Speedy Recovery

A patient-centred app based on the FHIR standard facilitating communication between paediatric patients, parents and hospital staff

Requirements and Architecture Analysis

Authors

Ahmed, Fabiha

Boecker, Niels

Huang, Ying

Meng, Fanbo

Wong, Ka Wai

Yang, Chong

Zeltser, Joshua

{ucabbah, ucabnbo, ucabcya, ucabuae, ucabfme, ucabkwo, ucabjze}@ucl.ac.uk

Version History

Version	Date	Comments
0.1	20/11/18	Business goals, stakeholders, system context, requirements (first draft)
0.2	27/11/18	Architecture models, introduction, background, revisions after TA's feedback
0.3	04/12/18	Requirements (security, quality, performance and scalability, availability and resilience), revisions after TA's feedback
0.4	11/12/18	Implementation, roadmap, future work, improve structure, revisions after TA's feedback
1.0	23/12/18	Improve structure, improve figures, proofread, revisions after project presentation, improve layout, rephrase and prune text

Table of Contents

1. Introduction

This report outlines the requirements elicitation and architectural analysis of *Speedy Recovery*, a HIT (Health IT) application that provides a treatment calendar to patients, their parents, and clinical staff in separate UIs.

GOSH (Great Ormond Street Hospital) and its digital unit GOSH DRIVE (Digital Research Informatics and Virtual Environments) are committed to making young patients' hospital stays as tolerable as possible. Inherently, this goal can only be achieved if parents are assured and confident, and hospital staff is well-informed and approachable.

We aim to use modern technology to help GOSH in their pursuit of their goals and generally explore the potential of patient-centred software in healthcare. Therefore, we will develop a web app that serves three different user groups: patient, parent and staff. The app will provide a calendar that is tailored to the respective user's needs. All three groups have very different demands and expectations, which makes our project interesting from a user experience perspective.

Currently, the communication between clinical staff, patients and their parents is very similar to how it was 50 years ago, not making use of modern technologies. Many manual steps are required. For example, parents have to call hospital staff to request an appointment, the staff will have to find an available time slot in their computer system, and then eventually the parents will copy the appointment to their calendar. If any of the parties would like to change the appointment, all steps have to be repeated. With the help of our app, all involved stakeholders would save time and the whole hospital experience would become much more streamlined and tolerable.

Also, we will make use of the cutting-edge healthcare data standard FHIR (Fast Healthcare Interoperability Resources), which is still in its draft phase, but is believed to revolutionise the HIT landscape. Other than the growing pains of this young, evolving standard, we expect the technical challenges to lie in the access control, authorisation and authentication, as well as using the same data for three distinct deliverables. By solving these challenges and evaluating the data standard, we will create valuable insights for our client, on which future student projects or research efforts at GOSH can build upon.

This document is intended to serve as a tool for planning and documentation, facilitating the development phase. Hence, the intended readers are mostly software developers. However, we also use the document to confirm that our understanding of the project is in line with our client's.

2. Background

This report and the corresponding development project are the proceedings of a student project at UCL (University College London), in the context of the digital unit of GOSH: DRIVE was launched in August 2018 and focuses on research and evaluation of new technologies and data analysis for healthcare. It represents a partnership between GOSH, UCL and multiple leading technology companies [1].

One of the topics that DRIVE currently explores is FHIR, a draft standard in healthcare integration and interoperability. It is commonly believed to become the most important standard in this domain and replace old, outdated standards like HL7 v2 and v3 [2]. As this standard is currently being introduced to GOSH and is still in the exploration phase, DRIVE as our client did not have a specific project for our student team. Instead, we developed our project idea hand-in-hand with the DRIVE development team. Finally, we settled for an idea that had, in rough strokes, been proposed at the FHIR DevDays Conference in Amsterdam in

November 2018. In her keynote *The Patient Experience* [3], Janet Campbell presented FHIR-based healthcare applications that are in demand, but do not yet exist. One of them was a patient-centred app that would allow patients to see their health records in a readable and appropriate way.

Based on this draft and taking into account that GOSH is a children's hospital, we developed the idea of an app that would show a patient's calendar in a child-appropriate manner, while also offering adjustable views for parents and hospital staff. We envision the patient's view to be simplistic and colourful, showing pictures of the providers and only limited information. The parent's view would include additional information as well as a feature to message the hospital staff, potentially requesting a rescheduling of appointments. The hospital staff would see and access this information in the highest level of detail.

3. Requirements analysis

3.1 Business goals

GOSH, as one of the world's leading children's hospitals, is constantly trying to meet the many goals its stakeholders require. These goals range from delivering world-class services to attracting top doctors from across the globe and need to be constantly analysed to ensure that the hospital is moving in the right direction. GOSH DRIVE also has a large set of goals, which focus on assisting GOSH in working towards its aims. Our app is being created to assist GOSH DRIVE to achieve its main goals and this will, in turn, help GOSH's top-level goals come to reality. In this section, we provide an in-depth analysis of the goals of both GOSH [4–6] and GOSH DRIVE [7].

3.1.1 GOSH

- Provide high quality and safe care to patients
  - GOSH aims to fulfil its motto of "The child first and always", in order to tailor the experience that a patient has at the hospital to the child.
  - GOSH aims to make the experience as seamless and easy as possible for parents as well, allowing them to focus on supporting their child.
  - GOSH would like to see a reduction of harm to zero.
- GOSH aims to exceed the expectations of patients, families, and deferrals as they carry out treatments and procedures at the hospital.
- Maintain world-leading position (Top 5)
  - GOSH aims to be able to have the tools to consistently deliver world-class outcomes in terms of both patient care and customer service.
  - GOSH aims to attract media attention.
  - GOSH aims to maintain the status of International Centre of Excellence in paediatric health care.
- Stay on the edge of research
  - GOSH aims to stay at the forefront of medical research.
  - It will do so by cooperating with academics and other hospitals e.g. UCLH etc.
- Offer service to as many children as possible
  - GOSH aims to maintain its role as the largest paediatric centre in the UK for certain areas.
  - GOSH aims to make it the place to send children with health problems.
- Offer a wide range of health services in one place

  - GOSH aims to offer specialist services for children with complex conditions to treat them in one place.
  - GOSH aims to reduce the need to travel between hospitals for separate treatments.
- Attract high-potential employees
  - GOSH aims to attract some of the best qualified doctors and practitioners from around the world.
  - GOSH aims to attract staff who are dedicated, hardworking and motivated by GOSH's main goals.

3.1.2 GOSH DRIVE

- Research and use modern technologies to improve the patients' experience
  - DRIVE aims to take into account the main stakeholders at GOSH, including: children, parents, clinicians, admin staff and nurses.
  - DRIVE take the advancements from the consumer space and transfer them into healthcare.
  - DRIVE identify real-life interactions by studying the hospital environment and attempt to digitally enhance the experience.
  - DRIVE aims to create tangible and measurable improvements to all aspects of the hospital.
- Bring together clinical staff, academics, and industry
  - DRIVE cooperates with giants in the tech industry to further research and improvements in patient care.
  - DRIVE encourages students and academics to conduct projects to further research and improve the overall hospital experience.

3.2 Competitor comparison

GOSH has many competitors who are other children's hospitals. The following three examples could be considered as competitors of GOSH.

3.2.1 Boston Children's Hospital

Boston Children's Hospital is a world-renowned children's hospital specializing in paediatric care in the US. Like GOSH, they are trying to integrate technological innovations and excel in research.

- Reputation: Was ranked top in 8 out of 10 clinical specialties by the U.S. News & World Report, and as the nation's number one paediatric hospital for 2018/9 [8]
- Apps: Boston Children's Hospital owns more than 20 Apps for patients, families, staffs and doctors (e.g. Pediatric Growth Charts, Boston Children's Hospital, Boston Children's MyChildren's, etc) [9]

3.2.2 Evelina London Children's Hospital

Evelina London Children's Hospital is a specialist NHS hospital in London. It can be considered a competitor because of its geographical closeness and similar profile.

- Location: Evelina London Children's Hospital is only two miles away from GOSH, so the parents who live nearby could take their kids to either of these two hospitals.
- Long history: Evelina London Children's Hospital was founded in 1869, its long history brought it great reputation.

- Paediatric Communication App: The Paediatric Communication App enables children who have difficulty speaking to communicate with hospital staff and their families about their treatment. Launched at Evelina London Children's Hospital, the app has pictures which children can use to say what they want or how they feel. They can let staff know where they are feeling pain and how much by using a 'feelings' option. [10]

3.2.3 Birmingham Children's Hospital

Birmingham Children's Hospital is the UK's leading specialist paediatric centre, caring for sick children and young people. Like GOSH, they try to excel in paediatric medicine in the UK.

- Reputation: Birmingham Children's Hospital is ranking 4th in the 30 most technologically advanced children's hospitals in the world. [11]
- Feedback App: Birmingham Children's Hospital has their own app, namely Feedback (myBWH), which connects patients, families and staffs in a successful way. [12]

3.3 Stakeholders

GOSH as a hospital has a large number of stakeholders that are affected by their everyday operations. Whenever a new project is started at GOSH, they must evaluate which stakeholders will be affected and whether the requirements can be tailored to their needs. As we embark on creating our app, we must explore which stakeholders are both positively and negatively affected, before any design decisions are finalised. The table below contains an in-depth analysis of the various stakeholders affected by our product.

Stakeholder	Type	Requirements
GOSH	Sponsor	• Efficiency: Want to be able to improve the efficiency of the medical staff. • Reputation: Want to continuously enjoy very good reputation • Patient satisfaction: Want to be able to increase the satisfaction rate of patients and their families. • Cost: Want to spend less money on development and maintenance.
GOSH DRIVE	Sponsor, technical assistance	• Utility: Evaluate whether the application is needed in clinical practice or can be used to support their current development. • Extendibility: Want our app to be easy to extend or modify for future work. • Maintainability: Want our app to be easy to maintain and with minimal expenditure on application maintenance.
System administrator	Operator	• Ease of use: Want to be able to create and maintain users easily and efficiently • Error prevention: Want the application to confirm the actions that are difficult to undo
Patient (child)	User	• Ease of understanding: Want to understand the state of illness and treatment procedure, in a simple and clear way.

