# Report 2 – Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* **Official name:** Just Walk Out Library
* **Vietnamese name:** Thư viện mượn sách tự động
* **Abbreviation:** JWL

### Problem Abstract

JWL system is provided in order to help librarians reduce work overload and give borrowers a much more comfortable experience in acquiring books. The role of this system is to automate the book-borrowing process with the help of RFID, NFC, QR Code, and iBeacon. However, those great technologies could come with some difficulties:

* Those are complicated and unfamiliar soft and hardware about which our team does not have enough knowledge. During the process of doing this research, this requires lots of time to study to understand clearly not only the operational principles but also detailed features and requirements. This is our main frustration.
* Not every borrower brings smart phone to the library. Thus, JWL needs to cooperate with the library’s traditional way in order to bring the most comfortable service to its user.

### Project Overview

### Current Situation

Below are the problems encountered in this project:

* **New technologies:** Our team have no experience in hardware. Therefore, we need time to do research about each technology, which devices to choose, and how to use them.
* **Traditional habit:** Borrowers have been using traditional way to obtain books for a long time, so libraries should provide proper introduction and assistance for their users to get use to JWL.
* **Devices problem:** RFID/NFC devices may be crashed or damaged after long time using, so backup solution should be prepared and periodically maintenance should be carried out.

### The Proposed System

* After doing many researches on technology for saving information, we choose RFID, NFC and QR code technologies as these technologies is very capable of resolve the current situations in borrowing book. The basic idea is to use a RFID tag that each tag contains a unique book ID.
* Our system includes two subsystems:
* Mobile Application:
  + - For borrower: to borrow book automatically when he/she steps out of the library.
    - For emulator: to read NFC/QR Code from the borrower’s mobile application.
  + Web Application:
    - For Admin: to manage user accounts.
    - For librarian: to manage borrowers.

#### Mobile Application

For borrower:

* Check in user through NFC (uses QR if NFC is not available)
* Manage user’s information.
* Search books.
* Add books to wish list.
* Track list of borrowed books.
* Notify user about:
  + Borrowed books.
  + Book return date.
  + Available books in user’s wish list.

For emulator:

* Scan NFC/QR Code for validation.

#### Web Application

For admins:

* + - Manage accounts (CRUD).

For librarians:

* + - Manage borrowers (CRUD).
    - Deactivate/activate borrowers.

### Boundaries of the system

* The system provides an outstanding solution to support the process of borrowing books in the library. This facilitates the automation of the borrowing process, as a result, implicit and explicit benefit would be brought to both librarians and borrowers:
  + Allow borrower to check in the library by scanning their phone to the RFID reader of the library.
  + Allow borrower to check out with their borrowed books by just walking out of the library.
  + Allow borrower to search for wanted book by title, author, category.
  + Allow borrower to add a book to a wish list.
  + Allow librarian to manage borrowers and their borrowed book list.
  + Allow admin to manage all accounts.
* Any libraries which deployed this system must set up their devices as required below:
  + One RFID tag on each book.
  + RFID reader, QR Code reader, and iBeacon at the entrance of the reading room.
  + Computer system with internet connection.
* Borrowers that use JWL have to make sure that:
  + They open the application when checking in and out of the library.
  + Their phones have internet and Bluetooth connection open.
* The completed product includes:
  + Android mobile application.
  + Website application.

### Future Plans

With further research and development, the system can apply the following features:

* Library can place RFID reader and iBeacon at the gate. Borrowers carrying books and mobile device just walk out. It is suitable for any borrowers who lack of knowledge in information technology.
* Library store book’s position and its status. System provide function allows borrows can search books, position and book status before going to library.
* Library can manage books on the website with RFID technology.

