PGDIT AUTOMATION SYSTEM

Software Requirements Specification and Analysis

Submitted to:

Mr. Emon Kumar Dey
Assistant Professor
Institute of Information Technology
University of Dhaka

Md. Saeed Siddik
Lecturer
Institute of Information Technology
University of Dhaka

Submitted by:

Eusha Kadir (BSSE 0708)
Satyaki Das (BSSE 0720)
3rd Year, 5th Semester, 7th Batch
Session: 2014-2015

Institute of Information Technology
University of Dhaka

Submission Date: 20th March, 2017

Letter of Transmittal

20th March, 2017
Coordinator
Software Project Lab 2
Institute of Information Technology
University of Dhaka

Subject: Submission of term report on "PGDIT Automation System."

Sir, We, the team on which the project on PGDIT Automation System was assigned, are submitting our report with due respect. We have tried our best for the report. However, it might lack perfection. So, may we therefore, hope that you would be kind enough to accept our report and oblige thereby.

Yours sincerely
Eusha Kadir (BSSE 0708)
Satyaki Das (BSSE 0720)

3rd Year, 5th Semester, 7th Batch
Institute of Information Technology
University of Dhaka
Session: 2014-15

Acknowledgement

We are highly indebted for getting such a tremendous opportunity to prepare the report on **PGDIT Automation System**. We would like to thank whole-heartedly our course teacher, Dr. Kazi Muheymin-Us-Sakib, Professor, Institute of Information Technology, University of Dhaka, for giving us guideline about how can we prepare this report. In completing this paper we have collected various important data and information from IIT. We are thankful to all of the works sited.

Special thanks to:
Mr. Shah Mostafa Khaled
Assistant Professor
Institute of Information Technology
University of Dhaka

Abstract

The study is made for the total management of PGDIT Program. The scope of study is to analyze the management system of PGDIT Program and to know the drawbacks of it. Also, enable to design SRS of this system. The object of this study is to develop a SRS (software requirements and specification) of the automation of PGDIT Program.

Contents

Chapter 1: Introduction 1
1.1 Purpose 1
1.2 Intended Audience 1
Chapter 2: Inception of PGDIT Automation System 2
2.1 Introduction 2
2.1.1 List of Stakeholders 2
2.1.2 Recognizing multiple viewpoint 3
2.1.3 Working towards collaboration 4
2.1.4 Requirements Questionnaire 5
2.2 Conclusion 5
Chapter 3: Elicitation of PGDIT Automation System 6
3.1 Introduction 6
3.2 Eliciting Requirements 6
3.2.1 Collaborative Requirement Gathering 6
3.2.2 Quality Function Deployment 7
3.3 Usage Scenarios 8
Chapter 4: Scenario Based Modeling of PGDIT Automation System 15
4.1 Definition of Use case 15
4.2 Use Case Diagrams 15
4.2.1 System Description from Level-0 use case 15
4.2.2 Level-0 Use Case Diagram 16
4.2.3 Level-1 Use Case Diagram 17
4.3 Activity Diagram 24
4.3.1 Authentication System 24
4.3.2 Admission System 30
4.3.3 Result Generation System 38
4.4 Swimlane Diagram 43
4.4.1 Authentication System 43
4.4.2 Admission System 49
4.4.3 Result Generation System 57
Chapter 5: Data Based Modeling of PGDIT Automation System 62
5.1 Noun Identification 62

5.1.1 Authentication System 62
5.1.2 Admission System 65
5.1.3 Result Generation System 69
5.2 Data Object Relation 72
5.2.1 PGDIT Automation System 72
5.3 ER Diagram 73
5.3.1 PGDIT Automation System 73
5.4 Relational Tables 74
5.5 Schema Tables 75
5.5.1 Authentication System 75
5.5.2 Admission System 77
5.5.3 Result Generation System 80
Chapter 6: Class Based Modeling of PGDIT Automation System 82
6.2 Coad and Yourdon's Six Selection Criteria 87
6.3 Noun-Verb Affiliation of Classes 89
6.4 Class Attributes and Class Methods 92
6.5 Class Cards 95
6.6 Class Responsibility Collaborator Diagram 98
Chapter 7: Behavioral Modeling of PGDIT Automation System 99
7.1 Event Initiator and Collaborators Identification 99
7.1.1 PGDIT Automation System 99
7.2 State Transition Diagrams 101
7.2.1 PGDIT Automation System 101
7.3 Sequence Diagrams 106
7.3.1 Authentication System 106
7.3.2 Admission System 107
Figure 7.3.2: Sequence Diagram for Admission System 107
7.3.3 Result Generation System 108
Figure 7.3.2: Sequence Diagram for Admission System 108
Chapter 8: Conclusion 109

Chapter 1: Introduction

This chapter is intended to specify the purpose of this document and the intended audiences of it.

1.1 Purpose

This document is the Software Requirement Specification (SRS) for PGDIT Automation System of Institute of Information Technology, University of Dhaka. It contains functional, non-functional and support requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

- The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
- The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
- The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.
- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

Chapter 2: Inception of PGDIT Automation System

In this chapter, the Inception part of the SRS will be discussed briefly.

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what is the scope and nature of the problem to be solved. The main goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. At project inception, we establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.
To establish the groundwork we have worked with the following factors related to the inception phases:

- List of stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Requirements questionnaire

2.1.1 List of Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will contribute input as requirements is elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: "Whom else do you think I should talk to?"
To identify the stakeholders we consulted with General Manager and asked her following questions:

- Who is paying for the project?
- Who will be using the project outcomes?
- Who gets to make the decisions about the project (if this is different from the money source)?
- Who has resources I need to get the project done?
- Whose work will my project affect? (During the project and also once the project is completed).

We identified the following stakeholders for our PGDIT Automation System.

Program Coordinator: The program coordinator of the PGDIT program will be affected deeply by the automation of this system. As most of the decisions of the system are made through him, his role/importance in the system is very large.
Director of IIT: The Director of Institute of Information Technology has the utmost authority in the institute. His role in this project is immense.
Teacher: The teachers of the program will also be greatly impacted by the software. Their role in the system is very important.
Receptionist: The receptionist of Institute of Information Technology, University of Dhaka currently handles the payment and application form distribution to applicants. As she is inextricably related to the PGDIT system, her responsibilities and therefore she will also be strongly affected by this system.
Developers: Developers are one of the stakeholders because they are also affected by this system. They develop this system and work for further development. If any system interruption occurs, they will find the problem and try to solve it.
University: Dhaka University will finance the project and it has some rules and regulations to maintain our system. We have to follow them strictly. They will also be affected by this system so we consider it as one of our stakeholder.
Testers: Testers are one of the stakeholders of this project because they will be testing the effectiveness and efficiency of the software project. If there are any problems in the system, they will be the ones to identify and report the problem.
Others: Other stakeholders of the project are the teachers, students, exam controller etc.

2.1.2 Recognizing multiple viewpoint

Different stakeholders achieve different benefits from the system. Consequently, each of them has a different view of the system. So we have to recognize the requirements from multiple points of view, as well as multiple views of requirements. Assumptions are given below:

Program Coordinator's viewpoint:

- User friendly and efficient system
- Error free system
- Easy to operate
- Minimum maintenance cost
- Availability of expected requirements within budget
- Storage of user related information

Director of IIT's viewpoint:

- Easy to access
- Strong authentication
- User friendly
- High security of user's information

Teacher's viewpoint:

- Easy to access
- Strong authentication
- User friendly

Receptionist's viewpoint:

- Easy to access
- User friendly
- Online guidance

Developer's viewpoint:

- Easy to develop
- No ambiguous requirement

University's viewpoint:

- User friendly
- Efficient system
- Error free system
- Cost within budget
- Less maintenance cost
- No disruption of rules and regulations

Tester's viewpoint:

- Easy to test project is useful for acquiring effective test results
- Clearly defined test cases

2.1.3 Working towards collaboration

Every stakeholder has their own requirements. There are some common and conflicting requirements of our stakeholder. That's why we followed the following steps to merge these requirements-

- Find the common and conflicting requirements
- Categorize them
- List the requirements based on stakeholder's priority points
- Make final decision about requirements

Common requirements:

- User friendly
- Efficient system
- Error free system
- Strong authentication

Conflicting requirements:

- Limited budget
- Cost within budget.
- High security of the system
- Easy access (Different stakeholder wants different type of access)

Final requirements: We finalize the following requirements based on stakeholder's priority point:

- User friendly system
- Strong authentication
- Maximum error free system. (5%-10% error is considerable)
- Restrict access to functionality of the system based upon user roles
- High security of member's information
- Automated submission of application

2.1.4 Requirements Questionnaire

We first ask the stakeholder some context free questions to understand the project's overall performance and goals. These questions are mentioned in section 2.1.1. These questions help us to identify the stakeholders of the project. Then we ask our next set of questions to better understand the problem and take stakeholder's opinion about the solution. The final set of question focused on the effectiveness of the communication activity itself.

2.2 Conclusion

Inception phase helped us to establish basic understanding about the PGDIT program of Institute of Information Technology, University of Dhaka, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system and establish a preliminary communication with our stakeholders.
In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both side (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

Chapter 3: Elicitation of PGDIT Automation System

The purpose of this chapter is to specify the elicitation part.

3.1 Introduction

To complete the elicitation step we face many problems like problems of scope, problems of volatility and problems of understanding. However, this is not an easy task. To help overcome these problems, we have worked with the Eliciting requirements activity in an organized and systematic manner.

3.2 Eliciting Requirements

Inception where Question and Answer approach is used; elicitation makes use of a requirements elicitation format that combines the elements of problem solving, elaboration, negotiation, and specification. It requires the cooperation of a group of end-users and developers to elicit requirements. To elicit requirements we completed following four works.

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenarios
4. Elicitation work products

3.2.1 Collaborative Requirement Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. We completed following steps. The meetings were conducted with the receptionist of IIT, DU, the program coordinator, the director of IIT and some teachers of the program; they were questioned about their requirements and expectations from the automated "PGDIT System". They were asked about the problems they were facing with the current manual system. At last we selected our final requirement list from the meetings.

3.2.2 Quality Function Deployment

Normal Requirements

1. User friendly interface
2. Facility for filling up and submitting application form online with supporting documents
3. Providing a notice menu in website
4. Generating a seat plan for written admission examination
5. Publishing list of eligible students for written examination
6. Publishing list of students who passed the written examination
7. Publishing list of eligible students for viva
8. Generating viva plan for each committee containing viva time
9. Publishing list of selected students and waiting list
10. Providing an authentication menu (login) in website for teachers, students, receptionist, program chairperson, exam controller and scrutinizing team
11. Grade submission for faculty members, grade evaluation by the concerned program coordinator
12. A list of such students (supplementary list) will also be visible in the dashboard.
13. After the publication of the supplementary examination result, a pdf containing the final grades and total marks of the students will be generated
14. Generating report card for each student containing the results of the semester

Expected Requirements

1. Generating admit card
2. Notifying students about their results and other important notices via email
3. Notifying the teachers of their shift through email
4. Providing secured system

Exciting Requirements

1. Notifying the teachers of their VIVA shift
2. Notifying students about their results and other important notices
3. Storing password using some hash functions

3.3 Usage Scenarios

PGDIT (Post Graduate Diploma in Information Technology) Program Automation

Authentication System

Users of this software will be provided with dashboards. There will be 6 types of users. These are: Teacher, Student, Exam controller, Program chairperson, Scrutinizing team, Receptionist. Accounts for exam controller, program chairperson, scrutinizing team and receptionist will be automatically created at the time of installation. In that case, the user must specify what type of account he/she is attempting to create along with other details necessary to create an account at the time of installation.

Student accounts will be created by the program chairperson when new students are enrolled in the program. To register as a teacher, one needs to fill up a form online with necessary information. Program chairperson can view all the requested accounts. If he/she approves this account, then a teacher account will be created. Otherwise, he/she will decline the request. The registration page will contain these fields: Username, Name, Father's name, Mother's name, email address, password, mobile number, and teacher registration number. Every student and teacher will be notified of their username and password for successful registration.

Each user would have to provide their username and password to log into their dashboard. If the username and password combination does not match, the user will be prompted to enter the username and password once again. If any user tries to log in with wrong password for 5 times continuously, the login attempt will be failed and the user will be blocked. The incident will also be reported to the account. Once successfully logged into the account, the user will be able to use the privileges granted to his/her type of account.

If a user forgets the password of respective account, he/she will be able to recover password. To recover password, one needs to submit his/her username. Then a recovery code will be sent to the user's email address. Then the user can submit his/her username and recovery code. If the username and recovery code are valid, then he/she can set a new password for his/her account.

All users can change their personal information, password etc. but they would have to provide their password for verification every time they wish to do so.

The template of registration page (for teachers) is given below:

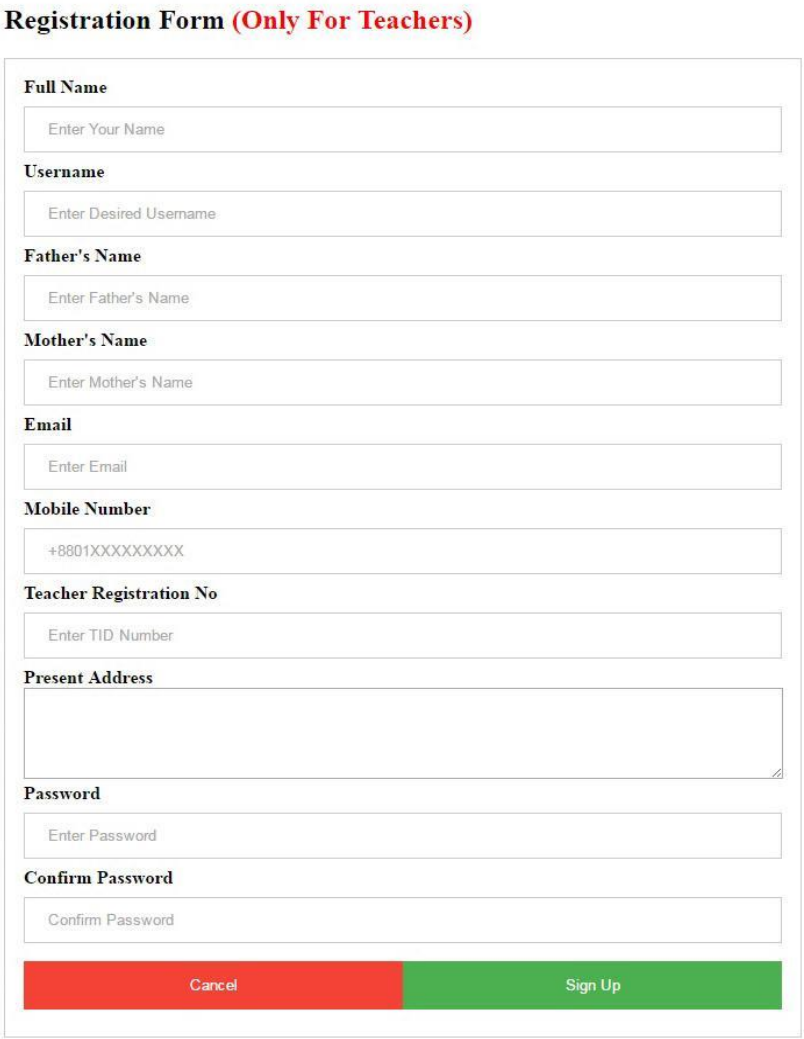


Figure 3.3.1.1: Registration From

Students can view their subject-wise marks, CGPA and exam results once it is uploaded in the server.

Teachers can upload student-wise marks of their respective courses, view notices and download attendance sheets from the site.

Exam controller will have access to the feature of generating seat plan for each room. He/she will also be able to initiate final result publication feature. He/she will also be able to generate tabulation sheet from his/her account.

Receptionist will be able to download the admission payment slip from the website. He/she will also be able to view candidate information from database. He/she will also verify payment status.

Scrutinizing team will provide validity of an application to the system. But for that they would have to download all the applications. So, scrutinizing team can view all applicants' information to verify.

Program chairperson will be able to update the email address of any user. He/she can update any Student's name (if it is necessary). In case of any crisis such as if any user forget his/her password and also can't log into his/her email, he/she can change the email address with the help of program chairperson. If any student did mistake in his/her name in the time of filling application form but want to correct his/her name, he/she will be able to do this modification with the help of program chairperson.

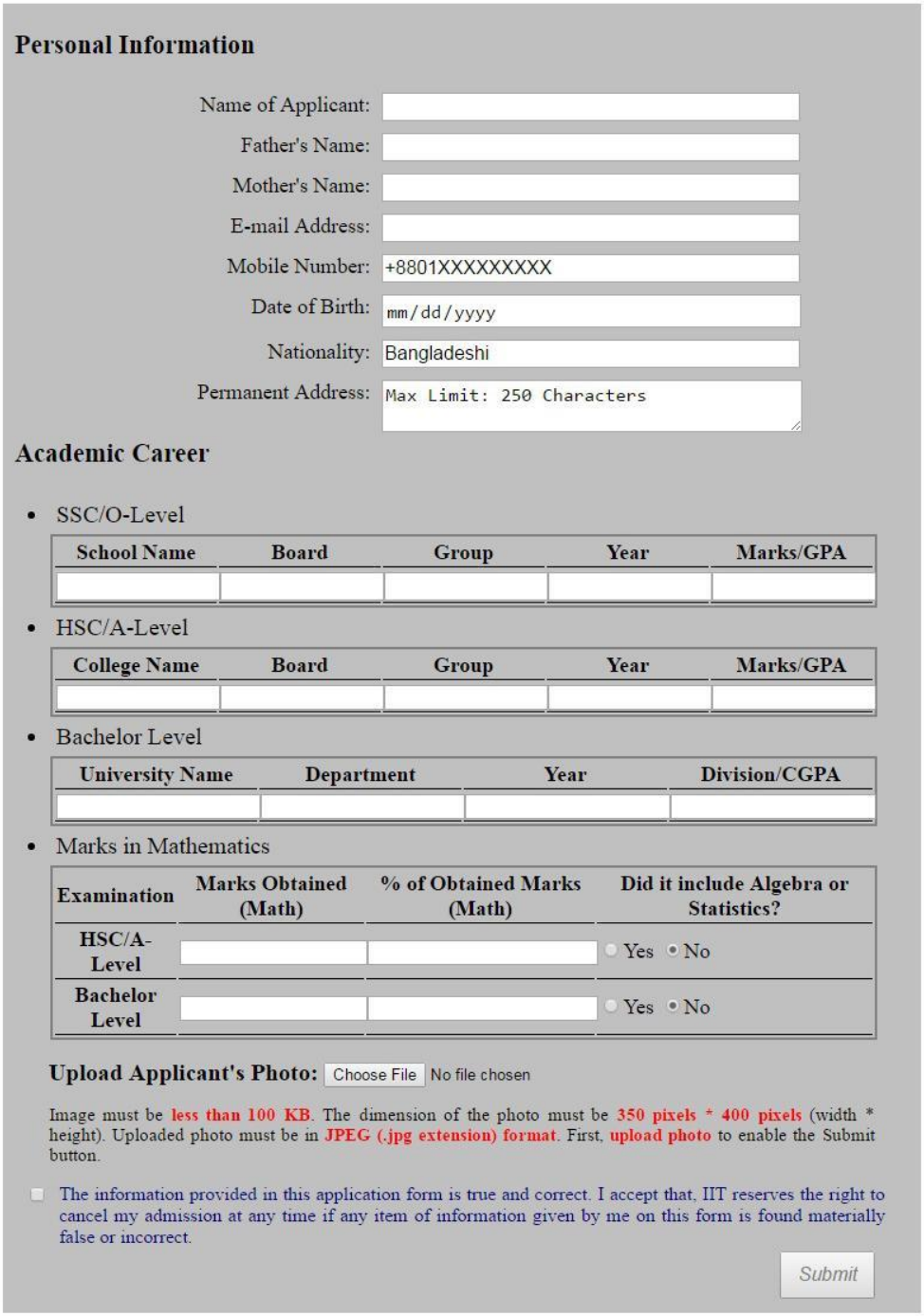
Admission System

The admission system is used to conduct the admission process into PGDIT program in its entirety. At first the program chairperson will upload the admission circular on the website. Afterwards, the site used for submitting application forms will be activated on the starting date mentioned in the circular. Interested candidates will be able to apply for admission online by filling up the application form. The application form will contain these fields: Name of Applicant, Father's Name, Mother's Name, E-mail Address, Mobile Number, Permanent Address, Date of Birth, and Nationality. The application form will also contain a table for academic information.

An applicant needs to upload his/her photo to enable "Submit" button. Image size must be less than 100 KB. Image must be in JPEG (.jpg extension) format. The resolution of the uploaded image is fixed (350 pixels * 400 pixels). After the applicant confirms the correctness of the information and agree to the terms and conditions of the program. Afterwards, he/she will click the "Submit" button to continue. Then the software will generate a 6-digit application ID. The software will also store the given information in the database. The applicant will receive the application ID and the submitted information.

The receptionist will be able to update the payment status of each application ID when the applicant pays the admission fees. After the completion of the fees, the application scrutinizing team will be able to download the list applicants who have paid their admission fees. Scrutinizing team will filter out all the applications. Then a list of students who are eligible to seat the admission examination will be published online alongside a unique exam roll number for each eligible applicant.

The snippet of the application form is given below. All the necessary fields for application form can be viewed in that snippet.



Personal Information

Name of Applicant:
Father's Name:
Mother's Name:
E-mail Address:
Mobile Number: +8801XXXXXXXXX
Date of Birth: mm/dd/yyyy
Nationality: Bangladeshi
Permanent Address: Max Limit: 250 Characters

Academic Career

- SSC/O-Level

School Name	Board	Group	Year	Marks/GPA

- HSC/A-Level

College Name	Board	Group	Year	Marks/GPA

- Bachelor Level

University Name	Department	Year	Division/CGPA

- Marks in Mathematics

Examination	Marks Obtained (Math)	% of Obtained Marks (Math)	Did it include Algebra or Statistics?
HSC/A-Level			Yes No
Bachelor Level			Yes No

Upload Applicant's Photo: Choose File No file chosen

Image must be less than 100 KB. The dimension of the photo must be 350 pixels * 400 pixels (width * height). Uploaded photo must be in JPEG (.jpg extension) format. First, upload photo to enable the Submit button.

The information provided in this application form is true and correct. I accept that, IIT reserves the right to cancel my admission at any time if any item of information given by me on this form is found materially false or incorrect.

Submit

Figure 3.3.2.1: Application From

The software will then generate the seat planning of the examination and all teachers will be notified of their respective viva shifts (if any). The exam controller will be able to download

the seat plan from his/her dashboard and make the viva schedule. He/she will also be able to download the attendance sheets generated by the software.
The software will generate the admit card for each valid application which the applicants can download from the website by providing his/her application ID and date of birth. The template of the admit card is given below:

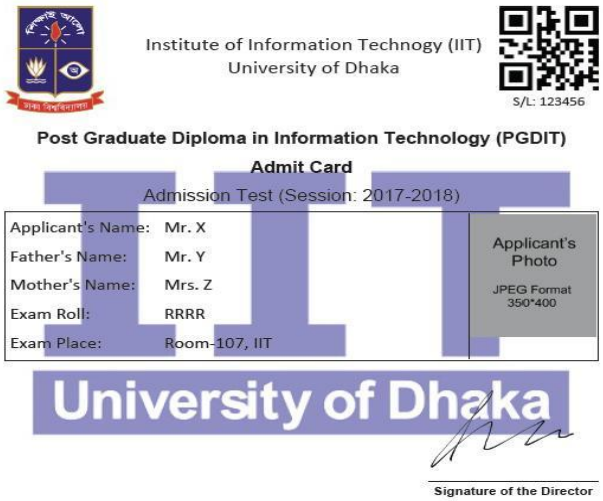


Figure 3.3.2.2: Admit Card

After the inspection of the exam papers the exam controller will upload the obtained marks against each application ID. The software will then sort the marks obtained by the examinees. The exam controller will then set the number of examinees who will be called for the viva. The exam controller will then provide the viva boards & shifts to the software. A list of eligible students along with their viva schedules will also be published online. The teachers will also be notified of their shifts.
After the viva is conducted, the exam controller will provide the viva marks of the examinees. The software will then sort the examinees by both their written exam marks and viva marks. Then a list of selected students along with two waiting lists will be published. The merit list will contain 40 entries, the first waiting list will have 10 and the second waiting list will have 5 entries. The receptionist will then verify the payment of the admission fees. After the admission fees are paid, a list of enrolled students will be published online.

Result Generation System

The program chairperson will enroll a student in some courses. To enroll in a course, a student may need to fulfill some prerequisites. Without filling up the prerequisites, one student will not enroll in a course. The prerequisite course table is given below:

Course Code	Course Name	Credit Hour	Prerequisite
PGD101	Computer Fundamentals and Office Automation	3	
PGD104	Structured Programming	3	
PGD203	Data Communication and Computer Networks	3	
PGD105	Introduction to Software Engineering	3	
PGD201	Data Structure & Algorithm	3	PGD 104
PGD202	Object Oriented Programming	3	PGD 104
PGD106	Operating System Concepts & UNIX OS	3	PGD 101, PGD 104
PGD107	Internet Programming	3	PGD 104, PGD 105
PGD204	DBMS & XML	3	PGD 107
PGD207	Mobile Application	3	PGD 106, PGD 201, PGD 202
PGD206	Microcontroller & Embedded System	3	PGD 106, PGD 201, PGD 202
PGD208	.Net Technology	3	PGD 201, PGD 202, PGD 107
PGD 210	Project for PGDIT	6	Completion of 30 credits

Table 3.3.3.1: Prerequisite Table

Program Chairperson will select a teacher as the instructor of a course. The result generation system provides term examination results in quick time. Firstly, teachers will provide the total marks obtained by each student in their (teachers') respective courses and the software will generate grade from it. A list of students who would have to take supplementary exam will be generated and uploaded in the website. After the exam papers of supplementary exams are evaluated, the teachers will provide the supplementary exam results to the

software. After the publication of the supplementary examination result, a pdf containing the final grades and total marks of the students will be generated.

