

|  |
| --- |
|  |

|  |
| --- |
| Capstone Project Final Report |
| **Internet-connected Devices Checking System** |
|  |
| |  |  |  | | --- | --- | --- | | **IDCS TEAM** | | | | **Group Members** | Nguyễn Quý Đôn | SE04468 | | Nguyễn Quý Tuấn | SE04330 | | Phạm Công Minh | SE04098 | | **Supervisor** | Lecture : Phan Duy Hùng | | |
|  |

|  |
| --- |
| - Hanoi, Dec 2019 - |

**TABLE OF CONTENTS**

**DEFINITIONS AND ACRONYMS**

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Note |
| **IDCS** | Internet-connected Devices Checking System |  |
| **UC** | Use case |  |
| **ERD** | Entity Relationship Diagram |  |
| **GUI** | Graphic User Interface |  |
| **UT** | Unit Test |  |
| **N/A** | Not available |  |
| **PM** | Project Manager |  |
| **Q&A** | Question and Answer |  |
| **MVC** | Model – View – Controller |  |
| **SRS** | Software Requirement Specification |  |
| **URL** | Universal Resource Locator |  |
| **UI** | User Interface |  |
| **DBMS** | Database Management System |  |

**DOCUMENT CONVENTION**

**Date Formatting**

All date in the document are written in yyyy/mm/dd format.

For example, 2019/08/31 means August 31st, 2018.

**Tables and Figures**

Tables and figures are numbered in the format (chapter number) – (# of table or figures).

For example:

* “Table 2-1” means the first table of the second chapter.
* “Figure 3-2” means the second figure of the third chapter.

1. **INTRODUCTION**
   1. ***Purpose***

This chapter is created as the introduction to Internet-connected Devices Checking System project, it is also a capstone project of team at FPT University. This document describes the overview of some existing systems, the initial idea, brief description about expected system and provides an overview of the project, including background information, review of existing systems, their problem and proposal for ideas of improvement.

* 1. ***Project Information***
* Project name: Internet-connected Devices Checking System
* Project code: IDCS
* Project group name: IDCS Team
* Project Type: Web Application
* Business domain: Security Checking System
  1. ***Project Team***
     1. ***Supervisor***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Full Name | Phone | Email | Title |
| Supervisor | Phan Duy Hùng |  | HungPD2@fe.edu.vn | Lecture |

Table 1-1 : Supervisor's Information

* + 1. ***Team member***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Full Name | Student ID | Phone | Email | Role |
| 1 | Phạm Công Minh | SE04098 | 0969307911 | [Minhpcse04098@fpt.edu.vn](mailto:Minhpcse04098@fpt.edu.vn) | Leader |
| 2 | Nguyễn Quý Đôn | SE04468 | 0358008828 | [Donnqse04468@fpt.edu.vn](mailto:Donnqse04468@fpt.edu.vn) | Member |
| 3 | Nguyễn Quý Tuấn | SE04330 | 0972654951 | [Tuannqse04330@fpt.edu.vn](mailto:Tuannqse04330@fpt.edu.vn) | Member |

Table 1-2: Team member's information

* 1. ***Background***

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT is powering transformation for enterprises, consumers and government. Emerging tools and technologies like smart speakers, machine learning, and 5G are enabling huge gains to efficiency and more control at home and in the workplace.

The continued growth of the IoT industry is going to be a transformative force across all organizations. By integrating all of our modern day devices with internet connectivity, the IoT market is on pace to grow to over $3 trillion annually by 2026. This rise in popularity of IoT-connected devices leading to rise in IoT app development dose come with its fair share of concern and security challenges.

IoT is in no way immune to hacking. Hackers can launch DDoS attacks by infiltrating and leveraging thousands or millions of unsecured devices. They can cripple infrastructure, down networks, and as IoT advances into our everyday lives, those attacks may very well put real human lives in jeopardy. And even if hackers don’t outright threaten lives, they can compromise gateways and deeper levels of IoT networks in order to reveal and exploit sensitive personal and corporate information.

In fact, one of the main problems with tech companies building these devices is that they are too careless when it comes to handling of device-related security risks.

Most of these devices and IoT products don’t get enough updates while, some don’t get updates at all.

Spending on IoT Endpoint Security solutions will increase to $631M in 2021. Worldwide IoT security spending will increase from $912 million in 2016, soaring to $3.1 billion in 2021, attaining a 27.87% CAGR in the forecast period.

All of the above, we will have tried to create a system to checking security IoT system with temporary features such as check sever open port, running servies in server, check MQTT security. Botnet, Shell attack, Firmware bug will be opening soon.

* 1. ***Literature Review of Existing System***
  2. ***The Proposed System***

After reviewing all properties of the current systems, we have come to a decision to choose which features and functions we will provide in our system.

* + 1. ***System functions***
* Allow user to login by account have created at website
* Allow user to check open port, services running in server when user enter ip/domain
* Allow user to check security
* Give specific solutions for each case
* Allow user to export result as exel, xml or file text
* Allow user to contact with support team
* Allow user to manage their account at account management board
  + 1. ***Out of scope functions***

Because of the time limitation and the effort limitation, we will not implement these following functions, however we believe they could be implement in the feature:

* Fast scan : Check if user internet-connected devices at home are public without ip/domain
* Live support : Instead of contact with support team with email, etc . We need live chat to support user 24/7.
* Connect user with security specialist to given clear and specific solutions, overcome if user is needed.
  + 1. ***Technical Approach***

Architecture: Three-Tier Client/Server

For back-end system:

Model – View – Controller Model 2

Project management:

* Using the GitHub with GIT to stores code version control.
* Using Trello to manage the task and bug.
* Using some social networking like Messenger, Skype to communicate online

1. **PROJECT PLAN**
   1. ***Introduction***

This document provides an overview of the project plan, including software model, project organization and project schedule plan. Moreover, the document also has communication management, configuration management and risk management.

All team member must use this chapter as a guideline for tracking assigned task and deadlines.

* 1. ***Project Organization***
     1. ***Project Description***

|  |  |
| --- | --- |
| Project Name : | Internet-connected Devices Checking System |
| Project Code : | IDCS |
| Project Type : | Web Application Platform |
| Project Category : | Checking Security System |
| Business Type : | Online Customer Services |
| Project Instructor : | Phan Duy Hùng |
| Project Manager : | Phạm Công Minh |
| Time line : | 10/9/2019 – 24/12/2019 |

In this project, we will develop a Web application will named Internet-connected Devices Checking System, which help a customer to check open port, services running in to server and vulnerability if have. We also give solution based on vulnerabilities you have.

* + 1. ***Scope***

The scope of this project includes these stages:

* Develop requirement team made and software requirement specification.
* Develop architecture and detailed design document.
* Coding and unit testing.
* Deployed the application in server.
* Develop test case and execute combination test

The application has these main function

* Check port, services, security check
* Export and Support user
  + 1. ***Standard Objectives***
* Project must be finished before 31 December 2019
* All team member give best effort to complete this project
* The final application covers more than 100% of requirement
  + 1. ***Milestone and deliverables***

|  |  |  |
| --- | --- | --- |
| *No* | *Milestone* | *Delivery Date* |
| 1 | Project Registration | 09/08/2019 |
| 2 | Submit Report No.1 | 23/09/2019 |
| 3 | Submit Report No.2 | 07/10/2019 |
| 4 | Submit Report No.3 | 24/10/2019 |
| 5 | Submit Report No.4 | 29/10/2019 |
| 6 | Submit Report No.5 | 13/12/2019 |
| 7 | Submit Report No.6 | 17/12/2019 |
| 8 | Submit Final Report | 18/12/2019 |
| 9 | Submit Project Resource | 25/12/2019 |
| 10 | Thesis Defense | 03/01/2019 |

* 1. ***Project Organization***
     1. ***Purpose***

This chapter provides an overview of the project plan, including software process model, project organization and project management plan. All team members must use this chapter as a guild line for tracking assigned task and deadlines. This chapter also included an overview of this project and team members.

* + 1. ***Software Process Model***

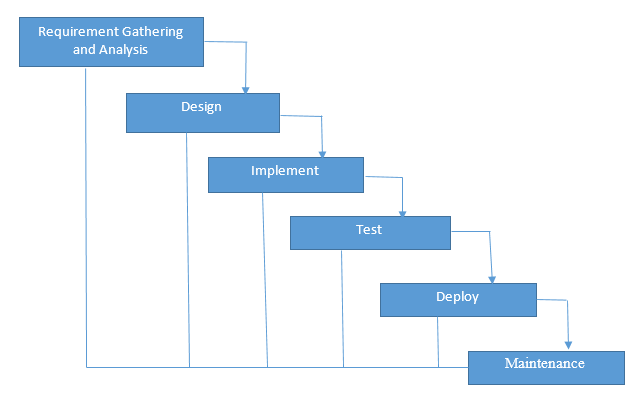


Figure 1

Figure 1-1 Waterfall Software Process Model (image: Internet)

Our Project uses The Waterfall Software Process Model.

The Waterfall Model is a linear or sequential approach to project management and works based on fixed dates, requirements, and outcomes. Teams do not require consistent communication and, unless specific integrations are required, can be self-contained. Team member can also work independently and are often required to provide status reports somewhat less frequently.

The Project scope stay relatively static, meaning cost and timelines can be determined early on in this project. A structure approach to a project means that everyone understands what needs to be done and when.

