National College of Ireland

BCs (Honours) in Computing - Evening

Software Development

2020/2021

Conor O Reilly

17101701

x17101701@student.ncirl.ie

Rare Disease Platform

Technical Report

Contents

Executive Summary

The following documentation pertains to the fourth year software development application entitled Rare Disease Platform (RDP). The main objective of this project was to provide a platform for users to become more knowledgeable about rare diseases. Users have the ability to sign in as a patient or healthcare professional and communicate and share information online. The application allows healthcare professionals the ability to share important information online in a blog post style, and patients can read these posts and comment on them. Furthermore, authenticated users can join a live group chat to discuss issues amongst each other. The application has a user friendly Graphical User Interface (GUI) that enables users to easily navigate around the application.

RDP is built on top of the Django Python framework that enables rapid development of secure and maintainable websites. Django is fast, fully loaded and secure, so was an excellent tool to use for developing my application. The application was developed using Visual Studio Code Integrated Developing Environment (IDE). Hypertext Mark-up Language (HTML), Cascading Style Sheets (CSS), and JavaScript were used to design the GUI. Python was used as the main programming language in the back-end. For my deployed version, I use the default SQLite database. I have deployed the application on a Heroku server.

https://rarediseaseplatform.herokuapp.com/

This application is important as it can provide users important information about rare diseases that they will not find easily on the internet. Furthermore, the information will be valid and creditable as it will be provided by healthcare professionals. The key benefit of this application is that it allows patients and healthcare professionals the ability to converse and share information that can potentially help them to overcome their condition.

1.0 Introduction

1.1. Background

During my internship in 3rd year, I spent 6 months as a System Administrator for a IT healthcare company, Open Applications. They provided patient registries, health data analytics, geo-spatial analytics and many more useful features. My experience in Open Applications provided me with an idea to create a platform which allows people to easily locate credible documentation from healthcare professionals and discover more by allowing them to engage with workers and other patients alike.

Millions of people suffer from rare diseases and may find it difficult to locate appropriate and reliable information. Rare diseases like Methemoglobinemia and Fibrodysplasia Ossificans Progressiva do not have an abundance of resources that allow patients and their family or friends learn about their disease. By allowing patients and healthcare professionals a channel to communicate and engage amongst each other could help a tremendous amount of patients around the world.

I also believe this application could benefit patients through these challenging times with Coronavirus becoming a part of our daily life's and restricting us to certain quarters. By allowing users the ability to engage with healthcare professionals online, instead of face-to-face, provides them an alternative way of learning more about their disease. Patients may not want to risk leaving their homes in case they contract the virus which could have damaging effect on their already existing health conditions. Providing a simple user interface to the user, they can easily navigate the application to find the information they need.

I am comfortable using Python within the Django framework environment. I have also created Python applications before that have certain features of the project I have outlined above. I hope to combine all these features together into one main platform that could prove to be immensely beneficial to millions of people around the world.

1.2. Aims

The purpose of this project was to create a functional Python application that provides users access to important information relating to rare diseases. The aim is to increase awareness about rare diseases and help as many patients suffering from such diseases. The applications aim is to serve as a guidance tool and reference point for patients hoping to understand their rare disease more.

The application is a user-friendly platform, which should provide an essential and beneficial service to the end user. The application will allow users to access information relating to rare diseases, which will be links to important information, images and more.

Once users are authenticated, they can create posts, categories and sub categories if they are a healthcare professional. The application has a front and back end. The front end shows all the posts relating to particular rare diseases. The back end acts as an Admin portal where healthcare professionals can create posts for all users to view.

Patients have the ability to read these posts and comment on them. All posts and comments must be verified by the administrator to ensure the information is correct and does not contain any explicit language.

The application will also allow authenticated users to communicate to each other in a live group chat as to talk amongst each other.

1.3. Technology

Front-End

CSS

Cascading Style Sheets will be used in most parts of my project for presentation and layout.

HTML

Similarly, HTML will be used to display information in my project.

Bootstrap

Bootstrap will also be used to style my application.

JavaScript

Over the past 6 months through my internship, I have completed multiple courses in relation to JavaScript. I hope to use these skills I have further developed to create a visual appealing design.

Back-End

Python

Similarly to JavaScript, I have completed multiple courses using Python. I hope to use these skills to develop my back-end for my application.

Django

Django is a high level Python web framework that I will use to secure and maintain my application.

Database

SQLite

My deployed version of the application uses the default SQLite database to store the information.

Host Provider

Heroku

I planned to host my application via Amazon Web Services (AWS). However, I encountered various problems and I could not deploy successfully with AWS. As a result, I had to deploy the application with Heroku and it was deployed successfully.

1.4. Structure

This document is separated into different sections. Below is a brief explanation of what each section covers:

- **Requirements**
    - This section covers Functional, Data, User, Environmental and Usability requirements of the application. Use case diagrams will be provided to help in the development of the system.
- **Design and Architecture**
    - This section includes information on how the system will operate at a high level overview. I will explain why it was chosen and provide diagrams of implementation.
- **Graphical User Interface**
    - This section will provide screenshots of some of the more important pages with the application and explanations behind their implementation. For the moment, this section will primarily be design prototypes.
- **Testing**
    - This section will discuss the different types of testing undertaken during the duration of developing the application. Unit testing, User Interface (UI) tests and usability tests will be conducing to test the application. Selenium will be the primary tool to accomplish this.

2.0 System

2.1. Requirements

2.1.1. Functional Requirements

2.1.1.1. Use Case Diagram

This Use Case Diagram provides an overview of all functional requirements.

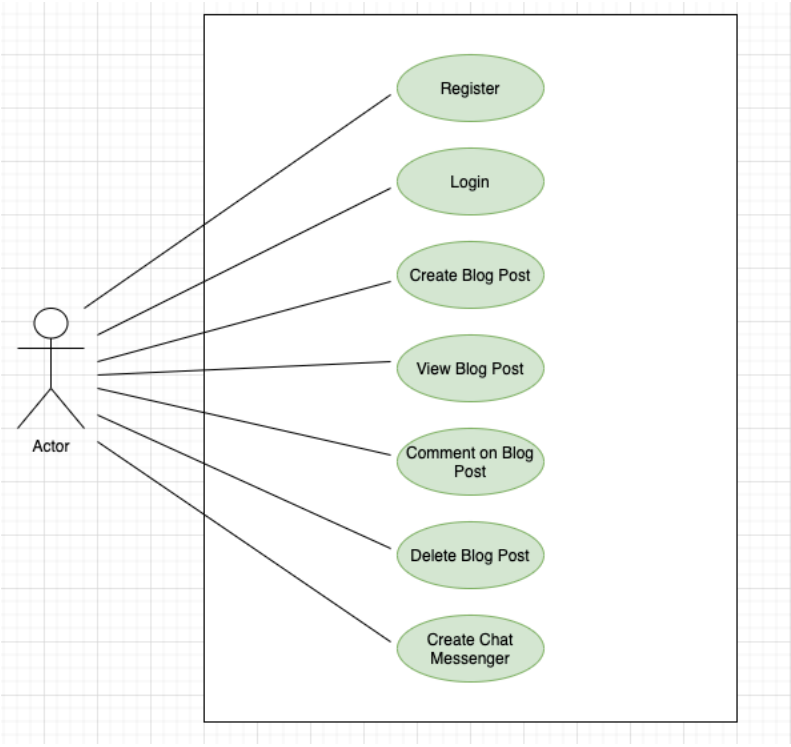


Figure 1: Rare Disease Platform Use Case Diagram

2.1.1.2. Requirement 1 <Registration>

2.1.1.2.1. Description & Priority

This use case describes how the user is able to register an account for the application by providing a unique email and password that will be stored in the database. There are two types of users for this application: patient and healthcare professional. All users must enter a valid email and password to sign in. Access is granted once the administrator has verified the account and allotted the user the correct permissions. This use case is essential as a user needs a to create an account to have access to the applications features.

2.1.1.2.1.1 Use Case

Scope

The scope of this use case is to register an account to allow access to the application.

Description

This use case describes the process of a user registering an account.

Use Case Diagram

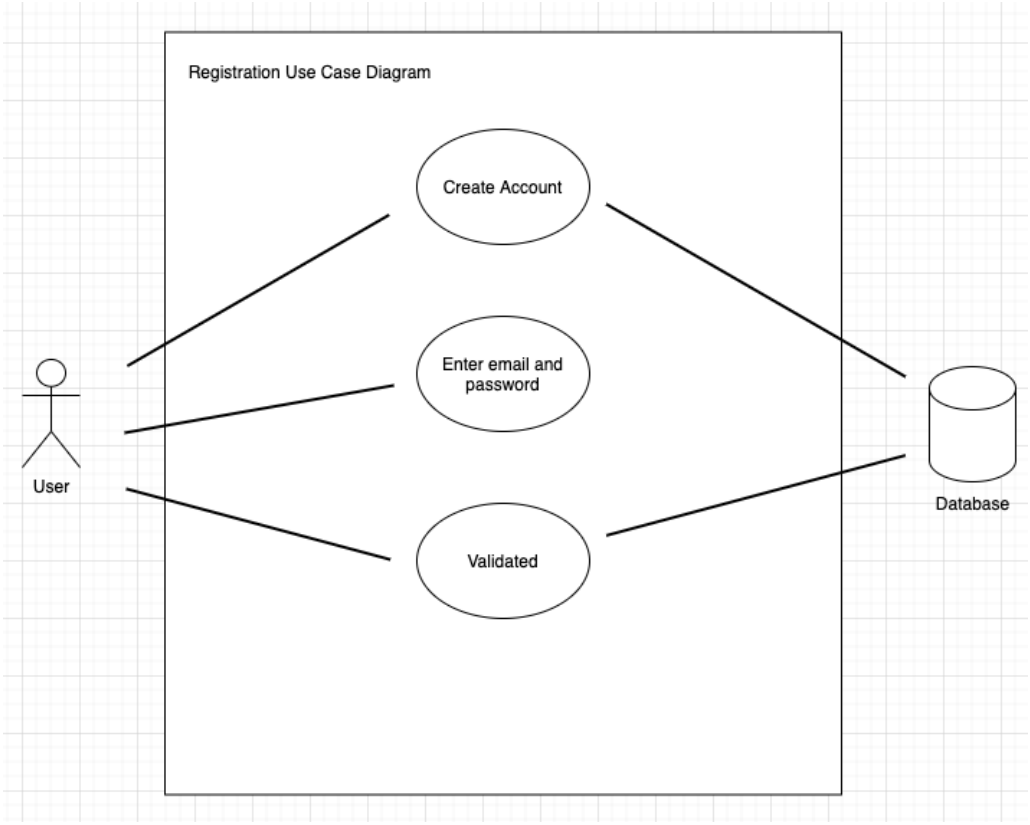


Figure 2: Registration Use Case Diagram

Flow Description

Precondition

An internet connection is required to register a new account.

Activation

This use case starts when a user selects the Registration button.

Main flow

1. The user accesses the registration page
2. The user enters an email and password for their account (See A1)
3. The Database checks the user's credentials and verifies the user (See E1)
4. The system grants the user access and sets up their account

Alternate flow

A1 : <User enters an invalid email or password>

1. The user enters an invalid email address or a weak password
2. The user enters a correct email address and a stronger password
3. The use case continues at position 3 of the main flow

Exceptional flow

E1 : <User already exists>

1. The user enters existing credentials for an already existing account

2. The system responds with an error message stating that the account already exists
3. The use case continues at position 2 of the main flow

Termination

The system presents the user with the landing page of the application.

Post condition

The system stores the users credentials in a database if the credentials are correct.

2.1.1.3. Requirement 2 <Login>

2.1.1.3.1. Description and Priority

This use case describes how the user can login into their already existing account by providing their unique login credentials. When an account has been created, the user shall be able to login into their account. The user will enter their credentials that will be compared to credentials stored in the database. Access is granted upon successful comparison of the users credentials. This use case is essential as a user needs a to create an account to have access to the applications features.

2.1.1.2.1.2 Use Case

Scope

The scope of this use case is to allow the user to sign into their account to access the application.

Description

This use case describes the process of a user signing into the application.

Use Case

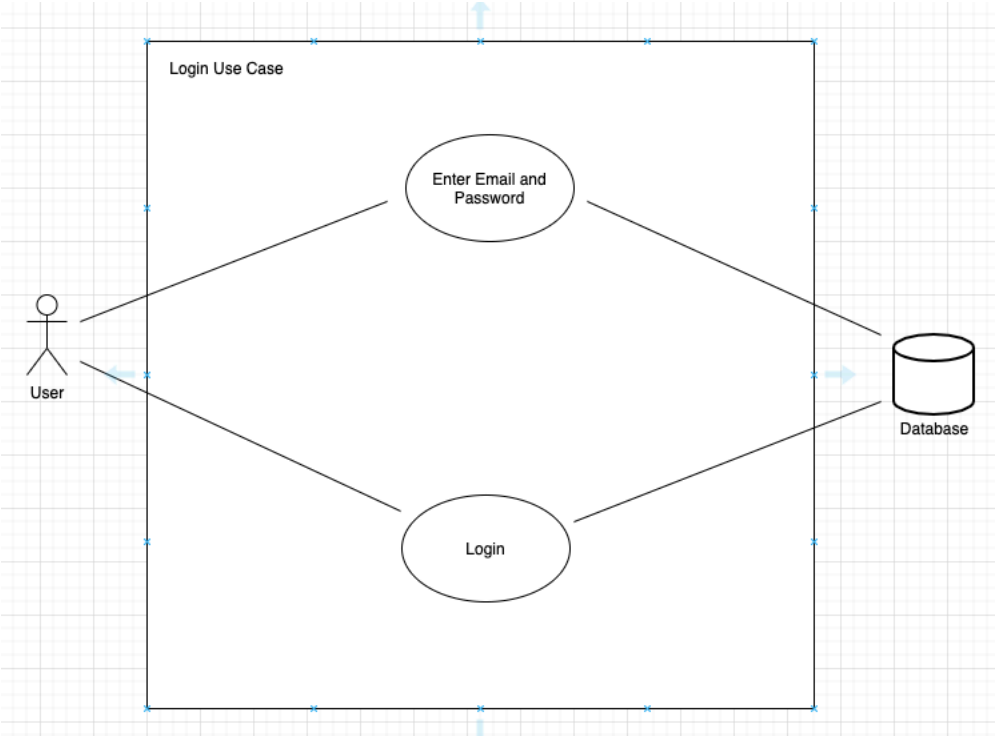


Figure 3: Login Use Case Diagram

Flow Description

Pre-Condition

An internet connection is required to sign into an account.

Activation

This use case starts when a user selects the Login button.

Main flow

1. The user has a registered account (See A1)
2. The user enters an email and password for their account (See E1)
3. The Database checks the user's credentials and verifies the user
4. The system grants the user access and logs them into their account

Alternate flow

A1 : <User has not registered for an account>

1. The user is not a registered user
2. The user navigates to the registration page and creates an account
3. The use case continues at position 1 of the main flow

Exceptional flow

E1 : <User enters invalid credentials>

1. The user enters invalid credentials
2. The system displays a message stating that the credentials are incorrect
3. The use case continues at position 2 of the main flow

Termination

The system presents the user with the landing page of the application.

Post condition

The system updates the users credentials in a database if the credentials are correct.

2.1.1.4. Requirement 3 <Logout>

2.1.1.4.1. Description and Priority

This use case describes how the user can logout of their already existing account. If an account has been successfully created, the user will have the option to sign out of their account. The users account details are stored in the database. A successfully logout is granted if the user is an authenticated user. This use case is of medium to high priority.

2.1.1.2.1.3 Use Case

Scope

The scope of this use case is to allow the user to sign out of their account to exit the application.

Description

This use case describes the process of a user signing out of the application.

Use Case

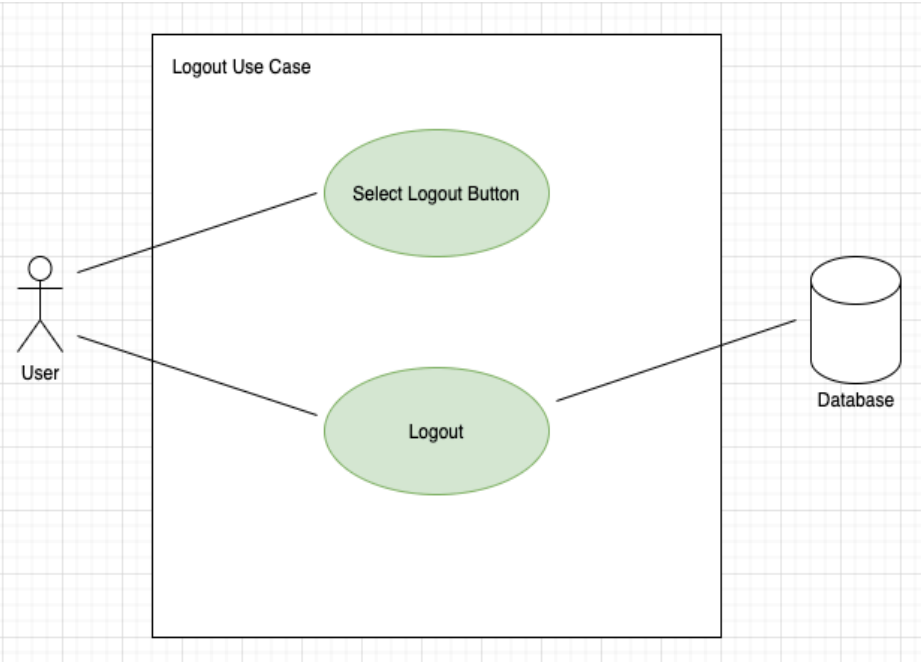


Figure 4: Logout Use Case Diagram

Flow Description

Pre-Condition

An internet connection is required to sign into an account.

Activation

This use case starts when a user selects the Logout button.

Main flow

1. The user is a Patient/Healthcare professional
2. The user is already logged into their account (See A1)
3. The user selects the sign out button
4. The database checks for the users' credentials
5. The system signs the user out of the application
6. The user is redirected back to the home page

Alternate flow

A1 : <User is not signed into their account>

1. The user is not signed into their account
2. The use case continues in position 6 of the main flow

Exceptional flow

-

Termination

The system presents the user with the landing page of the application.

Post condition

The system checks that the user has successfully signed out.

2.1.1.5. Requirement 4 <Create a Category>

2.1.1.5.1. Description and Priority

This use case describes how the user create a category for a particular rare disease. A category can be created by supplying a name. The category can be used to reference a particular blog post. This use case is medium to high priority.

2.1.1.2.1.4 Use Case

Scope

The scope of this use case is to allow the user to create category.

Description

This use case describes the process of creating a category.

Use Case

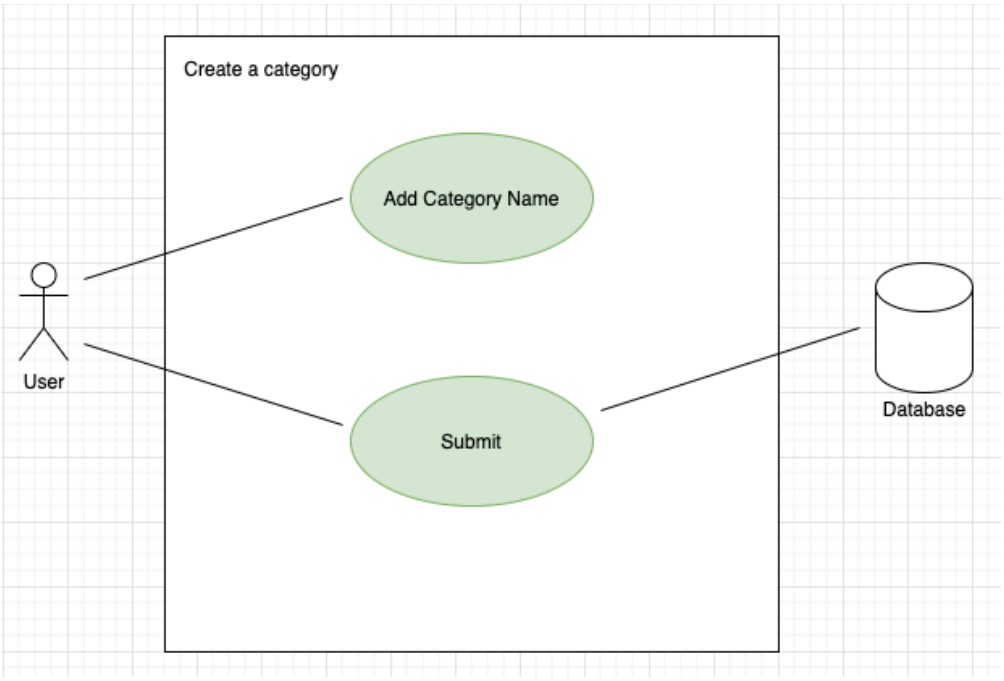


Figure 5: Create a Category Use Case Diagram

Flow Description

Pre-Condition

An internet connection is required to create a blog post.

