Project Plan

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**

**Contents**

1**.** Project Goals and Objectives

1.1 Background

1.2 Objectives

` 1.3 Goals

2. Project Scope

2.1 Project Deliverables

2.2 Project Boundaries

2.3 Product Backlog

3. Project Approach

3.1 Scrum Sprints

4. Project Organisation

4.1 Staff

4.2 Staff Chart

5. Risk Analysis

5.1 Risk Analysis

5.2 Risk Mitigation

6. Project Controls

7. Communication

7.1 Client Communication

7.2 Project Team Meetings

8. Appendix

8.1 Requirements Document

8.2 Software Design Specification

# 1. Project Goals and Objectives

## 1.1 Background

The project our team is working on is to create an Artificial Intelligence to determine whether a patient has an abnormal heartbeat by analysing ECG data.

The aim of the project is to train a deep learning model using ECG data to a level where it can recognise abnormal beats in a patient with heart arrhythmia.

Our project could have practical applications in a hospital environment and potentially save lives.

## 1.2 Objectives

The aim of our system is to create an Artificial Intelligence with a high degree of accuracy in its assessments. We plan to create an intuitive user interface that will allow the client to input their own ECG data, and if time permits, to potentially incorporate ECG readings from fitness trackers such as an Apple Watch or a FitBit. The end goal is to provide an accurate assessment of a patient's heart condition based on their ECG readings

## 1.3 Goals

Our goal is to deliver an intelligent deep learning model that has been trained on the “MIT-BIH Arrhythmia Database”, a dataset comprised of 48 half-hour ECG readings sampled at 360 samples per second to create approximately 110,000 data entries. These data entries are split at approximately 90,000 in the training dataset, and 20,000 in the testing dataset. Alongside this, we will deliver a clear and intuitive user-interface that would allow a user to input their own ECG data in csv format to determine whether patient(s) data contains irregular heartbeats.

# 2. Project Scope

## 2.1 Project Deliverables

This entire project can be considered a deliverable for the material examiners. The final submission of this project will contain all the features of the project that we have given to the client, as well as relevant documentation on system functionalities and module-relevant material such as this one. The features included in the items we will hand the client are mockups and prototypes for the architecture, user interface, logic, as well as the fully completed counterparts of the listed deliverables.

The code base will consist of a front-end runnable application and back-end server for handling uploaded ECG data where the computations are done. All the back-end code will be written in python. The front-end will be written in html, and python, and will pertain to a web application that the user will interact with.

So far, we have submitted a **Signed-off Requirements Document\*** and included with this document is the **Software Design Specification\*.** This leaves three documents left for submission to Blackboard - the **(1)Development Plan, (2)Management Report, (3)Individual Reflection Essay.**

## 2.2 Project Boundaries

Our project consists of two parts, the user interface and the artificial intelligence back-end server. The server receives data, where the AI determines if the person has heart problems based on the ECG data, and relays usable data to the front-end. Any restrictions will be centred around how extensively the data can be processed, as well as time-constraints.

***In-Scope:***

* Developing the user interface.
* Training an AI model that will return if a heart is healthy from ECG diagrams.
* Converting data received in CSV format into a more readable structure.
* Relaying data by means of endpoints serving requests coming from the web application.
* Completing the functional and non-functional requirements set out in the project description.

***Out-of-Scope:***

* Considering the limitations of time, additional features asked for such as getting ECG data from an apple watch or fitbit might not be entirely feasible and as such we would, for now, consider them out of scope of the project.

## 2.3 Project Backlog

| To do | Doing | Done |
| --- | --- | --- |
| Trying out different deep learning techniques to create different types of models | Setting up an initial deep learning model | Set up Github repository |
| Connect User Interface with AI | Create Flask Application | Retrieve Data for training AI |
| Create HTML UI | Create HTML UI templates | Learn Python for backend development |
|  |  | Learn HTML for user interface |

## 

# 3. Project Approach

# 3.1 Scrum Sprints

Due to the unique nature of our project, the first scrum will be taken from the beginning of our second project, i.e starting from 07/03/22, the Monday when we decided which AI project to do.

| **Scrum Title** | **Description** | **Start Date** | **End Date** | **Duration** |
| --- | --- | --- | --- | --- |
| Research | Meetings with our client to gain a better understanding of what we need to do. Doing research on deep learning and getting to grips with our dataset. Also deciding on which languages and softwares to use. | 07/03/22 | 14/02/22 | 1 week |
| AI and UI Development | Begin work on the development aspect of our project. Continued learning and research of TensorFlow and Keras for the Deep Learning AI team and Flask and HTML for the UI team. Accommodate the 2nd year’s continued efforts to learn new languages and GitHub. | 15/03/22 | 29/03/22 | 2 weeks |
| Integration of work from both teams | These two weeks will be dedicated to finishing off our project and integrating the work completed by both the UI and AI Deep Learning teams. Both the management report and the development report will be completed also. | 30/03/22 | 14/04/22 | 2 weeks |

