Design Document

"Corona Rapid Test Data Web Platform"

FONTYS UNIVERSITY OF APPLIED SCIENCES

HBO-ICT: English Stream

Data: October 22, 2020

Name: Zhicheng Yu

Student number: 2626845

Table of content

Introduction

In this document, I will discuss in detail the steps or techniques used to complete the Corona Rapid Test Data Web Platform.

The function of the platform is that researchers can send registration requests to administrators. The notification will be sent to the administrator via E-mail. After the administrator has verified that all information is correct, the administrator can create a new account for researchers. The system will also send username and password to researchers via email. The system works this way because not everyone can sign up for an account, only administrator can create accounts. What is more, The researcher cannot be confirmed as a researcher until he or she has been authenticated by administrator. Later on, researchers can log in or log out and if it is necessary, they also can reset their password. They also can access Corona Rapid Test Shared Data which comes from remote MYSQL database.

It is essential to describe this document well because it makes it easier for the readers to understand the information quickly and easily. It is broken down into technical parts. The first part gives the user experience design of the website. The second part includes C4 model for visualizing software architecture. The third part includes the sequence diagrams explain the type of interaction diagram because it describes how and in what order a group of objects works together. Lastly, I will introduce the security section of the web platform.

UX Design

Registration page:

The registration page is for every researcher who would like to be a member of Corona Rapid Test Data platform, the researcher must provide their email firstly. Researcher needs to click the link to verify researcher's email. Because of only registration is allowed through official email extension from the school e.g. @fontys.nl. Researcher will receive verification link in his email. Then researcher has to click that link. If the email is correct, the interface will jump to the website page and show that the authentication has been successful. If the email has been registered, the researcher will be prompted that you have registered and failed to register. After that, the researcher must provide his detail information (First name, last name, phone number, company/Institution name, URL, and position) in order to ask for an account. After that, all the detailed information will send to the administrator via email. The administrator will verify against this data.

Click the following link to check the register page in detail:
https://app.diagrams.net/#G1MATCAbvxSs2MAf16w0HxXFpSxP7SpAHO

Login page:

The Login page is for every researcher and the administrator who would like to access Corona Rapid Test Data platform, a researcher and the administrator must provide their username and password. After clicking the submit button, a researcher will receive verification link in his email. Researcher has to click that link to login in the web platform. The link 'forgot password' can help researcher to reset his password.

Click the following link to check the login page in detail:
https://app.diagrams.net/#G1rVbOSqm2SmzbF27Av4yQXSOyv0h_9Xzh

Admin page:

The admin page appears only after the administrator logs in to the web platform. After the administrator confirms that the researcher's identity is correct. The administrator must provide (username, researcher's email, and password) in order to create a new account for the researcher. The password must comply with 8-20 characters, 1 number, 1 uppercase letter and 1 lowercase letter. After that, system will send the username and password via email to the researcher.

Click the following link to check the admin page in detail:
https://app.diagrams.net/#G15uhtBPzMv8TldF5ldbn9gO8peZYR-iSF

Home page:

After the researcher logs in, it goes to the Home page. Research will get a menu card with all available data types to select. Due to the project is to create an open standard and platform for the exchange of this type of privacy-sensitive data. Not only Corona test data is available. But the Corona test app is the first application.
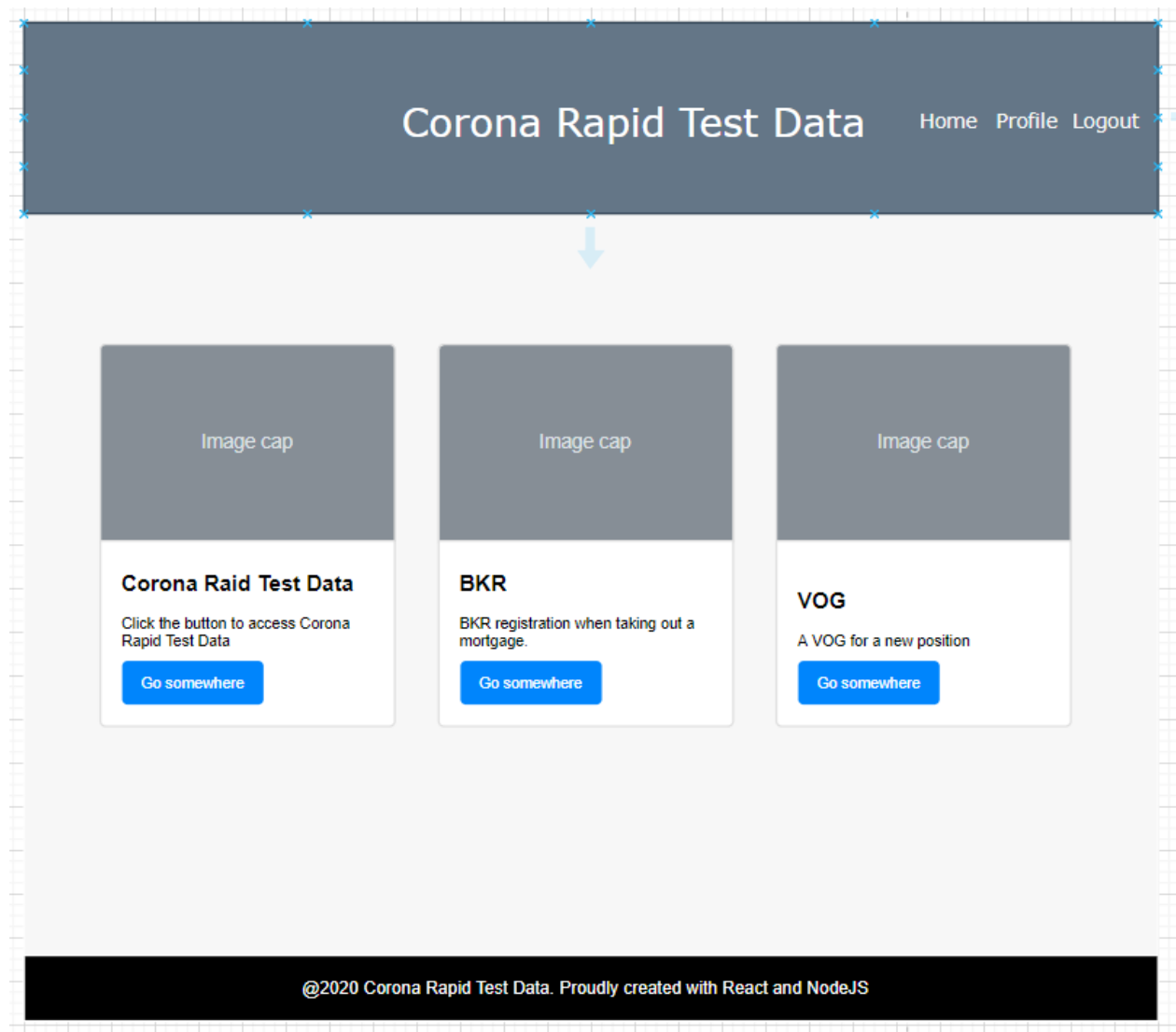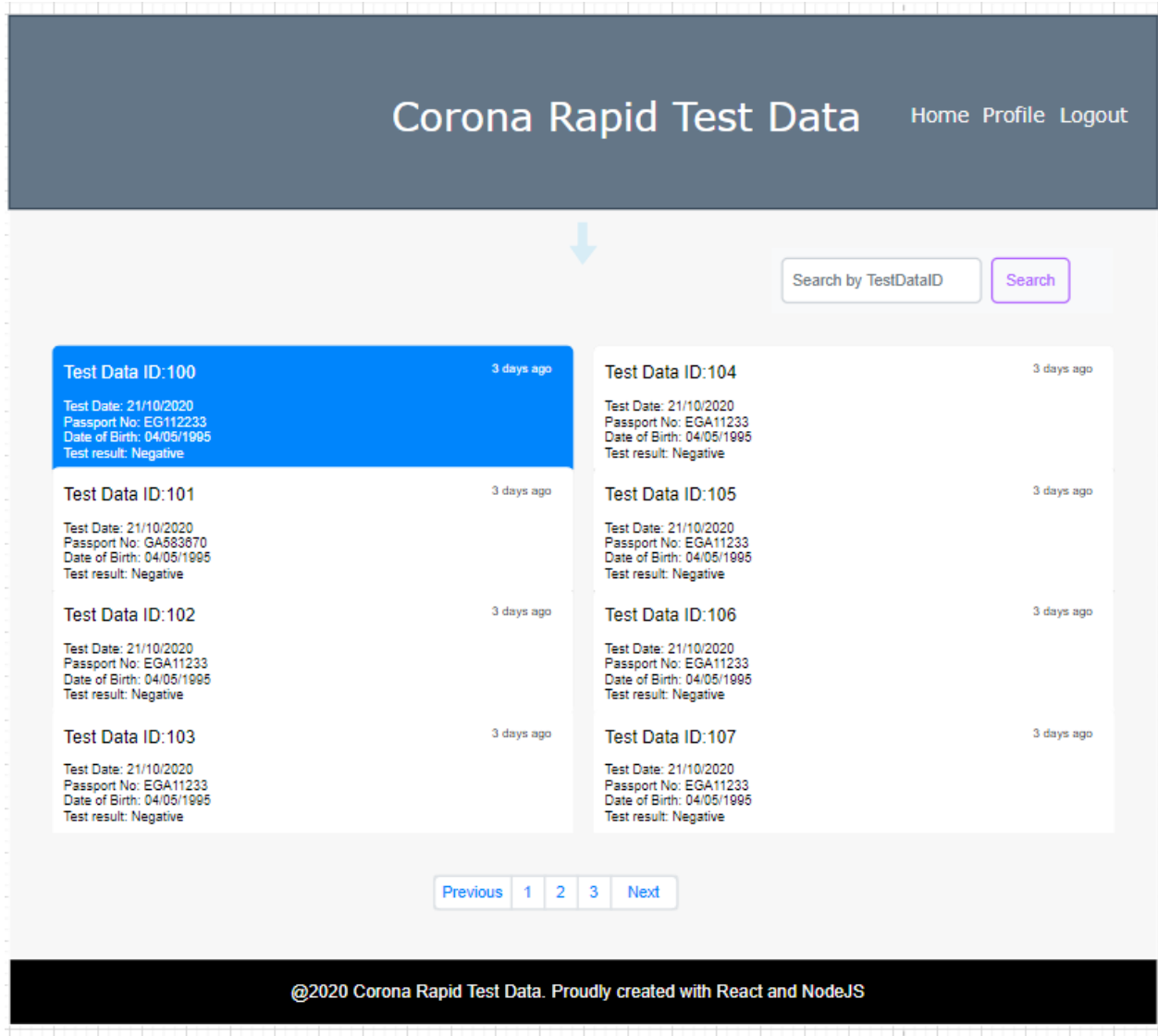