Activation

This use case starts when a user selects the Add Category button.

Main flow

1. The user is a Healthcare professional
2. The user adds a rare disease name (See A1)
3. The system stores the information in the database
4. The system verifies the category and shares it on the application

Alternate flow

A1 : <User doesn't enter all required fields>

1. The user does not enter information in the name field
2. The system displays a message stating that the user must enter these fields
3. The user adds a rare disease name (See E1)
4. The use case continues at position 2 of the main flow

Exceptional flow

E1 : <Category name has already been used>

1. The user has entered a category that already exist
2. The system shows a failed error page
3. The user is redirected to the Add Category page
4. The user case continues at position 2 of the main flow

Termination

The system presents the newly created category in the appropriate location.

Post condition

The system stores the new category in the database.

2.1.1.6. Requirement 5 <Create a blog style post>

2.1.1.6.1. Description and Priority

This use case describes how the user can create a blog style post in the application for users to view and comment. A blog post can be created by supplying a header, body text, and any other fields they required. This use case is medium to high priority.

2.1.1.2.1.5 Use Case

Scope

The scope of this use case is to allow the user to create a blog post and share it to all users who may be interested.

Description

This use case describes the process of creating a blog post.

Use Case

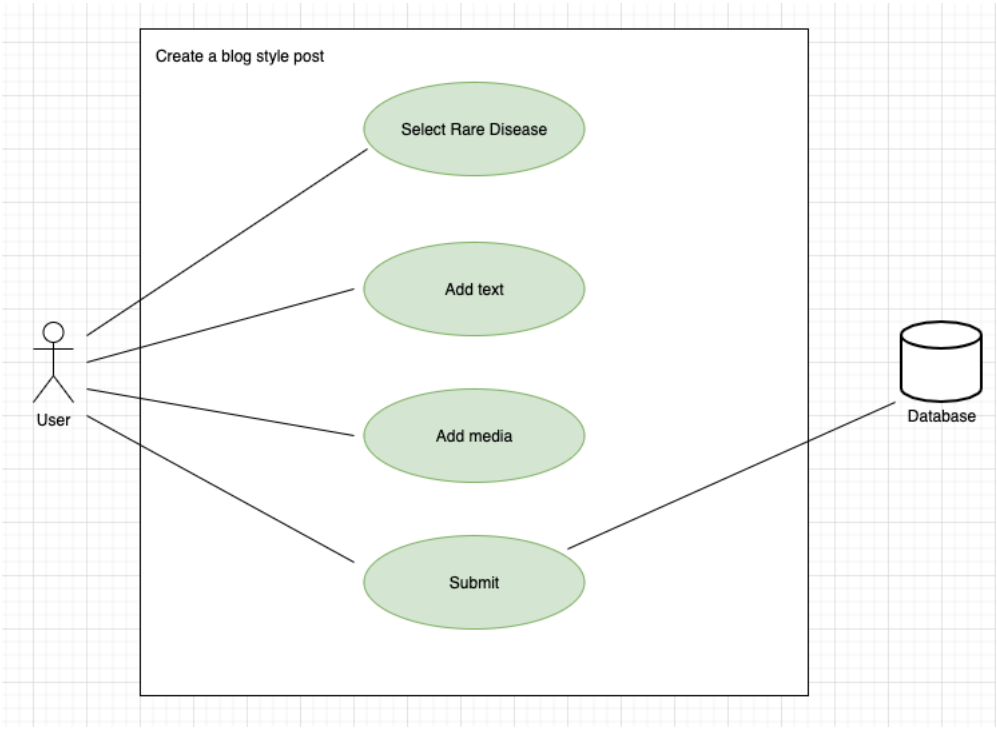


Figure 6: Create blog post Use Case Diagram

Flow Description

Pre-Condition

An internet connection is required to create a blog post.

Activation

This use case starts when a user selects the Create a Post button.

Main flow

1. The user is a Healthcare professional
2. The user adds a header, body text and any other fields (See A1)
3. The system stores the information in the database
4. The system verifies the post and shares it on the application

Alternate flow

A1 : <User doesn't enter all required fields>

1. The user does not enter information in the header or body text fields

2. The system displays a message stating that the user must enter these fields
3. The use case continues at position 2 of the main flow

Exceptional flow

E1 : <Database does not save blog post>

1. The database fails to save blogpost
2. The system shows a failed error page
3. The user is redirected to the Add Post page
4. The user case continues at position 2 of the main flow

Termination

The system presents the newly created blog post in the appropriate location.

Post condition

The system stores the new blog post in the database.

2.1.1.7. Requirement 6 <View a blog style post>

2.1.1.7.1. Description and Priority

This use case describes how the user view an existing blog post. The user can read about the contents of the post, view the comments, and exit the view. This use case is of medium priority.

2.1.1.2.1.6 Use Case

Scope

The scope of this use case is to allow the user to view blog posts that have been created.

Description

This use case describes the process of viewing a blog post.

Use Case

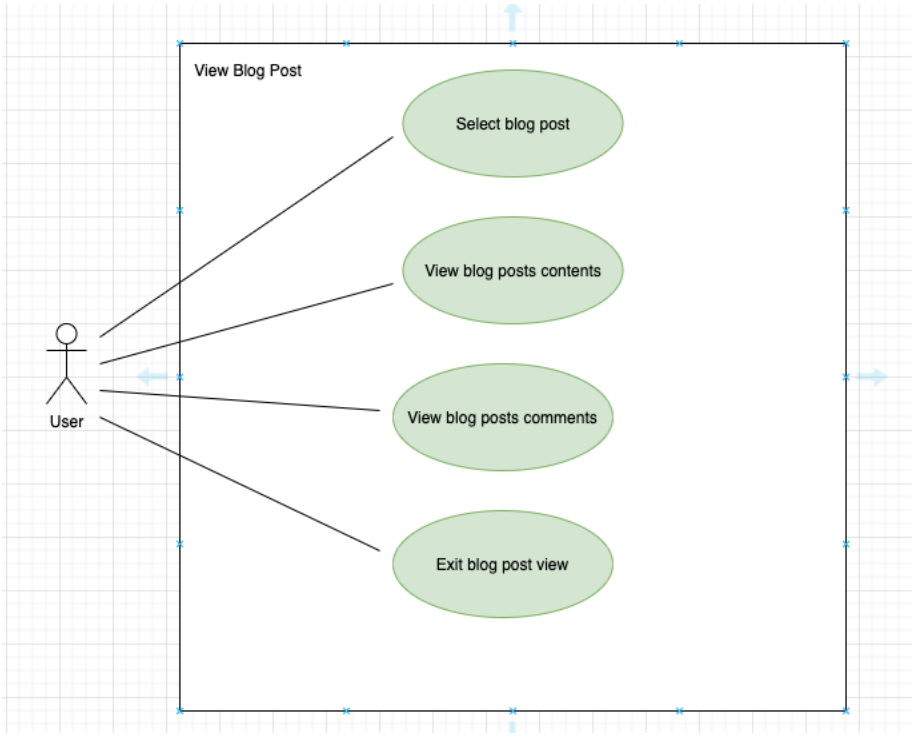


Figure 7: View Blog Post Use Case Diagram

Flow Description

Pre-Condition

An internet connection is required to view a blog post.

Activation

This use case starts when a user selects a desired post to read.

Main flow

1. The user selects a blog post (See A1)
2. The user reads the body of the post
3. The user views any media associated with the post
4. The user exits the post and returns to the home page

Alternate flow

A1 : <Blogpost has been deleted>

1. The user is redirected to the home page as post has been removed
2. The use case continues at position 1 of the main flow

Exceptional flow

-

Termination

The system presents the user back to the home page.

Post condition

The system updates any potential info regarding view count to the database.

2.1.1.8. Requirement 7 <Comment on a blog style post>

2.1.1.8.1. Description and Priority

This use case describes how any user can comment on an existing blog post. The user can post a comment on any post they wish. This use case is of low priority.

2.1.1.2.1.7 Use Case

Scope

The scope of this use case is to allow the user to comment on a blog post that has been created.

Description

This use case describes the process of commenting on a blog post.

Use Case

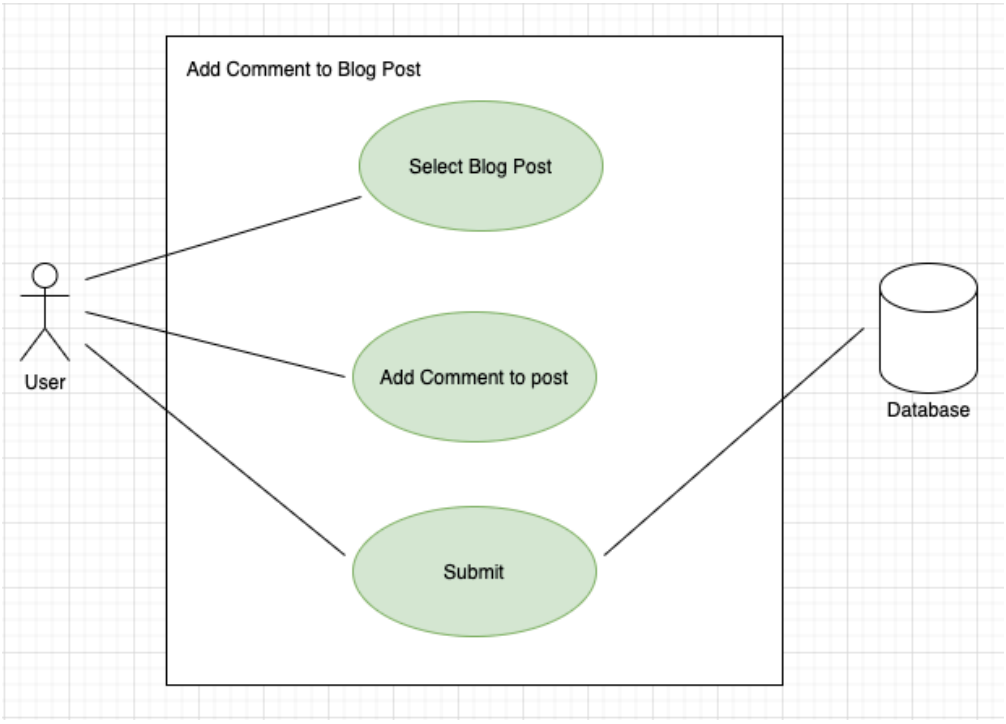


Figure 8: View Blog Post Use Case Diagram

Flow Description

Pre-Condition

An internet connection is required to view a blog post.

Activation

This use case starts when a user selects a desired post to comment on.

Main flow

1. The user selects a blog post
2. The user reads the body of the post
3. The user comments on the post (See A1)
4. The user exits the post and returns to the home page

Alternate flow

A1: <Administrator must validate comment>

1. The user's comment must be verified by the administrator
2. Once verified, the comment can be viewed

Exceptional flow

-

Termination

The system presents the user back to the home page.

Post condition

The system stores the new comment in the database.

2.1.1.9. Requirement 8 <Delete a blog style post>

2.1.1.9.1. Description and Priority

This use case describes how the user deletes an existing blog post. The user can delete the blog post if they created the post. This use case is of medium priority.

2.1.1.2.1.8 Use Case

Scope

The scope of this use case is to allow the user to delete blog posts that have been created.

Description

This use case describes the process of deleting a blog post they created.

Use Case

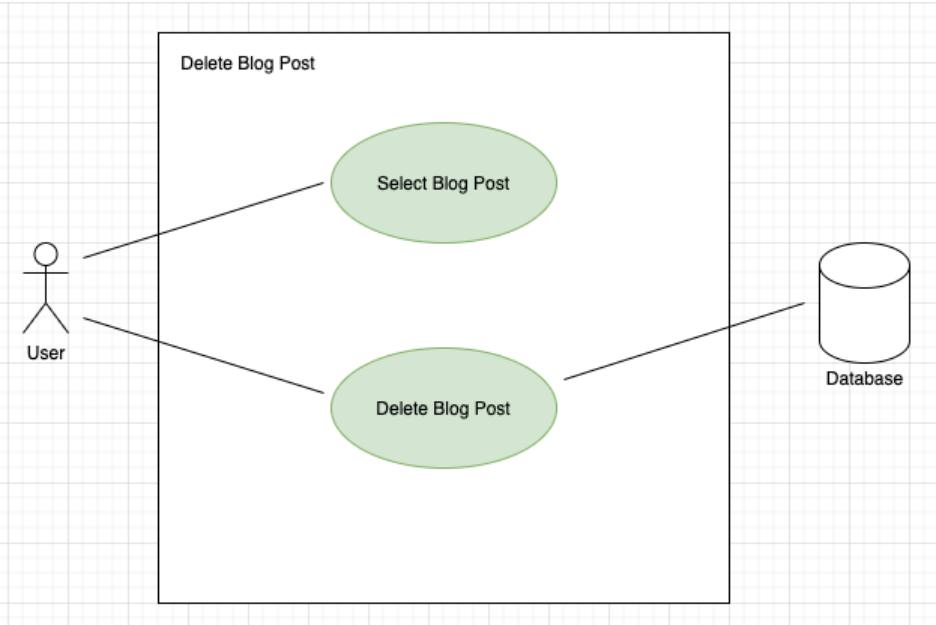


Figure 9: Delete Blog Post Use Case Diagram

Flow Description

Pre-Condition

An internet connection is required to view a blog post.

Activation

This use case starts when a user selects a desired post to delete.

Main flow

1. The user selects a blog post
2. The user selects the delete button
3. The blog post is deleted (See A1)
4. The user is redirected to the Post List page

Alternate flow

1. The database fails to successfully delete the blog post
2. The system shows a failed error page
3. The user is redirected to the Add Post page

Exceptional flow

-

Termination

The system presents the user back to the Post List page.

Post condition

The system deletes the blog post from the database.

2.1.2 Data Requirements

This section will describe the data requirements, which are pivotal to implementing the key functionalities outlined in this document.

- User email and password – An email and password will be required for a patient to register/sign into their account. The information will be stored in the database.
- Blog posts – Blog posts will be store in the database and will require a header, body text and any additional media.

2.1.3 User Requirements

This will describe a reasonable description of what the users of this application will expect if successful.

- The user will be required to have access to a web browser with an internet connection to reach the site.
- The user is also required to enter valid email and password information to access the application. This is important as it will allow the user access to all the applications features.
- The healthcare professional should be able to create, delete and modify blog posts with ease.
- Users should be able to comment on each blog post.
- Navigate to various blog posts on the platform and view comments, add comments and delete comments.
- Users should be able to participate in a group chat with other users to communicate.
- Search for a post based on the user's input.
- Filter content with defined metrics set out by user.

2.1.4 Environmental Requirements

This section will provide the vital requirements needed when developing the application.

- Windows/Apple/Linux Machine: This application will be developed using an Apple machine with Visual Studio Code as the development IDE. However, any Operating System (OS) will be able to develop this application.
- Internet Access: Internet access will be required during the running and testing of this application in the development stages.

2.1.5 Usability Requirements

The application will be designed in a way to make it easy and efficient for the user to view, add, update, and delete posts and comments. The user should be able to view the desired

content with the minimum amount of offered required. The user is expected to know how to register and sign up for a new account and be able to sign into their account once created.

2.2 Design & Architecture

This application was built using a Python framework called Django. Django Views are custom Python code that get executed when a certain URL is accessed. The views represent different pages on the website and they all extend from the Home Page. As this app is a Web Application, it will work on all OS devices.

A number of other programming languages were used to implement various features of the application. HTML, CSS, Bootstrap and JavaScript will be used to complete the front-end of the application. These languages will help the website to gain a sleek and well finished design.

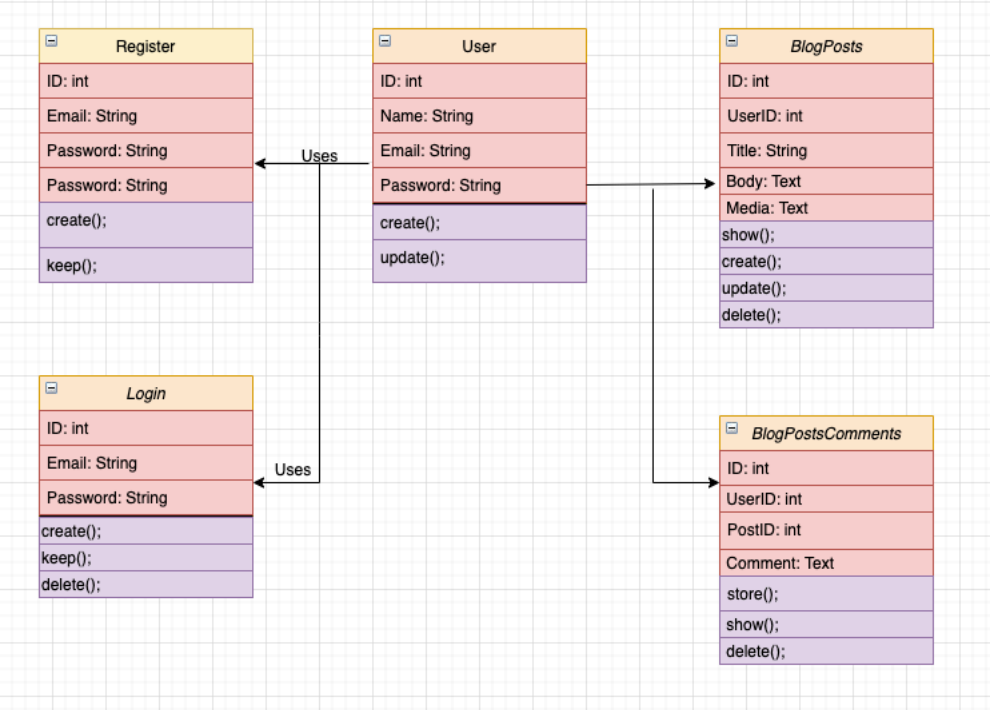


Figure 10: Class Diagram

The system architecture depicted represents all the various variables in the database. For example, the methods of the BlogPosts class describe each feature associated with BlogPosts.

Create(); directs the to the view to register a new user or create a new blog post.

Keep(); stores a new user or blog post in the database.

Delete(); simply deletes a user or blog post from the database.

Update(); updates an existing blog post in the database.

Show(); shows the desired blog post or comment.

2.3 Implementation

Setting up Virtual Environment and Django

Before setting up my Django framework, I set up virtual environment to store my application. I entered the below command to set up a virtual environment:

```
$ python3 -m venv myvenv
```

Next, I activated my virtual environment by entering the below command:

```
$ source myenv/bin/activate
```

Once activated, I installed Django in my application. This is achieved by adding the below command:

```
$ pip install django
```

Once Django is installed, I can create a project.

```
$ django-admin startproject rdpproject
```

I then made changes to my settings.py file to run my project. I was then able to see if my Django project was running by entering the below command:

```
$ python manage.py runserver
```

As a result, I am able to view the application and receive a success message stating that I have successfully ran the application. From here, I can start to add different pages to the application and make all the necessary changes I have listed below.

URLs

A urls.py file was created for each app in my project. Below is the code for my main app that highlights all the URL paths for each of my file paths.

```
urlpatterns = [
    url(r'^$', views.home, name='home'),
    url(r'^about/$', views.about, name='about'),
    url(r'^panel/$', views.panel, name='panel'),
    url(r'^login/$', views.mylogin, name='mylogin'),
    url(r'^logout/$', views.mylogout, name='mylogout'),
    url(r'^panel/setting/$', views.site_setting, name='site_setting'),
    url(r'^panel/about/setting/$', views.about_setting, name='about_setting'),
    url(r'^contact/$', views.contact, name='contact'),
    url(r'^panel/change/pass/$', views.change_pass, name='change_pass'),
    url(r'^register/$', views.myregister, name='myregister'),
    url(r'^answer/comments/(?P<pk>\d+)/$', views.answer_contactform, name='answer_contactform'),
]
```

Figure 11: urls.py file

This is important as it defines the mapping between the URLs and the views of my project. A urls.py file has been implemented in each individual app, i.e. posts, category. Unit testing was conducted on each URL and all successfully passed. A deeper analysis can be found in the Testing section below.

Models

A models.py file is created in each app by Django. The models.py file is a class that represents a table in our database. Every attribute of the class is a field of the database table. Each app has a models.py file that contains all the necessary fields for that app. Below is the code for the models.py class in the posts app.

```
class Posts(models.Model):

    name = models.CharField(max_length=250)
    short_txt = models.TextField()
    body_txt = models.TextField()
    date = models.CharField(max_length=12)
    time = models.CharField(max_length=12, default = "00:00")
    img = models.TextField()
    imgurl = models.TextField(default="-")
    author = models.CharField(max_length=50)
    catname = models.CharField(max_length=50, default='-')
    catid = models.IntegerField(default=0)
    ocatid = models.IntegerField(default=0)
    views = models.IntegerField(default=0)
    tag = models.TextField(default='')
    act = models.IntegerField(default=0)
    rand = models.IntegerField(default=0)

    def __str__(self):

        return self.name
```

Figure 12: models.py in posts app

As can be seen above, 15 attributes have been defined for this app. Each attributes has a different type, i.e. CharField, TextField etc. This information is stored in the database and can be called by the system. All models.py files can be found in each app folder.

