PERSONALIZED HEALTH AND WELLNESS ASSISTANT

CLASS DIAGRAM

Course Title – Software Engineering

Slot – G2 + TG2

Name – Vansh Kumar Payala

Registration No. – 22BAI1386

Introduction –

The ‘Personalized Health and Wellness Assistant’ will serve as a new generation mobile application that will include artificial intelligence and machine learning components to suggest a recommended health and wellness plan based on data inputted by the user. This is a new concept which will include a chatbot and will assist the users with medication and prescription guidance; therefore, the usage of the app will be enhanced while appropriate health support will be provided. The most important one is to educate and give specific recommendations concerning the state of one’s health and help to make the right decision.

To accomplish the mentioned above proposed objective of a well-structured and efficient manner of development, Software Development Life Cycle, namely the Incremental Model has been chosen. This way the application is develop in phases including this feedback of users and can be improved gradually by incorporating more features. Thus, when realizing the project in stages, it is also possible to ensure that each of the elements is functional and the final outcome is a usable application that will meet the needs of end-users. It also shows the concrete description of Incremental Model in the framework of the ‘Personalized Health and Wellness Assistant’, as well as naming all the activities of each phase and their outcomes.

Class Diagram –

The class diagram for the "Personalized Health and Wellness Assistant" project illustrates the structure and relationships between different classes in the system. It captures the key components such as User, ML Model, Firebase Database, and App Interface, and their attributes and methods. The User class represents the user of the app, containing attributes like name, age, gender, and medical history, with methods for registering and updating their profile. The App Interface class handles interactions between the user and the system, providing methods to take inputs such as age, symptoms, and other health data, and facilitating the prediction process. The ML Model class contains methods for processing inputs using machine learning algorithms to predict potential diseases based on user-provided symptoms. The Firebase Database class interacts with cloud storage, handling user data storage and retrieval operations, ensuring data persistence. Relationships such as association between the user and app, aggregation between the app and ML model, and composition between the app and Firebase database ensure clear interaction pathways between these components, highlighting the integral roles of each class in the system's functionality.

Steps –

1. Classes, Attributes, and Methods:

1. **User**

   ➢ ***Attributes:***

   - userId: String
   - name: String
   - email: String
   - password: String
   - age: int
   - gender: String
   - mobileNumber: String

   ➢ ***Methods:***

   - registerUser()
   - loginUser()
   - updateProfile()
   - getProfileDetails()

2. **HealthData**

   ➢ **Attributes:**

   - userId: String
   - age: int
   - bmi: double
   - hemoglobin: double
   - mcv: double

        - mch: double
    - **Methods:**
        - inputHealthData()
        - updateHealthData()
        - getHealthData()

3. **MLModel**
    - **Attributes:**
        - modelId: String
        - modelName: String
        - modelVersion: String
    - **Methods:**
        - processHealthData(healthData: HealthData): Prediction
        - updateModel()
        - getModelDetails()
4. **Prediction**
    - **Attributes:**
        - predictionId: String
        - userId: String
        - healthCondition: String
        - predictionDate: Date
        - accuracy: double
    - **Methods:**
        - generatePrediction(healthData: HealthData): String
        - getPredictionDetails()

5. **Chatbot**

   - **Attributes:**
     - chatbotId: String
     - chatbotName: String
   - **Methods:**
     - getResponse(userInput: String): String
     - provideHealthTips()
     - prescribeMedicine(healthCondition: String)

6. **FirebaseService** (Utility Class)

   - **Methods:**
     - saveUserData(user: User)
     - authenticateUser(email: String, password: String): boolean
     - storeHealthData(healthData: HealthData)
     - fetchHealthData(userId: String): HealthData
     - savePrediction(prediction: Prediction)
     - fetchPrediction(userId: String): Prediction
     - saveReport(report: Report)
     - fetchReport(userId: String): Report

Relationships:

1. **User ↔ HealthData**

   - Relationship Type: Association

- Description: A User has HealthData associated with them. This means a user can have one set of health parameters like age, BMI, MCH, etc.

2. **User ↔ Prediction**

   - Relationship Type: Association (One-to-Many)

   - Description: A User can have multiple Predictions generated over time based on different health data inputs.

3. **User ↔ FirebaseService**

   - Relationship Type: Dependency

   - Description: The User class depends on the FirebaseService class for actions like registering, logging in, and updating user details. It doesn't own or control the FirebaseService directly.

4. **HealthData ↔ MLModel**

   - Relationship Type: Association

   - Description: The MLModel processes HealthData to generate predictions. HealthData is passed to the MLModel for analysis.

5. **MLModel ↔ Prediction**

   - Relationship Type: Direct Association

   - Description: The MLModel directly generates a Prediction based on the given HealthData. There is a direct connection between the MLModel and Prediction.

6. **User ↔ Chatbot**

   - Relationship Type: Association

   - Description: The User interacts with the Chatbot to receive health tips, recommendations, and other advice based on their health condition.

7. **Chatbot ↔ FirebaseService**

   - Relationship Type: Dependency

- Description: The Chatbot depends on the FirebaseService to fetch and provide relevant health information and tips to the user based on their data.

