

1. Medical Secretary	2
1.1 Architecture	4
1.1.1 Back-End Architecture	5
1.1.2 Front-End Architecture	6
1.1.3 GENIE Script Architecture	10
1.1.4 High-Level Architecture	12
1.1.5 Technology Research	13
1.2 Background	17
1.2.1 Client Provided Materials	18
1.2.2 Database Access Research	19
1.2.3 Presentation Script	20
1.2.4 React Native Firebase Research	21
1.3 Inception	28
1.3.1 Prioritised Risk Register	29
1.3.1.1 Product Risk Register	30
1.3.1.2 Project Risk Register	31
1.3.1.3 Technical Risk Register	32
1.3.2 Requirements	33
1.3.2.1 Acceptance Test	34
1.3.2.2 Goal Model	37
1.3.2.3 Paper Prototype (Concept Design)	38
1.3.2.4 Digital Prototype	41
1.3.2.5 Personas	42
1.3.2.6 Product Backlog	45
1.3.2.7 User Stories	49
1.4 API References	51
1.5 Deployment	52
1.5.1 Back-End Deployment	53
1.5.2 Front End Deployment	65
1.5.3 GENIE Script Deployment	78

Medical Secretary

Recent space activity

**Tong Niu**

Back-end Quality Assurance updated 09 Nov, 2018 • [view change](#)

**Wenzhuo Mi**

Back-End Deployment updated 07 Nov, 2018 • [view change](#)

**William Pan**

Criteria List updated 02 Nov, 2018 • [view change](#)

Test Cases updated 02 Nov, 2018 • [view change](#)

Unit Test Cases updated 02 Nov, 2018 • [view change](#)

Space contributors

- [Tong Niu](#) (520 days ago)
- [Chuan Yang](#) (526 days ago)
- [Alston Chu](#) (526 days ago)
- [Wenzhuo Mi](#) (526 days ago)
- [Yuanchenxi Liu](#) (526 days ago)
- ...

Client Contact Detail

Name	e-mail
Susan Pickering	suepickering81@hotmail.com

Supervisor Detail

Name	e-mail
Eduardo Araujo Oliveira	eduardo.oliveira@unimelb.edu.au

Team Members Detail

Name	e-mail	phone
Chuan Yang	chuany1@student.unimelb.edu.au	0414656498
Yuanchenxi Liu	yuanchenxil@student.unimelb.edu.au	0452150916
Wenzhuo Mi	miw@student.unimelb.edu.au	0412082301
Ruoyi Wang	ruoyiw@student.unimelb.edu.au	0434049545

	tniu1@student.unimelb.edu.au	0403166001
	yueyanc1@student.unimelb.edu.au	0421450282
	wpan1@student.unimelb.edu.au	0430475252
	acchu@student.unimelb.edu.au	0479016602
	jiacongl1@student.unimelb.edu.au	0422830562
	zhuny1@student.unimelb.edu.au	0450075335
	x.fan22@student.unimelb.edu.au	0426667289

Architecture

Architecture is the set of techniques and design patterns to develop more structured applications.

Back-End Architecture

Development Diagram (Component Diagram)



Editable Link: [Component Diagram](#)

Class Diagram



Editable Link: [Class Diagram](#)

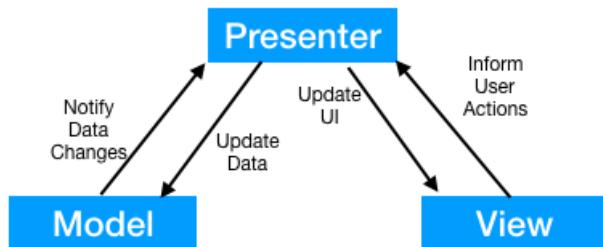
Sequence Diagram



Editable Link: [Sequence Diagram](#)

Front-End Architecture

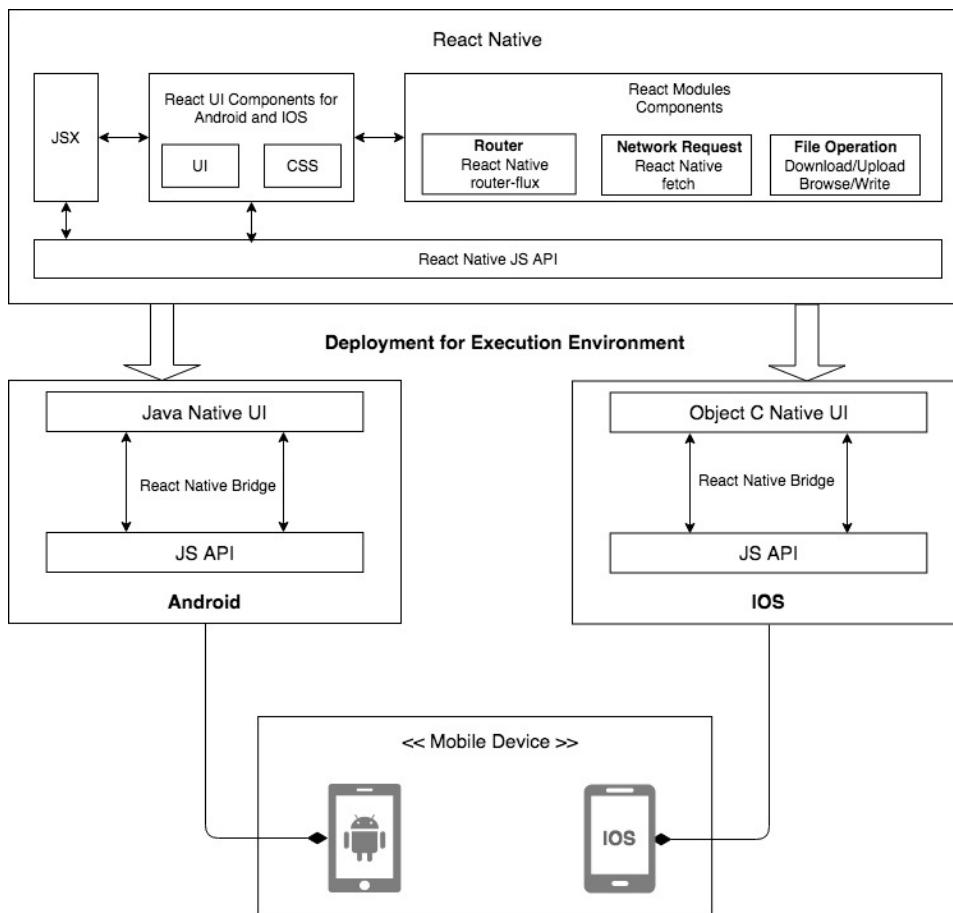
Model:



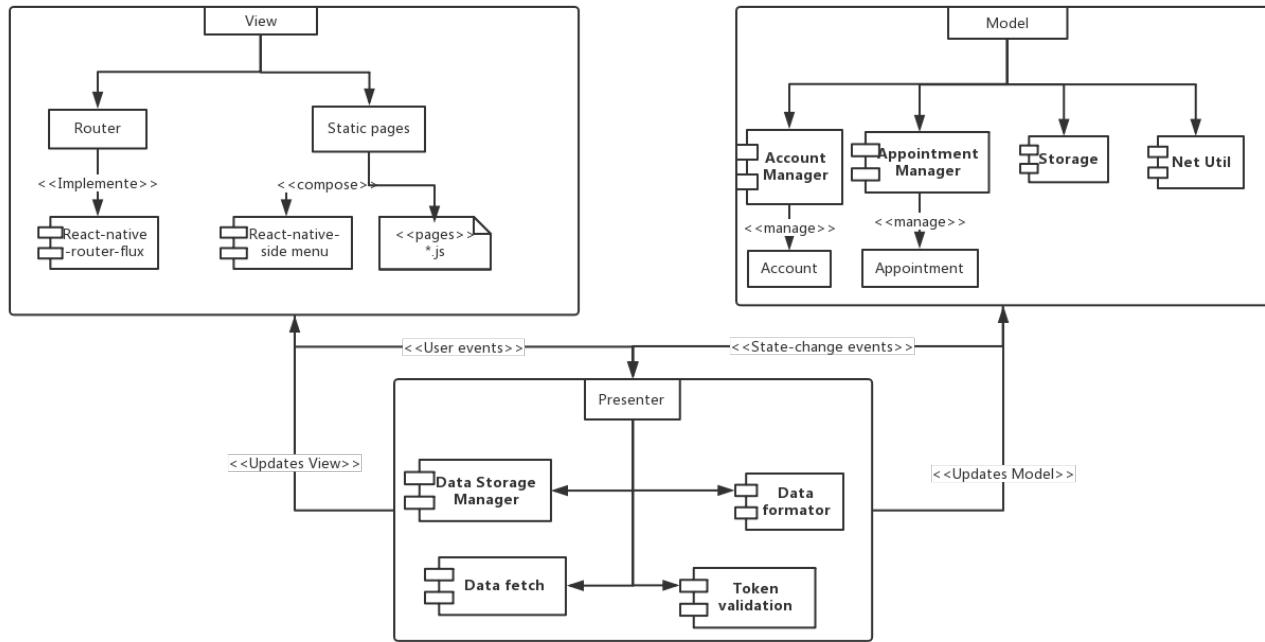
- The *model* is an interface defining the data to be displayed or otherwise acted upon in the user interface.
- The *view* is a passive interface that displays data (the model) and routes user commands (*events*) to the presenter to act upon that data.
- The *presenter* acts upon the model and the view. It retrieves data from repositories (the model), and formats it for display in the view.

Architecture 4+1 graph:

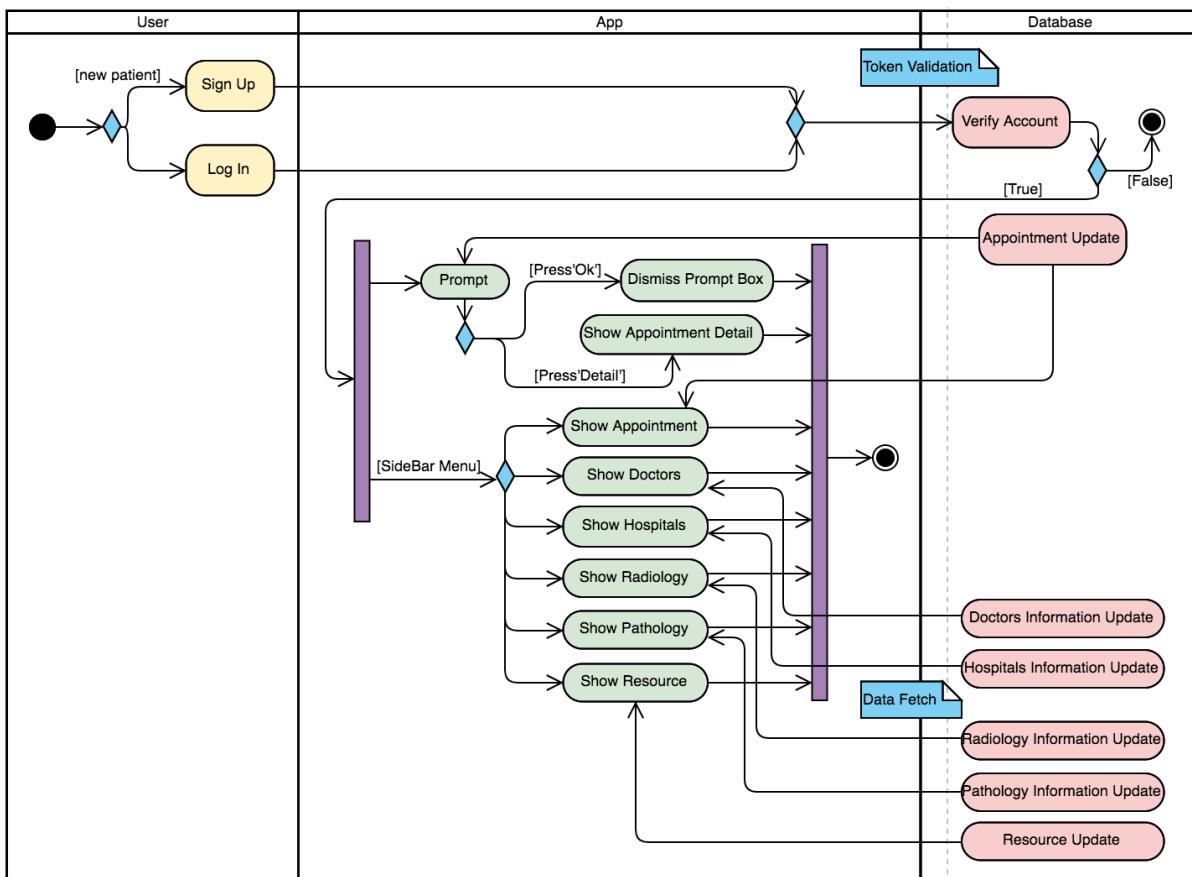
- Deployment View:



- Development View:



- **Process View:**
In UML, Activity diagrams - which can be used to model concurrent behaviors - are used to model the process view.

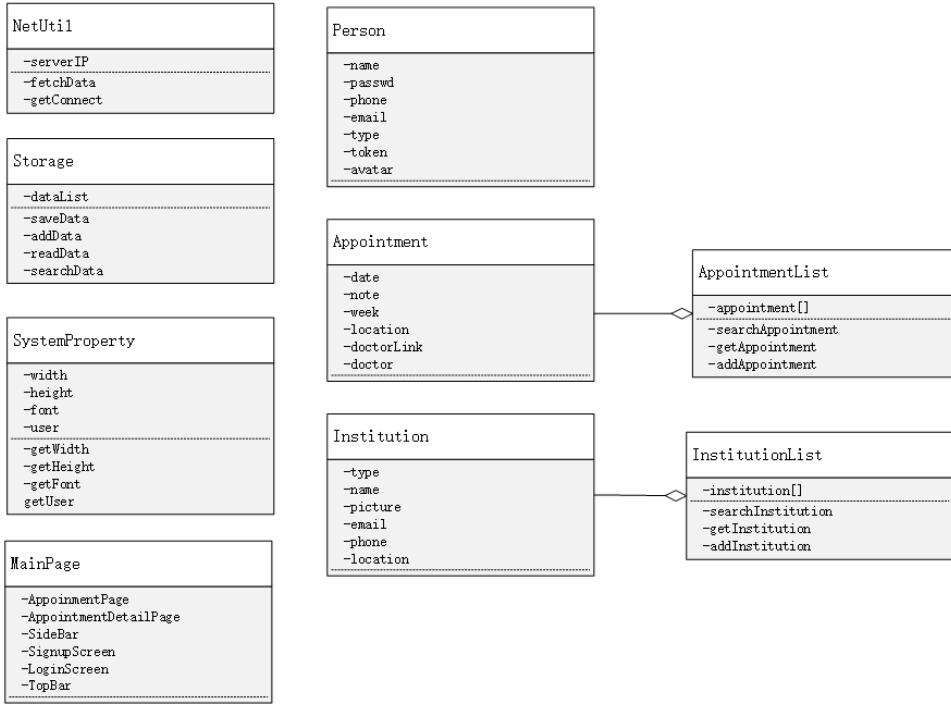


- Sequence View
- Scenario (Use Case):

Medical Secretary Mobile Application



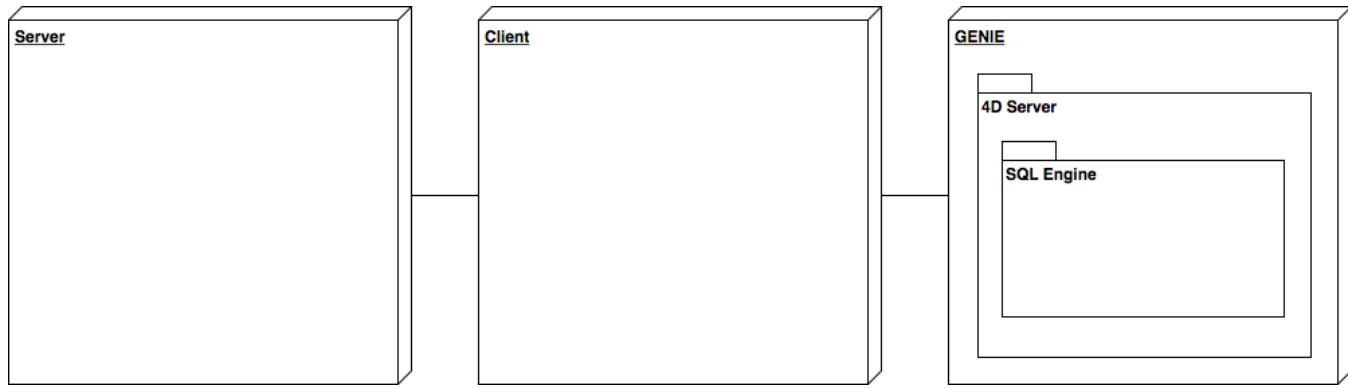
- *Logical view(Class Diagram).*



- Presentation Page:https://docs.google.com/presentation/d/1vCIHgG9_smwFoQ8FBM01DlOOFO4IEq8skI5QsAzWI4/edit?usp=sharing

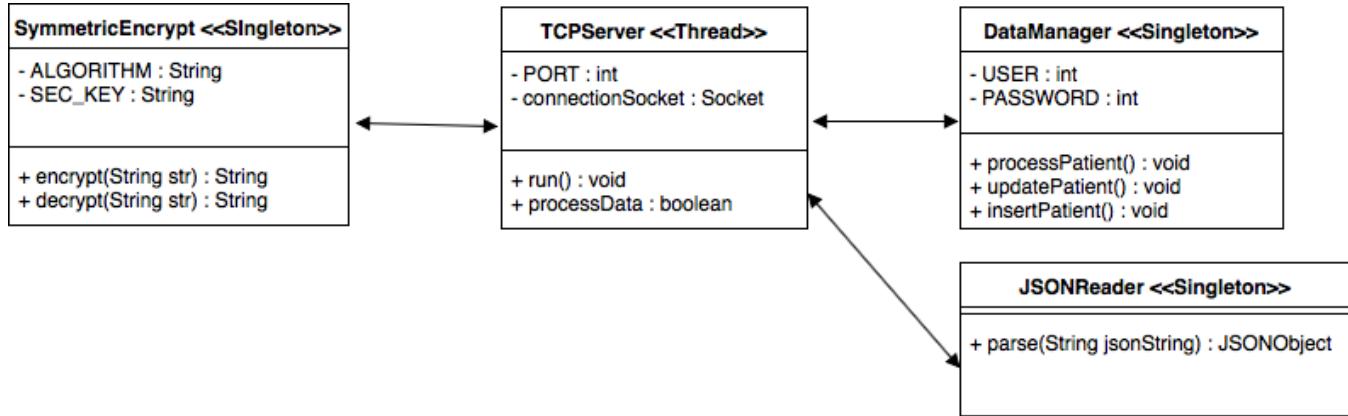
GENIE Script Architecture

High Level Architecture



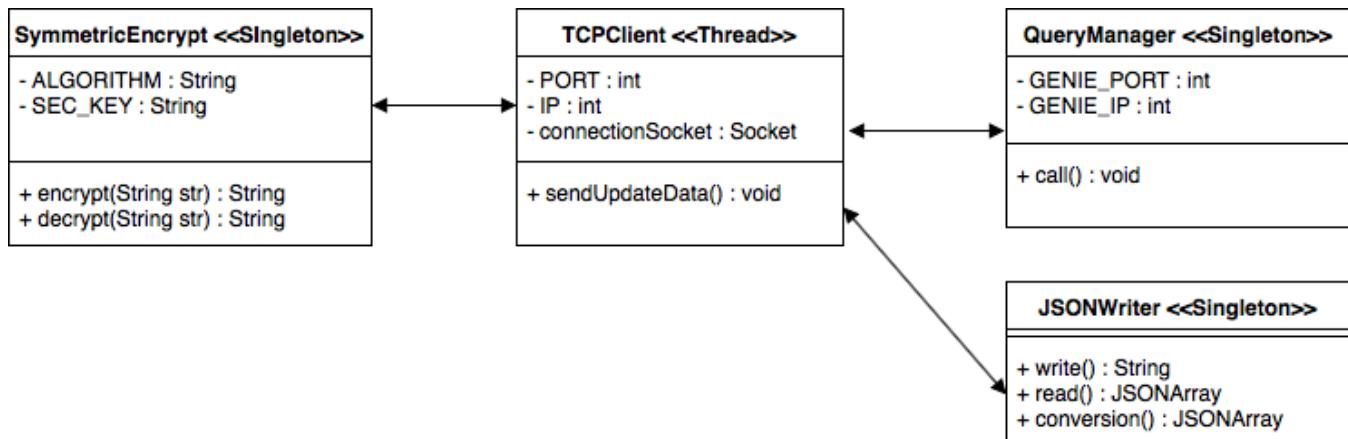
[highlev.xml](#)

Server Class Diagram



[server.xml](#)

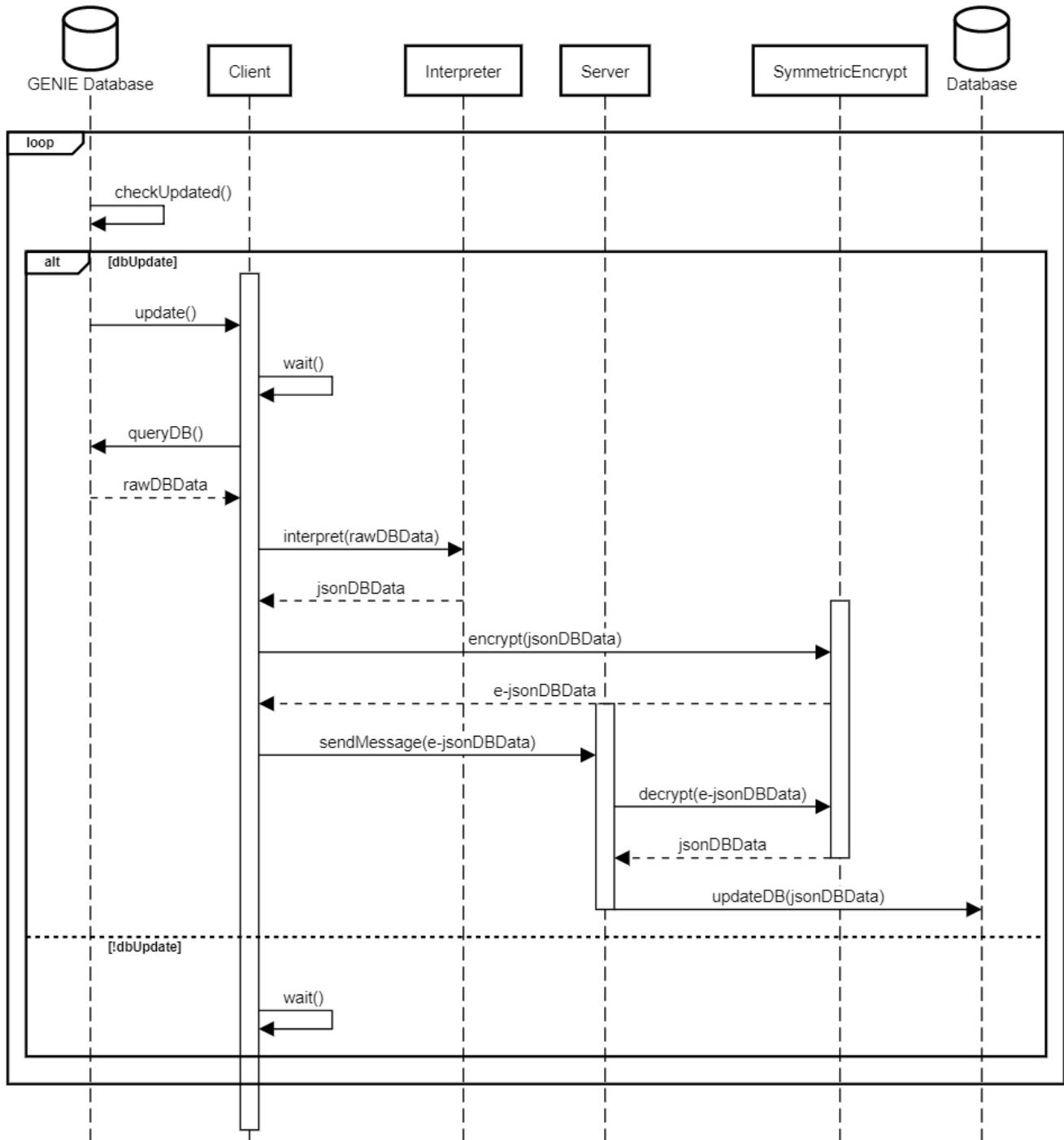
Client Class Diagram



[client.xml](#)

Sequence Diagram

GENIE Script Update



[sequencediagram](#)

High-Level Architecture

 Unknown macro: 'html'

Edit link: [High-level Architecture](#)

High-level architecture has two part: front-end and back-end.

