

**IT HELP DESK SYSTEM**

|  |
| --- |
| Nguyễn Hải Long  Lê Mai Hồng  Lê Đức Huy  Nguyễn Quang Minh |

|  |
| --- |
| Supervised by **Nguyễn Trọng Tài** |

|  |  |
| --- | --- |
| No. SE04.B – 011 |  |

Contents

## Introduction

[Introduction 3](#_Toc332493364)

[Background 10](#_Toc332493365)

[Introduction 10](#_Toc332493366)

[Aims and Objective 11](#_Toc332493367)

[eSmartDesk – Why is it needed? 12](#_Toc332493368)

[Problem Definition 12](#_Toc332493369)

[Name of this Capstone Project 12](#_Toc332493370)

[Problem Abstract 12](#_Toc332493371)

[Project Overview 12](#_Toc332493372)

[The Current System 12](#_Toc332493373)

[The Proposed System 13](#_Toc332493374)

[Boundaries of the System 13](#_Toc332493375)

[Project Organization 13](#_Toc332493376)

[Software Process Model 13](#_Toc332493377)

[Roles and Responsibilities 14](#_Toc332493378)

[Tools and Techniques 15](#_Toc332493379)

[Project Management Plan 15](#_Toc332493380)

[Task 15](#_Toc332493381)

[Task Sheet: Assignments and Timetable 17](#_Toc332493382)

[Coding Convention 18](#_Toc332493383)

[User Requirement Specification 19](#_Toc332493384)

[System Requirement Specification 19](#_Toc332493385)

[External Interface Requirements 19](#_Toc332493386)

[User Interfaces 19](#_Toc332493387)

[Hardware Interfaces 20](#_Toc332493388)

[Software Interfaces 20](#_Toc332493389)

[Communications Protocol 20](#_Toc332493390)

[System Features 20](#_Toc332493391)

[System Feature – Issue Managing 22](#_Toc332493392)

[System Feature – User Managing 29](#_Toc332493393)

[System Feature – User Notifying 36](#_Toc332493394)

[System Feature – Report Managing 41](#_Toc332493395)

[System Feature – FAQs Managing 43](#_Toc332493396)

[Software System Attributes 50](#_Toc332493397)

[Reliability 50](#_Toc332493398)

[Availability 50](#_Toc332493399)

[Security 50](#_Toc332493400)

[Maintainability 50](#_Toc332493401)

[Portability 50](#_Toc332493402)

[Performance 50](#_Toc332493403)

[Entity Relationship Diagram 50](#_Toc332493404)

[Design Overview 51](#_Toc332493405)

[System Architecture Design 52](#_Toc332493406)

[Choice of System Architecture 52](#_Toc332493407)

[Discussion of Alternative Designs 53](#_Toc332493408)

[MVP (Model-View-Presenter) 53](#_Toc332493409)

[MVVM (Model-View-View-Model) 53](#_Toc332493410)

[Description of System Interface 54](#_Toc332493411)

[Component Diagram 55](#_Toc332493412)

[Detailed Description of Components 56](#_Toc332493413)

[Managing message 56](#_Toc332493414)

[Class Diagram 56](#_Toc332493415)

[Class Diagram Explanation 56](#_Toc332493416)

[Managing issue 59](#_Toc332493417)

[Class Diagram 59](#_Toc332493418)

[Class Diagram Explanation 59](#_Toc332493419)

[Mail notification 63](#_Toc332493420)

[Class Diagram 63](#_Toc332493421)

[Class Diagram Explanation 63](#_Toc332493422)

[Managing FAQs 65](#_Toc332493423)

[Class Diagram 65](#_Toc332493424)

[Class Diagram Explanation 67](#_Toc332493425)

[DTO Component 70](#_Toc332493426)

[Class Diagram 70](#_Toc332493427)

[Class Diagram Explanation 70](#_Toc332493428)

[Login Component 76](#_Toc332493429)

[Class Diagram 76](#_Toc332493430)

[Class Diagram Explanation 76](#_Toc332493431)

[Managing user 79](#_Toc332493432)

[Class diagram 79](#_Toc332493433)

[Class diagram explanation 79](#_Toc332493434)

[Managing Report 84](#_Toc332493435)

[Class diagram 84](#_Toc332493436)

[Class diagram explanation 84](#_Toc332493437)

[Sequence Diagram 86](#_Toc332493438)

[Managing Issue Function: 86](#_Toc332493439)

[Log Basic Issue: 86](#_Toc332493440)

[Log Advanced Issue: 86](#_Toc332493441)

[Search Issue: 87](#_Toc332493442)

[Edit Issue: 88](#_Toc332493443)

[View Detailed Issue: 88](#_Toc332493444)

[Managing User: 89](#_Toc332493445)

[Create new user: 89](#_Toc332493446)

[Search user: 90](#_Toc332493447)

[View user information: 91](#_Toc332493448)

[Edit user information: 91](#_Toc332493449)

[Managing Report: 92](#_Toc332493450)

[Produce report: 92](#_Toc332493451)

[Export to PDF: 93](#_Toc332493452)

[Managing message, email notification 94](#_Toc332493453)

[Send message 94](#_Toc332493454)

[View message 94](#_Toc332493455)

[Mail notification 95](#_Toc332493456)

[Managing FAQs 95](#_Toc332493457)

[View FAQs 95](#_Toc332493458)

[Create FAQS 96](#_Toc332493459)

[Search FAQs 96](#_Toc332493460)

[Edit FAQs 97](#_Toc332493461)

[User Interface Design 98](#_Toc332493462)

[Login 99](#_Toc332493463)

[Screen Image 99](#_Toc332493464)

[Objects and Actions 99](#_Toc332493465)

[Issue Managing 100](#_Toc332493466)

[Log Ticket: Basic Mode 100](#_Toc332493467)

[Log Advanced Issue 101](#_Toc332493468)

[Search Issue 102](#_Toc332493469)

[View/Edit Issue 103](#_Toc332493470)

[User Managing 104](#_Toc332493471)

[Create User 104](#_Toc332493472)

[Search User 105](#_Toc332493473)

[View/Edit User 106](#_Toc332493474)

[FAQ Managing 107](#_Toc332493475)

[Create FAQ 107](#_Toc332493476)

[Search FAQs 108](#_Toc332493477)

[View/Edit FAQ 109](#_Toc332493478)

[Message Managing 110](#_Toc332493479)

[Create Message 110](#_Toc332493480)

[My Message 111](#_Toc332493481)

[Report Managing 112](#_Toc332493482)

[Create Report 112](#_Toc332493483)

[Configuration 113](#_Toc332493484)

[Add Department/ IT Type 113](#_Toc332493485)

[Search Department/ IT Type 114](#_Toc332493486)

[Edit Department/ IT Type 115](#_Toc332493487)

[Database Design 116](#_Toc332493488)

[Introduction 116](#_Toc332493489)

[System Overview 116](#_Toc332493490)

[Test Approach 116](#_Toc332493491)

[Test Plan 117](#_Toc332493492)

[Features to be test 117](#_Toc332493496)

[Features not to be test 117](#_Toc332493497)

[Test tool and environment 117](#_Toc332493498)

[Test tool 117](#_Toc332493499)

[Test environment 118](#_Toc332493500)

[Test Cases 118](#_Toc332493501)

[GUI Test 118](#_Toc332493502)

[Login screen 118](#_Toc332493507)

[Dashboard screen 118](#_Toc332493508)

[Tickets 119](#_Toc332493509)

[My ticket screen 119](#_Toc332493510)

[Log ticket screen 119](#_Toc332493517)

[Search ticket screen 122](#_Toc332493518)

[View detailed ticket screen 122](#_Toc332493519)

[User 124](#_Toc332493520)

[Create user screen 124](#_Toc332493521)

[Search user screen 125](#_Toc332493529)

[View detailed user screen 126](#_Toc332493530)

[FAQs 127](#_Toc332493531)

[Log FAQ screen 127](#_Toc332493532)

[Search FAQ screen 128](#_Toc332493541)

[View detailed FAQ screen 128](#_Toc332493542)

[Report screen 129](#_Toc332493543)

[Message 130](#_Toc332493544)

[My messages screen 130](#_Toc332493545)

[Create message screen 131](#_Toc332493546)

[Main screen 131](#_Toc332493547)

[Upload function 132](#_Toc332493548)

[Auto complete function 132](#_Toc332493549)

[Checklists 133](#_Toc332493550)

[GUI Test 133](#_Toc332493552)

[Main screen 133](#_Toc332493553)

[My ticket screen 134](#_Toc332493559)

[Log ticket basic mode screen 135](#_Toc332493560)

[Log ticket advanced mode screen 135](#_Toc332493561)

[View detail ticket screen 136](#_Toc332493562)

[Search ticket screen 137](#_Toc332493563)

[Create user screen 139](#_Toc332493564)

[View detail user screen 139](#_Toc332493565)

[Search user screen 140](#_Toc332493566)

[Log FAQ screen 141](#_Toc332493567)

[Search FAQ screen 142](#_Toc332493568)

[View detail FAQ screen 142](#_Toc332493569)

[Create report screen 143](#_Toc332493570)

[My message screen 144](#_Toc332493571)

[Create message screen 145](#_Toc332493572)

[Login screen 146](#_Toc332493576)

[Dashboard screen 146](#_Toc332493581)

[Tickets 147](#_Toc332493582)

[My ticket screen 147](#_Toc332493583)

[Log ticket screen 148](#_Toc332493584)

[Search ticket screen 150](#_Toc332493585)

[View detailed ticket screen 151](#_Toc332493586)

[User 152](#_Toc332493587)

[Create user screen 152](#_Toc332493588)

[Search user screen 153](#_Toc332493589)

[View detailed user screen 154](#_Toc332493590)

[FAQ 156](#_Toc332493591)

[Log FAQ screen 156](#_Toc332493592)

[Search FAQ screen 156](#_Toc332493593)

[View detailed FAQ screen 157](#_Toc332493594)

[Report 158](#_Toc332493595)

[Message 159](#_Toc332493596)

[My message screen 159](#_Toc332493597)

[Create message screen 159](#_Toc332493598)

[Main screen 160](#_Toc332493599)

[Upload function 160](#_Toc332493600)

[Auto complete function 161](#_Toc332493601)

[Installation Guide 162](#_Toc332493602)

[User’s Guide 162](#_Toc332493603)

[Minimum System Requirements 162](#_Toc332493604)

[Introduction to Smart Desk 163](#_Toc332493605)

[Features and How to Use 163](#_Toc332493606)

[Tickets 163](#_Toc332493612)

[Users 168](#_Toc332493613)

[FAQs 171](#_Toc332493614)

[Reports 174](#_Toc332493615)

[Configurations 175](#_Toc332493616)

[Notifications 179](#_Toc332493617)

[Tables 181](#_Toc332493618)

[Figures 181](#_Toc332493619)

Introduction

# Background

Nowadays with the development of information technology, many companies and organizations have applied technologies into their business processes. IT help desk system is becoming almost a necessity for any organizations. This system help organizations provide information and support for users to solve issues with computers and other technology products.

Demand of using IT help desk system is growing rapidly. Many small and big companies trying to improve employee productivity have to consider the necessity of integrating good IT Help Desk system within their daily operations.

From those reasons, we want to develop an IT help desk system that may help organization manage, improve their information system and achieve predefinedgoals.

# Introduction

An IT help desk system offers fully integrated features to simplify, automate, and accelerate internal help desk and technical support processes inside an organization. Some of the key features of IT help desk system include:

* **Issue Management** - Efficiently record, track and resolve user request.
* **Self Service (normal user interface)** - End users can log in to a web based portal to submit issues. They can also access their existing issues, find solutions in the Knowledge Base, and track the status of all their requests.
* **Knowledge Base (FAQs)** - Flexible knowledge base with option to add unlimited articles and allows users to easily search the information.
* **Help Desk Notifications** - Stay informed about all issues and keep technicians and end users up to date on the status of the issues by email.
* **Help Desk Reports** - Generate detailed reports about Help desk performance &issues.
* **Monthly survey** (not implement yet) - Help user feedback their feeling about system, IT staff. By those surveys, the company can identify weakness in IT system and improve them.

Comparison of some IT Helpdesk Software:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Help Desk | OTRS | Request Tracker | SiteHelpdesk-IT | EnterpriseWizard |
| Price | $0 | $0 | $2000 | $2850 |
| Type | Web Based | Web Based | Web Based | Web Based |
| Searchable database | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |
| F.A.Q. |  |  | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |
| Customer information database | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |
| Follow-up satisfaction surveys | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |  | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |
| Reporting / Statistics | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |  | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |
| Customer history | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |  | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |
| Technician email notifications | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |  | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |
| News/status notification announcements |  |  |  | D:\Sims\018669-glossy-black-icon-symbols-shapes-check-mark5-ps\018669-glossy-black-icon-symbols-shapes-check-mark5-ps.png |

Table 1 - Comparision of some IT Helpdesk Software

# Aims and Objective

The defined aim of the IT Help Desk project is to develop a software system that:

* Provides for users of an organization as center point of contact, to receive help on various IT issues.
* Allows keep track the user requests, assigning the user request for IT staffs.
* Provides efficient ways for managing issues from users.
* Provides an easier way to communicate with customers.
* Find and improve information system’s shortcomings.

The primary advantages of an IT help desk system are:

* Increased staff efficiency.
* Minimized support costs.
* Optimized performance of business systems.
* Improved employee morale and loyalty.

# eSmartDesk – Why is it needed?

* Increasing productivity of company’s employee as a result of providing quick, efficient services and avoiding repeated issues.
* Reducing ramp up time for new employee by helping them quickly become familiar with information systems of the company.
* Improving the company’s information technology base as user feedback is incorporated into future plans.
* Improving operational processes by reducing internal stress factors.
* Minimizing demands on the Help Desk staff allowing them to handle strategic issues.

Software Project Management Plan

# Problem Definition

## Name of this Capstone Project

Project full name: **Information Technology Help Desk System**

Project code: **ITHD** (Information Technology Help Desk)

Software name: **SmartDesk**

## Problem Abstract

Nowadays with the development of information technology, many companies and organizations have applied technologies into their business processes for increasing productivity, lowering time and cost. However by the shortage of IT skillful employees, many problems from light to critical may appear anywhere in the system, and cause unexpected effects to company.

By the above reason, demand of a software system that provides for users of an organization as center point of contact, to receive help on various IT issues appears. So we intend to develop software which may satisfy those necessary functions and provide plus services to help user feel most comfortable when working.

## Project Overview

### The Current System

There is not any current system. If users have IT issues, they must call, or send email for IT staff which may cause a lot of problems like:

* + There is not enough IT staff for undertaken appeared issues.
  + There is not something like knowledge base to refer; users may face to issues which have been solved before.
  + Interacting channels between user and IT staff are just phone, mail. So it will be very uncomfortable if those channels have problem.
  + IT team does not have visual report with performance statistics to identify their weakness for improving
  + It’s very difficult to manage issues from all departments for large companies.

### The Proposed System

With SmartDesk, company may manage all appeared IT issues following the most effective way helping reduce time, risks, and increase staff’s productivity. Here are some features of SmartDesk:

* + - Self service (user basic interface): Provide user a simple interface to submit issues, track issues, search necessary information.
    - FAQs (frequently asked questions) knowledge base: This function is built into the system which supports a large database of simple IT problems, their solutions and guidelines.
    - Performance report: SmartDesk also help the staff to hold on information about productivity of IT team with many types of reports. Those reports include performance statistic like number of fixed bugs in one month, percentage of bug in each technology field. Users may also export performance report to PDF format to read or print.
    - User notification: With this function user can view the updated status of submitted issues by message log, or automatic email. IT staff can be alerted when they receive new issues, or their current issues are near to deadline.
    - Monthly survey (not implemented yet): This function help user feedback their feeling about system, IT staff. By those surveys, the company can identify weakness in IT system and improve them.

### Boundaries of the System

* + SmartDesk can be used in companies and organizations which apply IT network to their business process and have IT teams
  + The company must also have an intranet, portal and mail system.
  + The software supports two language English and Vietnamese (not implemented yet)

# Project Organization

## Software Process Model

With the schedule of weekly reports for every stage, the software will be developed by using waterfall model which is very simple and require minimal resource for implementation. With waterfall model, developing process will include five main phases:

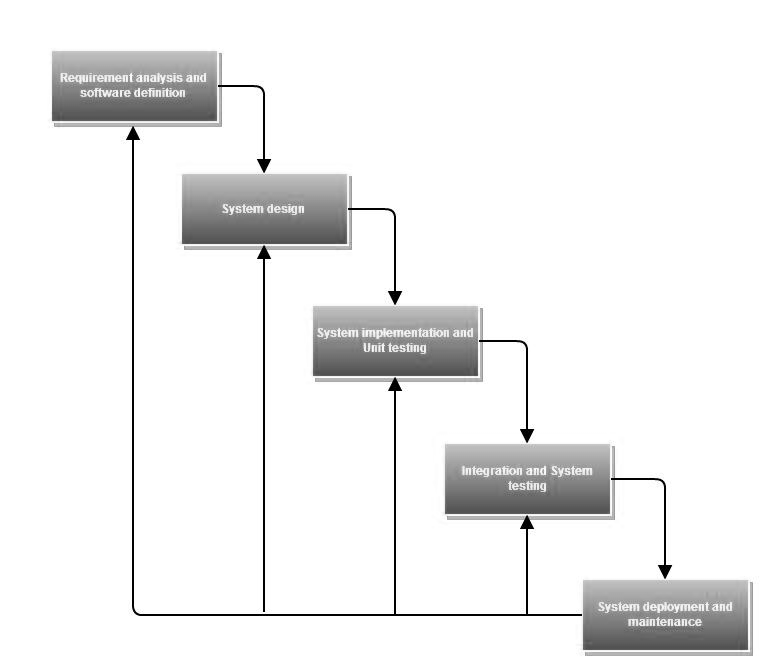


Figure 1 - Waterfall model

* + - Requirement analysis and software definition: The first phase is also the most important phase of the process which may cause great effect to other phases. This phase includes gathering as much as possible requirements from customer, or other information source like World Wide Web, and producing a most detail and accurate software definition.
    - System design: This phase is fundamental for implementation phase. Based on customer’s requirements to create logical modules, and definite their inter relations. Using algorithm and diagram to describe implementation of those modules.
    - System implementation and Unit testing: Developing software modules follow detail designs, and doing unit testing for each module.
    - Integration and System testing: Testing output, performance in modules integrating process, and retests all functions of whole system.
    - System deployment and maintenance: After testing completely, the software is handle over client, developing team will respond for maintenance of the system.

## Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Nguyễn Trọng Tài | Project Manager | Supporting business issues, tracking project progress |
| Nguyễn Hải Long | Team Leader | Controlling workflow, tracking task progress, supporting team members |
| Lê Mai Hồng | Team Member | Developing, testing, customer support |
| Nguyễn Quang Minh | Team Member | Developing, testing, customer support |
| Lê Đức Huy | Team Member | Developing, testing, customer support |

Table 2 - Members of Project

## Tools and Techniques

* + Software: NetBeans IDE 6.9.1, Visual SVN (subversion), Tortoise SVN, MSSQL server
  + Hardware: Personal Laptop, Computer
  + Techniques: AJAX (asynchronous java and xml), XML (extendible markup language), dynamic HTML with javascript, jquery, css3, framework struts 2, log4j logger, hibernate framework

# Project Management Plan

## Task

|  |  |
| --- | --- |
| Preliminary Investigation or Analysis: | |
| Description | Extracting raw requirements from users, identifying the feasibility of the project and needed resources for it |
| Deliverables | Feasibility report, developing direction for project |
| Resources Needed | 16 man-day |
| Dependencies and Constrains | Depend on raw requirements from customer |
| Risks | Project may not be feasible for developing because of strange requirements or lack of technologies |
| Prepare Introduction Report | |
| Description | Based on requirement from customer to describe the initial idea of capstone project, comparison with other system |
| Deliverables | Introduction report. |
| Resources Needed | 2 man-day |
| Dependencies and Constrains | Depend on feasibility report |
| Risks | N/A |
| Prepare Software Project Management Plan | |
| Description | Definite the problem of current system, project organization, and prepare the project plan with task list |
| Deliverables | Software project management plan |
| Resources Needed | 5 man-day |
| Dependencies and Constrains | Depend on the introduction report, and the task schedule must be in 15 weeks |
| Risks | Workflow of system may change when implementation  Developing team do not have much experience which may lead to fail in keeping schedule |
| Specification and Requirement analysis | |
| Description | Gather requirement from customer by meeting, or from other resource like World Wide Web; Analyze requirements and describe them in a proper way for programmer |
| Deliverables | Software requirement specification document |
| Resources Needed | 12 man-days |
| Dependencies and Constrains | Depend on customer requirements and must follow the SRS template |
| Risks | Difficulties in interacting with customer, they just describe requirements in simple way  Resource may not have a thorough grasp of template |
| System Design | |
| Description | Design system architecture, describe system component, user interface design, database in a meaningful way for programmer |
| Deliverables | Software design description document |
| Resources Needed | 19 man-day |
| Dependencies and Constrains | Resource must understand system requirements, have knowledge about design with diagram and follow the SDD template |
| Risks | Design document may not describe the feature clearly enough which may lead to misunderstand of programmer  Designer may not understand requirement enough |
| System Implementation and Unit Testing | |
| Description | Implement system’s units follow design document and execute unit testing to assure their qualities |
| Deliverables | Source code, Unit test document, Bug logging document |
| Resources Needed | 102 man-day |
| Dependencies and Constrains | Depend on project plan and software design description document; and must follow the schedule |
| Risks | Resource may not finish their task by individual problems  Resource may not understand design document  Resource may not have experience in testing |
| Integration and System Testing | |
| Description | Integrating system’s units follow workflow and execute system testing to assure the system can satisfy user requirements |
| Deliverables | Test case document, Bug logging document, Whole system source code |
| Resources Needed | 21 man-day |
| Dependencies and Constrains | Unit implementation and testing must complete |
| Risks | Difficulties in integrating system’s workflow  User want to change requirements |
| Prepare User Manual | |
| Description | Prepare document to help user understand how to operate and manage the system |
| Deliverables | User manual document |
| Resources Needed | 3 man-day |
| Dependencies and Constrains | Depend on system design document |
| Risks | Resource must understand workflow of system and describe in a meaningful way to user |

Table 3 - Task

## Task Sheet: Assignments and Timetable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Name | Duration | Star Date | End Date | Resource |
| Preliminary investigate | 4 days | 5/8/12 | 5/11/12 | LongNH, MinhNQ, HongLM, HuyLD |
| Prepare Introduction Report | 1 day | 5/14/12 | 5/15/12 | MinhNQ, HuyLD, LongNH |
| Prepare Software Project Management Plan | 4.5 days | 5/15/12 | 5/21/12 | LongNH |
| Specification and Requirement analysis | 4.5 days | 5/22/12 | 5/28/12 | LongNH, MinhNQ, HongLM, HuyLD, HongLM |
| System Design | 8 days | 5/28/12 | 6/7/12 | LongNH, MinhNQ, HongLM, HuyLD, HongLM |
| System Implementation and Unit Testing | 28 days | 6/7/12 | 7/7/12 | LongNH, MinhNQ, HongLM, HuyLD, HongLM |
| Integration and System Testing | 15.5 days | 7/6/12 | 7/27/12 | LongNH, MinhNQ, HongLM, HuyLD, HongLM |
| Prepare User Manual | 3 days | 7/16/12 | 7/18/12 | HongLM |

Table 4 - Task Sheet

# Coding Convention

|  |  |
| --- | --- |
|  |  |
| General | Keep your code simple and comprehensible. |
| Be precise and consistence. |
| Don’t optimize too soon. |
| File Organization | Java source file is not longer than 2000 lines or have more than 50 methods. |
| Each Java source file contains a single public class or interface. |
| Java source files have the following ordering: Beginning Comments, Package and Import Statement, Class and Interface Declaration. |
| Indentation and Braces | Avoid use of tab character. The unit of indentation is 4 spaces. |
| No lines longer than 120 characters. |
| Open curly brace “{” of class/method declarations and other code blocks should be at “END OF LINE” of the first statement of code block |
| Naming Conventions | Name indicates purpose of the file / variable / control / method. |
| Use terminology applicable to the domain. |
| Identifiers are shorter than 20 characters. |
| Avoid names that are similar or differ only in case, cryptic name, abbreviation |
| No special character in the method name |
| Class/interface starts with uppercase |
| Declaration | One declaration per line. |
| The variable must be placed at the beginning of the blocks. |
| The variable must be initialized before it's used. |
| Put declarations only at the beginning of blocks except index in for loop. |
| Opening bracket "{" always appears at end of line. |
| Closing bracket "}" should appear on a new line. |
| Methods are separated by a blank line. |
| Comment | Program can have four styles o implementation comments: block, singe-line, trailing, end-of-line |
| Block comment should be preceded by a blank line to set it apart from the rest of the code (/\*\*…\*/) |
| Singe-line comment can appear on a singeline intended to the level of the code that follows |

Table 5 - Coding Convention

Software Requirement Specification

# User Requirement Specification

SmartDesk provides a single point of contact for IT issue managing and fixing in company. Besides it also support useful features like FAQs, performance report, or user notifying:

* When employees got an issue, they can use FAQs to find relevant answers for their issues; or log in to the system to submit a new issue which describe problem that they face to. They can also access their existing issues and track the status of all their requests.
* Admin and IT staff members can use the system to manage issues (organize, prioritize, and respond to users’ requests, etc.), manage user account (create, edit info, etc.)
* User will be informed about all issues or changes (associated with them) by email or message.
* User can use the report function to analyze the performance of IT staff, or distribution of issue on IT fields.

# System Requirement Specification

## External Interface Requirements

### User Interfaces

* User can interact with the system and other users through the user interface. Below are screens available for users:

|  |  |
| --- | --- |
| Screen Name | Function |
| Login Screen | Allow registered users to login to the system |
| Log Basic Issue Screen | Allow users (Employee) to log issue with basic information |
| Log Advanced Issue Screen | Allow users(Admin) to log issue with detailed information |
| Search Issue Screen | Allow users to search for issues |
| View Detail Issue Screen | Allow users to view all issues’ information |
| Edit Issue Screen | Allow users to edit information of issue |
| Create User Screen | Allow admin to create account |
| Edit User Screen | Allow admin to edit information of user account |
| Search User Screen | Allow admin to search for user accounts |
| View Detail User Screen | Allow admin to view all user accounts’ information |
| My Message Screen | Allow users to view all their messages |
| Create Message Screen | Allow users to send message |
| Create Report Screen | Allow users to choose one in a list of report templates to generate |
| Create FAQ Screen | Allow user to create FAQ |
| Search FAQs Screen | Allow users to search for FAQs |
| Edit FAQ Screen | Allow user to edit information of FAQ |
| View Detail FAQs Screen | Allow user to view all FAQs’ information |
| Create Report | Allow user to create reports |

Table 6 - Screens

### Hardware Interfaces

* Network cable, network interface card

### Software Interfaces

* N/A

### Communications Protocol

* N/A

## System Features

This section includes all features which SmartDesk will support for users to interact and manage IT system more efficient. There are three kinds of users who will interact directly with the system:

|  |  |
| --- | --- |
| Actor | Description |
| Employee | Employee in company which applied SmartDesk system who will log happened IT issues |
| Admin | The person who respond for managing the whole system |
| IT Staff | The person who respond for correcting IT issue, and helping employees increase their productivities |

Table 7 - Actors

The overall use case model of SmartDesk application:



Figure 2 - Overall Use-Case Diagram

### System Feature – Issue Managing



Figure 3 - Issue Managing Feature Use-Case Diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-1 SPECIFICATION** | | | | |
| **Use-case No.** | <UC001> | **Use-case Version** | | <1.0> |
| **Use-case Name** | Log Basic Issue | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** Employee  **Summary:** This use case is about adding basic issues (Issue just has simple information like name, content…).  **Goal:** Employee can add basic issues with simple form to describe their problems  **Triggers***:* Employee got an IT problem.  **Preconditions:** User must log in with role “Employee”  **Post-Conditions:** Issue is added to database successful with basic information.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login and choose **LogBasic Issue** function 2. User input data | System Response:   1. Display **Log Basic Issue Screen**with input fields for basic information of issue. 2. Validate all input data. | | 1. Employee check for error input and click Submit button (when no error) |  | |  | 1. System insert new issue to database 2. Display result to user |   **Alternative Scenario:** N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:**N/A  **Business Rules:**   * User must enter data to all important field and solve input errors before submitting | | | | |
| **USE CASE-2 SPECIFICATION** | | | | |
| **Use-case No.** | <UC002> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Log Advanced Issue | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | High | |
| **Actor:** User (Admin/ IT staff)  **Summary:** This use case is about adding detailed issues (Issues added by admin have all necessary information).  **Goal:** Admin can add detailed issues to database successful  **Triggers***:* Admin received an email or phone call about an issue from user.  **Preconditions:** User must log in with role “Admin”  **Post-Conditions:** Issue is added with detailed information.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login with role Admin and choose **Log Advanced Issue** function 2. User input all necessary information | System Response:   1. Display **Log Advanced Issue Screen** with input fields for detailed information of issue. 2. Validate all input data. | | 1. Employee check for error input and click Submit button (when no error) |  | |  | 1. System insert new issue to database 2. Display result to user |   **Alternative Scenario:** N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:**   * User must enter data to all important field and solve input errors before submitting | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-3 SPECIFICATION** | | | | |
| **Use-case No.** | <UC003> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Search Issue | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Employee, IT Staff)  **Summary:** This use case is about searching issues with many criterions.  **Goal:** User can search for suitable issues more easily.  **Triggers***:* Actor’s behavior  **Preconditions:** User must log in into the system.  **Post-Conditions:** Search results are shown.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login and choose **Advanced Search Issue** function | System Response:   1. Display **Advanced Search Issue Screen.** This screen includes input fields, select box for user to search easily. | | 1. Input suitable field user wants to search. | 1. Validate input data. | | 1. Solve input errors, and click **Search** button | 1. Search for proper results in database 2. Display results to user | |  |  | |  |  |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-4 SPECIFICATION** | | | | |
| **Use-case No.** | <UC004> | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Detailed Issue | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Employee, IT Staff)  **Summary:** This use case is about view detail issue function which help user know all issue’s information  **Goal:** User can view issue info to find their needed information  **Triggers***:*  Actor’s behavior  **Preconditions:** User must log in into the system, and do actions to show the issue they want to view like search issue.  **Post-Conditions:** User can find their needed information from issue’s information  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User select issue which wants to view | System Response:   1. Search in database for data of selected issue 2. Display **View DetailIssue Screen.** This screen includes all issue’s information (name, problem, solution, status,…) |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-5 SPECIFICATION** | | | | |
| **Use-case No.** | <UC005> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit Issue’s Info. | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Employee, IT Staff)   * Admin can edit all issues’ information * Employees can edit own issues * IT Staffs can edit their assigned issues   **Summary:** This use case is about editing issue information when change appears  **Goal:** User can edit proper issue’s information when related events happen  **Triggers***:*  Related event happen which cause change to issue’s information like Admin want to assign issue to IT staff, or IT staff want to post solution for issue  **Preconditions:** User must log in into the system, and have suitable right to edit issue. They must do some action like search, view detail to see **Edit** button  **Post-Conditions:** User can edit issue successful according to their intention  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User select issue that they want to edit | System Response:   1. Display **View Detail Issue Screen** which a detailed form include fields that contain old information of issue | | 1. User enter new data to fields which they want to change | 1. Validate input data | | 1. Solve input error and click on **Submit** button | 1. Update corresponding data in database 2. Display result to user |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:**   * User must have right to edit issue * Only some field of issue can be edited | | | | |

### System Feature – User Managing



Figure 4 - User Managing Feature Use-Case Diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-6 SPECIFICATION** | | | | |
| **Use-case No.** | <UC006> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create User | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | High | |
| **Actor:** User(Admin)  **Summary:** This use case is about creating account for a user.  **Goal:** Admin can add new user to database successful  **Triggers:** When a new employee join a department  **Preconditions:** User must log in with role “Admin”  **Post-Conditions:** User’s account is created with detailed information.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login with role Admin and choose **Create User** function | System Response:   1. Display **Create User Screen** with input fields for detailed information of user. | | 1. User input all necessary information | 1. Validate all input data. | | 1. User check for error input and click Submit button (when no error) |  | |  | 1. System insert new user to database 2. Display result to user |   **Alternative Scenario:** N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:**   * User must enter data to all important fields and solve input errors before submitting. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-7 SPECIFICATION** | | | | |
| **Use-case No.** | <UC007> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit User’s Info. | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Employee, IT Staff)   * Admin can edit all user’s information * Employees and IT Staff can edit their own account   **Summary:** This use case is about editing user’s information  **Goal:** User can edit account information  **Triggers***:*  Related event happen which cause change to user’s information  **Preconditions:** User must log in into the system, and have suitable right to edit account.  **Post-Conditions:** User can edit account successful according to their intention  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User select user that they want to edit | System Response:   1. Display **View Detail User Screen** which a detailed form include fields that contain old information of user | | 1. User enter new data to fields which they want to change | 1. Validate input data | | 1. Solve input error and click on **Submit** button | 1. Update corresponding data in database 2. Display result to user |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:**   * User must have suitable right to edit account | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-8 SPECIFICATION** | | | | |
| **Use-case No.** | <UC008> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Search User | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Staff, Employee)  **Summary:** This use case is about searching users with many criterions.  **Goal:** Admin can search for user more easily.  **Triggers***:* Actor’s behavior  **Preconditions:** User must log in into the system.  **Post-Conditions:** Search results are shown.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login and choose **Advanced Search User** function | System Response:   1. Display **Advanced Search User Screen.** This screen includes input fields, select box for user to search easily. | | 1. Input suitable field user wants to search. | 1. Validate input data. | | 1. Solve input errors, and click **Search** button | 1. Search for proper results in database 2. Display results to user | |  |  | |  |  |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-9 SPECIFICATION** | | | | |
| **Use-case No.** | <UC009> | **Use-case Version** | | 1.0 |
| **Use-case Name** | View User Info | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Staff, Employee)  **Summary:** This use case is about viewing user information  **Goal:** Actor can view needed user information  **Triggers***:*  Actor’s behavior  **Preconditions:** User must log in into the system, and do actions to show the user they want to view like search user.  **Post-Conditions:** User can find their needed information  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User select user which wants to view | System Response:   1. Search in database for data of selected user 2. Display **View Detail User Screen.** This screen includes all user information (name, phone, email, etc.) |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

