





|  |  |  |  |
| --- | --- | --- | --- |
| **Faculty:** |  | **Mr.Hoan Duc Quang** |  |
| **Batch No:** |  | **F2131405T0** |  |
| **Group No:** |  | **4** |  |
| **Students:** |  | **Fullname** | **Roll No.** |
|  | 1. | Hua Tran Huu Trung | Student812912 |
|  | 2. | Ngo Ngoc Thoi | Student797693 |
|  | 3. | Pham Le Viet Tin | Student805972 |
|  | 4. | Le Minh Dong | Student806079 |

**Ho Chi Minh City,** **December 2015**

**Table of contents**

Contents

[REVIEW 1 3](#_Toc439087028)

[I. Introduction 4](#_Toc439087029)

[II. Existing Scenario 4](#_Toc439087030)

[III. Customer Requirement Specification 4](#_Toc439087031)

[IV. Functional Requirement Specification 5](#_Toc439087032)

[***1.*** **Login and logout*.*** 5](#_Toc439087033)

[***2.*** **Change password.** 5](#_Toc439087034)

[***3.*** **Change profile.** 5](#_Toc439087035)

[***4.*** **Manage draft document by Staff** 5](#_Toc439087036)

[***5.*** **Approve, return, remove draft by Manager** 5](#_Toc439087037)

[***6.*** **Process transfer draft by Manager** 6](#_Toc439087038)

[V. System requirements, Development software and Technology 8](#_Toc439087039)

[**1.** **System requirements** 8](#_Toc439087040)

[**2.** **Development software** 9](#_Toc439087041)

[**3.** **Technology** 9](#_Toc439087042)

[VI. Tasksheet 10](#_Toc439087043)

# **REVIEW 1**

1. **Introduction**

With rapidly-changing technology and an increasing emphasis on managing information properly, clients are looking for experts to help them address business-critical issues. Organizations have unique needs that are not often met by consumer products; for example, the search and retrieval functions of common consumer products are not appropriate for organizational purposes. And while you may know that you need assistance figuring out how to help your organization improve efficiency, profitability and productivity, you may not know where to begin.

Here at APTECH Soft®, we have a commitment to keep moving forward. As a part of this commitment, we actively support document management initiatives to increase knowledge about digital document management. From Web to white papers, APTECH Soft is committed to providing best solutions to help businesses increase efficiency and productivity.

You shouldn’t have to change the way you work to improve the efficiency of your work processes. With a quality digital document management solution, you can work the way you always have, just without the clutter and limitations of paper. In fact, according to research conducted by APTECH Soft, when you implement a digital document management solution, your organization can reduce overall document-related costs by 30%.

By implementing document and records management software, business owners can realize many benefits that noticeably improve organizational efficiency. Digital document management systems can help your organization succeed by:

* Saving money.
* Saving time
* Increasing efficiency
* Increasing productivity
* Increasing inter-departmental communication and collaboration

1. **Existing Scenario**

Makes 3 - 15 copies and spends 5$ - 15$ on paper of each document.

Loses 1 out of 20 office documents.

Spends 10$ for recreating each lost document.

Employees spend 20% of their day looking for information in hardcopy documents and that, 50% of the time, they can’t find what they need.

Too hard to find document related to others.

1. **Customer Requirement Specification**

A quality system makes it possible for multiple users to access the same files at the same time

Web systems should be fully searchable. Helping users quickly search through thousands of documents in order to pinpoint the information they need.

Share documents with others while protecting confidential information.

Manage millions of documents and retrieve the right one in seconds.

Answering information requests immediately, eliminating call-backs and phone tag.

Cutting time spent copying and distributing documents to staff, branch offices and outside contacts.

Speeding document-handling workflow by enabling automation

Support the entire group of organizational users concurrently.

Store all documents for the entire organization.

Document lifecycle management.

1. **Functional Requirement Specification**
   1. **Login and logout*.***
   2. Description

The system must provide a page to login user’s account and function log out (for logged in user) with the following information:

* Username.
* Password.
  1. Functional Requirements

1. User must provide Username and Password to login.
2. After user logged in the user can use function log out.
   1. **Change password.**
   2. Description

The system must provide a page to change user’s password, user must logged in in website can do this function. User must provide:

* Current password.
* New password.
* Retype new password.
  1. Functional Requirements

1. User must provide correct current password before changing.
2. New password and retype password must match.
   1. **Change profile.**
   2. Description

The system must provide a page to view, change user’s profile. User’s information which they can change such as:

* Full name
* Email
* Gender
* Day of Birth
  1. Functional Requirements

1. User provides Full name, email, day of birth must not be blank. Validate email and day of birth to change profile.
   1. **Manage draft document by Staff**
   2. Description

The system must provide a page which staff creates, update, remove draft document.

Create function with following information:

* Abstract content
* Type
* Date
* File

Update function with following information:

* Abstract content
* Type
* Date
* File
  1. Functional Requirements

1. Staff can view draft.
2. Staff can update draft.
3. Staff can remove draft which has not approved yet by his/her Manager.
   1. **Search document**
   2. Description

The system must provide a page which user can search document. User must input with following information:

* From date
* To date
* Abstract content
* Number
* Type of document
* Document type (published or draft or All)
  1. Functional Requirements

1. User should provide at least a condition to search document. If user leave a field to blank, this mean is no condition for that field.
   1. **Approve, return, remove draft by Manager**
   2. Description

Manager can approve, return or remove draft sent by his/her staff.

If he has chosen return, then his/her staff must to update this draft.

If he has chosen approve then system provides a pop-up information next step. Manager should input description to transfer to Top manager.

If he has chosen remove draft which has not transfer to other.

* 1. Functional Requirements

1. If Manager approves draft then the system provide a functional transfer to Top manager.
2. If Manager returns draft then his/her staff can update this.
3. If Manager removes draft then document’s status changed to “removed”.
   1. **View, cancel transferred draft by Manager**
   2. Description

Manager can view list or cancel draft which he/she sent to Top manager but Top manager has not action on this yet.

* 1. Functional Requirements.

1. System provides a page that Manager can view list of transferred draft.
2. Manager can remove draft’s history transfer to cancel transfer.
   1. **Process approve, not approve, return draft by Top manager**
   2. Description

Top manager will receive draft which was approved by Manager and make a decision if this draft can be approved or not.

* 1. Functional Requirements

1. If Top manager choose approve then this draft’s status changed to “approved”.
2. If Top manager choose not approve, he/she must leave a comment on this and its status changed to “denied”.
3. If Top manager choose return then this return to Manager who transfer to him/her and leave a comment to Manager edit content of draft.
   1. **Manage published document by Stationary**
   2. Description

The system must provide a page which Stationary can create, update and remove published document.

* Number
* Is Local or not
* Source’s number
* Document date
* Valid date from
* Valid date to
* Abstract content
* File
* Is need to be reply or not
* Can choose reference other published document or draft.
  1. Functional Requirements

1. Stationary can view list of published document.
2. Stationary can create published document.
3. Stationary can update published document.
4. Stationary can remove published document.
   1. **Process transfer published document which isn’t needed to reply by Stationary**
   2. Description

If published document isn’t needed to reply, then the system must provide a page which Stationary can transfer published document to other departments.

* 1. Functional Requirements

1. Stationary can view and choose multiple of department on list of departments to transfer.
   1. **Process transfer published document which is needed to reply by Stationary**
   2. Description

If the published document’s property is needed to reply then Stationary must choose a department which needs reply on this.

* 1. Functional Requirements

1. Stationary can view and choose multiple of department on list of departments to transfer.
2. Stationary can view and choose a department on list of departments to transfer if property of published document is needed to reply.
   1. **Process transfer published document by Manager**
   2. Description

The system must provide a page which manager can receive published documents transferred by Stationary. Then he/she should choose these for transferring to his/her Staff if he wants.

* 1. Functional Requirements

1. View list of his/her Staff who has not received this document yet, then he choose for transferring.
2. View and remove published document’s history transfer to cancel transfer.
   1. **View published document by Staff**
   2. Description

The system must provide a page which Staff can receive published documents are transferred by his/her Manager or Stationary.