Front End:

It is the mobile application that patients and clinics experience. For the mobile application design, we basically follow the Model-View-Controller or MVC design pattern which can ensure the testability of the code, faster development and make it easy to extend the features.

View:

Corresponds to all the UI logic of the application. This includes everything that is displayed to the user. This, however, should never contain any business logic or data services. The view should be the most lightweight class. Store your UI components and UI logic in this layer, that is the layout of the application and how the UI behaves. For example, the patient wants to view his or her next appointment so he or she first needs to go to the appointment page. The View model's job is to display the patient's appointment.

Presenter:

Corresponds to the interface that interacts between the view and the model. This handles all the business logic / data manipulation etc. The presenter should be the heaviest class. Store your logic inside this layer, that is how the application calls queries, how the application behaves under the hood. "For example, the Appointment controller will handle all the interactions and inputs from the Appointment View and update the database using the Appointment Model. The same controller will be used to view the Appointment data."

Model:

Corresponds to all the data-related logic of the application. This includes local database queries and networked database queries. The model also updates the UI according to information additions/changes/deletions. "For example, a Appointment object will retrieve the appointment information from the database, manipulate it and update it back to the database or use it to render data."

Back End:

It is about the implementation of data handling and integration with Genie system. We have an application server which uses Genie API to communicate with Genie Database for collecting data. Then storing critical data in our own database. It also handles all the requests from mobile application and gets responses to the application.

Technology Research

TBC Items

- SQL or NoSQL Selection
- Admin portal (or whole backend server) Necessity
- Scope and scale of Back-end Server (do we need to involve a framework e.g. Spring)
- ETL Process

Overview

This page lists and compares all the technology candidates that could be applied to the project.

The research is based on the following system component relationship proposal. The aim of this proposal is to identify all possible components and their relationship that could possibly included in the system.

Note that this is *NOT*a final version of design nor a formal architecture design.

 Unknown macro: 'html'

Editable diagram link (draw.io): https://drive.google.com/file/d/1pSsF0cHnQFatkrhcL0ui1uEVe_Dckmpv/view?usp=sharing

Design Principles (and discussions)

• Back-end Server

The current user story and acceptance test mentioned the need of storing and manipulating some kind of data that (maybe) beyond the scope of Genie DB (i.e patient profiles, login, and user interaction etc.). So here we assume that a back-end server is needed to process such kind of data and provide services to mobile clients.

One problem arises is the scope of the back-end server. By now the developed requirement doesn't mentioned too much about the back-end behaviours thus we are unable to determine the technology stack that fits the project. Although, we will try to cover the deviations as much as possible.

More detailed concerns are discussed below.

• MVC (or MVVM, MVP) Framework on Server

The server seems to have very light functions (assumption from requirements) and the (most of) interactive activities are on the mobile applications. This makes the whole View (or VM, P) is detached from the back-end server. It's more efficient (and possible) to achieve a loose coupling between the Mobile and Server, while high cohesion *within each module*. Involving a framework (mostly too heavy) is not ideal to fulfil the above objectives.

• Ajax and RESTful API

To ensure the extendability and reuseability, and loosing coupling between the system parts, we propose to encapsulate the server communications toward the mobile app (and the TBC admin portal) as a submodule of the back-end server. The RESTful style interface via Ajax HTTP requests is good in practice. This would completely isolate the Mobile from the Server and eliminate the dependency to the detailed implementation of each other.

• Admin Portal

The Client confirmed that all the extra data we need to deal with so far is generated in Genie. Interaction on this kind of data is limited. No Admin Portal required.

• DB Technology Alternative

Genie stores data in a D4 database which is a relational one. It also has a integrated SQL server (with SQL expansion it can bypass the limitation on the number of concurrency sql connection)

Client Mobile App

Category	Tech	Relavent Techs	Pros	Cons
----------	------	----------------	------	------

Native Mobile Framework	iOS SDK	Objective-C / Swift, Xcode	<ul style="list-style-type: none"> iOS is operating under a fixed hardware configuration, therefore developers only need to develop software based on the existing hardware configuration and focus more on the user experience of the software. Swift creators acknowledged the fact that in order to build a defining programming language, the technology needs to be open for all. A clean and expressive language with a simplified syntax and grammar, Swift is easier to read and write. It is very concise, which means less code is required to perform the same task, as compared to Objective-C. Objective-C is a fundamentally simple language. Its syntax is unambiguous, and easy to learn. Object-oriented programming, with its self-conscious terminology and emphasis on abstract design. 	<ul style="list-style-type: none"> Have to form two sub-teams to develop 2 versions on Android and IOS separately, needs more time and resource.
	Android SDK	Java / Kotlin, Android Studio	<ul style="list-style-type: none"> In our team, every team member is familiar with Java, and it is easier for us to adapt to that programming language for mobile app development. Android is built mostly in Java, so its adaptation becomes faster and easier. Developing apps in Android is cheaper. The company provides its SDK for free, therefore all the costs will be used to the application testing and deploying. 	
Cross-platform framework / lib	React Native	JavaScript, React	<ul style="list-style-type: none"> It's cross-platform, only one version need to build. Save time and resource. More resource to focus on testing. The code is more modular, making it easier to reuse code. 	<ul style="list-style-type: none"> Few people are familiar with react and JavaScript, need time to learn new tools.

Back-end Server

Category	Tech	Pros	Cons	Relavent Techs
Database	MySQL			SQL
	NoSQL Alternative?			NoSQL
Programming Language for back-end	Java	Overall moderate development experience		
	PHP			
	python			
Server API and Communication methods	RESTful API (for all programming language stream)	<ul style="list-style-type: none"> Loose coupling Uniform interface Highly scalable, easy to modify and extend Stateless 		JSON, JSONP, AJAX, HTTP Request (POST, GET, etc.)
	Java Web Service			Java, J2EE, XML
Server API Lib (Web framework)	Jersey (RESTful, for Java stream)	<ul style="list-style-type: none"> Lightweight JAX-RS compatible (easy to implement / understand) 	<ul style="list-style-type: none"> Configuration Integration with other frameworks 	Java, J2EE, Java Web Services (JAX-RS), RESTful
	Flask (RESTful, for python stream)			python, RESTful
MVC Framework (optional)	Spring			Java, MVC, Web Service
Database Integration	DBCP	<ul style="list-style-type: none"> Optimise db interaction performance manageable datasources 		
	Mybatis (Persistence Framework)	<ul style="list-style-type: none"> POJO mappings Good encapsulate Good way to manage / extent raw SQL statements 	<ul style="list-style-type: none"> Learning curve 	
Integrated Environment	xampp	<ul style="list-style-type: none"> Powerful and convenient, easy to deploy. By using this tool, the whole development environment will be established easily, we just need to develop our website and deploy it. 	<p>It's a integrated tool, including apache, mysql, php, perl</p>	PHP, Mysql

Interact with GENIE

Category	Tech	Available resources	Concerns
Database	D4 database	http://doc.4d.com/4Dv16R5/4D/16-R5/EXPORT-DATA.301-3482174.en.html	It's not very sure whether we can get read/write access to GENIE database.
API	Genie web service API	http://www.geniesolutions.com.au/downloads/common/pdfs/Genie_Guide_Web_Appointments_API	The API given may not be able to complete those functions we need.

ETL

Tech	Pros	Cons
python scripts		
Standalone Desktop Application (Deploy on the client's computer to access Genie DB or Local files)		
A module Integrated with back-end server (Access Genie DB remotely via provided API)		

Misc

- **Network Security: SSL/TLS**

According to the requirements that related to security and confidentiality, we should probably include data / transmission encryption to the system.

- Using a self-signed certificate for development to enable TLS on API is practical.
- We should make certificate configurable so the client can replace the dev certificate with their own ones (doesn't matter if its self-signed or trusted)
- And we should also give the client options to downgrade to insecure connections.
- SSL over IP rather than domain is practical, though not recommended.

- **Push Notification**

According to the requirements that mentioned the (realtime?) notification to the patients, we may need to involve a push notification services to the system. Some concerns about this include:

- [Firebase](#)
- Not possible to implement on iOS - APNs not available for free tier developer accounts, which is the **ONLY** way to do push notification on iOS.
- Also unavailable in Simulator. A simulated iOS device does not have a deviceToken which is required for APNs.
- However services inc. [Firebase](#) also support APNs, as long as a valid certificate is provided.

MVP

When designing the application we should ensure to follow the Model View Controller or MVC design pattern. This ensures faster development processes, ease of modification and easier testability.

MVC is a mobile architecture pattern that separates the design into three components - Model, View and Controller.

View (Presentation Layer):

Corresponds to all the UI logic of the application. This includes everything that is displayed to the user. This, however, should never contain any business logic or data services. The view should be the most lightweight class. Store your UI components and UI logic in this layer, that is the layout of the application and how the UI behaves.

Controller (Business Layer):

Corresponds to the interface that interacts between the view and the model. This handles all the business logic / data manipulation etc. The controller should be the heaviest class. Store your logic inside this layer, that is how the application calls queries, how the application behaves under the hood. "For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data."

Model (Data/Communications Layer):

Corresponds to all the data-related logic of the application. This includes local database queries and networked database queries. The model also updates the UI according to information additions/changes/deletions.

"For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data."

Background

The basic background for this project is an application is needed to assist clinic supply better service to patients.

Doctors in clinics are using a software called Giene to generate all save all kind of documents for different patients who are generally diagnosed with cancer or disease like this.

Thus, they always have tremendous documents to read. As a result, they would always lose their documents, on the other hand, clinic staffs have to spare more efforts to print those documents for patients again, costing time and money.

Besides, patients always have many appointments with doctors, they get notifications via email from clinic, which is inconvenient and easy to forget.

Therefore, clinic needs one app which can show all the paper documents and appointments on patients' phones and pads. On one hand, patients don't need to keep those tremendous documents, which means they don't need to worry about losing those files, also they can read those files whenever they can as long as they got pads/phones at hand. On the other hand, clinic staff don't need to print those files again and again if patients lose their files.

Additionally, appointment notification will become more convenient. Patients can receive notification on phone/pad timely, clinic staff don't need to email or call the patients to notify them.

Client Provided Materials

File	Modified
PDF File Private Specialist Centre.pdf	19 Apr, 2018 by Wenzhuo Mi
PDF File System Overview.pdf	19 Apr, 2018 by Wenzhuo Mi
PDF File Mock Patient Docs 1.pdf	19 Apr, 2018 by Wenzhuo Mi
PDF File Mock Patient Docs 2.pdf	19 Apr, 2018 by Wenzhuo Mi
PDF File Paperprototype from clients.pdf	20 Apr, 2018 by Tong Niu
PDF File Hospital New Patient Pack.pdf	20 Apr, 2018 by Wenzhuo Mi
PDF File Infusaport (Port) Insertion Pack.pdf	20 Apr, 2018 by Wenzhuo Mi
PDF File Chemotherapy Booking.pdf	20 Apr, 2018 by Wenzhuo Mi
PDF File Treatment Information 1.pdf	20 Apr, 2018 by Wenzhuo Mi
PDF File Treatment Information 2.pdf	20 Apr, 2018 by Wenzhuo Mi
PDF File Scans.pdf	20 Apr, 2018 by Wenzhuo Mi
PDF File Pathology.pdf	20 Apr, 2018 by Wenzhuo Mi
PDF File Clarification from client.pdf	27 May, 2018 by Ruoyi Wang
PDF File Clarification from client_Sprint 2.pdf	27 Jul, 2018 by Ruoyi Wang

Drag and drop to upload or [browse for files](#)

 [Download All](#)

Database Access Research

An important implementation question is how the app will draw on the Genie database. This page lists relevant information.

As reflected on the Genie online documentation¹, Genie in client-server mode runs a version of 4D server on the server computer. 4D sometimes implements internet-facing SOAP and SQL servers, as listed on the product page². The client in the second meeting³ confirmed that both servers were able to be run on their clinic's unmodified client-server setup. Additionally, demonstration of the Genie application showed that Genie comes with the 4D server expansion that allows for unlimited concurrent SQL connections rather than 2.

4D has a proprietary framework for building client applications for 4D servers. As the project is not too heavily coupled with the 4D server, and as this will affect portability and maintenance, this is an undesirable option. This leaves the options of SOAP and SQL servers. A SOAP server could potentially allow the app to be very lightweight, using the system's web browser code to open pages populated with data as appropriate. Given the Genie's web API documentation⁴, it appears that this should also not have limited connections. This API uses the WSDL detailed here⁵. An SQL server, while more technical, and sensitive to changes in database definition, is another option detailed here⁶.

References

1. <http://www5.geniesolutions.com.au/manual/HTML/index.html>
2. <http://www.4d.com/au/products/4dv12/4dserver.html>
3. 2018-03-29 Client Meeting
4. http://www.geniesolutions.com.au/downloads/common/pdfs/Genie_Guide_Web_Appointments_API
5. <https://www.w3.org/TR/2001/NOTE-wsdl-20010315>
6. ftp://ftp.ajar.ch/doc/4d_v11/english/4D_v11_SQL_Reference_r3.pdf

Presentation Script

Intro:

Thanks everyone for coming to our project presentation. We'd like to start by acknowledging the traditional custodians of the land, the Wurrundjeri people of the Kulin nation, pay respects to their elders, acknowledge any aboriginal people be present, and remember that this land was never ceded. It's been a great year and we're really excited to share our project with you.

Supervisor:

This is our supervisor, Eduardo, who's been with us from the very start. He's been a great help, whether it's with soft skill organisational stuff about how the project should run, with technical stuff like setting up repos, or general life advice.

Paper 1:

First up, I'd like to say I hate paper. It's hard to read, it's easy to lose, half of it's out of date by the time it's written, and the earth is on its way to becoming inimical to global human society within our lifetime. One of the things I've liked most about uni was everything being online and barely needing to use paper.

Paper 2:

In high school, at least for me, everything was on paper. Every teacher had to print off thirty timetables, thirty forms, thirty handouts, thirty worksheets, and then do it all again every time anyone lost anything. Which happened a lot, because we all had half a dozen classes and were a bit busy learning everything like our life depended on it.

Paper 3:

So what if your life really did depend on it? What if handing in something late or getting your calendar wrong could make you seriously ill? What if all of the material was vital and examinable? What if you were tired and scared and didn't know how long you'd need to keep it up? Okay Leon will say that all the material is vital and examinable but you get the point.

Client:

This is our client is Susan, from Darebin Street Specialist Centre. The centre is a private medical and surgical consulting centre that specialises in oncology, which is the treatment of cancer. For her patients, she's one of the most important members of the treatment team: the person that makes sure everyone gets to their appointments, and has all of the paperwork and documents they need to receive treatment.

Printer:

At the moment, that means Susan spends a lot of time doing two things: reminding people when their appointments are, and printing. That means patients are spending a lot of time doing two things: getting overwhelmed trying to remember appointments and getting overwhelmed managing stacks and stacks of paper. Clearly this is a situation where data needs to go online.

Genie:

Luckily for us, almost all of the data patients might want to see is already hosted on an internet-connected database. For a lot of clinics around Australia, all that data lives in a piece of clinic management software called Genie. The issue is, Genie is for doctors and administrators, with the lack of access control to match.

Accessing that data is the big question - can we get the Genie out of the bottle? We'll talk a bit more about that later.

Societal:

While they're increasingly tech-savvy, patients are in a high-stress, fatigue-heavy situation.

Process:

At the same time, clinic administration staff aren't going to want to spend a lot of time setting up and getting used to a new system.

Pains:

To summarise, we have two problems:

- How much time it takes to pass on information, for patients and clinics.
- Patients have trouble organising, storing, and managing this information.

Motivation:

Which means our project boils down to one thing: making life less stressful. That brings us to our solution, which Gordon will talk to you about.

React Native Firebase Research

Set up Firebase Notification in React-native App (Android only)

What need to be edited and applied is made bold in the code.

Settings

Because currently we don't have Paid Apple Developer Account, the Firebase configuration is only available on Android device.

Firstly, created a new project fcm (should be our project Medical Secretary here).

```
react-native init fcm
cd fcm
yarn
yarn add react-native-firebase
react-native link react-native-firebase
```

Go to Firebase console <https://firebase.google.com/console>, create a project there.

You can know the package name from android/app/src/main/java/com/fcm/MainApplication.java. In that file you can see something like this

```
package com.fcm;
```

It is the package name we need when we create the Android app in Firebase.

Create an Android app, download google-services.json file. Copy that json file to android/app

Go to android/build.gradle

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
```

```
buildscript {
    repositories {
        <b></b>google()
        jcenter()
        maven {
            url 'https://maven.google.com/'
            name 'Google'
        }
    }
    dependencies {
        <b></b>classpath 'com.android.tools.build:gradle:3.1.2'
        classpath 'com.google.gms:google-services:3.2.1'</b>
        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        mavenLocal()
        <b></b>google()
        jcenter()
        maven {
            // All of React Native (JS, Obj-C sources, Android binaries) is installed from npm
            url "$rootDir/../node_modules/react-native/android"
        }
        maven {
            url 'https://maven.google.com/'
            name 'Google'
        }
    }
}
<b>
ext {
    buildToolsVersion = "26.0.3"
    minSdkVersion = 16
    compileSdkVersion = 26
    targetSdkVersion = 26
    supportLibVersion = "26.1.0"
}</b>
```

Go to android/app/build.gradle

```

project.ext.react = [
    entryFile: "index.js"
]

apply from: "../../node_modules/react-native/react.gradle"

/**
 * Set this to true to create two separate APKs instead of one:
 *   - An APK that only works on ARM devices
 *   - An APK that only works on x86 devices
 * The advantage is the size of the APK is reduced by about 4MB.
 * Upload all the APKs to the Play Store and people will download
 * the correct one based on the CPU architecture of their device.
 */
def enableSeparateBuildPerCPUArchitecture = false

/**
 * Run Proguard to shrink the Java bytecode in release builds.
 */
def enableProguardInReleaseBuilds = false

android {
    <b> compileSdkVersion rootProject.ext.compileSdkVersion
    buildToolsVersion rootProject.ext.buildToolsVersion </b>

    defaultConfig {
        applicationId "com.fcm"
        <b> minSdkVersion rootProject.ext.minSdkVersion
        targetSdkVersion rootProject.ext.targetSdkVersion </b>
        versionCode 1
        versionName "1.0"
        ndk {
            abiFilters "armeabi-v7a", "x86"
        }
    }
    splits {
        abi {
            reset()
            enable enableSeparateBuildPerCPUArchitecture
            universalApk false // If true, also generate a universal APK
            include "armeabi-v7a", "x86"
        }
    }
    buildTypes {
        release {
            minifyEnabled enableProguardInReleaseBuilds
            proguardFiles getDefaultProguardFile("proguard-android.txt"), "proguard-rules.pro"
        }
    }
    // applicationVariants are e.g. debug, release
    applicationVariants.all { variant -
        variant.outputs.each { output -
            // For each separate APK per architecture, set a unique version code as described here:
            // http://tools.android.com/tech-docs/new-build-system/user-guide/apk-splits
            def versionCodes = ["armeabi-v7a":1, "x86":2]
            def abi = output.getFilter(OutputFile.ABI)
            if (abi != null) { // null for the universal-debug, universal-release variants
                output.versionCodeOverride =
                    versionCodes.get(abi) * 1048576 + defaultConfig.versionCode
            }
        }
    }
    dependencies {
        <b> implementation project('react-native-firebase')
        implementation fileTree(dir: "libs", include: ["*.jar"])
        implementation "com.android.support:appcompat-v7:${rootProject.ext.supportLibVersion}"
        implementation "com.facebook.react:react-native:+" // From node_modules
        implementation "com.google.android.gms:play-services-base:15.0.0"
        implementation "com.google.firebase:firebase-core:15.0.2"
        implementation "com.google.firebase:firebase-messaging:15.0.2"
        implementation 'me.leolin:ShortcutBadger:1.1.21@aar'</b>
    }
}

// Run this once to be able to run the application with BUCK
// puts all compile dependencies into folder libs for BUCK to use
task copyDownloadableDepsToLibs(type: Copy) {
    from configurations.compile
    into 'libs'
}
<b> apply plugin: 'com.google.gms.google-services'</b>

```

Go to android/gradle/wrapper/gradle-wrapper.properties

```
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https://services.gradle.org/distributions/<b>gradle-4.4-all.zip</b>
```

Go to android/app/src/main/java/com/fcm/MainApplication.java

```
package com.fcm;
import android.app.Application;
import com.facebook.react.ReactApplication;
import io.invertase.firebaseio.RNFirebasePackage;
<b>import io.invertase.firebaseio.messaging.RNFirebaseMessagingPackage;
import io.invertase.firebaseio.notifications.RNFirebaseNotificationsPackage;</b>
import com.facebook.react.ReactNativeHost;
import com.facebook.react.ReactPackage;
import com.facebook.react.shell.MainReactPackage;
import com.facebook.soloader.SoLoader;
import java.util.Arrays;
import java.util.List;
public class MainApplication extends Application implements ReactApplication {
    private final ReactNativeHost mReactNativeHost = new ReactNativeHost(this) {
        @Override
        public boolean getUseDeveloperSupport() {
            return BuildConfig.DEBUG;
        }
        @Override
        protected List<ReactPackage> getPackages() {
            return Arrays.<ReactPackage>asList(
                new MainReactPackage(),
                new RNFirebasePackage(),
                <b>new RNFirebaseMessagingPackage(),
                new RNFirebaseNotificationsPackage()</b>
            );
        }
        @Override
        protected String getJSMainModuleName() {
            return "index";
        }
    };
    @Override
    public ReactNativeHost getReactNativeHost() {
        return mReactNativeHost;
    }
    @Override
    public void onCreate() {
        super.onCreate();
        SoLoader.init(this, /* native exopackage */ false);
    }
}
```

Go to android/app/src/main/AndroidManifest.xml

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.fcm">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.VIBRATE" />

    <application
        android:name=".MainApplication"
        android:label="@string/app_name"
        android:icon="@mipmap/ic_launcher"
        android:allowBackup="false"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
            android:windowSoftInputMode="adjustResize"
            android:launchMode="singleTop">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="com.facebook.react.devsupport.DevSettingsActivity" />
        <service android:name="io.invertase.firebaseio.messaging.RNFirebaseMessagingService">
            <intent-filter>
                <action android:name="com.google.firebaseio.MESSAGING_EVENT" />
            </intent-filter>
        </service>
        <service android:name="io.invertase.firebaseio.messaging.RNFirebaseInstanceIdService">
            <intent-filter>
                <action android:name="com.google.firebaseio.INSTANCE_ID_EVENT" />
            </intent-filter>
        </service>
        <service android:name="io.invertase.firebaseio.messaging.RNFirebaseBackgroundMessagingService" />
    </application>

</manifest>

