

TSE2101 Software Engineering Fundamentals Group Project

Postgraduate Management System Final Report

Prepared by:

Group Name: Jets

Group Leader: Khew Kit Hahn (1151101813)

Programming Leader: Alfred Loo Wee How (1151100653)

Documentation Manager: Gan Teck Beng (1151102173)

Quality Manager: Foo Chuan Kang (1151100765)

Table of Contents

Project Planner	3
Functional Requirement.....	4
Non-Functional Requirement.....	6
Use-Case Diagram	7
Class Diagram.....	8
Entity Relationship Diagram (ERD).....	9
Flowchart	10
Testing Strategies.....	37
Data Testing.....	38
Test Cases	47
Maintenance Plan.....	53

Project Planner

Project Title	POST-GRADUATE MANAGEMENT SYSTEM
Group Members	<ol style="list-style-type: none"> 1. KHEW KIT HAHN (1151101813) [PROJECT LEADER] 2. ALFRED LOO WEE HOW (1151100653) [PROGRAMMING LEADER] 3. GAN TECK BENG (1151102173) [DOCUMENTATION MANAGER] 4. FOO CHUAN KANG (1151100756) [QUALITY MANAGER]
Problem Statement	Post-graduate management system has developed rapidly in past years. However, it faced some problems, such as bad interface and data redundancy, which result in inefficiency of the system. Thus, problems should be addressed and improvements should be implemented.
Objectives	<ol style="list-style-type: none"> 1. To provide a user-friendly interface management system dedicated to postgraduate student 2. To generate better alerts and notification 3. To improvise the check in and check out system
Specific topic area *	TRANSACTION/ PROCESS MANAGEMENT
Skill set required	<ol style="list-style-type: none"> 1. JAVA 2. DB2
Project schedule	<ol style="list-style-type: none"> 1. First prototype of project should be present on Week 10 2. Final working and prototype must be present on Week 14 <p>***Refer to project timeline for more detail schedule</p>
Supervisor name	DR. CHUA FANG FANG

Functional Requirement

Requirements Elicitations

1. Interview with Stakeholders

- We have visited the IPS office in MMU and interviewed the staff about the jobs and responsibilities done by administrators in their existing postgraduate system. In addition, we have also interviewed a postgraduate student and got a wealthy amount of information about what postgraduate students need in general and stated what they think they would like to see in future updates.

2. Site review

- We have done review on existing system and we figure out that the basic requirement to create a postgraduate management system. We realized there is still a lot of weaknesses on the existing postgraduate management system such as functionality that does not work, information that is not accurate and redundant.

3. Peer to peer review

- In the peer to peer review, we managed to obtain even more unexpected ways to improve our system such as a new way to provide feedback about the lecturers and a resume template which did not come across our minds at all. We knew we would need this and will implement them in our system accordingly.

Main Module: Transaction/ Process Management

1. Notification

I. Fees

- Pop up notifications will be generated as the student fees is release for postgraduate student.
- A student must have at least 1 payment fees.

II. Subject registration

- Pop up notifications will be generated when registration for courses are available for postgraduate students.

III. Result

- Pop up notifications will be generated upon the release of the final exam result for the postgraduate students.
- A student must have 1 final exam result.

IV. Exam timetable

- Pop up notifications will be generated when the final exam timetable is released for the postgraduate students.
- A student only can have one exam timetable.
- An exam timetable can be view by many students.

V. Maintenance

- Maintenance notifications will be sent to the postgraduate student emails whenever a maintenance is occurring.

2. Payment

- I. Multiple Choice of Payment Channel
 - Postgraduate students will be able to choose which payment channel they want since we support most of the payment channel.

3. Report Generation

- I. Accurate Data
 - Data of postgraduate student upon any transaction is correct and accurate.
- II. Progress Report
 - Allow postgraduate student and lecturer to view the what to do next, done or in progress.
- III. Exam Result generation
 - Postgraduate students can view their exam results generated by the admin.
- IV. Exam Slip generation
 - Postgraduate students can view their exam slip containing the timetable generated by the admin.

Module 2: Profile Management

1. Login/ Logout

- Ensure that an account can login and logout for postgraduate students and admins to access the system.

2. View and Edit profile information

- Admins and postgraduate students can edit certain particulars such as e-mail, phone number and Facebook profile.

Module 3: Item Management

1. Add, remove and edit course and subject

- Postgraduate students can add and remove subjects from their course.
- Admins can add, remove and edit courses and subjects in the course.
- A student must take at least 1 subject and maximum 4 subjects.
- Many subjects can have many students.
- A course can have many subjects.
- A subject only can have 1 course.

2. Feedback to lecturer

- Postgraduate students can provide feedback about their lecturer in a form that they can submit to the admins.
- Lecturers can see the students' feedback but the writer remains anonymous.

Non-Functional Requirement

1. Serviceability

Condition: When providing feedback about lecturer to the admin, admins will need to approve the feedback before the lecturers or the management could see them.

Reason: There is a possibility whereby students will provide spam or irrelevant comments that are incomprehensible or offensive.

Impact on the system: Feedback will take a longer time for the lecturer or the management to see it.

Implementation: Admin will have approved and disapprove buttons to allow admins to make such choices.

2. Reliability

Condition: Pop up notification will be generated as soon as the postgraduate student enter the system.

Reason: To inform postgraduate student availability of exam slip, result slip, and enroll in certain subject for that semester.

Impact on the system: Postgraduate student will be able to get the latest info on the system

Implementation: When the system detects the user is a student, the notification will be generated.

3. Security

Condition: Password should never be viewable at the point of entry.

Reason: To prevent other people from peeping user's password

Impact on the system: User will be able to protect their personal information more securely

Implementation: When the student typing the password, system will perform password masking

4. Confidentiality

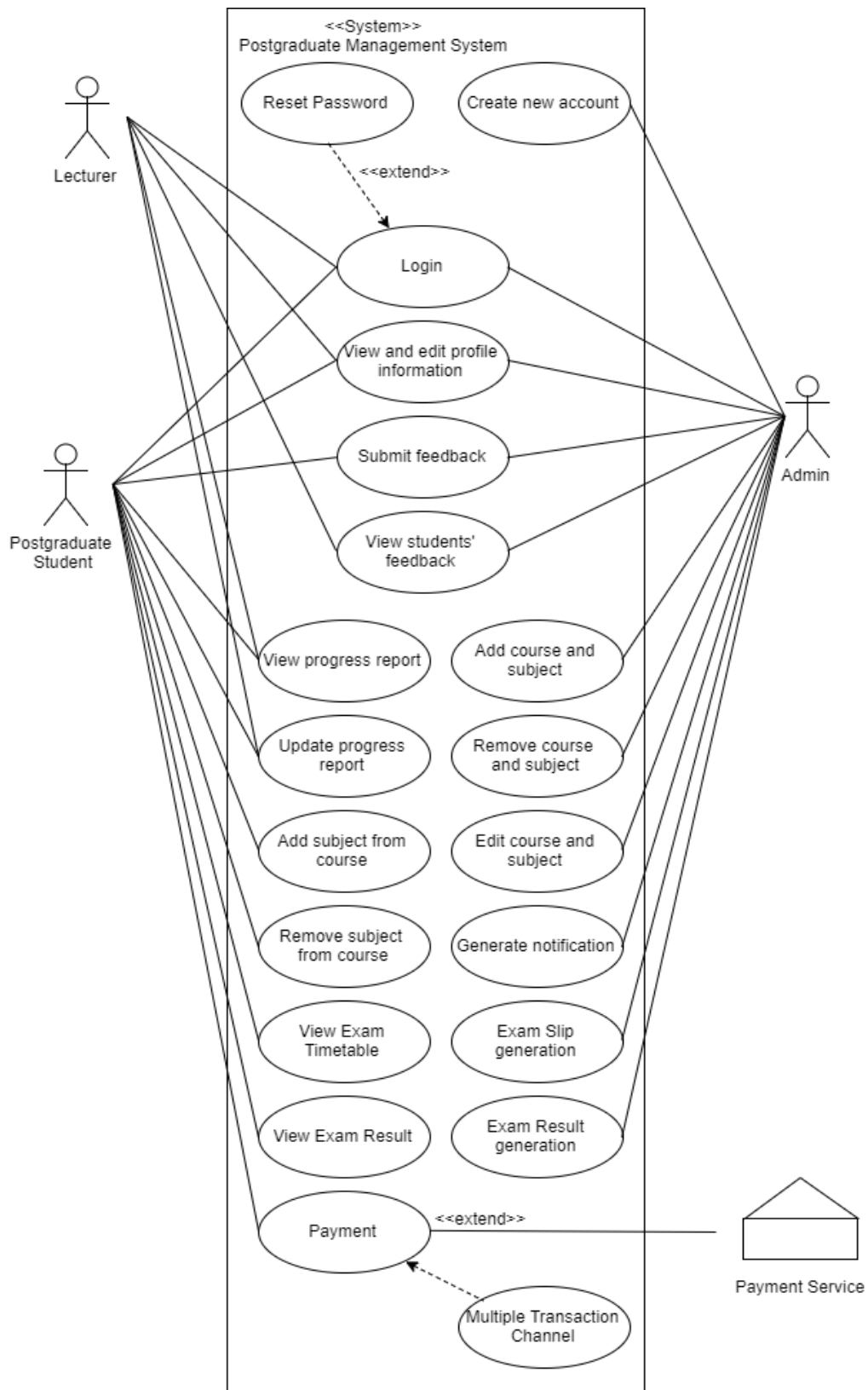
Condition: The postgraduate student should not be able to edit some important personal information, such as their grades and course taken.

Reason: To prevent the student from cheating their information and ensure the confidentiality of information.

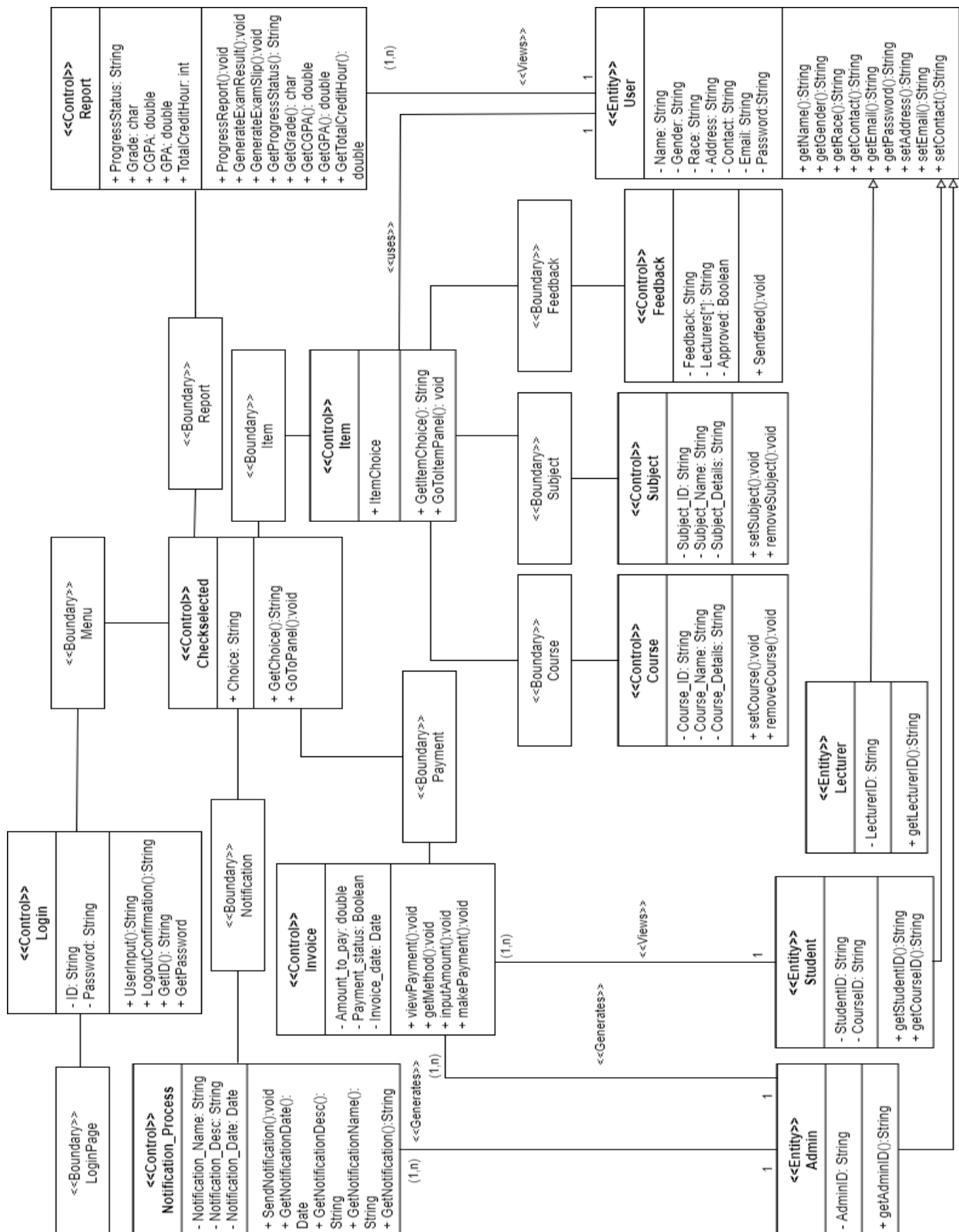
Impact on the system: It will be more secure and prevent valuable information from being manipulated.

Implementation: Some information will be for view only, and not editable

Use-Case Diagram

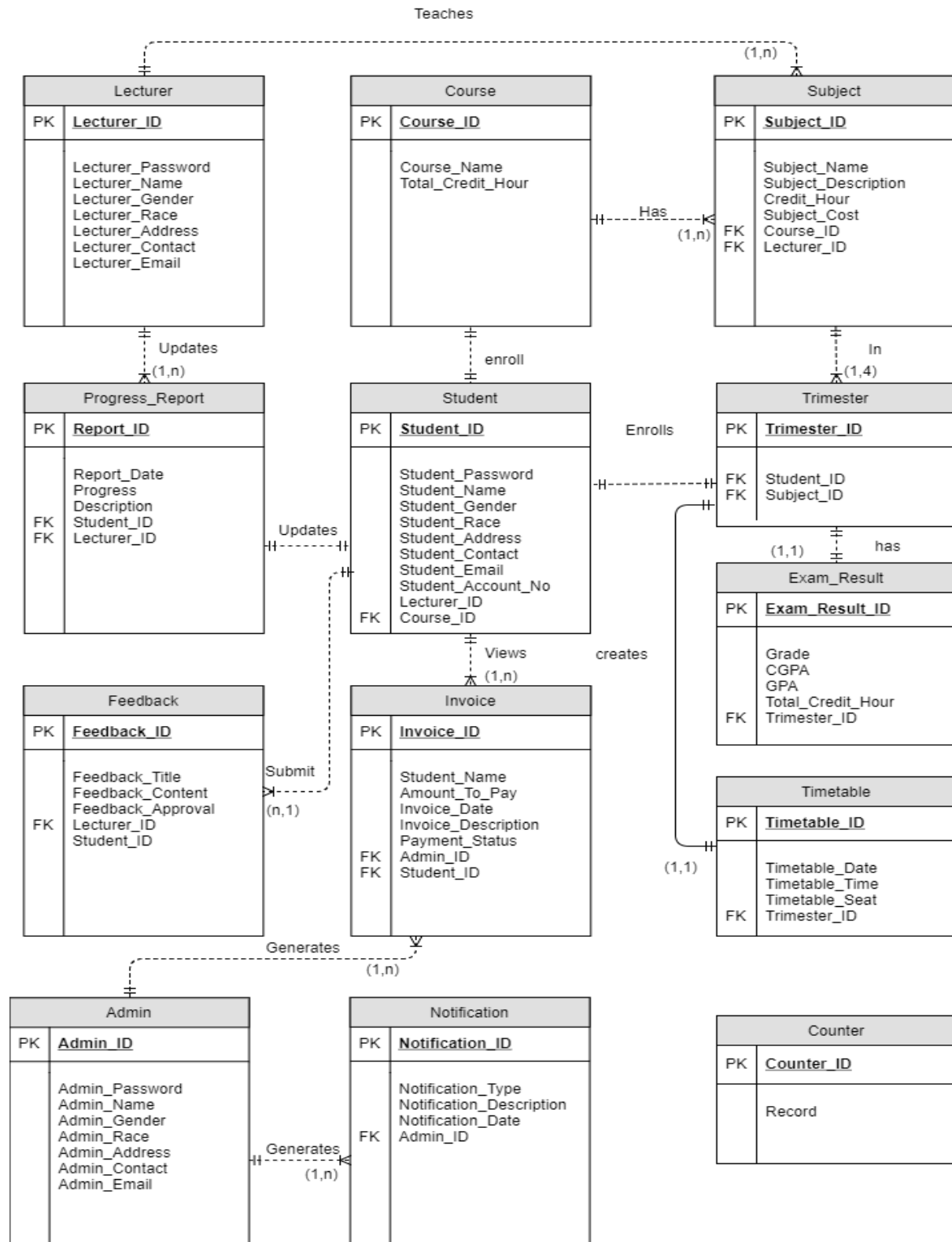


Class Diagram



Entity Relationship Diagram (ERD)

