Institute of Information Technology

University of Dhaka

S.R.S. Assignment

Dhaka University Form Management System

Course Code: SE406

Submitted by
Fahad Hossain (BSSE 1105)
Kazi Muktadir Ahmed (BSSE 1111)
Raisul Islam Rahat (BSSE 1117)
Shartaz Sajid Nahid (BSSE 1123)
Md. Rakib Trofder (BSSE 1129)
Nazmul Hasan Rupu (BSSE 1135)

Submitted to
Dr. Kazi Muheymin-Us-Sakib
Designation: Professor
Institute of Information Technology
University of Dhaka

Table of Contents

1. Introduction

This chapter is a part of our software requirement specification for the project “Form Management System”. In this chapter, we have focused on the intended audience for our project “Form Management System”.

1.1 Purpose

This document describes the Software Requirement Analysis of Dhaka University Form Management System. It contains all the functional, nonfunctional, supporting requirements and expected requirements. At the same time it establishes a requirement’s baseline for the development of the project. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. But as time goes on, our SRS document is expected to evolve as users and developers work together to validate, clarify and expand its contents. The SRS serves as an official medium of expressing users’ requirements to the developer and provides a common reference point for both the developer team and the stakeholder community.

1.2 Intended Audience

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

• The work of the developer team means the product which they have created can easily be verified by the customer whether it is acceptable to them or not by using this SRS document of the Form Management System.

• Project Managers will be able to plan milestones, delivery date and at the same time ensure whether the developing team is on track or not during the development of the system by using this SRS document.

• 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.

• For developing the system's functionality, this SRS document will be used by the developers. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customers' documented requirements.

• The testers will use this SRS in various phases of testing. They 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.

1.3 Conclusion

This analysis of the audience helped us to focus on the users who will be using our software requirements analysis. This overall document will help each and every person related to this project that includes users, project managers, designers, developers, testers, stakeholders to have a better idea about the project.

1.4. Form Management System Inception

The inception part of our SRS is briefly discussed in this part :

1.4.1 Introduction

Form management system is a Software that will help the students, Teacher and staff of University of Dhaka to go through all kinds of form fill up and payment processes in an efficient way. This document is a sample demand list of what the stakeholders want for their anticipated software. The various features and interfaces of the software will help them to ease their work.

1.4.2 Inception

At first, we entered into the inception stage. This stage includes how our project will start and what are the scopes and limitations. The main goal of this phase is to identify the requirements & demands and then establish some sort of mutual understanding between the software team and the customers. They are the intended users of our software. In order to make this phase effective we took the following steps:

- Identifying the client of our project
- Icebreaking
- Identifying the stakeholders of a course
- Identifying the multiple viewpoints of the stakeholders

1.4.2.1 Identifying the client of our project

First, we have identified the location from where we will start our expedition. Normally students will act as a stakeholder. But there are other things related to this as well. So we have to go through a systematic approach in order to identify all stakeholders. But overall students are expected to be our clients. We have analyzed our requirements with the consent of both of them.

1.4.2.2 Icebreaking

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this phase by talking with the students. The students informally expressed their expectations. We also carried out informal meetings with teachers and staff from the University. The behavior of students, teachers and staff was positive and all of them want this change in form management.

1.4.2.3 Identifying The Stakeholders

Stakeholders refer to any person or group who will be affected directly or indirectly by the system. Stakeholders include end-users who interact with the system and everyone else in an organization who may be affected by its installation. Identification of the stakeholders was done from the information provided by the Teacher and staff from various departments of the University of Dhaka. The stakeholders of our system are given below:

- Student
- Teacher
- Staff and officials

1.4.2.4 Identifying The Multiple Viewpoints Of The Stakeholder

Different stakeholders expect different benefits from the system as every person has his own point of view. So, we have to recognize the requirements from multiple viewpoints. Different viewpoints of the stakeholders about the expected software are given below:

• Students View Point

➢ Smartphone-based system.
➢ Keeping all forms and payment records in a single place.
➢ Making form fill-up process automated.
➢ Getting informed about any notice in an efficient way.
➢ Maintaining the privacy of transaction history for each student.
➢ Open platform for queries.

• Teacher's and official's View Point

➢ Simple and efficient system to manage forms.
➢ Easy user interface for interaction and verification.
➢ Generating forms for different processes.
➢ Storing payment history against student's profile.
➢ Generating receipts automatically from the stored transactions.
➢ Storing detailed student's information.

1.4.3 Conclusion

Our primary goal is to design software which will make form management a lot easier and more efficient, not only for university students but also for the whole community of campus. At the same time, where students can get what information they need . For these reasons, the software will be designed in such a way that it won't be complicated for both the students and teachers who will use it. The software will be so simple that a teacher or staff who does not have any idea about software, will be able to maintain it without any annoyance. Otherwise it will not meet its goal. That is to make the form management process easy and efficient. The software will be designed in such a way as it takes very little time to manage. To make this software project successful, collaboration with the stakeholders i.e. the students and Teacher was a main priority that what they want, how the software will work, how it can be more effective for all.

2. Brief Description Of The Current Management System

The purpose of our project is to automate the communication among hall, and department/institute, bank, university administration (Registrar office). The current student registration system of the University of Dhaka is based on the 19 resident halls that are in it. Whenever a student has to go through some kind of official work the student almost every time needs to include all of these entities. But the communication between them is almost non-existent. So the student has to go the extra mile to convey some simple information from one place to another. Our goal is to lessen the hassle of the students and automate the communication among these entities.

Currently, a student has to go through four types of official query/payment in his student life. They are -

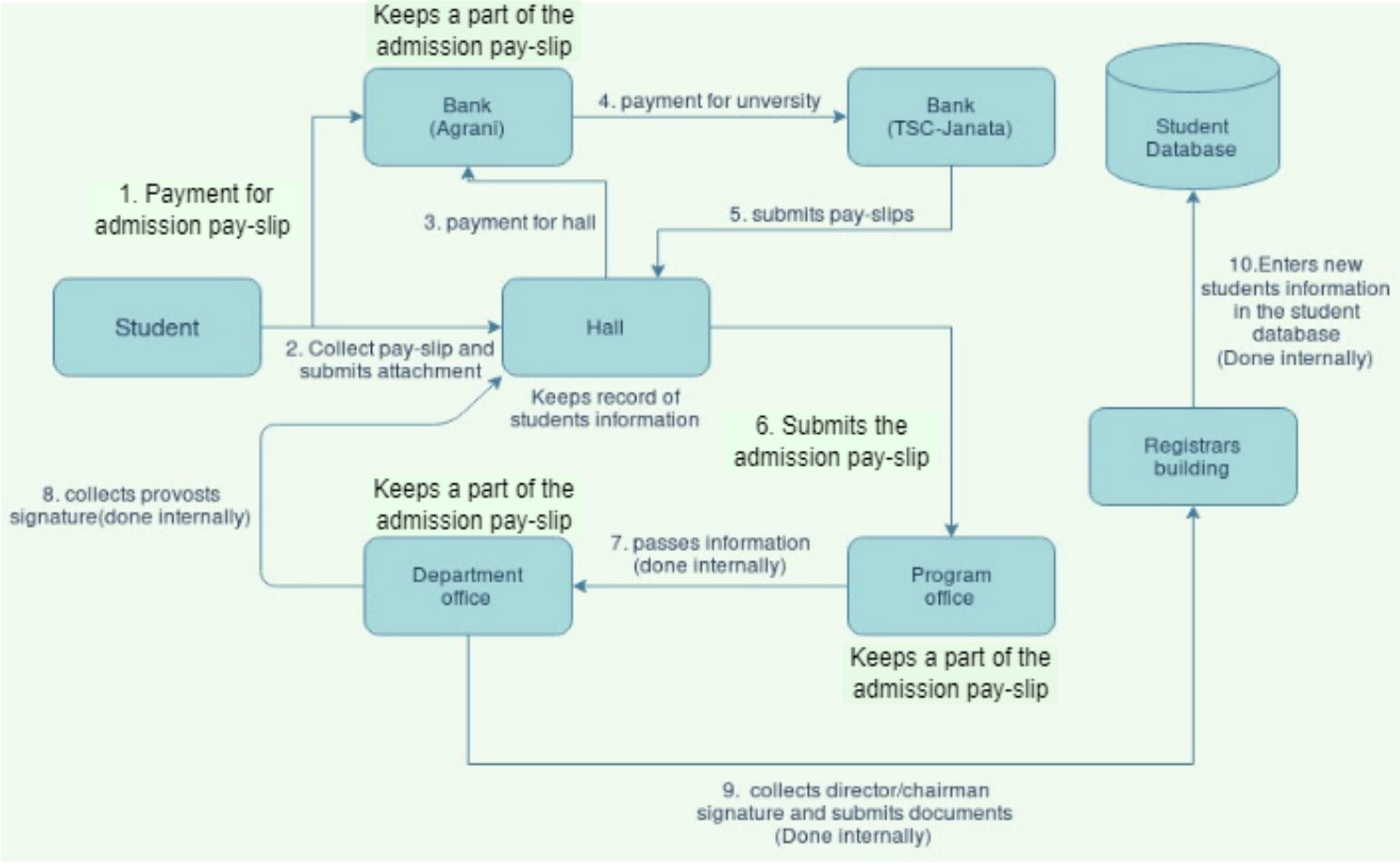
1. Undergraduate admission.
2. Promotion to the next academic year
3. Semester fees and Admit Card
4. Certification

We describe this in detail below.

2.1. Undergraduate admission

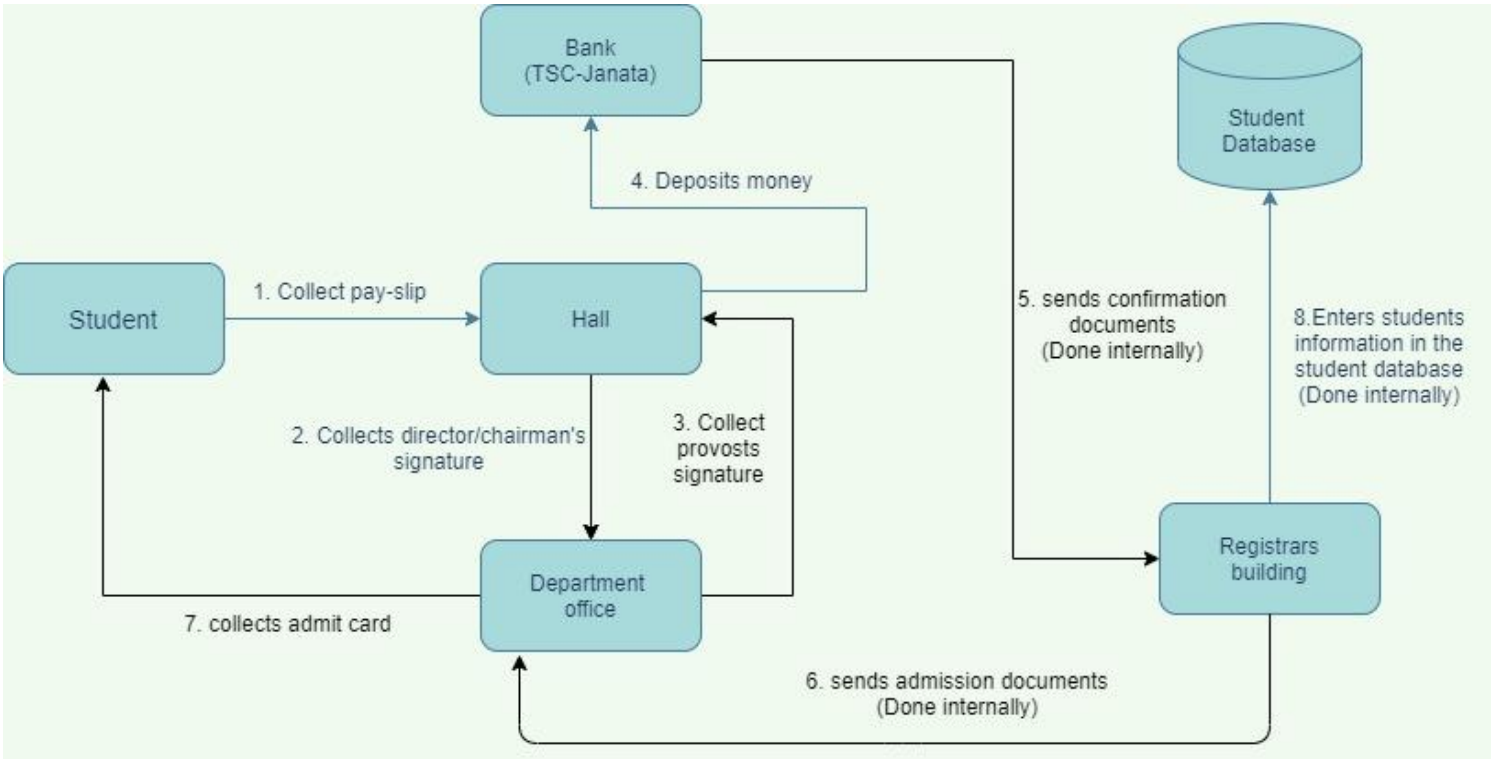
1. Take payslip from Dept.
2. Deposit department fee to Agrani Bank, Carzon Hall Branch
3. Go to the attached hall.
4. Hall verifies students from the given University list.
5. Students give their required documents to the Hall.
6. Hall entry students information to their record book.
7. Deposite hall fee to Agrani Bank, Carzon Hall Branch.
8. Get University’s payment receipt from Hall.

9. Deposit University fee to Janata Bank, TSC branch.
10. Take slip and documents and give it to the Department.
11. Department verifies all information of students and sends it to the hall.
12. Hall verifies all information of students and sends it to the department again.
13. Department sends all verified documents to .Register Building.
14. .Register Building entry students' information to their database and complete the system.



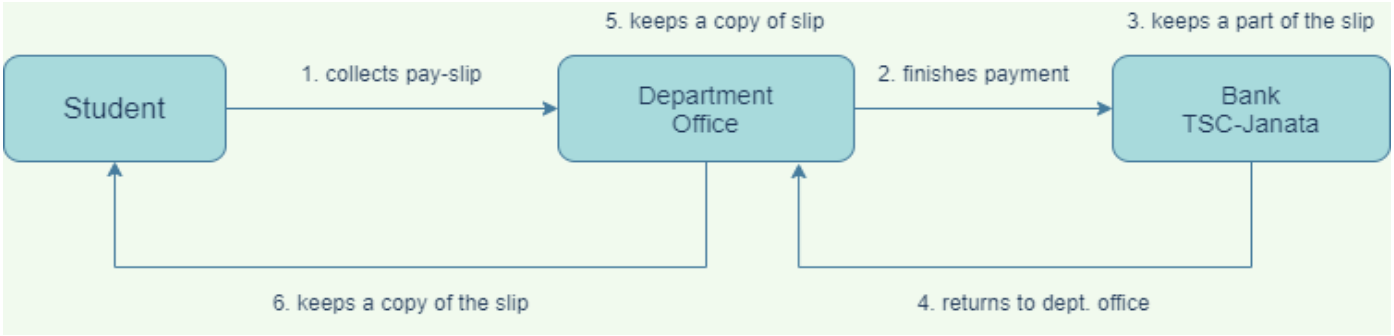
2.2. Admit Card

1. Take it from the Hall and fill it up.
2. Get Director/Chairman signature.
3. Get Provost Signature.
4. Deposit Money to Janata Bank, TSC branch
5. Bank sends the form to the Register Building.
6. The Register Building sends an Admit card to Dept.
7. Students get admit cards from the Dept Office.



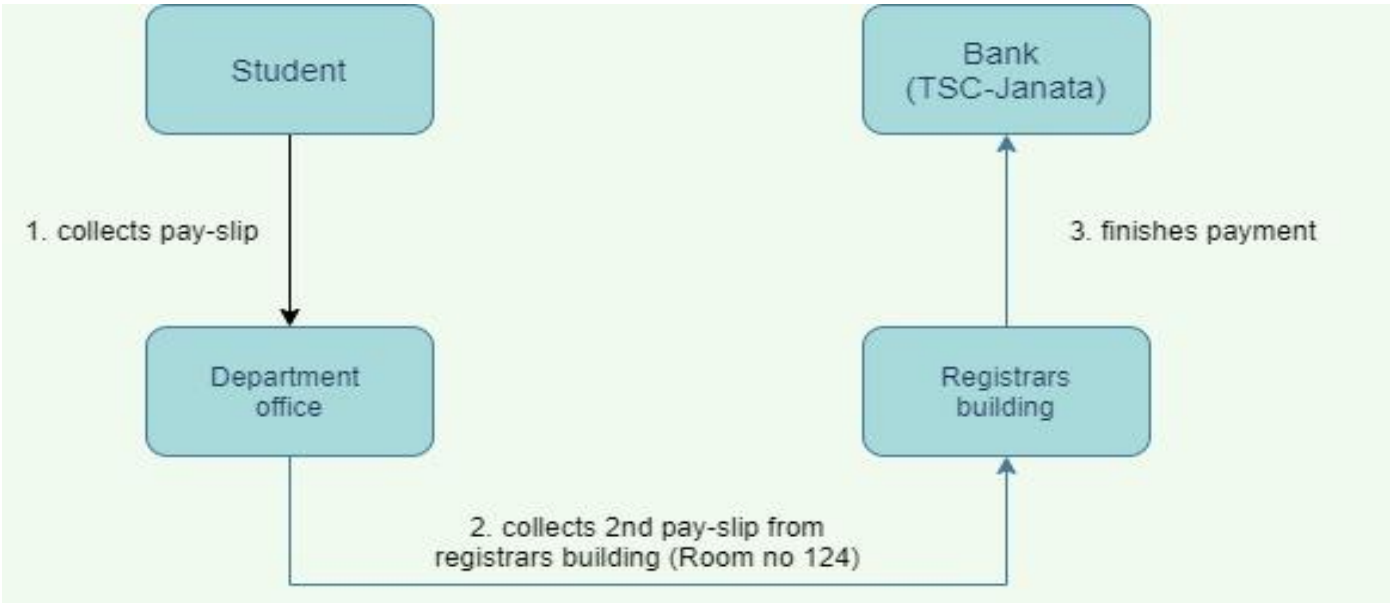
2.3. Semester fee submission

1. Takes payment-slip from department office
2. Goes to bank (TSC- Janata) to finish the payment
3. Bank keeps a part of the slip as proof of payment.
4. Returns to the department office. Office keeps a part of the slip as proof of payment.
5. Student keeps a part of the slip as proof of payment.



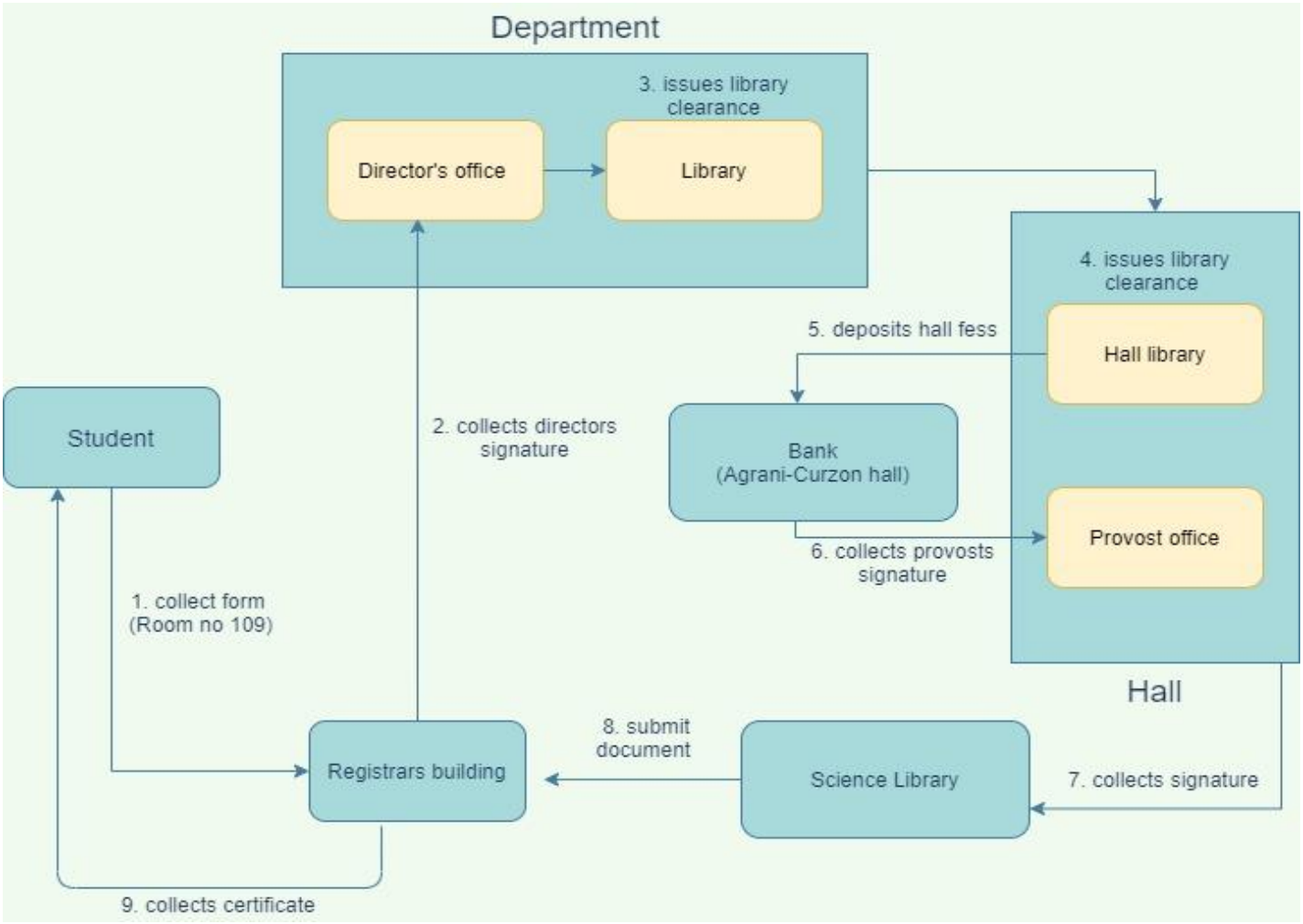
2.4. Next academic year Admission

1.Take a pay-slip from the Department and fill it up.
2.Take another form from Room No 124 in the Registrar's Building for hall.
3.Deposit Money to Janata Bank, TSC branch.
4.Bank sends information to the Register Building.
5.Register Building entry information and complete the process.



2.5. Certification

1. Take form from room 109 in the register building and fill up.
2. Take the director's sign
3. Take IIT library clearance
4. Take Hall library clearance
5.Deposite hall's fee to Agrani Bank, Carzon Hall Branch.
6. Take hall Provost sign
7. Take the science library sign
8. Submit to room 109 in the register building.
9.Registrar Building verifies all documents & information and prepares Certificate.



2.6. Transcript

1. Collect Form (50 tk) from the registrar building.
2. Collect Chairman/Director Signature.
3.Confirm Seminar Library Clearance.
4.Confirm Central Library Clearance.
5.Depostite Hall Fees.
6.Collect Provost Signature.
7.Submit Document to Registrar Building.
8.Collect Transcript.

3. Elicitation Of DU_FMS

So far we have discussed the Inception phase of our project. Now we need to focus on the Elicitation phase. So this chapter specifies the Elicitation phase.

3.1 Introduction

Elicitation is a part of requirements engineering that is. We have faced many difficulties, like understanding the problems, making questions for teachers, getting appointments from university staff and officials in spite of their busy schedule, making them understand the power of automation. Despite not being easy to gather requirements within a very short time, we have overcome all problems in a systematic manner. We have done several meetings and finalized the requirements for our software.

3.2 Eliciting Requirements

Elicitation phase is mainly combining the elements for problem solving, elaboration, negotiation and specification. Without the collaboration of the stakeholder eliciting would have been really hard. We have finished the following tasks for eliciting requirements-

- Collaborative Solution
- Quality Function Deployment
- Usage Scenarios

3.2.1 Collaborative Solution

We have met with many students and university officials and staff in the Inception phase . These meetings created an indecisive state and we could identify what the real problem is. To solve this problem, we have met with the stakeholders several times and came up with the solution which eventually helped us to elicit the requirements.

3.2.2 Quality Function Deployment

What is QFD: QFD means Quality Function Deployment. It is a structured approach of defining customer needs or requirements and translating them into specific plans to produce products to meet those needs.

There are 3 types of requirements in **Quality Function Deployment**. They are:

1. **Normal Requirements**: Normal requirements reflect objectives and goals stated for a product or system during meetings with the customer. Those are the basic requirements that fulfill client satisfaction.

2. **Expected Requirements**: These requirements are so obvious that the customer need not explicitly state them. Their absence can create significant dissatisfaction.

3. **Exciting Requirements**: These requirements are for features that go beyond the user's expectations and prove to be very satisfying when present.

We are working to automate the undergraduate admission and other payment or form related processes in University of Dhaka. This is one of the most complex and time-consuming tasks, Which is managed almost fully manually. We will try to make it automated as much as possible. To make this complex process simple, easy, and reliable. It also reduces human involvement, making it fast to complete the whole process. It increases accuracy, makes the process more efficient and productive.

3.1. Normal Requirements

- ❏ User Registration & Authentication
- ❏ Payment process can be handled manually or virtual transaction methods (Bikash, Nagad, Rocket)
- ❏ Students enter amounts of money, transaction ID, session, registration no to complete their payment transaction.
- ❏ After completing the payment transaction, the Director/Chairman & Provost approved student registration.
- ❏ The Register Building Officer will store and maintain students' information.
- ❏ Both the Department office and Hall office will update students' information.
- ❏ Students will fill up a registration form if they want to change their personal information.

3.2. Expected Requirements

- ❏ Easy to use.
- ❏ Autofill system will available for form
- ❏ Student personal information updating and correcting scope after form fills up.
- ❏ All information will be from anywhere & any time.
- ❏ Money transaction process will be customized with university students.

3.3. Exciting Requirements

- ❏ Expired student validation will automatically be detected by the system.
- ❏ Register officers will be able to detect problems if any provost/director or hall/dept. Can not work properly.
- ❏ Users are informed/notified by email/SMS to complete their tasks.

4. Use Case Scenario of DU Form Automation System

4.1. Introduction:

The purpose of our project is to automate the communication among hall, and department/institute, bank, university administration (Registrar office). The current student registration system of the University of Dhaka is based on the 19 resident halls that are in it. Whenever a student has to go through some kind of official work the student almost every time needs to include all of these entities. But the communication between them is almost non-existent. So the student has to go the extra mile to convey some simple information from one place to another. Our goal is to lessen the hassle of the students and automate the communication among these entities.

4.2. Target user base:

As the system handles all of the official transitions of a form in Dhaka University, all those who are directly related to these works must be registered to the system. They are the primary user base of the system. They consist of Students, Teachers, Department / Hall / Library / Bank / Registrars building staff.

Dhaka University students are the most prominent user base of the system. They can be divided into 2 categories -

a. The regular students of Dhaka University. They are identified by the registration number given to them by the university.
b. Newly admitted students who are yet to be registered into the system. They are identified by the admission exam roll and merit position from the admission exam (this information is taken from the admit card of the admission exam).

Every teacher in Dhaka university must be registered to the system. Depending on the designation, a teacher can have some administrative privileges in the system. Below are the designation that has administrative privilege -

a. Director (Institute)
b. Chairman (Department)
c. Dean (Faculty)
d. Provost (Hall)
e. House tutor (Hall)

Each of the staff will perform a certain role in the system. They are separated by their posts -

a. Department Staff
b. Hall Staff
c. University Stall
d. Library Staff

Besides all of them, anyone can access the general notice board of the system.

We have 5 parts in the DU Form Automation System. Those are:

1. User Management
2. Form Interface
3. Payment System
4. Verification System
5. Notification System

4.3. User management:

4.3.1. Sign Up

New users must create an account to be officially registered to the system. To create an account one needs to provide the following information:

- Full Name
- Contact number
- Email address
- Address
- User Type

There are four types of users:

1. Student
2. Provost/Chairman/Director/House Tutor/Dean
3. Librarian/ Library Staff/ University Staff
4. Official
5. Bank

Based on the selected user type, the user must provide some additional information to finish the registration process. This information will be different for each type of user.

The required information for each user type:

1. **Student**
    - Father's Name
    - Mother's Name
    - Blood group
    - Religion
    - Nationality
    - Present Address
    - Permanent Address
    - S.S.C. Information

    - Board
    - Year
    - Roll
    - G.P.A.

- H.S.C. Information
    - Board
    - Year
    - Roll
    - G.P.A.
- Department Name
- Hall name
- Residential Type
- Class Roll
- University Registration no
- Session
- University merit position

2. **Teacher**
    - Teacher id
    - Department name
    - Designation
    - Type
    - Hall name(Provost only)

3. **Library Staff**
    - Employee id
    - Type
    - Library name

4. **Department/Hall/University Official**
    - Employee id
    - Office post
    - Office name

5. **Bank**
    - Bank id
    - Branch name
    - Bank name

4.3.2. Login

Users can log in using their Email/Phone number and Password. Before 1st login users must verify their email and phone number.

4.3.3. Password Recovery

If a user forgets his/her password, there will be a password recovery system available. Which have 2 options:

1. Verification link can be sent by email.
2. OTP sent through users' phone numbers.

4.4. Form Interface

There are actually 6 types of fee forms. For each type of transaction, the system will take only the necessary information from the user and the auto-filling mechanism will fill the parts which are presented by the University, halls, or department.

The required input from the user and the detailed bills related to each form is described here:

1. Admission Fee Form
2. Semester Fee Form
3. Year Session Fee Form
4. Admit Card Fee Form
5. Certificate Application Form
6. Transcription Fee Form

4.4.1. Admission Fee Form

Fill Up by Students:

-Admission type
- ❏ Regular
- ❏ Irregular

-Any Fine available?
- ❏ Yes
- ❏ No

-Deans Committee Fine(This option enable if select Yes)
-Student Type:
- ❏ Bangladesh
- ❏ SAARC
- ❏ Outside SAARC

-Other fees

-Select Transaction Type
- ❏ Bank
- ❏ Online Banking
- ❏ Mobile Banking

-Add Transaction ID
-Amount of Money
-Account Name/ID/Number

Auto Fill:

-Admission Fee
-Re-admission Fee(This option enable if select Irregular)
-Monthly fee
-Hall Union Fee
-University Union Fee
-Hall Sports Fee
-University Union Fee
-Residential Visit Fee
-Library Deposit Fee
-Course Examination Fee
-University Registration Fee
-Students Welfare Fee
-Admission Form Fee
-Smart Card Fee
-Pay Slip Fee
-Hall Seat Fee
-Dishes Fee
-Payment Fine
-Library Fine
-TC Fee
-Health Card Fee
-Library Development Fee
-Transport Fee
-Exam Center Fee
-Session Fee
-Hon's Instruction Fee
-Academic Calendar Fee
-Transport Infrastructure Development Fee
-Laboratory Fee
-Computer Fee
-Proctorial Service Fee
-Counseling Service Fee
-Transcript Fee
-Non-Collegiate Fee

4.4.1.2. Hall Fee Form during Admission

There are 10 parameters for the hall fee forms during admission. Those are:

-Admission type
- ❏ Regular
- ❏ Irregular

-Hall name

-Select Transaction Type
- ❏ Bank
- ❏ Online Banking
- ❏ Mobile Banking

-Add Transaction ID
-Amount of Money
-Account Name/ID/Number

Auto Fill:
-Admission Fee
-Re-admission Fee(This option enable if select Irregular)
-M.Phil/Ph.D. fee
-Club fee
-IT fee
-Student fund
-ID card fee
-Certificate processing fee
-Testimonial
-Others(Magazine)

4.4.2. Semester Fee Form

There is only one parameter for department semester fees.

