

**International School**

**Capstone Project 2**

**CMU-SE 451 – C2SE.12**

**Project Plan**

**Version 1.0**

**Date: March 31th, 2021**

**Learn English Together**

**Submitted by**

**Ha, Le Thanh**

**Hieu, Le Xuan**

**My, Ngo Ngoc**

**Thong, Doan Trung**

**Approved by**

**MSc Huy, Truong Dinh**

**Proposal Review Panel Representative:**

Name Signature Date

**Capstone Project 2- Mentor:**

Name Signature Date

**PROJECT INFORMATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project acronym** | LET | | |
| **Project Title** | Learn English Together | | |
| **Start Date** | 26 Feb 2021 | **End Date** | 08 Jun 2021 |
| **Lead Institution** | International School, Duy Tan University | | |
| **Project Mentor** | MSc Huy, Truong Dinh | | |
| **Scrum master / Project Leader & contact details** | Ha, Le Thanh  Email: lethanhhadtu@gmail.com  Tel: 0334002818 | | |
| **Partner Organization** | Duy Tan University | | |
| **Project Web URL** |  | | |
| **Team members** | Name | Email | Tel |
|  | Ha, Le Thanh | lethanhhadtu@gmail.com | 0334002818 |
|  | Hieu, Le Xuan | xuanhieu.le.1999@gmail.com | 0399706614 |
|  | My, Ngo Ngoc | ngongocmy851999@gmail.com | 0764497391 |
|  | Thong, Doan Trung | doanthong002@gmail.com | 0886428208 |

**RECORD OF CHANGE**

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Effective Date** | **Changed Item** | **A\* M, D** | **Change Description** | **Reason for Change** | **Revision Number** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**TABLE OF CONTENTS**

[1. PROJECT OVERVIEW 5](#_Toc70463344)

[***1.1.* Project Description 5**](#_Toc70463345)

[***1.2.* Scope and Purpose 5**](#_Toc70463346)

[**1.2.1. Purpose 5**](#_Toc70463347)

[**1.2.2. Scope 6**](#_Toc70463348)

[***1.3.* Assumptions and Constraints 6**](#_Toc70463349)

[***1.4.* Project Objectives 7**](#_Toc70463350)

[**1.4.1. Standard Objectives 7**](#_Toc70463351)

[**1.4.2. Specific Objectives 7**](#_Toc70463352)

[***1.5.* Project Risk 8**](#_Toc70463353)

[2. PROJECT DEVELOPMENT APPROACH 10](#_Toc70463354)

[***2.1.* Technical Process 10**](#_Toc70463355)

[**2.1.1. Reasons for selecting 10**](#_Toc70463356)

[**2.1.2. Agile Methodology 10**](#_Toc70463357)

[***2.2.* Quality Management 13**](#_Toc70463358)

[**2.2.1. Strategy for Meeting Quality Objectives 13**](#_Toc70463359)

[**2.2.2. Quality Control 13**](#_Toc70463360)

[**2.2.3. Measurements Program 13**](#_Toc70463361)

[***2.3.* Unit Testing Strategy 14**](#_Toc70463362)

[***2.4.* System Testing Strategy 15**](#_Toc70463363)

[3. ESTIMATION 15](#_Toc70463364)

[***3.1.* Schedule 15**](#_Toc70463365)

[***3.2.* Resource 16**](#_Toc70463366)

[***3.3.* Infrastructure 16**](#_Toc70463367)

[***3.4.* Training Plan 17**](#_Toc70463368)

[***3.5.* Finance 18**](#_Toc70463369)

[4. PROJECT ORGANIZATION 20](#_Toc70463370)

[***4.1.* Organization Structure 20**](#_Toc70463371)

[***4.2.* Project Team 20**](#_Toc70463372)

[5. COMMUNICATION & REPORTING 23](#_Toc70463373)

[6. SECURITY ASPECTS 25](#_Toc70463374)

[7. REFERENCES 25](#_Toc70463375)

1. PROJECT OVERVIEW

Project Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Project code** | LET | **Contract type** | Time material |
| **Customer** | Student, Volunteer | **2nd Customer** |  |
| **Project Level** | Group | **Project rank** | A |
| **Group** | C2SE.12 | **Division** | International School |
| **Project Type** | Internal | **Project Manager/ Scrum master** | Le Thanh Ha |
| **Project Category** | Development | **Business domain** |  |
| **Application type** | Web application |  |  |

