
Smart Hospital

By

AUTHOR'S NAME



Department of Engineering Mathematics
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

APRIL 2013

Word count: ten thousand and four

Abstract

Here write the abstract

Dedication and acknowledgements

Here goes the dedication.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: DATE:

Table of Contents

	Page
List of Tables	vi
List of Figures	vii
1 Introduction	1
1.1 Motivation	1
1.2 Client Brief	1
1.3 Aims and Objectives	2
1.4 Challenges	3
2 Background	4
2.1 Relative Work	4
2.2 Existing Solutions	4
2.2.1 Teladoc Health	4
2.2.2 LINE Hospital Services	5
2.3 Requirements	6
3 Design and Implementations	7
3.1 Methodology	7
3.1.1 Work Flow	7
3.1.2 Process	7
3.1.3 Front-end Tools	7
3.1.4 Back-end Tools	10
3.1.5 Testing Tools	10
3.2 Front-end Design and Implementations	10
3.2.1 Early Stage-Virtual Hospital Africa(VHA)	10
3.2.2 Smart Hospital	10
3.3 Back-end Design and Implementations	13
4 Evaluation and Testing	14

TABLE OF CONTENTS

5 Conclusion	15
6 Reference	16
A Appendix A	17
Bibliography	18

List of Tables

Table	Page
2.1 Smart Hospital User Stories	6

List of Figures

Figure	Page
2.1 Homepage of Teladoc Health.	5
2.2 Service menu of Teladoc Health.	5
2.3 Screenshot of LINE services from Far Eastern Memorial Hospital, showing options like booking, announcements, and personal info links [1].	5

Chapter 1

Introduction

1.1 Motivation

The program will build a virtual hospital website called Smart Hospital, which aims to provide better access to medical care for residents in remote or medically underserved areas in Africa. Due to geographical limitations, transportation difficulties, and insufficient medical facilities, many residents often cannot receive good medical care. Through this virtual hospital website, residents in these areas will have the opportunity to consult with specialists remotely and receive more medical assistance and health support.

This project was initially part of a collaboration with the Africa Virtual Hospital initiative, which is dedicated to addressing healthcare disparities in remote and underserved communities across Africa. The initiative aims to connect these underserved regions with doctors and experts from around the world through its virtual hospital platform. However, despite the termination of the collaboration due to certain factors, our team chose to independently continue this project. In the context of rising global healthcare pressures, including an aging population, increasing rates of chronic diseases, and uneven distribution of medical resources, we recognize that establishing a digital healthcare website to provide telemedicine remains highly meaningful, particularly for regions with insufficient medical resources or limited access to healthcare.

1.2 Client Brief

This project was initially undertaken in collaboration with Africa Virtual Hospital. As the client, Africa Virtual Hospital proposed the development of a virtual healthcare software platform comprising the following five main components: Doctor Application, WhatsApp Chatbots for Patients and Service Providers, Caregiver Application, Nurse Application, and Electronic Health Record. Within this framework, our team's primary focus was on the doctor application, which was expected to run on multiple devices, including smartphones, tablets, and laptops. The Doctor Application component was expected to include the following elements:

- Registration
- SSO/Welcome
- Home page (Appointments and Reviews)
- Calendar (Availability)
- Notifications
- Patient Profile
- Connect (Video and Voice to Patient WhatsApp)
- Treatment Plan and Orders
- Patient Information Feed
- My Supporting Staff
- My Profile

During the collaboration period, we developed the various webpages and features of the doctor application based on the design and technical resources provided by the client. However, with the termination of the collaboration, our team also lost the original technical support. And then our new client is our University, but given the constraints on development time and resources, our team adjusted the scope based on existing resources and the original proposal, re-planned the website architecture and front-end design, and prioritized retaining core functional modules such as the doctor-patient interaction interface and basic medical information viewing, to ensure the project's feasibility under current conditions.

1.3 Aims and Objectives

The aim of this project is to design and develop an easy-to-use, cross-device virtual hospital platform to promote telemedicine and doctor-patient communication, thereby improving medical accessibility in remote areas.

Based on the aim, we have established the following specific implementation items:
Develop an intuitive user interface and create a web-based system that supports physician operations. Implement core system functions, including appointment scheduling, patient record access, and basic remote consultation workflows. Establish an integrated patient data interface to help physicians understand the patient's overall condition, thereby improving diagnostic efficiency and decision-making quality.

