**COMP 8045**

**Practicum Report**

**Suplus Office Management System**

Sabrina Xue

A00681622

Table of Contents

1. Introduction 4

1.1 Project Description 4

1.2 Current Problems 4

2. Company Background 4

3. Project Objectives 5

4. Solution chosen for this project 7

PHP 7

Bootstrap 7

jQuery 7

JSP 7

CSS 8

Mongo database 8

5. Development Methodology 9

6. Architecture and design 10

Component Diagram 11

7. Analysis and Design 12

7.1 System Function 12

Manage employee information 12

Manage customer information 12

Manage project information 13

Login 13

Communicate 13

7.2 Use Case Diagram 14

7.3 Domain Model Diagram 16

7.4 System Sequence Diagram 17

7.5 Software Design Class Diagram 24

7.6 Database Design 25

MySQL 25

Employee table 27

Customer table 27

Department table 29

Project table 29

Task table 30

NoSQL 31

Collection 31

Post message 32

Post picture 33

Comment 34

8. Access Control 36

Role based access control 36

9. Innovation 36

10. Development and Implementation 37

10.1 Programming Languages Chosen 37

10.2 System Perspective 37

Employee module 39

Administrator module 42

Team leader module 51

11. Research 55

11.1 Why NoSQL? 55

11.2 Advantages & Disadvantages 55

11.3 Why Mongo DB? 57

12. Test 58

12.1 User Test Plan 58

Result of user testing 59

Debugging 63

12.2 System Functional Testing 65

Result of system functional testing 65

All users’ functions testing result 65

Employee module testing result 66

Admin module testing result 67

Team leader module testing result 68

13. Conclusion 70

14. Future Enhancement 71

15. Appendix 72

A. Work Breakdown 72

B. Test Cases 74

C. Testers’ Survey of User Testing 103

Reference 104

## Introduction

### 1.1 Project Description

The aim of this practicum project is to develop a system to connect and organize the information of employees, projects, and customers of Suplus Technology. The Suplus Office Management System is designed to be utilized by the employees, administrators, and team leaders in the company. An important aspect of the Management System is that it permits its users to communicate efficiently and effectively. The system enables its users to exchange messages and share pictures. This paper will outline the functions of the system and how it benefits the aforementioned users. Firstly, the system allows the employees to view personal projects and get tasks’ information. Secondly, with the help of the system, the administrators can organize information of employees, customers and projects, such as adding, editing and deleting information. Thirdly, the system enables the team leaders to divide each project into small tasks and assign these tasks to employees in the software development department. This document includes all the necessary information that explains how the Management System operates.

### 1.2 Current Problems

The purpose of this project is to provide solution to the following:

1. How to arrange the system architecture according to the client’s (Suplus Technology Co. Ltd.) requirement.
2. How to design a user friendly and functional user interface.
3. How to design a social website with unstructured data supported by a suitable database.
4. How to apply a suitable database to support structured data.

## Company Background

**The industry sponsor** is Beijing Suplus Technology Co. Ltd.

Suplus Technology Co. Ltd aims at exploring China Enterprise Communications and application market and is dedicated to the development and promotion of new technologies and new products. Suplus Technology has helped China Unicom, China Mobile, China Telecom, and Jinjiang Inn to build call center system. In order to expand its market to the southern part of China, Suplus Technology Co. Ltd is going to set more branches in the region. Therefore, the company requires a better functional and more convenient system to manage the employees, customers and projects.

## Project Objectives

The objective of the project is to help the employees of Suplus Technology to organize projects and tasks better and to maintain the pace of the work schedule. As a result, this system would improve the work efficiency. At the same time, Suplus Office Management System provides a platform that allows employees to exchange information and share their personal information.

Suplus Office Management System consists of three actors, and based on these actors the system could be operated in three modules: employee module, administrator module, and team leader module. As mentioned before, each project could be divided into several small tasks. Employees could check personal projects and tasks. Administrators could manage the employees, customers and projects’ information. Team leaders could allocate and add tasks for employees. In addition, there is a communication page allowing employees to share information.

* The software development employee module includes the following main functions:
* Check personal projects and tasks’ information
  + Check personal tasks and projects’ information
  + Search a specific project or task
* Modify personal information
  + Update personal information such as username, password, status, address, post code, email, and phone number
* Communication
  + Post messages
  + Post pictures
  + Comment on posted messages or pictures
* The Administrator module consists of the following functions:
* Employee maintenance
* Check employees’ information
* Search for information of a specific employee
* Add new employees
* Edit employees’ information
* Delete employees’ information
* Customer maintenance
* Check customer’s profile
* Search for information of a specific customer
* Add new customers
* Edit customers’ information
* Delete customers’ information
* Project maintenance
* Check the whole projects’ information
* Search for information of a specific project
* Add new projects
* Edit projects’ information
* Delete projects’ information
* Modify personal information
  + Update personal information such as username, password, status, address, post code, email, and phone number
* Communication
  + Post messages
  + Post pictures
  + Comment on posted messages or pictures
* The team leader module has the following functions:
* Manage project
* Check the entire projects’ information
* Divide a project into small tasks
* Assign each task to an employee
* Update personal information
  + Update personal information such as username, password, status, address, post code, email, and phone number
* Communication
  + Post messages
  + Post pictures
  + Comment on posted messages or pictures

## Solution chosen for this project

The technologies are used to develop this system are all open source software according to the industry sponsor requirement in order to cut the cost of development. The technologies include:

* MAMP 3.0.7.3 (Manage apache and MySQL server)
* PHP 5.5 as programming language
* HTML CSS
* JavaServer Pages (JSP)
* jQuery
* Bootstrap
* MySQL Server 5.5.38
* Mongo Database
* RoboMongo 0.8.4

### PHP

PHP is a server-side scripting language designed for web development. Since PHP code can mix with HTML code, various engines and web frameworks, Suplus Office Management System adapted PHP as the major programming language.

### Bootstrap

As an extension of Java script, Bootstrap contains HTML and CSS designed templates. It is compatible with most major browsers and supports responsive design. In this system, Bootstrap is the most important part because almost every page includes Bootstrap themes, such as table style, search textbox, and button style.

### jQuery

jQuery's architecture allows developers to create plug-in code to extend its functionality. jQuery library allows the creation of powerful dynamic web pages and web applications. In this system, jQuery is combined with Bootstrap to update the interface design.

### JSP

JSP helps developers to create dynamically web pages based on HTML or other document types. It can help to change the presentation of the system without learning Java Script. Java Server pages provide a better solution to the creation of dynamic and interactive web pages by embedding programming language directly into HTML.

### CSS

CSS is a [style sheet language](http://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [looks and formatting](http://en.wikipedia.org/wiki/Presentation_semantics) a document written in a [mock-up language](http://en.wikipedia.org/wiki/Markup_language). CSS is a cornerstone technology used by most websites to create user interfaces. This system is using CSS to create a user-friendly interface, especially on the page layout arrangement.

### Mongo database

Mongo DB is particularly good at storing and organizing data as documents. Mongo DB could provide better data model to fit in data structure than relational database. In order to explore and practice the new technology, this system adapted Mongo DB to store the data coming from the communication page. For more detail, please check 7.6 Database Design and 11.3 Why Mongo DB.

## Development Methodology

The sponsor required using prototype as the development methodology. Therefore, this project will adapt the prototype methodology. According to the client’s requirement, I have designed the following versions:

1. Version 1.0

Version 1.0 is a simple mock-ups based on the users’ requirement. Though it is not the perfect version, it does help organize and list the functions of the employee, administrator and team leader modules. For example, updating personal information and communicating are the major functions for all the users. For the administrator module, there are functions for managing the information of employees, customers and projects. For the team leader module, functions include managing projects based on tasks and allocating tasks to selected employees. The version has been presented to the users of Suplus Technology, and the users have provided suggestions.

1. Version 1.1

Based on the users’ feedbacks, I updated version 1.0 to version 1.1. Version 1.1 helped create the next version.

1. Version 1.2   
   Version 1.1 generated version 1.2 with HTML and CSS. This version changed the previous mock-ups into a web system by using HTML and CSS.
2. Version 1.3

Version 1.3 adapted PHP, MySQL and Mongo DB for the development of its system. The user testing and the system functional testing are based on this version. The testers include the employees, administrators, team leaders of Suplus Technology, volunteers, and myself. All the testers were asked to provide the test results and suggestions.

1. Version 1.4

According to the testing results and suggestions, version 1.4 debugged and adapted Bootstrap to improve the page layout of user interface. The version 1.4 has been updated as the final version.

The prototype methodology helped refine the system and reduced the possibility of missing functions.

## Architecture and Design

Server Configuration

* Web server: Apache 2.4.9 or higher
* Server-side scripting language: PHP 5.5.12 or higher
  + Operation System: Windows 7/8, UNIX or Linux Operating System
  + Database: MySQL Server 5.5.38, Mongo Database V2.6.7
  + IE Browser: Safari, Chrome

Client Configuration

* + Software: Windows 7/8, UNIX or Linux Operating System
* IE Browser: Safari, Chrome

Suplus Office Management System allows users to access user interface by using web browser through the Internet. Since there are two databases adapted, user interface allows the system users to send SQL and NoSQL command from application server to database server in order to complete tasks.

### Component Diagram

There are three components, the first one is the user interface layer, users could login the system by using any web browser. The second layer is coding source, which includes the employee module, admin module, and team leader module. The programming languages adapted are PHP, jQuery, JavaScript, and Bootstrap. The last component is databases, the information are stored in MySQL server and NoSQL (Mongo DB). The following picture could explain the three components.

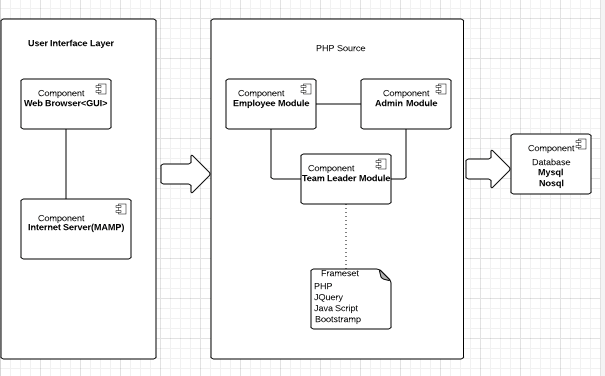


Figure 1 Component Diagram

## Analysis and Design

### 7.1 System Function

Suplus Office Management System has five major functions: manage employees’ information, manage customers’ information, manage projects’ information, login, and communicate. The function “Communicate” allows users to share information, and make comments on others’ status.

#### Manage employee information

The purpose of managing employee information is to organize the employees’ personal information and working tasks’ information. The Mange Employees’ Information includes:

* Search employees’ projects and tasks information

Employees are allowed to check personal projects and tasks’ information.

* Mange personal information

Employees, administrators and team leaders are the users of this system. They can change their personal information such as address, email and login password.

* Add, Update, Delete employees’ information

Administrators have the right to add new employees.

Administrators have the right to update the existing employees’ information.

Administrators have the right to delete the existing employees.

#### Manage customer information

Suplus Technologies has been working with many customers, organizations and companies, which are very important clients to the company. Recording customers’ information could help Suplus to maintain products and track feedback from clients. Therefore, manage customers’ information is a significant function in the whole system. This function could build a safe and steady customer information database, and it is easy for administrators to search for customers’ information.

* Search for customers’ information

Administrators have the right to search for customers’ information.

* Add, Update, and Delete customers’ information.

Administrators have the right to add new customers.

Administrators have the right to update the existing customers’ information.

Administrators have the right to delete the existing customers.

#### Manage project information

Since Suplus has been providing technical support to many customers, it requires a functional management method to organize these projects.

* Create new project  
  Administrators have the right to create new projects.
* Divide project into tasks  
  Team leaders are required to divide each project into many small tasks.
* Assign tasks to employees

When tasks are created, team leaders are required to assign the tasks to employees.

#### Login

The system includes three different users: employees, administrators and team leaders.

* Login to the system
* Log out from the system

#### Communicate

Login users (employees, administrators and team leaders) could share information.

* Post messages

Users could post messages on communication page.

* Post picture

Users are allowed to post pictures on communication page.

* Comment

Users could comment on any messages or pictures.

### 7.2 Use Case Diagram

The following figure presents the use case diagram, which helps to understand how the system functions work.

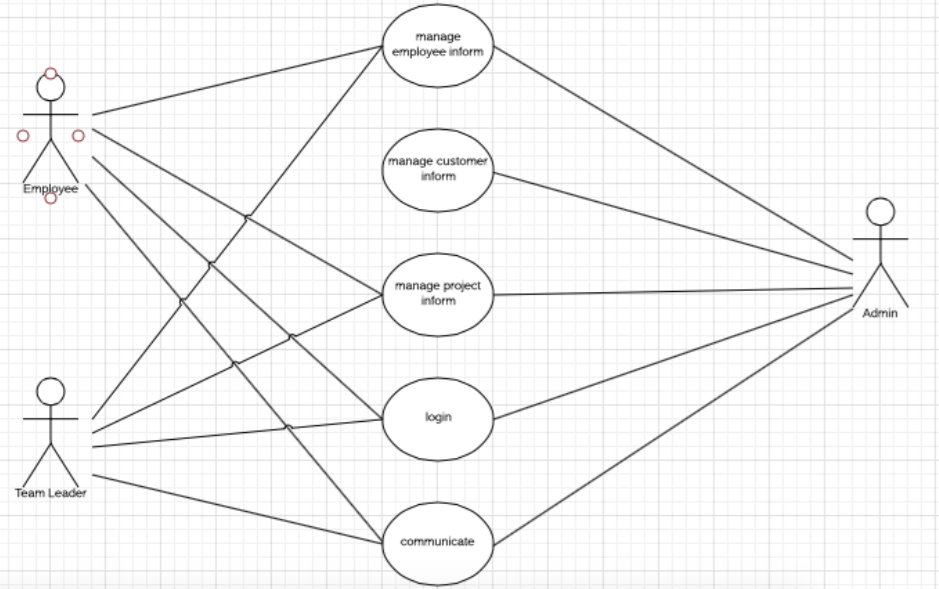


Figure 2 Use Case Diagram

Use Case Description

**Actor: Employee**

Use cases:

1. Login: It ensures the existing employee could login the system by entering the valid username and password.
2. Manage project information: Employee could check personal projects and tasks information.
3. Manage employee’s information: Employee could change his or her personal information such as password, address, phone number, and email address.
4. Communication: User who logins as employee has the right to post messages, share pictures, and comment on other users on the communication page.

**Actor: Administrator**

Use cases:

1. Login: When an administrator inputs the valid username and password, the system should display administrator home page.
2. Manage employees’ information: Administrator could add new employees, search for information of the existing employees, edit and delete information of the existing employees. Administrator could also change his or her personal information such as password, address, phone number, and email address.
3. Mange customers’ information: Administrator is able to create new customers, search for information of the existing customers, edit and delete information of the existing customers.
4. Mange projects’ information: Administrator is able to create new projects, search for information of the existing projects, edit and delete the information of the existing projects.
5. Communication: User who logins as administrator is able to post messages, share pictures, and comment on other users on the communication page.

**Actor: Team leader**

Use cases:

1. Login: The existing team leader could login the system by entering the valid username and password.
2. Maintain project: Team leader could create tasks to the currently project, search for information of the existing projects and tasks, edit and delete information of the existing tasks.
3. Manage employee’s information: Team leader could change his or her personal information such as password, address, phone number, and email address.
4. Communication: User who logins as team leader has the right to post messages, share pictures, and comment on other users on the communication page.

### 7.3 Domain Model Diagram

The following domain model diagram demonstrates the association between connectional classes in the Suplus Office Management System.

There are 5 classes: employee, admin, team leader, project and task. When the administrator creates a new project, the team leader divides the new project into tasks and passes these tasks to employees. The user as employee has the right to check personal projects and tasks’ information.

In addition, the following figure shows the relationship among these classes. When a customer signs up a contract with the company, it means there is a new project. In this diagram, the project is an important and central class, and it has a sub child class—task. Meanwhile, there are three other classes for three different roles in the diagram, which are employee, administrator and team leader.

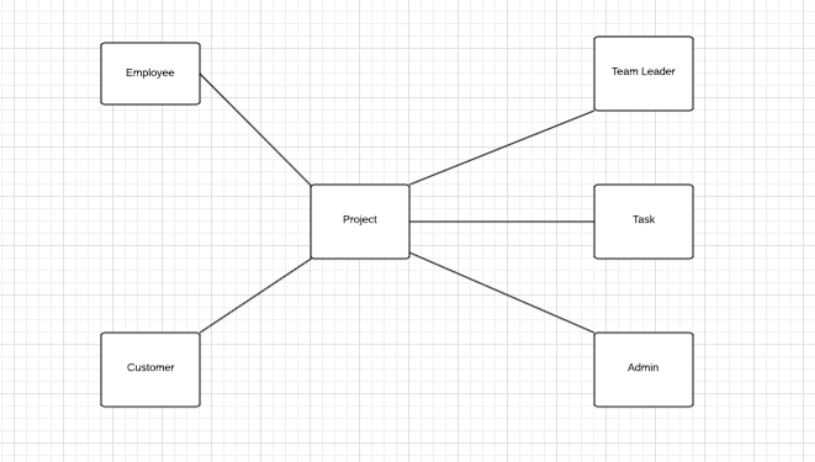


Figure 3 Domain Model Diagram

### 7.4 System Sequence Diagram

The following figures are sequence diagrams based on use cases diagram.

The first one is sequence diagram for login use case. This diagram represents the login use case for all the users (employees, administrators, and team leaders).

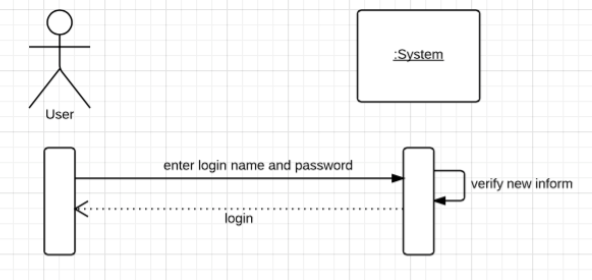
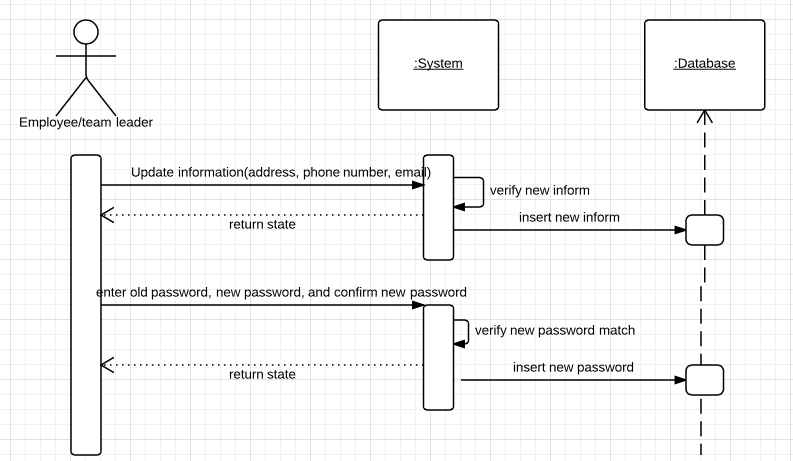
****

Figure 4 Login Sequence Diagram

The following is another system sequence diagram, which shows how employees and team leaders can perform similar functions in managing employees’ information.

Figure 5 presents the sequence diagram on how employees and team leaders can manage employees’ information

The following diagram is the sequence diagram for administrators to manage employees’ information. Administrators have the right to update personal information such as email and address, but the most important function is to maintain employees’ information, for example, creating new employees, updating employees’ information, and deleting employees.

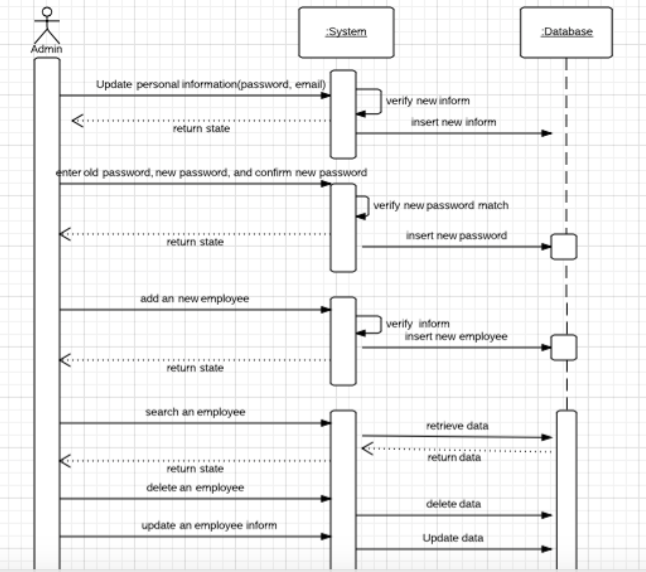


Figure 6 shows the sequence diagram on how administrators manage employees’ information.

The following figure is the system sequence diagram, which demonstrates how administrators manage customers’ information. Administrators could create new customers. They could also update and delete information of the existing customers.

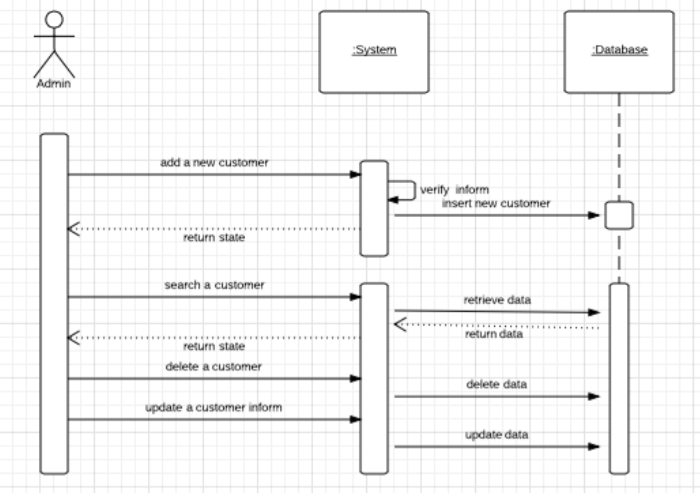


Figure 7 shows the sequence diagram on how administrators manage customers’ information.

Below is the system sequence diagram on how employees manage projects. Basically, employees have the right to search for projects and tasks’ information. When employees search for projects or tasks’ information, the webpage should show the searched projects or tasks’ information.



Figure 8 shows the sequence diagram on how employees manage projects.

The following figure is the system sequence diagram that shows how administrators manage project. Administrators have the right to create new projects, search for the detail of projects, update information of the existing projects and delete projects.

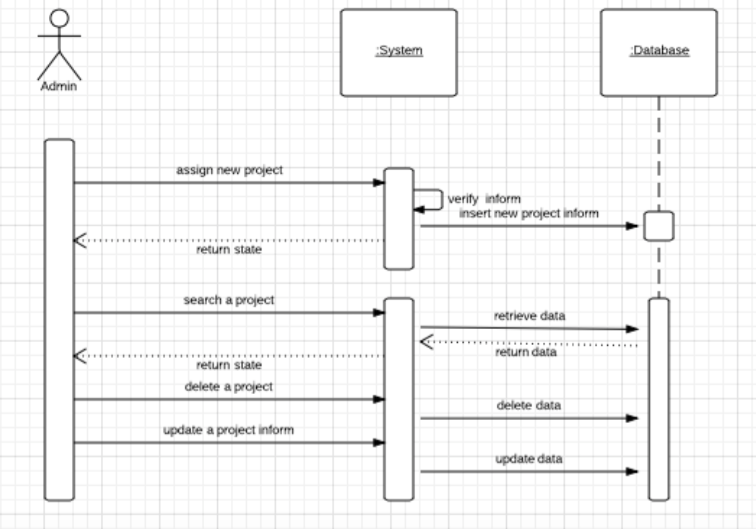


Figure 9 shows the sequence diagram on how administrators manage projects.

