DU Services System

SE-801: Software Project Lab-3

**Submitted by**

Amran Hossain

BSSE 0917

**Supervised by**

Dr. Ahmedul Kabir

Assistant Professor

Institute of Information Technology

University of Dhaka

**Submitted To**

SPL-III co-ordinators



Institute of Information Technology

University of Dhaka

Submission Date:04.04.2021

Letter of Transmittal

Dr. Ahmedul Kabir

Assistant Professor,

Institute of Information Technology,

University of Dhaka.

Subject: Submission of term report on “DU Services System”.

Sir,

With due respect, I am submitting the report on the above topic you assigned me. In this report, I have given my best effort being some shortcomings. I earnestly hope that you would excuse my errors and oblige thereby.

Sincerely yours,

Amran Hossain

BSSE0917

Institute of Information Technology

University of Dhaka

Session: 2016-2017

Table of Contents

[1.1 Purpose 1](#_Toc68467144)

[1.2 Intended Audience 1](#_Toc68467145)

[1.3 Conclusion 1](#_Toc68467146)

[2.1 Introduction 2](#_Toc68467147)

[2.1.1 Identifying Stakeholders 2](#_Toc68467148)

[2.1.2 Recognizing multiple viewpoints: 3](#_Toc68467149)

[2.1.3 Working towards collaboration: 3](#_Toc68467150)

[2.1.4 Asking the first questions: 4](#_Toc68467151)

[2.2 Conclusion 5](#_Toc68467152)

[3.1 Introduction 6](#_Toc68467153)

[3.2 Eliciting requirements 6](#_Toc68467154)

[3.2.1 Collaborative Requirements Gathering 6](#_Toc68467155)

[3.2.2 Quality Function Deployment 6](#_Toc68467156)

[3.2.2.1 Normal Requirements 7](#_Toc68467157)

[3.2.2.2 Expected Requirements 7](#_Toc68467158)

[3.2.2.3 Exciting Requirements 7](#_Toc68467159)

[3.2.3 Usage Scenarios 7](#_Toc68467160)

[3.2.4 Elicitation Work Products 8](#_Toc68467161)

[3.3 Conclusion 8](#_Toc68467162)

[4.1 Introduction 9](#_Toc68467163)

[4.2 Use Case Definition 9](#_Toc68467164)

[4.3 Use Case Descriptions 10](#_Toc68467165)

[5.1 Data Modeling Concepts 27](#_Toc68467166)

[5.2 Data Objects 27](#_Toc68467167)

[5.3 ER diagram 30](#_Toc68467168)

[6.1 Introduction: 31](#_Toc68467169)

[6.2 Identifying Analysis Classes: 31](#_Toc68467170)

[6.2.1 Selection Criteria 31](#_Toc68467171)

[6.2.2 Associate Noun and Verb Identification 31](#_Toc68467172)

[6.2.3 Attribute Selection and Method Identification 32](#_Toc68467173)

[6.3 Class Responsibility Collaboration (CRC) 34](#_Toc68467174)

[6.4 UML Class Diagram 36](#_Toc68467175)

[7.1 Architectural Design for OOP 37](#_Toc68467176)

[7.2 Representing the system in a context 37](#_Toc68467177)

[7.3 Defining Architypes 38](#_Toc68467178)

[7.4 Refining the architype into Component 38](#_Toc68467179)

[7.4.1 Instantiations of the system 39](#_Toc68467180)

[8.1 Introduction 40](#_Toc68467181)

[8.2 Transform Mapping 40](#_Toc68467182)

[9.1 Introduction 43](#_Toc68467183)

[9.2 Analysis class 43](#_Toc68467184)

[9.3 Elaboration of Design Components 43](#_Toc68467185)

[9.3.1 Student 44](#_Toc68467186)

[9.3.2 Transaction 45](#_Toc68467187)

[9.3.3 Document 46](#_Toc68467188)

[9.3.4 Authentication 47](#_Toc68467189)

[9.3.5 Department 48](#_Toc68467190)

[9.3.6 Admin 49](#_Toc68467191)

[9.4 Deployment Diagram 50](#_Toc68467192)

[10.1 Interface Analysis 51](#_Toc68467193)

[10.1.1 User analysis 51](#_Toc68467194)

[10.1.1.1 Identify User 51](#_Toc68467195)

[10.1.1.2 Know User 51](#_Toc68467196)

[10.1.2 Task analysis 52](#_Toc68467197)

[10.2 Interface Design Steps 53](#_Toc68467198)

[11.1 Introduction 62](#_Toc68467199)

[11.2 Software Testing Goals 62](#_Toc68467200)

[11.3 Levels of Testing 62](#_Toc68467201)

[11.4 Testing Procedure 63](#_Toc68467202)

[11.5 Risk and Contingencies 64](#_Toc68467203)

[12.1 Application run 65](#_Toc68467204)

[12.2 Use the Application 65](#_Toc68467205)

[13.1 Conclusion: 66](#_Toc68467206)

[13.2 Future Work: 66](#_Toc68467207)

List of Figures

Figure 1:Level 0 DU Services System 11

Figure 2: Level 1 -Use Case Diagram 12

Figure 3:Level 1.1-Authentication 13

Figure 4:Level 1.2 – Certificate Management 14

Figure 5:Level 1.3- Testimonial Management 15

Figure 6: Level 1.5 – Notification process 16

Figure 7: Level 1.6 - Search 17

Figure 8: Level 2.1- Applicant’s enrollment form 18

Figure 9: Level 2.2- Login 19

Figure 10: Level 2.3- Recovery account 20

Figure 11: Level 2.4- application for testimonial/certificate 21

Figure 12: Level 2.5 - Transaction 22

Figure 13: Level 2.6- Application service status 23

Figure 14: Level 2.7– Department Search 24

Figure 15: Level 2.8- Admin Search 25

Figure 16: Level 2.9– Applicant Search 26

Figure 17: ER diagram 30

Figure 18: UML class diagram 36

Figure 19:Architectural Context Diagram 38

Figure 20: Archetype Design 38

Figure 21: Instantiation 39

Figure 22: Context level DFD 41

Figure 23: Level 1 mapping with DFD 41

Figure 24: Level 2 mapping with DFD 42

Figure 25: Analyses class for student component 44

Figure 26: Analyses class for Transaction 45

Figure 27: Analysis class for document 46

Figure 28: Analyses class for authentication 47

Figure 29:Analyses class for department 48

Figure 30: Analysis class for admin 49

Figure 31: Deployment diagram 50

Figure 32: Home page 54

Figure 33: Application form 54

Figure 34: Search status or login 55

Figure 35:User profile 55

Figure 36: payment process 56

Figure 37: Transactional process 57

Figure 38: notification process 57

Figure 39:admin login 58

Figure 40: recover account 58

Figure 41:department list 59

Figure 42: Edit 59

Figure 43: Delete 60

Figure 44: Sidebar admin 60

Figure 45: Certificate of Completion 61

Figure 46: Testimonial process 61

Acknowledgement

I am highly indebted for getting such a tremendous opportunity to prepare this report. I would like to thank my first supervisor, Md. Saeed Siddik, Assistant Professor, Institute of Information Technology, University of Dhaka. He helped me a lot throughout this project. He gave me a proper guideline about how I can prepare this report and complete the project. Also, I would like to thank my current supervisor, Dr. Ahmedul Kabir, Assistant Professor, Institute of Information Technology, University of Dhaka for helping me a lot in this project. In completing this report, I have collected various important data and information from the authority and my supervisor. I am thankful to all of the works cited.

Abstract

This study is made for Online Services Management System. The scope of the study is to analyze the web-based service system and to know the functions and drawbacks and component level design, interface design, deployment diagram and user manual of the system. In short, Applicant applies for the testimonial or certificate with giving personal details accurately. Then transact the money with payment module integrated in this system. Admin may accept or reject the application and build testimonial or certificate for the applicants. The status will be known through the email. Also, applicants may log in the system and see the status of application. Then applicants get the certificate or testimonial.

The object of the study is to develop a system of any kind of Services System of DU.

Chapter 1: Introduction

This chapter is a part of the software requirement specification for the project “DU Services System”. In this chapter intended audience for the project are focused on.

# 1.1 Purpose

This document briefly describes the Software Requirement Analysis and user manual of DU Services system. It contains functional, non-functional, and supporting requirements and establishes a requirement baseline for the development of the system. The requirements specified in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as an official mean of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

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

# 1.3 Conclusion

This analysis of the audience helped us to focus on the users who will be using our analysis. This  
overall document will help each and every person related to this project to have a better idea  
about the project

Chapter 2: Inception

In this chapter, the inception part of the SRS will be described briefly.

# 2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team. To establish the ground work, the following factors have been worked onto the inception phase:

* List of stakeholders
* Recognizing multiple viewpoints
* Working towards collaboration
* Requirement questionnaire

## 2.1.1 Identifying Stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that might be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stake holders are contacted because every stakeholder will be asked: “whom else do you think I should talk to?”  
The following stakeholders were identified for the “DU Services System”.

**Student:** Students are directly affected to the system. They can apply for the testimonial or language certificate through the system. Student can get notification for his readymade testimonial or certificate through the system.

**Admin:** There are two types of admin in this system. One is super admin another is normal admin. Super-admin can control the system and given allowance to the admin’s access. Normal admin will be department centric. Admin can accept or reject the application and given notification to the student.

**Software Developer:** A software developer is concerned with faces the software development  
process, including the research, design, programming, maintenance and testing of computer software. He will be responsible for the outcomes of the software.

**Merchant Service Provider**: This system has included payment process through the bKash/Nagad. So, they are also the stakeholder of this system.

## 2.1.2 Recognizing multiple viewpoints:

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

**Student’s viewpoint:**

* User friendly
* Error free system
* Application process easily
* Transaction process should be easy
* Knowing status of application
* Searching status of application
* Keeping a backup of data

**Admin’s viewpoint:**

* User friendly
* Error free system
* Strongly authentication system
* Defining rules of admin
* Searching the applicant’s name and department’s name
* Keeping a backup of data
* Easy to use
* Approving the application
* Adding the department and admin (super-admin only)

**Developer’s viewpoint:**

* Easy to built
* Error free effective software
* No ambiguous requirement
* Getting a decent amount of money for project budget

## 2.1.3 Working towards collaboration:

While working with different stakeholders, some conflicting and common viewpoints can be noticed. For this reason, final requirements can be gotten by collaborating the viewpoints.

**Common viewpoints**

* Error free effective system
* User friendly
* Easy to maintain the software
* Strong authentication system

**Conflicting viewpoints**

* Final requirements
* Error free effective system
* Easy to maintain the software
* Strong authentication system
* Notifying the applications
* Payment system smooth
* Keeping a data backup

**Final requirements**

* Final requirements
* Error free effective system
* Easy to maintain the software
* Strong authentication system
* Notify the applicants
* Keeping data back up
* Admin rule fixed
* Payment gateway system

## 2.1.4 Asking the first questions:

At first some context free questions were asked for identifying the stakeholders. Context free questions are helpful to identifying some stakeholders who cannot be identified by structural questions. Overall project benefits and goals are –

* Who is subsidizing for the project?
* Who will be benefited from the project?
* Who gets to make the final verdict about the project (if this is different  
  from the money source)?
* Who has resources which I need in order to get the project done?
* Whose work will my project have an impact on? (During the project  
  and also once the project is completed).

# 2.2 Conclusion

In this project, a demonstration is made for 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 anticipation of the project. Hopes are high that the full  
functioning document will help us to define that future expectations.

Chapter 3: Elicitation

This chapter specifies the Elicitation phase.

# 3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering  
requirements from the users, customers and other stakeholders. Many difficulties were faced,  
like understanding the problems, making questions for the stakeholders, limited communication  
with the stakeholders due to a short amount of time and volatility. Though it is not easy to gather  
requirements within a very short time, these problems have been surpassed in an organized and  
systematic manner.

3.2 Eliciting requirements

The main task of this phase is to combine the elements of problem solving, elaboration,  
negotiation and specification. The collaborative working approach of the stakeholders is required  
to elicit the requirements. The following tasks were done for eliciting requirements-

* Collaborative Requirements Gathering
* Quality Function Deployment
* Usage Scenarios
* Elicitation work products

## 3.2.1 Collaborative Requirements Gathering

The meetings with the stakeholders created an indecisive state to elicitate requirements. To solve  
this problem, more than one meeting was held with the stakeholders. A slightly different scenario from these approaches has been found following activities have been completed to accomplish this task. The meetings were conducted with the Admin of the Department, Students and Supervisors. They were questioned about their requirements and expectations from the Services System. They were asked about the problems they were facing with the current manual process of getting those services.  
Lastly final requirement list was selected from the meetings.

## 3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. Ultimately the goal of QFD is to translate subjective quality criteria into objective ones that can be quantified and measured and which can be used to design and manufacture the product. It is a methodology that concentrates on maximizing customer satisfaction from the software engineering process. The requirements, which are given below, are identified successfully by the QFD.

### 3.2.2.1 Normal Requirements

The normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers’ satisfaction. These are the normal requirements for the project.

* Applications form for the applicant both testimonial and language certificates.
* Admin login
* Admin logout
* Applicant profile view
* Admin and applicant list view & edit
* Different page for both testimonial and language certificate
* Seeing the status of the applicant.
* Add, Edit, Delete of departments.
* Add, Edit, Delete of different kind of services.

### 3.2.2.2 Expected Requirements

These requirements are intrinsic to the product or system and may be so elementary that the  
customer does not explicitly state them. Their absence will be a cause for significant  
dissatisfaction. Below the expected requirements for our project are briefly described.

* Admin will be remained from each department. admin can accept or reject the application
* Testimonial processing
* Language certificate processing
* Notify the applicants
* Application’s status
* Demo payment gateway for transaction
* Effective System
* Unique application for each applicant

### 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. Following are some exciting requirements of this project.

* Real time transaction system
* Super admin and normal admin role. Super admin control whole the system and given role to normal admin to control the applications process.
* Real feature for handling the system

## 3.2.3 Usage Scenarios

Getting a testimonial or Language Certificate is a lengthy process in our University. Manually student write an application, pay money to bank and then they get a testimonial or certificate. They are often needed in those services but feeling bored in this manual process. It is a time-consuming process. The web system "DU Services System” can be the solution of the difficulties faced by the students and administration in our campus.

In this system, admin will enter the system giving email and password. Super admin and admin can see the application list, accept or reject the application, edit the application’s information. After approving student’s application, they receive notification through email. Admin also add departments of this panel or edit this. Super admin can add normal admin or delete it but normal admin can’t access this functionality.

Applicant/student can choose which services he/she needs. Then he/she enter into the application by giving his/her Personal details into the Application form. Students should give his Name, Father’s Name, Mother’s Name, University Registration Number, Date of Birth, Phone, Session, Department, Current year for the application. After login with id and email they can see their personal information and status. Whether their application accept or not. After application submitting, they can transact the money with bKash or other services. After acceptance notification they can download or collect the testimonial or certificate.

## 3.2.4 Elicitation Work Products

The work products of the requirement elicitation can be different depending on the size of the system or product to be build. The work products include:

* A statement of requirements for our system.
* A bounded statement of scope for our system.
* A list of customer, users, and other stakeholders who participated in requirement elicitation.
* A detailed description of the system’s technical environment.
* A set of usage scenarios.

Each of these work products are verified by all people who have participated  
in requirements elicitation.

# 3.3 Conclusion

All the necessary requirements to fulfill our project scope were identified. A generated scenario of the project has been developed. Based on this scenario, a scenario-based model is presented in the next chapter.

Chapter 4 – Scenario Based Model

This chapter describes the scenario-based model for the DU Services System.

# 4.1 Introduction

Although the success of an android-based system or product is measured in many ways, user satisfaction resides at the top of the list. If the software developer team understands how end users (and other actors) want to interact with a system, they will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

# 4.2 Use Case Definition

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 describe satisfactory using corresponding actors who perform important roles in the story and make 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 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.

