Software Requirements Specification

Version 1.1

Submission Date: - 28-01-2016

**Fees Management System**

Submitted in partial fulfillment

Of the requirements of

CS 223 Software Engineering

Karteek Venkata - UG201213020

Shivam Choudhary - UG201213034

Sunil Suthar - UG201213036

Vibhav Sharma - UG201213037

.

# Table of Contents

[Table of Contents](#h.1fob9te)

[List of Figures](#h.2et92p0)

[1.0. Introduction](#h.tyjcwt)

[1.1. Purpose](#h.3dy6vkm)

[1.2. Scope of Project](#h.1t3h5sf)

[1.3 Constraints](#h.4d34og8)

[1.4 Assumptions and Dependencies](#h.2s8eyo1)

[1.5 Glossary](#h.17dp8vu)

[1.6. References](#h.3rdcrjn)

[1.7. Overview of Document](#h.26in1rg)

[2.0. Overall Description](#h.lnxbz9)

[2.1 System Environment](#h.35nkun2)

[2.2 Functional Requirements Specification](#h.1ksv4uv)

[2.2.1](#h.44sinio) Use case Diagram of Student

2.2.2 Use Case Diagram of Administrator

[2.3 User Characteristics](#h.z337ya)

[2.4 Non-Functional Requirements](#h.3j2qqm3)

[3.0. Requirements Specification](#h.1y810tw)

[3.1 Functional Requirements](#h.4i7ojhp)

[3.1.1](#h.2xcytpi)User login

[3.1.2](#h.1ci93xb) View student’s basic info

[3.1.3](#h.1ci93xb) View fee structure

[3.1.4](#h.1ci93xb) Fee payment

[3.1.5](#h.1ci93xb) Complaint addressal

[3.1.6](#h.1ci93xb) Update student details

[3.1.7](#h.1ci93xb)search query

[3.1.8](#h.1ci93xb) DD approval

[3.1.9](#h.1ci93xb) Update fee structure

[3.3 Detailed Non-Functional Requirements](#h.3whwml4)

[***3.4*** ***Logical Structure of the Data***](#h.2bn6wsx)

[4.0 Supporting information](#h.qsh70q)

[4.1 Table of contents and index](#h.3as4poj)

[4.2 Appendixes](#h.1pxezwc)

**Document Version History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 17​th​ January, 2016 | Version 1.0 | Scope, Requirements and Use case Tables were added |
| 28​th​ January, 2016 | Version 1.1 | Use case Diagrams, Class  Diagrams, Sequence and Activity diagrams were added |

# List of Figures

|  |  |
| --- | --- |
| **Figure number** | **Description** |
| **2.2.0 (a)** | **Student Use Case Diagram** |
| **2.2.0 (b)** | **Admin Use Case Diagram** |
| **2.2.1(a)(b)** | **User Login Class Diagram, Sequence Diagram** |
| **2.2.2(a)(b)** | **View Student’s Basic Info Class Diagram, Sequence**  **Diagram** |
| **2.2.3(a)(b)** | **View Fee Structure Class Diagram, Sequence Diagram** |
| **2.2.4(a)(b)** | **Fees Payment Class Diagram, Sequence Diagram** |
| **2.2.5(a)(b)** | **Complaint Addressal Class**  **Diagram, Sequence Diagram** |
| **2.2.6(a)(b)** | **Search Query Class**  **Diagram, Sequence Diagram** |
| **2.2.7(a)(b)** | **Demand Draft Approval Class Diagram, Sequence Diagram** |
| **2.2.8(a)(b)** | **Update Fees Structure Class**  **Diagram, Sequence Diagram** |

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to provide a SRS of FMS (*Fees Management system*) for an institution.

SRS is the Software Requirements Specification that provides a complete description of software. It describes what the software will do without demonstrating how the software will do it.

SRS specifies the requirements of FMS, description of its features, restriction and limitations of the software and all the functional and nonfunctional requirements of the project.

Functional requirements describes the functions and the services you would expect the system to provide.

Nonfunctional requirements are not directly related to the essential function of the system, they are global qualities of the system, like usability, performance efficiency, cost etc.

## 1.2. Scope of Project

This software system will be designed to implement fees management system for an institution that can help both staff members and student.

For students it will ease their semester mess and other fee transactions. Using this software, a student can easily pay their semester, mess or hostel fee and see the transaction without any difficulty and at the same time can address its query.

