

**POJECT PLAN DOCUMENT**

**RESTAURANT MANAGEMENT SYSTEM**

|  |  |
| --- | --- |
| **Mentor:** | Nguyen Thi Anh Dao |
| **Team Members:** | Hoang Nghia Khue  Phan Minh Phu  Nguyen Ngoc Thanh  Huynh Viet Tri  Pham Hung Dat |

*Da Nang, 2021*

**PROJECT INFORMATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Acronym** | RMS | | | |
| **Project Title** | RESTAURANT MANAGEMENT SYSTEM | | | |
| **Start Date** | May 13, 2021 | **End Date** | | June 24, 2021 |
| **Lead Institution** | CMU, Duy Tan University | | | |
| **Project Mentor** | Nguyen Thi Anh Dao | | | |
| **Scrum master / Project Leader & contact details** | Hoang Nghia Khue  Email: hoangnghiakhue@gmail.com  Phone: 0793618988 | | | |
| **Partner Organization** | ABC Restaurant | | | |
| **Project Web URL** | https://abcrestaurant.com | | | |
| **Members** | Hoang Nghia Khue | | hoangnghiakhue@gmail.com | 0793618988 |
| Phan Minh Phu | | phu1237@gmail.com | 0365225603 |
| Huynh Viet Tri | | trihuynh11102000@gmail.com | 0365225603 |
| Nguyen Ngoc Thanh | | nguyenngocthanh99qna@gmail.com | 0387943672 |
| Pham Hung Dat | | dat8112000@gmail.com | 0123123123 |

Table 1: Project Information

**PROJECT PLAN DOCUMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Title** | Proposal Document | | |
| **Reporting Period** | May 2021 | | |
| **Author(s)** | Team | | |
| **Date** | May 13, 2021 | **Filename** | 2.ProjectPlan\_Group1\_RestaurantManagementSystem.docx |
| **Access** | Project and FIT program | | |

Table 2: Project Plan Document

REVISION HISTORY

| **Version** | **Date** | **Comments** | **Author** | **Approval** |
| --- | --- | --- | --- | --- |
| 1.0 | May 13, 2021 | Create Document | Team |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 3: Revision History

Contents

[1. PROJECT OVERVIEW 6](#_Toc75184809)

[1.1. Project Description 6](#_Toc75184810)

[1.2. Scope and Purpose 6](#_Toc75184811)

[1.2.1. Scope 6](#_Toc75184812)

[1.2.2. Purpose 6](#_Toc75184813)

[1.3. Assumptions and Constraints 7](#_Toc75184814)

[1.4. Project Objectives 7](#_Toc75184815)

[1.4.1. Standard Objectives 7](#_Toc75184816)

[1.4.2. Specific Objectives 8](#_Toc75184817)

[1.5. Critical Dependencies 8](#_Toc75184818)

[1.6. Project Risk 8](#_Toc75184819)

[2. PROJECT DEVELOPMENT APPROACH 9](#_Toc75184820)

[2.1. Technical Process 9](#_Toc75184821)

[2.1.1. Reasons for selecting 9](#_Toc75184822)

[2.1.2. Agile Methodology 9](#_Toc75184823)

[2.1.2.1. Scrum Process 9](#_Toc75184824)

[2.2. Quality Management 10](#_Toc75184825)

[2.2.1. Estimates of Defects to be detected 10](#_Toc75184826)

[2.2.2. Strategy for Meeting Quality Objectives 11](#_Toc75184827)

[2.2.3. Quality Control 11](#_Toc75184828)

[2.2.4. Measurements Program 12](#_Toc75184829)

[2.3. Unit Testing Strategy 12](#_Toc75184830)

[2.4. Integration Testing Strategy 13](#_Toc75184831)

[2.5. System Testing Strategy 13](#_Toc75184832)

[3. ESTIMATION 13](#_Toc75184833)

[3.1. Size 13](#_Toc75184834)

[3.2. Effort 14](#_Toc75184835)

[3.3. Schedule 14](#_Toc75184836)

[3.3.1. Work Breakdown Structure 14](#_Toc75184837)

[3.3.2. Detailed Schedule 16](#_Toc75184838)

[3.3.3. Project Schedule 20](#_Toc75184839)

