IEEE STANDARD 1016: Software Design Specification
University Management System (UMS)

Hala Ali Khan_007
Mahrukh_037

IEEE STANDARD 1016: Software Design Specification

The Software Design Specification

1. Introduction

1.1 Purpose of this document

The purpose of this document is to present the software of University Management System by describing several viewpoints and objectives. The document explains implementation of the software, and the steps followed in its implementation. It describes the main stakeholders and the activity performed. Architecture design is discussed along with design decisions and tradeoffs. The purpose of establishing this system is clearly stated in the document and the steps to use the component and user persona would help in easy use of the system.

1.2 Scope of the development project

The main features of the software include:

- User will be able to login
- User will be able to maintain record of attendance, teachers and students
- It allows admin and admission office to modify or remove teachers and students information
- It allows communication between admin and accounts
- It allows to schedule admission and select eligible students.

Main Stakeholders include:

- Admin
- Admission Office
- Teacher
- Student
- Accounts
- Software Programmers
- Software Architects

1.3 Definitions, acronyms, and abbreviations

SDS: Software Design Specification

UMS: University Management System

1.4 References

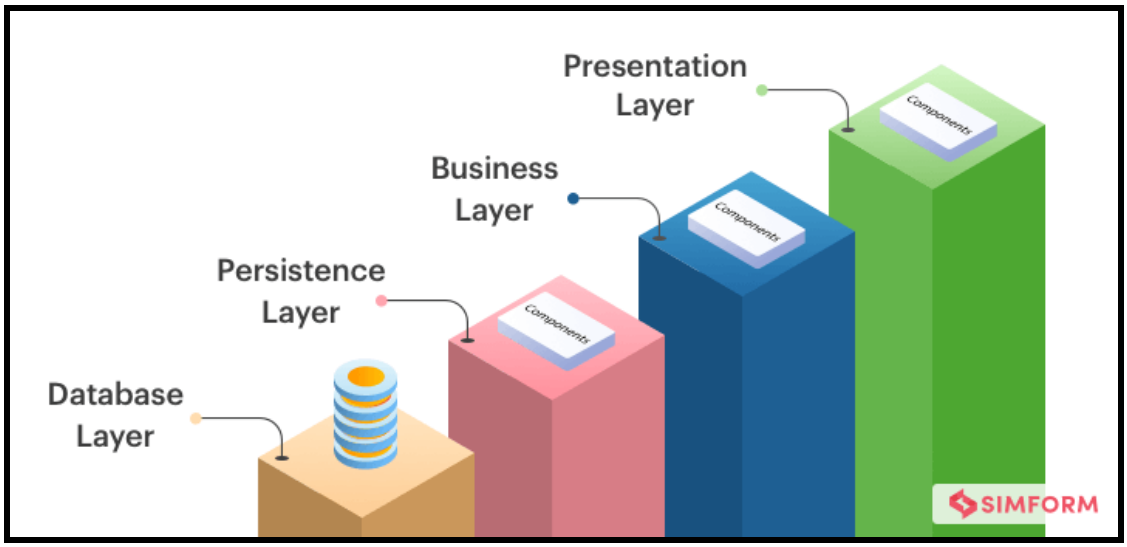
- Richards, M. (2015). *Software architecture patterns* (Vol. 4). 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Incorporated.
- Nielsen, L. (2013). *Personas-user focused design* (pp. 59-79). London: Springer.

1.5 Overview of document

This document includes overview of the working of the application system. It shows the arrangement of components their interaction with each other. Relationship and multiplicities of the components. Detailed diagrams and classes are shown to give a clear picture to give an idea.

After classes and diagrams personas and architectural designs are described. Pseudocode of components describes the path followed for performing a task.

2. System architecture description



Most of the software applications use layered architecture because it is maintainable and testable. It provides different layers, hence increases the security of the system. The data required by any of the layers has to go through all the layers that come in between. Moreover, it is beneficial because each layer contain the relevant data and can stand independently, this means that if the modification is required in any of the layer it won't affect other layers significantly.

