|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Logo_FPT_University_doc** |  | | | **FPT UNIVERSITY** | | |
| Capstone Project |
| Software Module for Delivery Helper  Software Requirements Specification |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | **Group 5** | | | | | **Group Members** | Nguyễn Hoàng Việt Khánh |  | 60566 | | Nguyễn Thị Yên Thịnh |  | 60269 | | Nguyễn Đỗ Vượng |  | 60267 | | An Ngọc Anh |  | 60277 | | **Supervisor** | Lâm Hữu Khánh Phương | | | | **Capstone Project code** | SMDH | | | | |
| * Ho Chi Minh City, Oct / 2012 - |

# Table of Contents

[Table of Contents 2](#_Toc351656406)

[3.1 User Requirement Specification 3](#_Toc351656407)

[3.1.1 Common Features 3](#_Toc351656408)

[3.1.2 Plan Management 3](#_Toc351656409)

[3.1.3 Item Management 3](#_Toc351656410)

[3.1.4 Hub’s Order Tracking 3](#_Toc351656411)

[3.1.5 Order info checking 3](#_Toc351656412)

[3.1.6 Order Management 4](#_Toc351656413)

[3.2 System Requirement Specification 4](#_Toc351656414)

[3.2.1 External Interface Requirements 4](#_Toc351656415)

[3.2.1.1 User Interfaces 4](#_Toc351656416)

[3.2.1.2 Hardware Interfaces 4](#_Toc351656417)

[3.2.1.3 Software Interfaces 4](#_Toc351656418)

[3.2.1.4 Communications Protocol 5](#_Toc351656419)

[3.2.2 System Features 6](#_Toc351656420)

[3.2.2.1 Manage Plan 7](#_Toc351656421)

[3.2.2.2 Manage Hub 14](#_Toc351656422)

[3.2.2.3 Manage Items 19](#_Toc351656423)

[3.2.2.4 Create Order 23](#_Toc351656424)

[3.2.3 Software System Attributes 24](#_Toc351656425)

[3.2.3.1 Reliability 24](#_Toc351656426)

[3.2.3.2 Availability 24](#_Toc351656427)

[3.2.3.3 Security 24](#_Toc351656428)

[3.2.3.4 Maintainability 24](#_Toc351656429)

# User Requirement Specification

## Common Features

* Only authenticated users can access the system (except Shopping Cart page). Users can log in and log out using their own accounts.
* Users can change their password.
* Only authorized users can use specific functions of the system.

## Plan Management

* Tiktak staff can make a new plan automatically by entering the number of plan(s), choosing the type of group request(s)/order(s). Then, add request(s)/order(s) to *Selected Request/Order* box.
* Tiktak staff can cancel those plans if they haven’t saved to database yet.
* Tiktak staff can view all plans by *Plan Management* screen.
* After plan created, delivery men will be doing if the delivery men are assigned in that plan.

## Item Management

* Vendors must have signed a contract with TikTak before they could use the delivery service. Then, office staff will create a new account. After **t**hat, they can use this account to access the system.
* Vendors can create new product information or modify existed product info.

## Hub’s Order Tracking

* At Hub, we can see orders which have status: Sending to Hub, In Hub, and Delivered, Waiting for Return, Returned.
* When products of order are in hub, Hub staff will be changed status from “Sending to Hub” to “In Hub”. Hub staff can click checkbox and then, click “Change status” button to change status of orders in the same time.
* Hub Staff can view all orders and items’ information for checking on their dates, statuses, etc.
* Office Staff can filter all orders to view and to do with it faster.

## Order info checking

* After receiving passcode, customers will go to the Hub and then, give hub staff that passcode to receive order info. The Hub staff will enter the passcode, if it’s right, system will be shown the receiver information. After that, hub staff will deliver the product to customers. If it’s wrong, the Hub staff will notify to customers and let their known they should check their passcode again.
* When hub staff clicks “Mark as finished” button, system will be updated status of that order becomes *Delivered* statusautomatically.