### Development Environment

#### Hardware requirements

* **For web application server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | Cable, Wi-Fi (4 Mbps) | Cable, Wi-Fi (8 Mbps) |
| Operating System | Window Server 2008 R2 | Window Server 2012 R2 |
| Computer Processor | Intel® Xeon ® 1.4GHz | Intel® Xeon ® Quad Core |
| Computer Memory | 2GB of RAM | 4GB of RAM or more |

*Table 1: Hardware Requirement for Server*

* **For Mobile (As Emulator)**

|  |  |  |
| --- | --- | --- |
| Android | Minimum Requirements | Recommended |
| Internet Connection | Wi-Fi or 3G (1 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.4.2 | Android 5.0.0 |
| Mobile Processor | Cortex-A7 Dual-Core 1.3GHz | Cortex-A7 Dual-Core 1.3GHz |
| Mobile Memory | 1GB of RAM | 2GB of RAM or more |
| Mobile Feature | NFC, Camera supporting, Google play service version more than 7.0 | NFC, Camera supporting, Google play service version more than 7.0 |

*Table 2: Hardware Requirement for Mobile as Emulator*

* **For Mobile (As User’s device)**

|  |  |  |
| --- | --- | --- |
| Android | Minimum Requirements | Recommended |
| Internet Connection | Wi-Fi or 3G (1 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.4.2 | Android 5.0.0 |
| Mobile Processor | Cortex-A7 Dual-Core 1.3GHz | Cortex-A7 Dual-Core 1.3GHz |
| Mobile Memory | 1GB of RAM | 2GB of RAM or more |
| Mobile Feature | - | NFC |

#### *Table 3: Hardware Requirement for Mobile as User’s device*

#### Software requirements

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Window 7 Professional SP1 | Operating system and platform for development |
| Environment | JDK version 1.8.0 update 101 | Specification for developing web application |
| IDE | IntelliJ IDE 2016.3.2, Android Studio v2.2.3, XCode | Use to implement website and Android Mobile App. |
| Design Model tool | StartUML v2.5.1 | Use to create modal and diagrams. |
| DBMS | PostgreSQL | Use to create & manage the database for system |
| Document storage | Github | Use to store document |
| Store and manage source code | Github & SourceTree | Use to store all source code |
| Web browser | Chrome 42 or above | Testing browser |

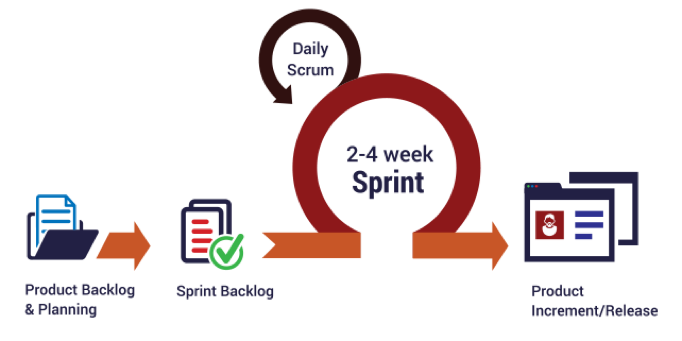
Table 4 : Software requirements

## Project organization

### Software Process Model

This project is developed using Scrum Model. This model is appropriate under consideration of 3 following reasons:

* Those technologies used in this project (RFID, NFC, iBeacon, and QR Code) require much attention in studying. This takes time to concentrate on technologies understanding, we plan to apply them gradually to the system.
* Prototypes are delivered frequently for evaluation.
* Team members can involve more in the development process.

  
Figure 1: Scrum model

Reference: <http://skytechnovation.com/scrum-development-model/>

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Kiều Trọng Khánh | Product owner, Project Manager | * Specify user requirements * Control the development process * Provide technical and business analysis support |
| 2 | Võ Hồng Hà | Scrum leader, B.A, Developer, Tester | * Manage process * Design database * Clarify requirements * Prepare and combine documents * GUI design * Create test plan * Code * Test |
| 3 | Nguyễn Tuấn Anh | Team member,  B.A, Developer,  Tester | * Design database * Clarify requirements * Prepare documents * GUI design * Create test plan * Code * Test |
| 4 | Đặng Nhật Thiên | Team member,  B.A, Developer,  Tester | * Design database * Clarify requirements * Prepare documents * GUI design * Create test plan * Code * Test |

*Table 5 : Roles and responsibilities*

### Tools and Techniques

|  |  |
| --- | --- |
| Tool | Name / Version |
| Web server | Tomcat |
| Development tool | IntelliJ IDEA 2016.3.2, Android Studio, Webstorm |
| DBMS | PG Admin 4 |
| Source control | Github & SourceTree |
| Modeling tool | StarUML v5.0.1 |
| Document tool | Microsoft Word 2010 |