1.4 Challenges

This project faced three main challenges: unexpected termination of cooperation, limited development time, and difficulties in obtaining back-end data resources.

First, the project began in June and was expected to end in early September. Under the original plan, we were to collaborate with Africa Virtual Hospital to complete the development of the Doctor Application. Although most team members lacked website development experience, the development work proceeded smoothly under the guidance of Africa Virtual Hospital in the early stages.

However, the partnership was terminated prematurely in mid-July, resulting in the team losing access to technical support and shared data, and significantly compressing the remaining development timeline. With only about one and a half months left for development, the team had to independently complete tasks such as system design and construction, leading to an extremely tight schedule.

Additionally, the team had originally planned to utilize existing data from Africa Virtual Hospital, such as patient basic information and medical records. However, due to the termination of the collaboration, our backend development team faced difficulties in accessing patient data and medical records, which are highly sensitive, thereby imposing some limitations on website functionality verification and data simulation.

Despite these challenges, our team adjusted the project scope, reallocated resources, and prioritized the implementation of core functions to ensure the platform's feasibility.

Chapter 2

Background

2.1 Relative Work

In many hospitals, patients need to wait for a long time just to have a short consultation with a doctor. This can be exhausting, especially for people who are already sick. In some places like the countryside, the problem is worse because there are not enough doctors or clinics [2]. Some people need to travel very far to get help, which is hard for old people or people with long-term illness.

During COVID-19, many doctors started using phone or video to talk to patients. In the UK, it went from 15% to 48% in just a few weeks [3]. It helped people talk to doctors from home and also saved time.

Our Smart Hospital system also uses phone and video consultations. We also have Health QA, where people can ask simple health questions and get advice more easily. These help people who live far away or cannot go outside, and they also make the waiting time in hospitals shorter.

Our system supports the United Nations' SDG Goal 3, which is about helping people stay healthy and live well [4]. Among all the targets, Goal 3.8 and Goal 3.4 are especially important to our project. Goal 3.8 is about giving basic and affordable health services to everyone. Goal 3.4 is about helping people with long-term sickness, like heart disease or mental problems. Our system lets them contact doctors or get advice online more easily, so they can get help earlier before the problem becomes worse.

2.2 Existing Solutions

2.2.1 Teladoc Health

Teladoc Health is a popular online medical platform[?] that offers services like video calls with doctors and support for mental health. It is used in many countries and is helpful for people

who find it hard to go to a hospital. But one problem is that it's not fully connected to real hospitals. This means doctors on Teladoc often can't see a patient's full medical records from other clinics or hospitals. As a result, some information might be missing, which can make it harder to give the right diagnosis or follow-up care.

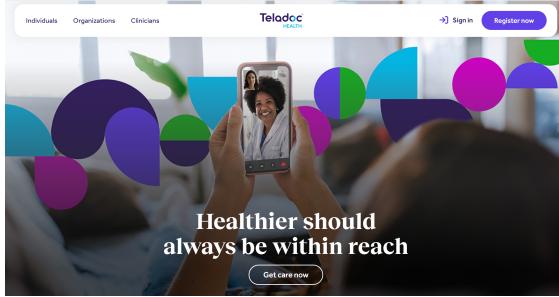


Figure 2.1: Homepage of Teladoc Health.

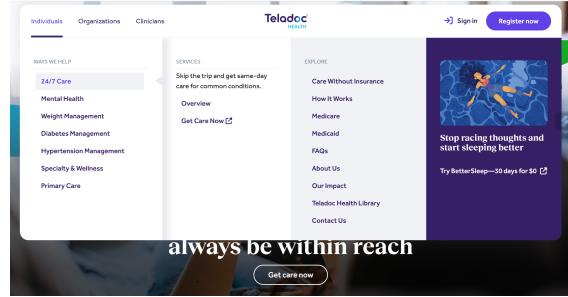


Figure 2.2: Service menu of Teladoc Health.

2.2.2 LINE Hospital Services

In Taiwan, many hospitals have their own official LINE accounts to provide online services. Patients can add the hospital's LINE to access booking and other basic functions. It is very easy to use, since most people in Taiwan already use LINE in their daily life. This makes it a common way for hospitals to offer simple medical services without asking people to download another app.



Figure 2.3: Screenshot of LINE services from Far Eastern Memorial Hospital, showing options like booking, announcements, and personal info links [1].

These features are quite easy to use. However, each hospital has its own design and menu. Some functions even open in separate websites. There is no unified system, so it can be confusing for some patients, especially older people or those not familiar with technology.