-Semester fee

-Select Transaction Type
- ❏ Bank
- ❏ Online Banking
- ❏ Mobile Banking

-Add Transaction ID
-Amount of Money
-Account Name/ID/Number

4.4.3. Admit Card Form

The admit card form consists of 3 parts. They are -
1. Admit Card
2. Application for Exam
3. Student Information Form for Exam

Each of these parts have individual parameters which need to be filled out by the students. The system has a simple user interface to take necessary inputs from students to generate a preliminary draft of the form. The parameters which will be taken as input in this interface are as follows:-

Fill Up by Students:

-semester & academic year
-exam year
-course name & code
1. Course No -1

2. Course No -2
3. Course No -3
4. Course No -4
5. Course No -5
6. Course No -6

-punishment duration
__years___months___days

-Previous Exam information
-Name
-exam type
-exam roll
-student's signature
-captcha

After the completion of the step the system creates a preliminary draft of the form. The draft contains all of the parameters necessary for the admit card form. The underlined parameters are auto-filled by taking formation from the student database. While the *italic* parameters are to be filled by taking input from other entities. The preliminary draft of the form contains -

1. Admit Card
-semester & academic year
-exam year
-student's signature
-controller of examination’s signature
-exam date
Auto Fill :
-name
-class/department roll
-session
-university registration no
-department name

-father’s name
-mother’s name
-ssc information
    -gpa
    -year
    -roll
-hsc information
    -gpa
    -year
    -roll

2. Application for Exam

This part of the draft contains all the parameters from the “Admit card” portion with 3 new additional parameters which must be verified/filled by a non-student user. They are -
-payment received approval/signature from the bank
-director/chairman's signature
-hall provosts signature

3. Student Information Form for Exam

Autofill:
-Present Address
    -Village
    -Post Office
    -Post Code
    -Police Station
    -District

-Date of Birth
-Age: __years___months___days
-Nationality
-Religion
-Religion Class

-Punishment Duration:

__years___months___days

-Previous Exam information
    -Name
    - Type
    - Roll
-Examination Medium
-*director/chairman's signature*
-S.S.C. information
    -Board
    -Year
    -Roll
    -G.P.A.

-H.S.C. information
    -Board
    -Year
    -Roll
    -G.P.A.

4.4.4. Year Session Fee Form

<u>Fill Up by Students:</u>

-Admission type
- ❏ Regular
- ❏ Irregular

-Any Fine available?
- ❏ Yes
- ❏ No

-Deans Committee Fine(This option enable if select Yes)
-Student Type:
- ❏ Bangladesh
- ❏ SAARC
- ❏ Outside SAARC

-Other fees

-Select Transaction Type
- ❏ Bank
- ❏ Online Banking
- ❏ Mobile Banking

-Add Transaction ID
-Amount of Money
-Account Name/ID/Number

Auto Fill:
-Admission Fee
-Re-admission Fee(This option enable if select Irregular)
-Monthly fee
-Hall Union Fee
-University Union Fee
-Hall Sports Fee
-University Union Fee
-Residential Visit Fee
-Library Deposit Fee
-Course Examination Fee
-University Registration Fee
-Students Welfare Fee
-Admission Form Fee
-Smart Card Fee
-Pay Slip Fee

-Hall Seat Fee
-Dishes Fee
-Payment Fine
-Library Fine
-TC Fee
-Health Card Fee
-Library Development Fee
-Transport Fee
-Exam Center Fee
-Session Fee
-Hon's Instruction Fee
-Academic Calendar Fee
-Transport Infrastructure Development Fee
-Laboratory Fee
-Computer Fee
-Proctorial Service Fee
-Counseling Service Fee
-Transcript Fee
-Non-Collegiate Fee

4.4.5. Transcript Fee Form

-Transcript Fee
-University Staff Signature

4.4.6. Certificate Application Form

-Certificate Application Fee
-Provost Signature
-Library Staff Clearance Signature

4.5. Payment system

- ❖ DU Form Automation system will use the API of online payment system methods like Bkash, Nagad, Ucash, Rocket or Sure Cash, MCash. We will use this API to create a user interface for users to simplify the process.
- ❖ This system also supports the bank's online payment system.
- ❖ If someone is unable to pay online then he can also pay money through Janata or Sonali Bank manually, and then he has to enter the transaction id on our system to confirm the transaction.

In every transaction form we can see these feature:

- ➢ Select Transaction Type
  - ○ Bank
  - ○ Online Banking
  - ○ Mobile Banking
- ➢ Add Transaction ID
- ➢ Amount of Money
- ➢ Account Name/ID/Number

4.6. Form Verify System

There are 3 types of an admin panel that verify students' all information during admission, taking admit cards, certificates, or transcripts.
Those 3 admins are:

1) **Hall Provost:** Who verifies documents during admission, taking admit card, certificate, or transcripts.
2) **Director/Chairman**: Who verifies documents during taking admit card, certificate, or transcripts.
3) **Staff**: Who verifies documents during taking certificates and transcripts.

Each will verify their students each of them separately or in a group. He will also issue or report any of the student forms if there is something problem or incomplete information. Hall Provost\Director/Chairman\Librarian can see the status of how many students are there, how many students verified or incomplete.

Verify System UI contains these features:

- Select Role
    - Hall Provost
    - Director/Chairman
    - Staff

- Type of Activities
    - Semester Exam
    - Year Final Exam
    - Certificate
    - Transcription
    - Admit Card
    - Others

- Verification Interface
    - Hall Name
    - Dept Name
    - Last Date of Verification
    - Number of Students
    - Number of Students Verified
    - Number of Students Incomplete
    - Students Status Preview
    - Verification Digital Signature
    - Issue/Report Student Form

4.7. Notification Dashboard

The dashboard will notify the user if there's any kind of pending payment. This dashboard will also feature a payment history option that will record all kinds of transactions made by the user. This also notifies upcoming exam or form fillup notices, verification status, etc.

The Notification System Dashboard looks like this.

- ➢ Today's Date
- ➢ Due Form Fill UP (if available)
- ➢ Time & Day Remaining for latest Form Fill Up(if available)
- ➢ Upcoming Form Fill Up
- ➢ Form Fill Up Verification Status
- ➢ Transaction History & Status (if available)

5. Use Case Diagram of DU Form Automation System

5.0. What is a Use Case Diagram?

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams to model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a website. The "actors" are people or entities operating under defined roles within the system.

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration.

5.0.1. Primary Actor

Primary actors interact directly to achieve the required system function and derive the intended benefit from the system. They work directly and frequently with the software. They produce some information and consume some information too.

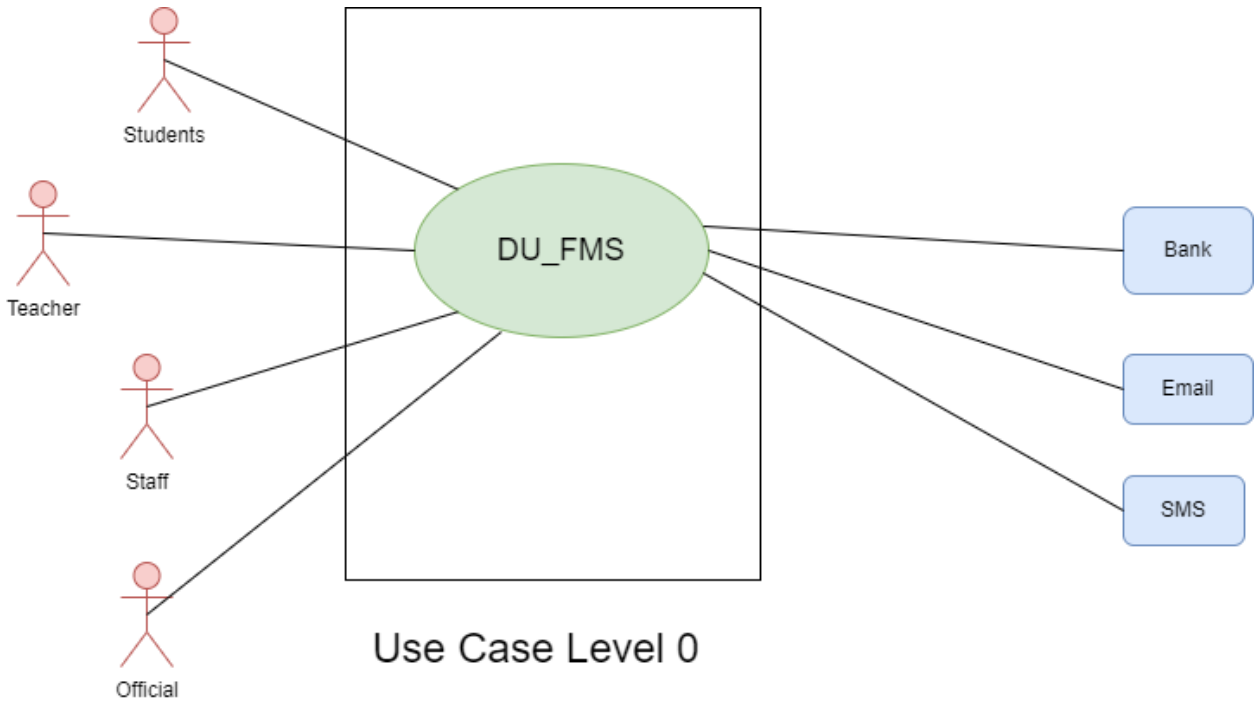
5.0.2 Secondary Actor

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

Use Case diagram gives a non-technical view of the overall system.

5.1. Level 0

Name: DU_FMS (DU Form Management System)
Primary Actor: Students, Teacher, Staff
Secondary Actor: Bank, Email, SMS, Official

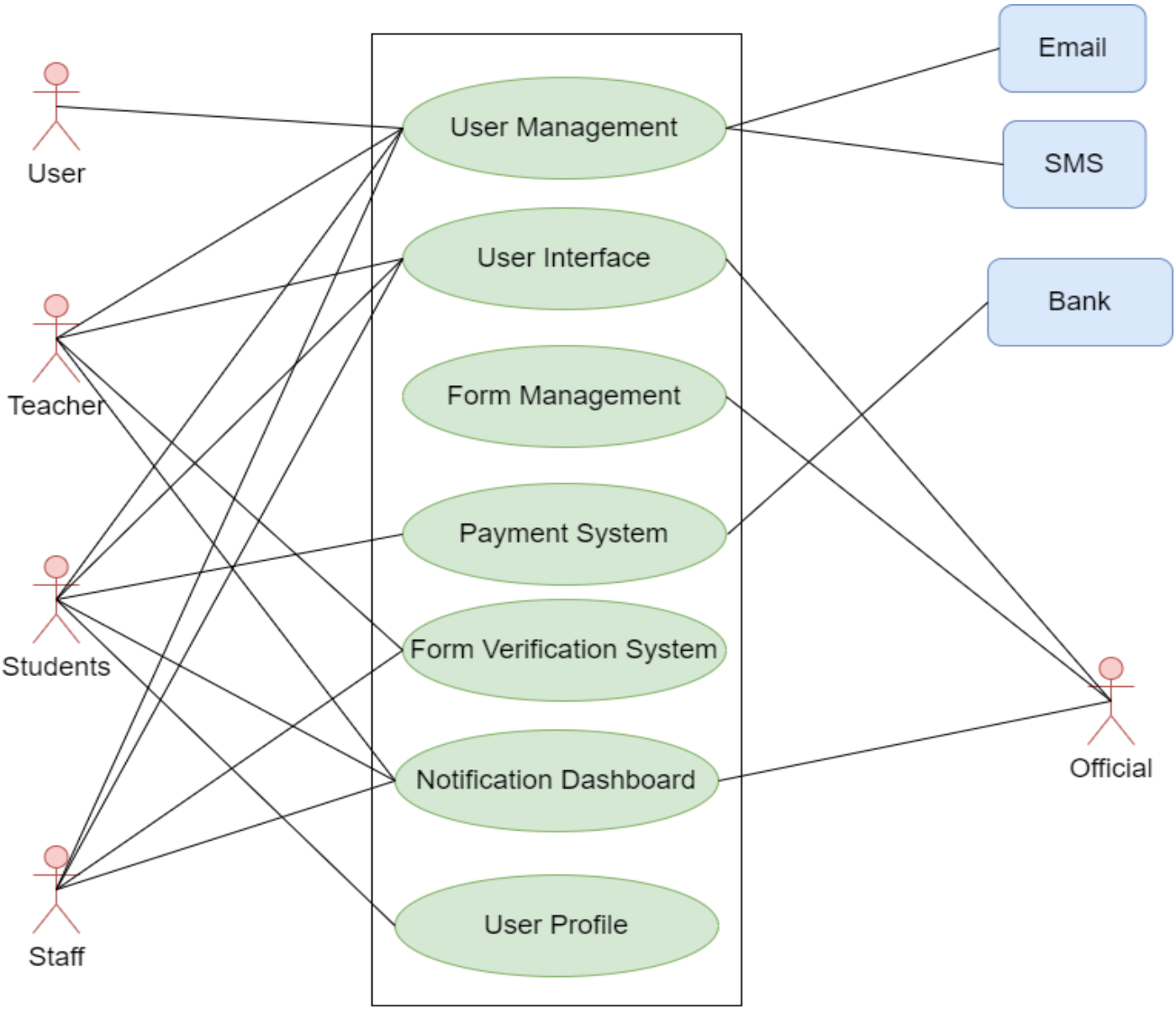


5.1.1. Description of use case diagram level-0

This is an overall view of all the primary and secondary actors altogether. As students, teachers, and library staff are the only ones who will initiate a task for the system, they are considered as the primary actors. University staff are the ones who will respond to the task initiated by the primary actors thus being the secondary actors. Bank, mobile banking system, online banking system, email, SMS are external software modules used by the system.

5.2. Level 1

Name: DU_FMS (DU Form Management System)
Primary Actor: Students, Teacher, Staff
Secondary Actor: Bank, Email, SMS, Official



Use Case Level 1

5.2.1. Description of use case diagram level-1

5.2.1.1. User Management

Users must create an account and then log into the system to get the full functionality of the system. He/she can update his/her profile and can recover a password if forgotten.

5.2.1.2. User Interface

Users can view the general notice board or can log in to view their dashboard. They can also initiate a form fill-up task.

5.2.1.3. Form Management

Students can choose a form from the menu and can start the process by giving necessary information into the form.

5.2.1.4. Payment System

Students can use mobile banking systems, online banking systems,s or manual payment systems to complete the payment process.

5.2.1.5. Form Verification System

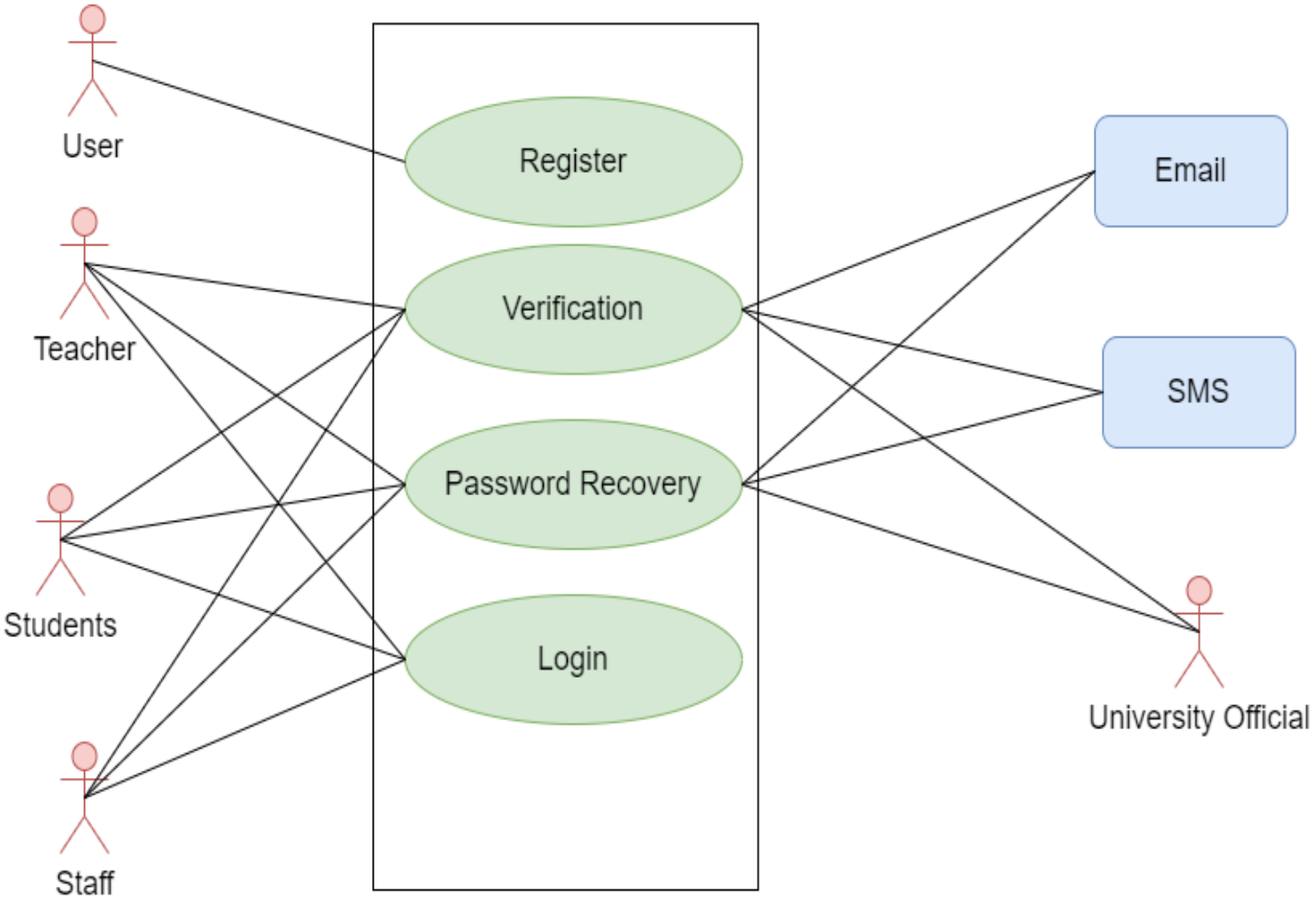
Teacher and university officials can verify the information given by the students in a given form.

5.2.1.6. Notice Dashboard

Users can view any task/form pending completion.

5.3. Level 1.1

Name: User Management
Primary Actor: Students, Teacher, Staff
Secondary Actor: Official, Email, SMS



Use Case Level 1.1

5.3.1. Description of use case diagram level-1.1

5.3.1.1. Registration

To create an account, users must provide their information to the system. They must choose a user type and depending on their user type they must provide additional information.

5.3.1.2. Verification

Users must verify their account before they can log into their account. They can do it using either their phone number, email id or by manually entering the verification code given to them by the admin.

5.3.1.3. Password Recovery

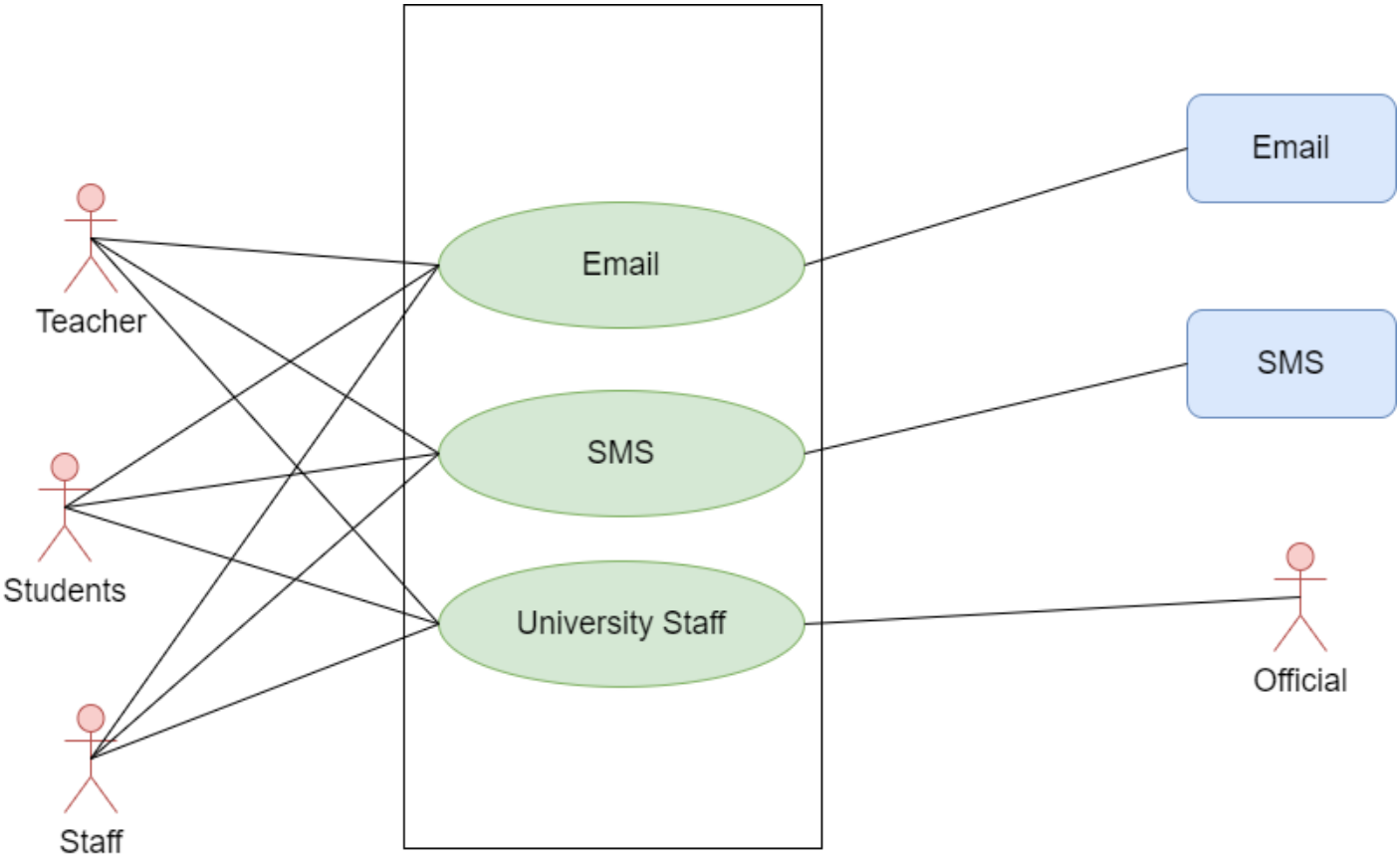
A user can recover his/her password if forgotten, by using his/her email or phone number or by contacting the admin manually.

5.3.1.4. Login

Users will log into the system by using his/her registered email-id/ phone number and password.

5.4. Level 1.1.2

Name: Verification
Primary Actor: Students, Teacher, Staff
Secondary Actor: Official, Email, SMS



Use Case Level 1.1.2

5.4.1. Description of use case diagram level-1.1.2

5.4.1.1. Verification By Email

A verification link will be sent to the user's provided email address from the system. Users can verify their accounts by clicking that link.

5.4.1.2. Verification By SMS

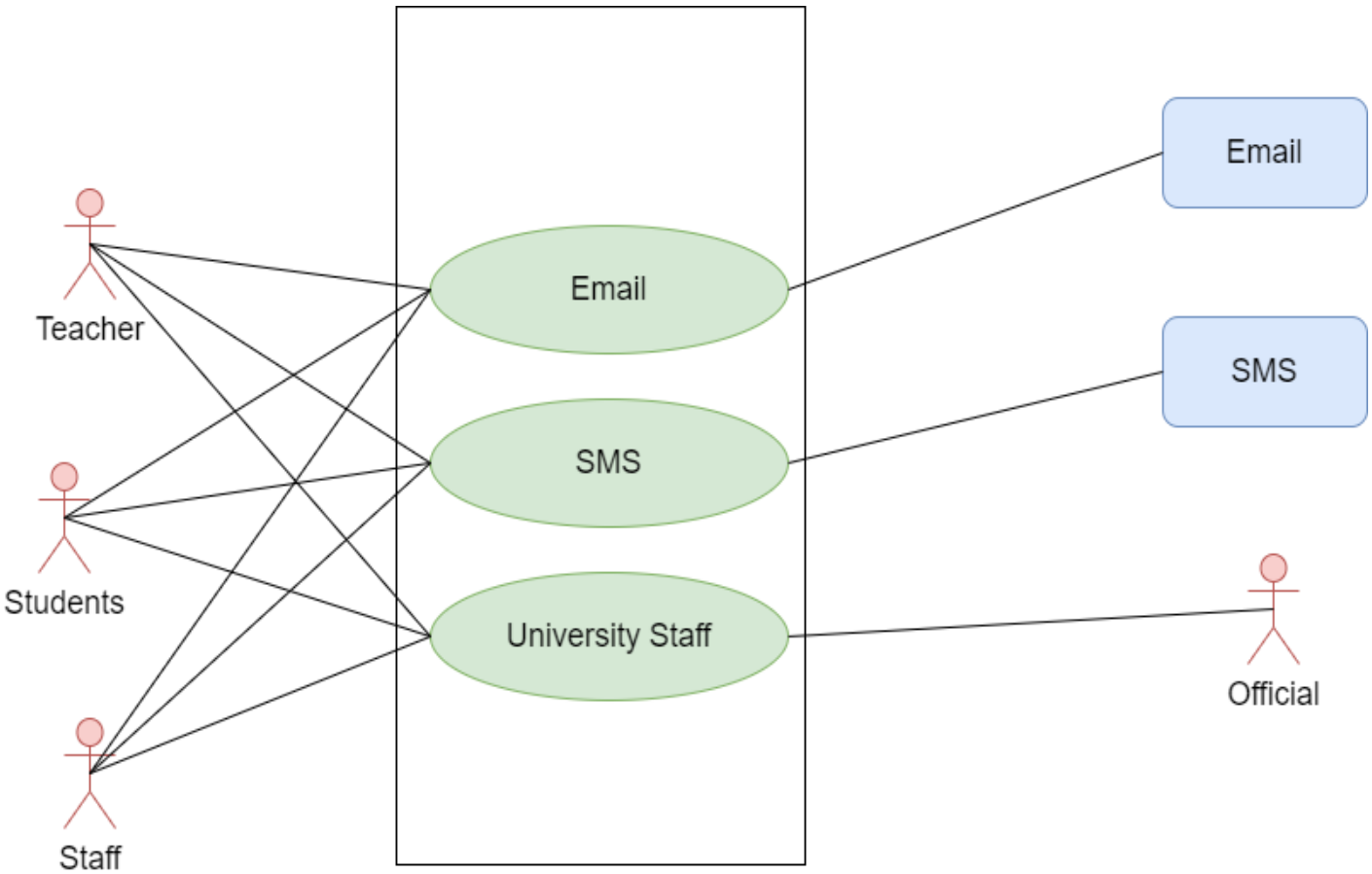
A one-time OTP code will be generated and sent to the users' phone numbers from the system. Users can verify their account by typing in that OTP in the system.

5.4.1.3. Verification By Official

Users can contact the Official (admin) and can get a one-time verification code. This code can be used to verify the account.

5.5. Level 1.1.3

Name: Password Recovery
Primary Actor: Students, Teacher, Staff
Secondary Actor: Official, Email, Sms



Use Case Level 1.1.3

5.5.1. Description of use case diagram level-1.1.3

5.5.1.1. Verification By Email

A password recovery link will be sent to the user's provided email address from the system. Users can reset their account passwords by clicking that link.

5.5.1.2. Verification By SMS

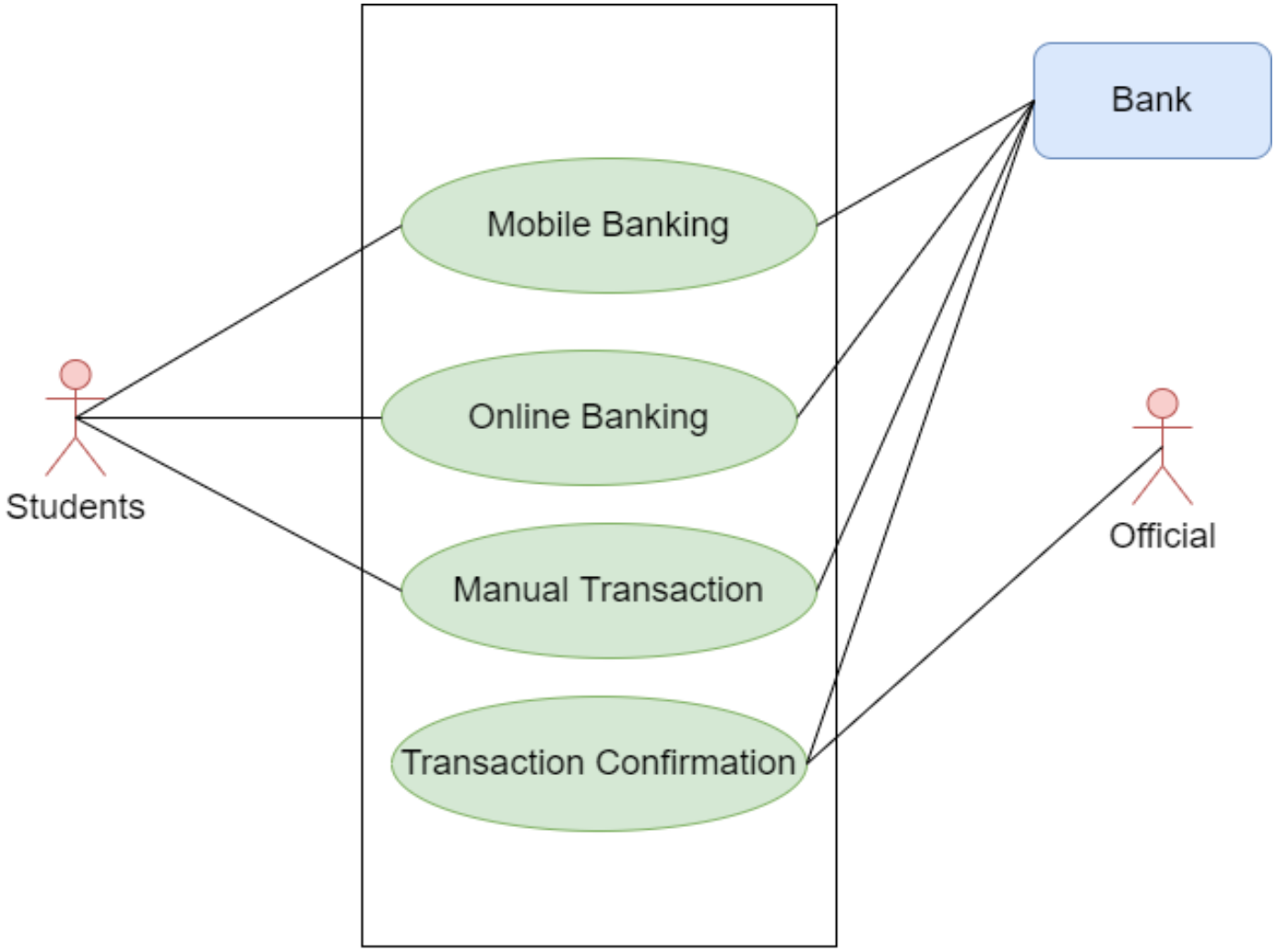
A one-time OTP code will be generated and sent to the user's phone number from the system. Users can log into their account by typing in that OTP in the system and from there they can change their password.

5.5.1.3. Verification By Official

Official: Users can contact the Official(admin) and can get a one-time login code. This code can be used to log into the account and from there they can change their password.

5.6. Level 1.4

Name: Payment system
Primary Actor: Students
Secondary Actor: Official, Department staff, Hall staff, Bank, Mobile Banking system, Online Banking system



Use Case Level 1.4

5.6.1. Description of use case diagram level-1.4

5.6.1.1. Payment by Mobile Banking

Students can either use the mobile banking system in the program or use their mobile phone to pay via a mobile banking account. Users must input the transaction id in the system for payment completion.

5.6.1.2. Payment by Online Banking

Students can either use the online banking system in the program or pay via an online banking account.