Stakeholder	Type	Requirements
		• Ease of operation: Want to be able to use the application easily.
Family member (usually parent)	User	• Ease of understanding: Want to understand the state of illness and treatment procedure in a clear way. • Ease of operation: Want to be able to use the application easily. • Privacy & security: Want all personal information (e.g. patient's condition or NHS number) to be kept private, with only people with the appropriate clearance level accessing it.
Clinical staff	User	• Ease of work: Want to be able to check or keep track of patient's conditions easily and efficiently.
Web application developer	Developer	• Technical feasibility: Want to have enough skills to use the required technologies • Schedule feasibility: Want to have clear plans and feasible roadmap for the application • Application quality: Want to be able to develop an application that satisfies the clients
Other medical systems in NHS	Adjacent systems	• Ease of communication: Want to be able to exchange medical information easily and securely
Regulators	Wider environment	• Need the application to comply with privacy and accessibility laws

3.4 The system context

One way to clearly outline how a system works is through using a context diagram. The context diagram shows us how "the Machine", or app in our case, understands the context in which it operates and how it relates to the world around it.

Before the creation of our app, the context diagram in Figure 1 shows the interactions of the current system with users and other systems. The main GOSH system receives information from external doctors and is updated constantly by staff and doctors as patients are treated. The staff and doctors interact with the parents of patients, while the patient mainly interacts with the doctors themselves as they receive their treatment. As mentioned in our introduction, this process could be simplified, and the experience of the patient improved by our app.

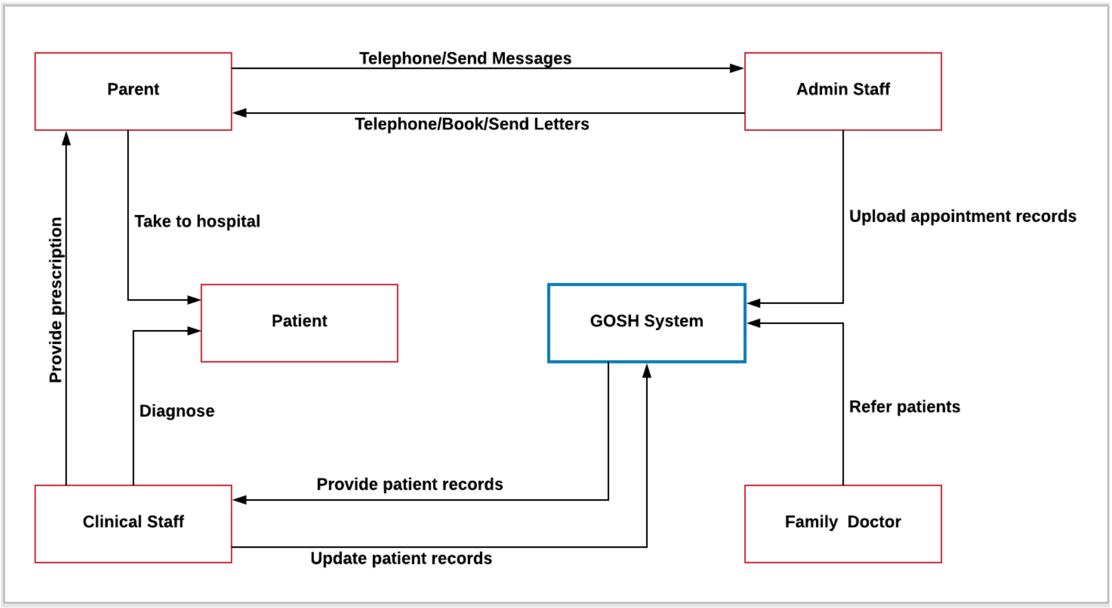

Figure 1: Context diagram depicting the state of GOSH's system before our project

With the creation of our app, this portion of GOSH's infrastructure will be improved, as seen in the context model in Figure 2. Speedy Recovery will now be the main source of information about a patient for staff, parents and the patient themselves. The app's three different levels of abstraction of the patients' data will be provided to the relevant stakeholders. In section 4.3, the different views are explained in detail.

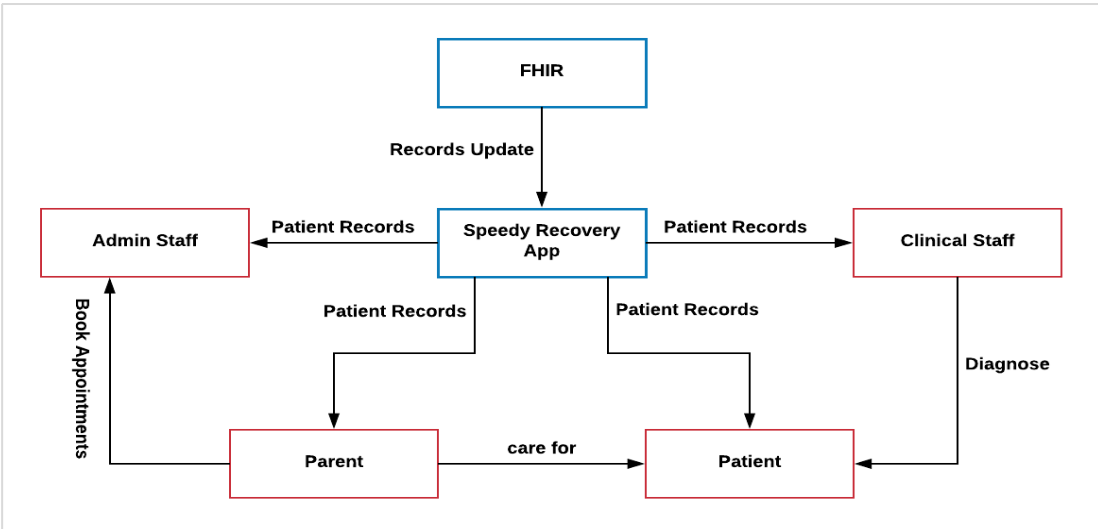

Figure 2: Context diagram depicting the state of GOSH's system after our project

3.5 Overview of requirements

In order for us to create an application that meets all of the needs of our stakeholders, it is necessary for us to create a list of requirements that our project must abide by. These requirements cover all areas of the app and ensure that all of our top-level goals are met by the end of our project. As we carry out the development phase in Term 2, we will be constantly checking this list of requirements to ensure that we are heading in the right direction.

3.5.1 Functional requirements

The following definitions describe the main functional requirements:

3.5.1.1 Sign-up process

REQUIREMENT Achieve [Send invite links to staff]
Responsibility System admin
GIVEN a staff member has joined the hospital
AND Speedy Recovery (back-end) has access to the email addresses of staff members sourced from FHIR resources
WHEN a system admin has verified the email address and name of the staff member in the database
THEN the staff member an receives an email with an invite link to Speedy Recovery (app)

REQUIREMENT Achieve [Send invite links to parent/patient]
Responsibility Administrative staff
GIVEN an administrative staff member has an account on Speedy Recovery (back-end)
AND a patient is currently under the care of GOSH
AND Speedy Recovery (back-end) has access to the email address of parent or patient sourced from the FHIR resources
WHEN the administrative staff member has verified the parent/patient
THEN send the parent/patient receives an email with an invite link to Speedy Recovery (app)

REQUIREMENT Achieve [sign up for an account]
Responsibility User
GIVEN a user has received an invite link to Speedy Recovery (app)
WHEN the user has clicked the link and created a password
THEN Speedy Recovery (back-end) service creates their account

3.5.1.2 Login and account management

REQUIREMENT Achieve [Login to an account]
Responsibility Speedy Recovery (back-end)
GIVEN a user has an email address and password for their account
WHEN the user enters their email address and password on the login page
AND the Speedy Recovery (back-end) has verified the email address and password
THEN Speedy Recovery (app) presents the homepage of the account

REQUIREMENT Achieve [Remove account when staff member resigns/retires]
Responsibility Speedy Recovery (back-end)
GIVEN a staff member has an account
AND the staff member resigns/retires
WHEN a system admin has verified that a staff member has resigned/retired
THEN the account will be removed by Speedy Recovery (back-end)

REQUIREMENT Achieve [Reset account password]
Responsibility Speedy Recovery (back-end)
GIVEN a user forgets their password
WHEN user enters their email address on the lost-login-details page
AND the email is verified with the database
THEN the back-end service should send an email to the user containing a link to reset the password

3.5.1.3 Display of information

REQUIREMENT Achieve [Display appropriate level of information to the user]
Responsibility Speedy Recovery
GIVEN a user has registered with Speedy Recovery
AND Speedy Recovery (back-end) has access to the patient's records sourced from FHIR
WHEN the user attempts to check the patient's record on Speedy Recovery
THEN the app displays the appropriate level of content for the user

REQUIREMENT Achieve [Display patients' record details]
Responsibility Speedy Recovery
GIVEN a patient/parent has registered with Speedy Recovery
AND Speedy Recovery (back-end) has access to the patients' records sourced from FHIR
WHEN a parent/patient attempts to check the patient record in Speedy Recovery (app)
THEN Speedy Recovery (app) displays the details of the patients' records to the parent/patient

REQUIREMENT Achieve [Display relevant information on the calendar for staff]
Responsibility Speedy Recovery
GIVEN a staff member has registered with Speedy Recovery
AND Speedy Recovery (back-end) has access to the patient's records sourced from FHIR
WHEN the staff member attempts to check the patient's records on Speedy Recovery (app)
THEN Speedy Recovery (app) displays the staff's relevant information in a calendar format

REQUIREMENT Achieve [Display relevant information on the calendar for patients]
Responsibility Speedy Recovery
GIVEN a patient has registered with Speedy Recovery
AND Speedy Recovery (back-end) has access to the patient's records sourced from FHIR
WHEN the patient attempts to check their records on Speedy Recovery (app)
THEN Speedy Recovery (app) displays the patient's relevant information in a calendar format

REQUIREMENT Achieve [Display relevant information on the calendar for parents]
Responsibility Speedy Recovery
GIVEN a parent has registered with Speedy Recovery
AND Speedy Recovery (back-end) has access to the patient's records sourced from FHIR
WHEN the parent attempts to check the child's records on Speedy Recovery (app)
THEN Speedy Recovery (app) displays the child's relevant information in a calendar format

3.5.1.4 Messaging

REQUIREMENT Achieve [send a message to a parent/staff]
Responsibility Speedy Recovery
GIVEN a parent selects a staff member to speak to
OR a staff member selects a parent to speak to
WHEN a parent/staff member sends a message to the other
THEN the parent/staff member should receive this message

REQUIREMENT Achieve [Message can be displayed on parent and staff's chat box]
Responsibility Speedy Recovery
GIVEN a parent and staff member have started a chat
WHEN a parent/staff member sends a message to the other
THEN both parent and staff member should be able to view the messages

REQUIREMENT Achieve [notify the user about unread message]
Responsibility Speedy Recovery
GIVEN a parent/staff member has an unread message
WHEN a parent/staff member logs on to Speedy Recovery (app)
THEN a notification box for unread messages should pop-up at the top of the page

REQUIREMENT Achieve [Display message history]
Responsibility Speedy Recovery
GIVEN a parent and a staff member have communicated before
WHEN a parent/staff member opens this chat
THEN the history of messages between the parent and the staff member should be displayed

3.5.2 Quality requirements

3.5.2.1 Security

A. Security analysis

(1) Sensitive assets and security goals

Our app will be processing lots of sensitive information, for example, a patients' records and some of the users' personal account information. The privacy of this information is very important. Therefore, it is crucial to assure that the security goals of confidentiality, integrity and availability are always achieved.