[3.4. Resource 20](#_Toc75184840)

[3.5. Infrastructure 20](#_Toc75184841)

[3.6. Training Plan 21](#_Toc75184842)

[3.7. Finance 22](#_Toc75184843)

[4. PROJECT ORGANIZATION 23](#_Toc75184844)

[4.1. Organization Structure 23](#_Toc75184845)

[4.2. Project Team 23](#_Toc75184846)

[5. COMMUNICATION & REPORTING 24](#_Toc75184847)

[6. CONFIGURATION MANAGEMENT 27](#_Toc75184848)

[7. SECURITY ASPECTS 28](#_Toc75184849)

[DEFINITIONS AND ACRONYMS 29](#_Toc75184850)

# 1. PROJECT OVERVIEW

## 1.1. Project Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Project code** | RMS | **Contract type** | Internal Project |
| **Customer** | ABC Restaurant | **End-User** | Customer, restaurant manager and staff |
| **Project Type** | Internal | **Project Manager/ Scrum master** | Hoang Nghia Khue |
| **Project Category** | Development | **Business domain** | abcrestaurant.com |
| **Application type** | Website |  |  |

Table 4: Project Description

## 1.2. Scope and Purpose

### 1.2.1. Scope

* This document provides an overview about product, process, and team in project.
* This is just a general plan then details plan will be updated throughout the life cycle of the project.

### 1.2.2. Purpose

* This document provides a summary of the project's objectives, division of work, the major milestones, required resources, time and overall schedule and budget allocation used and based on the document proposal to build an expense management application on time, at the request and plan.

## 1.3. Assumptions and Constraints

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Note** |
| **Assumptions** | | |
| 1. | Migration to XX for C# will not be done by this team. | Scope |
| 2. | Customer reviewers will get seven days to approve a milestone document. If no comments are received within this time period, it will be considered as approved. | External Interfaces |
| **Constraints** | | |
| 1. | Module A must be completed and delivered to customer before 15-Oct-08 because customer has to demo to its end user by 17-Oct | Schedule |
| 2. | The project shall conform to security requirements specified by the customer in the NDA | Security |

Table 5: Assumptions and Constraints

## 1.4. Project Objectives

### 1.4.1. Standard Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| **Metrics** | **Unit** | **Committed** | **Note** |
| Start Date | 13/05/2021 |  | The start date of the project |
| End Date | 24/06/2021 |  | The end date of the project |
| Duration | 43 days |  | Total day of project. |
| Team Size | 5 People |  | Number of people |
| Billable Effort / | Person-day |  | Effort of team |
| Number of work hours per day for one engineer | 8 hours |  | Number of work hours per day |

Table 6: Standard Objectives

**1.4.2. Specific Objectives**

* Create a resaurant management system with full feature focus on the instant, quickly and useful for management and staff.
* Create trust and become long-term partners with recent user and others.

## 1.5. Critical Dependencies

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Dependency** | **Expected delivery date** | **Note** |
| 1. | Proposal | 16/05/2021 |  |
| 2. |  |  |  |

Table 7: Critical Dependencies

## 1.6. Project Risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Description** | **Probability** | **Impact** | **Mitigation Strategy** |
| **Communication** | In the covid time, every team member has to work and learn online. Not everyone has the same thinking. | 4 | 4 | Learn to use online tools to improve performance, use zoom or google meets to teamwork. Discuss more to understand others. |
| **Programming experience** | Lack of experience in programming. Almost everyone is new to the project technology so they have to learn it from the start | 3 | 3 | Learn and do projects for more programming experience |
| **Estimated project schedule** | Time sort of work. | 4 | 4 | Try to do more to keep up with the project schedule |
| **Technical processes** | Technical processes may not be correct | 2 | 2 | Analyze, supervise technical process, promptly modify if the process deviate |

Table 8: Project Risk

# 2. PROJECT DEVELOPMENT APPROACH

## 2.1. Technical Process

### 2.1.1. Reasons for selecting

* We use Scrum to do the project. Because Scrum is a process suitable for small and medium software, implementation time short and easy change request Scrum is a process popular in present with the advantages and benefits such as: Timeline Flexibility- can later or earlier than originally planned. Good product quality and reduce production risk, low-cost. Interoperability between clients and developers, among the team members are set to high. The growth rate faster, saving time and errors detected earlier.

