

Software Requirement Specification, Analysis and Design

Online Graduate Record Validation System

Course: SE 803

Submitted by:

BSSE 6th batch

Submitted to:

Dr. Md. Mahbubul Alam Joarder
Professor
IIT, University of Dhaka



LETTER OF TRANSMITTAL

15th October, 2017

Dr. Md. Mahbubul Alam Joarder

Professor

Institute of Information Technology (IIT)

University of Dhaka

Sir,

We have prepared the report on Software Requirements Specification of “**Online Graduate Record Validation System (OGRVS)**” for your approval. This report details the requirements we gathered for the project.

The primary purpose of this report is to summarize our findings from the work that we completed as our Software Requirements Specification and Analysis course project. This report includes the details of each step we followed to collect the requirements.

Sincerely yours,
BSSE 6th batch

Acknowledgements

By the Grace of almighty Allah, we have completed the final report of “**Online Graduate Record Validation System (OGRVS)**”. We are grateful to our honorable Dr. Md. Mahbubul Alam Joarder Sir for his supervision throughout the working time. He helped us a lot by sharing his valuable knowledge to complete this document.

Table of Contents

LETTER OF TRANSMITTAL	i
Acknowledgements.....	ii
Chapter 1.....	1
Introduction	1
Chapter 2.....	2
Inception	2
2.1 Introduction	2
2.1.1 Identifying Stakeholders	2
2.1.2 Recognizing multiple viewpoints	3
2.1.3 Working towards collaborations.....	4
2.2 Conclusion.....	4
Chapter 3.....	5
Elicitation	5
3.1 Introduction	5
3.2 Eliciting Requirements	5
3.2.1 Collaborative Requirement Gathering.....	5
3.2.2 Quality Function Deployment.....	5
3.3 User Scenario	7
3.4 Conclusion.....	8
Chapter 4.....	9
Scenario Based Modeling.....	9
4.1 Definition of Use case	9
4.2 Use Case Diagrams.....	9
4.2.1 System Description from Level-0 use case:	9
4.2.2 System Description from Level-1 use case:	11
4.2.3 System Description from Level-1.1 use case:	13
4.2.4 System Description from Level-1.1.1 use case:	15
4.2.5 System Description from Level-1.1.2 use case:	16
4.2.6 System Description from Level-2.1. a. use case:.....	17
4.2.6 System Description from Level-2.1. b. use case:	18
4.3 Activity and Swim lane diagram.....	19
	29

Chapter 5.....	31
Data Model	31
5.1 Data Modeling Concept	31
5.2 Data Objects.....	31
5.2.1 Identify Data Objects	31
5.2.3 Entity Relationship Diagram.....	34
Chapter 6.....	35
Class-Based Model	35
6.1 Introduction	35
6.2 Class Diagram.....	35
Chapter 7.....	36
Flow-Oriented Model.....	36
7.1 Introduction	36
7.2 Data flow diagram.....	36
Chapter 8.....	44
Behavioral Model.....	44
8.1 Introduction	44
8.2 Sequence Diagram	45
Chapter 9.....	46
Software Design Architecture	46
9.1 Architectural Design for OOP:.....	46
9.1.1 Representing the system in context	46
9.1.2 Define Archetypes.....	47
9.1.3 Refining architecture into components	47
9.1.4 Describing Instantiation of the system	49
Chapter 10.....	50
Component Level Design	50
10.1 Analysis Class	50
10.2 Infrastructure Class	55
10.3 Elaborated Design Class	55
10.3.1 Collaboration Details.....	58
10.3.2 Appropriate Interfaces.....	59
10.3.3 Elaborate Attributes.....	59

10.3.3 Describe Processing Flow.....	62
10.4 Persistent Data.....	64
10.5. Develop or elaborate behavioral representation for a class or component	64
10.6. Elaborative Deployment	68
Chapter 11.....	69
User Interface design	69
11.1 Introduction	69
Chapter 12.....	81
Reference	81

Figure

Figure 4.1: Level-0 Use Case Diagram	10
Figure 4.2: Level-1 Use Case Diagram.....	12
Figure 4.3: Level-1.1 Use Case Diagram.....	14
Figure 4.4: Level-1.1.1 Use Case Diagram.....	15
Figure 4.5: Level-1.1.2 Use Case Diagram.....	16
Figure 4.6: Level-2.1. a. Use Case Diagram	17
Figure 4.7: Level-2.1. b. Use Case Diagram.....	18
Figure 4.8: Activity for Authentication.....	19
Figure 4.9: Swim lane for Authentication	20
Figure 4.10: Activity of Graduate Students.....	21
Figure 4.11: Swim lane for Graduate Students	22
Figure 4.12: Activity for System Admin.....	23
Figure 4.13: Swim lane for System Admin	24
Figure 4.14: Activity for Data Entry Operators	25
Figure 4.15: Swim lane for Data Entry Operators	26
Figure 4.16: Activity diagram for Registrar	27
Figure 4.17: Swim lane for Registrar	28
Figure 4.18: Activity for Foreign Universities.....	29
Figure 4.19: Swim lane for Foreign Universities	30

Figure 5.1: ER Diagram for data based modeling	34
Figure 6.1: Class Diagram.....	35
Figure 7.1: Level 1 Data Flow Diagram.....	37
Figure 7.2: Level 1.1 Data Flow Diagram	38
Figure 7.3: Level 1.2 Data Flow Diagram	39
Figure 7.4: Level 1.3 Data Flow Diagram	40
Figure 7.5: Level 1.4 Data Flow Diagram	41
Figure 7.6: Level 1.5 Data Flow Diagram	42
Figure 7.7: Level 1.6 Data Flow Diagram	43
Figure 8.1: Sequence Diagram	45
Figure 9.1: Representing the system in context	46
Figure 9.2: Refining archetype into component & classes	49
Figure 10.1: Design component of System Admin Class.....	52
Figure 10.2: Design component of System Admin Class.....	52
Figure 10.4: Design Component of Registrar Class.....	53
Figure 10.5: Design Component of Program Office	54
Figure 10.6: Design Component of UGC Stuff Class.....	54
Figure 10.7: Elaborated Design of System Admin Class.....	55
Figure 10.8: Elaborated Design of Student Class	56
Figure 10.9: Elaborated Design of Registrar Class	56
Figure 10.10: Elaborated Design of Program Office Class	57
Figure 10.11: Elaborated Design of UGC Class.....	57
Figure 10.12: Collaboration Details of System Admin Class	58
Figure 10.13: Collaboration Details of Program Office	58
Figure 10.14: Collaboration Details of Registrar Class	59
Figure 10.15: Processing Flow of Verification System	63
Figure 10.16: State chart fragment for “System Admin”	64
Figure 10.17: State chart fragment for “Student”	65
Figure 10.18: State chart fragment for “Registrar”	66
Figure 10.19: State chart fragment for “Program Office	67
Figure 10.20: Elaborative Deployment of OGRVS.....	68
Figure 11.1: Start window	69
Figure 11.2: Login window	70

Figure 11.3: Search Student window	71
Figure 11.4: System Admin dashboard window	72
Figure 11.5: Add User window.....	73
Figure 11.6: UGC Staff dashboard.....	74
Figure 11.7: Add University window.....	75
Figure 11.8: University Register dashboard.....	76
Figure 11.9: Add Course window.....	77
Figure 11.10: Verification window.....	78
Figure 11.11: Add Result window	79
Figure 11.12: Student dashboard	80

Chapter 1

Introduction

This chapter describes the objectives of this report as well as the audiences who should have to go through this report for individual purposes.

1.1 Purpose

This project is based on the requirements analysis for graduate verification system. It includes all over requirements to develop this system no matter whether they are functional or nonfunctional. The information about the requirements here have been organized well so that everyone can easily figure out a summarized concept about graduate verification system. The SRS serves as the 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.

The purpose of this project is to verify graduate students. This system will be used for validating graduate students of Institute of Information Technology of University of Dhaka. Here the end users are UGC staffs (Admin, Staff 1, Staff 2), University staffs (Registrar, Data entry operator 1, Data entry operator 2), Graduate students, Stakeholders (Foreign universities, Employers), Payment processing partners.

1.2 Intended Audience

This SRS is intended for several audiences, including the customer 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 for the customer.
- The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the development team is on track during development of the system.
- The designers will use this SRS as a guideline for creating the system's design.
- The designers will continuously refer back to their 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 testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are being completed, the testers will run their tests on the software to ensure that the software fulfills the requirements documented on the SRS. The testers will again run their tests on the entire system when it is completed and more use of that all requirements documented in the SRS have been fulfilled.

Chapter 2

Inception

In this chapter, we will discuss about the first step of Software Requirements Specifications Analysis, that is, Inception.

2.1 Introduction

Requirement Engineering comprises several sequential steps. Inception is the first one among them. Inception creates the entrance to the project for the requirements analysts. It refers them how the project should get started. It also provides a basic idea to the engineers about the problems ahead which are needed to be solved and how critical obstacles may come during the project. The main target of Inception phase is to identify the people related to the project and their needs. In order to complete this phase, we have focused on:

- Identifying Stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Asking the first questions

2.1.1 Identifying Stakeholders

Stakeholders are entities that have an interest in a given project. These stakeholders may be inside or outside an organization which:

1. Sponsor a project, or
2. Have an interest or a gain upon a successful completion of a project,
3. May have a positive or negative influence in the project completion

Stakeholders are given below:

- UGC staff (Admin, staff 1, staff 2)
- University staff (Registrar, data entry operator)
- Graduate students
- Foreign university
- Payment processing partner

1. UGC staff: UGC staff can login into the system with email address, password. He/she can add university (name, location, website), university's registrar with his/her information. UGC staff can generate report with all information (University wise, payment wise).

2. **University staff:** University staff (Registrar) will login into the system with his/her email address and password. In Registrar profile, he/she can view his/her personal information (Name, designation, university name), add, edit and delete student information, and add digital signature to verify graduate student information, which are pending for approval. Registrar can generate reports that belong to his/her university.

3. **Graduate student:** They are the main beneficial users of this system.

4. **Foreign university:** Foreign universities or employers can access the system to validate their candidate students' information. First, they have to enter student's name, registration number, university name, mobile number and email address. If students' information is available in the system, then stakeholder need to provide his/her credentials- name, designation, organizations' name, contact number and email.

5. **Payment processing partner:** They assist the system through processing the payment system.

2.1.2 Recognizing multiple viewpoints

Each stakeholder has his/her own viewpoint. I have talk to them to understand those viewpoints.

1. UGC staff viewpoints:

- He/she can add university (name, location, website), university's registrar with his/her information.
- UGC staff can generate report with all information (University wise, payment wise).

2. University staff viewpoints:

- He / she can view his/her personal information (Name, designation, university name), add, edit and delete student information.
- He/she can add digital signature to verify graduate student information, which are pending for approval.
- Registrar can generate reports that belong to his/her university.

3. Graduate students' viewpoint:

- They can get benefits from the system.
- They can stay connected through using the system.

4. Foreign university's viewpoints:

- Can access the system to validate their candidate students' information.
- need to provide his/her credentials- name, designation, organizations' name, contact number and email.

5. Payment processing partners' viewpoints:

- They process the payment system when needs.

2.1.3 Working towards collaborations

Every stakeholder has their own requirements. We have followed following steps to merge these requirements:

- Identify the common and conflicting requirements
- Categorize the requirements
- Take priority points for each requirements from stakeholders and on the basis of this voting prioritize the requirements
- Make final decision about the requirements.

Common requirements:

- User friendly and efficient system
- Easy to use
- Authentication
- Database containing detailed information about users

Conflicting requirements:

We found some requirements conflicting each other. We had to trade-off between the requirements.

- Easy access
- Strong authentication and high security
- No harmful effects on existing technology

Final Requirements:

Error free system (Maximum 5% error may be considerable)

- Web based application
- Accessible via internet
- Allow the users login and logout
- Restrict access to functionality of the system based upon user roles
- Allow administrators of the system to change user types and configure parameters of the system
- User friendly and efficient system
- Secure way to communicate
- Database contains all information.

2.2 Conclusion

In this 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 stake-holders. More studies and communication will help both side (developer and client) to understand the future prospect of the project. We believe that the full functioning document will help us to define that future prospect.

Chapter 3

Elicitation

After discussing the inception part, we need to keep focus on the elicitation part. So this chapter specifies the elicitation part.

3.1 Introduction

Requirements elicitation is a part of requirement engineering that is the practice of gathering requirements from the users, customers, and other stakeholders. We have faced many problems like understanding the problems, problems of making questions for the stakeholders, problems of less communication with the stakeholders for time limitation, problems of volatility. Though it is not too easy to gather requirements within a very short time, we have surpassed these problems 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.

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- 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 students and discussed with teachers and professionals. They were questioned about their requirements and expectations from the Graduate record verification system
- They were asked about their Graduate record verification system problems.
- At last we selected our final requirement list from the meetings.

3.2.2 Quality Function Deployment

Quality function deployment (QFD) known as a way to represent the “voice of the customer,” is a process for capturing customer requirements and translating them into requirements that can be used by designers, producers, and suppliers. It concentrates on maximizing customer satisfaction

from the Software engineering process. With respect to our project the following requirements are identified by a QFD.

3.2.2.1 Normal Requirements

Normal requirements consist of objectives and goals that are stated during the meeting with the customers. Normal requirements of our project are:

1. Allow valid user to login and logout
2. Check user validity
3. Feature that will allow dynamic report generation
4. Allow users to generate academic transcript and certificate
5. Efficient and user friendly
6. The user interface of the system would be easy
7. Allow user to view approval of the admin panel
8. Allow user to give approval/rejection (by registrar)
9. Allow user to search Information of students
10. Allow user to track information by using activity log
11. Security issue
12. Error free activity

3.2.2.2 Expected Requirements

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause of dissatisfaction. These are –

1. The application shall allow the user to log in based upon a user id and password.
2. The application shall automatically save the current date.
3. The application will allow the users to search, update or delete.
4. The user interface of the system shall be easy to use and shall make the least use of typing by users.
5. The application shall be easily maintainable.
6. The application will be stable.
7. The application will be open for future extension and modification.
8. All the records will be saved to database for future references.
9. The application shall be user friendly
10. The user interface shall make use of input such as drop downs, check boxes and radio buttons as much as possible to avoid invalid and incorrect input.

3.2.2.3 Exciting Requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. These are

1. The system will enable the user to change user passwords if they forget old ones.
2. The user interface should provide appropriate error messages for invalid input as well as tool-tips
3. The system will have activity log for tracking information.

3.3 User Scenario

Online Graduate Record Verification System

This system will be used for validating graduate students of Institute of Information Technology of University of Dhaka. Here the end users are UGC staffs (Admin, Staff 1, Staff 2), University staffs (Registrar, Data entry operator 1, Data entry operator 2), Graduate students, Stakeholders (Foreign universities, Employers), Payment processing partners.

System admin can add information (name, email address, mobile number, password, and role) about Admin, Registrar, Program officer of Institute/Department, Students, Stakeholder and assign their roles (Admin, Registrar, Program officer of Institute/Department, Students, Stakeholder). System admin can do all operations those defined for other users.

System admin can view activity log, generate report of all things. Activity log keeps track of information change.

(Student profile information depend on Registrar input) Student can log in to the system with his/her email address, password, and registration number. After logging in into the system, he/she can view his/her profile with personal information, academic information, and payment request from stakeholders. Personal information is name, email address, mobile number, present address, permanent address. Academic information is registration number, session, year, institute/department, university name. If any payment request is received, then he/she can pay through external system (PayPal). Response from external system will be recorded in database. Student can edit his/her personal information.

UGC staff can login into the system with email address, password. He/she can add university (name, location, website), university's registrar with his/her information. UGC staff can generate report with all information (University wise, payment wise).

Registrar will login into the system with his/her email address and password. In Registrar profile, he/she can view his/her personal information (Name, designation, university name), add, edit and

delete student information, and add digital signature to verify graduate student information, which are pending for approval. Registrar can generate reports that belong to his/her university.

Institute/Department program officer login into the system with email address, password. In PO profile, he can view his personal information. PO can define number of semester, add course in corresponding semester. PO can insert, update and delete students' semester wise course results. PO can generate reports that belong to his/her institute/department.

Foreign universities or employers can access the system to validate their candidate students' information. First, they have to enter student's name, registration number, university name, mobile number and email address. If students' information is available in the system, then stakeholder need to provide his/her credentials- name, designation, organizations' name, contact number and email. After inserting that information, a request will be sent to the corresponding student as a payment request. If the payment is completed, a notification will be sent to Registrar to attach digital signature (signature image) to verify that student's information. After completing verification, a link will be sent to stakeholders' email address. Through this link, stakeholder get student has verified information.

3.4 Conclusion

Elicitation phase helped us to understand about the problems of our scopes of the system. This phase also helped us to identify the requirements, negotiate different approaches and specify a preliminary set of solution requirements in an atmosphere that is conducive to the accomplishment of the goal.

Chapter 4

Scenario Based Modeling

This chapter describes the scenario based model.

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. Every user has one or more goals when using system.

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-scenario we found five 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.

They all will be our users, who will be using our system to fulfill their procedures.

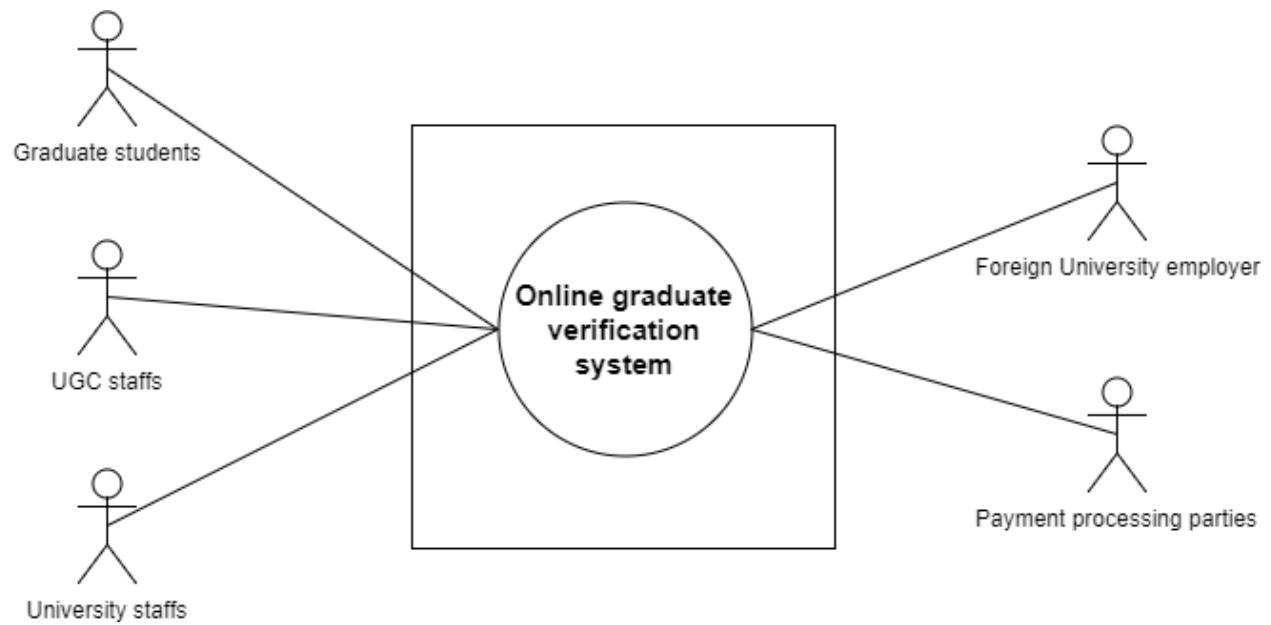


Figure 4.1: Level-0 Use Case Diagram

4.2.2 System Description from Level-1 use case:

Primary Actor : User
Secondary Actor : System
Goal in Context : To operate the application.
Scenario : The actors of our system have to play different actions and system will reply according to these actions –

Action 1: Enters signup.

Reply 1: Please fill up the required information.

Action 2: Enters the information

Reply 2: Registration successful.

Action 3: Enters username and password.

Reply 3: Sign in successful and shows User Dashboard.

Different user's action and reply (UGCs, Student, Registrar, Foreign University, Program Officer)

Action: Click add University (UGCs)

Reply: University added (UGCs)

Action: Click add Registrar (UGCs)

Reply: Registrar added (UGCs)

Action: Click add student (Registrar)

Reply: Student added (Registrar)

Action: Click add department (Registrar)

Reply: Department added (Registrar)

Action: Request for student verification (Foreign University)

Reply: Student verified

Action: Click to view student info (Student)

Reply: Show students details information (Student)

Action: Check payment request (Student)

Reply: Show the availability of the request (Student)

Action: Enter report generation (PO, UGCs)

Reply: Report generated (PO, UGCs)

Exception: No Exceptions (if system works correctly by fulfilling requirements)

Exceptions (if) : User is not authenticated.
 : System error.

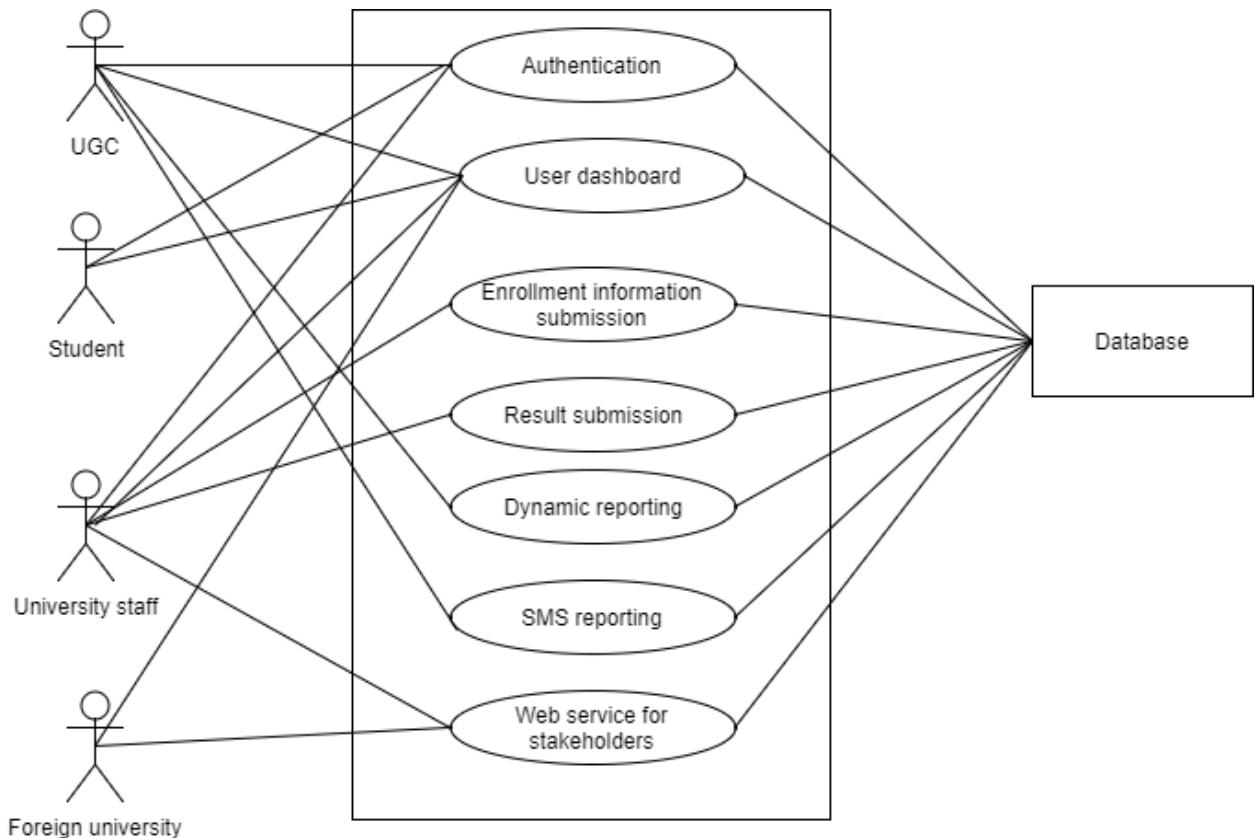


Figure 4.2: Level-1 Use Case Diagram

4.2.3 System Description from Level-1.1 use case:

Action 1:

- Enters First Name
- Enters Last Name
- Enters email.
- Enter password
- Clicks on **Save**.

Reply 1: Successfully saved.

Action 2: Enters **Change Password**.

Reply 2: Prompts for enter previous password.

Action 3: Enters previous password.

Reply 3: Prompts for new password.

Action 4:

- Enters new password.
- Re-enters new password.
- Enters **Save Changes**.

Reply 4: Password successfully changed.

Action 5: Enters sign out.

Reply 3: Logged out user.