**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.

Table 1:Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| Level 0 | Level 1 | Level 2 | Actor |
| DU Services System | Authentication | Applicant’s enrollment/sign up | Student, Admin,  Database |
| Sign in | Admin, Database |
| Sign out | Admin, Database |
| Recovery Account | Admin, Database |
| Admin’s role | Super admin, normal admin |
| Certificate Management | Certificate application | Student, Admin |
| Transaction money with bKash | Student, Database |
| Approve the application | Admin |
| Cancel the application | Admin |
| Download certificate | Student, Admin |
| View the certificate | Student, Admin |
| Testimonial Management | Testimonial application | Student, Admin |
| Transaction money with bKash | Student, Admin |
| Approve the application | Admin |
| Download testimonial | Student, Admin |
| Cancel the application | Admin |
| View the testimonial | Student, Admin |
| Notification System | Reject notification | Student, Admin |
| Completion notification | Student, Admin |
| Search | Search the applicant’s | Admin |
| Search the admin’s | Super admin |
| Search the department | Admin |

# 4.3 Use Case Descriptions

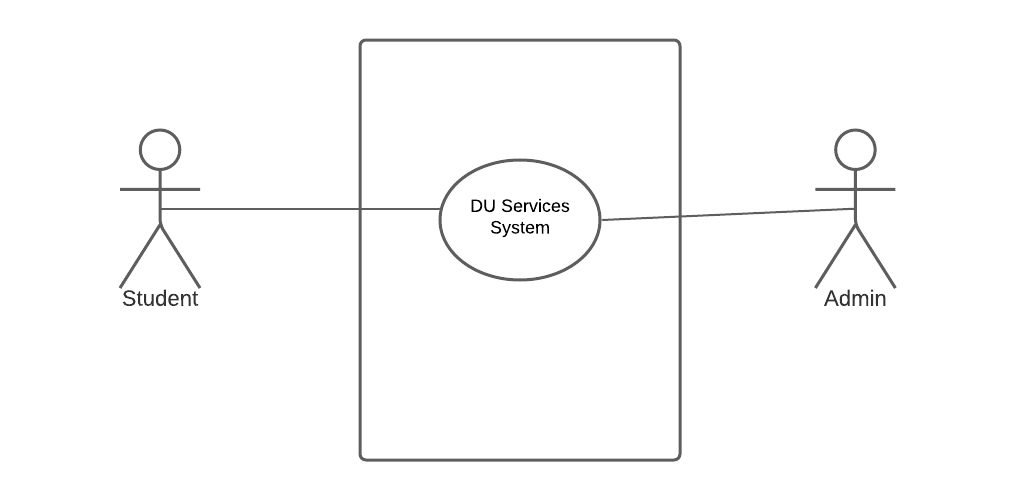


Figure 1:Level 0 DU Services System

**Use Case**: DU Services System

**Use Case ID**: 0

**Primary Actors**: Student, Admin

**Secondary Actor**:

**Goal in Context:** To get a service

**Preconditions**: User must have a web browser and internet connection.

**Trigger**: User decides to run the system to create an account.

**Scenario**: User run the system to give the url into browser.

**Exception**: null

**Priority**: Essential.

**When available**: First Increment.

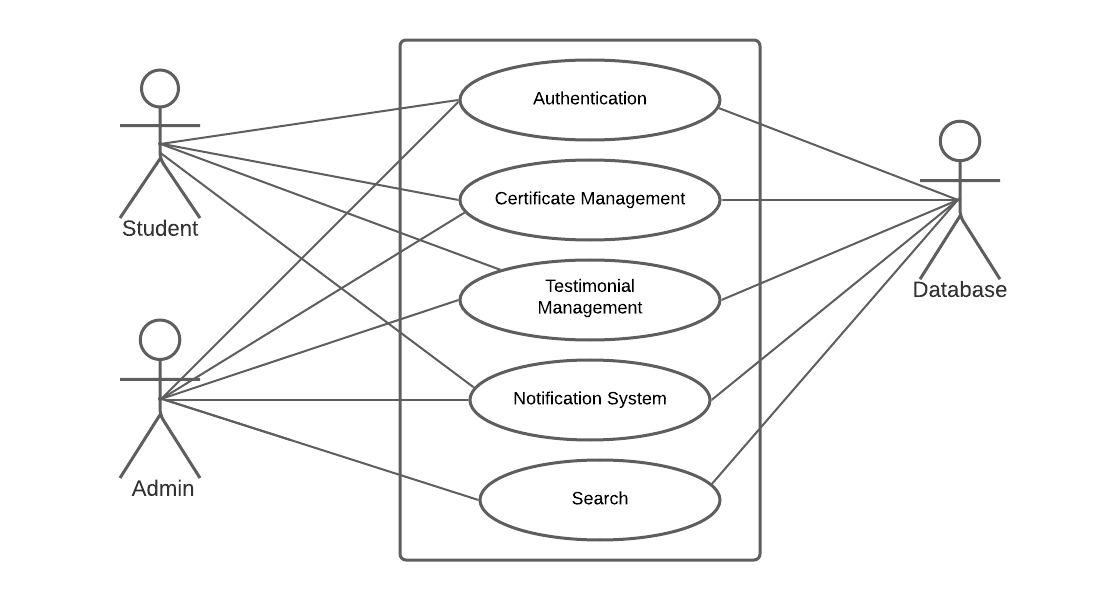


Figure 2: Level 1 -Use Case Diagram

**Use Case**: DU Services System

**Use Case ID**: 0

**Primary Actors**: Student, Admin

**Secondary Actor**: Database

**Goal in Context:** To get all the services from the system.

**Preconditions**: User must have a web browser and internet connection.

**Trigger**: User decides to run the system to create an account.

**Scenario**: User run the system to give the url into browser.

**Exception**: null

**Priority**: Essential.

**When available**: First Increment

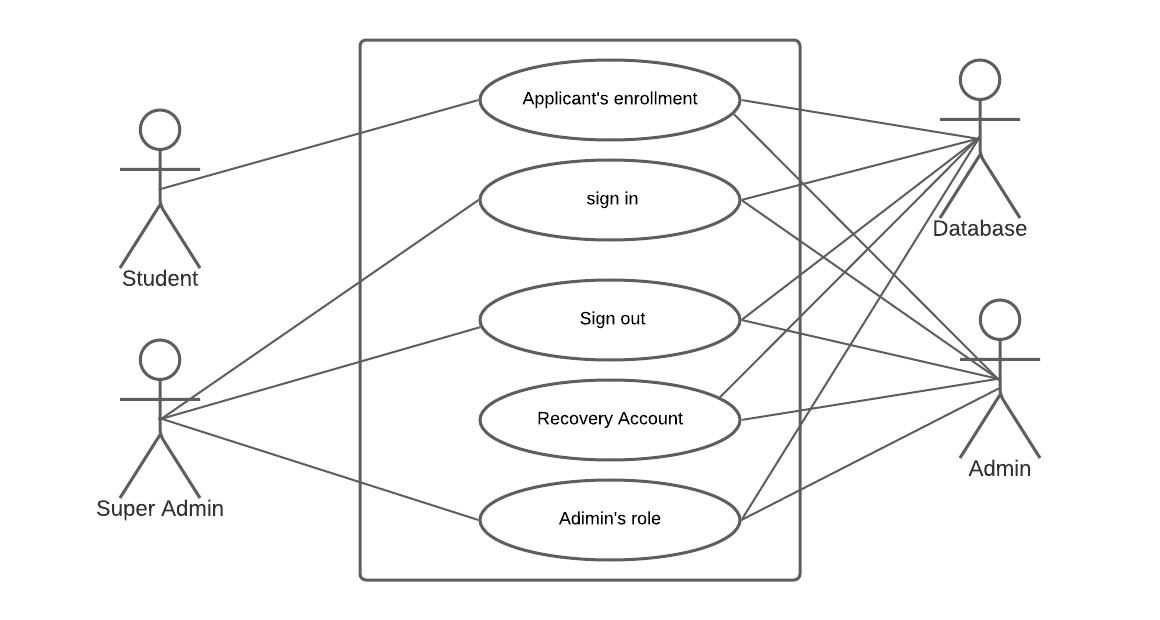


Figure 3:Level 1.1-Authentication

**Use Case**: Authentication

**Use Case ID**: 0

**Primary Actors**: Student, Super Admin

**Secondary Actor**: Normal Admin

**Goal in Context:** Authenticate to the user.

**Preconditions**: System has been designed for the authentication and user must have a web browser.

**Trigger**: User decides to run the system to create an account.

**Scenario**: User run the system to give the url into browser.

**Exception**: Internet Connection failure or input wrong data.

**Priority**: Essential.

**When available**: First Increment

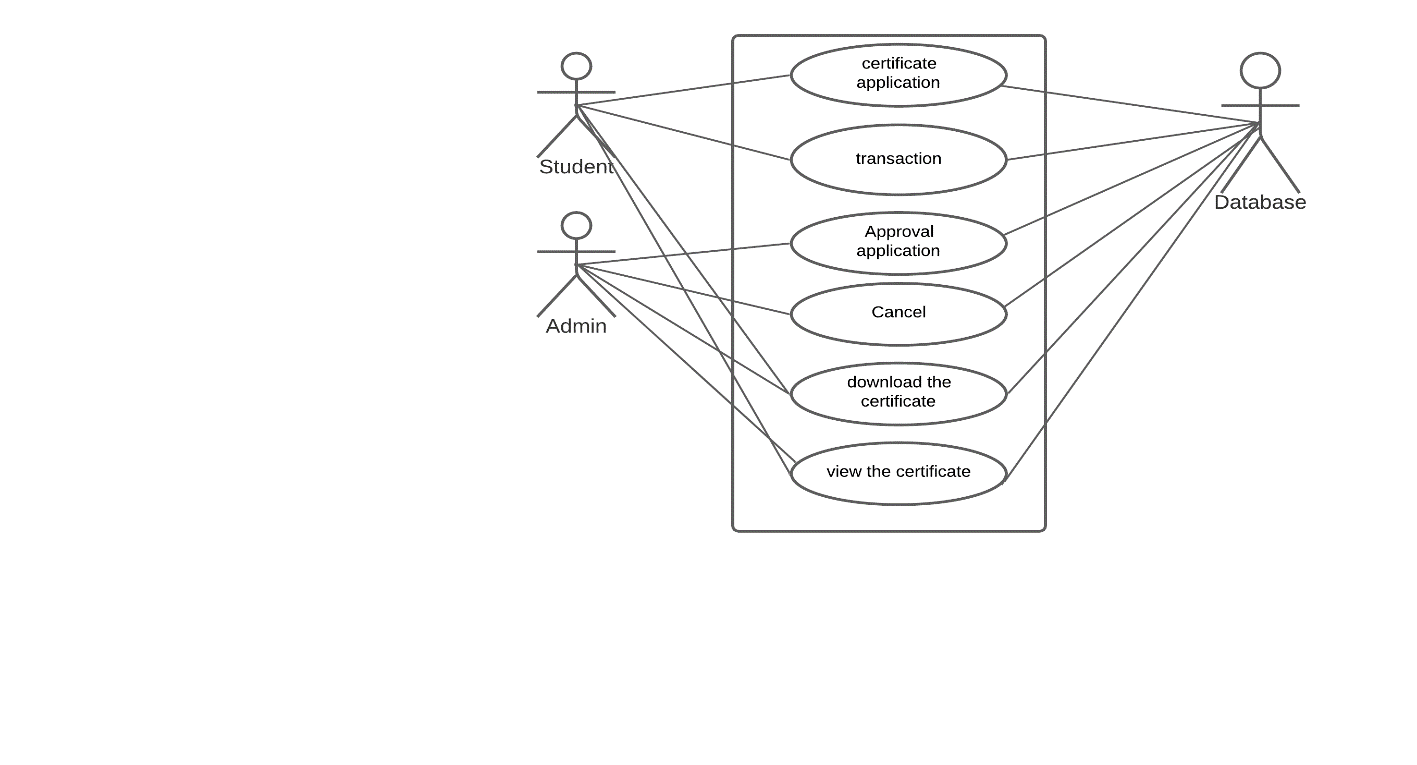


Figure 4:Level 1.2 – Certificate Management

**Use Case**: Certificate Management

**Use Case ID**: 0

**Primary Actors**: Student, Admin

**Secondary Actor**: Database

**Goal in Context:** To complete certificate from application to get.

**Preconditions**: User must have a web browser and internet connection.

**Trigger**: User decides to run the system to create an account.

**Scenario**: User run the system to give the url into browser.

**Exception**: null

**Priority**: Essential.

**When available**: First Increment

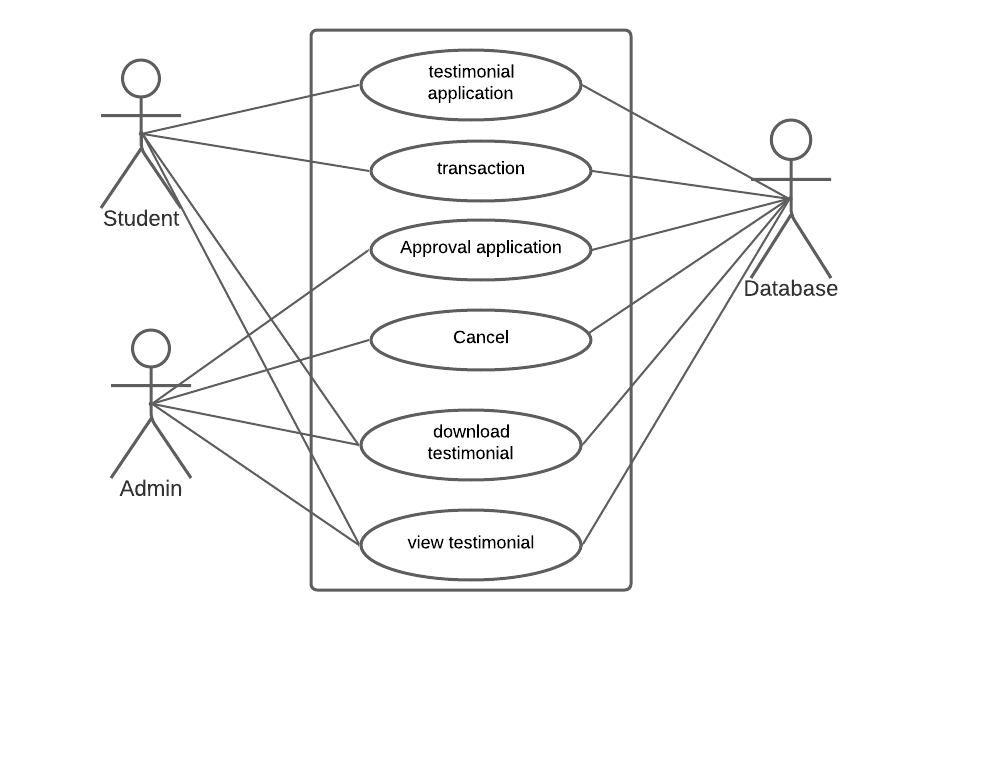


Figure 5:Level 1.3- Testimonial Management

**Use Case**: Testimonial Management

**Use Case ID**: 0

**Primary Actors**: Student, Admin

**Secondary Actor**: Database

**Goal in Context:** To get testimonial.

**Preconditions**: User must have a web browser and internet connection and have an application for getting this service.

**Trigger**: User decides to run the system to create an account.

**Scenario**: User run the system to give the url into browser.

**Exception**: null

**Priority**: Essential.

**When available**: First Increment

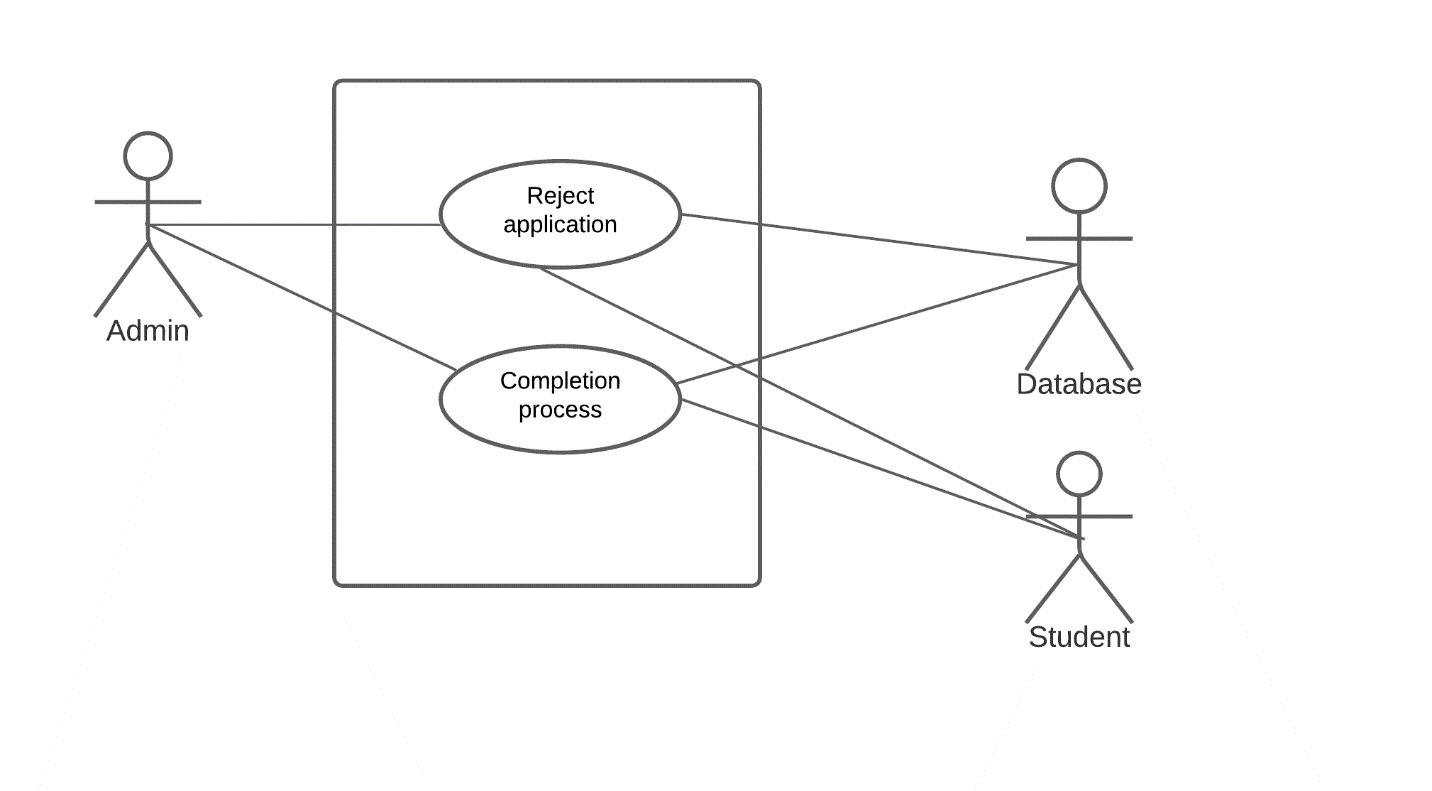


Figure 6: Level 1.5 – Notification process

**Use Case**: Notification process

**Use Case ID**: 0

**Primary Actors**: Admin

**Secondary Actor**: Student, Database

**Goal in Context:** To get notification after document ready.

**Preconditions**: User must have a web browser and internet connection. Document should be ready and admin accept the application.

**Trigger**: Student gets notification through email.

**Scenario**: Student apply for the testimonial and certificate admin accept or reject it and then get a notification.

**Exception**: If user fail to application then they might be fail to get a notification.

**Priority**: Essential.

**When available**: First Increment

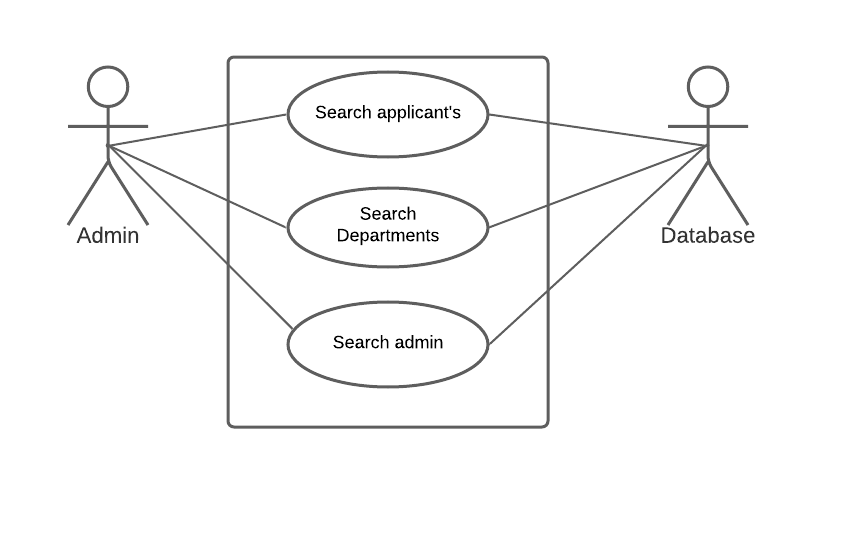


Figure 7: Level 1.6 - Search

**Use Case**: Search

**Use Case ID**: 0

**Primary Actors**: Admin

**Secondary Actor**: Database

**Goal in Context:** To get a specific department or applicants.

**Preconditions**: User must have admin and searching results should be in list.

**Trigger**: Admin decides to run the system to create an account.

**Scenario**: Admin enter the system giving information. Then search the specific component from specific phase. To search admin, Admin must be Super Admin.

