# Research Student Manager
# Project Report

Yingduo Tang

**May 11, 2014**

# Table of Contents

# Abstract

This project is about developing a web-based management tool that can be used to manage research students in a university by the administrators. The purpose of developing such a tool is to automate the research student management tasks usually carried out manually by administrative staff of the university. By introducing and implementing this tool, this project aims to improve the automation and the efficiency of the management process.

Through this process, the research student manager has been designed and built with software engineering principles and methodologies. It has realized functionalities including student manager, supervisor information, project management and notifications. Since the focus of administrative work of research student at a university is around managing students, the system in this project reflects just this throughout the design process.

The results of this implementation and the system built hopes to improve the work efficiency of research student management work at the university for the administrators, and bring about automation and innovation in their work process. It would be even more beneficial if the findings from this project can provide implications for other researches and projects.

## 摘要 (Chinese translation of the Abstract)

本项目的主要内容是开发一个基于网页的能够帮助大学里的教务管理人员进行研究生管理的管理工具和在线信息系统。开发这样一个工具的目的旨在帮助大学里的管理人员将他们日常的程序化的研究生管理任务实现自动化。因而引入和实现这样的一个工具，本项目的最终目标是提高研究生管理工作过程的工作效率。

在该项目进行的过程中，该“研究生管理系统”的开发始终遵循软件工程开发的原理和方法。该系统最终实现了包括学生管理、导师信息、项目管理和通知等的功能。在大学的管理中，研究生管理的核心任务依然在于对学生的管理，因此在该“研究生管理系统“的设计过程中，也重点体现了对于研究生的管理功能。

该项目成功开发的系统以及研究成果希望能够帮助大学的管理人员提高研究生管理工作的工作效率，并且为研究生管理工作的过程带来自动化和创新。若该项目的发现能够为其他研究和项目提供帮助亦将更加大有裨益。

# Chapter 1: Introduction

## 1.1 Project Overview

Student management is one of the major administration tasks carried out by administrative staff in today's universities which involves the manipulation of personal information of student and professors as well as educational resources from many different perspectives. With the development of database technology and active web page technology, many universities have transformed their student management tasks from the traditional paper-based work to electronic-based records in the university's internal information systems. The benefits of introduction such an information system is to bring about process automation and innovation to the traditional student management work, and improve work efficiency for the administrative staff of universities.

In research intensive universities, specifically, the management of research students involves management of research related specific information like research projects, supervision relations, and research funding. Therefore, the management system for research students has to be designed, built and customized to suit the needs of the administrators that manage research students, so that their work efficiency can be maximized. This project aims to build such a web-based research student manager with multiple common and extra functionalities and discover how it can automate the student management process and improve their work efficiency.

### 1.1.1 Project Objectives

The main objectives of this project is to design and implement a research student manager for the administrative staff in universities in order to automate their management work. To realize this objective, the needs and specification of the administrative staff should be thoroughly analysed through requirement analysis. The research student manager will have functionalities like viewing student information and supervision details which are parts of the daily management tasks of university administrators. These functionalities will then be justified in terms of how it realizes project objectives and satisfies the requirements.

### 1.1.2 Project Significance

The automation and innovation from traditional paper based tasks to electronic records and information systems has not only been a major information technology movement in business industry, but an objective to improve work efficiency in education industry. Like Stuppy puts it, "It has long been an objective in education to automate the handwritten pen and paper-and-pencil testing process and to develop a computer-based diagnostic and prescriptive student assessment system for use in tutoring and supplemental education franchises, corporate education and school-based learning centre.” (Stuppy, 2000) Management information systems have been introduced to many aspects of school administrative tasks and have been proved to reduce paper work cost.

However, there are many issues associated with designing and implementing information systems to a specific context of use. The requirements of users are sometimes not fully discovered, and the management system are not easily customized to be easily usable and accessible by its users.

The significance of this project is to discover how a research student manager can be built to better satisfy the specific needs of research student management work, and seek solutions to the issues existing with nowadays student management systems. By clearly defining the scope of this project, this project also hopes to provide implications for future research and implementation efforts in building research student management systems.

### 1.1.3 Project Outcomes

The outcome of this project is a completely implemented web-based research student manager that satisfies the requirements of administrative staff. The entire designing and implementation process of the research student manager will be well-documented, and users will be provided with help and user manuals to deploy and user the system. An evaluation will also be carried out to estimate how well the system satisfies the requirement specifications. In short, the outcomes of this project are:

- Design of research student management tool meeting the stated requirements.
- Working implementation of the tool and evaluation of the tool.

### 1.1.4 Technical Context

The research student manager is coded with HTML and PHP, and deployed on an Apache HTTP server. A MySQL database is designed and implemented in the management tool for storing information. Although all the programming language and applications used in this project are cross platform, most implementation and testing of the research student manager is done on the local server hosted on a personal computer running Microsoft Windows 7.

## 1.2 Report Overview

### 1.2.1 Background

This chapter main talks about the requirement finding results from documentation, literature review and interview. A detailed analysis of these requirement findings are also discussed. There’s also a section briefly talks about the technical terminologies related to this project.

### 1.2.2 Design and Implementation

This chapter presents the entire process of design and implementation of the research student manager built in this project. It starts from requirement engineering, which elicited functional and non-functional requirements from requirement gathering. Then it goes on talking about the detailed process of database design and implementation, and the webpages design and implementation of the system. In the end, there’s a section about testing and evaluation.

### 1.2.3 Results and Discussion

This chapter demonstrates the completed research student manager. It shows all the functional modules of the research student manager through some screenshots along with some discussions. The test outputs of system testing is also discussed. Some user reviews from some random users is also cited at the end of this chapter.

### 1.2.4 Conclusion and Further Work

This chapter concludes the paper with summarizing the work completed by this project and the discoveries and implications to future research. Also discussed is the further work that remains to be improved for the research student manager and this report.

# Chapter 2: Background

In this chapter we discuss the background information about this project including the analysis of the requirement findings which intends to provide an overview of the current existing research student management system used by administrative staff and seek possibilities for improvements. Another section of this chapter briefly describes all the relevant technology used in this project.

## 2.1 Requirement Finding Results

In the initial phase of this project, the requirement analysis of the research student manager tool is carried out using software engineering methodologies. In this period, the administrative staff at BUPT is engaged for requirement finding, and their needs as the main user group of the software is explored as an important source of requirements. Relevant literature of similar implementation reports are also reviewed as references for better analysis of the requirements of this project. But most importantly, the requirements are elicited from the project specification in the project initiation documents. We'll present here our findings from these requirement analysis.

### 2.1.1 Requirement Finding Techniques

The requirement finding techniques used in this project includes mainly the following three ways:

(1) Documentation: requirements are analysed from the project initiation documents that includes the project description, the main tasks of the project and the project outcomes.

(2) Literature Review: relevant implementation reports that builds similar management information systems are referred to as a source of requirement findings.

(3) Interview: an interview was carried out with the administrative staff from BUPT

### 2.1.2 Requirement Finding Results

In this section we present the completing findings from the three requirement finding techniques we've mentioned in the previous section. To better illustrate the needs of our user, mainly the administrators, we'll also apply UML use case modelling techniques. By organizing and analysing these findings, we'll also try to elicit and category our findings into functional and non-functional requirements.

#### 2.1.2.1 Requirement findings from documentation

As one of the major sources of requirements gathering, the project initial document is used in the very beginning of this project as the initial source of project requirement finding. A detailed project specification is included in the project initial documents, which includes: (1) Description of the project about project nature, (2) The main tasks of the project, and (3) The measurable outcomes of the project. From these in the PID, the following requirements findings for the research student manager has been organized.

- The problem that the research student manager is trying to solve is to automate the administration tasks that are usually carried out manually in a university.
- The software application of the research student manager will be web-based.
- The goal is to automate the management of research student right from the stage of applying through to graduation.
- The basic functionalities involved should include: keeping track of funding details, managing assessment stages and supervision details, and sending notifications to students and supervisors.
- More extra functionalities can be identified and implemented like: different levels of access control, and organizing assessment sessions.
- The tool can be developed specifically for either QM or BUPT.

#### 2.1.2.2 Requirement findings from literature review

After the project specification has been made clear with the specified information in the project initial document, some literature and development project reports of similar student management information systems has been reviewed.

The purposes and goals of reviewing these literature and reports are to: (1) have a basic understanding of the software engineering process applied to designing and building a student management system, (2) comparing the trade-offs of the technical infrastructure implemented by these projects (technical environment, programming language, etc.) and decide on the appropriate technical context for this project, and (3) by analysing the commonalities and differences between similar projects, determine any possible functional and non-functional requirements for this project. With these objectives in mind, the findings from the literature review indicate that most of the projects are carried out through a complete sequential software engineering project of requirement analysis, system design, system implementation and testing. Documentations are maintained for each of these phases. The “design and implementation of a university registry office information system” project is a very good example of maintaining detailed documentation from the beginning to end of the project (Zhonghua, 2008). This will also be the methodology applied in the QVW7 research student manager project, with agile development as the main development methodology.

The literature findings also revealed that the different projects have used different technology infrastructure for implementation. For example, the Zhonghua, 2008 project implemented the system as an independent JAVA application, the Lijia 2011 project built the system with ASP (Active Server Page) and MS SQL Server, the Songqing 2012 project used JSP+Servlet+Ajax framework and technology, while the Yun 2013 and the Jinlong 2009 project both used Apache+MySQL+PHP framework. Based on a preliminary estimation of the advantages and disadvantages of these technology, the QVW7 research student manager project has decided to use Apache+MySQL+PHP framework as the technical context. The benefits of it will be discussed in the 2.2 Technical Terminology section.

The requirement findings in the literature also included some common functional and non-functional requirements in the student management systems built in all the projects. For example, every project has the functionality where the administrator can view basic student and supervisor information, and in every project the administrator can have some control over certain records in the system database. These requirements will be further discussed in the Design and Implementation Chapter as functional and non-functional requirements.

#### *2.1.2.3 Requirement findings from interview*

After relevant literature has been reviewed, an interview is carried out with the administrative staff at BUPT main campus. The interviewed staff at the administrative office is able to provide the following comments on the current system used:

- The current system is not very easy to use and respond slow sometimes in refreshing pages.
- The project information is currently not available in the student management system.
- The relationship between the student and the supervisor is not easily viewable in the system.
- The look and feel of the system is not very friendly.

So the requirement findings from the interview is limited to only some non-functional comments from an overview aspect of the system, which can hardly lead to any functional requirements specified. The reasons of this can be that though the actual administrators are the users of the system, it’s very easy for them to comment on the overall system quality while they might not be in a good place to judge and determine the actual functionalities in the system. The actual functionalities are still determined by the existing work process of administrative tasks.

## 2.2 Requirement Analysis

Based on the requirement findings gathered and discussed in the previous section, the administrator has been identified as the single main user of the Research Student Manager. Notice that even though students, supervisors and other stakeholders are also relate closely to the Research Student Manager because their individual information is stored and processed in the system, the management system is still used only by the administrators of the university to manage research students, so it is only the administrator that is identified as the main actor. The use cases of the administrator is illustrated as the following use cases in the use case diagram.

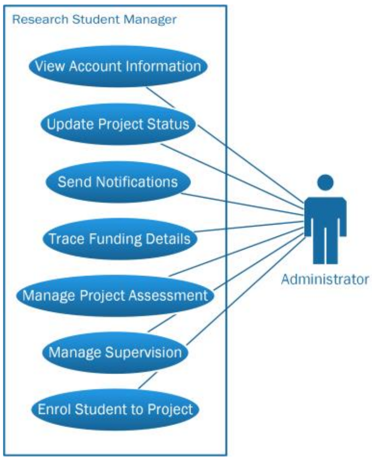


**Figure 1 Use Case Diagram - Research Student Manager: Administrator**

### 2.2.1 Use Case: View Account Information

**Precondition:**

The administrator logs in with their valid credentials and navigates to the student information or supervisor information module.

**Description:**

The administrator should be able to see a list of all the students and supervisors. The administrative should able to view the detailed information for each student and supervisor. For example, their names, IDs, and faculties. The administrator can filter the student and supervisor by their program or faculty, and can also search students and supervisors by their names or id number.

**Post-condition:**

The administrative staff is able to find the information of certain students or supervisors that they need, and is able to view detailed individual information.

**Exception:**

If the desired student or supervisor information is not entered into the database, the administrator can’t find the desired information maybe because it is not entered to the database.

### 2.2.2 Use Case: Update Project Status

**Precondition:**

The administrator logs in with their valid credentials and navigates to the project management menu and goes under the project status sub module.

**Description:**

The administrator should be able to view the current project status of all the projects. This include the current stage of the project (early, midterm and final, and also whether the project is dropped or completed or not). The administrator should be able to change the current stage of any specific project.

**Post-condition:**

The administrator is able to obtain the project status information of any given project, and also able to change the project status of any given project.

**Exception:**

If the project has been dropped or has been completed, the administrator will not be able to modify the status any more.

### 2.2.3 Use Case: Send Notifications

**Precondition:**

The administrator logs in with their valid credentials and navigates to the send notification module.

Research Student Manager

The administrator needs to notify students or supervisors of a certain project.

**Description:**

The administrator should be able to send notifications to students or supervisors that are related to one project. The administrator has to specify to which project the notifications are sending to, and whether the notifications are for students/supervisors or both.

**Post-condition:**

The notifications sent by the administrator is saved to the database.

**Exception:**

If the administrator does not specify to which project the notifications are targeted, the notification will not be sent and not saved to the database.

### 2.2.4 Use Case: Trace Funding Details

**Precondition:**

The funding details are available in the database of the research student manager. The administrator has logged in to the system with valid credentials.

**Description:**

The administrator should be able to check funding details including the source, purpose and amount of the funding in all departments.

**Post-condition:**

The administrator is able to view funding details for all the projects.

**Exception:**

If the funding details are not entered to the database, the administrator won’t be able to view it.

### 2.2.5 Use Case: Manage Project Assessment

**Precondition:**

The administrator has logged in to the system with valid credentials. And the project has an assessment score and assessment stage that needs to be updated.

**Description:**

The administrator should be able to update project assessment score and assessment stages.

**Post-condition:**

The assessment information is modified successfully.

**Exception:**

If the assessment score entered is not valid numeric value, it won't be saved to the database.

### 2.2.6 Use Case: Manage Supervision

**Precondition:**

The administrator has logged in to the system with valid credentials. And the supervisor of the student may need to be changed.

**Description:**

The administrator should be able to change the supervisor for a student.

**Post-condition:**

The supervisor of a specific student is modified successfully.

### 2.2.7 Use Case: Enrol student to Project

**Precondition:**

The administrator has logged in to the system with valid credentials. And the administrator needs to enrol a specific student to one of the projects.

**Description:**

The administrator should be able to enrol a student to one of the projects.

**Post-condition:**

The student is successfully enrolled to a project allocated by the administrator.

**Exception:**

If the administrator tries to enrol a student to a project that the student already exists, it will fail because the student is already in that project.

## 2.3 Technical Terminologies

The research student manager is determined to be able to run under LAMP architecture, which is short for Linux, Apache, MySQL and PHP. LAMP is a set of open source server and applications that can be used to build active websites. This architecture can fully satisfy the requirements for the research student manager built in this project. The following are short descriptions of the technology terminologies as well as the benefits of using.

### 2.3.1 Linux

Although the research student manager in this project is developed and tested under Windows environment, it can be easily transplanted to a Linux operation system. Linux is a free, open source operating system. It features its open source community, its high security level and the little cost.

### 2.3.2 Apache

Apache is the name for the web server application used for deploying all the website contents in this project. It is the most popular web server application used on the Internet. The benefits of using Apache is its fast and stable performance, its trans-platform feature and the fact that it is also open source. By installing PHP module, Apache can perfectly run PHP pages.

### 2.3.3 MySQL

MySQL is the database system used for this project. It is a database management system that supports SQL (Structured Query Language). The benefits of using it is its fast and flexible performance, and it's a stable and open source database solution.

### 2.3.4 PHP

PHP (Hypertext Processor) is a server side scripting language that is HTML embedded, which means it can be simply written inside HTML web pages and executed on server side. The most important benefit of using PHP is that it works perfectly and efficiently together with MySQL. And also that it is a simple but high-performance scripting language.

### 2.3.5 HTML

HTML is a mark-up language used to write the web pages in the research student manager. It is readable by web browsers to display HTML documents as web pages.

# Chapter 3: Design and Implementation

In this section, the entire process of designing and implementation of the research student manager will be discussed. Firstly, the findings from requirement gathering will be organized into the functional and non-functional requirements that provide as functionality specifications for design. Secondly, the finished design work of database and the web site pages will be discussed in detail. Finally, system testing and evaluation results will be presented.

## 3.1 Identifying Functional and Non-functional Requirements

The requirement findings results and the use cases derived previously have provide the foundations for determine the actual functionalities for the Research Student Manager. In this section, these functionalities are described as functional requirements. There will also be some non-functional requirements we derived from literature review and the interview.

