Project Requirements Document v2

Team #stub In Collaboration With Teladoc Health

Benjamin Lee
Dylan Kupsh
Paul Kuang
Jonathan Xu
Rohan Aren

Introduction

Background

Teladoc Health is a company that focuses on enabling virtual healthcare. Annually, Teladoc conducts over 10 million telehealth consultations for general medicine, behavioral health, dermatology, and acute care. In recent times, reliable, quality telehealth services are needed more than ever. To accommodate this growing demand, Teladoc Health has proposed the development of software that interfaces with consumer health peripherals and records the user's status so doctors can have readily available data rather than requiring patients to come in for a physical checkup.

Telehealth consults offer several logistical advantages over in-person consultations. Since neither the physician nor patient need travel, virtual meetings provide easy and convenient scheduling. Additionally, the medical equipment and hospital resources used in regular doctors appointments can be reallocated for other tasks. Limiting the number of in-person hospital visits also prevents the spread of disease, which can ease patients who would normally be too afraid to come into the hospital. During the Covid-19 pandemic, these advantages have caused the frequency of telehealth consultations to skyrocket.

However, telehealth consultations have one main drawback. Currently, remote consultations are limited by the lack of medical instruments that would normally be accessible in a doctor's office or other physical location. A key aspect of a normal doctor's visit is a general wellness checkup, which can involve measuring heart rate, respiratory rate, and other physical assessments. Without precise medical instruments available during a telehealth call, doctors have a limited amount of objective information to make a preliminary diagnosis, offer treatment suggestions, or provide treatment plans. Physicians often have to rely on patient testimony alone to understand a patient's situation, which can be unreliable. Consumer health peripherals, including Apple Watches, FitBits, and mobile phones, provide physicians with an alternative method of gathering objective health data. These devices hold a lot of latent, useful medical information that can be leveraged to offer physicians improved insight into a patient's situation, even over a telehealth call. By taking data from these peripherals and relaying it to Teladoc's patient-physician communication software, this project can

become a convenient solution that brings virtual telehealth consultations closer to the experience of a physical, in-person variant.

What is the problem?

In the virtual environment, there is no way for physicians to directly take a vital reading or other physical assessment of a patient's condition. The only metric by which a physician can make an initial diagnosis or suggest a treatment plan is patient testimony, which can raise questions of reliability and accuracy. Patients may not be aware of terminology to describe a certain symptom, or might miss subtle indicators that would normally be picked up by hospital equipment. Even if a patient has medical equipment at home, there is always a chance of misinterpreted or misspoken information that can jeopardize the accuracy of the device's measurements. A direct interface between telehealth software and at-home peripherals would alleviate these concerns and provide physicians with some degree of certainty that the information they are receiving is objective and direct from the source.

Connecting directly with consumer health peripherals also offers some information that cannot be measured in the hospital. For example, a physical exam can only record the patient's status at that moment, but there is no way to know the patient's condition over a period of time. Consumer health peripherals like the Apple Watch and FitBit provide this service, but information stored by these devices is not readily available for doctors to access. Ideally, our project would provide an easily accessible interface for physicians to view data not just at the time of a telehealth consult, but leading up to and possibly even following a meeting, such as to track the effectiveness of a treatment plan in the days or weeks after a consultation. There is also potential for further back-end analysis and processing of peripheral data after collection, making many useful applications of this information possible.

How is the problem addressed today?

During virtual consultations, if physicians want to collect patient vitals, a patient must either manually relay this information or rely on third-party software. Some forms of data, such as an EKG, cannot be manually described. Additionally, human error in measuring or relaying any information can cast doubt on its reliability. Moreso, while

third-party services with the purpose of recording and transmitting this health data exist, they are often costly and not directly integrated with the communication software, making them clumsy and inconvenient for physicians to use. There are also no in-depth means to analyze this data besides a manual inspection.

Goals

The project's main goal is to bridge the gap between physical patient-physician interactions and telehealth consults provided by services such as Teladoc's Solo application. A major step in bringing virtual consultations closer to the physical experience is allowing doctors to take real-time measurements of a patient's condition, collecting information such as heart rate, respiratory rate, and temperature. Additionally, due to their continuous collection of data, interfacing with health peripherals has the potential to offer information not normally available during a doctor's visit, such as health trends over time both leading up to an interaction and following a prescription or treatment plan. Finally, the project shall provide meaningful analysis of this data both during an appointment and over its collection period.

How will it be done?