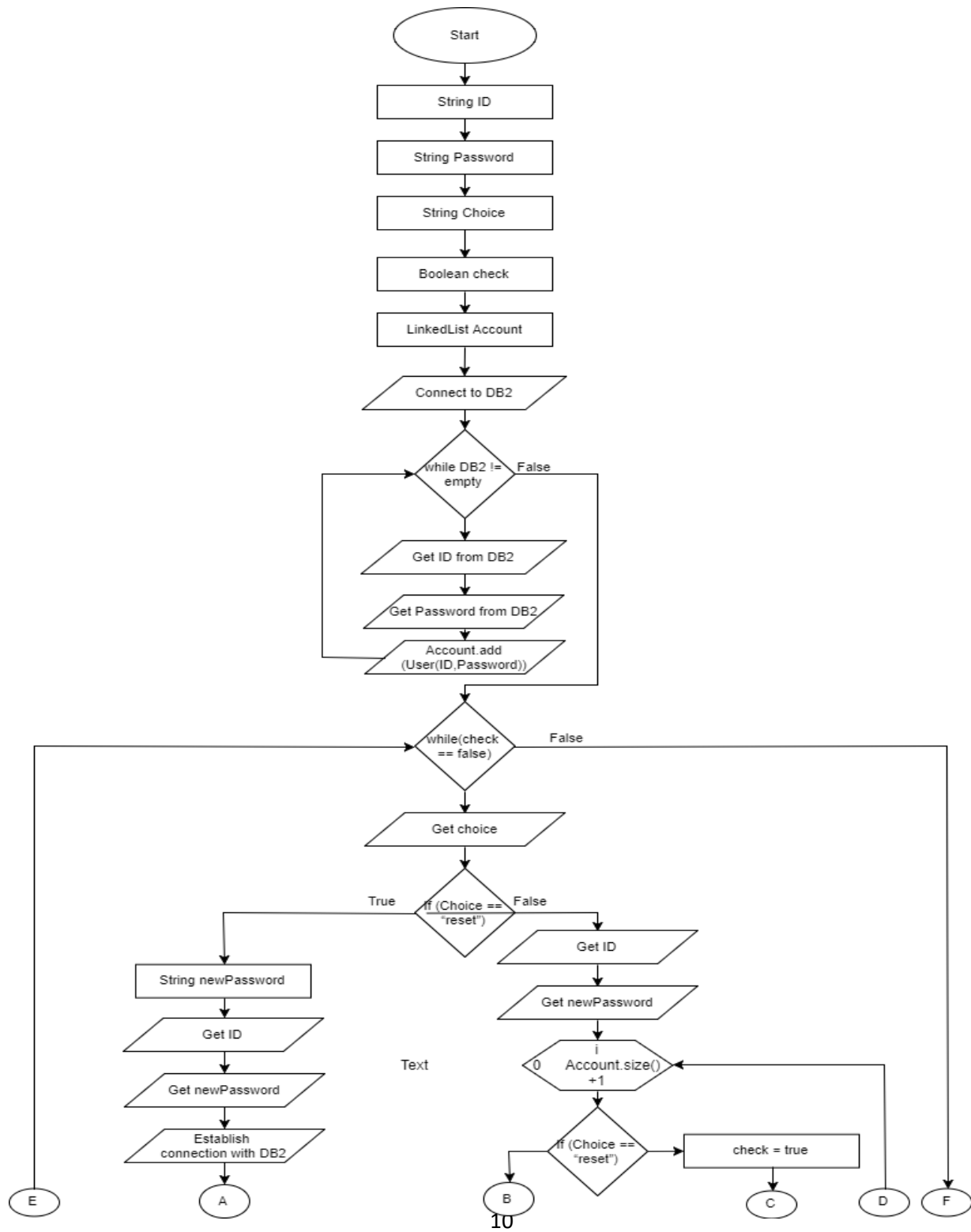
ER Diagram

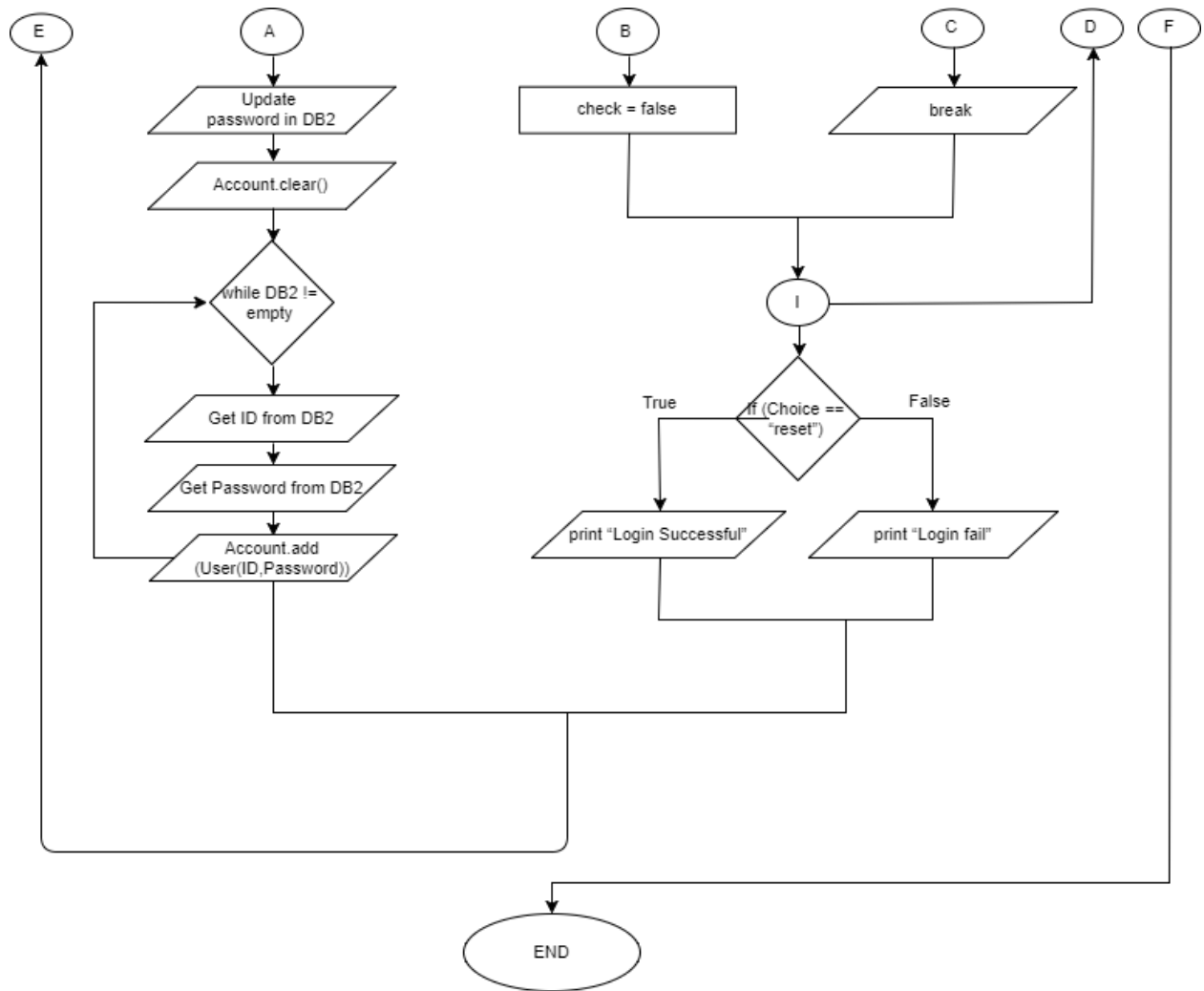


Flowchart

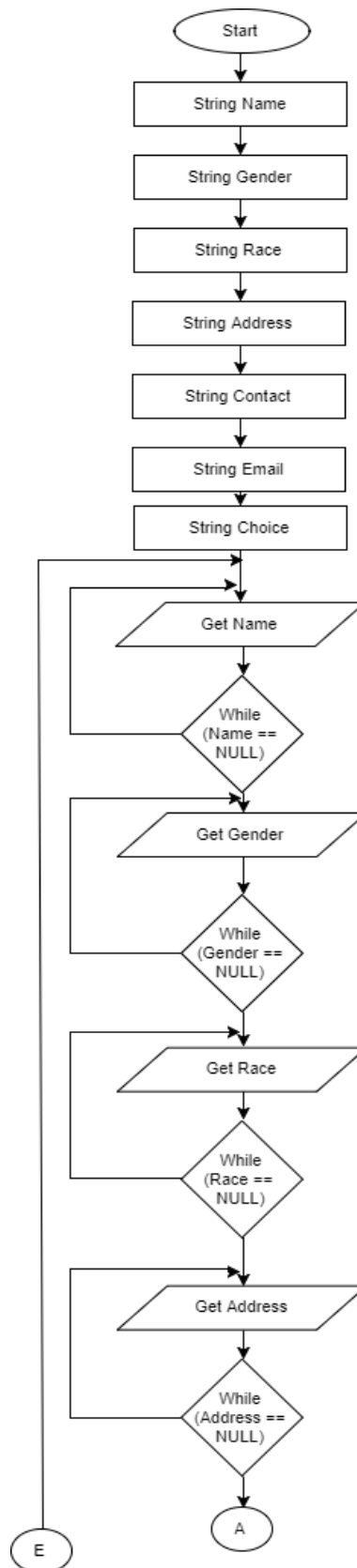
1. Profile Management

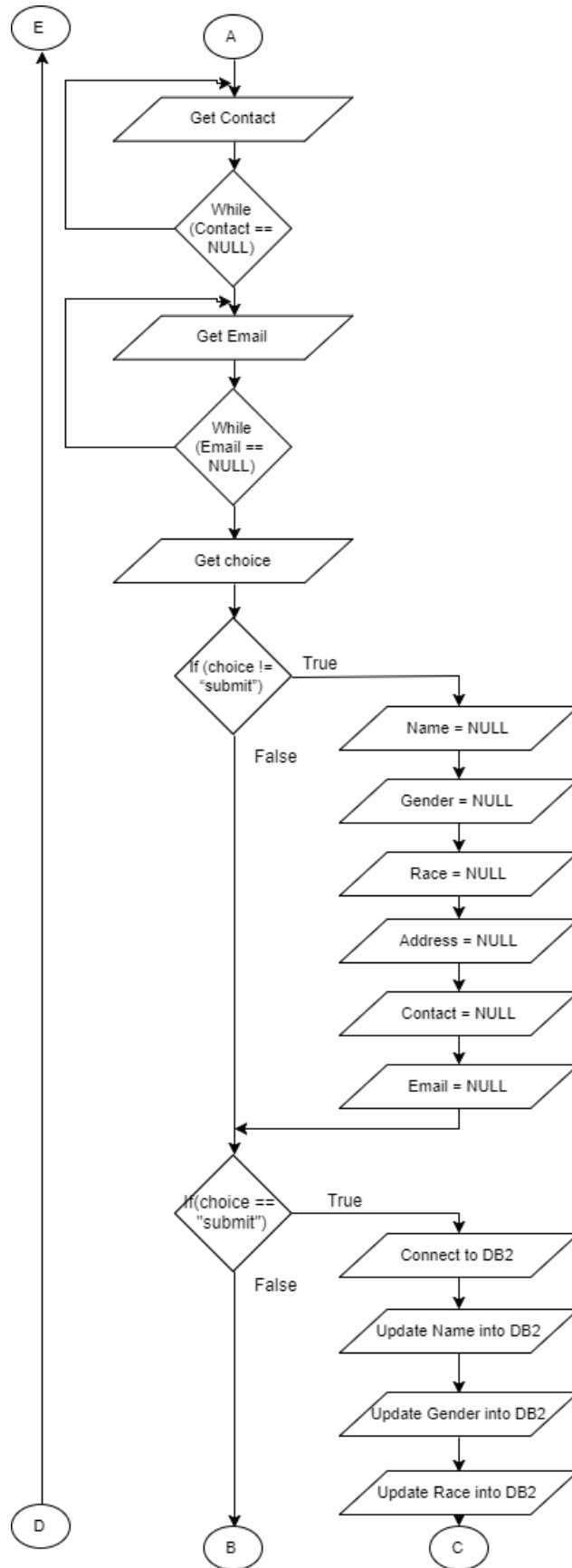
- Login function

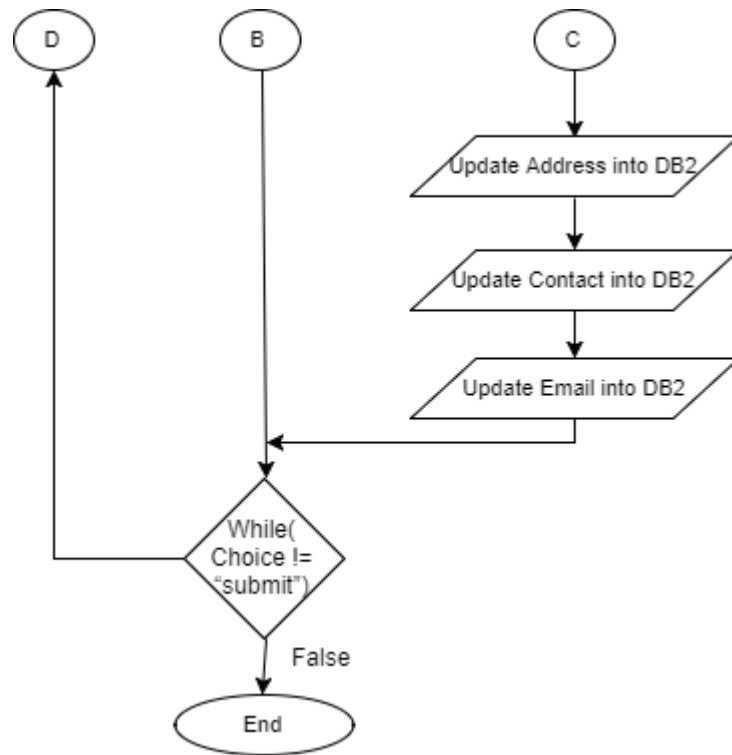




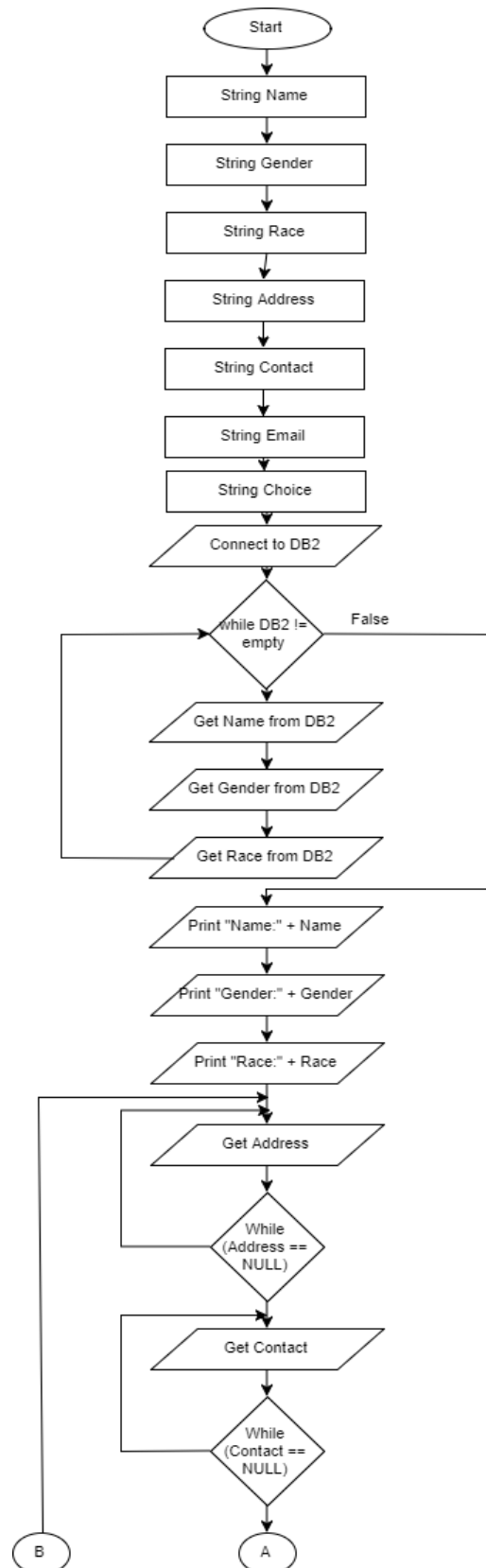
- Add Profile Function

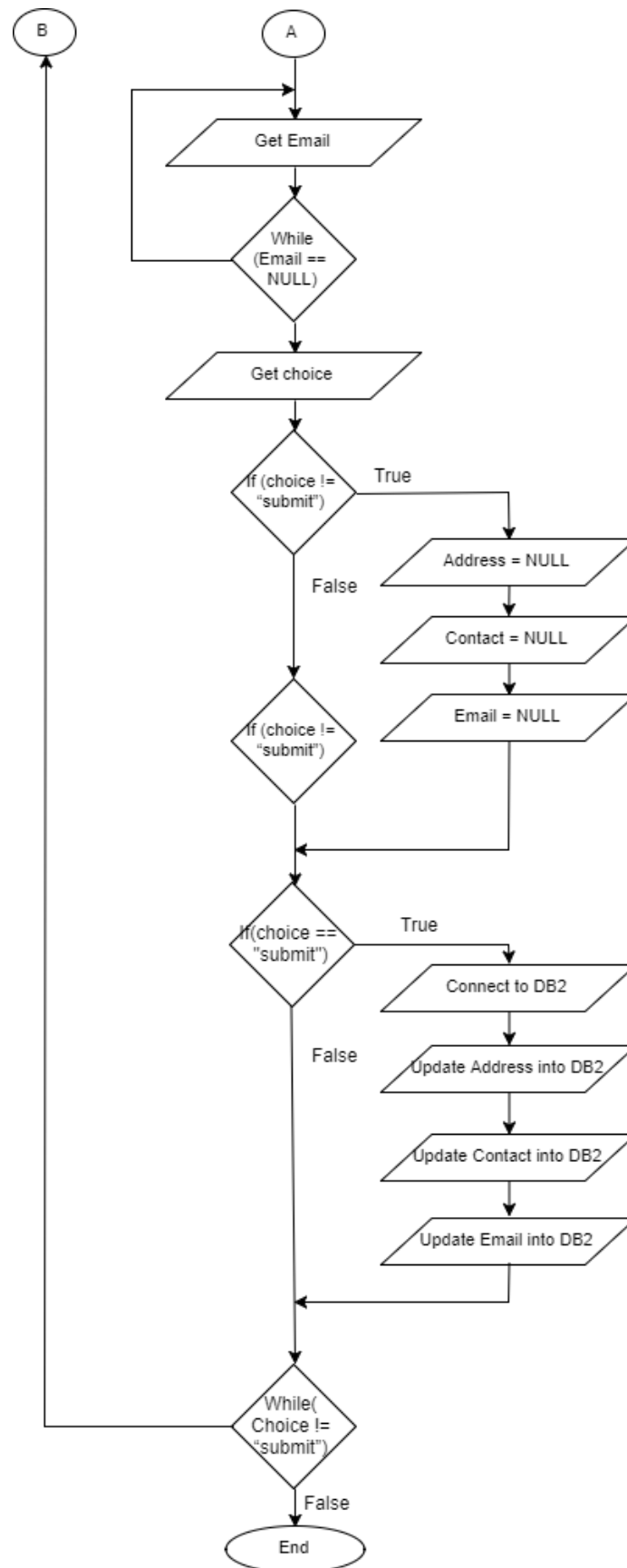




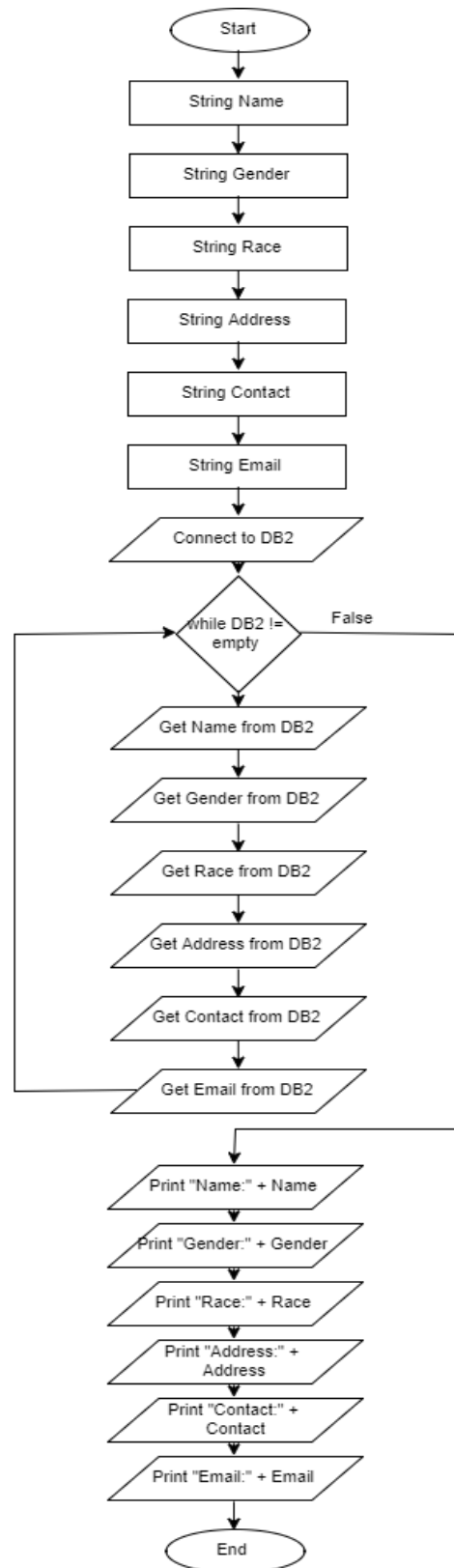


- Edit Profile Function



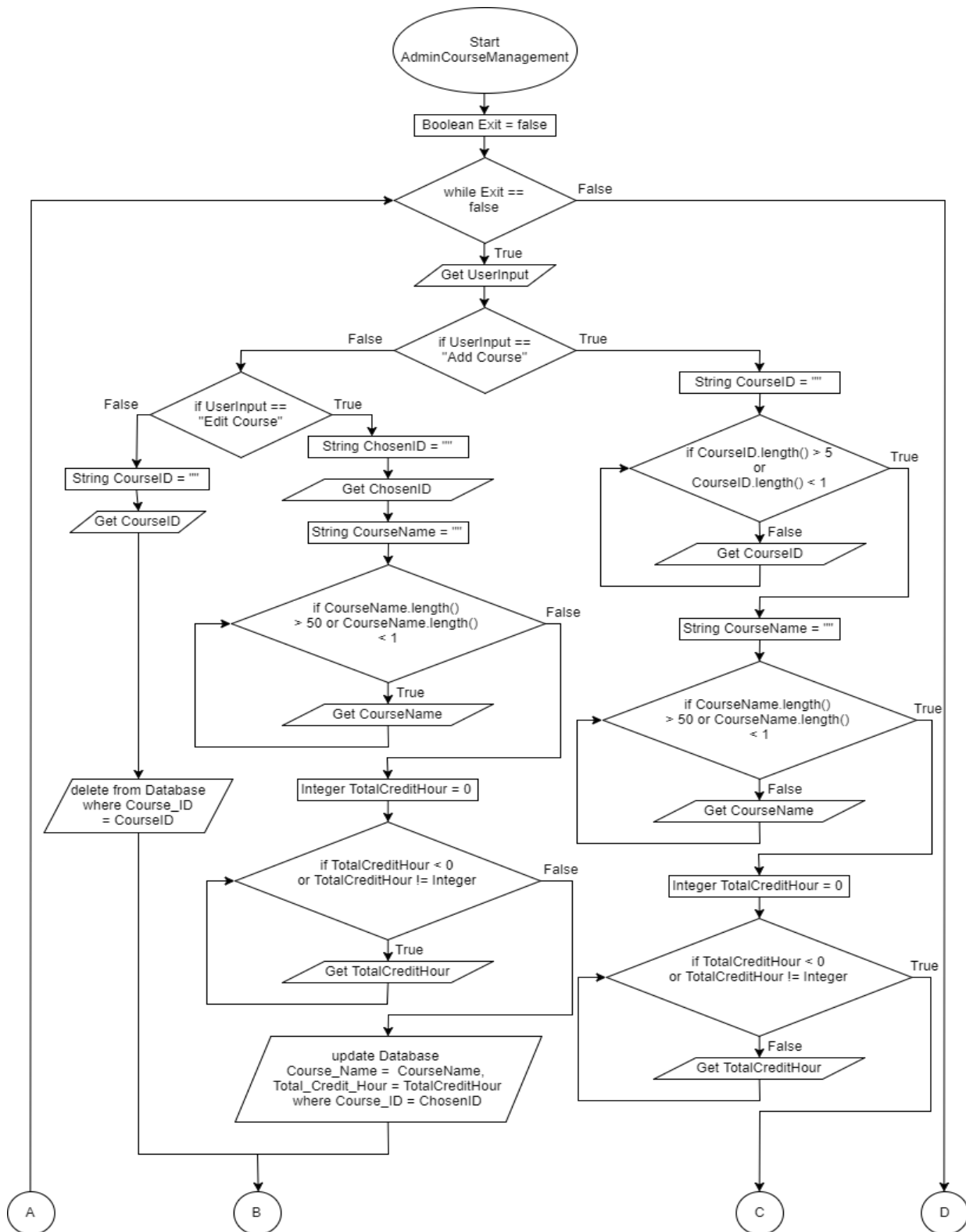


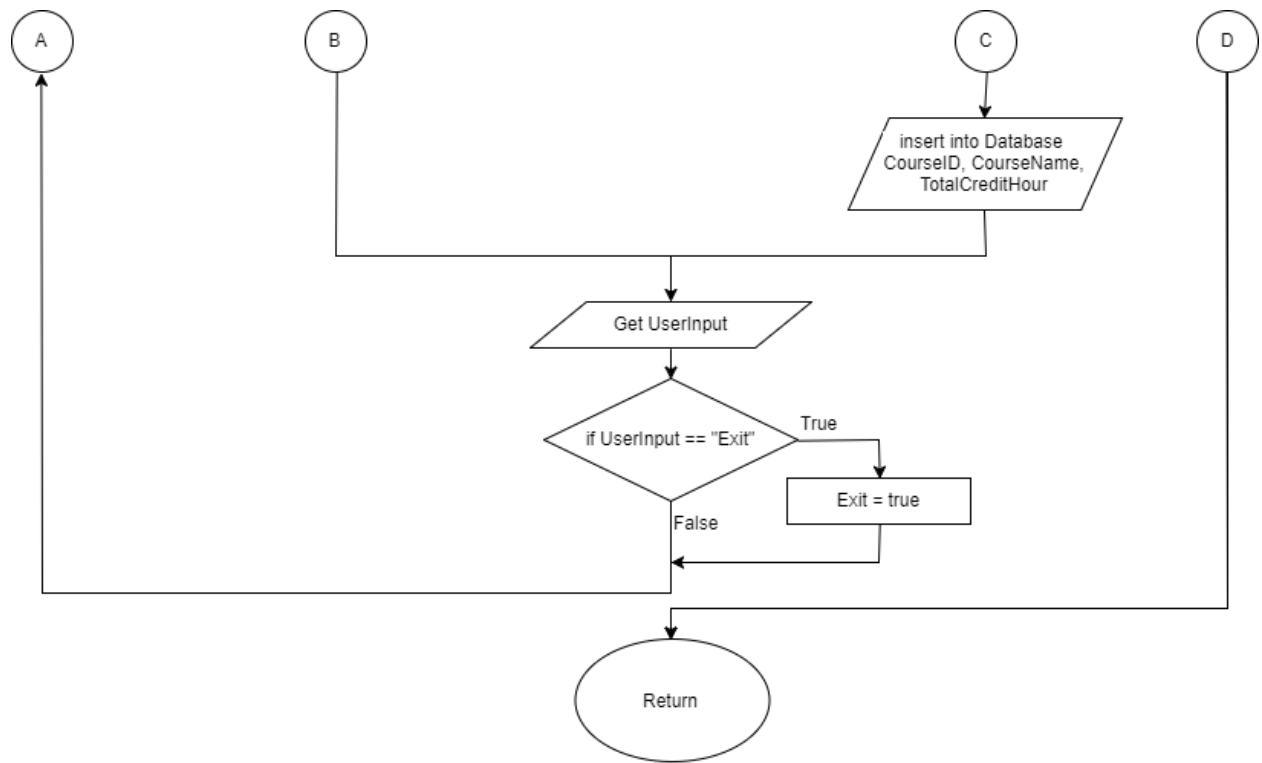
- View Profile Function



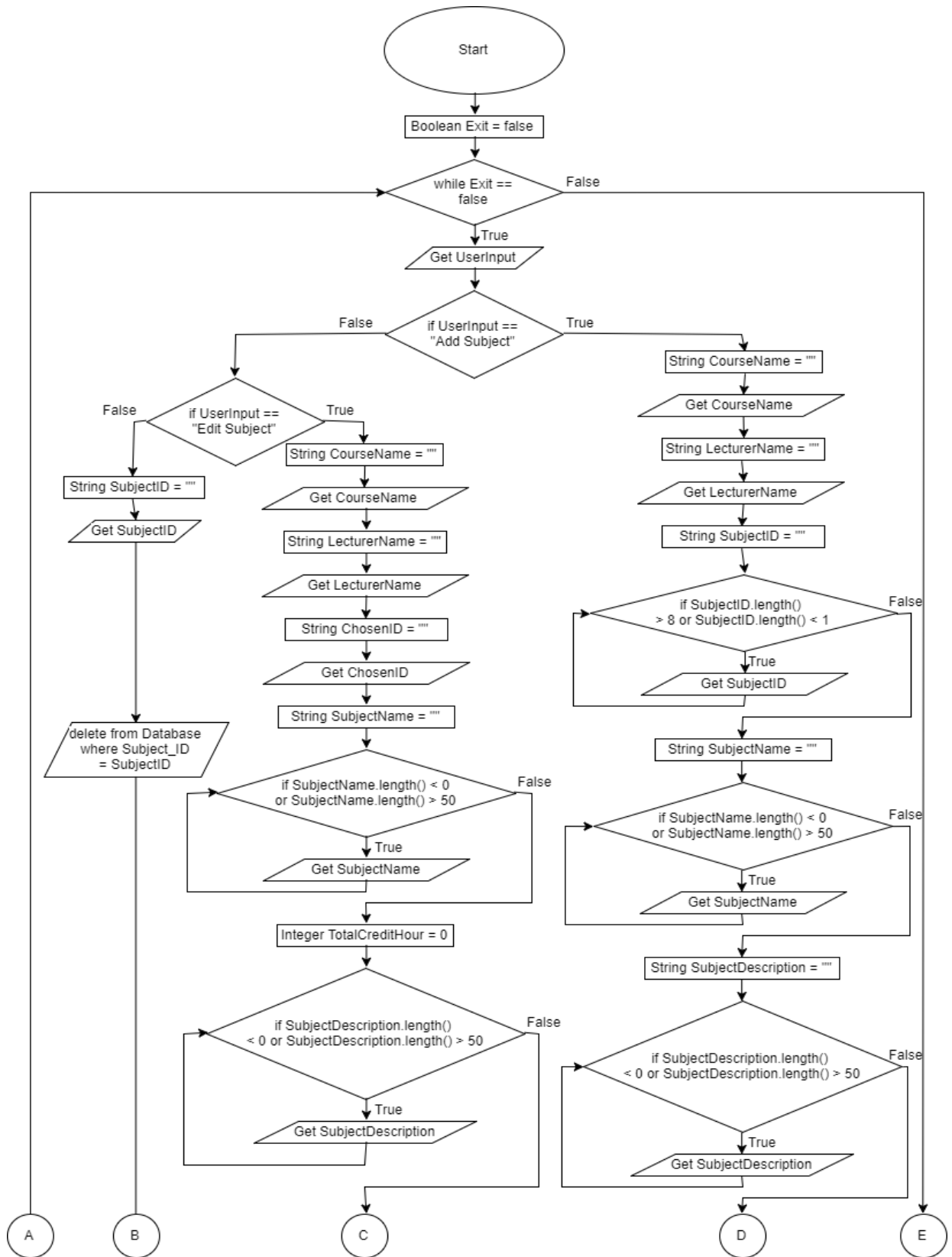
2. Item Management

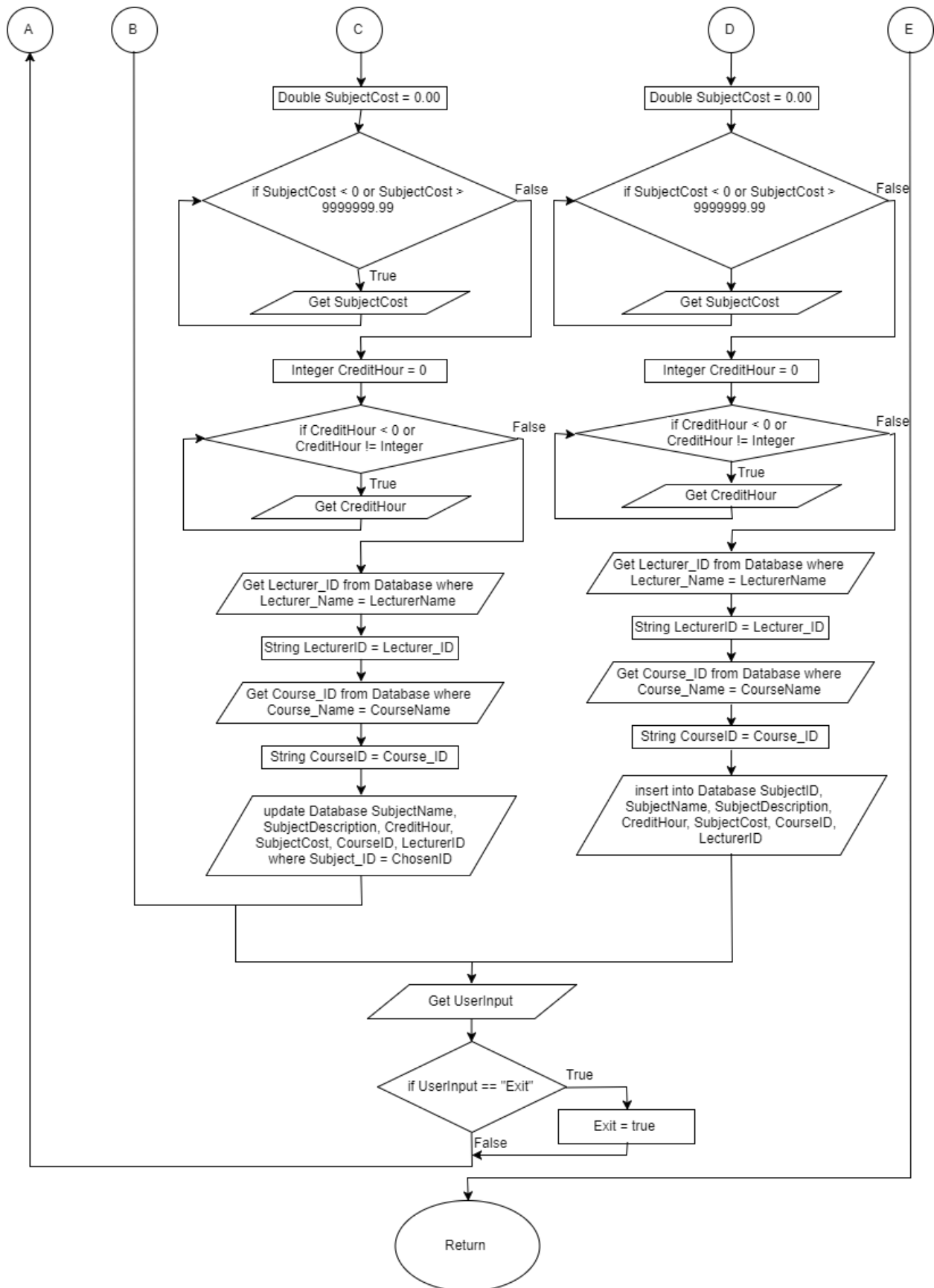
- Course Management Function



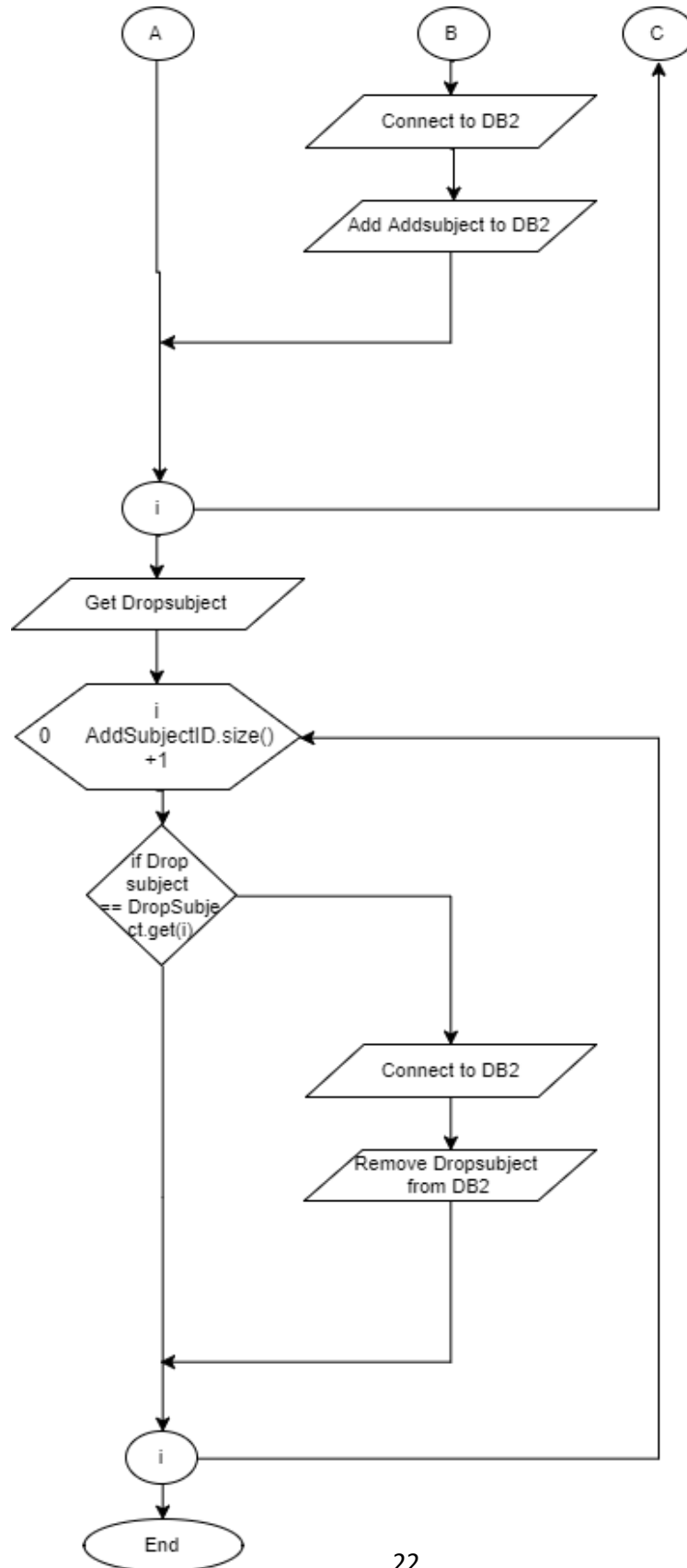


- Subject Management Function

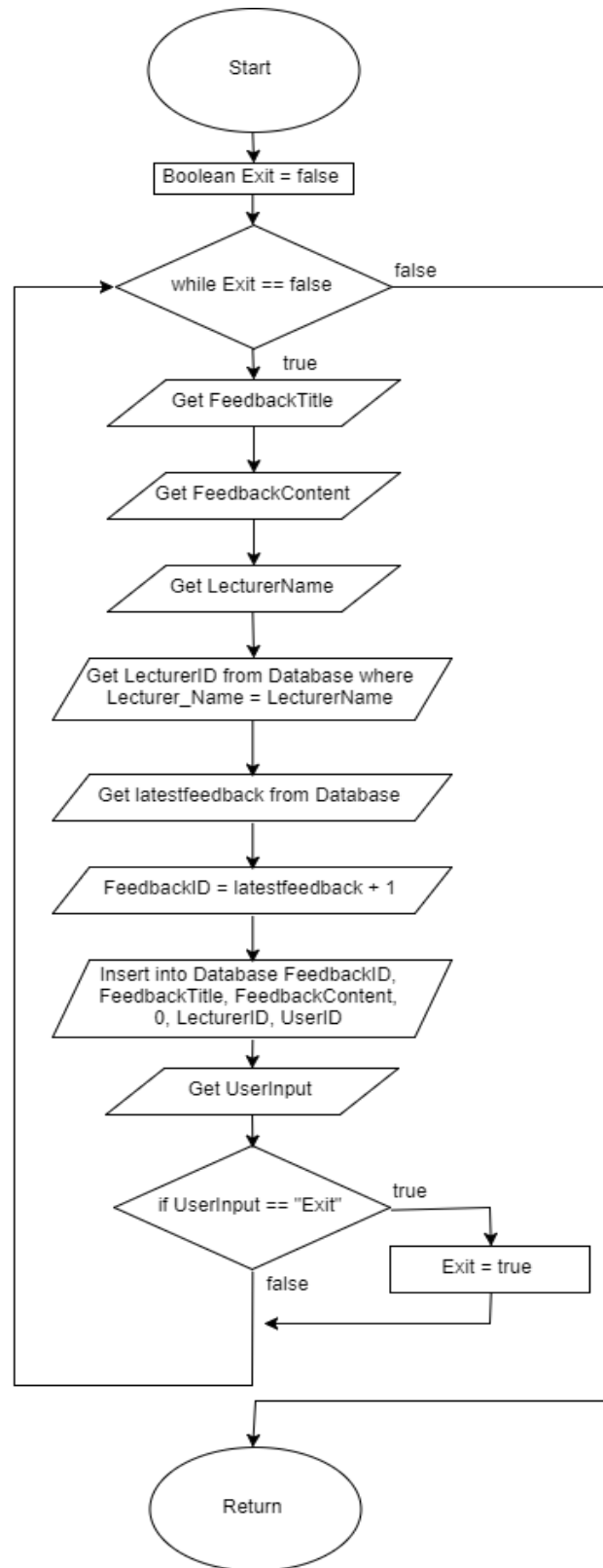




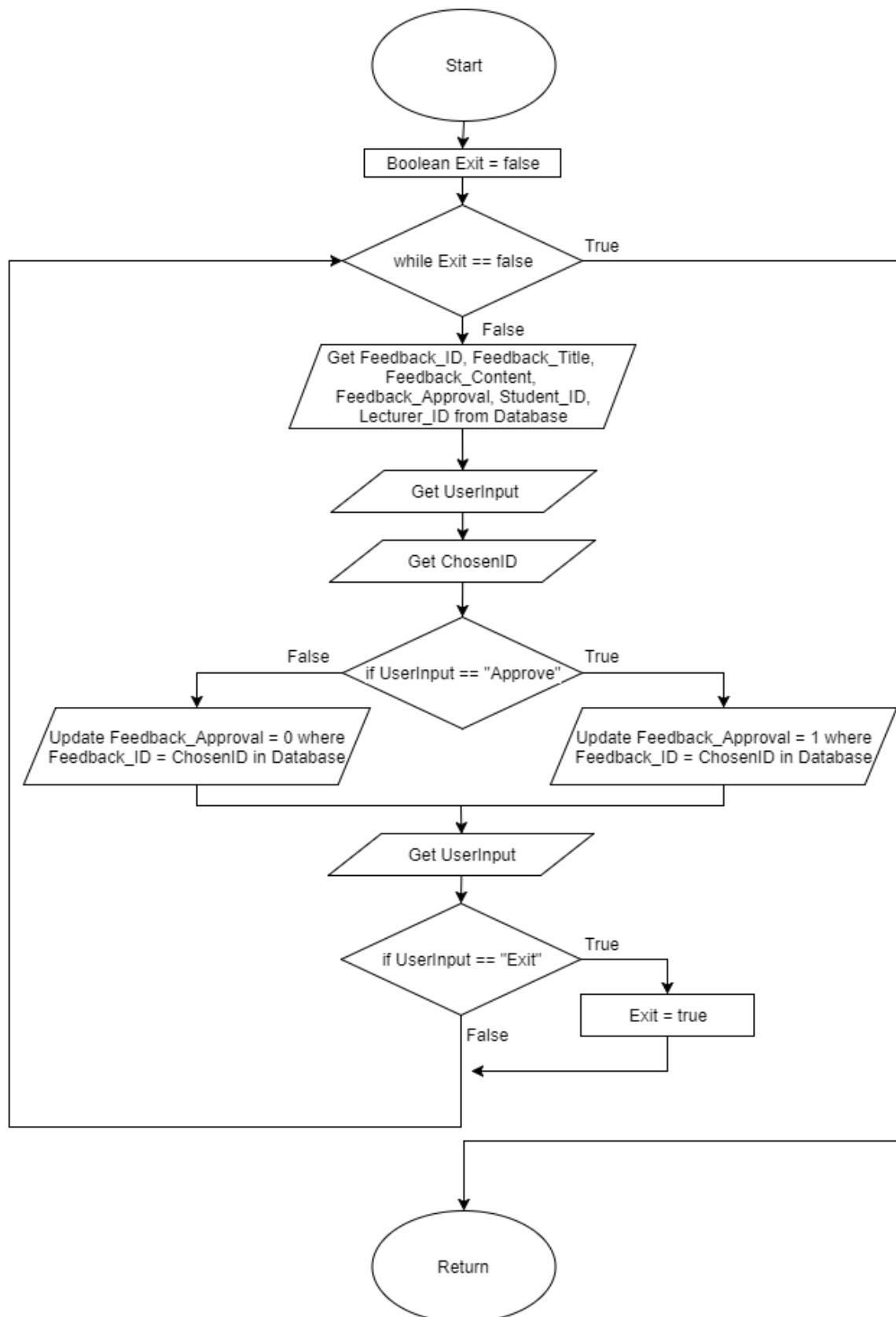
- Student Add and Drop Subject Function



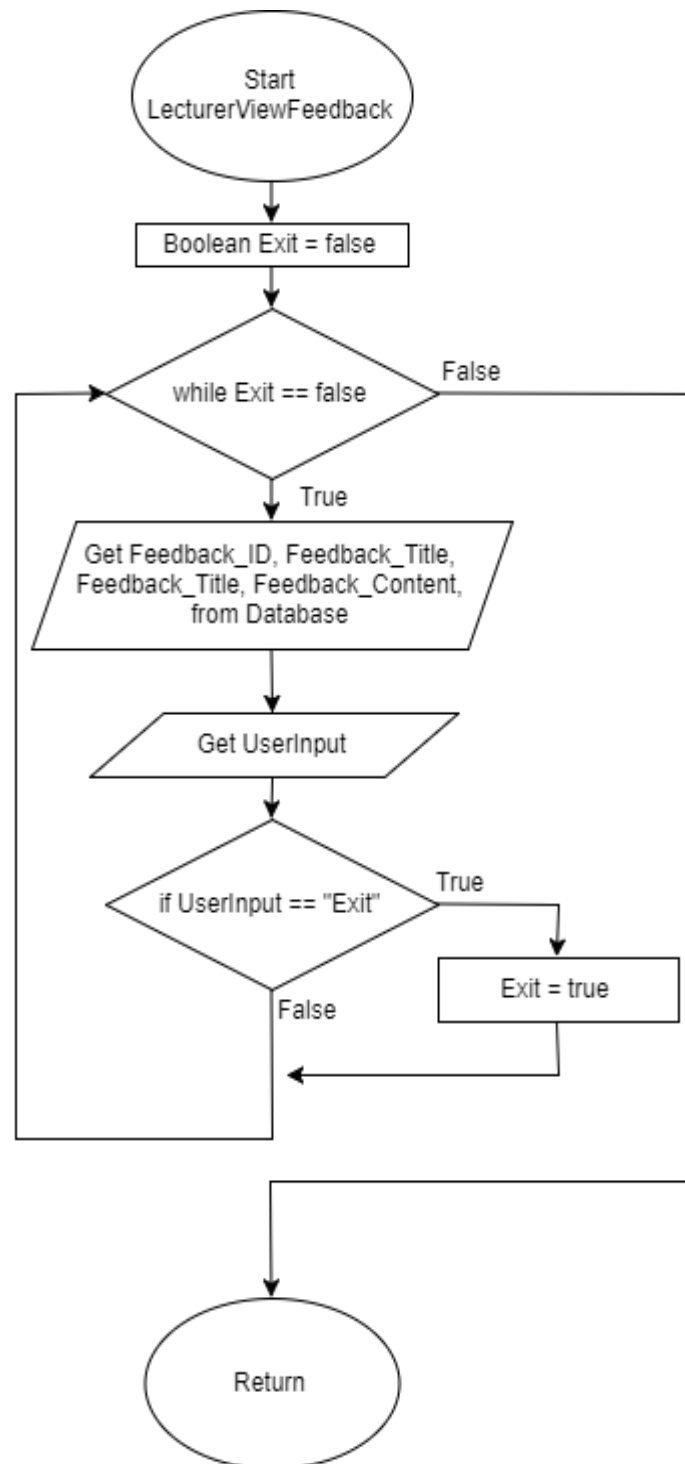
- Student Add Feedback Function



- Admin View Feedback Function

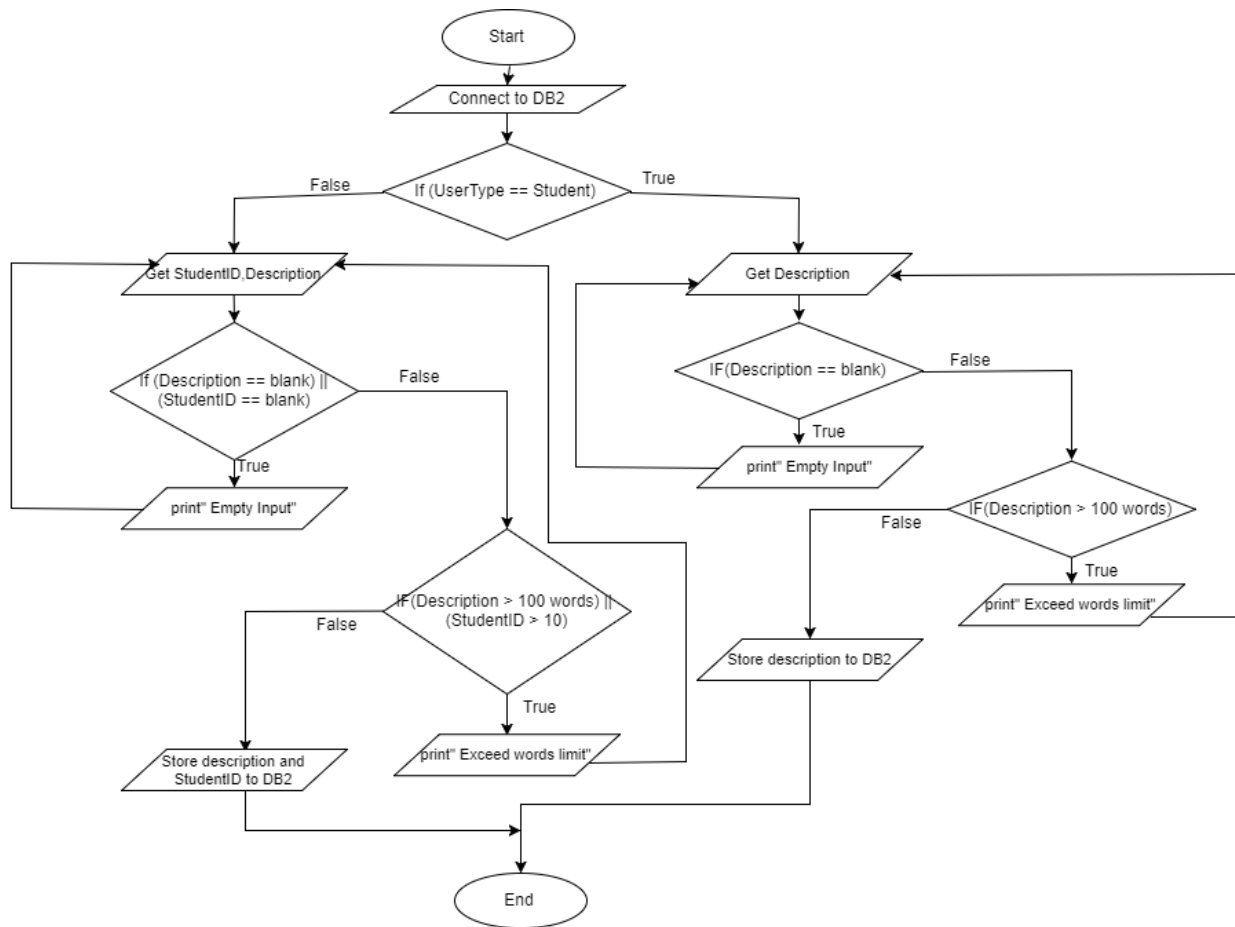


- Lecturer View Feedback Function

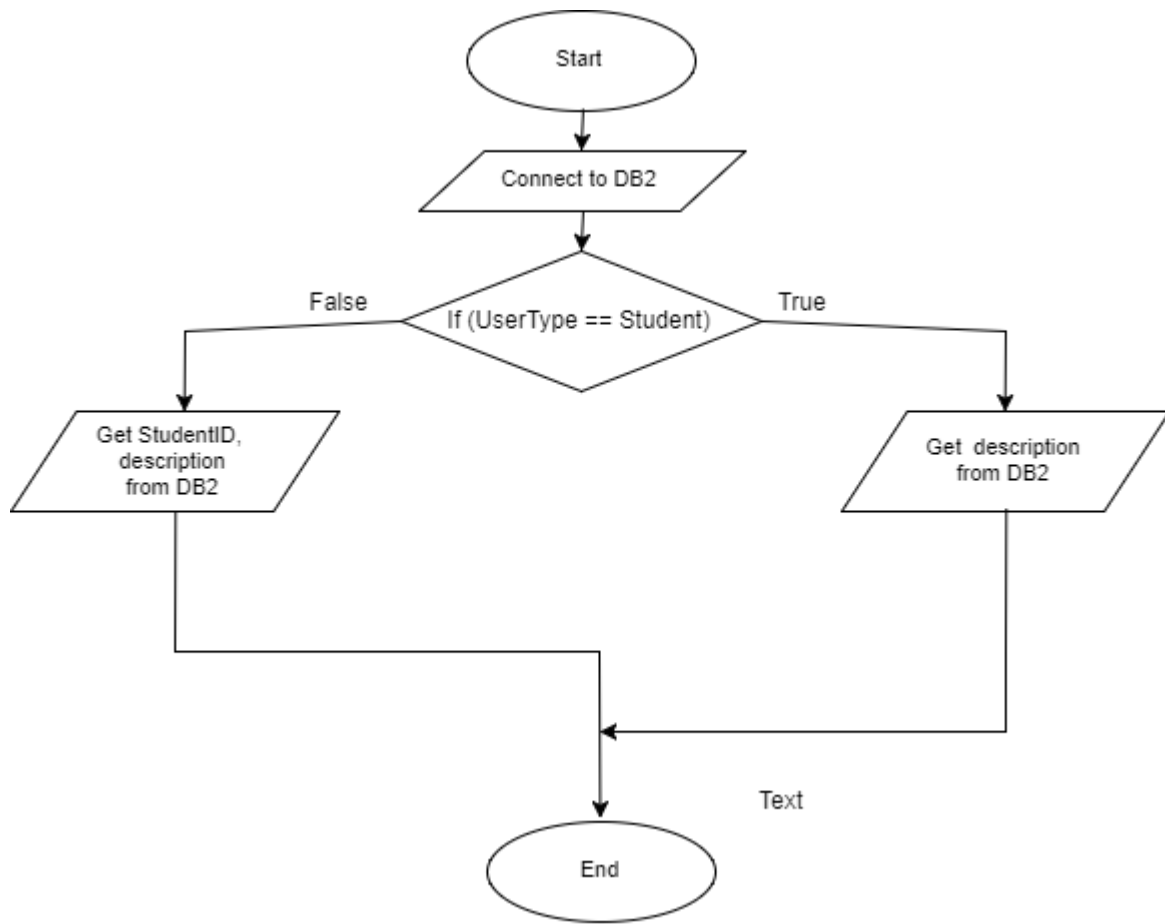


3. Transaction and Process Management

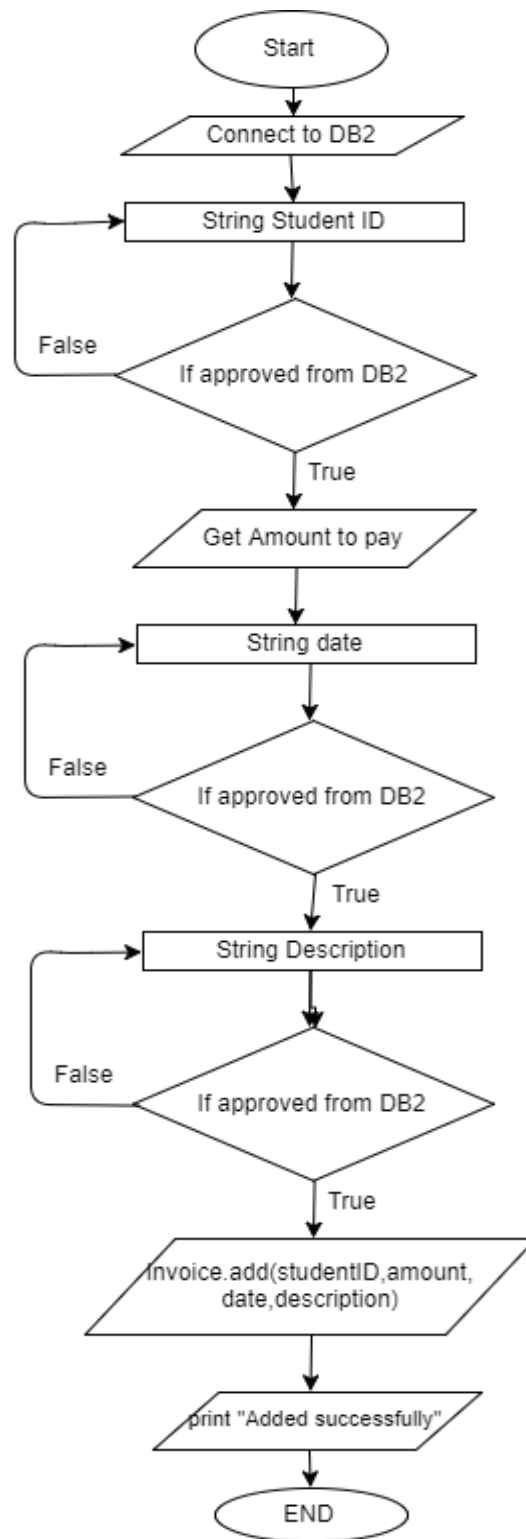
- Add Progress Report Function



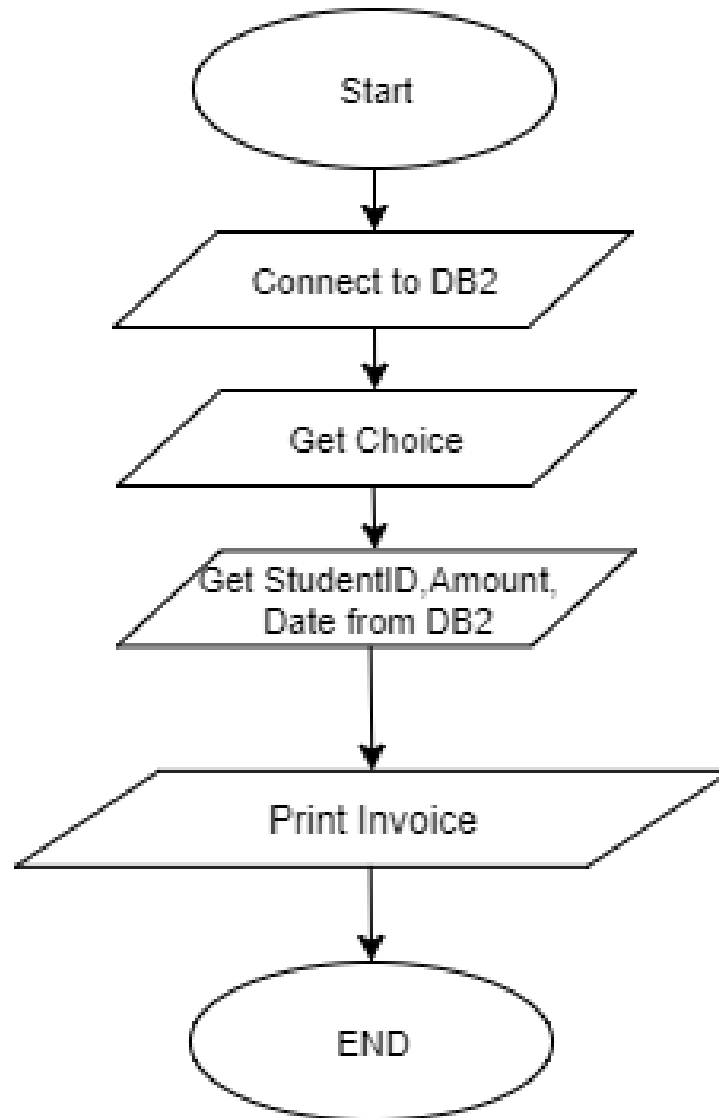
- View Progress Report Function



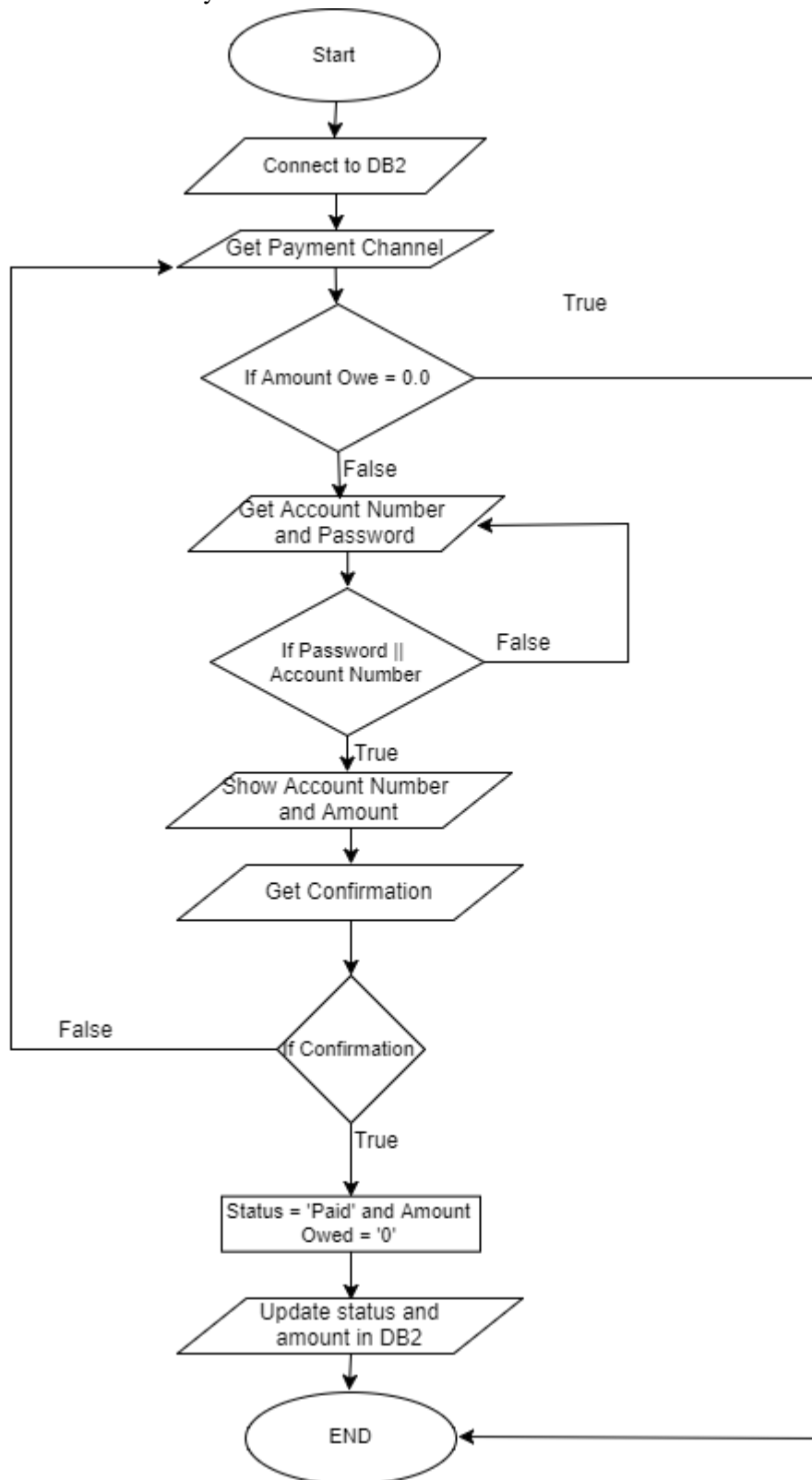
- Admin Add Invoice Function



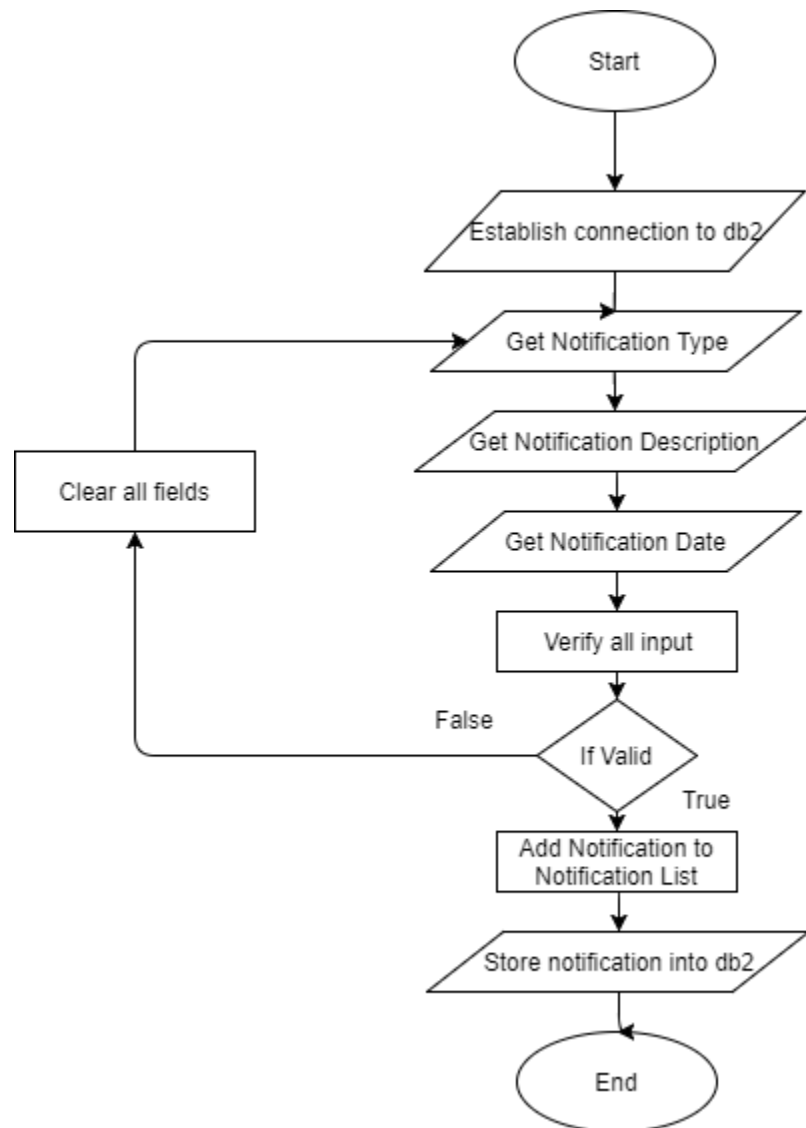
- Student View Invoice Function



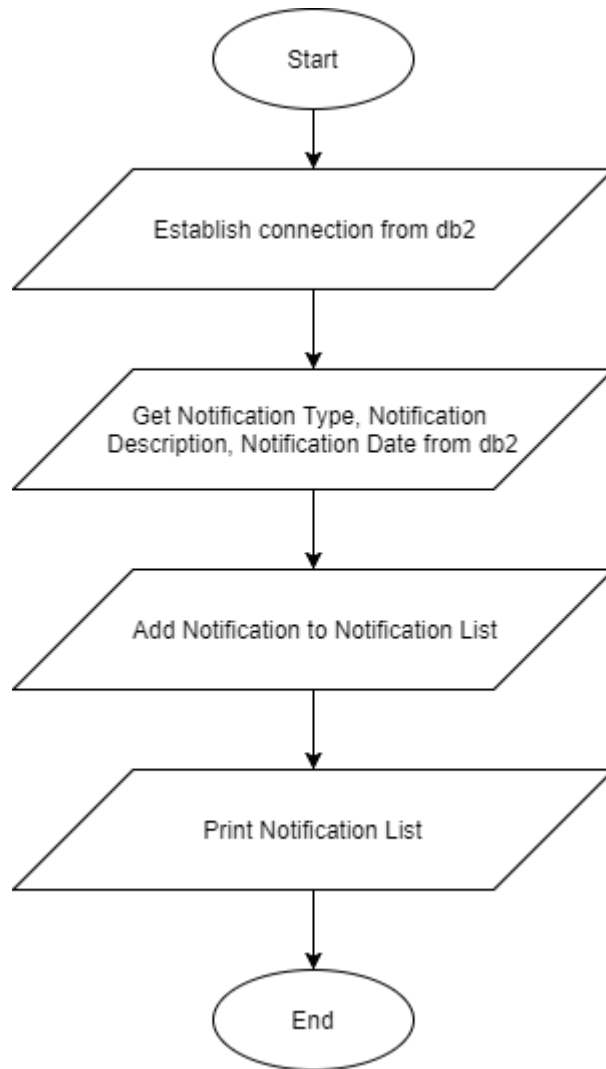
- Student Make Payment Function



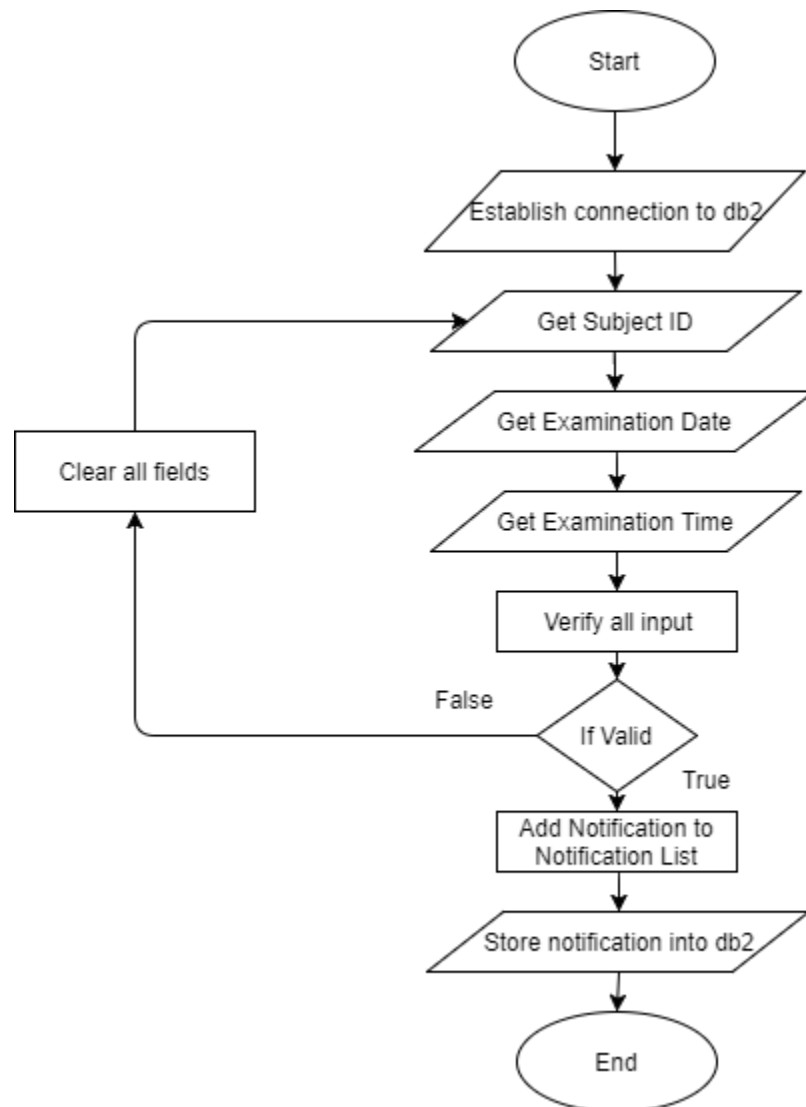
- Admin Generate Notification



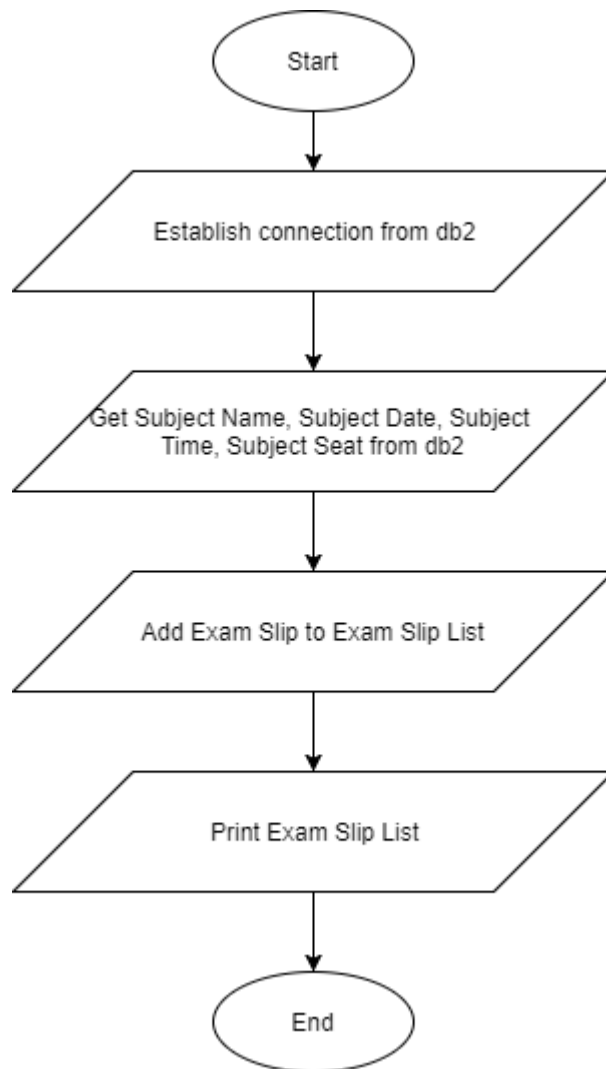
- User View Notification



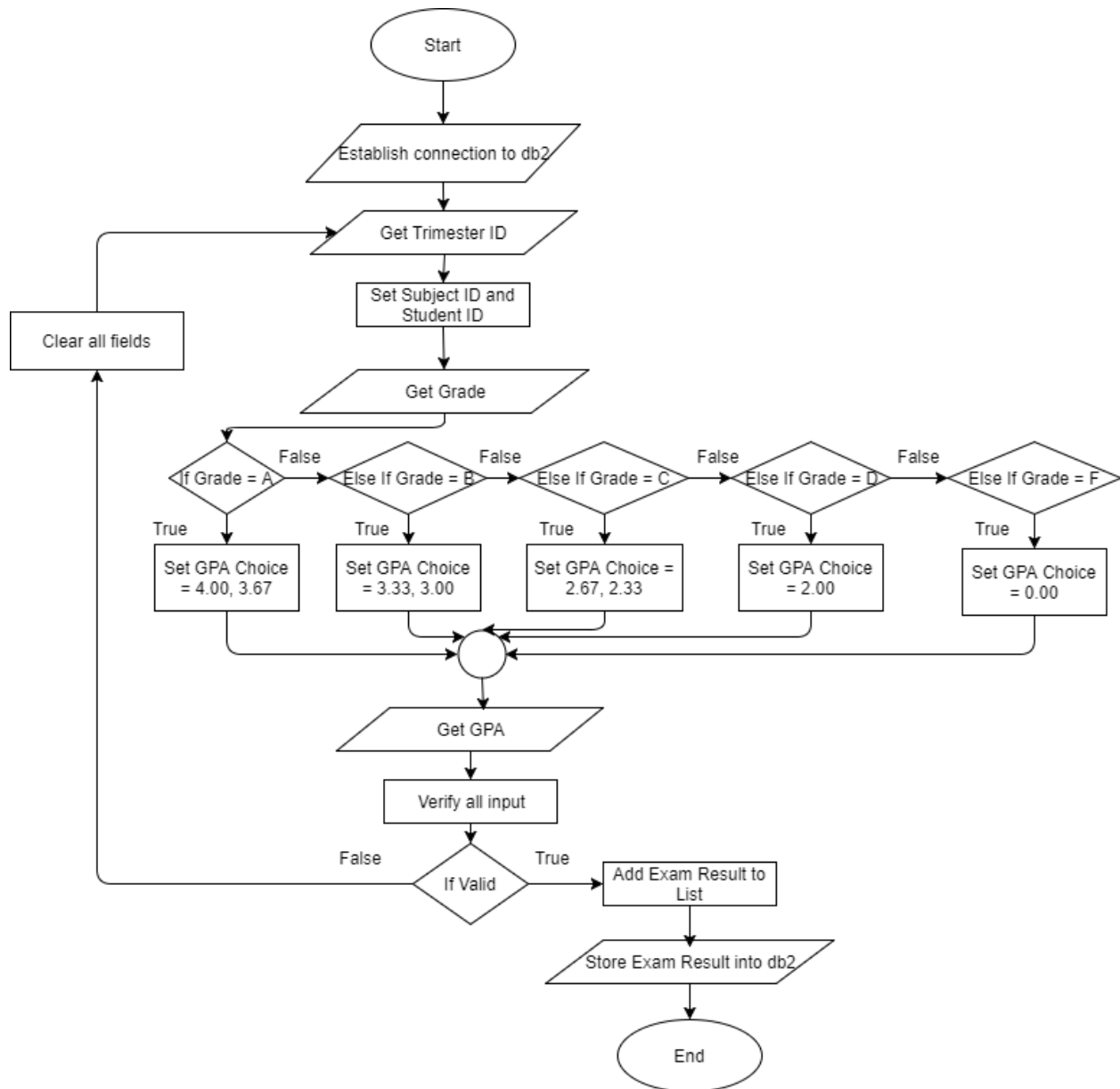
- Admin Generate Exam Slip



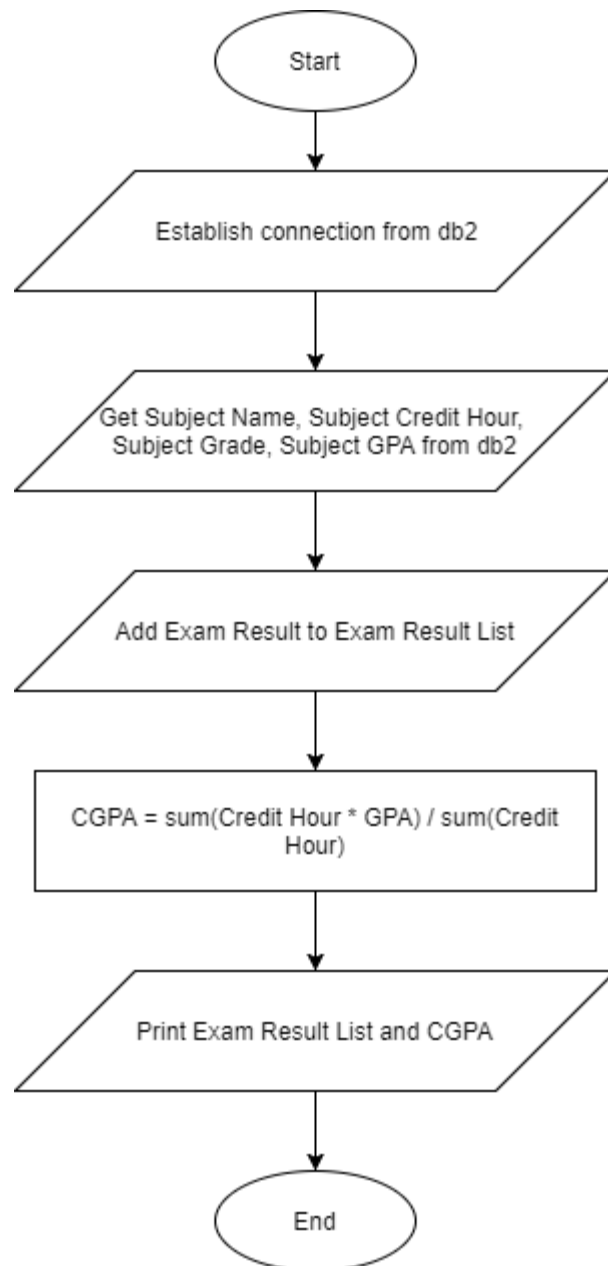
- Student View Exam Slip



- Admin Generate Exam Result



- Student Generate Exam Result



Testing Strategies

Unit Testing

We constantly input a lot of variations and combinations to see if the specific function does not crash or lead to a blank screen. We done this on each single new screen on our program and ensure that they are rectified the moment they are found. We also conduct smoke testing right after each function is tested properly.

Smoke Testing

Each team member constantly creates new builds for the program and constantly test them before implementing them into the main function. This will mean once the testing for the specific build is successful and is completed to our desired results, we combined them all into one huge structure to perform sandwich testing.