* 1. Functional Requirements

1. Staff can view list of published document which they have permission to view.
   1. **Process reply on published document by Manager**
   2. Description

When Stationary transfer a published document which is needed to reply, then relevant Manager should leave a comment or attach file to reply to Stationary.

* 1. Functional Requirements

1. Manager received published document and have a function to comment or attach file to reply Stationary.
   1. **Process convert draft to published document by Stationary**
   2. Description

Once Top manager approved a draft, it will be transferred to Stationary. Stationary should publish this like creating a published document.

* 1. Functional Requirements

1. Stationary receive draft from Top manager after being approved. Then Stationary should convert draft to published document.
   1. **Process update status of published document which is needed to reply by Stationary**
   2. Description

The system provides a page which Stationary can manage status of published documents transferred to a department for processing reply.

* 1. Functional Requirements

1. Stationary can update status of any published document to “replied”.
   1. **Process update status of incoming document from outside by Stationary**
   2. Description

The system provides a page which Stationary can manage status of incoming documents from outside which is replied to outgoing document sent before.

* 1. Functional Requirements

1. Stationary can update status of an incoming document from outside.
   1. **Process update outgoing published document needed to reply by Stationary**
   2. Description

The system provides a page which Stationary can manage status of outgoing published documents needed to reply.

* 1. Functional Requirements

1. Stationary can update status of an outgoing published documents needed to reply.
   1. **Process update outgoing published document not needed to reply by Stationary**
   2. Description

The system provides a page which Stationary can manage status of outgoing published documents not needed to reply.

* 1. Functional Requirements

1. Stationary can update status of an outgoing published documents not needed to reply.
   1. **Manage user by IT**
   2. Description

The system must provide a page which IT view, create, update, reset password for user with following information:

* User name.
* User password.
* Full name.
* Email.
* Gender.
* Date of Birth.
* User department
* Role
  1. Functional Requirements

1. IT can view list of users.
2. IT can create user.
3. IT can update user.
4. IT can reset password for user.
   1. **Manage role by IT**
   2. Description

The system must provide a page which IT view create, update, remove role with following information:

* Role name.
* Role description.
  1. Functional Requirements

1. IT can view list of role.
2. IT can create role.
3. IT can update role.
   1. **Manage department by IT**
   2. Description

The system must provide a page which IT view, create, update, remove department with following information:

* Department name.
* Department description.

There are 2 system department which user can not remove are “Phòng Văn thư” and “Phòng Giám đốc”.

* 1. Functional Requirements

1. IT can view list of department.
2. IT can create department.
3. IT can update department.
4. IT can remove department.
5. **System requirements Development software and Technology**
6. **System requirements**
   1. **Hardware**
      1. **Server**

|  |  |
| --- | --- |
| Processor | Dual core or higher. |
| Memory | 4GB RAM or greater. |
| HDD | 10GB or greater |
| Network | Connecting to local network |

* + 1. **Client**

|  |  |
| --- | --- |
| Processor | Pentium III or higher. |
| Memory | 512 MB RAM or greater. |
| Monitor | Support screen resolution at 1024x768 pixels |
| Pointing Device | Microsoft Mouse, Microsoft IntelliMouse®, or compatible pointing device. |
| Network | Connecting to local network |

* 1. **Software**
     1. **Server**

|  |  |
| --- | --- |
| Operation System | Windows 2000/XP/2003 |
| Browser | Internet Explorer 6.0 or higher. |
| Database | Microsoft SQL Server 2000 |
| Software | Java SDK 1.7 Glassfish server 4.0 |

* + 1. **Client**

|  |  |
| --- | --- |
| Operation System | Windows XP or later. |
| Browser | Internet Explorer 7.0 or higher. |

1. **Development software**

* Microsoft Windows XP Professional SP3
* Netbeans IDE
* Java SDK 1.7
* Glassfish 4.0
* Microsoft SQL Server 2005 SP2
* Internet Explorer 7.0

1. **Technology**

* J2EE
* Java Restful Service
* JSF
* HTML
* Durandal
* EJB

1. **Tasksheet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref.No: 4** | | **Project Title:**  **Document Management System** | **Date of Preparation of Activity Plan** | | | |
| **Sr. No.** | **Task** | | **Actual Start Date** | **Actual Days** | **Team Member Names** | **Status** |
| 01 | Introduction | | Dec, 12, 2015 | 6 | All Members | Completed |
| 02 | Exiting Scenario | | All Members | Completed |
| 03 | Customer Requirement Specification | | All Members | Completed |
| 04 | Functional Requirement Specification | | All Members | Completed |
| 05 | System requirements, Development software and Technology | | All Members | Completed |
| 06 | Tasksheet | | All Members | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 4 | Approved By: Faculty |
| Date: Dec, 2015 | Team Leader  **Hua Tran Huu Trung** |  |

# **REVIEW 2**

1. **Use case and Sequence diagram**
2. **Common function**
   1. **Login and Log out**

|  |  |  |
| --- | --- | --- |
| **Author** | Pham Viet Hien | |
| **Use Case Name** | Login | |
| **Actors** | Buyer or GoldBuyer or Seller | |
| **Description** | Buyer or GoldBuyer or Seller wants to login to website as a Buyer or GoldBuyer or Seller member. | |
| **Requirements** | Actor provides username and password | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: User logged in to website.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor go to Home page of website.  3. Actor inputs UserName and Password, then click the Login button. | System Responses:  2. System redirects to Home page and displays Login form on the left with the following control:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Go’ button  4. System checks the information  [Exception 1]  5. System redirects to Actor’s Home page. |
| **Alternative flow** | Actor Actions:  [Alternative 2]  1. Actor click ‘OK’ button, inputs UserName and Password, then click the Login button.  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System checks the information  [Exception 1]  3. System redirects to Actor’s Home page.  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor input invalid username and password. | System Responses:  - System redirect to Login page with the following controls:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Login’ button.  - ‘Back to Home’ button.  System show a message box: “Invalid Username and password”.[Alternative 2] |

* 1. **Change password**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Change password | |
| **Actors** | Staff | |
| **Description** | Staff changes password | |
| **Requirements** | Staff must be logged in to perform | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest creates a Buyer account in this website  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Staff click ‘ChangePassword’ link in Staff page  3. Staff may change password.  5.1 Staff click on ‘Save’ button  5.2 Staff click on ‘Back’ button | System Responses:  2. System redirects to ‘ChangePassword’ page.  4. System redirects to ‘ChangePassword’ page and displays sample form with the following control :  - ‘Current Password’text field  - ‘New Password’ text field  - ‘ReNew Password’ text field  - ‘Save’ button  - ‘Back’ button  6. System checks the information  [Exception 1]  7. System updates the password of staff into database, show message: ‘Change password success!’ and then redirects to Staff page. |
| **Alternative flow** |  |  |
| **Exception** | Staff Actions:  [Exception 1]  1. Staff lefts ‘CurrentPassword’ text field blank.  2. Staff enter the ‘CurrentPassword’ is not correct  2. Staff lefts ‘NewPassword’ text field blank.  3. Staff lefts ‘ReNewPassowrd’ text field blank.  4. ‘NewPassword’ and ‘ReNewPassword’ are not the same | System Responses:  System displays message to Staff :  1. ‘CurrentPassword cannot be blank’.  2. ‘CurrentPassword’ is incorrect.  2. ‘NewPassword cannot be blank’’.  3. ‘ReNewPassword cannot be blank’.  4. ‘NewPassword and ReNewPassword must be the same’. |

