

**INTERNATIONAL SCHOOL**

**Online Clinic Booking System**

**Project Plan**

Project Code: OnClinic-C1 SE.37

Document Code: Project Plan Document v1.1

**Mentor:**Master Nguyen Tan Thuan

**Group:** C1SE.37

* Truong Gia Huy
* Tran Nguyen Thanh Hong
* Le Van Xuan
* Ngo Van Kha

**SIGNATURE PAGE**

|  |  |  |
| --- | --- | --- |
| AUTHOR: | Team C1SE.37 | <Date> |
| REVIEWERS: | Team C1SE.37 | <Date> |
| MENTOR: | Master Nguyen Tan Thuan |  |

**RECORD OF CHANGE**

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Effective Date** | **Changed Item** | **A\* M, D** | **Change Description** | **Reason for Change** | **Revision Number** |
| 16/9/2021 |  | A |  | Initiate document | Draft |
| 17/9/2021 |  | A, M |  | Complete document | 1.0 |
| 18/9/2021 |  | M |  | Format correction and font editing | 1.1 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

*TABLE OF CONTENTS*

[**1. PROJECT OVERVIEW 6**](#_Toc82870860)

[**1.1. Project Description 6**](#_Toc82870861)

[**1.2. Scope and Purpose 6**](#_Toc82870862)

[**1.3. Assumptions and Constraints 7**](#_Toc82870863)

[**1.4. Project Objectives 7**](#_Toc82870864)

[**1.4.1. Standard Objectives 7**](#_Toc82870865)

[**1.4.2. Specific Objectives 8**](#_Toc82870866)

[**1.5. Critical Dependencies (NA) 8**](#_Toc82870867)

[**1.6. Project Risk 8**](#_Toc82870868)

[**2. PROJECT DEVELOPMENT APPROACH 9**](#_Toc82870869)

[**2.1. Technical Process 9**](#_Toc82870870)

[**2.1.1. Reasons for selecting 9**](#_Toc82870871)

[**2.1.2. Agile Methodology 10**](#_Toc82870872)

[**2.2. Quality Management 16**](#_Toc82870873)

[**2.2.1. Strategy for Meeting Quality Objectives 16**](#_Toc82870874)

[**2.2.2. Quality Control 16**](#_Toc82870875)

[**2.2.3. Measurements Program 17**](#_Toc82870876)

[**2.3. Unit Testing Strategy 17**](#_Toc82870877)

[**2.4. Integration Testing Strategy 18**](#_Toc82870878)

[**2.5. System Testing Strategy 18**](#_Toc82870879)

[**3. ESTIMATION 21**](#_Toc82870880)

[**3.1. Size 21**](#_Toc82870881)

[**3.2. Effort 21**](#_Toc82870882)

[**3.3. Schedule 22**](#_Toc82870883)

[**3.3.1. Project Milestone & Deliverables 22**](#_Toc82870884)

[**3.3.2. Detailed Schedule 22**](#_Toc82870885)

[**3.4. Resource 26**](#_Toc82870886)

[**3.5. Infrastructure 27**](#_Toc82870887)

[**3.6. Training Plan 28**](#_Toc82870888)

[**3.7. Finance 28**](#_Toc82870889)

[**4. PROJECT ORGANIZATION 29**](#_Toc82870890)

[**4.1. Organization Structure 29**](#_Toc82870891)

[**4.2. Project Team 29**](#_Toc82870892)

[**5. COMMUNICATION & REPORTING 31**](#_Toc82870893)

[**5.1. Communication methodology 31**](#_Toc82870894)

[**5.2. Reporting methodology 32**](#_Toc82870895)

[**6. CONFIGURATION MANAGEMENT 32**](#_Toc82870896)

[**7. SECURITY ASPECTS 33**](#_Toc82870897)

# PROJECT OVERVIEW

## Project Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Project code** | OnClinic-C1SE.37 | **Contract type** | Internal Project |
| **Customer** | Clinic/Hospital, Patient | **End-User** | Patient |
| **Project Type** | Internal | **Project Manager/ Scrum master** | Huy, Truong Gia |
| **Project Category** | Lifecycle | **Business domain** | Health Care |
| **Application type** | Application Android |  |  |

## Scope and Purpose

Online Clinic Booking System - OnClinic - means an application to book appointments at clinics and examine patients online. It uses intelligent AI technology, which helps to measure human health indicators. From there will give the state of human health. In addition, the application will help users book an appointment to avoid waiting for the clinic.

