Project Development Report

**Client Name : Salaheddin Alakkari**

**Project Name : Deep Learning Model to Detect Heart Arrhythmia in ECG Data**

**Group 32**

**Pascal Raos, 3rd Year ICS**

**Maryann Foley, 3rd Year ICS**

**Anton Tiscoschi, 3rd Year ICS**

**Ailbhe Merriman, 3rd Year CSB**

**Kevin O’Donnell, 2nd Year CSB**

**Minjuan Luo, 2nd Year ICS**

# **Zhongyuan Liu, 2nd Year ICSContents**

**1. Introduction**

*1.1.* [*Background & Problem Statement*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.up9t2l9fzr49)

*1.2.* [*Technical Approach*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.yao40i86rcge)

**2. Requirements**

*2.1.* [*Functional requirements*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.qai90n3az5gf)

*2.2.* [*Non-functional requirements*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.zdh4rs203fdx)

*2.3.* [*User Interaction Scenarios*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.72fgu6ts5eay)

**3. Design**

*3.1.* [*Architecture Diagram*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.y1x7631kz643)

*3.2.* [*UML Class and Sequence Diagram*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.qnnxtuejpo0z)*s*

**4. Implementation**

*4.1.* [*Tools, Libraries, Platforms*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.heo5df7ga9l9)

*4.2.* [*User Interfaces*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.yk67m6vy7ge)

*4.3.* [*Algorithms*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.jgyrgnbnb176)

**5. Conclusions**

*5.1.* [*Design and Implementation*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.4982y8x6xgrh)

*5.2.* [*Project Objectives*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.s5m4ufqx8gyi)

*5.3.* [*The Team*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.f0d36gt5ogg2)

*5.4.* [*Algorithms*](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.co2qxlxgnyo6)

**6. Appendices**

*6.1. Appendix 1*

*6.2. Appendix 2*

# Introduction

## 1.1. [Background & Problem Statement](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.up9t2l9fzr49)

*What is an ECG?*

An electrocardiogram (ECG) is a simple test that can be used to check your heart's rhythm and electrical activity. Sensors attached to the skin are used to detect the electrical signals produced by your heart each time it beats. These signals are recorded by a machine and are looked at by a doctor to see if they're unusual. An ECG may be requested by a heart specialist (cardiologist) or any doctor who thinks you might have a problem with your heart, including your GP. We will be analysing the signal pattern results that came as a result from this ECG reading.

In this group project we want to predict different arrhythmia using ECG. The database that we are using, ‘The MIT-BIH Arrhythmia Database’ contains 48 half-hour excerpts of two-channel ambulatory ECG recordings. These results were obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Twenty-three recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital. The remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample. (Arrhythmia on ECG Classification using CNN, 2022)

## 1.2. [Technical Approach](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.yao40i86rcge)

*General:*

Each team member took on individual roles within the group: Pascal Raos from third year acting as Team Lead and lead on Client Communication, he's also responsible for the AI group and helping solving issues. Ailbhe Merriman from third year and Minjuan Lou from second year were AI developers who implemented various deep learning models. Maryann Foley from third year is the leader of the UI group. Anton Tiscovschi from third year, Kevin O’Donnell from second year and Zhonguan Liu from second year are frontend developers and they are responsible for building up the front end environment. The two group leaders are responsible for the connection between back end and front end.

For version control and development we used GitHub. As we worked on different features or tested different things, we made different branches which we would then merge into our master branch after being reviewed. This allowed us to maintain a high coding standard and allowed us to keep our client up to speed on what was being worked on and the different adversities we were encountering.

*Front-End Specific:*

The technical approach of the front-end was geared towards displaying the estimated graphs of heartbeat type for the client in a way that was easy to understand and didn’t require them to do anything complicated. To achieve this, we used Heroku to build an intuitive user interface that allowed the user to navigate their local folder and input various amounts of ECG data in order to obtain analysed results. We are using the Numpy and Pandas in order to format inputs from the CSV files. Building the interface with Heroku required the use of HTML and Python through bootstrap which was backed up by the Flask application. The data displayed by the graphs was provided by the backend server in the form of a JSON. The front-end received that data, input it into a different deep learning AI model and then displayed the graph in the user-interface for the client to view.

*Back-End Specific:*

The approach for the back-end was split into four parts based on using Keras and Tensorflow deep learning functions along with numpy and pandas for input formatting. Firstly, reading in the data with resampling method and class weight method, both act on CNN model using convolutional layer, activation layer and then a pooling layer to self learning .(decided to use Sequential API other than Functional API) The third part is using another deep learning method called LSTM model which develops by using different layers as well. The last part is the ResNet model which is based on the F(x) + x formula. Once the final result is generated by the selected model, the data is then directly piped to the front-end where it is displayed using a graphing library.

# [Requirements](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.34rzl2oiciqh)

## 2.1. [Functional requirements](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.qai90n3az5gf)

The system should be able to do the following:

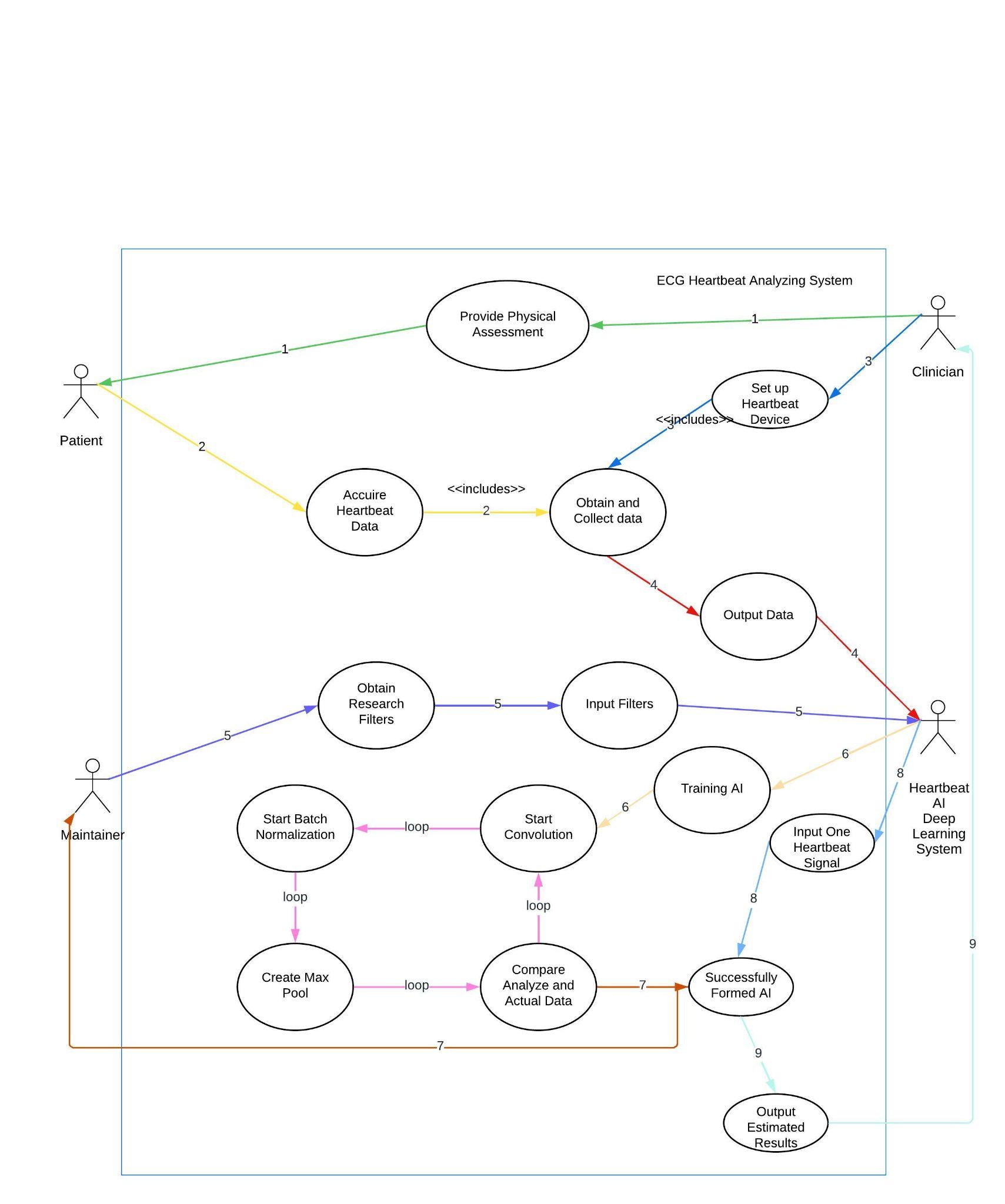
* Allow user to select different types of deep learning models to analyse their input
* Functioning CNN Model
* Functioning LSTM Model
* Functioning ResNet Model
* Each input of ECG data is be sorted into 5 different types of heartbeat patterns
* A User Interface that allows user to browse their own machine and select the desired CSV file to input the ECG data
* A selection box to change the deep learning model that is going to analyse the data
* An output form that displays all the results of the inputted ECG data from the deep learning model as well as the model accuracy

## 2.2 [Non-functional requirements](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.zdh4rs203fdx)

* *Easy to use:* The layout and design should be easy to follow for patients and clinicians to use
* *Maintainability:* The inputted ECG file does not need to be well structured to allow for deep learning model to analyse the data accurately
* *Web-based:* Portability is required to allow users to use the system in any location.
* *User Registration:* The system will have a front page that will require the patient or clinician to log in before starting to use the deep learning model. A user who has not registered in the system will be blocked.