Scope and Purpose

1.2.1. Purpose

* + - * The purpose of the Project Plan is gathering all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by managers to direct the development effort.
      * The following people use the Project 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.
      * This document defines the approach to be used by the Project team to deliver the intended project management scope of the project.
      * This document contains the details required to execute the project successfully. Once project execution begins, this plan will be reviewed, baseline, and updated on the regular basis.
    1. Scope
       - This document describes the overall plan to be used by the Attendance and Assessment Support System, including deployment of the product. The details of the individual iterations will be described in the Release Plans.
       - This document will schedule the project plan into Work Breakdown Structure to let Product Owner easily review and manage Scrum team working and delivery on time.
       - The plans as outlined in this document are based upon the product requirements as defined in the User Stories document.

Assumptions and Constraints

|  |  |  |
| --- | --- | --- |
| **No** | **Description** | **Note** |
| **Assumptions** | | |
| 1 | Migration to XX for Java 3.0 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 |

Project Objectives

* + 1. Standard Objectives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metrics** | **Unit** | **Committed** | **Re-committed** | **Note** |
| Start Date | dd-mm-yy | 02/03/21 | 02/03/21 |  |
| End Date | dd-mm-yy | 02/06/21 | 02/06/21 |  |
| Duration | Elapsed days | 67 days | 67 days |  |
| Maximum Team Size | Person | 4 | 4 |  |
| Billable Effort | Person-day | 4 hours | 4 hours |  |
| Calendar effort | Person-day | 4 hours | 4 hours |  |
| Effort Usage | Person-day | 4 hours | 1. hours |  |

* + 1. Specific Objectives

This system has been developed to enable schools to manage the quality of student training through class assessments and by faculty in pre-written evaluation forms. Besides, it also helps lecturers or event organizers to check student attendance quickly and save time. After evaluation, the forms will be aggregated and automatically reported by the statistical system on the satisfaction level as well as the quality of the lesson or event held.

Project Risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identify Risk** | **Descriptions** | **Probability** | **Impact** | **Mitigations** |
| Scope definition | Scope changes may arise during the project. The redundant scope may be  discovered. | 2 | 4 | Analysis and evaluation scope. |
| Requirement definition | Internal inconsistencies may exist within requirements that may be missing from formal requirement  specification. | 2 | 4 | Unifying requirements before proceeding with the  analysis. |
| Resources | Key resources may be unavailable when required. Specific skills may not be available  when required. | 2 | 3 | Training |
| Communication | The mentor’s requirement may be misunderstood. Project reporting needs may change during the project. | 2 | 3 | Having analyzed the requirement, the mentor should review to ensure that requirements are not  misunderstood. |
| Health | Health may change during the project. An accident or incident may occur delaying the project. | 2 | 3 | The project plan has backup time to ensure that if there are any health risks, they will not interrupt the  project. |
| Time | The project may not be finished on time. | 2 | 4 | Many of the  functions offered are of |

1. PROJECT DEVELOPMENT APPROACH

Technical Process

**SCRUM process.**

* + 1. Reasons for selecting
* Scrum is an iterative and incremental agile software development framework for

managing software projects and product or application development.

* Scrum focuses on project management institutions where it is difficult to plan ahead.
* Mechanisms of empirical process control, where feedback loops that constitute

the core management technique is used as opposed to traditional command-and

-control management.

* Its approach to planning and managing projects is by bringing decision-making

authority to the level of operation properties and certainties.

* + 1. Agile Methodology

The Agile software development methodology is one of the simplest and effective processes to turn a vision for a business need into software solutions. Agile is a term used to describe software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to change.

The agile software development emphasizes on four core values:

Individual and team interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

***2.1.2.1. Scrum Process***

**Scrum Definition**

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. A Product Owner orders the work for a complex problem into a Product Backlog.

2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.

3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.

