Diwas Lamsal
UN ID – 18406547

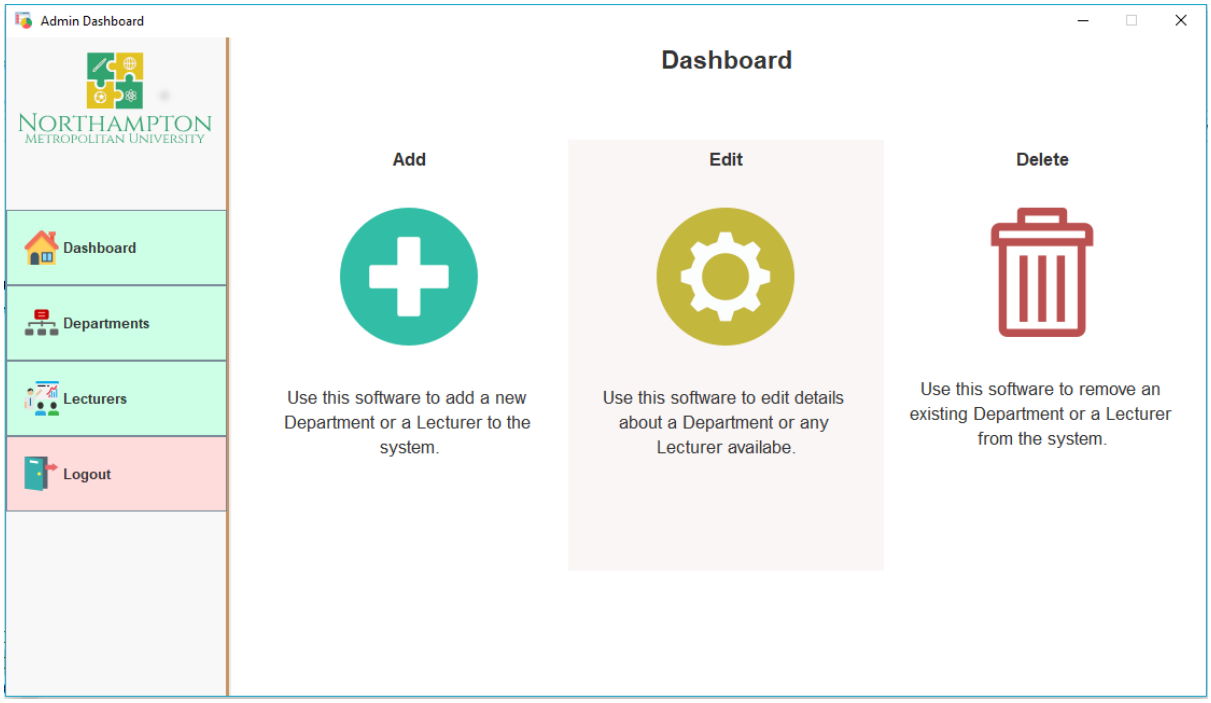
CSY2030

Systems Design and Development Term II Assignment

HR Management System

Username: admin

Password: admin


Admin Dashboard
NORTHAMPTON
METROPOLITAN UNIVERSITY
Dashboard
Departments
Lecturers
Logout
Dashboard
Add
Use this software to add a new Department or a Lecturer to the system.
Edit
Use this software to edit details about a Department or any Lecturer availabe.
Delete
Use this software to remove an existing Department or a Lecturer from the system.

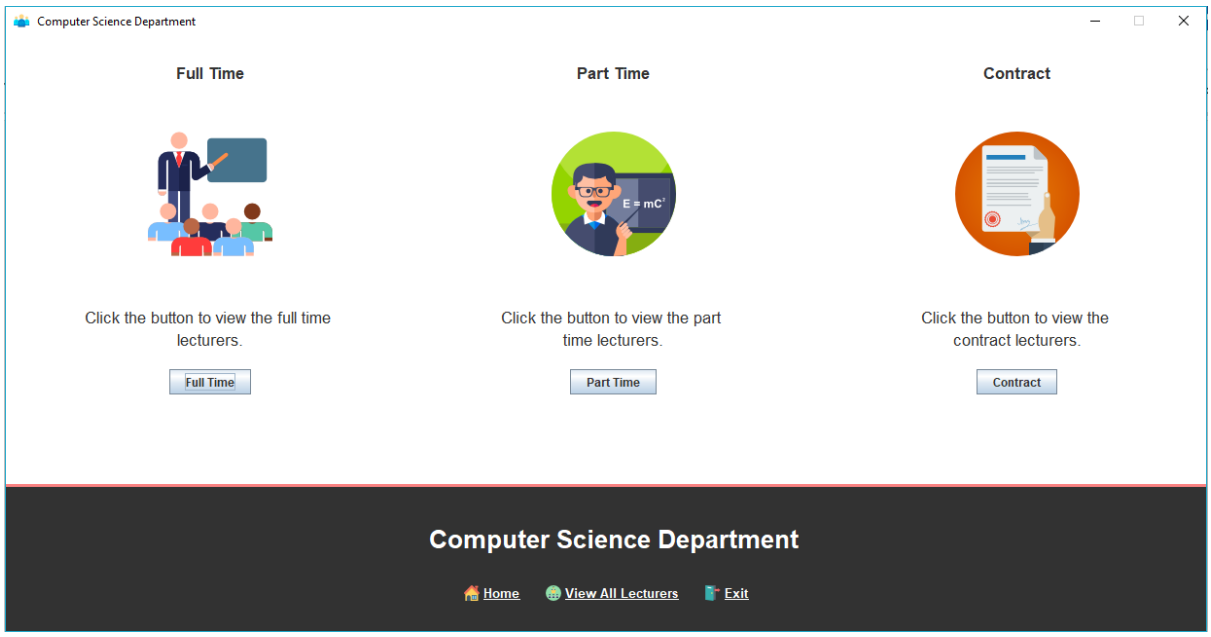

Computer Science Department
Full Time
Click the button to view the full time lecturers.
Full Time
Part Time
Click the button to view the part time lecturers.
Part Time
Contract
Click the button to view the contract lecturers.
Contract
Computer Science Department
Home
View All Lecturers
Exit

Table of Contents

List of Figures

1. INTRODUCTION

1.1 Project Background

The CSY2030 Assignment Term II asks for a Java Swing based GUI application. The purpose of the application would be to allow a university based human resource management. The system, which would be designed for the Northampton Metropolitan University, should allow the Human Resource staff to keep track of the lecturers and the various departments in the university. The system should also enable secretaries to view the different lecturers associated to their department.

1.2 Completed Project Requirements

Following is a list of all the completed features mentioned in the brief.

Basic System Requirements:

The system must, using a GUI, allow human resource staff to do the following:

1. Log onto the system with a user name and password
2. Exit the system
3. Add a new department
4. Change details of a department
5. Delete a department
6. Add a new lecturer of any type (full-time, part-time or contract) and allocate them to a department
7. Change details of a lecturer
8. Delete a lecturer

The system must, using a GUI, also allow departmental secretaries to do the following:

9. Log onto the system with a user name and password that is stored with each department
10. Exit the system
11. Show details of the full-time lecturers
12. Show details of all the part-time lecturers
13. Show details of all the contract lecturers
14. Query a specific lecturer's details including what type of lecturer they are (full-time, part-time or contract) by their staff identification number
15. Show details of all the details of all the lecturers (including their type) in their department

Additional System Requirements:

16. Exit the system and write all objects to file so they can be reloaded when the system is run again
17. An intuitive GUI
18. Appropriate exception handling
19. Use the Model View Controller
20. Use of Collections

1.3 Report Structure

- **Introduction:**
  The introduction section of the report presents the project's backgrounds and the major tasks associated with it. It also showcases the list of tasks completed and the structure of the report.

- **System Design:**
  The UML class diagram and the documented use case diagram for the proposed HR Management System are in this section. This part discusses the design techniques utilized and different approaches used to produce the system efficiently.

- **System Description:**
  The final product's description in the form of a user manual/guide is presented in this section along with any relevant screenshots. Going through this section would allow a user to use the system without any problems.

- **Testing:**
  This part of the report shows the test results from different kinds of tests performed. The tests include Black Box Testing and White Box Testing.

- **Evaluation:**
  This part of the report lists any bugs or weaknesses found in the system and their proposed solutions. It also shows a list of strengths of the solution.

- **Conclusions and Recommendations:**
  The feedbacks about using the Java swing library and the acquired knowledge and skills from this project are included in this section. Any different approaches that would be used, or new features that would be added to the system if more time was available are also listed in the Conclusion section.

- **References:**
  References for any extra tools used or code taken from somewhere are listed in this appendix.

2. SYSTEM DESIGN

2.1 System Diagrams

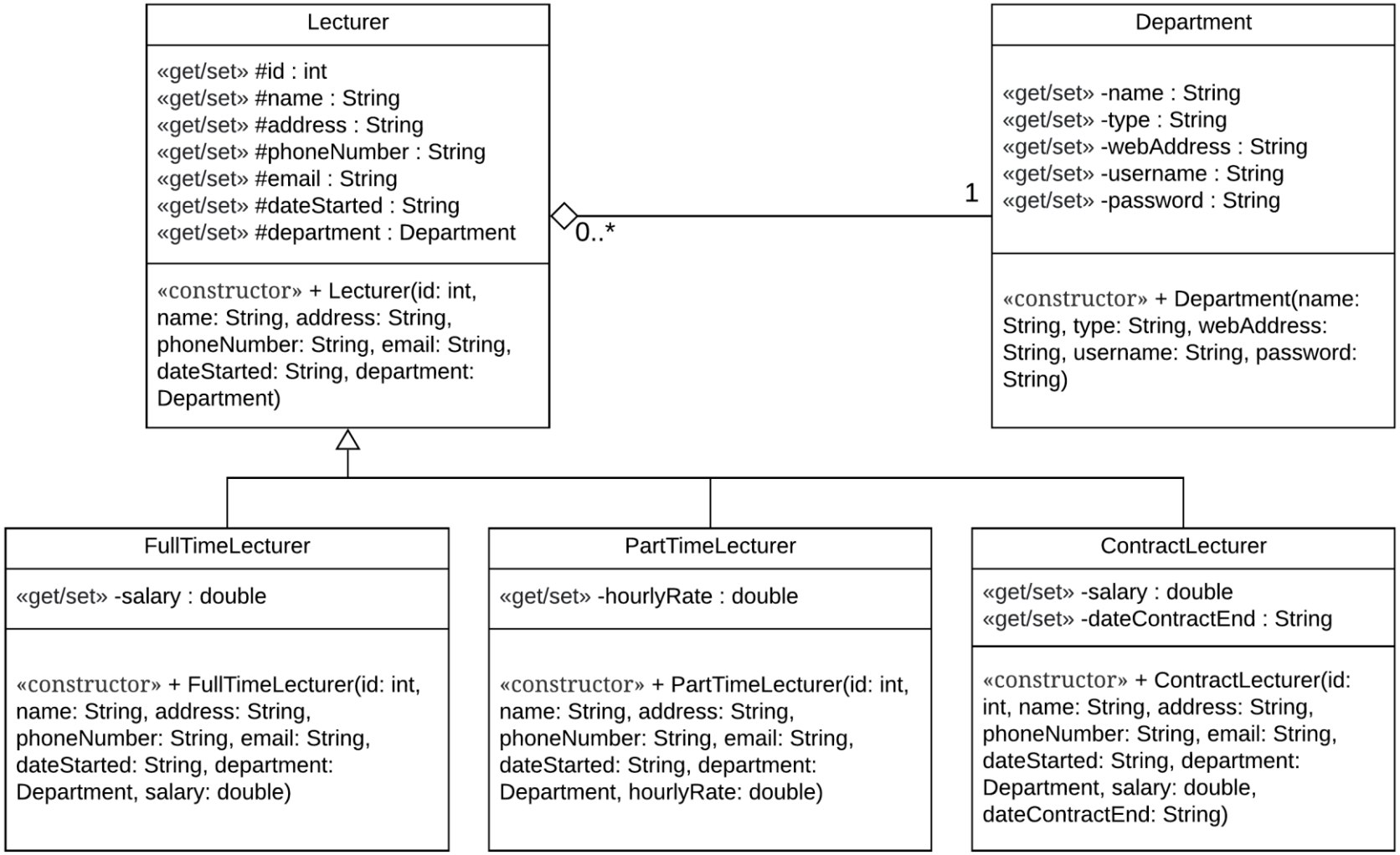


Figure 2. 1 - Class Diagram

Figure 2.1 shows the class diagram of system. The Department class holds the information about a department including the secretary username and password. The objects of this class are stored such that, the name of the department would allow unique identification (primary keys). Similarly, the Lecturer class is the super class for three categories of lecturers. The common type of data is used by the child classes. Each other kinds of lecturers have some extra properties unique from other lecturer types. The multiplicity shows that a department can have zero to many lecturers associated with it and a lecturer can be associated to only one department. The aggregation shows that a lecturer should have a department, but the relationship is not compulsory.