In our application software it is used majorly because of the reasons stated above, in order to ensure privacy and security of the sensitive and

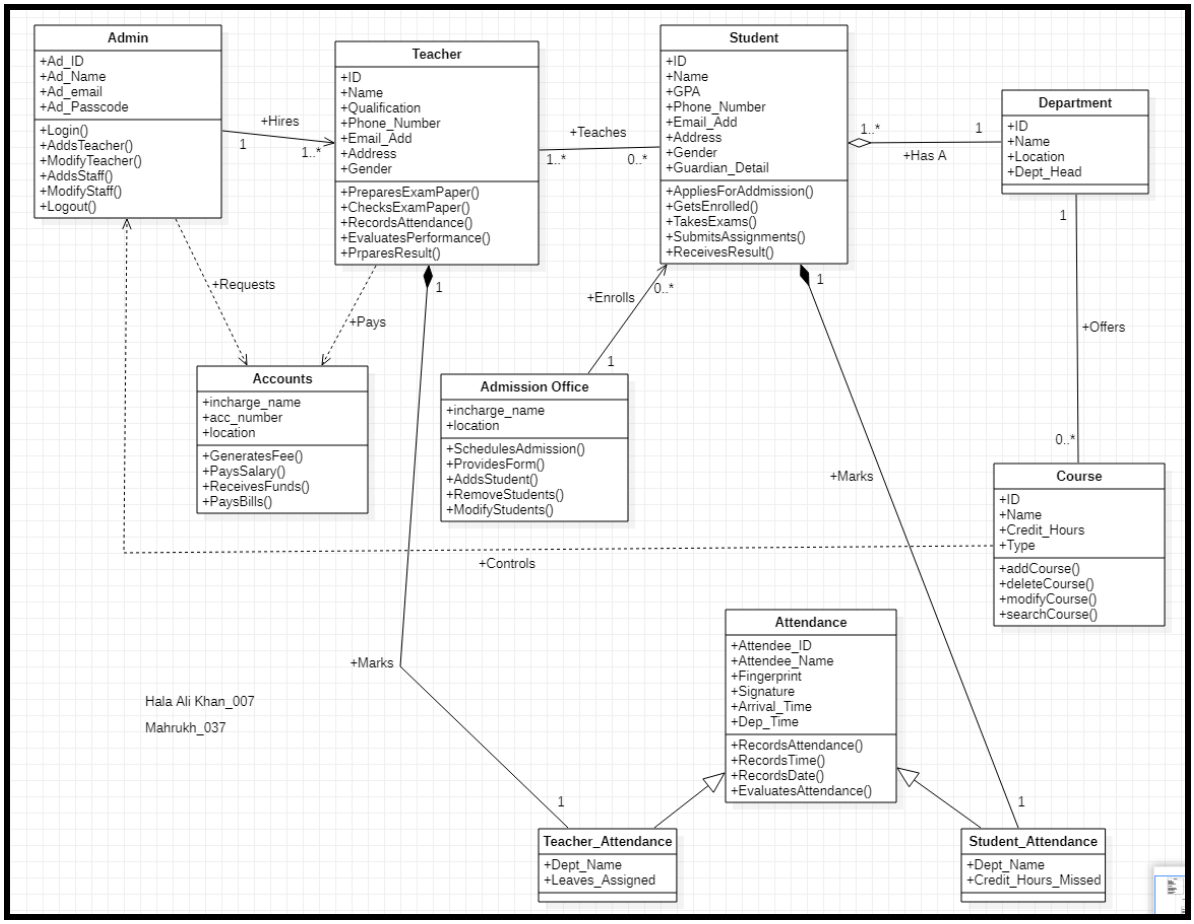
confidential information this architecture design best suits the system. Database layer would also help in storing large amounts of data therefore it is most suitable architecture design that can be used.

2.1 Overview of modules / components

The components used in this system are as follows:

- Admin: controls all the major issues, including hiring of faculty, generating pay slips, confirming admissions etc.
- Accounts: handles all the financial issues, including paying bills, salary etc.
- Admission Office: Generates information related to admission.
- Teacher: Interacts with students, prepares exams, result and records attendance.
- Student: Takes Admission in the specific course, takes exams and pays fee
- Course: this component can add, delete or modify course
- Attendance: it records attendance of students and teachers. Record off Student attendance is sent to teachers whereas record of teachers attendance is sent to the admin

2.2 Structure and relationships



Admin:

Admin and Teacher

Relationship: Directed Association because Admin hires Teacher

Multiplicity: one and one-to-many, because one sector of admin can hire one or more teachers.

Admin and Accounts:

Relationship: Dependency because admins are dependent on accounts to pay salary and bills.