4. Repeat Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.

Various processes, techniques and methods can be employed within the framework. Scrum wraps around existing practices or renders them unnecessary. Scrum makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

**Scrum Role:**

1. **Scrum Master**: The Scrum Master helps the product group learn and apply Scrum to achieve business value. The Scrum Master does whatever is in their power to help the Team, Product Owner and organization be successful. The Scrum Master is not the manager of the Team members, nor are they a project manager, team lead, or team representative. Instead, the Scrum Master serves the Team; he or she helps to remove impediments, protects the Team from outside interference, and helps the Team to adopt Agile development practices. He or she educates, coaches and guides the Product Owner, Team and the rest of the organization in the skillful use of Scrum.
2. **Product Owner**: The Product Owner is responsible for maximizing return on investment (ROI) by identifying product features, translating these into a prioritized list, deciding which should be at the top of the list for the next Sprint, and continually re-prioritizing and refining the list. The Product Owner has profit and loss responsibility for the product, assuming it is a commercial product. Product Owner in Agile is like a spokesperson for customer and needs to represent them.
3. **Development team**: A Development Team is a collection of individuals working together to develop and deliver the requested and committed product increments. It comprises of cross-functional members who are capable of achieving the sprint goals. This could include software engineers, architects, programmers, analysts, system admins, QA experts, testers, UI designers, etc.

**Advantages of Scrum:**

* 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.
* 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.

Quality Management

* + 1. 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 |

* + 1. Quality Control

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Review Item** | | | | **Type of Review** | **Reviewer** | | **When** | | |
| <work product to be reviewed> | | | | Group review or One-person review | List out people participate in review | MUST | Entry criteria or trigger to perform the review | | |
| Project plan Project schedule  CM Plan | | | | Group review Group review  One-person review | Senior Manager, QA, PTLs, Customers | | End of Initiation stage | | |
| Business analysis and requirements specification document, Use Case catalog | | | | Group review |  | | End of requirements | 90% | of |
| Design document, object model | | | | Group review |  | | End of 90% design | | |
| Stage plans | | | | One-person review |  | | Beginning of each stage | | |
| Complex/first specs incl. diagrams | time test | generated cases, | program interactive | Group review |  | | End of detailed design | | |
| Code | | | | Group review |  | | After coding for first few programs | | |
|  | | | |  |  | |  | | |

* + 1. Measurements Program

|  |  |  |  |
| --- | --- | --- | --- |
| **Data to be**  **collected** | **Purpose** | **Responsible** | **When** |
| Size: No. of  KLOC// FP | Calculate the money of the project | PM/SM | At the end of  stages |
| Effort: No. person-day | Predicting the most realistic amount of effort required to develop or maintain software based on  incomplete, uncertain and noisy input | Team members | Daily |
| Quality: No. defects detected | Identify the root causes of defects and improve the process to avoid introducing defects, which help to improve the quality of the software  product | Reviewer, Tester | Right after the review/test |
| Schedule | Achieve the goals and priorities in the time that have available | PM/SM | Weekly and at the end of  stages |

Unit Testing Strategy

The system is tested using Unit Testing, a testing technique that uses individual modules tested to determine if there are any problems caused by the developer himself. It is concerned with the functional correctness of the independent modules.

And the project will be tested by the black box and white box technique of unit testing:

* Black Box Testing - Using which the user interface, input and output are tested.
* White Box Testing - used to test each one of those functions’ behaviors is tested.

**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

System Testing Strategy

System Testing (ST) is a black-box testing technique performed to evaluate the complete system's compliance against specified requirements. System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non- Functional testing.

# Completion criteria:

* 95% of the test cases must be executed
* Can release when only 5% of test cases remain without affecting the project

1. ESTIMATION

Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Task Name** | **Duration** | **Start** | **Finish** |
| **1.** | **Initial** | **8 days** | **26 – Feb - 2021** | **06– Feb – 2021** |
| 1.1 | Gathering Requirement | 2 days | 26 – Feb – 2021 | 28 – Feb – 2021 |
| 1.2 | Create Proposal Document | 6 days | 01 – Mar – 2021 | 06 – Mar – 2021 |
| **2** | **Start Up** | **8 days** | **07 – Mar – 2021** | **14 – Mar – 2021** |
| 2.1 | Project Kick-off Meeting | 2 days | 07 – Mar – 2021 | 08 – Mar – 2021 |
| 2.2 | Create Document | 6 days | 09 – Mar – 2021 | 14 – Mar – 2021 |
| **3** | **Development** | **84 days** | **15 – Mar – 2021** | **01 – Jun– 2021** |
| 3.1 | Sprint 1 | 28 days | 15 – Mar – 2021 | 12 – Apr – 2021 |
| 3.2 | Sprint 2 | 28 days | 13 – Apr – 2021 | 10 – May – 2021 |
| 3.3 | Sprint 3 | 21 days | 11 – May – 2021 | 01 – Jun – 2021 |
| **4** | **Project’s Retrospective Meeting** | **03 days** | **05 – Jun – 2021** | **07 – Jun – 2021** |
| **5** | **Final Release** | **01 days** | **08 – Jun – 2021** | 1. **– Jun – 2021** |

