DocCare

**Software Development Plan (Small Project)**

**Version 1.**1

**Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 02/11/2023 | 1.0 | The initial version of Software Planning Document | DocCare Team |
| 8/11/2023 | 1.1 | Add iteration’s objectives and description | Phat Tran |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**1. Introduction 4**](#_heading=h.30j0zll)

[1.1 Purpose 4](#_heading=h.aw83g8xjfjai)

[1.2 Scope 4](#_heading=h.oys48evj3w7p)

[1.3 Overview 4](#_heading=h.p4fh0o3vbwpd)

[1.4 Abbreviations 4](#_heading=h.qo5dee7nfujp)

[**2. Project Overview 4**](#_heading=h.djf5bbhlkso0)

[2.1 Project Purpose, Scope, and Objectives 4](#_heading=h.o5n6df2tnmgs)

[2.1.1 Purposes 4](#_heading=h.rr7c374rmtgz)

[2.1.2 Scopes 5](#_heading=h.ktq1z4l8lz54)

[2.1.3 Objectives 5](#_heading=h.h9t0ruwx0g3l)

[2.1.4 Expectation 5](#_heading=h.sxli10rc68p)

[2.2 Assumptions and Constraints 5](#_heading=h.9z9g7zlfr6bp)

[2.3 Project Deliverables 5](#_heading=h.h63egnzah6k1)

[**3. Project Organization 5**](#_heading=h.p1n2a7fwsgsi)

[3.1 Organizational Structure 5](#_heading=h.dptcnwoykctj)

[3.2 Roles and Responsibilities 5](#_heading=h.bvx7ibd1d8ic)

[**4. Management Process 6**](#_heading=h.2ohm69vnmb8d)

[4.1 Project Estimates 6](#_heading=h.a5e9ujmkpgvt)

[4.2 Project Plan 6](#_heading=h.tiz5flarowa1)

[4.2.1 Phase plan and Iteration Objectives 6](#_heading=h.pbwmuodk32o9)

[4.2.2 Releases 7](#_heading=h.sgfexr8es03u)

[4.2.3 Project Schedule 7](#_heading=h.2ghkikasfn92)

[4.2.4 Project Resourcing 7](#_heading=h.6eq2a8ndrn69)

[4.3 Project Monitoring and Control 7](#_heading=h.epml5vvq6riu)

[4.3.1 Requirements Management 7](#_heading=h.hc8npmmwggjl)

[4.3.2 Reporting and Measurement 7](#_heading=h.gxxrvellq3yg)

[4.3.3 Risk Management 8](#_heading=h.2m6whl65feme)

[4.3.4 Configuration Management 9](#_heading=h.lw8r5dliyltr)

**Software Development Plan (Small Project)**

# **Introduction**

## Purpose

The purpose of the *Software Development Plan* is to define the development activities in term of phrases and iterations required for implementing a medical mobile application management, including: An interactive and user-prompt interface; a space for chronic EMR management for physicians; A convenient scheduler, secure medical application for patients; and a platform for communication between patients and staffs.

The following people use the *Software Development Plan*:

* The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.
* **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

## Scope

This *Software Development Plan* describes the overall plan to be used by the DocCare project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans.  
The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.

## Overview

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives.  It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

## Abbreviations

| **Term** | **Explanation** |
| --- | --- |
| **EMR** | Electronic Medical Records |
| **IT** | Information Technology |
| **U.I** | User Interface |
| **B.a.a.S** | Back-end as a Service |
| **A.P.I** | Application Programming Interface |

# Project Overview

## Project Purpose, Scope, and Objectives

### Purposes

The DocCare IT Developers Team aims to establish a medical smartphone application for EMR management and convenient scheduler for patients and medical staff in a private clinic in Britain.

### Scopes

This project mainly focuses on developing a mobile application for patients, physicians, and medical receptionists of a private clinic in Britain.

### Objectives

1. A secure medical application.
2. An interactive and user-prompt interface.
3. Safe and high performance despite experiencing high loads.
4. Ensuring all features are well-developed and methods correctly operated.
5. The mobile application is completed within a fixed period of time.

### Expectation

DocCare aims to bring the best medical service to the masses, there will be no need for patients’ distanced travel only to set up a check-up. Overall, with DocCare, patients now can access the best medical support from a private clinic right on their phones.

