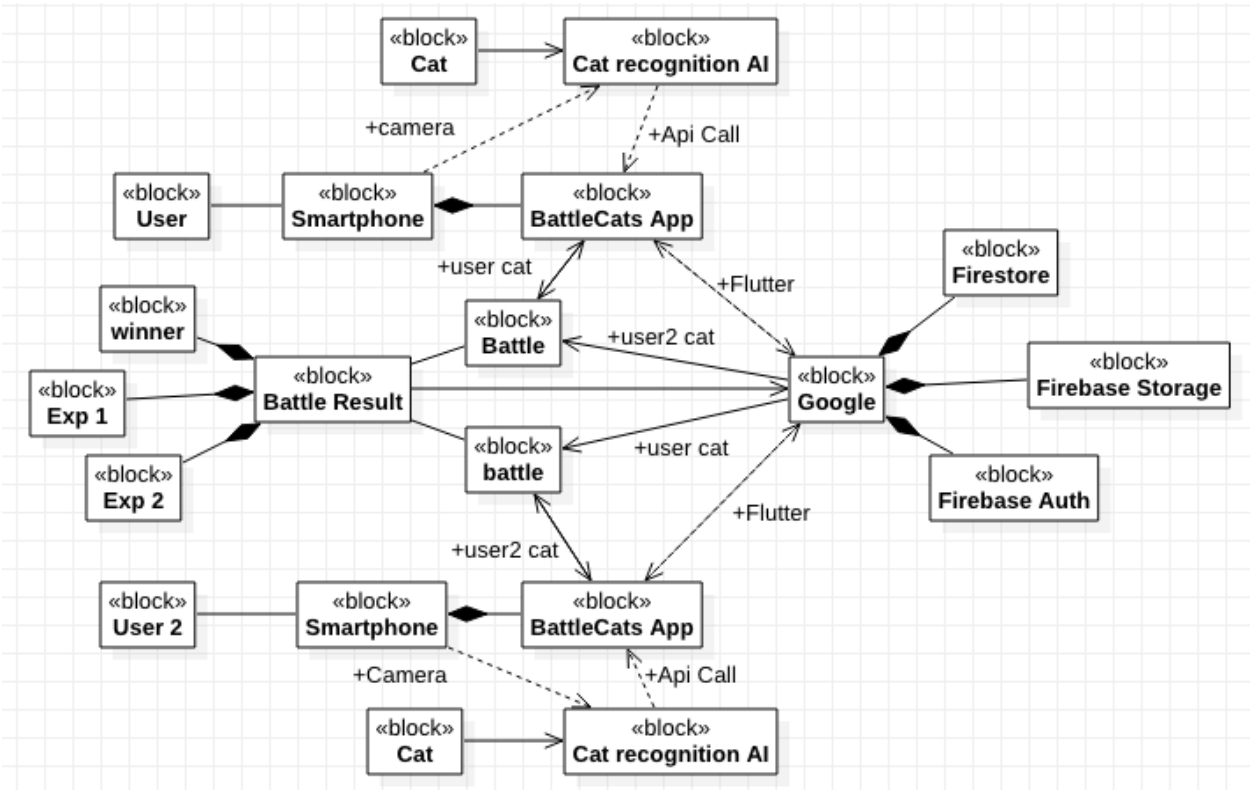
Kyle Handy
Oliver Lynch
Eric DeMoney
Jacob Carlson
Ryota Haba
Tyler Lopez
Team #9: Battle Cats