# 

# 4. Project Organisation

## 4.1 Staff

**SKILL REQUIRED BY ROLE:**

Table below provides information regarding the skills and responsibilities needed by each of the distinct roles needed in our project:

| **Role** | **Skills Required** | **Responsibility** |
| --- | --- | --- |
| User Interface Developer | Experience with Python Flask and HTML | To reate the front and back end of the User Interfacecreate an interface for the AI system that can read in a test ECG file and display the predicted assessment concluded by the AI |
| Deep Learning AI Developer | Experience in python with knowledge of Tensorflow, Keras and competency to properly format datasets | To create, train, and test a deep learning model which can conclude if a patient has abnormal heartbeats |
| Communicator | People skills and time management | Ensure consistent communication between the team and the client and organise team meetings. |

**SKILL ALLOCATION:**

The table below provides information on the skills used and/or needed by the members of our team, and the methods they must take to acquire previously unknown skills:

| **Skill** | **Team Member** | **Method to acquire skill** |
| --- | --- | --- |
| TensorFlow/Keras Deep Learning Libraries | Pascal Raos, Ailbhe Merriman, Minjuan Luo | Research of Tensorflow and Keras syntax through tutorials, code examples and library documentation |
| Method to Correct Imbalanced Datasets/Transforming Data into Different Shapes | Pascal Raos, Ailbhe Merriman, Minjuan Luo | Research into resampling datasets and class weighting in Keras |
| Python Flask Application | Maryann FoleyAnton Tiscovschi, Maryann, Kevin O’Donnel,  Zhongyuan Liu | Research Python Flask sample applicationsResearch of Web development and HTML/Flask syntax through tutorials, code examples and library documentation |
| HTML User Interface | Maryann FoleyAnton Tiscovschi, Maryann, Kevin O’Donnel,  zhongyuan Liu | ResearResearch into web development and HTML |

## 

**ROLE ALLOCATION:**

The table below provides information on the role allocation and which group members have been assigned to each

| **Role** | **Team Member** | **Responsibility** |
| --- | --- | --- |
| Deep Learning AI Manager  Communicator | Pascal Raos (3rd Year) | Guiding the AI team to correctly format input data and train the AI in an effective method to reach high accuracy predictions |
| Deep Learning AI Manager | Ailbhe Merriman (3rd Year) | Guiding the AI team to correctly format input data and train the AI in an effective method to reach high accuracy predictions |
| Deep Learning AI Developer | Minjuan Luo (2nd Year) | Working under the guidance of the managers, to create and train deep learning models using ECG datasets |
| User Interface Manager | Maryann Foley (3rd Year) | Leading UI team to develop a web application and connect it to the AI |
| User Interface Manager | Anton Tiscovschi | Co-Leading UI team to develop a web application and connect it to the AI |
| User Interface Developer | Kevin O’Donnel | Working under the guidance of the managers, to create a web application and connect it to the AI |
| User Interface Developer | Zhongyuan Liu | Working under the guidance of the managers, to create a web application and connect it to the AI |

## 4.1 Staff Chart



# 5. Risk Analysis

## 5.1 Risk Analysis

| **Risk Element** | **Impact**  **(1 to 5)** | **Likelihood (1 to 5)** | **Risk Factor**  **(I \* L)** |
| --- | --- | --- | --- |
| Miscommunication with Client | 4 | 1 | 4 |
| Deep Learning AI has poor accuracy | 5 | 1 | 5 |
| Unable to apply product in a simple real world scenario (basic input of ECG csv file) | 4 | 1 | 4 |
| Unable to apply product in a complex real world scenario due to time constraints(reading smart watch data) | 2 | 4 | 8 |

## 5.1 Risk Mitigation

| **Risk** | **Measures to Reduce Risk** |
| --- | --- |
| Miscommunication with Client | Frequent communication with client; discussions within team about clarifying questions; precision in documentation and communications |
| Deep Learning AI has poor accuracy | Ensuring the data is correctly formed and balanced before training the model will eliminate all risk |
| Unable to apply product in a simple real world scenario (basic input of ECG csv file) | Ensuring there is a planned connection between the input in the simple UI and the input to the AI will be the only difficulty, creating a simple UI to take in a file should cause no issues. |
| Unable to apply product in a complex real world scenario due to time constraints(reading smart watch data) | We talked about possibly extending our AI’s functionality to read in smart watch ECG data directly, however due to time constraints caused by the changing of our project, this level of application may not be feasible if we take too long to get the deep learning model trained |

# 6. Project Controls

In order to control our project execution, we used a multitude of resources. In order to increase efficiency and progress, we have regular meetings, both in person and over Zoom, and use them to establish a plan for our coming sprints. We have a shared google drive folder, where we collaboratively work on documentation using Google Docs and Google Slides and gather resources, and a github repository for sharing our code. To ensure team communication, we have an active team WhatsApp group message, where we will discuss all aspects of the project with great frequency. This active line of communication also allows us to keep on top of deadlines, as we will notify each other and make sure everyone is aware of pending submissions. Our team is in frequent communication with our client to ensure that all details have been ironed out and all of our expectations are made clear.

# 7. Communication

## 7.1 Client Communication

Our communication with our client, Salah Alakkari, has been primarily through email. Thus far, our meetings have been arranged through email and we have met both over Zoom and in person. Through these meetings, we have been able to discuss our project options and refine the work we will be doing, as well as learn about relevant tools we can use. We plan to meet with our client on a weekly basis.