Registration

The myregister() method successfully registers a user to the application and stores their information in the database. The method ensures that all necessary fields have been added. A simple algorithm has also been implemented to ensure the users password is strong and contains different characters.

```
# Check strenth of new password
count1 = 0
count2 = 0
count3 = 0
count4 = 0

for i in pass1 :

    if i > "0" and i < "9" :
        count1 = 1
    if i > "A" and i < "Z" :
        count2 = 1
    if i > 'a' and i < 'z' :
        count3 = 1
    if i > "!" and i < "@" :
        count4 = 1

if count1 == 0 or count2 == 0 or count3 == 0 or count4 == 0:
    msg = "Your Passwords is Not Strong Enough"
    return render(request, 'front/message.html', {'msg': msg})
```

Figure 13: views.py file in main app

The method also checks if the user has entered the minimum amount of characters permitted for a password. Furthermore, if all the criteria mentioned above is met and the information provided is not already in the database, then the system will create the new user and store the information in the database.

Once complete, the administrator of the system will check the users credentials before accessing them permission to the site. Once complete, the user will be assigned the role of a healthcare worker or patient depending on their status.

Login

The user must enter their username and password to successfully log into their account. The below code shows the implementation of the mylogin() method which will log the user into their account if the correct details have been submitted.

```
def mylogin(request):

    if request.method == 'POST':

        uuser = request.POST.get('username')
        upassword = request.POST.get('password')

        if uuser != "" and upassword != "":

            user = authenticate(username = uuser, password = upassword)

            if user != None:

                login(request, user)
                return redirect('panel')

    return render(request, 'front/login.html')
```

Figure 14: views.py file in main app

The system retrieves the information provided by the user and checks the details to that supplied in the database. If the fields match and the user is authenticated, they will be redirected to the Admin Panel page of the application. If the user has supplied incorrect login details, the user will be redirected to enter their login details again.

Create Blog Post

Creating a blog post is the most fundamental feature for this application. The healthcare professional can add a title, category, summary, body text, image, and multiple tags to the blog post. I created a form in the HTML for the user to fill in.

```
<form action="{% url 'posts_add' %}" method="post" enctype="multipart/form-data" class="form-horizontal form-bordered">
    {% csrf_token %}
    <div class="form-group">
        <div class="col-md-6">
            <input type="text"  id="poststitle" name="poststitle" class="form-control" placeholder="Post Title">
        </div>
        <div class="col-md-6">
            <select id="postscat" name="postscat" class="select-chosen" data-placeholder="Choose a Category" style="width: 250px;">
                {% for i in category %}
                <option value="{{i.pk}}">{{i.catname}} | {{i.name}}</option>
                {% endfor %}
            </select>
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-12">
            <textarea type="text" id="postsummary" name="postsummary" rows="5" class="form-control" placeholder="Summary">Summary</textarea>
        </div>
    </div>
    <script>
        CKEDITOR.replace('postsummary')
    </script>
    <div class="form-group">
        <div class="col-md-12">
            <textarea type="text" id="postbody" name="postbody" rows="5" class="form-control" placeholder="Body">Body</textarea>
        </div>
    </div>
```

Figure 15: login.html

```
    <script>
        CKEDITOR.replace('postbody')
    </script>
    <div class="form-group">
        <div class="col-md-6">
            <input type="file" id="myfile" name="myfile" rows="5" class="form-control" placeholder="Body">
        </div>
        <div class="col-md-6">
            <textarea type="text" id="tag" name="tag" rows="5" class="form-control" placeholder="Tags"></textarea>
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-12">
            <button type="submit" class="btn btn-sm btn-primary"><i class="fa fa-angle-right"></i> Submit</button>
        </div>
    </div>
</form>
```

Figure 16: login.html

A text editor was implemented into the summary and body text sections to allow users to edit and style their input.

Below is the code in the views.py that was implemented to successfully add a blog style post.

```
if request.method == 'POST':

    poststitle = request.POST.get('poststitle')
    postscat = request.POST.get('postscat')
    postsummary = request.POST.get('postsummary')
    postbody = request.POST.get('postbody')
    postid = request.POST.get('postscat')
    tag = request.POST.get('tag')

    if poststitle == '' or postsummary == '' or postbody == '' or postscat == '':
        error = "All Fields are Required"
        return render(request, 'back/error.html', {'error':error})

    try:

        myfile = request.FILES['myfile']
        fs = FileSystemStorage()
        filename = fs.save(myfile.name, myfile)
        url = fs.url(filename)

        if str(myfile.content_type).startswith("image"):

            if myfile.size < 5000000:

                postname = SubCategory.objects.get(pk=postid).name
                ocatid = SubCategory.objects.get(pk=postid).catid

                b = Posts(name = poststitle, short_txt = postsummary, body_txt = postbody, date = today, img = filename,
                          imgurl = url, author = request.user, catname = postname, catid = postid, views = 0, time = time,
                          ocatid = ocatid, tag = tag, rand=rand)
                b.save()
```

Figure 17: views.py file in posts app

The method retrieves all the information submitted by the user and ensures that all text fields have been submitted. If the user fails to input all the necessary fields, an error message will display. The image is also saved and stored using the FileSystemStorage class. Other checks are conducted to ensure the image that is submitted is valid and is below the required size. If successful, the post will be saved in the database and the user will be redirected to the post_list page. The administrator will then view the post to ensure it is creditable and publish or delete depending on the answer.

View and Edit Blog Post

All users have access to view blog posts created by healthcare professionals. However, only authenticated healthcare professionals can view and edit their blogposts that they created. Administrators have the ability to view all users blogposts and edit them if needed. Below is the code to view all blogposts related to the user.

```python
def posts_list(request):
    # Authenticating user
    if not request.user.is_authenticated:
        return redirect('mylogin')
    # End login check

    perm = 0
    for i in request.user.groups.all():
        if i.name == "masteruser" : perm = 1

    if perm == 0:
        posts = Posts.objects.filter(author=request.user)
    elif perm == 1:
        postss = Posts.objects.all()
        paginator = Paginator(postss,5)
        page = request.GET.get('page')

        try:
            posts = paginator.page(page)

        except EmptyPage:
            posts = paginator.page(paginator.num_page)

        except PageNotAnInteger:
            posts = paginator.page(1)


    return render(request, 'back/posts_list.html', {'posts':posts})
```

Figure 18: views.py file in posts app

The above code shows that if the administrator ('masteruser') or the author of the post is signed in, then they have the ability to view all posts relating to the user. Pagination was also implemented in this code to allow 5 posts per page.

The edit functionality is the same for the add post code shown above alters the changes made in the database.

Delete Blog Post

If the user is a healthcare professional who created the post or is the administrator, then they have the ability to delete blogposts. The code below shows the implementation of deleting a blogpost.

```
try:

    b = Posts.objects.get(pk=pk)

    fs = FileSystemStorage()
    fs.delete(b.img)

    ocatid = Posts.objects.get(pk=pk).ocatid

    b.delete()

    count = len(Posts.objects.filter(ocatid = ocatid))

    m = Category.objects.get(pk = ocatid)
    m.count = count
    m.save()

except:

    error = "Something Went Wrong"
    return render(request, 'back/error.html', {'error':error})

return redirect('posts_list')
```

Figure 19: views.py file in posts app

As can be seen above, the system retrieves the selected blogpost using the pk and deletes the post as well as the category ID associated to it. If successful, the user will be redirected to the posts_list view. If an error occurs, the user will be redirected to an error page.

Live Chat Room

A live group chat room has been created to allow all authenticated users the ability to communicate in a real time environment. ASGI, or the Asynchronous Server Gateway Interface, is the specification which Django Channels are built upon that was used to create my Live Chat Room functionality. Below is the asgi.py file that was created.

```
import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'rdpproject.settings')

application = get_asgi_application()
```

Figure 20: wsgi.py file in rdpproject app

A connections.py file was created to handle the connections between the client and the server. Any authenticated user connecting to the application will be added to the 'users' group and will receive messages sent to the server. When the client disconnects from the application, the channel is removed from the group, and the user will stop receiving messages. Below is a snippet of the code from the consumers.py class.

```
class ChatRoomConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope['url_route']['kwargs']['room_name']
        self.room_group_name = 'publicchat_%s' % self.room_name

        await self.channel_layer.group_add(
            self.room_group_name,
            self.channel_name
        )

        await self.accept()

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(
            self.room_group_name,
            self.channel_name
        )

    async def receive(self, text_data):
        text_data_json = json.loads(text_data)
        message = text_data_json['message']
        username = text_data_json['username']

        await self.channel_layer.group_send(
            self.room_group_name,
            {
                'type': 'chatroom_message',
                'message': message,
                'username': username,
            }
        )
```

Figure 21: consumers.py file in publicchat app

The below code was implemented as a script in my chatroom.html file.

```
<script>
    const user_username = JSON.parse(document.getElementById('user_username').textContent);
    document.querySelector('#submit').onclick = function (e) {
        const messageInputDom = document.querySelector('#input');
        const message = messageInputDom.value;
        chatSocket.send(JSON.stringify({
            'message': message,
            'username': user_username,
        }));
        messageInputDom.value = '';
    };

    const roomName = JSON.parse(document.getElementById('room-name').textContent);

    const chatSocket = new WebSocket(
        'ws://' +
        window.location.host +
        '/ws/publicchat/' +
        roomName +
        '/'
    );

    chatSocket.onmessage = function (e) {
        const data = JSON.parse(e.data);
        console.log(data)
        document.querySelector('#chat-text').value += (data.username + ': ' + data.message + '\n')
    }
```

Figure 22: chatroom.html file

2.4 Graphical User Interface (GUI)

Login/Register Page

The first page the user will encounter is the login or register page. The user will be asked to sign in using their login credentials. Alternatively, if they do not have an account, they can select to register.

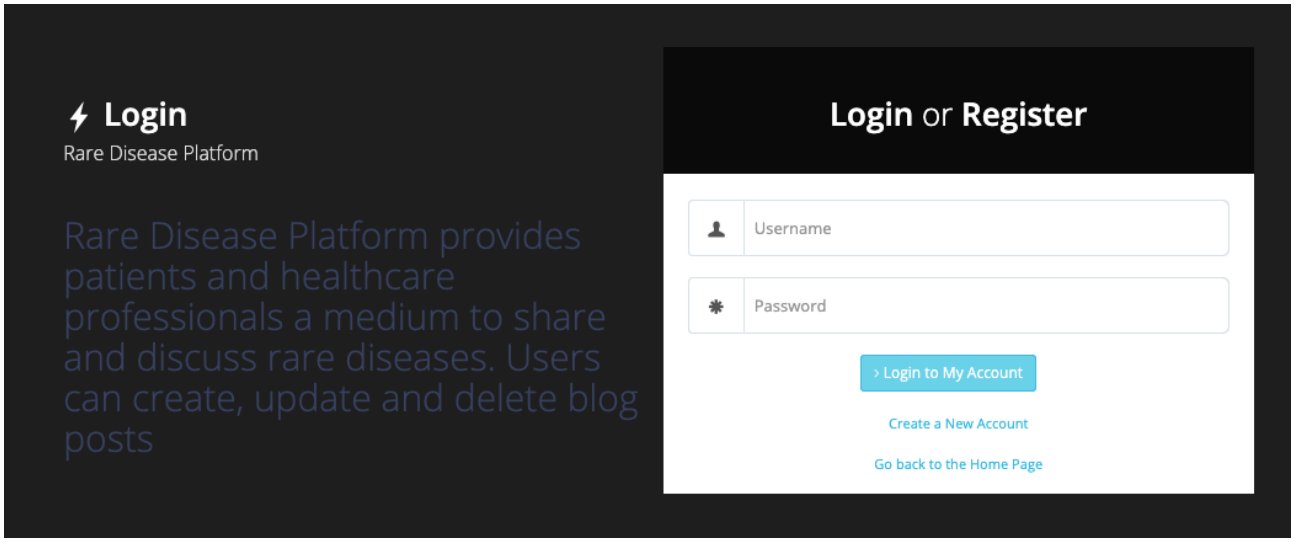


Figure 23: Login page

Here, the user can select to sign into their account. If they have not registered they can select ‘Create a New Account’.

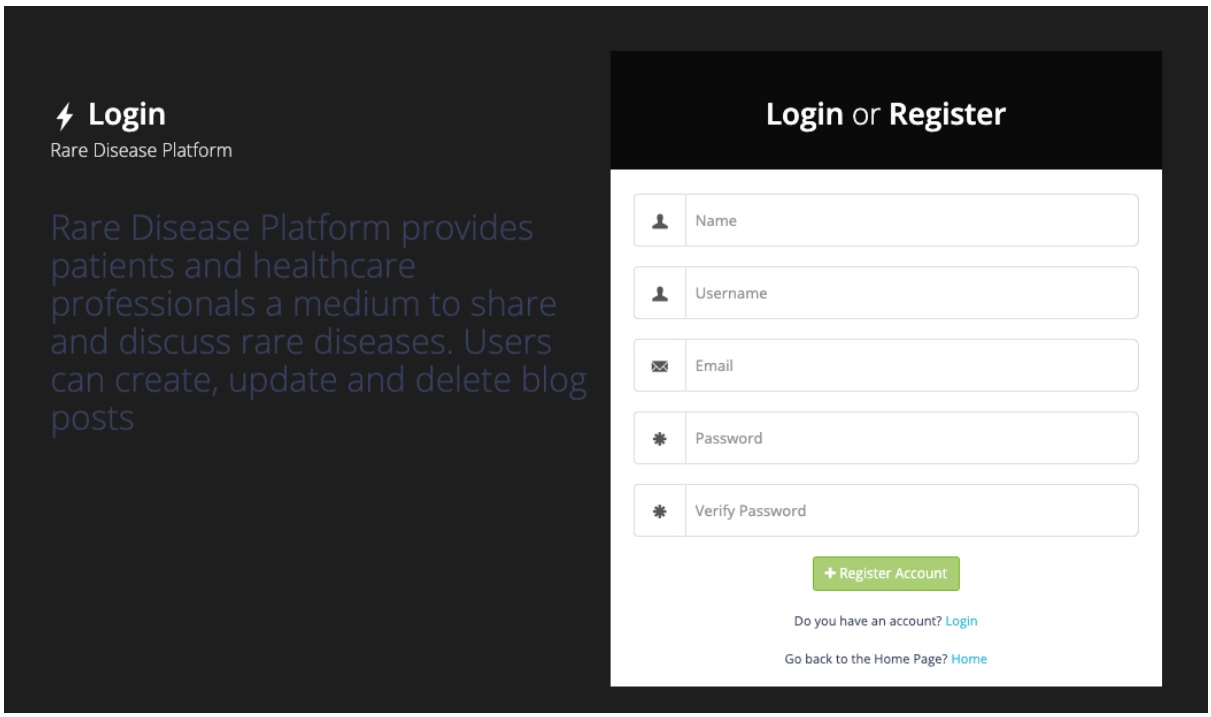


Figure 24: Register page

To register as a new user, the user will have to enter an name, username, email and a password. For healthcare professionals, they will need to upload a professional document that will be reviewed to ensure they are healthcare professionals. The super admin will have the ability to accept or revoke their application.

If the user already has an account, they can select the ‘Already have an Account?’ option.

Landing Page

The below is the main landing page all users will be able to view. This will consist of all the posts that have been created by all the healthcare professionals.

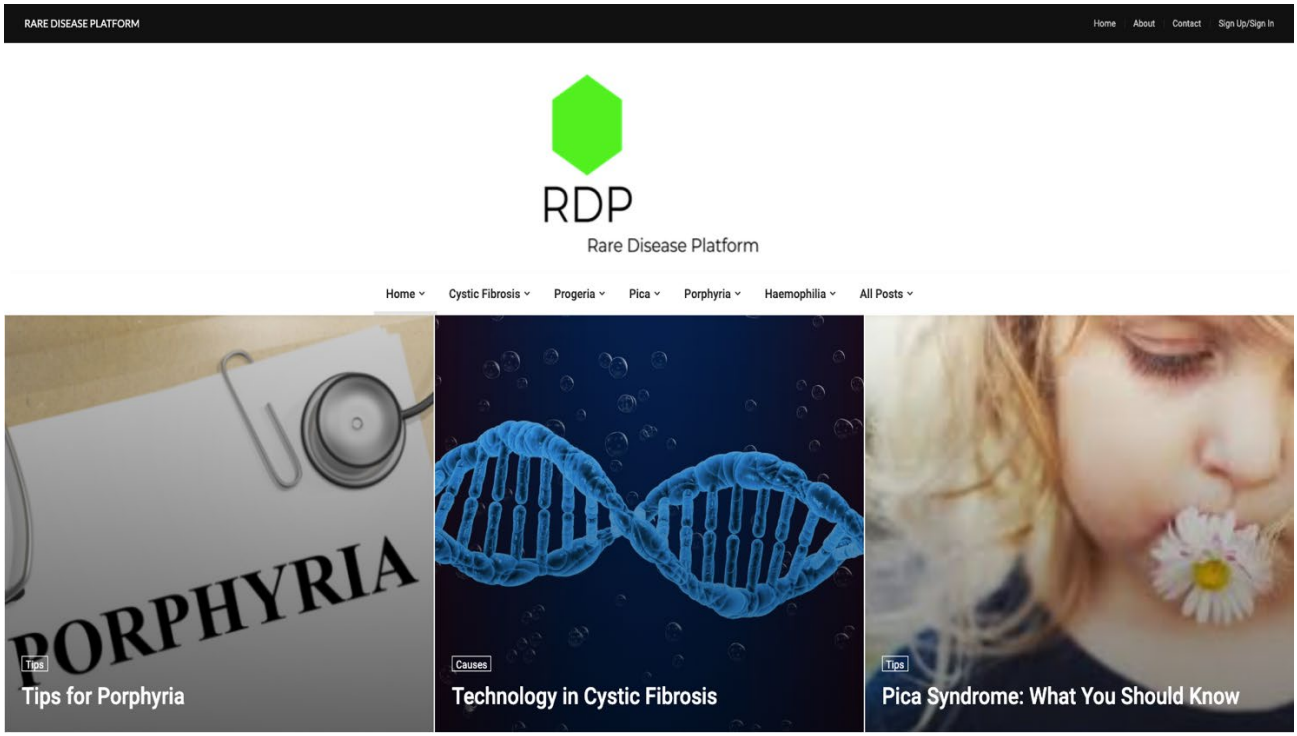


Figure 25: Home/Landing page

The top navigation bar will allow users to go to the About, Contact, and Sign In/Sign Up page if they have not signed in. If a user is signed in, the navigation bar will show Live Chat and Admin links. This will direct the user to the live Group Chat page and Admin portal pages appropriately.

The landing page also allows users to view all the blog posts that have been created. They can select on a category to view all posts in selected category. The user can also select All Posts to view all available posts.

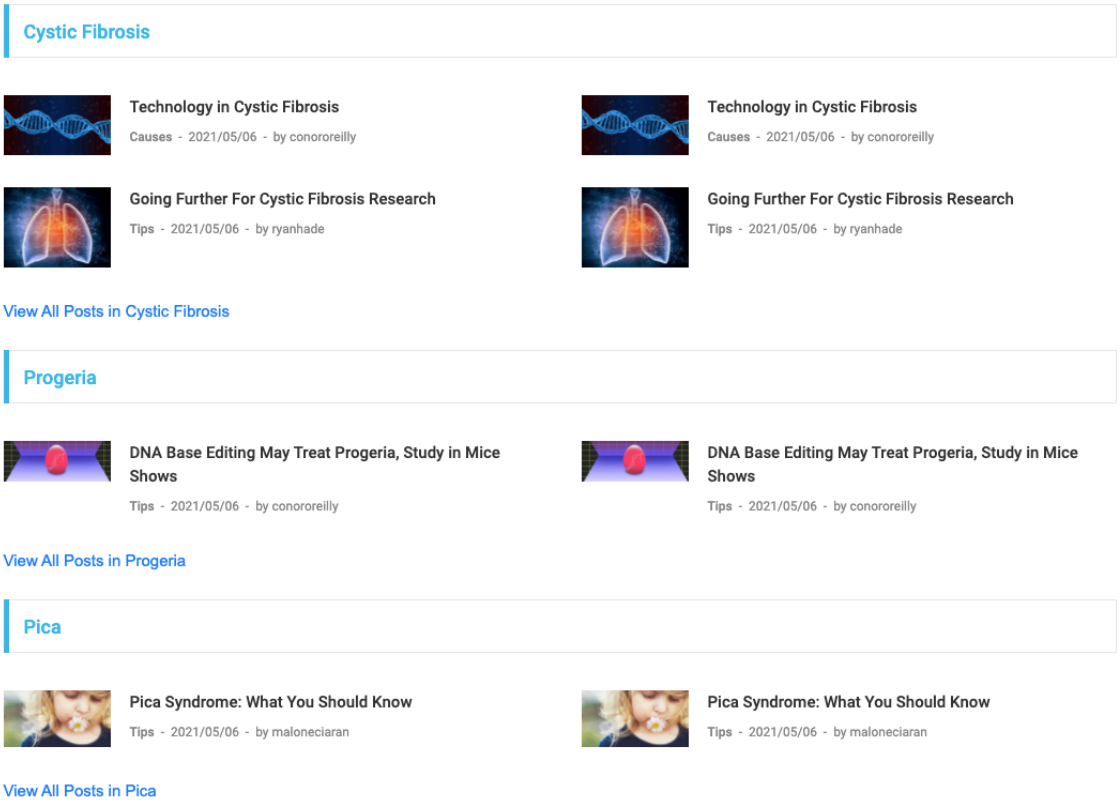


Figure 25: Home/Landing page

When a user scrolls down the page, they can see posts available by their selected category. The left hand side shows the most recent posts in order. The right hand side shows the most viewed posts in order.

