

Capstone Project Final Report

**FPT STUDENT INTERNSHIP MANAGEMENT SYSTEM**

|  |  |
| --- | --- |
| **FIMS Team** | |
| **Group Members** | Bùi Công Tùng \_ SE04450  Bùi Trung Kiên \_ SE05182  Đỗ Đình Đức \_ SE04936  Trần Quang Thọ \_ SE04596  Nguyễn Hữu Hóa \_ SE05118 |
| **Supervisor** | Lecturer: Nguyễn Tất Trung |
| **Capstone Project code** | FIMS |

**Hanoi, October 17, 2019**

# **Acknowledgements**

*We wish to express our deepest gratitude to our supervisor Mr. Nguyen Tat Trung for his continuously sharing and motivating throughout the project. Under the instructions of Mr. Trung, the project team has always felt comfortable and confident. Apart from all the technologies and methodologies, Mr. Trung shared with us the philosophy of keeping the right attitude towards problems. We believe that was the most important factor that led to the success of this project.*

*We would also like to thank the instructors at FPT University for all the classes. We hope you will find this project as a reflection of the knowledge and experiences you have given us during this period of four years.*

*Finally, we truly appreciate the random fellows on the internet who has discussed with us. Without them, it would have likely been a real struggle to solve every tiny problem.*

# **DEFINITIONS AND ACRONYMS**

**Terminologies**

|  |  |  |
| --- | --- | --- |
| **No** | **Definition** | **Explanation** |
| 1 | **FSIMS** | Website FPT Student Internship Management System. |
| 2 | **FSIMS** | A stand-alone application that manages students system in the 5th semester of the school. |
| 3 | **FSIMS Store** | Website FPT Student Internship Management System Store |
| 4 | **FSIMS Console** | Website FPT Student Internship Management System Console |
| 5 | **FSIMS Developer** | People who develops fpt student internship management system for web applications. |
| 6 | **FSIMS User** | People who need web fpt student internship management system for their website. |
| 7 | **User** | The term includes FSIMS Developer and FSIMS User |
| 8 | **Guest** | People who visit the FSIMS Store and haven’t had the account yet. |

Acronyms

|  |  |  |
| --- | --- | --- |
| **No** | **Definition** | **Explanation** |
| 1 | **API** | Application Programming Interface. |
| 2 | **DOC** | Document Object Model. |
| 3 | **GUI** | Graphical User Interface. |
| 4 | **HTTP** | Hypertext Transfer Protocol |
| 5 | **HTTPS** | Hypertext Transfer Protocol Secure |
| 6 | **UML** | Unified Modeling Language. |
| 7 | **N/A** | Not Available. |
| 8 | **OS** | Operating System. |
|  |  |  |
| 9 | **PM** | Project manager |
| 10 | **SRS** | Soft Ware Requirement Specification |  |  |  |
| 11 | **UC** | Use Case |  |  |  |
| 12 | **UI** | Use Interface |  |  |  |
| 13 | **UX** | Use Experience |  |  |  |
| 14 | **FB** | Facebook |  |  |  |
| 15 | **FU** | FPT University |  |  |  |
| 16 | **FSIMS** | FPT Student Internship Management System |  |  |  |
| 17 | **DEV** | Developer. |  |  |  |

**Chapter 1: Introduction**

## **Propose**

This chapter provides an overview of the capstone project including background information, a literature review of the existing system and raising a proposal for ideas of improvement.

## **Project Information**

* Project name: FPT Student Internship Management System.
* Project code: FSIMS.
* Project group name: FSIMS Team
* Project type: Web Application
* Timeline: 10/09/2019 - 26/12/2019

## **The people**

Supervisor:

|  |  |  |  |
| --- | --- | --- | --- |
| **Full name** | **Phone** | **E-Mail** | **Title** |
| Nguyễn Tất Trung | 0904399139 | TrungNT@fpt.edu.vn | Supervisor |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Full name** | **Code** | **Phone** | **E-mail** | **Role** |
| 1 | Bùi Trung Kiên | SE05182 | 0983268048 | Kienbtse05182@fpt.edu.vn | Leader |
| 2 | Bùi Công Tùng | SE04450 | 0356551659 | Tungbcse04450@fpt.edu.vn | Member |
| 3 | Đỗ Đình Đức | SE04936 | 0968480997 | Ducddse04936@fpt.edu.vn | Member |
| 4 | Nguyễn Hữu Hóa | SE05118 | 0326041206 | Hoahhse05118@fpt.edu.vn | Member |
| 5 | Trần Quang Thọ | SE04596 | 0985138384 | Thotqse04596@fpt.edu.vn | Member |

Members:

## **Background**

1. In recent years, the school has become more and more crowded. Therefore, during each student internship, the management of the school will be very difficult and time-consuming. Usually, Recruiter will connect with students via email or create group chat in Facebook. This job is very manual.

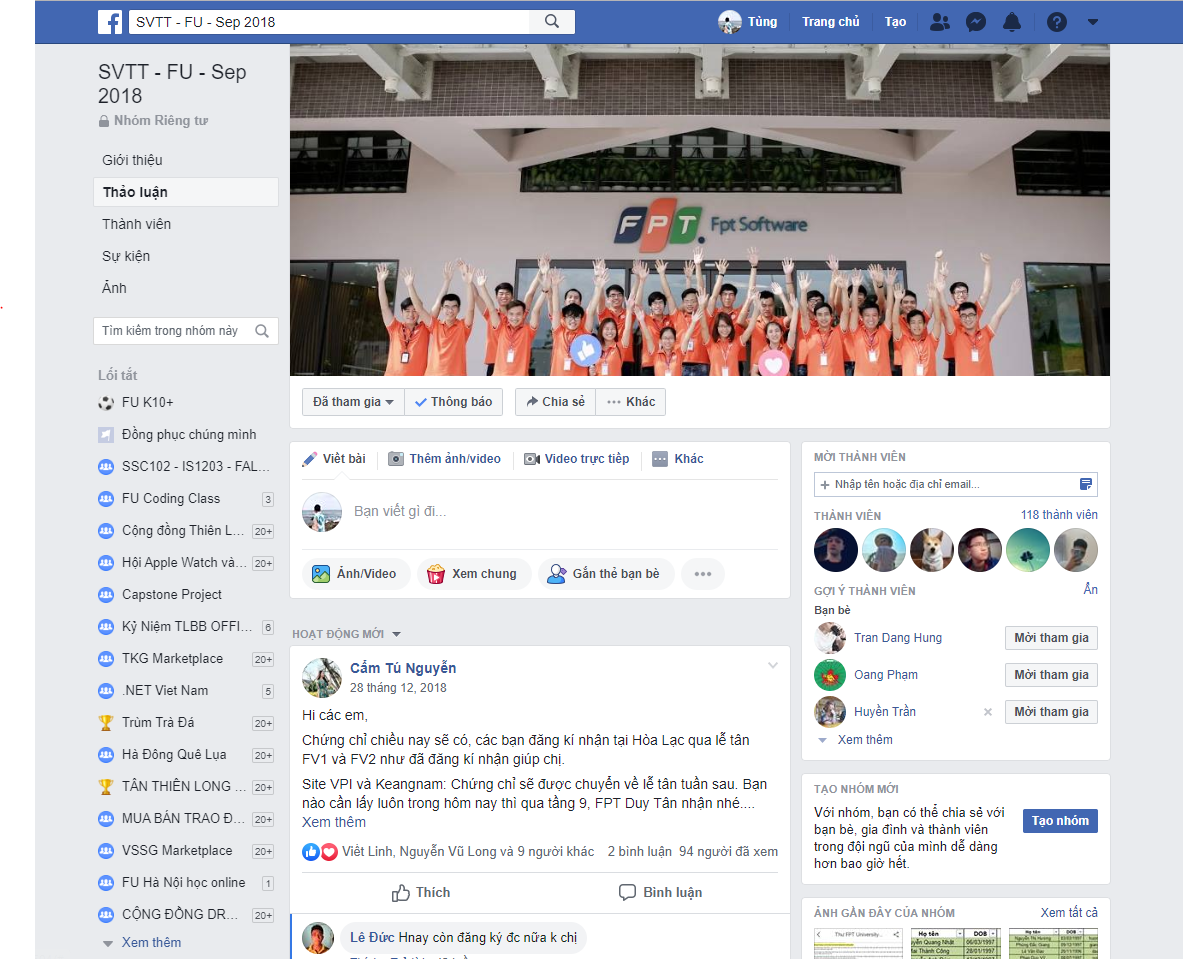
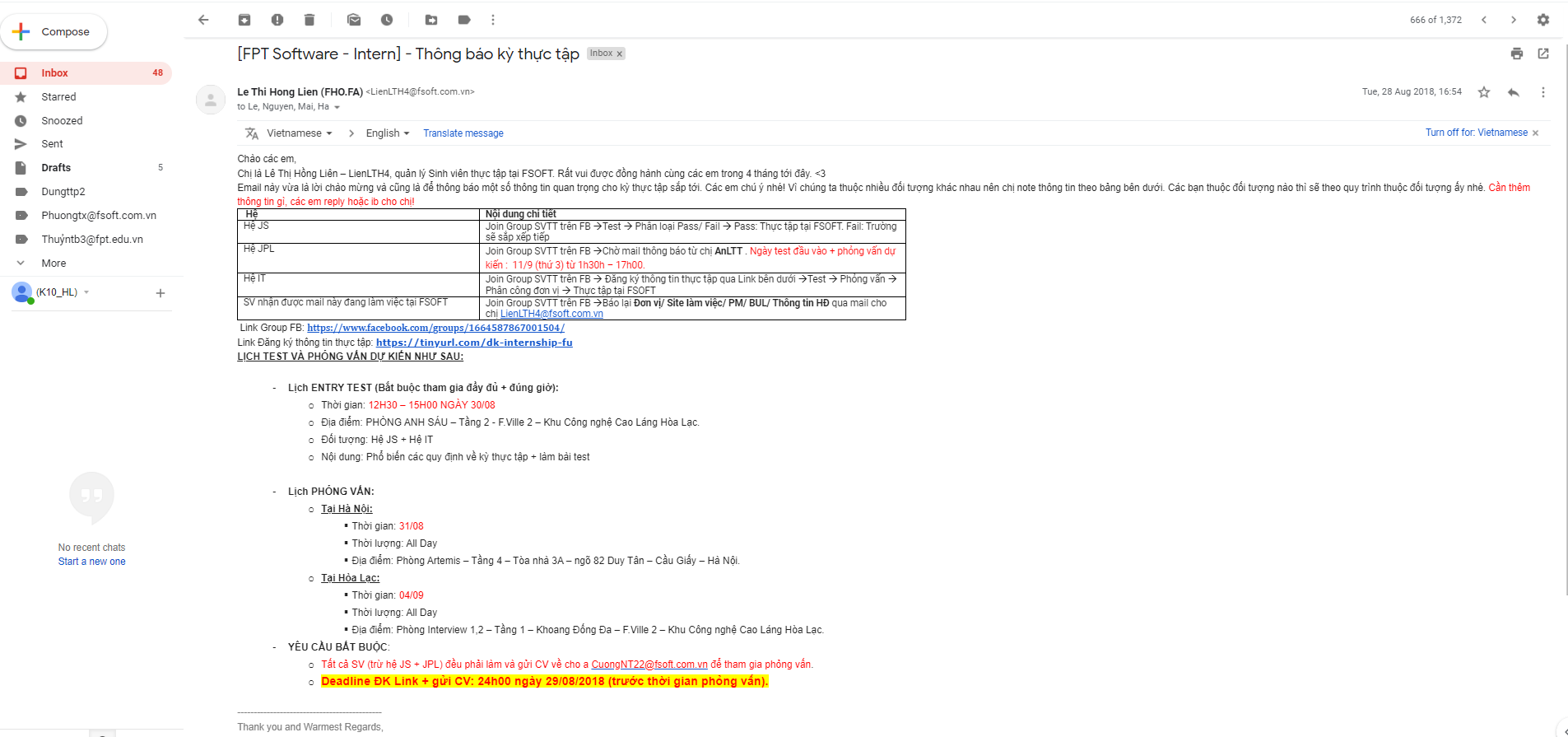


