|  |  |
| --- | --- |
|  | FPT ACADEMY INTERNATIONAL  FPT – APTECH COMPUTER EDUCATION |

Center Name: ACE-HCMC-2-FPT

Address: 590 Cach Mang Thang Tam Street, District 3, Ho Chi Minh City, Viet Nam



Online Help Desk

Design Document

|  |  |  |  |
| --- | --- | --- | --- |
| Faculty: | | Mr. Tran Phuoc Sinh |  |
| Batch No: | | T11708A0 |  |
| Group No: | | 07 |  |
| Student: |  | Full Name | Roll No. |
|  | 1. | NGUYEN HOANG TU | STUDENT1054235 |
|  | 2. | NGUYEN THANH NAM | STUDENT1057514 |
|  | 3. | TRAN VAN THANG | STUDENT1053909 |

October, 2018

Mr. Nguyen Hoang Tu

Mr. Nguyen Thanh Nam

Mr. Tran Van Thang



This is to certify that

have successfully designed and

developed 

Submitted by:

Date of Issue: \_\_\_\_\_\_\_\_

Authorized Signature: \_\_\_\_\_\_\_\_\_\_\_



# Table of Contents

[Table of Contents 2](#_Toc529651976)

[I. Acknowledgments 6](#_Toc529651977)

[II. Problem Definition 7](#_Toc529651978)

[1. Introduction 7](#_Toc529651979)

[2. Existing Scenario 7](#_Toc529651980)

[3. Requirement Specification 8](#_Toc529651981)

[1.1. Administrator 8](#_Toc529651982)

[1.2. Facility Heads (Staffs) 8](#_Toc529651983)

[1.3. Assignees (Staffs) 8](#_Toc529651984)

[1.4. End-user 9](#_Toc529651985)

[4. Hardware / Software Requirements 9](#_Toc529651986)

[III. Task sheet review 1 10](#_Toc529651987)

[I. Architecture & Design of the Project 12](#_Toc529651988)

[1. Presentation Tier: 13](#_Toc529651989)

[2. Business Logic Tier: 13](#_Toc529651990)

[3. Data Access Tier: 13](#_Toc529651991)

[II. Algorithms - Data Flowchart: 13](#_Toc529651992)

[Symbol generates: 13](#_Toc529651993)

[1. Login process: 14](#_Toc529651994)

[2. Log out: 15](#_Toc529651995)

[3. Create new user (Admin only): 16](#_Toc529651996)

[4. View list of user account (Admin): 17](#_Toc529651997)

[5. Search specific user account: 17](#_Toc529651998)

[6. View detail, block/unblock & delete user (Admin only): 18](#_Toc529651999)

[18](#_Toc529652000)

[18](#_Toc529652001)

[7. Reset to default password for specific user (Admin only): 19](#_Toc529652002)

[19](#_Toc529652003)

[8. Create new facility (Facilities head): 20](#_Toc529652004)

[9. Block/unblock & delete facility (Facilities head): 21](#_Toc529652005)

[10. View list of all requests (Facilities head): 22](#_Toc529652006)

[11. Send request to assignee (Facilities head): 22](#_Toc529652007)

[12. Close/Reject request (Facilities head & assignee): 23](#_Toc529652008)

[13. Create request (End-user): 24](#_Toc529652009)

[14. Cancel request (End-user): 24](#_Toc529652010)

[III. Data flow diagram (DFD): 25](#_Toc529652011)

[1. Context level diagram: 26](#_Toc529652012)

[2. Level 1 DFD: 27](#_Toc529652013)

[3. Level 2 DFD: 28](#_Toc529652014)

[3.1. Administrator manages end-user accounts: 28](#_Toc529652015)

[3.2. Administrator manages staff accounts: 29](#_Toc529652016)

[3.3. Administrator handles registration accounts: 30](#_Toc529652017)

[3.4. Facilities head (staff) manages facilities: 30](#_Toc529652018)

[3.5. Facilities head (staff) manages request: 31](#_Toc529652019)

[3.6. Assignee (staff) manages request: 31](#_Toc529652020)

[3.7. End-user manages request: 31](#_Toc529652021)

[IV. Use Case Diagram 32](#_Toc529652022)

[1. Administrator Use Case: 33](#_Toc529652023)

[2. Facilities Head Use Case: 34](#_Toc529652024)

[3. Assignees Use Case: 35](#_Toc529652025)

[4. End-user Use Case: 36](#_Toc529652026)

[V. Sequence Diagram 37](#_Toc529652027)

[1. Login 40](#_Toc529652028)

[2. Logout 42](#_Toc529652029)

[Change Profile 43](#_Toc529652030)

[3. Insert( Admin) 45](#_Toc529652031)

[4. Update (Admin) 47](#_Toc529652032)

[5. Delete ( Admin) 49](#_Toc529652033)

[6. Delete User ( Admin) 51](#_Toc529652034)

[7. Student ( User) send request 53](#_Toc529652035)

[VI. Entity Relationship (E-R) Diagram 55](#_Toc529652036)

[Entity 55](#_Toc529652037)

[E-R Diagram: 58](#_Toc529652038)

[VII. Class Diagram 59](#_Toc529652039)

[VIII. Task sheet review 2 60](#_Toc529652040)

REVIEW I

# **Acknowledgments**

We would like to acknowledge all those who have given support and help us make the project a success.

We wish to express our deep gratitude to all teachers who have been devoting their lives to teach us how to stand-alone and walk ahead.

We are grateful to our families as well as our friends who take care and encourage us even though we are successful or failed. They never leave us alone and always look forward to us when we are on any road of the life.

We are much thankful to the entire staff and chairpersons at the Head Office of FPT – Aptech Centre who have been organizing and looking after our studying course.