## 2.3. [User Interaction Scenarios](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.72fgu6ts5eay)

The use case diagram shows the most important user interaction scenarios that will take place within our system. The main actors of the system are the clinician, the patient and the model itself. The diagram shows how they interact and what steps are taken for our system to function correctly.



# [Design](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.b2l8rb1tc1xz)

## 3.1. [Architecture Diagram](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.y1x7631kz643)

## Architecture diagrams abstract our software systems and their relationships to each other. As you can see, the back end gets user input data from the front end. The AI system then manipulates the data, and the user interface displays visual data corresponding to that content.

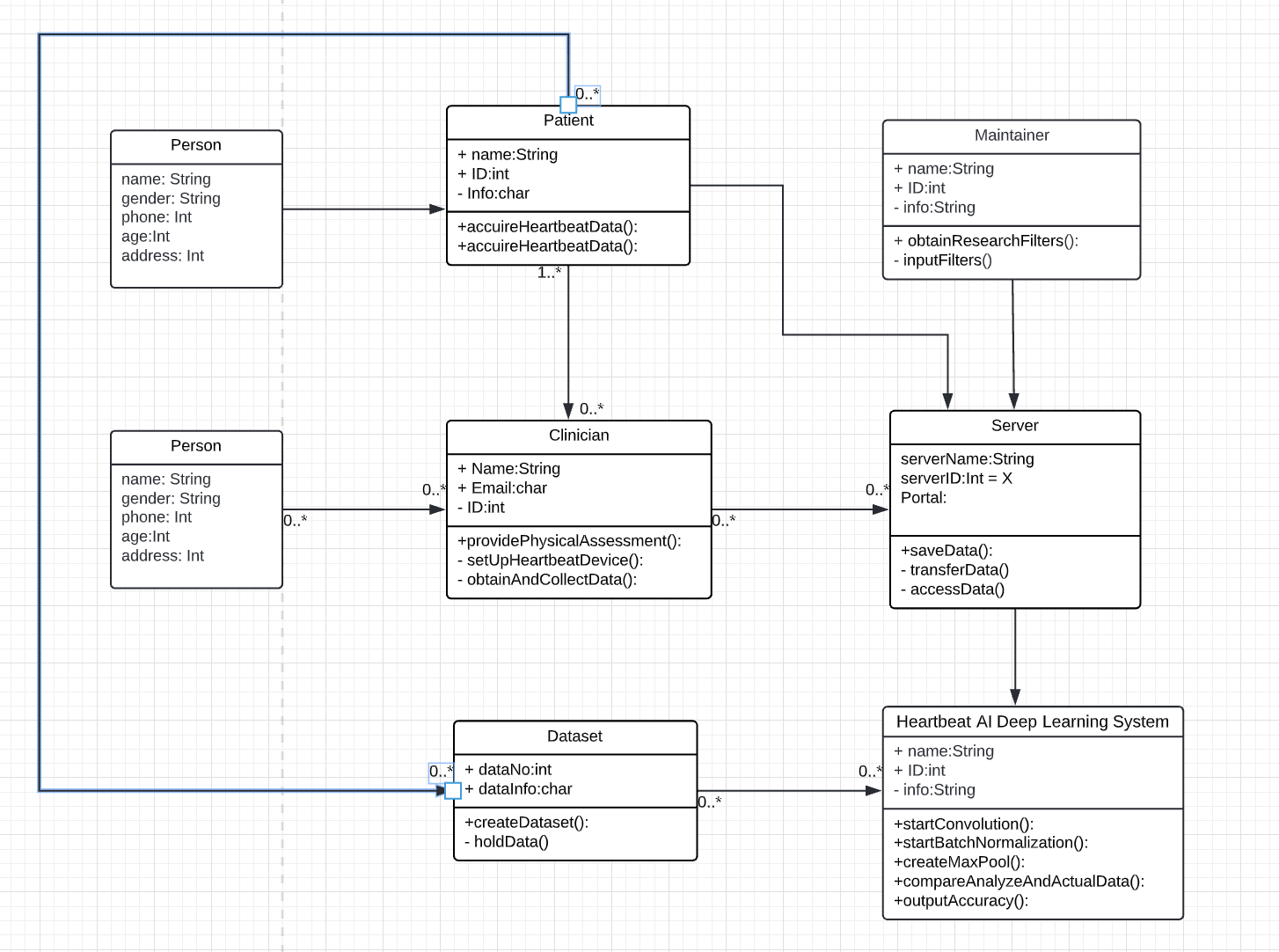
## 

## 

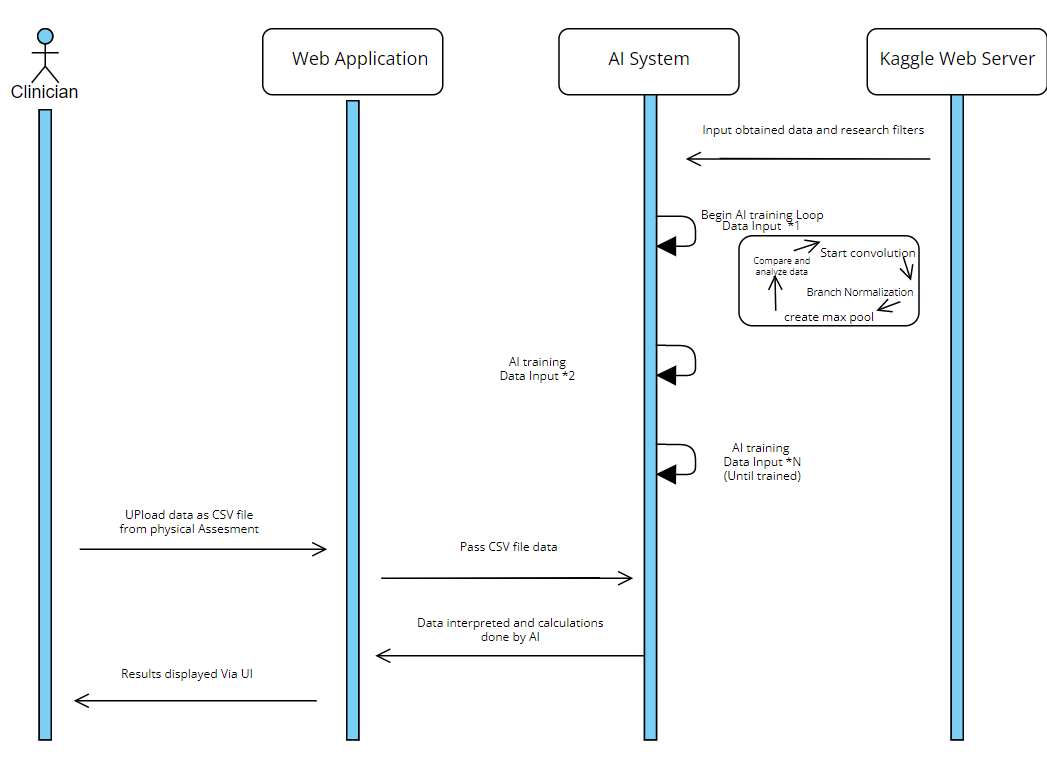
## 

## 3.2. [UML Class and Sequence Diagrams](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.qnnxtuejpo0z)

*The class diagram* shows a representation of how our two main deliverables, The AI deep learning system, and the AI interface may interact within a healthcare system similar to our last project with Vertigenius. The dataset is pushed into the interface held on a web server, where it is passed behind the scenes to be assessed by the AI.



*The sequence diagram* shows how objects in a system interact with each other. When the user requests visualisation, the application calls the AI system, which then calls the Kaggle Web Server to get the data. The data is then sent to the AI system, which takes the data and runs it and draws conclusions. The results are then returned to the Web application, which then returns a visualisation of the corresponding content



# [Implementation](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.gfdvc2nl3dn)

## 4.1. [Tools, Libraries, Platforms](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.heo5df7ga9l9)

**Tool/Library/Platform:** Visual Studio Code

**Description:** A code editor used to write and debug code .

**Advantages:** Contains multiple useful features: intelligent code completion and connection to github.   
**Disadvantages:** Not an IDE.

**Tool/Library/Platform:** Github

**Description:** Code hosting platform, allowing for version control and collaboration.

**Advantages:** Allows multiple developers to work at once, provides good backup, and is free.  
**Disadvantages:** Most members were unfamiliar with it and sometimes faced merge conflicts.

**Tool/Library/Platform:** Zoom

**Description:** A free messaging and video calling platform.

**Advantages:** Allows for separate channels, pining of messaging ,screen sharing and video calls.  
**Disadvantages:** It's only free for 40 minutes.

**Tool/Library/Platform:** Microsoft Teams

**Description:** A messaging and video calling platform with Client.

**Advantages:** Professional.  
**Disadvantages:** Not so commonly used and required email invite links.

**Tool/Library/Platform:** Python

**Description:** Object-oriented and functional programming language used for the front-End, and help to connect the AI system.

**Advantages:** Has a vast set of libraries, is readable, and is easy to use.  
**Disadvantages:** Execution time is slower for web-frameworks.

