

Student Management System

Software Requirements Specification

15 October 2015

Prasad Chandresh Kamleshwar
Lead Software Engineer

Prepared for
SOOAD
Instructor: Kalpana R. Bodke

Revision History

Date	Description	Author	Comments
15/10/2015		Chandresh K Prasad	First Revision

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Prasad Chandresh Kamleshwar	Lead Software Eng.	
	Kalpana R.Bodke	Instructor	

Table of Contents

REVISION HISTORY.....	II
DOCUMENT APPROVAL.....	II
1. INTRODUCTION.....	5
1.1 PURPOSE.....	5
1.2 SCOPE.....	6
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	6
1.4 REFERENCES.....	7
1.5 OVERVIEW.....	7
2. GENERAL DESCRIPTION.....	8
2.1 PRODUCT PERSPECTIVE.....	9
2.2 PRODUCT FUNCTIONS.....	10
2.3 USER CHARACTERISTICS.....	10
2.4 GENERAL CONSTRAINTS.....	11
3. SPECIFIC REQUIREMENTS.....	11
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	
3.1.1 User Interfaces.....	11
3.1.2 Hardware Interfaces.....	11
3.1.3 Software Interfaces.....	11
3.1.4 Communications Interfaces.....	11
3.2 FUNCTIONAL REQUIREMENTS.....	
3.2.1 Log in Module (LM)	11
3.2.2 Registered Users Module (RUM)	11
3.2.3 Normal Users Module (NUM)	12
3.2.4 Administrator Module (AM)	12
3.2.5 Virtual Bank Module (VBM)	12
3.2.6 Book Tickets Module (BTM)	12
3.2.7 Payment Module (PM)	12
3.2.8 Ticket Collection Module (TCM)	12
3.2.9 Server Module (SM)	12
3.3 USE CASES.....	
3.3.1 Use Case #1.....	13
3.4 CLASSES	
3.4.1 Class #1.....	14
3.5 NON-FUNCTIONAL REQUIREMENTS.....	
3.5.1 Performance.....	15
3.5.2 Reliability.....	15
3.5.3 Availability.....	15
3.5.4 Security.....	15
3.5.5 Maintainability.....	15
3.5.6 Portability.....	15
3.6 INVERSE REQUIREMENTS	16
3.7 LOGICAL DATABASE REQUIREMENTS.....	16

3.8 OTHER REQUIREMENTS.....	16
4. ANALYSIS MODELS.....	
4.1 SEQUENCE DIAGRAMS.....	17
4.2 DATA FLOW DIAGRAMS (DFD).....	19
4.3 STATE-TRANSITION DIAGRAMS (STD).....	23
A. APPENDICES.....	
A.1 APPENDIX-1.....	24
A.1 APPENDIX-2.....	27

1. Introduction

A Student Management System (SMS) is a System that manages the records of student regarding admission and examination part. A Student Management System (SMS) is designed to help collages for management of dental student. Extensive information is available at your fingertips through this System. Viewing student data, managing admission and reshuffling, managing seats, quota, board, semester, faculty, category and for examination, block allocation, subject management, scheduling exam, result and related issues are made simple and easy. There are custom search capabilities to aid in finding student information and working on student records. This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources. SMS allows the keeping of personnel data in a form that can be easily accessed and analysed in a consistent way.

1.1 Purpose

The project is about to handle all the information of the student regarding admission and examination. Also it manages resources which were managed and handled by manpower previously. The main purpose of the project is to integrate distinct sections of the organization into consistent manner so that complex functions can be handled smoothly by any technical or non-technical persons.

The project aims at the following matters:

- Automation of admission and enrolment as per board, quota, category and available seats.
- Assistance in decision-making.
- To manage information of student, faculty and courses.
- Consistently update information of all the students.

The main purpose of the Admin Module is to introduce new things and configure important aspects. For e.g. only admin is authorized to introduce quota, board, subject, category, etc. and only admin is allowed to configure exam and set fees structure. So the master screens for all these are visible to only admin role. This is done by the Admin Module. It also can create the users and Physical and Logical Locations. Thus the main purpose of the Admin Module is to managing the dynamic working of the system.

1.2 Scope

The scope of the project includes the following:

- Any college can use this system as it is not client centric.
- All admission and examination related work for the student can be done using this system.
- Deliver Electronic Workplace
- Provide Bi-lingual support
- Application Support & Maintenance after deployment to production
- The Admin Module can be reused for projects as well which have many users with different rights. Hence it is reusable.

1.3 Definitions, Acronyms, and Abbreviations

Definitions:

The student management system is an automated version of manual Student Management System. It can handle all details about a student. The details include college details, subject details, student personnel details, academic details, exam details etc.

Our system has two type of accessing modes, administrator and user. Student management system is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. When a user log in to the system. He would only view details of the student. He can't perform any changes.

Acronyms:

SMS: Student Management System

LM: Log in Module

RUM: Registered Users Module

NUM: Normal Users Module

AM: Administrator Module

SM: Server Module

DB: Database

1.4 References

[1] <http://www.slideshare.net/>

[2] <http://www.sourcecodesolutions.in/>

[3] <http://www.google.com/>

1.5 Overview

Student Management System (SMS) is a web-based application that tracks current student's academic information. It maintains academic information for ready access by office staff, students, their faculty advisors, and committee members. Instead of tedious paper work, students will be able to submit required information electronically, and the departments will be able to evaluate the submissions with a much quicker turnaround.

The Student Management System has been modularized into following modules.

