**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

**DOCUMENT REPORT**

Capstone Project Document

**VN Habit Tracker**

|  |  |
| --- | --- |
| **Group 3** | |
| **Group members** | Lưu Thành Đạt - SE61124  Nguyễn Quang Tuyến - SE62069  Nguyễn Hữu Thắng - SE62447  Phạm Thanh Tùng - SE61628 |
| **Supervisor** | Lại Đức Hùng |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | VHT |

- Ho Chi Minh city, September 14th 2018

Table of Contents

[Definitions, Acronyms, and Abbreviations 5](#_Toc528442800)

[**A.** **Introduction** 6](#_Toc528442801)

[1. Project Information 6](#_Toc528442802)

[2. Introduction 6](#_Toc528442803)

[3. Current Situation 6](#_Toc528442804)

[4. Problem Definition 6](#_Toc528442805)

[5. Proposed Solution 6](#_Toc528442806)

[5.1 Feature functions 7](#_Toc528442807)

[5.2 Advantages and disadvantages 7](#_Toc528442808)

[6. Functional Requirements 7](#_Toc528442809)

[7. Role and Responsibility 8](#_Toc528442810)

[**B.** **Software Project Management Plan** 9](#_Toc528442811)

[1. Problem Definition 9](#_Toc528442812)

[1.1. Name of this Capstone Project 9](#_Toc528442813)

[1.2. Problem Abstract 9](#_Toc528442814)

[1.3. Project Overview 9](#_Toc528442815)

[2. Project organization 12](#_Toc528442816)

[2.1. Software Process Model 12](#_Toc528442819)

[2.2. Roles and responsibilities 12](#_Toc528442820)

[2.3. Tools and Techniques 13](#_Toc528442821)

[3. Project Management Plan 14](#_Toc528442822)

[3.1. Software development life cycle 14](#_Toc528442824)

[3.2. Phase Detail 15](#_Toc528442825)

[3.3. All Meeting Minutes 16](#_Toc528442826)

[4. Coding Convention 16](#_Toc528442827)

[**C.** **Software Requirement Specification** 18](#_Toc528442828)

[1. User Requirement Specification 18](#_Toc528442829)

[1.1 Guest Requirement 18](#_Toc528442830)

[1.2 User Requirement 18](#_Toc528442831)

[1.3 Admin Requirement 19](#_Toc528442832)

[2. System Requirement Specification 19](#_Toc528442833)

[2.1 External Interface Requirement 19](#_Toc528442834)

[2.2 System Overview Use Case 21](#_Toc528442835)

[2.3 List of Use case 22](#_Toc528442836)

[2.3.1 <Guest> Overview use case 22](#_Toc528442837)

[2.3.2 <User> Overview use case 25](#_Toc528442838)

[*Table 25 – Edit group* 45](#_Toc528442839)

[2.3.3 <Admin> Overview use case 46](#_Toc528442840)

[3. Software System Attribute 46](#_Toc528442841)

[3.1. Usability 46](#_Toc528442842)

[3.2. Reliability 46](#_Toc528442843)

[3.3. Availability 46](#_Toc528442844)

[3.4. Security 47](#_Toc528442845)

[3.5. Maintainability 47](#_Toc528442846)

[3.6. Portability 47](#_Toc528442847)

[3.7. Performance 47](#_Toc528442848)

[4. Conceptual Diagram 48](#_Toc528442849)

[**D.** **Software Design Description** 49](#_Toc528442850)

[1. Design Overview 49](#_Toc528442851)

[2. System architecture design 49](#_Toc528442852)

[2.1. Web Application architecture description 50](#_Toc528442853)

[3. Component Diagram 51](#_Toc528442854)

[4. DetailedDescription 53](#_Toc528442855)

[4.1. Class diagram 53](#_Toc528442856)

[4.2. Class Diagram Explanation 54](#_Toc528442857)

[4.3. Interaction Diagram 56](#_Toc528442858)

[5. User Interface Design 58](#_Toc528442859)

[5.1. Login 58](#_Toc528442860)

[5.2. Register 58](#_Toc528442861)

[5.3. Main menu 58](#_Toc528442862)

[5.4. Settings 58](#_Toc528442863)

[5.5. Add habit 58](#_Toc528442864)

[5.6. Edit habit 58](#_Toc528442865)

[5.7. Delete habit 58](#_Toc528442866)

[**E. System Implementation & Test** 58](#_Toc528442867)

[1. Introduction 58](#_Toc528442868)

[1.1. Overview 58](#_Toc528442869)

[2. Database Relationship Diagram 59](#_Toc528442870)

[2.1. Physical diagram 59](#_Toc528442873)

[2.2. Data dictionary 59](#_Toc528442874)

[3. Test plan 59](#_Toc528442875)

[3.1. Features to be tested 59](#_Toc528442876)

[3.2. Features not to be tested 59](#_Toc528442877)

[4. System Testing Test Case 59](#_Toc528442878)

[4.1. Test case 59](#_Toc528442879)

[4.2. テストケース 63](#_Toc528442880)

List of tables

[Table 1: Roles and Responsibilities 9](#_Toc528484525)

[Table 2 : Hardware Requirement for Server 12](#_Toc528484526)

[Table 3 : Hardware Requirement for Mobile 12](#_Toc528484527)

[Table 4 : Software requirements 13](#_Toc528484528)

[Table 5 : Roles and responsibilities 14](#_Toc528484529)

[Table 6: Tools List 14](#_Toc528484530)

[Table 7: Technique List 14](#_Toc528484531)

[Table 8: Software Development Life Cycle Detail 15](#_Toc528484532)

[Table 9: Phase 1: Infrastructure 16](#_Toc528484533)

[Table 10: Phase 3: Web service 17](#_Toc528484534)

[Table 11: Phase 4: Mobile app 17](#_Toc528484535)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| VHT | VN Habit Tracker |
|  |  |

# **Introduction**

## Project Information

* Project name: **VN Habit Tracker**
* Abbreviation: **VHT**
* Product Type: **Web app & Mobile app**
* Start Date: **September 10th 2018**

## Introduction

Habits are routine behaviors done on a regular basis. They are recurrent and often unconscious patterns of behavior and are acquired through frequent repetition. Many of these are unconscious as we don’t even realize, we are doing them.

Good habits are hard to develop when we use the typical tactic of trying to pump up our motivation to overcome our psychological resistance. Research has shown that motivation is an ineffective factor or creating lasting behavior change. Fortunately, we have also will build an excellent alternative, which is called VHT. It will let us tunnel right through the resistance barrier.

First thing is to track habits! You can also track recurring behavior that you want to keep under control. This tracker can be made for daily use, so you get a very clear overview of all your good and bad habits. By keeping a habit tracker, you are more conscious of your behaviors. You know actually what you are doing, more importantly, what not, what you should do. Without this tracker it can be very easy to believe you are indeed doing everything you want, but in the meantime, secretly, you are not. The tracker is a tool to transform your goals into good habit.

## Current Situation

* Everyday our lives are governed by habit. These habits are the little routines and small ways of doing things.
* Most people are unaware or are only slightly aware that they’re doing them.
* VHT helps you to set goals, keep track of your habits and tackle your bad habits.

## Problem Definition

* **Disadvantages:**
  + Vietnamese users do not have the official application for tracking habit.
  + Customer easily give up on using.

## Proposed Solution

Our proposed solution is to build an application called “VN Habit Tracker” to resolve the current problems. The application collects (in real-time) all habits from customer. Moreover, it specifically helps them to track their habits daily, weekly and monthly. This application will motivate them by showing the notification and reminder which depend on their setting. Customer can check their currently process. We also design the app which has statistics that helps customer to see their efforts.