Click the following link to check the home page in detail:
https://app.diagrams.net/#G1IubIVBgUSK0CBaN6j2Dzom1VLttNbmC8

Data page:

After the researcher selects the menu card 'Corona Rapid Test Data'. All posted Corona Rapid Test data will be shown on the data page. Researchers only can browser these data and search by TestDataID to access the specific data. Researchers unable to edit or delete any test data.
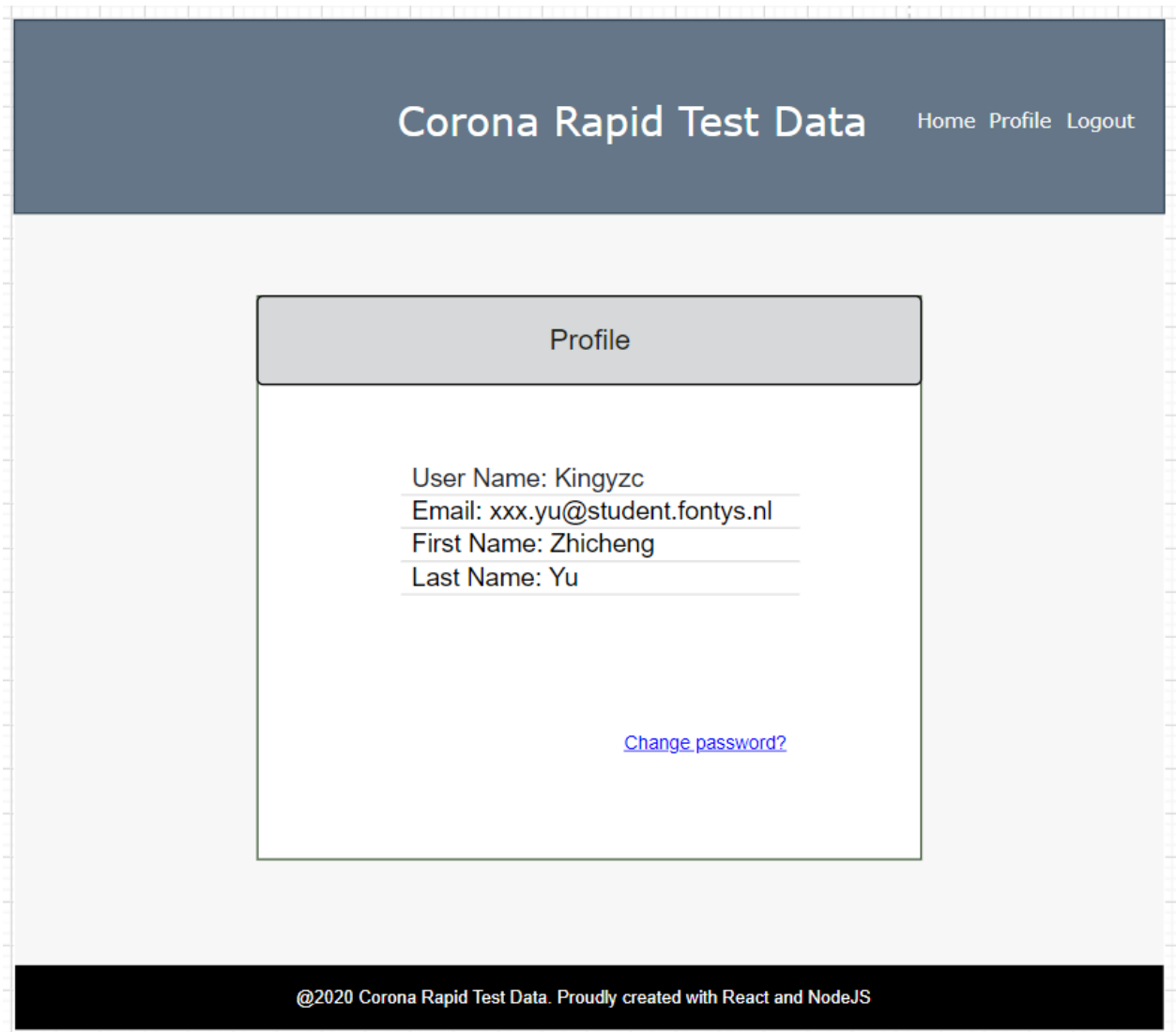