2.3 Requirements

The system is designed as a web-based platform for doctors, nurses, and patients to facilitate efficient clinical workflows and improve access to healthcare services. The main objective is to allow healthcare staff to manage patient information, record essential clinical data, and provide consultations remotely, thereby reducing unnecessary hospital visits. Patients are also supported with features to access their health records and receive guidance without needing to attend the hospital in person.

To ensure a functional and effective experience for hospital staff and patients, the system was designed to support a core set of features. These include secure user login and authentication for doctors, nurses, and patients; the ability to record patient vitals and view historical data trends; and an interface for remote consultation to minimise unnecessary hospital visits.

Drawing on the team's previous experience with similar healthcare platforms, including the Virtual Hospital Africa system, we referenced their approach as a conceptual foundation while adapting the design to our own project scope and requirements.

The system is initially populated with pre-existing mock patient data to facilitate testing and demonstration. In addition to these accounts, the platform supports new patient registration, enabling patients to create their own accounts and access the same core features, including secure login, health query submission, and medical record downloads.

Non-functional requirements include secure authentication with role-based access control, ensuring that only authorised users can access or modify clinical records. The system is designed for stable operation with fast response times to support real-time clinical workflows. In addition, the user interface is kept clear and intuitive to reduce training time for hospital staff. These system capabilities are closely aligned with the user stories presented below.

As a...	I want...	So that...	Technical ability
Patient	To ask health-related questions online	I can get medical guidance without visiting the hospital	2–3
Patient	To download my complete medical record	I can share it with a pharmacy or another healthcare provider	2–3
Nurse	To record patient vitals and update profile information	I can ensure accurate data is available for diagnosis	2–4
Doctor	To review patient history and vitals trends	I can make informed clinical decisions	3–5
Doctor	To add clinical notes and prescriptions	I can provide clear treatment guidance for the patient	3–5

Table 2.1: Smart Hospital User Stories

Chapter 3

Design and Implementations

Begins a chapter. Example: When the beloved cellist (Christopher Walken - outstanding) of a world-renowned string quartet receives a life-changing diagnosis, the group's future suddenly hangs in the balance: suppressed emotions, competing egos and uncontrollable passions threaten to derail years of friendship and collaboration. Featuring a brilliant ensemble cast (including Philip Seymour Hoffman, Catherine Keener and Mark Ivanir as the three other quartet members), it is a fascinating look into the world of working musicians, and an elegant homage to chamber music and the cultural world of New York. The music, of course, is ravishing (the score is the work of regular David Lynch collaborator Angelo Badalamenti): A Late Quartet hits all the right notes.

3.1 Methodology

3.1.1 Work Flow

3.1.2 Process

3.1.3 Front-end Tools

Figma

In our project, our goal is to design a website for medical use. As a team, we all agree that a high-quality website should include several essential factors. Among them, user interface (UI) and user experience (UX) are the most crucial. A well-designed website should maintain a consistent and visually appealing style across all pages, while also being intuitive to use and free from confusing workflows. Another key consideration is the design process itself. Since we are working as team, it is important to choose a tool and workflow that support efficient collaboration. First, team members should be allowed to work independently or collaboratively while still ensuring consistency across design. Second, we need a real-time collaborative environment to prevent

any individual's progress from being delayed by others. Therefore, Figma is our primary choice for designing the prototype of our website.

Figma is cloud-based design and prototyping tool that enables real-time collaboration among team members. It allows us to efficiently create UI, UX, wireframes, mockups and interactive prototype. These features allow our team to collaborate through the cloud. Meanwhile, we can work independently on individual tasks without interfering with one another's progress. From a broader perspective, Figma enable us to review each other's work instantly, ensuring our overall workflow and visual consistency remain coherent across all pages. One of the most practical benefits is that Figma can automatically generate CSS code for layout, fonts, and other design properties. This greatly enhances our efficiency when translating designs into front-end code. For those reasons, we choose Figma as the primary tool for prototyping our website.

HTML

We aim to provide a clean, accessible and responsive interface for medical users. To achieve this goal, we require tools that help us build structured, semantic layouts while maintaining visual consistency across devices and screen sizes. When evaluating which front-end tools were most suitable for our design, we considered several combinations, including HTML with Bootstrap, HTML with TailwindCSS, and SCSS-based modular design systems. According to the specific characteristics of our project and our situation. We all agree that using HyperText Markup Language(HTML) and Cascading Style Sheets(CSS) was the most appropriate choice for our needs after discussion. In the following section, we provide a brief introduction to these two technologies and explain how they contributed to the development of our prototype.