* 1. **Change information**
  2. **Search document**

1. **Staff**
   1. **Manage draft**
      1. Create new draft

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Create new draft document | |
| **Actors** | Staff | |
| **Description** | Staff want to create a new draft document to save into personal folders or send to manager | |
| **Requirements** | Staff must be logged in to perform | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest creates a Buyer account in this website  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Staff click ‘Create’ link in Staff page  3. Staff clicks on to Create button | System Responses:  2. System redirects to Create new draft document page.  4. System redirects to Create page and displays sample form with the following control :  - ‘Document type’ combo box  - ‘File’ choose file  - ‘AbstractContent’ text field  5. System checks the information  [Exception 1]  6. System inserts the draft document into database, show message: ‘Draft document’s ‘name’ has been created!’ and then redirects to Staff page. |
| **Alternative flow** |  |  |
| **Exception** | Staff Actions:  [Exception 1]  1. Staff don’t choose ‘Document type’.  2. Staff don’t choose file.  3. Staff lefts ‘AbstractContent’ text field blank. | System Responses:  System displays message to Staff :  1. ‘Please choose document type’.  2. ‘Please choose a file’.  3. ‘AbstractContent cannot be blank’ |

* + 1. Update draft document

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update draft document | |
| **Actors** | Staff | |
| **Description** | Staff update draft document that is returned from the manager | |
| **Requirements** | Staff must be logged in to perform | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest creates a Buyer account in this website  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Staff click ‘Update’ link in Staff page  3. Staff select one draft document to update.  4. Staff click on ‘Update’ button | System Responses:  2. System redirects to ‘Update’ page and express draft documents,  displays sample form with the following control :  A list including:  + Checkbox  + Documenttype  + File  + AbstractContent  - ‘Update’ button  5. System checks the information  [Exception 1]  6.1 System redirects to ‘UpdateDDocument2’ page. |
| **Alternative flow** |  |  |
| **Exception** | Staff Actions:  [Exception 1]  1. Staff don’t choose any draft document to update.  2. Staff choose multiple draft documents to update. | System Responses:  System displays message to Staff :  1. ‘Please select one draft document to update.’  2. ‘Can not select multiple draft documents to update.’ |

* 1. **View published document**

1. **Stationary**
   1. **Manage published document**
   2. **Process transfer published document**
   3. **Proccess convert draft to published document**
2. **Manager**
   1. **Approve, return, remove draft**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Approve, return, remove draft | |
| **Actors** | Manager | |
| **Description** | Manager receive draft from his/her Staff to approve. | |
| **Requirements** |  | |
| **Pre-conditions** | Staff must create a draft, then submit to his/her Manager. Manager must log in. | |
| **Post-conditions** | Success: Draft is forwarded to Top Manager to approve.  Fail: Draft is not sent to Top Manager. | |
| **Basic flow** | Actor Actions: Actor clicks “Draft” on menu then system’s default go to “Processing” tab, then click draft which sent from his/her Staff to open view. System show detail view this draft.  **[Basic 1]**  After deciding to approve then click “Approve”.  **[Basic 2]**  After deciding to refuse and need to edit then click “Return”.  **[Basic 3]**  After deciding to refuse and remove it then click “Remove”. | System Responses:  **[Basic 1] [Basic 2]**  System shows pop-up information about his/her action. He/she should fill description about his/her submit.  **[Basic 3]**  System shows pop-up about confirm action remove. |
| **Alternative flow** | Actor Actions:  **[Alternative 1]**   * Click “OK” to submit draft to Top Manager. * Click “Cancel” nothing change.   **[Alternative 2]**   * Click “OK” to return draft to his/her Staff. * Click “Cancel” nothing change.   **[Alternative 3]**   * Click “OK” to remove. * Click “Cancel” nothing change. | System Responses:  After all action “OK” successfully. System redirect to page “Draft” 🡪 “Processing” tab view list. If not successfully or “Cancel” action is clicked then nothing change. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor leave description field blank. | System Responses:  - Actor can not submit next step. |

* 1. **Process transfer draft**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Process transfer draft | |
| **Actors** | Manager | |
| **Description** | Manager receive published document from Stationary. | |
| **Requirements** |  | |
| **Pre-conditions** | Stationary create a published document and transfer to related department. Manager must log in. | |
| **Post-conditions** | Success: Published document is forwarded to Staff to process.  Fail: Published document is not sent to Staff. | |
| **Basic flow** | Actor Actions: Actor clicks “Published document” on menu then system’s default go to “Processing” tab, then click document which sent from Stationary to open view. System show detail view this document.  **[Basic 1]**  After deciding to transfer then click “Transfer” to related Staff.  **[Basic 2]**  If document type is “need is to reply”. Then Manager transfer to a Staff. After Staff finished, he/she submit to his/her Manager to approve to Stationary. | System Responses:  **[Basic 1]**  System shows pop-up information about his/her action. He/she should choose Staff(s) and fill description about his/her submit.  **[Basic 2]**  System shows pop-up information about his/her action. He/she should choose one Staff and fill description about his/her submit. |
| **Alternative flow** | Actor Actions:  **[Alternative 1]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. * Click “Cancel” nothing change.   **[Alternative 2]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. And wait to Staff submit his/her action. * Click “Cancel” nothing change. | System Responses:  After all action “OK” successfully. System redirect to page “Published Document” 🡪 “Processing” tab view list. If not successfully or “Cancel” action is clicked then nothing change. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor no check Staff to receive.  [Exception 2]  2. Actor leave description field blank. | System Responses:  - Choose at least one Staff to transfer  - Actor could not submit next step. |

* 1. **Process transfer published document**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Process transfer published document | |
| **Actors** | Manager | |
| **Description** | Manager receive published document from Stationary. | |
| **Requirements** |  | |
| **Pre-conditions** | Stationary create a published document and transfer to related department. Manager must log in. | |
| **Post-conditions** | Success: Published document is forwarded to Staff to process.  Fail: Published document is not sent to Staff. | |
| **Basic flow** | Actor Actions: Actor clicks “Published document” on menu then system’s default go to “Processing” tab, then click document which sent from Stationary to open view. System show detail view this document.  **[Basic 1]**  After deciding to transfer then click “Transfer” to related Staff.  **[Basic 2]**  If document type is “need is to reply”. Then Manager transfer to a Staff. After Staff finished, he/she submit to his/her Manager to approve to Stationary. | System Responses:  **[Basic 1]**  System shows pop-up information about his/her action. He/she should choose Staff(s) and fill description about his/her submit.  **[Basic 2]**  System shows pop-up information about his/her action. He/she should choose one Staff and fill description about his/her submit. |
| **Alternative flow** | Actor Actions:  **[Alternative 1]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. * Click “Cancel” nothing change.   **[Alternative 2]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. And wait to Staff submit his/her action. * Click “Cancel” nothing change. | System Responses:  After all action “OK” successfully. System redirect to page “Published Document” 🡪 “Processing” tab view list. If not successfully or “Cancel” action is clicked then nothing change. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor no check Staff to receive.  [Exception 2]  2. Actor leave description field blank. | System Responses:  - Choose at least one Staff to transfer  - Actor could not submit next step. |

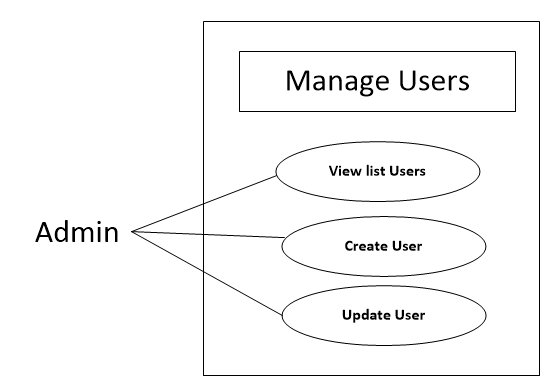
* 1. **Process reply on published document**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Process reply on published document | |
| **Actors** | Manager | |
| **Description** | Manager receive published document from Stationary. | |
| **Requirements** | Type of document must be “need to reply”. | |
| **Pre-conditions** | Manager must receive submit by his/her Staff. Manager must log in. | |
| **Post-conditions** | Success: Published document is forwarded to Stationary to process.  Fail: Published document is not sent to Stationary. | |
| **Basic flow** | Actor Actions: Actor clicks “Published document” on menu then system’s default go to “Processing” tab, then click document which sent from his/her Staff to open view. System show detail view this document.  **[Basic 1]**  After deciding to transfer then click “Transfer” to related Staff.  **[Basic 2]**  If document type is “need is to reply”. Then Manager transfer to a Staff. After Staff finished, he/she submit to his/her Manager to approve to Stationary. | System Responses:  **[Basic 1]**  System shows pop-up information about his/her action. He/she should choose Staff(s) and fill description about his/her submit.  **[Basic 2]**  System shows pop-up information about his/her action. He/she should choose one Staff and fill description about his/her submit. |
| **Alternative flow** | Actor Actions:  **[Alternative 1]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. * Click “Cancel” nothing change.   **[Alternative 2]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. And wait to Staff submit his/her action. * Click “Cancel” nothing change. | System Responses:  After all action “OK” successfully. System redirect to page “Published Document” 🡪 “Processing” tab view list. If not successfully or “Cancel” action is clicked then nothing change. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor no check Staff to receive.  [Exception 2]  2. Actor leave description field blank. | System Responses:  - Choose at least one Staff to transfer  - Actor could not submit next step. |

