CS 319 Term Project

PHS: Health Center Management App

Final Report

Alper Kandemir 21703062

Bora Fenari Köstem 21801823

Oğulcan Karakollukçu 21802642

Instructor: Eray Tüzün

Teaching Assistant(s): Elgun Jabrayilzade and Muhammad Umair Ahmed

May 9, 2022

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Object-Oriented Software Engineering course CS319.

1. Introduction

When the system is first installed, it automatically creates an admin account and requests a password change at the first login. Hospital staff or patient accounts can only be created by the admin. After logging in to the created patient account, you can view some of the profile information on the left of the first screen, view the news in the middle of the screen, and create an appointment from the appropriate appointment options just below. created appointments can be viewed in chronological order on the appointment tab. Another feature, the health history viewing feature, starts with the doctor's request for a test for that patient, and then becomes active after the nurse or doctor uploads the pdf containing the patient's health record. On the other hand, all records are kept here in pdf format and can be downloaded by the patient upon request. The doctor can create working hours from the setting tab, then the created working hours can be updated. Appointment options are presented to patients according to the established working hours. Appointments made are presented to the doctor in the appointment tab, with the closest time at the beginning, and the doctor can accept patients from this screen. The secretary can make an appointment for each patient. He can see all created appointments and has the authority to cancel them. Admin can view, download and upload new health histories of all users. Moreover, it can view all appointments.

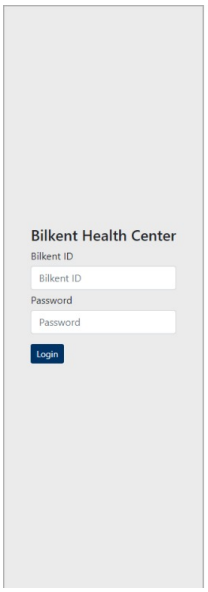
2. Lessons Learned

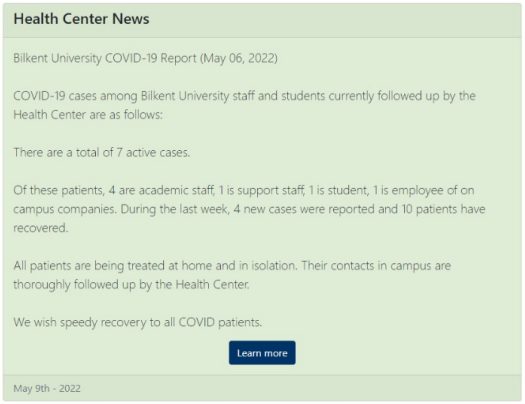
This project has helped us with our coordination skills since it needs to be done by a group. Initially, the meetings were face-to-face and then we continued online. At the very last parts of the project, we gathered in our friend’s house to merge the branches. After giving everyone their roles, we used GitHub for creating branches. Everyone has uploaded their codes to their branch. At first, we have divided the group by three-to-two and the plan was for our two foreign friends to code frontend but they didn’t even notify us of anything they have done; in fact they haven’t done any. Therefore we have completed the backend and frontend all by ourselves. This taught us to push our limits and get work done in a short period of time considering our lack of group friends which led us to work more hours and eventually made us feeling unwell. We have learned a lot about modelling and implementing and also about the PHP Laravel framework. We have struggled with the implementation of many details we

had in mind due to our low number of group members because the process in the frontend and backend took time. Overall, this project improved in many technical aspects.

