Functional Specification - Class Vision

Classroom Analysis using Computer Vision

Date: 03/12/2022 Gareth Hogan 20379616 & Joshua Ward 20460854

Table of Contents

1. Introduction **2**
1.1 Overview 2
1.2 Glossary 2

2. General Description **3**
2.1 Product / System Functions 3
2.2 User Characteristics and Objectives 3
2.3 Operational Scenarios 4
Setting up a class group 4
Activate the CV Tool 4
2.4 Constraints 4

3. Functional Requirements **5**
Use Case Diagram 5
Functional Requirements: 6
3.1 - User Registration 6
3.2 - User Login 6
3.3 - Edit Class Groups 7
3.4 - View Timetable 7
3.5 - Import Timetable 7
3.6 - Enable Computer Vision Tool 8
3.7 - View Insights Generated by Computer Vision Tool 8
3.8 - View Lecture Details 9
3.9 - User Logout 9
3.10 - Take Attendance 9

4. System Architecture **10**

5. High-Level Design **11**
Data flow diagram 11
Sequence Diagram 11

6. Preliminary Schedule **12**

7. Appendices **12**

1. Introduction

1.1 Overview

The system we are developing is a computer vision tool to be used in the research and development of lecture and teaching materials, it can be used by lecturers to gather insights into how well their classes are engaging with their lectures and specific points of interest during the lecture where interest spiked, or interest was lost.
The system will be presented as part of a web application. The main component of our web app will be the computer vision system, which can be turned on during a lecture and uses a webcam and our computer vision algorithm to gather data from the lecture, this will then be presented to the user as insights into how the class went. Our web application will also have the functionality to take attendance, view timetables, set up classes and review previous data on lectures.

1.2 Glossary

Term	Description
Computer Vision	Computer Vision is a field of AI. It enables computers and systems to be able to see and observe, it allows them to derive information and data from visual inputs such as images and videos.
OpenCV	(Open Source Computer Vision Library) OpenCV is an open-source library full of functions mainly aimed at real-time computer vision and machine learning
YOLO	(You Only Look Once) YOLO is a method to do object detection in computer vision. It is the strategy of how the code can detect objects in an image, it is well known for being much faster than other methods, as the name suggests it only does one pass over the image input.
OpenTimetable	OpenTimetable is a website made by DCU which allows users to see the timetable for particular modules, lecture rooms, and programmes of study. It also allows users to see the details of scheduled events, such as the type of event and where it takes place.

2. General Description

2.1 Product / System Functions

- Register User
- Login/Logout
- Import Timetable
- Setup Class Group
- Edit Class Group
- Set Lecture Times
- Link Lecture and Class Group
- View Timetable
- View lecture details via Timetable
- Start computer vision tool from timetable
- Start computer vision tool from lecture details
- View data insights after the lecture is complete
- View historic data insights from past lectures
- View past data on Timetable
- View past lecture data reports in a list
- Sort and search data reports by class name, time, date

2.2 User Characteristics and Objectives

The intended users of this system include lecturers and staff at colleges. We would expect their technical abilities to be good which will be more than adequate to use our system, we would expect them to have experience using other dashboard-style websites similar to our system (for example in DCU lecturers would have experience using Loop).
From the users' perspective, they would hope to be able to do the following things using our system:

- Import their timetable
- View and edit their timetable
- Setup & Manage Class groups
- View data insights on previous classes
- Start the analysis tool for a current class
- View results right after class has ended

2.3 Operational Scenarios

Below are two example use cases of our system, the most commonly used functions:

- Set Up a Class Group
- Activate the CV Tool for a lecture

Setting up a class group

An existing user wants to set up a new class group in the system, to achieve this they will

1. Log into the system website
2. Navigate to the “Your class groups” section on page
3. Click the add new class button
4. Add in the class details, such as name, description and number of students
5. They can then add a list of students in group
6. Then they click the “Save Group” button
7. Then they are presented with another page to link lectures
8. On this page, they can either link existing lectures on their timetable to the class group or they can make new lectures on their timetable
9. Once done they can save changes and exit the homepage

Activate the CV Tool

A user is starting a lecture and wants to activate the analysis tool, they will

1. Ensure they are logged into the website
2. Navigate to their timetable page to view the timetable
3. Click the lecture they are in and want to analyse
4. The lecture details are displayed and they can click the “Start Recording” button
5. After the lecture is done, they click “Stop Recording”
6. The results will be displayed on the screen to the user
7. They can also close this screen and the results will be collated and stored in the lecture slot that was just completed

