

Functional Specification

MMA Fight Calculator

Sean Sinnott & Cathal Neary

0. Table Of Contents

1. Introduction:	Page 1
1.1 Overview:	Page 1
1.2 Business Context:	Page 2
1.3 Glossary:	Page 2
2. General Description:	Page 2
2.1 Product / System Functions:	Page 2
2.2 User Characteristics & Objectives:	Page 3
2.3 Operational Scenarios:	Page 3
2.4 Constraints:	Page 4
3. Functional Requirements:	Page 4
4. System Architecture:	Page 6
5. High-Level Design:	Page 8
5.1 Class Diagram:	Page 8
5.2 Sequence Diagram:	Page 9
6. Preliminary Schedule:	Page 10
7. Appendices:	Page 10

1. Introduction

1.1 Overview

Our goal in completing this project is to develop an algorithm using artificial intelligence which we will use to predict the outcome of Mixed Martial Arts (MMA) fights. This algorithm will use data about two fighters, such as their reach, height, weight, etc. to predict which fighter should win the fight and the level of certainty (as a percentage) that it forecasts this outcome occurring. MMA is a combination of numerous different martial art disciplines such as Boxing, Kickboxing, Jiu Jitsu, Wrestling and many others which time will not permit us to note. As well as the combination of different disciplines, there are copious amounts of ways a fight can end such as knockout, submission, injury, disqualification, referee stoppage, or judge's decision.

The factors mentioned above contribute to MMA being one of the most unpredictable sports in the world today and increases the complexity of our prediction algorithm. It is very difficult, as a human, to predict what will happen when, for instance, a boxer who is an expert at knocking people out fights a wrestler who is an expert at taking down and submitting opponents. This is where our algorithm will come into play. By considering more detail than a human would be capable of it can look past these varying fight styles to check other contributing factors that a human may never have considered. For this reason, having a systematic and scientific predictor at your fingertips will appeal to both avid and casual fight fans globally.

Once we have found to the best of our ability the most accurate algorithm, we will go about making it available to users via an Android application.

1.2 Business Context

Since MMA is rapidly becoming one of the most affluent sports in the world, there are a few potential business opportunities upon completion of this project.

- We could generate revenue by allowing advertisements to be placed within our Android application. This leads to the opportunity of releasing a “Paid Version” of the Application which, for a small fee, will provide the user with an advertisement free version forever. We plan on providing our application in several foreign languages to cater for countries in which MMA is popular but where English is not the native tongue, such as Brazil, Spain and Poland.
- Another possible avenue we could explore is to create a website on which we place predictions made by our algorithm in relation to upcoming MMA events. If we implement Search Engine Optimisation (SEO) techniques to climb up Google search results related to “MMA” we could generate good footfall. The more traffic we receive, the more money we would make from advertisements placed on the website.
- We could also attempt to gain interest from one of the major MMA Journalism Websites who could purchase our algorithm and place it on their website to attract users.

1.3 Glossary

Android: a mobile operating system developed by Google

Search Engine Optimisation (SEO): techniques used to increase the amount of traffic to a website by obtaining a high placement in search results if a search engine

SeekBar: an Android element that allows the selection of integer values using the interface. It has a thumb that can be slid from left to right in order to choose a value between 0 and a maximum value set by the developer.

Learning Classifier System (LCS): a machine learning method that combines a discovery component (i.e. genetic algorithm) with a learning component (i.e. supervised learning).

Genetic Algorithm: a heuristic search method used in A.I. used for finding optimised solutions to search problems based on the theory of natural selection.

2. General Description

2.1 Product / System Functions

The MMA Fight Predictor Application will have two basic modes for the user to choose from. The first mode will contain our algorithm. In this mode, the user will be able to pit two MMA fighters of their choice against each other and see a prediction from our algorithm based on the data it receives either from the user or from a database that we created. The second mode involves the user's opinion. After choosing the two fighters that they wish to match against each other, they will

be presented with a screen which will allow them to build their own prediction algorithm. This will be achieved through a series of 'SeekBar' which correspond to attributes such as weight, height reach, etc. and the user will slide from left to right based on how important they believe each attribute to be when predicting the outcome of the fight. For example, some people may be of the opinion that striking is more important than wrestling when it comes to predicting the outcome of a fight. By allowing the user to build their own algorithm, they can see how well they fair against ours and it this makes it a much more interactive experience.