## Assumptions and Constraints

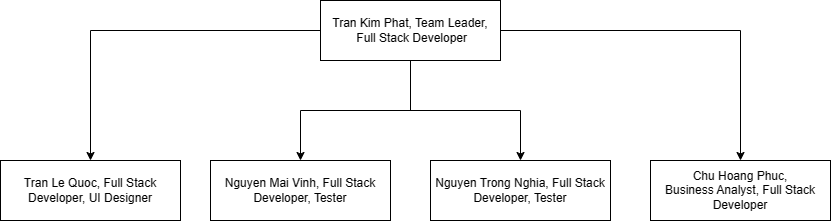
* Zero budget for the project
* Consists of 5 members, no more to be added in the future.
* Finished product will be delivered in 12 weeks
* Developed on Windows systems, while testing on Android devices.

## Project Deliverables

* Gantt Chart (4/11/2023)
* Application interface (2/12/2023)
* Application implementation (16/12/2023)
* Fixing bugs regarding interface and implementation (16/12/2023).
* Test cases (20/12/2023)
* Final test cases and review (30/12/2023)
* Official Releases (31/12/2023)

# Project Organization

## Organizational Structure



## Roles and Responsibilities

| **Person** | **Role** |
| --- | --- |
| Nguyen Trong Nghia | Full Stack Developer, Tester |
| Tran Le Quoc | Full Stack Developer, UI Designer |
| Tran Kim Phat | Team Leader, Full Stack Developer |
| Nguyen Mai Vinh | Full Stack Developer, Tester |
| Chu Hoang Phuc | Business Analyst, Full Stack Developer |

# Management Process

## Project Estimates

We will need at least 12 weeks to work and complete our project. The budget for our project must be at least 16,000,000 VND. Five people will work together for hours or more to complete all tasks in this project.

## Project Plan

### Phase plan

|  | **Objectives** | **Start date** | **End date** |
| --- | --- | --- | --- |
| Inception phase | In this phase, our team focuses on defining general features, advanced features, operations, and tasks for our applications. Additionally, we also aim to complete our database schema, and application design in Figma. Eventually, Vision, Planning documentation, and Gantt Chart for showing the allocation of time to the project phases or iterations will also be delivered. | 21/10/2023 | 4/11/2023 |
| Elaboration phase | During this phase, we focus on completing U.I for 4 roles of DocCare application.  The initial release will also be conducted. This release will provide basic functionalities: navigation between screens, internal state management, authentication flow. | 5/11/2023 | 2/12/2023 |
| Construction phase | Our team aims to implement all essential features, advanced features during this phase. | 3/12/2023 | 16/12/2023 |
| Transition phase | In this iteration, the second release will be delivered with all advanced features. We will invite some users to experience our application and give feedback. From that, we can test the user experience and improve our project. The testing and developing phases will run parallel in order to fix all bugs and enhance the app as quickly as possible. | 17/12/2023 | 30/12/2023 |
| Official Release | Our team plans to review the work, build the release version of DocCare. | 31/12/2023 | 31/12/2023 |