For staff members it will help them track each and every details of all the student’s fee transaction. They can check status of every student if he has paid the fees or not.

Its multi functionality and cross-platform availability makes it a very useful tool.

## 1.3 Constraints

***1.3.1 User interface constraint***

Using this system is fairly simple and intuitive. A user familiar with basic browser navigation skills should be able to understand all functionality provided by the system.

***1.3.2 Hardware Constraints***

The system should work on most home desktop and laptop computers which support JavaScript and HTML5.

***1.3.3 Software Constraints***

The system will be intended to run on Firefox 4 and above, Google Chrome 10 and above and Internet Explorer 8 and above.

## 1.4 Assumptions and Dependencies

1. User should be accustomed with working on web browser.
2. Only certain features are implemented which necessary for the users.
3. Users can submit fee via three modes – debit card, credit card or demand draft.
4. Registering new students feature is not added yet.

## 1.5 Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| FMS | Software meant to help to facilitate fee management and payment |
| User | Two category of users- students and office staff |
| DD | Demand Draft |

## 1.6. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.7 Overview of Document

The rest of the document is divided into chapters for better understanding.

* In chapter 2 an overall description of FMS is provided. First product perspective is presented with product features and main functions. Then follow user classes and characteristics, operating environments that FMS supports as well as design and implementation constraints. After all that user documentation is presented and will provide you with more details about each feature technology.
* In chapter 3 most important features are presented with detailed description.
* In chapter 4 user and communication interfaces are described.
* In chapter 5 requirements about safety and performance are presented.

***2.0. Overall Description***

***Product Perspective***

FMS is meant to provide a convenient platform for fee payment and management for all students and administrators. It will have a simple and easy user interface which can be used very comfortably. It can be used in all operating systems without compatibilty problems.

***Product Features***

FMS software is used for fees collection in college. Its main features are separate login for students and administrators. It gives details for all students about their fee structure and administrators to check all the details about student’s fee details (paid/unpaid).

Student features include their profile and payment through various methods like online payment (net banking) and payment through demand drafts(DD), after payment receipt will be updated in their profile.

Administrator features include each student name and their payment details and they have access to make changes to payment details that is changing paid to unpaid or vice versa and they can make DD approvals also they can check students errors regarding fee payments.

## 2.1 System Environment

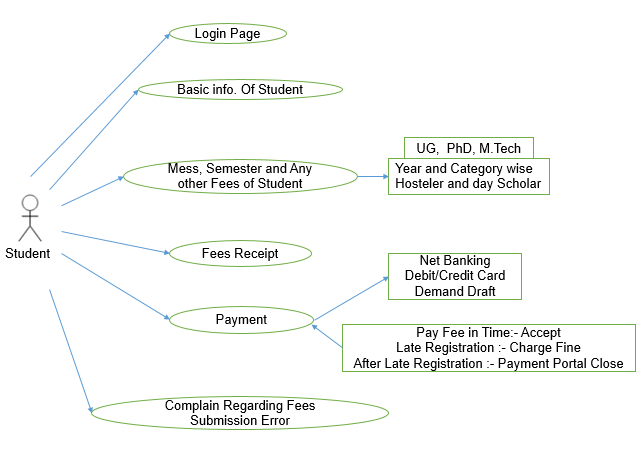
## 2.2 Functional Requirements Specification

### 

#### 

#### Use Case Diagram of Student

**Diagram:**



**Fig 2.2.0 (a) Use Case Diagram of Student**

**Brief Step By Step Description**

1. Login for students will be provided.

2. Basic information regarding student will be provided (Name, Year, Room No, CPI etc.)

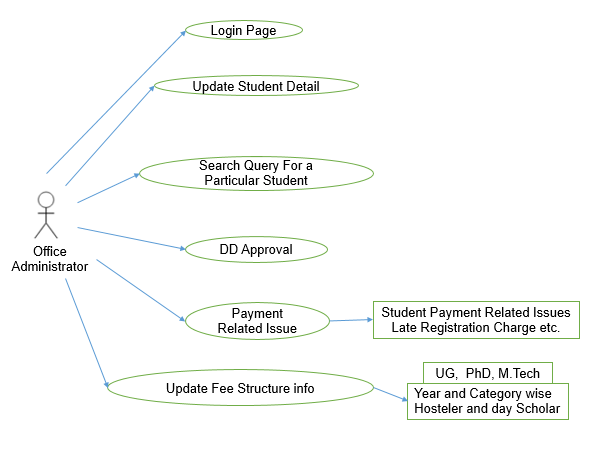
3. Fee structure will be shown for all students.