**Exception**: Giving wrong search key.

**Priority**: Essential.

**When available**: First Increment

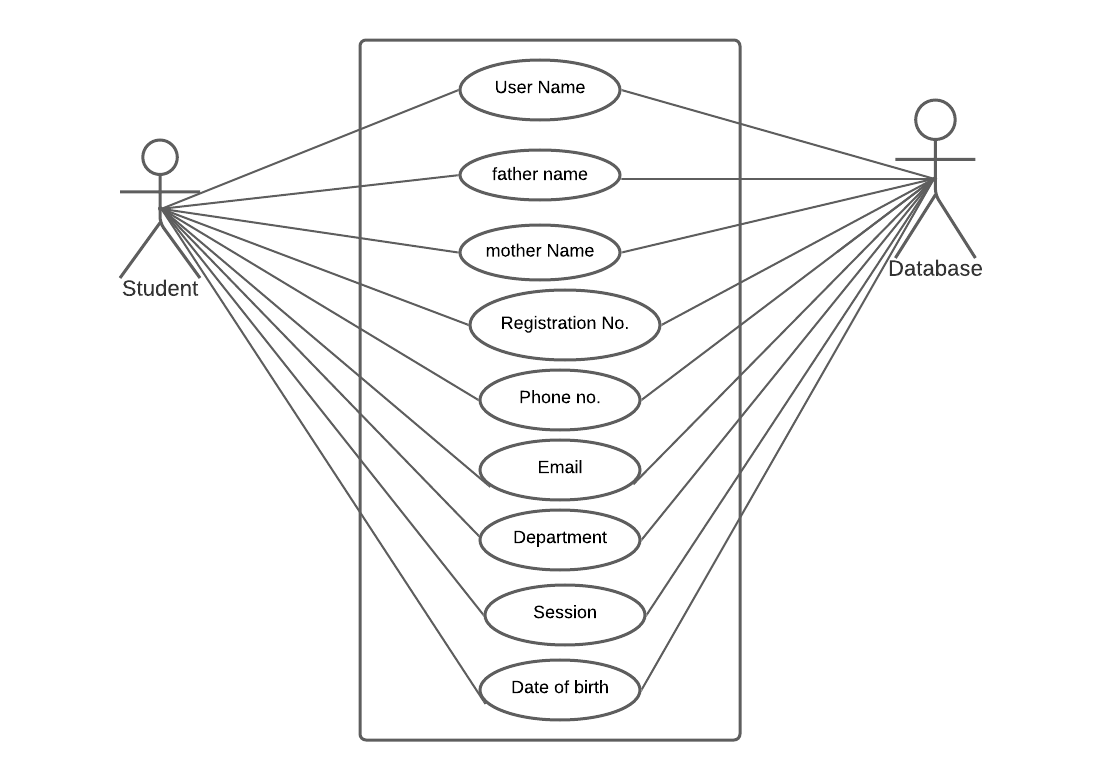


Figure 8: Level 2.1- Applicant’s enrollment form

**Use Case**: Applicant’s enrollment

**Use Case ID**: 0

**Primary Actors**: Student

**Secondary Actor**: Database

**Goal in Context:** Apply for a testimonial or certificate to create an account

**Preconditions**: User must have a web browser and internet connection. System has been designed for opening an account.

**Trigger**: Student must have an account.

**Scenario**: User run the system to give the url into browser. Then choose the services and click on get testimonial/certificate. Then enter data into the application form.

**Exception**: Ambiguous input. Data not matched.

**Priority**: Essential. Must be implemented.

**When available**: First Increment

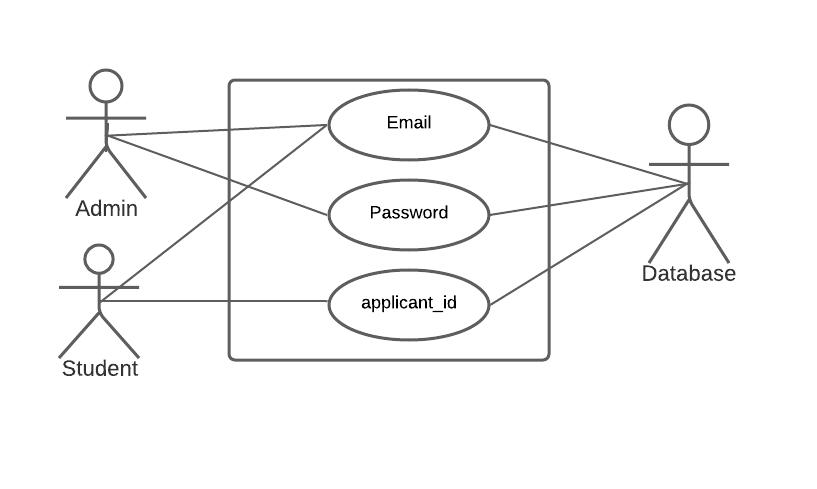


Figure 9: Level 2.2- Login

**Use Case**: Sign in

**Use Case ID**: 0

**Primary Actors**: Student, Admin

**Secondary Actor**: Database

**Goal in Context:** To sign in the system.

**Preconditions**: User must have a web browser and internet connection. System has been designed for signing in an account

**Trigger**: User have to sign in an account.

**Scenario**: User run the system to give the url into browser. Click application status or sign in page and given email, password and applicant\_id for sign in.

**Exception**: Email or password or applicant\_id wrong.

**Priority**: Essential. Must be implemented.

**When available**: First Increment

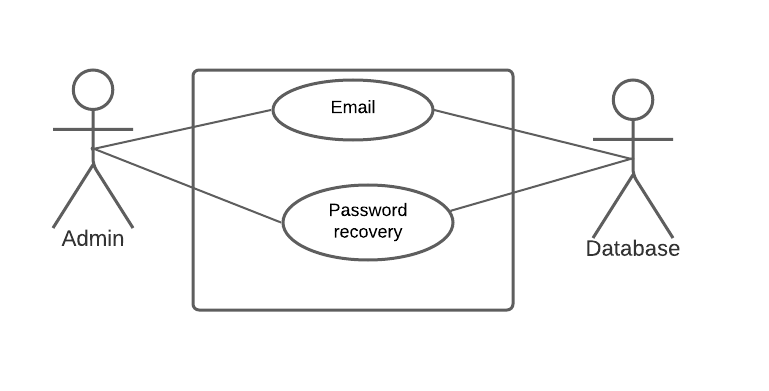


Figure 10: Level 2.3- Recovery account

**Use Case**: Recovery account

**Use Case ID**: 0

**Primary Actors**: Admin

**Secondary Actor**: Database.

**Goal in Context:** To recover an account

**Preconditions**: User must have a web browser and internet connection. System has been designed for recover an account.

**Trigger**: admin can recover their account.

**Scenario**: Recover account. Providing email address. Link will be sent on email.

**Exception**: Ambiguous input

**Priority**: Essential.

**When available**: First Increment.

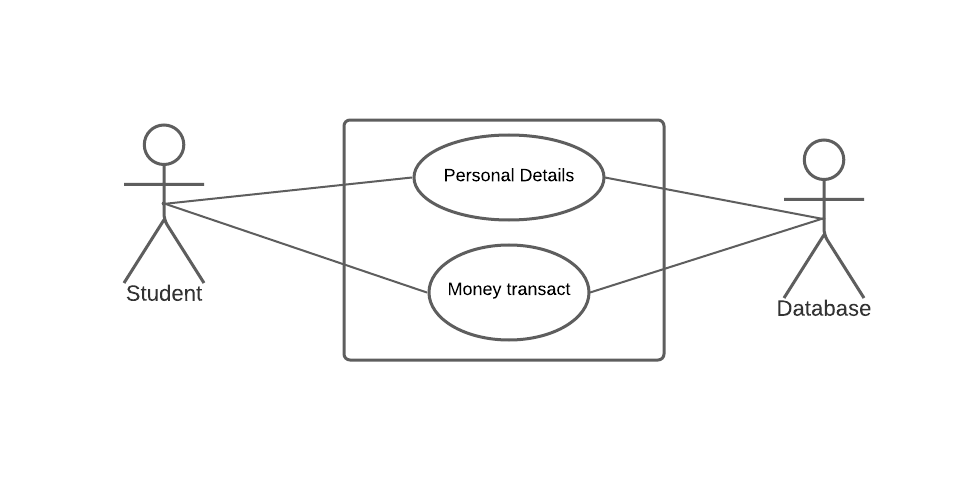


Figure 11: Level 2.4- application for testimonial/certificate

**Use Case**: application for document

**Use Case ID**: 0

**Primary Actors**: Student

**Secondary Actor**: Database

**Goal in Context:** Successful application with money transaction.

**Preconditions**: User must have a web browser and internet connection. Application complete.

**Trigger**: Student decides to run the system with giving personal details and run the system.

**Scenario**: User run the system to give the url into browser. After application go to transaction page and transaction complete.

**Exception**: Internet connection broken.

**Priority**: Essential.

**When available**: First Increment.

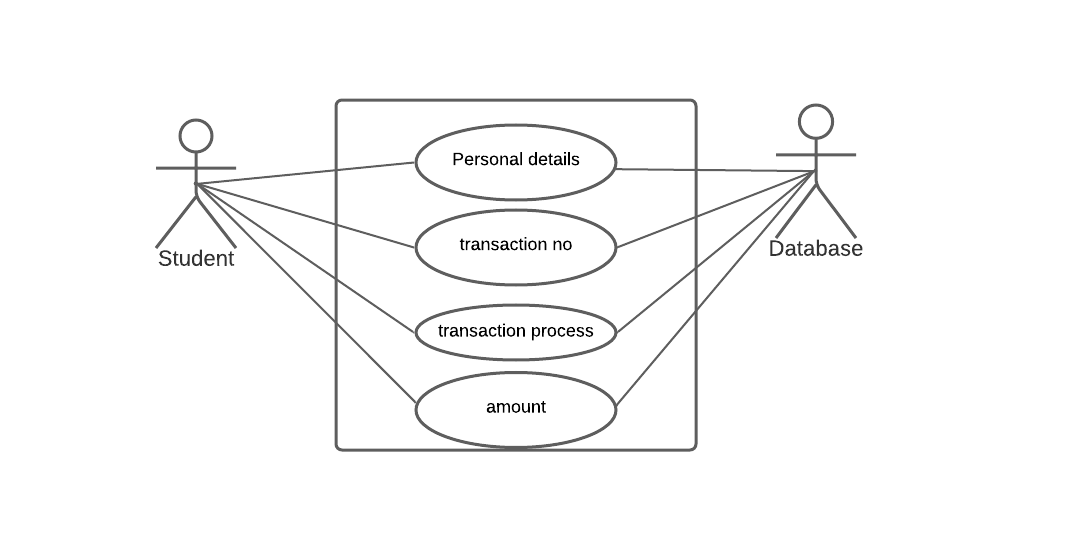


Figure 12: Level 2.5 - Transaction

**Use Case**: Transaction

**Use Case ID**: 0

**Primary Actors**: Student

**Secondary Actor**: Database

**Goal in Context:** Successful money transaction.

**Preconditions**: User must have a web browser and internet connection. Application complete. Transaction system build up.

**Trigger**: Student decides to complete the application after transaction.

**Scenario**: User run the system to give the url into browser. After application go to transaction page and giving accurately transaction no, amount etc.

**Exception**: Internet connection broken. Transaction account is wrong. Service not available.

**Priority**: Essential.

**When available**: First Increment.

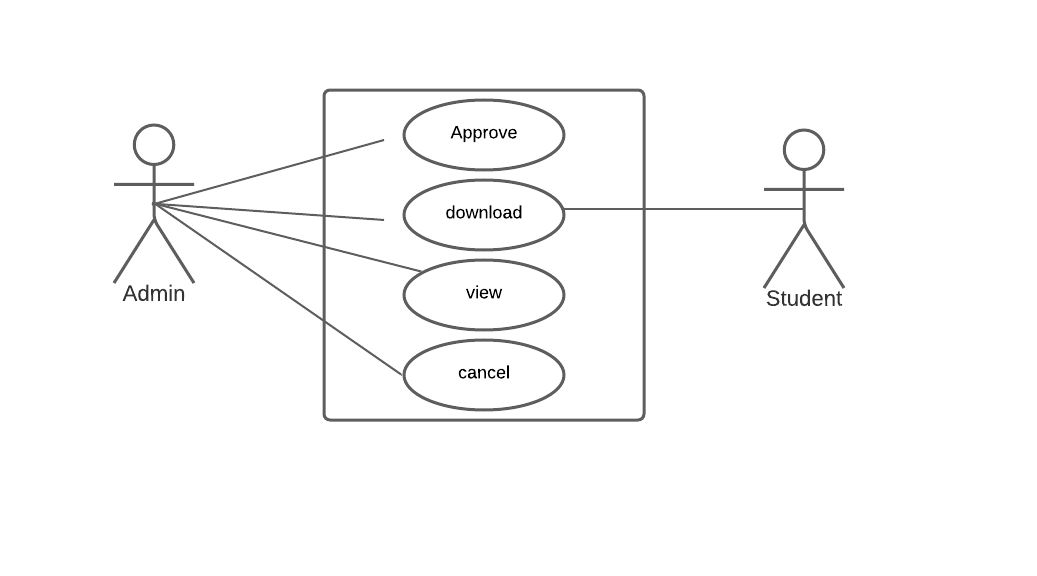


Figure 13: Level 2.6- Application service status

**Use Case**: Application service status

**Use Case ID**: 0

**Primary Actors**: Admin

**Secondary Actor**: Student

**Goal in Context:** Complete the application services.

**Preconditions**: User must have a web browser and internet connection. Application complete. Admin must be logged in.

**Trigger**: Admin can view, approve, download the application and build up document.

**Scenario**: User run the system to give the url into browser. Admin can approve/reject the application.

**Exception**: Internet connection broken. Application not successful. Document not ready.

**Priority**: Essential.

**When available**: First Increment.

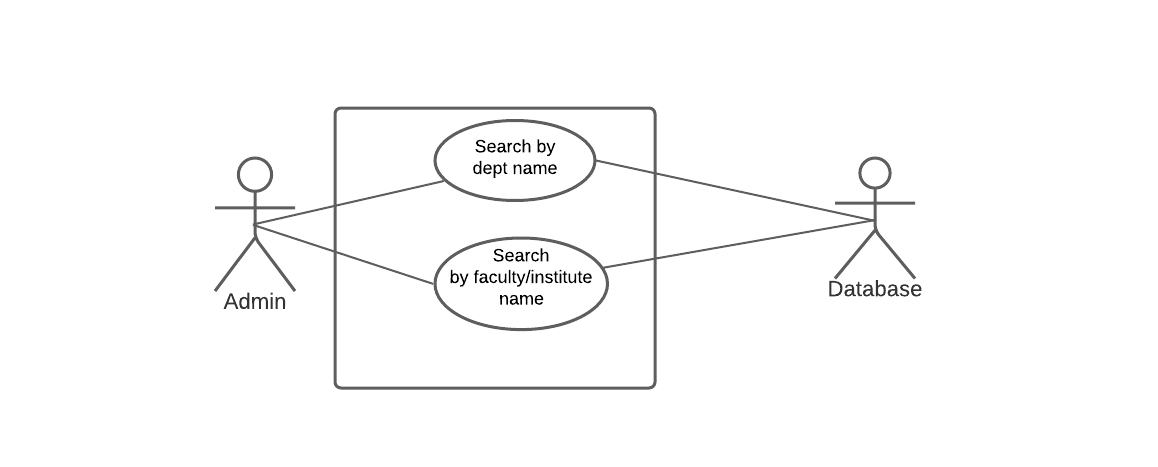


Figure 14: Level 2.7– Department Search

**Use Case**: Department Search

**Use Case ID**: 0

**Primary Actors**: Admin

**Secondary Actor**: Database

**Goal in Context:** Successful money transaction.

**Preconditions**: User must have a web browser and internet connection. Admin should be logged in. Department has been in the list.

**Trigger**: Admin search department according name or faculty/institute name.

**Scenario**: Admin logged in then click into Departments and search from the search icon.

**Exception**: Internet connection broken. Search input wrong.

**Priority**: Essential.

**When available**: First Increment.

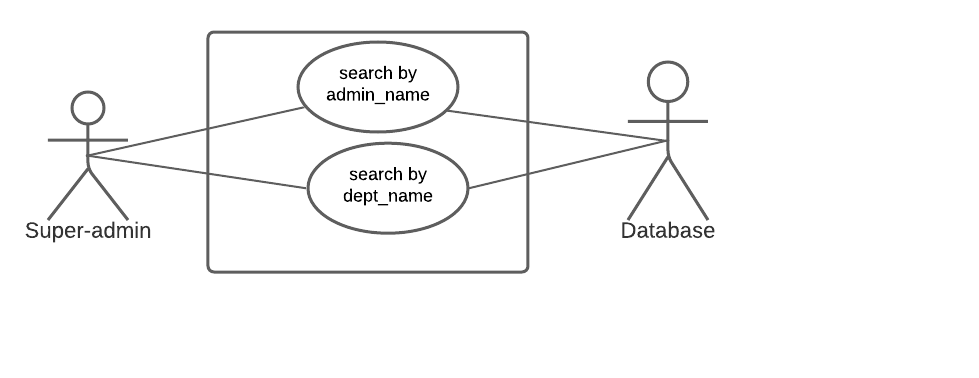


Figure 15: Level 2.8- Admin Search

**Use Case**: Admin Search

**Use Case ID**: 0

**Primary Actors**: Super-Admin

**Secondary Actor**: Database

**Goal in Context:** Search the admin from the list.

**Preconditions**: User must have a web browser and internet connection. Admin should be logged in. Admin has been in the list.