## 7.2 Project Team Meetings

Our team is communicating mainly through WhatsApp, as it is where we are all most likely to be active. Additionally, we have a Google Drive shared folder, where all of our documents are stored but also where we share resources and develop working plans. We meet at least once a week over Zoom for our SCRUM meeting, usually on Friday afternoons, but will usually also meet after client or demonstrator meetings to discuss any developments to our project.

# 

# 8. Appendix

## 8.1 Requirements Document (This is the old requirements document, so is not very relevant to our current project)

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

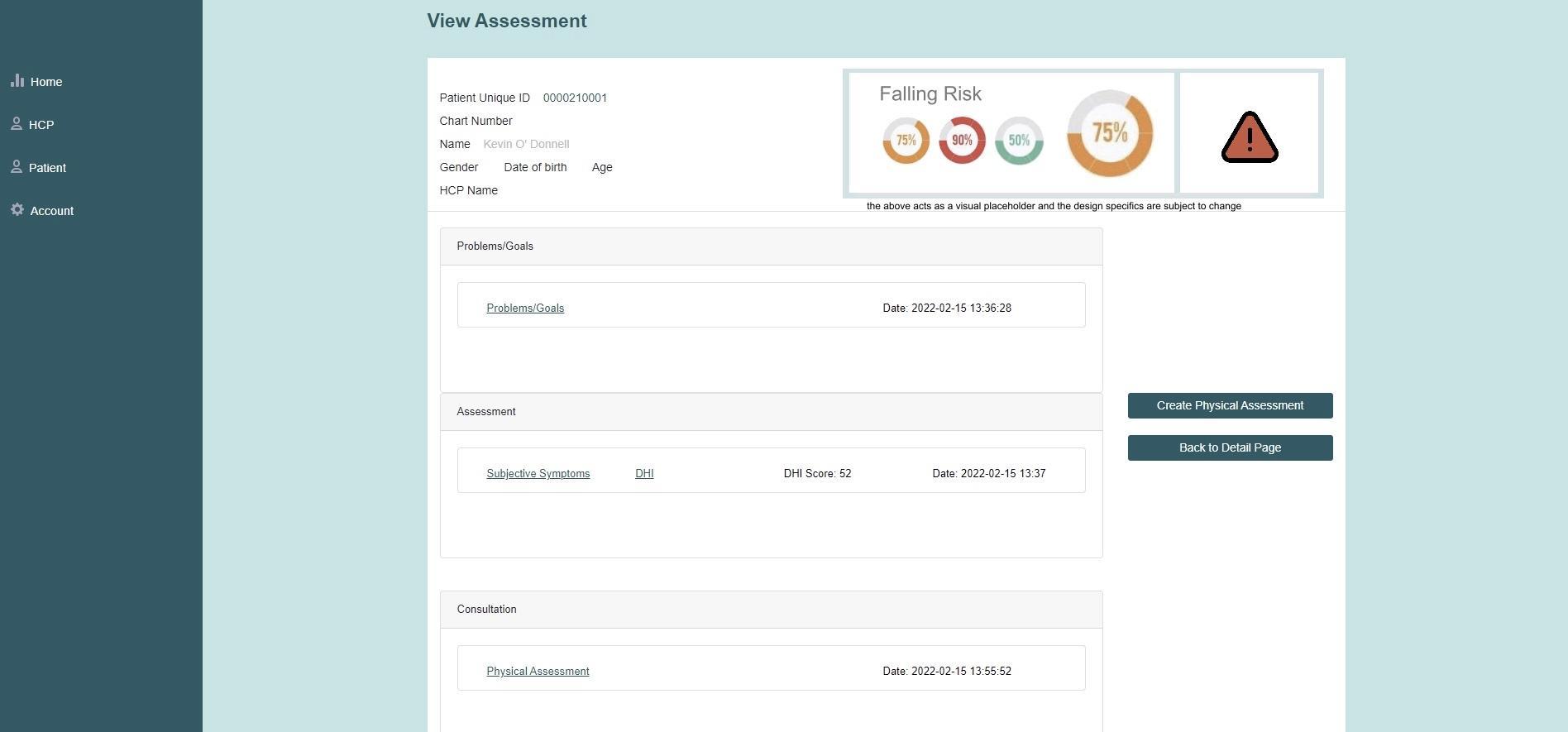
