***GRAPHIC ERA DEEMED TO BE UNIVERSITY***



***SESSION 2022-23***

Software Engineering Lab

**(PCS 602)**

Student Management System

**Software Requirement Specification Document**

  Submitted To:             Submitted To: Submitted By:

  Mr.  Ankit Tomar Sir Mrs. Neha Tripathi Priyank Gupta

Assistant Professor B. Tech CSE

Department of CSE Section SE (49)

2015092

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| 1. | [Introduction](#_TOC_250036) |  |
| 1.1 | [Purpose](#_TOC_250035) |  |
| 1.2 | [Scope](#_TOC_250034) |  |
| 1.3 | [Definitions](#_TOC_250033) And Abbreviations |  |
| 2 | Overview |  |
| 2.1 | Project in Detail |  |
| 2.2 | [Product Perspective](#_TOC_250029) |  |
| 2.3 | [Product Functions](#_TOC_250028) |  |
| 2.4 | [User Characteristics](#_TOC_250027) |  |
| 2.5 | [General Constraints](#_TOC_250026) |  |
| 3 | [Specific Requirements1](#_TOC_250025) |  |
| 3.1 | [External Interface Requirements](#_TOC_250024) |  |
| 3.2 | [User Interfaces](#_TOC_250023) |  |
| 3.3 | [Hardware Interfaces](#_TOC_250022) |  |
| 3.4 | [Software Interfaces](#_TOC_250021) |  |
| 3.5 | [Communications Interfaces](#_TOC_250020) |  |
| 4 | [Functional Requirements](#_TOC_250019) |  |
| 4.1 | [Log In Module (LM)](#_TOC_250018) |  |
| 4.2 | [Registered Users Module (RUM)](#_TOC_250016) |  |
| 4.3 | [Normal Users Module (NUM)](#_TOC_250015) |  |
| 4.4 | Administrator Module (VBM) |  |
| 4.5 | Server Module (SM) |  |
| 5 | [Use Cases](#_TOC_250013) and Classes/Objects |  |
| 6 | [Non-Functional Requirements](#_TOC_250011) |  |
| 6.1 | [Performance](#_TOC_250010) |  |
| 6.2 | [Reliability](#_TOC_250009) |  |
| 6.3 | [Availability](#_TOC_250008) |  |
| 6.4 | [Security](#_TOC_250007) |  |
| 6.5 | [Maintainability](#_TOC_250006) |  |
| 6.6 | [Portability](#_TOC_250005) |  |
| 7 | [Inverse Requirements](#_TOC_250004) |  |
| 8 | [Logical Database Requirements](#_TOC_250003) |  |
| 9 | [Other Requirements](#_TOC_250002) |  |
| 10 | [Analysis Models](#_TOC_250001) |  |
| 10.1 | Sequence Diagrams |  |
| 10.1.1 | Student Registration |  |
| 10.1.2 | Examination Procedure |  |
| 10.1.3 | Data Flow Diagrams (DFD) |  |
| 10.1.4 | State-Transition Diagrams (STD) |  |
| 11 | Appendices |  |
| 11.1 | Activity Diagram Manual |  |
| 11.3 | State Chart Diagram Manual |  |

**Introduction**

A Student Management System (SMS) is a system that manages the records of students regarding its personal and academic details. A student management system is designed to help college authorities to manage student records in the most efficient way possible.

The day-to-day tasks of a college management team include viewing student data, managing admissions, reshuffling of sections, seats, quota, semester, sections, faculty, student category, examination type, exam hall management, etc. The SMS has custom search capabilities to aid in access student information with the help of known information. This makes the system easier to navigate and to use maximizing the performance of the software in terms of time and effectiveness.

**Purpose**

The purpose of a student management system is to enable management authorities in different colleges to handle student data in an efficient way. Further, the software is designed to manage the resources which were earlier handled by human efforts. The main purpose of this project is to integrate distinct sections of the organizations into consistent manner so that complex functions, problems and varying situations can be handled smoothly. An important aspect of this system is that it is user-friendly, i.e., it does not need expertise in any particular field to operate the system.

The student management system is designed with the aim of automating the admission and enrolment as per the board, quota, category and available seats. Additionally, the system aims to handle all the required information of students, faculty and courses, along with their regular modifications.