**Trigger**: Super-Admin search admin according name or department name.

**Scenario**: Super-Admin logged in then click into admin and search from the search icon.

**Exception**: Internet connection broken. Search input wrong.

**Priority**: Essential.

**When available**: First Increment.

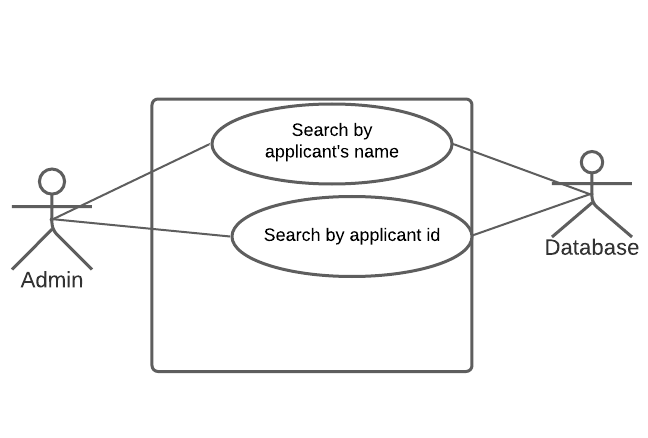


Figure 16: Level 2.9– Applicant Search

**Use Case**: Applicant Search

**Use Case ID**: 0

**Primary Actors**: Admin

**Secondary Actor**: Database

**Goal in Context:** Search the applicant who complete the application.

**Preconditions**: User must have a web browser and internet connection. Admin should be logged in. Applicant has been in the list.

**Trigger**: Admin search applicant according name or id.

**Scenario**: Admin logged in then click into applicant and search from the search icon.

**Exception**: Internet connection broken. Search input wrong.

**Priority**: Essential.

**When available**: First Increment.

Chapter 5 – Data Model

# 5.1 Data Modeling Concepts

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.

# 5.2 Data Objects

A data object is a representation of composite information that must be understood by the  
software. Here, composite information means an information 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. All the nouns in the scenario were identified.

Table 2: Data objects

|  |  |  |
| --- | --- | --- |
| No. | Noun | Attributes of |
| 1 | Name | 10,11,15,29,30 |
| 2 | Father’s name | 10,11 |
| 3 | Mother’s name | 10,11 |
| 4 | Registration no. | 10,11 |
| 5 | Session | 10,11 |
| 6 | Phone number | 10,11 |
| 7 | Department | 10,11,30 |
| 8 | Date of birth | 10,11 |
| 9 | Current year | 10,11 |
| 10 | User | 1,2,3,4,5,6,7,8,9 |
| 11 | Student | 1,2,3,4,5,6,7,8,9 |
| 12 | Testimonial | 27 |
| 13 | Language certificate | 28 |
| 14 | Transaction | 19,20,31,33 |
| 15 | Admin | 16,17,1 |
| 16 | Email | 10,11,15,29,30 |
| 17 | Password | 10,15,29,30 |
| 18 | Application | 22,23,24 |
| 19 | Transaction id | 14 |
| 20 | Transaction no | 14 |
| 21 | Search key |  |
| 22 | Application acceptance | 18 |
| 23 | Application cancelation | 18 |
| 24 | Status | 18 |
| 25 | Notification |  |
| 26 | Document |  |
| 27 | Testimonial id | 12 |
| 28 | Certificate id | 13 |
| 29 | Super admin | 1,16,17 |
| 30 | Normal admin | 1,16,17 |
| 31 | Transaction status | 14 |
| 32 | Applicant id | 10,11,18 |
| 33 | amount | 14 |

In this above data objects, we assume these data table for our database.

1. User (U\_id, name, email, password)

2. Student (applicant\_id, name, email, father\_name, mother\_name, birthdate, session, department)

3. Admin (id, name, email, password, type)

4. Document (id, type, date)

5. Certificate (certificate\_id, date,name)

6. Testimonial (testimonial\_id, date,name)

7. Transaction (transaction\_id, status, transaction\_no, amount)

8. Application (application\_id, date, application\_status)

9. Department (id, dept\_name, faculty/institute\_name)

|  |  |  |
| --- | --- | --- |
| User | | |
| **Attribute** | | **Type** | **Size** |
| User\_id | | NUMBER | 15 |
| Name | | VARCHAR2 | 25 |
| Email | | VARCHAR2 | 30 |
| Password | | VARCHAR2 | 14 |

|  |  |  |
| --- | --- | --- |
| Student | | |
| **Attribute** | | **Type** | **Size** |
| Applicant\_id | | NUMBER | 15 |
| Name | | VARCHAR2 | 25 |
| Email | | VARCHAR2 | 30 |
| Father\_name | | VARCHAR2 | 20 |
| Mother\_name | | VARCHAR2 | 20 |
| Session | | VARCHAR2 | 20 |
| Registration\_no | | VARCHAR2 | 20 |
| Roll\_no | | VARCHAR2 | 10 |
| Department\_name | | VARCHAR2 | 20 |

|  |  |  |
| --- | --- | --- |
| Admin | | |
| **Attribute** | | **Type** | **Size** |
| Admin\_id | | NUMBER | 15 |
| Name | | VARCHAR2 | 25 |
| Email | | VARCHAR2 | 30 |
| Password | | VARCHAR2 | 14 |
| Type | | VARCHAR2 | 14 |

|  |  |  |
| --- | --- | --- |
| Document | | |
| **Attribute** | **Type** | **Size** |
| Id | NUMBER | 15 |
| Date | NUMBER | 25 |
| Type | VARCHAR2 | 14 |

|  |  |  |
| --- | --- | --- |
| Testimonial | | |
| **Attribute** | **Type** | **Size** |
| Testimonial\_id | NUMBER | 15 |
| Date | NUMBER | 25 |
| Applicant\_name | VARCHAR2 | 20 |

|  |  |  |
| --- | --- | --- |
| Certificate | | |
| **Attribute** | **Type** | **Size** |
| Certificate\_id | NUMBER | 15 |
| Date | NUMBER | 25 |
| Applicant\_name | VARCHAR2 | 20 |

|  |  |  |
| --- | --- | --- |
| Transaction | | |
| **Attribute** | **Type** | **Size** |
| transaction\_id | NUMBER | 15 |
| status | VARCHAR2 | 25 |
| Transaction\_no | NUMBER | 20 |
| Amount | NUMBER | 20 |

|  |  |  |
| --- | --- | --- |
| Application | | |
| **Attribute** | **Type** | **Size** |
| Application\_id | NUMBER | 15 |
| Date | NUMBER | 25 |
| Status | VARCHAR2 | 20 |

|  |  |  |
| --- | --- | --- |
| Department | | |
| **Attribute** | **Type** | **Size** |
| Id | NUMBER | 15 |
| Dept\_name | VARCHAR2 | 25 |
| Faculty\_name | VARCHAR2 | 20 |

# 5.3 ER diagram

In this section relationship between different data entities are shown Chen diagram is used for the representation of the relationships.

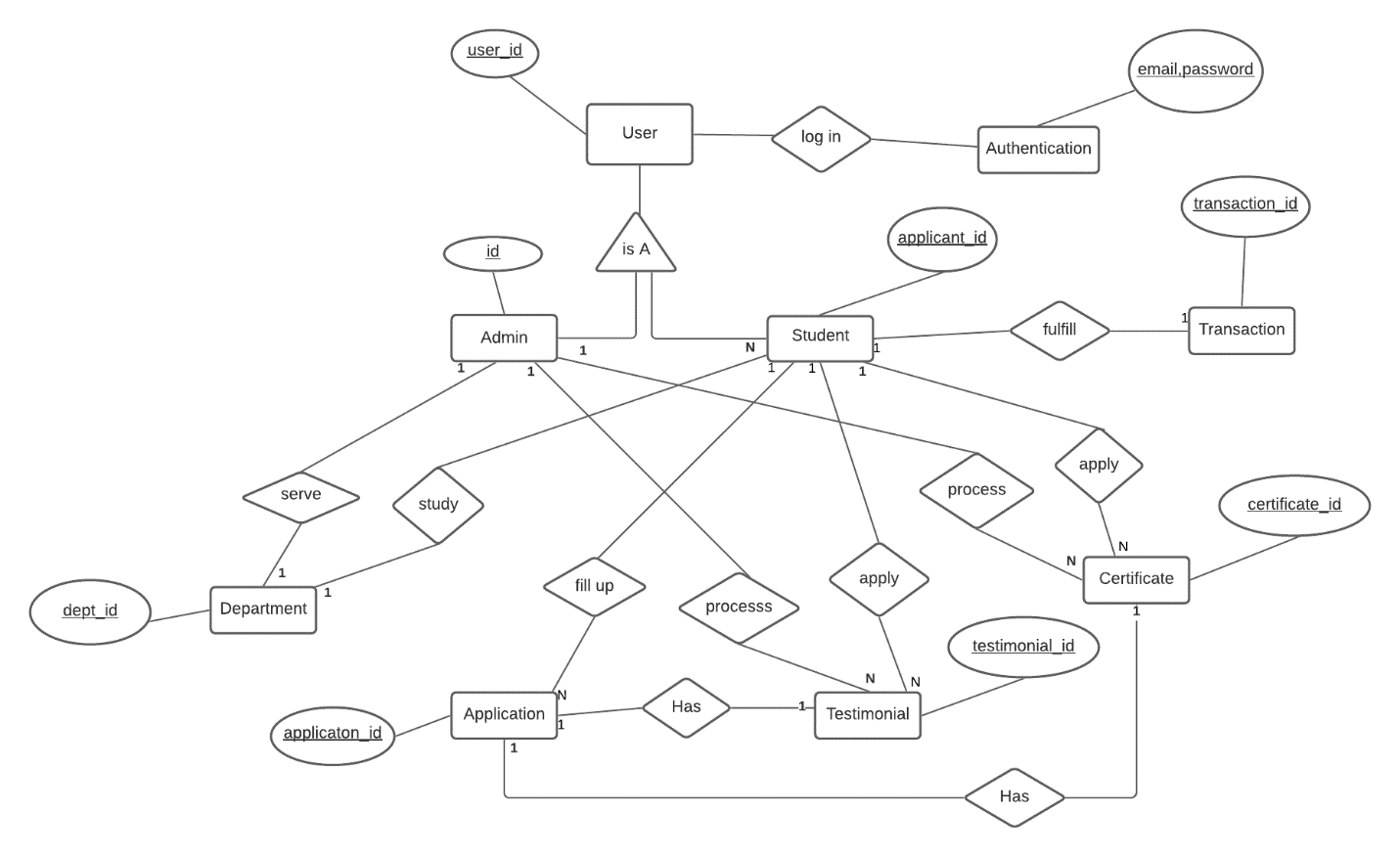


Figure 17: ER diagram

Chapter 6: Class Based Model

# 6.1 Introduction:

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

# 6.2 Identifying Analysis Classes:

To identify the potential classes, nouns were selected from the solution space of the story. These were then characterized in 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

Following are the specifications of the nouns according to the general classifications.

## 6.2.1 Selection Criteria

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information
2. Needed Services
3. Multiple Attributes
4. Common attributes
5. Common operations
6. Essential requirements

## 6.2.2 Associate Noun and Verb Identification

The nouns and the verbs associated with the potential classes are identified to find out the attributes and methods of each class.

Table 3: Noun and verb identification

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Potential Class | Noun | Verb |
| 1 | System | Admin,Student | Store, Update, Remove |
| 2 | Admin | Id,password,email,name,department | Approve request, add department, remove, edit department, receive application, ready document, Role given,  send notification, Receive notification |
| 3 | Student | Applicant\_id, name, email, session, mother\_name, father\_name,  registration\_no, department. | account, application, see message, receives notification, search, log in |
| 4 | Testimonial | Id,status, date | Accept, reject, sends notification, change status |
| 5 | Language Certificate | Id,status, date | Accept, reject, sends notification, change status |
| 6 | Transaction | Id, status, amount, transaction\_no | Confirm transaction, update |
| 7 | Department | Id, name, faculty name | Add, delete, update |
| 8 | Application | Id, status, date | See status, Accept, Reject, Update |
| 9 | User | Id, email, name, personal details | Account, log in, sign up, receive notification,search,update |
| 10 | Account | Id, type | Log in, application, verify, modify, delete |

## 6.2.3 Attribute Selection and Method Identification

|  |  |  |  |
| --- | --- | --- | --- |
| No | Potential Class | Attribute | Method |
| 1 | System | Admin,Student | updateDatabse()  storeInformation()  manageInformation()  view() |
| 2 | Admin | Id,password,email,name,department | acceptRequsest()  cancelRequest()  addDept()  removeDept()  editDept()  adnormalAdmin()  giveRole()  documentReady()  editAdmin()  deleteNormalAdmin()  viewApplicantdetails() |
| 3 | Student | Applicant\_id, name, email, session, mother\_name, father\_name,  registration\_no, department. | setPersonalDetails()  getPersonalDetails()  application()  SignIn()  receiveNotification()  getTestimonial()  getCertificate()  ViewDetails()  showStatus() |
| 4 | Testimonial | Id,status, date | makeTestimonial()  application()  acceptTestimonial()  rejectTestimonial()  changeStatus()  sendNotification() |
| 5 | Language Certificate | Id,status, date | makeCertificate()  application()  acceptCertificate()  rejectCertificate()  changeStatus()  sendNotification() |
| 6 | Transaction | Id, status, amount, transaction\_no | generateID()  createTransaction()  paidAmount()  updateCash()  changeTransactionStatus() |
| 7 | Department | Id, name, faculty name | addDept()  editDept()  removeDept() |
| 8 | Application | Id, status, date | sentApplication()  receiveApplcation()  status() |
| 9 | User | Id, email, name, personal details | Login()  Singup()  Receivenotification()  Search() |
| 10 | Account | Id, type | signUp() validateInput() setUser() login() GetUser() modifyaccount() deleteAccount() recoveryAccount() |

# 6.3 Class Responsibility Collaboration (CRC)

|  |  |
| --- | --- |
| System | |
| Responsibilities | Collaboration | |
| Handling Database  Handling Information  Sending Notification | Account  User  User, Testimonial, Certificate | |

|  |  |
| --- | --- |
| Account | |
| Responsibilities | Collaboration |
| Account Creation Getting Input for sign up & sign in Log Out Recovery Data | Student  Admin. Student  System |

|  |  |
| --- | --- |
| Admin | |
| Responsibilities | Collaboration | |
| Authentication  Manage Application  Prepare document  Managing Department’s info | Student  Student, Application  Testimonial, Certificate  Department | |

|  |  |
| --- | --- |
| Student | |
| Responsibilities | Collaboration | |
| Application for testimonial  Application for Certificate  Notification receives  Getting document  Status seen | Application, Testimonial  Application, Certificate  System  Testimonial, Certificate, Admin  System | |

|  |  |
| --- | --- |
| Testimonial | |
| Responsibilities | Collaboration | |
| Application for testimonial  Testimonial Processing  Sending Notification | Application, Student  Admin  System | |

|  |  |
| --- | --- |
| Certificate | |
| Responsibilities | Collaboration | |
| Application for certificate  Certificate Processing  Sending Notification | Application, Student  Admin  System | |

|  |  |
| --- | --- |
| Transaction | |
| Responsibilities | Collaboration | |
| Fill up transactional input  Payment procedure  Sending Notification | Student  System  System | |

|  |  |
| --- | --- |
| Application | |
| Responsibilities | Collaboration | |
| Application for document  Status procedure | Application, Student, Admin  System, Student | |

|  |  |
| --- | --- |
| Department | |
| Responsibilities | Collaboration | |
| Adding departmental information  Store department information | Admin  System | |

# 6.4 UML Class Diagram

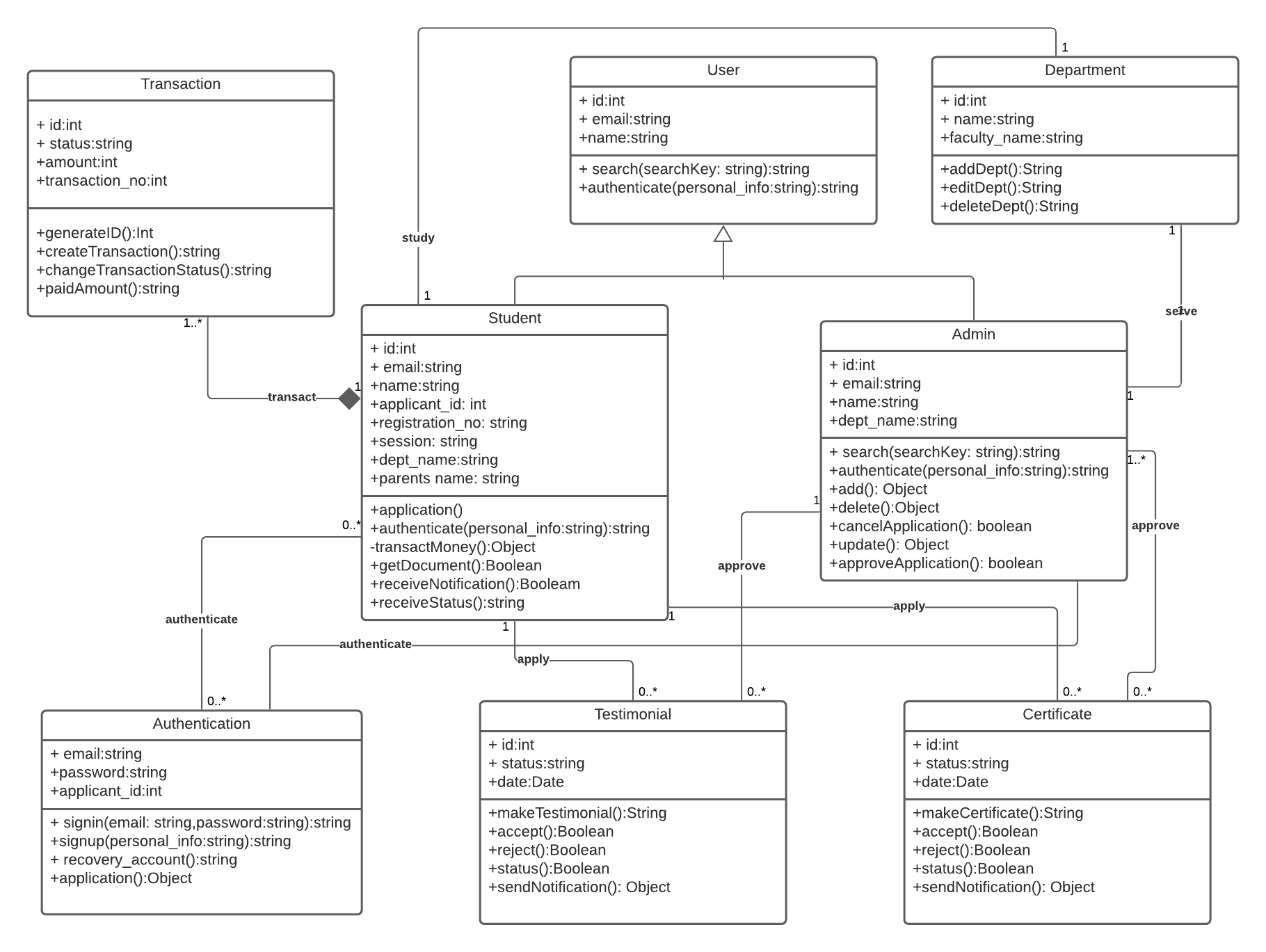


Figure 18: UML class diagram

Chapter 7: Software Design Architecture

# 7.1 Architectural Design for OOP

The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. At the architectural design level, a software architect uses an architectural context diagram (ACD) to model the manner in which software interacts with entities external to its boundaries.