5.6.1.3. Payment by Manual Transaction

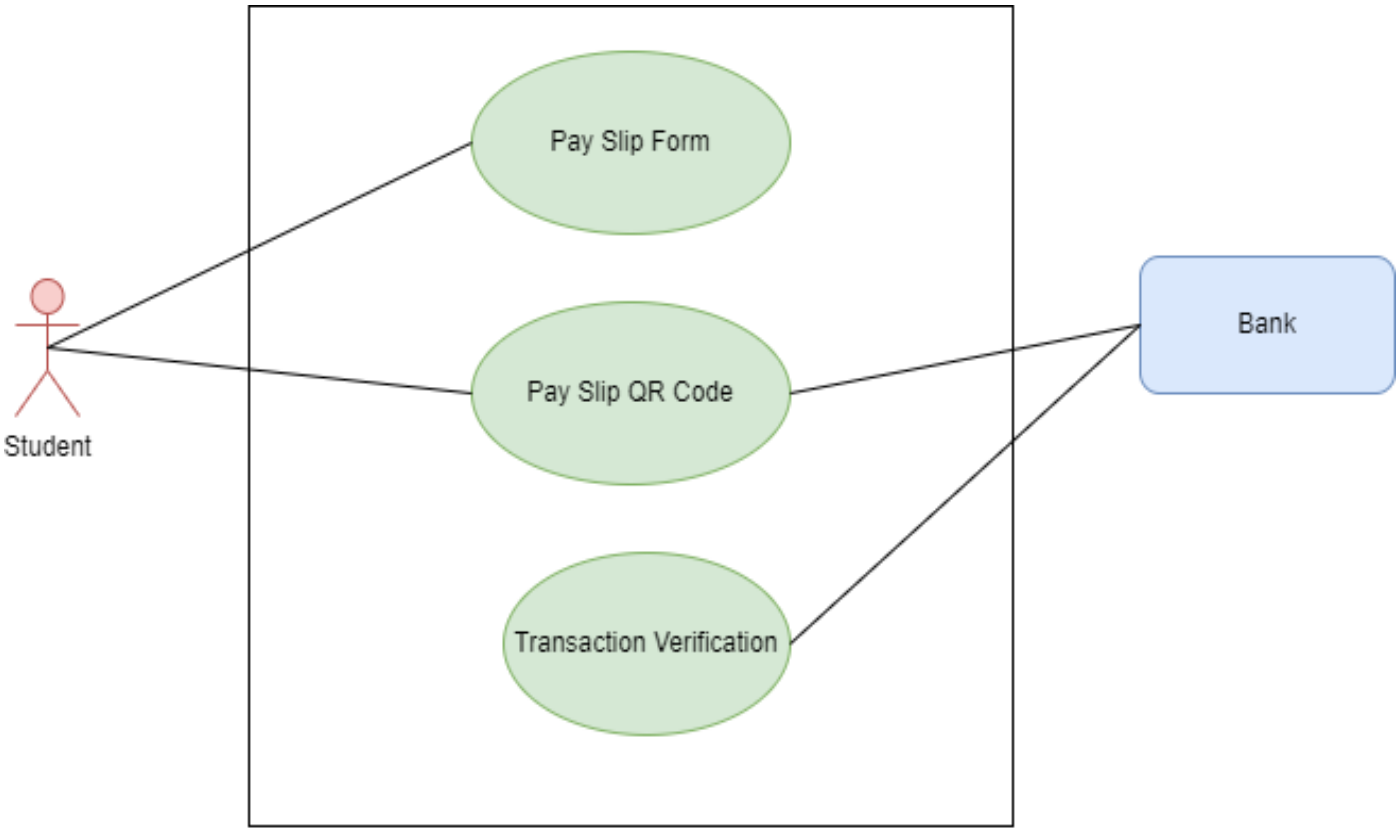
Students can also manually complete the payment through the bank.

5.6.1.4. Transaction Confirmation

Official, Hall Staff and Department Staff Confirm student transactions and update this information on the respective database.

5.7. Level 1.4.3

Name: Manual Transaction system
Primary Actor: Students
Secondary Actor: Bank



Use Case Level 1.4.3

5.7.1. Description of use case diagram level-1.4.3

5.7.1.1. Pay Slip Form

Students have to download and print out a copy of the payslip (Generated by the system) to complete their transaction manually. The payslip contains a QR code containing all the information needed to identify the student's account with a normal select query.

5.7.1.2. Pay Slip QR Code:

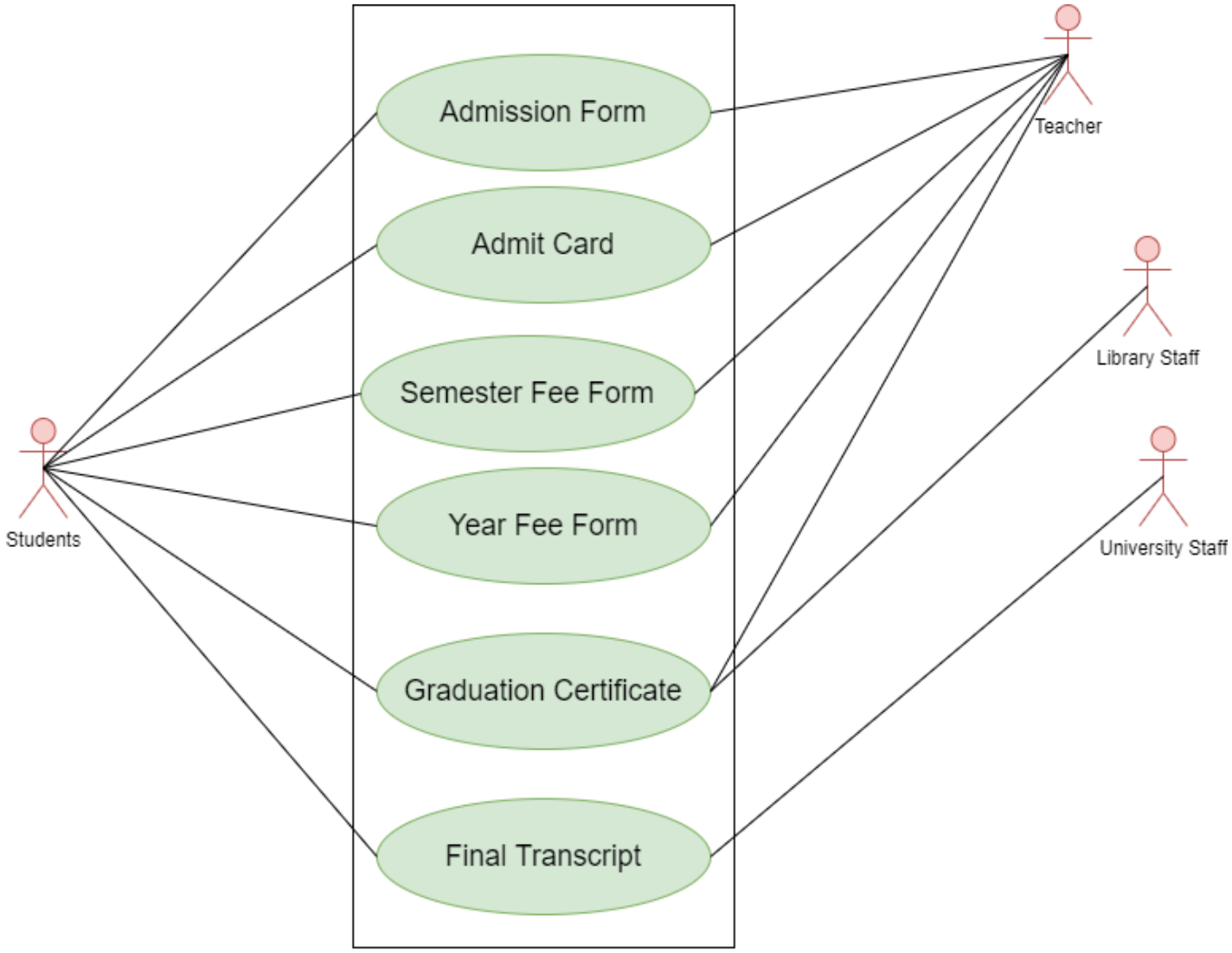
To complete the manual transaction the bank needs to confirm the payment. For this purpose, the system has a simple interface for the bank to use. Bank will take the payslip from the student and scan the QR code using the system to confirm the student's payment. The system will locate students' account information using the information from the QR code and confirm the payment for that account.

5.7.1.3. Transaction Verification

Bankers get QR code from the downloaded Payslip and then input this QR code into the Banking Transaction Interface for confirmation of the money transaction.

5.8. Level 1.5

Name: Form Verification system
Primary Actor: Students
Secondary Actor: Teacher, Staff



Use Case Level 1.5

5.8.1. Description of use case diagram level-1.5

5.8.1.1. Admission Form

Admission Form is the first and only one time filled up by students. By filling this form, all basic and necessary information is provided by students and it helps to complete future's all form fill up. After completing form fill up this is verified by Provost and Chairman/Director.

5.8.1.2. Admit Card

Before every Year Final/ Semester Final students fill up a form for admit card, payment exam fee, and then verified by the Provost & Chairman/Director.

5.8.1.3. Semester Fee Form

Students' payment fees for each semester and verified by Chairman/Director.

5.8.1.4. Year Fee Form

On every educational year, students pay to University Administration. So that they fill up a form, pay money verified by Provost & Chairman/Director.

5.8.1.5. Graduation Certificate

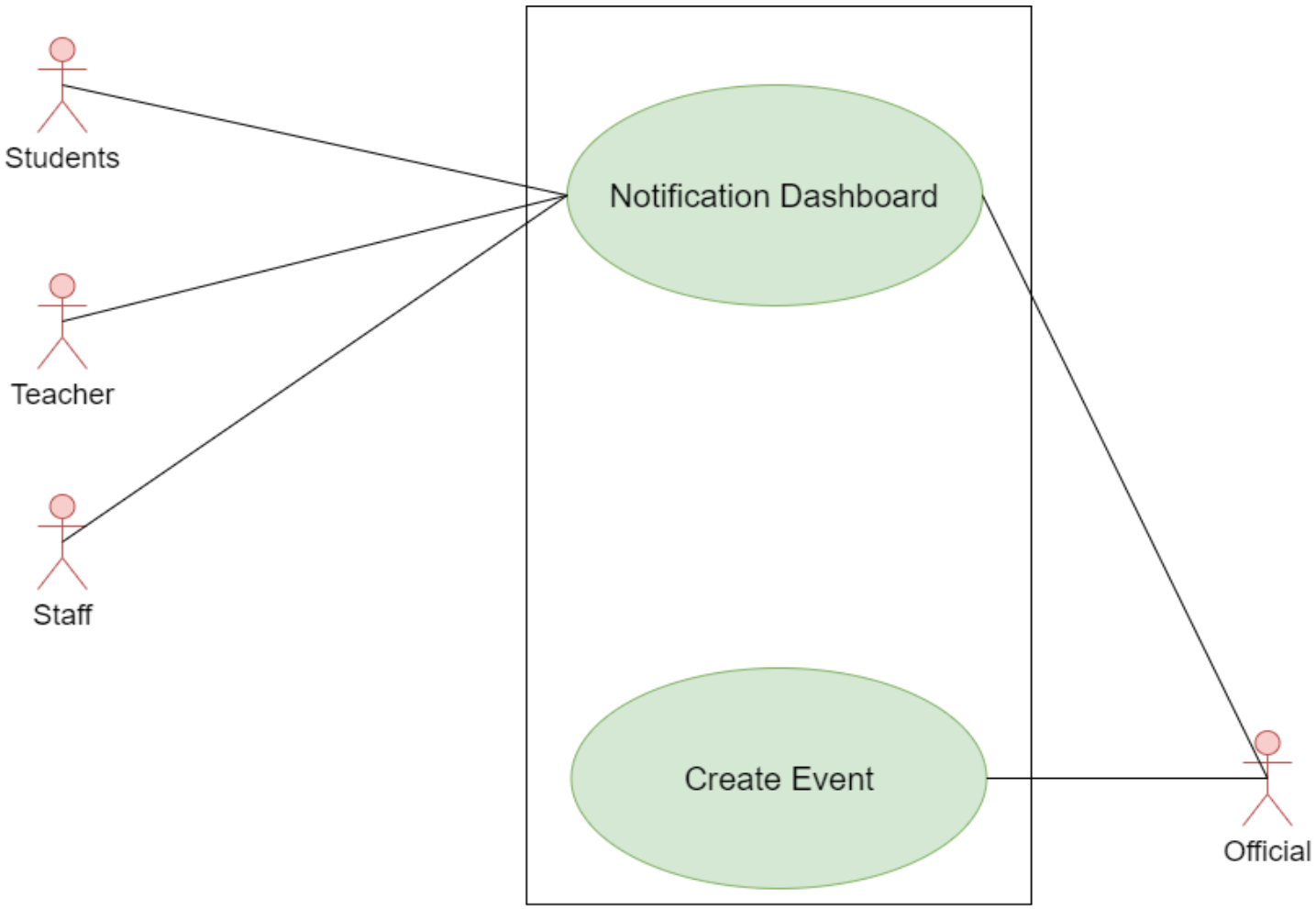
To get Hons/Masters Certificate, students must fill up their Graduation Certificate form, verified by Provost & Chairman/Director, then also verified by Central Library's Librarian and Hall Librarian after clearing book issues.

5.8.1.6. Final Transcript

To get Hons/Masters Certificate, students must fill up their Graduation Certificate form, payment fee and have it verified by Official.

5.9. Level 1.6

Name: Notification system
Primary Actor: Students, Teacher, Staff
Secondary Actor: Department staff, hall staff, Official



Use Case Level 1.6

5.9.1. Description of use case diagram level-1.6

5.9.1.1. Notification Dashboard

Students, Hall Provost, Department's Chairman, and Institute's Director, Central Library and Hall Library's Library Staff(only for Graduate Certificate), and Official(only for Final Transcript) get notification from University Administration/Staff(For Admission Form, Graduate Certificate and Final Transcript), Hall Administration(For Admit Card), or Department Administration(For Year Fee & Semester Fee). There is a list of previous forms which he/she fill Up and Check List if his payment and verification by Teacher and staff is ok or not.

5.9.1.2. Notification for Form Verification

There are several create events to notify students to fill up different forms and notify teachers to verify students' forms. Official creates events for Admission Form, Graduate Certificate and Final Transcript, Hall Staff create events for Admit Card and, Department Staff creates events for Year Fee & Semester Fee.

6. Activity Diagram

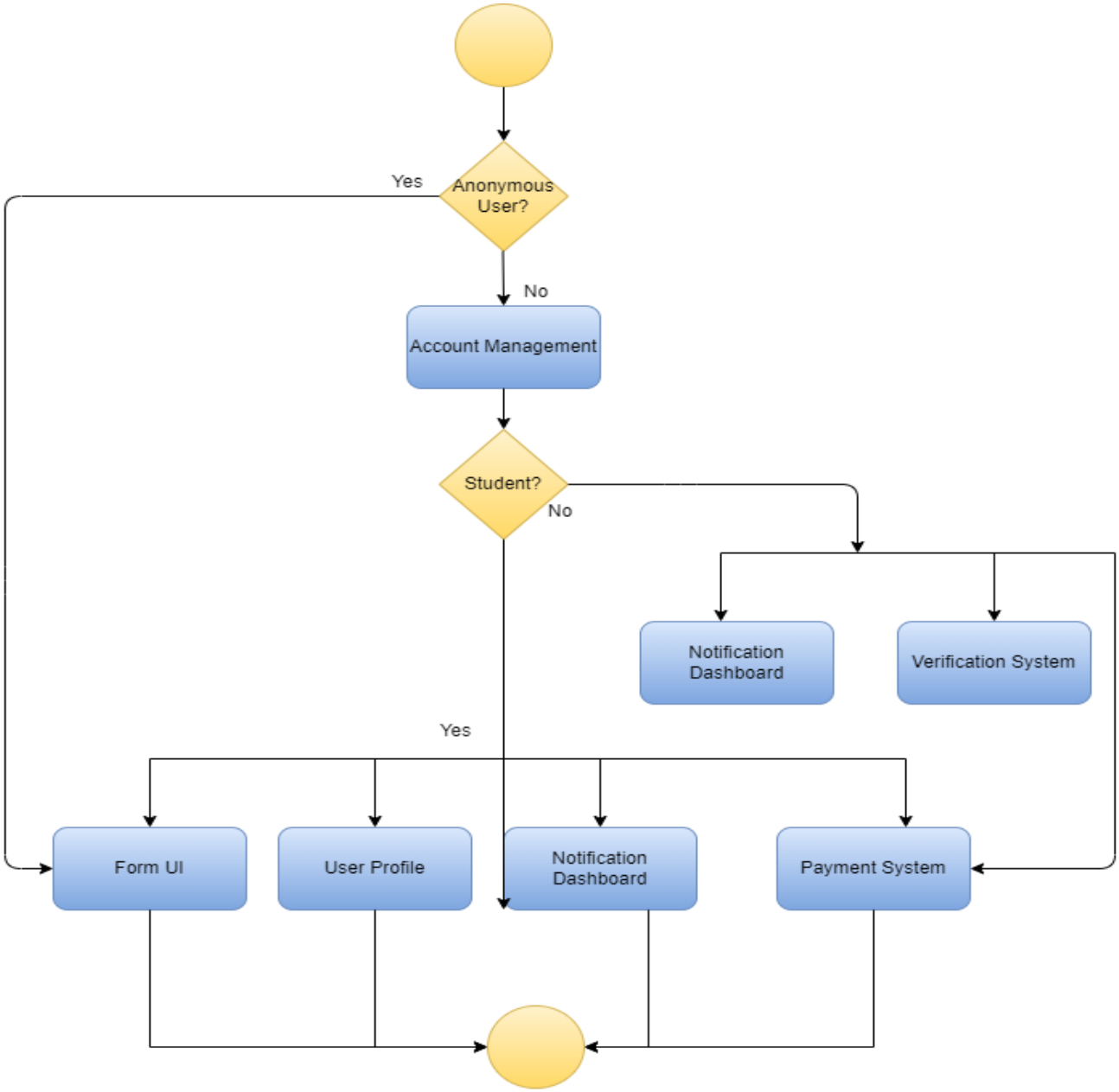
Definition

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency.

6.1. Level 1

Name: DU_FMS (DU Form Management System)