The Use Case Diagram:

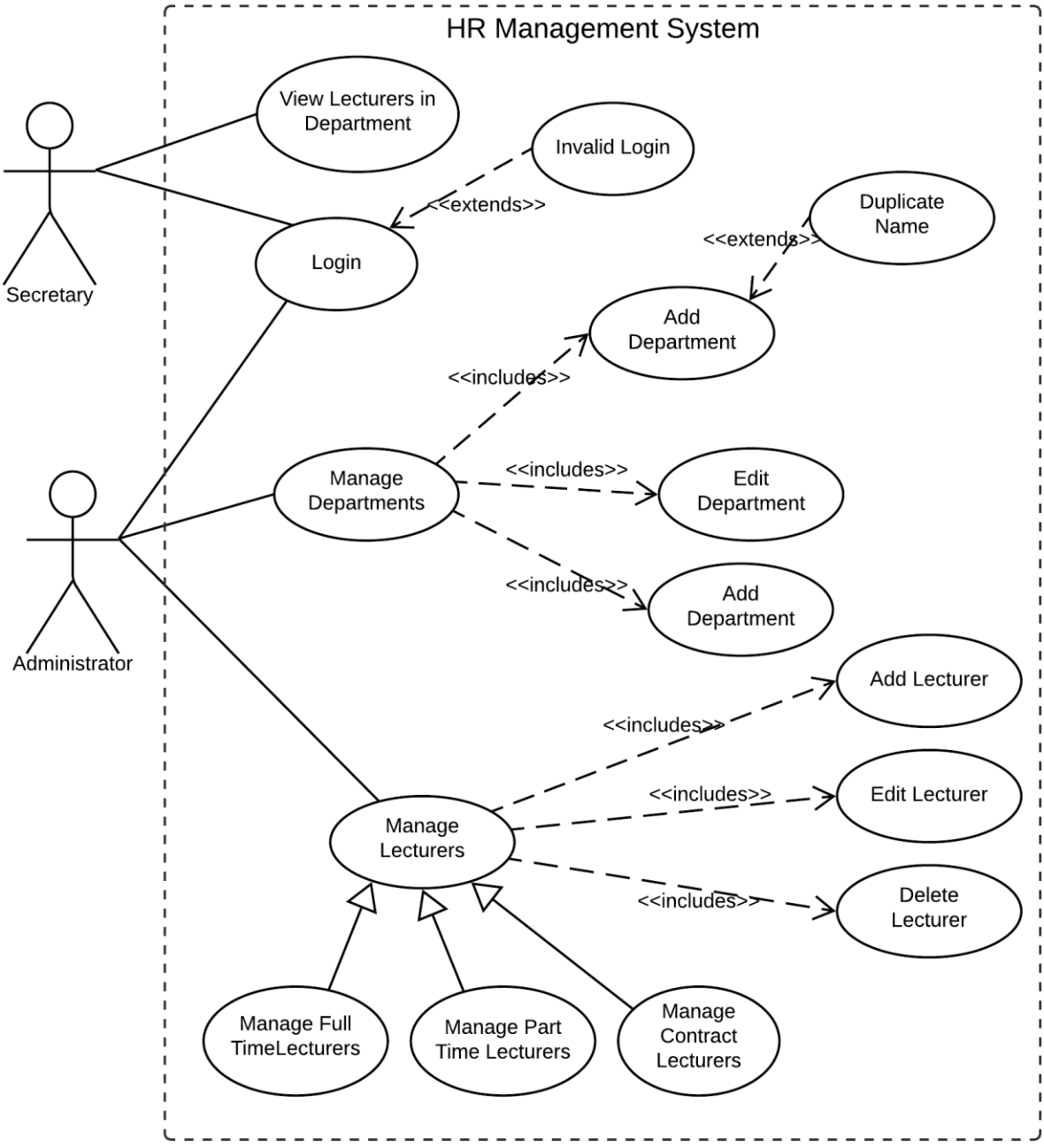


Figure 2. 2 – Use Case Diagram

Documentation of Use Cases

i. Administrator Login

Identifier and name: UC1 Login
Initiator: Administrator
Goal: Administrator logs into the system
Pre-condition: Nobody is currently logged in and the login window is open
Post-condition: Administrator logs into the system
Assumptions: Administrator enters valid username

Main Success Scenario:
1. Administrator opens the system
2. Administrator enters the username and password
3. System validates the login credentials
4. If valid, the system logs the administrator into their account

Extends: *Invalid Login*

ii. Administrator Manage Departments

Identifier and name: UC2 Manage Departments
Initiator: Administrator
Goal: Administrator performs desired actions on Departments
Pre-condition: Administrator is logged in
Post-condition: Administrator performs desired actions on Departments
Assumptions: Administrator enters valid data while performing desired actions

Main Success Scenario:
1. Administrator navigates to the departments area
2. Administrator attempts to perform an action
3. If valid, the action is performed

Includes: *Add Department, Edit Department, Delete Department*

iii. Administrator Manage Lecturers

Identifier and name: UC3 Manage Departments
Initiator: Administrator
Goal: Administrator performs desired actions on Lecturers
Pre-condition: Administrator is logged in
Post-condition: Administrator performs desired actions on a type of Lecturer
Assumptions: Administrator enters valid data while performing desired actions

Main Success Scenario:
1. Administrator navigates to the lecturer area
2. Administrator selects a specific type of lecturer
2. Administrator attempts to perform an action
3. If valid, the action is performed

Includes: *Add Lecturer, Edit Lecturer, Delete Lecturer*

iv. Secretary Login

Identifier and name: UC4 Login
Initiator: Secretary
Goal: Secretary logs into the system
Pre-condition: Nobody is currently logged in and the login window is open
Post-condition: Secretary logs into the system
Assumptions: Secretary enters valid username

Main Success Scenario:
1. Secretary opens the system
2. Secretary enters the username and password
3. Secretary checks the secretary radio button
4. System validates the login credentials
5. If valid, the system logs the secretary into their account

Extends: *Invalid Login*

v. Secretary View Lecturers

Identifier and name: UC5 View Lecturers in Department
Initiator: Secretary
Goal: Secretary views lecturers in their department
Pre-condition: Secretary is logged into the system
Post-condition: Secretary views the lecturers in their department
Assumptions: None

Main Success Scenario:
1. Secretary navigates to the lecturer area
2. Secretary views the required details about lecturers

2.2 System Design

The system is designed following the MVC pattern. The models, views and the controllers go into their separate packages. The model deals with the data such as storing Department or Lecturer information, retrieving the information, updating the information or deleting the information. The controller is like a mediator between the view and the model. The event handlers, written in the controller, update the view. The view retrieves data through the Controllers and not directly from the model. It is done such that without changing the view by any means, a different controller can be written for it which would handle the events differently.

There are three kinds of views with their respective controller and model. This separates the logics and GUI information of the views. The views, controllers, and models are separated for Login, Administrator Dashboard and the Secretary area. In this way, for example: the controller would have to handle events only for their own views, which would prevent conflicting variable names between views from affecting the system logic.

Likewise, the classes are separated into their own package named entities. These classes Department, Lecturer, FullTimeLecturer, PartTimeLecturer and ContractLecturer hold the data variables which would be used by the system. These are instantiated and stored in the system (How the departments and lecturers are added from GUI perspective is talked about in the System Description area) in the form of a binary (.dat) file. The binary file would contain serialized objects stored in ArrayLists. The collection type is used here which would make it easier to manipulate the data within it. For example: to edit a particular department in an ArrayList, you would only have to find the index of the department, make the necessary changes, and write the list of objects back to the file. Doing so directly would require rewriting all the objects one by one to the binary file.

In a similar manner, a separate package is included to hold the Exception classes. Three user defined exception classes namely AlreadyExistsException, InvalidLoginException and NullStringException have been added to the package. These exceptions are thrown and handled according to need.

The main method is called in the default package's HRManagementSystem class. This creates instances of the Login model view and controller.

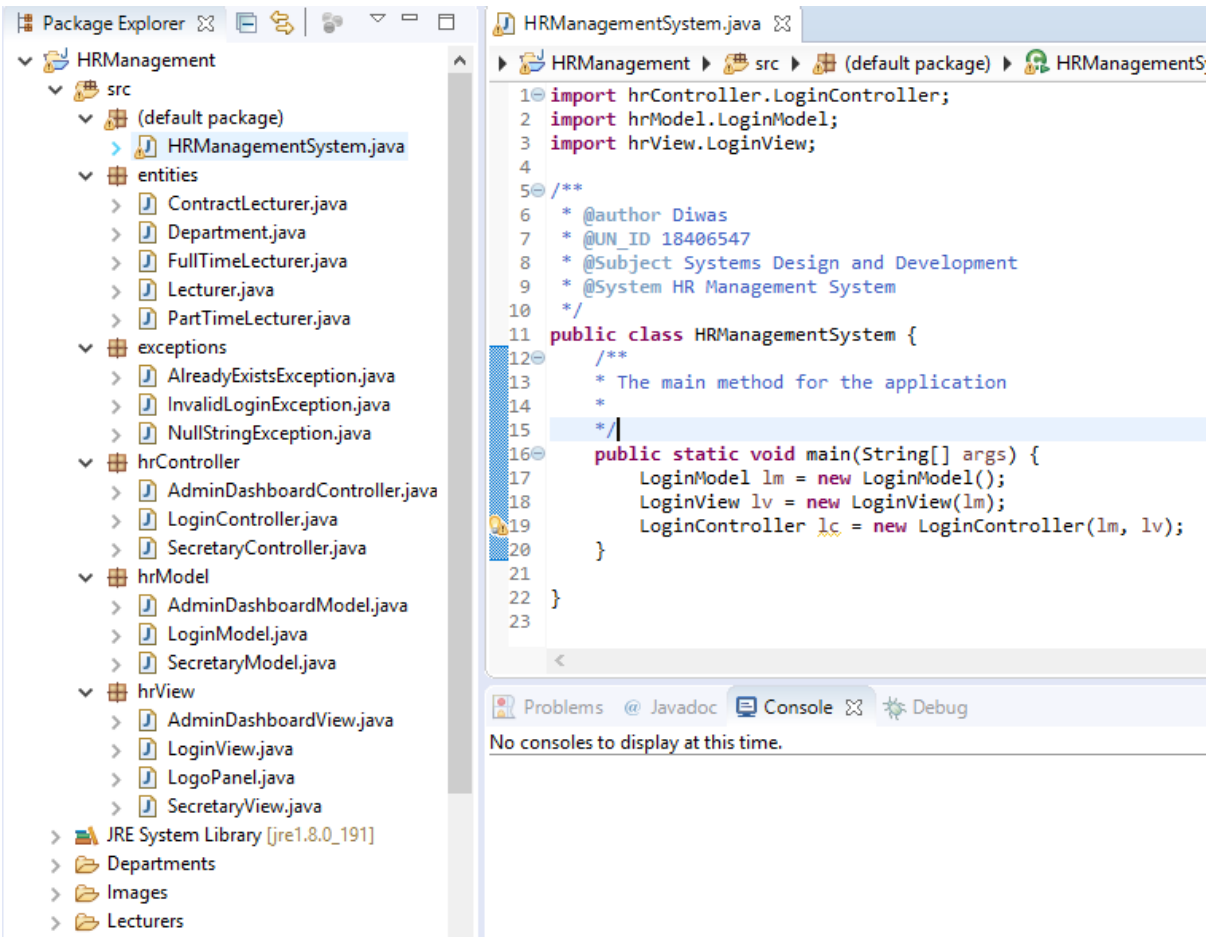


Figure 2. 3 - Package Explorer Eclipse

Inside the views, the main display panel contents are replaced with user interactions. For example: the administrator dashboard view contains a single main panel in which the contents are changed when the user navigates to lecturers or departments or performs other actions in those areas.

In such ways, the system has been designed to work and fulfill all the required criteria in an efficient way. A lot of code is reused, and it is easy for another developer to read and modify the code in attempt to amend or add some features.

3. SYSTEM DESCRIPTION

This part of the report acts as a user manual providing required information to use the system. The information regarding types of users and what tasks they can perform are discussed briefly. Detailed information regarding how to use the system will be covered as a tutorial in the video demo.

3.1 User Types

There are two types of users available in the system. The Administrator is responsible for managing all the available departments and lecturers. They are allowed to add, edit or delete departments and any specific type of lecturers. They are also allowed to assign a lecturer to a department. Whereas, the other type of users, the Secretaries, are allowed only to view the lecturers that are assigned in their department. They would not be able to assess any data related to another department whatsoever.

3.2 Login System

Running the application through the jar file or directly from the main method brings up the login menu. The user should enter correct username and password combinations in order to enter the system. The user needs to select correct radio button options to let the system know what kind of user they are. If the login is invalid, a popup is displayed showing the information whereas successful login attempt will show a message and take the user to their user space.

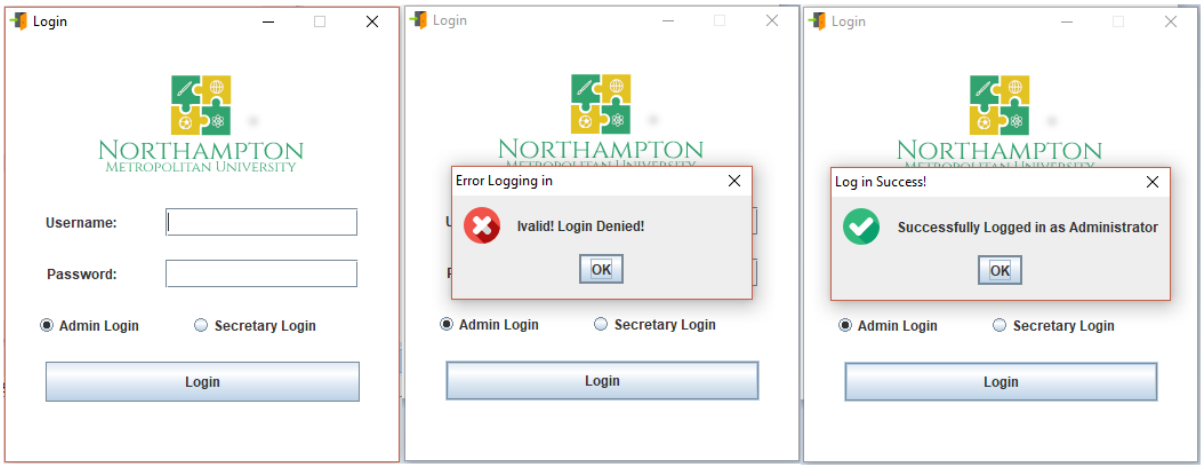


Figure 3. 1 – Login System

3.3 Administrator Features

The administrator, as discussed before, have access to all the crucial aspects of the system. They can view, add, edit and delete the entities about which the data is stored in the system. Administrator is the one who sets secretary usernames and passwords.

The Dashboard:

After successfully logging into the system as an administrator, the system takes the user to the Dashboard. The dashboard displays a navigation menu to the left and in the main dashboard area, three boxes showing what the user can do through the application are displayed.

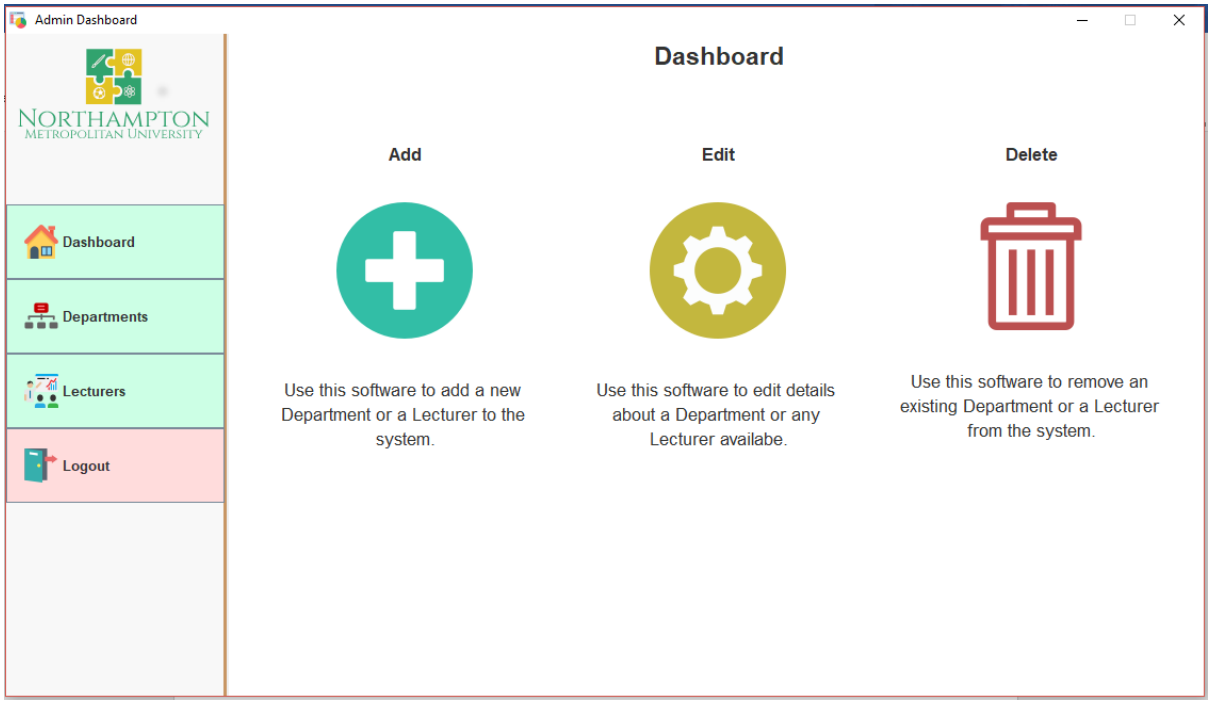


Figure 3. 2 – Administrator Dashboard

Departments:

The administrator can click on the departments navigation button to view the departments. A table of departments is displayed on doing so (Figure 3.3). The administrator can click Add new Department button to add a new department, or, click on any specific department in the table to edit the department. Doing so would bring up a form for departments. If the user opts for adding a new department, the form would be blank whereas clicking on any one department would bring up a filled-up form. The form to edit a department also contains a delete department button. Clicking the button would remove the selected department from the system. The secretary username and password are set using this form, which can be used by a secretary to log in to their own user space.