# 7.2 Representing the system in a context

Referring to the figure, systems that interoperate with the target system (the system for which an  
architectural design is to be developed) are represented as -

**Superordinate systems**— those systems that use the target system as part of some higher-level processing scheme.  
**Subordinate systems**—those systems that are used by the target system and provide data or  
processing that are necessary to complete target system functionality.  
**Peer-level systems**—those systems that interact on a peer-to-peer basis(i.e., information is either  
produced or consumed by the peers and the target system.  
**Actors**—entities (people, devices) that interact with the target system by producing or consuming information that is necessary for requisite processing. Each of these external entities communicates with the target system through an interface.

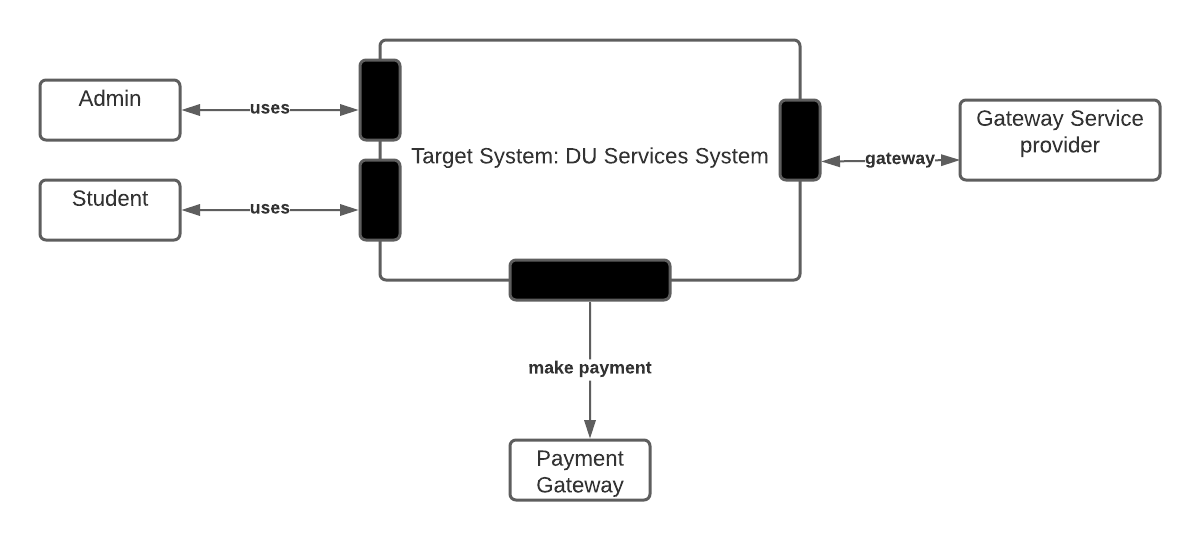


Figure 19:Architectural Context Diagram

# 7.3 Defining Architypes

An archetypeis a class or pattern that represents a core abstraction that is critical to the design of an architecture for the target system. In general, a relatively small set of archetypes is required to design even relatively complex systems. The target system architecture is composed of these archetypes, which represent stable elements of the architecture but may be instantiated many different ways based on the behavior of the system. The archetypes of this project are given below:

Analyzing class:

1. Admin: Admin control the testimonial and certificate process and add, modify the system class.
2. Student: Student apply for the document and get notification for his updated status.
3. Document: Testimonial and Certificate for the main purpose of this system.
4. Transaction: Control the payment process fully.
5. Authentication: Login, sign up, recovery etc. process is defined here.
6. Department: Student must be from a department and admin also.

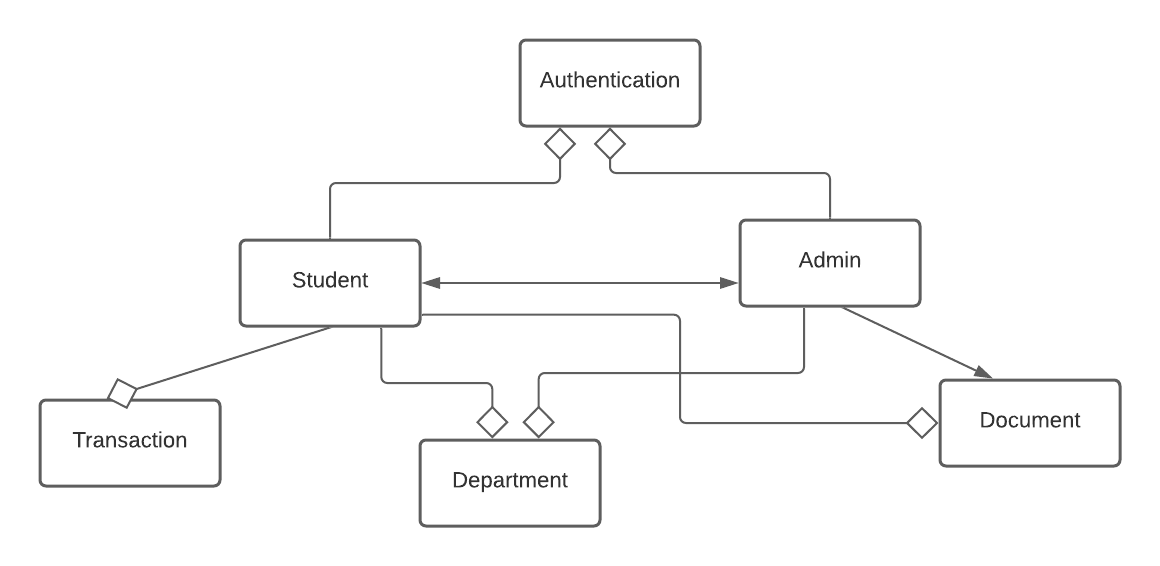


Figure 20: Archetype Design

# 7.4 Refining the architype into Component

As the software architecture is refined into components, the structure of the system begins to emerge. But how are these components chosen? In order to answer this question, you begin with the classes that were described as part of the requirements model. These analysis classes represent entities within the application(business)domain that must be addressed within the software architecture. Hence, the application domain is one source for the derivation and refinement of components. Another source is the infrastructure domain. The architecture must accommodate many infrastructure components that enable application components but have no business connection to the application domain. For example, memory management components, communication components, database components, and task management components are often integrated into the software architecture.

## 7.4.1 Instantiations of the system

The architectural design that has been modeled to this point is still relatively high level. The context of the system has been represented, archetypes that indicate the important abstractions within the problem domain have been defined, the overall structure of the system is apparent, and the major software components have been identified. However, further refinement (recall that all design is iterative) is still necessary.

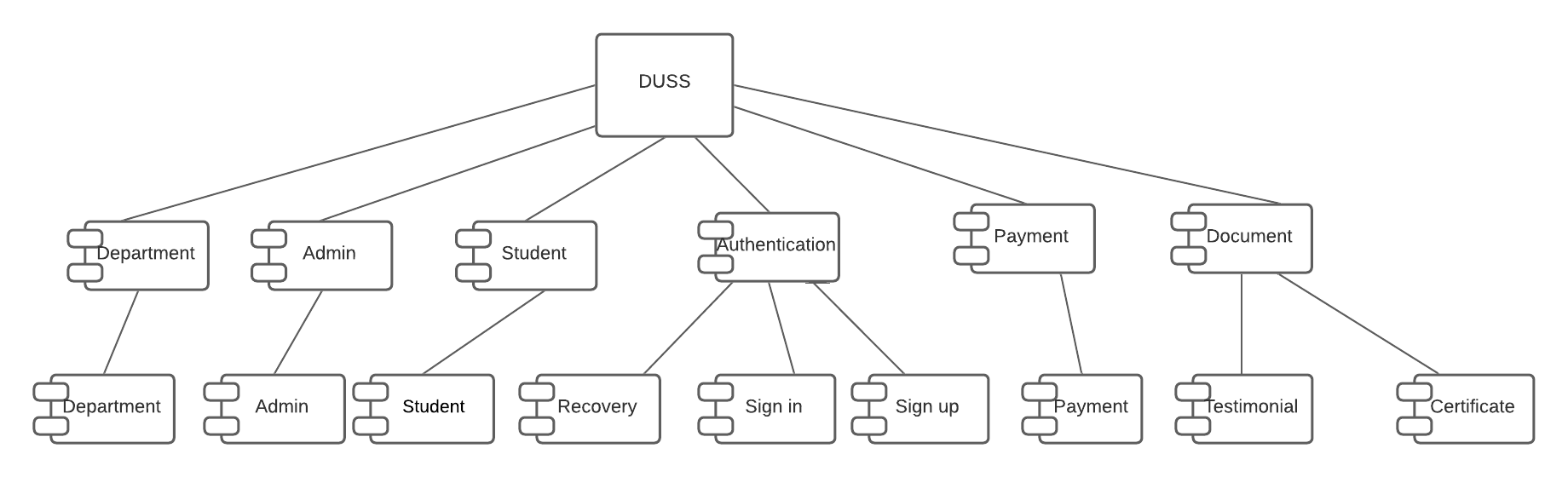


Figure 21: Instantiation

Chapter 8: Architectural Mapping

# 8.1 Introduction

A mapping technique, called structured design*,* is often characterized as a data flow-oriented design method because it provides a convenient transition from a data flow diagram to software architecture. This system is performing mapping with data where the type of information flow must be determined. One type of information flow is called transform flowand exhibits a linear quality. Data flows into the system along an incoming flow pathwhere it is transformed from an external world representation into internalized form. Once it has been internalized, it is processed at a transform center*.* Finally, it flows out of the system along an outgoing flow paththat transforms the data into external world form. There are six steps to transition the process. These are: (1) the type of information flow is established, (2) flow boundaries are indicated, (3) the DFD is mapped into the program structure, (4) control hierarchy is defined, (5) the resultant structure is refined using design measures and heuristics, and (6) the architectural description is refined and elaborated.

# 8.2 Transform Mapping

Transform mapping is a set of design steps that allows a DFD with transform flow characteristics to be mapped into a specific architectural style. To illustrate this approach, we consider our project. One element of the analysis model is a set of data flow diagrams that describe information flow within the system. To map these data flow diagrams into a software architecture, you would initiate the following design steps:

Step -1: Review the fundamental system model.

Step-2: Review and refine data flow diagrams for the software.

Step-3: Determine whether the DFD has transform or transaction flow11 characteristics.

Step-4: Isolate the transform center by specifying incoming and outgoing flow boundaries.

Step-5: Perform “first-level factoring.

Step-6: Perform “second-level factoring.

Step-7: Refine the first-iteration architecture using design heuristics for improved software quality.

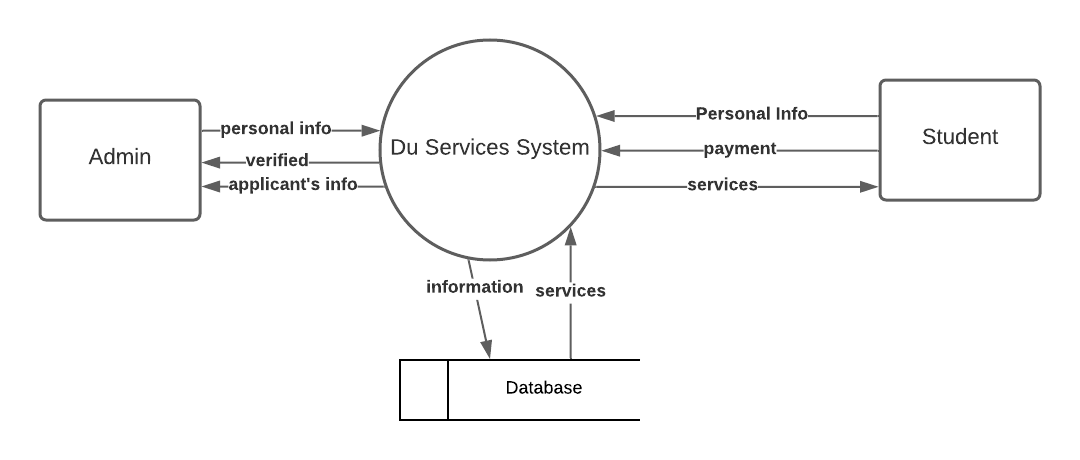


Figure 22: Context level DFD

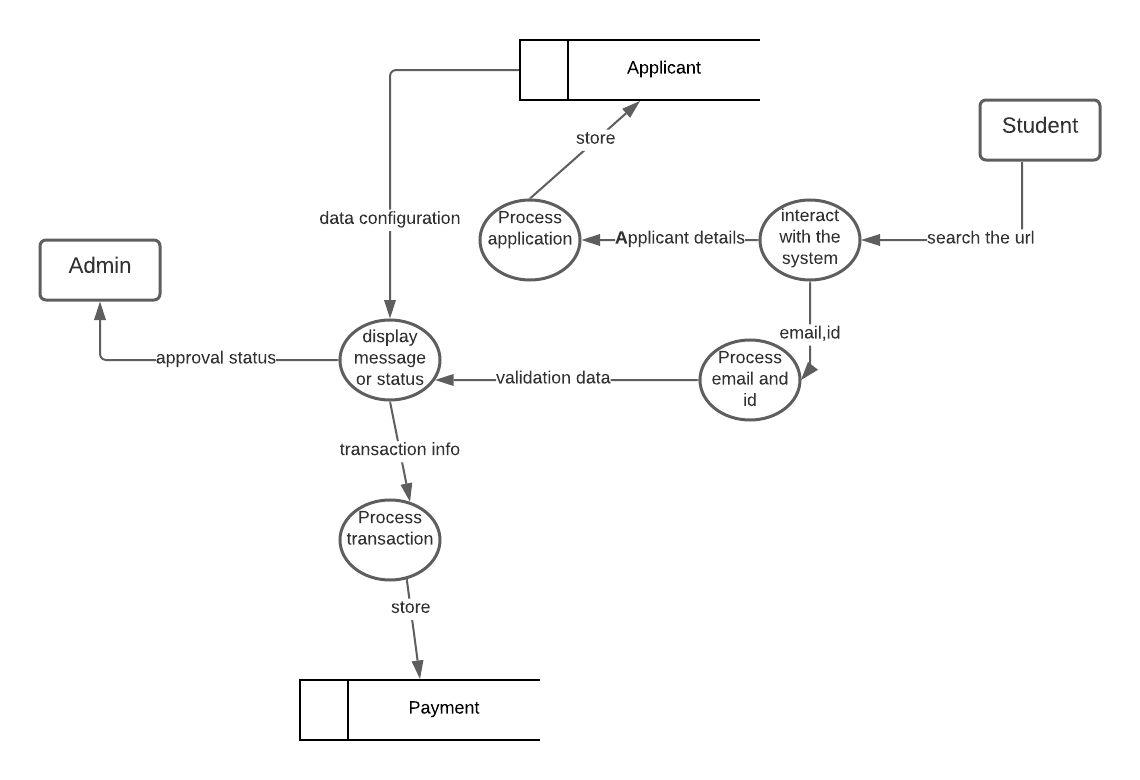


Figure 23: Level 1 mapping with DFD

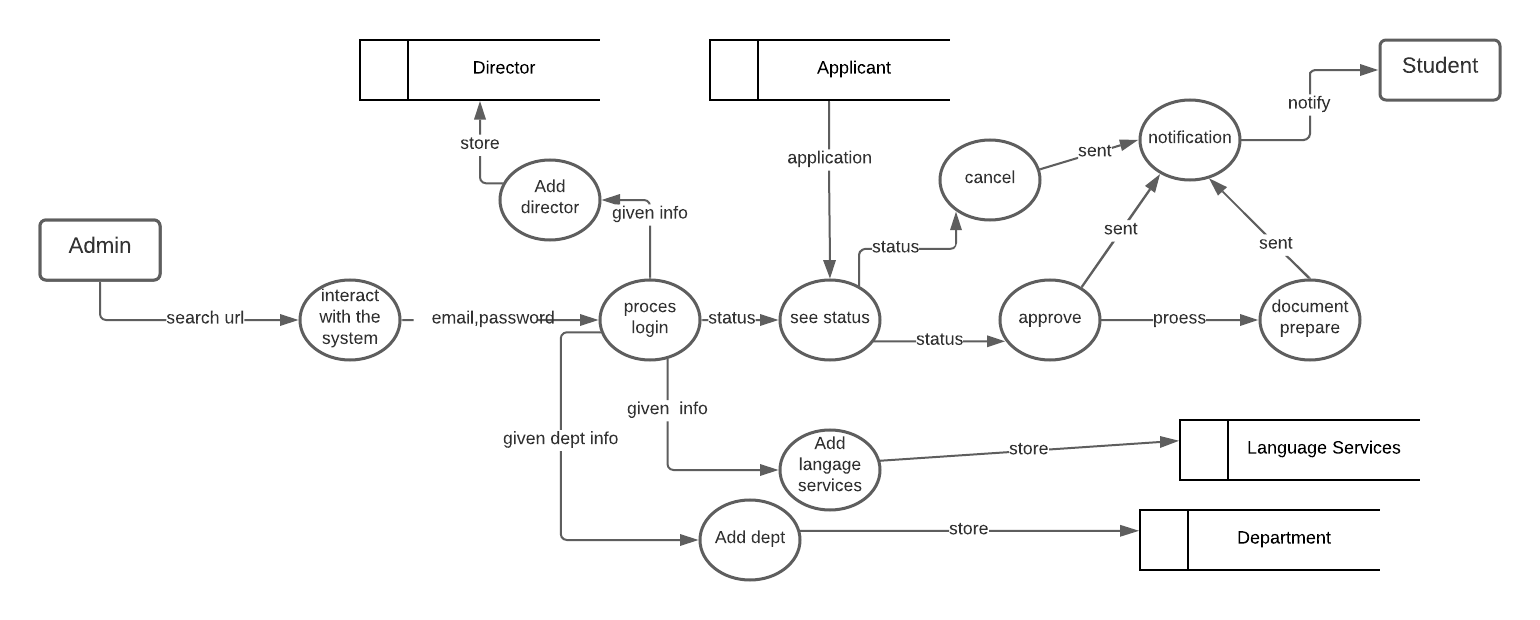


Figure 24: Level 2 mapping with DFD

Chapter 9: Component Level Design

# 9.1 Introduction

Component-level design occurs after the first iteration of architectural design has been completed. At this stage, the overall data and program structure of the software has been established. The intent is to translate the design model into operational software. But the level of abstraction of the existing design model is relatively high, and the abstraction level of the operational program is low. The translation can be challenging, opening the door to the introduction of subtle errors that are difficult to find and correct in later stages of the software process.