We use Agile Scrum to develop the project and spend 17 weeks

In this project scope, we just implement these features about hospital, clinic, patient, patient's relatives.

Book an appointment for medical examination offline

* Users enter the application to find a clinic
* Based on the reviews of the clinics, to choose the clinic you want to book an appointment with
* Make an appointment for a clinic
* Go to the clinic/hospital and get medical examination

Book a medical appointment online

* Choose online examination
* Choose the clinic/hospital you want to visit
* Conduct an online examination

Language:

* Vietnamese

## Assumptions and Constraints

|  |  |  |
| --- | --- | --- |
| **No** | **Description** | **Note** |
| **Assumptions** | | |
| 1 | Team members have the necessary skills of teamwork and programming skills | Resources |
| 2 | during the covid-19 pandemic convenient for app development | Market |
| 3 | All equipment is in good condition | Device |
| 4 | There is a mentor to guide | Human |
| **Constraints** | | |
| 1 | Team members must understand and know how to use Java and Python programming languages to develop applications | Technical |
| 2 | Team members must understand and know how to use SQLite programming languages to develop database of applications | Technical |
| 3 | Team members must understand of AI technology to be able to measure heart rate with the camera | Technical |
| 4 | Have full equipment such as: laptop, Macbook, ... for application development | Technical |
| 5 | People resource: 4 people | Human |
| 6 | The product is aimed at people who have a need for medical examination but are limited in terms of travel or time | Market |
| 7 | Total Cost Estimate 4560$; the application will collect an intermediary for 2% of each transaction made on the system | Cost and benefit |

## Project Objectives

### Standard Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| **Metrics** | **Unit** | **Committed** | **Note** |
| Start Date | dd-mmm-yy | 23-08-2021 |  |
| End Date | dd-mmm-yy | 14-12-2021 |  |
| Duration | days | 114 |  |
| Team Size | Person | 4 |  |
| Billable Effort / | Person-day | 19 |  |
| Number of work hours per day for one engineer | Person-hour | 4 |  |

### Specific Objectives

Functional goals:

* Works correctly
* Don’t have a conflict
* Correct to the requirement

Strategic goals:

* Satisfactory requirement documents
* Website will be available at VietNam only

Business goals:

* Income is based on effort
* Website is expected to be released in Dec 2021
* Collect 2% of the medical examination and treatment fee on each medical examination.

## Critical Dependencies (NA)

## Project Risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Description** | **Probability** | **Impact** | **Mitigation Strategy** |
| Lack of coding experiences | Only one member of the team learned Android programming | H | M | Each team member has to learn and help the other to learn quickly. |
| Source Code conflict | Problems while merging code between members to master branch | H | H | Each team member must resolve conflicts by using git merge CLI before merging to master branch. |
| Member conflict | Team member maybe conflict with each other while discussing | H | M | Team building, playing board games to get everyone together. |
| Teamwork tools | Teams maybe have trouble finding the suitable tools for teamwork | H | M | Try to find and learn how to use the tools suitable |
| Time management | Every member has to study at school. | H | H | Overtime |

|  |  |  |  |
| --- | --- | --- | --- |
| **Probability** | | **Severity** | |
| L | Rarely happened. | L | Low damaged |
| M | Sometime happened | M | Medium damaged |
| H | Usually happened | H | Serious damaged |

# PROJECT DEVELOPMENT APPROACH

## Technical Process

### Reasons for selecting

Why do we choose the Agile methodology?

Teams pick Agile to adapt rapidly to market changes or client input without losing a year's value. "Enough" planning and delivering in modest, regular increments enables your team to get input on each modification at a minimum expense.

But it's not only a game of numbers - it's about people first and foremost. Authentic human connections are more essential than strict processes, as articulated in the Agile Manifesto. Instead of established arrangements, working with clients and coworkers is more vital. More essential than hyper-detailed documentation is to provide a workable solution to the customer's problem.

An agile team unifies itself around the same vision and then delivers the finest method to life. Each team establishes its own quality, usability and integrity criteria. Their "definition of finished work" indicates how quickly they churn out. Even if it can be intimidating at first, companies' leaders find it more appropriate for the team to achieve or surpass management objectives if they place their faith in a more agile workforce. This team feels more ownership.

