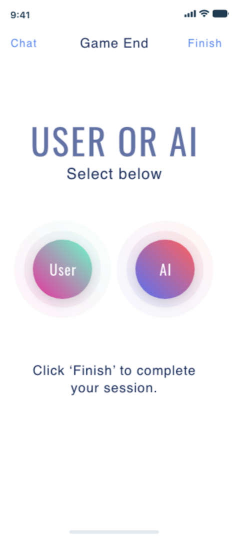
Hey James!
Where in Ireland are you from? I'm from Cork, but living up in Dublin. I'm wondering how smart this AI is.
Feb 11, 18:03

Hi Evan, I'm from London, but studying here in Dublin. We should meet up sometime! What college are you in?
Feb 11, 18:04

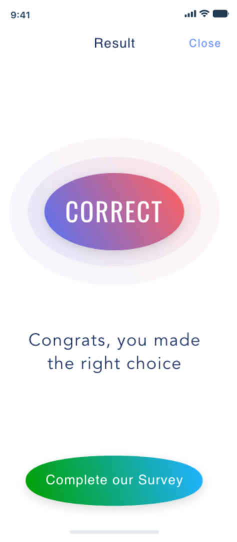
Feb 11, 18:05

Trinity!?
Feb 11, 18:06

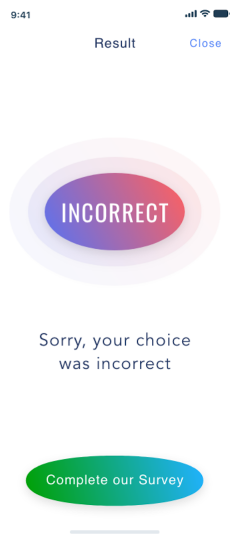
Write your message...



Click 'Finish' to complete your session.

