VeinViewer: DVT Detector

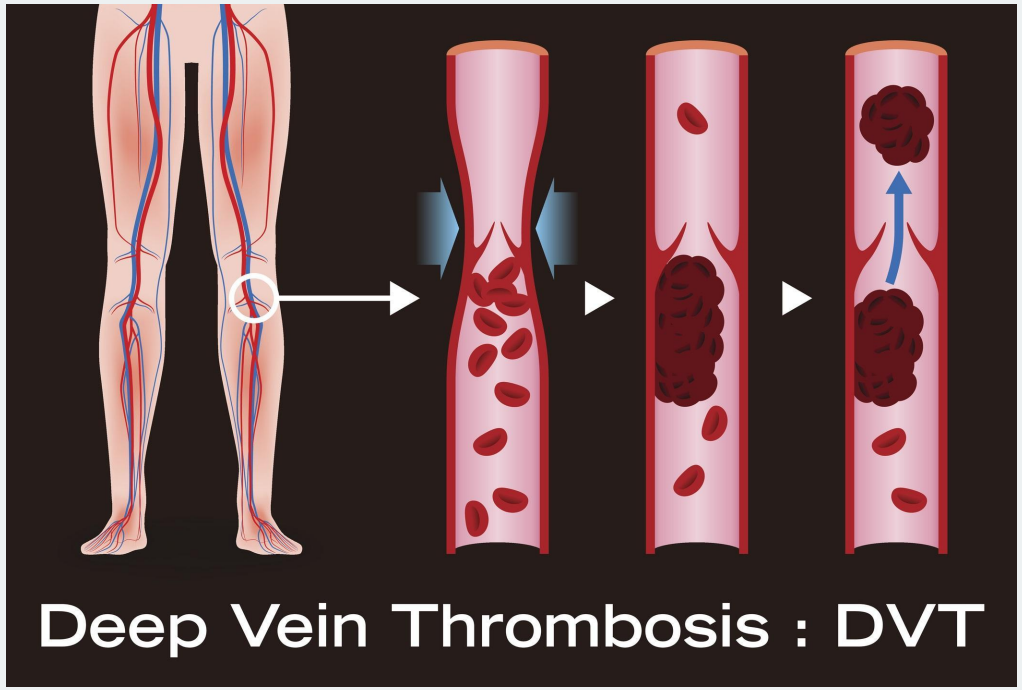
Brenton Li, Luke Johnston, Muyi Jin

Project Overview

- High-Level Summary highlights and updates of Project Plan
  - Software Requirements
  - Design
  - Testing
- Finalized Product Backlog and Project Burndown Chart
- Summary of Software Engineering Concepts
- Project Lessons Learned
- Configuration Management Items in Product Software Library
- Screenshots of primary Product features
- Demo of Product

The Problem

- Deep Vein Thrombosis (DVT) occurs when a blood clot forms in the deep/inner veins (most commonly in the legs)
- If left untreated, the clot can dislodge and travel through the veins until it gets stuck in the Pulmonary Arteries, causing a Pulmonary Embolism (PE)
  - 25% of all people who have a PE will die instantly (Beckman)
- People most at risk for DVT are the elderly, immobile, people who recently had surgery, and hospitalized people
  - 50% of all blood clots occur when a person is hospitalized (CDC)



High-Level Summary of Highlights and Updates to Life-Cycle Deliverables

Project Plan

- **Roles**
  - Brenton: Scrum Product Owner, Scrum Master, Tester
  - Luke: DB Designer & Developer, Application Developer, Tester
  - Muyi: UI Designer, UI Developer, Tester
- **Resources**
  - Android Studio
  - Google Firebase
  - GitHub
  - Sketch
  - Google Docs & Google Sheets
  - Discord
- **Effort**
  - 81.5 hours





Software Requirements

Features

- Account creation and email verification
- User sign in
- PPG Test
- Wells screening
- Test records page
- More information for each record
- DVT more information
- User account modification
- Account recovery
- Settings page

Software Requirements Cont.

User Interface Requirements

- Buttons (Login, Sign up, etc)
- Information Inputs (User name, email, phone, etc)
- Information Display
  - Displays information about DVT or how to take tests
- Result Records Display
  - Users will be to see their previous results
- Profile Display
  - The profile page displays their information such as names and emails

Software Requirements Cont.

Hardware Interface Requirements

- Fully functional on all android phones, that can support the supported software.
- Does not need to access location, camera, or microphone

Software Interface Requirements

- Google Firebase Realtime Database
- Android 8 (Oreo) and above will be supported