3. User's Guide

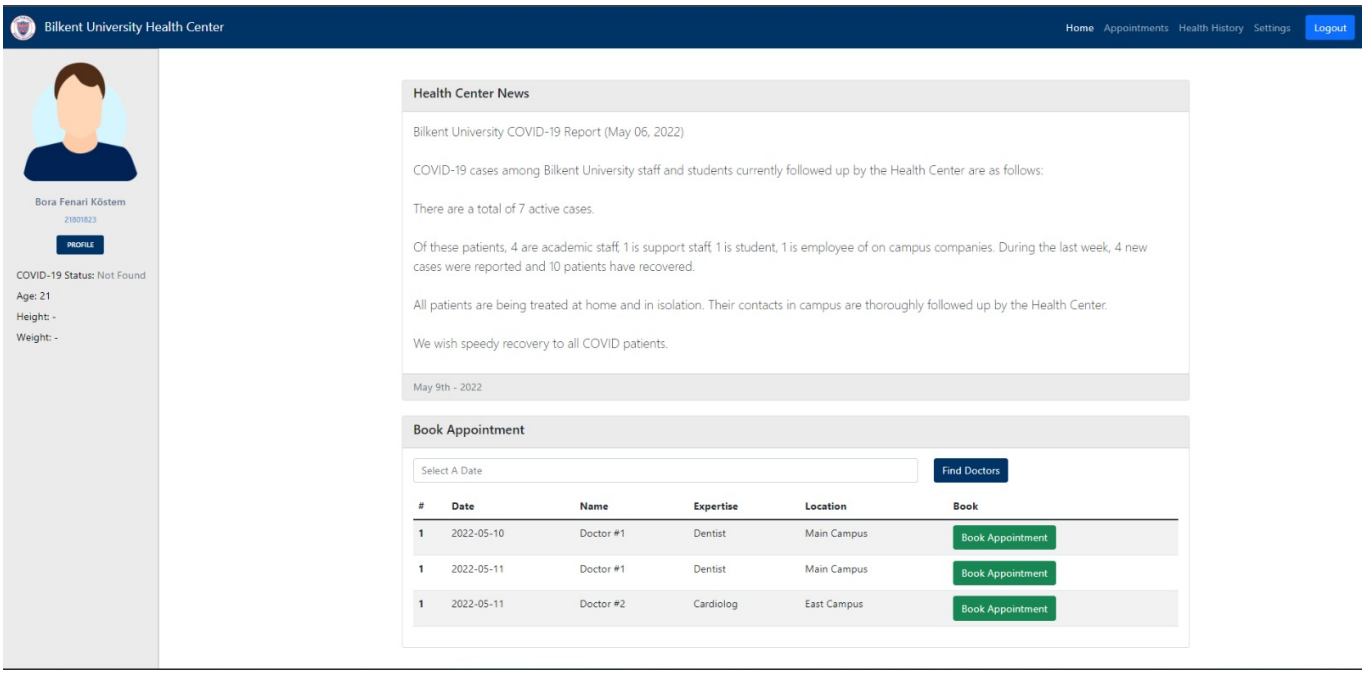
3.1 How to login



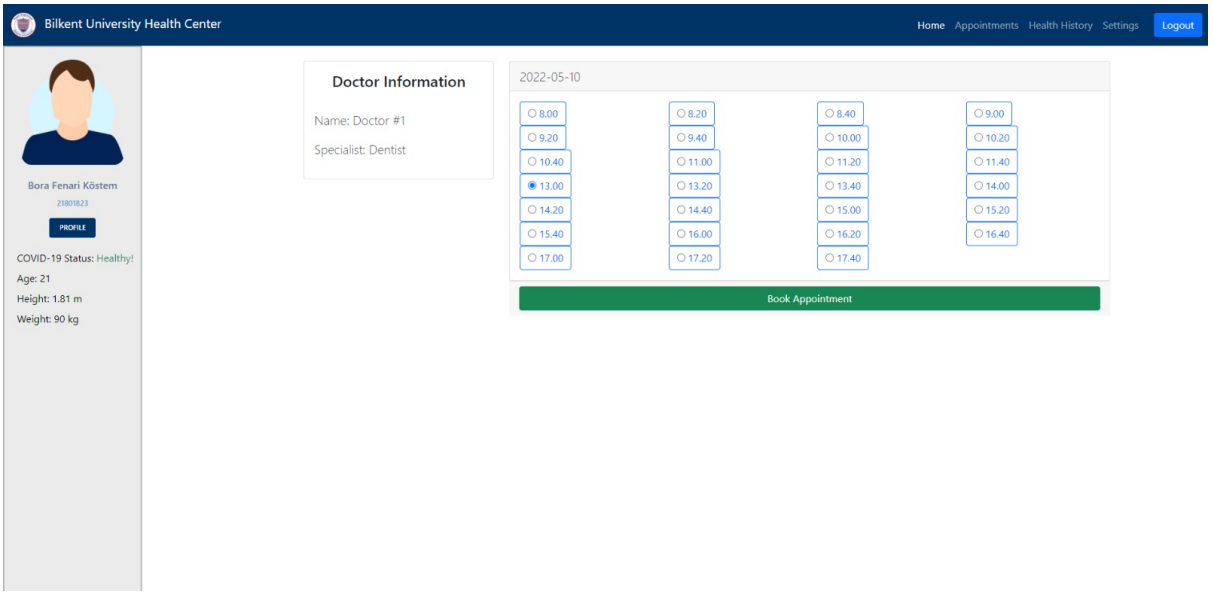


You can log in to the system with the Bilkent ID and password provided by Bilkent BCC. Moreover, the 'World Health Organization (WHO)' Covid-19 page can be accessed by clicking the 'More Info' button under the Health Center News on the login page.