LOGIN MODULE:

The purpose of this module is to provide entry to the system or website. Based on the type of login, the user is provided with various facilities and functionalities. The main function of this module is to allow the user to use SMS. This module provides two types of login —Admin login and Student login.

ADMINISTRATOR MODULE:

In this module when the administrator will enter his/her user name and password, then he/she will enter in to the administrator page and this page consists of two following sub modules.

Student Addition/ Updation / Deletion: In SMS each Student is added, updated or deleted according to its branch.

Notice/Attendance/Result Generation: In SMS information about notice, attendance and Internal result is generated.

Fee Detail and Schedules: Fee information detail and schedule detail are managed.

STUDENT MODULE:

In this module when a user enters his student id and password, then he can visit all the following pages.

Profile View: When the student clicks on this link he/she will get his/her information like student id, student name, password, father name, date of birth, nationality, city, address, country, phone number, mobile number, email. If he/she wants then he/she can change the profile.

Notice View: When the student clicks on this link, he can see latest notices released by the administrator.

Attendance View: When the student clicks on this one, the student can get his overall attendance percentage (present and absent).

Internal Results View: When the student clicks on this, he/she will get the internals result in all the subjects. How much grade point he/she secure out of 20 he/she can know.

Fee Detail View: When the student clicks this link he/she can get all the fees structure semester wise and annual fee.

The Student Helpdesk: This helpdesk is staffed by faculty who are there to help you. You may contact on (faculty phone no.).

2. General Description

There are many departments of administration for the maintenance of college information and student databases in any institution. All these departments provide various records regarding students. Most of these track records need to maintain information about the students. This information could be the general details like student name, address, performance, attendance etc. or specific information related to departments like collection of data. All the modules in college administration are interdependent. They are maintained manually. So they need to be automated and centralized as, Information from one module will be needed by other modules.

For example, when a student needs his course completion certificate it needs to check many details about the student like his name, reg. number, year of study, exams he attended and many other details. So it needs to contact all the modules that are once, department and examination and result of students. With that in mind, we overhauled the existing Student Database Management System and made necessary improvement to streamline the processes.

Administrators using the system will find that the process of recording and retrieving students information and managing their classes, including marking of attendance, is now a breeze. In general, this project aims to enhance efficiency and at the same time maintain information accurateness. Later in this report, features and improvement that allow achievement to this goal will be demonstrated and highlighted.

2.1 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.

Java Server Page (JSP)

Different pages in the applications are designed using JSP. A java sever page component is a type of java server that is designed to fulfill 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)

HTML is a syntax used to format a text document on the web.

CSS (Cascading Style Sheets)

CSS is a style sheet language used for describing the look and formatting of a document written in a mark-up language.

Java Script

JS 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 open source 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. It 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.

2.2 Product Functions

The primary function of the Student Management System web server is essentially to save the whole system information in sequentially into database server. The administration department 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 the changes.

2.3 User Characteristics

The user profiles identified to have interaction with Student Management System that anyone can register and login into system and use 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.

2.4 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 chances 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 is better security options available on the server that can prevent fake IP addresses to access system.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 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.

3.1.2 Hardware Interfaces

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

3.1.3 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.

3.1.4 Communications Interfaces

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

3.2 Functional Requirements

3.2.1 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.

3.2.2 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.

3.2.3 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.

3.2.4 Administrator Module (AM)

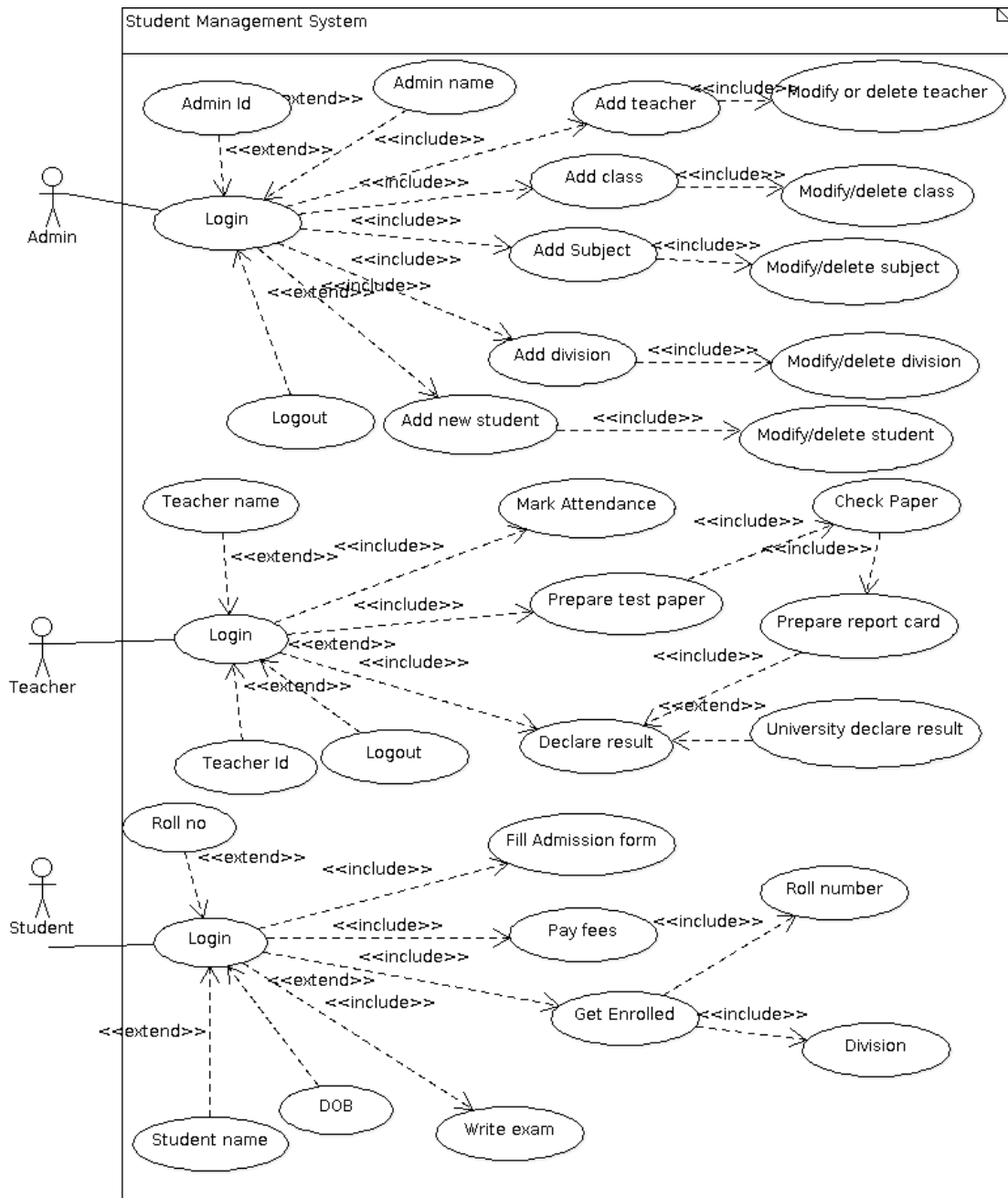
After successful login, system shall display administrative functions. Administrative functions shown shall be add and update. When administrator clicks on the add button, system shall display a section where administrator can add new student details, remove unused student details and many more. When 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.