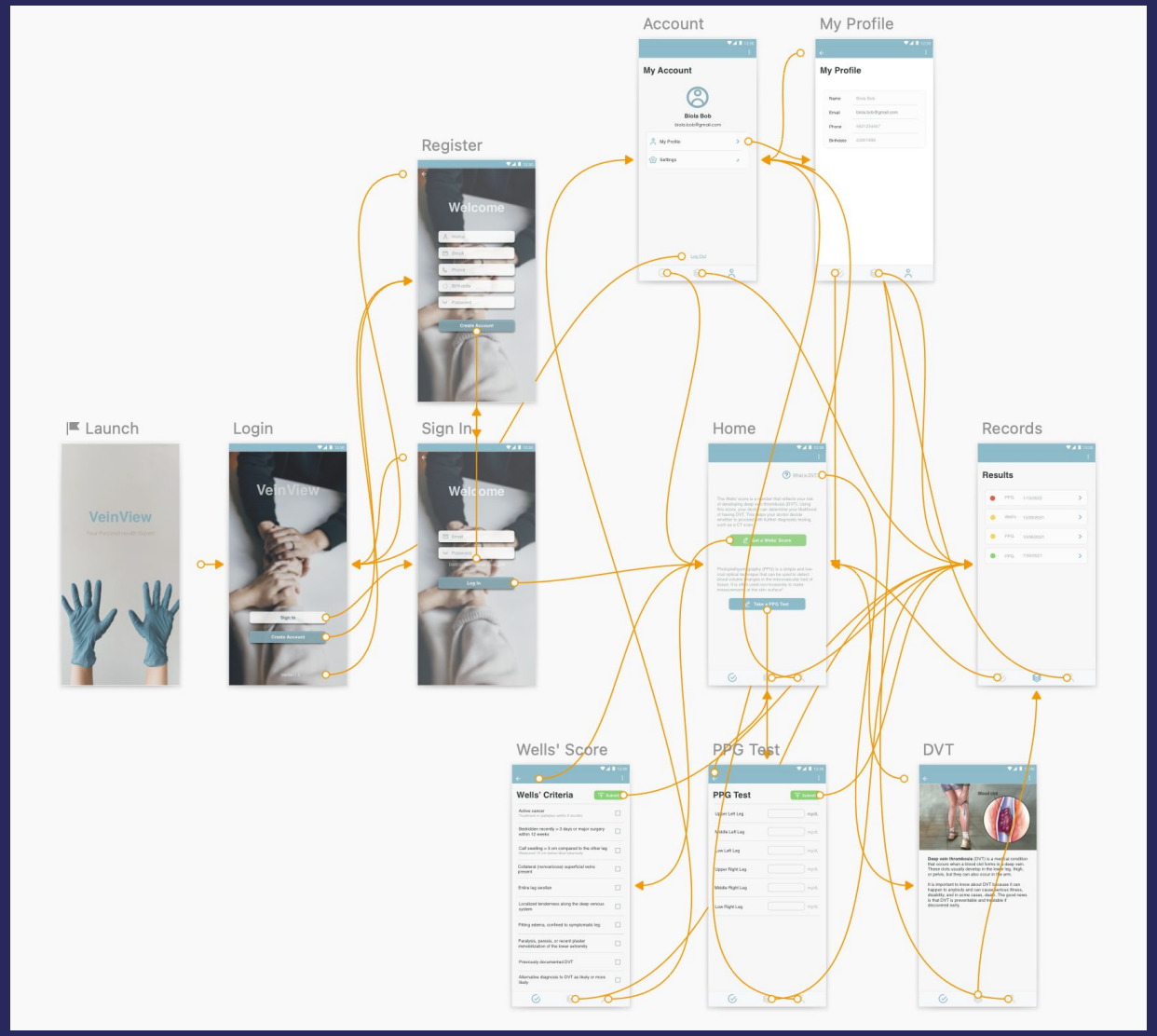
User Interface Design

Architecture Details

- Launch/Login
- Test Page
- Records/Results Page
- Profile Page
- How-to/Information Page

Components: Integrations

- **UI**: Android App UI Implementation
- **Application**: NavHost and NavHostFragment Navigation
- **Backend Database:** Google Firebase



Application Design

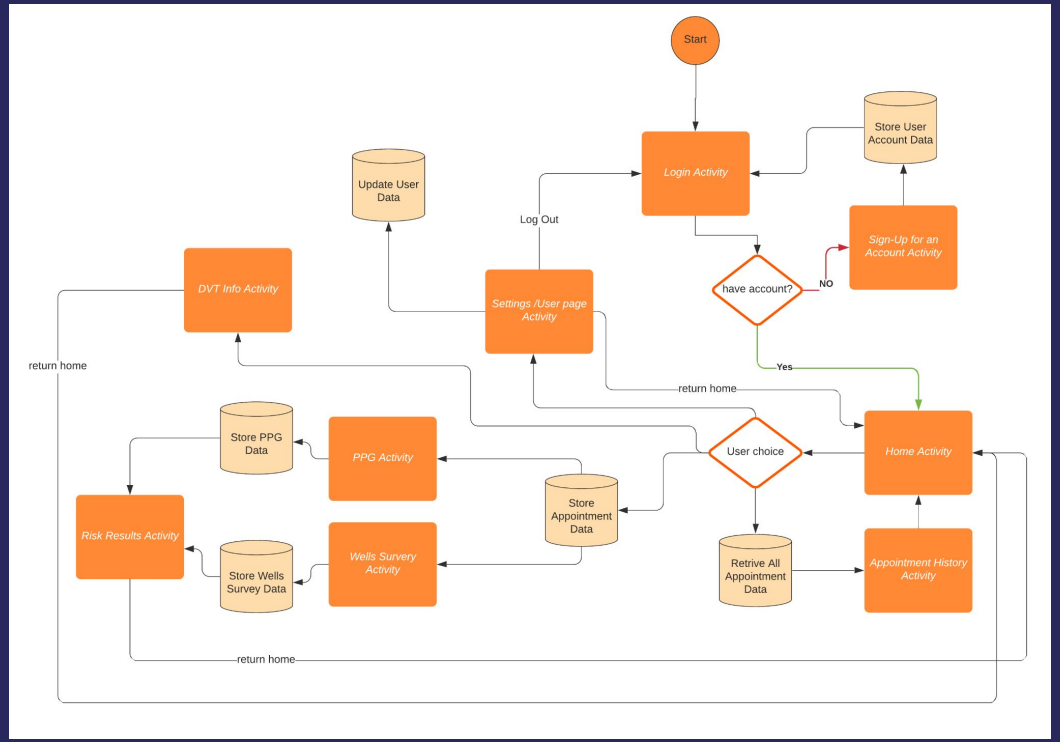
Components: Language & API

- Java
- Android 8 (Oreo) and above
- Emulator: API 32 on Pixel 5

Description

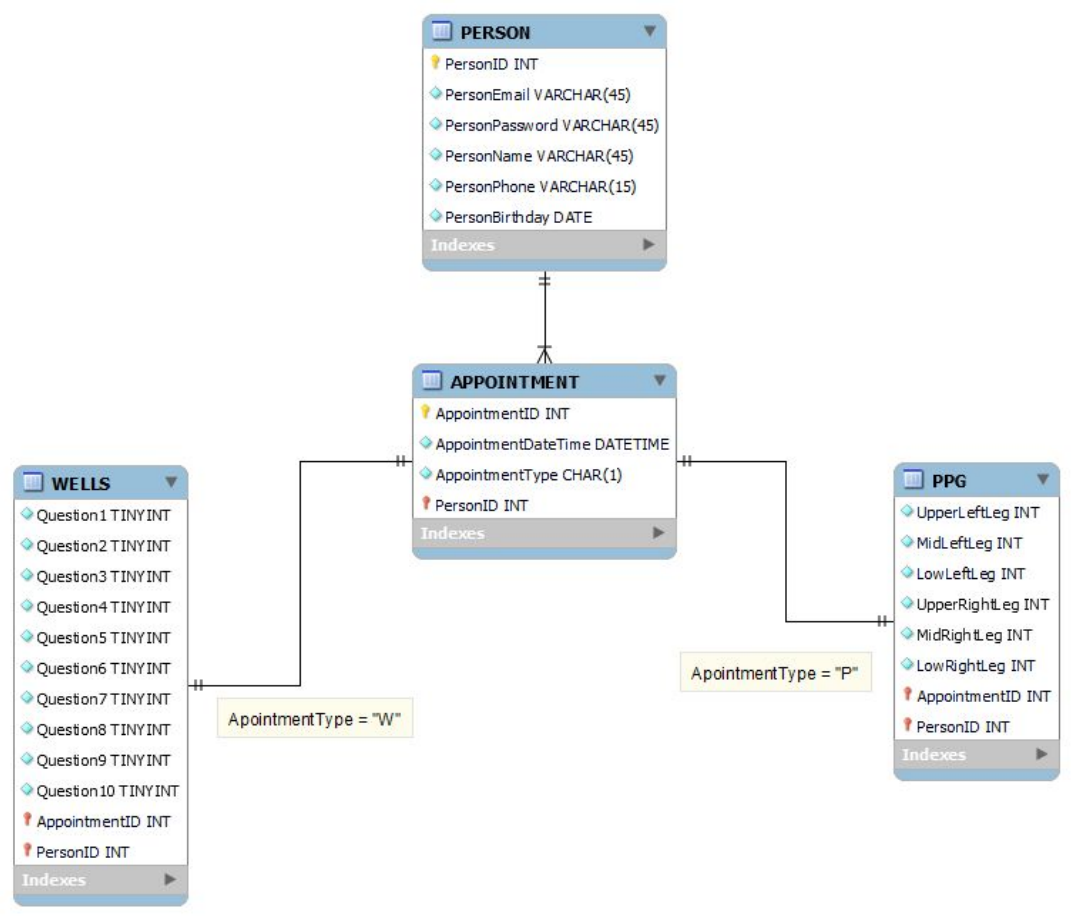
- Process all logic required for functional app
- Communicate with the database in order to store and retrieve data
- NavHost and Fragments used for every activity after user Signs in