This table shows some sensitive assets that our app contains:

Asset	Sensitivity
Patient's record	All patient records are confidential and are required for the patients' treatment
Clinical staff's account detail	Account details for clinical staff members which controls who can access patient data. Theft will violate personal privacy.
Patient's account detail	Account details for the specific patient, specifying who can access patient's records. Theft will violate personal privacy.
Parent's account detail	Account details for the patient's parent who can access their son/daughter's record. Theft will violate personal privacy.

(2) Threats

As part of this project, we will consider any security goals that may be required by our stakeholders and will select some security policies to help meet them. However, there are still some threats that our app may be vulnerable to. This table lists some of these threats and security requirements that we could use to avoid them:

Threats	Security Requirements
Clinical staff sells the patient's data without permission, and patient's records end up leaking to the third parties	Avoid [copying the content from the app]
Unauthorised external access to the application	Avoid [Unauthorised External access to the app]

Threats	Security Requirements
Users forget to sign out from the system, the patient's detail can be read by strangers	Achieve [Create a session for the login period, when session expires (15 minutes), system should log out automatically].
Vulnerabilities in the software that could be targeted by an attacker, e.g. SQL injection	Achieve [Filter invalid data before transfer to the database and integrate other security features]
Man-in-the-middle attack, when users want to login to the app and access patients' details	Achieve [all data transmission should be encrypted]
Data being shown to unauthorised people who do not have permission to access them	Achieve [strong authentication and verification procedures integrated into the app]

B. *Secure signups*

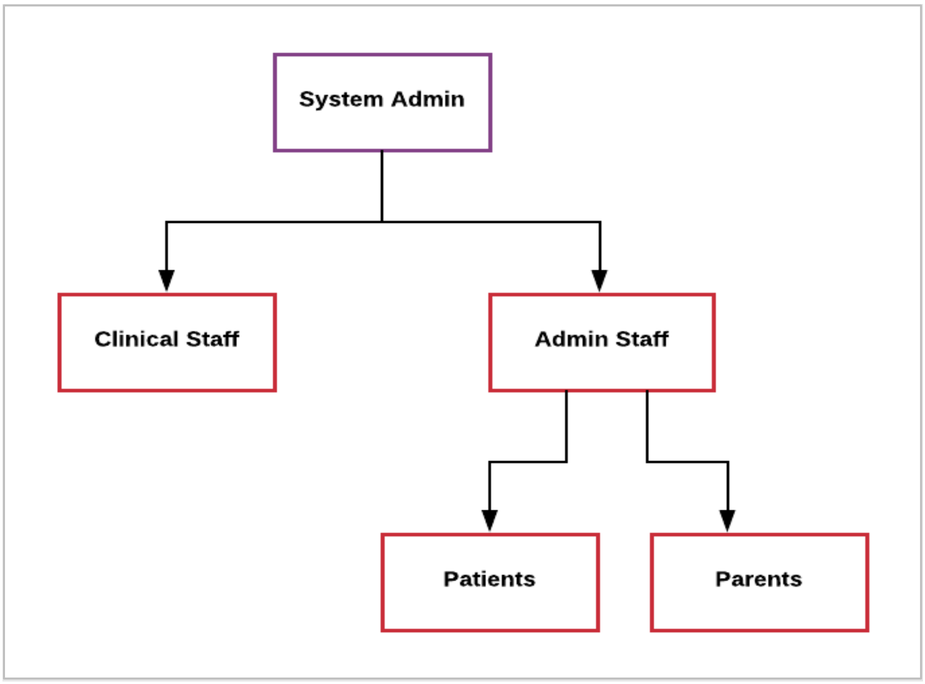

Figure 3: Secure signup hierarchy of GOSH's system

In order to ensure that security is maintained throughout the signup process, we have created a hierarchy that will be followed by the system as users sign up to our app (seen in Figure 3). The System Admin is in charge of the sign-up process. As clinical and administrative staff join the hospital, the System Admin will send them invite links to the app, allowing them to create an account. This will ensure that only verified staff at the hospital can set up these accounts. For the next part of the hierarchy, Administrative Staff will be able to send invite links to patients or to the parents of patients. This again ensures that only certain hospital stakeholders are able to access the system that our app provides.

These invite links rely on the fact that we have access to email addresses of staff and patients/parents sourced from the FHIR resources. This hierarchy will limit the number of people who gain access to our system and thereby to confidential patient information.

C. Access control system

Access control on our app involves the restriction of access to confidential information including controlling behaviour within the system for data security. The access control system contains the authentication functionality of this app. Meanwhile, it also should consider the permitted behaviour and resources for different users. The details of each part will be discussed in the following subsections.

(1) Authentication

Authentication is used to confirm users' identity in order to grant them access to the system. Speedy Recovery will use the common method of email addresses and passwords for its main authentication procedure.

The FHIR standard has no concept of security, but SMART on FHIR (Substitutable Medical Apps, Reusable Technology) specification can add an extra layer of authentication in a pre-determined fashion, and is, therefore, used to support the implementation of authentication on our app. SMART is a set of standards that can provide support for applications to run across the healthcare system securely. The specifications outline a robust authorisation model for apps based on the OAuth standard which keeps patients and providers in control of their data [13,14]. In terms of password management, OpenID Connect is used to allow app developers to authenticate users without taking on the responsibility of storing and managing passwords [15].

As for access tokens, Speedy Recovery should always invalidate the user session when the access token expires or sooner. If the user's complete session duration is longer than 15 minutes, the app must update the ID token, to ensure that the user is still valid and has the correct permissions [16].

(2) Resources classification level

Resources have different classification levels bases on the stakeholder that can access them. The level of resource that can be obtained by users is different depending on their role. For data security, each of our three views will be implemented to display information at an appropriate level of visibility. Our application provides a hierarchy where staff can see all information, parents can see a subset of this information, and the patients can see even less information (seen in Figure 4).

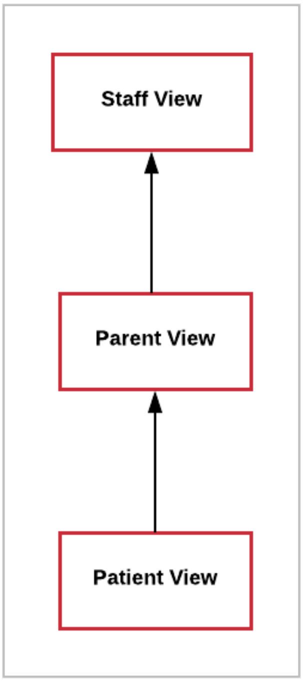

Figure 4: Diagram illustrating the visibility hierarchy

(3) Login process

Every user has an account which they are able to use to login to our app. Users will login using their email address and password to verify their identity as shown in Figure 5 and Figure 6. The figures show the successful and failed login scenarios.

Once users entered their email address and password in the login page, the account authentication controller will validate the login details with the user account, if the information is matched, the patient data controller will get the data from FHIR and display it in the user interface. If the account or password is wrong, the account authentication controller will reload the login page and require the user to enter email address and password again.