Click the following link to check the data page in detail:
https://app.diagrams.net/#G1j51tZTm85OmSG5Xk9_32BNVm3fZnz-vL

Profile page

The profile page contains the detail information about the researcher. Researcher is able to change the password in profile page. After the researcher clicks the link to change the password, his email will receive a link to change his password. Click the link and redirect to the web platform to change your password.

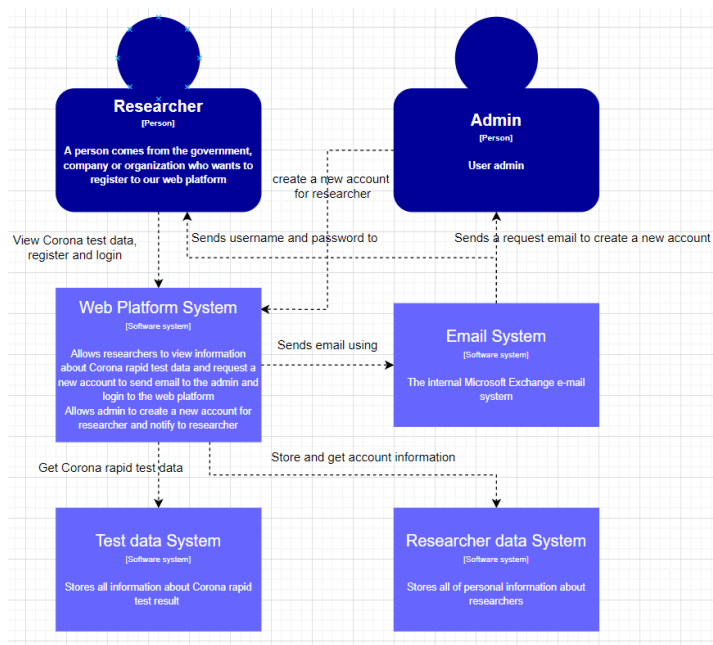Click the following link to check the profile page in detail:
https://app.diagrams.net/#G1vHgrv7xZPGj1VPpUt4fvUoBKn6d-ysM2

Design pattern

I am using the observer design pattern(https://www.freecodecamp.org/news/4-design-patterns-to-use-in-web-development/). The subject holds all of the data and the state of that data. Then I have observers, like different components, that will get that data from the subject when the data has been updated. The goal of the observer design pattern is to create this one-to-many relationship between the subject and all of the observers waiting for data so they can be updated. DataStream service processes updates by streaming data to observers. So anytime the state of the subject changes, all of the observers will be notified and updated instantly.

C4 Model

Level 1: System context diagram for Corona rapid test data system

The system context diagram is the highest level in a data flow diagram and contains only one process, representing the entire system, which establishes the context and boundaries of the system to be modeled.(https://online.visual-paradigm.com/knowledge/system-context-diagram/what-is-system-context-diagram/)

This diagram shows the people who use it, and other software systems that the Corona rapid test data system has a relationship with. Researcher use the Corona rapid test data system to view information about each person's test result and register an account. The administrator uses this system to create an account for the researcher. The system uses Microsoft Exchange email system to send email to researchers and the administrator. Limiting only one email system option is better for the administrator to manage.
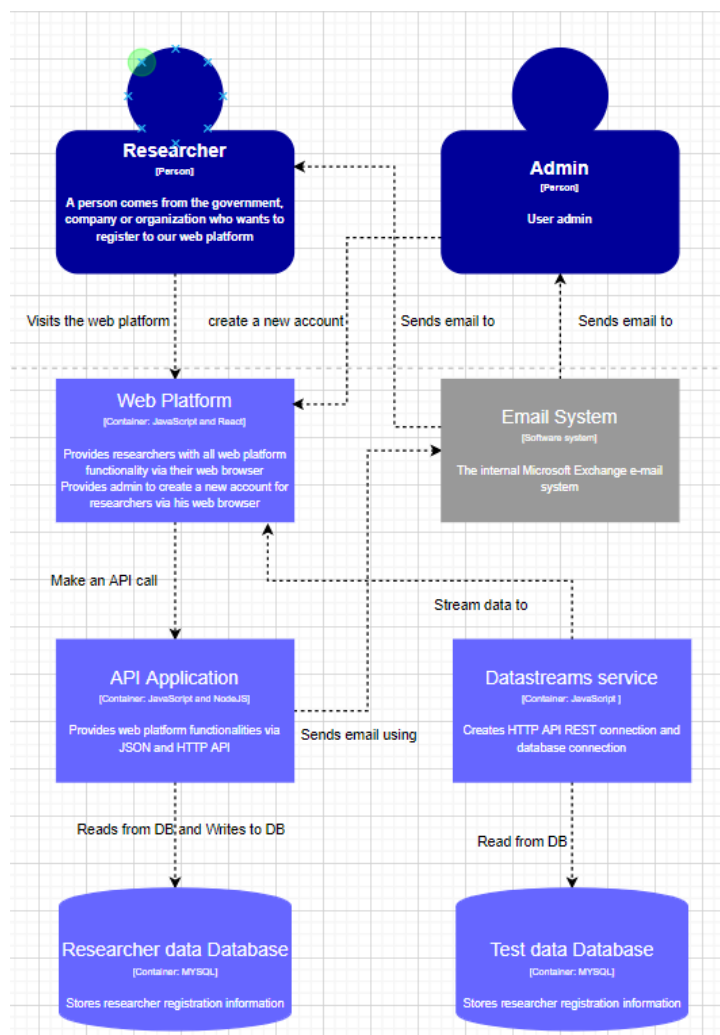

