National College of Ireland

National College of Ireland

BSc (Honours) in Computing

Software Development

2020/2021

Lucas da Silva

X17118361

X17118361@student.ncirl.ie

My Hair Done
STYLE OR GET STYLE WITH US

My Hair Done

Technical Report

Contents

Executive Summary

In this technical report I cover the main aspects of the developments stages of the My Hair Done application. I discuss in detail what specific problems this application intends to solve what as well is my inspiration to develop this application.

In the technical sections I present what are the main technologies used to achieve the results I had in mind when designing this application, as well as a justification of why I used each one of them to make My Hair Done an application that stands out as a very edge application using the latest concepts of containerization and Cloud Native Software Development strategies.

1.0 Introduction

1.1. Background

The idea for this project or at least the need for a solution in this area has been with me for over 10 years and I think that my final year project is the perfect opportunity to finally develop my own solution.

My main goal is to provide an application that streamlines the process of getting a haircut. To differentiate this application from other existing solutions, I'm planning to build it in way that helps independent hairdressers to promote their work and style and have the same level of visibility of established salons.

To add more value to the customer that decides to install the app, I plan to include an AR feature that allows the customer to try different haircuts using their camera. Once their happy with that hairstyle, they can look for hairdressers that are specialists in that specific hairstyle. This strategy in turn should also drive more traffic through the application.

1.2. Aims

The objective of this project is to build an application that Independent Hairdressers and Customers to find each other and structure the booking process making it a simple, intuitive, and consistent experience regardless of who the hairdresser or customer is.

For Hairdressers, the application will serve as a platform to publish their portfolio and style characteristics so that customers looking for a hairdresser specialized in each style or location can find a match easily.

For customers, the application will help them on 3 different fronts: 1st The AI Hairstyle builder will help users see what they would look like with a chosen haircut style. 2nd The application will help them to find Hairdressers that match their hairstyle choice and finally, customers will be able to book time with their Hairdresser of choice.

1.3. Technology

Web Frontend

The Web Frontend will be built using React which is a JavaScript Framework. I opted for using this framework because it is considered to be the standard choice for Frontend development these days and I want to experience how different it is to use a framework instead of plain HTML, CSS, and JavaScript.

Backend

The backend will be built as a separate application using NodeJS and will communicate with the Frontend using HTTP Calls following the REST API architectural style.

I chose to develop the backend completely independent from the Frontend to allow the backend API to serve multiple frontends that I may develop in the future like an iOS/Android application to integrate with the same Backend and Database as the Web application, and this will be possible with no changes necessary on the backend.

The feature that allows users to try new hairstyles virtually will require a Machine Learning/AR library to be implemented as part of the backend. I do not know yet what libraries are available for NodeJS and intend on investigating this further over the next month. If I am not satisfied with the libraries I find, I may opt to deploy this feature as a micro-service developed in Java as I know that ML libraries are more popular in Java than in NodeJS.

Database

I plan to use MySQL as the engine for the main database.

As only the backend will interact with the database, I will probably use the Knex ORM (Object Relational Mapping) library for NodeJS as my query builder.

Infrastructure

- **Frontend**
  Will be served by Amazon Linux 2 EC2 instances. The instances will be part of an Auto Scaling Group to allow for scalability. A Load balancer will be responsible for balancing the load correctly across the instances live in the Auto Scaling Group.
- **Backend**
  Will be served by Amazon Linux 2 EC2 instances. The instances will be part of an Auto Scaling Group to allow for scalability. A Load balancer will be responsible for balancing the load correctly across the instances live in the Auto Scaling Group.
- **Postgres Database**
  AWS RDS service will be used to host the Postgres Database.
- **Storage**
  AWS S3 will be used to store pictures uploaded to the application.

- **Mailing**
  AWS SES (Simple Email Service) will be used to support emailing out users when verifying email addresses, resetting the password, etc…
- **DNS**
  AWS Route53 DNS will be my DNS provider.
- **HTTPS Certificate**
  AWS Certificate Manager which natively integrates with AWS Route53 will be used to provide HTTPS certificate to my Web Frontend.
- **Secrets**
  Instead of storing secrets and API keys in the code or a .config file, I plan to use AWS Secrets Manager to store all my secret credentials safely which can be retrieved at runtime, drastically reducing the chances of these secrets being captured from my code.

CI/CD

To expedite how quickly I can test and deploy new versions of my frontend and backend, I will implement a basic CI/CD pipeline using AWS DevOps tools like AWS CodeBuild and AWS CodeDeploy. These tools natively integrate with the rest of my infrastructure automating the build, test, and deployment steps.

The application source code will be maintained in a public GitHub Repository which also integrates very well with the AWS DevOps suite.

1.4. Structure

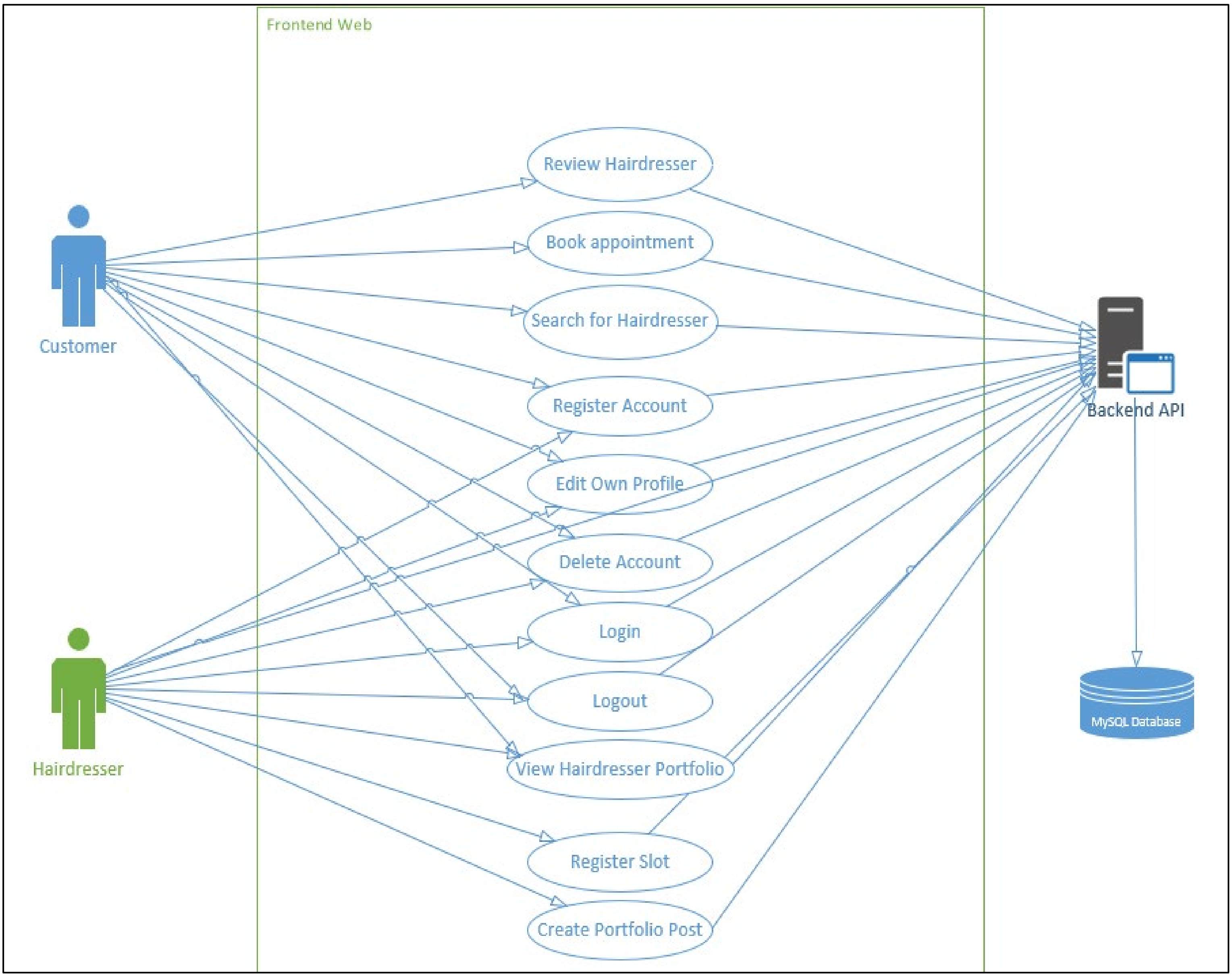
Provide a brief overview of the structure of the document and what is addressed in each section.

2.0 System

2.1. Requirements

2.1.1. Functional Requirements

2.1.1.1. Use Case Diagram



2.1.1.2. Requirement 1 <Register Account>

Description

This requirement describes the requirement of a customer or hairdresser registering a new account in the application. The user (hairdresser or customer) interacts with the Web frontend application, which sends a HTTP request to the backend server who process it and stores the data in the MySQL database.

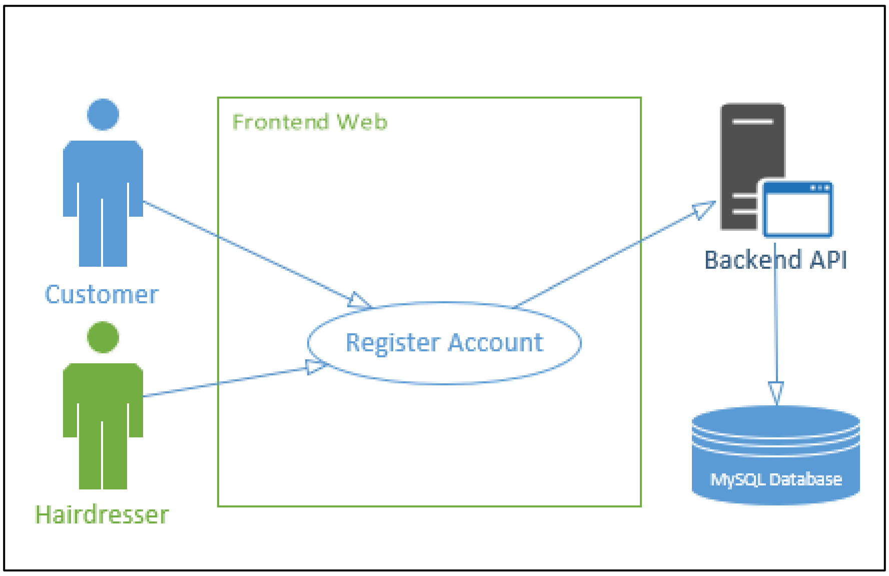
Use Case

UC-01

Scope

The scope of this use case is to describe the process of a hairdresser or customer registering a new account

Use Case Diagram



Flow Description

Precondition

The user must be at the My Hair Done website

Activation

This use case starts when a <Customer> or <Hairdresser> reaches the "Register" web page.

Main flow

1. User opens the Register web page
2. The <Customer> or <Hairdressers> inserts the required information and click on "Register" button

3. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload
4. The Backend API processes that data and saves it in the MySQL Database

Alternate flow

1. User opens the Register web page
2. The <Customer> or <Hairdressers> inserts the required information and click on “Register”
3. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload
4. The backend API identifies that email is already registered and communicates back to the frontend
5. Frontend displays error to user

Exceptional flow

User leaves the Register page before completing the registration form

Termination

Frontend web displays the message “User registered successfully”.

Post condition

A new user account is created in the database.

2.1.1.3. Requirement 2 <Login>

Description

This requirement describes the requirement of a customer or hairdresser logging into the application. The user (hairdresser or customer) interacts with the Web frontend application, which sends a HTTP request to the backend server who process it and stores the data in the MySQL database.

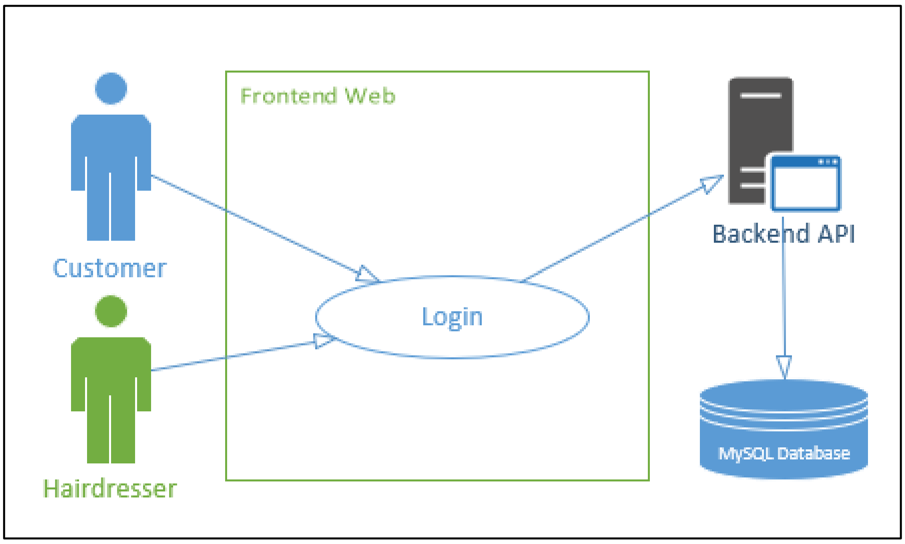
Use Case

UC-02

Scope

The scope of this use case is to describe the process of a hairdresser or customer logging into the application

Use Case Diagram



Flow Description

Precondition

The user must be at the My Hair Done website and have a registered account

Activation

This use case starts when a <Customer> or <Hairdresser> reaches the "Login" web page.

Main flow

1. User opens the Login web page
2. The <Customer> or <Hairdressers> inserts the required information and click on "Login" button
3. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload

4. The Backend API receives that data and compares the email and password received with what's present in the MySQL Database
5. If the information matches, backend API returns a JSON web token to the Frontend Web
6. Frontend Web saves the JSON web token in the browser storage and redirects the user to their profile page

Alternate flow

1. User opens the Register web page
2. The <Customer> or <Hairdressers> inserts the required information and click on "Register"
3. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload
4. The backend API can match the received email and password with what's already in the database and returns an error message to the Frontend Web
5. Frontend displays an authentication error to user

Exceptional flow

User leaves the Login page before completing the login form

Termination

Frontend web redirects user to the user's profile page.

Post condition

User can now access additional feature in the application.

2.1.1.4. Requirement 3 <Logout>

Description

This requirement describes the requirement of a customer or hairdresser being able to logout of the application.

