**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

Contents

[List of tables 6](#_Toc530327814)

[List of Figure 9](#_Toc530327815)

[Definitions, Acronyms, and Abbreviations 10](#_Toc530327816)

[A. Introduction 11](#_Toc530327817)

[1. Project Information 11](#_Toc530327818)

[2. Introduction 11](#_Toc530327819)

[3. Current Situation 11](#_Toc530327820)

[4. Problem Definition 11](#_Toc530327821)

[5. Proposed Solution 11](#_Toc530327822)

[5.1 Feature functions 12](#_Toc530327823)

[5.2 Advantages and disadvantages 12](#_Toc530327824)

[6. Functional Requirements 13](#_Toc530327825)

[7. Role and Responsibility 13](#_Toc530327826)

[B. Software Project Management Plan 14](#_Toc530327827)

[1. Problem Definition 14](#_Toc530327828)

[1.1. Name of this Capstone Project 14](#_Toc530327829)

[1.2. Problem Abstract 14](#_Toc530327830)

[1.3. Project Overview 14](#_Toc530327831)

[2. Project organization 17](#_Toc530327832)

[2.1. Software process model 17](#_Toc530327835)

[2.2. Roles and responsibilities 18](#_Toc530327836)

[2.3. Tools and Techniques 19](#_Toc530327837)

[3. Project Management Plan 20](#_Toc530327838)

[3.1. Software development life cycle 20](#_Toc530327840)

[3.2. Phase Detail 21](#_Toc530327841)

[3.3. All Meeting Minutes 23](#_Toc530327842)

[4. Coding Convention 23](#_Toc530327843)

[C. Software Requirement Specification 25](#_Toc530327844)

[1. User Requirement Specification 25](#_Toc530327845)

[1.1 Guest Requirement 25](#_Toc530327846)

[1.2 User Requirement 25](#_Toc530327847)

[1.3 Admin Requirement 26](#_Toc530327848)

[2. System Requirement Specification 27](#_Toc530327849)

[2.1 External Interface Requirement 27](#_Toc530327850)

[2.2 System Overview Use Case 28](#_Toc530327851)

[2.3 List of Use case 29](#_Toc530327852)

[3. Software System Attribute 92](#_Toc530327853)

[3.1. Usability 92](#_Toc530327854)

[3.2. Reliability 92](#_Toc530327855)

[3.3. Availability 92](#_Toc530327856)

[3.4. Security 92](#_Toc530327857)

[3.5. Maintainability 92](#_Toc530327858)

[3.6. Portability 93](#_Toc530327859)

[3.7. Performance 93](#_Toc530327860)

[4. Conceptual Diagram 93](#_Toc530327861)

[D. Software Design Description 94](#_Toc530327862)

[1. Design Overview 94](#_Toc530327863)

[2. System architecture design 95](#_Toc530327864)

[2.1. Web application architecture description 95](#_Toc530327865)

[2.2. Mobile application architecture description 96](#_Toc530327866)

[3. Component Diagram 98](#_Toc530327867)

[4. Detailed Description 100](#_Toc530327868)

[4.1. Class diagram 100](#_Toc530327869)

[4.2. Class Diagram Explanation 101](#_Toc530327870)

[4.3. Interaction Diagram 104](#_Toc530327871)

[4.4. Activity diagram 106](#_Toc530327872)

[4.5. State machine diagram 107](#_Toc530327873)

[5. User Interface Design 108](#_Toc530327874)

[5.1. Mobile application user interface 108](#_Toc530327875)

[6. Database Design 118](#_Toc530327876)

[6.1. Entity relationship diagram (ERD) 118](#_Toc530327877)

[6.2. Entity dictionary 118](#_Toc530327878)

[7. Algorithms 119](#_Toc530327879)

[7.1. Challenge proposal algorithm 119](#_Toc530327880)

[E. System Implementation & Test 124](#_Toc530327881)

[1. Introduction 124](#_Toc530327882)

[1.1. Overview 124](#_Toc530327883)

[1.2. Test approach 124](#_Toc530327884)

[2. Database Relationship Diagram 125](#_Toc530327885)

[2.1. Physical diagram 125](#_Toc530327888)

[2.2. Data dictionary 126](#_Toc530327889)

[3. Test plan 126](#_Toc530327890)

[3.1. Features to be tested 126](#_Toc530327891)

[3.2. Features not to be tested 126](#_Toc530327892)

[4. System Testing Test Case 127](#_Toc530327893)

[4.1. Test case 127](#_Toc530327894)

[4.2. テストケース 131](#_Toc530327895)

[F1. Software User’s Manual 139](#_Toc530327896)

[1. Installation Guide 139](#_Toc530327897)

[1.1. Hardware Requirement 139](#_Toc530327898)

[1.2. Software Requirement 139](#_Toc530327899)

[2. User Guide 140](#_Toc530327900)

[2.1. Login 140](#_Toc530327901)

[2.2. Register 141](#_Toc530327902)

[2.3. View main menu 143](#_Toc530327903)

[2.4. Add habit 144](#_Toc530327904)

[2.5. Edit habit 146](#_Toc530327905)

[2.6. Delete habit 146](#_Toc530327906)

[2.7. View list reminder 146](#_Toc530327907)

[2.8. Add reminder 146](#_Toc530327908)

[2.9. Edit reminder 146](#_Toc530327909)

[2.10. Delete reminder 146](#_Toc530327910)

[F2. ソフトウェアユーザーマニュアル 147](#_Toc530327911)

[1. インストレーションガイド 147](#_Toc530327912)

[1.1ハードウェア要件 147](#_Toc530327913)

[1.2ソフトウェア要件 147](#_Toc530327914)

[2. ユーザーガイド 148](#_Toc530327915)

[2.1ログイン 148](#_Toc530327916)

[2.2 レジスター 149](#_Toc530327917)

[2.3 メインメニューを表示 151](#_Toc530327918)

[2.4 習慣を追加する 152](#_Toc530327919)

[G. Appendix 154](#_Toc530327920)

[1. SOFTWARE ENGINEERING 9TH EDITION, BY IAN SOMMERVILLE 154](#_Toc530327921)

[2. STARUML DOCUMENTATION FOR ACTIVITY DIAGRAM: 154](#_Toc530327922)

[3. STARUML DOCUMENTATION OF IBM: 155](#_Toc530327923)

[4. UML FOR DRAWING DIAGRAM (2.0 STANDARD) 155](#_Toc530327924)

[5. VUEJS FEATURES 155](#_Toc530327925)

# **List of tables**

[Table 1: Roles and Responsibilities 12](#_Toc530343675)

[Table 2 : Hardware Requirement for Server 15](#_Toc530343676)

[Table 3 : Hardware Requirement for Mobile 15](#_Toc530343677)

[Table 4 : Software requirements 16](#_Toc530343678)

[Table 5 : Roles and responsibilities 18](#_Toc530343679)

[Table 6: Tools List 18](#_Toc530343680)

[Table 7: Technique List 18](#_Toc530343681)

[Table 8: Software Development Life Cycle Detail 19](#_Toc530343682)

[Table 9: Phase 1: Infrastructure 20](#_Toc530343683)

[Table 10: Phase 3: Web service 21](#_Toc530343684)

[Table 11: Phase 4: Mobile app 22](#_Toc530343685)

[Table 12 - <Guest> Login 30](#_Toc530343686)

[Table 13 - <Guest> Register 33](#_Toc530343687)

[Table 14 - <User> Sort Habit 35](#_Toc530343688)

[Table 15 - <User> Feedback and Contact Us 38](#_Toc530343689)

[Table 16 - <User> Set Reminder 40](#_Toc530343690)

[Table 17 - <User> Edit Reminder 42](#_Toc530343691)

[Table 18 - <User> Delete reminder 45](#_Toc530343692)

[Table 19 - <User> Export Statistic File 47](#_Toc530343693)

[Table 20 - <User> Add Habit 49](#_Toc530343694)

[Table 21 - <User> Delete Habit 52](#_Toc530343695)

[Table 22 - <User> Edit Habit 54](#_Toc530343696)

[Table 23 - <User> Set Reminder Habit 57](#_Toc530343697)

[Table 24 - <User> View Statistic Habit 59](#_Toc530343698)

[Table 25 - <User> Add Journal 61](#_Toc530343699)

[Table 26 - <User> Edit Journal 63](#_Toc530343700)

[Table 27 - <User> Delete Journal 65](#_Toc530343701)

[Table 28 - <User> Add Group 67](#_Toc530343702)

[Table 29 - <User> Delete Group 69](#_Toc530343703)

[Table 30 - <User> Edit Group 71](#_Toc530343704)

[Table 31 - <User> Edit Profile 73](#_Toc530343705)

[Table 32 - <User> Suggest Habit 75](#_Toc530343706)

[Table 33 - <User> Filter Habit 77](#_Toc530343707)

[Table 34 - <User> View Statistics 79](#_Toc530343708)

[Table 35 - <User> View Top Habits 80](#_Toc530343709)

[Table 36 - <User> Logout 82](#_Toc530343710)