4. Various Payment methods will be provided for students (online payment, DD details).

5. Receipt wil be updated in their profile after successful transaction.

Use Case Diagram of Administrator

**Diagram:**



**Fig 2.2.0 (b) Use Case Diagram of Student**

**Brief Step By Step Description**

1. Login for administrator will be provided.

2. Student details will be provided which can be edited by the administrator.

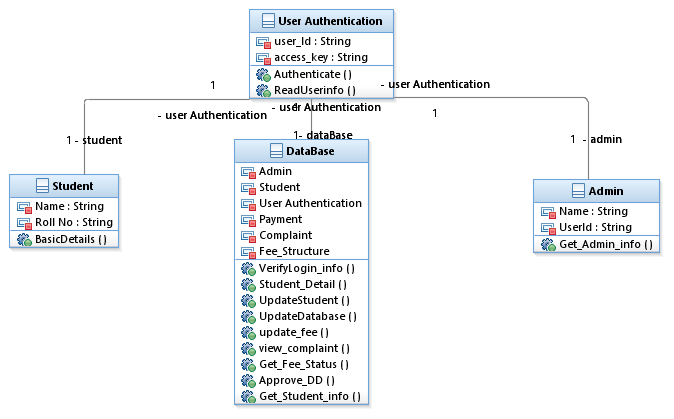
3. In case of DD after he receives DD hardcopy he can approve fee payment by checking details of DD in student profile.