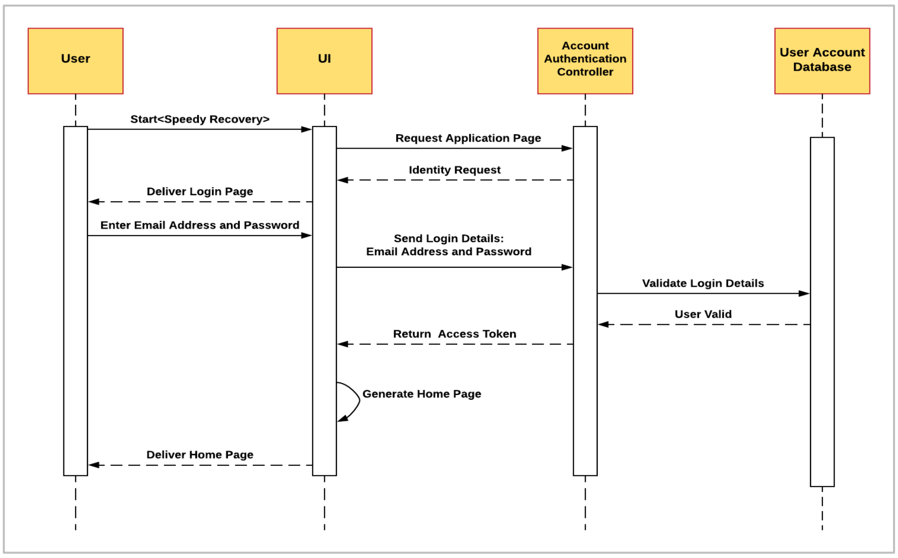

Figure 5: The sequence diagram for the scenario of successful login

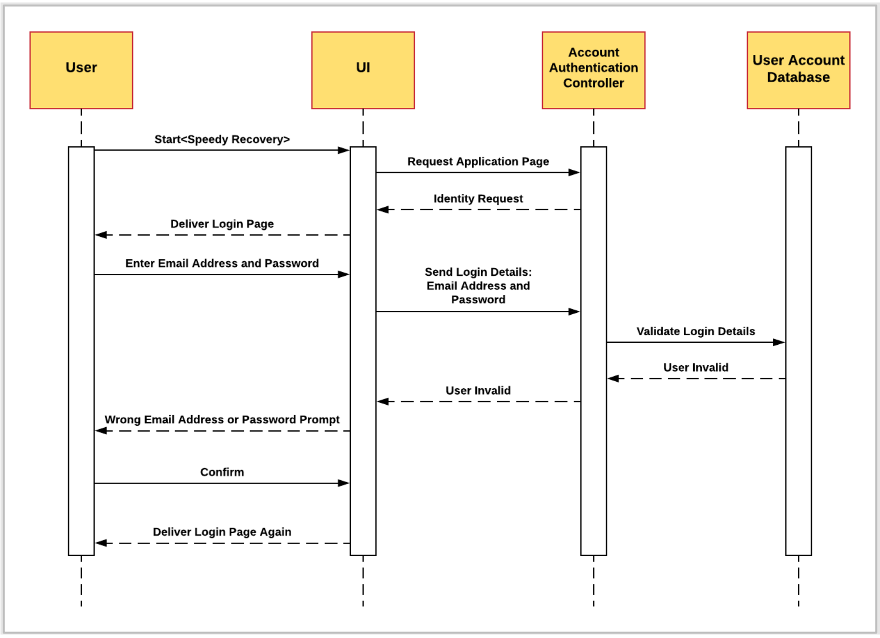

Figure 6: The sequence diagram for the scenario of wrong account ID or wrong password

3.5.2.2 Performance and scalability

A. Performance

(1) Response time

- Under normal load, the time needed to authorize staff/parents/patients should be less than 2 seconds in at least 90% cases.
- Under normal load, the browser render time (views of clinical staff/parents/patients) should be less than 1 second in at least 90% cases.
- Under high load, the time needed to authorize clinical staff/parents/patients should be less than 4 seconds in at least 95% cases.
- Under high load, the browser render time (views of clinical staff/parents/patients) should be less than 2.5 seconds in at least 95% cases.

(2) Throughput

- The system must be able to to process 10,000 requests for patient information (e.g. appointments) per minute.

Scalability

As the number of patients increases each day:

- The FHIR database should have enough storage to store all the patient's information.
- The system should be able to handle large number of concurrent users.

B. Performance bottleneck

- The web app first load: Usually the first load takes a longer time since the framework code and most of the app get transferred at this point. Later interactions will only cause very small loads, as only the updated data in JSON format will be transmitted.
- REST calls made to FHIR: The FHIR server might get overloaded with requests to provide patient information and cause system failure.

C. Architectural tactics for performance and scalability:

- Reuse resources and results (caching): By means of pre-loading core components of the app and caching results fetched from the back-end, load times will be decreased.
- Limit concurrent requests to FHIR: Limiting the number of REST calls made to FHIR can prevent the server from being overloaded and ultimately prevent system failure.

3.5.2.3 Availability and resilience

D. Availability

Goals (Services)	Availability Fit Criteria	Failure Handling Requirements
Achieve [Information from FHIR]	Crashes / no connection to FHIR: less than once a month	Caching can be used to save some contents; show error message, but make sure app is available and responsive
Achieve [Show patient information in web-app]	Downtime: less than 10 sec/week	Manually letting the patient/parent know about their schedule.
Achieve [Secure Login]	Downtime: less than 10 min/week	Backup user database.
Achieve [Messaging]	Downtime: less than 5 min/week	Backup message database.

E. Resilience

In order to minimise the effect of a system failure, the following things will be done:

- Backup patient information: Patient information can be backed up and kept redundantly in multiple secure locations by the hospitals providing the data from FHIR, to be recovered in the event of a system failure or other emergencies. This however is not within the scope of our system, rather it is the responsibility of the hospital providing the FHIR data.
- Backup databases: The transaction log of the databases that contain additional data to the FHIR resources (currently users accounts and messages) will be backed up so that all the login credentials and messages can be restored in the event of a system failure.

F. Architectural tactic for availability and resilience

Identify backup and disaster recovery solutions: Identifying what we need to back up (patient information, user and messaging database) will help protect and restore the data required to keep the system running in the event of a system failure. System recovery solutions such as transaction log can be used to restore the system to a point before failure to ensure availability.

3.6 Conceptual model

A conceptual model can be used to represent the mental picture we have of how a system should work. A good conceptual model can lead to software design that is easier to implement, *understand, and change. The conceptual diagram for the Speedy Recovery web application shows how we envision the system to work in Figure 7.*

Firstly, all the patient's records will be provided in the FHIR standard and tailored in the backend service. The administrative and clinical staff will receive all the patient's records to be used before diagnosis. However, since most of these records will not be relevant to the patients and their parents, they will only view a subset containing the relevant information.

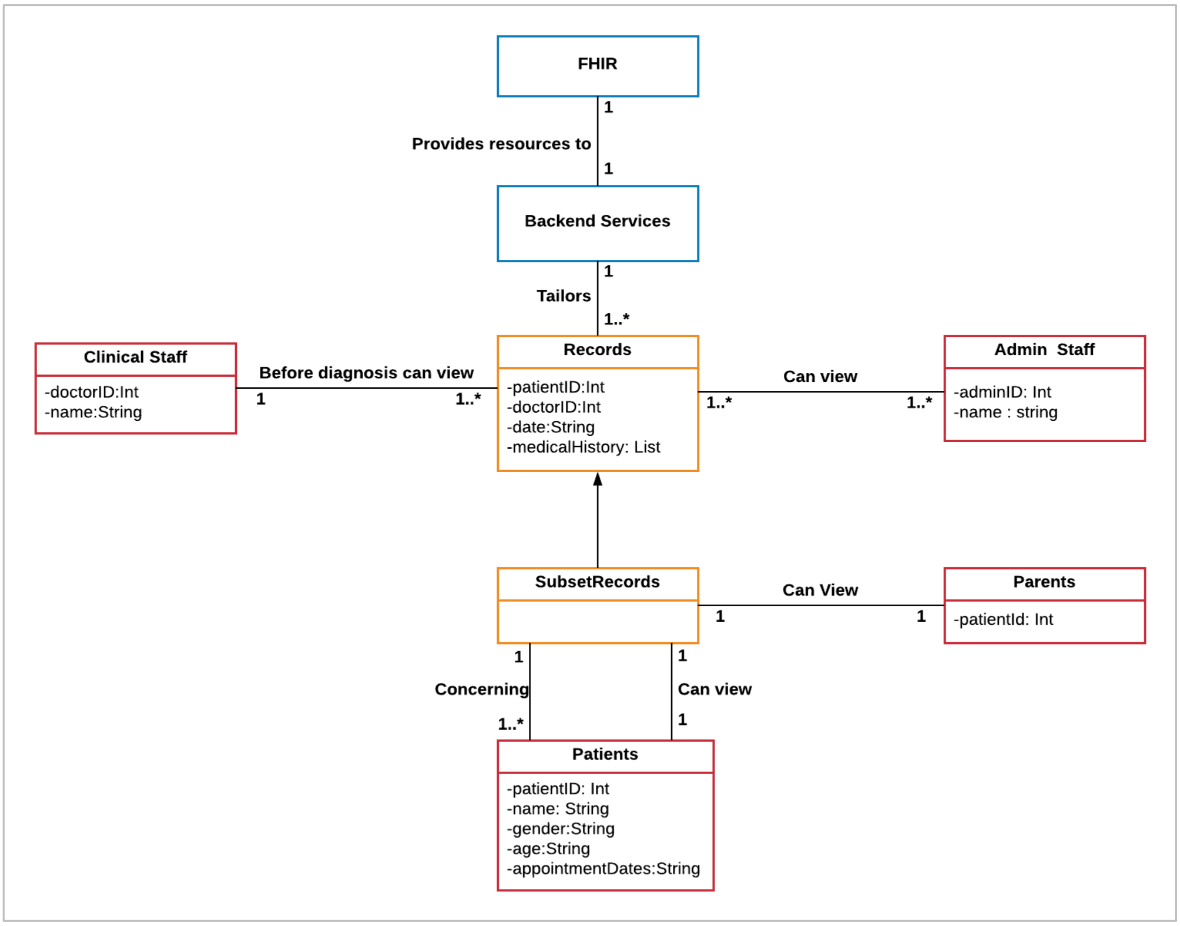

Figure 7: Concept Model depicting GOSH's system

3.7 Risks

Risk management is an important part of any software project. It involves identifying and analysing the risks that may arise as part of the development process and adopting reasonable countermeasures. By doing this, we will optimise the success of our project and it will help us to meet our goals. A risk is anything that might positively or negatively impact the project or its stakeholders [17]. We mainly consider the negative risks to this project, and we choose a broader approach that includes security threats as well as important usability issues we might likely encounter and potential implementation hinderances, helping us to identify the key areas to focus on in the development phase.

Project Components	Risks	Mitigation
Views	The users find the app hard to use	• Gather feedback from users • Modify the application according to feedback
	The patient interface is not suitable for children to view	• Investigate children 's preferences • Develop the patient view from a child's perspective • Pay attention to sensitive information
	The patient information displayed in different views is too vague/detailed or contains critical new information on users' health status	• Distinguish the information we need to display to the different participants (staff, parents, patients) • When determining which information stakeholders can see, consider the feelings of the patients (sensitive information)
Messaging	The message between parents and clinical staff is intercepted by third parties	• Use end-to-end encryption (e.g. Similar to WhatsApp [18])
Authentication	Authentication issues when users try to sign in	• The access control system should be developed with authentication tools like OAuth2 and OpenID Connect • Use a specific method to guide user's authentication (e.g. send authentication email links)
Privacy	The private data in this application is leaked to the public	• Set up the database protection mechanism • Maintain and check the security policies used in database regularly

	Patients are not willing to share their data	• Explain the use of private data when users use the application • Users need to agree to terms and conditions when they sign up
	The information of specific patients could be accessed by other doctors	• Build related database and ensure controllers extracts the correct data • Access control system should be considered properly in this project
Third party tools/participan ts	Third party tool (e.g. SMART on FHIR tools, React/Ionic) issues in this project	• Spend time setting up tools to ensure that they all work properly • Develop the application according to the official guidelines of these tools • Make sure the third party APIs only access the minimum necessary data • Make sure third party tools comply with terms and regulations
	The application cannot integrate into the other medical systems of the NHS	• Take into account requirements of integration in advance • Maintain the connection to these medical systems during development process
Schedule Constraints	The application cannot be completed on time	• Make a clear plan for each iteration • Reasonable division of labour to each team member
Resource Constraints	Some resources are not available for the application developers	• Prepare multiple plans for this project, including lots of research • Find other related resources as part of research
	Development work is disturbed due to the updates of FHIR versions	• Select a version of FHIR • Develop the application based on the fixed version • Leave integration with newer versions of FHIR to future work

4. Architecture analysis

In order to map out the software architecture of Speedy Recovery, in this section, we first present an overview of the functional coherences across all layers before describing each layer of the MVC-based architecture individually.