Click the following link to check the system context diagram in detail:
https://app.diagrams.net/#G1Oxt529Vah_Z9-ILhzvHmEOcxQ5A_SQ7i

Level 2: Container diagram for Corona rapid test data system

The Container diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it. It also shows the major technology choices and how the containers communicate with one another. (https://c4model.com/)

This diagram shows that the Corona rapid test data system is made up of five containers: a web platform, a server-side API application, DataStreams service, and two databases. The web platform is a React application that runs in the researcher's web browser, providing all features of the Corona rapid test data system. The API application gets researcher's information from the database. The API application also uses the existing E-mail system if it needs to send e-mails to researcher. The DataStreams capture the data from the other database and stream the data to web platform.
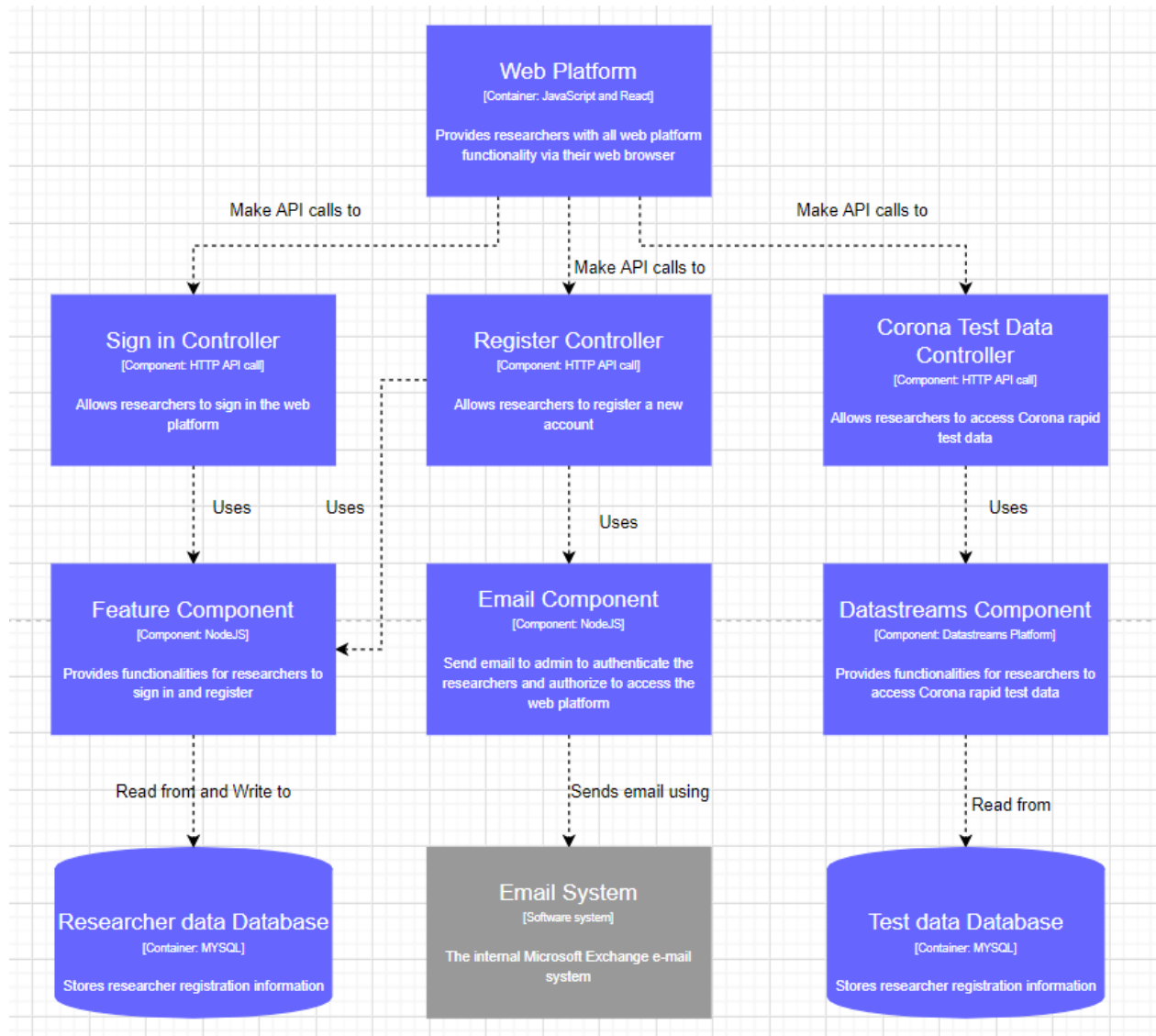