Activity Flowchart



Database Design

EER Diagram



Description

- Firebase is a NoSQL Cloud database, No tables and primary keys
- Only verified users may login

Sprint Artifacts

Product Backlog

Scrum Product Backlog

DVT App

Version Number 1.3

April 25, 2022

Computer Science Department, Biola University

Revisions

Overview:

The purpose of the app is to provide an interface for a low-cost, high accuracy test for the presence of a deep vein thrombosis (DVT). Hospital visits for DVT diagnosis are frequent, often involve long waiting room times, and about 45% of patients presenting with the classic symptoms have diagnostic tests that return negative for DVT (medscape.com).

This Android app is used to detect deep vein thrombosis (DVT) from home. The app requires either the Photoplethysmography (PPG) measurements from a user or a 10 question Wells Survey, and then assigns them a risk level for DVT. The user may manually enter their PPG numbers that they take from measuring their legs: high, mid, and low for each leg. The user may also take the Well's survey to assess if they are likely to be suffering from DVT, if they do not have access to the tools required for PPG numbers.

Target Audience:

The team members of this project and the client, Professor Lew. Additionally, if connected to a proper working monitor/device, clients who suspect or are at risk of developing a DVT.

Project Team Members:

Brenton Li, Luke Johnston, and Muyi Jin

Version Control History:

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Brenton Li, Luke Johnston, Muyi Jin	Scrum Product Backlog	2-11-22
1.1	Brenton Li	Adjustment of Man-Hour Estimates	2-18-22
1.2	Brenton Li	Addition of task 26, Man-Hour Estimate Adjustments	2-23-22
1.21	Brenton Li	Addition of Tasks 27, 28, 29	4-1-22
1.3	Brenton Li	Man-Hour Estimates changed to Man-Hours Spent according to actual Implementation Time, Cancellation of Tasks 10.2 & 29	4-25-22

Scrum Product Backlog:

Priority	Item #	Description	Man Hours Spent	By
Very High				
	1	As an authorized user, I want to **navigate between the different pages/views in the app.**	4	MJ BL LJ
	1.1	As an unauthorized user, I want to view the log-in screen.	1	MJ BL LJ
	1.2	As an unauthorized user, I want to view the "create an account" screen.	1	MJ BL LJ
	1.3	As an authorized user, I want to view the testing screen.	1.5	MJ BL LJ
	1.4	As an authorized user, I want to view the results screen.	1	MJ BL LJ
	1.5	As an authorized user, I want to view my profile information/settings.	1	MJ BL LJ
	2	As an unauthorized user, I want to create an account (connected to the database).	3	LJ
	3	As an unauthorized user, I want to sign in to my account (connected to the database).	4	LJ
High				
	4	Create the login page UI.	3.5	MJ
	5	Create the sign-up page UI.	2	MJ
	6	Create the splash screen.	2	MJ
	7	Create the home page UI.	4.5	MJ
	8	Create the PPG test page UI.	2	MJ
	9	Create the Well's survey page UI.	2.5	MJ
	10	Create the records page UI.	2	MJ

Product Backlog

	10.1	UI Implementation: Scrollable and selectable results list	6.5	MJ
	10.2	UI Implementation: Sort and Filter options for Results List (CANCELED)		
	11	Create the profile info/settings UI.	6.5	MJ
	12	As an authorized user, I want to take a Well's DVT Screening Test.	2	MJ
	13	As an authorized user, I want to take a PPG DVT Test.	2	MJ
	14	As an authorized user, I want to be able to access my previous appointment records from the database.		LJ
	14.1	Ability to retrieve past appointment records from the database	4	LJ
	14.2	Each appointment is displayed in a separate box	4	LJ
	15	PPG Test appointment is stored in the database.	1	LJ
	16	Well's Survey appointment is stored in the database.	1	LJ
	17	The result returned after the test/survey is appropriate to the score received.	1.5	LJ
Medium				
	18	As an authorized user, I want to sign out of my account	0.5	LJ
	19	As an authorized user, I want to change my password.	2	LJ
	20	As an authorized user, I want to learn how to read my results.	2	MJ BL
	21	As an authorized user, I want to receive a notification that my test/screening result has been added to my records.	0.5	MJ
	22	As an authorized user, I want to learn about what DVT is.	4.5	MJ BL
	23	**Error Handling**	1	LJ
	23.1	System has an appropriate error message when log-in information is not found in the database.	0.5	LJ
	23.2	System has an appropriate error message when the password does not match the log-in email.	0.5	LJ
	23.3	System has an appropriate error message when email or password is inappropriate when trying to change account password.	0.5	LJ
	23.4	System has an appropriate error message when an inappropriate number is entered into the PPG test.	0.5	LJ
	23.5	System has an appropriate error message when the PPG sensor reading is inadequate (WIP).	0.5	LJ
Low				
	24	Medical Expert Information on Deep Vein Thrombosis	1	MJ BL
	25	Account Verification	1	LJ
	26	Account Recovery	1	LJ
	27	Transition Animations	1	LJ
	28	Individual Record Page UI	1	MJ
	29	As an authorized user, I would like to delete one of my results. (CANCELED)		

Total Man-Hours Spent: 81.5

Sprint #1

User Interface:

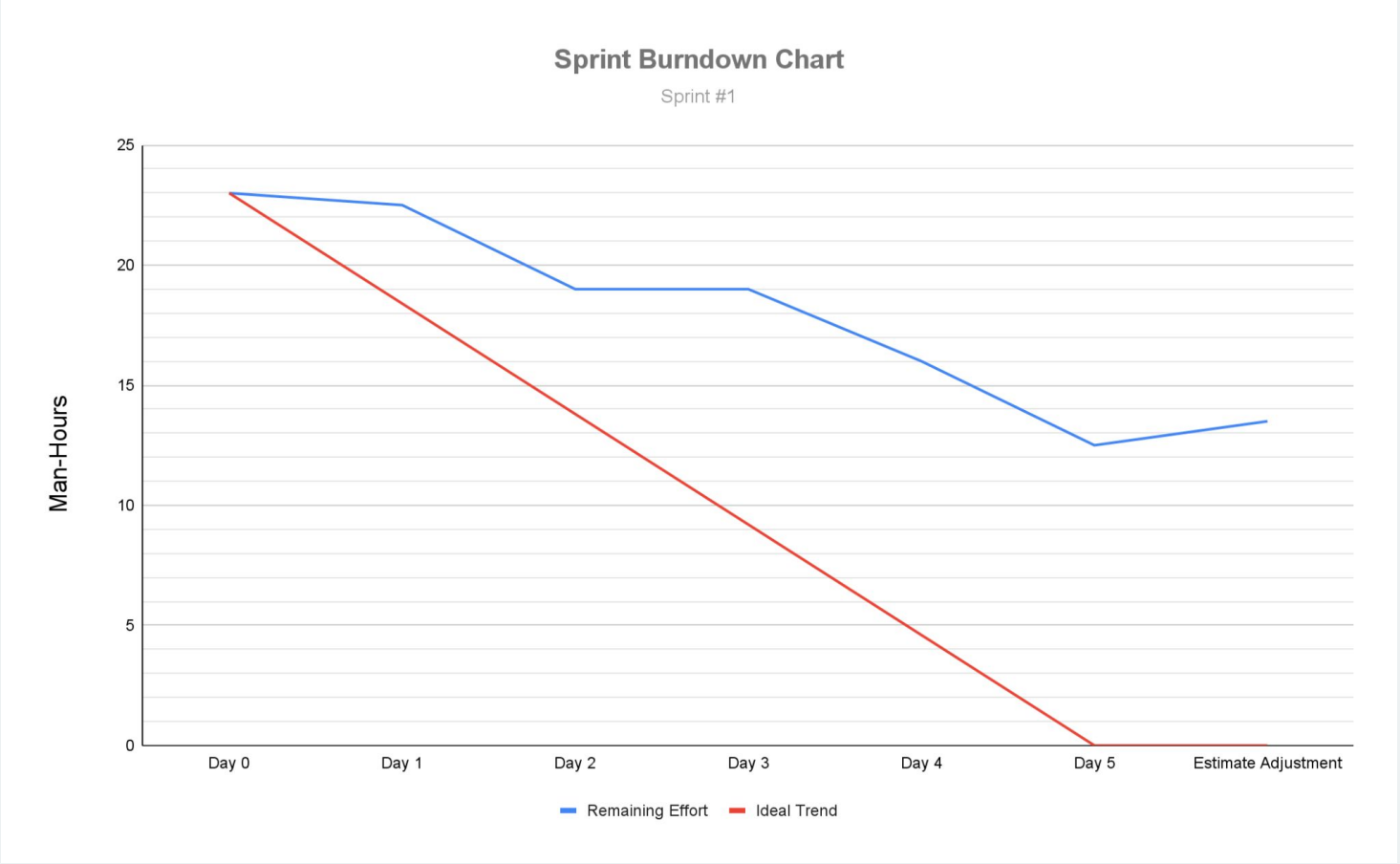
- UI design using Sketch
- UI development in Android Studio

Application:

- Create activities and basic navigation
- User SignUp and SignIn functionality

Database:

- Connection to Firebase
- Store user account

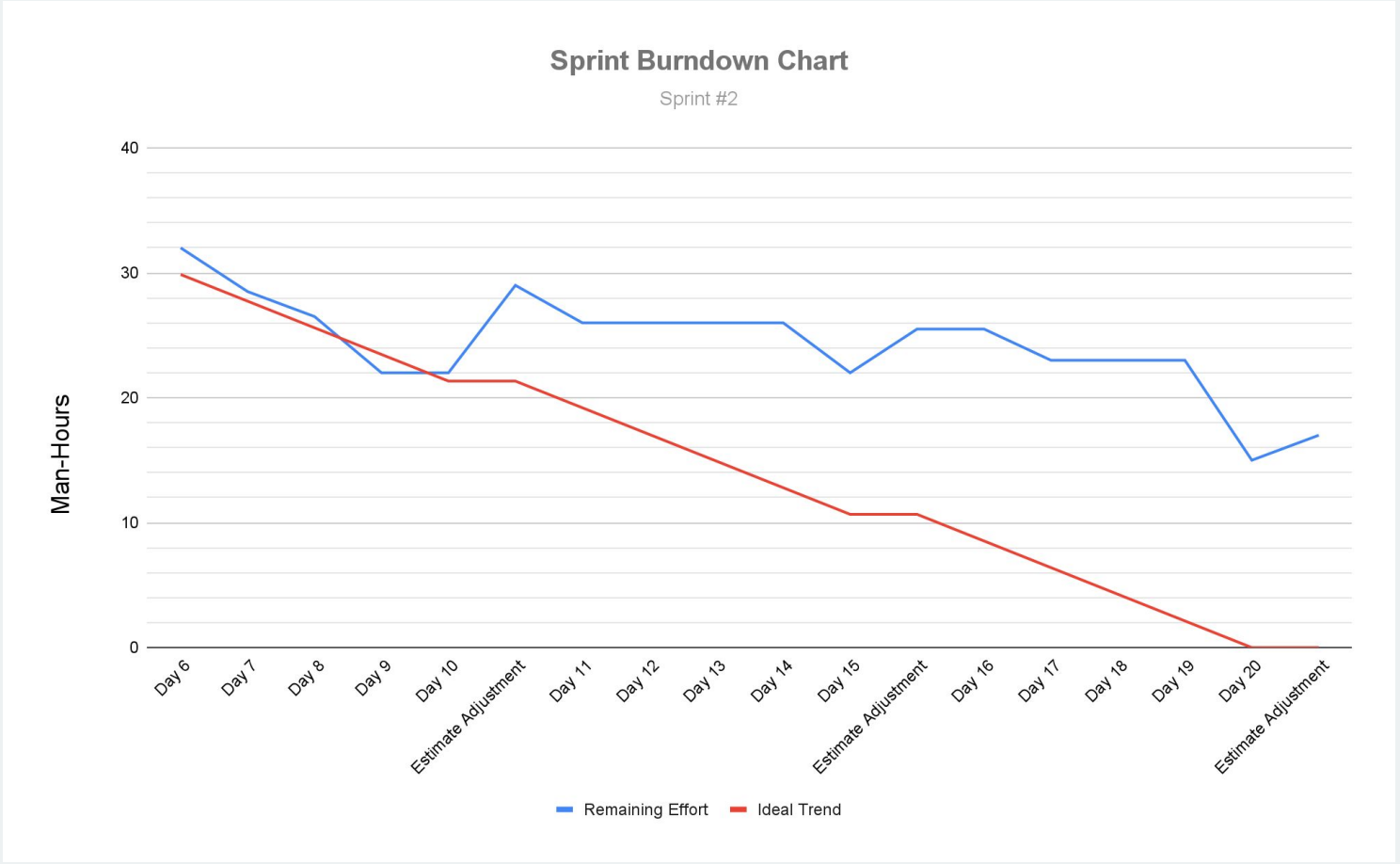


Sprint #2

User Interface:
- UI development in Android Studio

Application:
- Well's Screening and PPG Test
- Accurate risk values returned from test inputs
- Display all previous records
- Medical Information Section