**Tool/Library/Platform:** Bootstrap,HTML,CSS

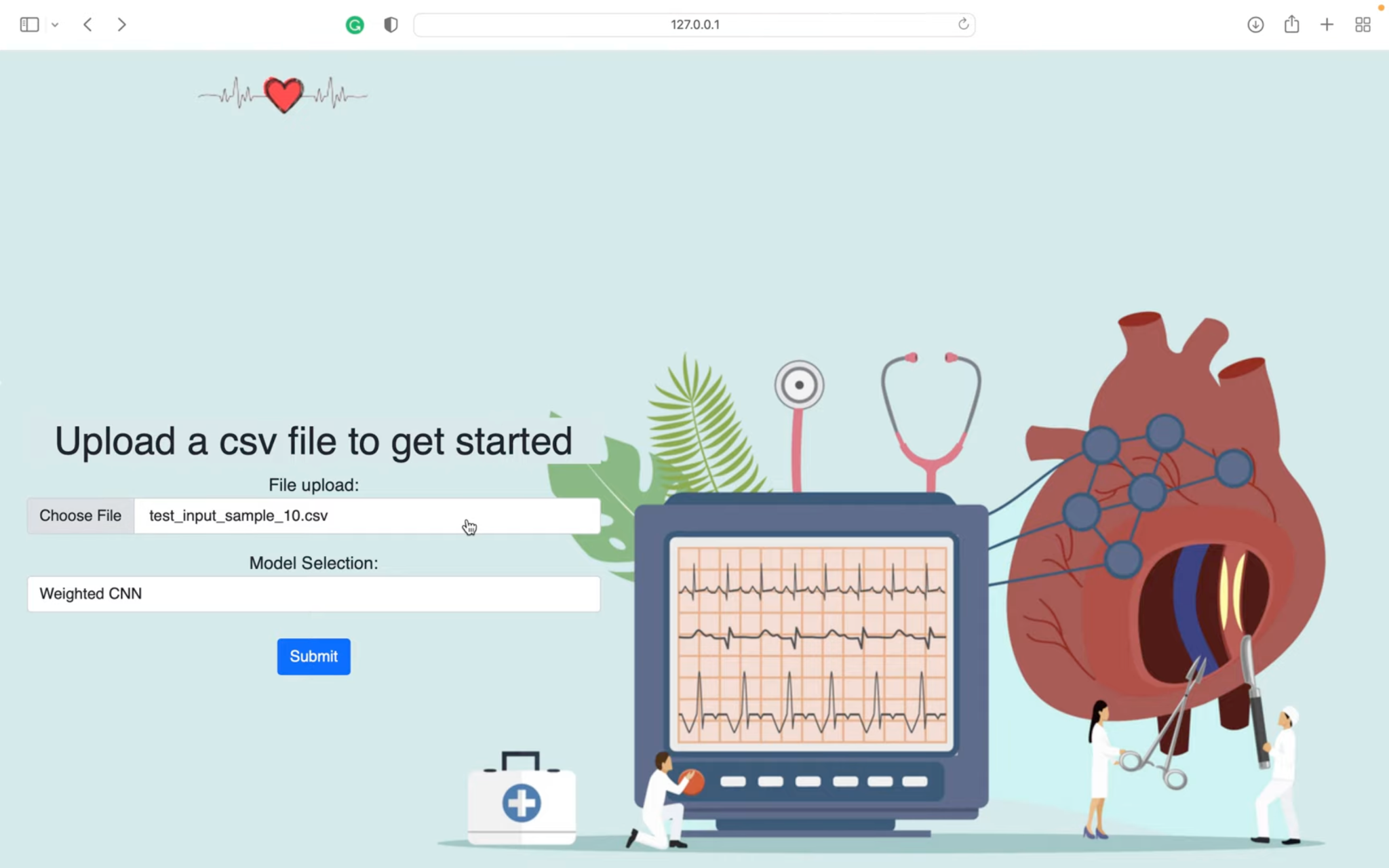
**Description:** The mixture of tools used to implement the layout and look of the front -end.

**Advantages:** Allowed for a lot of customisation to personalise the look of the website.  
**Disadvantages:** Had to be learned from scratch by the front end developers.

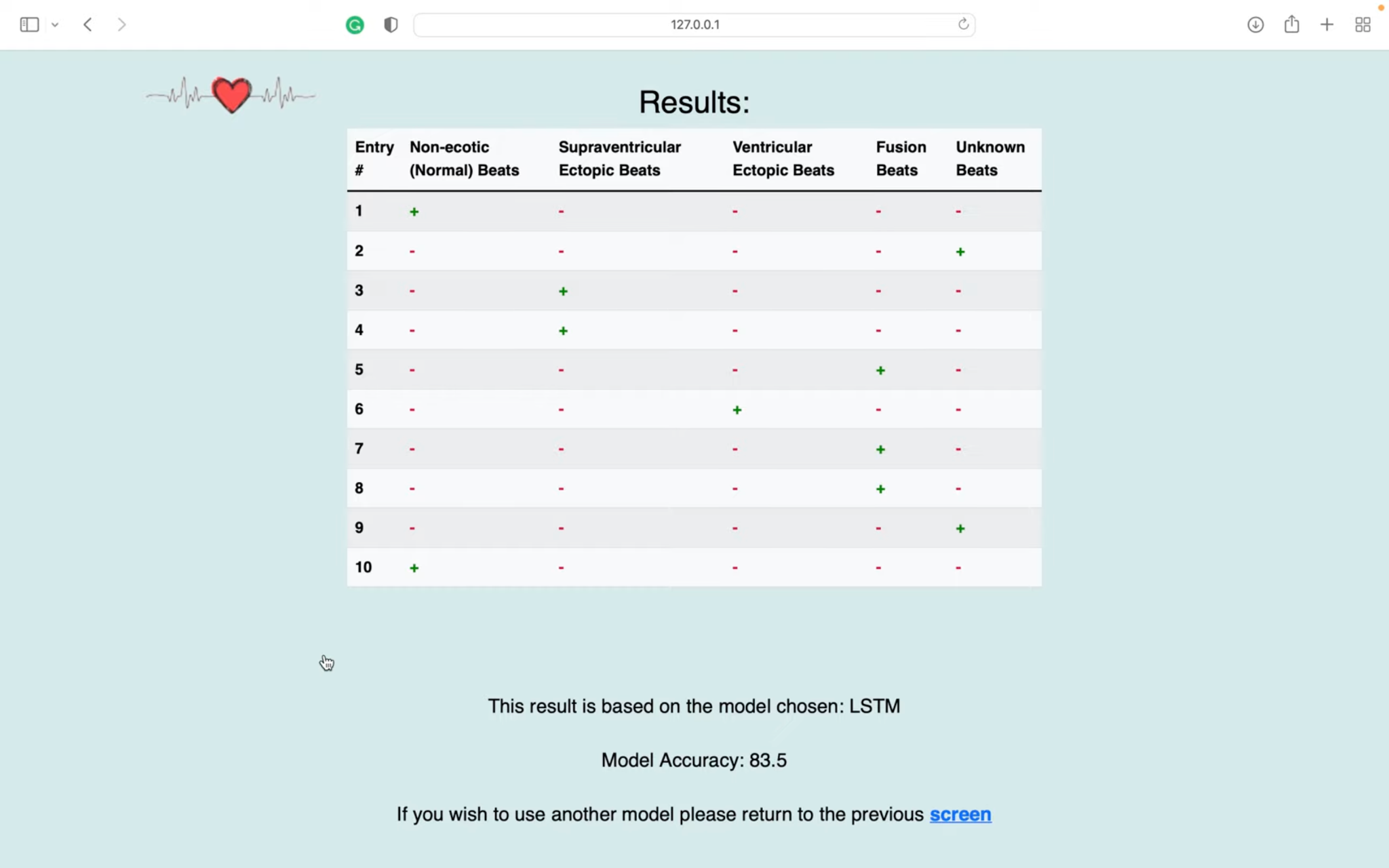
## 

## 4.2. [User Interfaces](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.yk67m6vy7ge)

We strive to make a simple, fresh and advanced UI interface. A simple and clear interface can better highlight the functions of our project, because the most important part of our project is the graphical visualisation of the AI system. We designed two pages, one being the input page and the other the results page. The input page clearly displays two input fields to the user. The first input field is responsible for the user to upload the relevant CSV file from their machine. The second input field is used to select different models for the AI system evaluation data. After the two fields have been dealt with by the user, they then submit their inputs through the ‘submit’ button. Then the AI system will run and produce results in the results screen Here you can see a table in the middle of the page that shows the results of the AI system. The corresponding text will appear below the table, giving the conclusion and result succinctly. There is also a back button at the bottom that takes you back.



## 



## 4.3. [Algorithms](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.jgyrgnbnb176)

The AI group created three AI deep learning models in this project which are outlined and described below.

*CNN model:*

CNN model also known as convolutional neural networks (CNN). A convolutional network ingests multi-channeled images as three separate strata of colour stacked one on top of the other. A normal colour image is seen as a rectangular box whose width and height are measured by the number of pixels from those dimensions. The depth layers in the three layers of colours (RGB) interpreted by CNNs are referred to as channels.

The first layer in a CNN network is the Convolutional Layer, which is the core building block and does most of the computational heavy lifting. Data or an image is convolved using filters or kernels. Filters are small units that we apply across the data through a sliding window. Second is the Activation Layer which applies the ReLu step(Rectified Linear Unit), in this step we apply the rectifier function to increase non-linearity in the CNN. Lastly, is the Fully Connected Layer, which involves flattening. This involves transforming the entire pooled feature map matrix into a single column which is then fed to the neural network for processing. With the fully connected layers, we combined these features together to create a model.

*LSTM model:*

LSTM model also known as long short-term memory (LSTM). LSTMs use a series of ‘gates’ which control how the information in a sequence of data comes into, is stored in and leaves the network. There are three gates in a typical LSTM; forget gate, input gate and output gate. These gates can be thought of as filters and are each their own neural network.