The Agile methodology is also straightforward. The rules, objects, actions, and roles are all simple to comprehend. Its approach really aids in the removal of uncertainties in the development process while allowing firms to add their own unique flavor.

It includes frequent releases, usually every four weeks, and the client obtains a fully functional application with more and more functionality added each time. Agile-Scrum supports adaptable, concurrent workflows. It divides projects into smaller, iterative phases, and always makes sure small parts of the project are completed correctly with less risk. Upgrades and edits or early error fixing can therefore be more easily integrated than when using a V-cycle or Waterfall model.

### Agile Methodology

Agile methodologies are approaches to product development that are aligned with the values ​​and principles described in the Agile Manifesto for software development. Agile methodologies aim to deliver the right product, with incremental and frequent delivery of small chunks of functionality, through small cross-functional self-organizing teams, enabling frequent customer feedback and course correction as needed.

In doing so, Agile aims to right the challenges faced by the traditional “waterfall” approaches of delivering large products over long periods of time, during which customer requirements frequently changed, resulting in the wrong products being delivered.

Agile methodology is an approach to project management that uses four values to organize projects used as four manifestos:

1. **Individuals and interactions** **over processes and tools** - The first value emphasizes teamwork and communication. We must understand that software development is a human activity and that the quality of interaction between people is vital. Tools are an important part of software development, but making great software depends much more on teamwork, regardless of the tools the team may use.

2. **Working software over comprehensive documentation** - Documentation has its place and can be a great resource or reference for users and coworkers alike. The main goal of software development, however, is to develop software that offers business benefits rather than extensive documentation.

3. **Customer collaboration over contract negotiation** - Development teams must work closely and communicate with their customers frequently. By listening to and getting feedback, teams will understand what all stakeholders really want.

4. **Responding to change** **over following a plan** - Changes are a reality in Software development, a reality that your Software process should reflect. A project plan must be flexible enough to change, as the situation demands.

**The Agile Process Flow**

The Agile Software Life Cycle is the organized set of steps through which a product progresses from start to finish. There are six phases in it:

1. Concept - Projects are envisioned and prioritized
2. Inception - Team members are identified, funding is put in place, and initial environments and requirements are discussed
3. Iteration - The development team works to deliver working software based on iteration requirements and feedback
4. Release - QA (Quality Assurance) testing, internal and external training, documentation development, and final release of the iteration into production
5. Maintenance - Ongoing support of the software
6. Retirement - End-of-life activities, including customer notification and migration

#### Scrum Process

Scrum is one of the working models to apply and implement Agile methodology principles.

Scrum Principles, Phases and Processes

The Scrum method relies on the incremental development of a software application while maintaining a completely transparent list of upgrade or correction demands to be implemented (backlog). It involves frequent deliveries, usually every four weeks, and the client receives a perfectly operational application that includes more and more features every time. This is why the method relies on iterative developments at a constant rhythm of 2-4 weeks. Upgrades can therefore be more easily integrated than when using a V-cycle.

Scrum processes address the specific activities and flow of a Scrum project, which are grouped into the following five phases:

1. Initiate - This phase includes the processes related to initiation of a project: Create Project Vision, Identify Scrum Master and Stakeholder(s), Form Scrum Team, Develop Epic(s), Create Prioritized Product Backlog, and Conduct Release Planning.
2. Plan and Estimate -This phase consists of processes related to planning and estimating tasks, which include Create User Stories, Approve, Estimate, and Commit User Stories, Create Tasks, Estimate Tasks, and Create Sprint Backlog.
3. Implement - This phase is related to the execution of the tasks and activities to create a project's product. These activities include creating the various deliverables, conducting Daily Standup Meetings, and grooming (i.e., reviewing, fine-tuning, and regularly updating) the Product Backlog at regular intervals.
4. Review and Retrospect - This phase is concerned with reviewing the deliverables and the work that has been done and determining ways to improve the practices and methods used to do project work.
5. Release - This phase emphasizes on delivering the Accepted Deliverables to the customer and identifying, documenting, and internalizing the lessons learned during the project.