### 2.1.2. Agile Methodology

* The Agile software development methodology is one of the most simple, yet effective ways to deliver a great product on the market. And yet, somewhere along the way, people started to really overcomplicate it. Properly implemented Agile is fast, flexible, error-proof and just plain better way to manage software development teams. In this article, we’ll explain Agile software development lifecycle phases and how to incorporate Agile principles.

### 2.1.2.1. Scrum Process

* The SCRUM methodology 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.

## 2.2. Quality Management

### 2.2.1. Estimates of Defects to be detected

* Pre-release review defects

|  |  |  |
| --- | --- | --- |
| **Process** | **Planned found by review** | **Actual found by review** |
| **Requirement** | **3** | **4** |
| RMS Software |  |  |
| **Design** | **4** | **5** |
| RMS Software |  |  |
| **Coding** | **7** | **9** |
| RMS Software |  |  |
| **Other** | **2** | **2** |
| RMS Software |  |  |
| **Total** | **16** | **20** |

Table 9: Pre-release Review Defects

* Pre-release test defects

|  |  |  |
| --- | --- | --- |
| **Process** | **Planned found by testing** | **Actual found by testing** |
| **Requirement** | **4** | **6** |
| RMS Software |  |  |
| **Design** | **5** | **6** |
| RMS Software |  |  |
| **Coding** | **9** | **10** |
| RMS Software |  |  |
| **Other** | **2** | **3** |
| RMS Software |  |  |
| **Total** | **20** | **25** |

Table 10: Pre-release Review Defects

### 2.2.2. 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 |

Table 11: Strategy for Meeting Quality Objectives

### 2.2.3. Quality Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Review Item** | **Type of Review** | **Reviewer** | **When** |
| RMS Software | Group review | Team | End of project |
| Project plan Project schedule  CM Plan | Group review | Team | End of Initiation stage |
| Business analysis and requirements specification document, Use Case catalog | Group review | Team | End of 90% of requirements |
| Design document, object model | Group review | Team | End of 90% design |
| Stage plans | One-person review | Khue | Beginning of each stage |
| Complex/first specs incl. diagrams time test generated cases, program interactive | Group review | Team | End of detailed design |
| Code | Group review | Team | After coding for first few programs |
|  |  |  |  |

Table 12: Quality Control

### 2.2.4. Measurements Program

|  |  |  |  |
| --- | --- | --- | --- |
| **Data to be collected** | **Purpose** | **Responsible** | **When** |
| Size: No. of KLOC// FP | Medium | PM/SM | At the end of stages |
| Effort: No. person-day | 8 hours | Team members | Daily |
| Quality: No. defects detected | As little as possible | Reviewer, Tester | Right after the review/test |
| Schedule | 7 days / 1 week | PM/SM | Weekly and at the end of stages |

Table 13: Measurements Program

## 2.3. Unit Testing Strategy

* Magento Functional Testing Framework - Tests the storefront and admin panel UI.
* Web API Functional - Tests the REST, SOAP, and GraphQL areas.
* Integration - Ensures that all the parts work together seamlessly.
* Performance - Tracks changes in CPU, Memory and other metrics. Executed nightly for develop branches. Custom builds can be configured using performance toolkit.
* Client-side performance - Measures total page load times in the browser.
* Load - Tracks the trends of system behavior under the load. Executed internally for each release using scenarios defined in the performance toolkit
* Upgrade - Ensures that a seamless upgrade is possible from previously released product versions to the new one. Executed internally for each release.

## 2.4. Integration Testing Strategy

* So, the modules can be put together in two other typical ways, they can integrated from the top to bottom or bottom to top.
* In Top Down Integration, the main program or the calling program is written first. Then this calls the components below. If a component below is yet to be developed, it is replaced by a passive(temporary) component called Stub.
* In Bottom Up Integration, we move from the bottom to top i.e. the components below are first written and these are integrated first. The integration happens from bottom to top. If the calling component is yet to be developed, it is replaced by a specially written component called a Driver.