Exception: No Exceptions

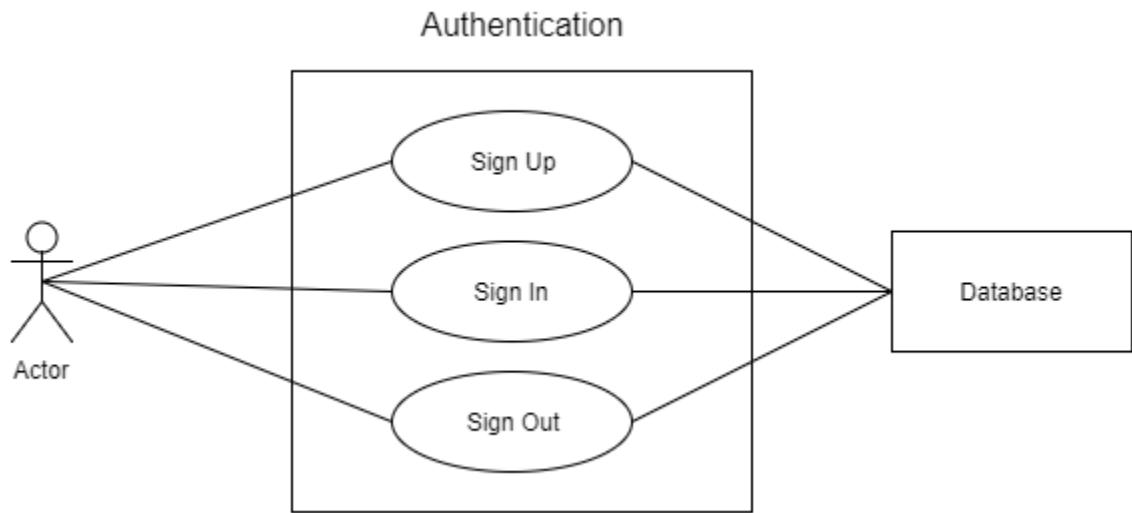


Figure 4.3: Level-1.1 Use Case Diagram

4.2.4 System Description from Level-1.1.1 use case:

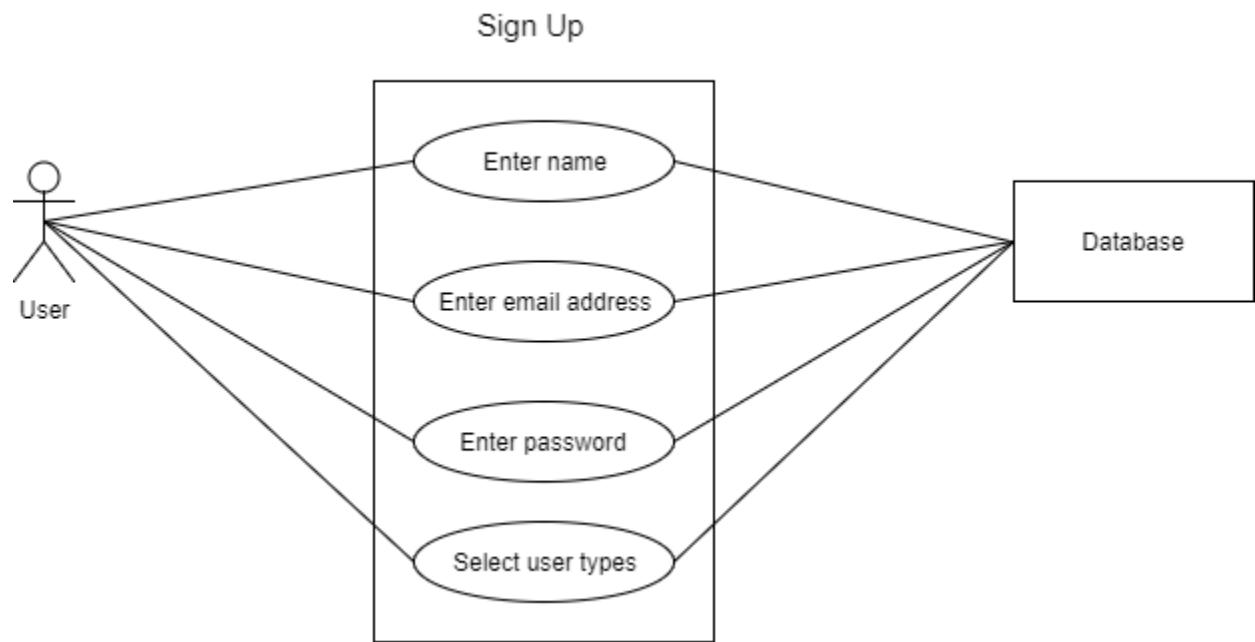


Figure 4.4: Level-1.1.1 Use Case Diagram

4.2.5 System Description from Level-1.1.2 use case:

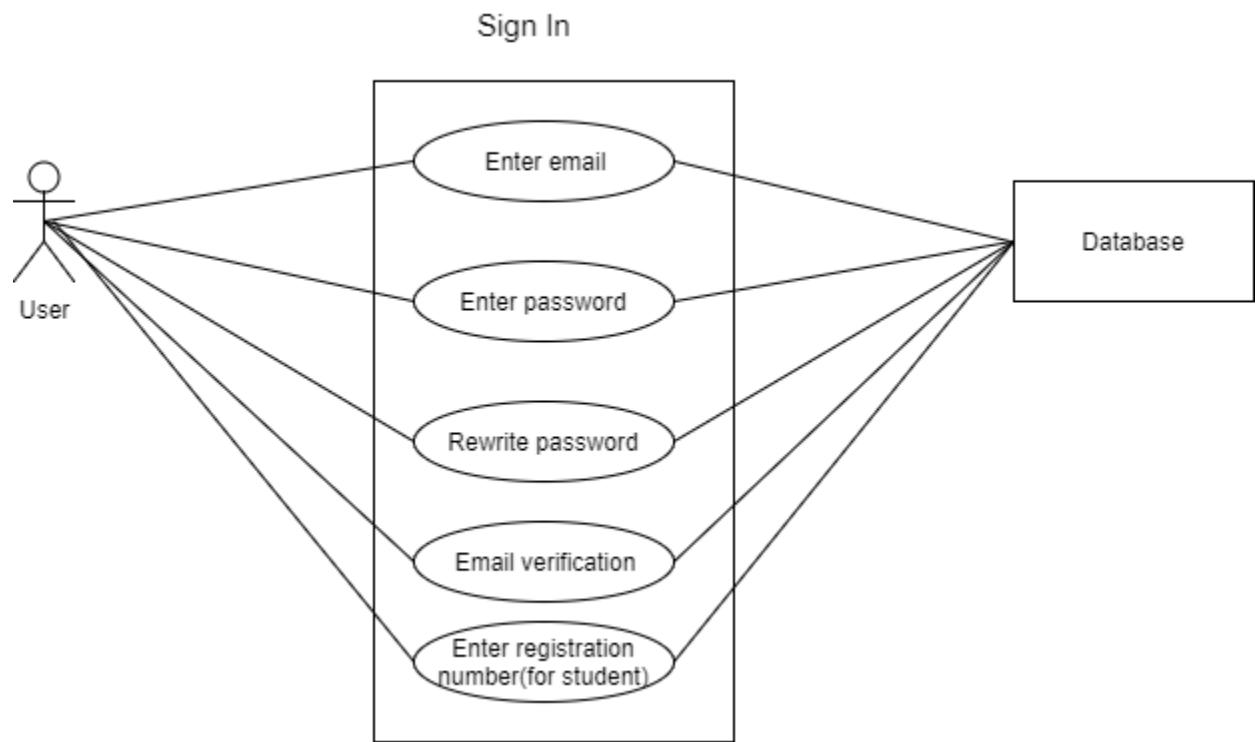


Figure 4.5: Level-1.1.2 Use Case Diagram

4.2.6 System Description from Level-2.1. a. use case:

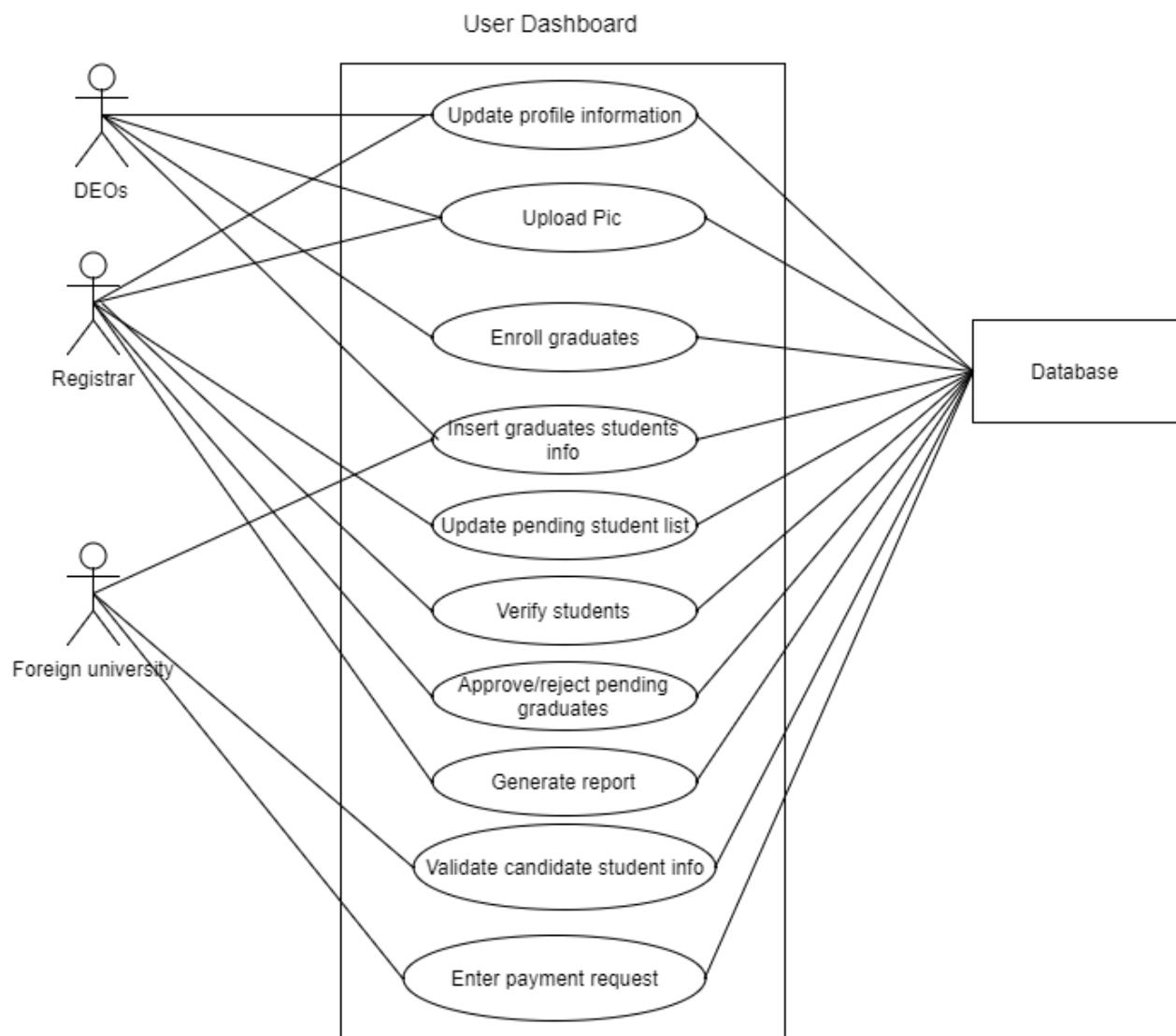


Figure 4.6: Level-2.1. a. Use Case Diagram

4.2.6 System Description from Level-2.1. b. use case:

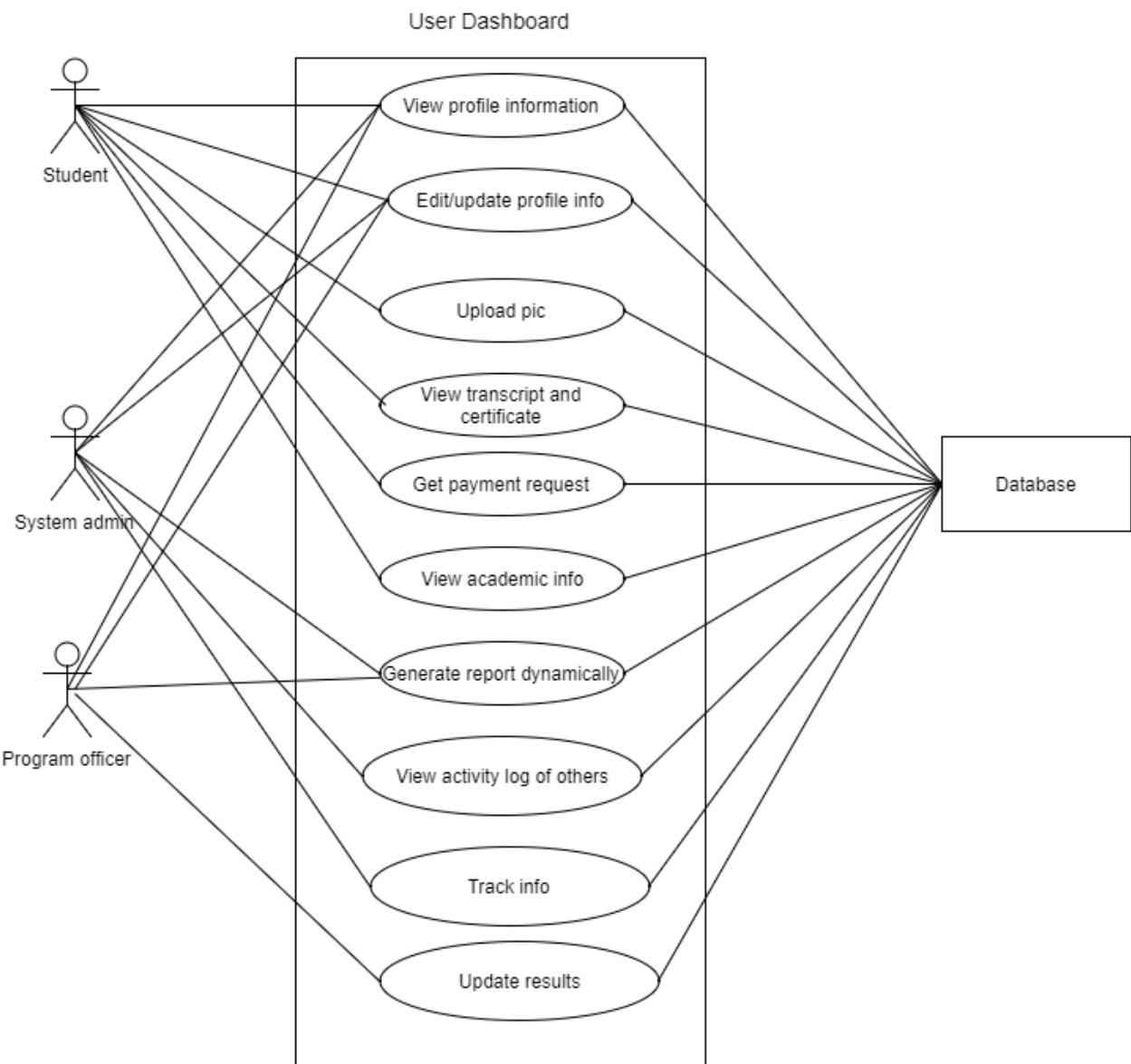


Figure 4.7: Level-2.1. b. Use Case Diagram

4.3 Activity and Swim lane diagram

Activity diagram shows the technical view of the system for every use case from which we can understand how the system actually works and how the actors interact with the system. Here is the activity diagram for our authentication system.

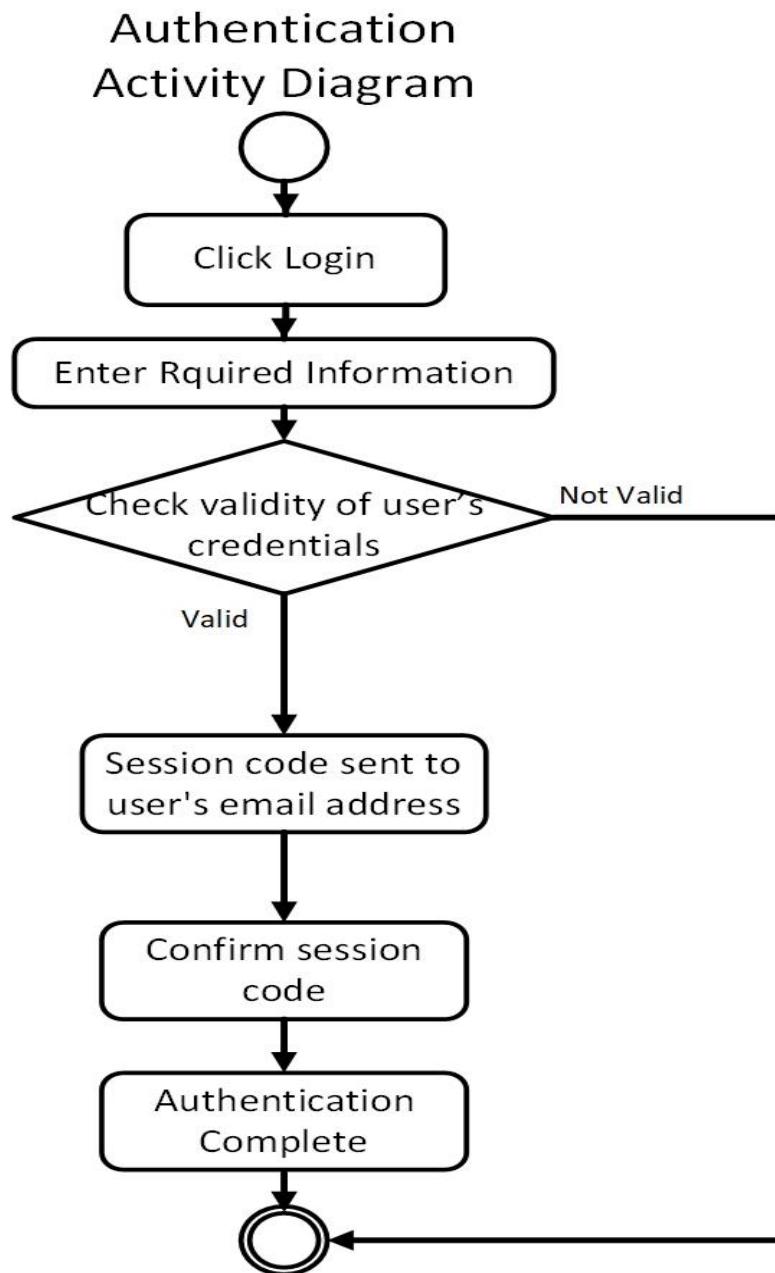


Figure 4.8: Activity for Authentication

Swim lane diagram of a specific activity diagram shows the responsibilities of each actor dividing them into lanes. From this diagram we can improve our understanding about how the system works and which actors play what role.

Following is the swim lane for authentication system.

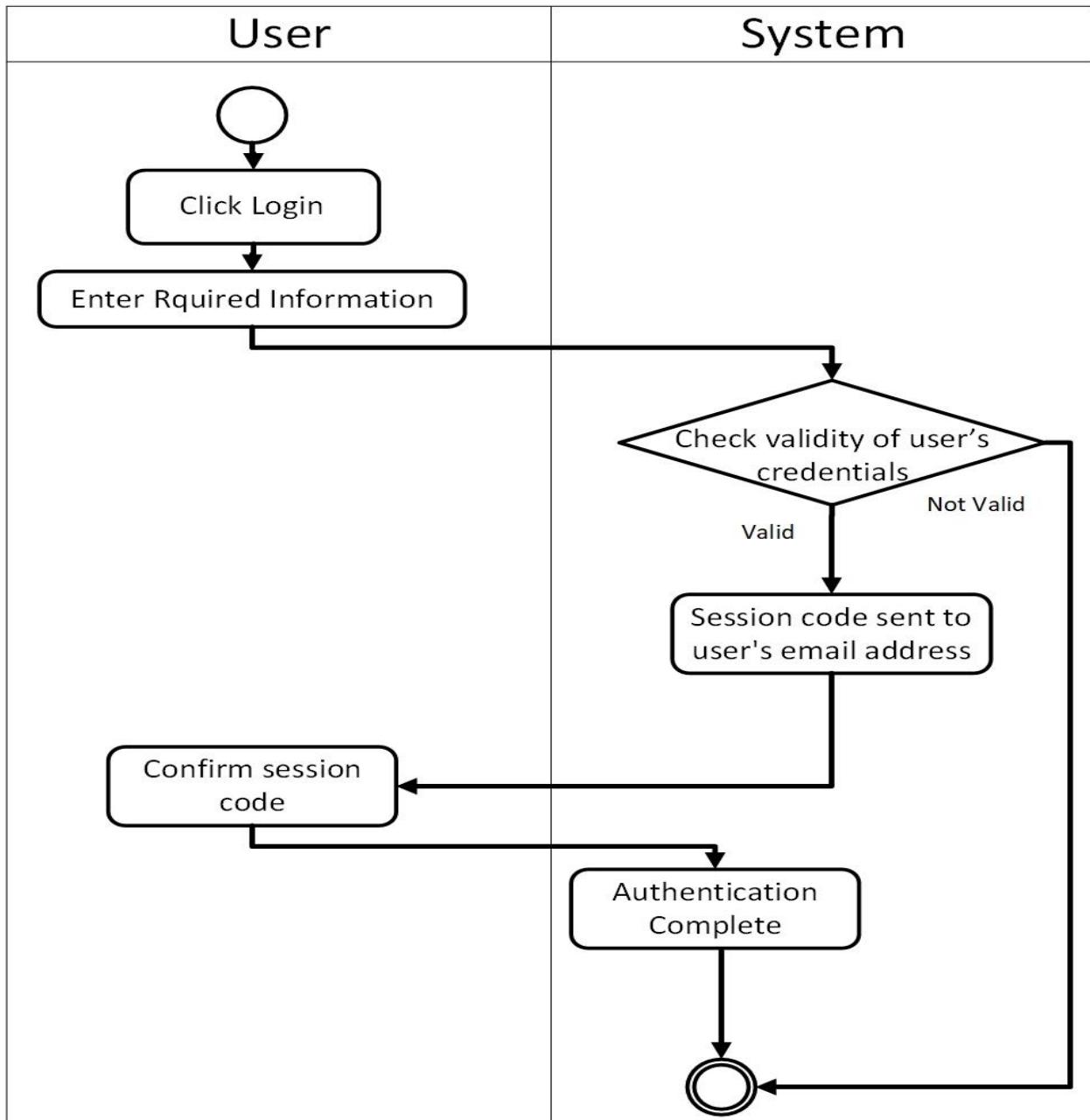


Figure 4.9: Swim lane for Authentication

User: Graduate Student Activity Diagram

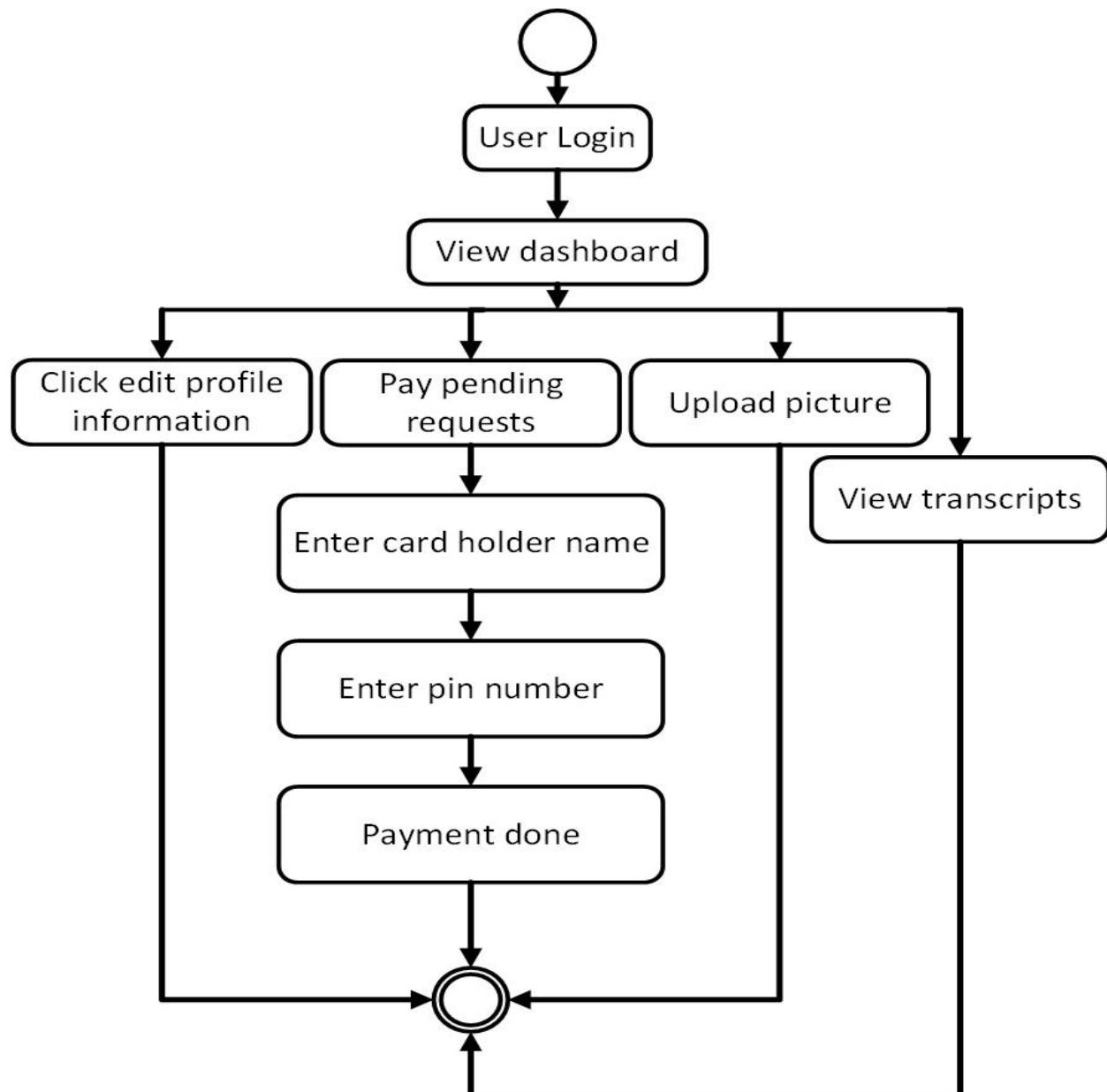


Figure 4.10: Activity of Graduate Students

Following is the swim lane diagram for user Graduate students.