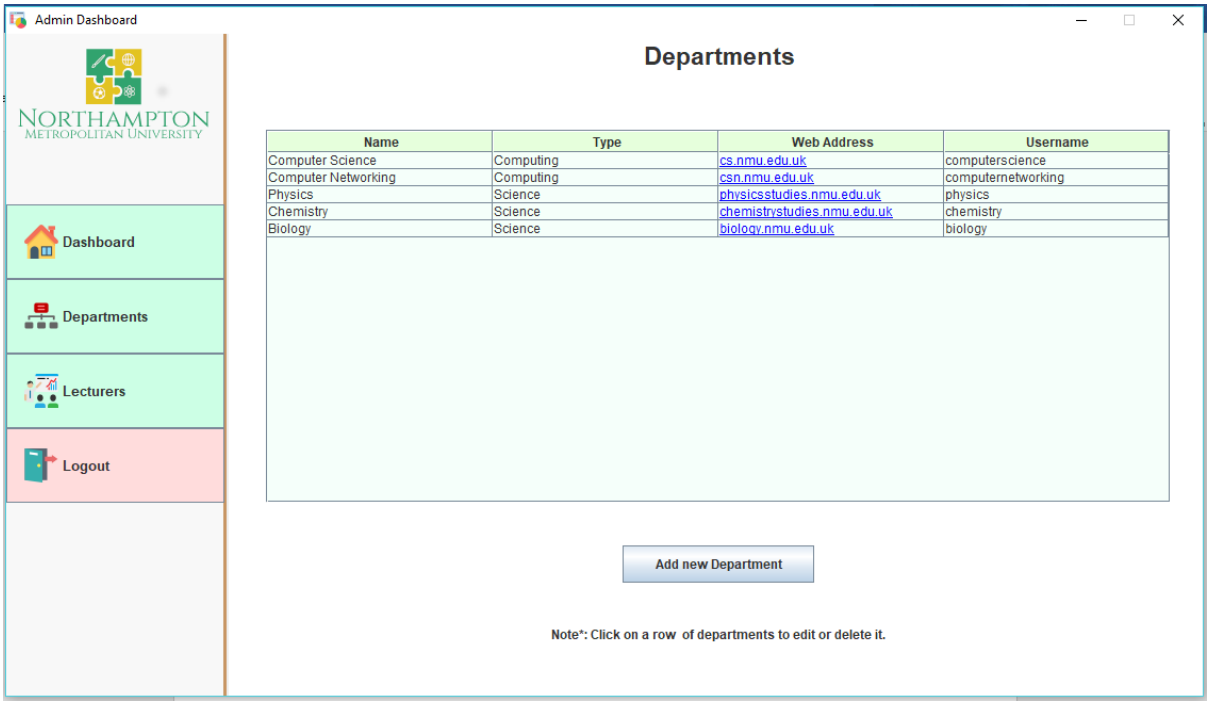


Figure 3. 3 – Departments View

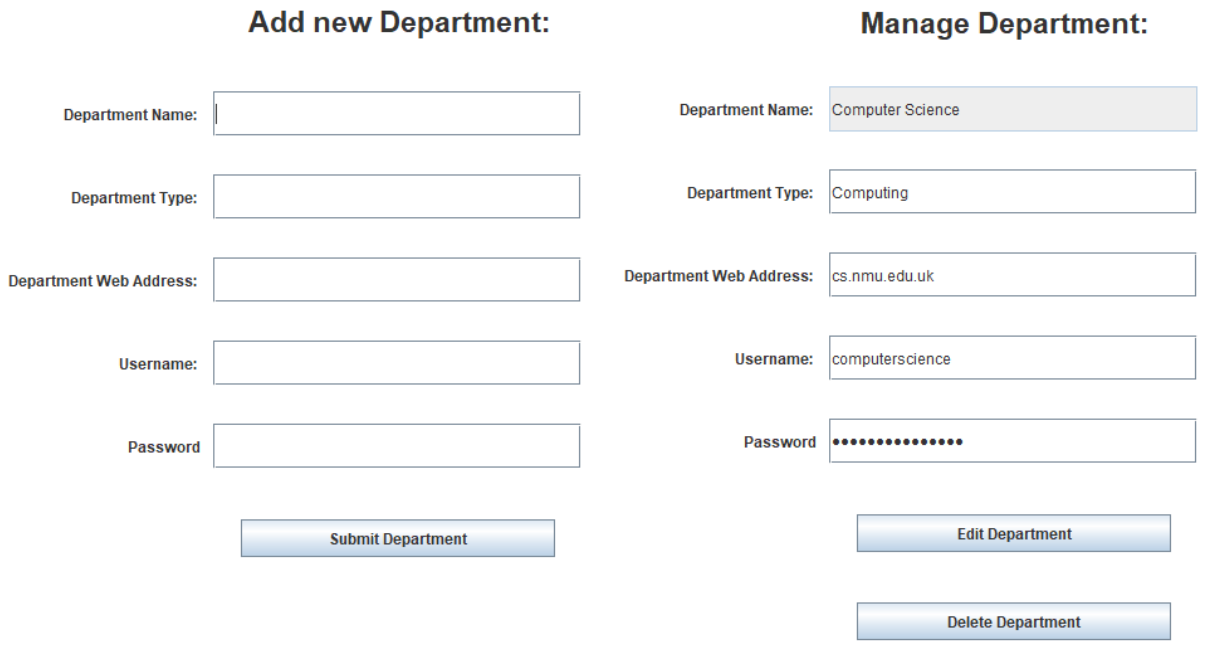


Figure 3. 4 – Add or Manage Department

Lecturers:

Like the departments, the administrator can navigate to the Lecturers area. There are three types of lecturers available, thus, these different lecturers are displayed in separate areas. Selecting the lecturers menu, the application would display three boxes showing three lecturer types from which the user can choose one (Figure 3.5). When the Administrator selects any one type of lecturer (Full Time Lecturers used as example in this guide), a list similar to that of departments is displayed showing details about lecturers. Likewise, the Add new Lecturer button can be clicked to add a lecturer, or a specific lecturer can be selected from the list to edit them. The unique option whilst adding a new lecturer would be selecting a department. The form relating to lecturers allows the administrator to chose from a dropdown, the department the lecturer is associated with.

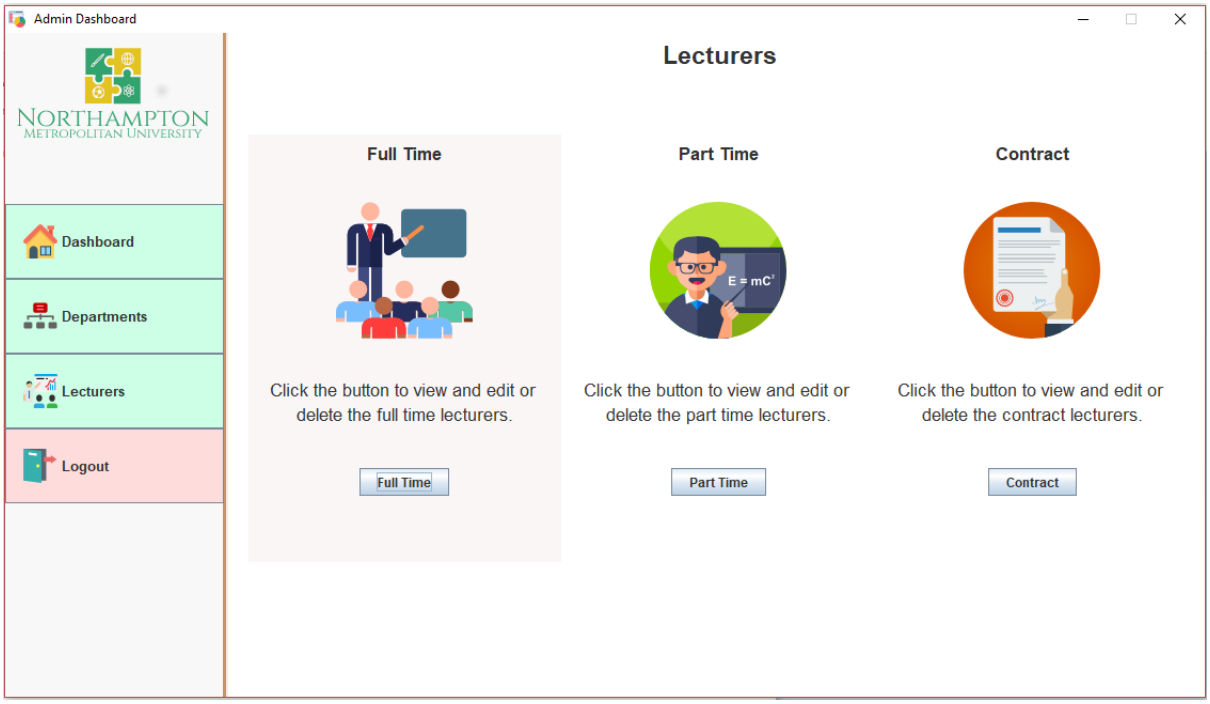


Figure 3. 5 – Lecturers View

ID	Name	Address	Department	Phone	Email Address	Date Started	Salary (£)
1	Nischal Khadka	New Baneshwor	Computer Science	9818181818	nischal@nmu.edu...	2018-08-20	1000.0
2	Deepak Kumar Ka...	Jorpati	Computer Networ...	9818252525	deepak@nmu.ed...	2014-07-08	2000.0
3	Nishant Neupane	Balaju	Computer Science	9825253515	nishant@nmu.ed...	2016-12-08	1000.0
4	Khagendra Sah	Chabahil	Computer Networ...	9808022336	khagendra@nmu....	2016-12-12	1500.0
5	Himalaya Kaksha...	Gongabu	Computer Science	9841424256	himalaya@nmu.e...	2015-07-12	1500.0
6	Sajit Chandra Sha...	Maitidevi	Physics	9853206825	sajitchandra@nm...	2014-08-06	1200.0
7	Balaram Katwal	Anamnagar	Physics	9860252326	balramkatwal@n...	2015-02-26	1200.0
12	Mamata Bhattrai	Kalanki	Computer Science	9843526523	mamata@nmu.ed...	2013-10-10	1500.0

Figure 3. 6 – Full Time Lecturer List

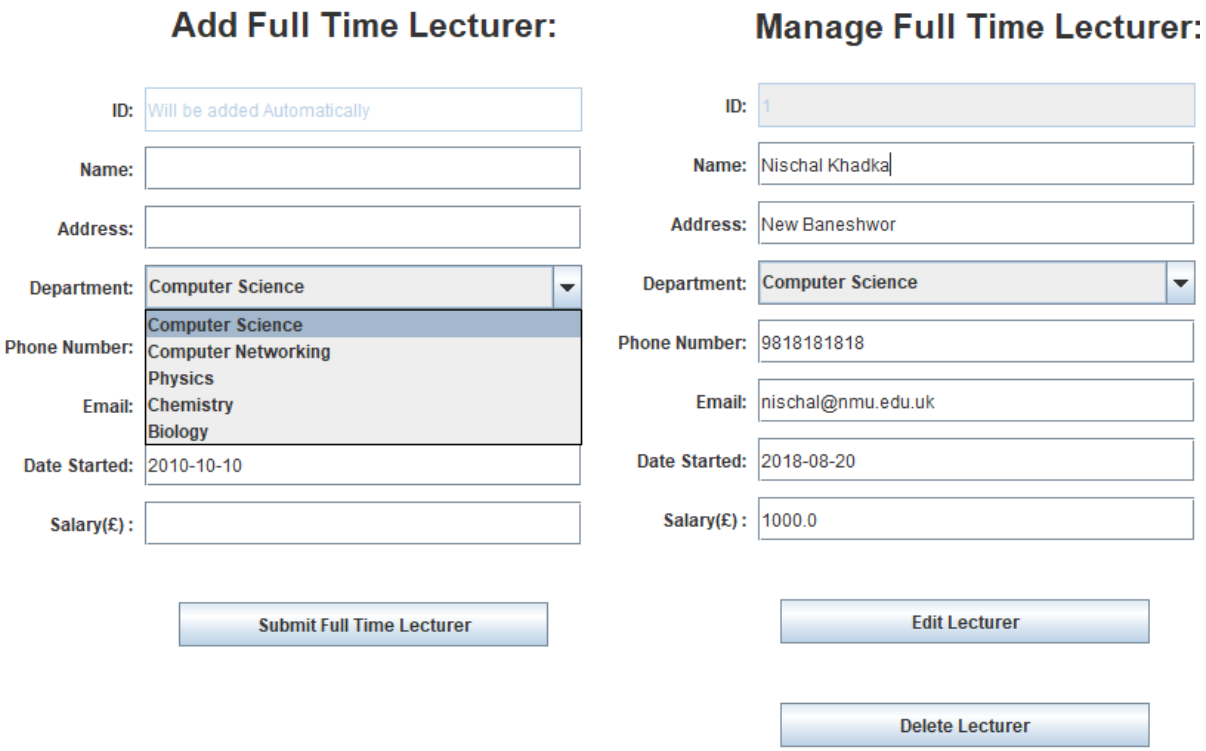


Figure 3. 7 – Add or Manage Full Time Lectuerer

Form Validation:

The point that is applicable for both the departments and lecturers is that, the administrator is not allowed to submit a blank form or partially filled form. Also, the salary and hourly pay for the lecturers should be a valid number and not text or invalid characters. Both the types of data contain unique identifiers. For the lecturers, it is their ID and it cannot be changed. For the departments, it is their department name – two different departments cannot have the same name either. It is made so that two departments cannot have the same username. It was done to prevent conflict for secretaries to log in. Relevant error messages are displayed when the administrator tries to perform an action that is not allowed.