## 2.5. System Testing Strategy

* Scope
* Test Approach
* Test Environment
* Testing Tools
* Release Control
* Risk Analysis
* Review and Approvals

# 3. ESTIMATION

## 3.1. Size

* The Size estimation is documented in Project Plan

## 3.2. Effort

* The Effort estimation is documented in Project Plan

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity/**  **Process** | **Total budgeted Effort Usage (pd)** | **Total % budgeted Effort Usage (%)** | **Sprint 1** | | **Sprint 2** | | **Sprint 3** | |
| **No.** | **%** | **No.** | **%** | **No.** | **%** |
| Requirement | 180$ | 9% | 1 | 3% | 1 | 3% | 1 | 3% |
| Design | 190$ | 9.5% | 2 | 3.16% | 2 | 3.16% | 2 | 3.16% |
| Coding | 190$ | 9.5% | 3 | 3.16% | 3 | 3.16% | 3 | 3.16% |
| Unit Testing | 180$ | 9% | 4 | 3% | 4 | 3% | 4 | 3% |
| Testing | 180$ | 9% | 5 | 3% | 5 | 3% | 5 | 3% |
| Deployment | 180$ | 9% | 6 | 3% | 6 | 3% | 6 | 3% |
| Support for Acceptance Test | 180$ | 9% | 7 | 3% | 7 | 3% | 7 | 3% |
| Project Planning | 180$ | 9% | 8 | 3% | 8 | 3% | 8 | 3% |
| Project monitoring | 180$ | 9% | 9 | 3% | 9 | 3% | 9 | 3% |
| Quality Assurance | 180$ | 9% | 10 | 3% | 10 | 3% | 10 | 3% |
| Training | 180$ | 9% | 11 | 3% | 11 | 3% | 11 | 3% |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Total | 2000$ | 100% |  | 33.33% |  | 33.33% |  | 33.33% |