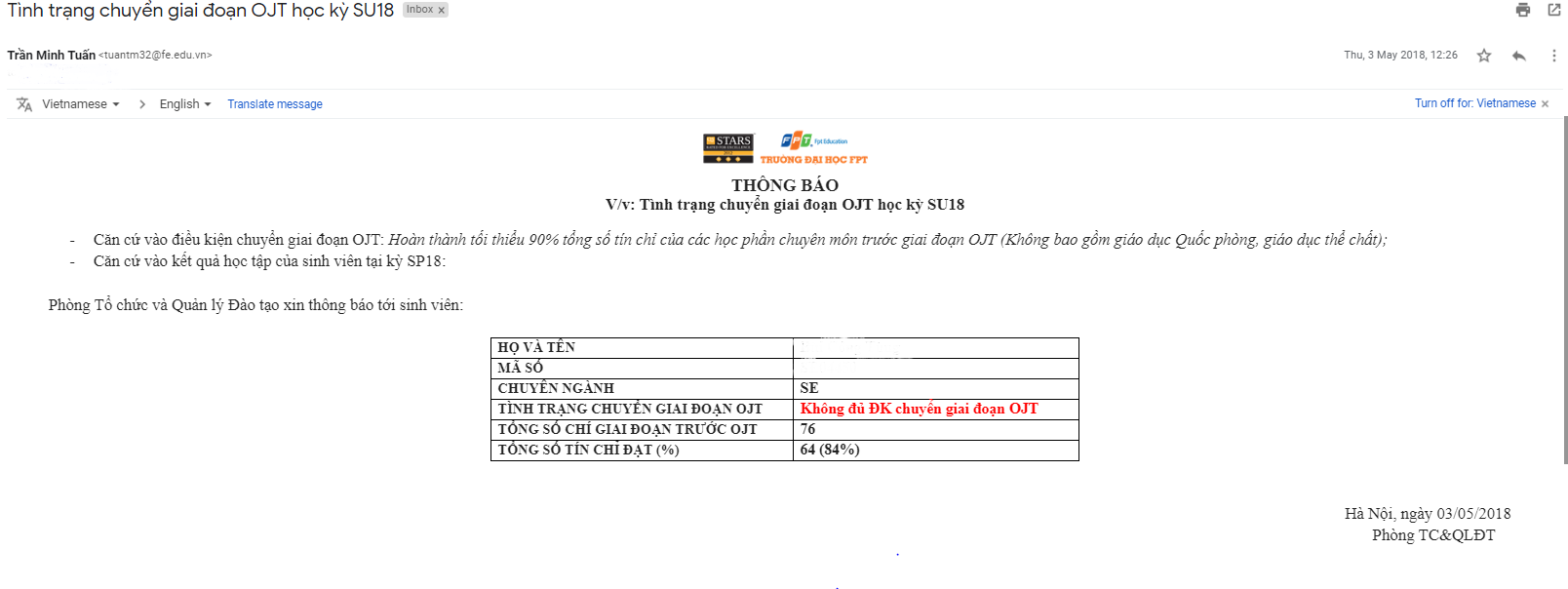
Figure 1-1: FU Facebook group

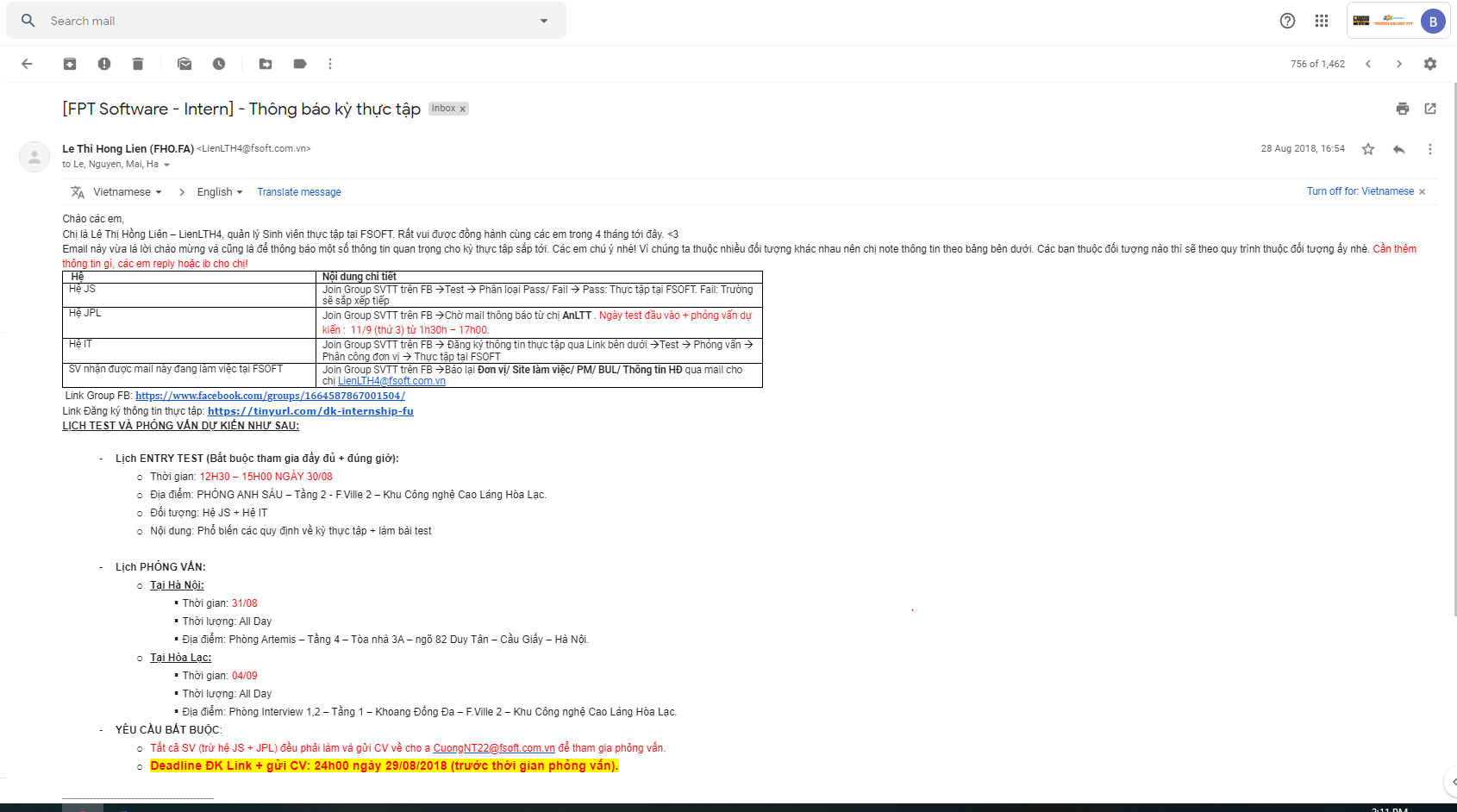
  
Figure 1-2: FU gmail

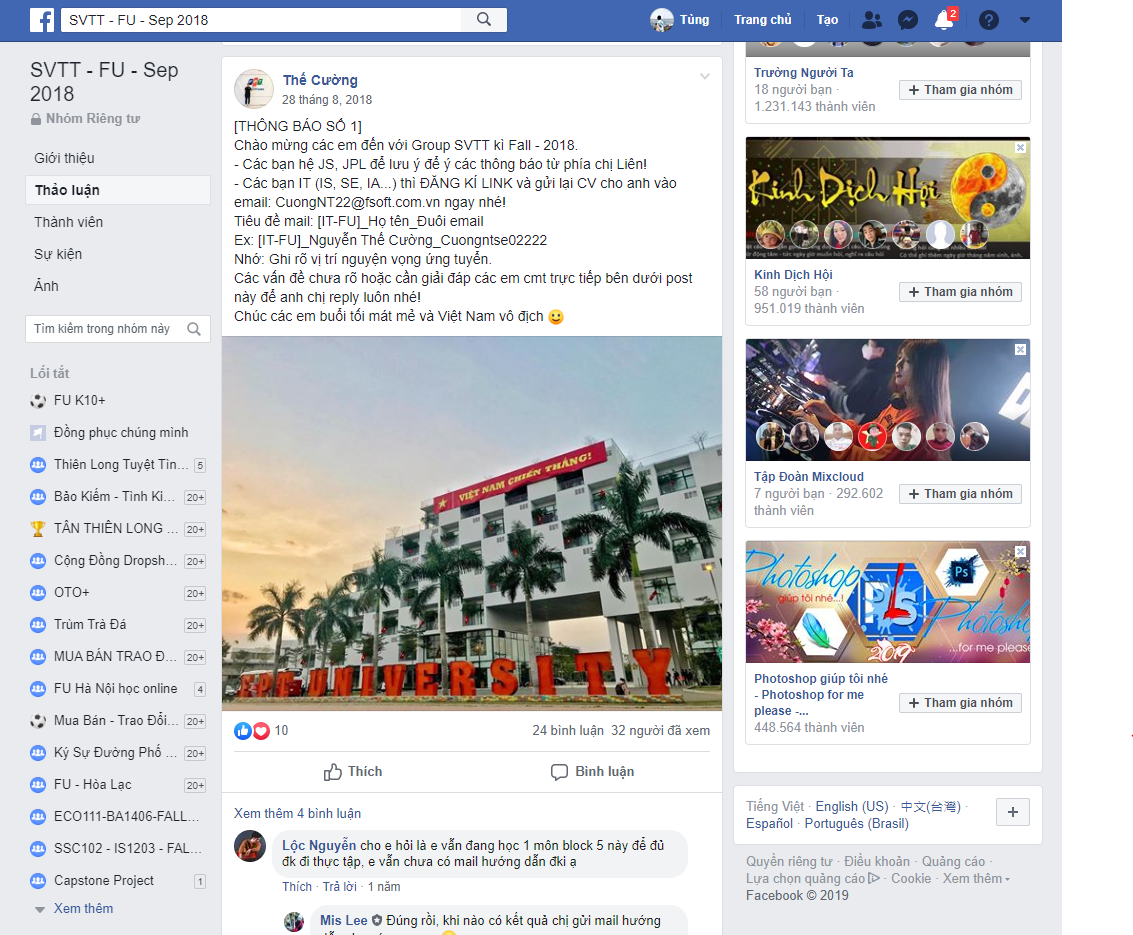
The communication between students and employers and the school is very manual. The connection and management will take a lot of time and cost. It will be an annoying problem for the school and students during the internship period. In order to alleviate this burden, we have developed a web application project. The main objective of this web site is to create the convenient management of student information, to develop a live chat box between students and employers about popular questions as well.

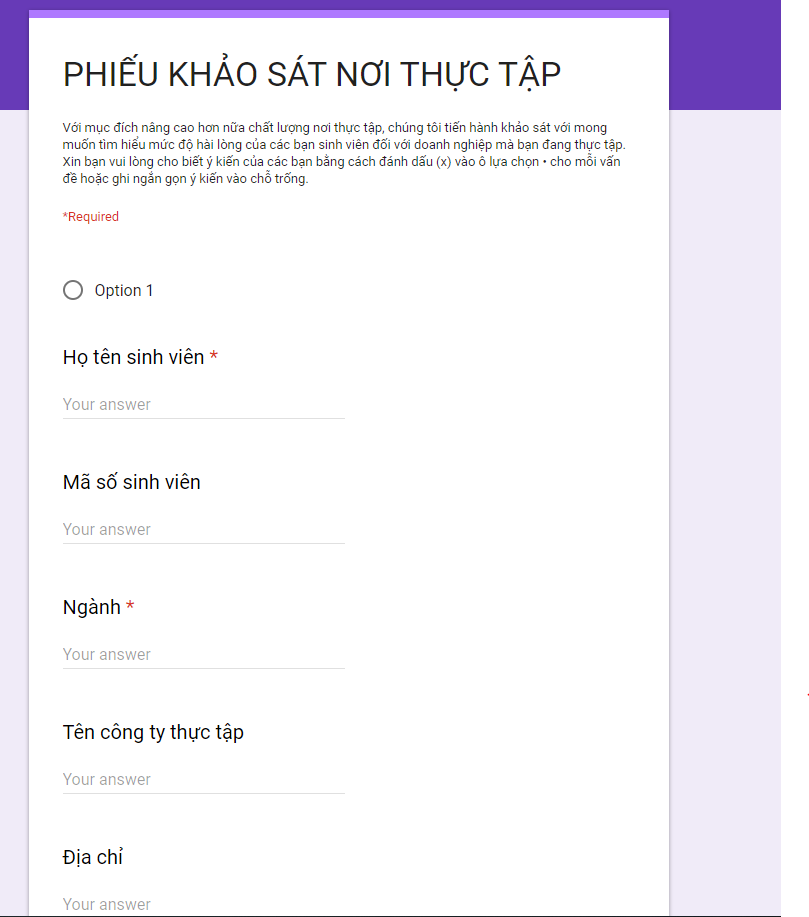
## **The Existing System**

As far as we know, major universities in Vietnam still manage their internships manually. For example, University of Technology manages internships with excel, and FPT University does the same. Currently, FPT University's students have increased year by year, the management of interns every semester is increasingly complicated. Moreover, by the internship period, whether being eligible to intern is also a big problem for students. The school can be considered as a bridge but has encountered difficulties in making notices. Email notification is also easy to miss. At that time, students will be worried about whether their internship is eligible or not. The following example is about a student who is not eligible to intern in the next semester:

  
Another issue that we want to talk about is that after students are eligible for internships, there will be some announcements about the time and place that the school was affiliated with the company. For example, the school has provided qualified student’s information to employers and then informed students of internship time via Facebook or e-mail:



  
In this case, it will be very detrimental for students who do not often use social networks. They will lose contact information with employers. Or in case of email notification, it is possible that the message is put in the spam folder because of being considered as strange notifications. Those are shortcomings in the current internship for both the school and the students. Because of the large number of students, the school's staff will not be able to handle all cases of notification errors. In addition, the school also sent the questionnaire to the students to evaluate, report back some information about where they have interned so that the students of the following course will have a realistic view.

In fact, these reviews are not realistic, so we have also improved this part of our project. Our project is based on a practical basis as well as the criteria that school personnel provide for project members.

## **The proposal of system**

This section covers the system in which the project team will develop to overcome the problems mentioned above

### System introduction

FSIMS (FPT Student Internship Management) is a bridge that connects the events in the next internship of FPT University students. FSIMS provides all necessary information as well as internship schedule for FPT students. At FSIMS, students can fill their aspirations about their own development direction and learn about a specific skill during an internship at the company, such as Java, C #, Python, AI, or learn about the BA. Moreover, through this program, students can talk directly with recruiter. With FSIMS, students have the opportunity to make comments, feedback on recruiter, giving students in the following course with a more general view of company .For recruiter, they can review and find suitable candidates for their companies with a large number of profiles, or post job listings. From the recruitment newsletters, students can see the internship opportunities that they can choose to best suit themselves.

FSIMS will provide multi-dimensional view with great value to students.

### System Function

There are 5 types of user in FIMS system:

|  |  |
| --- | --- |
| **Type of User** | **Description** |
| Guest | People who are visitors to FPT Student Internship Management System. |
| Recruiter | Recruiter come from company or subsidiaries of incorporation in the fields of technology, business, multimedia, accounting or other language industries. |
| Admin | People who have a role in managing the accounts. |
| Staff | They are staff at university who are responsible for maintaining the system, handling feedback reports of students and recruiter, monitoring the students of the university according to the weekly and monthly reports during their internship at the company. |
| Student | People who are studying in FPT University at Hoa Lac Campus. |

FSIMS system will provide the following functions to the above users :

* Allow all of recruiter check profile students in FPT University.
* Allow all of recruiter chat box with students.
* Allow all of recruiter schedule an interview send notifications for students.
* Allow all of recruiter evaluate the students final result after internship.
* Search by criteria of students, recruiter.
* Allow all of recruitment post job information.
* Allow all of students post your profile, filter in your aspirations.
* View information about your internship provided by the recruiter.
* View schedule interview, feedback and interview schedule changes.
* Filtering the aspirations of the employer send the profile directly to the recruiter.
* Evaluation, feedback, comments criteria of students to recruiter.
* Dashboard information all of recruiter.
* Detail information about one recruiter.
* Table statistics students who are wishing to practice under specific classification.
* Admin See the employers' live evaluation by week and student feedback.
* Evaluate the results of students after the internship, export excel data.
* Push notifications on the guest system.
* Accept recruiters into the system.
* Table of quality statistics of students of 1 course internship, employer quality.
* Manage current student status (add, edit, delete students).
* Manage current recruiter status (add, edit, delete students).

### 1.6.3 Out of functions

Because we were both studying and running the project, limited of time. We only do internship management for software engineering students of FPT University version 1.0. But, in the future of version 2.0, we will try to complete with the remaining major of FPT University. In addition, will try to develop the application on:

- Support for multi campuses.

- Mobile phone responsive layout.

- Dashboard page for administrators to view system statistics.

# **CHAPTER 2: SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)**

## **2.1 Purpose**

Project Management Plan chapter helps readers to understand how project organized and to view closer the project plan. All members must use this document to clarify each own responsibility. It also provides principles, which team members need to abide by during the whole time of developing project.

### **2.1.1 Software process model** D:\Downloads\1200px-Iterative_development_model.svg.png

*Figure 2-1: Iterative development Model*

#### 2.1.1.2 About the iterative model

The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. Each cycle (iteration) ends with a usable system.

* **Planning & Requirements:** the first step is go through an initial planning stage to map out the specification documents.
* **Analysis & Design:** the second step is performed to nail down the appropriate business logic, database models. The design stage also occurs here, establishing any technical requirements to meet the needs of the analysis and design stage.
* **Implementation:** the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.
* **Verification and testing:** the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up.
* **Evaluation:** the last step is evaluating the development up to this stage. This allows the entire team to examine where the project is at, where it needs to be, what should change.

#### 2.1.1.2 Advantages and disadvantages considering

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Innovative technology is involved-developer team can learn as they work on the project. | Heavily document changes when there are more and more iteration. |
| The major requirements are defined, while some functionalities enhancements evolve with the process of the development process. | Issues about system architecture may turn out to be a big problem when lacking of a full requirements specification. |
| Functional prototypes are developed early and delivery for each iteration. | Highly skilled resources are required for analysis skill. |

### **2.1.2 Roles and Responsibilities**

#### 2.1.2.1 Organization Structure

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Project Manager | Planning, developing schedules, coordinating communication, generally responsible for keeping the team’s focus on the main goal. |
| Technical Leader | Responsible for choosing and deciding what technologies should be used, as well as for overseeing the work being done by other developers. |
| Quality Assurance Manager | Ensuring the product meets the certain standards of quality from requirements. |
| Test Leader | Responsible for test execution, including test set-up and test run, evaluation of test run and error recovery, defect logging and test results recording. |
| Developer | Involve to code the product and review code of other developers. |
| Designer | Involve to design product’s user interface. |
| Tester | Involve to test the product. |
| Business Analyst | Analyzes an organization or business domain and documents its business or processes or systems. |

*Table 2-1 : Project Structure*

#### 2.1.2.2 Project Team Member

|  |  |
| --- | --- |
| **Team Member** | **Role** |
| KienBT | Project Manager, Developer, Business Analyst, Designer. |
| TungBC | Developer, Quality Assurance Manager, Business Analyst, Test Lead. |
| DucDD | Technical Leader, Developer, Tester. |
| ThoTQ | Developer, Tester. |
| HoaNH | Developer, Tester. |