3.2.9 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.

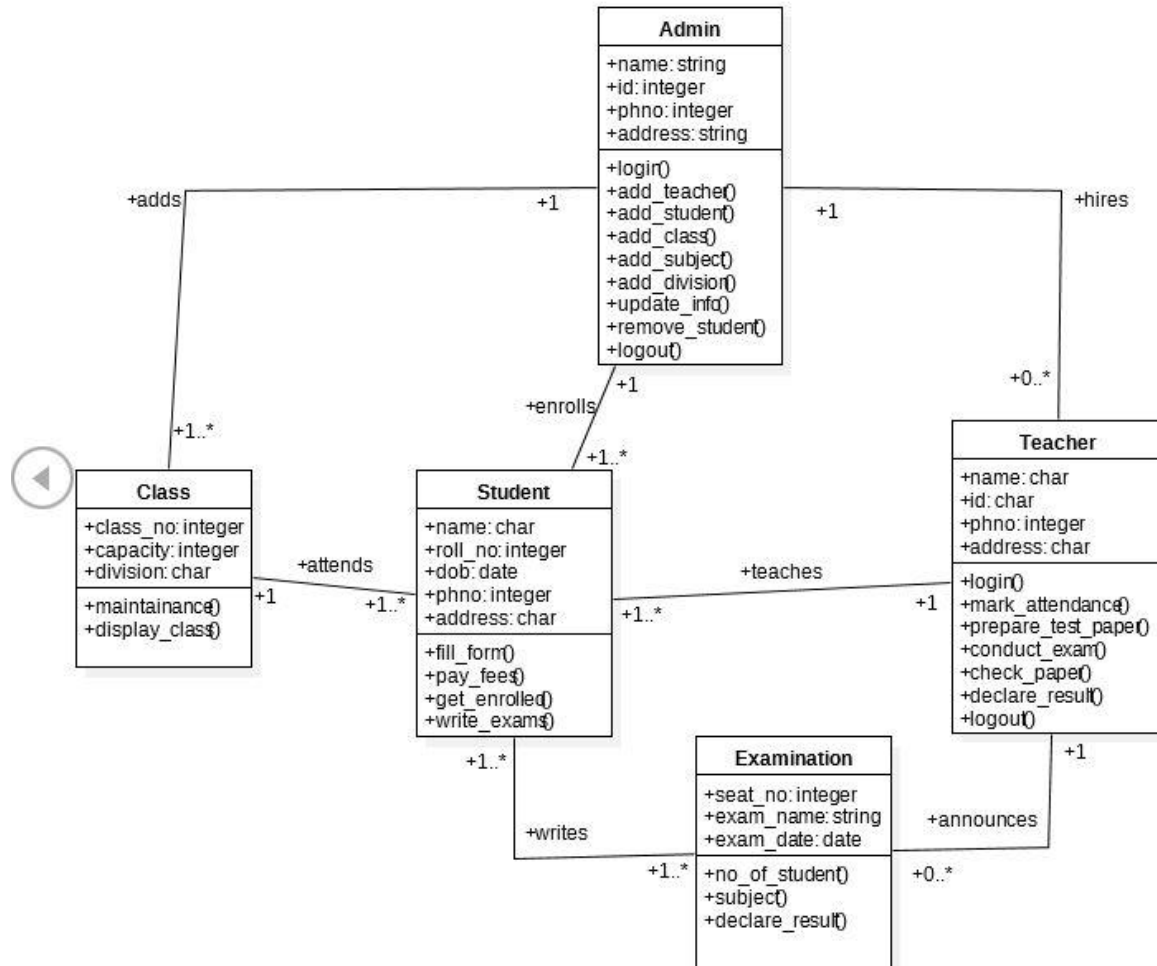
3.3 Use Cases

3.3.1 Use Case #1



3.4 Classes / Objects

3.4.1 Class / Object #1



3.5 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.)