Scrum organization

The Scrum framework involves the following three main players:

* Product owner: In most projects, the product owner is the leader of the client's project team. He is the one who will define and prioritize the product features and choose the date and content of each sprint based on values (workloads) that the team communicates to him.
* Scrum Master: He is a genuine facilitator on the project as he makes sure that everyone works at their full potential by eliminating impediments and protecting the team from external interference. Moreover, he pays particular attention to the respect of the different SCRUM phases.
* Team: A team is typically made up of 4-10 people and groups together all the IT specialists who are necessary on a project, i.e. an architect, a designer, a developer, a tester etc. The team is self-organizing and remains unchanged during an entire sprint.

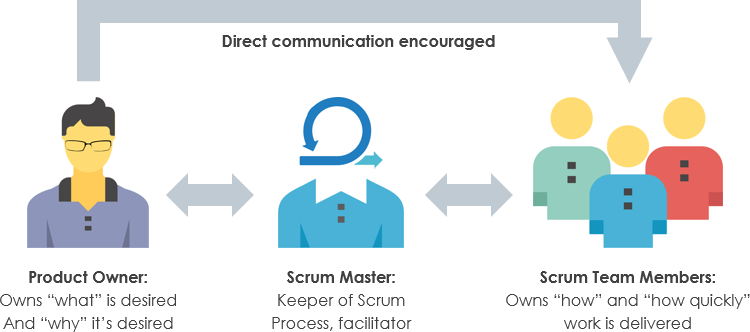


Figure 4.2: Scrum team members

Scrum advantages

In each business there are two significant advantages of using the Scrum technique. The first is that the Scrum teams finish a significantly better quality of their projects. The second greatest advantage of Scrum is that it assures the most effective way for Scrum teams to work.

Scrum efficiently uses time and money resources, since the Scrum teams plan and budget their finances before any project begins. This helps to eliminate unanticipated costs and time delays. Towards the conclusion of each sprint, Scrum teams also test their product for any defects and repair them before release.

A new project has numerous factors and aspects to consider to ensure its success. Scrum techniques make it much simpler to keep a track on these pieces, as large jobs have to be broken up into smaller, more manageable parts and teams.

Projects can evolve and alter over time, which can also affect project needs. Scrum is a change-free strategy since the Scrum teams operate in sprinters, and any modifications required may be quickly incorporated in the following sprint. They must not start at all, which is typically the case with teams who follow the methodology of Waterfall.

Scrum's daily Scrum meetings are another component of the Scrum team. This allows the whole team to monitor the progress of every iteration and to discover any hitches they could face in the project early. This also shows every team member's duties on a daily basis, so that their work is validated.

Because of the positive effects that the Scrum method brings, today this method has become increasingly popular around the world in the software production process.

***In short:***

[*Scrum*](https://www.simplilearn.com/tutorials/agile-scrum-tutorial/what-is-scrum) *can help teams complete project deliverables quickly and efficiently.*

* Scrum can help teams complete project deliverables quickly and efficiently
* Scrum ensures effective use of time and money
* Large projects are divided into easily manageable sprints
* Developments are coded and tested during the sprint review
* Works well for fast-moving development projects
* The team gets clear visibility through [scrum meetings](https://www.simplilearn.com/how-to-conduct-on-target-daily-scrum-meeting-article)
* Scrum, being agile, adopts feedback from customers and stakeholders
* Short sprints enable changes based on feedback a lot more easily
* The individual effort of each team member is visible during daily scrum meetings

Limit and overcome

The Scrum method does not provide a perfect solution to all of the issues that arise during software development. Teams must pay attention to the problem mentioned below, which provide a systematic solution based on extending the approach:

* Team size: team size is typically limited to 4 - 10 people and can become an impediment if it exceeds these recommendations. In the latter case, the organization of meetings becomes impossible and the very foundations of the method are affected. The solution is to set up a Scrum of Scrums. This consists in dividing the project into teams of appropriate sizes and adding an instance of a higher level which groups together the Scrum Master of each Scrum.
* With a whole project of a group of total 4 members, to apply Scrum, the team will divide the roles equally among the members. Each member will be alternately assigned responsibilities, as well as multiple tasks.
* Multiple requests: Requests may be transmitted through several channels on a project and can sometimes be difficult to manage due to their contradictory aspects. These contradictions can slow down the delivery validation process. In order to solve this problem, it is vital to use a single request management tool, which is a standard option with projects.
* Development quality: The more the number of teams increases, the more difficult it becomes to deal with quality. This rule is all the more true when the project is distributed among several centers. The main risks are related to code quality and the number of bugs identified during integration. This is why it is important to have a rigorous quality policy and a project quality plan which precisely defines the rules of the project. Frequent code audits and the implementation of indicators that measure the developers' performance allow minimizing this risk.