1. **Top manager**
   1. **Process approve, not approve, return draft**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Process approve, not approve, return draft | |
| **Actors** | Manager | |
| **Description** | Manager receive published document from Stationary. | |
| **Requirements** | Type of document must be “need to reply”. | |
| **Pre-conditions** | Manager must receive submit by his/her Staff. Manager must log in. | |
| **Post-conditions** | Success: Published document is forwarded to Stationary to process.  Fail: Published document is not sent to Stationary. | |
| **Basic flow** | Actor Actions: Actor clicks “Published document” on menu then system’s default go to “Processing” tab, then click document which sent from his/her Staff to open view. System show detail view this document.  **[Basic 1]**  After deciding to transfer then click “Transfer” to related Staff.  **[Basic 2]**  If document type is “need is to reply”. Then Manager transfer to a Staff. After Staff finished, he/she submit to his/her Manager to approve to Stationary. | System Responses:  **[Basic 1]**  System shows pop-up information about his/her action. He/she should choose Staff(s) and fill description about his/her submit.  **[Basic 2]**  System shows pop-up information about his/her action. He/she should choose one Staff and fill description about his/her submit. |
| **Alternative flow** | Actor Actions:  **[Alternative 1]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. * Click “Cancel” nothing change.   **[Alternative 2]**   * Click “transfer” with following checked Staff in his/her Department. Then click “OK” to transfer. And wait to Staff submit his/her action. * Click “Cancel” nothing change. | System Responses:  After all action “OK” successfully. System redirect to page “Published Document” 🡪 “Processing” tab view list. If not successfully or “Cancel” action is clicked then nothing change. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor no check Staff to receive.  [Exception 2]  2. Actor leave description field blank. | System Responses:  - Choose at least one Staff to transfer  - Actor could not submit next step. |

1. **IT**
   1. **Manage user**

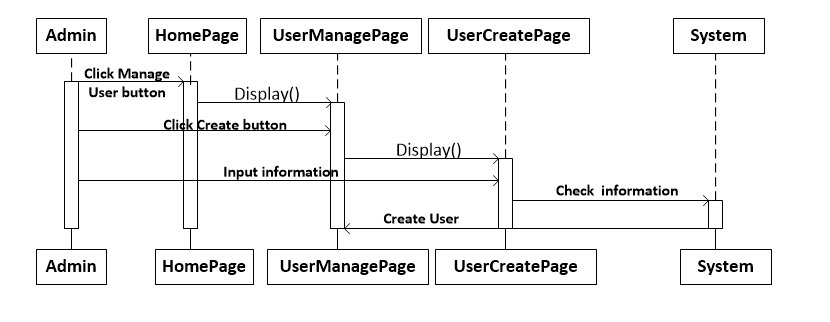
****

* 1. **View List, Detail user**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View list user | |
| **Actors** | Admin | |
| **Description** | Actor can view list user account | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Return manage list user page | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Manage User’ button in Actor’s Home page menu.  3.Actor click Reset button  5. Actor click Fullname link | System Responses:  2. System redirects to UserManage page and displays List user form with the following control : - ‘FullName’ link  - ‘UserName’ label  - ‘Email’ label  - ‘Department’ label  - ‘Reset’ button  - ‘Update’ button  4. Confirm after System reset password of this user to ‘123456’  6. System redirects to UserDetail page with the following control :  - ‘UserName’ label  - ‘FullName’ label  - ‘Birthday’ label  - ‘Gender’ label  - ‘Email’ label  - ‘Role’ label  - ‘Department’ label  - ‘status’ label  - ‘OK’ Button |

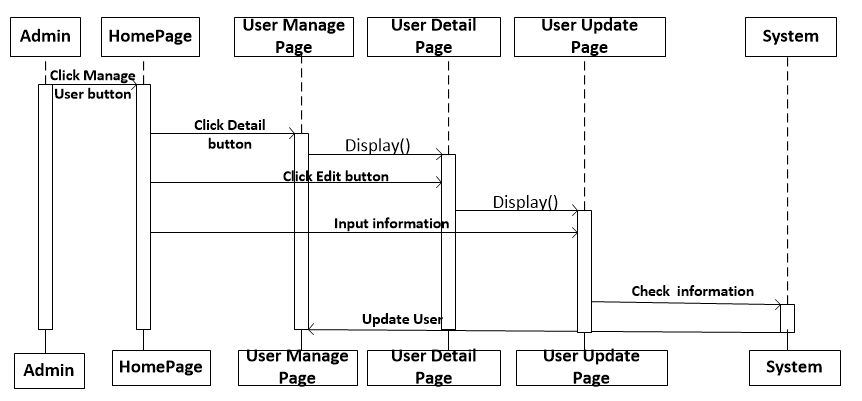
* 1. **Create user**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Create user | |
| **Actors** | Admin | |
| **Description** | Actor can Create user account | |
| **Requirements** | Actor provides profile information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Return UserManage page  Fail: Output fail | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Create user’ button in Actor’s UserManage page  3. Actor inputs user’s information, and click ‘Create’ button. [Alternative 1] | System Responses:  2. System redirects to UserCreate page and displays Create user form with the following control :  - ‘FullName’ text field  - ‘Gender’ checkbox group  - ‘Birthday’ datebox  - ‘UserName’ text field  - ‘Password’ text field  - ‘Password Confirm’ text field  - ‘Email’ text field  - ‘Role’ Combobox  - ‘Department’ Combobox  - ‘Save’ button  - ‘Cancel’ Button  4. System checks the information  [Exception 1]  5. System inserts the account into database with default status is true, show message: ‘Create Account Success’ and then redirects UserManage page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back’ button. | System Responses:  2. System returns to UserManage page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘UserName’ text field bank.  2. UserName has already exists.  3. Actor lefts input an invalid UserName.  4. Actor lefts ‘New Password’ text field blank.  5. Actor lefts ‘New Confirm Password’ text field blank.  6. New Password and New Password Confirm don’t match.  7. Actor lefts ‘Full Name’ text field blank.  8. Actor lefts ‘Birthday’ datebox blank.  9.Input age to ‘Birthday’ datebox less than 18.  10. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘Username is exists’.  - ‘UserName is invalid’.  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match Password.’  - ‘FullName cannot be blank’.  - ‘Birthday cannot be blank’.  - ‘Age must over 18’.  - ‘Email is invalid or is exists’. |