## Order Management

* Customer can create new order via Shopping Cart.
* After reviewed information of products, customer will click “Add” button after product he/she wants to buy on Shopping Cart page. Then, customer can choose option to delivery, and click “Buy” button.

# System Requirement Specification

## External Interface Requirements

### User Interfaces

* The design should be simple and user-friendly. White and dark will be 2 main colors of the website while charts may have more color to visually express data more effectively.
* The menu bar should be on the left and not take too much space of the screen. It consists of the list of main functions of the system that users can access.
* The design should be responsive. It means that the web components should be scaled according to a range of resolutions and devices to provide a consistent experience, no matter what.

### Hardware Interfaces

* To access to the system, users only need any type of computer, tablet, or mobile phone with a fair internet connection.

### Software Interfaces

* At the server side, the system should run on top of Windows 7, Windows Server 2008 or later versions of Windows Server. Besides, Microsoft .NET Framework 4 and MVC 3 should be installed on the server. The database management system use for SMDH is SQL Server 2008 R2.
* At the client side, users can use any modern browser that supports JavaScript and HTML 5 to access to the system.

### Communications Protocol

HTTP is the protocol used for loading the web site in browsers.

## System Features



Figure 1 - General Use Case Diagram

### Manage Plan



Figure 2 – Manage Plan

#### Create Collection Plan



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – CREATE COLLECTION PLAN | | | |
| Use-case No. | UC01.1 | Use-case Version | 1.0 |
| Use-case Name | Create collection plan | | |
| Author | KhanhNHV | | |
| Date | 03/02/2013 | Priority | Normal |
| **Actor:** Tiktak staff User  **Summary:** This use case is about how the tiktak staff create a collection plan automatically  **Goal:** Create a collection plan automatically successful  **Triggers:** Actor enter plan(s) no to create a plan  **Pre-conditions:** Actor logged in as Tiktak staff user successfully  **Post-conditions:** plan is created successfully  **Main Success Scenario:**   1. Enter number of plan(s) actor want to create. 2. Choose type of group (*all* or *multi (*enter number of request(s)) 3. Add *request(s)* to make a new collection plan. 4. Click *Solve* button.   **Alternative Scenario:**  1. If the number of plan(s) is not positive number.  **Exceptions:** If the number of plan(s) is not positive number, the system will show the message: “Invalid number”.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Cancel Collection Plan



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – CANCEL COLLECTION PLAN | | | |
| Use-case No. | UC01.1 | Use-case Version | 1.0 |
| Use-case Name | Cancel collection plan | | |
| Author | KhanhNHV | | |
| Date | 03/02/2013 | Priority | Normal |
| **Actor:** Tiktak staff User  **Summary:** This use case is about how the tiktak staff cancel collection plan that has just created.  **Goal:** Cancel plan that has just created.  **Triggers:** Plan has created, but it’s not saved to database.  **Pre-conditions:** Plan has just created but it is not saved to database.  **Post-conditions:** Plan is canceled successful.  **Main Success Scenario:**   1. Click *Back* button. 2. System will be redirect to *Create collection plan* page.   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Assign Delivery men to Collection Plan



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – ASSIGN DELIVERY MEN TO COLLECTION PLAN | | | |
| Use-case No. | UC01.3 | Use-case Version | 1.0 |
| Use-case Name | Assign delivery men to collection plan | | |
| Author | KhanhNHV | | |
| Date | 03/02/2013 | Priority | Normal |
| **Actor:** Tiktak staff User  **Summary:** This use case is about how the tiktak staff assigns delivery men to the collection plan.  **Goal:** Assign available delivery men to the collection plan.  **Triggers:** N/A  **Pre-conditions:** Plan has created successfully.  **Post-conditions:** Assign delivery men to plan successful.  **Main Success Scenario:**   1. Click *View Detail* button. 2. Check the DM(s) who will implement this plan.   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Create Delivery Plan