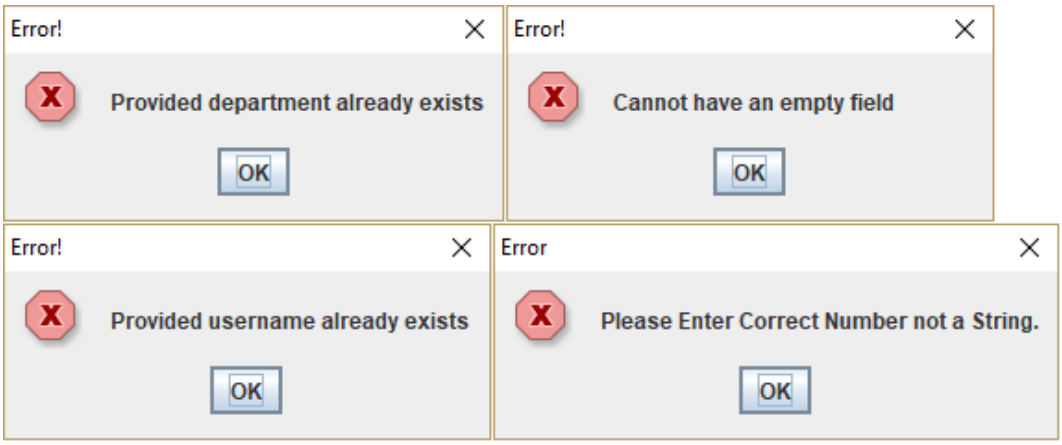


Figure 3. 8 – Form Errors

3.4 Secretary Features

The features available for a secretary are quite less in comparison to the administrator. The secretaries can only assess the data of lecturers who are assigned to their department. From the administrator guide, we learned that each department is provided a unique username and password. This allows a secretary to log in to their department. Logging into the system as a secretary displays the secretary dashboard. It shows three boxes showing the types of lecturers, a title showing their department, and some links (Figure 3.8). The secretary can select any one type of lecturer to view or click the view all lecturers link. This displays a list of lecturers available in the department and of selected category. The secretary can then click on any one specific lecturer to view their details.

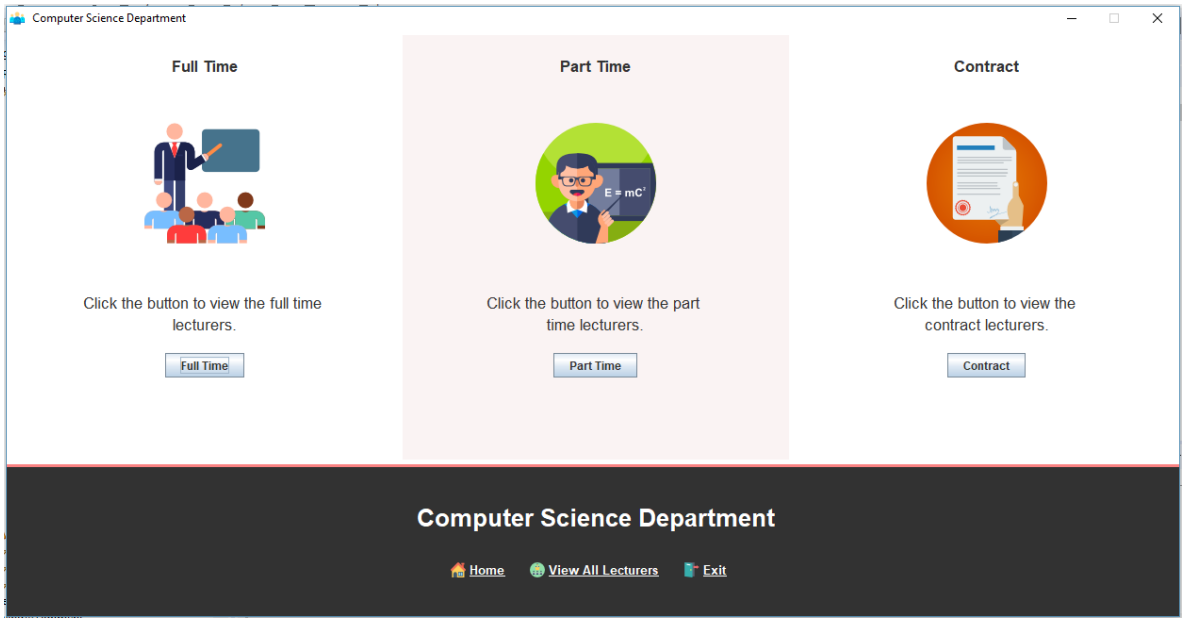


Figure 3. 9 – Secretary Dashboard

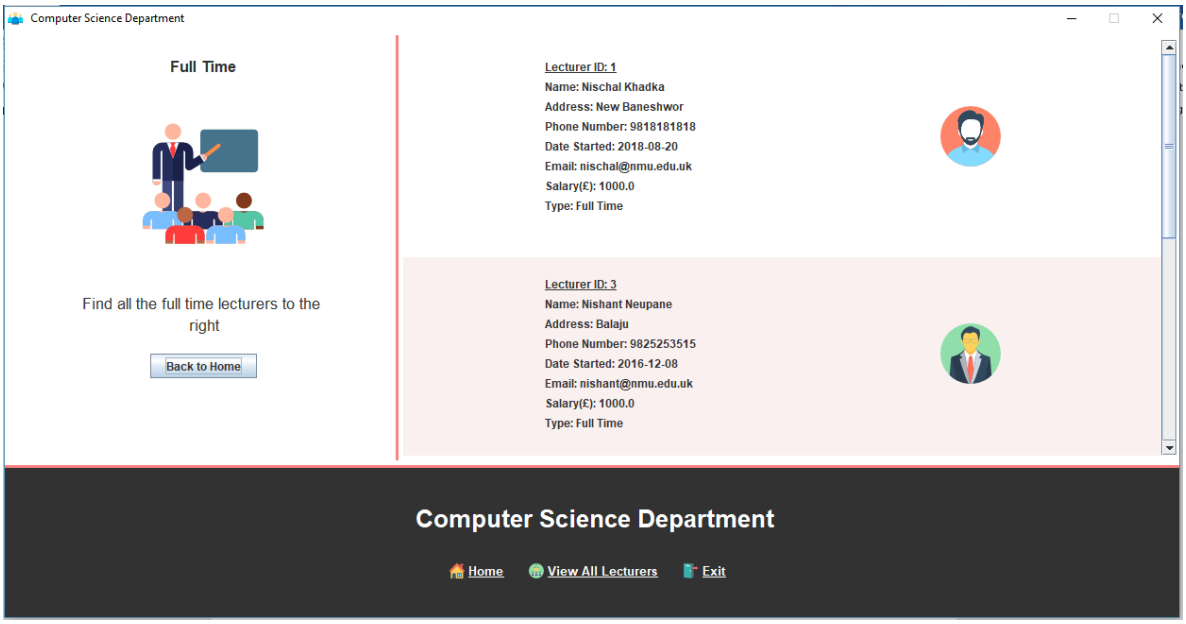


Figure 3. 10 – Secretary View Full Time Lecturers

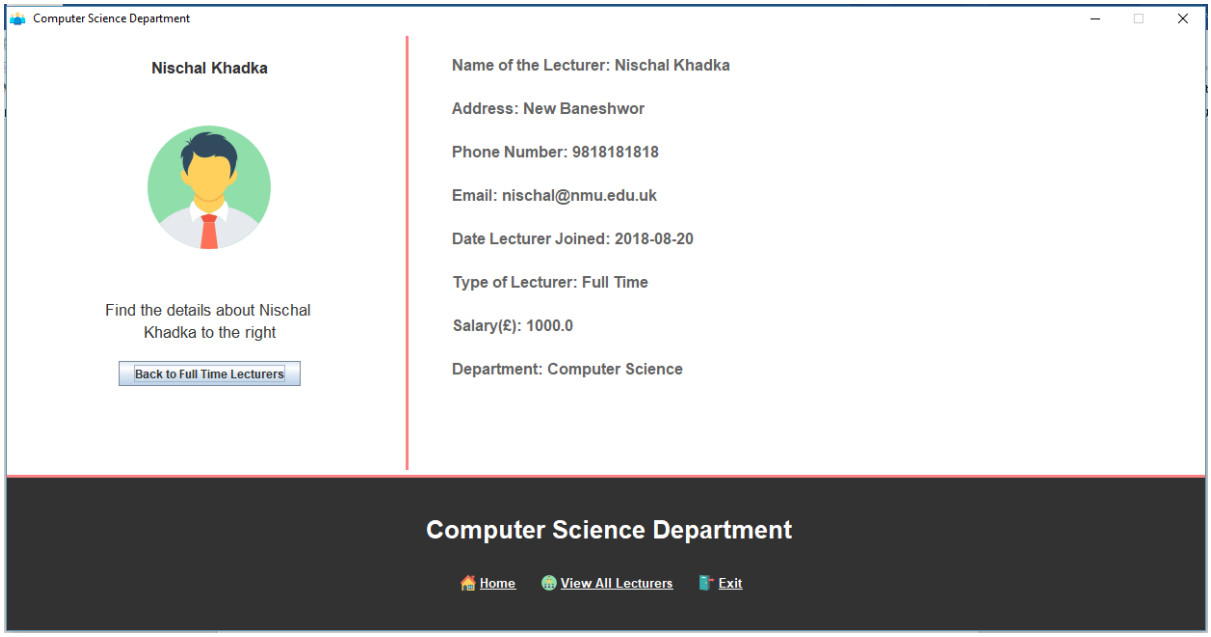


Figure 3. 11 – Secretary View Particular Lecturer

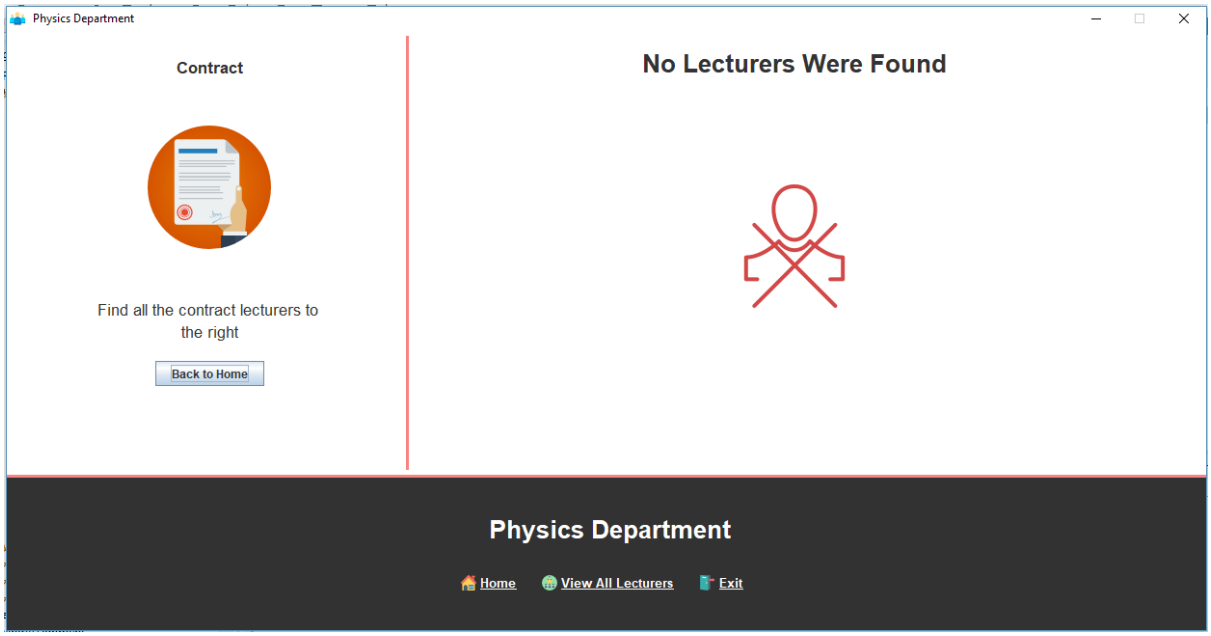


Figure 3. 12 – No Lecturer Found

4. TESTING

Two kinds of tests have been performed to test the usability of the system and for any kind of bug or weakness. The black box test involves testing of different aspects of the program and covers vital parts of the application. The white box test involves testing of decision points in four of the methods in the assignment.

4.1 Black Box Testing

The black box testing involves testing the program without access to any of the internal code structure. Essential functionalities in the application have been tested using this test strategy. The log provides information regarding functional tests and regression tests after fixing any found bugs. Any relevant screenshots have been added after the test log.

Table 4.1 – Login Test

	Test Case	Steps to Test	Expected Outcome	Actual Outcome
1.	Administrator login using correct information	• Administrator enters correct login information. • Administrator clicks Login button.	Successfully logged in message is displayed and the administrator is taken to the dashboard.	Same as expected.
2.	Administrator login using incorrect information	• User enters incorrect information. • User clicks Login button.	Invalid login message is displayed.	Same as expected.
3.	Administrator login selecting Secretary Login radio button	• Administrator enters correct information. • Administrator selects the Secretary Login radio checkbox. • Administrator clicks Login button.	Invalid login message is displayed.	Same as expected.
4.	Secretary login using correct information.	• Secretary enters correct login information. • Secretary selects the Secretary Login radio checkbox. • Secretary clicks Login button	Successfully logged in message is displayed and the secretary is taken to the secretary area.	Same as expected.

5.	Secretary Login using incorrect information.	• User enters incorrect information. • User clicks Login button.	Invalid login message is displayed.	Same as expected.
6.	Secretary login selecting Admin Login radio button	• Secretary enters correct information. • Secretary selects the Admin Login radio checkbox. • Secretary clicks Login button.	Invalid login message is displayed.	Same as expected.

