## **Current System**

The current Over Surgery system is very basic, it is all on paper and involves a lot of tireless searching through ring binders to find things. When looking for patients they must search physically, and to create new appointments or register new patients they must write it all by hand.  
There are a number of problems with using a paper system, such as:

* Paper can easily be damaged and destroyed.
* It is slow and is prone to mistakes when searching.
* Employee handwriting can vary and some is hard to read.
* Takes up a lot of space in the office and can become messy.
* Costs a lot to upkeep all of the stationary required.
* There is a learning curve to working the reception, it takes time to learn where things are. This means new reception employees can be slow for a long time.

## **Requirements**

One of the requirements is a login system that allows the user to securely have access to the system and reject those who do not have the authorization to use the system. The system will also have a feature which will allow the user to be able to check the current staff that is on duty; furthermore it will display the availability of each individual, for example if one of the doctors is working on Friday and available at 12:30pm. The user will be able to register patients that are new to the surgery and will be given a unique ID so they can be identified in the future. From the information the patient has registered such as name, date of birth and auto generated ID this will be used to identify a patient to display their prescriptions, appointments and tests. Whilst also allowing the user to extend desired prescriptions. Another feature is that the user will be able to book appointments for the registered patients on the system, however if the appointment was not able to be attended for any given reason or need to be rescheduled the appointment will have an option to be deleted or changed.

## **Project Aims**

The aims of this project is to create a system for Over Surgery that will streamline the process of managing appointments, availability of staff, registering patients and viewing relevant information of the patient such as tests and prescriptions. The login system will have a username and password entry for the user to verify they user has authorized access to the system. After the user has been verified, the dashboard form will be displayed and give options to the user to select which functionality of the system they require. Such as appointment, registering a new patient, availability and so on. The register form will have fields for patient information, including their title, first name, allergies. A submit button will validate and send all information to the database.   
There will be a results form, which will search for the patient with either the unique patient ID or by their name and date of birth. Once the patient is selected it will display the patient’s appointments, prescription and tests. From here it will allow extension of the patient’s prescriptions, given that the duration of the previous prescription is completed and will update the database accordingly.  
The appointment form will give the user the choice to either enter a new appointment for a patient or search for an existing appointment and delete it or modify it. Lastly there will be a form for Availability that requests the user to select a date, from this it will display the staff that is scheduled to work on the selected date; when one of the staff is selected, the availability panel displaying “available” or “not available” for different times throughout the working day will change in line to when the staff member if free.

## **Resources**

**GitHub** – We tracked our work and shared files using GitHub. This was the best way to keep each other up to date on each line we had changed.

**Slack** – Slack is an industry used communication platform, we have found that it encourages professionalism more than conventional social platforms for messaging.

**Git plugin** – we used a git plugin for slack, this allowed us to integrate conversations with GitHub. Whenever anyone committed or edited an issue, a message was sent to the slack thread. It aided with communicating around work flow.

**AdobeIllustrator** – We used adobe illustrator to build the UI design in a digital format**.**

**Visio & Draw.io**– All of our design diagrams where completed using Visio or draw.io. This is mainly because we all have previous experience with the software and we all have access to it.

**Visual Studio –** All of our development was completed in the Microsoft Visual Studio IDE.

## **Team Management**

We thought it would be best if people who had ideas/designs for elements of the code to perform on them and build those parts. This is so that they could further their ideas creatively as they built upon them. We knew that we would want to separate the front and back end of the project. To keep these separate we had different people coding these items. It was like assigning people to back end developer roles. These people had best understanding of their positions but could still expand outside and work on other elements of the project.

## **Time Management**

We decided we needed to build a Gantt chart in week 2. This was an estimated time management tool. It was not followed strictly but served as a guide for when tasks should be completed and started. 

Figure , Gantt chart for planning task timings.