*Table 2-4: Project Team Member*

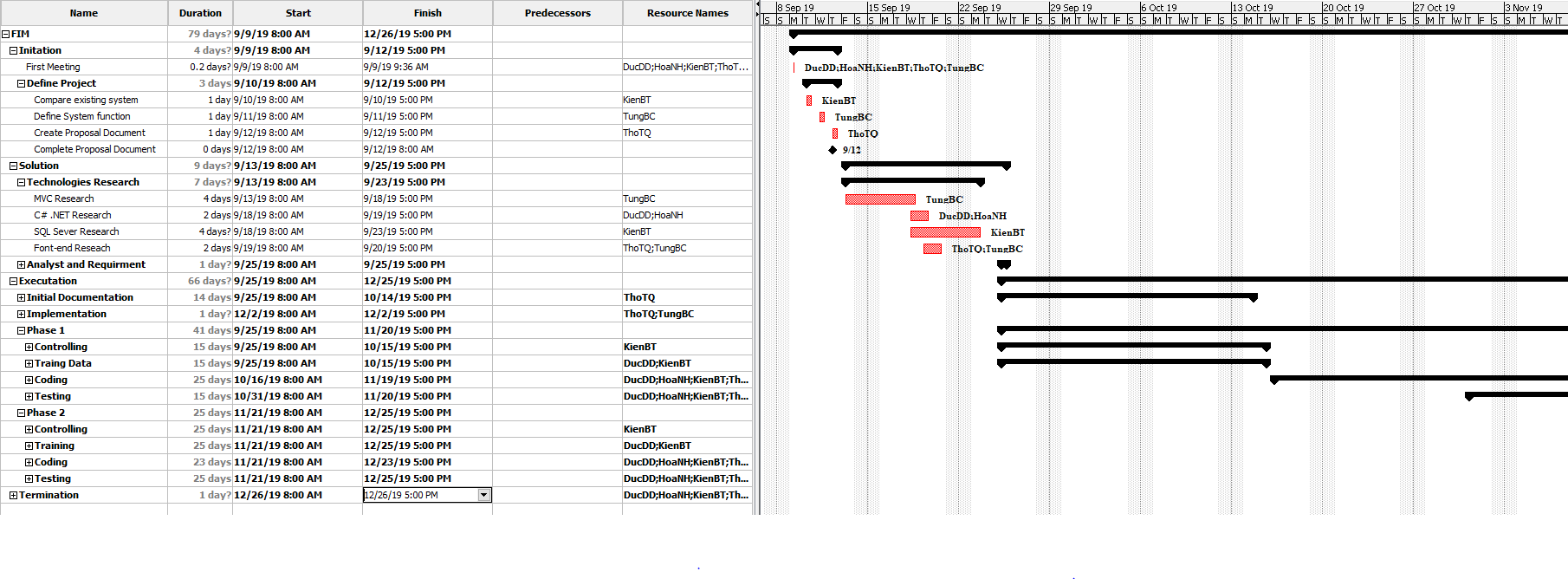
#### 2.1.2.3 Tool and Techniques

|  |  |  |
| --- | --- | --- |
| **Software Tool** | **Version** | **Description** |
| Microsoft project | 2013 | Management tools |
| Project | Online | Management tools |
| Gitlab | Online | Source code control |
| Source tree | 2.6.10 | Source code control |
| Sublime Text 3 | 3.2.2 | IDE for coding |
| Visual Studio Code 2017 | 15.9.16 | IDE for coding |
| Microsoft Words | 2016 | Document tools |
| Microsoft Excel | 2016 | Document tools |
| Microsoft SQL Server 2017 | 8.0.11 | Sql server |
| Draw.io | Online | Diagram, UML tool, chart |
| Facebook | Online | Communication tool |
| Process Model |  | Iterative Model |
| Trello | Online | Management tools |
| Skype |  | Communication tool |
|  |  |  |
|  |  |  |

*Table 2- : Tool and Techniques*

## **2.2 Project Management Plan**

### 2.2.1 Tasks



### 2.2.2 Meeting Minutes

All meeting minutes will be written following this template:

|  |  |  |  |
| --- | --- | --- | --- |
| *Meeting/ Project Name:* | FSIMS | | |
| *Date Of Meeting :* | 09/09/2018 | Time : (Type) | 1 hours (Face-to-face) |
| *Meeting Call by :* | KienBT | Location : | FPT University – P212 |
| *Note Take :* | TungBC | Time Keeper: | KienBT |
|  |  |  |  |
| 1. Meeting Objective | | | |
| 1. Assign new task for member 2. 2. Do report No.2 | | | |
| 1. Attendance | | | |
| *Name* | *Roles* | *E-mail* | *Phone* |
| Bui Trung Kien | Project Manager | *Kienbtse05182@fpt.edu.vn* | 0983268048 |
| Bui Cong Tung | Developer, Tester | [*Tungbcse04450@fpt.edu.vn*](mailto:Tungbcse04450@fpt.edu.vn) | 0356551659 |
| Do Dinh Duc | Developer, Tester | Ducddse04936@fpt.edu.vn | 0968480997 |
| Nguyen Huu Hoa | Developer, Tester | Hoahhse05118@fpt.edu.vn | 0326041206 |
| Tran Quang Tho | Developer, Tester | Thotqse04596@fpt.edu.vn | 0985138384 |
| 1. Done Task | | | |
| 1. Build project structure 2. Configure git repository with security 3. Do report No.1 | | | |
| 1. New task | | | |
| 1. Back-end a. Training .Net Core MVC b. Setup Environment  c. Code Modules Integration  2. Front-end  a. Setup environment  b. Code Dashboard Homepage  c. Code Login page  3. Do report No.2 | | | |
| 1. Risk & Difficulty | | | |
| N/A. | | | |

*Table 2-4: Meeting Minutes Template*

### 2.2.3 Coding Conventions

**Purposes:**

* Easy maintenance, fix bug.
* Let others understand your source code.
* Uniform code between team members.

**Follow Java Coding Convention at:**

<https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

### 2.2.4 Risk Management Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Description | Avoidance plan | Contingency plan | Status |
| R1 | Data loss | - Use GitLab for version control. - Teach members how to use Git and resolve conflicts. - Always have important backups | - Restore backed up data from GitLab. | Closed |
| R2 | Illness or absence of team members | - Provide schedules in advance. - For long periods of absence, members should notify the group in advance | - Assign the tasks of absent member to other members.  - Work overtime if necessary. | Closed |
| R3 | Misunderstanding of requirements | - Discuss requirements carefully with the customer. - Always ask for clarification if requirement is unclear. - Comment need to meet reality and possibility. | - Make sure idea’s business logic is carefully analyzed. | Closed |
| R4 | Requirement changed | Every new update of requirement has to be reviewed by all team members and supervisor. - Team member has to analyze requirement carefully before raise up to team. | - If requirement has new update, all members have to join the meeting to aware and make decision. | Closed |
| R5 | Conflict between team members | Everything must be documented. - Every team member has to express clearly and carefully. | - Make sure any miscommunication has to be resolved | Closed |
| R6 | Failure to meet deadline | - Plan and develop schedule carefully - Assign tasks carefully - Define punishment for team members who neglect work | Find the root cause of the problem - Reassign tasks - Focus on important functions first | Closed |

*Table 2-5: Risk Management*

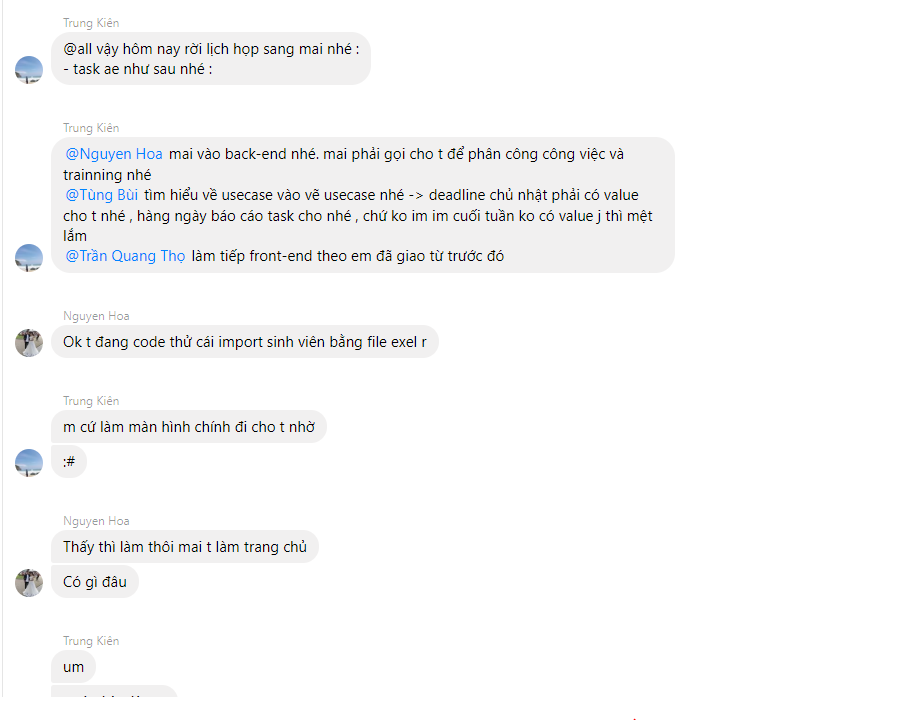
### 2.2.5 Communication Plan

*Weekly meeting schedule*: We use Iterative and Incremental Process Model. Each task is logged to Excel then estimated depending on difficulty and the amount of work by the whole team, after that the task will be assigned to team members by the Team Leader and depending on difficulty the Technical Leader will assign deadlines for each task. We will have a meeting every Sunday to inform to all team about what each member finished last week, the status (fast, on time or slow), the issues met and how to solve them. If any member raises any issue, the whole team will help to find out a solution together. After that, the team will define detailed stories for next week tasks.

*Daily meeting schedule*: Each sub-system has one development team with different schedule. When starting work-day, each team will have a stand-up meeting to inform to others: “What did I do yesterday?”, “What will I do today?” and “Is there any difficulty?”. By focusing on what each person accomplished yesterday and will accomplish today, the team gains an excellent understanding of what has been done and what remains.

*Unscheduled meeting*: If someone has an important problem that he wants to solve immediately, we will have a meeting for discussion, usually via some online channel: Skype, Zalo, Facebook or Phone.

*Communication channel*: Our main communication channel is Skype. On the other hand, we use face-to-face meeting, Facebook group and comment on Excel issues. However, we sometimes make a phone call or instant message if someone has a problem.



*Figure 2-3: Communication via Facebook*

# **CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**

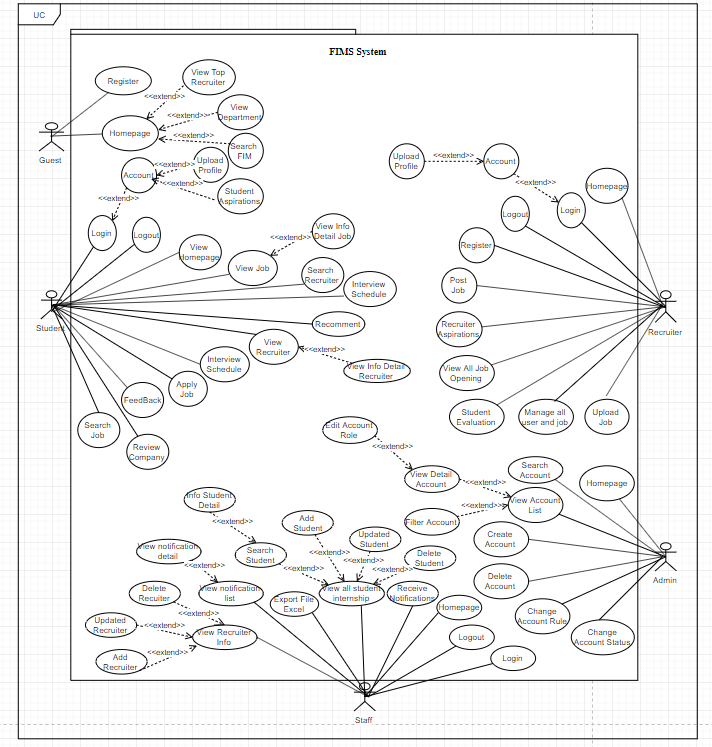
## **3.1 Purpose**

Software Requirements Specifications chapter provides general, as well as specific information of the system. It also shows the describing of functional requirements and non-functional requirements. Following these requirements, FPT student internship management system performs its intended functionality used in designing developing and testing. All members work based on the information that provided in this document.

## **3.2 Functional Requirements**

### 3.2.1 Use Case Diagram

The figure below is display our use case that be had in this project, and it is display on use case diagram.

  
*Figure 3-1: Use Case overview system.*

### 3.2.2 Business Rules

|  |  |
| --- | --- |
| No | Description |
| B1 | Username must not be empty. |
| B2 | Username must be unique. |
| B3 | User’s full name must not be empty. |
| B4 | User’s address must not be empty. |
| B5 | The email must not be empty. |
| B6 | Email address must be valid. |
| B7 | Each email address may be used for only one account. |
| B8 | Re-enter password must match with password. |
| B9 | Password must not be empty. |
| B10 | Password reset token expires after 24 hours. |
| B11 | The password length must not be less than 6. |
| B12 | Phone number must be valid. |
| B13 | The administrator is able to: active/ deactivate account, change account role, change request state. |
| B14 | Every user accessing dashboard has a role: staff or admin or recruiter. |
| B15 | Phone number must be numeric string only. |
| B16 | The administrator is able to: active/ deactivate account, change account role, change request state of recruiter. |

*Table 3-1: Business rules*

### 3.2.3 Use Cases

|  |  |
| --- | --- |
| Actor | Description |
| Guest | Everyone who does not log into the FSIMS system. |
| Student | Student who are studying in FPT University can quickly capture information about internship period happening in the school by viewing details of the event: time, location, internship content, special notes internship. they can also register to participate in addition in the list of personal aspirations. After the internship, students can respond and evaluate the quality of employers and company they had OJT. |
| Recruiter | The recruiter views student’s details, finds the candidate he/she wants. Direct contact with students to schedule interviews, introduce the company. Student achievement reports are sent to the school weekly and monthly and the end of the internship period. |
| Staff | The school reserves the right to view and evaluate students during the internship period and to end the internship period. The school has an overview of the employer and company of the company in which students participate. Comments and reviews help students make good choices and appropriate to their competence. |
| Admin | Administrators have a role in managing all accounts. Admin can view all but cannot modify any information. |

*Table 3-2: Actor description*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Actor** | **Name** | **Description** |
| **UC-001** | Student, Staff, Recruiter, Admin | Login | Login to dashboard in FIMS account. |
| **UC-002** | Student, Staff, Recruiter, Admin | Logout | Logout to dashboard in FIMS account |
| **UC-003** | Student, Staff, Recruiter, Admin | Reset Password | Reset a forgotten password using email. |
| **UC-004** | Student, Admin, Recruiter, Staff | Change password | Change password of account. |
| **UC-005** | Student, Admin, Recruiter, Staff | View Homepage | User can view homepage. |
| **UC-006** | Student, Admin, Recruiter, Staff | Review Company | All User Review Company |
| **UC-007** | Student, Admin, Recruiter, Staff | View Job | Allows users to view all current jobs |
| **UC-008** | Student, Admin, Recruiter, Staff | View Info Detail Job | Allows users to view the entire detailed description of the job |
| **UC-009** | Student, Admin, Recruiter, Staff | View Recruiter | Allows users to view all recruiter |
| **UC-010** | Student, Admin, Recruiter, Staff | View Info Detail Recruiter | Allows users to view all information of recruiter |
| **UC-011** | Admin | Search Account | Search for account by username or full name. |
| **UC-012** | Admin | Filter Account | Allow Staff to filter accounts by role. |
| **UC-013** | Admin | View Account List | View list of accounts in the system. |
| **UC-014** | Admin | Create Account | Create a new staff FIMS account. |
| **UC-015** | Admin | Change Account Status | Change Account Status for Staff |
| **UC-016** | Admin | Change Account Rule | Change Account Rule for Staff |
| **UC-017** | Admin | Delete Account | Delete Account for user FIMS |
| **UC-018** | Admin | View Detail Account | View Detail Account for user FIMS |
| **UC-019** | Admin | Edit Account Role | Edit Account Role for user FIMS. |
| **UC-020** | Guest | View Top Recruiter | View Top Recruiter for Student |
| **UC-021** | Guest | View Department | View Department for user. |
| **UC-022** | Guest | Search FIMS | Find all information on the FIMS website |
| **UC-023** | Student | Upload Profile | Upload Profile for Student. |
| **UC-024** | Student | Student Aspirations | Aspirations for Student |
| **UC-025** | Student | Apply Job | Apply Job for Student |
| **UC-026** | Student | Search Job | Search Job for Student |
| **UC-027** | Student | Search Recruiter | Search Recruiter for Student |
| **UC-028** | Student, Staff | Feed Back | User can view detail feedback for recruiter |
| **UC-029** | Student, Recruiter | Interview Schedule | Recruiter can book schedule interview with student. |
| **UC-030** | Student, Recruiter | Recomment | Show the aspirations of students and recruiter alike |
| **UC-031** | Staff | View Info Recruiter | Staff can view info Recruiter |
| **UC-032** | Staff | Add Recruiter | Staff can add recruiter |
| **UC-033** | Staff | Updated Recruiter | Staff can updated recruiter |
| **UC-034** | Staff | Delete Recruiter | Staff can delete recruiter |
| **UC-035** | Staff | View notification list | Staff can view notification list |
| **UC-036** | Staff | View notification detail | Staff can view notification detail |
| **UC-037** | Staff | Export File Excel | Staff can be exported into excel files transcript |
| **UC-038** | Staff | View all student internship semester | Staff can view all student internship according to semester. |
| **UC-039** | Staff | Search Student | Staff can search all student in semester |
| **UC-040** | Staff | Receive Notifications | Show notifications for work management staff |
| **UC-041** | Staff | Delete Student | To allow staff delete student violating regulations |
| **UC-042** | Staff | Updated Student | Allow staff to update information of students |
| **UC-043** | Staff | Add Student | Allow staff to add students at the beginning of each semester |
| **UC-044** | Staff | Info Student Detail | Show student details |
| **UC-045** | Recruiter | Register | Recruiter can register with admin provide account. |
| **UC-046** | Recruiter | Upload Profile | Upload Profile for Recruiter |
| **UC-047** | Recruiter | Upload Job | To upload jobs provided by different companies |
| **UC-048** | Recruiter | Manage all user and job | Manage all students who are interns at the company and the jobs they are recruiting |
| **UC-049** | Recruiter | Student Evaluation | Allows Recruiter to assess students who have practiced internships at their companies |
| **UC-050** | Recruiter | View all job opening | Allows employers to view all job openings on the website |
| **UC-051** | Recruiter | Recruiter Aspirations | Information about the aspirations of the Recruiter |
| **UC-052** | Recruiter | Post Job | Allows Recruiter to post jobs that the company needs |
| **UC-053** | Recruiter, Student | Chat Box | Interactive student with recruiter |