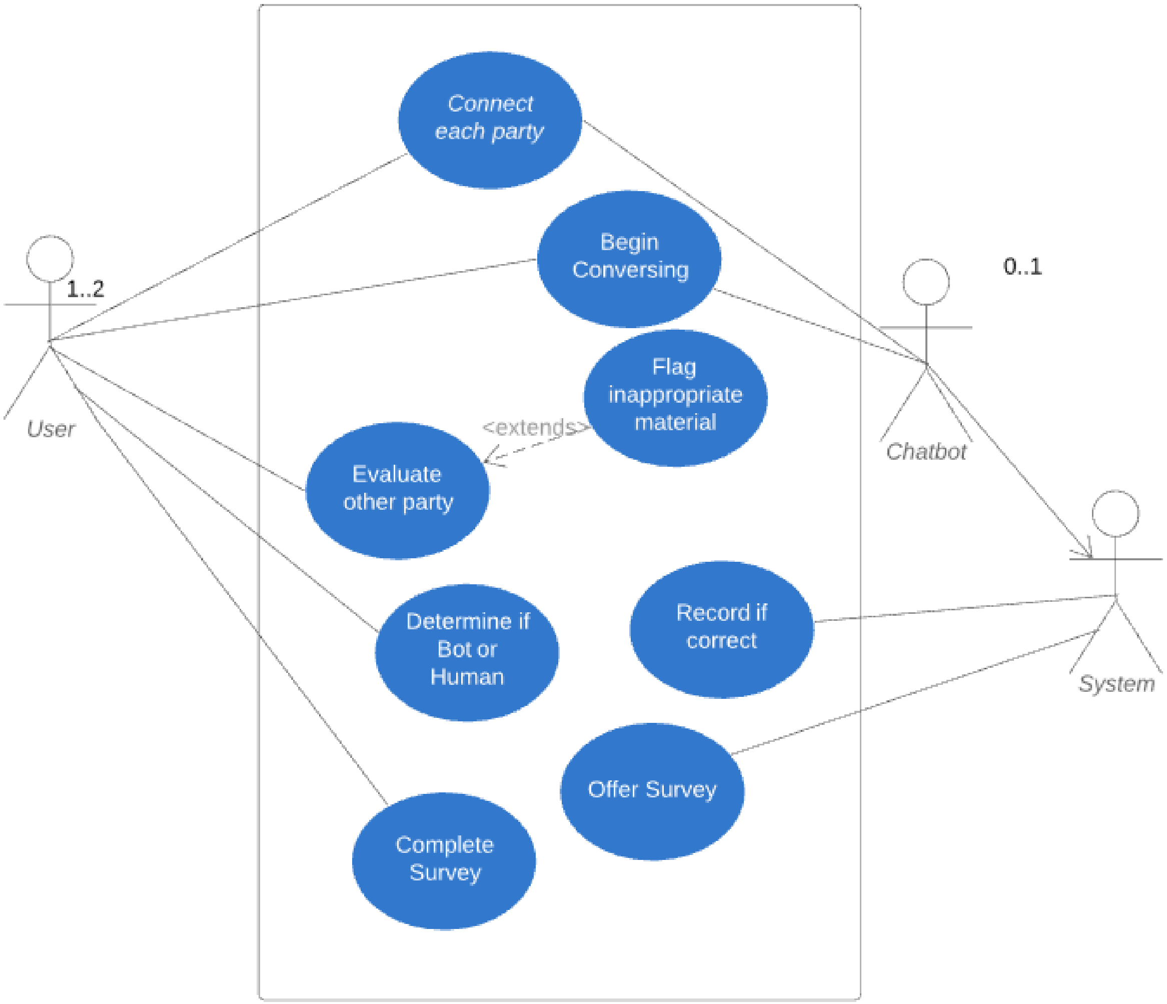


Congrats, you made the right choice



Sorry, your choice was incorrect

3.4.2 Use Case Diagram

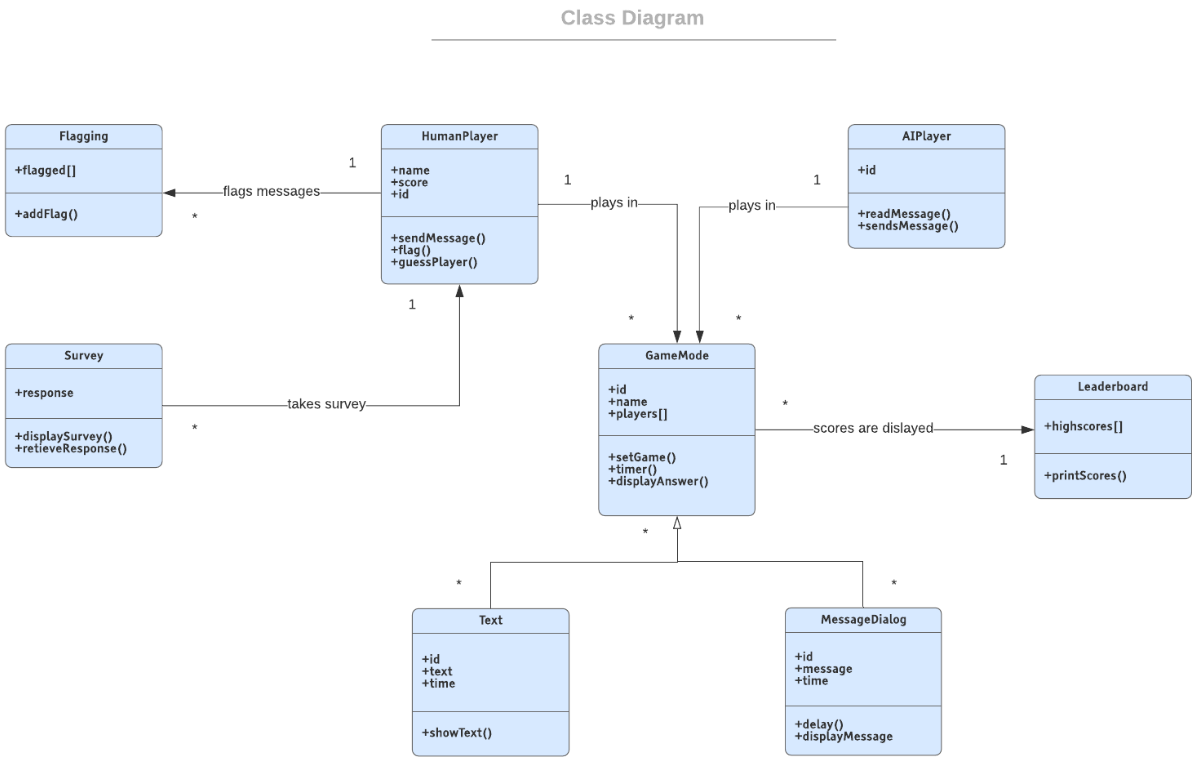


Use Case Name		Offer Survey
Actors		System and User
Precondtions		User has finished the game and is still on the app
Normal Flow	Description	User has finished the game. The system offers the option of a survey to the user who can accept or decline
	Postconditions	User fills out survey for information
Alternate flows and exceptions		User exits app after game and survey cannot be offered