### System Feature – User Notifying

****

Figure 5 - User Notifying Feature Use-Case Diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-10 SPECIFICATION** | | | | |
| **Use-case No.** | <UC010> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Notify by Email | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Employee, IT Staff)  **Summary:** This use case is about auto notifying user when their related issues are updated  **Goal:** User can watch for any change to their related issue  **Triggers***:* Actor’s behavior related to issue’s information like editing or assigning  **Preconditions:** User must fill correct email address in their account information; An event related to user’s issue occurs  **Post-Conditions:** User is notified about changes of their issues  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User do some actions which cause change to issue like editing, assigning | System Response:   1. Auto send an alert email to related users mail address |   **Alternative Scenario:**N/A  **Exceptions:**   * When mail system has problem, the system will try again after every 10 minutes   **Relationships:** N/A  **Business Rules:**   * Auto notify function must follow the setting of Admin | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-11 SPECIFICATION** | | | | |
| **Use-case No.** | <UC011> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Notify by Message Log | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Employee, IT Staff)  **Summary:** This use case is about auto notifying user when their related issues are updated  **Goal:** User can watch for any change to their related issue  **Triggers***:*  Actor’s behavior related to issue’s information like editing or assigning  **Preconditions:** An event related to user’s issue occurs  **Post-Conditions:** User is notified about changes of their issues  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User do some actions which cause change to issue like editing, assigning | System Response:   1. Auto insert message record to database with corresponding data |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:**   * Auto notify function must follow the setting of Admin | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-12 SPECIFICATION** | | | | |
| **Use-case No.** | <UC012> | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Message Log | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, Employee, IT Staff)  **Summary:** This use case is about view message log function of any user  **Goal:** User can watch for any change to their related issue in message log  **Triggers***:*  Actor’s behavior  **Preconditions:** User must login into the system  **Post-Conditions:** User can keep updating about their issue status  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login and choose **My Message**function | System Response:   1. Search in database for all message of user 2. Display **My Message Screen** which include all messages of user about their issues | | 1. User choose duration which they want to view | 1. Display corresponding messages |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

### System Feature – Report Managing



Figure 6 - Report Managing Feature Use-Case Diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-13 SPECIFICATION** | | | | |
| **Use-case No.** | <UC013> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Produce Report | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin)  **Summary:** This use case is about generating performance reports.  **Goal:** User can create report based on report template.  **Triggers***:* Actor’s behavior  **Preconditions:** User must log in into the system with role “Admin”.  **Post-Conditions:** Report is created and shown.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login and choose **Reports** function | System Response:   1. Display **Produce Report Screen** with available report template | | 1. Choose suitable template (statistics, diagrams, charts, etc.) and time duration. 2. Click Generate button. | 1. Search data in database and calculate follow formula, create charts, diagrams. 2. Display result of report follow template |   **Alternative Scenario:** User selects to save reports as PDF format.  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

### System Feature – FAQs Managing



Figure 7 - FAQs Managing Feature Use-Case Diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-14 SPECIFICATION** | | | | |
| **Use-case No.** | <UC014> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Create FAQs | | | |
| **Author** | HongLM | | | |
| **Date** | 13/05/2012 | **Priority** | High | |
| **Actor:** User(IT Staff)  **Summary:** This use case is about creating FAQs.  **Goal:** User can create FAQs  **Triggers***:* When there are frequently logged issues.  **Preconditions:** User must log in with role “It Staff”  **Post-Conditions:** FAQs is created with detailed information.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login and choose **Create FAQs** function | System Response:   1. Display **Create FAQs Screen** with input fields for detailed information of FAQs. | | 1. User input all necessary information | 1. Validate all input data. | | 1. User check for error input and click Submit button (when no error) |  | |  | 1. System insert new FAQs to database 2. Display result to user |   **Alternative Scenario:** N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:**   * User must enter data to all important fields and solve input errors before submitting. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-15 SPECIFICATION** | | | | |
| **Use-case No.** | <UC015> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Search FAQs | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, IT Staff, Employee)  **Summary:** This use case is about searching FAQs with many criterions.  **Goal:** User can search for suitable FAQs more easily.  **Triggers***:* Actor’s behavior  **Preconditions:** User must log in into the system.  **Post-Conditions:** Search results are shown.  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User login and choose **Search FAQs** function | System Response:   1. Display **Advanced Search FAQs Screen.** This screen includes input fields, select box for user to search easily. | | 1. Input suitable field user wants to search. | 1. Validate input data. | | 1. Solve input errors, and click **Search** button | 1. Search for proper results in database 2. Display results to user | |  |  | |  |  |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-16 SPECIFICATION** | | | | |
| **Use-case No.** | <UC016> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Edit FAQs. | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, IT Staff)   * Admin can edit all FAQs information * IT Staffs can edit all FAQs information except of approval info   **Summary:** This use case is about editing FAQs information  **Goal:** User can edit proper FAQs’ information when related events happen  **Triggers***:*  Related event happen which cause change to FAQ’s information like IT staff want to FAQs content  **Preconditions:** User must log in into the system, and have suitable right to edit FAQs. They must do some action like search, view detail to see **Edit** button  **Post-Conditions:** User can edit issue successful according to their intention  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User select FAQ that they want to edit | System Response:   1. Display **View Detail FAQ Screen** which a detailed form include fields that contain old information of FAQ | | 1. User enter new data to fields which they want to change | 1. Validate input data | | 1. Solve input error and click on **Edit** button | 1. Update corresponding data in database 2. Display result to user |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:**   * User must have right to edit FAQs * Only some field of FAQs can be edited | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-17SPECIFICATION** | | | | |
| **Use-case No.** | <UC017> | **Use-case Version** | | 1.0 |
| **Use-case Name** | View Detailed FAQ | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin, IT Staff, Employee)  **Summary:** This use case is about view FAQs info function which allow user to view all information of selected FAQ  **Goal:** User can find necessary information from FAQ  **Triggers***:* Actor’s behavior  **Preconditions:** User must log in into the system.  **Post-Conditions:** User can view information of selected FAQ  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User select FAQ that they want to view | System Response:   1. Search on database for data of selected FAQ 2. Display **View Detail FAQ Screen** which include all data of selected FAQ | |  |  | |  |  |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE-18 SPECIFICATION** | | | | |
| **Use-case No.** | <UC018> | **Use-case Version** | | 1.0 |
| **Use-case Name** | Approve FAQ | | | |
| **Author** | LongNH | | | |
| **Date** | 13/05/2012 | **Priority** | Normal | |
| **Actor:** User (Admin)  **Summary:** This use case is about Approve FAQ function which allow admin to approve completed FAQs  **Goal:** Admin can approve completed FAQs for user to search and view  **Triggers***:* Actor’s behavior  **Preconditions:** User must log in into the system with role Admin.  **Post-Conditions:** Completed FAQ is approved successful  **Main Success Scenario:**   |  |  | | --- | --- | | Actors action:   1. User search for FAQs which status Un-approve and check their content 2. If FAQ has been completed, change status of approval select box | System Response:   1. Update status of that FAQ to Approved in database 2. Display result to user | |  |  | |  |  |   **Alternative Scenario:**N/A  **Exceptions:**   * When database connection has problem, the system will ask user for trying again.   **Relationships:** N/A  **Business Rules:** N/A | | | | |

# Software System Attributes

## Reliability

* The system can be used like “always on” system.
* There is no requirement for system maintenance task from the user.
* Mean Time Between Failures (MTBF): more than 6 months.
* Mean Time To Repair (MTTR): less than 24 hours.
* Accuracy: 100%
* Maximum Bugs and Defect Rate: 0.3 bugs per thousand lines of code (0.3 bugs/KLOC).
* Critical bugs:
* Loss of data: not any
* Unable to use part of the system’s functionality: not any

## Availability

* The server shall be working 24/7.

## Security

- All sensitive information (password, etc.) must be encrypted when storing in database and during transmission over networks using MD5 hash.

- Validate input data in SQL query before execute to avoid SQL Injection

- There are 3 types of users in the SmartDesk system:

* **Admin:** the **Administrator** of the system, admin has the primitive privilege on the system features and database.
* **IT Staff:** the IT support Staff of a company, they can manage all issues of the system.
* **Employee:** employee can use the system to log issue or keep track of their logged issues.

## Maintainability

* All code shall be fully documented. All program files shall include comments concerning authorship and date of last change.
* The code shall be modular to permit future modifications.

## Portability

* The software shall be designed as cross-platform software.

## Performance

* The software shall be met the need of a mid-size company(100 – 500 employees)

Large tables and indexes must be partitioned data into smaller, more manageable sections by using partition in SQL Server 2005

# Entity Relationship Diagram

This diagram describes all the entities in this system and their import properties.

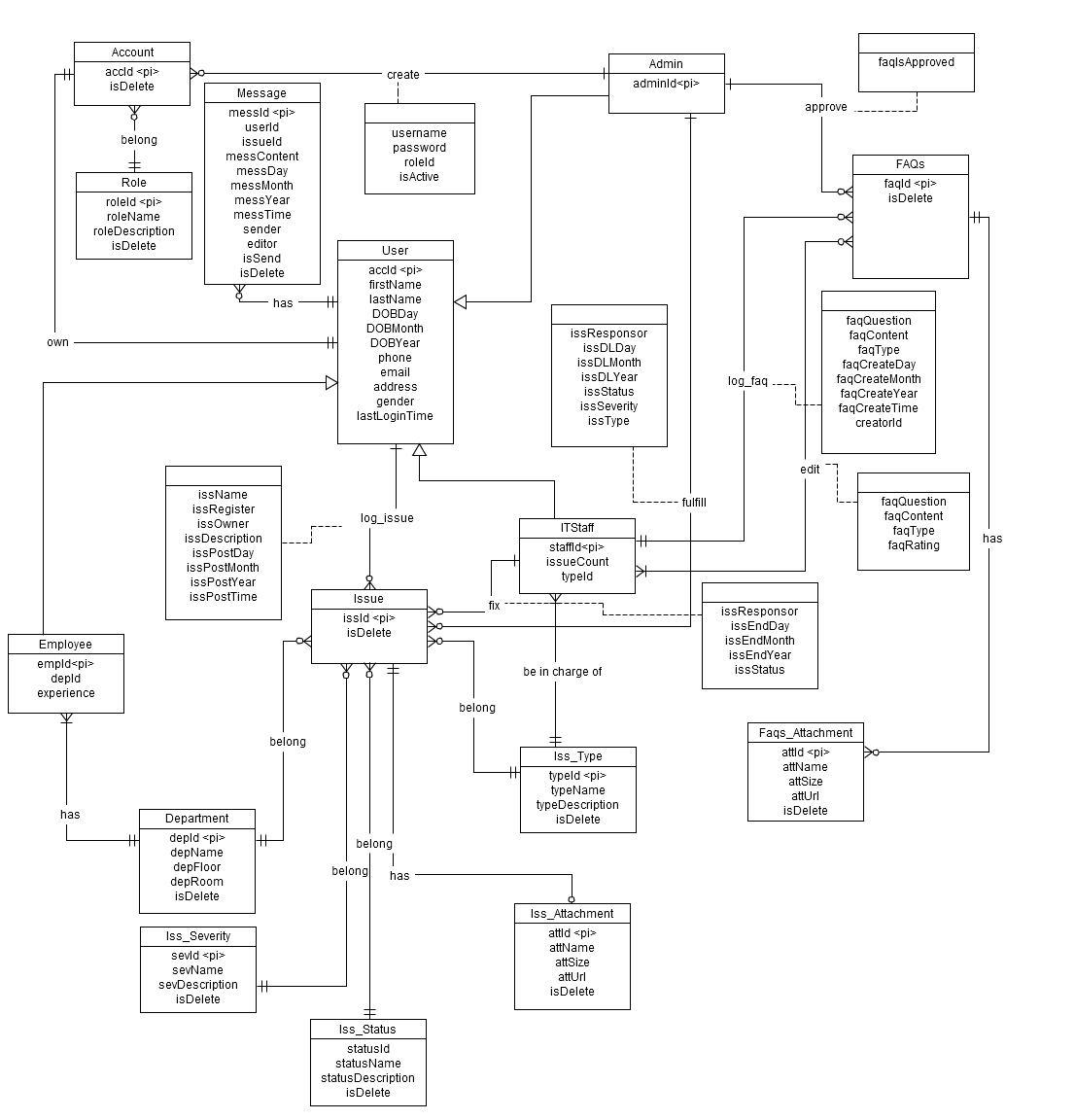


Figure 8 - Entity Relationship Diagram

Software Design Description

# Design Overview

SDD document objective is providing for project’s stakeholders multi specific views of SmartDesk system like:

* + System architecture design which describes the planned model of system, its’ layers, and each layer function
  + Component design which describes all system’s component (class, interface…) together with relationships between them. It also explains clearly their purposes, methods within each component and detailed algorithms, pseudo code to implement them.
  + Sequence diagram design which shows how system processes operate and in what order. Besides, this section depicts the objects which contained in each scenario and theirs interaction by transferred messages.
  + User interface design which describes all screens’ interface; each screen includes which types of input, output element and used by what action.
  + Database design section shows and explains the physical model of system database which includes all tables and their relationship

# System Architecture Design

## Choice of System Architecture

SmartDesk system is designed based on MVC (Model-View-Controller) model with struts 2 framework for java web application which contains three discrete layers. Each layer has their own responsibility which help stakeholder to implement, and monitor system easier:

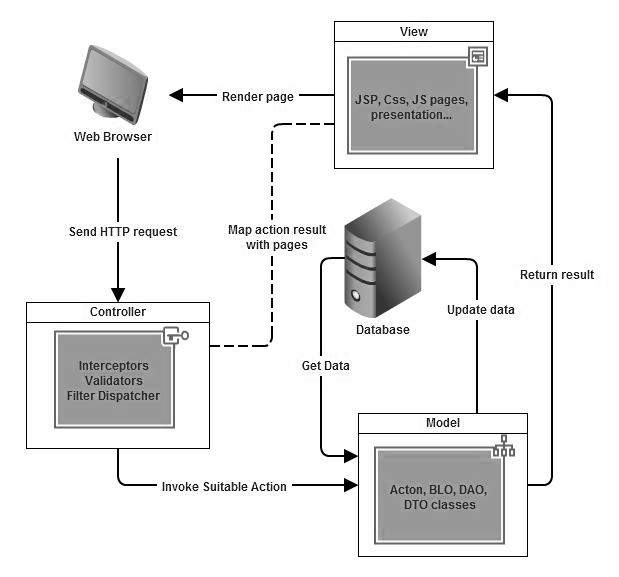


Figure 9 - MVC Architecture

* + - **Controller:** receive HTTP request from web browser and map to appropriate action by Filter Dispatcher (same as Controller) which setting by XML tags in struts.xml file (Controller layer may also contain Valuators, Interceptor to check, validate data before invoke the action)
    - **Model:** contains data and business logic (Action, BLO (business logic object), DAO (data access object) and DTO (data transfer object) classes). This layer interacts directly with database to get or update data, calls methods for business logic execution, and returns result to View
    - **View:** respond for presenting the data with designed interface to web browser

## Discussion of Alternative Designs

### MVP (Model-View-Presenter)

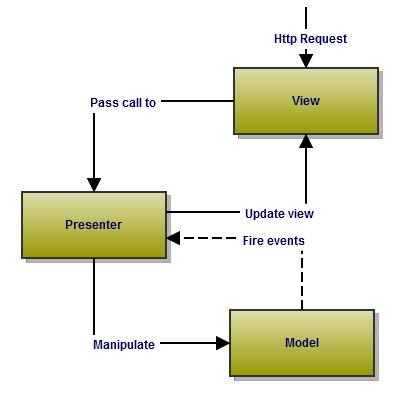


Figure 10 - MVP model

* + - MVP model also has three components like MVC mode, but Controller component is replaced by Presenter component which receives call from View component, interacts with Model component and bring changes back to the View.
    - In MVP model business logic section may be included directly in View or Presenter which can cause ambiguity and difficulty for implementation and testing.

### MVVM (Model-View-View-Model)

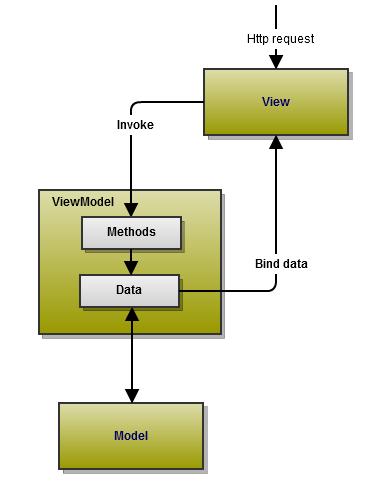


Figure 11 - MVVM model

In MVVM model, the actual component which control flow based on http request is ViewModel. This component has properties map 1:1 with appropriate View, all its methods and properties can be used by View component.

## Description of System Interface

N/A

# Component Diagram

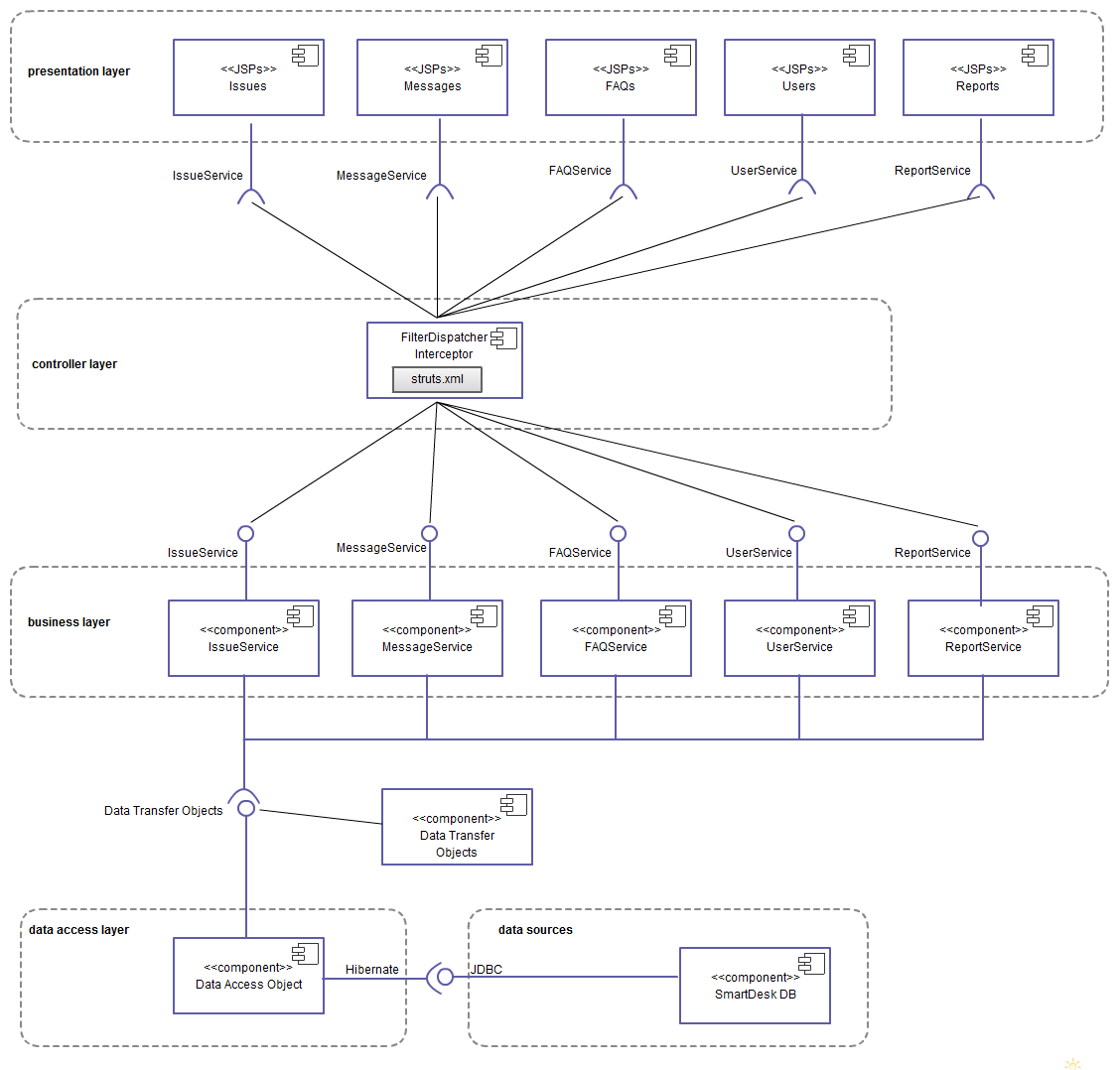


Figure 12 - Component Diagram

# Detailed Description of Components

## Managing message

### Class Diagram

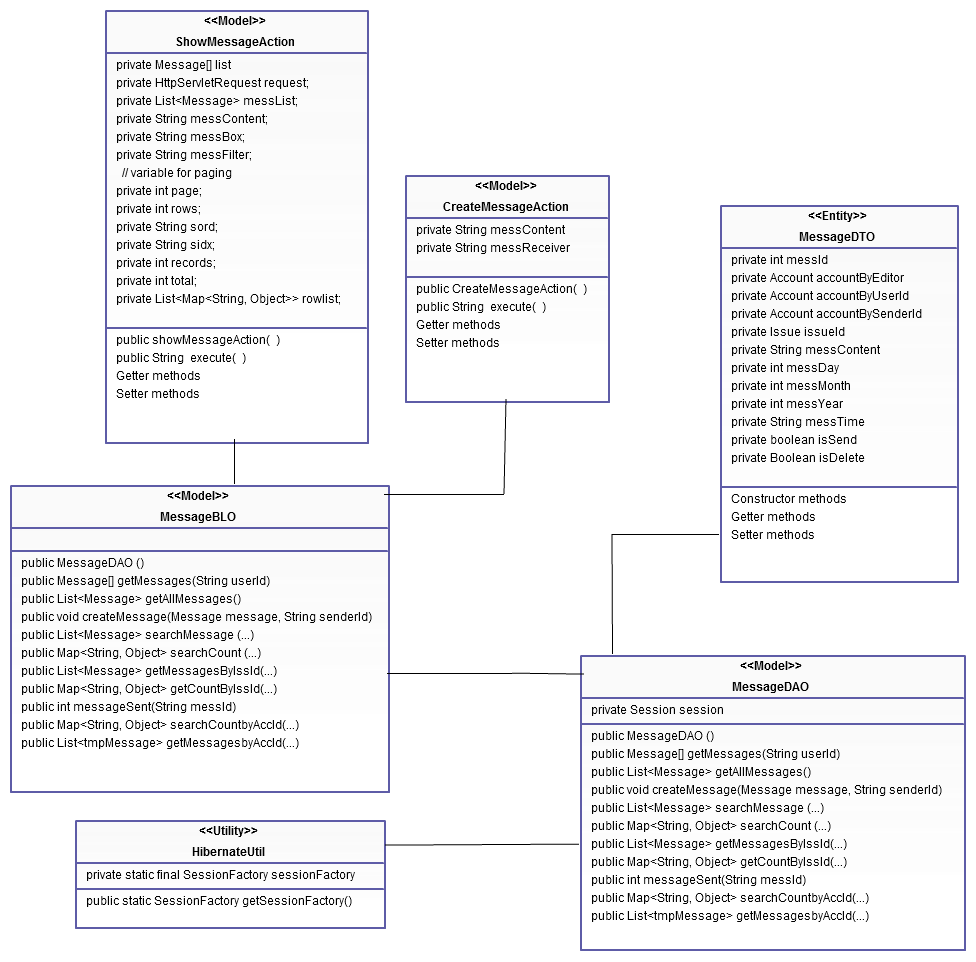


Figure 13 Class diagram of Managing message

### Class Diagram Explanation

|  |  |
| --- | --- |
| showMessageAction | |
| Properties | |
| private Message[] list ; | List of message |
| private HttpServletRequestrequest | HttpServletRequest object |
| private List<Message>messList; | List of message |
| private String messContent; | Content of message |
| private String messBox; | Message box |
| private String messFilter | Message filter |
| private int page; | Current page |
| private int rows; | Total number of rows in one page |
| private String sord; | Sorting column |
| private String sidx; | Ascending or descending sort |
| private int records; | Number of returned records in grid from the request |
| private int total; | Total records |
|  |  |
| Methods | |
| public showMessageAction( ) | Constructor |
| public String execute( ) | Process action’s business logic |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| CreateMessageAction | |
| Properties | |
| private String messContent | Content of message |
| private String messReceiver | Receiver of message |
| Methods | |
| public CreateMessageAction( ) | Constructor |
| public String execute( ) | Process action’s business logic |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| MessageBLO | |
| Properties | |
| Methods | |
| public MessageDao() Constructor | |
| public Message[] getMessages(String userId) | Get message by its owner |
| public Message[] getAllMessages() | Get all messages |
| public void createMessage(Message message,StringsenderId) | Create new message |
| public List<message>searchMessage(…) | Search message |
| public Map<String,Object>searchCount(…) | Get total number of records and pages |
| public List<message>getMessagesByIssId(…) | Get messages by issue id |
| public Map<String,Object>getCountByIssId(…) | Get total number of records and pages by issue id |
| public intmessageSent(String messId) | Update one message status |
| public Map<String,Object>searchCountbyAccId(…) | Get total number of records and pages by account id |
| public List<tmpMessage>getMessagesbyAccId(..) | Get messages by account id |

|  |  |
| --- | --- |
| MessageDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public MessageDAO() | Constructor |
| public Message[] getMessages(String userId) | Get message by its owner |
| public Message[] getAllMessages() | Get all message |
| public void createMessage(Message message,StringsenderId) | Create new message |
| public List<message>searchMessage(…) | Search message |
| public Map<String,Object>searchCount(…) | Get total number of records and pages |
| public List<message>getMessagesByIssId(…) | Get messages by issue id |
| public Map<String,Object>getCountByIssId(…) | Get total number of records and pages by issue id |
| public intmessageSent(String messId) | Update one message status |
| public Map<String,Object>searchCountbyAccId(…) | Get total number of records and pages by account id |
| public List<tmpMessage>getMessagesbyAccId(..) | Get messages by account id |

## Managing issue

### Class Diagram

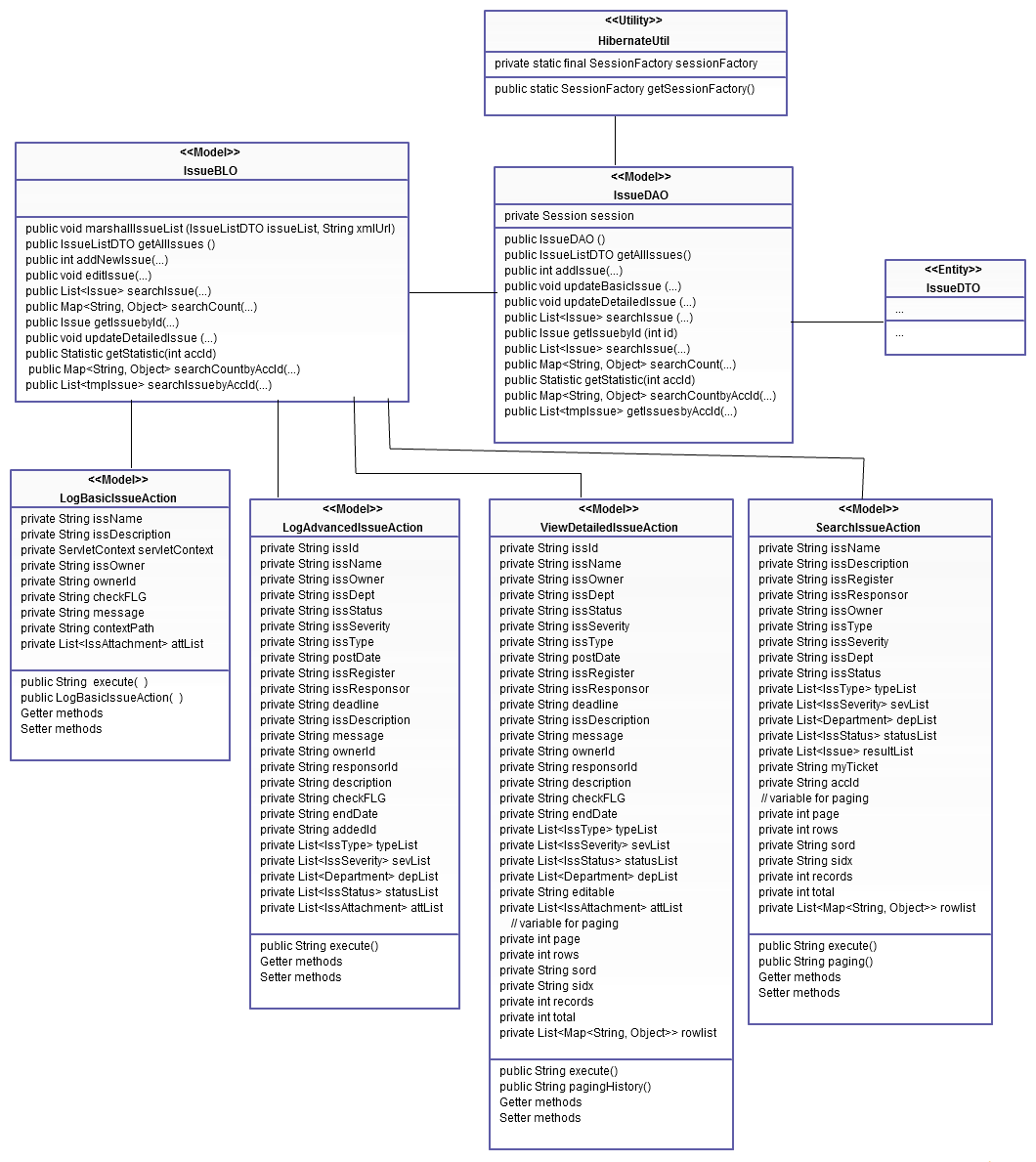
**

Figure 14 - ClassDiagram of Managing Issue

### Class Diagram Explanation

|  |  |
| --- | --- |
| LogBasicIssueAction | |
| Properties | |
| private String issName | Name of issue |
| private String issDescription | Description of issue |
| private ServletContextservletContext | ServletContext object |
| private String issOwner | Owner of issue |
| private String ownerId | Identification of owner |
| private String checkFLG | Flag which mode the method returns |
| private String message | Execution mode |
| private String contextPath | Project context path |
| private List<IssAttchment>attList | List of issue attachments |
| Methods | |
| public String execute( ) | Process action’s business logic |
| public LogBasicIssueAction( ) | Constructor |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| ViewDetailedIssueAction | |
| Properties | |
| private String issId | Issue identification |
| private String issName | Name of issue |
| private String issOwner | Owner of issue |
| private String issDept | Department of issue |
| private String issStatus | Status of issue |
| private String issSeverity | Severity of issue |
| private String issType | Type of issue |
| private String postDate | Day of logging issue |
| private String issRegister | Person who log issue |
| private String issResponsor | Person who fix issue |
| private String deadline | Deadline of issue |
| private String issDescription | Description of issue |
| private String message | Execution mode |
| private String ownerId | Identification of issue’s owner |
| private String responsorId | Identification of issue’s responser |
| private String description | Temporary input description |
| private String checkFLG | Flag which mode the method returns |
| private String endDate | Closed or cancelled date of issue |
| private List<IssType>typeList | List of issue’s type |
| private List<IssSeverity>sevList | List of issue’s severity |
| private List<IssStatus>statusList | List of issue’s department |
| private List<Department>depList | List of issue’s status |
| private List<IssAttchment>attList | List of issue attachments |
| private String editable | Flag to check if user can edit issue or not |
| private List<IssAttchment>attList | List of issue attachments |
| private int page; | Current page |
| private int rows; | Total number of rows in one page |
| private String sord; | Sorting column |
| private String sidx; | Ascending or descending sort |
| private int records; | Number of returned records in grid from the request |
| private int total; | Total records |
| private List<Map<String,Object>>rowList | List of mapping between each row and each issue |
| Methods | |
| public String execute() | Process action’s business logic |
| pubic String pagingHistory() | Process paging |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| SearchIssueAction | |
| Properties | |
| private String issName | Name of issue |
| private String issDescription | Description of issue |
| private String issRegister | Person who log issue |
| private String issResponsor | Person who fix issue |
| private String issOwner | Owner of issue |
| private String issType | Type of issue |
| private String issSeverity | Severity of issue |
| private String issDept | Department of issue |
| private String issStatus | Status of issue |
| private List<IssType>typeList | List of issue’s type |
| private List<IssSeverity>sevList | List of issue’s severity |
| private List<Department>depList | List of issue’s department |
| private List<IssStatus>statusList | List of issue’s status |
| private List<Issue>resultList | List of found issues |
| private String myTicket | Flag to check if user is in my ticket screen |
| private String accId | Identification of account |
| private int page; | Current page |
| private int rows; | Total number of rows in one page |
| private String sord; | Sorting column |
| private String sidx; | Ascending or descending sort |
| private int records; | Number of returned records in grid from the request |
| private int total; | Total records |
| private List<Map<String,Object>>rowList | List of mapping between each row and each issue |
| Methods | |
| public String execute() | Process action’s business logic |
| pubic String paging() | Process paging |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| LogAdvancedIssueAction | |
| Properties | |
| private String issId | Issue identification |
| private String issName | Name of issue |
| private String issOwner | Owner of issue |
| private String issDept | Department of issue |
| private String issStatus | Status of issue |
| private String issSeverity | Severity of issue |
| private String issType | Type of issue |
| private String postDate | Day of logging issue |
| private String issRegister | Person who log issue |
| private String issResponsor | Person who fix issue |
| private String deadline | Deadline of issue |
| private String issDescription | Description of issue |
| private String message | Execution mode |
| private String ownerId | Identification of issue’s owner |
| private String responsorId | Identification of issue’s responser |
| private String description | Temporary input description |
| private String checkFLG | Flag which mode the method returns |
| private String endDate | Closed or cancelled date of issue |
| private String addedId | Result of insert method |
| private List<IssType>typeList | List of issue’s type |
| private List<IssSeverity>sevList | List of issue’s severity |
| private List<Department>depList | List of issue’s department |
| private List<IssStatus>statusList | List of issue’s status |
| private List<IssAttchment>attList | List of issue attachments |
| Methods | |
| public String execute() | Process action’s business logic |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| IssueBLO | |
| Properties | |
| Methods | |
| public void marshallIssueList (IssueListDTOissueList, String xmlUrl) | Change all employees objects to XML format |
| public IssueListDTOgetAllIssues () | Get list of all issues |
| publicintaddNewIssue(…) | Add new issue |
| public void editIssue(…) | Edit one issue |
| public List<Issue>searchIssue(…) | Search issue |
| public Map<String,Object>searchCount(…) | Get total number of records and pages |
| public Issue getIssuebyId(int id) | Get one issue by its Id |
| public void updateDetailedIssue (…) | Update one issue |
| public Statistic getStatistic(intaccId) | Get issue statistics |
| public Map<String,Object>searchCountbyAccId(…) | Get total number of records and pages by account id |
| public List<tmpIssue>getMessagesbyAccId(..) | Get messages by account id |

|  |  |
| --- | --- |
| IssueDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public IssueDAO() | Constructor |
| public IssueListDTOgetAllIssues() | Get all issues |
| publicintaddIssue(…) | Add new issue |
| public void updateBasicIssue (…) | Update basic issue |
| public void updateDetailedIssue (…) | Update detail issue |
| public List<Issue>searchIssue (…) | Search issue |
| public Map<String,Object>searchCount(…) | Get total number of records and pages |
| public Issue getIssuebyId(int id) | Get one issue by its Id |
| public void updateDetailedIssue (…) | Update one issue |
| public Statistic getStatistic(intaccId) | Get issue statistics |
| public Map<String,Object>searchCountbyAccId(…) | Get total number of records and pages by account id |
| public List<tmpIssue>getMessagesbyAccId(..) | Get issues by account id |