4. They also receive errors of various payment related issues which can be debugged by them.

5. They also have privilege to update Fee structure of all students.

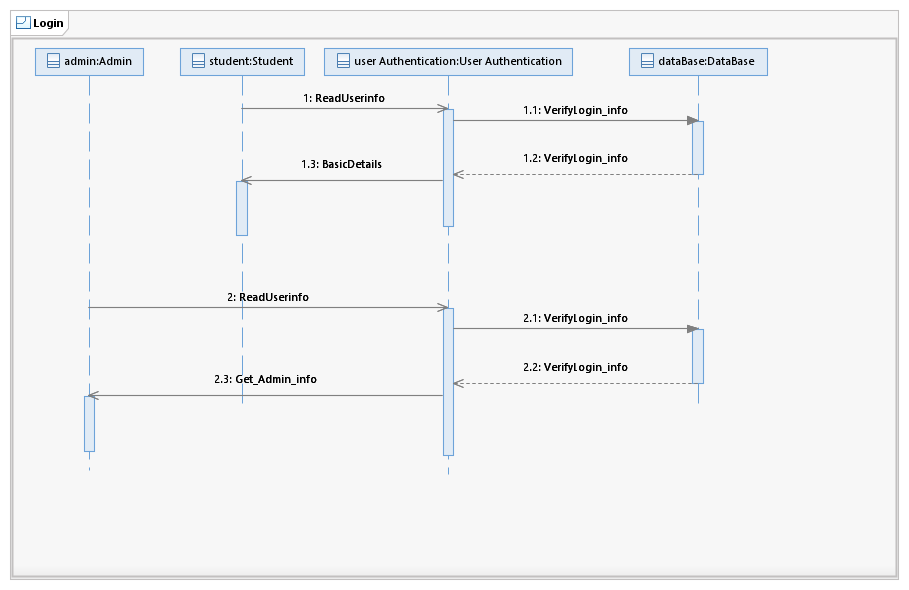
### ***2.2.1***

**Use Case: -User Login**

**Diagram :-   
**

## Fig 2.2.1(a): Class diagram for User Login use case

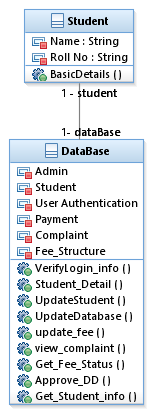
**Sequence Diagram:**



## Fig 2.2.1(b): Sequence diagram for User Login use case

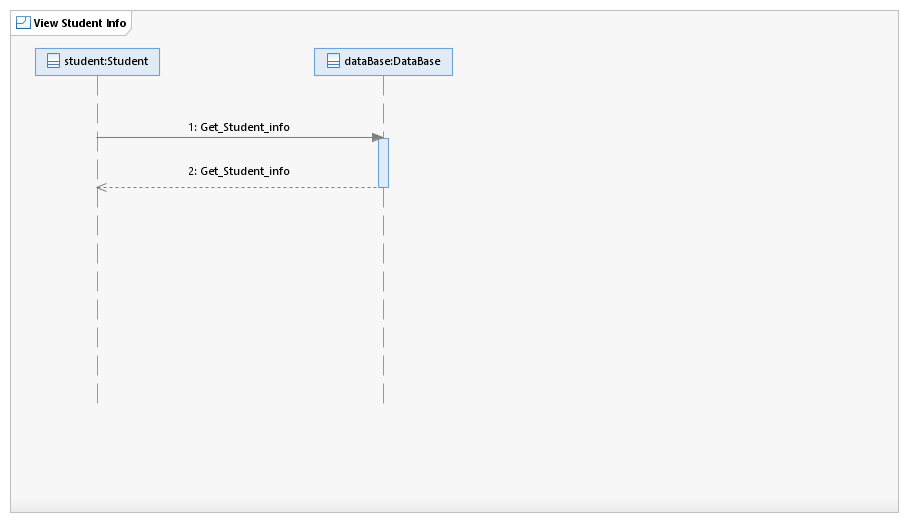
### ***2.2.2***

**Use Case: - View Student’s Basic Info**

**Diagram :-   
 **

**Fig 2.2.2(a): Class diagram for View Student Basic Info use case**

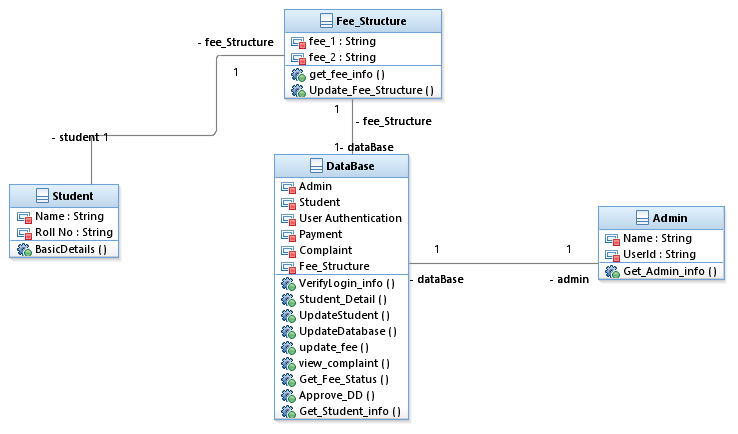
**Sequence Diagram:**



**Fig 2.2.2(b): Sequence diagram for View Student Basic Info use case**

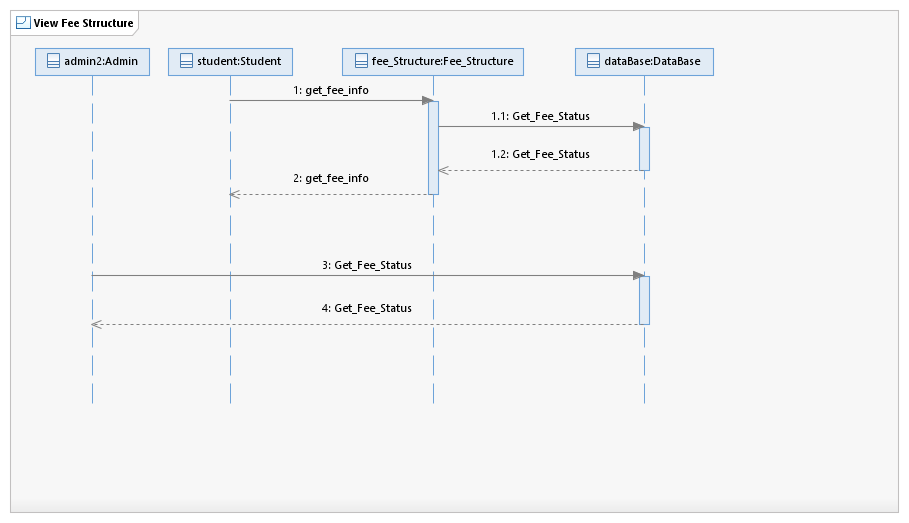