3.5.1 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.

3.5.2 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 behaviour shall not occur without reason.

3.5.3 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.

3.5.4 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.

3.5.5 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.

3.5.6 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.

3.6 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.

3.7 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.

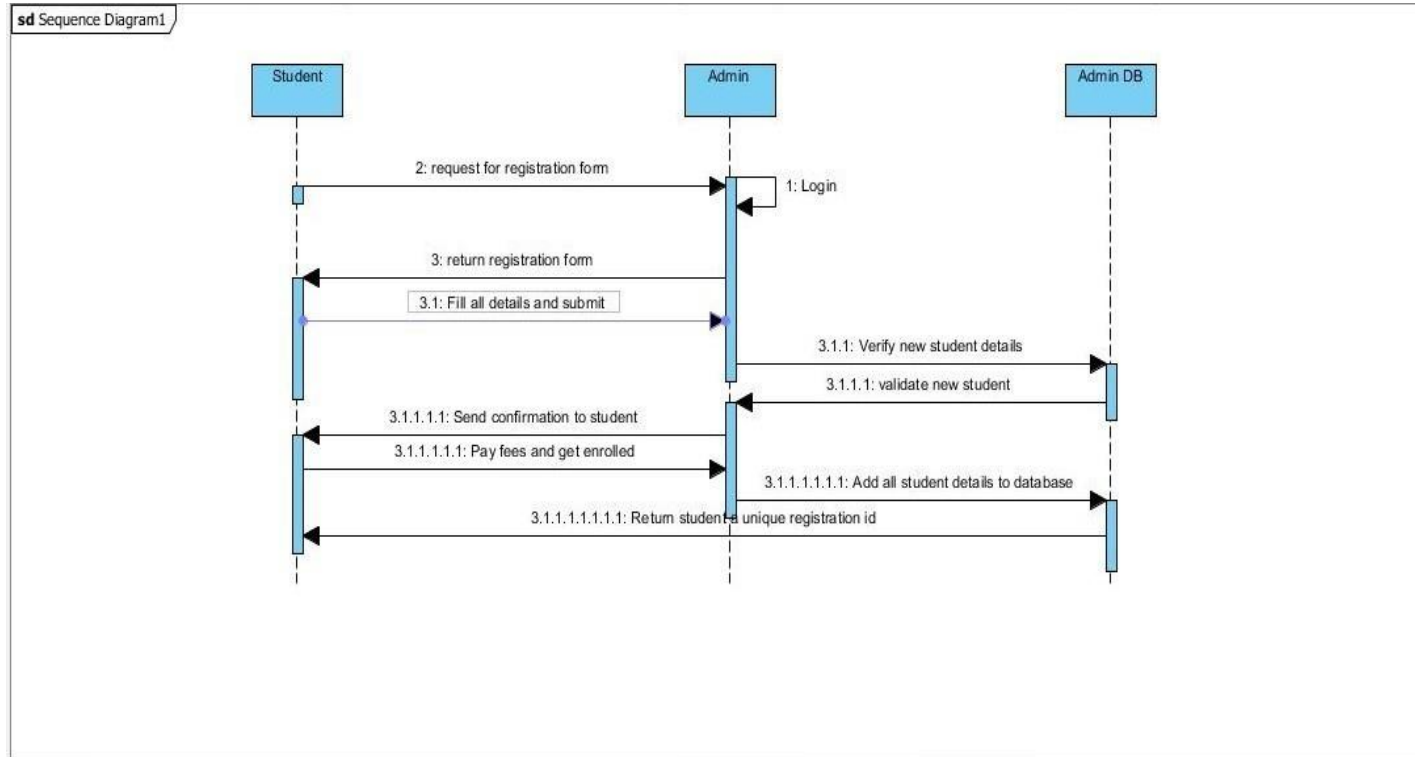
3.8 Other Requirements

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.