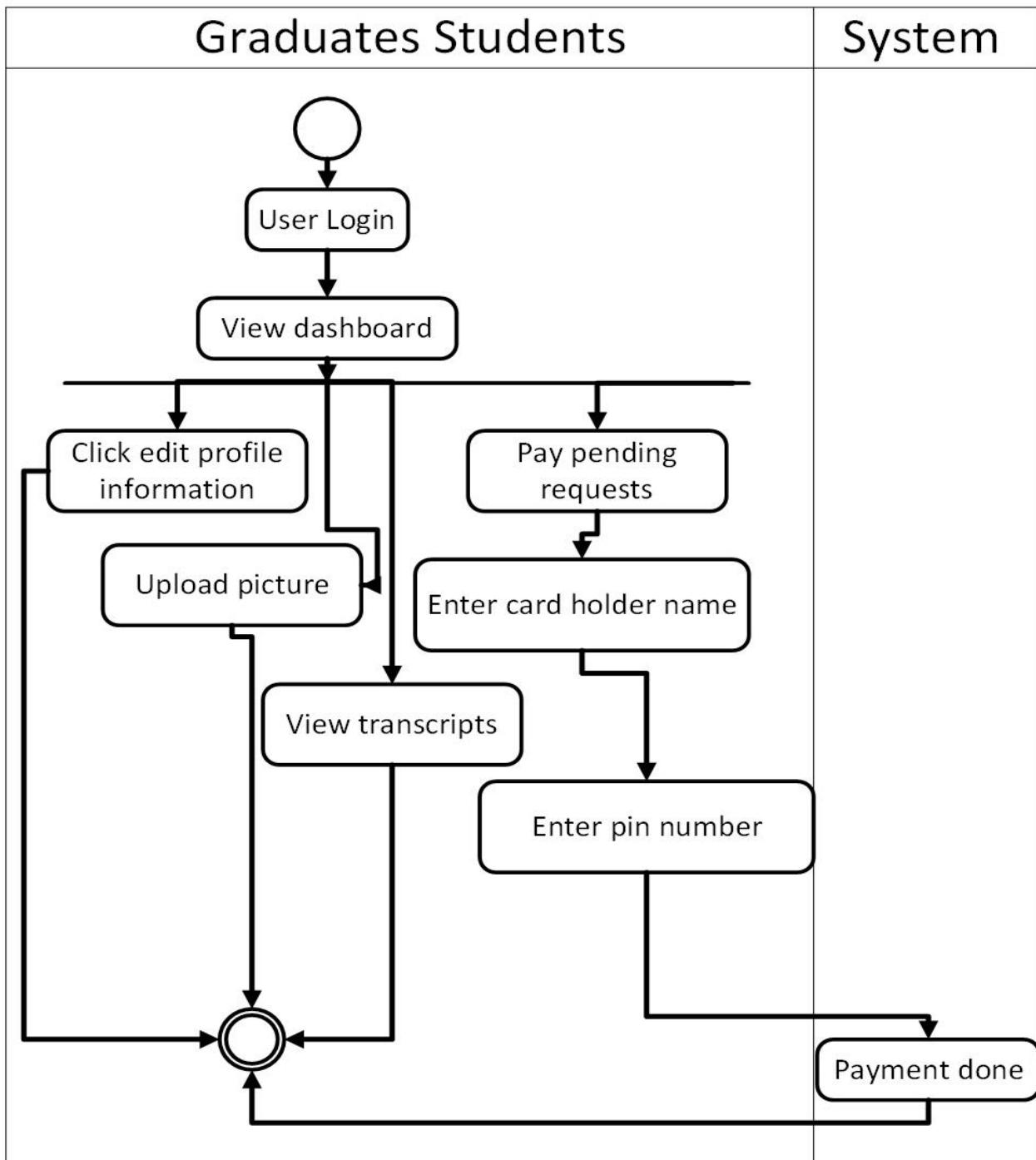


Figure 4.11: Swim lane for Graduate Students

Following is the activity diagram for System Admin.

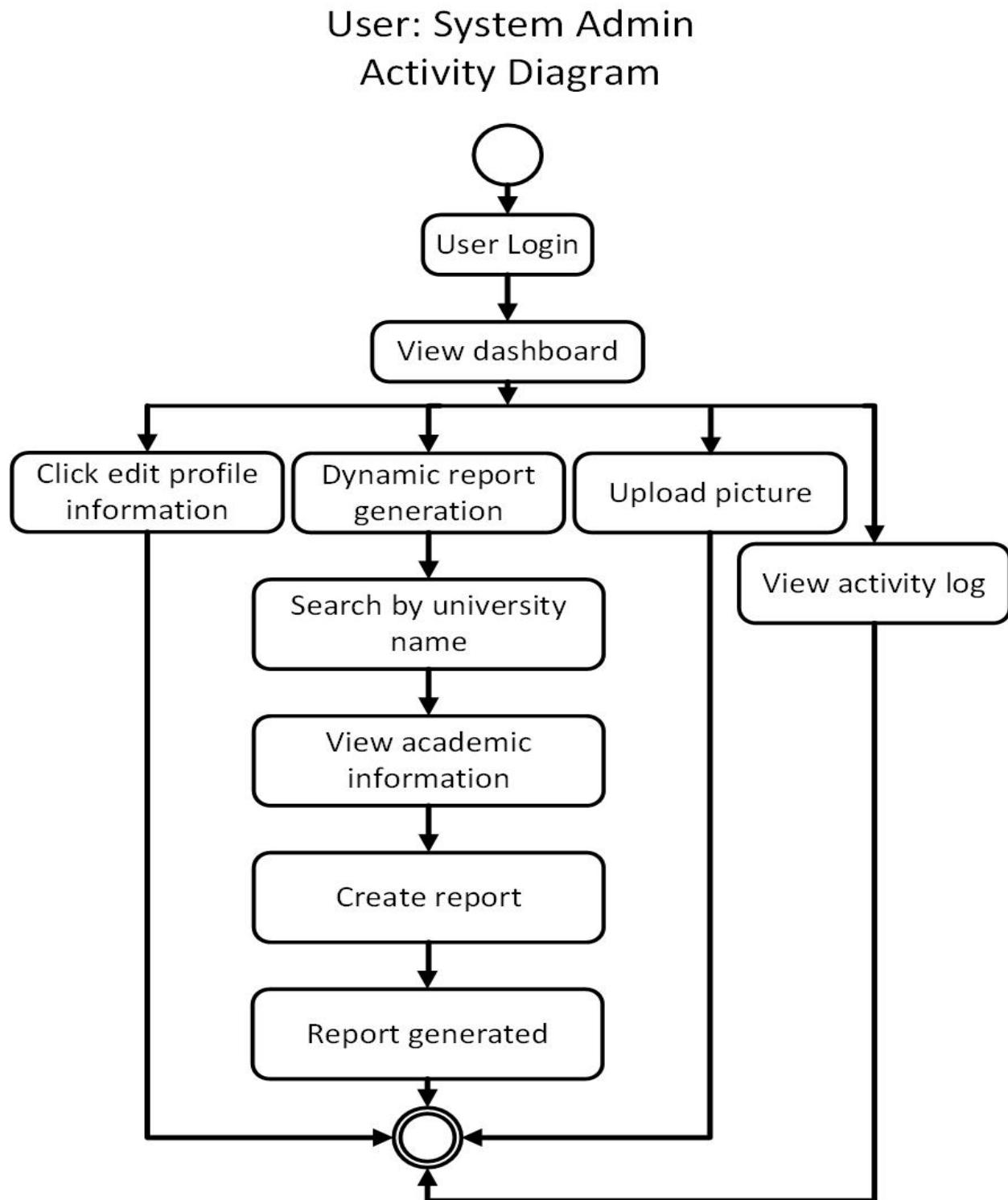


Figure 4.12: Activity for System Admin

Following is the swim lane diagram for System Admin.

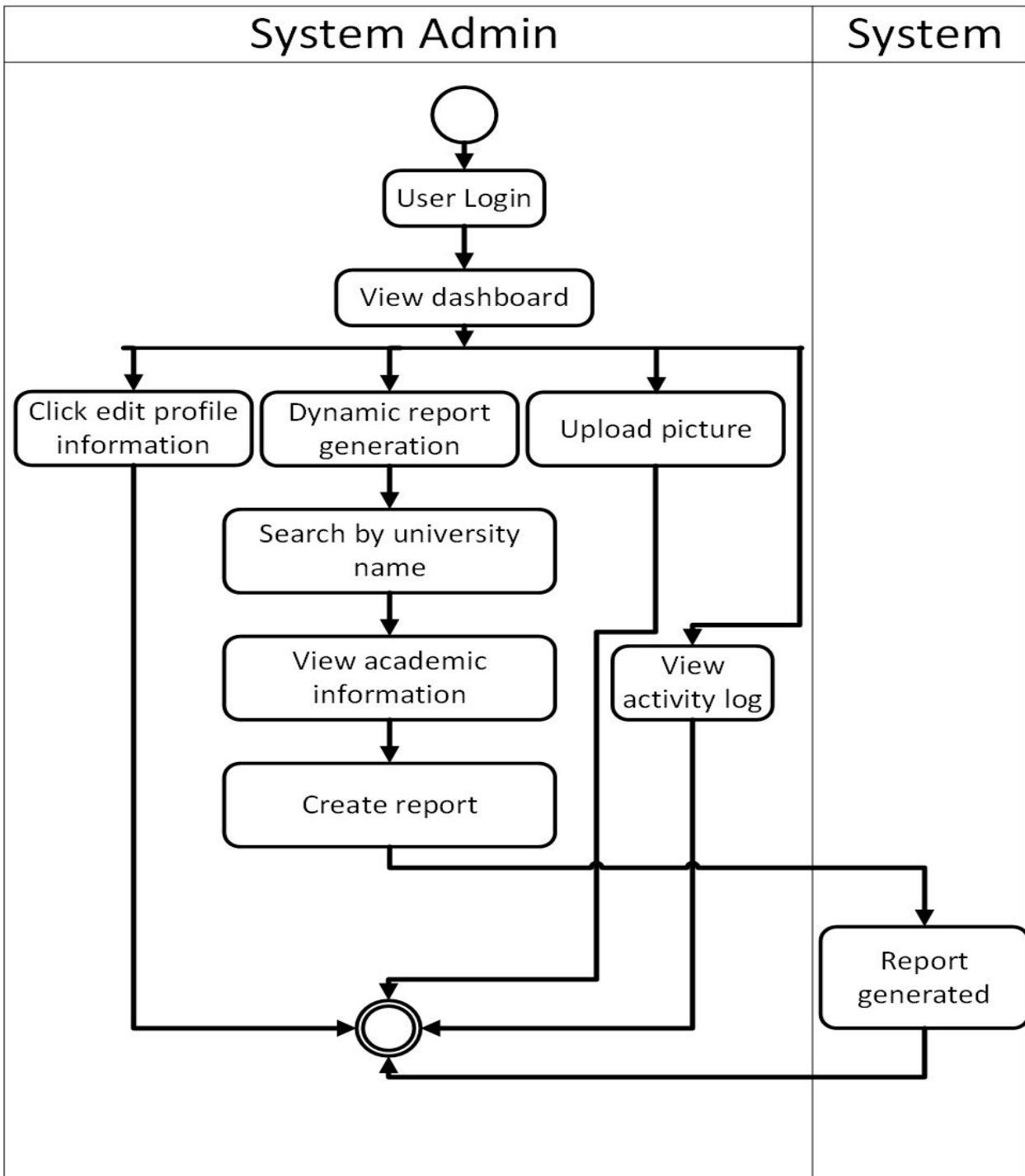


Figure 4.13: Swim lane for System Admin

Following is the activity diagram for Data Entry Operators.

User: Data Entry Operators Activity Diagram

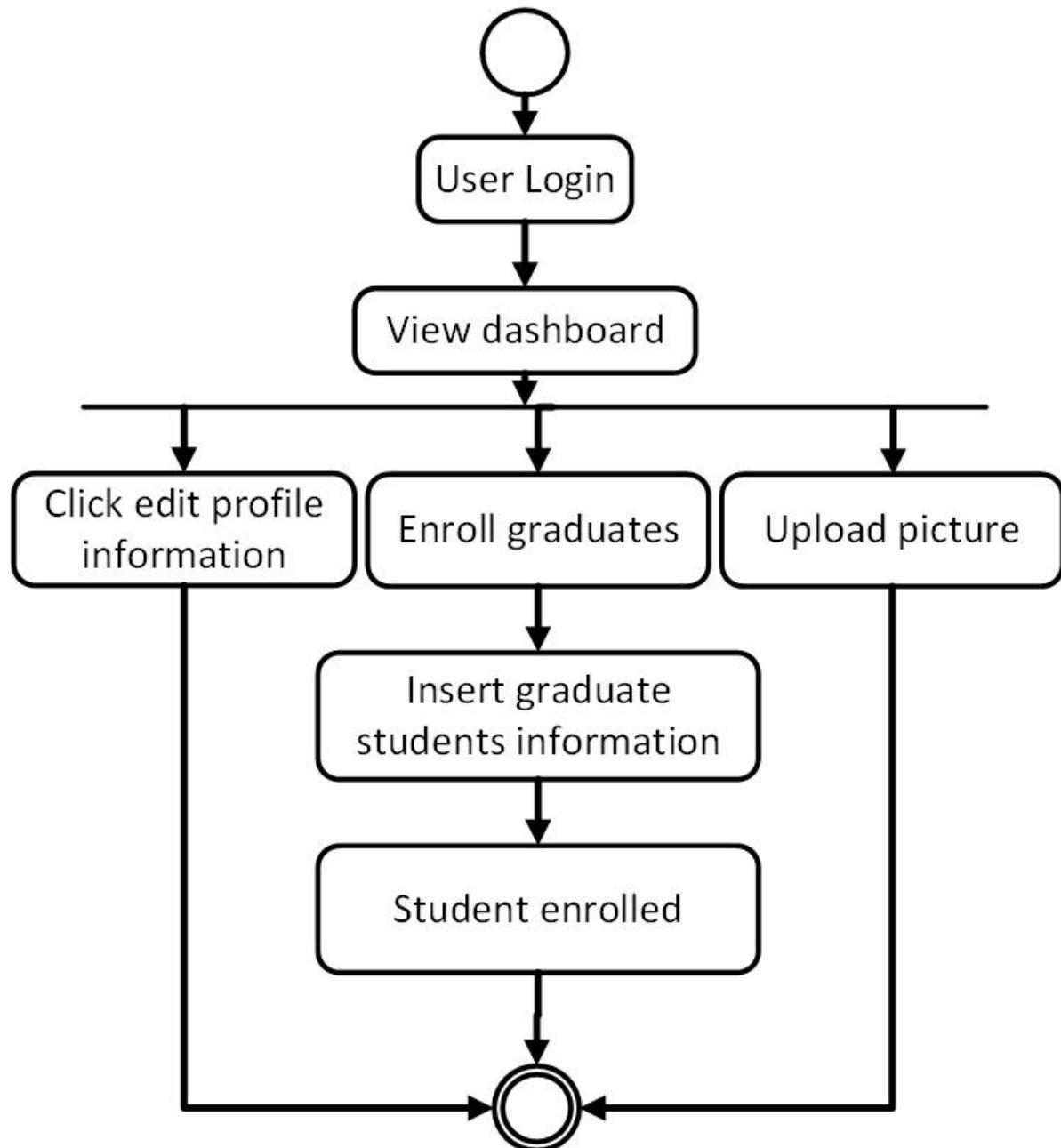


Figure 4.14: Activity for Data Entry Operators

Following is the swim lane for Data Entry Operators.

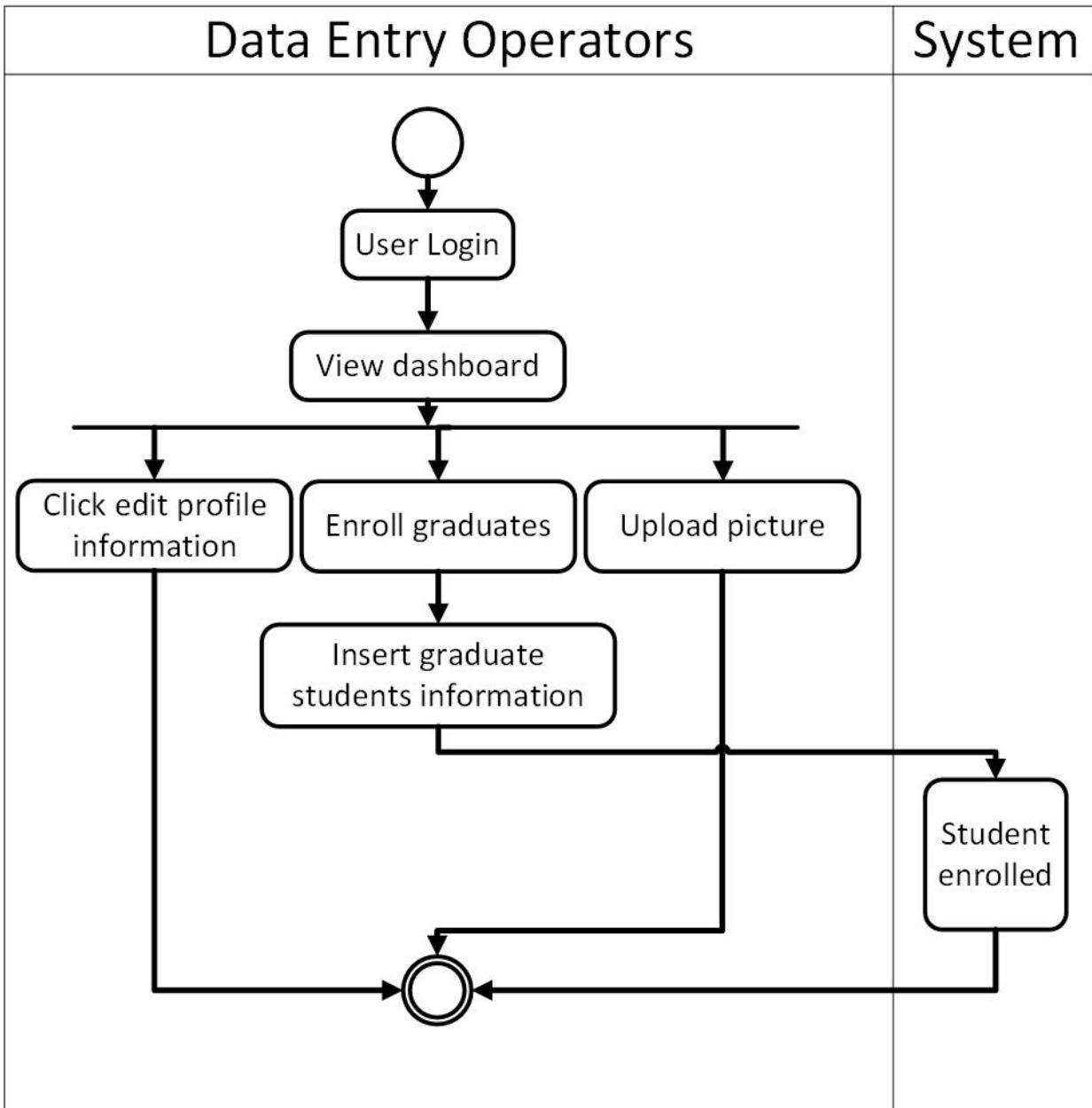


Figure 4.15: Swim lane for Data Entry Operators

Following is the activity diagram for Registrar.

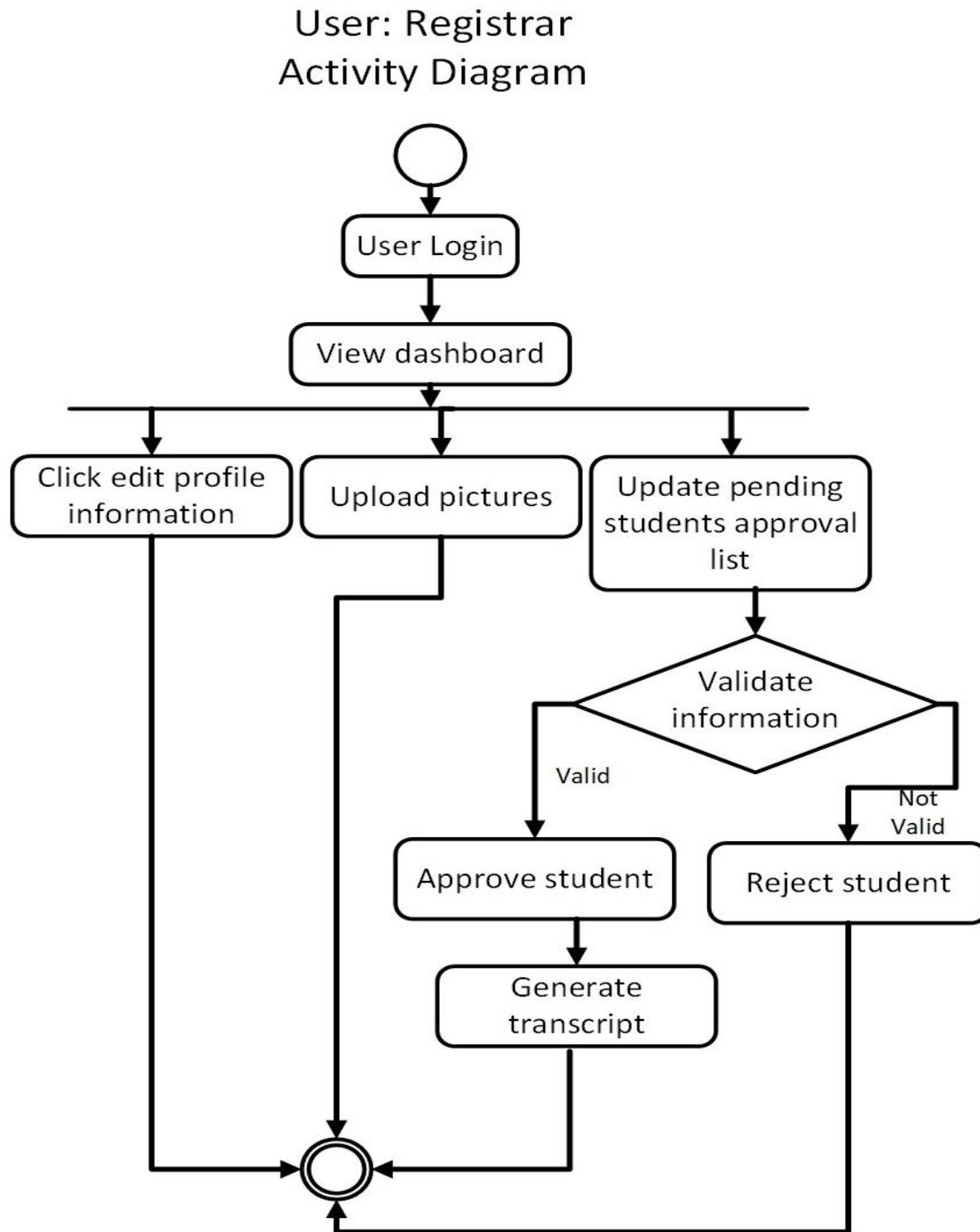


Figure 4.16: Activity diagram for Registrar

Following is the swim lane for registrar..

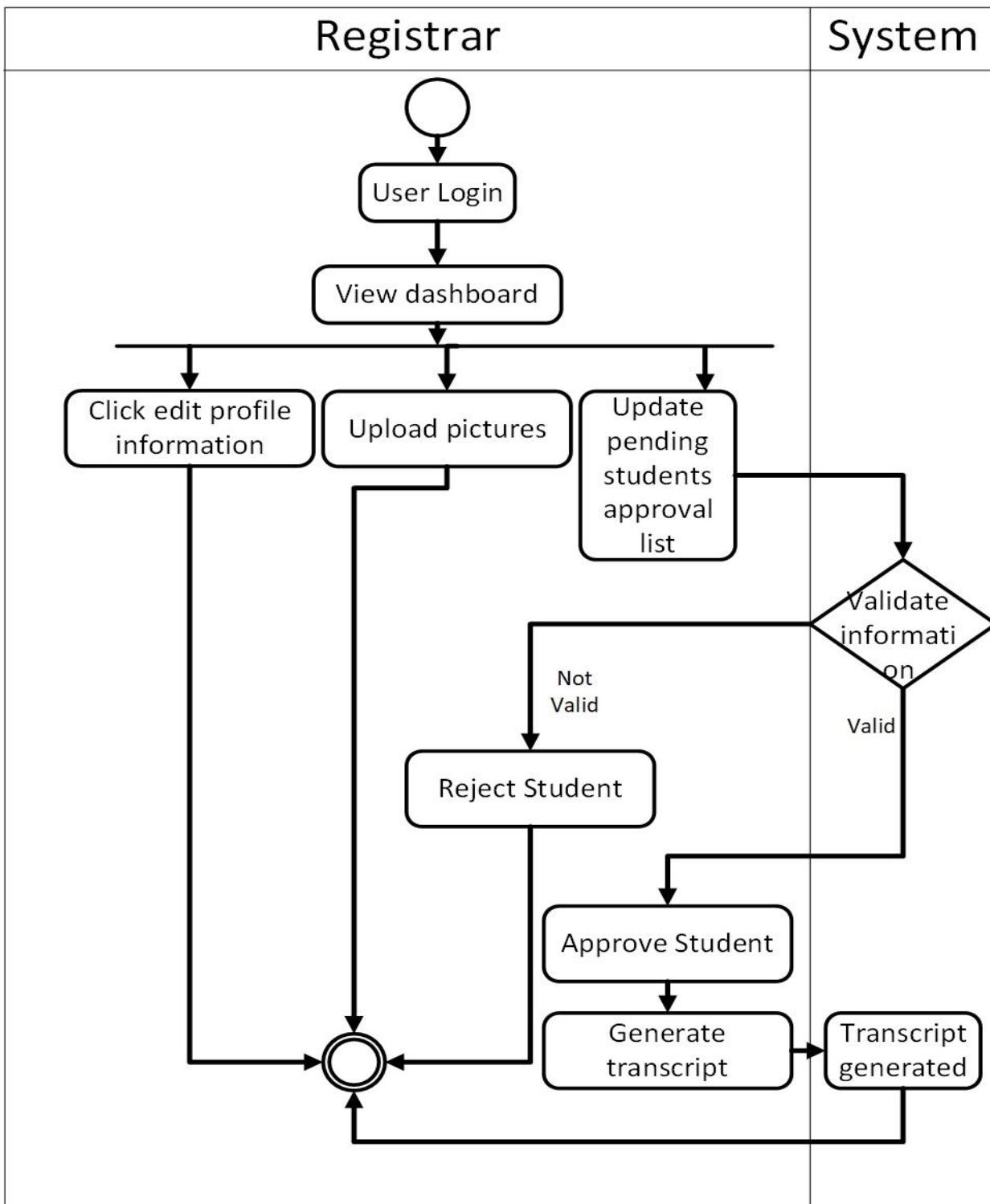


Figure 4.17: Swim lane for Registrar

Following is the activity diagram for foreign universities.