### 3.3 Non-Functional Requirements

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



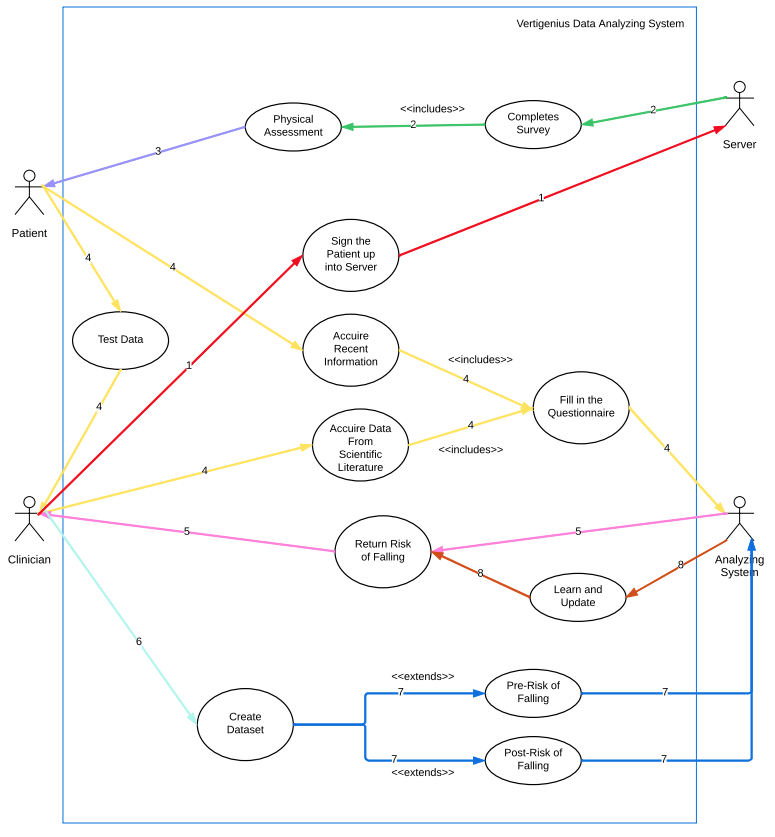
##### Expected User Experience:

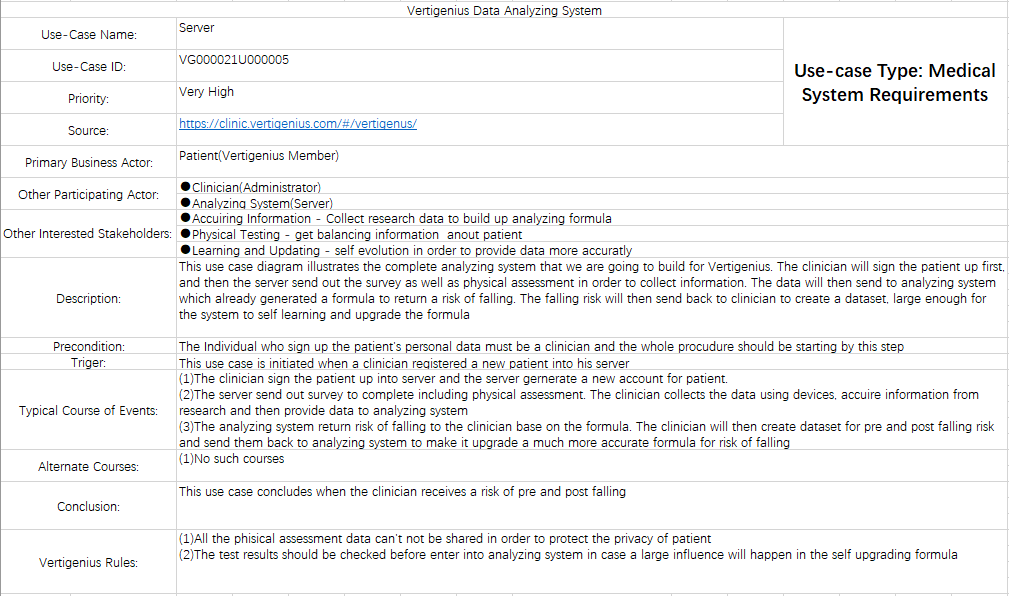
Because Vertigenius already has a web application, we don't need to make the UI from scratch, just modify it slightly. So we added a UI display about fall risk on the basis of its original.

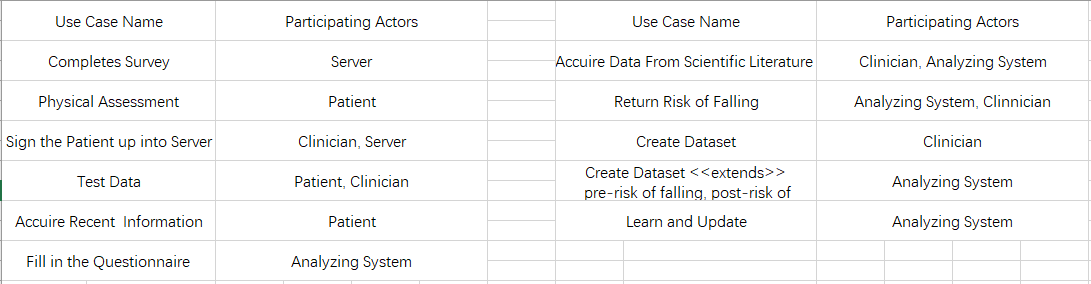
Once on this page, a reminder of fall risk will appear visually in the form of a percentage scale and an exclamation point reminder. But the colour of the visualisation will change with the risk of falling, and when the risk is higher, the colour will be redder. At the same time, a yellow exclamation 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.

