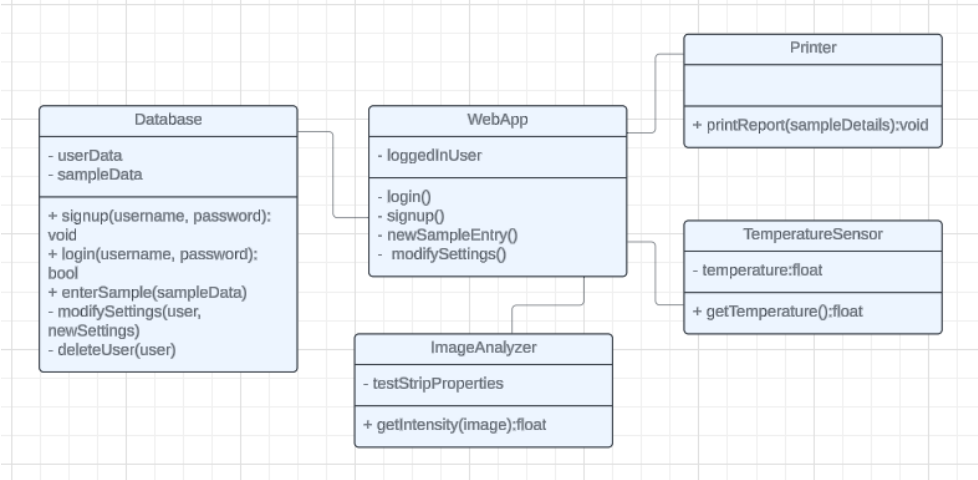
Product Design

Team 9 . Mayaank Ashok, Nitin Avuthu, Maitreya Chitale, Ketaki Shetye

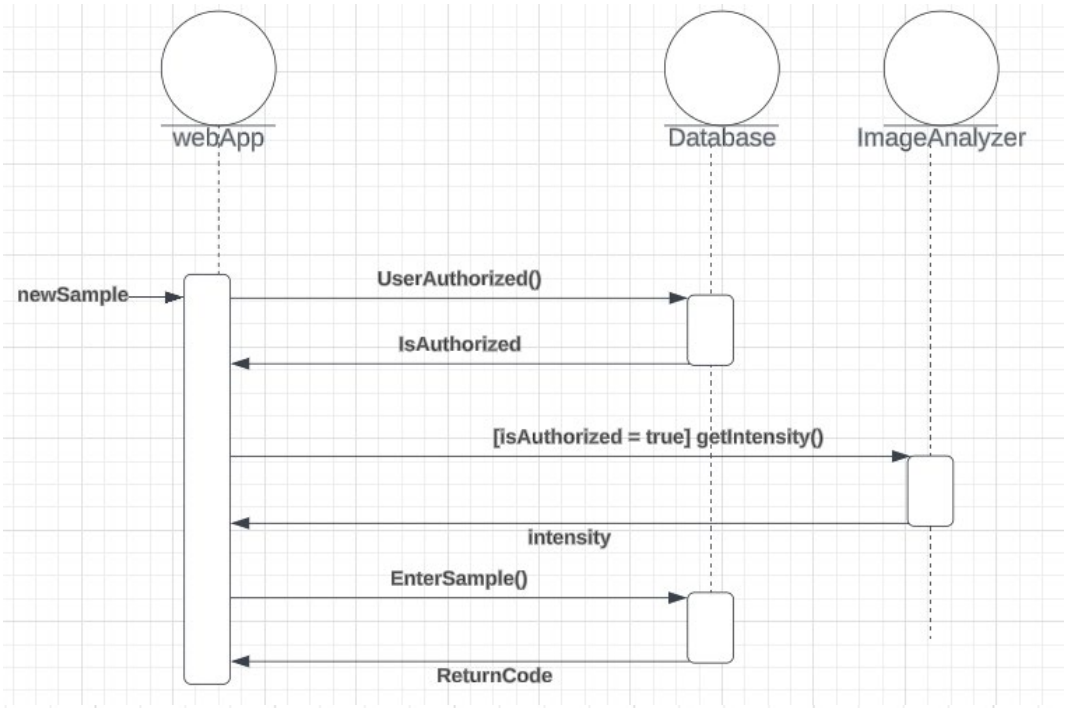
Design Model

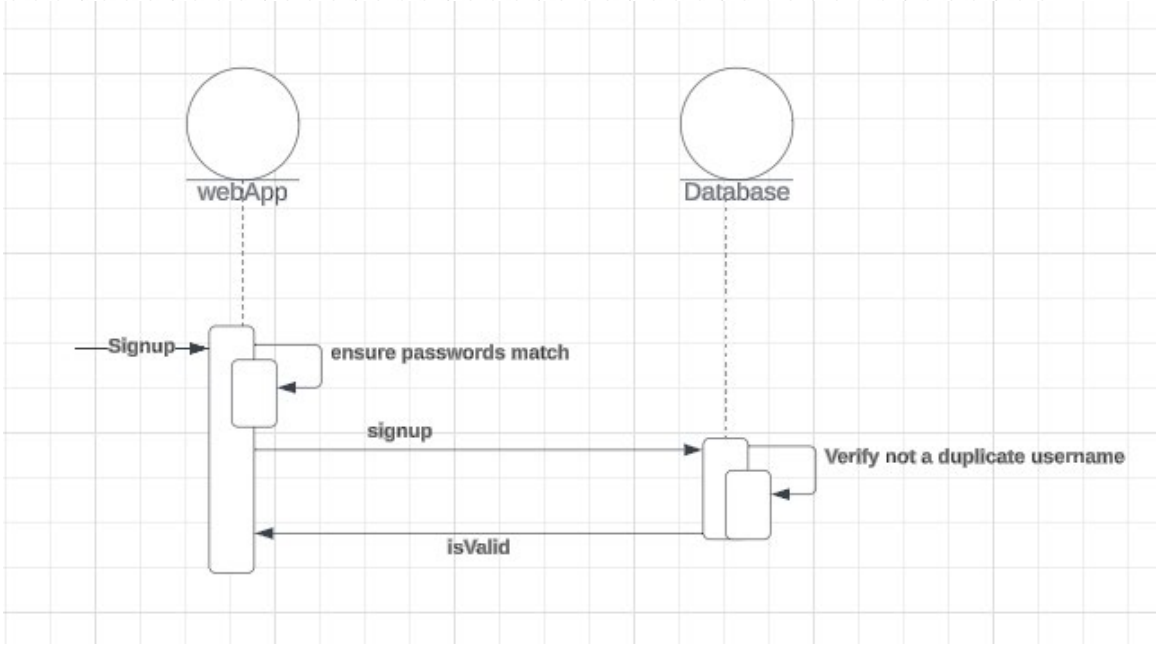


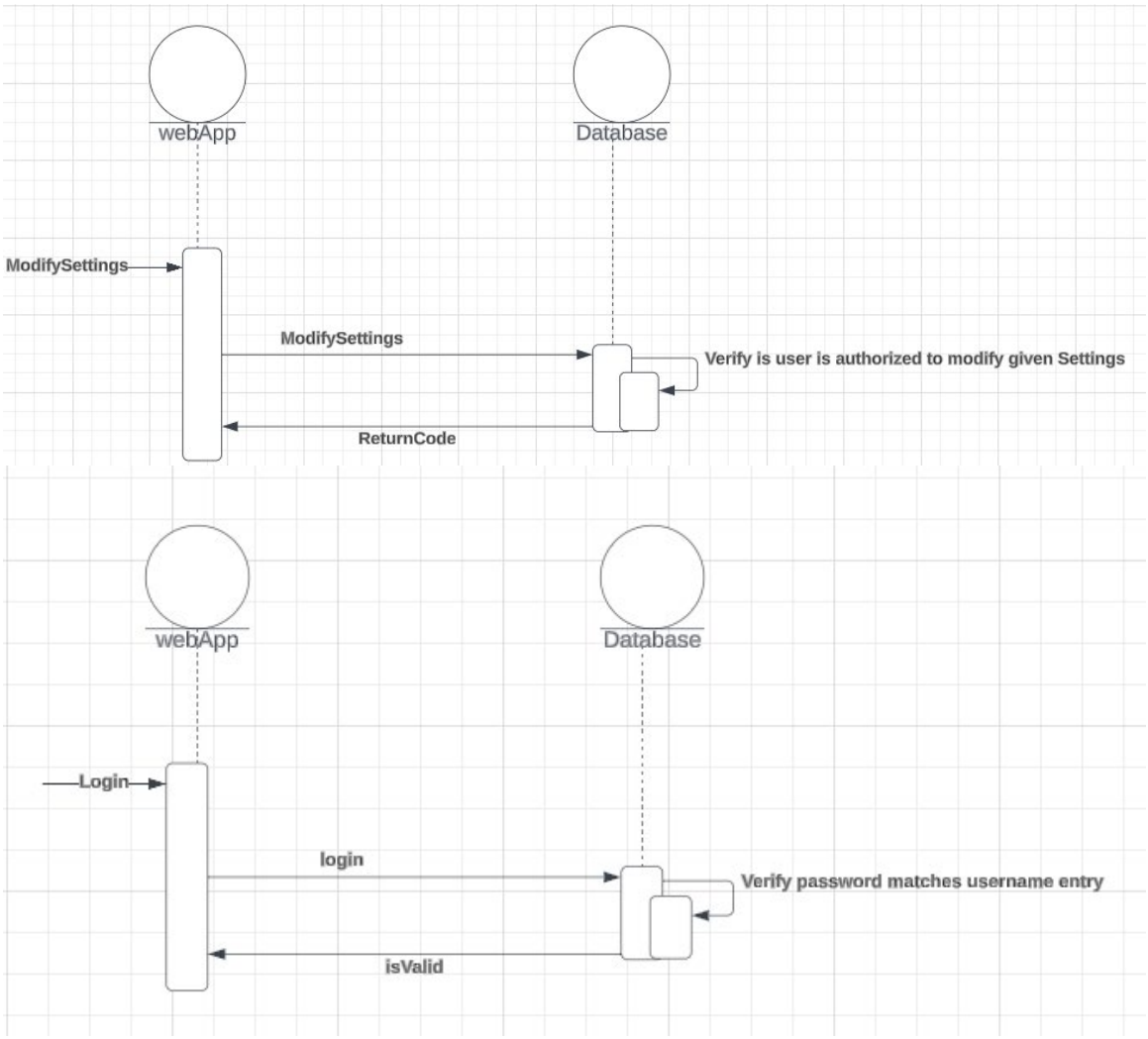
WebApp	• The webApp encapsulates the user Interface, it stores the logged in user and interfaces with the database, image analyzer, temperature sensor, and printer • login() : opens a form where the user can enter valid credentials to login. • Signup() : opens a form where a new user can register • newSampleEntry() starts a process to record the details of a new sample. • modifySettings() : allows a user with privileges to modify the settings of other users.
Printer	• The printer refers to the interface of the built in thermal printer. • printReport() takes in the details of a sample and the result of its to generate and print a report.
