



Bachelor of Science (Hons) Computer Science

Sun Life Dental Mobile Application Upgrade

Michael Gerber (20093265)

01 January 2024

Contents

Introduction	4
Background	4
Target Audience	5
Use Cases	6
Lite Member.....	6
Full Member.....	6
Use Cases Requirements.....	6
Risk Assessment	6
Time Constraint Risks	6
Risk of Meeting Requirements.....	6
Project Scope and Objectives.....	7
Front-end Scope	7
Backend Scope	7
Non-Functional Objectives.....	8
Technology Stack.....	8
Flutter	8
React Native	8
Framework Comparison.....	9
Framework Evaluation	9
System Architecture	10
Activity Diagram.....	10
Class Diagram.....	11
Application Architecture	12
Implementation Diagram	13
Agile Scrum Methodology.....	14
Overview	14
Iteration/Sprint Breakdown	14
Scaled Agile Framework (SAFe).....	14
Project Timeline	15
Data Source Integration	16
Sun Life Member Dental API	16
Request Example.....	16
Request	16
Example Response	17
Description	18

User Interface (UI) Design	19
Testing and Quality Assurance	24
References.....	25
Plagiarism Declaration	25

Introduction

This final year project involves the redevelopment of an existing mobile application in collaboration with Sun Life. The primary focus is on enhancing Sun Life's current mobile application, aiming to deliver a robust and user-friendly experience. The development process will leverage either the React Native or Flutter framework. Part of the project includes critically analysing the two frameworks and choosing the framework best fit for the app. The report will include a plan on critical analysing the two frameworks. The goal is to align the app closely with the needs and expectations of the Sun Life userbase. This report provides an overview of the project's objectives and methodologies for the upgraded mobile application.

Background

Sun Life has an existing app called [Benefits tools app](#) available for both [iOS](#) and [Android](#).

The Benefits tools app is used to manage users' Sun Life dental benefits. The app allows users to view their dental coverage, access their digital dental ID cards, find a dentist, and access information about their policies and claims.

Sun Life also have a web application that offers the same functionality as the mobile app, but with extra functionality on top of that as well. The backend for the web app is an upgrade of the backend for the current mobile app. This newer API is what will be used in the development on the new mobile application.

During a meeting with Sun Life, there was discussed an in-house solution they are developing for the "find a dentist" feature's backend. There is no certain completion date for this solution, but when developing the new mobile app, it must be easy to implement this upgrade in the future.

Target Audience

Because of the existence of an already existing app, we can assume the demographic data for the new upgraded application to be similar to the existing demographic data.

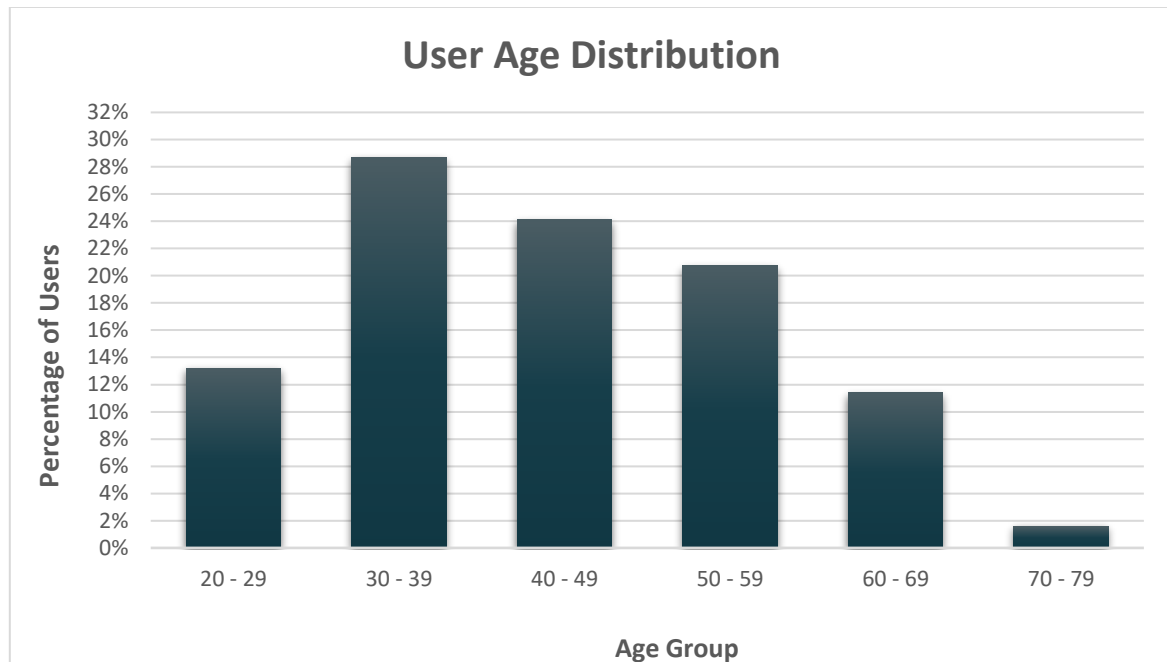


Figure 1 User Age Distribution for Benefits Tool App

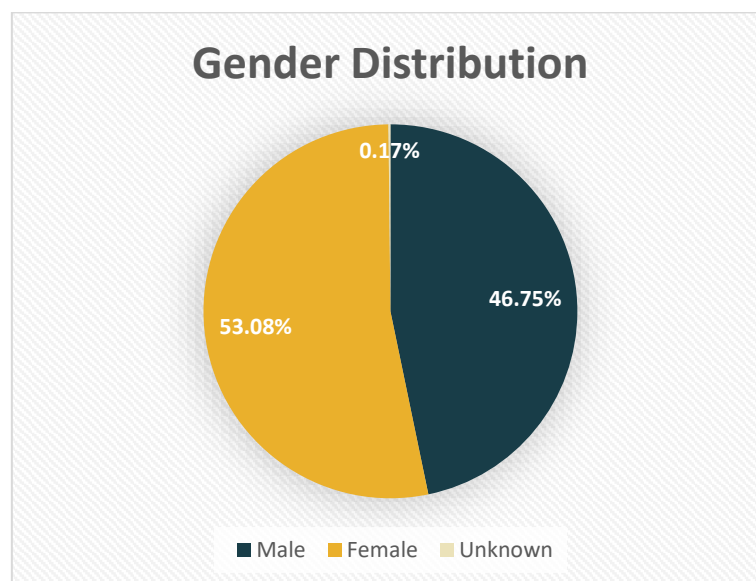


Figure 2 Gender Distribution for Benefits Tool App

The graphs above show the age and gender distribution for the current Benefits tool app. The majority (73%) of users are between the ages of 30 – 59 years of age. The gender distribution is mostly evenly split evenly between male and female, with a slight increase in female users.

Use Cases

The existing app has two types of users/members. These members have distinct levels of access to resources.

Lite Member

Lite members are only registered to a policy and is not associated with a social security number (SSN) or date of birth. This means they have limited access to resources. They only have access to claim related queries. They can only submit claims or submit an Evidence of Insurability for a claim.

Full Member

Full members are registered using a social security number (SSN) and a date of birth. They have access to all resources related to their user, including any enrolled insurance coverages and any claims they have made.