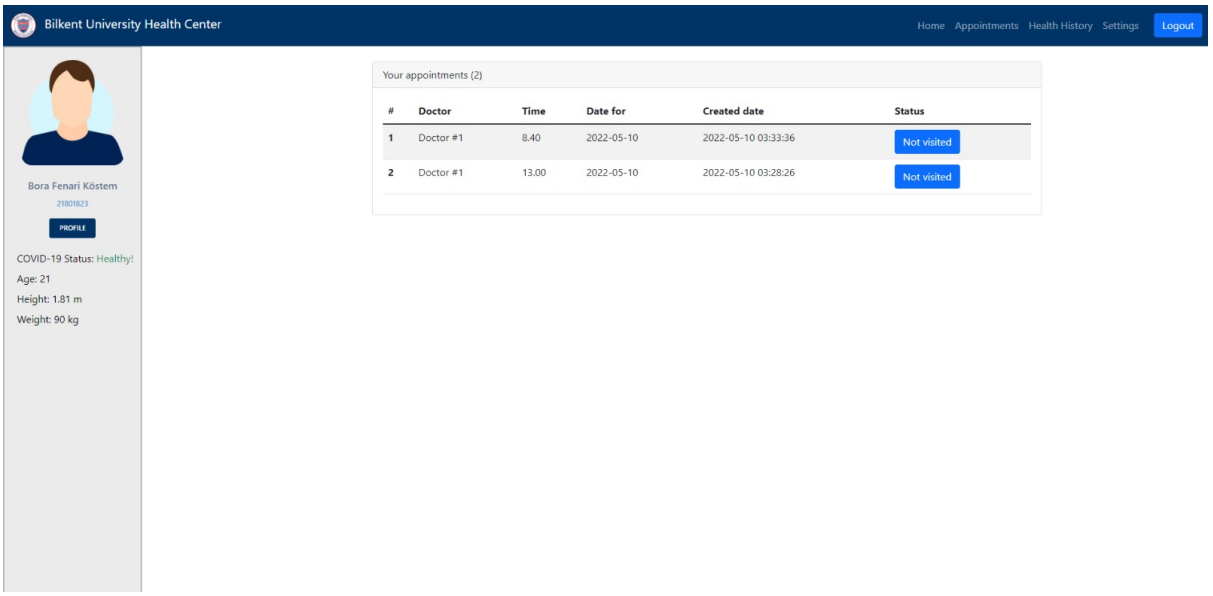
3.2 Book Appointment



All available appointments can be viewed on the main screen that appears after the patient logs in, and can also be filtered according to time.

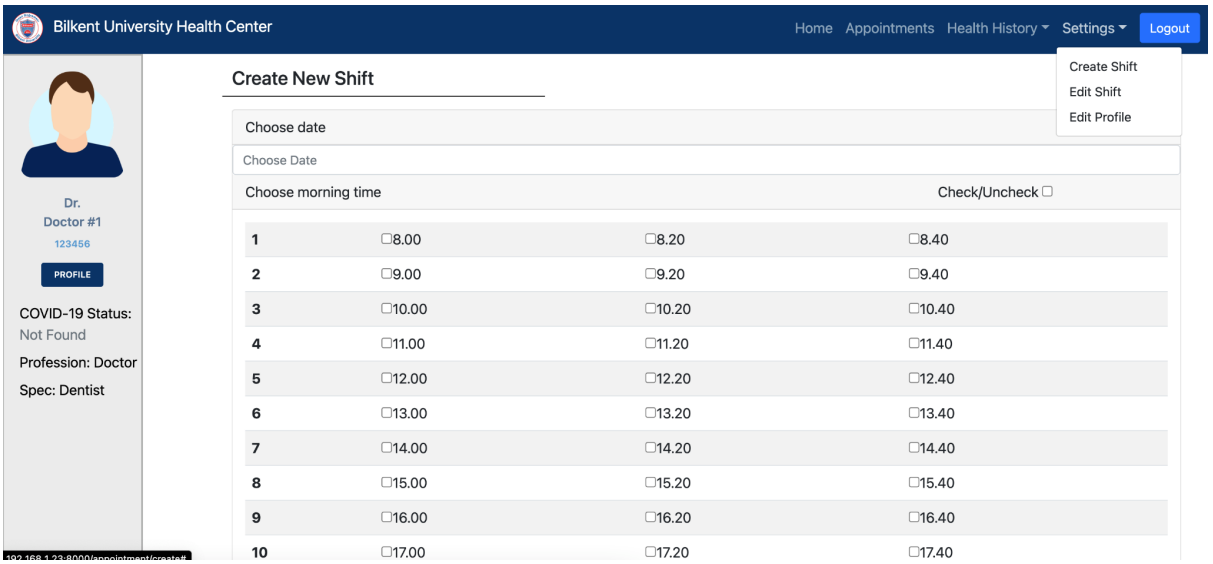


After clicking the 'Book Appointment' button, the appropriate appointment times of the doctor can be displayed. If the 'Book Appointment' button is pressed again after the time is selected, an appointment is created.