*Table 3-3: Use Case List*

### 3.2.4 Use Cases Detail

**2.4.1.39. Update Student *UC-001. Login***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **UC- Log in** | | |
| Create by: | TungBC | Date Created: | 10/12/2018 |
| Primary Actor: |  | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow admin, staff, student and recruiter who has FIMS account to login. | | |
| Preconditions: | Staff, admin, student, recruiter, account has been set up and username and password has been generated. | | |
| Post conditions: | 1. The staff student and recruiter is logged into FIMS successfully. 2. System displays Dashboard Home Page. | | |
| Normal Flow: | **Login:** 1. Staff visits the website.  2. Staff fills valid username and password then clicks “Login”.  3. System displays Dashboard Home page | | |
| Alternative Flows: | Staff violates some business rule in step 2  1. System displays help message popup.  2. Staff enters username and password complying the business rules then clicks “Login”.  3. Go to step 3 in normal flow. | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: |  | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-002. Logout***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **UC- Logout** | | |
| Create by: | TungBC | Date Created: | 10/12/2018 |
| Primary Actor: | Student, Admin, Recruiter, staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | To prevent other users from stealing information | | |
| Preconditions: | User has logged into FIM system | | |
| Post conditions: | 1. The staff student and recruiter is logged into FIM successfully. 2. System displays Dashboard Home Page. | | |
| Normal Flow: | **Logout:** - User is logged out of the system  - Browser displays Login screen  - Access token will be cleared on the browser | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIM displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: |  | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-003. Reset Password***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **UC- Reset Password** | | |
| Create by: | TungBC | Date Created: | 10/12/2018 |
| Primary Actor: | Student, Admin, Recruiter, Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Staff, Student, Admin, Register to reset a forgotten password. | | |
| Preconditions: | Student, Staff, Admin, Register account has been set up and username and password has been generated. | | |
| Post conditions: | New password is reset and sent to staff via his/her email. | | |
| Normal Flow: | **Reset password:** 1. Student, Admin, Register, Staff visits the dashboard page.  2. System displays Login page. 3. Staff clicks ‘Forgot Password’. 4. System displays Forgot Password page. 5. Staff enters his/her email, then clicks “Please provide password”.  6. System displays a message that asks the staff to check email and follow instruction.  7. Staff clicks on the link in email.  8. System displays Reset Password page.  9. Staff is required to input password twice, then clicks “Confirm”.  10. System displays message “Password change successfully”.  11. System displays Login page | | |
| Alternative Flows: | User violates some business rule in step 5  1. System displays help message popup.  2. Staff enters email address complying the business rules then clicks “Please provide password”.  3. Go to step 6 in normal flow.  4.0 User violates some business rule in step 9  1. System displays help message popup.  2. Staff enters password complying the business rules then clicks “Confirm”.  3. Go to step 10 in normal flow. | | |
| Exceptions: | Cannot communicate with server 1. FIM displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | Low | | |
| Business Rules: |  | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-004. Change Password***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Change password** | | |
| Create by: | TungBC | Date Created: | 10/12/2018 |
| Primary Actor: | Student, Admin,  Recruiter, Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Staff, Student, Recruiter to change his/her account password. | | |
| Preconditions: | All Staff, Student, Recruiter to change his/her account password. | | |
| Post conditions: | 1. New password is saved into database. 2. Staff Student, Recruiter logins successfully with new password. | | |
| Normal Flow: | Change Password  1. From the website header, staff clicks on avatar icon in the top right corner.  2. Staff clicks “Change the password”  3. System displays the Change Password page.  4. Staff enters current password, new password and re-enter password in the corresponding text boxes.  5. Staff clicks “Change password”.  6. System saves new data into database. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIM displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | Low | | |
| Business Rules: |  | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-005. View Homepage***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | View Homepage | | |
| Create by: | ThoTQ | Date Created: | 10/12/2019 |
| Primary Actor: | Student, Admin, Recruiter, Guest, Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow activities related to the student internship | | |
| Preconditions: | User has logged into FIMS website. | | |
| Post conditions: | “Homepage” screen will be shown. | | |
| Normal Flow: | **Main flow 1:** Role Student   1. Students login in FIMS system successfully. 2. System display “Student Homepage” screen with options below:   **Main flow 2:** Role Admin  **Main flow 3**: Role Recruiter  **Main flow 4**: Role Guest | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-006. Review Company***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Review Company | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Admin, Recruiter, Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | All User Review Company | | |
| Preconditions: | 1. User can access the system.  2. User logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | N/A. | | |
| Normal Flow: | Review Company  1. User clicks on “Feedback” on the left side of the screen.  2. System displays table of all Recruit.  3. In the “Đánh Giá” text box, student enters text.  4. In the box “Vote” student choose the number of stars User want to vote for this company.  5. User click “Save” button to upload Feedback of company | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIM displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-007. View Job***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View Job** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Admin, Recruiter, Staff | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Allows users to view all of the job application | | |
| Preconditions: | 1. User login successfully 2. User click to view all job from Home page. | | |
| Post conditions: | N/A | | |
| Normal Flow: | 1. The user clicks on any job on the Job Page. 2. System display all information of job 3. User can view and comment about job at the bottom of the page | | |
| Alternative Flows: | Alternative flow 1:  1. The user click on any job display on the Homepage  2. System displays all information of job. | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-008. View Info Detail Job***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View Info Detail Job** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Admin, Recruiter, Staff | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Allows users to view detailed information of the job application | | |
| Preconditions: | 1. User login successfully 2. User click to view all job from Home page. | | |
| Post conditions: | N/A | | |
| Normal Flow: | 1. The user clicks on any job on the Job Page. 2. System display all information of job 3. User can view and comment about job at the bottom of the page | | |
| Alternative Flows: | Alternative flow 1:  1. The user click on any job display on the Homepage  2. System displays all information of job. | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-009. View Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **UC- View Recruit** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Admin, Recruiter, Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allows users to view all recruiter | | |
| Preconditions: | No Preconditions | | |
| Post conditions: | The user accesses in to the page FIM | | |
| Normal Flow: | 1. User visits the website.  2. User click to button “See more” to view more Recruit  3. System displays Dashboard Recruit page | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIM displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: |  | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-010. View Info Detail Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **UC- View info Detail Recruiter** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Admin, Recruiter, Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Display Recruiter Information | | |
| Preconditions: | No Preconditions | | |
| Post conditions: | Display all information of recruiter | | |
| Normal Flow: | 1. Student, Staff, Guest visits the dashboard page.  2. System display all information of Recruiter  3. Guest scroll down to view more job information of Recruiter. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | Low | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-01 . Search Account***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Search Account | | |
| Create by: | ThoTQ | Date Created: | 10/12/2019 |
| Primary Actor: | Admin | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | All Admin to search account(s) belong to username. | | |
| Preconditions: | 1. Staff can access the system.  2. Admin logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | N/A. | | |
| Normal Flow: | Search Account by Username  1. Admin clicks on “All account” on the left side of the screen.  2. System displays table of all accounts.  3. In the “Account” text box, staff enters text.  4. System displays account(s) that related to search keyword. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-012. Filter Account***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Filter Account | | |
| Create by: | ThoTQ | Date Created: | 10/12/2019 |
| Primary Actor: | Admin | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Admin to filter accounts by role. | | |
| Preconditions: | 1. Admin can access the system.  2. Admin logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | N/A. | | |
| Normal Flow: | Filter Account by Role  1. Admin clicks role dropdown button  2. System displays role list in dropdown list.  3. Admin selects a specific role.  4. System hided role list and displays selected role and accounts list related to selected role. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-013. View Account List***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View account list** | | |
| Create by: | TungBC | Date Created: | 10/12/2019 |
| Primary Actor: | Admin | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Staff to view the list of accounts in the system | | |
| Preconditions: | 1. Staff can access the system.  2. Staff logins successfully.  3. Account data has been added to the system by the admin. | | |
| Post conditions: | System displays the account list screen with necessary elements. | | |
| Normal Flow: | View Account List  1. Staff clicks on “View all account” on menu in the left side bar of the screen.  2. System displays list of all accounts | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-014. Create Account***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Create Account** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Create a new staff FIM account. | | |
| Preconditions: | Admin has logged into the FIM system. | | |
| Post conditions: | Admin can create new account of website | | |
| Normal Flow: | 1. Admin clicks “Create Account” button at menu home. 2. System shows “Create Account” screen. 3. Admin fills in Create account form and clicks “Submit” button. 4. System saves the information of the new account and redirects admin to the new list of account. | | |
| Alternative Flows: | Extension flow 1:  1. At the step 3, instead of clicking “Submit” button, admin clicks “Back to list” button.  2. System redirects admin to the “View all categories” screen without changing anything at database.  Extention flow 2:  1. Admin inputs data in step 3 violates one or some of business rules.  2. System displays error message(s) on the create form. | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-015. Change Account Status***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Change Account Status** | | |
| Create by: | ThoTQ | Date Created: | 10/12/2018 |
| Primary Actor: | Admin | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Change Account Status for Staff | | |
| Preconditions: | 1. Admin logins successfully. | | |
| Post conditions: | N/A. | | |
| Normal Flow: | Verify Recruit Account :  1. Admin clicks Status dropdown button  2. System displays role list in dropdown list.  3. Admin selects a button “Staff Status”.  4. Admin view all information of Recruit and click “Active” or “Block” or “Processing” button | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-016. Change Account Rule***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Change Account Rule** | | |
| Create by: | ThoTQ | Date Created: | 10/12/2018 |
| Primary Actor: | Admin | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Change Account Rule for Staff | | |
| Preconditions: | 1. Admin logins successfully. | | |
| Post conditions: | N/A. | | |
| Normal Flow: | Verify Recruit Account :  1. Admin clicks Role dropdown button  2. System displays role list in dropdown list.  3. Admin selects a button “Verify Recruit”.  4. Admin view all information of Recruit and click “Accept” or “Decline” button | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-017. Delete Account***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Delete Account** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | To allow admin delete user violating regulations | | |
| Preconditions: | Admin going to login to their account and delete a user violent | | |
| Post conditions: | The Admin deletes the users successfully. | | |
| Normal Flow: | 1. The Admin login to their account . 2. The system checks the account validation and returns with the result of login. 3. The Admin go to Delete User tab and choose the user that they want to delete 4. The system sent a request to delete the user to the database and displays a message. 5. The “User deleted successfully”. | | |
| Alternative Flows: | Alternative flow 1:  1. User inputs data in step 1 violates one or some of business rules.  2. System displays error message(s) on the feedback form. | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-018. View Detail Account***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | View Detail Account | | |
| Create by: | ThoTQ | Date Created: | 10/12/2019 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Admin and see own profile to update the information includes avatar,  name, major,… These information could be used for student review recruiter. | | |
| Preconditions: | Student has logged into FIMS website. | | |
| Post conditions: | *“Account”* screen will be shown. | | |
| Normal Flow: | System shows “View Profile” screen which includes: Avatar of student, email, name, address, CV file. After that, students will revise and update the CV. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server. At step 2, system displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-019. Edit Account Role***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Edit Account Role | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Edit Account Role for user FIMS. | | |
| Preconditions: | Admin logins successfully. | | |
| Post conditions: | Account must be active by user. | | |
| Normal Flow: | 1. Admin clicks on “Manage Account” icon on the top side of the screen bar.  2. System displays all type of account button.  3. Admin click on any type of user to manage accounts.  4. Admin click to button role dropdown of each account to change role.  5. Admin click to “Save” button to save all role of user after changed. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |
|  |  | | |