The next one is the system sequence diagram on how team leaders manage projects. As mentioned above, team leaders could divide a project into several small tasks and assign tasks to employees.

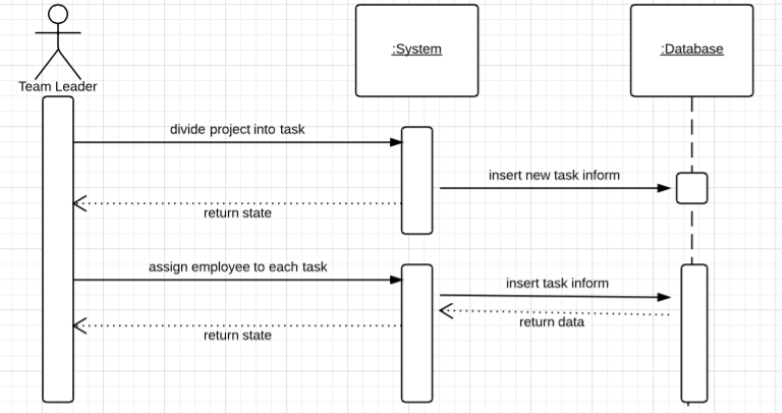


Figure 10 shows team leader manage projects sequence diagram

The last system sequence diagram presents the communicating function for all users in this system. All users could post messages, post pictures and comment on any messages or pictures.

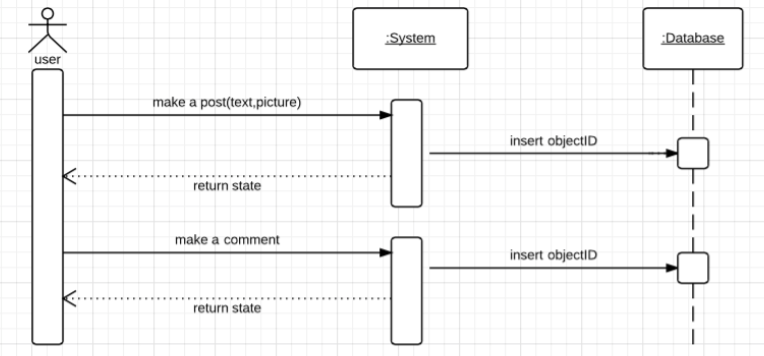


Figure 11 shows communication for the entire users sequence diagram

### 7.5 Software Design Class Diagram

The following diagram is the design class diagram of Suplus Office Management System. The system design class diagram is based on domain module class diagram. Generally, each project contains many tasks, so employees would be working on the project and tasks at the same time. Administrators could manage information of the projects, employees and customers. Team leaders divide the project into tasks and pass them to employees.

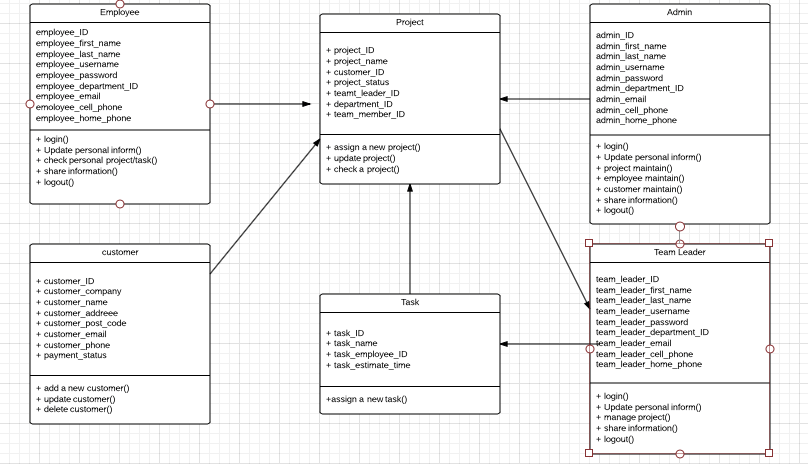


Figure 12 Software Design Class Diagram

### 7.6 Database Design

In this project, two kinds of database have been adapted; they are MySQL server database and NoSQL Mongo DB. MySQL server database helps to manage employees, customers, projects, tasks, and departments’ information. Mongo DB stores the data coming from the communication page.

### MySQL

MySQL is an open source database system, and it is inexpensive for developers. Suplus Office Management System is a PHP based web system, and it requires compatibility of the chosen database. MySQL supports cross platforms, and it could run on LINUX and Windows System. Consider the conditions mentioned above, this system adapted MySQL to manage employees, customers, projects, tasks and departments’ information.

The following diagram is the database relationship diagram of Suplus Office Management System. Database tables include employee table, customer table, project table, task table and department table. Employee table includes the personal information of the whole employees, team leaders and administrators. Customer table has customers’ information. The project table includes the whole projects’ information. Since a project could be divided into small tasks, task is a child table of a project. Department table includes the departments’ information.

The design of the database is based on domain class diagram, because the classes shown in domain class diagram are the significant objects. Suplus Office Management System includes five tables, which are employee, customer, project, task and department.

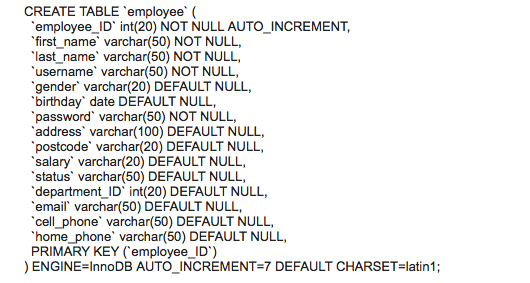
Figure 13 Database E-R Diagram

#### Employee table

Employee table keeps Suplus employees’ information including employees, administrators, and team leaders. Employee table contains:

|  |  |  |
| --- | --- | --- |
| Name | Data type | Description |
| First\_Name | varchar | Employee first Name |
| Last\_Name | varchar | Employee last Name |
| Employee\_ID | int | Primary key |
| Gender | varchar | Employee’s gender |
| Birthday | date | Employee’s birthday |
| Address | varchar | Employee’s address |
| Postcode | varchar | Postal code |
| Email | varchar | Email address |
| Username | varchar | Login username |
| Password | varchar | Log in password |
| Department\_ID | int | Foreign key |
| Salary | varchar | Employee’s salary |
| Status | Varchar | Marital Status: Single, Married, Divorced |
| Cell\_phone | varchar | Cell phone number |
| Home\_phone | varchar | Home phone number |

Table 1

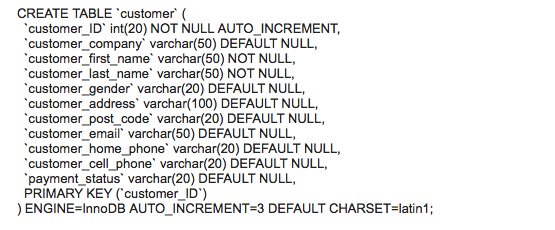


#### Customer table

Customer table includes customers’ information, which are:

|  |  |  |
| --- | --- | --- |
| Name | Data type | Description |
| Customer\_company | varchar | Customer’s company name |
| Customer\_fisrt\_name | varchar | Customer name |
| Customer\_last\_name | varchar | Customer name |
| Customer\_gender | varchar | Gender |
| Customer\_ID | int | Primary key |
| Customer\_ home\_phone | varchar | Phone number |
| Customer\_ cell\_phone | varchar | Phone number |
| Customer\_address | varchar | Customer’s address |
| Customer\_post\_code | varchar | Customer’s postal code |
| Customer\_email | Varchar | Customer’s email address |
| Payment\_status | varchar | Payment is paid by 3 instalments. When the project begins, 1/3 of the full payment should be paid; when the project is finished before test, another 1/3 should be paid; when the test is completed, the last instalment should be paid. |
| Project\_ID | int | Foreign key |

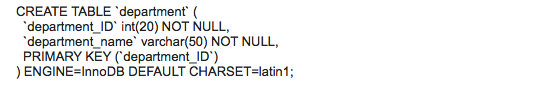
Table 2



#### Department table

Department table has the following information:

|  |  |  |
| --- | --- | --- |
| Name | Data type | Description |
| Department\_name | varchar | Name of department |
| Department\_ID | int | Primary key |

Table 3

#### Project table

Project table contains projects’ information.

|  |  |  |
| --- | --- | --- |
| Name | Data type | Description |
| Project\_name | varchar | Name of project |
| Project\_ID | int | Primary key |
| Customer\_ID | int | Foreign key |
| Project\_status | varchar | Beginning, testing, completion |
| Team\_leader\_ID | int | Team leader’s ID |
| Department\_ID | int | Foreign key |

Table 4

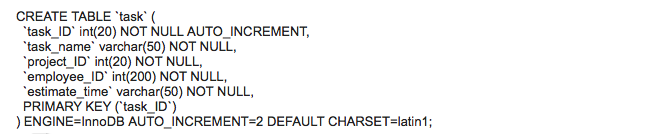
### 

#### Task table

Each project is divided into several tasks. Table task stores tasks’ information.

|  |  |  |
| --- | --- | --- |
| Name | Data type | Description |
| Task\_name | varchar | Name of task |
| Task\_ID | int | Primary key |
| Project\_ID | int | Foreign key |
| Employee­\_ID | int | ID of the employee who works on this task |
| Estimated\_time | Date | Working time calculated by hour |

Table 5

****

### NoSQL

As mentioned before, there is a social media page called “communication” in this system. For experiencing and exploring the new technology, this system adapted the Mongo DB stores data coming from the communication page.

Communication page allows users to post messages, pictures and to comment on other users’ posts. Each post is considered as a single document, and the post contains the information such as owner, post content, and comments. Since Mongo DB could store the data as documents, this system adapted the Mongo DB to store the posted data. If relational database were chosen to store the above information, it would require huge amount data mapping and take more time. Therefore, Mongo DB is preferred as it could fit in data structure without mapping data around and organizes each data as a collection. In other words, each posted data on the communication page includes ID, owner, postedBy, reference, image, create\_date, and update\_date. The following picture is an example of the posted data collection.

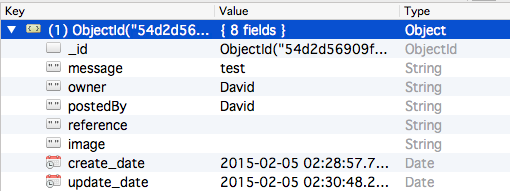


Figure 14

#### Collection

Collection is the data has been stored in Mongo DB:

ID: Same as relational database - when user saves new data, Mongo DB automatically attaches a unique identifier to it.

Owner: This is the person who originally posts the message/picture.

PostedBy: This is person who made a comment. If the post data is the original post, and nobody comments it yet, the postedBy could be as same as the owner.

Reference: When user comments on a message/picture, the reference will be changed to the original posted data’s ID.

Image: The image has been inserted.

Created\_date: Date and time of the original post created.

Updated\_date: Date and time of the comment created.

In the following part, screenshots of the user testing could explain more of the data design.

#### Post message

In this case, when a user logins as David, and David posts a message, “Hello!”

Because this is the original post without any comments, the owner and the postedBy are both David, and the created date and updated date would be the same.



Figure 15

The following picture presents the inserted collection in Mongo DB.

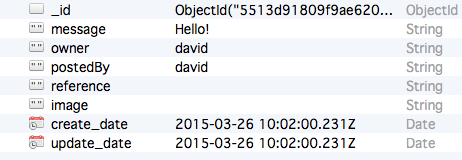


Figure 16

#### Post picture

The following figure is an example of the post picture function on communication page.



Figure 17

In this case, user David posts a picture on communication page. The owner and postedBy is still David, because he is the person posting the picture and this is an original message. The following picture shows that the image has been inserted into Mongo DB after it is posted on the communication page. In this case, an image has been inserted.

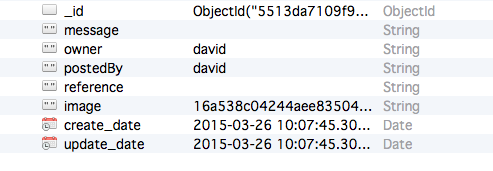


Figure 18

#### Comment

In this case, user David posts a message, “Good night, everyone!” and user Fitz writes a comment “Good night! ” below David’s message. 

Figure 19

The next picture demonstrates the original collection in Mongo DB. David is the owner of the message. The updated\_date is 05:45:55.568Z.

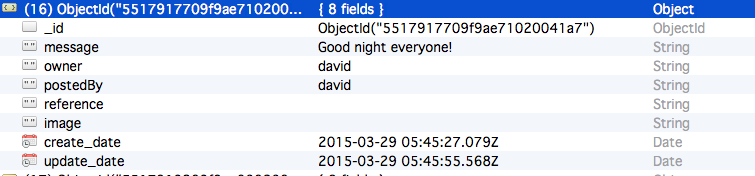


Figure 20

The following picture is Fitz comments’ collection in Mongo DB. As owner is the user who posts the original message, in this case it is David who posted the “Good night, everyone!”

PostedBy is changed from David to Fitz, because Fitz commented on David’s message.

Updated\_date is 05:45:55.566z (figure 21).

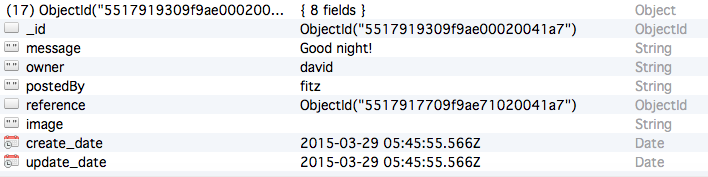
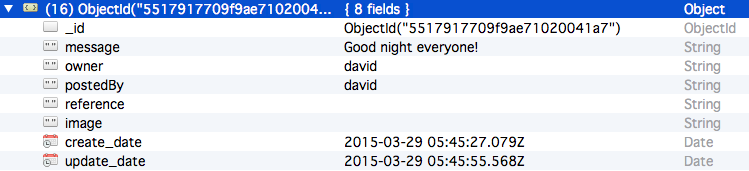


Figure 21

Now let me move back to David’s original post collection. After Fitz made a comment on David’s message, updated\_date of David’s collction has been changed to the time when Fitz made the comment, which is the same as figure 21: created\_time 05:45:55.566z.

Figure 22

For more detail of NoSQL and Mongo DB, please check 11 Research.

## Access Control

Since Suplus Office Management System is a web-side system, security problems should be considered an important part. For example, when administrators login, there are a lot of customer information, which are important assets to the company. The system also contains the employee information and project information, and unauthorized users may access the valuable information. In order to protect information, the system applies role based access control for users to access specific functions and modules.

### Role based access control

In this system, the users could be grouped into three different roles, which are employees, administrators, and team leaders. And these roles have different functions. A role could help to simplify database maintenance by giving an easy way to assign a set of privileges to each grouped users, and ensure the users to perform the particular actions in the system. Therefore, the system adapted the role based access control.

Generally, users are not assigned permissions directly, but their roles could help them identify themselves and allow them to do certain duties. Apart from the all users functions, the system authorizes different functions to different roles.

For example, administrators have permission to manage all the information of employees, customers and projects. Users who login as employees are allowed to check personal projects and tasks’ information. Users who login as team leaders are allowed to manage projects.

Users are required to enter username and password, which should match the stored information in the database. If users try to login with empty username and password, the system will give the users a warning message. When users login the system, they need to enter correct username and password. Based on the role access control, the user’s interface demonstrates different functions. In short, role based control access could protect Suplus Office Management System and enhance its security.

## Innovation

As mentioned in system function design, the system includes a social media page called “communication”. In my design, communication page allows users to post messages, pictures, and give comments on the posted information.

This social media page is a PHP based web page and the data has been stored in Mongo DB. In order to improve the design of user interface, Bootstrap is used. For the detail of NoSQL and Mongo DB, please check on 11 Research.

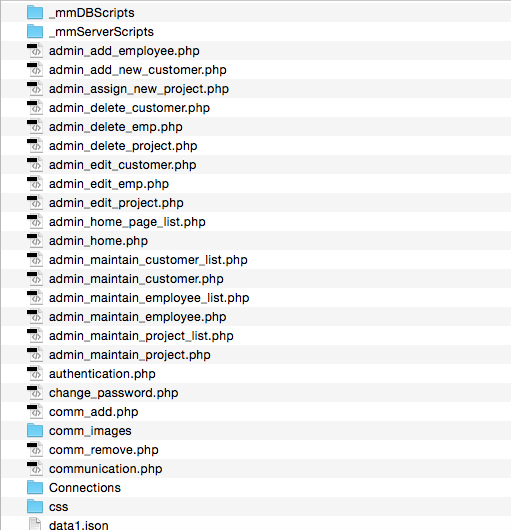
## Development and Implementation

### 10.1 Programming Languages Chosen

* PHP
* HTML
* JSP
* jQuery
* CSS
* Bootstrap

### 10.2 System Perspective

The next picture is system design tree map, which includes all web pages that have been designed in the system.

****

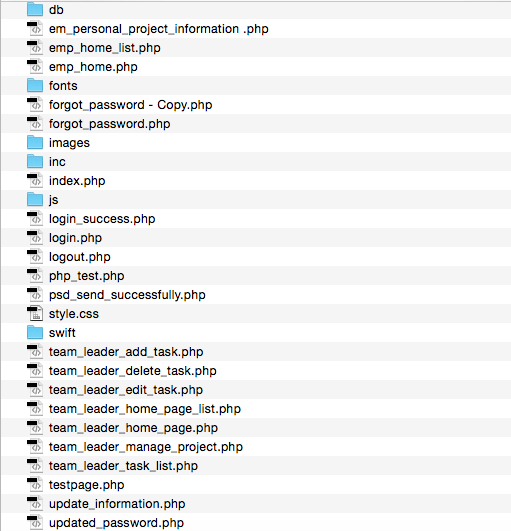
****

Figure 23

#### Employee module

1. Login

When the existing user enters valid username and password, the employee would login in the system successfully. This is the login page of this manage system, and all users should start from the login page.

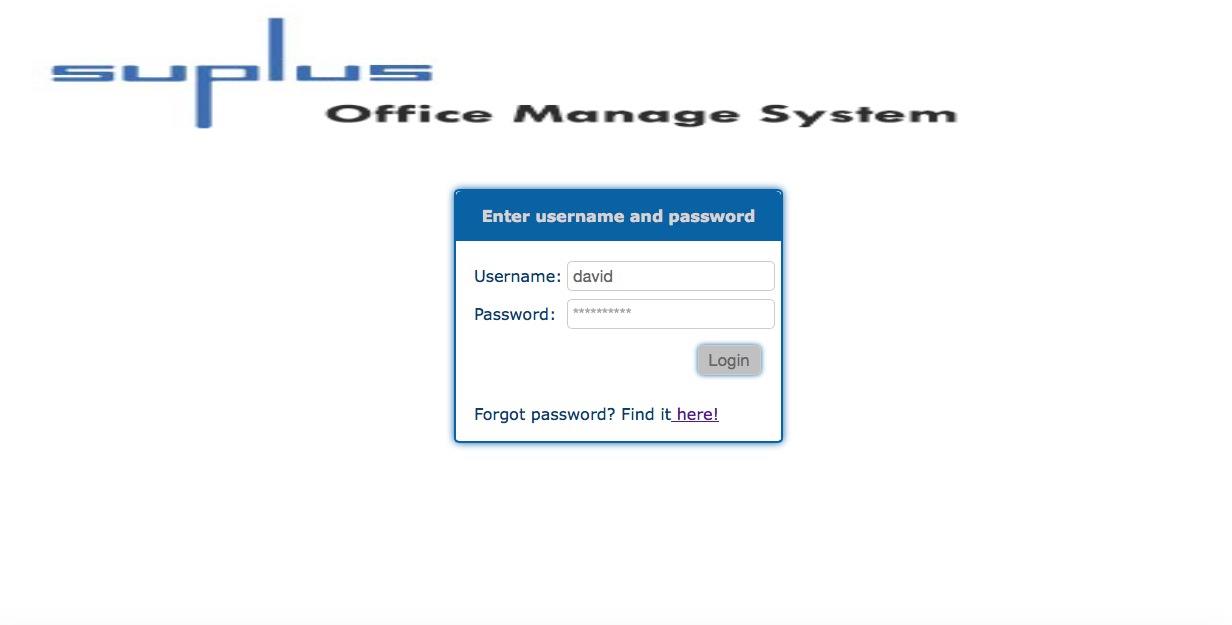


Figure 24

1. Employee home page

After the user logins successfully, the system should directly go to the home page of the employee, which presents the employee’s functions and lists all the tasks. It provides search function to help the employee to search for tasks or projects’ information.

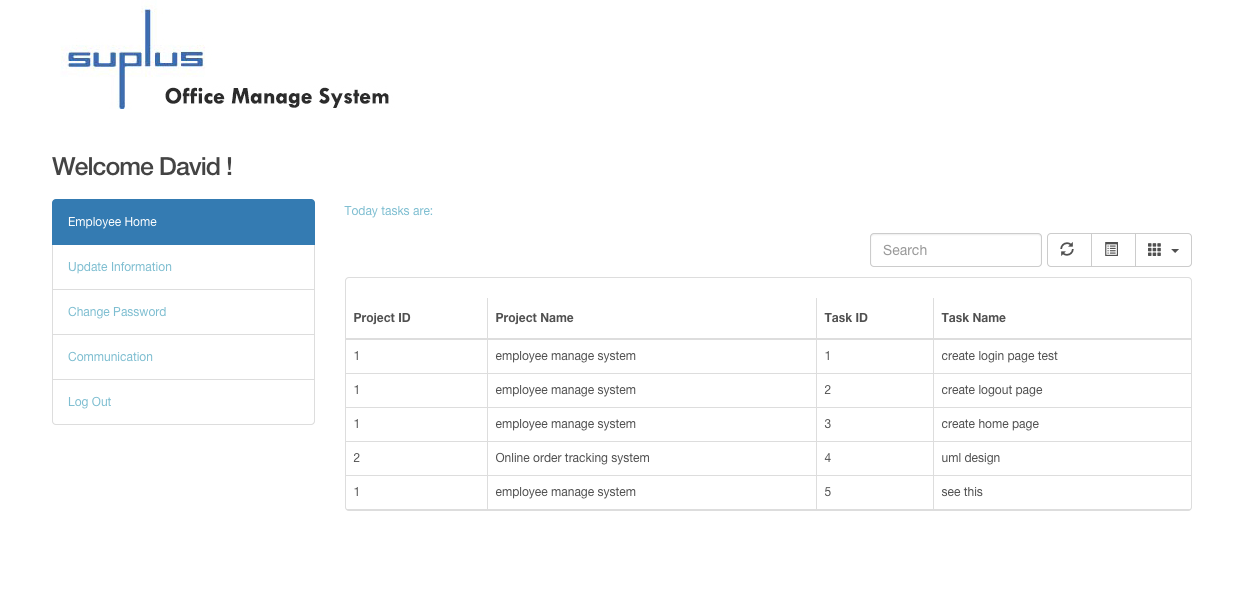


Figure 25

1. Update personal information

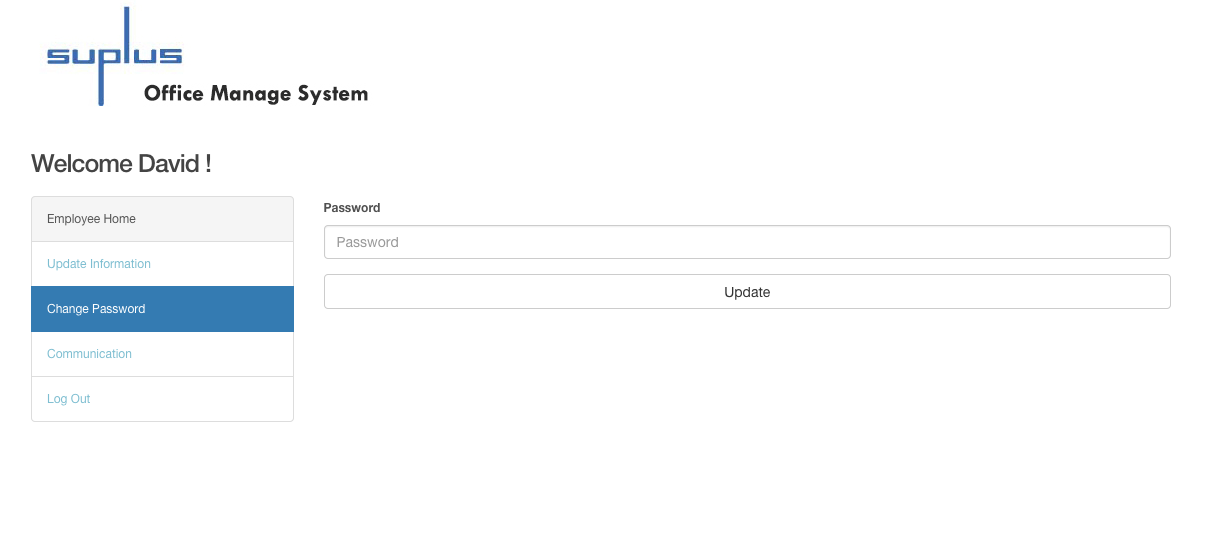
After clicking the Update Information function from navigation menu, the user could enter the information into the textboxes to update personal information.