**Scope**

The scope of student management system software includes the following:

* Any college can use this system, i.e., it is not client-centric.
* Applicable for storing, retrieving and modifying details of students taking admission, passing out and ongoing batch, along with the faculty information.
* Delivering electronic workplace in a multi-lingual sphere.
* Providing admin and client privileges.

**Definitions and Acronyms**

**Definitions**

The student management system software is an automated version of manual student management system. It is capable of handling all types of student and faculty data, including college details, subject details, personal information of students and faculties, academic details, etc.

The software has two modes of accessing the data stored, namely, administrator mode and user mode. For the administrator, it allows to access and modify some important features information, while this functionality may be disabled for the user mode.

**Acronyms**

SMS: Student Management System UM: User Module

RUM: Registered User Module AM: Administrator Module

SM: Server Module DB: Database

**Overview**

Student Management System (SMS) is a web-based application that tracks current student’s academic information. It maintains academic information for office staff, students, faculty and other authorities. Instead of doing tedious paper work, students will be able to submit their required information to the management digitally, which can be accessed by all the departments.

**Project in Detail (gen description)**

There are many departments of administration in a college or university to have a better management. All the departments may have common and uncommon attributes of student information which may be entirely interdependent or may not be. Hence, it is required to have a software that is capable of centralizing all the domains and gather all the information at one place and can enable accessing only the required information type of a student.

To maximize the efficiency of the software, it is modularized as into different modules.

**Login Module:** It aims to provide entry to the system or website. Based on the type of login, the user is provided with privileges and functionalities. It serves the function of providing user to use the SMS. It enables two types of logins to the audience: administrator login and user login.

**Administrator Module:** In this module, when administrator logs in using his/her credentials, the SMS redirects to administrator page, that has three subsections. These subsections include student addition/ modification/ deletion, notice/ attendance/ result generation and fee details and schedules.

**Student Module:** When a student logs in, he/she is redirected to home page that has following pages to access profile, notices, attendance, results, fee, helpdesk. *Profile* has the student information page, *Notices* has all time-to-time updated notices that are uploaded by the management, *Attendance* section has day-wise attendance update of the student. The *Fee* section has the status of fees of the student (academic or hostel) and the *Helpdesk* has a query handling mechanism for students.

**Product Perspective**

The various system tools that have been used in developing both the front end, back end and other tools of the project are being discussed in this section.

**FRONT END:**

JSP, HTML, CSS, JAVA SCRIPTS are utilized to implement the frontend.

**JSP (Java Server Page)** component is a type of java server that is designed to fulfil the role of a user interface for a java web application. Web development write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands. Using JSP, one can collect input from users through web page.

**HTML (Hyper Text Mark-up Language)** is a syntax used to format a text document on the web.

**CSS (Cascading Style Sheets)** is a style sheet language used for describing the look and formatting of a document written in a mark-up language.

**JS (Java Script)** is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

**BACK END:**

The back end is implemented using MYSQL which is used to design the databases.

**MYSQL:** MySQL is the world’s second most widely used opensource relational database management system (RDMS). The SQL phrase stands for structured query.

**PHP:** PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data.

**SMS GATEWAY:**

An SMS gateway allows a computer to send or receive short message services (SMS) transmissions to or from a telecommunications network. Most messages are eventually routed into the mobile phone networks. Many SMS gateways support media conversion from email and other formats. A direct-to-mobile gateway is a device which has built-in wireless.

*GSM connectivity* allows SMS text messages to be sent or received by email, from web pages or from other software applications by acquiring a unique identifier from the mobile phone's subscriber identity module, or "SIM card". Direct-to-mobile gateways are different from SMS aggregators, because they are installed on an organization's own network and connect to a local mobile network. The connection to the mobile network is made by acquiring a SIM card number from the mobile operator and installing it in the gateway.

### Product Functions

The primary function of the SMS web server is essentially to save the whole system information in sequentially into database server. The administrator will have access to whole system environment and that can be modified as per their needs. The architecture of whole system is made easy that any person can login to system and use the functions. The system database is only accessible to admin and admin can only modify them.

**User Characteristics**

The user profiles identified to have interaction with Student Management System that anyone can register and login into system and us the required resources. The students can easily fill up the registration form online and submit it. And the admin will check the details that is the student is eligible as per the admission criteria. After the student will successfully registered, he can use college/school system environments as per their limits decided by admin.