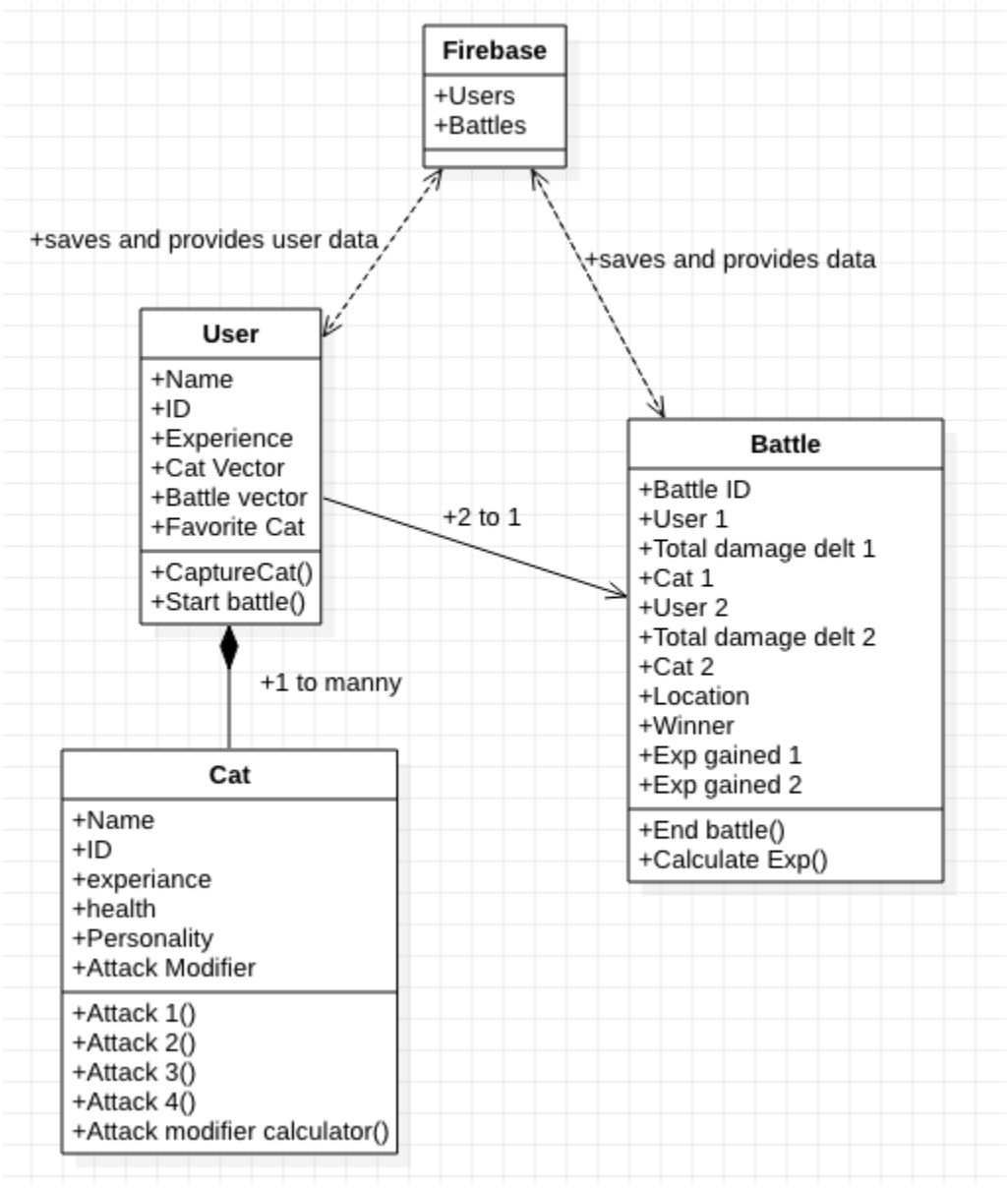
Exercise #6: Project Design Final Status Report

1. Block Diagram: This shows every physical and cloud based entity that two users interact with when adding a cat and starting a battle with another user.