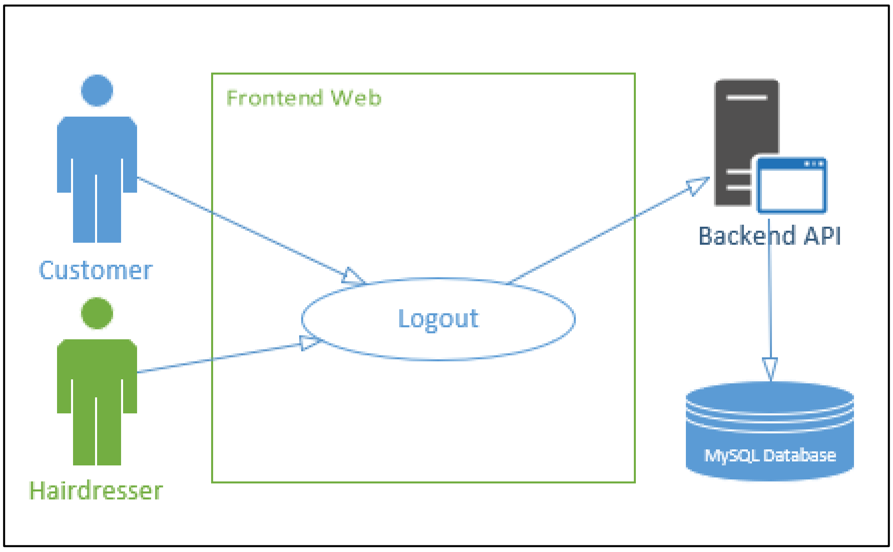
Use Case

UC-03

Scope

The scope of this use case is to describe the process of a hairdresser or customer logging out of the application

Use Case Diagram



Flow Description

Precondition

The user must be at the My Hair Done website and have already logged into the application

Activation

This use case starts when a <Customer> or <Hairdresser> hits the "Logout" button.

Main flow

1. User clicks in the "Logout" button
2. The Frontend Web deletes the JSON web token stored in their browser local storage
3. The Frontend Web redirects the user to the home page

Termination

Frontend web redirects user to the home page.

Post condition

User can no longer access features that require login.

2.1.1.5. Requirement 4 <Edit Own Profile>

Description

This requirement describes the requirement of a customer or hairdresser being able to edit the information in their own profile. The user (hairdresser or customer) interacts with the Web frontend application, which sends a HTTP request to the backend server who process it and stores the data in the MySQL database.

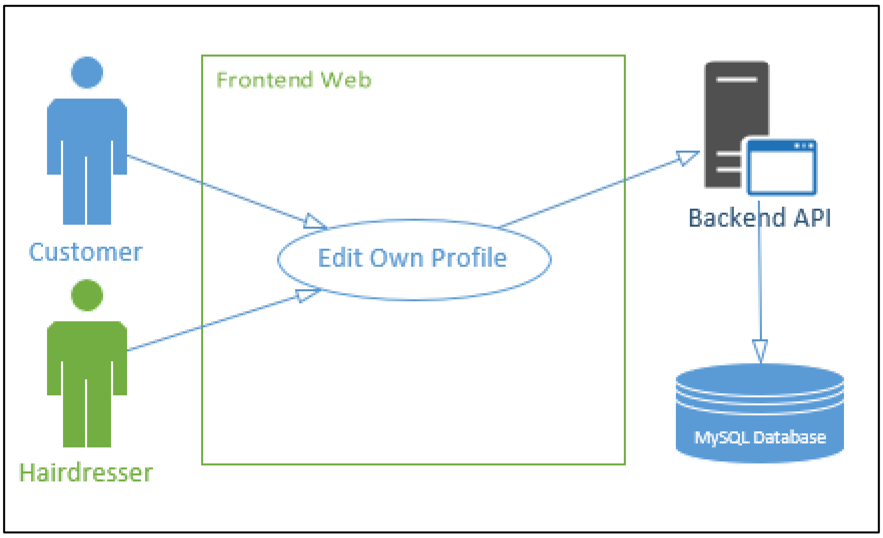
Use Case

UC-04

Scope

The scope of this use case is to describe the process of a hairdresser or customer updating their own profile

Use Case Diagram



Flow Description

Precondition

The user must be at the My Hair Done website, have a registered account and be logged into their account.

Activation

This use case starts when a <Customer> or <Hairdresser> reaches their profile page and hit the “Edit Profile” button.

Main flow

1. The <Customer> or <Hairdressers> fills the form with the required information to update their profile
2. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload
3. The Backend API receives that data and updates the user profile in the MySQL database using the new information provided and sends success message back to the frontend
4. Frontend Web redirects user to their own profile page

Exceptional flow

User cancels the profile update operation before completing the form.

Termination

Frontend web redirects user to the user’s profile page.

Post condition

User profile now contains the updated information.

2.1.1.6. Requirement 5 <Delete Account>

Description

This requirement describes the requirement of a customer or hairdresser being able to delete their registered account, erasing their personal data from the application database. The user (hairdresser or customer) interacts with the Web frontend application, which sends a HTTP request to the backend server who process it and stores the data in the MySQL database.

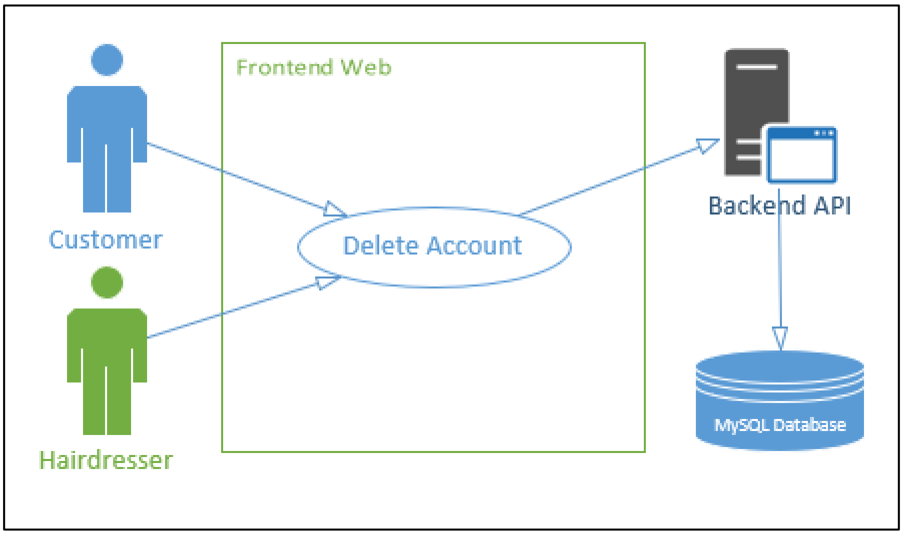
Use Case

UC-05

Scope

The scope of this use case is to describe the process of a hairdresser or customer deleting their own registered account

Use Case Diagram



Flow Description

Precondition

The user must be at the My Hair Done website, have a registered account and be logged using a registered account

Activation

This use case starts when a <Customer> or <Hairdresser> hits the "Delete Account" button at their account profile page.

Main flow

1. User opens their account profile page
2. The <Customer> or <Hairdressers> clicks on the “Delete Account” button
3. Frontend Asks for confirmation from the user
4. If user confirms the delete request, Frontend Web send request to Backend API with the account ID to be deleted
5. The Backend API receives that data and delete the account information form the MySQL database and sends success message back to the frontend
6. Frontend Web redirects the user to the home page

Alternate flow

1. User opens their account profile page
2. The <Customer> or <Hairdressers> clicks on the “Delete Account” button
3. Frontend Asks for confirmation from the user
4. User cancels the deletion request
5. Frontend Web redirects the user to the user’s profile page

Termination

Frontend web redirects user to the home page.

Post condition

User’s registered account no longer exists in the MySQL database.

2.1.1.7. Requirement 6 <View Hairdresser Portfolio>

Description

This requirement describes the requirement of a customer or hairdresser being able to see all the portfolio posts that hairdresser has made in the past. The user (hairdresser or customer) interacts with the Web frontend application, which sends a HTTP request to the backend server who process it and stores the data in the MySQL database.

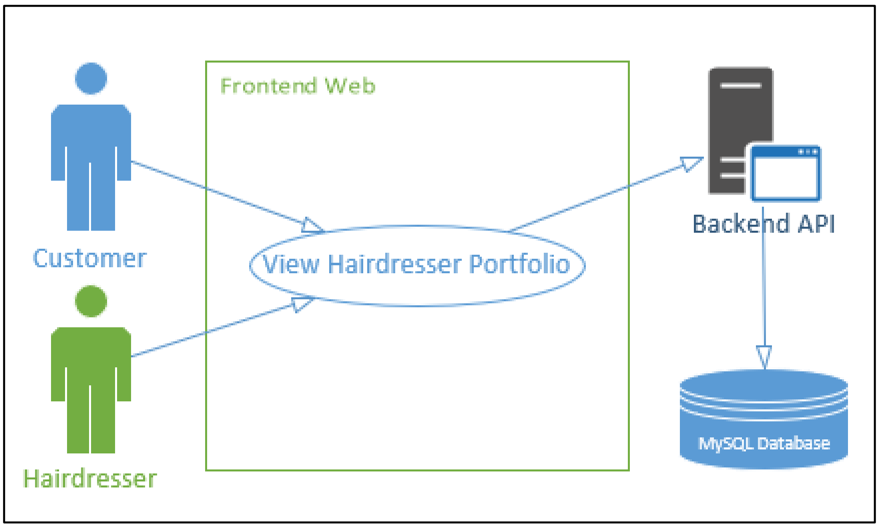
Use Case

UC-06

Scope

The scope of this use case is to describe the process of a hairdresser or customer viewing the past portfolio posts of a hairdresser

Use Case Diagram



Flow Description

Precondition

The user must be at the My Hair Done website

Activation

This use case starts when a <Customer> or <Hairdresser> hits the “View Portfolio” button at the hairdresser’s profile page.

Main flow

1. User reaches the profile page of a hairdresser
2. User hits the “View Portfolio” button
3. Frontend requests the portfolio posts from the backend API
4. Backend API retrieves the portfolio posts data from the MySQL database
5. Backend API sends back the portfolio posts data to the frontend
6. Frontend lists all posts

Alternate flow

1. User reaches the profile page of a hairdresser
2. User hits the “View Portfolio” button
3. Frontend requests the portfolio posts from the backend API
4. Backend API retrieves the portfolio posts data from the MySQL database, which comes out empty
5. Backend API sends back a response stating that no portfolio post was available
6. Frontend display message to user that no portfolio post has been made by that hairdresser.

Exceptional Flow

User leaves the page before the portfolio posts finished loading.

Termination

Frontend web displays all portfolio posts available in the MySQL DB to the user.

2.1.1.8. Requirement 7 <Register Slot>

Description

This requirement describes the requirement of a hairdresser being able to register the time slots where this professional is available to accept bookings.

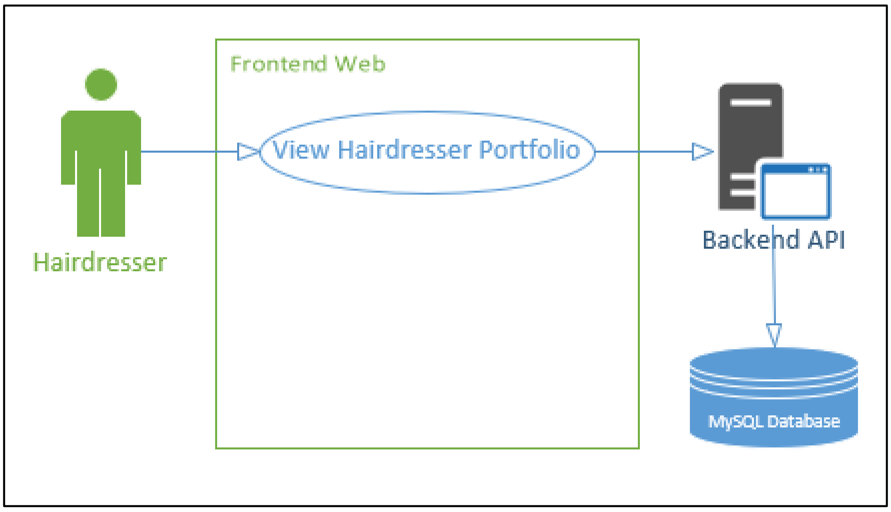
Use Case

UC-07

Scope

The scope of this use case is to describe the process of a hairdresser creating a time slot to receive bookings

Use Case Diagram



Flow Description

Precondition

The hairdresser must be at the My Hair Done website and have a registered account

Activation

This use case starts when a <Hairdresser> click on the “Create Time Slot” button at their profile page

Main flow

1. The <Hairdressers> inserts the required information and click on “Create Slot” button
2. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload

3. The Backend API receives that data and creates a new time slot in the MySQL Database
4. Backend API returns a success message to the Frontend Web
5. Frontend Web displays confirmation to the hairdresser and redirects the hairdresser to their profile page

Exceptional flow

User leaves the create slot page before completing the form

Termination

Frontend web redirects user to the user's profile page.

Post condition

Hairdresser now has a new time slot available for booking.

2.1.1.9. Requirement 8 <Create Portfolio Post>

Description

This requirement describes the requirement of a hairdresser being able to create portfolio posts to display their work to customers.

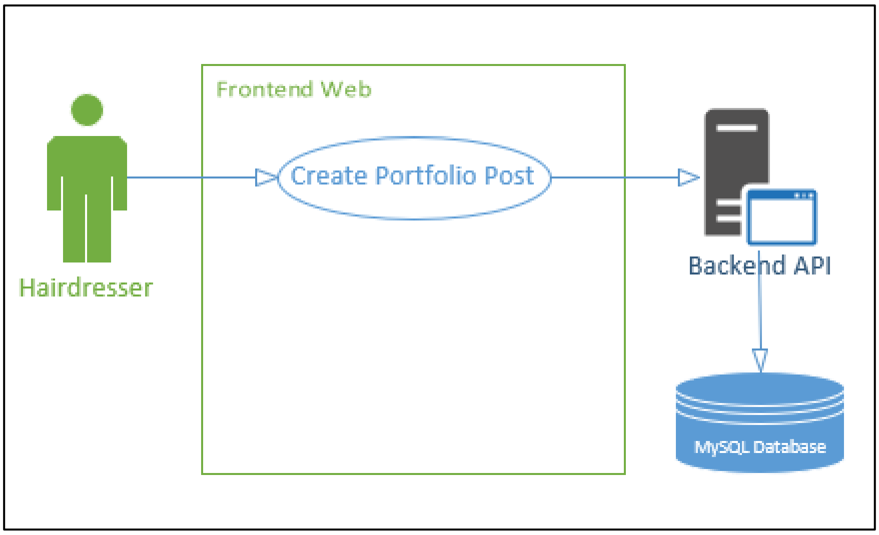
Use Case

UC-08

Scope

The scope of this use case is to describe the process of a hairdresser creating a a new portfolio post.

Use Case Diagram



Flow Description

Precondition

The hairdresser must be at the My Hair Done website and have a registered account

Activation

This use case starts when a <Hairdresser> click on the "New Portfolio Post" button at their portfolio page

Main flow

1. The <Hairdressers> inserts the required information and click on "Create Portfolio Post" button
2. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload
3. The Backend API receives that data and creates a new portfolio post in the MySQL Database
4. Backend API returns a success message to the Frontend Web
5. Frontend Web displays confirmation to the hairdresser and redirects the hairdresser to new post display page

Exceptional flow

User leaves the new portfolio page before completing the form.