The project consists of two main components: a mobile application implemented with React Native to receive data from consumer health peripherals, and an interface between this mobile application to Teladoc's communication software in order to make this information available to healthcare providers. Because team members have access to different mobile device OSs, the project will build off of React Native as a convenient middle ground that can be easily deployed to both iOS and Android devices. As patients go about their daily lives, their peripherals continuously collect information regarding their activity level and vitals. Using existing peripheral APIs and Teladoc Cloud, our application will allow users to consolidate their vitals data in a single location, from which physicians will be able to directly access objectively collected health information over a virtual format.

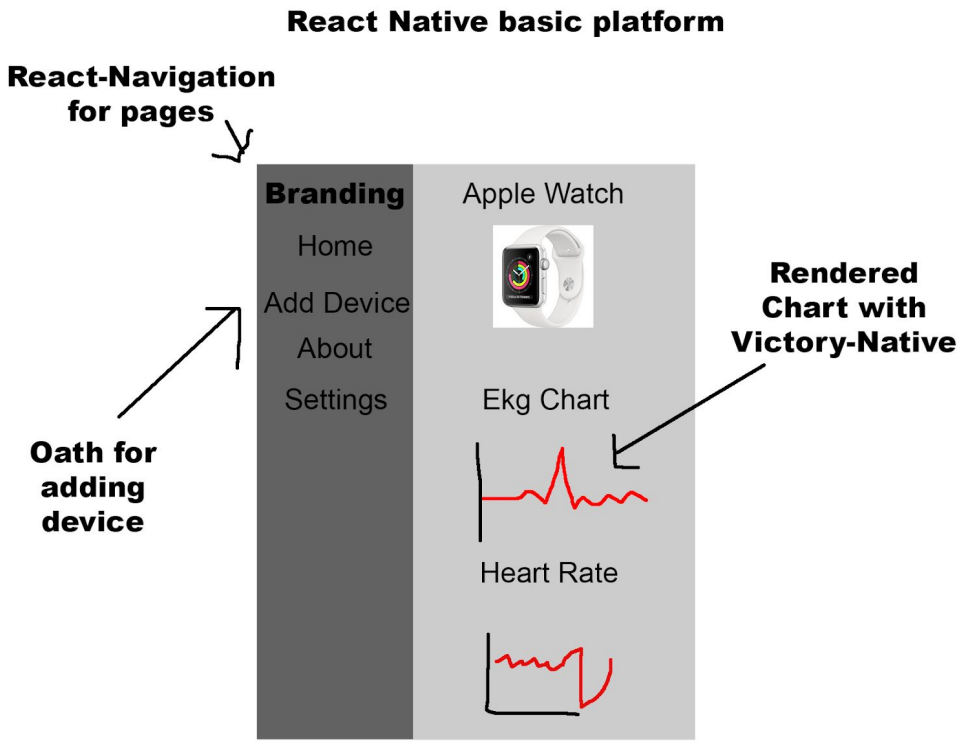
Assumptions

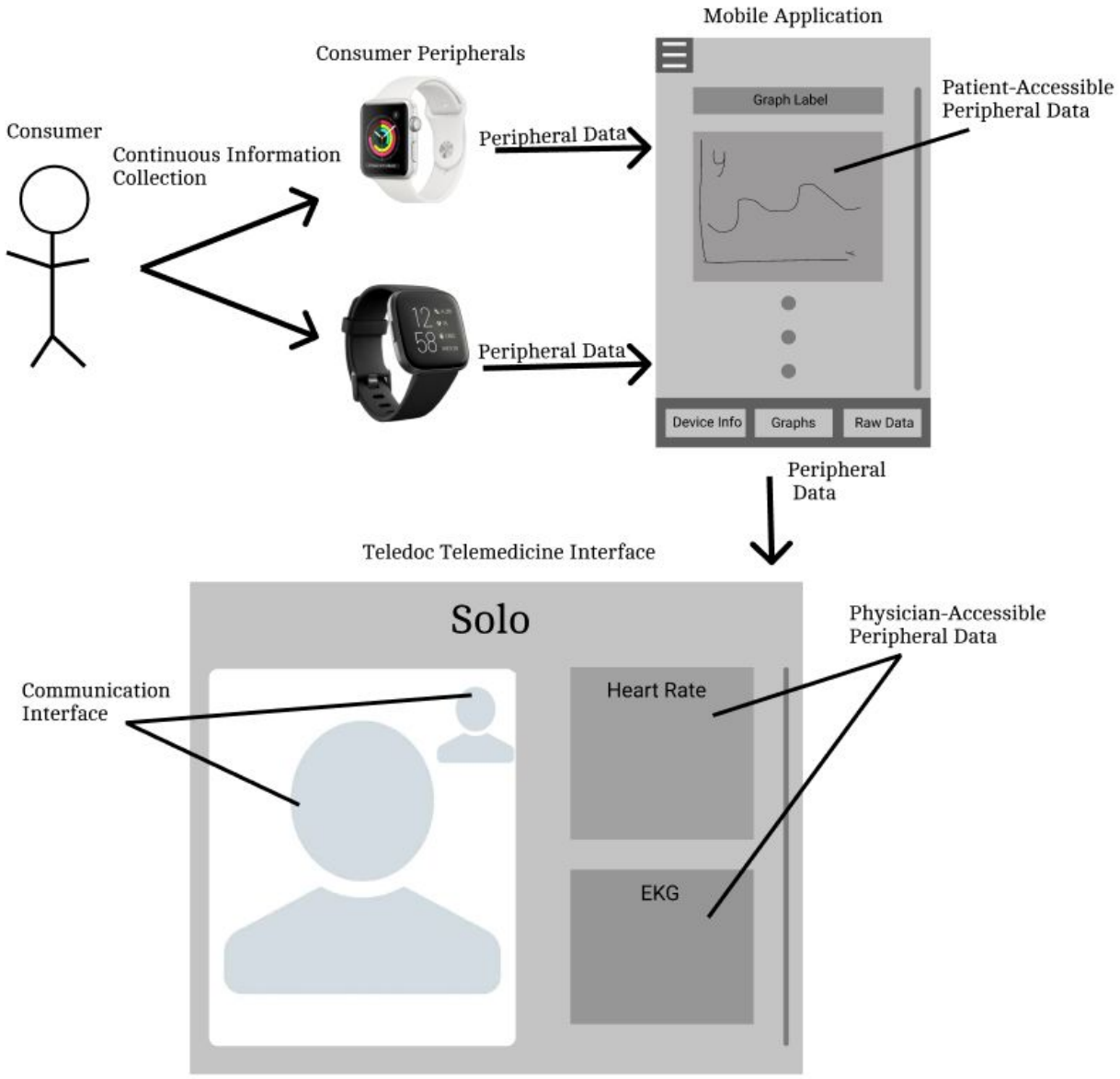
- Peripheral data is reliable and accurate

- Patients utilize their peripherals enough to provide a minimum amount of information
- Patient data will always be accessible through peripheral APIs

System Architecture

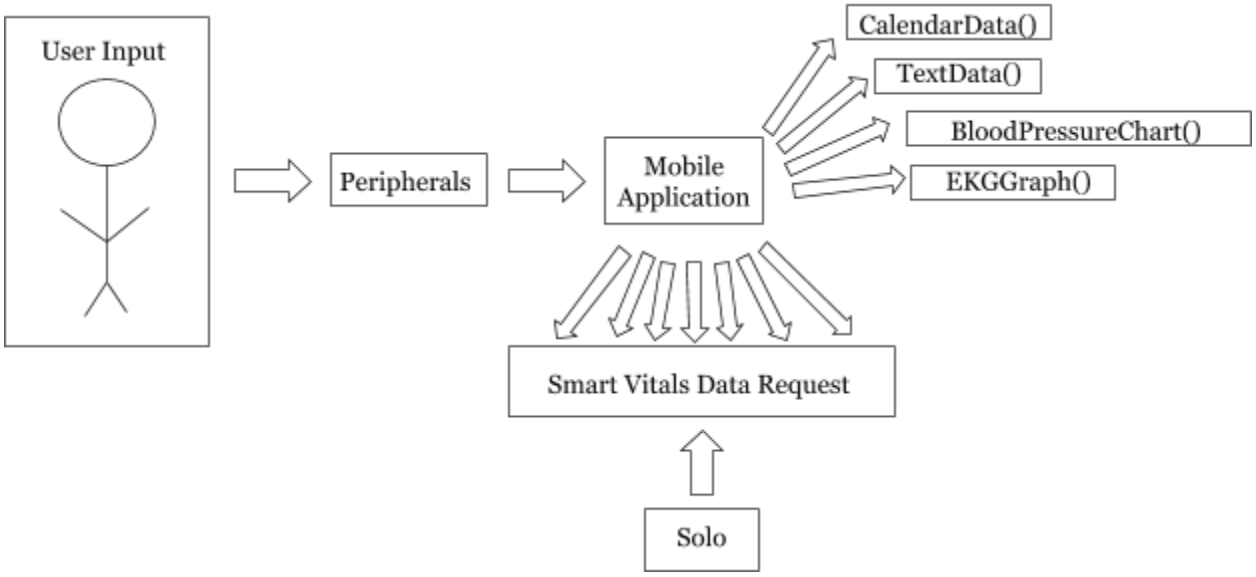
Our mobile application will collect data from consumer peripherals by utilizing their respective APIs. The application will then interface with Teledoc's communication software to relay appropriate data.