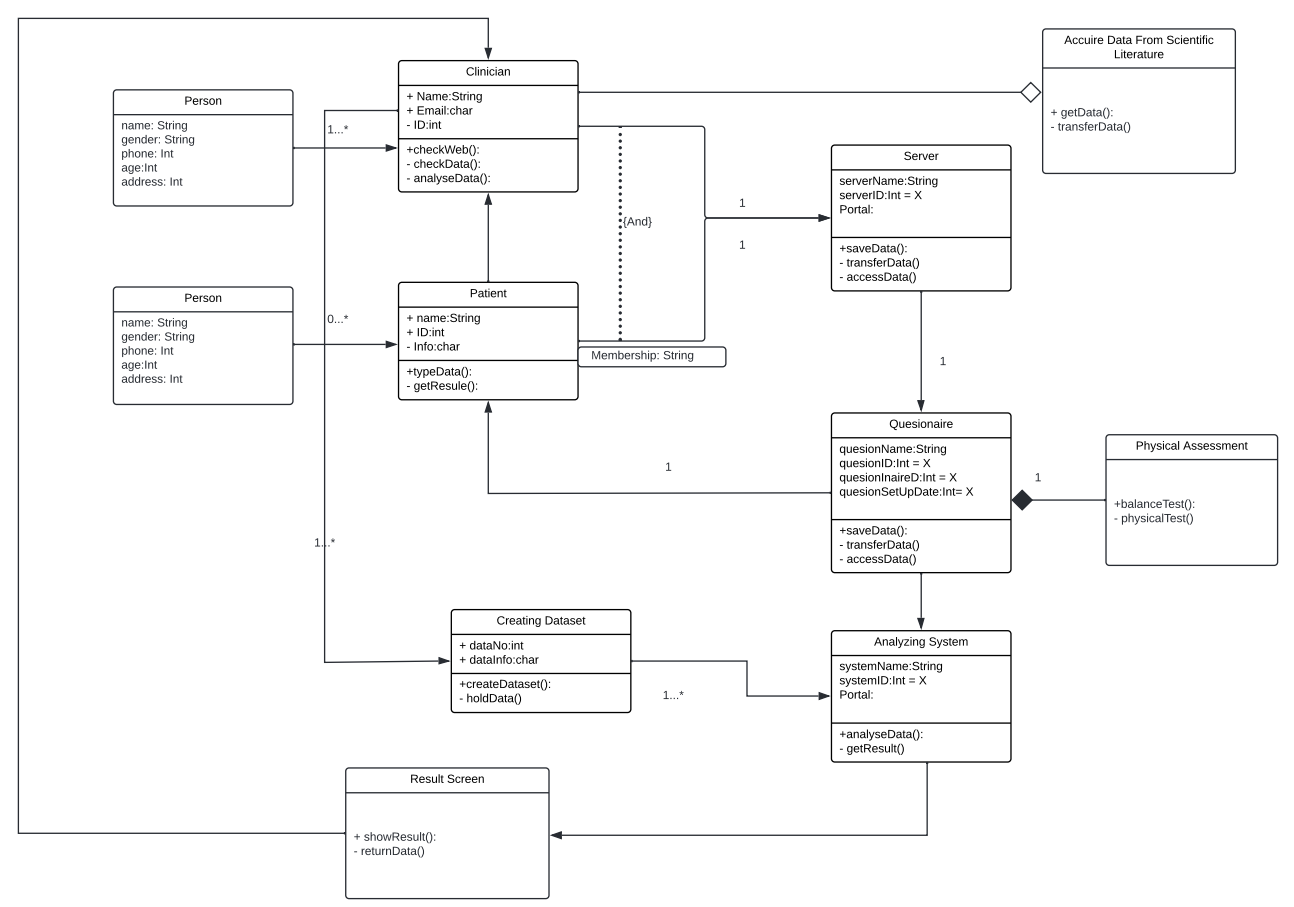
#### 3.4.2.Use cases (including text narratives)







#### 3.4.3.Object model



#### 

#### 3.4.4.Dynamic model

#### 

#### 

#### 

## 8.2 Software Design Specification Document

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.

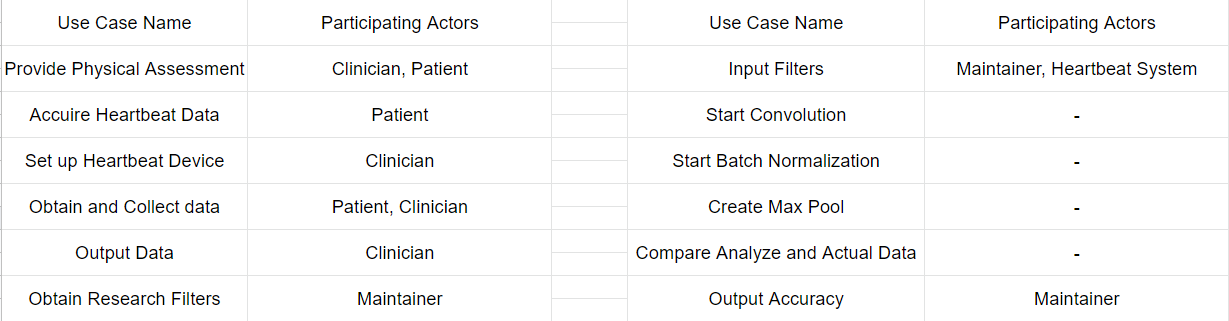
The UI aspect will be created using Python Flask, because of its ability to interface with our AI service. We will be using HTML to create the front-end. Since we are using Flask, we will also be using Jinja to create HTML templates that can be rendered using information from our Python scripts. Finally,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