ImageAnalyzer	• The image analyzer stores the details of the layout of the test strip. • The getIntensity() function takes in an image of the test strip and outputs the corresponding intensity of the reacting area.
TemperatureSensor	• It stores the ambient temperature of the surroundings. As recorded by the temperature sensor. • The getTemperature() returns the current temperature, as recorded by the sensor.
Database	• The database stores all the data in the software. This includes the login details and permissions of each user, as well as the details of each sample recorded. • The signup, login, enterSample, and modifySettings functions all implement the functionalities specified in the webApp. • deleteUser() allows a user with admin rights to delete the credentials of another user.

.

Sequence Diagram(s)







.

Design Rationale

our project required us to build a blood sample test kit. The client already had an overarching design and we were required to build upon it. The design from the client included a physical test kit, including raspberry pi and a touch screen module. We were asked to build an algorithm to quantify the intensity of red from the test sample and a software system which included processing of this data and presenting it to the user. To build the algorithm, we narrowed down the options of python libraries to OpenCV and PIL. On implementing the algorithms, we found OpenCV to be better suited for the task and chose to proceed with it. During the design process for the UI, we considered building a webapp and an android app but we settled on building just a webapp using React as it would be more convenient and would ensure a faster development cycle and hence adhere to the timeline. We chose MongoDB to build the database to ensure seamless integration with both python and the UI. During the development process, the client also requested the addition of a temperature sensor to the kit. We chose the model that would be best suited for the workload after considering the reliability of the sensor and the ease of integration with raspberry pi.