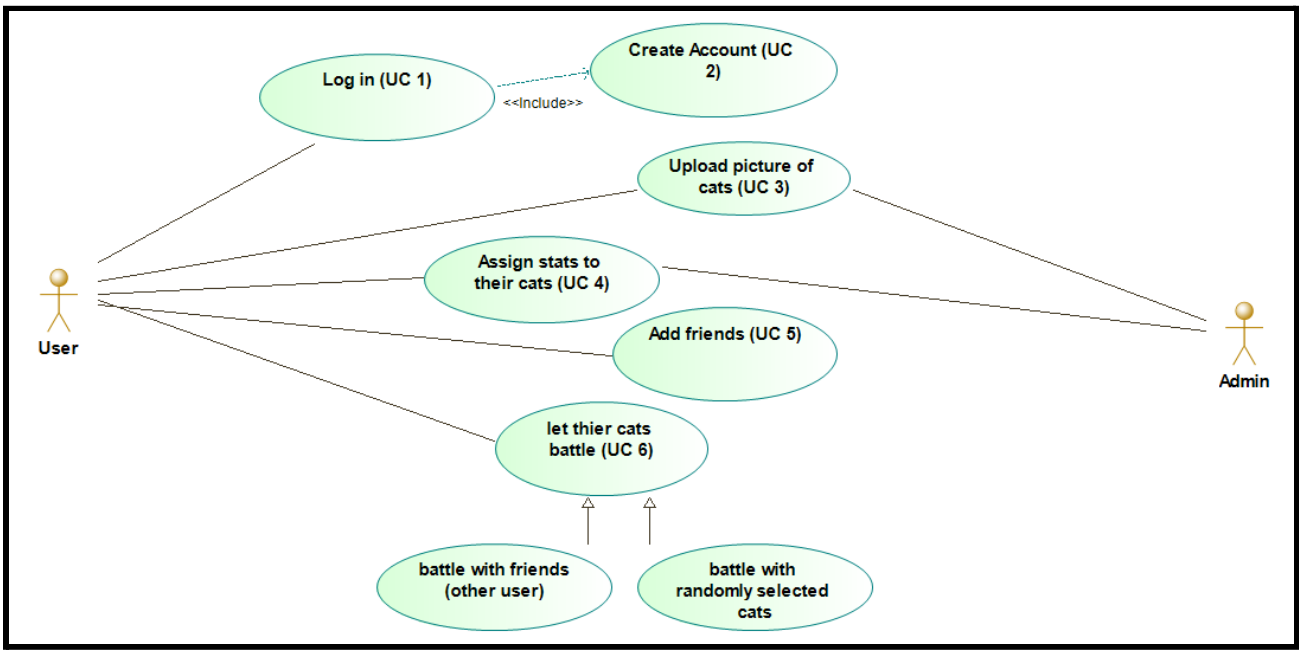


2.

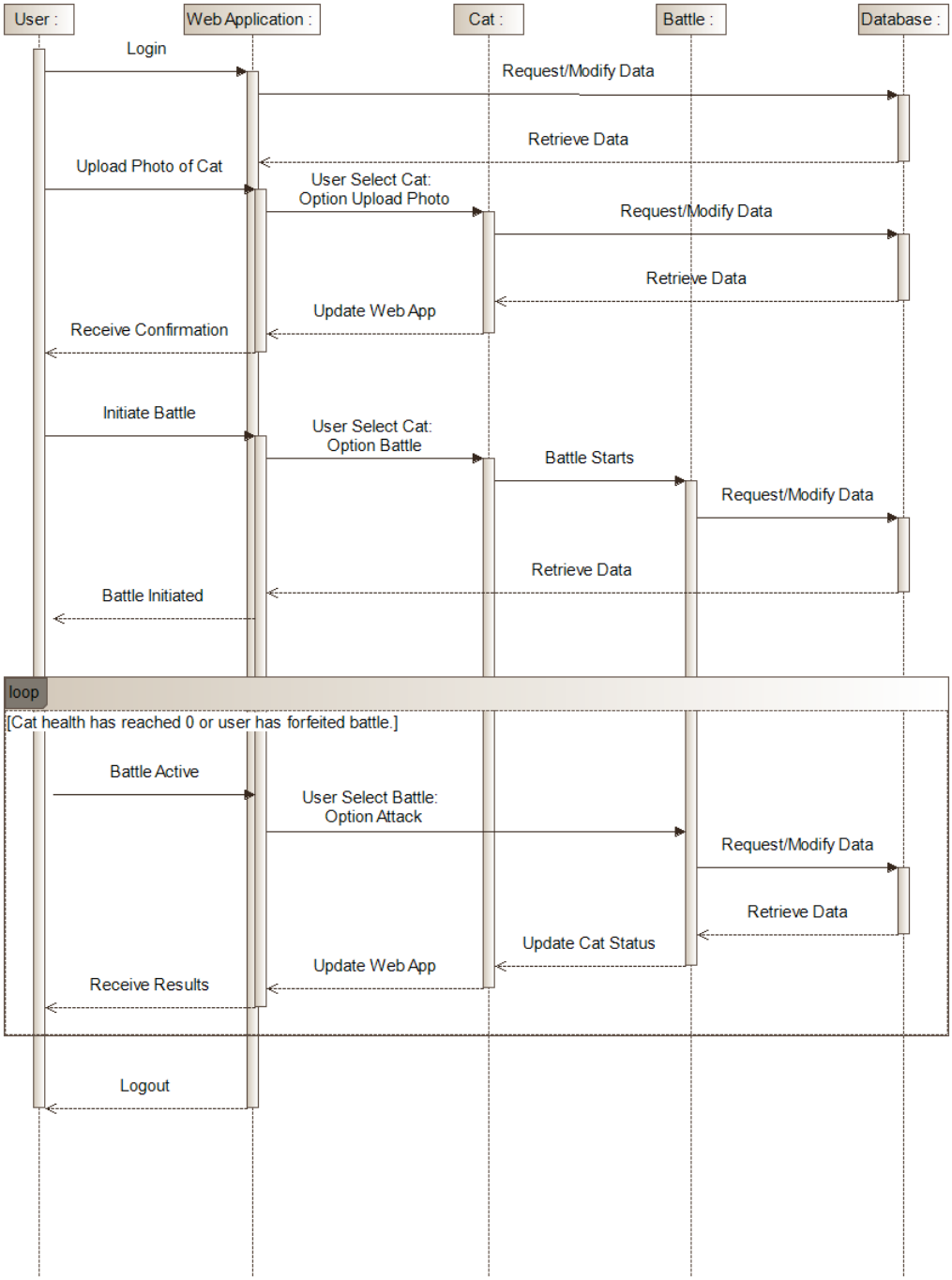
a. Class Diagram: This diagram shows the class systems we are working with for general entity relationships



