Team SteakOverflow

# Major Group Project

Blood Test Diary

## Client:

Dr Marianne Samyn, King's College Hospital

## Team SteakOverflow members:

- Abderraouf Boulahbal - 1754568
- Junyu Chen – 1458450
- Darren Cheong Hao Yuan - 1750315
- Wei Chew - 1750830
- Dillon Malde - 1622726
- Said Mawas Mejdoubi 1749059
- Anjali Raveendran - 1746862
- Katarina Willoch - 1745838

## Tools used:

- Laravel 5.8.2
- Composer 1.8.3
- Apache 2.4.37
- XAMPP 3.2.2
- PHP 7.2.12 VC15
- PHPUnit 7.5.6
- Selenium IDE 3.5.8
- MariaDB 10.1.37
- phpMyAdmin 4.8.3

## Introduction

For our project we have designed a blood test diary for the paediatric liver service at King's College Hospital. The system our client currently uses is inefficient, and outdated, in the sense that they use an excel spreadsheet to track when patients' blood tests are due, chase patients who have not attended their scheduled appointment and chase laboratories for overdue test results. The client wished for the product to make all of these aspects easier.

Therefore, our aim was to build a software application that enables data entry, blood test due dates, automating sending reminders by email, facilitate chasing results, support identification of complex patients. We hope to have achieved this by implementing an efficient web application.

## Outcome

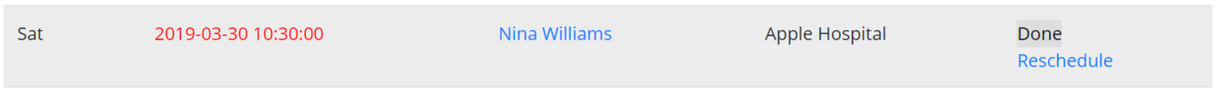
We decided to implement the blood test diary in the form of a web application. The reason that we found a web application better suited for our client is that they only have to install it to access it, in the way that they can access it on any device at any point as long as they have access to internet.

## Important Features

The system can be used by multiple users at the same time to access the database. For example, more than one person can create a new patient at the same time. However, this was not the case in the previous iteration of the excel spreadsheet blood test diary.

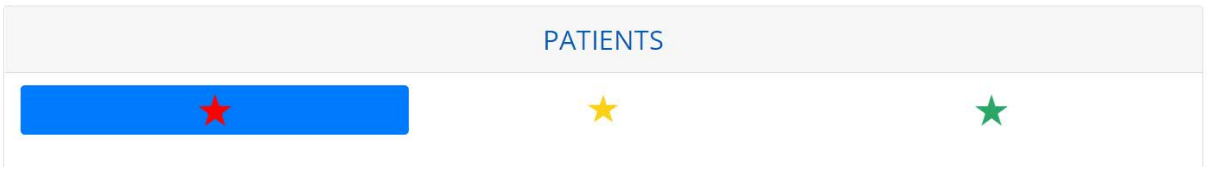
The new system allows for emails to be sent out automatically. For example, if a patient has an upcoming appointment, they will receive an email reminding them. If the patient has not completed a blood test, they will receive a reminder in the form of an email within a week. However, if the patient chooses to reschedule a blood test before the email is sent, the email will not be sent by the system.

As well as automatic email reminders, the system allows for new appointments to be automatically created. The system also allows admins to see whether blood test results have been received by the recipient. This features a color-coded system specifically requested by the client. In this case, each overdue appointment will have its date color-coded in red as shown below.



| | | | | |
|---|---|---|---|---|
| Sat | 2019-03-30 10:30:00 | Nina Williams | Apple Hospital | Done<br>Reschedule |

