**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 for librarians and give borrowers a much more comfortable experience in acquiring books, we provide the JWL system to make the book-borrowing process automatic. However, there exists some problems that need consideration:

* JWL uses NFC and RFID technology, which is quite unfamiliar for software engineering students. Thus, we will need to invest more time and effort in researching and applying those devices.
* Not every user has smart phone, and not every smart phone supports NFC. Therefore, we may need to develop some alternative solutions, like notice users about JWL and its benefit, or use QR code in parallel with NFC.
  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**
      2. **Web User**
      3. **Mobile Application**
    1. **Boundaries of the System**
    2. **Future Plans**
    3. **Development Environment**
       1. **Hardware requirements**
       2. **Software requirements**

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

The project is developed under scrum model. Scrum model is capable with current situation in our team. We choose this model because the following reasons:

* ??
* This project use RFID, NFC, QR Code technology, which is a new technology that may need many times to research and implement.
* Scrum adopts an empirical approach, accepting that the problem is not fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly, to respond to emerging requirements and to adapt to evolving technologies and changes in market conditions.



Figure 1 : Scrum model

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

* 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 * Create test plan * Coding * Test |

* 1. **Tools and Techniques**

|  |  |
| --- | --- |
| Tool | Name / version |
| Web server | Tomcat (?!) |
| Development tool | IntelliJ IDEA 2016.3.2, Android Studio (iOS development tool ?) |
| DBMS | ? |
| 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 | SpringMVC framework, Jersey RESTful Web Services framework, Android, 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 |  | - Unclear project scope.  - Lack of member share of understand. |
| 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” | - Unclear project scope.  - Lack of RFID knowledge.  - 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 app & 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 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: ASP .NET MVC.  - Determine software design pattern: Repository & Service.  - Determind 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 | - Determind 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 | - Determind 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**

1. **Coding Convention**