2.2 User Characteristics and Objectives

Our Application will be hosted on the Google Play Store. The user will need both access to the internet as well as an Android Smartphone/Tablet in order to download it, and should have the basic knowledge to operate such a device. Our fight predictor is designed for, but not exclusively, fans of MMA. The objective of the user is to receive a prediction on the outcome of a contest between two fighters of their choice. Ardent fans will enjoy being able to customise the algorithm to suit their opinions/beliefs, while casual fans with less knowledge about the sport will enjoy having access to our algorithm's predictions.

2.3 Operational Scenarios

Scenario 1: *User receives prediction from our algorithm*

The user opens our Application and is arrives on the main menu screen. From here the user chooses to receive a prediction from our algorithm. They select the two fighters and then either enter data about the fighters manually or the data can be taken from a database (still unsure of this). The user will then receive a prediction, something along the lines of "Fighter A should beat Fighter B 65% of the time", and then be given the option to return to the main menu or get another prediction from our algorithm.

Scenario 2: *Prediction results in a draw*

The user has chosen to receive a prediction on the outcome of a fight between two fighters of their choice. The algorithm is executed, but the weighting of the attributes makes it so that neither fighter comes out as a clear winner. As such, the user is presented with a screen informing them that the algorithm could not find a winner and as such believes the fight is a 50/50 split percentage-wise.

Scenario 3: *User receives prediction from algorithm using their chosen weightings*

From the home screen, the user chooses to make their own algorithm using SeekBars. They are presented with a screen containing a list of attributes, each of which is connected to its own SeekBar representing the importance of that attribute. The user moves the SeekBars from the centre of the screen left or right based on how important they believe a particular attribute to be. Upon completion of this exercise, they confirm their selections and receive a prediction.

Scenario 4: *User's algorithm weighting results in prediction in excess of 100%*

After creating their own algorithm using their own weightings for each attribute they confirm their selections and choose to receive a prediction. When the algorithm is executed, one fighter's chance of beating the other has surpassed 100%. Our application will now display a message saying "Fighter A should beat Fighter B 99% of the time". We reduce the likelihood of the outcome back down to 100% so that it aligns with statistical rules and then take off an extra 1% to allow for the fact that anything can happen in an MMA bout – nothing is certain.

2.4 Constraints

Data Constraints – both developing the best algorithm and allowing users to match fighter's against each other requires access to data about MMA fighters. This is not really a problem for well-known fighters. However, information such as reach and height are much harder to find when it comes to lesser-known fighters who may be making their professional debut.

Time Constraints – the nature of finding an algorithm using Artificial Intelligence is essentially an ongoing process. There are infinite possible algorithms and even when we find "the best" algorithm, that is only the best algorithm in the space of algorithms that we have searched so far. There may be a better algorithm out there that we may never find, or we could find the best one on the first day, there is just no way of knowing. We intend to put forward the best algorithm that we can achieve based on the time constraints of the project.

3. Functional Requirements

3.1 Version of Android

Description – The application will require that the user's device is running with an Android Operating System, as the Application is developed using Android Studio and as such will not function on Apple or Windows devices

Criticality – Fundamental

Technical Issues – The user must be running an Android version of at least 4.2 in order for it to function

Dependencies - None

3.2 Implement an Algorithm

Description – The application must be able to use the prediction algorithm that we create or use the algorithm that the user creates when making a prediction

Criticality – Fundamental

Technical Issues – Must be able to change the algorithm's weights dynamically as the user changes weights based on their opinion

Dependencies – Must have a way of receiving data about the fighters. This data must be exact for the prediction to be as accurate as possible

3.3 Take in Data From the User

Description – The application must be able to take in data from the user in formats such as strings (names) and integers (height, weight, etc.)

Criticality – Fundamental

Technical Issues – Must be able to display a section for the user to enter information and be able to deal with a situation where the user enters the wrong type of information, i.e. a word (string) when asked for a fighter's age (integer)

Dependencies – None

3.4 Store Data

Description – The application must be able to store information entered by the user, such as a fighter's name, and then reuse that data on later screens/when predicting the outcome

Criticality – Fundamental

Technical Issues – Must have a global class as well as getters and setters for such variables. This is because Android Applications consider each screen a different activity with no memory of data entered on a previous screen

Dependencies – Must be able to receive the data in order to store it

3.5 User Interface Must Adapt to Different Screen Sizes

Description – With Android Applications, a user interface that looks good on a small phone may not convert well to large tablet screens as things like positioning and font may be hindered / broken, so we must account for this.

Criticality – High

Technical Issues – We will have to design multiple different sized User Interfaces to suit different screen sizes and then store them inside the application. Android will then apply whichever User Interface size most suits the device on which it is running

Dependencies – The Android Operating System because it handles the retrieval of device screen size and the displaying of the User Interface

3.6 Change of Language Application

Description – The application will have the ability to display text in numerous languages, allowing people in countries where English is not the native language to use it with ease.