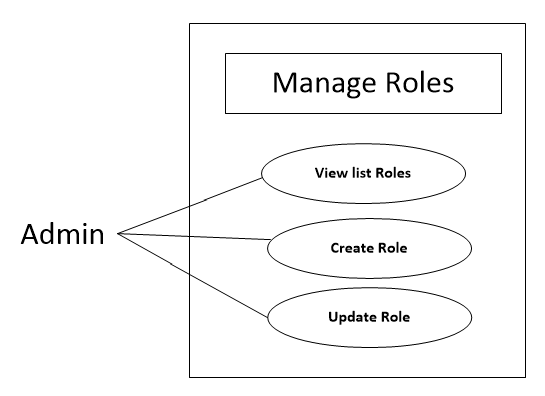
****

* 1. **Update user**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update user’s information | |
| **Actors** | Admin | |
| **Description** | Admin can update user’s information | |
| **Requirements** | Actor provides account’s information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Update Success.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Edit’ button in UserDetail page.  3. Actor input information and click ‘Update’ button. [Alternative 1] | System Responses:  2. System redirects to UserUpdate page and displays Update form with the following control :  - ‘FullName’ text field  - ‘Gender’ checkbox group  - ‘Birthday’ datebox  - ‘UserName’ text field  - ‘Email’ text field  - ‘Role’ Combobox  - ‘Department’ Combobox  4. System checks the information  [Exception 1]  5. System update the account into database and then redirects to UserManage page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back to Home’ button. | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  7. Actor lefts ‘Full Name’ text field blank.  9. Actor lefts input an invalid email address.  9. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘FullName cannot be blank’.  - ‘Email is invalid or is exists’. |

****

**2.Manage Role**

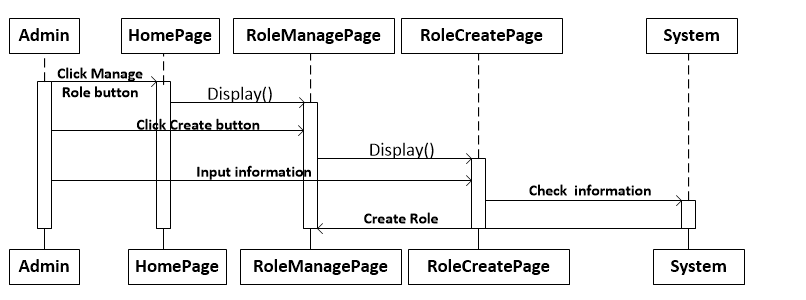
****

* 1. **View list Role**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View list user | |
| **Actors** | Admin | |
| **Description** | Actor can view list user account | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Return RoleManage page | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Manage Role’ button in Actor’s Home page. | System Responses:  2. System redirects to RoleManage page and displays List Role with the following control :  - ‘Role’ label  - ‘Description’ label  - ‘Status’(Enable) label  - ‘Edit’ button |

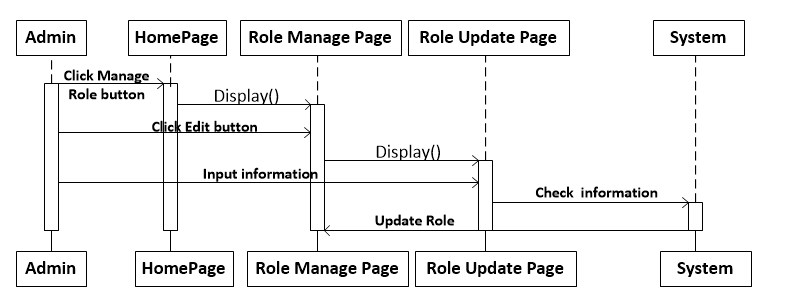
* 1. **Create Role**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Create Role | |
| **Actors** | Admin | |
| **Description** | Actor can Create Role on department | |
| **Requirements** | Actor provides Role’s information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Return manage list user page  Fail: Output fail | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Create Role’ button in Actor’s CreateRole page  3. Actor inputs Role’s information, and click ‘Create’ button. [Alternative 1] | System Responses:  2. System redirects to RoleCreate page and displays Create user form with the following control :  - ‘RoleName’ text field  - ‘Description’ textarea  - ‘Create’ Button  4. System checks the information  [Exception 1]  5. System inserts the Role into database, show message: ‘Create Role Success’ and then redirects RoleManage page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back’ button. | System Responses:  2. System returns to RoleManage page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘RoleName’ text field bank.  2. RoleName has already exists. | System Responses:  System displays message to Actor :  - ‘RoleName cannot be blank’.  - ‘RoleName is exists’ |

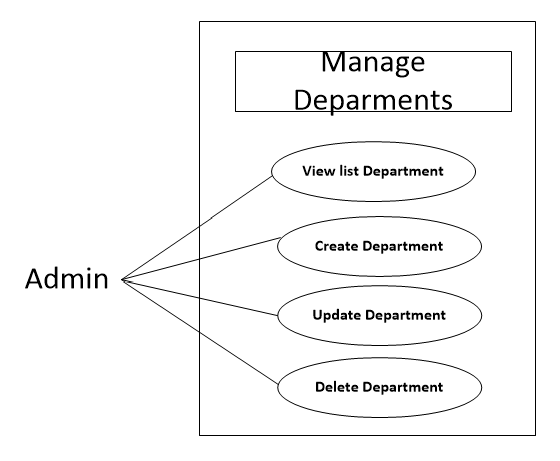
****

* 1. **Update Role**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Role’s information | |
| **Actors** | Admin | |
| **Description** | Admin can update Role’s information | |
| **Requirements** | Actor provides Role’s information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Update Success then redirects to RoleManage page.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Edit’ button in RoleManage page.  3. Actor input information and click ‘Update’ button. [Alternative 1] | System Responses:  2. System redirects to RoleUpdate page and displays Update form with the following control :  - ‘RoleName’ text field  - ‘Department’ listbox  - ‘Description’ textarea  - ‘Update’ Button  4. System checks the information  [Exception 1]  5. System update the account into database and then redirects to RoleManage page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back’ button. | System Responses:  2. System returns to RoleManage page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘RoleName’ text field bank.  2. RoleName has already exists.  3. | System Responses:  System displays message to Actor :  - ‘RoleName cannot be blank’.  - ‘RoleName is exists’ |

****

**3.Manage Department**

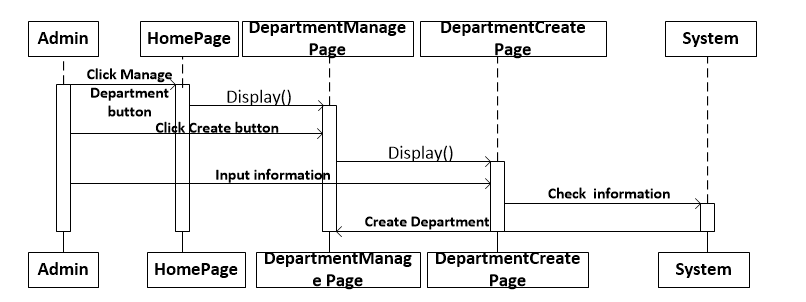
****

**3.1 View list Department**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | View list Department | |
| **Actors** | Admin | |
| **Description** | Actor can view list Department | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Return Department Manage page | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Manage Department’ button in Actor’s Home page. | System Responses:  2. System redirects to Department Manage page and displays Deparment List with the following control :  - ‘Department’ label  - ‘status’(Enable) label  - ‘Edit’ button |

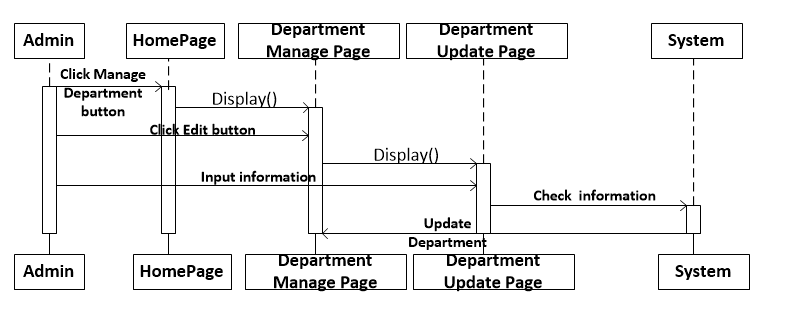
**3.2 Create Department**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Create Department | |
| **Actors** | Admin | |
| **Description** | Actor can Create Department | |
| **Requirements** | Actor provides Department’s information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Return manage list Department page  Fail: Output fail | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Create Department’s’ button in Actor’s Create Department page  3. Actor inputs Department’s information, and click ‘Create’ button. [Alternative 1] | System Responses:  2. System redirects to CreateDepartment page and displays Create Department form with the following control :  - “Department’s Name” text field  - ‘Create’ Button  4. System checks the information  [Exception 1]  5. System inserts the Department into database, show message: ‘Create Department Success’ and then redirects Department Manage page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back’ button. | System Responses:  2. System returns to Department Manage page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘Department Name’ text field bank.  2. Department Name has already exists. | System Responses:  System displays message to Actor :  - ‘Department Name cannot be blank’.  - ‘Department Name is exists’ |