## Quality Management

### Strategy for Meeting Quality Objectives

|  |  |
| --- | --- |
| **Strategy** | **Expected Benefits** |
| Do defect prevention using the standard defect prevention guidelines and process; use standards developed in ABC for coding. | 10–20% reduction in defect injection rate and about 2% improvement in productivity |
| Group review of program specs for first few/logically complex use cases.  Group review of design docs/first time-generated code by project leader, developer, and one consultant. | Improvement in quality as overall defect removal efficiency will improve; some benefits in productivity as defects will be detected early |
| Introduction of RUP methodology and implementing the project in iterations. Milestone analysis and defect prevention exercise will be done after each Iteration. | Approximately 5% reduction in defect injection rate and 1% improvement in overall productivity |

### Quality Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Review Item** | **Type of Review** | **Reviewer** | **When** |
| Project plan Project Schedule  CM Plan | Group review Group review  One-person review | Mentor | End of Initiation stage |
| Product Backlog, User Story | Group review | Mentor | End of 90% of requirements |
| Design document, object model | Group review | Mentor | End of 90% design |
| Sprint backlog | One-person review | Scrum Master | Beginning of each stage |
| Test Case | Group review | Product Owner | End of detailed design |
| Code | Group review | Mentor | After coding for first few programs |

### Measurements Program

|  |  |  |  |
| --- | --- | --- | --- |
| **Data to be collected** | **Purpose** | **Responsible** | **When** |
| Size: No. of KLOC// FP | Evaluate size of the system | SM | At the end of stages |
| Effort: No. person-day | Evaluate member productivity | Team members | Daily |
| Quality: No. defects detected | Evaluate, improve the quality of the product | Reviewer, Tester | Right after the review/test |
| Schedule | Control schedule | SM | Weekly and at the end of stages |

## Unit Testing Strategy

* Implemented by the developer.

**Completion criteria:** Completion criteria are stated to for two purposes:

* Identify acceptance criteria for product quality.
* Identify when the testing is successfully executed

A clear statement of completion criteria should include the following items:

* Function, behavior, or condition being measured
* Method of measurement

Criteria or degree of conformance to measurement Special considerations:

This section should identify any influences or dependencies, which may impact or influence the test effort described in the test strategy. Influences might include:

Human resources (such as availability or need for non-test resources to support / participate in test) Constraints, (such as equipment limitations or availability, or the need / lack of special equipment) Special requirements, such as test scheduling or access to systems

Testing may be stopped when

* It becomes unproductive
* It requires a certain coverage
* It requires a certain number of errors to be found
* Schedule time runs out

## Integration Testing Strategy

N/A

## System Testing Strategy

**UI Testing Strategy**

UI Testing involves testing all visual indicators and graphical icons, including menus, radio buttons, text boxes, checkboxes, toolbars, colors, fonts, and more....

UI testing is performed manually or with an automated testing tool. Regardless of the method used, the goal is to ensure all UI elements meet the requested specifications.

**Approaches to UI Testing:**

Using manual testing, a human tester performs a set of operations to check whether the application is functioning correctly and that the graphical elements conform to the documented requirements. Manual-based testing has notable downsides in that it can be time-consuming, and the test coverage is extremely low. Additionally, the quality of testing in this approach depends on the knowledge and capabilities of the testing team.

***Here are a checklist that UI tests tend to verify:***

* Data type errors – Ensure only valid data can be entered for specific data types such as currency and dates.
* Field widths – If a certain text box permits a specified amount of characters, then make it clear on the user interface that the data entered shouldn’t exceed the character limit. (For instance, a field that allows 50 characters in the application’s database should not allow users to enter more than 50 characters on the interface).
* Navigational elements – Verify all navigational buttons on the page are working correctly, and that they redirect users to the right page or screen.
* Progress bars – When displaying screens that take time to render results, a progress bar should be used to show the user that a process is still running.
* Type-ahead – If your UI uses drop-down lists, ensure you include type ahead. In a drop-down menu with hundreds of items, typing the first letter should skip the list to items beginning with that letter such that users will not have to check through a long list.
* Table scrolling – If data in your tables extends to another page, then the scroll function should allow users to scroll the data but keep all headers intact.
* Error logging – When the system experiences a fatal error, ensure the application writes the error details to an event viewer or log file for later review.
* Menu items – Ensure the application only displays valid menu items that are available at a particular state.
* Working shortcuts – For applications that support shortcuts, verify whether they work correctly, no matter the browser, platform, or device being used.
* Confirm action buttons – Ensure the UI has working confirm button every time the user wants to save or delete an item