VHT system is an app application with following functions:

### Feature functions

* + Plan habits with an easy-to-use interface.
  + Habits fit into a number of different schedules, including specific times or days.
  + View stats in a number of different ways, with different data point.
  + Reminders.
  + Make sure no habit is forgotten with daily reminders.
  + Data Export.
  + Export your data to Excel or Numbers (CSV) and chart your progress.
  + Interactive Notifications.
  + Complete or snooze a habit right from the notification banner.
  + Weekly Targets.
  + Set how many days a week you would like to complete a habit.
  + Notes.
  + Write or dictate notes to capture all details of a habit. Great for exercise routines and daily journals.
  + Skip Functionality.
  + Going on vacation or taking a day off? Skip habits without breaking your chain.
  + Report.

### Advantages and disadvantages

These are advantages and disadvantages of current situation:

* **Advantages:**
  + - Customer will be easy to understand.
    - Friendly interface.
    - Motivate customer to change their bad habits.
    - VHT saves times and space for customer.
  + **Disadvantages:**
    - Advertising strategy difficulty of application.

## Functional Requirements

Function requirements of the system are listed as below:

Add habits:

* + Choose habit: add a bad habit you want to reform or build a good one to improve.
  + Name habit: name that habit.
  + Choose period: choose daily, weekly, monthly.
  + Set goal: set up the goal you want to archive.
  + Choose group: set that habit belongs to which group that you want.
  + Adjust reminder: set time for the app to remind you.
  + Write description: write something that can motivate you.

Show statistics: show the statistics to display the chart in the present or in the past.

Adjust setting:

* + Set goal.
  + Set reminder.

Adjust filter: display the result in the form that customer want.

* + Choose goal period.
  + Choose goal type.
  + Choose goal values.

Adjust chosen habit:

* + Edit chosen habits.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Lại Đức Hùng | Project Supervisor | Supervisor | Hungld5@fe.edu.vn |
| 2 | Lưu Thành Đạt | Developer | Leader | Datltse61124@fpt.edu.vn |
| 3 | Nguyễn Quang Tuyến | Developer | Member | Tuyennq62069@fpt.edu.vn |
| 4 | Nguyễn Hữu Thắng | Developer | Member | Thangnhse62447@fpt.edu.vn |
| 5 | Phạm Thanh Tùng | Developer | Member | Tungptse61628@fpt.edu.vn |

Table 1: Roles and Responsibilities

# **Software Project Management Plan**

## Problem Definition

### Name of this Capstone Project

* **Official name**: VN Habit Tracker
* **Vietnamese name**: Ứng dụng theo dõi, cải tiến thói quen cho người Việt.
* **Abbreviation**: VHT

### Problem Abstract

Currently, in Viet Nam, the majority of people use smartphones every day. However, there is not a specific application for Vietnamese people to create habits, manage daily habits and track habits. The existing tracking habit applications are from foreign developer and thus may not really applicable for Vietnamese users. So we have to find the best convenient way to make our application simplest and easy to use for anyone.

To solve this particular problem, we decided to provide an application that helps users do that*.* VHT will help users to create habits, group habits, and prompt users to implement those habits. Create a goal for the user to try to achieve that habit over time and manage the statistics of the user’s system habits.

### Project Overview

#### Current Situation

Below are the problems encountered in this project:

* **User Habit:** people do not have a habit to set habit and manage habit by an application.
* **User:** easily give up when they depressed, unmotivated to continue a habit.
* Currently there is no application dedicated to Vietnamese.
* **Server crash**: The user's data will be lost when the server crashes. Because all data is stored on the server.
* **Interface**: user easy-to-use interface in VHT’s application.

#### The Proposed System

According to the survey of Vietnamese users, we find that most Vietnamese do not have a habit of scheduling, do not schedule what week they will do. So our solution now is to create an application specifically for Vietnamese users to improve and create their habits.

We build a system that is always maintained so that users can log in and use it on their mobile applications anytime, anywhere.

To solve the problem of creating habits and improving the habits of Vietnamese users, we need to know the Vietnamese people, what makes them do not create their daily routine.

Our system consists of two main systems: website and mobile applications. On the website, administrators manage users and statistical habits of Vietnamese. For mobile applications, we allow users to create and manage their habits, and we also receive feedback from Vietnamese users.

##### ***Web Application***

Web application consists of three main parts:

* For user:
  + View general data.
  + Export data to excel.
* For administrator:
  + Manage accounts.

Besides, website application also provides an API interface for two mobile applications to retrieve, update data from mobile applications.

##### *Mobile Application*

The mobile applications included functions as below:

* User:
  + Create habits.
  + Manage habits.
  + Group habits.
  + Habits statistic.
  + Complete or snooze a habit.
  + Export data.

#### Boundaries of the System

* This application is built on the habits of Vietnamese. Our main target is helps them to set goals, keep track of habits and tackle bad habits.
* VN Habit Tracker which deployed this application must set up devices, includes:
  + Daily reminders.
  + Data export to number and chart progress.
  + Interactive notifications.
  + Weekly targets
  + Notes details of a habit.
* The completed product includes:
  + Website application.
  + Android mobile application.

#### Future Plans

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

* + Extend and create community using VHT.
  + Develop new features to set and remind users to create a habit: real time, remind by location..v..v..
  + Combined with smart watch and external devices to manage and message users.
  + Development Environment with VHT.

##### ***Hardware requirements***

* **For web application server**

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

Table 2 : Hardware Requirement for Server

* **For Mobile**

|  |  |  |
| --- | --- | --- |
| Android | Minimum | Recommended |
| Internet Connection | Wi-Fi or 3G (1 Mbps) | Wi-Fi or 3G (8 Mbps) |
| Operating System | Android 4.4.2 | Android 6.0.0 |
| Mobile Processor | Cortex-A7 Dual-Core 1.3GHz | Cortex-A7 Dual-Core 1.3GHz |
| Mobile Memory | 1GB of RAM | 2GB of RAM or more |

Table 3 : Hardware Requirement for Mobile

##### ***Software requirements***

|  |  |  |
| --- | --- | --- |
| Software | Name / Version | Description |
| Operating system | Window Server 2012 R2 | Operating system and platform for development |
| Environment | Java | Specification for developing web application |
|  |  |  |
| IDE | IntelliJ IDEA, Android Studio v2.1 | Used for implement website and Android Mobile Application. |
| Design Model tool | Star UML v2.5.1 | Used for creating modal and diagrams. |
| DBMS | Microsoft SQL Server 2014 | Used to create & manage the database for system |
| Document storage | Trello | Used for storing document |
| Store and manage source code | Git Hub | Used to store all source code |
| Web browser | Chrome or above | Testing browser |

Table 4 : Software requirements

## Project organization



### Software Process Model

The software process model used in developing the VHT project is based on the Water Fall model. We choose this model because of the following reason:

* The features of this product is strongly connected, therefore, a solid architect and clear requirement need to be setup in order to un the project smoothly.
* The requirements are not supposed to be changed rapidly.



Figure 1 : Waterfall model

Reference: Page 30, chapter 2, Software process model, SOFTWARE ENGINEERING 9th Edition, by Ian Sommerville.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Lại Đức Hùng | Supervisor, Project Manager | * Specify user requirements * Control the development process * Give out technique and business analysis support |
| 2 | Lưu Thành Đạt | Team leader, B.A, Developer, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Nguyễn Hữu Thắng | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
|  | Phạm Thanh Tùng | Team member,  B.A, Developer,  Tester | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding |
| 4 | Nguyễn Quang Tuyến | Team member,  B.A, Developer,  Tester | * Clarifying requirements * Prepare documents * Create test plan * Coding * Testing |

Table 5 : Roles and responsibilities

### Tools and Techniques

|  |  |
| --- | --- |
| Tool | Name / version |
| Web server | IIS |
| Development tool | IntelliJ IDEA, Android Studio |
| DBMS | SQL Server 2014 |
| Source control | Git hub & SourceTree |
| Modeling tool | Star UML v5.0.1 |
| Document tool | Microsoft Word 2016 |

Table 6: Tools List

|  |  |
| --- | --- |
| Technique | Name / version |
| Frontend | HTML5, CSS, JavaScript, Angular |
| Backend | Java, Spring Boot, Hibernate, Android |

Table 7: Technique List

## Project Management Plan



### Software development life cycle

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

Table 8: Software Development Life Cycle Detail

### Phase Detail

#### Phase 1: Infrastructure

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Assessment | - Determine requirements.  - Determine requirements for System and API. | * DatLT * ThangNH * TuyenNQ * TungPT |
| 2. Selection | - Determine system architecture: Java Web & Android.  - Determine software design pattern: Repository & Service.  - Determined all core functions. | * DatLT * ThangNH * TuyenNQ * TungPT |
| 3. Development | - Create the main structure of project. | * DatLT * ThangNH * TuyenNQ * TungPT |
| 4. Review | - Review all completed works and presentation.  - Create sprint backlog. | * DatLT * TungPT |

Table 9: Phase 1: Infrastructure

#### Phase 3: Web service

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

Table 10: Phase 3: Web service

#### Phase 4: Mobile app

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

Table 11: Phase 4: Mobile app

### All Meeting Minutes

Meeting minutes are contained in folder “Meeting minutes”.