The first step in the process is the forget gate. In this stage we will decide which bits of the cell state (long term memory of the network) are useful given both the previous hidden state and the new input data. The forget gate decides which pieces of the long-term memory should now be forgotten (have less weight) given the previous hidden state and the new data point in the sequence. The next step involves the new memory network and the input gate. The goal of this step is to determine what new information should be added to the network's long term memory , given the previous hidden state and new input data. We can now move to the final step, the output gate, deciding the new hidden state .The step-by-step process for this final step is as follows: Apply the tanh function to the current cell state pointwise to obtain the squished cell state; pass the previous hidden state and current input data through the sigmoid activated neural network to obtain the filter vector; Apply this filter vector to the squished cell state by pointwise multiplication; output the new hidden state.

*ResNet model:*

The ResNet model is also known as The Residual Network (ResNet). The ResNet model has something called Residual blocks and many Residual blocks are stacked together to form a ResNet. The model has “skipped connections” too which are the major part of ResNet. The idea is to connect the input of a layer directly to the output of a layer after skipping a few connections. ‘X’ is the input to the layer which we are directly using to connect to a layer after skipping the identity connections and if we think the output from identity connection to be F(X). Then we can say the output will be F(X) + X.

The first step on the ResNet before entering the common layer behaviour is a block — called Conv1 — consisting of a convolution + batch normalisation + max pooling operation. The next step is the batch normalisation, which is an element-wise operation and therefore, it does not change the size of our volume. Finally, we have a Max Pooling operation with a stride of ‘X’. We can also infer that they first pad the input volume, so the final volume has the desired dimensions.

# [Conclusions](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.fo47mbtercb4)

## 5.1. [Design and Implementation](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.4982y8x6xgrh)

*Front-end Specific*

Overall in the front-end there were very few issues. The implementation of our aims were quite straightforward, especially with the peer-programming approach taken. One slight issue we encountered was when a big file was encountered. When this occurs the output table cuts off and not all of the lines in the table are visible.We were not quite sure what caused originally this but eventually through research it was rectified using one line of code. This issue besides the front-end team encountered very few, if any major issues.

*Back-end Specific*

The back-end team had a couple more issues than that of the front-end team. There were issues originally with setting up the Keras and Tensorflow environments but this was solved collaboratively so once one of the team members was able to set up the environment successfully they were then able to aid the others. Replacing the training method from functional API to sequential API also caused issues but through trial and error, and thorough research the issues were resolved. An issue also arised implementing the LSTM model due to the low accuracy of the model but it was solved by deleting unnecessary inputting formats.

## 5.2. [Project Objectives](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.s5m4ufqx8gyi)

Originally our objectives were quite simplistic. After the client change we were not overly confident in what we could aim towards. Our initial aim after our second first client meeting was to implement at least one accurate AI model that would be accessible to use through a clear and easy-to-use user interface. As the weeks progressed we became confident in our work and then extended our objectives into multiple AI models from which the user could choose from, all with different accuracies.

We talked about the potential to incorporate smart watch ECG readings into our models and perhaps create an application for mobile that could connect to these smart watches directly. This would allow us to have a very direct use for the model’s predictions. However, we decided that due to the unfortunate circumstances that occurred with our first client, the limited time remaining would be unsuitable to create a complex application such as that.

## 5.3. [The Team](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.f0d36gt5ogg2)

As a collective we are incredibly proud of our work, but not only our work. Our teamwork and communication are also something we feel we can boast about. Even including the rocky first six weeks of our project, we always communicated well allowing us to work in tandem. A key reason for this was our frequent meetings in which punctuality and attendance were impeccable. As a group we decided to meet twice a week, on Tuesdays at 1pm with all group members and on Wednesday with our client Salah. We would record our SCRUM meetings on Friday after discussing our goals for the coming week. We would strategically assign tasks to each other based on our strengths but also our weaknesses to learn something new and from one another. Each member of the team would work on their tasks for the week and we would reach out to one another if we ran into any obstacles to resolve any issues. Our team-roles were also an active part of our success. They allowed us to focus on our own specific aspect of the project and made communication much simpler.

## 5.4. [Algorithms](https://docs.google.com/document/d/1YxeipmpYOYHiVqo_92XtYbHbHvsLRlXB/edit#heading=h.co2qxlxgnyo6)

As previously mentioned, TensorFlow and Keras were used for the training of deep learning models and predicting the heartbeats of input data while Numpy and Pandas were used for formatting inputs from the csv files.

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It is an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning.

TensorFlow 2 is an end-to-end, open-source machine learning platform.It can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations.

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

***Bibliography***

* Kaggle.com. 2022. *Arrhythmia on ECG Classification using CNN*. [online] Available at: <https://www.kaggle.com/code/gregoiredc/arrhythmia-on-ecg-classification-using-cnn/notebook> [Accessed 13 April 2022].

# Appendices

## 6.1. *Requirements Document. p.g16*

6.2. *Software Design Specification Document. p.g28*

## 6.1

Requirements Document

*Group 32*



**Pascal Raos, 3rd Year**

**Anton Tiscoschi, 3rd Year**

**Ailbhe Merriman, 3rd Year**

**Maryann Foley, 3rd Year**

**Minjuan Luo, 2nd Year**

**Kevin O’ Donnell, 2nd Year**

**Zhongyuan Liu, 2nd Year**

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at [http://www.tcd.ie/calendar](http://www.tcd.ie/calendar/).

I have also completed the Online Tutorial on avoiding plagiarism ‘Ready Steady Write’, located at<http://tcd-ie.libguides.com/plagiarism/ready-steady-write>.

# **Contents**

**1. Introduction**

1.1. Overview - Purpose of system

1.2. Scope

1.3. Objectives and success criteria

1.4. Definitions

**2. Current system**

**3. Proposed System**

3.1. Overview

3.2. Functional Requirements

3.3. Non-functional requirements

3.4. System prototype (models)

3.4.1. User interface mockups

3.4.2. Use cases (including text narratives)

3.4.3. Object model

3.4.4. Dynamic model

## 1. INTRODUCTION

### 1.1 OVERVIEW - PURPOSE OF SYSTEM

People often take for granted the simplicity of their issues in life. While the common person may have shortcomings at work, difficulty in school or an abrasive relationship, some people struggle to stand up straight or even walk to their work. Vestibular balance disorders take a number one priority in the lives of those that have them. The vestibular system is also known as the inner ear and when affected by disease or injury causes dizziness, vertigo and imbalance. Individuals with vestibular disease avoid moving their heads as it makes them feel dizzy, and this in turn hinders recovery.

What Vertigenius are trying to do is improve patients’ vestibular balance disorders through physical exercises (also known as vestibular rehabilitation), and by monitoring their performance with a wearable head sensor. Through the use of mobile and web applications, they can gauge a patient’s improvement over time.

Our goal in this project is to provide Vertigenius with an automatic analysis of patient data in order to assess each individual's fall-risk. We plan to work in tandem with the client in order to provide accurate analysis of the data collected.

### 1.2 SCOPE

The software itself will be a standalone application that acquires Vertigenius patient data using AWS API calls to their database. We will then format and extract necessary data for our evaluation, and through the use of intelligent analytic algorithms, determine a fall-risk measure for the patient at hand. This specific value will then be able to be accessed by the Vertigenius web portal through subsequent API calls.

### 1.3 Objectives and Success Criteria

The System can be regarded as successful if it meets the requirements of the client, that is to create an application that will be able to accurately predict the fall risk of a patient as well as further improve data gathering of the web application by adding more questions into their Surveys.

Objectives:

* Create a sophisticated algorithm to evaluate fall risk of patients
* Add questions into survey to enhance remote data collection
* Conduct research on correlation between fall risk and tests conducted on patients to better understand which data carries more weight
* Effectively distribute workload base on the data that we receive
* Deliver an end product to the client

### 1.4 Definitions

The following definitions are relevant to the functionality we will be adding to the questionnaire:

* Vestibular System - The sensory system also known as the inner ear which helps to provide a sense of balance and spatial orientation. Diseases which affect the vestibular system can negatively impact a person’s sense of balance and lead to increased risk of falls.
* Dynamic Visual Acuity (DVA) - A test in which a static visual Acuity is firstly determined (their ability to distinguish details at a given distance by reading progressively smaller letters at a given distance on a chart known as a Snellen or ETDRS chart), and then repeats the task the task while rotating their head at a provided speed. Dynamic visual acuity is the degradation of the ability to see the same sized letters as during the static task and is indicative of a loss of function of the inner ear which allows an individual to see clearly when the head is moving.
* Timed 10 meter walk - A measurement of walking speed in metres per second. Gait speed can be an indicator of vestibular performance.
* Dynamic Gait Index (DGI) - A set of tests which measure the ability of patients to walk under various circumstances. Vertigenius are specifically using the 4 Item Modified DGI, which tests walking ability while: 1, turning the head horizontally, 2, turning the head vertically, 3, walking on a level surface, and 4, walking with speed changes.
* Dizziness Handicap Inventory (DHI) - A self assessment in which a patient answers questions about the effects of their dizziness on their lives. They answer questions, such as “Does your problem interfere with your job or household responsibilities?” on a scale of “Always”, “Sometimes”, or “Never”. Vertigenius uses the Dizziness Handicap Inventory short form (DHIsf), which consists of 25 questions out of the long-form’s 100 questions.
* Numerical Rating Scales (NRS) - Numerical Rating Scales are similar to the Pain Scale used by healthcare professionals, in which a patient rates the severity of a symptom on a scale of 0 to 10. The NDS asks about levels of dizziness, imbalance, nausea, anxiety and oscillopsia (visual blurring during head movement) on a daily basis, getting the range of the best and worst symptoms that day and symptom level at the time of the survey.
* Activities Balance Confidence Questionnaire (ABC Scale) - A self assessment in which a patient answers 16 questions about their percentage confidence in performing activities without losing their balance. Questions include ability to walk up stairs, get into a car, and other daily activities. A percentage score is derived from the average of the questions.

