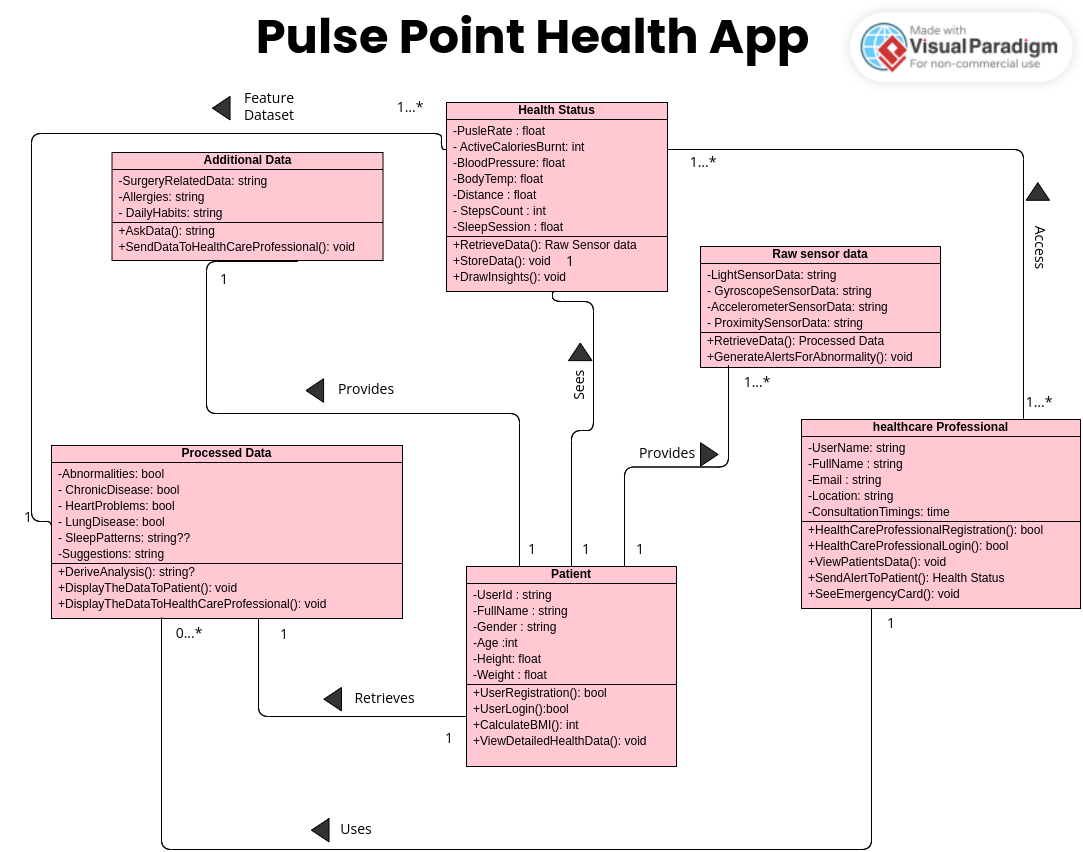
**Product Design**

| **Team-30** | **Pulse Point Health App** |
| --- | --- |
|  |  |

# Deekshitha Krishna Koushik Susheel krishna Swaroop Pratyush Jena

**Class Diagram**



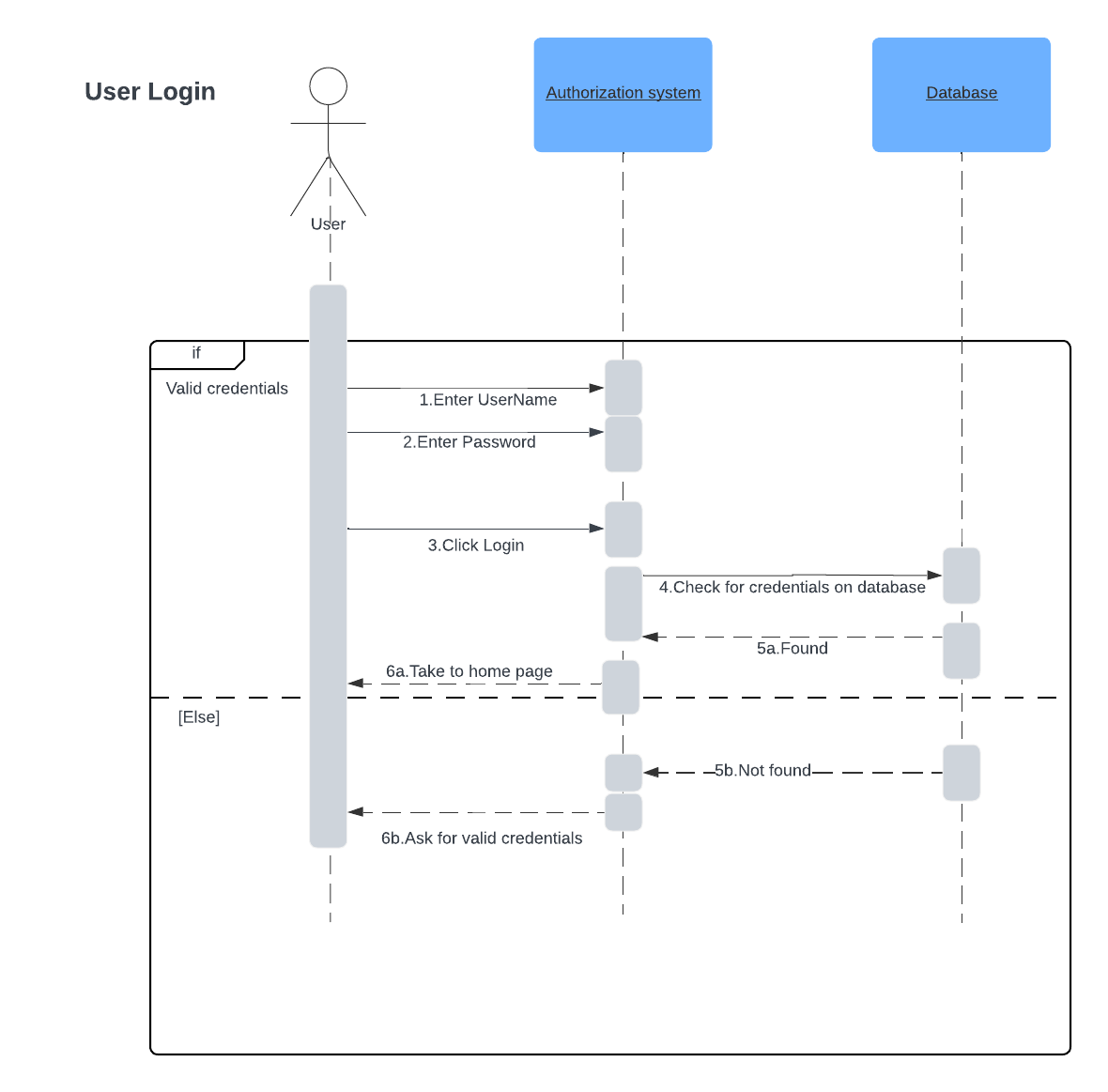
# Design Model

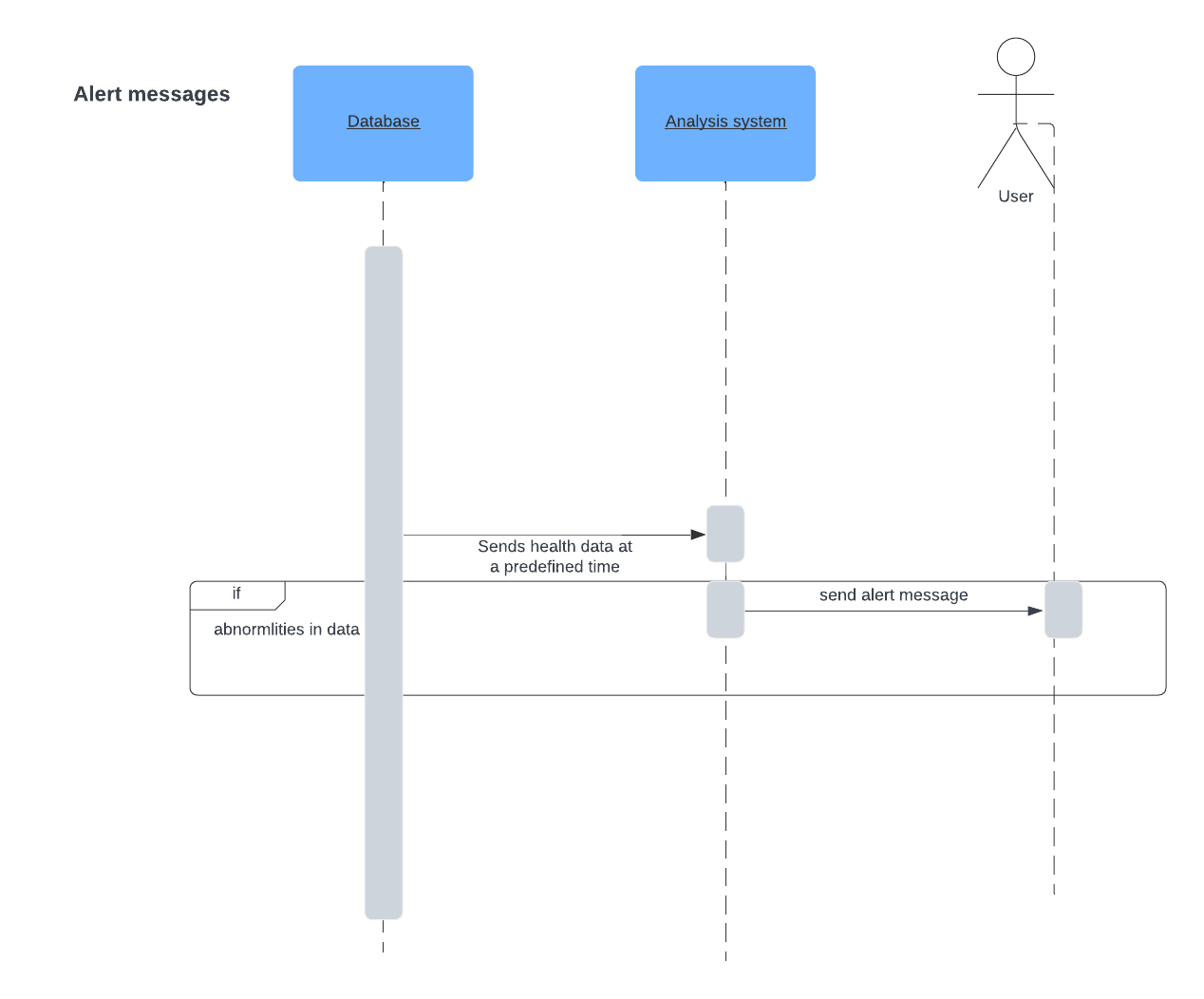
C 01

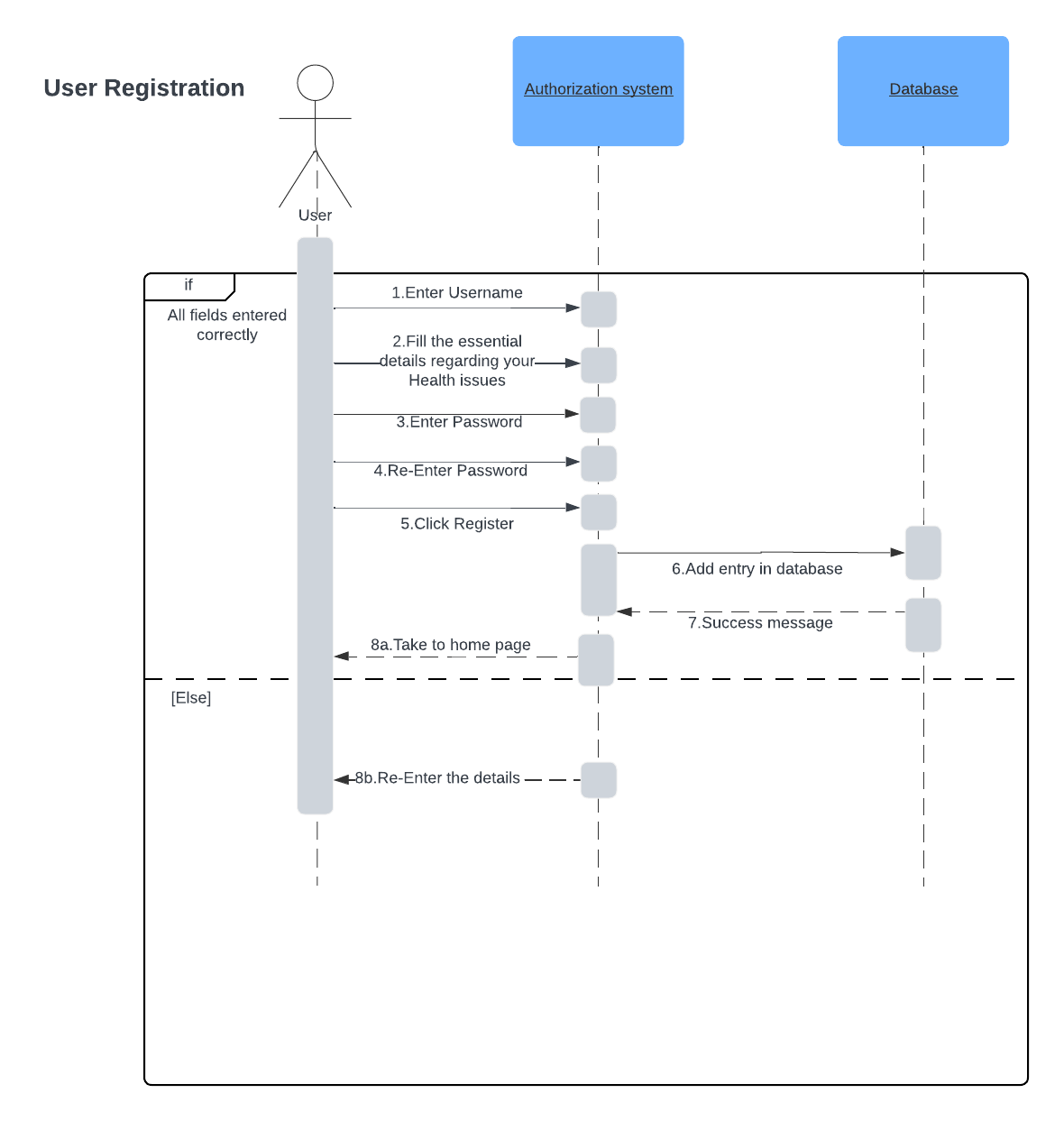
| C 01 | **Patient :**   * Username * Full Name * Gender * Age * Height * Weight   **Class behavior :**   * user registration * user login * calculate BMI * view detailed health data |
| --- | --- |
| C 02 | **Healthcare Professional :**   * Username * Full Name * email * Location * Consultation timings   **Class behavior :**   * Healthcare professional registration * Healthcare professional Login * view patients data * send alert to patient * see emergency card |
| C 03 | **Health Status :**   * Pulse rate * Active Calories burnt * Blood pressure * Body temperature * Distance * Steps Count * Sleep session   **Class behavior :**   * retrieve data * store data * derive insights |
| C 04 | **Raw sensor data :**   * Light sensor data * Gyroscope sensor data * Accelerometer sensor data * Proximity sensor data   **Class behavior :**   * retrieve data * generate abnormal alerts |
| C 05 | **Additional Data:**   * Surgery related data * patient problems * patient symptoms and allergies * Patient daily habits   **Class behavior :**   * ask data * send to healthcare professional |
| C 06 | **Processed Data:**   * Abnormalities * Chronic diseases * Heart problems * Lung diseases * Sleep Patterns * Suggestions   **Class behavior :**   * derive analytics * process data * show to patient * show to Healthcare professional |