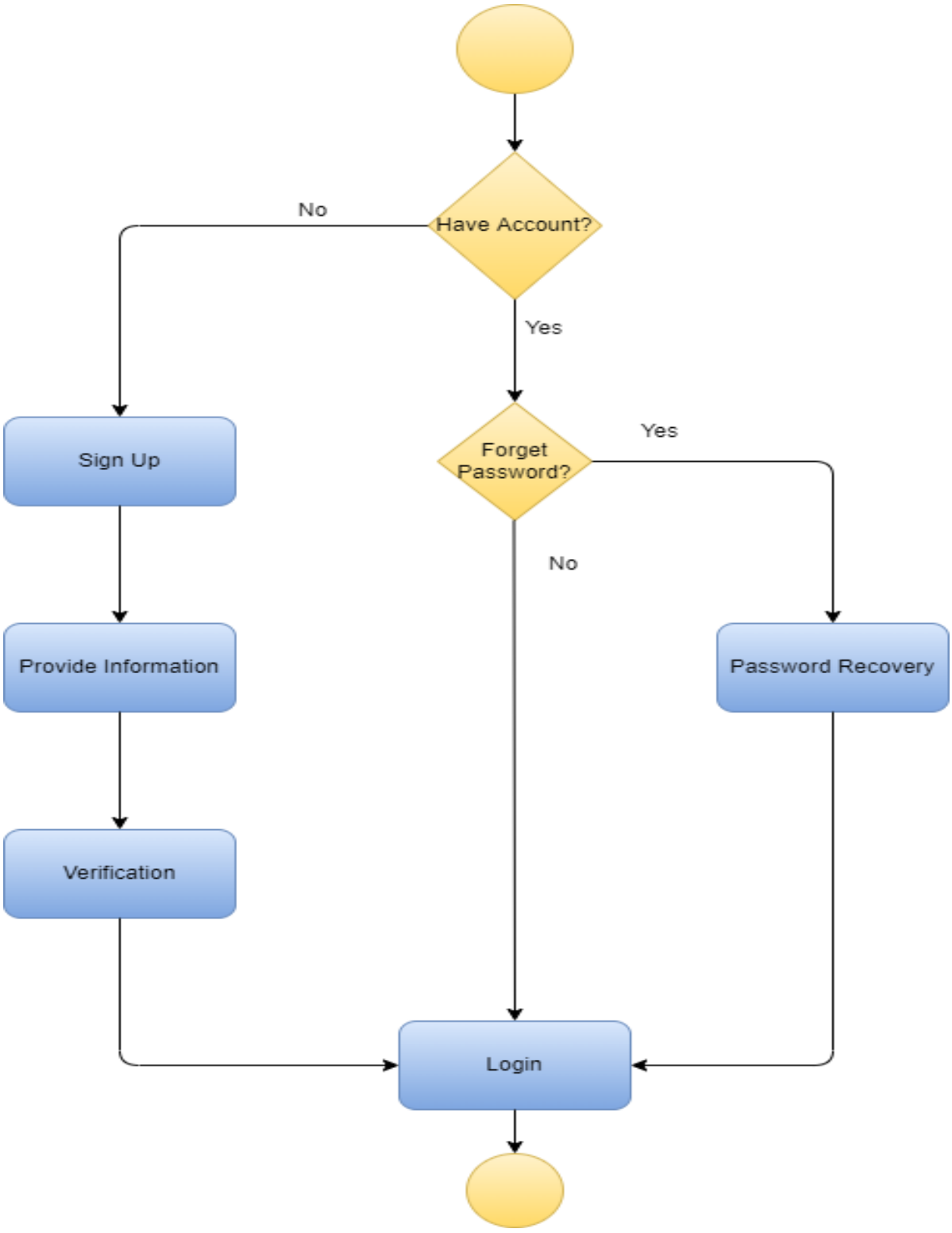
Reference: Use Case Level 1



6.2. Level 1.1

Name: User Management

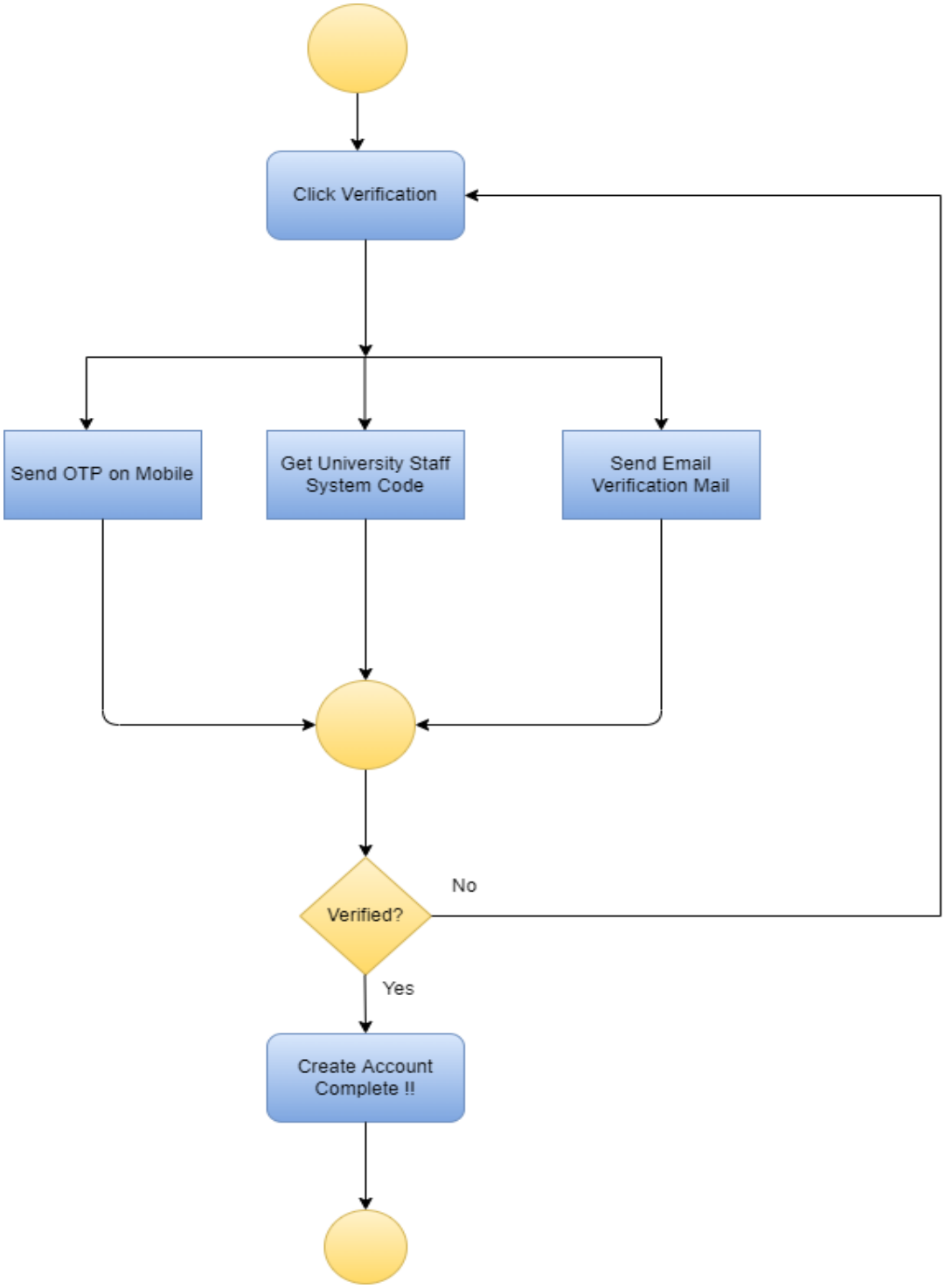
Reference: Use Case Level 1.1



6.3. Level 1.1.2

Name: Account Verification

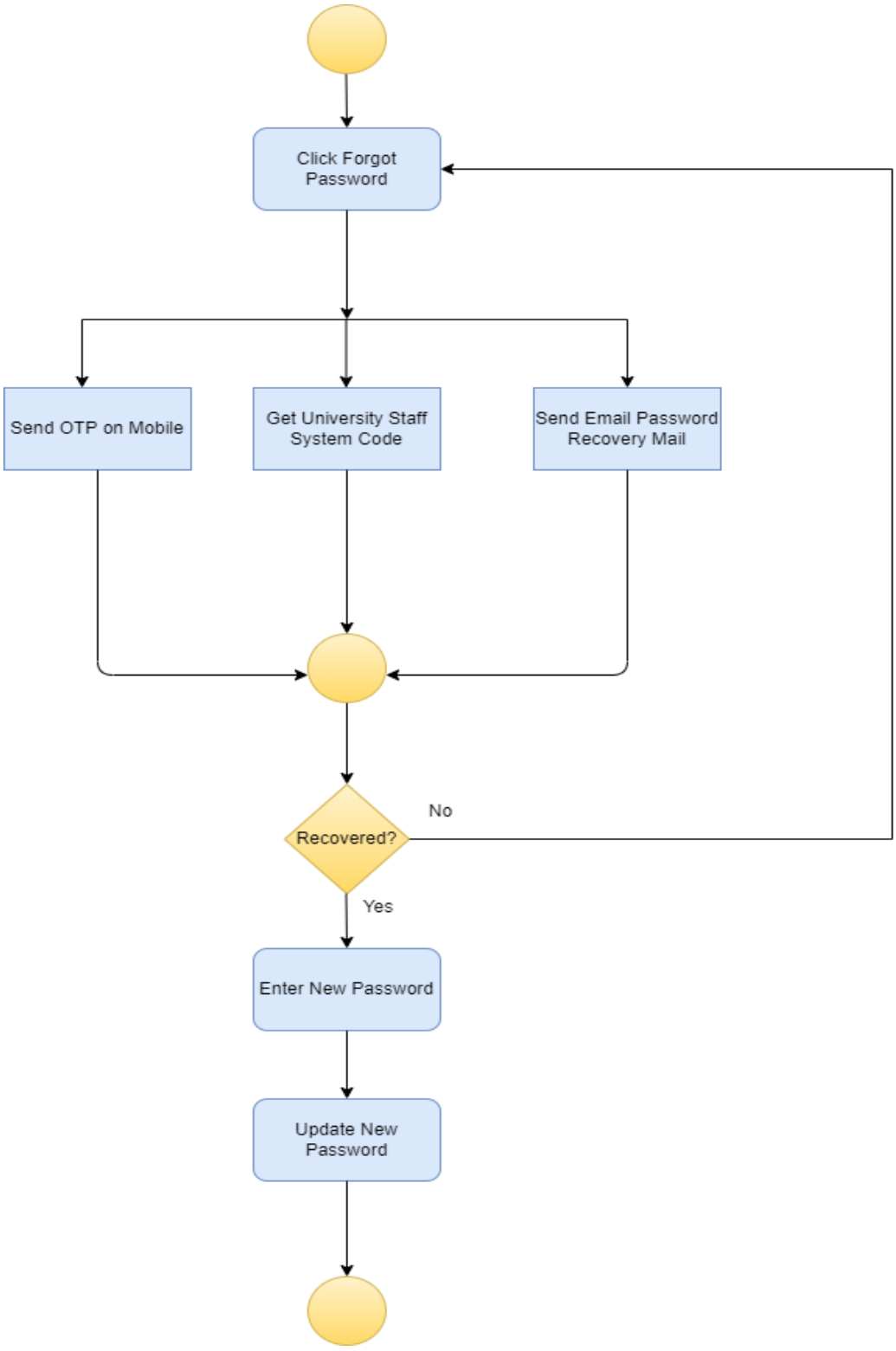
Reference: Use Case Level 1.1.2



6.4. Level 1.1.3

Name: Password Verification

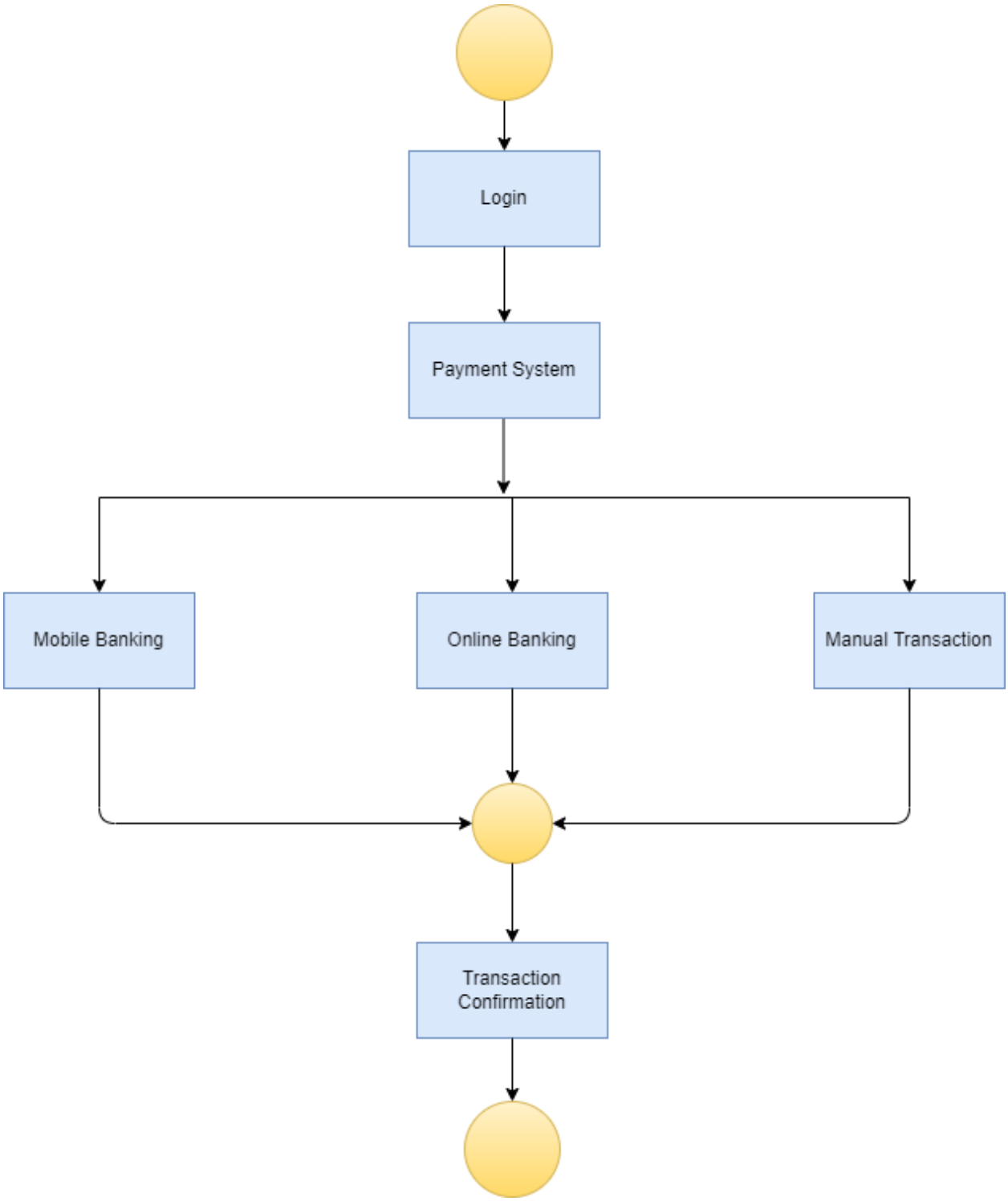
Reference: Use Case Level 1.1.3



6.5. Level 1.4

Name: Payment System

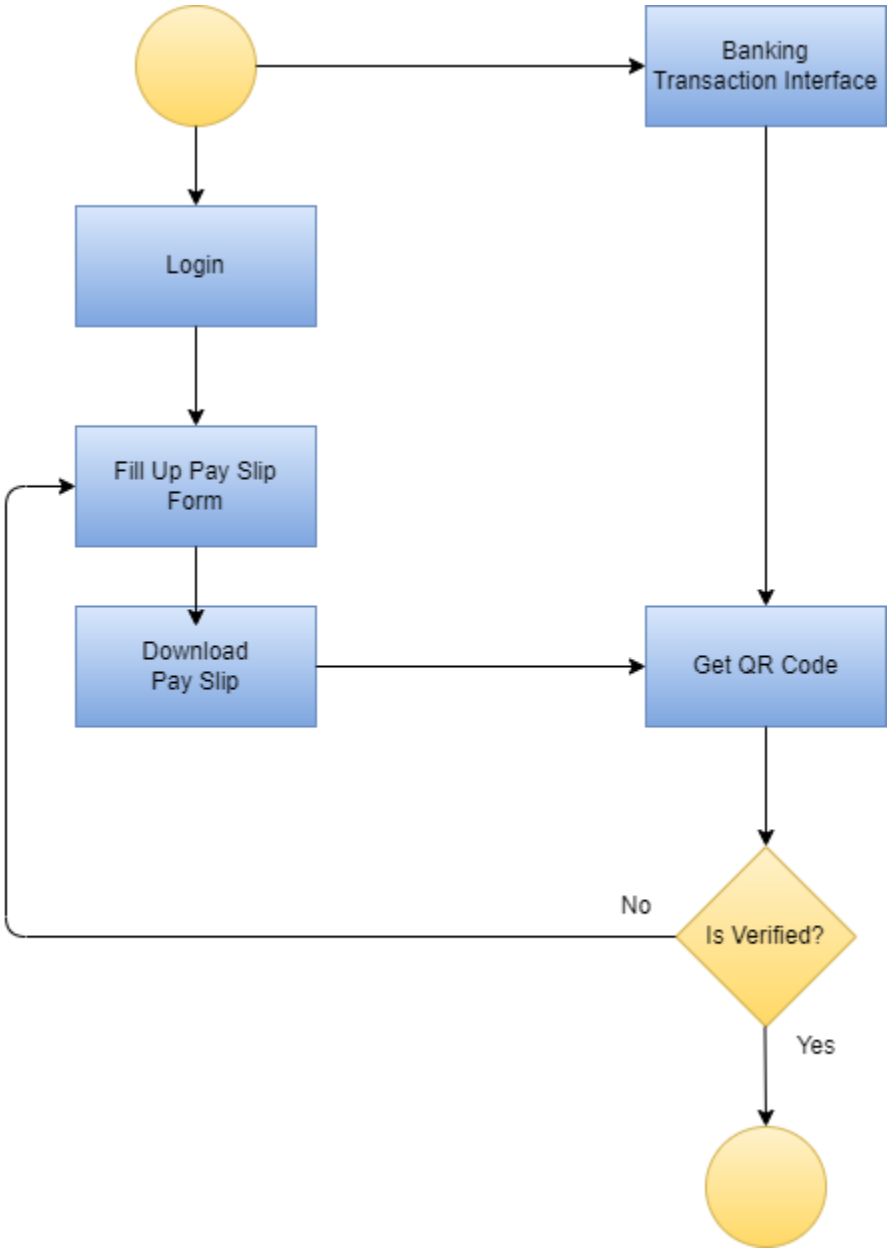
Reference: Use Case Level 1.4



6.6. Level 1.4.3

Name: Manual Transaction System

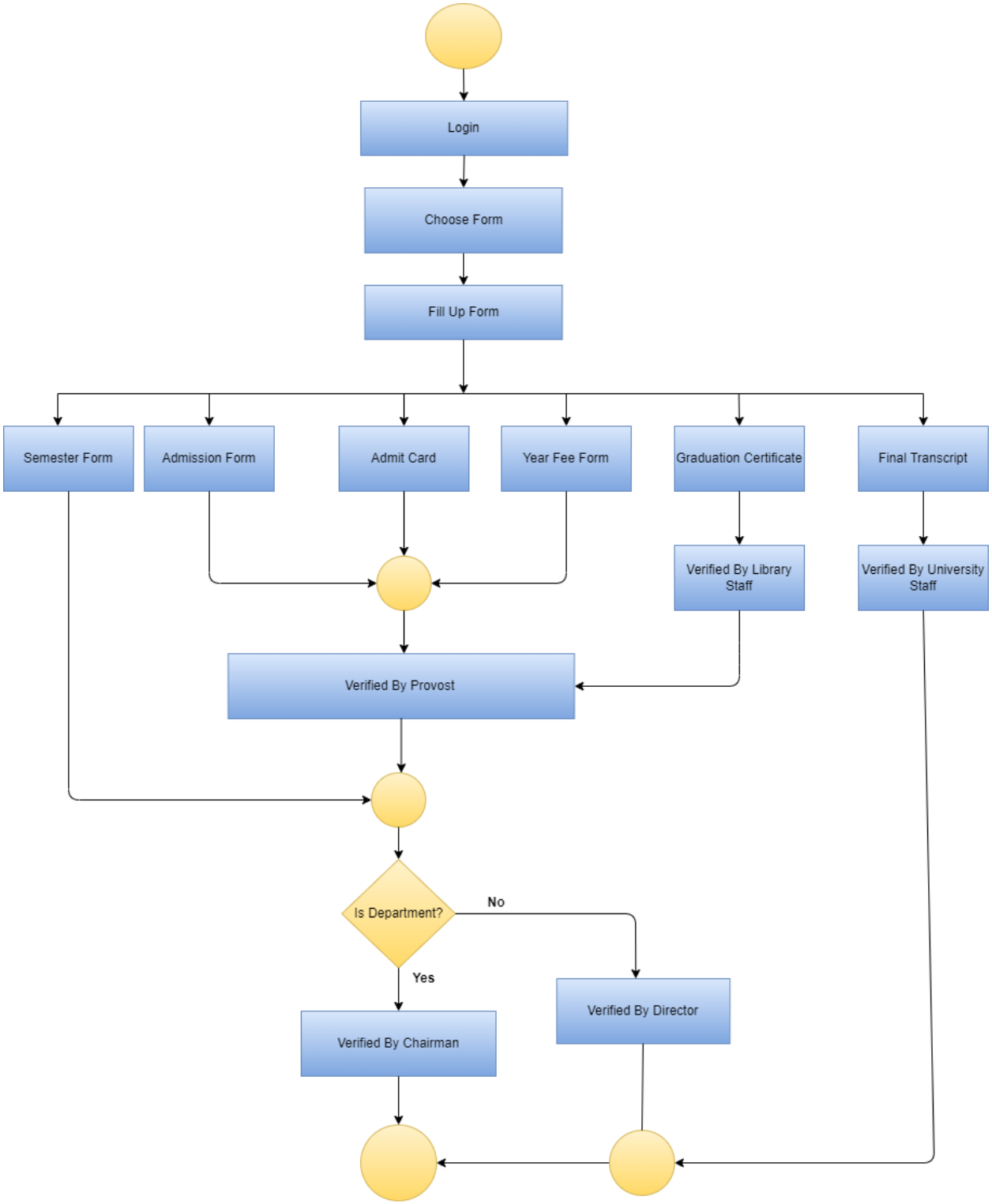
Reference: Use Case Level 1.4.3



6.7. Level 1.5

Name: Form Verification System

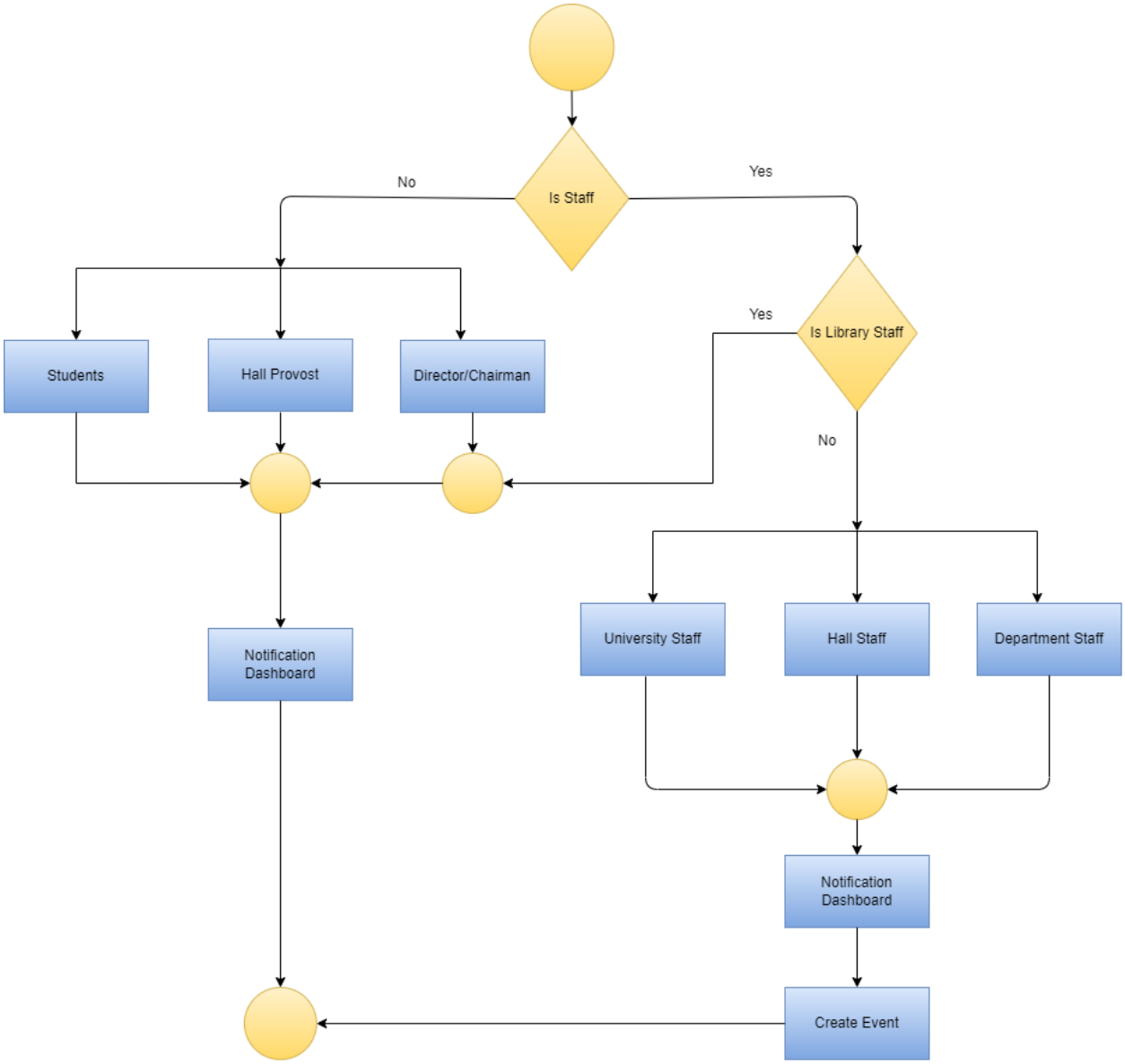
Reference: Use Case Level 1.5



6.8. Level 1.6

Name: Notification system

Reference: Use Case Level 1.6



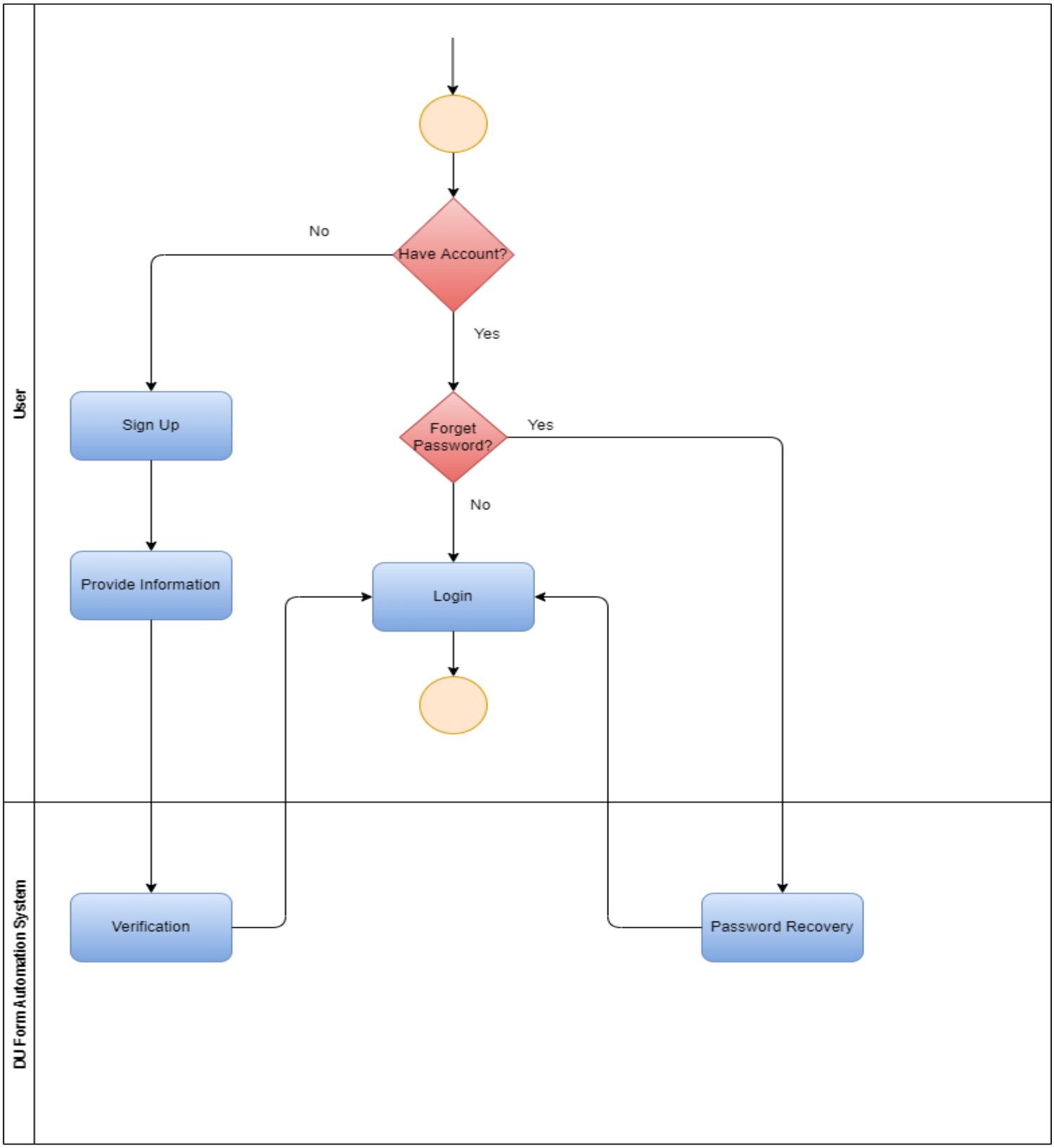
7. Swimlane Diagram

7.1. Definition of Swimlane Diagram

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects, and the information about how the data objects are entered, stored, transformed, and produced within the system.

7.2. Swimlane ID (SID) 1.1

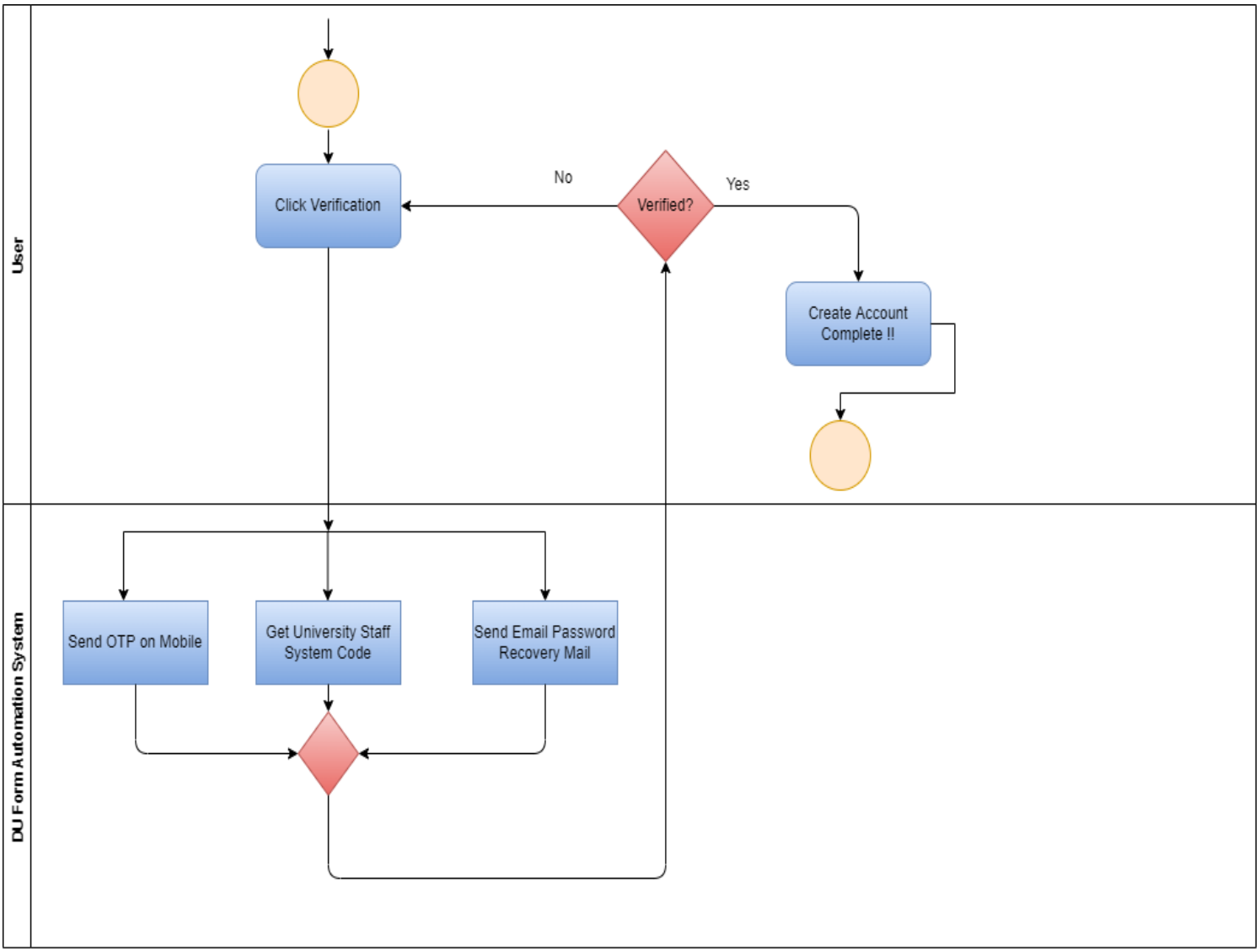
Name: User Management
Reference: Use Case & Activity Level 1.1



Swimlane Diagram Level 1.1

7.3 Swimlane ID (SID) 1.1.2

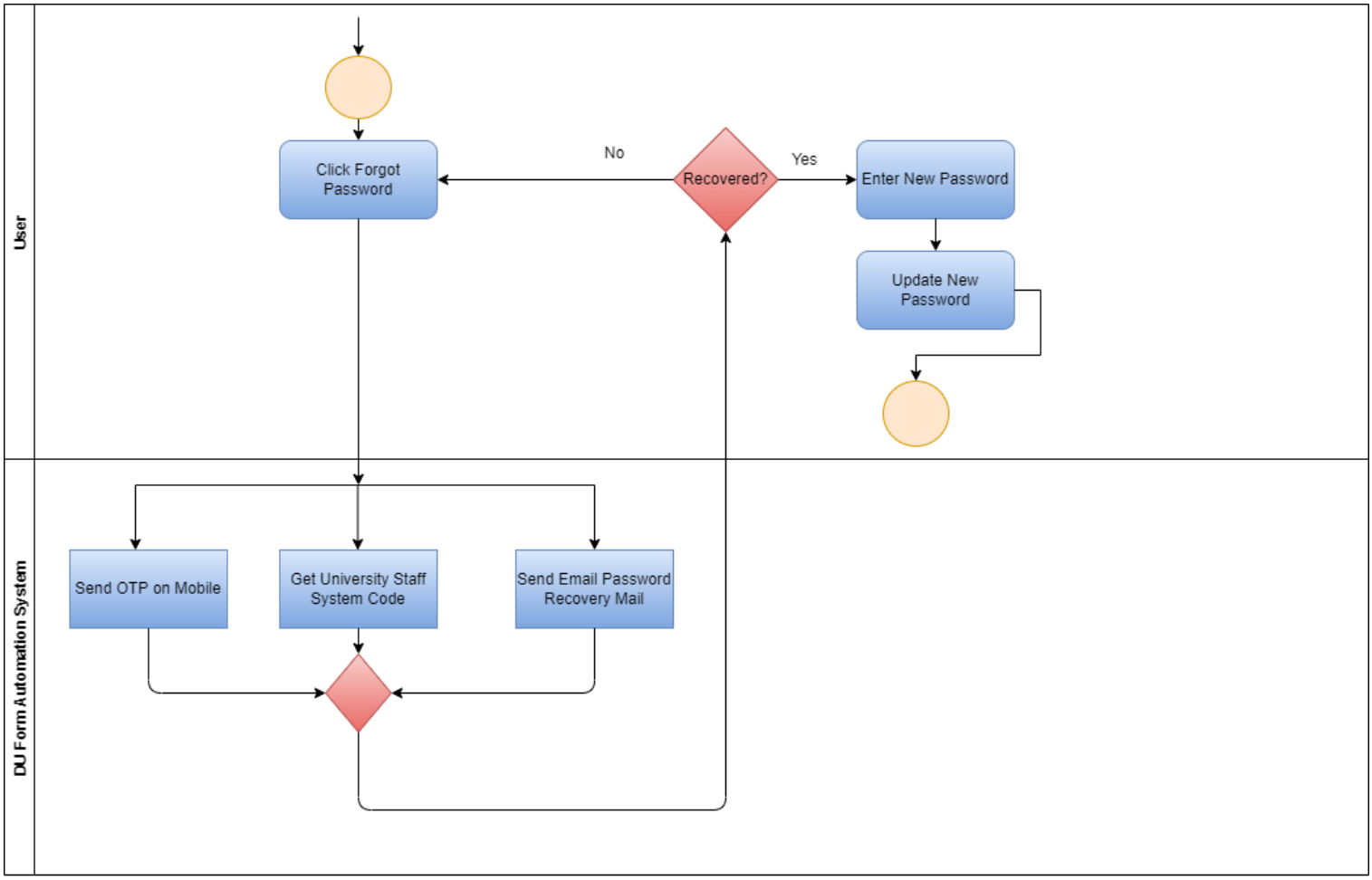
Name: Account Verification
Reference: Use Case & Activity Level 1.1.2



Swimlane Diagram Level 1.1.2

7.4. Swimlane ID (SID) 1.1.3

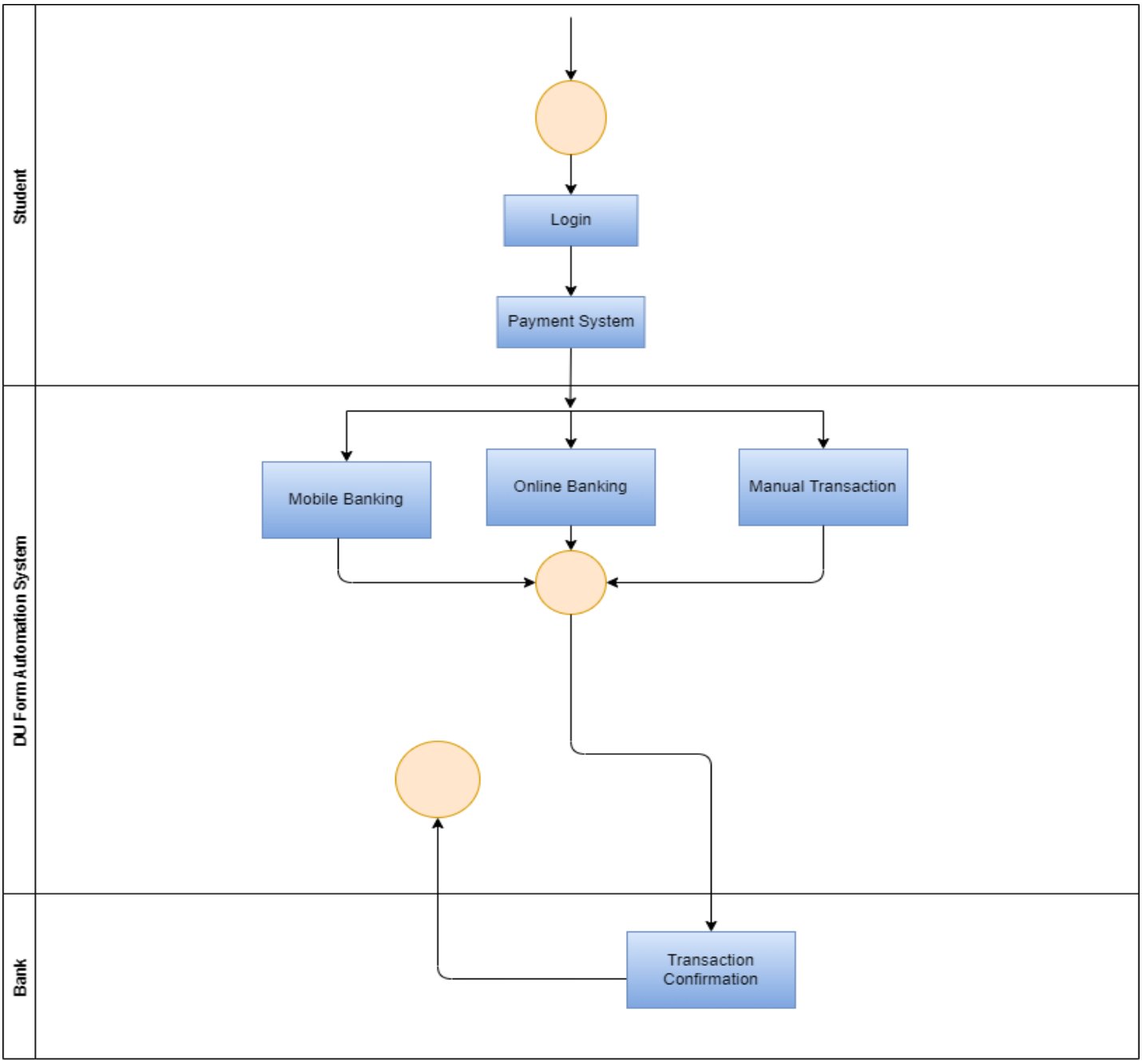
Name: Password Verification
Reference: Use Case & Activity Level 1.1.3



Swimlane Diagram Level 1.1.3

7.5. Swimlane ID (SID) 1.4

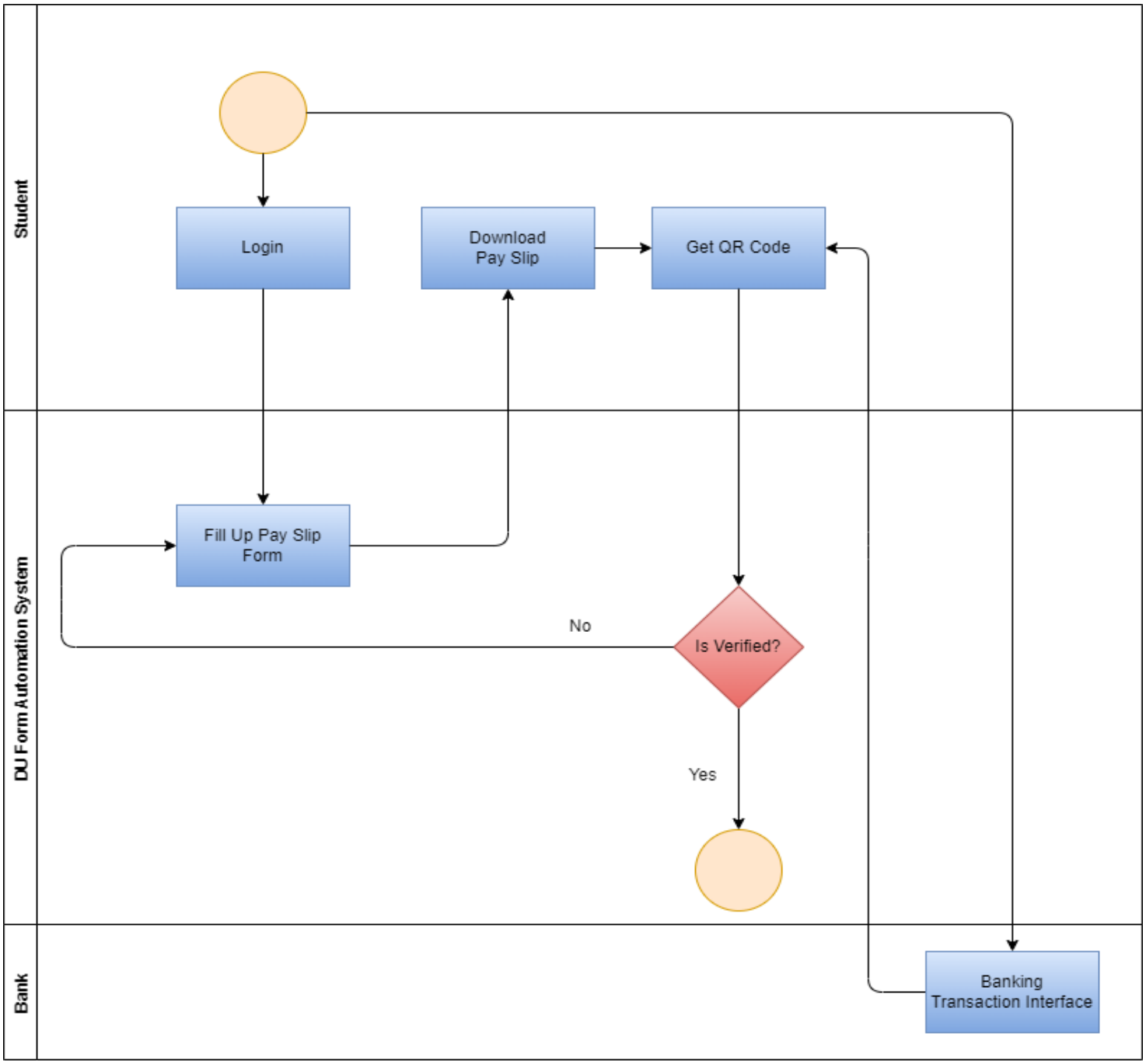
Name: Payment System
Reference: Use Case & Activity Level 1.4