## Mail notification

### Class Diagram

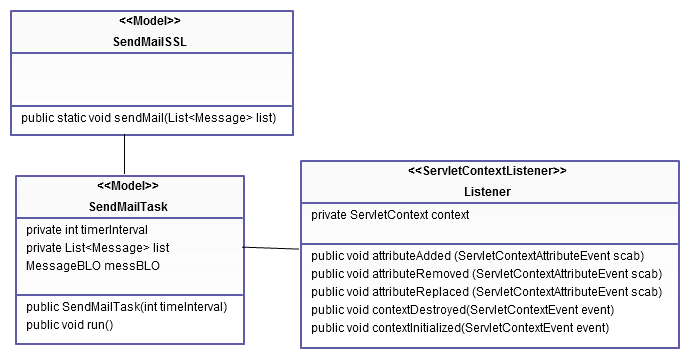


Figure 15 - ClassDiagram of Mail notificatiion

### Class Diagram Explanation

|  |  |
| --- | --- |
| SendMailTask | |
| Properties | |
| private inttimerInterval | Time to run SendMailTask method |
| private Message[] list | List of message |
| MessageBLOmessBLO | MessageBLO object |
| Methods | |
| public SendMailTask(inttimeInterval) | Constructor |
| public void run() | Execute mail function |

|  |  |
| --- | --- |
| SendMailSSL | |
| Properties | |
| Methods | |
| public static void sendMail(Message[] list) | Send message via email to user |

|  |  |
| --- | --- |
| Listener | |
| Properties | |
| private ServletContext context | **ServletContext object** |
| Methods | |
| public void attributeAdded(ServletContextAttributeEvent scab) | Add attribute to the ServletContext object |
| public void attributeRemoved(ServletContextAttributeEvent scab) | Remove attribute from ServletContext object |
| public void attributeReplaced(ServletContextAttributeEvent scab) | Replace attribute from ServletContext object |
| public void contextDestroyed(ServletContextEvent event) | Remove event from ServletContext object |
| public void contentInitialized(ServletContextEvent event) | Initialize event to ServletContext object |

## Managing FAQs

### Class Diagram

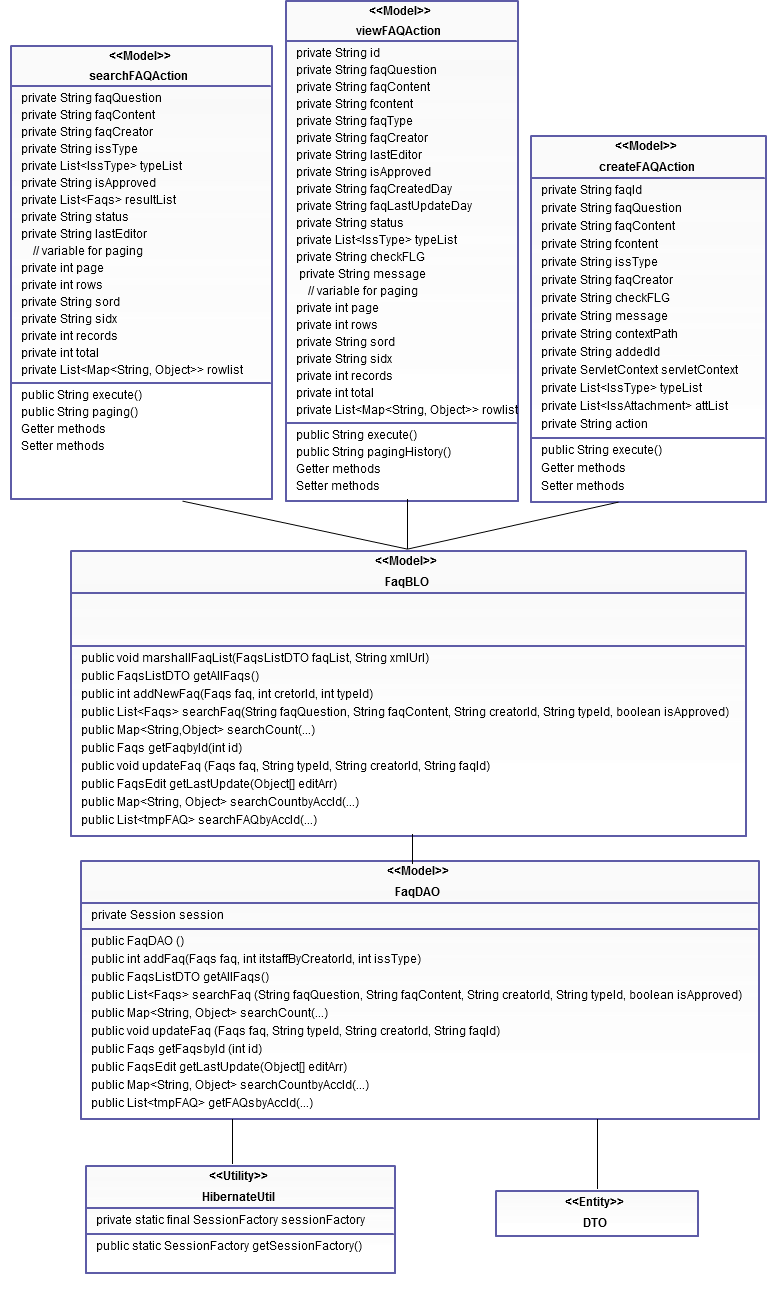


Figure 16 - ClassDiagram of Managing FAQ

### Class Diagram Explanation

|  |  |
| --- | --- |
| FaqDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public FaqDAO () | Constructor |
| public intaddFaq(Faqsfaq, intitstaffByCreatorId, intissType) | Add new FAQ |
| public FaqsListDTOgetAllFaqs() | Get all FAQs |
| public List<Faqs>searchFaq (String faqQuestion, String faqContent, String creatorId, String typeId, booleanisApproved) | Search FAQ by its question, content, creator, type, approvement |
| public Map<String,Object>searchCount(…) | Get total number of records and pages |
| public void updateFaq (Faqsfaq, String typeId, String creatorId, String faqId) | Update one FAQ |
| public FaqsgetFaqsbyId (int id) | Get FAQ by its id property |
| public FaqsEditgetLastUpdate(Object[] editArr) | Get last FaqsEdit object |
| public Map<String,Object>searchCountbyAccId(…) | Get total number of records and pages by account id |
| public List<tmpFAQ>getFAQsbyAccId(..) | Get FAQs by account id |

|  |  |
| --- | --- |
| FaqBLO | |
| Properties | |
| Methods | |
| public void marshallFaqList(FaqsListDTOfaqList, String xmlUrl) | Change all employees objects to XML format |
| public FaqsListDTOgetAllFaqs() | Get all FAQs |
| public intaddNewFaq(Faqsfaq, intcretorId, inttypeId) | Add new FAQ |
| public List<Faqs>searchFaq(String faqQuestion, String faqContent, String creatorId, String typeId, booleanisApproved) | Search FAQby its question, content, creator, type, approvement |
| public Map<String,Object>searchCount(…) | Get total number of records and pages |
| public FaqsgetFaqbyId(int id) | Get FAQ by its id |
| public void updateFaq (Faqsfaq, String typeId, String creatorId, String faqId) | Update one FAQ |
| public FaqsEditgetLastUpdate(Object[] editArr) | Get last FaqsEdit object |
| public Map<String,Object>searchCountbyAccId(…) | Get total number of records and pages by account id |
| public List<tmpFAQ>getFAQsbyAccId(..) | Get FAQs by account id |

|  |  |
| --- | --- |
| createFAQAction | |
| Properties | |
| private String faqId | Identification of FAQ |
| private String faqQuestion | Question of FAQ |
| private String faqContent | Content of FAQ |
| private String fcontent | Temporary input content |
| private String issType | Type of FAQ |
| private String faqCreator | Creator of FAQ |
| private String checkFLG | Flag which mode the method returns |
| private String message | Execution mode |
| private String contextPath | Project context path |
| private String addedId | Result of insert method |
| private ServletContextservletContext | ServletContext object |
| private List<IssType>typeList | List of FAQ type |
| private List<IssAttchment>attList | List of issue attachments |
| private String action | Process action |
| Methods | |
| public String execute() | Process action’s business logic |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| searchFAQAction | |
| Properties | |
| private String faqQuestion | Question of FAQ |
| private String faqContent | Content of FAQ |
| private String faqCreator | Creator of FAQ |
| private String issType | Type of FAQ |
| private StringisApproved | Status of FAQ (Approved or not) |
| private List<IssType>typeList | List of FAQ type |
| private List<Issue>resultList | List of FAQ result |
| private String status | Status for search FAQ page |
| private int page; | Current page |
| private int rows; | Total number of rows in one page |
| private String sord; | Sorting column |
| private String sidx; | Ascending or descending sort |
| private int records; | Number of returned records in grid from the request |
| private int total; | Total records |
| private List<Map<String,Object>>rowList | List of mapping between each row and each FAQ |
| Methods | |
| public String execute() | Process action’s business logic |
| pubic String paging() | Process paging |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| viewFAQAction | |
| Properties | |
| private String faqId | Identification of FAQ |
| private String faqQuestion | Question of FAQ |
| private String faqContent | Content of FAQ |
| private String fcontent | Temporary input content |
| private String faqCreator | Creator of FAQ |
| private String lastEditor | Last FAQ editor |
| private String faqType | Type of FAQ |
| private String createDay | Date of create FAQ |
| private String faqLastUpdateDay | Last FAQ update date |
| private String isApproved | Status of FAQ (Approved or not) |
| private String status | Status for view detailed FAQ page |
| private List<IssType>typeList | List of FAQ type |
| private String checkFLG | Flag which mode the method returns |
| private String message | Execution mode |
| private int page; | Current page |
| private int rows; | Total number of rows in one page |
| private String sord; | Sorting column |
| private String sidx; | Ascending or descending sort |
| private int records; | Number of returned records in grid from the request |
| private int total; | Total records |
| private List<Map<String,Object>>rowList | List of mapping between each row and each FAQ |
| Methods | |
| public String execute() | Process action’s business logic |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

## DTO Component

### Class Diagram

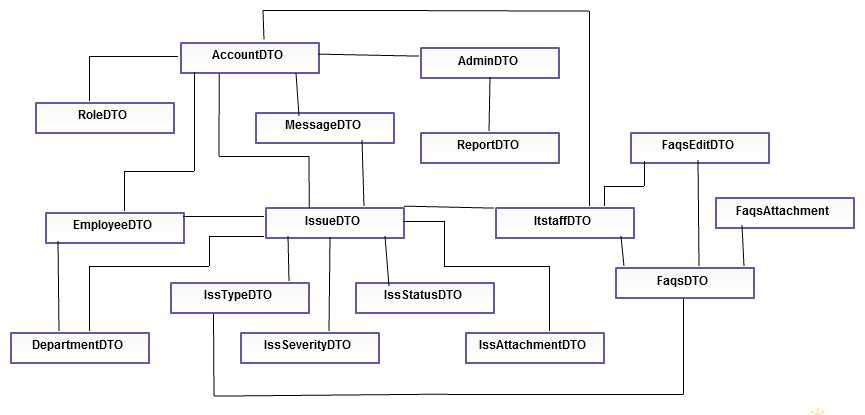


Figure 17 Class diagram of DTO component

### Class Diagram Explanation

|  |  |
| --- | --- |
| Account | |
| Properties | |
| private intaccId | Identification of account |
| private String username | Name of account |
| private String password | Password of account |
| private booleanisActive | Status of account |
| private String firstName | First name of user |
| private String lastName | Last name of user |
| private Integer dobday | User’s day of birth |
| private Integer dobmonth | User’s month of birth |
| private Integer dobyear | User’s year of birth |
| private Integer phone | Phone of user |
| private String email | Email of user |
| private String address | Address of user |
| private boolean gender | Gender of user |
| private Boolean isDelete | Deleted status of user |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Admin | |
| Properties | |
| private intadminId | Identification of admin |
| private Account account | Account of admin |
| private Boolean isDelete | Deleted status of admin |
| private Set<Report> reports | Reports of admin |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Department | |
| Properties | |
| private intdepId; | Identification of department |
| private String depFloor; | Floor of department |
| private String depRoom; | Room of department |
| private booleanisDelete; | Deleted status of department |
| private Set<Employee> employees | Employees that belong to department |
| private Set<Issue> issues | Issues that happen on department |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Employee | |
| Properties | |
| private intempId | Identification of employee |
| private Account account | Account of employee |
| private Department department | Department where employee works |
| private Integer experience | Number of year of employee’s experience |
| private Boolean isDelete | Deleted status of employee |
| private Set<Issue> issues | All issues of employee |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Faqs | |
| Properties | |
| private intfaqId | Identification of FAQ |
| private ItstaffitstaffByCreatorId | ItStaff who create FAQ |
| private IssTypeissType | Type of FAQ |
| private String faqQuestion | Question of FAQ |
| private String faqContent | Content of FAQ |
| private booleanfaqIsApproved | Approved status of FAQ |
| private Integer faqRating | Rating of FAQ |
| private intfaqCreateDay | Day that log FAQ |
| private intfaqCreateMonth | Month that log FAQ |
| private intfaqCreateYear | Year that log FAQ |
| private Boolean isDelete | Deleted status of FAQ |
| private Set<FaqsAttachment>faqsAttachments | All attachments of FAQ |
| private Set<FaqsEdit>faqsEdits |  |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| FaqsAttachment | |
| Properties | |
| private intattId | Identification of FAQ’s attachment |
| private Faqsfaqs | FAQ which attachment belong to |
| private String attUrl | Url of attachment |
| private booleanisDelete | Deleted status of attachment |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| FaqsEdit | |
| Properties | |
| private intfaqEditId | Identification of FAQ Edit |
| private Itstaffitstaff | ItStaff who edits FAQ |
| private Faqsfaqs | FAQ which has been editd |
| private intfaqLastUpdateDay | Day that update FAQ |
| private intfaqLastUpdateMonth | Month that update FAQ |
| private intfaqLastUpdateYear | Year that update FAQ |
| private Boolean isDelete | Deleted status of FAQ Edit |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| IssAttachment | |
| Properties | |
| private intattId | Identification of Issue’s attachment |
| private Issue issue | Issue which attachment belong to |
| private String attUrl | Url of attachment |
| private booleanisDelete | Deleted status of attachment |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| IssSeverity | |
| Properties | |
| private intsevId | Identification of severity |
| private String sevName | Name of severity |
| private String sevDiscription | Description of severity |
| private Set<Issue> issues |  |
| private booleanisDelete | Deleted status of severity |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| IssStatus | |
| Properties | |
| private intstatusId | Identification of issue’s status |
| private String statusName | Name of status |
| private String statusDiscription | Description of status |
| private booleanisDelete | Deleted status of status |
| private Set<Issue> issues |  |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| IssType | |
| Properties | |
| private inttypeId | Identification of issue’s type |
| private String typeName | Name of type |
| private String typeDiscription | Description of type |
| private Set<Itstaff>itstaffs | All itstaffs that belong to specify type |
| private Set<Faqs>faqses | All FAQs that belong to specify type |
| private Set<Issue> issues | All issues that belong to specify type |
| private booleanisDelete | Deleted status of type |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Issue | |
| Properties | |
| private intissId | Identification of issue |
| private IssSeverityissSeverity | Severity of issue |
| private IssStatusissStatus | Status of issue |
| private Account accountByLastEditor | Account of user who edits issue |
| private ItstaffissResponsor | Itstaff who fixes issue |
| private IssTypeissType | Type of issue |
| private Account issRegister | Account of user who log issue |
| private Employee issOwner | Owner of issue |
| private Department department | Department where issue happens |
| private String issName | Name of issue |
| private String issDescription | Description of issue |
| private intissPostDay | The posting day of issue |
| private intissPostMonth | The posting month of issue |
| private intissPostYear | The posting year of issue |
| private Integer issDlday | Day of issue’s deadline |
| private Integer issDlmonth | Month of issue’s deadline |
| private Integer issDlyear | Year of issue’s deadline |
| private Integer issEndDay | End day of issue |
| private Integer issEndMonth | End month of issue |
| private Integer issEndYear | End year of issue |
| private Boolean isDelete | Deleted status of issue |
| private Set<IssAttachment>issAttachments | All attachments of issue |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Itstaff | |
| Properties | |
| private intstaffId | Identification of itstaff |
| private Account account | Account of itstaff |
| private IssTypeissType | Issue’s type which itstaff can fix |
| private intissueCount | Number of issue that itstaff have been assigned |
| private Boolean isDelete | Deleted status of itstaff |
| private Set<FaqsEdit>faqsEdits | All FAQsEdits of itstaff |
| private Set<Issue> issues | All issues that itstaff have been assigned |
| private Set<Faqs>faqsesForCreatorId | All FAQs which itstaff create |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Message | |
| Properties | |
| private intmessId | Identification of message |
| private Account accountByEditor | Account of user who edit message |
| private Account accountByUserId | Owner of message |
| private Account accountBySenderId | Account of user who send message |
| private Issue issueId | Issue which related the message |
| private String messContent | Content of message |
| private intmessDay | Day that message created |
| private intmessMonth | Month that message created |
| private intmessYear | Year that message created |
| private String messTime | Time that message created |
| private booleanisSend | Email sent status of message |
| private Boolean isDelete | Deleted status of message |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Report | |
| Properties | |
| private intrepId | Identification of report |
| private Admin admin | Admin who creates report |
| private String repName | Name of report |
| private String repUrl | Url of report |
| private intrepCreateDay | Day that report created |
| private intrepCreateMonth | Month that report created |
| private intrepCreateYear | Year that report created |
| private Boolean isSend | Email sent status of report |
| private Boolean isDelete | Deleted status of message |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| Role | |
| Properties | |
| private introleId | Identification of role |
| private String roleName | Name of role |
| private String roleDiscription | Description of role |
| private Set<Account> accounts |  |
| private booleanisDelete | Deleted status of role |
| Methods | |
| Constructor methods | Initialize class variables |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

## Login Component

### Class Diagram

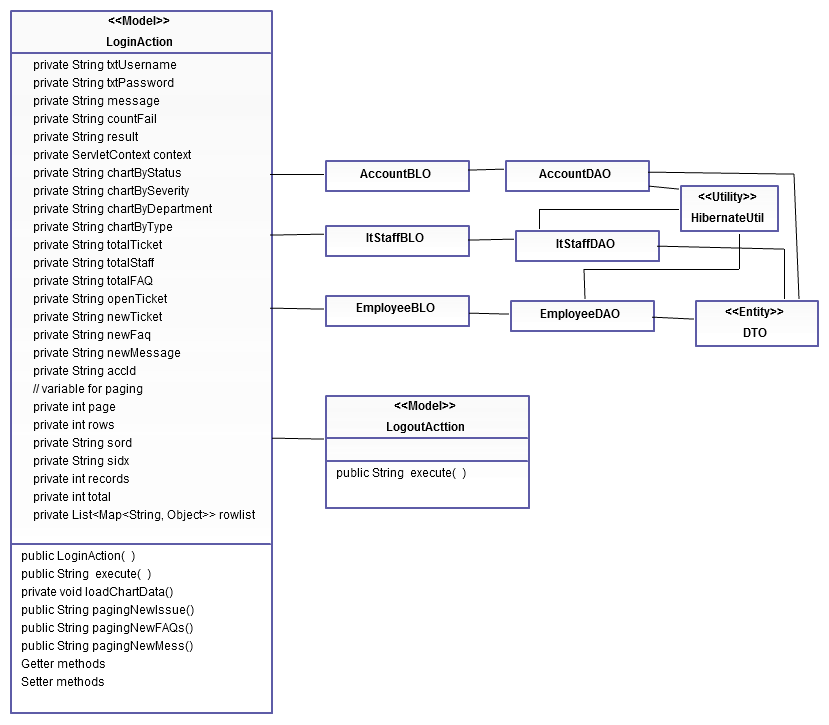


Figure 18 Login Class Diagram

### Class Diagram Explanation

|  |  |
| --- | --- |
| LoginAction | |
| Properties | |
| private String txtUsername | Username of account of user |
| private String txtPassword | Password of account of user |
| private String message | Execution mode |
| private String countFail | Number of login failed times |
| private String result | Flag which mode the method returns |
| private ServletContext context | ServletContext object |
| private String chartByStatus | Issue chart distributed by issue status |
| private String chartBySeverity | Issue chart distributed by issue severity |
| private String chartByDepartment | Issue chart distributed by issue department |
| private String chartByType | Issue chart distributed by issue type |
| private String totalTicket | Total issues |
| private String totalStaff | Total staffs |
| private String totalFAQ | Total FAQs |
| private String openTicket | Total open issues |
| private String newTicket | New issue from last login |
| private String new Faq | New FAQ from last login |
| private String newMessage | New message from last login |
| private String accId | Identification of account |
| private int page; | Current page |
| private int rows; | Total number of rows in one page |
| private String sord; | Sorting column |
| private String sidx; | Ascending or descending sort |
| private int records; | Number of returned records in grid from the request |
| private int total; | Total records |
| private List<Map<String,Object>>rowList | List of mapping between each row and each object |
| Methods | |
| public LoginAction( ) | Process action’s business logic |
| public String execute( ) | Constructor |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| AccountBLO | |
| Properties | |
| AccountDAOaccDao |  |
| Methods | |
| public AccountBLO() | Constructor |
| public Account getAccbyUNadnPW (String username, String password)) | Get account by its username and password |
| public Account getAccbyUserName (String username) | Get account by its username |
| public List<Account>getAllAccs() | Get list of all accounts |
| public void marshallAllAccs(String xmlURL) | Change all accounts objects to XML format |

|  |  |
| --- | --- |
| ItstaffBLO | |
| Properties | |
| Methods | |
| public List<Itstaff>getAllStaffs() | Get list of all it staffs |

|  |  |
| --- | --- |
| EmployeeBLO | |
| Properties | |
| Methods | |
| List<Employee>getAllEmpls() | Get list of all employees |
| public Employee getEmplbyAccId(intaccId) | Get employee by its account id |
| public void marshallAllEmpls(String xmlURL) | Change all employees objects to XML format |

|  |  |
| --- | --- |
| AccountDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public AccountDAO() | Constructor |
| public void save (Account account) | Save one account |
| public Account[] findByLastName (String lastname) | Find account by its lastname |
| public void delete (int id) | Delete one account by its id |
| public Account searchAccbyUNandPW(String username, String password) | Search account by its username and password |
| public List<Account>getAllAccs() | Get list of all accounts |

|  |  |
| --- | --- |
| EmployeeDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public EmployeeDAO () | Constructor |
| public List<Employee>getAllEmps() | Get list of all employees |
| public Employee getEmplsByAccId(intaccId) | Get employee by its accid |

|  |  |
| --- | --- |
| ItstaffDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public ItstaffDAO() | Constructor |
| public List<Itstaff>getAllStaffs() | Get list of all it staffs |
| public ItstaffgetStaffsByAccId(intacid) | Get it staff by his/her id |

## Managing user

### Class diagram

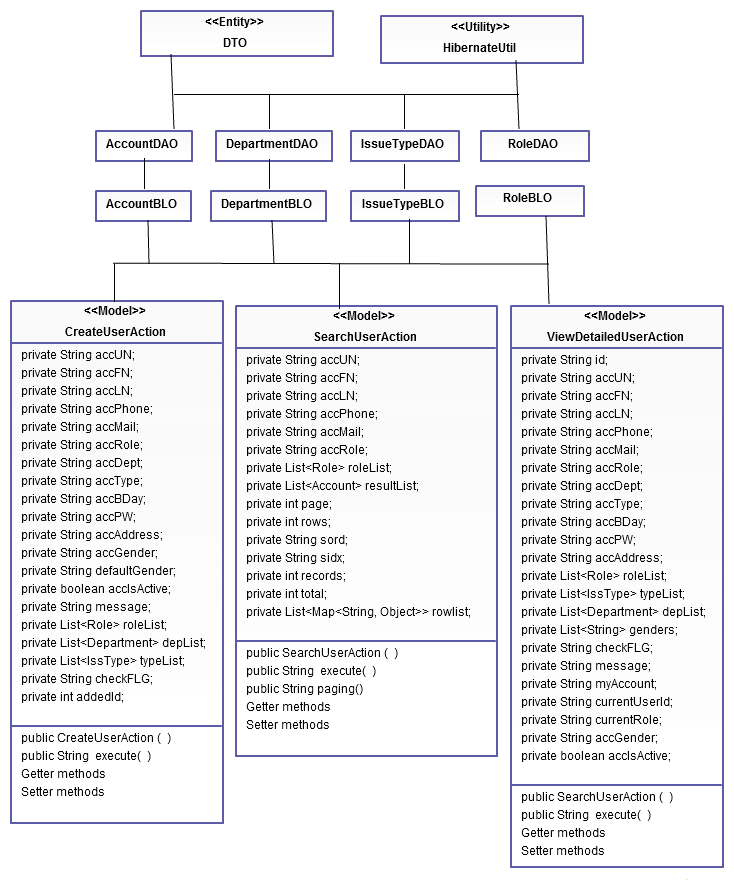


Figure 19 Class diagram of Managing user

### Class diagram explanation

|  |  |
| --- | --- |
| CreateUserAction | |
| Properties | |
| private String accUN; | Username of account |
| private String accFN; | First name of user |
| private String accLN; | Last name of user |
| private String accPhone; | Phone number of user |
| private String accMail; | Mail of user |
| private String accRole; | Role of user |
| private String accDept; | Department of user |
| private String accType; | Type of account |
| private String accBDay; | Birthday of user |
| private String accPW; | Password of account |
| private String accAddress; | Address of user |
| private String accGender | Gender of user |
| private String defaultGender | Default gender of user |
| private String accIsActive | Active status of user |
| private String message; | Execution mode |
| private List<Role>roleList; | List of role |
| private List<Department>depList; | List of department |
| private List<IssType>typeList; | List of type |
| private String checkFLG; | Flag which mode the method returns |
| private intaddedId; | Result of insert method |
|  |  |
| Methods | |
| public CreateUserAction ( ) | Constructor |
| public String execute( ) | Process action’s business logic |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |
|  | |

|  |  |
| --- | --- |
| SearchUserAction | |
| Properties | |
| private String accUN; | Username of account |
| private String accFN; | First name of user |
| private String accLN; | Last name of user |
| private String accPhone; | Phone number of user |
| private String accMail; | Mail of user |
| private String accRole; | Role of user |
| private List<Role>roleList; | List of role |
| private List<Account>resultList; | List of all result |
| private int page; | Current page |
| private int rows; | Total number of rows in one page |
| private String sord; | Sorting column |
| private String sidx; | Ascending or descending sort |
| private int records; | Number of returned records in grid from the request |
| private int total; | Total records |
| private List<Map<String,Object>>rowList | List of mapping between each row and each user |
| Methods | |
| public SearchUserAction ( ) | Constructor |
| public String execute( ) | Process action’s business logic |
| pubic String paging() | Process paging |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |

|  |  |
| --- | --- |
| ViewDetailedUserAction | |
| Properties | |
| private String id; | Identification of account |
| private String accUN; | Username of account |
| private String accFN; | First name of user |
| private String accLN; | Last name of user |
| private String accPhone; | Phone number of user |
| private String accMail; | Mail of user |
| private String accRole; | Role of user |
| private String accDept; | Department of user |
| private String accType; | Type of account |
| private String accBDay; | Birthday of user |
| private String accPW; | Password of account |
| private String accAddress; | Address of user |
| private List<Role>roleList; | List of role |
| private List<IssType>typeList; | List of type |
| private List<Department>depList; | List of department |
| private List<String> genders; | List of genders |
| private String checkFLG; | Flag which mode the method returns |
| private String message; | Execution mode |
| private String myAccount | Flag to check if user is view his/her account |
| private String currentUserId | Current user id |
| private String currentRole | Current user role |
| |  |  | | --- | --- | | AccountBLO | | | Properties | | | AccountDAOaccDao |  | | Methods | | | public AccountBLO() | Constructor | | public Account getAccbyUNadnPW (String username, String password)) | Get account by its username and password | | public Account getAccbyUserName (String username) | Get account by its username | | public List<Account>getAllAccs() | Get list of all accounts | | public void marshallAllAccs(String xmlURL) | Change all accounts objects to XML format | | public List<Account>searchAccount(String accUN, String accFN, String accLN,  String accPhone, String accMail, String roleId) | Get list of account by its username, first name, last name, phone, mail and role. | | public Account getAccountbyId(int id) | Get account by its Id | | public void updateAccount(Account account, String roleID, String accID) |  | | public intcreateAccount(Account account, String roleId, String depId, String typeId) | Create new account | | public void updateAccount(Account account, String accId, String roleId, String typeId, String depId) | Update existed account |   private String accGender | Gender of user |
| private booleanaccIsactive | Active status of user |
| Methods | |
| public ViewDetailedUserAction ( ) | Constructor |
| public String execute( ) | Process action’s business logic |
| Getter methods | Get value of variables |
| Setter methods | Set value of variables |
|  | |

|  |  |
| --- | --- |
| DepartmentBLO | |
| Properties | |
| Methods | |
| List<Department>getAllDepts() | Get list of all departments |

|  |  |
| --- | --- |
| IssTypeBLO | |
| Properties | |
| Methods | |
| public List<IssType>getAllTypes() | Get list of all issue’s types |

|  |  |
| --- | --- |
| AccountDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public AccountDAO() | Constructor |
| public void save (Account account) | Save one account |
| public Account[] findByLastName (String lastname) | Find account by its lastname |
| public void delete (int id) | Delete one account by its id |
| public Account searchAccbyUNandPW(String username, String password) | Search account by its username and password |
| public List<Account>getAllAccs() | Get list of all accounts |
| public Account getAccountByID(intaccId) | Get account by its Id |
| public void updateAccount(Account account, String roleId, String accId) | Update existed account |
| public intcreateAccount(Account account, String roleID) | Create new account |

|  |  |
| --- | --- |
| DepartmentDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public DepartmentDAO () | Constructor |
| public List<Department>getAllDepts() | Get list of all departments |

|  |  |
| --- | --- |
| IssTypeDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| public IssTypeDAO () | Constructor |
| public List<IssType>getAllTypes() | Get list of all issue’s types |

## Managing Report

### Class diagram

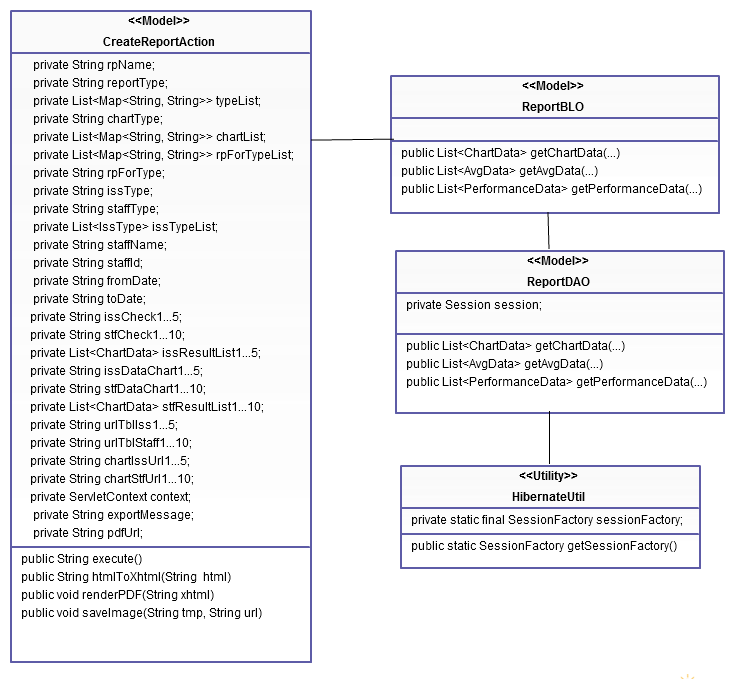


Figure 20 Class diagram of Managing Report

### Class diagram explanation

|  |  |
| --- | --- |
| ReportDAO | |
| Properties | |
| private Session session | Session of hibernate |
| Methods | |
| Public ReportDAO () | Constructor |
| public List<ChartData>getChartData(…) | Get data of chart (type: number of issue) |
| public List<AvgData>getAvgData(…) | Get data of chart (type: average resolve time) |
| public List<PerformanceData>getPerformanceData | Get data of chart (type: staff performance) |

|  |  |
| --- | --- |
| ReportBLO | |
| Properties | |
|  |  |
| Methods | |
| public List<ChartData>getChartData(…) | Get data of chart (type: number of issue) |
| public List<AvgData>getAvgData(…) | Get data of chart (type: average resolve time) |
| public List<PerformanceData>getPerformanceData | Get data of chart (type: staff performance) |

|  |  |
| --- | --- |
| CreateReportAction | |
| Properties | |
| private String rpName; | Name of report |
| private String reportType; | Type of report |
| private List<Map<String, String>>typeList; | List of report’s type |
| private String chartType; | Type of chart |
| private List<Map<String, String>>chartList; | List of chart |
| private List<Map<String, String>>rpForTypeList; | List of mapping for type of report |
| private String rpForType; | Flag to check type of report |
| private String issType; | Type of issue |
| private String staffType; | Type of staff |
| private List<IssType>issTypeList; | List of issue’s type |
| private String staffName; | Name of staff |
| private String staffId; | Id of staff |
| private String fromDate; | Start day of report |
| private String toDate; | End day of report |
| private String issCheck1...5; | Flag for checking metric (report type : issue) |
| private String stfCheck1...10; | Flag for checking metric (report type : staff) |
| private List<ChartData> issResultList1...5; | Result table of corresponding chart (report type: issue) |
| private String issDataChart1...5; | Data of corresponding chart (report type : issue) |
| private String stfDataChart1...10; | Data of corresponding chart (report type : staff) |
| private List<ChartData> stfResultList1...10; | Result table of corresponding chart (report type: staff) |
| private String urlTblIss1...5; | Flag for checking metric |
| private String urlTblStaff1...10; | Flag for checking metric |
| private String chartIssUrl1...5; | Flag for checking metric |
| private String chartStfUrl1...10; | Flag for checking metric |
| private ServletContext context; | Servlet context |
| private String exportMessage; | Message that will be exported |
| private String pdfUrl; | Url of exported pdf |
| Methods | |
| public String execute() | Process action’s business logic |
| public String htmlToXhtml(String html) | Convert html to xhtml |
| public void renderPDF(String xhtml) | Render PDF from xhtml |
| public void saveImage(String tmp, String url) | Save image |

# Sequence Diagram

## Managing Issue Function:

### Log Basic Issue:

* + - This feature allows normal users (employees of company) which do not have much IT knowledge can log their IT issue with basic information.
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files.