**DETAIL SCHEDULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Task Name | Start | Finish | Effort |
| **1** | **Initial** | **26/02** | **06/03** | **21 hrs** |
| **1.1** | **Gathering Requirement** | **26/02** | **28/03** | **8 hrs** |
| 1.1.1 | Get requirement from Mentor MSc Huy Truong Dinh | 26/02 | 27/02 | 2 hrs |
| 1.1.2 | Analyzing requirement | 27/02 | 28/02 | 6 hrs |
| **1.2** | **Create Proposal Document** | **01/03** | **06/03** | **16 hrs** |
| 1.2.1 | Product Definition | 01/03 | 02/03 | 3 hrs |
| 1.2.2 | Business Need | 02/03 | 03/03 | 2 hrs |
| 1.2.3 | Prior Art | 03/03 | 04/03 | 3 hrs |
| 1.2.4 | Proposed Solution | 04/03 | 05/03 | 3 hrs |
| 1.2.5 | Master Plan | 05/03 | 06/03 | 5 hrs |
| 2 | **Start Up** | 07/03 | 14/03 | 44 hrs |
| 2.1 | Project kick-off | 07/03 | 08/03 | 2 hrs |
| **2.2** | **Create Document** | **08/03** | **14/03** | **42 hrs** |
| 2.2.1 | Project’s Meeting | 08/03 | 08/03 | 3 hrs |
| 2.2.2 | Create User Stories | 09/03 | 09/03 | 6 hrs |
| 2.2.3 | Create Product Backlog | 10/03 | 10/03 | 10 hrs |
| 2.2.4 | Review Document | 11/03 | 11/03 | 5 hrs |
| 2.2.5 | Create Project Plan | 12/03 | 12/03 | 8 hrs |
| 2.2.6 | Create Architecture Design Document | 13/03 | 13/03 | 08 hrs |
| 2.2.7 | Review Document | 14/03 | 14/03 | 2 hrs |
| **3** | **Development** |  |  |  |
| **3.1** | **Sprint 1** | 15/03 | 12/04 | **103 hrs** |
| 3.1.1 | Sprint Planning Meeting | 15/03 | 15/03 | 4 hrs |
| 3.1.2 | Create Sprint Backlog | 15/03 | 15/03 | 2 hrs |
| 3.1.3 | Create Test Plan document for Sprint 1 | 16/03 | 16/03 | 2 hrs |
| 3.1.4 | Create Database document for Sprint 1 | 16/03 | 16/03 | 2 hrs |
| **3.1.5** | **Design** | 17/03 | 25/03 | **21 hrs** |
| 3.1.5.1 | Design interface of Home Page | 17/03 | 18/03 | 5 hrs |
| 3.1.5.2 | Design interface of Register | 18/03 | 18/03 | 4 hrs |
| 3.1.5.3 | Design interface of Login | 18/03 | 18/03 | 4 hrs |
| 3.1.5.4 | Design interface of profile | 19/03 | 19/03 | 4 hrs |
| 3.1.5.5 | Design interface of chat room group | 20/03 | 20/03 | 4 hrs |
| 3.1.5.6 | Design interface of list group | 21/03 | 21/03 | 4 hrs |
| 3.1.5.7 | Design interface of member group | 22/03 | 22/03 | 4 hrs |
| 3.1.5.8 | Design interface of call video group | 23/03 | 23/03 | 4 hrs |
| 3.1.5.9 | Design interface of record group | 24/03 | 24/03 | 4 hrs |
| 3.1.5.10 | Design interface of message | 25/03 | 25/03 | 4 hrs |
| **3.1.6** | **Coding** | **26/03** | **09/04** | **58 hrs** |
| 3.1.6.1 | Code Main Menu for users | 26/03 | 28/03 | 16 hrs |
| 3.1.6.2 | Code Login for users | 29/03 | 30/03 | 4 hrs |
| 3.1.6.3 | Code Register for users | 30/03 | 31/03 | 4 hrs |
| 3.1.6.4 | Code Profile for users | 01/04 | 02/04 | 10 hrs |
| 3.1.6.5 | Code Group for users | 03/04 | 06/04 | 10 hrs |
| 3.1.6.6 | Code Message | 07/04 | 09/04 | 10 hrs |
| **3.1.7** | **Testing & Fix Bug** | 09/04 | 11/04 | **10 hrs** |
| 3.1.7.1 | Test Main Menu | 09/04 | 09/04 | 2 hrs |
| 3.1.7.2 | Test Login | 09/04 | 09/04 | 2 hrs |
| 3.1.7.3 | Test Register | 10/04 | 10/04 | 2 hrs |
| 3.1.7.4 | Test profile | 10/04 | 10/04 | 4 hrs |
| 3.1.7.5 | Test group | 11/04 | 11/04 | 4 hrs |
| 3.1.7.6 | Test message | 11/04 | 11/04 | 4 hrs |