```

Try to run it to see if there is any problems

react-native run-android

Coding

Go to your App.js file

```

/**
 * Sample React Native App
 * https://github.com/yangnanall/react-native-fcm-demo
 *
 * @format
 * @flow
 */
import React, {Component} from 'react';
import {Platform, StyleSheet, Text, View} from 'react-native';
<b>import firebase from 'react-native-firebase'</b>
import type { Notification, NotificationOpen } from 'react-native-firebase';</b>

const instructions = Platform.select({
  ios: 'Press Cmd+R to reload,\n' + 'Cmd+D or shake for dev menu',
  android:
    'Double tap R on your keyboard to reload,\n' +
    'Shake or press menu button for dev menu',
});

export default class App extends Component {
  <b>async componentDidMount() {</b>
    // Before you are able to send and receive Cloud Messages,
    // you need to ensure that the user has granted the correct permissions
    const enabled = await firebase.messaging().hasPermission();
    if (enabled) {
      // user has permissions
    } else {
      // user doesn't have permission
      try {
        await firebase.messaging().requestPermission();
        // User has authorised
      } catch (error) {

```

```

        // User has rejected permissions
        alert('No permission for notification');
    }
}
//console.log("delete")
//firebase.iid().deleteToken();
// Retrieve the current registration token
const fcmToken = await firebase.messaging().getToken();
if (fcmToken) {
    // user has a device token
    console.log("token: " + fcmToken)
} else {
    // user doesn't have a device token yet
    alert('No device token yet');

}

// The onTokenRefresh callback fires with the latest registration token whenever a new token is generated.
firebase.messaging().onTokenRefresh(fcmToken => {
    // Process your token as required
    console.log("refreshtoken: " + fcmToken)

});

// when app is closed
const notificationOpen: NotificationOpen = await firebase.notifications().getInitialNotification();
if (notificationOpen) {
    // App was opened by a notification
    // Get the action triggered by the notification being opened
    const action = notificationOpen.action;
    const notification: Notification = notificationOpen.notification;
    if (notification.body!==undefined) {
        alert(notification.body);
    } else {
        console.log(notification)
        var seen = [];
        alert(JSON.stringify(notification.data, function(key, val) {
            if (val != null && typeof val == "object") {
                if (seen.indexOf(val) >= 0) {
                    return;
                }
                seen.push(val);
            }
            return val;
        }));
    }
}
firebase.notifications().removeDeliveredNotification(notification.notificationId);
}

const channel = new firebase.notifications.Android.Channel('test-channel', 'Test Channel', firebase.notifications.Android.Importance.Max)
    .setDescription('My apps test channel');
// Create the channel
firebase.notifications().android.createChannel(channel);

firebase.messaging().subscribeToTopic('news1');

/*
when app is opened
*/
// onNotificationDisplayed - Triggered when a particular notification has been displayed
this.notificationDisplayedListener = firebase.notifications().onNotificationDisplayed((notification: Notification) => {
    // Process your notification as required
    // ANDROID: Remote notifications do not contain the channel ID. You will have to specify this manually if you'd like to re-display the notification.
    console.log('onNotificationDisplayed')
    console.log(notification);
});

// onNotification - Triggered when a particular notification has been received
this.notificationListener = firebase.notifications().onNotification((notification: Notification) => {
    // Process your notification as required
    console.log('get Message');
    console.log(notification);
    notification
        .android.setChannelId('test-channel')
        .android.setSmallIcon('ic_launcher');
    firebase.notifications()
        .displayNotification(notification);

});

```

```

// If your app is in the foreground, or background,
// you can listen for when a notification is clicked / tapped / opened as follows:
this.notificationOpenedListener = firebase.notifications().onNotificationOpened((notificationOpen:
NotificationOpen) => {
    // Get the action triggered by the notification being opened
    const action = notificationOpen.action;
    // Get information about the notification that was opened
    const notification: Notification = notificationOpen.notification;
    // if app is in the foreground, notification is defined
    if (notification.body!==undefined) {
        alert(notification.body);
    } else {
        // if app is in the background, notification is undefined
        console.log(notification)
        var seen = [];
        alert(JSON.stringify(notification.data, function(key, val) {
            if (val != null && typeof val == "object") {
                if (seen.indexOf(val) >= 0) {
                    return;
                }
                seen.push(val);
            }
            return val;
        }));
    }
    firebase.notifications().removeDeliveredNotification(notification.notificationId);
});

componentWillUnmount() {
    this.notificationDisplayedListener();
    this.notificationListener();
    this.notificationOpenedListener();
}

render() {
    return (
        <View style={styles.container}>
            <Text style={styles.welcome}>Welcome to React Native!</Text>
            <Text style={styles.instructions}>To get started, edit App.js</Text>
            <Text style={styles.instructions}>{instructions}</Text>
        </View>
    );
}
const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'center',
        alignItems: 'center',
        backgroundColor: '#F5FCFF',
    },
    welcome: {
        fontSize: 20,
        textAlign: 'center',
        margin: 10,
    },
    instructions: {
        textAlign: 'center',
        color: '#333333',
        marginBottom: 5,
    },
});

```

Test

There are two [types of message](#) in that one way to send **Generic Notification** using [Notification Composer](#):

As per documentation,

- **Notification Message** - FCM automatically displays the message to end-user devices on behalf of the client app. Notification messages have a predefined set of user-visible keys and an optional data payload of custom key-value pairs.

- **Data Message** - Client app is responsible for processing data messages. Data messages have only custom key-value pairs.

Send Data Message using HTTP protocol with POSTMAN

You have to copy **Legacy Server Key** from *Firebase Console > Project Settings > Cloud Messaging*

Note: Firebase has upgraded our server keys to a new version. You may continue to use your Legacy server key, but it is recommended that you upgrade to the newest version.

- Select POST. Enter request URL as <https://fcm.googleapis.com/fcm/send>
- Add **Headers** Authorization: key=<legacy_server_key> OR Authorization: key=<server_key> and Content-Type: application/json.
- Now Select **Body > raw > JSON (application/json)** and add following code:

```
{
  "to" : <b>"YOUR_FCM_TOKEN_WILL_BE_HERE"</b>,
  "collapse_key" : "type_a",
  "notification" : {
    "body" : "Body of Your Notification",
    "title": "Title of Your Notification"
  },
  "data" : {
    "body" : "Body of Your Notification in Data",
    "title": "Title of Your Notification in Title",
    "key_1" : "Value for key_1",
    "key_2" : "Value for key_2"
  }
}
```

- Now You can send a **Generic** notification (using notification payload) or a Custom notifications (using notification and data payload) and Click on **Send**.

```
{
  "to" : <b>"YOUR_FCM_TOKEN_WILL_BE_HERE"</b>,
  "collapse_key" : "type_a",
  "data" : {
    "body" : "Sending Notification Body From Data",
    "title": "Notification Title from Data",
    "key_1" : "Value for key_1",
    "key_2" : "Value for key_2"
  }
}
```

- Note that **Custom** notification will only trigger if there is only data (without notification) node in the payload. Hence, you'd need to move the body and title to data node.

Inception

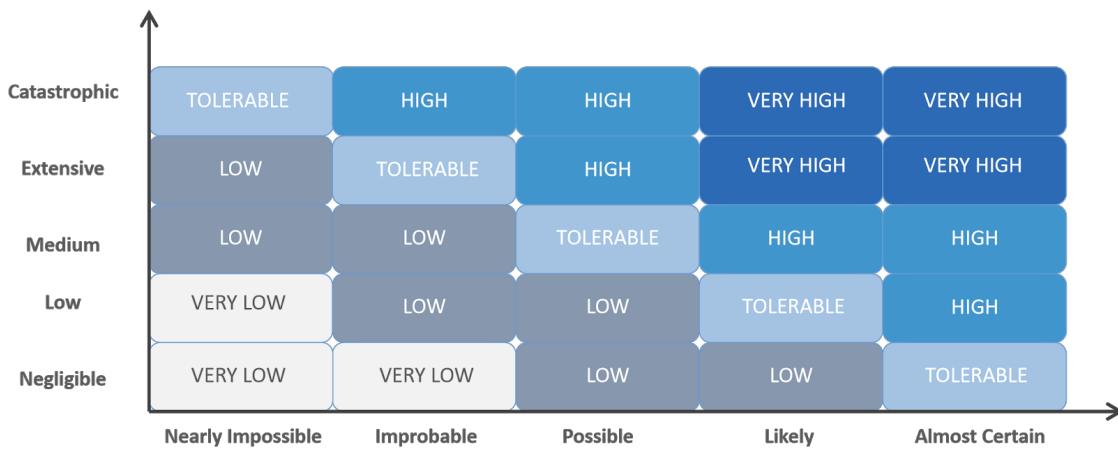
- Goal model, including quality goals
- User stories and personas
- Risk register
- Use of JIRA
- Assignment of roles for inception and for sprint 0 (to be reviewed for Sprint 1)
- Research on relevant technology
- Identify technology skills to be acquired
- Decide on development environments and architecture where applicable

Prioritised Risk Register

Risk Strategies: Avoid / Mitigate / Transfer / Accept

Level of Probability (Percentage)	Level of Impact (Score)
Nearly impossible (10%)	negligible (1)
Improbable (10% < & 25%)	Low (2)
Possible (25% < & 50%)	Medium (3)
Likely (50% < & 80%)	Extensive (4)
Almost Certain (80% < & < 100%)	Catastrophic (5)

Risk Exposure = Probability * Score for Impact



*The chart was described (Institute of Chartered Accountants Australia 2013).

Product Risk Register

Risk Scenario	Possible Trigger(s)	Risk Description	Pr o b a bili ty (0 - 1) t)	I m p a ct y (0 - 5) t)	Risk Exposure (Probability * Impact)	Risk Control Strategy
Users will not use the application	<ul style="list-style-type: none"> ■ Bad Usability (Hard to use) ■ Slow response times, unfriendly UI ■ Buggy application, makes the application unreliable ■ Application uses too much storage space ■ Patients don't know about the application ■ Patients continue to use paper forms 	The interface of the Application is very important for users. Since a part of users is patient and the age is mainly above the age of forty, the interface should be more user-friendly with clear guide.	0.6	5	3.0	<p>Mitigate</p> <ul style="list-style-type: none"> ■ After the conclusion of every sprint, ensure that the client spends time to use the application. This allows the team to get valuable feedback (feelings, concerns, positives) ■ Make test cases are written and are followed effectively (pass all test cases, high coverage) This ensures that as many bugs are resolved as possible ■ Separate patients in terms of technical knowledge, allow them to test the user experience of the application (for example, ask them to navigate to a page) so that we know how usable the application is.

Project Risk Register

Risk Scenario	Possible Trigger(s)	Risk Description	Probability (0 - 1)	Impact (0 - 5)	Risk Exposure (Probability * Impact)	Risk Control Strategy
Client makes major changes to the requirements	<ul style="list-style-type: none"> Client expresses some new ideas for the project Client expresses dissatisfaction about the project state Invalid requirements from clients hard to achieve The client's requirements are misunderstood by the implementation team Inadequate requirements 	The project delay might result from the unrealistic or unreasonable requirements therefore affecting the quality of project with expectation.	0.7	5	3.5	<p>Avoid:</p> <ul style="list-style-type: none"> Frequent communications with the client about the project state <p>Mitigate:</p> <p>Some padding to the project timeline to allow for this risk</p> <ul style="list-style-type: none"> Having a policy in the project contract for late-stage changes Decide the main direction and certain details of scope before moving the project from one Sprint to the next Sprint <p>Transfer:</p> <ul style="list-style-type: none"> Consult the major change with client after discussion and consideration for suggestions about managing all kinds of changes effectively.
The project delay (below 95%)	<ul style="list-style-type: none"> The unreasonable schedule Some parts of the project are not taken into consideration in the whole schedule due to the requirements changes of the client The implementation process unable to follow the time schedule 	The implementation might not be executed as expected due to the unreasonable time schedule, which is not only causing in poor quality of the App project but also triggering team members are not subject to rule or discipline	0.3	4	1.2	<p>Avoid:</p> <ul style="list-style-type: none"> Negotiate with Client to create a reasonable schedule taking all factors and incidents into consideration. Clearly established requirements for each Sprint that are signed off by all parties involved Focus project attention on modules required to achieve milestones <p>Mitigate:</p> <ul style="list-style-type: none"> Communication with the client ahead of time if schedule slips look likely
Disharmonious atmosphere amongst the team	<ul style="list-style-type: none"> Team members interrupt each other and prevent other team members from speaking during meetings Team members argue and fight amongst each others Members of project team possess negative and passive attitudes towards the project 	<p>Communication is a challenge in the project team since it will cause the misunderstanding of team members meaning and increase the potentially related conflicts.</p> <p>The lack of communication during project process might result in difficult teamwork when some divergence appeared.</p>	0.2	2	0.4	<p>Avoid:</p> <ul style="list-style-type: none"> Have team building exercises prior to project commencement Celebrate team achievements with social events <p>Mitigate:</p> <ul style="list-style-type: none"> Individually take team members aside to discuss any anti-team behaviour when it occurs
Team members are unfamiliar with some development or testing tools and technologies	<ul style="list-style-type: none"> A tool or technology has been chosen to be used that is not normally used by the development team The lack of enough working experience in the project When staff (Programmer and constructors) turnover, there will be problems with the task handover 	<p>Each team member has different education and technical background</p> <p>Team members may not have much time to learn the new tools or technologies</p>	0.3	1	0.3	<p>Avoid:</p> <ul style="list-style-type: none"> If appropriate, favour the use of familiar tools and technologies <p>Mitigate:</p> <ul style="list-style-type: none"> Running training workshop prior to commencing the project
The member is absent the meetings and not join discussion online	<ul style="list-style-type: none"> The member is too busy to make it The member expresses dissatisfaction with current plan of the project 	Each team member has his or her own schedule	0.3	2	0.6	<p>Mitigate:</p> <ul style="list-style-type: none"> The project manager should organize the group meeting frequently, members can communicate with each other about schedule and ideas in time Ask for the reasons, negotiate and arrange alternative time for meeting

Technical Risk Register

Risk Scenario	Possible Trigger(s)	Impact Description	Prob ability (0 - 1)	Im pact (0 - 5)	Risk Expo sure (Pro babili ty * Impa ct)	Risk Control Strategy
Bad application design	<ul style="list-style-type: none"> ■ The confidential information of patients needs to be secure such that it does not leak ■ Poor application user interface ■ Application takes a long time to load, patients have to wait ■ Poor application architecture ■ Application doesn't work well on all phones 	Security, functionality and compatibility are major parts of the satisfaction of the user. Poor design of these functionalities could have a major impact on management and future development.	0.6	5	3.0	<p>Mitigate</p> <ul style="list-style-type: none"> ■ Ensure that database queries are encrypted, user is properly authenticated, don't allow unauthorised database queries ■ After the conclusion of every sprint, ensure that the client spends time to use the application. This allows the team to get valuable feedback (feelings, concerns, positives) ■ Make test cases are written and are followed effectively (pass all test cases, high coverage) This ensures that as many bugs are resolved as possible ■ Make sure the application architecture is written up and followed ■ Test on as many devices as possible, make sure to test on devices >= target version
Bad maintenance of the project	<ul style="list-style-type: none"> • Test cases not written or used • Bad coding practices (High coupling, improper object oriented programming, lack of comments etc) 	Bad maintenance of the project is detrimental to the current and future development of the project. It means that team members are not able to work efficiently as time is spent understand and fixing improper code.	0.6	3	1.8	<p>Avoid</p> <ul style="list-style-type: none"> ■ Make sure that if possible, high coverage test cases are written alongside code. This ensures that the whole project is well tested. ■ Write code to conform to a common coding style (QA lead can specify this) ■ Make sure that code reviews are completed, make sure that any change also has a pull request
Problems whilst linking with the GENIE database	<ul style="list-style-type: none"> • The GENIE database may not provide API 	If the GENIE database cannot be accessed remotely, this is an extremely impactful issue. It means that the core functionality of the application needs to be modified.	If it hap pen ed, 1; oth erw ise, 0	5		<p>Transfer</p> <ul style="list-style-type: none"> • Find the alternative plan to solve <p>Avoid</p> <ul style="list-style-type: none"> • Contact with client to avoid this risk

Requirements

Acceptance Test

An acceptance test is a formal description of the behavior of a software product, generally expressed as an example or a usage scenario. A number of different notations and approaches have been proposed for such examples or scenarios. In many cases the aim is that it should be possible to automate the execution of such tests by a software tool, either ad-hoc to the development team or off the shelf.

An acceptance test generally has a binary result, pass or fail. A failure suggests, though does not prove, the presence of a defect in the product.

Acceptance tests are used as the main form of functional specification and the only formal expression of business requirements.

Goal Model

A goal model is an element of requirements engineering that may also be used more widely in business analysis.

Goals are objectives which the system should achieve through cooperation of actors in the intended software and in the environment. Goal modelling is especially useful in the early phases of a project. Projects may consider how the intended system meets organizational goals, why the system is needed and how the stakeholders' interests may be addressed.

A goal model:

- Expresses the relationships between a system and its environment. The understanding this gives, of the reasons why a system is needed.
- Clarifies requirements : Specifying goals leads to asking "why" and "how". Stakeholders' requirements are often revealed in this process, with less risk of either missing requirements, or of over-specifying (asking for things that are not needed).
- Allows large goals to be analyzed into small, realizable goals
- Deals with conflicts : goal modelling can identify and help to resolve tradeoffs between cost, performance, flexibility, security and other goals. It can reveal divergent interests between stakeholders. It can identify conflicts because meeting one goal can interfere with meeting other goals.
- Enables requirement completeness to be measured: requirements can be considered complete if they fulfil all the goals in the goal model.

Personas

A persona defines an archetypical user of a system, an example of the kind of person who would interact with it. In other words, personas represent fictitious people which are based on the knowledge of real users.

Personas are ideal to discuss solutions and to make decisions about alternative designs. When working with them, it will become quickly obvious that there is not one user. A user can be contradictory, sometimes cautious, sometimes daring, contemplative and quick to make decisions.

The goal is to cover all potential users of an application, a product or a system as extensively as possible. In all cases, a persona has a name and a face (in form of a picture). Depending on the desired or required depth of detail more characteristics may be added, e.g. job description, marital status, goals, needs, wishes, education, knowledge, attitude toward the product, hobbies, expectations and so on.

Product Backlog

The product backlog comprises an ordered list of product requirements that a scrum team maintains for a product. These will define features, bug fixes, non-functional requirements. The product owner prioritizes product backlog items (PBIs) based on considerations such as risk, business value, dependencies, size, and date needed.

The product backlog is what will be delivered, ordered into the sequence in which it should be delivered.

User Stories

A user story is a tool to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement.

A user story template often uses the following type of format:

As a <role>, I want <feature> so that <reason>.

A user story is meant to be short. The user stories should be written by the business in the language of the customer so that it is clear to both the business and the development team what the customer wants and why he wants it. The development team's job is to take care of how to develop the code that will satisfy the requirements of the user story. In best-case scenarios, developers collaborate closely with the business owners to clarify the details as the code gets developed.