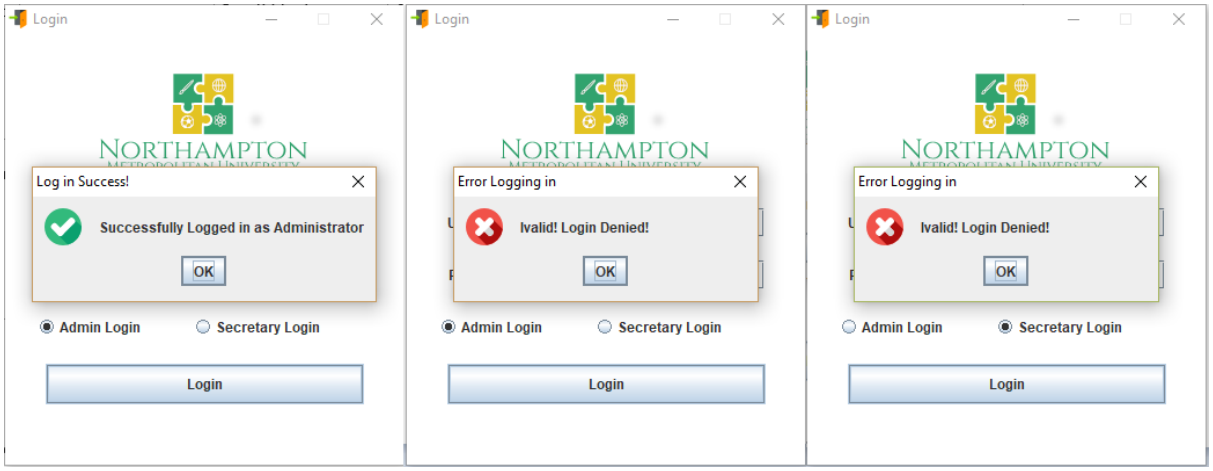


Figure 4. 1 – Administrator Login Tests

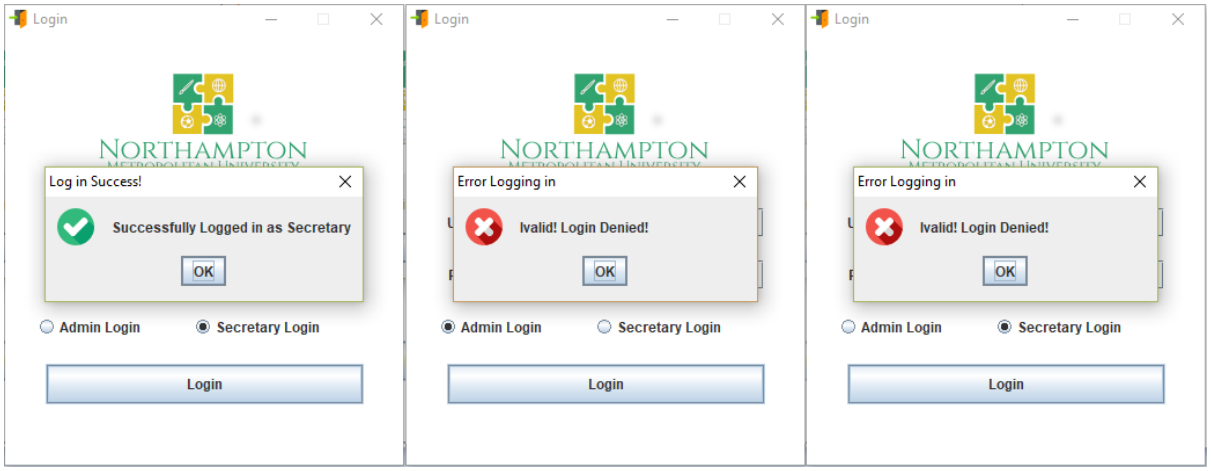


Figure 4. 2 – Secretary Login Tests

Table 4.2 – Administrator Department Tests

	Test Case	Steps to Test	Expected Outcome	Actual Outcome
1.	Administrator goes to the Departments area	• Administrator logs in • Administrator clicks the Departments navigation button	A list of all the added departments is displayed showing all the details in a table. A button that says Add new Department is displayed below the table.	Same as expected.
2.	Administrator clicks Add new Department button	• Administrator goes to the department area • Administrator clicks Add new Department Button.	An empty form showing all the department fields is shown	Same as expected
3.	Administrator clicks on a line of department from the table	• Administrator goes to the department area • Administrator clicks a line of department from the table.	A filled form that contains the data from the selected department is shown. The Edit Department and Delete Department buttons are shown underneath.	Same as expected
4.	Administrator deletes department	Administrator deletes a department	• Administrator selects a department to edit • Administrator clicks the Delete Department button • Administrator clicks the Yes button from the prompt	A prompt showing whether the user is sure to delete the department or not is shown. The department is deleted when Yes button is clicked
Add new Department				
5.	Administrator fills up all the details and	• Administrator goes to the form to add a new department.	Successfully added department message is displayed and the administrator is	Same as expected

	submits the form.	• Administrator fills up all the details • Administrator clicks Submit Department button	taken to the Departments list	
6.	Administrator fills up only partial fields and submits the form.	• Administrator goes to the form to add a new department. • Administrator fills up partial details • Administrator clicks Submit Department button	Error message saying fields cannot be empty is shown	Same as expected
7.	Administrator uses existing department name and fills up all the remaining fields correctly.	• Administrator goes to the form to add a new department. • Administrator fills up all the details but uses existing name of another department • Administrator clicks Submit Department button	Error message saying department already exists is shown	Same as expected
8.	Administrator uses existing username and fills up all the remaining fields correctly.	• Administrator goes to the form to add a new department. • Administrator fills up all the details but uses existing username of another department • Administrator clicks Submit Department button	Error message saying username already exists is shown	Same as expected
Edit Department				
9.	Administrator edits a department correctly	• Administrator clicks a line of department from the table. • Administrator edits some fields.	Success message showing successfully edited department is shown	Same as expected

		• Administrator clicks Edit Department button		
10.	Administrator erases some field data while editing a department	• Administrator selects a department to edit • Administrator erases some field data • Administrator clicks Edit Department button	Error message saying fields cannot be empty is shown	Same as expected
11.	Administrator tries to use an existing username while performing an edit	• Administrator selects a department to edit • Administrator fills up all the details but uses existing username of another department • Administrator clicks Edit Department button	Error message saying username already exists is shown	Same as expected