Termination

Frontend web redirects user to the new post display page.

Post condition

Hairdresser now has a new portfolio post to display to potential customers.

2.1.1.10. Requirement 9 <Search for Hairdresser>

Description

This requirement describes the requirement of a customer being able to search for hairdressers based on name, location and style.

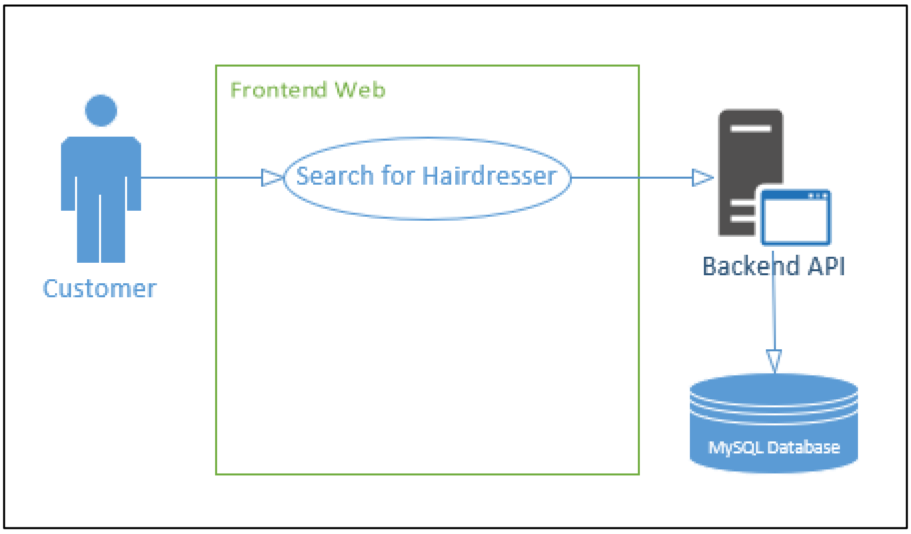
Use Case

UC-09

Scope

The scope of this use case is to describe the process of a customer searching for a hairdresser.

Use Case Diagram



Flow Description

Precondition

The customer must be at the My Hair Done website

Activation

This use case starts when a <Customer> click on the "Hairdressers" link on the My Hair Done Homepage.

Main flow

1. The <Customer> inserts the required hairdresser information that the user is looking for
2. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload
3. The Backend API receives that data and retrieves all the hairdressers that match the filter set by the customer from the MySQL Database

4. Backend API returns a success message to the Frontend Web with the list of hairdressers in the JSON payload
5. Frontend Web displays the list of hairdressers back to the user

Exceptional flow

User leaves the page before the loading of hairdressers is complete.

Termination

Frontend web display the list of hairdressers to the user.

Post condition

Customer now has a list of hairdressers to choose from.

2.1.1.11. Requirement 10 <Post Hairdresser Review>

Description

This requirement describes the requirement of a customer being able to post a review about a hairdresser after undertaking a haircut with the hairdresser.

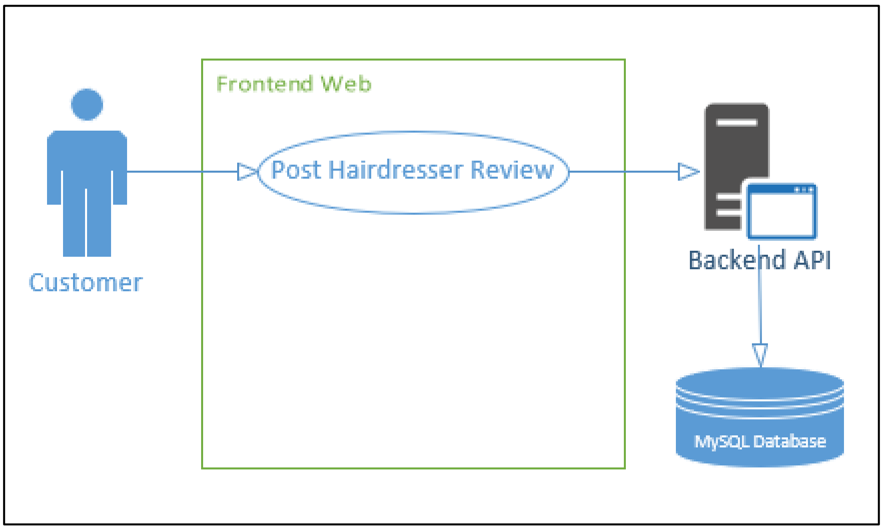
Use Case

UC-10

Scope

The scope of this use case is to describe the process of a user posting a review about his/her experience with a hairdresser.

Use Case Diagram



Flow Description

Precondition

The customer must be at the My Hair Done website, have a registered account, be logged in and have attended an appointment with the hairdresser that will receive this review.

Activation

This use case starts when a <Customer> clicks on the “Review your experience” button at their appointment booking page.

Main flow

1. The <Customer> inserts the required information and click on “Post Review” button
2. The Frontend Web send request to Backend API with the information filled in the JSON body as the payload
3. The Backend API receives that data and creates a new review in the MySQL Database
4. Backend API returns a success message to the Frontend Web
5. Frontend Web displays confirmation to the customer that his review has been published and redirects the customer to his new review display page

Exceptional flow

User leaves the new review page before completing the form.

Termination

Frontend web redirects customer to the new review display page.

Post condition

A new review is now publicly available at the hairdresser profile.

2.1.1.12. Requirement 11 <Book Appointment>

Description

This requirement describes the requirement of a customer being able to book an appointment with a hairdresser.

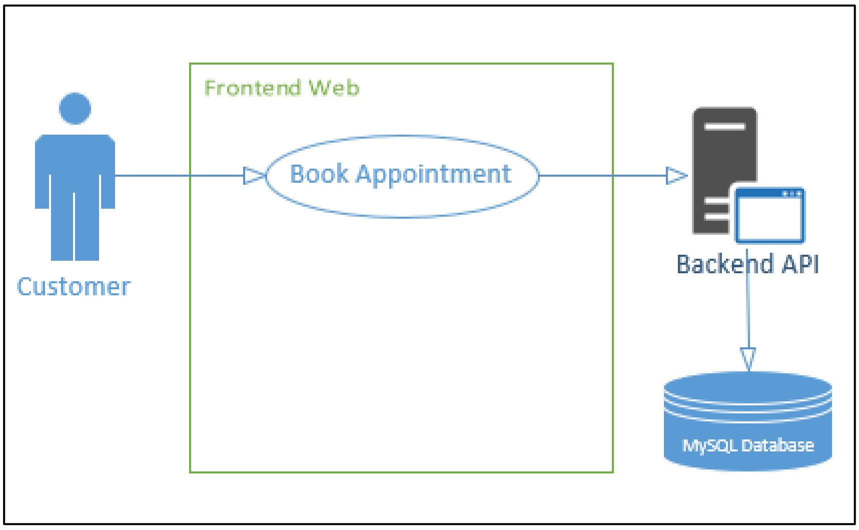
Use Case

UC-11

Scope

The scope of this use case is to describe the process of a customer creating a new booking with a hairdresser.

Use Case Diagram



Flow Description

Precondition

The customer must be at the My Hair Done website, have a registered account, be logged in and found a hairdresser that suits the user's needs.

Activation

This use case starts when a <Customer> click on the "Book Now" button at hairdresser's profile page.

Main flow

1. The <Customer> Selects the desired date for the booking
2. The Frontend Web send request to get all available slots on that date to the Backend API with the information filled in the JSON body as the payload
3. The Backend API receives that data and retrieves all available slots from the MySQL Database
4. Backend API returns a message to the Frontend Web with the list of slots for the user to choose from
5. Frontend Web displays all available slots
6. The <Customer> picks a slot from the list and click on “Confirm Booking” button
7. The Frontend Web send request to the Backend API with the information filled in the JSON body as the payload
8. The Backend API receives that data and creates a new booking the MySQL Database
9. Backend API returns a message to the Frontend Web with a success message
10. Frontend Web displays confirmation to the user

Exceptional flow

User leaves the new booking page before completing the booking request.

Termination

Frontend web redirects user to the new booking page.

Post condition

A new booking is now created for the hairdresser and user.

2.1.2. Data Requirements

Database details

Database Provider:	Amazon RDS
Database Engine:	MySQL 8.0.1
Name of database:	myhairdone
location of database:	myhairdone-db.cbs17gvkbruq.eu-west-1.rds.amazonaws.com

2.1.3. User Requirements

Availability across all browsers

All users expressed the desired to be able to access all application's features from any browser.

Use only non-colour-blind colours.

Users expressed concern over using colours that can make the UI painful to the eyes of colour-blind people and diminish their experience using their application.

Security

All users agreed that they wish their personal data is stored securely in the application database.

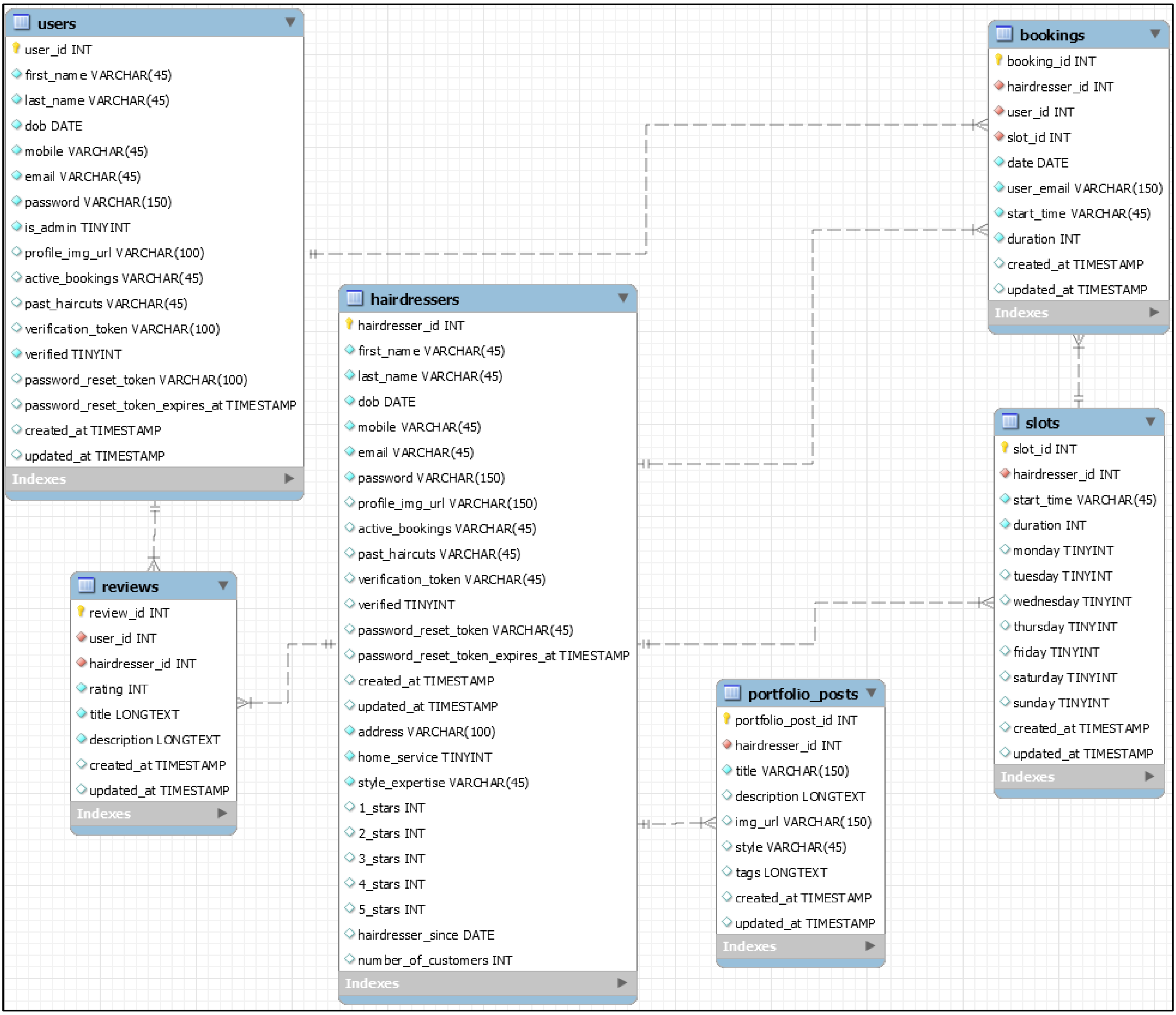
2.1.4. Environmental Requirements

Internet

The application requires the users to always have a healthy internet connection.