The grade generation policy is given below:

Grading Policy

Grades in each course will be assigned as follows: (As per University rule)

Marks	Letter Grade	Numeric Grade	Comments
80% or above	A+	4.00	Excellent
>= 75% but < 80%	A	3.75	Better
>= 70% but < 75%	A-	3.50	Good
>= 65% but < 70%	B+	3.25	Above Average
>= 60% but < 65%	B	3.00	Average
>= 55% but < 60%	B-	2.75	Below Average
>= 50% but < 55%	C+	2.50	Satisfactory
>= 45% but < 50%	C	2.25	Not Satisfactory
>= 40% but < 45%	D	2.00	Pass
Less than 40%	F	0.00	Fail
	I		Incomplete
	W		Withdrawn

The GPA (Grade Point Average) and CGPA (Cumulative Grade Point Average) will be calculated as follows:

$$GPA\ or\ CGPA = \frac{\sum Grade\ Points\ in\ a\ Course \times Credit\ for\ that\ course}{Total\ Credits\ Taken}$$

Figure 3.3.3.1: Grading Policy

Afterwards, the teachers will upload a pdf containing the detailed results of all the students in their (teachers') courses. This pdf will be sent to the exam controller and he/she will be able to download it. If any modification of the results is required, the teacher will update the given total marks list before the result updating facility is turned off. The exam controller will turn off result updating facilities. Then the software will generate the result and upload it online so that only the program chairperson and the exam controller can view the results. Then the software will generate a report card for each student containing the results of the courses taken by him/her. All students will be notified to collect their report card. The software will also keep track of which courses the students passed in order to handle prerequisites of some courses.

Chapter 4: Scenario Based Modeling of PGDIT Automation System

4.1 Definition of Use case

A use case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders**.** In essence, a use case tells a stylized story about how an end-user interacts with the system under a specific set of circumstances. A use case diagram simply describes a story using corresponding actors, who perform important role in the story and makes the story understandable for the users. The first step in writing a use case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators.

Primary Actor: Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor: Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

4.2 Use Case Diagrams

Use case diagrams give the non-technical view of overall system.

4.2.1 System Description from Level-0 use case

After analyzing the user story we found eight actors who will directly use the system as a system operator. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information.

Following are the actors of PGDIT Automation System–

4.2.2 Level-0 Use Case Diagram

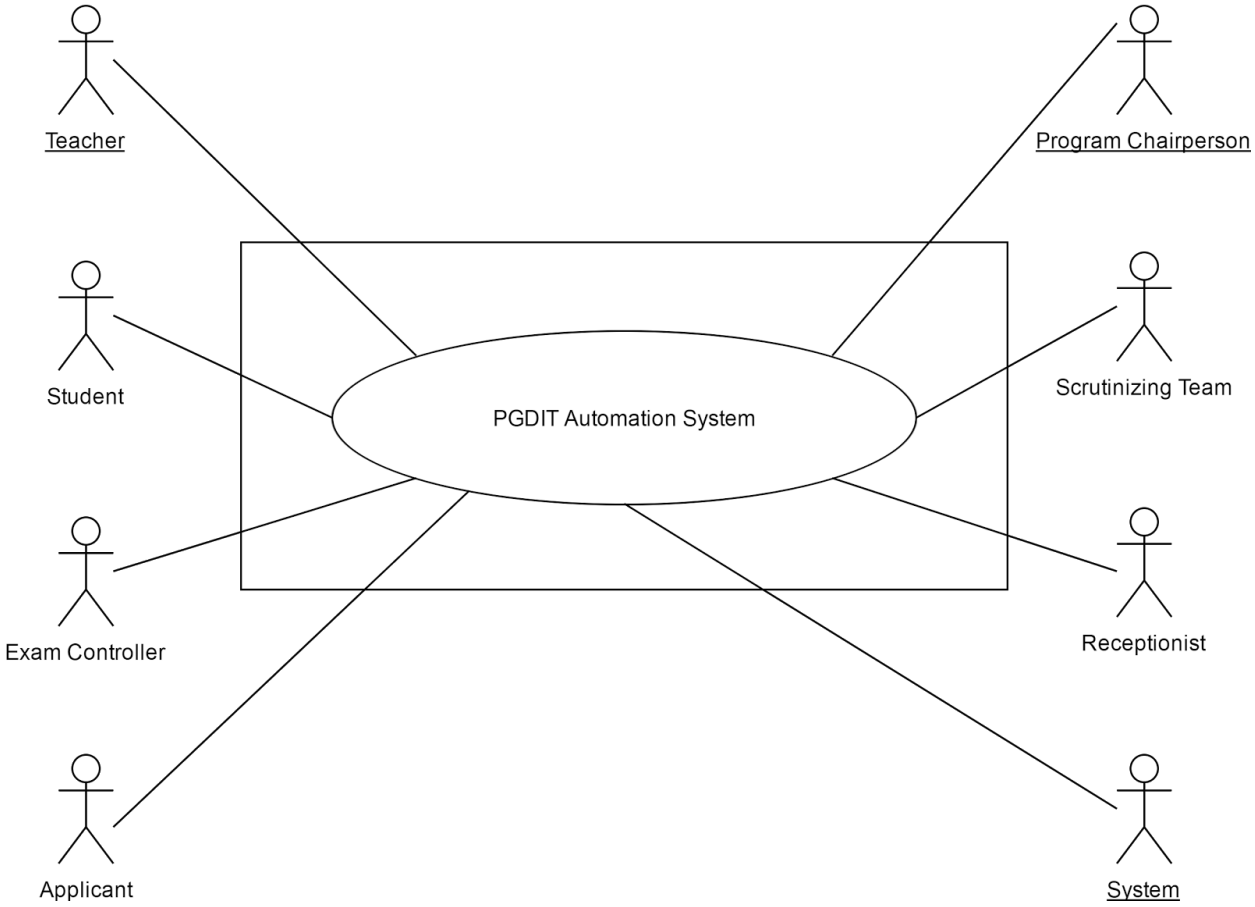


Figure 4.2.0: Level-0 Use case diagram

- Primary Actors: System, Teacher, Program Chairperson.
- Secondary Actors: Exam controller, Student, Scrutinizing Team, Applicant, Receptionist.
- It is level - 0 figure of the total system, club management system.

4.2.3 Level-1 Use Case Diagram

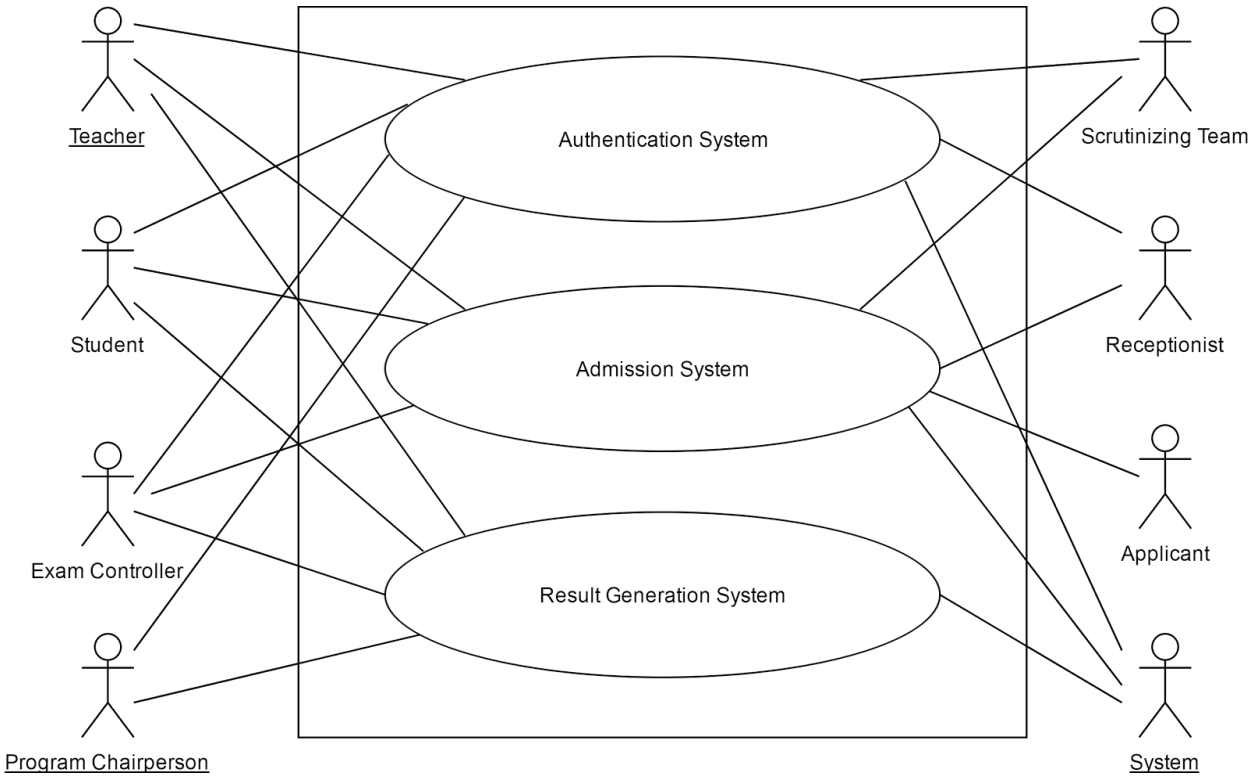


Figure 4.2.3: Level-1 Use case diagram

- Primary Actors: System, Teacher, Program Chairperson.
- Secondary Actors: Exam controller, Student, Scrutinizing Team, Applicant, Receptionist.
- It is level - 1 figure derived from Level-0 figure.
- In this figure, total subsystem is divided into three subsystems. They are: Authentication system, Admission System, Result Generation System.
- Actors related with corresponding subsystems are shown in this figure.

4.2.3.1 Authentication System

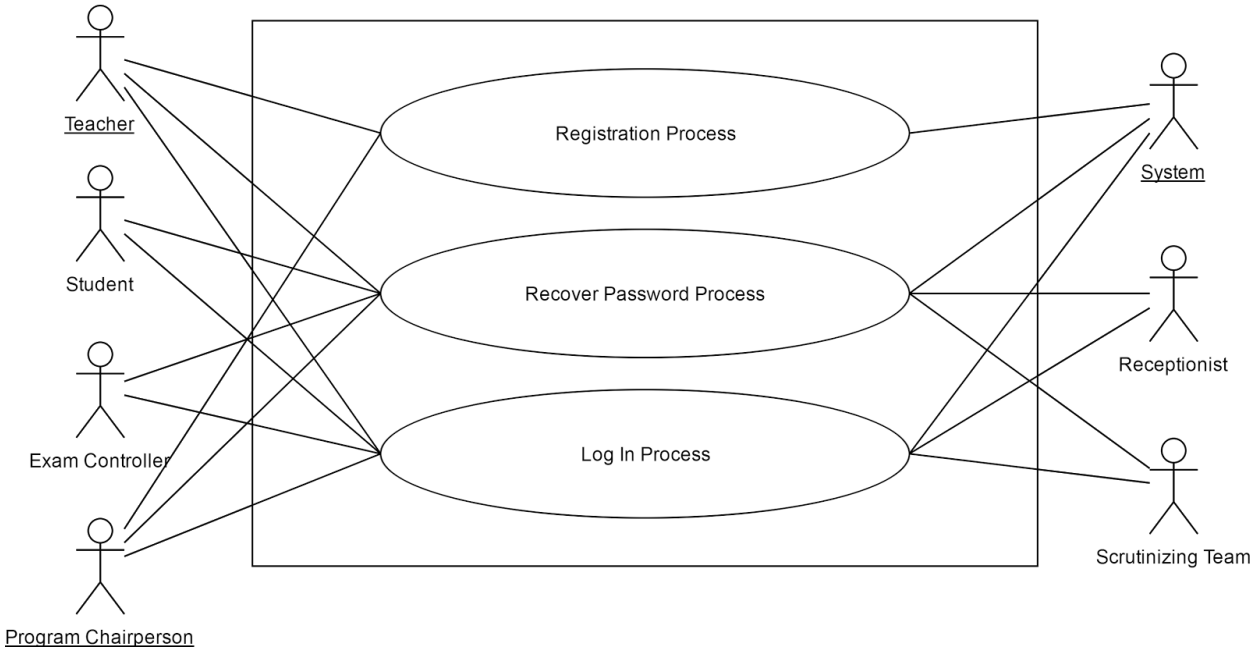


Figure 4.2.3.1: Level-1.1 Use case diagram

- Primary Actors: Teacher, System, Program Chairperson
- Secondary Actors: Student, Exam Controller, Receptionist, Scrutinizing Team
- This diagram is level 1.1 of the use case model
- This level is derived from level 1

Description:

This level represents the Authentication subsystem. The action-reply sequence of the use case is given below:

Action-Reply Sequence:

Exam Controller:

Action: Choose "Sign-in" option.
Reply: Display interface for providing username and password.
Action: Choose "Recover Password" option.
Reply: Send system generated recovery code.
Action: Provide recovery code.
Reply: Display password reset page.

Receptionist:

Action: Choose "Sign-in" option.
Reply: Display interface for providing username and password.
Action: Choose "Recover Password" option.
Reply: Send system generated recovery code.
Action: Provide recovery code.
Reply: Display password reset page.

Scrutinizing Team:

Action: Choose "Sign-in" option.

Reply: Display interface for providing username and password.
Action: Choose "Recover Password" option.
Reply: Send system generated recovery code.
Action: Provide recovery code.
Reply: Display password reset page.

Teacher:

Action: Choose "Sign-up" option.
Reply: Display interface for providing information.
Action: Provide information.
Reply: Send message about waiting for validation.
Action: Choose "Sign-in" option.
Reply: Display interface for providing username and password.
Action: Choose "Recover Password" option.
Reply: Send system generated recovery code.
Action: Provide recovery code.
Reply: Display password reset page.

Program Chairperson:

Action: Choose "Verify new Teacher" option.
Reply: Display Teacher information.
Action: Choose "Sign-in" option.
Reply: Display interface for providing username and password.
Action: Choose "Recover Password" option.
Reply: Send system generated recovery code.
Action: Provide recovery code.
Reply: Display password reset page.
Action: Choose "Add students' account" option.
Reply: Display message about adding students' accounts.

Student:

Action: Choose "Sign-in" option.
Reply: Display interface for providing username and password.
Action: Choose "Recover Password" option.
Reply: Send system generated recovery code.
Action: Provide recovery code.
Reply: Display password reset page.

4.2.3.2 Admission System

Figure 4.2.3.2: Level-1.2 Use case diagram

- Primary Actors: System, Program Chairperson
- Secondary Actors: Applicant, Teacher, Exam Controller, Receptionist, Scrutinizing Team

- This diagram is level 1.2 of the use case model
- This level is derived from level 1

Description:

This level represents the Admission subsystem. This subsystem has been further split into 8 subsystems. The action-reply sequence of the use case is given below:

Action-Reply Sequence:

Exam Controller:

Action: Choose "Generate Seat Plan" option.
Reply: Generate and display seat plan.
Action: Choose "Upload written marks" option.
Reply: Display interface for providing written marks.
Action: Choose "Upload VIVA marks" option.
Reply: Display interface for providing VIVA marks.

Receptionist:

Action: Choose "Verify Payment" option.
Reply: Display interface for verifying payment.

Scrutinizing Team:

Action: Choose "Download Application Form" option.
Reply: Display interface for downloading application form.
Action: Choose "Verify Applicant" option.
Reply: Display interface for verifying applicant.

Teacher:

Action: Choose "View written exam shift" option.
Reply: Display written exam shift.
Action: Choose "View VIVA shift" option.
Reply: Display VIVA shift.

Program Chairperson:

Action: Choose "Enable Application Form" option.
Reply: Display message about enabling application form.
Action: Choose "Disable Application Form" option.
Reply: Display message about disabling application form.

Applicant:

Action: Choose "Authenticate" option.
Reply: Display interface for providing application ID and Birthdate.

4.2.3.3 Result Generation System

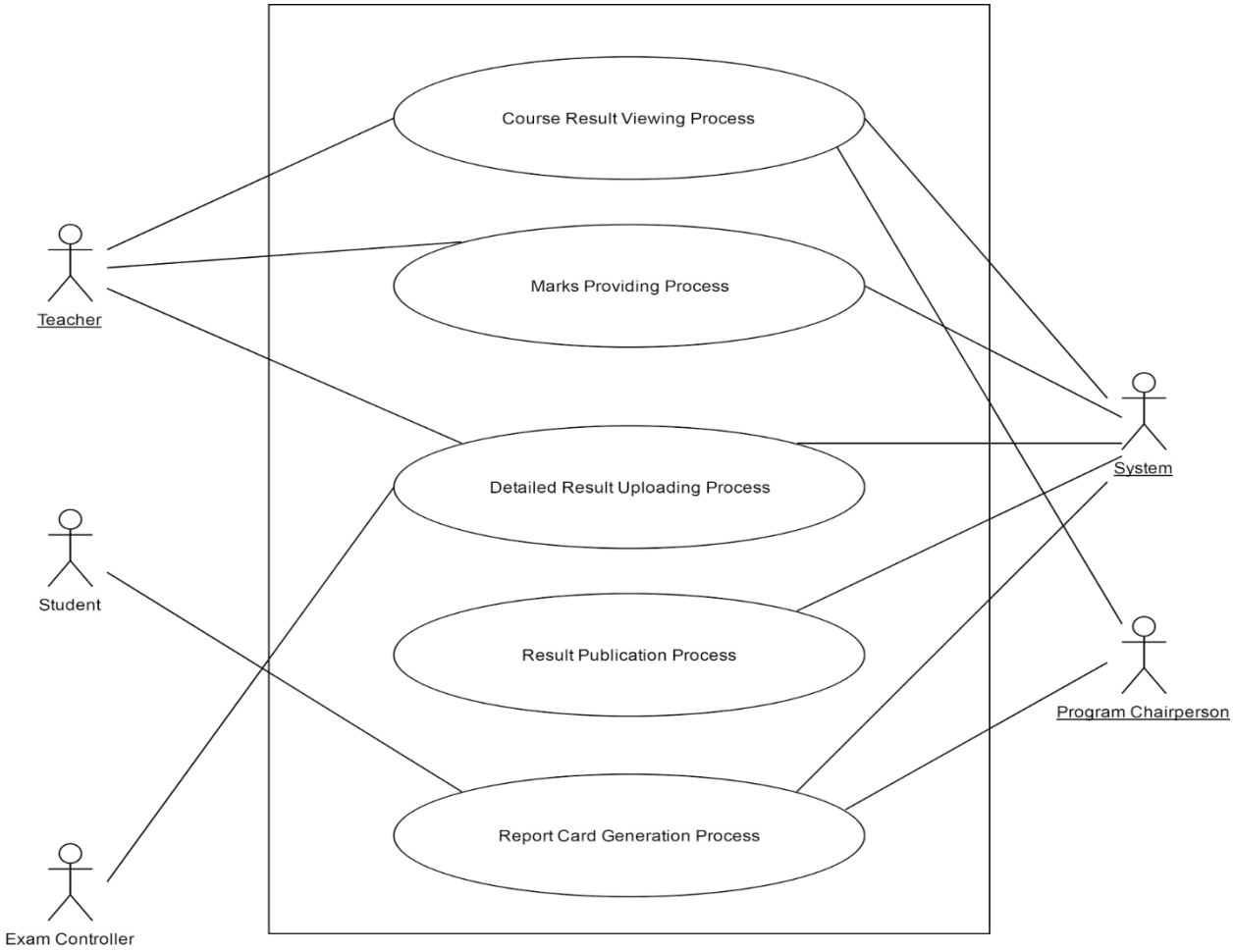


Figure 4.2.3.3: Level-1.3 Use case diagram

- Primary Actors: System, Program Chairperson, Teacher
- Secondary Actors: Student, Exam Controller
- This diagram is level 1.3 of the use case model
- This level is derived from level 1

Description:

This level represents the Result Generation subsystem. This subsystem has been further divided into 5 subsystem. The action-reply sequence of the use case is given below:

Action-Reply Sequence:

Exam Controller:

Action: Choose "Download Result" option.
Reply: Display interface for downloading detailed result.

Teacher:

Action: Choose "View course result" option.
Reply: Display teacher's course result.
Action: Choose "Provide marks" option.
Reply: Display interface for providing detailed result.

Action: Choose “Upload detailed marks” option.
Reply: Display interface for uploading detailed result.

Program Chairperson:

Action: Choose “Enable Course Result View” option.
Reply: Display message about enabling course result view.
Action: Choose “Enable Report Card Generation” option.
Reply: Display message about enabling report card generation.

Student:

Action: Choose “View Report Card” option.
Reply: Display report card.

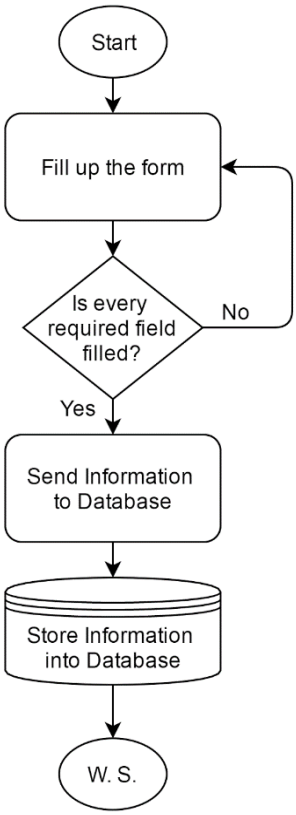
4.3 Activity Diagram

4.3.1 Authentication System

4.3.1.1 Registration Process

Refers Use Case figure 4.3.1.1:

4.3.1.1.(a) Registration Process (Teachers)



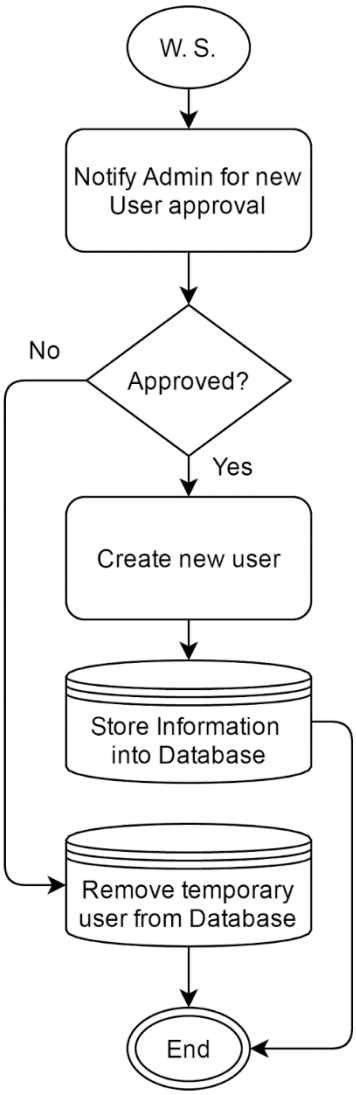


Figure 4.3.1.1.A: Activity diagram for Teacher's Registration Process in Authentication System

4.3.1.1.(b) Registration Process (Students)

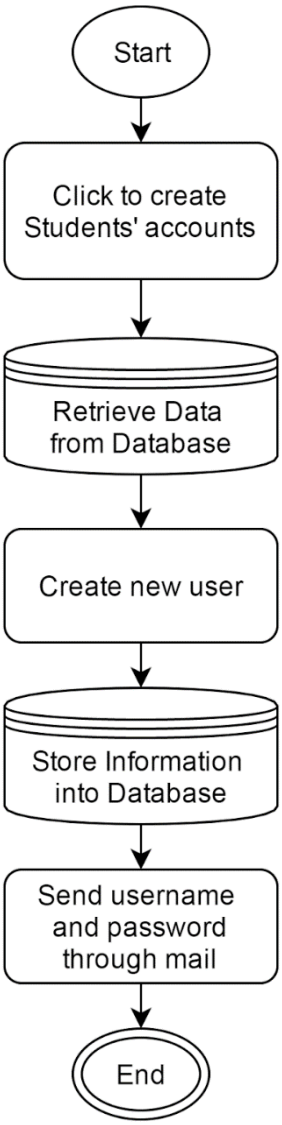


Figure 4.3.1.1.B: Activity diagram for Student's Registration Process in Authentication System

4.3.1.2 Login Process

Refers Use Case figure 4.2.3.2:

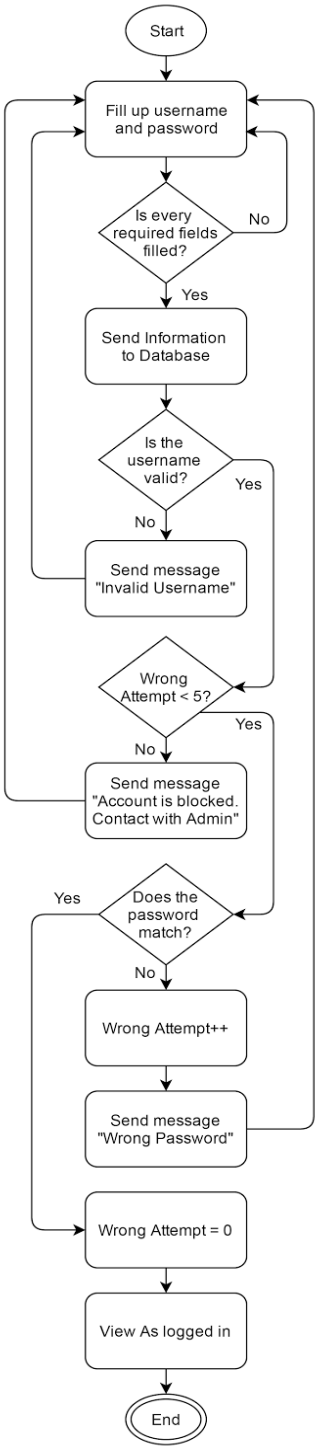
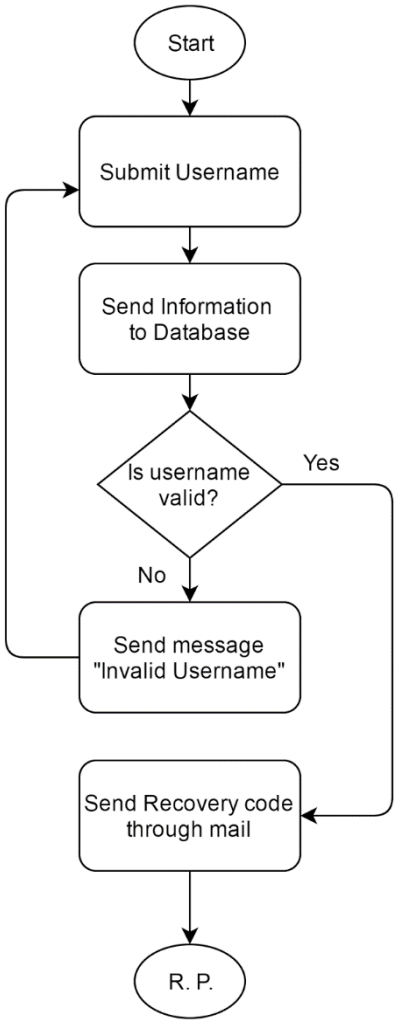