Click the following link to check the container diagram in detail:
https://app.diagrams.net/#G1Oxt529Vah_Z9-ILhzvHmEOcxQ5A_SQ7i

Level 3: Component diagram for Corona rapid test data system

Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.(https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/)

This diagram shows all of the components within the API application. There are three controllers providing access points for the JSON/HTTPS API, with each controller subsequently using other components to access data from database, sign in, register, and send emails.
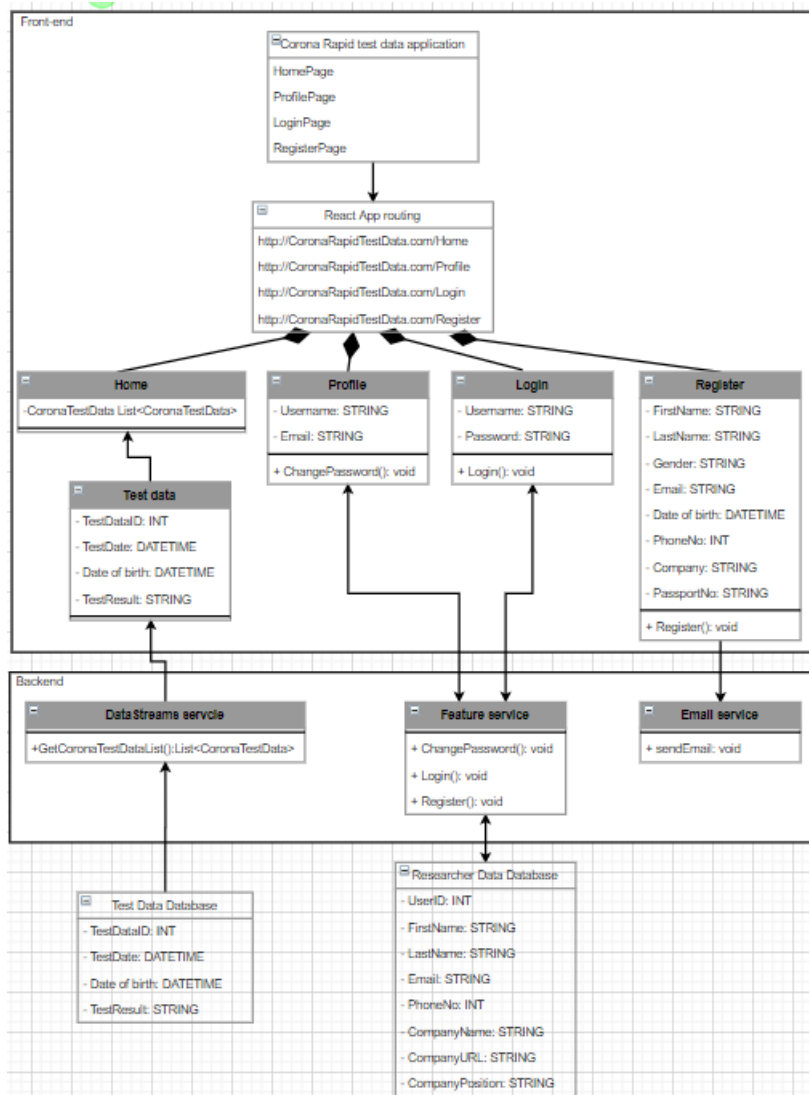

Click the following link to check the component diagram in detail:
https://app.diagrams.net/#G1Oxt529Vah_Z9-ILhzvHmEOcxQ5A_SQ7i

Level 4: UML diagram for Corona rapid test data system

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.(https://tallyfy.com/uml-diagram/)