2.2. Design & Architecture

2.2.1. Database Entity Relationship Diagram



users
- user_id INT
- first_name VARCHAR(45)
- last_name VARCHAR(45)
- dob DATE
- mobile VARCHAR(45)
- email VARCHAR(45)
- password VARCHAR(150)
- is_admin TINYINT
- profile_img_url VARCHAR(100)
- active_bookings VARCHAR(45)
- past_haircuts VARCHAR(45)
- verification_token VARCHAR(100)
- verified TINYINT
- password_reset_token VARCHAR(100)
- password_reset_token_expires_at TIMESTAMP
- created_at TIMESTAMP
- updated_at TIMESTAMP

Indexes

reviews
- review_id INT
- user_id INT
- hairdresser_id INT
- rating INT
- title LONGTEXT
- description LONGTEXT
- created_at TIMESTAMP
- updated_at TIMESTAMP

Indexes

hairdressers
- hairdresser_id INT
- first_name VARCHAR(45)
- last_name VARCHAR(45)
- dob DATE
- mobile VARCHAR(45)
- email VARCHAR(45)
- password VARCHAR(150)
- profile_img_url VARCHAR(150)
- active_bookings VARCHAR(45)
- past_haircuts VARCHAR(45)
- verification_token VARCHAR(45)
- verified TINYINT
- password_reset_token VARCHAR(45)
- password_reset_token_expires_at TIMESTAMP
- created_at TIMESTAMP
- updated_at TIMESTAMP
- address VARCHAR(100)
- home_service TINYINT
- style_expertise VARCHAR(45)
- 1_stars INT
- 2_stars INT
- 3_stars INT
- 4_stars INT
- 5_stars INT
- hairdresser_since DATE
- number_of_customers INT

Indexes

bookings
- booking_id INT
- hairdresser_id INT
- user_id INT
- slot_id INT
- date DATE
- user_email VARCHAR(150)
- start_time VARCHAR(45)
- duration INT
- created_at TIMESTAMP
- updated_at TIMESTAMP

Indexes

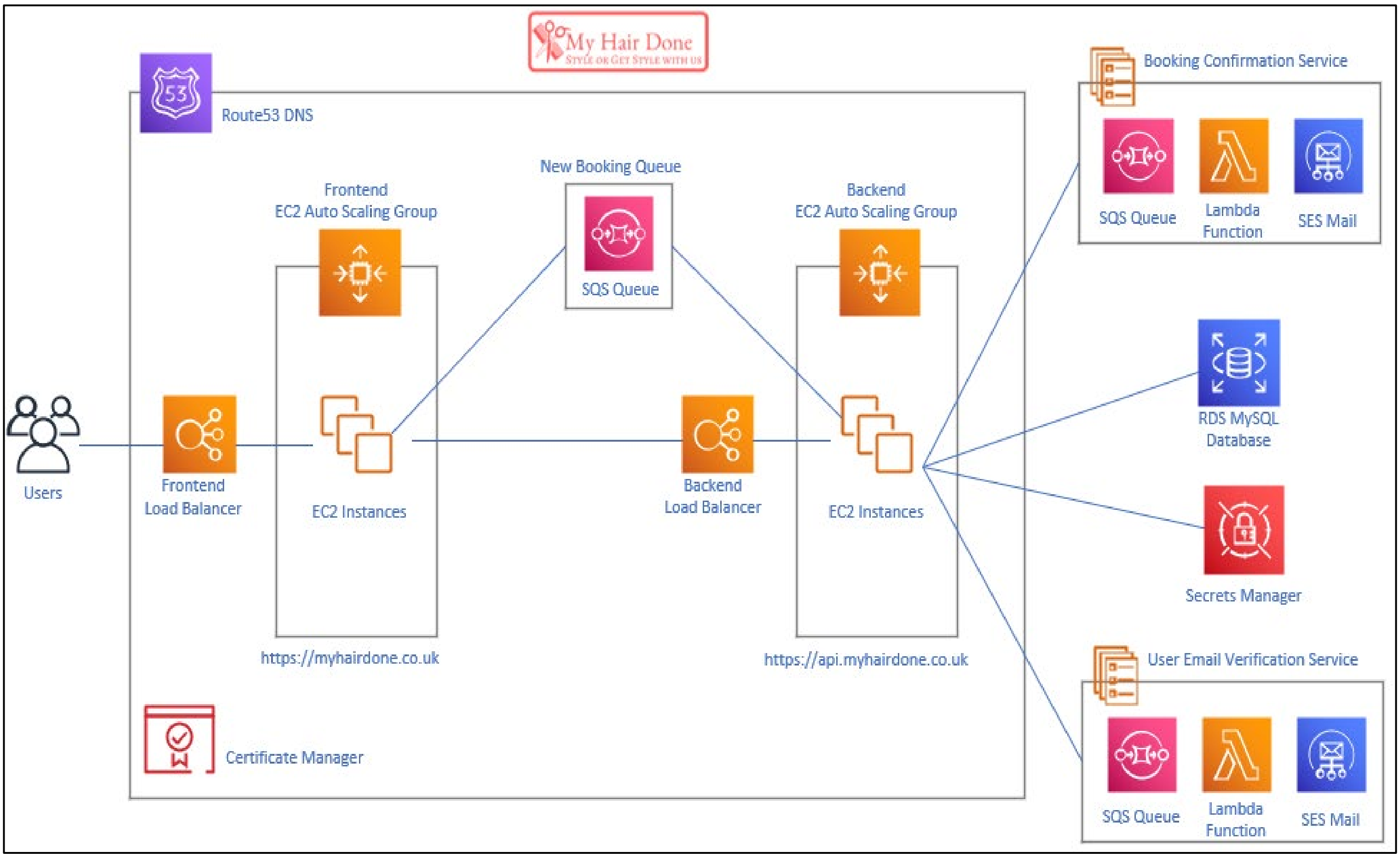
slots
- slot_id INT
- hairdresser_id INT
- start_time VARCHAR(45)
- duration INT
- monday TINYINT
- tuesday TINYINT
- wednesday TINYINT
- thursday TINYINT
- friday TINYINT
- saturday TINYINT
- sunday TINYINT
- created_at TIMESTAMP
- updated_at TIMESTAMP

Indexes

portfolio_posts
- portfolio_post_id INT
- hairdresser_id INT
- title VARCHAR(150)
- description LONGTEXT
- img_url VARCHAR(150)
- style VARCHAR(45)
- tags LONGTEXT
- created_at TIMESTAMP
- updated_at TIMESTAMP

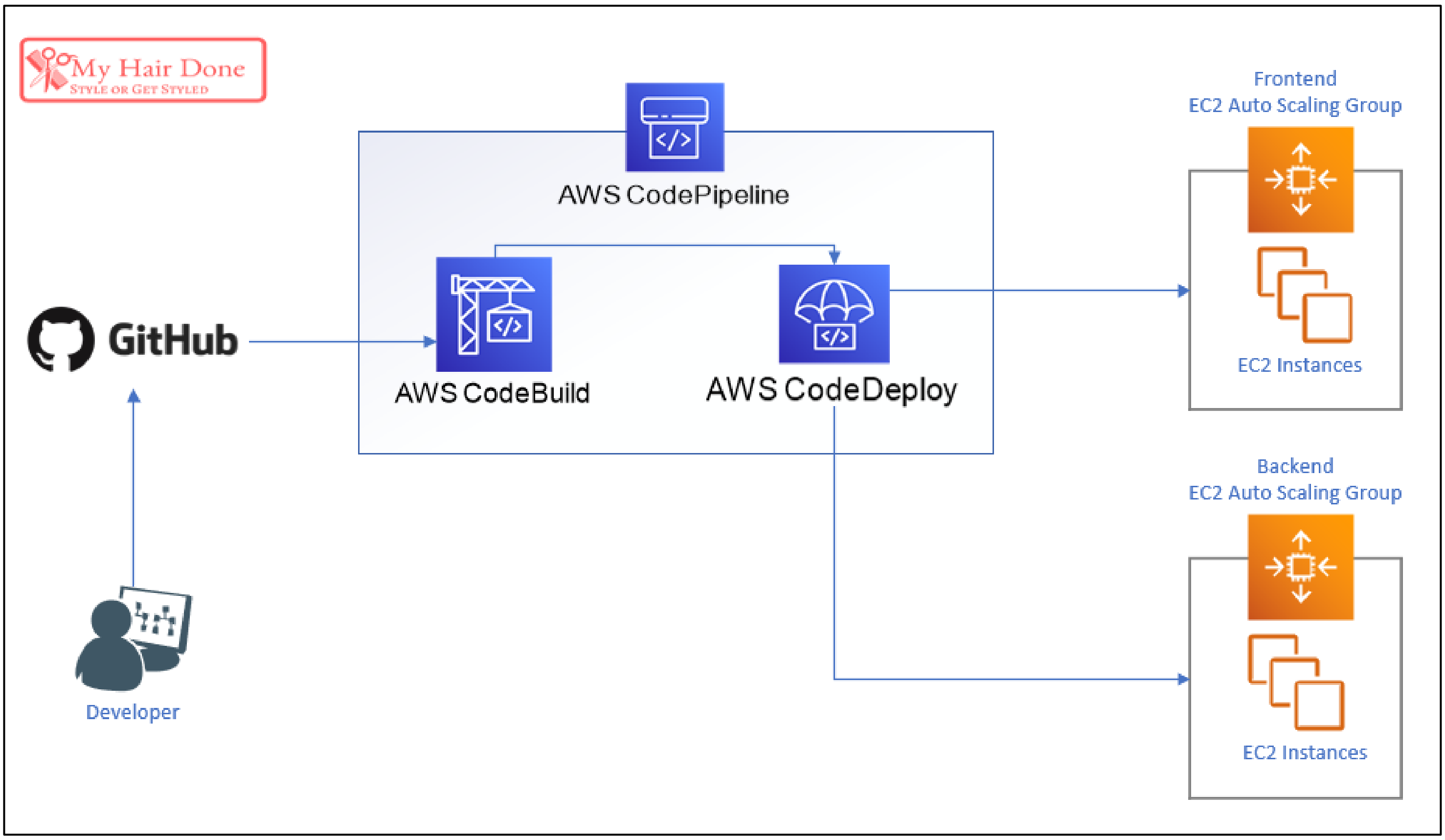
Indexes

2.2.2. System Architecture Diagram



2.3. Implementation

2.3.1. CI/CD Pipeline



2.3.2. Backend

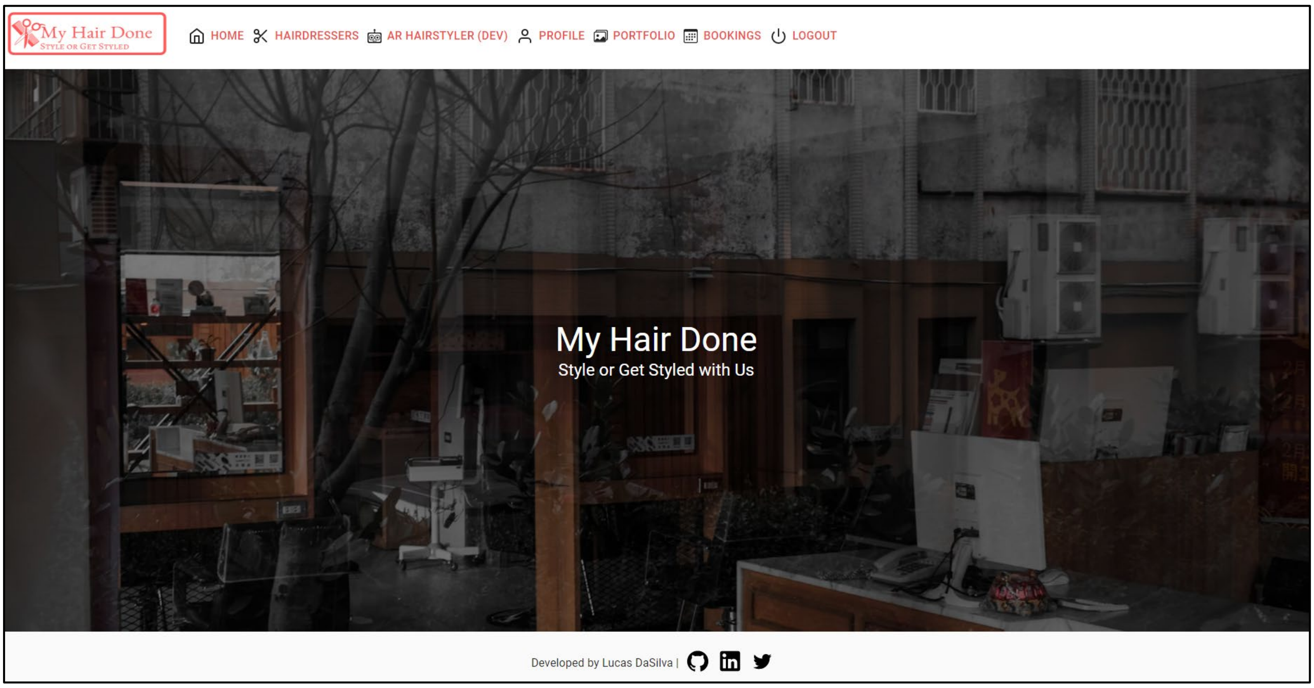
The backend of this web application is a RESTful API built using the Express Server library for NodeJS. As a RESTful API, the backend had a number of different Endpoints that are used to interact with different entities of the application and perform specific tasks.

I'm documenting the endpoints available along with the attributes of each entity using a tool called Swagger. This tool is considered a standard to document RESTful APIs and allows for quick and easy to maintain documentation via a .json configuration file at the root of the backend application. The API documentation is available at the root of the Backend API URL. https://api.myhairdone.co.uk

2.4. Graphical User Interface (GUI)

Below are a few screenshots of the main pages in the application.