Use Cases Requirements

The minimum viable product requires the full membership's functionality to be implemented. Adding both user types would be considered as an advanced requirement for the app and the possibility of implementing it should be assessed during development. Each member type would require a dynamic landing page that can recognise the member type and display the options available to the specific member.

Risk Assessment

The following risk assessment goes over risks involved in the development of this application.

Time Constraint Risks

Time is an issue with all projects, and it can be unpredictable. There is a risk of not completing all the required features. The project's timeline is created in a way that the features to create a minimum viable product (MVP) is placed as the first things that needs to be completed. This ensures that if there is not enough time to complete the full project in the end, there would be an MVP complete. Other modules' work and assignments will be required throughout the semester and will take time out of development on this project. With proper time management these time constraints should not be an issue.

Risk of Meeting Requirements

The framework of choice needs to be decided before any development takes place. This framework needs to be learned during development and that might make developing new features using an unfamiliar framework difficult. Luckily, Sun Life have provided great learning resources such as Udemy and access to other developers that can give advice on these new frameworks the project will be using. The requirements and objectives of the project is structured in a way that if the timeline is followed a minimum viable product will be delivered by the end even if the last few sprints are not as successful as they were planned to be.

Project Scope and Objectives

Front-end Scope

The front-end scope of the application includes the user facing components of the mobile application. It should be an intuitive, and appealing user interface for Sun Life's customers.

1. **User Authentication:**
Implementation of a secure and user-friendly authentication system, allowing users to register an account, log in, and change usernames or passwords.
2. **Dashboard:**
The dashboard should be designed to dynamically display a dashboard for each of the types of members. This dashboard needs to be clear and easy to understand.
3. **Dental ID Card Display:**
The member's dental card, and dental information needs to be digitally displayed within the application.
4. **Dentist Search Interface:**
An intuitive search functionality that enables the users to find a dentist based on the user's current location, or specified location.
5. **Modernized User Interface:**
The original Sun Life application's user interface is getting outdated. Redesigning the user interface needs to provide a modern and visually appealing look for the user. The existing colour scheme of gold and navy blue will be kept.
6. **Permission Handling:**
User permissions needs to be handled gracefully, ensuring that the app responds appropriately to permission requests without affecting the user experience.
7. **User Account Management:**
The users need to have the ability to view and update their personal information.

Backend Scope

The backend scope of the project includes the server-side functionality, data processing, storage, and communications of the mobile application. It is important to note, Sun Life has existing APIs and there is no development needed in that regard. The following includes the backend scope:

1. **User Authentication and Authorization:**
There must be an implementation of a secure authentication system to verify a user's identity during the login process.
2. **Data Retrieval and Storage:**
A robust system for accessing the data from the API and stored locally on the device needs to be developed.
3. **Integration with Sun Life Database:**
The current Sun Life Database needs to be accessed through the API provided by Sun Life to ensure real time access to accurate and up-to-date data.
4. **Dentist Card Management:**
This requires the implementation of features that handle the retrieval of the digital dental cards from the Sun Life API. The functionality required includes storing, updating, and securely serving the information.

5. **Dentist Search Functionality:**
Allow for an interchangeable search system. The system should allow the user to search for a dentist by providing a location.
6. **User Data Modification:**
The app must have the ability to update personal user data, securely and reliably.
7. **Digital Wallet Integration:**
Development of a service that would allow the digital dental card to be added to wallet applications for user convenience.
8. **Location Based Functionality:**
This is a more advanced feature of the dentist search functionality. The user must be able to search for dentist using the location provided by the device.
9. **Multi Factor Authentication:**
Multi factor authentication (MFA) is an advanced security feature that will give the users the ability to add MFA to their login process to keep their account secure.
10. **Scalability and Future Proofing:**
All backend features need to be developed in a way where it can be interchanged with newer, upgraded APIs. Sun Life is currently upgrading some backend APIs, such as the Dentist Search function, it is crucial to consider the implementation of these newer features in the future.

Non-Functional Objectives

1. **Security**
Login is managed by authentication tokens. These tokens need to be securely stored and managed by the application. Securely accessing the API and other features through the network is a big priority.
2. **Maintainability & Upgradability**
Keeping in mind the possibility of future API changes and updates. The API access system needs to be adapted to be upgraded in the future.
3. **Performance**
Keeping an app performant and responsive is crucial for a good user experience.

Technology Stack

Flutter

Flutter is a cross platform framework created by Google. Single codebases can be used to create natively compiled applications for mobile, web, and desktop. (Google LLC 2023)

Google showcases all Google apps being developed using Flutter. Another big business that uses Flutter is the Chinese giant Tencent. (Google LLC 2023)

React Native

React Native is a cross platform framework created by Meta. It can be used to create cross platform mobile applications. (Meta Platforms, 2023)

React Native is not only being showcased in Meta application, such as Facebook, Oculus, and Messenger, but also in Microsoft and Amazon products such as Microsoft Office, Outlook, Teams, Amazon Shopping, Alexa, and Kindle mobile apps to name a few.

Framework Comparison

Feature	Flutter	React Native
Programming Language	Dart by Google	JavaScript or TypeScript
Development Environment	Android Studio or IntelliJ IDEA	Visual Studio Code or IntelliJ IDEA
Cross Platform Capabilities	Android, iOS, and more.	Android, iOS, and more.
Performance	Indistinguishable from native performance.	Small unnoticeable performance hit from JS being an interpreted language.
Documentation	Well structured documentation on a wide range of topics. Extra bonus is the Android design documentation with Flutter code samples.	Simple documentation on the basics, but not very extensive.
UI - Creation	Flutter uses readymade widgets that is easy to customize.	React Native has built-in native design components that looks specific to the native OS.
Third Party Integration Support	Flutter has a smaller community than React, but still decently sized.	The community has grown to over 700 000 developers. These developers have created thousands of component libraries.

Table 1 Flutter vs React Native comparison (Lopez 2023)

Based on the table above it seems Flutter is convenient, and easy to get a prototype app working and looking good without a lot of work. Documentation is very robust and there seems to be official libraries for all features required by the application. Dart is a new language to learn but seems similar enough to other languages used in Android development, such as Kotlin or Java.

React Native has great support libraries from other developers. These libraries can do everything the application required and more. JavaScript can be improved by using TypeScript instead. Simple documentation can be useful to quickly put together a prototype app.

Framework Evaluation

In the project timeline there are 2 weeks delegated to creating proof of concepts using each framework. These proof of concepts apps will be a login screen that takes the user to a page where the user information is displayed.