* + 1. ***Roles and Responsibilities***

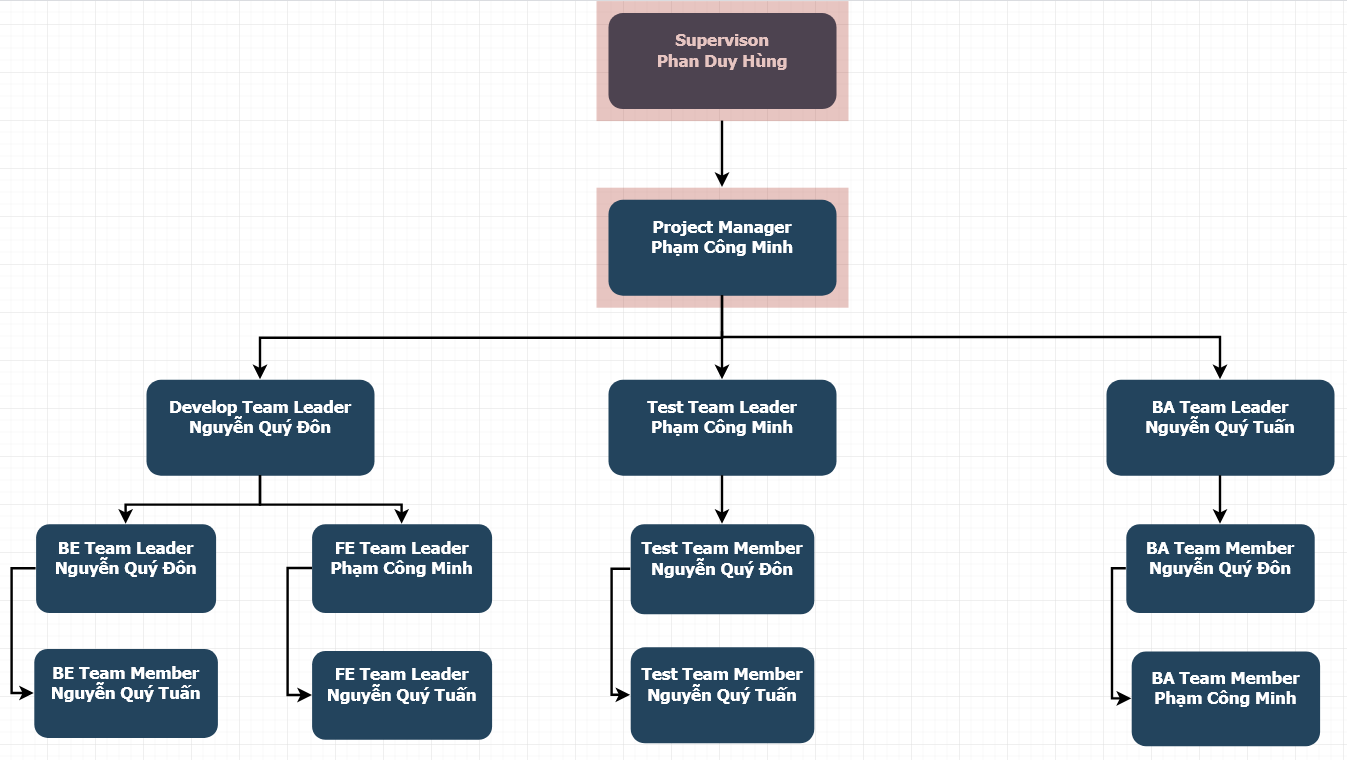


Figure 2

* + 1. ***Organizational Structure***

|  |  |
| --- | --- |
| *Role* | *Responsibility* |
| Project Manager | Responsible for leading the team and managing the whole project, planning, defining scope, developing schedules, coordinating communication, analyzing and managing risks. |
| Technical Leader | Responsible for choosing and deciding what technologies should be used, as well as for overseeing the work being done by other developers. |
| Business Analyst | Responsible for analyzing business, processes and systems. |
| Tester | Responsible for conducting tests. |
| Back-end Developer | Responsible for server-side web application logic and integration of the work front-end developers do. |
| Front-end Developer | Responsible for implementing visual elements that users see and interact with in a web application. |

Table 1

* + 1. ***Project Team Member***

|  |  |
| --- | --- |
| *Team Member* | *Roles* |
| Phạm Công Minh | Project Manager, Front-end Team Leader, Tester, Business Analyst, Designer |
| Nguyễn Quý Đôn | Back-end Team Leader, Technical Leader, Tester, Business Analyst, Designer |
| Nguyễn Quý Tuấn | Front-end Back-end Developer, Tester, Business Analyst, Designer |

Table 2

* 1. ***Project Management Plan***
     1. ***Project Schedule***

To deploy this project effectively, according to this schedule, we will follow the each tasks in project

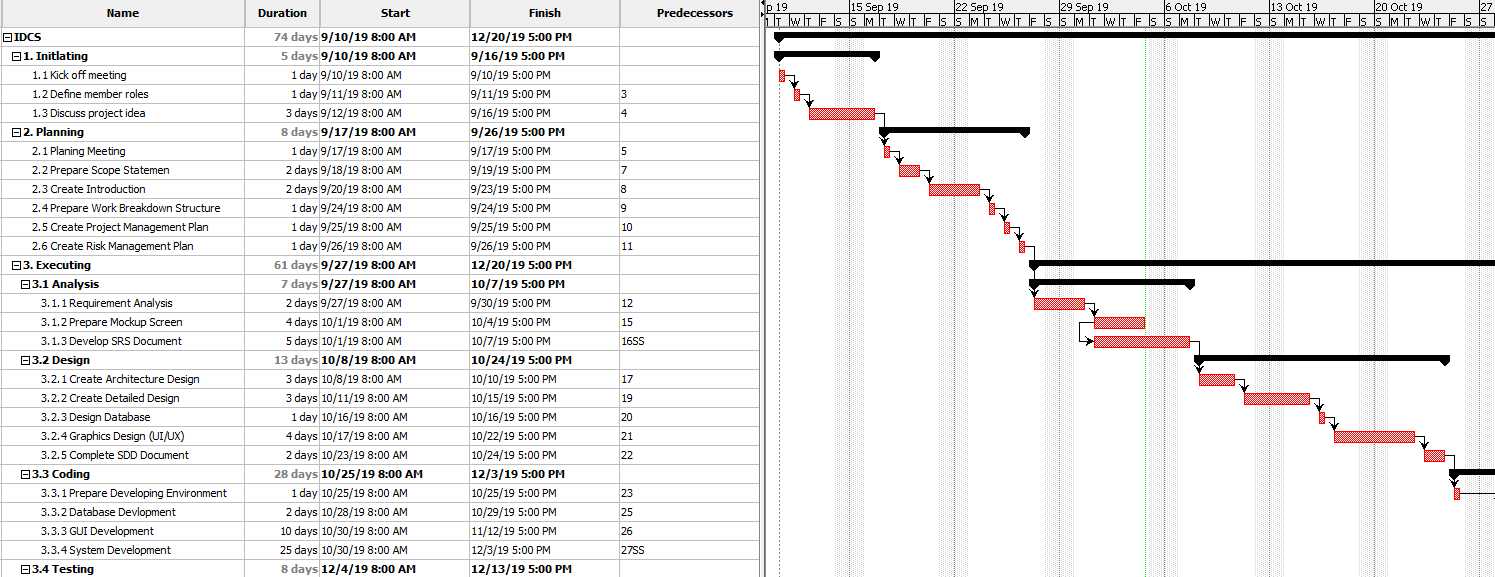


Figure 3

* + 1. ***Communication Management***
       1. ***Communication between members***

**Weekly meeting schedule:**

Team will have a meeting every Saturday to report the process of whole team’s current task and assign new task. If have any issue, we will discuss and find solution together. If it is too difficult and can’t be solved by ourselves, we will ask our supervisor for advises.

**Unscheduled meeting:**

If someone has an important problem want to be solved immediately, we will have a meeting with social network like Skype, FaceTime because we are apart.

**Communication channel:**

Our main communication channels are Facebook Messenger, Email, Skype, Trello, face-to-face meeting. However, sometimes can make a phone call or instant message if someone has problem.

* + - 1. ***Communication with supervisor***

**Face-to-Face meeting:**

Weekly on every Thursday afternoon to make sure that supervisor can keep tracking of the team’s progress.

**E-mail:**

Gmail is the faster way to get advice and document checking form supervisor.

**Mobile Phone:**

Is used to get time and place arranged for the meeting every weeks

* + - 1. ***Meeting Plan***
         1. ***Meeting minutes***

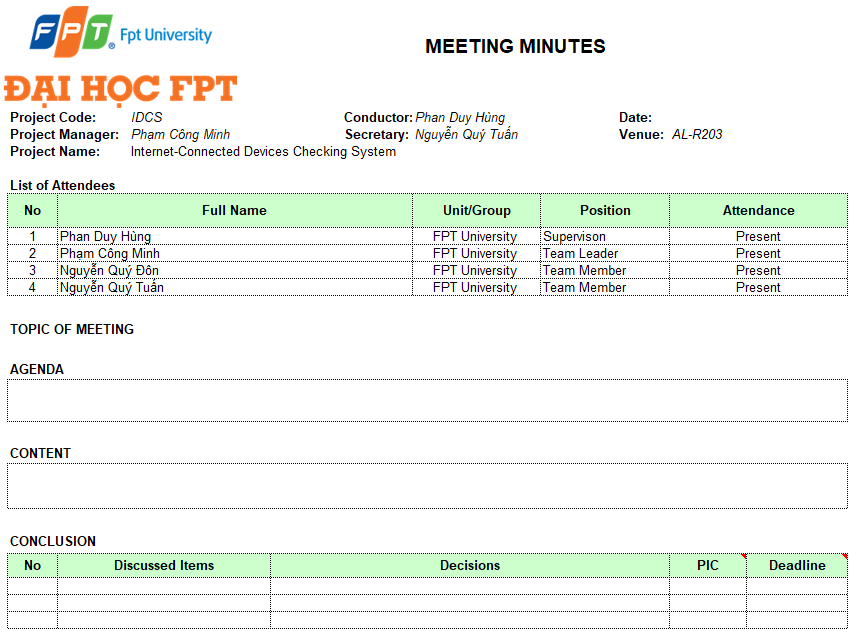


Table 3

* + - * 1. ***Progress Report***

Project Progress reports are delivered Supervisor every week. Below is the sample of our progress report:

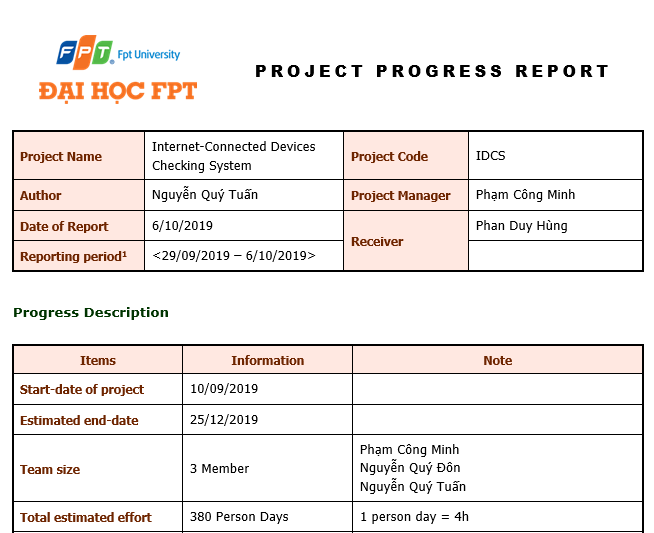


Figure 4

* + 1. ***Coding Conventions***
       1. ***Front end***

Folder/files naming:

* Written in “camelCase” and “PascalCase”, for example: “Components”, “Test”, and “login.js”. With some sub-class or sub-folder, we will use” underscore”, for example: “login\_s”, “home\_result\_s”.

Javascript Style Guide:

* We strictly follow javascript Standard Style Guide, with linter and automatic code fixer. Please refer to the official website at <https://standardjs.com/rules-en.html>
  + - 1. ***Back end***

Folder/files naming:

* Written in “camelCase” and “PascalCase”, for example: “Controller”, “Model”, and “LoginController.java”.