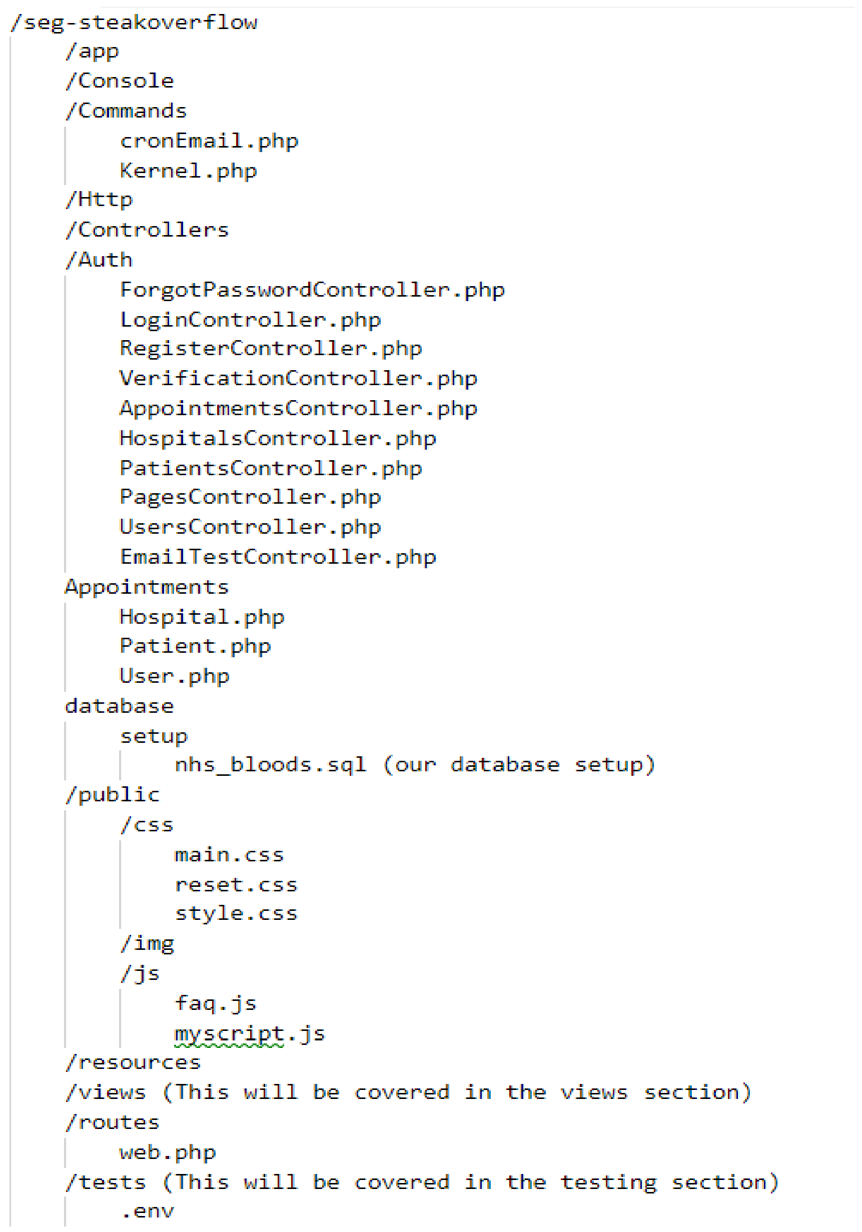
The rating system is used to identify complex patients, as requested by the client. Patients are rated based on how well they respond to appointments and blood test dates. If a patient is listed under the red star, the patient is less or non-responsive to blood test reminders. Likewise, if a patient is listed under the yellow and green stars, they are listed as moderate to very responsive respectively. This allows the admins to efficient direct resources to less responsive patients. The admins are also able to segregate each patient based on the star rating as shown below.



# Design

## Directory structure

The structure below is an abstraction of the key folders that were used for developing the web application, the remaining folders not covered here are just folders required for Laravel to run.



```
/seg-steakoverflow
    /app
    /Console
    /Commands
        cronEmail.php
        Kernel.php
    /Http
    /Controllers
    /Auth
        ForgotPasswordController.php
        LoginController.php
        RegisterController.php
        VerificationController.php
        AppointmentsController.php
        HospitalsController.php
        PatientsController.php
        PagesController.php
        UsersController.php
        EmailTestController.php
    Appointments
        Hospital.php
        Patient.php
        User.php
    database
        setup
            nhs_bloods.sql (our database setup)
    /public
        /css
            main.css
            reset.css
            style.css
        /img
        /js
            faq.js
            myscript.js
    /resources
    /views (This will be covered in the views section)
    /routes
        web.php
    /tests (This will be covered in the testing section)
        .env
```

### cronEmail.php

- Queries the database for patients who have appointments due the next day and sends them a reminder email.