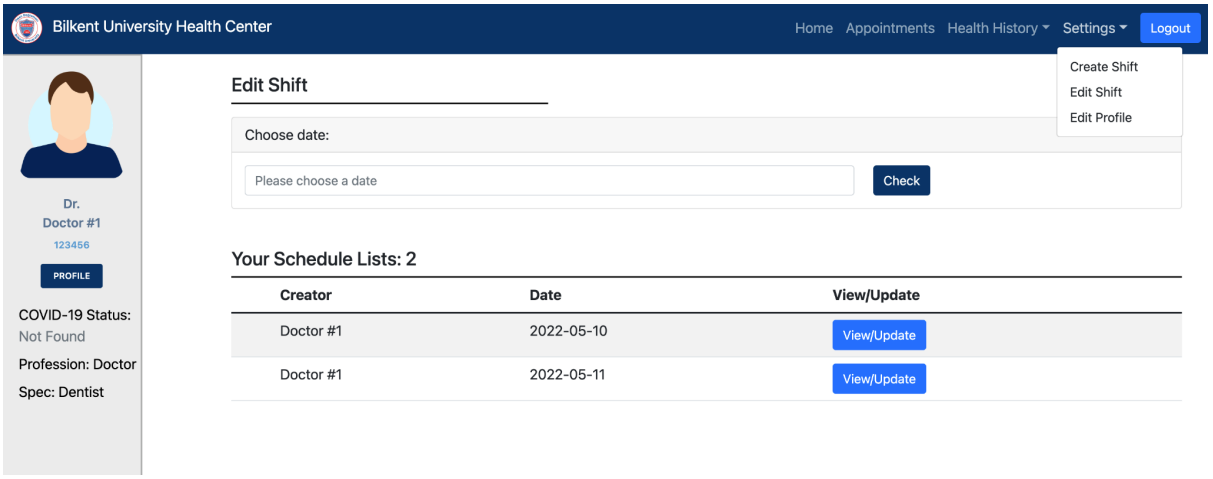


Appointments created are sorted by date and can be viewed by going to the 'appointments' tab.

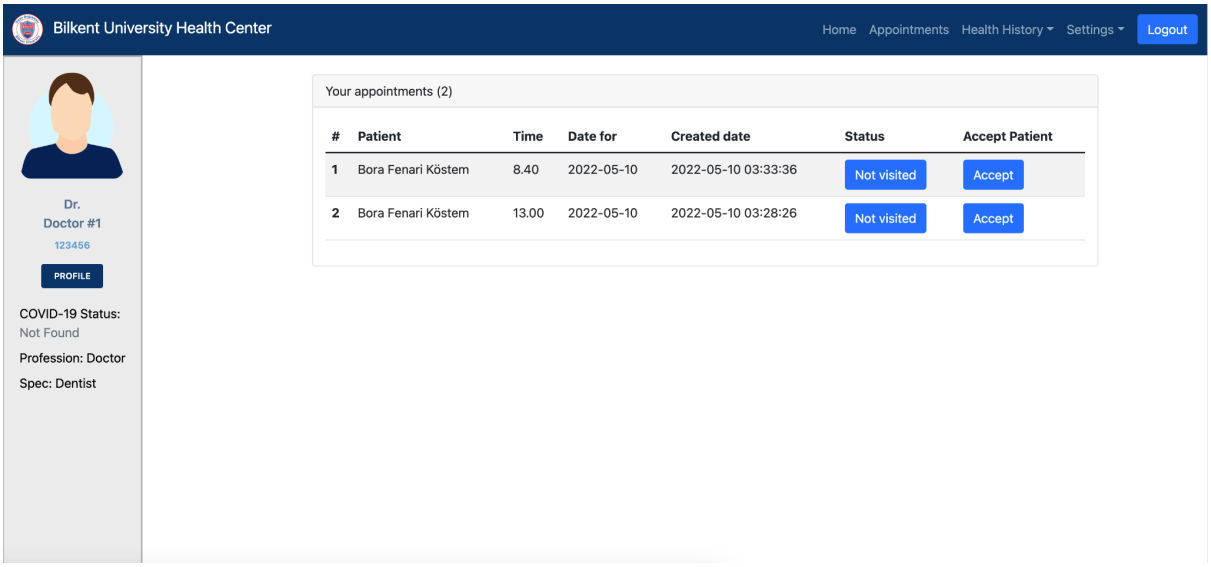
3.3 Doctor's Schedule and Appointments



The shift can be created or edited in the 'Settings' menu. On the creation screen, first, the date is selected, then the boxes for the appointment hours to be created are checked and the submit button is pressed. This completes the shift creation.