### Coding Convention

This “Coding Convention” is using for develop project in both Java web admin and Android application:

* Naming Conventions:
* Folders (Java web admin): folders name are always unique and written in lowercase.

Ex: customer, servlet, sample…

* Packages (Android application): packages/folders name are always unique and written in lowercase.
* Classes/Interfaces: classes or interfaces should be noun, name using Pascal case, in mixed case with the first letter of each internal word capitalized.

Ex: User, ImageSession, CustomerDAO…

* Methods: methods should be verb, name using Camel case, in mixed case with the first letter lower case, with the first of each internal word capitalized.

Ex: run (), crawlNew (), getSearchResults () …

* Variables: name using Camel case, in mixed case with the first letter lower case, with the first of each internal word capitalized. Variable names should not start with underscore \_ or dollar sign $ characters, even though both are allowed.

Ex: count, myVariable, isValidated…

* Constants: constants should be all uppercase with words separated by underscores.

Ex: MAX\_VALUE, GET\_SEARCH\_RESULT

* Comment:
* All source files should begin with a comment that lists the class name, description, date, author:

/\*

\* Class name

\*

\* Description

\*

\* Date

\*

\* Author

\*/

* All methods should begin with a comment that lists the method name, description, date, author:

/\*

\* Method name

\*

\* Description

\*

\* Date

\*

\* Author

\*/

* Comment code is 2 slashes “//”
* Comment to explain code need 4 slashes “////”
* Block comment used to provide description of files, methods, structures and algorithms.
* Indentation: Four spaces should be used as the unit of indentation
* Declarations: one declaration per line. Do not put different types on the same line

Ex: int minValue;

# **Software Requirement Specification**

* + - 1. User Requirement Specification
  1. Guest Requirement

Guest is a person who doesn’t have access to the system. Guest can use some functions in the system. To use all functions, guest must login. These are some functions that guest can use:

• Login.

* Register.
  1. User Requirement

User is a guest who logged into the system with user’s role. There are some functions that user can use:

* Manage habit

+ View habit

+ Delete habit

+ Edit habit

+ Add habit

+ Check times

* Manage group

+ Add group

+ Delete group

+ Edit group

* Manage settings

+ Sort habit

+ Set reminders

+ Feedback and contact

* View statistics
* Logout
  1. Admin Requirement

Admin is the person who manages the system, a super user of system. There is some functions admin can use:

* Manage user

+ View user

+ Reset password

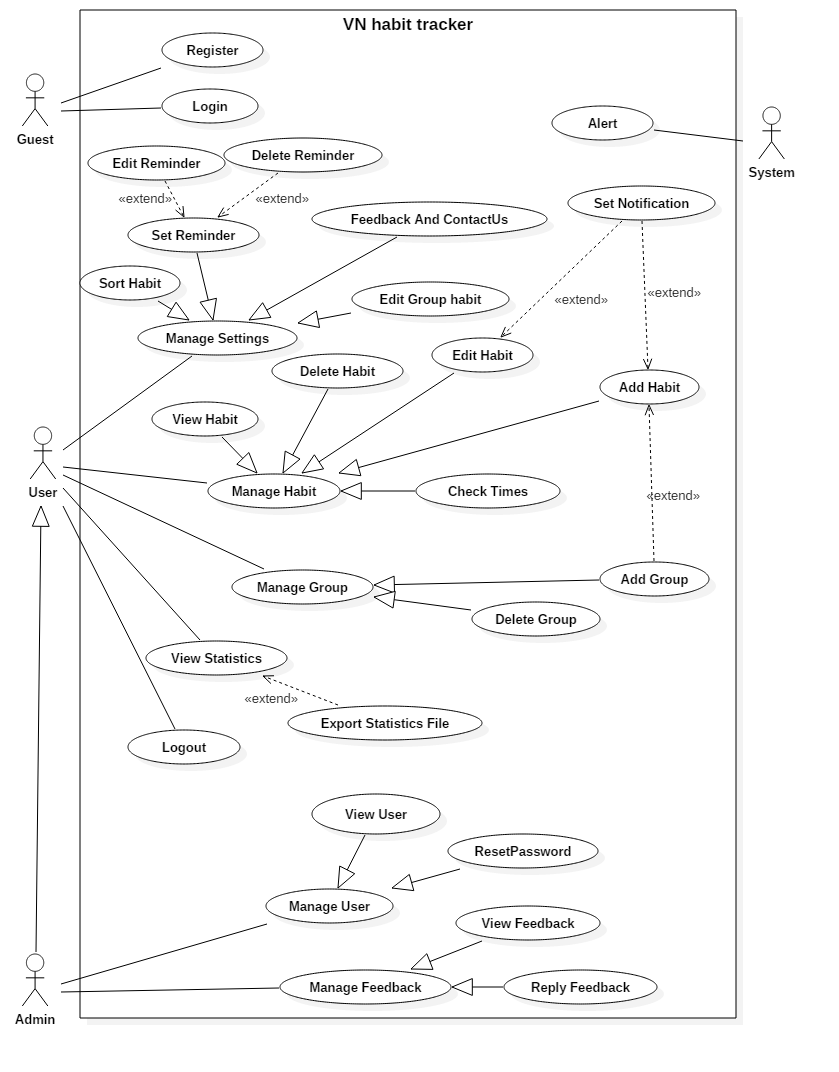
* Manage feedback

+ View feedback

+ Reply feedback

1. System Requirement Specification
   1. External Interface Requirement
      1. User Interface

* General requirement for graphic user interface is the GUI should be simple, clear, intuitive, and reminiscent.
* The interface design is an iterated process includes design, sketching and user assessment.
  + 1. Hardware Interface
* Smartphone, desktop, laptop or tablet has the ability to connect to the internet.
* The system will use the standard hardware and data communications resources of a standard computer.
  + 1. Software Interface
* Run with Chromes (v60 or higher), Firefox (v46.1 or higher)
* The screen must bigger than 1024x768.
* Mobile application: Android studio (version 3.1.4), Genymotion (version 2.12.2).
  + 1. Communication Protocol
* Using HTTP/HTTPS protocol.
  1. System Overview Use Case



*Figure 2- System Overview Use Case*

* 1. List of Use case
     1. <Guest> Overview use case



* + - 1. *<Guest> Login*

**Use Case Diagram**

****

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows the guest login the application.   **Goal:**   * Guest logged in the system.   **Triggers:**   * Guest clicks “Đăng nhập” button.   **Preconditions:**   * Guest at “Đăng nhập” page.   **Post Conditions:**   * **Success:** System shows “Đăng nhập thành công.” message. * **Fail:** System shows “Đăng nhập thất bại” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest goes to Login page by clicking “Đăng nhập” button. | System requires information:   * Username: text input. * Password: text input with least 6 characters. | | 2 | Guest inputs information (Username and password)  Guest clicks “Đăng nhập” button. | User will login the system with roles of this account.  [Exception 1,2,3,4,5] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Username is empty. | Show error message. | | 2 | Password is empty. | Show error message. | | 3 | Username is not correct | Show error message. | | 4 | Password is not correct. | Show error message. | | 5 | Network problem exception. | Show message “Lỗi kết nối!” |   **Relationships:**  N/A  **Business Rules:**   * After logged in, system will redirect to main page. | | | |