[Table 37 - <Admin> Reset Password 85](#_Toc530343711)

[Table 38 - <Admin> View Feedback 87](#_Toc530343712)

[Table 39 - <Admin> Reply Feedback 90](#_Toc530343713)

[Table 40 - <Admin> View Statistics 92](#_Toc530343714)

[Table 41 - Conceptual Diagram Dictionary 94](#_Toc530343715)

[Table 42 - Component Web Dictionary 98](#_Toc530343716)

[Table 43 - Component Android Application Dictionary 99](#_Toc530343717)

[Table 44 - Class Diagram Dictionary 101](#_Toc530343718)

[Table 45 - App User 101](#_Toc530343719)

[Table 46 - User 102](#_Toc530343720)

[Table 47 - Admin 102](#_Toc530343721)

[Table 48 - Habit 103](#_Toc530343722)

[Table 49 - Feedback 103](#_Toc530343723)

[Table 50 – Reminder 103](#_Toc530343724)

[Table 51 - Tracking 104](#_Toc530343725)

# **List of Figure**

[Figure 1 : Waterfall model 17](#_Toc530343772)

[Figure 2- System Overview Use Case 27](#_Toc530343773)

[Figure 3- <Guest>Overview Use Case 28](#_Toc530343774)

[Figure 4 - <Guest> Login 28](#_Toc530343775)

[Figure 5 - <Guest> Register 31](#_Toc530343776)

[Figure 6 - <User> Sort Habit 33](#_Toc530343777)

[Figure 7 - <User> Feedback and Contact Us 36](#_Toc530343778)

[Figure 8 - <User> Set Reminder 38](#_Toc530343779)

[Figure 9 - <User> Edit Reminder 40](#_Toc530343780)

[Figure 10 - <User> Delete Reminder 43](#_Toc530343781)

[Figure 11 - <User> Export Statistic File 45](#_Toc530343782)

[Figure 12 - <User> Add Habit 47](#_Toc530343783)

[Figure 13 - <User> Delete Habit 50](#_Toc530343784)

[Figure 14 - <User> Edit Habit 52](#_Toc530343785)

[Figure 15 - <User> Set Reminder Habit 55](#_Toc530343786)

[Figure 16 - <User> View Statistic Habit 57](#_Toc530343787)

[Figure 17 - <User> Add Journal 59](#_Toc530343788)

[Figure 18 - <User> Edit Journal 61](#_Toc530343789)

[Figure 19 - <User> Delete Journal 63](#_Toc530343790)

[Figure 20 - <User> Add Group 65](#_Toc530343791)

[Figure 21 - <User> Delete Group 67](#_Toc530343792)

[Figure 22 - <User> Edit Group 69](#_Toc530343793)

[Figure 23 - <User> Edit Profile 71](#_Toc530343794)

[Figure 24 - <User> Suggest Habit 74](#_Toc530343795)

[Figure 25 - <User> Filter Habit 76](#_Toc530343796)

[Figure 26 - <User> View Statistics 78](#_Toc530343797)

[Figure 27 - <User> View Top Habits 79](#_Toc530343798)

[Figure 28 - <User> Logout 81](#_Toc530343799)

[Figure 29 - <Admin> Overview Use Case 83](#_Toc530343800)

[Figure 30 - <Admin> Reset Password 83](#_Toc530343801)

[Figure 31 - <Admin> View Feedback 86](#_Toc530343802)

[Figure 32 - <Admin> Reply Feedback 88](#_Toc530343803)

[Figure 33 - <Admin> View Statistics 90](#_Toc530343804)

[Figure 34 - Conceptual Diagram 93](#_Toc530343805)

[Figure 35 - Web application architecture description 95](#_Toc530343806)

[Figure 36 - Mobile application architecture description 96](#_Toc530343807)

[Figure 37 - Component Web Diagram 98](#_Toc530343808)

[Figure 38 - Component Android Application Diagram 99](#_Toc530343809)

[Figure 39 - Class Diagram 100](#_Toc530343810)

[Figure 40 - Sequence Diagram for add new habit <User> 104](#_Toc530343811)

[Figure 41 -Sequence Diagram for edit habit <User> 105](#_Toc530343812)

[Figure 42 - Sequence Diagram for delete habit <User> 105](#_Toc530343813)

[Figure 43 - Activity Diagram 106](#_Toc530343814)

[Figure 44 - State Machine Diagram 107](#_Toc530343815)

# 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 administrator:
  + Manage User.
  + Manage Feedbacks
  + View statistics

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:
  + Manage habit.
  + Group habit.
  + Habits statistic.
  + Suggest Habit.
  + Tracking habit.
  + Feedback.
  + View top habits.
  + Manage profile.
  + Manage settings.
  + 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:
  + Data export to CSV files.
  + Interactive notifications.
* 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.
  + Habit follow group.
  + Habit challenges.
  + Statistical data by location.
  + Recommend habits according to user information.
  + Bigdata & AI.

##### ***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 v3.2.1 | Used for implement website and Android Mobile Application. |
| Design Model tool | Star UML v2.5.1 | Used for creating modal and diagrams. |
| DBMS | SQLite & MySQL | 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, NetBean, Android Studio |
| DBMS | MySQL |
| Source control | Github |
| Modeling tool | Star UML v5.0.1 |
| Document tool | Microsoft Word 2016 |

Table 6: Tools List

|  |  |
| --- | --- |
| Technique | Name / version |
| Frontend | HTML5, CSS, JavaScript, WordPress, Sketch. |
| Backend | Java, Spring Boot, Hibernate, PHP, Android, Retrofit. |

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. | 20 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. | 60 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 & Android): 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;

1. 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
* Manage group:
* Add group
* Delete group
* Edit group
* Manage settings:
* Sort habit
* Set reminders
* Export data
* Feedback and contact
* View statistics
* Sugguest habit
* View top habit
* Manage profile
* Log out
  1. **Admin Requirement**

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

* Manage users
* Reset password
* View statistics
* Manage feedbacks
* View feedbacks
* 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**

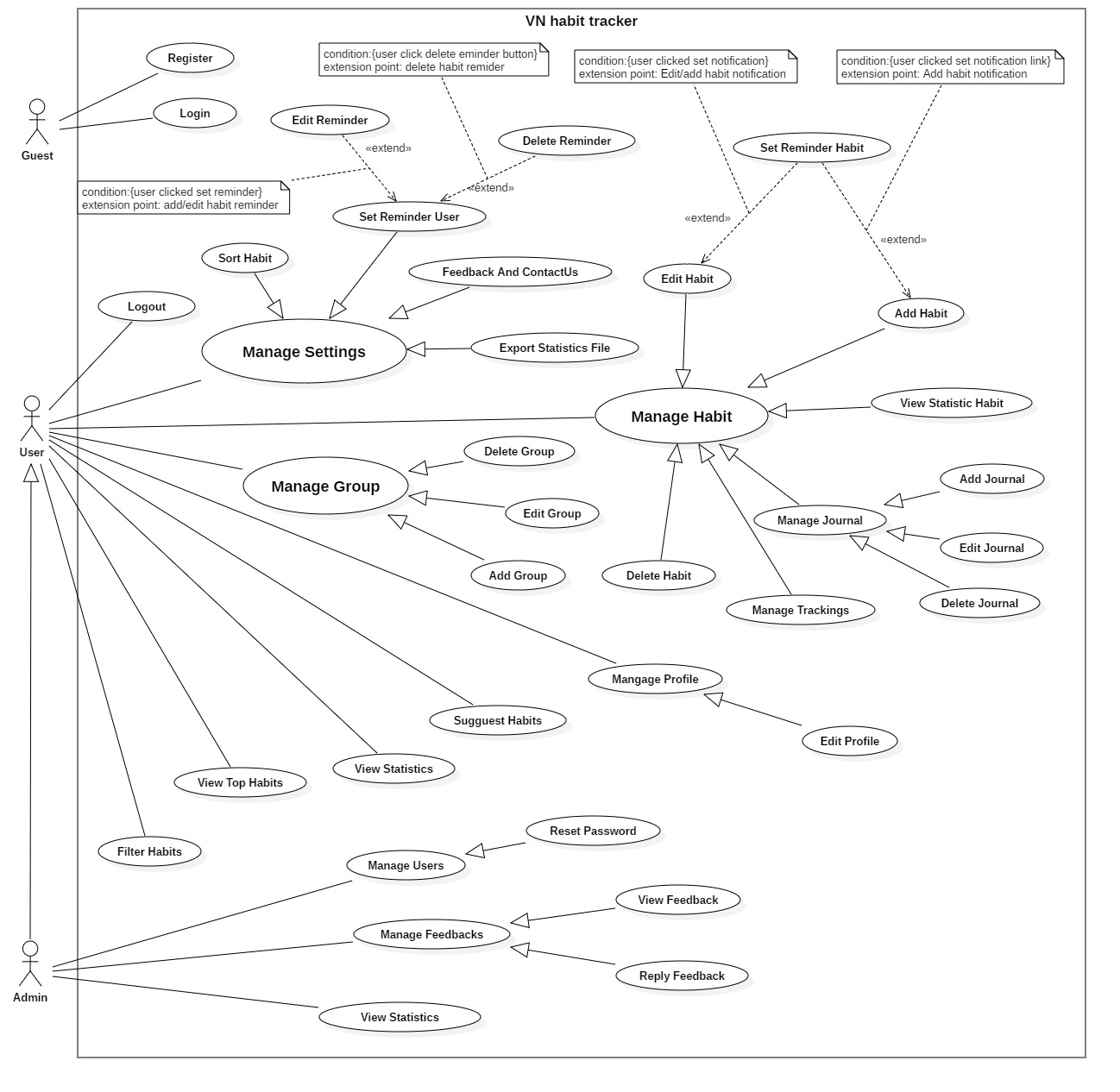


Figure 2- System Overview Use Case

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



Figure 3- <Guest>Overview Use Case

* + - 1. <Guest> Login

**Use Case Diagram**

****

Figure 4 - <Guest> Login

**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] |   **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 | Network problem exception. | Show message “Lỗi kết nối!” |   **Relationships:**  N/A  **Business Rules:**   * After logged in, system will redirect to main page. | | | |