Java Style Guide:

* We strictly follow Java Coding Convention with linter and automatic code fixer. Please refer to coding convention form oracle at : <https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

Python Style Guide:

* We strictly follow Python Coding Convention with linter and automatic code fixer. Please refer to official coding convention website at : <https://www.python.org/dev/peps/pep-0008/>
  + 1. ***Tools and Techniques***
       1. ***Tools***

|  |  |
| --- | --- |
| **Project Management tools** | Trello, Microsoft Project 2016, ProjectLibre |
| **Document tools** | Microsoft Office 2016 (Word,Excel,PowerPoint) |
| **UML tools** | Draw.io, lucidchart.com, |
| **Ides** | Notepad ++, NetBean 8.2, IDLE |
| **DBMS** | Microsoft SQL Server 2016, Google FireBase |
| **Source code version control** | GitHub, SourceTree, Git SCM 2.19.1 |
| **Testing tool** | Postman 6.4.4, TestNG |
| **Communication tools** | Facebook Group, Messenger, Email, Trello |
| **File Management tools** | GitHub, Google Driver, DropBox |

* + - 1. ***Techniques***
         1. ***Front end***
* Programming languages: JavaScript, HTML, CSS
* Framework : Bootstrap
  + - * 1. ***Back end***
* Programming languages : Java, Python

Server:

* Tomcat 9.0.27

Library:

* Python-nmap : library to help using nmap port scanner
  + 1. ***Risk and Issue Management Plan***
       1. ***Issue Management***

The table below is some issues has been met in making project. Besides the issues, we have solution with each issue

|  |  |  |
| --- | --- | --- |
| No | Name issue | Solution |
| 1 | Conflict between member | The Project Manager has to find out the root that caused problem and solve it. Face-to-face meeting is required method. Voting is also good method |
| 2 | A team member’s absence. | Motivate member and create positive work environment. Other member replace that team member’s work |

* + - 1. ***Risk Management***
         1. ***Risk Categories***

The table below is some Risk Categories may be had in this Project.

|  |  |  |
| --- | --- | --- |
| Category | Sub-Category | Acronym & Abbreviation |
| Technical | Requirement Definition | T-RD |
| Technology | T-T |
| Complexity and Interfaces | T-CI |
| Performance and Reliability | T-PR |
| Quality | T-Q |
| Management | Estimating | M-E |
| Human Resources | M-HR |
| Communication | M-C |
| Source | M-S |
| Controlling | M-C |

Table 4

* + - * 1. ***Risk Register***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Name | Risk | Category | Root Cause | Probability | Impact |
| 1 | R1 | Team don’t meet the deadline on time. | M-HR | Members lack of responsibility, lack of Management | High | High |
| 2 | R2 | Lack of knowledge about technology needs for project | T-T | Depend on requirement, new technology should be applied in the project, which some team members haven’t used before or training members before use new technology | Medium | High |
| 3 | R3 | The morale of work goes down | T-HR | Conflict among team members/ working alone ,meet a difficult problem | High | Low |
| 4 | R4 | Team member misunderstands requirement | T-RD | In the beginning, team member doesn't clear the requirement | Low | High |

* + - * 1. ***Risk Respond***

Risk in project cannot be avoid, so we must respond with risk suitably to continue the project conveniently. The table below is some risk respond which team will respond to risk.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Mitigation Plan | Contingency Plan | Fallback Plan | Status |
| 1 | R1 | Make rules and penalties for member who misses the deadline. | Working overtime to complete tasks on time. | Reduced time of another task to require | Active/Solved |
| 2 | R2 | Each team members must study to understand the technology, framework that is needed for project | Technical Leader needs to support carefully for team member in group to increase required skill and knowledge. Review complete task to make comment for team members | Discuss with supervisor about technical, may change to another easy technical or not | Active/Solved |
| 3 | R3 | Members have to read requirement specification and related document carefully. | Making sure that any miscommunication would be resolved | All team members will take a time together to fix requirement and adjust project on the right way | Active/Solved |
| 4 | R4 | Warm-up by teambuilding | Project Manager should talk to a member who gets the problem and helps that member solves the problem | Giving that member’s problems for the supervisor solve the solution | Active/Solved |

* + - * 1. ***Risk Probability and Impact***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Probability | |  |  |  |  | | --- | --- | --- | --- | | High | R3 |  | R1 | | Medium |  |  | R2 | | Low |  |  | R4 | |  | Low | Medium | High | |
|  | Impact |

The probability and impact of occurrence for each identified risk will be assessed by the project manager, with input from the project team using the following approach:

**Probability:**

* **High -** Likely or very likely to occur.
* **Medium -** May occur about half the time.
* **Low -** Very unlikely and unlikely to occur.

**Impact:**

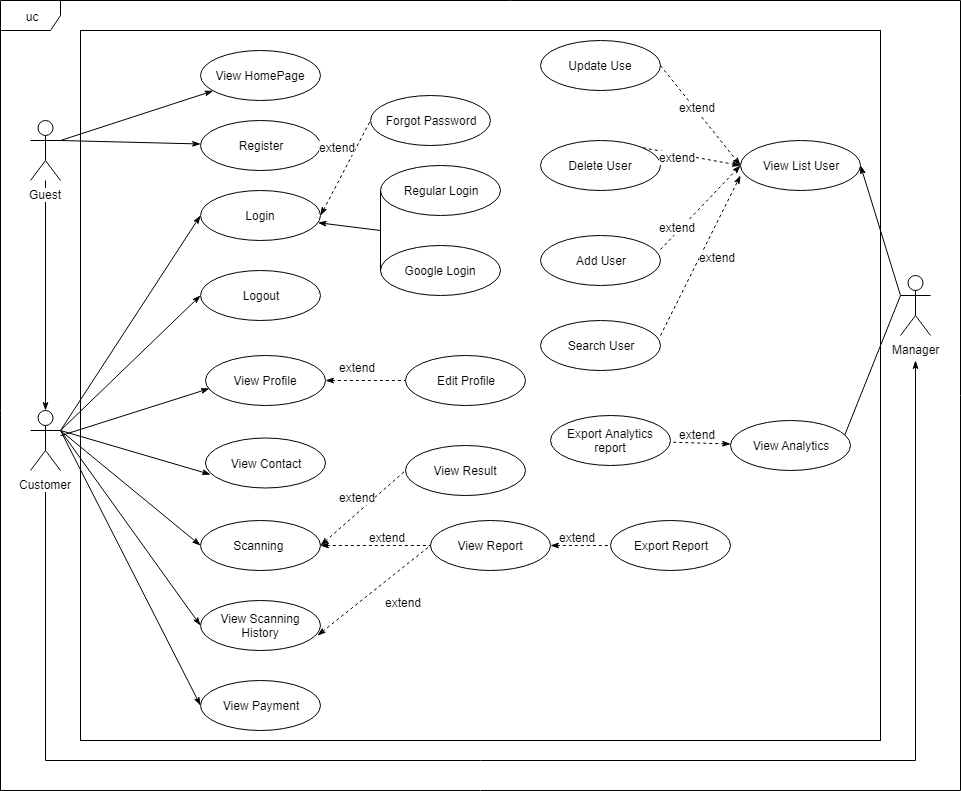
* **High -** Will jeopardize the project or bring the project to a halt
* **Medium -** Will impact the project in terms of timeline, cost, quality etc. However, the project will still move forward
* **Low -** Minimal impact to the project, not critical to project deliverables
  + - * 1. ***Closing Risk***

Risk could be closed in six case:

* When the time of a risk happening is over
* When the scope of a project is amended and risk becomes irrelevant (avoided).
* When a risk is addressed well and reduced to acceptable level.
* Risk closure at the direction of the Project Manager
* When risks become no longer relevant to the project.
* When the risk is considered successfully mitigated.
  + - 1. **SOFTWARE REQUIREMENT SPECIFICATION**
  1. ***Purpose***

This chapter provides the describing of functional requirements and non-functional requirements. These requirements assure that the system will correctly and reliably perform its intended functionality. This specification provides general, as well as specific requirements to be used in the design, testing and validation of the system.

* 1. ***Functional Requirements***
     1. ***Use Case Diagram***



* + 1. ***Business Rules***

|  |  |
| --- | --- |
| No | Description |
| B01 | Full Name cannot be empty |
| B02 | Email cannot be empty |
| B03 | Phone Number cannot be empty |
| B04 | Password cannot be empty |
| B05 | Phone Number must have from 10 to 11 digits |
| B06 | Email address must be valid form |
| B07 | Each email address may be used for only one account |
| B08 | Password must contains at least 6 character |
| B09 | Repeat password must match with password |
| B10 | New Password must different from current password |
| B11 | Scanning target not valid domain form or IP. |

* + 1. ***Use Case***

|  |  |
| --- | --- |
| Actor | Description |
| Guest | Anyone who does not have an account. |
| Manager | A person with the highest permission level and id responsible for managing the website. |
| Customer | Anyone who has an account |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Primary Actor | Secondary Actor | Function | Assignee |
| UC-01 | Guest | Customer, Manager | View Home Page | TuanNQ |
| UC-02 | Guest | N/A | Register | TuanNQ |
| UC-03 | Customer | Manager | Login | TuanNQ |
| UC-04 | Customer | Manager | Regular Login | TuanNQ |
| UC-05 | Customer | Manager | Google Login | TuanNQ |
| UC-06 | Customer | Manager | Forgot Password | TuanNQ |
| UC-07 | Customer | Manager | Logout | TuanNQ |
| UC-08 | Customer | Manager | View Profile | TuanNQ |
| UC-09 | Customer | Manager | Edit Profile | TuanNQ |
| UC-10 | Customer | Manager | View Payment and Contact | TuanNQ |
| UC-11 | Customer | Manager | Scanning | TuanNQ |
| UC-12 | Customer | Manager | View Result | TuanNQ |
| UC-13 | Customer | Manager | View Report | TuanNQ |
| UC-14 | Customer | Manager | Export Reports | TuanNQ |
| UC-15 | Customer | Manager | View Scanning History | TuanNQ |
| UC-16 | Manager | N/A | View List User | TuanNQ |
| UC-17 | Manager | N/A | Search User | TuanNQ |
| UC-18 | Manager | N/A | Update User Information | TuanNQ |
| UC-19 | Manager | N/A | Delete User | TuanNQ |
| UC-20 | Manager | N/A | Add User | TuanNQ |
| UC-21 | Manager | N/A | View Analytics | TuanNQ |

* + - 1. ***Guest***
         1. ***View Home Page***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-01 | | |
| **Use Case Name** | View Home Page | | |
| **Creator** | TuanNQ | Date Create | 22/10/2019 |
| **Version** | 1.0 | Last Update | 22/10/2019 |
| **Primary Actor** | Guest | Secondary Actor | Customer, Manager |
| **Description** | View the homepage of the website | | |
| **Pre-conditions** | User is currently not logged in | | |
| **Post-conditions** | The homepage of the website is displayed | | |
| **Normal Flow** | 1. User enter the URL of the website 2. The system, the result response is the homepage of website | | |
| **Alternative Flow** | 1. User click website’s logo 2. The system, the result response is the homepage of website | | |
| **Exceptions** | N/a | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | N/A | | |
| **Other Information** | N/A | | |