Acceptance Test

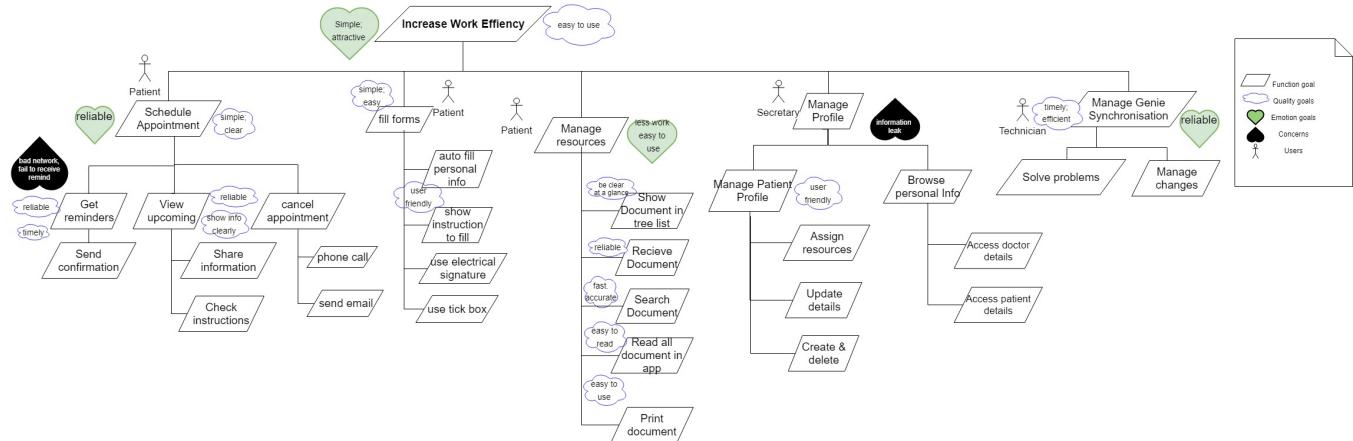
Category	Test Case ID	User Story	Precondition	Execution Step	Expected Results	Test Result	Comments
Account	T1.1	As a patient, I want my account bind my phone number, email, and family member contact, so that my family member and me can receive information timely.	User first time access to the system with a new account without personal details. Already login	1. Patient add the contact details of both themselves and their families. Click submit. 2. All added person check the confirm email.	1. Successfully add the contact details(phone number, email) of their families and themselves. After clicking submit, the system will show the added information on the profile page. 2. All added person will receive the same email (confirm email) timely and simultaneously.		
	T1.2	As a patient, I want to edit my personal profile, so that I can modify my personal detail when my personal information changes.	Patients have their current personal details. Already login	1. Patients click personal profile and choose edit. 2. Patient click the save button. 3. Check if the details is changed where it is been referred.	1. Show current personal details with the editable mode. 2. Check if the change is valid. ValidSave, invalidwarning. 3. The relevant document which referred to this should update synchronously. e.g. choose appointment, system automatically enters personal information which is the newest.		
	T1.3	As a patient, I want to reset my password, so that I can find my account back when I forget my password.	Patients have their Account. Not login	1. Patients click forget passwords through login page. 2. Patients input the new passwords and click submit.	1. Patients receive a confirm email to change the passwords. 2. Patient redirect to login page to login with the new passwords.		
Document	T2.1	As a patient, I want to receive new document when I have one, so that I can know the newest information.	User has logged in	1.Doctor/Clinic staff generate new documents for patient, and send to them.	1.Patients can receive those new documents in the app.		
	T2.2	As a patient, I want to receive the reminder each time I receive a new document, so that I will not miss it.	User has logged in,	1.Doctor/Clinic staff generate new documents for patient, and send to them.	1.Patients will receive notification on their phone/pad in the form of ring or shaking.		
	T2.3	As a patient, I want to see all the documents related to me in the app, so that I can read any of them when I need.	User has logged in,	1.Patient click the document tab.	1.Patient can see all the documents arranged in the page.		
	T2.4	As a patient, I want to see all the documents related to me arranged in a simple and clear structure, so that I can easily know what documents I have and find what I need.	User has logged in,	1.Patient click the document tab. 2.Patient click each SU bccategory .	1.Patient can see all the documents in document tab. 2.Patient can see all documents which are of the same type are in the same category.		
	T2.5	As a patient, I want to view documents in the app on my phone/pad, so that I do not need to print them.	User has logged in,	1.Patient choose and click one document.	1.Patient can view the content of the document, zoom in and out.		
Forms	T3.1	As a patient, I want the app automatically enters the personal information when I make an appointment, so that I just need to enter the appointment relevant description.	User has already logged in	1. Patient clicks button for booking an appointment 2. Patient enters the page of filling appointment form 3. The personal details of patient are auto-filled in the form 4. Patient enter the appointment relevant description	1. The appointment form is auto-filled with patients information details 2. Patient can type content in the description area		

	T3.2	As a patient, I want to see the instructions of filling forms, so that I can follow it to fill the forms correctly.	User has already logged in	<ol style="list-style-type: none"> 1. Patient clicks the form button 2. Patient enters the form page 3. Patient clicks the instruction button on each top of the form 4. The relevant instructions about filling that form are displayed 	<ol style="list-style-type: none"> 1. The form instructions can be shown for each form on the app 2. The instructions can be closed if the patient clicks close button 		
	T3.3	As a patient, I want to fill my health history form and save inside the app, so that I can save it in case it will be used later.	User has already logged in. On the form page	<ol style="list-style-type: none"> 1. Patient fills the relevant areas in his or her health history form 2. Patients clicks the save button 3. Check if the form with filled content is available in app 	<ol style="list-style-type: none"> 1. All areas in the health history form can be filled 2. After clicking the saving button and open the form again, the information that has been filled is kept with the form 		
	T3.4	As a patient, I want to receive the reminders for the form that I need to fill, so that I can fill it in time.	User has already logged in	<ol style="list-style-type: none"> 1. Patient receives the reminders from app to tell him or her there are new forms available and need to be filled 2. Patient opens the app and click form button 3. The new forms for next appointment are found 	<ol style="list-style-type: none"> 1. Successfully the reminders sent from app 2. When opens the app, new forms can be found within app and are available to be edited 		
	T3.5	As a patient, I want to send the filled forms back through the email, so that I can contact with my doctor.	User has already logged in On the form page	<ol style="list-style-type: none"> 1. Patient clicks the email button after filling the form 2. Patient enters the email address of receiver or just clicks the name of contact person 3. Patient clicks sent button 4. An email attached with that form is sent to the receiver 	<ol style="list-style-type: none"> 1. Patient is directed to the email editor when he or she clicks the email button 2. Patient can edit the content of email in the email editor 3. The receiver successfully received an email with any content or form attachment added by the patient 		
Appointment	T4.1	As a patient, I want to confirm my new appointment in the app so that the clinic know that I remember this appointment.	User already make an appointment through phone call or email. Then the secretary would send the confirmation notification through the app.	<ol style="list-style-type: none"> 1. Patient receive the notification and view it. 2. Patient click confirm button on the appointment detail page 	<ol style="list-style-type: none"> 1. Successfully confirm the appointment. 2. A message would send confirmation message to the secretary. 		
	T4.2	As a patient, I want to check all my appointment list by terms, such as time and types, so that I can know when is the appointment and what is it used for.	Patients have their current appointments. Appointment page	<ol style="list-style-type: none"> 1. Patients click filter button 2. Patients choose appointment sort by time, location or types 	<ol style="list-style-type: none"> 1. Show current appointment by time 2. Show current appointment by type 3. Show current appointment by location 		
	T4.3	As a patient, I want to know the detail of my appointment, such as date, location and type so that I won't forget and know where to go.	Patients have their current appointments Appointment page	<ol style="list-style-type: none"> 1. Patients click one appointment to go the detail page. 	<ol style="list-style-type: none"> 1. Patient can see the time, location, doctor and type of the appointment on the detail page. 		
	T4.4	As a patient, I want to know the time of all the appointments so that I can schedule my time table.	Patients have their current appointments Appointment page	<ol style="list-style-type: none"> 1. Patient choose on appointment. 	<ol style="list-style-type: none"> 1. Patient see the time on the detail page. 		
	T4.5	As a patient, I want to view the appointment instructions so that I can prepare for it.	Patients check one appointment page detail in the app.	<ol style="list-style-type: none"> 1. Patient find the instructions section and see the notes leave by the clinic. 	<ol style="list-style-type: none"> 1. Patient see the notes leave by the clinic or secretary. 2. Patient get to know what to prepare 		

	T4.6	As a patient, I want to attach some notes on the application so that it can remind me the important things.	Patients check one appointment detail page in the app.	<ol style="list-style-type: none"> 1. Patients click the add notes button. 2. Patients type what they want to record. 3. After editing, patients click the save button to save the notes. 	<ol style="list-style-type: none"> 1. Patient edit the note and type in what they think is important. 2. Patient save the notes and can view it other times. 		
	T4.7	As a patient, I want the application can give me notification before my next appointment so that I won't forget it.	Patients' next appointment will be in tomorrow or 2 hours	<ol style="list-style-type: none"> 1. Patient get the notification by the app. 2. Patient view the notification 	<ol style="list-style-type: none"> 1. Patients view the notification and know the time of next appointment. 		
	T4.8	As a patient, I want to know the right number to call on the appointment detail page so that I can reschedule or cancel the appointment	Patients want to cancel or reschedule their appointments	<ol style="list-style-type: none"> 1. Patients choose the appointment they want to cancel or reschedule. 2. Patients go the appointment detail page and find the contact detail section. 3. Patients click the phone number to make the phone call. 	<ol style="list-style-type: none"> 1. Patients find the right number to call for each appointment. 		
	T4.9	As a patient, I want to share my appointment schedule to people such as family members, so that they can know my time table.	Patients have current appointment and want to share.	<ol style="list-style-type: none"> 1. Patients choose the appointment they want to share. 2. Patients go the appointment detail page and click the share button. 3. Patients choose the share way 	<ol style="list-style-type: none"> 1. Patients share their appointment through email or text message. 		

Goal Model

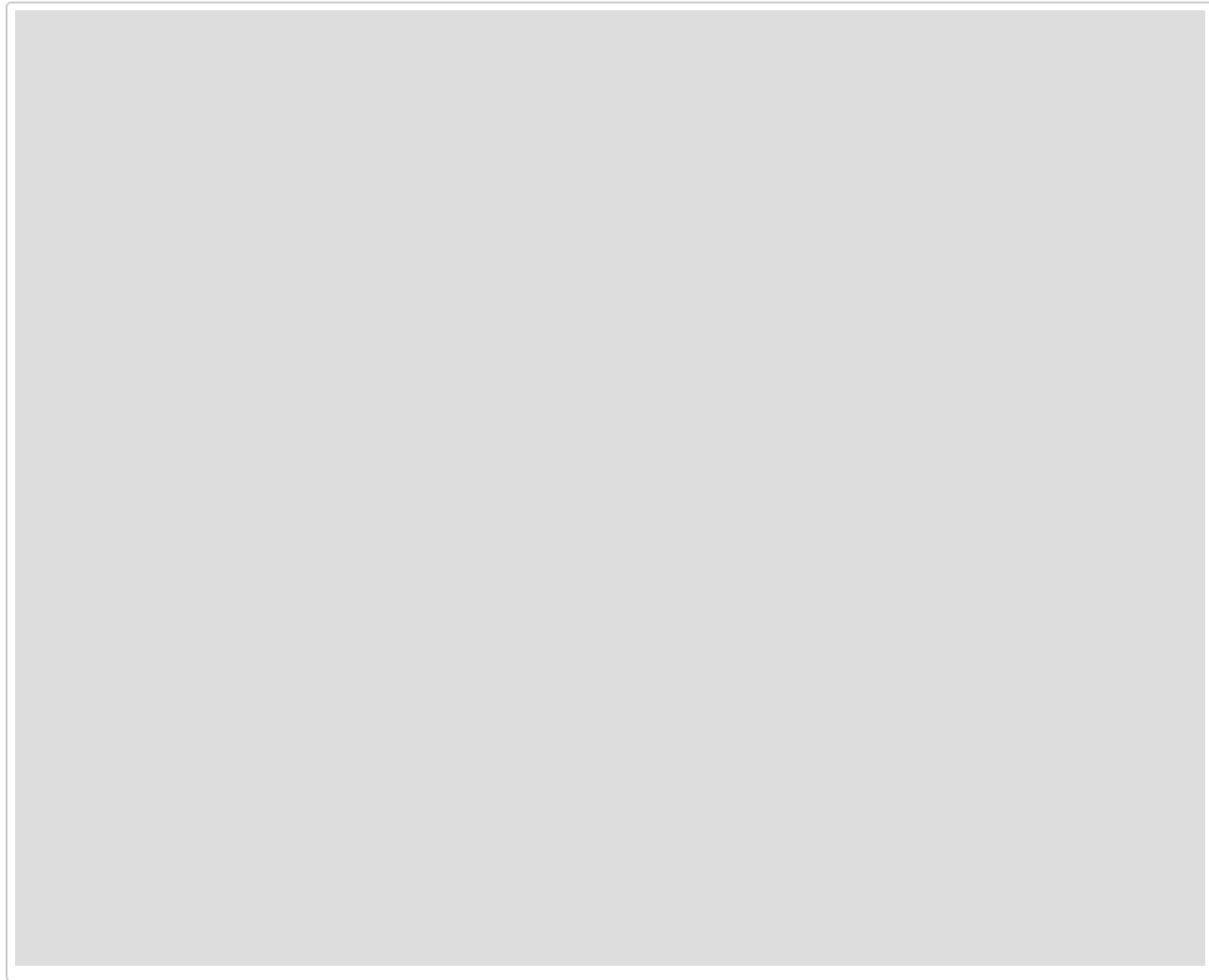
Version 3:



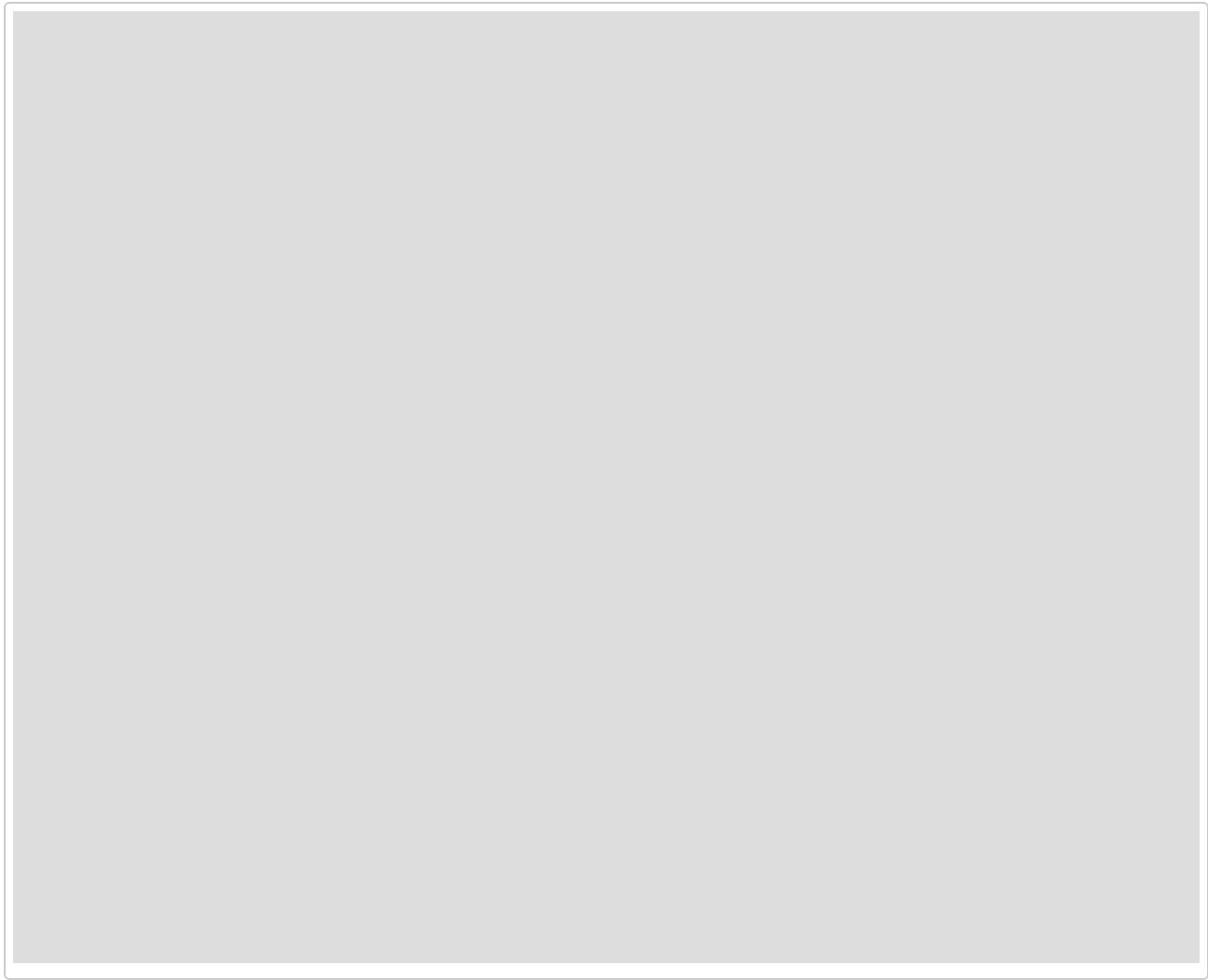
The XML file that can be edited at: <https://drive.google.com/file/d/1vSDeeuYOj3dJdAISYRhC8Ds202k8cz6L/view?usp=sharing>

Paper Prototype (Concept Design)

Logic and Activity Flow



Detailed Concept Design



Original Concept from Client

Digital Prototype

<https://w54t7s.axshare.com/#g=1&p=home>

 Unknown macro: 'html'

Personas

Users:

- Different clinics, patients, for free.
- The majority of the patients are in their 40s.

We followed these processes of creating personas. (Step1 to 5)

Based on the requirements of client, identified roles are **Patients, Families of patients, Clinics(doctors)**

Step1: Identify roles of interviewees

This is for [Patients mainly 40+]

Step2: Identify behavioural variables

- V1: Confident with technology
- V2: Medical knowledge
- V3: Learning ability
- V4: Willing to accept new things

Step3: Map interviewees to behavioural variables

V1	Very confident			Not confident at all
	R2	R1		
V2	Very confident			Not confident at all
	R2	R1		
V3	Very confident			Not confident at all
	R2	R1		
V4	Very confident			Not confident at all
	R2		R1	

Step4: Identify significant behaviour patterns

- Having trouble to learn how to use the new technologies
- Hard to understand the professional medical term
- Concerning about leak of private information

Step5: Synthesise characteristic and define goals

Candidate persona R1

- George Thomas
- He is a 57-year-old male.
- He is not married and lives with his dog, Mike.
- He finished senior high school.
- He was a truck driver and he is retired now. He has driven for 40 years.
- Although he has a phone with him, he doesn't know how to use it except answering phone calls.
- He prefers staying at home, watching TV and smoking. He doesn't like communicating with neighbours.
- He was diagnosed with lung cancer since he smoked a lot and lack of exercise.
- Goals
 - Experience: "I want to communicate with my attending doctor at any time".
 - End: "I can know my situation at any time".
 - Life: "My illness is well controlled and it does not affect my life".

Candidate persona R2

- Lucy Smith
- She is a 46-year-old lady and works as a math teacher leader in a senior high school.
- She is married and she has three children. Her husband is always very busy. She needs to take care of both her children and her students every day.
- She may have temporary work and urgent meeting at school.
- She is detected as breast cancer since she kept working under high pressure.
- She uses mobile phone on a daily basis for communicating with students' parents and checking emails.

- Goals

- Experience: "I can check my treatment records anytime anywhere".
 - End: "I can schedule the appointments more flexible".
 - Life: "I can balance my job and life".
-
-
-
-
-

Step1: Identify roles of interviewees

This is for [Family of patient]

Step2: Identify behavioural variables

- V1: Confident with technology
- V2: Medical knowledge
- V3: Learning ability
- V4: Willing to accept new things

Step3: Map interviewees to behavioural variables

V1	Very confident			Not confident at all
	R3			R4
V2	Very confident			Not confident at all
		R3 R4		
V3	Very confident			Not confident at all
	R3		R4	
V4	Very confident			Not confident at all
	R3			R4

Step4: Identify significant behaviour patterns

- Having trouble to learn how to use the new technologies
- Hard to understand the professional medical term
- Concerning about leak of private information
- Concerning about whether the treatment information is up to date

Step5: Synthesise characteristic and define goals

Candidate persona R3

- Alice Han
- She is a 22-year-old girl who is currently studying at medical collage. Her major is clinical medicine. She studies far away from her home.
- She shares her life on social medias very often by using mobile phones.
- Her parents were divorced when she was 10. She lived with her father and she loves her father a lot. She comes back home at every holiday to company with her father.
- Her father suffers liver cancer for one year. He needs to contact with his attendant doctor monthly.

- Goals

- Experience: "I can check my father's treatment records anytime anywhere".
- End: "I can communicate with doctor and I can make appointment monthly".
- Life: "I can know my father's condition better".

Candidate persona R4

- Stan Grant
- He is 64 year-old fiction writer. He always works at home.
- He is not familiar with computer and he can only write fiction by hand.

- His wife suffers Alzheimer for 3 years. She needs to go to hospital for physical review twice a week.
- He has the basic knowledge about Alzheimer.
- He is a very traditional person, and his daily life is take care of his wife and read newspapers. He doesn't like to use mobile phone.

- Goals

- Experience: "I hope doctors can come to my home for have a detection". "I want a more efficient and convenient way for appointment making".
 - End: "I can communicate with my wife's doctor frequently".
 - Life: "I can comprehensively look after my wife".
-
-
-
-
-

Step1: Identify roles of interviewees

This is for [Clinics(doctors)]

Step2: Identify behavioural variables

- V1: Confident with technology
- V2: Medical knowledge
- V3: Learning ability
- V4: Willing to accept new things

Step3: Map interviewees to behavioural variables

V1	Very confident			Not confident at all
	R5			
V2	Very confident			Not confident at all
	R5			
V3	Very confident			Not confident at all
	R5			
V4	Very confident			Not confident at all
	R5			

Step4: Identify significant behaviour patterns

- Concerning about whether has a efficient communication with patients.

Step5: Synthesise characteristic and define goals

Candidate persona R5

- Steph Allan
- He is a male, 33 years old.
- He is a professional cardiologist. He is very busy and he has many meetings and appointments every day.
- He may confused about the schedule of his appointments and meetings.
- Goals
 - Experience: "I want to organise my work more reasonably".
 - End: "I want combine my work (appointments and meetings) with calendar".
 - Life: "I hope my work be more orderly".

Product Backlog

Product Backlog identify the scope of the project (what should be delivered). At the beginning of each sprint, some items in product backlog should be moved to sprint backlog and continuously accomplishing all the backlog through the sprints.

Epic Functions

Epic Functions demonstrate the big picture of the whole project. It shows the high level functionalities of the product and was generated based on the goal model.

Epic ID	Epic User Stories
Account	Manage account/profile
Document	Manage documents
Form	Deal with forms
Appointment	Deal with appointments

Feature level

The functions in this level is more detailed than Epic level functions. Items in this level user are decomposed from Epic functions.

Feature ID	Related Epic ID	Feature Level User Stories
F1	Account	Log on and log off
F2	Account	Edit personal profile
F3	Account	Reset password
F4	Account	Set accounts for patients
F5	Document	Show documents
F6	Document	Receive documents
F7	Document	Search documents
F8	Document	Read documents
F9	Document	Print documents
F10	Form	View forms
F11	Form	Fill forms
F12	Form	Return forms
F13	Appointment	Check appointments
F14	Appointment	Attach notes
F15	Appointment	Share appointments
F17	Appointment	Edit appointments

User Stories

The product backlog will organise user stories with the Epics below. Furthermore, there are columns named "Time" representing the efforts to achieve the tasks (1 easiest and 5 most difficult) as well as "Priority" representing the priority of the tasks (1 lowest and 5 highest).

Following is the definition of the Time and Priority

Time	Priority
------	----------

1	The task will not cost any effort at all	1	The system can be completed without the tasks.
2	The task will cost some effort but not much	2	The tasks are not necessary for the system but nice to have.
3	The task will cost much effort	3	Necessary tasks of the system.
4	The task will cost much more effort but not a huge amount	4	Critical tasks of the system.
5	The task will cost huge amount of effort	5	Fundamentals of the system.