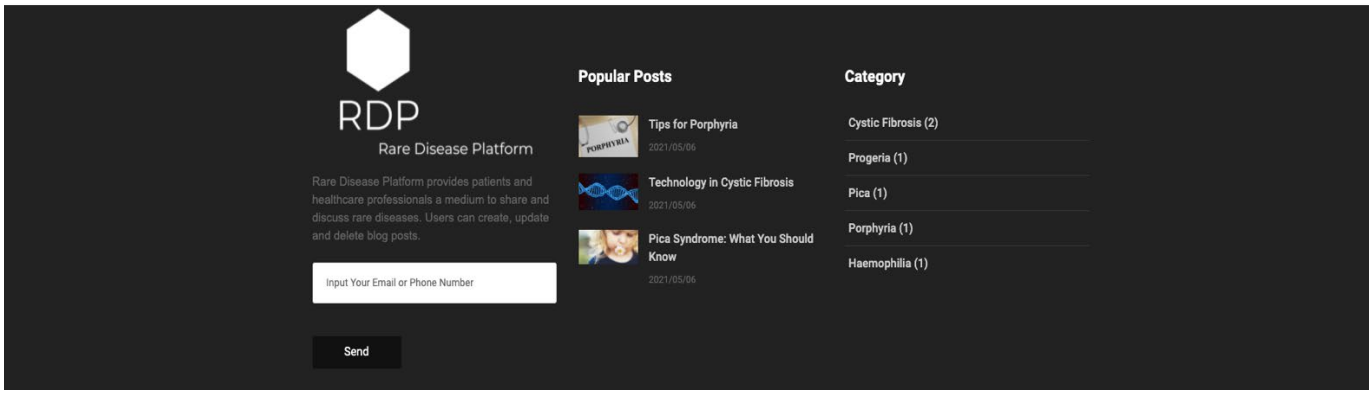


Figure 26: Footer

The footer also allows users the ability to enter their phone number or email to opt for a monthly newsletter. The footer also shows the 3 most popular posts on the website and a list of all the categories.

Add Page/Blog

The healthcare professional is able to add pages and blogs.

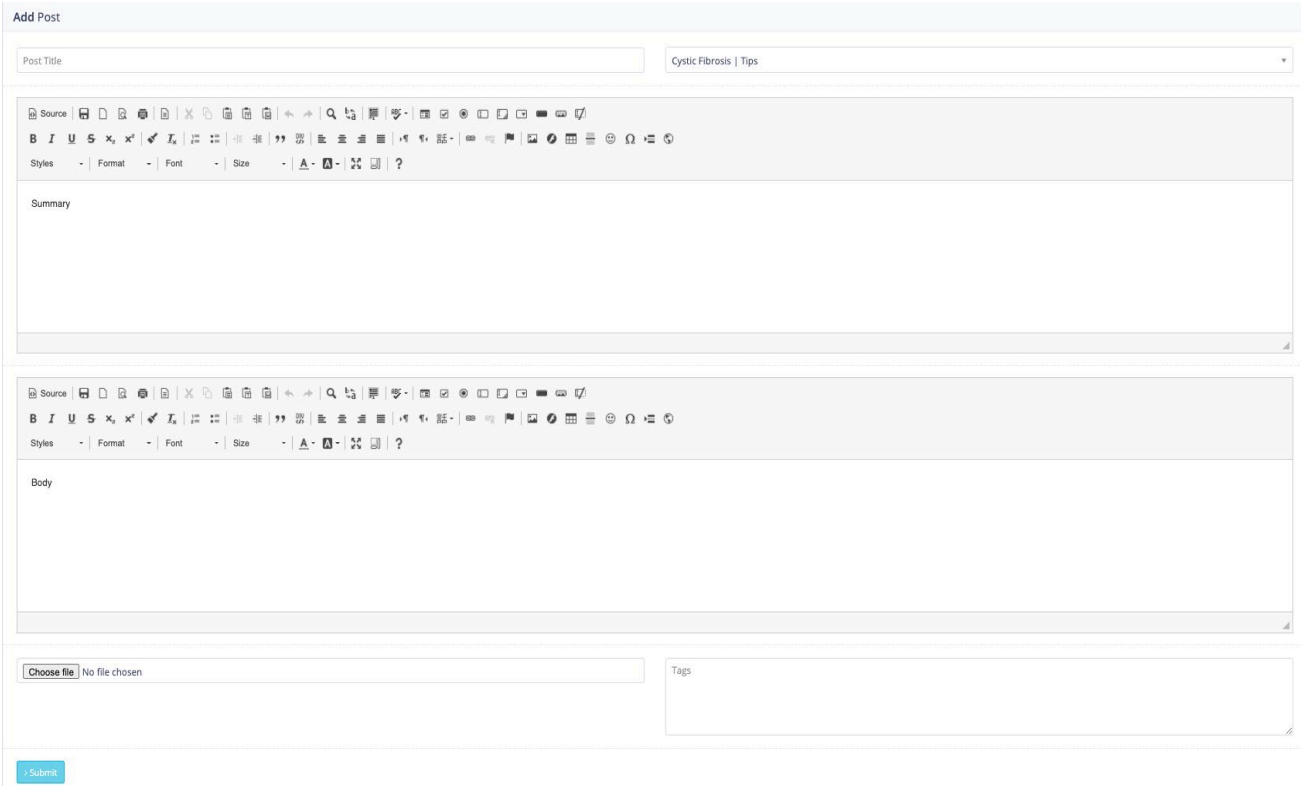


Figure 27: Add Post page

The Add Post page allows healthcare professionals and administrators the ability to create posts. The user must enter a Title, Summary, Body text and any tags associated to the post. The user must also select a category and upload an image.

Show Blog Posts

Once a blog post has been created, the user can view the content by selecting the title or image that will be attached.

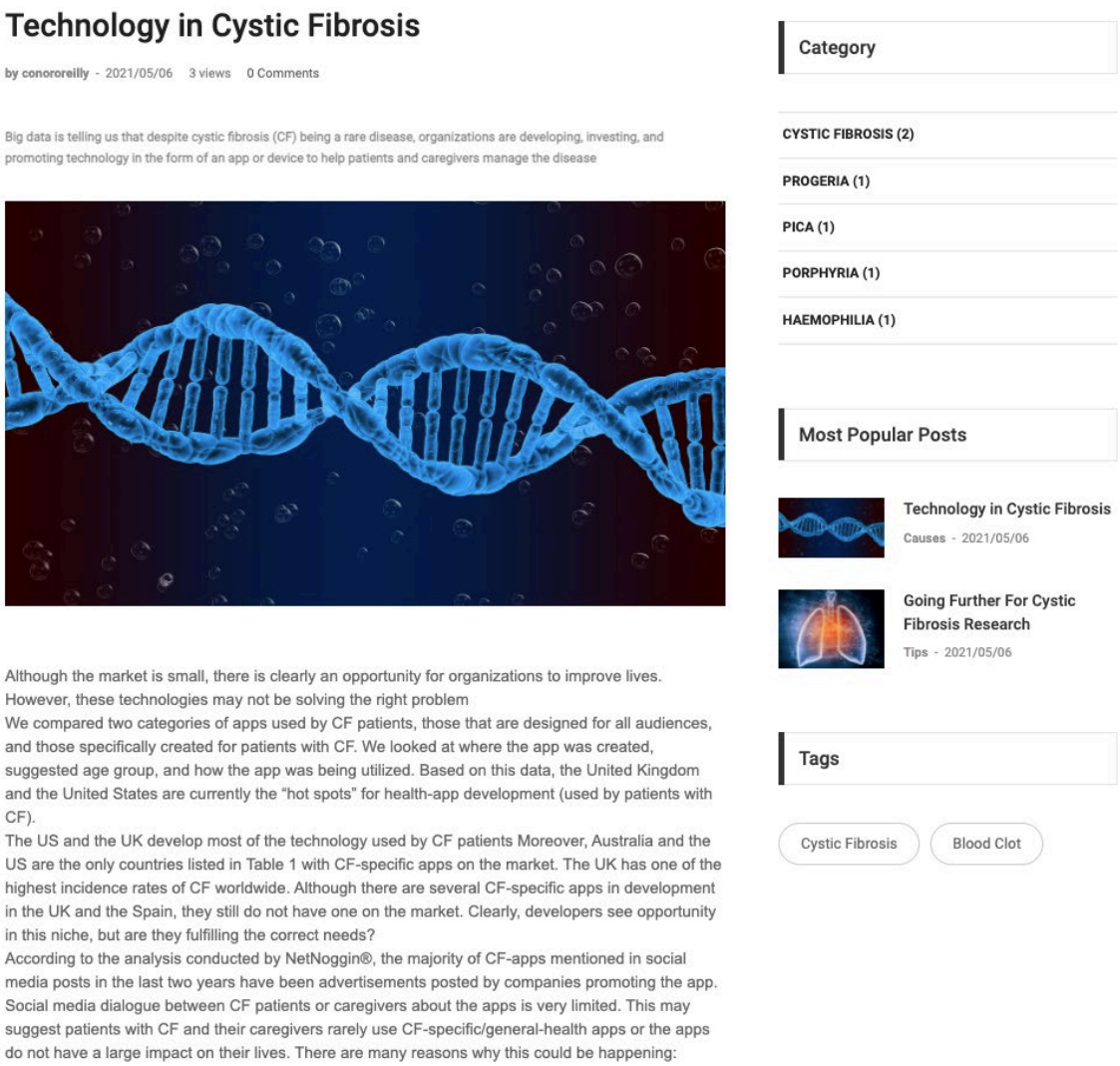


Figure 28: Post detail page

The user can leave comments and read previous comments left by other users. This is to help the user communicate with other patients to learn about the rare disease. This will hopefully be a great tool to encourage people to share valid information.

2.5 Testing

Unit, Integration, and End User Tests were carried out to ensure that the application was ready for production. Unit tests were performed on the applications URLs. Integration tests were performed on the applications views. Finally, we performed various end user tests to acquire feedback from potential customers of the application. The tests that were carried out played a huge factor in determining the reliability of the application. Important changes were made in relation to the functionality and view of the application due to these tests. The use of tests also provided confident that the application executes the expected requirements of the application.

2.5.1 Unit Testing

The Unit Tests were performed on my urls.py files across my entire project. Test Driven Development (TDD) was used regularly as a method of ensuring that all code with my URLs were correct. These tests were written in a tests folder within my project in Visual Studio Code. All the tests passed successfully and no errors have occurred. Below are a few snippets of the implementation and results of each test.

```python
class TestUrls(SimpleTestCase):

    def test_about_url_is_resolves(self):
        url = reverse('about')
        self.assertEquals(resolve(url).func, about)


    def test_home_url_is_resolves(self):
        url = reverse('home')
        self.assertEquals(resolve(url).func, home)


    def test_panel_url_is_resolves(self):
        url = reverse('panel')
        self.assertEquals(resolve(url).func, panel)


    def test_login_url_is_resolves(self):
        url = reverse('mylogin')
        self.assertEquals(resolve(url).func, mylogin)
```

Figure 29: tests.py file for URL testing for main app

```
conororeilly@Conors-MacBook-Pro rdpproject % python manage.py test main
System check identified no issues (0 silenced).
...........
----------------------------------------------------------------------
Ran 11 tests in 0.019s

OK
```

Figure 30: results from URL testing for main app

As can be seen above, 11 tests were performed on my URLs for the main app in my project. All tests passed successfully. I made an error within the code to show that the test requires the correct information to pass successfully.

```
class TestUrls(SimpleTestCase):

    def test_postlist_url_is_resolves(self):
        url = reverse('posts_list')
        self.assertEquals(resolve(url).func, posts_list)


    def test_postadd_url_is_resolves(self):
        url = reverse('posts_add')
        self.assertEquals(resolve(url).func, posts_add)


    def test_allposts_url_is_resolves(self):
        url = reverse('all_posts')
        self.assertEquals(resolve(url).func, all_posts)


    def test_allpostssearch_url_is_resolves(self):
        url = reverse('all_posts_search')
        self.assertEquals(resolve(url).func, all_posts_search)
```

Figure 31: tests.py file for URL testing for posts app

```
conororeilly@Conors-MacBook-Pro rdpproject % python manage.py test posts
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..........
----------------------------------------------------------------------
Ran 10 tests in 0.030s

OK
```

Figure 32: results from URL testing for posts app

The above tests relate to my posts app within the project. Similarly, 9 URL tests were performed and all tested successfully.

Unit tests for each app can be located under the tests folder as test_urls.py for each app.

2.5.2 Integration Testing

The Integration tests were performed on my views.py files to test the functionality of requests and responses. These tests were conducted to ensure all responses were being returned successfully for production. These tests were conducted within Visual Studio Code and within each individual test folder for each app. Below are a few selected tests of code and results.

```
def test_postadd_POST(self):

    response = self.client.post(self.posts_url, {
        'name': "Name",
        'short_txt': "Short txt",
        'body_txt': "Long txt",
        'date': 1,
        'time': 1,
        'img': "Image",
        'imgurl': "Img Url",
        'author': "Author",
        'catname': "Category",
        'catid': 1,
        'ocatid': 1,
        'views': 1,
        'tag': 'Tag',
        'act': 1,
        'rand':1
    })

    self.assertEquals(response.status_code, 302)
    self.assertEquals(self.posts1.name, 'Name')
```

Figure 33: tests.py file for view tests in posts app

```
conororeilly@Conors-MacBook-Pro rdpproject % python manage.py test posts
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..........
----------------------------------------------------------------------
Ran 10 tests in 0.030s

OK
```

Figure 34: results from views testing for posts app

As can be seen above, a test was performed to check if a blog post can be posted successfully. The result shows the successfully implementation of all tests within the posts app.

```
def test_catadd_POST(self):

    response = self.client.post(self.catadd_url, {
        'name': 'Pica',
    })

    self.assertEquals(response.status_code, 302)
    self.assertEquals(self.category1.name, 'Pica')


def test_catdel_DELETE(self):

    Category.objects.create(
        name="Pica"
    )

    response = self.client.delete(self.catdel_url, json.dumps({
        'name': 'Pica'
    }))

    self.assertEquals(response.status_code, 302)
    self.assertEquals(self.category1.name, 'Pica')
```

Figure 35: tests.py file for view tests in categories app

```
conororeilly@Conors-MacBook-Pro rdpproject % python manage.py test category
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
----------------------------------------------------------------------
Ran 5 tests in 0.028s

OK
```

Figure 36: results from views testing for categories app

The above tests relate to the category views. It shows the successful implementation of adding and deleting a category.

All integration tests on the views within the project can be located under the tests folder as test_views.py for each app.

2.5.3 Usability Testing

Usability tests were implemented in this project in order for end users to interact with the application. This form of testing was incorporated to determine the applications ease of use and the overall experience of the end user while interacting with the system. This testing was accomplished by adding users as testers to the application and observe how they used the applications features. Various different usability tests were carried out and will be described in detail below.

Due to Coronavirus restrictions, I could only do some in person tests with close family members. All testers were proficient users of technology and would have used similar websites before. I was able to conduct other usability testing using online tools and asking other NCI students to participate to gather information.

2.5.3.1 5 Second Test

This test was created by using a website called Usability Hub. A still image of the home screen was shown to the testers. The users had 5 seconds to take in as much information as possible on the home page. After this, they were poised various questions regarding the home page. As the home screen is the first page the user will see and is the page where the user can navigate to other sections of the application, I wanted to ensure all information was clear and the users could easily locate where to go.

Three questions were asked to the testers:

1. What does RDP do?
2. What did you like most, if anything, about the system?
3. What did you dislike, if anything, about the system?

Three family members agreed to take part in the test along with 2 other NCI students. All participants were computer literate and fully understood what was expected of them. I have outlined below the questions that were asked and the answers that were provided by the testers. Users were shown the below screen to observe for 5 seconds.

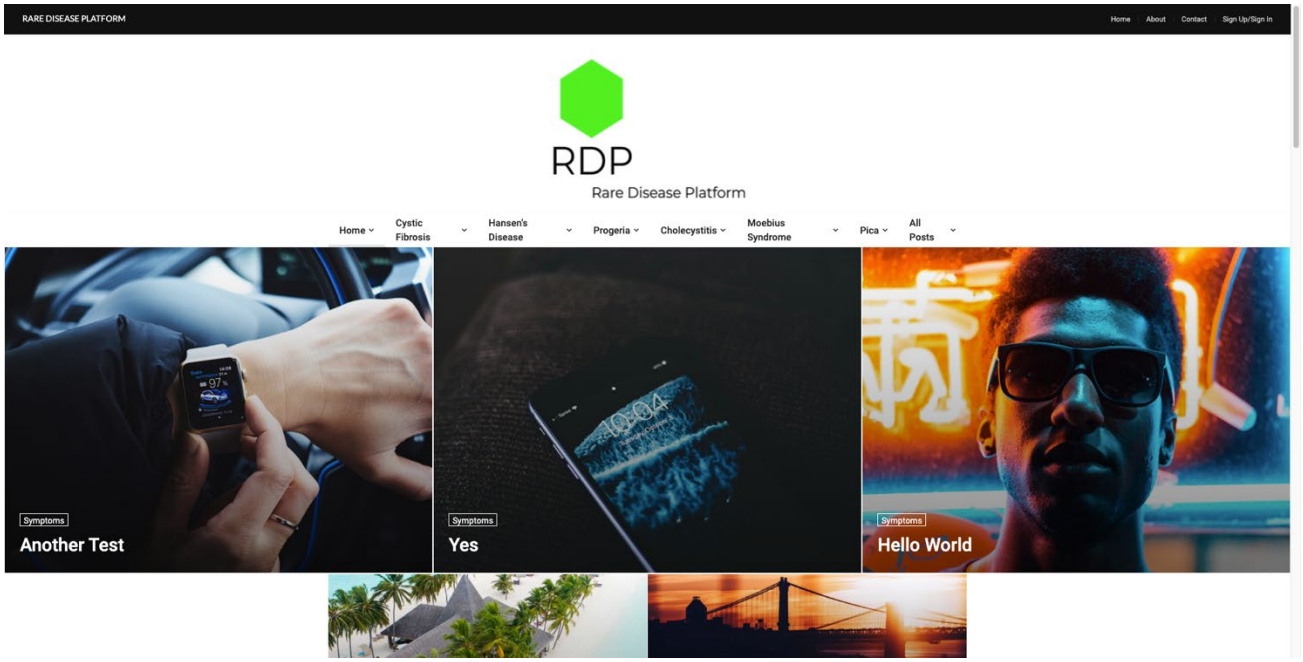


Figure 37: home/landing page

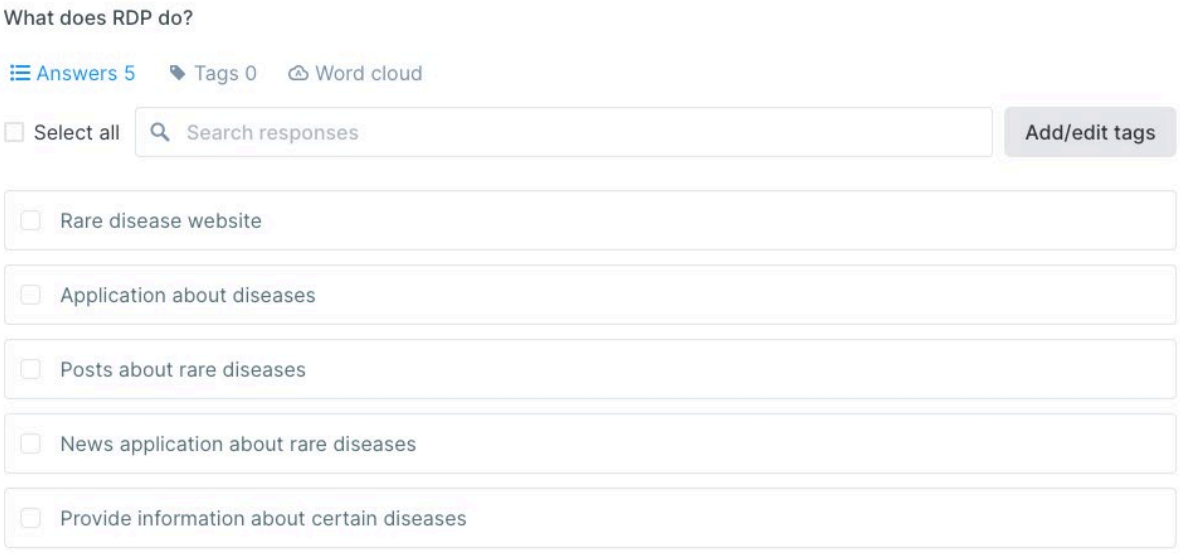


Figure 38: Results from 5 Second Test (1)