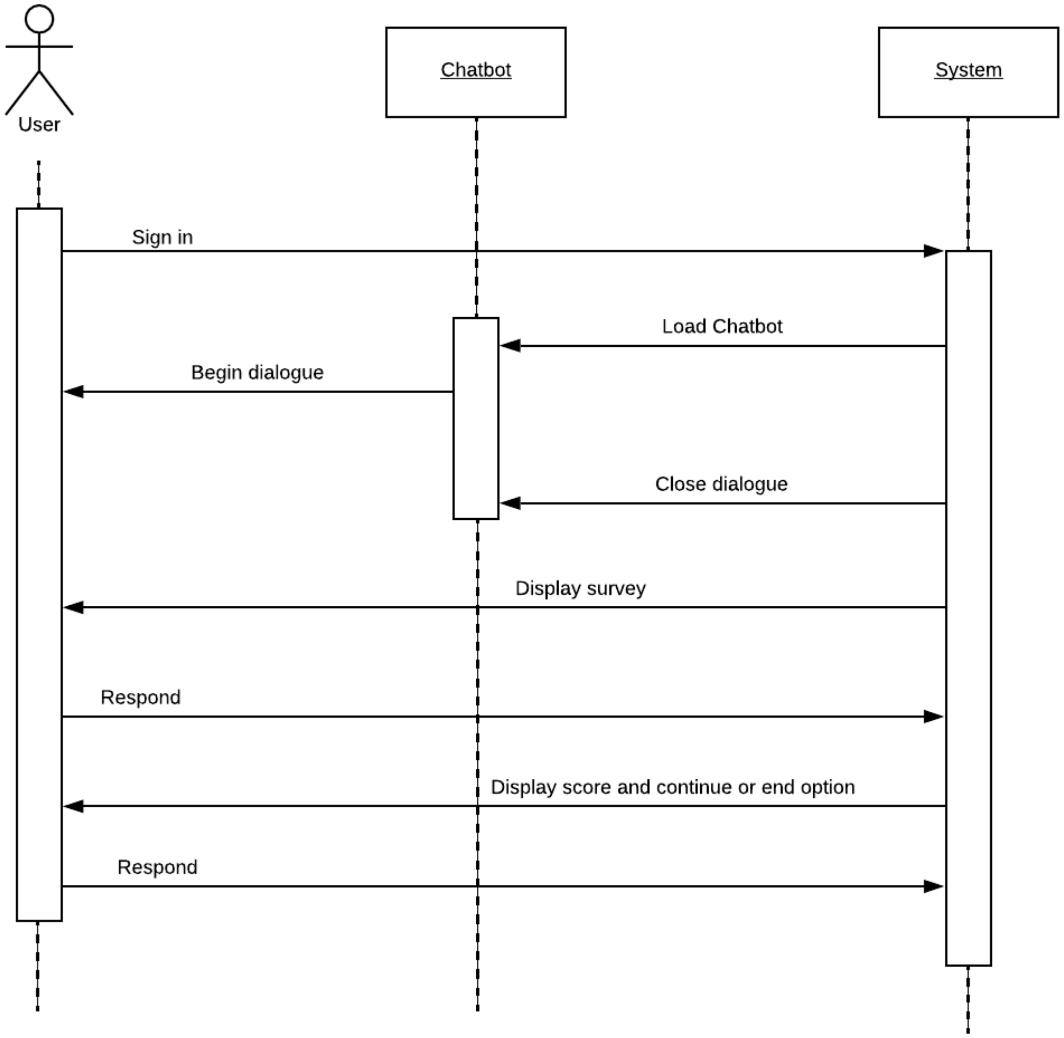
Use Case Name		Flag inappropriate material
Actors		Human Player
Precondtions		Player sees a message they believe shouldn't be on the platform
Normal Flow	Description	Player sees a message they believe to be inappropriate. They report it. The conversation is terminated and the conversation material is passed onto a trusted third party for review
	Postconditions	Player can no longer see material but is being acted upon by a separate party
Alternate flows and exceptions		No exceptions

Use Case Name		Connecting Parties
Actors		Two players or a player and a chatbot
Precondtions		User selects play and is connected to the internet
Normal Flow	Description	User connects to server and is matched with either a chatbot or another player
	Postconditions	User is ready to play the Turing Game against their partner and guesses if the player is a bot or another human
Alternate flows and exceptions		If the user disconnects or loses connection to the server, the game is stopped and connection failed.