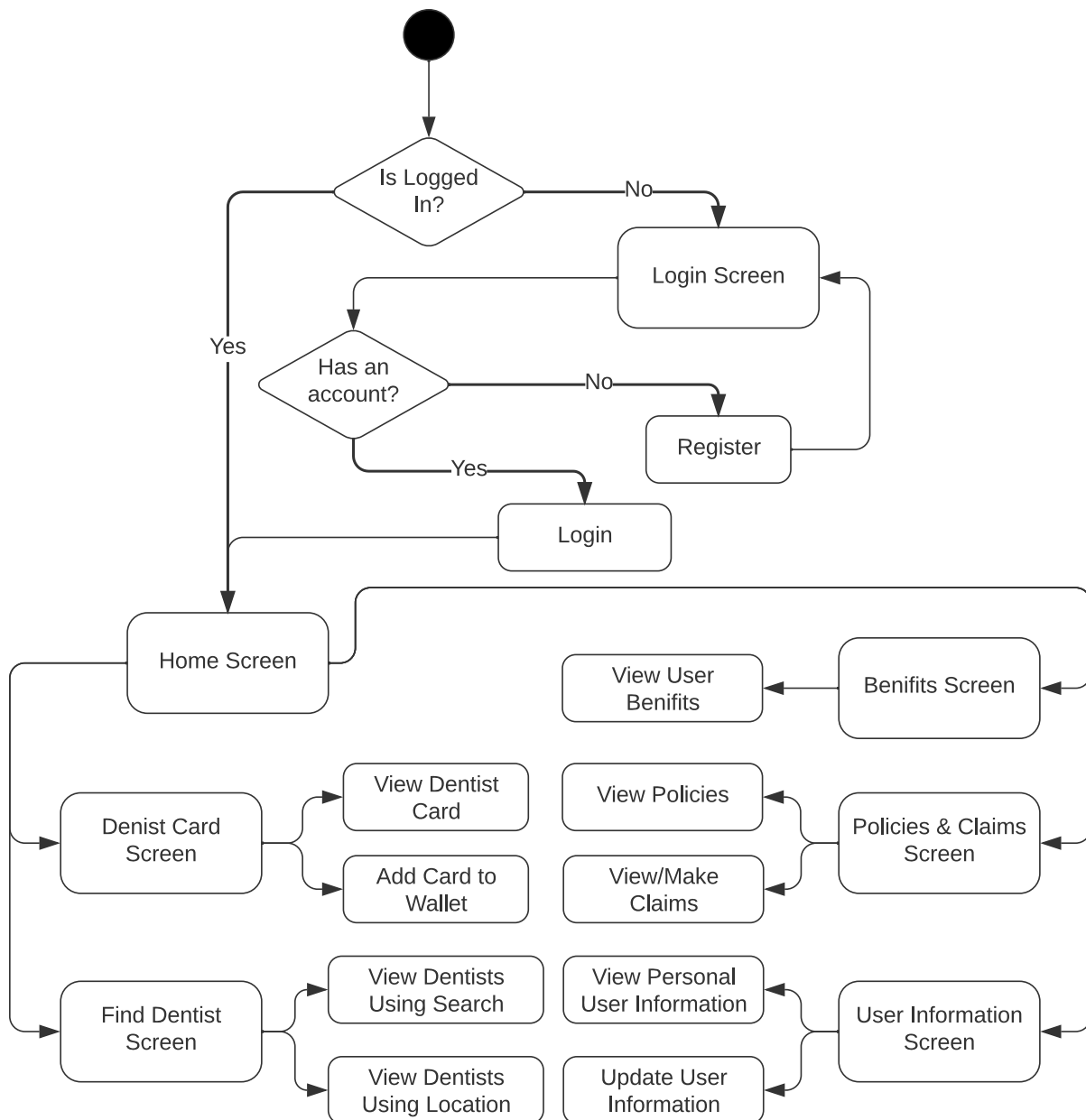
During the development the following 2 things need to be noted:

1. How easy it is to get these basic features working.
2. How easy is it to learn these frameworks using documentation and other resources.

Based on these results a framework can be chosen for the project.