4.1 System overview

To understand the relationships and interactions between the components of Speedy Recovery, we created a functional view model. This view can be used to represent the functional elements and their primary run-time interactions. The functional view for our system can be seen in Figure 8.

Firstly, we state there are three types of users in our system (staff, patient, patient), and we provide a specific interface for each user. Each interface shows the appropriate level of content for their user.

All the transmitting data in the system will pass through three different controllers and store in different databases. The authentication controller verifies the login details and passes them to the user account database. The user database stores all the accounts' information. The patient data controller modifies and filters the FHIR patient data to control the appropriate level and send it to the specific interface. The message access controller connects to the message database which contains the message history.

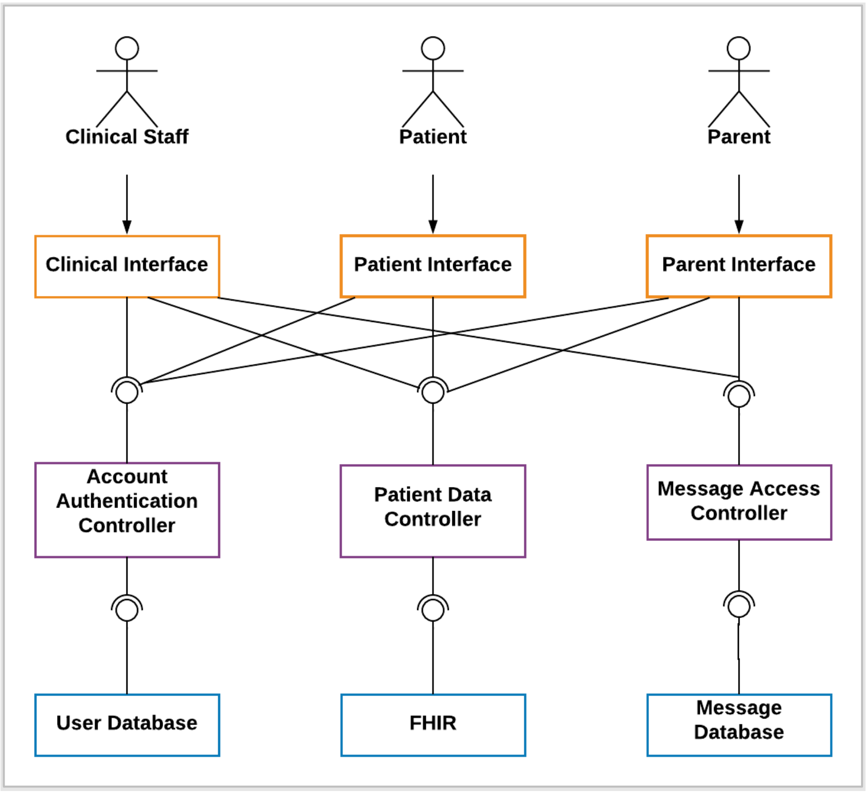

Figure 8: Functional View of GOSH's system

4.2 Data model layer

The app will make use of the resources defined in the FHIR standard (v3.0.1). It will also extend/wrap them where needed and add some separate data structures where appropriate. An overview of the most important entities could be seen in Figure 9.

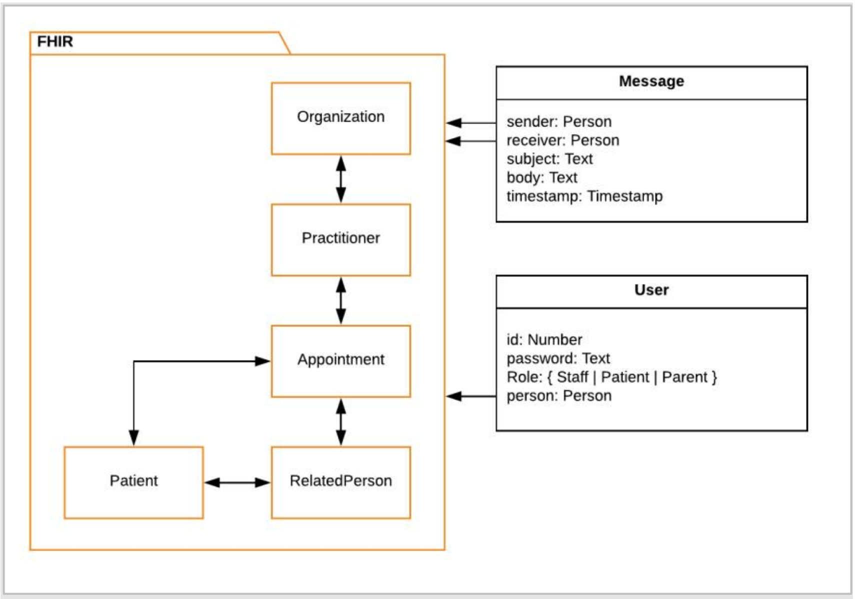

Figure 9: Overview of the core data model of GOSH's system

4.2.1 FHIR resources

- Person: Demographics and administrative information about a person independent of any specific health-related context. (Not in the diagram, however both Patient and Practitioner are subtypes of Person)
- Patient: Demographics and other administrative information about an individual or animal receiving healthcare.
- Practitioner: A person who is directly or indirectly involved in the provisioning of healthcare.
- RelatedPerson: Information about a person related to a patient. In our context, this will usually be a parent.
- Appointment: A booking of a healthcare event among patient(s), practitioner(s), and/or related person(s) for a specific date/time.
- Organization: A formally or informally recognized grouping of people or organizations.

All descriptions are taken from the official documentation [19]. For ease of reading, some resources that we will likely use are left out here, e.g. Location, Device, Task, multiple Diagnostics-themed resources and multiple Medication-themed resources.

4.2.2 Additional models

These data models will be created to store data not compatible with any of the available FHIR resources and are exclusively meant to be used in the context of the Speedy Recovery app.

- Message: Messages to be sent between staff and parents
- User: System users for authentication etc.

4.3 Presentation layer

The UI, as the main component for the users to interact with, is an essential part of any application. For Speedy Recovery what makes it so unique are the different view options based on the user. We are developing a top-level view for the administrative and clinical staff, a slightly simplified view for the parents of patients, and a basic view for the patients themselves. Each of these views will involve a calendar which describes the interactions a patient will have with the hospital, at different levels of detail depending on the stakeholder. The calendar will be formatted as a list displaying all of the interactions that a patient will have with the hospital over a certain day. Each of the stakeholders is then able to click on events to find out more information, depending on their access level. We hope that these three views will provide all of the necessary information that each of these stakeholders requires in a clear and concise way. The following sections describe in detail the makeup of these views.

4.3.1 Staff view

At GOSH, both administrative and clinical staff will need access to patient records and must be able to keep track of which procedures need to be performed on a patient. This information will all be provided through our app based on records stored on the FHIR infrastructure. The main functionality of this view will be comprised of a calendar with appointments that a given doctor has with a patient. By clicking on one of these appointments, the doctor, or any other relevant staff, will be able to see key medical information about the patient in a concise format. This will allow the doctor to quickly prepare for an appointment.

In addition to this, our app will store some personal information about the patient, such as their favourite football team, allowing the doctor to immediately interact with the young patient in an effective way. Only staff relevant to a given patient are able to see this information on our app. In addition to this basic functionality, the app also provides a messaging tool that allows parents to communicate with staff and vice versa. This will allow them to discuss with staff any questions they may have about their child's health.

The staff view of the app has, therefore, been created to make the life of staff at GOSH easier by making their interactions with patients as simple as possible.

4.3.2 Parent view

Staff at GOSH have access to a large amount of data about the patients they treat, sourced from FHIR resources. Not all of this information is relevant to parents, and information that they could find upsetting must be displayed in an appropriate way. Our app will, therefore, have a second view tailored to parents’ needs, which the patients themselves will not have access to.

This view again consists of a calendar containing the various appointments and treatments the patient will be receiving over the coming months. By clicking on one of these events, the parents will be shown important information that they need to know about the patient and their subsequent treatments. This information will be a subset of the staff’s view, containing only relevant information and any extra notes that staff may like to add. As mentioned above, the parent view will also provide a messaging feature allowing them to easily communicate with hospital staff.

The aim of this view is to provide parents with all of the information they need to know about their child’s treatment and for them to be able to communicate with the hospital whenever they need.

4.3.3 Patient view

The final view that our app provides is specifically tailored to the needs of a patient. Based on the calendar representation of the other views, this view provides the patient with information about which treatments they are scheduled to have, and which doctors they will be meeting in the future. One of GOSH's main goals is to ensure that patients feel as comfortable as possible. This is very hard, especially for the very young patients. The information within this view will, therefore, be displayed in an easy-to-read and fun way, including the addition of pictures and other animations. One example of this is that as part of the sign-up process our app aims to find out some interesting information about the doctors. This could include whether the doctor has a pet. This information will be shown to the patient when they click on the doctor that they are having an appointment with on a certain day. The idea behind this view is to provide the patient with some information that they might be interested in, allowing them to at least partly understand what is happening to them as they are treated at the hospital. All complex information will be left out, allowing the doctor to explain anything extra that the patient needs to know.