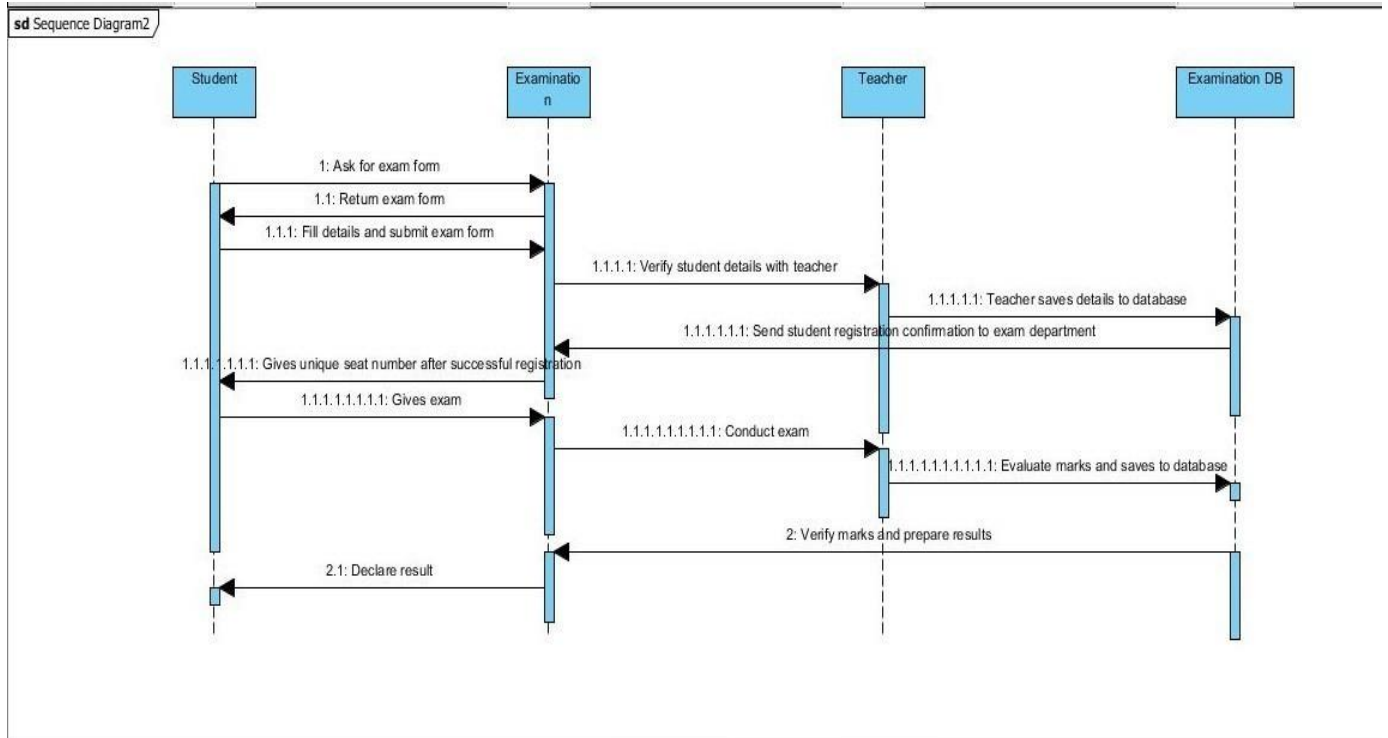
4. Analysis Models

4.1 Sequence Diagrams:

A] Student Registration:

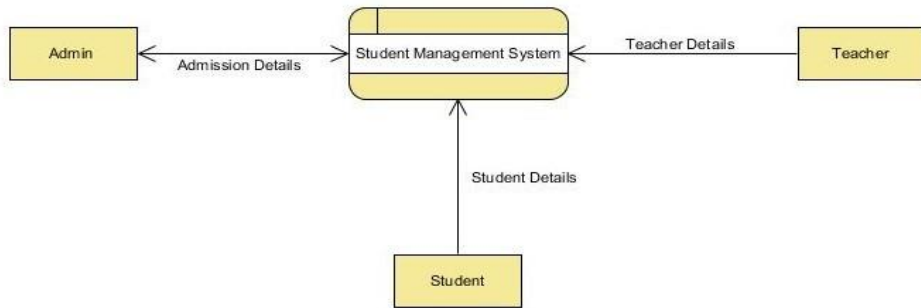


B] Examination Procedure:

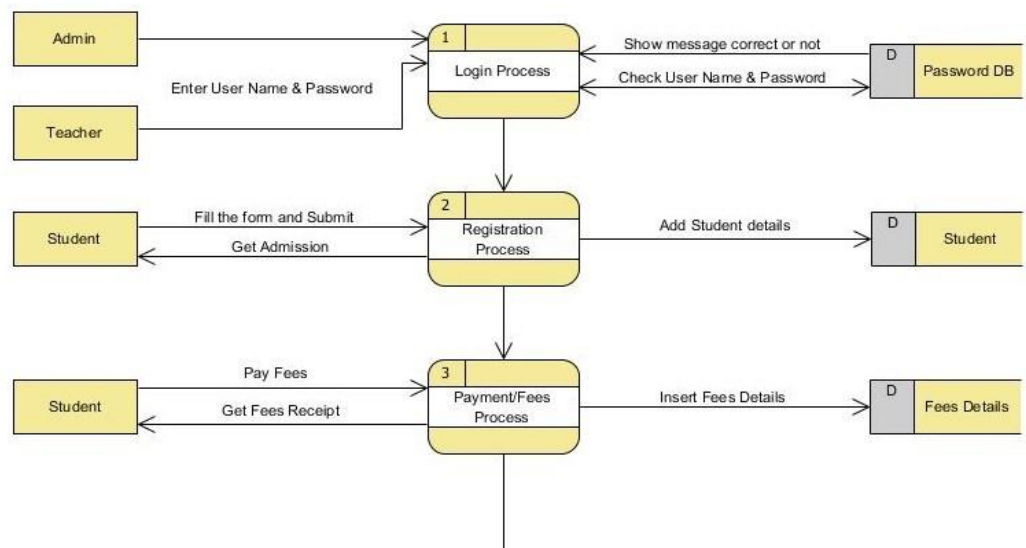


4.3 Data Flow Diagrams (DFD)

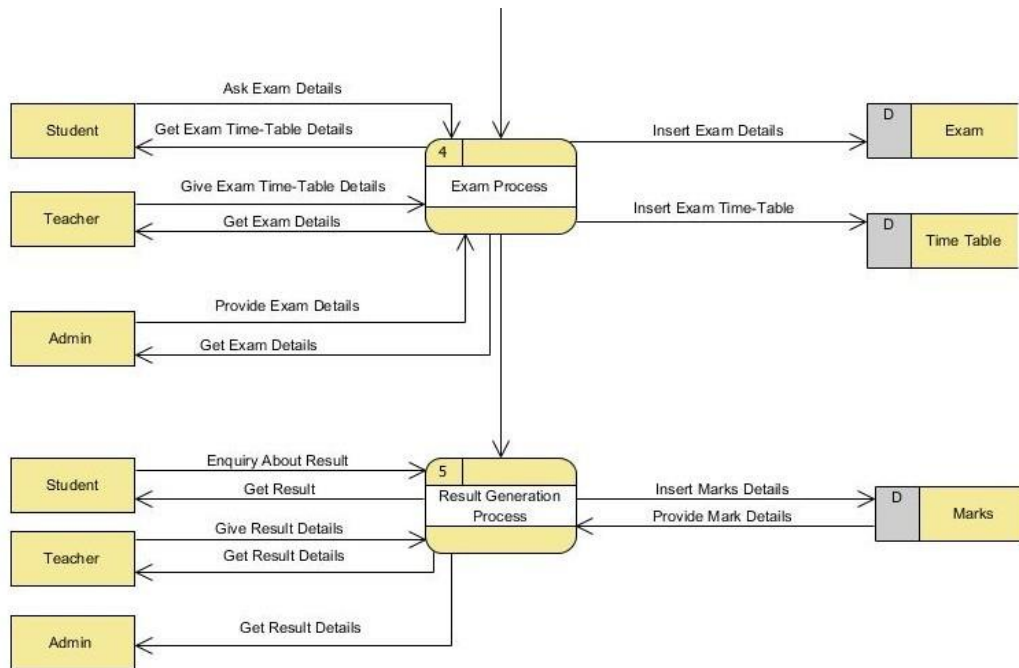
Level 0



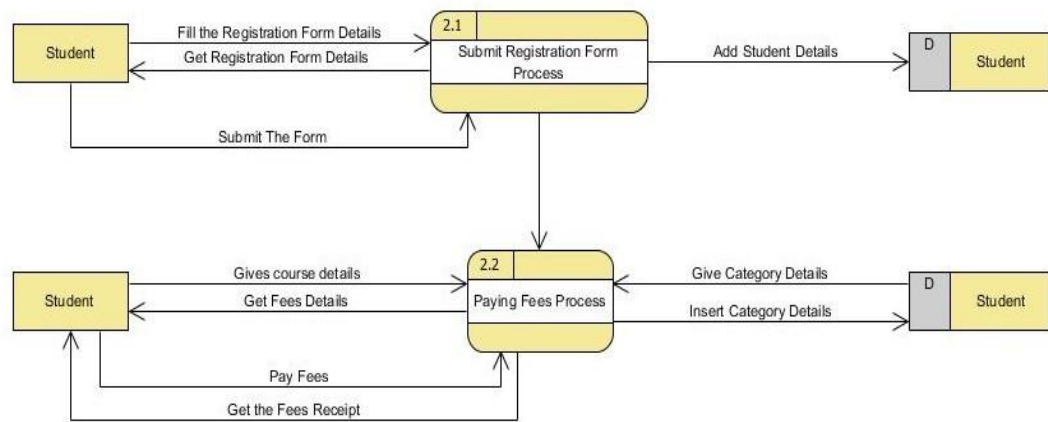
Level 1



Student Management System

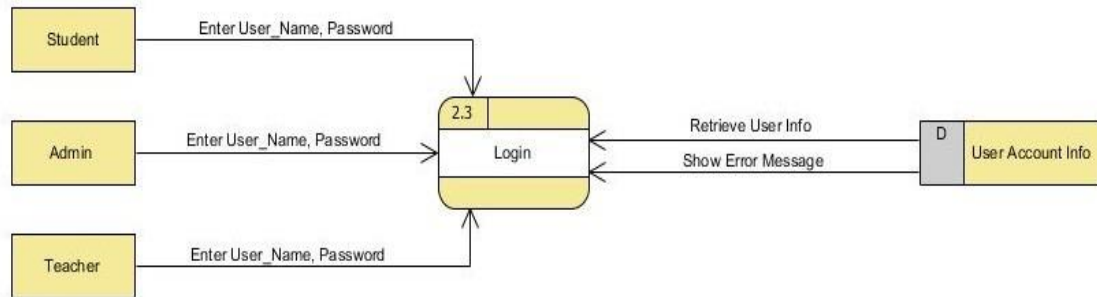