3.4.3 Object model - Class Diagram



3.4.4 Dynamic model - Sequence Diagram


User
Chatbot
System
Sign in
Load Chatbot
Begin dialogue
Close dialogue
Display survey
Respond
Display score and continue or end option
Respond

Appendix 2: Software Design Specification

1 Introduction

1.1 Overview

Our client is Erwan Moreau who is based in the ADAPT Centre in Trinity College Dublin. The purpose of the *The Turing Game* project is to introduce the concepts of AI and machine learning to average people in an interesting yet educational manner. Our client wants the project to be implemented as an application for Android devices. A prototype application was developed by a previous group as part of last year's *Software Engineering* module and one of the primary goals of this project is to improve on that application and to extend the features provided. Our group will be developing the client-side aspect of the project.

1.2 Scope

The scope of our project consists of developing a fully-functioning Android application that implements all of the client's initial requirements for *The Turing Game* as well as numerous additional features. We will be using Android Studio to develop the application as it is the 'gold-standard' when it comes to Android application development. Our application should be fully integrated with the server-side API which will be developed by a separate group - user authentication will be implemented using Google's Firebase tool. Additional care should be taken to ensure that the application is in full compliance with the GDPR.

1.3 Definitions and Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
JSON	JavaScript Object Notation
UI	User Interface
XML	eXtensible Markup Language

2 System Design

2.1 Design Overview

Our project consists of an Android application which will implement the client-side aspect of *The Turing Game*. Our application will allow the end-user to converse with other users or an AI for a limited period of time and subsequently guess the nature of their conversant. The application will communicate with the server-side part of the project using a shared API.

2.1.1 High-level Overview

Languages used:

- Java
- PHP
- XML

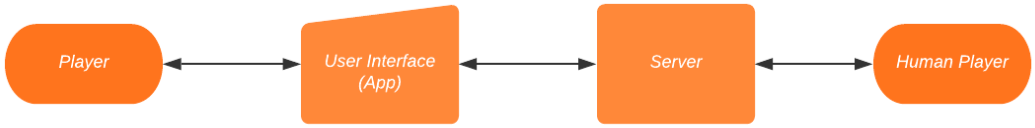
Tools used:

- Android Studio
- Firebase
- GitHub
- Server-side API
- Trello

2.2 System Design Models

Text.