Following is the product backlog of the system

User Story ID	Related Feature	User Story	Tasks breakdown			Priority (1~5)
			ID	Description		
U 1.1	F3:Reset password	As a patient, I want to reset my password, so that I can find my account back when I forget my password.	1	The patient can reset the password	3	3
U 1.2	F1: Log on and log off	As a patient, I want to log on and log out my account, so that I can access my own account information.	1	The patient can log on the system	2	
			2	The patient can log off the system	2	5
U 1.3	F4: Set accounts for patients	As a secretary, I want to be able to set up accounts for patients, so that they can access the application.	1	The clinic can set account for new patients	3	5
U 2.1	F8: Read documents	As a new patient, I want to see all the instruction documents in my app, so that I can follow the instructions when I am in hospital.	1	Apply document reader in the app	3	
			2	Display the documents in the reader when open it in app	4	5
U 2.2	F7: Search documents	As a patient, I want to search the instruction documents by title, so that I can find what I want quickly.	1	Create interface to show all saved documents in app	1	
			2	Create search box to search the document name by app	4	3
			3	Filer out the corresponding document from all saved documents	3	
U 2.3	F6: Receive documents	As a patient, I want to receive new document when I have one, so that I can know the newest information.	1	New document is saved in patients database	5	
			2	Send the documents to the patient's app through the server	3	3
			3	The secretary sends the reminder to inform patients there is a new document through the app	4	
U 2.4	F5: Show documents	As a patient, I want to see all the instruction documents arranged in a simple and clear structure, so that I can easily know what documents I have and find what I need.	1	Create the database interface to access the documents	4	2
			2	Display the documents name list or thumbnails in the app	3	
U 2.5	F9: Read documents	As a patient, I want to view all the instruction documents in the app on my phone/pad, so that I do not need to print them.	1	Show the instructions (e.g. the procedure and information of treatment) in documents	2	5
U 2.6	F9: Print documents	As a patient, I want to print documents in the app, so that I can read them on paper.	1	Implement UI button to selected multiple documents	3	1
			2	Implement UI button to print selected documents	3	
			3	Implement UI to select printer and sent documents to the printer	4	
U 3.1	F10: View forms	As a patient, I want to access to the forms that I have already filled, so that I can review my historical forms.	1	create the database interface to access the historical forms of a patient	3	2
			2	desplay the historical forms for patients	1	
U 3.2	F11: Fill forms	As a patient, I want to receive the reminders for the form that I need to fill, so that I can fill it in time.	1	send email to inform patients to fill in the form through email server	1	3
			2	add forms need to filled as attachments	2	
U 3.3	F11: Fill forms	As a patient, I want to access to the templates of forms and see the instruction of filling forms, so that I can follow it to fill the forms correctly.	1	create the database interface to access the form template. Then show the templates with instructions (like date formats).	2	2

U 3.4	F11: Fill forms	As a patient, I want the app automatically enters the basic personal information when I am filling a form, so that I just need to finish the critical part.	1	wirte basic info(like patient's gender, hospital address) to the revelant placeholder automatically	3	1
U 3.5	F11: Fill forms	As a patient, I want to use my electrical signature or use tick box when my agreement is needed, so that I can do it quickly and conveniently.	1	save the signatures of patients to database	1	4
U 3.6	F12: Return forms	As a patient, I want to send the filled forms back through the email, so that I can contact with my doctor.	1	use js to check if the form has been filled compeleted with the required format	3	4
			2	If not completed or with incorrect info, use ajax give feedback to patients.	2	
			3	If all fine, send the forms as the attachments of email through email server	2	
U 4.1	F13: Check appointments	As a patient, I want to confirm my new appointment in the app so that the clinic know that I remember this appointment.	1	Create interface to show a new appointment	4	5
			2	Privede confirm option coming with unconfirmed appointments.	3	5
			3	If a appointment is confirmed by patient, send confirmation to the clinic.	2	5
U 4.2	F13: Check appointments	As a patient, I want to check all my appointment list by terms, such as time and types, so that I can know when is the appointment and what is it used for.	1.	Implement UI to show the list of existing appointments	3	5
			2.	Create sorted buttons, so the appointments can be sorted by time or types.	2	3
U 4.3	F13: Check appointments	As a patient, I want to know the detail of my appointment, such as date, location and type so that I won't forget and know where to go.	1.	Display the content of a appointment with specific format.	3	5
U 4.4	F13: Check appointments	As a patient, I want to know the time of all the appointments so that I can schedule my time table.	1.	Create a calender in the app with the time and other basic informations of all appointments.	3	3
U 4.5	F13: Check appointments	As a patient, I want to view the appointment instructions so that I can prepare for it.	1.	Display the instructions of a appointment.	2	4
U 4.6	F14: Attach notes	As a patient, I want to attach some notes on the application so that it can remind me the important things.	1.	Create interface of editable notes in application.	3	2
			2.	Save the notes in the mobile phone through the app.	2	2
U 4.7	F13: Check appointments	As a patient, I want the application can give me notification before my next appointment so that I won't forget it.	1.	When opening the app at the first time, the app will ask permission of notification on the mobile phone.	2	5
			2.	Create interface for alert setting (2 hours before, 1 day before, etc.) of appointment.	2	2
			3.	Display the notification before next appointment according to the alert setting.	3	5
U 4.8	F13: Check appointments	As a patient, I want to know the right number to call on the appointment detail page so that I can reschedule or cancel the appointment.	1.	Display the contact number of a specific appointment on the appointment page.	1	5
U 4.9	F15: Share appointments	As a patient, I want to share my appointment schedule to people such as family members, so that they can know my time table.	1.	Provide sharing option on the appointment page.	1	4
			2.	The content of appointment can be shared by email, SMS or other third-party apps.	3	4
U 4.10	F15: Share appointments	As a patient, I want to export the appointment detail into documents,such as PDF or RTF file so that I can read them on another device or send them to a helper.	1.	Provide exporting option on the appointment page.	1	4
			2.	The content of appointment can be exported into multiple files in different formats.	2	4
			3.	The exported file can be saved on mobile phone or sent by email.	2	4
U 4.11	F13: Check appointments	As a patient, I want to get notification immediately when the clinics change or cancel my appointment so that I can reschedule my plan.	1.	The changes of appointments from clinics can be synchronized in the app.	3	5
			2.	Once any change is completed, the people involved will get notification in the app.	2	5
U 4.12	F13: Check appointments	As a secretary, I want to receive patient confirmations so that I know if a patient might have forgotten an appointment.	1.	The secretary can receive notification through the app, once a patient confirms the appointment.	2	5
U 4.13	F17: Edit appointments	As a secretary, I want to associate information with appointment locations so patients will be able to attend them more easily.	1.	The secretary can edit appointment locations through the app.	3	4
			2.	The secretary can offer help to patients if they will request through the app.	3	4
U 4.14	F17: Edit appointments	As a secretary, I want to associate instructions with appointment types so patients will remember what to do beforehand.	1.	The secretary can access the content of appointment.	3	4

			2.	The secretary can prepare instructions and remind patients of what to prepare before appointment through the app.	3	4
--	--	--	----	---	---	---

User Stories

ID	User Stories	Story Point
1.Account		
U1.1	As a patient, I want to reset my password, so that I can find my account back when I forget my password.	2
U1.2	As a patient, I want to log in and log out my account, so that I can access my own account information.	3
U1.3	As a secretary, I want to be able to set up accounts for patients, so that they can access the application.	8
2. Documents		
U2.1	As a new patient, I want to see all the instruction documents in my app, so that I can follow the instructions when I am in hospital.	5
U2.2	As a patient, I want to search the instruction documents by title, so that I can find what I want quickly.	3
U2.3	As a patient, I want to receive new document when I have one, so that I can know the latest information.	3
U2.4	As a patient, I want to see all the instruction documents arranged in a simple and clear structure, so that I can easily know what documents I have and find what I need.	2
U2.5	As a patient, I want to view all the information documents in the app on my phone/pad, so that I do not need to print them.	5
U2.6	As a patient, I want to print documents in the app, so that I can read them on paper.	1
3. Forms		
U3.1	As a patient, I want to access to the templates of forms and the instructions of filling forms, so that I can follow it to fill the forms correctly.	5
U3.2	As a patient, I want to receive the reminders for the form that I need to fill, so that I can fill it in time.	2
U3.3	As a patient, I want to access to the forms which I have already filled, so that I can review my historical forms.	2
U3.4	As a patient, I want the app automatically enters the basic personal information when I am filling a form, so that I just need to finish the critical part.	ASK CLIENT
U3.5	As a patient, I want to use my electronic signature or use tick box when my agreement is needed, so that I can do it quickly and conveniently.	3
U3.6	As a patient, I want to be able to fill in forms through the app, so that I don't need to print them	8
U3.7	As a patient, I want to send the filled forms back through the email automatically, so that I don't need to send it manually.	2
4. Appointments		
U4.1	As a patient, I want to confirm my new appointment in the app so that the clinic know that I remember this appointment.	3
U4.2	As a patient, I want to check all my appointment list by terms, such as time and types, so that I can know when is the appointment and what is it used for.	3
U4.3	As a patient, I want to know the detail of my appointment, such as date, location and type so that I won't forget and know where to go.	2
U4.4	As a patient, I want to know the time of all the appointments so that I can schedule my time table.	2
U4.5	As a patient, I want to view the appointment instructions so that I can prepare for it.	2
U4.6	As a patient, I want to attach some notes on the application so that it can remind me the important things.	3
U4.7	As a patient, I want the application to give me notification before my next appointment so that I won't forget it.	3
U4.8	As a patient, I want to know the right number to call on the appointment detail page so that I can reschedule or cancel the appointment.	1

U4.9	As a patient, I want to share my appointment schedule to people such as family members, so that they can know my time table.	2
U4.10	As a patient, I want to share the appointment detail documents, so that I can read them on another device or send them to a helper.	2
U4.11	As a patient, I want to get notification immediately when the clinics change or cancel my appointment so that I can reschedule my plan.	ASK CLIENT
U4.12	As a secretary, I want to receive patient confirmations about their appointments so that I know if a patient might have forgotten an appointment.	3
U4.13	As a secretary, I want to associate information with appointment locations so patients will be able to attend them more easily.	5
U4.14	As a secretary, I want to associate instructions with appointment types so patients will remember what to do beforehand.	5

API References

Please see current API documents in Bitbucket repo.

Here is the link: [API Documentation](#)

Deployment

Back-End Deployment

Medical Secretary Back-End Deployment Guide

Prepare the Server Environment

The back-end system requires a server with a dedicated underlying environment. This includes a webserver, a Java Servlet Container and a Database server. The easiest way to setup the environment is to deploy a "LAMP" stack with a Tomcat server.

Introduction

A "LAMP" stack is a group of open-source software that is typically installed together to enable a server to host dynamic websites and web apps. This term is actually an acronym which represents the **L**inux operating system, with the **A**pache web server. The site data is stored in a **M**ySQL database, and dynamic content is processed by **P**HP.

Apache Tomcat is a web server and servlet container that is used to serve Java applications. Tomcat is an open source implementation of the Java Servlet and JavaServer Pages technologies, released by the Apache Software Foundation. This guide covers the basic installation and some configuration of the latest release of Tomcat 9 on your Ubuntu server.

Prerequisites

In order to complete this guide, you will need to have an Debian Linux (e.g. Ubuntu) server with a non-root sudo-enabled user account and a basic firewall.

For macOS servers, use brew (<https://brew.sh>) instead of apt commands

Step 1 — Installing Apache and Updating the Firewall

The Apache web server is among the most popular web servers in the world. It's well-documented and has been in wide use for much of the history of the web, which makes it a great default choice for hosting a website.

Install Apache using Ubuntu's package manager, apt:

```
$ sudo apt update  
$ sudo apt install apache2
```

Since this is a sudo command, these operations are executed with root privileges. It will ask you for your regular user's password to verify your intentions.

Once you've entered your password, apt will tell you which packages it plans to install and how much extra disk space they'll take up. Press Y and hit ENTER to continue, and the installation will proceed.

Adjust the Firewall to Allow Web Traffic

Note: you may skip this section if you are setting up a macOS server as macOS uses a different firewall, refer to [this guide from Apple](#)

Next, assuming that you have followed the initial server setup instructions and enabled the UFW firewall, make sure that your firewall allows HTTP and HTTPS traffic. You can check that UFW has an application profile for Apache like so:

```
$ sudo ufw app list
```

Output:

Available applications:

```
Apache  
Apache Full  
Apache Secure  
OpenSSH
```

If you look at the Apache Full profile, it should show that it enables traffic to ports 80 and 443:

```
$ sudo ufw app info "Apache Full"
```

Output:

```
Profile: Apache Full  
Title: Web Server (HTTP,HTTPS)  
Description: Apache v2 is the next generation of the omnipresent Apache web server.
```

```
Ports:  
      80,443/tcp
```

Allow incoming HTTP and HTTPS traffic for this profile:

```
sudo ufw allow in "Apache Full"
```

You can do a spot check right away to verify that everything went as planned by visiting your server's public IP address in your web browser (see the note under the next heading to find out what your public IP address is if you do not have this information already):

```
http://your_server_ip
```

You will see the default Ubuntu Apache web page, which is there for informational and testing purposes. It should look something like this:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
    '-- ports.conf
|-- mods-enabled
    |-- *.load
    '-- *.conf
|-- conf-enabled
    '-- *.conf
|-- sites-enabled
    '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

If you see this page, then your web server is now correctly installed and accessible through your firewall.

How To Find your Server's Public IP Address

If you do not know what your server's public IP address is, there are a number of ways you can find it. Usually, this is the address you use to connect to your server through SSH.

There are a few different ways to do this from the command line. First, you could use the `iproute2` tools to get your IP address by typing this:

```
$ ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\.*$//'
```

This will give you two or three lines back. They are all correct addresses, but your computer may only be able to use one of them, so feel free to try each one.

An alternative method is to use the `curl` utility to contact an outside party to tell you how it sees your server. This is done by asking a specific server what your IP address is:

```
$ sudo apt install curl  
$ curl http://icanhazip.com
```

Regardless of the method you use to get your IP address, type it into your web browser's address bar to view the default Apache page.

Step 2 — Installing MySQL

Now that you have your web server up and running, it is time to install MySQL. MySQL is a database management system. Basically, it will organize and provide access to databases where your site can store information.

Again, use `apt` to acquire and install this software:

```
$ sudo apt install mysql-server
```

This command, too, will show you a list of the packages that will be installed, along with the amount of disk space they'll take up. Enter `Y` to continue.

When the installation is complete, run a simple security script that comes pre-installed with MySQL which will remove some dangerous defaults and lock down access to your database system. Start the interactive script by running:

```
$ sudo mysql_secure_installation
```

This will ask if you want to configure the `VALIDATE PASSWORD PLUGIN`.

Note: Enabling this feature is something of a judgment call. If enabled, passwords which don't match the specified criteria will be rejected by MySQL with an error. This will cause issues if you use a weak password in conjunction with software which automatically configures MySQL user credentials, such as the Ubuntu packages for phpMyAdmin. It is safe to leave validation disabled, but you should always use strong, unique passwords for database credentials.

Answer `Y` for yes, or anything else to continue without enabling.

`VALIDATE PASSWORD PLUGIN` can be used to test passwords and improve security. It checks the strength of password and allows the users to set only those passwords which are secure enough. Would you like to setup `VALIDATE PASSWORD plugin`?

Press `y|Y` for Yes, any other key for No:

If you answer "yes", you'll be asked to select a level of password validation. Keep in mind that if you enter 2 for the strongest level, you will receive errors when attempting to set any password which does not contain numbers, upper and lowercase letters, and special characters, or which is based on common dictionary words.

There are three levels of password validation policy:

```
LOW      Length >= 8  
MEDIUM   Length >= 8, numeric, mixed case, and special characters  
STRONG   Length >= 8, numeric, mixed case, special characters and dictionary file
```

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1

Regardless of whether you chose to set up the `VALIDATE PASSWORD PLUGIN`, your server will next ask you to select and confirm a password for the MySQL `root` user. This is an administrative account in MySQL that has increased privileges. Think of it as being similar to the `root` account for the server itself (although the one you are configuring now is a MySQL-specific account). Make sure this is a strong, unique password, and do not leave it blank.

If you enabled password validation, you'll be shown the password strength for the root password you just entered and your server will ask if you want to change that password. If you are happy with your current password, enter `N` for "no" at the prompt:

Using existing password for root.

```
Estimated strength of the password: 100  
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
```

For the rest of the questions, press **Y** and hit the **ENTER** key at each prompt. This will remove some anonymous users and the test database, disable remote root logins, and load these new rules so that MySQL immediately respects the changes you have made.

Note that in Ubuntu systems running MySQL 5.7 (and later versions), the **root** MySQL user is set to authenticate using the `auth_socket` plugin by default rather than with a password. This allows for some greater security and usability in many cases, but it can also complicate things when you need to allow an external program (e.g., phpMyAdmin) to access the user.

If you prefer to use a password when connecting to MySQL as **root**, you will need to switch its authentication method from `auth_socket` to `mysql_native_password`. To do this, open up the MySQL prompt from your terminal:

```
$ sudo mysql
```

Next, check which authentication method each of your MySQL user accounts use with the following command:

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

Output:

user	authentication_string	plugin	host
root	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	auth_socket	localhost
mysql.session	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
mysql.sys	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
debian-sys-maint	*CC744277A401A7D25BE1CA89AFF17BF607F876FF	mysql_native_password	localhost

4 rows in set (0.00 sec)

In this example, you can see that the **root** user does in fact authenticate using the `auth_socket` plugin. To configure the **root** account to authenticate with a password, run the following `ALTER USER` command. Be sure to change `password` to a strong password of your choosing:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
```

Then, run `FLUSH PRIVILEGES` which tells the server to reload the grant tables and put your new changes into effect:

```
mysql> FLUSH PRIVILEGES;
```

Check the authentication methods employed by each of your users again to confirm that **root** no longer authenticates using the `auth_socket` plugin:

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

Output:

user	authentication_string	plugin	host
root	*3636DACC8616D997782ADD0839F92C1571D6D78F	mysql_native_password	localhost
mysql.session	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
mysql.sys	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
debian-sys-maint	*CC744277A401A7D25BE1CA89AFF17BF607F876FF	mysql_native_password	localhost

4 rows in set (0.00 sec)

You can see in this example output that the **root** MySQL user now authenticates using a password. Once you confirm this on your own server, you can exit the MySQL shell:

```
mysql> exit
```

At this point, your database system is now set up and you can move on to installing PHP, the final component of the LAMP stack. In this guide we will skip this section as we use Tomcat server to handle the dynamic content rather than PHP.

Step 3—Installing Tomcat

Install Java

Tomcat requires Java to be installed on the server so that any Java web application code can be executed. We can satisfy that requirement by installing OpenJDK with apt.

```
$ sudo apt install default-jdk
```

Now that Java is installed, we can create a tomcat user, which will be used to run the Tomcat service.

Create Tomcat User

For security purposes, Tomcat should be run as an unprivileged user (i.e. not root). We will create a new user and group that will run the Tomcat service.

First, create a new `tomcat` group:

```
$ sudo groupadd tomcat
```

Next, create a new tomcat user. We'll make this user a member of the `tomcat` group, with a home directory of `/opt/tomcat` (where we will install Tomcat), and with a shell of `/bin/false` (so nobody can log into the account):

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

Now that our `tomcat` user is set up, let's download and install Tomcat.

Install Tomcat

The best way to install Tomcat 9 is to download the latest binary release then configure it manually.

Find the latest version of Tomcat 9 at the [Tomcat 9 Downloads page](#). At the time of writing, the latest version is **9.0.12**, but you should use a later stable version if it is available. Under the **Binary Distributions** section, then under the **Core** list, copy the link to the **tar.gz**.

Next, change to the `/tmp` directory on your server. This is a good directory to download ephemeral items, like the Tomcat tarball, which we won't need after extracting the Tomcat contents:

```
$ cd /tmp
```

Use `curl` to download the link that you copied from the Tomcat website:

```
$ curl -O http://mirror.cc.columbia.edu/pub/software/apache/tomcat/tomcat-9/v9.0.10/bin/apache-tomcat-9.0.10.tar.gz
```

We will install Tomcat to the `/opt/tomcat` directory. Create the directory, then extract the archive to it with these commands:

```
$ sudo mkdir /opt/tomcat  
$ sudo tar xzvf apache-tomcat-9*tar.gz -C /opt/tomcat --strip-components=1````
```

Next, we can set up the proper user permissions for our installation.

Update Permissions

The `tomcat` user that we set up needs to have access to the Tomcat installation. We'll set that up now.

Change to the directory where we unpacked the Tomcat installation:

```
$ cd /opt/tomcat
```

Give the `tomcat` group ownership over the entire installation directory:

```
$ sudo chgrp -R tomcat /opt/tomcat
```

Next, give the `tomcat` group read access to the `conf` directory and all of its contents, and **execute** access to the directory itself:

```
$ sudo chmod -R g+r conf  
$ sudo chmod g+x conf
```

Make the `tomcat` user the owner of the `webapps`, `work`, `temp`, and `logs` directories:

```
$ sudo chown -R tomcat webapps/ work/ temp/ logs/
```

Now that the proper permissions are set up, we can create a systemd service file to manage the Tomcat process.

Create a systemd Service File

We want to be able to run Tomcat as a service, so we will set up systemd service file.

Tomcat needs to know where Java is installed. This path is commonly referred to as "JAVA_HOME". The easiest way to look up that location is by running this command:

```
$ sudo update-java-alternatives -l
```

Output:

```
java-1.11.0-openjdk-amd64      1081      /usr/lib/jvm/java-1.11.0-openjdk-amd64
```

Your `JAVA_HOME` is the output from the last column (highlighted in red). Given the example above, the correct `JAVA_HOME` for this server would be:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64
```

Your `JAVA_HOME` may be different.

With this piece of information, we can create the systemd service file. Open a file called `tomcat.service` in the `/etc/systemd/system` directory by typing:

```
$ sudo nano /etc/systemd/system/tomcat.service
```

Paste the following contents into your service file. Modify the value of `JAVA_HOME` if necessary to match the value you found on your system. You may also want to modify the memory allocation settings that are specified in `CATALINA_OPTS`:

```

[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/.urandom'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target

```

When you are finished, save and close the file.

Next, reload the systemd daemon so that it knows about our service file:

```
$ sudo systemctl daemon-reload
```

Start the Tomcat service by typing:

```
$ sudo systemctl start tomcat
```

Double check that it started without errors by typing:

```
$ sudo systemctl status tomcat
```

Adjust the Firewall and Test the Tomcat Server

Now that the Tomcat service is started, we can test to make sure the default page is available.

Before we do that, we need to adjust the firewall to allow our requests to get to the service. If you followed the prerequisites, you will have a `ufw` firewall enabled currently.

Tomcat uses port 8080 to accept conventional requests. Allow traffic to that port by typing:

```
$ sudo ufw allow 8080
```

With the firewall modified, you can access the default splash page by going to your domain or IP address followed by :8080 in a web browser:

```
http://server_domain_or_IP:8080
```

You will see the default Tomcat splash page, in addition to other information. However, if you click the links for the Manager App, for instance, you will be denied access. We can configure that access next.

If you were able to successfully accessed Tomcat, now is a good time to enable the service file so that Tomcat automatically starts at boot:

```
$ sudo systemctl enable tomcat
```

Configure Tomcat Web Management Interface

In order to use the manager web app that comes with Tomcat, we must add a login to our Tomcat server. We will do this by editing the `tomcat-users.xml` file:

```
$ sudo nano /opt/tomcat/conf/tomcat-users.xml
```

You will want to add a user who can access the `manager-gui` and `admin-gui` (web apps that come with Tomcat). You can do so by defining a user, similar to the example below, between the `tomcat-users` tags. Be sure to change the username and password to something secure:

```
<tomcat-users . . .>
  <user username="admin_username" password="your_password" roles="manager-gui,admin-gui"/>
</tomcat-users>
```

Save and close the file when you are finished.

By default, newer versions of Tomcat restrict access to the Manager and Host Manager apps to connections coming from the server itself. Since we are installing on a remote machine, you will probably want to remove or alter this restriction. To change the IP address restrictions on these, open the appropriate `context.xml` files.

