Implementation Model

PRM

[**Introduction 1**](#_tn4i2zufk68l)

[**Implementation Strategy 2**](#_s063kwbfjmrb)

[Deployment Strategy 2](#_3wspaojzo8bc)

[**System Configuration 2**](#_iws1nahft8d)

[Hardware Specifications 2](#_ypi0lo5zhub3)

[Android Dependencies 3](#_83gw0ih4goe5)

[Compose 3](#_y5yfdk78ljsy)

[Firebase 3](#_bimifcqa59a8)

[Google Play Services 3](#_34blbu9xw0fu)

[Coil 3](#_bvhuepij4i13)

[Accompanist 3](#_qry22o2bwgyd)

[Room Database 3](#_mv6qu3vbxr7f)

[SQLite 4](#_fpeal38edcnp)

[KTX 4](#_jvcyxfqn4vb7)

[Testing 4](#_1uk01b843ooc)

[Firebase Settings 4](#_j07mtd37dtu3)

[**User Management 4**](#_oi80vlkpz7mp)

[Authentication Methods 4](#_m48q6b8vaueq)

[Access Control 5](#_c3wduquh1lvq)

[**Security Measures 5**](#_vcxxadmkkjav)

[**Documentation 5**](#_e5tauqp7q5w0)

[Application Users 5](#_s6o6pl7797kh)

[Development Team 5](#_yx0kjv65xw8v)

# Introduction

This document outlines the implementation strategy for the production-ready deployment of the ADHD Task Management application. This project has successfully developed a time management Android application for individuals with ADHD, and it has undergone rigorous testing and refinement during the beta testing phase. This document details the comprehensive plan for transitioning the application from beta testing to full production deployment. It covers the deployment strategy, system configuration, user management, security measures and documentation.

# Implementation Strategy

The implementation strategy for the final production version of our application is to use the rolling deployment model. We have decided to use this approach because it will allow us to make small changes to the application over time and has the following advantages and disadvantages.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Minimal Downtime | * Limited Testing |
| * Easy Rollback |  |

Overall, the project team feel that using this deployment model has benefits for deploying a mobile application that outweigh the disadvantages.

It also allows the project team to schedule small incremental changes once per week to ensure that the application has regular bug fixes and improvements. As noted above the other advantage of this approach is that it will allow the project team to easily roll back to an earlier version of the application if there is an issue with a release.

## Deployment Strategy

The application will be deployed on the Google Play store once it is approved for distribution. This will allow the project team an easy way to monitor usage, receive feedback from users and provide updates to the application users.

# System Configuration

## Hardware Specifications

The application can run on any Android mobile device that is using Android 9 (API v28.0) to Android 14 (API v34.0) which at the time of this document being written is the most up-to-date version of Android.

## Android Dependencies

Below is a list of dependencies that the project team have used while building the application.

### Compose

* Compose Live data
* Compose UI
* Compose UI Preview
* Compose Material
* Lottie Compose
* Compose Icons
* Compose BOM
* Activity Compose

### Firebase

* Firebase BOM
* Firebase Auth
* Firebase Storage
* Firebase Firestore

### Google Play Services

* Play Services Auth

### Coil

* Hilt
* Hilt Compiler

### Accompanist

* System UI Controller
* Permissions
* Constraints Layout
* Appcompat

### Room Database

* Room KTX
* Room Runtime
* Room Compiler

### SQLite

* SQLite KTX
* SQLite Framework
* Security Crypto KTX
* Android Database SQL Cipher

### KTX

* Kotlin StdLib
* Kotlin StdLib JDK

### Testing

* Compose Compiler
* UI Tooling
* UI Test Manifest
* JUnit 4
* Expresso Core
* JUnit 4 UI Test

## Firebase Settings

The project team are using Firebase to handle the authentication of our application users, storing the user’s account information in the cloud, including the user’s rewards earned and total points and we also use Firebase to get the data related to the leaderboard functionality of the application.

Firebase, as the backend to our application handles all the functionality for our user’s settings, for example, it handles the users changing their username, profile picture and country of origin. The project team decided to use Firebase for this application because we felt that it would remove a lot of the complexity of handling user accounts and allow the project team to focus their efforts on other features of the application.

# User Management

## Authentication Methods

The users of our application will have two methods for creating an account, the first will be for them to use an existing Google account, and the second is for the user to create an anonymous account that is not linked to them personally.

## Access Control

Users will only have access to information that is related to them, for example, users will only be able to see tasks that were created by them. Users will also only be able to change information about themselves and not other users.

The project team feel that this approach will allow the application to respect the privacy of the user and will allow the user to use the application with the confidence that their private information is being protected.

# Security Measures

In addition to the access control noted above, the project team has taken care to protect the information that is collected by the application. This includes encrypting the room database that is stored locally on the user’s device and only allowing information that belongs to the currently signed-in user to be revealed.

The project team have also used methods that are recommended by Google to store API keys that link the application to the Firebase services and to protect the passphrase that enables the locally store room database to be encrypted and decrypted.

# Documentation

## Application Users

The users of the application have a very detailed user manual available to them in the application that provides information on how to use all of the functionality of the application.

## Development Team

The project team has also created development documentation, that will help any future development teams that are working on this application. The development documentation includes information that explains what logic each code file contains, and important information about the logic of each method in the application.

The development document also contains important information about the application architecture and the dependencies used in the development of the application.

This will enable and future development team to gain a solid understanding of how the application was built and the logic behind it, which will speed up development times in the future.

The development document can be found [here](https://github.com/commet003/ITC303-9-Team1-Project/blob/testing/PRM%20Documents/ADHDTaskManager-ProgrammerDocumentation.pdf).