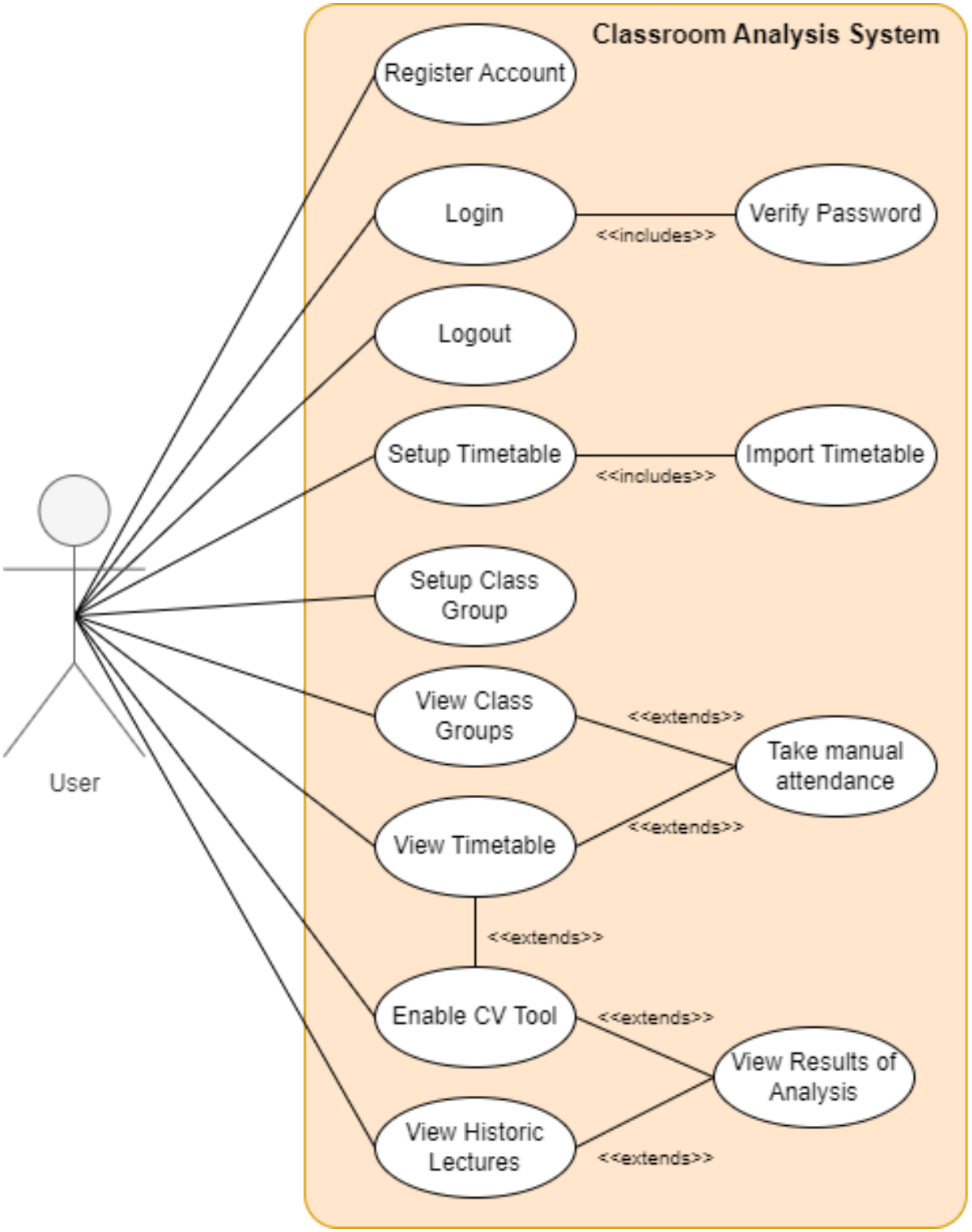
2.4 Constraints

- Software Versions - Python 3.X, OpenCV2 4.2.0
- Platform - Desktop Web Browser
- Hardware - External webcam required
- Camera - Resolution at least 640x480
- OS - Development on Linux

3. Functional Requirements

You can see below our use case diagram for the system, and below that, we have listed our functional requirements for the system.