Multiplicity: Due to dependency there is no as such multiplicity

Admission Office:

Admission Office and Student

Relationship: Directed Association because Admission Office enrolls students

Multiplicity: one and zero-to-many, because one office can enroll zero or many students.

Accounts:

Accounts and Teacher

Relationship: Dependency, because teachers depends on accounts for getting the salary

Multiplicity: Due to dependency there is no multiplicity.

Department:

Department and Student

Relationship: Aggregation, because department has students. Moreover, if we destroy the student class, department can still stand alone

Multiplicity: one and one-to-many, one department can have one or more students.

Department and Course

Relationship: Association, department offers course

Multiplicity: one and zero-to-many, because one department may offer zero or more courses

Teacher

Teacher and Student

Relationship: Association

Multiplicity: one-to-many and zero-to-many because one or more teachers can teach either zero or more students

Attendance:

Attendance have been generalized as student attendance and teacher attendance.

Attendance and Teacher

Relationship: Composition, if teacher class is destroyed attendance class will also be destroyed

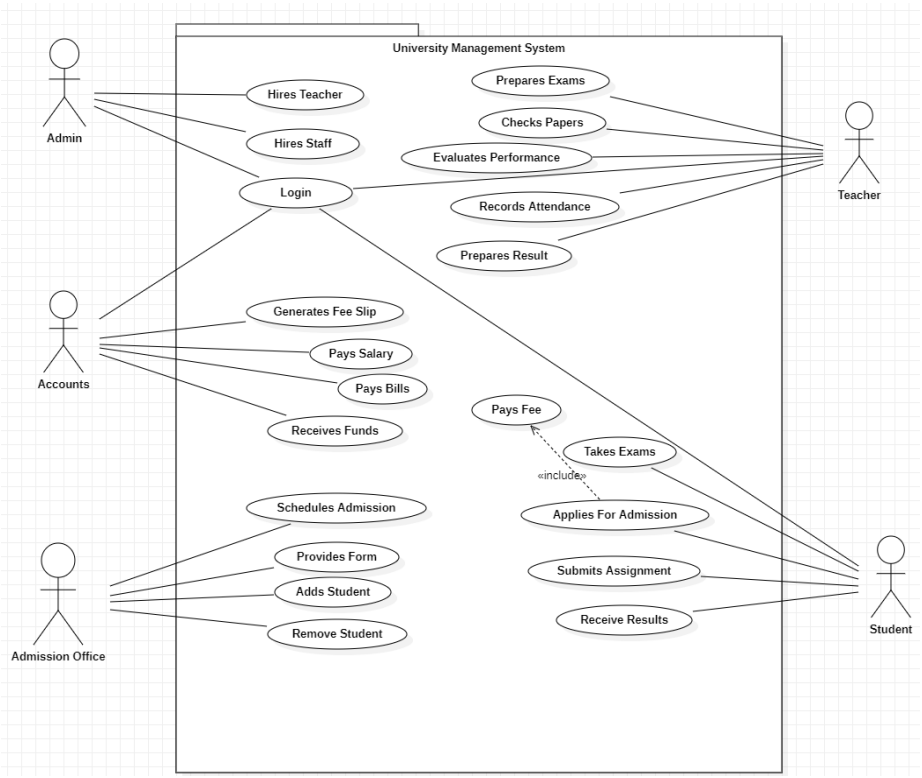
Multiplicity: one and one, a teacher can mark attendance only once at a time

Attendance and Student

Relationship: Composition, if student class is destroyed attendance class will also be destroyed

Multiplicity: one and one, a student can mark attendance only once at a time

2.3 User interface issues



Student Persona:

Sarah is a 19 years-old girl. She just completed her Fsc. She is looking forward to take admission in university. She login on the student portal of the university and downloads the admission form, fills it and submit it. After a week she receives a mail that she has been selected as eligible student of the university. She submits the admission fee. Then receives a confirmatory mail that her fee has been received and gets an admit card for the university.

Teacher Persona:

Helen is a qualified teacher. She is selected as a professor in the university. She prepares exam for her students. Login on the faculty portal, submits paper to examination office. After exam have been conducted she receives the papers and starts preparing the result. After preparing the results she submits it on the examination portal again by logging in on the website.