Swimlane Diagram Level 1.4

7.6. Swimlane ID (SID) 1.4.3

Name: Manual Transaction System
Reference: Use Case & Activity Level 1.4.3

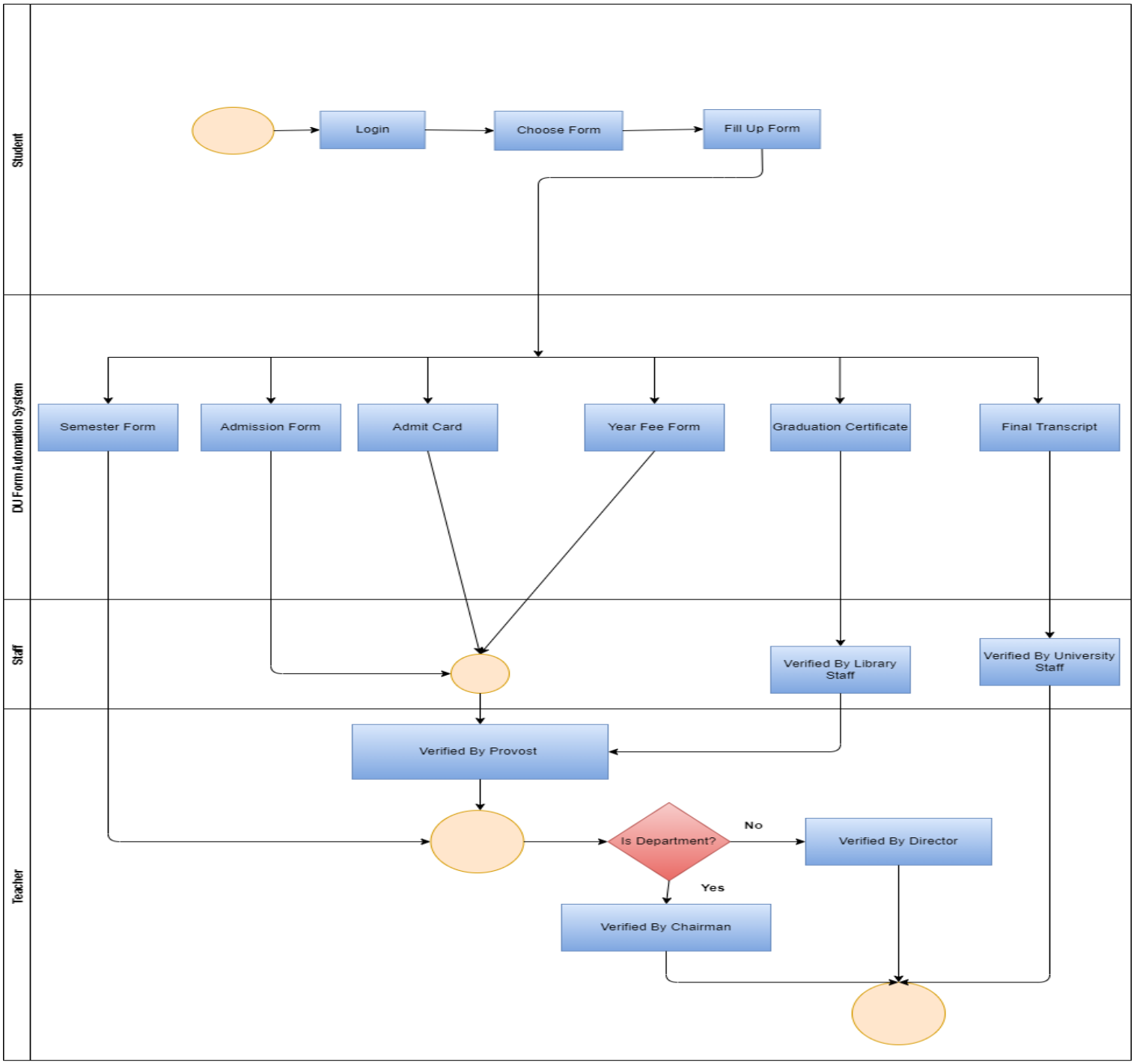


Swimlane Diagram Level 1.4.3

7.7. Swimlane ID (SID) 1.5:

Name: Form Verification System

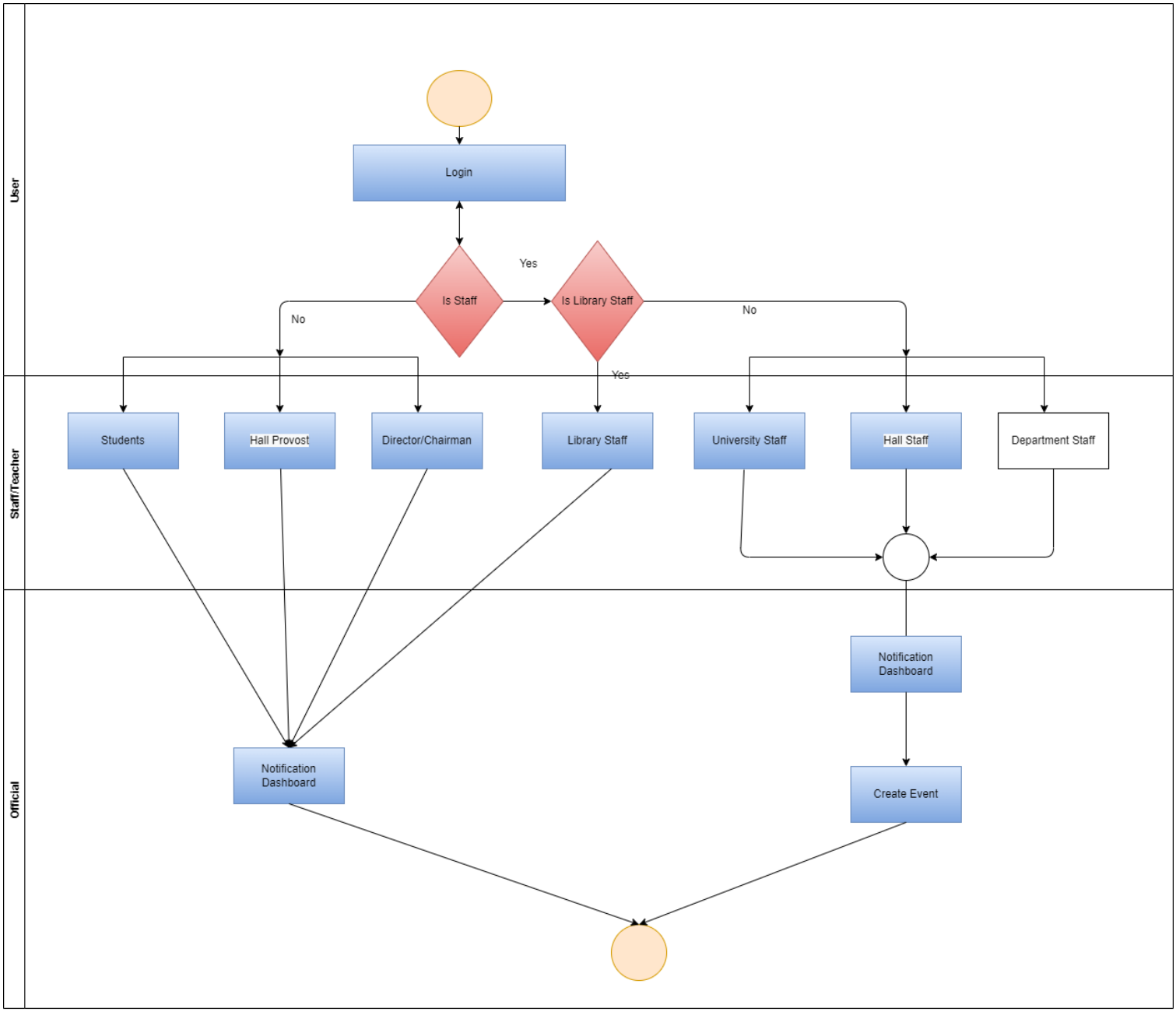
Reference: Use Case & Activity Level 1.5



Swimlane Diagram Level 1.5

7.8. Swimlane ID (SID) 1.6

Name: Notification system
Reference: Use Case & Activity Level 1.6



Swimlane Diagram Level 1.6

8. Data Based Modeling

8.1. Definition of Data Modeling Concept

Data modeling is a process used to define and analyze data requirements needed to support the business processes within the scope of corresponding information systems in organizations. Therefore, the process of data modeling involves professional data modelers working closely with business stakeholders, as well as potential users of the information system.

8.2. Data Objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different Page 60 of 126 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.

8.3. Data Objects Identification

SI No.	Noun	Solution (s) / Problem (p) Space	Attribute
1	DU_FMS	p	
2	Account	p	
3	Profile	p	
4	Student	s	5,6,7,8,9,10,11,12,13,14, 68,69
5	First Name	s	
6	Last Name	s	
7	User Name	s	
8	Password	s	

9	Mobile Number	s	
10	Email address	s	
11	Registration No	s	
12	Department Name	s	
13	Hall Name	s	
14	Session	s	
15	Address	s	
16	Teacher	s	5,6,7,8,9,10,12,13,34,38
17	Provost	p	
18	Director	p	
19	Chairman	p	
20	Institute	p	
21	Faculty	p	
22	Dean's office	p	
23	Library	p	
24	User Management	p	
25	Form	s	47,48,54,55,56
26	Admit Card	p	
27	Certificate	p	
28	Transcript	p	
29	Library Staff	p	
30	Department Staff	p	
31	Librarian	p	
32	Staff	s	5,6,7,8,9,10,35,37,39
33	Official	s	5,6,7,8,9,10,59,62
34	Teacher ID	s	
35	Staff ID	s	

36	Notification	s	57,58,59,60,62,63,64
37	Staff Type	s	
38	Teacher Designation	s	
39	Office Name	s	
40	Transaction ID	s	
41	Transaction Method	s	
42	Amount	s	
43	Bank	s	44,71
44	Bank ID	s	
45	Account No.	p	
46	Bank Name	s	
47	Form ID	s	
48	Form Type	s	
49	Dashboard	p	
50	Hall Fee Form	p	
51	Semester Fee Form	p	
52	Admission Fee Form	p	
53	Year Fee Form	p	
54	Student Information	s	
55	Verification Information	s	
56	Payment Information	s	
57	Time	s	
58	Date	s	
59	Notification ID	s	
60	Notification title	s	
61	Money Transfer Company	p	

62	Official ID	s	
63	Notification Type	s	
64	Notification Description	s	
65	Event Deadline	s	
66	Transaction	s	40,42,44,57,58,67,
67	Student ID	s	
68	SSC Information	s	
69	HSC Information	s	
70	University	p	
71	Bank name	s	

8.3.1. Final Data Object

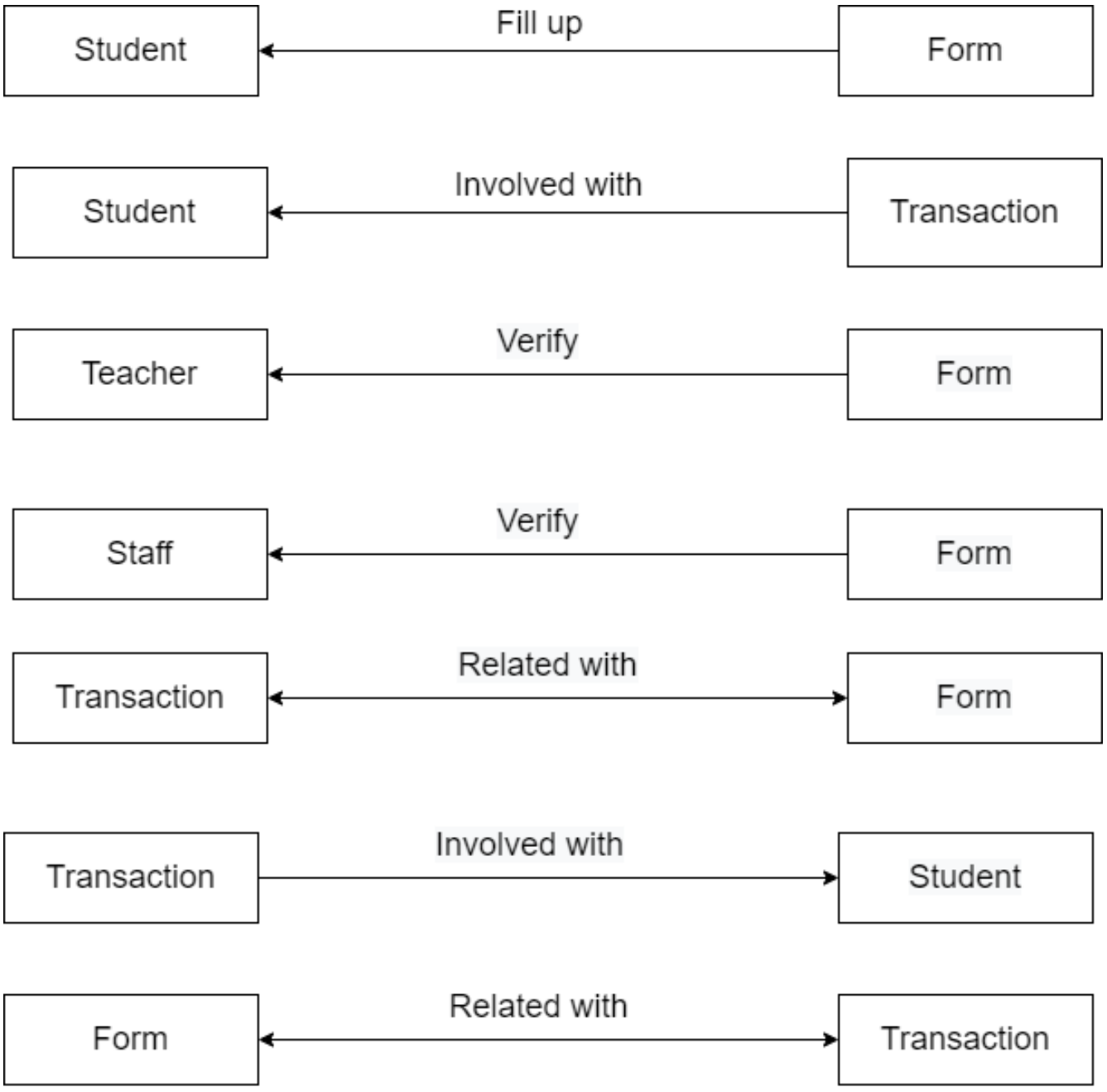
1. Student
2. Teacher
3. Staff
4. Bank
5. Transaction
6. Form
7. Notification
8. Official

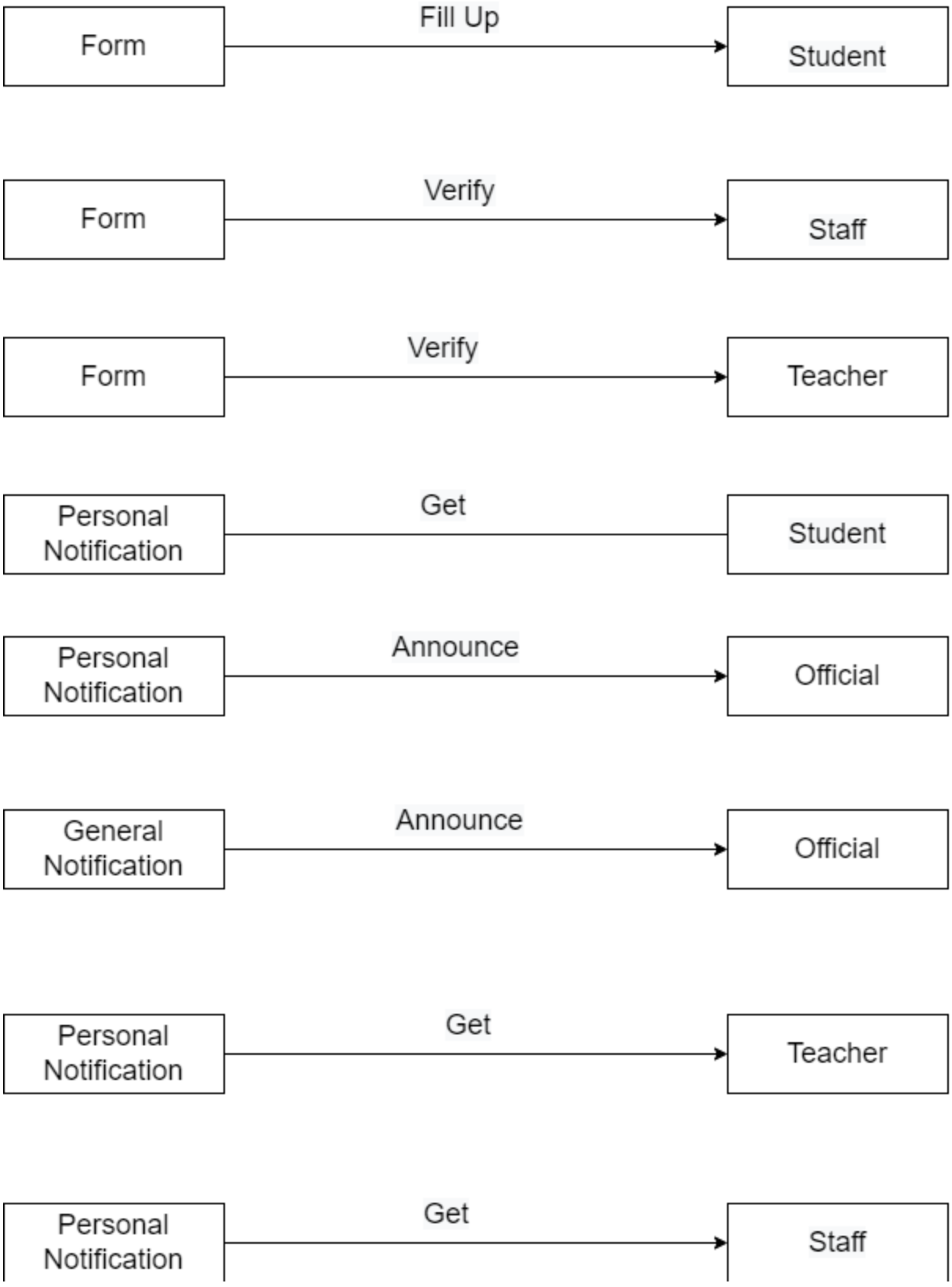
8.3.2 Analysis

After initially selecting the data objects and reviewing the selected ones, we decided to change the Notification object and split it into two distinct entities which are General and Personal Notification.

8.4. Data Object Relationship

8.4.1. Relationship between data objects

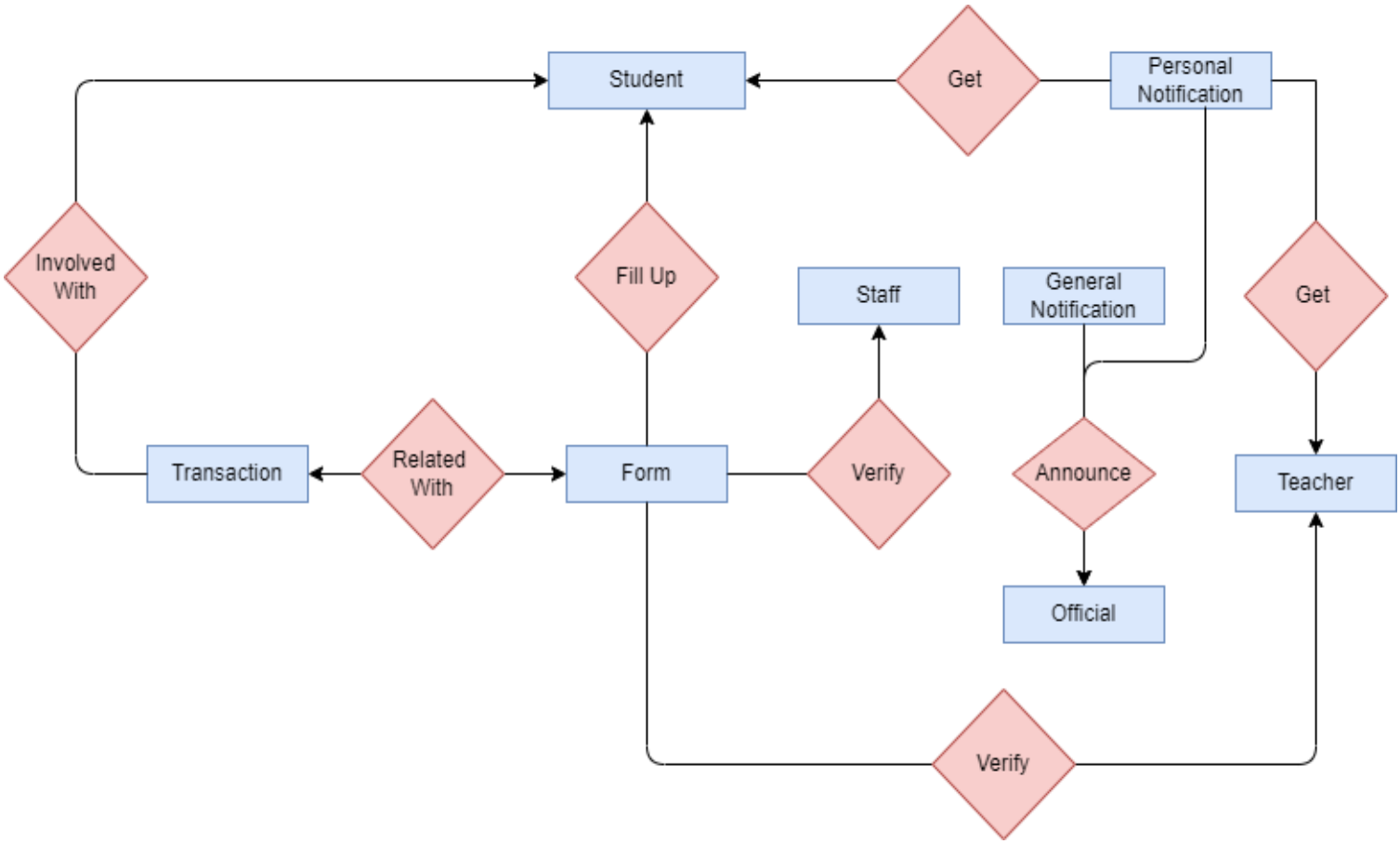



Form
Fill Up
Student
Form
Verify
Staff
Form
Verify
Teacher
Personal Notification
Get
Student
Personal Notification
Announce
Official
General Notification
Announce
Official
Personal Notification
Get
Teacher
Personal Notification
Get
Staff

8.5. ER Diagram

8.5.1. Definition of ER Diagram

An entity-relationship diagram (ERD) shows the relationships of entity sets stored in a database. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database.



8.6. Schema Diagram

Data Object	Attribute	Type	Size
Teacher	-First name -Last name -Mobile number -Email address -**Teacher ID** -Teacher designation -Hall Name -Department name -User name -Password	Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar	40 40 40 40 40 40 40 40
Student	-First name -Last name -Mobile number -Address -Email address **-Student ID** -Roll No -Registration No -Department name -Hall Name -Session -User name -Password -SSC Information -HSC Information	Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar	40 40 40 40 40 40 40 40 40 40 40 40 40 40

Staff	-First name -Last name -Mobile number -Email address **-Staff ID** -Staff Type -Office Name -Username -Password	Varchar Varchar Varchar Varchar Varchar Varchar Varchar Varchar	40 40 40 40 40 40 40 40
Official	-First name -Last name -Mobile number -Email address **-Official ID** -Username -Password -Notification ID	Varchar Varchar Varchar Varchar Varchar Varchar Varchar	40 40 40 40 40 40 40
Transaction	**-Transaction ID** -Student ID -Bank -Transaction method -Amount -Time -Date	Varchar Varchar Varchar Varchar Varchar Time Date	40 40 40 40 40 - -
General Notification	**-Notification ID** -Notification Title -Notification Description -Time -Date -Official ID -Notification Type -Event Deadline	Varchar Varchar Varchar Time Date Varchar Varchar Varchar	40 150 - - 40 40 40

Personal Notification	**-Notification ID** -Student ID -Teacher ID -Notification Title -Notification Description -Time -Date -Official ID -Notification Type -Event Deadline	Varchar Varchar Varchar Time Date Varchar Varchar Varchar Varchar Varchar	40 150 - - 40 40 40 40 40
Form	**-Form ID** -Form type -Student Information -Payment Information -Verification Information	Varchar Varchar Varchar Varchar Boolean	40 40 40 40 2

9. Class-Based Modeling

9.1. Class Based Modeling Concept

Class-based modeling 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.

9.2 Noun List from DU From Automation System

Sl No.	Noun
1	Student
2	Teacher
3	Director (institute)
4	Chairman (department)
5	Dean (faculty)
6	Provost (hall)
7	House tutor (hall)
8	Officials
9	University Staff
10	Department Staff
11	Hall Staff
12	Library staff
13	Department
14	Institute
15	Faculty

16	Dean's office
17	Bank
18	User Management
19	Process sign up
20	Form
21	University fee form
22	Semester fee form
23	Hall fee form
24	Hall fee form (admission)
25	Admit card form
26	SSC information
27	HSC information
28	Application for exam
29	Student information form for exam
30	Form selection
31	Transaction
32	Notification
33	Description (notification)
34	Student ID
35	Teacher ID
36	Staff ID
37	Bank ID
38	Notification ID

39	Address
40	Date & Time
41	Full name
42	Mobile no.
43	Email Address
44	User name
45	Password
46	Roll no.
47	Department name
48	Hall name
49	Session
50	Office name
51	Payment amount
52	Notification range
53	Type (User)
54	Type (Library staff)
55	Type (Teacher designation)
56	Type (Exam)
57	Students signature
58	Directors signature
59	Chairman's signature
60	Controller of examination’s signature
61	Nationality

62	Religion
63	Religion class
64	University registration no.
65	Type (transaction)
66	Online bank
67	Money transfer company
68	Transaction ID
69	Account name/ID/Number
70	Type (admission)
71	Signature from bank (payment received)
72	Hall provosts signature
73	User
74	OTP
75	Dashboard
76	Verification
77	Password recovery system
78	DU form automation system
79	Staff
80	Menu

9.3. Verb List

Sl No.	Verb
1	Create Account
2	Verify Account
3	Send Code
4	Input Code
5	Update Info
6	Change
7	Recover(password)
8	Send (Recovery Link)
9	Send (Recovery Request to Admin)
10	Click (Recovery Link)
11	Input (Password)
12	Input (Admin OTP)
13	Update (Database)
14	Send (OTP)
15	Input (New OTP)
16	Login
17	Select Form
18	Notification email
19	Upload notification
20	Create Event
21	Announce Event

22	Send money
23	Verify
24	Upload Form
25	Send Confirmation Email
26	Send Confirmation SMS
27	Show notification
28	Match QR code
29	Send QR code

9.4. General Classification

Candidate classes were then characterized into seven general classifications. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Potential nouns to become a class after general classification criteria given below

Sl no.	Noun	General classification
1	DU form automation system	2,3,7
2	Student	4,5,7
3	Teacher	4,5,7
4	Director (institute)	4,5,7
5	Chairman (department)	4,5,7
6	Dean (faculty)	4,5,7
7	House tutor (hall)	4,5,7
8	Officials	4,5,7
9	University Staff	4,5,7
10	Department Staff	4,5,7
11	Hall Staff	4,5,7

12	Library staff	4,5,7
13	Department	5,6,7
14	Institute	5,6,7
15	Faculty	5,6,7
16	Dean's office	5,6,7
17	Bank	1
18	User Management	3
19	Process sign up	3
20	Form	2,3,7
21	University fee form	2,3,7
22	Semester fee form	2,3,7
23	Hall fee form	2,3,7
24	Hall fee form (admission)	2,3,7
25	Admit card form	2,3,7
26	SSC information	7
27	HSC information	7
28	Application for exam	2,7
29	Student information form for exam	2,7
30	Form selection	3
31	Transaction	3
32	Notification	3,7
33	Announcement	3,7
34	Verification	3,7

35	Dashboard	2,3,7
36	Online bank	1
37	Money transfer company	1
38	Password recovery system	1,4,7
39	University registration no.	7
40	OTP	1
41	User	4,5,7
42	Signature	7
43	Staff	4,5,7
44	Student database	1,2,7
45	Teacher database	1,2,7
46	Staff database	1,2,7

9.5. Selection Criteria

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Potential nouns to become a class after selection criteria:

Sl no.	Noun	Selection criteria
1	DU form automation system	1,2,3,6 (selected)
2	User	1,3,4,5
3	Student	1,3,4,5 (selected)
4	Teacher	1,3,4,5 (selected)
5	Director (institute)	1,3,4,5
6	Chairman (department)	1,3,4,5
7	Dean (faculty)	1,3,4,5
8	House tutor (hall)	1,3,4,5
9	Officials	1,3,4,5 (selected)
10	University Staff	1,3,4,5
11	Department Staff	1,3,4,5

12	Hall Staff	1,3,4,5
13	Library staff	1,3,4,5
14	Department	4,5
15	Institute	4,5
16	Faculty	4,5
17	Dean's office	4,5
18	Bank	1,2 (selected)
19	Form	1,3,4,5 (selected)
20	University fee form	1,3,4,5
21	Semester fee form	1,3,4,5
22	Hall fee form	1,3,4,5
23	Hall fee form (admission)	1,3,4,5
24	Admit card form	1,3,4,5
25	Dashboard	2,3 (selected)
26	Online bank	1,2
27	Money transfer company	1,2
28	Password recovery system	1,2
29	OTP	1,2
30	Staff	1,3,4,5 (selected)
31	Student database	1,6 (selected)
32	Teacher database	1,6 (selected)
33	Staff database	1,6 (selected)

9.6. Attribute and Method Identification :

Class Name	Attribute	Method
User	-UserID -Password -FullName -Phone Number -Email Address -Address -UserType	+getuserID() +setUserID() +getuserType() +setUserType() +getFull_name() +setFull_name() +getMobile_number() +setMobile_number() +getEmail_address() +setEmail_address() +getPassword() +setPassword() +create_account() +login() +send_email_verification_link() +send_OTP_by_SMS() +verify_info()
Student	-FullName -Mobile number -Address -Email address -StudentId -Roll No -Registration No -Department name -Hall Name -Session -User name -Password -SSC Information -HSC Information	+select_form() +form_fill_up() +choose_payment_type() +make_payment() +recieve_document() +update_info() +cancel_form() +recieve_confirmation() +get_notification() +show_notification() +send_QR()

Teacher	-First name -Last name -Mobile number -Email address -TeacherId -TeacherDesignation -TeacherType -Hall Name -Department name -User name -Password	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Staff	-First name -Last name -Mobile number -Email address -Staff Id -staffType -Office Name -Username -Password	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Official	-First name -Last name -Mobile number -Email address -Official Id -Username -Password -Notification ID	+select_students() +make_group() +verify_info() +generate_notification() +send_notification_to_teacher() +send_notification_to_student() +send_notification_to_staff() +annouce_activities() +verify_form() +verify_user_information()
Form	-FormID -FormType -StudentName -StudentRegistrationNo -Session -DepartmentName	+update_student_info() +display_form() +update_student_database() +update_verification_info() +match_QR()

Dashboard	-userID -userType -notificationID -FormList	+notify_user() +highlight_unread_notification() +show_form_menu() +show_form_verification() +show_payment_verification()
Bank	-BankID -BranchName -PaymentAmount -PaymentConfirmation	+receive_payment() +generate_payment_confirmation() +send_confirmation() +send_QR()
Transaction	-Transaction ID -Student ID -Bank ID -Transaction method -Amount -Time -Date	+allow_confirmation() +update_form() +deny_confirmation()
Notification	-Notification ID -Notification Title -Notification Description -Time -Date -Notification Type	+setID() +setTitle() +setDate() +setType() +notification_status() +notification_visibility() +show_notification()
SMS		+send_confirmation() +verify_phone_number()
Email		+send_confirmation() +verify_email_address()
Database	-StudentList -TeacherList -StaffList -OfficialList -FormList -TransactionList	-update() -delete() -insert() -search()

Director (Institute)	-fullName -mobileNumber -email -teacherID -instituteName -instituteID -designation	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Chairman (Department)	-fullName -mobileNumber -email -teacherID -departmentName -departmentID -designation	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Dean (faculty)	-fullName -mobileNumber -email -teacherID -facultyName -facultyID -designation	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
House tutor (hall)	-fullName -mobileNumber -email -teacherID -hallName -hallID -designation	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
University Staff	-fullName -mobileNumber -email -staffID -staffType	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Department Staff	-fullName	+select_students()

	-mobileNumber -email -staffID -staffType	+make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Hall Staff	-fullName -mobileNumber -email -staffID -staffType	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Library staff	-fullName -mobileNumber -email -staffID -staffType	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
University fee form	-formID -formType -studentName -studentRegistrationNo -session -departmentName	+update_student_info() +display_form() +update_student_database()
Semester fee form	-formID -formType -studentName -studentRegistrationNo -session -departmentName	+update_student_info() +display_form() +update_student_database()
Hall fee form	-formID -formType -studentName -studentRegistrationNo -session -departmentName -hallName	+update_student_info() +display_form() +update_student_database()

Hall fee form (admission)	-formID -formType -studentName -studentRegistrationNo -session -hallName	+update_student_info() +display_form() +update_student_database()
Admit card form	-formID -formType -studentName -studentRegistrationNo -session -departmentName -hallName	+update_student_info() +display_form() +update_student_database()

9.6.1 Analysis

After initially selecting the classes and reviewing the selected candidate class we decided to change some of the parameters that we initially held to be a class. We've selected them and merged them into 3 separate classes.