Table 12 - <Guest> Login

* + - 1. <Guest> Register

**Use Case Diagram**

****

Figure 5 - <Guest> Register

**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 registers an account.   **Goal:**   * Guest registers 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 1,2,3,4,5,6] |   **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 | Password is less than 6 character. | Show error message. | | 4 | Email is empty. | Show error message. | | 5 | Email is wrong format. | Show error message. | | 6 | Network problem exception. | Show message “Lỗi kết nối!” |   **Relationships:**  N/A  **Business Rules:**   * After guest registered, system will redirect to login page. | | | |

Table 13 - <Guest> Register

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

**Use Case Diagram**



Figure 6 - <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 - <User> Sort Habit

* + - 1. <User> Feedback and Contact Us

**Use Case Diagram**



Figure 7 - <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 Contact Us | | |
| **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 - <User> Feedback and Contact Us

* + - 1. <User> Set Reminder

**Use Case Diagram**



Figure 8 - <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 - <User> Set Reminder

* + - 1. <User> Edit reminder

**Use Case Diagram**

****

Figure 9 - <User> Edit Reminder

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_04** | | | |
| **Use Case No.** | 04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Reminder | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/10/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user edit a reminder for system.   **Goal:**   * Set reminder for system.   **Triggers:**   * User click on “settings” button. * User click on “Edit 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 “Edit 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 “Update” 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:** Extent from Set Reminder User use case.  **Business Rules:**   * After update reminder success. The user will return to the settings page. * The maximum snooze time is 24 hours. | | | |

Table 17 - <User> Edit Reminder

* + - 1. <User> Delete reminder

**Use Case Diagram**

****

Figure 10 - <User> Delete Reminder

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_04** | | | |
| **Use Case No.** | 04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Reminder | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/10/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user delete a reminder for system.   **Goal:**   * Delete reminder of the system.   **Triggers:**   * User click on “settings” button. * User click on “Edit 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 “Edit reminder” button. | The system shows a form for user delete reminder | | 2 | User click on “Ok” button. | - The system validates the information and shows confirmation message.  [Exception 1] | | 3 | User click “Yes” button. | The system shows “Delete success” message. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “cancel” button. | Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | |  |  |  |   **Relationships:**  Extent from Set Reminder User use case.  **Business Rules:**   * After delete reminder success. The user will return to the settings page. | | | |

Table 18 - <User> Delete reminder

* + - 1. <User> Export statistic file

**Use Case Diagram**

****

Figure 11 - <User> Export Statistic File

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_04** | | | |
| **Use Case No.** | 04 | **Use Case Version** | 1.0 |
| **Use Case Name** | Export Statistic File | | |
| **Author** | TuyenNQ | | |
| **Date** | 27/10/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to export statistic file of this user.   **Goal:**   * Export the user's habits file.   **Triggers:**   * User click on “settings” button. * User click on “Export statistic file” button.   **Preconditions:**   * Guest must login. * User must have at least one habit.   **Post Conditions:**   * **Success:** System shows “Export file success” messages. * **Fail:** System shows “Export file failed” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Export statistis file” button. | The system shows a CSV file. | | 2 | User click on “Ok” button. | - The system validates the information and shows confirmation message.  [Exception 1,2] | | 3 | User click on “Yes” button. | The system shows “export CSV file success” message. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “cancel” button. | Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Failed to export | The system shows “Export to error!” message. | | 2 | Lost internet | The system shows “Internet was interrupt! Export file to error” message. |   **Relationships:**  N/A  **Business Rules:**   * After Export file success. The user will return to the settings page. * User to checks result. | | | |

Table 19 - <User> Export Statistic File

* + - 1. <User> Add Habit

**Use Case Diagram**



Figure 12 - <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** | 30/10/2018 | **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 “Yes” 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 20 - <User> Add Habit

* + - 1. <User> Delete Habit

**Use Case Diagram**



Figure 13 - <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** | 30/10/2018 | **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 21 - <User> Delete Habit

* + - 1. <User> Edit Habit

**Use Case Diagram**



Figure 14 - <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 22 - <User> Edit Habit

* + - 1. <User> Set Reminder Habit

**Use Case Diagram**

****

Figure 15 - <User> Set Reminder Habit

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_09** | | | |
| **Use Case No.** | 09 | **Use Case Version** | 1.0 |
| **Use Case Name** | Set Reminder Habit | | |
| **Author** | ThangNH | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to set the notification.   **Goal:**   * Set reminder for a habit.   **Triggers:**   * User click on “Set reminder habit” 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:**  Extent from Edit Habit, Add Habit use case.  **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 23 - <User> Set Reminder Habit

* + - 1. <User> View Statistic Habit

**Use Case Diagram**

**

Figure 16 - <User> View Statistic Habit

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Statistic Habit | | |
| **Author** |  | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user view statistic detail of the habit.   **Goal:**   * User view statistic detail of the habit.   **Triggers:**   * User click on added habits.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** Statistic detail will be displayed. * **Fail:** Statistic detail will not be displayed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on added habits | System shows statistic detail of habit.  + View by week.  + View by month.  + View by year. | | 2 | User click on “Tuần” button. | - The system will show the information of the habit belong to week.  [Exception 1] | | 3 | User click on “Tháng” button. | - The system will show the information of the habit belong to month.  [Exception 1] | | 4 | User click on “Năm” button. | - The system will show the information of the habit belong to year.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “back” icon. | Back to main page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | System fail to show statistic detail. | System shows “Error!” message |   **Relationships:**  N/A  **Business Rules:**   * After view statistic detail successfully. The user will return to the m. * User can view another added habit. | | | |

Table 24 - <User> View Statistic Habit

* + - 1. <User> Add Journal

**Use Case Diagram**

**

Figure 17 - <User> Add Journal

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Add Journal | | |
| **Author** |  | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user adds new journal.   **Goal:**   * User add new journal.   **Triggers:**   * User click on added habits. * User click on journal icon   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** Journal will be displayed. * **Fail:** Journal will not be displayed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on journal icon | System shows statistic detail of habit. | | 2 | User click on “Ghi chú” button. | - The system will show the text box to write in.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “back” icon. | Back to main page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | System fail to add a new journal. | System shows “Error!” message |   **Relationships:**  Extent from Manage Journal use case.  **Business Rules:**   * After add a new journal successfully. The user will return to the main page. * User can add another journal. | | | |

Table 25 - <User> Add Journal

* + - 1. <User> Edit Journal

**Use Case Diagram**

****

Figure 18 - <User> Edit Journal

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Journal | | |
| **Author** |  | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user edit a journal.   **Goal:**   * User edit a journal.   **Triggers:**   * User click on added habits. * User click on journal icon   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** Journal will be displayed. * **Fail:** Journal will not be displayed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on journal icon | System shows statistic detail of habit. | | 2 | User click on “Ghi chú” button. | - The system will show all the journal that added before.  [Exception 1] | | 3 | User click on “Edit” icon | -Text can be edited |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “back” icon. | Back to main page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | System fail to edit journal. | System shows “Error!” message |   **Relationships:**  Extent from Manage Journal use case.  **Business Rules:**   * After edit a journal successfully. The user will return to the main page. * User can edit another journal. | | | |

Table 26 - <User> Edit Journal

* + - 1. <User> Delete Journal

**Use Case Diagram**

****

Figure 19 - <User> Delete Journal

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Delete Journal | | |
| **Author** |  | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user deletes a journal.   **Goal:**   * User delete a journal.   **Triggers:**   * User clicks on added habits. * User click on journal icon   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** Journal will be displayed. * **Fail:** Journal will not be displayed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on journal icon | System shows statistic detail of habit. | | 2 | User click on “Ghi chú” button. | - The system will show all the journal that added before.  [Exception 1] | | 3 | User click on “Delete” icon | - That journal will be deleted |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “back” icon. | Back to main page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | System fail to delete a journal. | System shows “Error!” message |   **Relationships:**  N/A  **Business Rules:**   * After delete a journal successfully. The user will return to the main page. * User can delete another journal. | | | |