Through the use of this app we would hope that a patient is able to gain a basic idea of their treatment plans and the staff they are going to be interacting with in the future. We do not provide the messaging functionality for this view as we feel that it is the parent who performs most of the communications between the patient and hospital staff.

4.3.4 Prototype

This section presents a prototype of our application. The prototype shows a basic representation of what our app may look like, including its main features. We use it to test broad concepts of the application.

4.3.4.1 Staff View

When a user logs in successfully as a staff member, the user interface in Figure 10 will be shown. This example is the view of a fictional doctor. The staff member can only see the information about the relevant patients, the information of other patients cannot be seen. Users can click on the "Log Off" button to sign out and return to the login page. When the staff click a certain line in the calendar list, such as the second line in the example, the details of the patient will be shown as in Figure 11. The staff can also send messages to the parents of the patients. By clicking "Message" button, the messaging user interface will be shown as Figure 12. The user can click the "Back" button in the messaging page to return the main user interface.

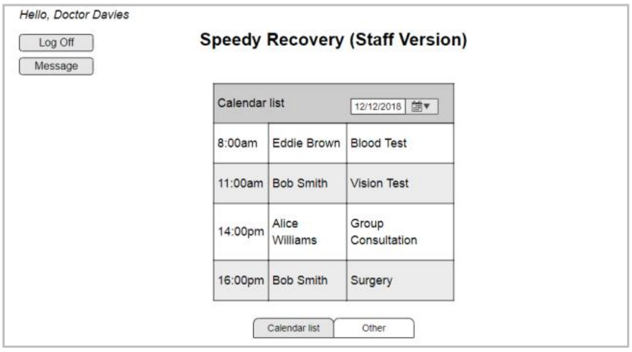

Figure 10: Main page after successfully login in Speedy Recovery (Staff Version)

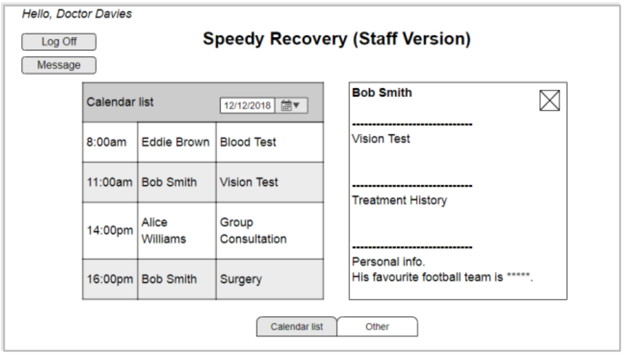

Figure 11: Page for showing details of a patient in Speedy Recovery (Staff Version)

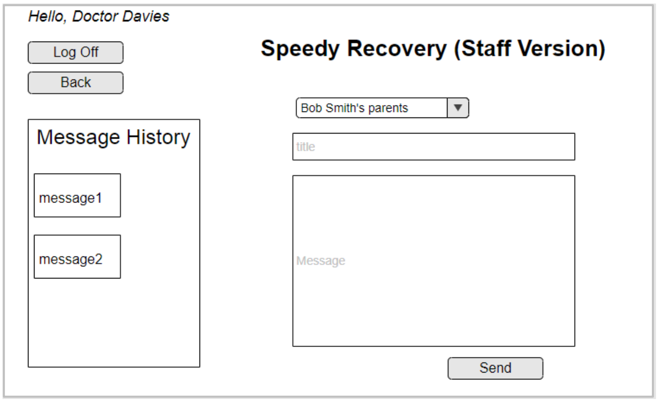

Figure 12: Page returned from messaging page in Speedy Recovery (Staff Version)

4.3.4.2 Parent View

When a user successfully logs in as a parent, the main user interface will be shown as in Figure 13. This example shows the view of the patient Bob Smith's parents. Parents can only see the information regarding their own children. By clicking a certain line in the calendar list, such as the first line in the example, the details will be shown as in Figure 14. Parents can use our messaging tool to communicate with staff. By clicking the "Message" button, the messaging user interface will be shown as in Figure 15.

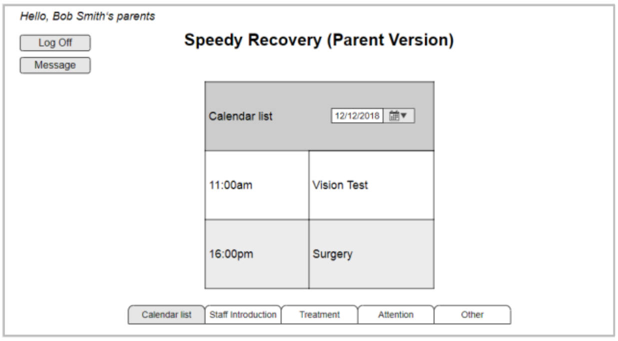

Figure 13: Main page after successfully login in Speedy Recovery (Parent version)

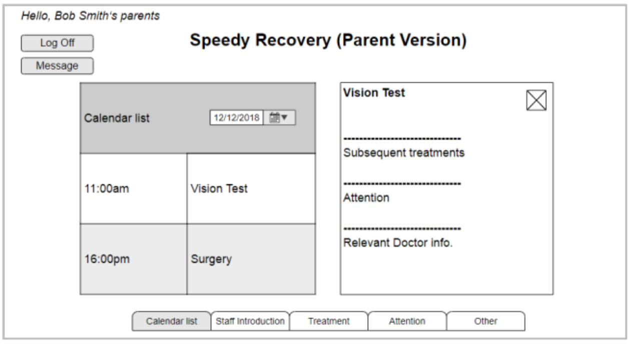

Figure 14: Page for showing details in Speedy Recovery (Parent Version)

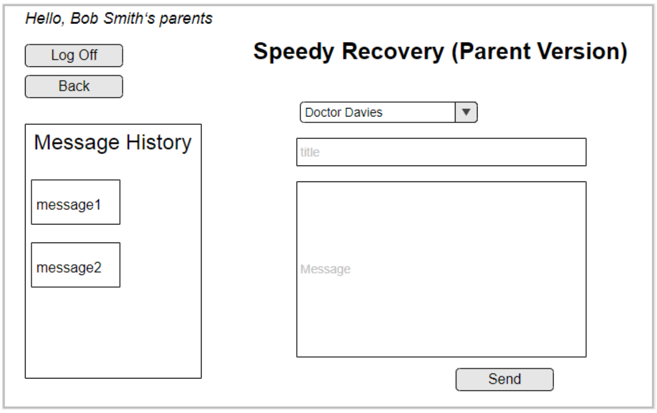

Figure 15: Page for messaging in Speedy Recovery (Parent Version)

4.3.4.3 Patient View

When a user successfully logs in as a patient, the main user interface will be shown as in Figure 16. This example is of the view of the patient Bob Smith. There is no messaging tool in the patient view. The information will be presented in an easy-to-read and fun way, including additional pictures and animations. By clicking a certain line in the calendar list, such as the first line in the example, the details will be shown as in Figure 17.

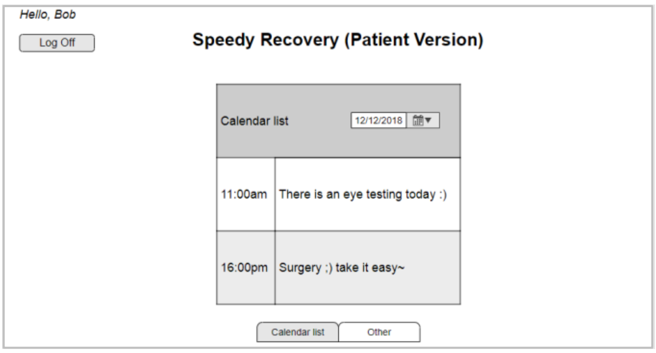

Figure 16: Main page after successfully login in Speedy Recovery (Patient version)

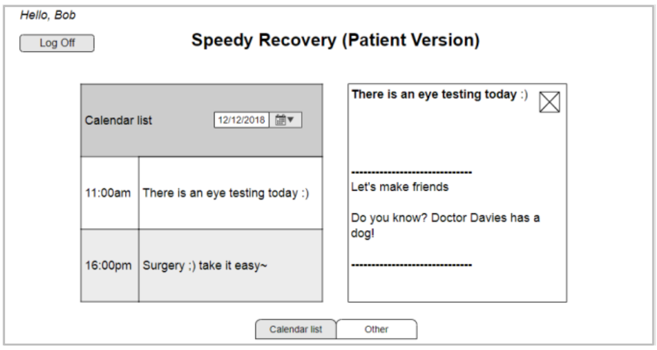

Figure 17: Page for showing details in Speedy Recovery (Patient Version)

4.4 Business logic layer

Based on the functional view diagram in Figure 8, the controller section is separated into three parts.

The Account Authentication Controller connects the user account database to the three user interfaces. The main function of this controller is to retrieve the login details from the user interface and try to match the details with entries in the database. If the data matches, it will bring the user to their homepage. This controller needs to make sure that the data is encrypted while be transferred, and that it always brings the user to the correct view.

The Patient Data Controller connects the FHIR infrastructure to the three user interfaces in order to provide patient data. However, for each interface the controller will provide a different level of patient detail. For example, for the patients' interface, as our patient is a child, we do not expect them to be able to understand the information contained in a patient record, so the controller will filter and adjust the details and before transmitting it to the view layer. The controller will perform similar edits to the data before displaying them to the other two stakeholders.

The Message Access Controller maintains the messaging functionality between users. The controller allows the message transfer between parents and staff members at the hospital, making sure that the message is encrypted during the transfer process. This is particularly important due to the highly sensitive data that is being discussed.

5. Implementation

5.1 Tools, languages and frameworks

5.1.1 Web framework

The front-end of the web-application will be developed using React, which is currently the most popular JavaScript library for this purpose, created by Facebook. React's most distinctive features are ability to build component-based architectures, one-way data flow, the Virtual DOM, the JSX language. [20]

We chose to use React for this project due to the large number of useful features it provides that we feel will help us as we carry out the development stage of our project. In particular, the fact that React provides an great testability and short learning curve, these will help us to develop all of our planned features for our app. React also has an extension called React Native, providing a framework for writing native mobile applications. This may be useful later on in our project, if we decide to create a native version of our app.