| **3.1.8** | **Release Sprint 1** | **11/04** | **12/04** | **4 hrs** |
| 3.1.8.1 | Sprint 1 Review Meeting | 11/04 | 11/04 | 2 hrs |
| 3.1.8.2 | Sprint 1 Retrospective | 12/04 | 12/04 | 2 hrs |
| **3.2** | **Sprint 2** | **13/04** | **10/05** | **185 hrs** |
| 3.2.1 | Sprint Planning Meeting | 13/04 | 13/04 | 4 hrs |
| 3.2.2 | Create Sprint Backlog | 13/04 | 13/04 | 2 hrs |
| 3.2.3 | Create Test Plan document for Sprint 2 | 14/04 | 14/04 | 2 hrs |
| 3.2.4 | Create Database document for Sprint 2 | 14/04 | 14/04 | 2 hrs |
| **3.2.5** | **Design** | 15/04 | 20/04 | **40 hrs** |
| 3.2.5.1 | Design user interface of Find a helper | 15/04 | 16/04 | 8 hrs |
| 3.2.5.2 | Design user interface of Support for user | 16/04 | 17/04 | 8 hrs |
| 3.2.5.3 | Design user interface of Search user | 17/04 | 18/04 | 8 hrs |
| 3.2.5.4 | Design user interface of Learn Vocabulary | 18/04 | 19/04 | 8 hrs |
| 3.2.5.5 | Design user interface of Quiz | 19/04 | 20/04 | 8 hrs |
| **3.2.6** | **Coding** | 20/04 | 06/05 | **99 hrs** |
| 3.2.6.1 | Code Find a helper | 20/04 | 21/04 | 18 hrs |
| 3.2.6.2 | Code Support for user | 21/04 | 23/04 | 18 hrs |
| 3.2.6.3 | Code Search | 24/04 | 26/04 | 16 hrs |
| 3.2.6.4 | Code Learn Vocabulary | 27/04 | 02/05 | 25 hrs |
| 3.2.6.5 | Code Quiz | 03/05 | 06/05 | 22 hrs |
| **3.2.7** | **Testing & Fix Bug** | 07/05 | 08/05 | **32 hrs** |
| 3.2.7.1 | Test Find a helper | 07/05 | 07/05 | 8 hrs |
| 3.2.7.2 | Test Support for user | 07/05 | 07/05 | 5 hrs |
| 3.2.7.3 | Test Case for Search | 08/05 | 08/05 | 5 hrs |
| 3.2.7.4 | Test Learn Vocabulary | 08/05 | 08/05 | 7 hrs |
| 3.2.7.5 | Test Quiz | 08/05 | 08/05 | 7 hrs |
| **3.2.8** | **Release Sprint 2** | 09/05 | 10/05 | **4 hrs** |
| 3.2.8.1 | Sprint 2 Review Meeting | 09/05 | 09/05 | 2 hrs |
| 3.2.8.2 | Sprint 2 Retrospective | 10/05 | 10/05 | 2 hrs |
| **3.3** | **Sprint 3** | **11/05** | **01/06** | **119 hrs** |
| 3.3.1 | Sprint Planning Meeting | 11/05 | 11/05 | 4 hrs |
| 3.3.2 | Create Sprint Backlog | 11/05 | 11/05 | 2 hrs |
| 3.3.3 | Create Test Plan document for Sprint 3 | 12/05 | 12/05 | 2 hrs |
| 3.3.4 | Create Database document for Sprint 3 | 12/05 | 12/05 | 2 hrs |
| **3.3.5** | **Design** | **13/05** | **17/05** | **30 hrs** |
| 3.3.5.1 | Design interface of Dashboard | 13/05 | 14/05 | 8 hrs |
| 3.3.5.2 | Design interface of Manager account | 14/05 | 15/05 | 5 hrs |
| 3.3.5.3 | Design interface of Admin | 15/05 | 16/05 | 10 hrs |
| 3.3.5.4 | Design interface of Event | 16/05 | 17/05 | 7 hrs |
| **3.3.6** | **Coding** | **18/05** | **28/05** | **59 hrs** |
| 3.3.6.1 | Code Dashboard | 18/05 | 20/05 | 12 hrs |
| 3.3.6.2 | Code Admin | 21/05 | 23/05 | 15 hrs |
| 3.3.6.3 | Code Manager account | 23/05 | 25/05 | 22 hrs |
| 3.3.6.4 | Code Event | 26/05 | 28/05 | 10 hrs |
| **3.3.5** | **Testing & fix Bug** | **29/05** | **30/05** | **16 hrs** |
| 3.3.7.1 | Test Dashboard | 29/05 | 29/05 | 4 hrs |
| 3.3.7.2 | Test Admin | 29/05 | 29/05 | 4 hrs |
| 3.3.7.3 | Test Manager account | 30/05 | 30/05 | 4 hrs |
| 3.3.7.4 | Test Event | 30/05 | 30/05 | 4 hrs |
| **3.3.6** | **Release Sprint 3** | **31/05** | **01/06** | **4 hrs** |
| 3.3.6.1 | Sprint 3 Review Meeting | 31/05 | 31/05 | 2 hrs |
| 3.3.6.2 | Sprint 3 Retrospective | 01/06 | 01/06 | 2 hrs |