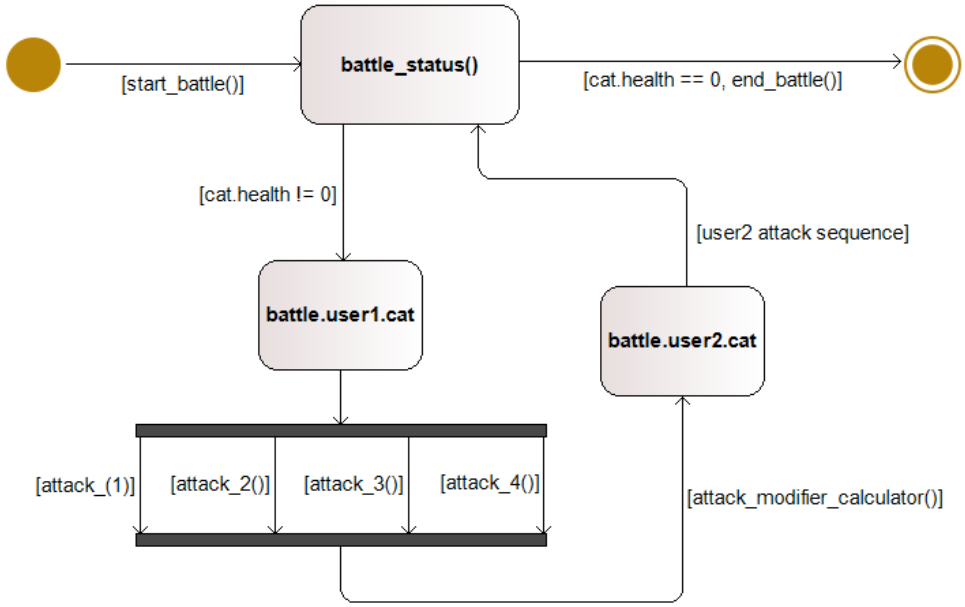
b. Use Case diagram:



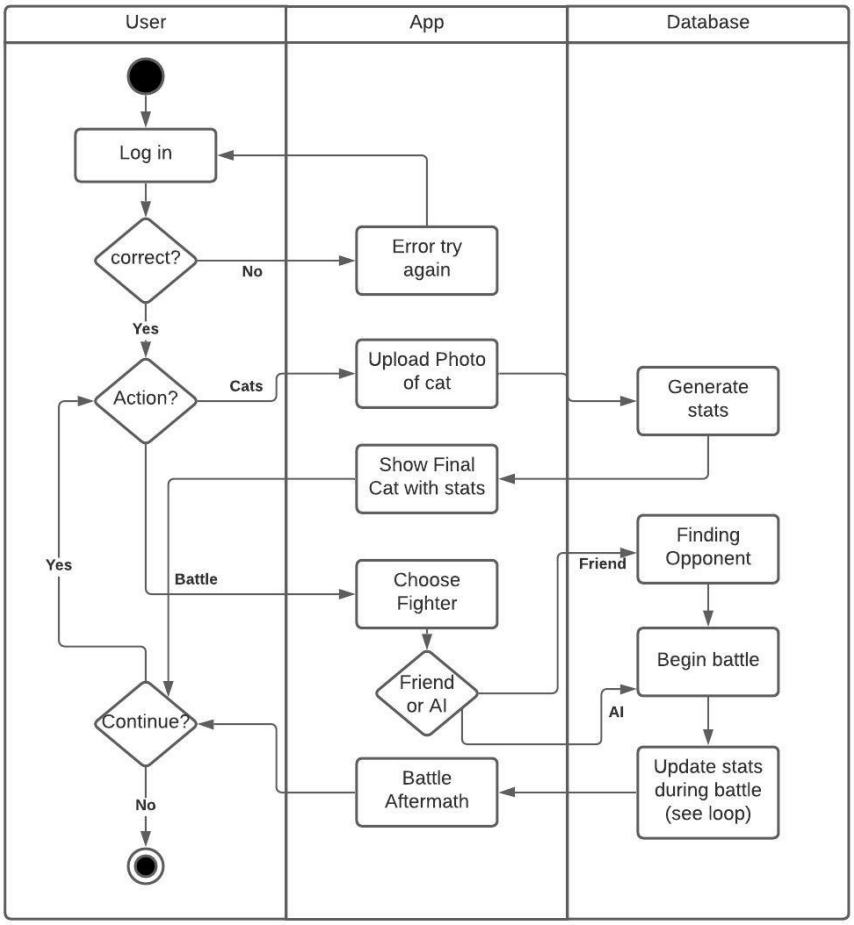
c. Behavioral Diagram:
    i. Interaction Sequence Diagram: This diagram represents the relationship between user, web application and database. Three specific cases are modeled: user login/logout, user upload cat, user battle sequence.