Teacher class: After carefully reviewing the class "Teacher" we saw that the attributes and methods are similar to the class "Director", "Chairman", "Dean", "House tutor". All of them have similar responsibility and also they collaborate with the same classes in the system. Thus we opted for all of these classes to be under one single class called "Teacher". There will be an attribute named "teacherType" which will determine the role of the class object.

Staff class: After carefully reviewing the class "Staff" we saw that the attributes and methods are similar to the class "University staff", "Library staff", "Department staff", "Hall staff". All of them have similar responsibility and also they collaborate with the same classes in the system. Thus we opted for all of these

classes to be under one single class called “Teacher”. There will be an attribute named “staffType” which will determine the role of the class object.

Form class: For handling the students' information we chose to create a single Form class which will encapsulate every type of information that is needed from a student when making any payment. This class will have an attribute called “formType” which will determine information the student has to give manually in order to complete the payment/event.

SMS and Email Class: We noticed that the SMS and Email class have very little functionality, so, rather than making separate classes, we decided to add these functionalities in User class to reduce complexity.

9.7. CRC Card

Class-responsibility-collaboration (CRC) cards are a brainstorming tool used in the design of object-oriented software. CRC cards are used after use case descriptions and before class diagrams within software development but can be skipped for smaller projects. CRC cards are usually created from index cards.

Class	Responsibility	Collaborator
User	• Login • Sign Up • Recover Password • Get user information • Set user information	Dashboard, Student, Teacher, Officials, Staff
Student	• Select Form • Fill Up Form • Select translation method • Money Transaction • Get Notification • Confirm Verification	User,Officials, Staff, Form, Transaction, Dashboard, Notification.
Teacher	• Get Notification • Verify Form	User,Officials, Form, Dashboard, Notification
Staff	• Get Notification • Verify Form	User,Officials, Form, Dashboard, Notification
Officials	• Recover user password • Post Notification • Get Notifications • Manage user • Manage database	User,Dashboard, Database, Notification

Form	• Get form information • Set form information	Student, Teacher, Staff, Database, Transaction,Bank
Dashboard	• Notify Announcement • Providing Form list • Show Form Verification Status • Show Payment Transaction Status	Student, Teacher, Staff, Officials, Form, Transaction.
Bank	• Receive payment • Generate payment confirmation • Send payment confirmation	Transaction.
Transaction	• Confirm Money Transaction • Update Form Info • Deny Transaction	Student,Form, Bank.
Notification	• Show Notification • Notification Expired Date • Notification Accessibility	Database,Teacher, Student, Staff, Officials.
Database	• Store info • Update Info	Student, Teacher, Staff, Officials, Form, Transaction,Notifications

9.8 Class Cards

After identifying our final classes we have generated the following class cards.

User	
Attribute	**Method**
-UserID -Password -FullName -Phone Number -Email Address -Address -UserType	+getuserID() +setUserID() +getuserType() +setUserType() +getFull_name() +setFull_name() +getMobile_number() +setMobile_number() +getEmail_address() +setEmail_address() +getPassword() +setPassword() +create_account() +login() +send_email_verification_link() +send_OTP_by_SMS() +verify_info()
Responsibilities	**Collaborator**
• Get account information • Set account information	Dashboard, Student, Teacher, Officials, Staff

Student	
Attribute	**Method**
-FullName -Mobile number -Address -Email address -StudentId -Roll No -Registration No -Department name -Hall Name -Session -User name -Password -SSC Information -HSC Information	+select_form() +form_fill_up() +choose_payment_type() +make_payment() +send_QR() +recieve_document() +update_info() +cancel_form() +recieve_confirmation() +show_notification() +get_notification()
Responsibilities	**Collaborator**
• Select Form • Fill Up Form • Select translation method • Money Transaction • Get Notification • Confirm Verification	User, Officials, Staff, Form, Transaction, Dashboard, Notification.

Teacher	
Attribute	**Method**
-First name -Last name -Mobile number -Email address -TeacherId -TeacherDesignation -TeacherType -Hall Name -Department name -User name -Password	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()
Responsibilities	**Collaborator**
• Get Notification • Verify Form • Search the database using the given search parameters • Update database	User,Officials, Form, Dashboard, Notification

Staff	
Attribute	**Method**
-First name -Last name -Mobile number -Email address -Staff Id -staffType -Office Name -Username -Password	+select_students() +make_group() +verify_form() +update_info() +recieve_confirmation() +get_notification()

Responsibilities	Collaborator
• Search the database using the given search parameters • Update database • Get Notification • Verify Form	User,Officials, Form, Dashboard, Notification

Official	
Attribute	**Method**
-First name -Last name -Mobile number -Email address -Official Id -Username -Password -Notification ID	+select_students() +make_group() +verify_info() +generate_notification() +send_notification_to_teacher() +send_notification_to_student() +send_notification_to_staff() +annouce_activities() +verify_form() +verify_user_information()
Responsibilities	**Collaborator**
• Recover aAccount password • Verify User Account • Post Notification • Get Notifications • Manage Account • Manage database	User,Dashboard, Database, Notification.

Dashboard	
Attribute	**Method**
-userID -userType -notificationID -FormList	+notify_user() +highlight_unread_notification() +show_form_menu() +show_form_verification() +show_payment_verification()
Responsibilities	**Collaborator**
• Notify Announcement • Providing Form list • Show Form Verification Status • Show Payment Transaction Status	Student, Teacher, Staff, Officials, Form, Transaction.

Bank	
Attribute	**Method**
-BankID -BranchName -PaymentAmount -PaymentConfirmation	+receive_payment() +generate_payment_confirmation() +send_confirmation() +send_QR()
Responsibilities	**Collaborator**
• Receive payment • Generate payment confirmation • Send payment confirmation	Transaction, Form

Form	
Attribute	**Method**
-FormID -FormType -StudentName -StudentRegistrationNo -Session -DepartmentName	+update_student_info() +display_form() +update_student_database() +update_verification_info()
Responsibilities	**Collaborator**
• Get form information • Set form information • Show verification info	Student, Teacher, Staff, Database, Transaction,Bank

Transaction	
Attribute	**Method**
-Transaction ID -Student ID -Bank ID -Transaction method -Amount -Time -Date	+allow_confirmation() +update_form() +deny_confirmation() +match_QR()
Responsibilities	**Collaborator**
• Confirm Money Transaction • Update Form Info • Deny Transaction	Student, Form, Bank

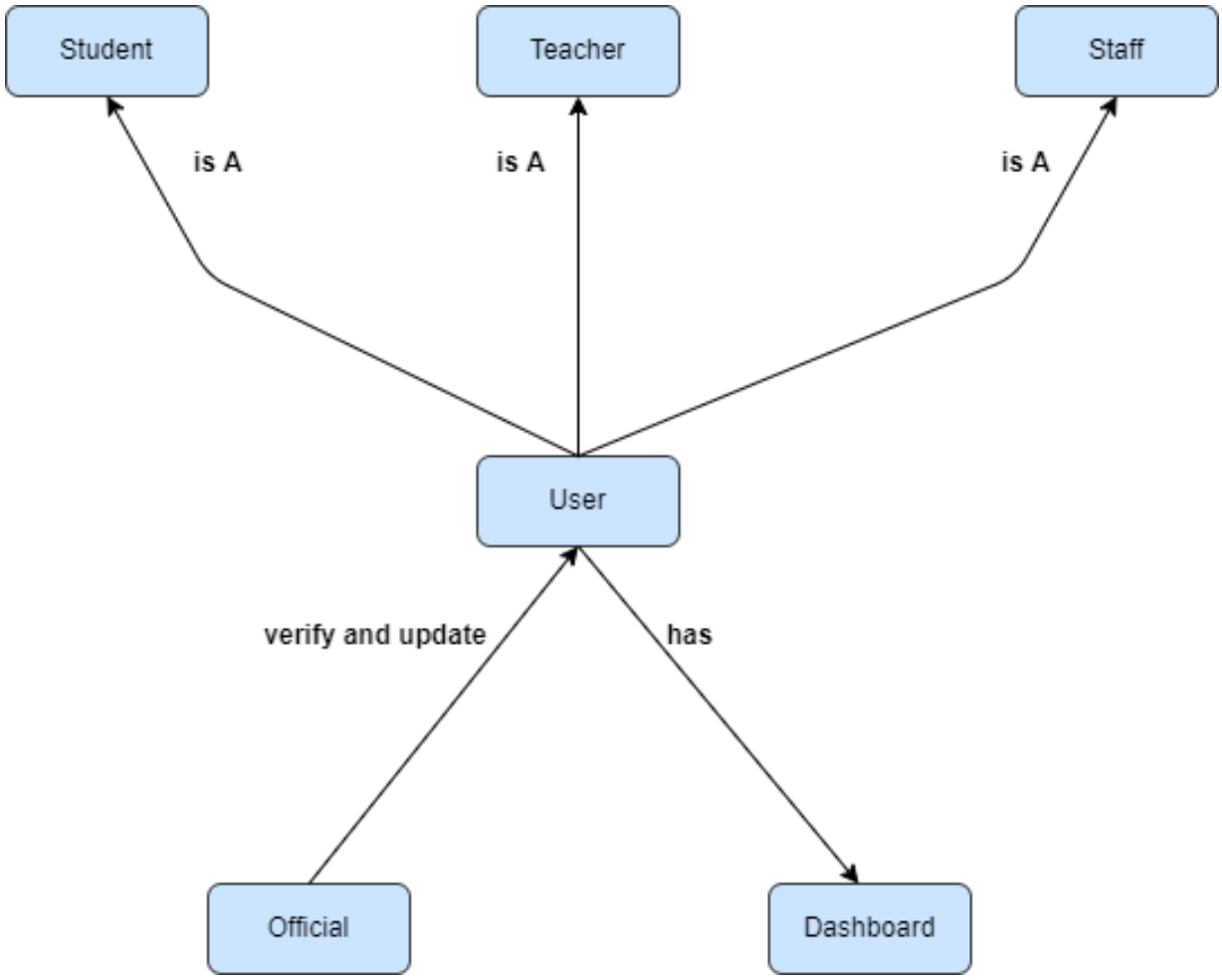
Notification	
Attribute	**Method**
-Notification ID -Notification Title -Notification Description -Time -Date -Official ID -Notification Type	+setID() +setTitle() +setDate() +setType() +notification_status() +notification_visibility() +show_notification()
Responsibilities	**Collaborator**
• Show Notification • Notification Expired Date • Notification Accessibility	Database,Teacher, Student, Staff, Officials.

Database	
Attribute	**Method**
-StudentList -TeacherList -StaffList -OfficialList -FormList -TransactionList	-update() -delete() -insert() -search()
Responsibilities	**Collaborator**
• Search the database using the given search parameters • Update database	Student, Teacher, Staff, Officials, Form, Transaction,Notifications.