This diagram shows code elements that make up all components. It shows that the component is made up of a number of classes, with the implementation details directly reflecting the code.
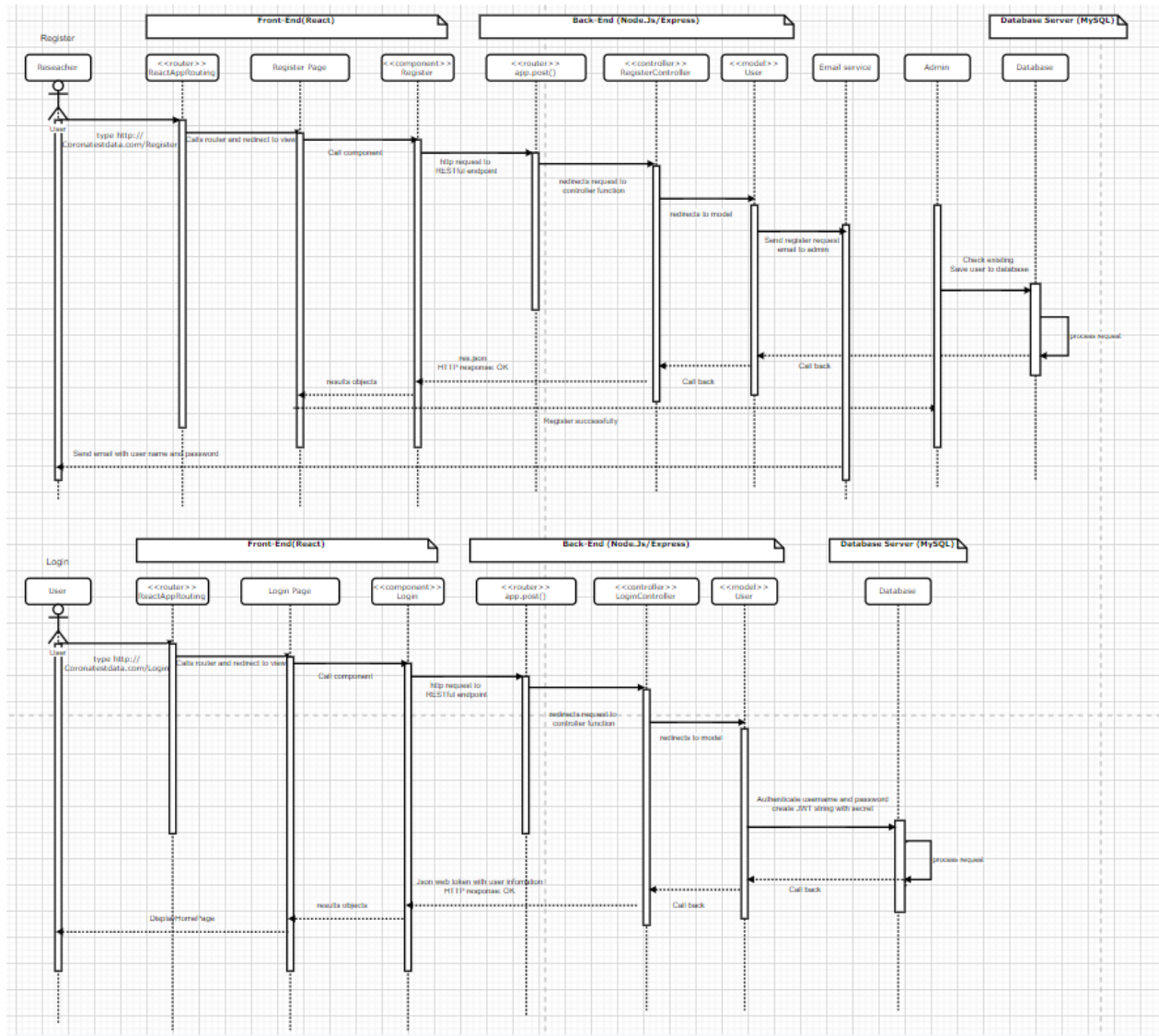