### Kernel.php

- Configures the commands in the commands folder so that it can be called in the terminal. In the terminal, one could use *php artisan command:mail* to run *cronEmail.php*. This allows for easy automation during deployment. For example, on heroku you could set this command to be run daily or hourly, based on the client's preference.

## Models

- Models are stored within the app folder eg: Hospital, Patient, & User.php.
- The models interact with the database using ORM which protects our system from SQL injection.

## Views

- Views are stored in resource/views. These files contain html code which work together with Laravel's Blade to display the pages onto the site.q

## Controllers

- Controllers are stored in app/Http/Controllers, these controllers handle logic and passes data into the views.
- We've decided to use an MVC framework because it allows for modifications to be added easily, making it very maintanable in the future. By separating the logic from the views, it is easier to debug or diagnose errors.

## Web.php

- Routes are defined here and are assigned the web middleware group, which provides features like session state and CSRF protection.

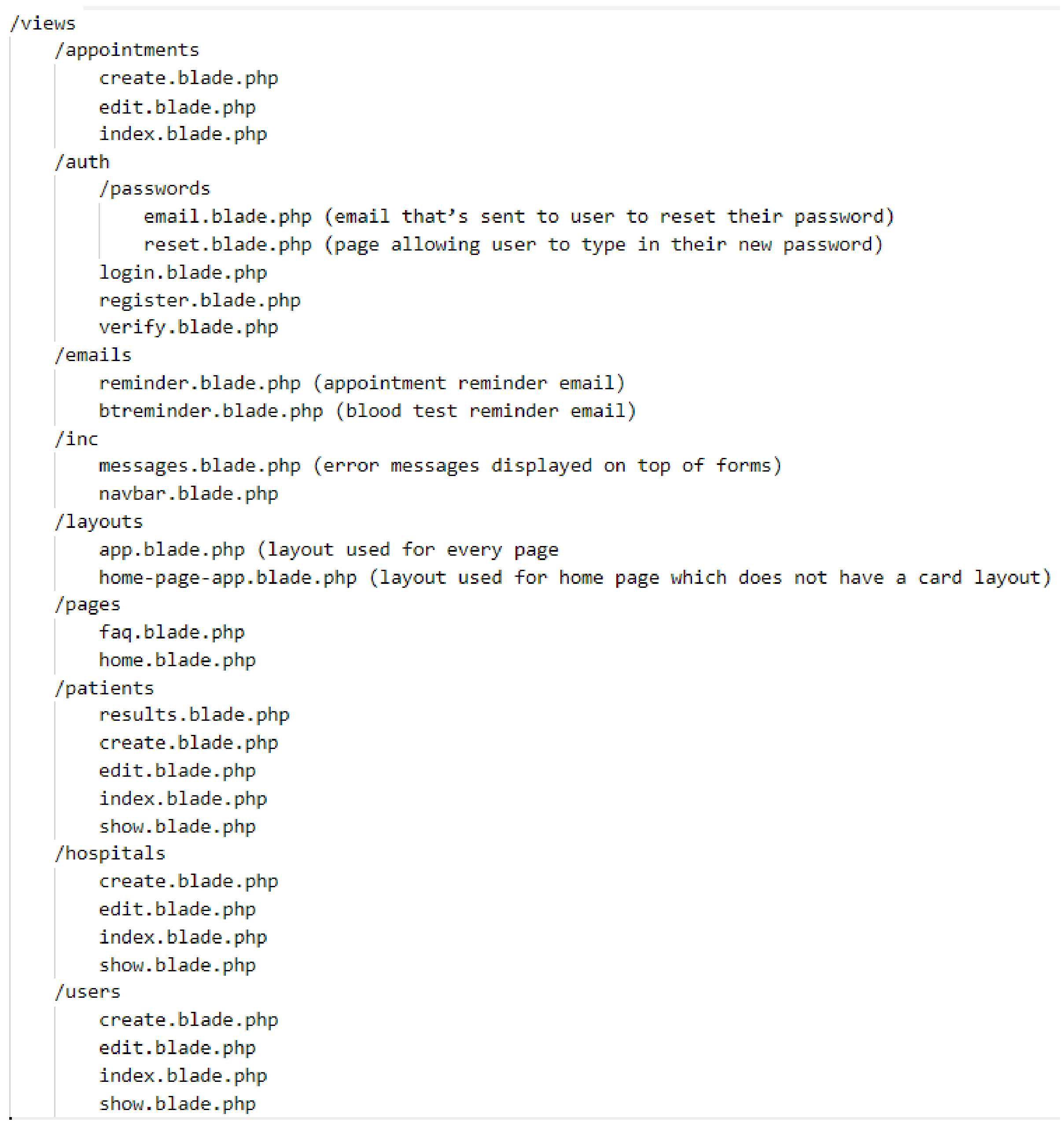
## EmailTestController

- This is used to test if emails were sent to the correct patients. It creates patients which fit the conditions for reminder emails to be sent, and it proceeds to send the emails to these patients.