**General Constraints**

Student Management System can be accessed successfully by any client location and it’s not necessary that every registration is genuine, so there is chance of fake registration that may reflect some errors. So, the system is designed such a way that the database will keep updated by administrator and there are better security options available on the server that can prevent fake IP addresses to access system.

# Specific Requirements

### External Interface Requirements

### User Interfaces

#### The Student Management System web server must provide a user interface that will be accessible through any internet browser, the major ones being Google Chrome and Internet Explorer 12.

#### Hardware Interfaces

#### All components able to be executed on personal computers with Windows OS platforms and other platforms like Linux, Unix.

#### Software Interfaces

#### All the interfaces will be ASPX pages running within the internet browser. The SMS must integrate with the DB though SQL Interface. The system will be hosted in a web server running on Windows Server 2005.

#### Communications Interfaces

#### Connections to the system will be over TCP/IP connection.

#### Functional Requirements

#### Log in Module (LM)

#### User (admin, student and teachers) shall be able to load the Login Module in the internet browser. The LM shall support the user to log into the system. The login panel shall contain fields to contain a user name and a field for password. The password field shall be masked with symbols when the user types. It shall also contain a button labelled as Login. When the user clicks on Login button the username and password will be verified by database administrator and then only the user will able to use the system functions.

#### Registered Users Module (RUM)

#### After successful login, user shall be able to continue navigating through the website and view school/college detailed information. After successful login, user (admin, student and teachers) shall be able to update and maintain their profile, such as changing password and personal details.

#### Normal Users Module (NUM)

#### Users who visit SMS but have not registered, are able to navigate through the website. Users shall be able to view currently held events & upcoming institute schedule. Users shall be able to view school/college timings and their faculties information. Users are able to register themselves as registered users, by clicking on the register now button.

#### Administrator Module (AM)

#### After successful login, system shall display administrative functions. Administrative functions shown shall be add and update. When administrator clicks on the add button, system shall display a section where administrator can add new student details, remove unused student details and many more. When administrator clicks on update button, system shall display a section where administrator can update student details and schedule of lecture which are currently stored in the database. When administrator adds, updates or delete and entry, the AM module will send the request to the Server Module which will do the necessary changes to the DB.

#### Server Module (SM)

#### SM shall be between the various modules and the DB. SM shall receive all requests and format the pages accordingly to be displayed. SM shall validate and execute all requests from the other modules.

#### Diagram Description automatically generatedUse Cases and Classes/Objects

**Non-functional Requirements**

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc.)

**Performance**

The Student Management System shall be built upon the web development technique and put on the web server online. The system and the server must be capable of handling the real- time error functionality occurs by the defined users. In addition, the system must be safety critical. All failures reported by the server side must be handled instantaneously to allow for user and system safety.

**Reliability**

The system is safety critical. If it moves out of normal operation mode, the requirement to drop or down the server and fix it as soon as possible and open it again. This emergency behavior shall not occur without reason.

**Availability**

When in normal operating conditions, request by a user for an online system shall be handled within 1 second. Immediate feedback of the systems activities shall be communicated to the user by clearing the system and giving space n speed to their hospitality.

**Security**

There shall be a strong security mechanism should be place in the server side of the system to keep unwanted users to hack or damage the system. However, all users of the system give and store the details of privacy related to personal information and many other. However, our system can be accessed online so we need very secured system as far as security is concerned.

**Maintainability**

There shall be design documents describing maintenance of the software and database used to save the user details as well as the daily updated and modification done in system. There shall be an access on the control system by the admin to maintained it properly at the front end as well as at back end.

**Portability**

There is portability requirement as far as our system is concern because it is an online as well as offline (local server based) system so we can access it from anywhere through the internet connection. And we have to maintain the copy of stored data into our database.

**Inverse Requirements**

As far as Inverse Requirement is concerned, our system has best inverse requirement which is most important according to our system view. That is if student want to cancel the registration so admin can access the student detail for cancellation and will refund the fees that was fetch from student and not complete payment will be back which is the trend and rule by every school/college management.

**Logical Database Requirements**

A one-to-many relational database shall be used in order to validate various student requests and details can be mismatched. Moreover, mismatches are to be logged for reference. The database shall be concurrent with the performance requirements of the Student Management System.