2.4.1. Landing Page

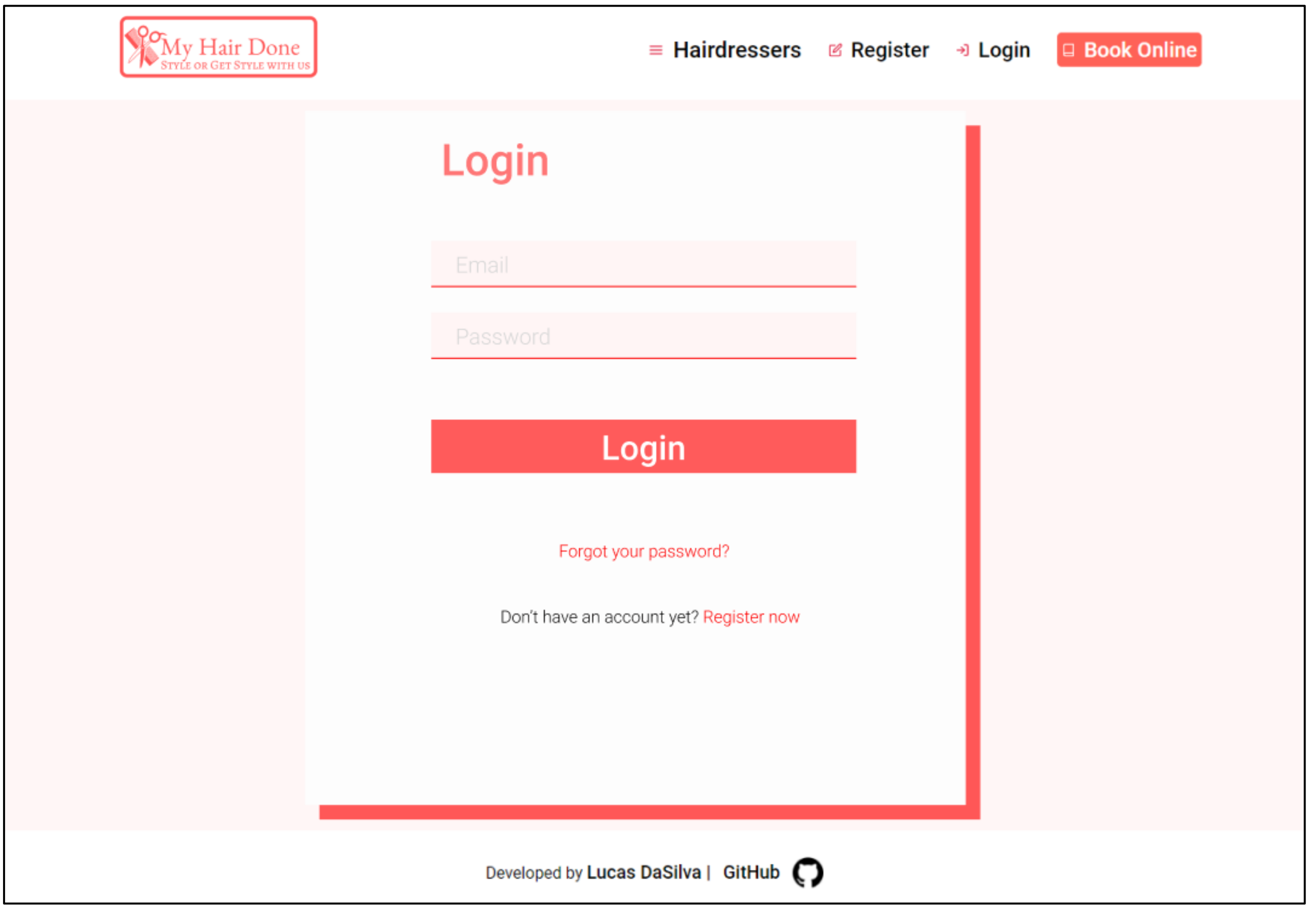


My Hair Done

Style or Get Styled with Us

Developed by Lucas DaSilva |

2.4.2. Login



Login

Email

Password

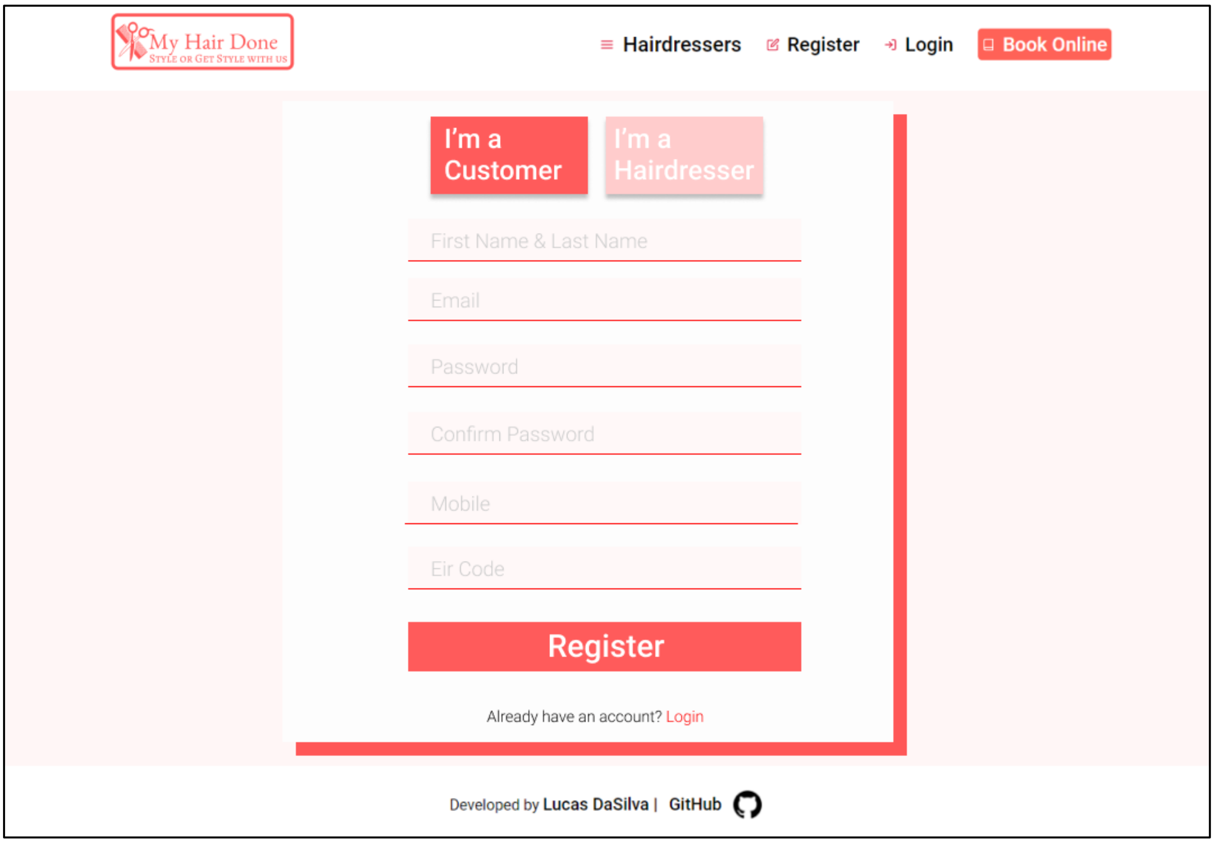
Login

Forgot your password?

Don't have an account yet? Register now

Developed by Lucas DaSilva | GitHub

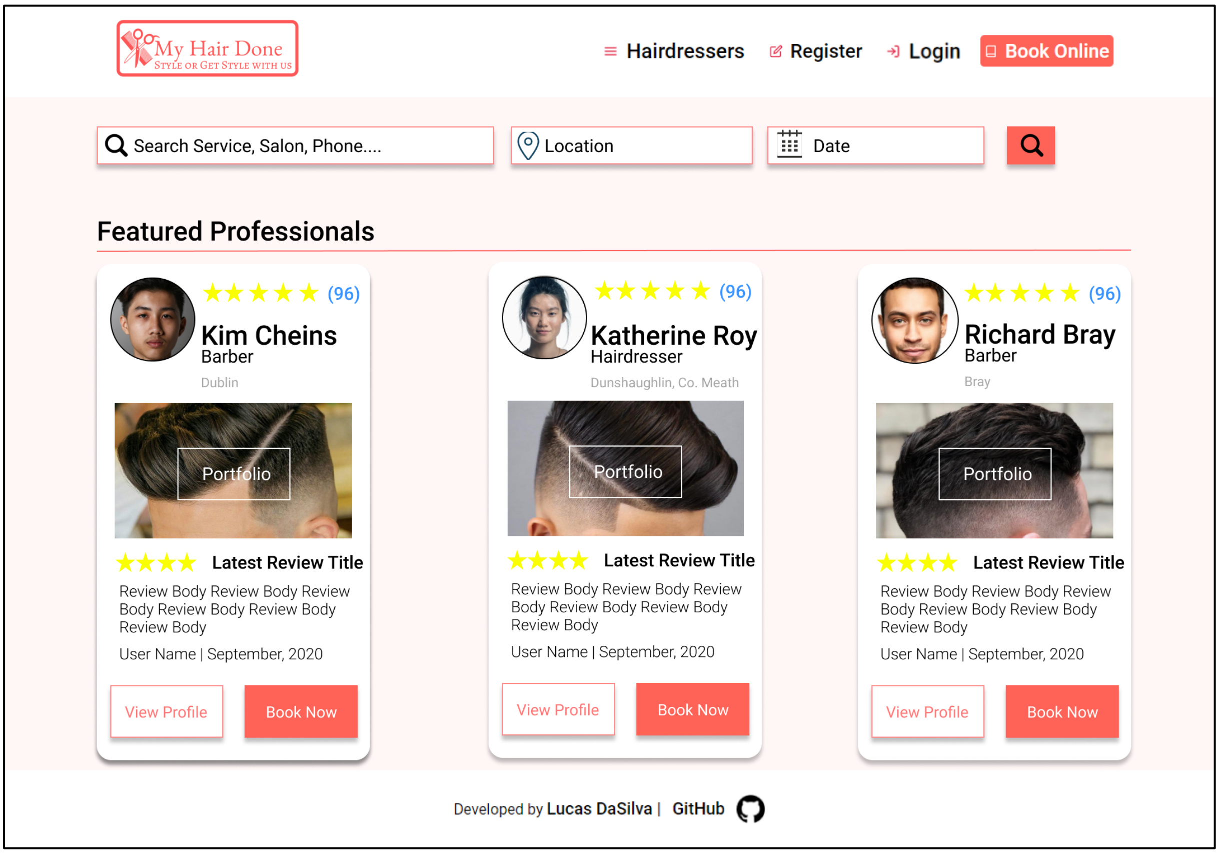
2.4.3. Register Page



Already have an account? Login

Developed by Lucas DaSilva | GitHub

2.4.4. Find Hairdressers Page



Featured Professionals

★★★★★ (96)
Kim Cheins
Barber
Dublin

★★★★ Latest Review Title
Review Body Review Body Review Body Review Body Review Body Review Body
User Name | September, 2020

★★★★★ (96)
Katherine Roy
Hairdresser
Dunshaughlin, Co. Meath

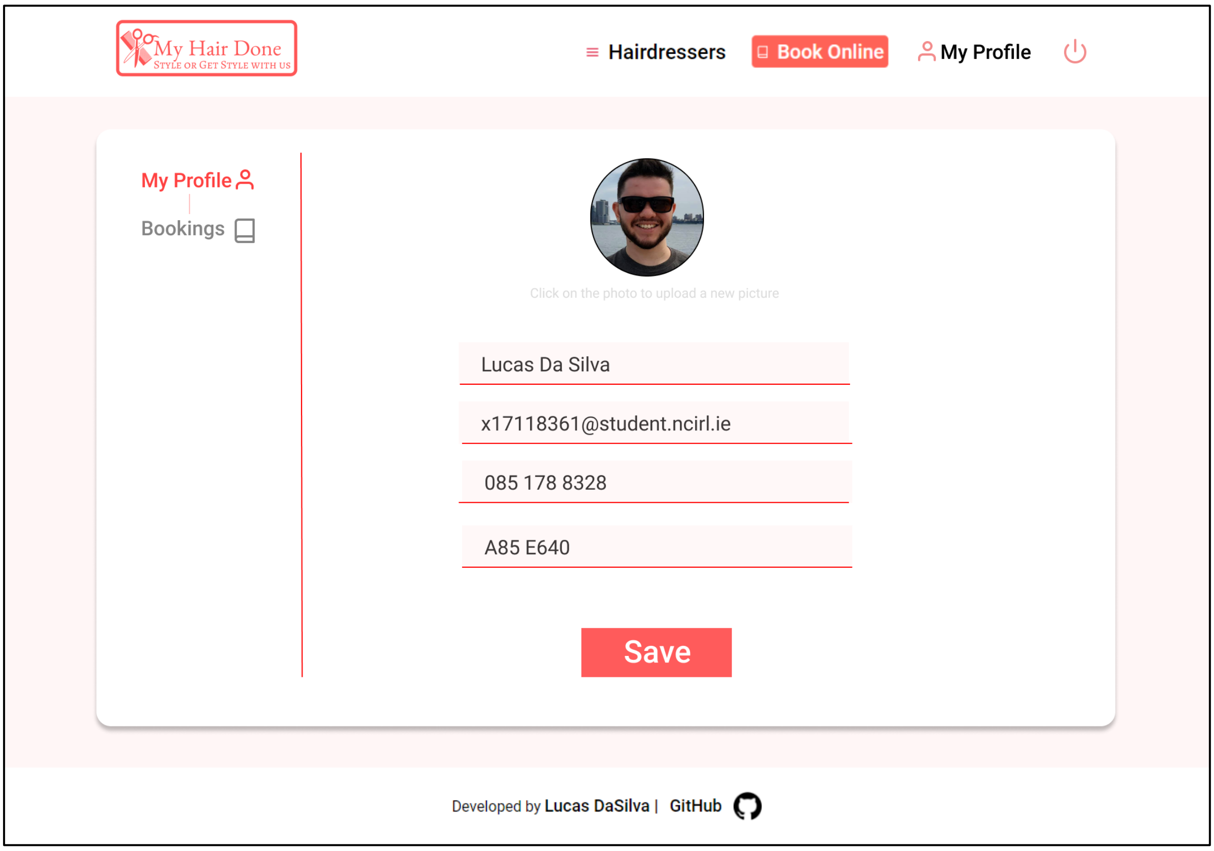
★★★★ Latest Review Title
Review Body Review Body Review Body Review Body Review Body
User Name | September, 2020

★★★★★ (96)
Richard Bray
Barber
Bray

★★★★ Latest Review Title
Review Body Review Body Review Body Review Body Review Body
User Name | September, 2020

Developed by Lucas DaSilva | GitHub

2.4.5. User Profile Page



My Hair Done
STYLE OR GET STYLE WITH US
≡ Hairdressers
Book Online
My Profile
My Profile
Bookings
Click on the photo to upload a new picture
Save
Developed by Lucas DaSilva | GitHub