Table 27 - <User> Delete Journal

* + - 1. <User> Add Group

**Use Case Diagram**



Figure 20 - <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:**  N/A  **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 28 - <User> Add Group

* + - 1. <User> Delete Group

**Use Case Diagram**



Figure 21 - <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 29 - <User> Delete Group

* + - 1. <User> Edit Group

**Use Case Diagram**



Figure 22 - <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 30 - <User> Edit Group

* + - 1. <User> Edit Profile

**Use Case Diagram**



Figure 23 - <User> Edit Profile

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Edit Profile | | |
| **Author** | DatLT | | |
| **Date** | 07/10/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user to edit profile.   **Group:**   * Edit profile of this user.   **Triggers:**   * User click on “profile” button. * User click on “edit profile” button.   **Preconditions:**   * Guest must login. * User must have information.   **Post Conditions:**   * **Success:** System shows “Successful to edit profile!” message. * **Fail:** System shows “Failed to edit profile!” message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Edit profile” button. | The system request information in form:   * Name: text * Age: number * Phone: number | | 2 | User click on “OK” button | - The system validates the request and shows the added information.  [Exception 1, 2] | | 3 | User click on “Save” button | - That information about the user will be edited. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Lost internet | Shows “update fail!” message. | | 2 | Name is empty | Shows “Your name not empty!” message. |   **Relationships:**  N/A  **Business Rules:**   * Name user not null. * The information of User is sent to the server. * After Edit profile success. The user will return to the profile page. | | | |

Table 31 - <User> Edit Profile

* + - 1. <User> Suggest Habit

**Use Case Diagram**



Figure 24 - <User> Suggest Habit

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Suggest Habit | | |
| **Author** | DatLT | | |
| **Date** | 13/11/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user will suggest from the application.   **Group:**   * Suggest for the user.   **Triggers:**   * User click on “Add habit” button. * User click on “Suggest habit” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows success message. * **Fail:** System shows error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Suggest habit” button. | The system request information in form: list the habits of each group. | | 2 | User click on one habit | The system auto fill information about this habit on add habit page. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response |   **Relationships:**  N/A  **Business Rules:**   * The information about the habit of the user is sent to the server. * After suggest habit success. The user will return to the add habit page. | | | |

Table 32 - <User> Suggest Habit

* + - 1. <User> Filter Habits

**Use Case Diagram**



Figure 25 - <User> Filter Habit

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | Filter Habit | | |
| **Author** | DatLT | | |
| **Date** | 13/11/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user want to search habit.   **Group:**   * Filter habit.   **Triggers:**   * User click on “Filter habit” button.   **Preconditions:**   * Guest must login. * User must have at least one habit.   **Post Conditions:**   * **Success:** System shows success message. * **Fail:** System shows error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Filter habit” button. | The system request information in form:   * Goal period: all, day, week, month, year. * Goal type: all, build, quit. * Goal tracking: all, tracked, untracked. | | 2 | User click on “Apply” button. | - The system validates the request and shows the added information. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Close this form. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | |  |  |  |   **Relationships:**  N/A  **Business Rules:**   * After filter habit success. The user will return to the main page. | | | |

Table 33 - <User> Filter Habit

* + - 1. <User> View Statistics

**Use Case Diagram**



Figure 26 - <User> View Statistics

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Statistics | | |
| **Author** | DatLT | | |
| **Date** | 13/11/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user want to view all statistics of the habit.   **Group:**   * View statistic of the habit.   **Triggers:**   * User click on “statistic” button.   **Preconditions:**   * Guest must login. * User must have at least one habit and track at least one.   **Post Conditions:**   * **Success:** System shows success message. * **Fail:** System shows error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “statistic” button. | Display information in chart about all habits follow week, month and year:   * Total of habit: text * Successful habits: text * List habit: list |   **Relationships:**  N/A  **Business Rules:**   * The information is loaded from the system. | | | |

Table 34 - <User> View Statistics

* + - 1. <User> View Top Habits

**Use Case Diagram**



Figure 27 - <User> View Top Habits

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Top Habits | | |
| **Author** | DatLT | | |
| **Date** | 13/11/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the users view habits that most user community.   **Group:**   * View top habits from community.   **Triggers:**   * User click on “top habits” button.   **Preconditions:**   * Guest must login.   **Post Conditions:**   * **Success:** System shows success message. * **Fail:** System shows error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “top habit” button. | The system request information in list habits that community use most. |   **Relationships:**  N/A  **Business Rules:** N/A | | | |

Table 35 - <User> View Top Habits

* + - 1. <User> Logout

**Use Case Diagram**



Figure 28 - <User> Logout

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | Logout | | |
| **Author** | DatLT | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows the user want to logout from application.   **Goal:**   * User logout the system.   **Triggers:**   * User clicks “Cài đặt” button. * User clicks “Đăng xuất” button.   **Preconditions:**   * Guest login success.   **Post Conditions:**   * **Success:** System shows success message. * **Fail:** System shows error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User clicks “Đăng xuất” button. | System shows dialog. | | 2 | User clicks “Yes” button. | User will logout the system.  [Exception 1,2,3] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click “cancel” button. | Close this form. |   **Relationships:**  N/A  **Business Rules:**   * After logout from the system, system will return login page. | | | |

Table 36 - <User> Logout

* + 1. <Admin> Overview Use Case



Figure 29 - <Admin> Overview Use Case

* + - 1. <Admin> Reset Password

**Use Case Diagram**

**

Figure 30 - <Admin> Reset Password

**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 send 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. | | | |

Table 37 - <Admin> Reset Password

* + - 1. <Admin> View Feedback

**Use Case Diagram**

**

Figure 31 - <Admin> View Feedback

**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 clicks into a feedback.  [Alternative 1] | - The system shows 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. | | | |

Table 38 - <Admin> View Feedback

* + - 1. <Admin> Reply Feedback

**Use Case Diagram**

**

Figure 32 - <Admin> Reply Feedback

**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 clicks on another user. | The system shows detail of this feedback. | | 3 | Admin click “trả lời” button.  [Alternatives 1] | - The system shows text box to admin reply feedback  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “Cancel” button. | - Closed 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. | | | |

Table 39 - <Admin> Reply Feedback

* + - 1. <Admin> View Statistics

**Use Case Diagram**



Figure 33 - <Admin> View Statistics

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | View Statistics | | |
| **Author** | DatLT | | |
| **Date** | 27/09/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows the Admin view statistic detail of user’s habits.   **Goal:**   * View statistics detail of user’s habits.   **Triggers:**   * Admin clicks on “Statistics” button.   **Preconditions:**   * Admin login with role admin.   **Post Conditions:**   * **Success:** Statistic detail will be displayed. * **Fail:** Statistic detail will not be displayed.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click on “Statistics” button. | System shows statistics detail of user’s habits.  + View by week.  + View by month.  + View by year. | | 2 | User click on “Tuần” button. | - The system will show the information of the habit belong to week.  [Exception 1] | | 3 | User click on “Tháng” button. | - The system will show the information of the habit belong to month.  [Exception 1] | | 4 | User click on “Năm” button. | - The system will show the information of the habit belong to year.  [Exception 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click on “back” icon. | Closed this page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | System fail to show statistics detail. | System shows “Error!” message. | | 2 | Lost internet | System shows “Error connection!” message. |   **Relationships:**  N/A  **Business Rules:**   * After view statistic of user’s habits successfully. The admin will return to the main page. * Admin can view all user’s habits. | | | |

Table 40 - <Admin> View Statistics

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.
* The system usability is easy to use that will need less than 2 days of training for company staffs to use the system.
  1. **Reliability**
* System notification success rate is less than 2 failed notifications per 1000 sent.
  1. **Availability**
* N/A
  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.
* All input data are validated before saving to database.
  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**

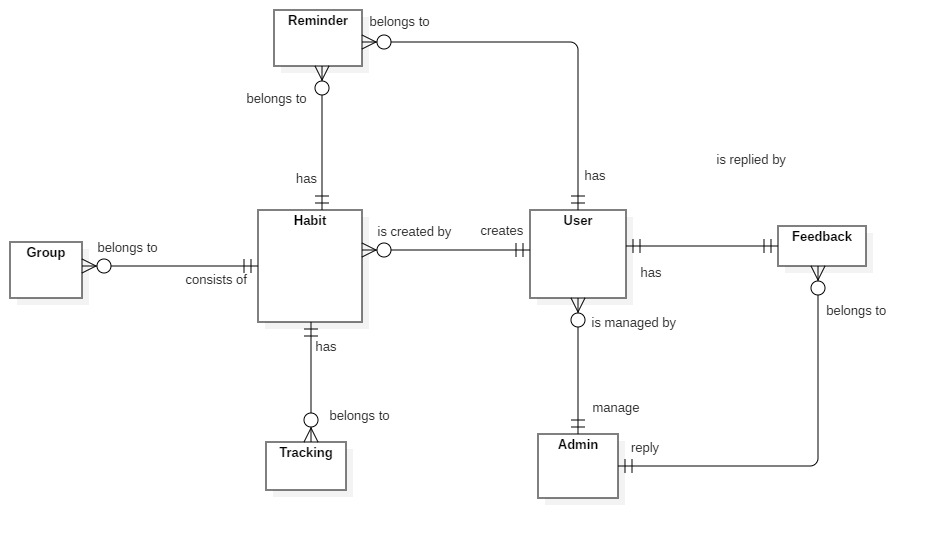


Figure 34 - Conceptual Diagram

***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 |

Table 41 - Conceptual Diagram Dictionary

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**
   1. **Web application architecture description**

****

Figure 35 - 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. **Mobile application architecture description**

****

Figure 36 - Mobile application architecture description

Numerous architectural patterns appeared, but MVP (Model View Presenter) can fulfill the complete requirements of Android developers.