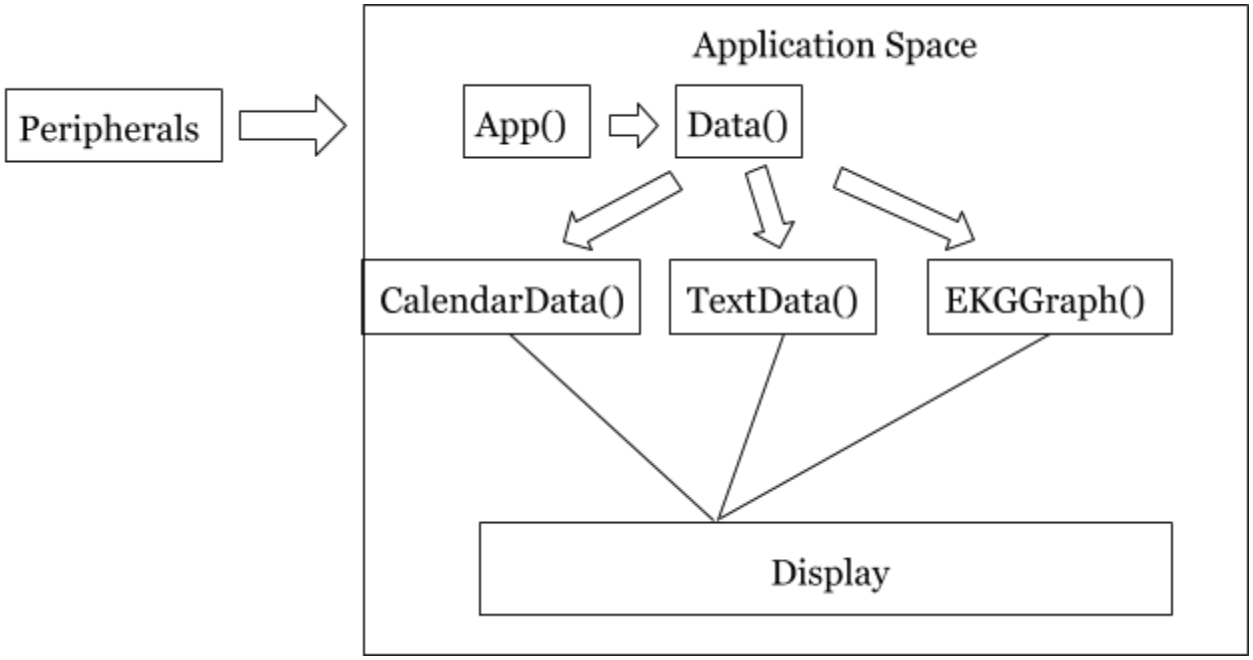


Sequence Diagrams

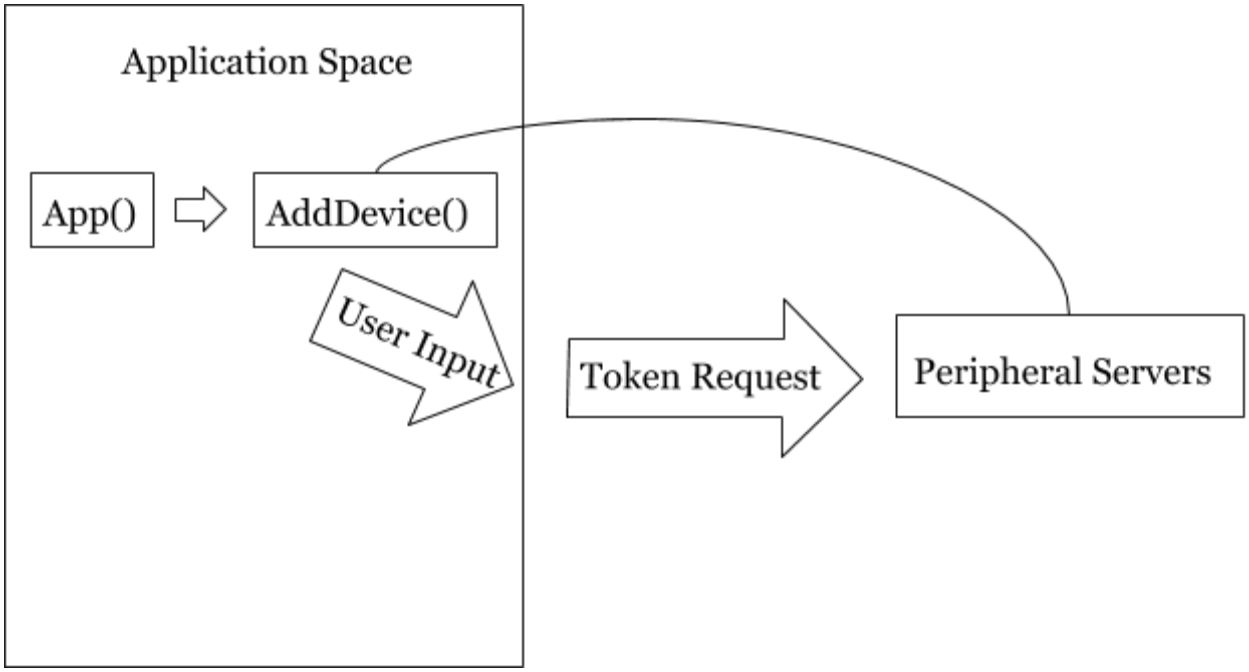
Overview



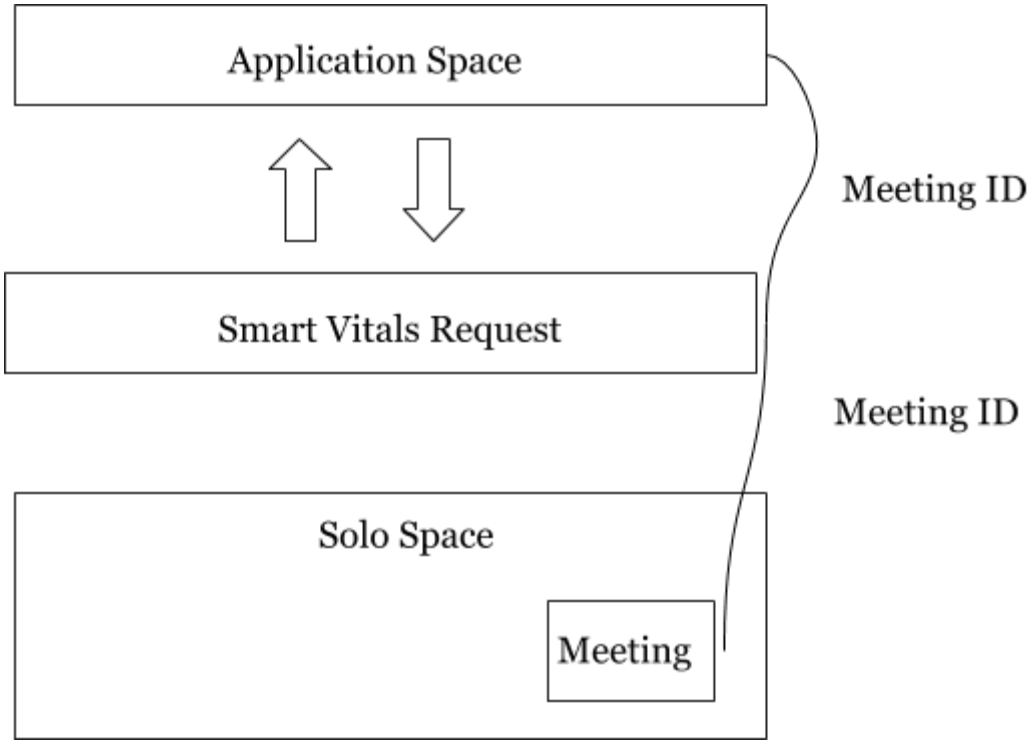
Displaying Data



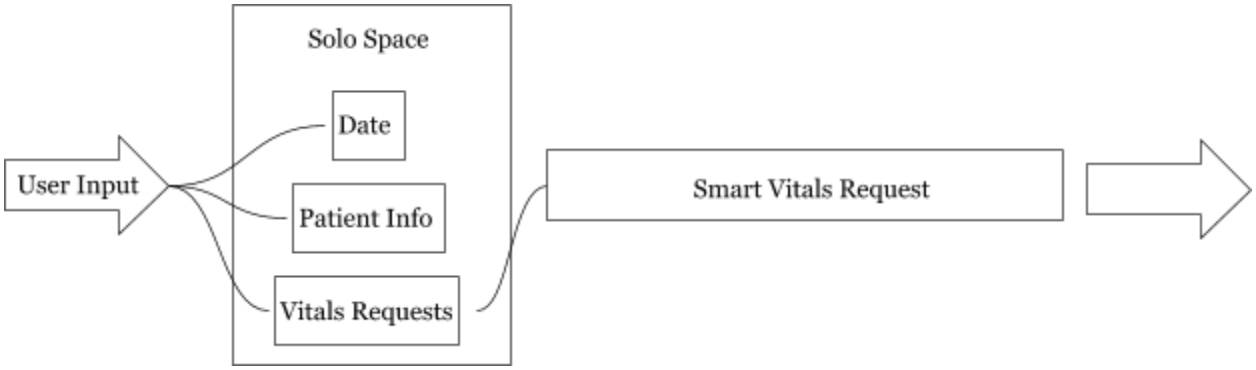
Adding A Peripheral



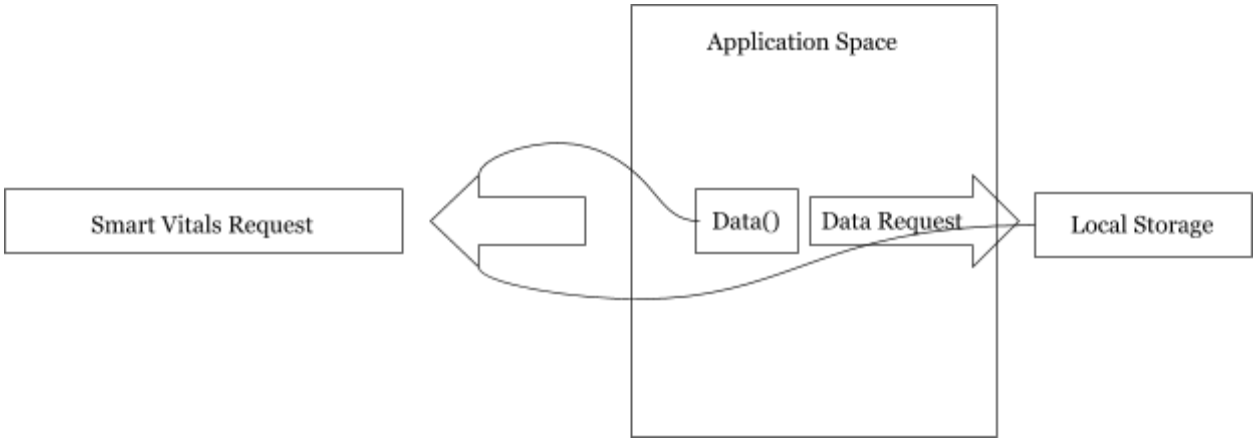
Data Request Flow



Appointment Data Request (Physician Side)



Appointment Data Request (Patient Side)



User Stories

Green entries indicate completion.

Highest Priority:

Story/Task	Testing Criteria
1-H Implement FitBit OAuth: As a Patient, I can link my FitBit account to easily give the app permission to access my personal data. Time Estimate: 2 days Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/42 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/21	Scenario: User has a FitBit account and Internet connection. ● User goes to Add a Device screen from the Hamburger Menu. ● User taps on FitBit option in the list. ● Open up an OAuth session that links to FitBit OAuth page. ● User should be redirected back into the app with a custom url scheme after a successful sign-in, with a valid access token for API usage.