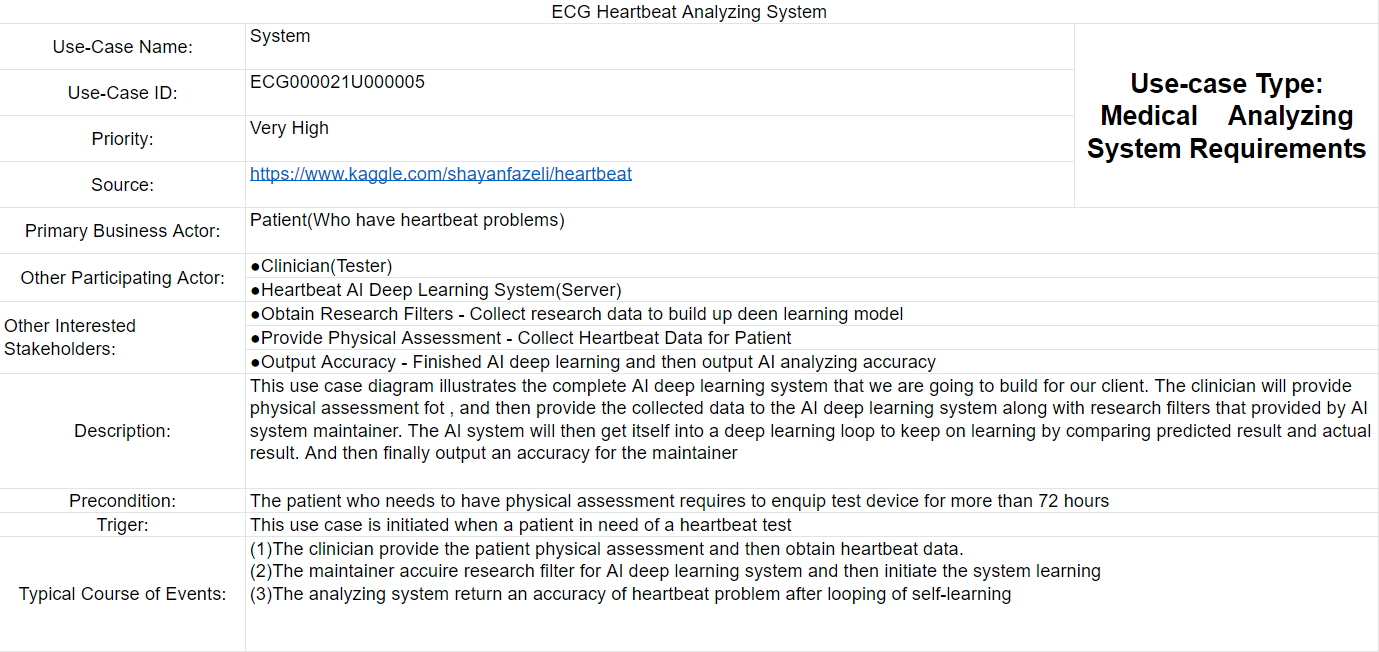
# 2.2. System Design Models

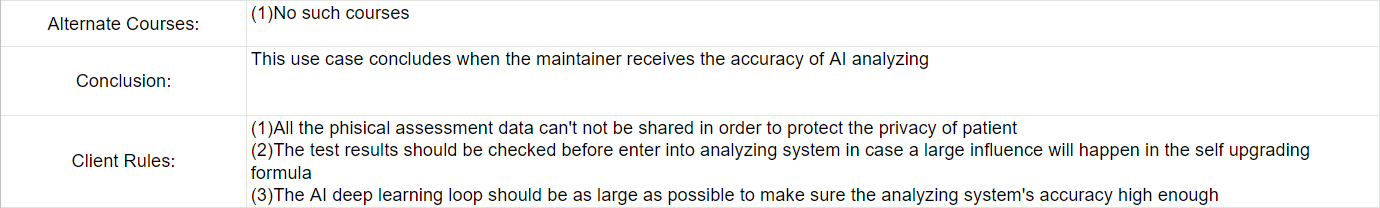
### 2.2.1. System Context

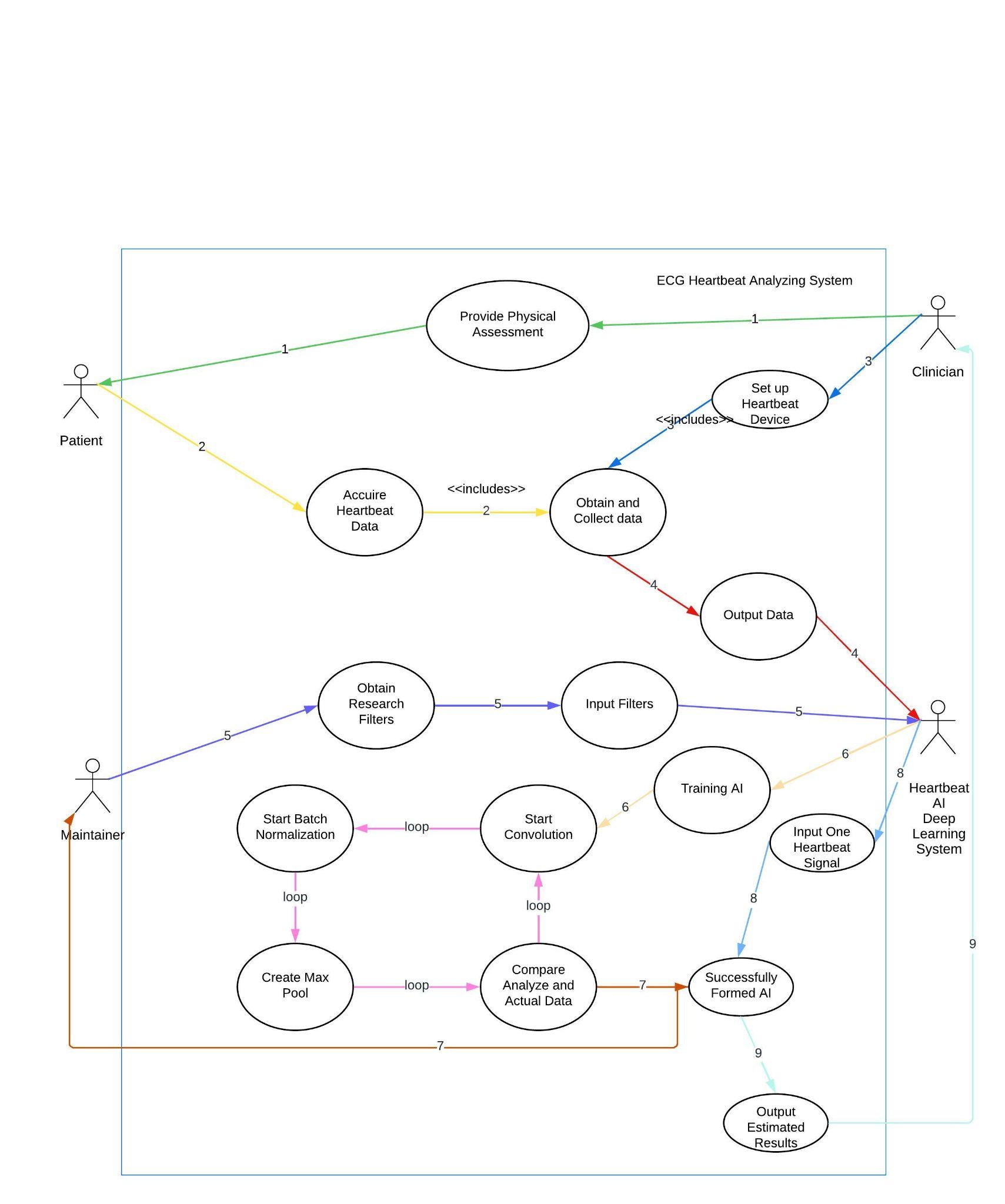
The aim of the system we will be creating is to develop a system which is a deep learning AI that can help detect abnormalities in a patient’s heart rhythm by reading ECG data taken in through a csv format. We will be implementing the AI aspect of this project by using Python, TensorFlow, Keras and for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