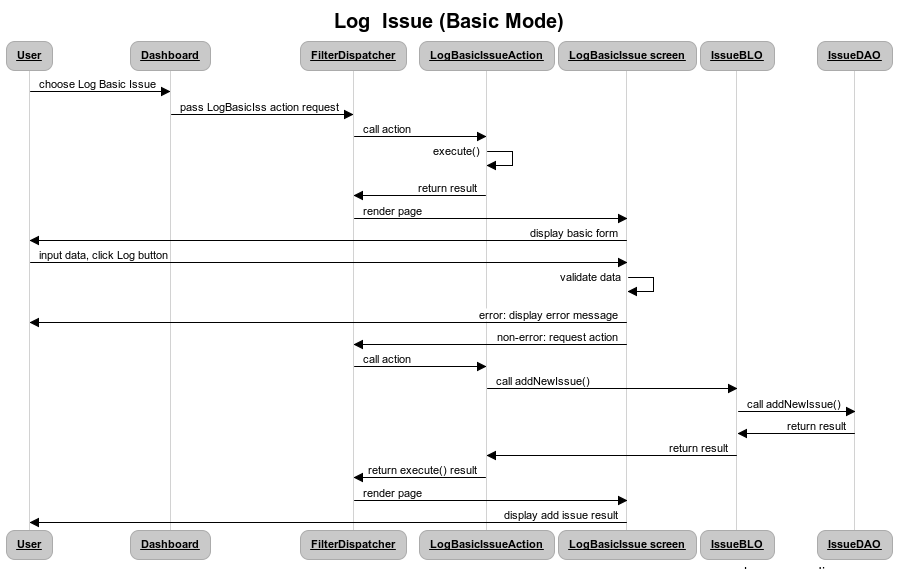


Figure 21 - Log Basic Issue SD

### Log Advanced Issue:

* + - This feature allows users who have IT knowledge like Admin, or IT staff can log issue received from company employees with more detailed information.
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files

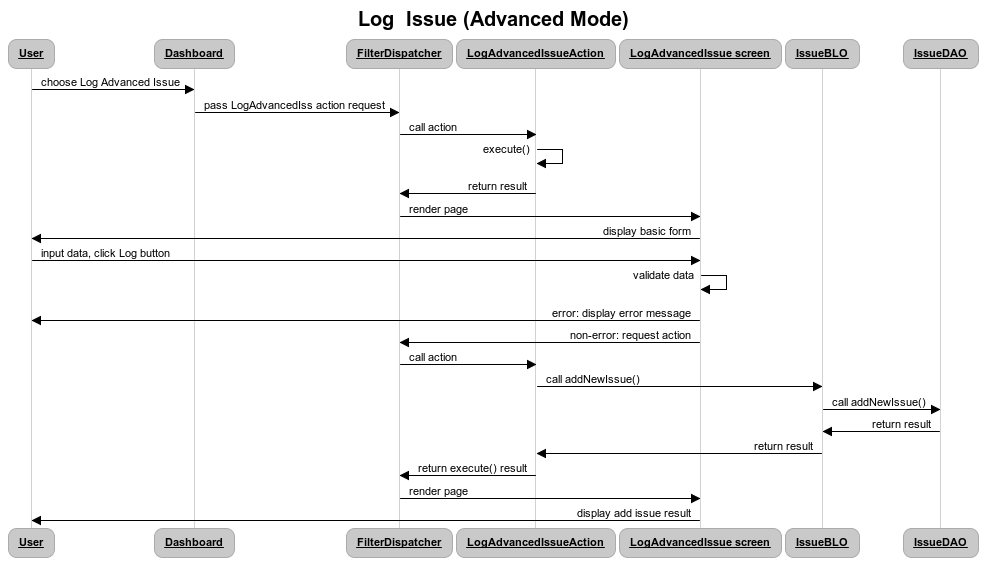


Figure 22 - Log Advanced Issue SD

### Search Issue:

* + - This feature helps users to search their needed issues with a lot of detailed criterions.
    - From searched issue list, user can direct to view issue info by clicking on issue name.

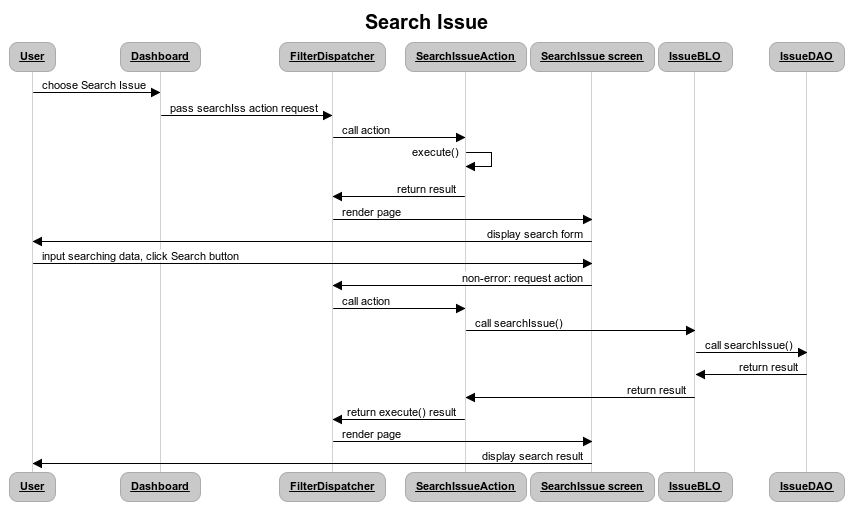


Figure 23 - Search Issue SD

### Edit Issue:

* + - This feature allows users to edit their related Issue information when needed.
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files
    - After Edit, editor name and edited time will be updated in database

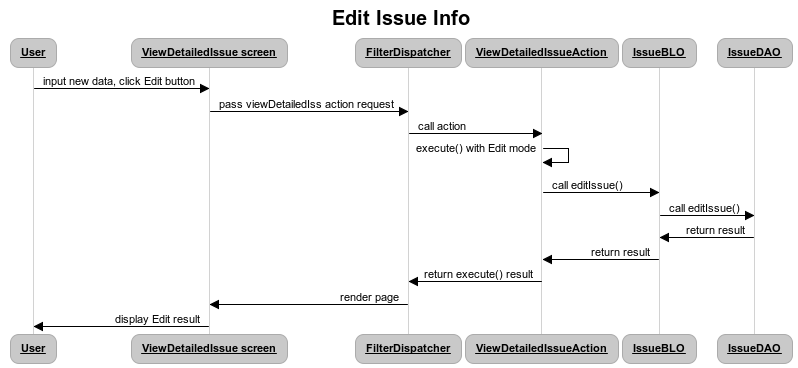


Figure 24 - Edit Issue SD

### View Detailed Issue:

* + - This feature helps users to view information of Issue which they are interested in. User can view detailed issue in some cases like after logging issue, or searching issue
    - User also can view issue history to know who are the last editor and when.

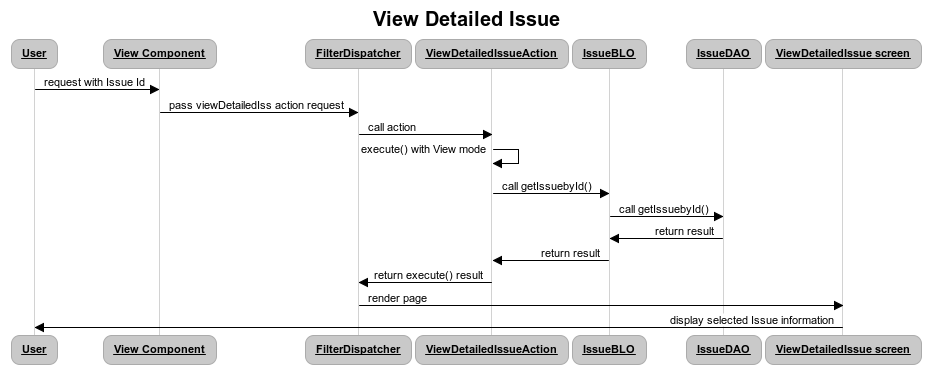


Figure 25 - View Detailed Issue SD

## Managing User:

### Create new user:

* + - This feature allows admin to create new accounts for employees who recently joined company. These accounts will include most of contact information of employee like phone, email, address, department…
    - Based on role of new account, called method in AccountDAO may be createAccount()
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files
    - User password will be hashed with md5 algorithm to prevent hacking



Figure 26 - Update new user SD

### Search user:

* + - This feature helps admin to find wanted users with many detailed criteria like name, address, department…
    - From searching result list, user can view user information by clicking on user name

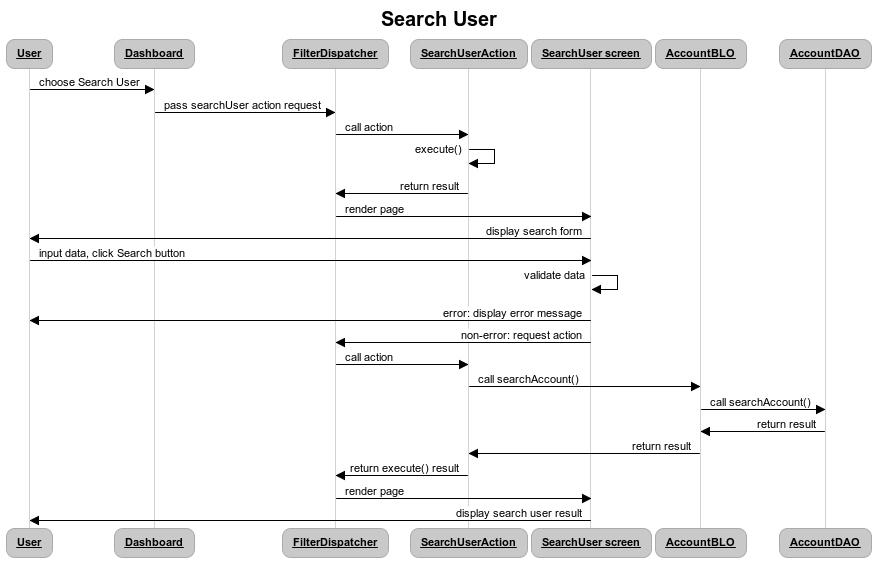


Figure 27 - Search User SD

### View user information:

* + - This feature allows admin to view information of other users after searching or creating user. Each user also can view his/her information.

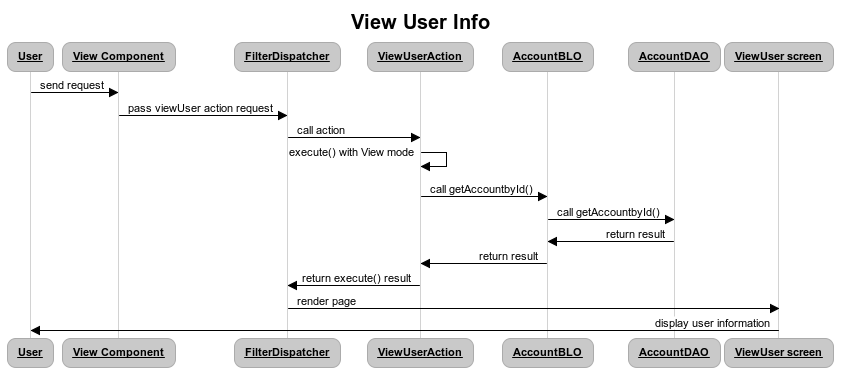


Figure 28 - View User SD

### Edit user information:

* + - This feature helps admin to edit user information when changes happen. Each user also has right to edit their account information.
    - Admin can choose user which needs to edit after doing user searching and viewing. Normal user can use My Account feature in Self-help service to modify account information.
    - Based on role of edited account, called method in AccountDAO may be updateAdmin (), updateStaff (), or updateEmployee ().
    - Edit detailed user also validate input data by JavaScript and XML.



Figure 29 - Edit detailed user SD

## Managing Report:

### Produce report:

* + - Based on the statistic of issue types, distribution and status, this feature helps admin to generate detailed reports including performance statistic like number of fixed bugs in one month, percentage of bug in each technology field
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files

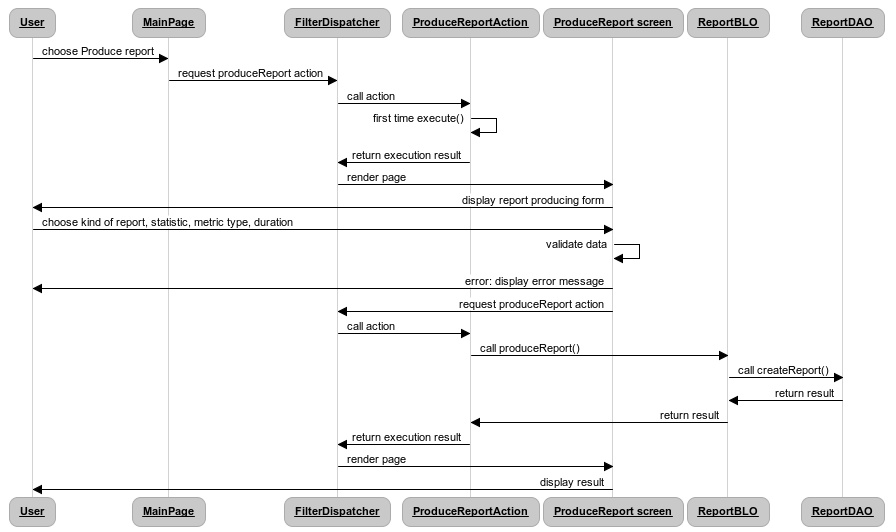


Figure 30 - Produce report SD

### Export to PDF:

* + - This feature allows admin to export their generated report to PDF, and save in their computer

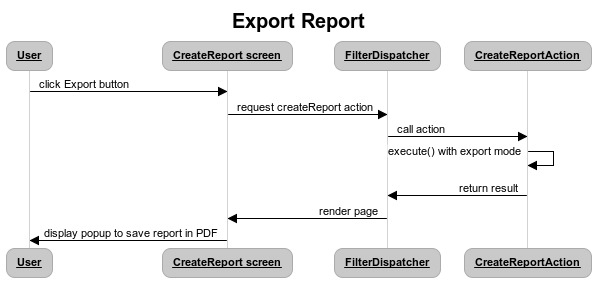


Figure 31 - Export Report to PDF

## Managing message, email notification

### Send message

* + - This feature helps users to view their incoming and outgoing message
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files

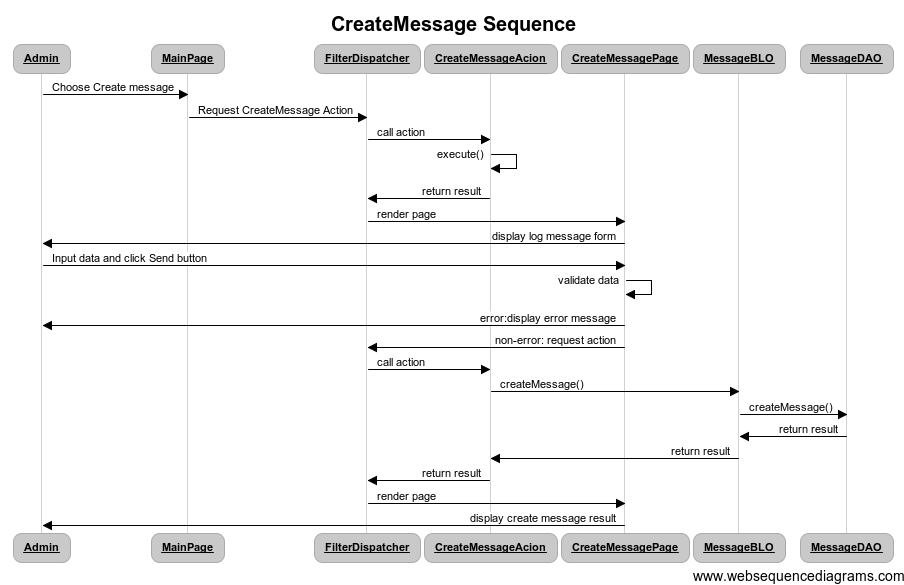


Figure 32 – Create Message SD

### View message

* + - This feature helps users to view their incoming and outgoing message.

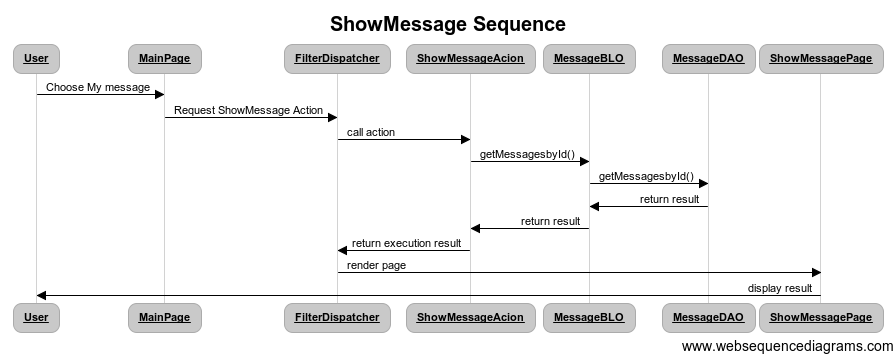


Figure 33 – Show Message SD

### Mail notification

* + - This feature helps users in tracking their own issue. When an issue is updated, an email will be sent to owner of this issue to inform about change.

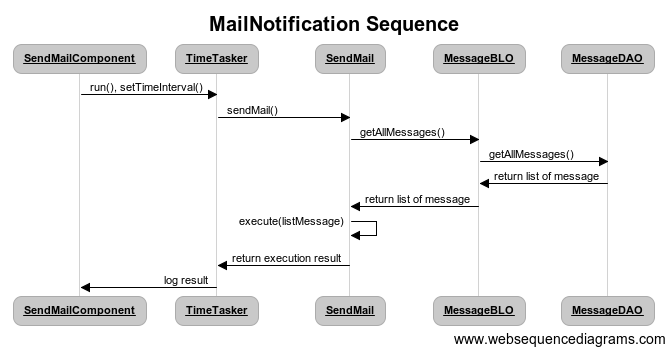
****

Figure 34 – Mail Notification SD

## Managing FAQs

### View FAQs

* + - This feature allows users to view detailed information of FAQs after searching or creating FAQs.

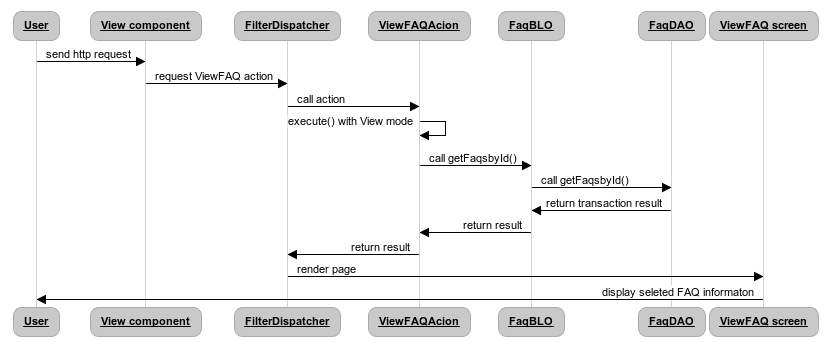


Figure 35 – View FAQs SD

### Create FAQS

* + - This feature allows ItStaffs create new FAQs topic.
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files.

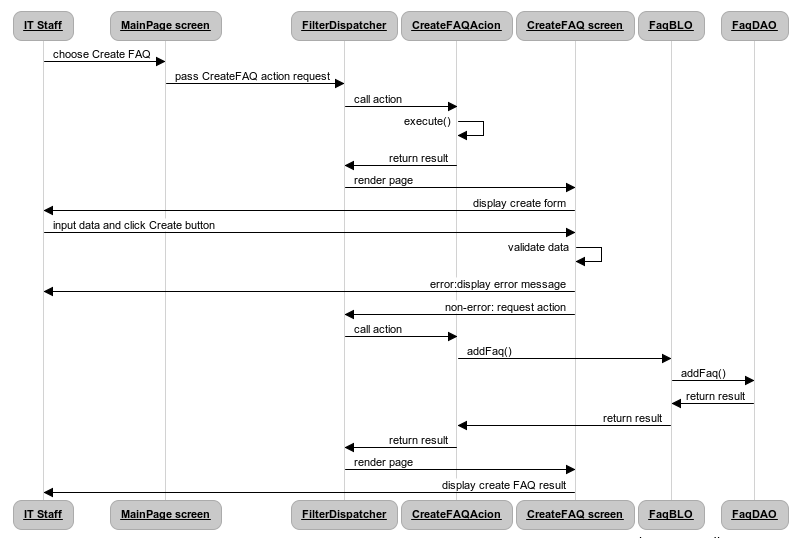


Figure 36 – Create FAQs SD

### Search FAQs

* + - This feature helps users to search FAQs with a lot of detailed criterions.

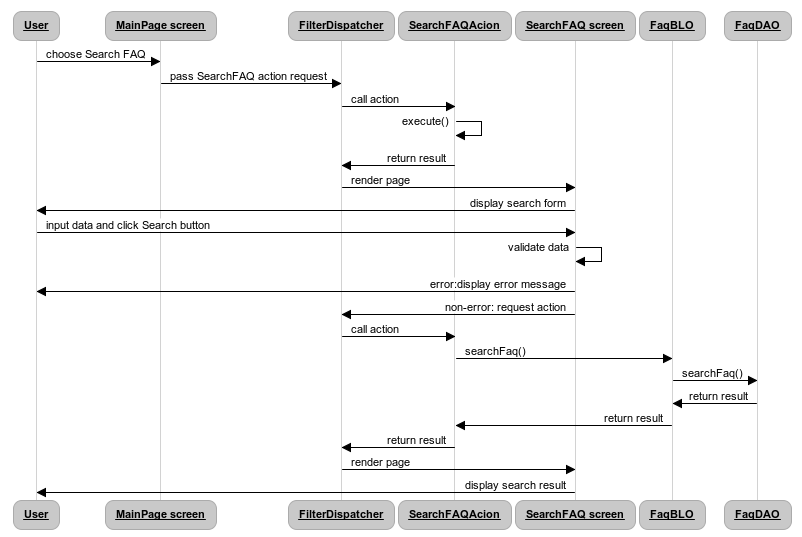


Figure 37 – Search FAQs SD

### Edit FAQs

* + - This feature allows ItStaff and Admin to edit FAQs information when needed.
    - Validating input data will be done in JavaScript based on restrictions of corresponding attributes in database and data in XML files

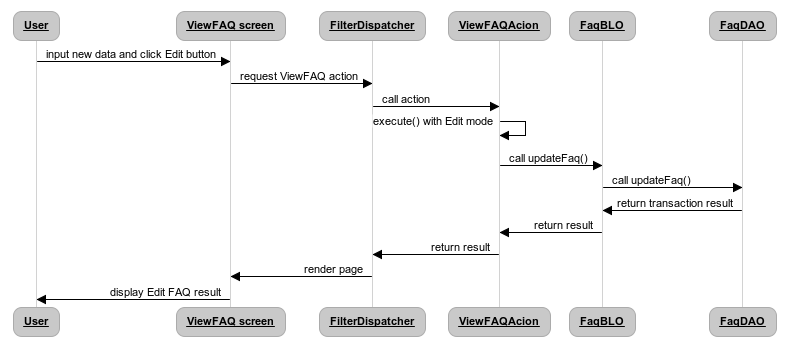


Figure 38 - Edit FAQs SD

# User Interface Design

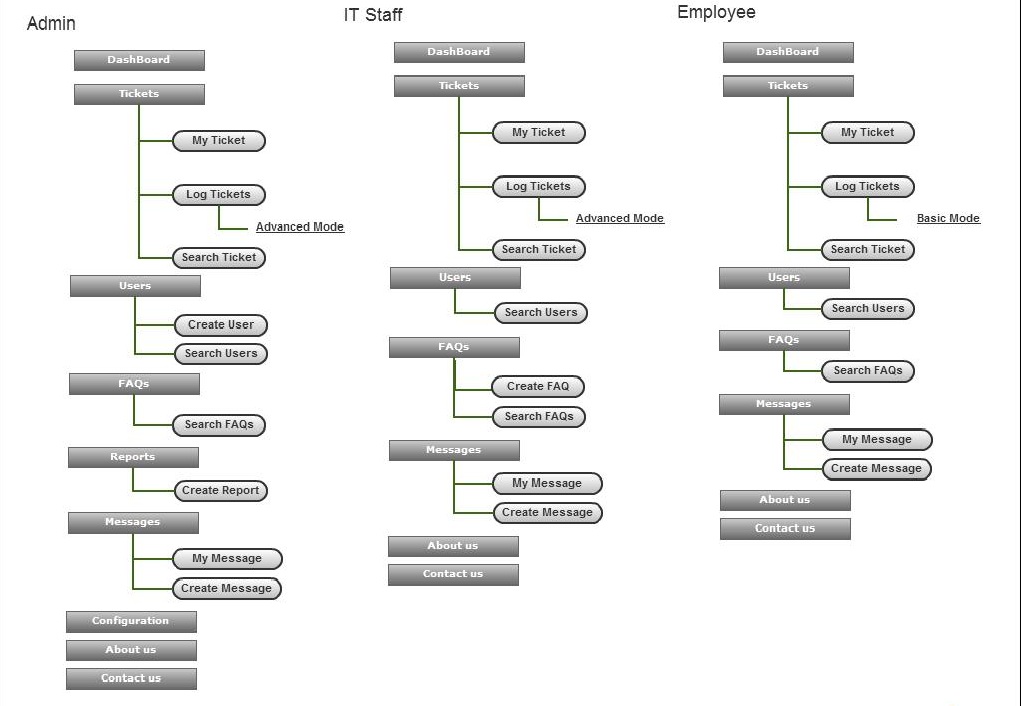
****

Figure 39 – Sitemap

## Login

### Screen Image



Figure 40 – Login Screen

### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtUsername  txtPassword  btnLogin  txtSysMessage | User types username to login  User types password to login  User clicks the button to login to the system  System displays the error message to user. (Only display when there’s any message) |

## Issue Managing

### Log Ticket: Basic Mode

#### Screen Image

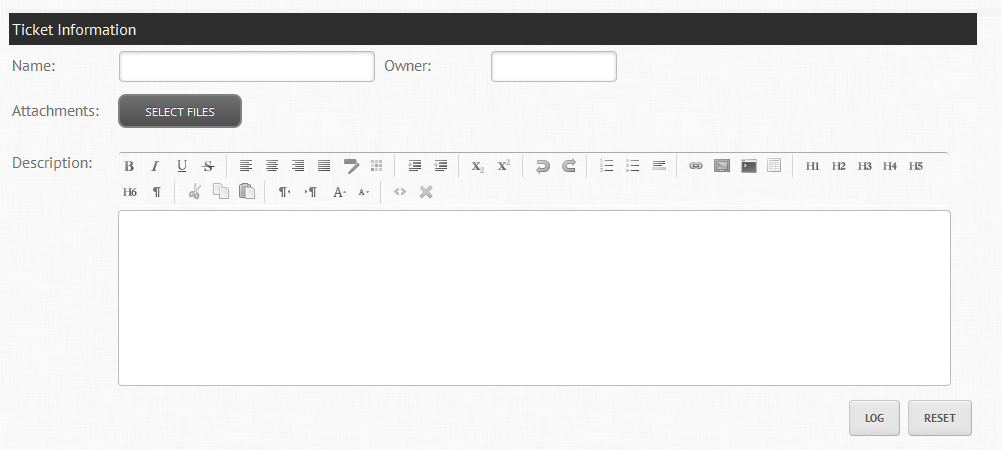


Figure 41 – Log Ticket: Basic Mode Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtIssName | Users types issue name to log issue |
| txtIssOwner | Users types issue’s owner to log issue |
| btnAttach | User click to upload issue’s attachments to log issue |
| txtIssDescrip | Users type issue’s description to log issue |
| btnLog | User clicks the button to Log issue |
| btnReset | User clicks the button to clear all input fields |
| txtSysMessage | System displays the error message to user. (Only display when there’s any message) |

### Log Advanced Issue

#### Screen Image

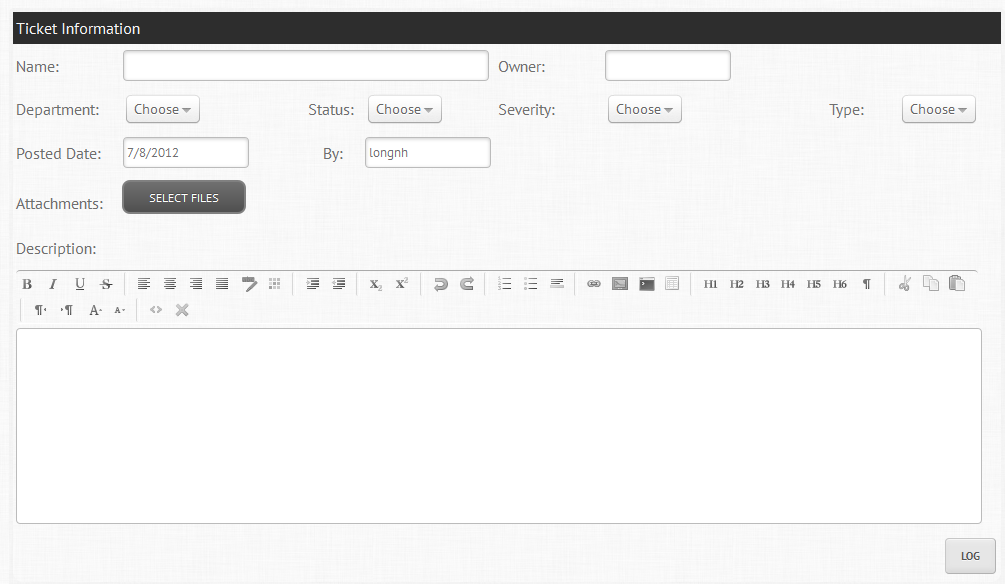


Figure 42 – Log Ticket: Advanced Mode Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtIssName | Users types issue name to log issue |
| txtIssOwner | Users types issue’s owner to log issue |
| cbDepartment | Users selects department to log issue |
| cbIssStatus | Users selects issue’s status to log issue |
| cbIssSeverity | Users selects issue’s severity to log issue |
| cbIssType | Users selects issue’s type to log issue |
| txtPostedDate | Users types issue’s posted date to log issue |
| txtPostedBy | Users types issue’s posted user to log issue |
| btnIssAttach | User click to upload issue’s attachments to log issue |
| txtIssDescript | Users types issue’s description to log issue |
| txtIssResponsor | User types issue’s respondent to log issue |
| txtDeadline | User choose issue’s deadline to log issue |
| btnLog | User clicks the button to Log issue |
| txtSysMessage | System displays the error message to user. (Only display when there’s any message) |

### Search Issue

#### Screen Image

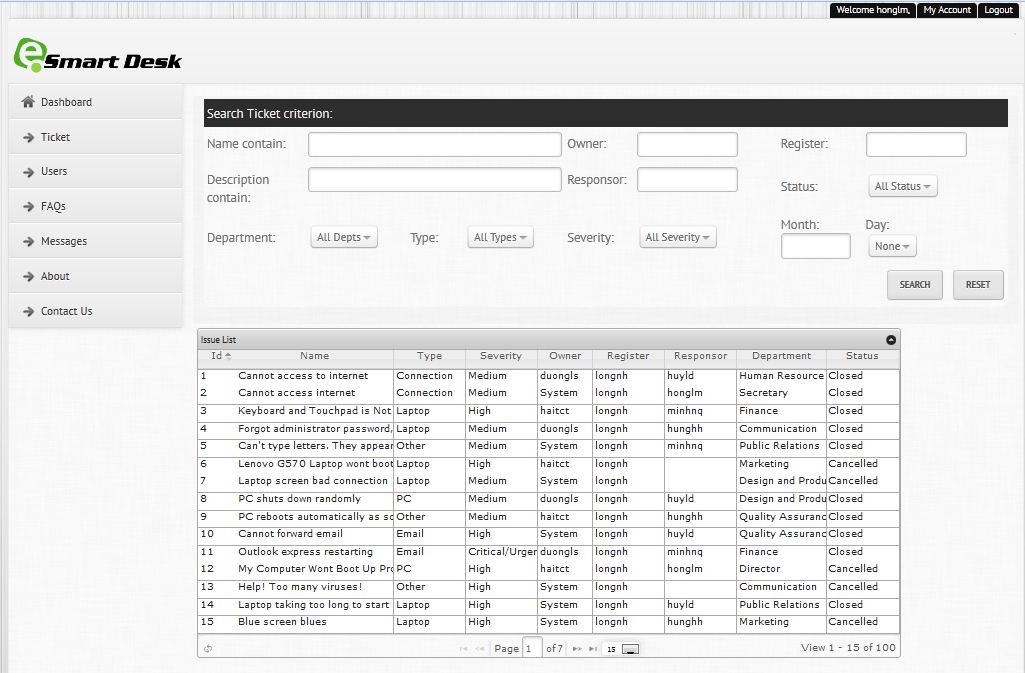


Figure 43 – Search Issue Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtName | User types issues name keyword to search |
| txtOwner | User types issue’s owner keyword to search |
| txtRegister | User types issue’s register keyword to search |
| txtDescrip | User types issue’s description keyword to search |
| txtResponser | User types issue’s respondent keyword to search |
| cbStatus | User selects issue’s status to search |
| cbDepartment | User selects issue’s department to search |
| cbType | User selects issue’s type to search |
| cbSeverity | User selects issue’s severity to search |
| btnSearch | User clicks the button to Search |
| tbResult | Table contains all search result |