### 3.1.1 Functional Requirements

#### *3.1.1.1 Displaying student Information and details*

The research student manager should have a module that is able to display student basic information of all the students in a table. This information will including*: Name, Student Number, Date of Birth, Gender, Program, and Email*. The system should display the results in order of the student number. It should support filtering student by program of study, and support searching student by their name or student number. The system should also allow the administrator to click on the student name and then open a pop up window with detailed information about that student,

#### *3.1.1.2 Displaying supervisor information and details*

The research student manager should have a module that is able to display basic information of all the supervisors in a table. This information will include: *Name, Supervisor ID, Faculty, Title, and Research Interest*. The system should display the results in order of the supervisor ID. It should support filtering supervisor by their faculty, and support searching supervisor by their name or supervisor id. The system should also allow the administrator to click on the supervisor name and then open a pop up window with detailed information about that supervisor,

#### 3.1.1.3 Managing various assessment stages

The research student manager should have a module that is able to display current stages of the projects including*: Proposing, Early Term, Mid Term, Final Term, Dropped and Completed*. The system should also allow the administrators to change the current stage of any specified project. Meanwhile, the system should display all the evaluation scores for the three terms on the same page, and allow the administrator to update the scores in percentage scale.

#### 3.1.1.4 Managing supervision status of a student

The research student manager should allow the administrator to change a supervisor of a specific student. This can be done with a page that allows the administrator to change supervisor of any student or in a pop up window when the administrator clicks the name of any student. The system should then reflect changes and make it viewable to the administrator.

#### 3.1.1.5 Manage student status in projects

The research student manager should have a module that is able to display all the project that a student is currently working in. And it should also display in a project detailed page of all the students that are currently working in that project. The administrator should be allowed to enrol student to a specific project or remove student from a specific project that the student is currently working in.

#### 3.1.1.6 Keeping track of funding details

The research student manager should have a module that is able to display funding details information including: *Funding Source, Funding Purpose and Funding Amount.* This should be corresponding to each of the projects

#### 3.1.1.7 Send notifications to supervisor/students in projects

The research student manage should have a module that allows the administrator to send notifications to students or supervisors or both that are associated with a specific project. The administrator should be allowed to specify which of the project to send the notifications to, and input the contents, then decide whether or not the notification is for students or supervisors or both.

#### 3.1.1.8 More extra functionalities

Besides the basic functionalities, there can be more functionalities determined by further project requirement and design refining. In this section we discuss some of these extra functionalities that can be further implemented to the system as time permits.

#### 3.1.1.8.1 Different levels of access permissions

As the research student manager now supports administrator logging in, it can also support multiple users to log in with different levels of access permission. For example, if the user logs in as some "project coordinator" role, the user may only have permission to view and change project status and assessments, while they have no access to view all the detailed information about all the students and supervisors.

#### 3.1.1.8.2 Organizing Assessment Sessions

The research student manager may also have a module that allows for organizing assessment sessions. The administrator may use this module to schedule an assessment session and announce it to students and supervisors through notifications.

### 3.1.2 Non-functional Requirements

Apart from all the functional requirements that the research student manager must be able to satisfy in order to deliver all the required basic functionalities, the system should also satisfy some non-functional requirements in terms of usability, performance and security.

#### *3.1.2.1 Usability Requirements*

The research student manager has to be easily usable by the administrator. It should be designed to facilitate the work efficiency of administrative tasks so the administrator should need only very little help in using every module of the system. All the functional modules should be easy to learn and easy to use.

#### *3.1.2.2 Performance Requirements*

The research student manager should have a good performance in loading web pages on the website and updating data in the database. The administrator should not wait long than normal for a task or action to be completed.

#### *3.1.2.3 Security Requirements*

The research student manager should have an adequate level of security. Session should be maintained to allow the system used by only users who has logged in with valid credentials. The user credentials should be stored safely in the database. Also the log in module should never be bypassed by malicious users.

## 3.2 Database Design and Implementation

Based on the functional requirements that have been identified in previous sections, the database entities have been identified in order to determine what information and data will be stored in the database. Then, the each of the entity attributes and their relationship are determined. And a complete entity relations diagram is then drawn to show these findings. Finally, the ER diagram is mapped to a relations model, which can be directly used to implement the database.

### 3.2.1 Identify Entities and Entity Attributes

The first step is to identify all the entities in the database that stand for the data subjects needed in the research student manager. The identified database entities for the research student manager are: *(1) Student, (2) Supervisor, (3) Project, (4) Notification and (5) Administrator*.

After all the entities have been identified, the next step is to identify all the attributes for each entity and their data type. The attributes for all the entities are described as the following.

**Student:**

Name (*String*), Gender (*String*), Student Number (*Integer*), Birth Date (*Date*), Program (*String*), Email (*String*), Supervisor ID (*Integer*)

**Supervisor:**

Name (*String*), Supervisor ID (*Integer*), Faculty (*String*), Title (*String*), Email (*String*), Research Interest (*String*)

**Project:**

Project ID (*Integer*), Project Status (*String*), Funding Amount (*Float*), Funding Source (*String*), Funding Purpose (*String*)

**Notification:**

Project ID (*Integer*), Message ID (*Integer*), Date & Time (*Date Time*), Content (*String*), To Supervisor (*Boolean*), To Student (*Boolean*)

### 3.2.2 Identify Entity Relationship

After all the entities and their attributes are specified, the relationship between the five entities is then specified as the following. This specification is based on the system requirements of the research student manager.

- A supervisor can instruct none or multiple students, while a student must be instructed by one only one supervisor.
- A student can work in none or multiple projects, and a project can have one or more student working in it.
- A supervisor manages zero or more project, while a project is managed by one or more supervisor.
- A notification must notify one and only one project, while a project can be notified by multiple notifications.

### 3.2.3 Entity Relations Diagram

Based on the identification of all entities, entity attributes and all the relationship. The following Entity Relation Diagram shown in Figure 2 can be drawn to put everything together. Notice that all the entities (in solid rectangle shape), the entity attributes (in oval shape) and the relationship (in diamond shape) are shown in the ER Diagram.

The primary key of each entity is underlined, for example, Student has the attribute “Student Number” as its primary key, and Project has “Project ID” as its primary key. The foreign key attribute is specified in a different colour in each of the entity that has a foreign key attribute. The participation relationship between entities is specified in the line that connects both entities and their relationship. The bolded line indicates total participation, meaning all the entity instances must participate in the relationship. For example, all the notification must have one and only one project that it intends to notify, and all the students must be instructed by one and only one supervisor.

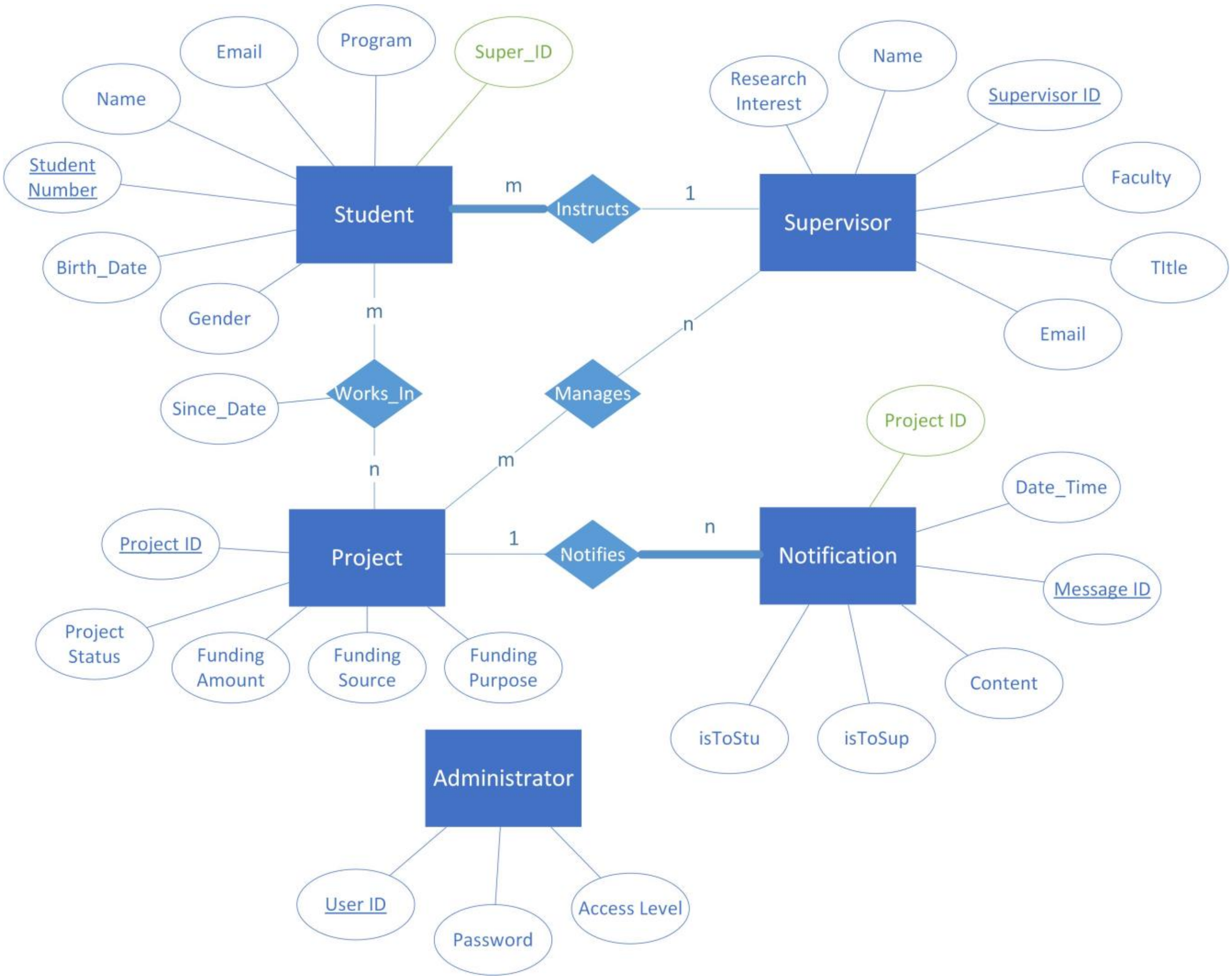


**Figure 2 Entity Relationship Diagram for Research Student Manager**

### 3.2.4 Relations Model mapped from ER Diagram

After the ER diagram has been drawn, the database design is partially complete. However, to make it easier for database implementation, the next step is to map the Entity Relations Diagram to the Relations Model that can be used directly to implement the database.

The Figure 4 (at the end of this chapter) shows the Relations Model that has been mapped from the ER Diagram, all the Entities and their attributes are shown in the Relations Model. As for the relationship, all the one-to-many relationship are specified by putting the primary key of the entity on the “one” side as a foreign key in the attributes of the entity that is on the “many side. For example, the Supervisor ID is put in Student table as a foreign key. And the Project ID is put in the Notification table as a foreign key.

For each of the many-to-many relationship, a new table of relationship is created to map the reference between the two entities. For example, the Works_In table and the Manages table are used to specify the many-to-many relationship between students and projects, supervisors and projects, respectively.

The mapping of such a relations model from the ER diagram makes it easier for database implementation because the tables in the Relations Model can be directly created as tables in the database, and the foreign key constraints and relationship can be directly translated and defined during database implementation.

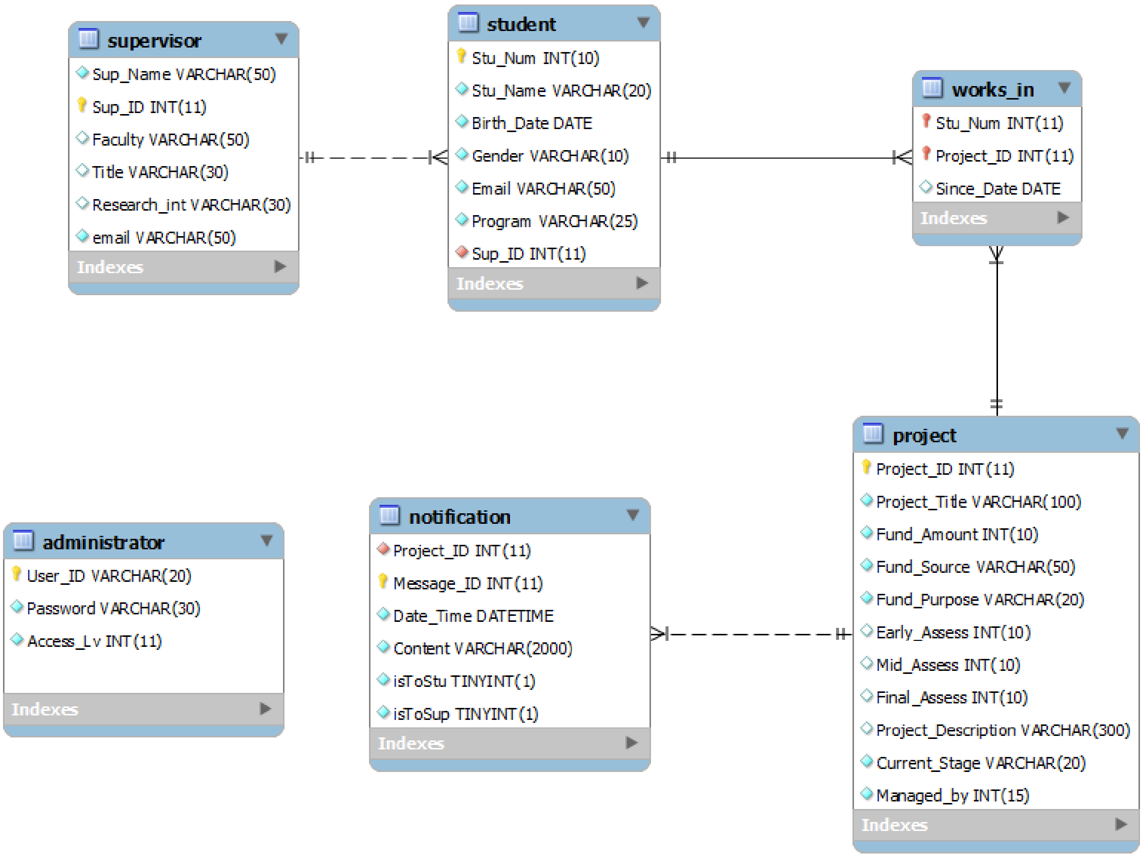


**Figure 3 Database Class Diagram of Research Student Manager**

### 3.2.5 Database Class Diagram

Based on the design of the database entity relationship diagram and the mapped Relations Model, all the entities of the database and their relationship can then be drawn into a design class diagram with the help of MySQL Workbench. MySQL Workbench (Community Edition) is a very powerful tool that can be used to directly export database (including all entity attributes and relationship) into SQL scripts that can be used for database implementation. Figure 3 shows the class diagram drawn in MySQL Workbench.

With this tool, all the data types of the attributes of the database entities can then be specified as well as the length or range of all the data types. And the class diagram is also able to specify all the relationship, including foreign key constraints and participation constraints between all the entities in the database.

### 3.2.6 Database Implementation

Based on the design class diagram drawn for the research student manager and the MySQL Workbench tool, all the SQL queries that can be used to create the research student manager database can then be automatically generated and refined. The SQL queries are then used to build the database, all the tables and their columns in the database. The phpMyAdmin tool is also used as an assistance during database implementation.

After all the tables are implemented in the database, the table rows (data entries) are then added to the tables with the help of phpMyAdmin. Notice that these data input are randomly generated data that are used for just testing the functions of the research student manager. We do this to input data because we assume that all the data has already been available in the database for the research student manager. (See 3.4.2 System Testing Assumptions)

The implementation results of the database, i.e. some screenshots captured in the MySQL command line will be presented in the Chapter 4 Section 4.1.1.