Figure 26 shows employee update personal information

1. Change password

In addition, the user could change login password. After new password is entered and the update button is clicked, the login password would be updated.

Figure 27 shows change password page

1. Communication

On communication page, the user can post messages and share pictures by clicking post button. At the same time, the user could see other users’ posts and comment on any messages or pictures.

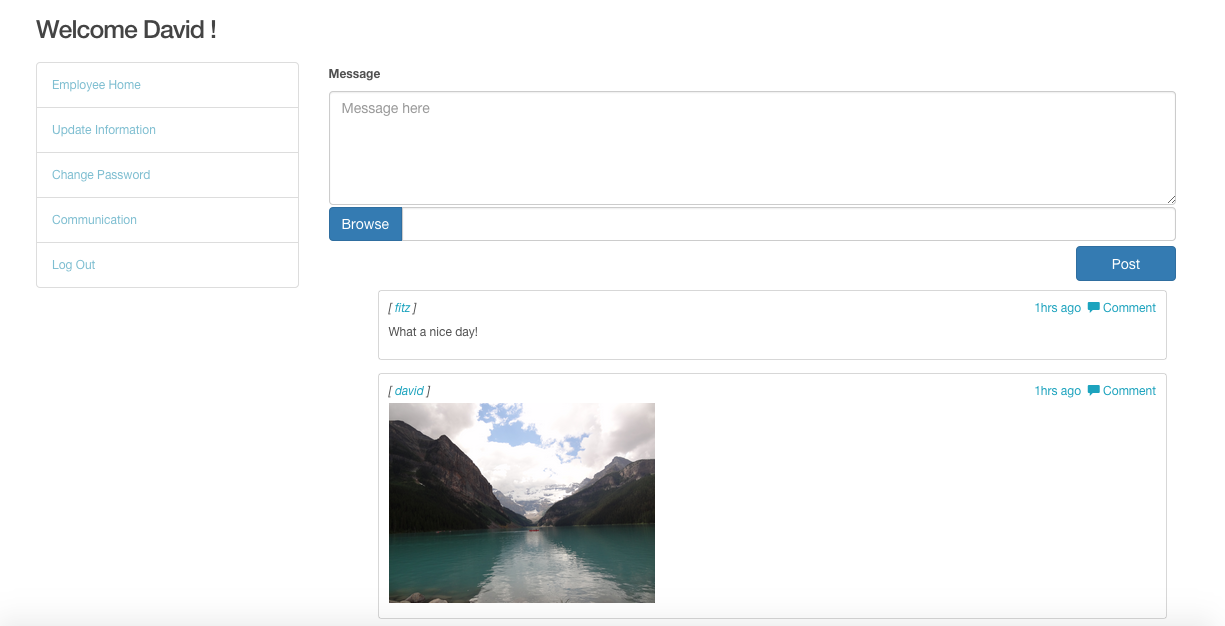


Figure 28 presents communication page

#### Administrator module

1. Home page

After the user logins as administrator, the system redirects the user to the administrator home page. The home page of administrator lists all the projects, team leaders and customers’ information. Moreover, the search menu on the top of this page allows the user to search for projects’ information.

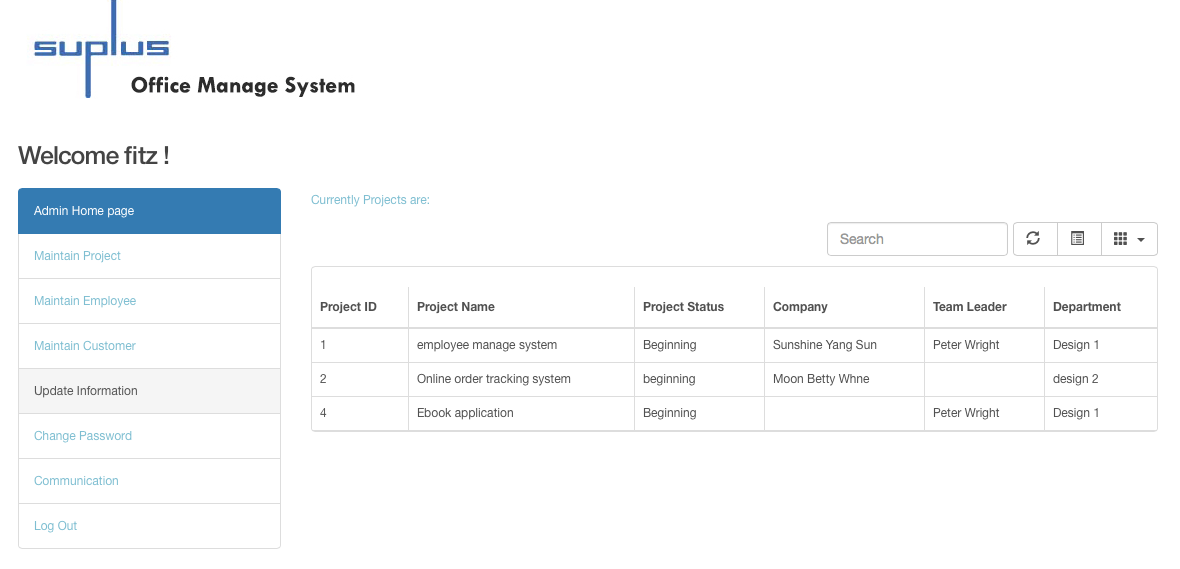
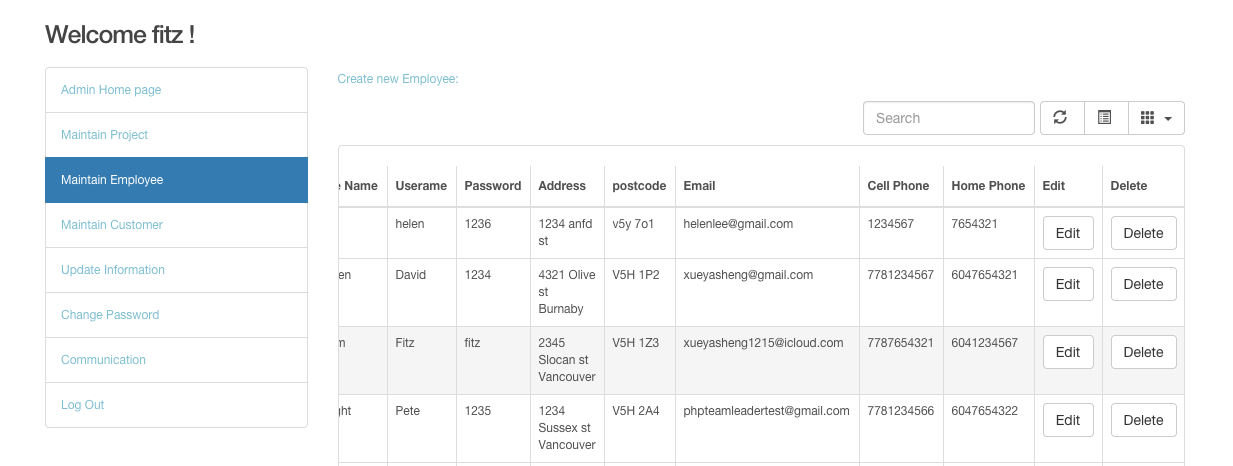


Figure 29 shows admin home page.

1. Maintain employee

Maintain employee function allows the user to search for a specific employee by entering the employee’s information, to create new employees by filling in the information list, to edit and delete the existing employees by clicking the edit and delete button.

Figure 30 presents maintain employee page.

The following picture shows create new employee page.

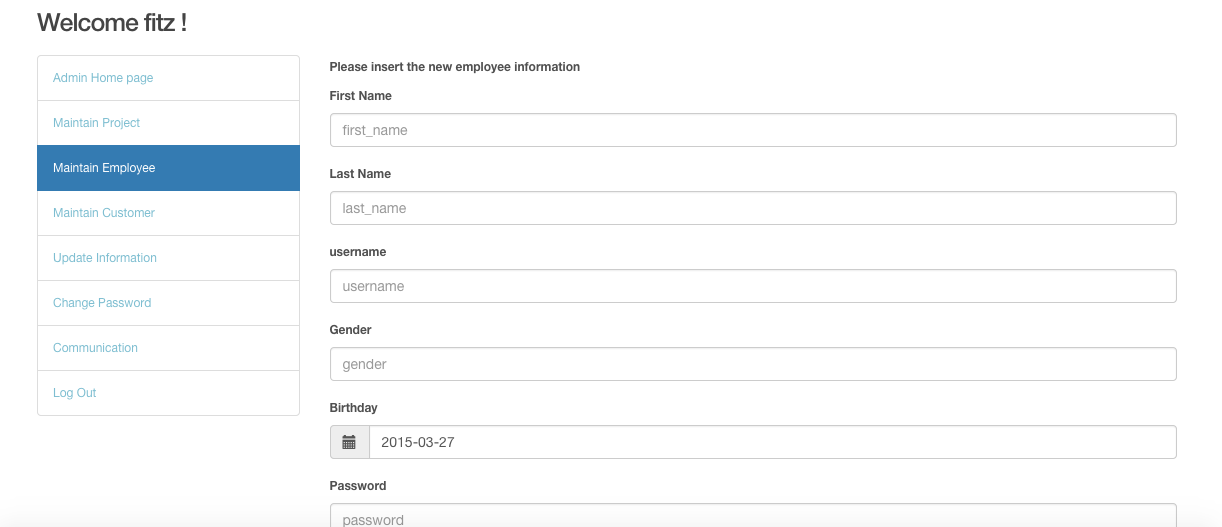


Figure 31

The next figure shows the code of administrator editing employees’ information.



Figure 32

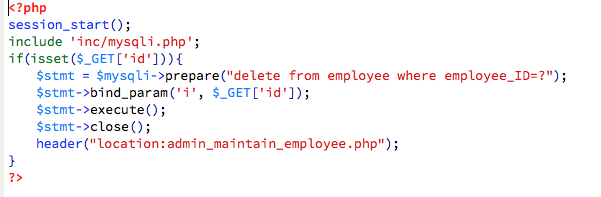
The following figure presents the code of administrator deleting employees’ information. 

Figure 33

1. Maintain Customer

Maintain customer function allows the user to search for customers’ information, to create customers by filling in the list, and to update the existing customers’ information. Administrator could also delete the existing customers.

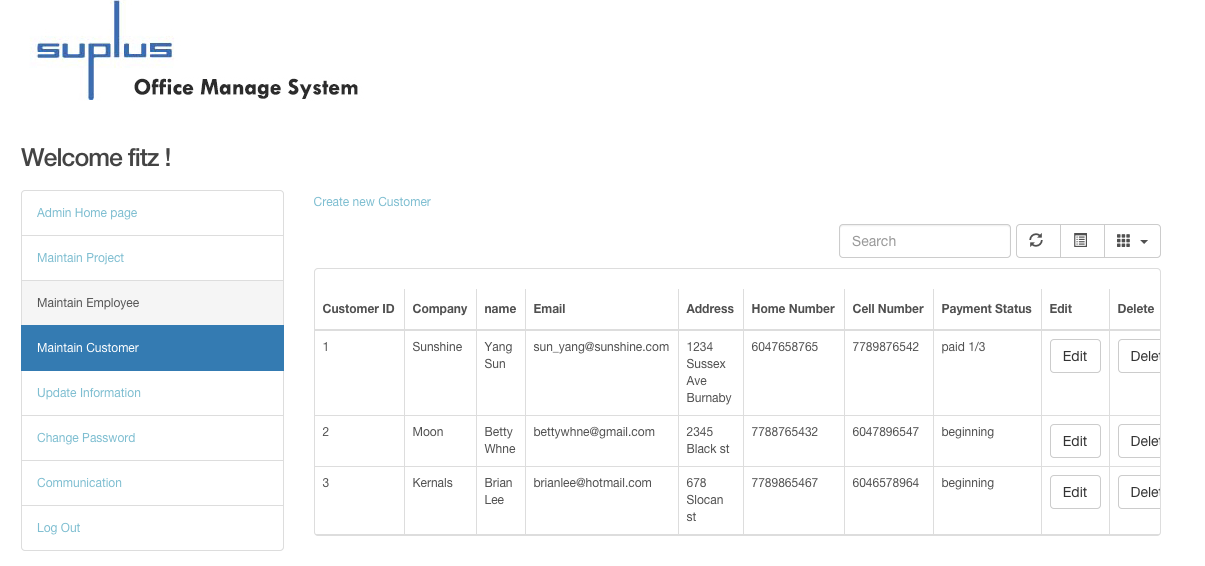


Figure 34 shows maintain customer page.

The next picture is the screenshot of creating new customer.

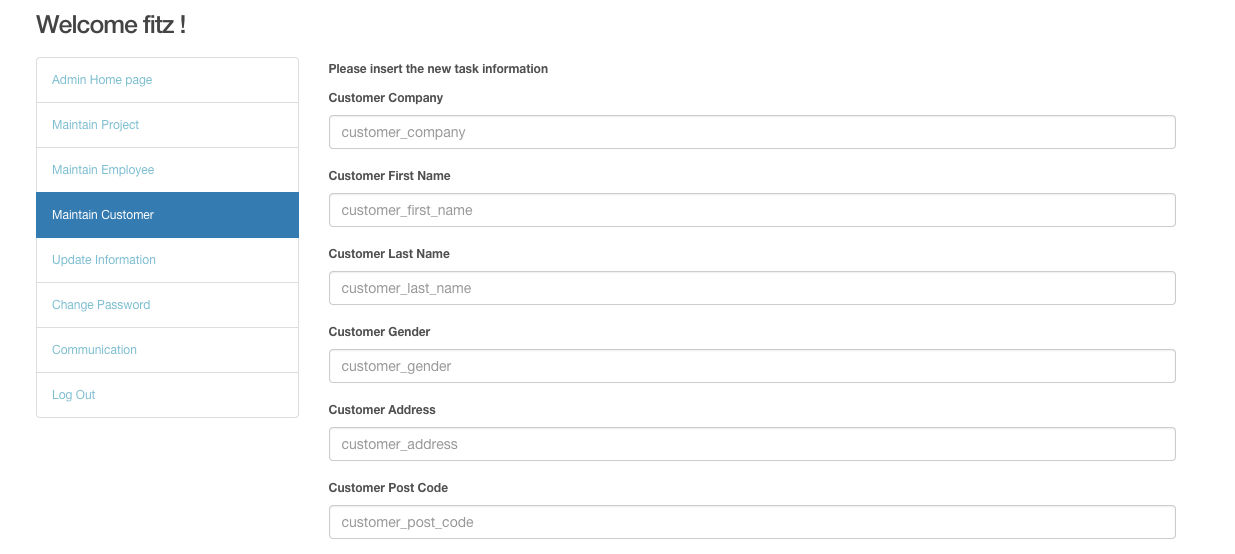


Figure35

The figure 36 presents the code of administrator deleting customers’ information.



Figure 36

The following figure shows the code of administrator editing customers’ information.

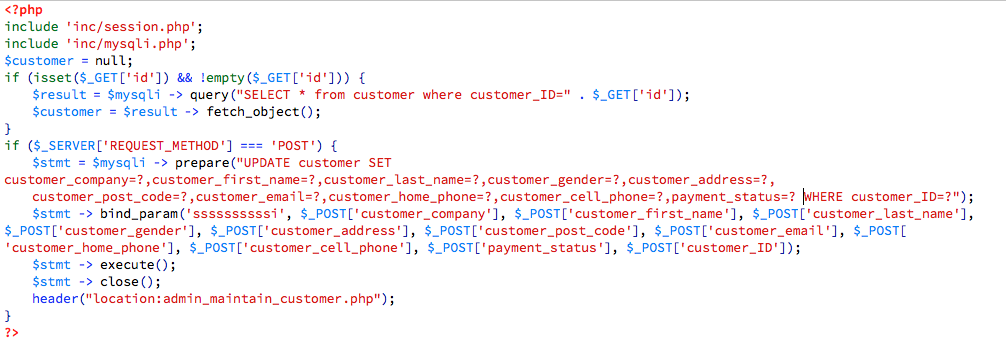


Figure 37

1. Maintain Project

Maintain Project function allows the user to search for projects’ information, to create new projects by filling in the list, and to update the existing projects’ information. Administrator could also delete the existing projects.

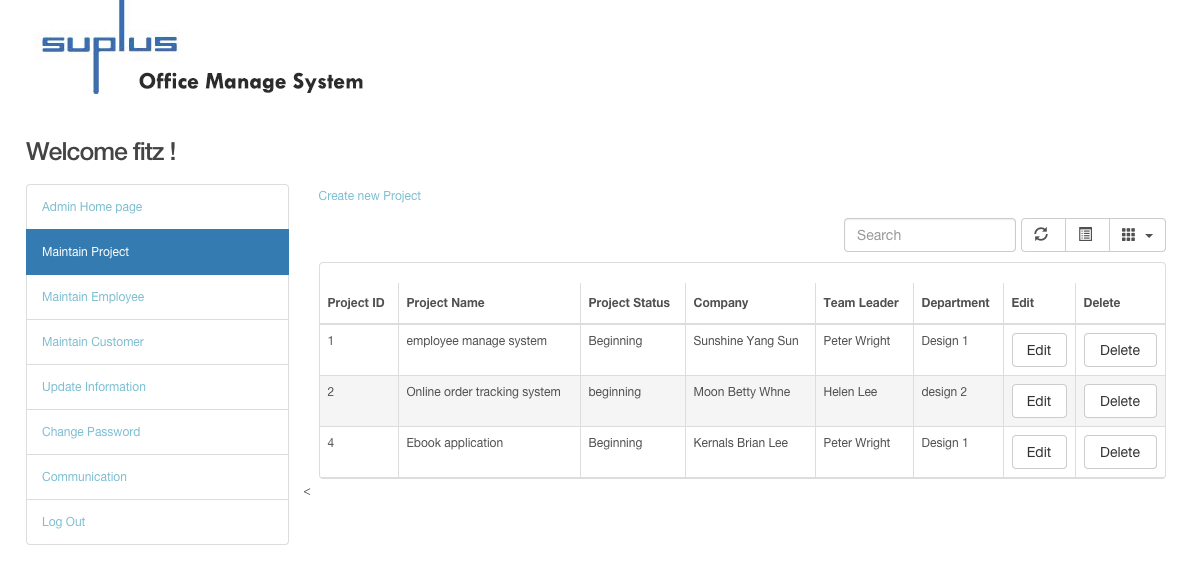


Figure 38 shows administrator maintaining the project page.

The next picture shows the page layout of the administrator creating new project.

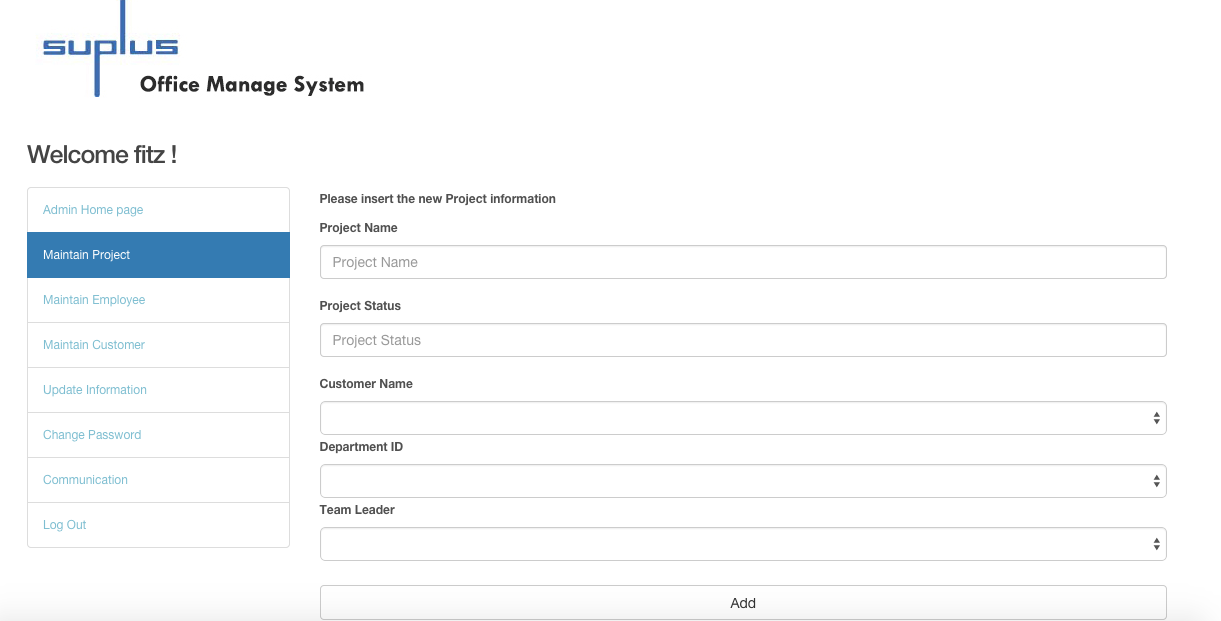


Figure 39

The next figure shows the code of administrator editing projects’ information.

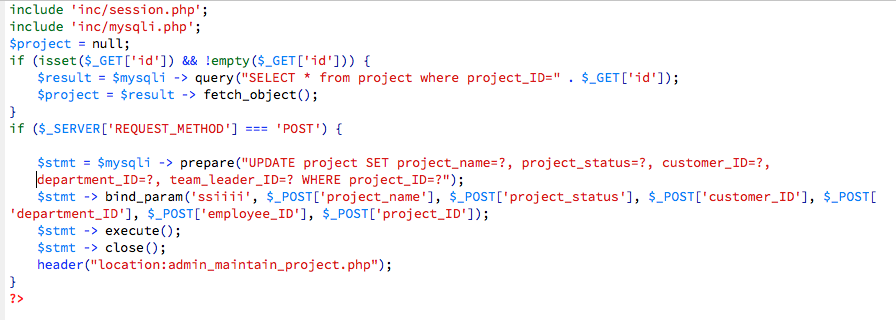


Figure 40

The following figure presents the code of administrator deleting projects’ information.



Figure 41

1. Update information

After clicking Update personal information, the user could update personal information such as address, phone number, and email address.