There are no words to show our appreciation for our teacher, Mr Tran Phuoc Sinh, who worked day by day to teach and guide us to complete this project.

Finally, we would like to offer many thanks to all our schoolfellows for their valuable suggestions.

We would like to thank sincerely!

Group 7 – FPT Aptech.

# **II. Problem Definition**

## 1. Introduction

Perfect Technological Innovation is one of the largest university of technology in the world. Its campus is very huge with many facilities. Therefore, students, staff and people who want to manage and use these facilities have a big difficulty. That leads to the urgent demand to build an application which helps them to feel convenient and comfortable in conducting these material bases.

## 2. Existing Scenario

This project is aimed at developing an Online Help Desk (OHD) for the facilities in the Perfect Technological Innovation’s campus. This is an Intranet based application that can be accessed throughout the campus. This system can be used to automate the workflow of service requests for the various facilities in the campus. This is one integrated system that covers different kinds of facilities like class-rooms, labs, hostels, mess, canteen, gymnasium, computer centre, faculty club etc. Registered users (students, faculty, lab-assistants and others) will be able to log in a request for service for any of the supported facilities. These requests will be sent to the concerned people, who are also valid users of the system, to get them resolved. There are features like email notifications/reminders, addition of a new facility to the system, report generators etc in this system.

## 3. Requirement Specification

### Administrator

Administrator will have managing abilities:

* Manage user accounts (creating new user account, editing user info, blocking user on accessing this application, supporting in restoring password etc).

### Facility Heads (Staffs)

Facility heads will have abilities:

* Login to the system through the first page of the application.
* Manage their information (View/Update their info and change password).
* See the list of the requests created by students.
* Send these requests to assignees who have responsibility to handle them.
* See the list of requests (both open and closed) sent by him/her to assignees over the past.
* Update request.
* Manage facilities (adding new facilities, deleting/block facilities).

### Assignees (Staffs)

Assignees will have abilities:

* Login to the system through the first page of the application.
* Manage their information (View/Update their info and change password).
* View the requests sent from facility heads.
* Change the status of the request (work in progress, close or reject)
* See the list of requests sent to them.
* Get help about Online Help Desk (OHD) System on how to use the different features of the system.

### End-user

Students have abilities:

* Login to the system through the first page of the application.
* Manage their information (Update their info and change password).
* Create a new request by specifying the facility, the severity of the request (there may be several levels of severity defined) and a brief description of the request
* See the status of the requests create by him/her (the status could be one of unassigned/assigned/work in progress/closed or rejected).
* Close a request created by him/her by giving an appropriate reason.
* Get help about Online Help Desk (OHD) System on how to use the different features of the system.

## 4. Hardware / Software Requirements

#### Software

|  |
| --- |
| * Visual Studio .Net / ASP * Sql Server Management Studio 2012 * .Net Framework 3.5+ * Web Browser(Chrome,Edge, Internet Explorer) |

#### Hardware

|  |
| --- |
| * A minimum computer system that will help you access all the tools in the courses is a Pentium 166 or better * 1GB Megabytes of RAM or better * Hardware 5GB or better |

# **Task sheet review 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. No: 4 | Project Title:  Railway Reservation Manage System | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Acknowledgment | Oct 24,2018 | 2 | All Members | Completed |
| 02 | Problem Definition | All Members | Completed |
| 03 | Customer Requirement | All Members | Completed |
| 05 | Hardware/Software | All Members | Completed |
| 06 | Task sheet | All Members | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 7 | Approved By: Faculty |
| Date: Oct 26, 2018 | Team Leader  Nguyen Hoang Tu | Tran Phuoc Sinh |

REVIEW II

# **Architecture & Design of the Project**



## Presentation Tier:

Is the tier in which the users interact with application . Presentation Tier contents Model, View, Controller used to receive a request and response to User.

Technology: ASP.NET MVC4, Razor, HTML, CSS, JavaScript, Ajax, JQUERY, Twitter Bootstrap

## Business Logic Tier:

Is mainly working as the bridge between Data Tier and Presentation Tier. All the Data passes through the Business Tier before passing to the Presentation Tier.

Technology: OOP, ASP.NET

## Data Access Tier:

Is basically the server which stores all the application’s data .Data tier contents Database Tables, Database Views and other means of storing Application Data .

Technology: SQL Server, LINQ, ADO.NET

# Algorithms - Data Flowchart:

## Symbol generates:



## Login process:

Start

Input Username & Password

Show error message

Validate exist?

Not exist

failure

Exist

Check level of account ?

End-user/staff level

Home page

End

Admin level

Admin page

## Log out:

Start

Homepage

Clear Session

Confirm?

End

yes

no

## Create new user (Admin only):

Start

Input information

Check username duplication ?

exist

Show error message

Not exist

Generate and encode default password

save

Data

base

End

## View list of user account (Admin):

Start

End

Show list of users

Get information from database

## Search specific user account:

Start

Input username

Show error message

Check username exist

Not exist

exist

Show information

End

## View detail, block/unblock & delete user (Admin only):

Start

Get specific user

Unblock user

Block user

Delete user

View user details

Confirm action ?

no

## 

yes

# 

Data

base

End

## Reset to default password for specific user (Admin only):

## 

no

save

yes

Generate and encode default password

Confirm action ?

Reset user’s password

Get specific user

Data

base

End

Start

## Create new facility (Facilities head):

Start

Input information

Check username duplication ?

Exist

Show error message

Not exist

Data

base

End

## 

## Block/unblock & delete facility (Facilities head):

Start

Confirm action ?