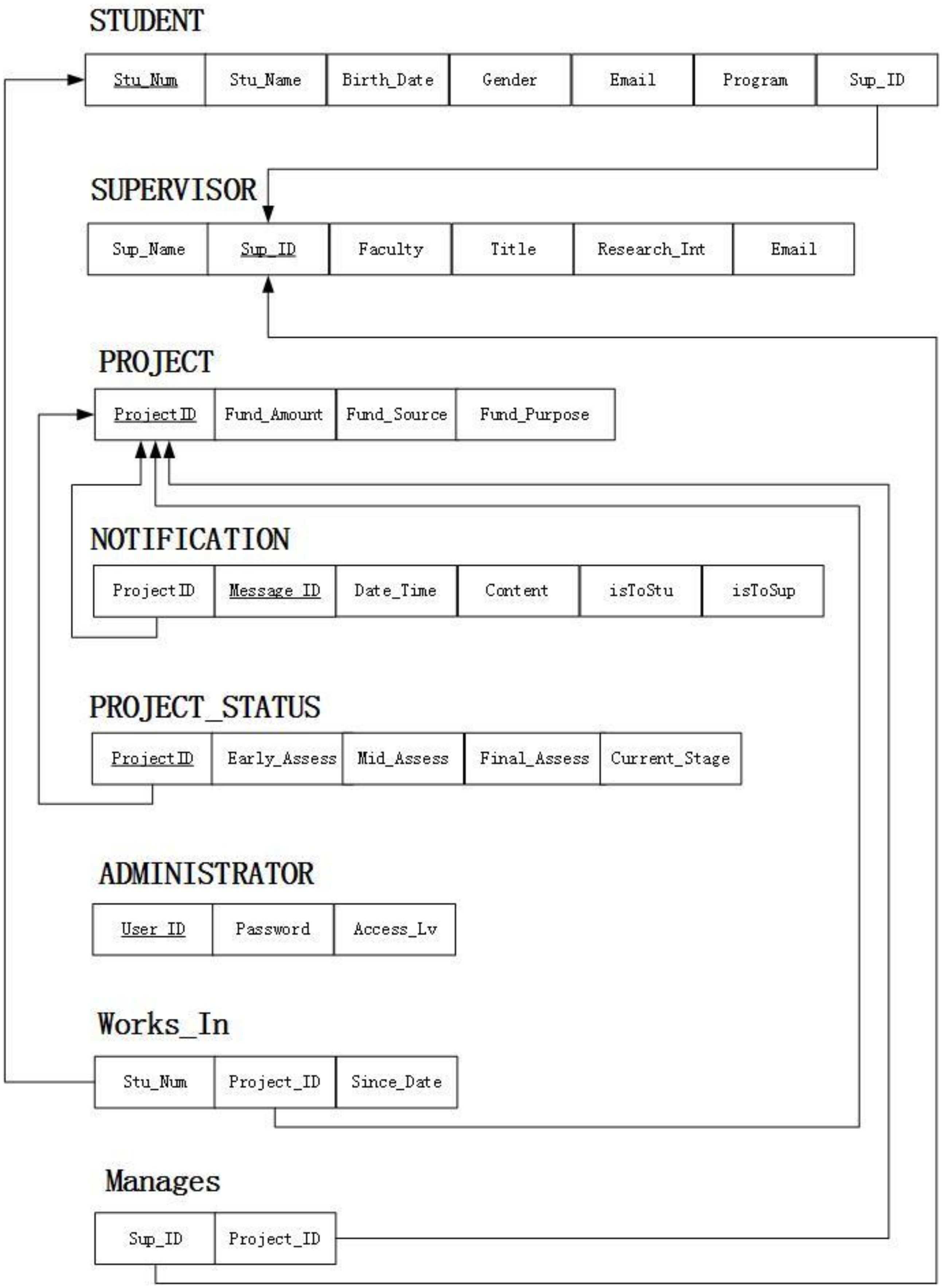


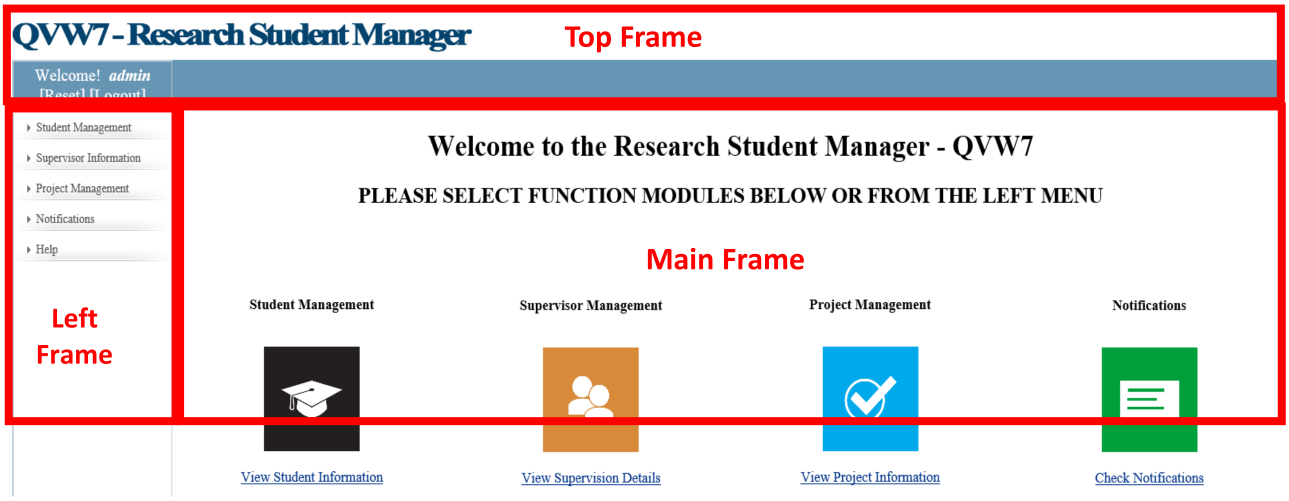
**Figure 4 Relations Model Mapped from ER Diagram**

## 3.3 Website Design and Implementation

The database is the back-end of the research student manager that manage the storage and manipulation of all the data. The website pages are the front-end side of the system that directly interacts with the user and is used by the user to access data from the database. After the database has been implemented, the web pages are then designed and implemented. The website pages are designed in a style that combines HTML pages with PHP code.

### 3.3.1 Webpage Style Design of the Website

Most static contents on the research student manager will be written in The HTML web pages are the contents written in HTML and can be readable by web browsers as the actual web pages that are displayed to the users. In the research student manager, the HTML web pages have been designed to be multi-framed.



Welcome to the Research Student Manager - QVW7

PLEASE SELECT FUNCTION MODULES BELOW OR FROM THE LEFT MENU

**Figure 5 Multi-framed view of the Research Student Manage**

- The left frame is the main menu and can provide access to all the functional modules of the student management system. This will always appear as administrator has logged in.
- The top frame shows the logo and banner of the system, and is designed to have some commonly used features like showing the current time and some quick launch buttons. This will also always appear as administrator has logged in.
- The main frame is the main area for system contents like tables with list of student or supervisor information and forms for manipulation data. This entire area will change depend

on the functional module chosen by the administrator.

The benefits of a multi-frame design is to utilize the screen that interacts with the users more efficiently, and enable the administrator to navigate quickly through each of the functional modules. Setting the main frame as the only area where different web pages in different modules are loaded can also reduce the need for refreshing the entire web page.

Overall, there are also some simple CSS files to control the overall style of the HTML pages. CSS files are used to define the style and appearance of the web pages for the benefits of reusing the defined style and separating "the presentation style" from the actual contents of the web page, which has made it much easier to manage the over style of the website.

### 3.3.2 Design of the Website Navigation

After the main interface of the webpage has been designed to be multi-framed, each of the modules are the designed and the site map of the research student manager website has been plotted out. The Figure 5 shows a site map of all the functional modules and subpages of the research student manager. There are five modules in the research student manager: Student Management, Supervisor Information, Project Management, Notifications and Help.

#### *3.3.2.1 Student Management*

This module is focused on the display of all the detailed information about all the students as well as the management of all supervisor and project information that is related to research students. There are three subpages: (1) Student Information, which displays basic information of all students, (2) Supervision Details, which shows all students and detailed information about their supervisors, and (3) Student Project Details, which shows all the projects that each student is currently working on. The each module have links that can be clicked on to open popup windows as indicated in Figure 5.

#### *3.3.2.2 Supervisor Information*

This module is focused only on the display of all the detailed information about all the supervisors. There are only one sub module which is called Supervisor Details, and it can display detailed information of all the supervisors. It also has links to popup windows.

### 3.3.2.3 Project Management

This module displays and manages all project related information. There are three sub modules: (1) project details, which displays detailed list of all projects, (2) project assessment, which displays all assessments scores and stages of all projects, and (3) funding tracking, which displays all funding details of all projects. Each module have links to a popup project details window, and in the assessment module this can be used to update assessments.

### 3.3.2.4 Notifications

This modules displays and manages all history notifications that have been sent by the administrator. It has two sub modules: View all notifications, which displays all history notifications sent by the administrator, and Manage Notifications, which can be used to remove notifications and send new notifications.

### 3.3.2.5 Help

This module is simply a user manual that tells the administrator how to use each of the functional modules of the research student manager.

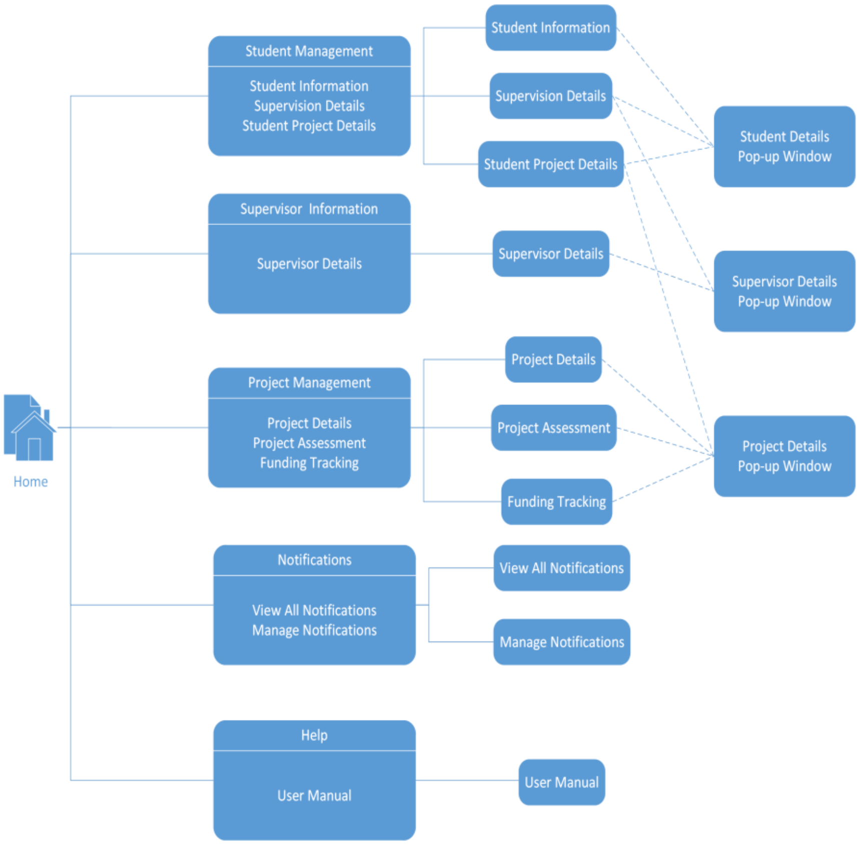


**Figure 6 Website Map of Research Student Manager**

### 3.3.3 Class Diagrams of Website Subsystems

After the HTML web page style design has been settled, all the web page elements in the research student manager system is then analysed and designed into the class in the subsystems. As described previously, the research student manager consists of five main subsystems: (1) Login Authentication (2) Student Management, (3) Supervisor Information, (4) Project Management, () Notification Management and. In this section, all the php files that correspond to these five subsystems will be discussed.

#### 3.3.3.1 *Login Authentication Subsystem*

The login authentication subsystem as shown in Figure 7 consists of two files, the login.php which is the user interface that interacts with the user and accepts username and password from the user, and the login_authenticate.php file, written in pure PHP code, which handles username and password passed by login.php and performs SQL queries to the administrator table in the database in order to authenticate if the username and password pair will be valid. If the pair is valid, it will further direct user to the research student manager system.

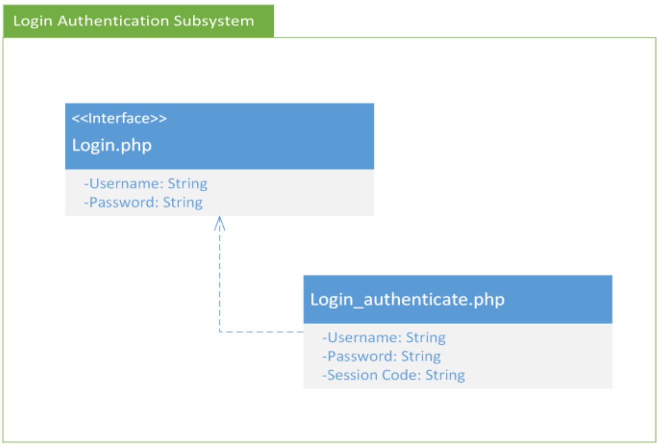


**Figure 7 Class Diagram - Login Subsystem**

#### 3.3.3.2 *Student Management Subsystem*

The student management subsystem, as shown in Figure 8 consists of the four interface class: student_information, student_supervision, student_project and student_details. In each of these interface classes, some columns of the student information or related supervisor and project information from the database are displayed as shown in the class diagram. The student_details.php is the contents of a pop up window that is opened by clicking on a student name. This window includes all the detailed basic information, the supervisor and project currently working in of that student. This pop up window is achieved by passing the student number from the parent window to the student details interface class through GET method by appending it at the end of the URL.

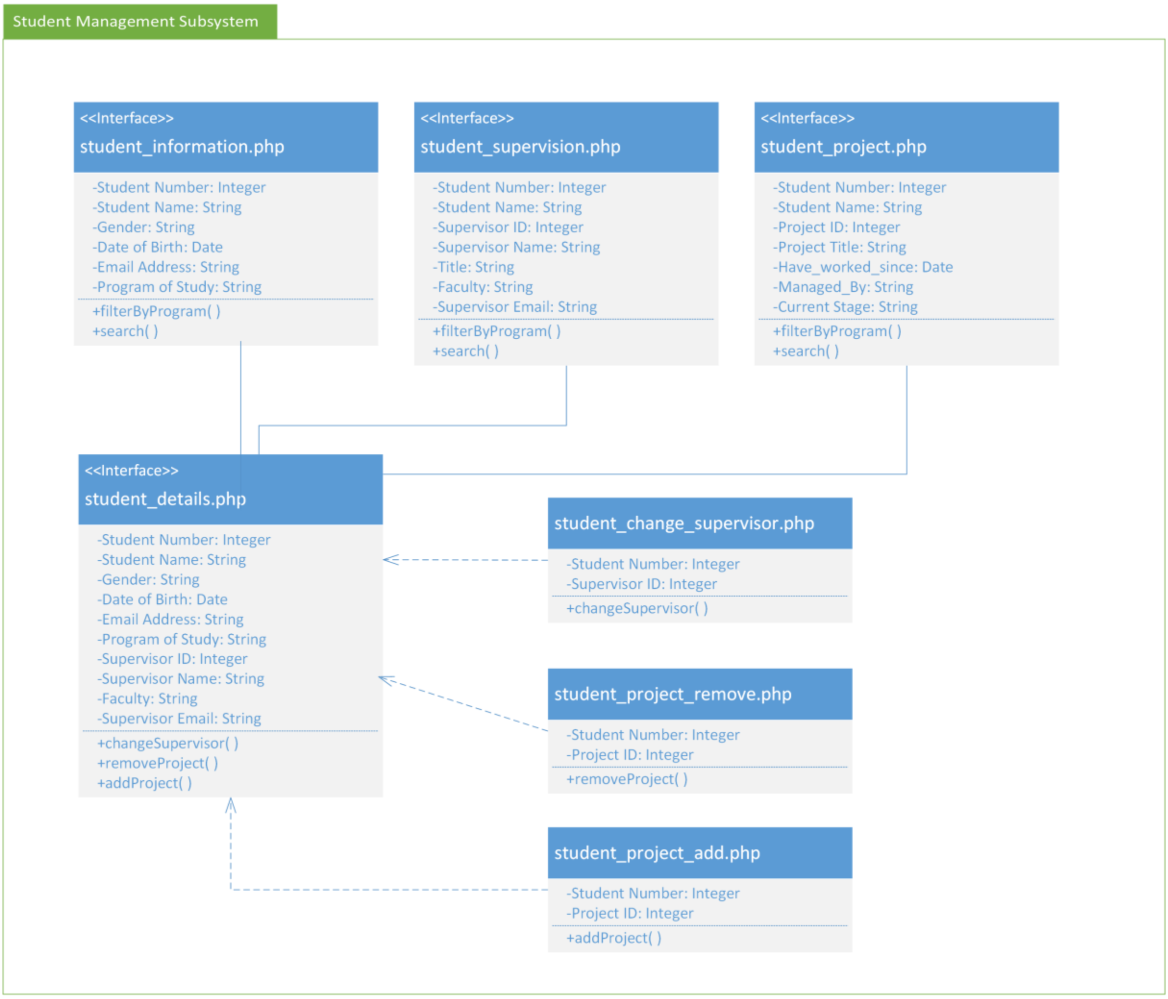


**Figure 8 Class Diagram - Student Management Subsystem**

There are also three control classes associated with the student_detail.php class, which are responsible for handling requests from the student details interface and execute SQL queries in order to change supervisor for a student (student_change_supervisor), add student to a project (student_project_add) and remove student from a project (student_project_remove).

### *3.3.3.3 Supervisor Information Subsystem*

The supervisor information subsystem consists of two interface classes: the supervisor_information for displaying a table view information of all supervisors, and it supports filtering supervisors by faculty and searching by name or id. The supervisor_details class provides a detailed view of that supervisor and shows all the students that the professor instructs and all the project that the supervisor is managing. This pop window works the same way as the student details window mentioned by rewriting the URL with supervisor id appended at the end.