Click the following link to check the UML diagram in detail:
https://app.diagrams.net/#G1Oxt529Vah_Z9-ILhzvHmEOcxQ5A_SQ7i
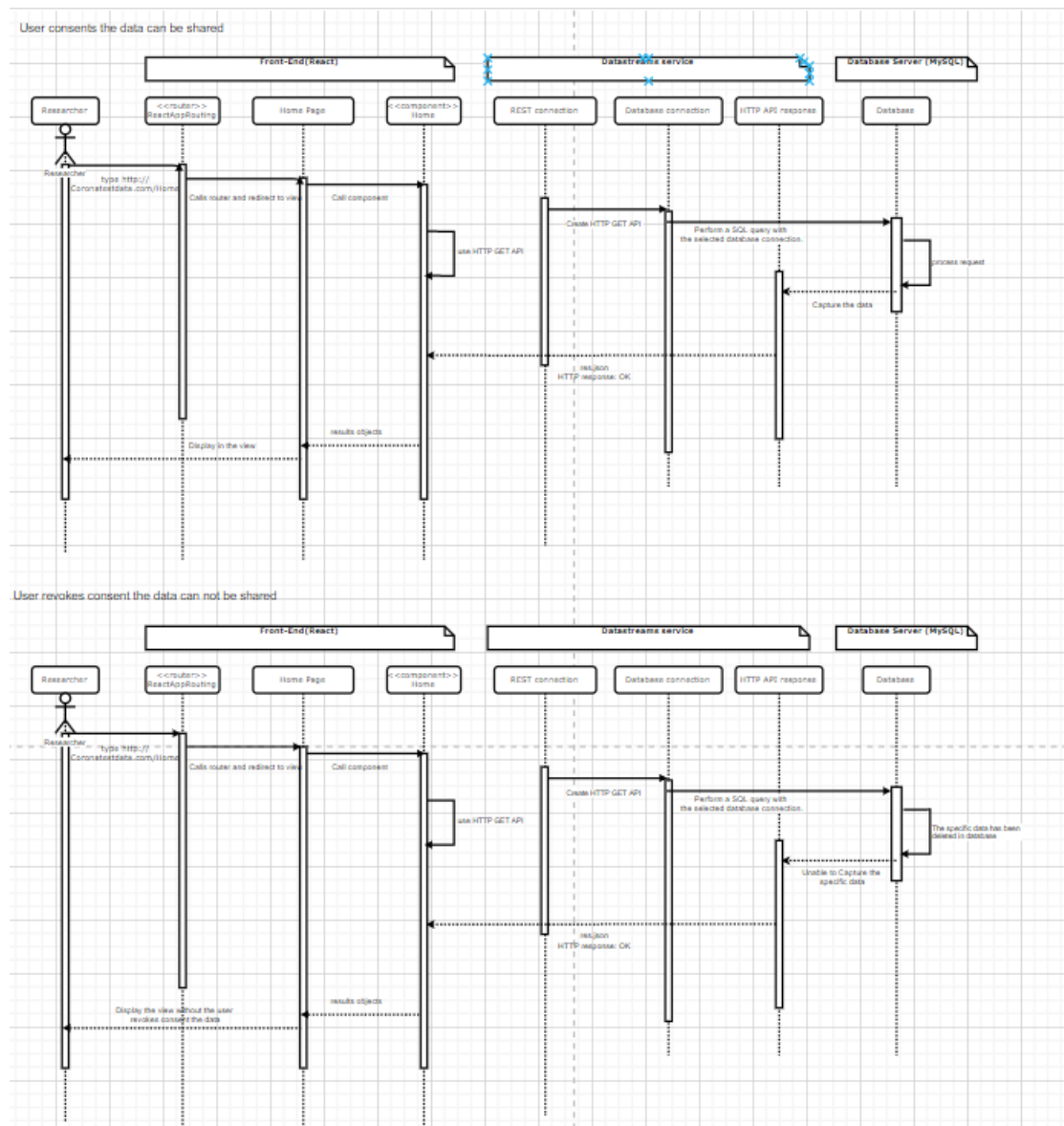
Sequence diagram

Authentication and authorization for researchers

Click the following link to check the sequence diagram in detail:
https://app.diagrams.net/#G1ld4EXi5HdwU2uzMe32ium8thnMxH82N2

Data updates after user(tester) consents/revokes consent to share Corona rapid test data

Click the following link to check the sequence diagram in detail:

https://app.diagrams.net/#G1xMaIzAf_35n4VJHBHwUpNgkvK_YQTxTB

Security

The web platform should be protected at two different layers (https://medium.com/@SilentHackz/simple-way-to-secure-react-apps-using-jwt-and-react-router-2b4a05d780a3). At the API layer, the data that the researcher can read. At the React layer, the paths of the React application that the researcher can access. So, the web platform will be secured by using JWT to secure API layer and React Router to secure React layer. On the client side, JWT will be stored in cookies. Because cookies are managed by the browser, they require less work on the application and can be more secure. So, The token will be managed at the browser level, so it makes it more difficult to create a bug that can potentially open some security vulnerability. About React Router, by using React Router Redirect component along with conditional rendering to easily implement protected routes in my React app.

For the back-end development, I decide to use NodeJS (https://www.veracode.com/blog/secure-development/how-secure-are-popular-web-frameworks-here-comparison). NodeJS does ship with some security pieces by default. For instance, it comes with the ability to start up a web application using HTTPS to protect your data from exposure. There is also a strong crypto library so you can protect your data at rest.

The next thing I need to protect is my database. All the data will store in Azure database (https://docs.microsoft.com/en-us/azure/mysql/concepts-security) for MYSQL server. There are multiple layers of security that are available to protect the data on the Azure Database for MySQL server. In addition, I will use NodeJS to store passwords on the database by hashing and salting (https://www.bigomega.dev/passwords-in-node). The salt is generated randomly every time a new password is stored on the database. The salt is a long and random string which increases the cost of doing a brute force attack.

For the mailing system, I decide to use Microsoft outlook. It can help users lower the risk of accidental or malicious data loss by making it easier for users to protect and read sensitive emails.