Table 12:Login

* + - 1. *<Guest> Register*

**Use Case Diagram**

****

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Register | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * Guest   **Summary:**   * This use case allows the guest register an account.   **Goal:**   * Guest register an account.   **Triggers:**   * Guest clicks “Đăng ký” button.   **Preconditions:**   * Guest must have an email account. * Guest at “Đăng ký” page.   **Post Conditions:**   * **Success:** System shows “Đăng ký thành công!” message. * **Fail:** System shows “Đăng ký thất bại” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Guest goes to Login page by clicking “Đăng ký” button. | System requires information:   * Username: text input. * Email: text input * Password: text input with least 6 characters. * Password again: text input | | 2 | Guest inputs information (Username and password)  User clicks “Login” button. | User will login the system with roles of this account.  [Exception 1,2,3,4,5] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | |  |  |  |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Username is empty. | Show error message. | | 2 | Password is empty. | Show error message. | |  | Password is less than 6 character. | Show error message. | |  | Email is empty. | Show error message. | |  | Email is wrong format. | Show error message. | | 5 | Network problem exception. | Show message “Lỗi kết nối!” |   **Relationships:**  N/A  **Business Rules:**   * After guest registered, system will redirect to login page. | | | |

Table 13: Register

* + 1. <User> Overview use case
       1. *<User> Sort habit*

**Use Case Diagram**



*Figure 5- <User> Sort habit*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Sort Habit | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user sort habit.   **Goal:**   * User sort habit.   **Triggers:**   * User click on “settings” button. * After that, user click on “Sort habit” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows “Successful to sort habit!” message. * **Fail:** System shows “Failed to sort habit” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Sort habit” button. | System shows filter for user choice to sort.  + Sort by period.  + Sort by type.  + Sort by tracking. | | 2 | User click on “Done” button. | - The system validates the information and shows confirmation message.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “cancel” button. | Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | System List goal fail. | System shows “Error!” message |   **Relationships:**  N/A  **Business Rules:**   * After set goals success. The user will return to the settings page. * User can add new group. | | | |

Table 14: Sort habit

* + - 1. *<User> Feedback and contact us*

**Use Case Diagram**



*Figure 6- <User> Feedback and contact us*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_03** | | | |
| **Use Case No.** | 03 | **Use Case Version** | 1.0 |
| **Use Case Name** | Feedback and contactUs | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to feedback.   **Goal:**   * Feedback the application.   **Triggers:**   * User click on “settings” button. * After that, user click on “Feedback and contact us” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows “successful to feedback!”message. * **Fail:** System shows “Failed to feedback!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Set goals” button. | The system shows a form for user feedback.  + Name: text input  + Your email: text input  + Your email again: text input  + Message: text input | | 2 | User click on “OK” button. | - The system validates the information and shows confirmation message.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “cancel” button. | Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost internet. | System shows “Error!” message. | | 2 | The field is not blank. | System shows “must be not blank” message. |   **Relationships:**  N/A  **Business Rules:**   * User can feedback about error, bug,… on application. * After feedback success. The user will return to the settings page. | | | |

Table 15: Feedback and contact us

* + - 1. *<User> Set reminder*

**Use Case Diagram**



*Figure 7 - <User> Set reminder*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_04** | | | |
| **Use Case No.** | 04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Set reminder | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user set a reminder for system.   **Goal:**   * Set reminder for system.   **Triggers:**   * User click on “settings” button. * After that, user click on “Set reminder” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows list reminder messages. * **Fail:** System shows error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Set reminder” button. | The system shows a form for use set reminder.  + General app reminder: list reminders  + reminder sound: list option  + Snooze time: list option | | 2 | User click on “Done” button. | - The system validates the information and shows confirmation message.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “cancel” button. | Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | System shows list reminders fail. | System shows “Error!” message |   **Relationships:**  N/A  **Business Rules:**   * After setting reminder success. The user will return to the settings page. * Only use sound default to set reminder. * The maximum snooze time is 24 hours. | | | |

Table 16: Set reminder

* + - 1. *<User> Add habit*

**Use Case Diagram**



*Figure 8- <User> Add habit*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_05** | | | |
| **Use Case No.** | 05 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add habit | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to add a new habit.   **Goal:**   * Add a new habit from the schedule.   **Triggers:**   * User click on “Add habit” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows “Successful to add a new habit!” message. * **Fail:** System shows “Failed to add a new habit!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Add habit” button. | The system request information in form:   * Habit name: input text * Habit type: input text * Habit color: select button * Group habits: select option * Start date: datetime * End date: datetime | | 2 | User click on “OK” button. | - The system validates the information and shows confirmation message.  [Exception 1, 2] | | 3 | User click on “Save” button | - System shows “Successful to add a new habit” message |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | User add an existed habit in the application | Show message to notify user this habit is existed. | | 2 | User add new fail | System shows “Failed to add a new habit!” message. |   **Relationships:**  Add group, Set notification.  **Business Rules:**   * The habit’s name must be under 255 characters and not null. * After adding habit success. The user returns to the habit page. | | | |

Table 17: Add habit

* + - 1. *<User> Delete habit*

**Use Case Diagram**



*Figure 9 - <User> Delete habit*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_06** | | | |
| **Use Case No.** | 06 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete habit | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to delete a habit.   **Goal:**   * Delete a habit from the schedule.   **Triggers:**   * User click on “Delete habit” button.   **Preconditions:**   * Guest must login. * The habit is existed.   **Post Conditions:**   * **Success:** System shows “Successful to delete the habit!” message. * **Fail:** System shows “Failed to delete this habit!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Delete habit” button. | Shows a form to ask user want to delete habit. | | 2 | User click on “OK” button. | - The system validates the information and shows confirmation message.  [Exception 1] | | 3 | User click on “Yes” button. | - That habit will be deleted. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | None of any habit was selected | Shows “Failed to delete a habit!” message. |   **Relationships:**  N/A  **Business Rules:**   * When a user deleted a habit, this habit will be hidden. * The information of the habit is deleted to the local. | | | |

Table 18: Delete habit

* + - 1. *<User> Edit habit*

**Use Case Diagram**



*Figure 10 - <User> Edit habit*

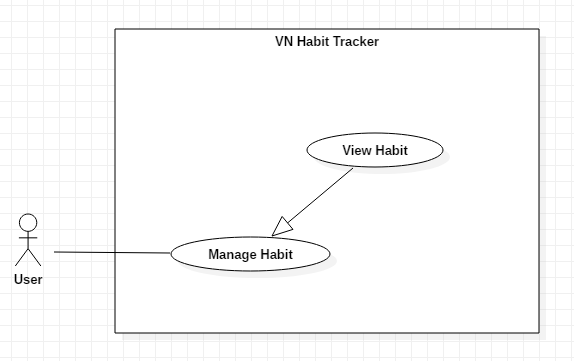
**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_07** | | | |
| **Use Case No.** | 07 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit habit | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to edit a habit.   **Goal:**   * Edit habit when user input.   **Triggers:**   * User click on “Edit habit” button.   **Preconditions:**   * Guest must login. * The habit is existed.   **Post Conditions:**   * **Success:** System shows “Successful to edit a habit!” message. * **Fail:** System shows “Failed to edit a habit!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Edit habit” button. | The system request information in form:   * Habit name: input text * Habit type: input text * Habit color: select button * Group habits: select option * Start date: datetime * End date: datetime | | 2 | User click on “OK” button | - The system validates the information and shows confirmation message.  [Exception 1, 2] | | 3 | User click on “YES” button | - System shows “Successful to edit a habit!” message. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | The field is blank | The system shows “Must not be blank!” message. | | 2 | System edit habit fail | System shows “Failed to edit a habit!” message. | |  |  |  |   **Relationships:**  N/A  **Business Rules:**   * The habit’s name must be under 255 characters and not null. * The information of Habit is sent to the server. * After Edit habit success. The user will return to the habit page. | | | |