Sandwich Testing

In some functions, they will lead to other sub-functions, thus we cluster them into groups and test that specific function along with its sub-functions. The top modules will be filled with stubs until we manage to create a function to work with the interface. Since unit testing and smoke testing would have been done at this rate, the functions and sub-functions do not need to be vigorously tested as they should have been as they have already been tested properly in the previous tests.

Data Testing

	Valid Input Data	Invalid Input Data	Remark
Profile Management			
Log in Function	Admin: ID: ADM-01 Password: Any Character Lecturer: ID: LEC-0001 Password: Any Character Student: ID: 1151100653 Password: Any Character	ID: If ID does not match or exist in database record. Password: If password does not match or exist in database record.	Invalid: Show pop up indicating invalid ID or Password Valid: Proceed and gain access to the system.

Forgot Password	Admin: ID: ADM-01 New Password: Any Character Email: Email in database Lecturer: ID: LEC-0001 New Password: Any Character Email: Email in database Student: ID: 1151100653 New Password: Any Character Email: Email in database	ID: If ID does not match or exist in database record. New Password: not more than 30 characters Email: If email does not match or exist in database record.	Invalid: -pop out message "invalid ID or Password" Valid: -pop out message "Reset password successful"
-----------------	---	--	---

Edit Profile Information	<p>Admin, Lecturer, Student Address: any character not more than 100</p> <p>Contact: any character not more than 12</p> <p>Email: any character not more than 50</p>	<p>Address: any character more than 100</p> <p>Contact: any character more than 12</p> <p>Email: any character more than 50</p>	<p>Message shown when all data input is empty: 'Input valid data'</p> <p>Message shown when exceed maximum character capacity: 'Exceed maximum character input'</p> <p>Message shown when valid input data: 'Profile successfully updated'</p>
