# REPORT 2

# Software Project Management Plan

## Problem Definition

* 1. **Name of this Capstone Project**
* Offical name: Website Checker System
* Vietnamese name: Hệ thống kiểm tra website
* Abbreviation: WCS
  1. **Problem Abstract**
* This project is concern about the automation of a massive range of checks that would otherwise take an extraordinary amount of time and expertise to perform. We call it Website Checker System (WCS). WCS will provide a user-friendly interface to check and view result about enterprises' website.
* Businesses use the website as a main channel for providing information, products, and services to partners and potential customers. With the current checker system, enterprises only able to check performance on their website. Therefore, it’s hard to evaluate the true quality of their website in other facets.
* The Website Checker System will be a useful assistance to enterprise ensuring the quality of the information their website provides, how easily their users can navigate the site, and how functional their website is.
* Finally, WCS gives result and present as charts, helps the enterprise to review their website.
  1. **Project Overview**
     1. **Current Situation**

Below are the problems encountered in this project:

• Limit time and human resource: Team has only 4 member and time for all project is about 13 weeks for writing a document, implementing the products and testing.

• Schedule of team member: team members can have a conflict in meeting schedule because of sick, or class and work schedule, etc.

• Framework study: team members have a problem when applying the play framework into project. The team needs an amount of time to get familiar new techniques.

• New technique: Some team members are new to the techniques used in the project. The team needs an amount of time to get familiar with those techniques.

• Lack of UI (user interface), UX (user experience) design skill: Our team members all study IS major, and no one has studied UI, UX design. Therefore, that some UI may misunderstand or hard to use with normal users.

* + 1. **The Proposed System**

A web application for admin and website owner. Admin can manage user, manipulate on each repository. Website owner can manage their website, check quality, experience, technology and content on their website.

Task will be assigned vertically to team members, so that if one member quits, the team will not lack of resources.

* + 1. **Boundaries of the System**

The system can:

* Allow Admin to manage user accounts .
* Allow Admin to manage dictionaries.
* Allow Admin to manage warning words list.
* Allow Website Owner to manage websites.
* Allow Website Owner to choose the test.
* Allow Website Owner to export report.
* Allow Website Owner to view result test.
* Allow Website Owner to view the position of mistakes on the website .

The system cannot:

* Let Website owner self-manage their staff account.
  + 1. **Future Plans**
* Current system can only export individually for each single function. We design the system to provide options for users:
  + Export all test report: system will export all result test.
  + Implement visualization of result test by charts.
* Current systems can only perform work under the influence of the user. We design the system that allows scheduling for automatic testing:
  + Check by hour, day, month: The system will be scheduled to automatically test.
  + Customizable functions for automatic testing: The system will allow user select each functions with defined input parameters.
* Current system can only provide reports on the test results independently. We design the system to analysis and evaluation based on test results of related reports.
  + 1. **Development environment**
       1. **Hardware requirements**

**For server**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Minimum Requirement** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (8 Mbps) | Cable, Wi-Fi (50 Mbps or more) |
| **Operation System** | XP, Vista, 7, 10, Window Server 2008 | 10, Window server 2008 |
| **Computer Processor** | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz) |
| **Computer memory** | 4GB RAM | 32 GB RAM or more |
| **Storage space** | 1GB | 5GB or more |

Table 1 - Hardware Requirement for Server

**For PC**

|  |  |  |
| --- | --- | --- |
| **PC** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| **Operating System** | Window 7 | Window 7 or more. |
| **Computer Processor** | Intel® Core i3 1.4GHz | Intel® Core i5 2.50GHz |
| **Computer Memory** | 1GB RAM | 2GB RAM or more |
| **Web Browser** | Chromes (v42 or higher) | Chrome latest stable version |