Figure 4.3.1.2: Activity diagram for Login Process in Authentication System

4.3.1.3 Password Recovery Process

Refers Use Case figure 4.2.3.3:



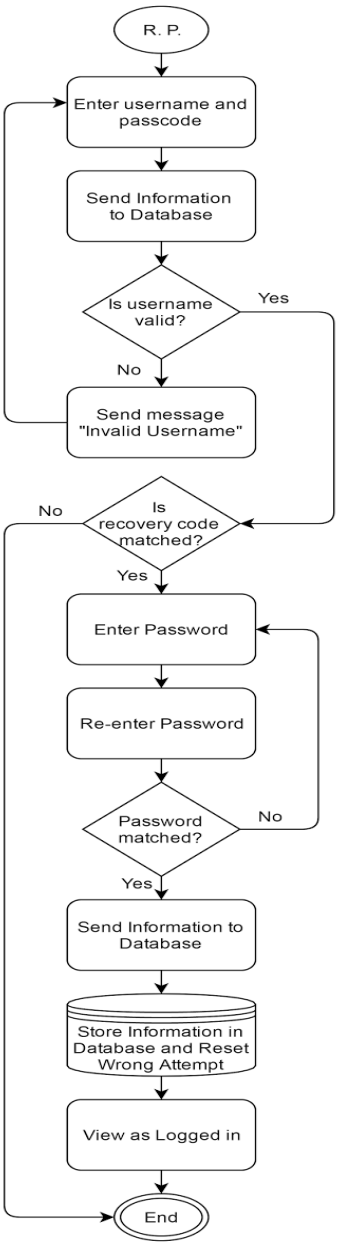


Figure 4.3.1.3: Activity diagram for Password Recovery Process in Authentication System

4.3.2 Admission System

Refers Use Case figure 4.2.3.2:
The activities of admission system are divided into the following processes:

4.3.2.1 Enable Application Form

Refers Use Case figure 4.2.3.2:

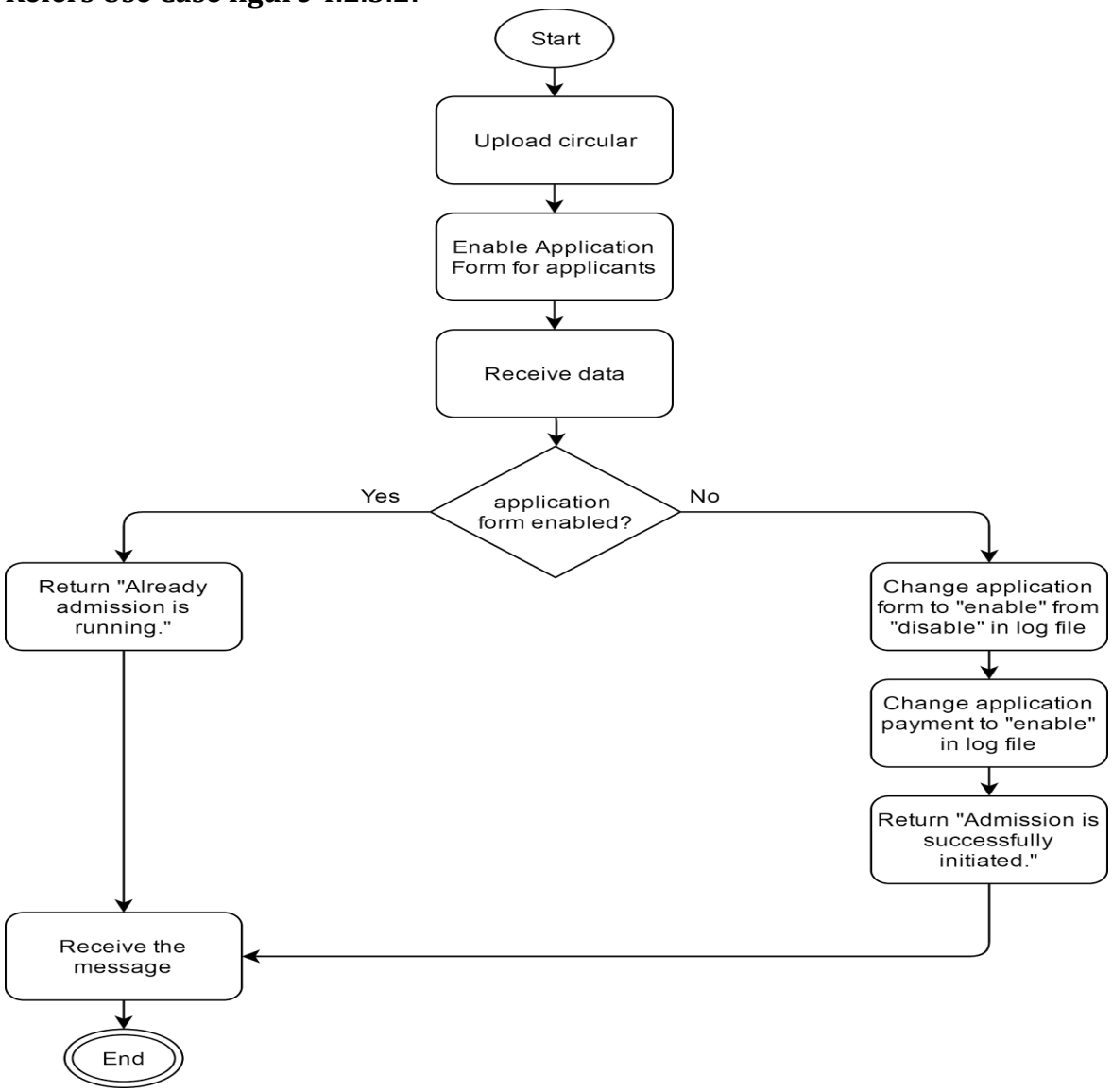


Figure 4.3.2.1: Activity diagram for Enabling Application Form in Admission System

4.3.2.2 Application Process

Refers Use Case figure 4.2.3.2:

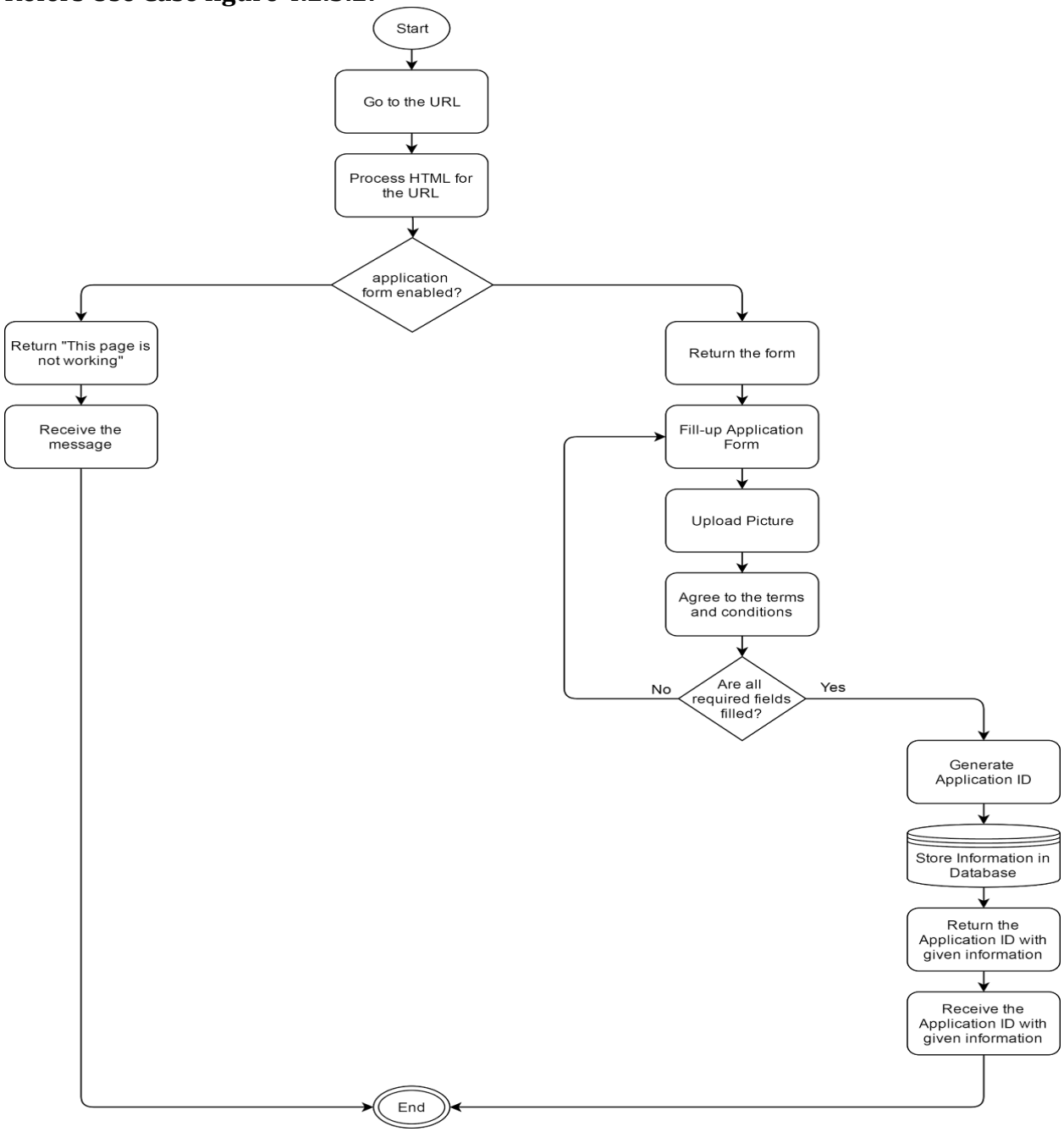


Figure 4.3.2.2: Activity diagram for Application process in Admission System

4.3.2.3 Disable Application Payment Process

Refers Use Case figure 4.2.3.2:

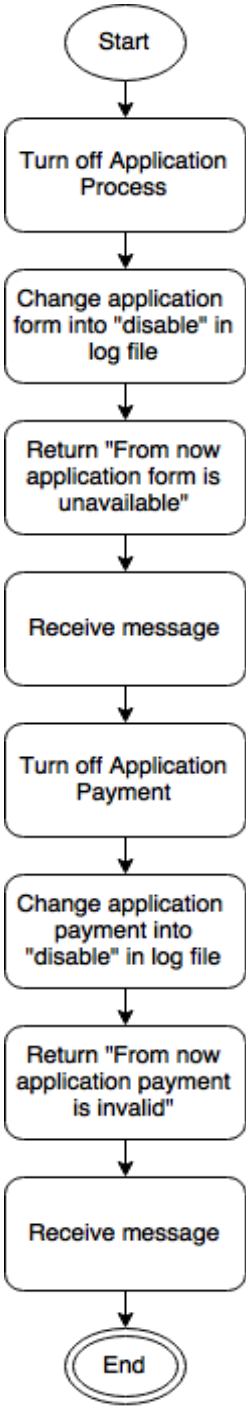


Figure 4.3.2.3: Activity diagram for disabling application payment process in Admission System

4.3.2.4 Scrutinizing Process

Refers Use Case figure 4.2.3.2:

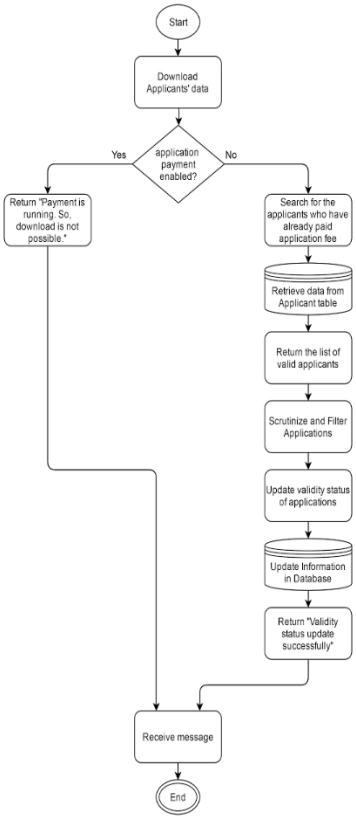


Figure 4.3.2.4: Activity diagram for scrutinizing process in Admission System

4.3.2.5 Seat Plan Generation Process

Refers Use Case figure 4.2.3.2:

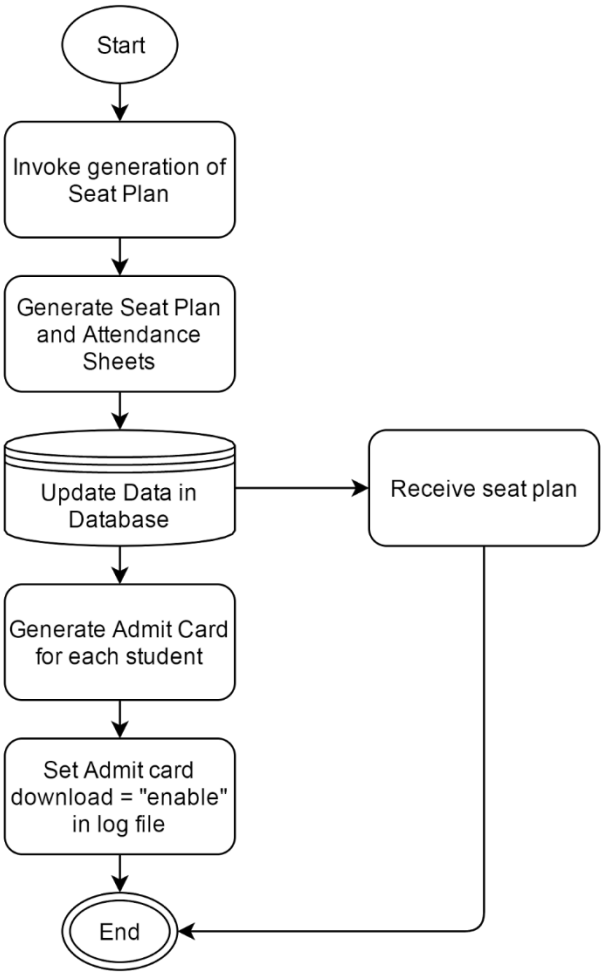


Figure 4.3.2.5: Activity diagram for seat plan generation process in Admission System

4.3.2.6 Viva Plan and Result Generation Process

Refers Use Case figure 4.2.3.2:

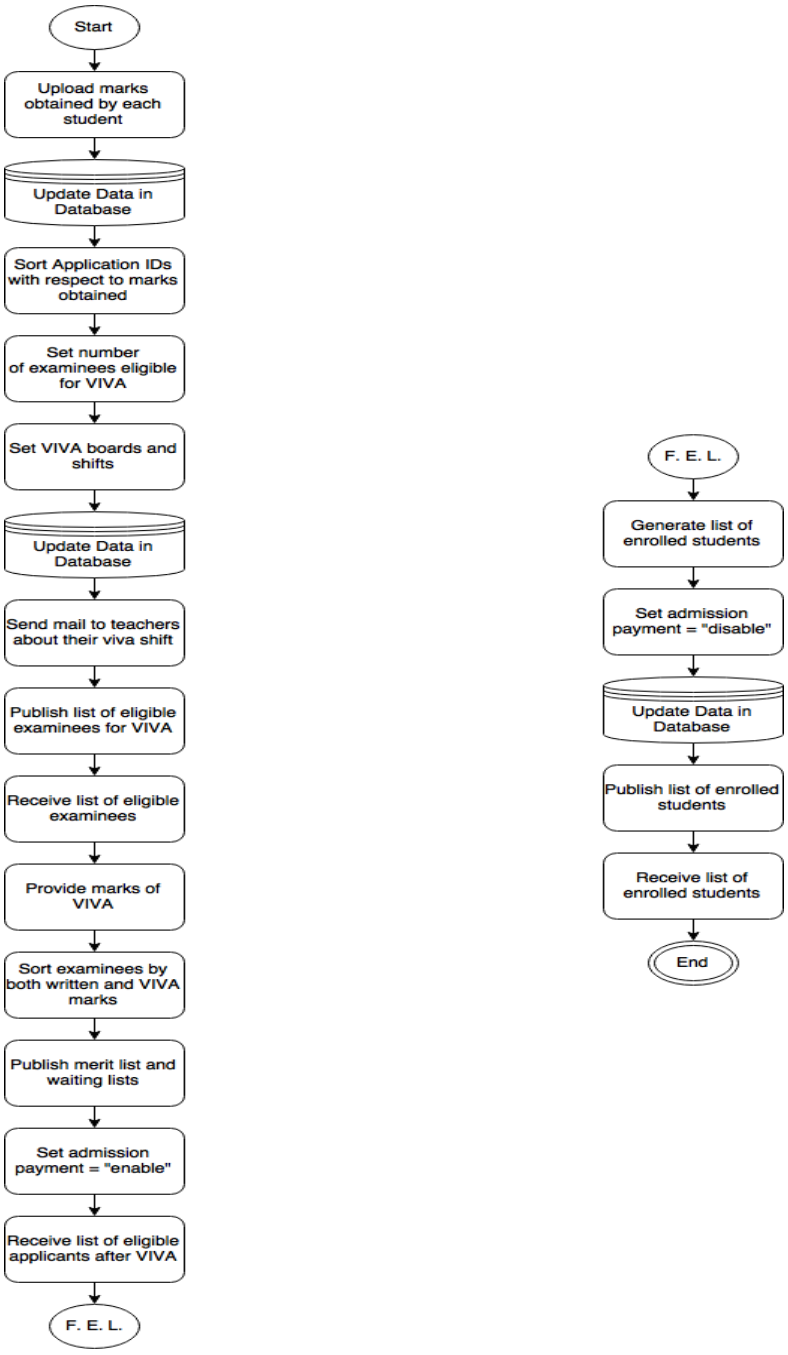


Figure 4.3.2.6: Activity diagram for VIVA plan and result generation process in Admission System

4.3.2.7.A Application Payment Procedure

Refers Use Case figure 4.2.3.2:

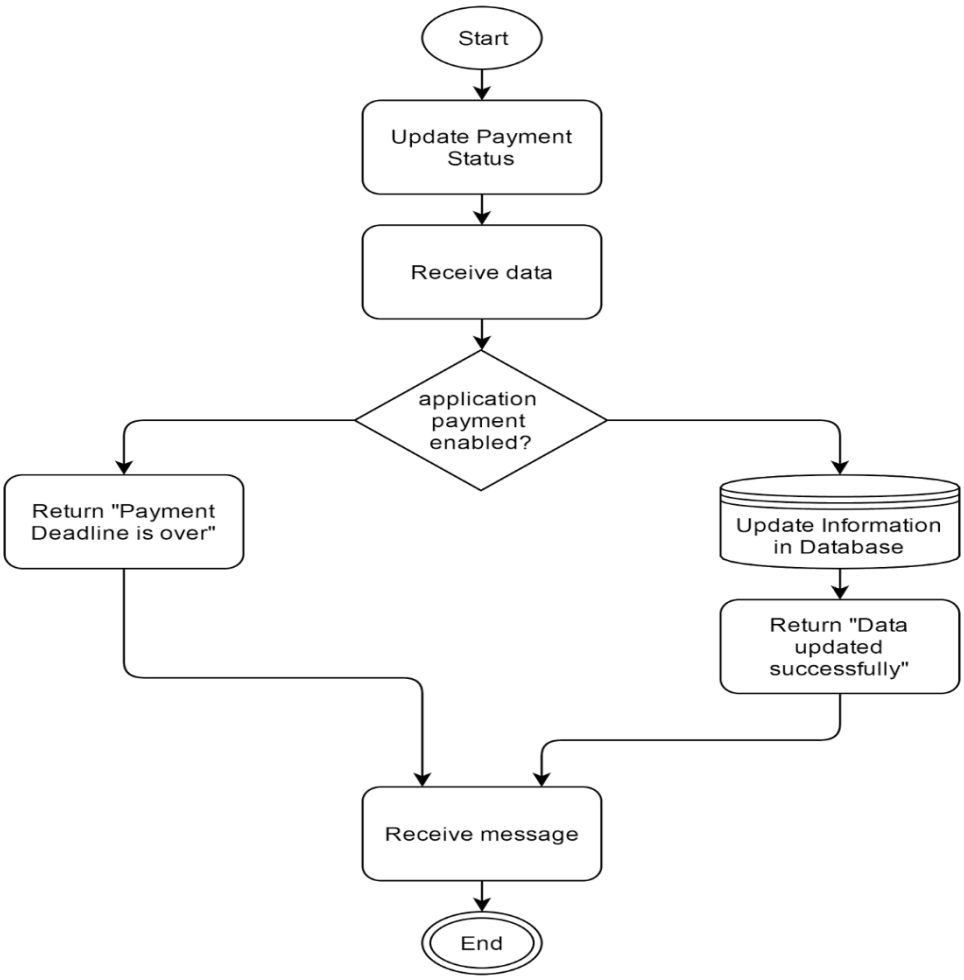


Figure 4.3.2.7.A: Activity diagram for Admission Payment Procedure in Admission System

4.3.2.7.B Admission Payment Procedure

Refers Use Case figure 4.2.3.2:

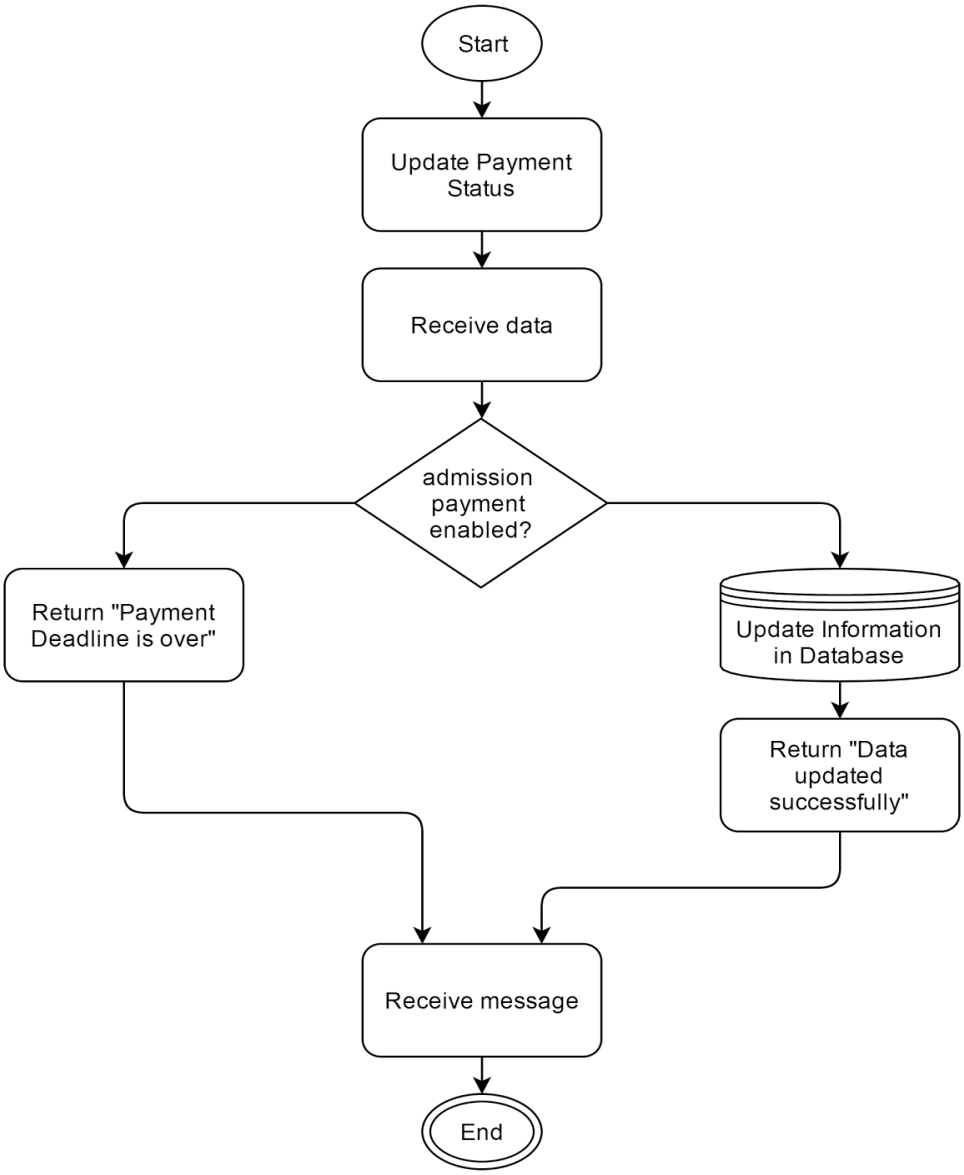


Figure 4.3.2.7.A: Activity diagram for Application Payment Procedure in Admission System

4.3.3 Result Generation System

Refers Use Case figure 4.2.3.3:
The activities of result generation system are divided into the following processes:

4.3.3.1 Marks Providing Process

Refers Use Case figure 4.2.3.3:

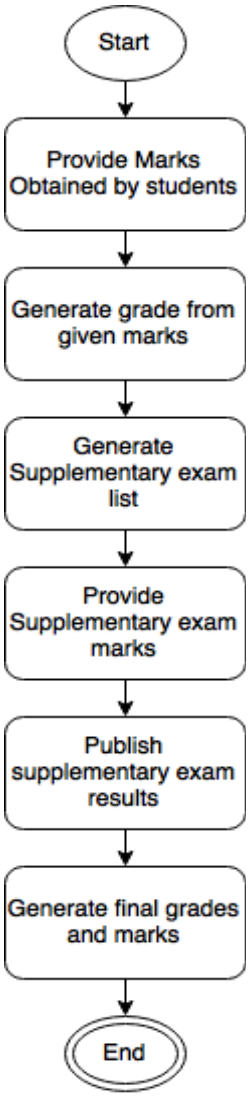


Figure 4.3.3.1: Activity diagram for Marks Providing Process in Result Generation System

4.3.3.2 Detailed Result Uploading Process

Refers Use Case figure 4.2.3.3:

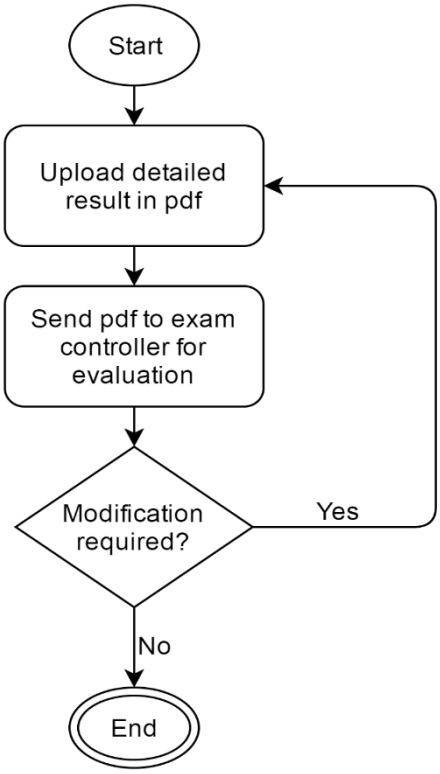


Figure 4.3.3.2: Activity diagram for Detailed Result Uploading Process in Result Generation System

4.3.3.3 Result Publication Process

Refers Use Case figure 4.2.3.3:

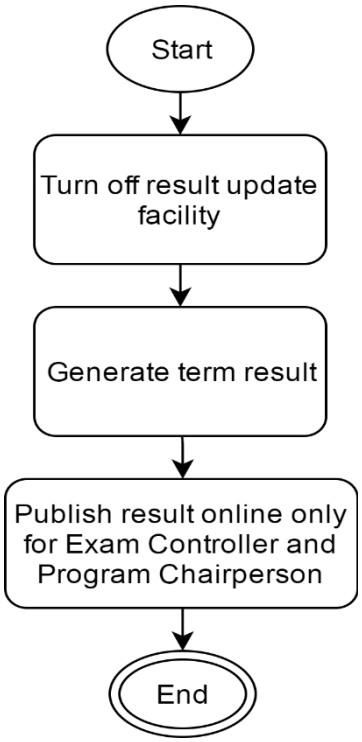


Figure 4.3.3.3: Activity diagram for Result Publication Process in Result Generation System

4.3.3.4 Report Card Generation Process

Refers Use Case figure 4.2.3.3:

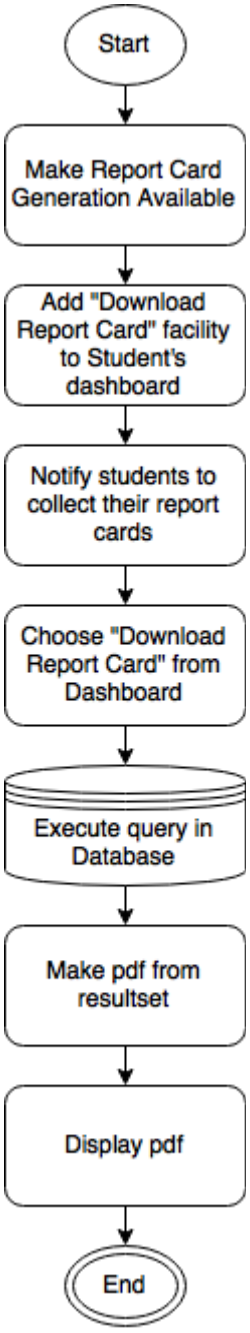


Figure 4.3.3.4: Activity diagram for Report Card Generation Process in Result Generation System

4.3.3.5 Course Result Viewing Process

Refers Use Case figure 4.2.3.3:

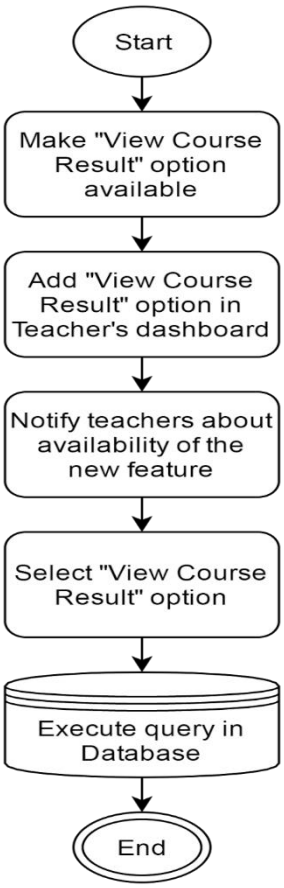


Figure 4.3.3.5: Activity diagram for Course Result Viewing Process in Result Generation System

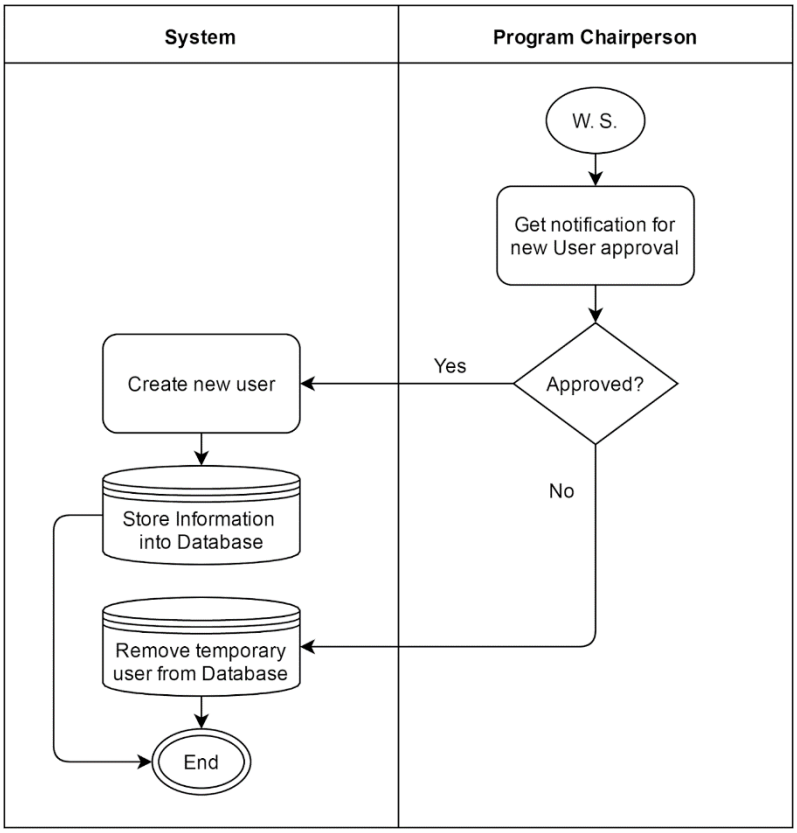
4.4 Swimlane Diagram

4.4.1 Authentication System

4.4.1.1 Registration Process

Refers Use Case figure 4.2.3.1:

4.4.1.1.(a) Registration Process (Teachers)



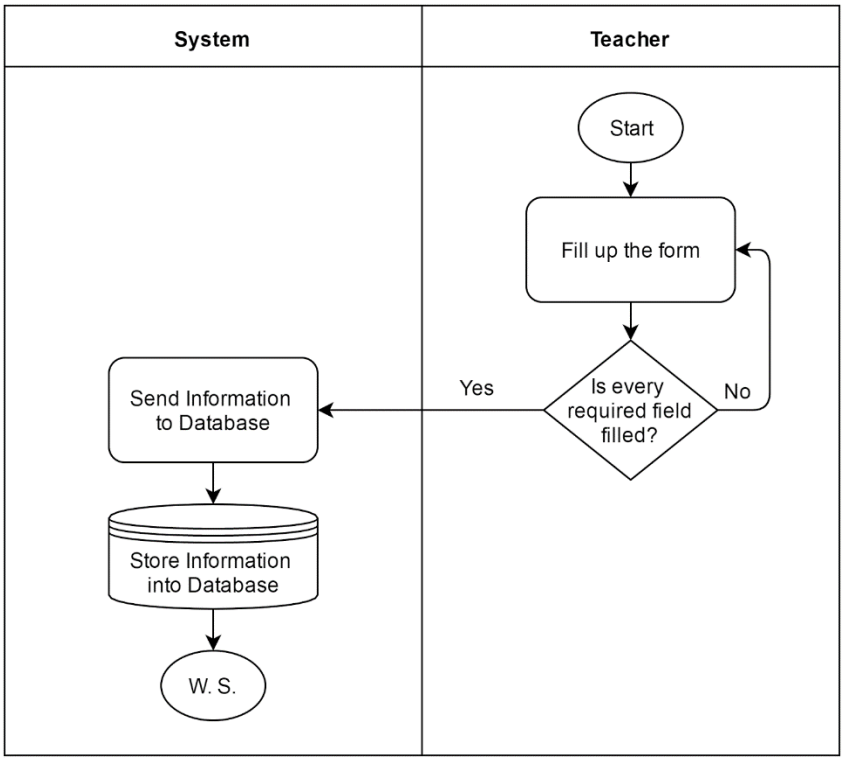


Figure 4.4.1.1.A: Swimlane diagram for Teacher's Registration Process in Authentication System

4.4.1.1.(b) Registration Process (Students)

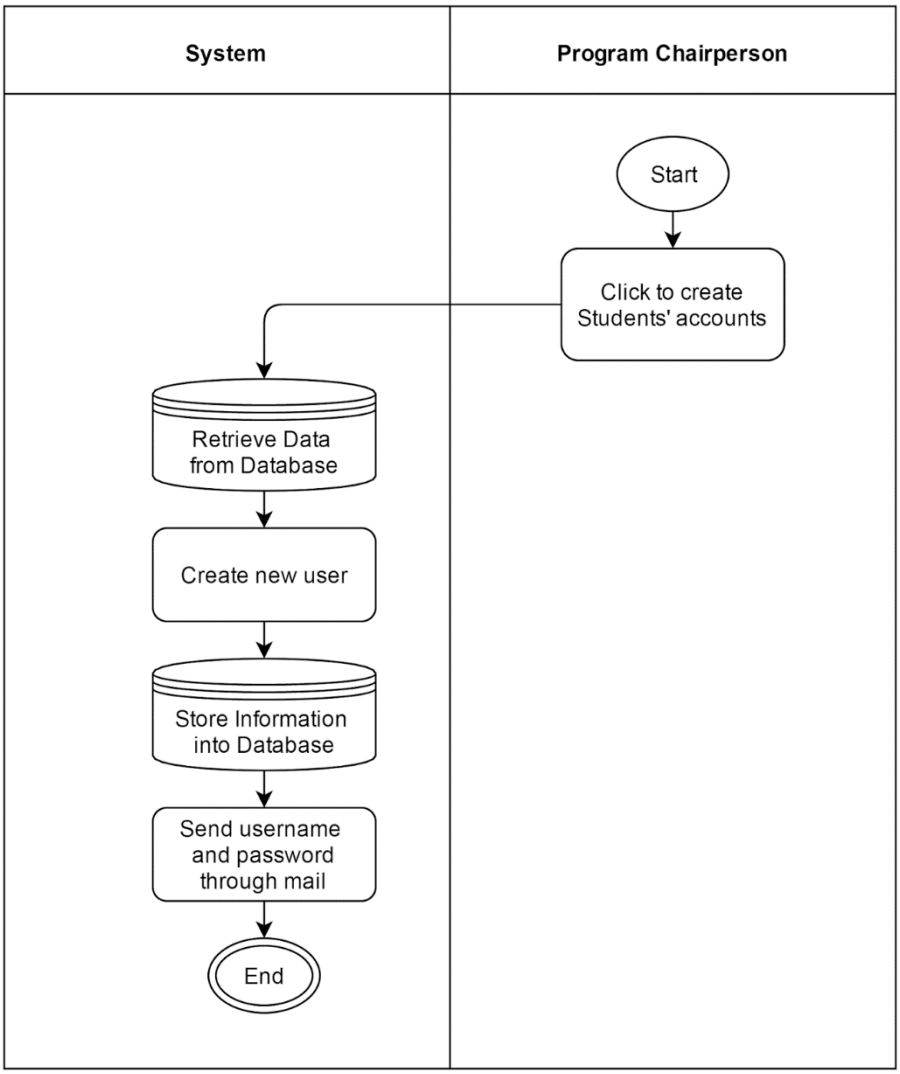


Figure 4.4.1.1.B: Swimlane diagram for Student's Registration Process in Authentication System

4.4.1.2 Login Process

Refers Use Case figure 4.2.3.2:

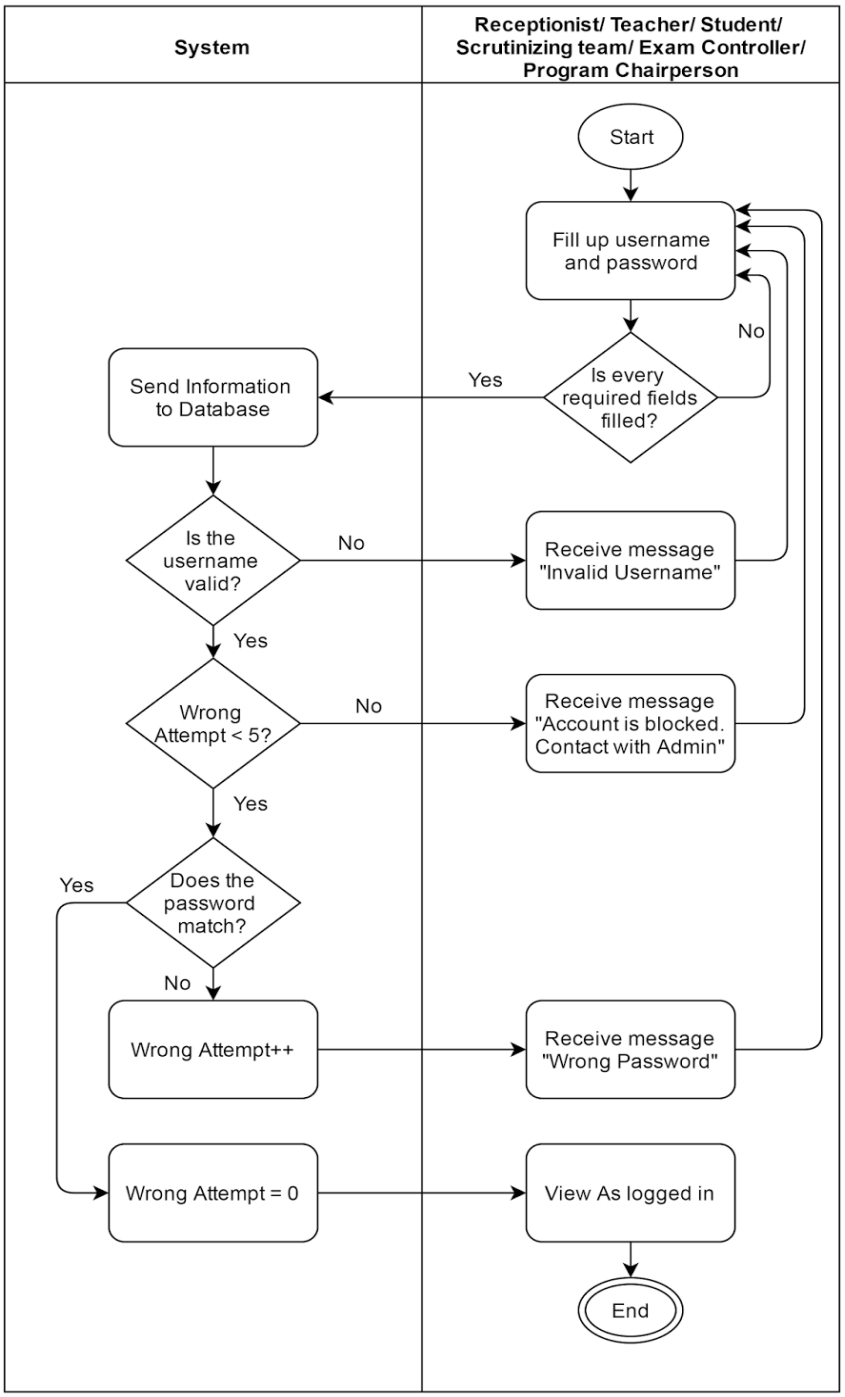
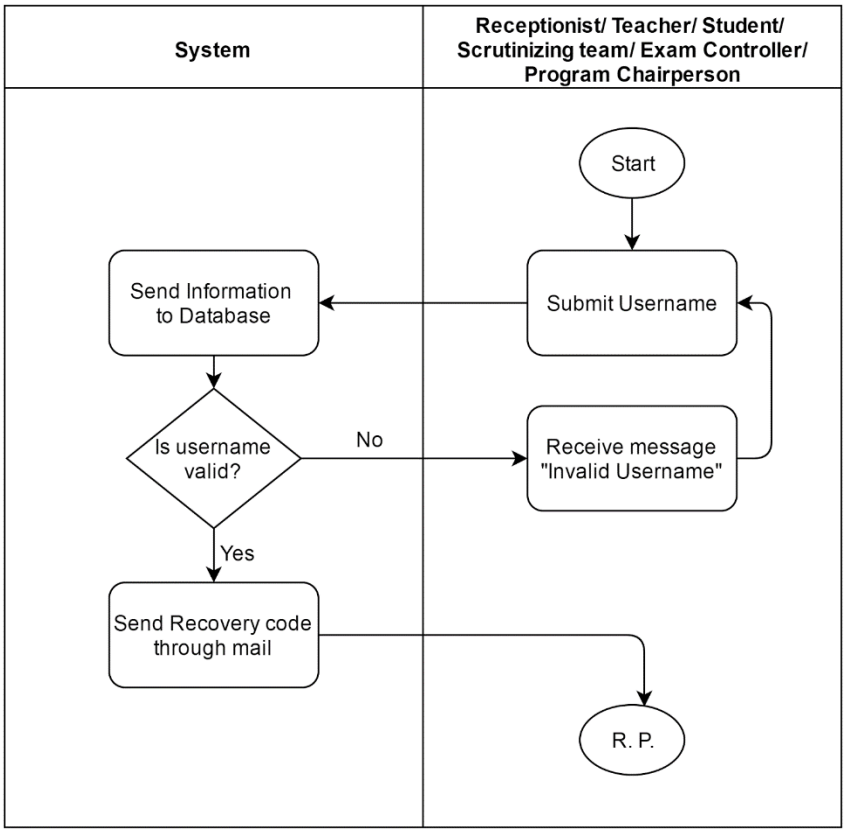


Figure 4.4.1.2: Swimlane diagram for Login Process in Authentication System

4.4.1.3 Password Recovery Process

Refers Use Case figure 4.2.3.3:



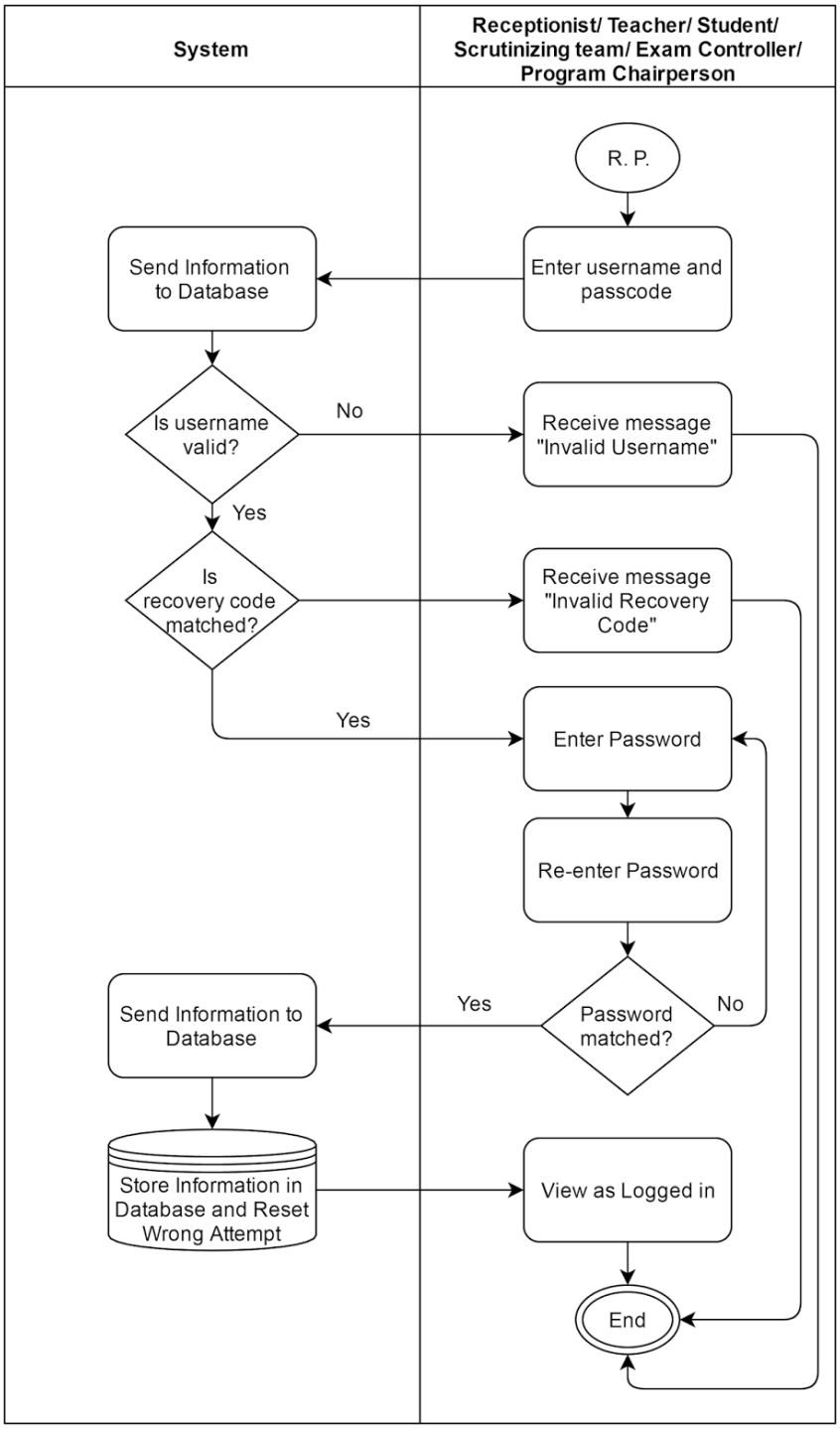


Figure 4.4.1.3: Swimlane diagram for Password Recovery Process in Authentication System

4.4.2 Admission System

4.4.2.1 Enable Application Form

Refers Use Case figure 4.2.3.2:

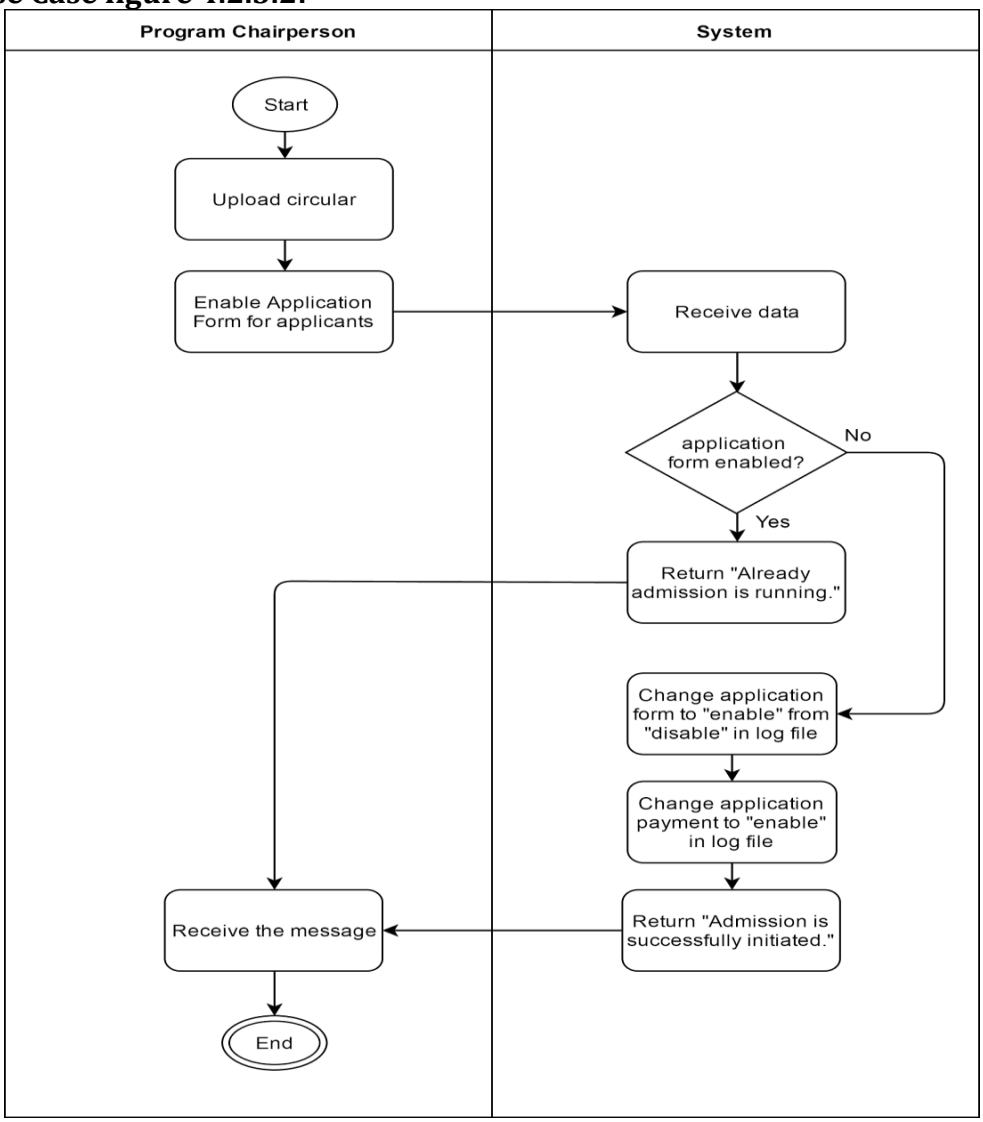


Figure 4.4.2.1: Swimlane diagram for Enabling Application Form in Admission System

4.4.2.2 Application Process

Refers Use Case figure 4.2.3.2:

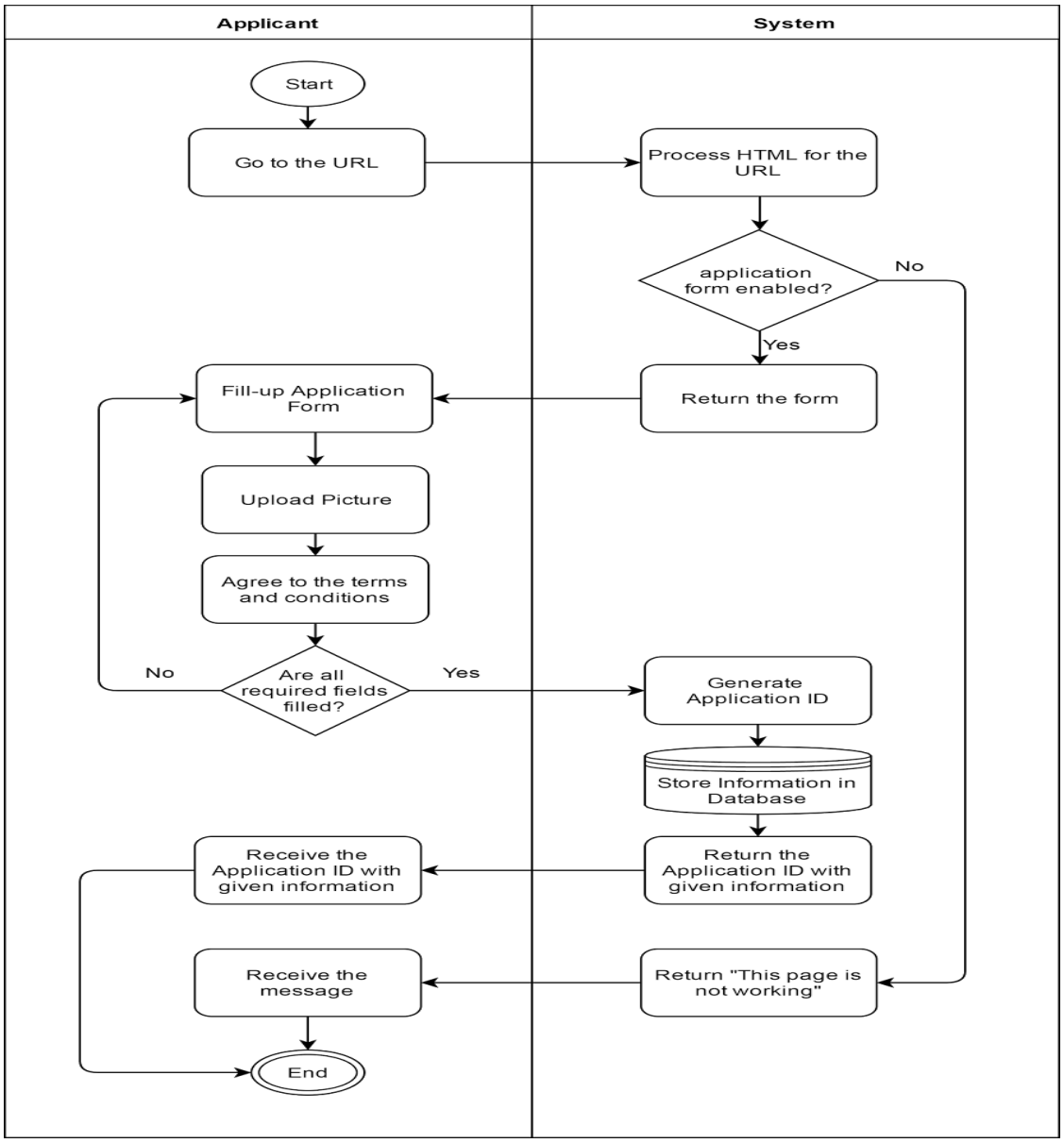


Figure 4.4.2.2: Swimlane diagram for Application Process in Admission System

4.4.2.3 Disable Application Payment Process

Refers Use Case figure 4.2.3.2:

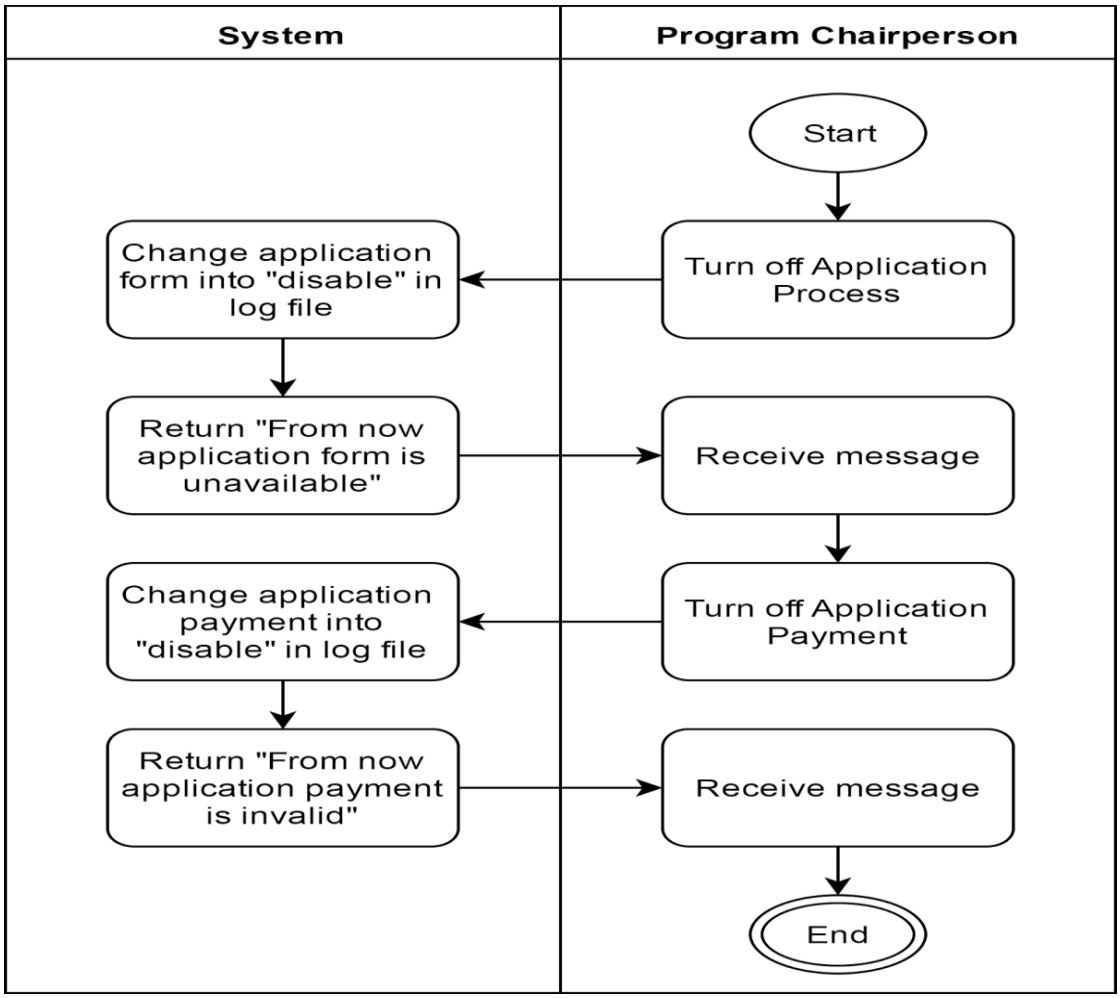


Figure 4.4.2.3: Swimlane diagram for Disabling Application Payment Process in Admission System

4.4.2.4 Scrutinizing Process

Refers Use Case figure 4.2.3.2:

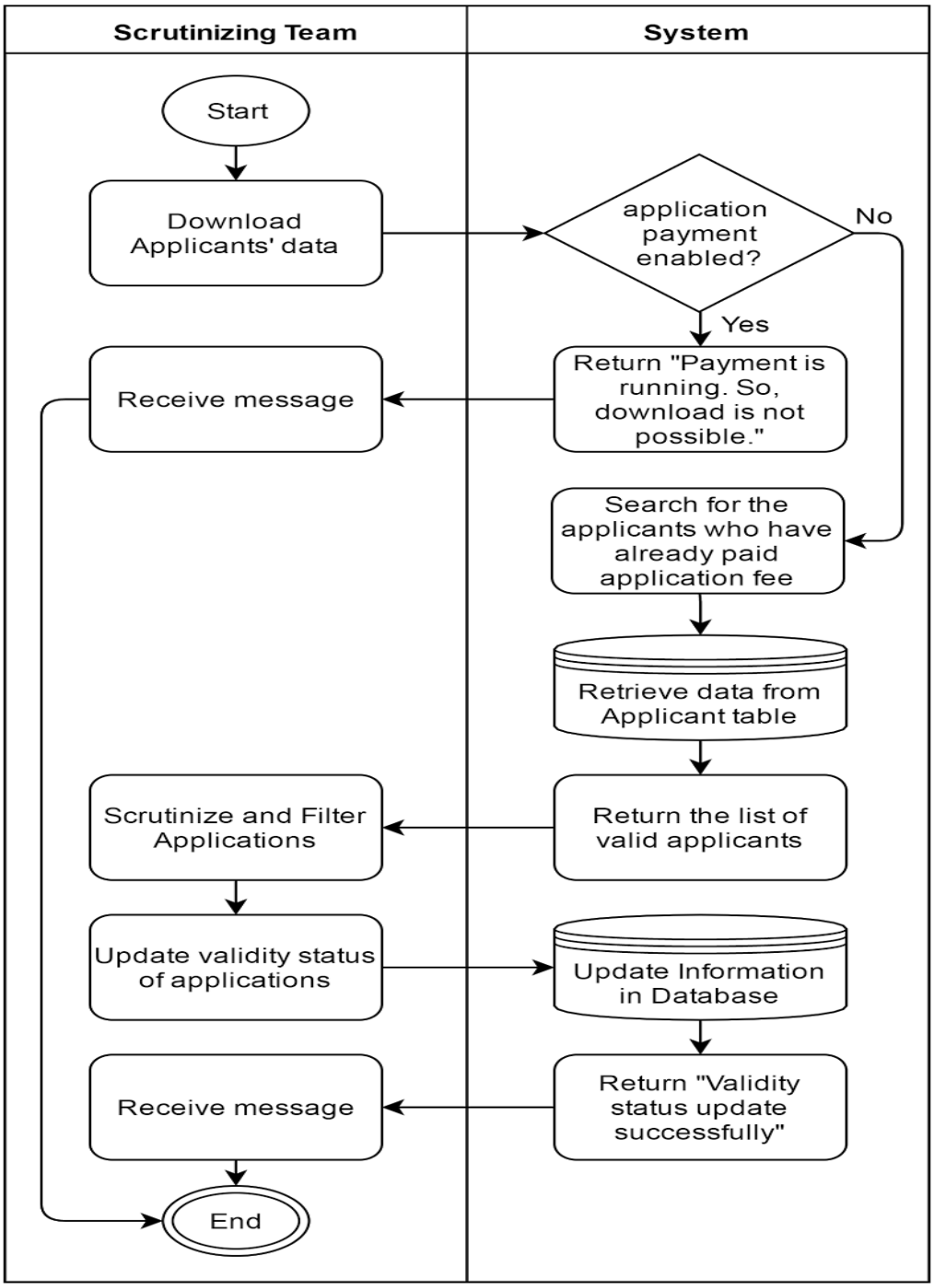


Figure 4.4.2.4: Swimlane diagram for Scrutinizing Process in Admission System

4.4.2.5 Seat Plan Generation Process

Refers Use Case figure 4.2.3.2:

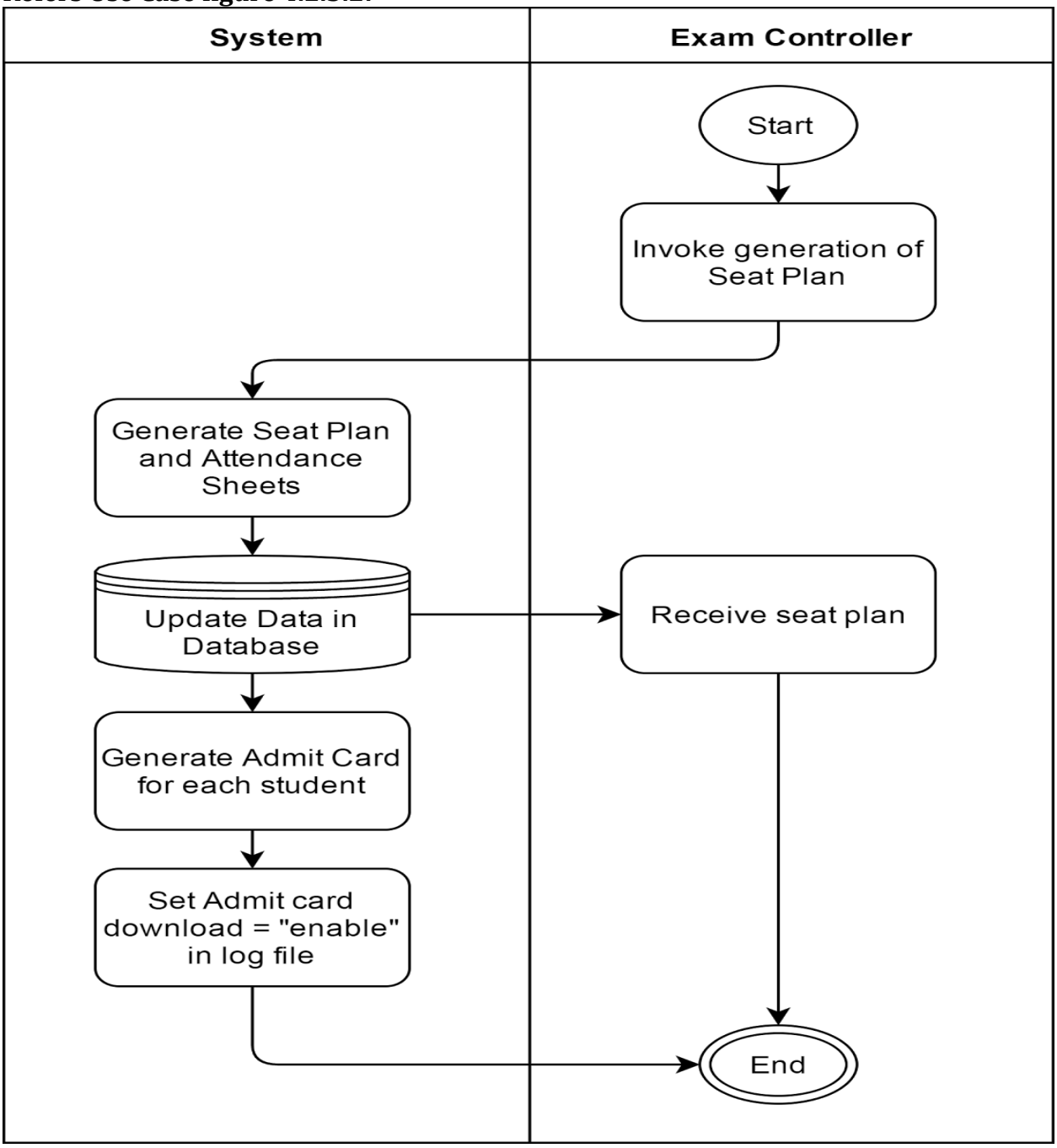


Figure 4.4.2.5: Swimlane diagram for Seat Plan Generation Process in Admission System

4.4.2.6 Viva Plan and Result Generation Process

Refers Use Case figure 4.2.3.2:

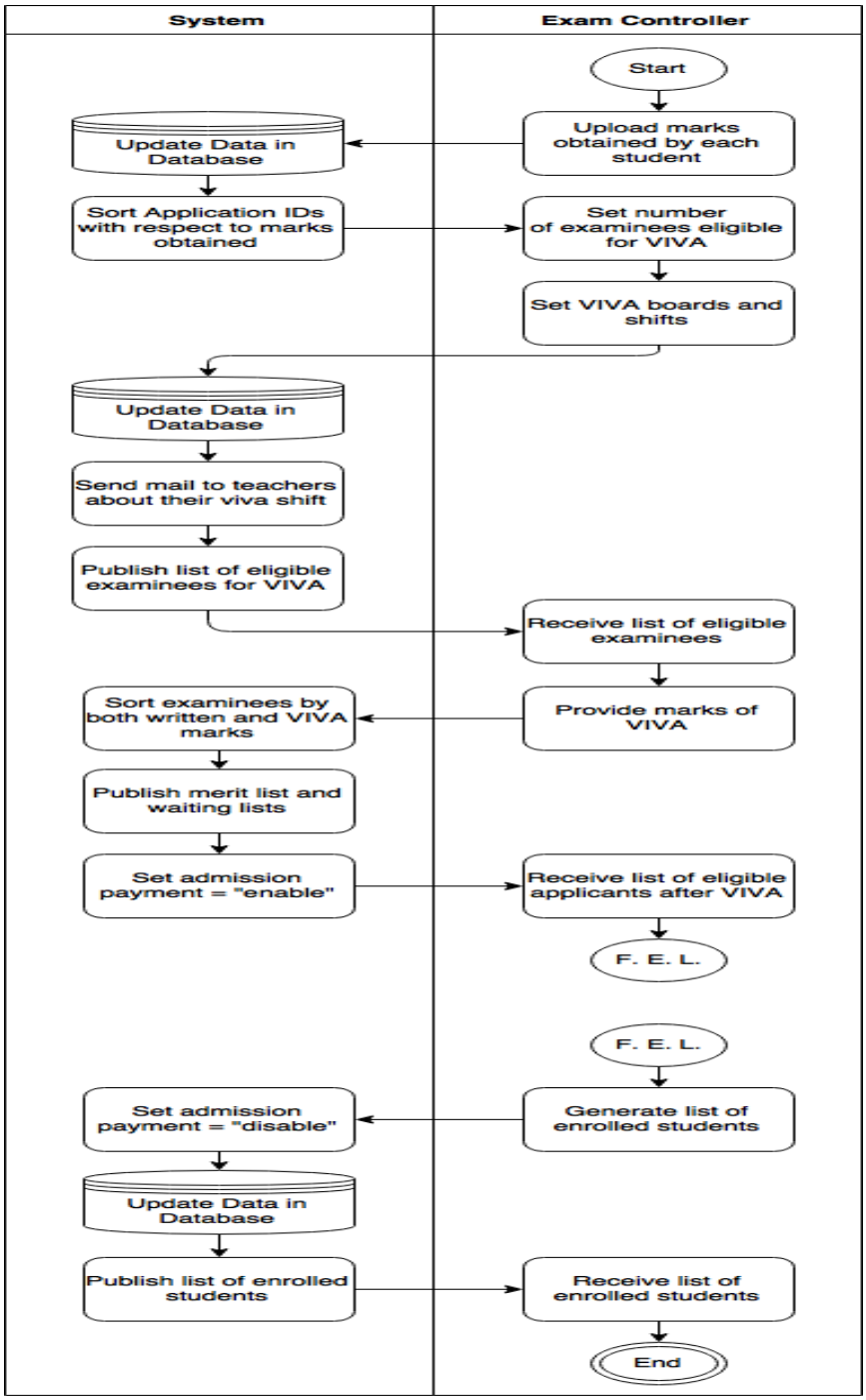


Figure 4.4.2.6: Swimlane diagram for Viva Plan and Result Generation Process in Admission System

4.4.2.7.A Application Payment Procedure

Refers Use Case figure 4.2.3.2:

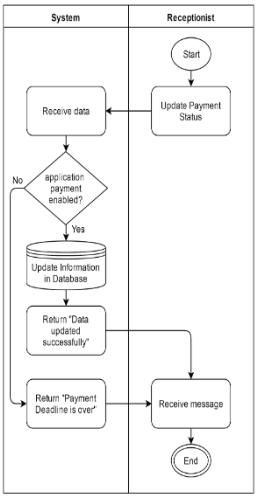


Figure 4.4.2.7.A: Swimlane diagram for Application Payment Procedure in Admission System

4.4.2.7.B Admission Payment Procedure

Refers Use Case figure 4.2.3.2:

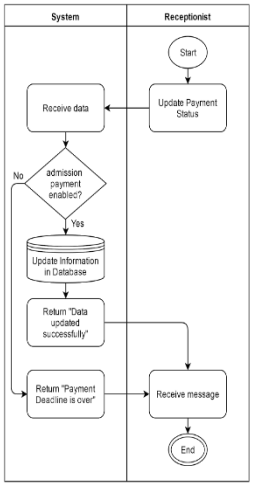


Figure 4.4.2.7.B: Swimlane diagram for Admission Payment Procedure in Admission System

4.4.3 Result Generation System

Refers Use Case figure 4.2.3.3:
The swimlane diagrams of result generation system are divided into the following processes:

4.4.3.1 Marks Providing Process

Refers Use Case figure 4.2.3.2:

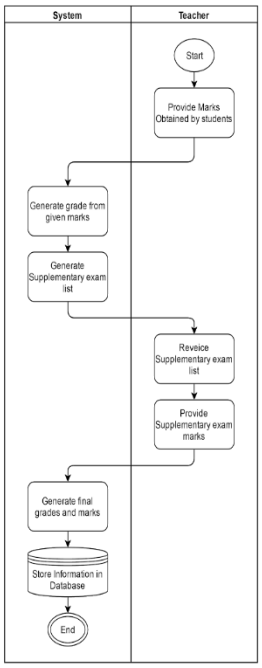


Figure 4.4.3.1: Swimlane diagram for Marks Providing Process in Result Generation System

4.4.3.2 Detailed Result Uploading Process

Refers Use Case figure 4.2.3.2:

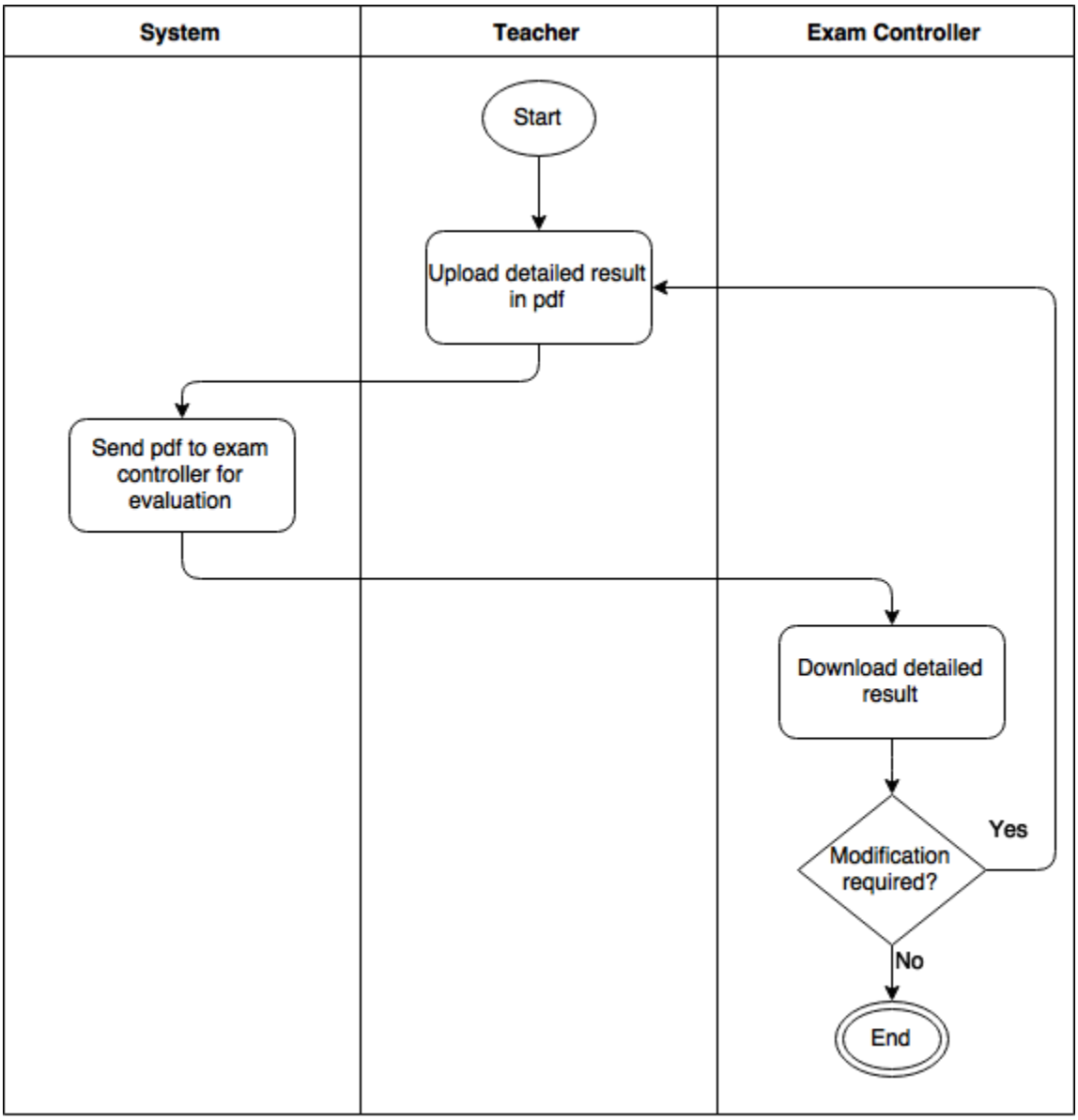


Figure 4.4.3.2: Swimlane diagram for Detailed Result Uploading Process in Result Generation System

4.4.3.3 Result Publication Process

Refers Use Case figure 4.2.3.2:

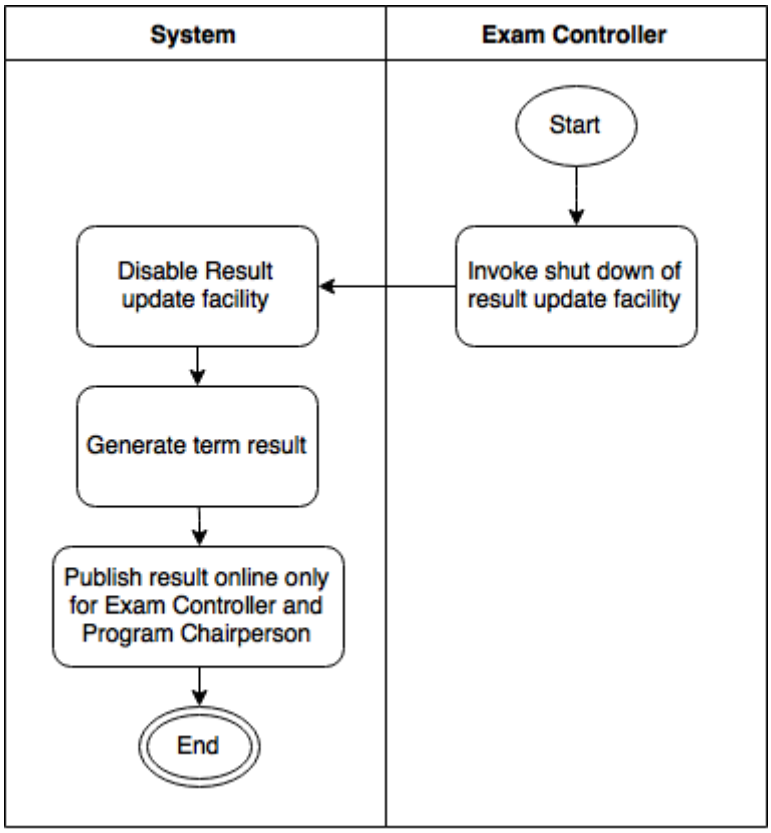


Figure 4.4.3.3: Swimlane diagram for Result Publication Process in Result Generation System