In addition to these tests, it is recommended to run UI tests on the significant end-to-end user processes.

* UI Testing is completed when the criteria expected in the above checklist are fulfilled.

**Functional Testing Strategy**

Functional testing is a process of verifying that a system performs as expected when its features are exercised by another system or directly by a user. This means that it lends itself nicely to test case and usage case definitions that can provide a stable, repeatable basis for evaluating the progress of system development.

The entire range of the development process comes under the purview of functionality verification.

**Approaches to functional testing:**

Using black box testing. It takes an array of inputs and looks for the generation of specified outputs. The idea behind the name is that the contents of the code under test is unknown to the test case and, by definition, to the tester who is only concerned with function verification.

**Functional testing techniques:**

Functional testing can be broadly divided into two categories – positive and negative testing. Let’s take a closer look at each of these categories:

***Positive Testing***

[This type of testing](https://www.perfecto.io/resources/types-of-testing#performance) ensures that a program meets the basic requirements of the end-users and runs efficiently upon valid inputs and user flows.

***Negative Testing***

[This type of testing](https://www.perfecto.io/resources/types-of-testing#performance) verifies that a program can handle invalid inputs or unintended flows. For example, the software should not crash when entering incorrect characters into a text field.

**Functional testing process:**

Functional testing will proceed in the following 6 steps:

1. Identify and clarify the functions that you expect the software or web application to perform
2. Create input data based on these functional specifications
3. Determine the output based on these functional specifications
4. Write and execute test cases to gather test data (these can be manual or automated tests)
5. Compare the results of actual and expected output
6. Make changes *if* the results don’t match the end user’s needs

* The goal of this testing is to check whether the system is functionally perfect!

# ESTIMATION

## Size

|  |  |
| --- | --- |
| **Modules** | **LOC** |
| Frontend | 1300 |
| Backend | 300 |
| **Total** | **1600** |

## Effort

|  |  |  |
| --- | --- | --- |
|  | **Effort** | **Position** |
| UI/UX Design | 4 | Designer |
| UI Implementation | 2 | Front-end Developer |
| BA | 1 | Business Analyst |
| PM | 1 | Project Manage |
| Development | 4 | Software Engineer |
| QA | 4 | Tester |

## Schedule

### Project Milestone & Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Phase** | **Iteration** | **Start Day** | **End Day** |
| 1 | Development | SPR1 | Sep 19, 2021 | Oct 03, 2021 |
| SPR2 | Oct 04, 2021 | Oct 24, 2021 |
| SPR3 | Oct 25, 2021 | Nov 21, 2021 |
| SPR4 | Nov 22, 2021 | Dec 19, 2020 |