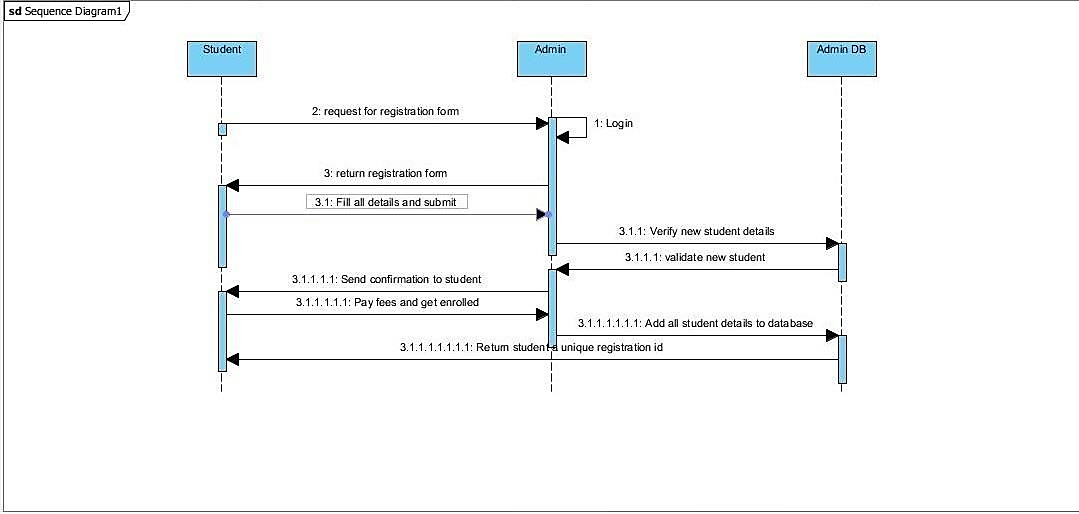
**Other Requirements**

A degraded mode of operation should be possible in which each system can operate independently of central scheduling. The software shall have failure and error recognition codes acting as a safety net, thus keeping the software from performing any major catastrophic functions.