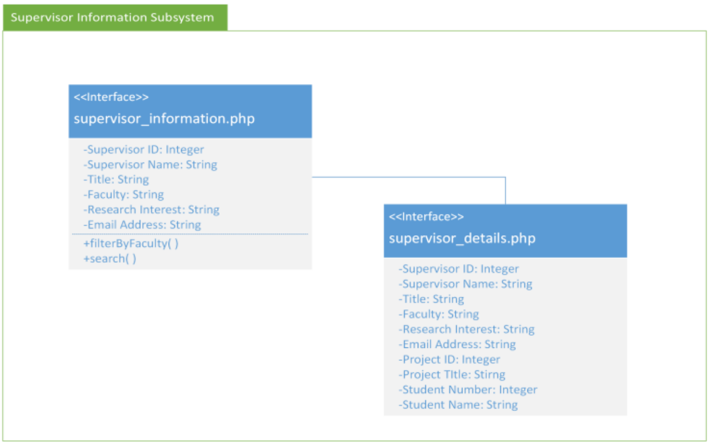


**Figure 9 Class Diagram - Supervisor Information Subsystem**

### 3.3.3.4 *Project Management Subsystem*

The project management subsystem as shown in Figure 10, consists of three main interface classes, the project_information, the project_assessment and the project_funding. They each display some project related information as shown by variables in the class, and support filtering and searching. There is also a project details pop up window accessible by clicking on the project title. It displays

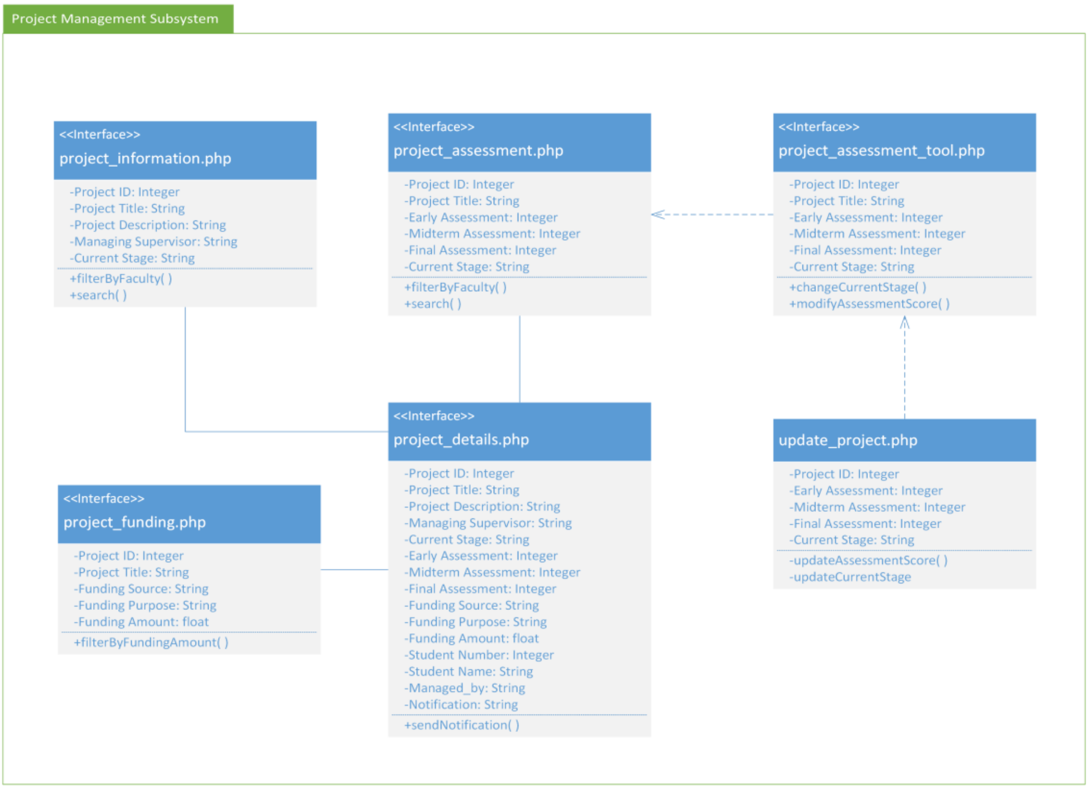


**Figure 10 Class Diagram - Project Management Subsystem**

detailed information about the project details, a list of student currently working in the project, and some information about the supervisor that manages the project. The project_details interface is associated with all these three main interface classes because it can be accessed simply from these three main interfaces, this project_details interface shows all information about a specific project. There are also another interface class that is part of the project_assessment interface, called the project_assessment_tool, which is used to update project assessment score and change project current stage. A control class is associated with this tool to handle the requests posted from the interface and perform SQL queries to the database to complete the tasks of updating data values.

#### 3.3.3.5 Notification Subsystem

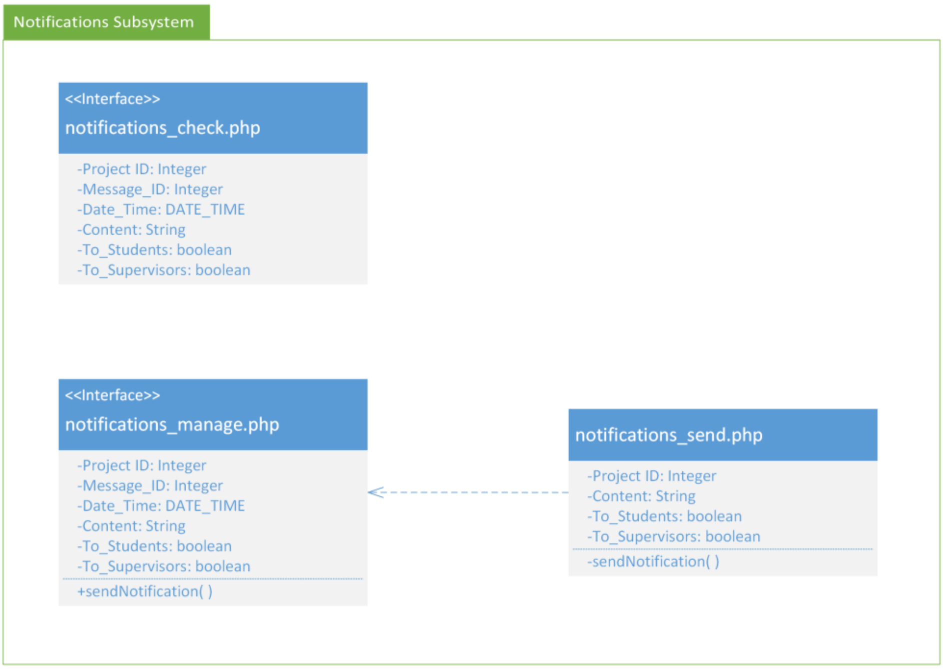


**Figure 11 Class Diagram - Notification Subsystem**

The notification subsystem as shown in Figure 11, consists of two interface classes: the notificaitons_check which displays with a table view of detailed information about all notifications, as shown in the diagram. And the notifications_manage interface which has a form that allows administrator to enter the content and targeted receiver (by choosing a project) of the notification. There’s a control class called notification_send that receives requests posted from the notificaitons_manage interface and access the database with related SQL queries to post the notification and save it to the database.

#### 3.3.3.6 Other System Classes

Apart from the above mentioned subsystems, interface classes and control classes. The system also has some common classes that are accessed by all the subsystems. These include tasks with database access and session control. An overview of how the relationship between these classes and all the subsystems of the research student manager can be seen in Figure 12. The session control class is responsible for verifying and maintaining the session control of currently logged in user. The database connection control class include the parameters needed for accessing the database, and have functions that are used to build database connections. These two controls are included in most of the classes of the website wherever session control and database connection is needed.

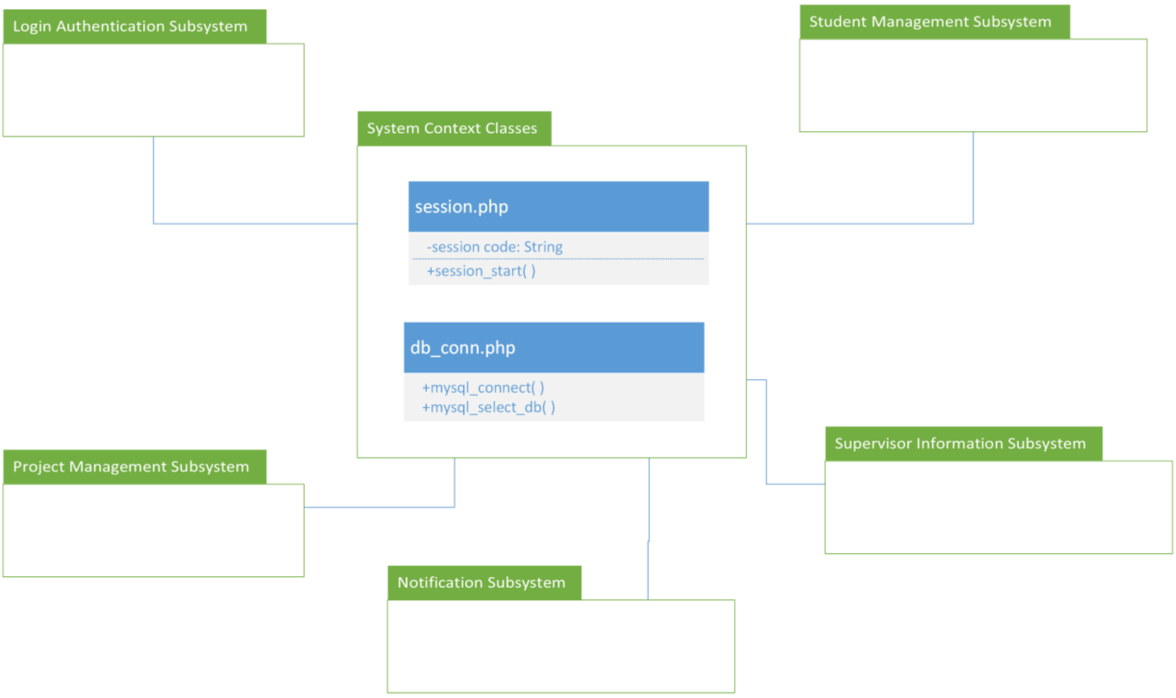


**Figure 12 System Context Classes and All Subsystems**

### 3.3.4 Website Architectural Design

As discussed in this chapter, the research student manager will have several subsystems accessing resources from the database, display information on each of the subsystems and manage data change via queries to the database. Therefore, the overall architectural design of the website can refer to the Repository Architecture where all the subsystems act as the surrounding terminals that can access to the central repository, which is the single database of the system used to store all data.

### 3.3.5 HTML and PHP Code Implementation

With the above established analysis and design work of the research student manager, the implementation of actual HTML and PHP pages can then begin. To simplify the work on the user interface of the website, the HTML web pages are mostly adapted from some online templates that are designed with friendly looking user interfaces for online content management systems. These templates have some already written JavaScript code that enhances user interaction. However, most of the website still conforms to a simplicity design for the research student manager.

After all the HTML web pages have been written. PHP code is then written inside the HTML web pages in order to retrieve and manipulate data from the database. The feature of PHP allows its code to be executed on the server side and then returned still as simple HTML language to the client web browser. This has made it possible to embed PHP code into HTML pages without influencing other contents that exist in that file. The PHP code in the research student manager is used in the following ways: Database Access, Session Control, and Form Handling.

#### *3.3.5.1 PHP code for database Access*

The main reason to write PHP code inside the HTML pages is to retrieve data from the database and print them on the web page. As database configurations and credentials need to be used every time a database connection is made, the database configurations (host name, database name) and credentials (database user name and password) are stored in separate PHP file (named db_conn.php). This file is included in other PHP files whenever database access is needed.

Then, PHP code for retrieving information from the database is embedded in the HTML web pages that show that information. The SQL queries are written in the PHP code and executed during runtime at the server side to run in the server database. To better format the data from the database, a loop in PHP is written to format every row fetched from the database into a row in a table of the HTML pages.

#### *3.3.5.2 PHP code for session control*

Another reason to use PHP code is that session control is supported very well by PHP. Typically, there're two ways of session control: cookies, which refers to writing session data into a file stored in the client computer, or URL renaming, which appends user identification at the end of URL and is ideal for client that does not support cookies. In the research student manager, cookie is used for session control.

The session is first opened during login authentication process where a separate PHP file (login_authenticate.php) is written to authenticate user credentials. If user credentials are valid, a new session will be started and session information will be written to cookies. Another separate PHP file (session.php) is also written with the only simple purpose to check if the session information exists. If it does not, meaning the session has not been started and the user has not logged in, the user will be directed to the login page. This session.php is included in all the functional modules to make sure the user has logged in before accessing any functional modules.

#### *3.3.5.3 PHP code for form handling*

PHP code is also used for handling all the form data posted in the research student manager. The forms written in HTML pages is able to post the name-value pairs entered by the user to a targeted PHP file via POST method. The PHP file for form data handling will then retrieve the posted data, and use it to update data in the database.

For example, in the Project Information web page there's a form that allows users to input assessment scores and update project status. The name and value pairs of this form is then posted to a separate PHP file update_project.php to make UPDATE queries to the database with the data values in order to manipulate project data. There're also some other similar PHP files like send_notifications.php.

A very important point to notice is data validation. This is achieved by writing some JavaScript functions on the HTML file and execute them every time the user tries to submit form data. If the input data is invalid, the users will be alerted to input valid values, and only when the values conforms to the rules pre-defined in the database can they be used by the PHP file to update entries in the database. The reasons and benefits for using JavaScript for data validation is that one of the main feature of JavaScript is with checking and alerting users at the front-end when input is not valid before any data is submitted to the server.

## 3.4 System Test and Evaluation

Through the process of implementation, system testing is periodically carried out by deploying the web contents on a local web server, linking them with the database and evaluating if all the functionalities work well. This is a very important task for the final stage of the project. An evaluation of the system is also carried out in terms of whether all the system requirements are met, whether all the functionalities are working properly and what needs to be further improved.

### 3.4.1 System testing environment

The research student manager is deployed on a computer that runs Windows 7. The HTTP server application used is Apache, and the database used is MySQL. The testing is done by loading all the wen contents into the Apache server and access the website through web browser on the local host computer.

### 3.4.2 System testing assumptions

In the testing of the Research Student Manager, a few very important assumptions have been made about the research student manager system. The first assumption to make here is the data assumption. The student, supervisor and project entries and their information are assumed to be already existing in the database. These information are assumed to be entered by the administrative staff at the university other that those staff that manage the research students, which is why the functions of adding and removing students, supervisors or projects are not enabled in the research student manager, though this is very easy to achieve. To realise data input in this project, the test data of students, supervisors and projects are entered using a web based database management system (DBMS) called “phpMyAdmin”. This is a full function DBMS that supports create, read, update and delete of any data entries in the database.

Another assumption which is partly based on the above assumption, is that there will definitely be many other information management systems in a much larger scope that are used by the other administrative staff at the university to handle personal information related to students and supervisors, and project information. Most of such must be some legacy system of the university that has been used for years, so the research student manager will consistently work together with these systems. For example, a personal information management system may be used to manage all the personal information of students and supervisors. And a project management system may be used to create and remove project by the project managers of the university. The synergy of these systems consist an entire environment for the management of information in the university.

### 3.4.3 System Test Results and System Evaluation

The test in this section and as well as discussed in this chapter of design and implementation will mostly focusing on testing that each of the subsystems work properly with their HTML, PHP and JavaScript code that have been written. Most of these will be “white box” testing of the system technical aspects, in which the focus is more on if the functions give the right results. In later sections in the Results and Discussion chapter, another testing will be conducted from a more overview level (black box testing) to see, for example, if the system satisfy the requirements well, or if the system is efficient enough and facilitates automation of student management tasks. To make the test results concise and clear, the results of each test cases are presented in a table.

In this section, the test case results of the system logic testing is presented. The goal of testing system logic of each of the subsystems of the research student manager is to verify if the functions implemented will produce the expected valid results.

**Table 1: Test Results of System Testing**

| Case # | Test Case | Pass or Fail | Comments |
|---|---|---|---|
| 1.1 | The login window can check if username and password is empty and give alert | Pass | |
| 1.2 | The login window can verify through querying the database to check if the use name and password entered is valid | Pass | Minor bug: when the username and password do not match that in the database, the alert dialog will cause to stay on the login authenticate page. |
| 1.3 | The login window will reset password if the administrator forgets the credentials | Fail | This function hasn't yet been implemented due to time constraints |
| 2.1 | The student management module gives the right student records in a table. | Pass | |
| 2.2 | The filter and search functions will produce the right results of student records. | Pass | Minor bug, when user selects the filtered program, the drop box will not stay on that program but change back to “Please Select” instead |
| 2.3 | The student names, supervisor names and project names are clickable and lead to a detailed information page | Pass | |
| 2.4 | The student details work properly and can change the supervisor of a student | Pass | |
| 2.5 | The student details can be used to assign student to a project or remove student from a specified project | Pass | Minor bug: the system is not able to determine if the new project added already existed and give warnings, it will simply ignore adding already existing projects |
| 2.6 | The student details show a list of the project that the student is working in and show the supervisor of the student | Pass | |
| 3.1 | Supervisor information module is able to show a table of all supervision information | Pass | |
| 3.2 | The filter by faculty and search function in | Pass | Minor bug, when user selects the |

| | | | |
|---|---|---|---|
| | the supervisor information module works | | filtered program, the drop box will not stay on that program but change back to “Please Select” instead |
| 3.3 | The supervisor details page works well and show all students instructed by the professor and show the project the supervisor is managing | Pass | |
| 4.1 | The project management module can show the project information, with respect to the three different sub modules. | Pass | |
| 4.2 | The project details page is able to show project detailed information of each project | Pass | |
| 4.3 | The project detailed page can be used by the administrator to send notifications to the students/supervisors related to a project | Pass | Minor improvement needed: currently the history notifications does not shown what students/supervisors the notifications are targeted at. |
| 5.1 | The check notification sub module is able to see a list of all history information | Pass | |
| 5.2 | The send notification tool can be used to send notification to project, and remove notification from the history. | Pass | Improvement needed: the removing notification function may also need to be available in the project details page |
| 6.1 | The help manual is able to give users a list of instructions on how to use each subsystem modules | Pass | Only thing is that the manual is not very interactive. |

Based on these test results, it is sufficient to determine that the functionalities of the system are fully and properly implemented. All the functionalities can produce the right results and all the modules work well. This would basically meet the functional requirements that are pre-defined in the project specification of the research student manager project.