Get specific facility

Delete facility

Unblock facility

Block facility

no

yes

Data

base

Insert the reason

End

## View list of all requests (Facilities head):

End

Show list of all requests

Get information from database

Start

## Send request to assignee (Facilities head):

Start

Get specific request

Choose an assignee to send this request

no

Confirm action ?

yes

Data

base

End

## Close/Reject request (Facilities head & assignee):

Start

Get specific request

Close/Reject

Confirm action ?

no

yes

Insert the reason

Data

base

End

## 

## Create request (End-user):

Input information of request

Start

Confirm to send ?

no

yes

Data

base

End

## Cancel request (End-user):

Get specific request

Cancel

Start

no

Confirm to cancel ?

yes

Data

base

End

# Data flow diagram (DFD):

Define: Data Flows Diagram (DFD) describes the information flow in the system. The next step of system analysis is to consider in detail the information necessary for the implementation for functions discussed above and the one necessary for the improvement of the functions. Modelling tool frequently used for this purpose is DFD. DFD will support 4 main activities:

Analysis: DFD is used to determine requirement of users.

Design: DFD is used to map out plan and illustrate solution to analysis and users while designing a new system.

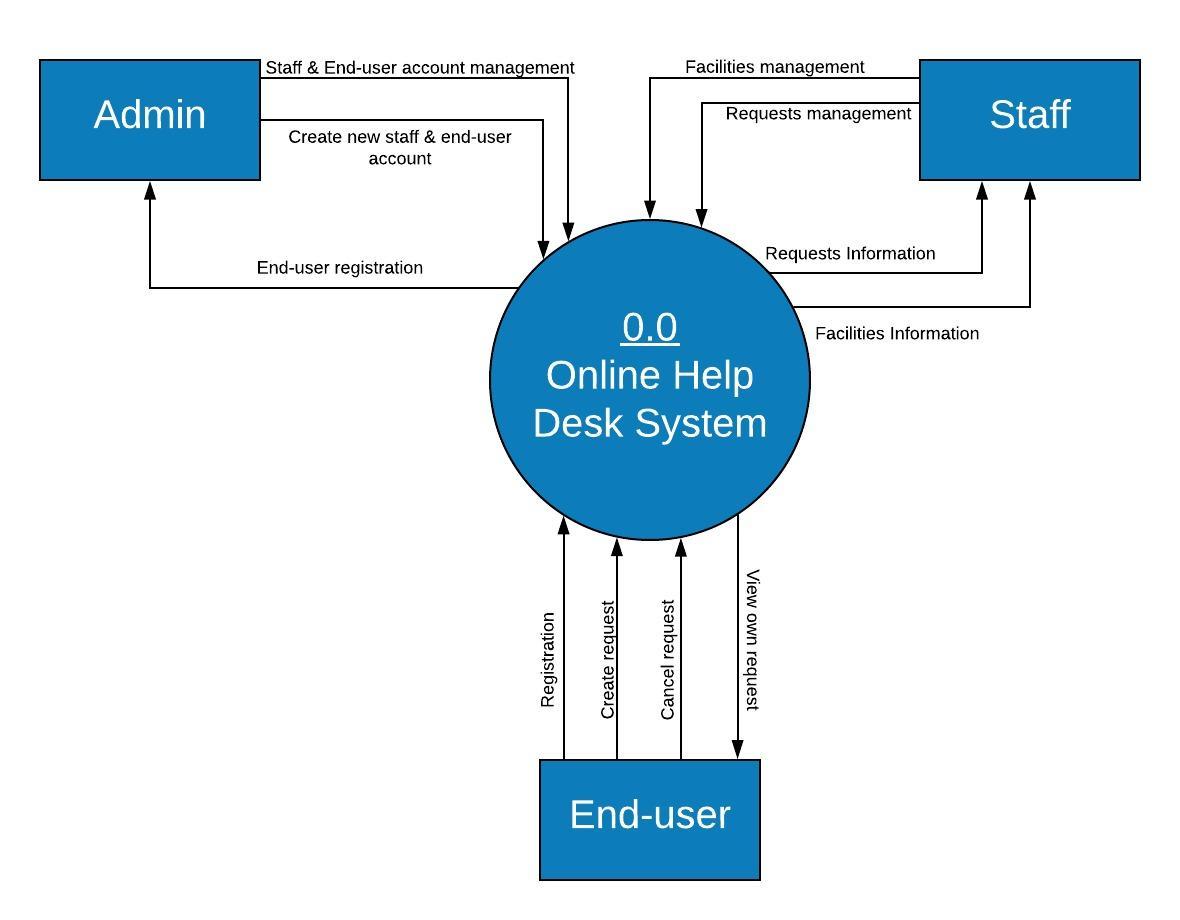
Communication: one of the strength of DFD is its simplicity and ease to understand to analysts and users;

Document: DFD is used to provide special description of requirement and system design. DFD provide an overview of key functional components of the system but it does not provide any detail on these components. We have to use other tools like database dictionary, process specification to get an idea of which information will be exchanged and how.