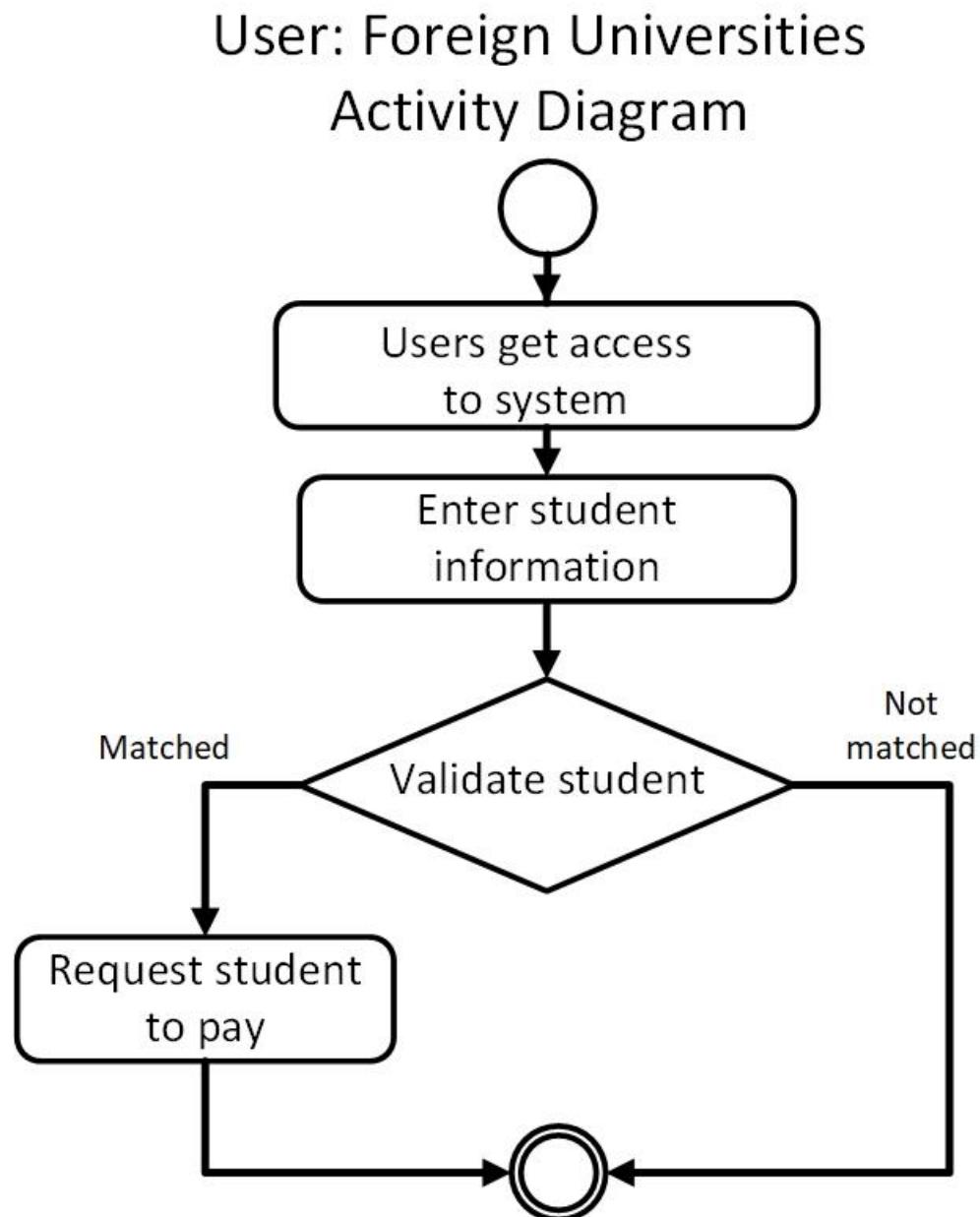


Figure 4.18: Activity for Foreign Universities

Following is the swim lane diagram for Foreign Universities.

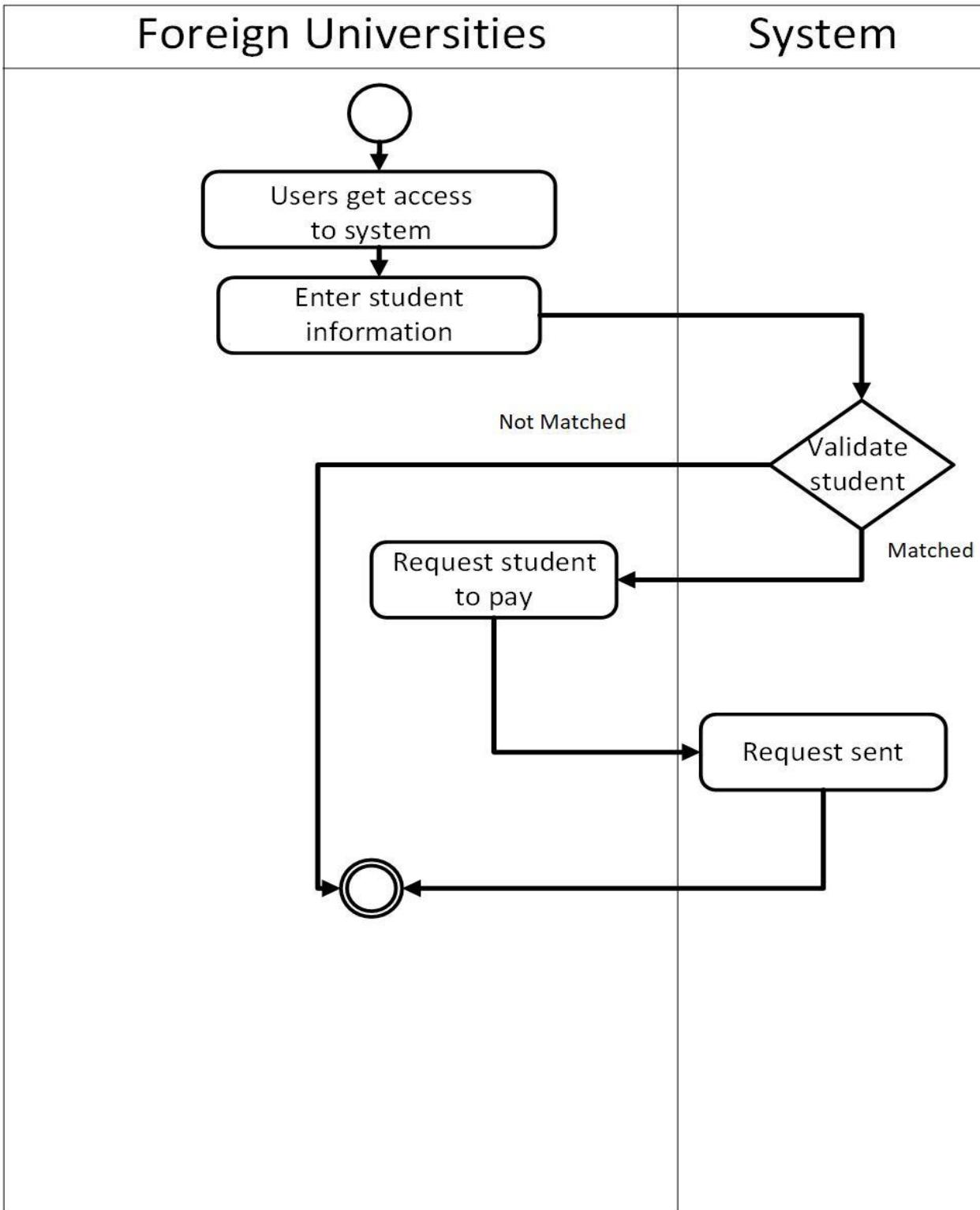


Figure 4.19: Swim lane for Foreign Universities

Chapter 5

Data Model

In this chapter we will discuss about the data models of our system.

5.1 Data Modeling Concept

If software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, a software team may choose to create a data model as part of overall requirements modeling.

5.2 Data Objects

A data object is representation of composite information that must be understood by software. Here, composite information means that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

5.2.1 Identify Data Objects

Nouns having attributes are selected as data object. Those who doesn't have any attributes have covered under the data objects.

Data Object: User

Attributes:

- User id
- Password
- First Name
- Last Name
- Email
- Mobile No.
- Role id
- Present Address
- Permanent Address
- Picture

Data Object: Student

Attributes:

- Student id
- User id
- Registration No.

- Session
- University
- Department/Institute

Data Object: University

Attributes:

- Id
- Name
- Location

Data Object: Course

Attributes:

- Id
- Name
- Semester No
- Credit

Data Object: Semester

Attributes:

- Id
- Name
- University

Data Object: Marks

Attributes:

- Id
- Student id
- Course id
- GPA

Data Object: Department

Attributes:

- Id
- University
- Name

Data Object: Stakeholder

Attributes:

- Id
- Name
- Institute
- Email
- Designation
- Country

Data Object: Verification

Attributes:

- Id
- Student id
- Stakeholder id
- Digital sign
- Verification Status

Data Object: Log

Attributes:

- Id
- User id
- Operation
- Location
- Table Row

Data Object: User Activations

Attributes:

- Id
- User id
- Token
- Created At

Data Object: Registrar

Attributes:

- Id
- User id
- University id

Data Object: PO

Attributes:

- Id
- User id
- Department id

5.2.3 Entity Relationship Diagram

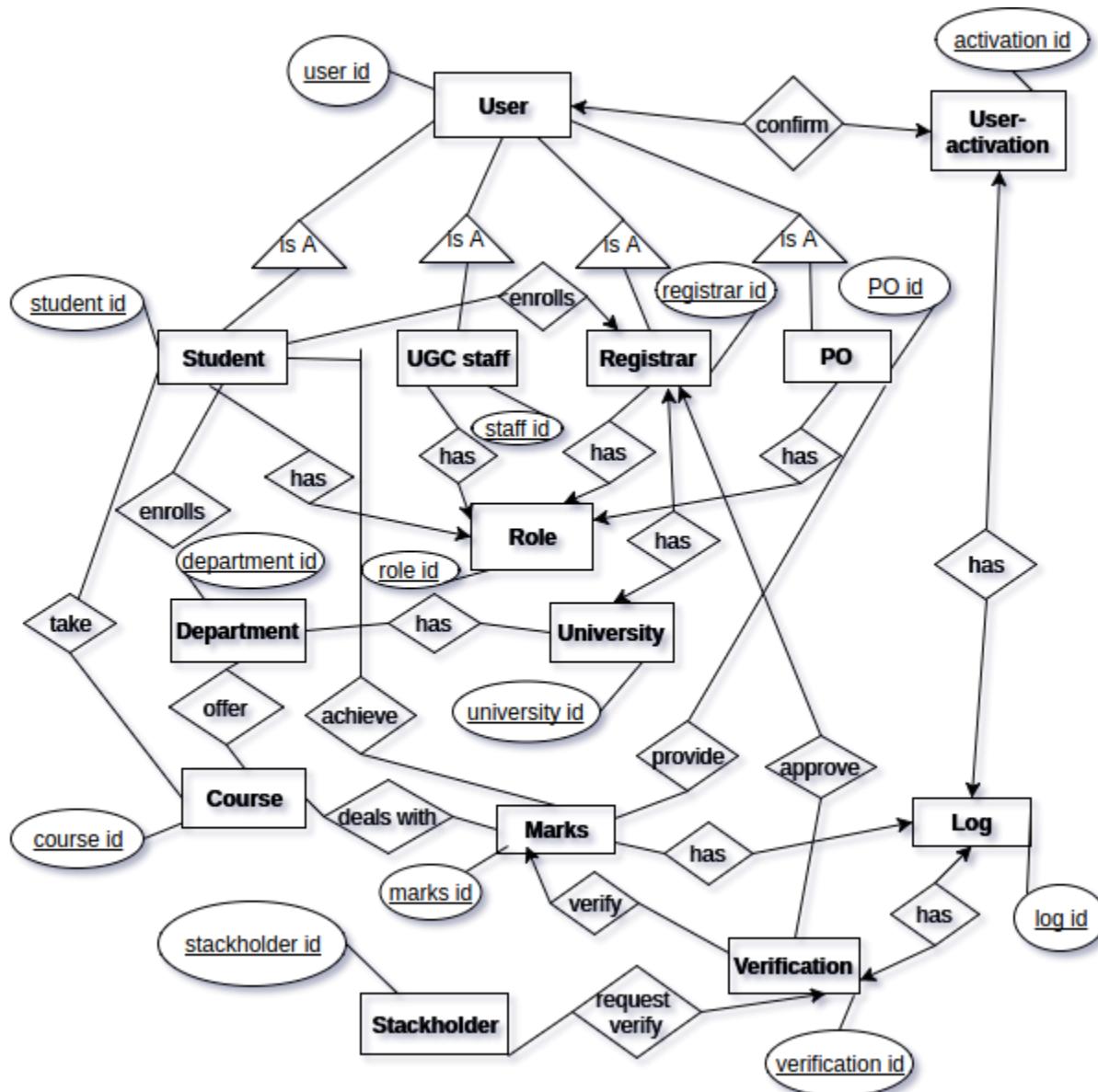


Figure 5.1: ER Diagram for data based modeling

Chapter 6

Class-Based Model

6.1 Introduction

Class-based modelling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.2 Class Diagram

We have shown here how the classes interact together to accomplish certain goal.

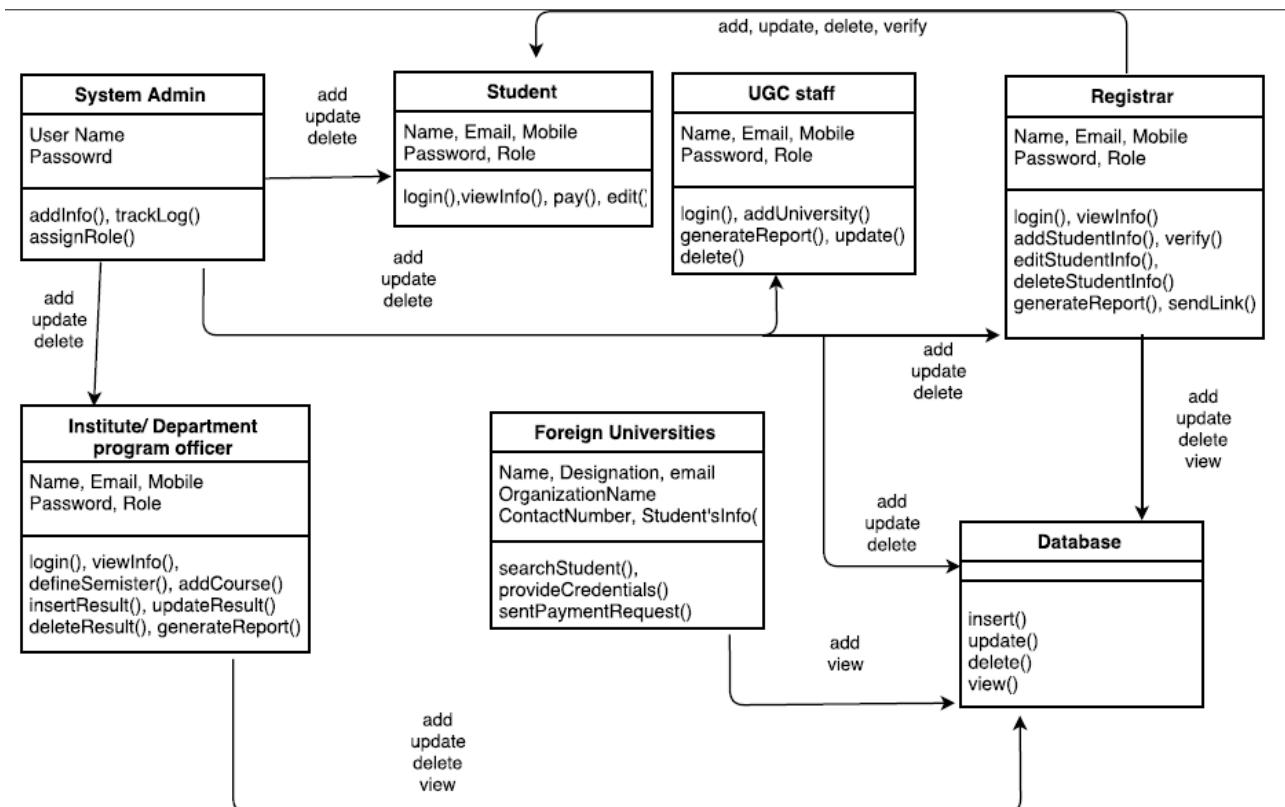


Figure 6.1: Class Diagram

Chapter 7

Flow-Oriented Model

7.1 Introduction

Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today.

7.2 Data flow diagram

A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates this flow of information in a process based on the inputs and outputs. A DFD can be referred to as a Process Model.