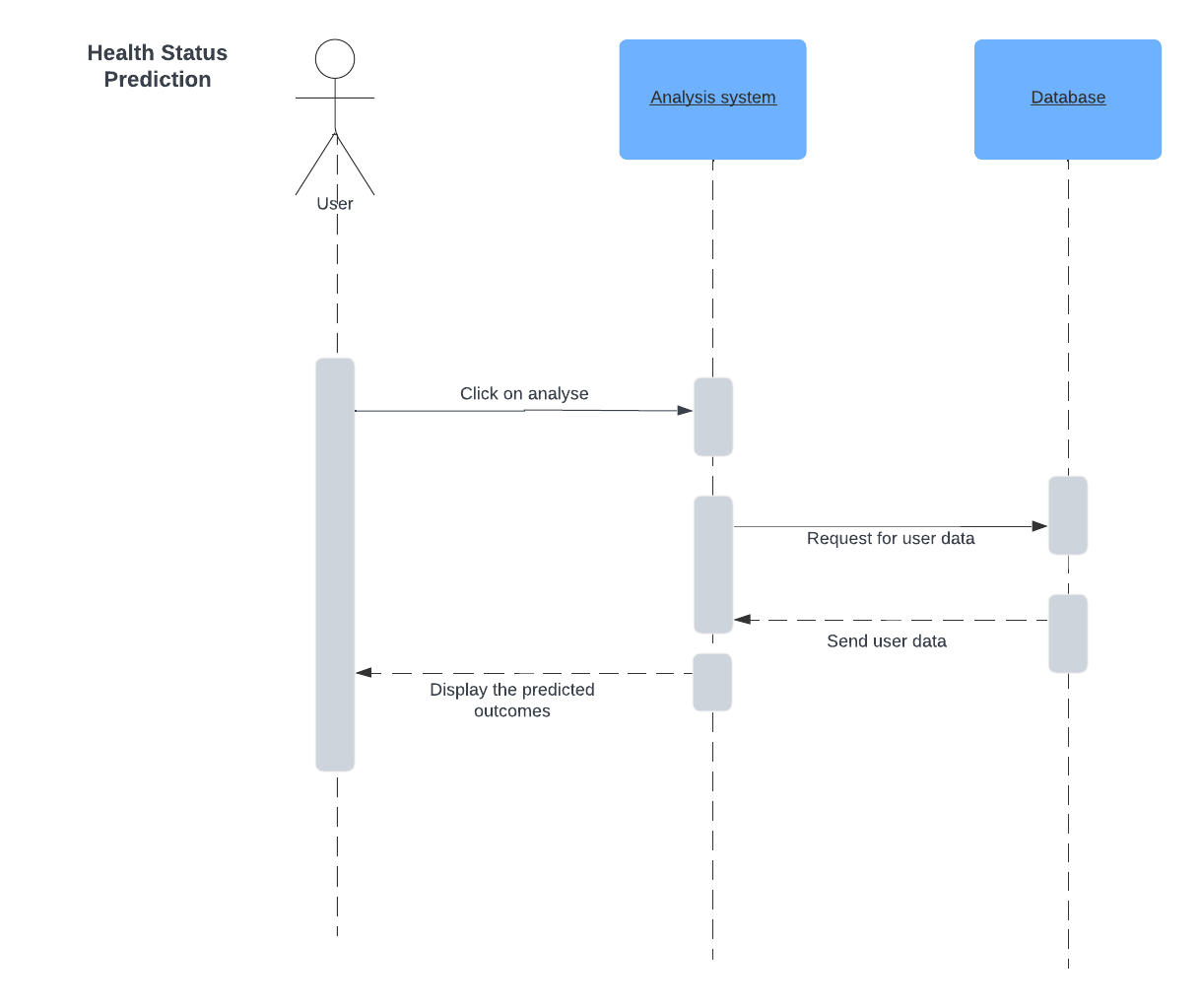
.

# Sequence Diagram(s)









# Design Rationale

**Design Version - 1 :**The initial design adhered to the client's specifications; however, both our team and the client were initially uncertain about the requisite programming language and technologies. Consequently, comprehensive research was conducted to ascertain the project's requirements. It was determined that an Application Programming Interface (API) was necessary to access the user's health data. Rather than recalculating the user's health records from the inception, our preference was to access the user's existing health data.

Upon further investigation, we discovered the relevance of Google APIs and the Google Fit App, which stores user health data in cloud storage. This data can be accessed using Google REST APIs. Consequently, after consulting the documentation of the fitness API (refer to the link provided), we opted to utilize the CURL language to send HTTP 1.1 requests to the Google Cloud, retrieving the data in JSON format. Subsequently, our plan was to analyze this data for the development of the Machine Learning Model.

However, we encountered certain challenges during the implementation phase. Directly accessing health data proved unfeasible due to the necessity of generating tokens, each with a duration of one hour. Additionally, each token creation attempt required user permission through their Google account, which presented a considerable obstacle. Furthermore, obtaining a substantial amount of data for the Machine Learning model was impeded by the limitations of REST APIs, preventing access to extensive datasets.