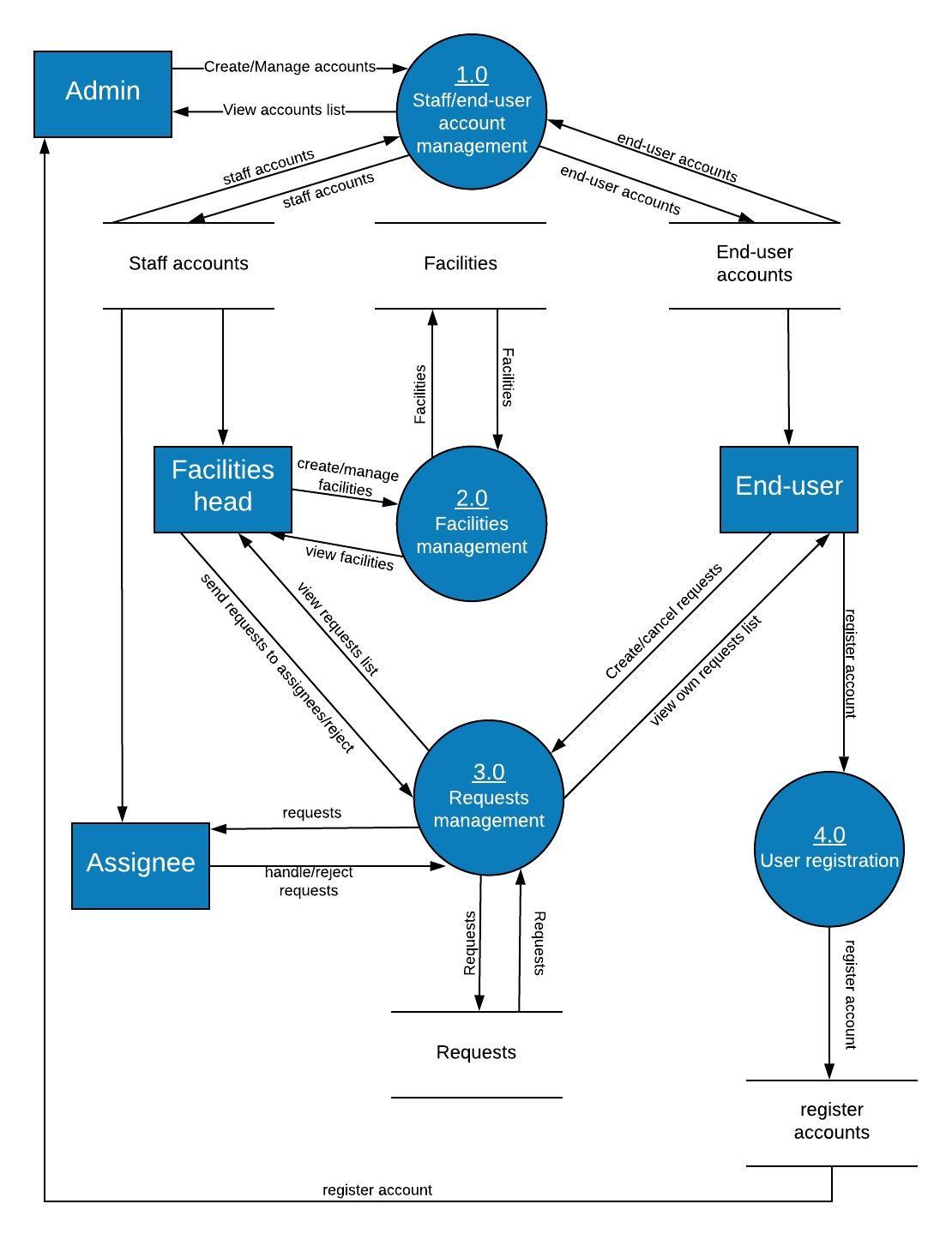
The main components of Context Diagram:

|  |  |  |
| --- | --- | --- |
|  | External | The external factors: External factors can be a person, a group of persons or an organization that are sources of information for the systems and are where system products are transferred to.  The process: Shows the common function of system  The data flow: Describe the movement of information from one part of the system to another.  The data store: The Data Store is used to model a collection of data packets at rest. A store is represented graphically by two parallel lines. The name of a Data Store that identifies the store is the plural of the name of the packets that are carried by flows into and out of the Data Store |
|  | The process |
|  | Data flow |
|  | Data store |

## Context level diagram:



## Level 1 DFD:



## Level 2 DFD:

### Administrator manages end-user accounts:

1.1  
Create

1.4  
Delete

1.3  
Block/ Unblock

1.2  
Update

Administrator

End-user accounts

### Administrator manages staff accounts:

2.1  
Create

2.4  
Delete

2.3  
Block/ Unblock

2.2  
Update

Administrator

Staff accounts

### Administrator handles registration accounts:

Registration accounts

Administrator

3.1  
Create

End-user accounts

### Facilities head (staff) manages facilities:

4.1  
Create

4.4  
Delete

4.3  
Block/ Unblock

4.2  
Update

Facilities head

Facilities

### Facilities head (staff) manages request:

5.2  
Reject

5.1  
send to assignee

Facilities head

Requests

### Assignee (staff) manages request:

6.2  
Reject

6.1  
Update status

Assignee

Requests

### End-user manages request:

7.2  
Cancel

7.1  
Create

Assignee

Requests

# Use Case Diagram



Online help desk system



Administrator

Facilities Head (Staff)

Assignee (Staff)

End-User

## Administrator Use Case:

<<extend>>

<<extend>>

<<extend>>

<<extend>>

<<extend>>



Administrator

## Facilities Head Use Case:

<<extend>>

<<extend>>

<<include>>

<<extend>>



<<extend>>

## Assignees Use Case:

<<extend>>

<<extend>>



Assignee

## End-user Use Case:

<<extend>>

<<extend>>



End-user

# Sequence Diagram

Class roles: describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

Object: Class

Activation: boxes represent the time an object needs to complete a task.



Object: Class

Object: Class

Actor

Activations

Messages: are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.



Object: Class

Object: Class

Actor

Messages

Lifelines: are vertical dashed lines that indicate the object's presence over time.