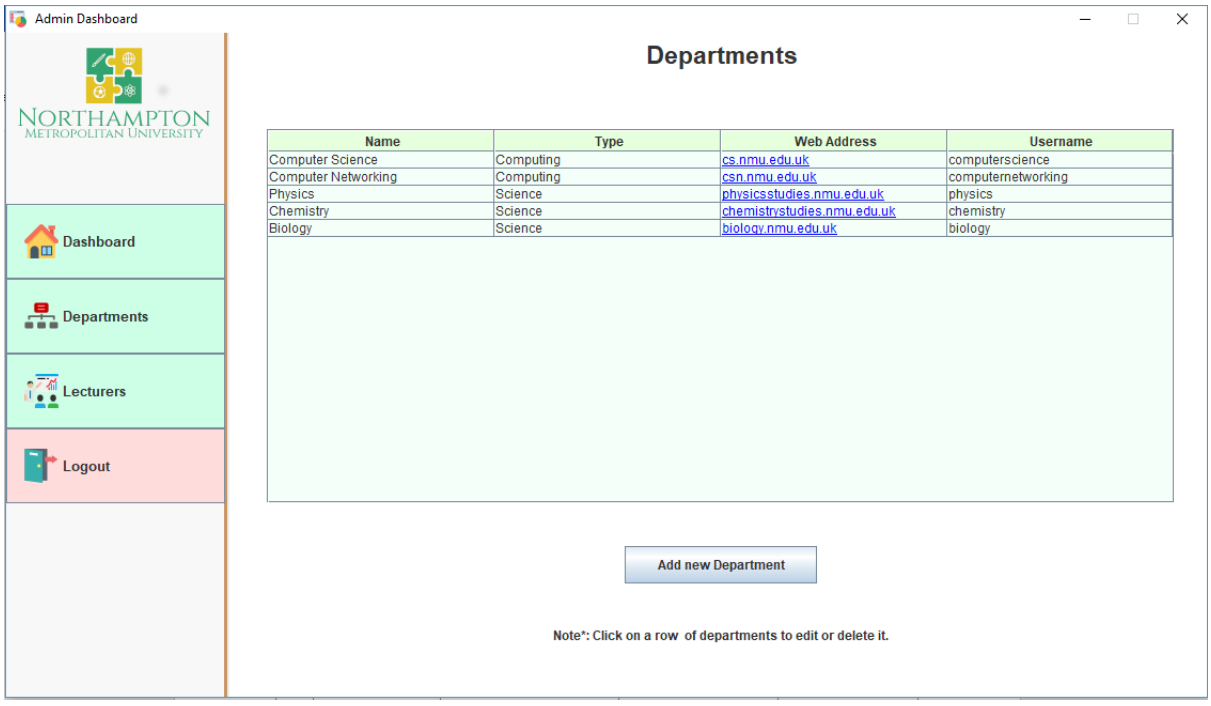


Figure 4. 3 – Administrator Dashboard: Departments

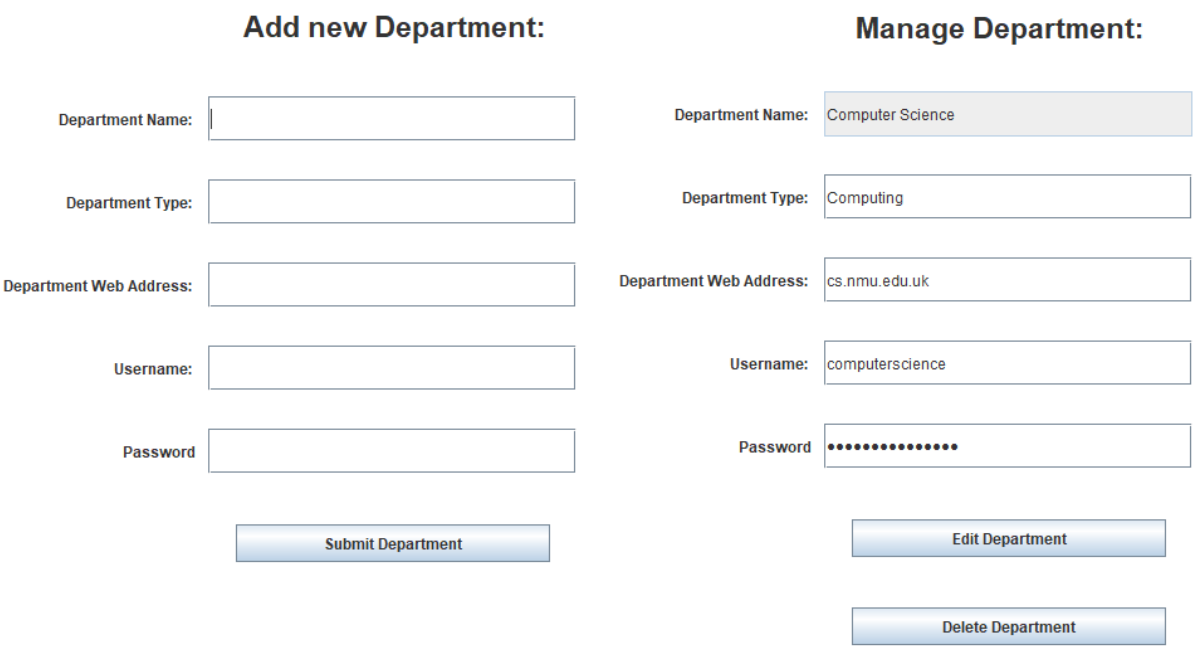


Figure 4. 4 – Administrator Dashboard: Add or Edit Department

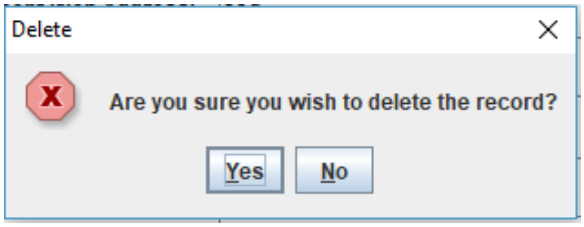


Figure 4. 5 – Administrator Dashboard: Delete Department Prompt

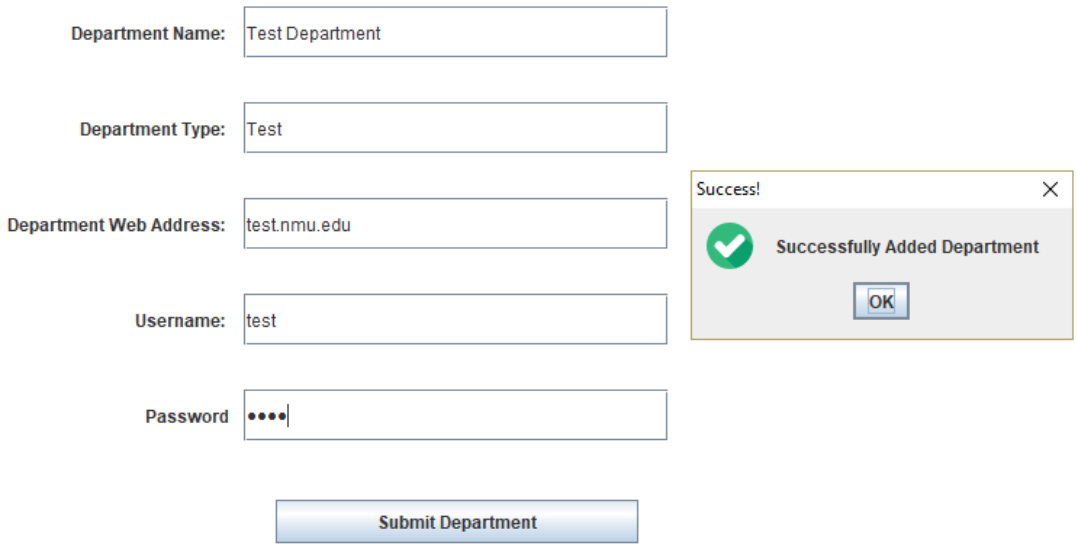


Figure 4. 6 – Administrator Dashboard: Add Department Test Valid

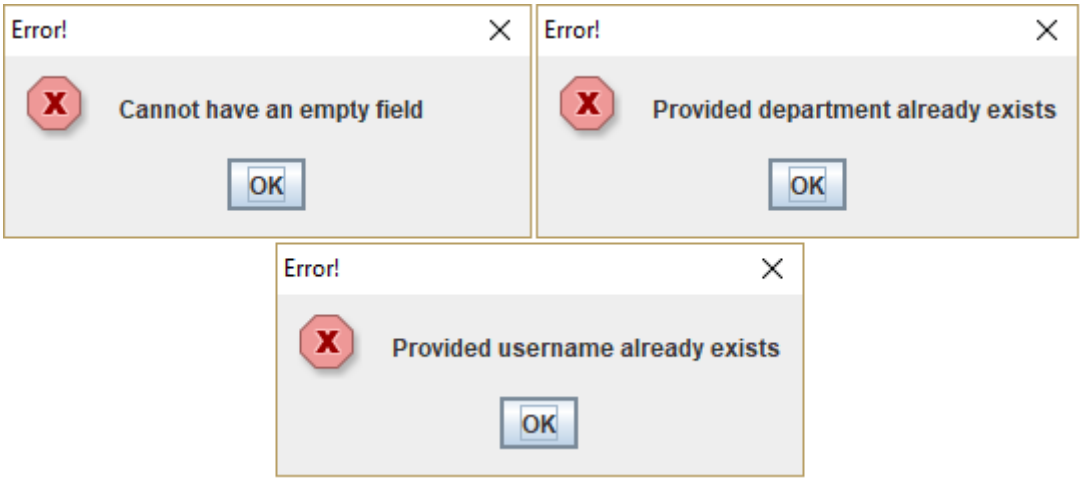


Figure 4. 7 – Administrator Dashboard: Add Department Test Invalid

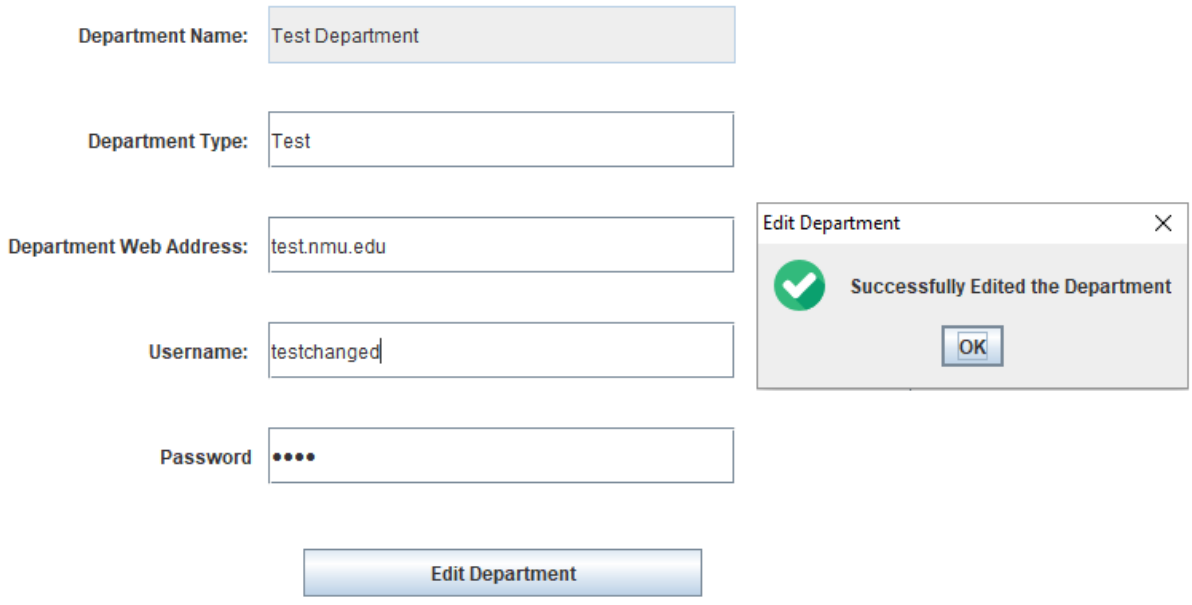


Figure 4. 8 – Administrator Dashboard: Edit Department Test Valid

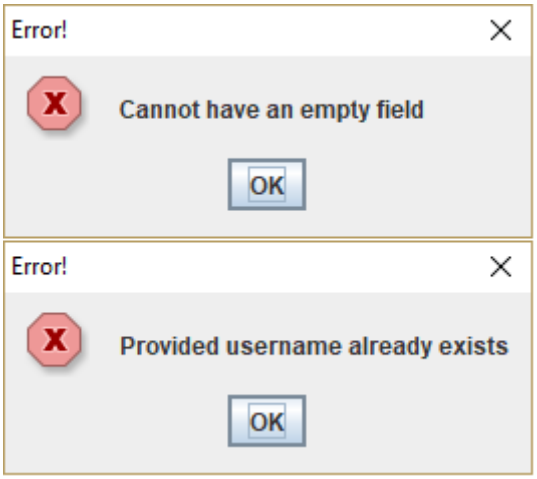


Figure 4. 9 – Administrator Dashboard: Edit Department Test Invalid

Table 4.3 – Administrator Lecturer Tests

	Test Case	Steps to Test	Expected Outcome	Actual Outcome
1.	Administrator goes to the Lecturers area	• Administrator logs in • Administrator clicks the Lecturers navigation button	Three boxes showing the three types of lecturers are shown. The boxes contain heading, some text and a button	Same as expected.
2.	Administrator goes to the Full Time Lecturers area	• Admin goes to the Lecturers area • Admin clicks the Full Time Lecturers button from the first box.	A table showing all the details of all the available Full Time Lecturers is shown	Same as expected
3.	Administrator clicks Add new Full Time Lecturer button	• Administrator goes to the full-time lecturers area • Administrator clicks Add new Full Time Lecturer Button.	An empty form showing all the full-time lecturer fields is shown. The form also contains a select drop down list showing the available departments.	Same as expected
4.	Administrator clicks on a line	• Administrator goes to the full-time lecturer area.	A filled form that contains the data from the selected	Same as expected, except for the

	of lecturer from the table	• Administrator clicks a line of lecturer from the table.	lecturer is shown. The Edit Full Time Lecturer and Delete Full Time Lecturer buttons are shown underneath.	departments dropdown. The selected department is not the same as previous department.
5.	Administrator deletes full-time lecturer	• Administrator goes to the full-time lecturer area. • Administrator clicks a line of lecturers from the table • Administrator clicks the Delete Full Time Lecturer button.	A prompt asking whether the user is sure to delete the lecturer or not is shown. The lecturer is deleted when Yes button is clicked	Same as expected
Add new Lecturer				
6.	Administrator fills up all the details and submits the form.	• Administrator goes to the form to add a new full-time lecturer • Administrator fills up all the details • Administrator clicks Submit Full Time Lecturer button	Administrator is taken to the Full Time Lecturer list where the new record is added	Same as expected
7.	Administrator fills up only partial details and submits the form.	• Administrator goes to the form to add a new full-time lecturer • Administrator fills up partial details • Administrator clicks Submit Full Time Lecturer button	Error message saying fields cannot be empty is shown	Same as expected
8.	Administrator enters text value in salary field	• Administrator goes to the form to add a new full-time lecturer	Error message saying incorrect input is shown	Same as expected

		• Administrator fills up all the details but enters text in salary field instead of number • Administrator clicks Submit Full Time Lecturer button		
Edit Lecturer				
9.	Administrator edits a lecturer correctly	• Administrator clicks a line of lecturer from the table. • Administrator edits some fields. • Administrator clicks Edit Full Time Lecturer button	Success message showing successfully edited lecturer is shown	Same as expected
10.	Administrator erases some field data while editing a lecturer	• Administrator clicks a line of lecturer from the table. • Administrator erases some field data. • Administrator clicks Edit Full Time Lecturer button	Error message saying fields cannot be empty is shown	Same as expected
11.	Department where the selected lecturer belongs is deleted.	• Administrator views a lecturer's department. • Administrator goes to the Departments area and deletes the Department. • Admin returns to the Lecturers area.	• The Lecturer's department is not changed, and the removed department's name is shown until manually changed by editing the lecturer.	Same as expected.

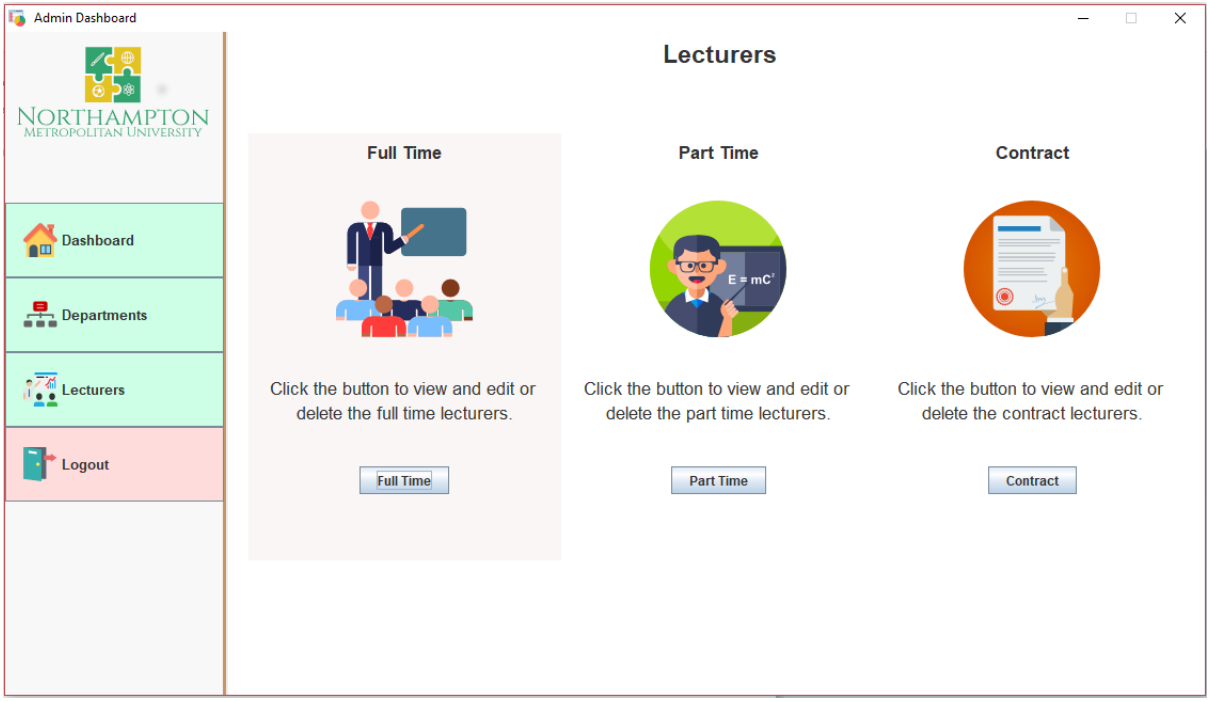


Figure 4. 10 – Administrator Dashboard: Lecturers

ID	Name	Address	Department	Phone	Email Address	Date Started	Salary (£)
1	Nischal Khadka	New Baneshwor	Computer Science	9818181818	nischal@nmu.edu...	2018-08-20	1000.0
2	Deepak Kumar Ka...	Jorpati	Computer Networ...	9818252525	deepak@nmu.ed...	2014-07-08	2000.0
3	Nishant Neupane	Balaju	Computer Science	9825253515	nishant@nmu.ed...	2016-12-08	1000.0
4	Khagendra Sah	Chabahil	Computer Networ...	9808022336	khagendra@nmu....	2016-12-12	1500.0
5	Himalaya Kaksha...	Gongabu	Computer Science	9841424256	himalaya@nmu.e...	2015-07-12	1500.0
6	Sajit Chandra Sha...	Maitidevi	Physics	9853206825	sajitchandra@nm...	2014-08-06	1200.0
7	Balaram Katwal	Anamnagar	Physics	9860252326	balramkatwal@n...	2015-02-26	1200.0
12	Mamata Bhattrai	Kalanki	Computer Science	9843526523	mamata@nmu.ed...	2013-10-10	1500.0