Database:
- Store User's test results and information
- Retrieve Records

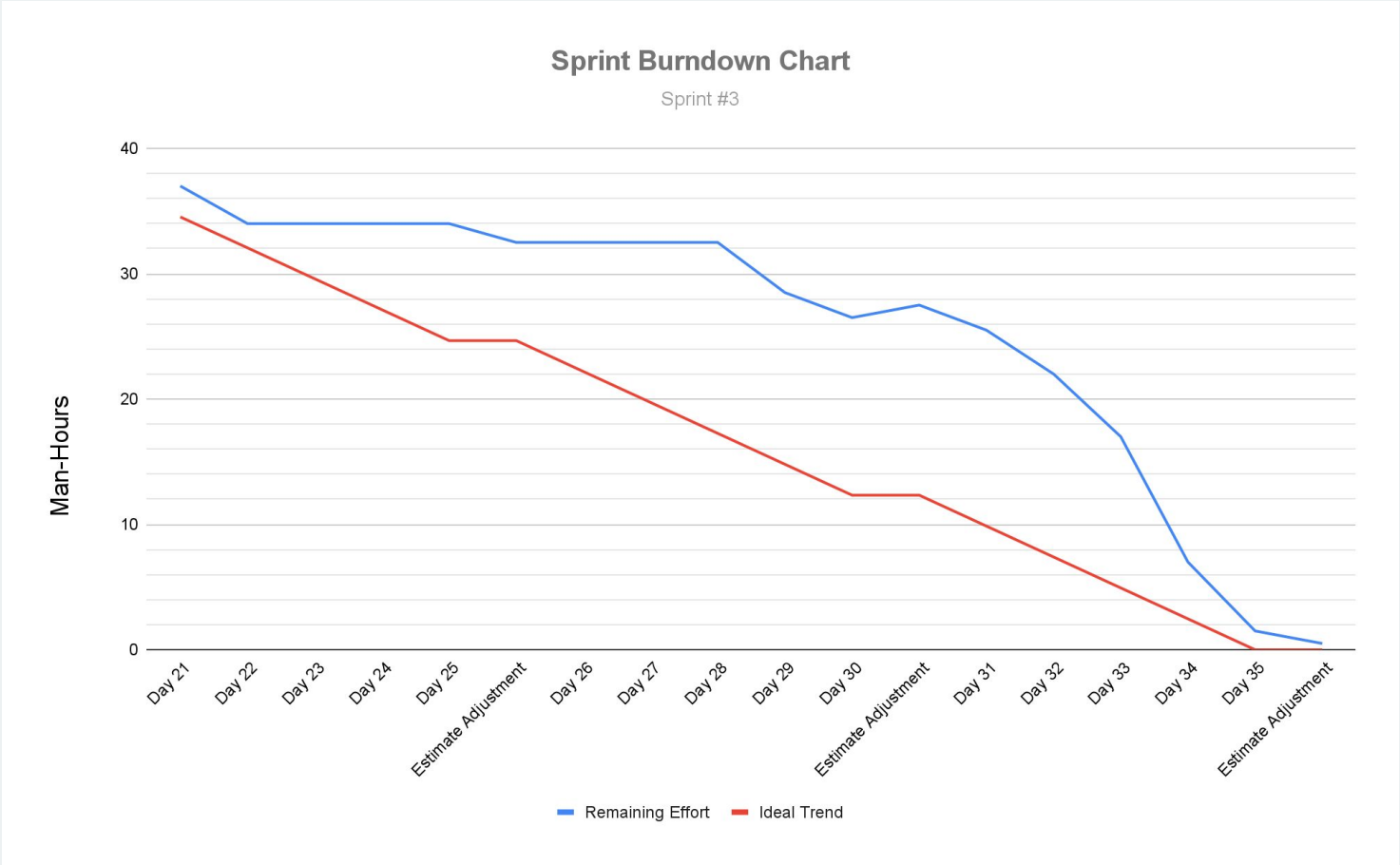


Sprint #3

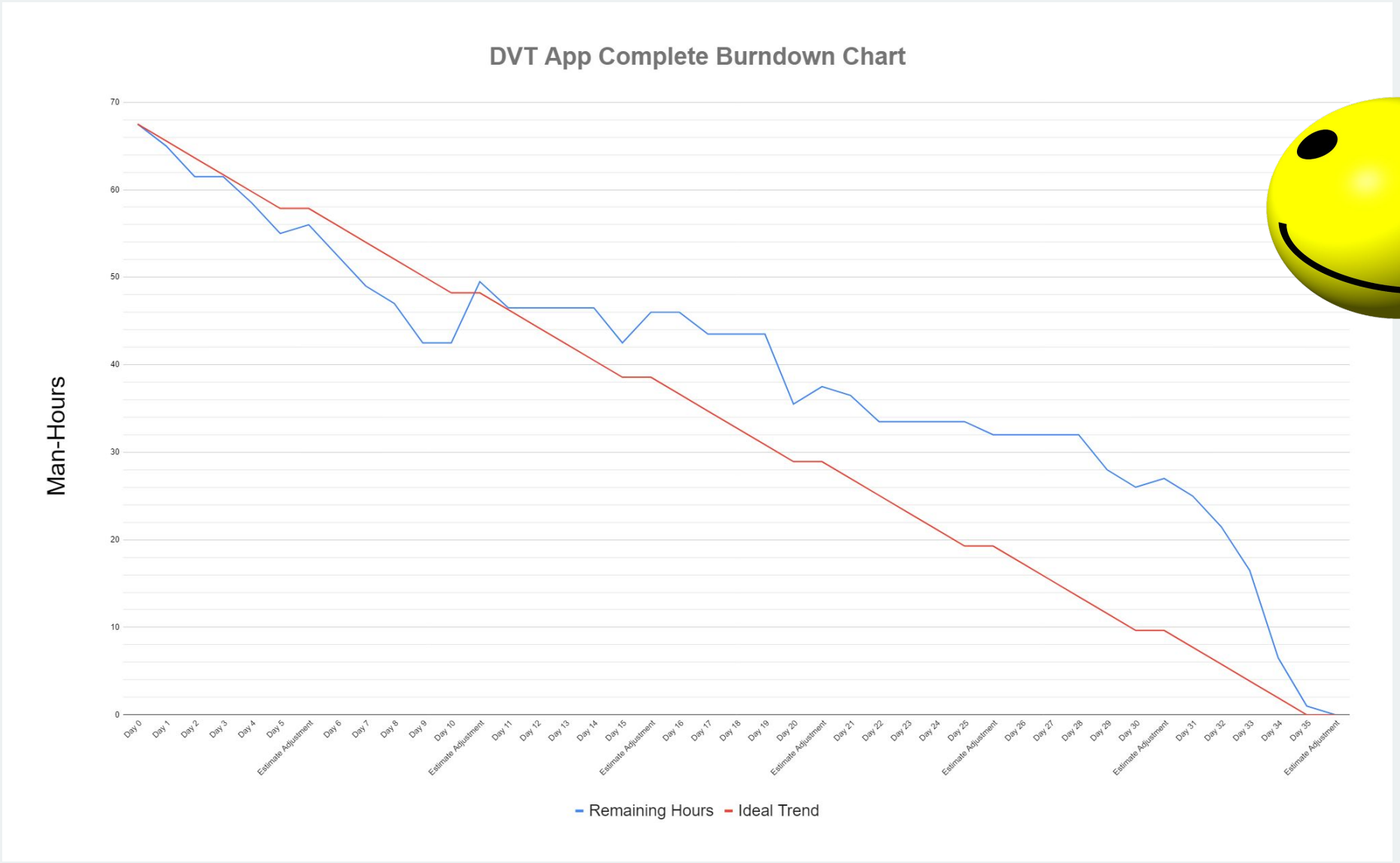
User Interface:
- UI fix
- Added pages UI

Application:
- Fixing results list
- Result more info page
- Error handling
- Account recovery
- How to Read Results Section

Database:
- User info editing
- Account verification

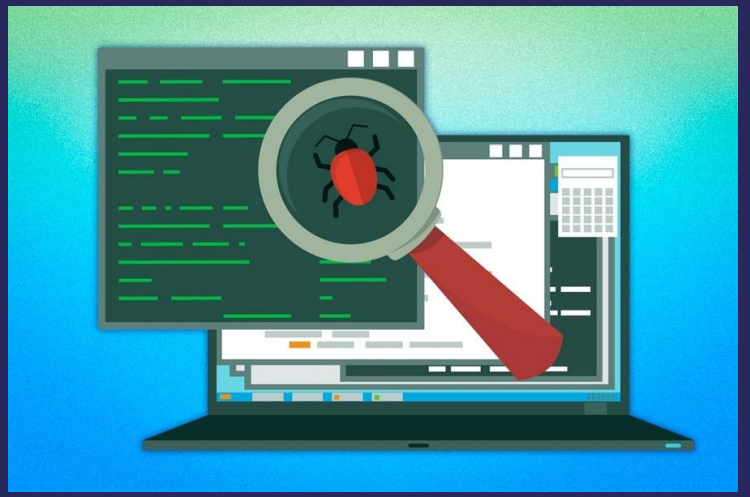


Complete Project Burndown



Testing

- Unit and System Testing
- Test Plan and Case Tracking
- Outcomes



Unit and System Testing

Unit Test Cases

- Well’s Screening
- PPG Test

System Test Cases

- Account creation
- Account sign in
- Accessing Test Records List
- Accessing Specific Test Records
- Account profile edits
- Account recovery
- App navigation

Test Plan and Case Tracking

Test Plan

- Each of the nine test cases were tested according to the test description and user inputs. The output was verified with the expected output.

Case Tracking

- Test logs for each case which record when the test was completed, the tester, description, and results.
- Any failed test includes an incident report.

Outcomes

Unit Tests

- Well's Screening ~ Passed
- PPG Test: Failed ~ Failed
  - The app allowed the user to input non-numerical values, causing the app to crash
  - This bug has been resolved

System Tests

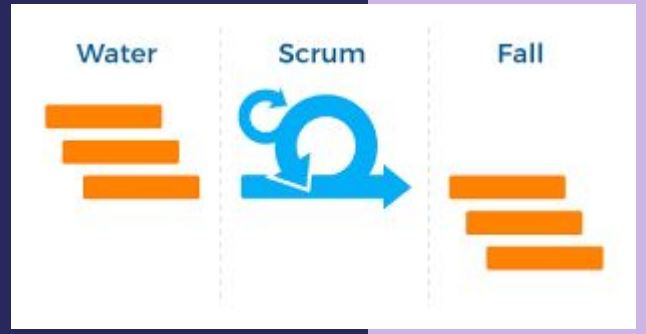
- Account creation ~ Passed
- Account sign in ~ Passed
- Accessing Test Records List ~ Passed
- Accessing Specific Test Records ~ Passed
- Account profile edits ~ Passed
- Account recovery ~ Passed
- App navigation ~ Passed

Software Engineering Concepts

Software Engineering Concepts

Software Engineering Life Cycles

- Water-scrum-fall model (part linear, part iterative) was used for this project.
- Three scrum development sprints



Configuration Management

- Android App Studio Used for Development
- SRS, SDD, Working Code, and SUSTD Utilized
- Workspace Management and Product Software Library Handled through Github

Software Engineering Concepts (continued)

Processes

- Iterative Agile Scrum Model
- Burndown Charts and Sprints

Models

- Prototyping
- Agile Scrum Cycle Method

Software Engineering Concepts (continued)

Software Quality

- Cost: Estimated 68.5 Man-Hours
  - Actual: 81.5 Man-Hours
- Schedule: 7 Weeks (3 Sprints)
- Quality: Fully functional, all reported bugs have been fixed

Software Testing

- Types of testing
  - Functional Testing
  - Non-functional Testing
  - Acceptance or Qualification Testing
  - Regression Testing
  - Recovery Testing
  - Interface Testing
- Techniques
  - Control Flow-Based Criteria
  - Fault-Based Techniques
  - Operational Profile

Lessons Learned

- How much planning is required prior to any development of a software project.
- Always plan for more time than expected.
    - Doubled all initial Man-Hour Estimates, Still 13 Man-Hours more than estimated
- Communication is key, especially as groups get larger and work is divided.