Table 14: Effort estimation

## 3.3. Schedule

### 3.3.1. Work Breakdown Structure

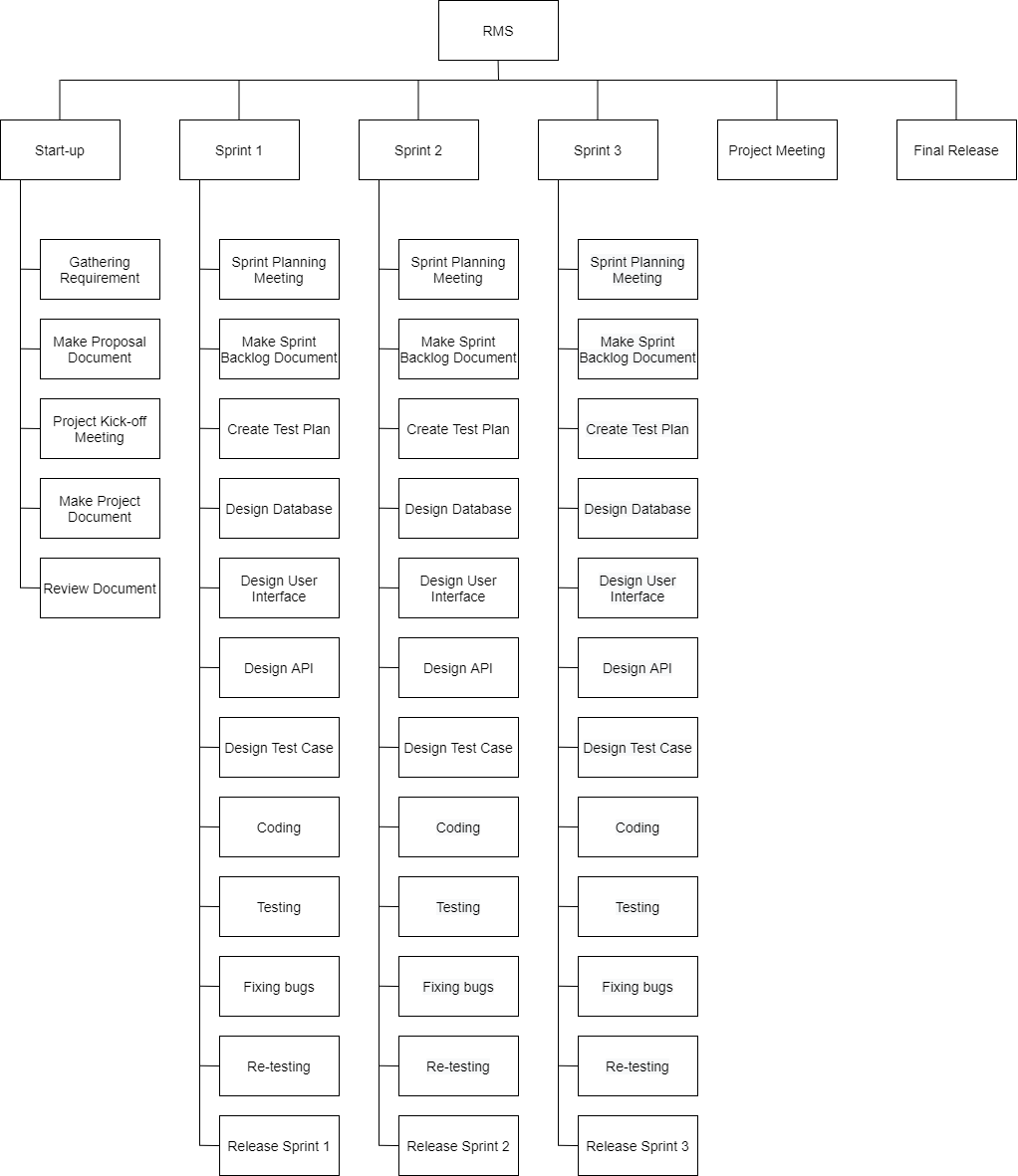


Figure 1: Work Breakdown Structure

### 3.3.2. Detailed Schedule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Task** | **Duration** | **Starting Day** | **Ending Day** | **Person** | **Total Hour** |
| **1** | **Start Up** | **4** | **May 13, 2021** | **May 16, 2021** | **Team** | **32** |
| **1.1** | **Gathering Requirement** | 1 day | May 13, 2021 | May 13, 2021 | Team | 8 |
| **1.2** | **Create Proposal Document** |
| **1.3** | **Project Kick-off Meeting** |
| **1.4** | **Make Project Plan Document** | 1 day | May 14, 2021 | May 14, 2021 | Team | 8 |
| **1.5** | **Make User Story Document** | 1 day | May 15, 2021 | May 15, 2021 | Team | 8 |
| **1.6** | **Make Product Backlog Document** |
| **1.7** | **Review Document** | 1 day | May 16, 2021 | May 16, 2021 | Team | 8 |
| **2** | **Development** | **37 days** | **May 17, 2021** | **June 22, 2021** | **Team** | **296** |
| **2.1** | **Sprint 1** | **12 days** | **May 17, 2021** | **May 28, 2021** | **Team** | **96** |
| **2.1.1** | **Sprint plan meeting** | 1 day | May 17, 2021 | May 17, 2021 | Team | 8 |
| **2.1.2** | **Create Sprint Backlog Document** | 1 day | May 18, 2021 | May 18, 2021 | Team | 8 |
| **2.1.3** | **Create Test Plan Document** | 1 day | May 19, 2021 | May 19, 2021 | Team | 8 |
| **2.1.4** | **Design Database** | 1 day | May 20, 2021 | May 20, 2021 | Team | 8 |
| **2.1.5** | **Design UI** | 1 day | May 21, 2021 | May 21, 2021 | Team | 8 |
| **2.1.6** | **Design API** | 1 day | May 22, 2021 | May 22, 2021 | Team | 8 |
| **2.1.7** | **Design Test case** | 1 day | May 23, 2021 | May 23, 2021 | Team | 8 |
| **2.1.8** | **Coding** | 1 day | May 24, 2021 | May 24, 2021 | Team | 8 |
| **2.1.9** | **Testing** | 1 day | May 25, 2021 | May 25, 2021 | Team | 8 |
| **2.1.10** | **Fix Bugs** | 1 day | May 26, 2021 | May 26, 2021 | Team | 8 |
| **2.1.11** | **Re-Testing** | 1 day | May 27, 2021 | May 27, 2021 | Team | 8 |
| **2.1.12** | **Release Sprint 1** | 1 day | May 28, 2021 | May 28, 2021 | Team | 8 |
| **2.2** | **Sprint 2** | **12 days** | **May 29, 2021** | **June 09, 2021** | **Team** | **96** |
| **2.2.1** | **Sprint plan meeting** | 1 day | May 29, 2021 | May 29, 2021 | Team | 8 |
| **2.2.2** | **Create Sprint 2 Backlog Document** | 1 day | May 30, 2021 | May 30, 2021 | Team | 8 |
| **2.2.3** | **Create Test Plan Document** | 1 day | May 31, 2021 | May 31, 2021 | Team | 8 |
| **2.2.4** | **Design Database** | 1 day | June 01, 2021 | June 01, 2021 | Team | 8 |
| **2.2.5** | **Design UI** | 1 day | June 02, 2021 | June 02, 2021 | Team | 8 |
| **2.2.6** | **Design Test Case** | 1 day | June 03, 2021 | June 03, 2021 | Team | 8 |
| **2.2.7** | **Design API** | 1 day | June 04, 2021 | June 04, 2021 | Team | 8 |