***UC-020. View Top Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | View Top Recruiter | | |
| Create by: | ThoTQ | Date Created: | 10/12/2019 |
| Primary Actor: | Guest | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow visitor users to see the top recruiter list | | |
| Preconditions: | User has logged into FIMS website. | | |
| Post conditions: | “Homepage” screen will be shown. | | |
| Normal Flow: | **Main flow :**   1. Guest user access in FIMS website. 2. System display “Top Recruiter” screen in section 2. 3. Guest user click to “See More” button to view more recruiter | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-021. View Department***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View Department** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Guest | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | View all deparment in university | | |
| Preconditions: | Guest user click to “View Department” in bottom of home page | | |
| Post conditions: | N/A | | |
| Normal Flow: | 1. Guest user click to “View Department” in bottom of home page 2. System show all department of school. 3. Guest click on any one deparment to show information. | | |
| Alternative Flows: | 1. Guest click to another department after that click to view all department  2. System displays error message(s) on the Pop-up | | |
| Exceptions: | N/A | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-022. Search FIMS***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Search FIMS** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Guest | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Allow guest user find all information on the FIMS website | | |
| Preconditions: | Guest user click to “Search Box” in top left of home page | | |
| Post conditions: | N/A | | |
| Normal Flow: | 1. Guest user click to “Search Box” in top left of home page 2. System show all job, department , recruiter have keyword searched. 3. Guest click on any one to show information. | | |
| Alternative Flows: | 1. Guest click to another department after that click to view all department  2. System displays error message(s) on the Pop-up | | |
| Exceptions: | N/A | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-023. Upload Profile***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Upload Profile** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | All Student to upload information of CV. | | |
| Preconditions: | 1. Student can access the system. 2. Student logins successfully. 3. Student click to “Tài khoản” dropdown menu in the top right to click “Upload CV” menu | | |
| Post conditions: | System displays a function to up load file pdf | | |
| Normal Flow: | 1. Student clicks on “Upload CV” on the mid side of the screen.  2. System display CV of student uploaded  3. Student fill in the desired job content in “Công việc mong muốn” textbox  4. Student clicks Save CV to post CV to website FIM | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIM displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-024. Student Aspirations***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Student Aspirations | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Admin | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Aspirations for Student | | |
| Preconditions: | Student logins successfully. | | |
| Post conditions: | Account must be active by user. | | |
| Normal Flow: | 1. Student clicks on “Manage Account” icon on the top side of the screen bar.  2. System displays all type of account button.  3. Student click on any type of user to manage accounts.  4. Student click to Aspirations fill and fill all information.  5. Student click to “Save” button to save all role of user after changed. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-025. Apply Job***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Apply Jobs** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | To accept student to apply for a particular job. | | |
| Preconditions: | A student going to login to account search for a job and fill the application form and submit the button | | |
| Post conditions: | The user submit the form and receive a message “Application submitted successfully”. | | |
| Normal Flow: | 1. The user login to his account. 2. The system checks the account validation and return with the result of login 3. The user then clicks the search job option and search for particular job to apply for. 4. The system tells and evaluate search criteria enter by user and return the result with appropriate search “if found”. 5. The system displays the search result. 6. The user then go to application form tab and fill out details and click “submit” button. 7. The system saves the record and displays the message “Application submitted successfully”. 8. Then user waits for company to response for his/her application. | | |
| Alternative Flows: | Alternative flow 1:  1. User inputs data in step 1 violates one or some of business rules.  2. System displays error message(s) on the feedback form. | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-026. Search Job***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **UC- Search Jobs** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Student search for job information | | |
| Preconditions: | No Preconditions | | |
| Post conditions: | 1. The user click to search button in Homepage 2. Search all Jobs page open with new tab. | | |
| Normal Flow: | **Search Jobs:** - Guest user click to address fill to search Jobs by address. - Guest user click to “Phân loại” dropdown to search follow Old and new  - In the “Category” checkbox, user check the boxes that users want to categorize  - In the “More fillter” checkbox, user check the boxes that users want to categorize by work time.  - Guest user see all job by filter and click to “apply for job” then user login display. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-027. Search Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **UC- Search Recruiter** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Search Recruiter for Student | | |
| Preconditions: | No Preconditions | | |
| Post conditions: | 1. The user click to search button in Recruit Page 2. Search all recruiter page open with new tab. | | |
| Normal Flow: | **Search Recruiter:** - Student click to address fill to search Recruiter by name. - Student user click to “Phân loại” dropdown to search follow Old and new  - In the “Category” checkbox, user check the boxes that users want to categorize  - In the “More fillter” checkbox, user check the boxes that users want to categorize by work time. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-028. Feedback***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Feedback | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Student can be choose and write feedback with Recruit | | |
| Preconditions: | 1. Studnet can access the system.  2. Student logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | N/A. | | |
| Normal Flow: | Feedback  1. Student clicks on “Feedback” on the left side of the screen.  2. System displays table of all Recruit.  3. In the “Đánh Giá” text box, student enters text.  4. In the box “Vote” student choose the number of stars Student want to vote for this company.  5. Student click “Save” button to upload Feedback of company | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-029 . Interview Schedule***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Interview Schedule** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Recruiter | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Student to view and book the list schedule interviews in the system | | |
| Preconditions: | 1. Student can access the system.  2. Staff logins successfully.  3. Account data has been added to the system by the admin. | | |
| Post conditions: | System displays the schedule screen with necessary elements. | | |
| Normal Flow: | Online scheduling for interviews  1. Student clicks on “Book lịch PV” on menu in the left side bar of the screen.  2. System displays list of information of schedule  3. Student clicks on Calendar frame to book date and time interviews  4. Recruit clicks on “Xem lịch phỏng vấn” to see all schedule interviews. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: |  | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-030. Recomment***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Recomment** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student,Recruiter | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Show the aspirations of students and employers alike | | |
| Preconditions: | N/A | | |
| Post conditions: | Postconditions will be determined during the next iteration. | | |
| Normal Flow: | 1. Student & Recruiter click to “View Recomment” in bottom of home page 2. System show all aspirations 3. Student & Recruiter click on any one to show more information of aspirations | | |
| Alternative Flows: | 1. Student & Recruiter to another year after that click to view all Job opening  2. System displays error message(s) on the Pop-up | | |
| Exceptions: | N/A | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-031. View Info Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View Info Recruiter** | | |
| Create by: | TungBC | Date Created: | 10/12/2018 |
| Primary Actor: | Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | All Staff to view detail information of an account. | | |
| Preconditions: | 1. Staff can access the system.   2. Staff logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | System displays a model with chosen account information. | | |
| Normal Flow: | View Account Information  1. Staff clicks on “View all account” on the left side of the screen.  2. System displays list of all accounts.  3. Staff clicks “Show more” button on each row in table.  4. System displays account information. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Low | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-032. Add Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Add Recruiter** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: | Admin |
| Trigger: | N/A | | |
| Description: | Allow staff to add recruiter at the beginning of each semester | | |
| Preconditions: | 1. Staff can access the system.  2. Staff logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | Postconditions will be determined during the next iteration. | | |
| Normal Flow: | Basic Flow – Add Recruiter:  1. The Staff selects "add recruiter.".  2. The system displays a blank recruiter form.  3. The Staff enters the following information for the Recruiter: name, address, phone, status, and description.  4. The system validates the data to insure the proper format and searches for an existing Recruiter with the specified name. If the data is valid the system creates a new student and assigns a unique system-generated id number. | | |
| Alternative Flows: | Modify a Recruiter  Issue: Must ensure the flows for modifying and deleting Recruiter are similar to the flows for modifying and deleting professors.  Delete a Recruiter  Issue: Must ensure the flows for modifying and deleting Recruiter are similar to the flows for modifying and deleting professors. | | |
| Exceptions: | Extension points of the business use case will be identified during the Elaboration Phase. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-033. Update Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Update Recruiter | | |
| Create by: | TungBC | Date Created: | 10/12/2018 |
| Primary Actor: | Staff | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Staff can update their profile information. | | |
| Preconditions: | Staff is at “Account” screen. | | |
| Post conditions: | The information of user profile will be saved and updated to “Account” screen. | | |
| Normal Flow: | 1. User changes information of the profile and click “Account” button. 2. System updates information of use’s profile in database and shows new profile to user. | | |
| Alternative Flows: | Alternative flow 1:  1. User inputs data in step 1 violates one or some of business rules.  2. System displays error message(s) on the feedback form. | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | - The First name field is required.  - The Last name field is required.  - The Phone field is required.  - The phone field must be a valid number of telephone | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-034. Delete Recruiter***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Delete Recruiter** | | |
| Create by: | ThoTQ | Date Created: | 10/12/2019 |
| Primary Actor: | Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Staff can delete recruiter | | |
| Preconditions: | 1. Staff can access the system.   2. Staff logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | There are no postconditions associated with this use case. | | |
| Normal Flow: | Delete Recruiter   1. The Staff selects "delete Recruiter." 2. The system displays a blank Recruiter form. 3. The Staff types in the student id number for the student that’s being deleted. 4. The system retrieves the Recruiter and displays the Recruiter information in the form. 5. The Staff selects "delete." 6. The system displays a delete verification dialog confirming the deletion. 7. The Staff selects "yes." 8. The Recruit is deleted from the system. | | |
| Alternative Flows: | Steps 2-8 are repeated for each recruiter deleted from the system. When the Registrar is finished deleting recruiter to the system the use case ends | | |
| Exceptions: | If in the "Modify a Recruiter" or "Delete a Recruiter" sub-flows the student name is not located, the system displays an error message, "Recruiter Not Found". The Registrar can then type in a different id number or cancel the operation at which point the use case ends. | | |
| Priority: | Medium | | |
| Frequency of Use: | Low | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-035. View Notification List***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View Norification** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Staff can view notification list | | |
| Preconditions: | Staff click to “See more” in top of home page | | |
| Post conditions: | N/A | | |
| Normal Flow: | 1. Staff click to “See More” in top of home page 2. The system returns all notification list. 3. Staff can view, hide and delete any notification. | | |
| Alternative Flows: | 1. Staff Click to another notification after that click to view all Notification List  2. The system returns all notification list. | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-036. View Notification Detail***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View Notification Detail** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Staff can view notification detail | | |
| Preconditions: | 1. Staff can access the system.  2. Staff logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | N/A. | | |
| Normal Flow: | View Notification Detail  1. Staff clicks “Notification” buttons in top right Homepage  2. System displays all notification in dropdown list.  3. Recruit click in each notification to see more information of notification | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-037. Export File Excel***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Export file Excel** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Staff can be exported into excel files transcript | | |
| Preconditions: | Staff has logged into the FIMs system. | | |
| Post conditions: | Staff can export all data of website | | |
| Normal Flow: | 1. Staff clicks on kind of data the employee wants to export 2. The page displays the basic list of data 3. Staff click in to “Export Data” button at the bottom of the page 4. The system allows staff to download data to computers using excel files | | |
| Alternative Flows: | 1. At the step 1, instead of clicking data type button, staff clicks to “Back to list” button.  2. System redirects admin to the “View all categories” screen without changing anything at database.  3. System displays error message(s) on the create form. | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-038. View all student internship semester***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | ***View all student internship semester*** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Display all student have been preparing for internship | | |
| Preconditions: | No Preconditions. | | |
| Post conditions: | Guest user click to “student list” from Home page.  System displays the student list screen with necessary elements. | | |
| Normal Flow: | Student list:   * List all student have been preparing for internship * Staff user scroll to see more student in list | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | Medium | | |
| Business Rules: |  | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-039. Search Student***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Search Student** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | User can access the information of one member in the quick way through searching. | | |
| Pre-conditions: | User has logged into the FIMS system. | | |
| Post conditions: | User find out the information of members that he/she wants. | | |
| Normal Flow: | 1. Userclicks *“View all semester”* button at menu home. 2. System shows *“View all semester”* screen. 3. Userclicks the *“Details”* hyperlink of an event. 4. System shows *“Group details”* screen. 5. Userclicks the *“Student”* button to see all group’s member. 6. System shows list group members to user. 7. Userinputs the keyword to the *“Search”* box. 8. System shows *“View all members”* screen with members have the information related to the searching keyword. | | |
| Alternative Flows: | 1. Staff to another year after that click to view all Semester  2. System displays error message(s) on the Pop-up | | |
| Exceptions: | N/A | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-040. Receive Notifications***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Receive Notification | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Show notifications for work management staff | | |
| Preconditions: | 1. Staff can access the system.  2. Staff logins successfully. | | |
| Post conditions: | The staff must have admin rights and account | | |
| Normal Flow: | 1. Staff clicks on “Notification” icon on the top side of the screen.  2. System displays unread notifications.  3. Staff clicks on “View All” button to see all notification.  4. Staff can click on “More” button to manage notification functions. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-041. Delete Student***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Delete Student** | | |
| Create by: | ThoTQ | Date Created: | 10/12/2019 |
| Primary Actor: | Staff | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Staff can delete Student | | |
| Preconditions: | 1. Staff can access the system.   2. Staff logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | There are no postconditions associated with this use case. | | |
| Normal Flow: | Delete Student   1. The Registrar selects "delete student." 2. The system displays a blank student form. 3. The Registrar types in the student id number for the student that’s being deleted. 4. The system retrieves the student and displays the student information in the form. 5. The Registrar selects "delete." 6. The system displays a delete verification dialog confirming the deletion. 7. The Registrar selects "yes." 8. The student is deleted from the system. | | |
| Alternative Flows: | Steps 2-8 are repeated for each student deleted from the system. When the Registrar is finished deleting students to the system the use case ends. | | |
| Exceptions: | If in the "Modify a Student" or "Delete a Student" sub-flows the student name is not located, the system displays an error message, "Student Not Found". The Registrar can then type in a different id number or cancel the operation at which point the use case ends. | | |
| Priority: | High | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-042. Updated Student***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Updated Student** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Allow staff to update students at the beginning of each semester | | |
| Preconditions: | The staff has been granted the right to update students from the admin | | |
| Post conditions: | Staff re-input password before go to “Update Student” | | |
| Normal Flow: | 1. In “Manage Student” page, staff click to “Update Student” button. 2. The system requires employees to fill in a password to verify 3. Staff update student by upload file Excel contain Student info 4. Staff click to update to “update student” follow semester. | | |
| Alternative Flows: | Alternative flow 1:  1. Staff inputs data in step 3 violates one or some of business rules.  2. System displays error message(s) on the Update Student form. | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-043. Add Student***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Add Student** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: | Admin |
| Trigger: | N/A | | |
| Description: | Allow staff to add students at the beginning of each semester | | |
| Preconditions: | 1. Staff can access the system.  2. Staff logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | Postconditions will be determined during the next iteration. | | |
| Normal Flow: | Basic Flow – Add Student:  1. The Staff selects "add student.".  2. The system displays a blank student form.  3. The Registrar enters the following information for the student: name, date of birth, social security number, status, and graduation date.  4. The system validates the data to insure the proper format and searches for an existing student with the specified name. If the data is valid the system creates a new student and assigns a unique system-generated id number. | | |
| Alternative Flows: | Modify a Student  Issue: Must ensure the flows for modifying and deleting students are similar to the flows for modifying and deleting professors.  Delete a Student  Issue: Must ensure the flows for modifying and deleting students are similar to the flows for modifying and deleting professors. | | |
| Exceptions: | Extension points of the business use case will be identified during the Elaboration Phase. | | |
| Priority: | Medium | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-044. Info Student Detail***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Info Student Detail** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Staff | Secondary Actor: | Admin |
| Trigger: | N/A | | |
| Description: | Allow Student to view & edit them information. | | |
| Preconditions: | 1. Staff can access the system.  2. Staff logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | N/A. | | |
| Normal Flow: | Student Detail  1. Student clicks on “My Account” menu on the left side of the screen.  2. System display all fields that Student fill in  3. Student click to “Save” Button to upload student information  4. System displays status “Update information successful”. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-045. Register***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Register** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruiter | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | To allow guest user register an account on the FIM website | | |
| Preconditions: | Guest user click to Register button in top-right page. | | |
| Post conditions: | The Guest user register account successfully. | | |
| Normal Flow: | 1. Guest user click to Register button in top-right page. 2. Fill in all required fields including necessary photos 3. Click button register after fill all information 4. System display registration status. | | |
| Alternative Flows: | Alternative flow 1:  1. User inputs data in step 1 violates one or some of business rules.  2. System displays error message(s) on the feedback form. | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-046. Upload Profile***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Upload Profile | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruiter | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Upload Profile for Recruiter | | |
| Preconditions: | 1. Recruit can access the system.  2. Recruit logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | There are no postconditions associated with this use case. | | |
| Normal Flow: | Upload Profile  1. Recruit clicks on “Profile” on the left side of the screen.  2. System display all fields that Recruit fill in  3. Recruit click to “Save And Up Profile” Button to upload recruitment information  4. System displays all information of profile | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-047. Upload Job***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Upload Job | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruit, Admin | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allow Recruit to post recruitment information. | | |
| Preconditions: | 1. Recruit can access the system.  2. Recruit logins successfully.  3. Account data has been added to the system by the admin | | |
| Post conditions: | There are no postconditions associated with this use case. | | |
| Normal Flow: | Post recruitment information  1. Recruit clicks on “Post Tin” on the left side of the screen.  2. System display all fields that Recruit fill in  3. Recruit click to “Save And Up Post” Button to upload recruitment information  4. System displays all list Job Description in Homepage | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-048. Manage all user and job***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Manage all user and job | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruiter | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Manage all students who are interns at the company and the jobs they are recruiting | | |
| Preconditions: | 1. Recruiter can access the system.  2. Recruiter logins successfully. | | |
| Post conditions: | The Recruiter must have admin rights and account | | |
| Normal Flow: | 1. Recruiter clicks on “Manage Company” icon on the top side of the screen.  2. System displays unread notifications.  3. Staff clicks on “View All” button to see all notification.  4. Staff can click on “More” button to manage notification functions. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-049. Student Evaluation***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Student Evaluation | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruiter | Secondary Actor: |  |
| Trigger: | N/A | | |
| Description: | Allows employers to assess students who have practiced internships at their companies | | |
| Preconditions: | 1. Recruiter can access the system.  2. Recruiter logins successfully. | | |
| Post conditions: | Students have practiced in this company of recruiter | | |
| Normal Flow: | 1. Recruiter clicks on “Manage student internships” on the left side of the “Company Detail” screen.  2. System displays table of all Student internships.  3. In the “Đánh Giá” text box, recruiter enters text.  4. In the box “Score” recruiter choose the number of score recruiter want to for student grades. 5. Recruiter click “Save” button to upload Student Evaluation of company | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-050. View all job opening***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **View All Job Opening** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruiter | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Allows employers to view all job openings on the website | | |
| Preconditions: | Recruiter click to “View Job” in bottom of home page | | |
| Post conditions: | N/A | | |
| Normal Flow: | 1. Staff click to “View Job” in bottom of home page 2. System show all job opening 3. Staff click on any one year to show 4 semester in each year 4. Staff click to specific semester to see details and internships for that semester | | |
| Alternative Flows: | 1. Staff to another year after that click to view all Job opening  2. System displays error message(s) on the Pop-up | | |
| Exceptions: | N/A | | |
| Priority: | Low | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-051. Recruiter Aspirations***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | Recruiter Aspirations | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruiter | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Information about the aspirations of the Recruiter | | |
| Preconditions: | Recruiter logins successfully. | | |
| Post conditions: | There are no postconditions associated with this use case. | | |
| Normal Flow: | 1. Recruiter clicks on “Manage Account” icon on the top side of the screen bar.  2. Recruiter displays all type of account button.  3. Recruiter click on any type of user to manage accounts.  4. Recruiter click to Aspirations fill and fill all information.  5. Recruiter click to “Save” button to save all role of user after changed. | | |
| Alternative Flows: | N/A | | |
| Exceptions: | Cannot communicate with server 1. FIMS displays error message. | | |
| Priority: | Medium | | |
| Frequency of Use: | Medium | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-052. Post Job***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Post Jobs** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Recruiter | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | To upload jobs provided by different companies | | |
| Preconditions: | A company going to login to their account and upload vacanies. | | |
| Post conditions: | The company uploads the new vacancy successfully. | | |
| Normal Flow: | 1. The company login to their account . 2. The system checks the account validation and returns with the result of login. 3. The company go to upload tab and fill out the new vacancy details and click submit . 4. The system upload job into their database and displays message. 5. The “Job uploaded successfully. | | |
| Alternative Flows: | Alternative flow 1:  1. User inputs data in step 1 violates one or some of business rules.  2. System displays error message(s) on the feedback form. | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

***UC-053. ChatBox***

*Use Case Specification*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name:** | **Chatbox** | | |
| Create by: | ThoTQ | Date Created: | 10/11/2019 |
| Primary Actor: | Student, Recruit | Secondary Actor: | N/A |
| Trigger: | N/A | | |
| Description: | Interactive student with recruiter | | |
| Preconditions: | Student click to “Chatbox” in right bottom of Recruit Detail page | | |
| Post conditions: | Student & Recruiter logged in successfully | | |
| Normal Flow: | 1. Students are viewing the recruit detail page 2. Student click to “Chatbox” in right bottom of Recruit Detail page 3. Chatbox will show and Recruiter & Student can be chat and information exchange 4. Student click “X” to close the chatbox. | | |
| Alternative Flows: | 1. Student Click to button “Chat with Recruiter” in the “Recruit List” page  2. System displays status message(s) on the pop-up form. | | |
| Exceptions: | N/A | | |
| Priority: | High | | |
| Frequency of Use: | High | | |
| Business Rules: | N/A | | |
| Other Information: | N/A | | |
| Assumptions: | N/A | | |

# 3.3 Non-functional Requirement

## **3.3.1 Security**

- The system must use token-based authentication, in which the authentication server returns an access token user logged in with email.