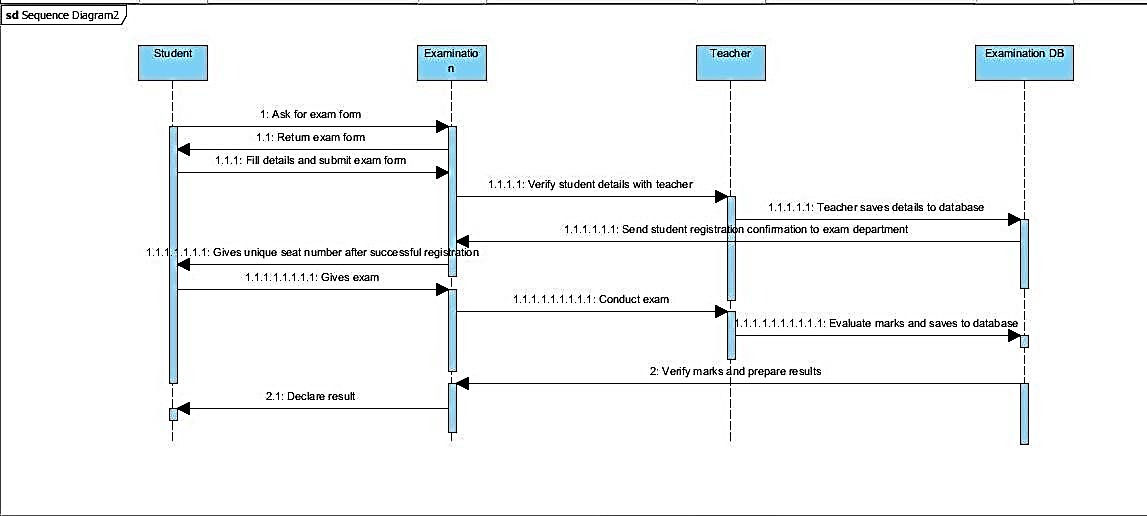
**Analysis Models**

**Sequence Diagrams:**

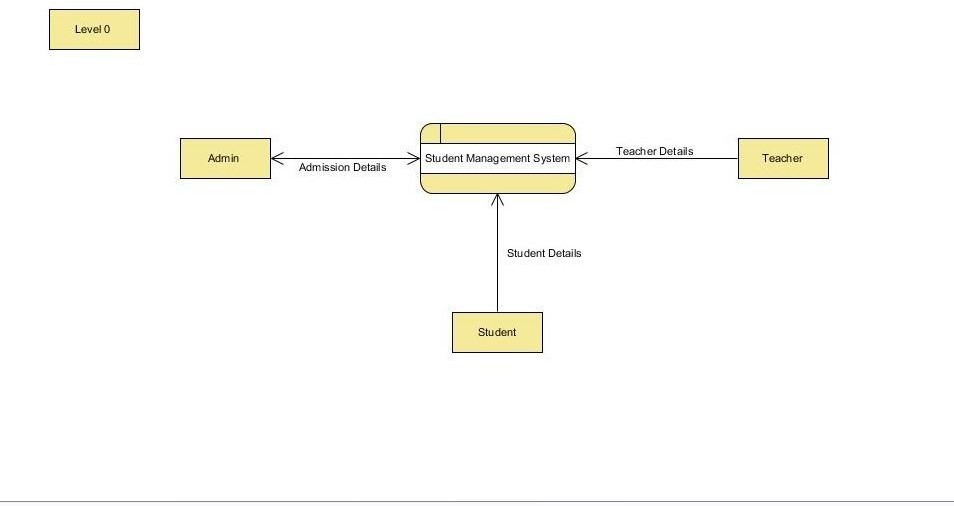
**A] Student Registration:**

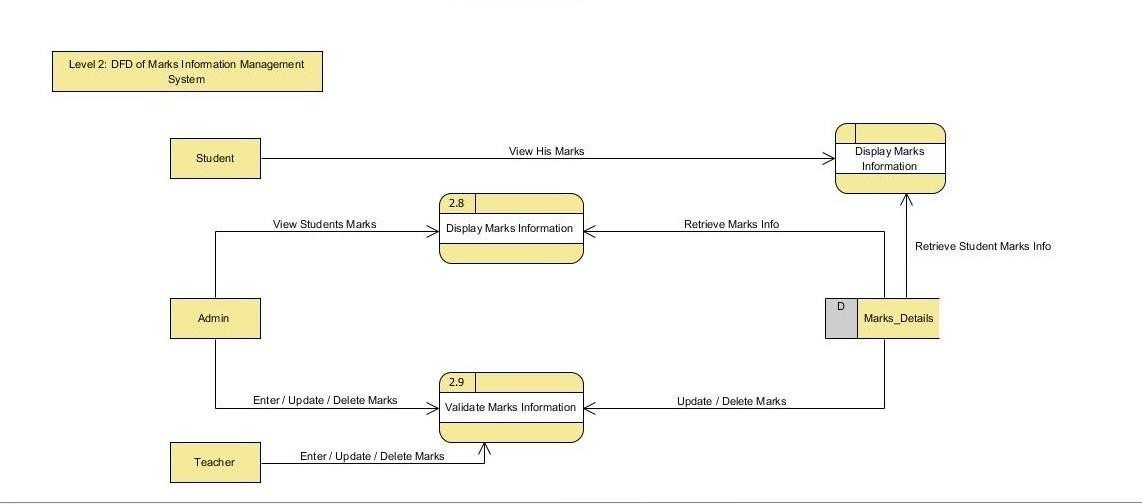
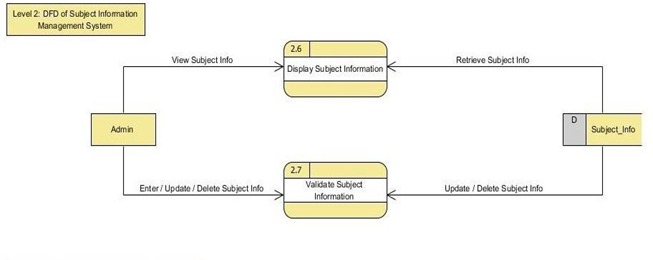