Resource

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Phone** | **Email** | **Position** |
| MSc Huy, Truong Dinh | 0982132352 | huy.truongdinh@gmail.com | Mentor |
| Ha, Le Thanh | 0334002818 | lethanhhadtu@gmail.com | Scrum Master and Member |
| Hieu, Le Xuan | 0399706614 | xuanhieu.le.1999@gmail.com | Member |
| My, Ngo Ngoc | 0764497391 | ngongocmy851999@gmail.com | Member |
| Thong, Doan Trung | 0886428208 | doanthong002@gmail.com | Member |

Infrastructure

|  |  |  |  |
| --- | --- | --- | --- |
| **Work/Product** | **Purpose** | **Expected**  **Availability by** | **Note** |
| **Development Environment** | | | |
| Linux | Operating System | Initiation stage |  |
| Window 10 | Operating System | Initiation stage |  |
| PostgreSQL | DBMS | Construction stage |  |
| React.js | Development language for Web  interface | Construction stage |  |
| Nodejs | Development language for Server  side/ Backend | Construction stage |  |
| **Other Tools** | | | |
| Github | Source version control | Definition stage |  |
| Postman | API Test | Construction stage |  |
| Docker | Building, Deploying and Running | Construction stage |  |
| Slack | Division of work | Initiation stage |  |
| Trello | Task tracking | Initiation stage |  |

Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| **Technical** | | | |
| JavaScript Language | All members | 1 hrs. | Mandatory |
| NodeJs Language | All members | 1 hrs. |  |
| **Process** | | | |
| Quality system | All members | 1 hr. | If already trained |
| Configuration management | All members | 1 hr. | If already trained |
| Group review | All members | 1 days | Mandatory |
| Defect prevention | All members | 1 hr. | Mandatory |
| RUP methodology | All members | 1 hr. | Mandatory |