* + - * 1. ***Register***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-02 | | |
| **Use Case Name** | Register | | |
| **Creator** | TuanNQ | Date Create | 22/10/2019 |
| **Version** | 1.0 | Last Update | 22/10/2019 |
| **Primary Actor** | Guest | Secondary Actor |  |
| **Description** | Create a new account | | |
| **Pre-conditions** | * User is currently not logged in | | |
| **Post-conditions** | * The account is added to the system * User is automatically logged into the system * User is redirected to Login Page | | |
| **Normal Flow** | 1. From the homepage, user click on button “Login”. 2. The system will load the Login page. 3. From the Login page, user click on “Sign Up” link. 4. User fills information into the required form. 5. User click button “Sign Up”. 6. The system will verification email is sent to user’s email. 7. User clicks to hyperlink in email. 8. The system display a message: “Successfully, you can login to system.” | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | EXC1: At step 4 of normal flow, user leaves Full Name blank, then proceeds to step 5.  The system displays the error message “Name is required”. User is not registered.  EXC2: At step 4 of normal flow, user leaves Email blank, then proceeds to step 5.  The system displays the error message “Email is required”. User is not registered.  EXC3: At step 4 of normal flow, user leaves Phone Number blank, then proceeds to step 5.  The system displays the error message “Phone Number is required”. User is not registered.  EXC4: At step 4 of normal flow, user leaves Password blank, then proceeds to step 5.  The system displays the error message “Password is required”. User is not registered.  EXC5: At step 4 of normal flow, user leaves Repeat Password, then proceeds to step 5.  The system displays the error message “Repeat Password is required”. User is not registered.  EXC6: At step 4 of normal flow, user enter Repeat Password not match Password, then proceeds to step 5.  The system displays the error message “Confirm Password does not match”.  EXC7: At step 4 of normal flow, user enter Password that below 6 character, then proceeds to step 5  The system displays the error message “Password must contain at least 6 character”  EXC8: At step 4 of normal flow, user enter Phone Number not match valid format, then proceeds to step 5  The system displays the error message “Invalid Phone Number, Phone Number must have 10 to 11 digits”.  EXC9: At step 4 of normal flow, user enter email that already exists in the system.  EXC10: At step 4 of normal flow, user enter invalid email. | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | B01, B02, B03, B04, B05, B06, B07, B08, B09. | | |
| **Other Information** | N/A | | |

* + - 1. ***Customer***
         1. ***Login***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-03 | | |
| **Use Case Name** | Login | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | View the Login Page of the website | | |
| **Pre-conditions** | User is currently not logged in | | |
| **Post-conditions** | The Login page of the website is displayed | | |
| **Normal Flow** | 1. From the homepage, user click on button “Login”. 2. The system will load the Login page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | N/A | | |
| **Other Information** | N/A | | |

* + - * 1. ***Regular Login***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-04 | | |
| **Use Case Name** | Regular Login | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | Login an account using username and password. | | |
| **Pre-conditions** | 1. User can access the system. 2. User has already registered an account. 3. User is currently not logged into the system. | | |
| **Post-conditions** | 1. User is currently not logged in. 2. User is redirected to Homepage. | | |
| **Normal Flow** | 1. From the Login Page, User enter email and password. 2. User click “Login” button. 3. The system checks your input data. If user entered true, user will be login successful. 4. The system redirects user to the Homepage | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | EXC1: At step 1 of normal flow, user leaves email blank, then proceeds to step 2.  The system displays the error message “Email is required”. User is not logged in.  EXC2: At step 2 of normal flow, user leaves password blank, then proceeds to step 2.  The system displays the error message “Password is required”. User is not logged in.  EXC3: At step 2 of normal flow, user enters invalid login credentials, then proceeds to step 2. | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | B02, B04, B06 | | |
| **Other Information** | N/A | | |

* + - * 1. ***Google Login***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-05 | | |
| **Use Case Name** | Google Login | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | Login an account using Google Sign-In | | |
| **Pre-conditions** | 1. User can access the system. 2. User has already registered an account. 3. User is currently not logged into the system. | | |
| **Post-conditions** | 1. User is currently not logged in. 2. User is redirected to Homepage. | | |
| **Normal Flow** | 1. From the Login Page, User click “Sign in with google”. 2. The system shows Google Account pop-up. 3. User choose Google account want to login. 4. The system checks your authentication. If this true, user will be login successful. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | EXC1: At step 3 of normal flow, user choose Google account not registered in the system  The system displays the error message “Login Failed, this email not register in the system, please sign in”. User is not logged in. | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | N/A | | |
| **Other Information** | N/A | | |

* + - * 1. ***Forgot Password***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-06 | | |
| **Use Case Name** | Forgot Password | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | User creates new password | | |
| **Pre-conditions** | 1. User must be at the Login Page. | | |
| **Post-conditions** | 1. User create new password 2. User is currently not logged in | | |
| **Normal Flow** | 1. On the Login Page, user click “Forgot Password?” hyperlink. 2. The system proceeds, user will be redirected to Forgot Password Page. 3. Enter your name, your registered email and Click “Submit” button. 4. The system display a message “We will send new password to your email. Please check your email in 24 hours”. 5. In your email, there will be a confirmation link that you forgot password and want to change your password. 6. The system, user will be redirected to the “Forgot Password Page”. 7. User enters new password and re-enter new password in the corresponding text box. 8. User click “Submit” button. 9. The system updates user’s new password and display home page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | EXC1: At step 3 of normal flow, user leaves Name blank, then proceeds to step 4  The system displays the error message “Please fill out this field” on bottom of your Name text box.  EXC2: At step 3 of normal flow, user leaves Email blank, then proceeds to step 4  The system displays the error message “Please fill out this field” on bottom of your Email text box.  EXC3: At step 3 of normal flow, user entered invalid email form, then proceeds to step 4  The system displays the error message “Invalid Email, please enter valid email form”.  EXC4: At step 3 of normal flow, user entered email not exist in the system.  The system displays the error message “Can’t find this email, please enter another one.”  EXC5: At step 7 of normal flow, user leaves Password blank, then proceeds to step 8.  The system displays the error message “Password is required”. User is not registered.  EXC6: At step 7 of normal flow, user leaves Repeat Password, then proceeds to step 8.  The system displays the error message “Repeat Password is required”. User is not registered.  EXC7: At step 7 of normal flow, user enter Repeat Password not match Password, then proceeds to step 8.  The system displays the error message “Confirm Password does not match”.  EXC8: At step 7 of normal flow, user enter Password that below 6 character, then proceeds to step 8.  The system displays the error message “Password must contain at least 6 character” | | |
| **Priority** | High | | |
| **Frequency of Use** | Low | | |
| **Business Rules** | B01, B02, B04, B06, B08, B09 | | |
| **Other Information** | N/A | | |

* + - * 1. ***Logout***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-07 | | |
| **Use Case Name** | Logout | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | Log out of the account | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. | | |
| **Post-conditions** | 1. User is logged out of the system. 2. The system show Login Page Screen. | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Logout”. 3. User will redirected to the Login page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/A | | |
| **Other Information** | N/A | | |

* + - * 1. ***View Profile***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-08 | | |
| **Use Case Name** | View Profile | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | View personal profile | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. | | |
| **Post-conditions** | 1. System displays the User Infomation page. | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “User Information”. 3. User will redirected to the User Information page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/A | | |
| **Other Information** | N/A | | |

* + - * 1. ***Edit Profile***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-9 | | |
| **Use Case Name** | Edit Profile | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | View Payment and Contact Information | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. | | |
| **Post-conditions** | 1. System update user profile information. | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “User Information”. 3. User will redirected to the User Information page. 4. User input Full Name if want to update. 5. User input Email if want to update. 6. User input Phone Number if want to update. 7. User input Password. 8. User input New Password if want to change password. 9. User click “Update” button. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | EXC1: At step 4, Full Name cannot blank.  EXC2: At step 5, Email cannot blank and cannot curren exist in system.  EXC3: At step 6, Phone Number cannot blank.  EXC4: At step 7, Password must match current Password.  EXC5: At step 8, New Password must difference current password | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | B01, B02, B03, B04, B05, B06, B07, B10 | | |
| **Other Information** | N/A | | |

* + - * 1. ***View Payment and Contact***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-10 | | |
| **Use Case Name** | View Payment and Contact | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | View Payment and Contact Information | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. | | |
| **Post-conditions** | 1. System displays payment and contact information. | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Payment”. 3. User will redirected to the Payment Information page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/A | | |
| **Other Information** | N/A | | |

* + - * 1. ***Scanning***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-11 | | |
| **Use Case Name** | Scanning | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | Scanning domain or IP target | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. 3. Users have remaining uses. | | |
| **Post-conditions** | 1. System scan target. | | |
| **Normal Flow** | 1. From the Home page, user enter domain or IP need to scan. 2. Click “Scan” button. 3. The system scan target. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | EXC1: At step 1 of normal flow, user enter invalid domain form or IP.  The system displays the error message “Domain or IP must valid form”. | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | B11 | | |
| **Other Information** | N/A | | |

