

# #stub

Teladoc Health

---

Ben Lee, Rohan Aren, Jonathan Xu, Paul Kuang, Dylan Kupsh

# Our Problem

- With COVID-19, telehealth consultations have become increasingly important
- Doctors cannot directly take patient vitals in this setting
- Doctors are waiting for patient's health data and cannot use their time efficiently
- Patient testimony can be inaccurate, unreliable
- Information in consumer health devices is not easily accessible to doctors

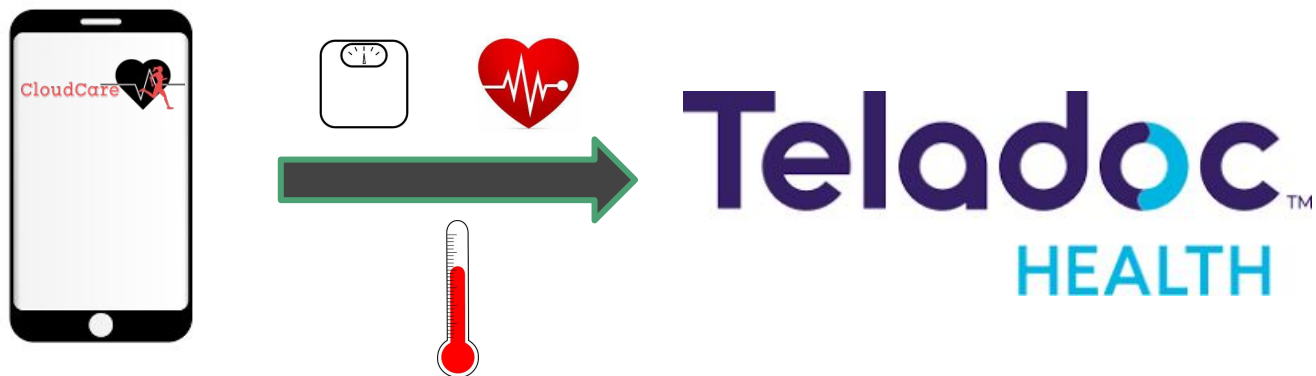
## Practical Goals

- Foster meaningful, productive patient-physician virtual consultations
- Reduce hospital workloads by offering alternatives to in-person appointments
- Streamline and introduce reliability to self-vital collection



# Our Solution

- Create a mobile app that gathers health data from various consumer peripherals and user input
- Interface with Teladoc's platforms to consolidate data and format for healthcare providers to use



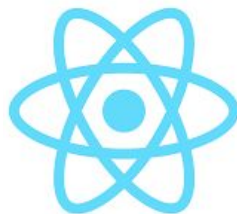
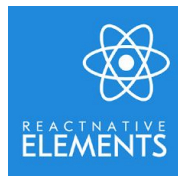
# Demo

The screenshot displays the Teladoc MyChart patient portal. The interface is divided into several sections:

- Header:** The Teladoc MyChart logo is in the top left, and a search bar is in the top center.
- Left Navigation Menu:** A dark sidebar contains icons and labels for: Dashboard, Queue, Schedule, Home, Analytics, Encounters, and Waiting Rooms.
- Search Results:** A search bar at the top of the main content area has yielded results for "World, Herta". The results are displayed in a list format with the following details:
  - Status:** Active
  - Yes**
  - Insurance:** N/A
  - Vitals:** (expandable section)
  - Insurance:** (expandable section)
  - Episodes of Care:** (expandable section)
  - Family:** (expandable section)
- Right Panel:** A detailed view for "World, Herta" is shown, including:
  - Header:** World, Herta
  - Metadata:** Member ID, Age, Sex, and Date of Birth.
  - Active Visits:** A section with tabs for Visits, Upcoming Visits, and Episodes.
  - Documents:** A section for documents related to the patient.

# Technical Details

- React Native framework to make our app iOS and Android compatible
  - Victory Native and React Native Elements also used for front-end
- Apple HealthKit and FitBit APIs fetch user data from respective peripheral devices
  - Expo Secure Store as current secure storage solution
- Smart Vitals API to bridge with Teladoc Solo



# Challenges

- **Compatibility**
  - Cross-platform concerns: Expo Client, Android Standalone, iOS Standalone
  - Expo and React Native conflicts
  - Emulator vs Physical Device
- **Accessibility**
  - Reduce need for user input
  - Minimize clutter (separate apps, etc)
- **Efficiency**
  - Minimize API calls while keeping data secure

# Next Steps for 189B

- Expand vitals collection with physical devices
- Google Fit OAuth
- Add notifications to remind users to submit vitals
- Perform backend analysis of collected data
- Add more testing for the data components





Questions?