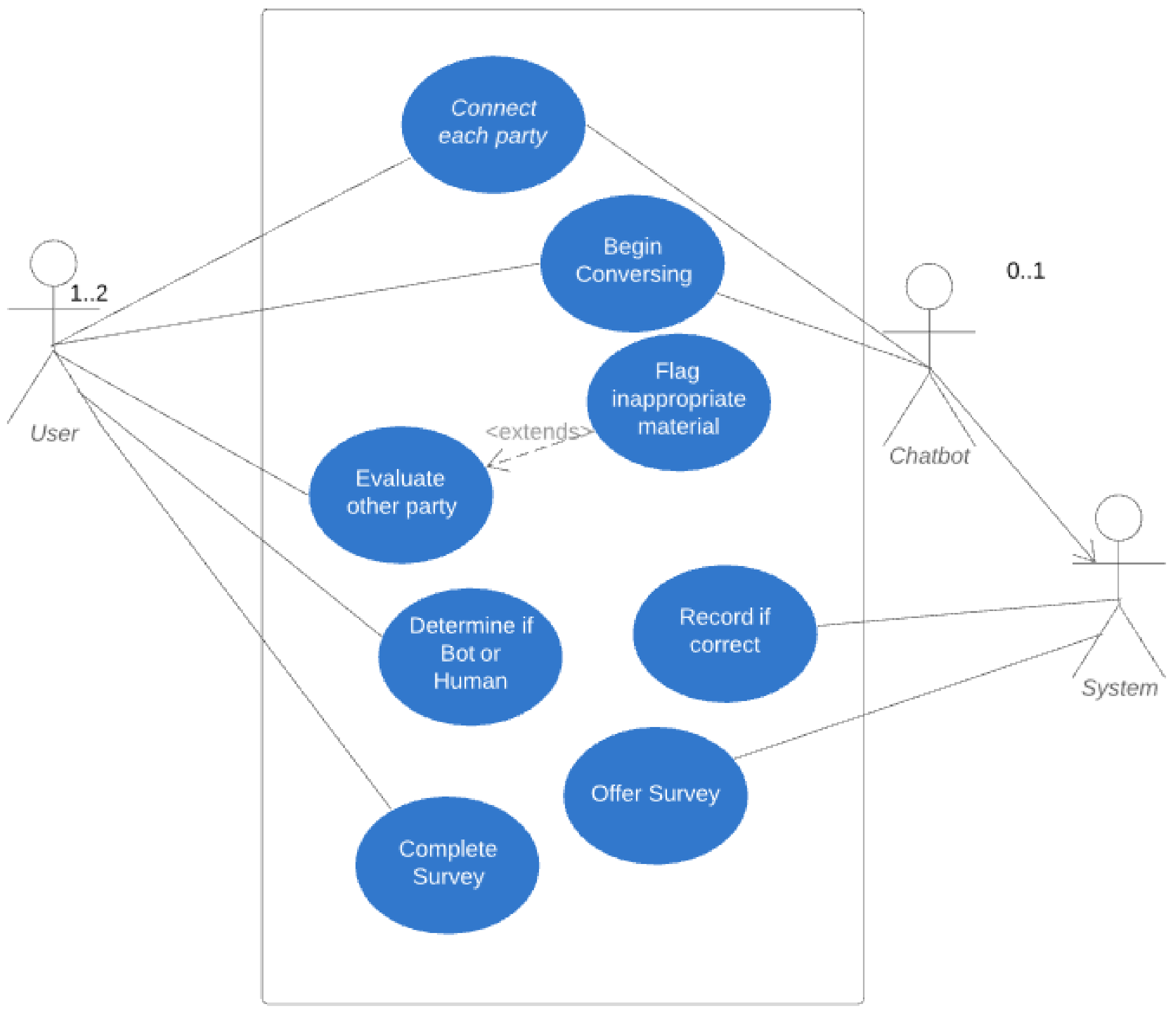
2.2.1 System Context



1. The user requests a new game, this request and details of the game is sent to the server which sources either a human player or AI player.
2. The ID of the human player or AI player, and type is returned.
3. Conversation is exchanged for the set time frame between the users. Any flagged messages are pushed to the server where a log is kept.
4. User decision is recorded and compared to the type passed from the server. If true, the player wins, and this is recorded on the players profile.
5. Survey data is passed to the server alongside the user ID and chat log for further improvement.

2.2.2 Use Cases

Use Case Diagram:

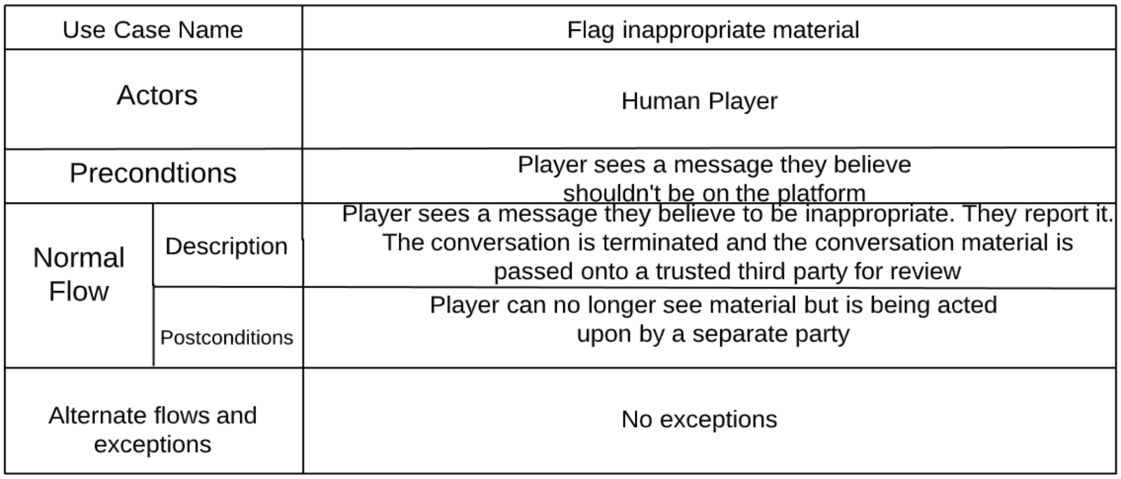


Use Case Textual Descriptions:

1.