Use Case Diagram



Functional Requirements:

3.1 - User Registration

Description: The system must allow users to input a username and password to create an account that they can store and access their personal data in the future. User registration details will need to be stored in a database to allow users to log in in the future.

Criticality: User registration is essential to the system, to allow lecturers to create accounts to manage their information in a private manner in the future. Users won't be able to access the system without having an account registered.

Technical Issues: The system needs to be able to make sure that multiple users can't register with the same username or email, as this could lead to the user logging into the wrong account,

Dependencies with other requirements: This requirement is linked with the 3.2 - User Login requirement, as that uses the details input here to determine whether the user's account exists.

3.2 - User Login

Description: The system must be able to input a username and password, and check if these have been registered. If the system confirms that the account exists and the password is correct, it will allow the user to enter the account and access the service.

Criticality: The ability to log in is essential to the functionality of the system, as users will need to log in every time they wish to access the system and any of its functionalities.

Technical Issues: The system must ensure that account login information is kept secure so that users' accounts cannot be accessed by malicious actors, while still making sure the user is able to access their account whenever they need it.

Dependencies with other requirements: This requirement is dependent on the 3.1 - User Registration function happening before it takes place, and all other requirements are dependent on the login function to allow users to access them.

3.3 - Class Group Management

Description: The user will be teaching multiple class groups (e.g. Modules), the user should have the ability to create and update class groups with lists of participants and link them to lectures in the timetable

Criticality: The user should be able to separate and differentiate easily between their different class groups, so being able to set up and link groups to the timetable is important for the visual experience of the user

Technical Issues: The system must be able to store the details of a class group and it's participants and details, and maintain a relationship between the timetable and group, showing what group each lecture is associated with

Dependencies with other requirements: This requirement is dependent on the 3.5 - Import Timetable and 3.4 - View Timetable requirements, to allow for easy integration with particular modules.

3.4 - View Timetable

Description: Users should be able to view their timetable on the website, allowing them to check when their lectures are, and check details for current and future lectures while also allowing them to revisit data from previous lectures.

Criticality: This function is crucial to the system, as it is one of the core functionalities being supplied to users. It is also required to allow users to access the data of old lectures, which is necessary to utilise the computer vision part of the system to the fullest extent.

Technical Issues: The information conveyed by the timetable must be documented clearly so that users can easily understand and use the tool without needing to spend a lot of time getting familiar with the layout. It is also important for the timetable to seamlessly integrate access to previous lecture data, to allow for easy access to the results of the computer vision aspect of the system.

Dependencies with other requirements: This requirement is dependent on 3.5 - Import Timetable, as having a timetable entered into the system is a requirement for being able to view the details. It is also a dependency for 3.9 - View Lecture Details, as users need a method to be able to access these.

3.5 - Import Timetable

Description: Users should be able to import existing timetable data from other sources into the system, such as from OpenTimetable or Google Calendar, to prevent the waste of time from requiring users to manually enter each data entry into their personal timetable

Criticality: This function is not essential to the system, as even if the function were not to work users would still be able to manually create their timetables within the system. It will save a lot of time for users however, so it is still important to get it working.

Technical Issues: An issue that could arise would be that data being imported could be an unfamiliar type and not be supported, so we will need to account for all the most common file types and potentially some more niche options that we know people may use.

Dependencies with other requirements: This requirement heavily ties into the 3.4 - View Timetable requirement, as the user cannot access their timetable if they do not have one added to their account.

3.6 - Enable Computer Vision Tool

Description: The user should be able to enable the computer vision tool before a lecture, which will then anonymously record the interactivity of the students in the lecture before providing the information gathered to the lecturer in a helpful format after the lecture's conclusion.

Criticality: The enabling of the computer vision tool is essential to our system, as it forms the basis of the project's application. Users should be able to easily and quickly determine whether or not the computer vision tool has been enabled, and enable or disable it depending on the circumstances.

Technical Issues: We will need to make sure that the computer vision microservice is integrated well into the user interface so that it is simple for users to enable and access the tool. It is also important for the tool to be intuitive for users to access so that there isn't a large amount of prerequisite learning of how to use the tool before being able to use it.

Dependencies with other requirements: This requirement is a dependency for both 3.7 - Generate Insights using Computer Vision Tool, and 3.8 - Viewing Insights Generated by Computer Vision Tool, as the user must first enable and use the tool before they can see the resulting feedback from the lecture.