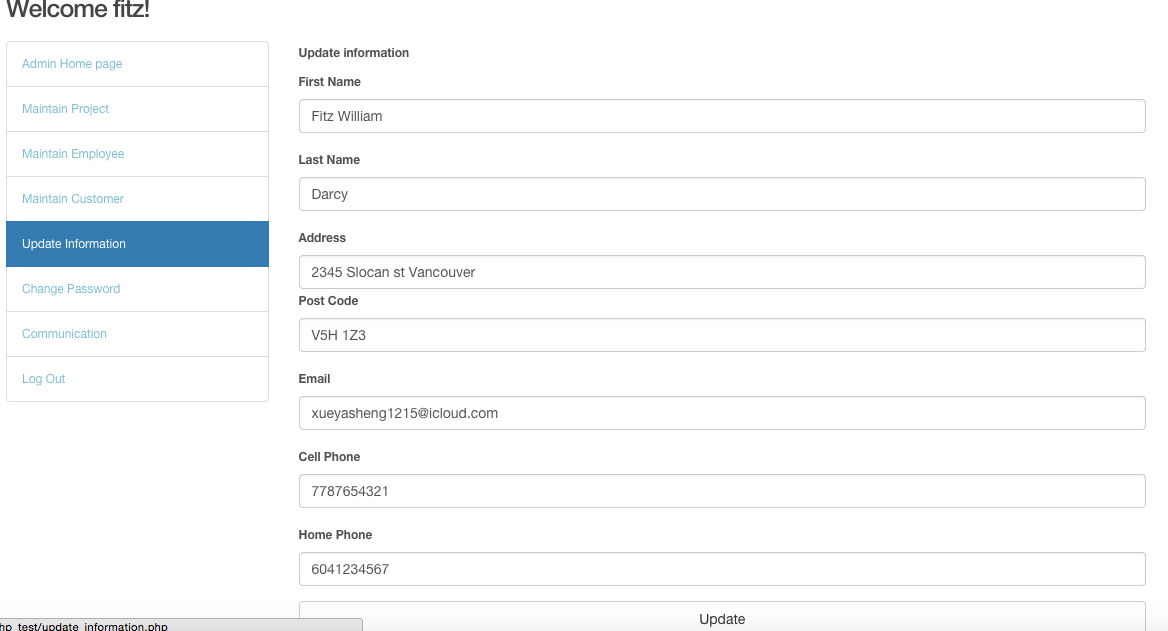


Figure 42

1. Change password

In addition, the user could change the login password.

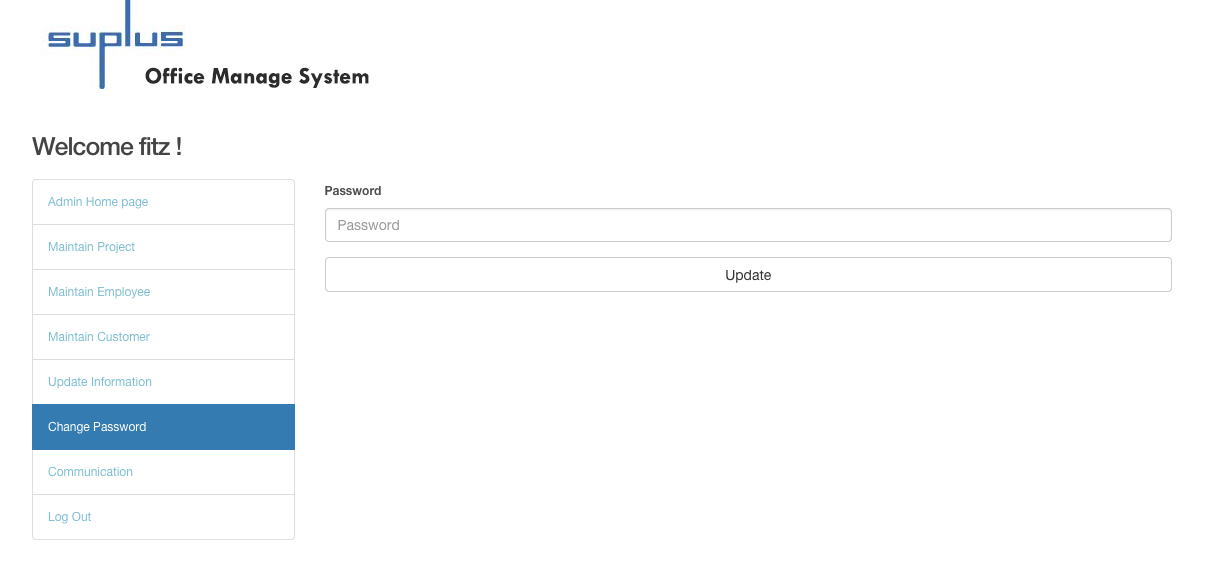


Figure 43 shows the page layout of change password

1. Communication

Administrator as the current user can post messages and share pictures by clicking the post button. At the same time, the user can see other posts and comment on other posts.

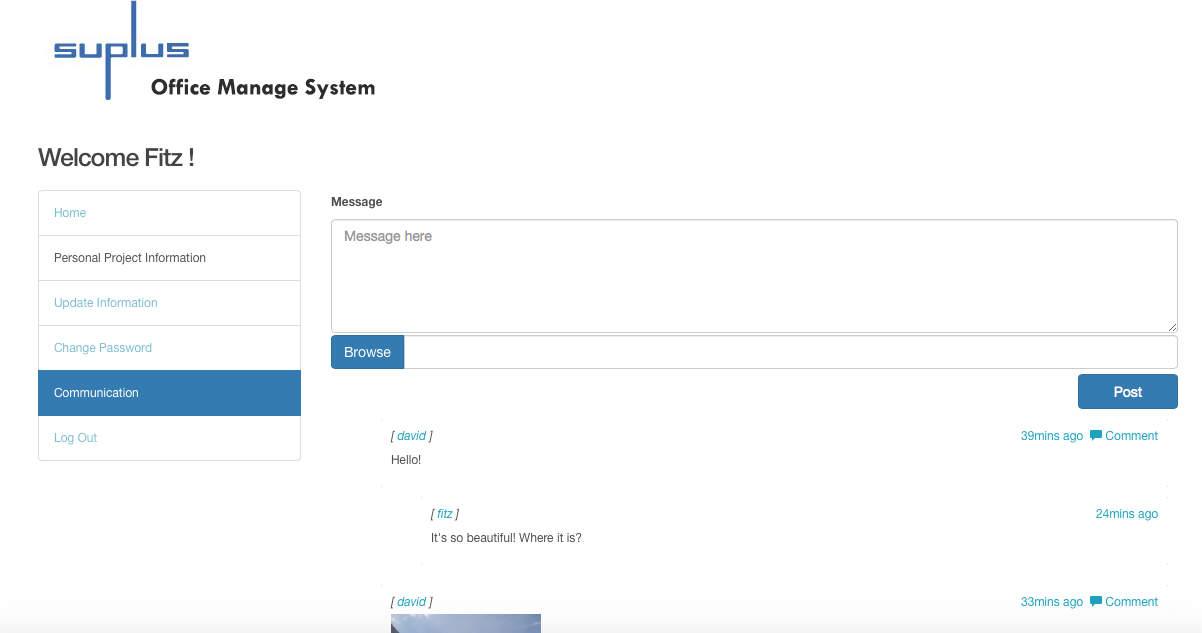


Figure 44

#### Team leader module

1. Home page

When the user logins as team leader, the system redirects the user to team leader home page. The home page lists all the currently projects and the customers’ information.

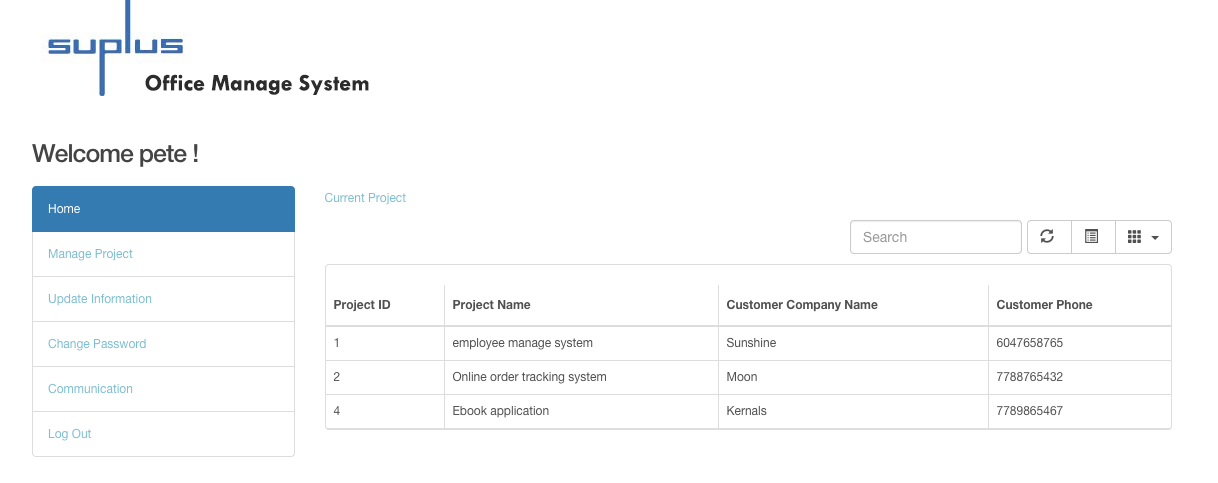


Figure 45 shows team leader home page.

1. Manage project

Manage project page lists all the tasks’ information. On the top of manage project page, there is a “insert a new task” link, which allows the user to insert new tasks to a selected project. The user could update the existing tasks’ information, and delete the existing tasks.

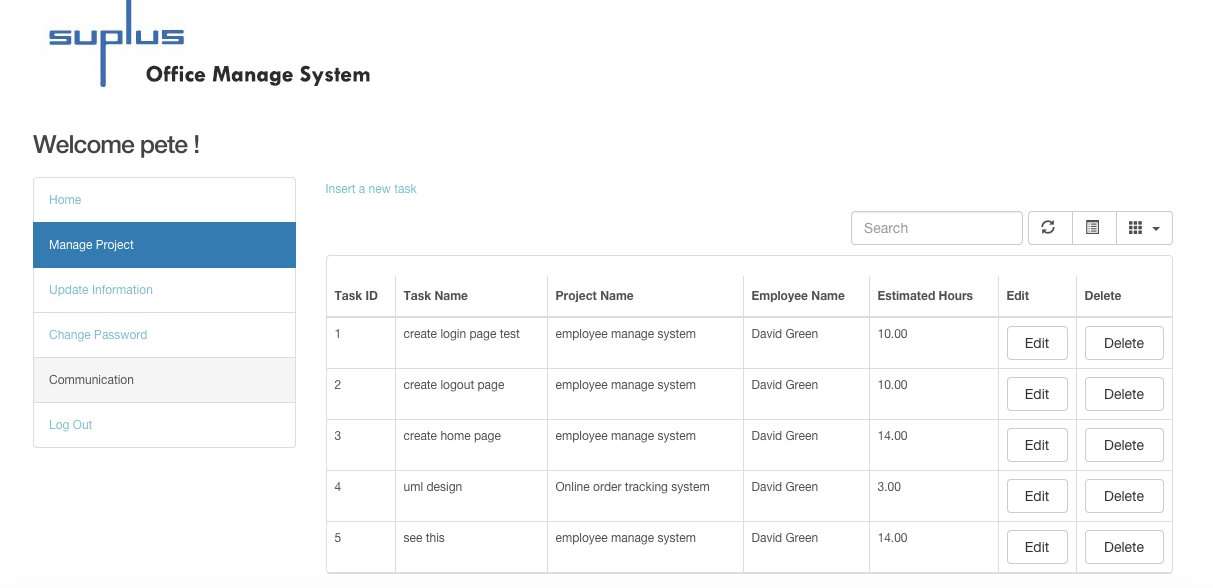


Figure 46

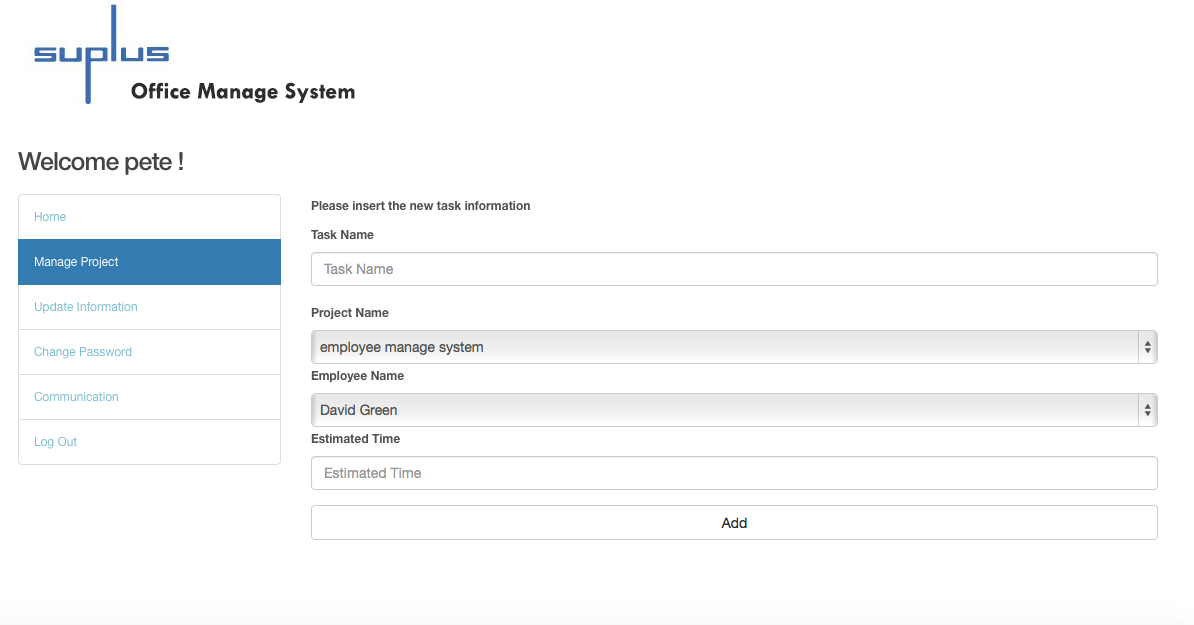
Next picture shows the page layout of a team leader insert new task.

Figure 47

The following picture shows the team leader editing the existing task’s information.

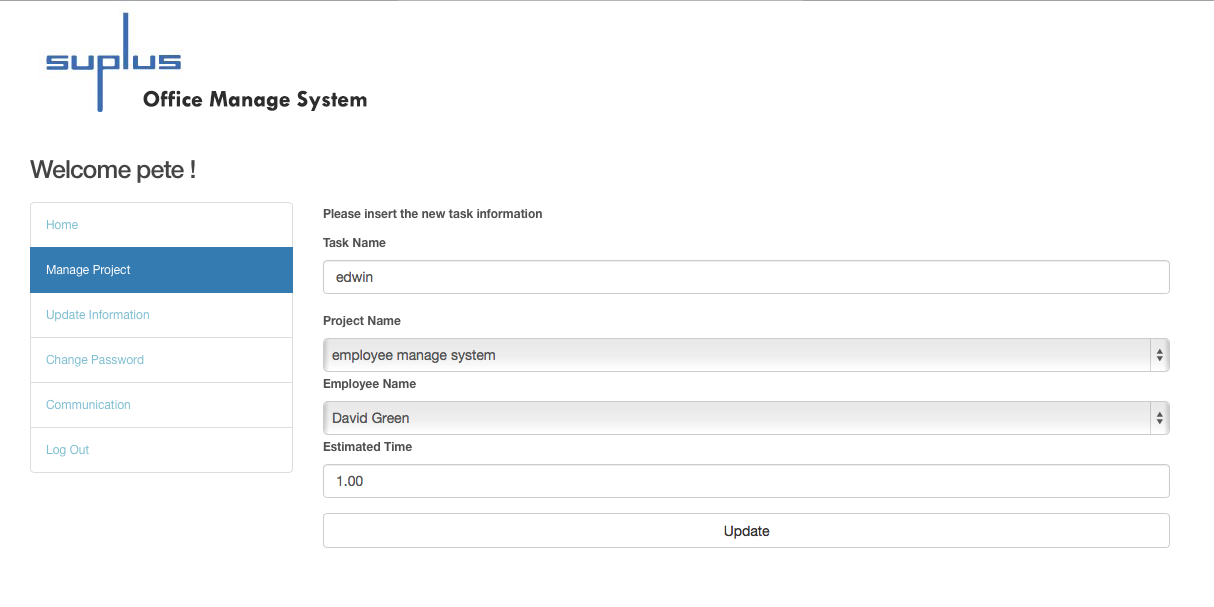


Figure 48

When the user deletes a task, there is a pop-up window asking, “Are you sure to delete the task?”

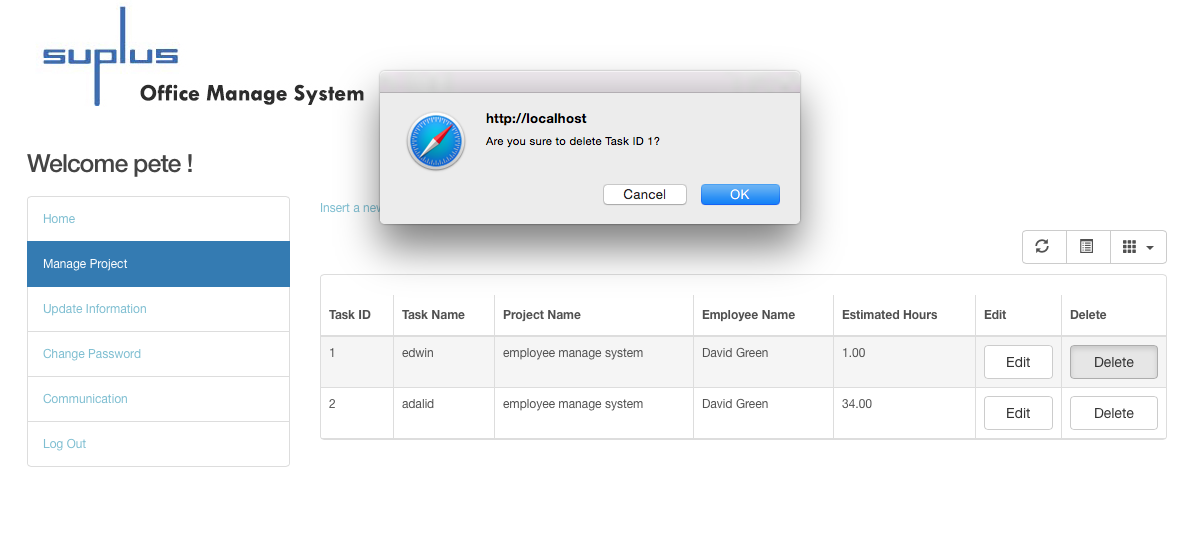


Figure 49

1. Update personal information

The user could update personal information such as address, phone number, and email address.

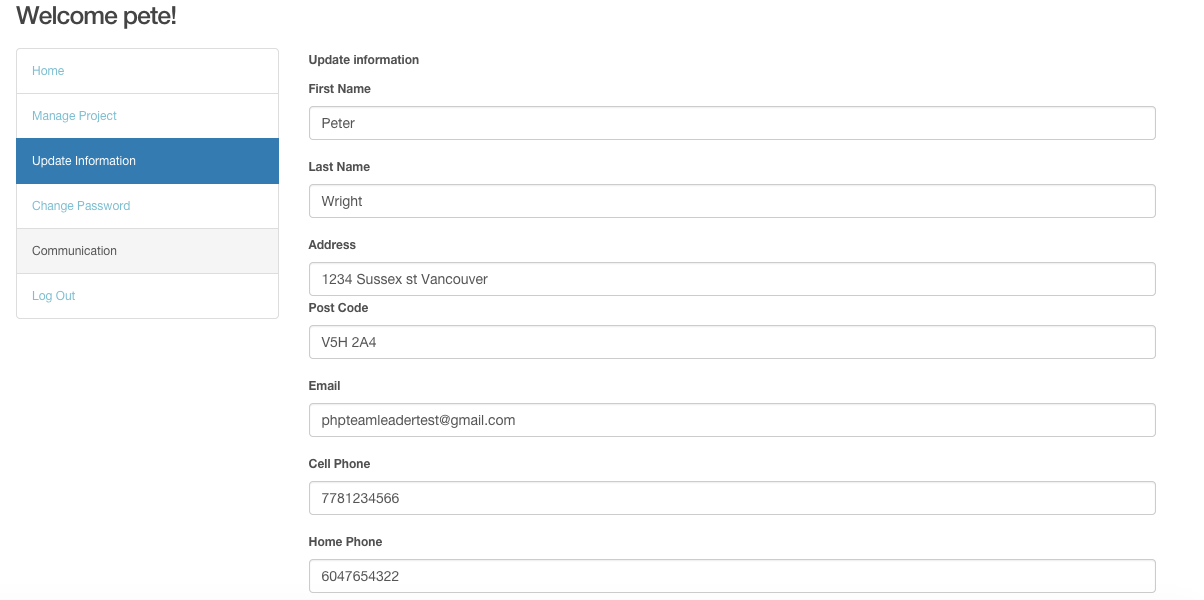


Figure 50 shows the page layout of an administrator updating personal information

In addition, the user could change the login password.



Figure 51 shows the page layout of change password

1. Communication

The user could post messages and share pictures by clicking the post button and comment on others’ posts.

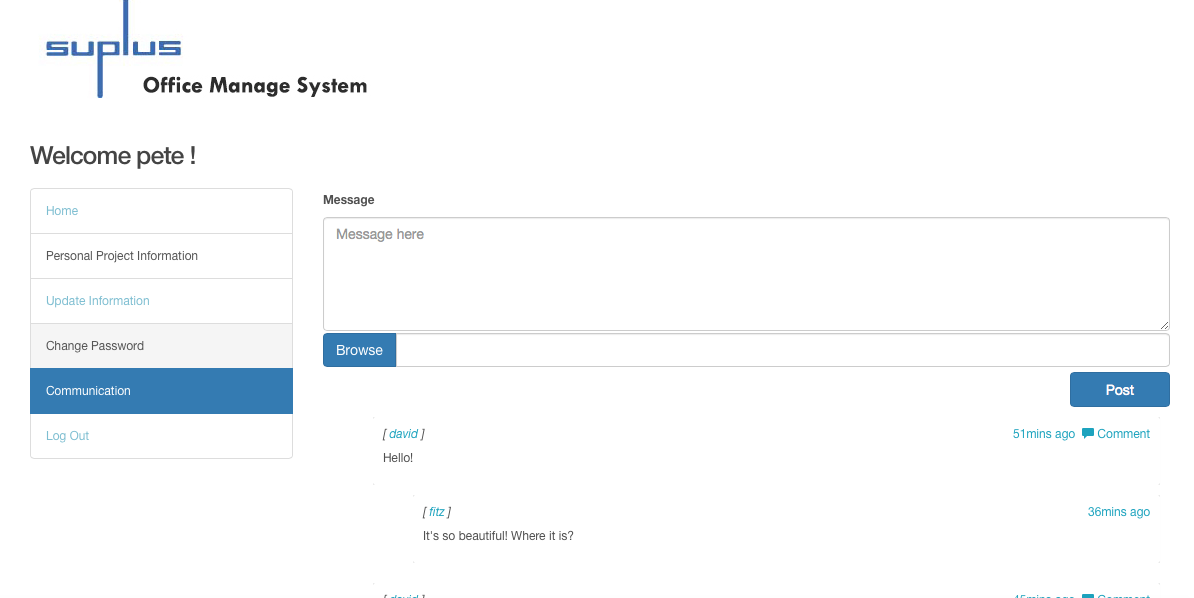


Figure 52

## Research

### 11.1 Why NoSQL?

To explore and experience the new technology, this system adapts the NoSQL. Currently, social media websites such as Facebook and Twitter are all conducting NoSQL, which plays a significant role in current database field. Suplus Office Management System adapts Mongo DB to collect the data from the communication page, the social media web page in this system. The reasons for choosing NoSQL are:

Loading Balance: NoSQL could run over multiple servers, balance the load, and duplicate data to keep the system running in case of hardware failure. As a social website with a huge amount of data, this system requires a steady database to operate the system.

File storage: NoSQL contains many different types of file storage, such as key-value stores, document database, wide-column stores, and graph database.

The large number of users combined with the dynamic nature of usage patterns increases the need for more easily scalable database technology. With relational technologies, it is difficult or even impossible to get the dynamic scalability and maintain the performance as users demand (Sadalage & Fowler, 2013). This is the most significant reason for choosing NoSQL when creating communication page as a social media web page.