### View/Edit Issue

#### Screen Image

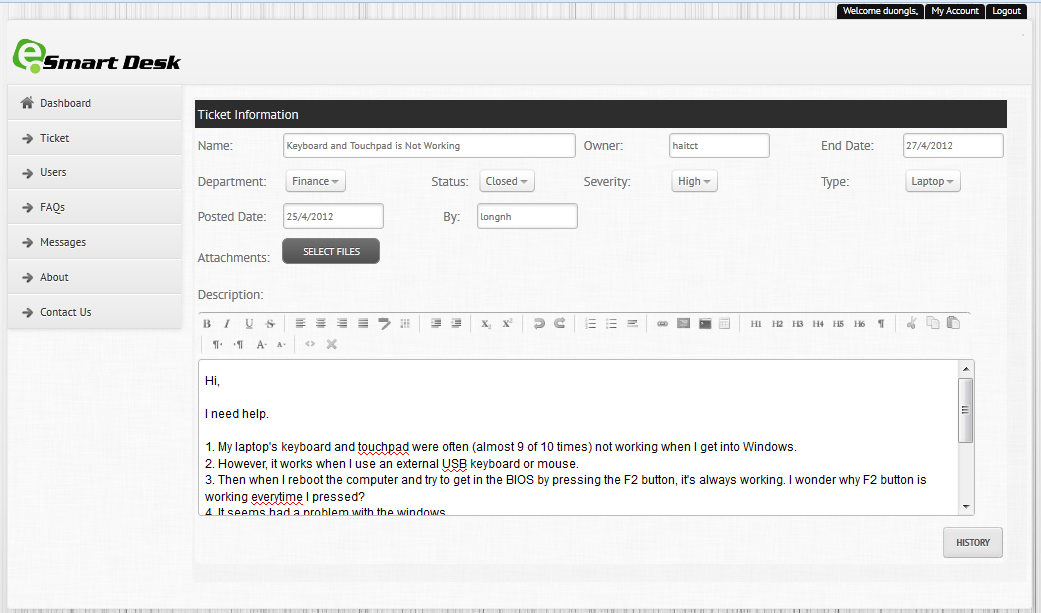


Figure 44 – View Issue Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtName | System displays the issue’s name for user to view  User changes the issue’s name to edit |
| txtOwner | System displays the issue’s owner for user to view  User changes the issue’s owner to edit |
| cbDepartment | System displays the owner’s department for user to view  User changes the owner’s department to edit |
| cbStatus | System displays the issue’s status for user to view  User changes the issue’s status to edit |
| cbSeverity | System displays the issue’s severity for user to view  User changes the issue’s severity to edit |
| cbType | System displays the issue’s type for user to view  User changes the issue’s type to edit |
| txtPostedDate | System displays the issue’s posted date for user to view |
| txtPostBy | System displays the issue’s posted user for user to view |
| txtAssignedTo | System displays the issue’s assigned staff for user to view  User changes the issue’s assigned staff to edit |
| txtDeadline | System displays the issue’s deadline for user to view  User changes the issue’s deadline to edit |
| btnAttach | User click to upload issue’s attachments to log issue |
| txtDescrip | System displays the issue’s description for user to view  User changes the issue’s description to edit |
| txtEndDate | System display ended date of issue |
| btnHistory | User click button to view updated history |
| btnEdit | User click button to edit |
| txtSysMessage | System displays the error message to user. (Only display when there’s any message) |

## User Managing

### Create User

#### Screen Image

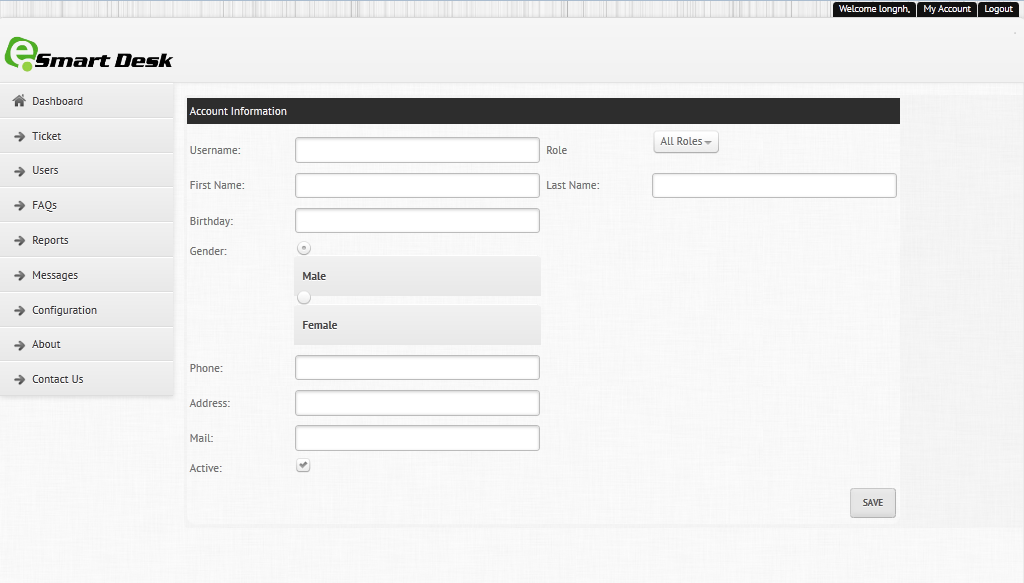


Figure 45 – Create User Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtUserName | Users types account’s user name to create |
| txtFirstName | Users types account’s first name to create |
| txtLastName | Users types account’s last name to create |
| txtBDay | Users types account’s birthday to create |
| rdGender | User checks Male/Female to choose Gender |
| txtPhone | Users types account’s phone to create |
| txtAdd | Users types account’s address to create |
| txtMail | Users types account’s e-mail to create |
| ckIsActive | User check to create Active Account, uncheck to create inactive Account |
| cbRole | User selects account role to create |
| cbType | User selects account IT type to create(Only for role IT Staff) |
| cbDepartment | User selects account department to create (Only for role employee) |
| btnSave | User clicks the button to save |
| txtSysMessage | System displays the error message to user. (Only display when there’s any message) |

### Search User

#### Screen Image

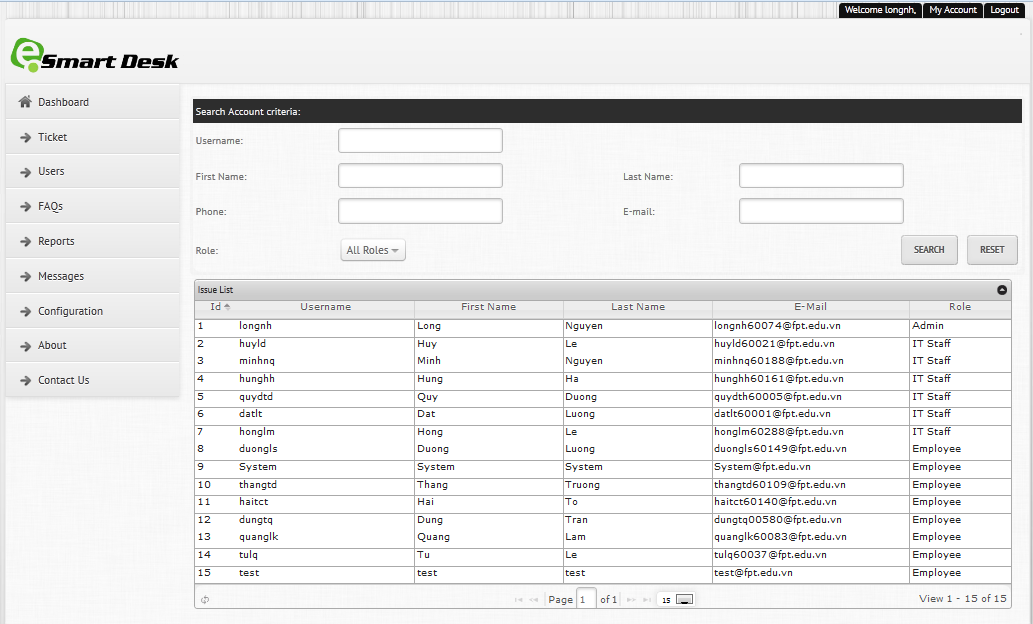


Figure 46 – Search User Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtUserName | User types account’s username keyword to search |
| txtFirstName | User types user’s first name keyword to search |
| txtLastName | User types user’s last name keyword to search |
| txtPhone | User types user’s phone keyword to search |
| txtMail | User types user’s mail keyword to search |
| cbRole | User selects user’s role to search |
| btnSearch | User clicks the button to Search |
| tbResult | Table contains all search result |

### View/Edit User

#### Screen Image

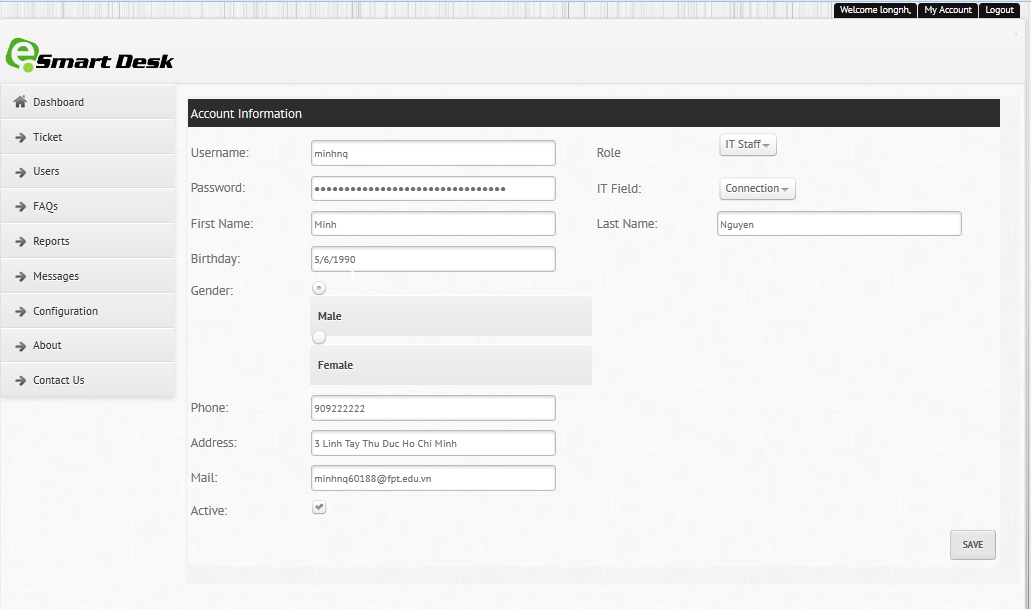


Figure 47 – View User Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtUsername | System displays the user name for user to view |
| txtPassword | User changes the account password to edit |
| txtFirstName | System displays the first name for user to view  User changes the first name to edit |
| txtLastName | System displays the last name for user to view  User changes the last name to edit |
| cbRole | System displays the user’s role for user to view  User selects the role to edit |
| cbItType | System displays the IT fieldtype for user to view (only for Staff Account)  User selects the IT Field to edit |
| cbDepartment | System displays the user’s department for user to view (only for Employee Account)  User selects the Department to edit |
| txtBirthday | System displays the birthday for user to view  User changes the birthday to edit |
| rdGender | System displays user’s gender for user to view  User changes user’s gender to edit |
| txtPhone | System displays the account’s phone for user to view  User changes the account’s phone to edit |
| txtAdd | System displays the account’s address for user to view  User changes the account’s address to edit |
| txtMail | System displays the account’s mail for user to view  User changes the account’s mail to edit |
| ckActive | System displays the account’s Active for user to view  User changes the account’s Active to edit |
| btn Edit | User click button to edit |
| txtSysMessage | System displays the error message to user. (Only display when there’s any message) |

## FAQ Managing

### Create FAQ

#### Screen Image

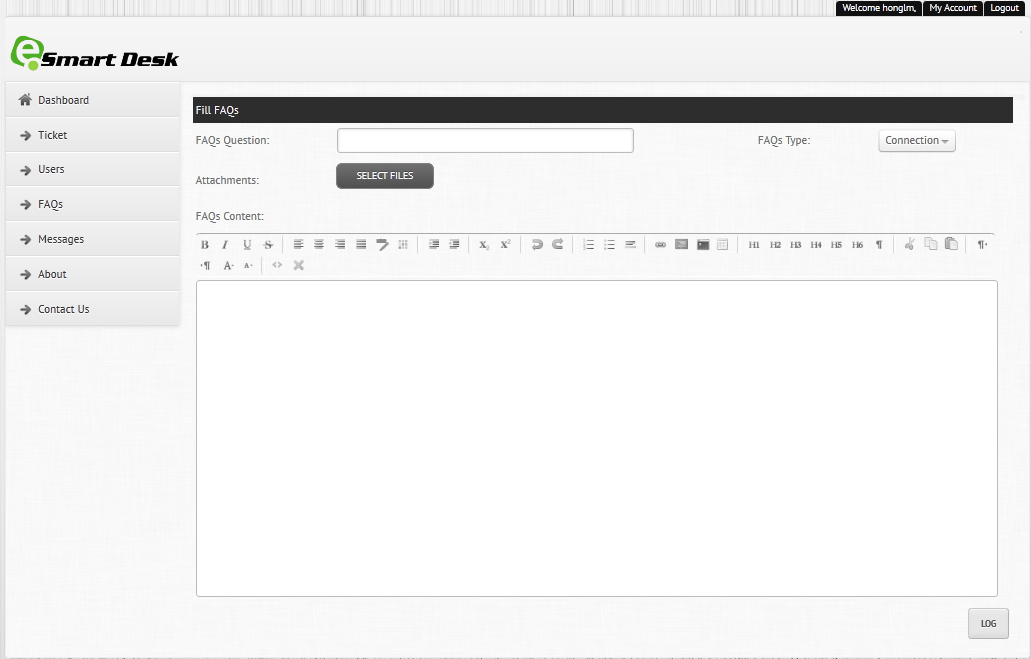


Figure 48 – Create FAQs Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtQuestion | Users types FAQ to create |
| cbType | User chooses FAQ’s type to create |
| btAttach | User click to upload FQA’s attachments to create |
| txtContent | Users types FAQ’s content to create |
| btnLog | User clicks the button to log |
| txtSysMessage | System displays the error message to user. (Only display when there’s any message) |

### Search FAQs

#### Screen Image

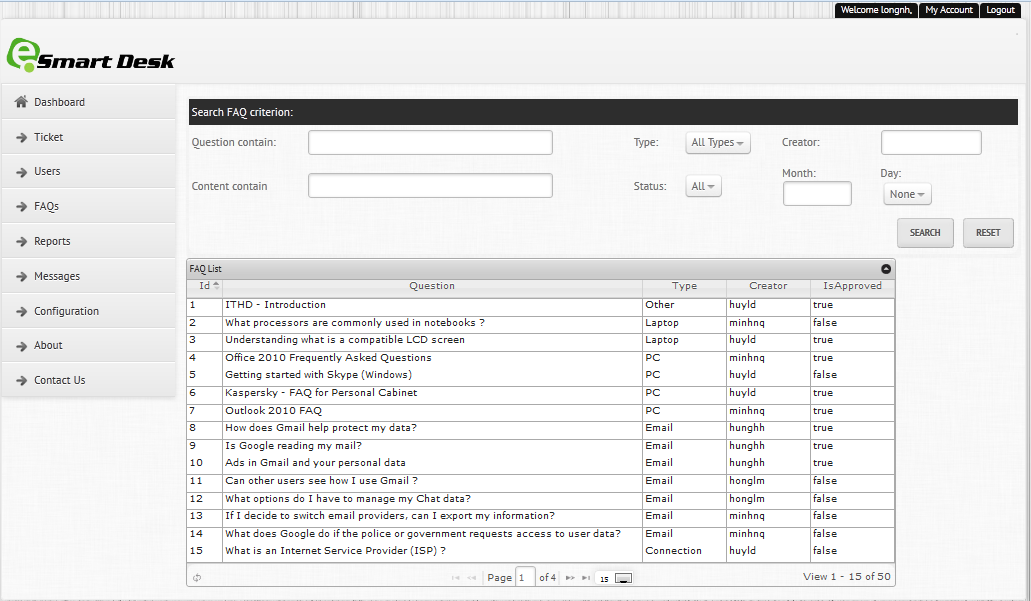


Figure 49 – Search FAQs Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtQuestContain | User types FAQ question keyword to search |
| txtCreator | User types FAQ creator keyword to search |
| cbType | User selects FAQ type to search |
| txtContentContain | User types FAQ content keyword to search |
| cbStatus | User selects FAQ status to search |
| btnSearch | User clicks the button to Search |
| tbResult | Table contains all search result |

### View/Edit FAQ

#### Screen Image

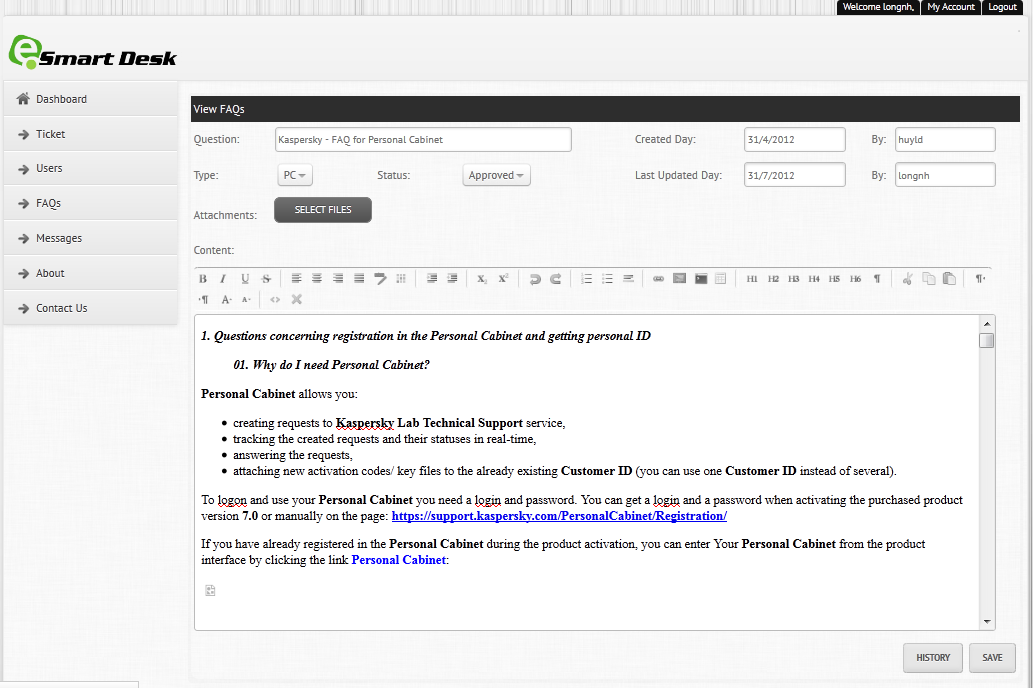


Figure 50 – View FAQs Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtQuestion  txtContent  txtCreator  txtCreatorDate  txtLastEditor  txtLastUpdate  cbType  cbStatus  btn Edit  txtSysMessage | System displays the FAQ’s question for user to view  User changes the FAQ’s question to edit  System displays the FAQ’s content for user to view  User changes the FAQ’s content to edit  System displays the FAQ’s creator for user to view  System displays the FAQ’s create day for user to view  System displays the FAQ’s last editorfor user to view  System displays the FAQ’s last updatefor user to view  System displays the FAQ’s type for user to view  User changes the type to edit  System displays the FAQ’s status for user to view  User changes the status to edit  User click button to edit  System displays the error message(if there’s any) to user |

## Message Managing

### Create Message

#### Screen Image

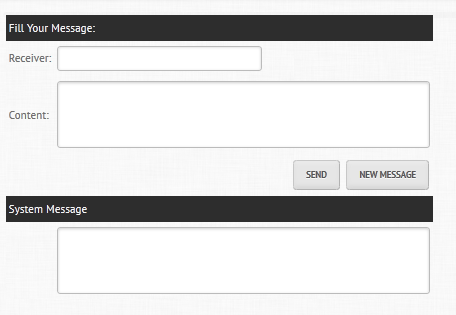


Figure 51 – Create Message Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| txtReceiver | Users types message’s receiver |
| txtContent | Users types message’s content |
| btnSend | User clicks the button to send message |
| btnNew | User clicks the button to clear all entered content |
| txtSysMessage | System displays the error message to user. (Only display when there’s any message) |

### My Message

#### Screen Image

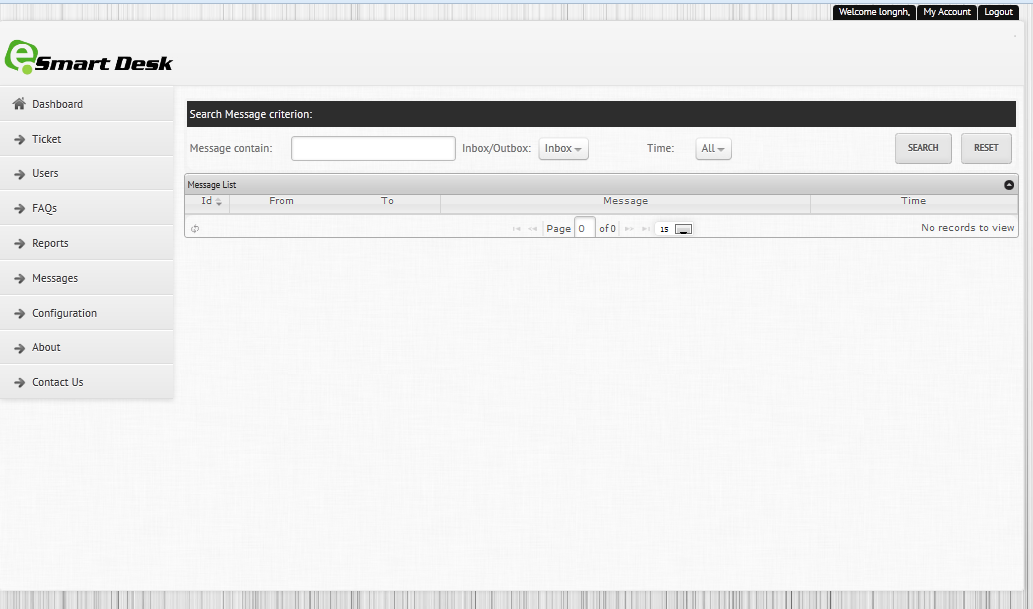


Figure 52 – My Message Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| criterionSearch   * txtMessContain * cbInOut * cbTime * btSearch | Users types message’s contain to search  User selects message’s type(Inbox/Outbox) to search  User selects time(All/To Day/This Month/This Year) to search  User clicks the button to search |
| grMessList | System displays a list of user’s messages depends on search condition. |

## Report Managing

### Create Report

#### Screen Image

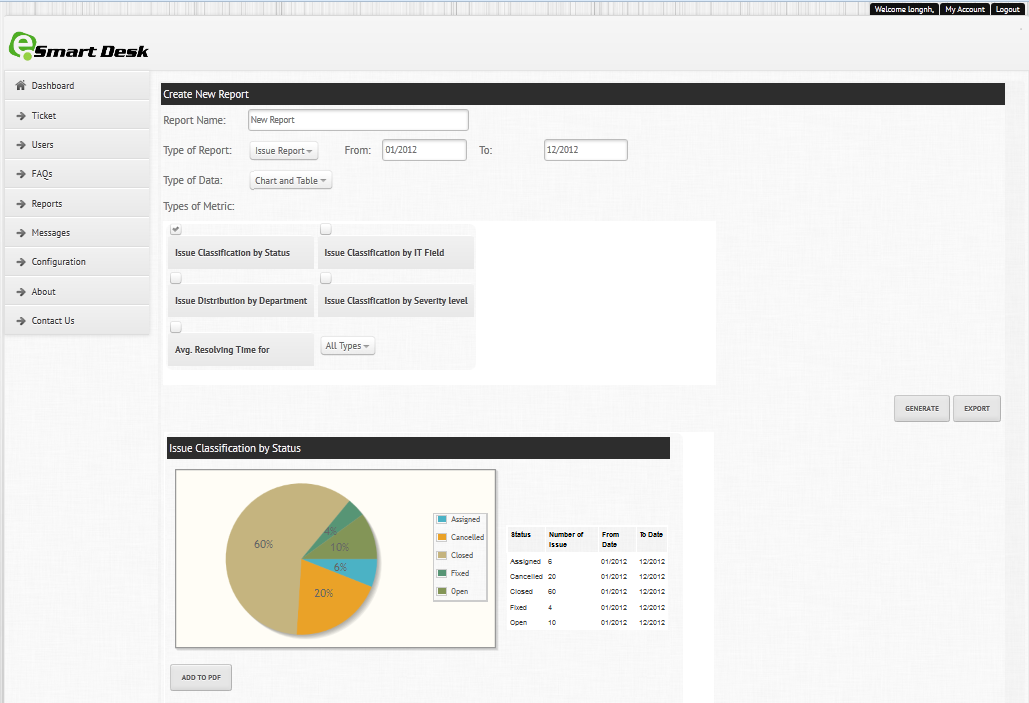


Figure 53 – Create Report Screen

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| cbRpType | User selects report type |
| txtReportName | User types report’s name |
| txtFrom | User types report’s Time range from |
| txtTo | User types report’s time range to |
| cbRpDataType | User selects report’s data type |
| ckTypeOfMetric | User checks to selects report’s type of metric |
| btnGenerate | User clicks to generate report |
| txtExport | User clicks to export report to pdf |
| btnAddToPdf | User click to add a chart to report |

## Configuration

### Add Department/ IT Type

#### Screen Image

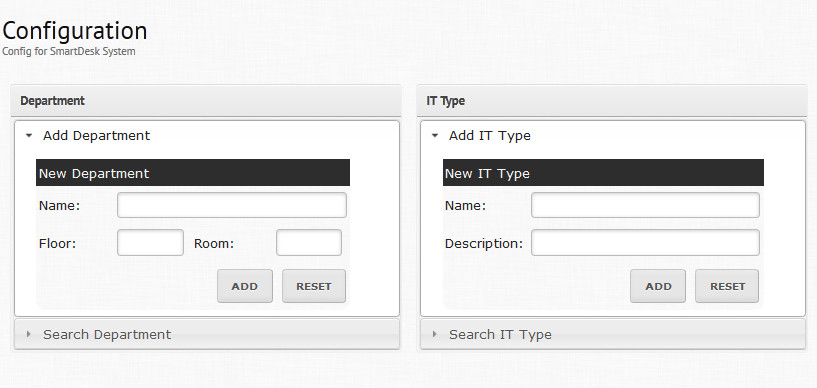


Figure 54 – Configuration Screen (Create)

#### Objects and Actions

|  |  |
| --- | --- |
| Object Name | Action |
| Department   * txtDepName * txtDepFloor * txtDepRoom * btnDepAdd * btnDepReset | * User types department’s name * User types department’s floor * User types department’s room * User clicks to add department * User clicks to clear all fields |
| IT Type   * txtTypeName * txtTypeDescript * btnTypeAdd * btnTypeReset | * User types IT Type’s name * User types IT Type’s Description * Users clicks to add IT Type * Users clicks to reset all fields |

### Search Department/ IT Type

#### Screen Image

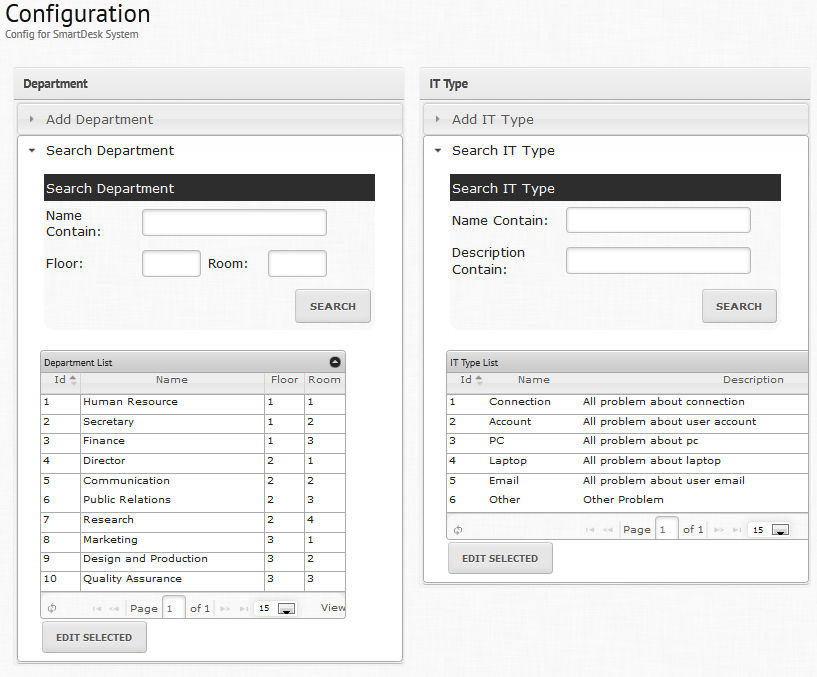


Figure 55 - Configuration Screen(Search)

#### Objects and Actions

|  |  |  |
| --- | --- | --- |
| Object Name | Action | |
| Department   * txtNameContains * txtFloor * txtRoom * btnSearch * gdResult * btnEditSelected | | * User types department’s name’s keyword to search * User types department’s floor’s keyword to search * User types department’s room’s keyword to search * User clicks button to search * System displays search’s result * User clicks to edit selected department |
| IT Type   * txtNameContains * txtDescript * btnSearch * gdResult * btnEditSelected | * User types IT type’s name’s keyword to search * User types IT type’s description’s keyword to search * User clicks button to search * System displays search’s result * User clicks to edit selected IT Type | |

### Edit Department/ IT Type

#### Screen Image

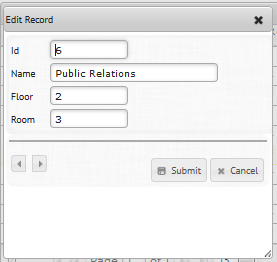


Figure 56 - Configuration Screen(Edit Department)

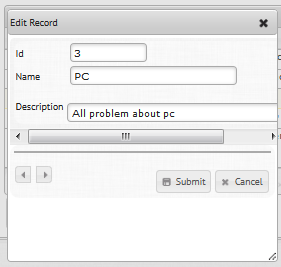


Figure 57 - Configuration Screen(Edit IT Type)

#### Objects and Actions

|  |  |  |
| --- | --- | --- |
| Object Name | Action | |
| Department   * txtID * txtName * txtFloor * txtRoom * btnSubmit * btnCancel | | * System displays department’s ID for user to view * User types department’s name to edit * User types department’s floor name to edit * User types department’s room to edit * User clicks button to submit * User clicks button to cancel |
| IT Type   * txtID * txtName * txtDescript * btnSubmit * btnCancel | * System displays department’s ID for user to view * User types IT type’s name to edit * User types IT type’s description to edit * User clicks button to submit * User clicks button to cancel | |

# Database Design



Figure 58 – Database Design

Software Test Documentation

# Introduction

## System Overview

This document provides all the features about test plans, test cases, test results, test environments, pass/fail criteria and risks estimation as well as checklist to cover all cases that can happen in Smart Desk system.

## Test Approach

* Test goal: To find all remain bugs in Smart Desk system.
* Test type: Black box testing.
* Test size: System Component.
* Test technique: Check list.

# Test Plan

Test plan document is used to verify and ensure that SmartDesk system meets its design specifications and other user requirements. The following part will describe which features will be tested and which is not, test tools and test environment.



## Features to be test

All SmartDesk system’s features will be test following screens which can include one or more functions

* GUI test
* Login screen
* Dashboard screen
* Tickets:
* My ticket screen.
* Log ticket screen (Basic mode)
* Log ticket screen (Advance mode)
* Search ticket screen
* View detailed ticket screen
* User:
* Create user screen
* Search user screen
* View detailed user screen
* FAQs:
* Log FAQ screen
* Search FAQ screen
* View detailed FAQ screen
* Report screen
* Messages:
* My messages
* Create message
* Main screen
* Upload function
* Auto fill function

## Features not to be test

N/A

## Test tool and environment

### Test tool

Log4j tool, supporting from DBMS (SQL Serve) and IDE (NetBeans)

### Test environment

Windows 7; Inter(R) Core(TM)2 Duo CPU 2.00GHz or above, RAM at least 1GB, HDD free at least 1GB; Firefox 4+, Internet Explorer 8+.

# Test Cases

## GUI Test

Purpose: Test whether GUI of smart desk system is match with the screen design in Software Detail Design document

Inputs: N/A

Results and pass/fail criteria:

* Pass: All screen’s elements, effects match with the design
* Fail: One or many screens did not match with the design

Test Procedure:

* View all of smart desk system screens, and check follow SDD document

## Login screen

Purpose: Test whether user can login into Smart Desk system with their correctly username and password according to Database

Inputs:

* Username
* Password

Results and pass/fail criteria:

* Pass: Navigate to Home screen (Dashboard screen)
* Fail:
* Notification “Please input Username!!” appears if username is not filled.
* Notification “Please input Password!!” appears if password is not filled.
* Notification “Incorrect Username or Password!!” if username or password not match with data in database

Test Procedure:

* Input username
* Input password
* Press “Login”

## Dashboard screen

Purpose: Test whether user can use each of every widget on dashboard screen

Inputs: N/A

Results and pass/fail criteria:

* Pass:
* Chart tabs appear correctly with title and data from database
* Statistic numbers appear correctly with data from database. Hyperlinks from each number can link to corresponding page
* “New Tickets”, “New FAQs”, “New Messages” display data table correctly with data from database
* Fail:
* Chart did not appear, or not match with data from database
* Statistic numbers did not appear, or not match with data from database
* “New Tickets”, “New FAQs”, “New Messages” did not appear correctly with data from database. Hyperlinks fail to direct to associated pages

Test Procedure:

* Click on each tab on “Issue Charts” widget.
* Click on each tab on “Number” widget
* Click on each tab on “New Joys” widget

## Tickets

### My ticket screen

Purpose: Test whether user can view list of his/her tickets and can use sort ticket function.