## 2. Current System

There is no fully functioning system in place at the moment for Vertigenius. The current system allows a clinic to sign up patients, these patients are then sent an automated email where they will create their account and fill in a questionnaire for the clinician. This allows the clinician to gauge what the patient is currently going through and what they aim to achieve from signing up to this service. This questionnaire is answerable through a never/sometimes/always domain with each question being scored at 0, 2 and 4 respectively. The higher the figure, the higher the severity of this case.

While the current system does calculate some individual fall-risk correlated measures, it does not currently have any sort of automated analysis of them for a broad overview of their total estimated fall risk.

## 

## 3. Proposed System

### 3.1 Overview

The proposed system is an application that uses data gathered from the Vertigenius web application surveys which are filled out by patients to predict their risk of falling.

This will allow the user (clinician) to gain useful insight on what kind of treatment a patient requires and to assess efficacy of treatment. The data used within the project we will pull from the API, which will be transformed into a JSON file. Our application will pull from this file and then perform the necessary calculations. The web application will then be able to retrieve the fall risk calculated by our app and display it to the user (clinician).

On top of this we will also expand the current web application system by adding in questions which will allow for more useful data to be gathered for our algorithm to process and produce a more accurate guess on the patient's fall risk.

### 3.2 Functional Requirements

The system should be able to do the following:

* Allow users to input their answers to the new questions (also known as outcome measures) that we will add to the existing system; including
  + Activities Balance Confidence Questionnaire: Sixteen questions with automatic scoring
  + Timed up and go test – clinician input of timing three trials
  + Five times sit to stand test
  + History of previous falls
  + Medication usage
* Use the data that is collected from the patient to calculate their risk of falling, through way of an advanced algorithm
* Present this data to both the patient and the clinician

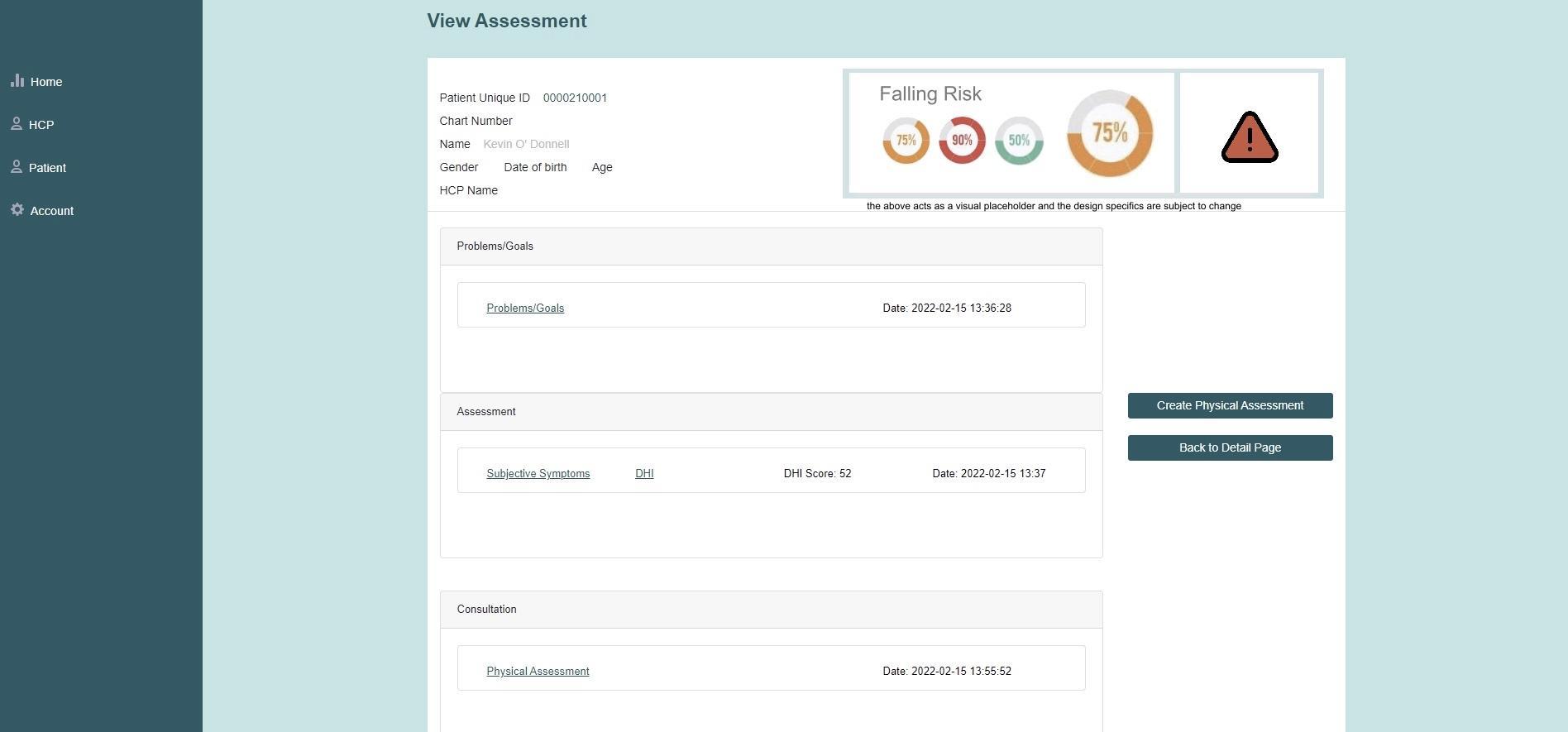
### 3.3 Non-Functional Requirements

The system should be:

* Easy to use: The layout and design should be easy to follow for patients and clinicians to use
* Maintainability: The software must be well documented and commented to allow for our clients to maintain the system after we hand it over
* Web-based: Portability is required to allow users to use the system in any location.

### 3.4. System prototype (models)

#### 3.4.1.User interface mockups



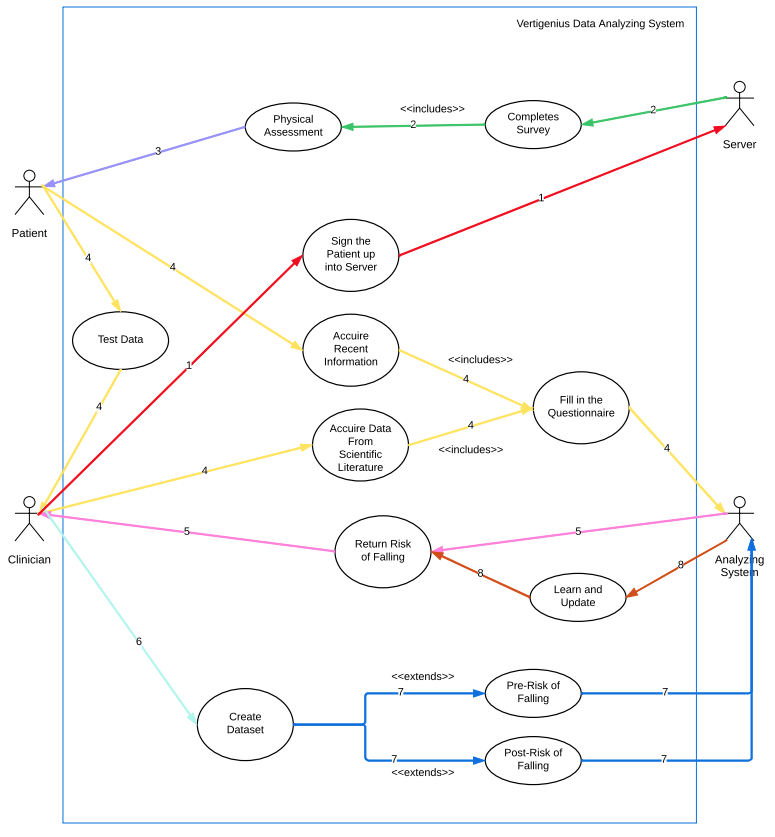
##### Expected User Experience:

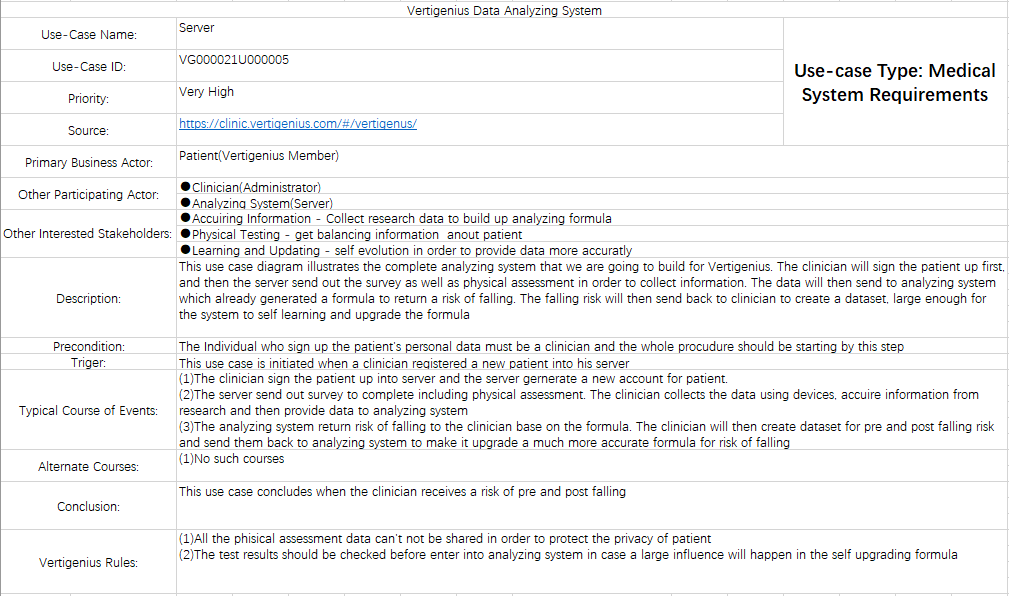
Because Vertigenius already has a web application, we don't need to make the UI from scratch, just modify it slightly. So we added a UI display about fall risk on the basis of its original.