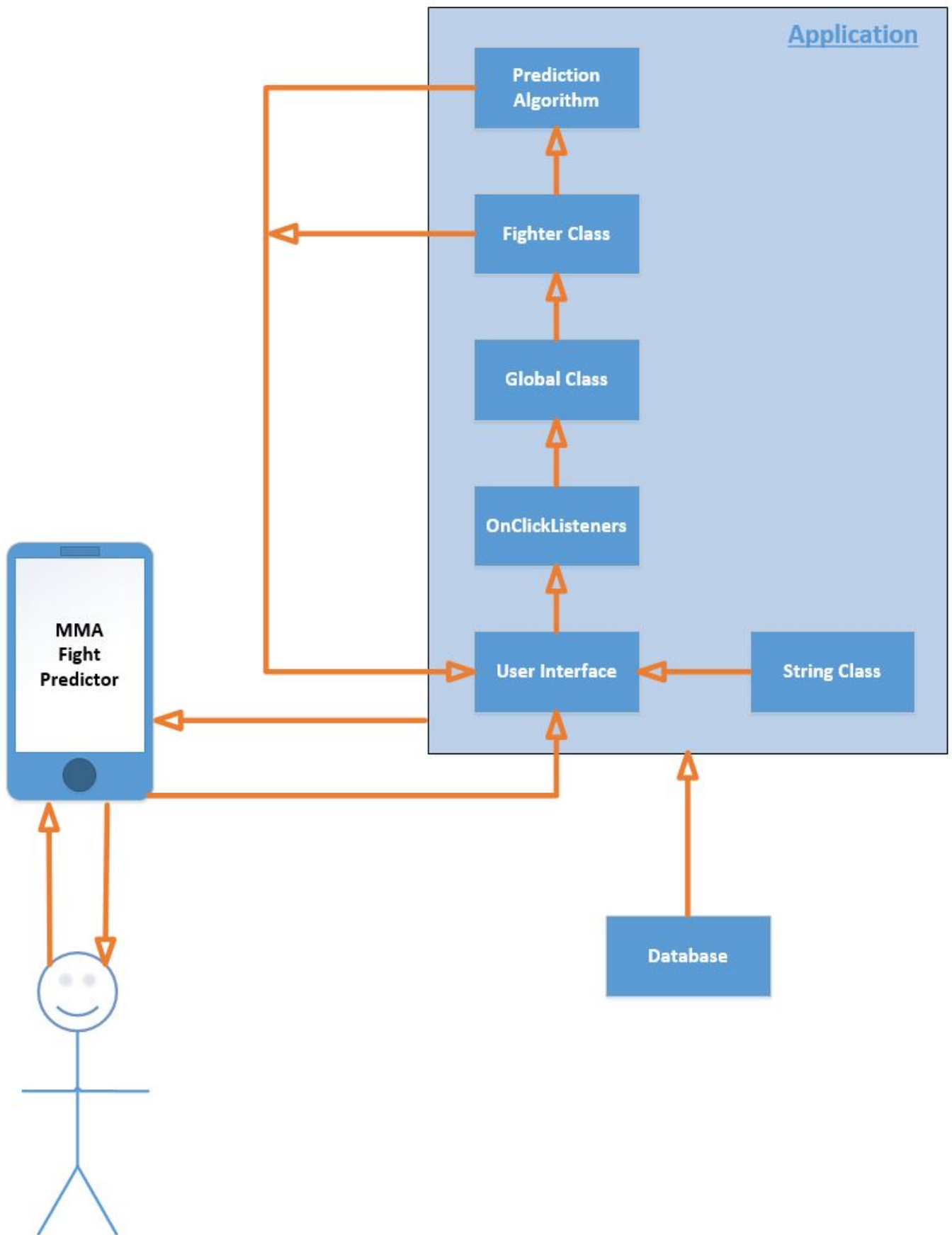
Criticality – Medium

Technical Issues – We will need multiple files containing strings that are used in the application, all

of which will be in a specific language. These files will be used to display text inside the application in whatever language the user chooses.