- The security matrix is as the following table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function** | **Staff** | **Recruiter** | **Student** | **Admin** | **Guest** |
| Login | V | V | V | V |  |
| Logout | V | V | V | V |  |
| Reset Password | V | V | V | V |  |
| Change Password | V | V | V | V |  |
| View Homepage | V | V | V | V | V |
| Review Company | V | V | V | V | V |
| View Job | V | V | V | V |  |
| View Info Detail Job | V | V | V | V |  |
| View Recruiter | V | V | V | V |  |
| View Info Detail Recruiter | V | V | V | V |  |
| Search Account |  |  |  | V |  |
| Filter Account |  |  |  | V |  |
| View Account List |  |  |  | V |  |
| Create Account |  |  |  | V |  |
| Change Account Status |  |  |  | V |  |
| Change Account Rule |  |  |  | V |  |
| Delete Account |  |  |  | V |  |
| View Detail Account |  |  |  | V |  |
| Edit Account Role |  |  |  | V |  |
| View Top Recruiter |  |  |  |  | V |
| View Department |  |  |  |  | V |
| Search FSIMS |  |  |  |  | V |
| Upload Profile |  |  | V |  |  |
| Student Aspirations |  |  | V |  |  |
| Apply Job |  |  | V |  |  |
| Search Job |  |  | V |  |  |
| Search Recruiter |  |  | V |  |  |
| Feed Back | V |  | V |  |  |
| Interview Schedule |  | V | V |  |  |
| Recomment |  | V | V |  |  |
| View Info Recruiter | V |  |  |  |  |
| Add Recruiter | V |  |  |  |  |
| Updated Recruiter | V |  |  |  |  |
| Delete Recruiter | V |  |  |  |  |
| View notification list | V |  |  |  |  |
| View notification detail | V |  |  |  |  |
| Export File Excel | V |  |  |  |  |
| View all student internship semester | V |  |  |  |  |
| Search Student | V |  |  |  |  |
| Receive Notifications | V |  |  |  |  |
| Delete Student | V |  |  |  |  |
| Updated Student | V |  |  |  |  |
| Add Student | V |  |  |  |  |
| Info Student Detail | V |  |  |  |  |
| Register |  | V |  |  |  |
| Upload Profile |  | V |  |  |  |
| Upload Job |  | V |  |  |  |
| Manage all user and job |  | V |  |  |  |
| Student Evaluation |  | V |  |  |  |
| View all job opening |  | V |  |  |  |
| Recruiter Aspirations |  | V |  |  |  |
| Post Job |  | V |  |  |  |
| Chat Box |  | V | V |  |  |

## **3.3.2 Supportability**

The application should be easy to extend.

## **Accuracy**

The overall accuracy is calculated by diving total number of correct answers by the number of questions asked.

## **3.3.4 Maintainability and Extensibility**

Source code must follow the coding convention in order to help developers to read, understand and find it easy to maintain or modify the code.

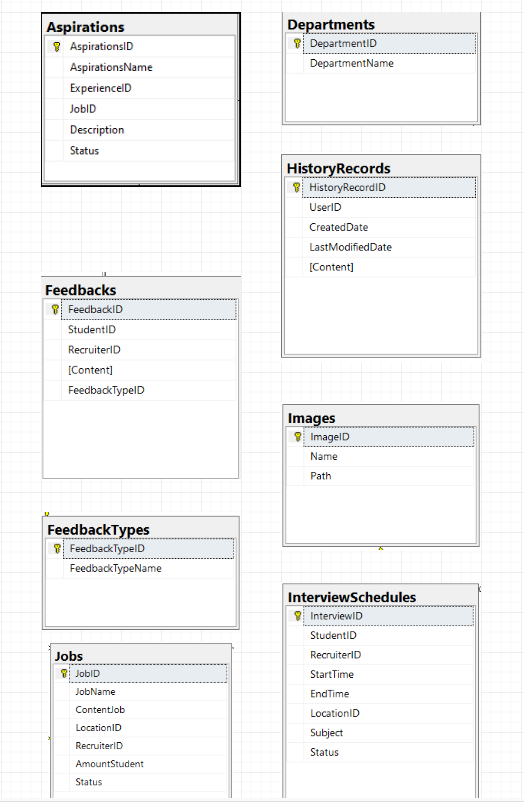
## **3.3.5 Performance**

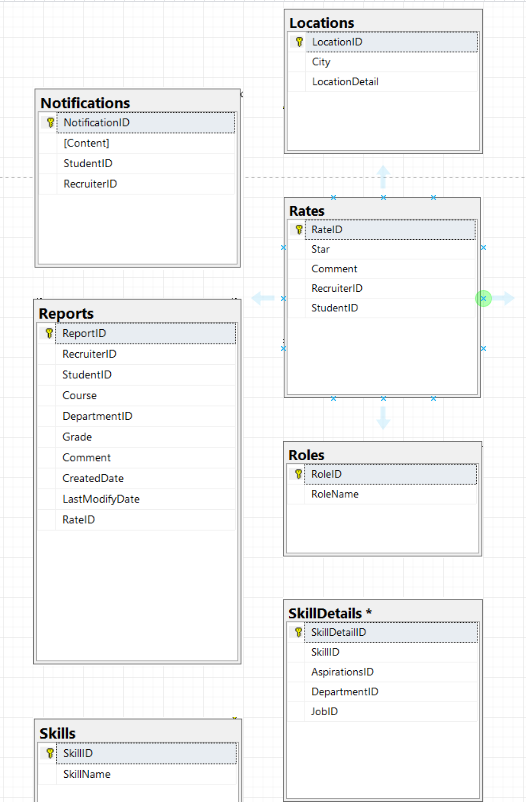
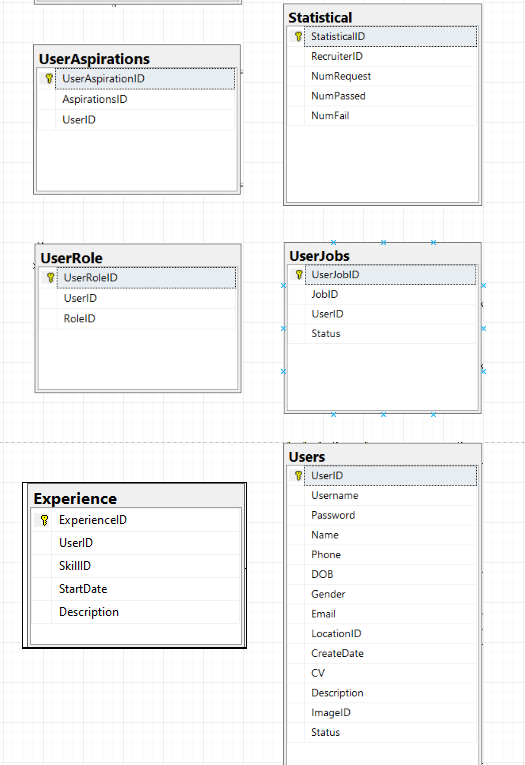
* Since this software is going to web – based, it does require a powerful server machine with high band internet access.
* Performance requirement by the user side is, web application should be developed as a lightweight web app so that it can work on almost any platform even with slower internet connections.
* Expected number of simultaneous user should be at least 1000. System should be able to deal with 1000 users at the same time. Also database of the system should handle at least a thousand of users at any periods

## **3.3.6 Usability**

* The interface is friendly, simple and easy to use.
* The steps are short and easy to understand so that the user can interact with the application without training.
* Main functions are organized into tabs for easier access.
* Links, buttons, and checkboxes are easily clickable.
* Important commands are displayed as buttons with distinct background colors.
* The screen layout and color should be appealing.
* The guideline in “FIMS Document” are clearly and easy to understand.

# **3.4 Database document diagram**



# **CHAPTER 4: SOFTWARE DESGIN**

## 4.1 Purpose

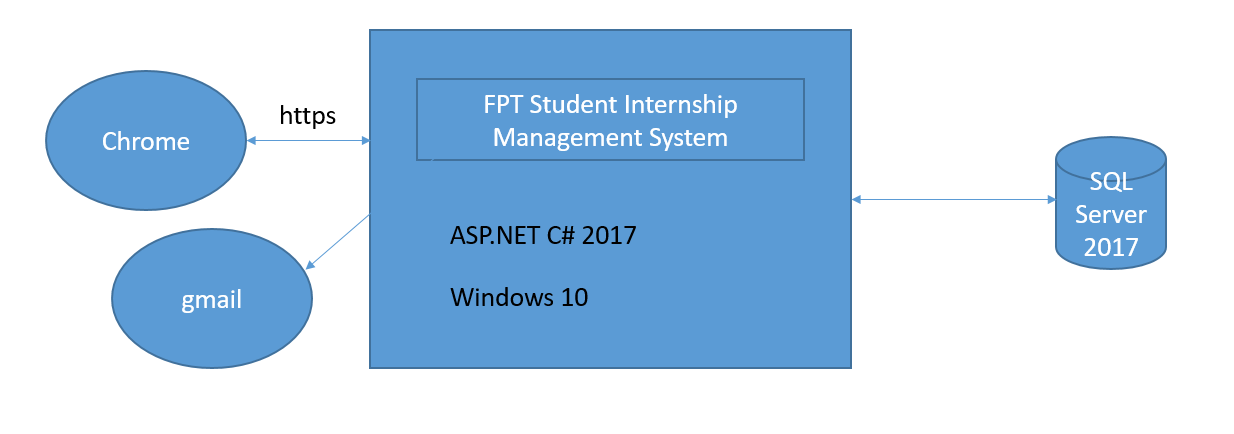
This chapter is to give the developer team an overview of what the system’s architecture is, and how they should be implemented. This chapter consists of:

* Architecture overview
* Component diagram
* Detailed design
* Detailed description of components
* Database design

## 4.2 Architectural overview

### 4.2.1 System Architecture

#### 4.2.1.1 Diagram



*Figure 4-1: System overview*

### 4.2.2 System Architectural Explanation

#### 4.2.2.1 Sever application

#### 4.2.2.2 Presentation tier

##### 4.2.2.2.1 Web application – HTML5 & CSS3



*Figure 4-2: HTML & CSS*

Using **HTML5 and CSS3** provides advantages to businesses that develop and deploy web content and web applications, to create refined and accurate web pages and web systems that **operate across devices, operating systems and web browsers**.

#### 4.2.2.3 Data tier

##### 4.2.2.3.1 SQL Database – SQL Server



*Figure 4-3: SQL Server 2017*

**SQL Sever 2017** which was be developed, distributed and supported by Microsoft Corporation. It is a powerful and reliable free data management system that delivers a rich and reliable data store for lightweight Web Sites and desktop applications.