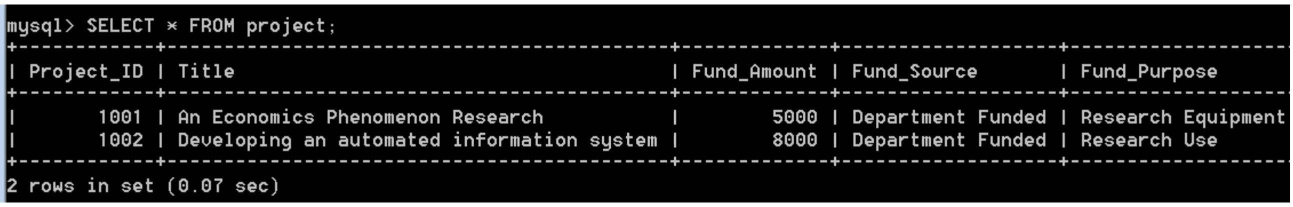
# Chapter 4: Results and Discussion

In this chapter, the completed research student manager built in this project is demonstrated. All the functional modules of the research student manager through some screenshots along with some discussions. The test outputs of system testing is also discussed. Some user reviews from some random users is also cited at the end of this chapter.

## 4.1 The Complete Research Student Manager

### 4.1.1 Screenshots from the Database

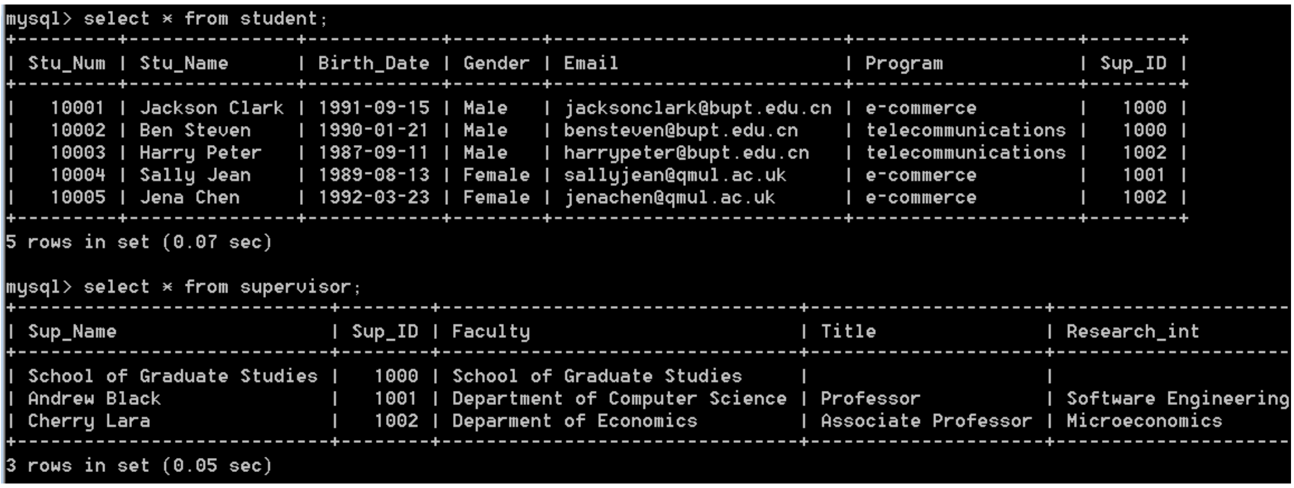
In this section a complete research student manager is shown via some screen shots and discussion. First of all, some screen shots of database from MySQL command line are presented. Figure 13 shows the result of selecting all the rows from the project table. This include a list of all the project information.



```
mysql> SELECT * FROM project;
+------------+--------------------------------------------+-------------+-------------------+-------------------
| Project_ID | Title                                      | Fund_Amount | Fund_Source       | Fund_Purpose
+------------+--------------------------------------------+-------------+-------------------+-------------------
|       1001 | An Economics Phenomenon Research           |        5000 | Department Funded | Research Equipment
|       1002 | Developing an automated information system |        8000 | Department Funded | Research Use
+------------+--------------------------------------------+-------------+-------------------+-------------------
2 rows in set (0.07 sec)
```

**Figure 13 Screenshot of selecting all rows from the Project table**

Figure 14 shows the results of selecting all the rows from the student table and the supervisor table. This shows lists of all information about students and supervisors.

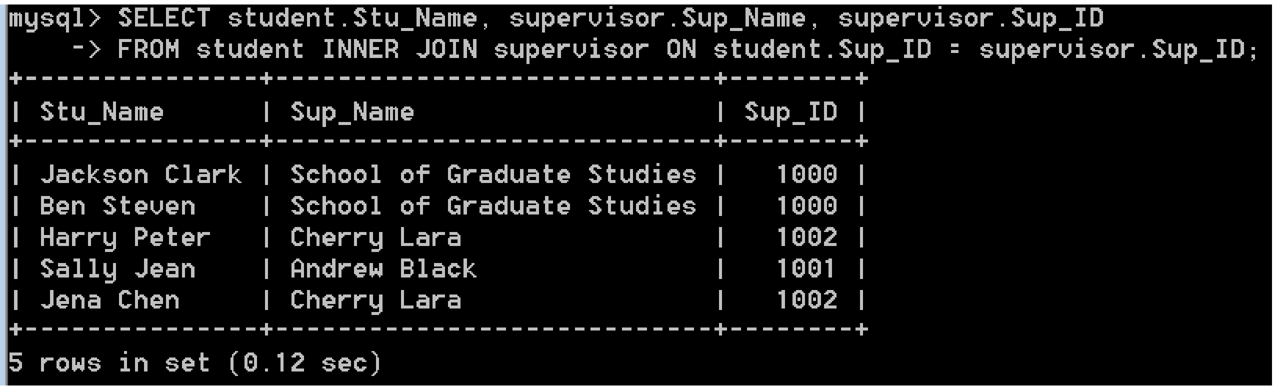


```
mysql> select * from student;
+---------+---------------+------------+--------+--------------------------+--------------------+--------+
| Stu_Num | Stu_Name      | Birth_Date | Gender | Email                    | Program            | Sup_ID |
+---------+---------------+------------+--------+--------------------------+--------------------+--------+
|   10001 | Jackson Clark | 1991-09-15 | Male   | jacksonclark@bupt.edu.cn | e-commerce         |   1000 |
|   10002 | Ben Steven    | 1990-01-21 | Male   | bensteven@bupt.edu.cn    | telecommunications |   1000 |
|   10003 | Harry Peter   | 1987-09-11 | Male   | harrypeter@bupt.edu.cn   | telecommunications |   1002 |
|   10004 | Sally Jean    | 1989-08-13 | Female | sallyjean@qmul.ac.uk     | e-commerce         |   1001 |
|   10005 | Jena Chen     | 1992-03-23 | Female | jenachen@qmul.ac.uk      | e-commerce         |   1002 |
+---------+---------------+------------+--------+--------------------------+--------------------+--------+
5 rows in set (0.07 sec)

mysql> select * from supervisor;
+----------------------------+--------+--------------------------------+---------------------+--------------------
| Sup_Name                   | Sup_ID | Faculty                        | Title               | Research_int
+----------------------------+--------+--------------------------------+---------------------+--------------------
| School of Graduate Studies |   1000 | School of Graduate Studies     |                     |
| Andrew Black               |   1001 | Department of Computer Science | Professor           | Software Engineering
| Cherry Lara                |   1002 | Deparment of Economics         | Associate Professor | Microeconomics
+----------------------------+--------+--------------------------------+---------------------+--------------------
3 rows in set (0.05 sec)
```

**Figure 14 Screenshot of Selecting all rows from student and supervisor table**

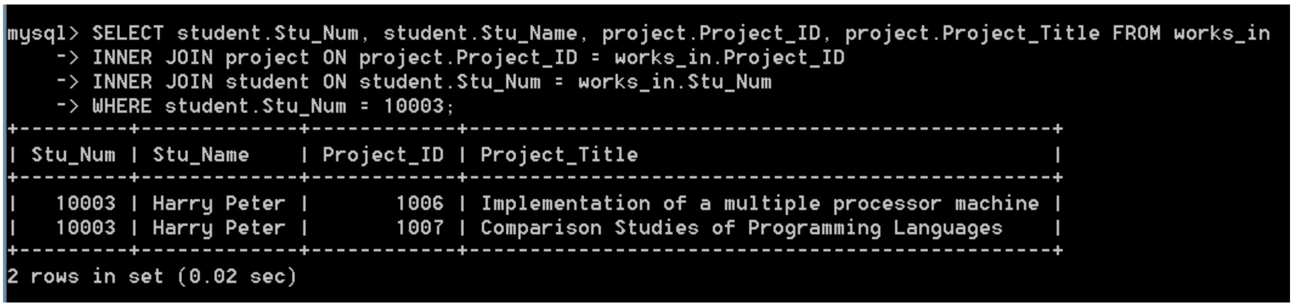
Figure 15 shows a screenshot of the student table joining together with the supervisor table to show all the names of the students and the names of their supervisors as well as the supervisor ID.



```
mysql> SELECT student.Stu_Name, supervisor.Sup_Name, supervisor.Sup_ID
    -> FROM student INNER JOIN supervisor ON student.Sup_ID = supervisor.Sup_ID;
+--------------+----------------------------+--------+
| Stu_Name     | Sup_Name                   | Sup_ID |
+--------------+----------------------------+--------+
| Jackson Clark | School of Graduate Studies |   1000 |
| Ben Steven   | School of Graduate Studies |   1000 |
| Harry Peter  | Cherry Lara                |   1002 |
| Sally Jean   | Andrew Black               |   1001 |
| Jena Chen    | Cherry Lara                |   1002 |
+--------------+----------------------------+--------+
5 rows in set (0.12 sec)
```

**Figure 15 Screenshot of Joining the Student and Supervisor tables**

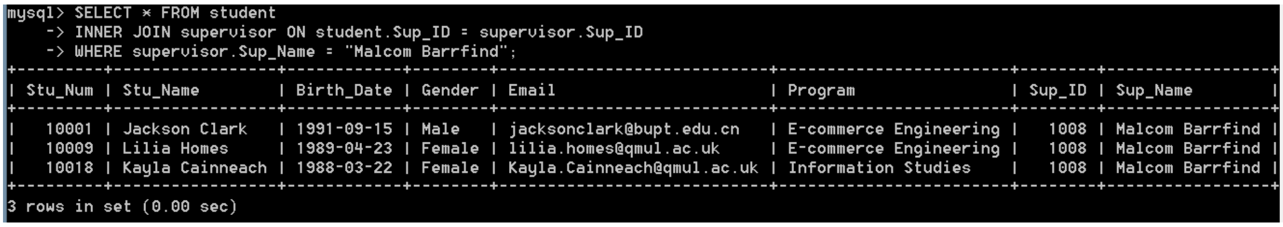
Figure 16 shows selecting a list of projects that one particular student is currently working in. (the Student Number selected in this project is 10003). It can be seen that all the project that the students are working in are presented with inner join of three tables.



```
mysql> SELECT student.Stu_Num, student.Stu_Name, project.Project_ID, project.Project_Title FROM works_in
    -> INNER JOIN project ON project.Project_ID = works_in.Project_ID
    -> INNER JOIN student ON student.Stu_Num = works_in.Stu_Num
    -> WHERE student.Stu_Num = 10003;
+---------+-------------+------------+------------------------------------------------+
| Stu_Num | Stu_Name    | Project_ID | Project_Title                                  |
+---------+-------------+------------+------------------------------------------------+
|   10003 | Harry Peter |       1006 | Implementation of a multiple processor machine |
|   10003 | Harry Peter |       1007 | Comparison Studies of Programming Languages    |
+---------+-------------+------------+------------------------------------------------+
2 rows in set (0.02 sec)
```

**Figure 16 Screenshot of Selecting projects that the student is working in**

Figure 17 then shows a list of the students that a particular supervisor is currently instructing. Noticing that the supervisor specified here is with name "Malcom Barrfind".



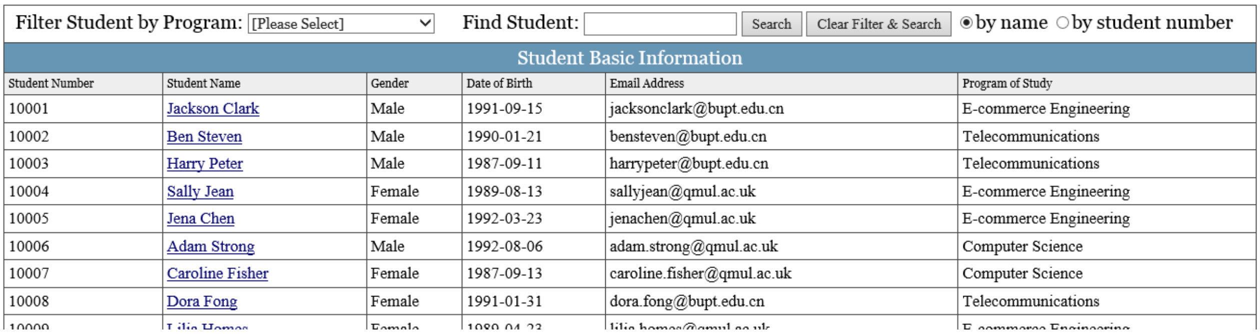
```
mysql> SELECT * FROM student
    -> INNER JOIN supervisor ON student.Sup_ID = supervisor.Sup_ID
    -> WHERE supervisor.Sup_Name = "Malcom Barrfind";
+---------+-----------------+------------+--------+----------------------------+------------------------+--------+-----------------+
| Stu_Num | Stu_Name        | Birth_Date | Gender | Email                      | Program                | Sup_ID | Sup_Name        |
+---------+-----------------+------------+--------+----------------------------+------------------------+--------+-----------------+
|   10001 | Jackson Clark   | 1991-09-15 | Male   | jacksonclark@bupt.edu.cn   | E-commerce Engineering |   1008 | Malcom Barrfind |
|   10009 | Lilia Homes     | 1989-04-23 | Female | lilia.homes@qmul.ac.uk     | E-commerce Engineering |   1008 | Malcom Barrfind |
|   10018 | Kayla Cainneach | 1988-03-22 | Female | Kayla.Cainneach@qmul.ac.uk | Information Studies    |   1008 | Malcom Barrfind |
+---------+-----------------+------------+--------+----------------------------+------------------------+--------+-----------------+
3 rows in set (0.00 sec)
```

**Figure 17 List of Students instructed by a particular supervisor**

The purpose of demonstrating these screenshots in this section is to give the readers a brief look of how the database is successfully implemented to store the database entities and their attributes as tables, and how foreign key constraints and participation constraints have been able to specify their relationship.

### 4.1.2 Screenshots from the Research Student Manager Website

In this section, some screenshots of the main functional modules of the research student manager website is presented. Figure 18 shows a screen shot of the student information module, where all the student information is retrieved from the student table in the database and is printed on the web page in a table. Administrators can filter students by program or find them by name or student number. The name of each student is also a clickable hyper link that can lead to a pop up window like in Figure 19, which shows all the detailed information of that student.



Filter Student by Program: [Please Select]
Find Student:
Search
Clear Filter & Search
● by name ○ by student number

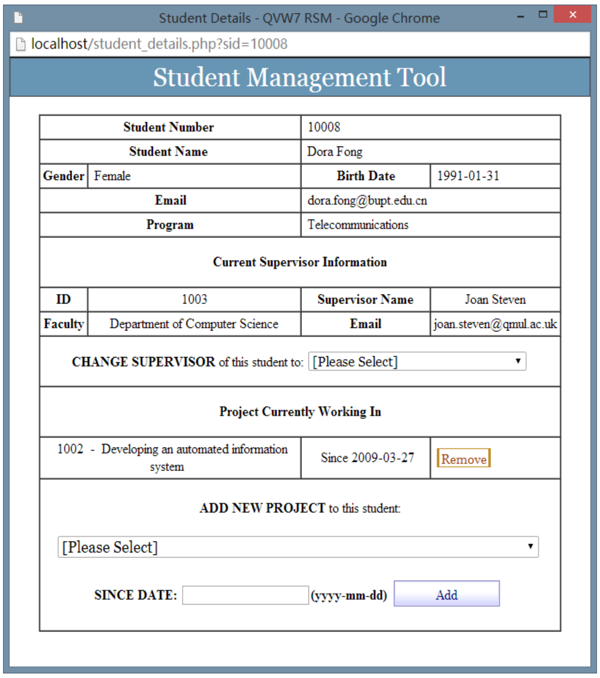
Student Basic Information

| Student Number | Student Name | Gender | Date of Birth | Email Address | Program of Study |
|---|---|---|---|---|---|
| 10001 | Jackson Clark | Male | 1991-09-15 | jacksonclark@bupt.edu.cn | E-commerce Engineering |
| 10002 | Ben Steven | Male | 1990-01-21 | bensteven@bupt.edu.cn | Telecommunications |
| 10003 | Harry Peter | Male | 1987-09-11 | harrypeter@bupt.edu.cn | Telecommunications |
| 10004 | Sally Jean | Female | 1989-08-13 | sallyjean@qmul.ac.uk | E-commerce Engineering |
| 10005 | Jena Chen | Female | 1992-03-23 | jenachen@qmul.ac.uk | E-commerce Engineering |
| 10006 | Adam Strong | Male | 1992-08-06 | adam.strong@qmul.ac.uk | Computer Science |
| 10007 | Caroline Fisher | Female | 1987-09-13 | caroline.fisher@qmul.ac.uk | Computer Science |
| 10008 | Dora Fong | Female | 1991-01-31 | dora.fong@bupt.edu.cn | Telecommunications |
| 10009 | Lilia Homes | Female | 1989-04-23 | lilia.homes@qmul.ac.uk | E-commerce Engineering |

**Figure 18 Screenshot of the Student Information Module**

In the student details pop up window, all the basic information of a student can be viewed by the administrator. Also can be viewed is some basic information of the supervisor who instructs this student and a list of all the projects that the students is currently working in. As one of the main features of managing research students, the student details pop up window also allows the administrator to change supervisor of the student through the drop down box, remove student from a currently working project by clicking the "remove" button on the line of each project, and enrol student to a new project by choosing the project, enter the since date and click on "add".