## .env

- This stores the app, database and mail credentials needed to run the web app. The SMTP is currently set to Mailtrap, please refer to the testing section for more details.

## Views



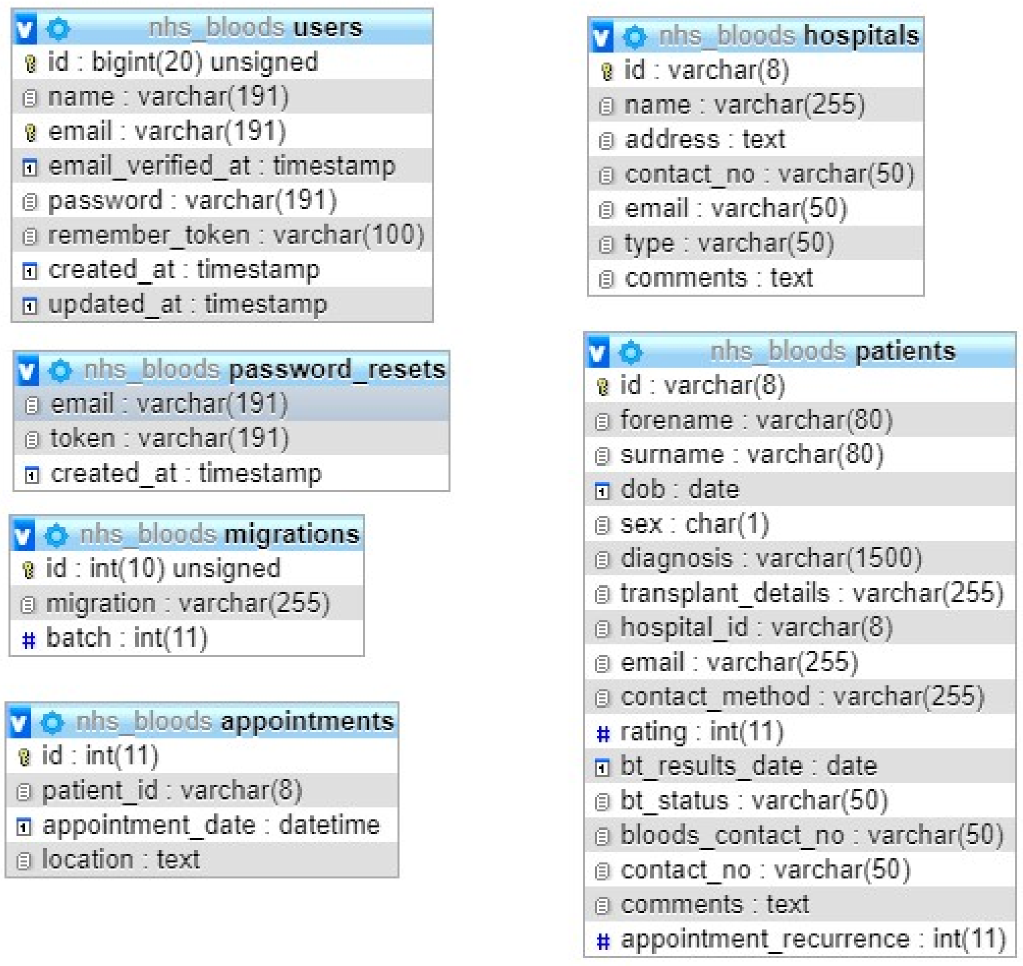
```
/views
    /appointments
        create.blade.php
        edit.blade.php
        index.blade.php
    /auth
        /passwords
            email.blade.php (email that’s sent to user to reset their password)
            reset.blade.php (page allowing user to type in their new password)
        login.blade.php
        register.blade.php
        verify.blade.php
    /emails
        reminder.blade.php (appointment reminder email)
        btreminder.blade.php (blood test reminder email)
    /inc
        messages.blade.php (error messages displayed on top of forms)
        navbar.blade.php
    /layouts
        app.blade.php (layout used for every page
        home-page-app.blade.php (layout used for home page which does not have a card layout)
    /pages
        faq.blade.php
        home.blade.php
    /patients
        results.blade.php
        create.blade.php
        edit.blade.php
        index.blade.php
        show.blade.php
    /hospitals
        create.blade.php
        edit.blade.php
        index.blade.php
        show.blade.php
    /users
        create.blade.php
        edit.blade.php
        index.blade.php
        show.blade.php
```