### ***2.2.3***

**Use Case: - View Fee Structure**

**Diagram :-   
**

**Fig 2.2.3(a): Class diagram for View Fee Structure use case**

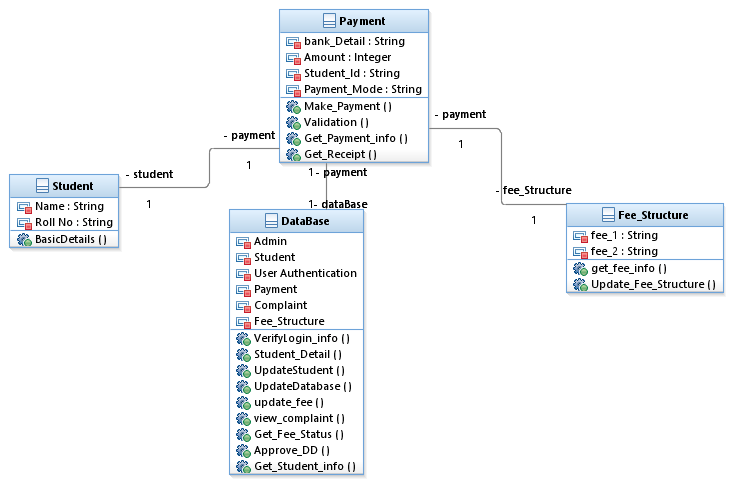
**Sequence Diagram:**

****

### **Fig 2.2.3(b): Sequence diagram for View Fee Structure use case**

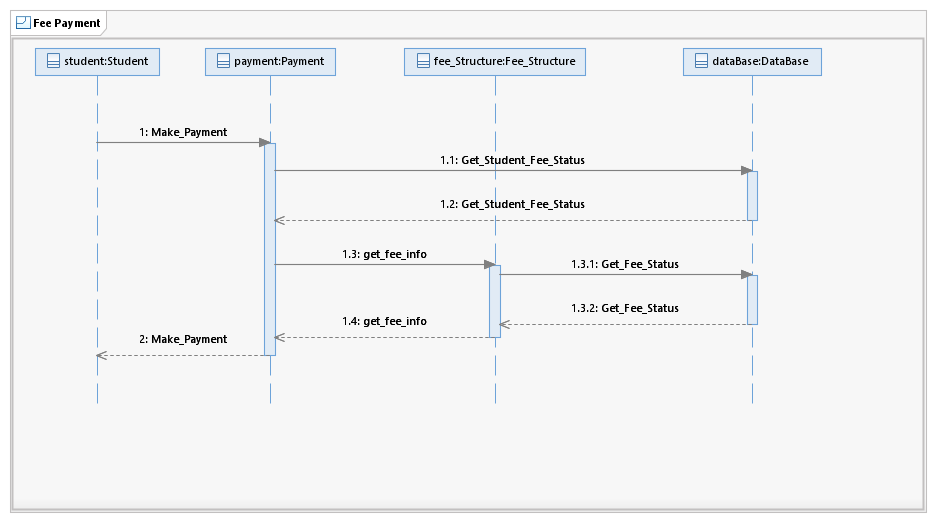
### ***2.2.4***

**Use Case: - Fees Payment**

**Diagram :-   
**

**Fig 2.2.4(a): Class diagram for Fee Payment use case**

**Sequence Diagram:**

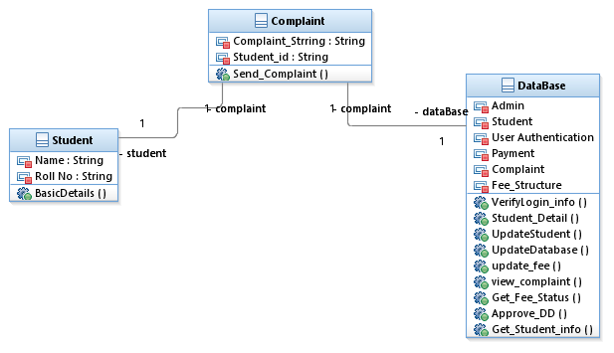
****

**Fig 2.2.4(b): Sequence diagram for Fee Payment use case**

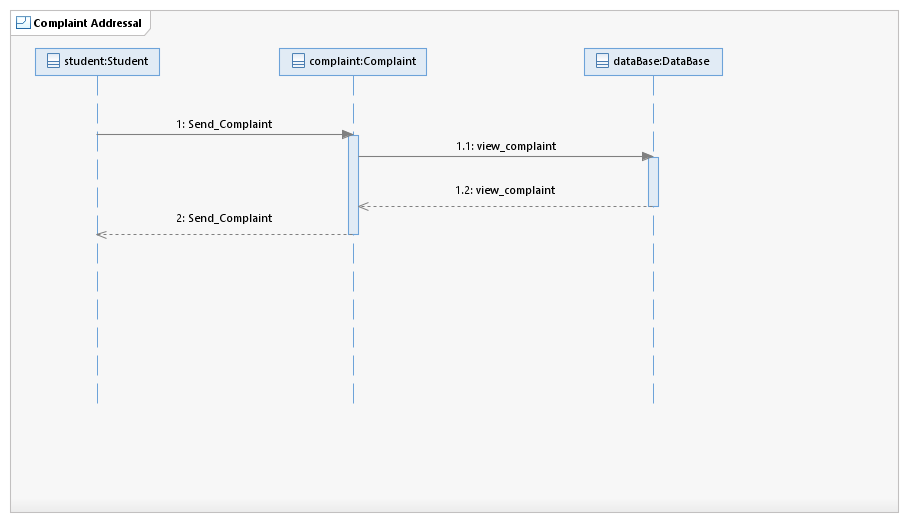
### ***2.2.5***

**Use Case: - Complaint Addressal**

**Diagram :-**

**  
 Fig 2.2.5(a): Class diagram for Complaint use case**

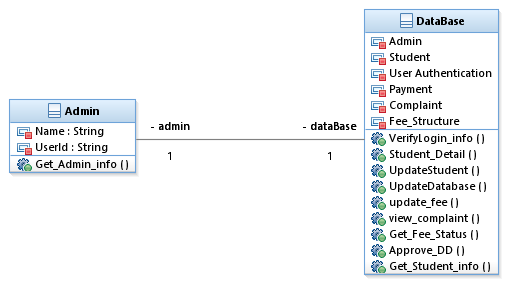