3.7 - Generate Insights using Computer Vision Tool

Description: The system should be able to review the data being gathered from the lecture it was enabled for and pick out useful information from it before presenting this in a legible format for users to be able to read and understand.

Criticality: The generation of the insights is essential to our system, as it is one of the key tenets of our project's application. The system must be able to determine the relevant information gained from the use of the computer vision tool.

Technical Issues: We will need to make sure that the results are insights which will be relevant to the users, as providing them with too much irrelevant information will make it more difficult to determine what worked and what they need to focus on in the future.

Dependencies with other requirements: This requirement is dependent on 3.6 - Enable Computer Vision Tool, to enable the tool which allows for the insights to be generated. It is

also a dependency for 3.8 - View Insights Generated by Computer Vision Tool, as the insights must be generated before they can be presented in a useful manner.

3.8 - View Insights Generated by Computer Vision Tool

Description: Users should be able to view the data insights generated by the computer vision tool once the lecture has concluded. This will allow the lecturer to know what parts of the lecture kept student attention best, and which parts had the most students failing to pay attention, and the lecturer could potentially adapt their future content based on this information.

Criticality: Viewing the insights generated by the computer vision tool is essential to our system, as it also forms the basis of the project's application. If users cannot easily access the data insights generated by the lecture, there will be little application of the tool.

Technical Issues: We will need to make sure that the insights generated by the tool are shown in an understandable format so that users can easily take note of key information from the insights. This may require the creation of graphs based on the data, or highlights based on unusual data.

Dependencies with other requirements: This function is entirely dependent on the 3.6 - Enable Computer Vision Tool and 3.7 - Generate Insights using Computer Vision Tool, as without the tool being enabled and generating insights there would be no data to view. It is also dependent on 3.9 - View Lecture Details, as it is important to have an easy way to access the data insights of previous lectures.

3.9 - View Lecture Details

Description: Users should be able to view the details of their past lectures through the timetable, including the data insights generated by the computer vision tool if it was enabled for that lecture.

Criticality: This function is highly important for the system, as it allows the user to utilise the data insights generated by the computer vision tool, while allowing the user to see particular information from a lecture that they may need, such as the location or relevant module.

Technical Issues: We need to ensure that all relevant information is listed and formatted well to ensure that users can easily navigate through the information to find what they're looking for.

Dependencies with other requirements: This functionality is dependent on 3.4 - View Timetable, as it needs a simple method to see a list of the lectures and view their details from there. It is also a dependency for 3.8 - View Insights Generated by Computer Vision Tool, as it provides a simple method for showing relevant insights gained from a particular lecture.

3.10 - User Logout

Description: Once a user is finished using the system, there should be an easily accessible option for them to log out of the system.

Criticality: The ability to log in is extremely important to the system, as it helps to ensure that accounts are not compromised while keeping the user's private information confidential.

Technical Issues: The system must ensure that account login information is not automatically stored locally after a user logs out, as this could lead to account compromisation.

Dependencies with other requirements: This requirement is dependent on 3.1 - User Registration and 3.2 - User Login, as there must be an existing account logged in to allow for the user to exit the account.

3.11 - Take Attendance

Description: The user shall be able to manually input either a number or checkmark the names of students that are in attendance in person.

Criticality: The user should have the option to take attendance however an estimate of numbers can be given by the CV tool, and marking attendance is not required, it is up to the personal preference of the user whether to use this feature

Technical Issues: The system must be able to access the details of a class group to get numbers and participants, then create an attendance sheet for the specific lecture in the timetable