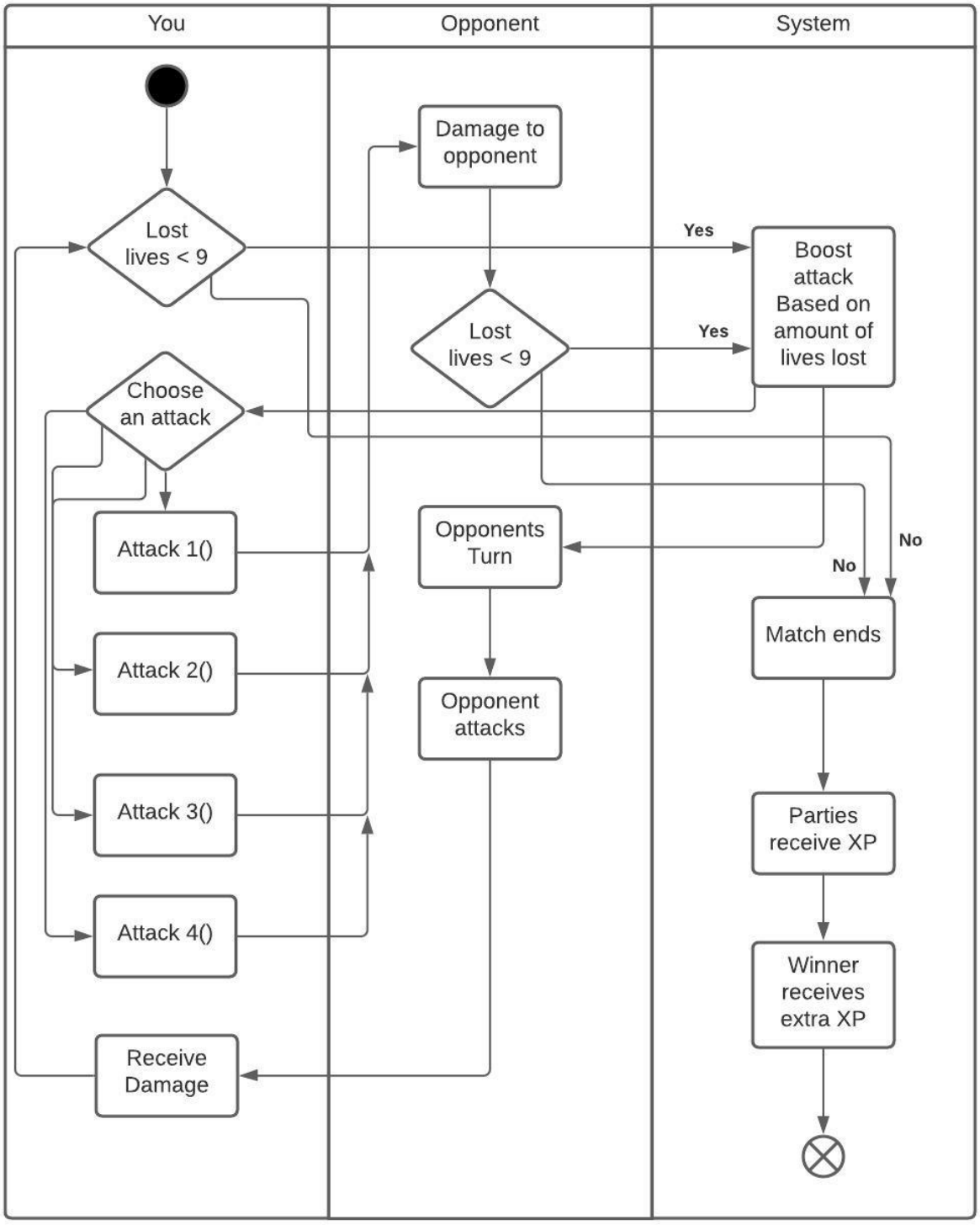


ii. State Machine Diagram: This model is a general layout of the battle sequence utilizing classes and functions from the Class Diagram.



d. Activity Diagram: This diagram represents the general process/all the steps that one person can take while using this application




You
Opponent
System
Damage to opponent
Lost lives < 9
Yes
Boost attack Based on amount of lives lost
Lost lives < 9
Yes
Choose an attack
Attack 1()
Opponents Turn
No
No
Match ends
Attack 2()
Opponent attacks
Attack 3()
Parties receive XP
Attack 4()
Winner receives extra XP
Receive Damage

3. a. Major requirements:
    i. This application must allow for users to enter photos of their cats and assign stats to them.

    ii. Battle Cats will allow for users to battle other users' cats or randomly generated opponents.

    iii. This application will allow users to add friends and have real time battles with them.

    iv. This application must use a real time database (Firebase) to allow for constant updates to the users view.

    v. Battle Cats will be extremely user friendly and have a simple and efficient design.

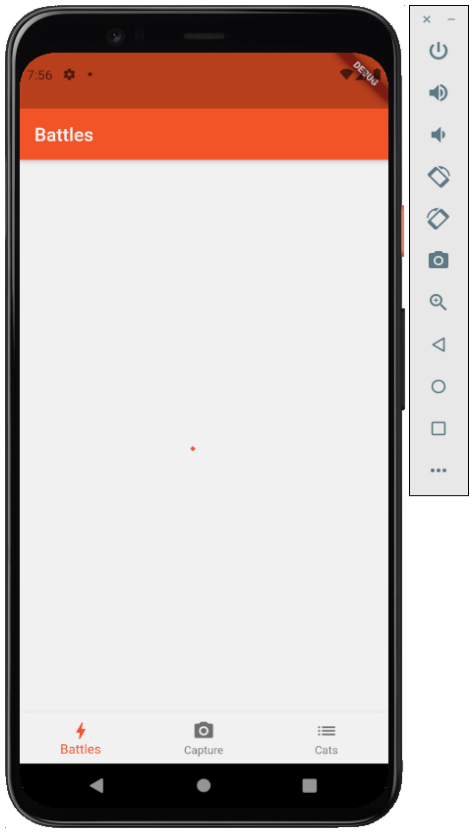
As a team we walked through these major requirements and decided that there is still no need to change or add any. These were the original requirements we created in assignment 4 and they are still very prevalent in our application at this stage. All five requirements are necessary and we feel that they sum up everything we hope to accomplish with this application.

b. For our analysis model walkthroughs, we decided to keep everything mostly the same. We feel that our models accurately represent what we are wanting to create. One thing that we did add to our activity diagram is more of the actual data representation during battles. We think that the data that will be collected during the battle simulation will be a hard part to implement, so we are trying to design it out as much as possible to make it easier on the programmers.

c. For our design model walkthroughs, we did make some UI changes. These will be described in detail in the next question, but it was mostly just changing some UI elements, not the overall design. In our block diagram, we did add a section for using an API that uses AI to identify what the picture is of. We will be using this to ensure that all photos uploaded are of cats and nothing else. This was originally part of our design, but we only added it in recently once we made sure it was a feasible goal.