Add Profile Information	<p>Admin, Lecturer, Student Type: Student, Lecturer or Admin</p> <p>Course (Student): Course that is available</p> <p>Name: any character not more than 30</p> <p>Gender: M or F</p> <p>Race: Chinese, Malay, Indian, Others</p> <p>Address: any character not more than 100</p> <p>Contact: any character not more than 12</p> <p>Email: any character not more than 50</p>	<p>Type: any input that is not Student, Lecturer or Admin</p> <p>Course (Student):</p> <p>Name: any character not more than 30</p> <p>Gender: character that is not M or F</p> <p>Race: any input that is not Chinese, Malay, Indian, Others</p> <p>Address: any character more than 100</p> <p>Contact: any character more than 12</p> <p>Email: any character more than 50</p>	<p>Message shown when all data input is empty: 'Input valid data'</p> <p>Message shown when exceed maximum character capacity: 'Exceed character input'</p> <p>Message shown when valid input data: 'Profile successfully added'</p>

	Valid Input Data	Invalid Input Data	Remark
Transaction Process Management			
Payment - Verify Account	Student: Account Number: 7057788999 Password: 123456	Invalid Account Number: 7057788932 Unmatched Password: 908765	Invalid: - pop out message “invalid account number” - pop out message “invalid password” Valid: direct to payment window
Payment - Add Invoice	Admin: Student ID: 1151100653 Date: 2017-08-20 Description: Invoice for 2017/18 Semester	Student ID not in database: 1130979876 Invalid date: 20-08-2018 Invalid Description: (words more than 100) Empty Input at one of the column: (Blank)	Invalid: - pop out message “invalid student id” - pop out message “invalid date format” - show message “words exceed character limit” - pop out message “Cannot leave blank” Valid: add those input from admin to database

Progress Report - Add report	Lecturer: Student ID: 1151100653 Description: any character not more than 100 Student: Description: any character not more than 100	Student ID not in database: 1130979876 Empty Input: (Blank) Invalid Description: (Words more than 100)	Invalid: -pop out message "invalid student id" Message shown when invalid input data: 'Empty input' //occurs when empty input data 'Words exceed limit' //occurs when words exceed the limit Message shown when valid input data: 'Added successfully' Valid: add those input from lecturer to database
---------------------------------	---	--	---

Progress Report - View report	Lecturer: Description: any character not more than 100 Student: Description: any character not more than 100	Empty Input: (Blank) Invalid Description: (Words more than 100)	Message shown when invalid input data: 'Empty input' //occurs when empty input data 'Words exceed limit' //occurs when words exceed the limit Message shown when valid input data: 'Added successfully' Valid: add those input from lecturer and student to database