Table 19: Edit habit

* + - 1. *<User> View habit*

**Use Case Diagram**



*Figure 11 - <User> View habit*

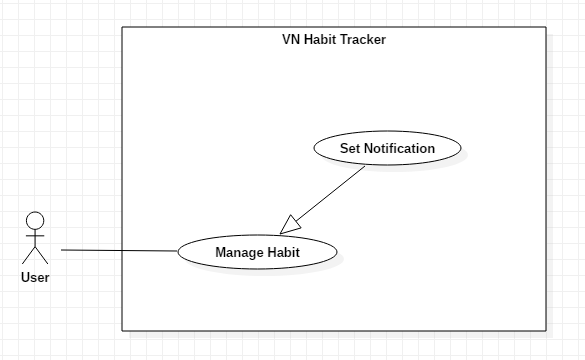
**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_08** | | | |
| **Use Case No.** | 08 | **Use Case Version** | 1.0 |
| **Use Case Name** | View habit | | |
| **Author** | ThangNH | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to view all habit are added.   **Goal:**   * View habit when user want to see their added habit.   **Triggers:**   * User click on “View habit” button. * User click on “Login habit” button.   **Preconditions:**   * Guest must login. * The habit is existed.   **Post Conditions:**   * **Success:** System shows all habit are added. * **Fail:** User can’t see their added habit.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “View habit” button.  User click on “Login” button | The system show information in form:   * Habit name: text * Habit description: text * Habit color * Group habits: text * Habit time: date time | | 2 | User click on “View habit” button | - The system validates the request and shows the added habit.  [Exception 2] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | There are no existed habit | The system shows nothing | | 2 | System login habit fail | System shows “Please reconfirm your password!” message. | |  |  |  |   **Relationships:**  N/A  **Business Rules:**   * The habit’s information must be existed first. * The information of Habit is sent to the server. * After View habit successfully. The user will see the list of all habit. | | | |

Table 20: View habit

* + - 1. *<User> Set notification*

**Use Case Diagram**



*Figure 12 - <User> Set notification*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_09** | | | |
| **Use Case No.** | 09 | **Use Case Version** | 1.0 |
| **Use Case Name** | Set notification | | |
| **Author** | ThangNH | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to set the notification.   **Goal:**   * Set notification for a habit.   **Triggers:**   * User click on “Set notification” button.   **Preconditions:**   * Guest must login. * User click on “Edit habit” button. * User click on “Add habit” button. * The habit is existed.   **Post Conditions:**   * **Success:** Mobile devices will show the pop-up with sound. * **Fail:** Nothing will be showed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Set notification” button. | The system request information in form:   * Habit name: text * Habit description: text * Habit time: date time * Snooze length * Sound | | 2 | User click on “OK” button | - The system validates the request and shows the added information.  [Exception 1, 2] | | 3 | User click on “Yes” button | System shows “Successful to edit a habit!” message. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | The field is blank | The system shows “Must not be blank!” message. | | 2 | System fail to set notification | System shows “Failed to set notification!” message. | |  |  |  |   **Relationships:**  N/A  **Business Rules:**   * The habit must be existed first. * After set notification successfully. The user will see the notification of the habit when time comes. | | | |

Table 21: set notification

* + - 1. *<User> Add group*

**Use Case Diagram**



*Figure 13 - <User> Add group*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_10** | | | |
| **Use Case No.** | 10 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add group | | |
| **Author** | TungPT | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to add group.   **Group:**   * Add a group while add the habit.   **Triggers:**   * User click on “Add group” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows “Successful to add a group!” message. * **Fail:** System shows “Failed to add a group!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Add group” button. | The system request information in form:   * Group period: select input * Set your group: input text | | 2 | User click on “Ok” button. | - The system validates the request and shows the added information.  [Exception 1] | | 3 | User click on “Save” button | - That group will be added. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | None of any group was selected | Shows “Failed to add a group!” message. |   **Relationships:**  add habit  **Business Rules:**   * The group name must be under 255 characters and not null. * After adding group success. The user returns menu list habit page. | | | |

Table 22: Add group

* + - 1. <*User> Delete group*

**Use Case Diagram**



*Figure 14 - <User> Delete group*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_11** | | | |
| **Use Case No.** | 11 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete group | | |
| **Author** | TungPT | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to delete group.   **Group:**   * Delete a group from the group.   **Triggers:**   * User click on “Delete group” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows “Successful to delete a group!” message. * **Fail:** System shows “Failed to delete a group!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Delete group” button. | Shows a form to ask user want to delete group. | | 2 | User click on “OK” button | - The system validates the request and shows the added information.  [Exception 1] | | 3 | User click on “Save” button | - That group will be deleted. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | None of any group was selected | Shows “Failed to add a group!” message. |   **Relationships:**  N/A  **Business Rules:**   * When a user deleted a group, this group will be hidden. * The information of the group is deleted to the local. | | | |

Table 23: Delete group

* + - 1. *<User> Edit group*

**Use Case Diagram**



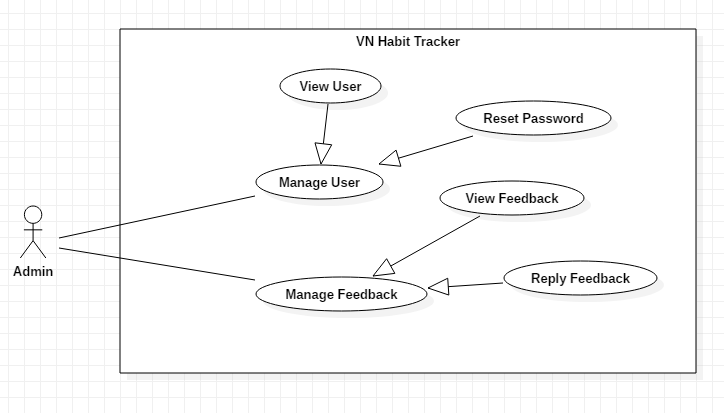
*Figure 15 - <User> Edit group*

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit group | | |
| **Author** | TungPT | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to edit group.   **Group:**   * Edit a group from the group.   **Triggers:**   * User click on “Edit group” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows “Successful to edit a group!” message. * **Fail:** System shows “Failed to edit a group!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Edit group” button. | The system request information in form:   * Group period: select input * Set your group: input text | | 2 | User click on “OK” button | - The system validates the request and shows the added information.  [Exception 1] | | 3 | User click on “Save” button | - That group will be edit. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | None of any group was selected | Shows “Failed to edit a group!” message. |   **Relationships:**  N/A  **Business Rules:**   * The group name must be under 255 characters and not null. * The information of Group is sent to the server. * After Edit group success. The user will return to the menu list habit page. | | | |

Table 24: Edit group

* + 1. <Admin> Overview use case



* + - 1. *<Admin> View user*

**Use Case Diagram**

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | View User | | |
| **Author** | TuyenNQ | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows the admin to view all user.   **Group:**   * View infomartion user.   **Triggers:**   * Admin click on “quản lý người dùng” button. * Admin click on “thông tin người dùng” button   **Preconditions:**   * Admin must login.   **Post Conditions:**   * **Success:** System shows message. * **Fail:** System shows message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click on “Thông tin người dùng” button. | The system shows all list user use applications. | | 2 | User click into a user  [Alternative 1] | - The system show all information of this user.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close list of user. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost internet | Shows “mất kết nối! Xin thử lại” message. |   **Relationships:**  N/A  **Business Rules:**   * After view information of user. The admin will return to the main page. * The information about a user is private and only admin is seen. | | | |