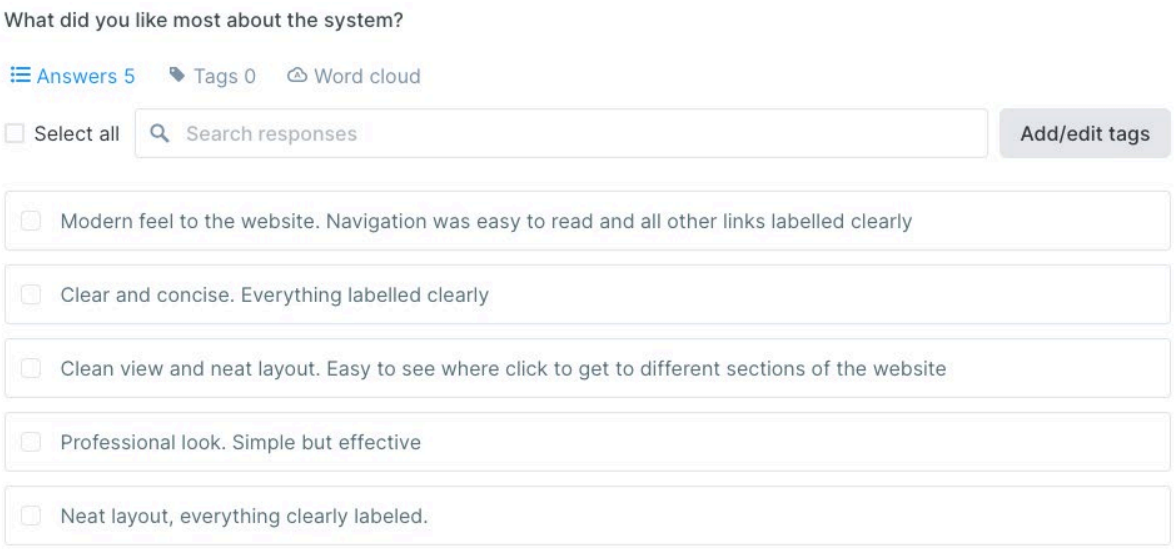


Figure 39: Results from 5 Second Test (2)

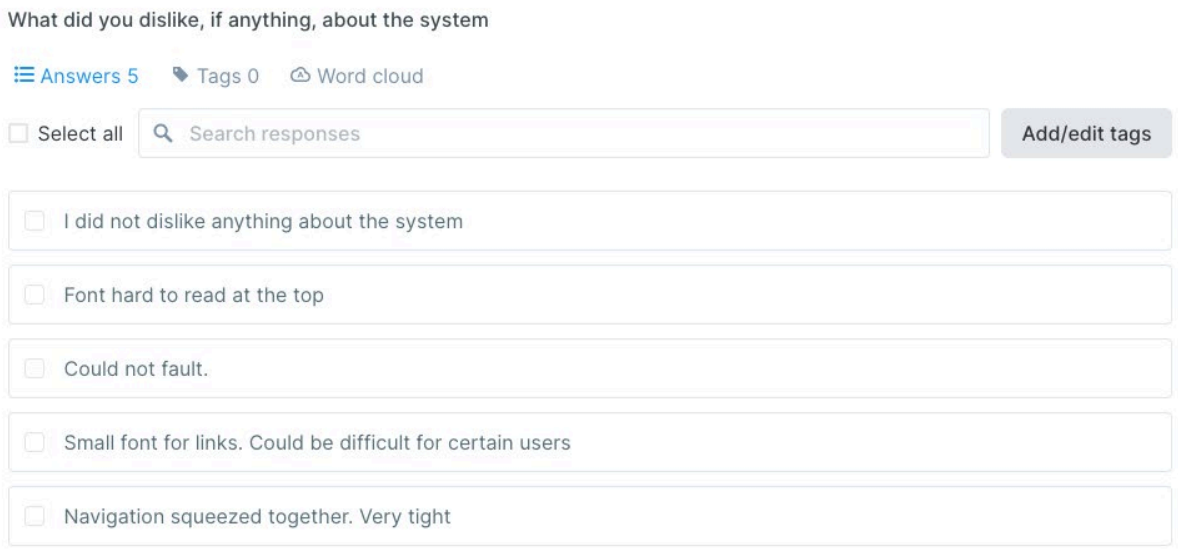


Figure 40: Results from 5 Second Test (3)

I was very pleased with the results of the test as all testers seemed to understand the concept of the system and interested in the applications other features. All testers showed clear knowledge of what the application was about. Testers seemed to like the simple design of the home page and could clearly note where the navigation bar and other links were located. Some testers also said that the font on certain links were too small to read and the navigation bar was 'squeezed together'. I will increase the font size of the links and edit my CSS for the navigation bar to provide a cleaner look.

2.5.3.2 Trunk Test

I also conducted a Trunk Test on my users to gain further information about the users experience while using the RDP system. I gave 3 testers that were in my household access to the RDP system and developed various tasks for them to complete. I asked the testers to

complete 3 different tasks and recorded whether they completed the test successfully and how long it took them to complete.

Tester 1

Q1. Sign up as a new user

Result: Selected 'Sign Up/Sign In' and navigated to sign up page. Entered correct login details and successfully logged in.

Time Taken: 45 seconds.

Remark: 'Easy to see where to select and what details to provide. As I have set up personal accounts before, I found the task easy to complete.'

Q2. Create a new category

Result: Once signed in and user permissions authenticated, the user navigated to the admin page. From here, they selected the category drop down menu and entered a new category. Selected 'Submit' and category successfully generated.

Time Taken: 50 seconds

Remark: 'I was unsure whether to navigate to the admin page or not to create a new category. After a quick glance over the main home page, I noticed no option to create a new category. I entered the admin portal and easily located the Add Category option.'

Q3. Create a new blog post

Result: As a result of previously setting up a category, user could easily locate the Add Post option. Entered dummy text into the fields and uploaded a sample image. Selected 'Submit' and post uploaded successfully.

Time Taken: 45 seconds

Remark: 'As I was aware of the admin portal, I knew where to locate the Add Post function. The text fields were clearly labelled and I knew exactly what to insert.'

Tester 2

Q1. Sign up as a new user

Result: Selected 'Sign Up/Sign In' and navigated to sign up page. Could not locate the 'Create a new User' option initially. Once found, had no issue with the login details and successfully signed in.

Time Taken: 55 seconds.

Remark: ‘Could not find he create a new user option initially. However, once I found the correct page I was able to successfully sign in with my login details.’

Q2. Create a new category

Result: Once signed in and user permissions authenticated, the user browsed the home page to find the information they required. The tester navigated to the top of the page and selected Admin correctly. They selected the Add Category option and successfully created a new category.

Time Taken: 90 seconds

Remark: ‘I was not aware there was an Admin portal option for the user. However, once I could see that there was no option to create a category on the home page, I was able to easily locate the admin option at the top of the page and had no issues after.’

Q3. Create a new blog post

Result: As a result of previously setting up a category, user could easily locate the Add Post option. The user had an issue with the image option as they inserted a document by mistake. They navigated back to the page and successfully uploaded a correct format.

Time Taken: 55 seconds

Remark: ‘I was able to easily find the Add Blog Post option. I selected a document by mistake instead of an image. The error message was clear and I understood my error. I was able to easily navigate back to the page and successfully added a new blog post.’

Tester 3

Q1. Sign up as a new user

Result: Selected ‘Sign Up/Sign In’ and navigated to sign up page. Once found, they entered their details wrong for certain fields. They also entered their verify password incorrectly. However, when they retried, they had no issues and signed in successfully.

Time Taken: 70 seconds.

Remark: ‘Found the correct page to set up an account easily. I did not read the text fields correctly and entered by first name and second name in the ‘Name’ and ‘Username’ sections. I also entered my second password incorrectly. However, I was able to see where I went wrong due to the error page and entered the correct details the second time around‘

Q2. Create a new category

Result: Once signed in and user permissions authenticated, the user selected the Admin portal option immediately. They were able to locate the Add Category option and successfully create a category.

Time Taken: 25 seconds

Remark: 'I noticed the Admin option as soon as I was redirected back to the home page. From here, the page is clearly labelled and had no issue creating a category'

Q3. Create a new blog post

Result: As a result of previously setting up a category, user could easily locate the Add Post option. The user entered information into the fields and had no issue uploading the correct image file format.

Time Taken: 35 seconds

Remark: 'It was quite clear where to go to create a blog post. I liked the text editor that was included and would use it if I was a user of the website.'

Overall, this was a good insight into the main functionality of the application. All three users were able to sign up successfully with very little issues. They were also able to navigate to Admin portal in order to create a category and blog post. The Admin link has been changed to clearly display to users as some users did not noticed the new links available once signed in. I am happy with these results as they are the main functions of the application.

2.5.3.3 Click Test

A Click Test is a beneficial technique used to see if users know where to click to find certain sections of the application. I used the Usability Hub website to create a Click Test for users to answer. I sent a link to family and college friends to complete the task if they could. I received a total of 6 responses. The two tasks I asked the users were:

1. Please indicate where you would click to view All Posts
2. Please indicate where you would click to go to the authenticated user portal, i.e. Admin

I also asked a follow up question for each task to better understand their reasoning for their answers.

1. Was it clear that this is where you would find all the posts?
2. Would you change the word Admin to make it more obvious this is the main admin portal?

Below are the results of my two tests.

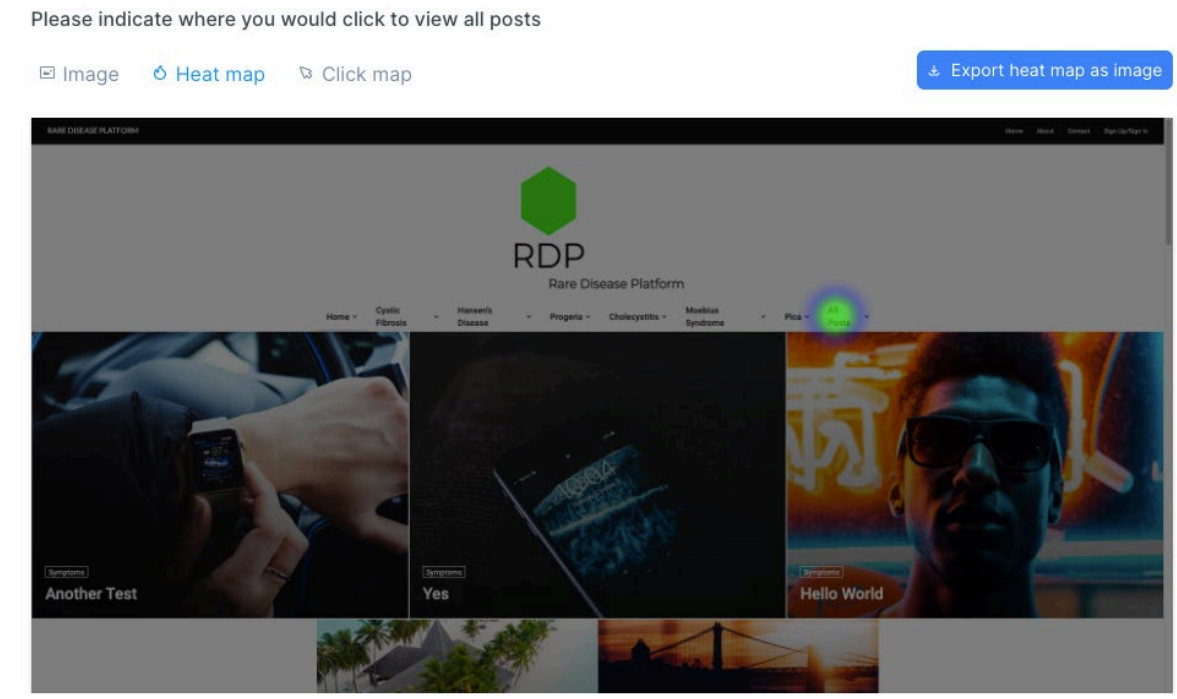


Figure 41: Results from click test (1)

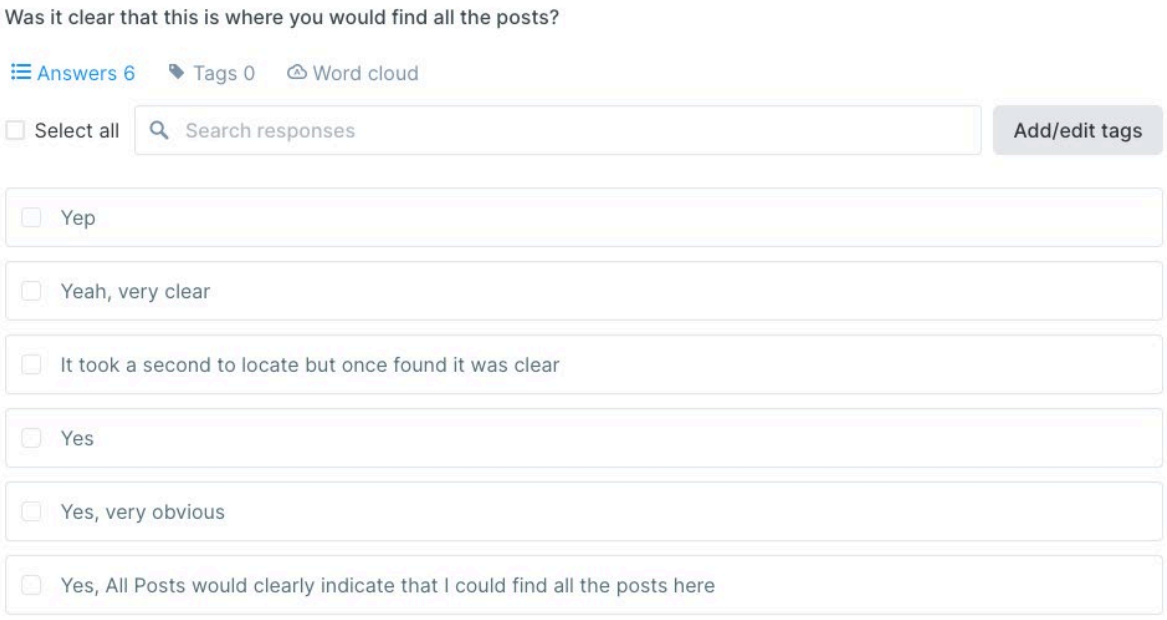


Figure 42: Results from click test (2)

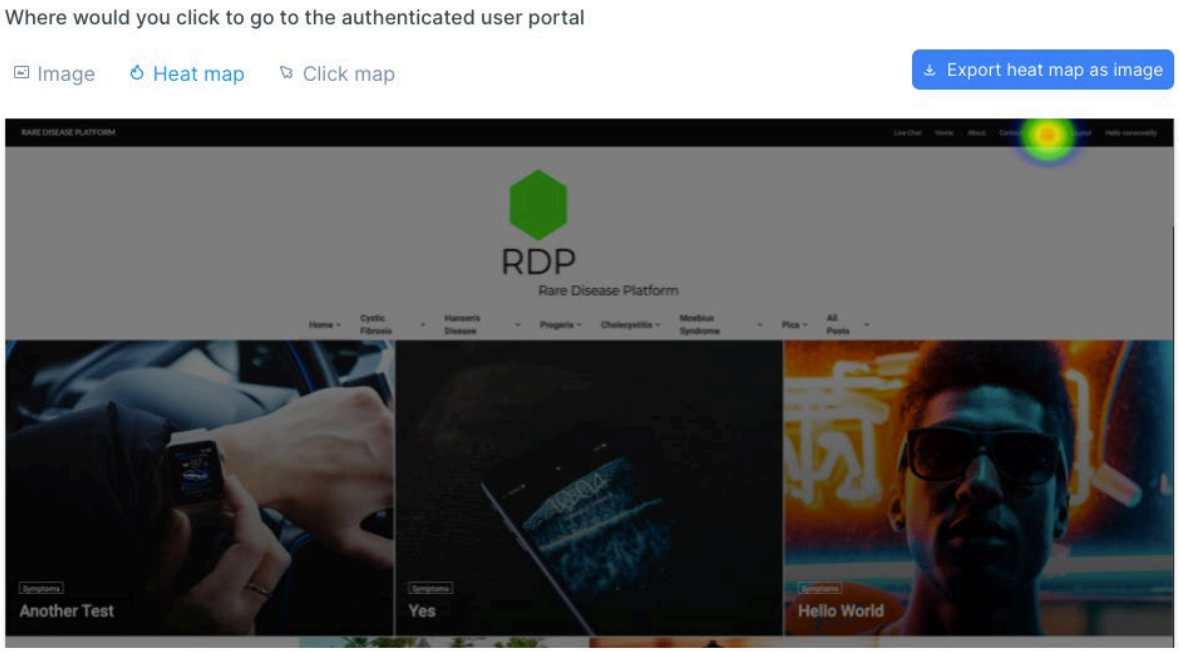


Figure 43: Results from click test (3)

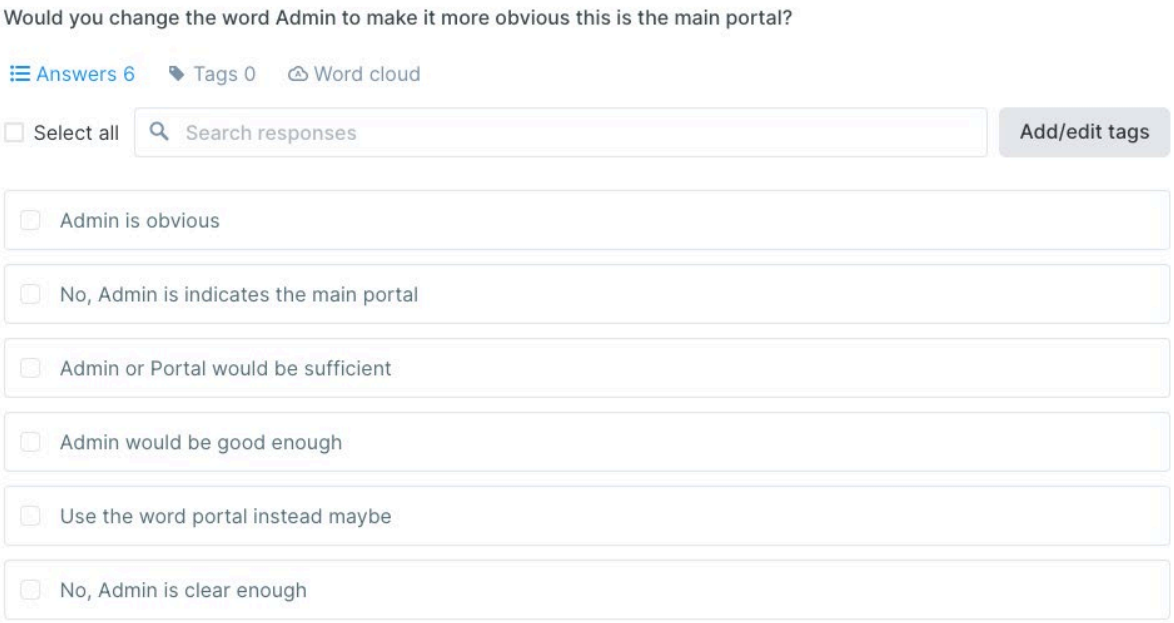


Figure 44: Results from click test (4)

As can be seen above, all the users were able to successfully locate the correct link for both tests. I am very happy with this response as I thought they may have been difficult to locate on the home screen. All users were also very happy with the use of the word ‘Admin’ and that it would strongly convey where this would lead the user. Overall, this test was a massive success and I am happy to keep the current version of the application.