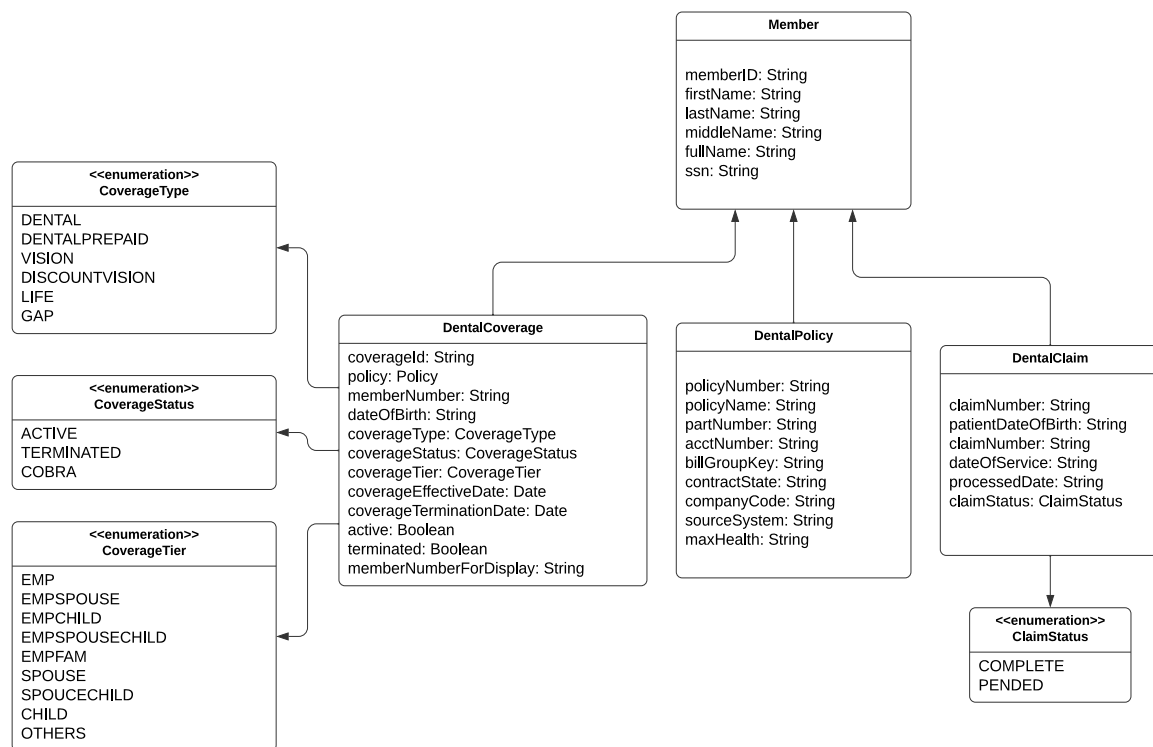
System Architecture

Activity Diagram



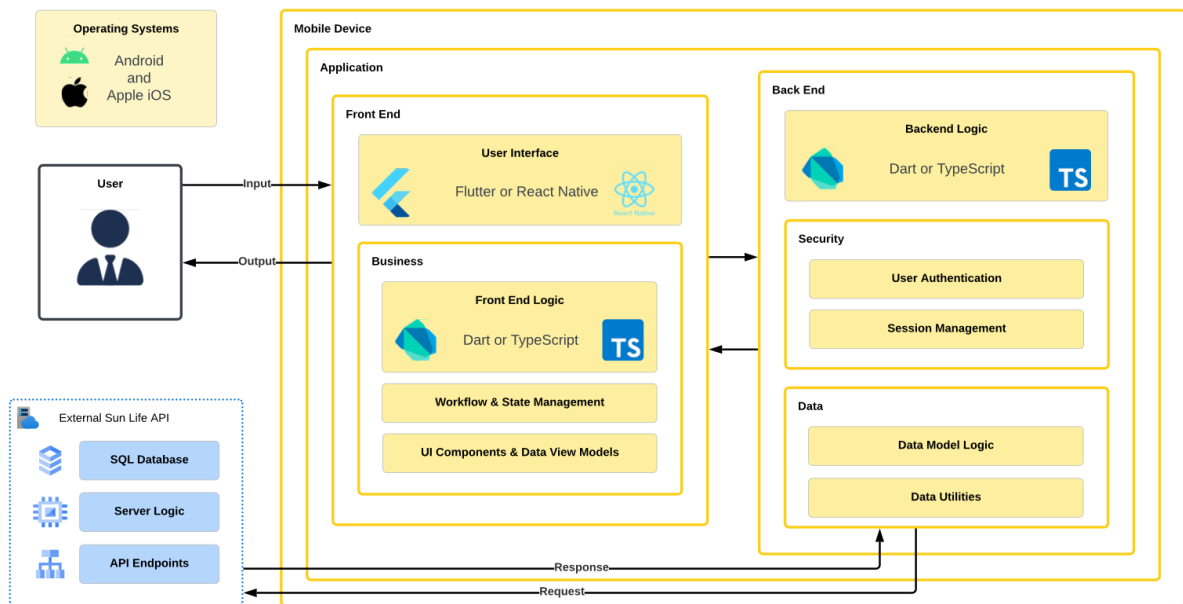
The following diagram shows the user flow of the mobile app. The flow starts with checking if a user's authentication is still valid and if they are still logged in. If they are not logged in, they can either register or log in. Once they are logged in, they can view their benefits and policies. A few other features like viewing their dentist card and finding a dentist is available to them. The home screen is where the user should be able to access all the app's features.

Class Diagram



The class diagram shows the entities and their attributes for the dental coverage system. There are 4 main entities, Member, DentalCoverage, DentalPolicy, and DentalClaim. The Member class holds the basic user information. DentalCoverage, DentalPolicy, and DentalClaims holds all the information that the mobile app should provide to the user. These class and property names were taken from API responses from the API provided by Sun Life.

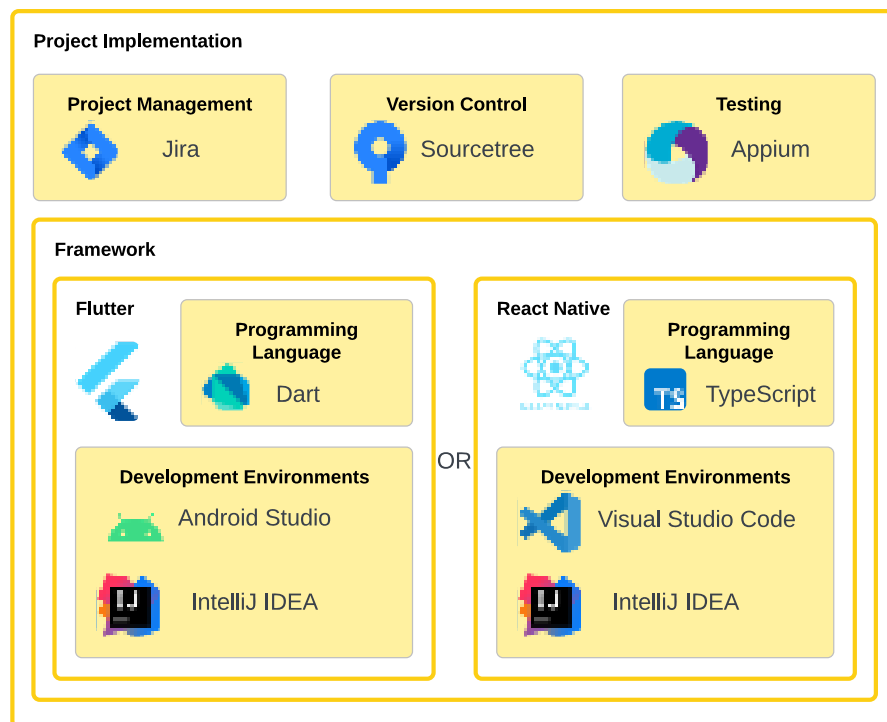
Application Architecture



Above is a diagram that shows the architecture of the mobile app and the external API integration. The application is split into 2 sections the front and backend. The diagram shows both the Flutter and React Native framework for the user interface. The back end is where the data, security and other logic is managed. The frontend communicates with the backend and processes and displays the data provided by the backend. The backend makes sure to stay up to date with the external API provided by Sun Life. The external API uses a SQL database and provides the data through endpoints that the app can access. In the end the user can provide inputs and together the frontend and backend can provide a valuable output to the user.

Note: the mobile operating systems supported are Android and Apple iOS.

Implementation Diagram



This implementation diagram shows the tools used for project implementation. The two frameworks are also compared. Flutter uses the Dart programming language while React Native uses TypeScript. They both share IntelliJ IDEA as a development environment, but Flutter can be used in Android Studio, and React Native can be programmed with Visual Studio Code.

Agile Scrum Methodology

Overview

Scrum is a software development methodology that uses small iterations of development to deliver a fully developed software product (Agile Scrum Methodology Explained n. d.). Sun Life uses Jira Software, a tool to manage agile projects, to keep track of the development of software projects. Planning and keeping track of this project's development iterations will be done through Jira.

Sun Life has provided a document on how they manage their development using Agile. The following headings is based on this document.

Iteration/Sprint Breakdown

Iterations are defined by Sun Life as a development cycle of 2 weeks. These iterations are also known as sprints. During this development cycle the developer needs to define the goal for the cycle, build the solution, integrate the solution into the system, and test and evaluate the solution. The overall goal for each iteration is to deliver a working software solution at the end of the iteration.

The following table is a breakdown of an iteration as provided by Sun Life.

Event	Approximate time	Value
Backlog Refinement	1 hour	The agile team gets to prepare requirements for the iteration planning step
Iteration Planning	2 to 4 hours	The team commits to a set of goals to be delivered in the iteration
Daily Stand-Up (DSU)	15 minutes	The team members have a daily discussion regarding the progression of iteration goals
Iteration Review	1 hour	The team meets with management to review the deliverables and receive feedback
Iteration Retrospective	1 to 2 hours	Tea reviews and improved its process before the next iteration