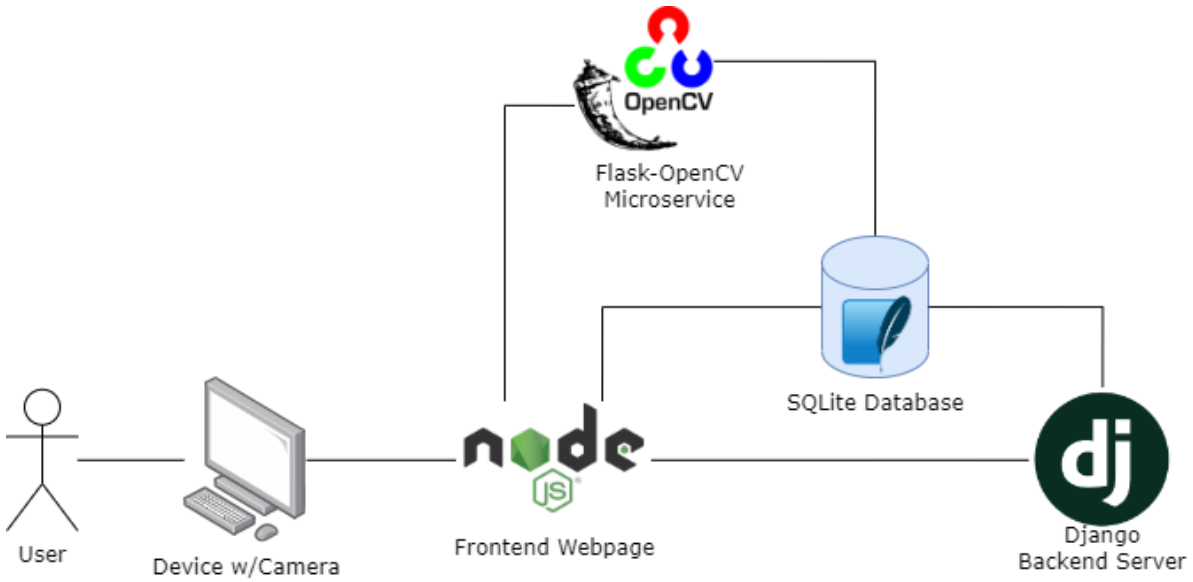
Dependencies with other requirements: This requirement is dependent on 3.4 - View Timetable, 3.9 - View Lecture Details and 3.3 - Class Group Management as the user will need to be able to select a particular lecture, and from there be able to select the students which attended that particular lecture, with the information showing up in the lecture details.

4. System Architecture

At a high level, our system comprises four main components that will need to be developed:

- Our Frontend, a web page made using NodeJS
- The Backend, a server using the Django Framework
- A Database, an SQLite database connected to our backend
- The Microservice, a flask microservice that will use OpenCV

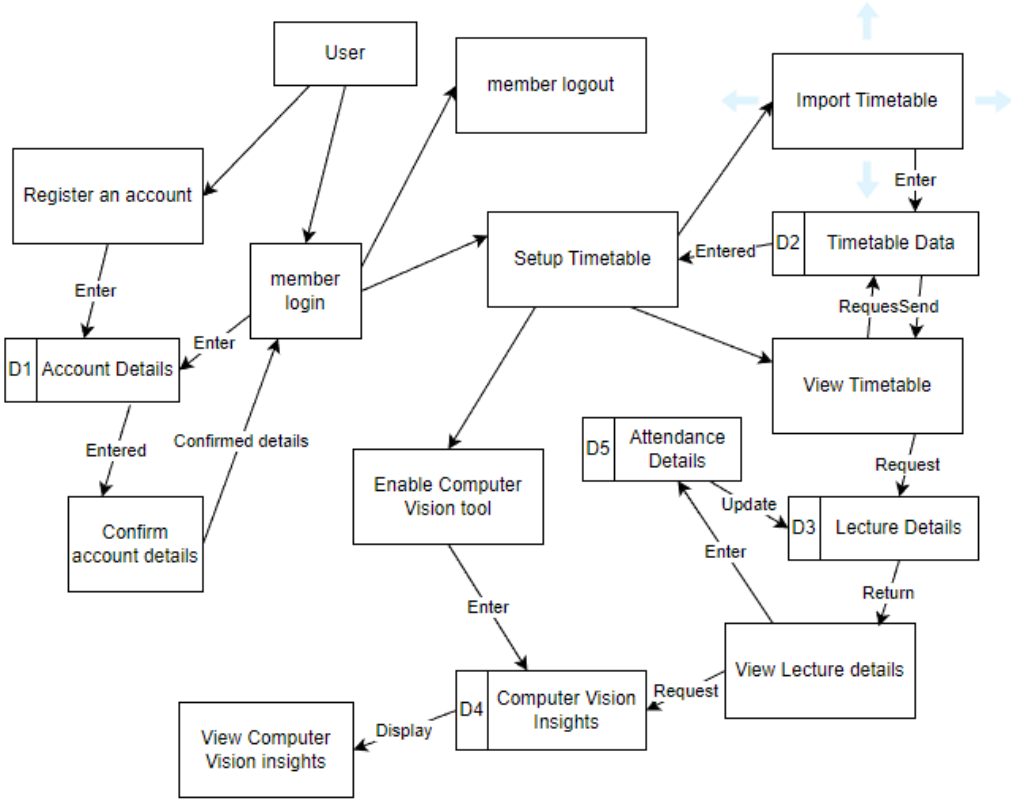
The user will be able to interact with our frontend web page, this will be a nodeJS web page written in HTML, CSS & JavaScript. Our backend Django server will contain models and functionality for the users to be able to edit and set up classes, view their timetables etc. Our computer vision programs will be contained within a flask microservice that can be used to analyse incoming video data when the user activates that feature. All the data and models will be stored in our SQLite Database via our backend.