***Strengths:***

1. Adherence to client specifications: The initial design demonstrates a commitment to meeting the client's specifications, ensuring alignment with their requirements.

2. Comprehensive research: The team conducted thorough research to understand the project's requirements, indicating a dedication to finding optimal solutions.

3. Integration of Google APIs: The decision to leverage Google APIs and the Google Fit App for accessing user health data demonstrates an awareness of existing technologies and a strategic choice for data retrieval.

***Deficiencies:***

1. Uncertainty about technologies: The initial uncertainty regarding the choice of programming language and technologies highlights a potential lack of clarity in the project planning phase.

2. Token generation and user permission challenges: The need to generate tokens with a limited duration and requiring user permission through Google accounts poses significant obstacles, impacting the ease of access to health data.

3. Limitations of REST APIs: The restrictions imposed by REST APIs, hindering the access to extensive datasets for the Machine Learning model, signify a limitation in the chosen approach and technology stack.

4. Lack of alternative solutions: The design does not mention exploring alternative solutions or contingency plans when faced with challenges, potentially limiting adaptability in the face of unexpected issues.

5. Potential security concerns: While not explicitly mentioned, the need for user permission through Google accounts and token generation raises potential security and privacy concerns that need to be addressed and communicated effectively.

**Design Version - 2 :**

In response to the aforementioned challenges, our team identified a potential solution in the form of an application named Health Connect. This application bears similarity to the Health Kit of IOS and is designed to collect health data from various trackers, storing it in local storage. Accessibility to this data, including the option to overwrite our own data, is facilitated through Health Connect. It is imperative to note, however, that Health Connect is currently in the developmental stage.

To interact with the stored data, permissions are required, adding an additional layer of complexity to the process. Furthermore, the absence of a robust community support system poses a challenge in resolving potential queries or concerns. Complicating matters further, the documentation is predominantly written in the Kotlin language, necessitating a shift in our programming language to align with the requirements of the application.

After careful consideration, the decision was made to embark on the development of a dedicated application designed to access user health data. Subsequently, we plan to establish a local server to retrieve the data, enabling the implementation of the Machine Learning model for the derivation of meaningful insights.

***Strengths:***

1. Adaptive response to challenges: Design Version - 2 demonstrates the team's ability to adapt and respond to challenges by identifying a potential solution in the form of Health Connect, an application designed to collect health data.

2. Local storage solution: Storing health data in local storage through Health Connect may enhance data accessibility and provide more control over data management, potentially addressing the challenges faced in Design Version - 1.

3. Strategic shift in approach: The decision to develop a dedicated application and establish a local server to access user health data indicates a strategic shift in the approach, potentially overcoming limitations associated with external APIs.

***Deficiencies:***

1. Developmental stage of Health Connect: The fact that Health Connect is still in the developmental stage introduces uncertainty and potential risks, as the application may not be fully stable or feature-complete.

2. Additional complexity with permissions: The requirement for permissions to interact with stored data adds an extra layer of complexity to the process, potentially causing user friction and posing challenges in the implementation phase.

3. Lack of community support: The absence of a robust community support system for Health Connect may pose challenges in resolving queries or concerns, potentially impacting the project's progress and troubleshooting capabilities.

4. Language shift challenges: The need to shift programming language to align with the Kotlin-based documentation introduces challenges, such as potential delays, additional learning curves, and adjustments in the development process.

5. Limited information on security: The design does not explicitly address security measures for handling health data, and it is crucial to ensure that the transition to a dedicated application and local server does not compromise data security and privacy.

**Design Version - 3 :**

The preceding version of the design was formulated autonomously due to prolonged non-responsiveness from the client. In the absence of client input, several assumptions were made, and the aforementioned design was implemented accordingly. The focal point of our efforts was centered around extracting data from the device. Upon finally meeting with the client, it became apparent that she was dissatisfied with the project. Her concerns primarily revolved around Health Connect, as she perceived it to be a third-party application. Consequently, she recommended extracting raw data directly from the smartwatch, bypassing reliance on external third-party applications. Although we were granted permission to utilize Google Fit data, this directive necessitated a significant redesign of the project, which unfortunately transpired later than optimal.