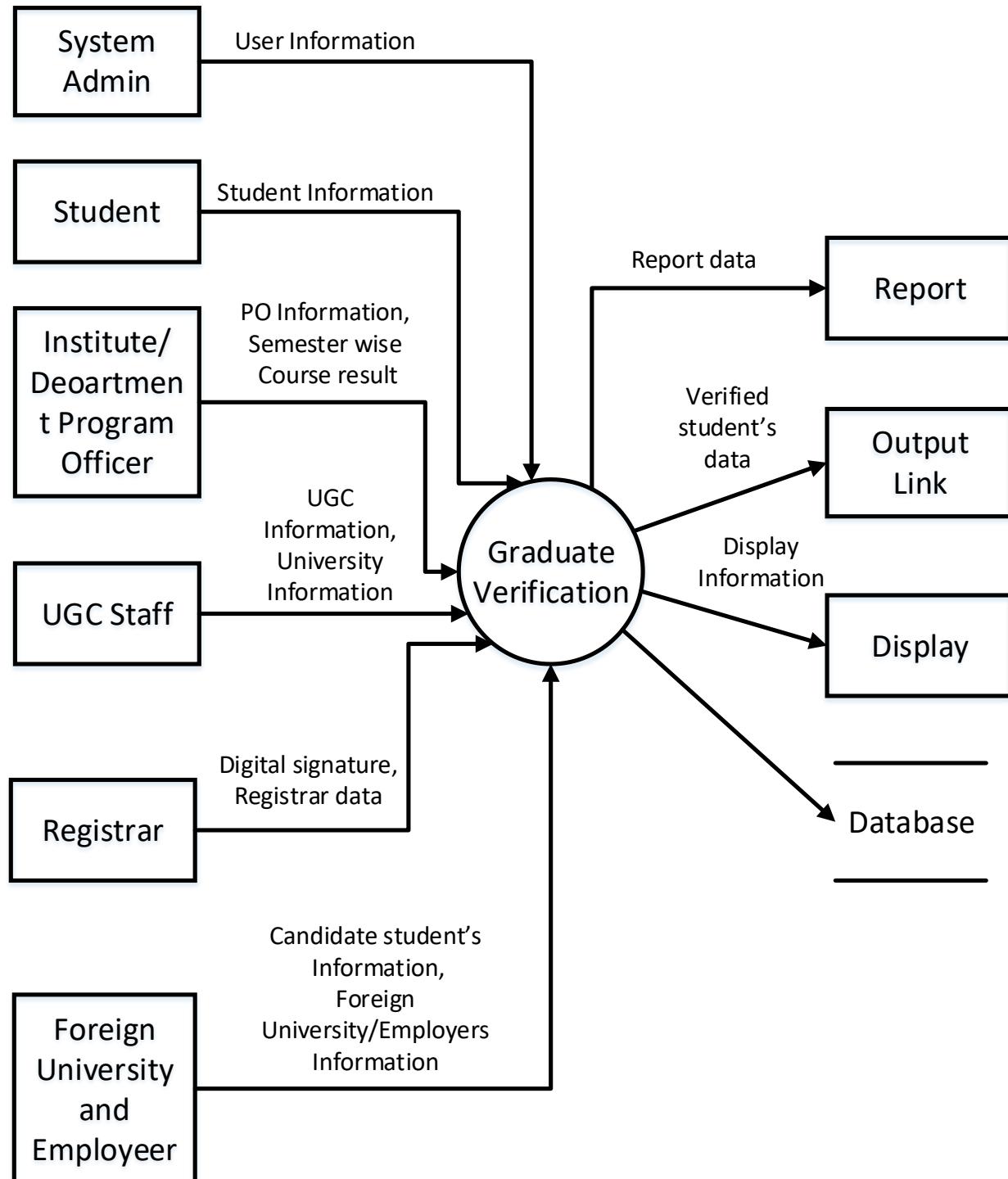


Figure 7.1: Level 1 Data Flow Diagram

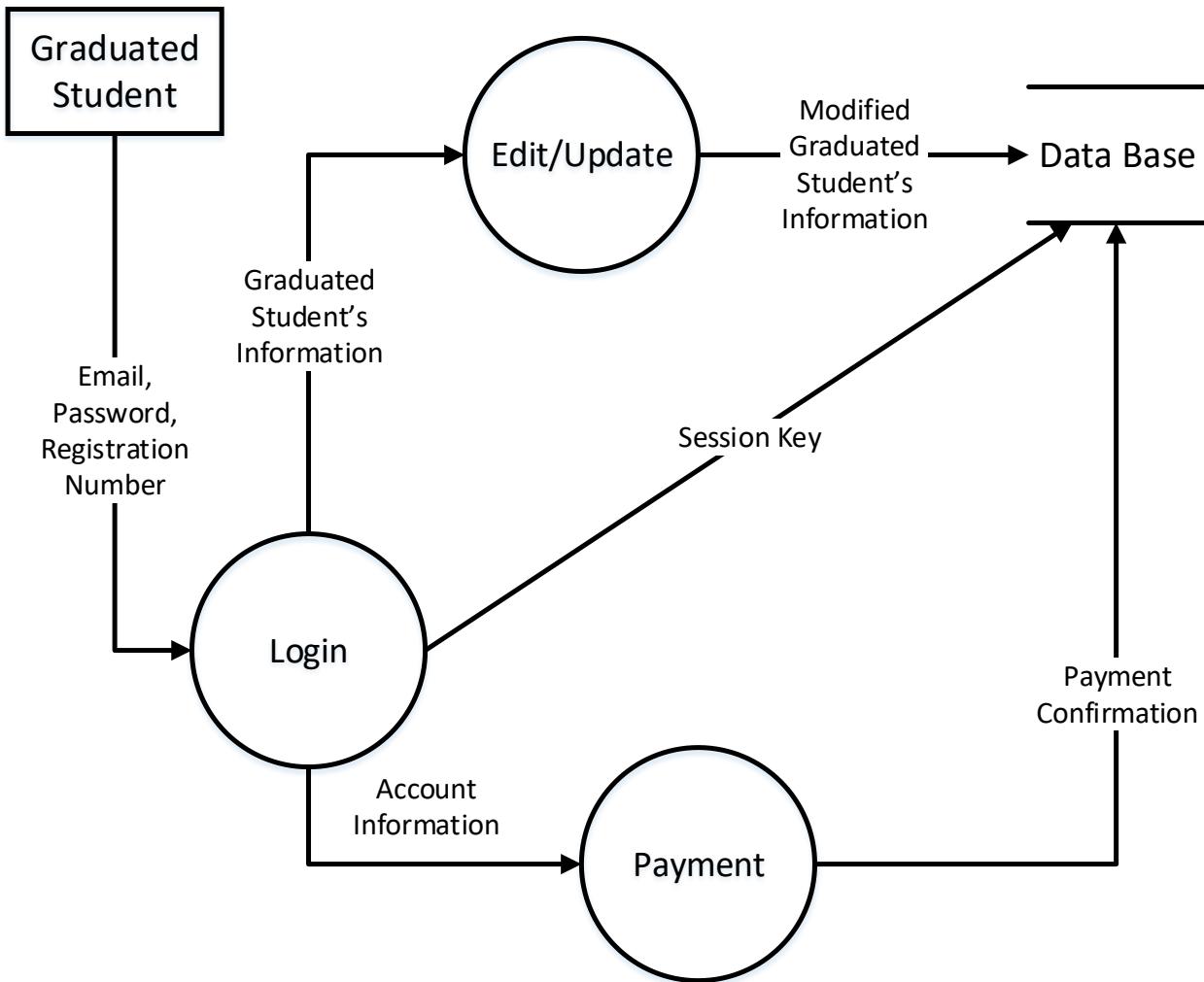


Figure 7.2: Level 1.1 Data Flow Diagram

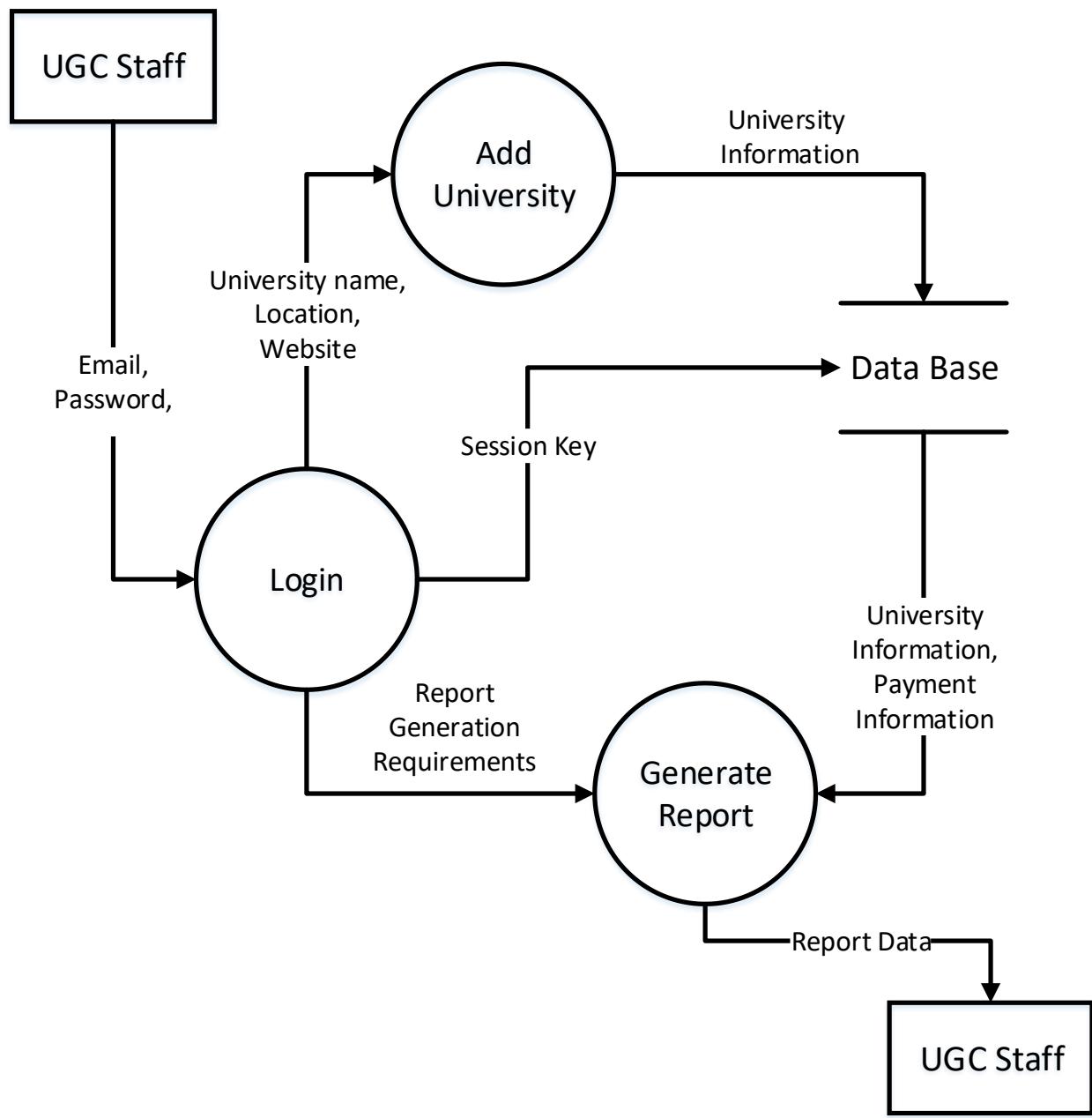


Figure 7.3: Level 1.2 Data Flow Diagram

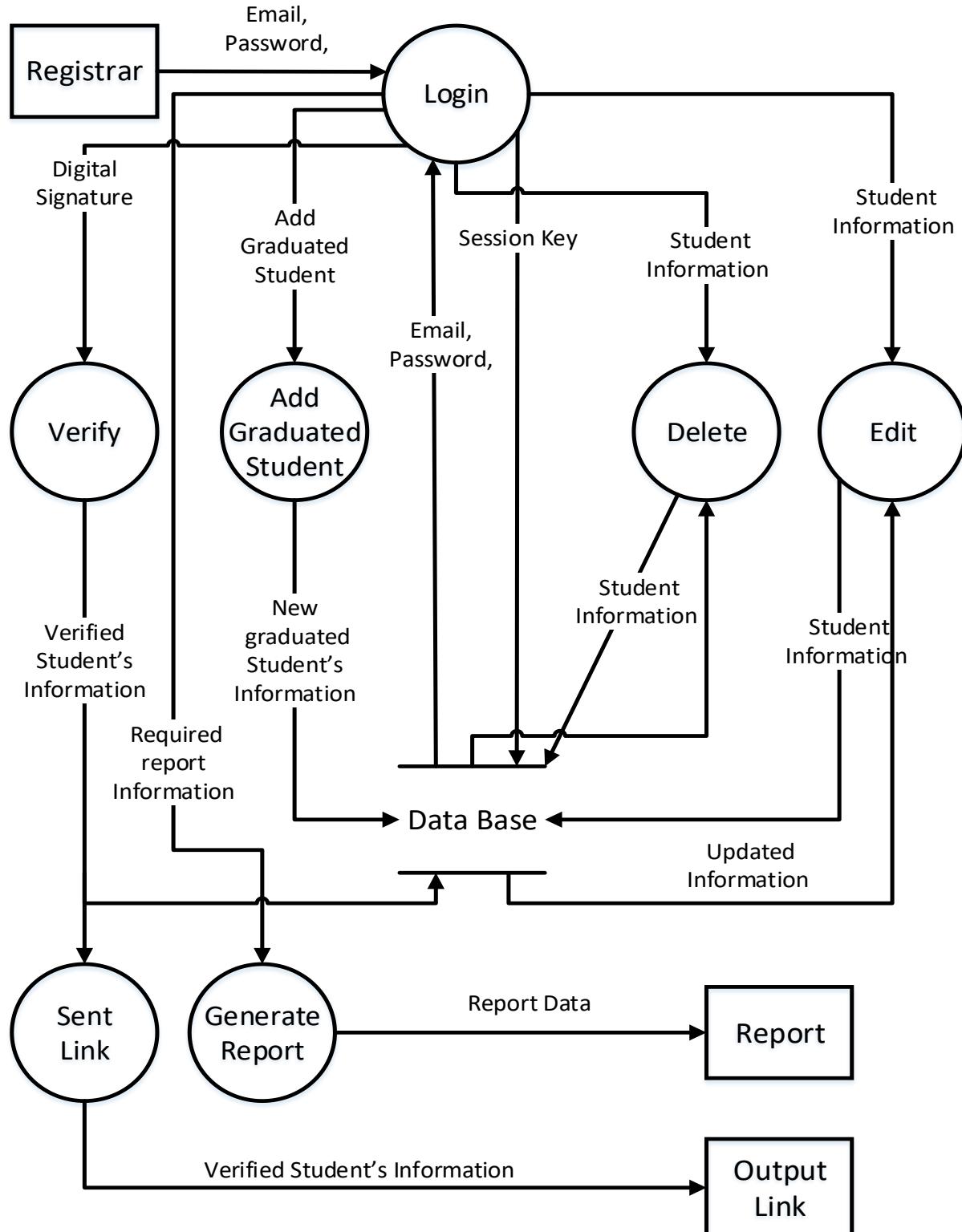


Figure 7.4: Level 1.3 Data Flow Diagram

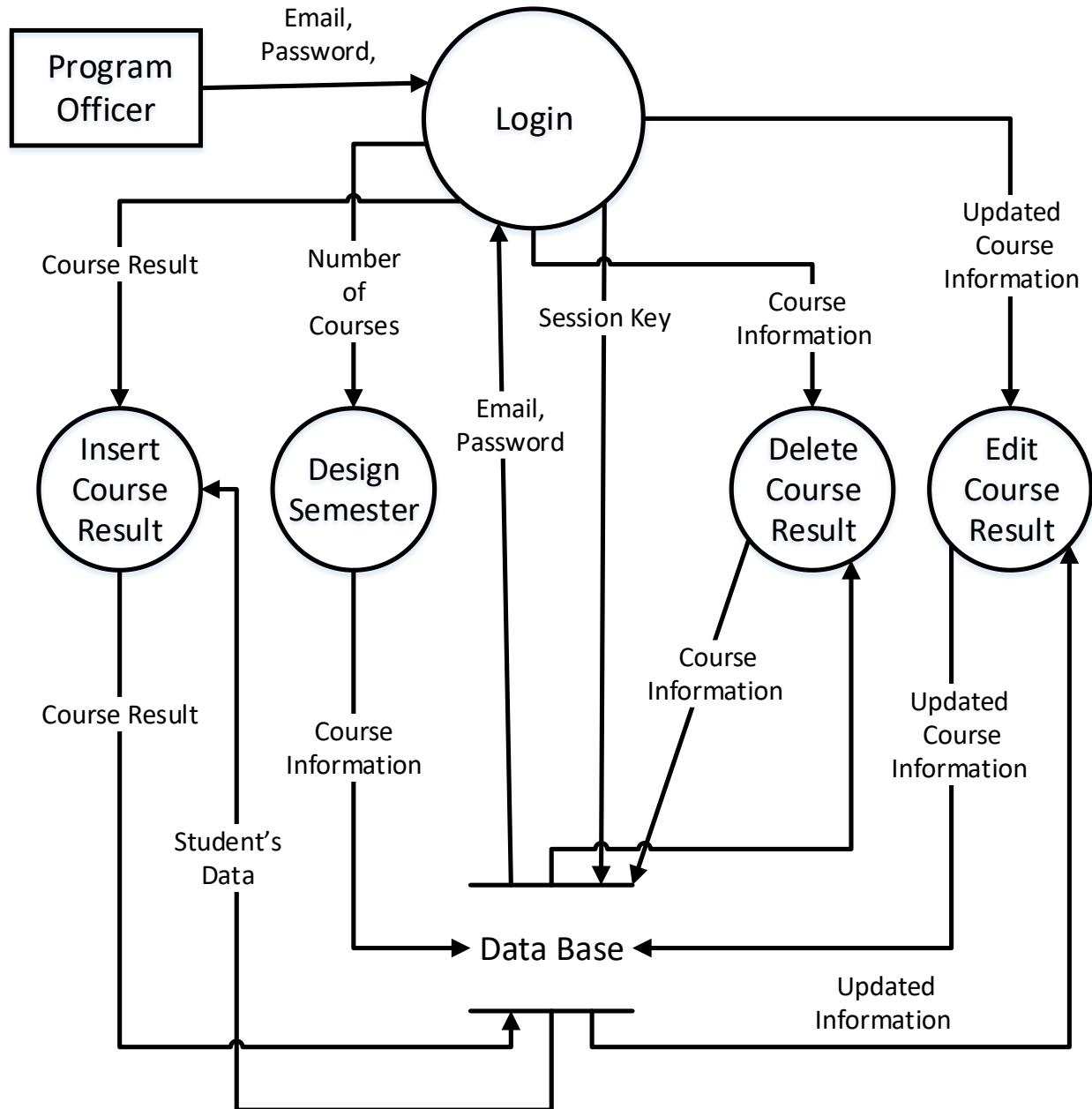


Figure 7.5: Level 1.4 Data Flow Diagram

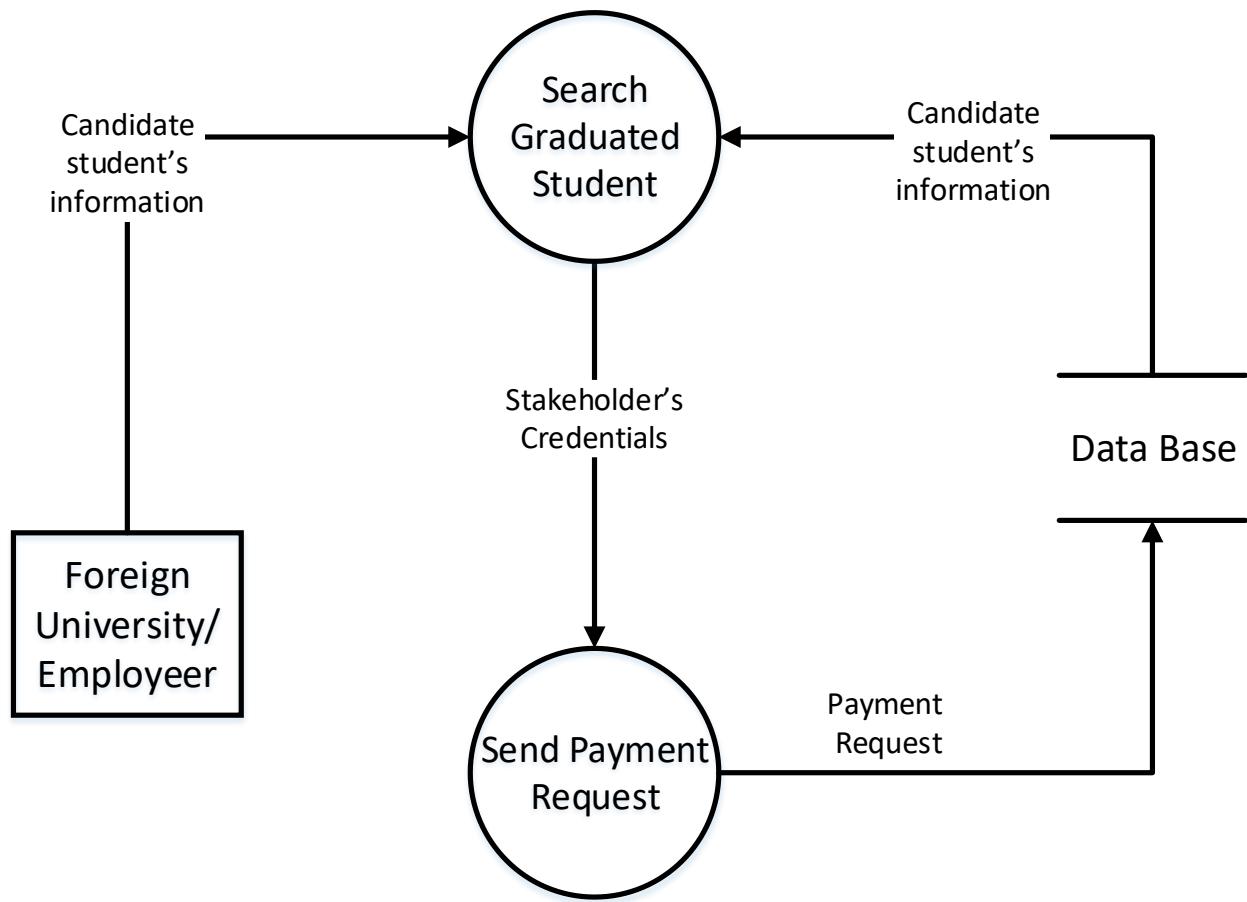


Figure 7.6: Level 1.5 Data Flow Diagram

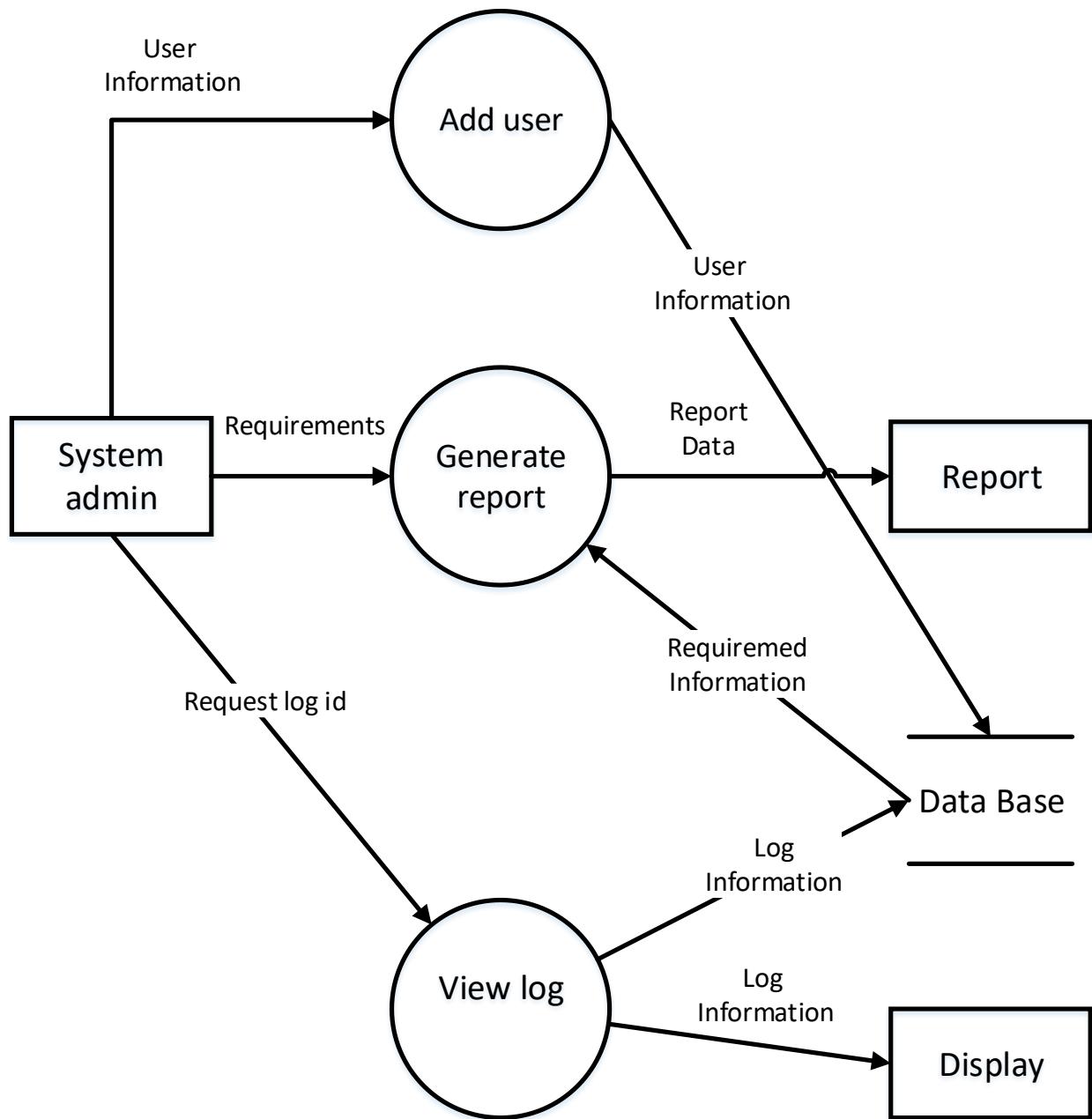


Figure 7.7: Level 1.6 Data Flow Diagram

Chapter 8

Behavioral Model

8.1 Introduction

Behavior modeling is also referred to as State modeling, State machines and State transition matrix. Behavior modeling is when one thinks of his ideas in terms of states and transitions. This requires both identifying all of the interesting states of being that software or its components are likely to be in. And also, at a high level, abstracting what events are likely to cause software or its components to change between states of being.

8.2 Sequence Diagram

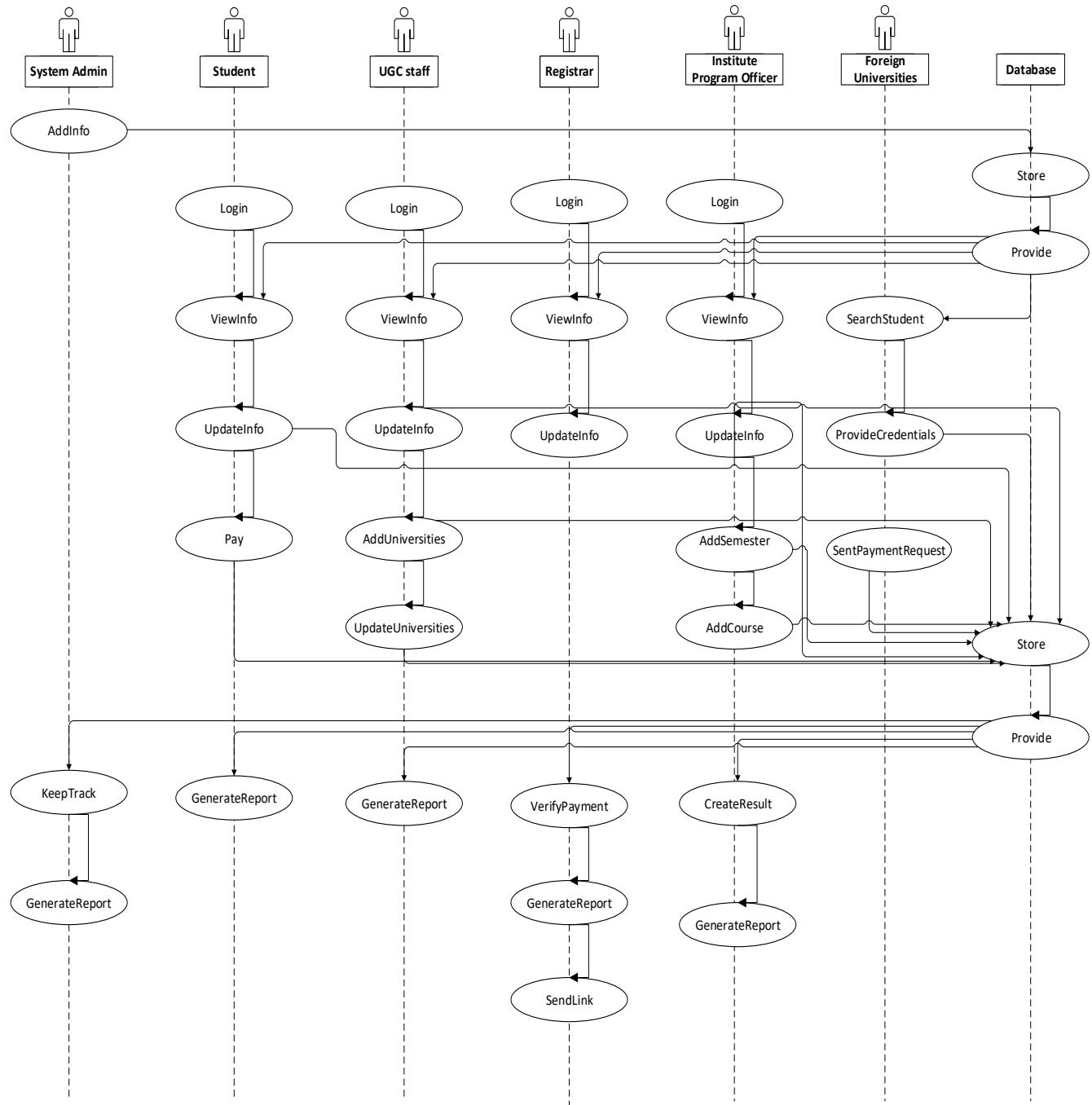


Figure 8.1: Sequence Diagram

Chapter 9

Software Design Architecture

9.1 Architectural Design for OOP:

9.1.1 Representing the system in context

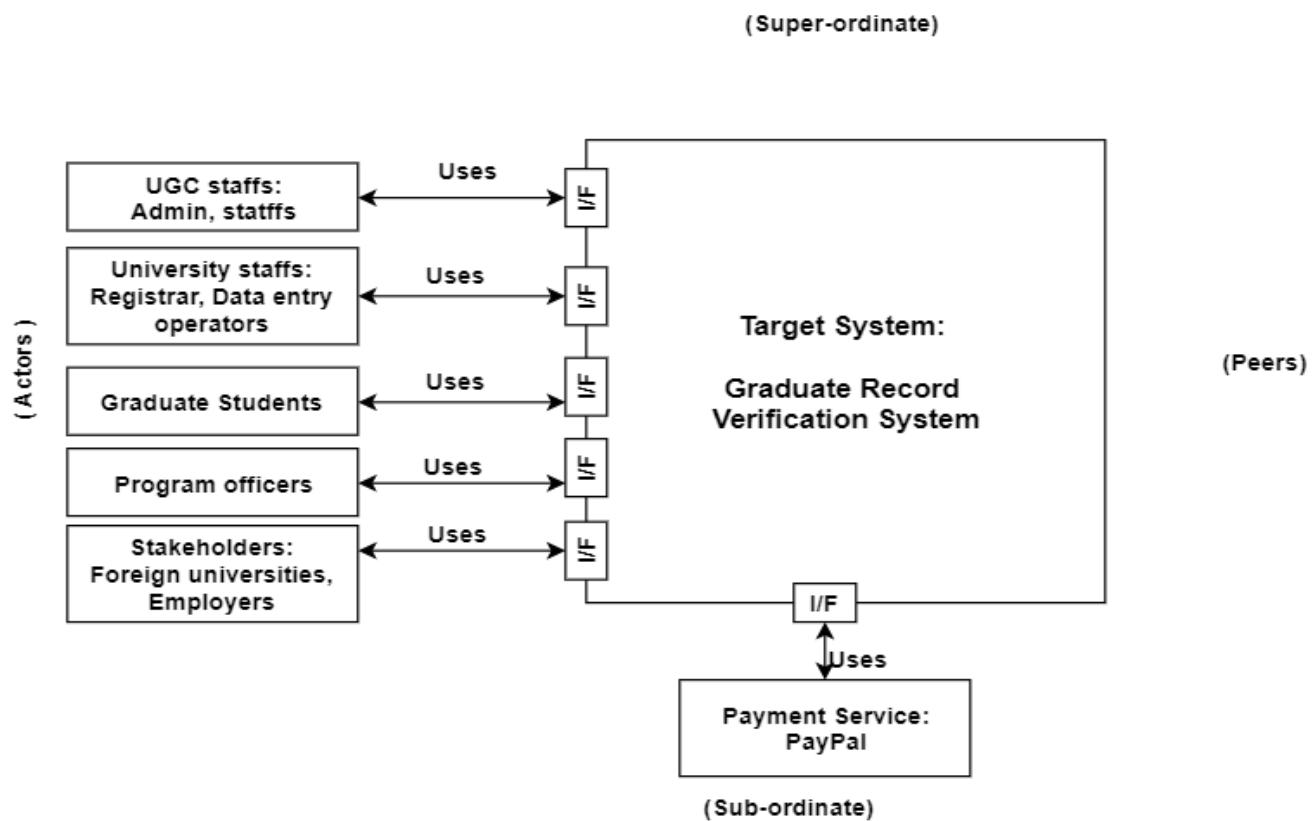


Figure 9.1: Representing the system in context

9.1.2 Define Archetypes

1. User creation
2. Authentication
3. Viewing activity log
4. Report generation
5. Viewing and editing personal information
6. Viewing academic information
7. Viewing payment request from stakeholders
8. Paying through PayPal
9. Adding universities information and its registration's details
10. Adding, removing, updating student information
11. Adding digital signature to verify students
12. Adding course and semester
13. Insertion of semester wise result
14. Validating candidate student's information
15. Sending payment request
16. Sending notification to attach digital signature
17. Sending link to stakeholder