#### 4.2.2.4 Service providers

##### 4.2.2.4.1 Mail service



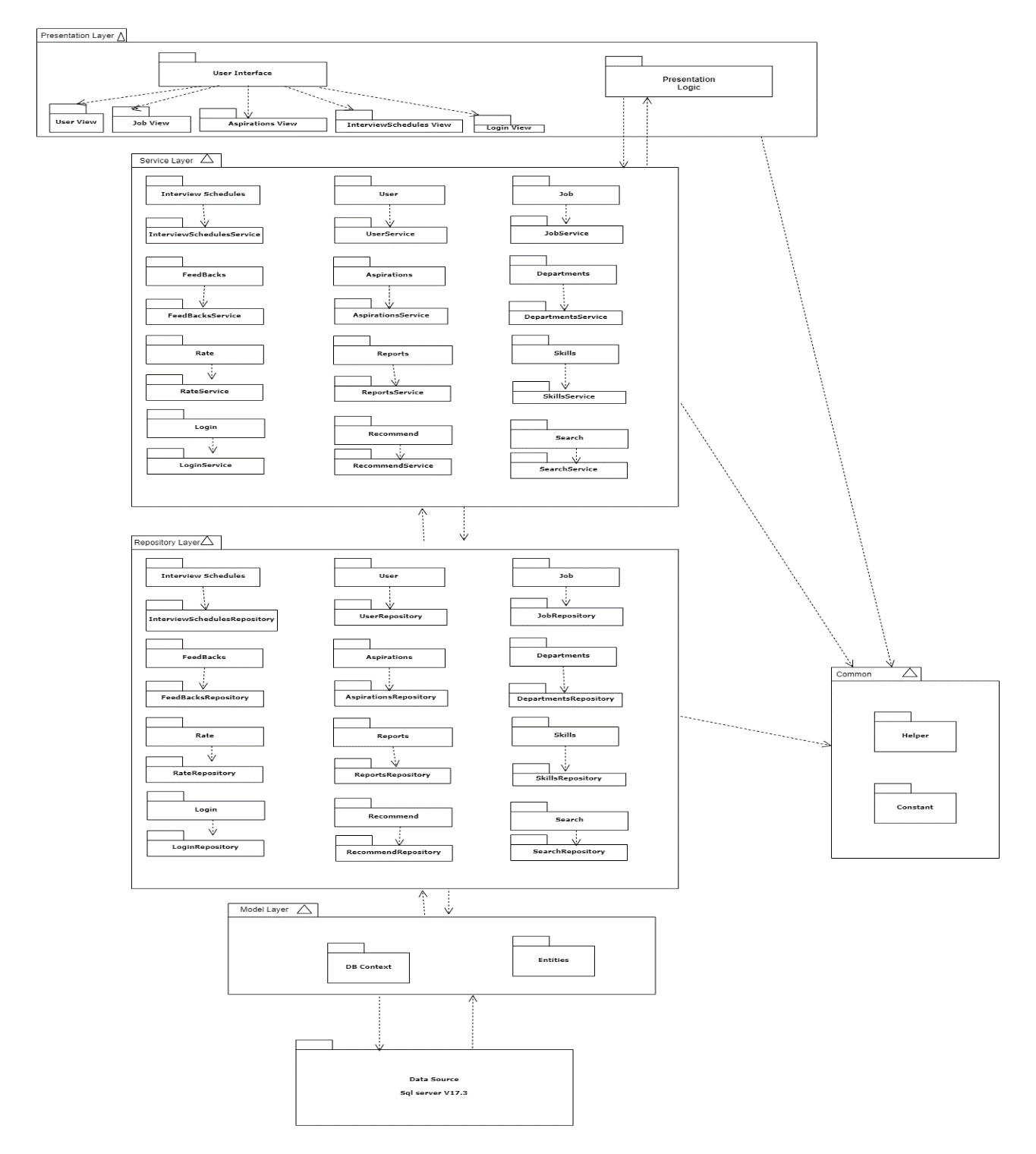
*Figure 4-4: Gmail*

* User gmail @fpt.edu.vn login in system

## 4.3 Design of FPT Internship Management System

### 4.3.1 Architectural Layers Design

#### 4.3.1.1 FIMS – API Layers Design

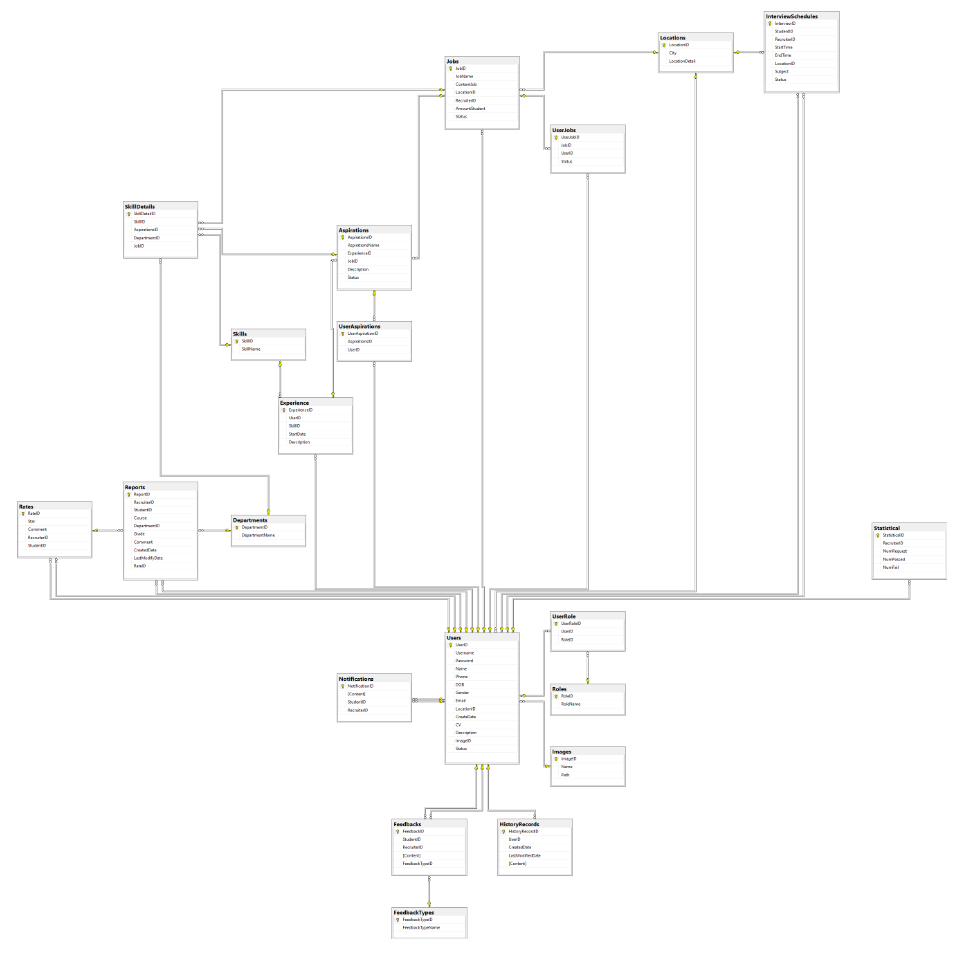


*Figure 4-5:FIMS- API Layers Design*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Directory | Description | Convention |
| 1 | /Controllers | Where we put Controller classes that handle URL request. | N/A |
| 2 | /Models | Where we put classes that represent and manipulate data and business objects. | N/A |
| 3 | /Views | Where we put UI template files that are responsible for rendering output, such as HTML | N/A |
| 4 | /Scripts | Where we put JavaScript library files and scripts (.js) | N/A |
| 5 | /Images | Where we put images used in our site. | N/A |
| 6 | /Content | Where we put CSS and other sit content, other than scripts and images. | N/A |
| 7 | /Helper | Where we put Helper classes that helps handle some business logic | N/A |
| 8 | /Jobs | Where we put code for features about status of issues. | N/A |
| 9 | /App\_Data | Where we store data files we want to read/ write | N/A |
| 10 | /App\_Start | Where we put configuration code for features like Routing, Bundling and Web API. | N/A |

### 4.3.2 Database Design

#### 4.3.2.1 Database diagram



#### 4.3.2.2 Explanation of database decision

As our system have following characteristic:

* Users manage data frequently on dashboard.
* Flexible data model.
* System may need to scale.
* Provide real-time experience for statistics

So, we chose SQL Sever because:

* SQL Sever is free and open-source.
* Install different versions of MS SQL on the same machine.
* Maintain separate production, development, and testing environments.
* Minimize temporary database problems.
* Separate security privileges.
* Maintain a backup server.

#### 4.3.2.3 Data Dictionary

##### 4.3.2.3.1 Table description

|  |  |  |
| --- | --- | --- |
| **No** | **Table Name** | **Description** |
| 1 | Aspirations | Save aspirations of student and recruiter |
| 2 | Departments | However, this project is save currently working on software engineering |
| 3 | Experience | Save experience of student when adding Aspirations, 1 Aspirations has 1 Experience, 1 experience has many skills -> this board of students |
| 4 | Feedbacks | Feedback of student about recruiter after send for staff |
| 5 | FeedbackTypes | Categorize the type of students sent to the staff or recruiter sent to the staff |
| 6 | HistoryRecords | Save history with transactions such as adding students, adding aspirations ... |
| 7 | Images | Save path images |
| 8 | InterviewSchedules | Save information of interview schedule book |
| 9 | Jobs | Save the job information of the employer who wants to recruit ... |
| 10 | Locations | Save address information |
| 11 | Notifications | Save notified the school to students, or recruiter to students |
| 12 | Rates | Save student votes for hiring |
| 13 | Reports | Send evaluation information about recruiter to students. |
| 14 | Roles | 4 role: admin, staff, recruiter, student. |
| 15 | SkillDetails | Denotes the many skills for department store, for job |
| 16 | Skills | Example: java, C#, python,… |
| 17 | Statistical | Statistics of the school, the total number of guys requesting to enter the company, the total number of guys failing, the total number of passes |
| 18 | UserAspirations | to help 1 user save a lot of wishes |
| 19 | UserJobs | to help a user save multiple jobs |
| 20 | UserRole | To save which user corresponds to which role |
| 21 | Users | Save information users. |

##### 4.3.2.3.2 Table Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Collection** | **Field** | **Type** | **Allow Nulls** | **Description** |
| Aspirations | AspirationsID | Int |  |  |
| AspirationsName | Nvarchar(50) |  |  |
| ExperienceID | int |  |  |
| JobID | int | Yes |  |
| Description | Nvarchar(50) | Yes |  |
| Status | Nvarchar(50) | Yes |  |
| Departments | DepartmentsID | int |  |  |
| DepartmentsName | Nvarchar(50) |  |  |
| HistoryRecords | HistoryRecordsID | int |  |  |
| UserID | Int |  |  |
| CreateDate | Date |  |  |
| LastModifiedDate | Date |  |  |
| [Contend] | Nvarchar(max) |  |  |
| Feedbacks | FeedbacksID | Int |  |  |
| StudentID | Int |  |  |
| RecruiterID | Int |  |  |
| [Content] | Nvarchar(max) |  |  |
| FeedbackTypeID | Int |  |  |
| FeedbackTypes | FeedbackTypeID | Int |  |  |
| FeedbackTypeName | Nvarchar(50) |  |  |
| Images | ImageID | Int |  |  |
| Name | Nvarchar(50) |  |  |
| Path | Nvarchar(50) |  |  |
| InterviewSchedules | InterviewID | Int |  |  |
| StudentID | Int |  |  |
| RecruiterID | Int |  |  |
| StartTime | Datetime |  |  |
| EndTime | Datetime |  |  |
| LocationID | Int |  |  |
| Subject | Nvarchar(50) |  |  |
| Status | Nvarchar(50) |  |  |
| Jobs | JobID | Int |  |  |
| JobName | Nvarchar(50) |  |  |
| ContendJob | Nvarchar(50) | Yes |  |
| LocationID | Int | Yes |  |
| RecruiterID | Int | Yes |  |
| AmountStudent | Int | Yes |  |
| Status | Nvarchar(50) | Yes |  |
| Locations | LocationID | Int |  |  |
| City | Nvarchar(50) |  |  |
| LocationDetail | Nvarchar(50) |  |  |
| Notifications | NotificationID | Int |  |  |
| [Content] | Nvarchar(max) |  |  |
| StudentID | Int |  |  |
| RecruiterID | Int |  |  |
| Rates | RateID | Int |  |  |
| Star | Float |  |  |
| Comment | Nvarchar(50) |  |  |
| Recruiter | Int |  |  |
| StudentID | Int |  |  |
| Reports | ReportID | Int |  |  |
| RecruiterID | Int | Yes |  |
| StudentID | Int | Yes |  |
| Course | Nvarchar(50) | Yes |  |
| DepartmentID | Int | Yes |  |
| Grade | Float | Yes |  |
| Comment | Nvarchar(50) | Yes |  |
| CreatedDate | Date | Yes |  |
| LastModifyDate | Date | Yes |  |
| RateID | Int | Yes |  |
| Roles | RoleID | Int |  |  |
| RoleName | Nvarchar(50) |  |  |
| SkillDetails | SkillDetailID | Int |  |  |
| SkillID | Int |  |  |
| AspirationsID | Int | Yes |  |
| DepartmentID | Int |  |  |
| JobID | Int | Yes |  |
| Skills | SkillID | Int |  |  |
| SkillName | Nvarchar(50) |  |  |
| Statistical | StatisticalID | Int |  |  |
| RecruiterID | Int |  |  |
| NumRequest | Int |  |  |
| NumPassed | Int |  |  |
| NumFail | Int |  |  |
| UserAspirations | UserAspirationID | Int |  |  |
| AspirationsID | Int |  |  |
| UserID | Int |  |  |
| UserJobs | UserJobID | Int |  |  |
| JobID | Int |  |  |
| UserID | Int |  |  |
| Status | Nvarchar(50) |  |  |
| UserRole | UserRoleID | Int |  |  |
| UserID | Int |  |  |
| RoleID | Int |  |  |
| Users | UsersID | Int |  |  |
| UserName | Nvarchar(50) |  |  |
| Password | Nvarchar(50) |  |  |
| Name | Nvarchar(50) | Yes |  |
| Phone | Int | Yes |  |
| DOB | Date | Yes |  |
| Gender | Bit | Yes |  |
| Email | Nvarchar(50) | Yes |  |
| LocationID | Int | Yes |  |
| CreatedDate | Date | Yes |  |
| CV | Nvarchar(50) | Yes |  |
| Description | Nvarchar(50) | Yes |  |
| ImageID | Int | Yes |  |
| Status | Nvarchar(50) | Yes |  |
| Experience | ExperienceID | Int |  |  |
| UserID | Int |  |  |
| SkillID | Int |  |  |
| StartDate | Date |  |  |
| Description | Nvarchar(50) |  |  |

*Table 4-2: FIMS data dictionary*

## 4.4 Detail Design

### 4.4.1 Authentication

#### 4.4.1.1 Login

#### 4.4.1.2 Logout

#### 4.4.1.3 Register

# **CHAPTER 5: SOFTWARE TESTING DOCUMENT**

## **5.1 Introduction**

### 5.1.1 Purpose

The test plan contains a detailed understanding of the workflow and functions of the system and documents how each of those will be tested in order to find out if the system works according to its design, to find bugs, and to determine its actual limitations. Using the contents of this chapter, developers and testers can work together to ensure that the system is thoroughly tested and functions as intended. It contains the following sections:

* Test strategy
* Test plan
* Test approach
* Defect log
* Test report

## **5.2 Test strategy**

### 5.2.1 Objective and scope

The objective of testing is to satisfy all the requirement of the business use cases. The scope of this test will be limited to testing FIMS web application. The quality goal is 100% test cases are executed and 98% passed.

### 5.2.2 Testing tools and environment

#### 5.2.2.1 Testing tools

The following tools are used for testing:

**Google Chrome:** Use to view the web page, bug logging page



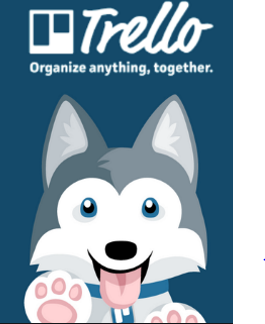
*Figure 5-1: Google Chrome* version *78.0.3904.108*

**Microsoft Excel:** Use to manage test cases



*Figure 5-2: Microsoft Excel 2016*

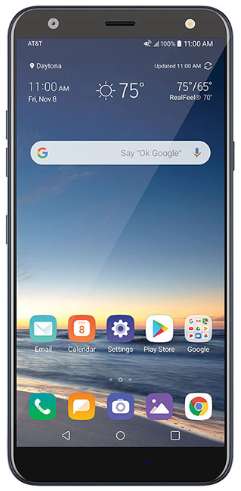
**Trello:** User to manage bug



*Figure 5-3: Trello*

**MSTest:** Use to perform unit tests

**SmartPhone:** Use to view the web page, bug logging page, etc



*Figure 5-4: Smartphone*

#### 5.2.2.2 Testing environment

|  |  |  |
| --- | --- | --- |
| **Type of testing** | **Software** | **Hardware** |
| Unit test | MSTest | Laptop Lenovo Thinkpad:  o Windows 10 Pro  o Intel core i7 2.6GHz  o 8GB Memory  o Screen resolution: 1920 x 1080 |
| Integration test | -Google chrome version 78.0.3904.108  - Microsoft Office Excel 2016.  - Microsoft Office Word 2016 | Laptop Lenovo Thinkpad:  o Windows 10 Pro  o Intel core i7 2.6GHz  o 8GB Memory  o Screen resolution: 1920 x 1080 |
| System test | -Google chrome  version 78.0.3904.108  - Microsoft Office Excel 2016.  - Microsoft Office Word 2016 | Laptop Lenovo Thinkpad:  o Windows 10 Pro  o Intel core i7 2.6GHz  o 8GB Memory  o Screen resolution: 1920 x 1080 |

### 5.2.3 Resources and responsibilities

|  |  |  |
| --- | --- | --- |
| No | Resources | Responsibilities |
| 1 | Project manager | * Responsible for project schedules and overall success of the project. |
| 2 | Test leader | * Manage overall of the software quality. * Define test strategy. * Define test plan, test approach. * Create and review test case, defect log. |
| 3 | Tester | * Understand the requirements of the project. * Implement integration test following requirement. * Create test cases. * Execute test. * Log bug. |
| 4 | Developer | * Create unit test * Fix bug |

### 5.2.4 Acceptance criteria

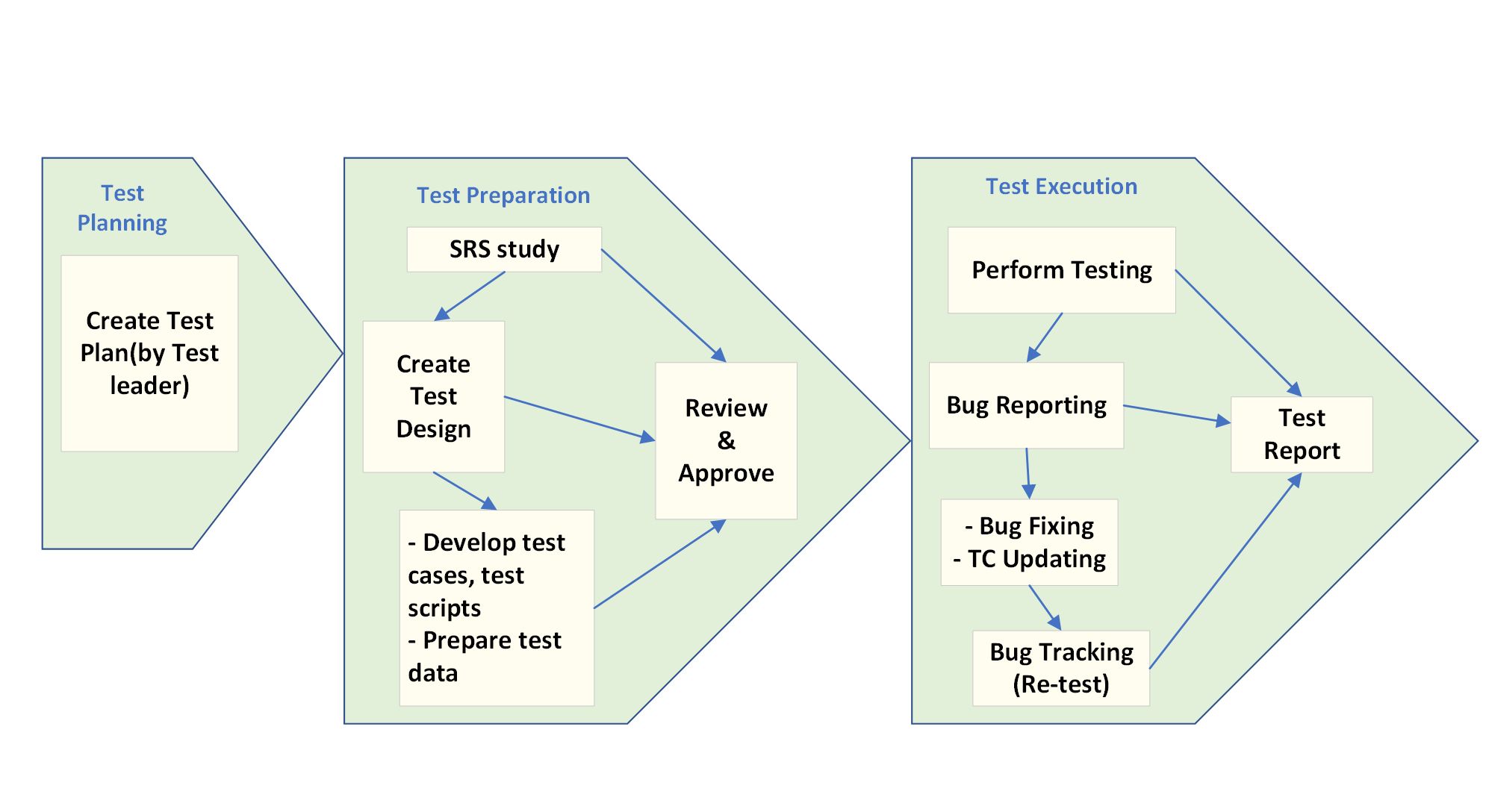
|  |  |  |
| --- | --- | --- |
| No | Acceptance criteria | Note |
| 1 | 100% test cases are executed |  |
| 2 | 98% test cases are passed |  |
| 3 | 95% of medium severity defects have been closed |  |
| 4 | 100% of high severity defects have been closed |  |

### 5.2.5 Risk List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Condition | Mitigation | Contingency | Doer |
| 1 | The hardware maybe corrupted when testing | Change hardware | Re-plan schedule | TungBC |

### 5.2.6 Test Process

The test process is following the FPT-Software process.