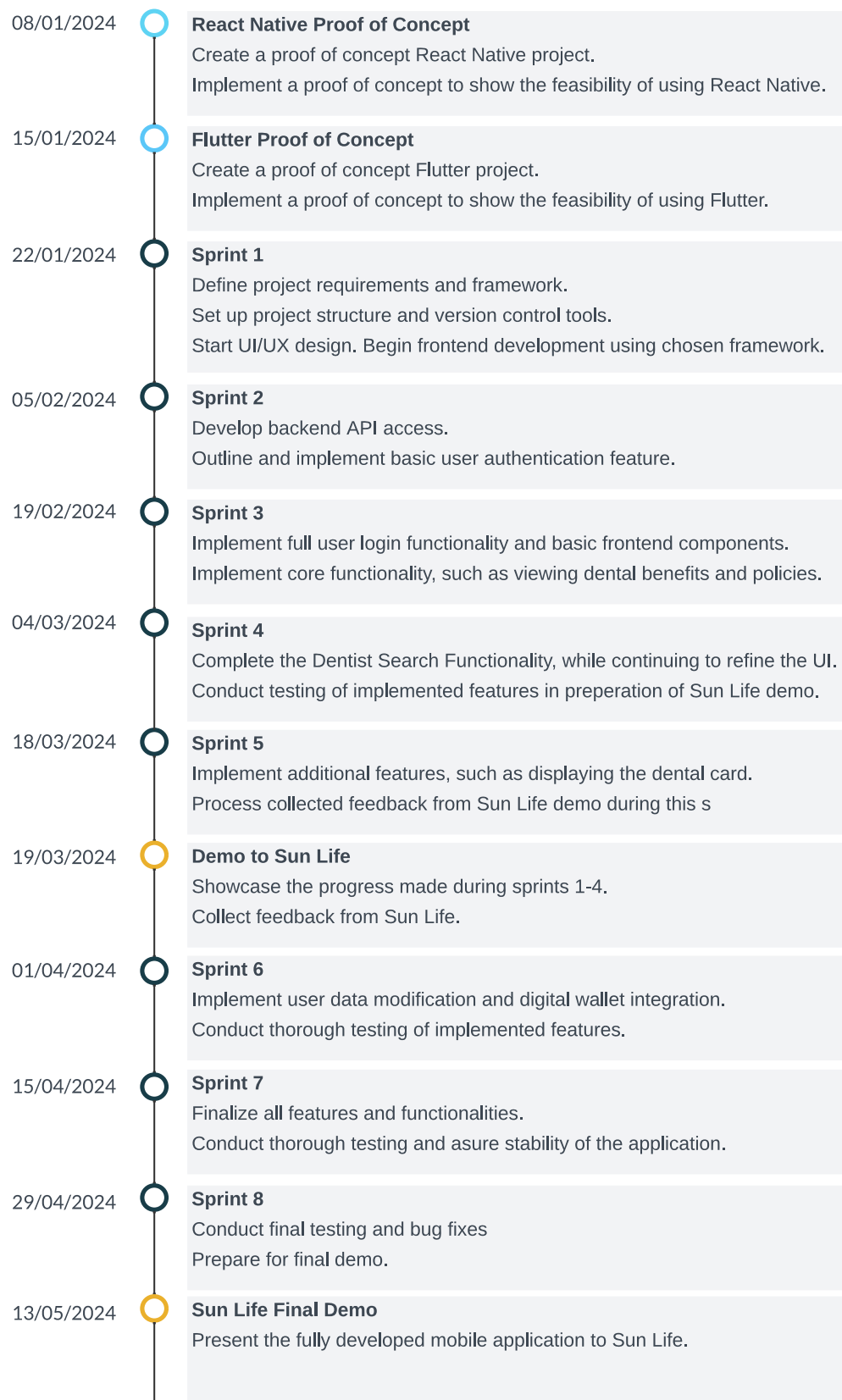
Table 2 Agile Iteration Events (McMahon, 2023)

Scaled Agile Framework (SAFe)

Sun Life uses an Agile Framework called Scaled Agile Framework or short SAFe. This framework defines a set of workflow patterns used to implement agile practices at the scale of large enterprises. These patterns allow a large number of agile teams to collaborate on a large scale of development (Piikkila, J n. d.). This is not very relevant to the development of this specific application, but it is important to note as development would be taking place in collaboration with Sun Life.

Project Timeline

The following project timeline is for the design and development of the application. Each sprint consists of 2 weeks. Adjustments can be made based on the actual progress during each sprint.



Data Source Integration

The data source for the project come in the form of a REST API provided by Sun Life.

Sun Life Member Dental API

Sun Life have provided online documentation on accessing the data using their API. This documentation shows the different endpoints available, and the responses expected for the requests. Responses are in the form of JSON objects.

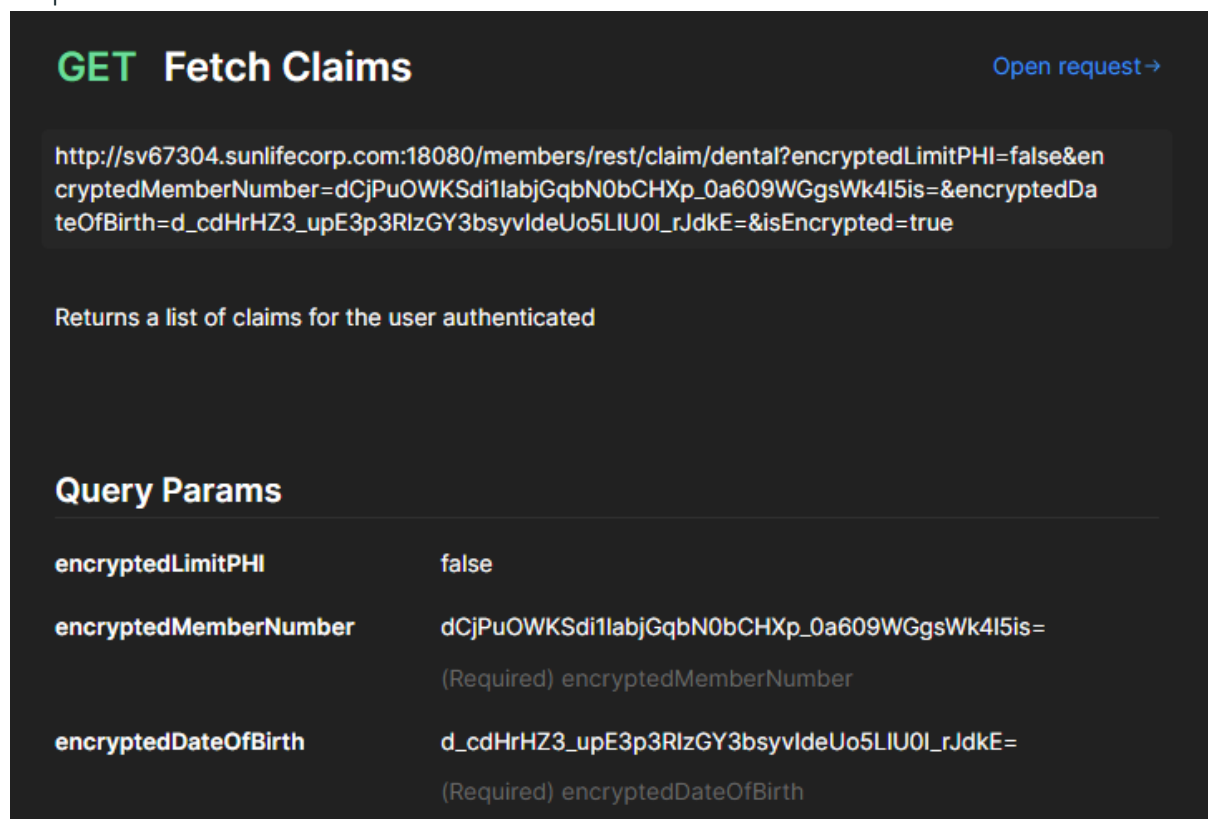
Authentication for the requests are handled by adding 2x parameters to the request. The parameters being an encrypted member number and an encrypted date of birth.

The digital dental ID card can also be requested through this API.

The following request example shows the format of the requests to the API and also shows the response of the request.