### Iteration Objectives

1. Sprint 1
   1. Timeline: from 21/10/2023 to 04/11/2023
   2. Objectives:
      1. In this iteration, our team focuses on developing the foundation for the application, including: project proposals, project design, project color and text themes.
      2. In addition, project vision, software planning documents are also conducted during this sprint.
      3. Main B.a.a.S platform shall be decided together with database schema built and set up in the service. Simple constraints and triggers shall also be implemented.
2. Sprint 2
   1. Timeline: from 5/11/2023 to 18/11/2023
   2. Objectives:
      1. Setup B.a.a.S in our project: Firebase, Supabase
      2. Review and rewrite the Vision, and Software planning document (if necessary)
      3. Work on Use-case Specification document (PA02)
      4. The team shall develop the authentication flow of the application successfully, including:
         1. Splash screen, Login screen, Register screen, Forget screen
         2. Internal state management for each screen
         3. A.P.I methods for email/password authentication, Google authentication, and Email reset password
      5. Identifying and constructing reusable widget components
      6. Develop screens, and navigation flow for admin-specific:
         1. Create new staff: Doctor or Receptionist
         2. Add working shifts
         3. Create staff
         4. Building report template widget
         5. Export report screen
      7. Develop A.P.I methods for all data models C.R.U.D operations
      8. Fill the database with sample data
      9. Build complex constraints, triggers and stored procedures as transactions to handle conflict cases: book / unbook an appointment, create new staff
3. Sprint 3
   1. Timeline: from 19/11/2023 to 2 /12/2023
   2. Objectives:
      1. Review and rewrite Use-case Specification document (PA02) (if necessary)
      2. Work on Software Architecture Document (PA03)
      3. The team shall successfully develop all the screens (only the representation layer, with internal state management) in the design with respect to each role, using reusable components developed in sprint 2:
         1. Customer (Patient): Home, Intake, Prescriptions, Scheduling (Booking and Cancelling), Messaging, Notifications, Profile
         2. Doctor: Home, Patients Scheduling, Absence Request, Prescriptions, Intake, Profile
         3. Receptionist: Same as Doctor except Prescriptions and Intake
      4. Navigation flow should be conducted between screens for each role and connect with authentication flow developed in sprint 2.
      5. Build the initial release of the application as the first deliverable.
4. Sprint 4
   1. Timeline: from 3/12/2023 to 16/12/2023
   2. Objectives:
      1. Review and rewrite Software Architecture Document (PA03) (if necessary)
      2. Work on UI prototype (PA04)
      3. Our team shall create interaction between domain layer and representation layer, domain layer and data layer, including implementing streaming data using A.P.I methods defined in sprint 02 to enable real-time data, and:
         1. Developing essential operations / functions like scheduling and unbooking in domain repository and data layer, then connect to the state manager.
         2. Developing updates, and delete operations in domain repository and connect to the state manager for screens that require data modification: Prescriptions(Doctor), Intake(Doctor), Profile(3 non-administrative roles),...
      4. Developing exporting monthly report, and create staff A.P.I in administrative account
      5. Test U.I manually when using emulators and fix if finding ones
5. Sprint 5
   1. Timeline: from 17/11/2023 to 30/12/2023
   2. Objectives:
      1. Review and rewrite UI prototype (PA04) (if necessary)
      2. Writing the Test cases and Test plan (PA05)
      3. Following the test plan to test the application: unit testing, widget testing, state management testing, and integration testing
      4. Build the second release of the application, invite users to use and provide feedback. Fixing bugs related to A.P.I logic, U.I if found
      5. Ensure that the project meets all the functional requirements
      6. Ensure that the project meets non-functional requirements listed in the vision document
      7. Build the final release version of the application
      8. Preparing the slides for presentation for the final lab examination

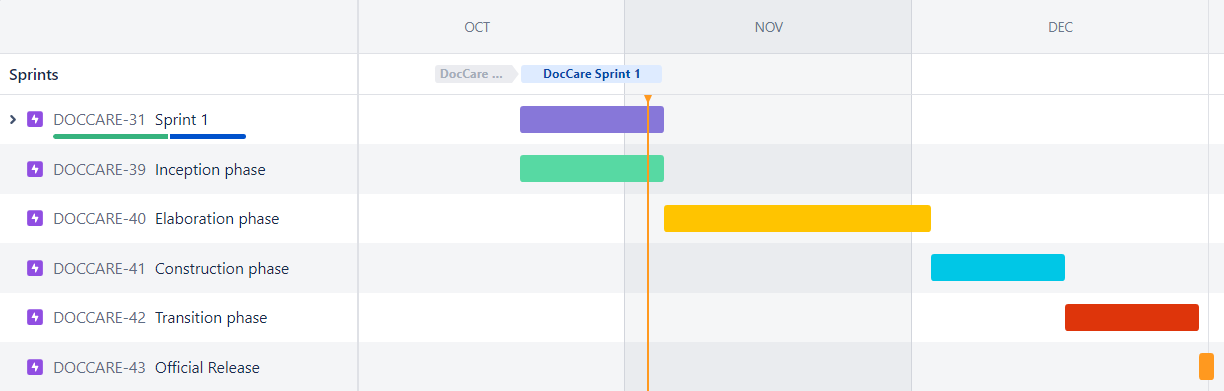
### Releases

The first release version shall be delivered on 2/12/2023, in which the team will have developed basic interface, basic functionalities: navigation, internal state management.

The second release version shall be delivered on 30/12/2023, in which the team shall have resolved all the bugs which are found during testing phase, and added the advanced tested features.

The official release version shall be delivered on 31/12/2023, which is the final version of DocCare.