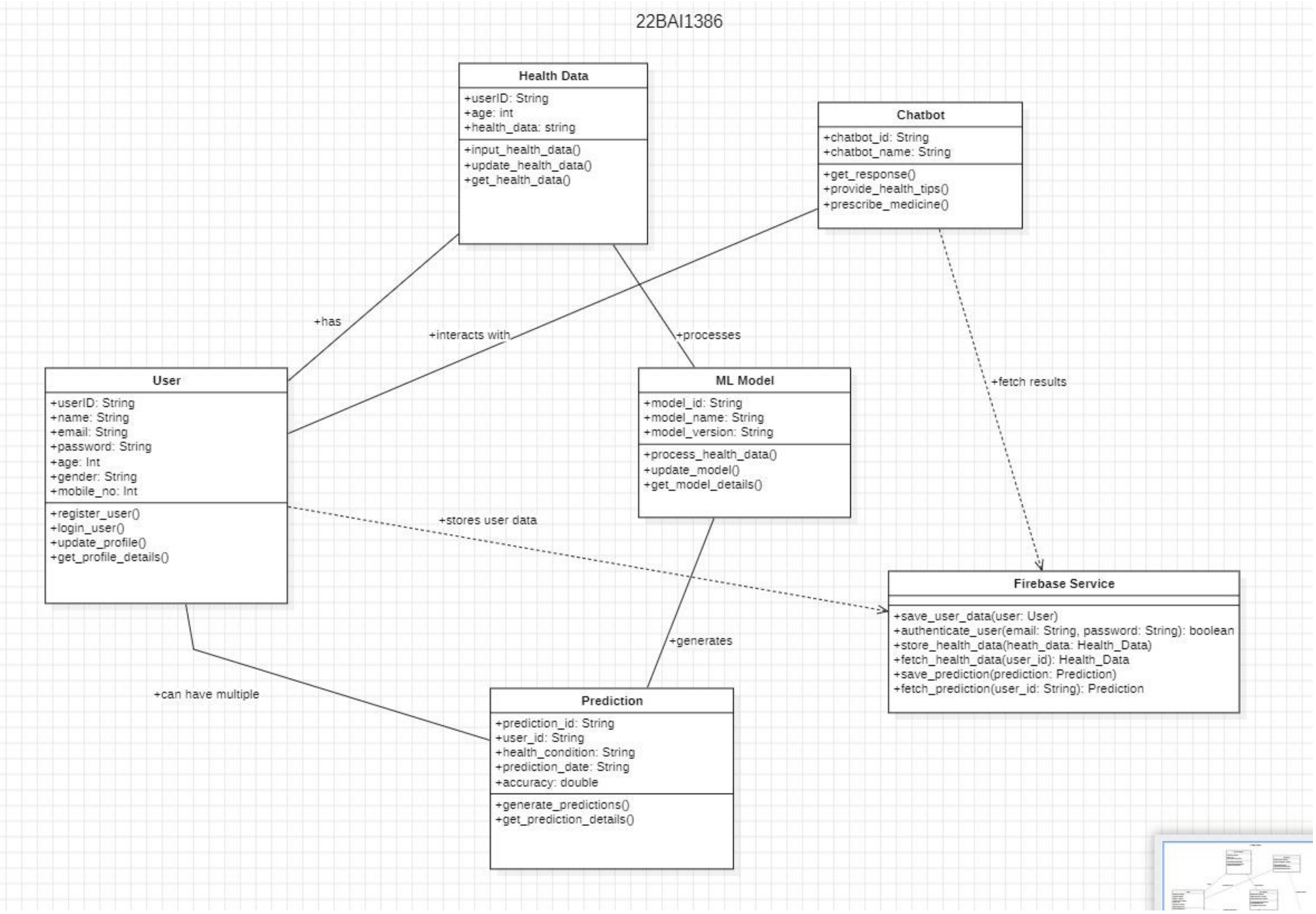
8. User ↔ MLModel

- Relationship Type: Dependency
- Description: The User depends on the MLModel to process their health data and generate predictions, but the User does not have direct control over the model.

9. MLModel ↔ FirebaseService

- Relationship Type: Dependency
- Description: The MLModel uses FirebaseService to fetch health data, store prediction results, and manage model updates.

Final Class Diagram –


22BAI1386
Health Data
+userID: String
+age: int
+health_data: string
+input_health_data()
+update_health_data()
+get_health_data()
Chatbot
+chatbot_id: String
+chatbot_name: String
+get_response()
+provide_health_tips()
+prescribe_medicine()
+has
+interacts with
+processes
+fetch results
User
+userID: String
+name: String
+email: String
+password: String
+age: Int
+gender: String
+mobile_no: Int
+register_user()
+login_user()
+update_profile()
+get_profile_details()
ML Model
+model_id: String
+model_name: String
+model_version: String
+process_health_data()
+update_model()
+get_model_details()
+stores user data
Firebase Service
+save_user_data(user: User)
+authenticate_user(email: String, password: String): boolean
+store_health_data(heath_data: Health_Data)
+fetch_health_data(user_id): Health_Data
+save_prediction(prediction: Prediction)
+fetch_prediction(user_id: String): Prediction
+generates
+can have multiple
Prediction
+prediction_id: String
+user_id: String
+health_condition: String
+prediction_date: String
+accuracy: double
+generate_predictions()
+get_prediction_details()