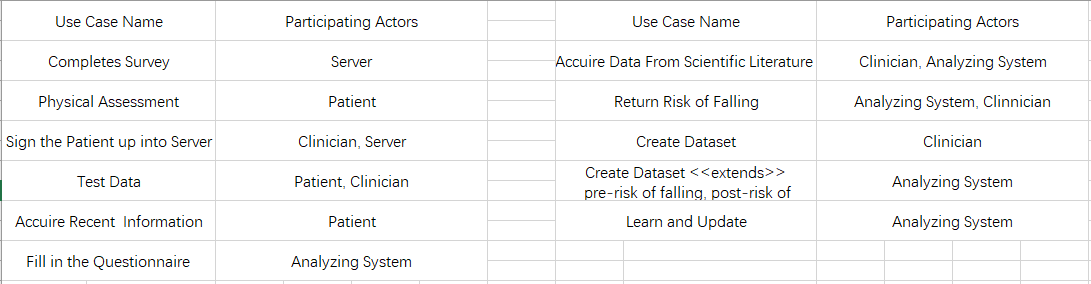
Once on this page, a reminder of fall risk will appear visually in the form of a percentage scale and an exclamation point reminder. But the colour of the visualisation will change with the risk of falling, and when the risk is higher, the colour will be redder. At the same time, a yellow exclamation mark will appear if the patient's total risk of falling is greater than 50% or less. When a patient's fall risk exceeds 75%, a red exclamation mark will appear to make it easier for users to notice the risk.

The above is just a placeholder design and location, and is subject to change depending on the needs of the client and the circumstances of the data collected.

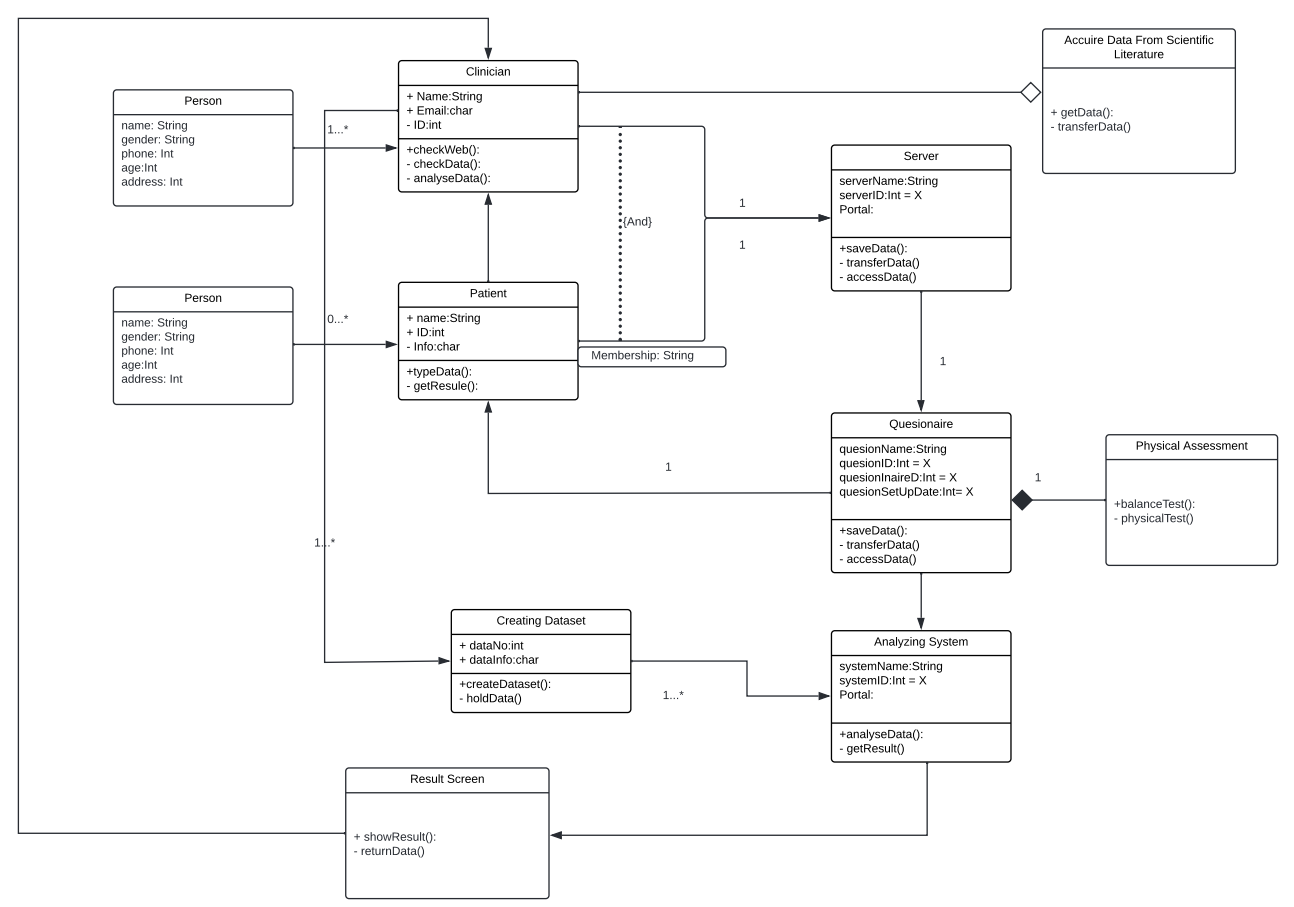
#### 3.4.2.Use cases (including text narratives)







#### 3.4.3.Object model



#### 

#### 3.4.4.Dynamic model

#### 

# 

6.2

Software Design Specification Document

Client Name : Salaheddin Alakkari

Project Name : Deep Learning Model to Detect Heart Arrhythmia in ECG Data

Group 32

**Pascal Raos, 3rd Year ICS**

**Maryann Foley, 3rd Year ICS**

**Anton Tiscoschi, 3nd Year ICS**

**Ailbhe Merriman, 3nd Year CSB**

**Kevin O’Donnell, 2nd Year CSB**

**Minjuan Luo, 2nd Year ICS**

**Zhongyuan Liu, 2nd Year ICS**

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at [http://www.tcd.ie/calendar](http://www.tcd.ie/calendar/).

I have also completed the Online Tutorial on avoiding plagiarism ‘Ready Steady Write’, located at<http://tcd-ie.libguides.com/plagiarism/ready-steady-write>.

**Contents**

1**.**Introduction

1.1 Overview

1.2 Scope

` 1.3 Definitions

1.4 References

2.System Design

2.1 Design Overview

2.1.1 High-level overview

2.2 System Design Models

2.2.1 System Context

2.2.2 Use Cases

2.2.3 System Architecture

2.2.4 Class Diagrams

2.2.5 Sequence Diagrams

2.2.6 State Diagrams

# **INTRODUCTION**

# 1.1. Overview

In a place such as the healthcare sector where a doctor's decision can determine the life or death of an individual, it is imperative that the information provided to the doctor is as exact as it can be. IT has accomplished much in hospitals, whether that be safer surgeries, life support systems or even databases to organise patient data. AI is the next step in improving the life saving capabilities of hospitals.

The system we are developing is a deep learning AI that can help detect abnormalities in a patient’s heart rhythm by reading ECG data taken in through a csv format. We plan to train our AI using an MIT ECG dataset containing different types of heart abnormalities along with normal heart beats. We plan to create an intuitive user interface to input these ECG readings, and once the AI has been adequately trained, we hope that an accurate conclusion can be reached based on the inputted data.

1.2. Scope

The primary focus of the project, the AI, will be implemented using Python. We will use a deep learning library known as Tensorflow, with its accompanying API library Keras. Keras is used to provide a more user-friendly interface into Tensorflow, abstracting the very complicated artificial neural network workings behind the scene. We will be using some accessory libraries such as numpy and pandas in case the data is unbalanced and to separate the data before feeding it into Keras deep learning models.

In addition to the AI, we will be creating a UI to interface with the AI. This will be made using Python Flask as a backend framework, with HTML templates and the front end framework Bootstrap.

1.3. Definitions

**AI** - Artificial Intelligence is the ability of a computer or a robot controlled by a computer to do tasks that are usually done by humans because they require human intelligence and discernment.