3. Detailed description of components

3.1 Component description

The components used in this system are to support the UMS and to organize and maintain the data. Interaction between different components is shown. Flow of the activity helps in tracing the flow of the tasks. The system formed is flexible and maintainable, it is also testable. It helps keeping the record well in form and accessible. Moreover, official emails and passwords would be allowed to login the system. The major components formed in the system are listed below:

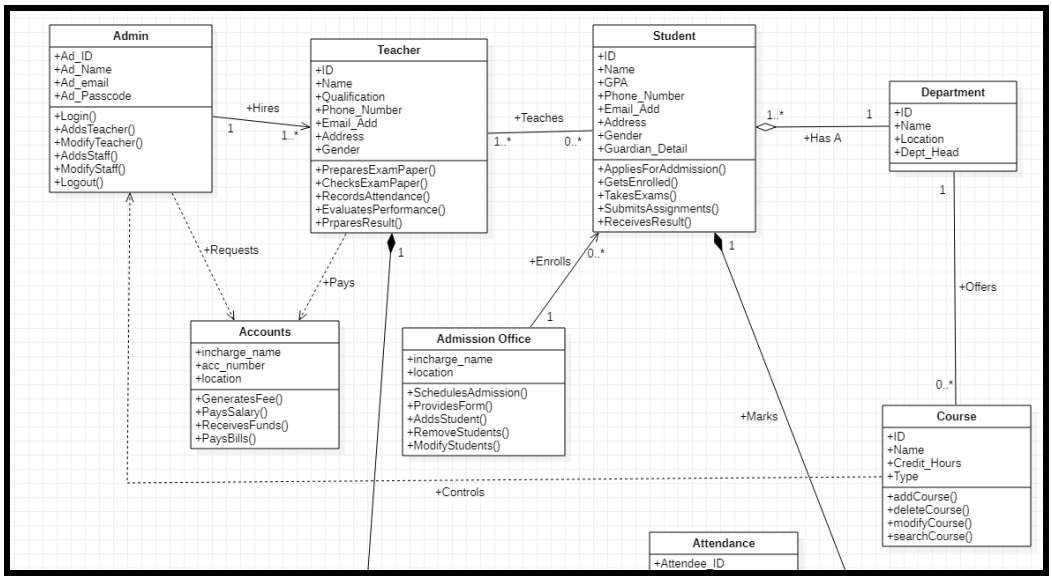
- Admin
- Admission Office
- Accounts

3.2 Admin Component (or Class or Function ...)

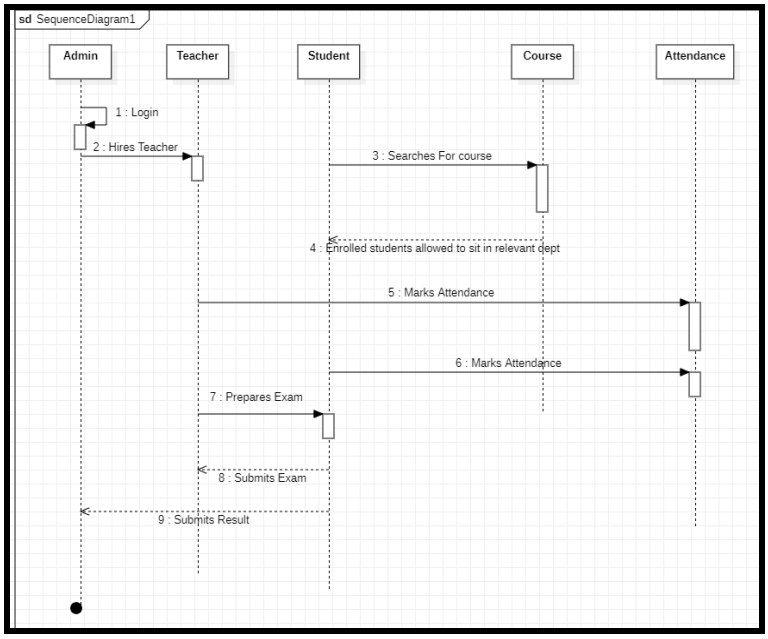
Identification	Admin is the primary actor in the system.