Inputs: N/A

Results and pass/fail criteria:

* Pass:
* “Issue list” table appears with all user’s issues
* Sort issue successful when click on each header of “Issue list” table
* Navigate to view detailed ticket screen if click on issue name
* Fail:
* Can’t sort issue when click on each header of “Issue list” table

Test Procedure:

* Click “Ticket” and then click “My ticket” from main menu
* Click on each header of “Issue list” table
* Click on issue name to navigate to View Detailed Issue screen



### Log ticket screen

#### Basic mode

Purpose: Test whether employee can log and edit new basic ticket (tickets just contain simple data like name, content, owner)

Inputs:

* Issue’s Name
* Issue’s Owner
* Attachments
* Issue’s Description

Results and pass/fail criteria:

* Pass:
* Notification “You have added new ticket successful!!” appears if press “Log”.
* Navigate to new log ticket screen if press “Log another”
* Notification “You have edited current ticket successfully!!” if press “Edit”
* Fail:
* Notification “Please input Ticket Owner!!” appears if issue’ owner is not filled
* Notification “Please input Ticket Name!!” appears if issue’ name is not filled
* Notification “Please input Ticket Description!!” appears if issue’s description is not filled.
* Notification “This Owner account is not exit!!” appears if issue’s owner is not match with any available employee account’s username.
* Notification “Ticket Name is too short!!” appears if issue’s name is shorter than 10 characters.
* Notification “Ticket Name limit is 50 characters!!” appears if issue’s name lengthen than 50 characters
* Notification “Ticket Description is too short!!” appears if issue’s description is shorter than 10 characters
* Notification “Ticket Description limit is 500 character!!” appears if issue’s description lengthen than 500 characters

Test Procedure:

* Click “Ticket” and then click “Basic mode” from main menu
* Input issue’s name
* Input issue’s owner
* Click “SELECT FILES” and choose some picture.
* Input issue’s description
* Press “Log”
* Press “Edit”
* Press “Log another”



#### Advanced mode

Purpose: Test whether IT staff can log a new ticket

Inputs:

* Name
* Owner
* Department
* Status
* Severity
* Type
* Assigned to
* Deadline
* Attachments
* Description

Results and pass/fail criteria:

* Pass:
* Deadline and assigned to only appears when status is “Assigned” or “Fixed”
* End Date only appears when status is “Closed” or “Cancelled”
* Username of user who logs issue and posted day appears correctly
* Navigate to view detailed ticket screen when press “Log”
* Fail:
* Notification “Please input Ticket Owner!!” appears if issue’ owner is not filled
* Notification “Please input Ticket Name!!” appears if issue’ name is not filled
* Notification “Please input Ticket Description!!” appears if issue’s description is not filled.
* Notification “Please input Ticket Deadline!!” if issue’s deadline is not filled
* Notification “Please input Ticket Respondent!!” appears if issue’s respondent is not filled
* Notification “Staff (respondent) account is not exist!!” appears if issue’s respondent is not match with any available staff account’s username
* Notification “Employee (owner) account is not exit!!” appears if issue’s owner is not match with any available employee account’s username
* Notification “Issue Name is too short!!” appears if issue’s name is shorter than 10 characters.
* Notification “Issue Name limit is 50 characters!!” appears if issue’s name lengthen than 50 characters
* Notification “Ticket Description is too short!!” appears if issue’s description is shorter than 10 characters
* Notification “Ticket Description limit is 5000 character!!” appears if issue’s description lengthen than 5000 characters
* Notification “Please choose department!!” appears if chosen department is “All Depts”

Test Procedure:

* Click “Ticket” and then click “Advanced Mode” from main menu
* Input name
* Input owner
* Choose department
* Choose status
* Choose severity
* Choose type
* Input assigned to
* Choose deadline
* Click “SELECT FILES” and choose some picture.
* Input description
* Press “Log”

### Search ticket screen

Purpose: Test whether user can view and search list of issue

Inputs:

* Name contain
* Description contain
* Owner
* Register
* Respondent
* Department
* Status
* Severity
* Type
* Month
* Day

Results and pass/fail criteria:

* Pass:
* Displayed result is match correctly with database
* Sort issue successful when click on each header of “Issue list” table
* Navigate to view detailed ticket screen if click on issue name
* Fail:
* Displayed result is not match correctly with database
* Can’t sort issue when click on each header of “Issue list” table

Test Procedure:

* Click “Ticket” and then click “Advanced Search” from main menu
* Input name contain
* Input description contain
* Input owner
* Input register
* Input respondent
* Choose status
* Choose department
* Choose type
* Choose severity
* Choose month
* Choose day
* Press “Search”
* Click on issue name

### View detailed ticket screen

Purpose: Test whether user can edit one issue

Inputs:

* Name
* Owner
* Department
* Status
* Severity
* Type
* Assigned to
* Deadline
* End Date
* Attachments
* Description

Results and pass/fail criteria:

* Pass:
* Deadline only appears when status is “Assigned” or “Fixed”
* End Date only appears when status is “Closed” or “Cancelled”
* Notification “You have edited ticket successfully!!” if press “Edit”
* Fail:
* Notification “Please input Ticket Owner!!” appears if issue’ owner is not filled
* Notification “Please input Ticket Name!!” appears if issue’ name is not filled
* Notification “Please input Ticket Description!!” appears if issue’s description is not filled.
* Notification “Please input Ticket Deadline!!” if issue’s deadline is not filled
* Notification “Please input Ticket Respondent!!” appears if issue’s respondent is not filled
* Notification “Staff (respondent) account is not exist!!” appears if issue’s respondent is not match with any available staff account’s username
* Notification “Employee (owner) account is not exit!!” appears if issue’s owner is not match with any available employee account’s username
* Notification “Issue Name is too short!!” appears if issue’s name is shorter than 10 characters.
* Notification “Issue Name limit is 50 characters!!” appears if issue’s name lengthen than 50 characters
* Notification “Ticket Description is too short!!” appears if issue’s description is shorter than 10 characters
* Notification “Ticket Description limit is 5000 character!!” appears if issue’s description lengthen than 5000 characters

Test Procedure:

* Input name
* Input owner
* Choose department
* Choose status
* Choose severity
* Choose type
* Input assigned to
* Choose or input deadline
* Choose or input end day
* Click “SELECT FILES” and choose some picture
* Input description
* Press “Edit”

## User

### Create user screen

Purpose: Test whether admin can create new user

Inputs:

* Username
* First Name
* Last Name
* Birthday
* Gender
* Phone
* Address
* Mail
* Role
* Active
* IT Field or Department

Results and pass/fail criteria:

* Pass:
* IT field only appears when type is “IT Staff”
* Department only appears when type is “Employee”
* Navigate to view detailed user screen if press “Save”
* Fail:
* Notification “Please input username” appears if username is not filled
* Notification “Invalid Username” appears if username is shorter than 3 characters and lengthen than 15 characters.
* Notification “This username is already in use” appears if username is already registered
* Notification “Please input Email” appears if email is not filled
* Notification “Invalid Email Address” appears if email is not correct
* Notification “Invalid Phone number” appears if phone number is not number and lengthen than 11 characters
* Notification “Please choose a role” appears if chosen role is “All Roles”
* Notification “Please choose a type” appears if chosen IT field is “All Types”
* Notification “Please choose a department” appears if chosen department is “All Departments”

Test Procedure:

* Click “Users” then click “Create User” from main menu
* Input username
* Input first name
* Input last name
* Choose or input birthday
* Choose gender
* Input phone
* Input address
* Input mail
* Choose role
* Check active
* Choose IT filed or department
* Press “Save”



### Search user screen

Purpose: Test whether user can view and search list of user’s information

Inputs:

* Username
* First Name
* Last Name
* Phone
* Email
* Roles

Results and pass/fail criteria:

* Pass:
* Displayed result is matched correctly with database
* Sort user successful when click on each header of “User list” table
* Navigate to view detailed user screen if click on user name
* Fail:
* Displayed result is not matched correctly with database
* Can’t sort user when click on each header of “Issue list” table

Test Procedure:

* Click “Users” and then click “Advanced Search” from main menu
* Input username
* Input first name
* Input last name
* Input phone
* Input email
* Choose roles
* Press “Search”
* Click on username

### View detailed user screen

Purpose: Test whether user can view and edit his/her information

Inputs:

* Username
* Password
* Re-enter password
* First Name
* Last Name
* Birthday
* Gender
* Phone
* Address
* Mail
* Role (Admin only)
* Active (Admin only)
* IT Field or Department

Results and pass/fail criteria:

* Pass:
* IT field only appears when type is “IT Staff”
* Department only appears when type is “Employee”
* Only admin can change user’s role
* Only admin can change user’s status (active or not)
* Button “Edit” is not appear when user view another user’s information or they were not admin
* Notification “You have edited account successfully!!” appears if press “Edit”
* Fail:
* Notification “Please input username” appears if username is not filled
* Notification “Invalid Username” appears if username is shorter than 3 characters and lengthen than 15 characters.
* Notification “This username is already in use” appears if username is already registered
* Notification “Password must be at least 5 characters” appears if password is shorter than 5 characters
* Notification “Please input Email” appears if email is not filled
* Notification “Invalid Email Address” appears if email is not correct
* Notification “Invalid Phone number” appears if phone number is not number and lengthen than 11 characters
* Notification “Please choose a role” appears if chosen role is “All Roles”
* Notification “Please choose a type” appears if chosen IT field is “All Types”
* Notification “Please choose a department” appears if chosen department is “All Departments”

Test Procedure:

* Input username
* Input password
* Input re-enter password
* Input first name
* Input last name
* Choose or input birthday
* Choose gender
* Input phone
* Input address
* Input mail
* Choose role (admin only)
* Check active (admin only)
* Choose IT filed or department
* Press “Edit”

## FAQs

### Log FAQ screen

Purpose: Test whether IT staff can create new FAQ

Inputs:

* FAQs Question
* FAQs Type
* Attachment
* FAQs Content

Results and pass/fail criteria:

* Pass: Navigate to view detailed FAQ screen if press “Log”
* Fail:
* Notification “Please input FAQs question” appears if FAQs question is not filled
* Notification “Please input FAQs content” appears if FAQs content is not filled
* Notification “FAQs Question is too short!!” FAQs question is shorter than 10 characters
* Notification “FAQs Question limit is 100 character!!” appears FAQs question lengthen than 100 characters
* Notification “FAQs Content is too short!!” FAQs content is shorter than 50 characters

Test Procedure:

* Click “FAQs” and then click “Log FAQ” from main menu
* Input FAQs question
* Choose FAQs type
* Click “SELECT FILES” and choose some picture
* Input FAQs content
* Press “Log”

### Search FAQ screen

Purpose: Test whether user can view and search list of FAQ

Inputs:

* Question contain
* Content contain
* Type
* Creator
* Status (disable if user was employee)
* Month
* Day

Results and pass/fail criteria:

* Pass:
* Choose status is not appear if user was employee
* Result list only show approved FAQs if user was employee
* Displayed result is matched correctly with database
* Sort FAQ successful when click on each header of “FAQ list” table
* Navigate to view detailed FAQ screen if click on FAQs question
* Fail:
* Displayed result is not matched correctly with database
* Can’t sort FAQ when click on each header of “FAQ list” table

Test Procedure:

* Click “FAQs” and then click “Search FAQ” from main menu
* Input question contain
* Input content contain
* Choose type
* Choose status (disable if user was employee)
* Choose month
* Choose day
* Press “Search”
* Click on FAQs question

### View detailed FAQ screen

Purpose: Test whether user can view and edit one FAQ

Inputs:

* Question
* Type
* Attachment
* Content

Results and pass/fail criteria:

* Pass:
* Button “Edit” is not appear if user was employee
* Choose status is not appear if user was not Admin
* Username of user who create FAQ and created day is displayed correctly
* Username of user who edit FAQ and last updated day is displayed correctly
* Notification “You have edited FAQ successfully” if press “Edit”
* FAQ History is pop-up and displayed correctly if press “History”
* Fail:
* Notification “Please input FAQs question” appears if FAQs question is not filled
* Notification “Please input FAQs content” appears if FAQs content is not filled
* Notification “FAQs Question is too short!!” FAQs question is shorter than 10 characters
* Notification “FAQs Question limit is 100 character!!” appears FAQs question lengthen than 100 characters
* Notification “FAQs Content is too short!!” FAQs content is shorter than 50 characters

Test Procedure:

* Input question
* Choose type
* Choose status
* Click “SELECT FILES” and choose some picture
* Input content
* Press “Edit” (disable if user was employee)
* Press “History”

## Report screen

Purpose: Test whether admin can create new report

Inputs:

* Report name
* From (start date)
* To (end date)
* Type of Report
* Type of Data
* Type of Metric
* For (if type of report is “Staff Report”)
* IT Team (if type of report is “Staff Report”)
* IT staff username (if type of report is “Staff Report”)
* Type of issue (if type of report is “Issue Report”)

Results and pass/fail criteria:

* Pass:
* For only appears if type of report is “Staff Report”
* IT Team only appears if type of report is “Staff Report” and “For” is “One IT Team”
* IT staff username only appears if type of report is “Staff Report” and “For” is “One IT Staff”
* Type of issue only appears if type of report is “Issue Report”
* Can save report in PDF format if press “Export”
* Can generate report if press “Generate”
* Can add report to PDF file if press “Add report”
* Fail:
* Notification “Please choose From Date!!” appears if “From” is not filled
* Notification “Please choose To Date!!” appears if “To” is not filled
* Notification “Please input staff name!!” appears if “IT staff username” is not filled

Test Procedure:

* Input report name
* Choose start date
* Choose end date
* Choose type of report
* Choose type of data
* Choose type of metric
* For (if type of report is “Staff Report”)
* IT Team (if type of report is “Staff Report”)
* Input IT staff username (if type of report is “Staff Report”)
* Choose type of issue (if type of report is “Issue Report”)
* Press “Generate”
* Press “Add to PDF”
* Press “Export”

## Message

### My messages screen

Purpose: Test whether user can view and search his/her message

Inputs:

* Message contain
* Inbox/Outbox
* Time

Results and pass/fail criteria:

* Pass:
* Displayed result is match correctly with database
* Sort message successful when click on each header of “Message list” table
* Navigate to view detailed ticket screen if click on “Click here”
* Fail:
* Displayed result is not matched correctly with database
* Can’t sort message when click on each header of “Message list” table

Test Procedure:

* Input message contain
* Choose Inbox/Outbox
* Choose time
* Press “Search”

### Create message screen

Purpose: Test whether admin can create new message

Inputs:

* Receiver
* Content

Results and pass/fail criteria:

* Pass:
* Notification ”You have sent new message successful !!” if press “Send”
* Clear all input if press “New Message”
* Fail:
* Notification “Please input Receiver!!” appears if receiver is not filled
* Notification “This Receiver account is not exist!!” appears if receiver is not match with any available account’s username
* Notification “Please input Message content!!” if content is not filled
* Notification “Message Content is too short!!” if content is shorter than 10 characters
* Notification “Message Content limit is 500 character!!” if content is lengthen than 500 characters

Test Procedure:

* Input receiver
* Input content
* Press “Send”
* Press “New Message”

## Main screen

Purpose: Test whether main menu or sub-menu appears suitably with each role and user can navigate to view detailed user page or log out.

Inputs: N/A

Results and pass/fail criteria:

* Pass:
* Navigate to view detailed user screen if click “My account”
* Navigate to Login screen if click “Log out”
* Main menu and sub-menu appear suitably with each role
* Fail: N/A

Test Procedure:

* Login with each roles
* Check main menu and sub-menu
* Click “My account”
* Click “Log out”

## Upload function

Purpose: Test whether user can upload picture file from his/her computer

Page: Log ticket (Advanced Mode) screen, Edit ticket screen, Log FAQ screen, Edit FAQ screen

Results and pass/fail criteria:

* Pass:
* Notification “’total number of uploaded file’ file were successfully uploaded” appears if user upload picture
* Notification “Are you sure you wish to delete this file?” appear when user click on icon “X”
* Can view picture if click “View”
* Upload files were appears correctly in next screen.
* Fail:
* Notification “Some file were not added to the queue: The file ‘uploaded file’ exceeds the size limit (500KB)” appears when user upload picture with size larger than 500 kilobyte
* Notification “The file named ‘name of the repeated file’ is already in the queue. Do you want to replace the existing item in the queue?” appears if user upload the same picture twice

Test Procedure:

* Go to log FAQ screen
* Click “SELECT FILES” and choose some picture
* Click on icon “X”
* Press “Log”
* Check all upload file from next screen
* Click “View”

## Auto complete function

Purpose: Test whether user need to type a username, there are a suggested list appears and user can choose from this.

Page: Log ticket screen, Search ticket screen, View detailed ticket screen, Search FAQ screen, Search user screen, Create report screen, Create message screen.

Results and pass/fail criteria:

* Pass: Suggested list appears correctly and user can choose from this
* Fail: Suggested list do not appear correctly or user can choose from this

Test Procedure:

* Go to each screen that have auto complete function
* Input on every textbox that have auto complete function

# Checklists

Abbreviation:

* State: [A]bnormal – [N]ormal – [B]oundary
* Result: [P]ass = [F]ail

## GUI Test

### Main screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_001 |  | Look at the left of screen | From top to bottom, menus is displayed:  + Dashboard  + Ticket  + Users  + FAQs  + Reports  + Messages  + About  + Contact us |  | P |  |  |
| TC\_GUI\_002 |  | Look at the left of screen, click on menus | When click on Ticket, sub-menu is displayed:  + My Ticket  + Log Ticket  + Advanced Mode or Basic Mode  + Search Ticket  When click on Users, sub-menu is displayed:  + Create User  + Search User  When click on FAQs, sub-menu is displayed:  + Create FAQ  + Search FAQ  When click on Reports, sub-menu is displayed:  + Create Reports  When click on Reports, sub-menu is displayed:  + My Messages  + Create Message |  | P |  |  |
| TC\_GUI\_003 |  | Look at the right of screen | Following widgets is displays:  + Calendar  + Number  + Issue Charts: 4 tabs  -Issue by status  -Issue by department  -Issue by severity  -Issue by IT field  +New Joys |  | P |  |  |



### My ticket screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Ticket -> My Ticket | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_004 |  | Look at the right of screen | Issue list table is displayed:  + Id  + Name  + Type  + Severity  + Owner  + Register  + Respondent  + Department  + Status |  | P |  |  |

### Log ticket basic mode screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Ticket -> Log Ticket -> Basic Mode | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_005 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Issue’s Name”  + [Issue’s Name]Textbox  + Label: “Issue’s Owner”  + [Issue’s Owner]Textbox  + Label: “Department”  + Label: “Attachments”  + [Select Files]Button  + Label: “Issue Description”  + Toolbar menu  + [Issue Description] Text area  + [Log] button  +[Reset] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### Log ticket advanced mode screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Ticket -> Log Ticket -> Advanced Mode | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_006 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Name”  + [Name]Textbox  + Label: “Owner”  + [Owner]Textbox  + Label: “Department”  + [Department]Combo box: 1.All Depts-2.Human Resource-3.Secretary-4.Finance-5.Director-6.Planning  + Label: “Status”  + [Status]Combo box: All status-2.Open-3.Assigned-4.Fixed-5.Closed-6.Cancelled  + Label: “Severity”  + [Severity]Combo box: 1.All Severity-2.Easy-3.Normal-4.Hard  + Label: “Type”  + [Type]Combo box: 1.All Types – 2.Connection -3.Account-4.Desktop-5.Laptop-6.Email-7.Other  + Label: “Posted Date”  + [Posted Date]Textbox  + Label: “By”  + [By]Textbox  + Label: “Attachments”  + [Select Files]Button  + Label: “Description”  + Toolbar menu  + [Description] Text area  + [Log] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### View detail ticket screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Ticket -> Search Ticket -> Advanced Mode -> Click on one row of result table | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_007 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Name”  + [Name]Textbox  + Label: “Owner”  + [Owner]Textbox  + Label: “Department”  + [Department]Combo box: 1.All Depts-2.Human Resource-3.Secretary-4.Finance-5.Director-6.Planning  + Label: “Status”  + [Status]Combo box: All status-2.Open-3.Assigned-4.Fixed-5.Closed-6.Cancelled  + Label: “Severity”  + [Severity]Combo box: 1.All Severity-2.Easy-3.Normal-4.Hard  + Label: “Type”  + [Type]Combo box: 1.All Types – 2.Connection -3.Account-4.Desktop-5.Laptop-6.Email-7.Other  + Label: “Posted Date”  + [Posted Date]Textbox  + Label: “By”  + [By]Textbox  + Label: “Assigned to”  + [Assigned to”]Textbox  + Label: “Deadline”  + [Deadline”]Textbox  + Label: “Attachments”  + [Select Files]Button  + Label: “Description”  + Toolbar menu  + [Description] Text area  + [History] button  + [Edit] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### Search ticket screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Ticket -> Search Ticket -> Advanced Mode | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_008 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Name contain”  + [Name contain]Textbox  + Label: “Owner”  + [Owner]Textbox  + Label: “Register”  + [Register]Textbox  + Label: “Description”  + [Description] Text box  + Label: “Respondent”  + [Respondent]Textbox  + Label: “Status”  + [Status]Combo box:1.All status-2.Open-3.Assigned-4.Fixed-5.Closed-6.Cancelled  + Label: “Department”  + [Department]Combo box:1.All Depts-2.Human Resource-3.Secretary-4.Finance-5.Director-6.Planning  + Label: “Type”  + [Type]Combo box: 1.All Types – 2.Connection -3.Account-4.Desktop-5.Laptop-6.Email-7.Other  + Label: “Severity”  + [Severity]Combo box:1.All Severity-2.Easy-3.Normal-4.Hard  + [Search] button  + Issue List table:  - Id  - Name  - Type  - Severity  - Owner  - Register  - Respondent  - Department  - Status |  | P |  |  |

### Create user screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Users -> Create User | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_009 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Username”  + [Username]Textbox  + Label: “Role”  + [Role] combo box: 1.All Roles-2.Admin-3.IT Staff-4.Employee  + Label: “Password”  + [Password]Textbox  + Label: “First Name”  + [First Name]Textbox  + Label: “Last Name”  + [Last Name]Textbox  + Label: “Gender”  + [Gender]Radio button : Male/Female  + Label: “Phone”  + [Phone]Textbox  + Label: “Address”  + [Address]Textbox  + Label: “Mail”  + [Mail]Textbox  + Label: “Active”  + [Active]Checkbox  +[Save] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### View detail user screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Users -> Search User -> Click one row of result table | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_010 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Username”  + [Username]Textbox  + Label: “Role”  + [Role] combo box: 1.All Roles-2.Admin-3.IT Staff-4.Employee  + Label: “Password”  + [Password]Textbox  + Label: “First Name”  + [First Name]Textbox  + Label: “Last Name”  + [Last Name]Textbox  + Label: “Gender”  + [Gender]Radio button : Male/Female  + Label: “Phone”  + [Phone]Textbox  + Label: “Address”  + [Address]Textbox  + Label: “Mail”  + [Mail]Textbox  + Label: “Active”  + [Active]Checkbox  +[Edit] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### Search user screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Users -> Search User -> Advanced Mode | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_011 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Username”  + [Username]Textbox  + Label: “First Name”  + [First Name]Textbox  + Label: “Last Name”  + [Last Name]Textbox  + Label: “Phone”  + [Phone]Textbox  + Label: “Mail”  + [Mail]Textbox  + Label: “Role”  + [Role] combo box:1.All Roles-2.IT Staff-3.Employee  +[Search] button  + User List table:  - Id  - Username  - First Name  - Last Name  - Email  - Role |  | P |  |  |

### Log FAQ screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click FAQs -> Log FAQ | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_012 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “FAQs Question”  + [FAQs Question]Textbox  + Label: “FAQs Type”  + [FAQs Type] combo box  + Label: “Attachments”  + [Attachments]Button  + Label: “FAQs Content”  +Toolbar menu  + [FAQs Content]Text area  +[Log] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### Search FAQ screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click FAQs -> Search FAQ | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_013 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Question contain”  + [Question contain]Textbox  + Label: “Type”  + [Type] combo box: 1.All Types – 2.Connection -3.Account-4.Desktop-5.Laptop-6.Email-7.Other  + Label: “Creator”  + [Creator] Textbox  + Label: “Content contain”  + [Content contain]Textbox  + Label: “Status”  + [Status] combo box:1.All-2.Approved-3.Unapproved  +[Search] button  + FAQ list table  -Id  -Question  -Type  -Creator  -IsApproved |  | P |  |  |

### View detail FAQ screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click FAQs -> Search FAQ -> Click one row of result table | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_017 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Question”  + [Question]Textbox  + Label: “Created Day”  + [Created Day]Textbox  + Label: “By”  + [By]Textbox  + Label: “Type”  + [Type] combo box  + Label: “Status”  + [Status] combo box  + Label: “Last Updated Day”  + [Last Updated Day]Textbox  + Label: “By”  + [By]Textbox  + Label: “Attachments”  + [Attachments]Button  + Label: “Content”  +Toolbar menu  + [Content]Text area  +[History] button  +[Edit] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### Create report screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Reports -> Create Report | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_014 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Report Name”  + [Report Name]Textbox  + Label: “Type of report”  + [Type of report] combo box:1.Issue report -2.Staff report  + Label: “From”  + [From] Textbox  + Label: “To”  + [To] Textbox  + Label: “Type of data”  + [Type of data] combo box:1.Chart and table -2.Only chart  +Label: “Types of Metric”  +Label:” Issue Classification by status”  +[ Issue Classification by status] checkbox  +Label:”Issue Classification by IT Field”  +[ Issue Classification by IT Field] checkbox  +Label: “Issue Classification by department”  +[ Issue Classification by department] checkbox  +Label: “Issue Classification by severity level”  +[ Issue Classification by severity level] checkbox  +Label: “Avg. Resolving Time for”  +[ Avg. Resolving Time for] checkbox  + [Avg. Resolving Time] Combo box:1.All Types – 2.Connection -3.Account-4.Desktop-5.Laptop-6.Email-7.Other  +[Generate] button  +[Export] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |

### My message screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Messages -> My messages | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_015 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Message contain”  + [Message contain]Textbox  + Label: “Inbox/Outbox”  + [Inbox/Outbox] combo box: 1.Inbox – 2.Outbox  + Label: “Time”  + [Time] combo box: 1.All – 2.Today -3.This month -4. This Year  +[Search] button  + Message list table  -Id  -From  -To  -Message  -Time |  | P |  |  |

### Create message screen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Description | Test Case Procedure | Expected Output | Dependence | Result | Test date | Note |
|  | Steps   1. Go to home page 2. Login 3. Click Messages ->Create messages | | | | | | |
|  | Components | | | | | | |
| TC\_GUI\_016 |  | Look at the right of screen | From top to bottom, from left to right, component is displayed:  + Label: “Receiver”  + [Receiver]Textbox  + Label: “Content”  + [Content] Text area  +[Send] button  +[New message] button  +Label: “System message”  +[ System message] Text area |  | P |  |  |



## Login screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| LS001 | Username | “huyld1” | System Message: You have failed 1 time(s).You have only 4 time(s) to login. | A | P |
| Password | “huyld” |
| LS002 | Username | “huyld” | Navigate to Home screen | N | P |
| Password | “huyld” |
| LS003 |  | Failed to login in 5 time | System Message:  Please wait for 5 minute(s)!! | A | P |



## Dashboard screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| DS001 | Click on “Status” tab from “Issue Charts” widget | Chart appear correctly with data from database | N | P |
| DS002 | Click on “Department” tab from “Issue Charts” widget | Chart appear correctly with data from database | N | P |
| DS003 | Click on “Severity” tab from “Issue Charts” widget | Chart appear correctly with data from database | N | P |
| DS004 | Click on “Type” tab from “Issue Charts” widget | Chart appear correctly with data from database | N | P |
| DS005 | Click on “50 Total Tickets” from “Number” widget | Navigate to search ticket screen | N | P |
| DS006 | Click on “6 Total Staffs” from “Number” widget | Navigate to search user screen with condition is role is “Staff” | N | P |
| DS007 | Click on “27 Total FAQs” from “Number” widget | Navigate to search FAQ screen | N | P |
| DS008 | Click on “Un-Fixed Tickets” from “Number” widget | Navigate to search ticket screen with condition is issue’s status is “Open” | N | P |
| DS009 | * Login with staff account: huyld * Log out * Login with staff account: minhnq * Log new FAQ * Log new Ticket * Log out * Login with staff account: huyld | “New Joys” widget display:  1 New Tickers  1 New FAQs  1 New Messages | N | P |
| DS010 | * Login with staff account: huyld * Log out * Login with staff account: minhnq * Log new FAQ * Log new Ticket * Log out * Login with staff account: huyld * Click on “1 New Tickets” | Ticket List appear with all new tickets | N | P |
| DS011 | * Login with staff account: huyld * Log out * Login with staff account: minhnq * Log new FAQ * Log new Ticket * Log out * Login with staff account: huyld * Click on “1 New Tickets” * Click on one of the ticket name appeared on ”Ticket List” | Navigate to view detailed ticket screen | N | P |
| DS012 | * Login with staff account: huyld * Log out * Login with staff account: minhnq * Log new FAQ * Log new Ticket * Log out * Login with staff account: huyld * Click on “1 New FAQs” | FAQ List appear with all new tickets | N | P |
| DS013 | * Login with staff account: huyld * Log out * Login with staff account: minhnq * Log new FAQ * Log new Ticket * Log out * Login with staff account: huyld * Click on “1 New FAQs” * Click on one of the FAQs question appeared on ”FAQ List” | Navigate to view detailed FAQ screen |  | P |
| DS014 | * Login with staff account: huyld * Log out * Login with admin account: longnh * Create new message with receiver is huyld * Log out * Login with staff account: huyld | “New Joys” widget display:  0 New Messages | N | F |

## Tickets

### My ticket screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| MT001 | Click on “Id” header from “Issue List” table | “Issue List” was sorted by Id from largest to smallest | N | P |
| MT002 | Click on “Name” header from “Issue List” table | “Issue List” was sorted by name in ascending alphabetical order | N | P |
| MT003 | Click on “Type” header from “Issue List” table | “Issue List” was sorted by type in ascending alphabetical order | N | P |
| MT004 | Click on “Severity” header from “Issue List” table | “Issue List” was sorted by severity in ascending alphabetical order | N | P |
| MT005 | Click on “Owner” header from “Issue List” table | “Issue List” was sorted by owner in ascending alphabetical order | N | P |
| MT006 | Click on “Register” header from “Issue List” table | “Issue List” was sorted by register in ascending alphabetical order | N | P |
| MT007 | Click on “Respondent” header from “Issue List” table | “Issue List” was sorted by respondent in ascending alphabetical order | N | P |
| MT008 | Click on “Department” header from “Issue List” table | “Issue List” was sorted by department in ascending alphabetical order | N | P |
| MT009 | Click on “Status” header from “Issue List” table | “Issue List” was sorted by status in ascending alphabetical order | N | P |
| MT010 | Click on issue’s name “Can’t open desktop” from “Issue List” table | Navigate to view detailed issue screen | N | P |
| MT011 | Login with staff account:huyld  In main menu click “My Ticket” | “Issue list” table appears with all user’s issues (Respondent is huyld) | N | P |
| MT012 | Login with employee account: duongls  In main menu click “My Ticket” | “Issue list” table appears with all user’s issues (Owner is duongls) | N | P |

### Log ticket screen

#### Basic mode

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| ET001 | Issue’sName | “” | System Message:  Please input TicketOwner!!  Please input Ticket Name!!  Please input Ticket Description | A | P |
| Issue’sOwner | “” |
| Attachments | “” |
| IssueDescription | “” |
| ET002 | Issue’s Name | “123456789” | System Message:  Ticket Name is too short!!  Ticket Description is too short!!  This Owner account is not exit!! | B | P |
| Issue’s Owner | “huyld” |
| Attachments | “” |
| Issue Description | “123456789” |
| ET003 | Issue’s Name | 51 characters | System Message:  Ticket Name limit is 50 characters!!  Ticket Description limit is 5000 characters!! | B | P |
| Issue’s Owner | “haitct” |
| Attachments | “” |
| Issue Description | 5001 characters |
| ET004 | Issue’s Name | “Cannot login” | System Message:  You have added new ticket successful!! | N | P |
| Issue’s Owner | “haitct” |
| Attachments | Choose 3 picture file |
| Issue Description | “I can login with my account duongls” |
|  | Press “Log” |
| ET005 | Issue’s Name | “Cannot login” | System Message:  You have edited current ticket successfully!! | N | P |
| Issue’s Owner | “haitct” |
| Attachments | Choose 3 picture file |
| Issue Description | “I can login with my account duongls” |
|  | Press “Log” |
| Issue’s Owner | “duongls” |
|  | Press “Edit” |
| ET006 | Issue’s Name | “Cannot login” | Navigate to new log ticket screen (basic mode) | N | P |
| Issue’s Owner | “haitct” |
| Attachments | Choose 3 picture file |
| Issue Description | “I can login with my account duongls” |
|  | Press “Log” |
| Issue’s Owner | “duongls” |
|  | Press “Edit” |
|  | Press “Log another” |

#### Advanced mode

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| IT001 | Name | “” | System Message:  Please input Ticket Owner!!  Please input Ticket Name!!  Please input Ticket Description  Please choose department!! | A  m | P |
| Owner | “” |
| Department | All Depts |
| Status | All Status |
| Severity | All Severity |
| Type | All Types |
| Attachments | “” |
| Description | “” |
| IT002 | Name | “123456789” | System Message:  Issue Name is too short!!  Issue Description is too short!!  Please input Ticket Deadline!!  Please input Ticket Respondent!! | B | P |
| Owner | “haitct” |
| Department | Finance |
| Status | Assigned |
| Severity | High |
| Type | Account |
| Attachments | “” |
| Description | “123456789” |
| Assigned to | “” |
| Deadline | “” |
| IT003 | Name | 51 characters | System Message:  Staff (respondent) account is not exist!!  Employee (owner) account is not exit!!  Issue Name limit is 50 characters!!  Ticket Description limit is 5000 character!! | B | P |
| Owner | “huyld” |
| Department | Finance |
| Status | Assigned |
| Severity | High |
| Type | Account |
| Attachments | “” |
| Description | 5001 characters |
| Assigned to | “duongls” |
| Deadline | 18/07/212 |
| IT004 | Name | “Cannot login” | Navigate to view detailed ticket screen | N | P |
| Owner | “duongls” |
| Department | Finance |
| Status | Assigned |
| Severity | High |
| Type | Account |
| Attachments | Choose 3 picture file |
| Description | “I can login with my account duongls” |
| Assigned to | “huyld” |
| Deadline | 18/07/2012 |
| IT005 | Status | Assigned | “Deadline”, “Assigned to” appear | N | P |
| IT006 | Status | Fixed | “Deadline”, “Assigned to” appear | N | P |
| IT007 | Status | Closed | “End Date” appear | N | P |
| IT008 | Status | Cancelled | “End Date” appear | N | P |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| IT010 | Login withaccount huyld in the date 18/07/2012 | Posted Date: 18/07/2012  By : huyld | N | P |