Object: Class

Object: Class

Actor

Lifelines

## Login



Login

Home Page

User

1: Request Login Page

2: Display Login Page

3: Input UserName,

Password

4:Process

5: Login Successful

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Login** | |
| **Actors** | Administrator, Staff, Passenger | |
| **Description** | Actors use this use case to login to system | |
| **Requirements** |  | |
| **Pre-Conditions** |  | |
| **Post-Condition** |  | |
| **Basic Flow** | Actors action  1. Actors click on “Login “ button  3. Actors enter username and password into form.  4. Actors click “Login”  button  [alternative] | System Response  2. The system display ”Login” page the following control:  - “Username” text field  - “Password” text field  - “Login” button  - “Cancel” button  5. The system validate Username and Password  6. The system log Actors into system. |
| **Alternative Flow** | Actors action  [alternative] Actors click Cancel button | System Response  System return to Homepage |
| **Exceptions** | Actors action  [exception1] Actors provide invalid Username and Password | System Response  1. System display message: “The username or password provided are incorrect” |

## Logout



Login

Login

User

1:Request Logout 2: Logout

3: Request to Login 4:Logged out

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Logout** | |
| **Actors** | Administrator, User | |
| **Description** | Actors use this use case to login to system | |
| **Requirements** |  | |
| **Pre-Conditions** | Already login into the System | |
| **Post-Condition** |  | |
| **Basic Flow** | Actors action  1. Actors click on “Logout “ button  3. Actors click “OK” button  [alternative] | System Response  2. System display message: “Do you want to logout?”  5. System logout Actors. |
| **Alternative Flow** | Actors action  [alternative] Actors click Cancel button | System Response  System return to previous page |
| **Exceptions** |  |  |

## Change Profile



Change

Change Profile Page

Form Change

User

1: Request change Profile 2: Get change Page

4: Display change Page 3: Display change Page

5: User enter update information of profile

6: Update information

8: If true, display message successful 7: Validate

If false, come back step 5 data

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Change Profile** | |
| **Actors** | Administrator, User | |
| **Description** | User change information | |
| **Requirements** | Login | |
| **Pre-Conditions** | Already login into the System | |
| **Post-Condition** | Logged in on website  Success: Information of User was updated  Fails: Display message error | |
| **Basic Flow** | Actors action  1. User select change profile    3. User enter new information  4. User click submit  6. Display message successful | System Response  2. System display Change Profile page  5. System validate entered information |
| **Alternative Flow** | Actors action  [alternative1] Actors click Cancel button | System Response  Back to the profile page |
| **Exceptions** | [exception1] Actor provide invalid information | System display message error |

## Insert( Admin)



Add New facilities Page

Manage facilities

Facilities

Admin

1: Request add new facilities

2: Get add new facilities Page

3: Display add new facilities Page

4: Display add new

facilities Page

5: Input facilities information 6: Create new facilities

7: Return result of validation Valid

8: If true, display message add new facilities successful

If false, come back step 5

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add new facilities** | |
| **Actors** | Administrator | |
| **Description** | Admin add new facilities | |
| **Requirements** | Login | |
| **Pre-Conditions** | Login by Admin | |
| **Post-Condition** | Success: Information of Item was add new  Fails: Display message error | |
| **Basic Flow** | Actors action  1. Actor click on “Insert”  3. Enter information of facilities, and click “Add” button  5. Display message successful | System Response  2. System display add new facilities form  4. System will validate info  6. New facilities was added to database (with ID is unique) |
| **Alternative Flow** | Actors action  [alternative1] Actors click Cancel button | System Response  Back to the previous page |
| **Exceptions** | [exception1] Actor provide invalid information | System display message error |

## Update (Admin)



Update facilities Page

Manage facilities

Facilities

Admin

1: Request update facilities

2: Find facilities

If not fount, 3: If fount facilities select

come back step1 to edit

6: Display edit

facilities Page 4: Get edit facilities Page

5: Display edit facilities Page

7: Input facilities information 8: Edit facilities

9: Return result of validation Valid

10: If true, display message edit facilities successful

If false, come back step 7

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update facilities** | |
| **Actors** | Administrator | |
| **Description** | Admin Update facilities | |
| **Requirements** | Login | |
| **Pre-Conditions** | Login by Admin | |
| **Post-Condition** | Success: information is updated into system database  Fails: Display message error | |
| **Basic Flow** | Actors action  1. Actor click on “Update”  3. Update new information of facilities  4. Click “Update” button [alternative1]  6. Display message successful | System Response  2. System display Update form  5. System will validate info [exception1]  7. System update new information to database |
| **Alternative Flow** | Actors action  [alternative1] Actors click Cancel button | System Response  Back to the previous page |
| **Exceptions** | [exception1] Actor provide invalid information | System display message error |

## Delete ( Admin)



Facilities

Manage facilities

Admin

1: Request delete facilities

2: Find facilities

If not fount, 3: If fount select facilities to delete

come back step 1

4: Delete facilities

5: Process

6: Return result of deletion

7: Display delete

success or not

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Delete Facilities** | |
| **Actors** | Administrator | |
| **Description** | Admin Delete facilities | |
| **Requirements** | Login | |
| **Pre-Conditions** | Login by Admin | |
| **Post-Condition** | Success: information of Facilities was deleted  Fails: Display message error | |
| **Basic Flow** | Actors action  1. Search facilities which want to search by Name or ID  3. select facilities which want to delete  4. Click “Delete” button  6. Display message successful | System Response  2. System display result of search  5. System display message to confirm  7. Facilities was deleted in database |
| **Alternative Flow** | Actors action  [alternative1] if result is empty, come back step 1  [alternative2] if Admin No select option, come back step 2 | System Response |
| **Exceptions** | [exception1] Cannot delete if it has some relative of foreign | System display message error |