| **2.2.8** | **Coding** | 1 day | June 05, 2021 | June 05, 2021 | Team | 8 |
| **2.2.9** | **Testing** | 1 day | June 06, 2021 | June 06, 2021 | Team | 8 |
| **2.2.10** | **Fix bug** | 1 day | June 07, 2021 | June 07, 2021 | Team | 8 |
| **2.2.11** | **Re-Testing** | 1 day | June 08, 2021 | June 08, 2021 | Team | 8 |
| **2.2.12** | **Release Sprint 2** | 1 day | June 09, 2021 | June 09, 2021 | Team | 8 |
| **2.3** | **Sprint 3** | **13 days** | **June 10, 2021** | **June 22, 2021** | **Team** | **104** |
| **2.3.1** | **Sprint plan meeting** | 1 day | June 10, 2021 | June 10, 2021 | Team | 8 |
| **2.3.2** | **Create Sprint Backlog Document** | 1 day | June 11, 2021 | June 11, 2021 | Team | 8 |
| **2.3.3** | **Create Test Plan Document** | 1 day | June 12, 2021 | June 12, 2021 | Team | 8 |
| **2.3.4** | **Update Database** | 1 day | June 13, 2021 | June 13, 2021 | Team | 8 |
| **2.3.5** | **Design UI** | 1 day | June 14, 2021 | June 14, 2021 | Team | 8 |
| **2.3.6** | **Design Test Case** | 1 day | June 15, 2021 | June 15, 2021 | Team | 8 |
| **2.3.7** | **Design API** | 1 day | June 16, 2021 | June 16, 2021 | Team | 8 |
| **2.3.8** | **Coding** | 2 day | June 17, 2021 | June 18, 2021 | Team | 16 |
| **2.3.9** | **Testing** | 1 day | June 19, 2021 | June 19, 2021 | Team | 8 |
| **2.3.10** | **Fix bug** | 1 day | June 20, 2021 | June 20, 2021 | Team | 8 |
| **2.3.11** | **Re-test** | 1 day | June 21, 2021 | June 21, 2021 | Team | 8 |
| **2.3.12** | **Release Sprint 3** | 1 day | June 22, 2021 | June 22, 2021 | Team | 8 |
| **3** | **Project’s Meeting** | **1 day** | **June 23, 2021** | **June 23, 2021** | **Team** | **8** |
| **4** | **Final release** | **1 day** | **June 24, 2021** | **June 24, 2021** | **Team** | **8** |

Table 15: Detailed Schedule

### 3.3.3. Project Schedule

* The detail project schedule is available in here The Project Schedule is weekly updated by the Project Manager. Below table is just an example! (sprint backlog)

## 3.4. Resource

* Specified as in the section Project Team

## 3.5. Infrastructure

|  |  |  |  |
| --- | --- | --- | --- |
| **Work/Product** | **Purpose** | **Expected**  **Availability by** | **Note** |
| **Development Environment** | | | |
| NT Server | Operating System | Initiation stage |  |
| Mainframe | Operating System |  |  |
| Win NT | Operating System |  |  |
| DB2 | DBMS |  |  |
| PHP | Development language for Web interface |  |  |
| C# | Development language for |  |  |
| **Hardware & Software** | | | |
| 1GB space on server |  |  |  |
| Rational Rose | Design |  |  |
| **Other Tools** | | | |
| CVS | Source version control | Definition stage |  |
| Nunit | Unit Test | Construction stage |  |
| DMS | Defect logging and tracking | Definition stage |  |
| Timesheet | Effort logging | Initiation stage |  |
| FI | Project management tool | Initiation stage |  |
| MS Project | Task tracking | Initiation stage |  |