Product Software Library

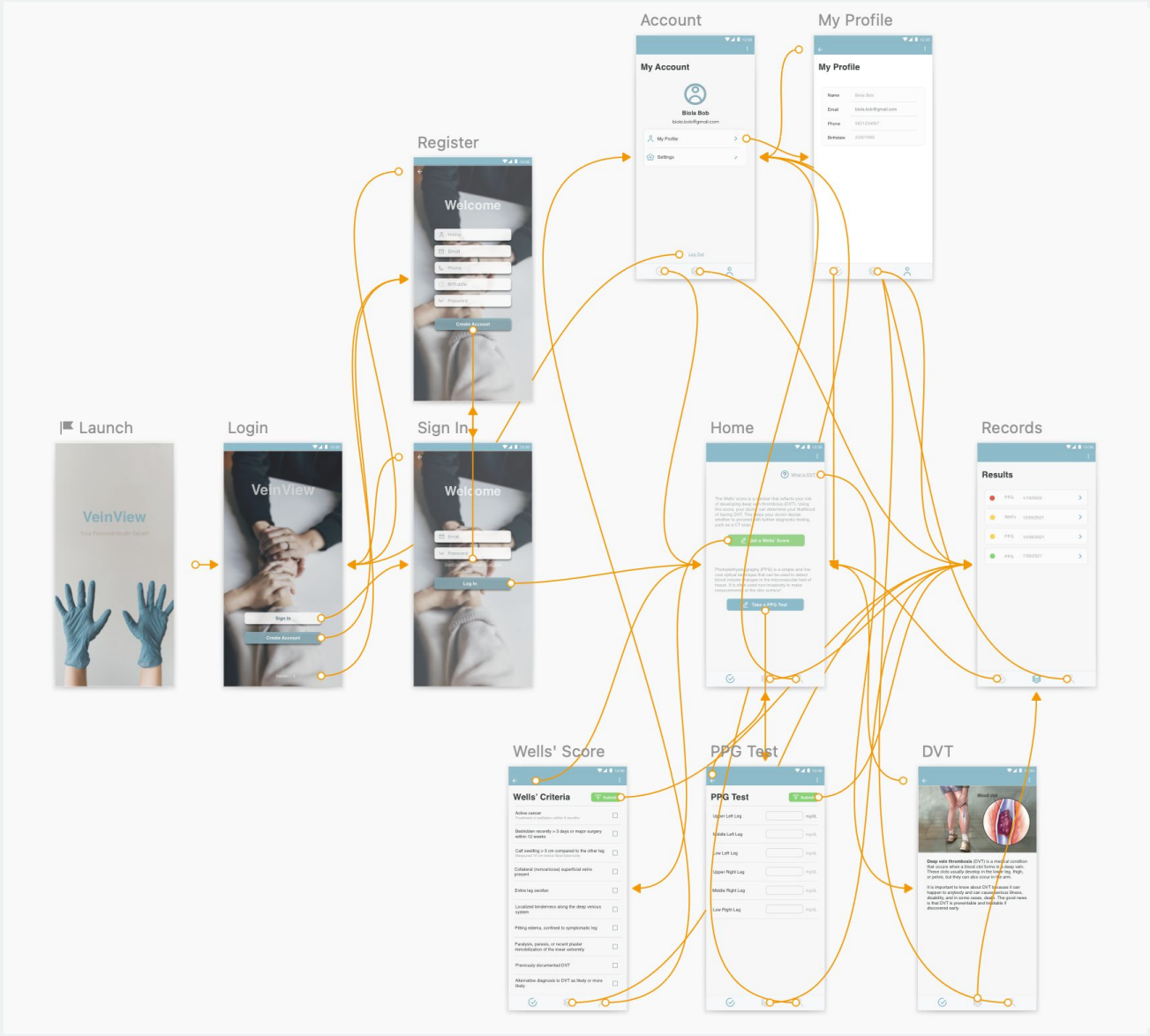
Version 1.0

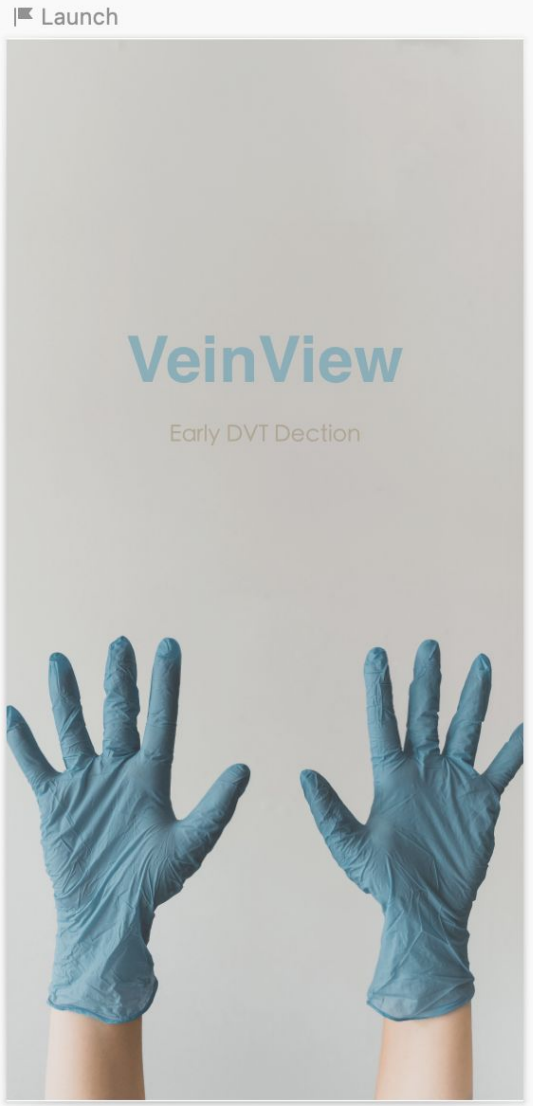
Github: public repository
https://github.com/lukemjohnston/VeinView