----------------------------------	---	--	---

Notification - Generate Notification	<p>Admin: Notification Type: Enrollment, Invoice, Exam Slip, Exam Result, Maintenance</p> <p>Notification Content: Notification message not more than 500 words.</p> <p>Notification Date: Notification Date with format YYYY-MM- DD</p> <p>Save button: Click save button</p>	<p>Admin: Notification Type: Anything other than the 5 options</p> <p>Notification Content: Message more than 500 words.</p> <p>Notification Date: Date format other than YYYY- MM-DD, invalid date smaller than current date, invalid value, e.g. 2017-02- 30</p>	<p>Invalid: Display message “Invalid notification type chosen”</p> <p>Display message “The content cannot be more than 500 words”</p> <p>Display message “The date format should be YYYY-MM-DD. Ensure the date value is correct”</p> <p>Valid: Display message “Notification created successfully!”</p>
--	--	--	---

Examination - Generate Exam Slip	<p>Student Subject Taken: Subject taken by at least one student</p> <p>Subject Date: Date with valid format: YYYY-MM-DD</p> <p>Subject Time: Time in 24-hour format, 09:00 and 14:00</p> <p>Subject Seat Number: Integer from 1 to 500</p> <p>Save button: Click save button</p>	<p>Student Subject Taken: Subject is not taken by any student</p> <p>Subject Date: Date with invalid format, and invalid date value</p> <p>Subject Time: Time in invalid format or exceed 5 characters or not the 2 options</p> <p>Subject Seat Number: Integer smaller than 1 and bigger than 500</p>	<p>Invalid: Student Subject Taken: Display message “No student is enrolled into this subject”</p> <p>Subject Date: Display message “The date format should be YYYY-MM-DD. Ensure the value is correct”</p> <p>Subject Time: Display message “The time should be 24 hour format HH:MM. “</p> <p>Subject Seat Time: Display message” the seat number cannot be less than one or more than 500”</p> <p>Valid: Display message “Exam slip generated successfully”</p>
--	---	--	---