Table 16: Infrastructure

## 3.6. Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| **Technical** | | | |
| C# Language | Team | 7 days | If already trained |
| VueJS framework | Team | 8 hrs | If already trained |
| PHP Language / Laravel Framework | Team | 8 hrs | If already trained |
| HTML / CSS | Team | 4 hrs | If already trained |
| **Business domain** | | | |
| Banking | Team | 7 days |  |
| **Process** | | | |
| Quality system | Team | 3 hrs | If already trained |
| Configuration management | Team | 2 hrs | If already trained for  CC. For others, on-the- job training |
| Group review | Team | 4 hrs | If already trained |
| Defect prevention | Team | 4.5 hrs | Mandatory |
| SPC tool | Team | 4.5 hrs | If already trained |
| RUP methodology | Team | 2 hrs | Mandatory |

Table 17: Training Plan

## 3.7. Finance

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Item** | **Total Budget** | **%**  **Budget** | **Budget in Period** | | | | | | | | **Note** |
| **W1**  **May** | **W2**  **May** | **W3**  **May** | **W4**  **May** | **W1**  **Jun** | **W2**  **Jun** | **W3**  **Jun** | **W4**  **Jun** |  |
| Purchases (COTS) | 285$ | 14.25% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.79% |  |
| Team building | 285$ | 14.25% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.79% |  |
| Tools | 285$ | 14.25% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.79% |  |
| Travel costs | 285$ | 14.25% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.79% |  |
| Training | 285$ | 14.25% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.79% |  |
| Review activities | 285$ | 14.25% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.75% | 1.79% |  |
| Other | 290$ | 14.5% | 1.8% | 1.8% | 1.8% | 1.8% | 1.8% | 1.8% | 1.8% | 1.9% |  |
| Total | 2000$ | 100% | 12.3% | 12.3% | 12.3% | 12.3% | 12.3% | 12.3% | 12.3% | 12.64% |  |

Table 18: Finance

**4. PROJECT ORGANIZATION**

**4.1. Organization Structure**

**4.2. Project Team**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role** | **Responsibility** | **Qualification** | **Full name** | **Type** | **%**  **Effort** |
| Senior Manager | * Provide resource & funding * Approve Project plan * Review project status * Resolve escalated issues * Project financial plan | Manage | Khue | Training | 90% |
| PM | * Have overall responsibility of the project * Project planning and scheduling * Task assignment and tracking * Reporting * Ensure delivery as per contract * Interface with other departments as per need * Customer interaction * Ensure open issues/customer complaints are closed properly | Manage | Khue | Onsite/ Offshore/Training | 90% |
| Business Analyst | * Requirement development * Requirement analysis | Analysis | Thanh, Dat | Training | 90% |
| Designer | * Architectural design | Design | Phu, Tri | Training | 90% |
| **Role** | **Responsibility** | **Qualification** | **Full name** | **Type** | **% Effort** |
| Develop ment Leader #1 | If the Project Manager has appointed Project Technical Leader (sync.: Development Project Manager), who is only responsible for the technical project execution, this should also be specified | Manage | Phu | Training | 90% |
| Developer #1 | Work mainly on C# | Analysis | Thanh | Training | 90% |
| Develop ment Leader #2 | Work mainly on Website | Manage | Khue | Training | 90% |
| Developer #2 | Work mainly on Website | Analysis | Tri | Training | 90% |
| Test Leader | Work mainly on Website | Analysis | Dat | Training | 90% |
| Tester #1 | Design test case and execute test module A, B | Analysis | Tri | Training | 90% |
| Tester #2 | Design test case and execute test module C,D | Analysis | Dat | Training | 90% |