* + - 1. *<Admin> Reset password*

**Use Case Diagram**

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Reset password | | |
| **Author** | TuyenNQ | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows the admin to reset password if user sent request.   **Group:**   * Reset password when user sent request.   **Triggers:**   * Admin click on “quản lý người dùng” button. * Admin accept request of user. * Admin click on “tạo lại mật khẩu” button   **Preconditions:**   * Admin must login. * User must sent request.   **Post Conditions:**   * **Success:** System shows “Tạo mới mật khẩu thành công.” message. * **Fail:** System shows “Tạo mới mật khẩu thất bại” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin accept request of user to reset password. | The system shows “Đã chấp nhận yêu cầu” message. | | 2 | Admin click on “Tạo mới mật khẩu” button. | The system shows all list user use applications. | | 3 | Admin click into a user want to change the password.  [Alternatives 1] | - The system sent a link to the user to change password.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close list of user. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost internet | Shows “mất kết nối! Xin thử lại” message. |   **Relationships:**  N/A  **Business Rules:**   * After reset password of a user. The admin will return to the main page. | | | |

* + - 1. *<Admin> View feedback*

**Use Case Diagram**

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Feedback | | |
| **Author** | TuyenNQ | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows the admin to view all feedback of users.   **Group:**   * View feedback of users.   **Triggers:**   * Admin click on “quản lý phản hồi” button.   **Preconditions:**   * Admin must login. * User sent feedback.   **Post Conditions:**   * **Success:** System shows message. * **Fail:** System shows message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click on “quản lý phản hồi” button. | The system shows all feedback of users. | | 2 | Admin click into a feedback.  [Alternative 1] | - The system show content of this feedback.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close list of feedbacks. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost internet | Shows “mất kết nối! Xin thử lại” message. |   **Relationships:**  N/A  **Business Rules:**   * After view feedback of user. The admin will return to the main page. * The feedback of user is private and only admin is seen. | | | |

* + - 1. *<Admin> Reply feedback*

**Use Case Diagram**

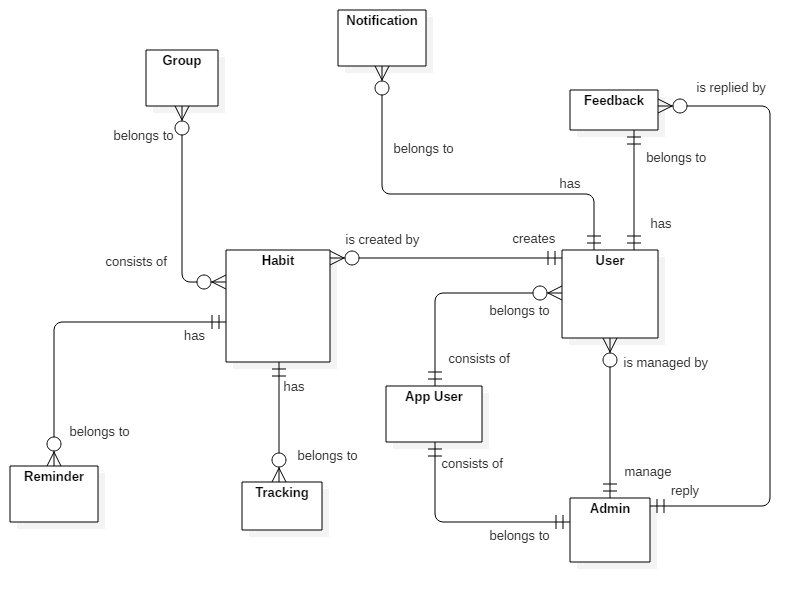
**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Reply feedback. | | |
| **Author** | TuyenNQ | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows the admin to reply feedback if user sent feedback.   **Group:**   * Admin reply feedback when user feedback about this application.   **Triggers:**   * Admin click on “quản lý phản hồi” button. * Admin accepts feedback of users. * Admin click on “trả lời yêu cầu” button   **Preconditions:**   * Admin must login. * User must sent feedback.   **Post Conditions:**   * **Success:** System shows message. * **Fail:** System shows message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin click “quản lý phản hồi” button. | The system shows all feedback of users. | | 2 | Admin click on another user. | The system shows detail of this feedback. | | 3 | Admin click “trả lời” button.  [Alternatives 1] | - The system show text box to admin reply feedback..  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this feedback. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost internet | Shows “mất kết nối! Xin thử lại” message. |   **Relationships:**  N/A  **Business Rules:**   * After reset password of a user. The admin will return to the main page. | | | |

1. Software System Attribute
   1. Usability

* Provide a convenient way to interact with system.
* Screen layout and navigation are clear and easy to use.
* Interface are simple and clear, user can easy to manage habit and group.
* Icons that indicate the actions should be easy to understand and users will not meet any troubles to recognize the feature of application.
  1. Reliability
* The number of sending notification failure is 1 time per 200 notifications.
* The data should be backed up every day.
  1. Availability
* Website available 24/24.
* User connects to internet to login into the application.
* User uses offline after login and connect internet to push data to the server.
  1. Security
* Private: Each role of user has a specific permission to interact with the system.
* Only admin can manage user and feedback of user.
  1. Maintainability
* The code is easy to maintain and upgrade.
* Maintain the whole system every 1 months.
* Accidental problem will be resolved within acceptable time.
  1. Portability
* The website is running on Windows 7 or above.
* The mobile application running on a device with API 23 or above.
  1. Performance
* All requests are handled in maximum time of 5 seconds.

1. Conceptual Diagram



*Conceptual diagram*

***Data Dictionary***

|  |  |
| --- | --- |
| Entity Data dictionary: describe all content of all entities | |
| Entity Name | **Description** |
| App User | Abstract entity describes a user in system |
| User | Contain the user information. |
| Admin | Contain the admin information. |
| Feedback | Contain the feedback information. |
| Habit | Contain the habit information. |
| Notification | Contain the notification information. |
| Tracking | Contain the vehicle information. |
| Reminder | Contain the reminder information. |
| Group | Contain the group information |
|  |  |
|  |  |

1. **Software Design Description**
2. Design Overview

- This document describes the technical and user interface design of MSSC System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.

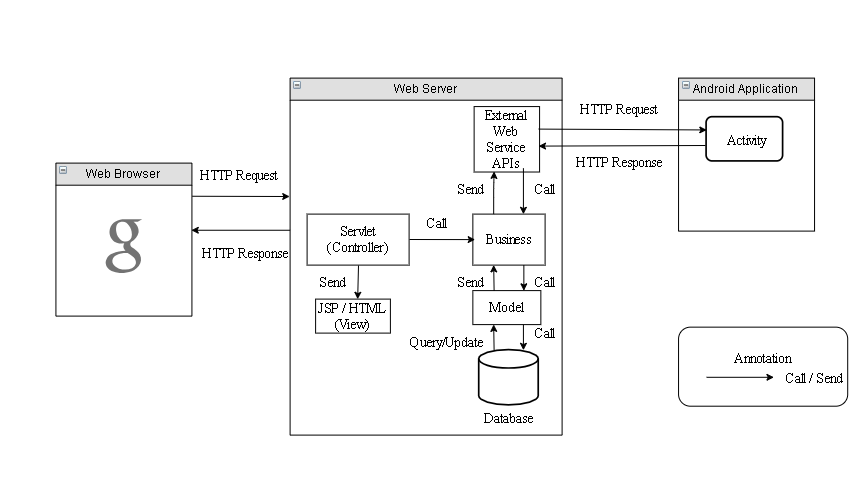
- The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

- The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.

- The database design describes the relationships between entities and details of each entity. - Document overview:

* + Section 2: gives an overall description of the system architecture design.
  + Section 3: gives component diagrams that describe the connection and integration of the system.
  + Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
  + Section 5: describe screens design.
  + Section 6: describe a fully attributed ERD.
  + Section 7: describe algorithms.

1. System architecture design

****

*Figure 9 System architecture design*

* 1. Web Application architecture description

In Web Application, the system is developed under J2EE MVC architecture style. We choose this architecture for Web application because of in scope of the 4 - member team, MVC architecture makes it easier to split the big project into small modules and make it easier to assign each module for members in our team.