2.6 Evaluation

The initial unit testing, as described above, was successful during the development process and helped me to ensure that the functionality of the application was correct and that any further functionality that I would be adding would not cause the application to crash. URLs are very important in every application and I was confident that my approach to adding the links were successful for further development.

The Integration tests also proved to be a major success. The views and functionality all proved to be correct and I had no issue going forward when implementing more of the applications functions. I found both the Unit and Integration tests hugely beneficial as they provided me with assurance that my application was working correctly and would not cause any further issues down the line.

Although both the above tests were massively beneficial, I found the end user testing to be the most informative and provided me with the most interesting results. The feedback I received from this form of testing was mainly positive in relation to speed, responsiveness, navigation and the overall look and quality of the application. All of my testers successfully completed the tasks that were provided to them. They also supplied feedback for sections that could be improved in relation to the design of the application. I used this feedback to improve the application.

Overall, the results of the testing meant that I had succeeded in developing a commercially ready application that provided users with a fast, reliable and user friendly platform.

3 Conclusions

The strengths of the Rare Disease Platform project is the overall user experience of the application. The feedback I received from the end user testing is that the GUI was very appealing and it was very easy to navigate. One of my main goals was to allow users the ability to easily locate all the important information on the application. My feedback from testing proved that this was a massive success and all my future users will be able to locate any information they require.

Another strength of my application is the CRUD functionality of creating posts, categories, and sub categories. It is very easy for users to create a post or category. The users need to enter all the required fields that are necessary. These posts will not be published until the administrator verifies that they are correct and trustworthy. This is an important aspect of the application as users must be able to trust the information that is shared on the website.

The ability for the user to sign up is also another major strength as without verified users they will not be able to upload posts and categories. To ensure that users are verified, the administrator must view their credentials and verify that they are in fact healthcare professionals or patients. This will provide a trustworthy aspect to the application.

The Live Group Chat functionality is also a major strength to this application. The ability for users to communicate in a live environment is very beneficial for patients and healthcare professionals. They will have the ability to talk and receive information instantly. They will only be able to talk to authenticated users of the application. This feature is only available locally and I will continue to try add this feature to my deployed version.

A weakness of my application that I could not solve was the issue regarding the database. I had initially planned to develop the application using a Firebase Database. However, this proved to be very challenging so I migrated my database to a Postgres database instead. Unfortunately, this caused issues with some of my application functionality including adding a blog post. As a result, I had to deploy my application on Heroku using the default SQLite database. I am very disappointed with this outcome and I plan to continue to work on this section to solve the issue.

I also wanted to deploy my application using Amazon Web Services. However, I have never deployed this application using this resource and I struggled to deploy via AWS. As a result, I deployed the application using Heroku and it has been successful.

I also wanted to create a function that could verify whether a user was a healthcare professional. Unfortunately, I could not find a method to complete this. I researched thoroughly to find a possible solution but I could not find anything that could help me with this section. Instead, I have installed a procedure where users must be verified by the administrator. The administrator can grant certain permissions to users whether they are a healthcare professional or patient. I hope to find a solution to this problem in the future.

4 Further Development or Research

I would love the opportunity to further develop this application and fix all the issues I have mentioned above in my weaknesses. If I had more time, I would definitely like to fix the database and live group chat functionality. I would also like to create a method that would check the status of the users credentials and grant them access to be a healthcare professional if all criteria was met.

I would also like to create a more functional database like Postgres or MySQL. The current default database is not ideal for commercial use and I would like to migrate the SQLite database to a more reliable and robust database like Postgres or MySQL.

I would also have liked to research how to implement an E-Commerce style functionality that would allow healthcare professionals the ability to sell merchandise on the website. This could be completed using Stripe as the payment resource. However, I unfortunately ran out of time and could not attempt to implement this in my project. I would like to continue working on this project after I have submitted for grading.

The system could also benefit from a private chat messenger. I had initially planned to have this in my project but I opted to go for a live group chat. The private messenger could allow users the ability to confidentially talk to healthcare professionals or other patients. This would further enhance the functionality of the application and appeal to users more.

5 References

- Django. (2021). *Django Documentation.* [Online] Available at: https://docs.djangoproject.com/en/3.2/ [Accessed 16 November 2020].
- Django Software Foundation. (2021). *Django Documentation.* Available at: https://buildmedia.readthedocs.org/media/pdf/django/latest/django.pdf [Accessed 12 February 2021].
- Ellingwood, J. (2015). *How to use PostgreSQL with your Django Application on Ubuntu 14.04.* Available at: https://www.digitalocean.com/community/tutorials/how-to-use-postgresql-with-your-django-application-on-ubuntu-14-04 [Accessed 14 March 2021].
- Frietas, V. (2016). *How to Deploy Django Application on Heroku.* [Online] Available at: https://simpleisbetterthancomplex.com/tutorial/2016/08/09/how-to-deploy-django-applications-on-heroku.html [Accessed 2 May 2021].
- GetBootstrap. (2021). *Getting Started with Bootstrap.* [Online] Available at: https://getbootstrap.com/docs/5.0/getting-started/introduction/ [Accessed 12 January 2021].
- GitHub. (2021). *jschneir/django-storages.* [Online] Available at: https://github.com/jschneier/django-storages [Accessed 14 April. 2021].
- Herman, M. (2021). *Storing Django Static and Media Files on Amazon S3.* [Online] Available at: https://testdriven.io/blog/storing-django-static-and-media-files-on-amazon-s3/ [Accessed 29 March 2021].
- JavaScript. (2021). *The Modern JavaScript Tutorial.* [Online] Available at: https://javascript.info/ [Accessed 3 January 2021].
- LearnDjango. (2020). *Django Testing Tutorial.* [Online] Available at: https://learndjango.com/tutorials/django-testing-tutorial [Accessed 21 January 2021].
- MDN Web Docs. (2021). *JavaScript Basics.* [Online] Available at: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics [Accessed 19 January 2021].
- MDN Web Docs. (2021). *Django Tutorial: Creating our Home Page.* [Online] Available at: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Home_page [Accessed 23 November 2020].
- Sedhain, S. (2020). *Web Framework for Python: Django Book.* [Online] Available at: https://www.gti.bh/Library/assets/djangobookwzy482.pdf [Accessed 2 December 2020].
- Predum, R. (2020). *Building Your Own Django Blog.* [Online] Available at: https://medium.com/swlh/building-your-own-django-blog-part-2-78adbc516992 [Accessed 21 November 2020].
- RealPython. (2020). *Get Started with Django: Build a Portfolio App.* [Online] Available at: https://realpython.com/get-started-with-django-1/ [Accessed 30 November 2020].

- Udemy. (2021). *The Complete JavaScript Course 2021: From Zero to Expert.* [Online] Available at: https://www.udemy.com/course/the-complete-javascript-course/ [Accessed 12 January 2021].
- Udemy. (2020). *Python and Django Full Stack Web Developer Bootcamp.* [Online] Available at: https://www.udemy.com/course/python-and-django-full-stack-web-developer-bootcamp/learn/lecture/6637844?start=480#overview [Accessed 12 November 2020].
- Udemy. (2019). *Build Responsive Real World Websites with HTML5 and CSS3. [Online] Available at: https://www.udemy.com/course/design-and-develop-a-killer-website-with-html5-and-css3/learn/lecture/2630778?start=0#overview [Accessed 24 October 2020].*
- Verma, A. (2020). *Django Custom Authentication with Email.* [Online] Available at: https://medium.com/analytics-vidhya/django-custom-authentication-with-email-cc7e3919f11c [Accessed 12 January 2021].
- YouTube. (2018). *Django Testing Tutorial.* [Online] Available at: https://www.youtube.com/watch?v=hA_VxnxCHbo&list=PLbpAWbHbi5rMF2j5n6imm0enrSD9eQUaM&index=4 [Accessed 20 April 2021].
- YouTube. (2020). *Learn Django – Build an Asynchronous Chatroom with Django and Channels.* [Online] Available at: https://www.youtube.com/watch?v=F4nwRQPXD8w&t=3460s [Accessed 10 April 2021].
- YouTube. (2019). *CSS Tutorial – Zero to Hero (Complete Course).* [Online] Available at: https://www.youtube.com/watch?v=1Rs2ND1ryYc [Accessed 19 December 2020].
- YouTube. (2020). *User Registration and Login Authentication | Django (3.0) Crash Course Tutorials (pt 14).* [Online] Available at: https://www.youtube.com/watch?v=tUqUdu0Sjyc [Accessed 14 January 2021].
- YouTube. (2020). *Heorku Postgres Connection | Django (3.0) Crash Course Tutorials (pt 24).* [Online] Available at: https://www.youtube.com/watch?v=TFFtDLZnbSs Accessed [29 April 2021].
- YouTube. (2019). *Python Django Tutorial 2020 – Full Course for Beginners.* [Online] Available at: https://www.youtube.com/watch?v=JT80XhYJdBw [Accessed 27 October 2020].

6 Appendices

6.1 Project Plan

National College of Ireland

Project Proposal

Rare Disease Platform

05/11/2020

BSc (Honours) in Computing - Evening

Software Development

2020/2021

Conor O Reilly

X17101701

X17101701@student.ncirl.ie

Contents

Objectives

The objective of my application is to provide a platform that allows a diverse group of professionals and patients communicate and share information online. The information shared will be related to rare diseases such as Cystic Fibrosis, Isaacs Syndrome etc.

My objectives are to integrate all of the below features into my application.

Objective 1 - User Friendly Interface

My first objective will be to develop an appealing user interface. HTML, CSS, Bootstrap and JavaScript will be used to develop my front-end. The application will be available across desktop and mobile devices and will contain many user-friendly features.

Objective 2 - Firebase Login

Another objective will be to develop a real-time Firebase Database to store user login credentials. A user will be able to sign up as either a patient or a healthcare professional. In order for a user to sign up as a healthcare professional, the application will require the healthcare professional to provide proof that they are, in fact, working in the healthcare industry and can be credited as reliable sources.

Objective 3 - Add Pages and Blog Style Posts

I plan to include a feature that will let healthcare professionals the ability to create pages designated to particular rare diseases. Once set up, healthcare professionals can then add individual posts that may help users understand the rare disease. They can add helpful tips, treatments, videos, and other resources that may be beneficial to the user.

Objective 4 - Private Chat Messenger

I plan to include a private chat messenger to allow users to have confidential conversations with healthcare professionals. Certain information may not want to be shared online so a private chat messenger helps to allow discrete conversations.

Objective 5 - Payment through Stripe

I plan to implement a payment method that allows users the ability to pay healthcare professionals for one-on-one private sessions or to pay for certain healthcare related products or equipment. I hope to implement this feature using the Stripe platform.

Background

During my internship in 3rd year, I spent 6 months as a System Administrator for a IT healthcare company, Open Applications. They provided patient registries, health data analytics, geo-spatial analytics and many more useful features. My experience in Open Applications provided me with an idea to create a platform which allows people to easily locate credible documentation from healthcare professionals and discover more by allowing them to engage with workers and other patients alike.

Millions of people suffer from rare diseases and may find it difficult to locate appropriate and reliable information. Rare diseases like Methemoglobinemia and Fibrodysplasia Ossificans Progressiva do not have an abundance of resources that allow patients and their family or friends learn about their disease. By allowing patients and healthcare professionals a channel to communicate and engage amongst each other could help a tremendous amount of patients around the world.

I also believe this application could benefit patients through these challenging times with Coronavirus becoming a part of our daily life's and restricting us to certain quarters. By allowing users the ability to engage with healthcare professionals online, instead of face-to-face, provides them an alternative way of learning more about their disease. Patients may not want to risk leaving their homes in case they contract the virus which could have damaging effect on their already existing health conditions. Providing a simple user interface to the user, they can easily navigate the application to find the information they need.

I am comfortable using Python within the Django framework environment. I have also created Python applications before that have certain features of the project I have outlined above. I hope to combine all these features together into one main platform that could prove to be immensely beneficial to millions of people around the world.

Technical Approach

Prototyping

I have created various prototypes for my application and I am happy that they are of a high standard. As a result, I have started the progress of implementing these prototypes into my application.

Research

In relation to documentation, I will be continuously reading articles related to Python and Django as well as other sources like Udemy, Coursea to further enhance my skillset. Although the library is closed at the time of writing, I will use online resources supplied by NCI to get the most out of my application and implement them accordingly if required.

Technical Details

To build the application, I will be using HTML, CSS, and JavaScript to build my front-end while I will use Python and the Django Framework to complete my back-end. I will use a real-time Firebase Database to store my users login credentials. I plan to use Stripe to secure payments between users and ensure security.

To host my website, I am looking at many options but I think I will use Amazon Web Services (AWS) to host my application. I have used Heroku in the past but I believe AWS will be the most secure and beneficial hosting environment for my application.

Testing

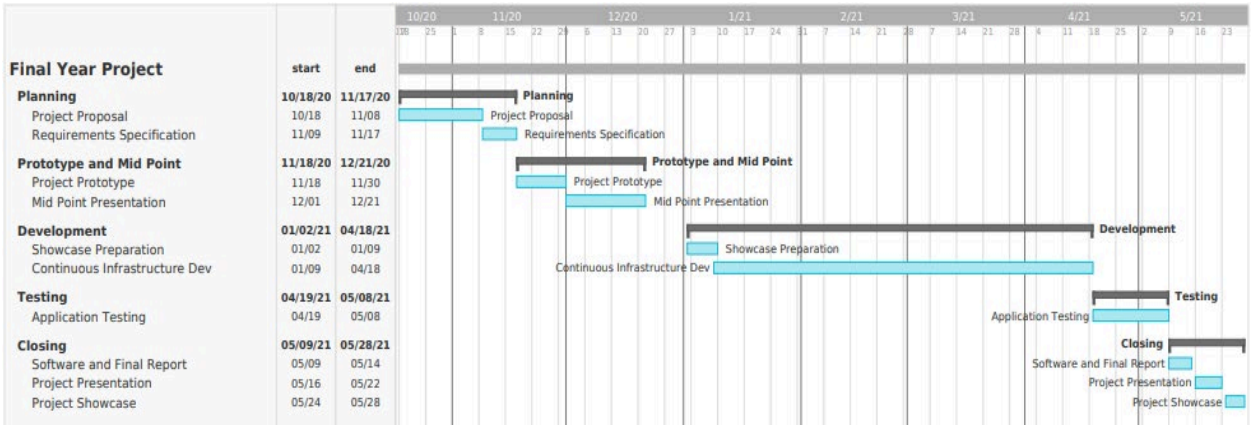
During my internship, one of the software testers was providing me with interesting resources that I could use to conduct my testing process. He provided me with a brief tutorial on Selenium and the benefits it provides for applications. I hope to implement this in my project so I can have a secure application without any dangerous bugs.

Special Resources Required

Visual Studio Code – The IDE that will be used for the development of this project be Visual Studio Code. I have used this IDE in developing projects before so I am experienced in this field.

Selenium – A portable network for testing web applications.

Project Plan



Technical Details

Front-End

CSS

Cascading Style Sheets will be used in most parts of my project for presentation and layout.

HTML

Similarly, HTML will be used to display information in my project.

Bootstrap

Bootstrap will also be used to style my application.

JavaScript

Over the past 6 months through my internship, I have completed multiple courses in relation to JavaScript. I hope to use these skills I have further developed to create a visual appealing design.

Back-End

Python

Similarly to JavaScript, I have completed multiple courses using Python. I hope to use these skills to develop my back-end for my application.

Django

Django is a high level Python web framework that I will use to secure and maintain my application.

Database

Firebase

Firebase is a Backend-as-a-Service that I will use to store my users login credentials.

Payment Service

Stripe

Stripe is a financial services company that offers payment processing software that I will use to allow users to pay for goods and services on my application.

Host Provider

Amazon Web Services

I will deploy my application using the web hosting component on Amazon Web Services.

Evaluation

Testing

For the latter stages of my application, I will use Selenium to write unit and integration tests to establish that he application will work when it is deployed. I will also use my contact in Open Applications to ask for advice and tips on how to productively use Selenium. I will also research automated testing techniques as they are very common among large companies around the world.

Once the above testing is complete and my application is deployed, I will begin distributing my application to my family and friends to ask for feedback on the applications features and ease of use. I work in a technical environment and work with software developers. I will ask them to test my application and provide any feedback from a technical point of view.

6.2 Reflective Journals

October 2020 Reflective Journal

Student Name: Conor O Reilly

Student Number: x17101701

Programme: BSc (Honours) in Computing – Evening

Supervisor: Vikas Sahni

Week 1

28th September – 3rd October 2020

My Reflection

Our first week back in college after an interesting and beneficial 6 month internship in Open Applications. Due to the Coronavirus, our lectures/seminars have been moved online and is new to both students and lecturers. I have installed Microsoft Teams on my laptop and accept all invitations for my lectures and labs.

Our software project lecture is on every Saturday at 12pm. Enda Stafford will be our lecturer and will guide us throughout the year. In week 1, Enda explains what is expected of us for the year and explains the ethics to those it applies too. He explains we must submit our video project pitch by 18th October. Throughout my internship, I have been researching potential ideas for my final year project. I bought many Udemy courses that I completed in my spare time over the last 6 months. I learned several languages such as Python, JavaScript, React, VueJS and more. I believe learning these languages will help me tremendously in my final year project.

My Achievements

An idea I have had is to create a rare disease platform that allows patients and professionals the ability to communicate and share information online. As a result of learning Python over the last few months, I will research developing the app using Python and the Django framework. I have worked with Firebase and Stripe as well over the last few months so I hope to implement these technologies as my database and potential payment option.

Issues

I found learning online different and challenging but NCI have a good system in place and I believe it will not disrupt our learning. I will begin developing a plan for my final year project and begin my project pitch video.

Week 2

5th October – 11th October 2020

My Reflection

This week we learned more about what is expected of us in regard to the project pitch video and reflective journals. I have been researching thoroughly over the last week and I have been finding further resources I could use for my final year project. I have also started my project pitch video and set out all the points I wish to make. The video must be 5 minutes max. I hope I can keep to this time frame and explain all my ideas and features within the 5 minutes.

I have begun making prototypes of what I hope my application to look like. I have also researched market competitors. I believe there are not many similar apps like the one I plan to develop. Many apps are simple forums where anyone can post content. I hope to develop an app where I can verify that the user providing information is a professional.

I have set up a Trello board on my laptop and organise when assessments for all my modules are due. I develop an initial plan and dedicate times for study of each module.

My Achievements

- Began prototypes of my application
- Conducted a market analysis
- Started a plan for semester 1.
    - Aware of all my assessments and the due dates
- Began planning my project pitch and learn how to make a Microsoft Stream video and how to upload it to Moodle.

Issues

- My project pitch is next week.
    - I hope to develop all my ideas on paper and use this as a guide for my video.

Week 3

12th October 2020 – 18th October 2020

My Reflection

This week, Enda delivered a lecture about time management and using that time efficiently. I have created a concrete plan for semester 1 and set due dates for when I hope to finish these goals. This year is already challenging but if I keep to my plan I should not have a problem.

I have been practicing for my project video pitch and I believe I have kept below the required 5 minutes and I have explained all my goals in a clear and concise manner. I upload my project pitch on Moodle and ensure I have submitted the video in the correct format. I review the video before uploading and I believe I have explained my project clearly. We hope to receive our results soon. I will begin my Project Proposal form and finish it when I get graded for my pitch.

I have begun researching further resources for my project and start assessments for other modules. I hope to complete my other assessments on time and to a high standard early so I can focus on my final year project and stay ahead.

My Achievements

- Uploaded my project pitch video.
- Continued researching for potential features and resources for my project.
- Started my other assessments in Data Applications Development, Mobile Application, and Introduction to Artificial Intelligence.

Issues

- No major issues in regard to my final year project.
    - I hope to hear back soon and then I will begin the project proposal report that is due for 8th November 2020.

Week 4

19th October – 25th October 2020

My Reflection

There are no lectures this weekend due to the bank holiday weekend. As I am waiting to hear feedback from my project pitch, I start on my project proposal and set up the report.

I have also started my other assignments in other modules. I have a due date coming up so I use this time to start the project.

I also use this time to create concrete prototypes for my final year project. I also find more resources online that will help me create an interesting and technical application.

We also discover this week who are supervisors are. Vikas Sahni will be my supervisor for the year.

My Achievements

- Started other assignments for other modules.
    - Strategic Management -> Completed my introduction and 2 paragraphs.
    - Introduction to Artificial Intelligence -> All chess pieces moving.
    - Mobile Application and Development -> Have main activity and 4 others.
    - Web Services and API -> Completed my UDP connection.
- Learned who my supervisor for the year will be
    - Vikas Sahni

Issues

- No major issues to report.
    - Happy with my progress so far in all modules.
    - Following my plan and delivering all my goals.