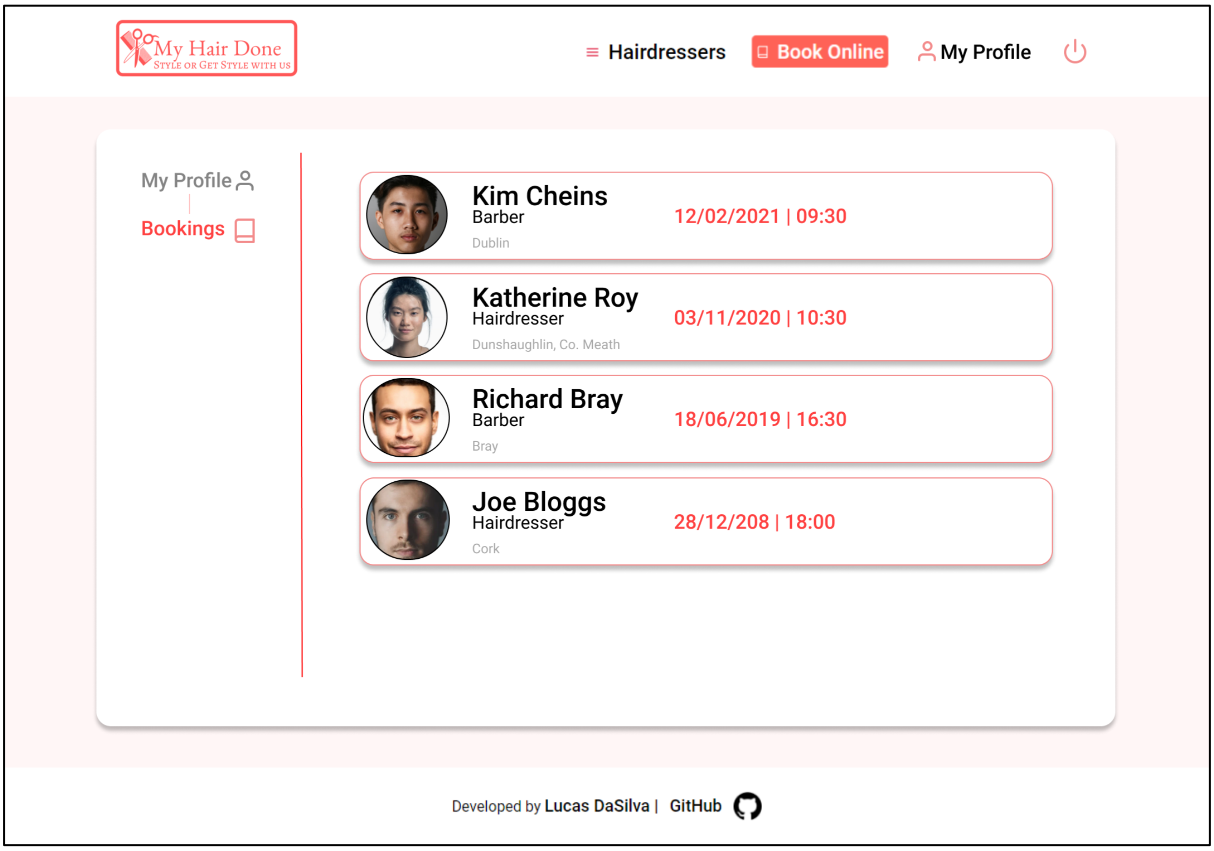
Lucas Da Silva

x17118361@student.ncirl.ie

085 178 8328

A85 E640

2.4.6. Bookings History



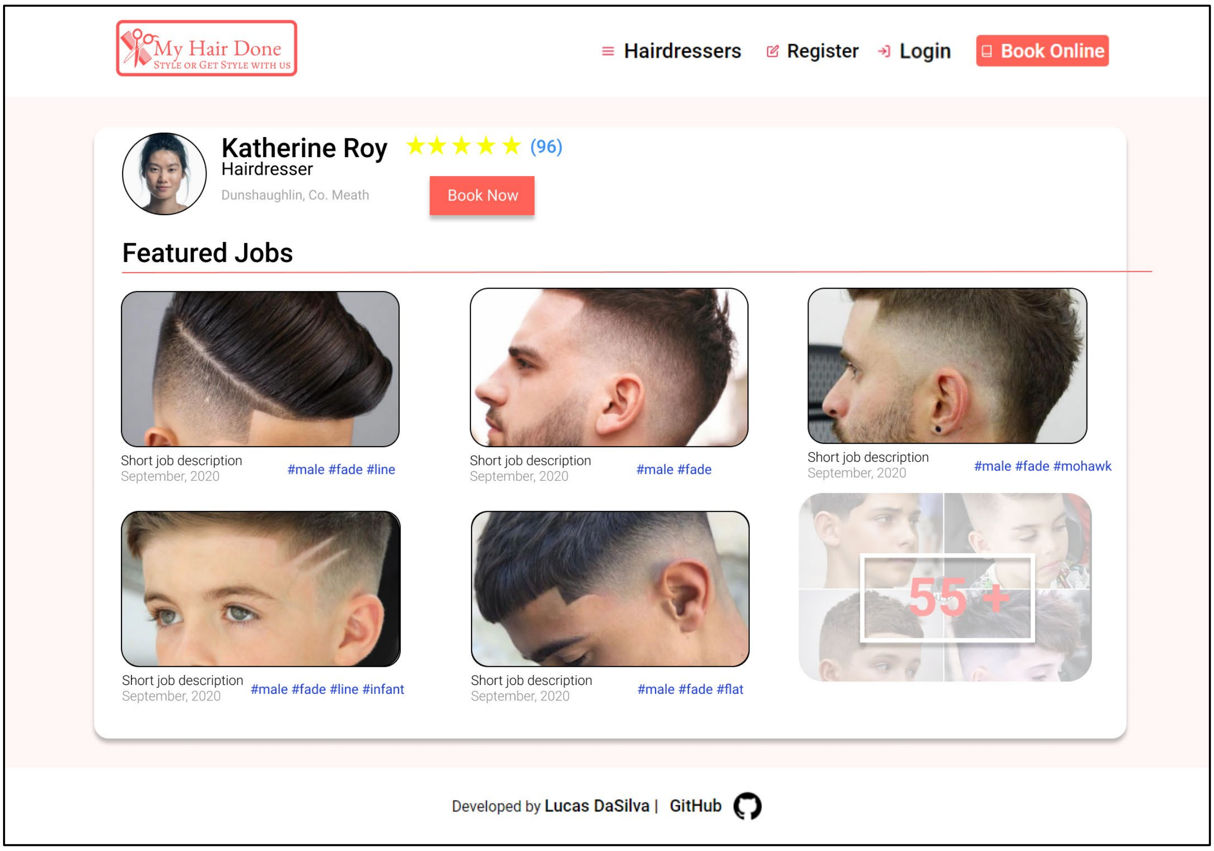
Kim Cheins
Barber
Dublin
12/02/2021 | 09:30

Katherine Roy
Hairdresser
Dunshaughlin, Co. Meath
03/11/2020 | 10:30

Richard Bray
Barber
Bray
18/06/2019 | 16:30

Joe Bloggs
Hairdresser
Cork
28/12/208 | 18:00

2.4.7. Hairdresser Portfolio



Katherine Roy ★★★★★ (96)
Hairdresser
Dunshaughlin, Co. Meath

Featured Jobs

Short job description
September, 2020
#male #fade #line

Short job description
September, 2020
#male #fade

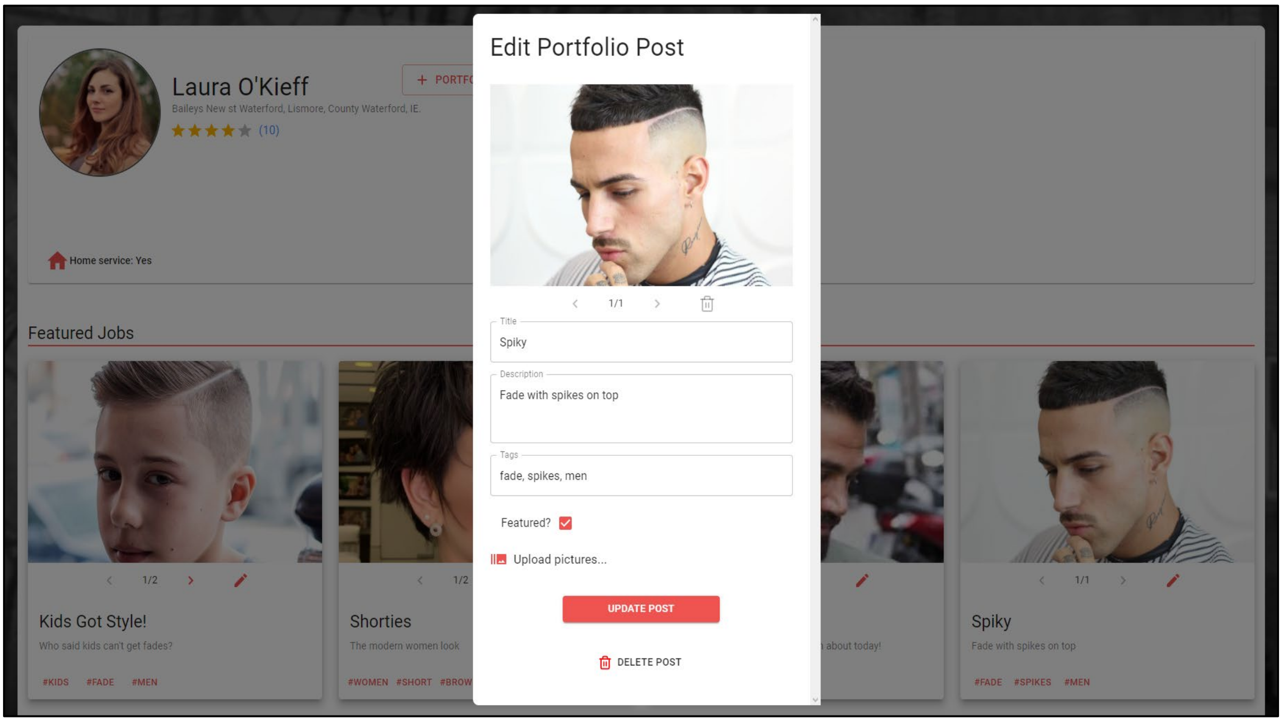
Short job description
September, 2020
#male #fade #mohawk

Short job description
September, 2020
#male #fade #line #infant

Short job description
September, 2020
#male #fade #flat

Developed by Lucas DaSilva | GitHub

2.4.8. Portfolio Post



Laura O'Kieff
Baileys New st Waterford, Lismore, County Waterford, IE.
★★★★☆ (10)
Home service: Yes

Featured Jobs

Kids Got Style!
Who said kids can't get fades?
#KIDS #FADE #MEN

Shorties
The modern women look
#WOMEN #SHORT #BROW

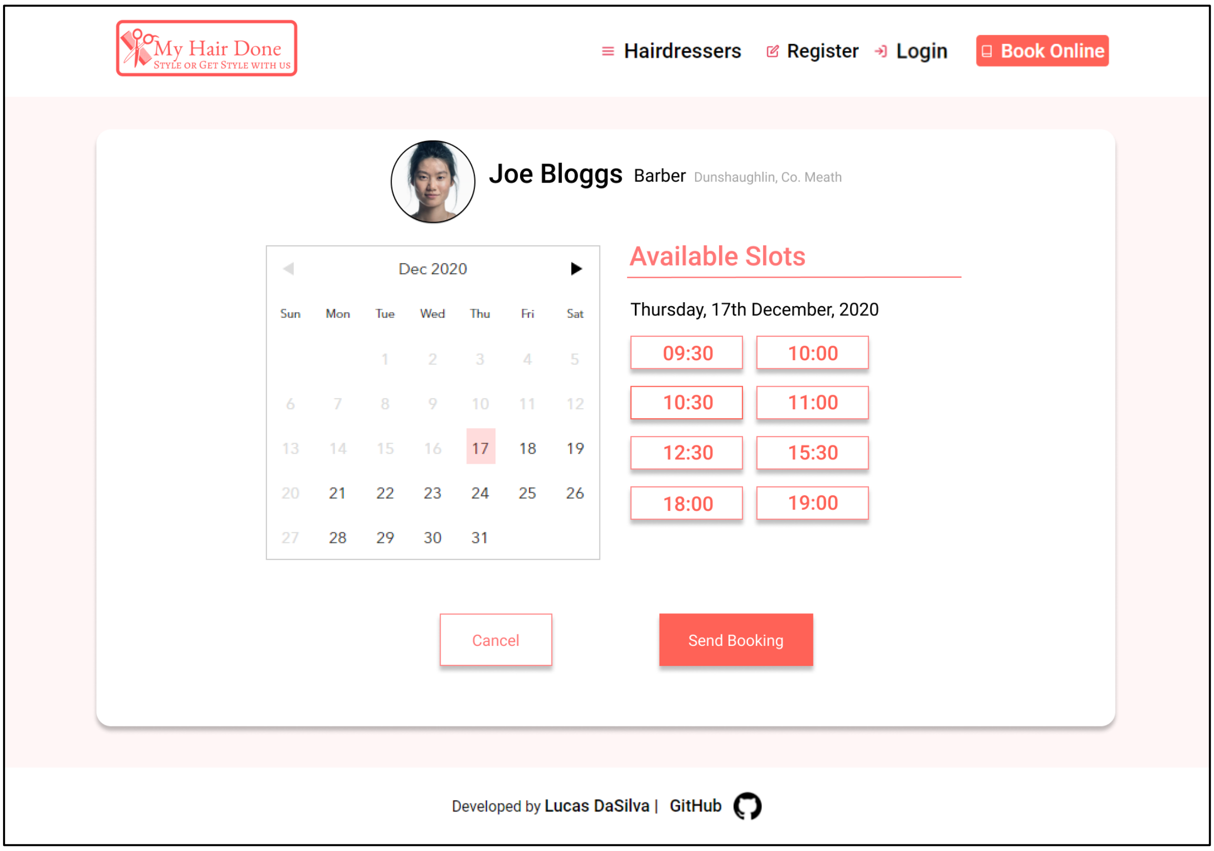
about today!

Spiky
Fade with spikes on top
#FADE #SPIKES #MEN

Edit Portfolio Post

Title: Spiky
Description: Fade with spikes on top
Tags: fade, spikes, men
Featured? ☑

2.4.9. Booking Page



Joe Bloggs Barber Dunshaughlin, Co. Meath

◀ Dec 2020 ▶

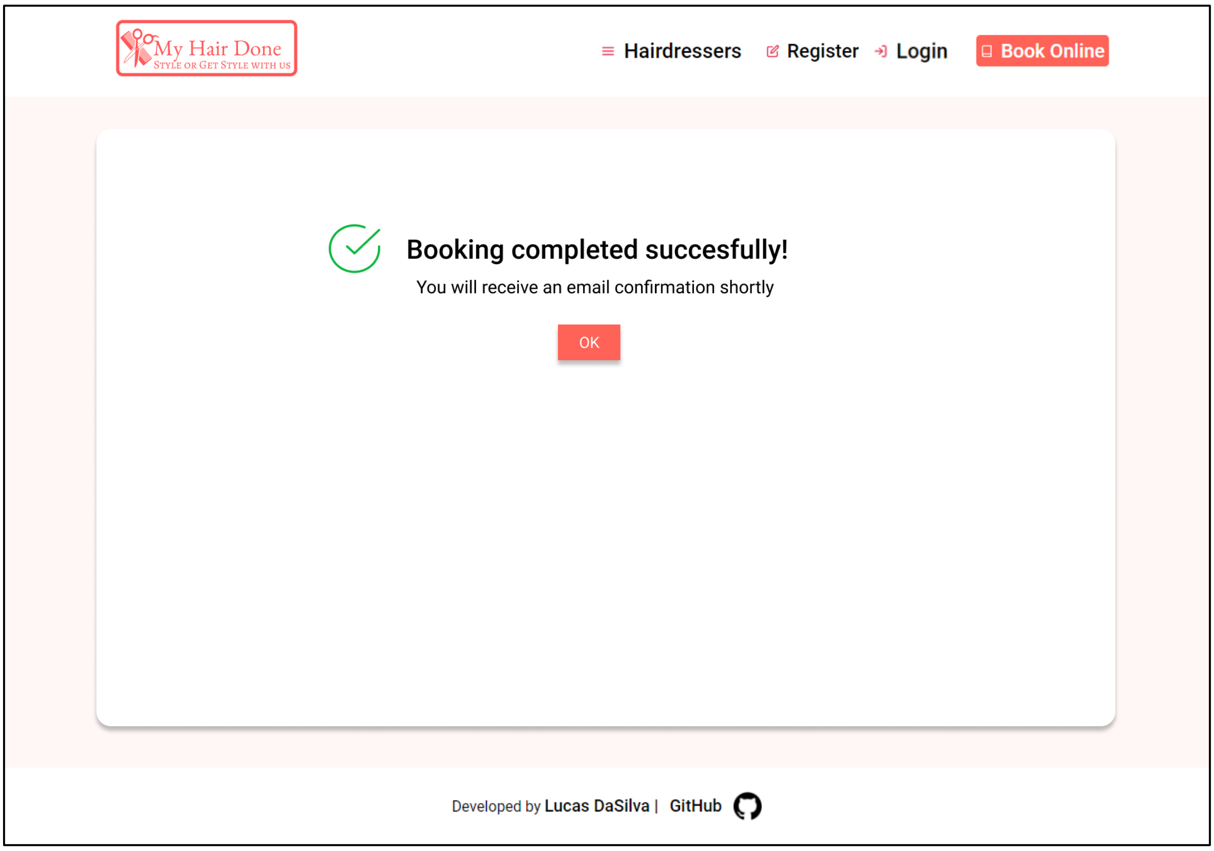
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Available Slots

Thursday, 17th December, 2020

Developed by **Lucas DaSilva** | **GitHub**

2.4.10.Booking Confirmation



Booking completed succesfully!