### Search ticket screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| ST001 | Name Contain | “o” | Displayed result is match correctly with database | N  m | P |
| Owner | duongls |
| Register | mingnq |
| Description Contain | “o” |
| Respondent | huyld |
| Status | Open |
| Department | Finance |
| Type | Connection |
| Severity | Critical/Urgent |
| Month | 8/2012 |
| Day | 27 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| ST002 | Click on “Id” header from “Issue List” table | “Issue List” was sorted by Id from largest to smallest | N | P |
| ST003 | Click on “Name” header from “Issue List” table | “Issue List” was sorted by name in ascending alphabetical order | N | P |
| ST004 | Click on “Type” header from “Issue List” table | “Issue List” was sorted by type in ascending alphabetical order | N | P |
| ST005 | Click on “Severity” header from “Issue List” table | “Issue List” was sorted by severity in ascending alphabetical order | N | P |
| ST006 | Click on “Owner” header from “Issue List” table | “Issue List” was sorted by owner in ascending alphabetical order | N | P |
| ST007 | Click on “Register” header from “Issue List” table | “Issue List” was sorted by register in ascending alphabetical order | N | P |
| ST008 | Click on “Respondent” header from “Issue List” table | “Issue List” was sorted by respondent in ascending alphabetical order | N | P |
| ST009 | Click on “Department” header from “Issue List” table | “Issue List” was sorted by department in ascending alphabetical order | N | P |
| ST010 | Click on “Status” header from “Issue List” table | “Issue List” was sorted by status in ascending alphabetical order | N | P |
| ST011 | Click on issue’s name “Can’t open desktop” from “Issue List” table | Navigate to view detailed issue screen | N | P |

### View detailed ticket screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| VT001 | Name | “” | System Message:  Please input Ticket Owner!!  Please input Ticket Name!!  Please input Ticket Description | A  m | P |
| Owner | “” |
| Department | Finance |
| Status | Open |
| Severity | High |
| Type | Connection |
| Attachments | “” |
| Description | “” |
| VT002 | Name | “123456789” | System Message:  Issue Name is too short!!  Issue Description is too short!!  Please input Ticket Deadline!!  Please input Ticket Respondent!! | B | P |
| Owner | “haitct” |
| Department | Finance |
| Status | Assigned |
| Severity | High |
| Type | Account |
| Attachments | “” |
| Description | “123456789” |
| Assigned to | “” |
| Deadline | “” |
| VT003 | Name | 51 characters | System Message:  Staff (respondent) account is not exist!!  Employee (owner) account is not exit!!  Issue Name limit is 50 characters!!  Ticket Description limit is 5000 character!! | B | P |
| Owner | “huyld” |
| Department | Finance |
| Status | Assigned |
| Severity | High |
| Type | Account |
| Attachments | “” |
| Description | 5001 characters |
| Assigned to | “duongls” |
| Deadline | 18/07/212 |
| VT004 | Name | “Cannot login” | System Message:  You have edited ticket successfully!! | N | P |
| Owner | “duongls” |
| Department | Finance |
| Status | Assigned |
| Severity | High |
| Type | Account |
| Attachments | Choose 3 picture file |
| Description | “I can login with my account duongls” |
| Assigned to | “huyld” |
| Deadline | 18/07/2012 |
| VT005 | Status | Assigned | “Deadline”, “Assigned to” appear | N | P |
| VT006 | Status | Fixed | “Deadline”, “Assigned to” appear | N | P |
| VT007 | Status | Closed | “End Date” appear | N | P |
| VT008 | Status | Cancelled | “End Date” appear | N | P |

## User

### Create user screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| LU001 | Username | “” | System Message:  Please input username  Please input Email  Please choose a role | A  m | P |
| Role | All Roles |
| First Name | “” |
| Last Name | “” |
| Birthday | “” |
| Gender | Male |
| Phone | “” |
| Address | “” |
| Mail | “” |
| Active | checked |
| LU002 | Username | “hu” | System Message:  Invalid Username  Invalid Email Address  Invalid Phone number  Please choose a type | B | P |
| Role | IT Staff |
| IT Field | All Types |
| First Name | Huy |
| Last Name | Le |
| Birthday | 01/01/1990 |
| Gender | Male |
| Phone | “abc123” |
| Address | “1/1 Pasteur street” |
| Mail | “123” |
| Active | checked |
| LU003 | Username | 16 characters | System Message:  Invalid Username  Please choose a department  Invalid Phone number | B | P |
| Role | Employee |
| Department | All Departments |
| First Name | Huy |
| Last Name | Le |
| Birthday | 01/01/1990 |
| Gender | Male |
| Phone | “123456789012” |
| Address | “1/1 Pasteur street” |
| Mail | “123@yahoo.com” |
| Active | checked |
| LU004 | Username | “huyld” | System Message:  This username is already in use | A | P |
| Role | Employee |
| Department | Finance |
| First Name | Huy |
| Last Name | Le |
| Birthday | 01/01/1990 |
| Gender | Male |
| Phone | “0909090909” |
| Address | “1/1 Pasteur street” |
| Mail | “123@yahoo.com” |
| Active | checked |
| LU005 | Username | “diemntk” | Navigate to view detailed user screen | N | P |
| Role | Employee |
| Department | Finance |
| First Name | Diem |
| Last Name | Nguyen |
| Birthday | 01/01/1990 |
| Gender | Female |
| Phone | “0909090909” |
| Address | “1/1 Pasteur street” |
| Mail | “ntk@yahoo.com” |
| Active | checked |
| LU006 | Role | IT Staff | “IT field” appears | N | P |
| LU007 | Role | Employee | “Department” appears | N | P |

### Search user screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| SU001 | Username | “g” | Displayed result is matched correctly with database | N | P |
| First Name | “g” |
| Last Name | “g” |
| Phone | “9” |
| E-mail | “g” |
| Role | Admin |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| SU002 | Click on “Id” header from “User List” table | “User List” was sorted by Id from largest to smallest | N | P |
| SU003 | Click on “Username” header from “User List” table | “User List” was sorted by username in ascending alphabetical order | N | P |
| SU004 | Click on “First Name” header from “User List” table | “User List” was sorted by first name in ascending alphabetical order | N | P |
| SU005 | Click on “Last Name” header from “User List” table | “User List” was sorted by last name in ascending alphabetical order | N | P |
| SU006 | Click on “E-Mail” header from “User List” table | “User List” was sorted by email in ascending alphabetical order | N | P |
| SU007 | Click on “Role” header from “User List” table | “User List” was sorted by role in ascending alphabetical order | N | P |
| SU008 | Click on user name “longnh” from “User List” table | Navigate to view detailed user screen | N | P |

### View detailed user screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| VU001 | Username | “” | System Message:  Please input username  Please input Email  Please choose a role  Please input password | A  m | P |
| Password | “” |
| Re-Enter Password | “” |
| Role | All Roles |
| First Name | “” |
| Last Name | “” |
| Birthday | “” |
| Gender | Male |
| Phone | “” |
| Address | “” |
| Mail | “” |
| Active | checked |
| VU002 | Username | “hu” | System Message:  Invalid Username  Invalid Email Address  Invalid Phone number  Please choose a type  Password is too short  Please confirm | B | P |
| Password | “1234” |
| Re-Enter Password | “” |
| Role | IT Staff |
| IT Field | All Types |
| First Name | Huy |
| Last Name | Le |
| Birthday | 01/01/1990 |
| Gender | Male |
| Phone | “abc123” |
| Address | “1/1 Pasteur street” |
| Mail | “123” |
| Active | checked |
| VU003 | Username | 16 characters | System Message:  Invalid Username  Please choose a department  Invalid Phone number | B | P |
| Password | “12345” |
| Re-Enter Password | “12345” |
| Role | Employee |
| Department | All Departments |
| First Name | Huy |
| Last Name | Le |
| Birthday | 01/01/1990 |
| Gender | Male |
| Phone | “123456789012” |
| Address | “1/1 Pasteur street” |
| Mail | “123@yahoo.com” |
| Active | checked |
| VU004 | Username | “huyld” | System Message:  This username is already in use | A | P |
| Password | “12345” |
| Re-Enter Password | “12345” |
| Role | Employee |
| Department | Finance |
| First Name | Huy |
| Last Name | Le |
| Birthday | 01/01/1990 |
| Gender | Male |
| Phone | “0909090909” |
| Address | “1/1 Pasteur street” |
| Mail | “123@yahoo.com” |
| Active | checked |
| VU005 | Username | “diemntk” | You have edited account successfully!! | N | P |
| Password | “12345” |
| Re-Enter Password | “12345” |
| Role | Employee |
| Department | Finance |
| First Name | Diem |
| Last Name | Nguyen |
| Birthday | 01/01/1990 |
| Gender | Female |
| Phone | “0909090909” |
| Address | “1/1 Pasteur street” |
| Mail | “ntk@yahoo.com” |
| Active | checked |
| VU006 | Role | IT Staff | “IT field” appears | N | P |
| VU007 | Role | Employee | “Department” appears | N | P |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| VU008 | Login with employee account: haitct  Login with IT staff account: huyd | Role is read-only | N | P |
| VU009 | Login with employee account: haitct  Login with IT staff account: huyd | Status is read-only | N | P |
| VU010 | Login with IT staff account: huyd  View detailed user of account: haitct | Button “Edit” is not appear | N | P |

## FAQ

### Log FAQ screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| LF001 | FAQs Question | “” | System Message:  Please input FAQs Question!!  Please input FAQs Content!! | A | P |
| FAQs Type | Connection |
| Attachments | “” |
| FAQs Content | “” |
| LF002 | FAQs Question | “123456789” | System Message:  FAQs Question is too short!!  FAQs Content is too short!! | B | P |
| FAQs Type | Connection |
| Attachments | “” |
| FAQs Content | 49 characters |
| LF003 | FAQs Question | 101 characters | System Message:  FAQs Question limit is 100 characters!! | B | P |
| FAQs Type | Connection |
| Attachments | “” |
| FAQs Content | “Generate detailed reports about Help desk performance & metrics” |
| LF004 | FAQs Question | “What is Help desk Report?” | Navigate to edit FAQ screen | N | P |
| FAQs Type | Other |
| Attachments | Choose 3 picture file |
| FAQs Content | “Generate detailed reports about Help desk performance & metrics” |

### Search FAQ screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| SF001 | Question contain | “a” | Displayed result is matched correctly with database | N | P |
| Content contain | “a” |
| Type | Connection |
| Creator | “” |
| Status | All |
| Month | 8/2012 |
| Day | 27 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| SF002 | Click on “Id” header from “FAQ List” table | “FAQ List” was sorted by Id from largest to smallest | N | P |
| SF003 | Click on “Question” header from “FAQ List” table | “FAQ List” was sorted by question in ascending alphabetical order | N | P |
| SF004 | Click on “Type” header from “FAQ List” table | “FAQ List” was sorted by type in ascending alphabetical order | N | P |
| SF005 | Click on “Creator” header from “FAQ List” table | “FAQ List” was sorted by creator in ascending alphabetical order | N | P |
| SF006 | Click on “IsApproved” header from “FAQ List” table | “FAQ List” was sorted by status in ascending alphabetical order | N | P |
| SF007 | Click on FAQs question “ITHD – WHY IS IT NEEDED” from “FAQ List” table | Navigate to edit FAQ screen | N | P |
| SF008 | Login with employee account : duongls | * Status is read-only * “FAQ List” table only show approved FAQ | N | P |

### View detailed FAQ screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| EF001 | Question | “” | System Message:  Please input FAQs Question!!  Please input FAQs Content!! | A | P |
| Type | Connection |
| Status | Approved |
| Attachments | “” |
| Content | “” |
| EF002 | Question | “123456789” | System Message:  FAQs Question is too short!!  FAQs Content is too short!! | B | P |
| Type | Connection |
| Status | Approved |
| Attachments | “” |
| Content | 49 characters |
| EF003 | Question | 101 characters | System Message:  FAQs Question limit is 100 characters!! | B | P |
| Type | Connection |
| Status | Approved |
| Attachments | “” |
| Content | “Generate detailed reports about Help desk performance & metrics” |
| EF004 | Question | “What is Help desk Report?” | System Message:  You have edited FAQ successfully | N | P |
| Type | Other |
| Status | Approved |
| Attachments | Choose 3 picture file |
| Content | “Generate detailed reports about Help desk performance & metrics” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| EF005 | Login with IT staff account : huyld | Status is read-only | N | P |
| EF006 | Login with employee account : duongls | Button “Edit” is not appear | N | P |
| EF007 | Login with IT staff account : huyld  Create new FAQ in the date 17/07/2012  Login with IT staff account : minhnq  Edit this FAQ in the date 18/07/2012  Login with admin account : longnh  Edit this FAQ in the date 19/07/2012 | Created Day : 17/07/2012  By : huyld  Last Updated Day: 19/07/2012  By : longnh | N | P |

## Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| CR001 | Report Name | “” | System Message:  Please choose From Date!!  Please choose To Date!!  Please input staff name!! | A | P |
| Type of Report | Staff Report |
| From | “” |
| To | “” |
| Type of Data | Chart and table |
| For | One IT staff |
| IT Staff | “” |
| Type of Metric |  |
| CR002 | Report Name | “” | Generate report | N | P |
| Type of Report | Issue Report |
| From | 01/2012 |
| To | 12/2012 |
| Type of Data | Chart and table |
| Type of Metric | Check all |
| Type of issue | All Types |
|  | Press “Generate” |
| CR003 | Report Name | “” | All generated report was displayed on PDF file. | N | P |
| Type of Report | Issue Report |
| From | 01/2012 |
| To | 12/2012 |
| Type of Data | Chart and table |
| Type of Metric | Check all |
| Type of issue | All Types |
|  | Press “Generate” |
|  | Press “Add report” |
|  | Press “Export” |
|  | Save and open PDF file |
| CR004 | Type of report | Staff Report | “For” appears  “IT staff username” appears | N | P |
| For | One IT staff |
| CR005 | Type of report | Staff Report | “For” appears  “IT Team” appears | N | P |
| For | One IT staff |
| CR006 | Type of report | Issue Report | Type of issue appears | N | P |

## Message

### My message screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| MM001 | Message contain | “Log” | Displayed result is match correctly with database | N | P |
| Inbox/Outbox | Inbox |
| Time | This month |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| MM002 | Click on “Id” header from “Message List” table | “Message List” was sorted by Id from largest to smallest | N | P |
| MM003 | Click on “From” header from “Message List” table | “Message List” was sorted by sender’s username in ascending alphabetical order | N | P |
| MM004 | Click on “To” header from “Message List” table | “Message List” was sorted by receiver’s username in ascending alphabetical order | N | P |
| MM005 | Click on “Message” header from “Message List” table | “Message List” was sorted by message’s content in ascending alphabetical order | N | P |
| MM006 | Click on “Time” header from “Message List” table | “Message List” was sorted by time in ascending order | N | P |
| MM007 | Click on “For more detailed: click here” | Navigate to view detailed ticket screen | N | P |

### Create message screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case no. | Fields | Input | Output | State | Result |
| CM001 | Receiver | “” | System Message:  Please input Receiver!!  Please input Message content!! | A | P |
| Content | “” |
| CM002 | Receiver | “huyld111” | System Message:  This Receiver account is not exist!!  Message Content is too short!! | B | P |
| Content | “123456789” |
| CM003 | Receiver | “huyld” | System Message:  Message Content limit is 500 character!! | B | P |
| Content | 501 characters |
| CM004 | Receiver | “huyld” | System Message:You have sent new message successful!! | N | P |
| Content | “Closed issue id 14” |
| CM005 |  | Press “New Message” | Clear all input | N | P |

## Main screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| MS001 | Click “My account” | Navigate to view detailed user screen | N | P |
| MS002 | Click “Log out” | Navigate to login screen | N | P |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case no. MS003 | | | | | | |
| Input | Login with each user role and check which menu appears | | | | | |
| Output | x:appears | | | | | |
|  | Role  Menu | | | Admin | IT Staff | Employee |
| Dashboard |  |  | x | x | x |
| Ticket | My Ticket |  | x | x | x |
|  | Log Ticket | Basic Mode |  |  | x |
| Advanced Mode | x | x |  |
| Search Ticket |  | x | x | x |
| User | Create User |  | x |  |  |
| Search User |  | x | x | x |
| FAQs | Log FAQ |  |  | x |  |
| Search FAQ |  | x | x | x |
| Reports | Create Report |  | x |  |  |
| Messages | My message |  | x | x | x |
|  | Create message |  | x |  |  |
| About |  |  | x | x | x |
| Contact Us |  |  | x | x | x |
| State | N | | | | | |
| Result | P | | | | | |

## Upload function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case no. | Input | Output | State | Result |
| UF001 | Go to Log FAQ screen and assume that other inputs were validate   * Click “SELECT FILES” and choose 3 files | Notification: 3 files were successfully uploaded | N | P |
| UF002 | Go to Log FAQ screen and assume that other inputs were validate   * Click “SELECT FILES” and choose 3 files * Click on “x” icon * In notification “Are you sure you wish to delete this file?” choose “OK” | File is no longer in the queue | N | P |
| UF003 | Go to Log FAQ screen and assume that other inputs were validate   * Click “SELECT FILES” and choose 3 files * Click “View” * Click on icon “>” | Can view all image from the pop-up windows | N | P |
| UF004 | Go to Log FAQ screen and assume that other inputs were validate   * Click “SELECT FILES” and choose 3 files * Press “Log” | Queue appears correctly in next screen | N | P |
| UF005 | Go to Log FAQ screen and assume that other inputs were validate   * Click “SELECT FILES” and choose 2 files * Click “SELECT FILES” and choose the same 2 file again | Notification:  The file named “1.jpg” is already in the queue. Do you want to replace the existing item in the queue?  The file named “2.jpg” is already in the queue. Do you want to replace the existing item in the queue? | A | P |
| UF006 | Go to Log FAQ screen and assume that other inputs were validate   * Click “SELECT FILES” and choose picture file that size larger than 500 kilobyte | Notification  Some file were not added to the queue: The file “4.jpg” exceeds the size limit (500KB) | B | P |

## Auto complete function

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Page | Case no. | Field | Input | Output | State | Result |
| Log Ticket screen | AC001 | Owner | “h” | Suggested list appear with value “thangtd”, “haitct” | N | P |
| AC002 | Status | Assigned | Suggested list appear with value “minhnq”, “quydtd” | N | P |
| Assigned To | “q” |
| Search Ticket screen | AC003 | Owner | “o” | Suggested list appear with value “duongls” | N | P |
| AC004 | Register | “o” | Suggested list appear with value “duongls, longnh, honglm” | N | P |
| AC005 | Respondent | “o” | Suggested list appear with value “honglm” | N | P |
| View detailed Ticket screen | AC006 | Owner | “h” | Suggested list appear with value “thangtd”, “haitct” | N | P |
| AC007 | Status | Assigned | Suggested list appear with value “minhnq”, “quydtd” | N | P |
| Assigned To | “q” |
| Search FAQ screen | AC008 | Creator | “q” | Suggested list appear with value “minhnq”, “quydtd” | N | P |
| Create Report screen | AC009 | Type of report | Staff report | Suggested list appear with value “minhnq”, “quydtd” | N | P |
| For | One IT Staff |
| IT Staff | “q” |
| Create Message screen | AC009 | Receiver | “c” | Suggested list appear with value “haitct” | N | P |

Software User Manual

# Installation Guide

* Server must support Apache Tomcat 6.0.26 or an upper version, and Microsoft SQL Server 2005 or upper.
* Copy TMP.war into folder webapps of Tomcat. Run file script TMP\_Database.sql in SQL Server.
* Start server and provide user an url to eSmartDesk service

# User’s Guide

## Minimum System Requirements

The server that host eSmartDesk must meet the following minimum requirement:

1GHz CPU or above.

256MB RAM or above.

500GB space on HDD or above.

Window XP or an upper version.

Have an internet connection.

The client computer that use eSmartDesk must meet the following minimum requirement:

1GHz CPU or above.

256MB RAM or above.

Window XP or an upper version.

Screen resolution need to be 1024x768 or higher.

Have internet connection.

Have Firefox 3.0 or later.

## Introduction to Smart Desk

eSmartDesk provides a single point of contact for IT issue managing and solving in company. Reduces complexity for help desk management, optimizes your employees' work and allows you to stay focus on your core business.

## Features and How to Use



### Tickets

#### My Ticket

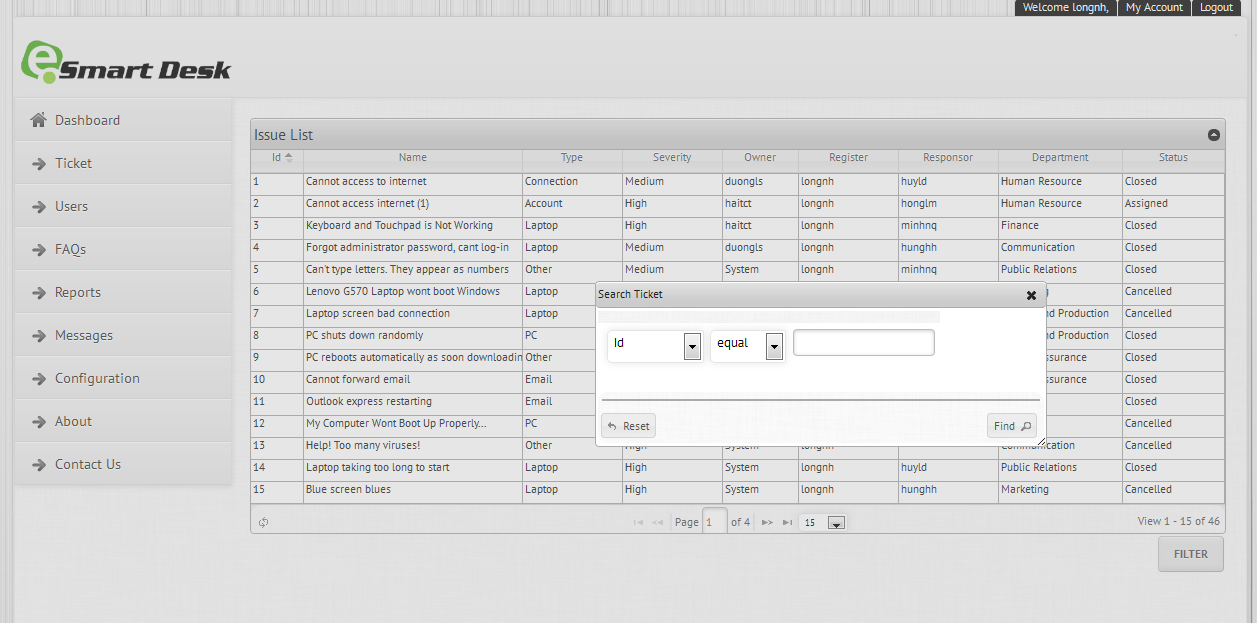


Figure 59 – My Ticket Screen

* This function is for all account (Admin, IT Staff, Employee)
* With this function, you can view a list of tickets belong to you.
* To search for ticket:

1. Click the Filter button.
2. Enter the search’s condition.
3. Click the Find button to perform the search.
4. Click on the link under Ticket’s Name column to view the detailed information of tickets.

#### Log Basic Ticket

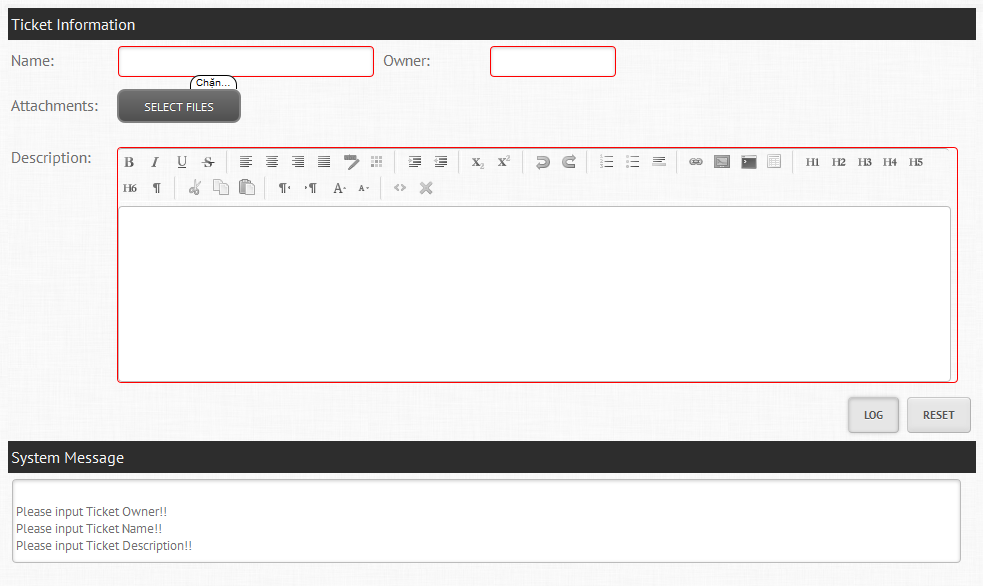


Figure 60 – Log Basic Ticket Screen

* This is **Employee’s** function.
* To log a ticket in basic mode:

1. Click on Ticket -> Log Ticket -> Basic Mode in Navigation Menu
2. Enter the following information:

|  |  |
| --- | --- |
| Name | Enter a name of the ticket.  The entered name must be at least 10 characters and max 50 characters.  This is **Required Field.** |
| Owner | Enter the owner’s account of ticket.  The entered account must be exist and activate in system.  This is **Required Field.** |
| Attachments | Upload the evidence image of issue. |
| Description | Enter information about any troubleshooting you have performed for this request.  This is **Required Field.** |

1. Click the Log button to submit ticket.

#### Log Advanced Ticket

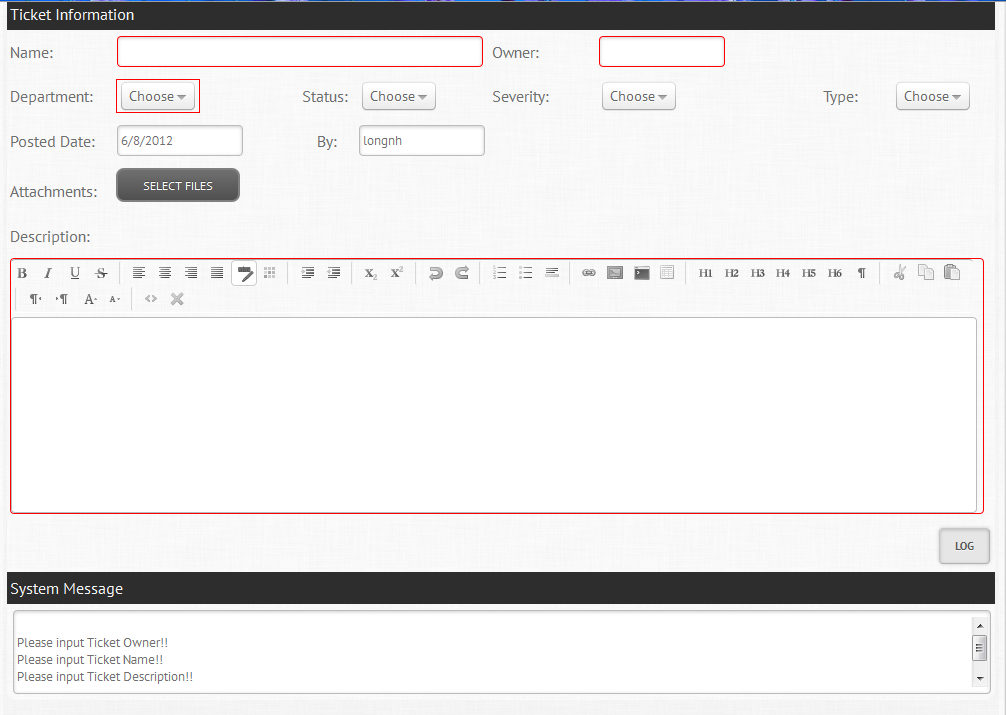


Figure 61 – Log Advanced Ticket Screen

* This is **Admin** and **ITStaff’s** function.
* To log a ticket in advanced mode:

1. Click on Ticket -> Log Ticket -> Advanced Mode in Navigation Menu
2. Enter the following information:

|  |  |
| --- | --- |
| Name | Enter a name of the ticket.  The entered name must be at least 10 characters and max 50 characters.  This is **Required Field.** |
| Owner | Enter the owner’s account of ticket.  The entered account must be exist and activate in system.  This is **Required Field.** |
| Department | Use the drop down menus to select the appropriate department where issue’s happened.  This is **Required Field.** |
| Status | Use the drop down menus to select the appropriate status of ticket.  Default is **Open.** |
| Severity | Use the drop down menus to select the appropriate severity of ticket. |
| Type | Use the drop down menus to select the appropriate type of ticket. |
| Posted Date | This is the posted date of issue.  System loads it automatically. |
| By | This is the account which log issue.  System loads it automatically. |
| Attachments | Upload the evidence image of issue. |
| Description | Enter information about any troubleshooting you have performed for this request.  This is **Required Field.** |

1. Click the Log button to submit ticket.

#### Search Ticket

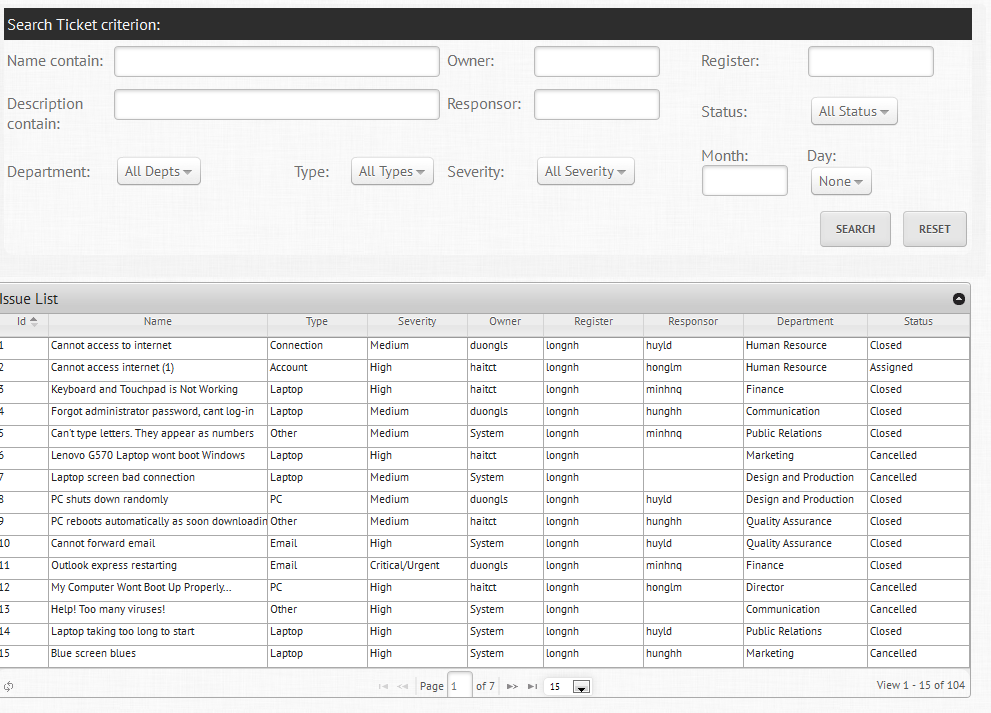


Figure 62 – Search Ticket Screen

* This function is for all account (Admin, IT Staff, Employee)
* To search a ticket:

1. Click on Ticket -> Search Ticket in Navigation Menu.
2. Enter your search criteria.
3. Click Search button to perform the search.
4. Click on the link under Ticket’s Name column to view the detailed information of tickets.