### 2.2.2. Use Cases

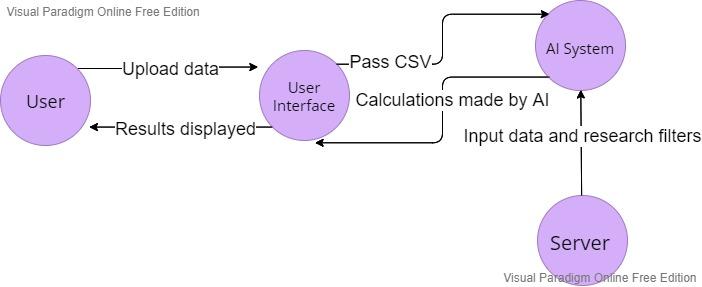




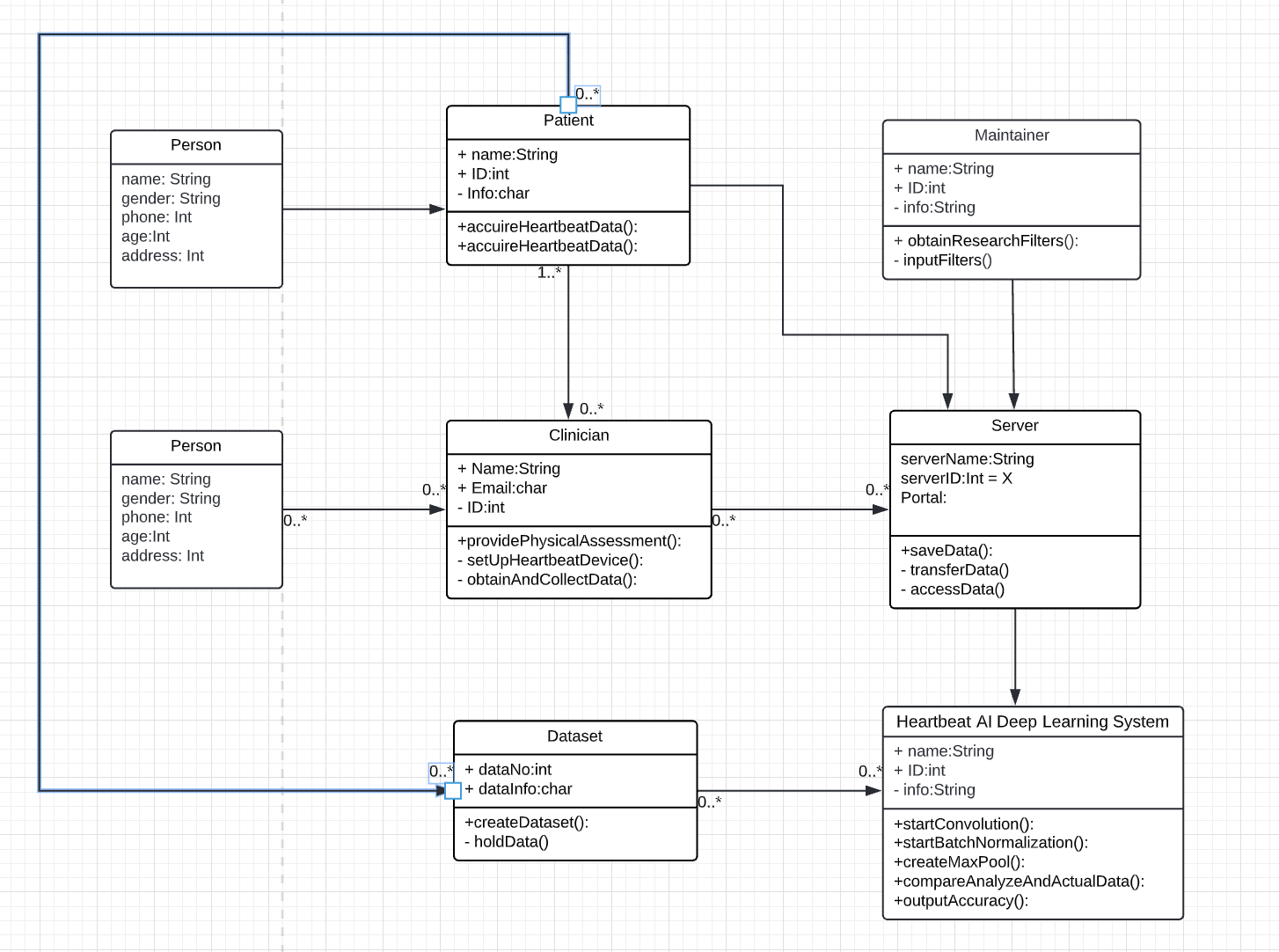
****

****

# 2.2.3 System Architecture

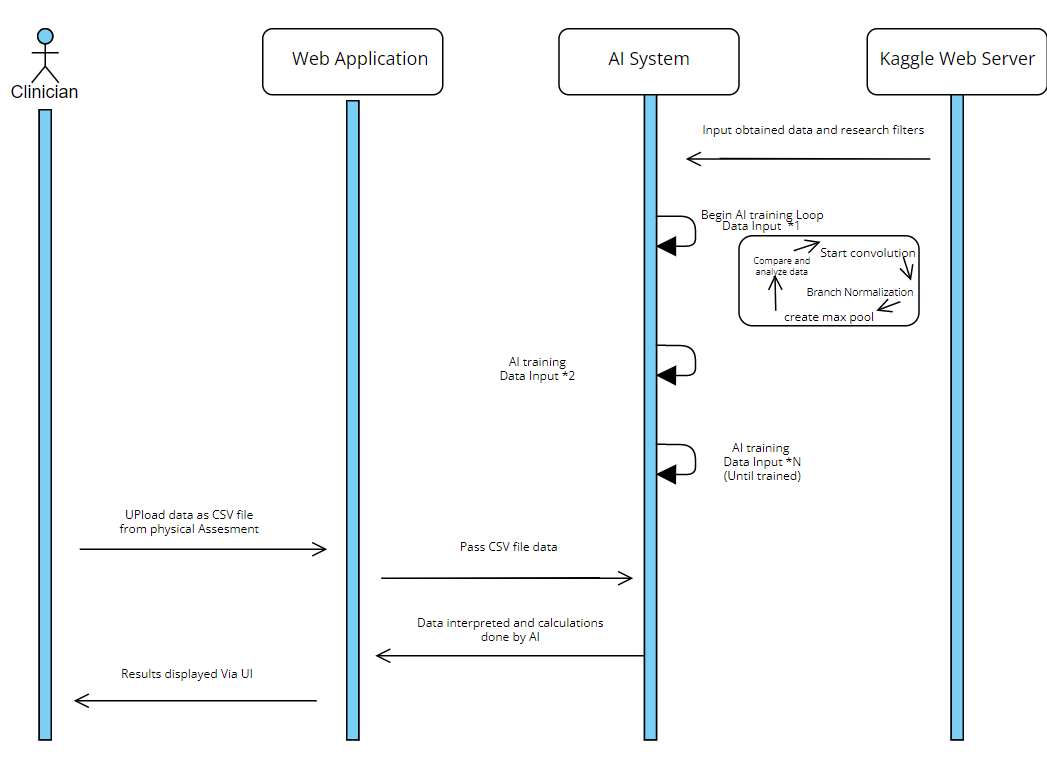


# 2.2.4 Class Diagrams



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

# 