### 11.2 Advantages & Disadvantages

Compared to NoSQL, relational database is a set of tables containing data that can be fitted into the existing categories. However, NoSQL usually handles unstructured and non-defined data. Since it is a very new and young technology, NoSQL still has a few challenges. The next section of the paper outlines the advantages and disadvantages of NoSQL.

**Advantages:**

Processing data is faster than relational database. Usually, relational database requires the same set of ACID (atomicity, consistency, insolation, durability) restraints, but it could decrease the processing speed. As NoSQL does not support ACID, it increases performances. Besides, NoSQL data model is simpler than relational database. These are the reasons that show NoSQL performs faster than relational database.

Document-based store: NoSQL database stores and organizes data as collections of documents. As mentioned before, users can customize documents rather than using structured and predefined information in relational database.

Sharding and load balancing: Sharding could store data records across multiple machines when the data requires a huge amount of growth in the future. NoSQL (especially MongoDB) could solve this problem with horizontal scaling and add more machines to support data growth (would be mentioned in the 11.3 Why Mongo DB?).

**Disadvantages:**

1. Complex consistency: Since NoSQL database does not follow the ACID transactions, it may have problem with consistency. Even though inconsistency could improve the performance and scalability, it could cause application and transaction problems, especially in banking management.

2. No joints: NoSQL database does not include joints as relational database does. When the data requires a joint function, it may make multiple quires to join the data manually. It may reduce performance.

3. Absence of standardization. Because NoSQL does not have standard APIs or query language, it could cost more when there is migration of a solution from different vendors (Leavitt, N, 2010).

Although NoSQL is not a mature technology, it allows data stored over different servers or clusters. It is perfect for social media page. Therefore, NoSQL technology provides a new environment to develop data and system.

### 11.3 Why Mongo DB?

The data storage model of NoSQL includes document database and graph database.

Graph database offers many advantages to users when operating graph data; however, it is not a good choice for communication page. Although the system users are able to post information including pictures, the users could post text messages as well. Therefore, this system could not use graph database.

For the part on exploring and practicing new technology and innovation, this system adapted Mongo DB. There are several important reasons to choose Mongo DB. Mongo DB is a better solution to this system because Mongo DB is a document-based database. Mongo DB stores and organizes data as collections of documents, which could provide better data model to fit in data structure than relational database. In other words, Mongo DB can save more time for mapping between data and data structure than relational database. The post data from communication page could be a simple message, a picture or even an article, which are all collections and are stored in Mongo DB. Therefore, using Mongo DB to store posted data is more efficient.

Another reason to choose Mongo DB is scalability. Communication page as a social media web page contains a lot of posting data sets. In this situation, the high query rates and large data sets could exceed the storage capacity. However, Mongo DB could provide vertical scaling and sharding to solve this problem.

Vertical scaling approach could add more CUP and storage resources to increase capacity, but it costs more money. Another approach, sharding, can divide the data sets and distributes the data to multiple servers or other shards. Shard is an independent database. When users insert data, application only requires accessing the shard responsible for that record (Francia, 2012). Shard could fix the server capacity problem and reduce the development budget. Therefore, in terms of the data type storage and scaling, Mongo DB is the best solution for this system.

## Test

I designed 33 tests based on the key system functions. During the testing, I had completed two categories of tests: user testing and system functional testing.

### 12.1 User Test Plan

At first, I tested the employee, administrator, and team leader’s functions based on the functional testing list I created. Secondly, I invited six testers in total. Three of them are general testers who are not the staff of Suplus Technology, and the other three testers are an employee, an administrator and a team leader at Suplus Technology. They were all required to follow the test cases I created and to provide feedback for debugging.

The testers are also required to fill in the survey list that could help to improve the system. For the detail of the survey list, please check Appendix C.

Testers were required to time the duration of the tested operation according to the test cases. If the testers have to take a long time to go through each module, it means that this system is hard to follow or understand, and the system needs to be fixed.

Scale: The functions are workable or not. The hours the testers spent to operate the system according to the functional testing list.

Test: The tests were conducted under the Windows and Linux System. The testers were invited to test the system functions of a single module (there are three modules: employee, administrator, and team leader). The testers were given the existing user name and password to login the system. The duration of operation was timed according to the functional testing list.

Test scenario: The testers were grouped to test different modules. Each module includes two testers. One of them is working for Suplus Technology, and the other one is a general tester who had been invited to do the test. The testers were required to login the system and went through all the functions.

Worst case: The testers find functions unworkable. The testers spend more than an hour to complete the entire operation.

Best case: All the functions are workable. The testers spend less than half an hour to complete the entire operation.

Planned: The testers take 20-40 minutes to finish the whole test. And the system functions are all workable.

#### Result of user testing

For the testers’ survey, please check Appendix C.

**A. All Users Testing**

All of the testers were required to test all the following functions after login in to the system. Half of the testers are working for Suplus Technology.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test NO. | Testing | Expected Outcomes | Result | | | | Tester |
|  | Test if information could be filled in the login textboxes on the login page. | Information could be entered in the textboxes of username and password. | **Passed** | | | | E  A  T  G |
|  | Test if an error message is present to indicate invalid data, or empty blank. | * Enter not existing username and password—fail to login to system. * Empty username and password—fail to login to system. | **Passed** | | | E  A  T  G | |
|  | Test if the existing user could login to system by entering the valid username and password. | The user could login to system. | **Passed** | | E  A  T  G | | |
|  | Test if the system could send email with password to binding email address when user clicks the forgot password button on login page. | An email with password could be sent. | **Passed** | E  A  T  G | | | |
|  | Test if the user could edit personal information on update information page. | The information could be updated. | **Passed** | E  A  T  G | | | |
|  | Test if the user could change password once signed in the system. | The user could enter the new password on change password page.  After logout, the user could login system with the new password. | **Passed** | E  A  T  G | | | |
|  | Test if the user could post message on communication page. | The user who logins with valid username and password could post message on communication page. | **Passed** | E  A  T  G | | | |
|  | Test if the user could post a picture on communication page. | The user could select a picture from local directory, and the user could post a picture. | **Passed** | E  A  T  G | | | |
|  | Test if the user could post comment on communication page. | The user could post a comment. | **Passed** | E  A  T  G | | | |

E is an employee working for Suplus Technology. A is the administrator of Suplus Technology, T is the team leader of Suplus. G is the tester invited to do the test.

**B. Employee Testing**

The testers were required to test all the following functions after login as employees. One of the testers is an employee of Suplus Technology.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test NO. | Testing | Excepted Outcomes | Result | Testers | |
|  | Test if the search textbox on employee home page allows the user to input the information. | The search textbox could allow the user to insert the information. | **Passed** | E  G | |
|  | Test the search function on employee home page. | The data could be found by filling in following search conditions: task ID, tasks’ names and project ID. | **Passed** | | E  G |

E is an employee working for Suplus Technology. G is the tester has been invited to do the test.

**C. Administrator Testing**

The testers were required to test all the following functions after login as administrators. One of the testers is an administrator of Suplus Technology.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test NO. | Testing | Excepted Outcomes | Result | Testers | |
|  | Test the search function on admin home page. | The data could be found by filling in the following search conditions: project ID, projects’ names, and customer ID. | **Passed** | A  G | |
|  | Test if the administrator could search for employees’ information on maintain employee page. | The data could be found by filling in the following search conditions: employee ID and employees’ names. | **Passed** | | A  G |
|  | Test if the user could edit employees’ information on maintain employee page. | The data could be updated. | **Passed** | | A  G |
|  | Test the “Create new employee” link on maintain employee page. | The system could redirect the user to create new employee page. | **Passed** | | A  G |
|  | Test if the user could insert information in the textboxes on create new employee page. | The textboxes could allow the user to insert information on create new employee page. | **Passed** | | A  G |
|  | Test if the administrator could create new employee on maintain employee page. | The new employee could be created. | **Passed** | | A  G |
|  | Test if the administrator could delete the existing employee on maintain employee page. | The employee could be deleted. | **Passed** | | A  G |
|  | Test if the administrator could create new customer on maintain customer page. | The new customer could be created. | **Passed** | | A  G |
|  | Test if the user could edit the customers’ information on maintain customer page. | The customers’ information could be updated. | **Passed** | | A  G |
|  | Test if the user could search for customers’ information on maintain customer page. | The data could be found by filling in the following search conditions: customer ID, and customers’ names. | **Passed** | | A  G |
|  | Test if the user could delete customer on maintain customer page. | The customer could be deleted. | **Passed** | | A  G |
|  | Test if the select list for departments’ information can be chosen on create new project page. | The select list for departments’ information could allow the user to choose a specific department. | **Passed** | | A  G |
|  | Test if the user could create new project. | The new project could be created. | **Passed** | | A  G |
|  | Test the search project function on maintain project page. | The data could be found by filling in the following search conditions: project ID, projects’ names, and customer project status. | **Passed** | | A  G |
|  | Test if the user could edit the projects’ information on maintain project page. | The projects’ information could be updated. | **Passed** | | A  G |
|  | Test if the user could delete project on maintain project page. | The project could be deleted. | **Passed** | | A  G |

A is an administrator working for Suplus Technology. G is a tester invited to do the test.

**D. Team leader Testing**

The testers were required to test all the following functions after login as team leaders. One of the testers is a team leader of Suplus Technology.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test NO. | Testing | Excepted Outcomes | Result | Testers | |
|  | Test if the team leader could search for projects’ information on manage project page. | The data could be found by filling in the following search conditions: project ID and projects’ names. | **Passed** | | T  G |
|  | Test if the textboxes allow the user to enter information and if the select lists allow the user to choose project and employeeon insert new task page. | The textboxes could be inserted; the project and the employee name select lists would include the project information and employees’ names. | **Passed** | | T  G |
|  | Test if the user could insert new task on insert new task page. | The new task could be inserted. | **Passed** | | T  G |
|  | Test if the user could search for tasks’ information on manage project page. | The data could be found by filling in the following search conditions: task ID, tasks’ names and employees’ names. | **Passed** | | T  G |
|  | Test if the edit task function allows user to update the existing tasks’ information. | The tasks’ information could be updated. | **Passed** | | T  G |
|  | Test if the delete task function allows user to delete the existing task on manage project page. | The task could be deleted. | **Passed** | | T  G |

T is the team leader of Suplus Technology. G is the tester invited to do the test.

### Debugging

During the user testing, it was discovered administrator was unable to create a new project, because the code was counting the number of records (the number of projects) and prepared to insert new project. However, in database, auto increment is enabled for project\_ID, and the project\_ID is supposed to be created automatically. Therefore, the code of counting projects was commented. The code was changed from Figure A to Figure B.

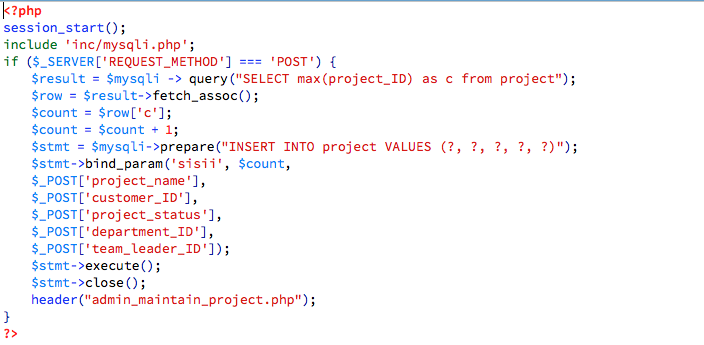


Figure A-1st version of create new project

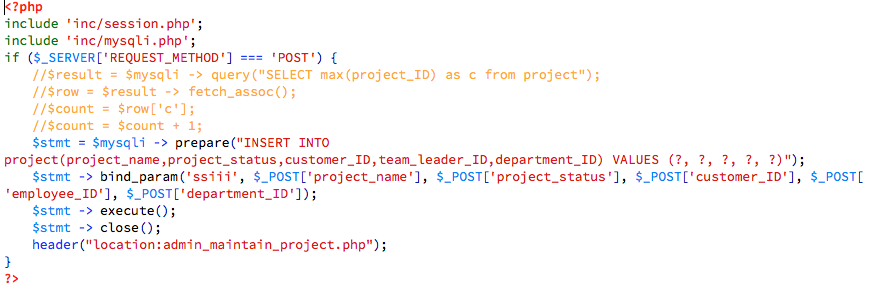


Figure 2- 2nd version of create new project

After changing the code, I asked the testers of administrator module to test create new project function again. As a result, the project could be created.

As the result, all the testers could complete the whole operation in 40 minutes. In addition, some testers made a comment that the user interface was not user friendly. To fix this problem, I adapted Bootstrap and jQuery to redesign the system, such as changing table style, navigation menu, and search menu.

### 12.2 System Functional Testing

The system function testing of this project consisted of manual testing, and I am the only tester. In the manual test, all the functionality had to be tested. For the detail of the test cases, please check Appendix B.

#### Result of system functional testing

Based on function design, Suplus Office Management System has been divided into three parts, which are employee module, administrator module, and team leader module. Therefore, the employee module, the administrator module, and the team leader module all have been tested.

#### All users’ functions testing result

Some basic functions of employee, administrator, and team leader are the same, such as login, update personal information, change password, and post message. Therefore, in this document, some test cases are combined together in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| Test No | Testing | Expected Outcome | Actual Outcome |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test if the information could be filled in the login textboxes on the login page. | Information could be entered in the textboxes of username and password. | Passed |
|  | Test if an error message is present to indicate invalid data or empty blank. | * Enter not existing username and password—fail to login the system. * Empty username and password—fail to login the system. | **Passed** |
|  | Test if the existing user could login to the system by entering the valid username and password. | The user could login the system. | **Passed** |
|  | Test if the system could send email with password to binding email address when user clicks the forgot password button on login page. | An email with password could be sent. | **Passed** |
|  | Test if the user could edit personal information on update information page. | The information could be updated. | **Passed** |
|  | Test if the user could change the password once signed in the system. | The password could be changed. | **Passed** |
|  | Test if the user could post a message on communication page. | The user who logins with valid username and password could post a message on communication page. | **Passed** |
|  | Test if the user could post a picture on communication page. | The user could select a picture from local directory, and the user could post a picture. | **Passed** |
|  | Test if the user could post comment on communication page. | The user could post a comment. | **Passed** |

#### Employee module testing result

In this case, tester should login as employee, and test all the employee functions listed below.

|  |  |  |  |
| --- | --- | --- | --- |
| Test NO. | Testing | Excepted Outcome | Actual Outcome |
|  | Test if the search textbox on employee home page allows user to input information. | The search textbox could allow the user to insert information. | **Passed** |
|  | Test the search function on employee home page. | The data could be found by filling in the following search conditions: task ID, tasks’ names and project ID. | **Passed** |

#### Admin module testing result

In the following part, the tester should login as administrator, and test the following listed functions.

|  |  |  |  |
| --- | --- | --- | --- |
| Test NO. | Testing | Excepted Outcome | Actual Outcome |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test that the administrator search function on admin home page. | The data could be found by filling in the following search conditions: project ID, projects’ names, customer ID. | Passed |
|  | Test if the administrator could search employees’ information on maintain employee page. | The data could be found by filling in the following search conditions: employee ID and employees’ names. | **Passed** |
|  | Test the user edit employees’ information function on maintain employee page. | The data could be updated. | **Passed** |
|  | Test the “Create new employee” link on maintain employee page. | The system could redirect the user to create new employee page. | **Passed** |
|  | Test if the information could be inserted into the textboxes on create new employee page. | The textboxes could allow the user to insert information on create new employee page. | **Passed** |
|  | Test the administrator create new employee function on maintain employee page. | The new employee could be created. | **Passed** |
|  | Test if the administrator could delete the existing employee on maintain employee page. | The employee could be deleted. | **Passed** |
|  | Test if the administrator could create new customer on maintain customer page. | The new customer could be created. | **Passed** |
|  | Test if the user could edit the existing customers’ information on maintain customer page. | The customers’ information could be updated. | **Passed** |
|  | Test if the user could search customers’ information on maintain customer page. | The data could be found by filling in the following search conditions: customer ID and customers’ names. | **Passed** |
|  | Test if the user could delete the existing customer on maintain customer page. | The customer could be deleted. | **Passed** |
|  | Test if the select list for departments’ information can be chosen on create new project page. | The select list could be chosen. | **Passed** |
|  | Test if the user could create new project on create new project page. | The new project could be created. | **Passed** |
|  | Test the search project function on maintain project page. | The data could be found by filling in the following search conditions: project ID, projects’ names and customer project status. | **Passed** |
|  | Test if the user could edit projects’ information on maintain project page. | The projects’ information could be updated. | **Passed** |
|  | Test if the user could delete the existing projects on maintain project page. | The project could be deleted. | **Passed** |

#### Team leader module testing result

In this case, the tester should login as team leader, and test all the team leader functions listed below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test if the team leader could search for projects’ information on manage project page. | The data could be found by filling in the following search conditions: project ID and projects’ names. | Passed |
|  | Test if the textboxes allow user to enter information and if the select lists allow user to choose project and employeeon insert new task page. | The textboxes could be inserted; the project and the employee name select lists would include the project information and employees’ names. | **Passed** |
|  | Test if the user could insert new task on insert new task page. | The new task could be inserted. | **Passed** |
|  | Test if the user could search for the existing tasks’ information on manage project page. | The data could be found by filling in the following search conditions: task ID, tasks’ names and employees’ names. | **Passed** |
|  | Test if the edit task function allows the user to update the existing tasks’ information. | The tasks’ information could be updated. | **Passed** |
|  | Test if the delete task function allows the user to delete the existing task on manage project page. | The task could be deleted. | **Passed** |

## Conclusion

Overall, this project has met all the requirements of Suplus Technology. According to the requirements of the client, Suplus Office Management System has adapted the prototype development methodology. The system is also using PHP, HTML and CSS as major programming languages, and adapting jQuery, JSP and Bootstrap as means to enhance the user interface layout. In addition, there is a social network page maintained by Mongo DB to allow the all users to post status (messages or pictures) and make comments.

In this system, two databases are adapted: MySQL and NoSQL (Mongo DB). On the one hand, MySQL database designs and manages information of employees, customers, department, projects, and tasks. On the other hand, for the sake of innovation and experience in advanced technologies, Mongo DB (NoSQL) collects data from the social media web page “Communication”, which would allow users of this office management system to post messages, pictures, and make comments on others’ messages and images.

The user interface of the web application is user friendly because this system adapts a new leading language, Bootstrap, which is the most popular language being used to develop responsive mobile project on the Web. During the testing part, Bootstrap can fix simple interface design problems that have been brought up by most testers.

In the process of developing the system, I practiced and learned a lot of programming languages and challenged myself to use new technologies such as NoSQL and Bootstrap to develop social media web page. Moreover, I used popular languages such as PHP, Bootstrap, jQuery, and JSP, to develop this system. This practicum is a meaningful challenge and a great opportunity to explore and practice new technologies and readily learned programming languages.

Lastly, I conducted users testing and functional testing to ensure that all the functions designed are working smoothly. In the user testing, participants were asked to follow the functional testing list to provide feedback that includes the results of the test and suggestions for future improvement. Based on test result, I debugged and fixed unfeasible functions by changing codes, and using Bootstrap to redesign the user interface. In conclusion, this practicum project perfectly satisfied the needs of the users of Suplus Technology.

## Future Enhancement

For future enhancement, a sending report function could be added to the system. In the future, the entire body of employees would be required to fill in a checklist to explain today’s work situation based on achieved tasks. The data generated by the checklist will be turned into a report that will be sent to customers at the end of every month. This report helps customers to be aware of the stage that they are at in the project.

In addition, to the targeted users mentioned in this paper, employees in the finance department could use this system. In the future design, the system could provide functions such as calculating employees’ salary. For example, the system could generate employee’s payment that may include bonuses given based on their work ethics and performance. Another function is to create a budget sheet for all the departments in the company. The budget sheet could keep track of the income and expenses. Due to the limited amount of time allotted for this project, this practicum version did not include the aforementioned functions.

## Appendix

### Work Breakdown

Based on the client’s requirement, I have designed five versions for Suplus Office Management System. These versions have been discussed in Part 5 Development Methodology.

According to the client’s requirement, the first mock-up version is also designed (see Develop Initial Prototype in the table below). The purpose of the initial mock-up version is to present and discuss ideas with users of Suplus Technology. Improvement in the mock-ups has been made according to users’ suggestions.

During the system design, I transformed the latest mock-ups to a HTML and CSS version. Then I adapted PHP, MySQL, Mongo DB and Bootstrap to the system. After the transformation, I tested the system based on the test cases and invited the general testers and users of Suplus Technology to test the system. All the testers were required to provide test results and suggestions.

At the end, according to the test results and suggestions collected from testers, the problems of the system have been fixed and the user interface layout has been improved. This is the final version of the system.

In order to demonstrate the work I have been doing since last year, I spent almost five months to write the report document. This documentation explains the system functions and detail of the design process.

|  |  |
| --- | --- |
| **Components** | **Time Breakdown** |
| **1.Identify basic functional requirement** | |
| Decide the system functions based on the user requirement. | March-April 16, 2014 |
| **2. Develop initial prototype** | |
| Based on the user requirement to create initial mock-up | April 17– May 10, 2014 |
| Improve mock-ups | May 14– June 1, 2014 |
| **3.System design** | |
| Website design/review | June 3– June 5, 2014 |
| Develop MySQL database | June 8- July 6, 2014 |
| Develop employee maintain function | July 10- August 20, 2014 |
| Develop customer maintain function | August 25-September 6, 2014 |
| Develop project maintain function | September 9- October 15, 2014 |
| Develop Mongo DB | October 17- November 5, 2014 |
| Develop communication page | November 8- December 29, 2014 |
| Enhance User Interface design | January 3- January 23, 2015 |
| **4. Test** | |
| Test and debug | January 23-Febuary 20, 2015 |
| **5.Final report** | |
| Documentation | November 20,2014 – March 31, 2015 |

### Test Cases

This section demonstrates the test cases have been conducted during the system function testing.

A. All user test cases

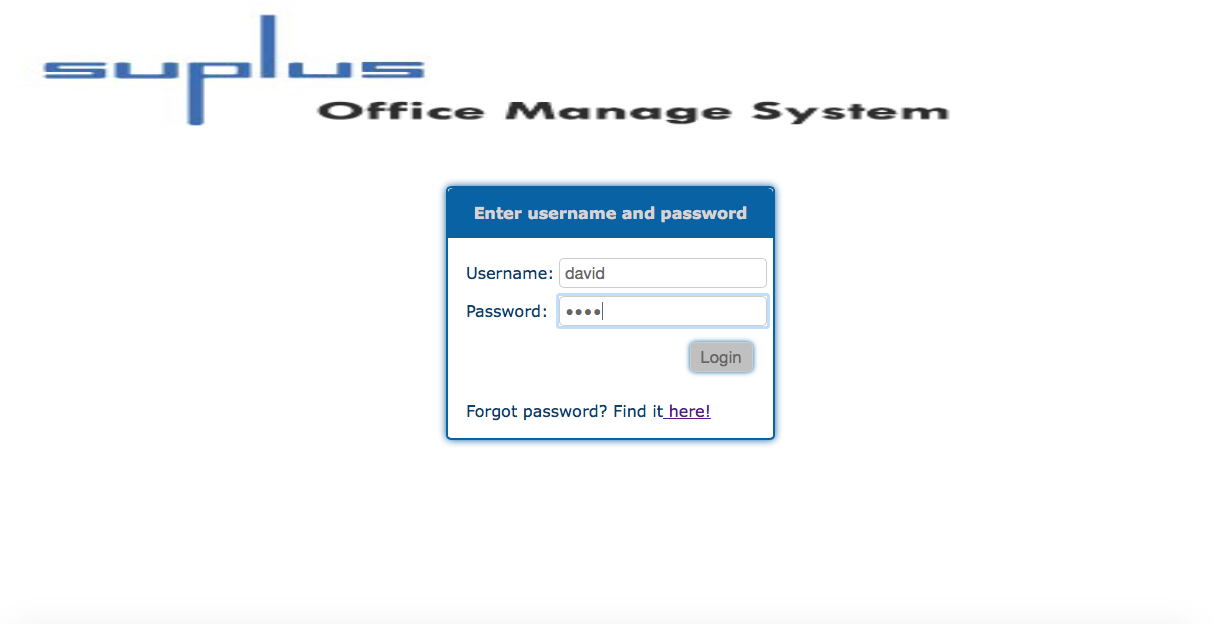
Login

1. Test that the login textboxes could be filled with information on the login page.

Test Scenario T1:

1. Enter username into username textbox.
2. Enter password into password textbox.

Test result:



Information could be inserted in the login textboxes. The test is passed.