This project follows MVC architecture with following components:

* **Servlet (Controller)** is the parts of the application that acts like event handler to handle user interaction. Typically, the controller reads data from a request and calls appropriate Business’s method, then selects view to return to user.
* **JSP/HTML (View)** is the parts of the application that handles the display of the data. The selection of View is under control of Controller.
* **Business** is the parts of the application that do business processing to solve domain problems.
* **Model** is the parts of the application that acts like a data transfer object between the system and database.
* **Web Service** is the parts of the application that acts like an event handler for web and mobile communication via REST method.

1. Component Diagram



Figure 1: Component Web Diagram

| **Components Dictionary: Describes components** | |
| --- | --- |
| Web Apllicaiton | Component to controll the system and process request from mobile. Contains sub component: Model, Web Service, Controller. |
| View | Component that display data |
| Controller | Component of website to handle request from web |
| Service | Component to handle business logic of Website component |
| Repository | Component that communicate with database |
| Data Model | Component that do the interaction between the system and database. Contain sub components: Repository and Service |
| Server Database | Component that store data of system |

Table 1- Components Web Dictionary



Figure 2: Component Android Application Diagram

| **Components Dictionary: Describes components** | |
| --- | --- |
| Android Apllicaiton | Component to controll the system and process request from mobile. Contains sub component: Model, Android Service, Controller. |
| View | Component that display data. |
| Controller | Component of application to handle request from application. |
| Android Service | Component to handle business logic of application component. |
| Model | Component that do the interaction between the system and database. Contain sub components: Repository and Service |

Table 2- Components Application Dictionary

1. DetailedDescription
   1. Class diagram



*Figure 11 Class Diagram*

|  |  |  |
| --- | --- | --- |
| **CLASS DICTIONARY: DESCRIBE CLASS** | | |
| **Class Name** | **Mapping column with Conceptual diagram** | **Description** |
| **App User** | N/A | Contains the app user information |
| **Admin** | Admin | Contains the admin information |
| **User** | User | Contains the user information |
| **Habit** | Habit | Contains the habit information |
| **Reminder** | reminder | Contain the reminder information |
| **Feedback** | Feedback | Contains the feedback information |
| **Tracking** | Tracking | Contains the Tracking information |

* 1. Class Diagram Explanation
     1. *App User*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique identifier of a user |
| username | String | Private | User’s username |
| name | String | Private | User’s name |
| password | String | Private | User’s password |
| fullName | String | Private | User’s full name |
| phone | String | Private | User’s phone |
| email | String | Private | User’s email |
| roleId | Integer | Private | Id of user’s role |
| isActive | Boolean | Private | User’s status |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter | Attribute type | Public | Get value of attribute |
| setter | Void | Public | Set value for attribute |
| checkLogin | Boolean | public | Check user login the system |

* + 1. *User*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique identifier of a user |
| username | String | Private | User’s username |
| password | String | Private | User’s password |
| fullName | String | Private | User’s full name |
| phone | String | Private | User’s phone |
| email | String | Private | User’s email |
| roleId | Integer | Private | Id of user’s role |
| isActive | Boolean | Private | User’s status |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter | Attribute type | Public | Get value of attribute |
| setter | Void | Public | Set value for attribute |

* + 1. *Admin*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique identifier of a admin |
| username | String | Private | Admin’s username |
| name | String | Private | Admin’s name |
| password | String | Private | Admin’s password |
| fullName | String | Private | Admin’s full name |
| phone | String | Private | Admin’s phone |
| email | String | Private | Admin’s email |
| roleId | Integer | Private | Id of admin’s role |
| isActive | Boolean | Private | Admin’s status |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter | Attribute type | Public | Get value of attribute |
| setter | Void | Public | Set value for attribute |

* + 1. *Habit*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique identifier of a habit |
| name | String | Private | Habit’s name |
| createDate | datetime | Private | Habit’s create date |
| endDate | datetime | Private | Habit’s end date |
| startDate | datetime | Private | Habit’s start date |
| finishDate | datetime | Private | Habit’s finish date |
| group | String | Private | Habit’s group |
| description | String | Private | Habit’s description |
| status | String | Private | Habit’s status |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter | Attribute type | Public | Get value of attribute |
| setter | Void | Public | Set value for attribute |
| getHabitList | List | Public | Get all habit of user |

* + 1. *Feedback*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique identifier of user |
| content | String | Private | Feedback’s content |
| **Method** | **Return Type** | **Visibility** | **Description** |

* + 1. *Reminder*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique identifier of the habit |
| notificationName | String | Private | Notification’s notification name |
| time | Datetime | Private | Notification’s time |
| description | String | Private | Notification’s description |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter | Attribute type | Public | Get value of attribute |
| setter | Void | Public | Set value for attribute |

* + 1. *Tracking*

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | Private | Unique identifier of a habit |
| type | Boolean | Private | Tracking’s type |
| thisDay | datetime | Private | Tracking’s this day |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getter | Attribute type | Public | Get value of attribute |
| setter | Void | Public | Set value for attribute |

* 1. Interaction Diagram
     1. *Add new habit*

*Summary: this diagram show process of user add new habit*



Figure - Sequence Diagram for add new habit <User>

* + 1. *Edit habit*

*Summary: this diagram show process of user edits a habit.*



Figure - Sequence Diagram for edit habit <User>

* + 1. *Delete habit*

*Summary: this diagram show process of user deletes a habit*



Figure - Sequence Diagram for add delete habit <User>

1. User Interface Design
   1. Mobile application user interface
      1. *Login*

**

**Field**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Username | Username to login | No | Yes | Textbox | String |  |
| 2 | Password | Password to login | No | Yes | Textbox | String |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Login | Login into system | Validate all required fields | Navigate to page |
| 4 | Login with facebook | Login into system | Validate all required fields | Navigate to page |
| 5 | Login with google | Login into system | Validate all required fields | Navigate to page |
| 6 | Register | Creates account |  | Register page |

* + 1. *Register*



**Field**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Username | Username to register | No | Yes | Textbox | String |  |
| 2 | Email | Email to register | No | Yes | Textbox | String |  |
| 3 | Password | Password to register | No | Yes | Textbox | String |  |
| 4 | Re-enter password | Input password again | No | Yes | Textbox | String |  |

**Buttons/Hyperlinks:**

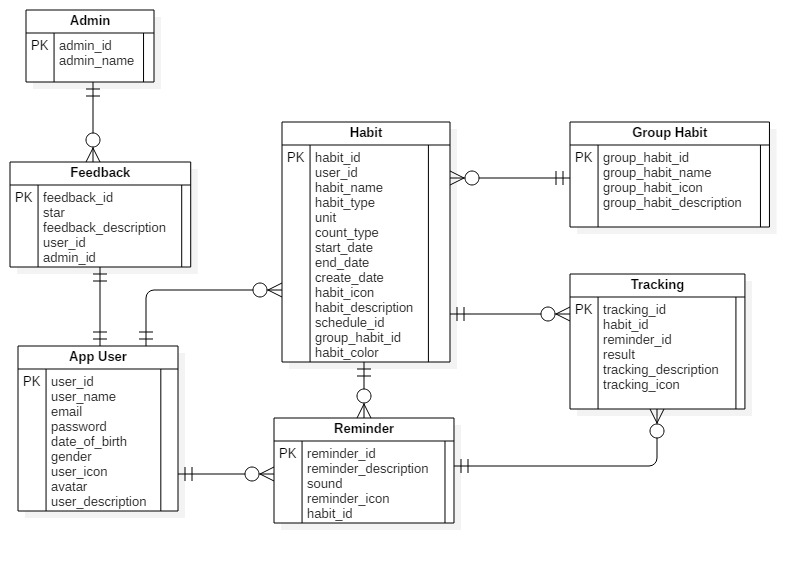
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Register | Create an account | Validate all required fields | Navigate to page |
| 6 | Login with an account | Login into system |  | Login page |

* + 1. *Main menu*

**

* + 1. *Settings*
    2. *Add habit*
    3. *Edit habit*
    4. *Delete habit*

1. Database Design
   1. Entity relationship diagram (ERD)



* 1. Entity dictionary

|  |  |
| --- | --- |
| Entity Data Dictionary: describe content of all entities | |
| Entity name | **Description** |
| App User | Contains the user’s information |
| Admin | Contains the admin’s information |
| Habit | Contains the habit’s information |
| Reminder | Contains the reminder’s information |
| Group | Contains the group’s information |
| Tracking | Contains the tracking’s information |
| Feedback | Contains the feedback’s information |

# **E.** **System Implementation & Test**

1. Introduction
   1. Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of VHT application. It provides in detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

* 1. Test approach
* Goal: Test all features in the whole VHT application based on the core flow.
* Method: black-box testing
* Technique: check list

The testing for this project will consists of Integration System test level. Testing the program which was integrated and as a complete system to ensure that the software requirements have been met.

