**Report 2 – Software Project Management Plan**

1. **Problem Definition**
   1. **Name of this Capstone Project**

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

To help librarians reduce of the amount of work and give borrowers a much more comfortable experience in acquiring books, we provide the JWL system. Our system will automate the book-borrowing process. However, there exists some problems that need consideration:

* JWL does not have face recognition. Therefore, the system may need the add from the librarian in checking in the exact user.
* 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.
  1. **Project Overview**
     1. **Current Situation**

Below are the problems that JWL may face:

* + **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.
    1. **The Proposed System**

JWL includes two parts:

* + Web Application:
    - Web Admin: for admin/librarian to manage user accounts and books.
    - Web User: for user to manage their own profile through browser, view borrowing books, search for wanted book, and track when a wanted book is available.
  + Mobile Application: for user to borrow book automatically when he/she steps out of the library. Besides, Mobile Application has functions of Web User.
    - 1. **Web Admin**

For admins:

* + - Manage accounts (CRUD).
      1. **Web User**

For librarians:

* + - Manage borrowers (CRUD).
    - Manage books (CRUD).
      1. **Mobile Application**

There will be 2 applications which will be used by borrowers and emulators. The mobile applications included functions as below:

* For borrowers:
  + Make NFC connection.
  + Show QR Code.
  + 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 emulators:
  + Read NFC, QR code.
  + Verify borrowers.
    1. **Boundaries of the System**
* The system is mostly built based on real processes of library management. Our main target is improving the current process and makes it more convenient and efficient.
* Any library which deployed this system must set up devices to operate, includes:
  + Emulator can read a mobile device with NFC supporting, with internet connection.
  + Emulator can read QR codes, with internet connection.
  + Emulator can read RFID, with internet connection.
* The completed product includes:
  + Website application
  + Android mobile application for borrowers and for emulators.
    1. **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.
  + 1. **Development Environment**
       1. **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 2 : Hardware Requirement for Server.

* **For Mobile (As Emulator)**

|  |  |  |
| --- | --- | --- |
| Android | Minimum | Recommended |
| Internet Connection | Wi-Fi or 3G (1 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.4.2 | Android 6.0.0 |
| 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 3 : Hardware Requirement for Mobile as Emulator.*

* **For Mobile (As User’s device)**

|  |  |  |
| --- | --- | --- |
| Android | Minimum | Recommended |
| Internet Connection | Wi-Fi or 3G (1 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.4.2 | Android 6.0.0 |
| 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. | NFC, camera supporting. |

*Table 4: Hardware Requirement for Mobile as User’s device.*

* + - 1. **Software requirements**

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Window 7 Professional SP1 | Operating system and platform for development |
| Environment | JDK version 1.8 | Specification for developing web application |
| IDE | IntelliJ IDE 2016.3.2, Android Studio v2.2.3, XCode | Used for implement website and Android Mobile App. |
| Design Model tool | StartUML v2.5.1 | Used for creating modal and diagrams. |
| DBMS | Postgre SQL | Used to create & manage the database for system |
| Document storage | Github | Used for storing document |
| Store and manage source code | Github & SourceTree | Used to store all source code |
| Web browser | Chrome 42 or above | Testing browser |

Table 5: Software requirements

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

The project is developed under waterfall model. We apply customized waterfall model to capable with current situation in our team. We choose this model because the following reasons:

* Based on researches and clarify library’s management process, the requirements of this project are stable, clear, fixed and well understood by all team members.
* This project use RFID, NFC, QR Code technology. They are must be understood clearly before implementing.
* This model is simple and easy to understand and use. It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process. In this model phases are processed and completed one at a time.



Figure 1 : Waterfall model.

*Reference:* *Page 30, chapter 2, Software process model, SOFTWARE ENGINEERING 9th Edition,*

*by Ian Sommerville.*

* 1. **Roles and responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Kiều Trọng Khánh | Supervisor, Project Manager | * Specify user requirements * Control the development process * Give out technique and business analysis support |
| 2 | Võ Hồng Hà | Team leader, B.A, Developer, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Nguyễn Tuấn Anh | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |
| 4 | Đặng Nhật Thiên | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Test |

* 1. **Tools and Techniques (WIP)**

|  |  |
| --- | --- |
| Tool | Name / version |
| Web server | Tomcat |
| Development tool | IntelliJ IDEA 2016.3.2, Android Studio, XCode. |
| DBMS | Postgre SQL |
| 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, JavaScript, jQuery |
| Backend | Spring MVC framework |
| Mobile | Android, iOS, NFC, RFID, QR Code. |

Table 7: Technique List

1. **Project Management Plan**
   1. **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 |  | - Lack of member share of understand.  - Unclear project scope. |
| System  &  Web app | - Identify software and hardware requirements.  - Implements all web app modules.  - Design the web UI  - Build the web app | - Complete web app for all roles of the system. | 60 man-days | - Depends on “Infrastructure” | - Lack of RFID knowledge.  - Lack of experience.  - Unclear project scope. |
| Web services | - Identify requirements for mobile app.  - Build required API for mobile app. | - API for mobile app. | 20 man-days | - Depends on “Web app & System” | - Lack of experience. |
| Mobile apps | - Design the mobile UI  - Build mobile apps for end users and emulator. | - Complete Android Apps  - Complete iOS Apps | 20 man-days | - Depends on “Web services” | - Lack of experience.  - Lack of NFC knowledge |

Table 8: Software Development Life Cycle Detail

* 1. **Phase Detail**
     1. **Phase 1: Infrastructure**

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements.  - Create product backlog. | * HaVH * AnhNT * ThienDN |
| 2. Selection | - Determine system architecture: Java Spring MVC.  - 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 |

Table 9: Phase 1: Infrastructure

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

|  |  |  |
| --- | --- | --- |
| 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 |

Table 11: Phase 3: Web service

* + 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 |

Table 12: Phase 4: Mobile app

* 1. **All Meeting Minutes**

**(WIP)**

1. **Coding Convention**

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

**Android:** Using to develop mobile application