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – CREATE DELIVERY PLAN | | | |
| Use-case No. | UC01.4 | Use-case Version | 1.0 |
| Use-case Name | Create delivery plan | | |
| Author | KhanhNHV | | |
| Date | 03/02/2013 | Priority | Normal |
| **Actor:** Tiktak staff User  **Summary:** This use case is about how the tiktak staff create a collection plan automatically  **Goal:** Create a collection plan automatically successful  **Triggers:** Actor enter plan(s) no to create a plan  **Pre-conditions:** Actor logged in as Tiktak staff user successfully  **Post-conditions:** plan is created successfully  **Main Success Scenario:**   1. Enter number of plan(s) actor want to create. 2. Choose type of group (*all* or *multi (*enter number of order(s)) 3. Add *order(s)* to make a new delivery plan. 4. Click *Solve* button.   **Alternative Scenario:**  1. If the number of plan(s) is not positive number.  **Exceptions:** If the number of plan(s) is not positive number, the system will show the message: “Invalid number”.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Cancel Delivery Plan



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – CANCEL DELIVERY PLAN | | | |
| Use-case No. | UC01.5 | Use-case Version | 1.0 |
| Use-case Name | Cancel delivery plan | | |
| Author | KhanhNHV | | |
| Date | 03/02/2013 | Priority | Normal |
| **Actor:** Tiktak staff User  **Summary:** This use case is about how the tiktak staff cancel delivery plan that has just created.  **Goal:** Cancel plan that has just created successful.  **Triggers:** Plan has created, but it’s not saved to database.  **Pre-conditions:** Plan has just created but it is not saved to database.  **Post-conditions:** Plan is canceled successful.  **Main Success Scenario:**   1. Click *Back* button. 2. System will be redirect to *Create delivery plan* page.   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Assign Delivery men to Delivery Plan



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – ASSIGN DELIVERY MEN TO DELIVERY PLAN | | | |
| Use-case No. | UC01.6 | Use-case Version | 1.0 |
| Use-case Name | Assign delivery men to delivery plan | | |
| Author | KhanhNHV | | |
| Date | 03/02/2013 | Priority | Normal |
| **Actor:** Tiktak staff User  **Summary:** This use case is about how the tiktak staff assigns delivery men to the delivery plan.  **Goal:** Assign available delivery men to the delivery plan.  **Triggers:** N/A  **Pre-conditions:** Plan has created successfully.  **Post-conditions:** Assign delivery men to plan successful.  **Main Success Scenario:**   1. Click *View Detail* button. 2. Check the DM(s) who will implement this plan.   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

### Manage Order



Figure 3: Hub Management

#### Check Order Info to delivery

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – CHECK ORDER INFO TO DELIVERY | | | | |
| Use-case No. | UC02.2 | Use-case Version | | 1.0 |
| Use-case Name | Check Order Info to delivery | | | |
| Author | ThinhNTY | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** hub Staff  **Summary:** This use case is about how hub staff checks order info to delivery  **Goal:** User can check order info to delivery  **Triggers:**  **Pre-conditions:** User logged in with hub staff role  **Post-conditions:** Order  **Main Success Scenario:**   1. In main page at Hub, hub Staff enter the passcode and click “Submit” 2. System will show receiver information below.   **Alternative Scenario:** 2. If the passcode is incorrect. System will show error message: “There’s no orders in system. Please check it again!”  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### View Order Status



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – VIEW ORDER STATUS AT HUB | | | | |
| Use-case No. | UC02.3 | Use-case Version | | 1.0 |
| Use-case Name | View order status | | | |
| Author | ThinhNTY | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Hub Staff, Customer  **Summary:** This use case is about how to view order status  **Goal:** Hub staff or customer can view order status by role  **Triggers:**  **Pre-conditions:** User logged in with hub staff/ customer role.  **Post-conditions:** View order status successful  **Main Success Scenario:**  1. In the main page of hub/Customer, choose the tab that status you want to view  2. System will show all records which are belong to that tab  **Alternative Scenario:** 2. If the status has no order belong to, output return by message “No data available”  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Change status