Finance

|  |  |  |
| --- | --- | --- |
| **Category** | **Detailed** | **Description** |
| Start date | March 2nd, 2021 | The start date of the project. |
| End date | Jun 2nd, 2021 | The end date of project. |
| Duration (1) | 77 | Total day of project. |
| Working time (2) | 4 hours/day | In one day and for one member. |
| Total effort (3) = (1) \* (2) \* 4 | 1232 hrs. | For four team members and the  entire project. |
| Other cost (4) = 100 \* 4 | $400 | For four team members with party |
| Labor cost (5) = (3) \* 2 + (4) | $2864 | For four team members and the  entire project. ($2.0/ member) |

1. PROJECT ORGANIZATION

Organization Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Phone** | **Email** | **Position** |
| MSc Huy, Truong Dinh | 0982132352 | huy.truongdinh@gmail.com | Mentor |
| Ha, Le Thanh | 0334002818 | lethanhhadtu@gmail.com | Scrum Master and Member |
| Hieu, Le Xuan | 0399706614 | xuanhieu.le.1999@gmail.com | Member |
| My, Ngo Ngoc | 0764497391 | ngongocmy851999@gmail.com | Member |
| Thong, Doan Trung | 0886428208 | doanthong002@gmail.com | Member |

Project Team

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Role** | **Responsibility** | **Full name** | **Type** | **Effort** | **Start**  **date** | **End**  **date** |
| Product owner | * Approve Project plan * Review project status * Resolve escalated   issues | Ha, Le Thanh | Onsite | 80% | 02/03/21 | 02/06/21 |
|  | - Project financial  plan |  |  |  |  |  |
| Scrum Master | 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 | Ha, Le Thanh | Onsite | 100% | 02/03/21 | 02/06/21 |
| Designer | UI design | My, Ngo Ngo | Onsite | 100% | 02/03/21 | 02/06/21 |
| Coder | - Translation of the UI/UX design wireframes to actual code that will  produce visual | All members | Offshore | 100% | 02/03/21 | 02/06/21 |

1. COMMUNICATION & REPORTING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of communication** | **Methods, tools** | **Frequency** | **Information** | **People** |
| ***Communication among in group*** | | | | |
| **Daily Meetings** | Messenger, Google Meet | Every day | Informed about what was done in the last 24 hours, working on plans for today, the difficulties encountered and the solutions required, just meeting 10-15 minutes. | Project team |
| **Task Planning Meeting** | Meet face to face, Google Meet, Messenger, Slack | Every 3  days | All members in team together to analyze the requirements, functions, working on the sprint going to do, planning and design for the sprint. | Project team |
| **Task Review Meeting** | Meet face to face, Google Meet, Messenger, Slack | 10-15 days | Complete documentation. For each stage, sharing materials, given the strengths and weaknesses for each. Period for each member and the solution calculated measurement project. | Project team |
| **Task Management** | Trello, Slack | Every day | A web-based task tracking system. To manage or divide task, report bugs/issues. | Project team. |

1. SECURITY ASPECTS

* Viruses
* Restrict clicking on unknown links as well as installing crack software
* Preventing viruses
* Using the antivirus software such as Bitdefender Antivirus Plus, McAfee Antivirus Plus.
* Turn on Window Defender and Firewall.
* Internet Security
* Using HTTPs instead of HTTP

1. REFERENCES
   * *Scrum Process*. Learn how to scrum with the best of ‘em from [https://ww](http://www.atlassian.com/agile/scrum)w.atlas[sian.com/agile/scrum](http://www.atlassian.com/agile/scrum)
   * *Security Aspects*. Computerscience.slc.gr. Retrieved 26 March 2021, from <http://computerscience.slc.gr/securityaspects.html>.
   * *System Testing - Tutorialspoint*. Tutorialspoint.com. Retrieved 26 March 2021, from <https://www.tutorialspoint.com/software_testing_dictionary/system_testing.htm>.
   * Choudary, A. *What is Integration Testing? | How to perform integration testing? | Edureka*. Edureka. Retrieved 26 March 2021, from [https://ww](http://www.edureka.co/blog/what-is-integration-testing-a-simple-guide-on-how-)w.edure[ka.co/blog/wha](http://www.edureka.co/blog/what-is-integration-testing-a-simple-guide-on-how-)t[-is-integration-testing-a-simple-guide-on-how-](http://www.edureka.co/blog/what-is-integration-testing-a-simple-guide-on-how-) to-perform-integration-testing/

### 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 |  |