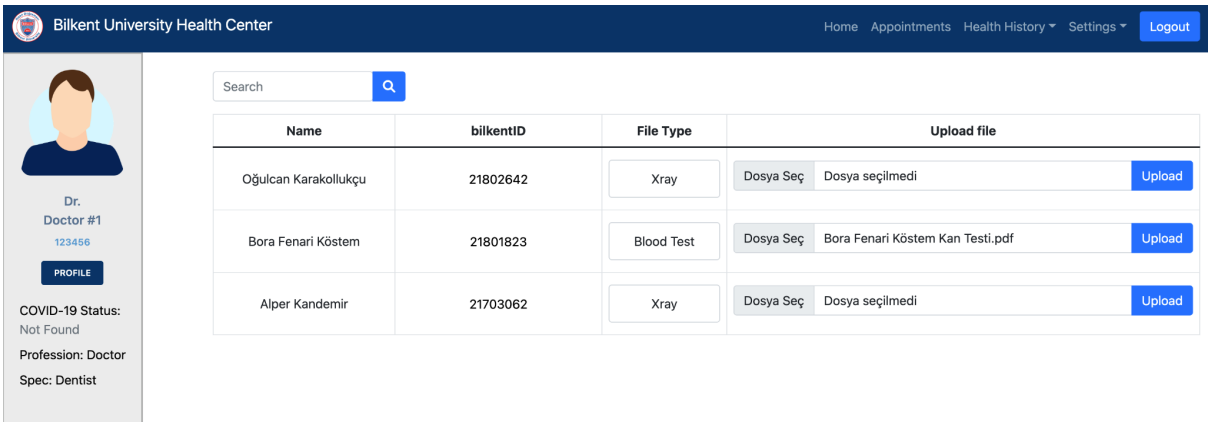


If the 'edit' option is selected from the 'Settings' menu, all shifts created or by date can be viewed. If the 'Update' button is clicked, the appointment times can be updated.

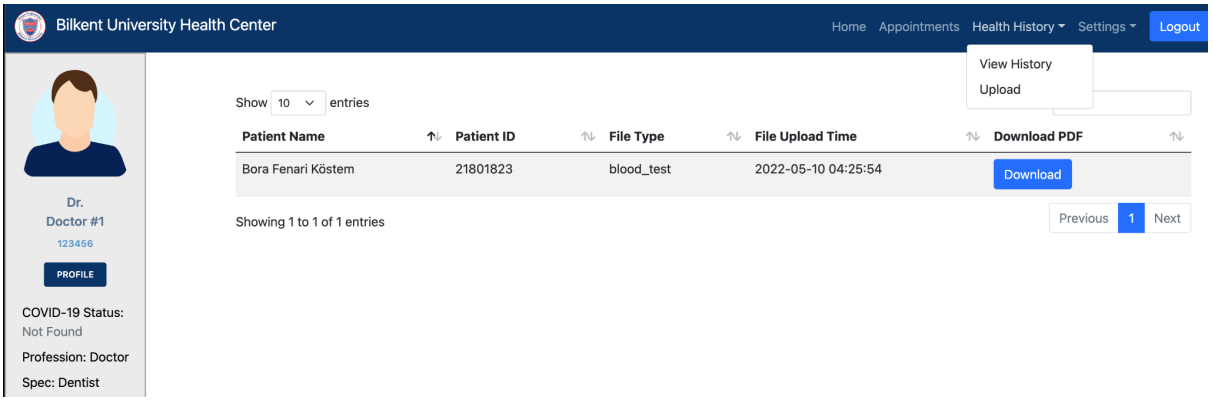


All appointments on that day to this Doctor are seen in chronological order. If the 'Accept' button is pressed, a warning will appear first. If confirmation is given, the patient is accepted, so the appointment status becomes 'visited'.

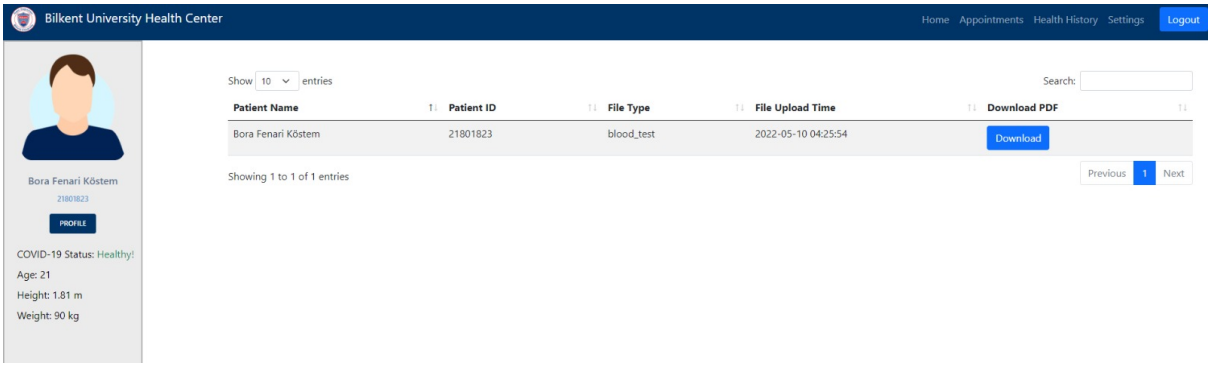
3.4 Health History



The doctor can add the patient's health record by selecting the file type and file for the desired patient.