A componentis a modular building block for computer software. In the context of object-oriented software engineering, a component contains a set of collaborating classes. Each class within a component has been fully elaborated to include all attributes and operations that are relevant to its implementation. As part of the design elaboration, all interfaces that enable the classes to communicate and collaborate with other design classes must also be defined. To accomplish this,  
you begin with the requirements model and elaborate analysis classes (for components that relate to the problem domain) and infrastructure classes (for components that provide support services for the problem domain).

# 9.2 Analysis class

In Architectural design, I have selected some analyzing classes based on criteria. Those classes are:

* Admin: Admin control the testimonial and certificate process and add, modify the  
  system class.
* Student: Student apply for the document and get notification for his updated status.
* Document: Testimonial and Certificate for the main purpose of this system.
* Transaction: Control the payment process fully.
* Authentication: Login, sign up, recovery etc. process is defined here.
* Department: Student must be from a department and admin also.

# 9.3 Elaboration of Design Components

Here I depict the design components elaborately.

## 9.3.1 Student

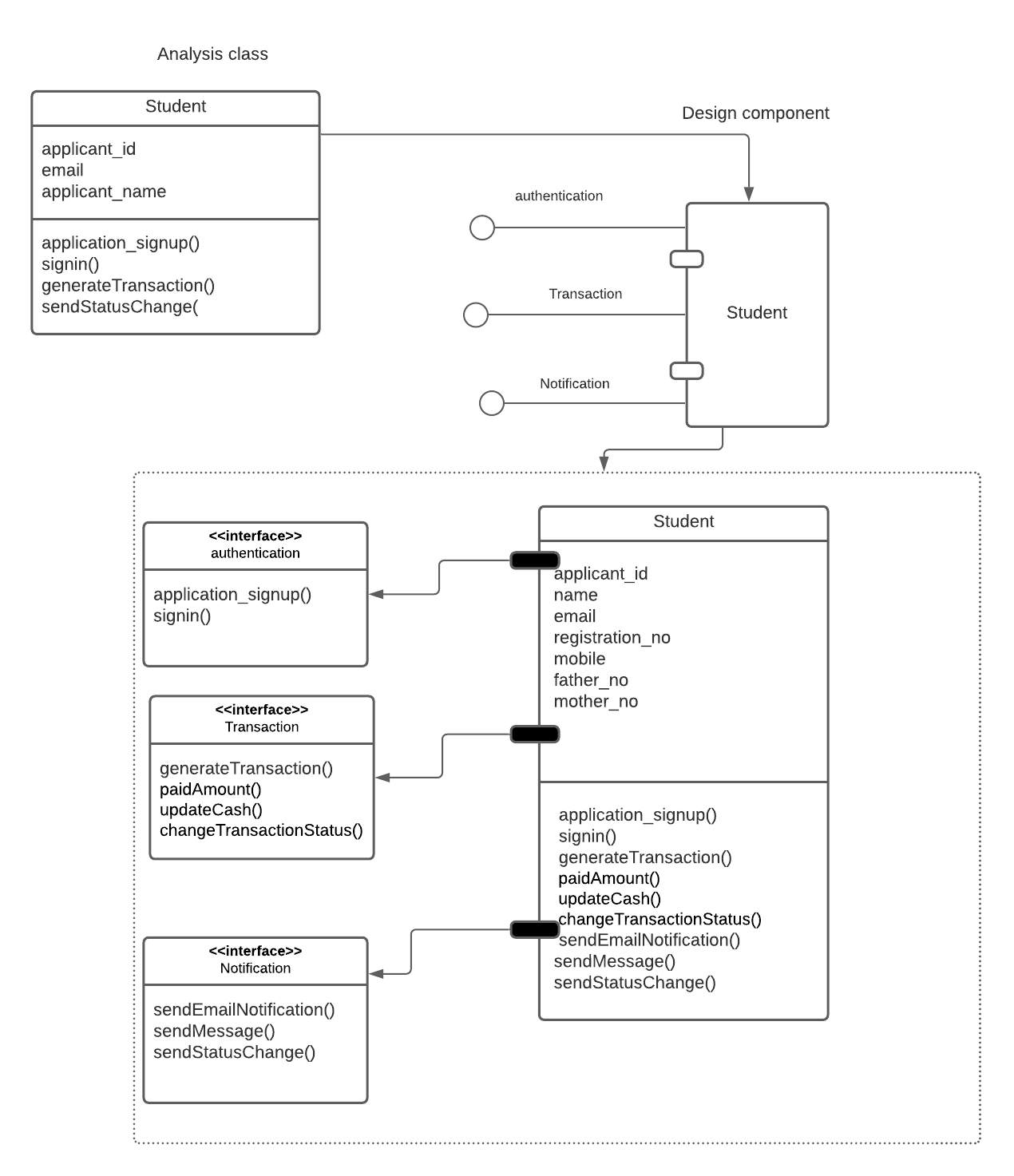


Figure 25: Analyses class for student component

## 9.3.2 Transaction

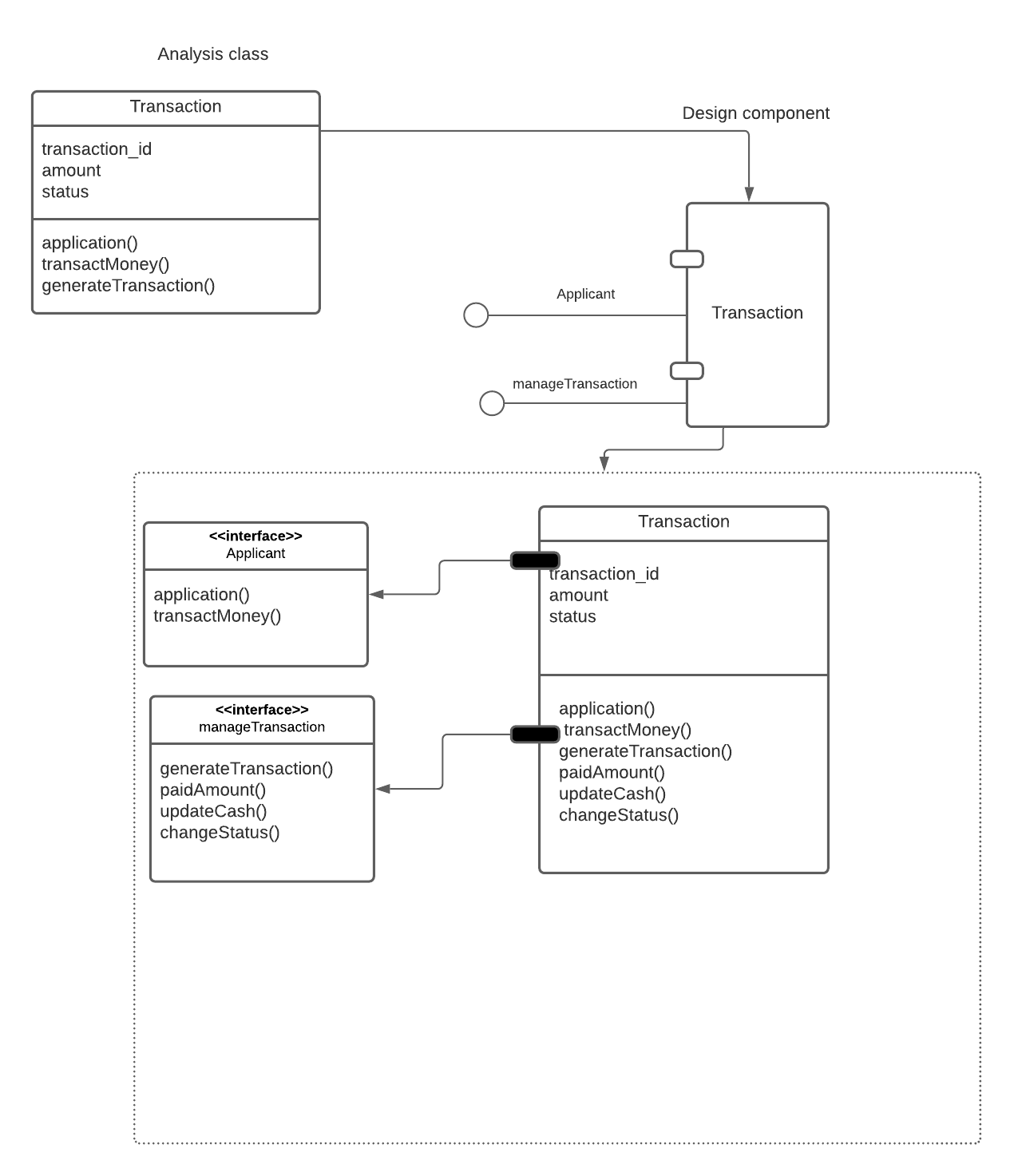


Figure 26: Analyses class for Transaction

## 9.3.3 Document

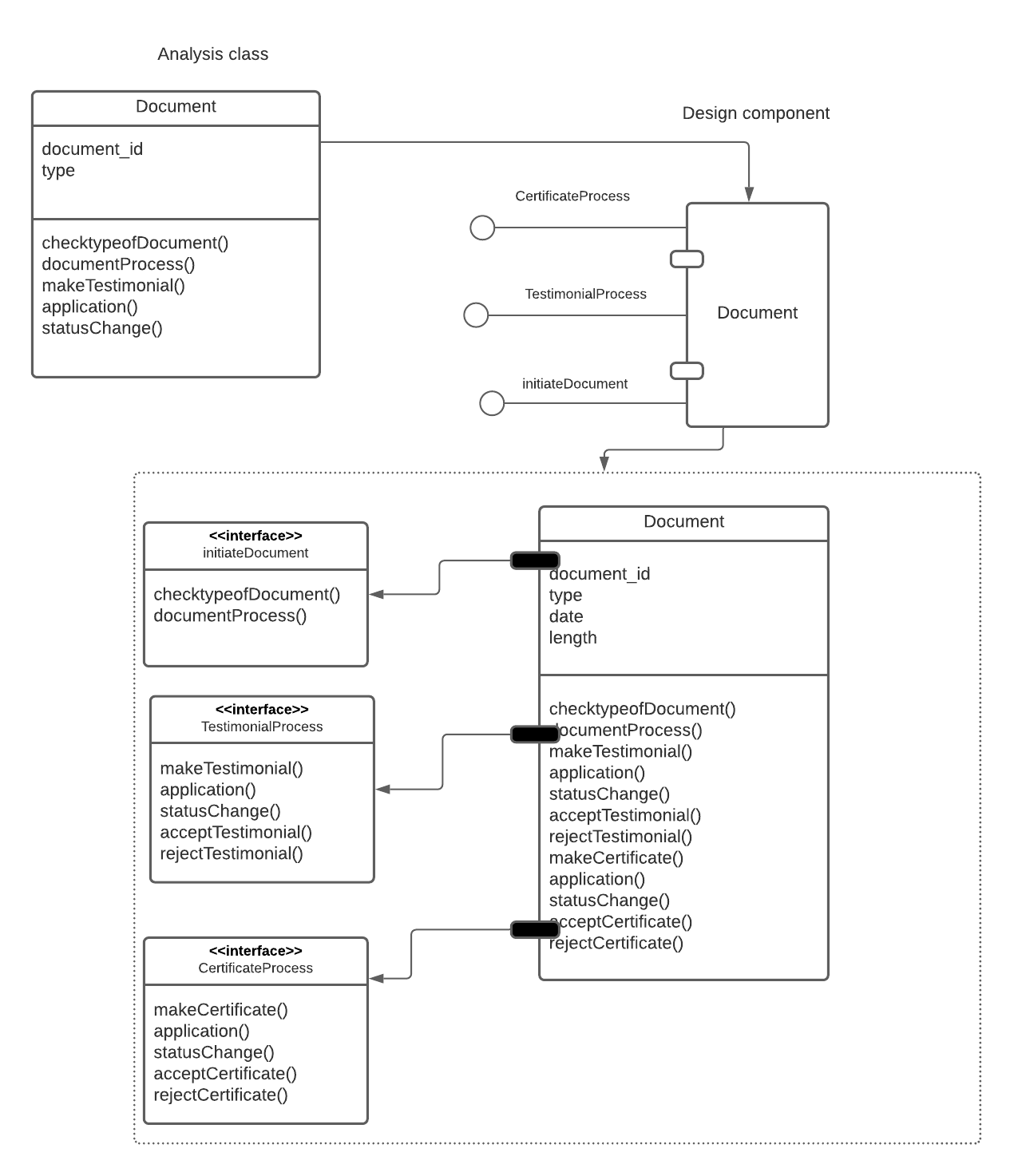


Figure 27: Analysis class for document

## 9.3.4 Authentication

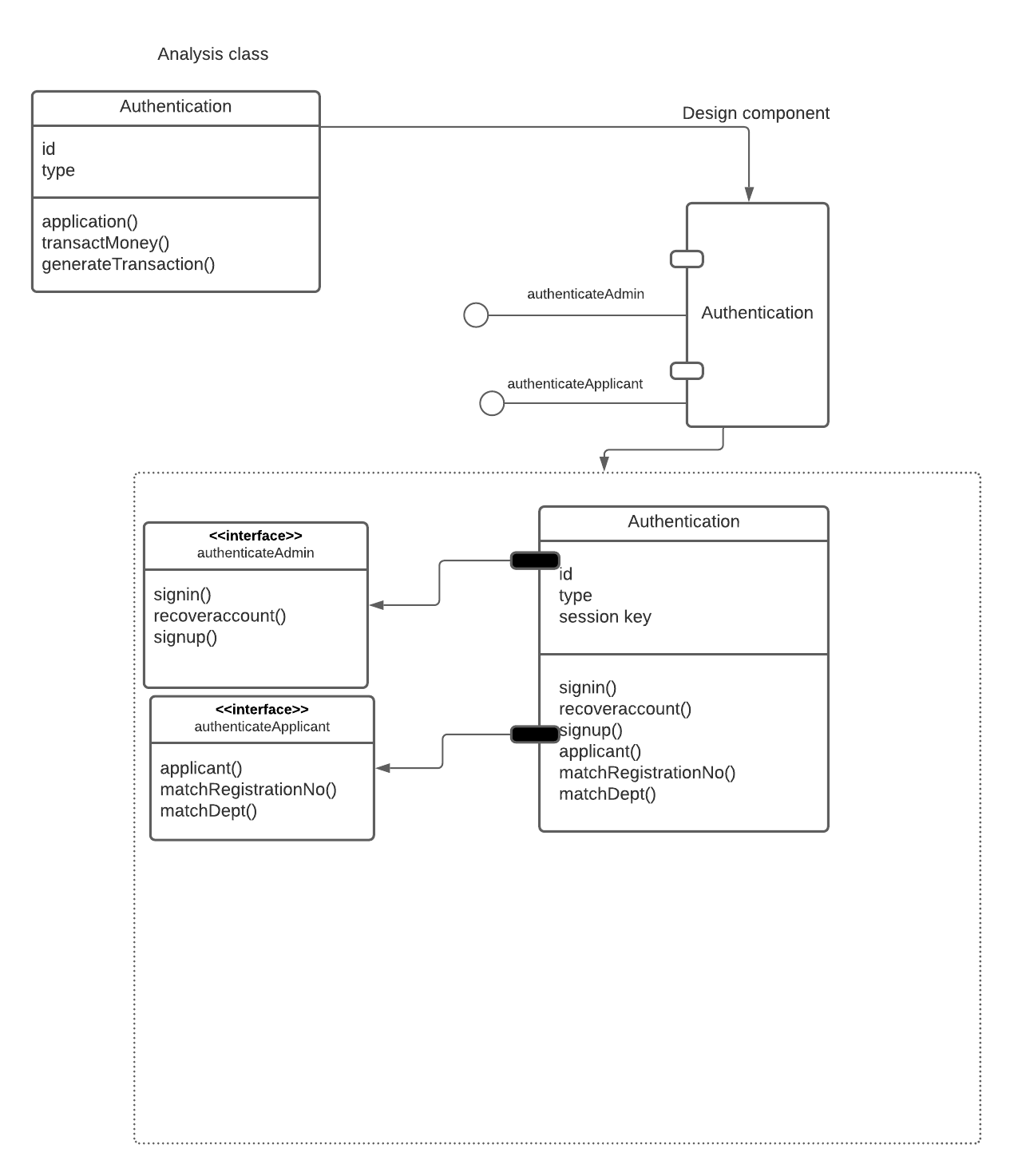


Figure 28: Analyses class for authentication

## 9.3.5 Department

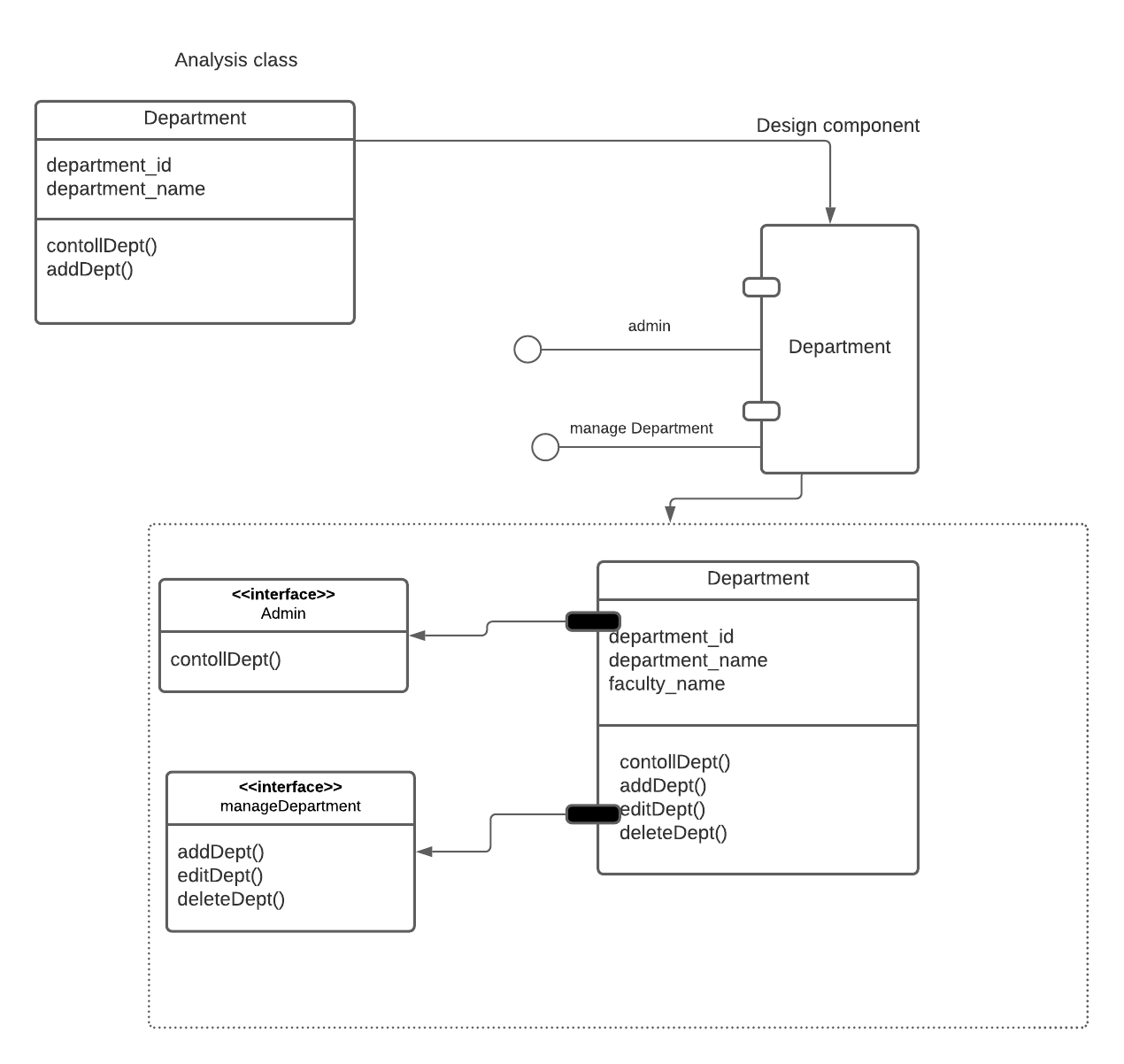


Figure 29:Analyses class for department

## 9.3.6 Admin

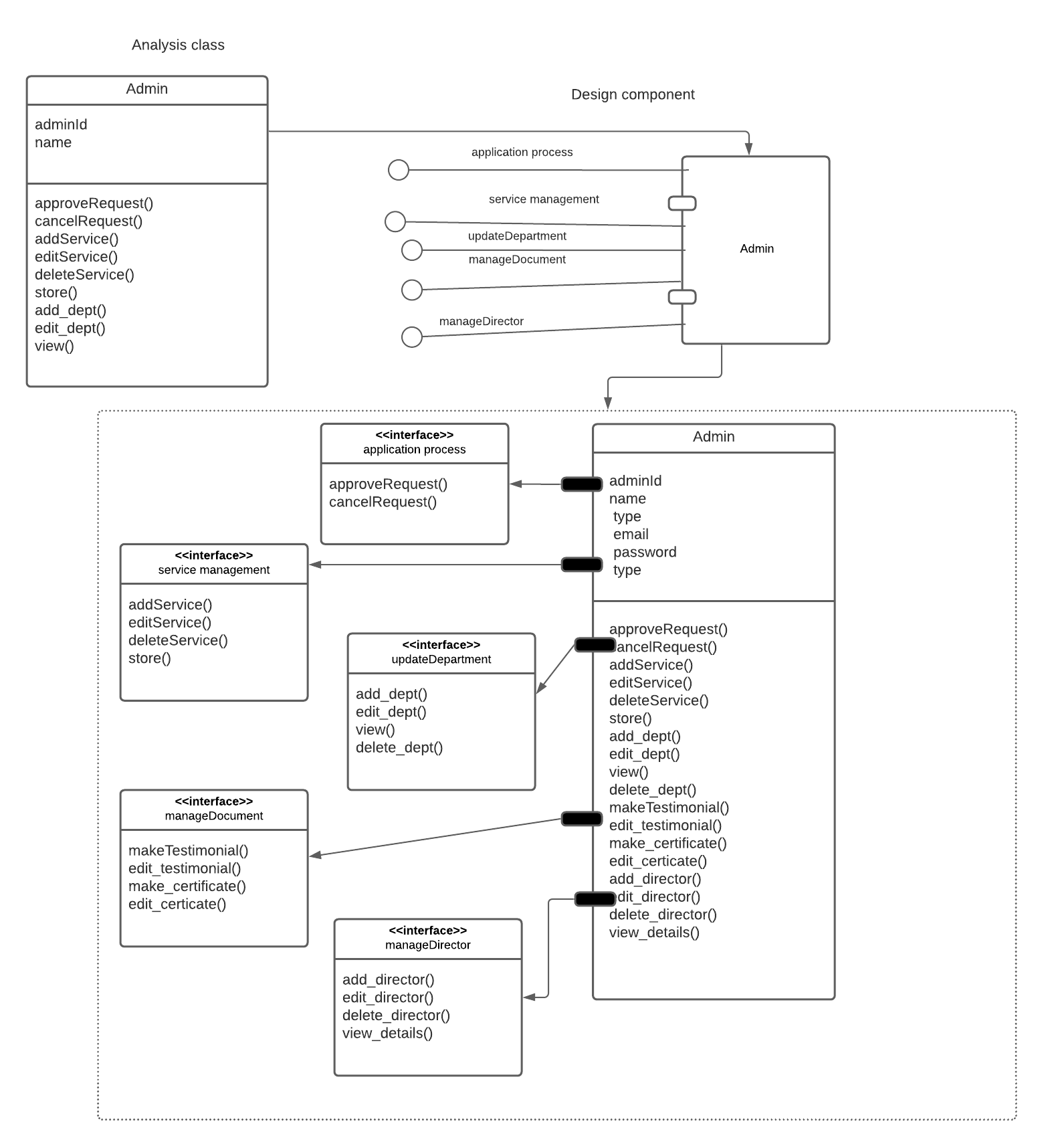


Figure 30: Analysis class for admin

# 9.4 Deployment Diagram

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. Elaborating deployment diagrams provide additional implementation details. Deployment diagrams are used as part of architectural design and are represented in descriptor form. In this form, major system functions (often represented as subsystems) are represented within the context of the computing environment that will house them.

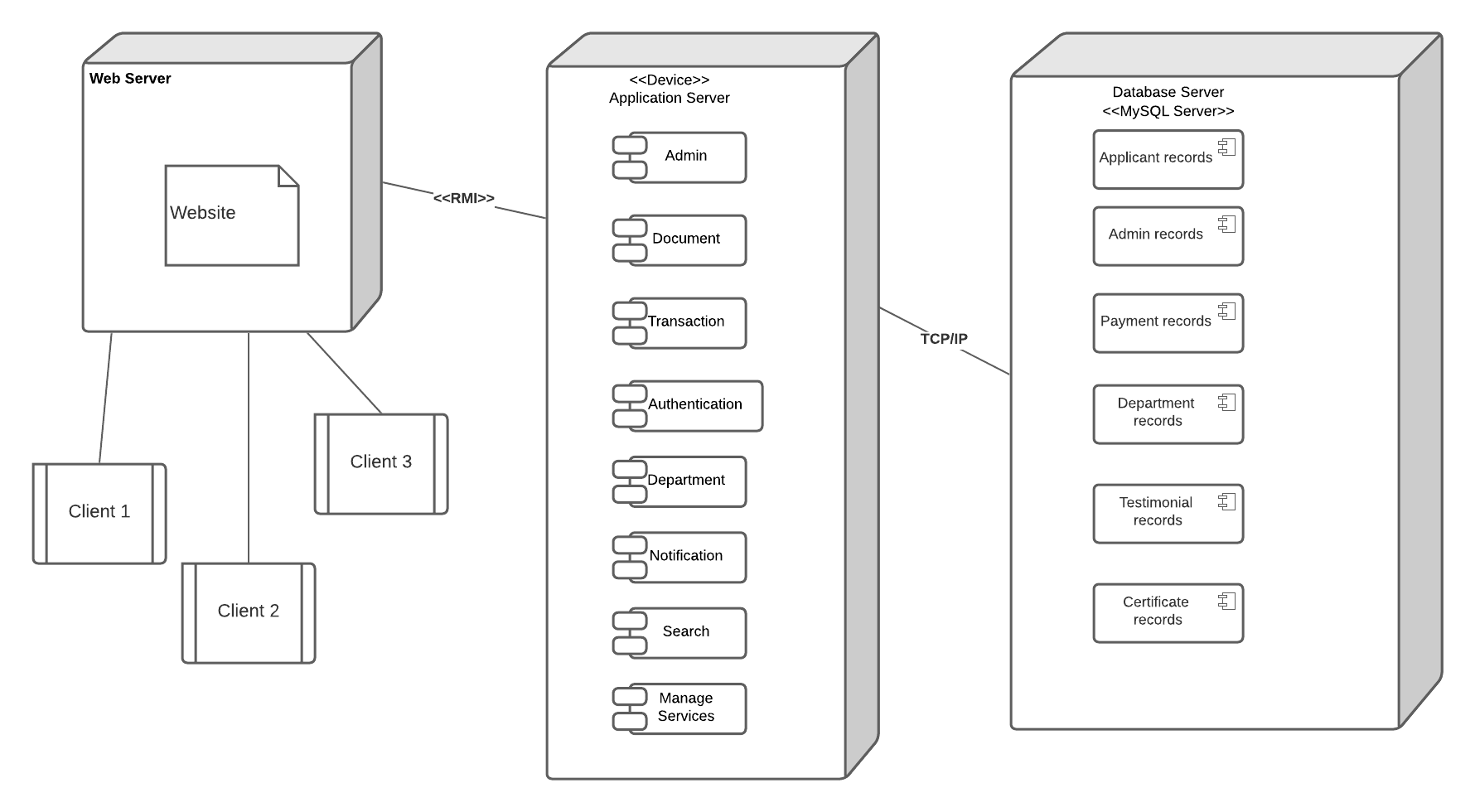


Figure 31: Deployment diagram

Chapter 10: User Interface Design

User Interface Design is the design of websites, computers, appliances, machines, mobile communication devices and software applications with the focus on the user's experience and interaction. The Golden Rules:

Three golden rules for interface design are:

1. Place the user in control.

2. Reduce the user’s memory load.

3. Make the interface consistent.

These golden rules actually form the basis for a set of user interface design principles that guide this important aspect of software design.

# 10.1 Interface Analysis

We divide interface analysis into the following part:

1. User analysis

2. Task analysis

## 10.1.1 User analysis

There are two steps in this part. 1) Identify User 2) Know User

### 10.1.1.1 Identify User

From the requirement and specification document we found two types of user in this system. These are:

1. Admin
2. Applicant

### 10.1.1.2 Know User

Table 4: Know User

|  |  |  |
| --- | --- | --- |
| User Type | Characteristics | Value |
| Admin | Age | 25-40 |
|  | Skills | Above Average |
|  | Domain expert | Yes |
|  | Application expert | Yes |
|  | Frequency of use | Frequently |
|  | Consequence of a mistake | No |
|  | General computer experience | Yes |
| Applicant | Age | 18-50 |
|  | Skills | Average |
|  | Domain expert | No |
|  | Application expert | Yes |
|  | Frequency of use | Frequently |
|  | Consequence of a mistake | No |
|  | General computer experience | Yes |

## 10.1.2 Task analysis

* Admin Task analysis

Table 5: Admin Task analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Process Goal** | **Precondition** | **Subtasks** |
| Application processing | Applicants apply for the testimonial and certificate and admin checked it | Logged in as admin using email and password | Approve application  Cancel application |
| Department manages | Adding department by admin | Logged in as an admin | Add department  Remove department  Delete department |
| Services manages | After admin login admin manages services | Logged in as an admin | Add service |
| Admin role | After admin login admin gives the role | Logged in as an admin | Super admin  Sub admin |

* Applicant Task analysis

Table 6: Applicant Task analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Process Goal** | **Precondition** | **Subtasks** |
| Application processing | Applicant application with giving details | Applicant must be valid and must have a valid registration no. | Give valid input |
| Transaction | Applicant log in or sign up and then pay now button click and | Logged in as an applicant or applicant must have an application | Generate transaction  Paid amount |
| Notification management | Applicant applies for an application. | Must have application | Email notification  Message notification |