**Artificial Neural Network** - An artificial neural network is a computer based learning system modelled after the neurons and networks of neurons present in the human brain

**Deep Learning-** Deep Learning is a subtopic of artificial intelligence that focuses on AIs based on artificial neural networks to find patterns hidden in data by repeatedly processing data through multiple layers and progressively becoming better at inferring from the patterns discovered

**ECG** - An Electrocardiogram. This is a graph of the electrical activity of the human heart, and can help identify patterns in a person’s heartbeat

**Arrhythmia** - An irregularity in a person’s heartbeat

1.4. References

* Original MIT dataset link: <https://www.physionet.org/content/mitdb/1.0.0/>
* Beginner Keras tutorial link: <https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>
* IBM deep learning theory link <https://developer.ibm.com/articles/an-introduction-to-deep-learning/?fbclid=IwAR3RVDz5CmaX2VTjCKDRl8U0VCXyUWUsIcwrWQj22xhwDB1rkD7_qoHJe-o>
* Reference for resampling imbalanced dataset: <https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>

# **System Design**

# 

# 2.1. Design Overview

Our proposed system will be mainly an artificial intelligence system accompanied by a simple application that will allow for a physician to upload an ECG and will receive a classification as normal or abnormal. The AI will be trained using deep learning techniques and a data set which contains half hour recordings of ECGs, both normal and with arrhythmias. After we have trained the AI to a satisfactory level, when an ECG is uploaded we will use Keras to analyse the data and return a response.

### 2.1.1. High-level overview of how the system is implemented, what tools, frameworks and languages are used etc.

For our project, we will primarily be using Python because of its extensive resources for Artificial Intelligence. In particular, we will use NumPy, Tensorflow, and Keras. NumPy is a maths library for Python, which adds additional support for large datasets. Tensorflow is a library with a wide range of machine learning resources and Keras, a library built on top of TensorFlow, contains deep learning algorithms.

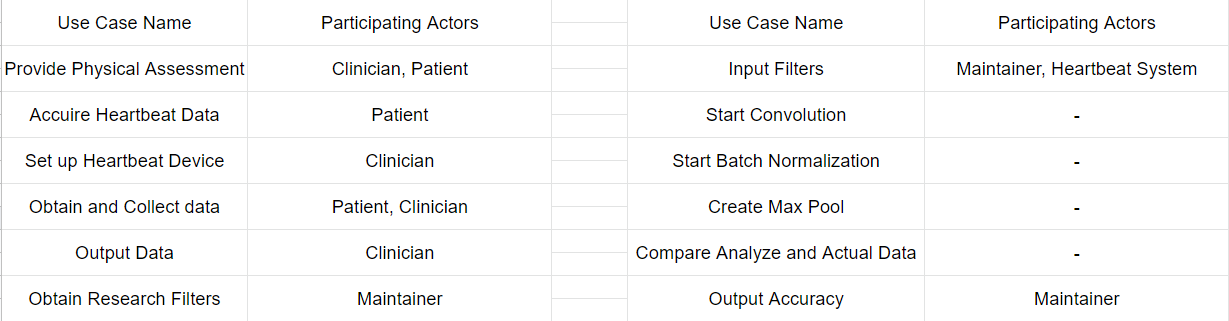
The UI aspect will be created using Python Flask, because of its ability to interface with our AI service. We will be using HTML to create the front-end. Since we are using Flask, we will also be using Jinja to create HTML templates that can be rendered using information from our Python scripts. Finally, to make our front-end more visually appealing, we will use the front-end framework Bootstrap.

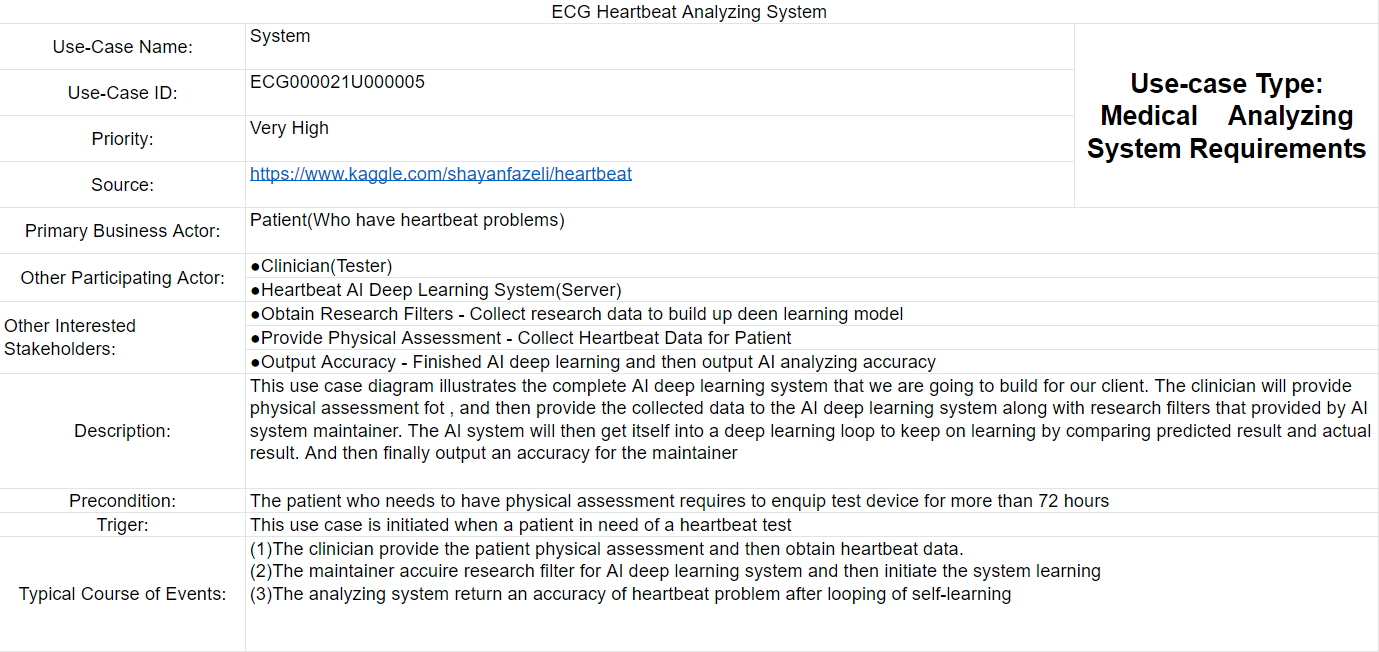
# 2.2. System Design Models

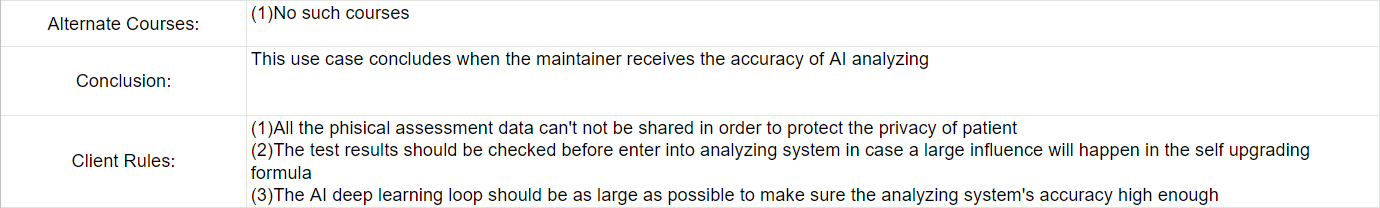
### 2.2.1. System Context

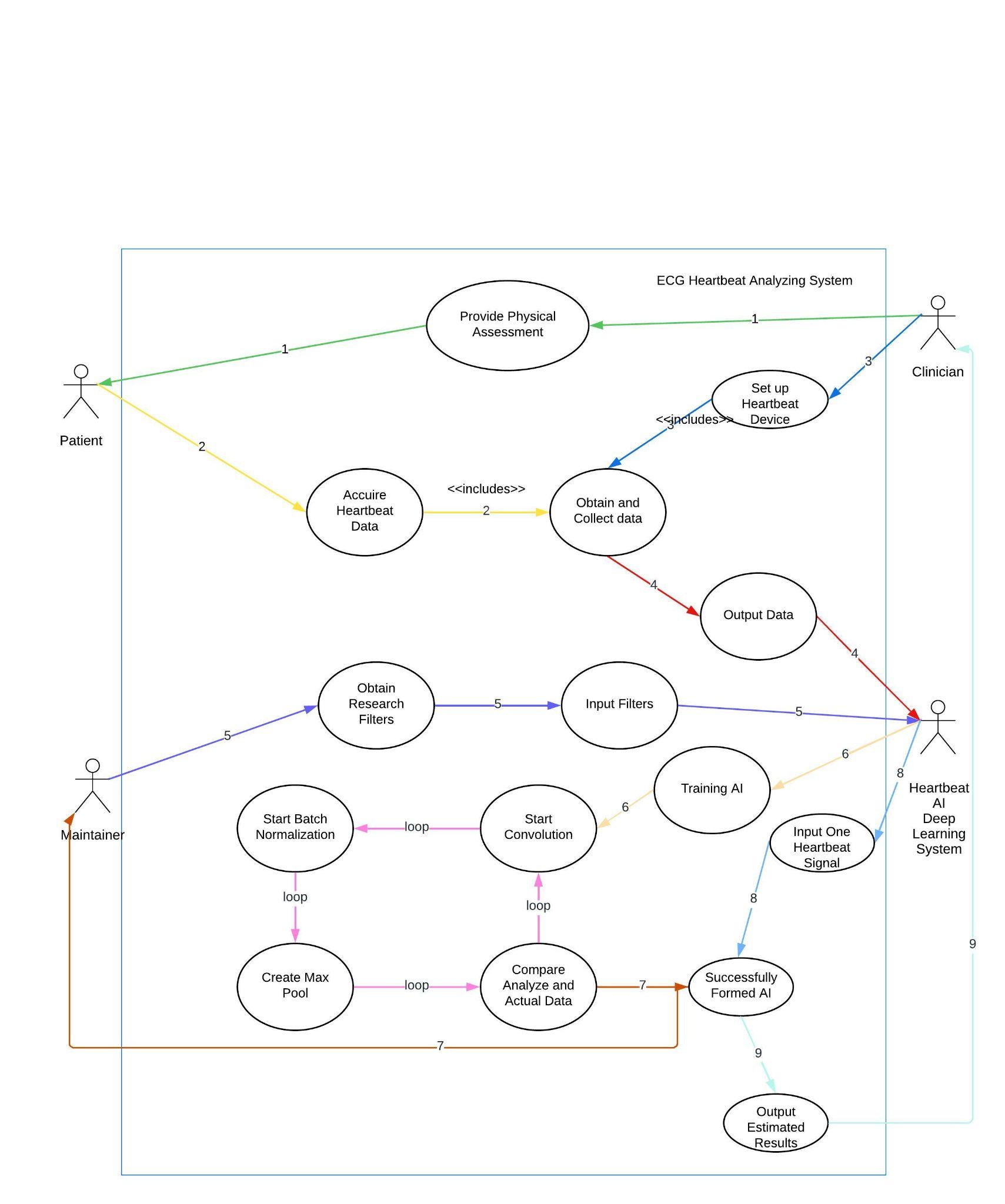
The aim of the system we will be creating is to develop a system which is a deep learning AI that can help detect abnormalities in a patient’s heart rhythm by reading ECG data taken in through a csv format. We will be implementing the AI aspect of this project by using Python, TensorFlow, Keras and for the UI element, we will be using Python Flask.