## Delete User ( Admin)



Users

Manage users

Admin

1: Request delete users

2: Find users

If not fount, 3: If fount select users to delete

come back step 1

4: Delete users

5: Process

6: Return result of deletion

7: Display delete

success or not

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Delete Users** | |
| **Actors** | Administrator | |
| **Description** | Admin Delete Users | |
| **Requirements** | Login | |
| **Pre-Conditions** | Login by Admin | |
| **Post-Condition** | Success: information of Users was deleted  Fails: Display message error | |
| **Basic Flow** | Actors action  1. Search users which want to search by Name or ID  3. select users which want to delete  4. Click “Delete” button  6. Display message successful | System Response  2. System display result of search  5. System display message to confirm  7. Users was deleted in database |
| **Alternative Flow** | Actors action  [alternative1] if result is empty, come back step 1  [alternative2] if Admin No select option, come back step 2 | System Response |
| **Exceptions** | [exception1] Cannot delete if it has some relative of foreign | System display message error |

## Student ( User) send request



Request Page

Request

Home Page

Student

1: Request send “Request”

2: Select facilities

and write descriptions

3: Get data

4: Process

5: Return result request

6: Notifies send successful message

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Student send Request** | |
| **Actors** | Users | |
| **Description** | Student send Request | |
| **Requirements** | Login | |
| **Pre-Conditions** | Login by Student | |
| **Post-Condition** | Success: Request has been sent  Fails: Display message error | |
| **Basic Flow** | Actors action  1. Student select facilities and write descriptions  2. Click “Send” button  4. Display message successful | System Response  3. System display message to confirm  8. Request was created in database |
| **Alternative Flow** | Actors action  [alternative1] Actors click Cancel button | System Response  Back to the previous page |
| **Exceptions** | [exception1] Actor provide invalid information | System display message error |

# Entity Relationship (E-R) Diagram

## Entity

Staffs:

Staff

End-user:

End-user

Request:

Request

Admin:

Admin

StaffReply:

StaffReply

Register:

Register

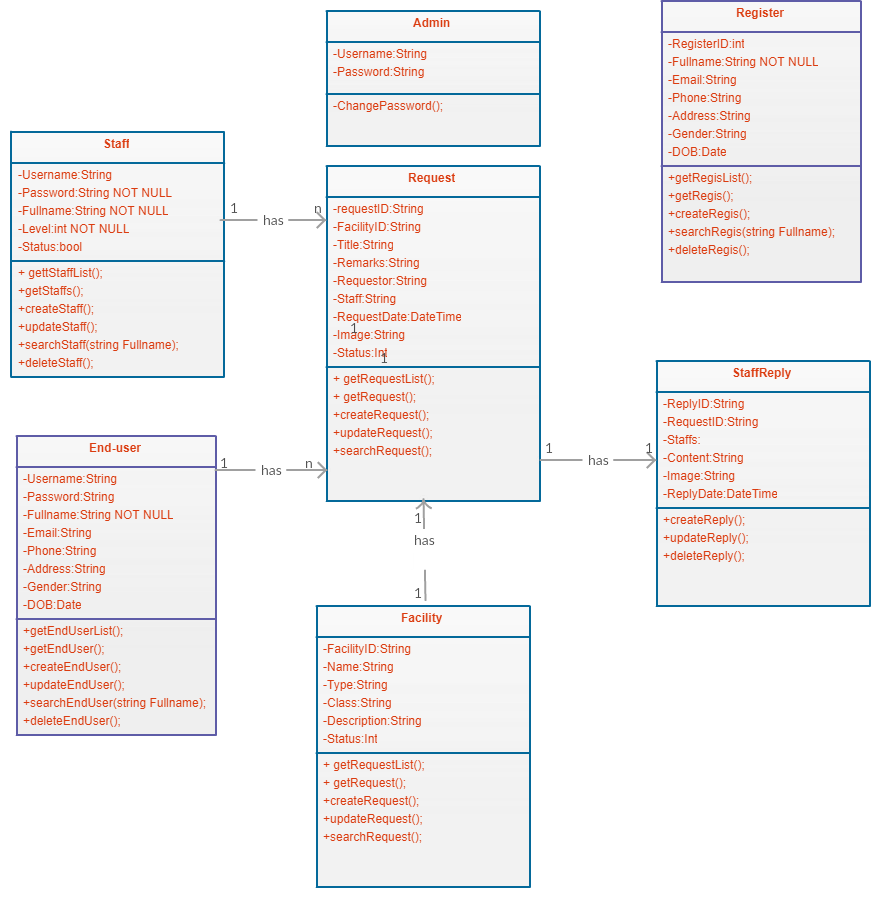
Facility:

Facility

## E-R Diagram:

|  |
| --- |
| Register  n  1  Has  Facility  1  1  1  1  n  1  1  Has  Has  Request  Has  StaffReply  Admin  Staffs  End-user |

# Class Diagram



# **Task sheet review 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. No: 4 | Project Title:  Railway Reservation Manage System | Date of Preparation of Activity Plan | | | |
| No. | Task | Actual Start Date | Actual Days | Team Member Names | Status |
| 01 | Architecture & Design of the Project | Oct ,28,2018 | 5 | All Members | Completed |
| 02 | Algorithms - Data Flowchart | All Members | Completed |
| 03 | USE CASE DIAGRAM | All Members | Completed |
| 05 | Sequence DIAGRAM | All Members | Completed |
| 06 | Entity Relationship (E-R) Diagram | All Members | Completed |
| 07 | Class DIAGRAM | All Members | Completed |
| 08 | Task Sheet | All Members | Completed |

|  |  |  |
| --- | --- | --- |
|  | Prepare By: Group 7 | Approved By: Faculty |
| Date: Nov,2, 2018 | Team Leader  Nguyen Hoang Tu | Tran Phuoc Sinh |