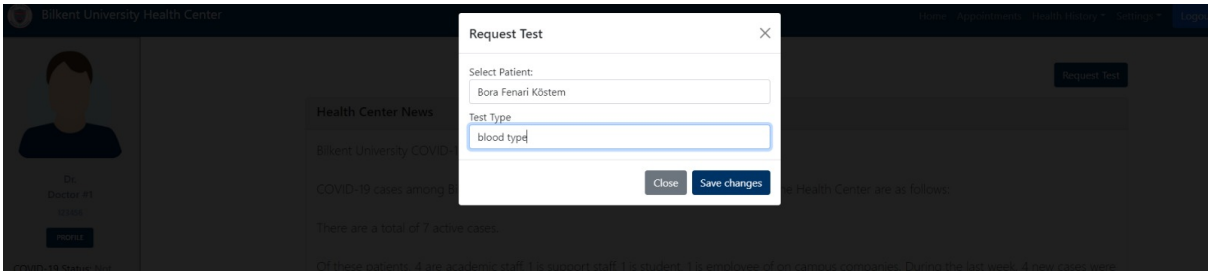


Added records can be downloaded from the 'View History' screen.

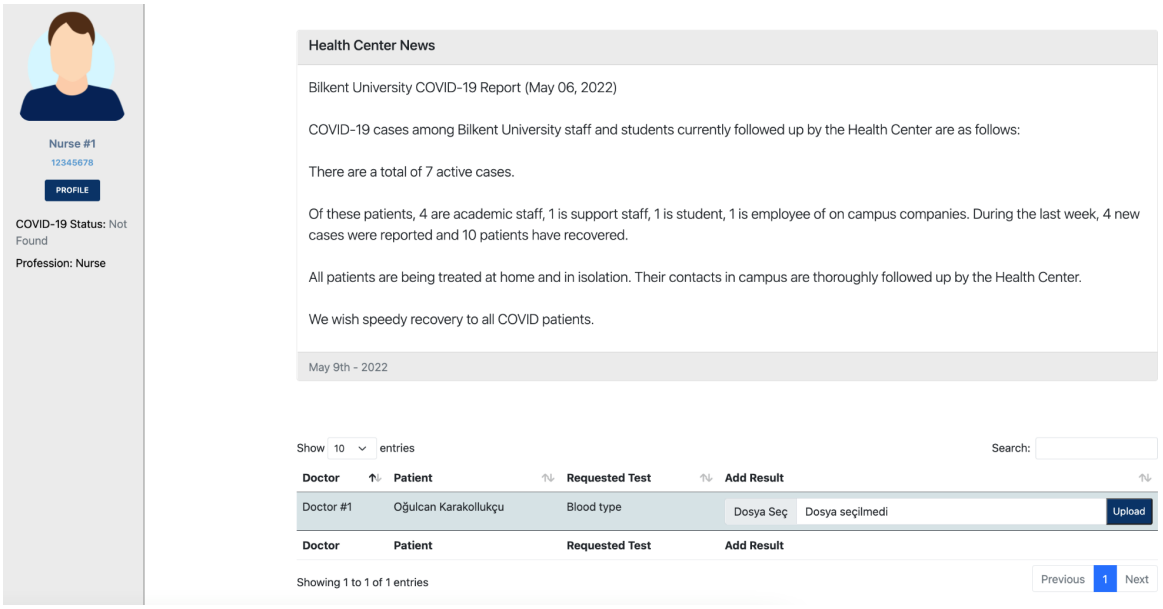


The patient can view the health record uploaded by the doctor or the nurse on the 'Health History' screen.