### 2.2.2. Use Cases





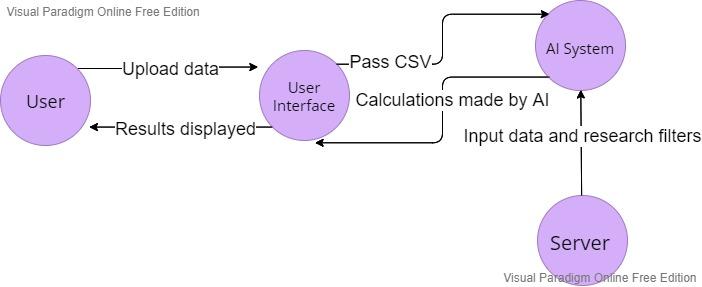
****

****

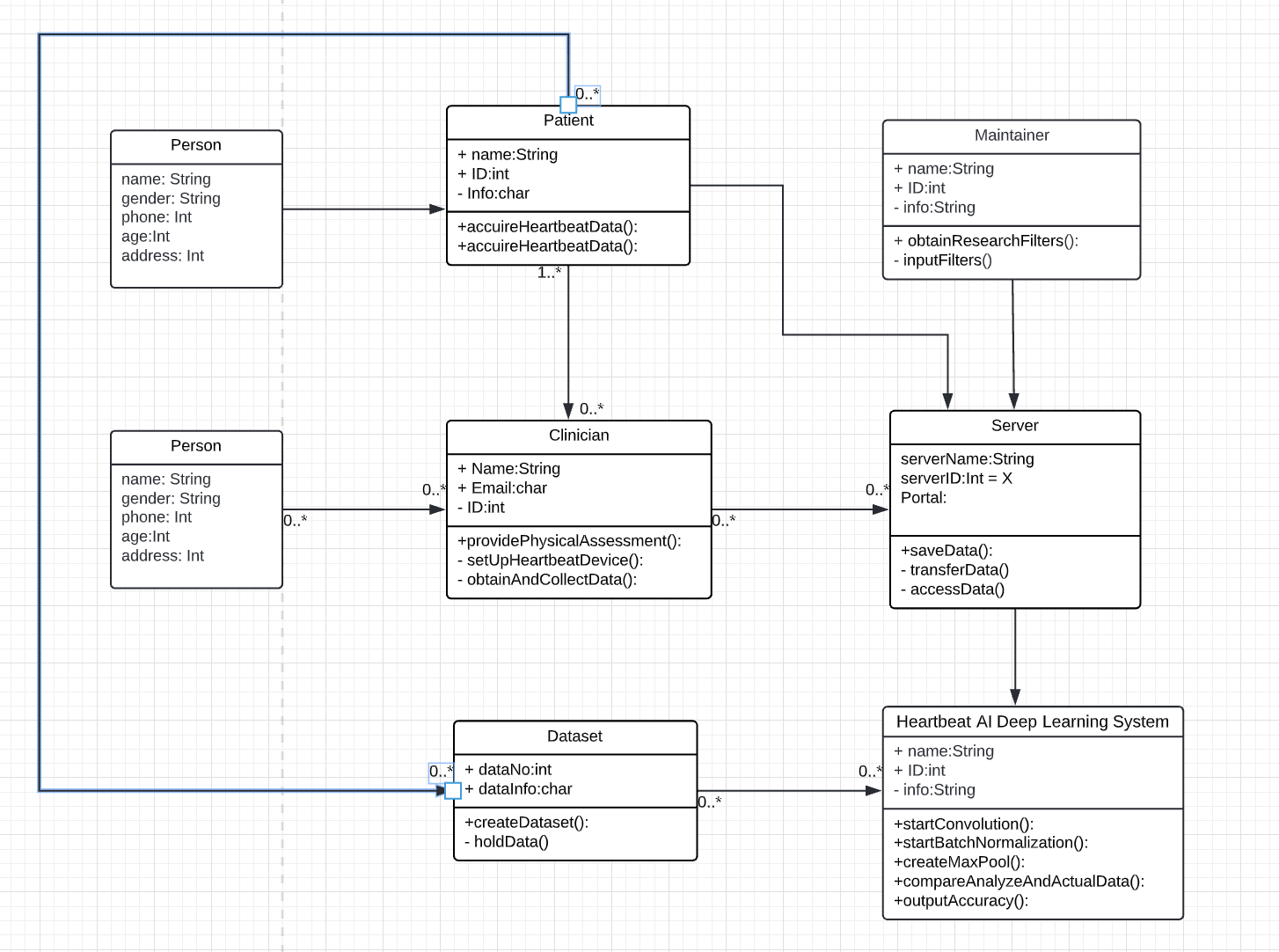
# 

# 

# 2.2.3 System Architecture

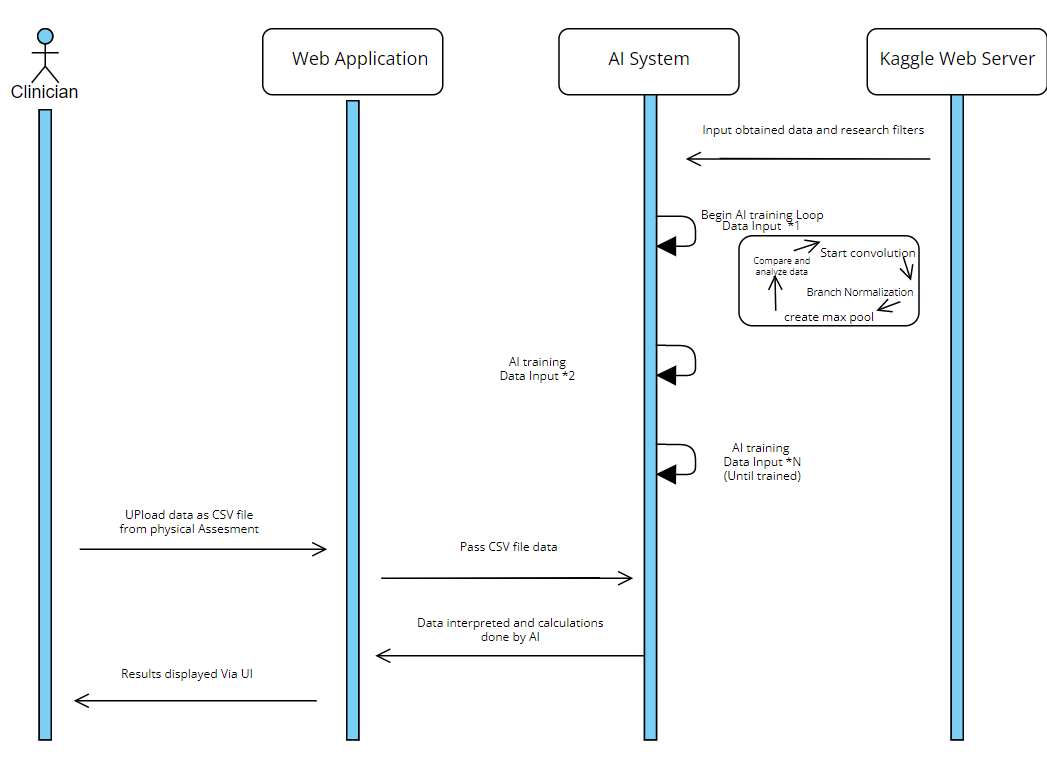


# 2.2.4 Class Diagrams



The class diagram shows a representation of how our two main deliverables, The AI deep learning system, and the AI interface may interact within a healthcare system similar to our last project with Vertigenius. The dataset is pushed into the interface held on a web server, where it is passed behind the scenes to be assessed by the AI.

# 2.2.5 Sequence Diagrams



# 2.2.6 State Diagrams

# 