Table 6: Tools List

|  |  |
| --- | --- |
| Technique | Name / version |
| Frontend | HTML5, CSS, ReactJS |
| Backend | Spring Boot |
| Mobile | Android, NFC, RFID, QR Code |

*Table 7: Techniques List*

## Project Management Plan

### Software development life cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Phase | Description | Deliverables | Resource needed | Dependencies and Constrains | Risks |
| Infrastructure | - Identify and clarify overall requirements.  - Determine the system architecture.  - Build infrastructure for the project. | - Database design.  - System main structure. | 20 man-days |  | - Unclear project scope.  - Lack of member share of understand. |
| System  &  Web admin | - Identify software and hardware requirements.  - Implements all web admin modules.  - Design the web UI  - Build the web admin | - Complete web admin for all roles of the system. | 60 man-days | - Depends on “Infrastructure” | - Unclear project scope.  - Lack of experience. |
| Web services | - Identify requirements for mobile app.  - Build required API for mobile app. | - API for mobile app. | 20 man-days | - Depends on “Web admin & System” | - Lack of experience. |
| Mobile apps | - Design the mobile UI  - Build mobile apps for end users and emulator. | - Complete Android Apps | 20 man-days | - Depends on “Web services” | - Lack of experience.  - Lack of RFID, NFC knowledge |

### Phase Detail

* + 1. ***Phase 1: Infrastructure***

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements.  - Create product backlog. | * HaVH * AnhNT * ThienDN |
| 2. Selection | - Determine system architecture: Spring Boot.  - Determine software design pattern: Repository & Service.  - Determine all core functions. | * HaVH * AnhNT * ThienDN |
| 3. Development | - Create the main structure of project. | * HaVH * AnhNT * ThienDN |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HaVH * AnhNT * ThienDN |

* + 1. ***Phase 2: System & Web admin***

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements for System and Web app.  - Update product backlog. | * HaVH * AnhNT * ThienDN |
| 2. Selection | - Determine all functions according to requirements of System and Web app. | * HaVH * AnhNT * ThienDN |
| 3. Development | - Design and build prototype for web UI  - Create conceptual diagram  - Design class diagram  - Design database  - Implement the entire web UI: layouts, detail pages, etc.  - Implement all the functions in controllers.  - Build needed utility classes | * HaVH * AnhNT * ThienDN |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HaVH * AnhNT * ThienDN |

* + 1. ***Phase 3: Web service***

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements for Web service.  - Update product backlog. | * HaVH * AnhNT * ThienDN |
| 2. Selection | - Determine all functions according to requirements of Web service. | * HaVH * AnhNT * ThienDN |
| 3. Development | - Create API for mobile app based on functions on the web app. | * HaVH * AnhNT * ThienDN |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HaVH * AnhNT * ThienDN |

* + 1. ***Phase 4: Mobile app***

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements for System and Mobile app.  - Update product backlog. | * HaVH * AnhNT * ThienDN |
| 2. Selection | - Determine all functions according to requirements of Mobile app. | * HaVH * AnhNT * ThienDN |
| 3. Development | - Implement all the functions based on the designed UI and the provided API. | * HaVH * AnhNT * ThienDN |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * HaVH * AnhNT * ThienDN |

### All Meeting Minutes

Meeting minutes are contained in folder “Meeting minutes” in the attached CD.

## Coding Convention

**Java:** Using to develop website and web service.

Summary:

* + Naming Convention:
* For variable’s name, use camel case. Eg: minValue, maxValue…
* For function name, class name, use Pascal case. Eg: AddIncome, AddExpense…
  + Layout Convention:
* Indent contrinuation one tab stop (four spaces).
* Add at least one blank line between method definitions and property definitions.
  + - Use parenteses to make clauses in an expression apparent.
* Using Java Code Convention from: <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

**Android:** Using to develop mobile application

Summary:

* Naming Convention:
  + For variable’s name, use camel case. Eg: minValue, maxValue…
  + For function name, class name, use Pascal case. Eg: AddIncome, AddExpense…
  + For resource file names are written in lowercase underscore. Eg: my\_name
  + Declarations Convention:
    - One declaration per line is recommended.
  + Using Android Code Convention from <https://source.android.com/source/code-style.html>