5.1.2 Development languages

As mentioned above, for this project we have decided to use React. This means that a majority of our development will be carried out using JavaScript. JavaScript is a very powerful language that will enable us to effectively create all the necessary functionality of our app.

TypeScript is an open source programming language developed and maintained by Microsoft. The language is a superset of the JavaScript and has been developed to be used for the creation of large applications[21]. We have decided to also use TypeScript for our project, as this allows us to write better structured, type-safe code. In fact, there is an open source project providing type definitions for FHIR resources that we can use.[22] This will make the development process that much easier from the start, allowing us to quickly connect up our app to the resources it relies on.

In addition to this in order to create the front-end UI's we will be using HTML and CSS.

In terms of our server-side languages, we have decided to use Node.js to build and maintain our server functionality. We chose to use Node.js for this part of the project due to the large amount of features it provides, and because we are using JavaScript in other areas of the project. This will make the overall codebase much more readable and will significantly reduce the learning cost in the implementation phase.

5.1.3 Version control

Version Control tracks all the changes during the development process. This allows developers to work on multiple features at the same time, while a production version of the app is open to the public. For the version control of this project we will use the industry standard Git, as it allows us to easily maintain different branches for each part of the project that team members will be working on. These branches can then be merged together as they are completed.

In order to manage our Version Control system, we have decided to use GitHub. GitHub provides an easy-to-use way to manage our project. Features such as being able to easily look back at old commits and how the project has changed, and how pull requests can be used to ensure that code review is done before branches are merged, make GitHub very useful for our project. On top of this, the fact that GitHub integrates with other tools such as those used for continuous integration, makes the entire development process run much more smoothly.

5.1.4 Continuous integration

In order to manage the overall progress of our app and to ensure that each build passes all of its test cases, we will be using continuous integration (CI). Multiple team members will be working on the codebase at any given moment, constantly pushing changes to the main repository. In order to avoid any potential breakages in the code and to ensure that the app is continuously running, CI will allow us to periodically build and test our code. This will ensure that our product is running 24/7. We have chosen Travis as our CI tool. This is due to its easy-to-use interface and its easy integration with GitHub. This allows us to easily setup an overall testing and build framework without too much maintenance. Also, if any breakages do occur in our code, the team will be immediately notified, and an appropriate fix can be made.

5.1.5 FHIR

As mentioned earlier, the main source of patient information for our app will be through the FHIR standard. The FHIR standard is made up of several different resources, each corresponding to an important area of the medical profession. FHIR resources can be displayed as XML or JSON data, and we have decided to use JSON for its better readability.

SMART is an open, standards-based technology platform that allows organizations to develop apps that run on different electronic health record systems.

SMART on FHIR is an open set of specifications built on top of FHIR to integrate apps with EHRs and other HIT systems. It specifies standards for scopes and permissions, sign-ins, and UI integration. As our app relies on using real patient data for its main functionality, for the testing and development phases we have to rely on synthetic patient data. SMART on FHIR provides us with a sandbox with this synthetic data, allowing us to easily test out our app's functionality on realistic scenarios. This will ensure that we make our app ready for its usage on real-world patient records.

5.1.6 Database

In order to provide the functionality that we set out to create as part of our app, we will need a database. This database will store all the users of our app, as well as all of the messaging data that our app creates as users communicate with each other. We have decided to use MySQL for this task, due to the powerful features it provides.

5.1.7 Deployment platform

In order to physically deploy our app, we have decided to use Heroku. Heroku provides an easy to use framework to deploy our app at different stages throughout the development process. It integrates well with GitHub and other tools, allowing deployment to occur as smoothly as possible. It will also monitor our app to ensure that it is running as expected, notifying us if there are any problems. Heroku supports many different languages including JavaScript, thereby making it the ideal tool for us to use to deploy our web-app.

5.2 Deployment and infrastructure

In this section, we outline the deployment and infrastructure on which our system will be built upon. Also, a deployment view will be used to understand the physical environment in which a system is intended to run, including any hardware components where software components are deployed. Deployment diagrams consist of nodes and their relationships.

5.2.1 Tooling and environment

As has already been described in section 5.1, we use several tools to create a dependable deployment workflow:

- GitHub
- Heroku
- Travis

Also, in order to access FHIR resources while still developing in development or integration environments, we will use a FHIR sandbox.

5.2.2 Deployment view

Figure 19 shows the deployment view of our web-application which consists of:

- **A database server (node)** which contains the user and messaging database required for storing login credentials and messages between parents and staff.
- **A FHIR server (node)** which will provide patients' records.
- **A back-end server (node)** that will host a list of backend services such as checking login credentials, tailoring patient data for three separate views and exchanging messages between parents and staff.
- **The devices** used by staff, parents and patients from where they can access our web-application.

When a user tries to login to the web-application, their login information will be checked against the backend server which will retrieve information from the login database to verify it. Patient information provided by FHIR will then be tailored to the user in the backend server, before being displayed to them. Moreover, all the messages exchanged between parents and admin staff will also be controlled by the back-end server and stored in the messaging database.

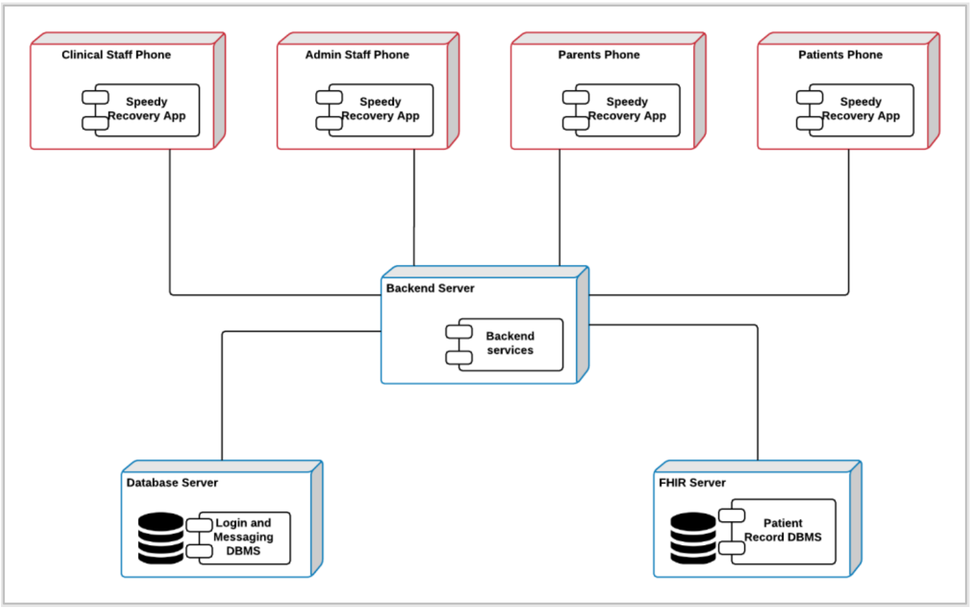

Figure : Deployment view of the web app

6. Term 2 Project

In term 2, we will begin the development stage of this project, creating the app together with its core functionality. We will structure the term based on five iterations, each concluding with a working prototype of our app. Each iteration will build upon our achievements of the previous phase, and together with user feedback, we will introduce more features as we go along. The timings specified for each iteration are based on our estimation of how long each of the tasks will take, however, some may take a shorter or longer amount of time than planned. We, therefore, ensured to allow ourselves plenty of time for each, bringing some freedom if any iteration overruns.

6.1 Iteration 1: Week 1

This iteration involves the initial setup of the various tools and frameworks that we will be using to construct our app. This includes setting up a git repository which every member of the team has access to, setting up Continuous Integration for any testing that we carry out, as well as ensuring that every team member is on the same page for the subsequent iterations. We have dedicated an entire week to this phase, in order to ensure that the entire team is properly prepared and ready to properly start developing. Once this iteration is complete, we will be ready to start creating our first prototype of the app.

6.2 Iteration 2: Week 2-3

In this iteration we will be creating the first working prototype of our app. This means we will create a basic web application that is able to take information directly from FHIR and display it to the user. There will also be some development of a simple front-end interface that allows this information to be displayed in an appropriate way. For this iteration, we will split up into sub teams, each focussing on different areas of the prototype. By the end of this iteration we will have a functioning web app which can take information from FHIR and display it to the user in a clear way. Whilst carrying out this phase, we will ensure to make our code easily adjustable and ready to be built on in the next iteration. Testing will also be carried out to ensure that all of this basic functionality is working as expected.

6.3 Iteration 3: Week 4-6

The next iteration builds on the prototype that we created in the second phase. We will first add some general improvements to the web app including the front-end design and how the data from FHIR is displayed to the user. This will also involve us starting to sort the data into three separate views to display depending on the stakeholder using the app. In addition to this, we will also start to add some additional functionality. We will add a basic form of the messaging functionality that will allow parents and hospital staff to communicate. As part of this, we will also implement most of the back-end functionality of the app including the database which will store users and their message information. This iteration will require a large amount of work for our team, therefore, we have given it a full three weeks to carry it out. This will allow us to create a second complete prototype which now contains all of the main functionality that we aim to deploy as part of our app.

6.4 Iteration 4: Week 7-8

This iteration involves us adding another few pieces of important functionality that are fundamental to our app. This includes adding authentication to ensure that only authorised users are able to see a patient's medical records. This requires the setting up of a sign-up process, as well as updating the back-end and database accordingly. We will also have to update the front-end to include this authentication procedure. In addition to implementing these new features, we will also continue refining the already implemented functionality in both the front and back ends, to ensure that the app is working as efficiently as possible. In addition to this, we will continue to test the system to ensure that it is able to function correctly in any given scenario. This will provide us with a third prototype of our app containing all of the integral functionality that we intend to introduce next term.

6.5 Iteration 5: Week 9-10

This is our final iteration, where we intend to finish our app according to the various requirements laid out in this document. This iteration will mainly involve applying various tweaks to the front-end, as well fixing any outstanding bugs in the system. We will also use this iteration as a chance to gain some detailed user feedback from users, to see what they think about our app. This will allow us to either make some important adjustments or add plans for future work based on this feedback. By this stage we will have completed our project brief and created an app that makes the communication between hospital staff, parents and patients as smooth as possible.