## View Structure Explanation

- Create.blade.php in the patients folder would refer to the data entry form.
- Edit.blade.php would refer to the edit form.
- Index.blade.php would refer to the page with the table eg: patients/index.blade.php is the patients page
- Show.blade.php would refer to a the patient details form

## Database

Overview of the database:



| nhs_bloods **users** |
|---|
| id : bigint(20) unsigned |
| name : varchar(191) |
| email : varchar(191) |
| email_verified_at : timestamp |
| password : varchar(191) |
| remember_token : varchar(100) |
| created_at : timestamp |
| updated_at : timestamp |

| nhs_bloods **password_resets** |
|---|
| email : varchar(191) |
| token : varchar(191) |
| created_at : timestamp |

| nhs_bloods **migrations** |
|---|
| id : int(10) unsigned |
| migration : varchar(255) |
| batch : int(11) |

| nhs_bloods **appointments** |
|---|
| id : int(11) |
| patient_id : varchar(8) |
| appointment_date : datetime |
| location : text |

| nhs_bloods **hospitals** |
|---|
| id : varchar(8) |
| name : varchar(255) |
| address : text |
| contact_no : varchar(50) |
| email : varchar(50) |
| type : varchar(50) |
| comments : text |

| nhs_bloods **patients** |
|---|
| id : varchar(8) |
| forename : varchar(80) |
| surname : varchar(80) |
| dob : date |
| sex : char(1) |
| diagnosis : varchar(1500) |
| transplant_details : varchar(255) |
| hospital_id : varchar(8) |
| email : varchar(255) |
| contact_method : varchar(255) |
| rating : int(11) |
| bt_results_date : date |
| bt_status : varchar(50) |
| bloods_contact_no : varchar(50) |
| contact_no : varchar(50) |
| comments : text |
| appointment_recurrence : int(11) |

- The hospitals table stores the hospitals, carers and GPs. It distinguishes between them with the types column
- appointment_recurrence is the number of days an appointment would recur, for example if a patient has weekly appointments, it would be set to 7

## Team Organisation

Each team member was not limited to task related only to their group.