HTML is a necessary and standard core language that can be interpreted by any modern web browsers. In web development, HTML is used to arrange the content of the website and provide basic structural syntax for page design. As a fundamental tool, HTML has a relatively simple and easy-to-learn syntax, allowing developers to create websites with clear semantic structures that enhances both accessibility and user-friendliness. HTML includes ranges of labels, such as `<head>`, `<p>`, and `<div>`, which are used to organise the layout of a page. These elements support essential functions like hyperlinks, image embedding, list creation and tables formatting. With just basic syntax, developers can easily achieve a certain level of responsive functionality. Although various tools exist to assist in web development—such as page builders or frameworks—all of them ultimately compile down to HTML. As a result, HTML is irreplaceable in the front-end development process. Due to the situation our team has encountered, we all agree that the cost of learning alternative tools would exceed their benefits. Therefore, we decide to directly use plain HTML as our primary front-end tool. However, as mentioned above, HTML only provides basic structure and limiting styling capabilities. Managing layout and appearance directly within HTML would lead to inefficiencies and inconsistency. To address this issue, we employed CSS to handle visual design and layout, which will be discussed in next section.

CSS

One crucial issue in web development is how to manage code efficiently and make it maintainable. A widely adopted and effective solution is to separate the content, functionality and the style into distinct layers. While tools such as SCSS, Tailwind CSS and Bootstrap provide advanced styling features, we ultimately chose to use standard CSS for this project. This decision was based on the simplicity and clarity that CSS provides—especially valuable for a small team working on a focused prototype. CSS allows us to directly translate our Figma designs into code without introducing additional syntax or dependencies. This made it easier for all team members to understand and contribute to the styling process. Besides, one of the key advantages of CSS is its clear separation of content and presentation, which improves maintainability and enables us to update visual styles without modifying the fundamental structure. CSS also plays a crucial role in transforming our HTML structure into a clean, professional, and user-friendly interface. Furthermore, CSS enhances our collaboration by allowing us to define a shared layout and style system. Each member can easily apply consistent styles across pages, significantly reducing time spent on detail adjustments, and improving the efficiency of our workflow.

Spring Boot and Thymeleaf

Before selecting our technology stack for the Smart Hospital web application, we weighed a number of backend and frontend framework possibilities with consideration of equating development speed, maintainability, and integration efficiency. The top contenders included Spring Boot with Thymeleaf, Node.js with Express and React, and Django with HTML templates. Table 1(@TODO:haven't applied yet) summarizes the strengths and weaknesses of each stack based on thoroughly documented pros and cons according to credible sources.

We weighed these options and opted to go with Spring Boot in the backend and Thymeleaf in the frontend. This Java-focused stack provided us with a consistent and streamlined development environment, which was particularly crucial for our small team of individuals working on tight deadlines. Because both frameworks are based on the Java ecosystem, we were able to build full-stack functionality in one integrated project. This reduced the complexity that would have otherwise been caused by having two separate frontend and backend codebases and improved coordination among team members [3].(@TODO:haven't applied yet)

Spring Boot is a well-used enterprise-level framework that streamlines Java web application development. It provides an embedded Tomcat server, auto-configuration, and dependency management, allowing us to concentrate on core functionality instead of boilerplate setup [1]. Its innate RESTful API support enabled us to plan a scalable, modular backend architecture, with Controller classes as the basis for API endpoints, a core feature for connecting the client interface to backend services.

At the frontend, Thymeleaf was selected as a server-side templating engine that has native support for Spring Boot's MVC framework. Its dynamic binding of backend data in HTML templates ensured tight coupling of presentation logic with backend processes. This eliminated the need for extra JavaScript frameworks like React or Angular, reducing the learning curve and allowing the team to progress faster. Thymeleaf's human-readable syntax also made collaboration easier and the codebase more maintainable [2](@TODO:haven't applied yet).

Alternatives such as Node.js with Express and React were also considered, which would have offered highly interactive UIs and would have leveraged JavaScript's extensive ecosystem, but would also have meant two codebases to manage and deployment complexity [3]. Django with HTML templates, knowingly able to achieve rapid development and secure by design [1](@TODO:haven't applied yet), was also a consideration but would have required the team to learn Python, an undesirable learning curve considering the time constraints we had.