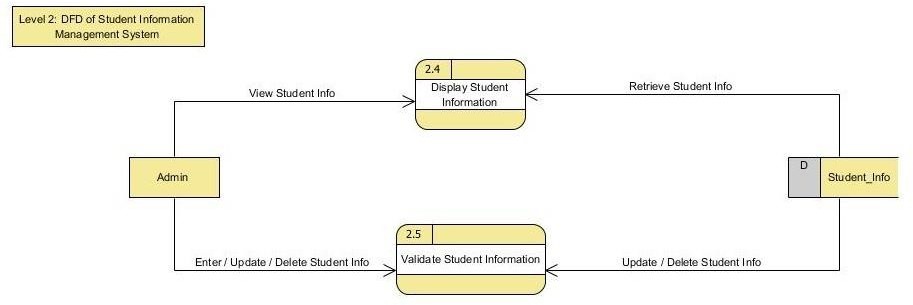
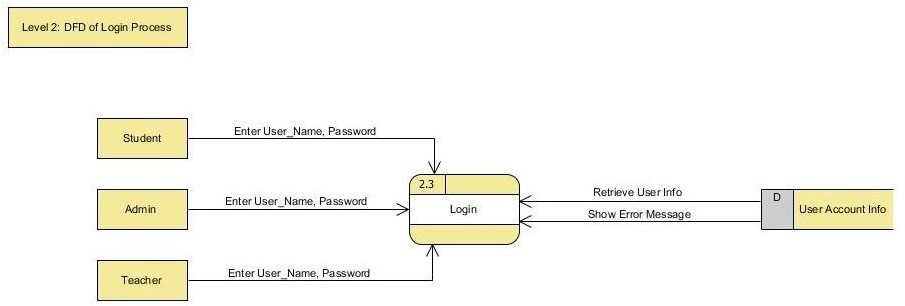
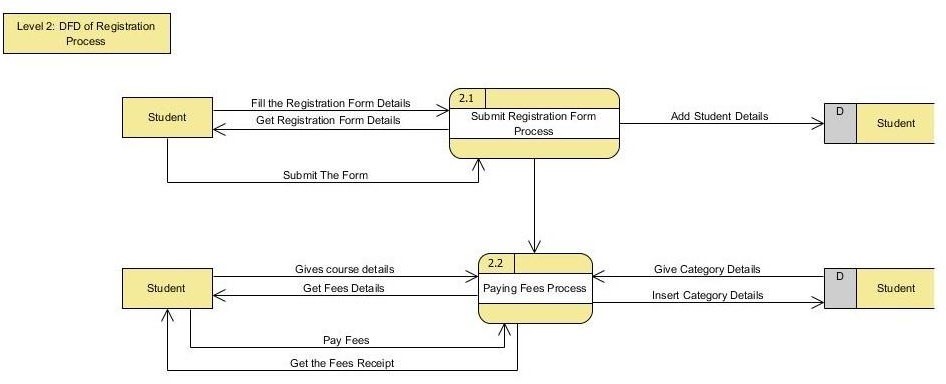
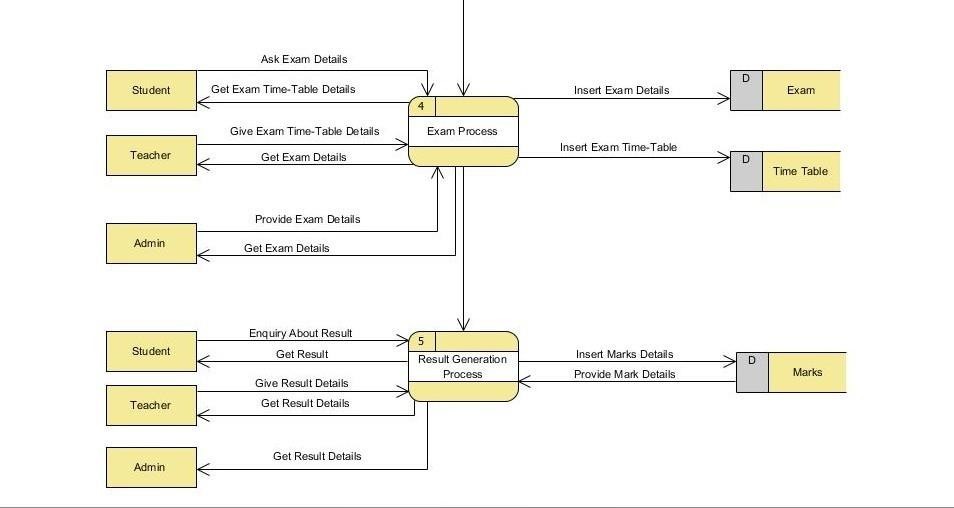
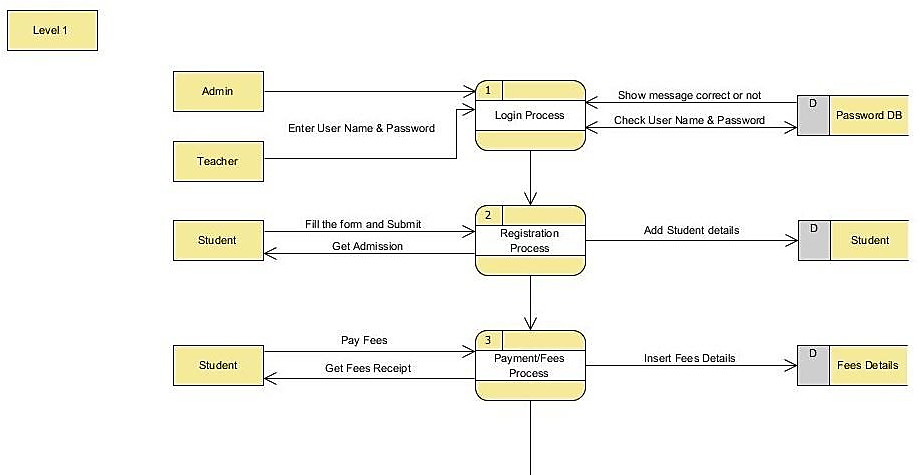