Request Example

Request



GET Fetch Claims [Open request →](#)

`http://sv67304.sunlifecorp.com:18080/members/rest/claim/dental?encryptedLimitPHI=false&encryptedMemberNumber=dCjPuOWKSdi1labjGqbN0bCHXp_0a609WGgsWk4I5is=&encryptedDateOfBirth=d_cdHrHZ3_upE3p3RlZGY3bsyvIdeUo5LIU0l_rJdKE=&isEncrypted=true`

Returns a list of claims for the user authenticated

Query Params

encryptedLimitPHI	false
encryptedMemberNumber	dCjPuOWKSdi1labjGqbN0bCHXp_0a609WGgsWk4I5is= (Required) encryptedMemberNumber
encryptedDateOfBirth	d_cdHrHZ3_upE3p3RlZGY3bsyvIdeUo5LIU0l_rJdKE= (Required) encryptedDateOfBirth

Figure 3 Fetch Claims for Member

Example Response

```
{
  "dentalClaimsDTO": {
    "memberId": null,
    "dateOfBirth": "p0rU4L9MHPcFLtAVPeJFIaRNxCKmsyDJwHW_Ukg4bT4=",
    "claims": [
      {
        "patientDateOfBirth":
"2rwnfUI40mbvbf2GE9JRur49aUDkgdFv1pcmeojxmYI=",
        "limitPHI": false,
        "claimNumber": "613402354",
        "dateOfService": "2016-03-07",
        "processedDate": "2016-05-19",
        "claimStatus": "COMPLETE",
        "charges": 575,
        "deductible": 11.75,
        "paidAmount": 429.3,
        "claimNumberSuffix": "00",
        "patientName": "John Smith",
        "relationship": "Self",
        "documentumLoaded": false,
        "values": [
          "613402354-00",
          "John Smith",
          "Self",
          "2016-03-07",
          "Complete",
          575,
          11.75,
          429.3,
          "2016-05-19"
        ],
        "claimNumberDisplay": "613402354-00",
        "claimNumberDocumentum": "61340235400",
        "headerOrder": "DENTAL",
        "headers": [
          "Claim number",
          "Patient",
          "Relationship",
          "Date of service",
          "Status",
          "Charges",
          "Deductible",
          "Paid amount",
          "Processed date"
        ]
      }
    ]
  }
}
```

```
}  
  }  
]
```

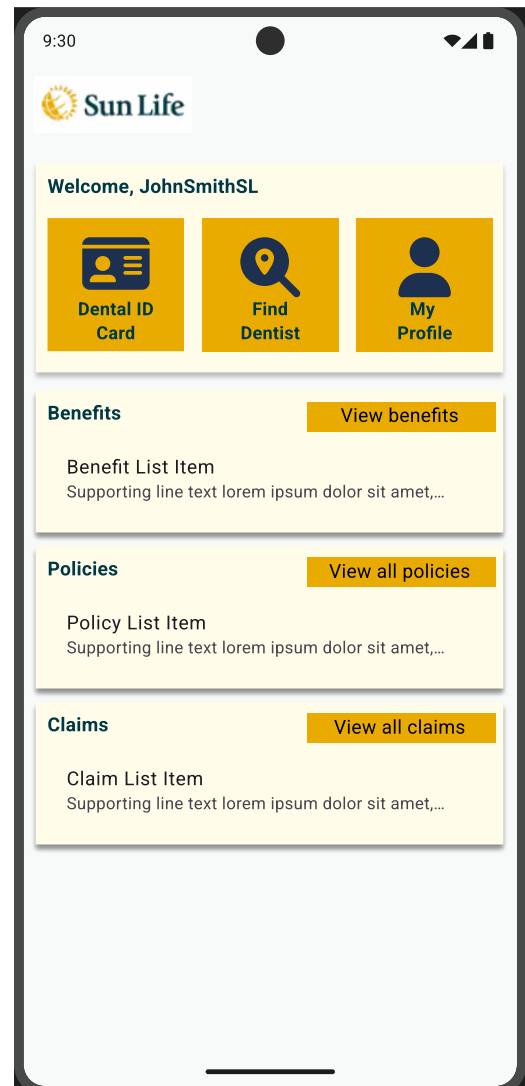
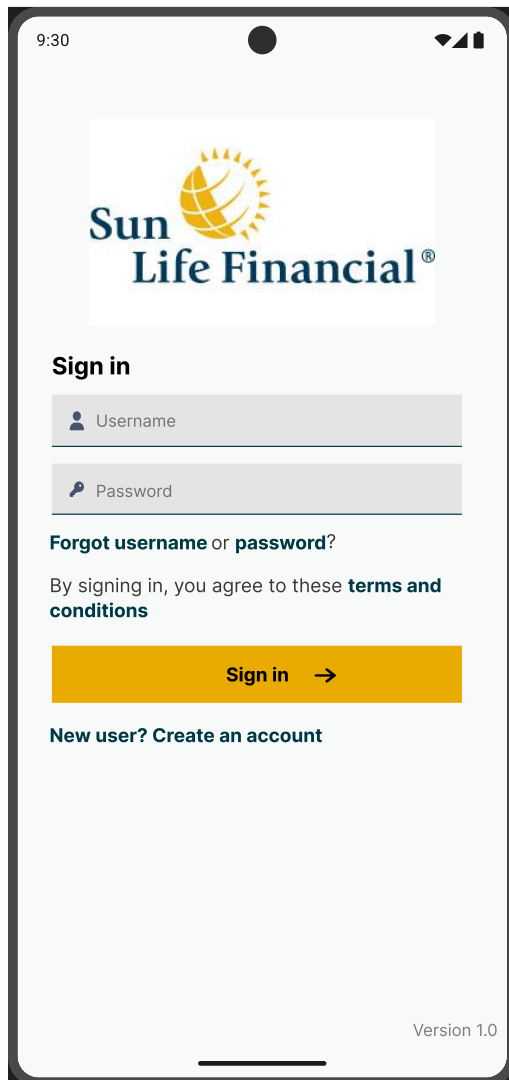
Description

This GET request is used to fetch claims for a member. It requires the 2 parameters, 'encryptedMemberNumber' and 'encryptedDateOfBirth'. These parameters act as authentication for the request.

Two things that stand out about this JSON object are the 'headers' and 'values' properties. This can be used to easily display the properties of these objects.