* Integration testing would be performed by all member of team and approved by team leader.

System testing is focused on assessing the system’s reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

## Database Relationship Diagram



### Physical diagram

### Data dictionary

1. Test plan

The overall purpose of testing is to ensure EMS meets its entire technical, functional and business requirement. The purpose of this document is to describe the overall test plan and strategy for testing the EMS. The approach described in this document provides the framework for all testing related to this application. Individual test cases will be written for each version of the application that is released. This document will also be updated as required for each release.

* 1. Features to be tested
* Guest: login, register.
* User: Add habit, edit habit, delete habit.
  1. Features not to be tested
* Admin: view user, reset password, reply feedback.

1. System Testing Test Case
   1. Test case
      1. *<Guest> Login*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Pre-condition | Test case procedure | Expected output | Result | Test Date |
| AL\_1 | Test guest login into the mobile application successfully | Guest is at “Login” screen. | 1. Guest enters “Luudat01” into “Tên đăng nhập” required field.  2. Guest enters “123456” into “Mật khẩu” required field  3. User clicks “Đăng nhập” button | System shows “Đăng nhập thành công” |  |  |
| AL\_2 | Test guest login with a blank “Tên đăng nhập” text box | Guest is at “Login” screen. | 1. Guest inputs a blank in field “Tên đăng nhập”  2. User clicks “Đăng nhập” button. | System shows error message “Tên đăng nhập không được trống”. |  |  |
| AL\_3 | Test guest login with a blank “Mật khẩu” text box | Guest is at “Login” screen. | 1. Guest inputs a blank in field “Mật khẩu”  2. User clicks “Đăng nhập” button. | System shows error message “Mật khẩu không được trống”. |  |  |
| AL\_4 | Test guest input wrong username or password. | Guest is at “Login” screen. | 1. Guest inputs “Luudat02” in field “Tên đăng nhập”.  2. Guest inputs “123456” in field “Mật khẩu”  2. User clicks “đăng nhập” button. | System shows error message “Đăng nhập thất bại! Tên đăng nhập hoặc mật khẩu không đúng” |  |  |

* + 1. *<Guest> Register*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Pre-condition | Test case procedure | Expected output | Result | Test Date |
| AL\_1 | Test guest register on the mobile application successfully | Guest is at “Đăng ký” screen. | 1. Guest enters “Qtuyen1” into “Tên đăng nhập” required field.  2. Guest enters “0969696969” into “Số điện thoại” required field.  3. Guest enters “tuyen69@gmail.com” into “Email” required field.  4. Guest enters “654321” into “Mật khẩu” required field.  5. Guest enters “654321” into “Nhập lại mật khẩu” required field.  6. User clicks “Đăng ký” button. | System shows “Đăng ký tài khoản thành công” |  |  |
| AL\_2 | Test guest register with a blank “Tên đăng nhập” text box | Guest is at “Đăng ký” screen. | 1. Guest inputs a blank in field “Tên đăng nhập”  2. User clicks “Đăng nhập” button. | System shows error message “tên đăng nhập không được trống”. |  |  |
| AL\_3 | Test guest login with with a blank “Mật khẩu” text box | Guest is at “Đăng ký” screen. | 1. Guest inputs a blank in field “Mật khẩu”  2. User clicks “Đăng nhập” button. | System shows error message “Mật khẩu không được trống”. |  |  |
| AL\_4 | Test guest input wrong email in field “Email”. | Guest is at “Đăng ký” screen. | 1. Guest inputs “abc@.com”  2. User clicks “đăng nhập” button. | System shows error message “Email không đúng” |  |  |

* + 1. *<User> Add habit*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Pre-condition | Test case procedure | Expected output | Result | Test Date |
| AL\_1 | Test user adds a habit successfully | Guest logins at user role.  User is at “add habit” screen. | 1. User enters “Chạy bộ” into “Tên thói quen” required field.  2. User clicks “Xây dựng” into “Xây dựng/Từ bỏ” required field  3. User clicks choose “Hàng ngày” button into “Loại mục tiêu” required field.  4. User chooses “sức khỏe” of the list into “Nhóm thói quen” required field.  5. User tick and choose all day into “Ngày theo dõi trong tuần” required field.  6. User ticks “Ngày bắt đầu”, choose “We 31.10.2018” and tick “Ngày hoàn thành”, choose “Mo 31.12.2018” into “Thời gian thực hiện” required field.  7. User ticks choice red color button into “Màu thói quen” required field.  8. User add “7:00 AM gọi tôi chạy bộ nhé” into “Nhắc nhở” required field.  8. User enters “Sức khỏe là trên hết” intro “Hãy viết động lực cho thói quen này” required field.  9. User clicks “Lưu lại” button. | System shows “Tạo mới thói quen thành công” |  |  |
| AL\_2 | Test user creates a blank “tên thói quen” text box. | Guest logins at user role.  User is at “add habit” screen. | 1. User inputs a blank in field “Tên thói quen”  2. User clicks “Lưu lại” button. | System shows error message “Tên thói quen không được để trống!” |  |  |

* + 1. *<User> Edit habit*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Pre-condition | Test case procedure | Expected output | Result | Test Date |
| AL\_1 | Test user edit a habit successfully | Guest logins at user role.  User is at “Edit habit” screen. | 1. User enters “Đọc sách” into “Tên thói quen” required field.  2. User clicks choose “Xây dựng” button into “Xây dựng/Từ bỏ” required field  3. User clicks choose “Hàng tháng” button into “Loại mục tiêu” required field.  4. User chooses “sức khỏe” of the list into “Nhóm thói quen” required field.  5. User tick and choose all day into “Ngày theo dõi trong tuần” required field.  6. User ticks “Ngày bắt đầu”, choose “We 31.10.2018” and tick “Ngày hoàn thành”, choose “Mo 31.12.2018” into “Thời gian thực hiện” required field.  7. User ticks choice blue color button into “Màu thói quen” required field.  8. User add “7:00 AM gọi tôi chạy bộ nhé” into “Nhắc nhở” required field.  8. User enters “Sức khỏe là trên hết” intro “Hãy viết động lực cho thói quen này” required field.  9. User clicks “Cập nhật” button. | System shows “Cập nhật thói quen thành công” |  |  |
| AL\_2 | Test user edit a habit with a blank “tên thói quen” text box. | Guest logins at user role.  User is at “Edit habit” screen. | 1. User inputs a blank in field “Tên thói quen”  2. User clicks “Cập nhật” button. | System shows error message “Tên thói quen không được để trống!” |  |  |

* + 1. *<User> Delete habit*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Case Description | Pre-condition | Test case procedure | Expected output | Result | Test Date |
| AL\_1 | Test user delete a habit successfully | Guest logins at user role.  User is at “Edit habit” screen. | 1. User click choose “Chạy bộ” habit.  2. Uer click “Xóa” button | System shows “ Xóa thói quen thành công” |  |  |

* 1. テストケース