4.4.3.4 Report Card Generation Process

Refers Use Case figure 4.2.3.2:

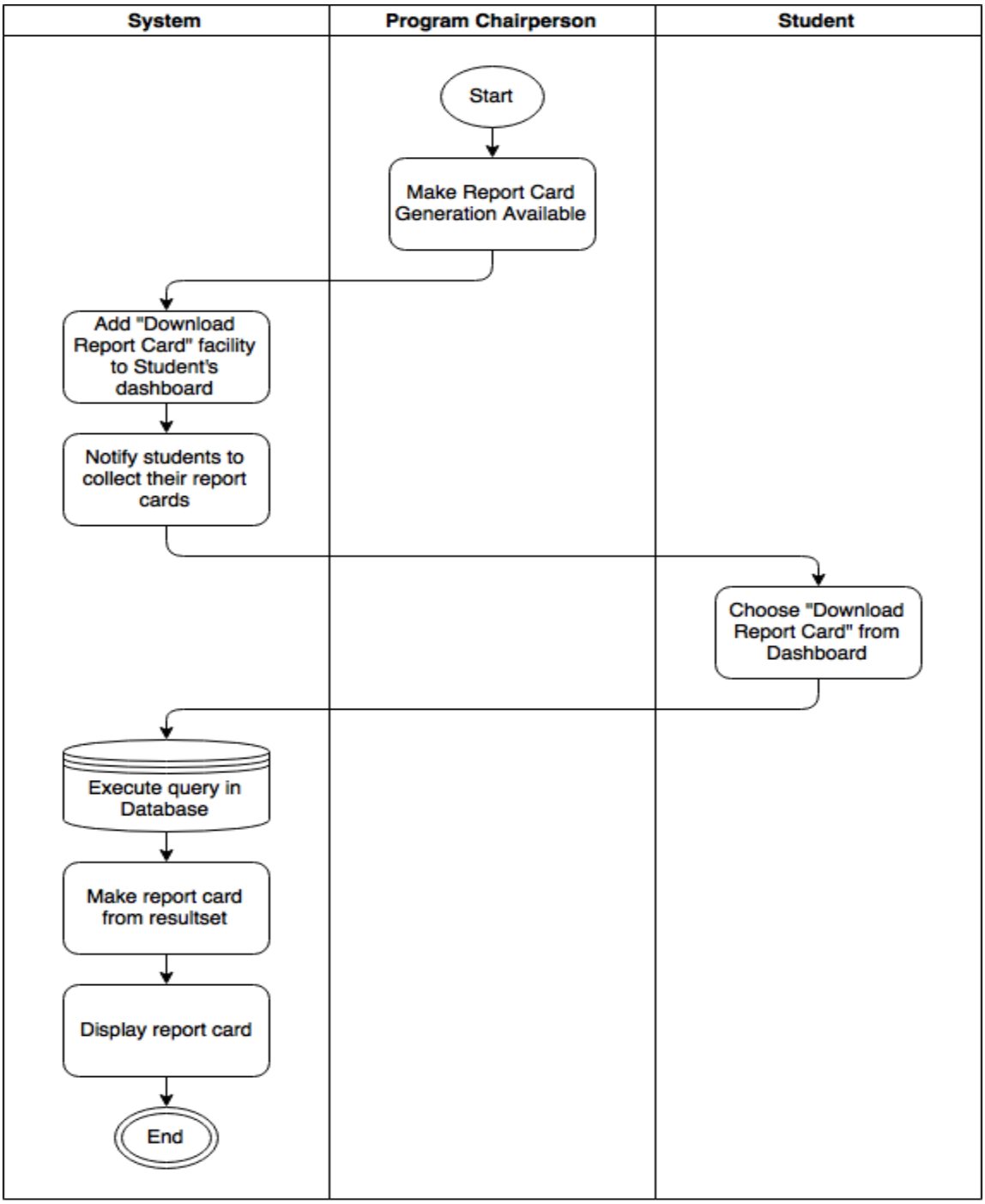


Figure 4.4.3.4: Swimlane diagram for Report Card Generation Process in Result Generation System

4.4.3.5 Course Result Viewing Process

Refers Use Case figure 4.2.3.2:

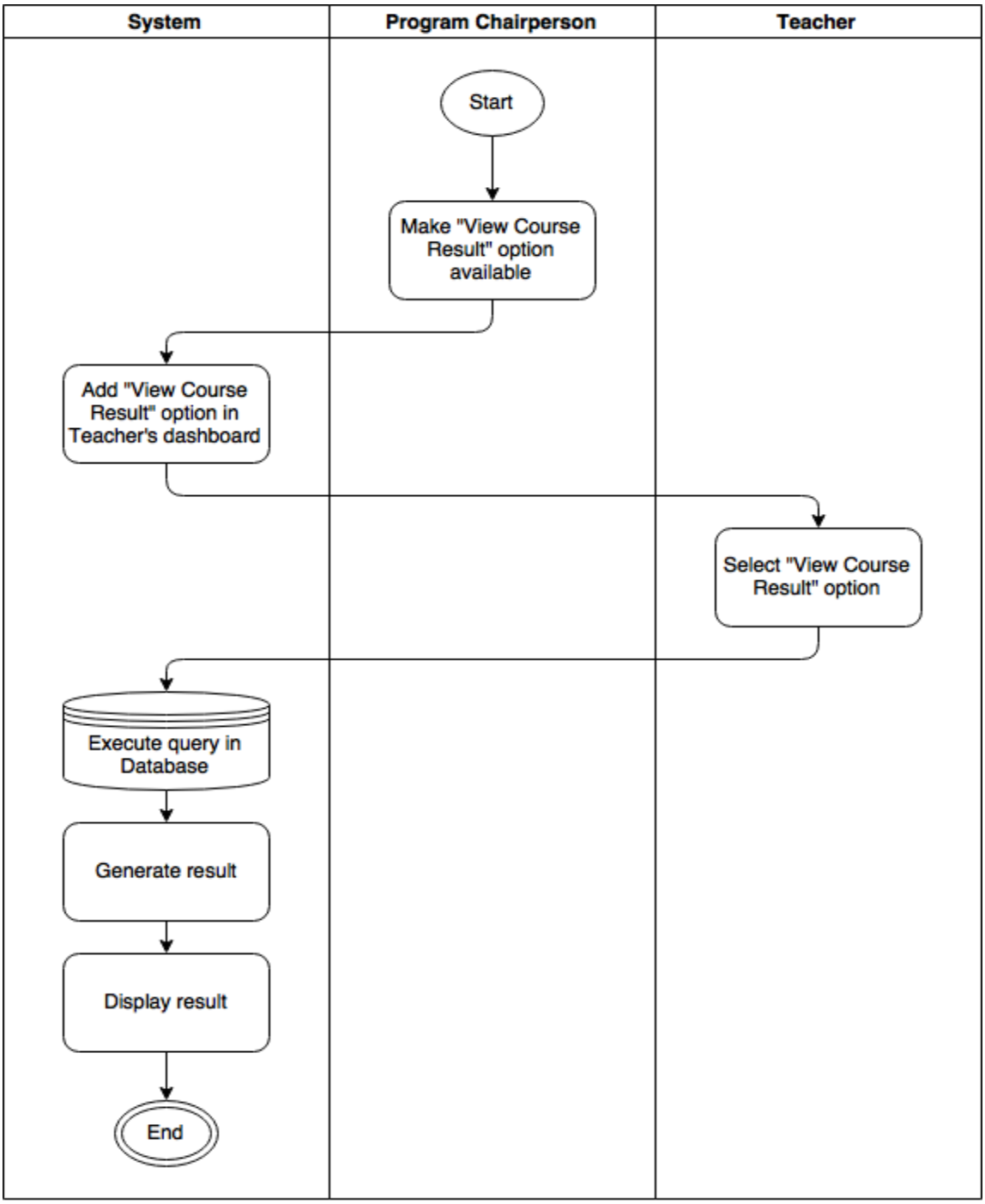


Figure 4.4.3.5: Swimlane diagram for Course Result Viewing Process in Result Generation System

Chapter 5: Data Based Modeling of PGDIT Automation System

5.1 Noun Identification

5.1.1 Authentication System

No.	Noun/Noun Phrase	P/S	Attribute
1.	User	S	11-14, 34, 35
2.	Dashboard	P	
3.	Teacher	S	11, 9, 15, 36-38
4.	Student	S	11, 16, 19
5.	Exam controller	P	
6.	Program chairperson	P	
7.	Scrutinizing team	P	
8.	Receptionist	P	
9.	Name	S	
10.	Installation	P	
11.	Username	S	
12.	Email address	S	
13.	Type of user	S	
14.	Password	S	
15.	Mobile number	S	

16.	Personal information	S	9, 36-40
17.	Verification	P	
18.	Marks	S	
19.	CGPA	S	
20.	Exam schedule	P	
21.	Exam results	S	
22.	Server	P	
23.	Course	S	
24.	Attendance sheets	P	
25.	Seat plan	P	
26.	Room	S	
27.	Tabulation sheet	P	
28.	Application payment slip	P	
29.	Admission payment slip	P	
30.	Candidate information	P	
31.	Database	P	
32.	Payment status	S	
33.	Validity of an application	S	
34.	Wrong Attempt	S	
35.	Recovery Code	S	
36.	Father's Name	S	
37.	Mother's Name	S	
38.	Address	S	
39.	Birth Date	S	
40.	Nationality	S	

Table 5.1.1: Noun Identification Table for Authentication System

Prospective Data Objects:

- **User:** Username, Email address, Type of user, Password, Wrong Attempt, Recovery Code
- **Teacher:** Username, Name, Mobile number, Father's Name, Mother's Name, Address
- **Student:** Username, Name, Father's Name, Mother's Name, Address, Birth Date, Nationality, CGPA
- **Personal Information:** Name, Father's Name, Mother's Name, Address, Birth Date, Nationality

Analysis on Prospective Data Objects:

Since all attributes of the "Personal Information" object belong to "Student" object, we will merge the two data objects and make them "Student".

Actual Data Objects:

- **User:** Username, Email address, Type of user, Password, Wrong Attempt, Recovery Code
- **Teacher:** Username, Name, Mobile number, Father's Name, Mother's Name, Address
- **Student:** Username, Name, Father's Name, Mother's Name, Address, Birth Date, Nationality, CGPA

5.1.2 Admission System

No.	Noun/Noun Phrase	P/S	Attributes
1.	Admission system	P	
2.	Admission process	P	
3.	PGDIT program	P	
4.	Institute	P	
5.	Admission circular	P	
6.	Starting date	P	
7.	Candidates	S	
8.	Interested candidates	S	
9.	Application form	S	11-41, 43
10.	Applicant Data	S	9, 44, 47, 49, 50, 56, 57, 58
11.	Name of Applicant	S	
12.	Father's Name	S	
13.	Mother's Name	S	
14.	E-mail Address	S	
15.	Mobile Number	S	
16.	Address	S	
17.	Permanent Address	S	
18.	Date of Birth	S	
19.	Nationality	S	
20.	School Name	S	
21.	School Board	S	
22.	School Group	S	
23.	School Year	S	

24.	School GPA	S	
25.	College Name	S	
26.	College Board	S	
27.	College Group	S	
28.	College Year	S	
29.	College GPA	S	
30.	University Name	S	
31.	Department Name	S	
32.	University Year	S	
33.	University GPA	S	
34.	Algebra in HSC	S	
35.	Algebra in Bachelor	S	
36.	Math in HSC	S	
37.	Math Percentage in HSC	S	
38.	Math in Bachelor	S	
39.	Math Percentage in Bachelor	S	
40.	Eligibility for Viva	S	
41.	Enrollment status	S	
42.	Given information	P	
43.	Application ID	S	
44.	Application fee payment status	S	
45.	Receptionist	P	
46.	Scrutinizing team	P	
47.	Eligibility of application	S	
48.	Prerequisites	S	
49.	Unique roll number	S	

50.	Room No	S	
51.	VIVA shifts	S	
52.	Teacher	S	53
53.	Teacher's username	S	
54.	VIVA Schedule	S	
55.	Attendance sheet	S	11, 49
56.	Written marks	S	
57.	VIVA marks	S	
58.	Admission fee payment status	S	
59.	Room Capacity	S	
60.	Room	S	50, 59

Table 5.1.2: Noun Identification Table for Admission System

Prospective Data Objects:

- **Application Form:** ApplicationID, Name of Applicant, Father's Name, Mother's Name, E-mail Address, Mobile Number, Permanent Address, Date of Birth, Nationality, School Name, School Board, School Group, School Year, School GPA, College Name, College Board, College Group, College Year, College GPA, University Name, Department Name, University Year, University GPA, Algebra in HSC, Algebra in Bachelor, Math in HSC, Math Percentage in HSC, Math in Bachelor, Math Percentage in Bachelor
- **Applicant Data:** ApplicationID, Year, Name of Applicant, Father's Name, Mother's Name, E-mail Address, Mobile Number, Permanent Address, Date of Birth, Nationality, School Name, School Board, School Group, School Year, School GPA, College Name, College Board, College Group, College Year, College GPA, University Name, Department Name, University Year, University GPA, Algebra in HSC, Algebra in Bachelor, Math in HSC, Math Percentage in HSC, Math in Bachelor, Math Percentage in Bachelor, Eligibility for Viva, Enrollment status, Application fee payment status, Eligibility of application, Unique roll number, Room No, Written marks, VIVA marks, Admission fee payment status
- **Room:** Room No, Room Capacity
- **Attendance Sheet:** Exam roll number, Name of Applicant

Analysis on Prospective Data Objects:

All attributes of data objects "Application Form" and "Attendance Sheet" are present in the data object "Applicant Data". Therefore, we will merge the three objects and make it

"Applicant Data". We will also not use the "Attendance Sheet" data object as the system does not need to save the contents of the object. The system will only generate an attendance sheet once and then may forget it.

Actual Data Objects:

- **Applicant Data:** ApplicationID, Year, Name of Applicant, Father's Name, Mother's Name, E-mail Address, Mobile Number, Permanent Address, Date of Birth, Nationality, School Name, School Board, School Group, School Year, School GPA, College Name, College Board, College Group, College Year, College GPA, University Name, Department Name, University Year, University GPA, Algebra in HSC, Algebra in Bachelor, Math in HSC, Math Percentage in HSC, Math in Bachelor, Math Percentage in Bachelor, Eligibility for Viva, Enrollment status, Application fee payment status, Eligibility of application, Unique roll number, Room No, Written marks, VIVA marks, Admission fee payment status
- **Room:** Room No, Room Capacity

5.1.3 Result Generation System

No.	Noun/Noun Phrase	P/S	Attribute
1.	Result generation system	P	
2.	Examination result	P	
3.	Total marks	S	
4.	Course	S	5, 17-21, 36, 37
5.	Course ID	S	
6.	Student roll	S	
7.	Grade	S	
8.	Supplementary exam	P	
9.	Final grades	S	
10.	Detailed result	P	
11.	Term Result	P	
12.	Report Card	P	
13.	Prerequisite	S	
14.	Year of exam	S	
15.	CGPA	S	
16.	Final Result	P	
17.	Course name	S	
18.	Course Credit	S	
19.	Prerequisite 1	S	
20.	Prerequisite 2	S	
21.	Prerequisite 3	S	
22.	Academic Profile	S	6, 15, 23-35

23.	Grade of PGD_101	S	
24.	Grade of PGD_104	S	
25.	Grade of PGD_105	S	
26.	Grade of PGD_106	S	
27.	Grade of PGD_107	S	
28.	Grade of PGD_201	S	
29.	Grade of PGD_202	S	
30.	Grade of PGD_203	S	
31.	Grade of PGD_204	S	
32.	Grade of PGD_206	S	
33.	Grade of PGD_207	S	
34.	Grade of PGD_208	S	
35.	Grade of PGD_210	S	
36.	Course_Outline	S	
37.	Description	S	

Table 5.1.3: Noun Identification Table for Result Generation System

Prospective Data Objects:

- **Course:** Course_ID, Course_Name, Course_Credit, Course_Outline, Description, Prerequisite_1, Prerequisite_2, Prerequisite_3
- **Academic Profile:** Student_Roll, Grade_of_PGD_101, Grade_of_PGD_104, Grade_of_PGD_105, Grade_of_PGD_106, Grade_of_PGD_107, Grade_of_PGD_201, Grade_of_PGD_202, Grade_of_PGD_203, Grade_of_PGD_204, Grade_of_PGD_206, Grade_of_PGD_207, Grade_of_PGD_208, Grade_of_PGD_210, CGPA

Actual Data Objects:

- **Course:** Course_ID, Course_Name, Course_Credit, Course_Outline, Description, Prerequisite_1, Prerequisite_2, Prerequisite_3
- **Academic Profile:** Student_Roll, Grade_of_PGD_101, Grade_of_PGD_104, Grade_of_PGD_105, Grade_of_PGD_106, Grade_of_PGD_107, Grade_of_PGD_201, Grade_of_PGD_202, Grade_of_PGD_203, Grade_of_PGD_204, Grade_of_PGD_206, Grade_of_PGD_207, Grade_of_PGD_208, Grade_of_PGD_210, CGPA

Normalizing Tables:

- In the Student table, the following attributes "Name, Father's Name, Address, Birth Date, Nationality" are redundant. Because "Applicant" table consumes these attributes. There is a (has a) relation between "Student" and "Academic Profile". We can now merge these two tables and rename it as "Student". Then the "Student" entity will contain the following attributes.
  - ★ **Student:** Username, Grade_of_PGD_101, Grade_of_PGD_104, Grade_of_PGD_105, Grade_of_PGD_106, Grade_of_PGD_107, Grade_of_PGD_201, Grade_of_PGD_202, Grade_of_PGD_203, Grade_of_PGD_204, Grade_of_PGD_206, Grade_of_PGD_207, Grade_of_PGD_208, Grade_of_PGD_210, CGPA

5.2 Data Object Relation

5.2.1 PGDIT Automation System

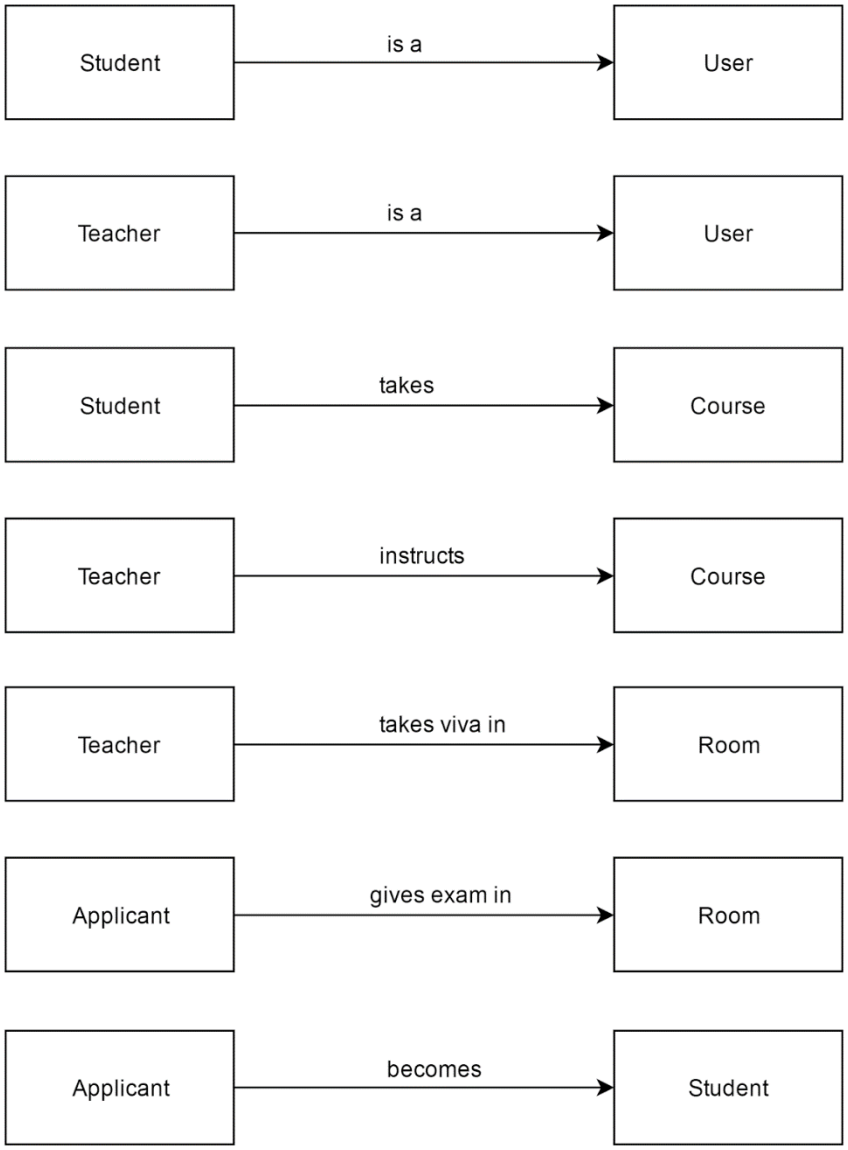


Figure 5.2.1: Data Object Relationship Diagram of PGDIT Automation System

5.3 ER Diagram

5.3.1 PGDIT Automation System

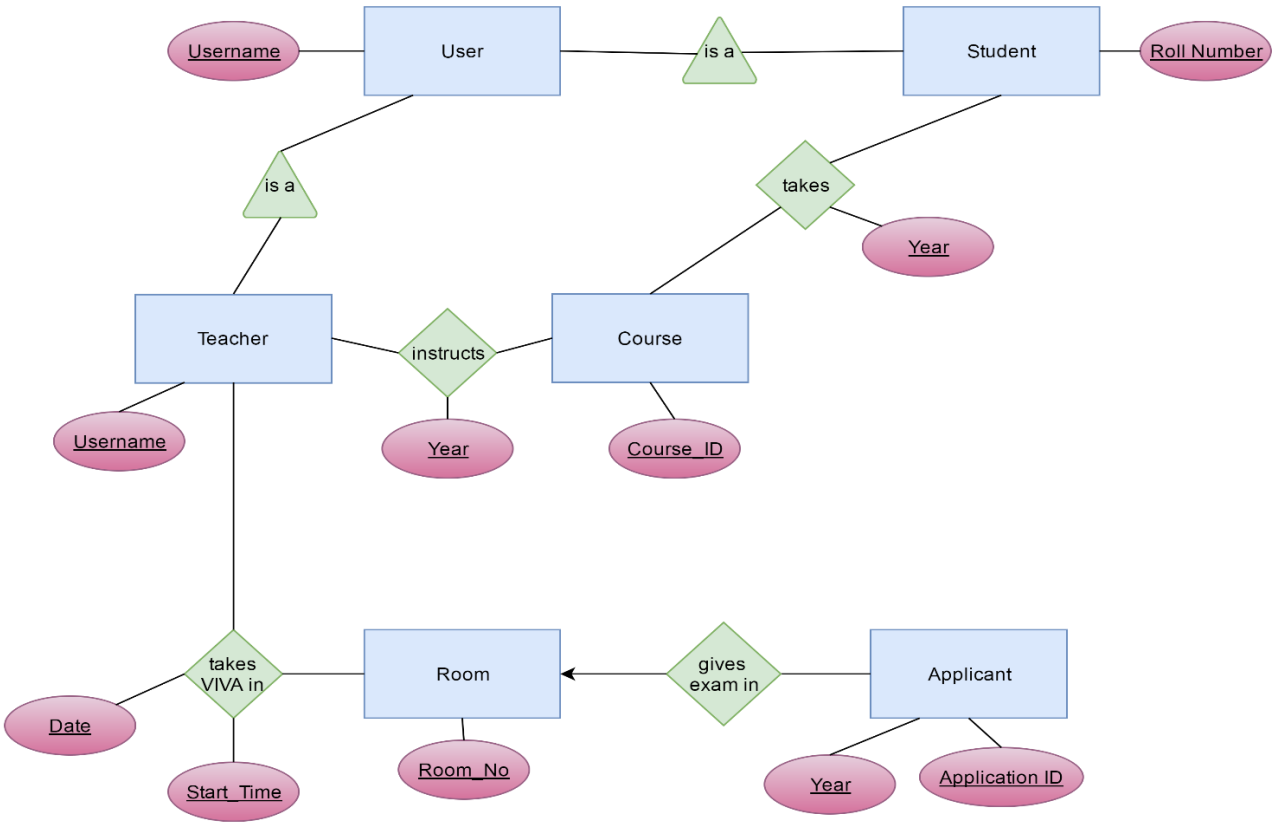


Figure 5.3.1: Entity Relationship Diagram of PGDIT Automation System

5.4 Relational Tables

The following tables will be introduced in order to store the data of the relations of the data object:

- **Student takes Course:** Course_ID, Unique_roll_number, Year_of_exam, Total_marks, Grade
- **Teacher Instructs Course:** Username, CourseID, Year_of_exam, Course State
- **VIVA Schedule:** Username, VIVA Date, Start_time, End_time, Room_No

5.5 Schema Tables

5.5.1 Authentication System

User		
Column	**Type**	**Size**
Username	VARCHAR	20
Password	VARCHAR	30
Email_Address	VARCHAR	50
Wrong_Attempt	INTEGER	1
User_Type	VARCHAR	30
Recovery_code	VARCHAR	6
Full Name	VARCHAR	50
Father Name	VARCHAR	50
Mother Name	VARCHAR	50
Address	VARCHAR	250
Mobile Number	VARCHAR	20