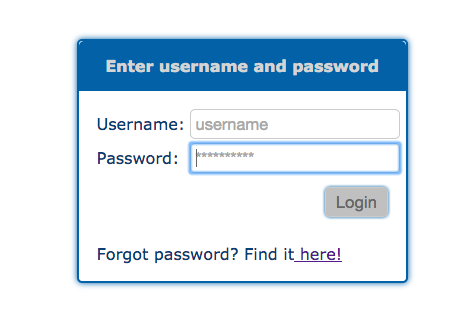
1. Test that an error message is present to indicate the user enter the invalid data or empty blank.

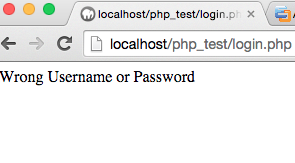
Test Scenario T2:

1. First, do not enter anything in the textboxes.
2. Second, input incorrect information (information that does not exist) in the textboxes.
3. Click login button.

Test result:

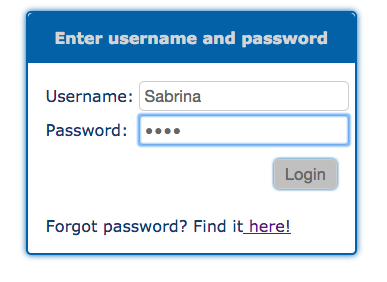
First time:

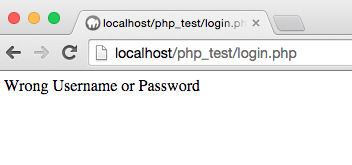




When the user logged in without entering any username or password, he or she could not login to the system.

Second time:





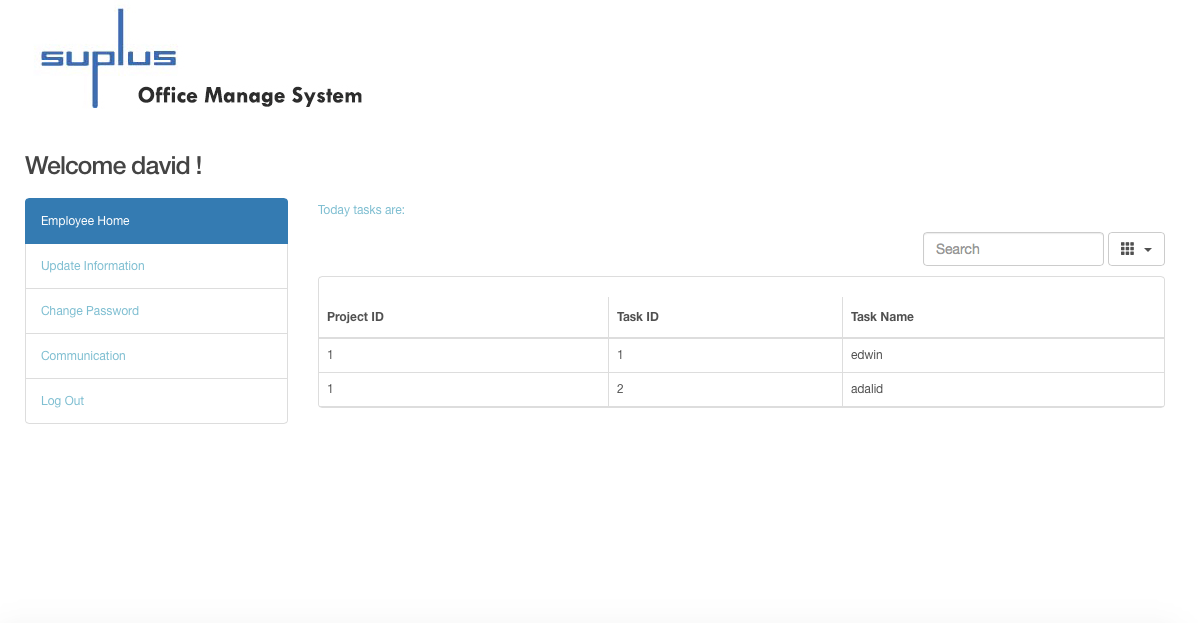
If the user logged in the system with username or password that did not exist, he or she could not login the system. The test is passed.

1. Test if the existing user could login to the system by entering the valid username and password.

Test Scenario T3:

1. Input correct username and password. For example, enter username as David, password as1234.
2. Click login button.

Test result:



When user entered existing login information, the user could login to the system, the above picture showed that after entering David’s username and password, David could login to his personal home page. This test is passed.

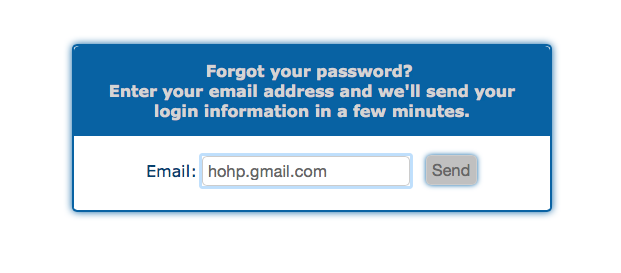
1. Test if the system could send the email with password to binding email address when the user clicks the forgot password button on the login page.

Test Scenario T4:

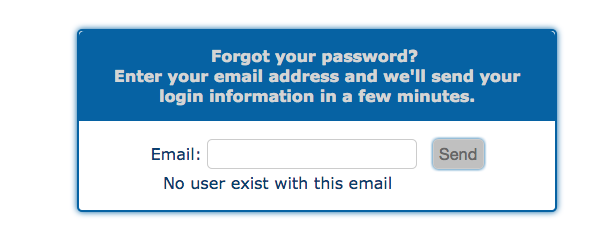
1. Click the forgot password link on the login form.
2. First, input non-existing or invalid email address.
3. Second, input existing and valid email address.
4. Click login button.
5. Check email inbox whether the user receives the new letter with the password.

Test result:

First time:



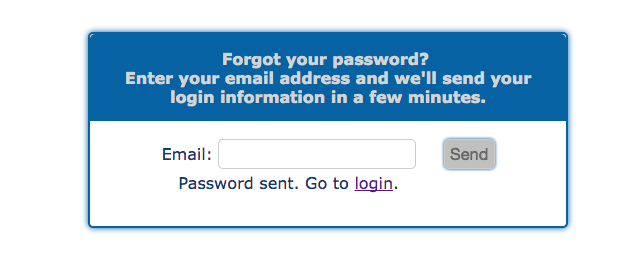
When the previous incorrect email address was entered and the send button was clicked, the system gave a reply “No user exist with this email.”

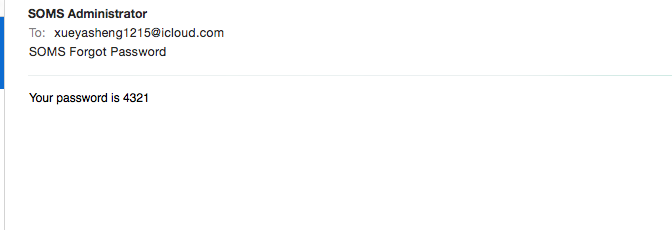


When the user enters incorrect email address, the system could not connect database with this invalid information.

Second time:

When the existing user’s email address was entered, password was sent instantly.





When entering the correct and existing email address, the user got an email with password. The test is passed.

Update personal information

1. Test if the user could edit personal information on update information page.

Test Scenario T5:

1. Enter the valid information into the textbox, change postal code from V5H 1P2 to V5H 1P1.
2. Click update button.

Test result:





The information could be updated. The test is passed.

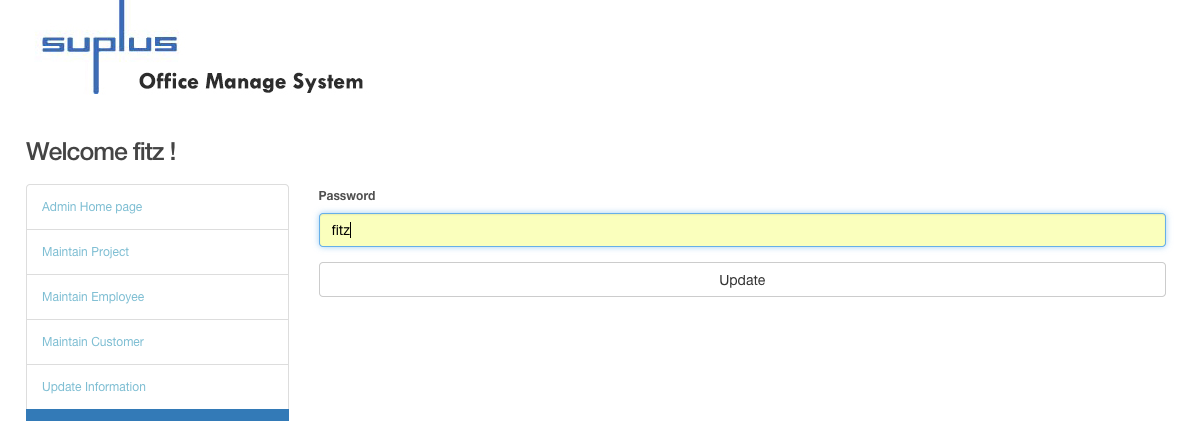
Change password

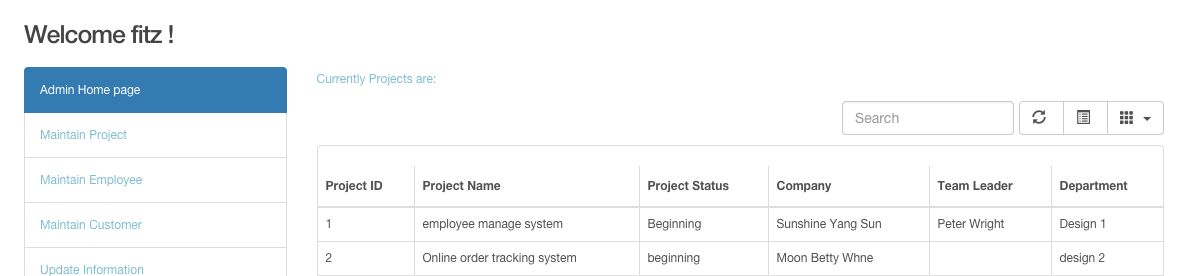
1. Test if the user could change password once have signed in the system.

Test Scenario T6:

1. User logins to the system already.
2. Click the change password link.
3. Enter new password, change user Fitz’s password from “4321” to “fitz”.
4. Click Update.

Test result:





The user Fitz logged in as administrator with password “fitz”. The test is passed.

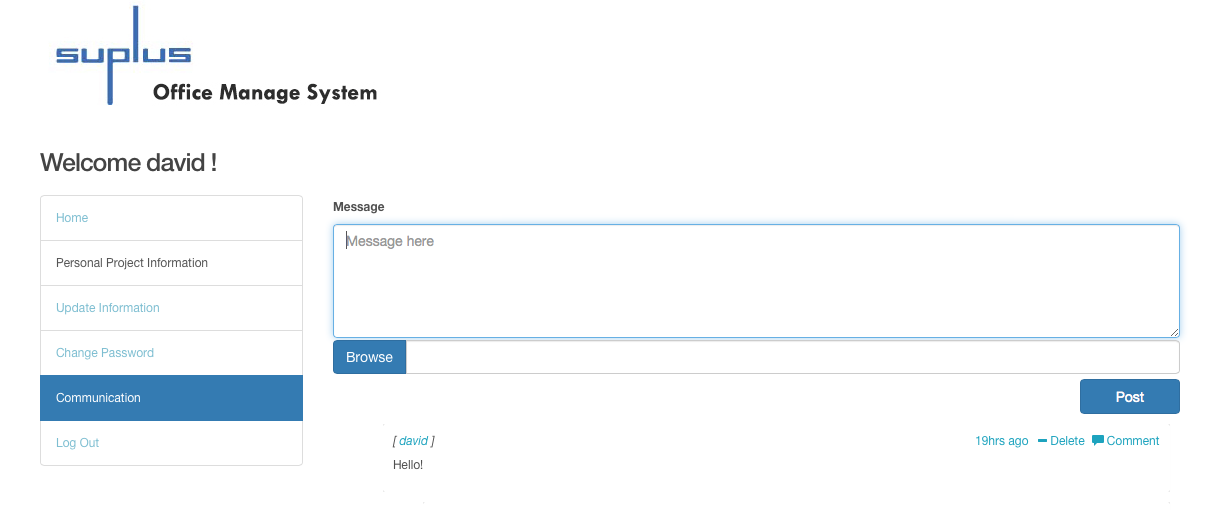
Communication

1. Test if the user could post a message on communication page.

Test Scenario T7:

1. Login to the system by using user name David, and password 1234.
2. After the user logins to system, user enters text “Hello!” on communication page.
3. Click post button.

Test result:

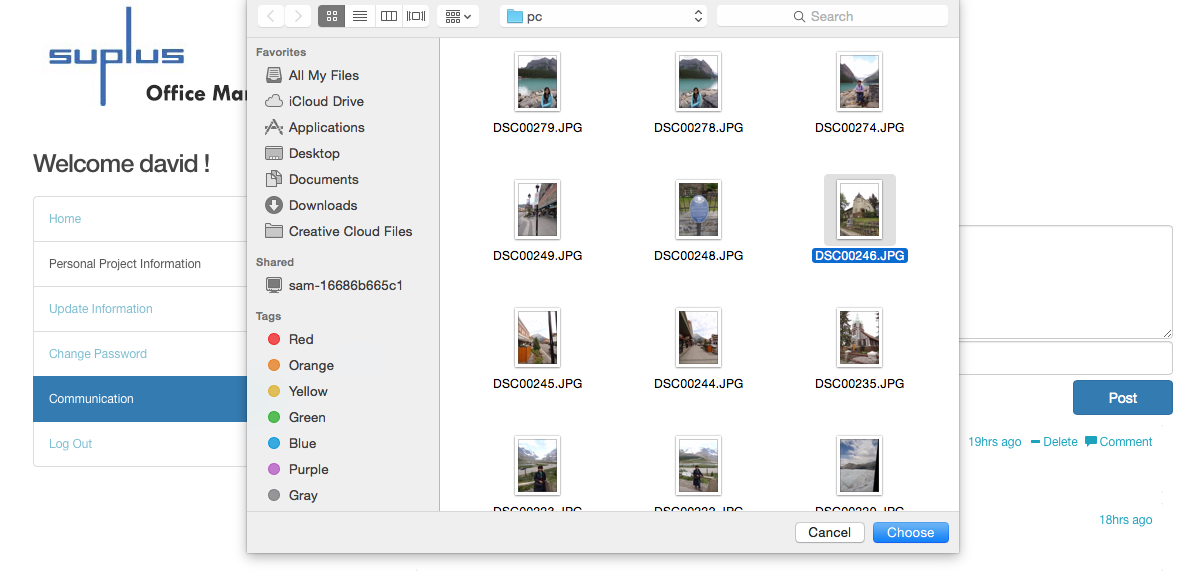


The message was posted successfully. The test is passed.

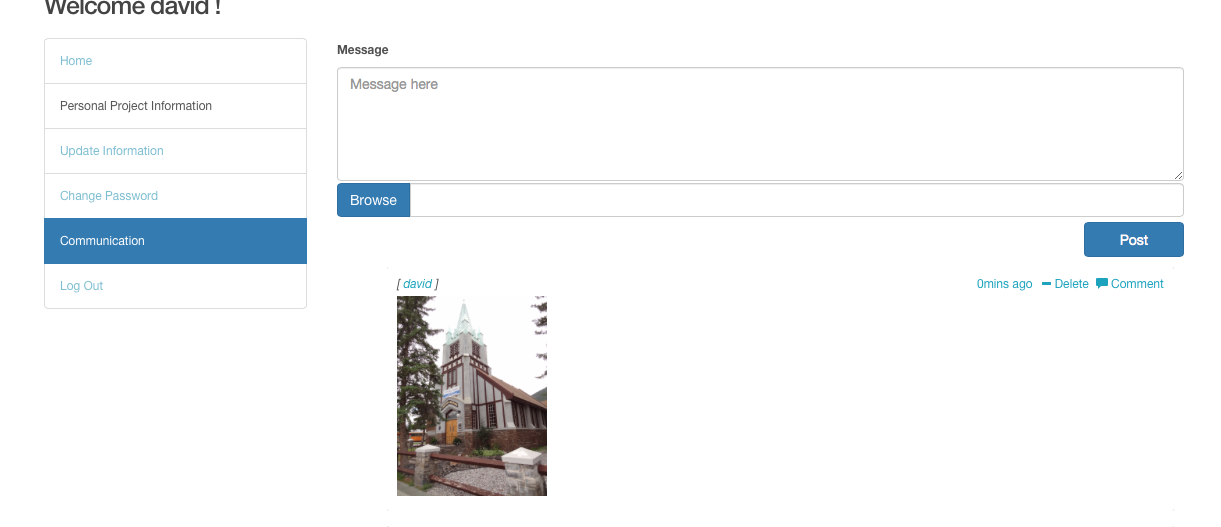
1. Test if the user could post a picture on communication page.

Test Scenario T8:

1. Click Browse button.
2. Choose a picture from the pop-up window.
3. Click the choose button.
4. Choose the picture, and click the post button.

Test result:

After the user clicks the choose button, the picture is selected, and click the post button.



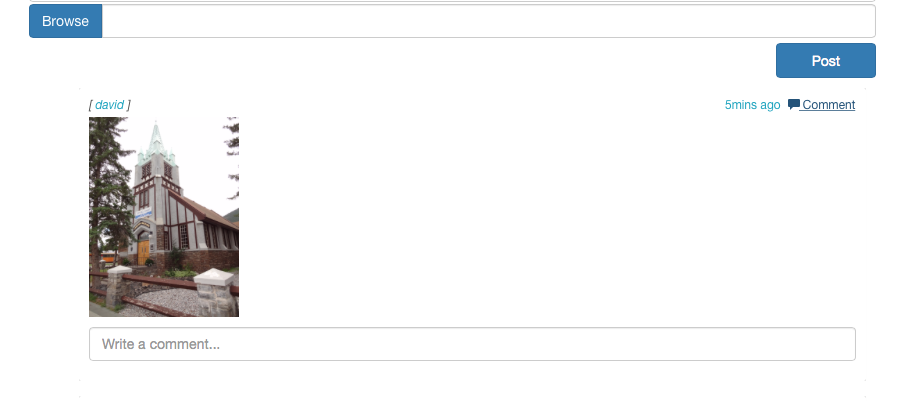
The picture was posted successfully, and the test is passed.

1. Test that the user could post comments on communication page.

Test Scenario T9:

1. Login as user Peter and prepare to comment on David’s picture.
2. Click comment.
3. Type in a message.

Test result:





The message from Peter that comments on David’s picture has been posted. The test is passed.

B. Employee test case:

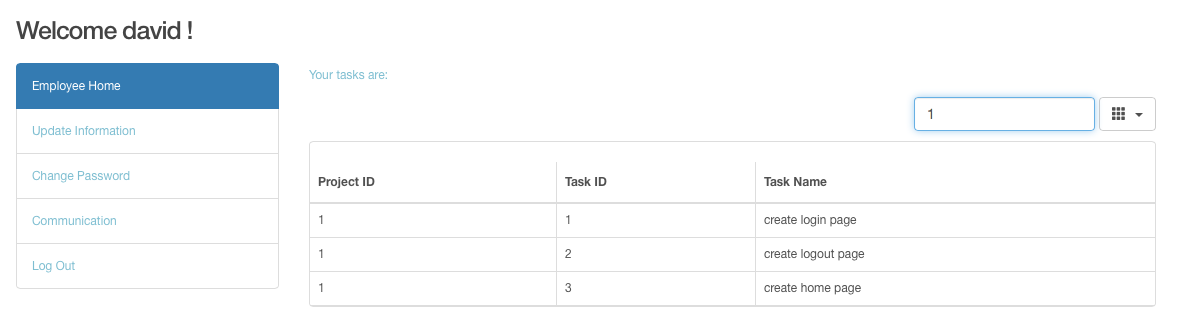
Check personal project information

1. Test if the search textbox on employee home page allows the user to input information.

Test Scenario T10:

1. Enter Task\_ ID as 1.

Test result:



Information could be inserted to the search textbox. The test is passed.

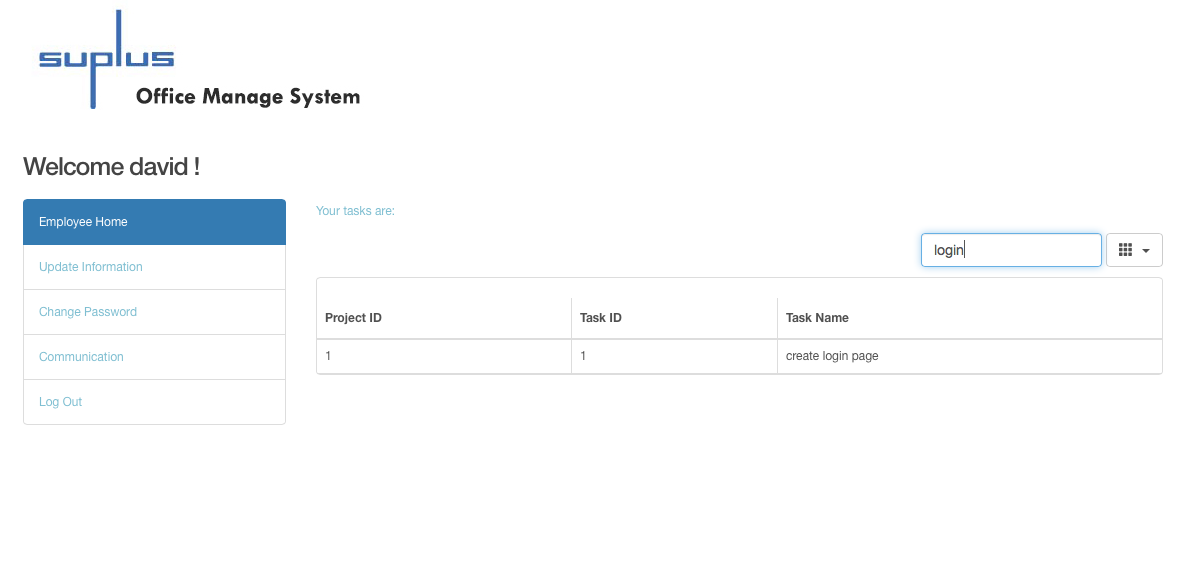
1. Test the search function on employee home page.