### Detailed Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Task** | **Task owner** | **Start** | **End** | **Days** |
| **1** | **Start-Up** | **All member** | **23/8/2021** | **19/9/2021** | **28** |
| 1.1 | Gathering Requirement | All member | 23/8/2021 | 4/8/2021 | 2 |
| 1.2 | Create Proposal Document | All member | 25/8/2021 | 28/8/2021 | 4 |
| 1.3 | Project’s Kick-off Meeting | All member | 29/8/2021 | 29/8/2021 | 1 |
| 1.4 | Create Documents:   * Project Plan * Database Design * Product Backlog & US * Test Case * UI, Architecture Design | All member | 30/8/2021 | 18/9/2021 | 20 |
| 1.5 | Review Documents | All member | 19/9/2021 | 19/9/2021 | 1 |
| **2** | **Development** | **All member** | **20/9/2021** | **12/12/2021** | **84** |
| **2.1** | **Sprint 1** |  | **20/9/2021** | **10/10/2021** | **21** |
| 1 | Design database of application users | All member |  |  |  |
| 2 | Physical Database Design | All member |  |  |  |
| 3 | Set up Database | All member |  |  |  |
| 4 | Design UI of login/logout/register function | All member |  |  |  |
| 5 | Implementation | All member |  |  |  |
| 6 | Integrate |  |  |  |  |
| 7 | Testing | All member |  |  |  |
| **2.2** | **Sprint 2** |  | **11/10/2021** | **31/10/2021** | **21** |
| 1 | Design for List of clinics function | All member |  |  |  |
| 2 | Finding Data sources | All member |  |  |  |
| 3 | Crawl Data | All member |  |  |  |
| 4 | Implementation for List of clinics and support features (Rating, feedback) | All member |  |  |  |
| 5 | Building UI | All member |  |  |  |
| 6 | Integrate | All member |  |  |  |
| 7 | Testing | All member |  |  |  |
| **2.3** | **Sprint 3** |  | **01/11/2021** | **21/11/2021** | **21** |
| 1 | Design for “Online Clinic Booking” function | All member |  |  |  |
| 2 | Expanding database | All member |  |  |  |
| 3 | Implementation for “Online Clinic Booking” feature | All member |  |  |  |
| 4 | Building UI | All member |  |  |  |
| 5 | Integrate | All member |  |  |  |
| 6 | Testing | All member |  |  |  |
| **2.4** | **Sprint 4** |  | **22/11/2021** | **12/12/2021** | **21** |
| 1 | Complete documents | All member |  |  |  |
| 2 | Design for “Online Examination” function | All member |  |  |  |
| 3 | Build “Measure heart rate with camera” function | All member |  |  |  |
| 4 | Implementation for “Online Examination” function | All member |  |  |  |
| 5 | Building UI | All member |  |  |  |
| 6 | Integrate | All member |  |  |  |
| 7 | Testing | All member |  |  |  |
| **3** | **Project Meeting** | **All member** | **13/12/2021** | **13/12/2021** | **1** |
| **4** | **Final Release** | **All member** | **14/12/2021** | **14/12/2021** | **1** |

## Resource

|  |  |  |
| --- | --- | --- |
| ***Position*** | ***Member*** | ***Effort*** |
| *Designer* | *All member* | *4* |
| *Front-end Developer* | *Trương Gia Huy*  *Trần Nguyễn Thanh Hồng* | *2* |
| *Back-end Developer* | *Ngô Văn Kha*  *Lê Văn Xuân* | *2* |
| *Data Engineer* | *Trương Gia Huy* | *1* |
| *Tester* | *All member* | *4* |

## Infrastructure

|  |  |  |  |
| --- | --- | --- | --- |
| **Work/Product** | **Purpose** | **Expected Availability by** | **Note** |
| **Development Environment** | | | |
| NT Server | Operating System | Initiation stage |  |
| Mobile: Android | Operating System | Construction Stage |  |
| SQLite | DBMS | Construction Stage |  |
| Java | Development language for App interface | Construction Stage |  |
| Python | Development language for App Backend | Construction Stage |  |
| **Hardware & Software** | | | |
| 50GB space on server |  |  |  |
| **Other Tools** | | | |
| Google Sheet | Track member activities. At the end of each day, team members will post on time log and scrum master will check. | Construction stage |  |
| Slack | Store document resources and designed components | Definition stage |  |
| Github | Repositories for source code version management. | Definition stage |  |
| Discord | Discuss online, stream and share problems | Definition stage |  |

## Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| Technical | | | |
| Java Language | All members | 14 days | If already trained |
| Python language | All members | 14 days | If already trained |
| Process | | | |
| Quality system | All member | 3 hrs | If already trained |
| Configuration management | All member | 2 hrs | If already trained for |
| Google Sheet tool | All member | 4.5 hrs | Mandatory |
| Scrum | All member | 2 hrs | Mandatory |

## Finance

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Duration(hours)** | **Salary Rate (USD / Hour)** | **Cost ($)/ per** |
| 1 | 105 | 2.0 | $210 |
| 2 | 105 | 2.0 | $210 |
| 3 | 105 | 2.0 | $210 |
| 4 | 105 | 2.0 | $210 |
| **Total** | **420** | 2.0 | **$840** |

# PROJECT ORGANIZATION

The project team applies the scrum process to project management and development.