Level 2: DFD of Registration Process

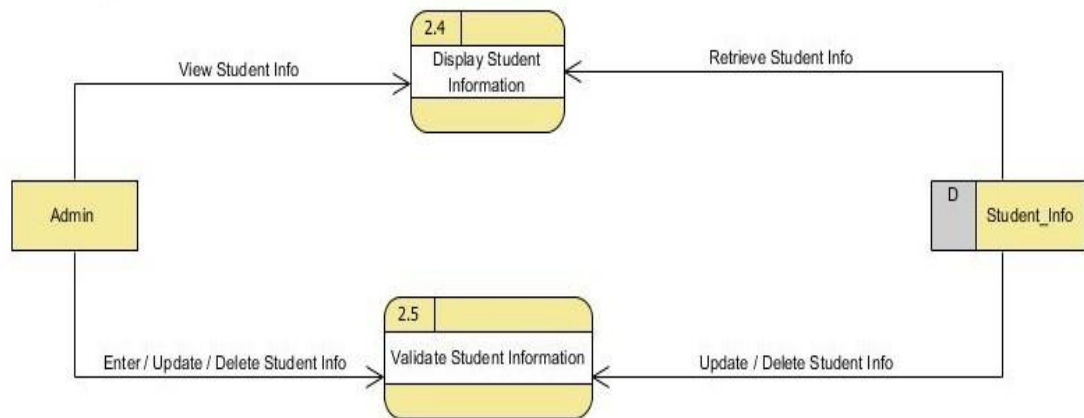


Student Management System

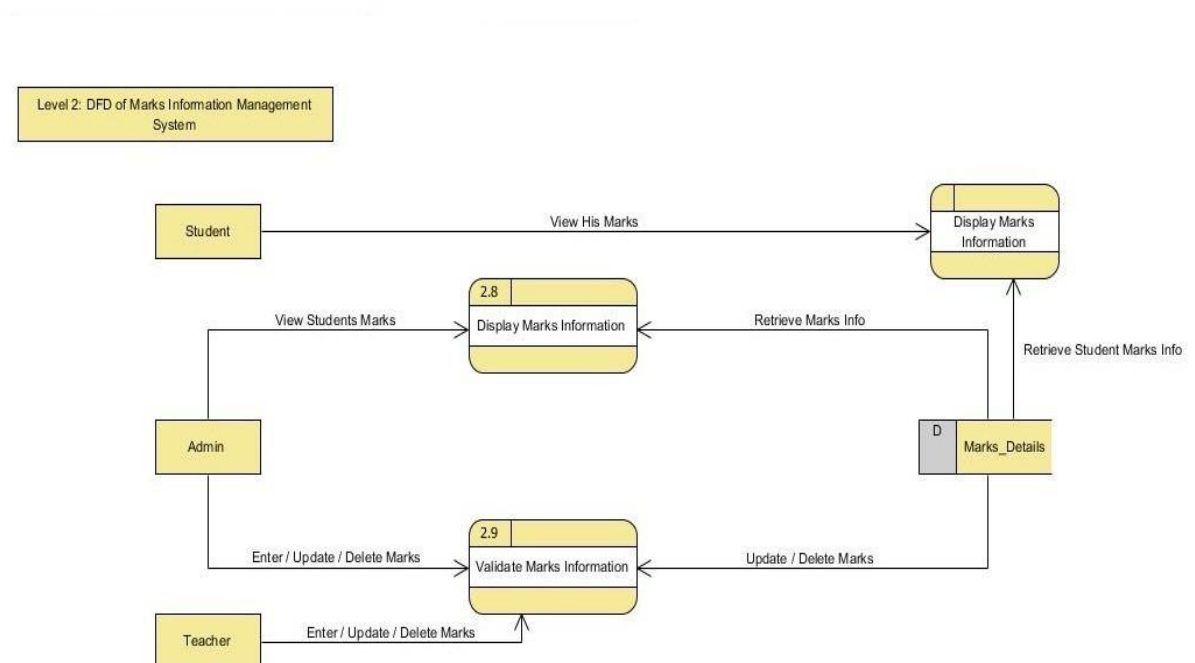
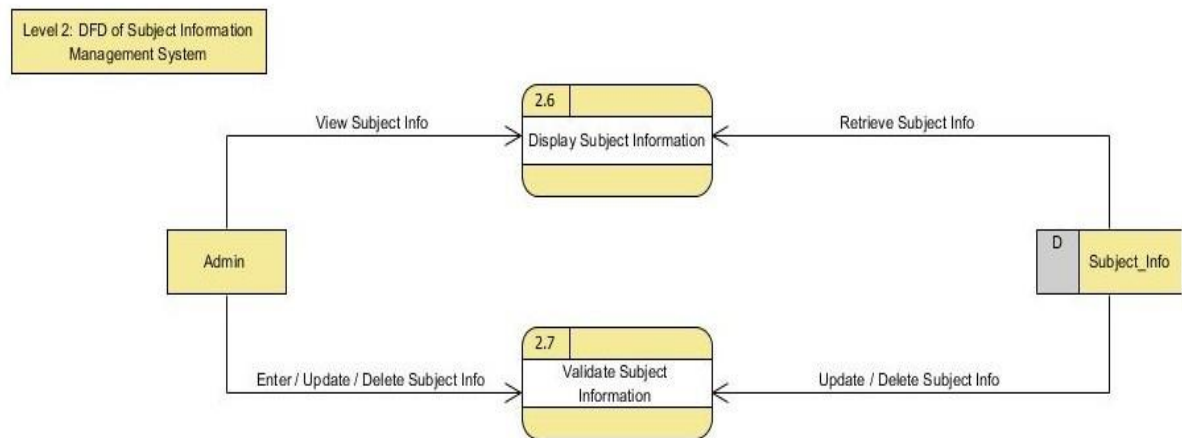
Level 2: DFD of Login Process