# 10.2 Interface Design Steps

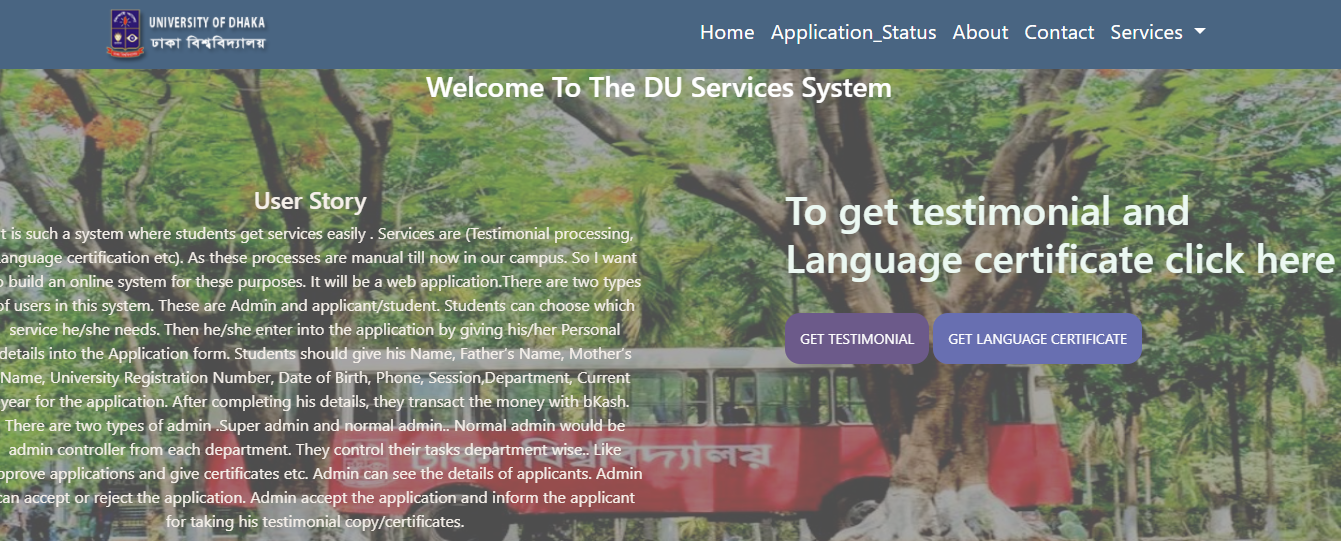


Figure 32: Home page

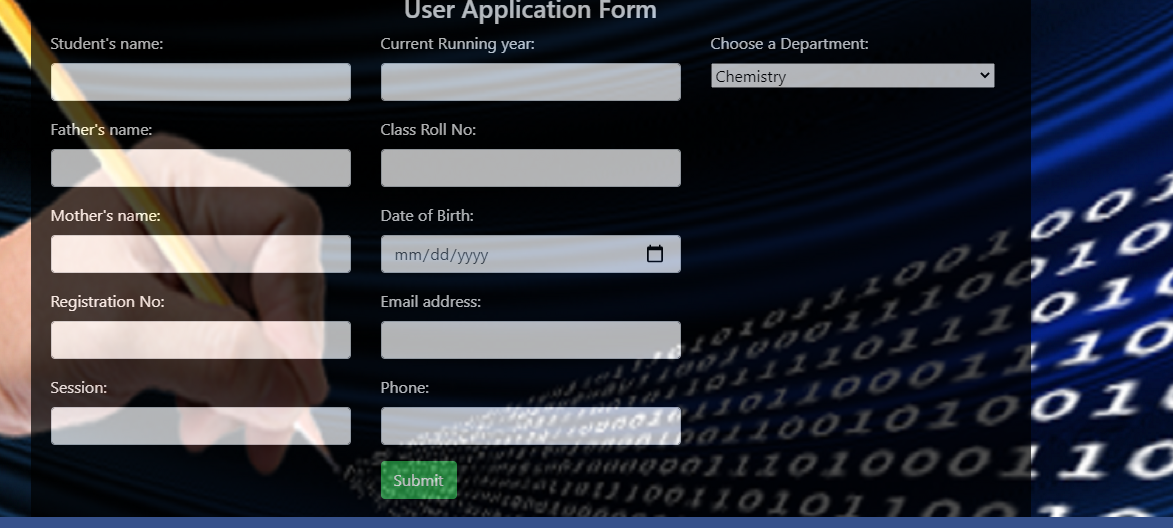


Figure 33: Application form

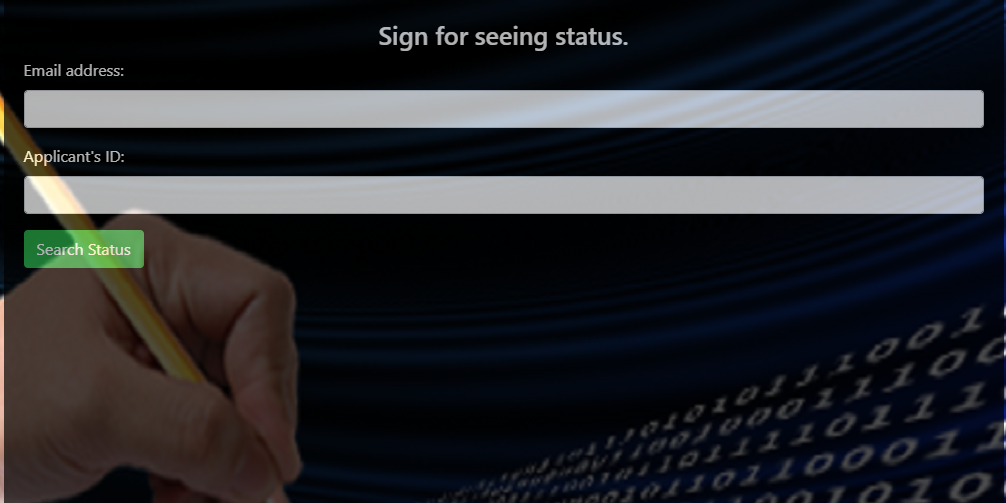


Figure 34: Search status or login



Figure 35:User profile

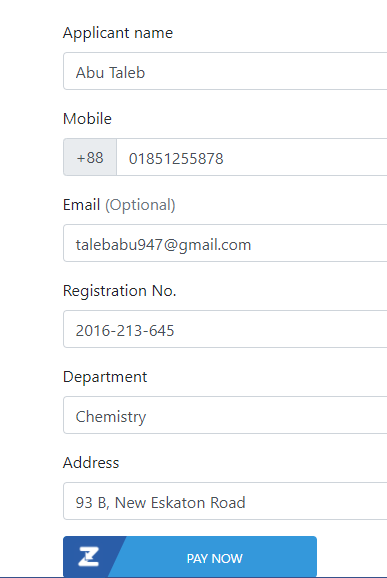


Figure 36: payment process

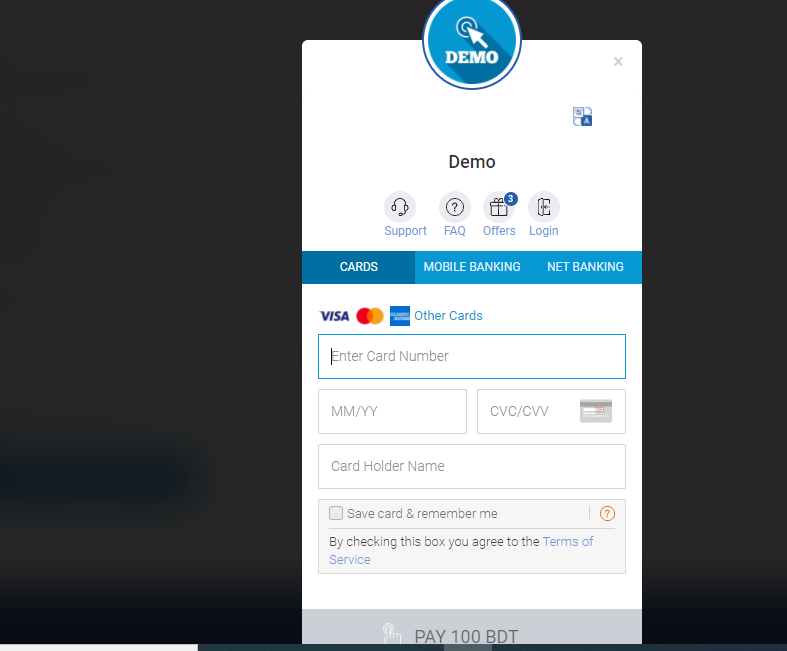


Figure 37: Transactional process

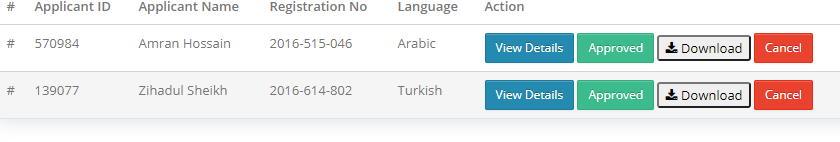


Figure 38: notification process

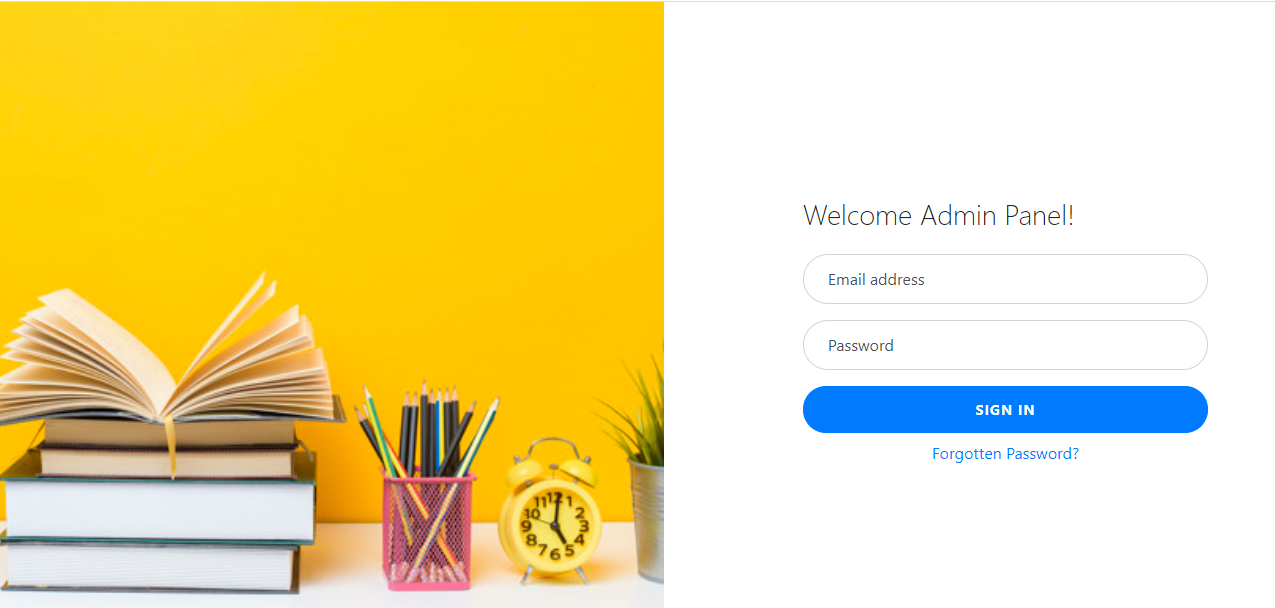


Figure 39:admin login

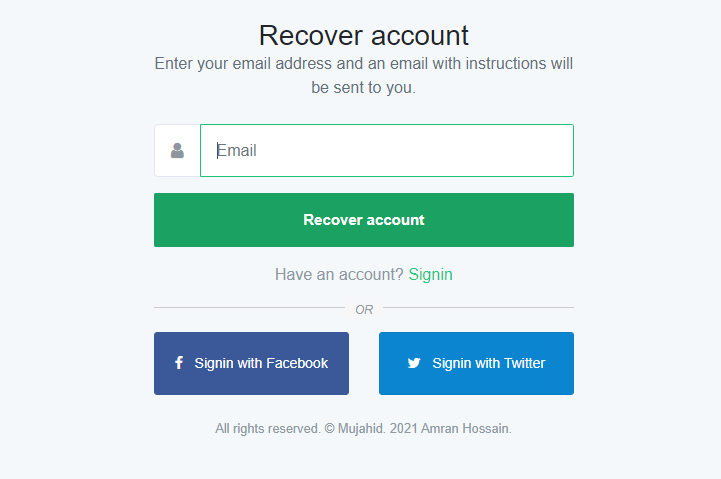


Figure 40: recover account



Figure 41:department list

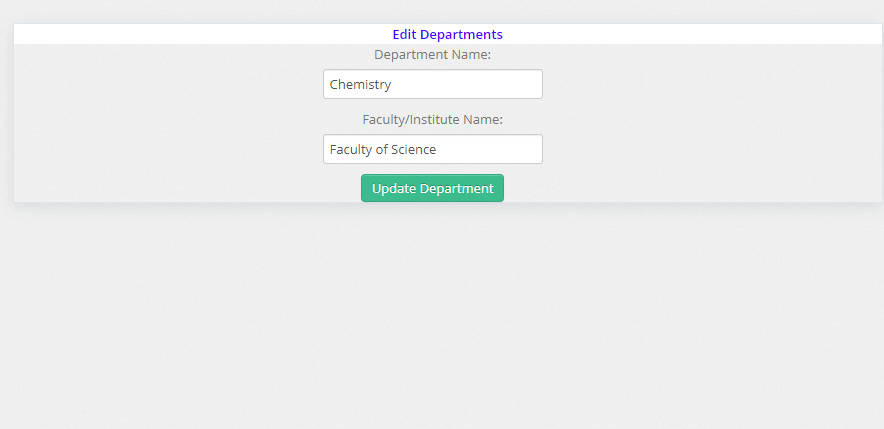


Figure 42: Edit

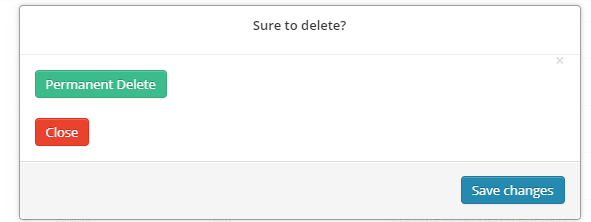


Figure 43: Delete

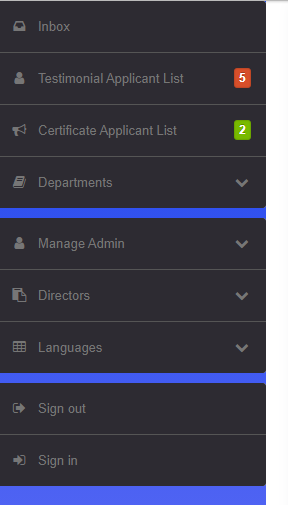


Figure 44: Sidebar admin

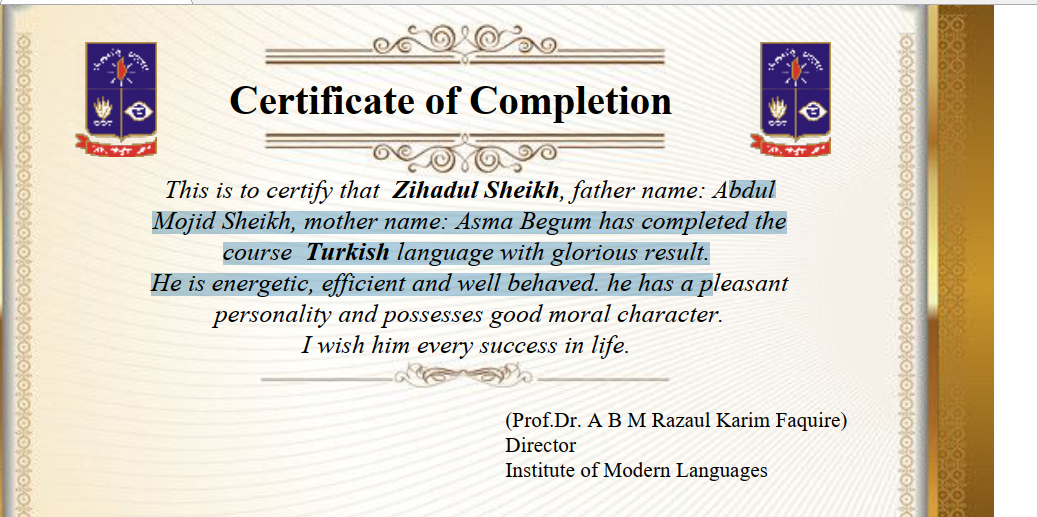


Figure 45: Certificate of Completion

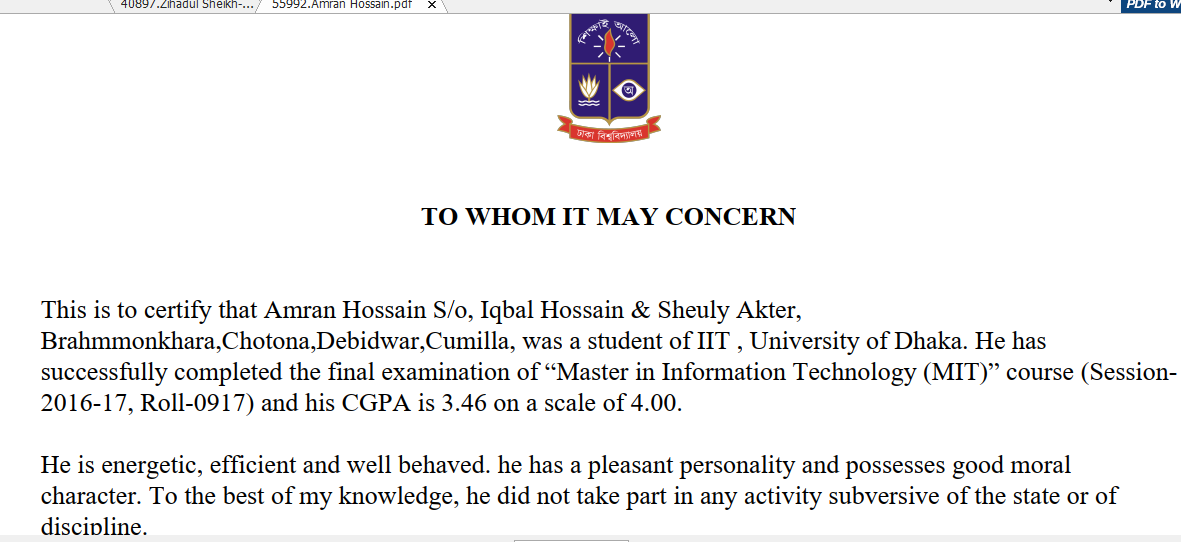


Figure 46: Testimonial process

Chapter 11: Testing Component

# 11.1 Introduction

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is[defect](https://www.guru99.com/defect-management-process.html)free. Actually, there are two types of testing. Black box testing and White box testing. I have used black box testing for this software.

# 11.2 Software Testing Goals

Most essential software testing goals are described in here,

* Bug detection: QA experts detect and root out bugs and malfunctions before customers find them. It’s a short-term objective that requires a scrutinous approach which can be provided by the manual software testing.
* Bug prevention: The bug-prevention objective is superior to others and implies not only anticipation but also prevention of defects from recurring in the future. In the long run, bug prevention helps to shorten the product time to market, reduce the cost of software quality maintenance and increase the customer satisfaction and loyalty to your product.
* User satisfaction: The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.
* Software quality and reliability: It is an essential requirement of any software product. Testing ensures a quality and reliable product is delivered to customers.
* Cost effective**:**It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.
* Security: It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.

# 11.3 Levels of Testing

Here are important strategies in levels of testing in software engineering:

* **Unit Testing:** This software testing approach is followed by the programmer to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not.
* **Integration Testing:** It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.
* **System Testing:** In this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.
* **Acceptance Testing**: Checks the requirements of a specification or contract are met as per its delivery.

# 11.4 Testing Procedure

Following test cases have been performed on this system. To run these cases, internet must be in  
working condition.

Table 7: Test case

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Case** | **Input Test Data** | **Steps to be Executed** | **Expected Result** | **Actual Result** | **Pass/ Fail** |
| T1 | Test if application successful | Applicant personal details | 1. Input required field 2. Click “Sign Up” | Application successfully submitted | Application Successful | Pass |
| T2 | Test if applicant log in works | Email and applicant id | 1. Input required field 2. Click “Search Status” | Successfully logged in | Successfully logged in | Pass |
| T3 | Test if payment completed | Details for payment | 1. Select “Pay now”  2. Given input data.  3. Choose payment option. | Payment completed | Payment completed | Pass |
| T4 | Test if notification received | Application data | 1.Logged in to search status.  2. Check email | Notification received | Notification not received | Pass |
| T5 | Test if admin log in works | Email and password | 1. input required field 2. Click “Log in” | Successfully logged in | Successfully logged in | Pass |
| T6 | Test if password reset worked | Previous password, current password and email. | 1.Click password reset option  2. Give Email | Password reset successfully | Password reset successfully | Pass |
| T7 | Test if application approval worked | Application | Click approve button | Application approved | Application approved | Pass |
| T8 | Test if application cancel worked | Application | Click cancel button | Application cancels | Application cancels | Pass |
| T9 | Test if testimonial/certificate is ready | Application | Click download button | Document is ready | Document is ready | Pass |
| T10 | Test if department added in system | Department name and faculty name | Click add button | Department added successfully | Department added successfully | Pass |

# 11.5 Risk and Contingencies

There will some risks when using this system. Transaction process will be fall down for occurring internet problem or payment gateway company’s internal server error occurs. For successfully application start to end user have to fulfill user credentials as needed carefully. Otherwise, system may not work properly.

Chapter 12: User Manual

# 12.1 Application run

It is a web application. For run the application locally we need xampp control panel. Laravel installation, MySQL database and an IDE. To run the application just write “php artisan serve” in command line.

If this application host in internet just open the browser and hit the url.

# 12.2 Use the Application

Firstly, after hitting the url we get the homepage of the website and select services from the navbar what services do we want. Then click the button on get testimonial or get certificate and find an application form. Fill up the form accurately and submitting this get a unique applicant id which will be needed for log in this website for applicant. Then click the pay now button and get payment form and pay the money according to procedure.

After applicant complete his application to know his application status, he/she may click the application status button and give email and applicant id. After log in applicant see the status either admin accept or reject the application. Also, applicant know his status through the email. Also, applicant pay after login the system not in sign up time.

For Admin panel admin must log in with giving email and password. Also, admin may recover password also by clicking forgotten account. After log in the system admin see the applicant list who are apply for the certificate or testimonial. Admin accept or reject the application after click those buttons. Admin may download the testimonial and certificate copy. Admin click the approve button and change the status automatically. System gives email notification to the applicant. Also, admin add, delete or edit department, language and also director as much needed.

Chapter 13: Conclusions and Future Work

13.1 Conclusion:

Working with DU Services system was a new experience for me. I got the opportunity to learn about building a website. I have learnt how transaction system added in a system. I have tried my level best to implement the system as flawlessly as possible. However, there is still room for improvement. In future, I will try to host it in the server. Also, I have been experienced with web development work by building this system. Now I will be able to build any kind of website.

# 13.2 Future Work:

For transaction process of this system, I used sandbox sslcommerz free api. I will do it real time transaction in next. Also, I will host this website into internet. Now it is in my localhost. Also, I will add any other services system in automatically by this system which are related to transaction system. I will build this more user friendly so that user will be happy after using this system. In future I will try to add machine learning in this system so that system works with more data within a shorten time.

References

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.
4. <https://www.qamadness.com/what-is-the-ultimate-goal-and-key-objectives-of-software-testing/#:~:text=Serving%20as%20a%20bridge%20between,quality%20of%20a%20resulted%20product>. Last access 19 February 2021 at 9:30 a.m.
5. <https://www.guru99.com/software-testing-introduction-importance.html> . Last access 19 February 2021 at 9.45 a.m.
6. <https://www.guru99.com/levels-of-testing.html>.Last access 19 February 2021 at 9.45 a.m.
7. [https://creately.com/blog/diagrams/deployment-diagram tutorial/#:~:text=A%20deployment%20diagram%20is%20a,and%20software%20of%20a%20system](https://creately.com/blog/diagrams/deployment-diagram%20%20tutorial/#:~:text=A%20deployment%20diagram%20is%20a,and%20software%20of%20a%20system). Last access 26 March 2021 at 12:30 a.m.
8. <https://laravel.com/docs/4.2/introduction> Last access 26 March 2021 at 10:30 p.m.
9. <https://github.com/sslcommerz/SSLCommerz-Laravel> Last access 26 March 2021 at 10:34 p.m
10. <https://www.positronx.io/laravel-pagination-example-with-bootstrap-tutorial/> Last access 02 April 2021 at 07:20 p.m.