5. High-Level Design

Shown below are some models that display some of the preliminary high-level design of the system, you can see first a data flow diagram that shows the flow of data through our system.
There is also an example sequence diagram which shows the process of what happens when a user wants to start the CV analysis tool

Data flow diagram



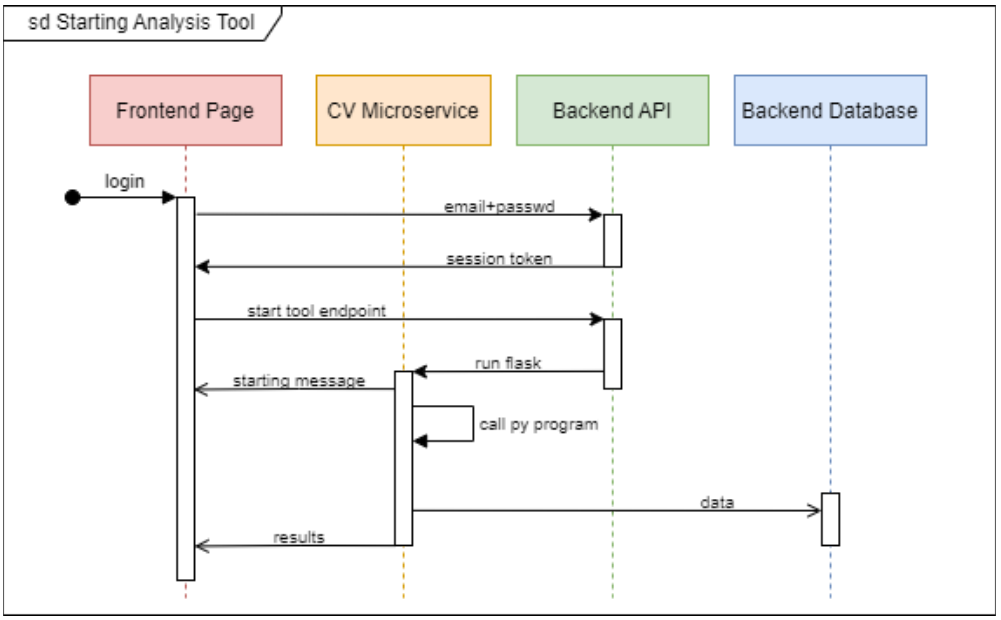
The user begins by either attempting to register an account or logging in to an account, both of which require the user to enter account details. This data is taken and checked by the system, and if the details are confirmed to be correct the user is allowed to move on to access other parts of the system.

The user can attempt to import a timetable, which has the system entering the timetable details into the system before returning to the user. When the user enables the computer vision tool for their lecture, it generates the data before entering it into the system to be viewed later.

When the user attempts to view their timetable, it requests the timetable data that has been entered and sends it to the user's screen. From the timetable, a user is able to request to see the details for a particular lecture, which leads them to a screen where they can view all of the lecture information.

From this screen they are able to update the attendance details for a particular lecture, which enters the data into the system before it is added to the lecture details, bringing the user back to the same page. They are also able to request to see the computer vision insights generated for that particular lecture, which the system then requests before displaying for them.