Research Student Manager



Student Details - QVW7 RSM - Google Chrome

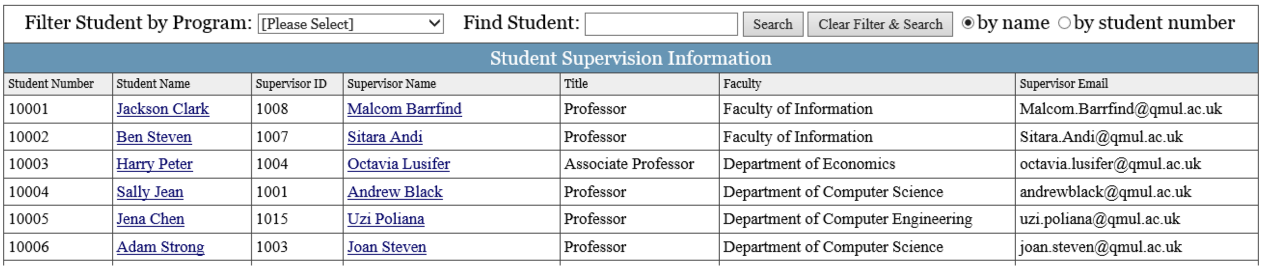
localhost/student_details.php?sid=10008

Student Management Tool

| | | | |
|---|---|---|---|
| **Student Number** | | 10008 | |
| **Student Name** | | Dora Fong | |
| **Gender** | Female | **Birth Date** | 1991-01-31 |
| **Email** | | dora.fong@bupt.edu.cn | |
| **Program** | | Telecommunications | |
| **Current Supervisor Information** | | | |
| **ID** | 1003 | **Supervisor Name** | Joan Steven |
| **Faculty** | Department of Computer Science | **Email** | joan.steven@qmul.ac.uk |
| **CHANGE SUPERVISOR** of this student to: [Please Select] ▾ | | | |
| **Project Currently Working In** | | | |
| 1002 - Developing an automated information system | | Since 2009-03-27 | Remove |
| **ADD NEW PROJECT** to this student: | | | |
| [Please Select] ▾ | | | |
| **SINCE DATE:** (yyyy-mm-dd) Add | | | |

**Figure 19 Student Details Pop-up Window**

In the Student Management module, there are also two sub modules: Supervision Details, where a list of student and their supervisor will be shown like in Figure 20, and Student Project Details, where a list of projects that the student is working in is shown like in Figure 21. These two modules work similarly like the Student Information module in terms of filtering and searching functions.

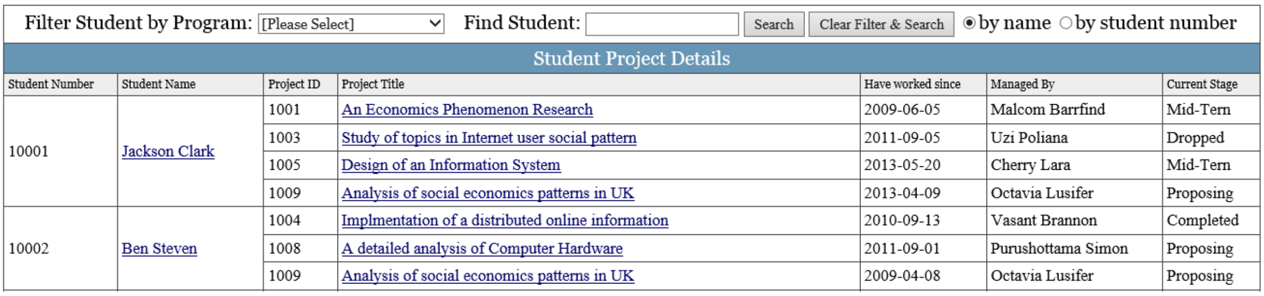


Filter Student by Program: [Please Select] ▾ Find Student: Search Clear Filter & Search ● by name ○ by student number

Student Supervision Information

| Student Number | Student Name | Supervisor ID | Supervisor Name | Title | Faculty | Supervisor Email |
|---|---|---|---|---|---|---|
| 10001 | Jackson Clark | 1008 | Malcom Barrfind | Professor | Faculty of Information | Malcom.Barrfind@qmul.ac.uk |
| 10002 | Ben Steven | 1007 | Sitara Andi | Professor | Faculty of Information | Sitara.Andi@qmul.ac.uk |
| 10003 | Harry Peter | 1004 | Octavia Lusifer | Associate Professor | Department of Economics | octavia.lusifer@qmul.ac.uk |
| 10004 | Sally Jean | 1001 | Andrew Black | Professor | Department of Computer Science | andrewblack@qmul.ac.uk |
| 10005 | Jena Chen | 1015 | Uzi Poliana | Professor | Department of Computer Engineering | uzi.poliana@qmul.ac.uk |
| 10006 | Adam Strong | 1003 | Joan Steven | Professor | Department of Computer Science | joan.steven@qmul.ac.uk |

**Figure 20 Student Management - Supervision Details**

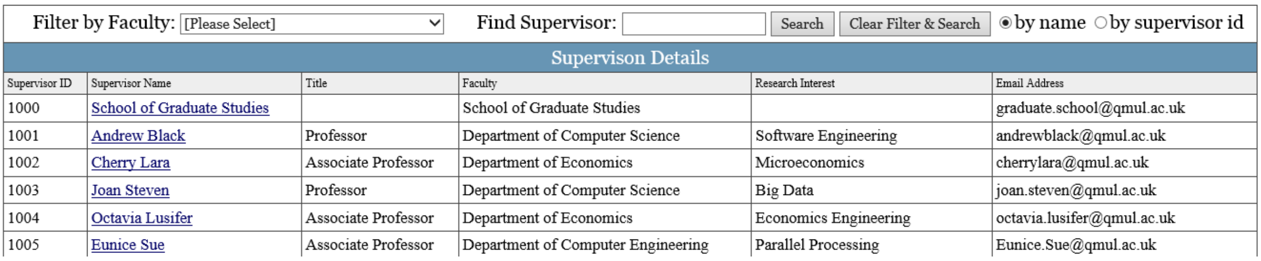


Filter Student by Program: [Please Select] ▾ Find Student: Search Clear Filter & Search ● by name ○ by student number

Student Project Details

| Student Number | Student Name | Project ID | Project Title | Have worked since | Managed By | Current Stage |
|---|---|---|---|---|---|---|
| 10001 | Jackson Clark | 1001 | An Economics Phenomenon Research | 2009-06-05 | Malcom Barrfind | Mid-Tern |
| | | 1003 | Study of topics in Internet user social pattern | 2011-09-05 | Uzi Poliana | Dropped |
| | | 1005 | Design of an Information System | 2013-05-20 | Cherry Lara | Mid-Tern |
| | | 1009 | Analysis of social economics patterns in UK | 2013-04-09 | Octavia Lusifer | Proposing |
| 10002 | Ben Steven | 1004 | Implmentation of a distributed online information | 2010-09-13 | Vasant Brannon | Completed |
| | | 1008 | A detailed analysis of Computer Hardware | 2011-09-01 | Purushottama Simon | Proposing |
| | | 1009 | Analysis of social economics patterns in UK | 2009-04-08 | Octavia Lusifer | Proposing |

**Figure 21 Student Management - Student Project Details**



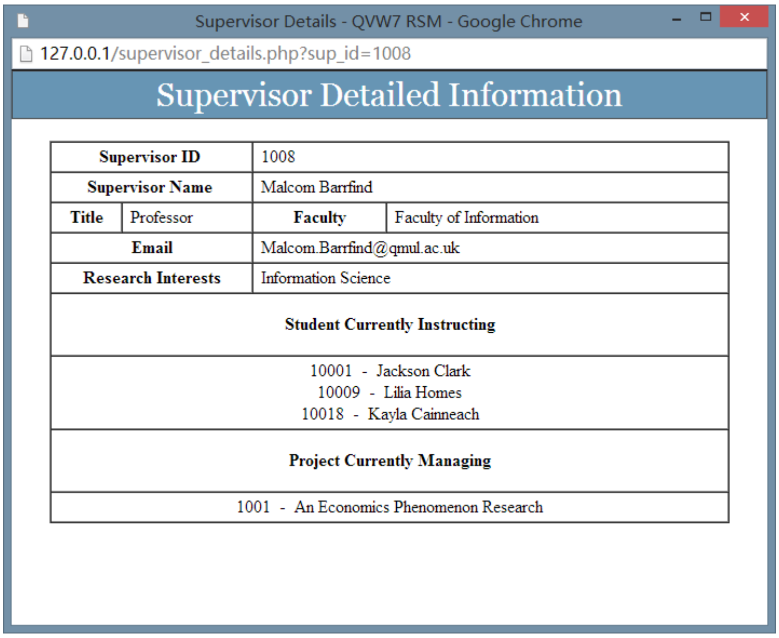
Filter by Faculty: [Please Select] Find Supervisor: Search Clear Filter & Search ● by name ○ by supervisor id

Supervison Details

| Supervisor ID | Supervisor Name | Title | Faculty | Research Interest | Email Address |
|---|---|---|---|---|---|
| 1000 | School of Graduate Studies | | School of Graduate Studies | | graduate.school@qmul.ac.uk |
| 1001 | Andrew Black | Professor | Department of Computer Science | Software Engineering | andrewblack@qmul.ac.uk |
| 1002 | Cherry Lara | Associate Professor | Department of Economics | Microeconomics | cherrylara@qmul.ac.uk |
| 1003 | Joan Steven | Professor | Department of Computer Science | Big Data | joan.steven@qmul.ac.uk |
| 1004 | Octavia Lusifer | Associate Professor | Department of Economics | Economics Engineering | octavia.lusifer@qmul.ac.uk |
| 1005 | Eunice Sue | Associate Professor | Department of Computer Engineering | Parallel Processing | Eunice.Sue@qmul.ac.uk |

**Figure 22 Supervisor Information Module**

Figure 22 then shows another module of the research student manager, the Supervisor Information module. A list of detailed information of all supervisors are shown here, and filtering and searching is also supported. The supervisor names can be clicked to show a detailed information pop up window where the student instructed and project managed by that supervisor is also viewable. (See Figure 23)



Supervisor Details - QVW7 RSM - Google Chrome

127.0.0.1/supervisor_details.php?sup_id=1008

Supervisor Detailed Information

| | | | |
|---|---|---|---|
| **Supervisor ID** | | 1008 | |
| **Supervisor Name** | | Malcom Barrfind | |
| **Title** | Professor | **Faculty** | Faculty of Information |
| **Email** | | Malcom.Barrfind@qmul.ac.uk | |
| **Research Interests** | | Information Science | |

**Student Currently Instructing**

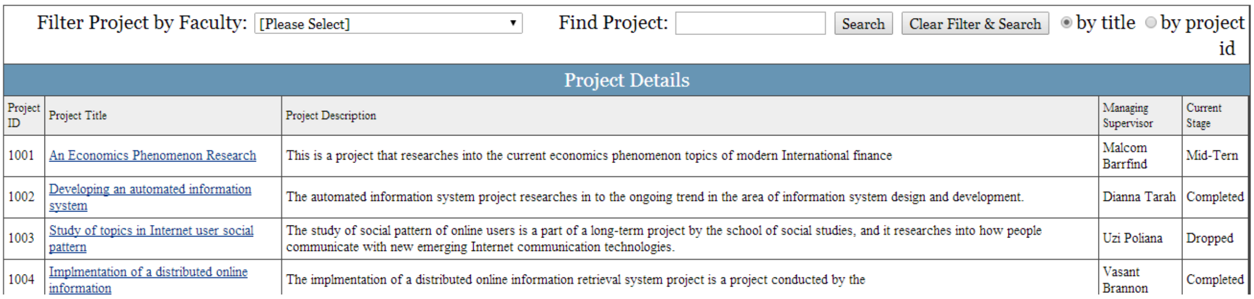
10001 - Jackson Clark
10009 - Lilia Homes
10018 - Kayla Cainneach

**Project Currently Managing**

1001 - An Economics Phenomenon Research

**Figure 23 Supervisor Details Pop-up Window**

The Project Management Module also works similarly like the Student Management module. It has a sub module listing a table of basic project information of all projects like shown in Figure 24, a sub module with a list of the project assessment scores shown like in Figure 25, and a sub module with a list of funding details of each project like shown in Figure 26. These three sub modules are such designed to focus on different aspects of project information.

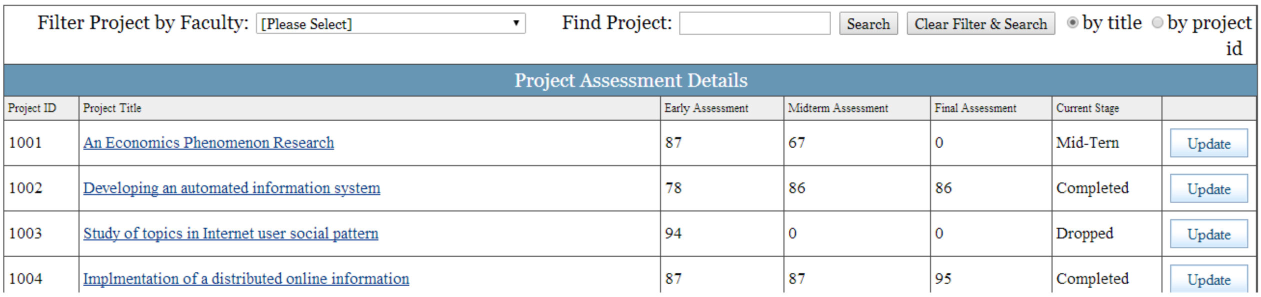


Filter Project by Faculty: [Please Select] ▾
Find Project:
Search
Clear Filter & Search
● by title ○ by project id

Project Details

| Project ID | Project Title | Project Description | Managing Supervisor | Current Stage |
|---|---|---|---|---|
| 1001 | An Economics Phenomenon Research | This is a project that researches into the current economics phenomenon topics of modern International finance | Malcom Barrfind | Mid-Tern |
| 1002 | Developing an automated information system | The automated information system project researches in to the ongoing trend in the area of information system design and development. | Dianna Tarah | Completed |
| 1003 | Study of topics in Internet user social pattern | The study of social pattern of online users is a part of a long-term project by the school of social studies, and it researches into how people communicate with new emerging Internet communication technologies. | Uzi Poliana | Dropped |
| 1004 | Implmentation of a distributed online information | The implmentation of a distributed online information retrieval system project is a project conducted by the | Vasant Brannon | Completed |

**Figure 24 Project Management - Project Basic Information**

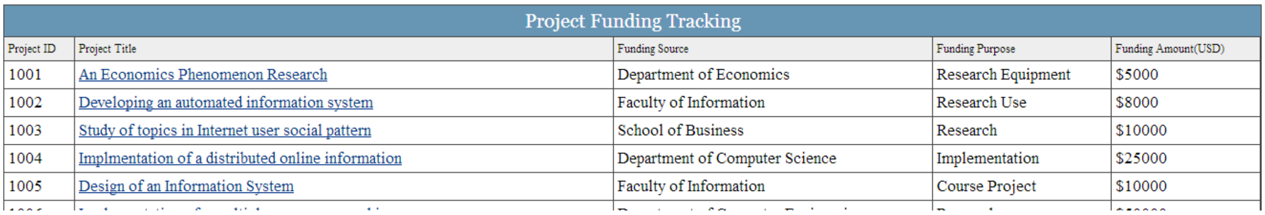


Filter Project by Faculty: [Please Select] ▾
Find Project:
Search
Clear Filter & Search
● by title ○ by project id

Project Assessment Details

| Project ID | Project Title | Early Assessment | Midterm Assessment | Final Assessment | Current Stage | |
|---|---|---|---|---|---|---|
| 1001 | An Economics Phenomenon Research | 87 | 67 | 0 | Mid-Tern | Update |
| 1002 | Developing an automated information system | 78 | 86 | 86 | Completed | Update |
| 1003 | Study of topics in Internet user social pattern | 94 | 0 | 0 | Dropped | Update |
| 1004 | Implmentation of a distributed online information | 87 | 87 | 95 | Completed | Update |

**Figure 25 Project Management - Project Assessment**



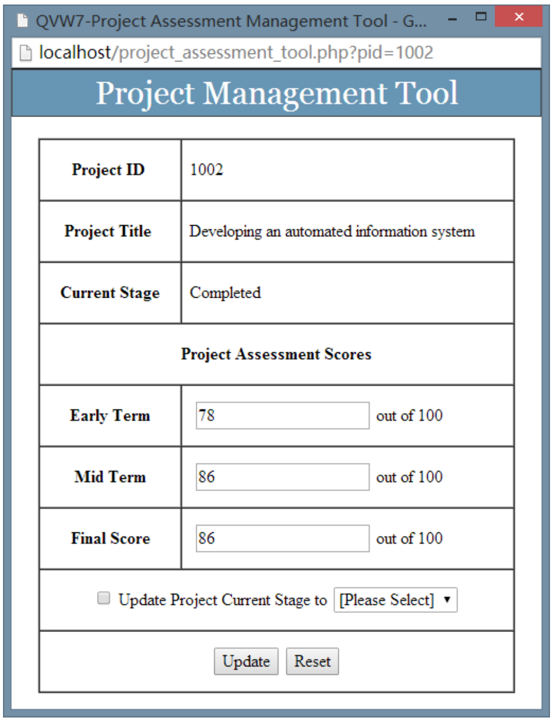
Project Funding Tracking

| Project ID | Project Title | Funding Source | Funding Purpose | Funding Amount(USD) |
|---|---|---|---|---|
| 1001 | An Economics Phenomenon Research | Department of Economics | Research Equipment | $5000 |
| 1002 | Developing an automated information system | Faculty of Information | Research Use | $8000 |
| 1003 | Study of topics in Internet user social pattern | School of Business | Research | $10000 |
| 1004 | Implmentation of a distributed online information | Department of Computer Science | Implementation | $25000 |
| 1005 | Design of an Information System | Faculty of Information | Course Project | $10000 |

**Figure 26 Project Management - Project Funding Tracking**

In the Project Assessment module, there is a “Update” button where the administrator can click and open a pop-up window like shown in Figure 27. This window is the project assessment management tool that allows them to manage assessment scores and project current stage.