****

* 1. **Update Department**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Update Department’s information | |
| **Actors** | Admin | |
| **Description** | Admin can update Department’s information | |
| **Requirements** | Actor provides Department’s information | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Update Success then redirects to Department Manage page.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Edit’ button in Department Manage page.  3. Actor input information and click ‘Update’ button. [Alternative 1] | System Responses:  2. System redirects to Department Update page and displays Update form with the following control :  - ‘Department Name’ text field  - ‘Status’(Enable) Combobox  - ‘Update’ Button  4. System checks the information  [Exception 1]  5. System update the Department into database and then redirects to DepartmentManage page. |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Back’ button. | System Responses:  2. System returns to DepartmentManage page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor lefts ‘DepartmentName’ text field bank.  2. DepartmentName has already exists. | System Responses:  System displays message to Actor :  - ‘DepartmentName cannot be blank’.  - ‘DepartmentName is exists’ |

****

1. **Entity Relationship Diagram**

User

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | **user\_Id** | uniqueidentifier |  | No |  |  | User’s Id |
|  | user\_Name | varchar | 20 | No |  |  | UserName |
|  | user\_Password | varchar | 200 | No | 123456 |  | User’s Password |
|  | user\_CreateDate | datetime | 8 | No |  |  | Create date user |
|  | user\_UpdateDate | datetime | 8 | Yes |  |  | Update date user |
|  | user\_Gender | tinyint | 1 | No | 0 |  | User’s gender:  0: Female  1: Male |
|  | user\_DOB | varchar | 10 | No |  |  | User’s Date Of Birth |
|  | user\_FullName | nvarchar | 100 | No |  |  | User’s Full Name |
|  | user\_Status | tinyint | 1 | No | 1 |  | User’s status:  0: Disable  1: Enable |
|  | user\_Email | varchar | 100 | No |  |  | User’s Email |
| FK | dep\_Id | uniqueidentifier |  | No |  | Department.dep\_Id | Department Id, FK of Department Table |
| FK | Role\_Id | uniqueidentifier |  | No |  | Role.role\_Id | Role Id, FK of Role Table |

Action

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | **act\_Id** | tinyint | 1 | No |  |  | Action’s Id |
|  | act\_Name | nvarchar | 100 | Yes |  |  | Action’s Name |
|  | act\_Description | nvarchar | 150 | Yes |  |  | Action’s Description |

Application

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | **app\_Id** | uniqueidentifier |  | No |  |  | Application’s Id |
|  | app\_Name | nvarchar | 100 | Yes |  |  | Application’s Name |
|  | app\_IsActive | Bit | 1 | Yes | 1 |  | Application’s |
|  | app\_CreateDate | Datetime | 8 | Yes |  |  | Create date Application |
|  | app\_IsWorkFlow | bit | 1 | Yes |  |  |  |
|  | app\_Type | tinyint | 1 | Yes |  |  |  |
|  | app\_isVisible | bit | 1 | Yes | 1 |  |  |

Document

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | doc\_Id | uniqueidentifier |  | No |  |  | Document’s Id |
|  | doc\_Number | varchar | 50 | Yes |  |  | Document’s Number |
|  | doc\_SourceNumber | varchar | 50 | Yes |  |  | Document’s SourceNumber |
|  | doc\_UpdateDate | datetime | 8 | Yes |  |  | Update date document |
|  | doc\_CreateDate | datetime | 8 | Yes |  |  | Create date document |
| FK | user\_Id | uniqueidentifier |  | Yes |  | User.user\_Id | User’s Id, FK of User Tables |
| FK | docType\_Id | varchar | 30 | Yes |  | DocumentType.docType\_Id | DocumentType’s Id, FK of Document Table |
|  | doc\_Content | nvarchar | 500 | Yes |  |  | Document’s content |
|  | doc\_Status | tinyint | 1 | Yes |  |  | Document’s Status:  0:  1: |
| FK | app\_Id | uniqueidentifier |  | Yes |  | Application.app\_Id | Application’s Id, FK of Application Table |
|  | doc\_IsValid | bit | 1 | Yes |  |  |  |
|  | doc\_ValidFrom | varchar | 10 | Yes |  |  | Valid From date of document |
|  | doc\_ValidTo | varchar | 10 | Yes |  |  | Valid To date of document |
|  | doc\_Date | varchar | 10 | Yes |  |  | Document’s date |
|  | doc\_Type | tinyint | 1 | Yes |  |  | Document’s type:  0:  1: |

DocumentDepartment

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | docDep\_Id | uniqueidentifier |  | No |  |  | DocumentDeparment’s Id |
| FK | dep\_Id | uniqueidentifier |  | No |  | Department.dep\_Id | Department’s Id, FK of Department Table |
|  | docDep\_IsActive | bit | 1 | Yes | 1 |  |  |
| FK | docDetail\_Id | uniqueidentifier |  | No |  | DocumentDetail.docDetail\_Id | DocumentDetail’s Id, FK of DepartmentDetail Table |

DocumentDetail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | docDetail\_Id | uniqueidentifier |  | No |  |  | DocumentDetail’s Id |
| FK | doc\_Id | uniqueidentifier |  | Yes |  | Document.doc\_Id | Document’s Id, FK of Document Table |
|  | docDetail\_UserCreate | uniqueidentifier |  | Yes |  |  | User create Document Detail |
|  | docDetail\_DepCreate | uniqueidentifier |  | Yes |  |  | Department Create Document Detail |
|  | docDetail\_UserReceive | uniqueidentifier |  | Yes |  |  | User Receive Document Detail |
|  | docDetail\_DepReceive | uniqueidentifier |  | Yes |  |  | DocumentDetail’s Department Receive |
|  | docDetail\_FileName | varchar | 150 | Yes |  |  | documentDetail’s File Name |
|  | docDetail\_FileContent | image |  | Yes |  |  | DocumentDetail’s File Content |
| FK | workFlow\_Id | uniqueidentifier |  | Yes |  | WorkFlow.workFlow\_Id | workFlow’s Id, FK of WorkFlow Table |
|  | docDetail\_CreateDate | datetime | 8 | Yes |  |  | Date Create of Document Detail |
|  | docDetail\_Description | nvarchar | 500 | Yes |  |  | DocumentDetail’s Description |
| FK | act\_Id | tinyint | 1 | Yes |  | Action.act\_Id | Action’s Id, FK of Action Table |
|  | docDetail\_IsActive | bit | 1 | Yes |  |  |  |
|  | docDetail\_UpdateDate | datetime | 8 | Yes |  |  | Date Update of DocumentDetail |

DocumentStaff

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | docStaff\_Id | uniqueidentifier |  | No |  |  | DocumentStaff’s Id |
| FK | docDep\_Id | uniqueidentifier |  | No |  | DocumentDepartment.docDep\_Id | DocumentDepartment’s Id,FK of DocumentDepartment Table |
| FK | user\_Id | uniqueidentifier |  | No |  | User.user\_Id | User’Id, FK of User Table |
|  | docStaff\_IsActive | bit | 1 | Yes | 1 |  |  |

DocumentStorage

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | docSto\_Id | uniqueidentifier |  | No |  |  | DocumentStorage’s Id |
| FK | doc\_Id | uniqueidentifier |  | No |  | Document.doc\_Id | Document’s Id, Fk of Document Table |
| FK | user\_Id | uniqueidentifier |  | No |  | User.user\_Id | User’Id, FK of User Table |
|  | docSto\_CreateDate | datetime | 8 | No |  |  | Date Create of DocumentStorage |