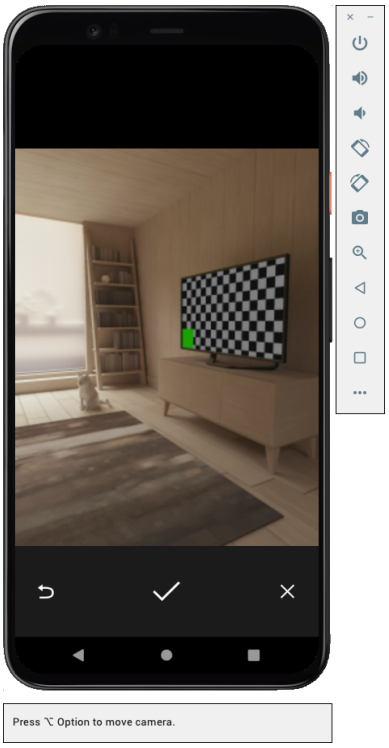
4. Overview of GUI:

We have made some significant changes since our last assignment to the GUI of our application. We changed the overall color scheme of the app from blue to orange. Here is the overview of how the app looks like now:

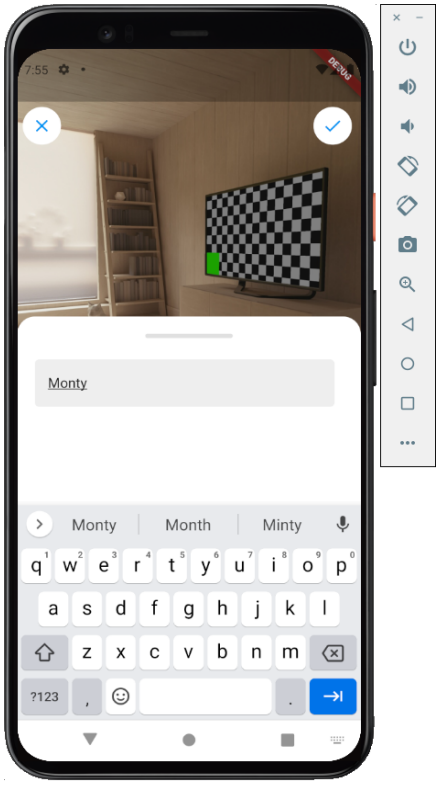


The bottom navigation bar is how users will traverse between the different sections. The three sections we have are Battles, Capture, and Cats. As of now, the battle screen just loads forever, but once it is working it will have two buttons to either battle a random opponent or search for a friend to battle.

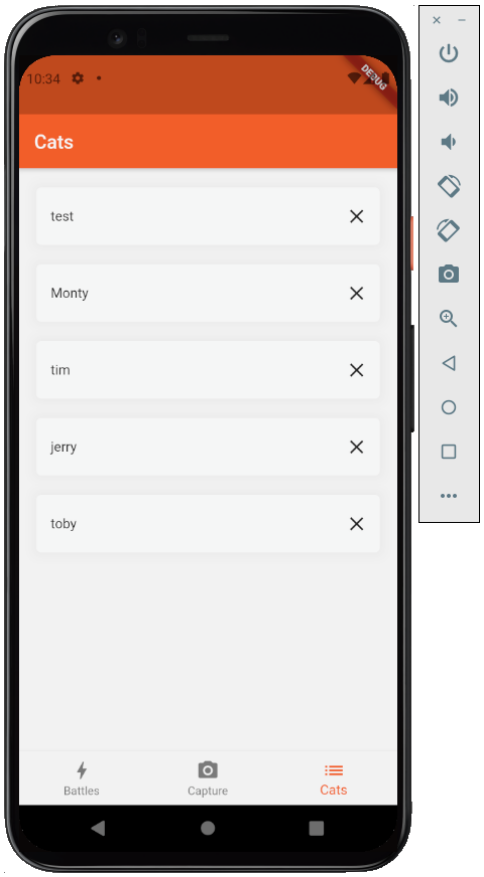
The capture page is currently functional on the app. A preview of what it looks like is below:



This is what will pop up when the capture button on the bottom navigation bar is clicked. The Android emulator has a fake AR area to take pictures in, so the picture is not of an actual cat but in practice this would be a picture of a cat. Once the picture is taken, this is where the API of making sure the picture really is a cat would be used. If the API ensured that this is a cat picture, then the keyboard and text field will pop up to allow you to give your cat a name. This is what that would look like:



Once you give your cat a name, it will be saved and added to the users personalized cats collection in Firebase and subsequently added to the Cats page. This is also where stats for the cat would be created and saved, however we have not yet implemented that. Below is a view of the Cats page:

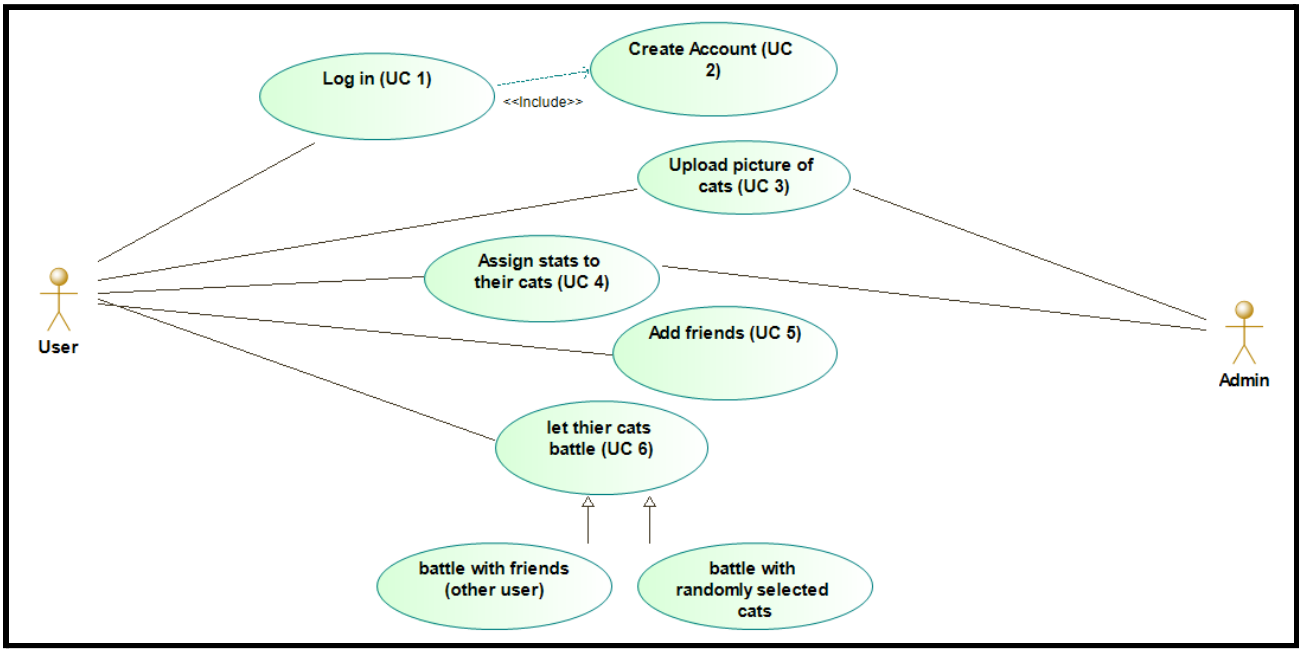


On this page, the cat data is pulled from Firebase and displayed to the user. Users will be able to click on each cat to see the cat's picture, stats, and attacks. Currently, it just grabs the cat names from Firebase and displays them.

5. Requirements:
    1. This application must allow for users to take photos of their cats and assign stats to them.
    2. Battle Cats must allow for users to battle other users' cats or randomly selected cats.
    3. This application must allow users to add friends and have real time battles with them.
    4. This application must use a real time database (Firebase) to allow for constant updates to the users view.

5. An admin shall upload cats' pictures and manipulate the stats, which are used in battle with randomly selected cats.
6. Users must have accounts and log in to play Battle Cats.

Use-case diagram:



Use-Case and Requirements Trace Analysis:

	Priority Weight	UC1	UC2	UC3	UC4	UC5	UC6
R1	3			X	X		
R2	2						X
R3	1					X	X
R4	2						
R5	2			X	X		
R6	2	X	X				
Score		2	2	5	5	1	3

Test Plan for requirements:
Since this is a mobile application, all the tests for the requirement can be done on mobile devices.
R1: Take a picture of a cat and non-cat objects. The app should correctly recognize a cat as a cat, and the app should reject non-cat pictures. Assign statistics to the recognized cats.
R2: Start a battle with selected cats or randomly selected cats.
R3: Search a friend by username and become friends with them. Battle with the friend.
R4: Start a battle. Once the battle is finished, statistics on a cat should be updated.
R5: Login as an admin role. Change statistics of cats and pictures. Once you save a change, the data should be updated.
R6: Log in with unregistered username, which should result in login failure. Create an account and login with the new account. Now the user should be able to play the game.

Code:

https://github.com/0llievr/BattleCat/tree/master

Team Plan/Roles:

Kyle Handy - Front end programming using Flutter

Oliver Lynch - Front end programming / UML Diagram Engineer

Eric DeMoney - Back end programming mostly working with Firebase

Ryota Haba - UML Diagram engineer / back end programming

Jacob - UML Diagram engineer

Tyler - UML Diagram engineer