	Scenario: No Internet connection. ● When opening up an OAuth session, the browser should notify the user that they need to check their internet access.
2-H Support Syncing Data from FitBit: As a Patient, I can use our app to fetch my FitBit data such that I can get a clear understanding of recent vital trends. Time Estimate: 2 days Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/2 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/21	Scenario: User has linked FitBit account as per 1-H. ● Navigating to the Devices page shows a history of vitals information in an appropriate format, such as a chart or calendar format. Scenario: User has not linked FitBit. ● The activity should show text that guides the user into syncing their vitals data.
3-H Fetch Apple Health Info: As a Patient, I can use our app to find my Apple Health information such that I can get a clear understanding of recent vital trends. Time Estimate: 2 days Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/3 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/35/commits	Scenario: User enters health data to app for the first time through Apple Health ● Standard app permissions on iOS will pop up asking if app can access these new data fields Scenario: User has given permission to access Apple Health info. ● Navigating to 'Add Device' page shows all Apple Health data given by user and based on timeframe
4-H Fetch Blood Pressure: As a Patient, I want to view my blood pressure changes during the day, so that my doctor can view my health status and have a more accurate diagnosis.	Scenario: User has network connectivity, has blood pressure measurements from peripherals. ● App should retrieve blood pressure history when attempting to sync. ● This information should be

Time Estimate: 3 days Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/4 Completed:	viewable to the user and over Solo if permission was given. Scenario: User has no network connectivity, but cached data exists. ● App should retrieve blood pressure history from secure cache system. Scenario: No blood pressure data found. ● The Devices->Charts and Devices->Data pages should display "No blood pressure measurements found."
5-H Fetch Weight: As a Patient, I want to view my weight data so that I can track changes over time. Time Estimate: 1 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/43 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/41	Scenario: User has recorded weight data in FitBit or Apple Health. ● Fetch the relevant weight data from FitBit API/Apple Health/Secure Store. ● Display the weight data in the calendar screen inside a TextData component. Scenario: User has not recorded weight data for the selected date. ● Weight data will be absent or hold default "No Data" string value.
6-H View Heart Rate Data: As a Patient, I want to view my heart rate data so that I can track changes over time. Time Estimate: 2 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/50 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/20	Scenario: User wishes to view their heart rate data over time in a simplified format. ● There will be a graph with the heart rate data in a simplified format ● The graph should use sample heart rate data.
7-H Create Basic Information Display Component with Single Title and Data Value: As a patient, I	Scenario: User wishes to view a single piece of data with a label and value. ● Render an information display

want to be able to view pulled single pieces of peripheral data in a sensible format. Time Estimate: 1 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/44 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/18	component with the provided label and value taken from the peripheral.
8-H Create Vital Graphs: Use Victory Native to create graphs for mapping vitals over time. Time Estimate: 2 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/45 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/51	Scenario: User wants to see graphs of peripheral data ● Render a component using the peripheral data and Victory Native. The graph should change based upon the data provided.
9-H Create Component to Display Data over a Period of Time: As a patient, I want to be able to view data collected over a period of time through an intuitive display. Time Estimate: 1 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/36 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/26	Scenario: User wishes to view the same type of data collected over a period of time. ● Render an information display component in the form of a calendar. ● Display appropriate information after user input when a day is pressed.

10-H Allow Patient to Doublecheck Data: As a patient, I want to be able to verify if my information is correct before sending. Time Estimate: 1 day Issue Link: https://github.com/benjaminleeo/InterfacingConsumerHealthPeripherals/issues/6 Completed:	Scenario: User is about to manually send information to app ● Display a prompt asking if information is correct and give option to go back if user wants to change info again
11-H Add App Notifications when Concerning Data is Found: As a patient, I can track symptoms as monitored by health peripherals so that I can be informed if I am displaying signs that I should seek medical attention. Time Estimate: 2 days Issue Link: https://github.com/benjaminleeo/InterfacingConsumerHealthPeripherals/issues/7 https://github.com/benjaminleeo/InterfacingConsumerHealthPeripherals/issues/37 Completed:	Scenario: Data shows concerning symptoms. ● Display a notification from the app: "Concerning vitals data found." ● Allow users to note down the symptoms for their next appointment. ● Users can view concerns in the Devices page, if they navigate to the Concerns section in the bottom bar. Scenario: The app detects concerning vitals data prior to sending. ● Backend pre-processing should note concerning items as a higher priority. ● Backend analysis should highlight the prioritized information with a [High Priority] tag.
12-H Add Jest Tests: Use Jest for unit tests and component rendering. Time Estimate: 3 days Issue Link: https://github.com/benjaminleeo/InterfacingConsumerHealthPeripherals/issues/46 Completed:	Details: ● Unit Test for individual functions or classes ● Combine several modules of unit test and create an integration test ● Testing Rendered Output for components ● Testing User Interactions: handle events like onChangeText for TextInput or onPress for Button.

https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/32	

Medium Priority:

Story/Task	Testing Criteria
1-M Fetch Permissions: As a Patient, I want to have control over what data to sync so that I can feel like my data is private. Time Estimate: 1 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/1 Completed: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/pull/21	Scenario: User is linking FitBit. ● Users will select what scopes to allow. ● Ensure that only approved scopes can be fetched with the API. Scenario: User is linking Apple Health. ● User will be notified of necessary permissions. Scenario: Data fails to send/fetch. ● Check network connectivity and prompt the user to check connection if offline. ● Otherwise, retry the sending/fetching operation once. ● If the send fails again, inform the user of the error with "Failed to send/fetch data."
2-M Add Secure Caching: Add secure caching to reduce the number of API calls and keep track of historical data. Time Estimate: 3 days Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/47 Completed:	Details: ● Will most likely use *expo-secure-store*. ● Note that entries have a *2kb size limit* with this module.
3-M Create List of Common Healthy Vital Ranges: As a Patient, I want to check metrics with baselines and averages to better understand the severity/level of their health.	Scenario: Metrics within healthy ranges. ● Backend analysis does not return any areas of concern. Scenario: Patient has metrics outside healthy ranges. ● Backend analysis returns patient

Time Estimate: 1 day Issue Link: https://github.com/benjaminleeo/InterfacingConsumerHealthPeripherals/issues/9 Completed:	vitals info, with a message that states “Data outside healthy range of X to Y” where X and Y are the limits of the range. • Change formatting of unhealthy data to stand out.
4-M Create Page to Allow Users to Create Own Data Categories and Manually Input Data: As a patient, I want to enter custom or arbitrary data into the app that might not be measured by peripherals. Time Estimate: 2 days Issue Link: https://github.com/benjaminleeo/InterfacingConsumerHealthPeripherals/issues/38 Completed:	Scenario: Patient wants to enter pain level for a particular day. • App allows users to input arbitrary values for specific data. Scenario: Patient wants to indicate that they are experiencing a certain symptom on a given day. • App allows users to create data categories and provide manual input for each day.
5-M Generate Push Notifications over a period of time specified by user to collect self-inputted data: As a patient, I want to be able to set reminders over a period of time to enter data for a particular custom data category. Time Estimate: 2 days Issue Link: https://github.com/benjaminleeo/InterfacingConsumerHealthPeripherals/issues/39 Completed:	Scenario: Patient believes they are getting sick and wishes to input pain level and their symptoms over a period of two weeks. • App allows users to specify a period of time they wish to collect self-inputted data, and will produce a notification reminder to input data once per day.
6-M Detect Local Outlier Data after Syncing: As a Patient, I want to be notified when the app finds inaccurate data, so that I can fix it before potentially sending it to Solo.	Scenario: Concerning data is clearly impossible. • Ex: If the user suddenly records a weight of several thousand pounds. • Filter out the data when it is synced.

Time Estimate: 1 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/5 Completed:	• If newer, more reliable data is not present, prompt the user to record it again.

Lowest Priority:

Story/Task	Testing Criteria
1-L Add Google Fit OAuth: As a Patient, I want to link my Google Fit account so that the app is able to fetch data. Time Estimate: 3 days Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/48 Completed:	Scenario: User has a Google Fit account and Internet connection. • User goes to Add a Device screen from the Hamburger Menu. • User taps on Google Fit option in the list. • Open up an OAuth session that links to Google Fit OAuth page. • User should be redirected back into the app with a custom url scheme after a successful sign-in, with a valid access token for API usage. Scenario: No Internet connection. • When opening up an OAuth session, the browser should notify the user that they need to check their internet access.
2-L Fetch Google Fit Data: As a Patient, I want to fetch information from my Google Fit account so that I don't have to manually enter pre-recorded data. Time Estimate: 1 day Issue Link: https://github.com/benjaminlee0/InterfacingConsumerHealthPeripherals/issues/49	Scenario: User has linked Google Fit account as per 1-L. • Navigating to the Devices page shows a history of vitals information in an appropriate format, such as a chart or calendar format. Scenario: User has not linked Google Fit. • The activity should show text that

Completed:	guides the user into syncing their vitals data.

Appendix

Technologies employed:

React Native: Develop cross platform mobile application

Jest: Testing for React Native

Expo: Command line environment. Serve, build and publish Expo projects

Victory Native: Library that contains a set of modular charting components for React Native to create charts and graphs

TypeScript/JavaScript: Front-end elements, basis for React

Apple HealthKit API: Used to track user's health data

Fitbit API: Used to track user's health data

Expo Auth Session: Used to handle OAuth 2 authorization flows