DocumentType

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | docType\_Id | varchar | 30 | No |  |  | DocumentType’s Id |
|  | docType\_Name | nvarchar | 50 | Yes |  |  | DocumentType’s Name |

Role

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | **role\_Id** | uniqueidentifier |  | No |  |  | Role’s Id |
|  | role\_Name | varchar | 20 | No |  |  | Role’s Name |
|  | role\_Description | nvarchar | 250 | Yes |  |  | Role’s Description |
|  | role\_CreateDate | datetime | 8 | No |  |  | Date Create of Role |

Department

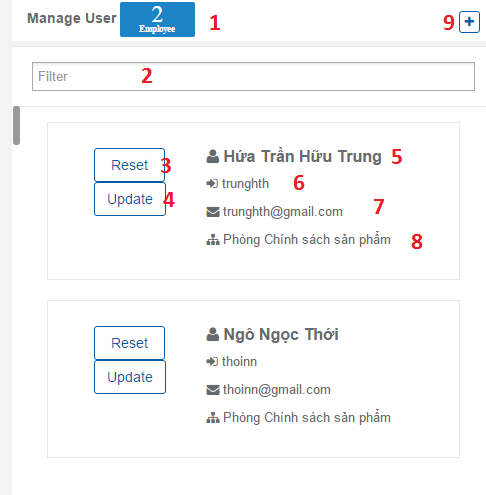
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | **dep\_Id** | uniqueidentifier |  | No |  |  | Department’s Id |
|  | dep\_Name | nvarchar | 100 | No |  |  | Department’s Name |
|  | dep\_Status | tinyint | 1 | No | 1 |  | Department’s Status:  0: Disable  1: Enable |
|  | dep\_Code | nvarchar | 10 | Yes |  |  | Department’s Code |

WorkFlow

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Length** | **Allow Null** | **Default value** | **Table-Field-Ref** | **Description** |
| **PK** | workFlow\_Id | uniqueidentifier |  | No |  |  | workFlow’s Id |
|  | workFlow\_Name | nvarchar | 200 | Yes |  |  | workFlow’s Name |
|  | workFlow\_Step | tinyint |  | Yes |  |  | workFlow’s step |
| FK | role\_Id | uniqueidentifier |  | Yes |  | Role.role\_Id | Role’s Id, FK of Role Table |
|  | workFlow\_Type | tinyint | 1 | Yes |  |  | workFlow’s Type |
| FK | app\_Id | uniqueidentifier |  | No |  | Application.app\_Id | Application’s Id, FK of Application Table |
|  | workFlow\_IsAllowReturn | bit | 1 | No |  |  |  |
|  | workFlow\_IsAllowRemove | bit | 1 | No |  |  |  |
|  | workFlow\_IsAllowFinish | bit | 1 | No |  |  |  |
|  | workFlow\_IsAllowUpload | bit | 1 | No |  |  |  |
|  | workFlow\_IsSetViewed | bit | 1 | No |  |  |  |
|  | workFlow\_IsSubmit | bit | 1 | No |  |  |  |
|  | workFlow\_IsTransferMultiple | bit | 1 | No |  |  |  |
|  | workFlow\_IsGenerateNumber | bit | 1 | Yes |  |  |  |
|  | workFlow\_CreateDate | datetime | 8 |  |  |  |  |

1. **Database Relationship Diagram**
2. **Table Definition**
3. **Gui Design**

List User

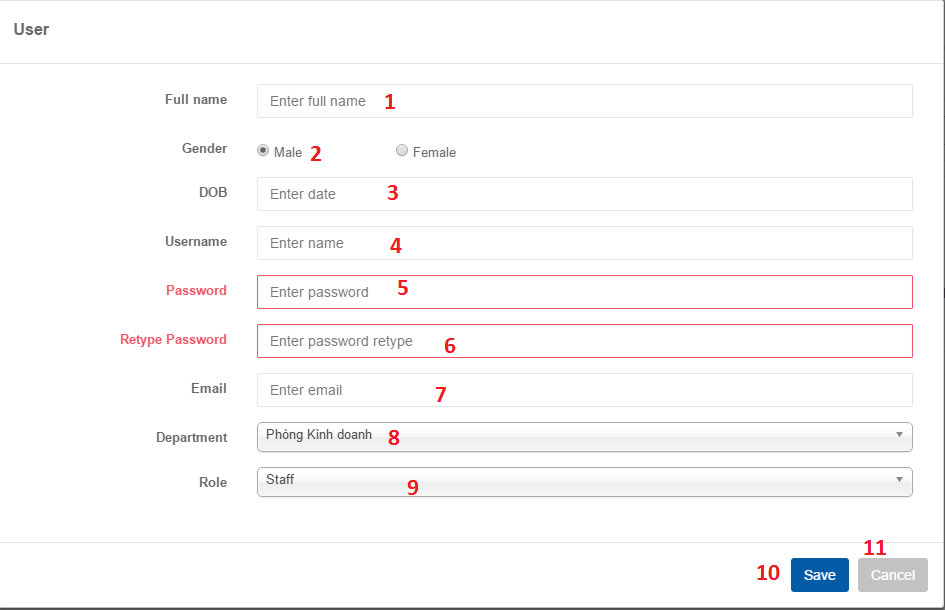


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lblCount | Label |  |  | Count size account |  |
| 2 | txtFilter | Text field |  | On change | Filter account |  |
|  | GridList Of User |  |  |  | Grid contains controls: 3,4,5,6,7,8  data get from user table |  |
| 3 | btnReset | Button |  | Click | Confirm Reset password this account |  |
| 4 | Btnpdate | Button |  | Click | Redirect to update user page |  |
| 5 | lblFullname | Label |  |  | User’s full name |  |
| 6 | lblUsername | Label |  |  | User’s user name |  |
| 7 | lblEmail |  |  |  | User’s email |  |
| 8 | lblDepartment |  |  |  | User’s Department |  |
| 9 | btnCreate | Button |  | Click | Redirect to UserCreate page |  |

Detail user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lblUserName | Label |  |  | User’s username |  |
| 2 | lblFullName | Label |  |  | User’s fullname |  |
| 3 | lblEmail | Label |  |  | User’s email |  |
| 4 | lblGender | Label |  |  | User’s gender |  |
| 5 | lblBirhday | Label |  |  | User’s DOB |  |
| 6 | lblDateCreate | Label |  |  | User’s date create |  |
| 7 | lblDateUpdate | Label |  |  | User’s date update |  |
| 8 | lblRole | Label |  |  | User’s role |  |
| 9 | lblDepartment | Label |  |  | User’s department |  |
| 10 | lblStatus | Label |  |  | User’s status |  |
| 11 | btnEdit | Button |  | Click | Redirect to Update user page |  |

Create user

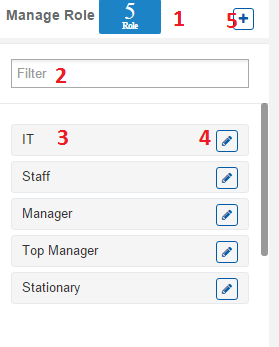


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtFulName | Text field | Not null  Length <= 100 |  | Input User’s full name |  |
| 2 | RGender | Radio Button |  |  | Choose User’s gender |  |
| 3 | txtBirthday | Text field | Age must greater than 18 |  |  |  |
| 4 | txtUserName | Text field | Not null  Not exists  5<= length <= 20 |  | Input User’s username |  |
| 5 | txtPassword | Text field | Not null  5<= length <= 200 |  | Input User’s Password |  |
| 6 | txtConfirmPass | Text field | txtConfirmPass and txtPassword Confirm is match |  | Input Confirm Password |  |
| 7 |  |  |  |  |  |  |
| 8 | txtEmail | Text field | Not null  Length <= 100 |  | Input User’s email |  |
| 9 | cmbDepartment | Combobox | Must selected a Department | Click | Choose User’s Department |  |
| 10 | cmbRole | Combobox | Must selected a Role | Click | Choose User’s Role |  |
| 11 | btnCreate | Button |  | Click | Check information after create user |  |
| 12 | btnCancel | Button |  | Click | cancel |  |