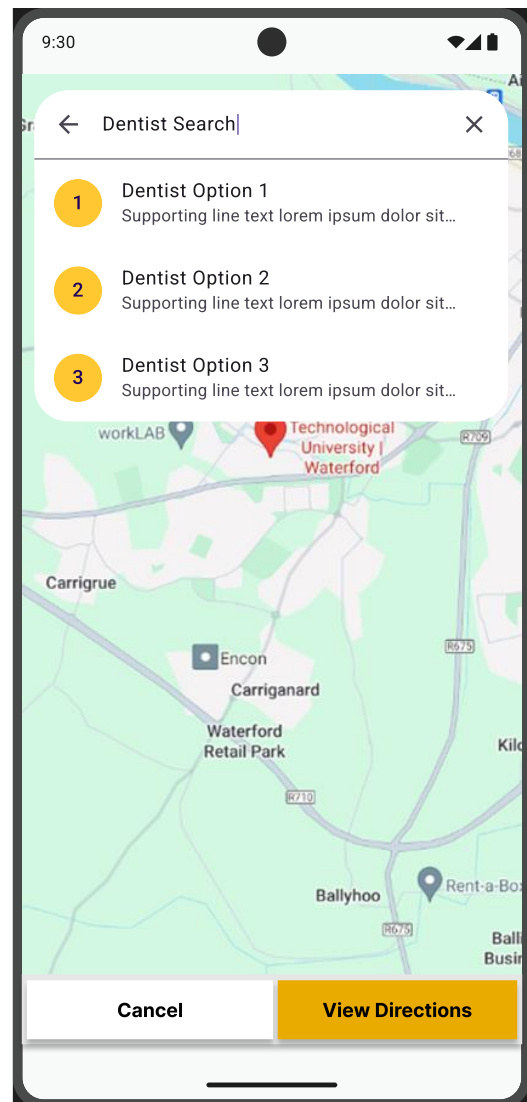
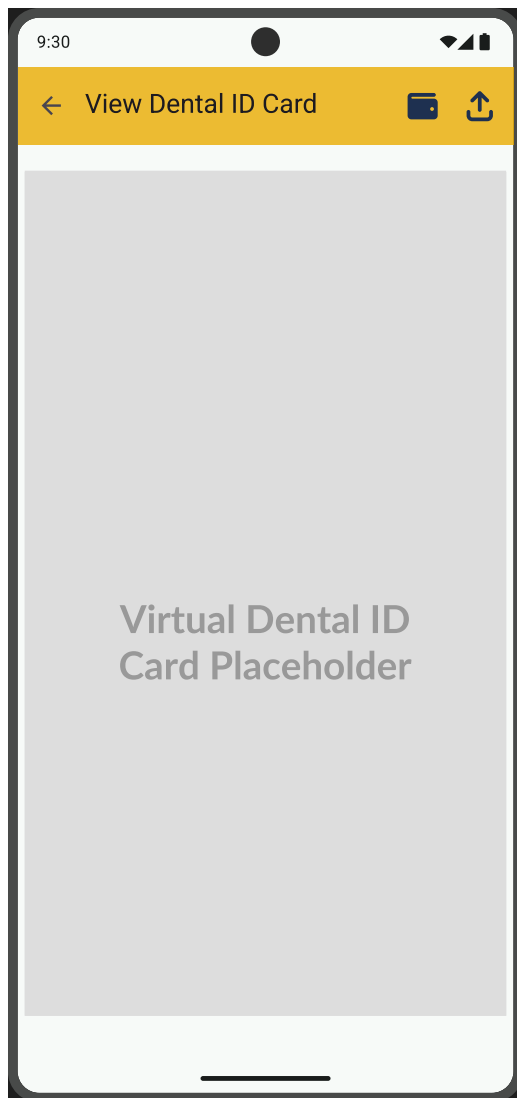
The rest of the requests provided by the Sun Life documentation will not be discussed in this report as they are all similarly structured to the one above.

User Interface (UI) Design



The login screen shows a username and password input. It's simple and mostly derived off the web application's login screen.

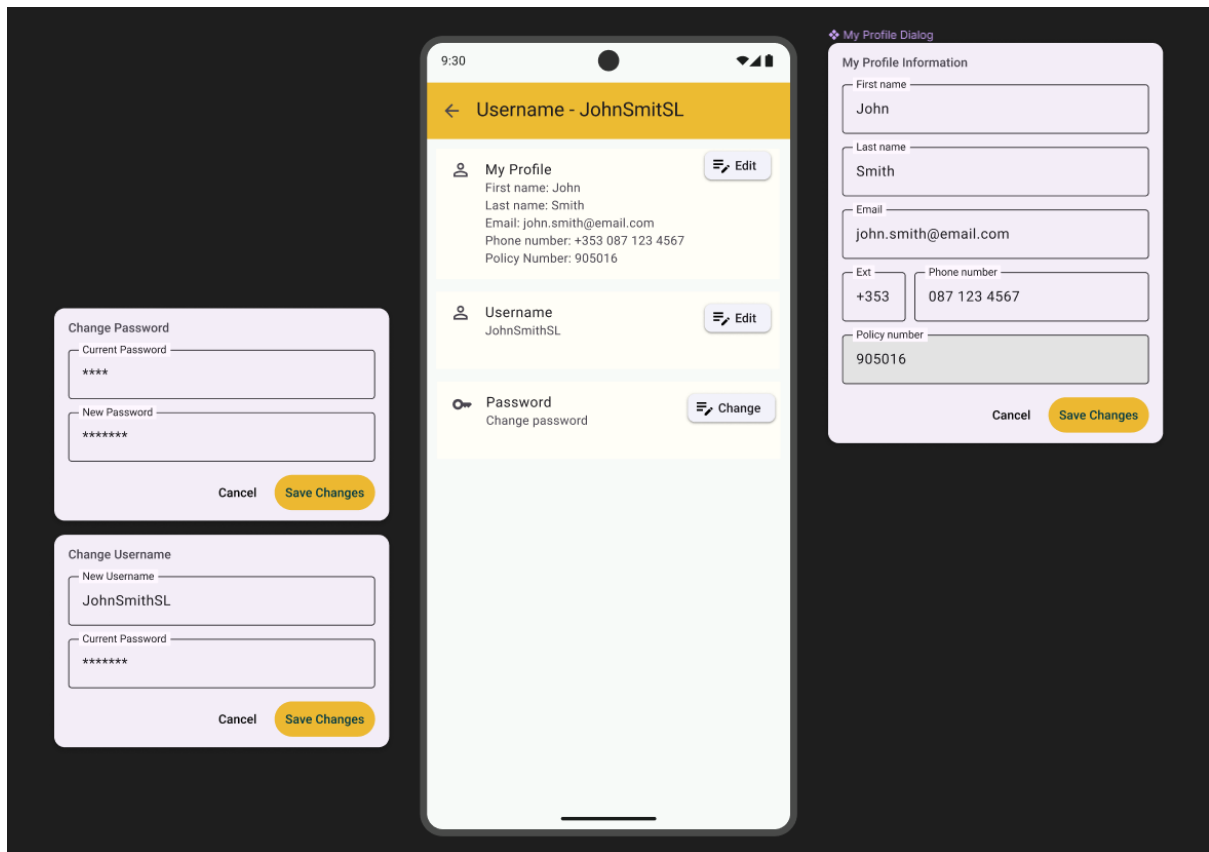
The next screen when logged in would be the home screen. This screen has a list of cards that displays the app's functionality ordered by importance. The first card has a set of quick actions that are features that the user can find quickly and are frequently used. If the user wants to do something more specific the rest of the cards would have that information for the user. The cards displayed on the home screen will be dynamic based on the user type.