MVP is one of the patterns, which Android community prefers at this time.

In Android, the application should be easily extensible and maintainable. Therefore, in order to maintain the level, it is important to define separated layers well. And, MVP makes things easier for developers and it makes the views independent of the data source.

* **The Model**
* It represents the layer, which holds the business logic as well as controls how data is created, stored and modified. In Android, it is a data access layer, for example, database API or Remote server API.
* The Model consists of components that are responsible for functionalities like for generating, storing, exposing and fetching the data.
* All these functionalities usually perform in the background thread. Because, these functionalities could be time-consuming, and they can potentially block the main thread UI.
* **The View**
* It is a passive interface, which displays data, and the routes user actions to Presenter.
* In Android, the View is represented by Activity, Fragment or View.
* The View component contains a visual part of the application.
* The View contains the UI and it does not contain any logic or knowledge of the displayed data.
* **The Presenter**
* The Presenter is in between Model and the View. And, it triggers the business logic, and lets to know ‘the View’ when to update.
* It recovers data received from the Model and shows it in the View.
* It interacts with the Model, then fetches and transforms the data from the Model to update the view.

1. **Component Diagram**



Figure 37 - 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 42 - Component Web Dictionary



Figure 38 - 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 43 - Component Android Application Dictionary

1. **Detailed Description**
   1. **Class diagram**



Figure 39 - 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 |

Table 44 - Class Diagram Dictionary

* 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 |

Table 45 - App User

* + 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 |

Table 46 - User

* + 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 |

Table 47 - Admin

* + 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 |

Table 48 - Habit

* + 1. Feedback

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

Table 49 - Feedback

* + 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 |

Table 50 – Reminder

* + 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 |

Table 51 - Tracking

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

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



Figure 40 - Sequence Diagram for add new habit <User>

* + 1. Edit habit

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



Figure 41 -Sequence Diagram for edit habit <User>

* + 1. Delete habit

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



Figure 42 - Sequence Diagram for delete habit <User>

* 1. **Activity diagram**



Figure 43 - Activity Diagram

* 1. **State machine diagram**

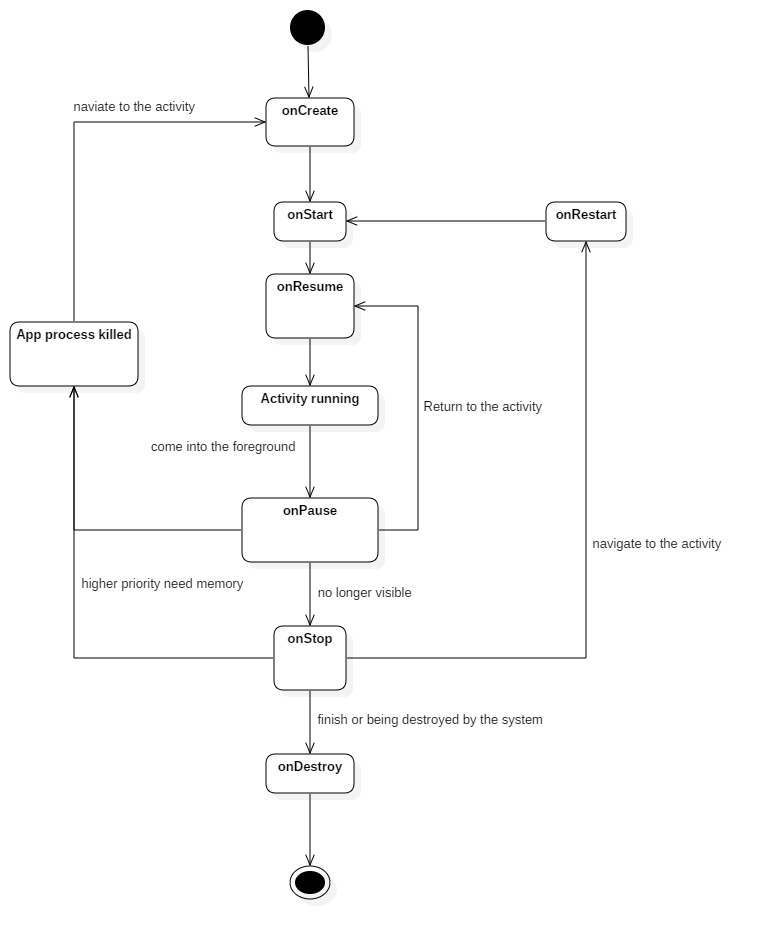


Figure 44 - State Machine Diagram

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 |  |

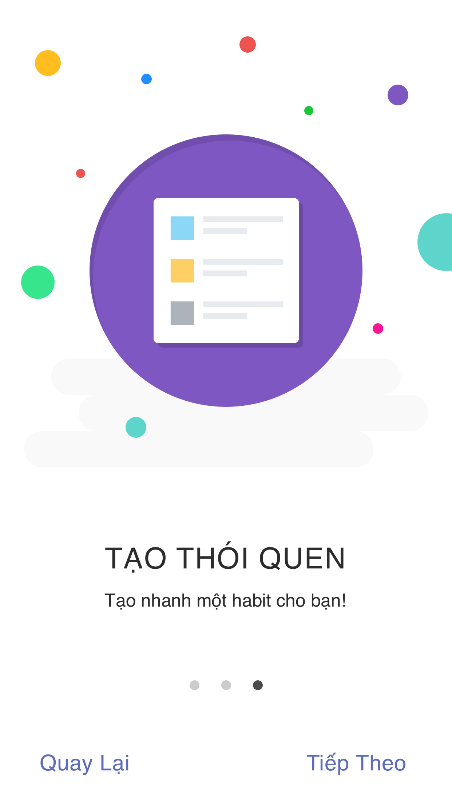
**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. View Introduction

**

**

**

* + 1. View main menu

**

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Settings | Settings of user | N/A | Settings page |
| 2 | Filter | Arrange habits by type | N/A | Filter page |
| 3 | Add a habit | Create new build or quit a habit | N/A | Add habit page |
| 4 | Statitis | Show statitis | N/A | Statitis page |
| 5 | Schedule |  | N/A |  |
| 6 | List habit | Show all habit | N/A | Main menu page |
| 7 |  |  | N/A |  |

* + 1. Setting
    2. Add habit

**

**Field**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Tên thói quen | Name habit want to create | No | Yes | Textbox | String |  |
| 7 | Ngày theo dõi trong tuần | Choice day want to tracking | No | Yes | Checkbox | String |  |
| 8 | Thời gian hiện thực | Start time to tracking | No | Yes | Checkbox | String |  |
| 9 | Màu habit | Choice a color for the habit | No | Yes | Checkbox | String |  |
| 11 | Động lực Habit | Write motivation for the habit | No | Yes | Textbox | String |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Suggest | Suggest a habit for user | N/A | Auto fill into “tên thói quen” textbox. |
| 3 | Build/Quit | Build/Quit status | N/A | Change build or quit status |
| 4 | Habit type | Habit type status | N/A | Change habit type status |
| 5 | Tracking | Tracking status | N/A | Change tracking status |
| 6 | Group | Group of habit | N/A | Group of habit page |
| 10 | Reminder | Create a reminder for the habit | N/A | Reminder page |
| 12 | Cancel | Cancel | N/A | Close add habit’s form. |
| 13 | Create | Create a new habit | Validate all required info | Go to the main page. |