Sequence Diagram



6. Preliminary Schedule

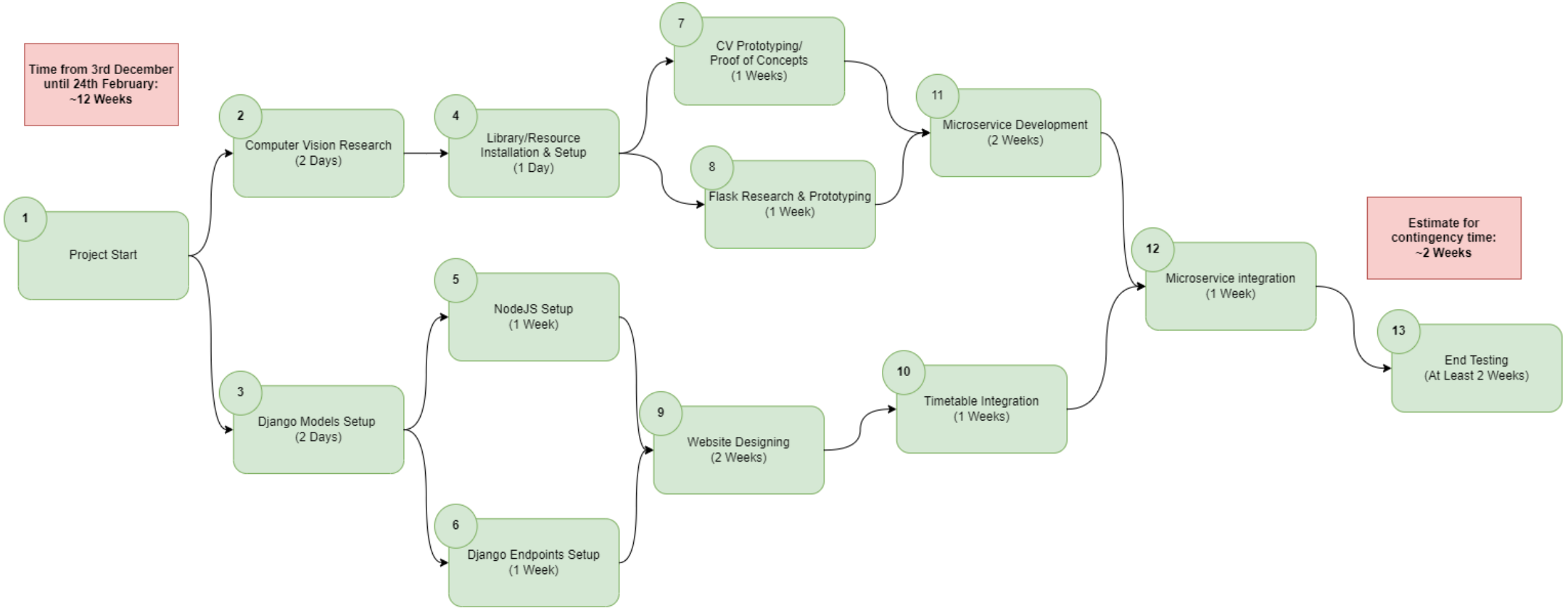
Provided on the next page is a preliminary PERT chart that we used to map out the main tasks we believe need to be completed and in what general order they will be done. Our project has two main paths to be completed, one for each of us to do.

Gareth will focus on the Computer Vision path along the top of the chart, researching and developing the techniques we can use to implement our microservice into the overall system, this part of the project will be more time-consuming at the beginning, as we learn the technology and libraries before getting down to prototyping and implementation.

Josh will focus on the web page aspect of the system shown in the bottom half of the chart, this will have a much quicker setup time but might take more time overall especially near the end of the chart when designing and implementing a good UI for the users to interact with.

From the time of submission of this specification to the final due date of the project we have about 12 weeks to get everything done, in our PERT chart we have not scheduled work all the way up until the submission, we have left a lot of contingency in our plans to account for potential problems, other commitments and unforeseen circumstances. We hope however to get the bulk of the work done early, that way we have a lot of time at the end to rigorously test and polish our final system.

Our PERT Chart



7. Appendices

Here are some links to things discussed in this document, and further reading on topics such as Computer Vision and the tools we will be using.

- IBM guide to Computer Vision:
  https://www.ibm.com/topics/computer-vision

- Link to the OpenCV website:
  https://opencv.org/

- YOLO from Darknet: Open Source Neural Networks in C
  https://pjreddie.com/darknet/yolo/

- An Introduction to YOLO with OpenCV using Python
  https://towardsdatascience.com/yolo-object-detection-with-opencv-and-python-21e50ac599e9

- Introduction to the YOLO algorithm for computer vision
  https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/

- DCU's Open Timetable website:
  https://opentimetable.dcu.ie/