You will receive an email confirmation shortly

Developed by **Lucas DaSilva** | **GitHub**

3.0 Conclusions

My Hair Done is the type of application that has a very specific focus and tries to solve a specific problem, in the best way possible. Because of these goals and because this is an application that provides a service to end-users, the implementation and overall usability of the application must be very optimized succinct and designed in such a way that allows users to get through it as quickly as possible.

With this approach comes opportunities to make the positive aspects of the application shine, but at the same time imposes challenges that are natural these types of applications. Some of specific positive and negative points:

3.1. Positive Aspects

3.1.1. Highly Scalable Infrastructure

Coming from an infrastructure background, one of my main concerns for this application was to make the underlying infrastructure as robust and automated as possible in order to avoid users to experience any downtime and make the navigation as performant as possible. Using AWS IaaS (Infrastructure as a Service) I was able to achieve complete automation of the infrastructure provisioning using Terraform, an infrastructure as code scripting language that allows AWS components to be initialized automatically when one of the nodes serving the application fails or reaches 90% utilization.

3.1.2. DevOps

Since I am working with an IaaS service for the infrastructure of my application, it is important to have test, nodes provisioning, build and deployments all lined up correctly and automatically to allow the development cycle to not be impeded with long and repetitive deployments. Using GitHub and AWS CodeBuild, CodeDeploy, CodePipeline, I have a fully automated CI/CD pipeline in place which automatically deploys any test approved changes in my master branch to the nodes that serve both the backend and the frontend.

3.1.3. Cloud Native

As the IT industry progresses to develop more and more cloud native application, I wanted My Hair Done to use as many Cloud Native services as possible so that instead of reinventing the wheels on basic functionalities such as Mailing services, I could gain experience working with Cloud Services so that I can implement them on real-world applications in the future, while using the extra time to implement brand new features. AWS many useful services that in most time will be much superior than any custom implementation for trivial tasks such as mainlining, monitoring, logging and orchestration.

3.1.4. Secure

As all the components of the application infrastructure are hosted in AWS, it was important to make sure all AWS best practices were implemented to guarantee no networking vulnerabilities were exposed, specially for the Database, which is served by AWS RDS (Relational Database Service). On top of that, the web frontend is served completely using HTTPS Certificate which uses TLS 2.0 encryption standard, the same encryption standard used by international banks. All sensitive information is also hashed before being saved in application's database using a secret which is stored in a completely independent location and encrypted using AWS Secrets Manager.

3.1.5. Minimalistic UI

When developing the UI, my main goal which took precedence over everything else, was to develop an interface that would only display essential components in the page at any given time to avoid having a clunky UI at all costs. I think I have achieved that having filtered all non-essential information from all pages and making the essential information as attractive and easy to visualize as possible.

3.1.6. Responsive UI

Since My Hair Done would not have a mobile application at the first cycle of development, it was essential that the interface was completely responsive and attractive to use in mobile devices. The whole UI was developed with this in mind, using styling components such as cards and flexible grids to make the navigation in a mobile browser as close as possible to the experience of using a native mobile application.

3.2. Negative Aspects

3.2.1. Niche specific

Because My Hair Done is a service application with the focus in solving one specific problem for a very specific types of users, it can be hard to design and market the application to try and reach bigger mass of users. This is a bladed sword because it also allows the application to specialise in solving the problem it proposes to very well.

3.2.2. Hard to retain users for longer periods of time

Also, because the application was developed to be as succinct and effective as possible, there are no feature to retain the user for longer periods of time or to compel them to come back to the application before they need to come back to book their next hair appointment. the AR Hairstyler may help with this aspect since users can always come back to try new hair styles or share that with other users.

4.0 Further Development or Research

Working on this application exposed me to technologies and best practices in programming that I simply had not come across with which made the development process a lot more rewarding and challenging at the same time. Due to the very limited time, I could spend on this application, there are still a lot more enhancements to be implemented, from security measures, styling consistency to brand new features that would make My Hair Done a much more attractable application.

With that in mind, I am convinced that what I was able to implement in time for deliver is of high standard, especially considering my level of experience going into this project. I am especially proud of the infrastructure part, which is 100% cloud native and totally scalable.

The following are specific points that I will be working on over the next months to have fully implemented on the application as I want My Hair Done to be the main application in my portfolio:

Mobile App (Android/iOS) – Although My Hair Done started as a web application, I see the main platform for this application becoming the mobile app which is already in my plans to start the development. Since the backend is completely decoupled from the web frontend, I will be able to integrate a future application developed using React Native with the current backend to have both a web and mobile apps working concurrently.

Google Maps API – The idea is that users can use their own current location to find professionals within a given radius from them, initially this radius will be fixed in the backend to something like 20km for example, but I could potentially set this distance to be customized by the user when performing the search.

Real-Time Push Notifications – Since My Hair Done is mainly a web application for now, real-time notifications are not very useful for end users since they would need to have a browser tab open in the application to receive these notifications but going forward into the development of a mobile app, real time notifications will be a lot more useful and will be made part of the application features for sure.

Real-Time Chat – To refrain users and hairdressers from having to use their own mobile messaging services to communicate with each other, I will be implementing a real-time chat feature so that users can ask questions to hairdressers prior to booking or confirm any details after placing the booking.

Finalize the AR Hairstyler feature – Unfortunately, I was not able to complete the implementation of the AR Hairstyler due to time constraints. Having the face recognition successfully implemented was already a massive achievement to me since I had never worked with machine learning before but there is still a lot of room for improvement, which will be worked on the next development cycles for My Hair Done.

Implementation of Design Patterns – A not so much attractive feature to end users but one that can make the continuous development of this application is the use of Design Patterns in the code implementation. Throughout the development I tried to implement concepts such as SOLID and DDD to the best of my abilities, but I am conscious that a good amount of time could be spent refactoring some components of the frontend to make future implementations much easier.

My Hair Done has a great potential to become a very popular and useful application in the Irish Market since it focusses on a specific niche of users and tries to be as simple and effective as possible into what it proposes to do. Continuous development is needed, and optimizations are also needed like any other application in the market. I will keep working on this application and implementing new concepts and technologies as I come across with them to enhance the performance and user experience to the maximum.

5.0 Appendices

This section should contain information that is supplementary to the main body of the report.

5.1. Project Proposal

National College of Ireland

BSc (Honours) in Computing

Software Development

2020/2021

Lucas da Silva

X17118361

X17118361@student.ncirl.ie

My Hair Done

Project Proposal

Contents

Annotations

Professional – Hairdresser, Barber or hairstyling professional

Customer – user engaging with the professional

Objectives

General

The objective of this project is to build an application that Independent Hairdressers and Customers to find each other and structure the booking process making it a simple, intuitive, and consistent experience regardless of who the hairdresser or customer is.

For Hairdressers, the application will serve as a platform to publish their portfolio and style characteristics so that customers looking for a hairdresser specialized in a given style or location can find a match easily.

For customers, the application will help them on 3 different fronts: 1st The AI Hairstyle builder will help users see what they would look like with a chosen haircut style. 2nd The application will help them to find Hairdressers that match their hairstyle choice and finally, customers will be able to book time with their Hairdresser of choice.

Hairdresser Portfolio

When building their portfolio, the professional will be able to upload pictures of previous jobs and provide text descriptions and tags along with the picture. I plan to then use the tags and descriptions to match this professional with other customers looking for professionals who can provide them with similar haircut styles.

I would also like to make popular portfolio pictures available to all users in a "Popular Haircuts" tab in the application so that customers looking for a new style can browse through popular options and when they find it, they will immediately know who could be their next professional.

Bookings

The professional will be able to customize their available time slots calendar. Customers then can see what slots are available before booking a time with the professional via the application. When a time slot has already been booked by a different user, that same slot will NOT be available to other users.

Reviews

As part of the efforts to help independent Hairdressers to market their work, the Hairdresser profile within the application will display the average stars reviews the professional had in the past along with a dedicated tab where customers can browse through the past written reviews created by other users.

Hairstyle preview

This feature will allow customers to try new hairstyles by choosing available models and applying them to a picture from the customer's face. I don't know yet how this will be implemented specifically but I will need to use a specialized AI algorithm that fits the requirements which I am currently researching.

Background

The need for a solution that would simplify the booking process with a hairstyling professional is something I have felt for over 10 years now, since when I started to go to a professional more frequently and had to do it myself, instead of my parents, etc. I always felt that simply walking in a barbershop and wait until the barber is free is a big waste of time and a very inefficient way of managing my time.

Back in the neighbourhood, I grew up in, I remember that some barbers would take appointments over the phone, which already eliminated the need for walk-ins, but as a very shy and introspective teen, I never liked using the phone and would rather have other means book services.

When I moved to Ireland from Brazil, I was 18 years old and did not know anyone on the Island, didn't have any family or friends that could introduce me to the local services and part of my settling in process was naturally to find a barber to get my hair cut. Even though this may sound like a simple task, the cultural and language barriers were overwhelming enough for me to put off this task as much as possible and when I finally took the courage to go to the first barber in Ireland, I end up with a completely different hair cut than what I wanted, for three main reasons: 1st I didn't know how to explain exactly what hairstyle I wanted and 2nd my English was really bad at the time and 3rd I only realized later that the barber I went to wasn't experienced in doing the hair cut I wanted.

So, after a few years maturing the idea of simplifying this whole process, the smartphone phenomenon happened and with these incredibly popular devices, a few different applications came to existence that tackles everyday problems with elegant solutions. A Good example would be Just Eat or Deliveroo, which allows users to order food without leaving the house or even calling the restaurant (another dreaded process to me).

I took inspiration in these applications and my goal now is to make the experience of getting a haircut a process that can be managed straight from one simple application that eliminates the need for waiting in walk-ins and calling into barbershops looking for available slots.

To add value to the application and increase the chance of users spending more time using it, I decided to implement the "Hairstyle Preview" feature within the application. I expect that users go back to the app not only to book their next haircut but to also try new different styles which may serve as an additional push for them to book a time with a professional that is specialized in the haircut they chose.

Technical Approach

Requirements Specification

To capture this application's requirements, I will be using a UML Diagram where all the different Use Cases and possible actors will be identified. A comprehensive and detailed explanation of each use case will be then provided. I plan to include the UML diagram in this proposal before the final submission.

Project Management

Scrum and Kanban are the Agile Project Management methodologies that I will use during the development of this project.

Trello Board will be used to manage the bi-weekly sprints that I will be running until the end of this project.

Development Lifecycle

To develop the application, I plan to prioritize what feature gets developed first following the requirements specification document. Each use case then will follow a very similar development lifecycle which will roughly follow the following sequence:

User Interface

A Wireframe for all the pages I intend to have in the application will be built first before any code is written for the Web Frontend.

Backend

Once the Wireframe is complete, I will then move to develop that use case in the backend which includes the Database schema setup. As part of my CI/CD approach, the new version code is pushed to the GitHub repo, a new deployment will then be made so that this new version of my backend code with the new use case implemented is available in the production environment.

Frontend

With the use case functionality available in my backend API, I can now work on developing the user interface in the Web Frontend and see how it interacts with the backend. Once I'm happy with the Frontend implementation, the new frontend version is pushed to GitHub and a new deployment is performed to the Frontend Prod environment.

Version Control

GitHub will be used for the source code version control. Link to the repository created for this Software Project. https://github.com/lucasfdsilva/my-hair-done

An additional repository may be required to maintain the Frontend Code separate from the Backend to enable continuous deployment to take place more smoothly.

Special Resources Required

All resources required to complete this project are available online. I will be using a few different frameworks and libraries which are all free to use and have extensive documentation online.

For the application's infrastructure, Amazon AWS Services will be utilized extensively to serve all the different components of the application. The free tier offered by AWS should be enough to cover all the costs generated by the usage required during development and testing.

Technical Details

Frontend

The Web Frontend will be built using React which is a JavaScript Framework. I opted for using this framework because it is considered to be the standard choice for Frontend development these days and I want to experience how different it is to use a framework instead of plain HTML, CSS, and JavaScript.

React has its own “language”, JSX which is HTML written in JavaScript notation. The framework provides a boilerplate to help speed up the initial stages of development along with some very useful libraries that handle routing (React-Router) and deployment (Serve) very smoothly.

Backend

The backend will be built as a completely separate application using NodeJS and will communicate with the Frontend using HTTP Calls following the REST API architectural style.

I chose to develop the backend completely independent from the Frontend to allow the backend API to serve multiple frontends that I may develop in the future like an iOS/Android application to integrate with the same Backend and Database as the Web application, and this will be possible with no changes necessary on the backend.

The feature that allows users to try new hairstyles virtually will require a Machine Learning/AR library to be implemented as part of the backend. I don’t know yet what libraries are available for NodeJS and intend on investigating this further over the next month. If I’m not satisfied with the libraries I find, I may opt to deploy this feature as a micro-service developed in Java as I know that ML libraries are more popular in Java than in NodeJS.

I have also never implemented payments in any application before so I will have research in which NodeJS libraries can handle credit card payment.

Database

I plan to use MySQL as the engine for the main database.

As only the backend will interact with the database, I will probably use the Knex ORM (Object Relational Mapping) library for NodeJS as my query builder.

Infrastructure

As I would like to expand my knowledge in the Public Cloud Computing space, I plan to use a few different Amazon AWS services to support this application. The integration between these services are outstanding and I believe that for me to be ready for Cloud-Native Software development I need to be comfortable implementing these services into my applications:

- **Frontend**
  Will be served by Amazon Linux 2 EC2 instances. The instances will be part of an Auto Scaling Group to allow for scalability. A Load balancer will be responsible for balancing the load correctly across the instances live in the Auto Scaling Group.
- **Backend**
  Will be served by Amazon Linux 2 EC2 instances. The instances will be part of an Auto Scaling Group to allow for scalability. A Load balancer will be responsible for balancing the load correctly across the instances live in the Auto Scaling Group.
- **Postgres Database**
  AWS RDS service will be used to host the Postgres Database.
- **Storage**
  AWS S3 will be used to store pictures uploaded to the application
- **Mailing**
  AWS SES (Simple Email Service) will be used to support emailing out users when verifying email addresses, resetting the password, etc…
- **DNS**
  AWS Route53 DNS will be my DNS provider.

- **HTTPS Certificate**
  AWS Certificate Manager which natively integrates with AWS Route53 will be used to provide HTTPS certificate to my Web Frontend.
- **Secrets**
  Instead of storing secrets and API keys in the code or a .config file, I plan to use AWS Secrets Manager to store all my secret credentials safely which can be retrieved at runtime, drastically reducing the chances of these secrets being captured from my code.