* + - * 1. ***View Result***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-12 | | |
| **Use Case Name** | View Result | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | View result of scanning target | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. 3. Users have click “Scan” button. | | |
| **Post-conditions** | 1. System show result of target. | | |
| **Normal Flow** | 1. System show popup “Scan successfully, do you want see result?”. 2. User click “Show Result” button. 3. User will directed to Result page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - * 1. ***View Report***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-13 | | |
| **Use Case Name** | View Report | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | View report of scanning target | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. | | |
| **Post-conditions** | 1. System show report of target. | | |
| **Normal Flow** | 1. On Show Result page, user click “View report” button. 2. User will directed to View Report page. | | |
| **Alternative Flow** | 1. On Scanning History, user click “View Report” button on the right of history you want. 2. User will directed to View Report page. | | |
| **Exceptions** | N/a | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - * 1. ***Export Report***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-14 | | |
| **Use Case Name** | Export Report | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | Export report to customer. | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. 3. Users in View Report page. | | |
| **Post-conditions** | 1. Report downloaded Report. | | |
| **Normal Flow** | 1. On View Report page, user click “Export Report” button. 2. User choose type of report. 3. Click “Export Button”. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | High | | |
| **Frequency of Use** | High | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - * 1. ***View Scanning History***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-15 | | |
| **Use Case Name** | View Scanning History | | |
| **Creator** | TuanNQ | Date Create | 23/10/2019 |
| **Version** | 1.0 | Last Update | 23/10/2019 |
| **Primary Actor** | Customer | Secondary Actor | Manager |
| **Description** | View scanning history of customer | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in. | | |
| **Post-conditions** | 1. User view scanning history | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “History” button. 3. User will directed to History page. | | |
| **Alternative Flow** | 1. Form center of Website, user click “Last Check” hyperlink 2. User will directed to History page with date sort | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - 1. ***Manager***
         1. ***View List User***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-16 | | |
| **Use Case Name** | View List User | | |
| **Creator** | TuanNQ | Date Create | 24/10/2019 |
| **Version** | 1.0 | Last Update | 24/10/2019 |
| **Primary Actor** | Manager | Secondary Actor | N/A |
| **Description** | View list user in system | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in with admin account sufficient privileges. | | |
| **Post-conditions** | 1. View list user in system. | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Manage” button. 3. User will directed to List User page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - * 1. ***Search User***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-17 | | |
| **Use Case Name** | Search User | | |
| **Creator** | TuanNQ | Date Create | 24/10/2019 |
| **Version** | 1.0 | Last Update | 24/10/2019 |
| **Primary Actor** | Manager | Secondary Actor | N/A |
| **Description** | Search user in system | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in with admin account sufficient privileges. | | |
| **Post-conditions** | 1. The search result for words is displayed the course. | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Manage” button. 3. User will directed to List User page. 4. In the Search text box on top, enter the text you want to search. 5. User click “Search” button. 6. The system displays User’s name which user need to search. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - * 1. ***Update User Information***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-18 | | |
| **Use Case Name** | Update User Information | | |
| **Creator** | TuanNQ | Date Create | 24/10/2019 |
| **Version** | 1.0 | Last Update | 24/10/2019 |
| **Primary Actor** | Manager | Secondary Actor | N/A |
| **Description** | Update user is selected | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in with admin account sufficient privileges. 3. User is existed | | |
| **Post-conditions** | 1. This selected user is updated | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Manage” button. 3. User will directed to List User page. 4. User click to a user and click icon update. 5. System redirect to update user page. 6. User input information which user want to update 7. User input reason. 8. User click “Update” button | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | At step 6 cannot input blank information. | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - * 1. ***Delete User***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-19 | | |
| **Use Case Name** | Delete User | | |
| **Creator** | TuanNQ | Date Create | 24/10/2019 |
| **Version** | 1.0 | Last Update | 24/10/2019 |
| **Primary Actor** | Manager | Secondary Actor | N/A |
| **Description** | Delete user is selected | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in with admin account sufficient privileges. 3. User is existed | | |
| **Post-conditions** | 1. Store is selected is delete | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Manage” button. 3. User will directed to List User page. 4. User click to a user and click icon delete. 5. System display dialog confirm delete. 6. User click “OK” to delete user. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* + - * 1. ***Add User***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-20 | | |
| **Use Case Name** | Add new User | | |
| **Creator** | TuanNQ | Date Create | 24/10/2019 |
| **Version** | 1.0 | Last Update | 24/10/2019 |
| **Primary Actor** | Manager | Secondary Actor | N/A |
| **Description** | Create success new User in system. | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in with admin account sufficient privileges. | | |
| **Post-conditions** | 1. This selected user is updated | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Manage” button. 3. User will directed to List User page. 4. User click to a user and click icon create. 5. System redirect to create user page. 6. User input Name. 7. User input Email. 8. User input Phone Number. 9. User click button “Create”. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | ECX1: At step 6 cannot blank.  ECX2: At step 7 cannot blank and cannot duplicate with email created before. | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | B01, B02, B06, B07 | | |
| **Other Information** | N/a | | |

* + - * 1. ***View Website Analytics***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-21 | | |
| **Use Case Name** | View Website Analytics | | |
| **Creator** | TuanNQ | Date Create | 24/10/2019 |
| **Version** | 1.0 | Last Update | 24/10/2019 |
| **Primary Actor** | Manager | Secondary Actor | N/A |
| **Description** | View analytics of website include total view, total scan. | | |
| **Pre-conditions** | 1. User accesses the system. 2. User is currently logged in with admin account sufficient privileges. | | |
| **Post-conditions** | 1. This selected user is updated | | |
| **Normal Flow** | 1. From the Website header, user move cursor to top right corner, with the user’s personal name. 2. User click “Analytics” button. 3. User will directed to Website Analytics page. | | |
| **Alternative Flow** | N/a | | |
| **Exceptions** | N/a | | |
| **Priority** | Medium | | |
| **Frequency of Use** | Medium | | |
| **Business Rules** | N/a | | |
| **Other Information** | N/a | | |

* 1. ***Non-functional Requirement***
     1. ***Security***

As a Website about Security, connect to information of user, security is indispensable and obligatory in this system:

* Apply account & password to restrict user access.
* Check role of user when login and use function of system.
  + 1. ***Availability***

As a Website about Security, information is necessary and important

* Information always constantly updated.
  + 1. ***Usability***

Below are usability requirement of website IDCS:

* The user interface is simple, clear and easy to use. Users don’t need the guidance of others.
* IDCS system is designed simple, clear, and easy to use.
* Support minimizes the number of user actions
  + 1. ***Performance***

Below are the performance requirements of website IDCS:

* Under good internet connection, the majority of pages should load within 5-10 seconds.
* The system doesn’t need to operate real time but it must process as quickly as possible.
  + 1. ***Maintainability***

The system must be built in such a way this it is maintainable. Below are maintainability requirements of website IDCS:

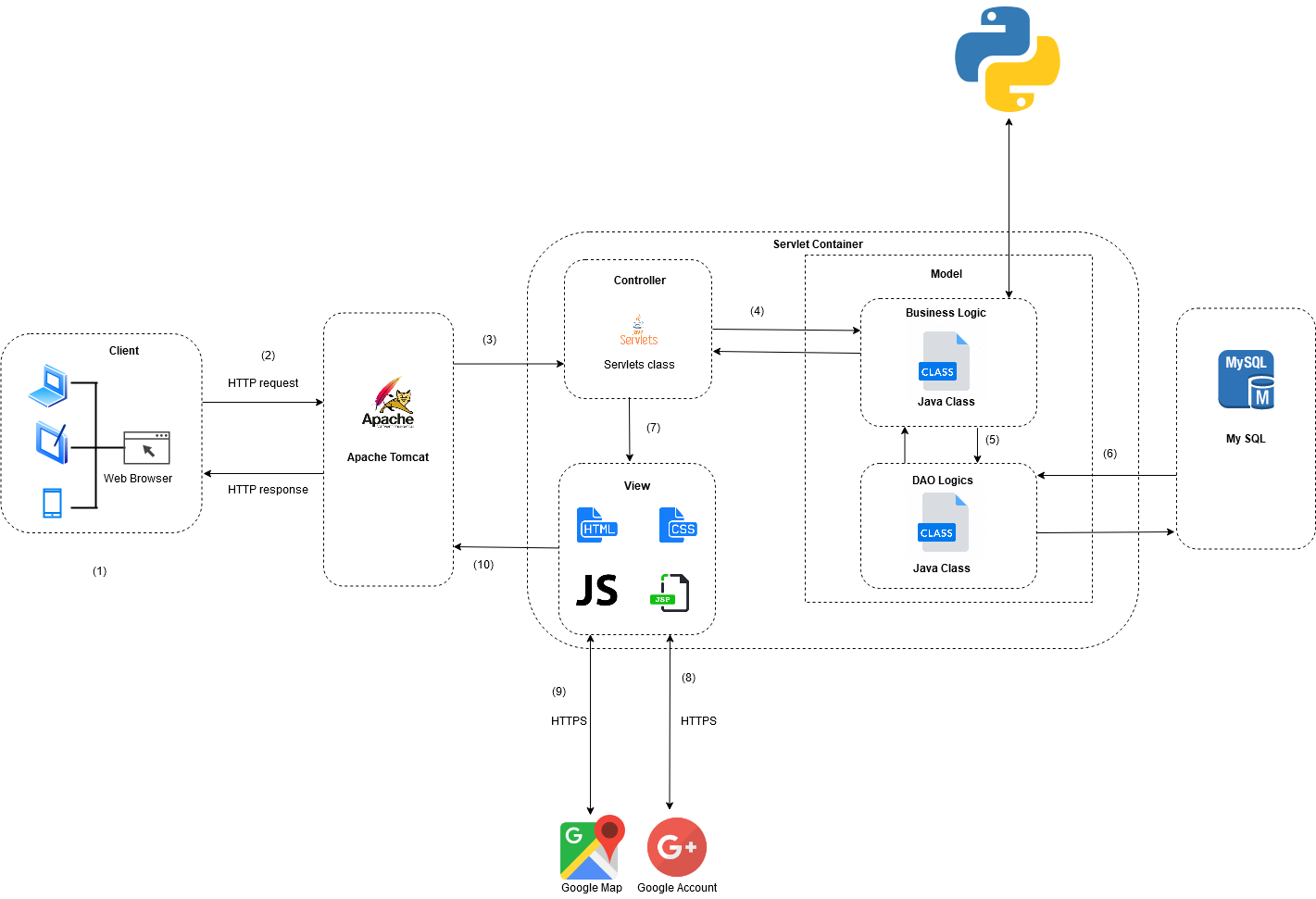
* All code must have comment clearly.
* Strictly follow coding conventions.
* Easy to fix bug and update
  + 1. ***Entity Relationship Diagram***

Below is the Entity Relationship Diagram of the system :

1. **SOFTWARE DESIGN DESCRIPTION**
   1. ***Purpose***

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

* Architecture overview
* Component diagram
* Detailed design
* Detailed description of components
* Database design
  1. ***Architecture Overview***
     1. ***System Architecture***
        1. ***Diagram***



* + 1. ***System Architecture Explanation***
       1. ***Apache Tomcat***



**The Apache Tomcat**® software is an open source implementation of the Java Servlet, Java Server Pages, Java Expression Language and Java Web Socket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.

* + - 1. ***MySQL***

******

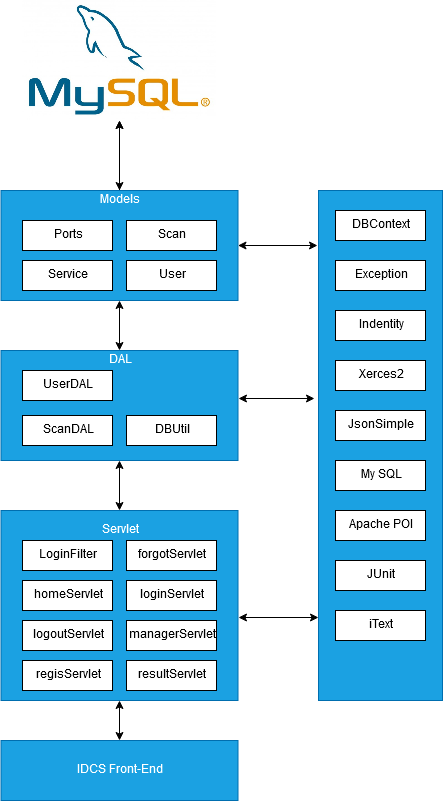
**MySQL** is the world's most popular open source database. With its proven performance, reliability and ease-of-use, **we use it to stores database.**

* + - 1. ***Python***



**Python** is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. With Python, **we use it to scanning server and client target**

* 1. ***System Design***
     1. ***Architecture Layers Design***
        1. ***IDCS – Backend Layer Design***



* + - 1. ***Explanation choice of architecture design***

For the back-end of IDCS, we have decided to use a 3-tier architecture style.

1. **Presentation tier**: The presentation tier is the front end layer in the 3-tier system and consists of the user interface. The main function of the interface is to translate tasks and result to something the user can understand.
2. **Logic tier:** This tier contains the functional business logic. Its consists of all the data of the application. Its written in Java and Python. It’s also moves and processes data between Data and Presentation layers.
3. **Data tier:** This tier comprises of the database/data storage system and data access layer. We use My SQL to store data. The information is then passed back to the logic tier for processing, and then back to the user.