9.9. CRC Diagram

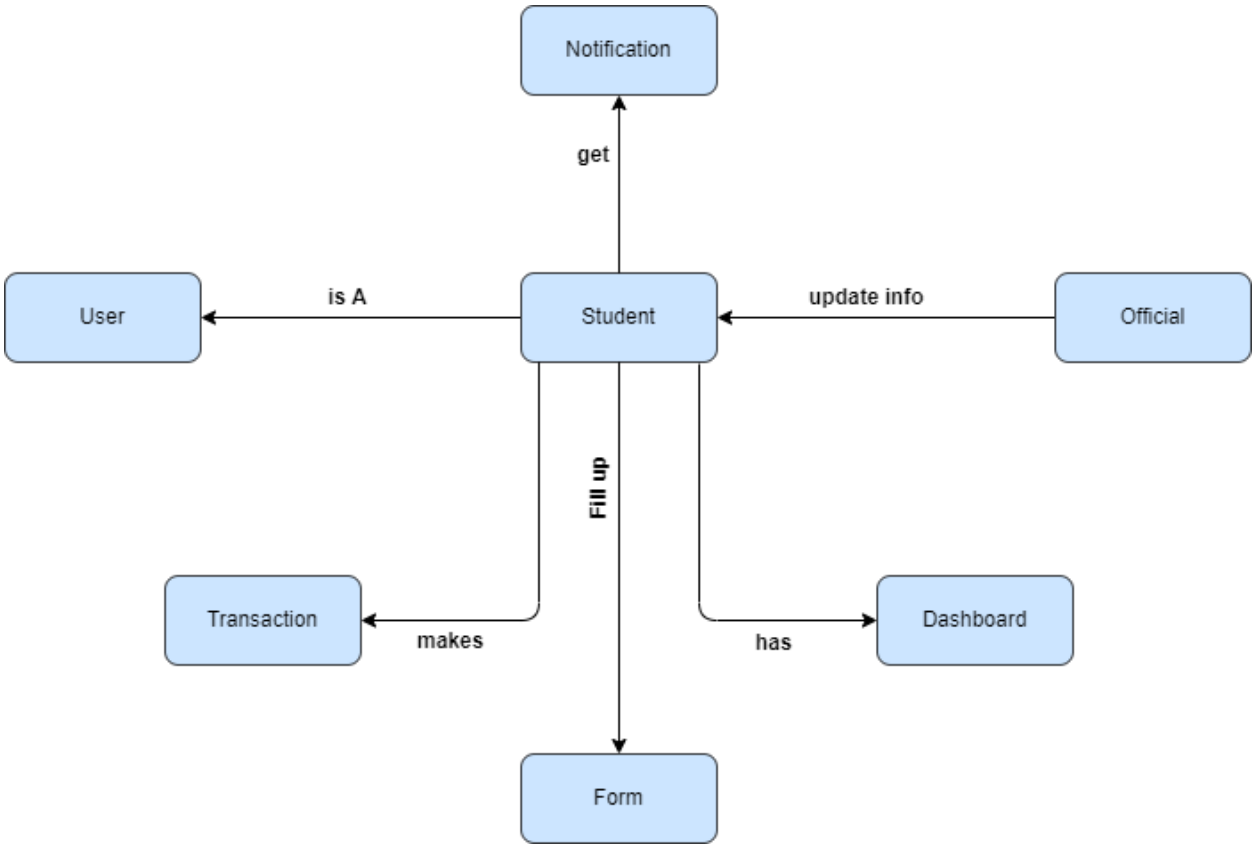
9.9.1. Diagram ID: 1

Name: User



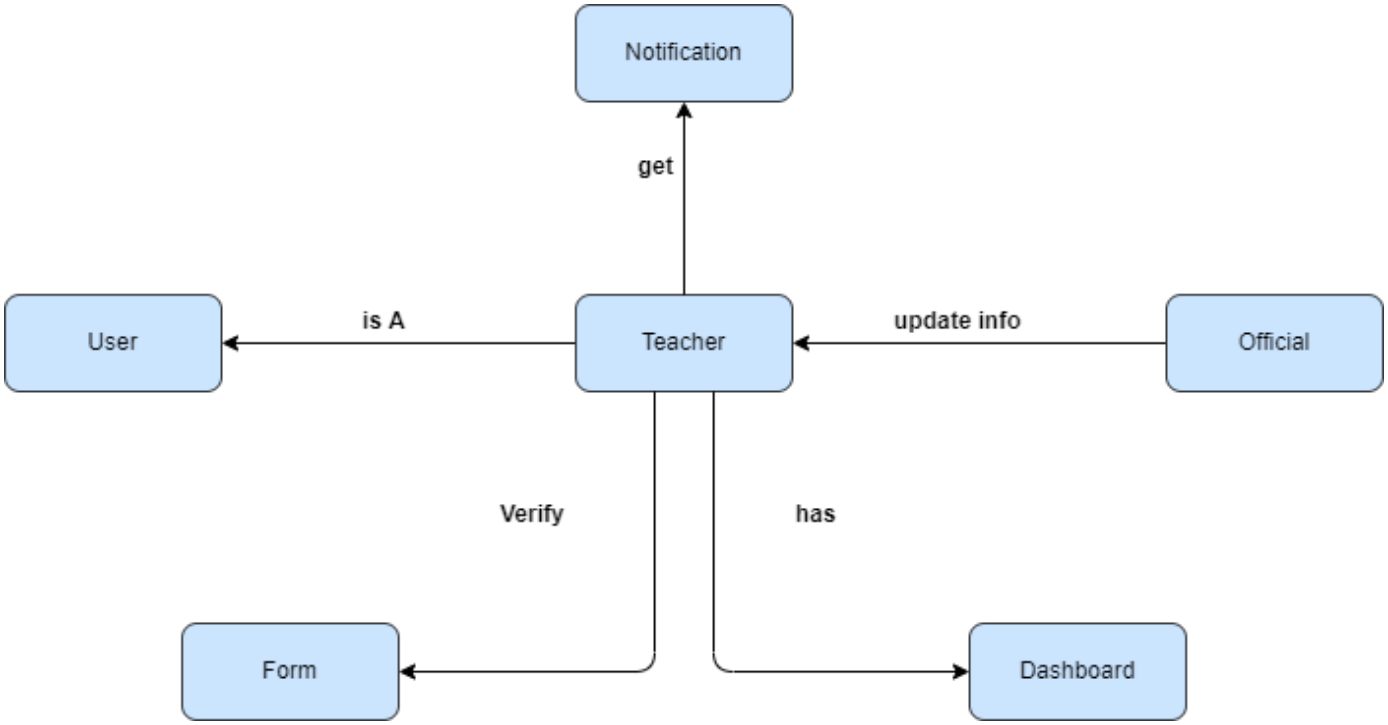
9.9.2. Diagram ID: 2

Name: Student



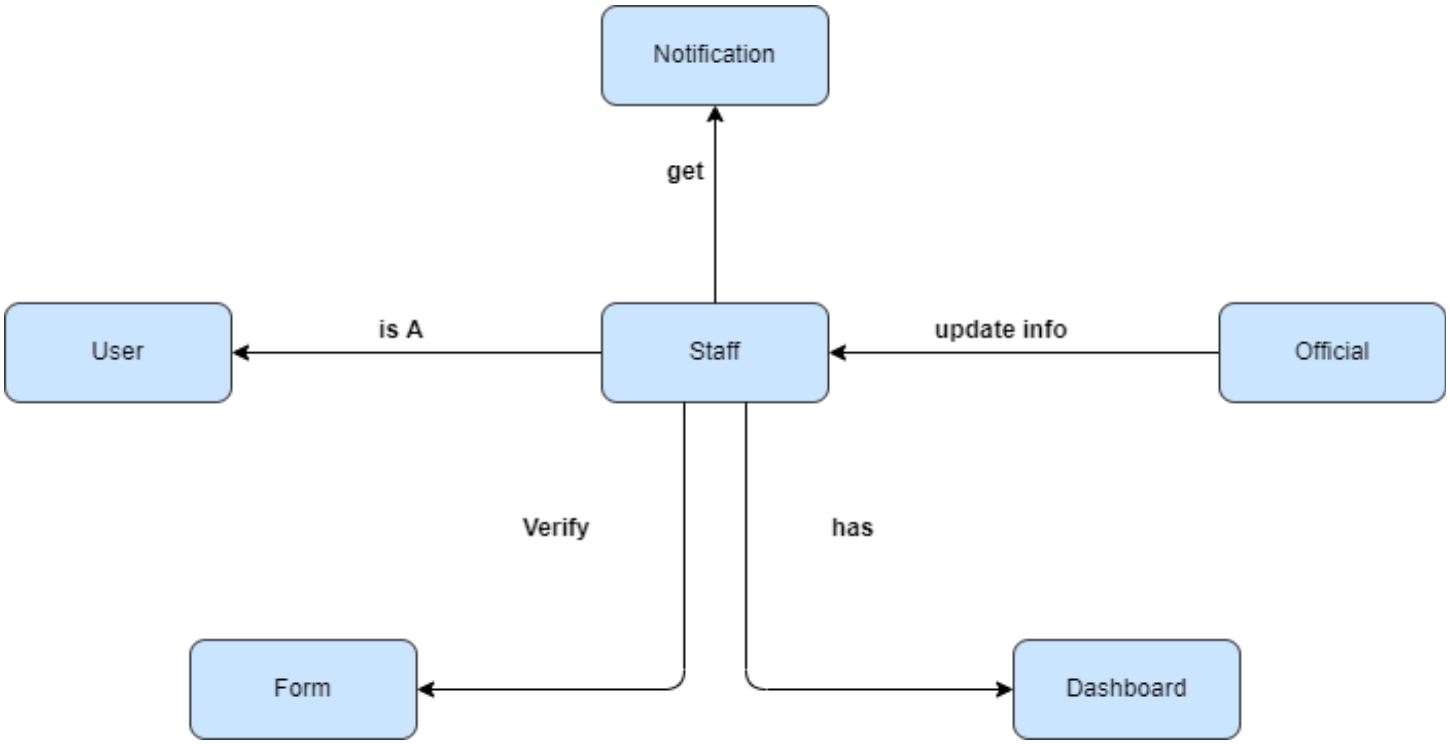
9.9.3. Diagram ID: 3

Name: Teacher



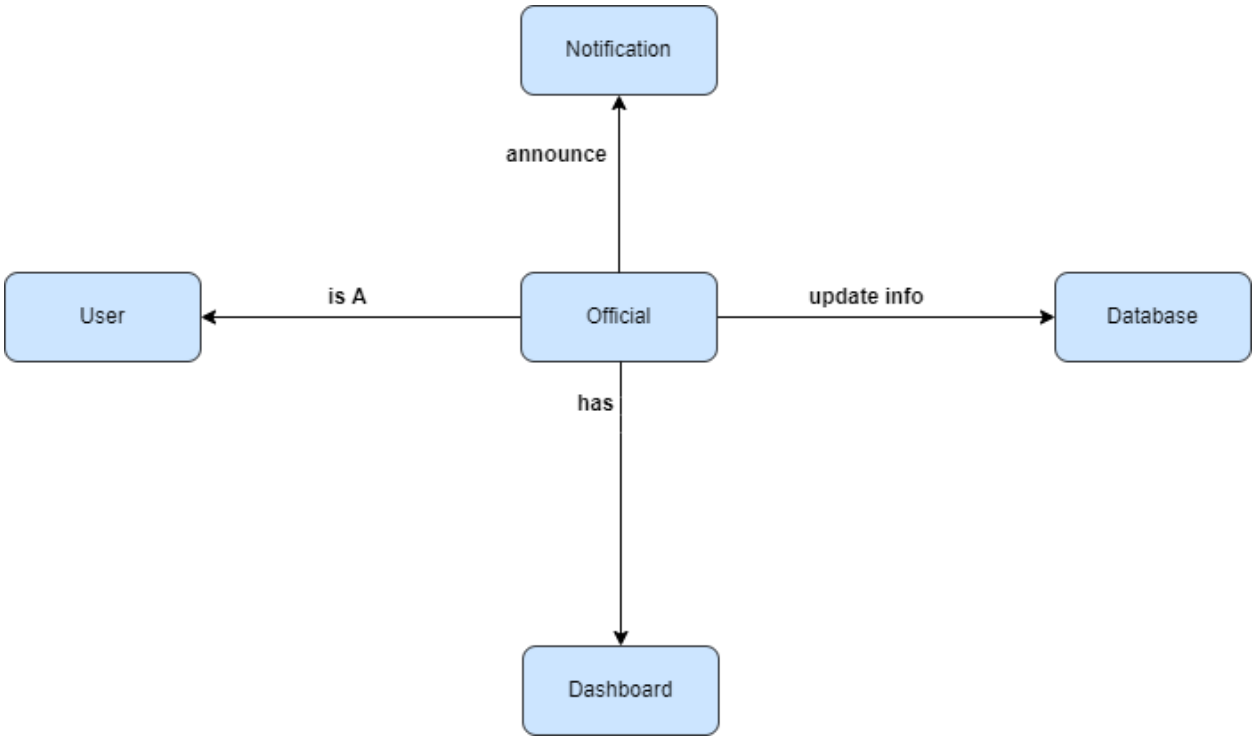
9.9.4. Diagram ID: 4

Name: Staff



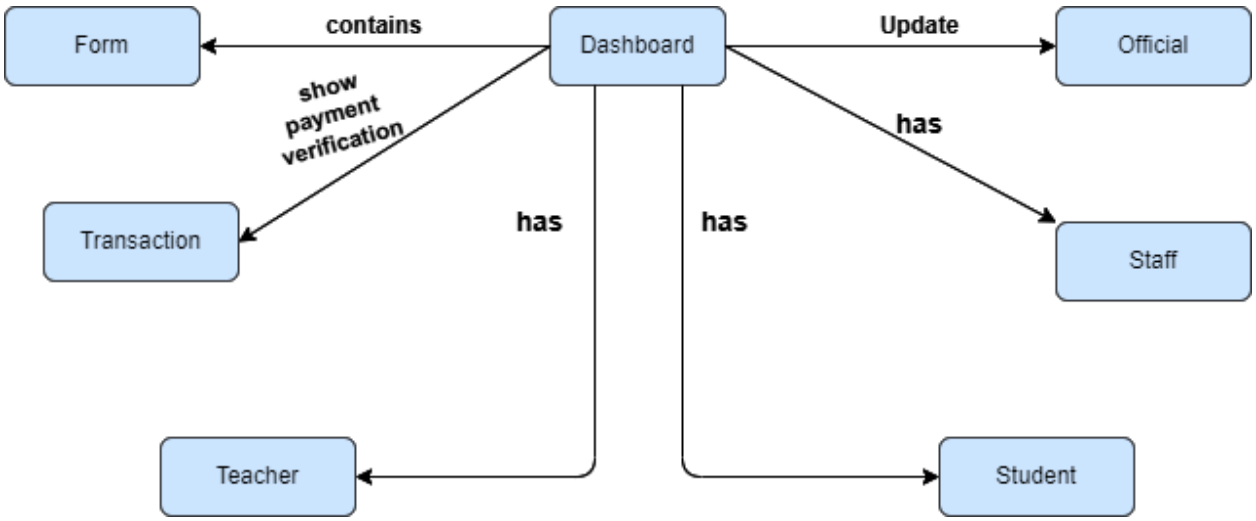
9.9.5. Diagram ID: 5

Name: Official



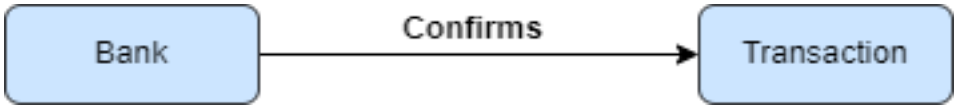
9.9.6. Diagram ID: 6

Name: Dashboard



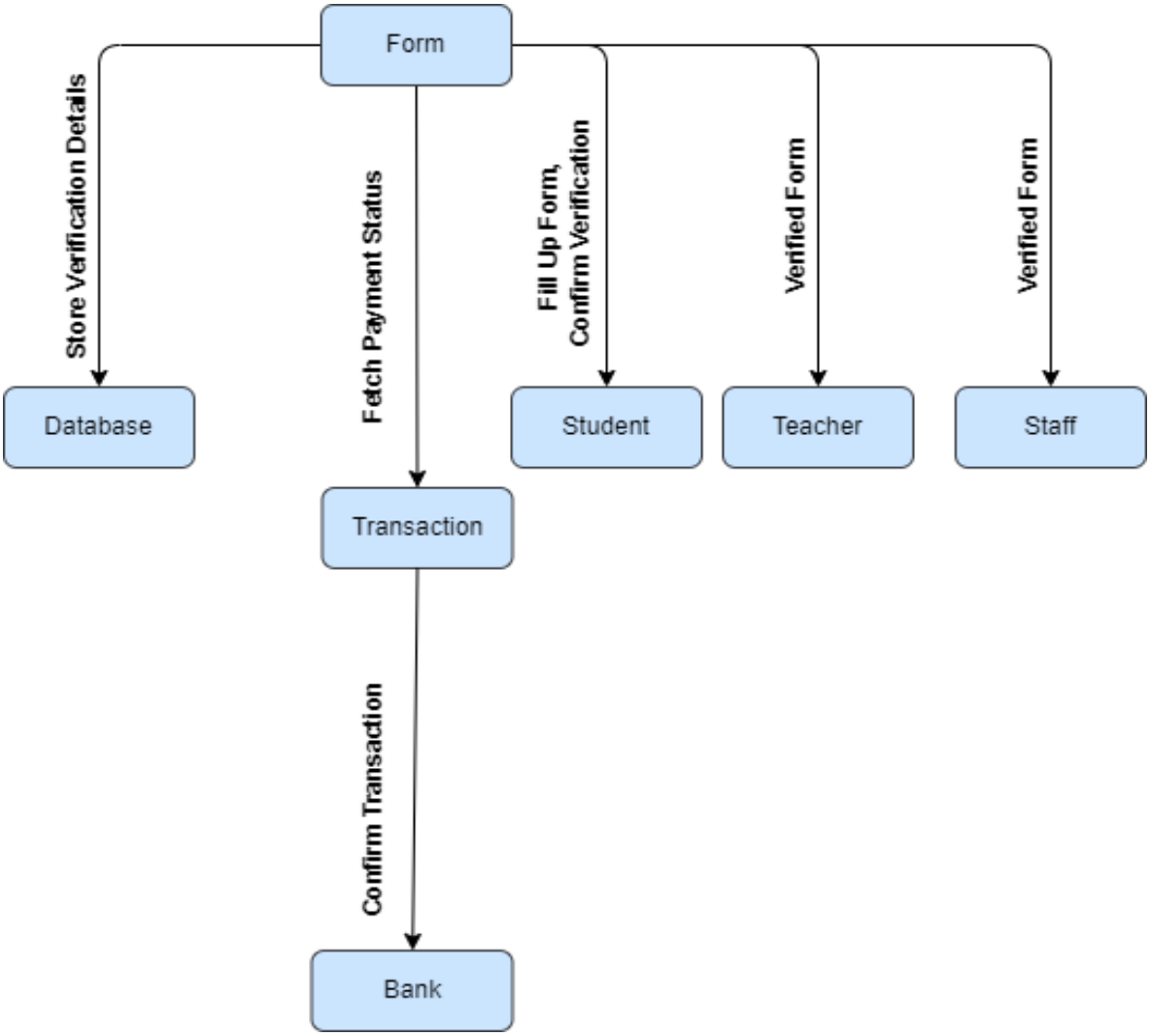
9.9.7. Diagram ID: 7

Name: Bank



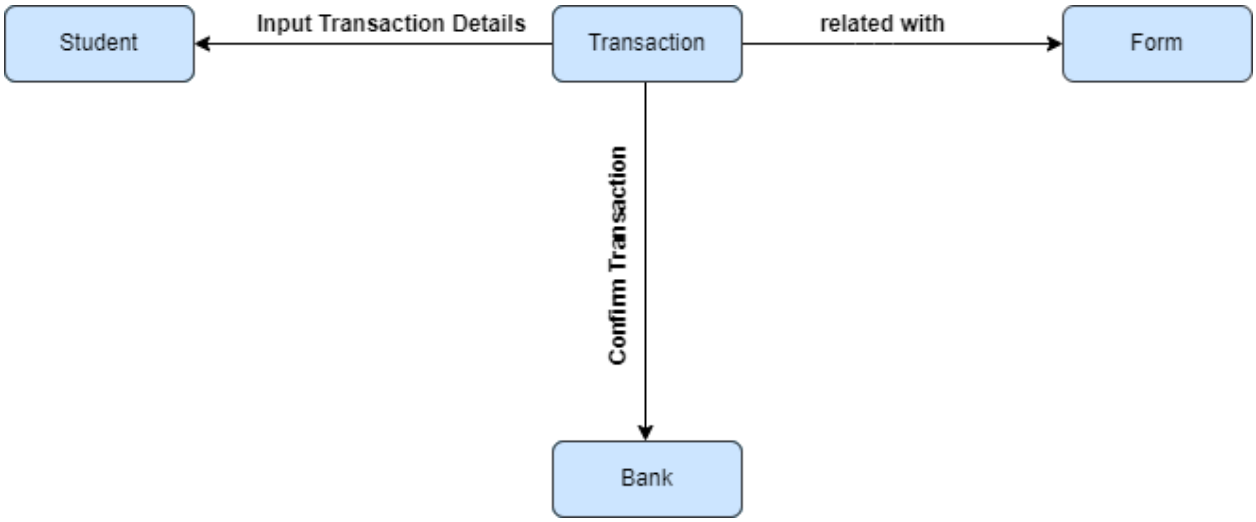
9.9.8. Diagram ID: 8

Name: Form



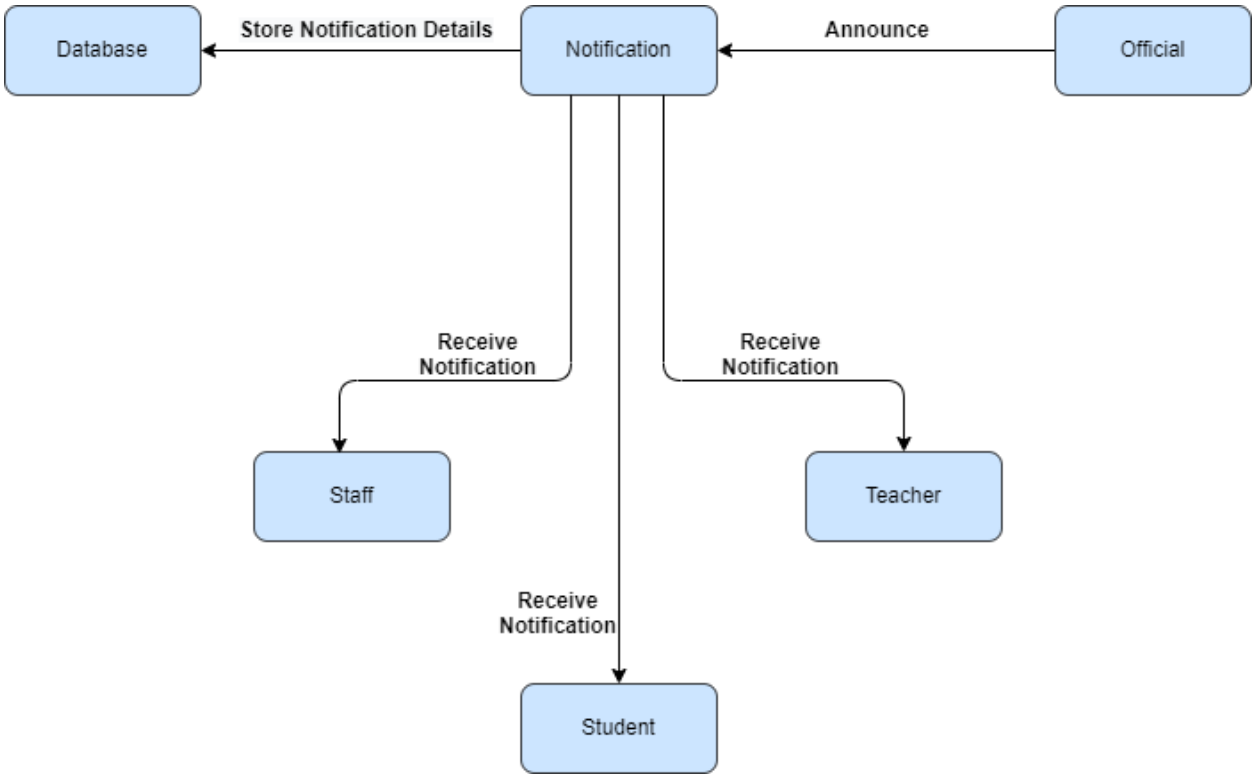
9.9.9. Diagram ID: 9

Name: Transaction



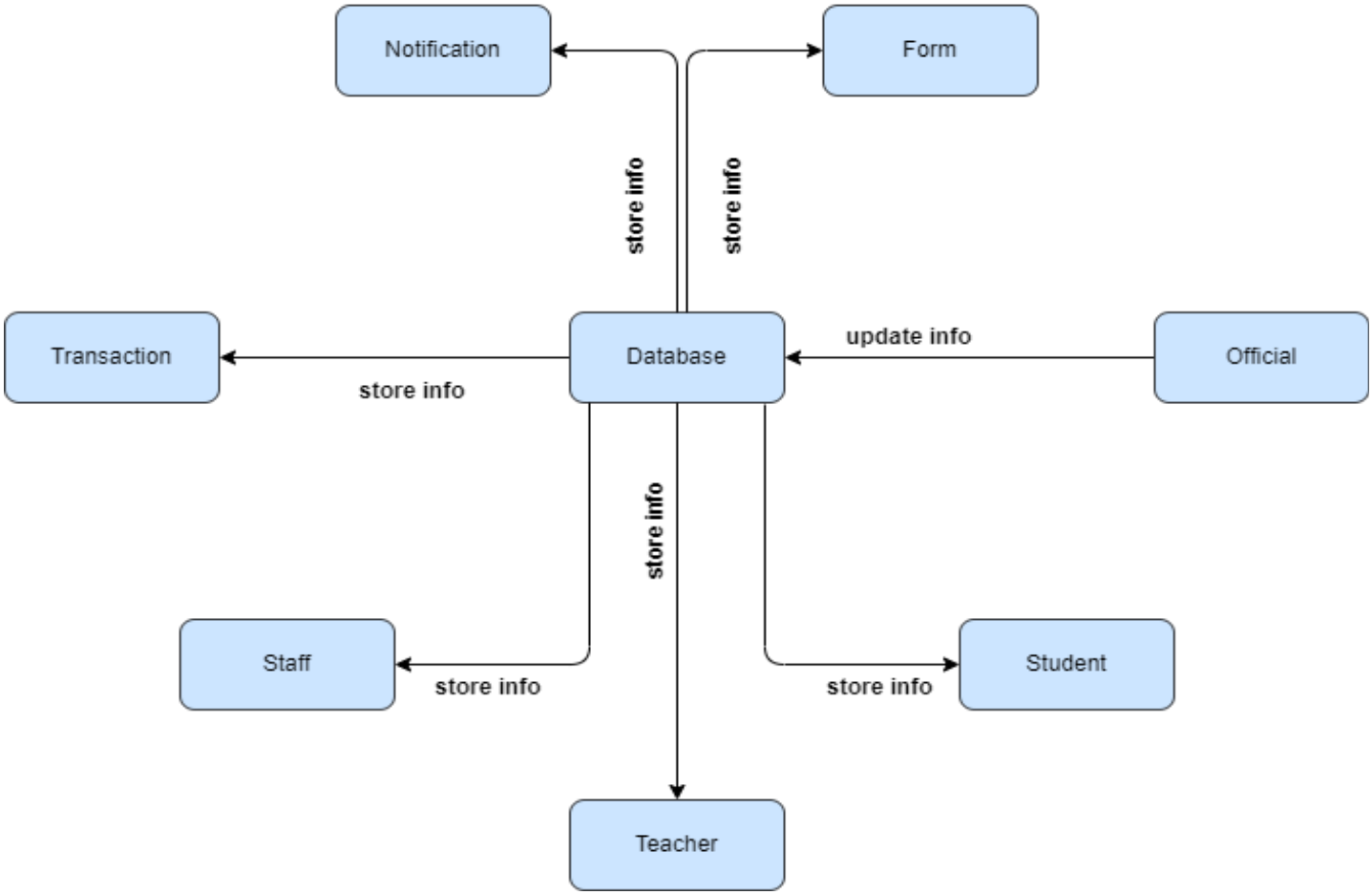
9.9.10. Diagram ID: 10

Name: Notification



9.9.11. Diagram ID: 11

Name: Database



10. Behavioral Modeling

10.1. State Transition Diagram

State diagram represents active states for each class of the events (triggers). For this, we identified all the events, their initiators, and collaborators.

10.2. Event Table

Sl. No	Event	State Name	Initiator	Collaborator	Associated Method
1.	Will create an account	Create_ account	Student, Teacher, Staff	User	+create_ account() +verify_ info() +notify_ user() +send_confirmation()
2.	Will provide information	Provide_ info	User		+create_ account()
3.	Users' credentials Will be verified	Verify_ info	Official	User	+verify_ info()
4.	Will update information	Update_ information	Student, Teacher, Staff, Official	User, Database	+update_ info() +setFull_ name() +setMobile_ number() +setEmail_ address() +setTeacher/officer_ id() +setDepartment_ name() +setRegNo() +setLocation()

5.	Will be able to recover password	Recover_ password	User, Student, Teacher, Staff, Official	Official	+recover_ password()
6.	Will login to System	log_in	Student, Teacher, Staff, Official		+login()
7.	Will Select Form	Select_Form	Student	Dashboard, Form	+display_ form() +select_form()
8.	Will be able to see the form	Display_Form	Form	Student, Teacher, Staff, Officials	+display_ form()
9.	Will be able to fill up Form information	fill_up_form	Student	Form	+form_fill_up() +display_ form()
10.	Will be able to update form information	update_form	Form	Student	+update_student_info()
11.	Will be able to update database	update_database	Form, User, Teacher, Student, Staff, Official	Database	+update_student_database()
12.	Will provide from menu	Update_ menu	Official	Dashboard	+fix_ menu() +update_ menu()
13.	Will verify form	Verify_Form	Teacher, Staff	Form	+show_list_of_request() +make_group_reques

					t() +verify() +select() +select_all() +view_full_form_details() +add_comment()
14.	Will add to dashboard	Add_to_Dashboard	Officials	Dashboard, Form	+add_dashboard() +show_dashboard()
15.	Will Confirm verification	confirm_verification	Teacher, Staff, Student	User	+send_sms() +send_email() +add_status_to_dashboard()
16.	Can pay the from fee	Pay_Form_Fee	Student	Transaction, Form, Bank	+select_payment_type() + +online_payment() +mobile_banking_payment() +manual_payment() +make_ transaction()
17.	Will send payment confirmation	confirm_transaction	Bank, Transaction	SMS, Email, Student	+notify_ after_transaction() +send_ confirmation() +send_sms() +send_email()
18.	Will Announce Event	Announce_Event	Official	Teacher, Student, Notification	+make_event_title() +add_starting_date() +add_deadline() +add_description() +add_form_type() +make_group_type() +select() +select_all() +view_full_annoucement_details()

19.	Will generate notification	generate_notification	Official	Notification	+generate_notification()
20.	Payment Confirmation	payment_confirmation	Bank	Transaction, Form	+payment_confirmation()
21.	show event notification	show_notification	Notification	User, Dashboard	+get_notification() +show_notification()
22.	Show Notification Status	show_notification_status	Notification		+show_notification_status()
23.	Send Confirmation	send_confirmation	Bank	Staff, Officials	+send_confirmation()
24.	Receive Confirmation	receive_confirmation	Staff, Officials	Bank	+receive_confirmation()
25.	Show dashboard	display_dashboard	Dashboard		+show_dashboard()
26.	show menu content according to user type	show_menu	Dashboard		+show_menu()
27.	Cancel form	cancel_form	Student	Form	+cancel_form()
28.	Select Accounts from database	select_student_account	Official	Database	+select_account()
29.	Make group of students	make_group	Official, Teacher, Staff	Database, Student	+make_group()

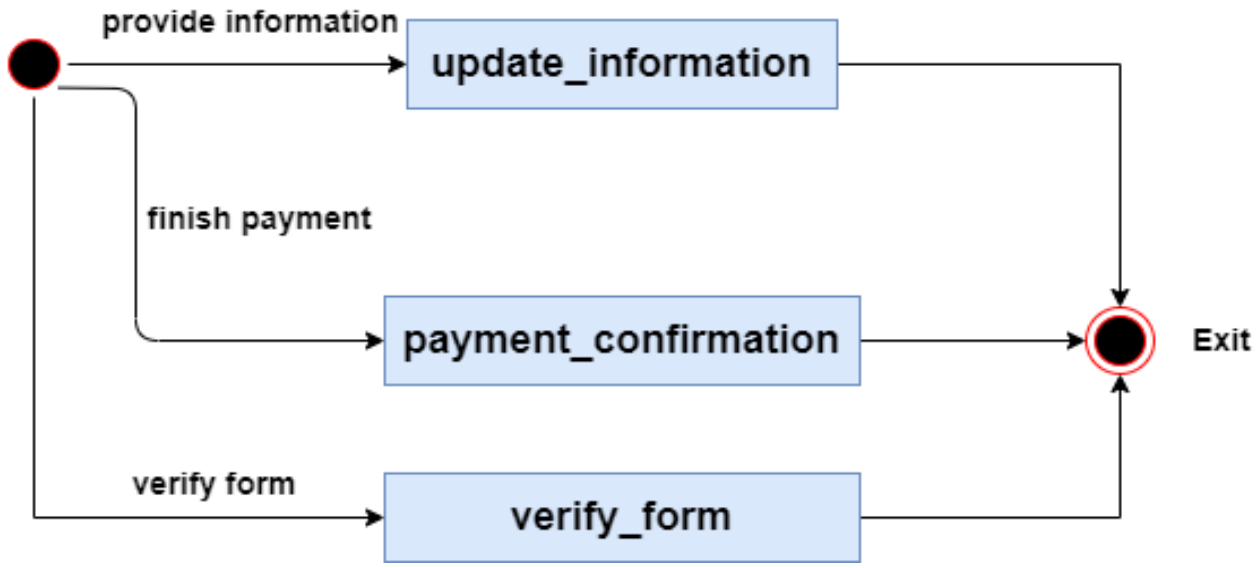
30.	Verify account information	verify_account_information	User		+verify_account_information
31.	Send Notification	send_notification	Official	Notification, Dashboard	+send_notification
32.	Will make payment	make_payment	Student	Transaction	+make_payment
33.	Choose payment method	choose_payment_type	Student	Transaction	+choose_payment_type
34.	Will provide payment info	provide_info	Student, Transaction	Bank	+input_transaction_method_type() +input_transaction_id()
35.	Cancel payment confirmation	cancel_confirmation	Bank	Transaction	+cancel_transaction_confirmation()
36.	Update status in form	update_form	Bank	Transaction, Form, Database	+update_payment_confirmation_status
37.	Wants to verify by email	verify_info_email	User		+send_email_verification_link() +verify_info()
38.	Wants to verify by SMS	verify_info_SMS	User		+send_OTP_by_SMS() +verify_info()
39.	Wants to verify manually	verify_info_manual	User	Official	+verify_info()
40.	Get notification	receive_notification	Notification	Student, Teacher,	+get_notification() +show_notification()

	according to user type			Staff, Official	
41.	Input QR code for manual payment	send_QR_code	Student, Bank	Transaction	+send_QR()
42.	Match QR code from both bank and student	match_QR_code	Transaction	Student, Bank	+match_QR()
43	Wants to notify user	notify_user	Dashboard	Student, Teacher, Staff	+show_notification()