CI/CD

To expedite how quickly I can test and deploy new versions of my frontend and backend, I'll implement a basic CI/CD pipeline using AWS DevOps tools like AWS CodeBuild and AWS CodeDeploy. These tools natively integrate with the rest of my infrastructure automating the build, test and deployment steps.

The application source code will be maintained in a public GitHub Repository which also integrates very well with the AWS DevOps suite.

Evaluation

Unit Testing

To perform these tests a professional account and a customer account have been provisioned.

Area 1: Mailing Service

- Can a customer or a professional reset their password?
  - Using the test professional account, at the login page, the "Forgot Password" link will be used.
  - The email address associated with the test professional account will be requested.
  - After input the email address, the application will display an error If the email address is not found.
  - If an email address found, and email with a password reset link will be sent to the email address registered with the professional account

Area 2: Professional/Customer Profile Management

- Are the permissions set correctly for both professionals and customers?
  - Customers can see professional's profiles but can't edit them.
  - Customers can't see other customer's profiles.
  - Professionals can see other professional's profiles but can't edit them.
  - Professionals can't see customer's profiles.
  - If these are successful, then this test passes.

- Is a customer able to add a review to the professional profile after a haircut?
  - A test appointment is booked between the test Professional account and the test Customer account.
  - After the date of the appointment is passed, the Customer can add a review regarding their experience with the professional.
  - If the customer can review the professional before the appointment time is past, this test fails.
  - If the customer is not able to review after the appointment time has passed, this test fails.

- If the customer can write the review only after the appointment time has passed, this test is successful.

Area 3: Data transfer to the Database

- When performing a CRUD operation to the Database, is the data being transferred securely?
  - A change in the test professional profile will be made.
  - Before sending it to the database; an interjection program will be used to view the data being sent to the database.
  - If the data sent by the CRUD operation is accessible to the interjection application, then this test failed.
  - If the data is not visible during the transfer, then this test is successful.

Area 4: Professional Portfolio

- Can a professional upload a new picture to their portfolio?
  - Using the test professional account, navigate to the portfolio tab
  - Click on the “Add Picture” button
  - The professional should be prompted with a File Explorer window to select the desired picture in either .jpg or .png format.
  - The file should then be uploaded and be visible in the professional’s portfolio.
  - If the image is available on the portfolio and no errors were prompted, then this test is successful.
  - If any errors were shown in the process, then this test failed.

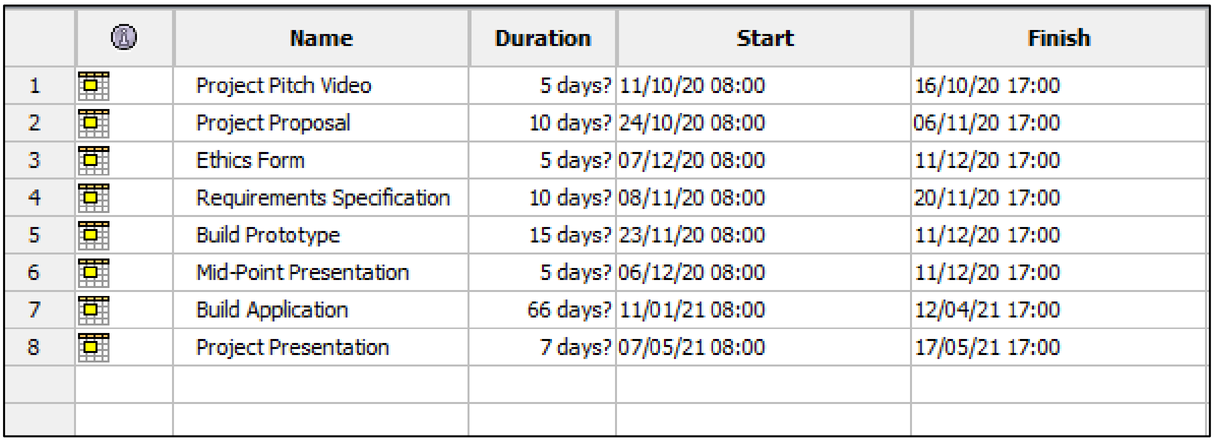
Load/Performance Testing

For the performance and load test, I will be using the “WebLOAD” tool to put both the frontend and backend infrastructure under high load.

The parameters that will be tested are:

- Check that the Database response time is considered appropriate (Under 3 seconds) for a scenario where under 1000 users are registered.
- Check what’s the maximum number of concurrent connections to the website before it’s completely overloaded.

5.2. Project Plan



		Name	Duration	Start	Finish
1		Project Pitch Video	5 days?	11/10/20 08:00	16/10/20 17:00
2		Project Proposal	10 days?	24/10/20 08:00	06/11/20 17:00
3		Ethics Form	5 days?	07/12/20 08:00	11/12/20 17:00
4		Requirements Specification	10 days?	08/11/20 08:00	20/11/20 17:00
5		Build Prototype	15 days?	23/11/20 08:00	11/12/20 17:00
6		Mid-Point Presentation	5 days?	06/12/20 08:00	11/12/20 17:00
7		Build Application	66 days?	11/01/21 08:00	12/04/21 17:00
8		Project Presentation	7 days?	07/05/21 08:00	17/05/21 17:00

5.3. Reflective Journals

5.3.1. Introduction

My name is Lucas da Silva, a Brazilian expat living in Ireland for 6 years at the time of writing. I am 26 years old married to Michely with no kids and a Beagle as a pet.

This is my 4th and final year at NCI undertaking the Bachelor of Science Degree in Computing (Part Time) specializing in Software Development.

I currently work full-time as a Cloud Engineer at Asystec Ltd. My role involves working with customers that are already using the cloud as an alternative for their on-prem IT infrastructure or are yet contemplating if the cloud could be the right thing for them. As an AWS Certified Developer, I help customers identify if their workloads are suitable for AWS and if so, I work with them to migrate and modernize their application in the cloud.

I am also a CloudHealth certified SA and work closely with a few customers helping them in their Multi-Cloud management.

The idea for my 4th year project has been with me for a while now as I explained in my project pitch which allowed me to mature it a bit more in terms of requirements and technologies I'd like to work with.

Writing this journal is also a strange experience to me, I have never needed to do this and honestly so far, it feels like a check box exercise which does not really add any value to my experience and knowledge while developing this project. It will be interesting to see if my perception changes before the end of the college year.

5.3.2. October 2020

October has been an extremely busy month for me and my family. After ups and downs we finally got the keys to our first house which had been our biggest dream for as long as I can remember. Managing the moving in process while juggling work and college was one of the hardest things, I have done but things are finally falling into place.

At work, after a long wait and different challenges I've officially been promoted to Cloud Engineer and now have the opportunity of helping customers on their cloud journey specialized in software development in AWS and Multi-Cloud management using CloudHealth by VMware.

I have been working in a large-scale project with our customer Boston Scientific in the deployment of CloudHealth in their AWS environment. This project was a massive undertake for the company, which makes me very happy for having this opportunity, but it also requires long hours of dedication which has been difficult to have this month.

In college, I feel that this year started a bit more intense than usual, obviously Covid-19 plays a role in it but also the assignments and modules feel more complex and require a lot more dedication than in previous years. It is incredibly hard not to feel overwhelmed and anxious, but I am trying to manage my expectations in relation to what I can realistically deliver.

I found that working on my project pitch was considerably easier than I was expecting, probably because I have explained this idea to so many people before that I got used to it.

One thing that is frustrating me is the fact that the project proposal deadline is a week from now and I have not received any feedback regarding my project pitch, which leaves me uncertain if my idea was well received and meet the requirements.

I also would like to have had the chance of speaking with my project coordinator, Paul Hayes, but NCI explained that the coordinator will only contact me when the project pitch feedback is concluded.

I expect that in November things will be calmer at home and at work, I will be able to get a lot more done in my college assignments and have a better routine to work consistently in the development of my project idea.

5.3.3. November 2020

In November things finally started to calm down in my personal life which allowed me to get much more work done in college.

The high number of assignments due in November pushed me away from my main project "My Hair Done" and by consequence I got much less completed on it that I wanted. But thankfully I managed to complete the infrastructure configuration along with securing the domain https://myhairdone.co.uk which is now live with a blank page on it.

I am now working on finalizing my CI/CD pipeline for this application. Having this pipeline in place will streamline so much the process of deploying to production the new code I write and make the experience of iterative deployment much more pleasant.

My focus for this month has been to catch up with the content I missed from the first few weeks of classes due to work promotion/house move. I am happy with my performance, I was able to submit all assignments several days ahead, while making progress in my main project while not missing classes or dropping my performance at work. If I can keep this to the end of the academic year, I will feel accomplished already.

5.3.4. December 2020

December has been an incredible month for me and definitely the busiest month I've had in years. With the last month of the college semester and of the calendar year, I had to put a lot of work on getting all my assignments completed and delivered in time while managing the increasing number of tasks and urgency in my full-time job.

December was also the month that my wife and I finally finished the basic setup of our house to allow us to have some comfort in the new place to be more relaxed to focus in work and studies. Of course, getting it all done took a huge amount of our free time since we're trying to do as much as possible ourselves to save money.

Even though there was a lot to deliver in this month, I feel I stepped up my productivity and was able to deliver all my assignments which included 6 different assignments and 2 TABAs while making significant progress in my software project.

My application expanded significantly this month and the development experience has been great, in great part due to the infrastructure setup I implemented in the first two months of development including the CI/CD pipeline.

The backend API at this stage has all the entity controls finalized which finally unlocks me to focus on developing the customer front-end, which is my weakest point, but I am confident that spending quality time learning new UX techniques and implementing those will grant me a quality product.

I am very anxious about implementing the AR aspect of my project though. I have never done something like this, and the workload of this semester has definitely impacted my confidence of spending the required time to learn it.

2020 was an incredibly difficult year for everyone, either from a personal or professional view everyone had to overcome a lot. Personally, this year was one of the most important for my personal and professional life which I am grateful for. It did not come without a lot of scars and traumas, I feel exhausted and in the edge of burning out, honestly, I still do not know how I am handling everything I have in my hands at the moment. I will make sure to get quality rest time in the 2 or 3 days of no work or college work I have at the end of this year.

As we are approaching 2021, I have a new strategy to get as much work done as possible as quickly as possible. I feel that I finally realized I should not seek perfection, but progress and applying this in the last month or so resulted in me getting more done than I thought possible so I hopeful that this strategy will make the next 5 months of development much easier and productive.

5.3.5. January 2021

I have accomplished a lot less in my final year project and in all other areas of my life in January than I thought I would. For the last 2 ½ months I have struggled with anxiety and a sense of being overwhelmed by literally everything around me, every little thing seems very tall, difficult, and exhausting to the point where thinking about doing any of the things I had planned out feels literally impossible. I am still capable of realizing what I must do but thinking about reaching for anything is very hard, I am not coping well.

5.3.6. February

Much like January, February has been an incredibly hard month to get through. It is difficult to understand exactly what is wrong with me by it feels like I will never be able to get through this stage of my life where everything seems to be impossible to survive. Even writing about it feels wrong and makes me feel bad for feeling bad for not obvious reason, no one I know if sick or unemployed, so feeling this way feels unjustified. I do not know what to do anymore.

My final semester in College just began, I remember feeling excited and happy at this time on previous years, but this is not the case anymore. All I can think about is being incredibly under pressure to deliver all the assignments that all be required this semester on top of working on my main project. It feels impossible.

Work is now busier than ever with more AWS migration projects and CloudHealth customer to onboard. Ever since I had to be more and more present in conversations with customers, work seems to be heavier and heavier. On top of that, in the past I was able to use a couple of hours a day from work to work on my college assignments, this is not the case anymore, which makes it even more difficult to get through the weeks.

I know that for these reflective journals I was supposed to write more about the development stages of my application, but at the moment I cannot get past this limbo I am in. Apologies for this.

5.3.7. March

March has been an incredible month to me, I finally took the initiative to seek help for how I was feeling and started to talk with a therapist in Brazil. Verbalizing how I feel and having a completely outsider and professional listening and giving insights has been extremely helpful and I am now training to focus more on the day-to-day tasks, without thinking too much about external factors I cannot control.

As a direct result in the month of March I was able to be a lot more pro-active with all college and work-related tasks. For the first time in almost an entire year, I feel that I am on top of my tasks and 100% conscious of what is to come.

I have already delivered part 1 of my usability design CA while already working on part 2. Usability Design seems to be an easy enough module but also very interesting. I wish we had more time to dedicate to each module.

Distributed systems seem to be the most daunting module this semester. Honestly, college resources on the subject have not been very helpful or even relevant to today's standards so I started a new course on Udemy to learn more about Protocol Call, specially gRPC. I am already working on my DS application and feel confident I will not struggle to deliver it on time.

Finally, I feel very disappointed with the Cloud App Development module. Since I am a Cloud Engineer, I was hoping this module would focus a lot more on the public cloud and its services, with actual development of cloud native applications that leverage the main benefits of the cloud. Instead, we are being asked to develop an application using Ruby on Rails, a really dated framework which cannot compete with other languages such as NodeJS and Python, specially when it comes to Cloud Software Development. I am not excited to work on the application that is being asked of us for the final CA.

Since my final project application has all the infrastructure and underlying components already built from last year, my strategy for this month is to focus on all other assignments I have to deliver, get all of them done before April comes so I can spend to whole last 90 days working 100% on My Hair Done to get it all done without any distractions of jumping on and out projects.

As always, the uncertainty if I am going to be able to coup with this strategy and be able to deliver it all makes me anxious and makes it all the harder to get through. Hopefully I can use this opportunity to learn how to work with my body and mind as much as possible, so in the future I don't need to feel this same way.

5.3.8. April

April has been an incredibly productive month for me on all fronts. I have been able to deliver all my Cas for all modules in college while maintaining an incredible level of productivity at work. This is obviously a direct result from the work I have been making with my Therapist.

After delivering all the Cas for the other module, I went straight onto working on My Hair Done, more specifically on the Frontend React as the Backend in NodeJS was mostly complete. It has been taking a huge amount of effort to get the basic styling to work as ReactJS compiles the CSS files altogether into one at run time, which makes the behaviour of some styling components very difficult to predict.

To combat that, I started to use an UI framework called “Material-UI”, which was developed by Google and provides bare bone components which have pre-made styling but are also very flexible to receive any additional styling, as necessary. I also started to use JavaScript styling to get around the problem of React compiling all CSS as one file.

Right now I’m working on the Augmented Reality feature that will allow users to try different haircuts, this is by far the most challenging component of the application and the one have the least experience with but I have been able to get the basic of facial recognition to work so now I need to optimize it and implement the capabilities of recognizing basic expressions before I can work on projecting a hairstyle on top of the user’s head.