Table 5.5.1.1: Schema Table for table User

Teacher		
Column	**Type**	**Size**
Username	VARCHAR	20
Name	VARCHAR	50
Father_Name	VARCHAR	50
Mother_Name	VARCHAR	50
Address	VARCHAR	250
Mobile_Number	VARCHAR	14

Table 5.5.1.2: Schema Table for table Teacher

5.5.2 Admission System

Applicant Data		
Column	**Type**	**Size**
Application_ID	VARCHAR	6
Year	INTEGER	4
Name	VARCHAR	50
Father_Name	VARCHAR	50
Mother_Name	VARCHAR	50
E-mail_Address	VARCHAR	50
Mobile_Number	VARCHAR	14
Permanent_Address	VARCHAR	250
Date_of_Birth	DATE	
Nationality	VARCHAR	30
School_Name	VARCHAR	50
School_Board	VARCHAR	20
School_Group	VARCHAR	20
School_Year	INTEGER	4
School_GPA	DECIMAL	1, 2
College_Name	VARCHAR	50
College_Board	VARCHAR	20
College_Group	VARCHAR	20
College_Year	INTEGER	4
College GPA	DECIMAL	1, 2
University_Name	VARCHAR	50

Department_Name	VARCHAR	50
University_Year	INTEGER	4
University_GPA	DECIMAL	1, 2
Algebra_in_HSC	VARCHAR	6
Algebra_in_Bachelor	VARCHAR	6
Math_in_HSC	VARCHAR	6
Math_Percentage_in_HSC	DECIMAL	2, 2
Math_in_Bachelor	VARCHAR	6
Math_Percentage_in_Bachelor	DECIMAL	2, 2
Eligibility_for_Viva	VARCHAR	6
Enrollment_status	VARCHAR	6
Application_fee_payment_status	VARCHAR	6
Eligibility_of_application	VARCHAR	6
Exam_roll_number	VARCHAR	6
Room_No	VARCHAR	10
Written_marks	INTEGER	3
Eligibility for VIVA	VARCHAR	6
VIVA_marks	INTEGER	3
Eligibility for Admission	VARCHAR	6
Admission_fee_payment_status	VARCHAR	6
Enrollment status	VARCHAR	5

Table 5.5.2.1: Schema Table for table Applicant Data

Room		
Column	**Type**	**Size**
Room_No	VARCHAR	30
Room_Capacity	INTEGER	3

Table 5.5.2.2: Schema Table for table Room

VIVA Schedule		
Column	**Type**	**Size**
Username	VARCHAR	20
Viva date	DATE	
Start_time	TIME	
End_time	TIME	
Room_No	VARCHAR	30

Table 5.5.2.3: Schema Table for table VIVA Schedule

5.5.3 Result Generation System

Student		
Column	**Type**	**Size**
Roll Number	VARCHAR	6
Username	VARCHAR	20
AdmissionYear	INTEGER	4
Grade_of_PGD_101	DECIMAL	1, 2
Grade_of_PGD_104	DECIMAL	1, 2
Grade_of_PGD_105	DECIMAL	1, 2
Grade_of_PGD_106	DECIMAL	1, 2
Grade_of_PGD_107	DECIMAL	1, 2
Grade_of_PGD_201	DECIMAL	1, 2
Grade_of_PGD_202	DECIMAL	1, 2
Grade_of_PGD_203	DECIMAL	1, 2
Grade_of_PGD_204	DECIMAL	1, 2
Grade_of_PGD_206	DECIMAL	1, 2
Grade_of_PGD_207	DECIMAL	1, 2
Grade_of_PGD_208	DECIMAL	1, 2
Grade_of_PGD_210	DECIMAL	1, 2
CGPA	DECIMAL	1, 2

Table 5.5.3.1: Schema Table for table Student

Course		
Column	**Type**	**Size**
CourseID	VARCHAR	10
Course_Name	VARCHAR	50
Course_Credit	DECIMAL	1, 2
Course_Outline	VARCHAR	500
Description	VARCHAR	500

Table 5.5.3.2: Schema Table for table Course

Teacher Instructs Course		
Column	**Type**	**Size**
Username	VARCHAR	20
CourseID	VARCHAR	10
Year_of_exam	INTEGER	4
Course State	VARCHAR	10

Table 5.5.3.3: Schema Table for table Teacher Instructs Course

Student Takes Course		
Column	**Type**	**Size**
Course_ID	VARCHAR	10
Unique_roll_number	VARCHAR	6
Year_of_exam	INTEGER	4
Total_marks	INTEGER	2
Grade	DECIMAL	1, 2

Table 5.5.3.4: Schema Table for table Student Takes Course

Chapter 6: Class Based Modeling of PGDIT Automation System

6.1 General Classification

1. **External Entities** (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
2. **Things** (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. **Occurrence or events** (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. **Roles** (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. **Organizational units** (e.g., division, group, and team) that are relevant to an application.
6. **Places** (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. **Structure** (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

No.	Noun/noun phrases	General Classification	Remarks
1.	User	4, 7	✓
2.	Dashboard	2, 7	✓
3.	Teacher	4, 7	✓
4.	Student	4, 7	✓
5.	Exam controller	4, 7	✓
6.	Program chairperson	4, 7	✓

7.	Scrutinizing team	4, 7	✓
8.	Receptionist	4, 7	✓
9.	Admin	4, 7	✓
10.	Installation	3	
11.	Username		
12.	Email address		
13.	Type of user		
14.	Password		
15.	Mobile number		
16.	Personal information		
17.	Verification		
18.	CGPA		
19.	Exam schedule		
20.	Server	2, 7	✓
21.	Course	2, 7	✓
22.	Attendance sheet	2, 7	✓
23.	Seat plan		
24.	Tabulation sheet		
25.	Application payment slip		
26.	Admission payment slip		
27.	Database	2, 4, 7	✓
28.	Institute	4, 5	✓
29.	Admission circular	2	
30.	Starting date		
31.	Interested candidates	4, 7	✓

32.	Application form	2, 4, 7	✓
33.	Applicant data	4, 7	✓
34.	Name		
35.	Father's name		
36.	Mother's name		
37.	Present address		
38.	Permanent address		
39.	Date of Birth		
40.	Nationality		
41.	School name		
42.	School board		
43.	School group		
44.	School year		
45.	School GPA		
46.	College name		
47.	College board		
48.	College group		
49.	College year		
50.	College GPA		
51.	University name		
52.	Department name		
53.	University year		
54.	University GPA		
55.	Algebra in HSC		
56.	Algebra in Bachelor		

57.	Math in HSC		
58.	Math Percentage in HSC		
59.	Math in Bachelor		
60.	Math Percentage in Bachelor		
61.	Eligibility for Viva		
62.	Enrollment status		
63.	Given information		
64.	Application ID		
65.	Application fee payment status		
66.	Eligibility of application		
67.	Prerequisites		
68.	Unique roll number		
69.	Room no		
70.	VIVA shifts		
71.	Teacher's username		
72.	Viva schedule	2, 7	✓
73.	Written marks		
74.	Viva marks		
75.	Admission fee payment status		
76.	Result generation system		
77.	Examination result	2, 7	✓
78.	Total marks		
79.	Course ID		
80.	Grade		
81.	Supplementary exam		

82.	Final grades		
83.	Detailed result		
84.	Term result	2, 7	✓
85.	Report card		
86.	Prerequisite		
87.	Year of exam		
88.	Total CGPA		
89.	Final result	2, 7	✓

Table 6.1: General Classification Table for PGDIT Automation System

6.2 Coad and Yourdon's Six Selection Criteria

Coad and Yourdon's six selection criteria are:

1. **Retained Information:** Will the system need to remember information about this class of objects?
2. **Needed Services:** Do objects in this class have identifiable operations that change the values of their attributes?
3. **Multiple Attributes:** If the class only has one attribute, it may be better represented as an attribute of another class.
4. **Common Attributes:** Does the class have attributes that are shared with all instances of its objects?
5. **Common Operations:** Does the class have operations that are shared with all instances of its objects?
6. **Essential Requirements:** External entities that produce or consume information essential to the system should be included as classes.

No.	Potential Classes	Selection Criteria	Remarks
1.	User	1, 2, 3, 4, 5	✓
2.	Dashboard		
3.	Teacher	1, 2, 3, 4, 5	✓
4.	Student	1, 2, 3, 4, 5	✓
5.	Exam controller	1, 2, 3, 4, 5	✓
6.	Program chairperson	1, 2, 3, 4, 5	✓
7.	Scrutinizing team	1, 2, 3, 4, 5	✓
8.	Receptionist	1, 2, 3, 4, 5	✓
9.	Server		
10.	Course	1	
11.	Attendance sheet	3, 4	
12.	Database Operator	1, 2, 3, 4, 5	✓
13.	Institute		

14.	Interested candidates	1, 3, 4, 5	✓
15.	Application form	1, 3, 4	✓
16.	Applicant	1, 3, 4, 5	✓
17.	Viva schedule	3, 4	
18.	Examination result	3, 4	
19.	Term result	3, 4	
20.	Final result	3, 4	

Table 6.2: Selection Criteria Table for PGDIT Automation System

Selected classes:

1. User
2. Teacher
3. Student
4. Exam controller
5. Program chairperson
6. Scrutinizing team
7. Receptionist
8. Database Operator
9. Interested candidates
10. Application form
11. Applicant

6.3 Noun-Verb Affiliation of Classes

No.	Selected Class	Noun	Verb
1.	User	Username Email address Password Mobile number Name	Register Log in Verify View Dashboard Change Password Change personal data Log out
2.	Teacher	Username Email address Password Mobile number Name	Register Log in Verify Change Password Log out View course result Upload marks Change Description View supplementary list View result
3.	Student	Username Email address Password Mobile number Name Roll number	Register Log in Verify Change Password Log out View Report Card
4.	Exam controller	Username Email address Password Mobile number Name	Register Log in Verify Change Password Log out Invoke process Upload marks Generate seat plan
5.	Program chairperson	Username Email address Password Mobile number Name	Register Log in Verify Change Password Log out Upload circular

			Turn on process Turn off process Verify new user Generate new student account Enroll teachers Enroll students
6.	Scrutinizing team	Username Email address Password Mobile number Name	Register Log in Verify Change Password Log out Download applicant data Verify eligibility of application
7.	Receptionist	Username Email address Password Mobile number Name	Register Log in Verify Change Password Log out Verify payment
8.	Database Operator	MySQL Connector Statement Query Result Set	Store in Database Update Database
9.	Interested candidates	Application ID Name of Applicant E-mail Address Mobile Number Date of Birth Nationality Eligibility of application Eligibility for Viva Enrollment status	Authenticate Download Admit Card
10.	Application form	Application ID Name of Applicant E-mail Address Mobile Number Date of Birth Nationality	
11.	Applicant	Application ID Name of Applicant	Authenticate Download Admit Card

		E-mail Address Mobile Number Date of Birth Nationality Eligibility of application Eligibility for VIVA Enrollment status	

Table 6.3: Noun-Verb Affiliation of Classes Table for PGDIT Automation System

Analysis of Selected Classes:

- **<u>Merging into *User* Superclass:</u>**
  - User, Teacher, Student, Exam Controller, Program Chairperson, Receptionist, Scrutinizing Team classes have some common noun-verb affiliations. Therefore, these common affiliations will be in the class User. Teacher, Student, Exam Controller, Program Chairperson, Receptionist, Scrutinizing Team will be subclasses of the User class. Since, the User class represents all of them but there can be no instance of User class, it will therefore be an abstract class.
- **<u>Merging into *Applicant* class:</u>**
  - The class Applicant contains all attributes and methods of the classes Interested candidates and Application form. Therefore, we will merge these classes and name it "Applicant" class which will contain the responsibilities of these classes.

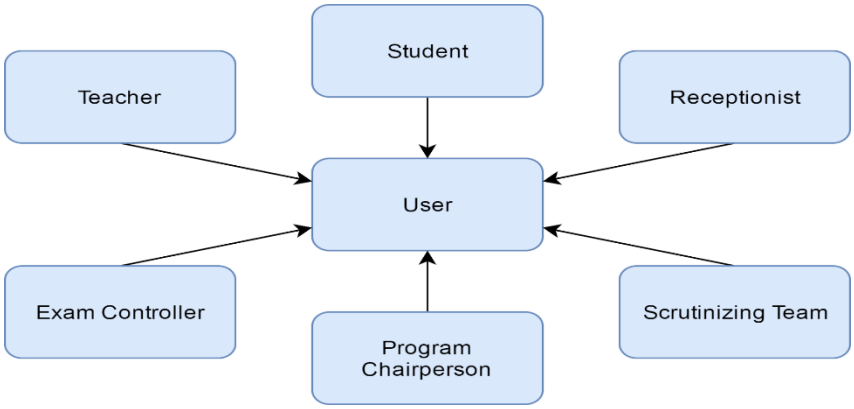


Figure 6.3: User Superclass

6.4 Class Attributes and Class Methods

After analysis, we get following classes, their attributes and methods:

No.	Final Class	Attributes	Methods
1.	User	Username FullName	viewHeader () viewDashboard() viewPasswordField () viewNotification ($notificationID) getNotification () updatePassword ($username, $password) updateEmail ($username, $email) updateMobileNo ($username, $mobile) updateAddress ($username, $address) updatePicture ($username) verifyPassword ($username, $password)
2.	Teacher		viewListOfCourses () viewResultSheet ($courseID, $year) updateStudentMarks ($courseID, $year) viewSupplymentaryList ($courseID, $year) viewVivaSchedule () viewCourseInformation ($courseID, $year) notifyForSupplimentaryExam ($courseID, $year) updateCourseInformation ($courseID, $year, $outline, $description) updatePdf ($username, $courseID, $year)
3.	Student	Roll number	viewResultSheet () viewRunningCourses () viewCourseInfo ($courseID)

4.	ExamController		viewListOfRoom () updateRoomCapacity ($roomno, $capacity) addNewRoom ($roomno, $capacity) viewVivaSchedule () addVivaSchedule ($usernamelist, $vivaDate, $startTime, $endTime, $roomNo) viewListOfApplicants () introduceNewAdmission ($AdmissionYear, $FormFillUpStartAt, $FormFillUpEndAt, $ApplicationPaymentStartAt, $ApplicationPaymentEndAt, $ScrutinizingStartAt, $AdmissionDate, $AdmissionPaymentStartAt, $AdmissionPaymentEndAt) updateAdmissionInfo ($FormFillUpStartAt, $FormFillUpEndAt, $ApplicationPaymentStartAt, $ApplicationPaymentEndAt, $ScrutinizingStartAt, $AdmissionDate, $AdmissionPaymentStartAt, $AdmissionPaymentEndAt) viewAdmissionPanel () viewListOfEligibleCandidatesOnScrutinize () changeScrutinizingStatus () viewListOfEligibleCandidatesAfterScrutinize () updateAvailabilityOfAdmitCard () viewListOfEligibleCandidatesAfterWrittenExam () updateWrittenMarks ($applicantList, $markList) lockWrittenMarks ($capacity) viewListOfEligibleCandidatesAfterVIVAExam () updateVivaMarks ($applicantList, $markList) lockVivaMarks ($capacity) viewListOfEligibleCandidatesAfterFullResult ()

			viewListOfEnrolledCandidates () updateEnrollmentState ($applicantList) lockEnrollmentState ($applicantList) viewListOfEnrolledApplicants () endAdmissionSession ()
5.	ProgramChairperson		viewListOfCourses () viewCourseWiseResult ($courseID, $year) changeInstructor ($courseID, $year, $instructor) lockCourseResult ($courseID, $year) addNewStudentsIntoCourse ($courseID, $year, $rollList) viewListOfUser () viewInfoOfOneUser ($username) updateUserInformation ($username, $fullname, $fathername, $mothername, $email)
6.	Scrutinizer	List of Students	updateStatusOfEligiblity ($applicationID, $index) viewSearchBar () searchByApplicationID ($applicationID) searchByIndex ($index)
7.	Receptionist		viewPaymentBoard () updateStatusOfPayment ($paymentType, $applicationID)
8.	Database Operator	DBConnector	executeSelectCommand ($selectSql) executeUpdateCommand ($sql) executeInsertCommand ($sql)

Table 6.4: Class Attributes and Methods Table for PGDIT Automation System

6.5 Class Cards

User	
Responsibilities	**Collaborators**
Viewing Personal Data	DatabaseOperator
Changing Personal Data	DatabaseOperator
Viewing Notification	DatabaseOperator
Viewing dashboard	DatabaseOperator

Class Card 1: Class Card for Class User

Teacher	
Responsibilities	**Collaborators**
Uploading marks	DatabaseOperator
Changing course information	DatabaseOperator
Viewing supplementary list	DatabaseOperator
Viewing course result	DatabaseOperator
Viewing list of courses	DatabaseOperator
Viewing VIVA Schedule	DatabaseOperator

Class Card 2: Class Card for Class Teacher

Student	
Responsibilities	**Collaborators**
Viewing Report Card	DatabaseOperator
Viewing Running Courses	DatabaseOperator

Class Card 3: Class Card for Class Student

ExamController	
Responsibilities	**Collaborators**
Uploading Admission Examination Marks	DatabaseOperator
Generating seat plan	DatabaseOperator
Generating VIVA plan	DatabaseOperator
Turning off "Result update policy"	DatabaseOperator
Updating room capacity	DatabaseOperator
Adding new rooms	DatabaseOperator
Introducing new admission session	DatabaseOperator
Updating admission session information	DatabaseOperator
Updating signature of Director	DatabaseOperator
Locking and unlocking scrutinizing session	DatabaseOperator
Updating availability of admit cards	DatabaseOperator
Updating enrollment state	DatabaseOperator

Class Card 4: Class Card for Class ExamController

ProgramChairperson	
Responsibilities	**Collaborators**
Viewing list of running courses	DatabaseOperator
Viewing course wise result	DatabaseOperator
Changing instructor of a course	DatabaseOperator
Notifying about the end of a course	DatabaseOperator
Viewing list of users	DatabaseOperator
Updating user information	DatabaseOperator
Verifying new user account	DatabaseOperator

Generating new student account	DatabaseOperator
Enrolling students into courses	DatabaseOperator
Introducing new course	DatabaseOperator

Class Card 5: Class Card for Class ProgramChairperson

Scrutinizer	
Responsibilities	**Collaborators**
Getting list of applicants	DatabaseOperator
Viewing applicant information	DatabaseOperator
Verifying eligibility of application	DatabaseOperator

Class Card 6: Class Card for Class ScrutinizingTeam

Receptionist	
Responsibilities	**Collaborators**
Verifying payment	DatabaseOperator
Searching applicant information	DatabaseOperator

Class Card 7: Class Card for Class Receptionist

DatabaseOperator	
Responsibilities	**Collaborators**
Storing data into database	User, Applicant
Updating database	User, Applicant
Retrieving information from database	User

Class Card 8: Class Card for Class DatabaseOperator

6.6 Class Responsibility Collaborator Diagram

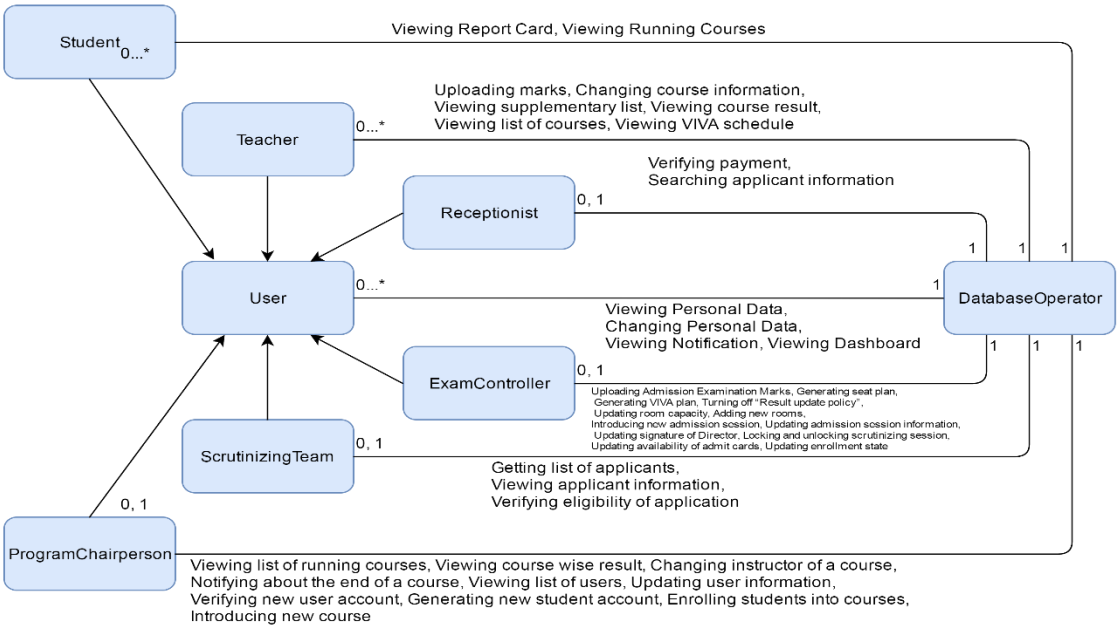


Figure 6.6: Class Responsibility Collaborator Diagram for PGDIT Automation System

Chapter 7: Behavioral Modeling of PGDIT Automation System

7.1 Event Initiator and Collaborators Identification

7.1.1 PGDIT Automation System

Events	Initiator	Collaborators
Chose to change personal data	User	DatabaseOperator
Logged out	User	DatabaseOperator
Uploaded marks	Teacher	DatabaseOperator
Changed course description	Teacher	DatabaseOperator
Chose to view report card	Student	DatabaseOperator
Uploaded admission examination marks	ExamController	DatabaseOperator
Invoked generation seat plan	ExamController	DatabaseOperator
Turned on process	ProgramChairperson	DatabaseOperator
Turned off process	ProgramChairperson	DatabaseOperator
Verified new user account	ProgramChairperson	DatabaseOperator
Invoked generation of new student account	ProgramChairperson	DatabaseOperator
Enrolled students into courses	ProgramChairperson	DatabaseOperator
Enrolled teachers into courses	ProgramChairperson	DatabaseOperator
Got list of applicants	ScrutinizingTeam	DatabaseOperator
Verified eligibility of application	ScrutinizingTeam	DatabaseOperator

Verified payment	Receptionist	DatabaseOperator
Chose to download admit card	Applicant	DatabaseOperator