Dependencies - None

4. System Architecture



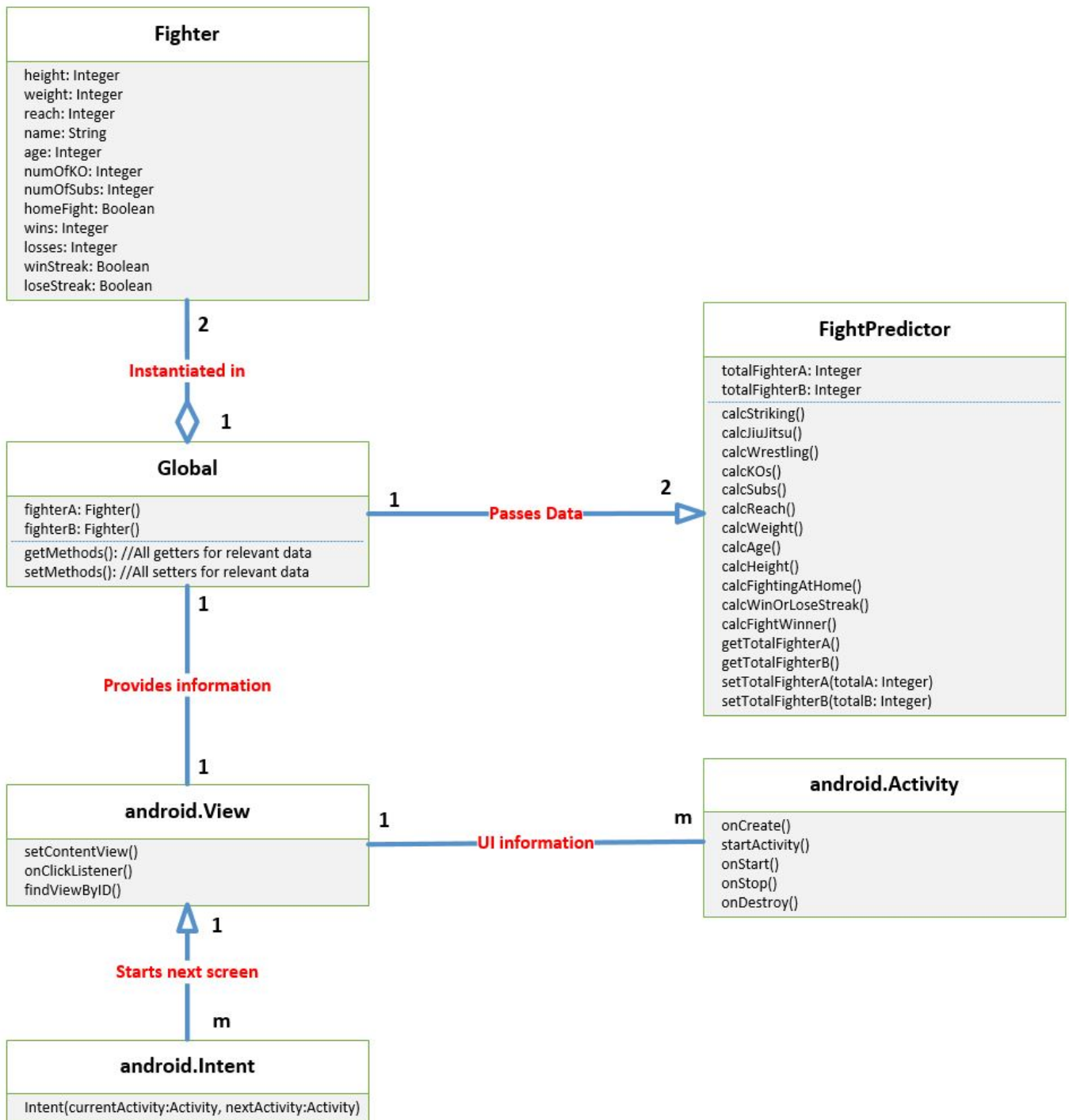
The system architecture of our project is relatively straight forward as it is all contained inside an Android application. As you can see in the above diagram, there is intercommunication between the user and our application via their Android device. The application displays the User Interface upon launch of the application. The messages displayed on the UI, which guide the user on how to

proceed, come from the string class which can vary the language it displays its text in depending on the user's settings and preferences. The application awaits input / activity through event listeners and appropriately responds to the user's requests. Information entered by the user is collected and stored in a global class which all activities can access. This allows the application to use and manipulate information gathered across various activities (screens). This information is used by our algorithm to predict the outcome of a fight and return the result to the user through the user interface. Optionally if we decide to include a database of our own it will reside somewhere alongside the application itself.