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – CHANGE STATUS | | | | |
| Use-case No. | UC02.4 | Use-case Version | | 1.0 |
| Use-case Name | Change status | | | |
| Author | ThinhNTY | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Hub Staff, Customer  **Summary:** This use case is about how to change status  **Goal:** User can change status successfully.  **Triggers:**  **Pre-conditions:** User logged in with office staff/ customer role.  **Post-conditions:** Order is added successfully.  **Main Success Scenario:**   1. In main page of order, user click *Sending to Hub* (if Hub staff), *Draft* (if Customer) tab. 2. User chooses the order want to change status. 3. User clicks *Change status* button. 4. The order will be removed from *Sending to Hub/Draft* tab. It will be moved to *In Hub/Approval* tab   **Alternative Scenario:**  **Exceptions:** If there’s no order belong to *Sending to Hub/Draft.* The *Change status* button is not appeared.  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.2.4 Make an Order



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – MAKE AN ORDER | | | | |
| Use-case No. | UC02.4 | Use-case Version | | 1.0 |
| Use-case Name | Make an order | | | |
| Author | AnhAN | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Buyer  **Summary:** This use case is about how to make an order  **Goal:** User can make an order successfully.  **Triggers:** N/A  **Pre-conditions:** N/A  **Post-conditions:** Order is created successfully.  **Main Success Scenario:**   1. In main page of Shopping Cart, user can add product to Cart. 2. User fill in information about 3. User clicks *Change status* button. 4. The order will be removed from *Sending to Hub/Draft* tab. It will be moved to *In Hub/Approval* tab   **Alternative Scenario:**  **Exceptions:** If there’s no order belong to *Sending to Hub/Draft.* The *Change status* button is not appeared.  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

### Manage Item



#### Add Item



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – ADD ITEM | | | | |
| Use-case No. | UC03.1 | Use-case Version | | 1.0 |
| Use-case Name | Add Item | | | |
| Author | VuongND | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Vendor  **Summary:** This use case is about how vendor create a new item.  **Goal:** User can create a new item successfully.  **Triggers:** In main page of item management, user click “ADD” button.  **Pre-conditions:** User logged in with vendor role.  **Post-conditions:** A new item is created successfully.  **Main Success Scenario:**   1. In main page of item management, user input product information (name, height, weight, width, length, price and image) and clicks “ADD” button. 2. System will save new product info into database.   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Update item information



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – UPDATE ITEM INFORMATION | | | | |
| Use-case No. | UC03.2 | Use-case Version | | 1.0 |
| Use-case Name | Update item information | | | |
| Author | VuongND | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Vendor  **Summary:** This use case is about how to update existed item information.  **Goal:** User can update existed item information.  **Triggers:** In main page of item management, click “EDIT” action on each record.  **Pre-conditions:** User logged in with vendor role.  **Post-conditions:** Existed item is updated.  **Main Success Scenario:**   1. In main page of item management, user clicks ‘Edit’ icon on record that user wants to update. 2. User input info into the field user wants to update. 3. User clicks “OK”. 4. System will update item information   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Remove Item



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – REMOVE ITEM | | | | |
| Use-case No. | UC03.3 | Use-case Version | | 1.0 |
| Use-case Name | Remove item | | | |
| Author | VuongND | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Vendor  **Summary:** This use case is about how to remove an item.  **Goal:** User can remove item.  **Triggers:** In main page of item management, user click “delete” icon.  **Pre-conditions:** User logged in with vendor role.  **Post-conditions:** Order will be deleted successful.  **Main Success Scenario:**   1. In main page of item management, user clicks “Delete” icon of that order. 2. System will appear a message box to confirm. 3. User clicks “OK” button to delete.   **Alternative Scenario:** In step 3, if user wants to cancel the process, user can clicks to “Cancel” button to back the main page of item management.  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