Update user

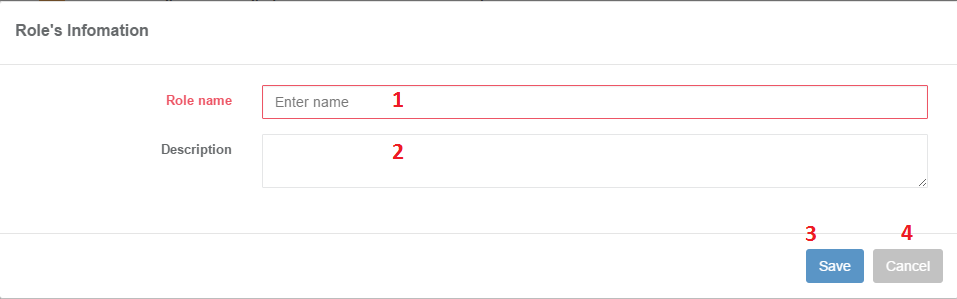
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtFulName | Text field | Not null  Length <= 100 |  | Input User’s full name |  |
| 2 | RGender | Radio Button |  |  | Choose User’s gender |  |
| 3 | txtBirthday | Text field | Age must greater than 18 |  |  |  |
| 4 | txtUserName | Text field | Not null  Not exists  5<= length <= 20 |  | Input User’s username |  |
| 5 | txtPassword | Text field | Not null  5<= length <= 200 |  | Input User’s Password |  |
| 6 | txtConfirmPass | Text field | txtConfirmPass and txtPassword Confirm is match |  | Input Confirm Password |  |
| 7 |  |  |  |  |  |  |
| 8 | txtEmail | Text field | Not null  Length <= 100 |  | Input User’s email |  |
| 9 | cmbDepartment | Combobox | Must selected a Department | Click | Choose User’s Department |  |
| 10 | cmbRole | Combobox | Must selected a Role | Click | Choose User’s Role |  |
| 11 | btnCreate | Button |  | Click | Check information after create user |  |
| 12 | btnCancel | Button |  | Click | cancel |  |

List Role



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lblCount | Label |  |  | Count size Role |  |
| 2 | txtFilter | Text field |  | On change |  |  |
|  | GridList Of Role |  |  |  | Grid contains controls: 3,4  data get from Role table |  |
| 3 | lblRole | Label |  |  | Role’s Name |  |
| 4 | btnEdit | Button |  |  | Redirect to Update Role page |  |
| 5 | btnCreate | Button |  | Click | Redirect to Create Role page |  |

Create Role

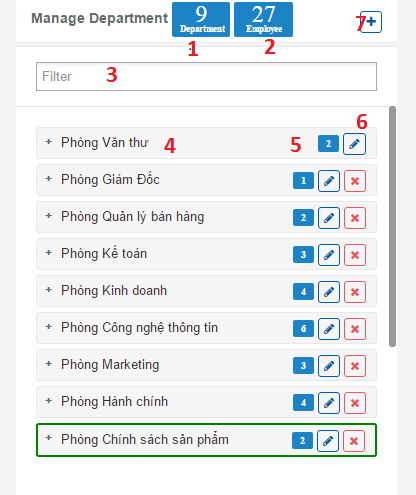


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtName | Text field | Not null  Not exists  5<= length <= 20 |  | Input Role’s Name |  |
| 2 | txtDescription | Text field | Length <= 250 |  | Input Role’s Desceiption |  |
| 3 | btnCreate | Button |  | Click | Check information after create role |  |
| 4 | btnCancel | Button |  | Click | cancel |  |

Update Role

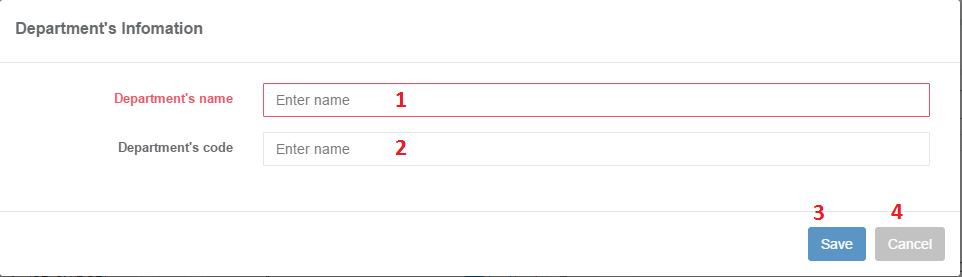
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtName | Text field | Not null  Not exists  5<= length <= 20 |  | Input Role’s Name |  |
| 2 | txtDescription | Text field | Length <= 250 |  | Input Role’s Desceiption |  |
| 3 | btnCreate | Button |  | Click | Check information after create role |  |
| 4 | btnCancel | Button |  | Click | cancel |  |

List Department



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | lblCountDep | Label |  |  | Count size Department |  |
| 2 | lblCountEmp | Label |  |  | Count all Employee on all Department |  |
| 3 | txtFilter | Text field |  | On change | Filter Department |  |
|  | GridList Of Department |  |  |  | Grid contains controls: 4,5,6  data get from Department table |  |
| 4 | lblName | Label |  |  | Department’s Name |  |
| 5 | btnEdit | Button |  | Click | Redirect to update Department page |  |
| 6 | btnDelete | Button |  | Click | Delete Department |  |
| 7 | btnCreate | Button |  | Click | Redirect to Create Department page |  |
| 7 | btnCreate | Button |  |  | Redirect to Department Create page |  |
| 4 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Create Department



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtName | Text field | Not null  Not exists  5<= length <= 100 |  | Input Department’s name |  |
| 2 | txtDescription | Text field |  |  | Input Department’s Description |  |
| 2 | btnCreate | Button |  | Click | Check information after create Department |  |
| 3 | btnCancel | Button |  | Click | cancel |  |

Update Department

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtName | Text field | Not null  Not exists  5<= length <= 100 |  | Input Department’s name |  |
| 2 | txtDescription | Text field |  |  | Input Department’s Description |  |
| 2 | btnCreate | Button |  | Click | Check information after create Department |  |
| 3 | btnCancel | Button |  | Click | cancel |  |

List Type of Document

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | btnCreate | Button |  |  | Redirect to Type Create page |  |
| 2 | GridList Of Type |  |  |  | Grid contains controls: 3,4,5,6  data get from DocumentType table |  |
| 3 | lblName | Label |  |  | Type’s Name |  |
| 6 | btnEdit | Button |  | Click | Redirect to update Type page |  |

Create Type of Document

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtName | Text field | Not null  Not exists  5<= length <= 50 |  | Input Type’s name |  |
| 2 | btnCreate | Button |  | Click | Check information after create Type of Document |  |
| 3 | btnCancel | Button |  | Click | cancel |  |

Update Type of Document

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Type** | **Validation** | **Event** | **Description** | **Status** |
| 1 | txtName | Text field | Not null  Not exists  5<= length <= 50 |  | Input Type’s name |  |
| 3 | btnUpdate | Button |  | Click | Check information after update Type |  |
| 4 | btnCancel | Button |  | Click | cancel |  |

**Tasksheet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref.No: 4** | | **Project Title:**  **Document Mangement System** | **Date of Preparation of Activity Plan** | | | |
| **Sr. No.** | **Task** | | **Actual Start Date** | **Actual Days** | **Team Member Names** | **Status** |
| 01 | Introduction | | Dec, 12, 2015 | 6 | All Members | Completed |
| 02 | Exiting Scenario | | All Members | Completed |
| 03 | Customer Requirement Specification | | All Members | Completed |
| 04 | Functional Requirement Specification | | All Members | Completed |
| 05 | System requirements, Development software and Technology | | All Members | Completed |
| 06 | Tasksheet | | All Members | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 4 | Approved By: Faculty |
| Date: Dec, 2015 | Team Leader  **Hua Tran Huu Trung** |  |