#### Edit Ticket

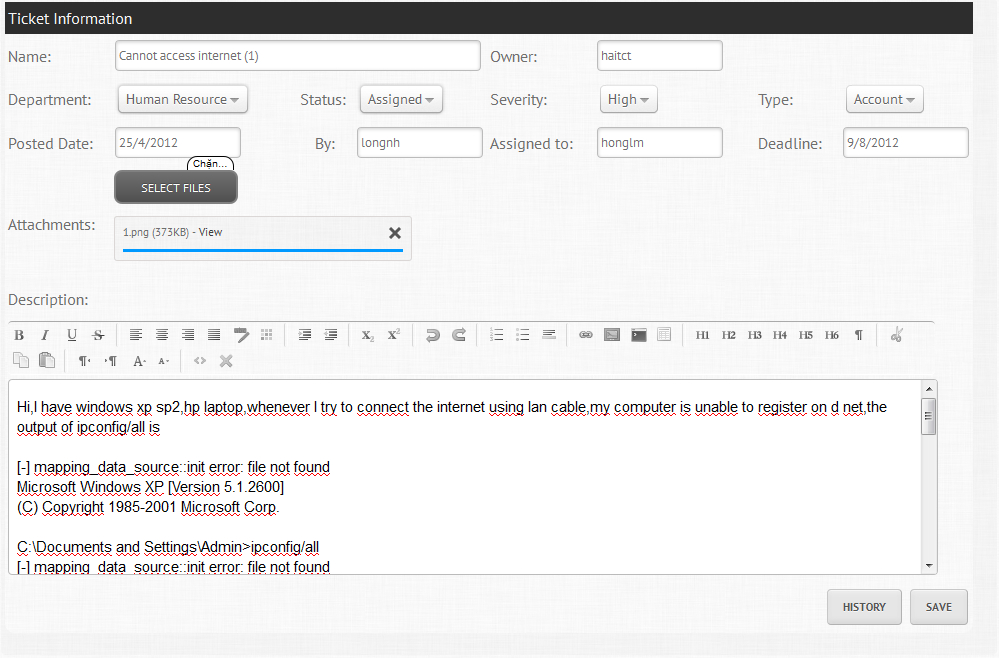


Figure 63 - Edit Ticket Screen

* This function is for all account (Admin, IT Staff, Employee)
* To edit a ticket:

1. Click link to view ticket.
2. Edit any information you want:

|  |  |
| --- | --- |
| Name | Enter a name of the ticket.  The entered name must be at least 10 characters and max 50 characters.  This is **Required Field.** |
| Owner | Enter the owner’s account of ticket.  The entered account must be exist and activate in system.  This is **Required Field.** |
| Department | Use the drop down menus to select the appropriate department where issue’s happened.  This is **Required Field.** |
| Status | Use the drop down menus to select the appropriate status of ticket. |
| Severity | Use the drop down menus to select the appropriate severity of ticket. |
| Type | Use the drop down menus to select the appropriate type of ticket. |
| Posted Date | This is the posted date of issue.  System loads it automatically. |
| By | This is the account which log issue.  System loads it automatically. |
| Attachments | Upload the evidence image of issue. |
| Description | Enter information about any troubleshooting you have performed for this request.  This is **Required Field.** |

1. Click Save to submit.

### Users

#### Create User

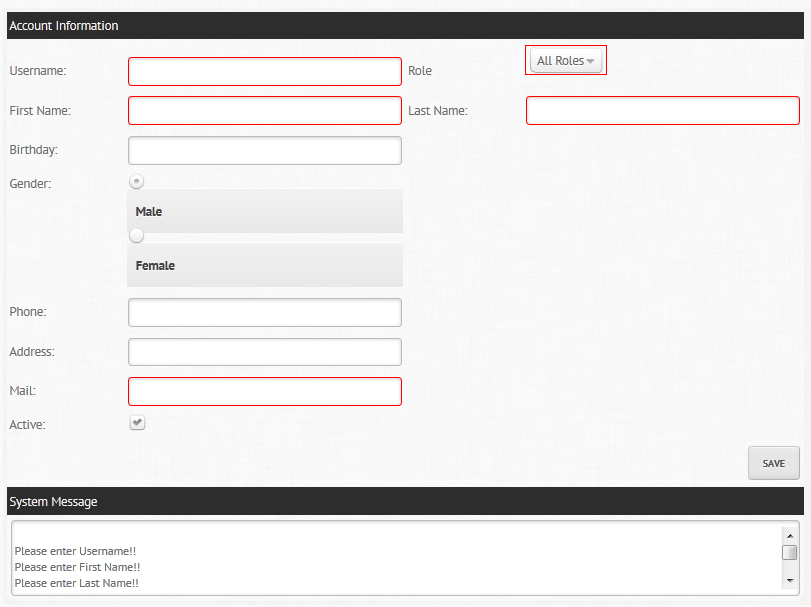


Figure 64 – Create User Screen

* This is **admin’s** function
* To create an account:

1. Click on User -> Create User in Navigation Menu
2. Enter the following information:

|  |  |
| --- | --- |
| Username | The entered username must be at least 3 characters, max 16 characters and cannot contain space or special characters.  The entered username cannot be duplicate with any exist account in system.  This is **Required Field.** |
| Role | Use the drop down menus to select the appropriate role.  This is **Required Field.** |
| IT Type (Only for role IT Staff) | Use the drop down menus to select the appropriate IT Type.  This is **Required Field.** |
| Department (Only for role Employee) | Use the drop down menus to select the appropriate department.  This is **Required Field.** |
| First Name | Enter the first name of user.  This is **Required Field.** |
| Last Name | Enter the last name of user.  This is **Required Field.** |
| Birthday | Select the birthday of user. |
| Gender | Select the gender of user.  Default is Male. |
| Phone | Enter the Phone of user.  The entered phone must contains only number and max 11 characters. |
| Address | Enter the address of user. |
| Email | Enter the email of user.  This is **Required Field.** |
| Active | Check to mark user as active or inactive.  Default is Active. |

1. Click save button to submit user.

#### Search User

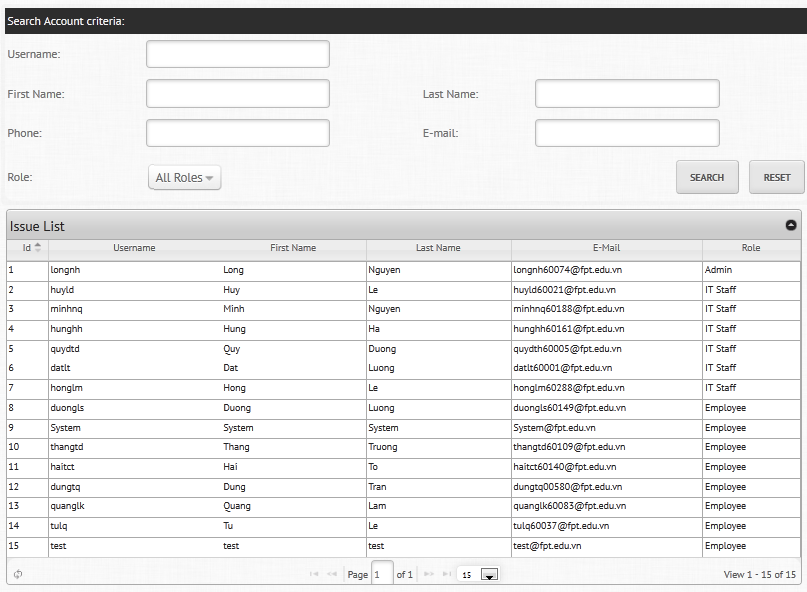


Figure 65 - Search User Screen

* This function is for all account (Admin, IT Staff, Employee)
* To search a user:

1. Click on Users -> Search User in Navigation Menu.
2. Enter your search criteria.
3. Click Search button to perform the search.
4. Click on the link under Username column to view the detailed information of User.

#### Edit User

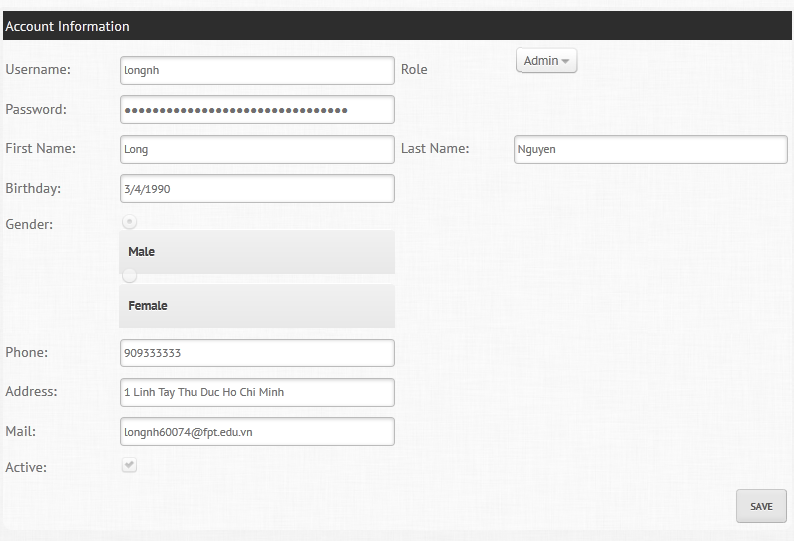


Figure 66 - Edit User Screen

* This function is for all account (Admin, IT Staff, Employee)
* To edit a user:

1. Click link to view ticket.
2. Edit any information you want:

|  |  |
| --- | --- |
| Username | The entered username must be at least 3 characters, max 16 characters and cannot contain space or special characters.  The entered username cannot be duplicate with any exist account in system.  This is **Required Field.** |
| Role | Use the drop down menus to select the appropriate role.  This is **Required Field.** |
| IT Type (Only for role IT Staff) | Use the drop down menus to select the appropriate IT Type.  This is **Required Field.** |
| Department (Only for role Employee) | Use the drop down menus to select the appropriate department.  This is **Required Field.** |
| First Name | Enter the first name of user.  This is **Required Field.** |
| Last Name | Enter the last name of user.  This is **Required Field.** |
| Birthday | Select the birthday of user. |
| Gender | Select the gender of user.  Default is Male. |
| Phone | Enter the Phone of user.  The entered phone must contain only number and max 11 characters. |
| Address | Enter the address of user. |
| Email | Enter the email of user.  This is **Required Field.** |
| Active | Check to mark user as active or inactive.  Default is Active. |

1. Click Save button to submit.

### FAQs

#### Create FAQ

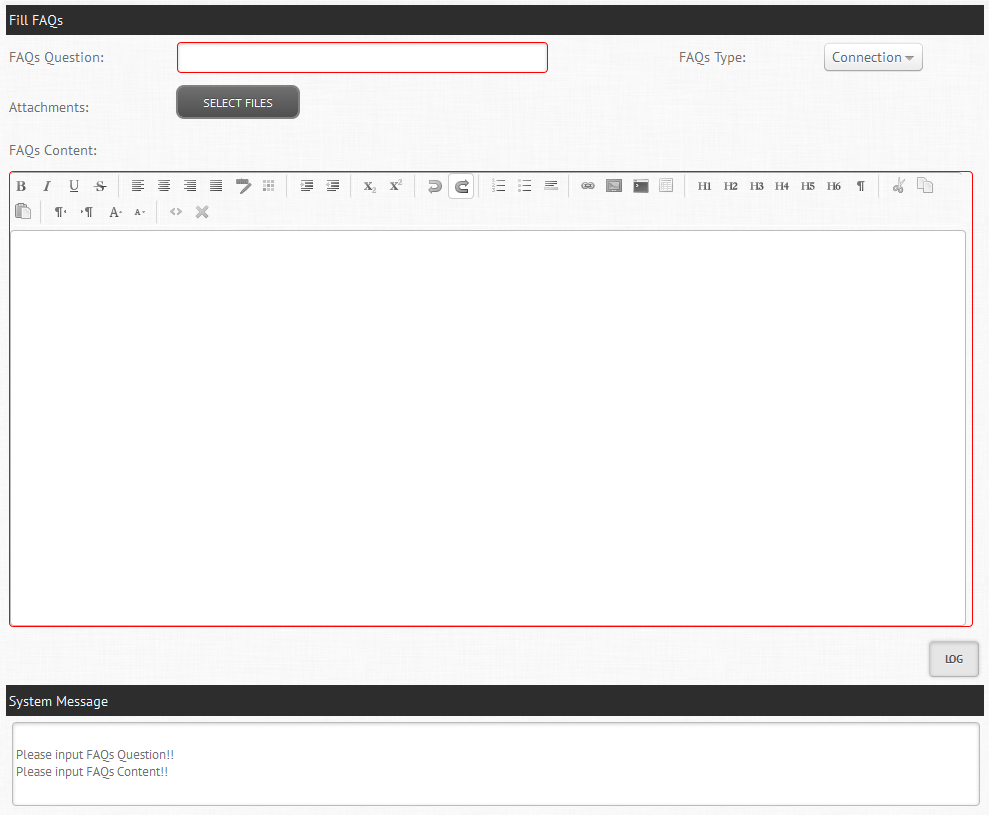
****

Figure 67 - Create FAQs Screen

* This is **IT Staff’s** function
* To create an FAQ:

1. Click on User -> Create User in Navigation Menu
2. Enter the following information:

|  |  |
| --- | --- |
| FAQ Question | Enter the question of FAQ.  The entered question must be at least 10 characters and max 100 characters.  This is **Required Field.** |
| FAQ Type | Use the drop down menus to select the appropriate type. |
| Attachments | Upload the evidence image of FAQ. |
| FAQ Content | Enter the content of FAQs.  The entered content must be at least 50 characters.  This is **Required Field.** |

1. Click save button to submit FAQ.

#### Search FAQ

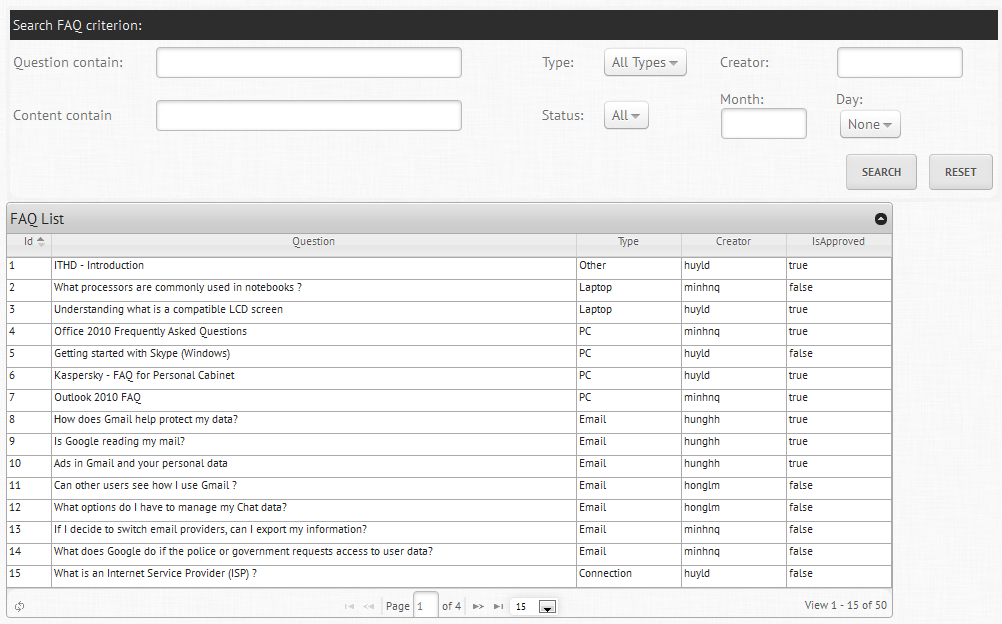


Figure 68 - Search FAQs Screen

* This function is for all account (Admin, IT Staff, Employee)
* To search a FAQ:

1. Click on FAQs -> Search FAQ in Navigation Menu.
2. Enter your search criteria.
3. Click Search button to perform the search.
4. Click on the link under Question column to view the detailed information of FAQ.

#### Edit FAQ

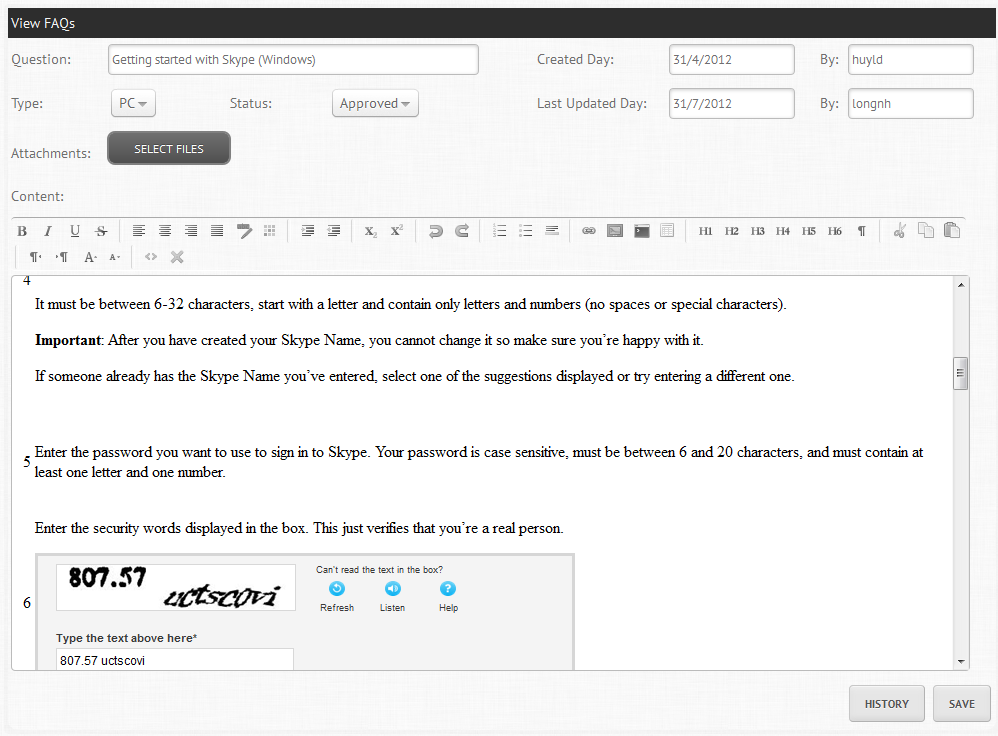


Figure 69 - Edit FAQs Screen

* This function is for all account (Admin, IT Staff)
* To edit a FAQ:

1. Click link to view FAQ.
2. Edit any information you want:

|  |  |
| --- | --- |
| FAQ Question | Enter the question of FAQ.  The entered question must be at least 10 characters and max 100 characters.  This is **Required Field.** |
| FAQ Type | Use the drop down menus to select the appropriate type. |
| Attachments | Upload the evidence image of FAQ. |
| FAQ Content | Enter the content of FAQs.  The entered content must be at least 50 characters.  This is **Required Field.** |

1. Click Save button to submit.

### Reports

#### Create Reports

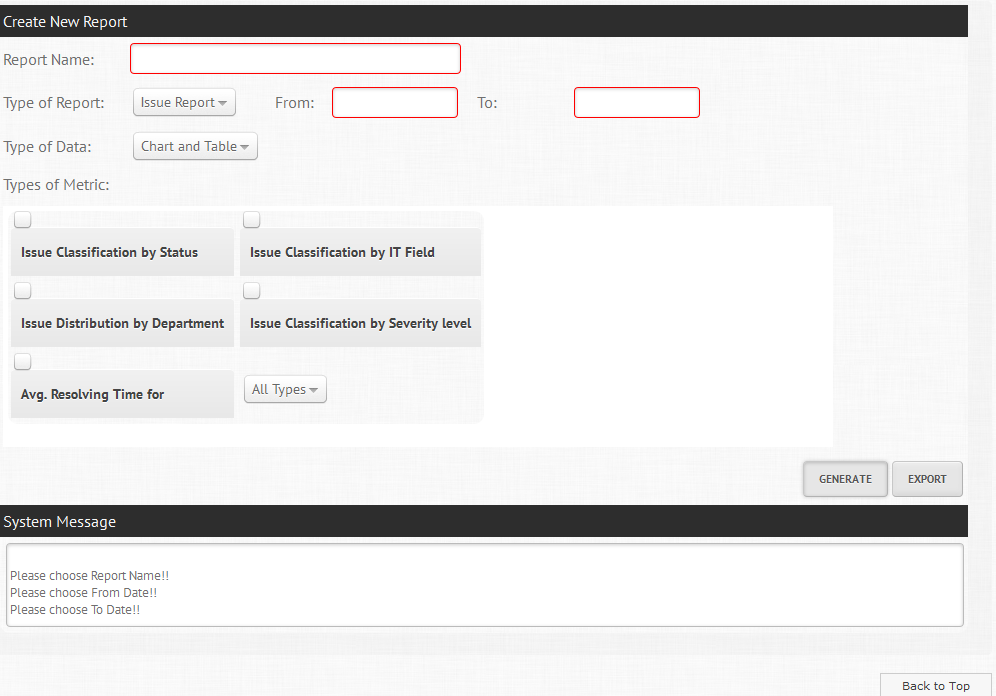
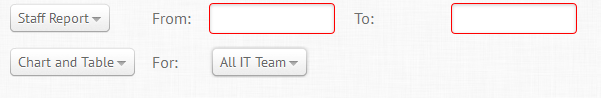
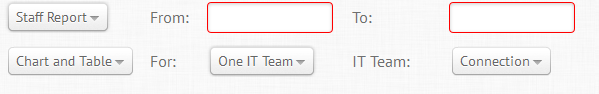
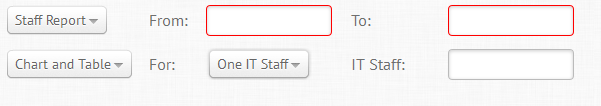


Figure 70 - Create Report Screen







* This is admin’s function.
* To create report:

1. Click on Reports -> Create Report in Navigation Menu.
2. Enter the following information:

|  |  |
| --- | --- |
| Report Name | Enter the name of report.  This is **Required Field.** |
| From | This is **Required Field.** |
| To | This is **Required Field.** |
| Type of Data | Use the drop down menus to select the appropriate type of data. |
| Type of Metric | Check type of metrics you want for your report. |
| Type of Report | Use the drop down menus to select the appropriate type of report. |
| For (Only exist for **Staff Report**) | Use the drop down menus to select the appropriate value (All IT Team/One IT Team/One IT Staff) |
| IT Team (Only exist for **One IT Team** option) | Use the drop down menus to select the appropriate IT Team. |
| IT Staff (Only exist for **One IT Staff** option) | Enter the IT Staff’s account.  The entered account must be exist and activate in system.  This is **Required Field.** |

1. Click Generate button to generate the report.
2. Click Add to PDF to add the chart to PDF File.
3. Click Export to generate PDF File.

### Configurations

#### Create Department/IT Type

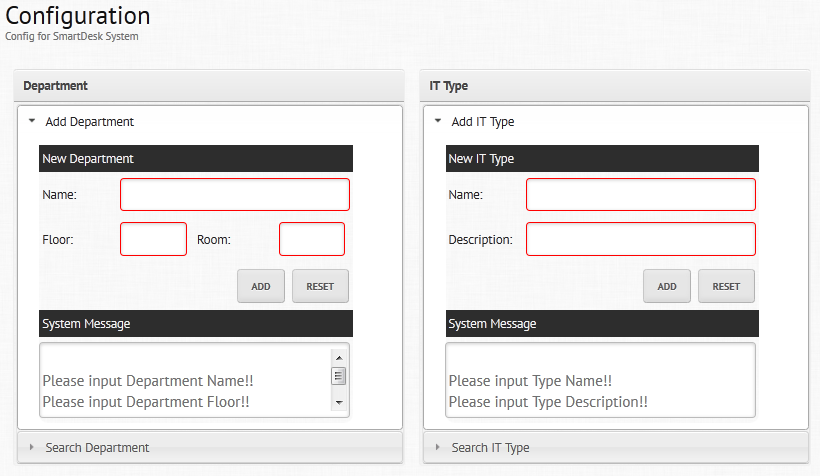


Figure 71 - Create Department/IT Type Screen

* This is admin’s function.
* To add new **Department**:

1. Click on Configuration in Navigation Menu.
2. Enter the following information:

|  |  |
| --- | --- |
| Name | Enter the name of Department.  This is **Required Field.** |
| Floor | Enter the floor of Department.  This is **Required Field.** |
| Room | Enter the room of Department.  This is **Required Field.** |

1. Click the Add button to Submit.

* To add new **IT Type**:

1. Click on Configuration in Navigation Menu.
2. Enter the following information:

|  |  |
| --- | --- |
| Name | Enter the name of IT Type.  This is **Required Field.** |
| Description | Enter the description of IT Type.  This is **Required Field.** |

1. Click the Add button to Submit.

#### Search Department/ IT Type

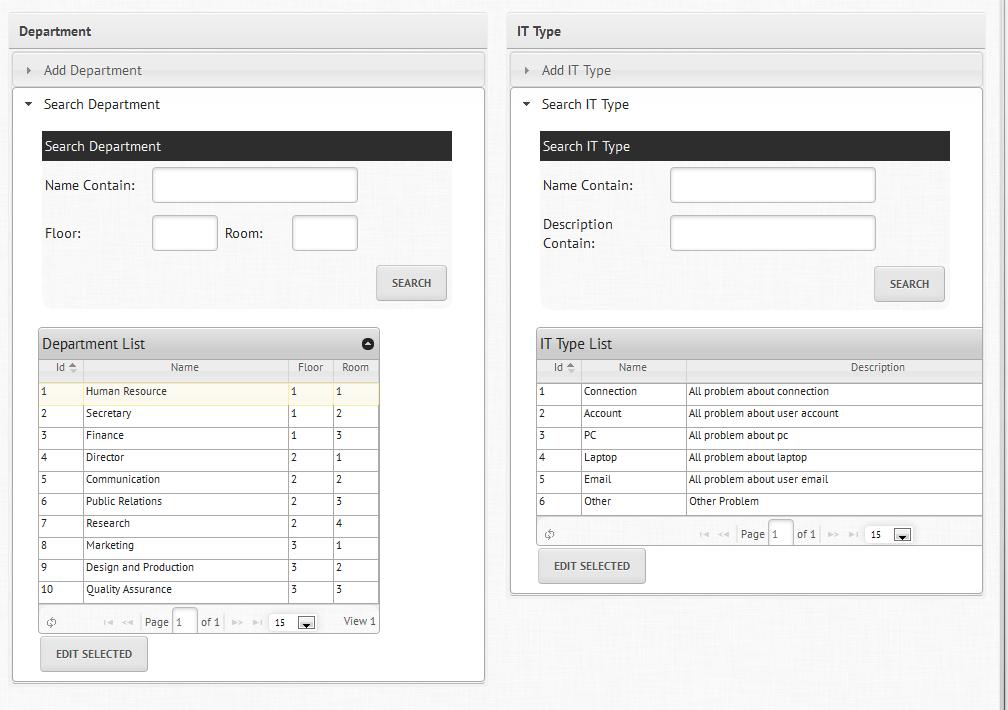


Figure 72 – Search Department/IT Type Screen

* This is admin’s function.
* To search a Department/IT Type:

1. Click on Search Department/Search IT Type.
2. Enter your search criteria.
3. Click Search button to perform the search.
4. Click on the Department/IT Type’s Name column and click the button Edit Selected to edit the information of Department/IT Type.

#### Edit Department/IT Type

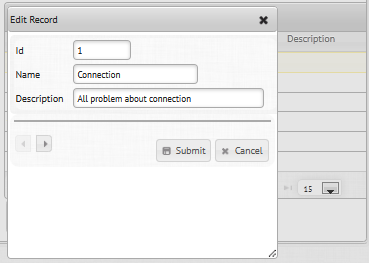
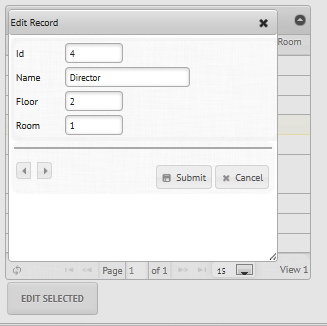


Figure 73 - Edit Department/IT Type Screen

* This is admin’s function.
* To edit a Department:

1. Enter any information you want to edit.

|  |  |
| --- | --- |
| Name | Enter the name of Department.  This is **Required Field.** |
| Floor | Enter the floor of Department.  This is **Required Field.** |
| Room | Enter the room of Department.  This is **Required Field.** |

1. Click Submit button to edit.

* To edit a IT Type:

1. Enter any information you want to edit.

|  |  |
| --- | --- |
| Name | Enter the name of IT Type.  This is **Required Field.** |
| Description | Enter the description of IT Type.  This is **Required Field.** |

1. Click Submit button to edit.

#### Email Notification Setting

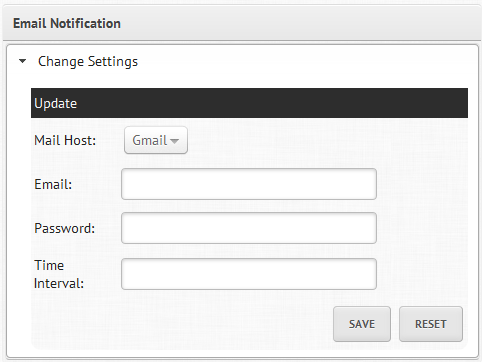


Figure - Email Notification Setting Screen

* This is admin’s function.
* To change email notification settings:

1. Enter any information you want to change.

|  |  |
| --- | --- |
| Mail Host | Choose the mail host |
| Email | Enter mail account.  This is **Required Field.** |
| Password | Enter mail password  This is **Required Field.** |
| Time Interval | Enter auto send mail after time interval  This is **Required Field.** |

1. Click Save button to change.

### Notifications

#### View Message



Figure 75 - View Message Screen

* This function is for all account (Admin, IT Staff)
* To view you message:

1. Click on Message -> My Message in Navigation Menu. (System will show you a list of Messages you have sent (Outbox) and you have received (Inbox))
2. Fill the search criteria if you want to search for any message.
3. Click Search button to Search.
4. Click on the **Click here** link under Message column to view detailed information of message.

#### Create Message



Figure 76 - Create Message Screen

* This function is for all account (Admin, IT Staff)
* To create a message:

1. Click on Message -> Create Message in Navigation Menu.
2. Enter the following information:

|  |  |
| --- | --- |
| Receiver | Enter the message’s receiver.  The entered account must be exist and activate in system.  This is **Required Field.** |
| Content | Enter the content of message.  The entered content must be at least 10 characters and max 500 characters.  This is **Required Field.** |

1. Click the Send button to send message.

Appendix

# Tables

[Table 1 - Comparision of some IT Helpdesk Software 4](#_Toc332354045)

[Table 2 - Members of Project 8](#_Toc332354046)

[Table 3 - Task 10](#_Toc332354047)

[Table 4 - Task Sheet 11](#_Toc332354048)

[Table 5 - Coding Convention 12](#_Toc332354049)

[Table 6 - Screens 13](#_Toc332354050)

[Table 7 - Actors 13](#_Toc332354051)

# Figures

[Figure 1 - Waterfall model 14](#_Toc332493288)

[Figure 2 - Overall Use-Case Diagram 21](#_Toc332493289)

[Figure 3 - Issue Managing Feature Use-Case Diagram 22](#_Toc332493290)

[Figure 4 - User Managing Feature Use-Case Diagram 29](#_Toc332493291)

[Figure 5 - User Notifying Feature Use-Case Diagram 36](#_Toc332493292)

[Figure 6 - Report Managing Feature Use-Case Diagram 41](#_Toc332493293)

[Figure 7 - FAQs Managing Feature Use-Case Diagram 43](#_Toc332493294)

[Figure 8 - Entity Relationship Diagram 51](#_Toc332493295)

[Figure 9 - MVC Architecture 52](#_Toc332493296)

[Figure 10 - MVP model 53](#_Toc332493297)

[Figure 11 - MVVM model 54](#_Toc332493298)

[Figure 12 - Component Diagram 55](#_Toc332493299)

[Figure 13 Class diagram of Managing message 56](#_Toc332493300)

[Figure 14 - ClassDiagram of Managing Issue 59](#_Toc332493301)

[Figure 15 - ClassDiagram of Mail notificatiion 63](#_Toc332493302)

[Figure 16 - ClassDiagram of Managing FAQ 67](#_Toc332493303)

[Figure 17 Class diagram of DTO component 70](#_Toc332493304)

[Figure 18 Login Class Diagram 76](#_Toc332493305)

[Figure 19 Class diagram of Managing user 79](#_Toc332493306)

[Figure 20 Class diagram of Managing Report 84](#_Toc332493307)

[Figure 21 - Log Basic Issue SD 86](#_Toc332493308)

[Figure 22 - Log Advanced Issue SD 87](#_Toc332493309)

[Figure 23 - Search Issue SD 88](#_Toc332493310)

[Figure 24 - Edit Issue SD 88](#_Toc332493311)

[Figure 25 - View Detailed Issue SD 89](#_Toc332493312)

[Figure 26 - Update new user SD 90](#_Toc332493313)

[Figure 27 - Search User SD 91](#_Toc332493314)

[Figure 28 - View User SD 91](#_Toc332493315)

[Figure 29 - Edit detailed user SD 92](#_Toc332493316)

[Figure 30 - Produce report SD 93](#_Toc332493317)

[Figure 31 - Export Report to PDF 93](#_Toc332493318)

[Figure 32 – Create Message SD 94](#_Toc332493319)

[Figure 33 – Show Message SD 95](#_Toc332493320)

[Figure 34 – Mail Notification SD 95](#_Toc332493321)

[Figure 35 – View FAQs SD 96](#_Toc332493322)

[Figure 36 – Create FAQs SD 96](#_Toc332493323)

[Figure 37 – Search FAQs SD 97](#_Toc332493324)

[Figure 38 - Edit FAQs SD 97](#_Toc332493325)

[Figure 39 – Sitemap 98](#_Toc332493326)

[Figure 40 – Login Screen 99](#_Toc332493327)

[Figure 41 – Log Ticket: Basic Mode Screen 100](#_Toc332493328)

[Figure 42 – Log Ticket: Advanced Mode Screen 101](#_Toc332493329)

[Figure 43 – Search Issue Screen 102](#_Toc332493330)

[Figure 44 – View Issue Screen 103](#_Toc332493331)

[Figure 45 – Create User Screen 104](#_Toc332493332)

[Figure 46 – Search User Screen 105](#_Toc332493333)

[Figure 47 – View User Screen 106](#_Toc332493334)

[Figure 48 – Create FAQs Screen 107](#_Toc332493335)

[Figure 49 – Search FAQs Screen 108](#_Toc332493336)

[Figure 50 – View FAQs Screen 109](#_Toc332493337)

[Figure 51 – Create Message Screen 110](#_Toc332493338)

[Figure 52 – My Message Screen 111](#_Toc332493339)

[Figure 53 – Create Report Screen 112](#_Toc332493340)

[Figure 54 – Configuration Screen (Create) 113](#_Toc332493341)

[Figure 55 - Configuration Screen(Search) 114](#_Toc332493342)

[Figure 56 - Configuration Screen(Edit Department) 115](#_Toc332493343)

[Figure 57 - Configuration Screen(Edit IT Type) 115](#_Toc332493344)

[Figure 58 – Database Design 116](#_Toc332493345)

[Figure 59 – My Ticket Screen 163](#_Toc332493346)

[Figure 60 – Log Basic Ticket Screen 164](#_Toc332493347)

[Figure 61 – Log Advanced Ticket Screen 165](#_Toc332493348)

[Figure 62 – Search Ticket Screen 166](#_Toc332493349)

[Figure 63 - Edit Ticket Screen 167](#_Toc332493350)

[Figure 64 – Create User Screen 168](#_Toc332493351)

[Figure 65 - Search User Screen 169](#_Toc332493352)

[Figure 66 - Edit User Screen 170](#_Toc332493353)

[Figure 67 - Create FAQs Screen 171](#_Toc332493354)

[Figure 68 - Search FAQs Screen 172](#_Toc332493355)

[Figure 69 - Edit FAQs Screen 173](#_Toc332493356)

[Figure 70 - Create Report Screen 174](#_Toc332493357)

[Figure 71 - Create Department/IT Type Screen 175](#_Toc332493358)

[Figure 72 – Search Department/IT Type Screen 176](#_Toc332493359)

[Figure 73 - Edit Department/IT Type Screen 177](#_Toc332493360)

[Figure 74 - Email Notification Setting Screen 178](#_Toc332493361)

[Figure 75 - View Message Screen 179](#_Toc332493362)

[Figure 76 - Create Message Screen 180](#_Toc332493363)