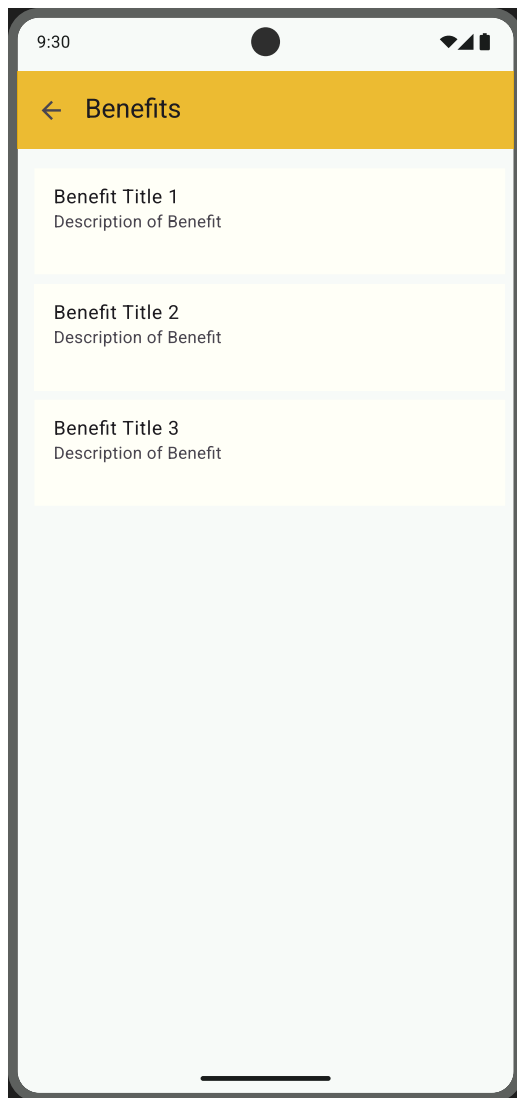
Here are the screens for the 'dental ID card' feature, and the 'find a dentist' feature.

The view dental ID Card screen displays the ID card in the form of an image in the full size of the screen. There are also 2 buttons on the top right, one to add/view the card in the wallet, and another to share the card.

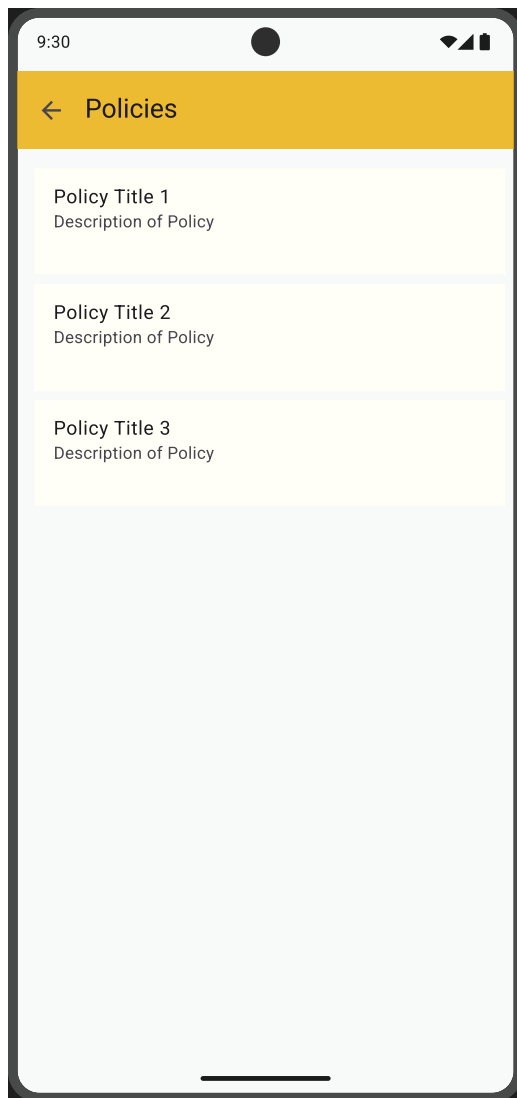
The other find a dentist screen has a map and a search bar. The search bar should search for a location and display the dentists in the location selected. When selecting a dentist, the 'view directions' can be used to open the directions on the phone's installed maps application.



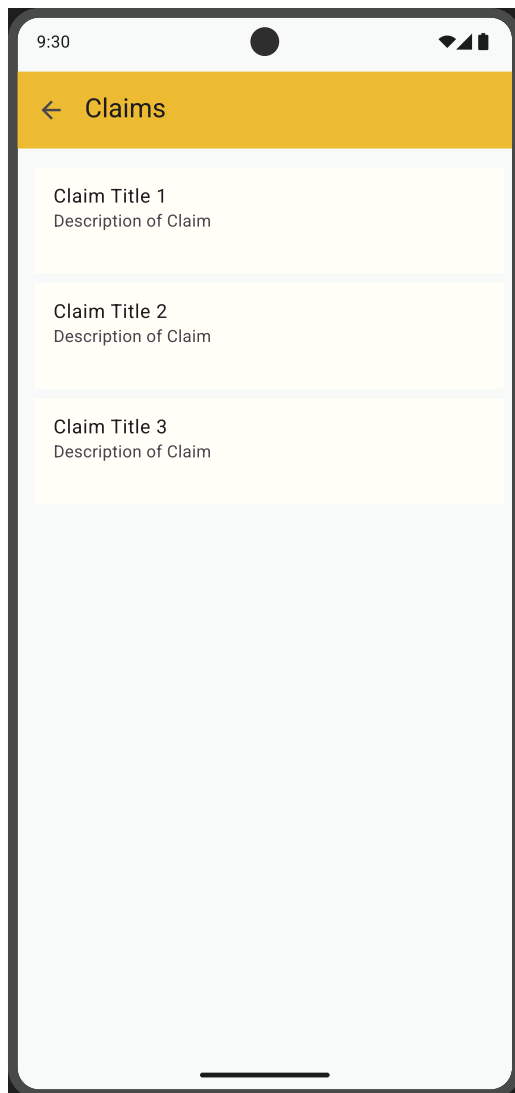
This screen is used to view or edit the member's personal information. Each edit button brings up a dialog with the personal information. The dialogs on the side are the dialogs that should be displayed.



The 'view all benefits' button on the homescreen should show the benefits screen. Clicking on a tile should take you to the benefit form screen which should show a form with all the details about the benefit.



Just like for the benefits, the 'view all policies' button on the homescreen should show the policies screen. Clicking on a tile should take you to the policy form screen which should show a form with all the details about the policy.



Just like for the benefits and policies, the 'view all claims' button on the homescreen should show the claims screen. Clicking on a tile should take you to the claim form screen which should show a form with all the details about the claim.

Testing and Quality Assurance

In addition to class tests done in the development's language, the current application tests are written in Java with a library called Appium. This library is used to automate interactions with an application. With every feature change the tests can be run to see if the changes had any implications on other features. Tests are a part of the end of each development sprint and is factored into the development timeline.

References

Agile Scrum Methodology Explained (n. d.) Inflectra Corporation, available:

<https://www.inflectra.com/Solutions/Methodologies/Scrum.aspx>

Piikkila, J (n. d.) *What is scaled agile framework? (SAFe)*, Atlassian, available:

<https://www.atlassian.com/agile/agile-at-scale/what-is-safe>

Google LLC (2023) *Flutter*, available: <https://flutter.dev/> [Accessed 01 Jan. 2024]

Meta Platforms, Inc. (2023) *React Native*, available: <https://reactnative.dev/> [Accessed 01 Jan. 2024]

Lopez M. (Jan 2023) *Flutter vs. React Native: A 2023 Developer Perspective*, Devlane, available:

<https://www.devlane.com/blog/flutter-vs-react-native-a-2023-developer-perspective>

McMahon, S. (2023) 'Summary of iterations and SAFe portfolios' [Presentation] [Accessed 31 December 2024]

Plagiarism Declaration

- I declare that all material in this report is entirely my own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams, or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in preparation of this submission.
- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name: Michael Gerber

Student Number: 20093265