Type	A module and a class
Purpose	It controls the hiring of teachers and faculty.
Function	Hires Teachers Modifies Teachers Hires Staff Modifies Staff Login/Logout
Subordinates	Not Applicable
Dependencies	Depends on Account to pay Salary to teachers and staff
Interfaces	The interface used is user-friendly and user-focused. While logging in if any error takes place the authentication error is prompted. Moreover, if any database error occurs database error message is prompted
Resources	Resources used are internet, computer, human resource, authorized email and password
Processing	Initiates the login process Hires Teachers/Staff Sends request to pay salary to account Receives confirmation of salary Receives confirmation of fee submission

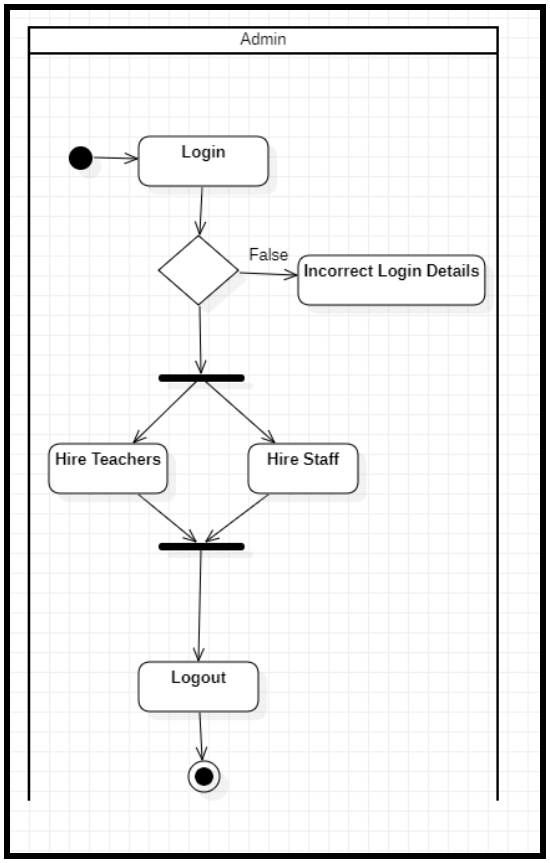
Data	Authorized Email and Password, Teacher record, staff record



This is the class diagram of the system and here it clearly shows the relationship and multiplicities of admin class. It is connected to teachers, accounts and course. It depends on account whereas courses are controlled by admin, also teachers are hired by admin.



The above diagram shows interaction of admin with other minor components. It shows the sequence of the steps involved while performing a task. Admin performs the tasks in the above shown sequence.

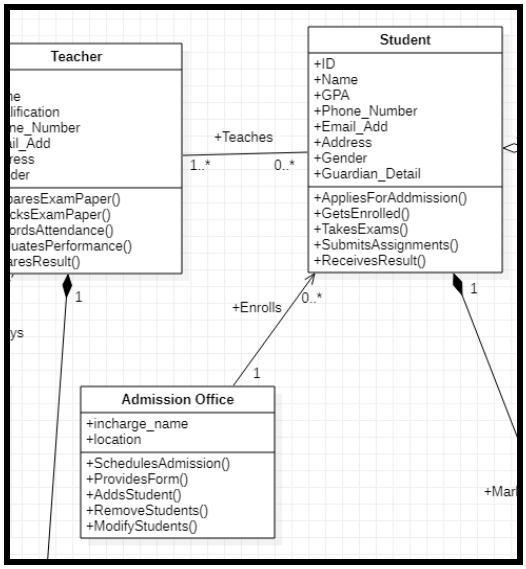


The flow of the activity performed by admin is shown in the above diagram.

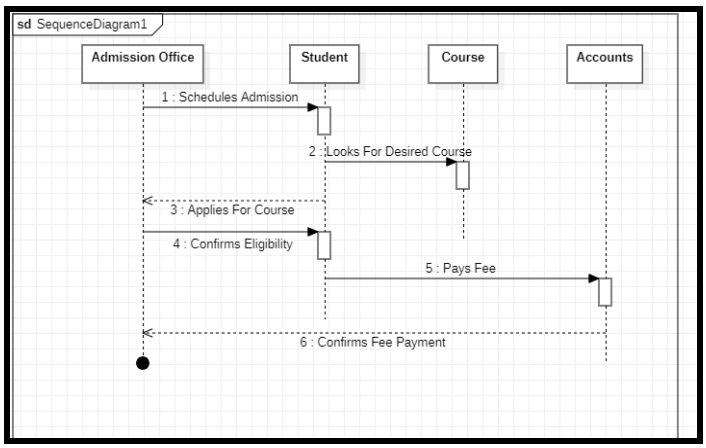
3.3 Admission Office Component (or Class or Function ...)