* + 1. Edit habit

**

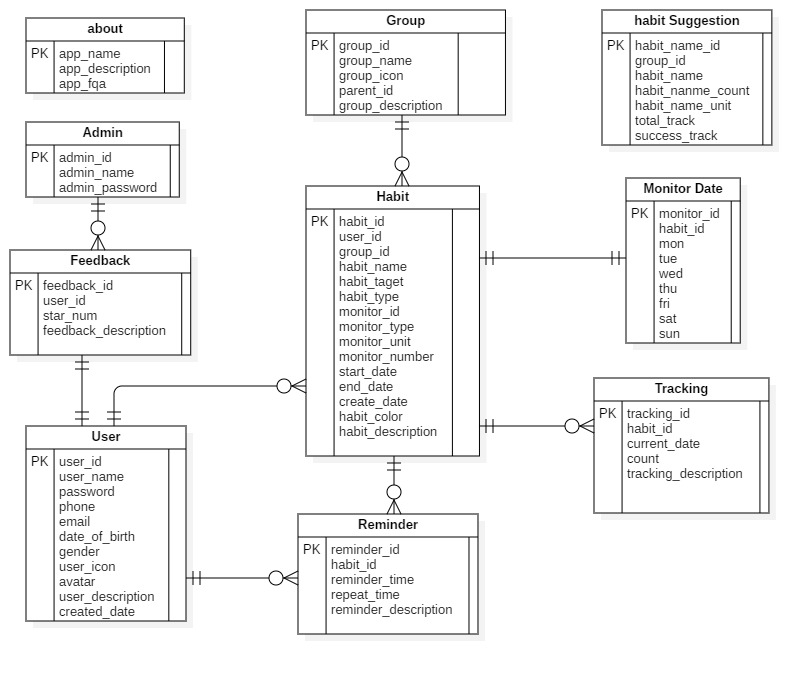
**Field**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** | **Length** |
| 1 | Tên thói quen | Name habit want to create | No | Yes | Textbox | String |  |
| 7 | Ngày theo dõi trong tuần | Choice day want to tracking | No | Yes | Checkbox | String |  |
| 8 | Thời gian hiện thực | Start time to tracking | No | Yes | Checkbox | String |  |
| 9 | Màu habit | Choice a color for the habit | No | Yes | Checkbox | String |  |
| 11 | Động lực Habit | Write motivation for the habit | No | Yes | Textbox | String |  |

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Suggest | Suggest a habit for user | N/A | Auto fill into “tên thói quen” textbox. |
| 3 | Build/Quit | Build/Quit status | N/A | Change build or quit status |
| 4 | Habit type | Habit type status | N/A | Change habit type status |
| 5 | Tracking | Tracking status | N/A | Change tracking status |
| 6 | Group | Group of habit | N/A | Group of habit page |
| 10 | Reminder | Create a reminder for the habit | N/A | Reminder page |
| 12 | Cancel | Cancel | N/A | Close add habit’s form. |
| 13 | Update | Update habit | Validate all required info | Go to the main page. |

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 |
| Monitor Date | Contains the monitor date’s information |
| About | Contains the about’s information |
| Habit Suggestion | Contains the habit suggestion’s information |

## Algorithms

* 1. **Challenge proposal algorithm**
     1. Definition

This algorithm is intended to suggest habits that are commonly used by the community and that are comparable to the current user.

* + 1. Define problem

Helps newcomers or those who have long participated in good habits or give up bad habits to improve themselves.

* + 1. Solution

Supposed:

- c1 is time that habit is tracked and < 30

- c2 is time that habit is tracked between 30 and 60

- c3 is time that habit is tracked and > 60

----------------------------------------------

Based on a habit:

Supposed:

a: The total of successful tracking

b: The total times of tracking

Supposed:

p: The habit is done successfully.

p = a / b with a / b >= 0.8

f: The habit is unsuccessful

f = a / b with a / b < 0.8

Supposed

h: is the difficulty of the habit

k1: is successful habits

k2: is unsuccessful habits

If:

h >= 0.8 -> Easy habit

0.5 < h < 0.8 -> Average habit

h < 0.5 -> Difficult habit

The system will then recommend based on

For c1 -> easy habit

For c2 -> average habit

For c3 -> difficult habit

Let x (n) be the used habit of many people, \* (x1 is the most users use it, x2 is the second users use it after x1, ….)

Then:

User A will be suggested for habit number 4

* + 1. Complexity

In total, the complexity of this algorithm is **O ().**

* + 1. Example

For example, 10 users use a habit "gym"

User1: Habit "Gym" has a daily type

Start date is 10/10/2018

Current date is 20/10/2018

Day of implementation is 10 days, of which 8 days is successfully completed, 2 days is failure.

* a / b = 0.8 => implementation of the habit is (Pass)

User2: Habit "gym" type yes / no

Start date is 1/10/2018

Current date 15/10/2018

The implementation date is 15 days, of which 9 days are yes, 6 days are no.

a / b = 0.6 => implementation of the is (Fail)

Assumptions:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | User 1 | User 2 | User 3 | User 4 | User 5 | User 6 | User 7 | User 8 | User 9 | User 10 |
|  | Pass | Fail | Pass | Pass | Fail | Pass | Pass | Pass | Pass | Pass |
| Habit | gym | gym | gym | gym | gym | gym | gym | gym | gym | gym |

8: Pass

2: Fail

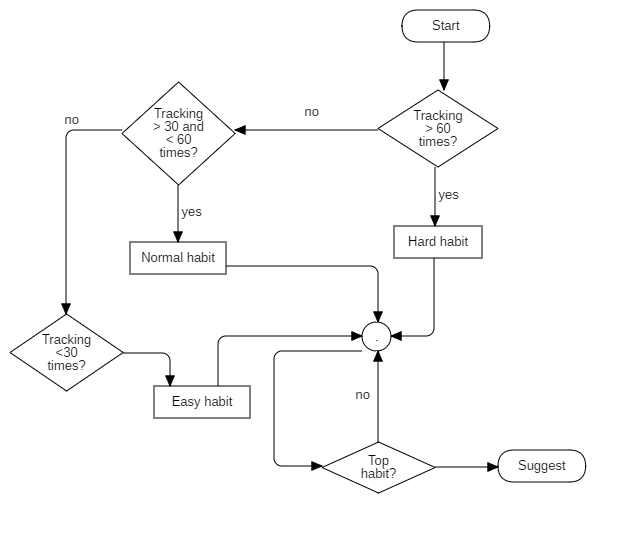
* h >= 0.8 -> easy habit

For the user A with the participate time is 2 months = > apply d1

Assume the following top habit:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Habit  (top1) | Habit  (top2) | Habit  (top3) | Habit (top4) | Habit  (top5) |
| Times of using | 553 | 500 | 465 | 326 | 254 |
| Difficulty of habit | Difficult | Average | Difficult | Easy | Difficult |

* + 1. Flowchart



# **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.

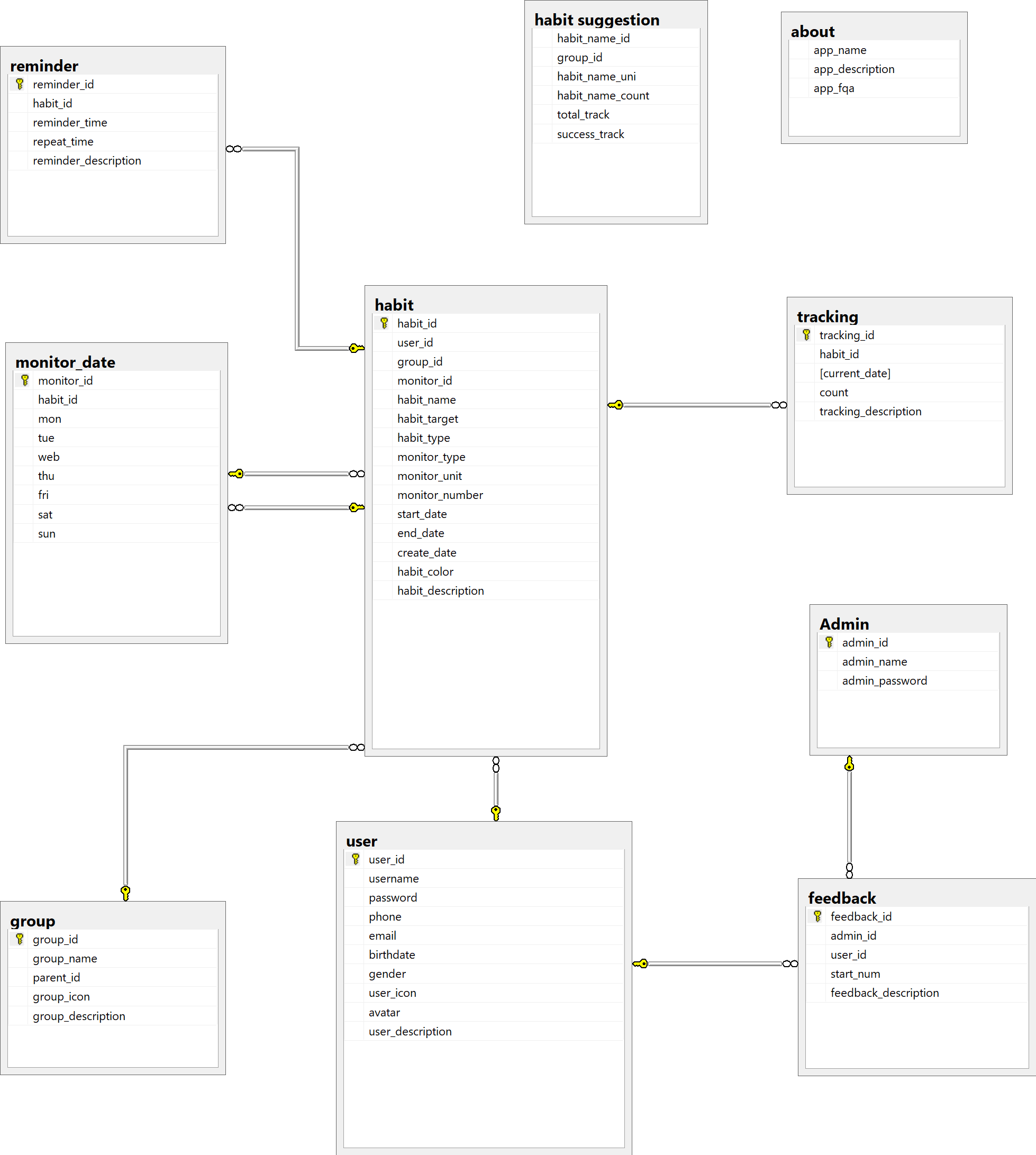
* Integration testing would be performed by all member of team and approved by team leader.