### Frontend

- Dillon Malde
- Said Mawas Mejdoubi
- Anjali Raveendran
- Katarina Willoch

### Backend

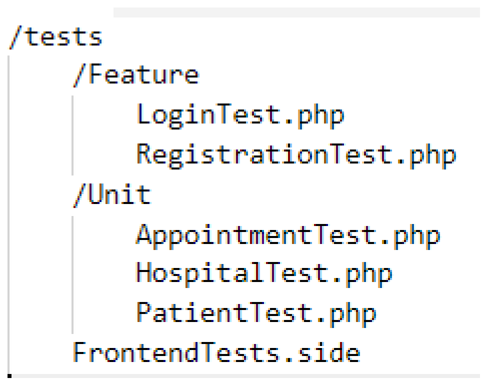
- Abderraouf Boulahbal
- Junyu Chen
- Darren Cheong Hao Yuan
- Wei Chew

## Major Changes

1. The addition of using a framework in our project.
    - Web development can be quite a daunting task, due to the amount of care that had to be provided to each operation involved in the development, deployment and testing of the project at hand. So using a framework not only saves a lot of time by providing premade website necessities, such as authentication, routing and session management, but also eases up the development process and keeps everything in order.
    - The reason why Laravel was used for this project is because it is best for the specific aspects mentioned before while maintaining a powerful usage of superb tools, such as controller containers, expressive migration systems added to an integrated testing system. Everyone in the team had the chance to learn and apply this new feature and build the project we are all proud to present.
    - 
2. Unexpected difficulties
    - While trying to implement the emailing system, the team learnt that this required the deployment of funds, hence the team needed to source a free deployment service that would allow us to send automated emails.
    - The team initially wanted to implement a popup-based navigation system. Popups utilizes JavaScript predominantly, of which the team has little experience with it. It is also worth noting that utilising a popup required the database to be reloaded at every instance. As such, we've decided to use this as the gains did not justify the means.

# Testing

## Tests Folder Structure



```
/tests
    /Feature
        LoginTest.php
        RegistrationTest.php
    /Unit
        AppointmentTest.php
        HospitalTest.php
        PatientTest.php
    FrontendTests.side
```

## Frontend Testing

- The team relied heavily on automating frontend testing using Google Chrome's Selenium IDE extension.
- Frontend tests are located in *tests/FrontendTests.side*, simply load this with the extension and ensure that you are logged out from the site before running all tests.
- For development purposes, we used Mailtrap's SMTP to trap all outgoing emails from our system in its inbox as this would prevent us from spamming actual emails. When this web app is deployed, our clients are free to use whichever SMTP they prefer.
- These are the credentials used for mailtrap should you require it
    - https://mailtrap.io/
    - Email: weisheng3725@gmail.com
    - Password : test1234!

## Backend Testing

Backend testing was done using PHPunit. The team was careful to ensure that all features and its interaction with database were tested. PHPunit was especially useful to test our database since we have modified it several times.

It was worth noting that the email automation systems were tested on the frontend testing system, as it was complicated to test through backend testing.

## Conclusion

In terms of key learning points, the team's understanding of frameworks and php has increased exponentially. A significant portion of the project relied on understanding the frameworks used (Laravel & Selenium) and how to alter the structure of our code based. In the beginning, the strengths and weaknesses of each member of the team was clearly defined to prevent unnecessary delays in task allocation. This allowed us to work on both the front-end and back-end systems simultaneously.

Furthermore, while we've enjoyed liaising with the Client Dr. Marianne, the team has often found her requests somewhat difficult to accomplish. This is mainly due to the team's fresh technical expertise and a lack of understanding on the technical aspect of programming from the client's perspective. Nevertheless, the team was able to implement most of the core features that were detailed in earlier parts of this report. Overall, it was an exciting learning experience, particularly dealing with a real-life client and certainly opens doors for future opportunities for the team.

Ultimately, the team set out to prioritize functionality and security while maintaining a clean and simplified graphical user interface. This is to ensure that the end product requires little to no technical experience, while ensuring that future staff members are also able to easily access the blood test diary. To further ensure that each staff member understands the functions of each feature, the team has also included video tutorials (Uploaded on Youtube and unlisted) for the perusal of staff members.

Lastly, the team members of SteakOverflow convey our sincere thanks to Jeroen, Brendan and Dr. Marianne for a challenging yet exciting module, and we look forward to your feedback.

## Deployed Version

https://nhs-bloods-2019.herokuapp.com/home

Email: test@gmail.com
Password: test1234!