For the Manager app, type:

```
$ sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

For the Host Manager app, type:

```
$ sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml
```

Inside, comment out the IP address restriction to allow connections from anywhere. Alternatively, if you would like to allow access only to connections coming from your own IP address, you can add your public IP address to the list:

```
<Context antiResourceLocking="false" privileged="true" >
  <!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127.\d+.\d+.\d+|:1|0:0:0:0:0:0:1" />-->
</Context>
```

Save and close the files when you are finished.

To put our changes into effect, restart the Tomcat service:

```
$ sudo systemctl restart tomcat
```

Access the Web Interface

Now that we have created a user, we can access the web management interface again in a web browser. Once again, you can get to the correct interface by entering your server's domain name or IP address followed on port 8080 in your browser:

```
http://server_domain_or_IP:8080
```

The page you see should be the same one you were given when you tested earlier:

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

Let's take a look at the Manager App, accessible via the link or `http://server_domain_or_IP:8080/manager/html`. You will need to enter the account credentials that you added to the `tomcat-users.xml` file. Afterwards, you should see a page that looks like this:

Tomcat Web Application Manager

Message: <input type="button" value="OK"/>					
Manager					
List Applications		HTML Manager Help	Manager Help	Server Status	
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	<i>None specified</i>	Welcome to Tomcat	true	0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>
/docs	<i>None specified</i>	Tomcat Documentation	true	0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>
/examples	<i>None specified</i>	Servlet and JSP Examples	true	0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>
/host-manager	<i>None specified</i>	Tomcat Host Manager Application	true	0	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>
/manager	<i>None specified</i>	Tomcat Manager Application	true	1	<input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>

Deploy					
Deploy directory or WAR file located on server					
Context Path (required): <input type="text"/> XML Configuration file URL: <input type="text"/> WAR or Directory URL: <input type="text"/> <input type="button" value="Deploy"/>					
WAR file to deploy					
Select WAR file to upload <input type="button" value="Choose File"/> No file chosen <input type="button" value="Deploy"/>					

The Web Application Manager is used to manage your Java applications. You can Start, Stop, Reload, Deploy, and Undeploy here. You can also run some diagnostics on your apps (i.e. find memory leaks). Lastly, information about your server is available at the very bottom of this page.

Now let's take a look at the Host Manager, accessible via the link or http://server_domain_or_IP:8080/host-manager/html/:

Tomcat Virtual Host Manager

Message:

Host Manager

List Virtual Hosts	HTML Host Manager Help (TODO)	Host Manager Help (TODO)	Server Status
--------------------	-------------------------------	--------------------------	---------------

Host name

Host name	Host aliases	Commands
localhost		Host Manager installed - commands disabled

Add Virtual Host

Host

Name: <input type="text"/>
Aliases: <input type="text"/>
App base: <input type="text"/>
AutoDeploy <input checked="" type="checkbox"/>
DeployOnStartup <input checked="" type="checkbox"/>
DeployXML <input checked="" type="checkbox"/>
UnpackWARs <input checked="" type="checkbox"/>
Manager App <input checked="" type="checkbox"/>
CopyXML <input type="checkbox"/>
<input type="button" value="Add"/>

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture
Apache Tomcat/8.0.33	1.8.0_03-Ubuntu-8u77-b03-3ubuntu3-b03	Oracle Corporation	Linux	4.4.0-21-generic	amd64

From the Virtual Host Manager page, you can add virtual hosts to serve your applications from.

Compile and Build from the Source

Note: if you have a pre-build artifact (a .war file), you can deploy it directly without build from the source. Jump to [Deploy to Tomcat](#)

Install Maven

Apache Maven is a free and open source project management tool used for Java projects. You can easily manage a project's build, reporting, and documentation from a central piece of information using Apache Maven. Apache Maven provides a complete framework to automate the project's build infrastructure.

Build of this project requires maven.

Install Apache Maven

You can download the latest stable version of Apache Maven from its official website, otherwise you can download it directly with the following command:

```
$ cd /opt/
$ wget http://www-eu.apache.org/dist/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz
```

Once the download has completed, extract the downloaded archive.

```
$ sudo tar -xvzf apache-maven-3.3.9-bin.tar.gz
```

Next, rename the extracted directory.

```
$ sudo mv apache-maven-3.3.9 maven
```

Setup environment variables

Next, you will need to setup the environment variables such as M2_HOME, M2, MAVEN_OPTS, and PATH. You can do this by creating a `mavenenv.sh` file inside of the `/etc/profile.d/` directory.

```
$ sudo nano /etc/profile.d/mavenenv.sh
```

Add the following lines:

```
export M2_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}
```

Save and close the file, update its permissions, then load the environment variables with the following command:

```
$ sudo chmod +x /etc/profile.d/mavenenv.sh  
$ sudo source /etc/profile.d/mavenenv.sh
```

Verify installation

Once everything has been successfully configured, check the version of the Apache Maven.

```
$ mvn --version
```

You should see the following output:

```
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T22:11:47+05:30)  
Maven home: /opt/maven  
Java version: 1.8.0_101, vendor: Oracle Corporation  
Java home: /usr/lib/jvm/java-8-oracle/jre  
Default locale: en_US, platform encoding: ANSI_X3.4-1968  
OS name: "linux", version: "3.13.0-32-generic", arch: "amd64", family: "unix"
```

Now retrieve the source code

Say you have downloaded the project, had it unpacked and navigated into that folder.

Configuration

First you need to configurate the project for your real environment

Open the `env_setup.sh` to prepare the configuration:

```
$ nano env_setup.sh
```

Locate the Init parameters section, it should looks like this:

```
# Init parameters  
URL="jdbc:mysql://frank.czb2g0pebcye.ap-southeast-2.rds.amazonaws.com:3306/medsec?serverTimezone=UTC"  
USERNAME="$1"  
PASSWORD="$2"  
VERSION_STRING="Automatically built and deployed. Build date: `date`"
```

Replace the sample URL, USERNAME, and PASSWORD parameters of database with real ones that correspond to your environment:

```
# Init parameters  
URL="jdbc:mysql://ip_or_address_to_your_database?serverTimezone=UTC"  
USERNAME="your_database_username"  
PASSWORD="your_database_password"  
VERSION_STRING="Automatically built and deployed. Build date: `date`"
```

Save and exit.

Now add execute privallage to this script and execute it:

```
$ sudo chmod +x env_setup.sh  
$ sh env_setup.sh
```

Build with Maven

Build and run integration tests as follows:

```
$ mvn clean install
```

This will compile the project and pack all the packages. The built artifact `medsec.war` will be generated in `target` directory.

Initialise the Database

For the first time, you need to import the pre-defined database to initialise the database.

```
$ mysql -u <username> -p<password> < sql/db_with_data.sql
```

Deploy to Tomcat

Go to your tomcat home directory. Remove previous deployed application in `webapps`:

```
$ rm webapps/medsec.war  
$ rm -rf webapps/medsec
```

Clear tomcat runtime temporary files:

```
$ rm -rf work/Catalina
```

Then copy the built artifact medsec.war to webapps directory.

Start your tomcat:

```
$ bin/catalina.sh run
```

Access the API at:

```
http://localhost:8080/medsec
```

Front End Deployment

Deployment of Mobile Application

Provide by Project Medical Secretary - Front End Team

Google Doc Link: https://docs.google.com/document/d/1TMyBVAOUlFZ7hNx3ga_jfyLdmvFWfZA2b5hZfH4yceQ/edit?usp=sharing

- Yuanchenxi Liu
- Yueyan Chen
- Jiacong Ling
- Chuan Yang

Description:

This document is about how to setup environment of development and release the mobile application based on the environment.

Setup environment

Environment Information in total:

For IOS:

Software	Version
react-native cli	2.0.1
react-native	0.55.4
npm	5.6.0
node	v10.1.0
Xcode	v9.4.1

For Android:

Software	Version
react-native cli	2.0.1
react-native	0.55.4
JRE/JDK	1.8.0
npm	5.6.0
node	v10.1.0
Android Studio	3.1.2
Android Virtual	Oreo 8.1
Python	2.x

Please make sure the softwares above are in the same or higher version.

Windows:

1. Install node, python2, JDK:

Open an Administrator Command Prompt (right click Command Prompt and select "Run as Administrator"), then run the following command:

```
choco install -y nodejs.install python2 jdk8
```

2: Install React native CLI:

Run the following command in a Command Prompt or shell:

```
npm install -g react-native-cli
```

If you get an error like *Cannot find module 'npmlog'*, try installing npm directly: curl -O -L <https://npmjs.org/install.sh> / sudo sh.

3. Install Android Studio:

Download and install [Android Studio](#). Choose a "Custom" setup when prompted to select an installation type. Make sure the boxes next to all of the following are checked:

- Android SDK
- Android SDK Platform
- Performance (Intel ® HAXM)
- Android Virtual Device

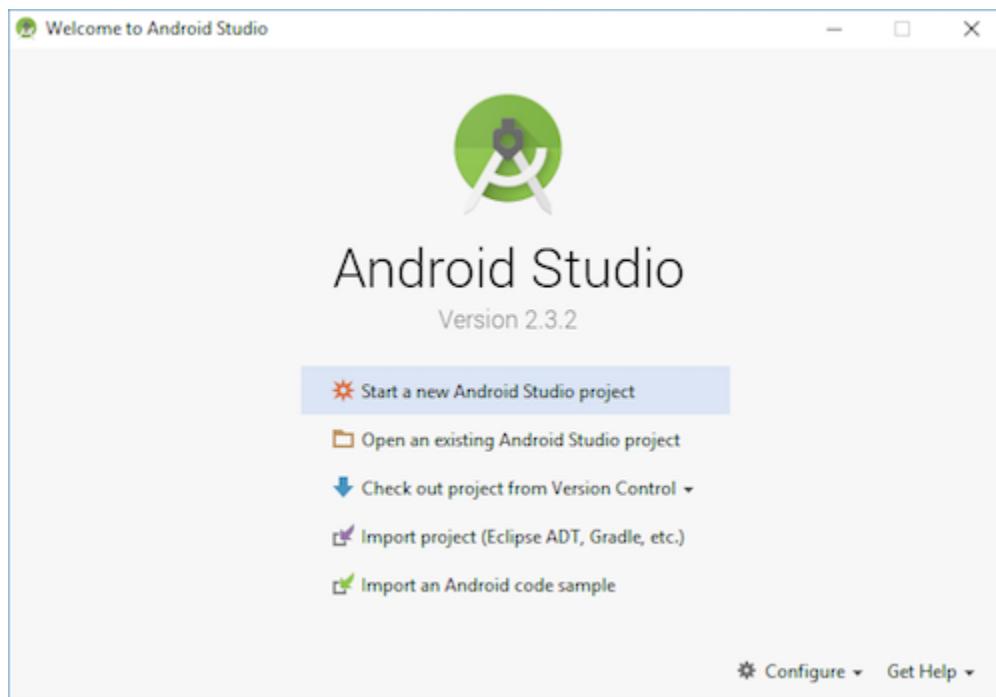
Then, click "Next" to install all of these components.

If the checkboxes are grayed out, you will have a chance to install these components later on.

4. Install Android SDK:

Android Studio installs the latest Android SDK by default. Building a React Native app with native code, however, requires the Android 8.0 (Oreo) SD K in particular. Additional Android SDKs can be installed through the SDK Manager in Android Studio.

The SDK Manager can be accessed from the "Welcome to Android Studio" screen. Click on "Configure", then select "SDK Manager".



The SDK Manager can also be found within the Android Studio "Preferences" dialog, under Appearance & Behavior System Settings Android SDK.

Select the "SDK Platforms" tab from within the SDK Manager, then check the box next to "Show Package Details" in the bottom right corner. Look for and expand the Android 8.0 (Oreo) entry, then make sure the following items are all checked:

- Android SDK Platform 26
- Google APIs Intel x86 Atom_64 System Image

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Android TV Intel x86 Atom System Image	27	4	Not installed
<input type="checkbox"/>	Google APIs Intel x86 Atom System Image	27	6	Not installed
<input type="checkbox"/>	Google Play Intel x86 Atom System Image	27	3	Not installed
▼ <input checked="" type="checkbox"/>	Android 8.0 (Oreo)			
<input checked="" type="checkbox"/>	Android SDK Platform 26	26	2	Installed
<input type="checkbox"/>	Sources for Android 26	26	1	Not installed
<input type="checkbox"/>	Android TV Intel x86 Atom System Image	26	9	Not installed
<input type="checkbox"/>	China version of Android Wear Intel x86 Atom System Image	26	4	Not installed
<input type="checkbox"/>	Android Wear Intel x86 Atom System Image	26	4	Not installed
<input type="checkbox"/>	Google APIs Intel x86 Atom System Image	26	10	Not installed
<input checked="" type="checkbox"/>	Google APIs Intel x86 Atom_64 System Image	26	10	Installed
<input type="checkbox"/>	Google Play Intel x86 Atom System Image	26	7	Not installed
▼ <input type="checkbox"/>	Android 7.1.1 (Nougat)			
<input type="checkbox"/>	Android SDK Platform 25	25	3	Not installed
<input type="checkbox"/>	Sources for Android 25	25	1	Not installed
<input type="checkbox"/>	Android TV Intel x86 Atom System Image	25	11	Not installed
<input type="checkbox"/>	China version of Android Wear ARM EABI v7a System Image	25	4	Not installed
<input type="checkbox"/>	China version of Android Wear Intel x86 Atom System Image	25	4	Not installed
<input type="checkbox"/>	Android Wear ARM EABI v7a System Image	25	3	Not installed

Show Package Details

Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the "Android SDK Build-Tools" entry, then make sure that 26.0.3 is selected.

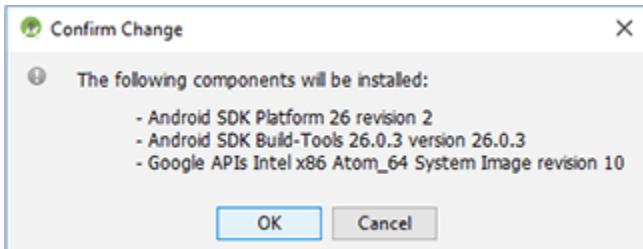
SDK Platforms **SDK Tools** SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

	Name	Version	Status
<input type="checkbox"/>	22.0.1	22.0.1	Not installed
<input type="checkbox"/>	23.0.0 (Obsolete)	23.0.0	Not installed
<input type="checkbox"/>	23.0.1	23.0.1	Not installed
<input type="checkbox"/>	23.0.2	23.0.2	Not installed
<input type="checkbox"/>	23.0.3	23.0.3	Not installed
<input type="checkbox"/>	24.0.0	24.0.0	Not installed
<input type="checkbox"/>	24.0.1	24.0.1	Not installed
<input type="checkbox"/>	24.0.2	24.0.2	Not installed
<input type="checkbox"/>	24.0.3	24.0.3	Not installed
<input type="checkbox"/>	25.0.0	25.0.0	Not installed
<input type="checkbox"/>	25.0.1	25.0.1	Not installed
<input type="checkbox"/>	25.0.2	25.0.2	Not installed
<input type="checkbox"/>	25.0.3	25.0.3	Not installed
<input type="checkbox"/>	26.0.0	26.0.0	Not installed
<input type="checkbox"/>	26.0.1	26.0.1	Not installed
<input type="checkbox"/>	26.0.2	26.0.2	Not installed
<input checked="" type="checkbox"/>	26.0.3	26.0.3	Installed
<input type="checkbox"/>	27.0.0	27.0.0	Not installed
<input type="checkbox"/>	27.0.1	27.0.1	Not installed
<input type="checkbox"/>	27.0.2	27.0.2	Not installed

Show Package Details

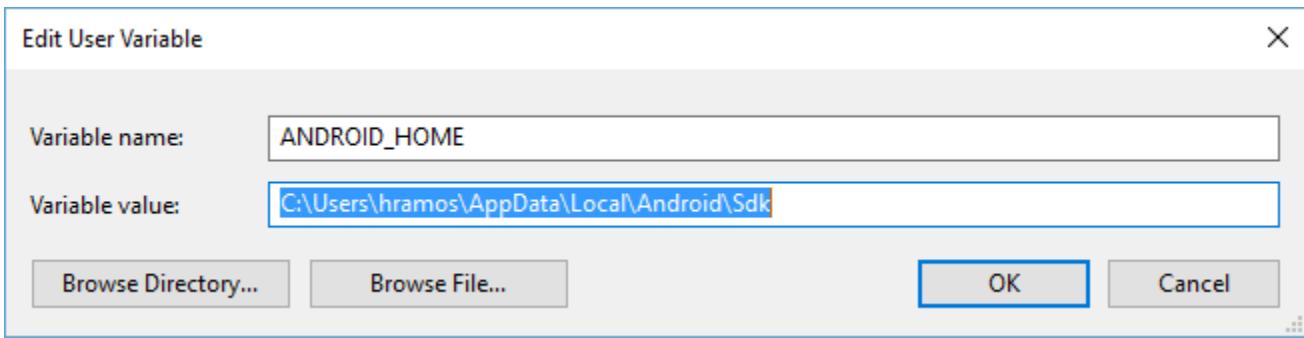
Finally, click "Apply" to download and install the Android SDK and related build tools.



5. Configure the ANDROID_HOME environment variable

The React Native tools require some environment variables to be set up in order to build apps with native code.

Open the System pane under System and Security in the Windows Control Panel, then click on Change settings.... Open the Advanced tab and click on Environment Variables.... Click on New... to create a new ANDROID_HOME user variable that points to the path to your Android SDK:



The SDK is installed, by default, at the following location:

```
c:\Users\YOUR_USERNAME\AppData\Local\Android\Sdk
```

You can find the actual location of the SDK in the Android Studio "Preferences" dialog, under Appearance & Behavior System Settings Android SDK.

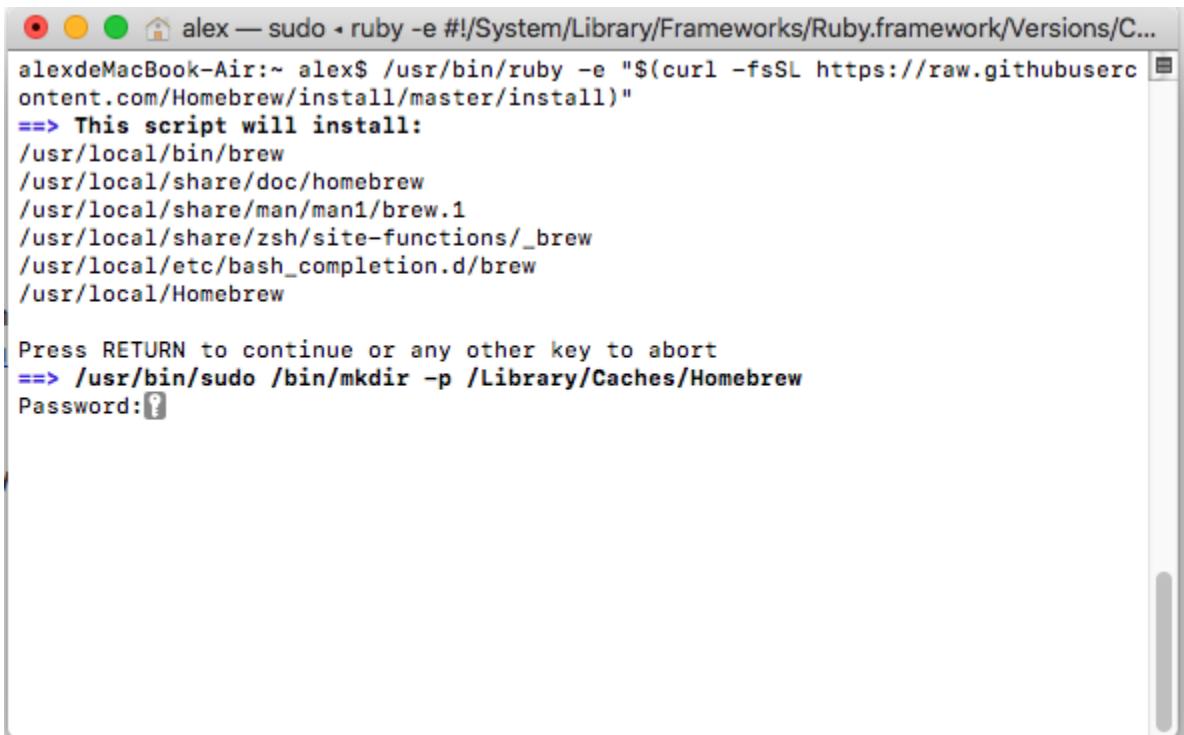
Open a new Command Prompt window to ensure the new environment variable is loaded before proceeding to the next step.

Mac:

1. Setup brew:

Open terminal on Mac and enter the command below.

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

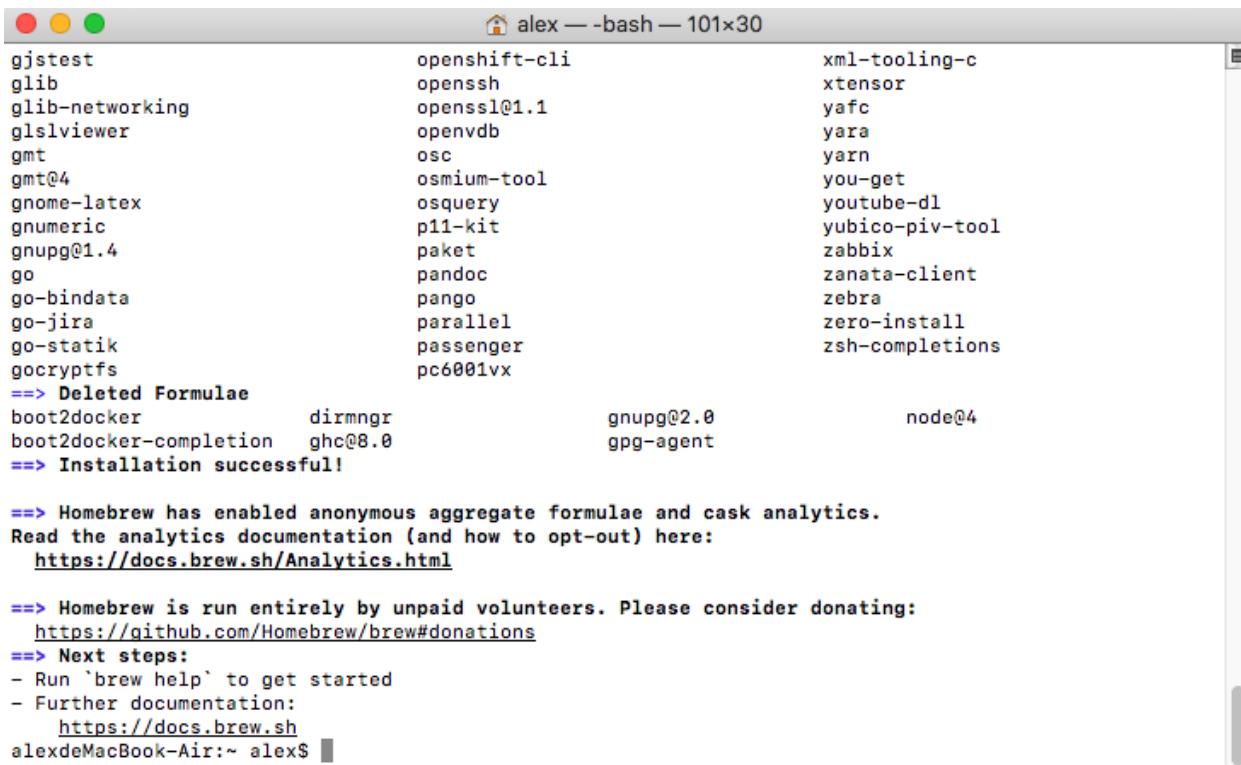


```
alex — sudo - ruby -e #!/System/Library/Frameworks/Ruby.framework/Versions/C...
alexdeMacBook-Air:~ alex$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/mkdir -p /Library/Caches/Homebrew
Password: ?
```

Enter the password of administrator. It might ask the password for several times according to

Then the brew should be successfully installed on the Mac.