9.1.3 Refining architecture into components

Components:

1. Admin details, registrar details, student details, program officer details, foreign university details, UGC staff details.
2. Admin details, registrar details, student details, program officer details, foreign university details, UGC staff details.
3. Activity log details
4. Report details
5. Admin details, registrar details, student details, program officer details, foreign university details, UGC staff details.
6. Student details
7. Foreign university details
8. Student details
9. University details, registrar details
10. Student details
11. Student details
12. Course details, semester details
13. Result details, semester details
14. Foreign university details
15. Student details, foreign university details
16. Registrar details
17. Foreign university details

Classes:

1. Admin, Registrar, Student, Program Officer, Foreign University, UGC staff, DAL
2. Admin, Registrar, Student, Program Officer, Foreign University, UGC staff, DAL
3. DAL
4. Report, DAL
5. Admin, Registrar, Student, Program Officer, Foreign University, UGC staff, DAL
6. Student
7. University
8. Student
9. Registrar, University
10. Student
11. Student
12. Course, Semester
13. Result, Semester
14. Foreign University, Student
15. Student, Foreign University
16. Registrar
17. Foreign University

9.1.4 Describing Instantiation of the system

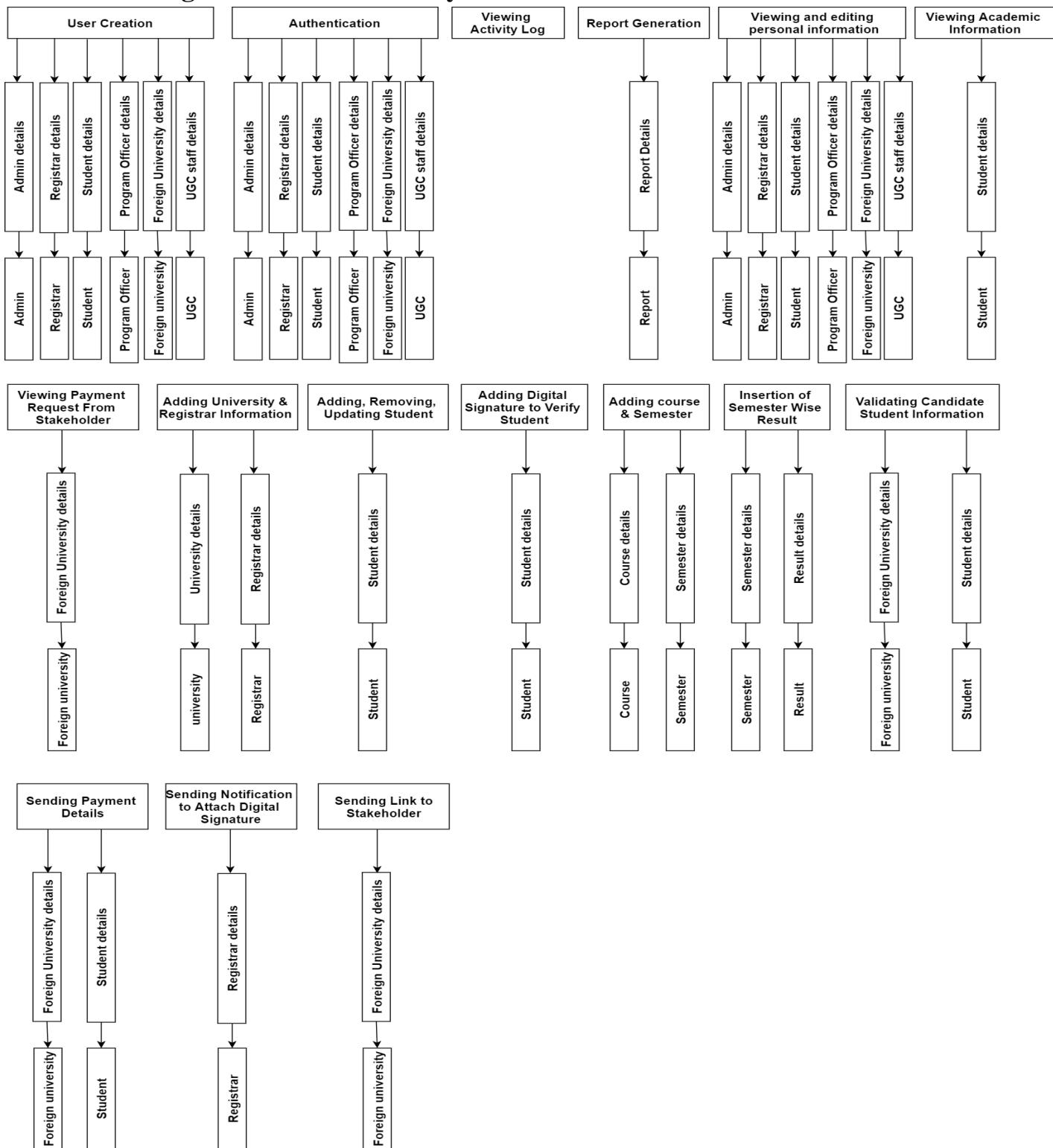


Figure 9.2: Refining archetype into component & classes

Chapter 10

Component Level Design

10.1 Analysis Class

This is the first step of object-oriented approach in component level design. In this section, all design classes that correspond to the problem domain as defined in the analysis model and architectural model are identified. The analysis classes for this project are:

1. System Admin
2. Student
3. UGC Stuff
4. Registrar
5. University/Department Program Office
6. Foreign Universities
7. Database

System Admin

-FirstName
-LastName
-Username
-Password

+AddInfo()
+UpdateInfo()
+AssignRole()
+TrackLog()

Student

-FirstName
-LastName
-Email
-Mobile Number
-Role
-Password

+Login()
+ViewInfo
+Pay()
+Edit()

Registrar

-FirstName
-LastName
-Email
-Mobile Number
-Role
-Password

+Login()
+ViewInfo()
+AddStudentInfo()
+EditStudentInfo()
+DeleteStudentInfo()
+Verify()
+GenerateReport()
+SendLink()

Program Officer

-FirstName
-LastName
-Email
-Mobile Number
-Role
-Password

+Login()
+ViewInfo()
+DefineSemester()
+AddCourse()
+InsertResult()
+UpdateResult()
+DeleteResult()
+GenerateReport()

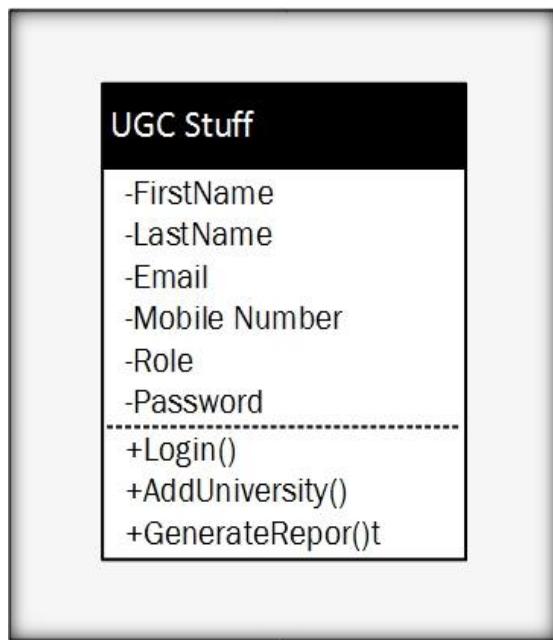


Figure 10.1: Design component of System Admin Class

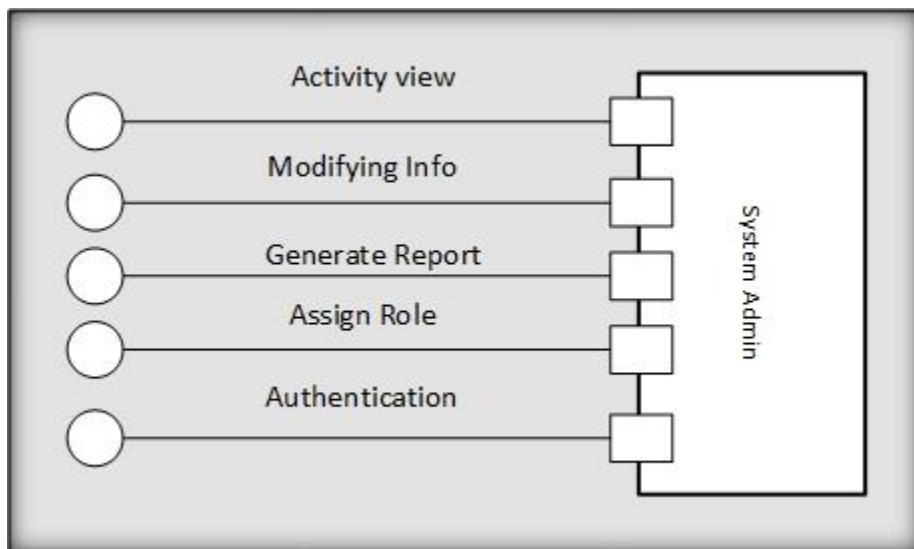


Figure 10.2: Design component of System Admin Class

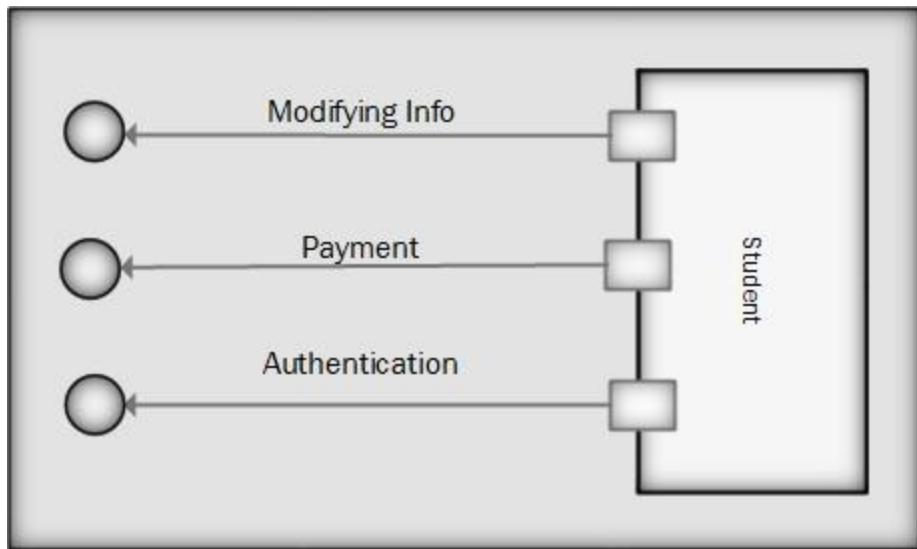


Figure 10.3: Design component of Student Class

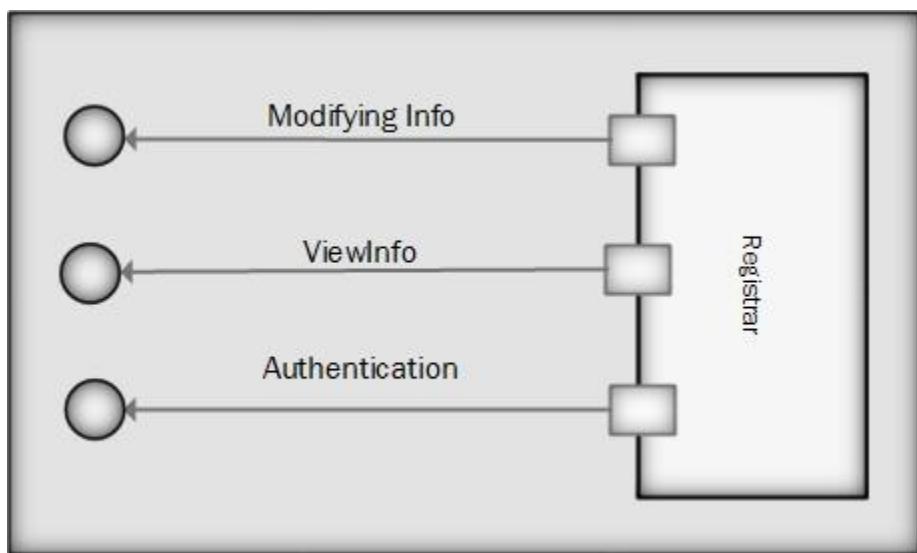


Figure 10.4: Design Component of Registrar Class

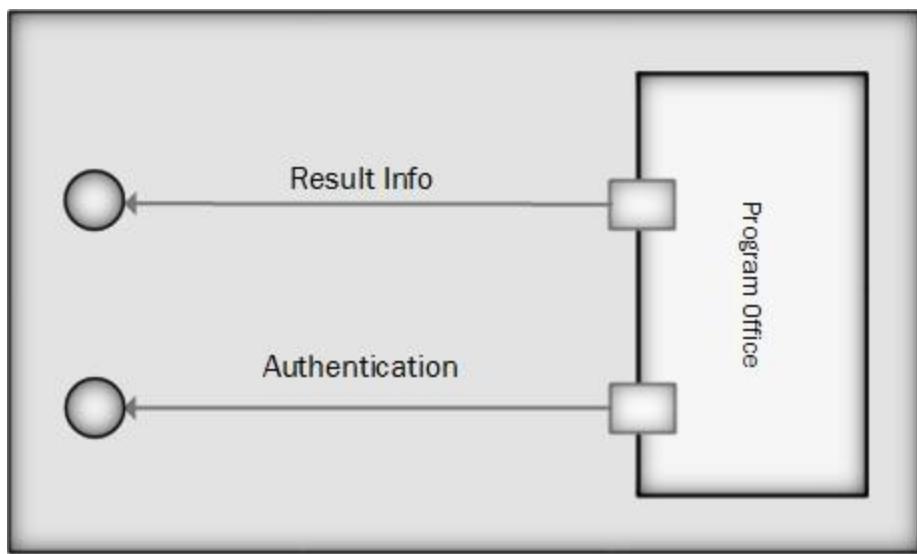


Figure 10.5: Design Component of Program Office

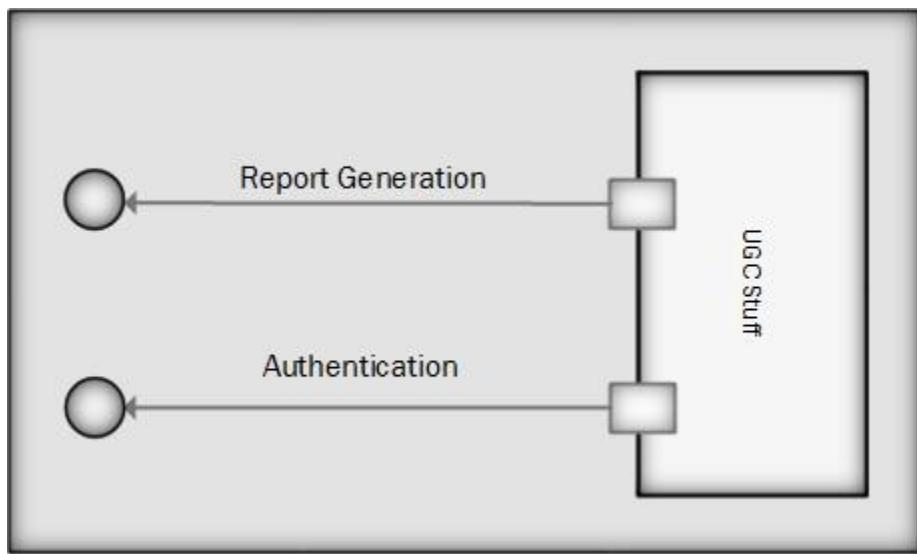


Figure 10.6: Design Component of UGC Stuff Class

10.2 Infrastructure Class

In this section, all design classes that correspond to the infrastructure domain are identified and listed. These classes are usually not present in the analysis or architectural models. These classes include GUI components, operating system components, data management components, networking components, etc. For OGRVS, the infrastructure classes are listed below:

1. Report
2. Course
3. Semester
4. Result
5. DAL
6. University

10.3 Elaborated Design Class

The elaborated design classes are the following:

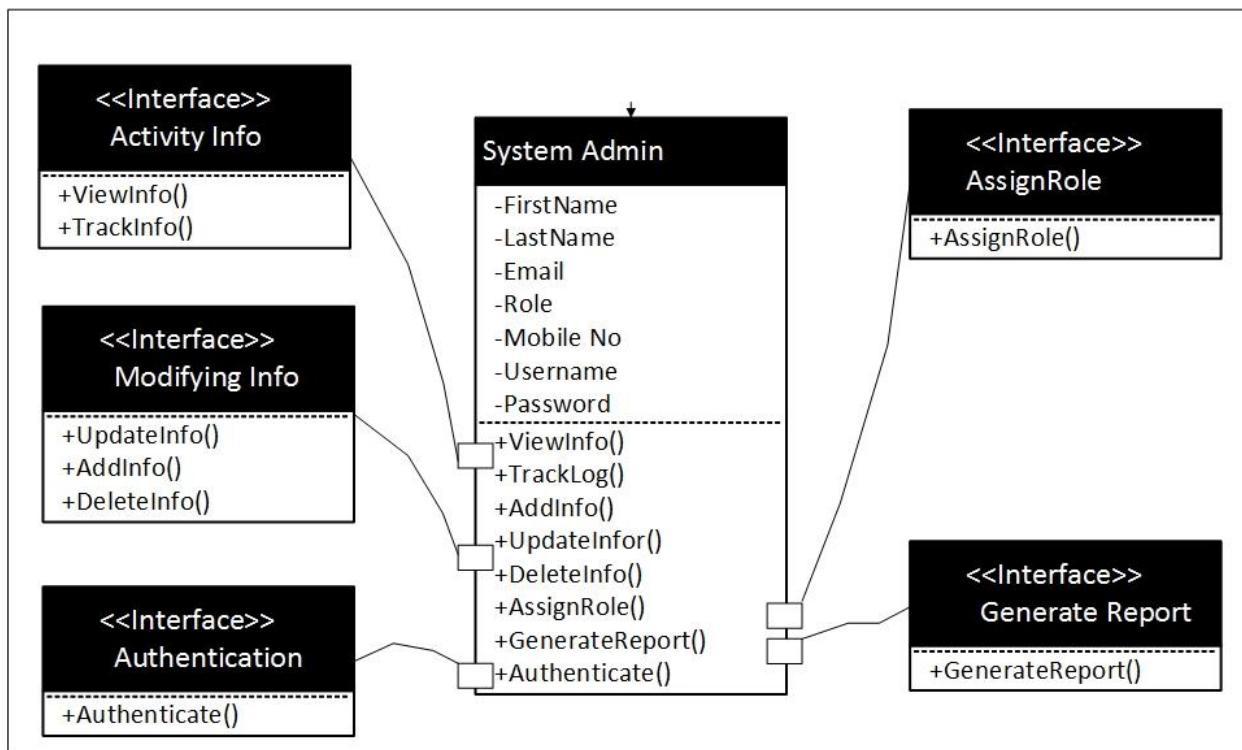


Figure 10.7: Elaborated Design of System Admin Class

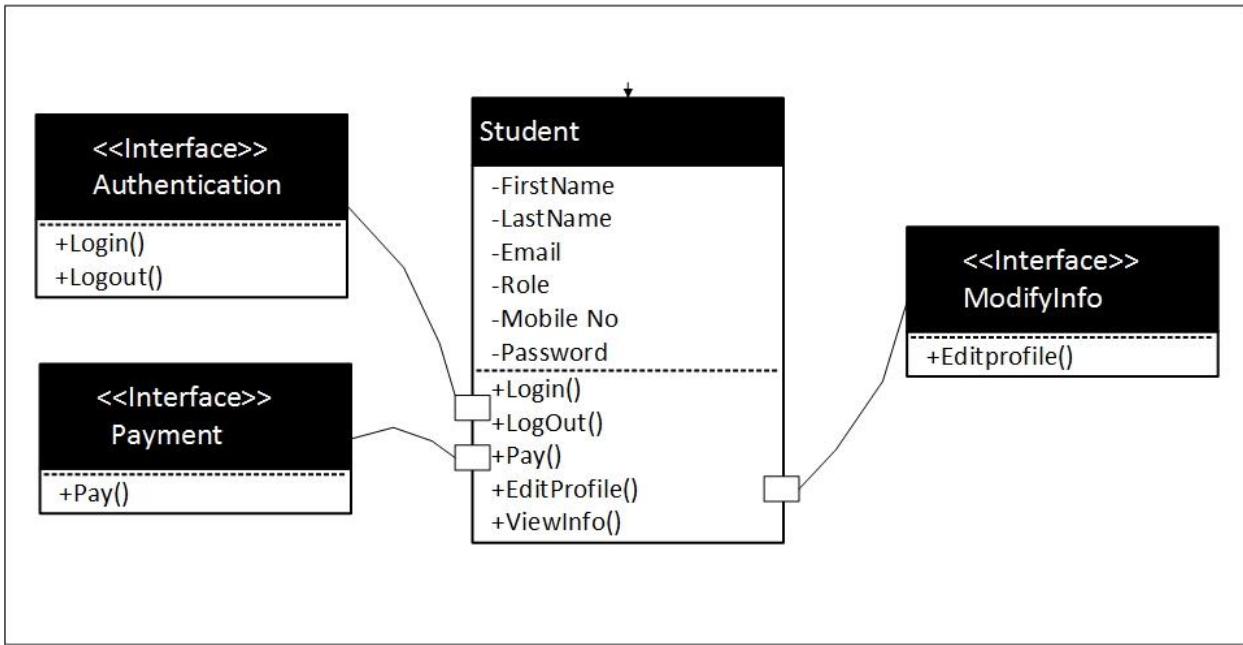


Figure 10.8: Elaborated Design of Student Class

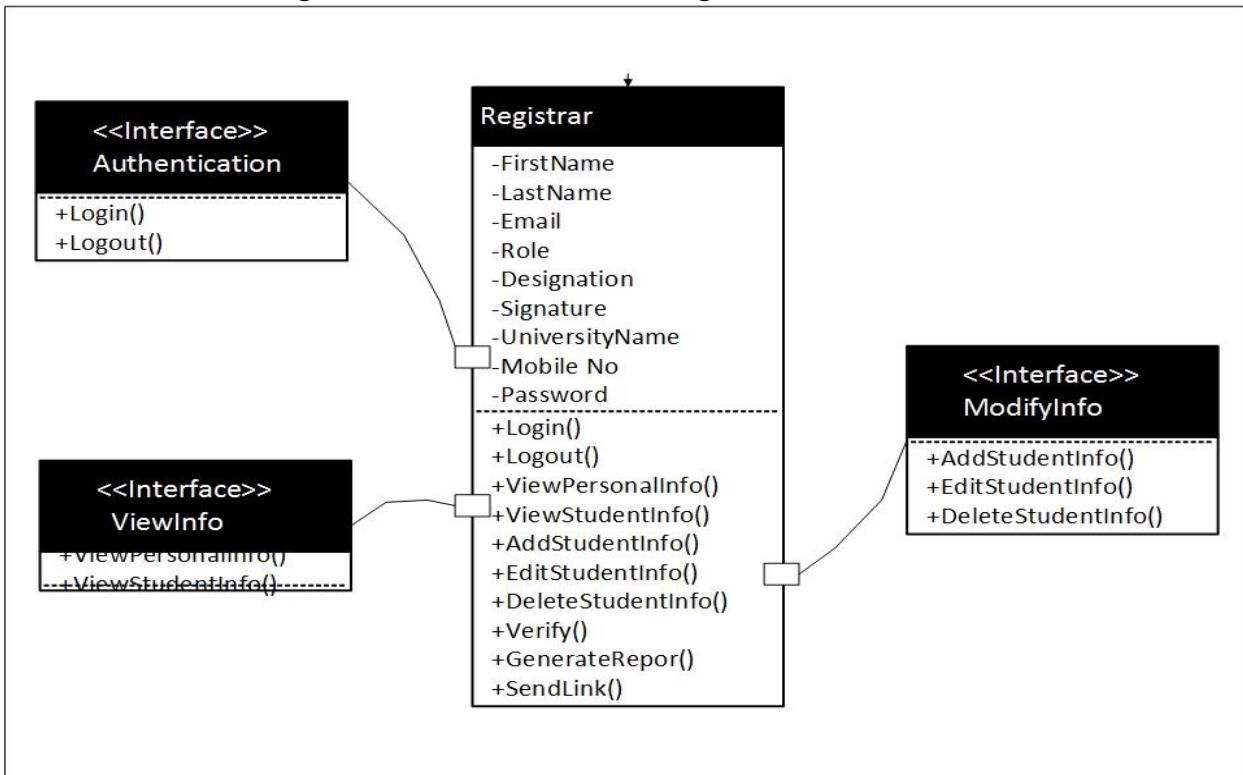


Figure 10.9: Elaborated Design of Registrar Class

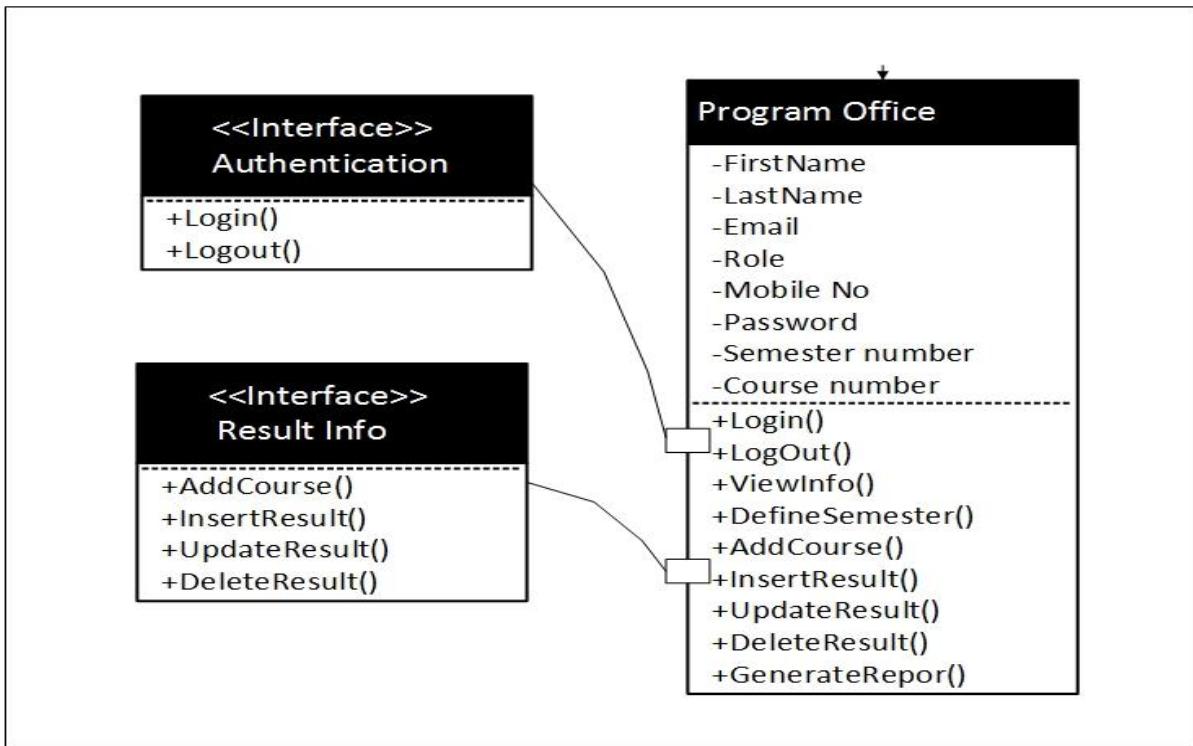


Figure 10.10: Elaborated Design of Program Office Class

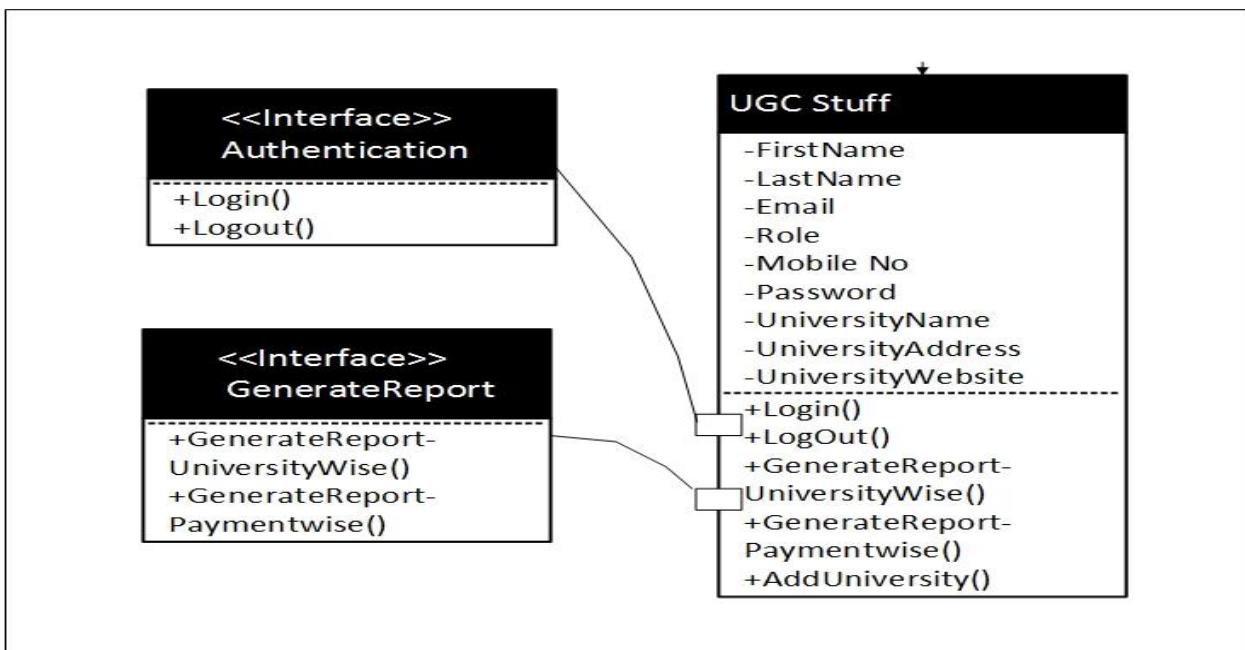


Figure 10.11: Elaborated Design of UGC Class

10.3.1 Collaboration Details

In this section, the collaboration between classes will be designed. Following is the diagrams:

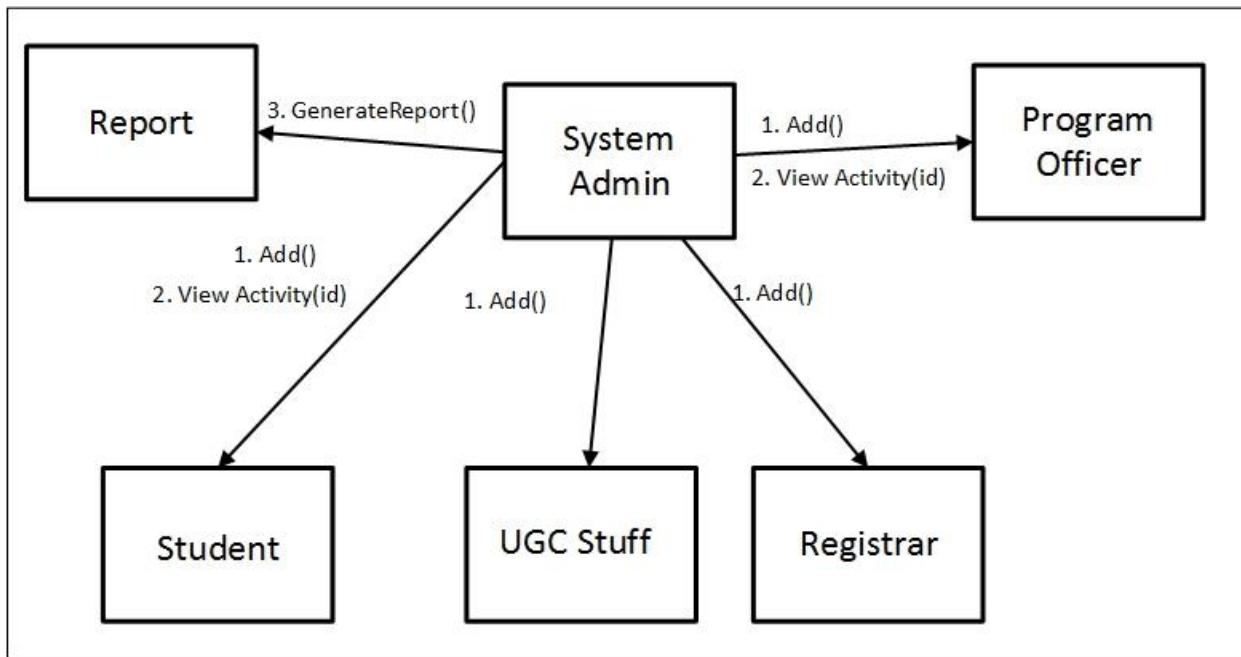


Figure 10.11: Collaboration Details of System Admin Class

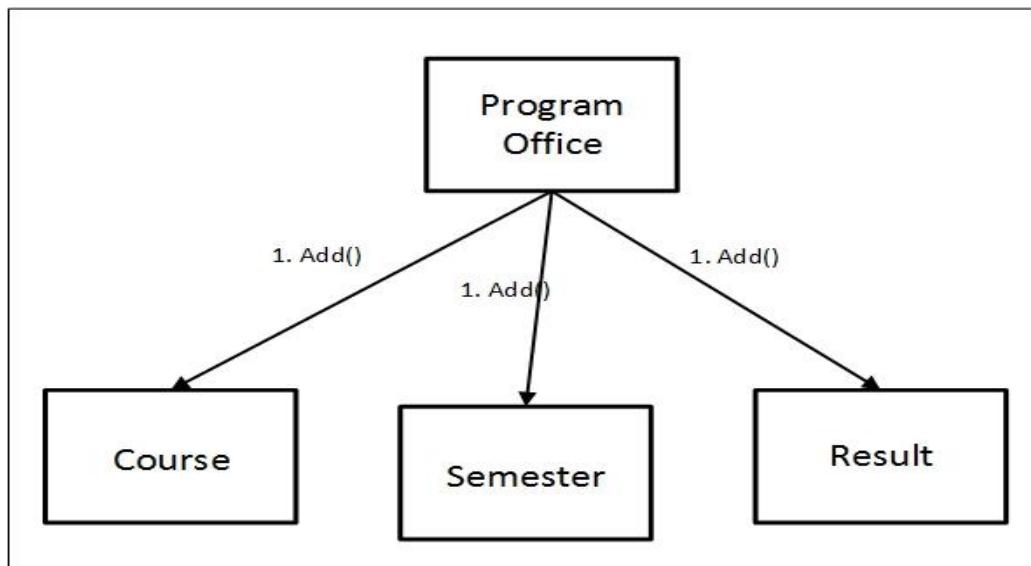


Figure 10.12: Collaboration Details of Program Office

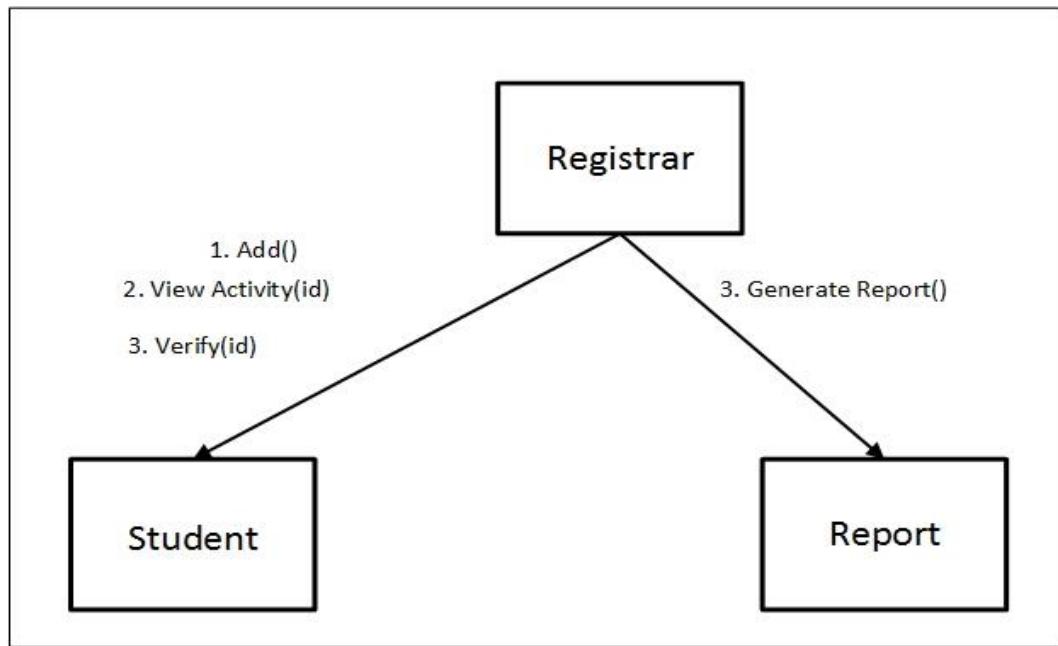


Figure 10.13: Collaboration Details of Registrar Class

10.3.2 Appropriate Interfaces

There is no necessity to divide the classes in subclasses as they exhibit sufficient cohesion. So there is no need to define appropriate interfaces.

10.3.3 Elaborate Attributes

Elaborate attributes for System **Admin** class:

First-name: string = “name1” {contains name value – Rubaiyat, Tasnim, Farzana}

Last-name: string = “name2” {contains name value – Mumu, fabiha, Muna}

Email: email = “null” {Contains value – rubaiyat@gmail.com, tasnim@hotmail.com}

Role: string = “null” {contains name value – student, admin, PO}

Username: string = “username1” {All Characters}

Password: string= “password1” {All Characters}

Mobile Number: Number = “088 01716724859” {contains name value – 088-01711467599}

Elaborated attributes for **Student** class:

First-name: string = “name1” {contains name value – Rubaiyat, Tasnim, Farzana}

Last-name: string = “name2” {contains name value – Mumu, fabiha, Muna}

Email: email = “null” {Contains value – rubaiyat@gmail.com, tasnim@hotmail.com}

Role: string = “role1” {contains name value – student, admin, PO}

Password: string= “null” {All Characters}

Mobile Number: Number = “088 01716724859” {contains name value – 088-01711467599}

Elaborated attributes for **Registrar** class:

First-name: string = “name1” {contains name value – Rubaiyat, Tasnim, Farzana}

Last-name: string = “name2” {contains name value – Mumu, fabiha, Muna}

Email: email = “null” {Contains value – rubaiyat@gmail.com, tasnim@hotmail.com}

Role: string = “null” {contains name value – student, admin, PO}

Password: string= “password1” {All Characters}

Mobile Number: Number = “088 0000000000” {contains name value – 088-01711467599}

University-name: String= “null” {Contains name value – Dhaka University, Chittagong university}

Designation: string = “null” {Contains name value- Manager, Senior manager}

Elaborated attributes for **Program Office** class:

First-name: string = “name1” {contains name value – Rubaiyat, Tasnim, Farzana}

Last-name: string = “name2” {contains name value – Mumu, fabiha, Muna}

Email: email = “null” {Contains value – rubaiyat@gmail.com, tasnim@hotmail.com}

Role: string = “null” {contains name value – student, admin, PO}

Password: string= “password1” {All Characters}

Mobile Number: Number = “088 0000000000” {contains name value – 088-01711467599}

Semester number: Int = “1” { automatically increments – 1,2,3,..... }

Course number: Int = “1” { automatically increments – 1,2,3,..... }

Elaborated attributes for **UGC Stuff** class:

First-name: string = “name1” {contains name value – Rubaiyat, Tasnim, Farzana}

Last-name: string = “name2” {contains name value – Mumu, fabiha, Muna}

Email: email = “null” {Contains value – rubaiyat@gmail.com, tasnim@hotmail.com}

Role: string = “null” {contains name value – student, admin, PO}

Password: string= “password1” {All Characters}

Mobile Number: Number = “088 0000000000” {contains name value – 088-01711467599}

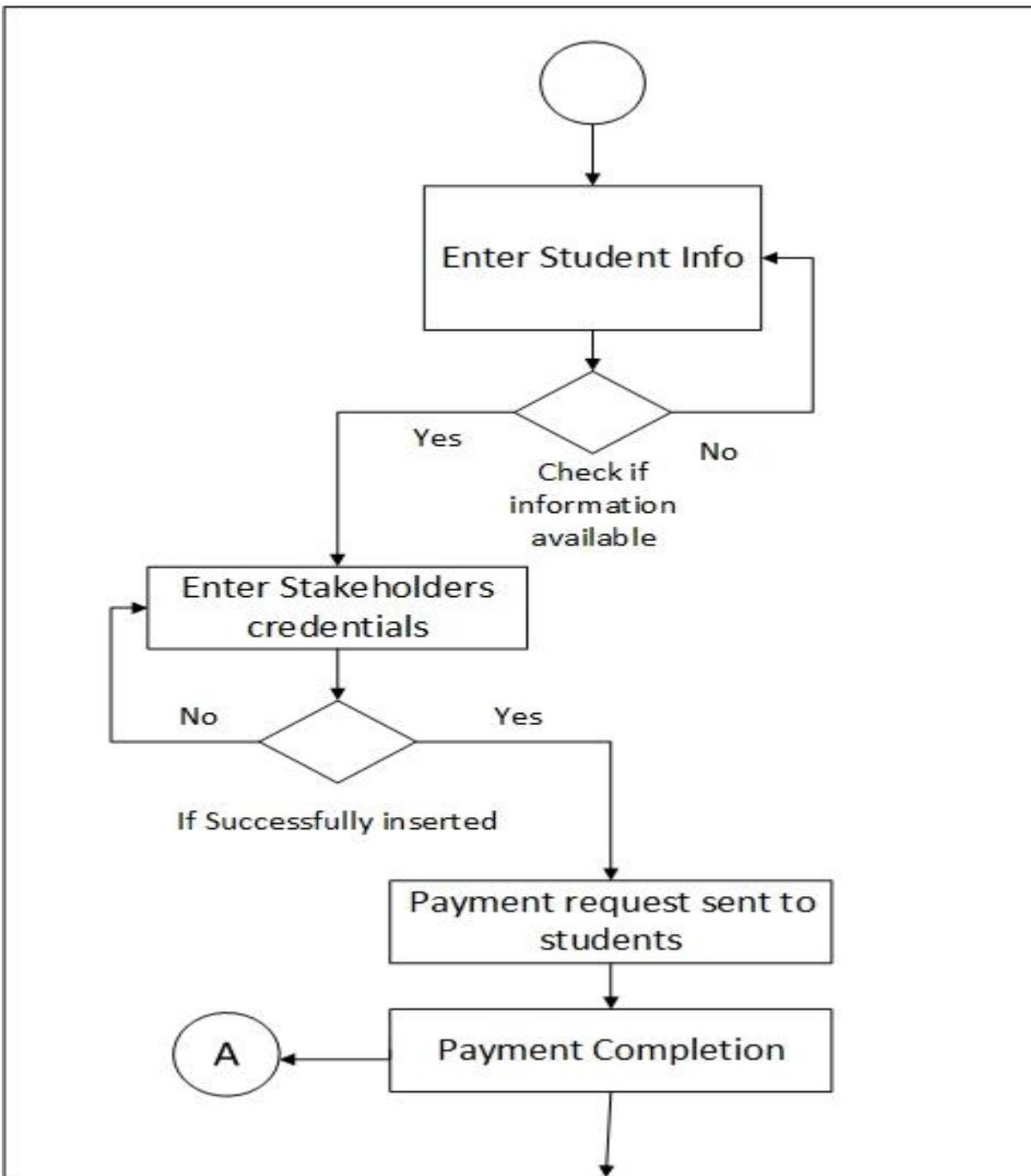
University-name: String= “null” {Contains name value – Dhaka University, Chittagong university}

University-address: String = “null” {All Characters}

University-website: String = “null” {All Characters – www.du.ac.bd}

10.3.3 Describe Processing Flow

The processing flow of the verification system is described below:



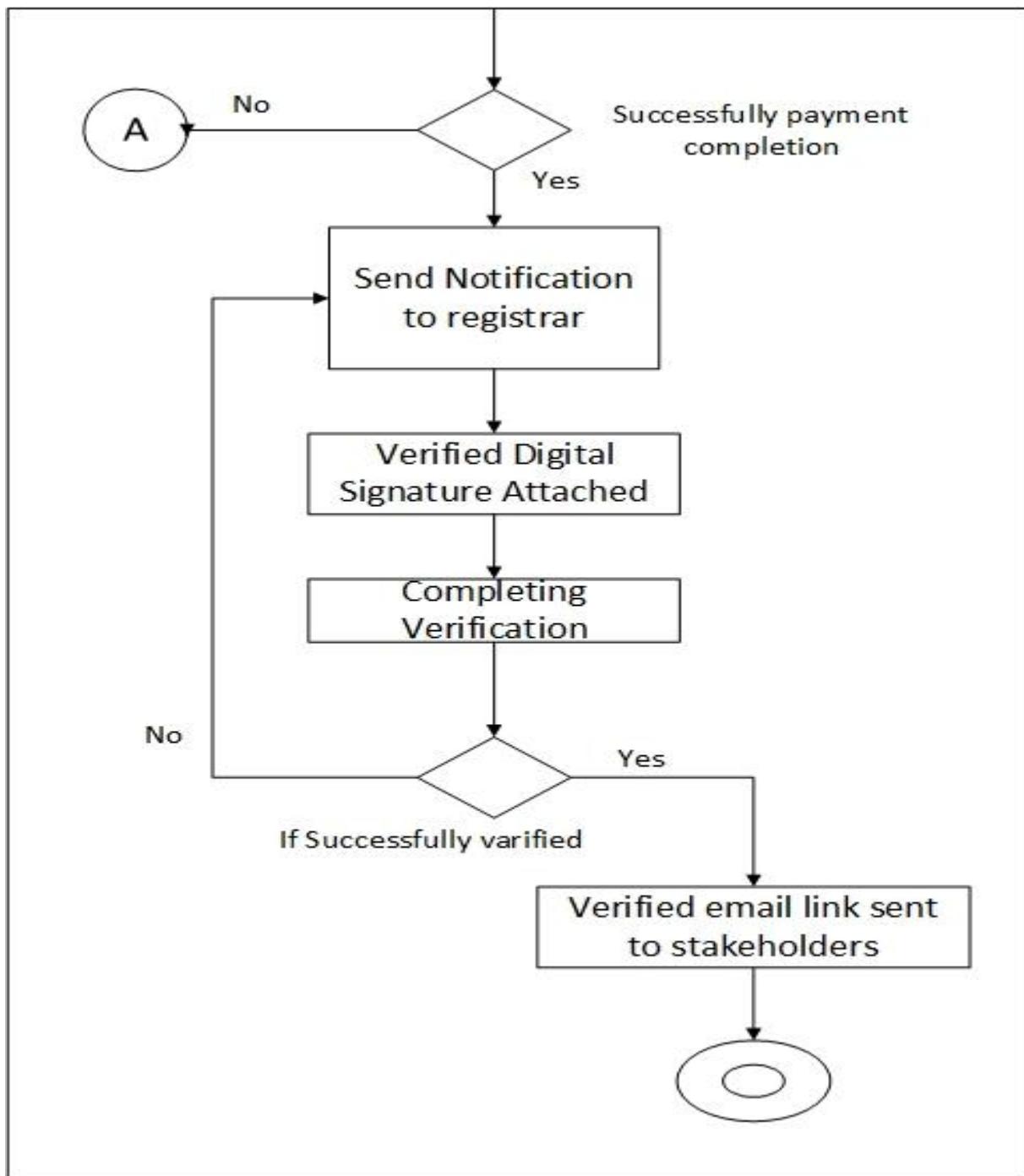


Figure 10.14: Processing Flow of Verification System

10.4 Persistent Data

The persistent data sources (databases and files) and the classes required to manage them are following:

Persistent data source: Database

Classes to manage data source: Database classes for each entity

10.5. Develop or elaborate behavioral representation for a class or component

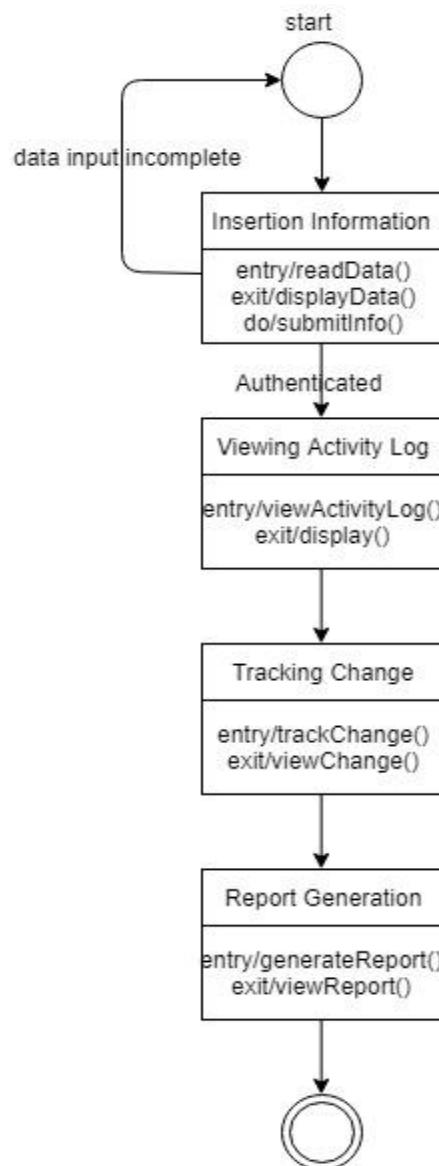


Figure 10.16: State chart fragment for “System Admin”

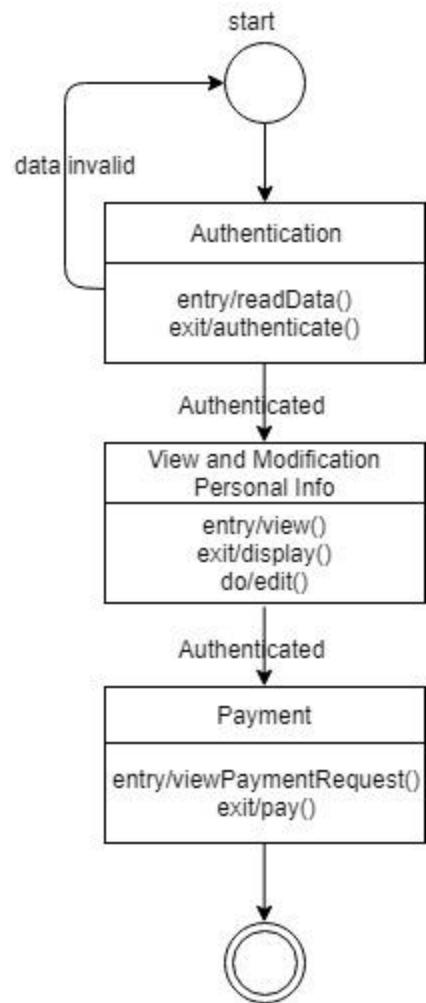


Figure 10.17: State chart fragment for “Student”