This architecture makes system Scalability, Centralized Security and Fault Tolerance.

* + - 1. ***Model-View-Controller Architecture***
         1. ***What is Model-View-Controller***

**Model-View-Controller** (usually knows as MVC) is a software design pattern that separates the business logic, presentation logic and data.

1. **Controller** acts as an interface between View and Model. Controller intercepts all the incoming request
2. **Model** is layer contain object information (Data), interacting with Database. Mainly responsible for object modeling.
3. **View** is interface of the system to interact directly with users.

MVC is a systematic way to use the application where the flow starts from the view layer, where the request is raised and processed in controller layer and sent to model layer to insert data and get back the success or failure message.

**Pros:**

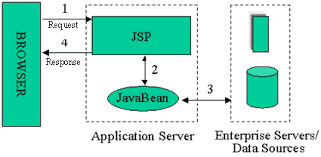
* Demonstrate professionalism in programming and object analysis, because it is divided into separate components, so independent activities help develop application faster, simpler and easier to upgrade and maintain.

**Cons:**

* As a very methodical working model, for small applications, using MVC is very time-consuming and causes a lot of complexity.
  + - * 1. ***Two Types of MVC***

MVC is divided into two type: MVC1 and MVC2:

**MVC Model 1:**

****

* In MVC model 1, Servlet technology does not create process, instead it creates a Thread to process the request. The advantage of creating a thread is that it does not allocate memory areas separately. Therefore, multiple consecutive request can easily be handle by the Servlet. The main problem with Servlet is that it needs to be recompiled if any code is modified.
* JSP handles most of the Servlet problems. You do need to re-deploy the application if the JSP page is modified. JSP provides support for developing web application using JavaBean, Custom Tag and JSLT custom tag so you can set specific logic with JSP, so testing and debugging become easier.

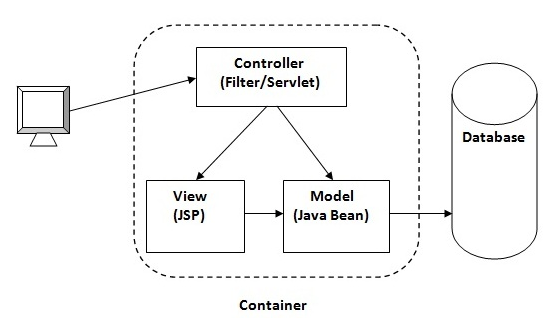
**Pros:**

* Easy and fast to develop web application

**Cons:**

* The navigation controls are not centralized as each page has its own logic to determine the next page. If the JSP page name is changed and is referenced by another page, then we must also change that name in the reference page.
* Time consuming. You need to spend more time developing Custom Tag custom tags in JSP.

**MVC model 2**

****

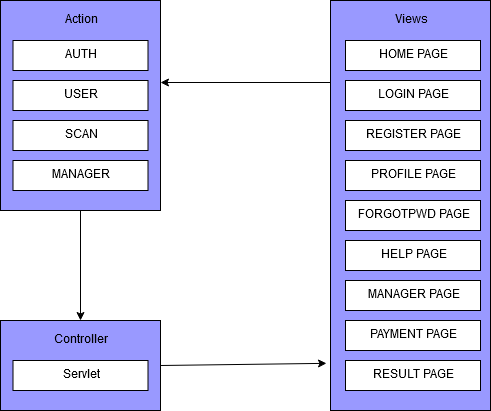
* **Model:** State representation (data), and is the lowest layer in the sample. It responds to requests from the view and it also responds to directives from the controller to update itself.
* **View:** Responsible for displaying data in a specific format.
* **Controller:** Act as an Interface between View and Model. It intercepts and is responsible for responding to all requests (receiving inputs and commands) to the Model / View to perform the corresponding interactions. The controller receives input, it performs validation and then performs operations that modify the state of the Model.

With IDCS, we have decided to use MVC model 2.

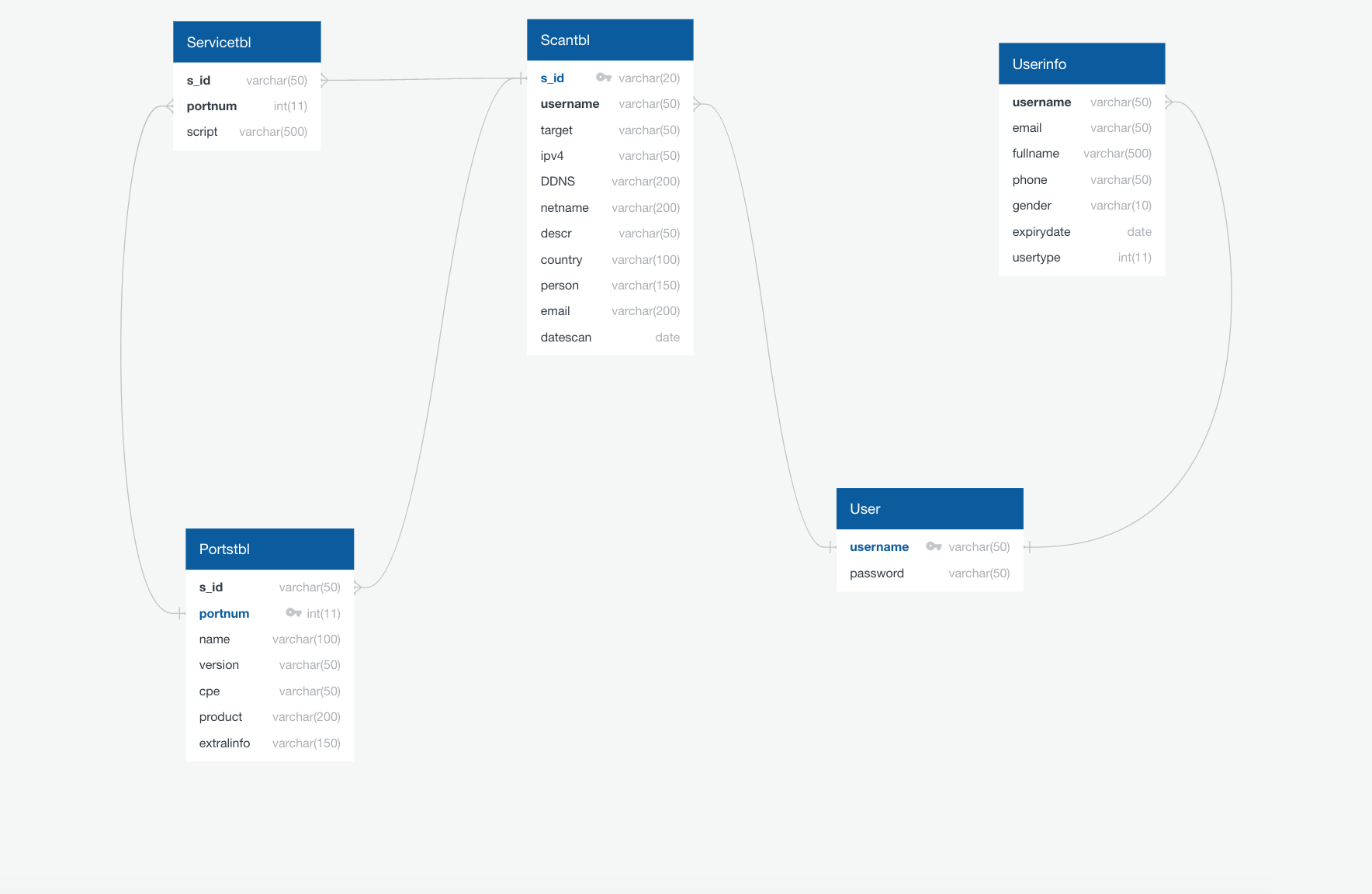
* + - * 1. ***Why MVC model 2?***
* **Reliability:** dividing each part separately helps us to modify each component separately, no effect, can be replaced into equivalent parts, can divide jobs in groups, compile independently, enhance integration with high correctness.
* **Compatibility:** able to use different types of technologies that are not dependent on us because we have been separated and conceptual for each type of component separately.
* **Reusable**: we can use components to re-divide in other applications or reuse multiple times in the same application, increasing efficiency in programming.
* Quick deployment and quick maintenance (quick deploy and easy maintenance): because components are independent of each other.

We using MVC pattern in 3-tier architecture style:

* **Presentation Tier:** “Controller and View” form MVC Pattern.
* **Business Tier: “**Model**”** from MVC Pattern.
* **Data Access Tier:** Original Data Access Tier.
  + 1. ***IDCS – Frontend***



* + 1. ***Class Diagram***
    2. ***Database Diagram***



* + - 1. ***Data Dictionary***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Scantbl | | | | | | |
| Table Description | Store Scan target information | | | | | |
| Attribute | Attribute Description | Datatype | Allow NULL | Unique | Index | Key |
| s\_id | id of scan | varchar(20) |  |  |  | Primary key |
| username | username of customer | varchar(50) |  |  |  | Foreign key: User(username) |
| target | domain target | varchar(50) |  |  |  |  |
| ipv4 | ipv4 of target | varchar(50) |  |  |  |  |
| DDNS | DDNS of target | varchar(200) |  |  |  |  |
| netname | netname of target | varchar(200) |  |  |  |  |
| descr | description of target | varchar(50) |  |  |  |  |
| country | country host of target | varchar(100) |  |  |  |  |
| person |  | varchar(150) |  |  |  |  |
| email |  | varchar(200) |  |  |  |  |
| datescan | Date time scan | varchar(date) |  |  |  |  |