```
alex — -bash — 101x30
gjstest          openshift-cli      xml-tooling-c
glib             openssh           xtensor
glib-networking openssl@1.1       yafc
glslviewer       openvdb           yara
gmt              osc               yarn
gmt@4            osmium-tool      you-get
gnome-latex     osquery          youtube-dl
gnumeric         p11-kit          yubico-piv-tool
gnupg@1.4        paket            zabbix
go               pandoc           zanata-client
go-bindata      pango            zebra
go-jira          parallel         zero-install
go-statik        passenger        zsh-completions
gocryptfs       pc6001vx

==> Deleted Formulae
boot2docker      dirmngr          gnupg@2.0          node@4
boot2docker-completion  ghc@8.0      gpg-agent

==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics.html

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations
==> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://docs.brew.sh
alexdeMacBook-Air:~ alex$
```

2. Install node and watchman with Homebrew

Open terminal on Mac and enter the command respectively below.

```
brew install node
brew install watchman
```

3. Install React Native CLI

Open terminal on Mac and enter the command below.

```
npm install -g react-native-cli
```

4. Install Xcode and the Xcode Command line tools

Install Xcode from the Mac app store. Or you can download the whole package of Xcode from

<https://itunes.apple.com/us/app/xcode/id497799835?mt=12>

Run the application on emulator

Android:

Step 1:Open command and switch to the location of the project by using "cd " command.

Step 2:Enter the command: npm-install

Step 3:Open Android Studio and boot an android emulator from AVD(Android Virtual Device). Before doing this, please make sure you have setup a Android virtual device correctly. About how to setup a virtual device on Android Studio, please follow the official guide: <https://developer.android.com/studio/run/managing-avds>

Then enter the command:

```
react-native run-android
```

Now the application should be runned on Android emulator.

iOS:

Step 1:Open terminal and switch to the location of the project by using "cd " command.

Step 2:Enter the command: npm-install

After install the necessary package for the project then enter the command:

```
react-native run-ios
```

Now the iOS emulator should be boot and the application shall be runned on it. Please make sure your Xcode is installed correctly in the latest version.

Deploy the release version:

Note:All the template are based on Mac OS. It will be slightly different on Windows, and publish iOS version is not available on Windows.

Android:

Generating a signing key:

Generate a private signing key by using command below on terminal

```
keytool -genkey -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000
```

This command prompts you for passwords for the keystore and key, and to provide the Distinguished Name fields for your key. It then generates the keystore as a file called my-release-key.keystore.

The keystore contains a single key, valid for 10000 days. The alias is a name that you will use later when signing your app, so remember to take note of the alias.

On Mac if you not sure where is your jdk bin folder is then perform the following command to find it, \$ /usr/libexec/java_home it will output the directory of jdk which looks like this:

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home
```

then navigate to that directory by the following command,

```
cd /Library/Java/JavaVirtualMachines/jdk1.8.0_161.jdk/Contents/Home/
```

Now you can perform the keytool command with sudo permission as shown below

```
sudo keytool -genkey -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000
```

Note: Remember to keep your keystore file private.

Setting up gradle variables

1.Place the my-release-key.keystore file under the android/app directory in the project folder.

2.Edit the file ~/.gradle/gradle.properties or android/gradle.properties and add the following (replace ****with the correct keystore password, alias and key password):

```
MYAPP_RELEASE_STORE_FILE=my-release-key.keystore
MYAPP_RELEASE_KEY_ALIAS=my-key-alias
MYAPP_RELEASE_STORE_PASSWORD=*****
MYAPP_RELEASE_KEY_PASSWORD=*****
```

3>Edit the file android/app/build.gradle in your project folder and add the signing config

```
android {
    defaultConfig { ... }
    signingConfigs {
        release {
            if (project.hasProperty('MYAPP_RELEASE_STORE_FILE')) {
                storeFile file(MYAPP_RELEASE_STORE_FILE)
                storePassword MYAPP_RELEASE_STORE_PASSWORD
                keyAlias MYAPP_RELEASE_KEY_ALIAS
                keyPassword MYAPP_RELEASE_KEY_PASSWORD
            }
        }
    }
    buildTypes {
        release {
            ...
            signingConfig signingConfigs.release
        }
    }
}
```

Generating the release APK

Simply run the following in terminal:

```
cd android
./gradlew assembleRelease
```

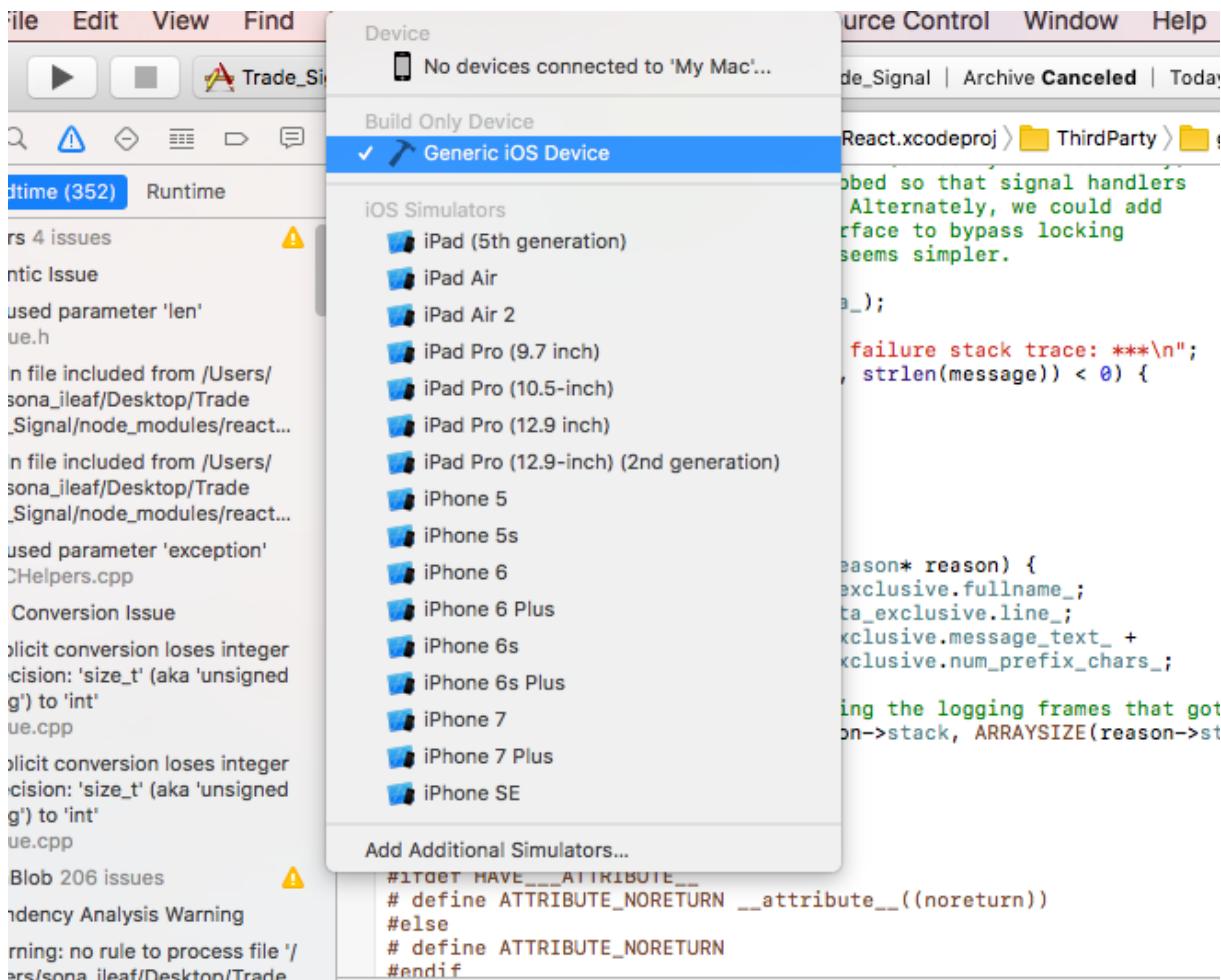
Note: Make sure gradle.properties does not include org.gradle.configureondemand=true as that will make release build skip bundling JS and assets into the APK.

The generated APK can be found under android/app/build/outputs/apk/app-release.apk, and is ready to be distributed. If you want to publish to Google Play, please follow the guide provided by Google:<https://play.google.com/apps/publish/>

iOS:

Create the appropriate apple developer account (personal or business). Then login into Xcode with this account to make sure the certificate is valid for publish application.

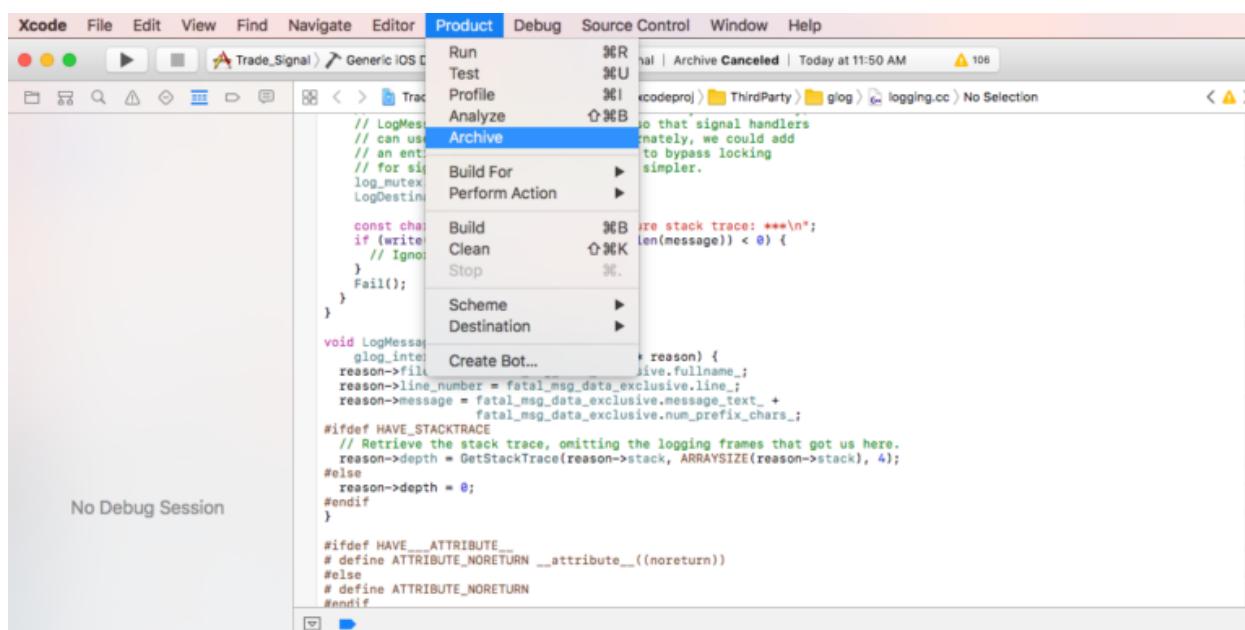
After creating the provisioning certificate, In Xcode menu,Select the device as Generic IOS Device .



Product->Clean and build your application.

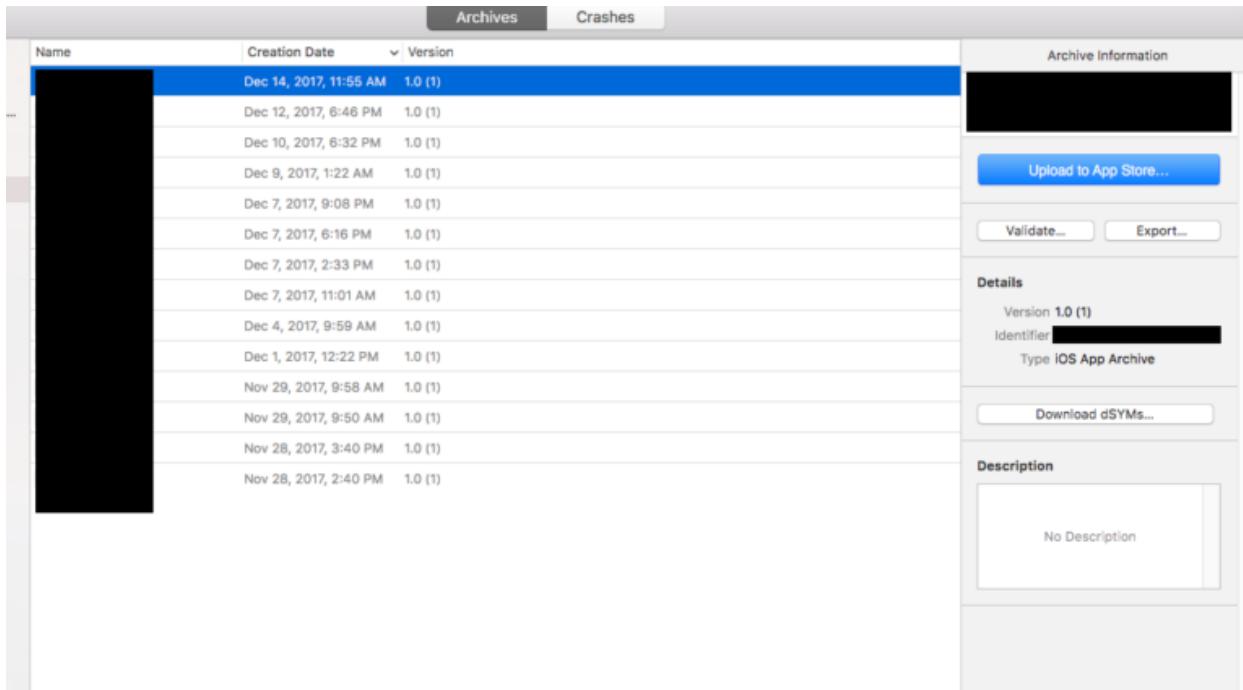
If the provisioning certificate is not valid, it will lead to build failed with an error. If so you have to change it to a valid certificate through your project->build settings. Else if build succeeded just leave this step.

Then select Product->archive, it will create an archive file in Organiser->archives and open it up for you after archive finished.

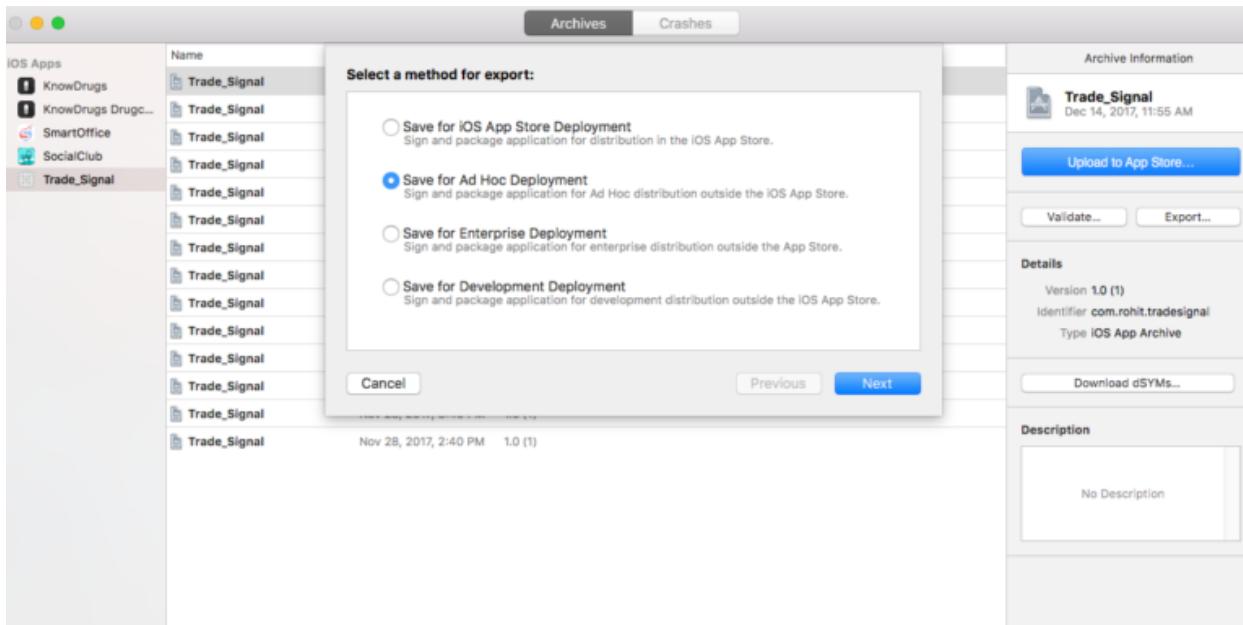


Once archive is successfully completed it will open in Organizer window

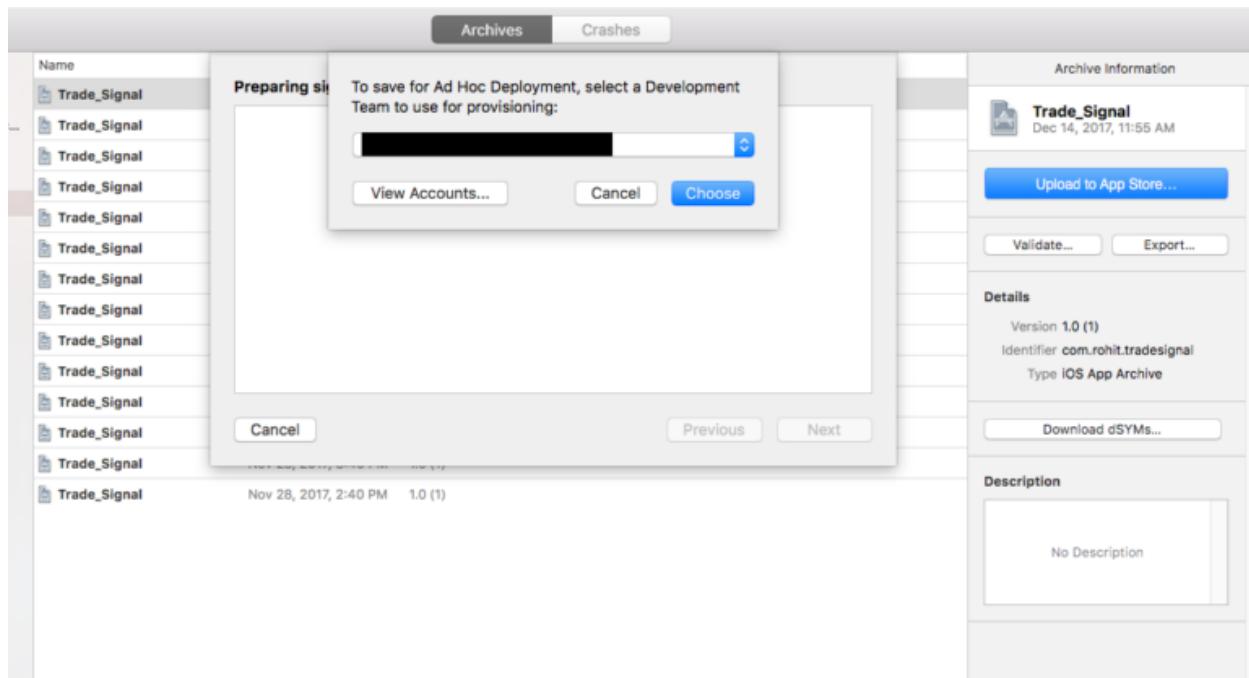
Then select the export function from the right side section.



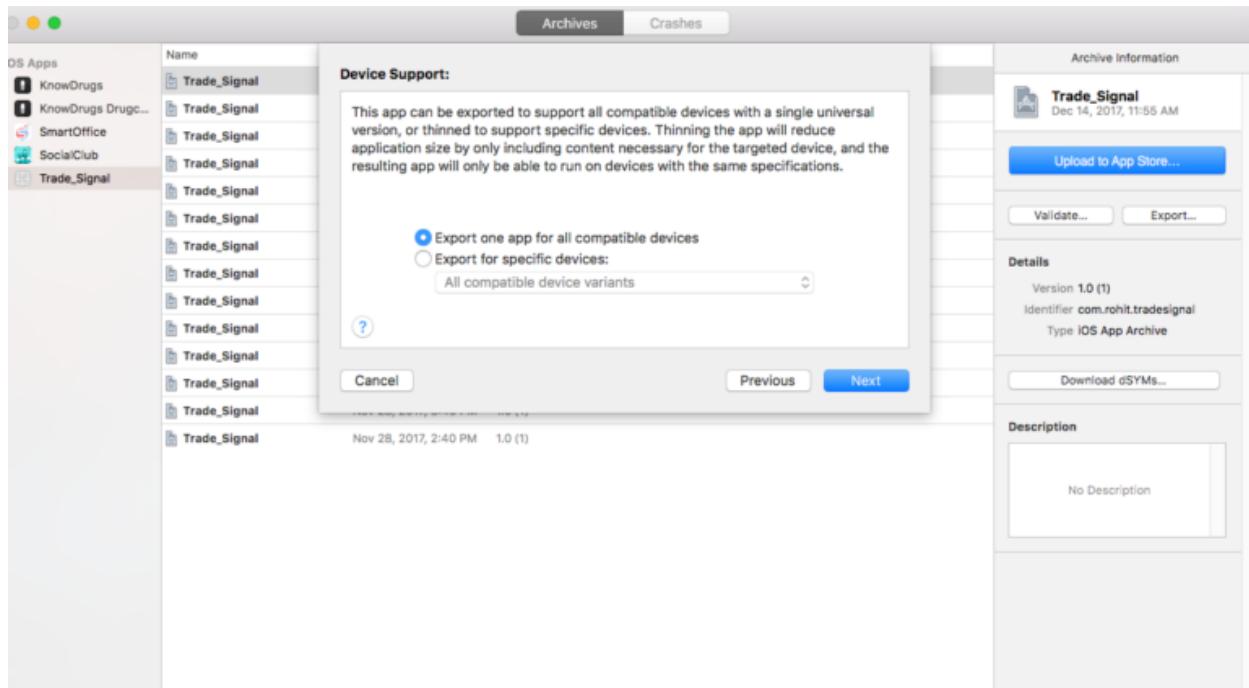
Select a method for export as per your need and click on Next button.