REVIEW III

# Database Design:

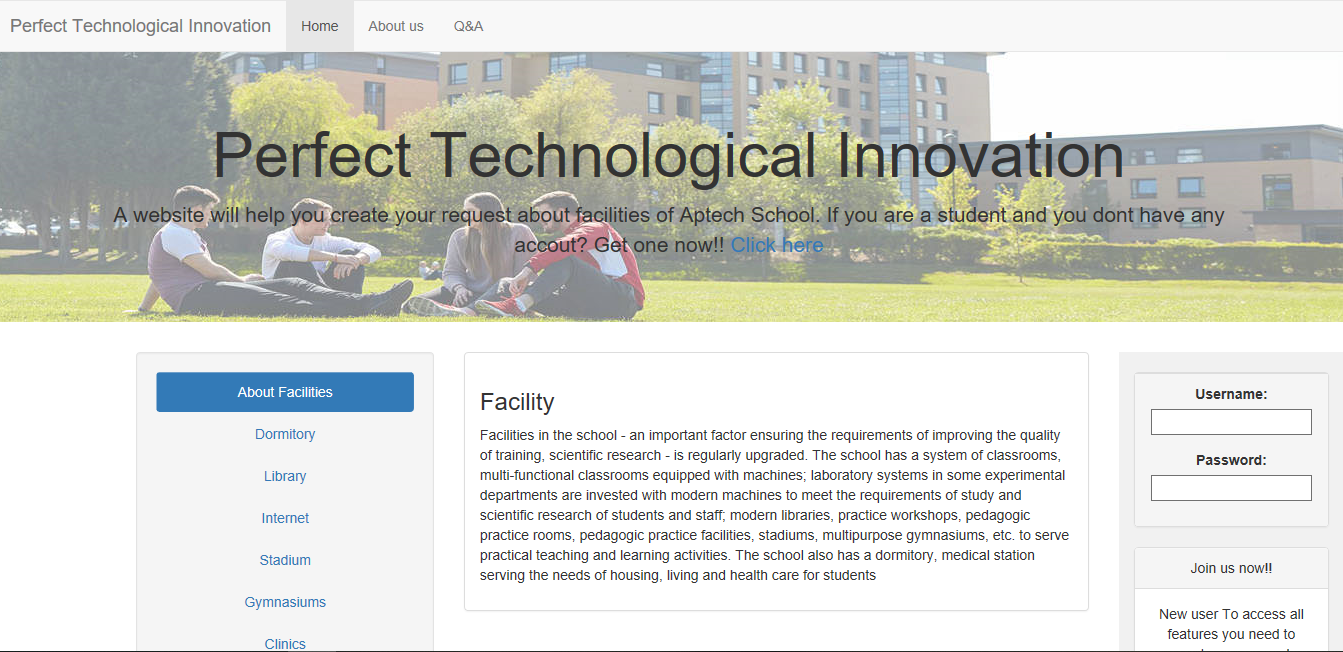
## Database design diagram:

## Database structure:

### 2.1

# GUI Designs – Front End:

## Home page:



12

11

10

9

8

7

6

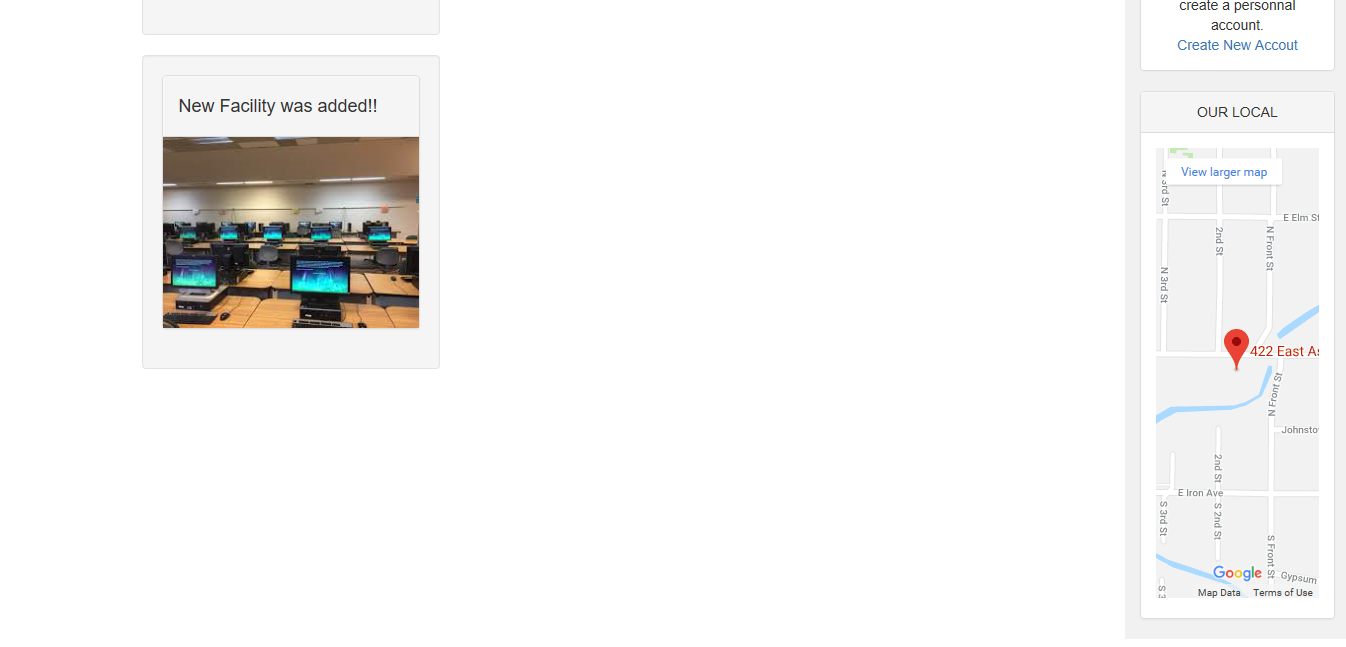
5

4

3

2

1



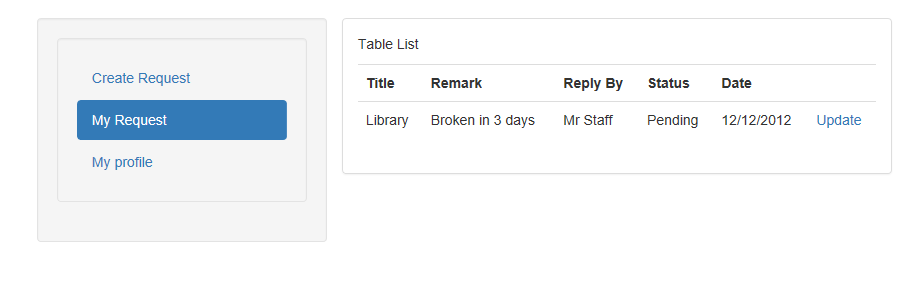
13

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Home page** | | | | | | |
| **No.** | Name | Type | Validate | Event | Description | Status |
| 1 | Home | Hyper Link |  | Click | Url to Home page action | Enable |
| 2 | About us | Hyper Link |  | Click | Url to Home page action | Enable |
| 3 | Q&A | Hyper Link |  | Click | Url to Home page action | Enable |
| 4 | About Facilities | Hyper Link |  | Click | Url to Home page action | Enable |
| 5 | Dormitory | Hyper Link |  | Click | Url to Home page action | Enable |
| 6 | Library | Hyper Link |  | Click | Url to Home page action | Enable |
| 7 | Internet | Hyper Link |  | Click | Url to Home page action | Enable |
| 8 | Stadium | Hyper Link |  | Click | Url to Home page action | Enable |
| 9 | Gymnasium | Hyper Link |  | Click | Url to Home page action | Enable |
| 10 | Clinic | Hyper Link |  | Click | Url to Home page action | Enable |
| 11 | Username | Text box | Not null | Click | Input username | Enable |
| 12 | Password | Text box | Not null | Click | Input password | Enable |
| 13 | Register | Hyper Link |  | Click | Show Register model | Enable |

## Register page:

## End-user logged in page:

### 3.1 Request list:



5

4

1

2

3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Home page** | | | | | | |
| **No.** | Name | Type | Validate | Event | Description | Status |
| 1 | Create request | Hyper Link |  | Click | Url to Home page action | Enable |
| 2 | My Requests | Hyper Link |  | Click | Url to Home page action | Enable |
| 3 | My Profile | Hyper Link |  | Click | Url to Home page action | Enable |
| 4 | Table List | Table |  |  | List of requests that end-user created | Enable |
| 5 | Update | Hyper Link |  | Click | Show update model | Enable |

### 3.2 Create new request:

### 

4

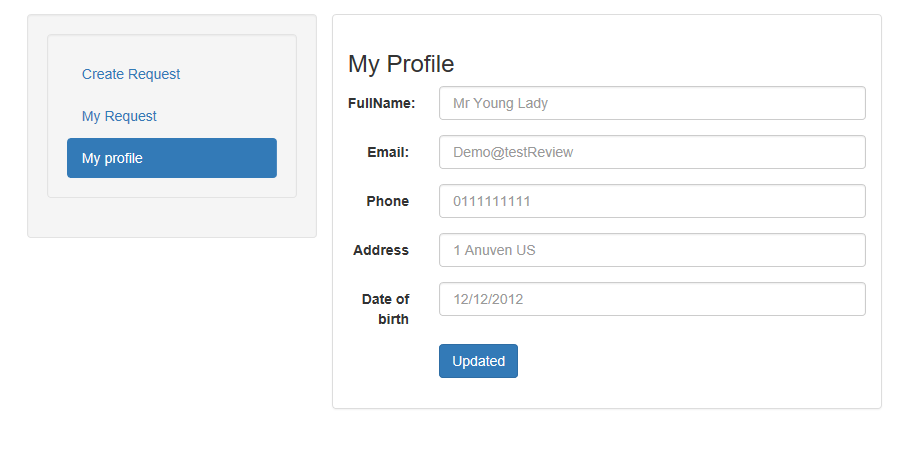
3

2

1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Home page** | | | | | | |
| **No.** | Name | Type | Validate | Event | Description | Status |
| 1 | Title | Combo box |  | Click | Show list of facility classes | Enable |
| 2 | Remark | Text area | Not null | Click | End-user types description for his/her request | Enable |
| 3 | Choose Image | Button |  | Click | Click to choose an image to make the request more detail | Enable |
| 4 | Submit | Button |  |  | Submit request | Enable |

### 3.3 End-user profile:



6

5

4

3

2

1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Home page** | | | | | | |
| **No.** | Name | Type | Validate | Event | Description | Status |
| 1 | Full name | Text box | Not null | Click | End-user full name | Enable |
| 2 | Email | Text box | Not null | Click | End-user email | Enable |
| 3 | Phone | Text box | Not null | Click | End-user phone number | Enable |
| 4 | Address | Text box | Not null | Click | End-user address | Enable |
| 5 | Date of birth | Text box | Not null | Click | End-user date of birth | Enable |
| 6 | Update | Button | Not null | Click | Update end-user profile | Enable |