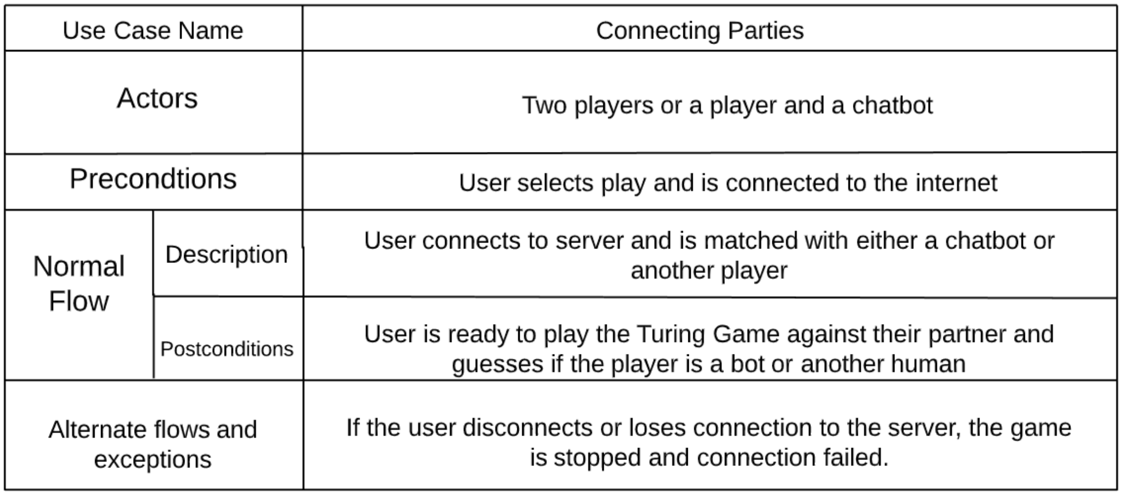
Use Case Name		Offer Survey
Actors		System and User
Precondtions		User has finished the game and is still on the app
Normal Flow	Description	User has finished the game. The system offers the option of a survey to the user who can accept or decline
	Postconditions	User fills out survey for information
Alternate flows and exceptions		User exits app after game and survey cannot be offered

2.



Use Case Name		Flag inappropriate material
Actors		Human Player
Precondtions		Player sees a message they believe shouldn't be on the platform
Normal Flow	Description	Player sees a message they believe to be inappropriate. They report it. The conversation is terminated and the conversation material is passed onto a trusted third party for review
	Postconditions	Player can no longer see material but is being acted upon by a separate party
Alternate flows and exceptions		No exceptions

3.



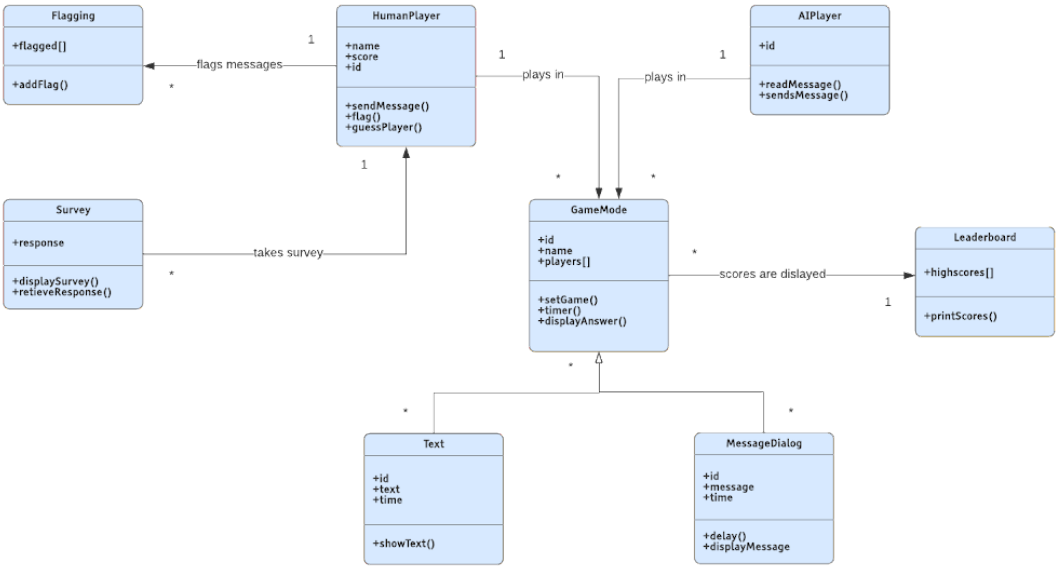
Use Case Name		Connecting Parties
Actors		Two players or a player and a chatbot
Precondtions		User selects play and is connected to the internet
Normal Flow	Description	User connects to server and is matched with either a chatbot or another player
	Postconditions	User is ready to play the Turing Game against their partner and guesses if the player is a bot or another human
Alternate flows and exceptions		If the user disconnects or loses connection to the server, the game is stopped and connection failed.