**Sequence Diagram:**

****

**Fig 2.2.5(b): Sequence diagram for Complaint use case**

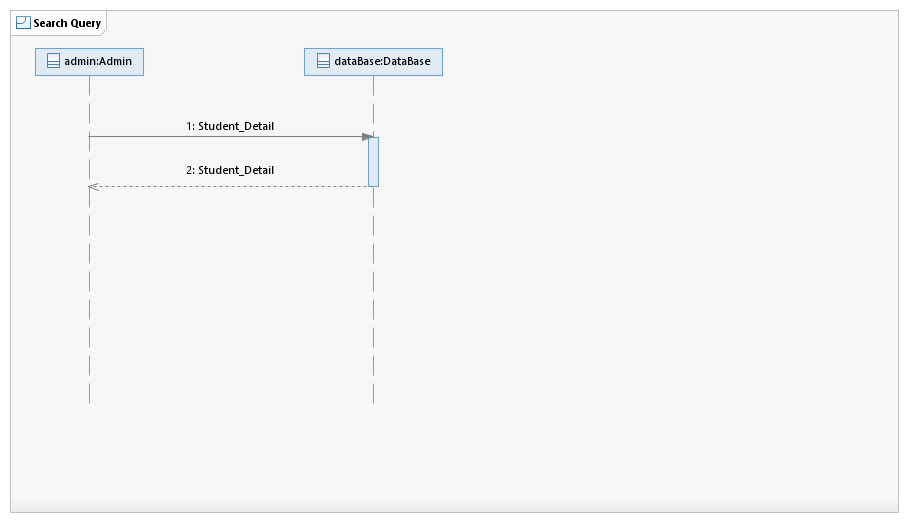
### ***2.2.6***

**Use Case: - Search Query**

**Diagram :-   
**

**Fig 2.2.6(a): Class diagram for Search Query use case**

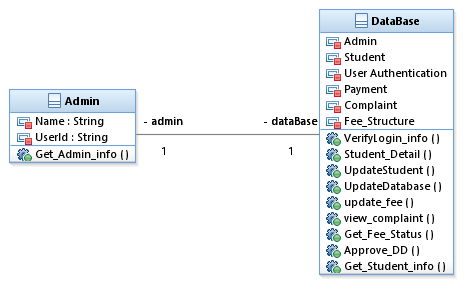
**Sequence Diagram:**

****

**Fig 2.2.6(b): Sequence diagram for Search Query use case**

### ***2.2.7***

**Use Case: - Demand Draft Approval**

**Diagram :-   
**

**Fig 2.2.7(a): Class diagram for DD Approval use case**

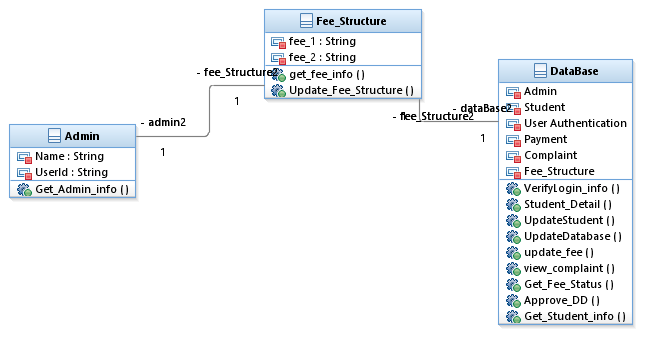
**Sequence Diagram:**

****

**Fig 2.2.7(b): Sequence diagram for DD Approval use case**

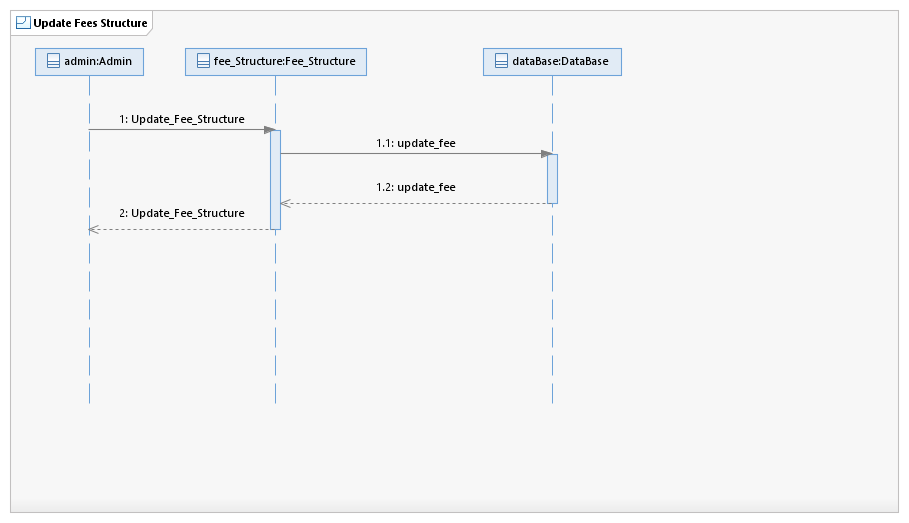
### ***2.2.8***

### **Use Case: - Update Fees Structure**

**Diagram :-   
**

**Fig 2.2.8(a): Class diagram for Update Fee Structure use case**

**Sequence Diagram:**

****

**Fig 2.2.8(b): Sequence diagram for Update Fee Structure use case**

## 2.3 User Characteristics

***2.3.1 Students***

Students are the primary consumers of this system. They are accessing information regarding fee structure and pay according to it.

***2.3.2 Administrators***

System administrators are primarily responsible for maintaining the FMS. They spend more time modifying the system’s configuration and making appropriate updates of fee structure and debug the errors regarding fee payment caused by students.

**2.4 *Non-Functional Requirements***

The FMS will be based on server with high speed internet. The system must be

Interactive with less number of delays. If the internet service gets disrupted while sending information to the server, the information can be send again for verification.