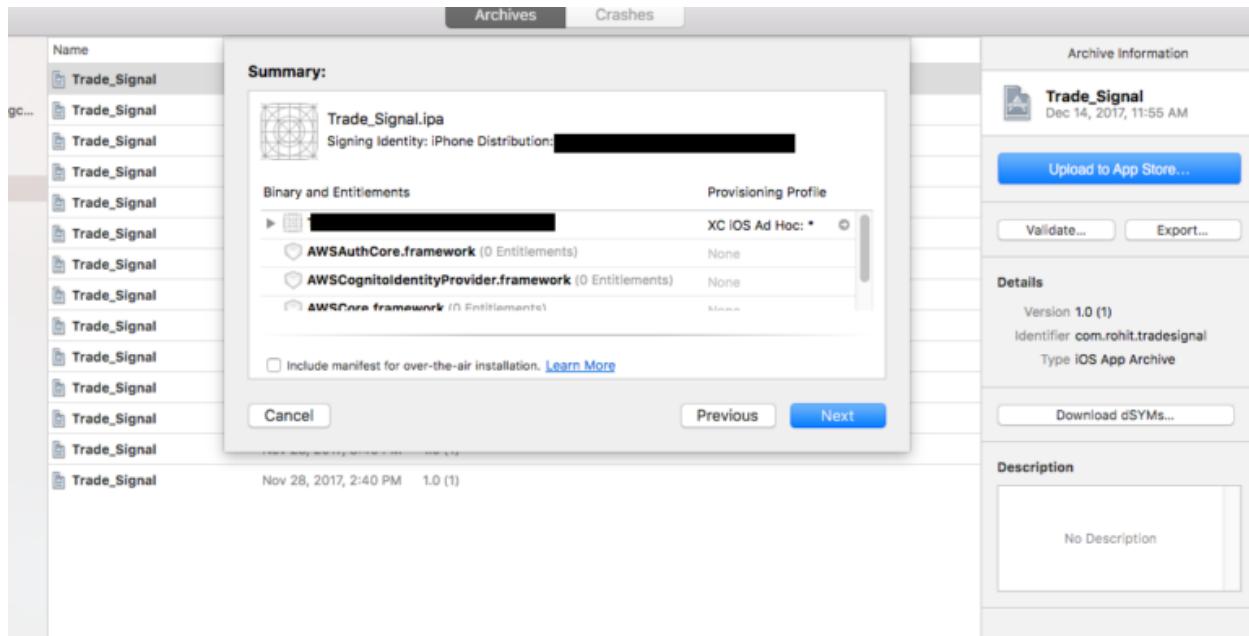
Then choose your development team to use for provisioning.



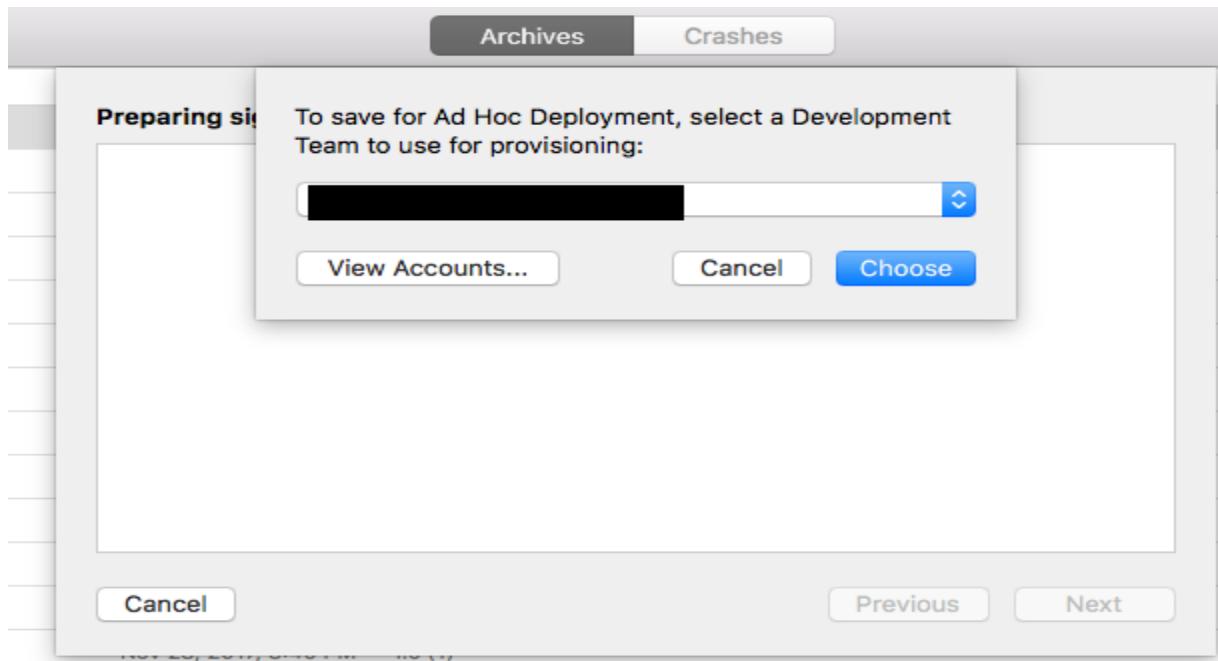
Then select the required device support as per your need and click on Next button.



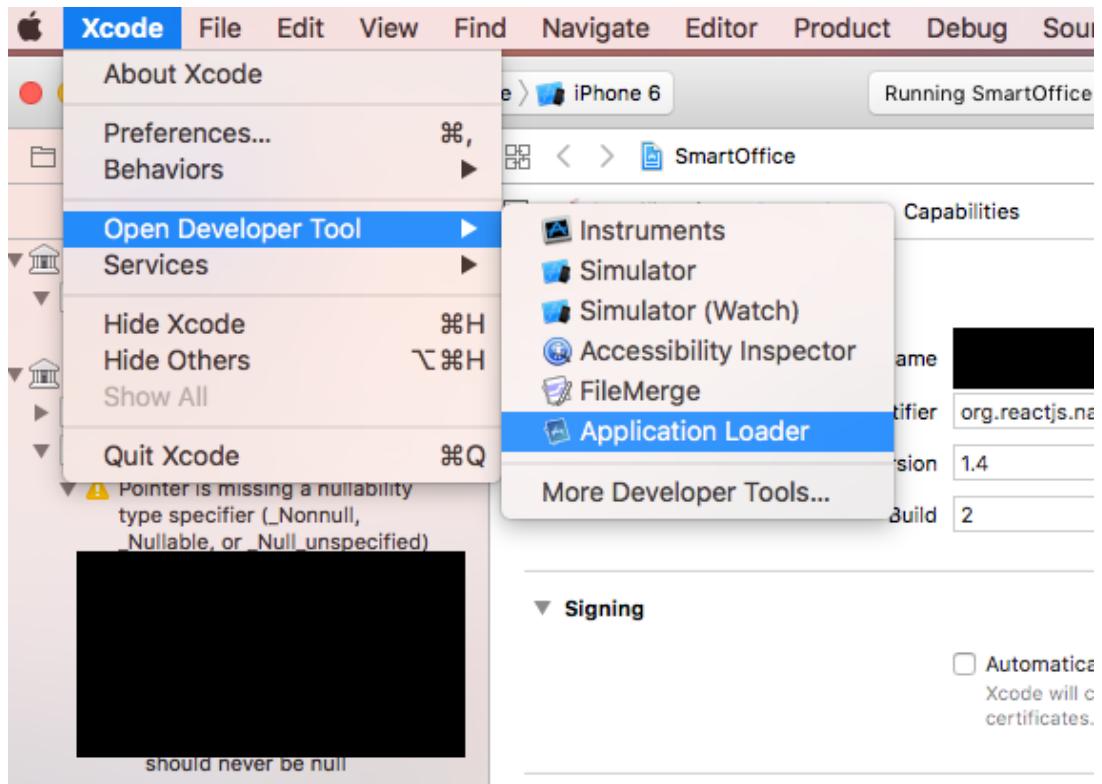
Then it shows the summary and click on Next button.



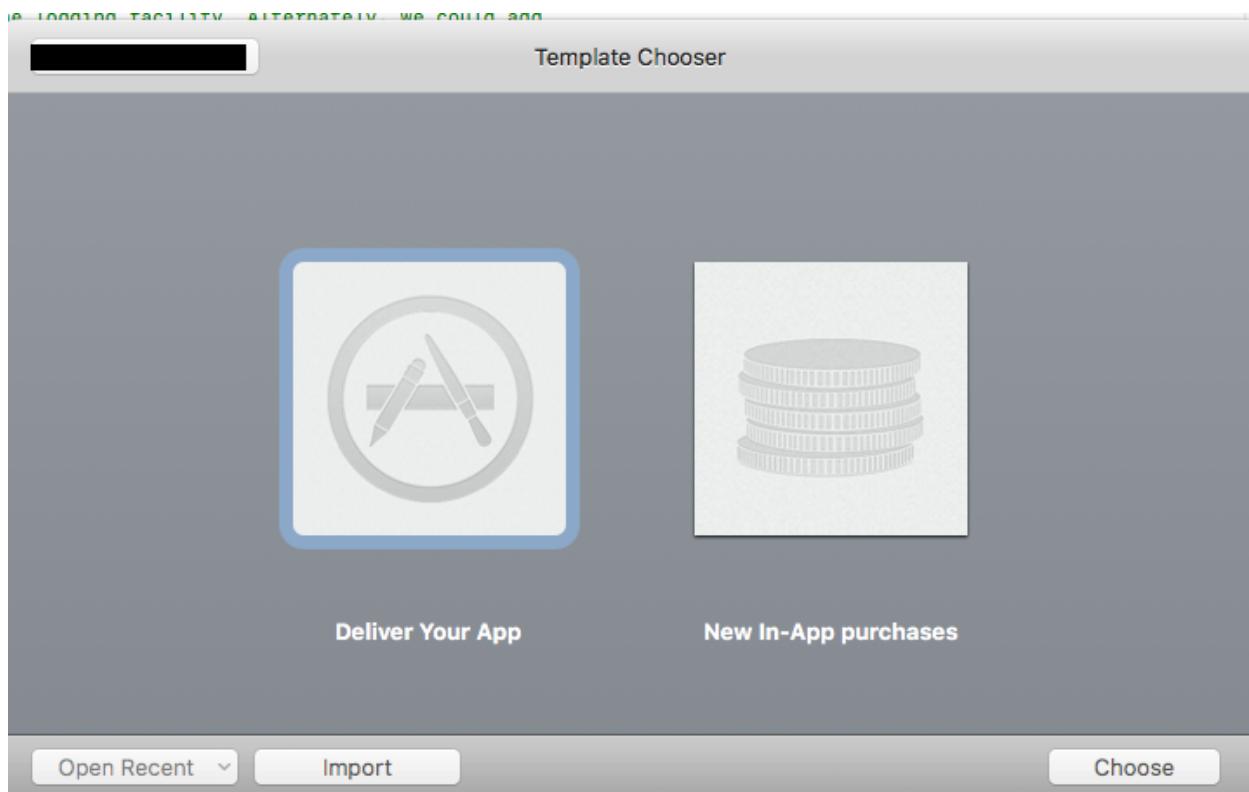
Now choose the location where folder is to be saved and Click on Export



Please select open xcode ->development tool ->Application Loader from xcode seen in the top left corner of the window



In application Loader choose your development team to use for provisioning from the left top corner and choose the .ipa file from the bottom left side



Submitting the app to itunes in <https://developer.apple.com>.



deployment_MAC.command

GENIE Script Deployment

Description:

This document is about how to setup environment of development and release of the GENIE script.

Setup environment

Environment Information in total:

Software	Version
JRE	1.7
Maven	3.5.4

Please make sure the software above is in the same or higher version.

Mac:

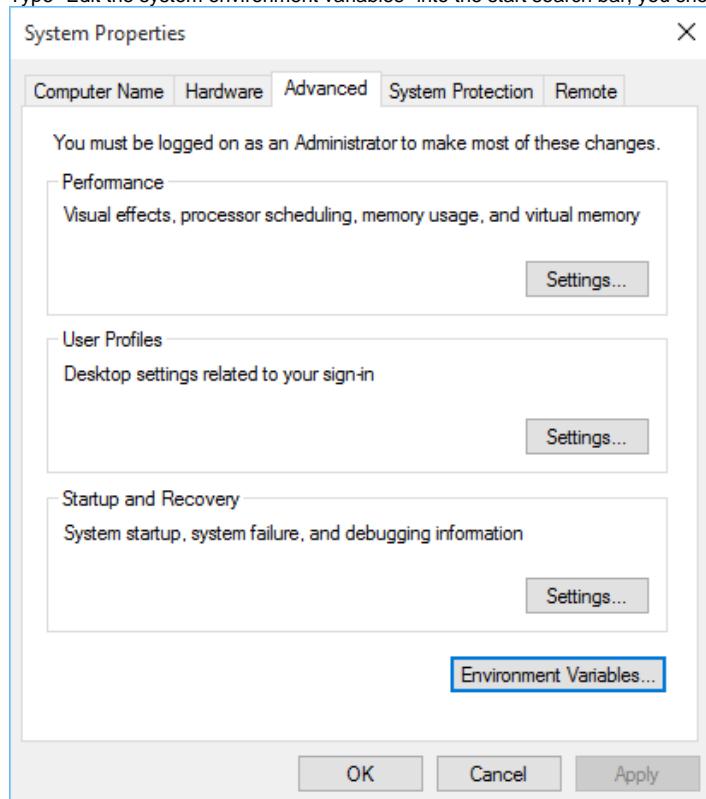
1. Install the latest version JDK. It can be downloaded from: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Install Maven
 - a. Open a terminal and type the following commands

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```
 - b. brew install maven
3. Open a terminal window and navigate to the folder where you have downloaded the source code
4. Type in the following commands
 - a. mvn clean compile assembly:single
 - b. cd target
 - c. java -jar GENIEUpdateClient.jar
5. The script should now be running in a window, and should maintain itself. In the event that the window closes, or the computer is restarted, restart the script using step 4. In the case that problems occur, closing and opening the script may resolve some issues.

Windows:

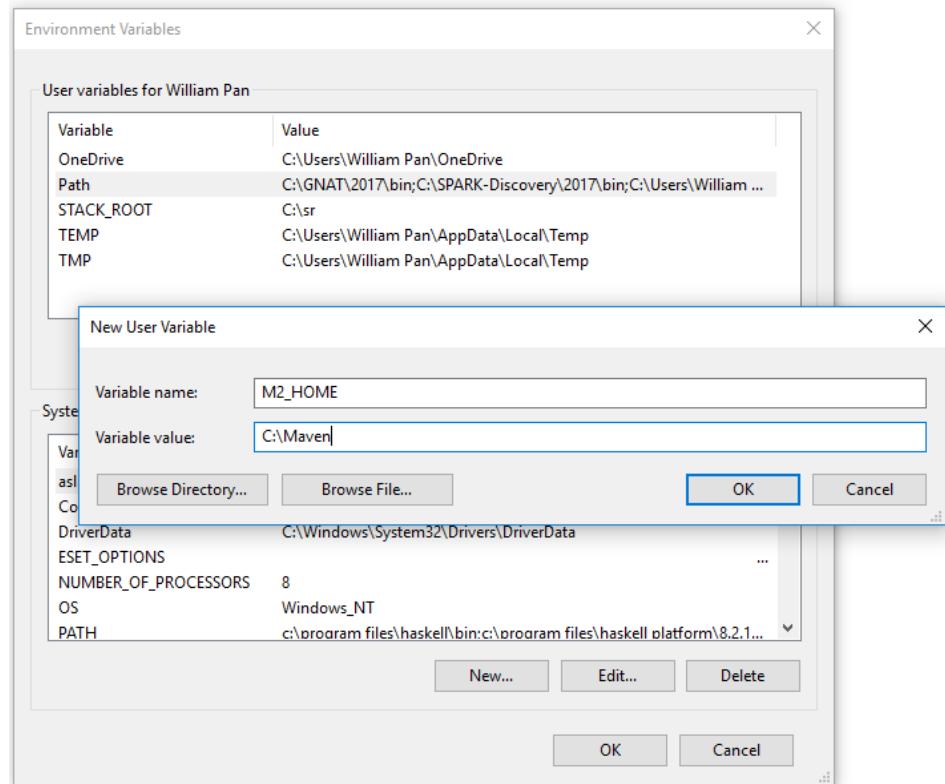
1. Install the latest version JDK. It can be downloaded from: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Install Maven (More detailed instructions can be found here <https://www.mkyong.com/maven/how-to-install-maven-in-windows/>)
 - a. To install Maven on windows, head over to [Apache Maven site](#) to download the latest version, select the Maven zip file, for example apache-maven-3.5.4-bin.zip.

- b. Type "Edit the system environment variables" into the start search bar, you should end up with this

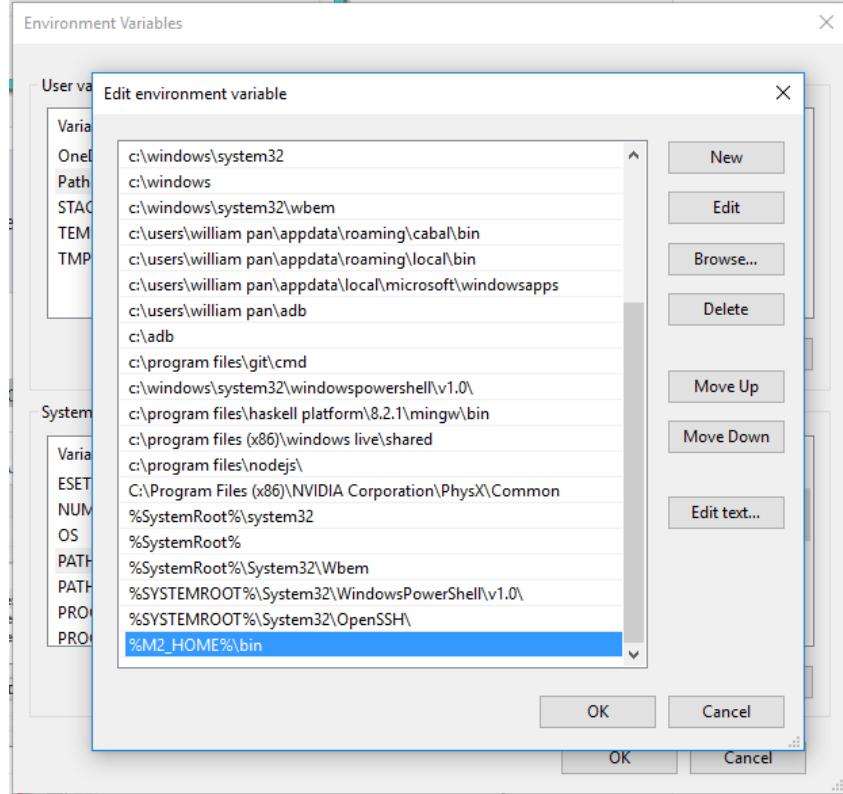


- c. Then click on Environment Variables

- d. Click on New and add both **M2_HOME** and **MAVEN_HOME** variables in the Windows environment, and point both to your Maven folder (Where you downloaded Maven to).



- e. Then click on edit PATH in the system variables box and add %M2_HOME%\bin

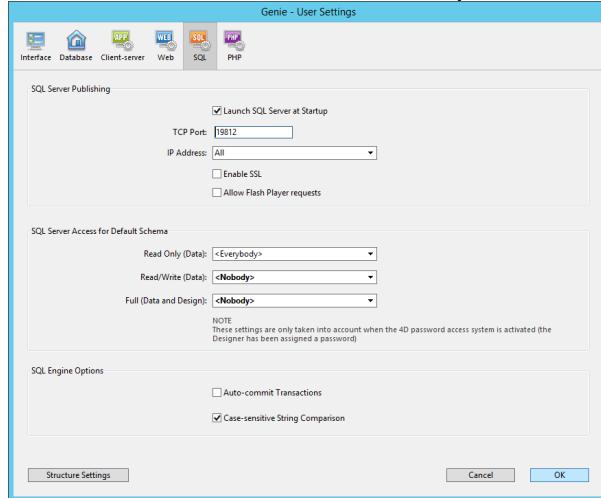


3. Navigate to the folder where you have downloaded the source code
4. Shift + Right click in the folder and click on "Open Powershell Window Here"
5. Type in the following commands
 - a. mvn clean compile assembly:single
 - b. cd target
 - c. java -jar GENIEUpdateClient.jar
6. The script should now be running in a window, and should maintain itself. In the event that the window closes, or the computer is restarted, restart the script using step 5. In the case that problems occur, closing and opening the script may resolve some issues.

Configuring GENIE:

1. For security reasons, the script should be set up on the same computer (in the same instance) as the GENIE server.
2. Configure the firewall
 - a. In running GENIE, the server should already have an existing, configured firewall. In the case that it does not, it is possible to configure the SQL Server to only accept local traffic, but this does not guarantee the security of the Application Server. As such, we do not support it as a base configuration.
 - b. Configuration will vary based on your firewall's vendor; consult their documentation if you require additional assistance.
 - c. The script will require the right to send and receive local traffic, as well as send traffic to your database server in the case that it is hosted remotely.
 - d. GENIE Server will require the right to receive local traffic on the port configured for the SQL Server. If it is not being used by an existing application, for security reasons, we recommend that only local traffic be accepted.
3. Configure GENIE's SQL User Settings
 - a. In the GENIE server window menu bar, select Edit > Database Settings... > User Settings...
 - b. In the User Settings dialog toolbar, select SQL.
 - c. Ensure that the "Launch SQL Server at Startup" option is ticked.

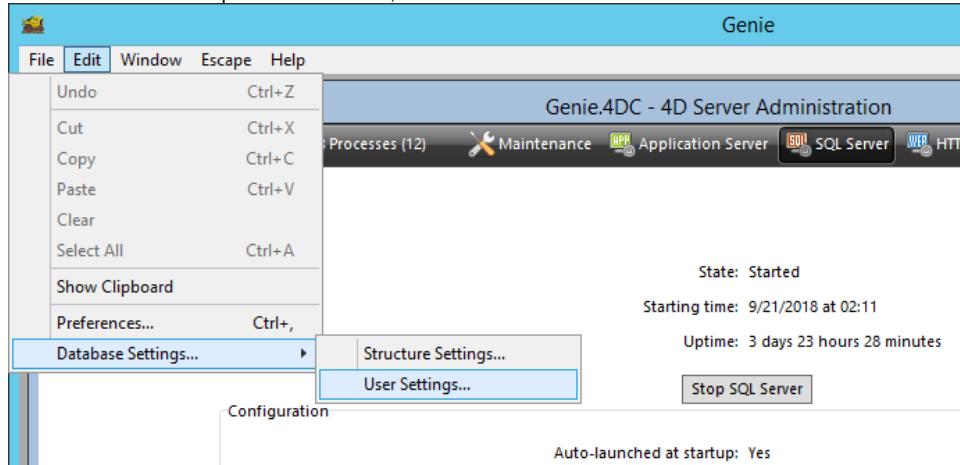
- d. If "Read/Write (Data)", or "Full (Data and Design)", under "SQL Server Access for Default Schema" are set to "<Everybody>", and your other business activities do not use this functionality, set them to "<Nobody>", as shown here:



- e. (If there is no firewall, IP Address will need to be set to only allow local traffic. Again, this is not a supported configuration, as a firewall is necessary to ensure the security of the Application Server.)
f. If the TCP Port is set to a port other than 19812, note this down in order to configure the script later.
g. Click the OK button to save these changes.

4. Start the SQL server

- a. In the 4D Server Administration window toolbar, select SQL Server.
b. "Auto-launched at startup" should read Yes, as shown below:



- (Also shown is the menu path to select User Settings as required for step 3.)
c. If the State does not read Started, press Start SQL Server.
d. In the case that problems occur, Stopping and Starting the SQL Server may resolve the issue. In particular, this may help if "Number of connections" under Connections reads 1 or more for long periods of time, or the State is not Started.