7. Future work

The app that we are creating as part of this project brings with it a large number of benefits for GOSH and its stakeholders. There are, however, several areas that could be worked on further, in order to allow our app to provide even more useful features for users. This section discusses several areas that we think should be worked on if this project was to be taken further.

In order to create and test our app's functionality, throughout this project we will be working inside a sandbox with synthetic patient data. This data is made up of synthetic information about a patient together with their medical records and is not based on any real patients. This data is very useful for us as we develop our app and test it to ensure that it is meeting our requirements. However, in order to properly deploy the system, we will need to access real patient data. Therefore, the next step for our app once it has been developed, will be to deploy it on GOSH's internal systems and receive FHIR patient data from their EHR system. GOSH has recently started using an EHR system by EPIC, which might require us to adapt our API calls, as SMART and EPIC behave differently in some instances. This will allow the app to access FHIR resources from real patients, giving GOSH's stakeholders the ability to use our app in the real world. Our app will then be able to be used both on stakeholder's own devices, but also, especially for patients, on iPads and other devices that GOSH provides. Once this has been set up and our app has established itself at GOSH, our long-term plan would be to then deploy our system in other hospitals around the world. This will help us make Speedy Recovery a leading application for hospital communication technology.

Currently, our plan is to create an app with three different views, one for staff, one for parents and one for patients. The patient view of the app will be focused on children aged 7 to 10. Our app may not, however, be necessarily appropriate for older or younger children. Therefore, another potential future extension to our project would be to add several other views to cater for other age brackets. For example, a new slightly more detailed and informative version could be developed for teenagers. By adding these extra views, we will be able to cater for different types of patients in hospitals, bringing them all of the benefits that our app provides. This could also be extended to different types of staff in the hospital, who also may require different types of information from our app.

FHIR is a constantly evolving standard. The current version that we are using to develop our app exists only as a draft standard. Next year, however, a production version of FHIR is due to be released. This standard will then start to be rolled out at hospitals using real patient data. This release will allow us to, in the future, deploy our app on hospital servers in order for them to receive real patient data from FHIR. This will allow hospital stakeholders to physically benefit from the features that our app provides. As further versions of FHIR are released, we will also have to update our app accordingly.

A further extension that we would like to implement in the future is the creation of a native app. At the moment our application will be a web-app that is able to be accessed on any device. However, often the most aesthetically pleasing and easy-to-use features can only be created using a native app. As our app is mainly going to be used on mobile devices, the creation of a native app would be a useful addition to the web UI. Once this version is created, other features such as the ability to integrate the calendar of our app into a device's calendar would also be a useful addition. Other features that we would also look at implementing would be a messaging feature for patients in order for them to stay in contact with each other while they are at the hospital, and the ability to add hospital social events to a patients' calendar that they may be interested in attending. These extra features would make our app even more useful for patients.

Finally, one of the most important requirements that our app must achieve, is a high level of security. In particular, in terms of the protection of a patients' private medical data. Although we will have already put in place several security protocols and procedures, more could still be done. Therefore, one additional feature that we would like to add in the future would be 2-factor authentication. This will add extra protection helping to ensure that only a valid user can login to a given account and only they can see any sensitive information. The addition of this and other further security measures will help to ensure that all personal information about patients are kept secure.

8. Glossary and acronyms

De-identified Patient Health Information	PHI from which all data elements that could allow the data to be traced back to the patient have been removed. [23]
DRIVE	Digital Research, Informatics and Virtual Environments. A Unit of GOSH, launched in 2018 and focused on research and evaluation of new technology and data analysis for healthcare. It represents a partnership between GOSH, UCL and multiple leading technology companies. [1]
EHR	Electronic Health Record. A stakeholder wide electronic record of a patient's complete health situation. [23]
FHIR	Fast Healthcare Interoperability Resources. A modern healthcare-related standards framework created by HL7. FHIR combines features of previous, less elaborate standards (HL7 v2, HL7 v3, CDA) and builds on web standards for communication. [24]
GOSH	Great Ormond Street Hospital
HIT	Health Information Technology. The set of tools needed to facilitate electronic documentation and management of healthcare delivery. [23]
HL7	Health Level 7. A not-for-profit global organization to establish standards for interoperability. [23]
Interoperability	The ability of diverse information systems to seamlessly share data and coordinate on tasks involving multiple systems. [23]
MVC	Model-View-Controller
NHS	National Health Service.
OAuth	Open Authorization is an open standard for token-based authentication and authorization on the Internet. It allows an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password. OAuth acts as an intermediary on behalf of the end user, providing the service with an access token that authorizes specific account information to be shared. [25]
Patient	While a patient is generally just a "person under health care"[26], in the context of this report it is used to describe a patient in childhood, as all patients of GOSH implicitly are.

Provider	Health professionals including physicians, nurse practitioners, physicians’ assistants that are engaged in direct patient care. [23]
SMART on FHIR	Substitutable Medical Apps, Reusable Technology. An open set of specifications built on top of FHIR to integrate apps with EHRs and other HIT systems. It specifies standards for scopes and permissions, sign-ins, and UI integration. [14]
Synthetic Health Data	Facsimile clinical data created by a software system to realistically resemble actual patient data. [23]
UCL	University College London.
UI	User Interface.

9. References

[1] GOSH looks to DRIVE healthcare technology forward. Digital Health 2018. https://www.digitalhealth.net/2018/10/gosh-drive-healthcare-technology/ (accessed November 27, 2018).

[2] Olschesky M. Introduction to FHIR | The Datica Academy 2018. https://datica.com/academy/introduction-to-fhir/ (accessed November 27, 2018).

[3] Stand-up keynote Janet Campbell | Schedule 2018. FHIR DevDays - Europe n.d. https://www.fhirdevdays.com/amsterdam/schedule-2018/ (accessed December 16, 2018).

[4] Great Ormond Street Hospital for Children NHS Foundation Trust. Who we are n.d. https://www.gosh.nhs.uk/about-us/who-we-are (accessed December 8, 2018).

[5] Great Ormond Street Hospital for Children NHS Foundation Trust. Our vision as an independent NHS Foundation Trust n.d. https://www.gosh.nhs.uk/about-us/foundation-trust/what-it-means-be-nhs-foundation-trust/our-vision-independent-nhs-foundation-trust (accessed December 7, 2018).

[6] Great Ormond Street Hospital for Children NHS Foundation Trust. Our place in the wider world n.d. https://www.gosh.nhs.uk/about-us/our-priorities/quality-improvement/our-progress/our-place-wider-world (accessed December 7, 2018).

[7] Great Ormond Street Hospital for Children NHS Foundation Trust. New unit opening at Great Ormond Street Hospital set to revolutionise how technology is used in hospitals n.d. https://www.gosh.nhs.uk/news/latest-press-releases/new-unit-opening-great-ormond-street-hospital-set-revolutionise-how-technology-used-hospitals (accessed November 27, 2018).

[8] Boston Children's Hospital. Wikipedia 2018.

[9] Medical Apps | Boston Children's Hospital n.d. http://www.childrenshospital.org/clinician-resources/for-boston-childrens-staff/mobility-resources/medical-apps (accessed December 16, 2018).

[10] New app for children unable to speak n.d. https://www.evelinalondon.nhs.uk/about-us/news-events/2016-news/20160907-new-app-for-children-unable-to-speak.aspx (accessed December 16, 2018).

[11] 30 Most Technologically Advanced Children's Hospitals in the World. Top Master's in Healthcare Administration n.d. https://www.topmastersinhealthcare.com/30-most-technologically-advanced-childrens-hospitals-in-the-world/ (accessed December 16, 2018).

[12] Lintern S. Birmingham Children's Hospital hails feedback app. Health Service Journal n.d. https://www.hsj.co.uk/birmingham-childrens-hospital-nhs-foundation-trust/birmingham-childrens-hospital-hails-feedback-app/5063883.article (accessed December 16, 2018).

[13] SMART. What Is SMART? SMART Health IT 2012. https://smarthealthit.org/an-app-platform-for-healthcare/about/ (accessed December 7, 2018).

[14] Mandel JC, Kreda DA, Mandl KD, Kohane IS, Ramoni RB. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. J Am Med Inform Assoc 2016;23:899–908. doi:10.1093/jamia/ocv189.

[15] OpenID. OpenID Connect FAQ and Q&As 2018. https://openid.net/connect/faq/ (accessed December 10, 2018).

[16] MozillaWiki. Security/Guidelines/OpenID connect 2017. https://wiki.mozilla.org/Security/Guidelines/OpenID_connect#Implement_authentication_with_OpenID_Connect_.28OIDC.29_securely_in_my_web_applications_.28RP.29 (accessed December 10, 2018).

[17] The Risk Management Process in Project Management. ProjectManagerCom 2017. https://www.projectmanager.com/blog/risk-management-process-steps (accessed December 4, 2018).

[18] WhatsApp Security. WhatsApp.com n.d. https://www.whatsapp.com/security/ (accessed December 9, 2018).

[19] HL7.org. Resourcelist - FHIR v3.0.1 2017. https://www.hl7.org/fhir/resourcelist.html (accessed December 4, 2018).

[20] What Is ReactJS and Why Should We Use It? n.d. https://www.c-sharpcorner.com/article/what-and-why-reactjs/ (accessed December 22, 2018).

[21] What Is TypeScript? Program With Erik 2015. https://www.programwitherik.com/what-is-typescript/ (accessed December 22, 2018).

[22] The repository for high quality TypeScript type definitions.: DefinitelyTyped/DefinitelyTyped. DefinitelyTyped; 2018.

[23] Braunstein ML. Glossary of Health and Health IT Terms and Acronyms. Health Informatics on FHIR 2015. https://www.coursera.org/learn/fhir/resources/vgvHg (accessed November 27, 2018).

[24] HL7. Summary - FHIR v3.0.1 2017. https://www.hl7.org/fhir/summary.html (accessed November 27, 2018).

[25] Mann S. What is OAuth? - Definition from WhatIs.com. SearchMicroservices 2012. https://searchmicroservices.techtarget.com/definition/OAuth (accessed December 7, 2018).

[26] Definition of Patient. MedicineNet 2016. https://www.medicinenet.com/script/main/art.asp?articlekey=39154 (accessed November 27, 2018).

10. List of figures