Test Scenario T11

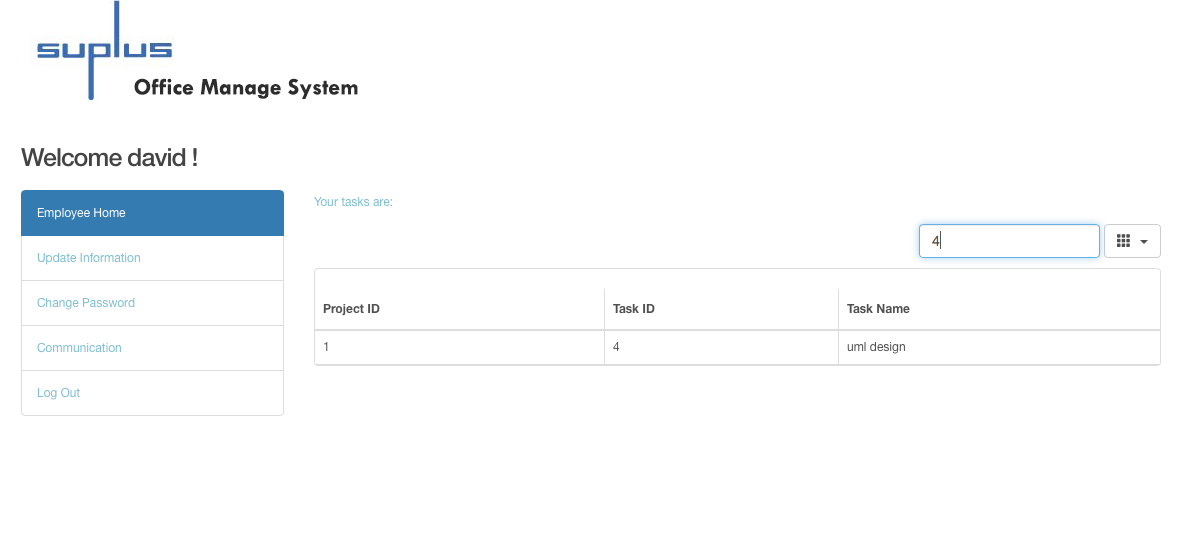
1. Enter valid task name in this case, enter “login” as task name.
2. Enter valid task ID, type in 4 as task ID in the search bar.
3. Enter valid project ID in this case, search for related information when project ID is 1.
4. Enter non-existing information, such as “rest”.

Test result:

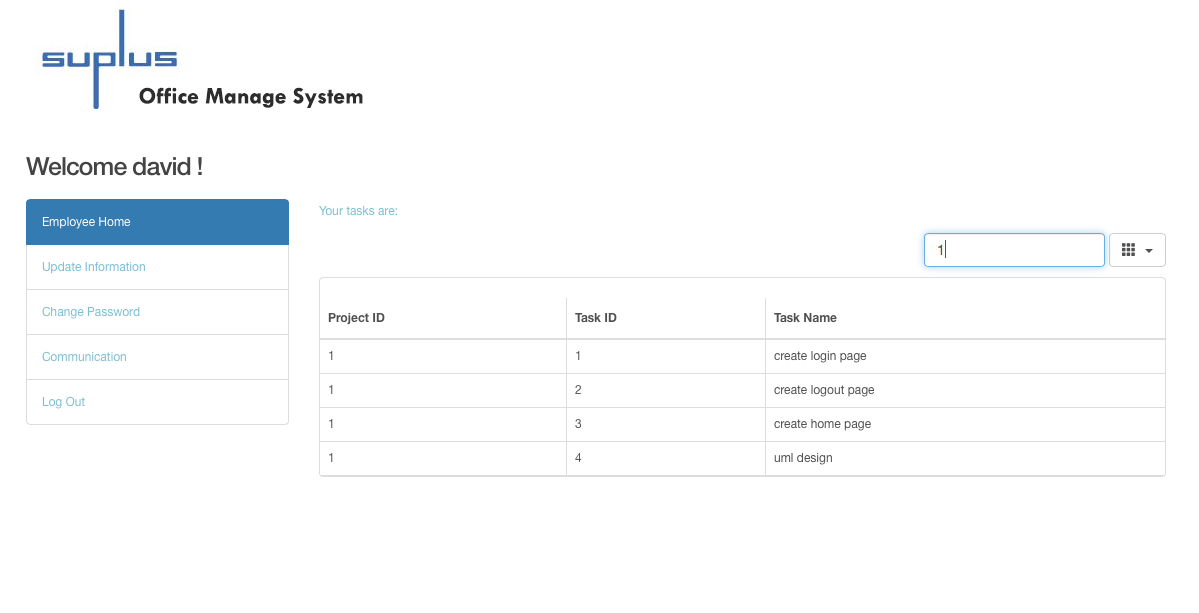
Search by task name “login”:



The data could be found.

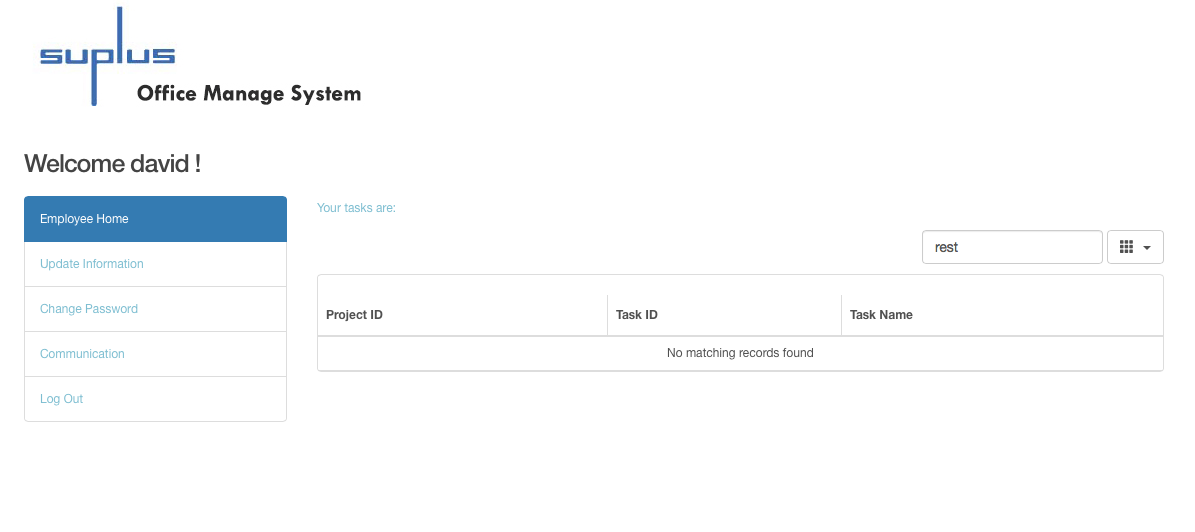
Search task ID as 4:

The data could be found.

Search project ID as 1: 

The data could be found.

Search for non-existing information “rest”:



The system presented “No matching records found” when searching for non-existing information.

The data could be found when the user entered the existing information. The test is passed.

1. Administrator test case

Home Page

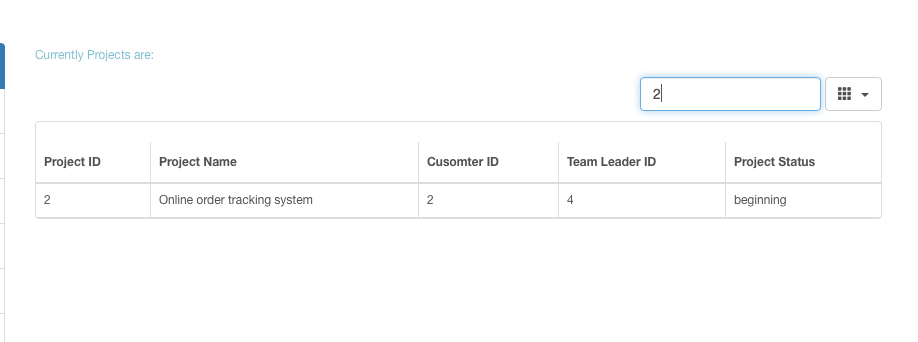
1. Test the administrator search function on admin home page**.**

Test Scenario T12:

1. Enter valid project ID, input 2 as project ID.
2. Enter valid project name, input existing project name.
3. Enter valid customer ID, type 1 as customer ID.

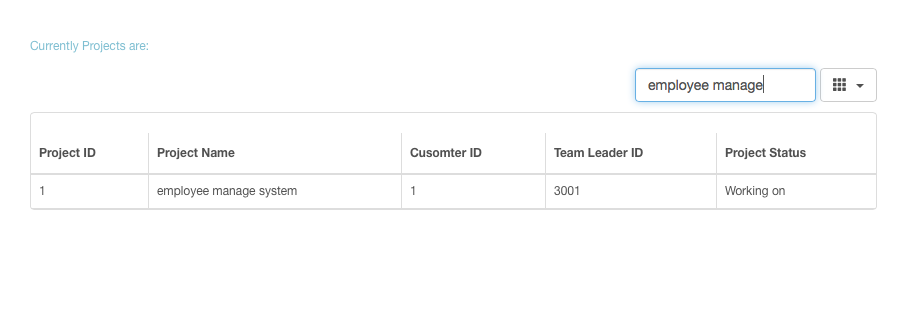
Test result:

Search project information when project ID is 2:



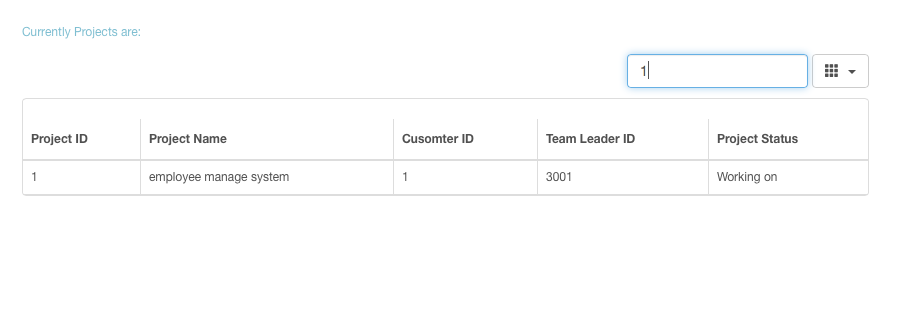
Typed in 2 as project ID, and a record was found. The test is passed.

Search for project name when project name as “employee manage”:



Figures show the existing project could be searched by project name.

Search customer information when customer ID is 1:

  
Type 1 as customer ID, and 1 record is found. The test is passed.

Therefore, administrator could search for project related information on the home page. The test is passed.

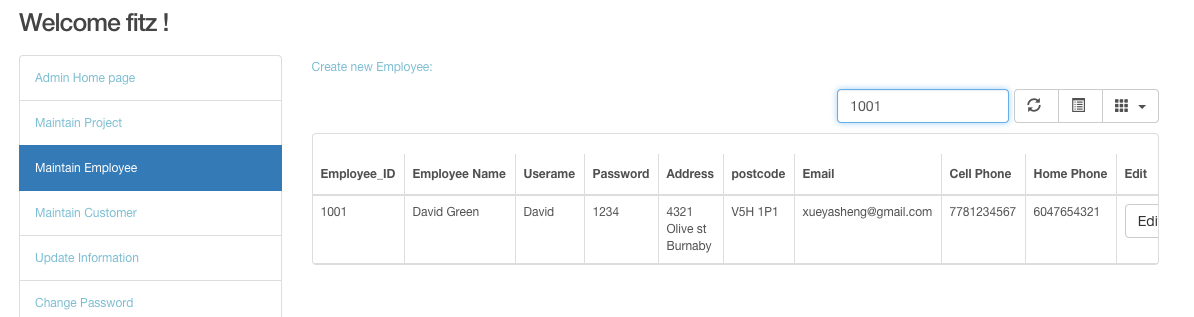
Maintain Employee

1. Test if the administrator could search for employee’s information on maintain employee page.

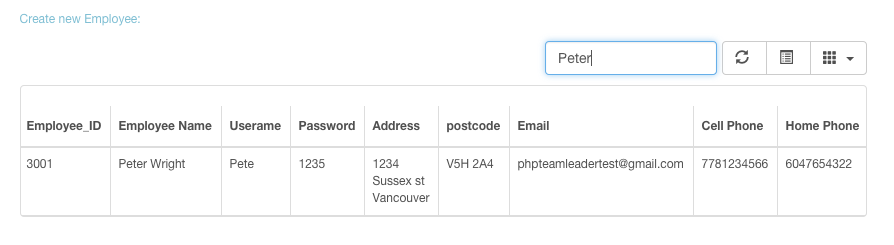
Test Scenario T13:

1. Search employee information by entering employee ID as 1001.
2. Search employee information by entering employee name as Peter.

Test result:

Search for an employee whose ID is 1001: 

Search for an employee named Peter:



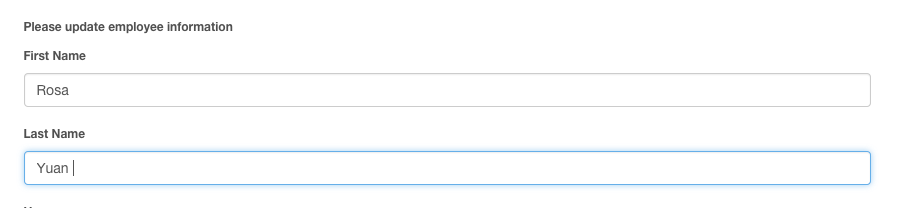
The data could be found. The test is passed.

1. Test the user edit the existing employees’ information function on maintain employee page.

Test Scenario T14:

1. Choose data that employee name is “testdfd”.
2. Click the edit button.
3. Change employee name to “Rosa Yuan”.
4. Click update button.



Insert Rosa Yuan to the name textboxes: 

Test result:



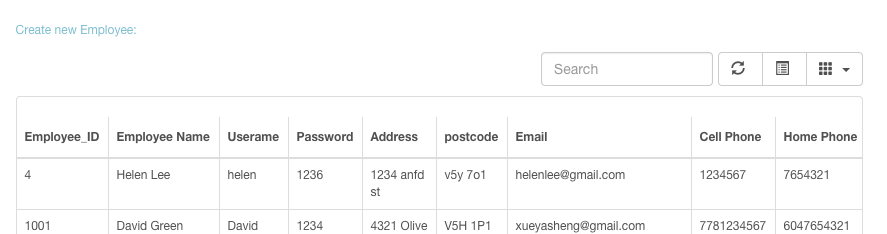
The employee name has been changed. The test is passed.

1. Test the “Create new employee” link on maintain employee page.

Test Scenario T15:

* + - 1. Click add new employee link on maintain employee page.

Test result:



After the user clicked the link, the system went to create new employee page.



Clicking the link, the user was directed to the adding new employee page. The test is passed.

1. Test if the information could be inserted in the textboxes on create new employee page.

Test Scenario T16:

* + - 1. Enter new employee name as “Roman Wang”.

Test result:



Information could be inserted in the textboxes. The test is passed.

1. Test the administrator create new employee function on maintain employee page.

Test Scenario T17:

1. Enter new employee name as “Roman Wang”.
2. Click Add button.

Test result:



Roman Wang has been created. The test is passed.

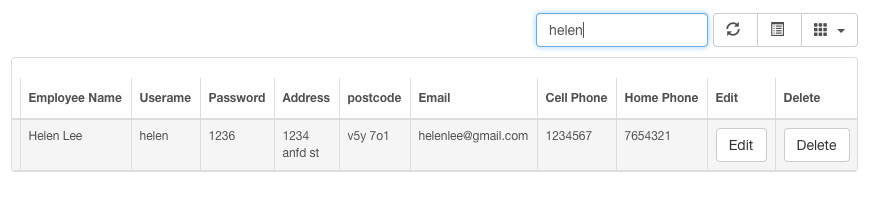
1. Test if the administrator could delete the existing employee on maintain employee page.

Test Scenario T18:

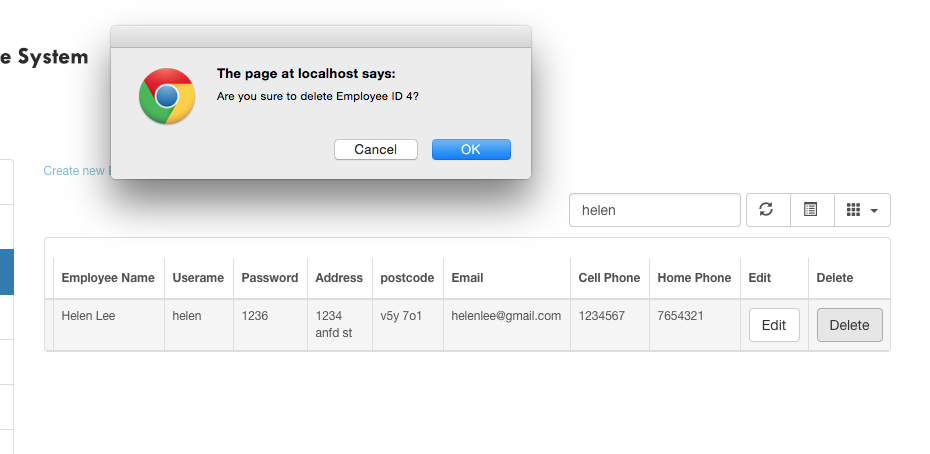
1. Search for the information of an employee whose name is Helen.
2. Click delete button.
3. Click ok button on “Are you sure to delete information” pop-up window.

Test result:

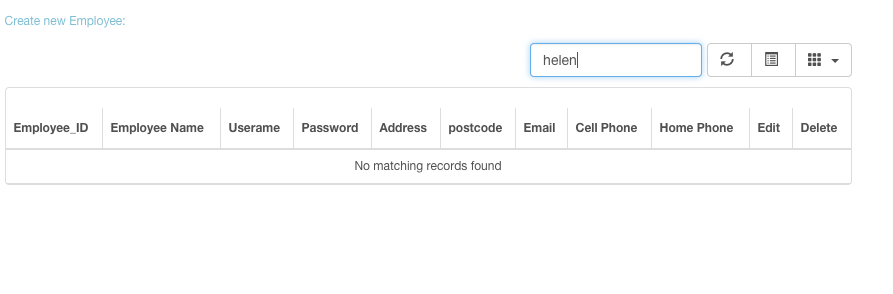
Search for Helen and click the delete button:



Click ok button on pop-up window:



Search for Helen again, and date could not be found.



The information has been deleted. The test is passed.

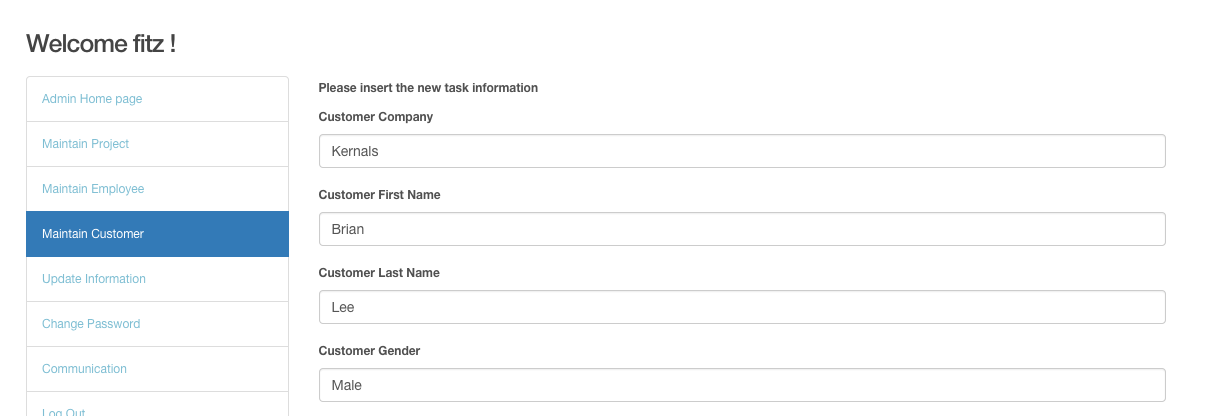
Customer Maintain

1. Test the administrator create new customer function on maintain customer page.

Test Scenario T19:

* + - 1. Click create new customer link
      2. Insert customer’s information
      3. Click add button

Test result:



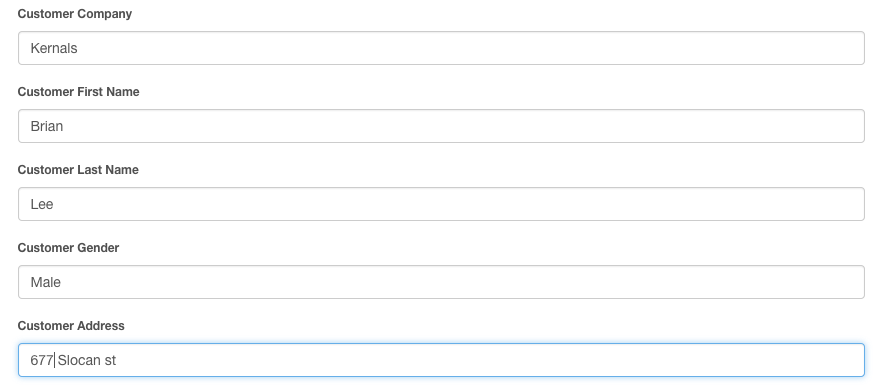
The picture above indicates that a new customer has been inserted. The test is passed.

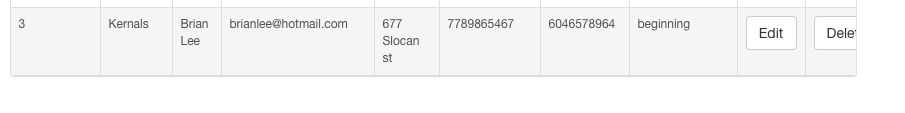
1. Test if the user could edit the existing customers’ information on maintain customer page.

Test Scenario T20:

1. Choose the data customer ID as 3.
2. Click edit button.
3. Change customer’s address from “678 Slocan” to “677 Slocan”.
4. Click update button.

Test result:





Customer’s address has been changed from 678 to 677. The test is passed.

1. Test the user search customer function on maintain customer page.

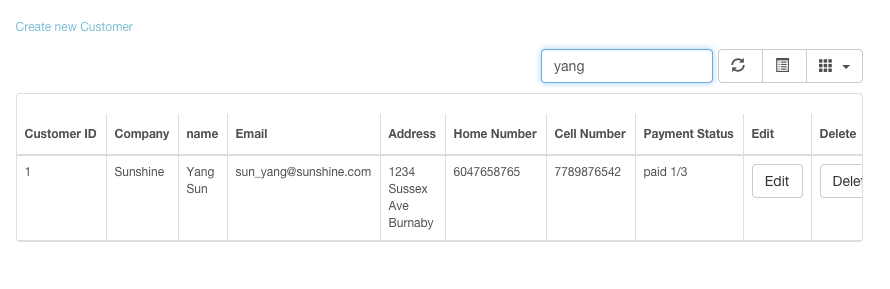
Test Scenario T21:

1) Enter customer name as “Yang” in search textbox.

2) Enter customer ID as“1” in search textbox.

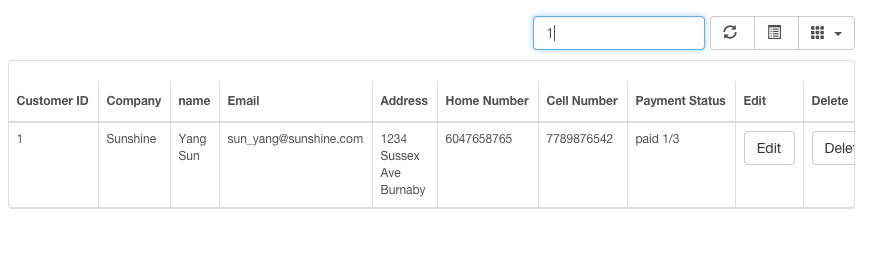
Test result:

Search for customer name:



When the user entered the existing customer’s name, the data could be found.