Week 5

26th October – 1st November 2020

My Reflection

This week, Enda explained that we should hear back soon of our results soon and confirmed our supervisors for the year. We then went over the project proposal template. I have a clear understanding of what is expected of me and I start to complete some of the tasks. I have to wait for my result for my project pitch before I continue further but I hope that I have completed some of the sections efficiently.

I have started a Gantt chart that is a required element of the report on Excel. I have also added all the technical details of the report and can change appropriately if needed.

I have also continued my progress in other modules and the continuous assessments. I hope to have 3 assessments done by next week.

My Achievements

- Began my project proposal report.
    - Filled in all sections that I can before receiving my result.
- Almost complete 3 assessments in other modules
    - Strategic Management -> Reviewing my essay and creating bibliography.
    - Web Services and API -> Testing my application and started written section.
    - Mobile Application and Development -> Adding finishing touches to CA1 of my application.

Issues

- No major issues to report.
    - Waiting to receive my result for the project pitch.
    - Will have to start immediately on project proposal report to deliver on time and to a high standard.

November 2020 Reflective Journal

Student Name: Conor O Reilly

Student Number: x17101701

Programme: BSc (Honours) in Computing – Evening

Supervisor: Vikas Sahni

Week 6

2nd November - 8th November 2020

My Reflection

This week, I had my first meeting with my supervisor Vikas Sahni. He explained that my project pitch was good and that I could continue to write my proposal with my current project idea. He guided me on many aspects of the project and that I should send him a proposal draft for him to review and recommend any changes.

I started my project proposal straight away and have made a massive step in the completion of this report. I hope to have this completed and sent to Vikas by the end of the week so he can have a look and make any necessary adjustments. I will then submit it on Sunday 8th November to be reviewed.

My Achievements

- Started my project proposal -> Sent my draft proposal to Vikas and submitted my final proposal on Moodle
- Completed 2 assessments in another modules
    - Strategic Management -> Submitted CA1.
    - Mobile Application and Development -> Submitted CA2.
- Almost complete one assessment in another module
    - Web Services and API -> Completed application and finishing written section.

Issues

- No major issues to report.
    - Struggled to meet the project proposal deadline but submitted proposal on time.

Week 7

9th November – 15th November 2020

My Reflection

This week, Anu Sahni delivered an interesting lecture about commercialisation. This was very beneficial and provided me with an insight into an area I was not too familiar with.

I researched for various resources that would benefit me for my final year project. I also practiced my prototypes on pen and paper and hope to complete a functional prototype in the coming weeks.

This week is busy with assignments in other modules so I have been focused on completing them on time.

My Achievements

- Completed an assessment in another module
    - Web Services and API -> Completed written section of CA and submitted on time.
- Almost complete one assessment in another module
    - Data Application Development -> Continued my process in this CA. R is a new language for me so I am trying to learn the concepts behind this tool.

Issues

- One issue to report.
    - For the Data Application Development, I am finding it challenging to troubleshoot some code in my project.
    - I had a video call with the lecturer and he helped me to successfully solve the issue and I was able to continue with my project.

Week 8

16th November – 22nd November 2020

My Reflection

This week, we had a seminar about report writing and referencing from Cory Newbigging. I found this seminar to be very beneficial as I forget how important report writing is in our projects and that incorrect referencing can create major repercussions. I take notes on the lecture and will implement this in my future assessments.

I had a meeting with my supervisor this week and we talk about updates to my final year project. He says I may added too much functionality to my proposal and recommends to remove any unnecessary features. He set a deadline for me to send him prototypes for next week for him to review and provide feedback.

My Achievements

- Completed an assessment in another module
    - Data Application Development -> Submitted my assignment on time.
- I start my prototype for me final year project for Vikas to review

Issues

- No issues to report.
    - I hope to finish my prototype by next week and begin a group project in another CA.

Week 9

23rd November – 29th November 2020

My Reflection

This week, we had a lecture about the fundamentals of GitHub. I have been using GitHub since first year so I am quite proficient and use it on a daily basis in college and work. However, it was quite an interesting reminder of certain aspects of the application. I have always struggled to fully understand the concept of branching within GitHub but this lecture was informative and allowed me to fully understand how to implement it efficiently.

It has been a very tough month for assessments in all the modules this semester. I have not been able to dedicate as much time to the final year project as I would have liked too. However, I have completed my prototypes and have sent them to my supervisor for him to review and provide feedback to me next week.

My Achievements

- Completed my draft prototype for my supervisor to review.
- I have continued on other assessments in other modules.
    - Web Services and API -> Started coding our API and began report
    - Artificial Intelligence -> Continuing to make progress in my chess game
    - Mobile Development -> Adding final touches to my mobile application
    - Data Application Development -> Planning CA2

Issues

- No major issues to report
    - I am working hard to finish all my assessments on time but I am following my plan and should have everything completed on time.

December 2020 Reflective Journal

Student Name: Conor O Reilly

Student Number: x17101701

Programme: BSc (Honours) in Computing – Evening

Supervisor: Vikas Sahni

Week 10

30th November - 6th December 2020

My Reflection

This week, we had a quick overview of the deliverables need for the mid term presentation for our software project. I have started my final year report and have begun to set up my Django project.

I have been busy with other projects in other modules so I still have not been able to dedicate as much time as I hoped to the software project. I hope to have a few projects completed by next week.

My Achievements

- I have continued other assessments in other modules.
    - Web Services and API -> Continued coding and report
    - Artificial Intelligence -> Adding final touches to my chess game and starting written question
    - Mobile Development -> Finishing my mobile application. Starting report
    - Data Application Development -> Started CA2

Issues

- No major issues to report.
    - Hope to start the coding aspect of my software project as soon as possible.

Week 11

7th December – 13th December 2020

My Reflection

This week, we have had no official lectures for our software project module. Enda was on standby in case we had any questions.

I have started the coding part to my final year project and hope to have at least a landing page completed for the midpoint presentation.

Other projects in other modules are still ongoing.

My Achievements

- I have continued other assessments in other modules.
    - Web Services and API -> Continued coding and report
    - Artificial Intelligence -> Finished the coding and video for my chess game.
    - Mobile Development -> Finishing my mobile application. Finished report
    - Data Application Development -> Continuing CA2

Issues

- One issue to report.
    - I had an issue with a part of my chess game. A bug was found and was corrupting my app. However, I was able to troubleshoot and submitted my project.

Week 12

14th December – 20th December 2020

My Reflection

This week, we have had no official lectures for our software project module. Enda was on standby in case we had any questions.

I have successfully completed my landing page for my final year project. I hope to include a log in page as well if I have time. I have prepared my presentation and adding my finishing touches to the report.

My Achievements

- I have continued other assessments in other modules.
    - Web Services and API -> Finished coding and report. Conducted presentation and submitted.
    - Data Application Development -> Finished CA2 and submitted.

Issues

- No issues to report.

Week 13

21st December – 27th December 2020

My Reflection

This week, we have had no official lectures for our software project module. Enda was on standby in case we had any questions.

I have completed my presentation for the software project. I have also completed the final year report and submitted all the deliverables.

My Achievements

- I have finished all projects in my other modules.
- I have submitted my presentation and other deliverables for software project.
- We have received our TABAs for 2 modules so I will complete these over the Christmas period.

Issues

- No major issues to report

January 2021 Reflective Journal

Student Name: Conor O Reilly

Student Number: x17101701

Programme: BSc (Honours) in Computing – Evening

Supervisor: Vikas Sahni

Week 1

28th December 2020 – 3rd January 2021

My Reflection

This week, we had no lectures as the term had ended. I have started by TABAs for 2 modules

My Achievements

- I have continued other assessments in other modules.
    - Strategic Management – I have started my essay for the SM TABA
    - Artificial Intelligence – I have started my essay for the AI TABA

Issues

- No major issues to report.

Week 2

4th January 2021 – 10th January 2021

My Reflection

This week, I have 2 due dates for my TABAs. I have made good progress and we have them delivered on time.

I have continued to do work on my final year project. I will make the most of the time off.

My Achievements

- I have continued other assessments in other modules.
    - Strategic Management – I am adding the finishing touches to my SM TABA
    - Artificial Intelligence – I have finished my AI TABA

Issues

- No issues to report.

Week 3

11th January 2021 – 17th January 2021

My Reflection

This week, I have dedicated all my time to my final year project as all Semester 1 deliverables have been submitted. I hope to improve my HTML, CSS, and JavaScript. I hope to also look for Python guides for my project

My Achievements

- All other assignments have been completed.

Issues

- No issues to report.

Week 4

18th January 2021 – 24th January 2021

My Reflection

We had our first lecture on Saturday 23rd January. We received our CA brief and what is expected of us.

I have had more time to make progress on my application. I have created an appealing user interface and will make further advancements later on. I have started to implement Python to my code. I am happy with the progress I have made so far and hope to speak to my supervisor regarding it once the new semester starts.

My Achievements

- All other assignments have been completed.

Issues

- No issues to report.

Week 5

25th January 2021 – 31st January 2021

My Reflection

This was our first full week back for Semester 2. We will have a total of 3 modules and also our final year project. Having 3 modules will allow us to focus on our final year project than in Semester 1.

I have begun to make continued advancements in my final year project and have implemented many features to the project. I have a registration function which will need a few minor adjustments. I also have implemented other features such as adding, editing and deleting a post. All posts now appear on the home page for users to view.

My Achievements

- Usability Design – I have started my CA1 for this module which is a group report.

Issues

- No major issues to report

February 2021 Reflective Journal

Student Name: Conor O Reilly

Student Number: x17101701

Programme: BSc (Honours) in Computing – Evening

Supervisor: Vikas Sahni

Week 6

1st February 2021 – 7th February 2021

My Reflection

This week, I continued to make progress with my final year project. I have added more front-end changes to make the user interface more appealing. I will schedule a meeting with my supervisor to discuss my plans.

We also went over programming frameworks in our last class with Enda. I am using the Django framework for my project, so it was good to learn even more about this and the other frameworks.

My Achievements

- I have continued other assessments in other modules.
    - Usability Design – Continued with group project
    - Cloud Application – Received CA brief
    - Distributed Systems – Received CA brief

Issues

- No major issues to report.

Week 7

8th February 2021 – 14th February 2021

My Reflection

This week, I focused on my other CAs in other modules. I have almost completed my first CA for Usability Design and have started research and preparation for my other 2 projects.

I have met my supervisor to discuss my final year project and my Mid Term result. I was happy with my mid term result but will work to improve. Vikas said I have made good progress and to continue and report back if I have any issues.

My Achievements

- I have continued other assessments in other modules.
    - Usability Design – Finished a first draft of CA1
    - Cloud Application – Planned for my Ruby on Rails application
    - Distributed Systems – Researched possible ideas for my project

Issues

- No issues to report.

Week 8

15th February 2021 – 21st February 2021

My Reflection

This week, I mainly focused on my other modules. I am enjoying these modules especially the Cloud Application module. I have never used Ruby before and it has giving me an important insight into this language and framework.

This week, Enda did not have anything planned for the class. He said that we could work on our project and ask him any questions we may have. I have not been able to focus on my final year project but will work on it as soon as possible.

My Achievements

- I have continued other assessments in other modules.
    - Usability Design – Completed a final draft and ready for submission
    - Cloud Application – Set up a Rails application and added Bootstrap
    - Distributed Systems – Created a GitHub repo

Issues

- No issues to report.

Week 9

22nd February 2021 – 28th February 2021

My Reflection

This week, I have submitted my first CA and started CA2 for Usability Design. I have continued to make progress in the other two modules.

Enda went over testing this week in class. This is a very important concept and is a requirement in our final year project. I researched sources and information that will help me for testing my application.

My Achievements

- I have continued other assessments in other modules.
    - Usability Design – Submitted CA and started CA2
    - Cloud Application – Added more functionality to my project. Sign up/sign in and CRUD functionality
    - Distributed Systems – Submitted my project proposal

Issues

- No issues to report.

March 2021 Reflective Journal

Student Name: Conor O Reilly

Student Number: x17101701

Programme: BSc (Honours) in Computing – Evening

Supervisor: Vikas Sahni

Week 10

1st March 2021 – 7th March 2021

My Reflection

This week, I met with my supervisor for a catch up meeting. He was happy with my progress and said to keep working hard. I have been unable to continue my progress on the final year project as I want to complete the CAs in other modules.

I received feedback from my project proposal for Distributed Systems. My lecturer provided valuable feedback and I understand what is expected of me. I hope to have this project completed in the coming weeks. I have continued to make progress with my other CAs.

My Achievements

- I have continued other assessments in other modules.
    - Usability Design – Continued group work for CA2
    - Cloud Application – Added more functionality to my project. Added more partials and layouts. Added JavaScript code to enhance the project
    - Distributed Systems – Developed a plan on the structure and timeline of my project

Issues

- No major issues to report.

Week 11

8th March 2021 – 14th March 2021

My Reflection

This week we did not have a Software project lecture. We continued with our various projects.

I have finished my part of the Usability Design project and hope to have it complete by the end of the week. We will review our collective work and submit it when we are happy. Continued to make progress in Cloud Application Development and started my Distributed Systems project

My Achievements

- I have continued other assessments in other modules.
    - Usability Design – Finished my part of project
    - Cloud Application – Added more functionality to my project. Added a custom Gem and design patterns
    - Distributed Systems – Set up Eclipse project. Added proto file and began on one service

Issues

- No issues to report.

Week 12

15th March 2021 – 21st March 2021

My Reflection

This week, Enda had said that the Software project class would be predominately for Data Analytics students so I focused on doing project work.

I discussed the group project with my partner and we were happy with the outcome. We submitted the project and began progress on other projects and I hope to continue on the final year project.

My Achievements

- Usability Design – Submitted project
- Cloud Application – Focused on testing this week. Will use this for my final year project as well
- Distributed Systems – Completed all my services and started on the client code. I hope to have this complete next week

Issues

- No issues to report.

Week 13

22nd March 2021 – 28th March 2021

My Reflection

This week we did not have a Software project lecture. We continued with our various projects.

I have almost finished the Distributed Systems project and will submitted it before the deadline this Sunday. I am also adding the finishing touches to my Cloud project. We were giving an extension but I hope to have this complete before the Easter break. This will allow me to focus on the final year project for all of April.

My Achievements

- Cloud Application – Added finishing touches. Began report and will do the presentation on the bank holiday weekend
- Distributed Systems – Finished off the code for the project. I have completed the brief report that is associated with the CA and conducted a presentation. Submitted before the deadline.

Issues

- No issues to report.

April 2021 Reflective Journal

Student Name: Conor O Reilly

Student Number: x17101701

Programme: BSc (Honours) in Computing – Evening

Supervisor: Vikas Sahni

Week 14

29th March 2021 – 4th April 2021

My Reflection

This week we focused on technical report documentation and an overview on the final presentation/demo. This was a good insight into what is expected of us and I feel like I have all the resources and support necessary to complete my project to a high standard. This is our final software project class. We have most of the month to complete our project.

I have now finished all my projects for my final semester. I have handed up my code and report for Cloud Application. I have been able to focus predominantly on my final year project now. I also have a TABA on the 1st May that I will study for

My Achievements

- Cloud Application – Completed. Uploaded code and report/video presentation.

Issues

- No issues to report.

Week 15

5th April 2021 – 11th April 2021

My Reflection

This week I have been able to make great advancements in my project. I have been able to complete my front end design and almost completed my CRUD functionality. I have my sign up/sign in complete and have various permissions enabled. I still have a few features to implement but I am happy with my progress. I have started adding to my documentation and will continue to add to it every day.

My Achievements

- No projects due

Issues

- No issues to report.

Week 16

12th April 2021 – 18th April 2021

My Reflection

This week I migrated my database over to a Postgres database. I initially wanted to implement a Firebase Database but I encountered various issues. I have decided to migrate to a Postgres environment and my supervisor advised that this was ok to do. I have also implemented AWS S3 storages into my project for static files.

I have started to implement a Live Group Chat for my users to communicate easily. I hope to have this completed next week.

My Achievements

- No projects due

Issues

- Issue migrating database to a Firebase Database. After approval from my supervisor, I successfully migrated to a Postgres database.

Week 17

19th April 2021 – 25th April 2021

My Reflection

I have started to revise for my TABA in Distributed Systems on 1st May. I hope to achieve a high result in this module to maintain my high GPA.

I was able to implement a Live Group Chat in my Python project. Users are able to engage in a live conversation.

Unfortunately, I noticed several bugs in my project due to the migration I did last week. I am trying to fix these issues and hope to be error free by next week.

My Achievements

- No projects due

Issues

- Issues regarding the database migration. Some functionality not working correctly such as adding a blog post. Currently testing and trying to debug.

Week 18

26th April 2021 – 2nd May 2021

My Reflection

This week I continued to study for my TABA for Distributed Systems.

I also continued to debug my project from the issues I experienced last week. I am having issues and had a meeting with my supervisor. He showed me potential fixes and we are meeting again to go over this and look over the documentation. He was happy with my progress so far.

My Achievements

- TABA for Distributed Systems on 1st May

Issues

- Ongoing migration issues.

6.3 How to guide

Run Application Locally

- In a terminal, change to the rdpproject directory
- Enter the below command to run the server locally

    - $ python manage.py runserver

How to Register

- Go to Sign Up/Sign In
- Select 'Create a new account'
- Enter the following:
    - A name
    - A unique username
    - A functional email
    - Fully secure password
- Once all credentials met, select Register Account

Note: You will not be able to see full Admin page until permissions have been granted by the administrator.

How to Login

- Once account has been granted permissions, go to Sign Up/Sign In
- Enter the following:
    - Verified Username
    - Verified Password
- If successful, user will be able to view Admin board and engage in Live Group Chat (Local only)

Sign in as Administrator

Username: conororeilly

Password: FinalYearProject2021!

Create a Category

- Sign into your account
- Go to the Admin board
- Select Category -> Add Category
- Enter a unique name into the 'Category Title' field and select Submit

Category will be successfully created

Create a Subcategory

- Sign into your account
- Go to the Admin board
- Select Subcategory -> Add Subcategory
- Enter a unique name into the 'Subcategory Title' field and select a Category from the list
- Select Submit

Subcategory will be successfully created

Create a Post

- Sign into your account
- Go to the Admin board
- Select Posts -> Add Posts
- Enter the following:
    - Post Title
    - Select a Category from the list
    - Summary Text
    - Body Text
    - Select an image
    - Enter as many tags as require (Separate each tag with a comma)

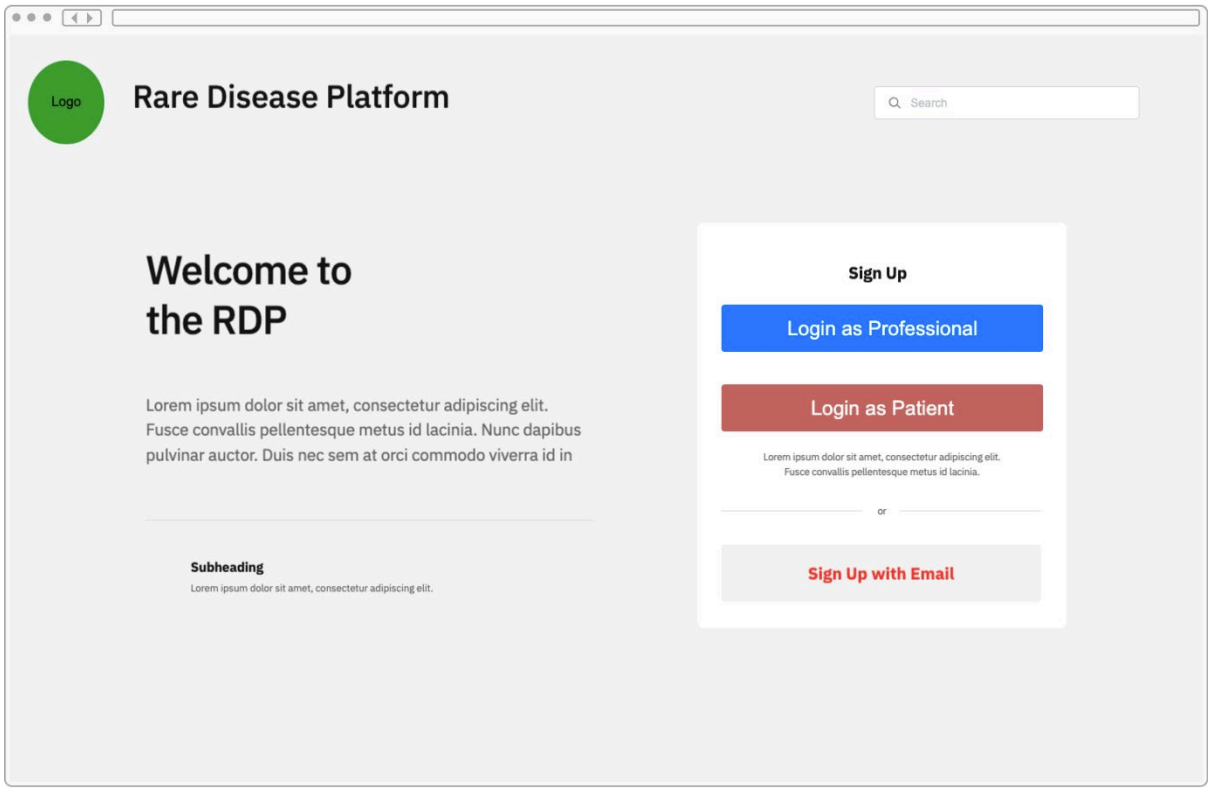
Note: The post will not be displayed on the home page until the admin has selected to publish the post.