### Project Schedule

**

### Project Resourcing

The project requires the development team of 5 members, including:

1. One member who is team leader, no special skills required except having a basic understanding of task scheduling, management, and presentation.
2. One member who is responsible for business analysis.
3. One member who is responsible for UI design.
4. Two members are responsible for testing.
5. All team members are fullstack developers, accountable for developing both front-end - interactive user interface, and back-end - implementing business logic.
6. No additional members will be added throughout the entire development of the app.

No special training required.

## Project Monitoring and Control

### Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

### Reporting and Measurement

Each sprint will be 2 weeks in duration. All reports will be generated at the end of each sprint, including:

1. One scrum meeting for each week, with a weekly report as the outcome.
2. One sprint review document before the sprint deadline.
3. One sprint planning document before going on to the next sprint.

### Risk Management

Risks will be identified in the Inception Phase using the steps identified in the RUP for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

| **ID** | **Risk** | **Probability** | **Loss** | **Risk Exposure** | **Mitigation Strategy and/or Contingency Plan** |
| --- | --- | --- | --- | --- | --- |
| 1 | Inaccurate time estimation for tasks | Frequently | Serious | 6,750 | Re-estimate the task, if the task is overestimated, so do more tasks in that sprint. On the other hand, if the task is underestimated, so move some tasks in this sprint to the next sprint in order to have enough time to complete this task |
| 2 | Incomplete or inadequate testing, causing bugs in other process | Frequently | Serious | 6,750 | Require immediate hotfix to that specific problem.  Update the backlogs of this sprint. |
| 3 | Project manager has illness during the important phase | Occasional | Serious | 3,750 | Vote another candidate to be the temporary leader, manager. Connect with the ill leader to collect planned work, requirements, and pre-scheduled tasks. Reestimate the tasks. Moving heavy, non urgent tasks to the next sprint.  Prepare a brief report to the teaching assistants to inform them about the situation. |
| 4 | The feature was too complex that the developer team do not have enough time to complete | Occasional | Serious | 3,750 | Ask teachers to get the instructions for the feature, if it is still too complicated, so the only way is to cancel the feature’s implementation. |
| 5 | Schedule delays due to conflict in time with examination preparation in college | Occasional | Moderate | 2,500 | Try to prepare for the exam as quickly as possible, then return to the task. If we can not complete the task on time, so move the task to the next sprint. |
| 6 | Some team members do not understand code of others | Occasional | Moderate | 2,500 | Ask the team members who write that code and establish communication until all parties have enough understanding of the task, |
| 7 | Changes in requirements, usages and policies for public APIs use in the project | Seldom | Serious | 1,875 | Migrate the system to use backup APIs that provide roughly the same services.  Gather information regarding price A.P.I calls. If not significantly affecting the budget, the team shall be willing to pay for it. |
| 8 | Database suppliers: Firebase and Supabase change policies and not free anymore | Seldom | Serious | 1,875 | Use Sqlite as a temporary alternative database. May find another non-profit supplier or gather information regarding price for storage and A.P.I calls. If not significantly affecting the budget, the team shall be willing to pay for it. |
| 9 | While implementing the project, libraries for some dependencies are outdated, some functions are deprecated. | Seldom | Serious | 1,875 | Search for alternative solutions for these functions in the document of those libraries or try another library. |
| 10 | Data exceeds the allowed rate makes paid-free B.a.a.S become pay-on-usage | Likely | Insignificant | 1,250 | Consider other B.a.a.S solutions: Firebase if the data does not require a structural hierarchy.  Consider paying a small money as compensation for data storage with the current B.a.a.S solution. |
| 11 | B.a.a.S solutions are not available for a short period of time | Seldom | Moderate | 1,250 | Replace the current service with the more friendly, easy-to-implement service: temporary storage on the phone, sqlite an alternative solution.  During the unavailability, continue to develop service to interact with those B.a.a.S solutions beside the alternative solution. |
| 12 | Team member drop out of university (for study abroad, personal reasons) | Unlikely | Catastrophic | 900 | Made changes to schedule so that the remaining members could finish the leftover tasks. If needed, request lecturer and TAs to delay the release schedule, based on how hard the remaining works are. |

### Configuration Management

1. Google Drive for sharing and storing documents, timelog, project guidelines, project vision, planning, and database constraint documentation, change requests documentation.
2. Git, Github for storing project source code.
3. Jira for task scheduling and management.
4. Figma for application design.
5. Google Sheets for logging task estimation and actual spending.