Table 19: Project Team

# 5. COMMUNICATION & REPORTING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Type** | **Method / Tool** | **When** | **Information** | **Participants / Responsible** |
| **Project Task Tracking** | | | | |
| Task scheduling | MS Project | At the beginning of every stage, and weekly  Refinement and rescheduling as necessary |  | Project Mgr(s) |
| Task assignment | In Excel file and via project weekly meeting | Weekly |  | Project leader, Technical |
| **Project Meeting** | | | | |
| Kick-off Meeting | Face to face | Initiation stage | Project introduction; Project plan review; Risk identification; Obtainment of commitment of relevant stakeholders | Project Mgr(s),  Project Senior Manager, Project Team Members, QA |
| Project Progress Review Meetings | Face to face | Weekly and on event | Communicate project status  Communicate and resolve any open issue, risks, and changes  Discuss any suggested improvement | Project Mgr(s),  Project Team Members |
| Milestone Meetings | Face to face | Before milestones | Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage | Project Mgr(s),  Project Senior Manager, Project Team Members, QA |
| **Communication Type** | **Method / Tool** | **When** | **Information** | **Participants / Responsible** |
| Project Post-mortem Meeting | Face to face | Termination stage | Wrap-up  Evaluate project performance; Team performance; share experiences | Project Mgr(s), Project Senior Manager, Project Team Members, QA |
| Transfer/Sharing of project documentation/information | <Shared Project Repository/FT P/CVS/MS  Share Point Server> | When available | All project documentation and information | Project Mgr(s) Project Team Members, QA |
| **Customer Communication and Reporting:** | | | | |
| Project Report | Agreed standard format between company and customer | <5pm Monday, Weekly> | Project status report, Issue requiring clarifications, escalation, if any | Project Manager Sub-Project Managers |
| Project Meetings with customer | Teleconferenc e /TV Meeting | <2pm Tuesday, Weekly> | As above | Project Manager |
| Requirement gathering/clarification | Email/TV meeting/Face to face meeting | During requirement analysis phase | As in Q&A list | Project Manager Business Analyst |
| **Communication with Senior Management** | | | | |
| Review Project Plan & Project schedule | By email or attend project meeting | Significant changes to WO, PP and Project schedule (scope, objectives Organization, HR, major milestone, deliverables ) |  | Project Mgr |
| Project Progress Review | By email and/or via Operation meeting -at Group/Divisio n level | Weekly | Project status report, Issue requiring clarifications, escalation, if any | Project Mgr |
| Project Milestone Review | By email and via project milestone review meeting | End of every stage | Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage | Project Mgr |
| **Other Communication and Reporting:** | | | | |
| Raise issue or request service/support of BA groups (IT, Admin, QA, HR, Training, Recruitment,etc) | Call log; email; phone | Upon request | Request content, expected completion date | Project Manager |

Table 20: Communication & Reporting

# 6. CONFIGURATION MANAGEMENT

* Refer to the CM plan

# 7. SECURITY ASPECTS

* References

|  |  |  |
| --- | --- | --- |
| **No.** | **References** | **Document Information** |
| 1 | **Scrum Process** | https://www.scrum.org/resources/what-is-scrum |
| 2 | **Technical** | https://stackoverflow.com/ |
| https://topdev.vn/blog/restful-api-la-gi |
| https://vuejs.org/ |
| https://github.com/ |
| https://laravel.com/ |
| https://docs.microsoft.com/en-us/dotnet/desktop/winforms |
| https://www.mysql.com/ |
| 3 | **Information** | https://ipos.vn/top-phan-mem-quan-ly-nha-hang/ |

Table 21: References

# DEFINITIONS AND ACRONYMS

|  |  |  |
| --- | --- | --- |
| **Acronym** | **Definition** | **Note** |
| PM | Project Manager |  |
| PTL | Project Technical Leader |  |
| QA | Quality Assurance Officer |  |
| CC | Infrastructure Configuration Controller |  |
| DV | Developer |  |
| URD | User Requirement Document |  |
| SRS | Software Requirement Specification |  |
| ADD | Architecture Design Document |  |
| DDD | Detail Design Document |  |
| TP | Test Plan |  |
| TC | Test Case |  |
| SC | Source Code |  |
| CM | Configuration Management |  |
| CSCI | Computer Software Configuration Items |  |
| CI | Configuration Item |  |
| CCB | Change Control Board |  |

Table 22: Definitions And Acronyms