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**

|  |  |
| --- | --- |
| **DATA DICTIONARY: DESCRIBE CONTENT OF ALL TABLES** | |
| **Table name** | **Description** |
| User | Contains the user information |
| Admin | Contains the admin information |
| Habit | Contains the habit information |
| Feedback | Contains the feedback information |
| Group | Contains the group information |
| Monitor date | Contains the monitor date information |
| Tracking | Contains the tracking information |
| Reminder | Contains the reminder information |
| Habit suggestion | Contains the habit suggestion information |
| About | Contains the about information |

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: 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**
     1. <ゲスト>ログイン

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **テストケースの説明** | **事前条件** | **テストケースの手順** | **予想出力** | **結果** | **実行日** |
| AL\_1 | ゲストのモバイルアプリケーションへのログインを正常にテストする | ゲストは “Login screen” にあります | 1. ゲストは必須フィールド "Tên đăng nhập"に "Luudat 01"と入力します。  2. ゲストは必須フィールド "Mật khẩu"に "123456"を入力します。  3. ユーザーが“Đăng nhập”ボタンをクリックする | システムは "Đăng nhập thành công" |  |  |
| AL\_2 | 空白のテキストボックスを使用したゲストログイン | ゲストは「Login screen」にあります。 | 1. ゲストは「Tên đăng nhập」フィールドに空白を入力します。  2. ユーザーが「Đăng nhập」ボタンをクリックします。 | システムにエラーメッセージ「Tên đăng nhập không được trống」が表示されます。 |  |  |
| AL\_3 | 空白のテキストボックスを使用したゲストログイン | ゲストは「Login screen」にあります。 | 1.ゲストは「Mật khẩu」フィールドに空白を入力します。  2. ユーザーが「Mật khẩu」ボタンをクリックします。 | システムにエラーメッセージ「Mật khẩu không được trống」が表示されます。 |  |  |
| AL\_4 | テストゲストのユーザー名またはパスワードが間違っています。 | ゲストは「Login screen」にあります。 | 1. ゲストは必須フィールド "Tên đăng nhập"に "Luudat 01"と入力します。  2. ゲストは必須フィールド "Mật khẩu"に "123456"を入力します。  3. ユーザーが“Đăng nhập”ボタンをクリックする | システムは「Đăng nhập thất bại! Tên đăng nhập hoặc mật khẩu không đúng」 |  |  |

* + 1. <ゲスト>登録

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **テストケースの説明** | **事前条件** | **テストケースの手順** | **予想出力** | **結果** | **実行日** |
| AL\_1 | モバイルアプリケーションのゲスト登録を正常にテストする | ゲストは "Đăng ký"画面にあります。 | 1.「Qtuyen1」を「Tên đăng nhập」必須フィールドに入力します。  2.ゲストは、 "Số điện thoại"必須フィールドに "0969696969"を入力します。  3.ゲストは「Email」必須フィールドに「tuyen69@gmail.com」と入力します。  4.ゲストは、 "Mật khẩu"必須フィールドに "654321"を入力します。  5.ゲストは、 "Nhập lại mật khẩu"必須フィールドに "654321"を入力します。  6.ユーザーが「Đăng ký」をクリックします。 | システムは "Đăng ký tài khoản thành công" |  |  |
| AL\_2 | 空白のテキストボックスでゲスト登録をテストする | ゲストは "Đăngký"画面にあります。 | 1.ゲストは「Tên đăng nhập」フィールドに空白を入力します。  2.ユーザーが「Đăng nhập」ボタンをクリックします。 | エラーメッセージ "tên đăng nhập không được trống"が表示されます。 |  |  |
| AL\_3 | 空白のテキストボックスでゲスト登録をテストする | ゲストは "Đăngký"画面にあります。 | 1.ゲストは「Mật khẩu」フィールドに空白を入力します。  2. ユーザーが「Đăng nhập」ボタンをクリックします。 | エラーメッセージ " Mật khẩu không được trống "が表示されます。 |  |  |
| AL\_4 | 空白のテキストボックスでゲスト登録をテストする | ゲストは "Đăngký"画面にあります。 | 1. ゲストは "abc @ .com"に入力します  2. ユーザーが「Đăng nhập」ボタンをクリックします。 | エラーメッセージ " Email không đúng "が表示されます。 |  |  |

* + 1. <ユーザー>習慣を追加する

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **テストケースの説明** | **事前条件** | **テストケースの手順** | **予想出力** | **結果** | **実行日** |
| AL\_1 | Test user adds a habit successfully | テストユーザーが習慣を正常に追加する  ユーザーは「add habit」画面にあります。 | 1. ユーザは、 "Tên thói quen"必須フィールドに "Chạy bộ"を入力する。  2. ユーザーが "Xây dựng"を "Xây dựng / Từ bỏ"の欄にクリックする  3. ユーザーは "Loại mục tiêu"の必須フィールドから "Hàng ngày"ボタンを選択します。  4. ユーザーは、リストの「sức khỏe」を「Nhóm thói quen」必須フィールドに選択します。  5. ユーザーのチェックボックスをオンにして、必要なフィールドにすべての曜日を選択します。  6. ユーザーは「Ngày bắt đầu」を選択し、「We 31.10.2018」を選択して「Ngày hoàn thành」にチェックを入れ、「Thờ igian thực hiện」の必須フィールドに「Mo 31.12.2018」を選択します。  7. ユーザーは選択した赤色のボタンを「Màu thói quen」の必須フィールドに合わせます。  8. ユーザーは、 "Nhắc nhở"必須フィールドに "7:00 AM gọi tôi chạy bộ nhé"を追加します。  8. ユーザーは、「必須フィールド」に「Sức khỏe là trên hết」と入力してください。  9. ユーザーは "Lưu lại"ボタンをクリックします。 | システムは "Tạo mới thói quen thành công" |  |  |
| AL\_2 | テストユーザーは、空白の "テンキー"テキストボックスを作成します。 | ユーザーロールのゲストログイン。  ユーザーは「習慣を追加」画面にあります。 | 1.ユーザは、フィールド "Tên thói quen"に空白を入力します。  2.ユーザーが「Lưu lại」ボタンをクリックします。 | システムにエラーメッセージ "Tên thói quen không được để trống！"が表示されます。 |  |  |

* + 1. <ユーザー>癖を編集する

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **テストケースの説明** | **事前条件** | **テストケースの手順** | **予想出力** | **結果** | **実行日** |
| AL\_1 | テストユーザーが習慣を正常に編集する | ユーザーロールのゲストログイン。  ユーザーは「習慣を編集」画面にあります。 | 1.ユーザーは、「姓」フィールドに「召集」を入力します。  2.ユーザーが "Xây dựng"ボタンを "Xây dựng/Từ bỏ"の必須フィールドに選択します。  3.ユーザーは,[オプション]ボタンをクリックして、[必須]フィールドに必要なフィールドを入力します。  4.ユーザーは、リストの「sức khỏe」を「Nhóm thói quen」必須フィールドに選択します。  5.ユーザーのチェックボックスをオンにして、必要なフィールドに「日にち」を入力します。  6.ユーザーは「Ngày bắt đầu」を選択し、「31.10.2018」を選択し、「Ngày hoàn thành」にチェックを入れ、「Thời gian thực hiện」必須フィールドに「Mo 31.12.2018」を選択します。  7.ユーザーが選択した青色のボタンを「Màu thói quen」の必須フィールドに合わせます。  8.ユーザーは、 "Nhắc nhở"必須フィールドに "7:00 AM gọi tôi chạy bộ nhé"を追加します。  8.ユーザは、必須フィールドに「重要な情報を入力してください」と入力します。  9.ユーザーが「クリック」ボタンをクリックします。 | システムは "Cập nhật thói quen thành công" |  |  |
| AL\_2 | テストユーザーは、空白の "テンキー"テキストボックスを使用して習慣を編集します。 | ユーザーロールのゲストログイン。  ユーザーは「習慣を編集」画面にあります。 | 1.ユーザは、フィールド "Tên thói quen"に空白を入力します。  2. ユーザーが[Cập nhật]ボタンをクリックします。 | システムにエラーメッセージ "Tên thói quen không được để trống！"が表示されます。 |  |  |