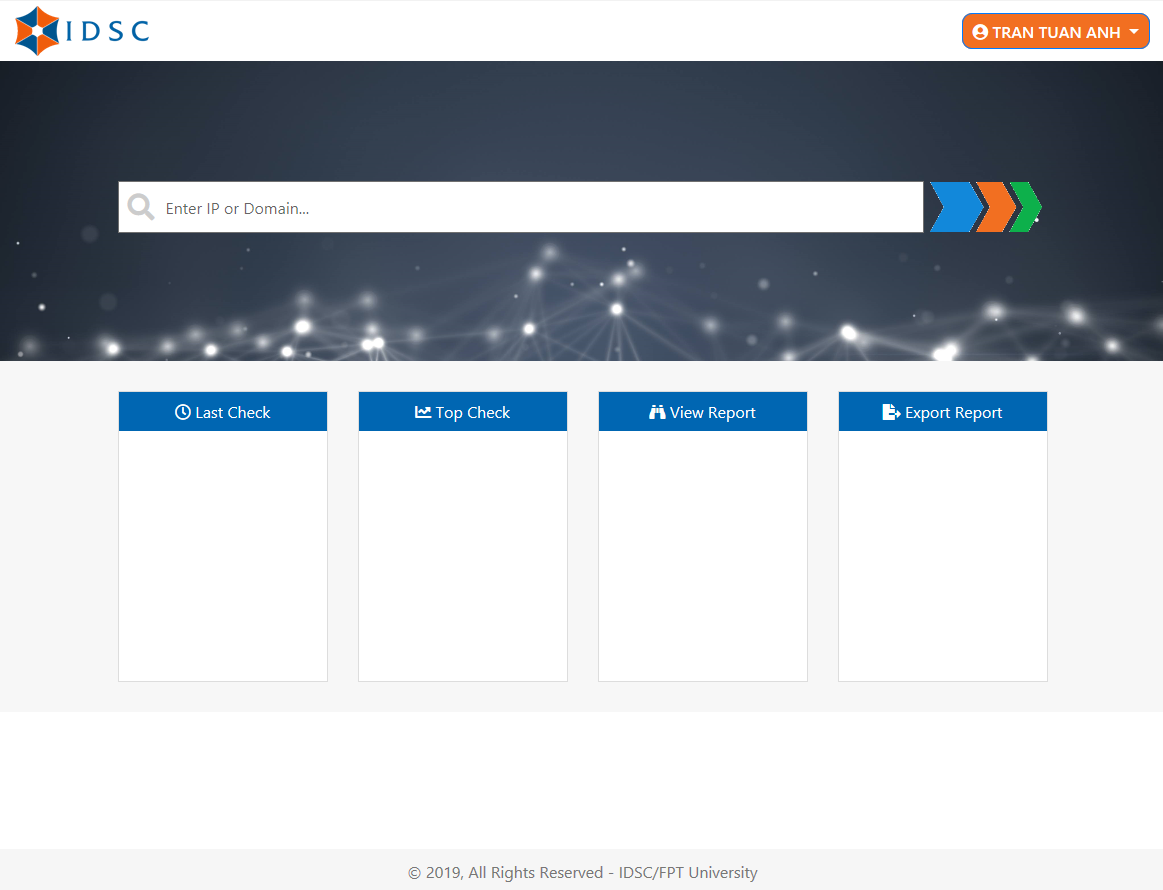
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Servicetbl | | | | | | |
| Table Description | Store Services information | | | | | |
| Attribute | Attribute Description | Datatype | Allow NULL | Unique | Index | Key |
| s\_id | id of scan | varchar(50) |  |  |  | Foreign key: Scantbl(s\_id) |
| portnum | port number of target | int(11) |  |  |  | Foreign key: Porttbl(portnum) |
| script |  | varchar(500) |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Portstbl | | | | | | |
| Table Description | Store Port target information | | | | | |
| Attribute | Attribute Description | Datatype | Allow NULL | Unique | Index | Key |
| s\_id | id of scan | varchar(50) |  |  |  | Foreign key: Scantbl(s\_id) |
| portnum | port number of target | int(11) |  |  |  | Primary key |
| name | name of port | varchar(100) |  |  |  |  |
| version | version of service run in host | varchar(50) |  |  |  |  |
| cpe |  | varchar(50) |  |  |  |  |
| product |  | varchar(200) |  |  |  |  |
| extralinfo | extra infomation | varchar(150) |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User | | | | | | |
| Table Description | Store username and password | | | | | |
| Attribute | Attribute Description | Datatype | Allow NULL | Unique | Index | Key |
| username | username of customer | varchar(50) |  |  |  | Primary key |
| password | password of customer | varchar(50) |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Userinfo | | | | | | |
| Table Description | Store customer infomation | | | | | |
| Attribute | Attribute Description | Datatype | Allow NULL | Unique | Index | Key |
| username | username of customer | varchar(50) |  |  |  | Foreign key: User(username) |
| email | password of customer | varchar(50) |  |  |  |  |
| fullname | Full name of customer | varchar(500) |  |  |  |  |
| phone | Phone number | varchar(10) |  |  |  |  |
| gender | Gender | varchar(10) |  |  |  |  |
| expirydate | expiration dates | date |  |  |  |  |
| usertype | Type of user | int(11) |  |  |  |  |

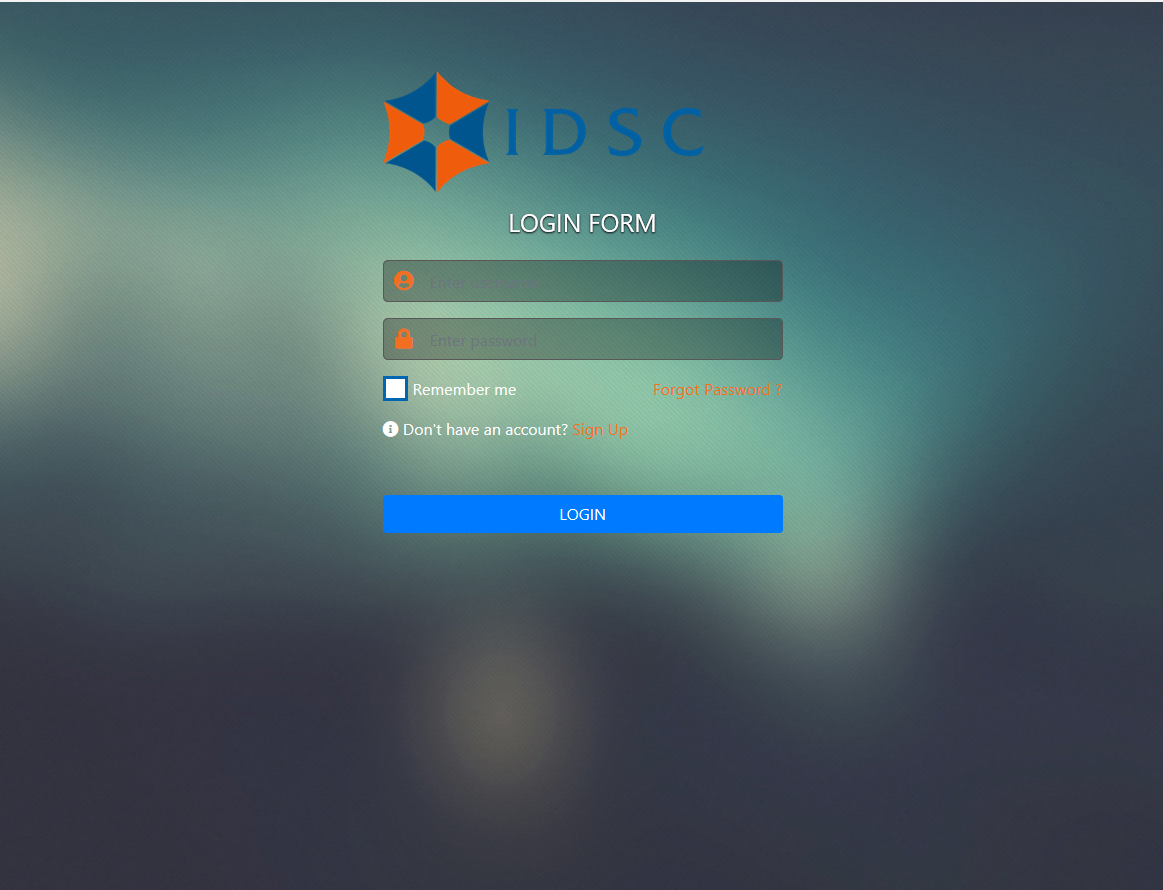
* + 1. ***Detail Design***
       1. ***View Home Page***
          1. ***Design***



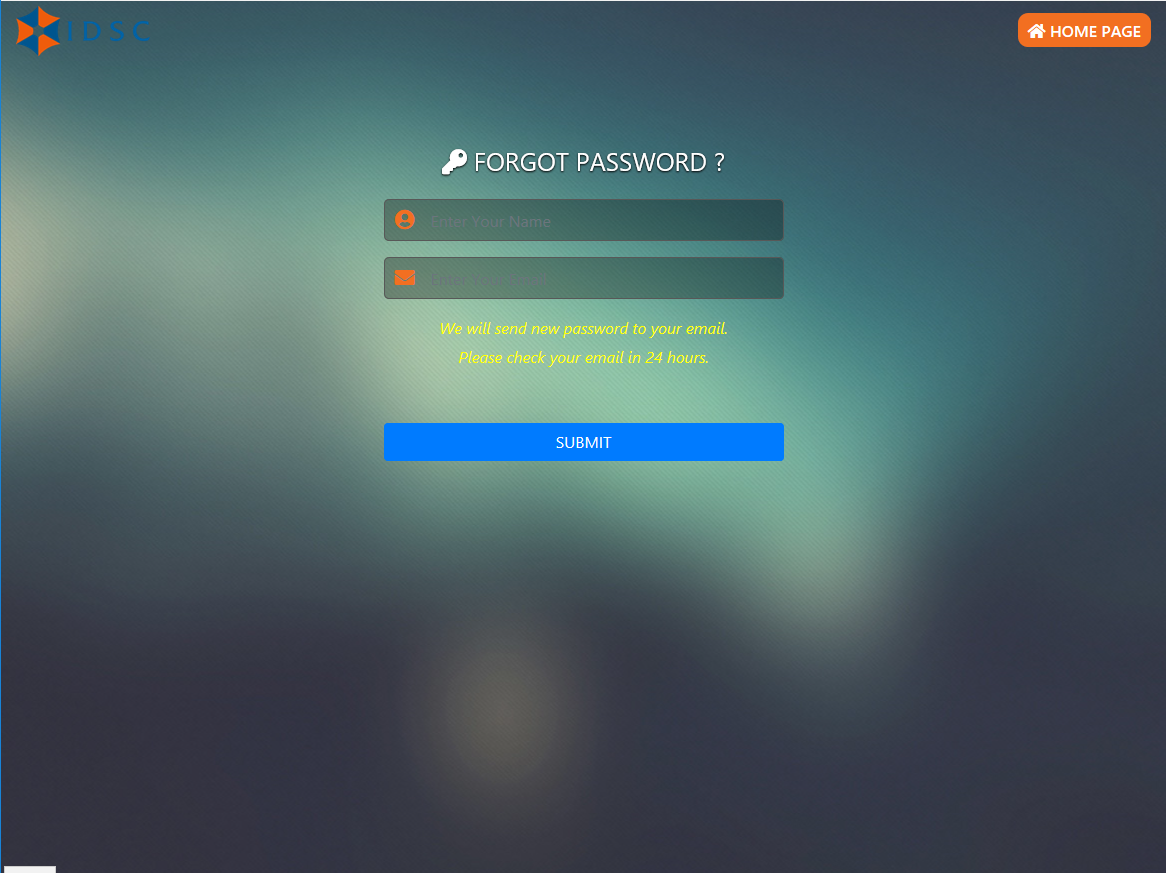
* + - * 1. ***Class Diagram***
        2. ***Sequence Diagram***
      1. ***Register***