Identification	Admission office is the secondary actor in the system.
Type	A module and a class
Purpose	It controls the admission of the students.
Function	Schedules admission Enrolls Students Modify Students Provides form
Subordinates	Not Applicable

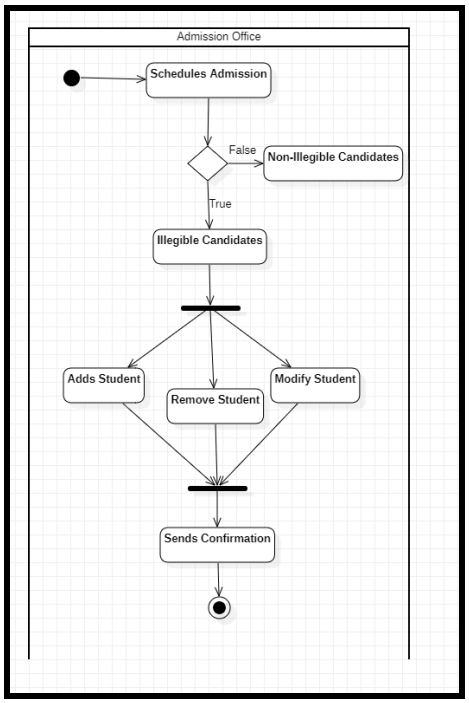
Dependencies	Not Applicable
Interfaces	The interface used is user-friendly and user-focused. While entertaining application form if information is not in required pattern it will prompt an error message
Resources	Resources used are internet, computer, human resource, admission form
Processing	Uploads the admission details Receives applications Selects eligible statements Sends them confirmation mails Receives confirmation mails from accounts
Data	Admission form, Student Application record



Admission office provides all the necessary details relevant to admission to the students. Admission office enrolls students, it can enroll many or no students at all.



Sequence of tasks performed by admission office is shown in the above diagram

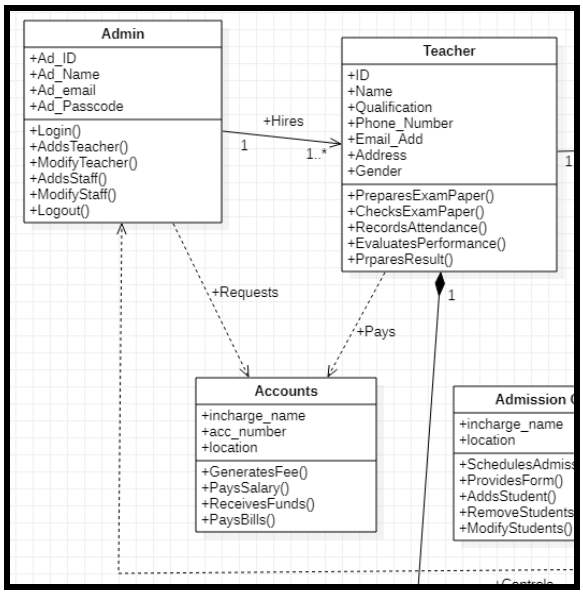


Flow of the activity performed by admission office is shown in the activity diagram. Here eligible students are allowed to follow the procedure for admission.

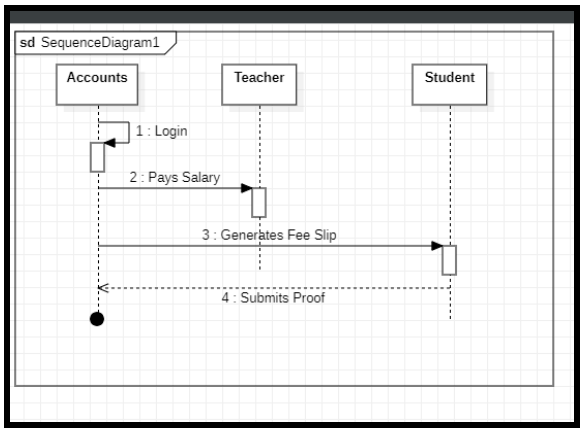
3.4 Accounts Component (or Class or Function ...)