<p>Examination - Generate Exam Result</p>	<p>Trimester ID: Any ID in database</p> <p>Student Subject Taken: List all the subject and student</p> <p>Subject Grade: “A”, “B”, “C”, “D”, “F”</p> <p>Subject GPA: Decimal, between 0.00 and 4.00, depending on grade</p> <p>Subject CGPA: Decimal, between 0.00 and 4.00, depending on CGPA of last semester</p> <p>Subject Credit Hour: Integer, 4</p> <p>Save button: Click save button</p>	<p>Trimester ID: ID not in database</p> <p>Student Subject Taken: Subject is not taken by the student</p> <p>Subject Grade: Grades exceeding 1 character or not in the options available</p> <p>Subject GPA: Wrong decimal format and invalid value less than 0.00 and more than 4.00</p> <p>Subject CGPA: Wrong decimal format and invalid value less than 0.00 and more than 4.00</p> <p>Subject Credit Hour: Not integer or negative value</p>	<p>Invalid Trimester ID: Display message “ID not found in database”</p> <p>Student Subject Taken: Display message “Invalid subject taken by student”</p> <p>Subject Grade: Display message “Invalid grade. Please choose the available grades.”</p> <p>Subject GPA: Display message “The grade format for GPA should be x.xx. The value should be between 0.00 and 4.00.</p> <p>Subject CGPA: Display message “The grade format for GPA should be x.xx. The value should be between 0.00 and 4.00.</p> <p>Subject Credit Hour: Display message “The credit hour should be integer or more than 0!”</p> <p>Valid: Display message “Exam result generated successfully”</p>
---	---	---	--

	Valid Input Data	Invalid Input Data	Remark
Item Management			
Add Course	Admin: Course ID: CS011 Course Name: MASTER OF ENGINEERING Total Credit Hour: 40	Admin: Course ID: (More than 5 characters) Course Name: (More than 50 characters) Total Credit Hour: (Not integer or negative integer)	<p>'ID should only have 5 characters or less!' // Occurs if Course ID exceeds 5 characters</p> <p>'Course Name should only have 50 characters or less!' // Occurs if Course Name exceeds 50 characters</p> <p>'Total Credit Hour should be integer only!' // Occurs if Total Credit Hour is not an integer.</p> <p>'Total Credit Hour should be more than 0!' // Occurs if Total Credit Hour is less than 0</p>
Edit Course	Admin: Course Name: MASTER OF ENGINEERING Total Credit Hour: 40	Admin: Course Name: (More than 50 characters) Total Credit Hour: (Not integer or negative integer)	<p>'Course Name should only have 50 characters or less!' // Occurs if Course Name exceeds 50 characters</p> <p>'Total Credit Hour should be integer only!' // Occurs if Total Credit Hour is not an integer.</p> <p>'Total Credit Hour should be more than 0!' // Occurs if Total Credit Hour is less than 0</p> <p>'ID should only have 8 characters or less!' // Occurs if Subject ID exceeds 8 characters</p> <p>'Subject Name should only have 50 characters or less!'</p>