The revised design now entails accessing raw data from the smartwatch and identifying any anomalies within it. Subsequently, we obtain data from Google Fit to ascertain whether the patient exhibits signs of chronic diseases. Challenges arose during the extraction of Google Fit data, prompting the utilization of Google Fit Android APIs for access. However, it became apparent that this solution was functional exclusively on mobile devices. For the extraction of sensor data, React Native was employed. Ultimately, a decision was reached to consolidate all extracted data for comprehensive analysis and subsequent implementation.

***Strengths:***

1. Adaptability to client feedback: Design Version - 3 shows the team's flexibility in adapting to client feedback, particularly regarding concerns about Health Connect, leading to a significant redesign of the project.

2. Focus on direct smartwatch data extraction: The revised design prioritizes direct extraction of raw data from the smartwatch, aligning with the client's preference and potentially improving data accuracy.

3. Permission granted for Google Fit data: The team obtained permission to utilize Google Fit data, showcasing effective communication and collaboration with the client to address concerns and make necessary adjustments.

4. Comprehensive analysis approach: The decision to consolidate all extracted data for comprehensive analysis suggests a holistic approach to deriving meaningful insights from the collected information.

***Deficiencies:***

1. Autonomously formulated design: The initial formulation of the design without active client input led to assumptions that later proved to be misaligned with the client's expectations, highlighting the importance of consistent communication and collaboration throughout the project.

2. Delayed client interaction: Prolonged non-responsiveness from the client resulted in a delayed understanding of their preferences and concerns, leading to a redesign that transpired later than optimal.

3. Challenges in Google Fit data extraction: The difficulties faced during the extraction of Google Fit data, specifically the reliance on Google Fit Android APIs functional exclusively on mobile devices, introduces potential limitations and complexities in the project implementation.

4. Technology stack diversity: The use of both React Native for sensor data extraction and Google Fit Android APIs for health data retrieval introduces a diverse technology stack, which may pose challenges in integration, maintenance, and consistency across platforms.

5. Unclear mention of security measures: The design does not explicitly mention security measures for handling health data extracted from the smartwatch and Google Fit, and it's crucial to address potential privacy and security concerns in the implementation.

**Design Version - 4 :**

Upon further investigation, we discovered that Health Connect functions not as a standalone application but rather as an API, providing access, control, and data storage capabilities. Remarkably, even Google Fit requires Health Connect's authorization to store data on mobile devices. Recognizing the security and data aggregation benefits offered by Health Connect, we concluded that accessing its data would be a more prudent choice.

To implement this decision, we successfully developed a React Native application capable of accessing raw sensor data and interfacing with Health Connect data. Consequently, our chosen approach involves using React Native to construct the application, allowing seamless access to health data. The acquired data will be securely stored in a Firebase database, which will serve as both the backend service and the authentication mechanism. we are using Firebase because it is fast, easy to use and globally accessible.

Our comprehensive strategy extends to the utilization of TensorFlow for the execution of the Machine Learning model on the data stored within the Firebase database. The final step involves deriving meaningful insights from the processed data and presenting them within the application interface. This approach ensures a secure, efficient, and integrated solution for the extraction, storage, and analysis of health data.

***Strengths:***

1. Integration of Health Connect API: Design Version - 4 highlights the recognition of Health Connect's capabilities as an API, providing access, control, and data storage, which can enhance security and data aggregation benefits.

2. Prudent choice based on security considerations: The decision to access data through Health Connect is based on security considerations, indicating a thoughtful approach to data handling and storage.

3. Seamless access to health data: The use of React Native to construct the application allows for seamless access to health data, providing a user-friendly experience.

4. Utilization of Firebase for backend and authentication: The choice of Firebase as the backend service and authentication mechanism is based on its speed, ease of use, and global accessibility, contributing to the efficiency of the solution.

5. Integration of TensorFlow for Machine Learning: The inclusion of TensorFlow for executing the Machine Learning model on data stored in the Firebase database demonstrates a comprehensive strategy for deriving meaningful insights from health data.

***Deficiencies:***

1. Limited mention of potential challenges: The design does not explicitly mention potential challenges or considerations that might arise during the implementation phase, leaving room for uncertainty in addressing unexpected issues.

2. Lack of details on data security measures: While Firebase is mentioned for data storage, the design does not provide specific details on the security measures implemented to ensure the confidentiality and integrity of health data.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_