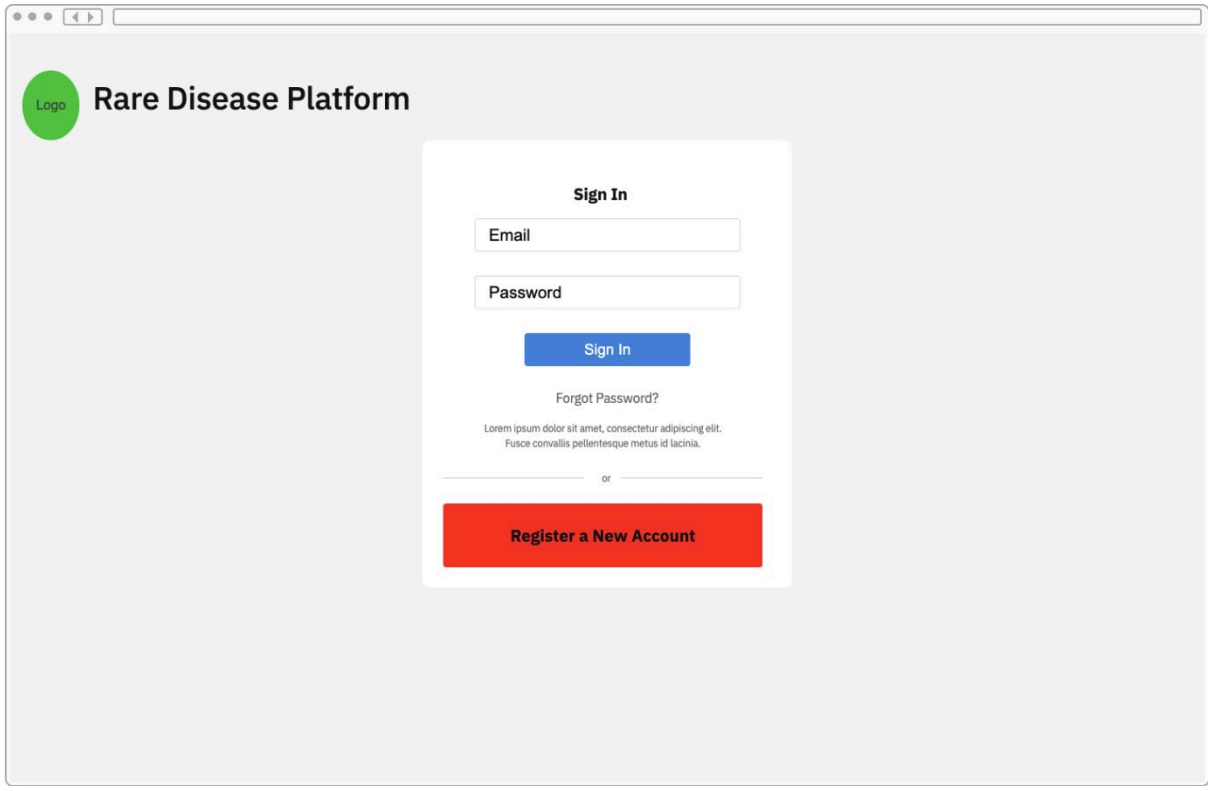
Live Group Chat (Locally Only)

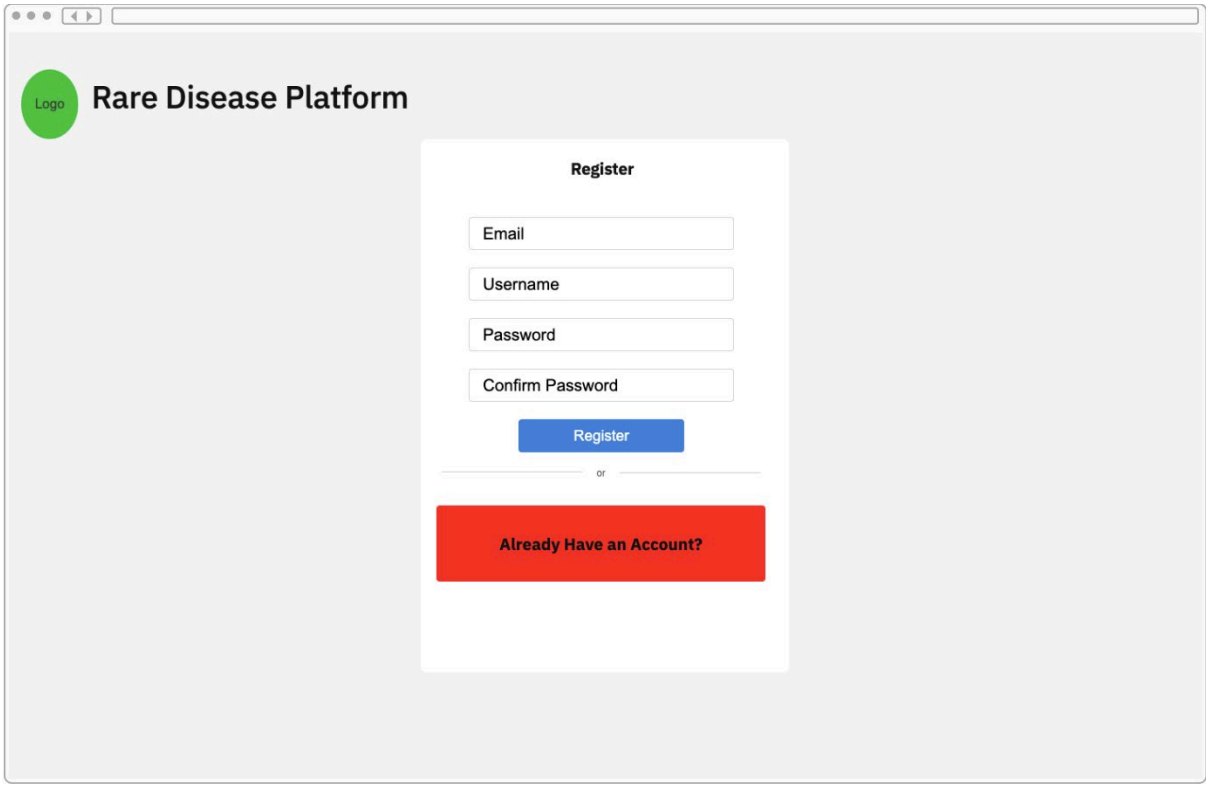
- Once signed in and verified, a user can use the Live Group Chat
- Sign into verified account
- Go to the Home page
- Select 'Live Chat'
- You will now be in a live chat with other verified users (Patients and Healthcare workers)
- Enter a message to communicate with other users or read all the live comments

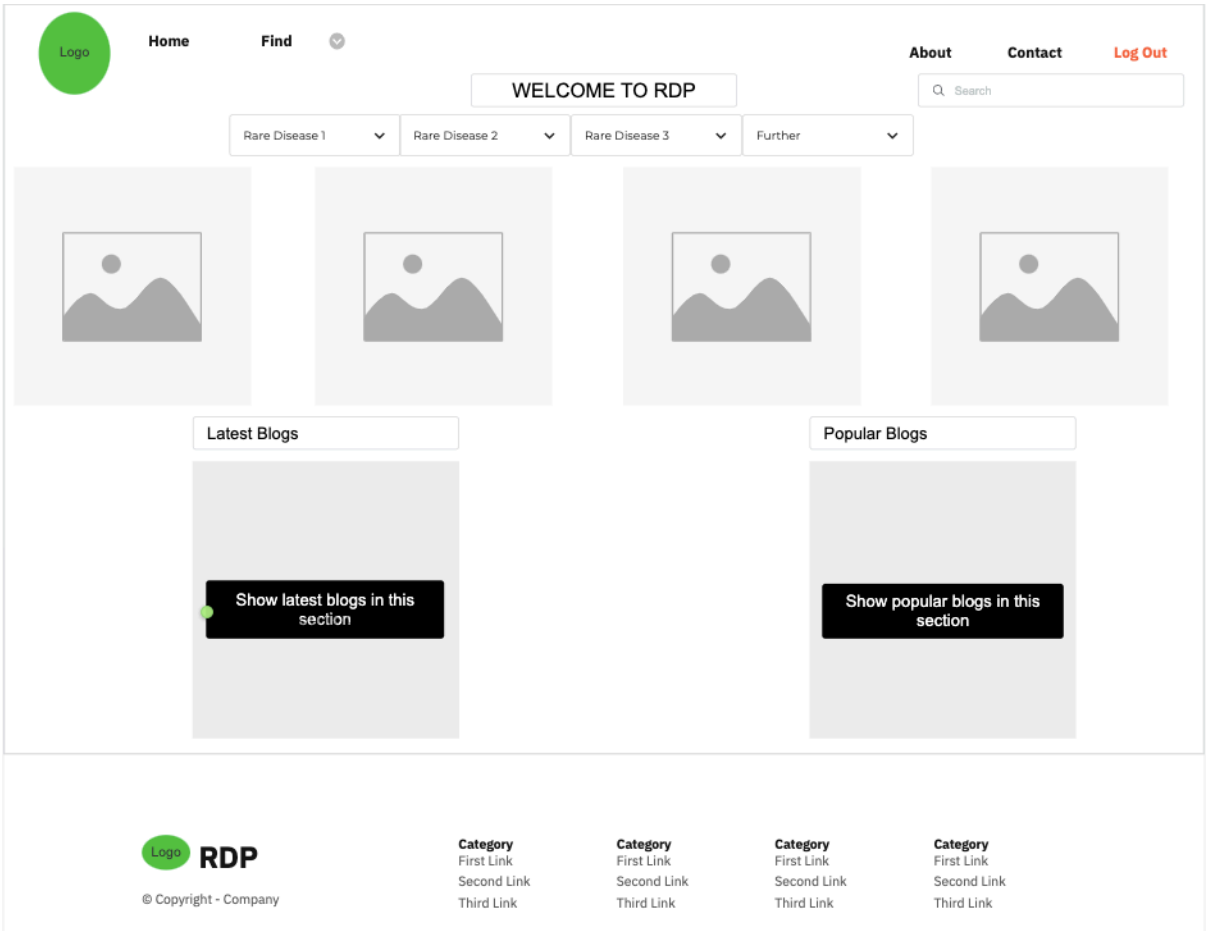
6.4 Other materials used

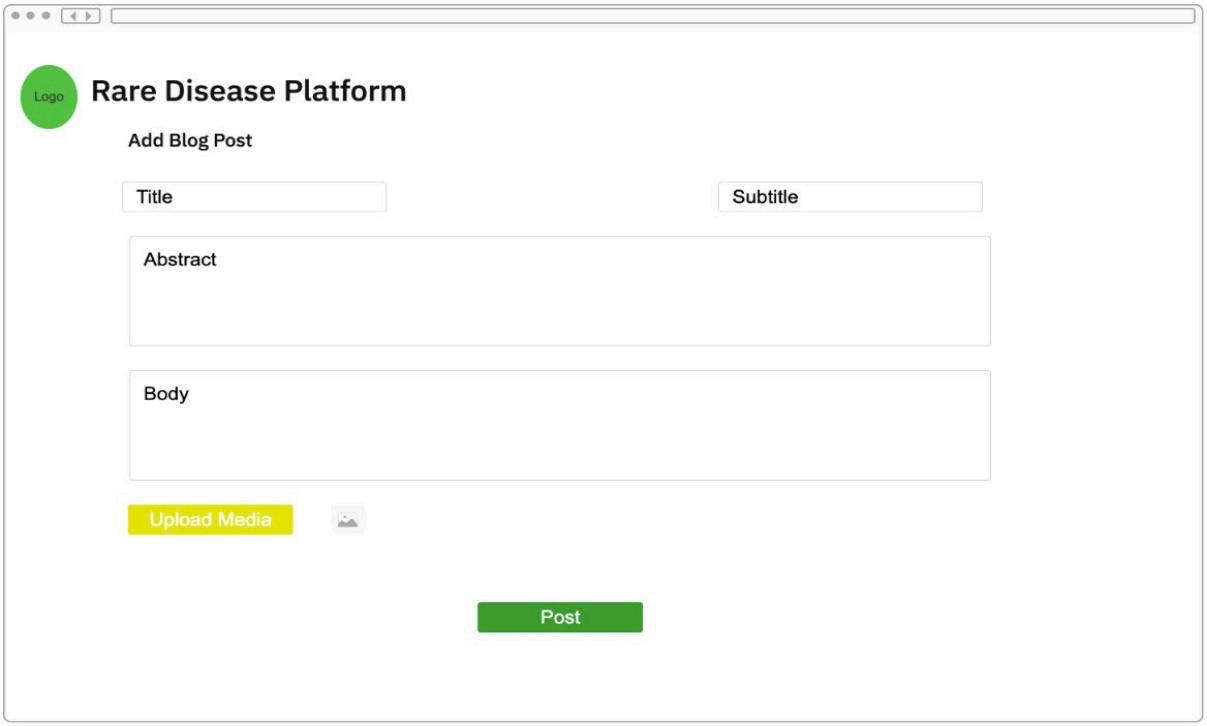
Prototypes

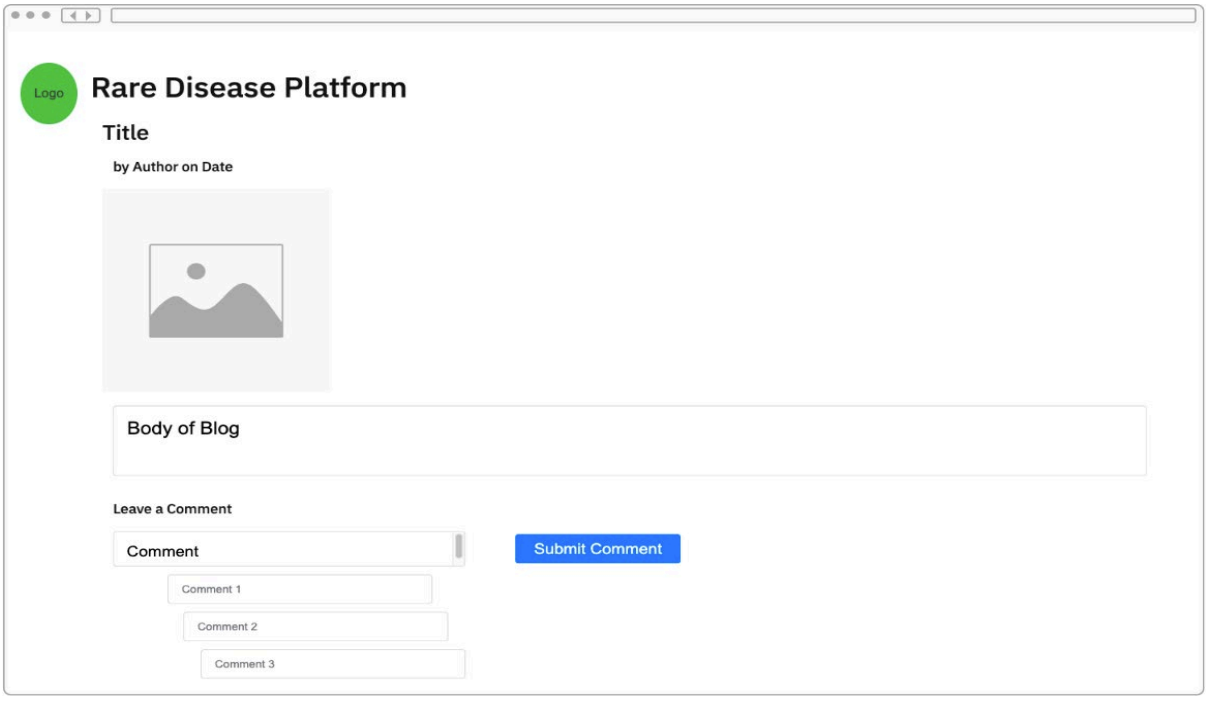


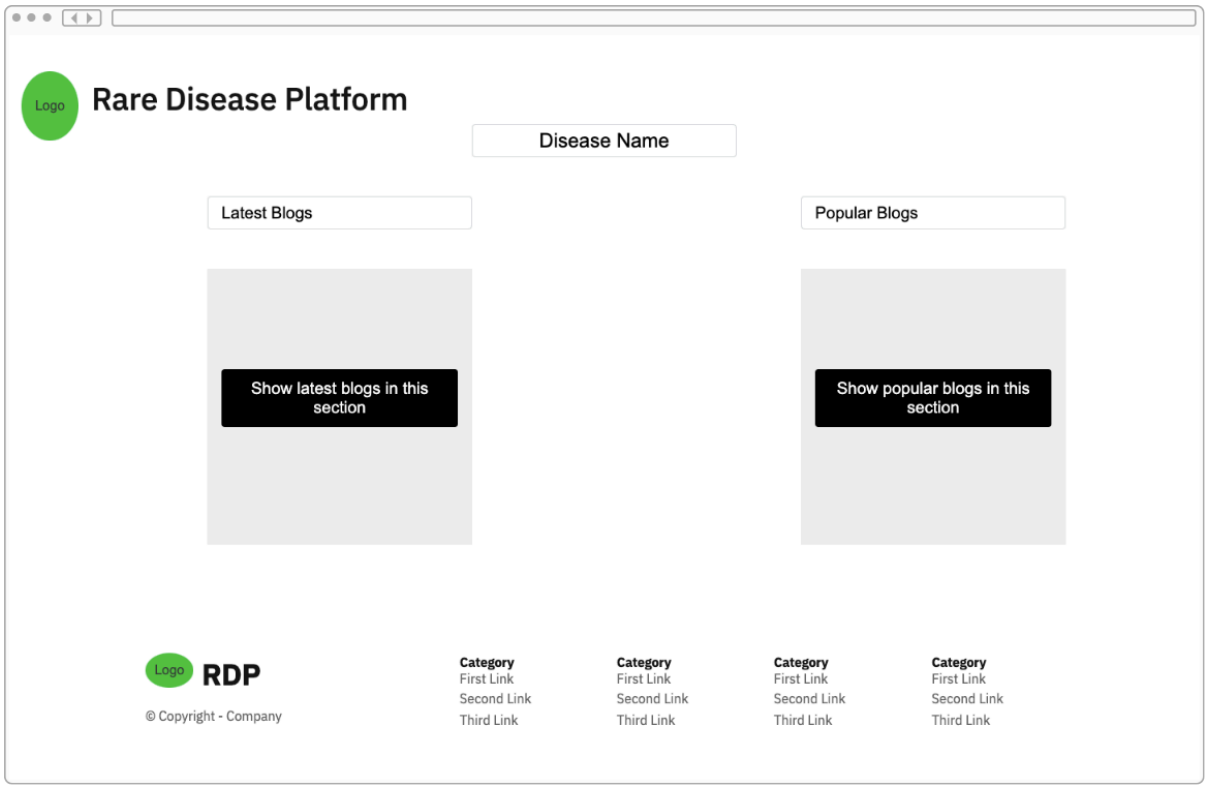



Logo
Rare Disease Platform
Register
Email
Username
Password
Confirm Password
Register
or
Already Have an Account?


Logo
Home
Find
About
Contact
Log Out
WELCOME TO RDP
Search
Rare Disease 1
Rare Disease 2
Rare Disease 3
Further
Latest Blogs
Popular Blogs
Show latest blogs in this section
Show popular blogs in this section
Logo
RDP
© Copyright - Company
Category
First Link
Second Link
Third Link
Category
First Link
Second Link
Third Link
Category
First Link
Second Link
Third Link
Category
First Link
Second Link
Third Link


Logo
Rare Disease Platform
Add Blog Post
Title
Subtitle
Abstract
Body
Upload Media
Post


Logo
Rare Disease Platform
Title
by Author on Date
Body of Blog
Leave a Comment
Comment
Submit Comment
Comment 1
Comment 2
Comment 3


Logo
Rare Disease Platform
Disease Name
Latest Blogs
Popular Blogs
Show latest blogs in this section
Show popular blogs in this section
Logo
RDP
© Copyright - Company
Category
First Link
Second Link
Third Link
Category
First Link
Second Link
Third Link
Category
First Link
Second Link
Third Link
Category
First Link
Second Link
Third Link