3.5 Additional Test

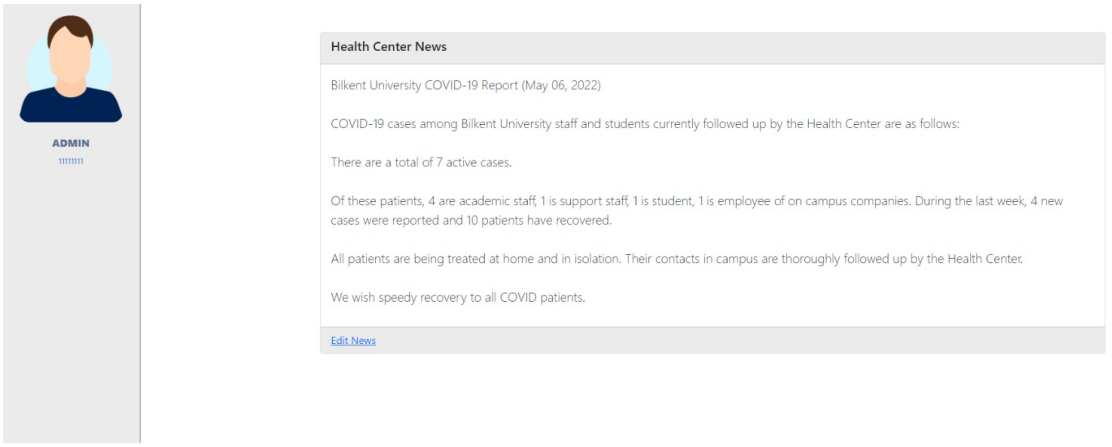


Thanks to the 'Request Test' button, the doctor can request a test for the selected patient.

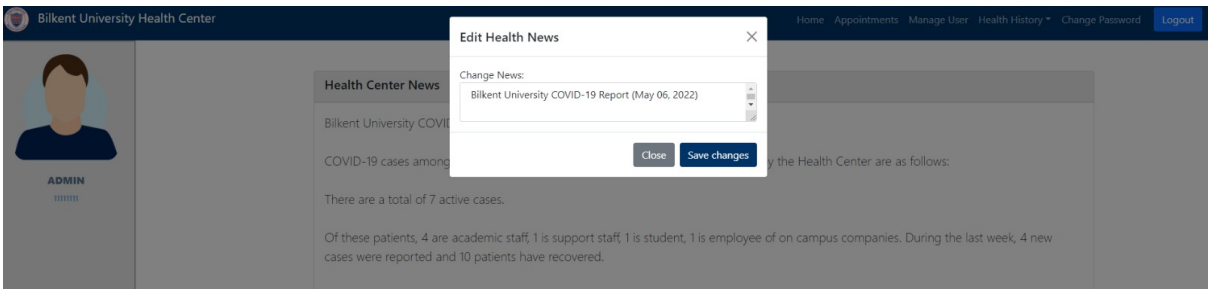


The tests requested by the doctor can be displayed on the nurse's screen.

3.6 Health News

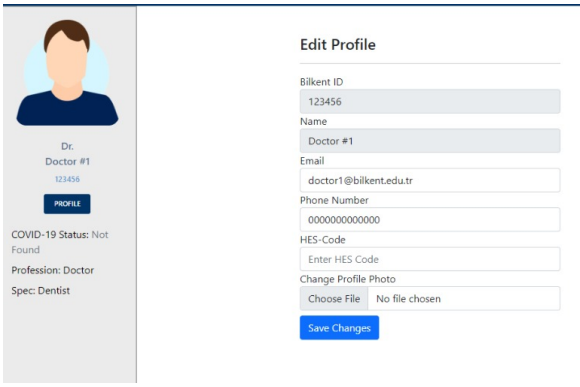


The 'Health News' that appears on the main page can be updated by the admin by clicking the 'Edit News' button.

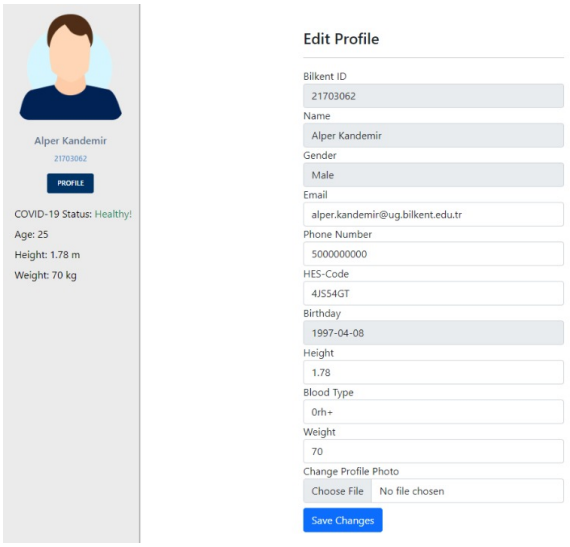


After the entry of the text, the news is updated by pressing the button.