Figure 4. 11 – Administrator Dashboard: Full Time Lecturer List

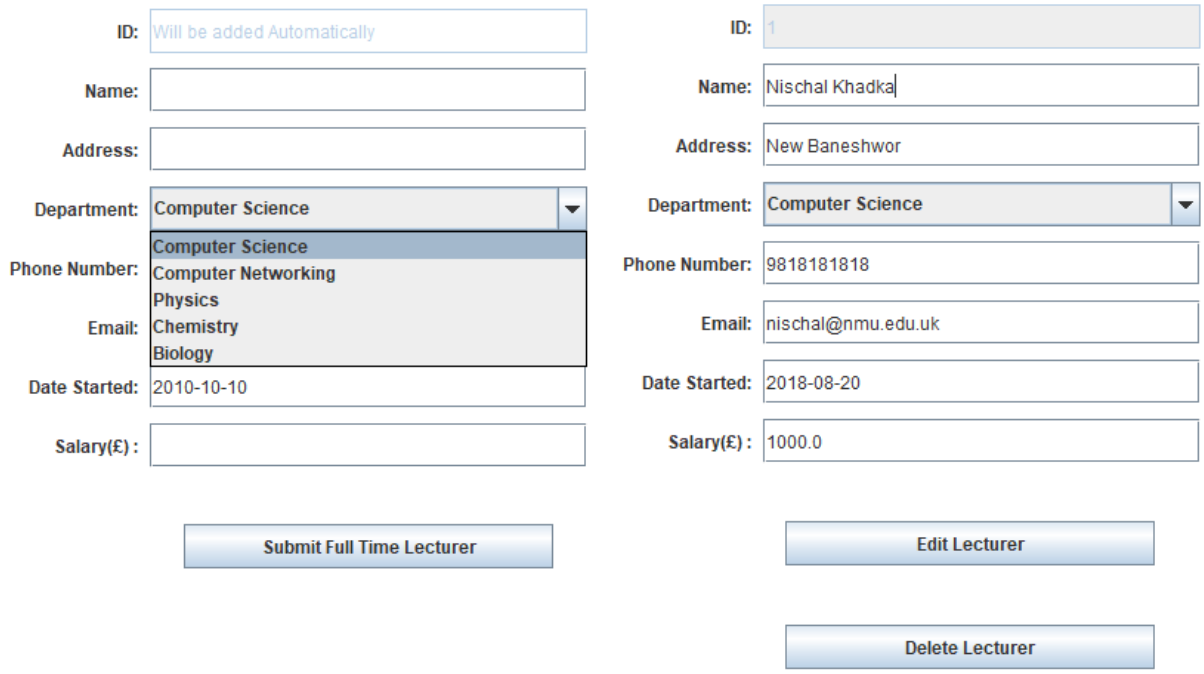


Figure 4. 12 – Administrator Dashboard: Add or Manage Full Time Lecturers

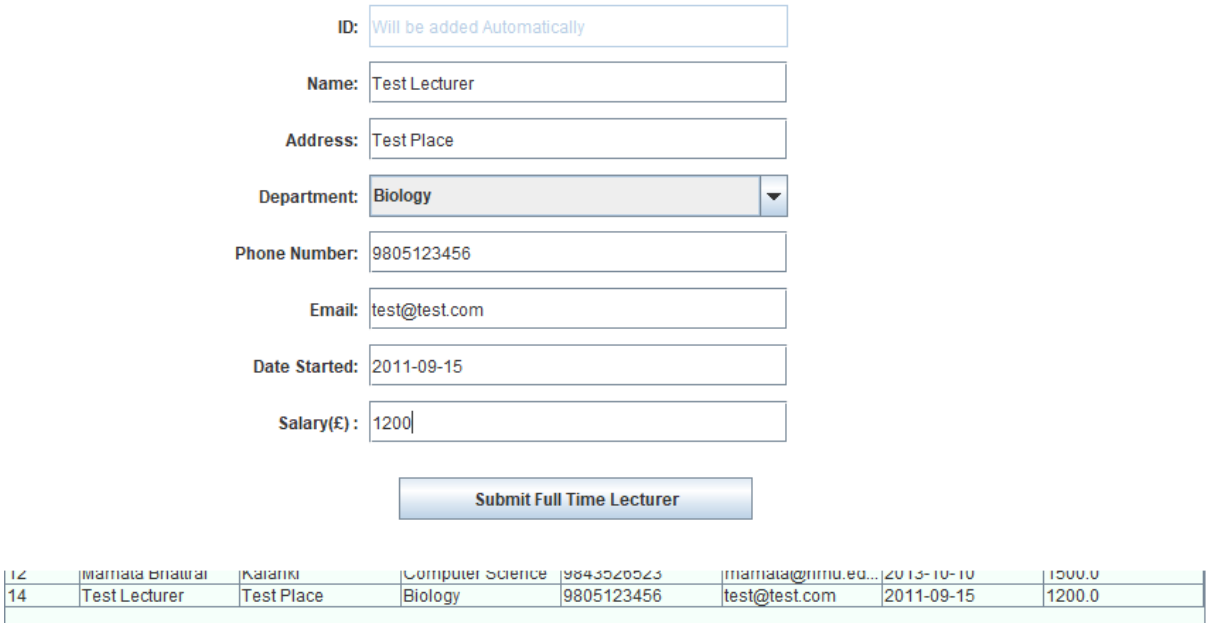


Figure 4. 13 – Administrator Dashboard: Add new Lecturer Valid

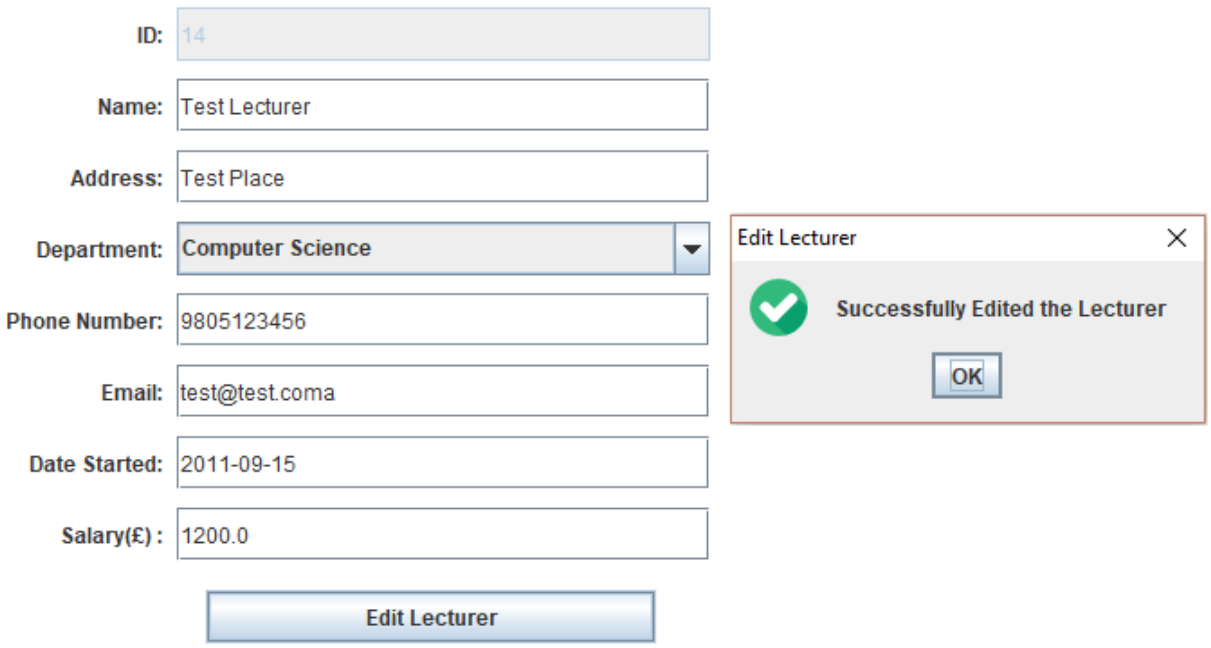


Figure 4. 14 – Administrator Dashboard: Edit Lecturer Valid

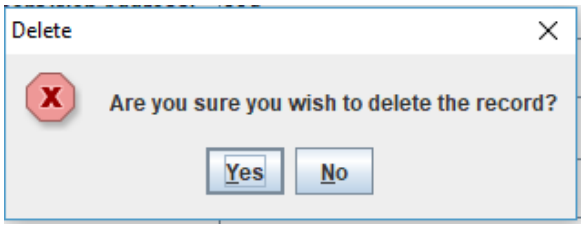


Figure 4. 15 – Administrator Dashboard: Delete Lecturer Prompt

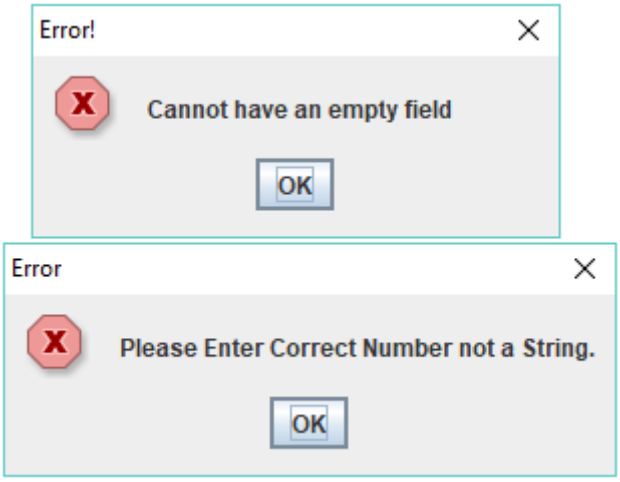


Figure 4. 16 – Administrator Dashboard: Add or Edit Lecturer Invalid

Table 4.4 – Secretary Tests

	Test Case	Steps to Test	Expected Outcome	Actual Outcome
1.	Secretary goes to the Secretary Area	• Secretary logs in	Secretary area is displayed	Same as expected
2.	Secretary clicks one specific type of lecturer	• Secretary goes to the Secretary Area • Secretary clicks Full Time Lecturers button	All the full-time lecturers are displayed if they exist and if not, no lecturers available text is shown	Same as expected
3.	Secretary clicks a lecturer's ID	• Secretary goes to the full-time lecturers area • If there are lecturers available, secretary clicks on a lecturer's ID	All the details of the lecturer are displayed	Same as expected
4.	Secretary views all available lecturers	• Secretary goes to the secretary area • Secretary clicks the view all lecturers link at the bottom.	All the lecturers available in the department are shown in a single panel	Same as expected