Finally, our academic supervisor recommended picking up Spring Boot with Thymeleaf for its combination of performance, integration ease, and suitability for a Java-experienced team. This also simplified deployment, as both backend and frontend could be deployed together in a single application. Moving further to the backend implementation phase, Spring Boot Controller classes will accept incoming HTTP requests and will form the basis of our API layer, which, later on, will be tested by tools discussed in the Backend Tools section.

3.1.4 Back-end Tools

3.1.5 Testing Tools

3.2 Front-end Design and Implementations

3.2.1 Early Stage-Virtual Hospital Africa(VHA)

3.2.2 Smart Hospital

Design Process

From Concept to User Needs: Ideation and Story Mappingigma

Bridging Stories and Systems: Workflow Design

Visualizing Functionality: Prototyping with Figma (@TODO:need organised)

After we completed the workflow diagram, we used it as the foundation to divide tasks among team members. Each member is responsible for designing specific pages for different

parts of the website, and we continually review each other's work using the comment feature to ensure real-time feedback and update.**(TODO:need picture)** To evaluate whether the designs meet our requirements, we test various user scenarios using Figma's prototyping feature to link different steps and assess the overall flow. For example, we add a "view" button on the patient profile page to allow doctors to access records of past visits, which helps them make more informed diagnoses.**(TODO:need picture)**

Color selection plays a vital role in website design. In our project, we choose a range of blue tones as the primary color scheme, as blue often represents inclusiveness and trust. To make the overall atmosphere feel more approachable and friendly, we incorporate some earthy tones into the blue. Meanwhile, white and gray is used for elements like buttons and text areas to create a clean and orderly appearance. Through these color combinations, we aim to convey a sense of stability and warmth to our users. The overall visual design reflects the spirit of our platform—to serve as a strong and trustworthy support for users of all ages, genders, physical conditions and cultural background.**(TODO:need picture to present the style)**

Next, we will introduce our layout design. The primary goal of our layout is to ensure intuitive user interaction. For example when doctors are diagnosing a patient, they typically start by reviewing the patient's profile to gain a comprehensive understanding of the patient's condition. This information is displayed on the "Patient Profile" page.**(TODO:need picture)** At the same time, doctors may need access to the patient's visits history, so we design a clear and concise list that displays the date of each visit, a brief summary of the diagnosis, and the attending physician.**(TODO:need picture)** If more detailed information is needed, doctors can click the "view" button to open a pop-up window showing the full report.**(TODO:need picture)** Another example is during the diagnosis process—doctors may want to take temporary notes before writing the final prescription or summary. To support this, we included a text area on the "Clinical Notes" page where doctors can write their notes down.**(TODO:need picture)** These features are all designed to enhance the experience for medical professionals interacting with our platform.

Front-end Implementation

(@TODO:need organised)

In the front-end implementation section, we illustrate the process of how we created, organised, and tested our front-end code. We divide this process into 5 stages. First, we describe how we structured and developed the codebase. Then, we explain how we integrate shared layouts using Thymeleaf to improve consistency and maintainability across pages. Next, we explain how different pages are connected to form a smooth user flow. We also discuss the steps we took to test each function to ensure the website behaves as intended. Finally, we reflect on the main challenges we encountered during the front-end development and how we addressed them. These topics are covered in following subsections: Code Structure and Component Organisation,

Integrating Shared Layouts with Thymeleaf, Page Navigation and User Flow, User Flow and Functional Testing, and Challenges Encountered During Frontend Development, respectively.

Code Structure and Component Organisation

In this section, we describe how we constructed the front-end codebase. As mentioned earlier, each member was responsible for designing different parts of the website in Figma. We continued the pattern during implementation—each member developed their assigned parts using HTML and CSS, focusing on accurately reproducing the layout and visual style from our Figma design.

At this stage, our goal is to replicate the layout and visual style of our Figma design. Most interactive elements, such as buttons, pop-up windows, and form submissions, have not yet been functionalized. We have also not yet integrated a shared layout system using Thymeleaf. These shared elements are currently duplicated across individual pages,(**TODO:need picture**) and we plan to organize them in the next development stage to improve maintainability and consistency.

Integrating Shared Layouts with Thymeleaf

After developing all the pages, we noticed that many structural elements—such as headers, sidebars, and forms—were repeated across different pages. To reduce redundancy and improve maintainability, we started integrating a shared layout system using Thymeleaf.