Add Subject	Admin: Subject ID: TSN-1101 Subject Name: SOFTWARE ENGINEERING FUNDAMENTALS Subject Description: This course focuses on the fundamentals of software engineering. Credit Hour: 4 Subject Cost: 1111.10	Admin: Subject ID: (More than 8 characters) Subject Name: (More than 50 characters) Subject Description: (More than 100 characters) Credit Hour: (Not integer or negative integer) Subject Cost: (More than 999999999.99 or not positive value or not integer.	// Occurs if Subject Name exceeds 50 characters ‘Subject Description should only have 100 characters or less!’ // Occurs if Subject Description exceeds 100 characters ‘Credit Hour should be integer only!’ // Occurs if Credit Hour is not an integer or not positive value or more than 999999999.99.
-------------	--	---	--

Test Cases

- Login Function

Project Name: Postgraduate Management System						
Profile Management - Login						
Test Case ID: Test-01				Test Designed by: Foo Chuan Kang		
Test Priority: High				Test Designed date: 14/9/2017		
Module Name: Login Module				Test Executed by: Khew Kit Hahn		
				Test Executed date: 02/10/2017		
Test ID	Test Title	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-01-01	Login Authentication(Valid data)	The user must be registered to the system	1. Enter ID into User ID textfield 2. Enter Password into Password field 3. Click on the "login" button	Pop up message "Login Successful"	As expected	Pass
Test-01-02	Login Authentication(Invalid data)	The user must be registered to the system	1. Enter invalid ID into User ID textfield 2. Enter wrong Password into Password field 3. Click on the "login" button	Pop up message "Invalid ID or password"	As expected	Pass
Test-01-03	Reset Password Authentication(Valid data)	The user must be registered to the system	1. Click on "reset password" button on the login screen 2. Enter ID into ID textfield 3. Enter new password less than 30 characters 4. Enter email into Email textfield 5. Click "confirm" button	Pop up "Successfully changed password"	As expected	Pass
Test-01-04	Reset Password Authentication(Invalid data)	The user must be registered to the system	1. Click on "reset password" button on the login screen 2. Enter invalid ID into ID textfield 3. Enter new password more than 30 characters 4. Enter invalid email into Email textfield 5. Click "confirm" button	Pop up "Invalid ID, Email or exceed maximum password range"	As expected	Pass