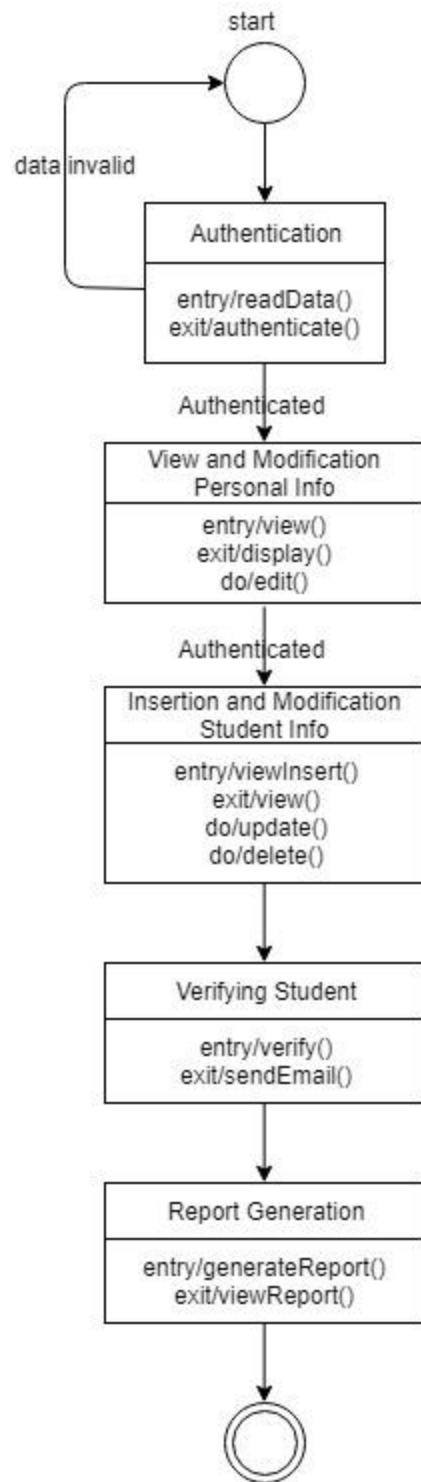


Figure 10.18: State chart fragment for “Registrar”

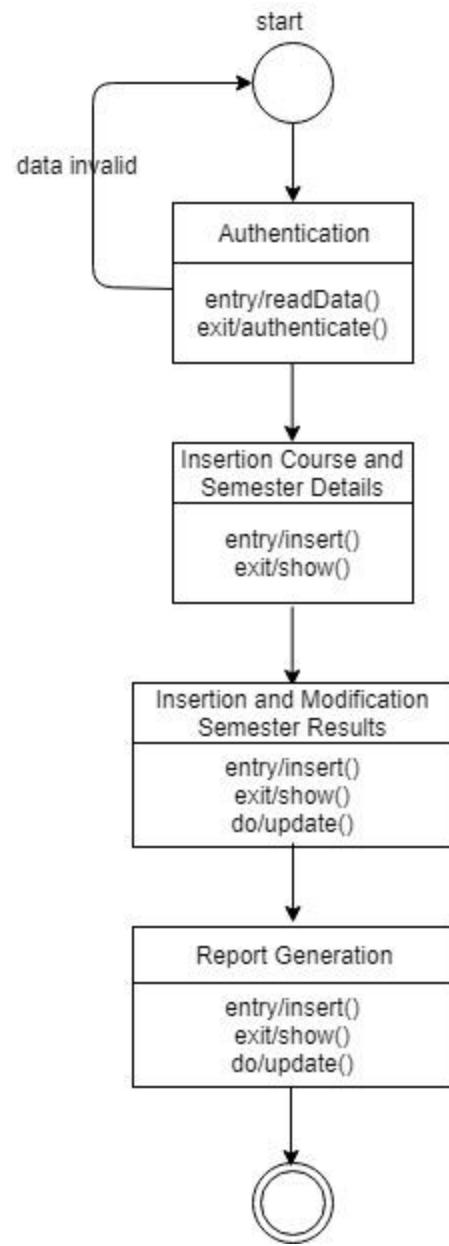


Figure 10.19: State chart fragment for “Program Office”

10.6. Elaborative Deployment

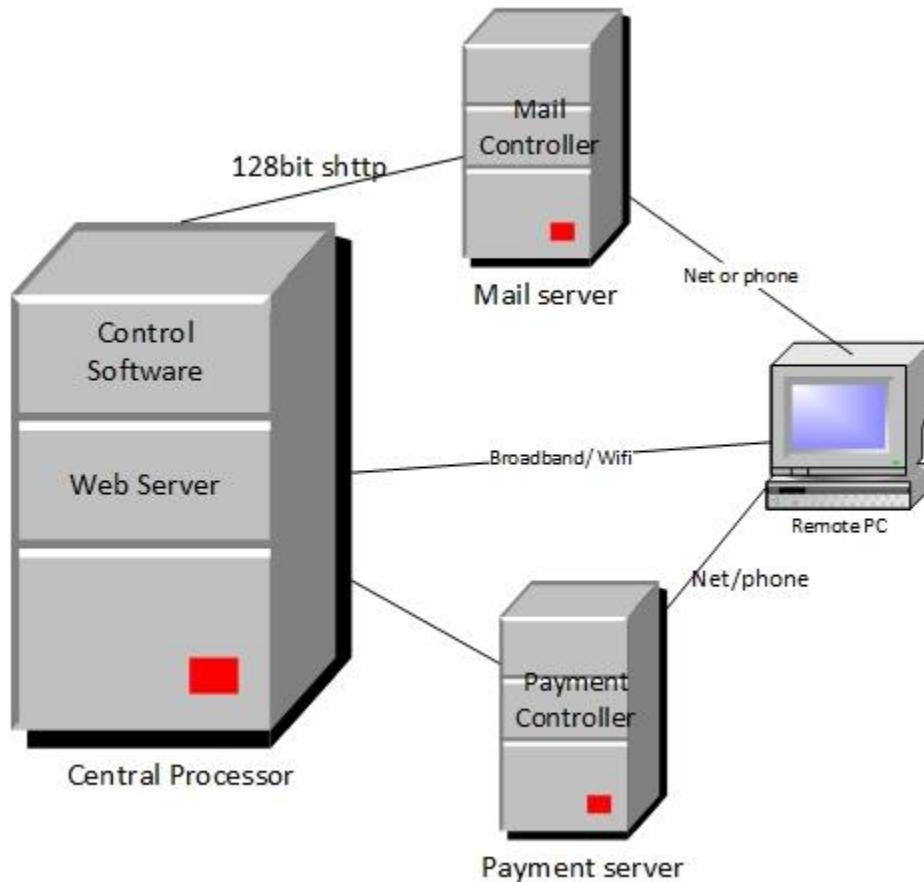


Figure 10.20: Elaborative Deployment of OGRVS

Chapter 11

User Interface design

11.1 Introduction

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. User interface design creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.

When we run the application, the following window will appear:

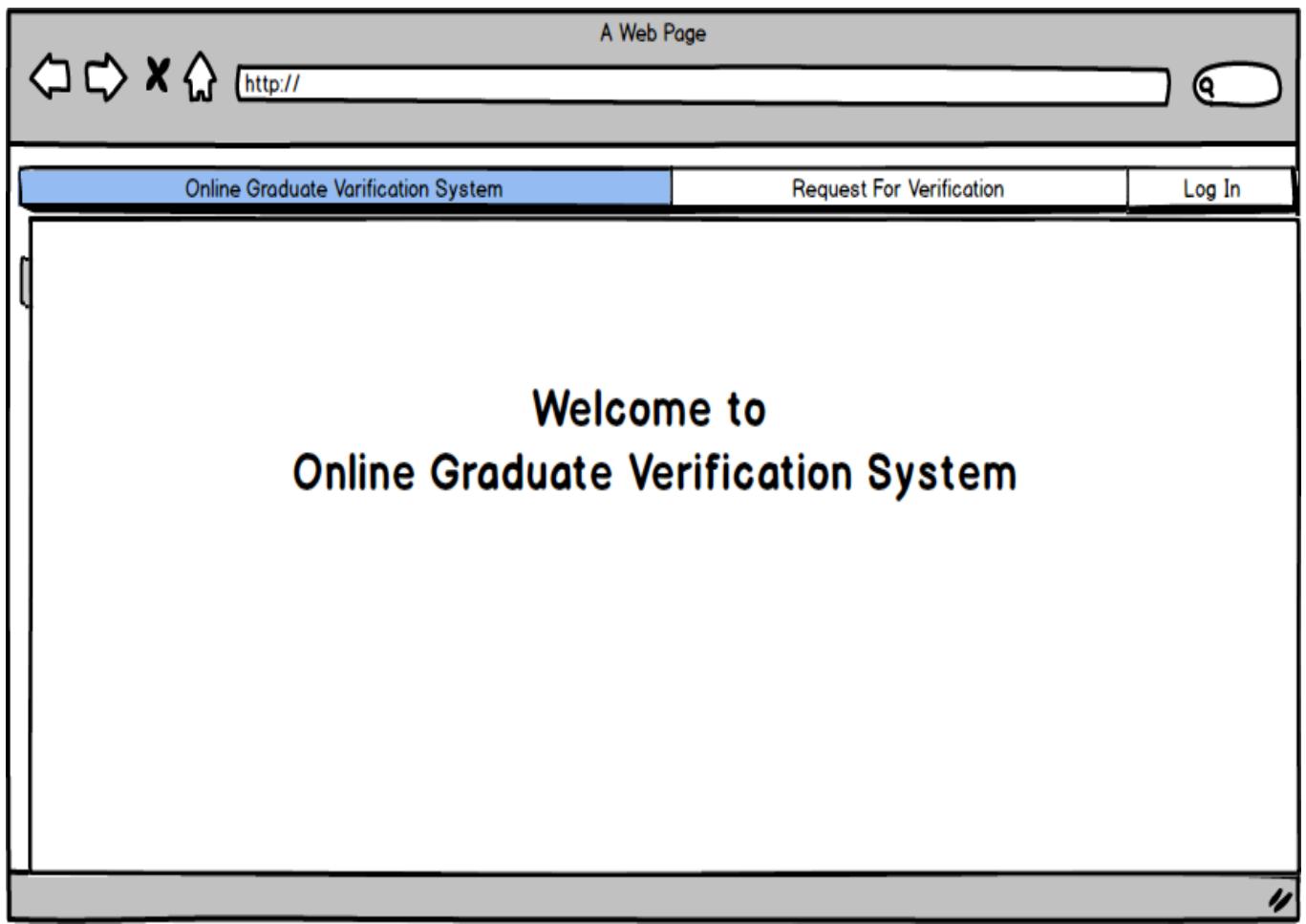


Figure 11.1: Start window

After Clicking Login button the log in window will appear:

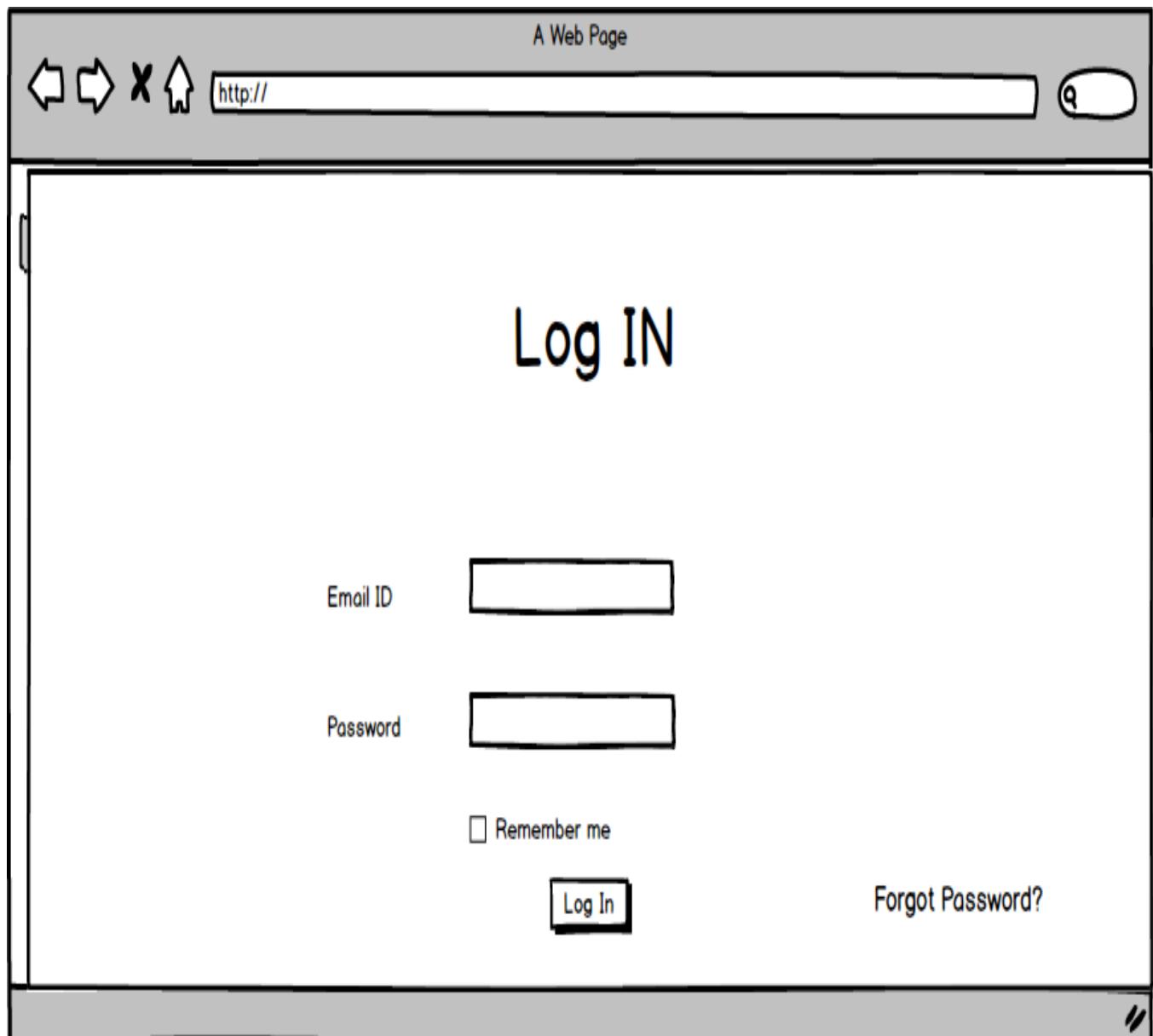


Figure 11.2: Login window

When there is need to request verification for any student the user click the “Request for Verification” button and then will show this:

A screenshot of a web browser window titled "A Web Page". The address bar shows "http://". The main content area has a blue header bar with three buttons: "Online Graduate Verification System", "Request For Verification", and "Log In". Below the header is a section titled "Search Student". Underneath it, "Student Credentials" are listed with four input fields: "Email", "Registration Number", "Date of Birth", and "University Name" (which is a dropdown menu labeled "Select University"). A "Search" button is located at the bottom right of the credentials section.

Figure 11.3: Search Student window

After successful log in as System Admin the following window will appear:

The screenshot shows a web browser window titled "A Web Page". The address bar contains "http://". The main content area has a header "Online Graduate Verification System" with "Profile" and "Log out" links. On the left, there's a vertical sidebar with buttons for "Manage Users", "Manage Universities", "Manage Departments/Institutes", "Manage Result", and "Manage Verification". The "Manage Verification" button is highlighted with a dark grey background. The central area is titled "Manage Users" and contains search/filter controls: "Select User Role" (dropdown), "Select University" (dropdown), "Select Department" (dropdown), "Search" (button), and "Add User" (button). Below these controls is a table listing user data:

Sl	Role	Name	University	Department	Email	Action
1	UGC	Tanvir	Du	IIT	a@xyz	Edit/Delete
2	UGC	Tanvir	Du	IIT	a@xyz	Edit/Delete
3	UGC	Tanvir	Du	IIT	a@xyz	Edit/Delete

Figure 11.4: System Admin dashboard window

By clicking the “Add User” system admin can add users which is like below:

A Web Page

The screenshot shows a web browser window titled "A Web Page". The address bar contains "http://". The main content area has a blue header bar with the text "Online Graduate Verification System", "Profile", and "Log out". To the left, there is a vertical sidebar with menu items: "Manage Users", "Manage Universities", "Manage Departments/Institutes", "Manage Result", and "Manage Verification". The "Manage Verification" item is currently selected and highlighted in grey. The main content area is titled "Add User". It contains five input fields: "Name" (empty), "Email" (empty), "Select User Role" (set to "Student"), "Select University" (set to "University Of Dhaka"), and "Select Department" (set to "IIT"). A "Submit" button is located at the bottom right of the form.

Online Graduate Verification System

Profile Log out

Add User

Name

Email

Select User Role

Student

Select University

University Of Dhaka

Select Department

IIT

Submit

Figure 11.5: Add User window

After log in as UGC Staff the dashboard will be like below:

The screenshot shows a web browser window with the title 'A Web Page'. The address bar contains 'http://'. The main content area has a blue header bar with the text 'Online Graduate Verification System' on the left, 'Profile' in the middle, and 'Log out' on the right. To the left of the main content is a vertical sidebar with five buttons: 'Manage Users', 'Manage Universities' (which is highlighted in blue), 'Manage Departments/Institutes', 'Manage Result', and 'Manage Verification'. The main content area is titled 'Manage Universities' in large bold letters. It features a search bar with a dropdown menu labeled 'Location' and a 'Search' button. To the right of the search bar is a 'Add University' button. Below the search bar is a table with the following data:

SI	University Name	Location	Website	Action
1	University of Dhaka	Dhaka	www.du.ac.bd	Edit/Delete
1	Jahangirnogor University	Savar	www.ju.ac.bd	Edit/Delete

Figure 11.6: UGC Staff dashboard

UGC Staff can add universities by clicking “Add University” button:

The screenshot shows a web browser interface with the title "A Web Page". The address bar contains "http://". Below the address bar is a navigation bar with icons for back, forward, stop, and home, along with a search icon. The main header reads "Online Graduate Verification System" and includes links for "Profile" and "Log out". On the left side, there is a vertical sidebar with buttons for "Manage Users", "Manage Universities" (which is highlighted in blue), "Manage Departments/Institutes", "Manage Result", and "Manage Verification". The main content area is titled "Add University". It contains three input fields labeled "University Name", "Location", and "Website", each with a corresponding text input box. A "Submit" button is located at the bottom right of the form.

Figure 11.7: Add University window

After Login as University Register, the register can manage, add courses:

A Web Page

The screenshot shows a web browser interface with the following elements:

- Header:** A Web Page, Back, Forward, Stop, Refresh, http://, and a search icon.
- Top Bar:** Online Graduate Verification System, Profile, Log out.
- Left Sidebar:** Manage Users, Manage Universities, Manage Departments/Institutes, Manage Result, Manage Verification.
- Main Content:**

Manage Course

. Below it are four dropdown menus: Select University, Select Department, Select Semester, and a Search button. An Add Course button is also present.
- Data Table:** A table listing course details. The columns are SI, University, Inst./Dept, Semester, Course Name, Course Code, Course Credit, and Action. The data is as follows:

SI	University	Inst./Dept	Semester	Course Name	Course Code	Course Credit	Action
1	DU	IIT	1	Software Engineering	SE-101	3	Edit/Delete
2	DU	IIT	1	Software Engineering	SE-101	3	Edit/Delete
3	DU	IIT	1	Software Engineering	SE-101	3	Edit/Delete

Figure 11.8: University Register dashboard

He / She can add courses by clicking “Add Course” button:

A Web Page

The screenshot shows a web page titled "A Web Page" with a header bar containing icons for back, forward, search, and a URL field set to "http://". Below the header is a navigation bar with "Online Graduate Verification System", "Profile", and "Log out" links. A sidebar on the left lists management options: "Manage Users", "Manage Universities", "Manage Departments/Institutes", "Manage Result", and "Manage Verification". The main content area is titled "Create Course". It contains several input fields: "Select University Name" dropdown set to "University Of Dhaka", "Select Department/Inst." dropdown set to "IIT", "Select Semester" dropdown set to "3rd", and three text input fields for "Enter Course Name", "Enter Course Code", and "Enter Course Credit". A "Submit" button is located at the bottom right of the form.

Online Graduate Verification System

Profile Log out

Manage Users

Manage Universities

Manage Departments/Institutes

Manage Result

Manage Verification

Create Course

Select University Name

University Of Dhaka

Select Department/Inst.

IIT

Select Semester

3rd

Enter Course Name

Enter Course Code

Enter Course Credit

Submit

Figure 11.9: Add Course window

Register can also verify the request for any graduate student information and generate the report:

A Web Page

http://

Online Graduate Verification System Profile Log out

Request to verify

Stackholder Information		Student Information	
Name	<input type="text"/>	University	<input type="text"/>
Uni./Company	<input type="text"/>	Department	<input type="text"/>
Designation	<input type="text"/>	Registration	<input type="text"/>
Email	<input type="text"/>	Name	<input type="text"/>
Country	<input type="text"/>	Email	<input type="text"/>

Manage Users
Manage Universities
Manage Departments/Institutes
Manage Result
Manage Verification

Figure 11.10: Verification window

After log in as program officer, he/she can add the result:

A Web Page

The screenshot shows a web browser interface with the title 'A Web Page'. At the top, there are navigation icons (back, forward, stop, home) and a search bar with the URL 'http://'. Below the header, a blue navigation bar contains the text 'Online Graduate Verification System', 'Profile', and 'Log out'. On the left, a vertical sidebar menu lists 'Manage Users', 'Manage Universities', 'Manage Departments/Institutes', 'Manage Result', and 'Manage Verification'. The 'Manage Verification' option is highlighted with a grey background. The main content area is titled 'Add Result'. It contains four dropdown menus: 'Select University' (set to 'University of Dhaka'), 'Select Department' (set to 'IIT'), 'Select Semester' (set to '1st'), and a text input field for 'Student Registration ID'. A 'Submit' button is located at the bottom right of the form.

Online Graduate Verification System

Profile Log out

Add Result

Select University

University of Dhaka

Select Department

IIT

Select Semester

1st

Student Registration ID

Submit

Figure 11.11: Add Result window

After successful log in as a student he/she can view in his/her dashboard with the options of view notifications and online payment:

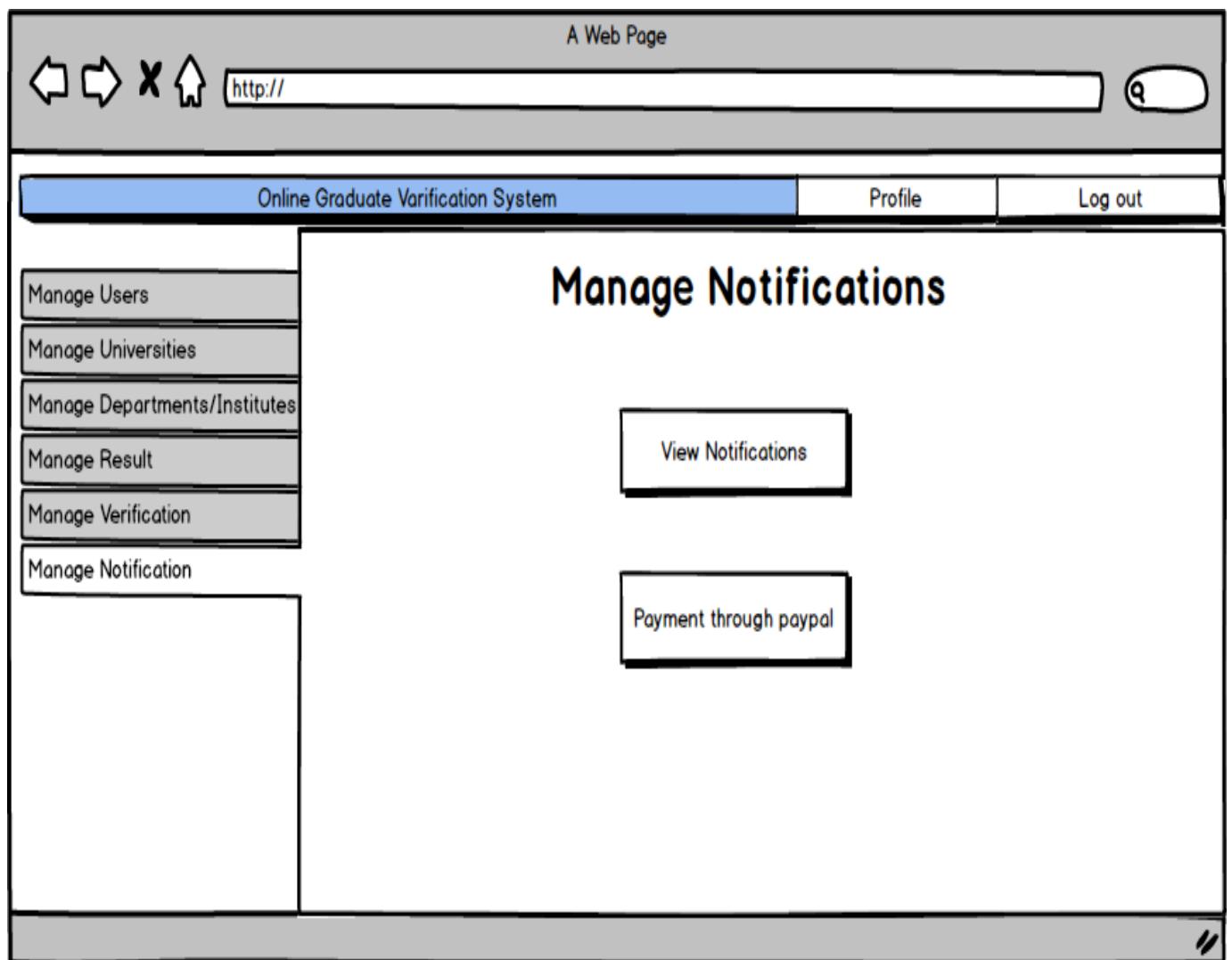


Figure 11.12: Student dashboard

Chapter 12

Reference

1. Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th ed.). Boston, Mass: McGraw-Hill. ISBN 0-07-285318-2
2. Database System Concepts, 5th Ed. ©Silberschatz, Korth and Sudarshan
3. Sommerville, I. Software Engineering, 7th ed. Harlow, UK: Addison Wesley, 2006