2.2.3 System Architecture

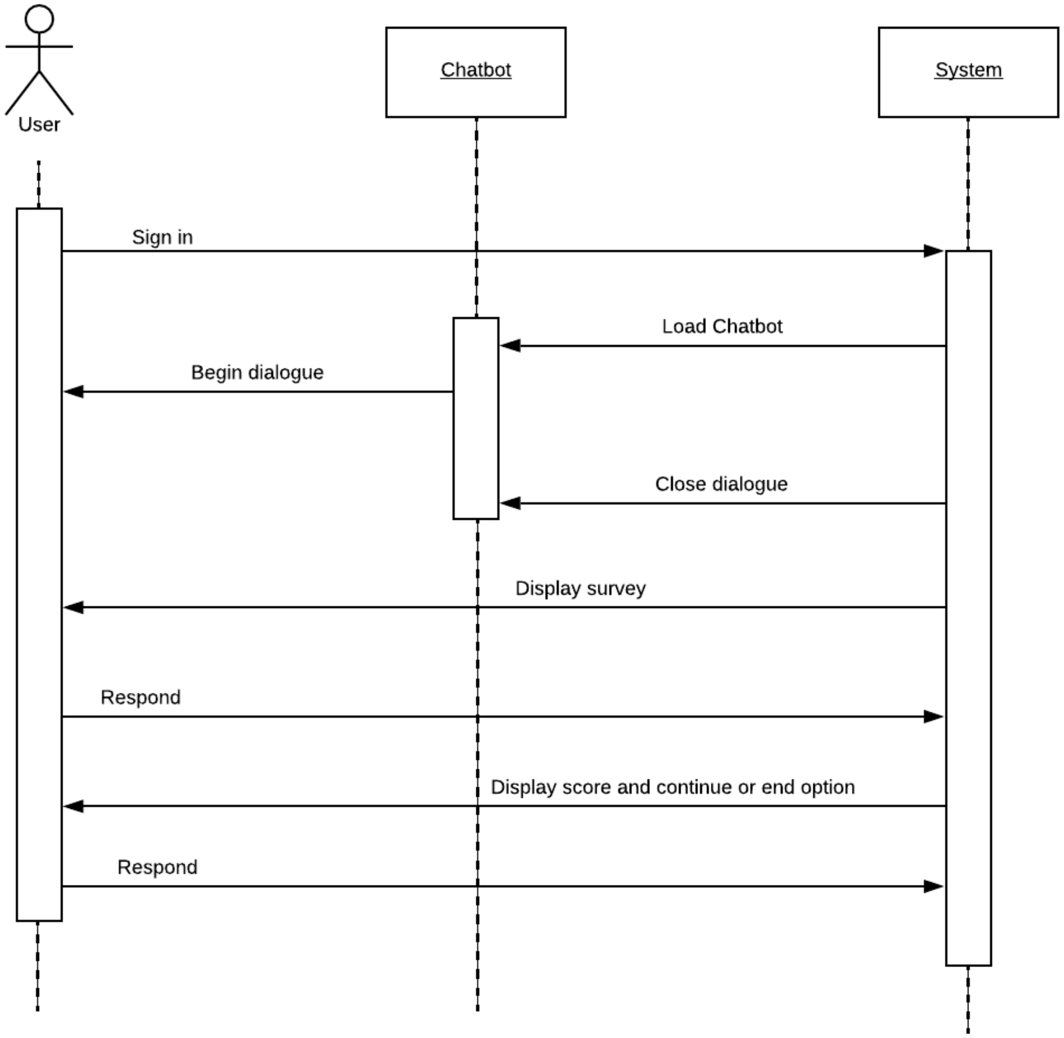
The system will allow a user to initially sign in using Google's Firebase tool. Upon authentication the user can choose to initiate a game, view the leaderboards or view the settings screen. During a game the user will converse with either the chatbot (which is designed by another group) or another human. Dialogue will be displayed on screen, which is all designed using Android Studio. The user can also view the leaderboards screen or change settings in relation to the application by navigating to the settings screen.. Both of these screens are also designed using Android Studio.