3.7 Profile

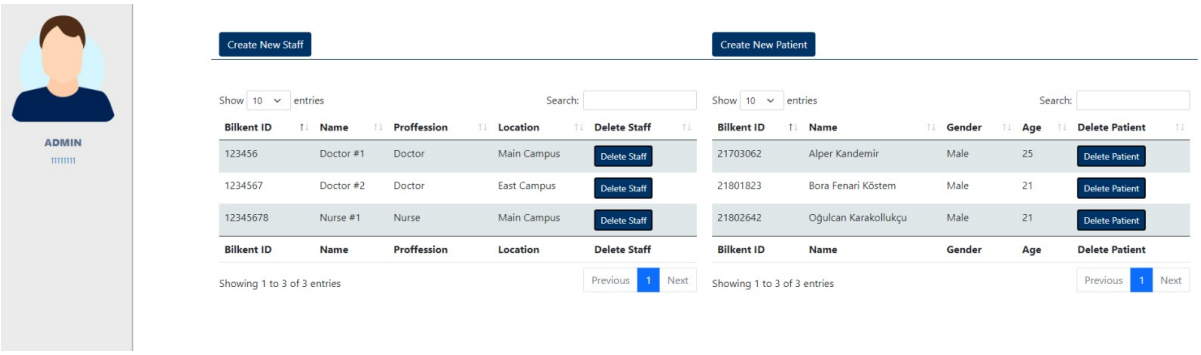


Doctors can update their profiles from the page they are directed to when they click the 'Profile' button and save these updates by clicking the 'Save Changes' button.



Users can update their profiles from the page they are directed to when they click the 'Profile' button and save these updates by clicking the 'Save Changes' button.

3.8 Manage User



Admin can create or delete new users and staff on the 'Manage User' page.

4. Build Instructions

1. Clone project's Github repo
2. Cd into project directory
3. Install Composer Dependencies

   At any time you clone a brand new Laravel project you should install project dependencies. When we start composer, it looks at the composer.json file in the github repository and lists all of the composer (PHP) packages that your repository requires. Because these packages are continually changing, we don't publish the source code to github and instead rely on composer to handle the updates. So we run composer with the following command to install all of this source code.

   ```
   composer install
   ```

4. We must install necessary NPM packages in the same way that we must install composer packages to continue forward. Vue.js, Bootstrap.css, Lodash, and Laravel Mix will all be installed. This is similar to step 4, when we installed the composer PHP packages, except that this time we're installing the Javascript (or Node) packages. The packages.json file that is sent to the github repo contains a list of

packages that a repo requires. We do not commit the source code for these packages to version control (github), as we did in step 4, and instead rely on NPM.

```
npm install
```

5. For security reasons,.env files are rarely committed to source control. However, there is a.env.example file that serves as a template for the.env file that the project requires. So, in the next few stages, we'll make a copy of the.env.example file and create a.env file that we can start filling out to accomplish things like database configuration.

   ```
   cp .env.example .env
   ```

   This will duplicate the.env.example file in your project and rename it simply .env.

6. Create an empty database

   For example MySQL

7. Add database information in the .env file

8. Migrate the database

   ```
   php artisan migrate
   ```

9. Running the app

   ```
   php artisan serve
   ```

5. Work Allocation

Analysis Report:

-Introduction (Alper)

-Proposed System (Alper, Oğulcan)

-Use Case Model (Alper, Bora, Oğulcan)

-Sequence Diagram (Nafissa, Kylian -1st iteration) - (Alper, Oğulcan -2nd iteration)

-State Diagram(Nafissa, Kylian -1st iteration) - (Alper, Bora -2nd iteration)

-Activity Diagram(Alper, Bora)

-Class Diagram (Alper, Bora, Oğulcan)

-Mock-Ups (Bora)

-Improvement and Summary (Alper)

-References (Alper)

Design Report:

-Introduction (Alper, Bora, Oğulcan)

-Subsystem Decomposition (Alper, Oğulcan)

-Deployment Diagram (Oğulcan)

-Hardware/Software Mapping (Oğulcan)

-Persistent Data Management (Oğulcan)

-Access Matrix (Oğulcan)

-Boundary Conditions (Alper, Oğulcan)

-Object Design Trade-offs (Alper, Oğulcan)

-Final Object Design (Oğulcan)

-Packages (Alper, Bora, Oğulcan)

-Design Patterns (Alper, Bora)

-Class Interfaces (Alper, Bora, Oğulcan)

-Improvement Summary (Alper)

-References (Alper)

Design Report:

-(Alper, Oğulcan)

Implementation

-Login (Bora)

-Authority Level Creation (Bora)

-Manage Users (Bora)

-User Profiles (Bora)

-Profile Edit (Bora)

-Request Additional Tests (Bora)

-Health News (Bora)

-Health History Upload & Download (Oğulcan)

-Health History View (Oğulcan)

-Doctor Schedule Create & Update (Alper)

-Appointments Booking (Alper)

-Filter Appointments (Alper)

-Appointment and schedule merging to login and user system (Alper,Oğulcan)

-Health history system merging to main system (Bora)

-Frontend (Bora, Alper, Oğulcan)

-Bug Fixing (Alper,Bora,Oğulcan)

-Testing(Bora,Alper,Oğulcan)

-Everyone has created their own system’s databases.(Alper,Bora,Oğulcan)

6. Status of the Implementation

Incomplete:

-Color Blind Mode

-Email Notifications

-Secretary Booking Appointments

-Forgot Password

-Remaining Time to Appointment

-Pre-Examination