### Manage Request (not implement yet)

#### 3.2.2.4.1 Add New Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – CREATE ORDER | | | | |
| Use-case No. | UC04.1 | Use-case Version | | 1.0 |
| Use-case Name | Check order detail information | | | |
| Author | ThinhNTY | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Customer  **Summary:** This use case is about how customer create a new order  **Goal:** Customer can create a new order successfully  **Triggers:** N/A  **Pre-conditions:**  N/A  **Post-conditions:** The order is created successfully.  **Main Success Scenario:**   1. In main page of customer, user clicks “Add” button. 2. System will appear a popup to input information. 3. User inputs request information and clicks “Create” button. 4. New customer appears in customer list.   **Alternative Scenario:** In step 3, if user wants to cancel the process, user can clicks to “Cancel” button to back the main page of customer.  **Exceptions:** When inputs require information is invalid format or missed require information, the system will notify and require input again.  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.4.2 Edit Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – CREATE ORDER | | | | |
| Use-case No. | UC04.1 | Use-case Version | | 1.0 |
| Use-case Name | Check order detail information | | | |
| Author | ThinhNTY | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Customer  **Summary:** This use case is about how customer create a new order  **Goal:** Customer can create a new order successfully  **Triggers:** N/A  **Pre-conditions:**  N/A  **Post-conditions:** The order is created successfully.  **Main Success Scenario:**   1. In main page of customer, user clicks “Add” button. 2. System will appear a popup to input information. 3. User inputs request information and clicks “Create” button. 4. New customer appears in customer list.   **Alternative Scenario:** In step 3, if user wants to cancel the process, user can clicks to “Cancel” button to back the main page of customer.  **Exceptions:** When inputs require information is invalid format or missed require information, the system will notify and require input again.  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.4.3 Remove Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – CREATE ORDER | | | | |
| Use-case No. | UC04.1 | Use-case Version | | 1.0 |
| Use-case Name | Check order detail information | | | |
| Author | ThinhNTY | | | |
| Date | 03/02/2013 | Priority | Normal | |
| **Actor:** Customer  **Summary:** This use case is about how customer create a new order  **Goal:** Customer can create a new order successfully  **Triggers:** N/A  **Pre-conditions:**  N/A  **Post-conditions:** The order is created successfully.  **Main Success Scenario:**   1. In main page of customer, user clicks “Add” button. 2. System will appear a popup to input information. 3. User inputs request information and clicks “Create” button. 4. New customer appears in customer list.   **Alternative Scenario:** In step 3, if user wants to cancel the process, user can clicks to “Cancel” button to back the main page of customer.  **Exceptions:** When inputs require information is invalid format or missed require information, the system will notify and require input again.  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

## Software System Attributes

### Reliability

* The database can be backed up easily and recovered if necessary.
* The system should never crash or hang, except for the cause from an operating system or network error.
* Mean Time Between Failures (MTBF): The acceptable failure is once a year. The failure should not be because of the database, or else the data may be lost and cannot be recovered.
* Mean Time To Repair (MTTR): When the failure occurs, it should take as little time as possible to repair. The acceptable mean time for a particular failure must be less than 8 hours.
* Maximum Bugs or Defect Rate: 05 bugs / KLOC

### Availability

* The uptime percentage should be at least 99.95%.
* The acceptable time for maintenance or backup should not be more than 8 hours per month.
* When the system goes in under-maintenance, the website should display a message to inform that.

### Security

* Only users with proper account can access certain information of the system. All the information of users must not be available for anyone or software that is not part of the system. User password is also encrypted and not available to the system administrators.

### Maintainability

* The code must follow C#.NET coding and naming convention.
* There should be comments in code files that explain the functions of each code segment.
* All the errors should be logged, which supports bug fixing and maintenance.
  1. **Entity Relationship Diagram**