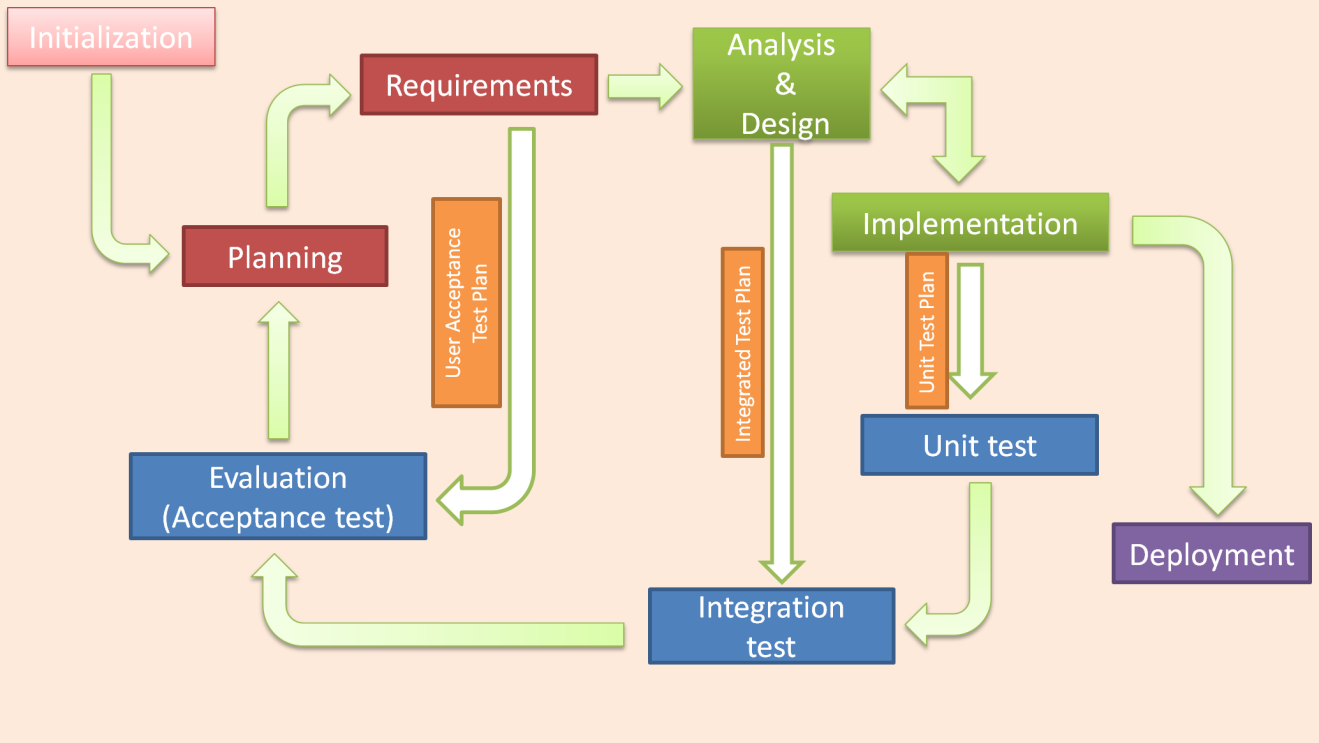
*Figure 5-1.Test process*

**Description:**

1. **Test plan:** Test leader takes responsible to create a test plan.
2. **Test preparation:** Testing team study the SRS document, create test design for each stage.
3. **Test execution:** Tester and developer perform the test following the plan. The bug is logged by tester and developer is assigned to fix the code in order to fix the bugs.

## **5.3 Test Plan**

### 5.3.1 Test Model



*Figure 5‑2.Iterative testing model*

### 5.3.2 Stage of testing

|  |  |  |  |
| --- | --- | --- | --- |
| No | Testing level | Description | Implementation |
| 1 | Unit test | * **Unit test is performed by developers.** * **All the test cases are NOT documented.** | Development team:   1. Bùi Trug Kiên 2. Bùi Công Tùng 3. Đỗ Đình Đức 4. Trần Quang Thọ |
| 2 | Integration test | * **Integration test is performed by testers.** * **Material is “integration\_test\_case\_v2.0.xls”.** | Tester team:   1. Trần Quang Thọ 2. Bùi Công Tùng 3. Nguyễn Hữu Hóa 4. Đỗ Đình Đức |
| 3 | System test | * **System test is performed by testers.** * **Material is “System\_test\_case.xls”.** | Tester team:   1. Trần Quang Thọ 2. Bùi Công Tùng 3. Nguyễn Hữu Hóa 4. Đỗ Đình Đức |
| 4 | Acceptance test | Acceptance test is performed by BA.  Material “User requirement”, “Check list”. | BA team:   1. Bùi Trung Kiên |
| No | Testing level | Description | Implementation |
| 1 | Unit test | * **Unit test is performed by developers.** * **All the test cases are NOT documented.** | Development team:   1. Bùi Trung Kiên 2. Bùi Công Tùng 3. Đỗ Đình Đức 4. Trần Quang Thọ |
| 2 | Integration test | * **Integration test is performed by testers.** * **Material is “integration\_test\_case\_v2.0.xls”.** | Tester team:   1. Trần Quang Thọ 2. Bùi Công Tùng 3. Nguyễn Hữu Hóa 4. Đỗ Đình Đức |
| 3 | System test | * **System test is performed by testers.** * **Material is “System\_test\_case.xls”.** | Tester team:   1. Trần Quang Thọ 2. Bùi Công Tùng 3. Nguyễn Hữu Hóa 4. Đỗ Đình Đức |
| 4 | Acceptance test | Acceptance test is performed by BA.  Material “User requirement”, “Check list”. | BA team:   1. Bùi Trung Kiên |

### 5.3.3 Function to be tested

No comment

### 5.3.4 Function not to be tested

No comment

### 5.3.5 Type of testing

The following type of testing is used in this project are:

|  |  |  |
| --- | --- | --- |
| No | Type of test | Description |
| 1 | Automation unit test | * Testing on code level to identify bugs in functions. |
| 2 | GUI test | * GUI test is used to test the user interface of the software. It will be performed fully on all screens. The target is to cover the verification of mocks defined before. It also includes fields:  1. Check if error messages are displayed correctly. 2. Check if Font is readable. 3. Check the screen solution |
| 3 | Functional test | * Functional test is used to test for each logical path of use case. |
| 4 | Acceptance test | * Passing all the use case without critical errors. |

The table below are the stages in which test are executed:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of test | Stage of test | | | |
| Unit test | Integration test | System test | Acceptance test |
| Automation unit test | √ |  |  |  |
| GUI test |  | √ | √ | √ |
| Functional test |  | √ | √ | √ |
| Acceptance test |  |  |  | √ |

### 5.3.6 Test Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Task | | Start date | End date |
| Iterator 1 | Create test plan | 09/15/2019 | 16/10/2019 |
| Create and execute unit test | 09/18/2019 | 23/10/2019 |
| Create and execute integration test | 09/18/2019 | 22/10/2019 |
| Create and execute system test | 09/25/2019 | 26/10/2019 |
| Create and execute acceptance test | 09/28/2019 | 28/10/2019 |
| Iterator 2 | Create test plan | 09/30/2019 | 02/11/2019 |
| Create and execute unit test | 11/02/2019 | 07/11/2019 |
| Create and execute integration test | 11/10/2019 | 17/11/2019 |
| Create and execute system test | 11/18/2019 | 20/11/2019 |
| Create and execute acceptance test | 11/22/2019 | 24/11/2019 |

### 5.3.7 Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| No | Deliverables | Responsibilities | Delivered date |
| 1 | Test plan | Trần Quang Thọ | 02/11/2019 |
| 2 | Unit test | Bùi Công Tùng, Trần Quang Thọ, Nguyễn Hữu Hóa | 07/11/2019 |
| 3 | Integration test | Bùi Công Tùng, Nguyễn Hữu Hóa, Bùi Trung Kiên | 17/11/2019 |
| 4 | System test | Đỗ Đình Đức | 20/11/2019 |
| 5 | Acceptance test | Bùi Trung Kiên | 24/11/2019 |
| 6 | Defect logs | Đỗ Đình Đức | 09/12/2019 |
| 7 | Final test summary report | Trần Quang Thọ | 11/12/2019 |

## **5.4 Test Approach**

### 5.4.1 Unit Test

No Comment

#### 5.4.1.1 Implement junit library

No Comment

#### 5.4.1.2 Create Test Case and Test Method

No Comment

#### 5.4.1.3 Execute Unit Test

No Comment

#### 5.4.1.4 Test Result

No Comment

### 5.4.2 Integration Test

### 5.4.3 Acceptance Test

The purpose of this test is to having end-user perception while testing and understand business need for each requirement and test accordingly. But our project will use the checklist as a substitute for acceptance testing.

The content of the checklist is shown in the table below:

|  |  |  |
| --- | --- | --- |
| ID | CheckLists | Status |
| **General** | | |
| AT-001 | All mandatory fields are validated. | Done |
| AT-002 | All error messages are displayed in red color. | Done |
| AT-003 | Delete functionality for any record on a page are asked for confirmation. | Done |
| AT-004 | Text on all pages for spelling and grammatical errors. | Done |
| AT-005 | Application crash or unavailable pages are redirected to an error page. | Done |
| **GUI & USABILITY** | | |
| AT-006 | The screen is well organized and easy to use. | Done |
| AT-007 | All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly. | Done |
| AT-008 | The most important fields are located where they are easy to see. | Done |
| AT-009 | Information is presented in the order that the user needs it. | Done |
| AT-010 | The screen designed to fit the requirements for international use. | Done |
| AT-011 | The static text is clear, concise, and meaningful. | Done |
| AT-012 | System display notification message when meeting trouble, error. | Done |
| **DATABASE** | | |
| AT-013 | Correct data is getting saved in the database upon a successful page submit. | Done |
| AT-014 | Values columns are not accepting null values. | Done |
| AT-015 | Data should be stored in single or multiple tables based on design. | Done |
| AT-016 | Database fields are designed with correct data type and data length. | Done |
| **SECURITY** | | |
| AT-017 | Test cookie allowed (disabled or allowed to be edited) | Done |
| AT-018 | Check login, logout functionality. | Done |

### 5.4.4 Defect Log

FSIMS project using Trello to manage bug . Processes are:

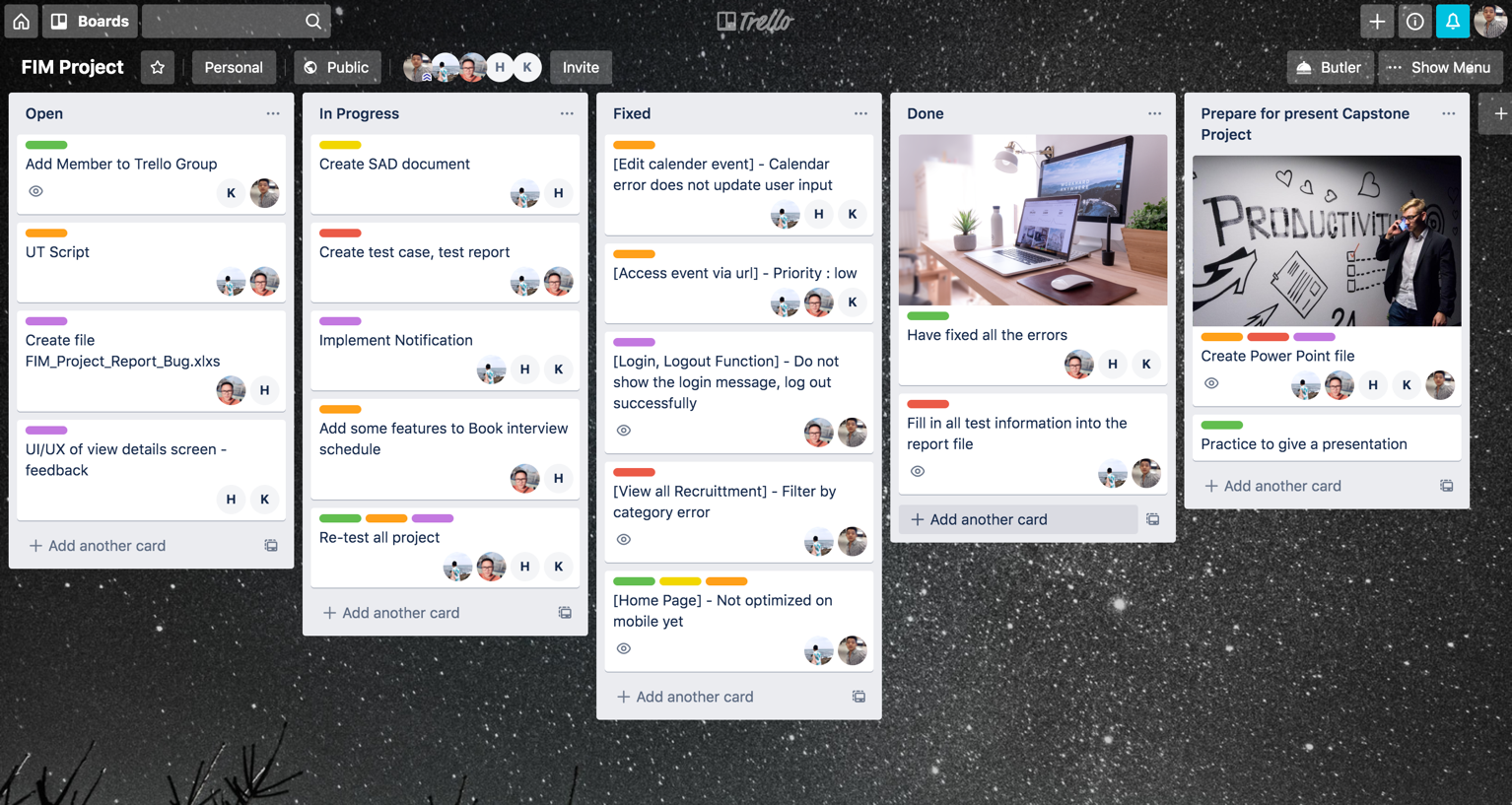
- During the testing process, when a bug is found, testers access trello.com to log in with the “Bug” label (highlight color).

- The bug must be described as detailed as possible with screenshots.

- The issue is then assigned to a developer, who is in charge of developing that module.

- Assigned developer checks out the issue in trello.com, move that card to “In Progress” List and fixes it. Once the bug is resolved, the assigned developer drag that card to “Fixed” List

- The tester then verifies if the bug in “Fixed” List. If the bug is fixed, drag that card to “Done” List. If the bug is not fixed, tester move card to “Open” List and assign it to the developer again.



*Figure 10 : Defect log management tool.*

## **5.5 Test Report**

### 5.5.1 Unit Test Report

No Comment

### 5.5.2 Integration Test Report

No Comment

# **Chapter 6: User Manual**

# 6.1 Deployment guidelines

## 6.1.1 Environment for development

## 6.1.2 Environment for deployment

# 6.2 User Guidelines

## 6.2.1 Guest Actor

## 6.2.2 Student Actor

## 6.2.3 Staff Actor

## 6.2.4 Admin Actor

## 6.2.5 Recruiter Actor