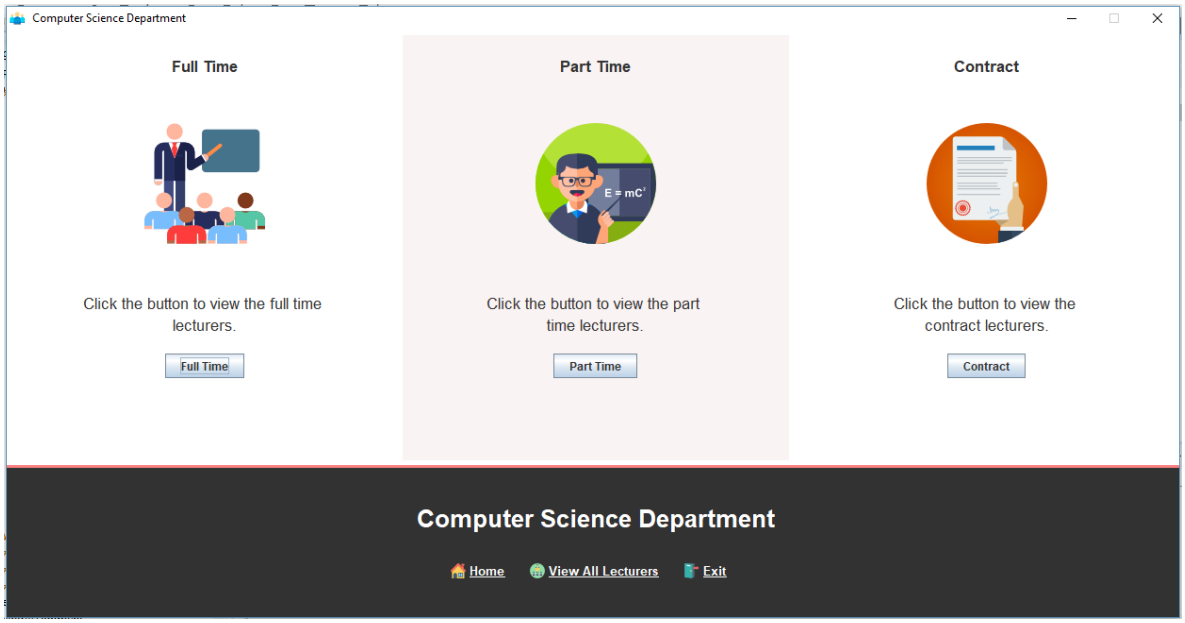


Figure 4. 17 – Secretary Dashboard

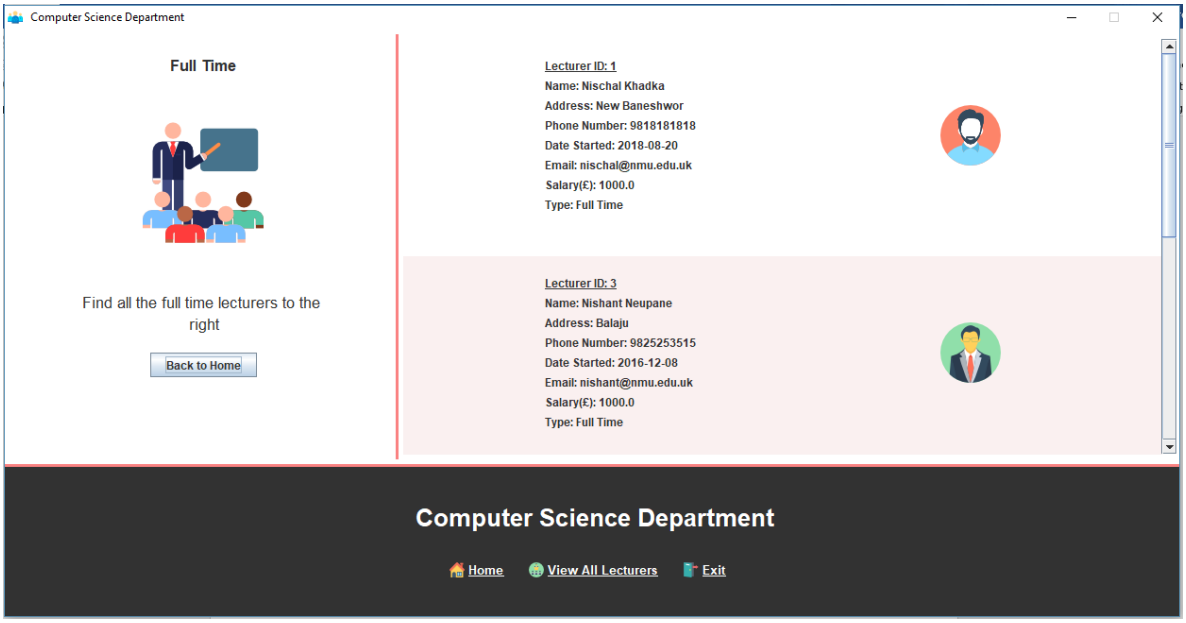


Figure 4. 18 – Secretary Dashboard: Full Time Lecturers

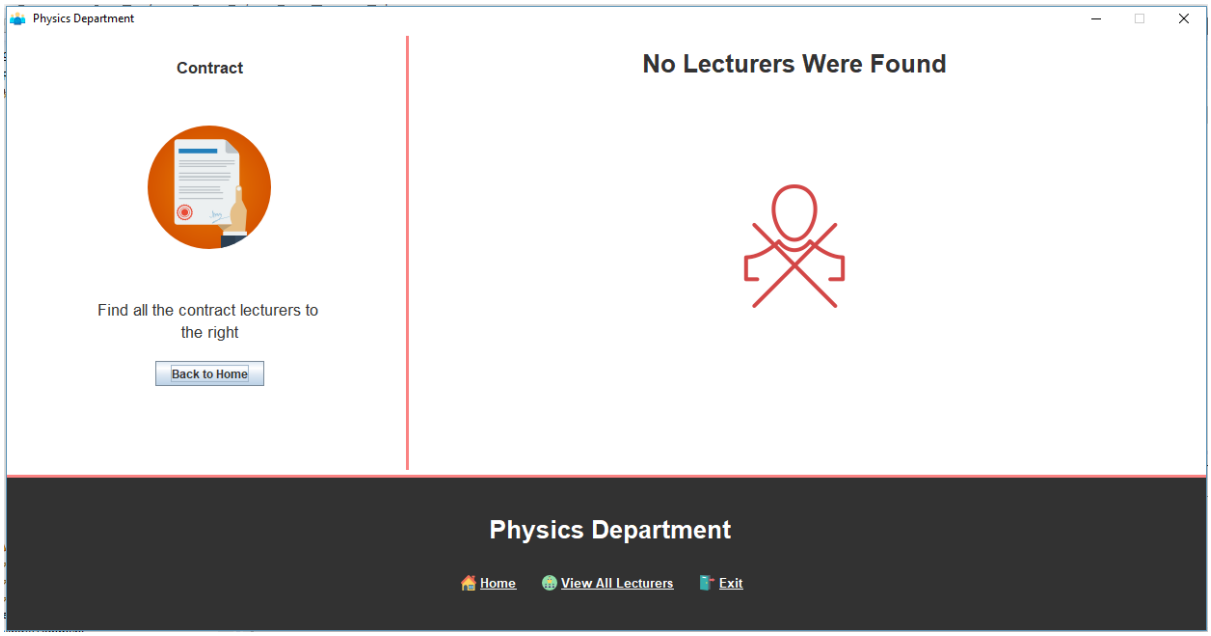


Figure 4. 19 – Secretary Dashboard: No Lecturer Found

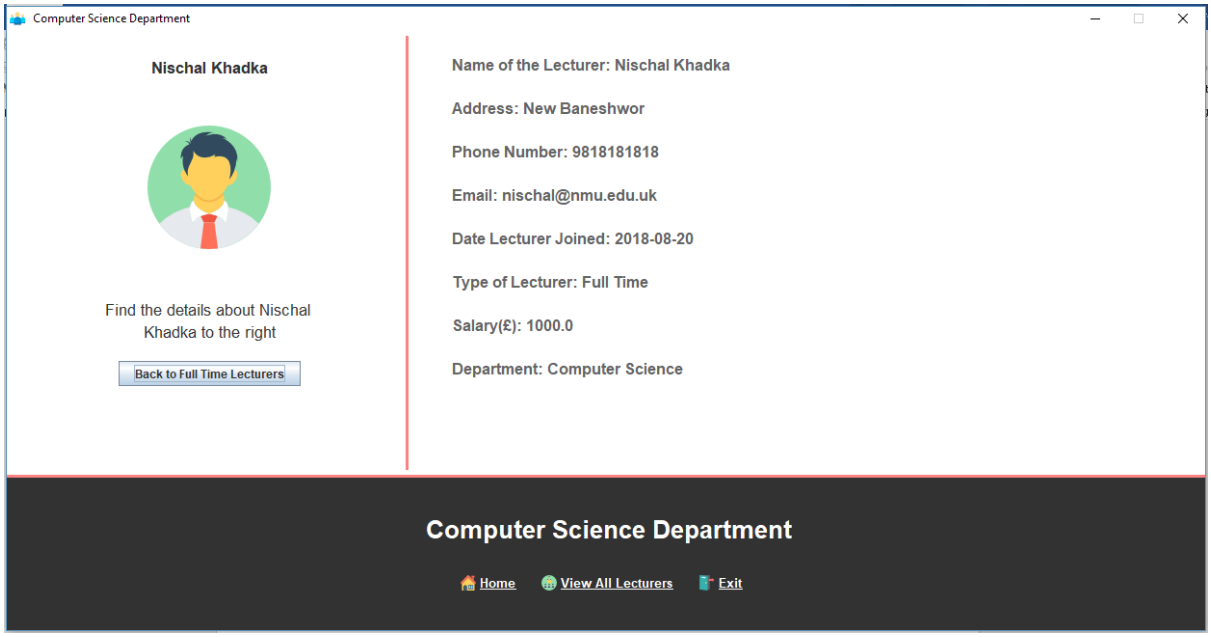


Figure 4. 20 – Secretary Dashboard: Specific Lecturer

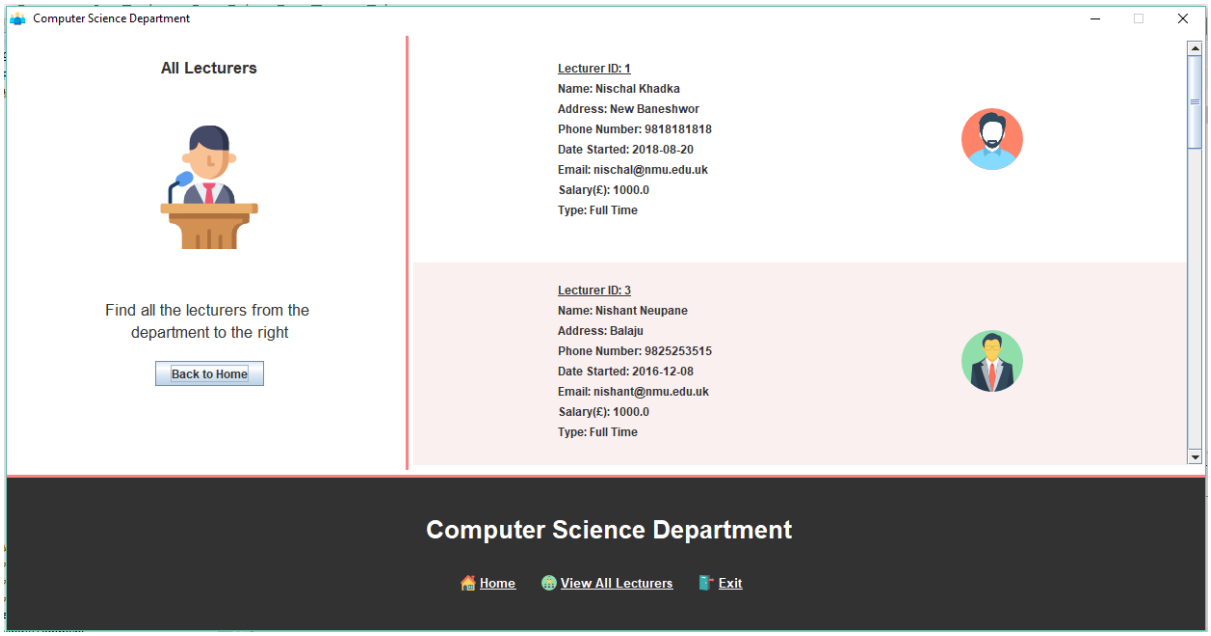


Figure 4. 21 – Secretary Dashboard: View All Lecturers

Black Box Test Report

Bugs Found: 1

Test Case: Administrator clicks on a line of lecturer from the table

Bug: When editing a lecturer, the lecturer's previously set department is not selected from the dropdown. Instead, the first department in the system is selected by default.

Proposed Solution: In the edit lecturers area, set the department of the Lecturer in the dropdown as similarly done to other fields.

4.2 White Box Testing

White box tests are performed on four methods taken from the project. These are chosen with increasing difficulty level.

Basic White Box Test (2 Decision Points)

```
/**
 * Check if provided credentials match
 * @return true if username and passwords match
 */
public boolean checkAccount() {
    departments = getDepartments();
    for(int i = 0; i < departments.size(); i++) {          1
        if(username.equals(departments.get(i).getUsername()) 2
                && password.equals(departments.get(i).getPassword())){
                this.userExists = true;
            }
    }
    return userExists;
}
```

Figure 4. 22 – White Box Test Basic

Cyclomatic Complexity = 1 + ∑ decision points = 1 + 2 = 3

Table 4.1 – Basic Test Paths

Path	Test Case
Decision point 1 false	i=11, departments.size() = 11
Decision point 1 true, decision point 2 false	i = 5, departments.size() = 10, username = “physics”, departments.get(i).getUsername() = “chemistry”
Decision point 1 true, decision point 2 true	i = 5, departments.size() = 10, username = “physics”, departments.get(i).getUsername() = “physics”

Intermediate White Box Test I (4 Decision Points)

```
/**
 * Takes the username as parameter and returns the department it belongs to
 * @param username Username for the department
 * @return the department
 */
public Department getDepartment(String username){
    try {
        //Get all the departments first from the model
        departments = sm.getAllDepartments();
    } catch (ClassNotFoundException e) { 1
        System.out.println(e);
    } catch (IOException e) { 2
        System.out.println(e);
    }

    //Then check the username and return the department if it matches
    for(int i = 0; i < departments.size(); i++) { 3
        if(departments.get(i).getUsername().equals(username)) { 4
            department = departments.get(i);
        }
    }

    return department;
}
```

Figure 4. 23 – White Box Test Intermediate I

Cyclomatic Complexity = 1 + ∑ decision points = 1 + 4 = 5

Table 4.2 – Intermediate I Test Paths

Path	Test Case
Decision point 1 false	Class does not exist
Decision point 1 true, decision point 2 false	IOException occured
Decision point 1 true, decision point 2 true, decision point 3 false	(Assume no exceptions occurred) i=11, departments.size() = 11
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 false	i = 5, departments.size() = 10, username = "physics", departments.get(i).getUsername() = "chemistry"
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 true	i = 5, departments.size() = 10, username = "physics", departments.get(i).getUsername() = "physics"

Intermediate White Box Test II (4 Decision Points)

```
/**
 * The method returns the ID with the highest value
 * This method's return value allows to allocate IDs for new Lecturers
 *
 * @return highest ID
 */
public int getHighest() {
    try {
        lecturers = getAllLecturers();
    } catch (ClassNotFoundException e) { 1
        e.printStackTrace();
    } catch (IOException e) { 2
        e.printStackTrace();
    }
    int highest = 0;
    int id = 0;
    for(int i = 0; i < lecturers.size(); i++) { 3
        id = lecturers.get(i).getId();
        if(id>highest) { 4
            highest = id;
        }
    }
    return highest;
}
```

Figure 4. 24 – White Box Test Intermediate II

Cyclomatic Complexity = 1 + ∑ decision points = 1 + 4 = 5

Table 4.3 – Intermediate II Test Paths

Path	Test Case
Decision point 1 false	ClassNotFoundException occurred
Decision point 1 true, decision point 2 false	IOException occured
Decision point 1 true, decision point 2 true, decision point 3 false	(Assume no exceptions occurred) i=25, lecturers.size() = 25
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 false	i = 15, lecturers.size() = 25, id = 10, highest = 15
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 true	i <= lecturers.size(), id = 20, highest = 15

Advanced White Box Test (6 Decision Points)

```
//Check if the provided Department or username already exists
for(int i = 0; i < departments.size(); i++) { 1
    if(adv.getDepartmentDetails()[0].equals(departments.get(i).getName())) 2
        throw new AlreadyExistsException("Department");
    if(adv.getDepartmentDetails()[3].equals(departments.get(i).getUsername())) 3
        throw new AlreadyExistsException("Username");
}
//Check if the provided Details contain empty fields
for(int i = 0; i < departmentDetails.length; i++) {4
    if(!adv.getDepartmentDetails()[i].equals("") && !adv.getDepartmentDetails()[i].equals(" "))
        departmentDetails[i] = adv.getDepartmentDetails()[i];                                    5
    else
        throw new NullStringException("Null Field");
}

//Add the New Department to the file
try {
    adm.addDepartmentToFile();
} catch (Exception e) { 6
    e.printStackTrace();
}
```

Figure 4. 25 – White Box Test Advanced

Cyclomatic Complexity = 1 + ∑ decision points = 1 + 6 = 7

Table 4.4 – Advanced Test Paths

Path	Test Case
Decision point 1 false	i=11, departments.size() = 11
Decision point 1 true, decision point 2 false	i = 5, departments.size() = 10, adv.getDepartmentDetails()[0] = “Physics”, departments.get(i). getName() = “Chemistry”
Decision point 1 true, decision point 2 true, decision point 3 false	i = 5, departments.size() = 10, adv.getDepartmentDetails()[0] = “Physics”, departments.get(i). getName() = “Physics”, adv.getDepartmentDetails()[3] = “physics”, departments.get(i). getUserame() = “chemistry”

Assume all of above true	
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 false	adv.getDepartmentDetails()[3] = “physics”, departments.get(i). getUserame() = “physics”, i=6, departmentDetails.length = 6
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 true, decision point 5 false	i=3, departmentDetails.length = 6, adv.getDepartmentDetails()[3] = “” or adv.getDepartmentDetails()[3] = “ ”
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 true, decision point 5 true, decision point 6 false	i=3, departmentDetails.length = 6, adv.getDepartmentDetails()[3]= “physics”
Decision point 1 true, decision point 2 true, decision point 3 true, decision point 4 true, decision point 5 true, decision point 6 true	Any kind of exception occured

In the basic method, the complexity count was lower, and it was a method with very little risk. The other methods had higher complexity count and the path was longer. As such, it would be difficult to perform tests on longer methods with higher complexity count. From the data, the basic and intermediate path and test cases are easier to read whereas it is slightly difficult to extract information from the advanced test by a second reader.

5. EVALUATION

5.1 Bugs or Weaknesses and Proposed Solutions

Table 5.1 – Bugs/Weaknesses and Proposed Solutions

Bug/Weakness	Proposed Solution
Bug found from Black Box test - When editing a lecturer, the lecturer's previously set department is not selected from the dropdown. Instead, the first department in the system is selected by default.	In the edit lecturers area, set the department of the Lecturer in the dropdown as similarly done to other fields.
The administrator's username and password are not set and saved in a different file unlike the secretaries. As such, the administrator username and passwords cannot be changed by a person without knowledge of Java.	The administrator username and password could be saved to a separate file, also allowing them to change the credentials if they would like to do so.
When a department, under which a lecturer is assigned, is deleted, the information is not updated in the lecturer's details. The same deleted department's name is displayed until edited manually by the user.	It could be made so that removing the departments would set the department of the lecturers under it to a blank department. For example: A department with name "No Department"
When the administrator is adding or editing a particular lecturer, there is no way to go back to the previous page. Instead, they should navigate to the Lecturers dashboard and find their way back to the previous page.	A back button could be introduced in every possible area which would allow efficient transition between system states.

5.2 Strengths of the Solution

- Completed all the basic as well as additional requirements from the brief.
- Appealing GUI with modern design
- Focused on usability – Any kind of user with basic computer knowledge will be able to use the system to its full potential
- Proper form validation where users are not allowed to leave empty fields
- Properly assigned lecturer ids where they are set automatically, do not conflict with each other and does not concern the users, i.e. they do not have to remember ID of a lecturer.
- Prompts before deleting data, as the users could accidentally click the wrong buttons. This prevents sensitive data from being lost by accident.

6. CONCLUSIONS AND RECOMMENDATIONS

In a nutshell, completing the assignment gives a clear understanding of the MVC pattern. The Java Swing library allows proper separation of Model, View and Controller contents to different java files and they can easily be linked to one another simply by creating the instances. It would not be tough to move to a different library or programming language as a whole using the knowledge and skills gained with this assignment. Exception handling, and file handling, which too work in a similar manner across different programming languages, could be handled with much ease. Although one could complain that the Java Swing library is old and not to be used in this time, the Swing clearly provides the basics of what is required to create an application, allowing to transition the skills to modern approaches.

Although the completed application has met all the required criteria, it still cannot be said as a perfect solution. Some weaknesses still exist in the program. If the application was to be remade from scratch, the addressed bugs and weaknesses would be solved. Also, a cleaner approach to the MVC pattern would be made.

Extra features that could have been added if more time was available:

- The system could be changed to a larger application which would store not only lecturers, but all the different types of staff associated with the university.
- The system could be extended to manage payrolls, tax and leaves of staff
- The data could be stored in a much secure database, which would also prevent unauthorized access and deletion of files.
- The information stored in the system could be moved to a separate trash area before deleting them permanently from the system. This would prevent data loss from accidental deletion.
- Notification system could be added which would allow the administrator to send email notifications to all the concerning party at once.
- A notice board could be added which would allow notices to be published under each department. From their login, the secretaries would be able to view the notice board for their department.

APPENDIX – REFERENCES

Java Swing MVC Pattern

Ashraf Sarhan (2016), Java Swing MVC Example. Available from: https://examples.javacodegeeks.com/core-java/java-swing-mvc-example/ [Accessed 14/02/2019]

Java ArrayList

W3Schools (2019), Java ArrayList. Available from: https://www.w3schools.com/java/java_arraylist.asp [Accessed 14/02/2019]

JButton Hover Effect

Salah – Stackoverflow (2014), How to put Hover effect on jbutton? Available from: https://stackoverflow.com/questions/22638926/how-to-put-hover-effect-on-jbutton [Accessed 20/02/2019]

JPanel Hover Effect

Resueman – Stackoverflow (2016), How can I highlight JPanels on mouse hover? Available from: https://stackoverflow.com/questions/40769190/how-can-i-highlight-jpanels-on-mouse-hover [Accessed 21/02/2019]

Using Listener on JTable

Comscigate (2019), Using the TableModleListener. Available from: http://www.comscigate.com/tutorial/KjellStyle/AjaySabhaney/Chapter%201/Page_11.html [Accessed 22/02/2019]

Set Default Date Value in JTextField

Joval – Stackoverflow (2012), how to make a jtextfield having a fixed date format? Available from: https://stackoverflow.com/questions/12655811/how-to-make-a-jtextfield-having-a-fixed-date-format [Accessed 25/02/2019]

Class Diagram Shortcuts

Gerd Wagner – Stackoverflow (2015), Shortcut for denoting or implying getters and setters in UML class diagrams. Available from: https://stackoverflow.com/questions/28139621/shortcut-for-denoting-or-implying-getters-and-setters-in-uml-class-diagrams [Accessed 01/03/2019]

Classroom Icon

Flaticon (2019), Classroom free icon. Available from: https://www.flaticon.com/free-icon/classroom_906175#term=teacher&page=1&position=4 [Accessed 15/02/2019]

Teacher Icon

Flaticon (2019), Teacher free icon. Available from: https://www.flaticon.com/free-icon/teacher_1089129#term=teacher&page=1&position=14 [Accessed 15/02/2019]

Analytics App Icon

Flaticon (2019), Analytics free icon. Available from: https://www.flaticon.com/free-icon/analytics_1094833 [Accessed 19/02/2019]

Man Avatar Icon

Flaticon (2019), Man free icon. Available from: https://www.flaticon.com/free-icon/man_236831#term=person&page=1&position=67 [Accessed 17/02/2019]

Man Avatar Icon II

Flaticon (2019), Teacher free icon. Available from: https://www.flaticon.com/free-icon/teacher_234694#term=teacher&page=1&position=35 [Accessed 17/02/2019]