We created a base template that includes the common layout structure and used Thymeleaf’s th:replace and th:fragment features to inject shared components into each page.(**TODO:need picture**) This allows individual pages to focus only on their unique content while inheriting the same visual framework.

Page Navigation and User Flow

In this stage, we began to functionalize interactive elements such as buttons, hyperlinks to ensure the website provides a smooth and intuitive experience. The overall user flow had already been defined in our early-stage Figma design, a clear user flow that reflects typical interactions between patients and doctors in our early-stage Figma design, reflecting typical interactions between patients and doctors. Each page was connected according to the logical steps users would take when navigating through the platform. For example, clicking the “Register” button on the homepage directs users to the registration form, and buttons on the “Dashboard” page are linked to corresponding pages. These workflows were clearly visualized in our Figma prototype.(**TODO:need picture**) The main goal of this stage is to ensure that all

the interactive elements function as the origin design. In the next stage, we plan to test the whole website work appropriately and collect the disadvantages that can be improved.

User Flow and Functional Testing

Challenges Encountered During Frontend Development

One of the biggest challenges encountered while performing frontend development was how to break down the Figma-based design system into Thymeleaf templates. Even though the Figma designs created a solid visual reference for layout, typography, and spacing between elements, the server-side rendering approach of Thymeleaf required the interface to be split into smaller, more manageable parts. It introduced additional difficulty in the process of converting pixel-perfect designs into coherent HTML templates with alignment, responsiveness, and style consistency preserved. Visual parity usually needed to be delivered by iterative adjustments of the HTML structure as well as accompanying CSS.

An analogous problem involved the handling of shared styles and bits within the multi-page Thymeleaf application. Shared interface pieces, the sidebar navigation, header, and footer, were included as reusable fragments to foster consistency and reduce redundancy. With this came, however, the risk of side effects: a modification of a shared fragment would likely disrupt the styling or layout of multiple pages. To tackle this, the development team implemented a formalized update process, such as fragment-specific testing and peer review prior to merge of changes. This procedure ensured global component updates maintained visual correctness without causing regressions in other parts of the system.

3.3 Back-end Design and Implementations

Chapter 4

Evaluation and Testing

Begins a chapter. Example: When the beloved cellist (Christopher Walken - outstanding) of a world-renowned string quartet receives a life-changing diagnosis, the group's future suddenly hangs in the balance: suppressed emotions, competing egos and uncontrollable passions threaten to derail years of friendship and collaboration. Featuring a brilliant ensemble cast (including Philip Seymour Hoffman, Catherine Keener and Mark Ivanir as the three other quartet members), it is a fascinating look into the world of working musicians, and an elegant homage to chamber music and the cultural world of New York. The music, of course, is ravishing (the score is the work of regular David Lynch collaborator Angelo Badalamenti): A Late Quartet hits all the right notes.

Chapter 5

Conclusion

Begins a chapter. Example: When the beloved cellist (Christopher Walken - outstanding) of a world-renowned string quartet receives a life-changing diagnosis, the group's future suddenly hangs in the balance: suppressed emotions, competing egos and uncontrollable passions threaten to derail years of friendship and collaboration. Featuring a brilliant ensemble cast (including Philip Seymour Hoffman, Catherine Keener and Mark Ivanir as the three other quartet members), it is a fascinating look into the world of working musicians, and an elegant homage to chamber music and the cultural world of New York. The music, of course, is ravishing (the score is the work of regular David Lynch collaborator Angelo Badalamenti): A Late Quartet hits all the right notes.

Chapter 6

Reference

Appendix A

Appendix A

Begins an appendix

Bibliography

- [1] F. E. M. HOSPITAL, *Official line account.*
<https://line.me/R/ti/p/@femh>, 2025.
Screenshot taken by the author.
- [2] INTERNATIONAL LABOUR ORGANIZATION, *Decent work for rural health workers.*
https://www.ilo.org/global/topics/economic-and-social-development/rural-development/WCMS_768518/lang--en/index.htm, 2021.
Accessed: 2025-08-02.
- [3] N. TRUST, *Digital primary care: Where next?*
<https://www.nuffieldtrust.org.uk/research/digital-primary-care-where-next>,
2022.
Accessed: 2025-08-02.
- [4] UNITED NATIONS, *Ensure healthy lives and promote well-being for all at all ages.*
<https://sdgs.un.org/goals/goal3>, 2022.
Accessed: 2025-08-02.