## Organization Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Full name** | **Email** | **Phone number** | **Role** |
| Huy, Trương Gia | truonggiahuy7799@gmail.com | 0972173687 | Scrum Master |
| Kha, Ngo Van | ngokha437@gmail.com | 0935950384 | Team member |
| Hong, Tran Nguyen Thanh | thanhhong24102000@gmail.com | 0782363961 | Team member |
| Xuan, Le Van | levanxuan5518@gmail.com | 0905596460 | Team member |

## Project Team

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibility** | **Full name** |
| **Product Owner** | Understand the user and customers with their needs.  Collaborate with the development team.  Manage the stakeholders.  Describe the user experience and product features.  Provides detailed user stories. | Msc Thuan, Nguyen Tan |
| **Scrum Master** | Communicate the value of Scrum  Teach the organization on Scrum to maximize business value  Facilitate Sprint Planning, Daily Scrums, Sprint Reviews and Retrospective Meetings  Create the Task Board and Sprint Burndown Chart at the start of every Sprint  Attend all Scrum meetings  Preserve the integrity and spirit of the Scrum framework  Maintain the focus of the Team  Make the Team aware of impediments and facilitate efforts to resolve them  Serve as a coach and mentor to members of the Team  Respectfully hold the Team, Product Owner and Stakeholders accountable for their commitments  Continually work with the Team and business to find and implement improvements | Huy, Truong Gia |
| **Secretary** | Record the content of group meetings and activities of the member | Hong, Tran Nguyen Thanh |
| **Reviewer** | Analysis of the functions and requirements of the product.  Review documents related to the project | Kha, Ngo Van |
| **Team member** | Frontend Developer | Huy, Truong Gia |
| **Team member** | Back-end Dev | Xuan, Le Van |
| **Team member** | Back-end Dev | Kha, Ngo Van |
| **Team member** | Front-end Dev, Tester, Secretary | Hong, Tran Nguyen Thanh |

# COMMUNICATION & REPORTING

## Communication methodology

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of**  **Communication** | **Communication**  **Schedule** | **Communication**  **way** | **Who Initiates** | **Recipient** |
| Status Report  (Daily meeting) | Daily | Slack, Discord | Scrum Master | Scrum Team |
| Schedule and  Effort Tracking | Daily | Face to face or GSheets | Scrum Master | Scrum Team |
| Work Review | Daily | Face to face | Scrum Master | Scrum Team |
| Work Report | Every Wednesday, Sunday | Face to face or Discord | Scrum Master | Scrum Team |
| Project Review,  ask problems | Every Wednesday, Sunday  (flexible) | Face to face, Discord or Zoom | Scrum Master | Scrum Team,  Mentor |
| Ask & Review  problems | Anytime | Face to face, Slack, Discord, Zoom | Scrum's Member | Mentor and  Scrum Team |

## Reporting methodology

|  |  |  |  |
| --- | --- | --- | --- |
| **Audience/ Attendees** | **Topic/ Deliverable** | **Frequency** | **Method** |
| - Product Owner  - Scrum Master  - Team Members | Project Progress Review | Weekly | Email, Skype or Zoom |
| - Product Owner  - Scrum Master  - Team Members | Explicit Requirement | When needed | Email, Skype or Zoom |
| - Mentor  - Scrum master  - Team members | Milestone review | End of each Milestone | Skype, Zoom |
| - Scrum master  - Team members | Daily tasks | Each day | Slack, Discord, Messenger |

# CONFIGURATION MANAGEMENT

|  |  |  |
| --- | --- | --- |
| **No** | **Tool** | **Content** |
| 1 | Google Document | Track the changing of documents & manage versions of documents. |
| 2 | Github | Repositories for source code version management. |
| 3 | Slack | Store document resources and designed components |

# SECURITY ASPECTS

* Project team members and partners agree not to provide or share user information to third parties in any way.
* Research on network attack prevention solutions to ensure data security, avoid being exploited and stolen data by hackers.

**REFERENCES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Reference item** | **Issued Date** | **Source** | **Note** |
|  |  | dd-mmm-yy |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**DEFINITIONS AND ACRONYMS**

|  |  |  |
| --- | --- | --- |
| **Acronym** | **Definition** | **Note** |
| UI | User Interface |  |
| UX | User Experience |  |
| SPR | Sprint |  |
| SM | Scrum Master |  |
| US | User Story |  |