5. High-Level Design

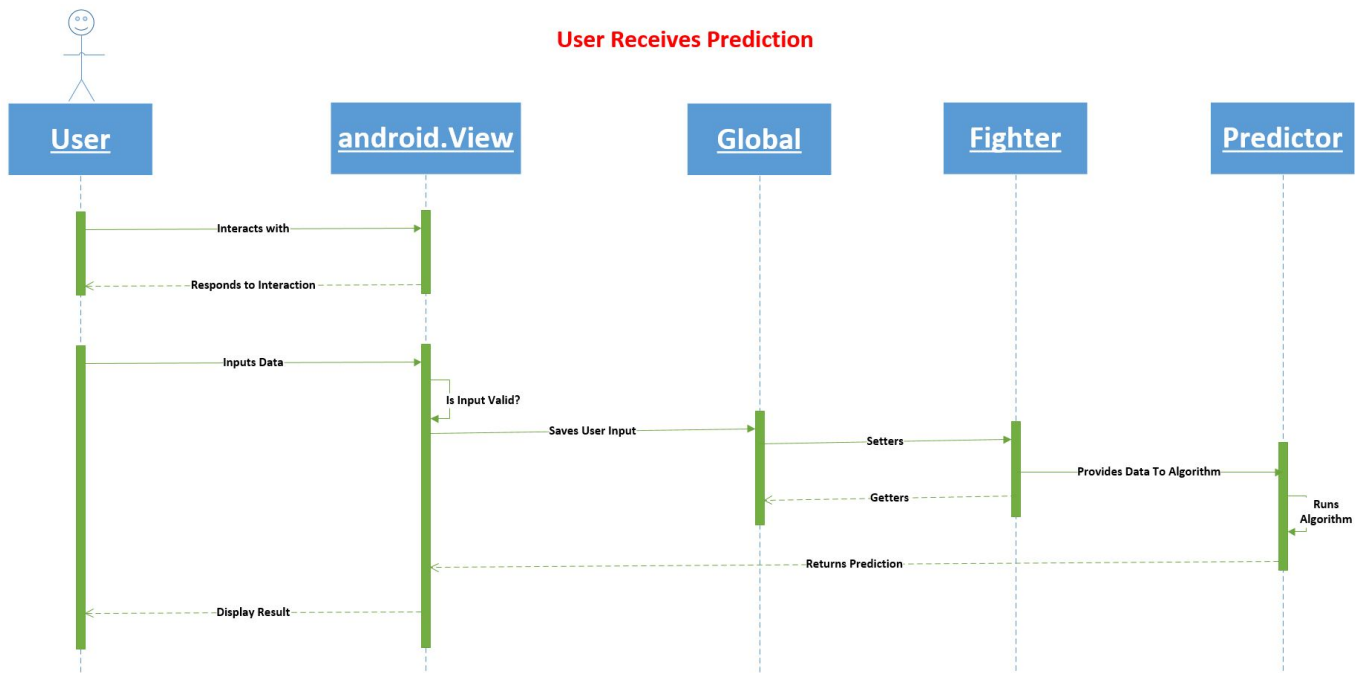
5.1 Preliminary Class Diagram

The following class diagram portrays a preliminary design of the system by showing the system's classes, their variables, methods, and the relationships between objects.



5.2 Preliminary Sequence Diagram

The following sequence diagram shows how, and in what order, objects in our system communicate with one another when a user is receiving a prediction.



6. Preliminary Schedule

Here, we provide a GANTT Chart which outlines the schedule which we aim to follow for the entirety of the project.

ID	Task Name	Start	Finish	Duration	Dec 2016				Jan 2017				Feb 2017				Mar 2017	
					4/12	11/12	18/12	25/12	1/1	8/1	15/1	22/1	29/1	5/2	12/2	19/2	26/2	5/3
1	Build Sample Space of Algorithms	02/12/2016	12/12/2016	7d														
2	Find Best Algorithm in Space	12/12/2016	02/01/2017	16d														
3	Study For / Sit Semester 1 Examinations	02/01/2017	23/01/2017	16d														
4	Explore Learning Classifier Systems	23/01/2017	27/01/2017	5d														
5	Implement Learning Classifier System	27/01/2017	10/02/2017	11d														
6	Work On Android Application	10/02/2017	02/03/2017	15d														
7	Implement Algorithm in Application	10/02/2017	17/02/2017	6d														
8	Research Database for Fighter Data	17/02/2017	21/02/2017	3d														
9	Implement User-Algorithm Builder	21/02/2017	27/02/2017	5d														
10	User Interface / Bug Fixes	27/02/2017	02/03/2017	4d														
11	Technical Specification / User Manual / Video walkthrough	02/03/2017	09/03/2017	6d														
12	Submit The Project	09/03/2017	10/03/2017	2d														
13	Prepare for Demonstration	10/03/2017	13/03/2017	2d														

7. Appendices

Here is a link to the official Android website which is the mobile operating system that is required to run our application.

[Click here to go to the Android website](https://www.android.com/)