Identification	Account is the secondary actor in the system.
Type	A module and a class
Purpose	It controls financial records.
Function	Generate fee slips Pays salary Pays bills Receive funds
Subordinates	Not Applicable
Dependencies	Not applicable
Interfaces	The interface used is user-friendly and user-focused. While logging in if any error takes place the authentication error is prompted. Moreover, if any database error occurs database error message is prompted.
Resources	Resources used are internet, computer, human resource, authorized email and password, financial record
Processing	Initiates the login process

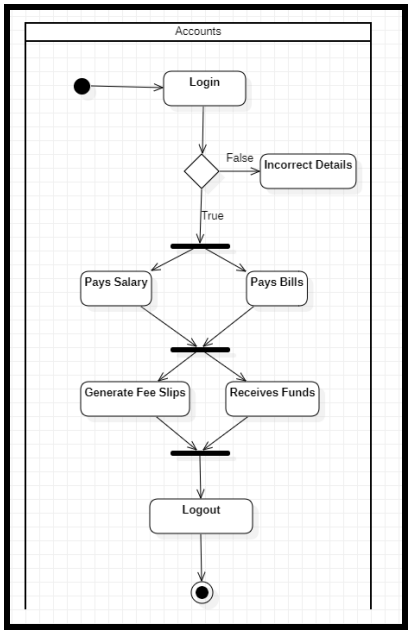
	Receives admission fee of student Sends confirmation mail to student and admin Confirms admission to admission office Receives request for sending salary to faculty Prepares checks Sends them to the recipients Sends confirmation to admin Ends the login session
Data	Authorized Email and Password, Financial records including teacher salaries paid, fee submitted by students, bills paid etc.



Accounts maintain all the monetary information, for example paying salary, confirming fee submission, paying bills etc.



Accounts interaction with teacher and students is shown in the above diagram. The sequence of the steps followed is picturized



Activity diagram of accounts shows the flow of the task.

4.0 Reuse and relationships to other products

The components that can support reusability are accounts and attendance.

In a university management system many components are involved in order to ensure its smooth working. Some components can be reused if used in some other management systems. To explain this further following subsections are formed:

4.1 Reusability

If after making a university management system we are asked to design a bank management system or employee management system. Attendance and Accounts are two such components that can be reused as-it-is. Reason for this reusability is their generic nature.

4.2 Implementation

The operation being performed by these two components are such that can be used in any other system which requires maintaining attendance and financial record. Paying bills, salary is something that is required in any sector. Moreover, maintaining and organizing attendance record is another generic factor. These two components can be reused in any such systems with minimal or no changes.

4.3 Non-reusable Components

The non-reusable components are such which cannot operate in any other environment than the ones they are designed for. For example Admin, it

is a component with operations like hiring teachers, modifying teachers etc. Now these operation are specific to an educational institutions. Moreover Teachers, course and department is also something specific to an educational institute and hence it cannot withstand the reusability factor.

5.0 Design decisions and tradeoffs

The design decision that best suits this system is **User-Focused Design.** The main focus of this design is to look into the goals and needs of the user and how quickly and efficiently they are being fulfilled.

In the university management system, the goal of the software is to be available to everyone even having a least knowledge about system handling. This type of design helps in creating an outstanding experience for the user. Hence, user focused techniques are used to enhance the overall experience and achieve the goal. The students will be able to get information regarding admissions will be able to apply for admission. Admin and Accounts will be able to carry out their assigned tasks by logging in onto the portal. Similarly teachers will be able to submit papers and results, and record student attendance.

As the efficiency of the system is increased its portability and understandability is decreased. Similarly reusability is increased, cost is increased likewise.

6.0 Pseudocode for components

The user will follow following steps, while using the UMS:

Student:

- Student will login on the portal
- Downloads the university application form
- Selects the course
- Fills the form
- Submits the form
- Receives the confirmation mail
- Submits the admission fee
- Receives confirmation mail
- Gets the admit card

Account:

- Initiates the login process
- Receives admission fee of student
- Sends confirmation mail to student and admin
- Confirms admission to admission office
- Receives request for sending salary to faculty
- Prepares checks
- Sends them to the recipients
- Sends confirmation to admin

- Ends the login session

Admission office:

- Uploads the admission details
- Receives applications
- Selects eligible statements
- Sends them confirmation mails
- Receives confirmation mails from accounts

7.0 Appendices (if any)

Not applicable