Also noticeable is that the filtering and searching functions are also supported on the tables, and that the project title can be clicked on to pop up a project detail window like shown in Figure 28. This window shows the complete detailed information about that project, the supervisor that manages that project and a list of students that are currently working on that project. The notification sending tool is also embedded into this pop up window. This tool will show all the history notification that has been sent by the administrator to this very project. The administrator can view this as well as sending new notifications with this tool.

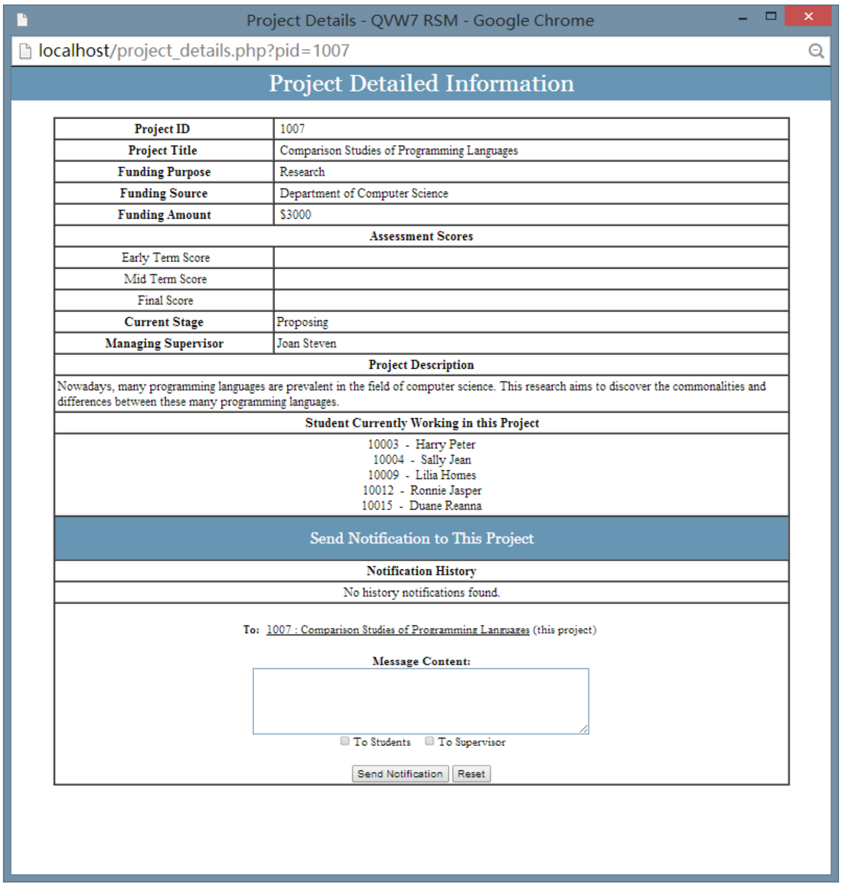

QVW7-Project Assessment Management Tool - G...
localhost/project_assessment_tool.php?pid=1002

Project Management Tool

|  |  |
|---|---|
| **Project ID** | 1002 |
| **Project Title** | Developing an automated information system |
| **Current Stage** | Completed |
| **Project Assessment Scores** |  |
| **Early Term** | 78 out of 100 |
| **Mid Term** | 86 out of 100 |
| **Final Score** | 86 out of 100 |

□ Update Project Current Stage to [Please Select] ▾

**Figure 27 Project Assessment Management Tool**


Project Details - QVW7 RSM - Google Chrome
localhost/project_details.php?pid=1007

Project Detailed Information

|  |  |
|---|---|
| **Project ID** | 1007 |
| **Project Title** | Comparison Studies of Programming Languages |
| **Funding Purpose** | Research |
| **Funding Source** | Department of Computer Science |
| **Funding Amount** | $3000 |
| **Assessment Scores** |  |
| Early Term Score |  |
| Mid Term Score |  |
| Final Score |  |
| **Current Stage** | Proposing |
| **Managing Supervisor** | Joan Steven |

**Project Description**

Nowadays, many programming languages are prevalent in the field of computer science. This research aims to discover the commonalities and differences between these many programming languages.

**Student Currently Working in this Project**

10003 - Harry Peter
10004 - Sally Jean
10009 - Lilia Homes
10012 - Ronnie Jasper
10015 - Duane Reanna

Send Notification to This Project

**Notification History**

No history notifications found.

**To:** 1007 : Comparison Studies of Programming Languages (this project)

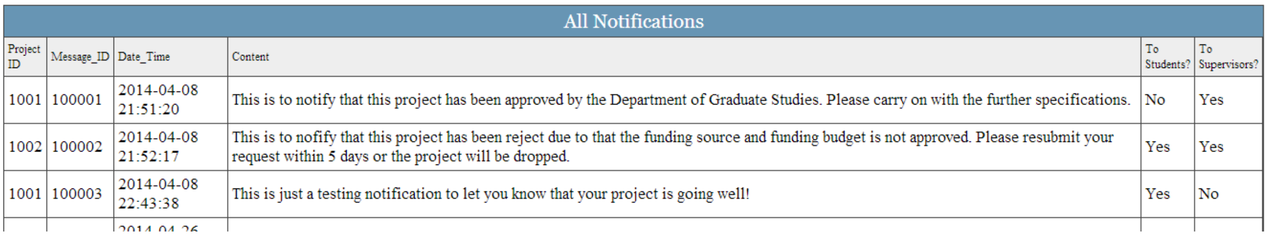
**Message Content:**

□ To Students □ To Supervisor

**Figure 28 Project Details Pop-up Window**

Research Student Manager

The last module to demonstrate here is the Notifications module in which the administrator can view a list of the detailed information of all the notifications (Figure 29). New notification can also be sent through the Manage Notifications sub module. There's also a user Help module, which is very straight forward and will not be shown here.



All Notifications

| Project ID | Message_ID | Date_Time | Content | To Students? | To Supervisors? |
|---|---|---|---|---|---|
| 1001 | 100001 | 2014-04-08 21:51:20 | This is to notify that this project has been approved by the Department of Graduate Studies. Please carry on with the further specifications. | No | Yes |
| 1002 | 100002 | 2014-04-08 21:52:17 | This is to nofify that this project has been reject due to that the funding source and funding budget is not approved. Please resubmit your request within 5 days or the project will be dropped. | Yes | Yes |
| 1001 | 100003 | 2014-04-08 22:43:38 | This is just a testing notification to let you know that your project is going well! | Yes | No |
|  |  | 2014-04-26 |  |  |  |

**Figure 29 Notifications Module**

## 4.2 Test Outputs and Evaluation

In this section, the test outputs and evaluation will be focused on testing the research student manager from a system perspective, which means the "black box testing" of how the system satisfies requirements and conforms to the specification.

### 4.2.1 Test Outputs

The testing will be conducted around three questions: (1) How well does the implemented system meet system requirements in specification? (2) Do all the functionalities in the system work properly as the specification? (3) Are there any existing issues, defects, bugs that will result in failure of a user's use case? Based on these questions as testing goals, the functional and non-functional requirements of the research student manager has been gone through as well as the project specifications at the initial stage of the project, and the following testing results are produced.

**Table 2: Test Outputs – Final Black Box Testing of Research Student Manager**

| Deliverables Tested | Achieved/ Not Achieved | Comments |
|---|---|---|
| The system should show account information of students and supervisors | Yes | Research student manager is assumed not being able to insert new students or supervisors |
| The system should allow administrators to change supervisor of a student | Yes |  |
| The system should allow administrators | Yes |  |

| | | |
|---|---|---|
| to manage project assessment scores | | |
| The system should allow administrators to trace funding details | Yes | The research student manager is not assumed to have the rights to modify funding details |
| The system should allow administrators to update project status | Yes | |
| The system should allow administrators to enrol students in project to work in it | Yes | |
| The system should allow administrators to send notifications to projects | Yes | |

### 4.2.2 Evaluation

Based on analysing the test outputs and the entire research student manager system, the following questions are raised and discussed about the research student manager.

#### *4.2.2.1 How automation of administrative work is facilitated by the system*

The research student manager has brought possible automation to administrative work carried out by administrators at the university. Most importantly, the student, supervisor and project records are turned from paper-based to electronic-based records that are maintained and processed in databases. This has made it possible for computer-based programs to manipulate and process data automatically. Also, the programs can be modified by the technical support of the system to satisfy different needs through automation.

#### *4.2.2.2 How well does the system improve work efficiency?*

To improve work efficiency is no doubt one of the key goals of designing and implementing the research student manager, and the system has achieved this in various ways. Firstly, the filtering and searching functionalities can be used in multiple modules to very quickly locate and identify information needed by the administrator. Secondly, the management tools, including the student management tools and the project assessment tools are designed to complete the daily administrative tasks through a few clicks, which will definitely improve work efficiency for the administrator. Last but not least, the notification sending tool embedded in both the project details page and the notifications management page can make the work of announcing to projects very efficiently and conveniently.

#### *4.2.2.3 Possible innovation that can be triggered by introducing this system*

The introduction of the research student manager can not only facilitate the automation and efficiency of daily administrative work of managing research students, it also brings about changes and possible innovation to the work process of administrative staff of the university. For example, the functions of enrolling students in projects and removing students from them may bring innovative changes to the work process of project personnel management, thus the research students may be enabled to work in multiple projects while the manager system is still able to keep up to date records about students involved in each project.

#### *4.2.2.4 Possible problems and concerns that can be caused*

Though the research student manager may enhance the work efficiency of research student management tasks and bring innovation to the administrative work process, it may also cause possible problems and concerns with the introduction of such a new information system. For example, the research student manager will greatly change the work process of administrative staff so they will need to be trained to learn the system, and it will also take some time before they can utilize the benefits from using the system and actually improve their work efficiency. Plus, privacy and security concerns may be introduced by the availability of electronic records as they will become more vulnerable to data intrusion when the research student manager system is running online.

## 4.3 User Reviews

In this section, some quotes are presented from some of the users who are given a chance to use the final version of the research student manager. All of these user comments have been carefully reviewed by the author of this report, and most of them have been taken as very valuable advices to improve the research student manager system.

*"The interface of the research student manager looks not very friendly for the users, it should conform to some simple design principles and needs to be improved to give the user a sense of simplicity and convenience* – **Chixiang Chen**

*"The system should allow users to change their password and should allow users to register new accounts and have the differentiation of account privileges."* – **Jing Wang**

*"It is very good that the system has a help page, but the user manual should be described in more detail, and it would be better to have a tutorial for first time users of this system."* – **Chenfei Gao**

# Chapter 5: Conclusion and Further Work

## 5.1 Conclusion

This project aimed to develop a web-based management tool that can be used to manage research students in a university in order to automate administrative tasks carried out by the administrative staff of the university. With such a tool, the goal is to improve the work efficiency of the research student management process. This project report discussed the entire software engineering process of building the research student manager, from the background gathering, analysis phase, through to designing and implementation of the system. In the end, the testing and evaluation are discussed as well as a complete screenshot demonstration of the system.

The research student manager has achieved all the functionalities planned in the project initial specification, and can satisfy all the use cases as required by the administrators of the university. The functional and non-functional requirements are also met in each of the functional modules and their sub modules. The process of building the research student manager conforms strictly to the software engineering principles and methodologies. The requirement engineering phase applied some common requirement finding techniques to elicit requirements from stakeholders, the database design originated from identifying elements, drawing diagrams and ended in the complete implementation of the database, and the website is built from both designing the interface and programming the back end logic of the management system.

As evaluated, the research student manager system has demonstrate benefits of enhancing the automation of administrative work, improving work efficiency for the administrators, and bringing about changes and innovation to the entire work process. This is also the goals and missions of what this project set out to accomplish. Even though there might also be some concerns, it can be concluded that the introduction of the research student manager will be beneficial to the research student management work for the administrators at the university. The author of this report and this project hopes to provide this project as an implication to similar researches in the information system design and implementation of today’s universities and colleges.

## 5.2 Further Work

While the research student manager has been completely designed and implemented, there are still many aspects of the system that can be further improved in future work. These future improvements can include further realization of more advanced functionalities for the administrators, improving the current version of the system to minimize bugs and defeats, and further discovering more aspects of the administrative work that can be automated, innovated and improved.

For further advance functionalities, multiple user support will be one of the most important features to include in the system in order to allow administrators of different responsibilities to work on the same system. Based on the user privilege differentiation, other features that will enable more management work on the students, the supervisor and the projects can then be added.

Also the current version of the system will still need to be further evaluated in terms of its bugs and defects, and other security and privacy concerns. For example, the password need to be more confidentially stored, the login process will need to be more secured.

Finally, more aspects of the administrative may be automated, innovated and improved. This report and the research student manager built in this project hopes to provide some implications to other related researches and projects. The source code of the system is open for study and research purpose.

# References

Stuppy, J. J. (2000). *"Automated testing and electronic instructional delivery and student management system."* U.S. Patent No. 6,146,148. Washington, DC: U.S. Patent and Trademark Office. Issued November 14, 2000.

Lijia Wu, (2011). *"Research Student Grades Management System, a Database Application course project"*, Department of Computer Science, Henan Industry University, December 2011

Yun, (2013), *"A PHP based database management system development report"*, Dept. of Computer Science, Anhui Industry University. August 2013.

Jinlong Pan, (2009), *"Design of a PHP based Student Information System"*, Huadong Transportation University, June 2009.

Zhonghua Shi, (2011), *"Design and implementation of a university registry office information system"*, Hebei University, June 2011.

Songqing Dong, (2012) *"Design and Implementation of a Java based Research Student Management System"*, Anhui Industry University, June 2012.

***List of Windows 8 Metro style logo used:***

| **Image Source in the website folder** | **Original Source** |
|---|---|
| Images\login_logo.png | buttons\System Icons\User Accounts alt.png |
| Images\student.png | buttons\Other\Graduation.png |
| Images\ notifications.png | buttons\Other\notifications.png |
| Images\ project.png | buttons\Other\task.png |
| Images\ supervisor.png | buttons\System Icons\User Accounts.png |

Source Package: *metro_ui_icon_set___725_icons_by_dakirby309-d4n4w3q*

# Acknowledgement

**Vindya Wijeratne**

For supervising this project and providing valuable and constructive comments and feedback, and also for being the project coordinator and arranging project meetings and the project term checks.

**Chixiang Chen**

For providing user review on the system interface design.

**Jing Wang**

For providing user review on the system security related features.

**Chenfei Gao**

For providing user review on the system help and manual module.