# 3.0. Requirements Specification

## 3.1 Functional Requirements

### 3.1.1 **User Login**

|  |  |
| --- | --- |
| **Use Case Name** | User Login |
| **Trigger** | User logs into system using existing profile. |
| **Basic Path** | 1. 1User id 2. Password 3. Type (Student or Office Administrators) |
| **Alternative Paths** | Invalid password, invalid username, or mismatched username and password redirect to error message and previous page. |
| **Precondition** | User is not logged in to a profile, input profile exists in database, user password matches profile |
| **Post condition** | User's computer has been supplied with appropriate cookie, page data is appropriate for selected profile |
| **Other** | This allows users to log in to their profile from anywhere. |

### 3.1.2 **View Student’s Basic Info**

|  |  |
| --- | --- |
| **Use Case Name** | View student’s basic info |
| **Trigger** | student decides to view his updated information |
| **Precondition** | student’s information is present in the database |
| **Basic Path** | click on tab of basic info |
| **Post condition** | student is able to see his information |

### 3.1.3 **View Fee Structure**

|  |  |
| --- | --- |
| **Use Case Name** | View Fee Structure |
| **Trigger** | Student wants to view fee information |
| **Precondition** | Particular Students Fee info is updated using a Fee Structure Table |
| **Basic Path** | 1. Category  2. Year  3. UG or PG |
| **Post condition** | User's computer has been supplied with appropriate cookie, page data is appropriate for selected profile |
| **Other** | This allows users to log in to their profile from anywhere. |

### 3.1.4 **Fees Payment**

|  |  |
| --- | --- |
| **Use Case Name** | Fees Payment |
| **Trigger** | Student wants to pay tuition fees and dues etc.. |
| **Precondition** | Particular Students Fee info is updated using a Fee Structure Table |
| **Basic Path** | 1. Payment Mode (Net Banking or Debit Card )  2. Bank Name 3. Gateways |
| **Post condition** | Payment Done and Fee Receipt Generated |
| **Other** | This allows users to pay fee from anywhere. |

### 3.1.5 **Complaint Addressal**

|  |  |
| --- | --- |
| **Use Case Name** | Complaint addressal |
| **Trigger** | Student not satisfied with the system or find a faulty error or fees related issues |
| **Basic Path** | 1. user login 2. complaint box |
| **Alternative Paths** | Invalid password, invalid username, or mismatched username and password redirect to error message and previous page. |
| **Precondition** | User is logged in to a profile, input profile exists in database, user password matches profile |
| **Post condition** | Complaint dialog box will automatically appear and complaint will directly delivered to the mail-box of admin |
| **Other** | This allows users to log in to their profile from anywhere. |

### 3.1.6 **Update Student Detail**

|  |  |
| --- | --- |
| **Use Case Name** | Update Student Detail |
| **Trigger** | Administration Office Update the student details |
| **Precondition** | Particular Students Details is required, Particular update is required by student |
| **Basic Path** | 1. Student Profile Search  2. Edit Profile 3. Save Edited Information |
| **Post condition** | Updated Student Details will be available to the student |

### 

### 3.1.7 **Search Query**

|  |  |
| --- | --- |
| **Use Case Name** | Search Query |
| **Trigger** | Admin wants to search for a particular Student or Particular Batch |
| **Basic Path** | 1. Admin Log in.  2. Enter Keyword in Search Box.  3. Get the Search Result |
| **Precondition** | Admin is logged in. |
| **Post condition** | User's computer has been supplied with appropriate cookie, page data is appropriate for selected profile |
| **Other** | This allows admin to Search profile from anywhere. |

### 3.1.8 **Demand Draft Approval**

|  |  |
| --- | --- |
| **Use Case Name** | Demand Draft Approval |
| **Trigger** | Student want to submit fee via demand draft. |
| **Basic Path** | 1. admin login 2. student search 3. payment approved after submission of DD |
| **Precondition** | student should have correctly uploaded the DD details |
| **Post condition** | Complaint dialog box will automatically appear and complaint will directly delivered to the mail-box of admin |
| **Other** | This allows users to log in to their profile from anywhere. |

### 

### 

### 

### 3.1.9 **Update Fees Structure**

|  |  |
| --- | --- |
| **Use Case Name** | Update Fees Structure |
| **Trigger** | Fees structure will be updated according to updated fees norms batch wise |
| **Basic Path** | 1. Admin login  2. Update fees structure |
| **Precondition** | Approved fee structure details to be updated for payment |
| **Post condition** | Updated fees structure will be visible to all the students |
| **Other** | This allows admin to Search profile from anywhere. |

### 

## 3.3 Detailed Non-Functional Requirements

### ***3.4 Logical Structure of the Data***

# 4.0 Supporting information

## 4.1 Table of contents and index

## 4.2 Appendixes