Table 2 - Hardware requirement for PC

* + - 1. **Software requirement**

|  |  |  |
| --- | --- | --- |
| **Software** | **Name / Version** | **Description** |
| **Operation System** | Windows Server 2014 | Operating system and platform for development |
| **Environment** | Java EE 8.0, Node v10, npm v6 | Specification for developing web application |
| **Modeling tool** | StarUML | Used to design diagram |
| **IDE** | Intellij IDEA 2018.1.5, Visual Studio Code 1.27.2 , A5M2 2.12.1 (SQL) | Programming tools |
| **DBMS** | MYSQL 8.0, Elastic Search | Used to create & manage the database for system |
| **Source control** | Git on IDE (Git lab) | Used for source control |
| **Web browser** | Chrome 69 or above | Testing browser |

Table 3 - Software Requirement

1. **Project** **organization** 
   1. **Software Process Model**

This project is developed using Scrum model – part of an agile framework for Software development project. Our team choose Scrum model because of the following reasons:

* Our team only has 4 members, and tasks are assigned vertically, do all steps from design, coding, testing and implementation. Scrum is the most suitable model for small and medium project.
* After research about Trie tree, Radix tree data structure, Elastic search, we have defined problems. The risk of changing algorithm is high because proving accuracy of those algorithms is complicated. We need to use “try and test” method.
* The project contains complicated system and the concept of machine learning is very new for us, so we need to try many designs before the system run stability.
* In the project there are many new technologies that need to be learned. With the Scrum model, the team can learn and develop in parallel to meet deadline.
* There is no leader, no hierarchy in team, so team members work cheerfully, stimulating the initiative and creativity of each member.
* Product owner can change requirement or extend scope. The team will adapt to change better.



Figure 1 Scrum framework

Reference: <https://www.scrum.org/resources/what-is-scrum>

* 1. **Roles and responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Nguyễn Huy Hùng | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Nguyễn Trương Thúy Vi | Scrum Master | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing * Arrange Meeting * Risk Management |
| **3** | Lê Ngọc Trường | Scrum team member | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **4** | Nguyễn Đức Trịnh | Scrum team member | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **5** | Trần Phúc Anh | Scrum team member | * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |

Table 4 - Role and responsibilities

* 1. **Tools and Techniques**

|  |  |
| --- | --- |
| Tool/Technique | Name |
| Front-end | HTML, CSS, JavaScript, jQuery, React Semantic UI |
| Back-end | JavaEE, SpringBoot framework, JPA, Hibernate |
| IDE | NetBeans 8.2, IntelliJ IDEA 17.2 |
| DBMS | MySQL |
| Source Control | GitLab |
| Modelling tool | StarUML |

1. **Project Management Plan**
   1. **Product Backlog**

Product Backlog could be found [here](https://docs.google.com/spreadsheets/d/1l8cS3FOWHDAiH4pbiMn_CICsUAXiHBG2nwvYuGTO77g/edit?usp=sharing)

* 1. **Product Backlog**

Sprint Backlog can be found [here](https://docs.google.com/spreadsheets/d/1ULv0lvInrsGMDslUHAnGDAeZHYmFXplAJl1QXATGYdY/edit?usp=sharing)

* 1. **Deliverables**

|  |  |  |
| --- | --- | --- |
| **No** | **Deliverable** | **Note** |
| 1 | Introduction, Entity Relationship Diagram, Use Case Overview, Mock UI | Sprint 1 |
| 2 | Study Spring Boot Framework, React JS, Design User Interface for Web Application | Sprint 2 |
| … |  |  |
|  |  |  |

* 1. **All Meeting Minutes**

All sprint meeting minutes could be found here.

1. **Coding Convention**

**Summary**:

* **Naming Convention**:
  + Variable names should be short yet meaningful. The choice of a variable name should be designed to indicate to the casual observer the intent of its use.
  + Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.
* **Indentation**:
* One declaration per line is recommended since it encourages commenting.
* In absolutely no case should variables and functions be declared on the same line.
* Do not put different types on the same line.
* **Declarations Convention:**
  + One declaration per line is recommended since it encourages commenting.
  + Using Java Code Convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>