10.3. State Transition

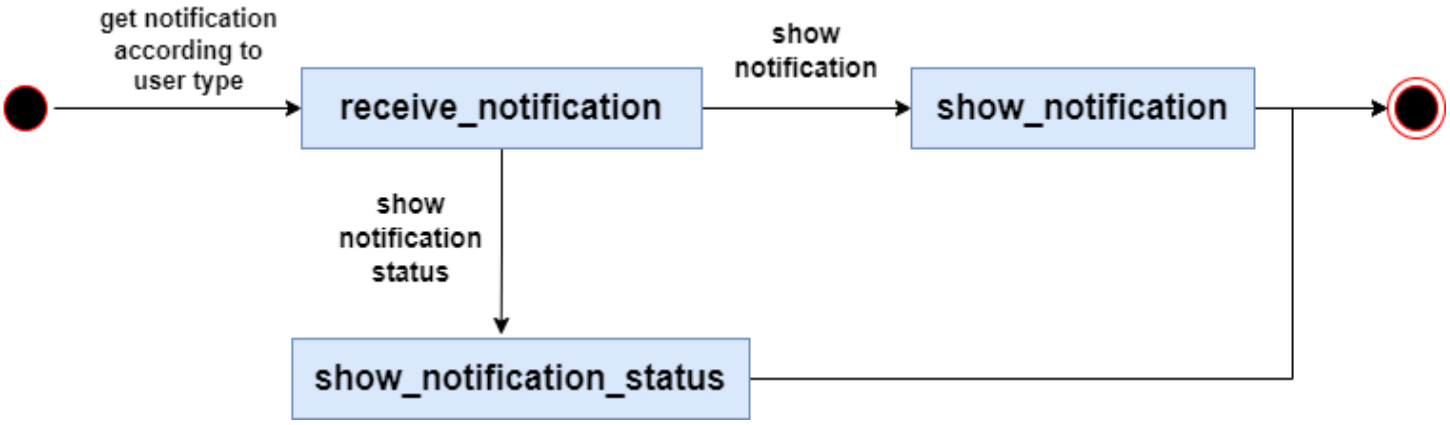
10.3.1. ID: 1

Name: Form



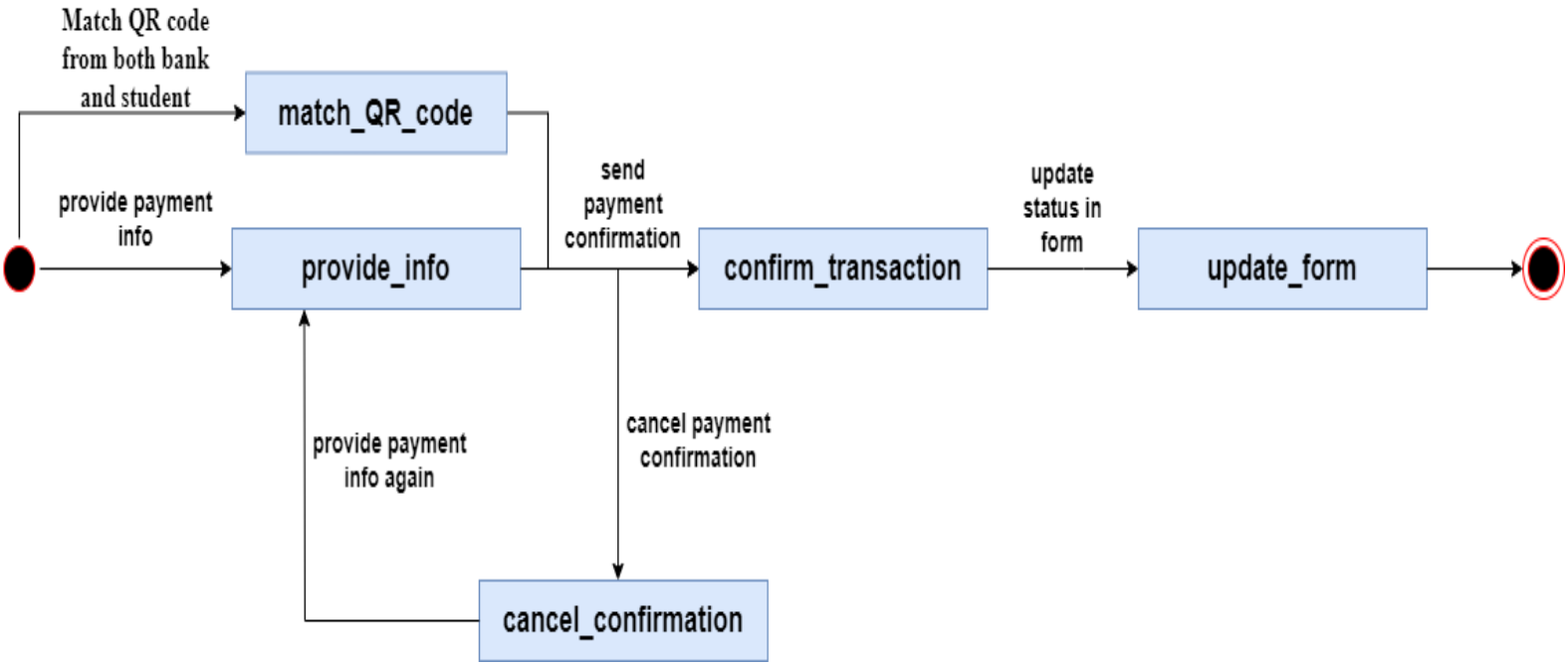
10.3.2. ID: 2

Name: Notification



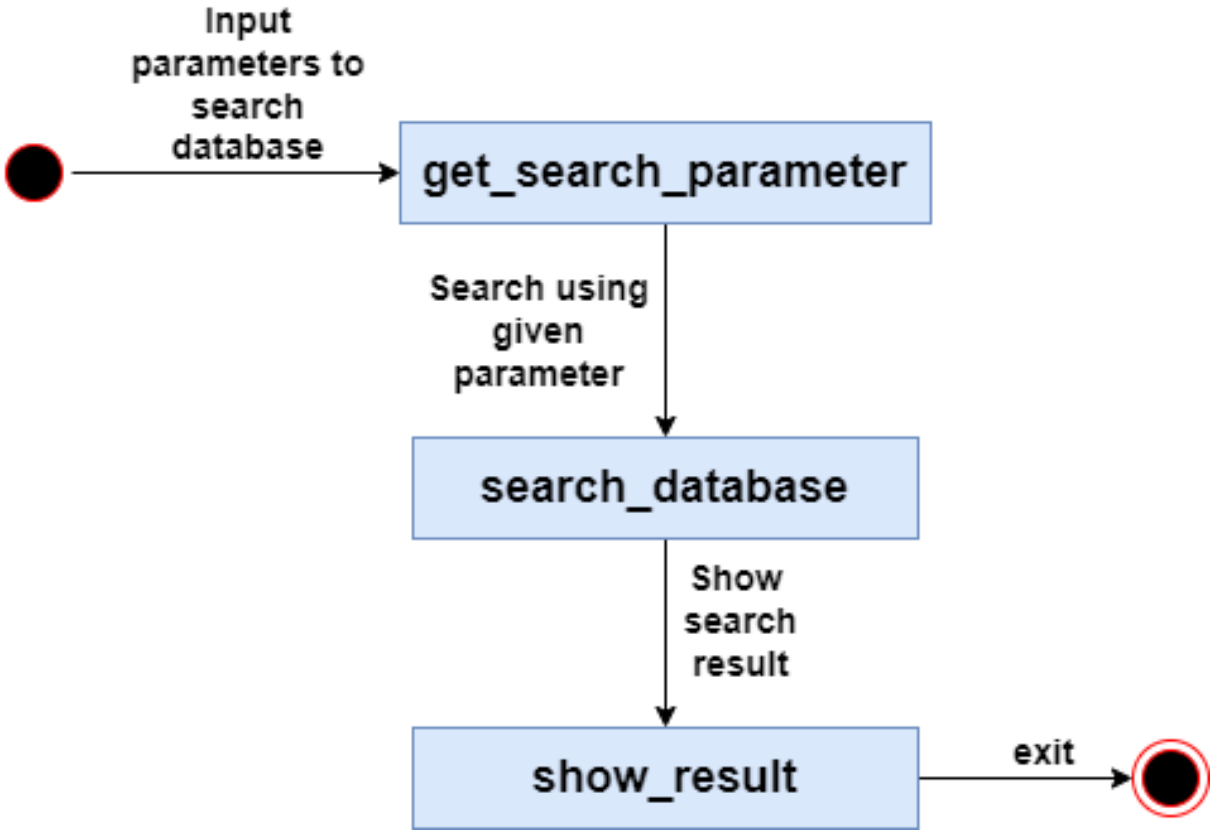
10.3.3. ID: 3

Name: Transaction



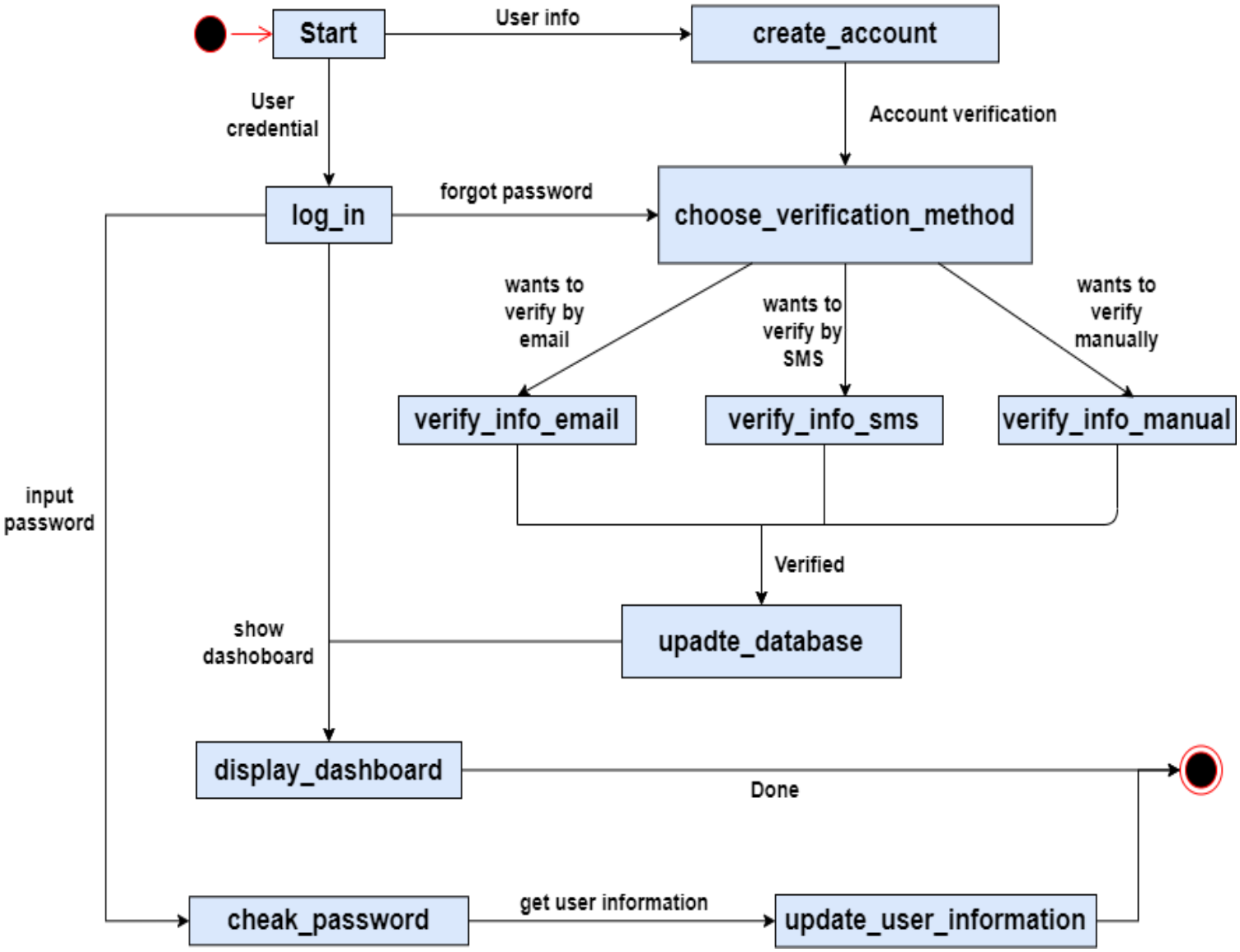
10.3.6. ID: 4

Name: Database



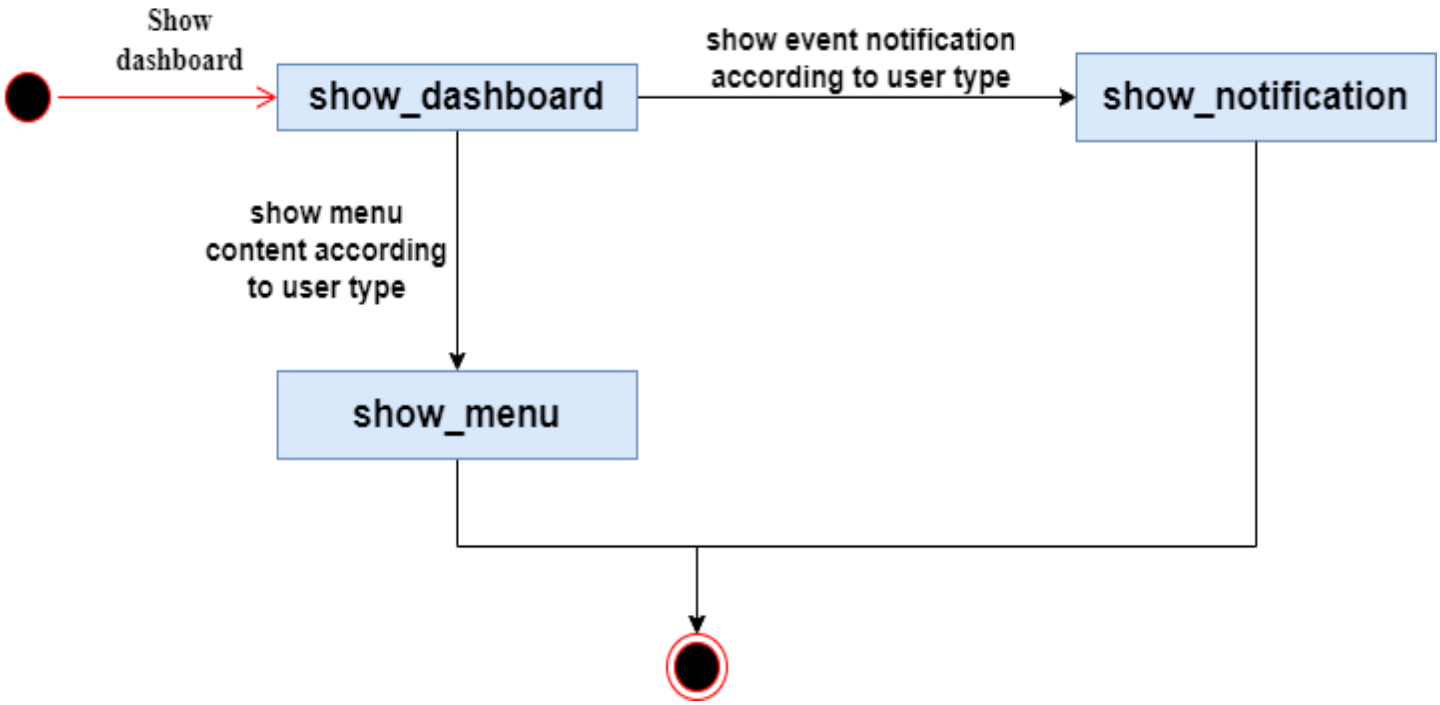
10.3.9 ID: 5

Name: User



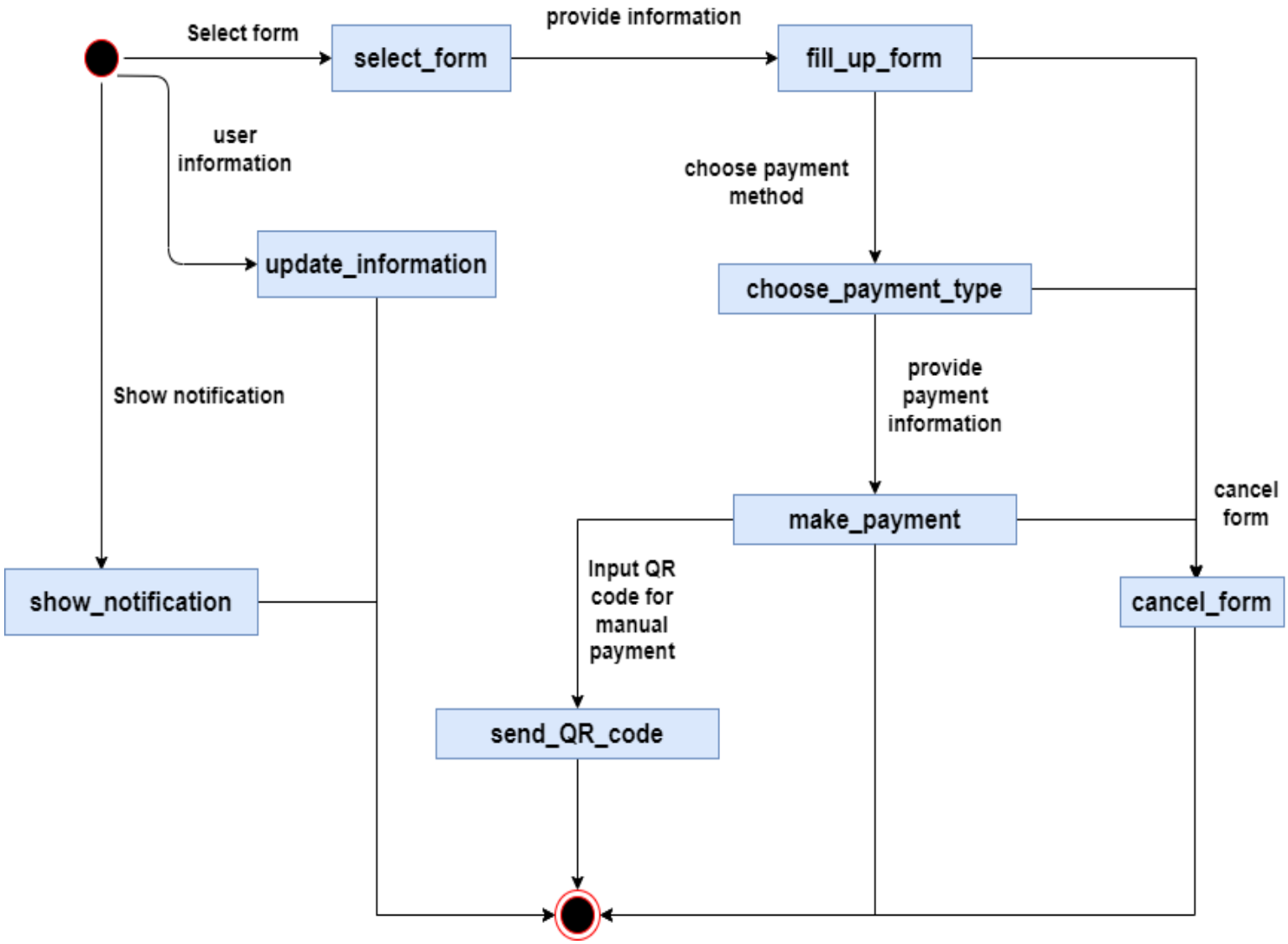
10.3.10. ID: 6

Name: Dashboard



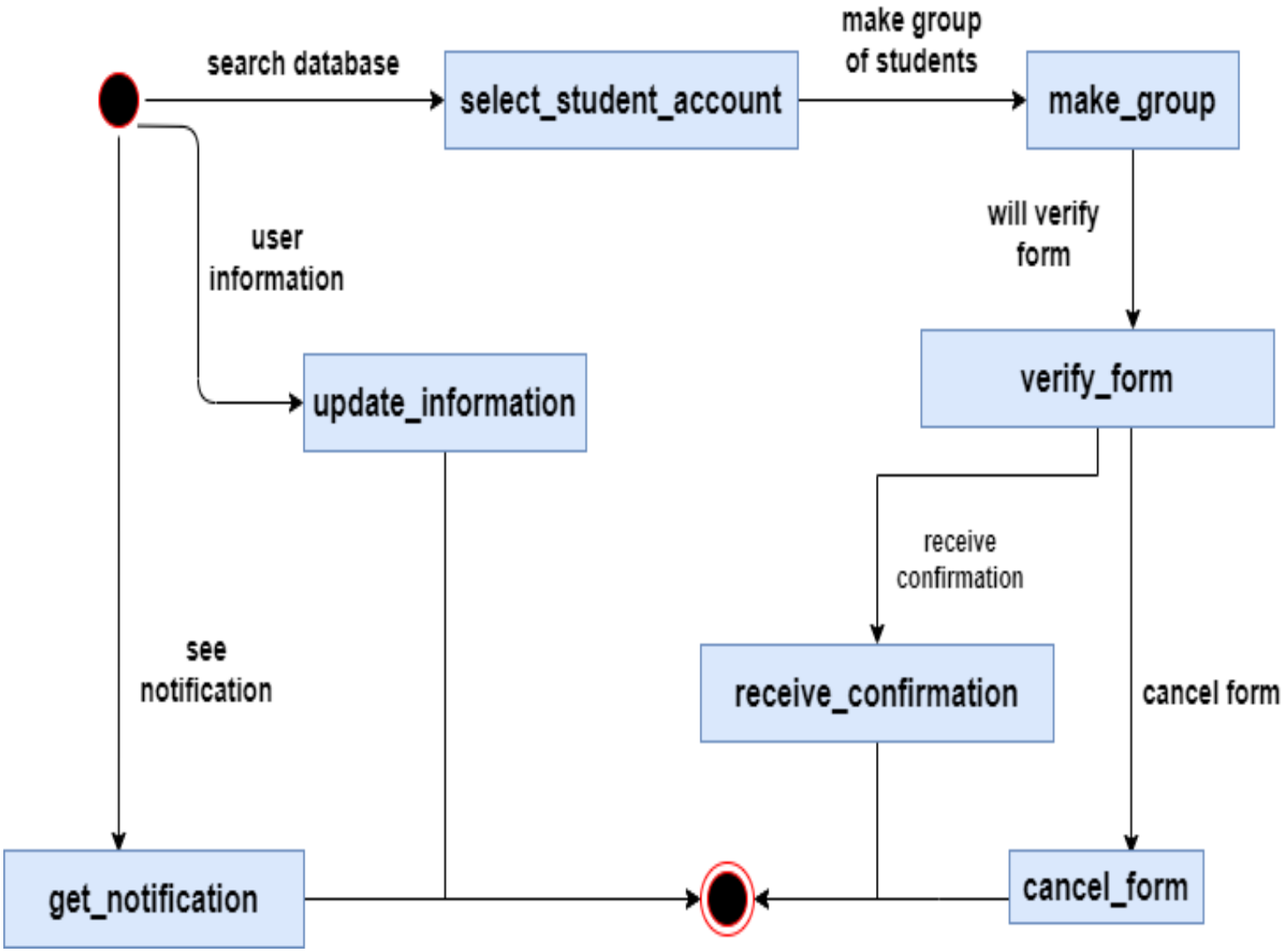
10.3.12 ID: 7

Name: Student



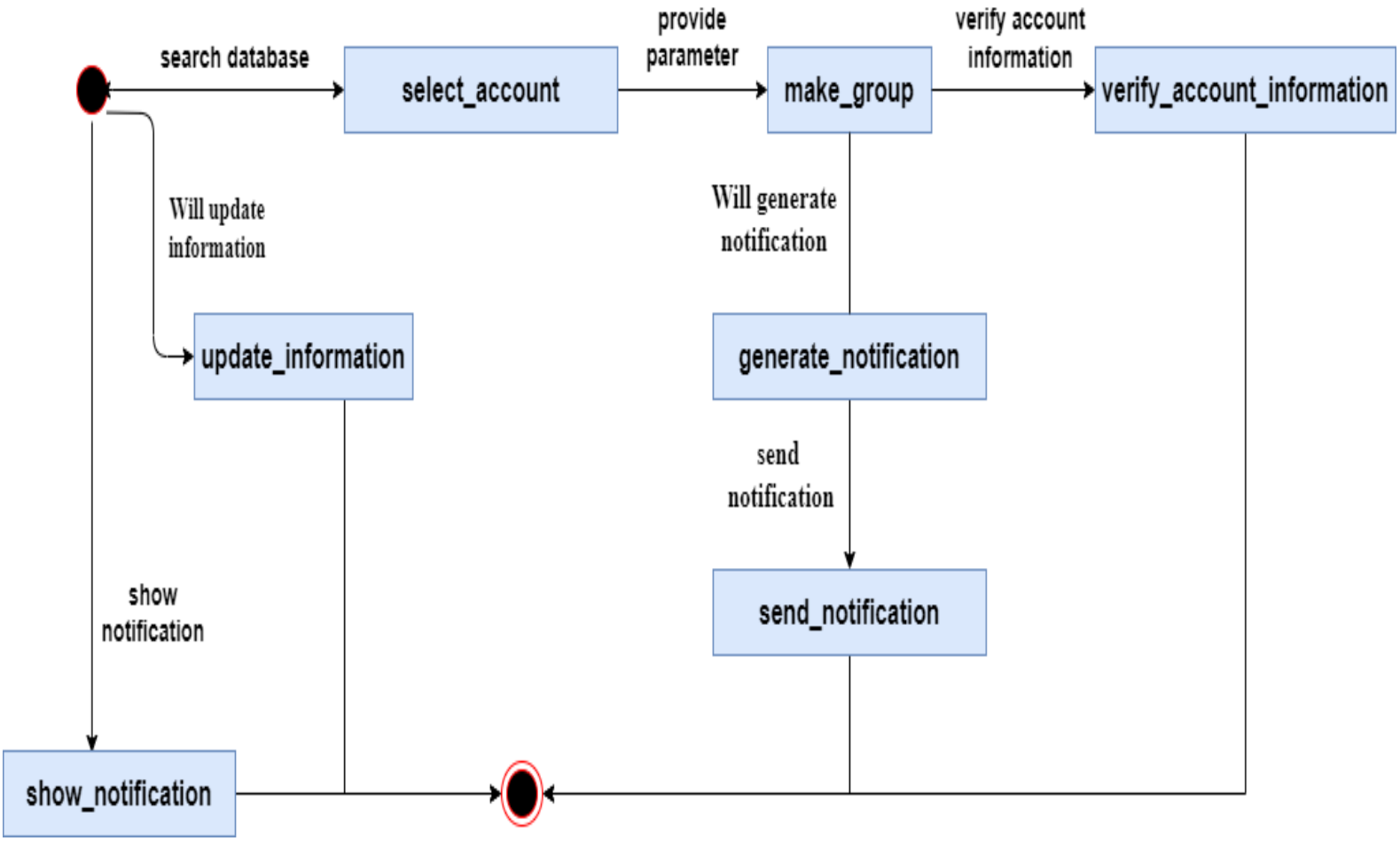
10.3.13. ID: 8

Name: Teacher



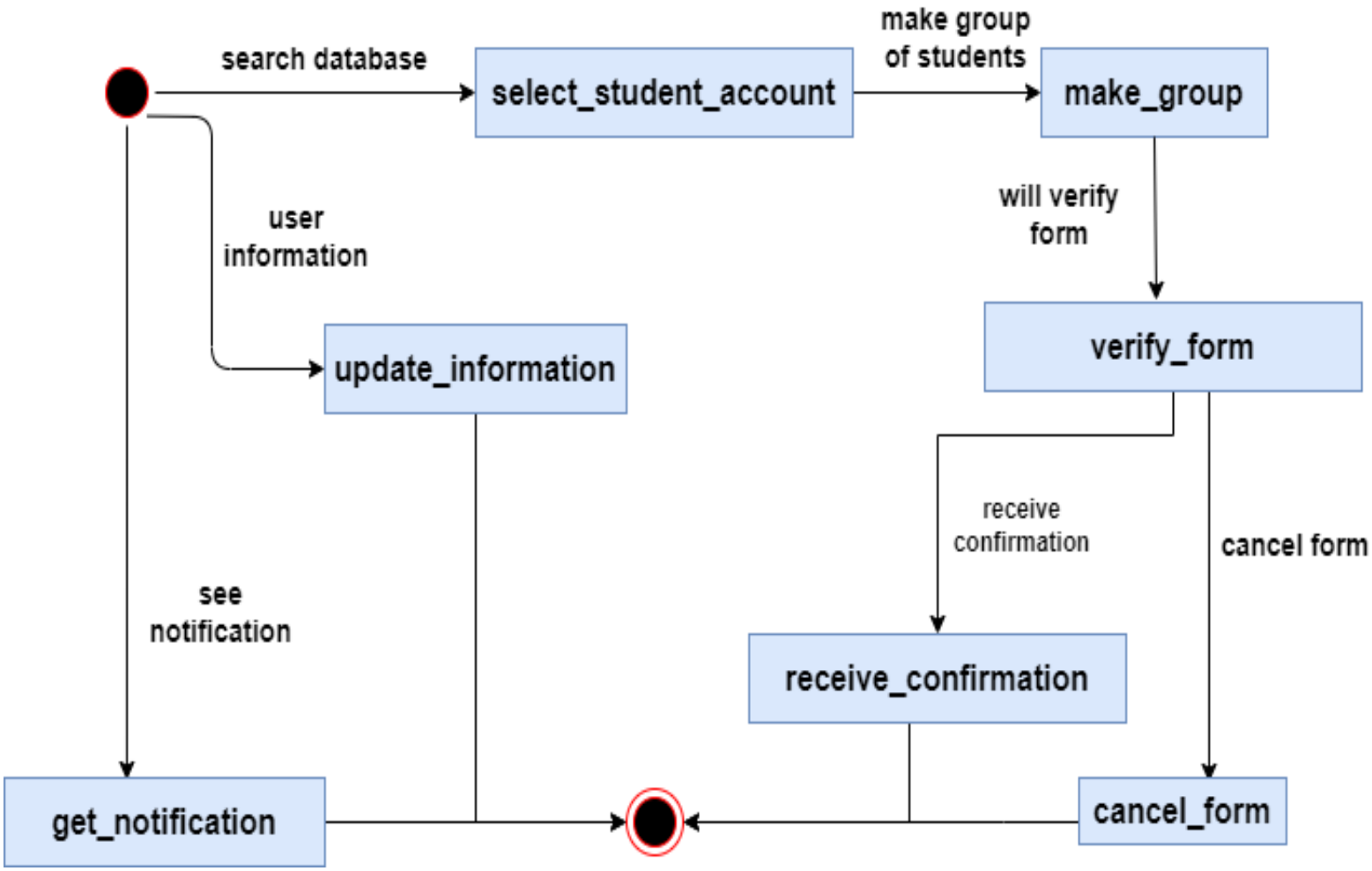
10.3.14. ID: 9

Name: Official



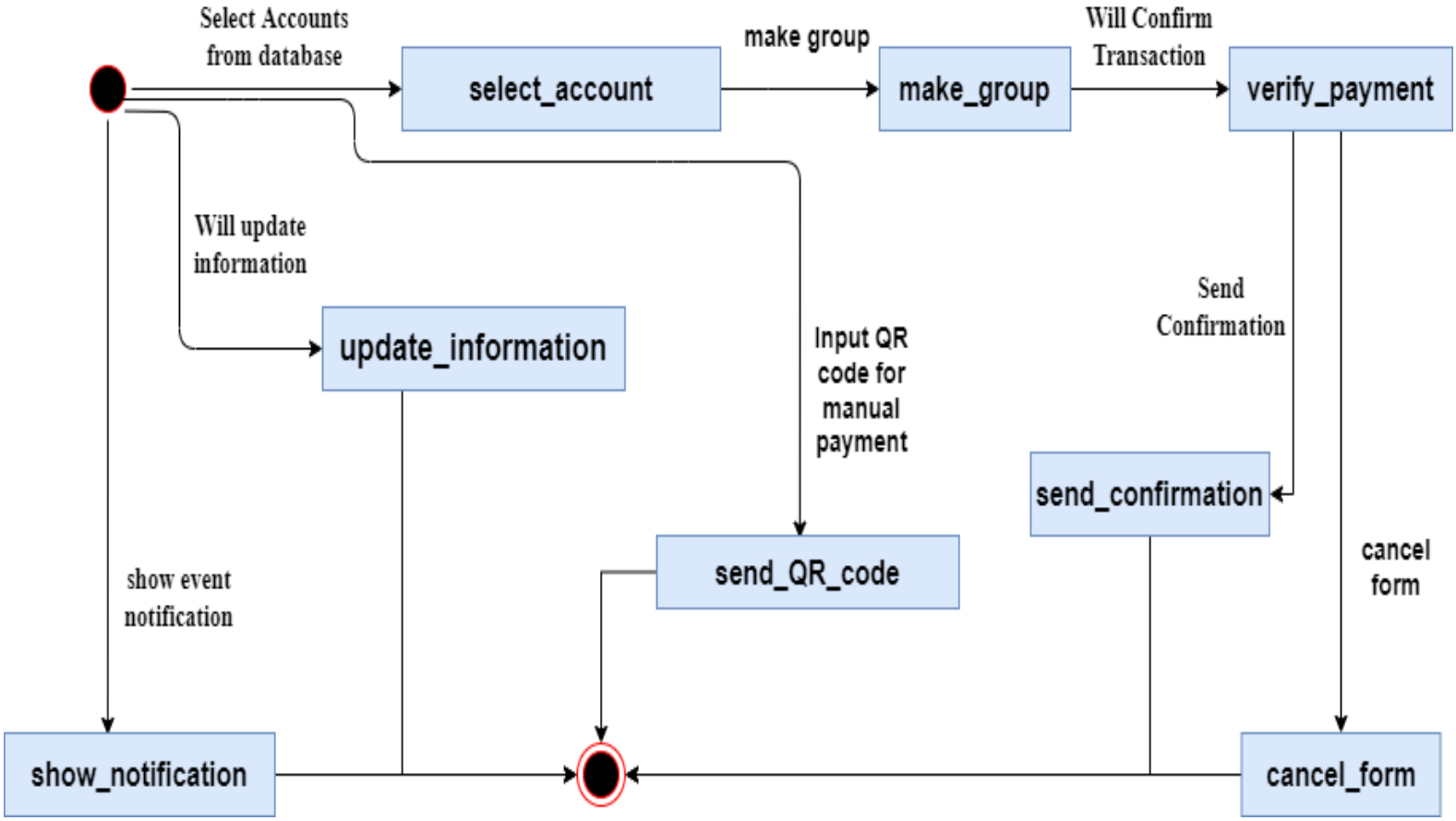
10.3.15. ID: 10

Name: Staff



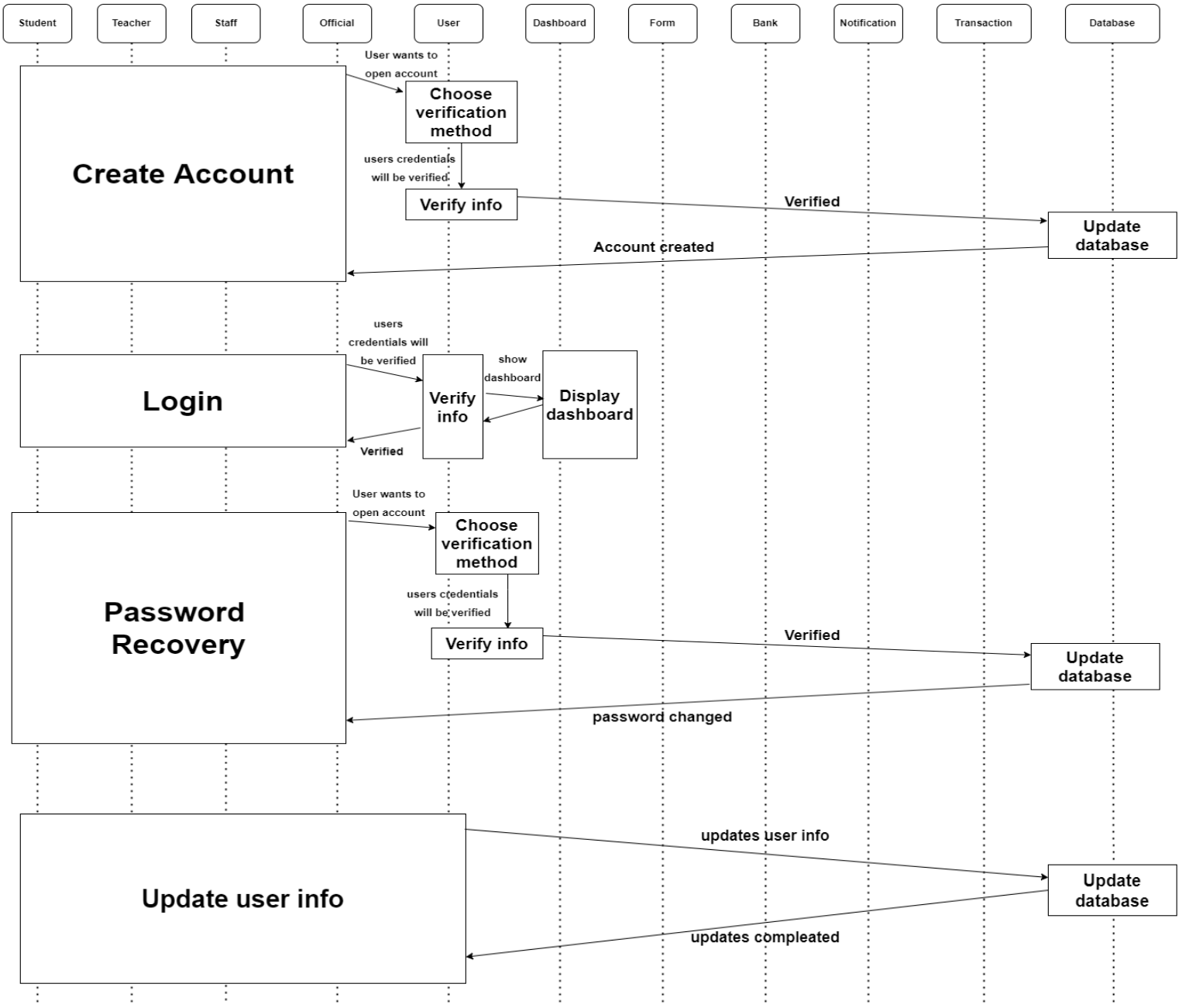
10.3.16 ID: 11

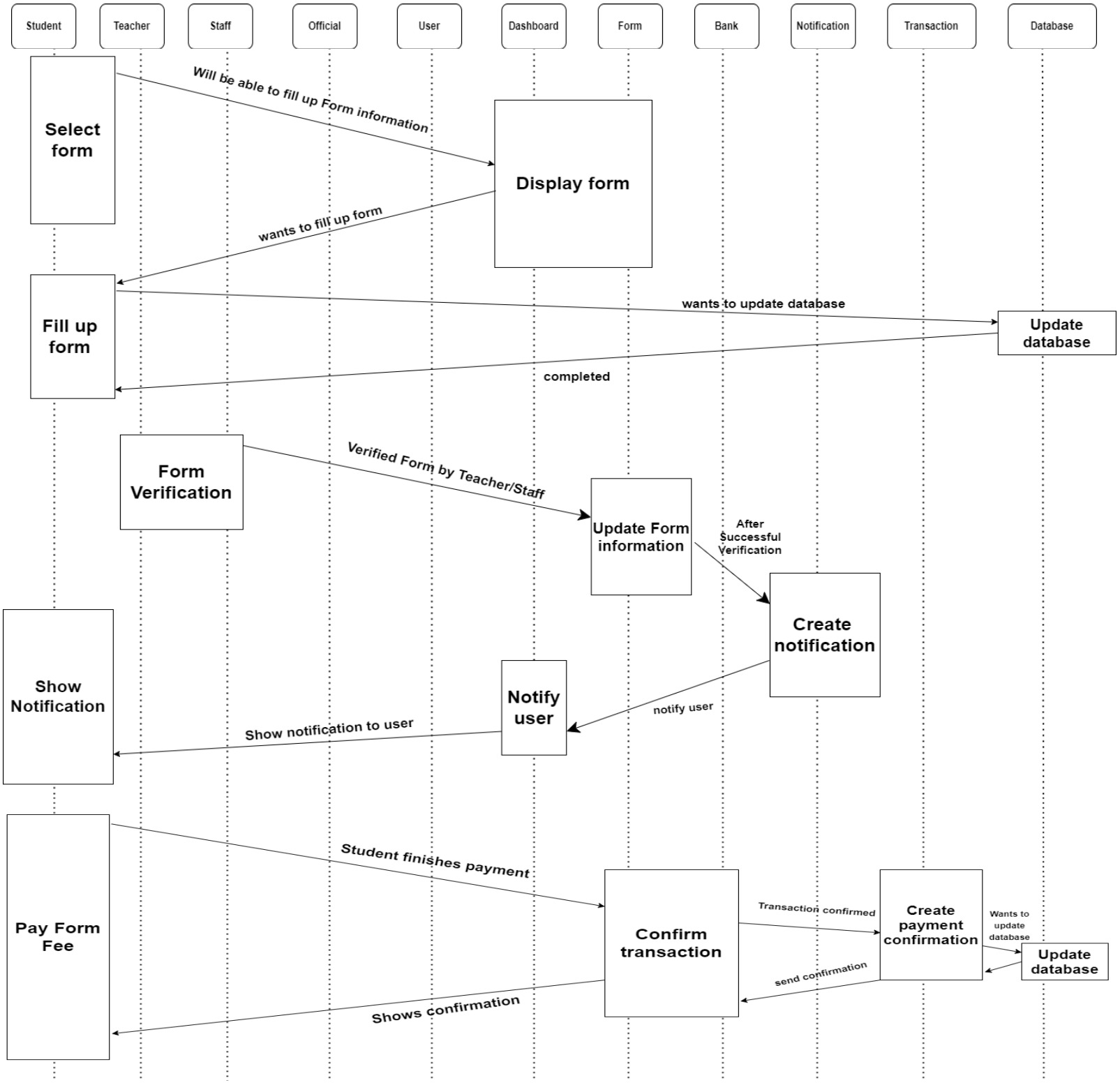
Name: Bank

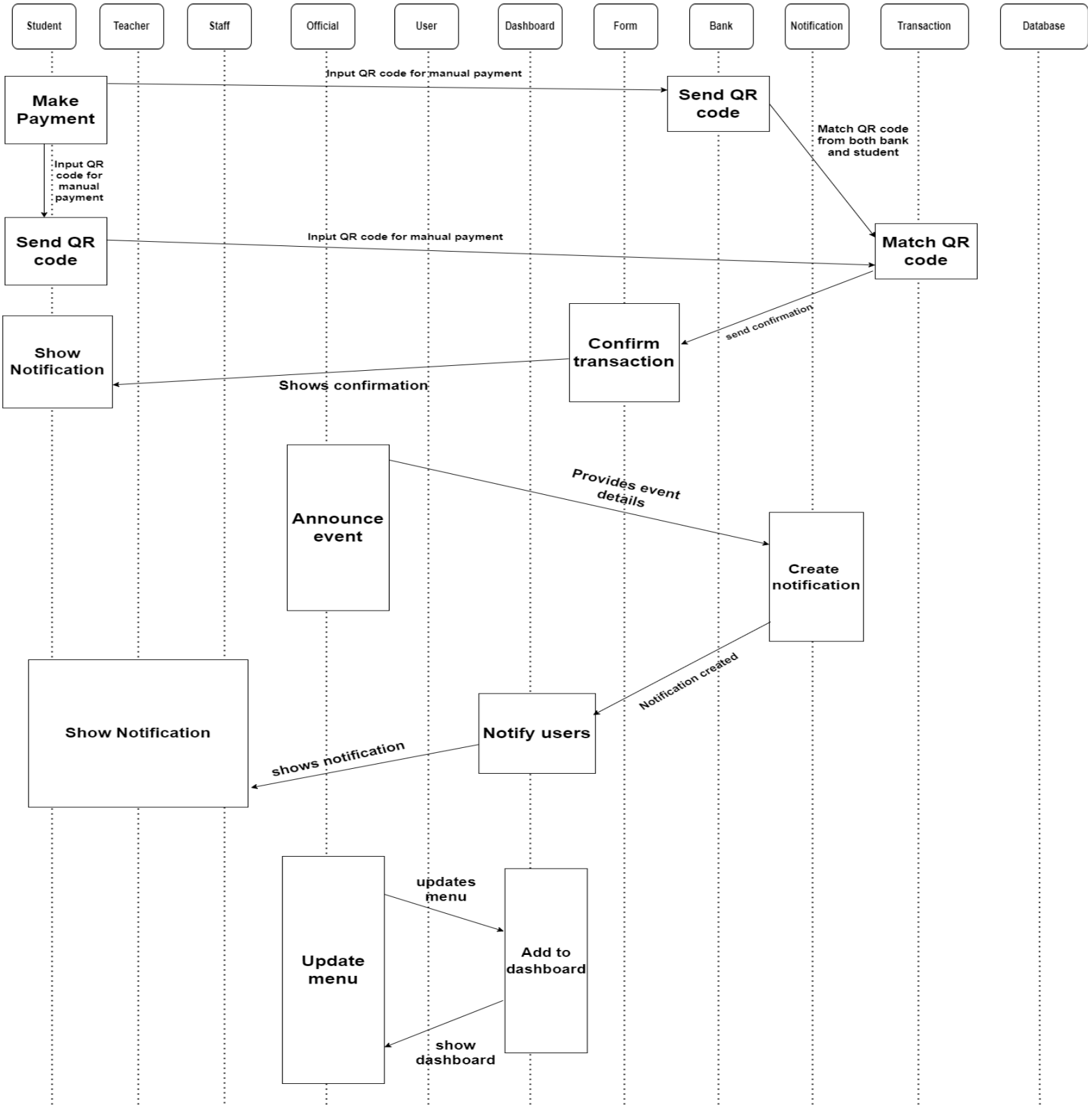


10.4. Sequence Diagram:

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction




Student
Teacher
Staff
Official
User
Dashboard
Form
Bank
Notification
Transaction
Database
Select form
Will be able to fill up Form information
Display form
wants to fill up form
Fill up form
wants to update database
Update database
completed
Form Verification
Verified Form by Teacher/Staff
Update Form information
After Successful Verification
Create notification
Show Notification
Notify user
notify user
Show notification to user
Pay Form Fee
Student finishes payment
Confirm transaction
Transaction confirmed
Create payment confirmation
Wants to update database
Update database
send confirmation
Shows confirmation


Student
Teacher
Staff
Official
User
Dashboard
Form
Bank
Notification
Transaction
Database
Make Payment
Input QR code for manual payment
Send QR code
Match QR code from both bank and student
Input QR code for manual payment
Send QR code
Input QR code for manual payment
Match QR code
send confirmation
Show Notification
Confirm transaction
Shows confirmation
Announce event
Provides event details
Create notification
Show Notification
Notify users
Notification created
shows notification
Update menu
updates menu
Add to dashboard
show dashboard