# Appendix

- Project Specification

| | | | | | |
|---|---|---|---|---|---|
| 北京邮电大学 | | | | | |
| 本科毕业设计（论文）任务书 | | | | | |
| **Project Specification Form** | | | | | |
| 学院 School | International School | 专业 Programme | E-Commerce (H6NF) | 班级Class | 2010215111 |
| 学生姓名 Name | HUANG Yajun | 学号 BUPT student no | 10213092 | 学号 QM student no | 100671751 |
| 设计（论文）编号Project No. | | | IC_3092 | | |
| 设计（论文）题目Project Title | | | Research Student Manager | | |
| 题目分类 Scope | Implementation | | Computer Software | | Software |
| 主要任务及目标Main tasks and target： | | | By | | |
| Task 1: Requirement analysis and literature survey of similar tools. | | | January 2014 | | |
| Task 2: Design a research student management tool including the main functionalities desired. | | | February 2014 | | |
| Task 3: Implement and test the tool. | | | April 2014 | | |
| Task 4: Evaluate and add extra functionality. | | | May 2014 | | |
| Measurable outcomes | | | | | |
| 1) Design of research student management tool meeting the stated requirements. | | | | | |
| 2) Working implementation of the tool. | | | | | |
| 3) Evaluation of the tool. | | | | | |
| 主要内容 Project description： | | | | | |
| Managing research students in a university involves a number of administration tasks that are usually carried out manually. It is clear that some of these tasks could be automated. The aim of this project is to build a software application (stand-alone or web-based) in order to fully/partially automate the management of research student right from the stage of applying through to graduation. This would involve keeping track of funding details, various assessment stages and supervision details etc. It can also send relevant notifications to students and supervisors. The project can be extended to include extra functionalities such as different levels of access permissions, also other functions such as organising assessment sessions, if time permits. This tool can be developed specifically for either QM or BUPT. | | | | | |
| Project outline | | | | | |
| This project is about developing a web-based management tool that can be used to manage research students in a university from the stage of applying to graduation. The purpose of developing such a tool is to automate the research student management tasks usually carried out manually by administrative staff of the university. By introducing and implementing this tool, this project aims to improve the automation and the efficiency of the management process. To tackle the project, I'll conform to a systematic approach applying software engineering theories and methods. And I'll develop the manager tool specifically for the use of the administrative staff at BUPT so that the scope of the project can be better targeted. The initial phase of this project will be the requirement analysis of the research student manager tool. In this period, the administrative staff at BUPT will be engaged for requirement finding, and their needs as the main user group of the software will be explored as an important source of requirement. Relevant literature of similar tools will also be reviewed for better analysis on the requirement of this project. The second stage will involve designing the management tool to include the desired functionalities, and the technical implementation aspects of the project will be handled at this stage. For example, the design of the database to be implemented, the general structure of the tool application, and the how will the tool be implemented. The next stage is the implementation and testing of the research student management tool. The programming of the web application pages, and the implementation of the database will be done. Testing will then involve deploying the web contents on a local web server, linking them with the database and evaluating if all the functionalities work well. The final stage of the project will involve an evaluation of the management tool from aspects including whether user requirements are met, whether all the functionalities are included and what needs to be further improved. There may also be extra functionalities added to the management tool on condition that all the basic ones are operating well and time permitting. The entire process of this project will be well documented in the final report, and a Gantt chart will be used to keep track of the progress. | | | | | |

| | Nov | | Dec | | Jan | | Feb | | Mar | | Apr | | May | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Task 1: Requirement analysis and literature survey of similar tools.** | | | | | | | | | | | | | | |
| Initial literature review of similar tools for preliminary ideas of the management tool | | ■ | ■ | | | | | | | | | | | |
| Requirement finding through engaging research student management administrative staff of BUPT | | | ■ | ■ | ■ | | | | | | | | | |
| Analyse desired functionalities based on findings | | | | ■ | ■ | ■ | | | | | | | | |
| | | | | | | | | | | | | | | |
| **Task 2: Design a research student management tool including the main functionalities desired.** | | | | | | | | | | | | | | |
| Design an appropriate database for storing data records for the management tool | | | | ■ | ■ | ■ | | | | | | | | |
| Design the structure of the management tool with desired functionalities | | | | | ■ | ■ | ■ | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| **Task 3: Implement and test the tool** | | | | | | | | | | | | | | |
| Implement the database for the tool, enter mock data entries for testing | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | |
| Implement the main functionalities for the tool, deploy web contents and link database on local server for testing | | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | | |
| Test with functionalities: test through all desired functionalities to see if all functional parts work properly | | | | | | | | | ■ | ■ | ■ | ■ | | |
| Test with data: test with single data entry and then multiple data entries to see if the tool operates well with database | | | | | | | | | | ■ | ■ | ■ | | |
| **Task 4: Evaluate and add extra functionality.** | | | | | | | | | | | | | | |
| Evaluation of the tool by developer for performance analysis | | | | | | | | | | ■ | ■ | ■ | ■ | |
| Evaluation of the tool by user group and the developer to see if all requirements are met | | | | | | | | | | | ■ | ■ | | |
| Evaluate the tool for further improvement and add extra functionality | | | | | | | | | | | ■ | ■ | ■ | |
| Project Report writing and other project documenting tasks | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |

Research Student Manager

- Preliminary Report

北京邮电大学
**BBC6521 Project**毕业设计
**[Preliminary]Report**
**[**第一期**]** 进度报告

<table>
<tr><td colspan="2">学院<br>School</td><td colspan="2">International School</td><td colspan="2">专业<br>Programme</td><td colspan="2">E-Commerce Engineering with Law</td><td colspan="2">班级<br>Class</td><td colspan="2">2010215111</td></tr>
<tr><td colspan="2">学生姓名<br>Student Name</td><td colspan="2">HUANG Yajun</td><td colspan="2">学号<br>BUPT Student No.</td><td colspan="2">10213092</td><td colspan="2">学号<br>QM Student No.</td><td colspan="2">100671751</td></tr>
<tr><td colspan="3">设计（论文）编号<br>Project No.</td><td colspan="3">IC_3092</td><td colspan="3">电子邮件<br>Email</td><td colspan="3">10213092@bupt.edu.cn</td></tr>
<tr><td colspan="6">设计（论文）题目<br>Project Title</td><td colspan="6">QVW7 -Research Student Manager</td></tr>
<tr><td colspan="12">已完成工作/Finished Work：<br>From the beginning of this project to now is the initial phase of this project, as also reflected in the project Gantt chart as the Requirement Analysis phase during which Task 1 is being done. From mid-November to now I’ve done some site review of several similar student management tools. Some have detailed design and implementation document in paper (e.g. SQ Dong, <i>Java based Research Student Management System</i>, Anhui Industrial University) while others are reviewed by interacting and observing (e.g. the existing URP of BUPT for undergraduates). From my review I have gained a basic understanding of the common core functionalities and main goals for a student management system.<br>By searching through catalogues over the Internet, I’ve also looked at some relevant literature on topics like information system and software engineering. The <i>Implementing Data-Informed Decision Making in Schools—Teacher Access, Supports and Use</i>(US Dept. of Education, 2009), though discussed in a general scale of introducing data system to school settings, raised me some thoughts of how effective the research student management system I’m going to develop will help decision making and enhance process efficiency.<br>These requirement and literature/site review findings have helped me in preparing for later engaging with the administrative staff of BUPT for more specific requirement gathering for this management tool. I’ve also decided that the technical setting for my project will be PHP + MySQL, and I’ll further illustration the reasons and their advantages in the requirement analysis document.</td></tr>
<tr><td colspan="6">是否符合进度？<b>On schedule as per GANTT chart?</b></td><td colspan="6">[YES]</td></tr>
<tr><td colspan="12">下一步/Next steps:<br>The next step after this preliminary report will be engaging the administrative staff of BUPT to gather more specific requirements for the research student management tool. The requirement gathering techniques of software engineering will be followed, and the findings from this and the previous site and literature reviews will be well-documented. The requirement analysis document will be the foundation for designing the management tool in the next phase of this project. (I’m planning to refer to Software Engineering by Ian Sommerville,8th Edition; Addison Wesley 2007as my primary guide book for carrying out the software engineering methods)<br>For the designing phase, the requirement findings from this phase will be used to decide which functionalities to implement in the management tool. And the database will also be designed and illustrated using an Entity Relationship Diagram.(I’ll be referring to <i>Learning PHP and MySQL</i> by Davis, Michele E. and Phillips, Jon A. Feb 9 2009, which is a very handy textbook, as my guidance for programming PHP for MySQL)</td></tr>
</table>

- Early-term Progress Report

北京邮电大学
**BBC6521 Project**毕业设计
**[Early-term Progress]Report**
**[**第二期**]** 进度报告

| | | | | | |
|---|---|---|---|---|---|
| 学院<br>School | International School | 专业<br>Programme | E-Commerce Engineering with Law | 班级<br>Class | 2010215111 |
| 学生姓名<br>Student Name | HUANG Yajun | 学号<br>BUPT Student No. | 10213092 | 学号<br>QM Student No. | 100671751 |
| 设计（论文）编号<br>Project No. | IC_3092 | | 电子邮件<br>Email | 10213092@bupt.edu.cn | |
| 设计（论文）题目<br>Project Title | | | QVW7 -Research Student Manager | | |

已完成工作/Finished Work：

During this early term period, I’ve managed to carry on what I’ve planned to continue working on since the first term report. I’ve had several brief talks with some of the administrative staff from International School of BUPT who were responsible for managing all the undergraduate students in International School. They were using an online based University Resource Planning System which has been developed and used for years. I’ve also had a talk with an administrative staff from BUPT main campus, who works in the registrar’s office and also using part of the web portal of BUPT to do daily managing tasks on students and faculty.

The interview with them have helped be to gather some first-hand information about what functionalities the administrative staffs care most about the system, what are the advantages and disadvantages of the existing systems. I have documented these and this will be the requirement finding results for further going on with the design and development of the research student management system I am building. Based on this, I’ve begun to sketch some design for the database I’d like to build (considering tables, primary keys and relations). Also, I’ve summarized the basic, the necessary and the extended functionalities that I will implement.

Due to the final exam taking place during this period, my most priorities are given to preparation for the exams. But I did take some time to pick up something I’ve learnt about PHP and MySQL in the Internet Database course by looking through the slides and reading the textbook(*Learning PHP and MySQL* by Davis, Michele E. and Phillips, Jon A. Feb 9 2009).

| | |
|---|---|
| 是否符合进度？**On schedule as per GANTT chart?** | [YES] |

下一步/Next steps:

With the findings from the literature reviews (mostly done in the preliminary period) and the requirement analysis findings (as explained in the Finished Work of this early-term report), I think I am ready to focus on the design and implementation of the research student management system. As indicated in the Gantt Chart in the Project Initial Document, the Implementation work will start around the end of January, based on my requirement findings and preliminary design since December. I’m hoping to get the basic functionalities ready for demonstration at the Mid Term.

Research Student Manager

- Mid-term Check Form

| | | | | | |
|---|---|---|---|---|---|
| 北京邮电大学<br>本科毕业设计（论文）中期进展情况检查表<br>**Mid Term Check Form** | | | | | |
| 学院 School | International School | 专业 Programme | E-Commerce (H6NF) | 班级 Class | 2010215111 |
| 学生姓名 Name | HUANG Yajun | 学号 BUPT student no | 10213092 | 学号 QM student no. | 100671751 |
| 设计（论文）编号 Project No. | IC_3092 | | | | |
| 设计（论文）题目 Project Title | Research Student Manager | | | | |
| 题目分类 Scope | Implementation | Computer Software | Software | | |
| 主要内容: (毕业设计（论文）进展情况，字数一般不少于 1000 字) | | | | | |
| Main body: The progress of the research on the project. Total number of words is no less than 1000. | | | | | |
| 目标任务<br>Targets set at project initiation | At the mid-term of this project, I will make sure to finish the requirement analysis of the research student management tool and be able to present the findings from the requirement analysis phase. I will expect to be working on and finish most of the design for the tool, including designing an appropriate database for storing data records, designing the structure of the tool with desired functionalities and solving other technical aspect issues. A preliminary design of the research student manager will also be presented at the mid-term oral. | | | | |
| 是否完成目标<br>Targets met? Yes/No | Yes | | | | |
| 目前已完成任务<br>Finished Work | I've finished the requirement gathering and analysis phase of my project which involved using requirement finding techniques such as interview and literature to gather requirements from administrative staff and relevant research articles/development reports. Some requirement findings lead to functional requirements of the research student manager system like the display of project stages (project details shall be able to present project current status and previous assessment results), notification functions (administrator should be able to send notifications to students and supervises based on certain rules and relevance), while some lead to non-functional requirements like appearance (the system should look friendly), performance (data manipulation and page loading should respond in a reasonable time), and usability (the research student manager should be easy to use and the site map makes it easy to navigate through). Based on findings from these sources and the main objectives of my project, I've identified the main actors of the system as: student, supervisor, and the administrator. Though the administrators are the actual core users of the research student manager, the student and the supervisor are very important stakeholders within my project scope because their data and information are stored and manipulated in the research student manager. Then I've identified use cases for each of the actors based on their requirements, and documented specifications for each use cases. Based on these requirement findings and analysis results, I've designed five basic functionalities for the research student manager: (1) Student Information (student information like name, program of study and email shall be displayable to the admin), (2) Supervision Details (supervisor detail like supervisor name, faculty, student supervised shall be displayable to the admin), (3) Funding Details (project funds detailed information like source, and use status need to be tracked by the admin), (4) Assessment Stages (project information like current status, assessment results shall be traceable by the admin)and (5) Notifications (the admin shall be able send notifications to students and supervisors based on relevance). After this requirement analysis phase, I've completed designing the database for the research student manager in the following steps: 1. identifying all the entities (students, supervisor, project, notification) and their attributes. 2. Identifying relationship between all the entities and drawing an ER diagram (using Microsoft Visio). 3. Refining ER diagram and mapping it to a relational model which can be conveniently used for database implementation. Then based on the database design, I've done implementing the database in MySQL with database management tools (mainly phpMyAdmin, assisted by also managing via MySQL command line) | | | | |

Research Student Manager

| | | |
|---|---|---|
| 尚需完成的任务<br>Work to do | The database used by the research student manager has been completed designed and implemented, and all the basic functionalities of the system have been designed. Therefore, the work still need to do for now is to implement the http contents based on these functionalities with HTML+PHP, deploy and test them on a local server. The five basic functionalities, as designed into five basic modules of the system, will also be implemented and tested module by module. The development process, as planed since the beginning of this project, will be an agile and iterative process, agile to produce deliverable prototypes quickly for evaluation, and iterative to allow for future changes and refinements in functionalities. The process of developing and testing will also include connecting the system with the MySQL database, and achieving manipulation of the tables in the database (for testing purpose, the entries in database tables (e.g. student, table) will be randomly generated, though in real world situations, these are entered from other related information systems). The planned deliverable outcomes are: (1) the initial deliverable version of the research student manager will be accessible from client browser, all the five basic function modules should work properly and able to display data retrieved from the database. (2) The second version of the system will have been refined to minimize bugs and defects, it has been evaluated based on planned criteria. (3) The final version of the system will be well-evaluated and tested, and may include some extra functionalities that are further identified. At the end of the implementation phase, the whole system will be evaluated in terms of how well it satisfies the functional and non-functional requirements as planned in the specification. A final report of this project will also be completed based on all the documents that has been maintained since the beginning of the project. | |
| | 能否按期完成设计（论文）<br>Can finish the project on time or not: Yes/No | Yes |
| 存在的问题和解决办法<br>Problems and Solutions | 存在问题<br>Problems | In the requirement gathering process, the requirement findings gathered by interviewing with the administrative staff from BUPT did not provide much constructive results as expected. Most results from the interviews lead to overall non-functional requirements rather than functional requirements that help to design the system functionalities. As a result, most functional requirements derived currently have to be based on initial project specification and findings from relevant research papers and development reports. |
| | 拟采取的办法<br>Solutions | Further interviews and contacts with administrative staff from BUPT and QMUL will be carried out, with the requirement gathering techniques refined in order to obtain better results. For example, the interview questions may be revised to encourage more opinions provided on what functionalities are needed and expected, which helps to refine and improve the function requirements results. Also, the requirement findings can be better analysed by prioritizing them based the level of satisfaction that can be achieve, which means core requirements that lead to higher satisfaction will be attached with higher priority. Since the project is guided by an agile and iterative software engineering process, which has the adaptability to changes, the potential requirements refinements can then be reflected by redesigning, implementation and testing of the system modules. |
| 日期<br>Date | 03/03/2014 | |

# Risk Assessment

This section will talk briefly about assessments of risks and concerns associated with the design and implementation of the research student manager system. The risks are mainly introduced by the limit time constraint that may result in the system not fully developed. Also the development of the research student manager also introduce some risks with changing requirements at the initial stage, the discovery of more functionalities during middle stage of implementation, and at final stage when the system is completed. The risks analysis results will be presented here, the formula used is R = L. C. where R is the risk level and L is the occurrence likelihood, and C is the consequence. The results are in the table.

**Table 3: Risk Assessment of Research Student Manager**

| Risk | Impact | Likelihood Rating | Impact Rating | Prevention Actions |
|---|---|---|---|---|
| Technical issues with the database resulting in data can’t be accessed | The system will not be able to working without the database where all data are stored | 1 | 5 | If happens try to use other tools to fix the issue |
| Bugs with the system functionalities will result in system failure | Result in the functional modules failing to work, and user can’t access the system | 3 | 3 | If happens, will solve by examine the code |
| Problems with the local server will cause trouble for local testing | The developer will not able to continue developing and testing the system | 2 | 4 | Use alternative local machine is an option |
| Introducing new modules may require changing code in the existing system and have risks | Existing functional modules may be made unable to work | 4 | 2 | Can roll back to previous versions to reduce impact to minimum |
| Serious problems with the research student manager system | Result in system failure | 2 | 5 | Debug and find solutions to the problems |

# Environmental Impact Assessment

This section will analyse the impact of this project on the environment. It is always necessary to evaluation possible impacts on the environment of a design and build project, through it is evident that some projects may have very little environment impact with only software used.

### Manufacturing Cost

In this project, no hardware is manufactured, so there's no cost of making the hardware that may introduce metal pollution to the environment.

### Waste Disposal and Recycle

This project has no waste to dispose, and none resources need to be recycled. This is because most part of the system built in this project is software loaded on the local testing computer.

### Energy Use and Energy Saving

The research student manager system, when introduced in practical use, may increase the cost of electronic energy as the server system should be always held online on a 24 hour running server machine. This may introduce energy cost to electricity. However, as this is inevitable to make sure the server is on, this cost can only be made more efficiently with more efficient energy consumption.