- Profile Management Function

Project Name: Postgraduate Management System						
Profile Management - Profile						
Test Case ID: Test-02				Test Designed by: Foo Chuan Kang		
Test Priority: High				Test Designed date: 14/9/2017		
Module Name: Profile Module				Test Executed by: Alfred Loo		
				Test Executed date: 02/10/2017		
Test ID	Test Title	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-02-01	View Profile	Registered user of the system	1. Click on "View Profile" button	View profile information screen	As expected	Pass
Test-02-02	Edit Profile	Login as student, admin or lecturer	1. Click on "Edit Profile" button 2. Input address into address textfield within 100 words 3. Input contact into contact textfield within 12 words 4. Input email into email textfield within 50 words 5. Click "confirm" button	Prompt message "Profile successfully updated"	As expected	Pass
Test-02-03	Edit Profile(Invalid data)	Login as student, admin or lecturer	1. Click on "Edit Profile" button 2. Leave all field empty 5. Click "confirm" button 2. Input address into address textfield more than 100 words 3. Input contact into contact textfield more than 12 words 4. Input email into email textfield more than 50 words 5. Click "confirm" button	Prompt error message	As expected	Pass
Test-02-04	Add Profile	Login as admin	1. Click on "Add Profile" button 2. Select user type from "Type" drop down box 3. Enter name into Name textbox 4. Select gender from "Gender" drop down box 5. Select race from "Race" drop down box 6. Input address into address textfield within 100 words 7. Input contact into contact textfield within 12 words 8. Input email into email textfield within 50 words 9. Click "confirm" button	Prompt message "Profile successfully added"	As expected	Pass
Test-02-05	Add Profile(Invalid data)	Login as admin	1. Click on "Add Profile" button 2. Select user type from "Type" drop down box 3. Enter name into Name textbox 4. Select gender from "Gender" drop down box 5. Select race from "Race" drop down box 6. Input address into address textfield more than 100 words 7. Input contact into contact textfield more than 12 words 8. Input email into email textfield more than 50 words 9. Click "confirm" button	Prompt error message	As expected	Pass

- Subject Management Function

Project Name: Postgraduate Management System						
Item Management - Subject Management						
Test Case ID: Test-03				Test Designed by: Khew Kit Hahn		
Test Priority: Medium				Test Designed date: 17/09/2017		
Module Name: Subject Management Module				Test Executed by: Gan Teck Beng		
				Test Executed date: 02/10/2017		
Test ID	Test Title/Description	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-03-01	Add new subject into the database with valid data	1. Login as admin	1. Enter Subject ID with 8 characters 2. Enter Subject Name with 50 characters 3. Enter Subject Description with 100 characters 4. Enter Cost with positive decimal with 2 decimal places 5. Enter Credit Hour with positive integer 6. Select an existing course from course dropdown box. 7. Select an existing lecturer from lecturer dropdown box. 8. Click Submit	Subject will be entered into database normally	As expected	Pass
Test-03-02	Add new subject into the database with invalid data	1. Login as admin	1. Enter Subject ID with 9 characters or subject ID that already exists 2. Enter Subject Name with 51 characters 3. Enter Subject Description with 101 characters 4. Enter Cost with positive decimal with more than 2 decimal places or more than 999999.99 5. Enter Credit Hour with negative integer or not an integer 6. Did not select an existing course from course dropdown box. 7. Did not select an existing lecturer from lecturer dropdown box. 8. Click Submit	Error popup with specific errors.	As expected	Pass
Test-03-04	Edit subject in the database with invalid data	1. Login as admin 2. Subject must already exist in database and selectable in table	1. Click any subject in table 2. Change Subject ID into 9 characters 3. Change Subject Name into 51 characters 4. Change Subject Description into 101 characters 5. Change Cost with negative decimal with more than 2 decimal places 6. Change Credit Hour with negative integer 7. Did not select an existing course from course dropdown box. 8. Did not select an existing lecturer from lecturer dropdown box. 9. Click Submit	Error popup with specific errors.	As expected	Pass
Test-03-05	Delete subject in database	1. Login as admin 2. Course must already exist in database and selectable in table	1. Click any subject in table 2. Click the Delete Course button.	The subject will be deleted in the table and the database.	As expected	Pass
Test-03-06	Enroll subject	Login as student	1. Click "Add Subject" button 2. Select subject depends on number of subject enrolled or available 3. Click "confirm" button	Subject successfully enrolled	As expected	Pass
Test-03-07	Enroll subject(Invalid data)	Login as student	1. Click "Add Subject" button 2. Select duplicate subject or already enrolled 4 subjects 3. Click "confirm" button	Prompt error message	As expected	Pass
Test-03-08	Drop subject	Login as student	1. Click "Drop Subject" button 2. Select subject to drop from the table 3. Select yes from pop up window for confirmation	Subject will be dropped	As expected	Pass
Test-03-09	Drop subject	Login as student	1. Click "Drop Subject" button 2. Select subject to drop from the table with only one subject enrolled or none	Prompt error message	As expected	Pass

- Feedback Function

Project Name: Postgraduate Management System						
Item Management - Feedback						
Test Case ID: Test-04				Test Designed by: Khew Kit Hahn		
Test Priority: Low				Test Designed date: 17/09/2017		
Module Name: Feedback Module				Test Executed by: Foo Chuan Kang		
				Test Executed date: 02/10/2017		
Test ID	Test Title/Description	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-04-01	View Feedback	1. Login as admin 2. Feedback added by student already must exist in the database	1. Click any feedback in the table 2. Click approve or disapprove	The selected feedback will be approved or disapproved	As expected	Pass
Test-04-02	Add new Feedback	1. Login as student	1. Enter Feedback within 500 characters. 2. Select a lecturer if necessary. 3. Click Submit	Feedback will be submitted with the correct data from dropdown box.	As expected	Pass

- Course Management Function

Project Name: Postgraduate Management System						
Item Management - Course Management						
Test Case ID: Test-05				Test Designed by: Khew Kit Hahn		
Test Priority: Medium				Test Designed date: 15/09/2017		
Module Name: Course Management Module				Test Executed by: Alfred Loo		
				Test Executed date: 02/10/2017		
Test ID	Test Title/Description	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-05-01	Add new course into the database with valid data	1. Login as admin	1. Enter Course ID with 5 characters 2. Enter Course Name within 50 characters 3. Enter Total Credit Hour with a positive Integer 4. Click Submit	Course will be entered into database normally	As expected	Pass
Test-05-02	Add new course into the database with invalid data	1. Login as admin	1. Enter Course ID with 6 characters 2. Enter Course Name with more than 50 characters 3. Enter Total Credit Hour with negative Integer or not an Integer 4. Click Submit	Error popup with specific errors.	As expected	Pass
Test-05-03	Edit course in the database with valid data	1. Login as admin 2. Course must already exist in database and selectable in table	1. Click any course in table 2. Change Course Name into another entry within 50 characters. 3. Change Total Credit Hour into another entry with a positive integer 4. Click Submit	The edited course will be updated in the database and the table in the interface will also be updated	As expected	Pass
Test-05-04	Edit course in the database with invalid data	1. Login as admin 2. Course must already exist in database and selectable in table	1. Click any course in table 2. Change Course Name into another entry with more than 50 characters. 3. Change Total Credit Hour into another entry with a negative integer or not integers. 4. Click Submit	Error popup with specific errors.	As expected	Pass
Test-05-05	Delete course in database	1. Login as admin 2. Course must already exist in database and selectable in table	1. Click any course in table 2. Click the Delete Course button.	The course will be deleted in the table and the database.	As expected	Pass
Test-05-07	Delete course with students registered in database	1. Login as admin 2. Course must already exist in database and selectable in table 3. Student must already register for the course that is about to be deleted	1. Click any course in table 2. Click the Delete Course button.	Error popup with specific errors..	As expected	Pass

- Payment Function

Project Name: Postgraduate Management System						
Transaction Management - Payment						
Test Case ID: Test-06				Test Designed by: Gan Teck Beng		
Test Priority: High				Test Designed date: 14/9/2017		
Module Name: Payment Module				Test Executed by: Foo Chuan Kang		
				Test Executed date: 02/10/2017		
Test ID	Test Title	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-06-01	Generate Invoice Function with Valid Test Data Description: An admin is able to generate invoice successfully	1. Login as admin 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as admin 2. Choose invoice at the main page 3. Choose add invoice at the invoice page 4. Enter Student ID 5. Enter invoice content 6. Enter invoice date Click submit button	Generate invoice successfully	As expected	Pass
Test-06-02	Generate Invoice Function with Invalid Test Data. Description: An admin fails to generate an invoice	1. Login as admin 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as admin 2. Choose invoice at the main page 3. Choose add invoice at the invoice page 4. Enter Student ID 5. Enter invoice content 6. Enter invoice date Click submit button	Display error message	As expected	Pass
Test-06-03	View Invoice Function. Description: Student is able to view all invoice sorted by latest to oldest	1. Login as student 2. Ensure the module has been implemented completely.	1. User login as student 2. Choose payment at the main page 3. Choose view invoice at the payment page 4. Display all available invoice	Display all invoice successfully	As expected	Pass
Test-06-04	Make Payment with Valid Test Data Description: Student is able to make payment	1. Login as student 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as student 2. Choose payment at the main page 3. Choose make payment at the payment page 4. Choose payment channel 5. Enter student account number 6. Enter password Click submit button	Make payment successfully and the amount deduct from database	As expected	Pass
Test-06-05	Make Payment with Invalid Test Data Description: Student is unable to make payment	1. Login as student 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as student 2. Choose payment at the main page 3. Choose make payment at the payment page 4. Choose payment channel 5. Enter student account number 6. Enter password Click submit button	Display error message	As expected	Pass

- Notification Function

Project Name: Postgraduate Management System						
Transaction/Process Management - Notification						
Test Case ID: Test-07				Test Designed by: Alfred Loo		
Test Priority: High				Test Designed date: 14/09/2017		
Module Name: Notification Module				Test Executed by: Gan Teck Beng		
				Test Executed date: 02/10/2017		
Test ID	Test Title/Description	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-07-01	Generate Notification Function with Valid Test Data Description: An admin is able to generate notification successfully	1. Login as admin 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as admin 2. Choose notification at the main page 3. Choose generate notification at the notification page 4. Choose notification type 5. Enter notification content 6. Enter notification date Click submit button	Generate notification successfully	As expected	Pass
Test-07-02	Generate Notification Function with Invalid Test Data. Description: An admin fails to generate a notification	1. Login as admin 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as admin 2. Choose notification at the main page 3. Choose generate notification at the notification page 4. Choose notification type 5. Enter notification content 6. Enter notification date Click submit button	Display error message	As expected	Pass
Test-07-03	View Notification Function. Description: All user is able to view all notification	1. Login as admin/student/lecturer 2. Ensure the module has been implemented completely.	1. User login as admin/student/lecturer 2. Choose notification at the main page 3. Choose view notification at the notification page 4. Display all available notification	Display all notification successfully	As expected	Pass
Test-07-04	Pop up notification Description: A pop up notification is displayed on the user main menu	1. Login as admin/student/lecturer 2. Ensure the module has been implemented completely.	1. User login as admin/student/lecturer 2. Pop up the notification with today's date	Display a pop up notification	As expected	Pass

- Examination Module

Project Name: Postgraduate Management System						
Transaction/Process Management - Examination						
Test Case ID: Test-08				Test Designed by: Alfred Loo		
Test Priority: High				Test Designed date: 20/09/2017		
Module Name: Examination Module				Test Executed by: Khew Kit Hahn		
				Test Executed date: 02/10/2017		
Test ID	Test Title/Description	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-08-01	Generate Exam Slip Function with Valid Test Data Description: An admin is able to generate exam slip successfully	1. Login as admin 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as admin 2. Choose examination at the main page 3. Choose generate exam slip at the examination page 4. Choose Subject 5. Enter subject date 6. Enter subject time Click submit button	Generate exam slip successfully	As expected	Pass
Test-08-02	Generate Exam Slip Function with Invalid Test Data. Description: An admin fails to generate an exam slip	1. Login as admin 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as admin 2. Choose examination at the main page 3. Choose generate exam slip at the examination page 4. Choose Subject 5. Enter subject date 6. Enter subject time Click submit button	Display error message	As expected	Pass
Test-08-03	View Exam Slip Function. Description: A student is able to view his/her exam slip based on subject enrolled	1. Login as student 2. Ensure the module has been implemented completely.	1. User login as student 2. Choose examination at the main page 3. Choose view exam slip at the examination page 4. Display the exam slip	Display exam slip successfully	As expected	Pass
Test-08-05	Generate Exam Result Function with Invalid Test Data. Description: An admin fails to generate an exam result	1. Login as admin 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as admin 2. Choose examination at the main page 3. Choose generate exam result at the examination page 4. Choose Trimester 5. Choose subject grade 6. Choose subject GPA Click submit button	Display error message	As expected	Pass
Test-08-06	View Exam Result Function. Description: A student is able to view his/her exam result based on subject enrolled	1. Login as student 2. Ensure the module has been implemented completely.	1. User login as student 2. Choose examination at the main page 3. Choose view exam result at the examination page 4. Display the exam result	Display exam result successfully	As expected	Pass

- Progress Report Function

Project Name: Postgraduate Management System						
Transaction Management - Progress Report						
Test Case ID: Test-09				Test Designed by: Gan Teck Beng		
Test Priority: High				Test Designed date: 14/9/2017		
Module Name: Progress Report Module				Test Executed by: Alfred Loo		
				Test Executed date: 02/10/2017		
Test ID	Test Title	Precondition	Test Step	Expected result	Actual Result	Status(Pass/Fail)
Test-09-01	Generate Progress Report Function with Valid Test Data Description: Student and Lecturer is able to generate progress report successfully	1. Login as student/lecturer 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as lecturer/student 2. Choose Progress Report at the main page 3. Choose add progress report at the progress report page 4. Enter Student ID (only when login as lecturer) 5. Enter progress report content 5. Enter progress report date Click submit button	Generate progress report successfully	As expected	Pass
Test-09-02	Generate Progress Report Function with Invalid Test Data. Description: Student and lecturer fails to generate progress report	1. Login as student/lecturer 2. Ensure the module has been implemented completely. 3. Complete test data table	1. User login as lecturer/student 2. Choose Progress Report at the main page 3. Choose add progress report at the progress report page 4. Enter Student ID (only when login as lecturer) 5. Enter progress report content 5. Enter progress report date Click submit button	Display error message	As expected	Pass
Test-09-03	View Progress Report Function. Description: Student and lecturer able to view all progress report sorted by latest to oldest	1. Login as student or lecturer 2. Ensure the module has been implemented completely.	1. User login as student/lecturer 2. Choose progress report at the main page 3. Choose view progress report at the progress report page 4. Display all available progress report	Display all progress report successfully	As expected	Pass

Maintenance Plan

1. Object Oriented Structure

- Since our software was developed in an object-oriented manner, bug fixing will become easier as the bug will solely be in that specific class.
- If more features were to be added in the future, it can easily be added since adding more functions just mean adding more classes.
- If features are to be removed, buttons leading to those functions can easily be replaced with dummy values, enabling major repairs or temporary blocks.

2. Maintenance Log

- Since software is not built to be error free due to human imperfection and tend to overlook some minor mistake done during the development process, maintenance on the software is needed.
- During every maintenance service, whom, when, and what change is done to the software must be recorded to make future maintenance easier.
- If an error were to occur, the maintenance log can be used as a guidance to identify the affected area faster and deal with the problem efficiently.

3. Exception display

- Whenever there is an error found in program, it will display a message dialog, informing the user that what error had occurred and where is it located.

4. Maintenance Report

- All modification to the program will require to fill up the following form before modifying it.

Modification Request

Help Desk Number	Date Received	Product	Originator
Short Name of Problem		User Priority 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5 <input type="checkbox"/>	
Description:			
MR Number		Engineer Assigned	
SW Version	Test Level	Test ID	Priority
CSUs Affected		Documents Affected	
Work Around:			
Analysis Results:		<input type="checkbox"/> Corrective <input type="checkbox"/> Enhancement	
Software Engineer:	Date	<input type="checkbox"/> Re-creatable <input type="checkbox"/> Design Change <input type="checkbox"/> Document Change	
Software Engineer Comments:		Resource Estimate	
Software Manager:	Date	<input type="checkbox"/> Proceed <input type="checkbox"/> Hold <input type="checkbox"/> Reject	
Software Manager Comments:			
Quality Assurance:	Date	<input type="checkbox"/> Concur <input type="checkbox"/> Hold <input type="checkbox"/> Reject	
Quality Assurance Comments:			
Configuration Control Board:	Date	<input type="checkbox"/> Approved <input type="checkbox"/> Hold <input type="checkbox"/> Disapproved	

FORM XXXX Modification Request pg 1