Table 7.1.1.1: Event Initiator, Collaborator table for PGDIT Automation System

7.2 State Transition Diagrams

7.2.1 PGDIT Automation System

Class 1: User

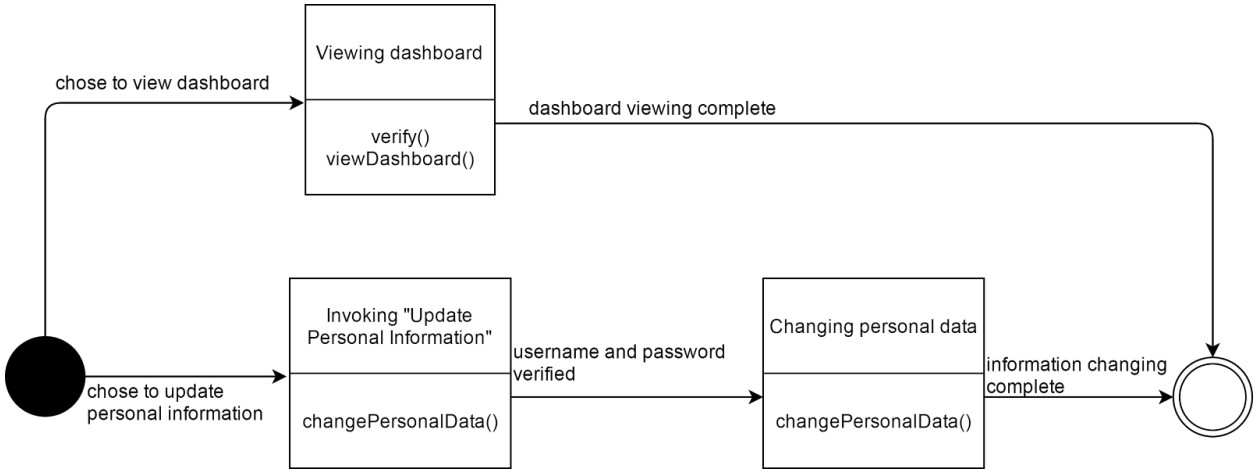


Figure 7.2.1.1: State Transition Diagram for class User

Class 2: Teacher

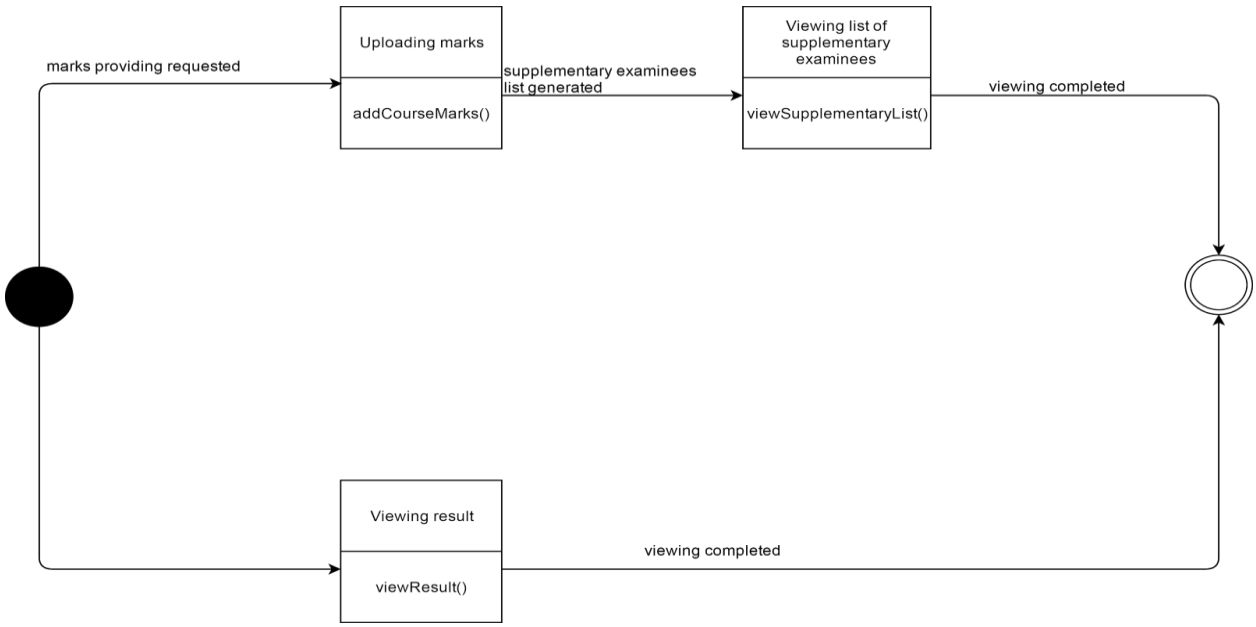


Figure 7.2.1.2: State Transition Diagram for class Teacher

Class 3: Student

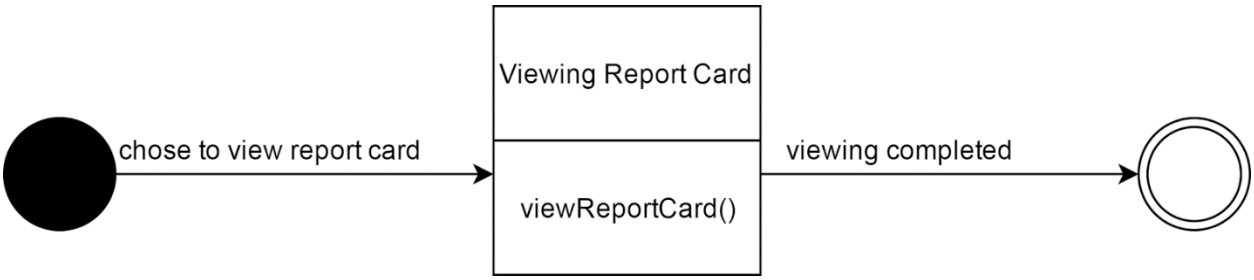


Figure 7.2.1.3: State Transition Diagram for class Student

Class 4: ExamController

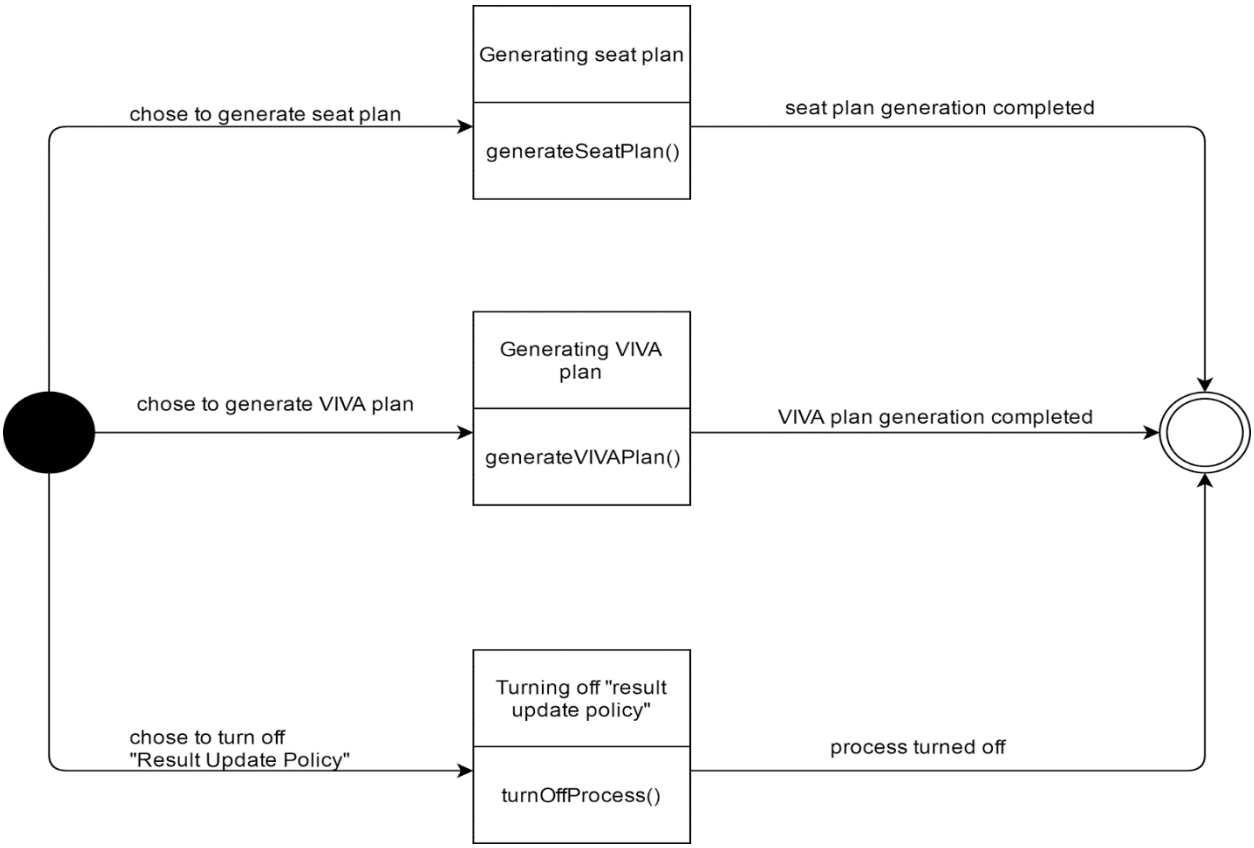


Figure 7.2.1.4: State Transition Diagram for class ExamController

Class 5: ProgramChairperson

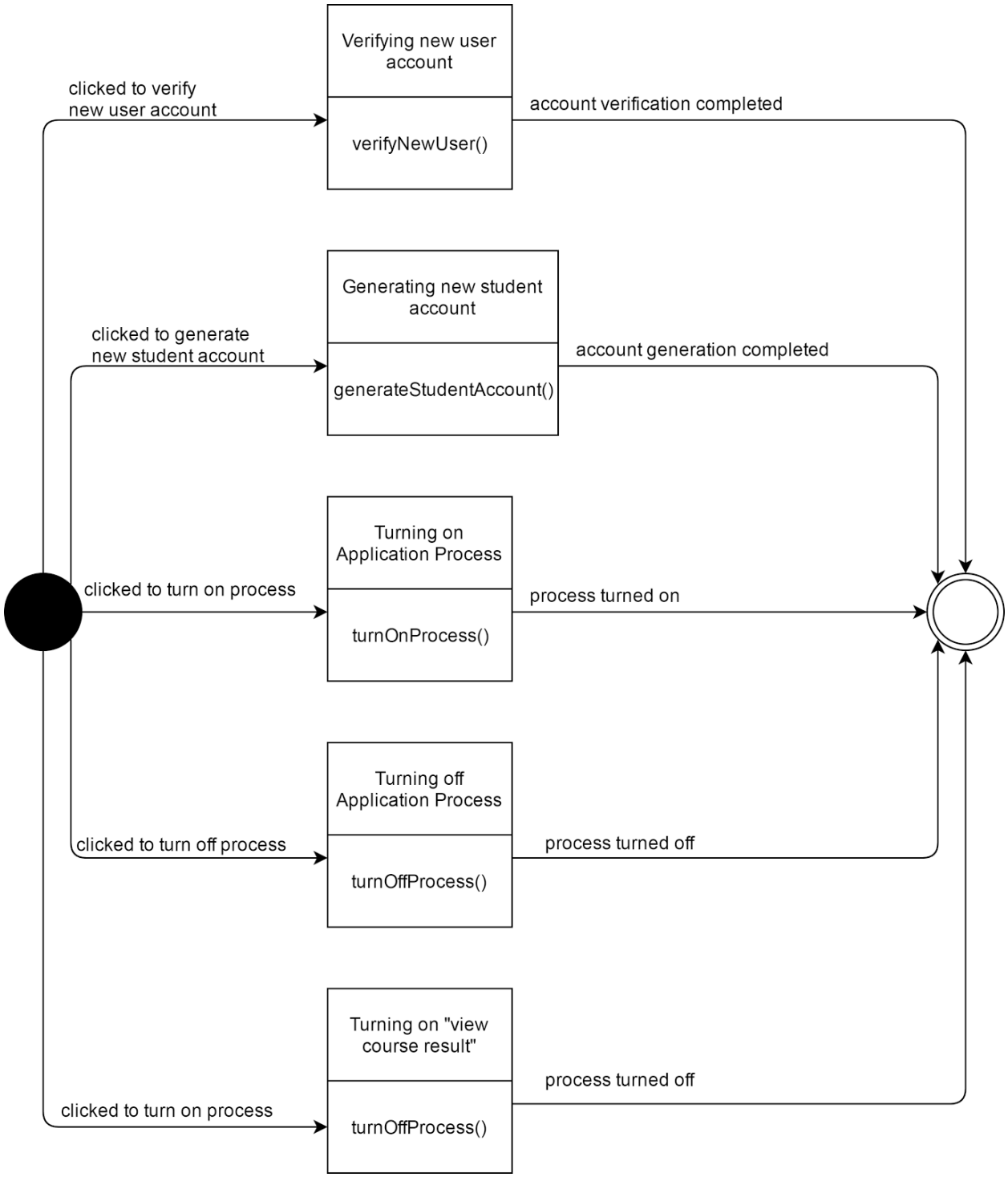


Figure 7.2.1.5: State Transition Diagram for class ProgramChairperson

Class 6: ScrutinizingTeam

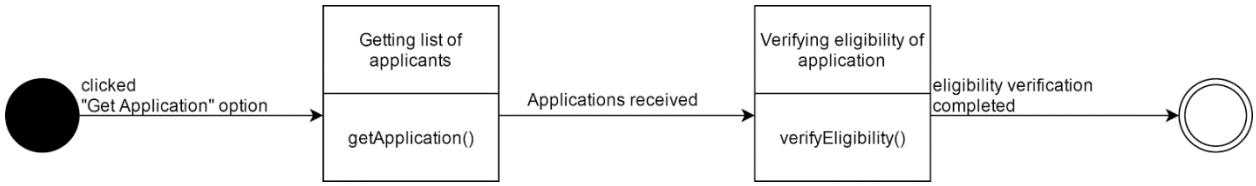


Figure 7.2.1.6: State Transition Diagram for class ScrutinizingTeam

Class 7: Receptionist

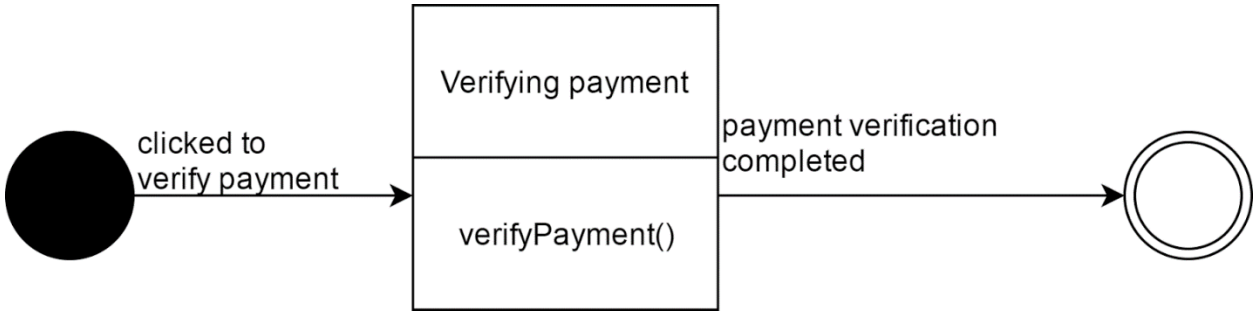


Figure 7.2.1.7: State Transition Diagram for class Receptionist

Class 8: DatabaseOperator

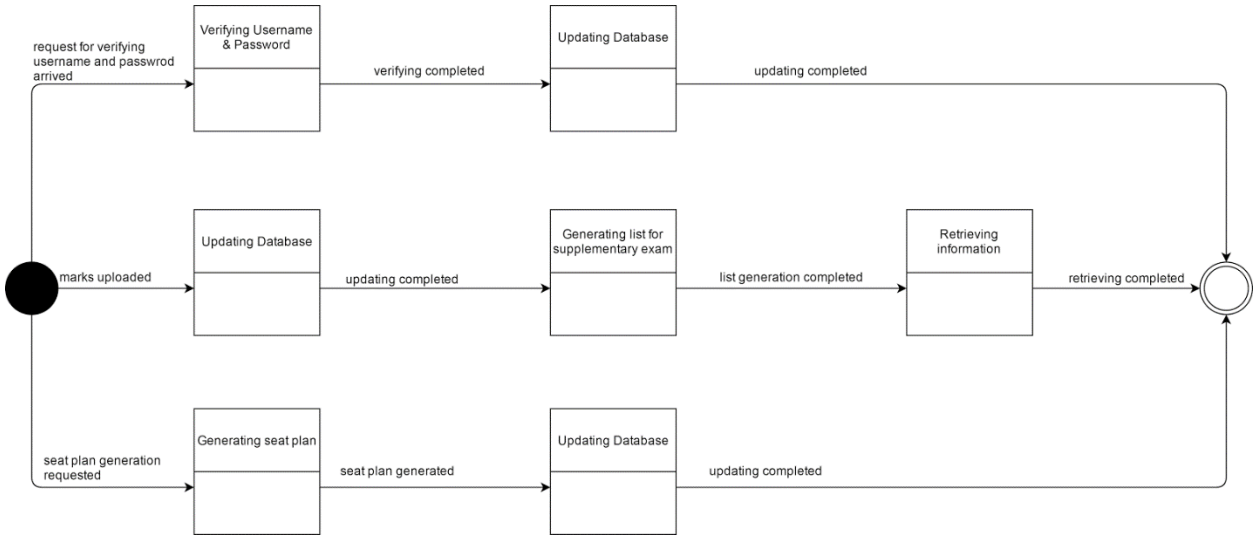


Figure 7.2.1.8: State Transition Diagram for class DatabaseOperator

Class 9: Applicant

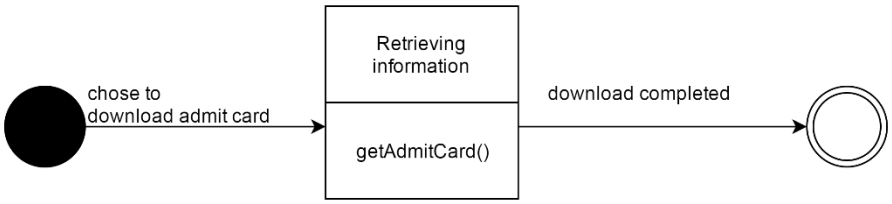


Figure 7.2.1.9: State Transition Diagram for class Applicant

7.3 Sequence Diagrams

7.3.1 Authentication System

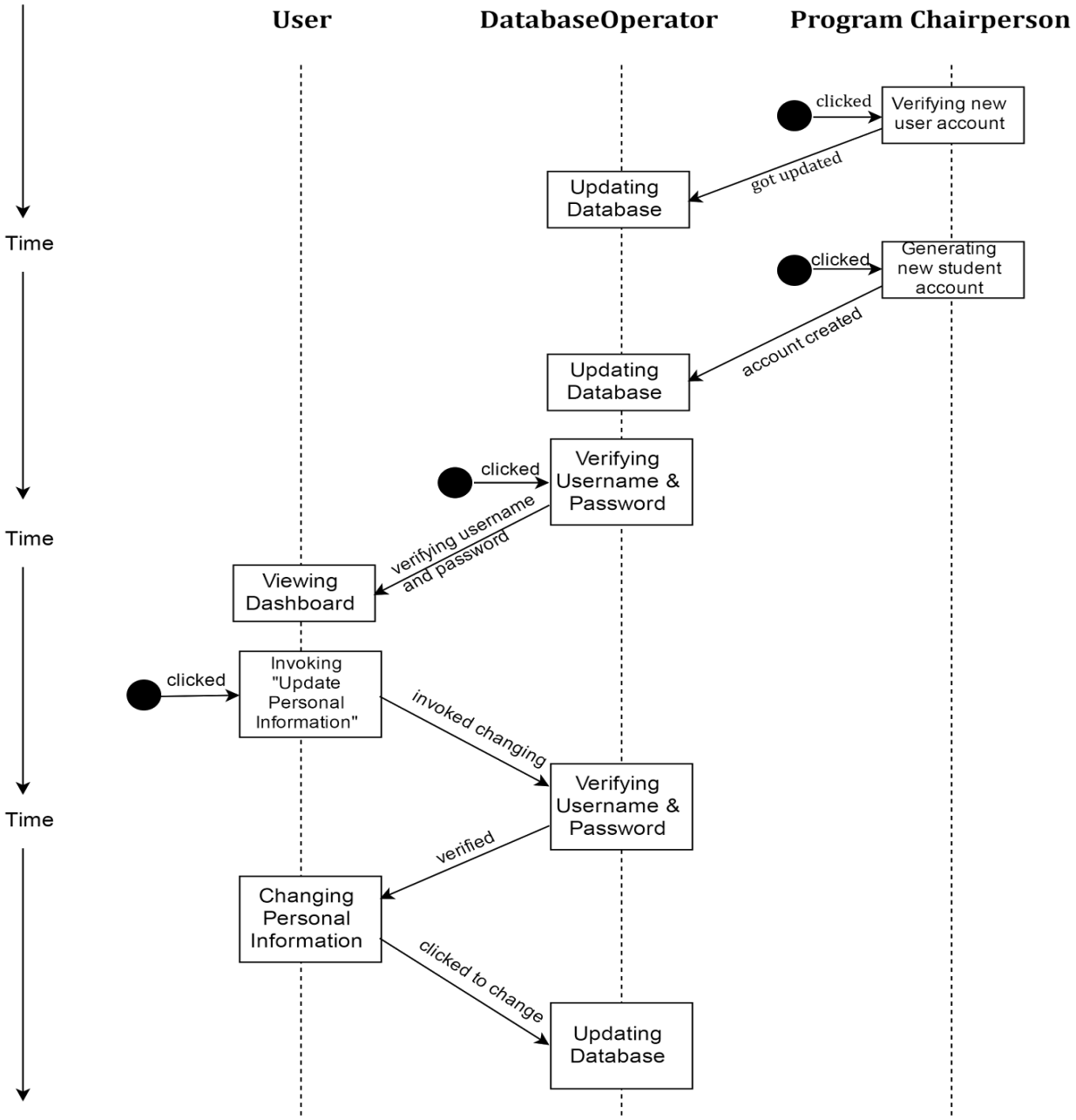


Figure 7.3.1: Sequence Diagram for Authentication System

7.3.2 Admission System

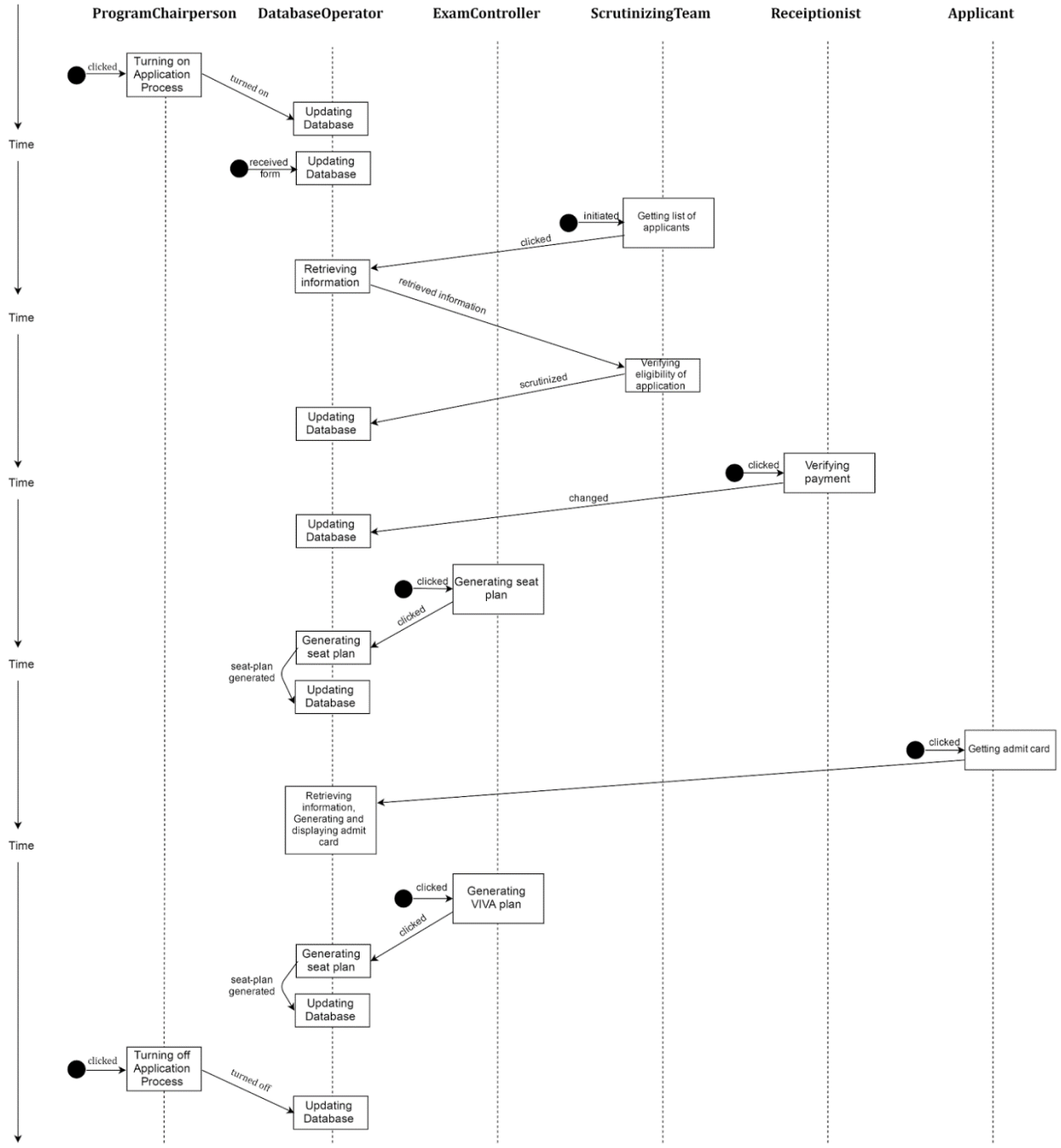


Figure 7.3.2: Sequence Diagram for Admission System

7.3.3 Result Generation System

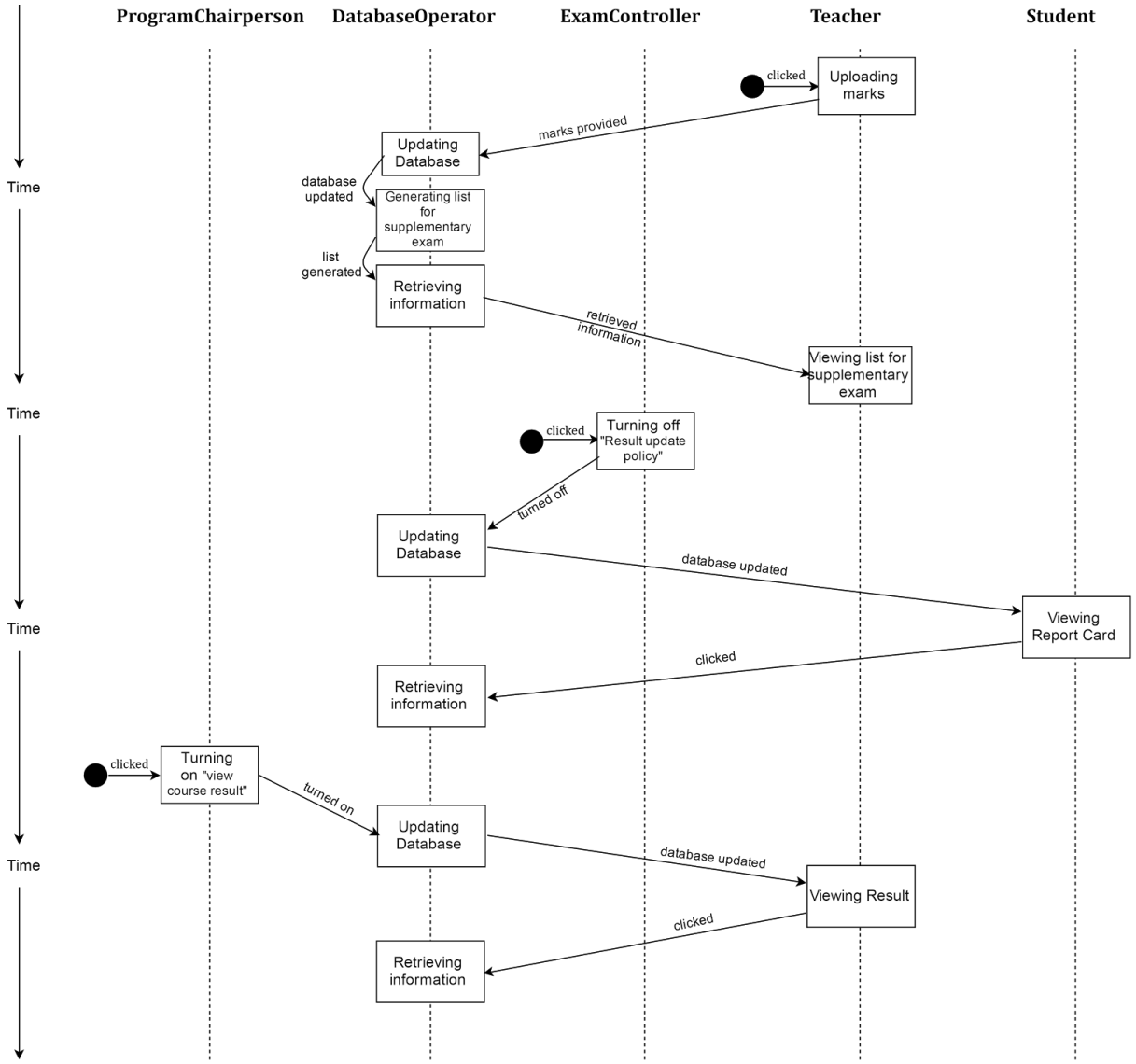


Figure 7.3.2: Sequence Diagram for Admission System

Chapter 8: Conclusion

We are pleased to submit the final SRS report on PGDIT Automation System. From this report, the readers will get a clear and easy view of PGDIT Automation system. This Software Requirement Specification document can be used effectively to maintain software development cycle. It will be very easy to conduct the whole project using this SRS. Hopefully, this document can also help our junior BSSE batch students. We tried our best to remove all dependencies and make effective and fully designed SRS. We believe that reader will find it in order.