* + 1. <ユーザー> 習慣を削除する

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **テストケースの説明** | **事前条件** | **テストケースの手順** | **予想出力** | **結果** | **実行日** |
| AL\_1 | テストユーザーが習慣を正常に削除する | ユーザーロールのゲストログイン。  ユーザーは「習慣を編集」画面にあります。 | 1.ユーザーが「Chạy bộ」の習慣をクリックします。  2. "Xóa"ボタンをクリックします。 | システムは "Xóa thói quen thành công" |  |  |

# **F1. Software User’s Manual**

1. **Installation Guide**
   1. **Hardware Requirement**

|  |  |
| --- | --- |
| Hardware | Description |
| Internet | Over 8 Mbps |
| Processor | Over Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| RAM | Over 4GB |
| Hard disk driver | Over 40GB |

* 1. **Software Requirement**

**For client:**

|  |  |
| --- | --- |
| Software | Application name / version |
| Operating system | Window 10 Pro-64bit |
| Development Environment | AngularJS |
| DBMS | Microsoft SQL server 2014 v17.4 |
| Source control | Github |
| Web browser | Chrome 63 or above  Firefox 49 or above |

**For android:**

|  |  |
| --- | --- |
| Software | Application name / version |
| Operating system | Android 4.4 or above |
| Development Environment | Android studio |
| DBMS | Microsoft SQL server 2014 v17.4 |
| Source control | Github |

1. **User Guide**
   1. **Login**



|  |  |
| --- | --- |
| Step | Description |
| 1 | Fill in field: “Tên đăng nhập” |
| 2 | Fill in field: “Mật khẩu” |
| 3 | Click on “Đăng nhập” button or “Facebook” button or “Google” button |

* 1. **Register**





|  |  |
| --- | --- |
| Step | Description |
| 1 | Click on “Tôi chưa có tài khoản?Đăng ký” hyperlinks. |
| 2 | Fill in field: “Tên đăng nhập” |
| 3 | Fill in field: “EMail” |
| 4 | Fill in field: “Mật khẩu” |
| 5 | Fill in field: “Nhập lại mật khẩu” |
| 6 | Click on “Đăng ký” button. |

* 1. **View main menu**



* 1. **Add habit**





|  |  |
| --- | --- |
| Step | Description |
| 1 | Click on “+” button |
| 2 | Fill in field: “Tên thói quen” |
| 3 | Click on “đề xuất” button |
| 4 | Click choice “Xây dưng” or “Từ bỏ” button |
| 5 | Click choice “Hàng ngày” or “Hàng tuần” or “Hàng tháng” or “Hàng năm” button |
| 6 | Click choice “Có/Không” button or count time |
| 7 | Click on “Thêm” hyperlink to add group habit. |
| 8 | Select date. |
| 9 | Set start date and finish date. |
| 10 | Click choice another color button. |
| 11 | Click on “Thêm” hyperlink to add reminder for habit. |
| 12 | Fill in field:”Động lực” |
| 13 | Click on “Lưu” button |

* 1. **Edit habit**
  2. **Delete habit**
  3. **View list reminder**
  4. **Add reminder**
  5. **Edit reminder**
  6. **Delete reminder**

# **F2. ソフトウェアユーザーマニュアル**

## 1. インストレーションガイド

### 1.1ハードウェア要件

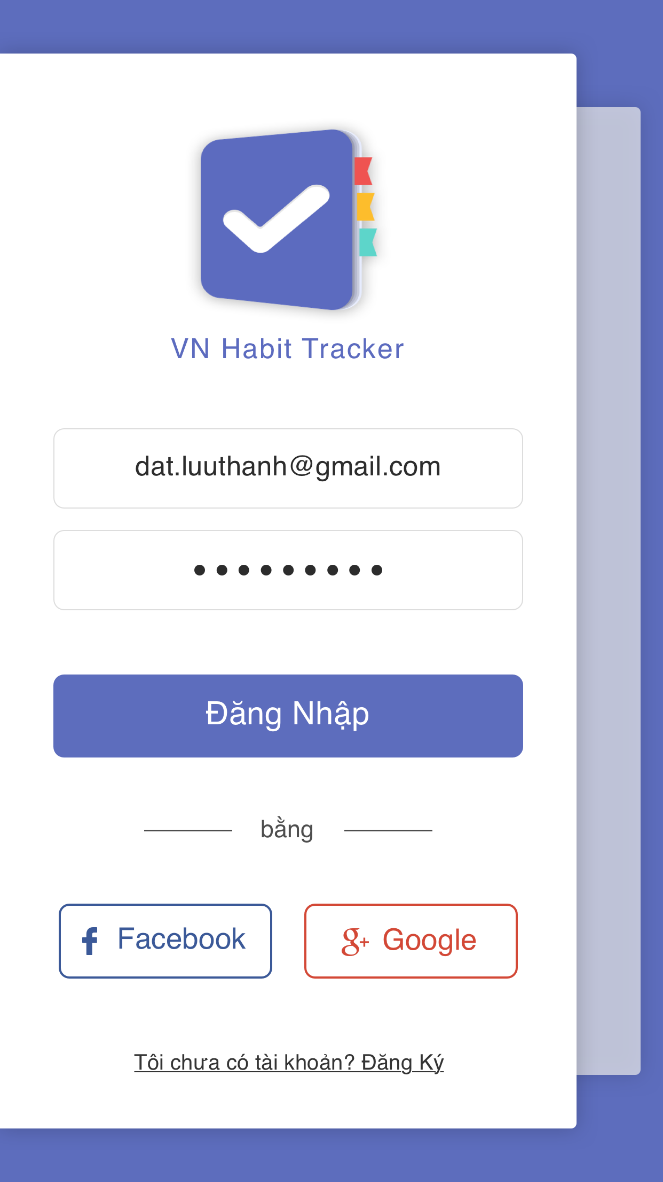
|  |  |
| --- | --- |
| ハードウェア | 記述 |
| インターネット | 8 Mbps以上 |
| プロセッサー | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz以上 |
| RAM | 4GB以上 |
| ハードディスクドライブ | 40GB 以上 |

### 1.2ソフトウェア要件

|  |  |
| --- | --- |
| ソフトウェア | アプリケーション名前／バージョン |
| オペレーティングシステム | Window 10 Pro-64bit |
| 環境 | VueJS 2.0 |
| DBMS | Microsoft SQL server 2014 v17.4 |
| ソース管理 | Github |
| ウェブブラウザ | Chrome 63 以上 |

## 2. ユーザーガイド

### **2.1ログイン**



|  |  |
| --- | --- |
| ステップ | 説明 |
| 1 | ユーザー名の内容を入力します。 |
| 2 | パスワードの内容を入力します。 |
| 4 | 「Login 」のボタンをクリックします。 |

**2.2 レジスター**





|  |  |
| --- | --- |
| ステップ | 説明 |
| 1 | "Tôi chưa có tài khoản？Đăng ký"ハイパーリンクをクリックします。 |
| 2 | 入力フィールド： "Tên đăng nhập" |
| 3 | 入力フィールド： "E-Mail" |
| 4 | 入力フィールド: “Mật khẩu” |
| 5 | 入力フィールド: “Nhập lại mật khẩu” |
| 6 | "Đăng ký" ボタンをクリックしてください。 |

* 1. **メインメニューを表示**



* 1. **習慣を追加する**





|  |  |
| --- | --- |
| ステップ | 説明 |
| 1 | 「+」ボタンをクリック |
| 2 | フィールドに入力してください: “Tên thói quen” |
| 3 | "đề xuất" ボタンをクリックしてください |
| 4 | 選択 "Xây dưng"または "Từ bỏ"ボタンをクリックします |
| 5 | 「hàng ngày」または「hàng tuần」または「hàng tháng」または「hàng năm」ボタンをクリックします |
| 6 | クリックして "Có/Không" ボタンをクリックするか、時間を数える |
| 7 | グループの習慣を追加するには、 "Thêm" ハイパーリンクをクリックしてください。 |
| 8 | 日付を選択してください。 |
| 9 | 開始日と終了日を設定します。 |
| 10 | 選択した別のカラーボタンをクリックします。 |
| 11 | 「Thêm」ハイパーリンクをクリックして、習慣のためのリマインダを追加します。 |
| 12 | フィールドに入力してください： "Động lực" |
| 13 | “Lưu" ボタンをクリック |

1. **Appendix**
2. **SOFTWARE ENGINEERING 9TH EDITION, BY IAN SOMMERVILLE**
3. **STARUML DOCUMENTATION FOR ACTIVITY DIAGRAM:**

<http://www.uml-diagrams.org/activity-diagrams-controls.html>

1. **STARUML DOCUMENTATION OF IBM:**

<https://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep03/f_umlbasics_db.pdf>

1. **UML FOR DRAWING DIAGRAM (2.0 STANDARD)**

<http://www.omg.org/spec/UML/2.0>

1. **VUEJS FEATURES**

[1] [2] [3] <https://en.wikipedia.org/wiki/Vue.js>