**B] Examination Procedure:**

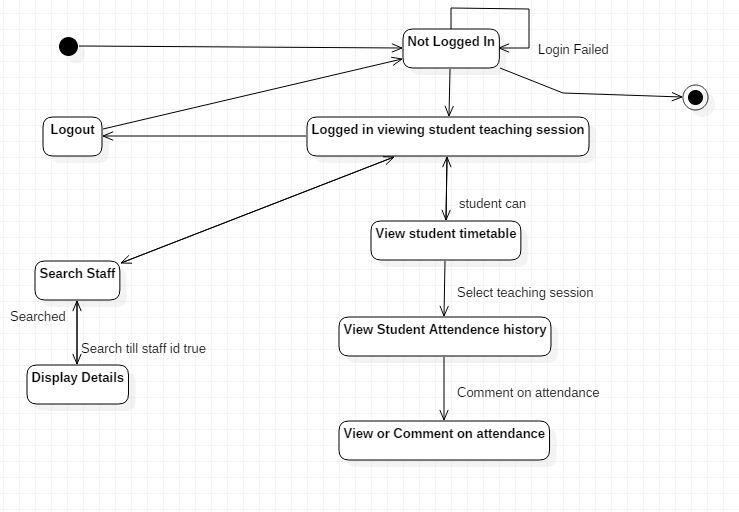
****

**C] Data Flow Diagrams (DFD)**





**State-Transition Diagrams (STD)**

****

**Appendices**

**Appendix 1- Activity diagram manual**

**UML Activity Diagram Overview:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system.

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system.

So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

**Purpose:**

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

So, the purposes can be described as:

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

**Appendix 2- State Chart Diagram manual**

UML State chart Diagram Overview:

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A State chart diagram describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. Activity diagram explained in next chapter, is a special kind of a State-chart diagram. As State chart diagram defines states it is used to model lifetime of an object.

**Purpose:**

State chart diagram is one of the five UML diagrams used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. So Statechart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Following are the main purposes of using State chart diagrams:

* To model dynamic aspect of a system. To model life time of a reactive system.
* To describe different states of an object during its life time. Define a state machine to model states of an object.

Where to use State chart Diagrams?

From the above discussion we can define the practical applications of a State-chart diagram. State chart diagrams are used to model dynamic aspect of a system like other four diagrams.

disused in this tutorial. But it has some distinguishing characteristics for modeling dynamic nature.

State chart diagram defines the states of a component and these state changes are dynamic in nature. So its specific purpose is to define state changes triggered by events. Events are internal or external factors influencing the system.

State chart diagrams are used to model states and also events operating on the system. When implementing a system it is very important to clarify different states of an object during its life time and statechart diagrams are used for this purpose. When these states and events are identified, they are used to model it and these models are used during implementation of the system.

If we look into the practical implementation of State chart diagram then it is mainly used to analyze the object states influenced by events. This analysis is helpful to understand the system behavior during its execution. So, the main usages can be described as:

* To model object states of a system.
* To model reactive system. Reactive system consists of reactive objects. To identify events responsible for state changes.
* Forward and reverse engineering.