Search for customer ID:



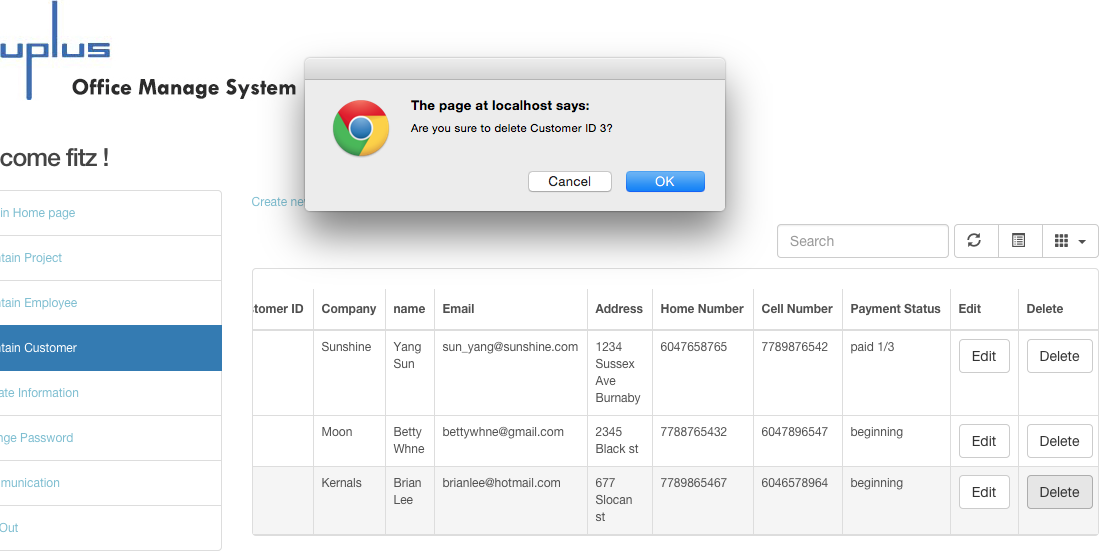
When the user entered customer ID as “1” to search for information, the data could be found. The test is passed.

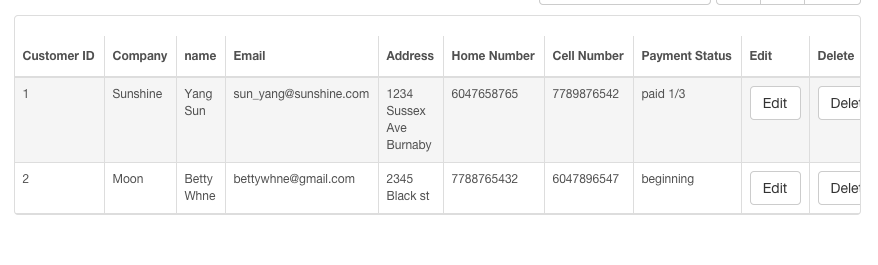
1. Test if the user could delete the existing customer on maintain customer page.

Test Scenario T22:

1. Search for customer’s company name “Kernals”.
2. Click delete button.
3. Click ok button on pop-up window.

Test result:





Customer’s information with kernels has been deleted. The test is passed.

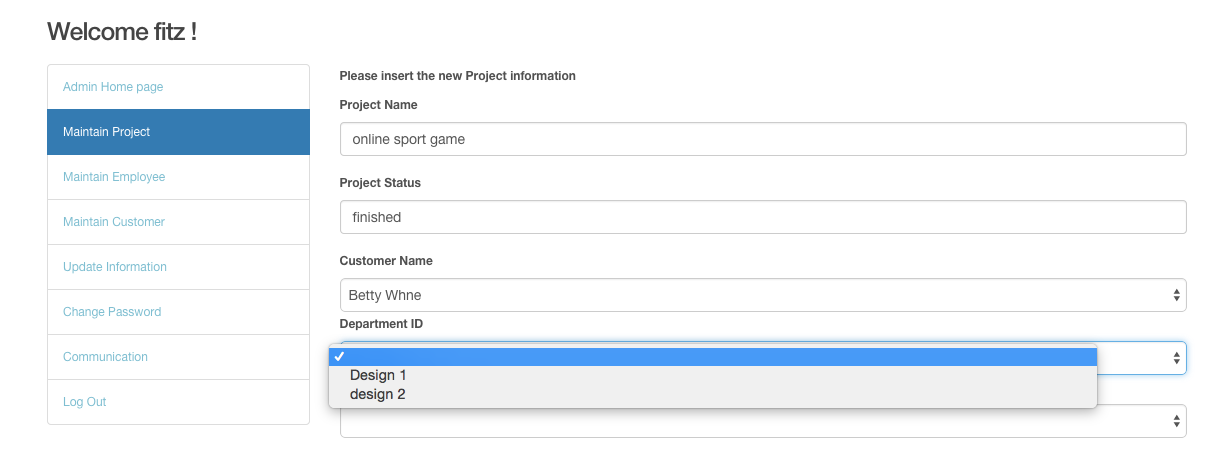
Maintain Project

1. Test if the select list for departments’ information can be chosen on create new project page.

Test Scenario T23:

1. Click create new project.
2. Enter project name as “online sport game”.
3. Click “Department ID” select list button.

Test result:



The text has been inserted. The select list contains Design 1 and Design 2. The test is passed.

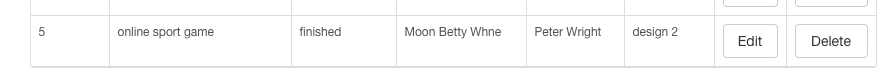
1. Test if the user could create new project on create the project page.

Test Scenario T24:

1. Insert project name.
2. Choose customer’s name.
3. Type in project status.
4. Choose department.
5. Choose Team leader.
6. Click add button.

Test result:





The new project has been created. The test is passed.

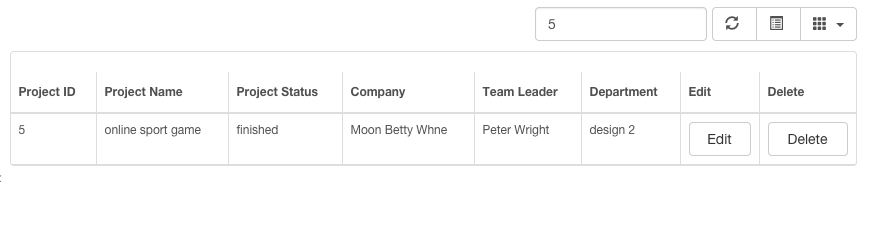
1. Test the search project function on maintain project page.

Test Scenario T25:

1. Search for project information by entering project ID as 5.
2. Search for project information by entering project status as “fin”.
3. Search for project information by entering project name as “track”.

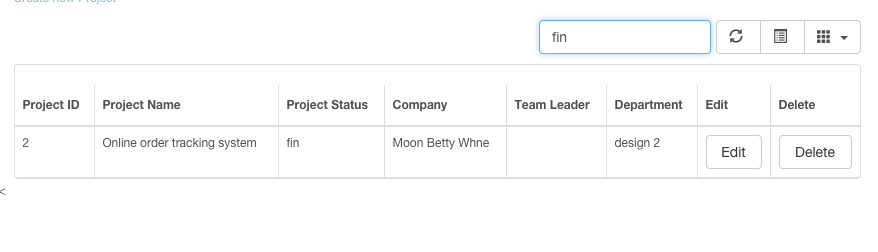
Test result:

Search for project ID as 5:



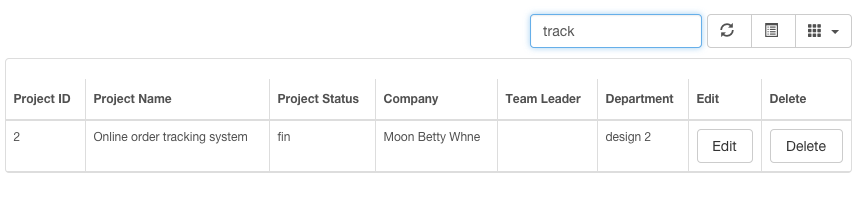
The data could be found.

Search for project status as “fin”:



The data could be found.

Search for project name as “track”:



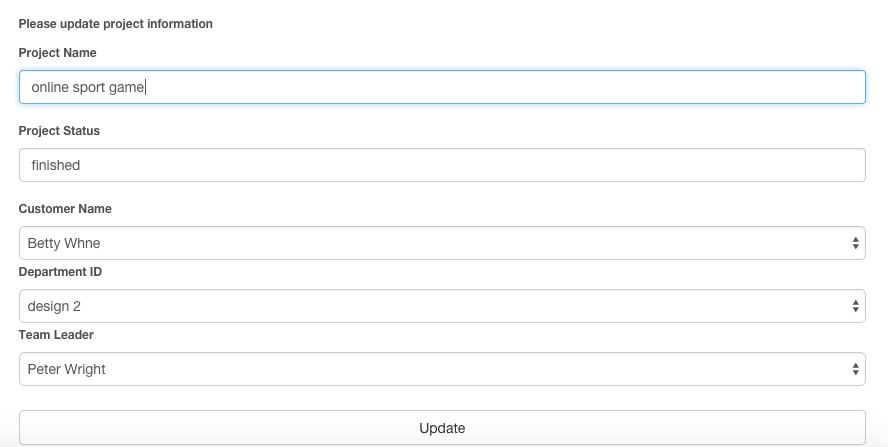
The data could be found. The test is passed.

1. Test if the user could edit the existing projects’ information on maintain project page.

Test Scenario T26:

1. Choose the data project ID as 5.
2. Click edit button.
3. Change project status from “finished” to “paid 1/3”.
4. Click update button.

Test result:





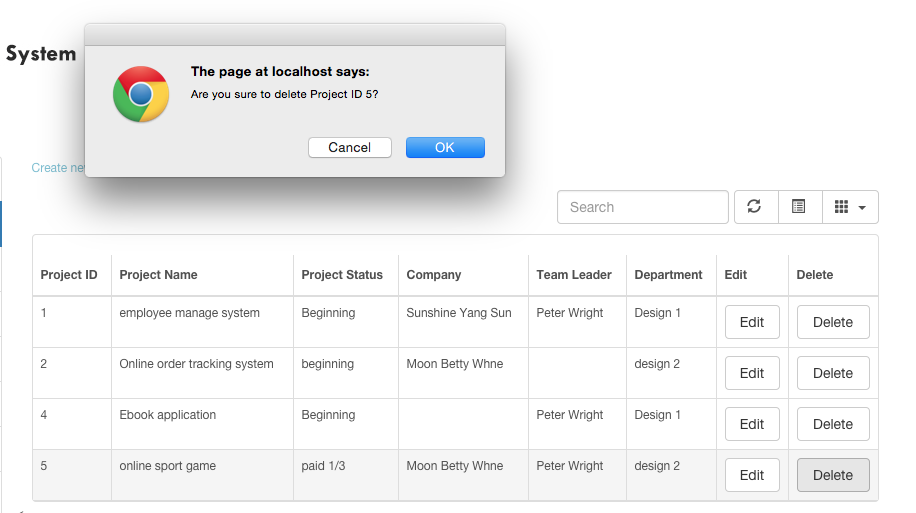
The project status has been changed. The test is passed.

1. Test if the user could delete the existing project on maintain project page.

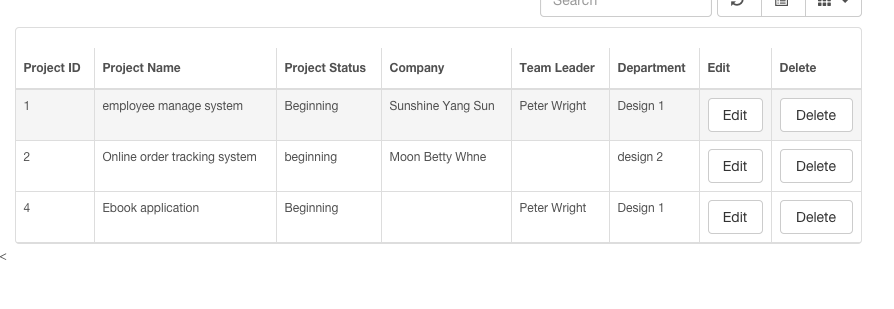
Test Scenario T27:

1. Choose data project ID 5.
2. Click delete button.
3. Click ok button on pop-up window.

Test result:



After clicking ok button:



The No. 5 project’s information has been deleted. The test is passed.

D. Team leader test case

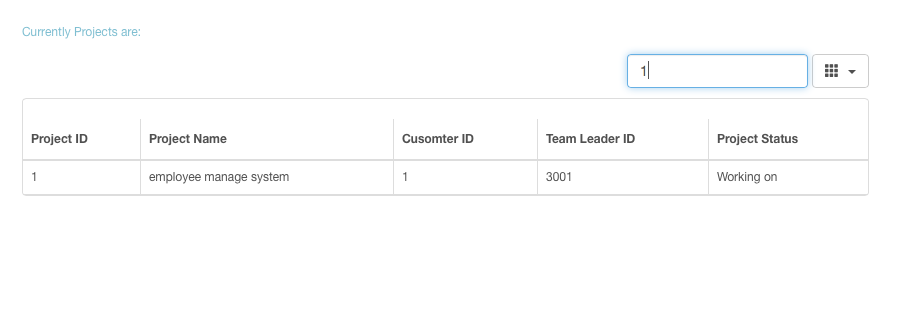
28. Test if the team leader could search for projects’ information on manage project page.

Test Scenario T28:

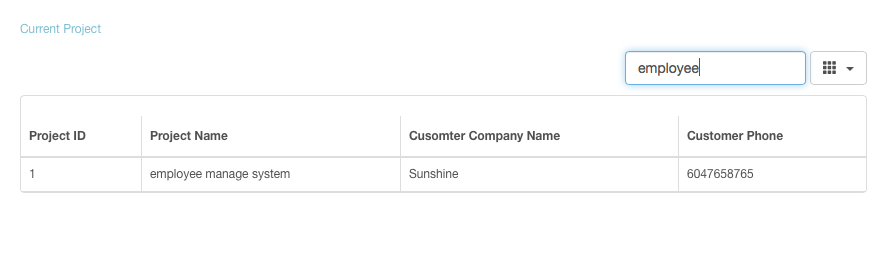
1. Search for project information by entering project ID as 1.
2. Search for project information by entering keyword “employee” of project name.

Test result:

Search for project ID:



Search for project name:



When the user searched for project ID, a record could be found, and when the user searched for project name by entering the keyword, the existing data could also be found. The test is passed.

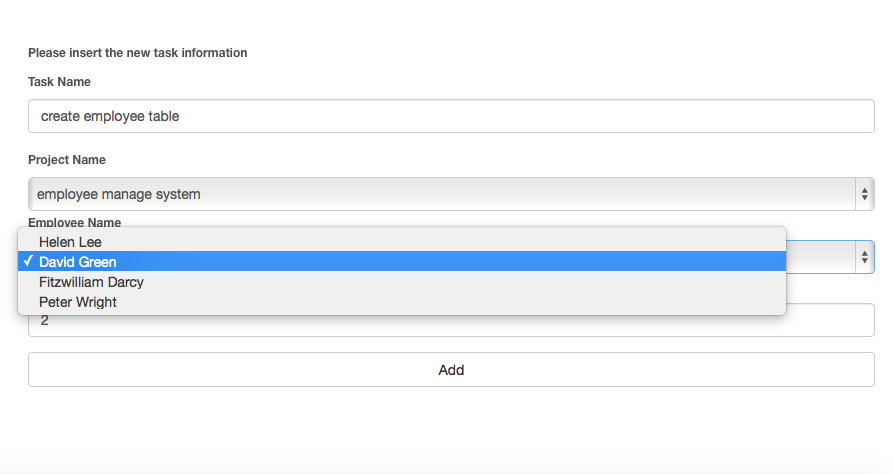
Manage projects

29. Test if the textboxes allow user to enter information and the select lists allow the user to choose project and employeeon create new task page.

Test Scenario T29:

1. Click the insert a new task link on the top of manage project page.
2. Insert the task name as “create employee table”.
3. Click the project name select list button and choose the existing project.
4. Choose employee name from the select list.
5. Enter the estimated time.

Test result:



The information could be inserted into the textboxes, and the select lists could be chosen. The test is passed.

30. Test if the user could insert new task on insert new task page.

Test Scenario T30:

1. Click the Insert a new task link on the top of manage project page.
2. Insert the task name as “create employee table”.
3. Click the select list button and choose the existing project.
4. Choose an employee name.
5. Enter estimated time.
6. Click add button.

Test result:



The task has been created. The test is passed.

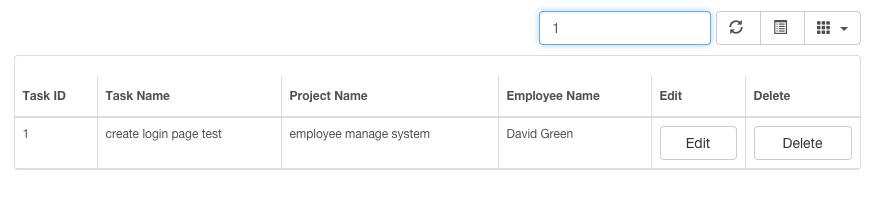
31. Test if the user could search for tasks’ information on manage project page.

Test Scenario T31:

1. Search for task’s information by entering task ID as 1.
2. Search task’s information by entering task name as “login”.
3. Search task’s information by entering employee name as “green”.

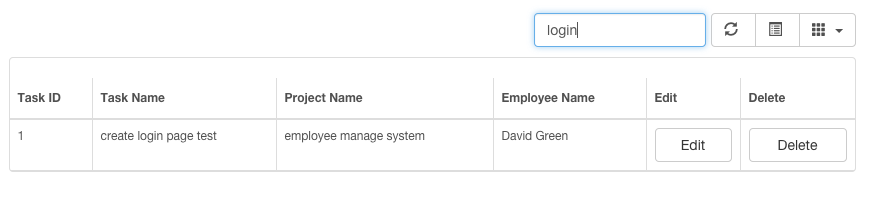
Test result:

Search task information by entering task ID as 1:



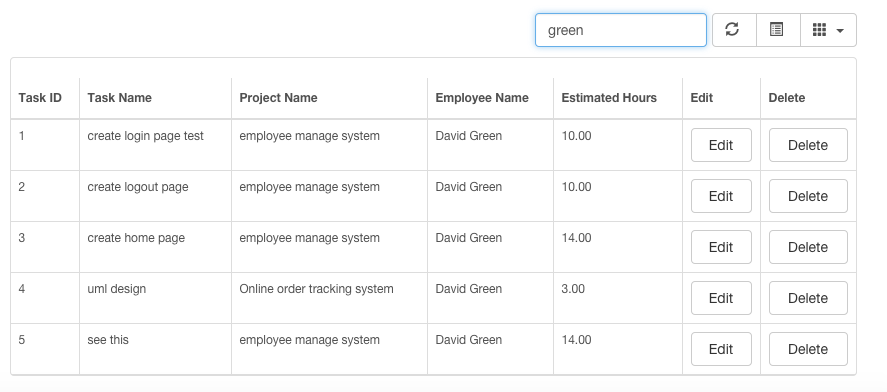
The data could be found.

Search for task information by entering task name as “login”:



The data could be found.

Search for task information by entering employee name as “green”:



The data could be found. The test is passed.

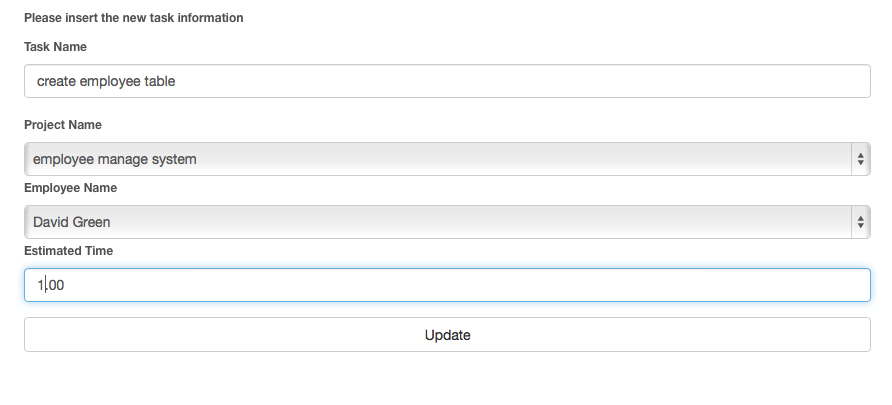
32. Test if the edit task function allows the user to update the existing task’s information.

Test Scenario T32:

1. Choose task “create employee table”.
2. Click the edit button.
3. Change the estimated time from 2 to 1.
4. Click the update button.



Change the estimated time from 2 to 1:



Test result:



The estimated time has been changed to 1. The test is passed.

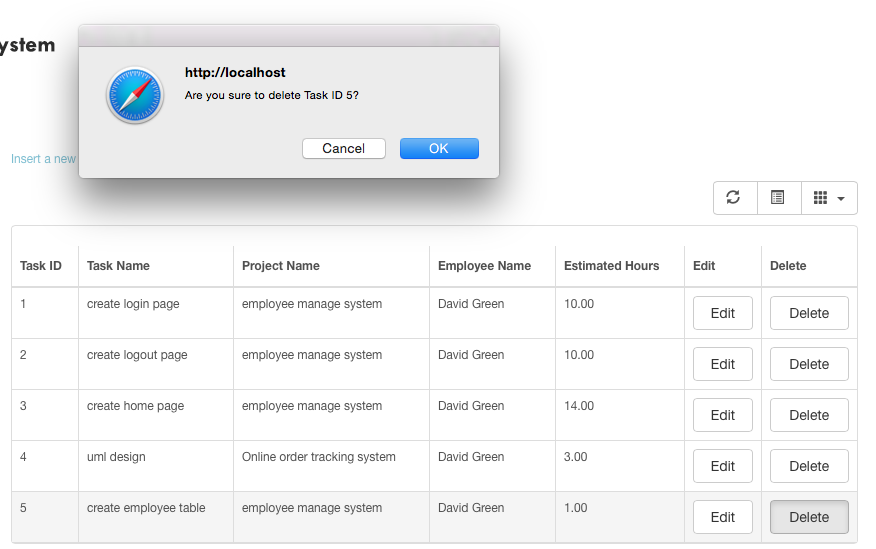
33. Test if the delete task function allows the user to delete the existing task on manage project page.

Test Scenario T33:

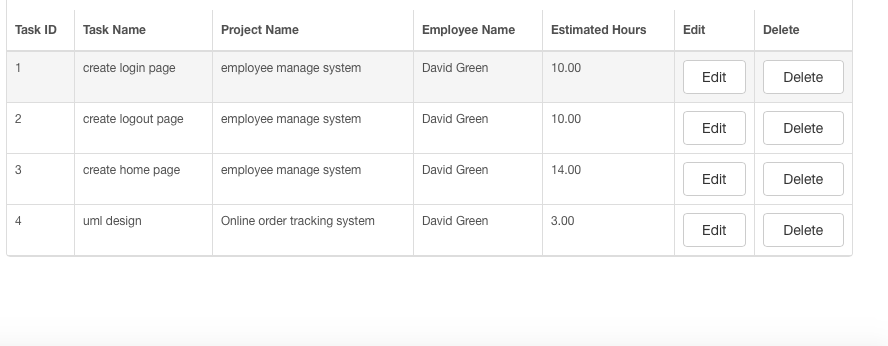
1. Choose task “create employee table”.
2. Click the delete button.
3. Click ok on the pop-up window.

Click the delete button on NO.5 task:



Click ok on pop-up window: 

The result:



Information has been deleted. The test is passed.

### C. Testers’ Survey of User Testing

To improve the system operation, I conducted a survey, and I listed one survey as an example.

Web browser: Google Chrome

Could you select the module you have been assigned?

B

A. Employee Module B. Administrator Module C. Team Leader Module

How long did it take you to complete all the tasks based on the task list?

A

A. 20min-30 min B. 30min-1hour C. more than 1 hour

Do you have any comment?

No.

## Reference

[Leavitt, N.](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Leavitt,%20N..QT.&searchWithin=p_Author_Ids:37284031700&newsearch=true) (2010). Will NoSQL Databases Live Up to Their Promise? *Computer.*

Retrieved from: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5410700&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D5410700

Francia, S. (2012). *MongoDB and PHP*.

O’Reilly Media, CA

Sadalage, P., Fowler, M. (2013). *NoSQL distilled: a brief guide to the emerging world of polyglot.*

Person Education. Retrieved from:

https://books.google.ca/books?id=AyY1a6-k3PIC&printsec=frontcover&dq=NoSQL+database&hl=en&sa=X&ei=GsYTVd2YCNCuogShuoKIDQ&ved=0CDAQ6AEwAA#v=onepage&q=NoSQL%20database&f=false