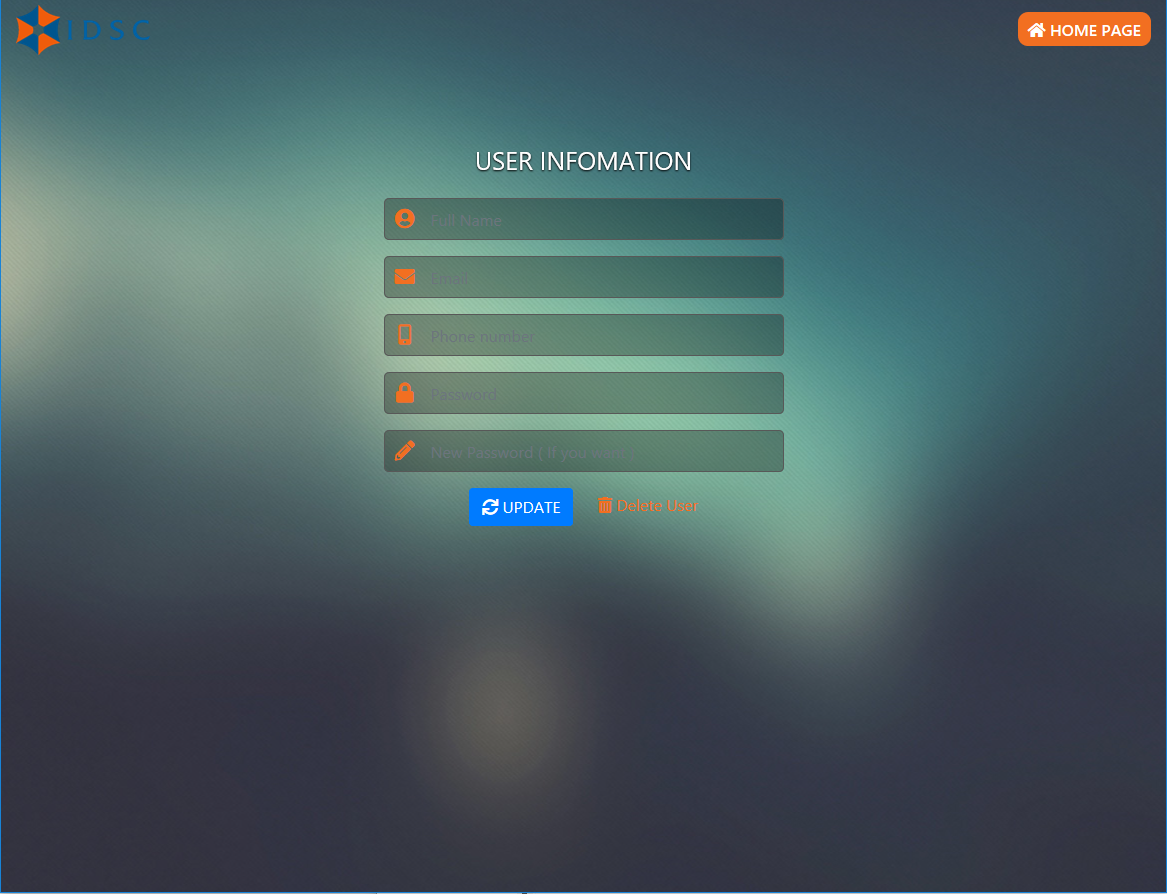
* + - * 1. ***Design***
        2. ***Sequence Diagram***
      1. ***Login***



* + - * 1. ***Design***
        2. ***Sequence Diagram***
        3. ***Forgot Password***



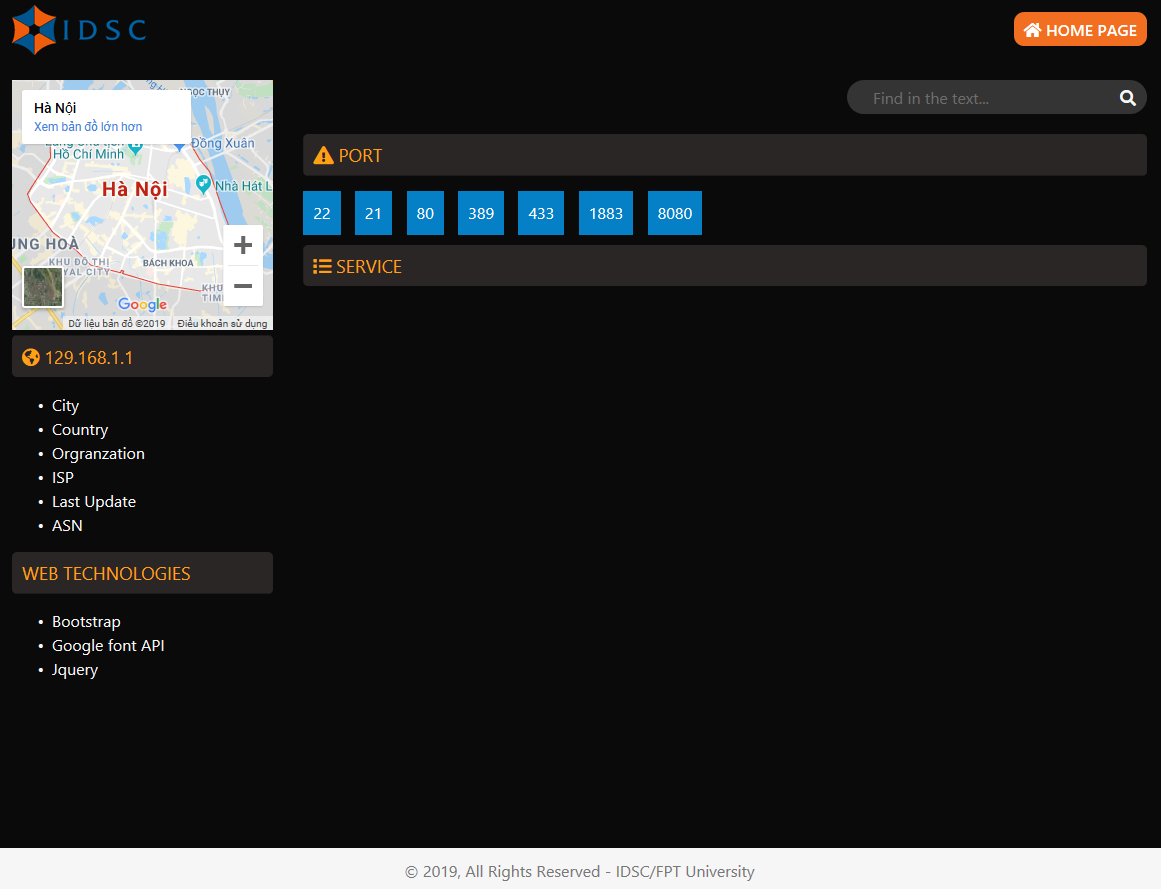
* + - * 1. ***Design***
        2. ***Sequence Diagram***
      1. ***View Profile***
         1. ***Design***



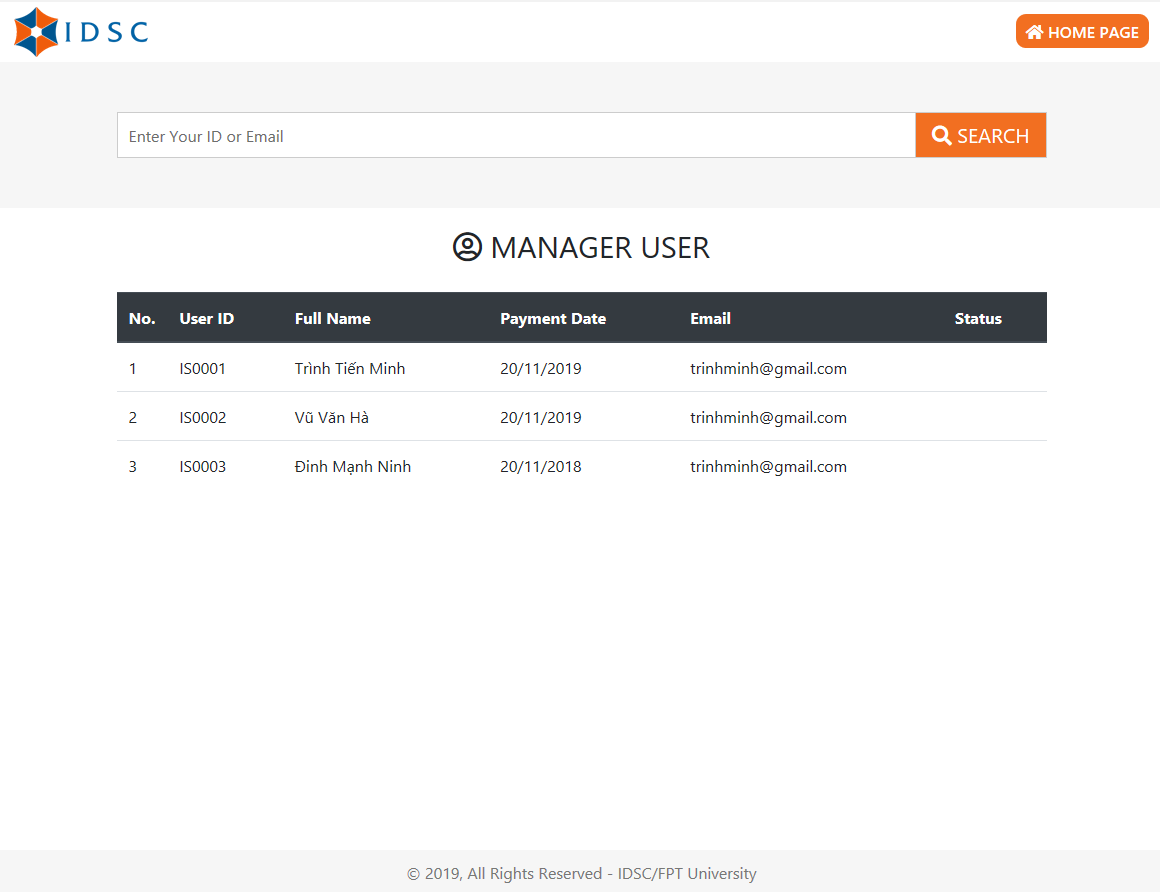
* + - * 1. ***Sequence Diagram***
      1. ***View Payment and Contact***
         1. ***Design***



* + - * 1. ***Sequence Diagram***
      1. ***Scanning***
         1. ***Sequence Diagram***
      2. ***View Result***
         1. ***Design***



* + - * 1. ***Sequence Diagram***
      1. ***View Report***
         1. ***Design***
         2. ***Sequence Diagram***
      2. ***Export Report***
         1. ***Design***
         2. ***Sequence Diagram***
      3. ***View Scanning History***
         1. ***Design***
         2. ***Sequence Diagram***
      4. ***View List User***
         1. ***Design***



* + - * 1. ***Sequence Diagram***
      1. ***Search User***
         1. ***Design***
         2. ***Sequence Diagram***
      2. ***Update User Information***
         1. ***Design***
         2. ***Sequence Diagram***
      3. ***Delete User***
         1. ***Design***
         2. ***Sequence Diagram***
      4. ***Add User***
         1. ***Design***
         2. ***Sequence Diagram***
      5. ***View Analytics***
         1. ***Design***
         2. ***Sequence Diagram***