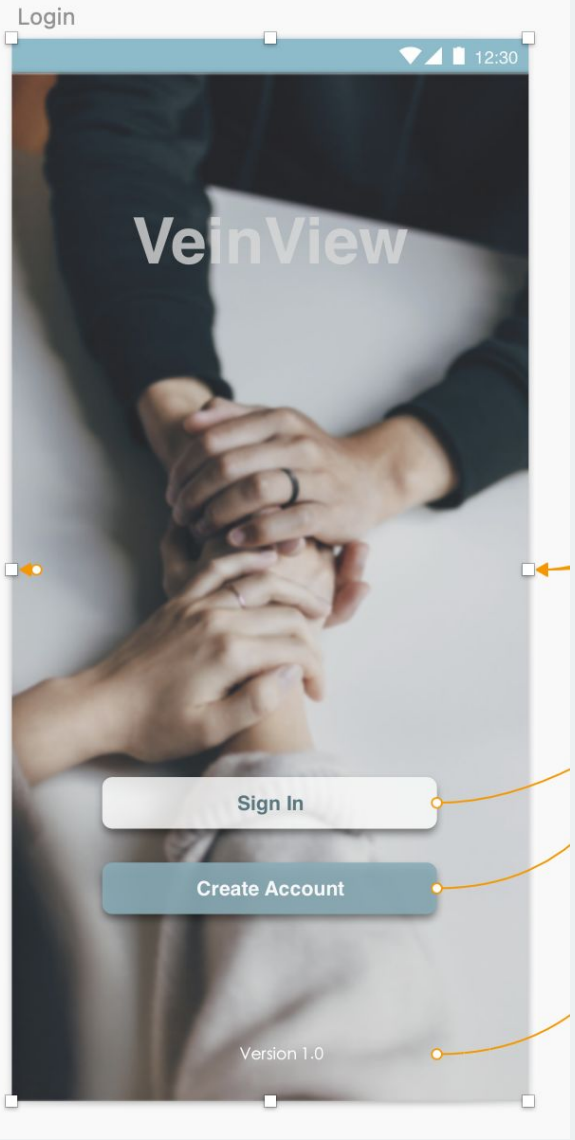
Holds android studio files and all documentation

Primary Product Features

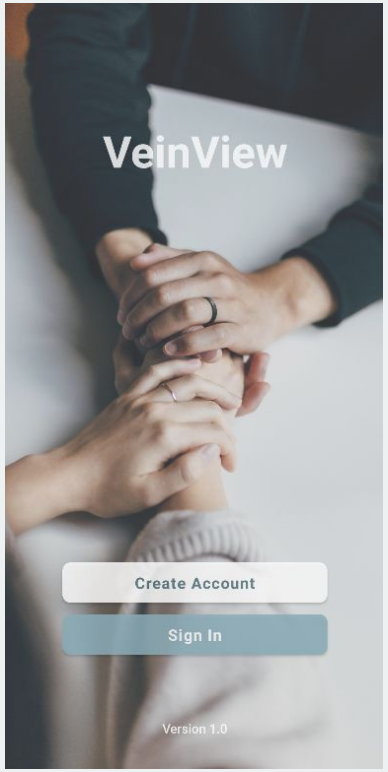
Initial UI Design

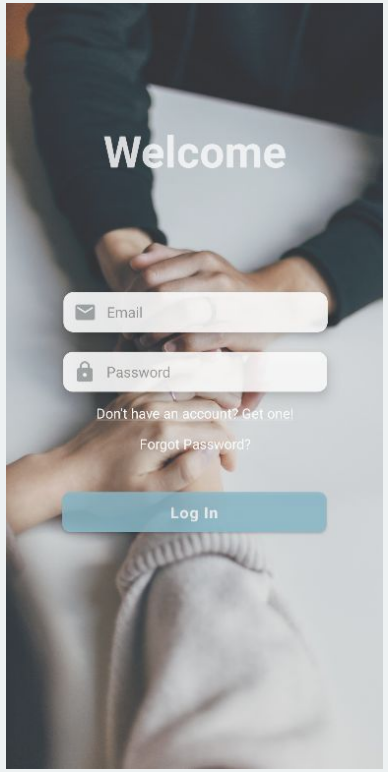


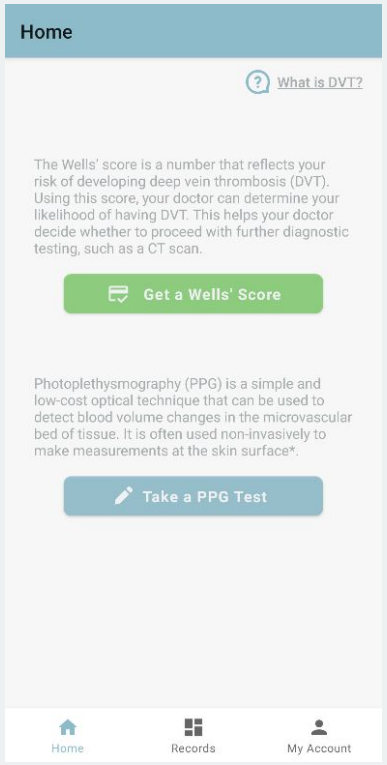


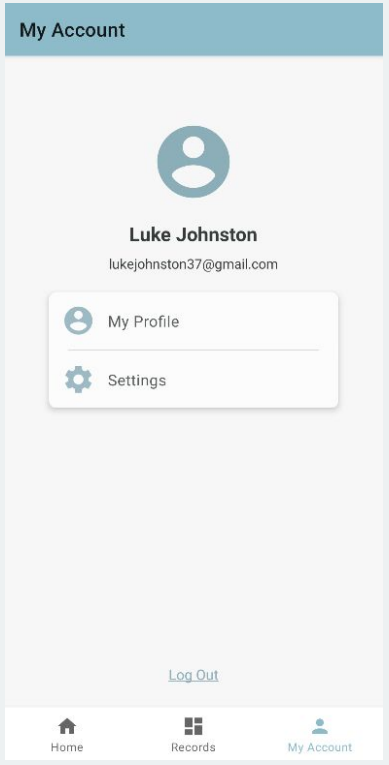


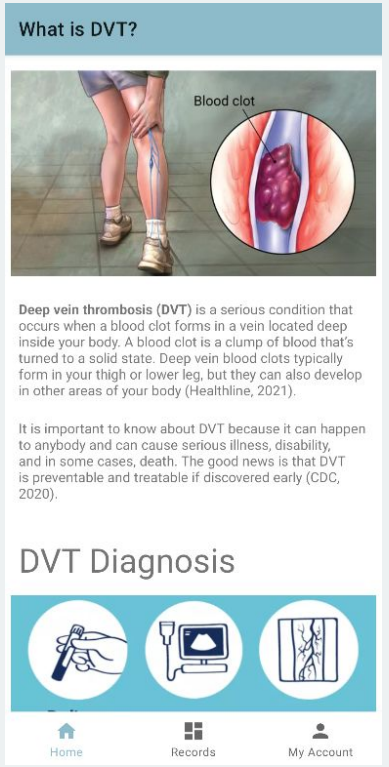
Application Screenshots











Demonstration