2.2.4 Class Diagrams

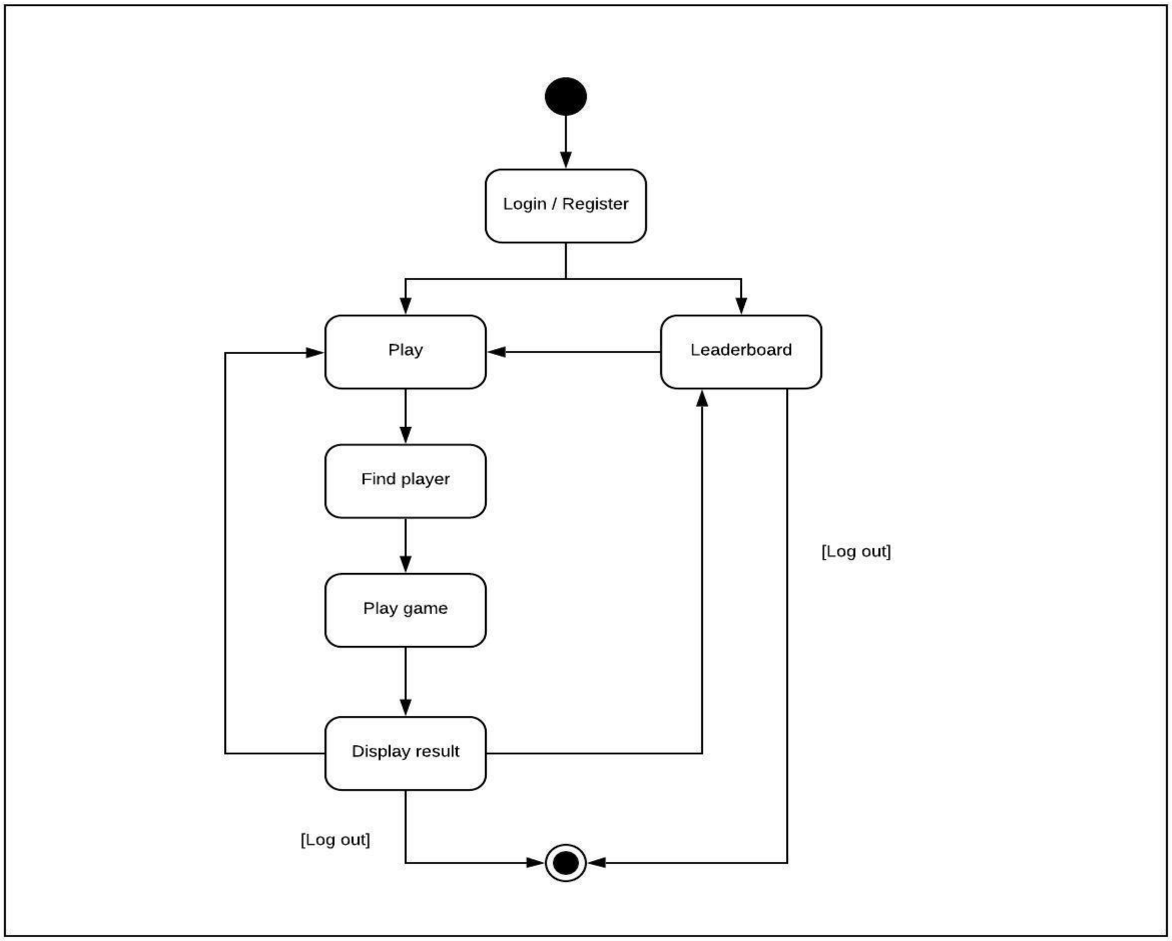
Class Diagram



2.2.5 Sequence Diagrams



2.2.6 State Diagrams


Login / Register
Play
Leaderboard
Find player
Play game
Display result
[Log out]
[Log out]