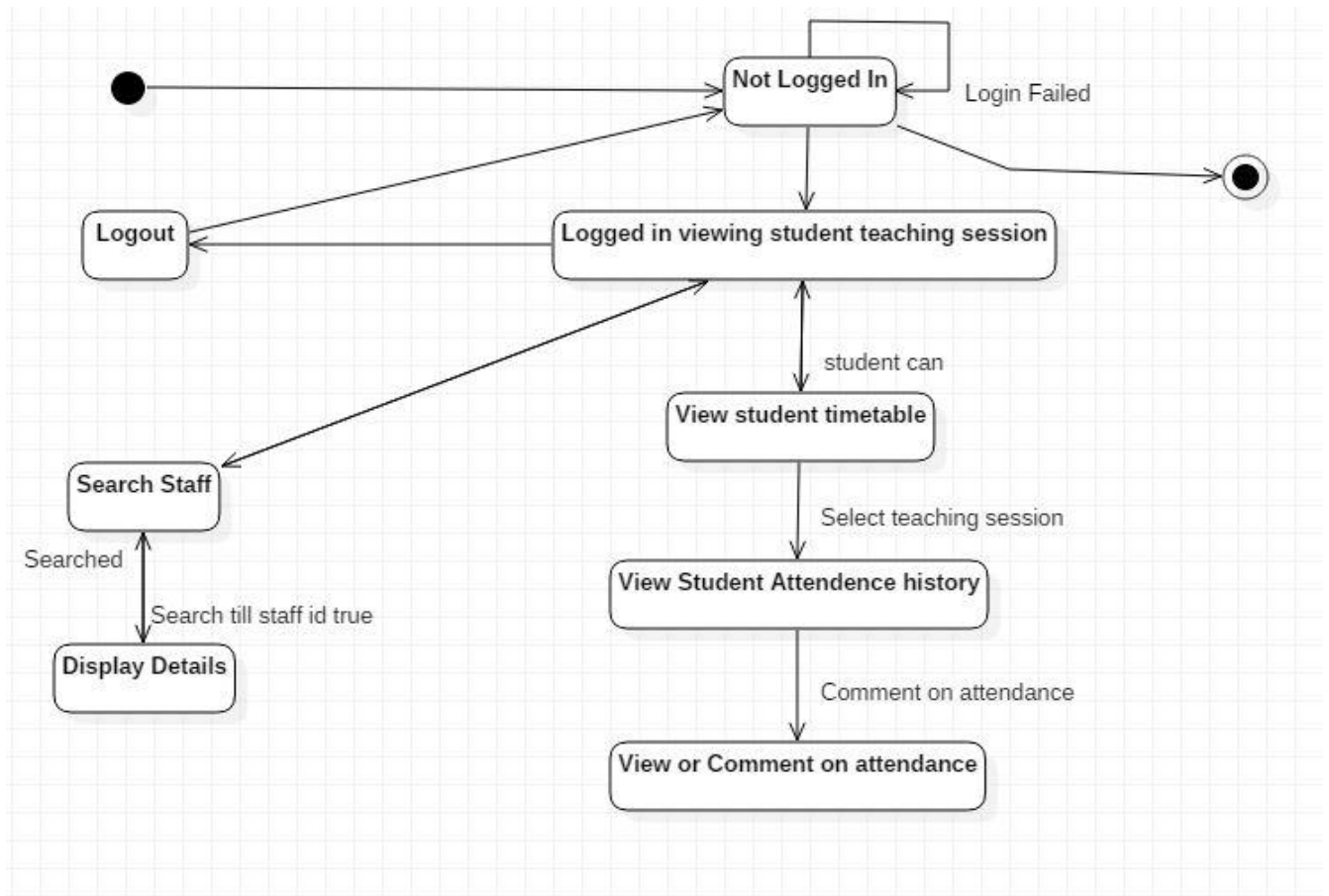
Level 2: DFD of Student Information Management System



Student Management System



4.2 State-Transition Diagrams (STD)



A. Appendices

A.1 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 from 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 deal with all types 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.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flow chart. Although the diagram looks like a flow chart but it is not. It shows different flows like parallel, branched, concurrent and single.

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.

Basic Activity Diagram Symbols and Notations

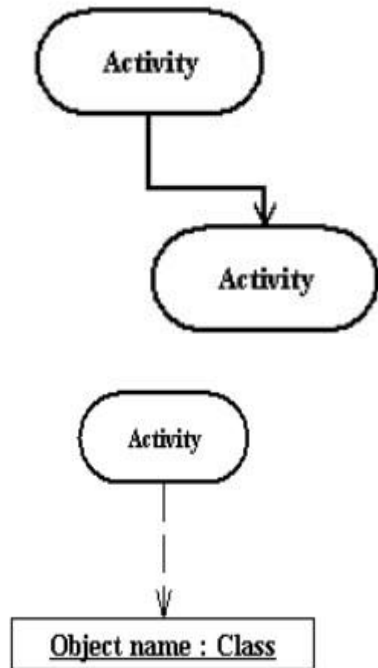
Action states

Action states represent the noninterruptible actions of objects. You can draw an action state in SmartDraw using a rectangle with rounded corners.



Action Flow

Action flow arrows illustrate the relationships among action states.



Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



Initial State

A filled circle followed by an arrow represents the initial action state.

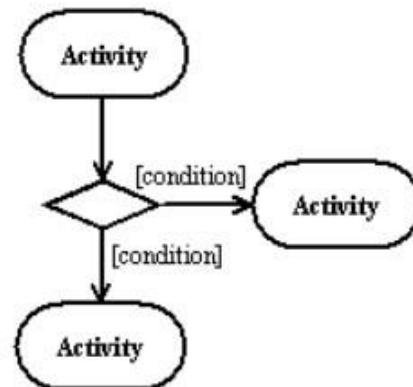
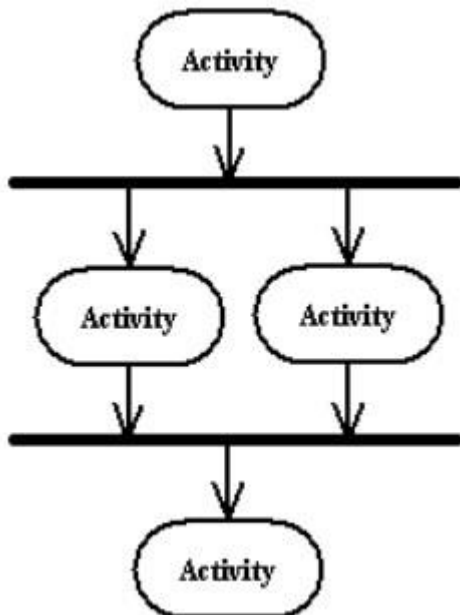
An arrow pointing to a filled circle nested inside another circle represents the final action state.



Final State

Branching

A diamond represents a decision with alternate paths. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."

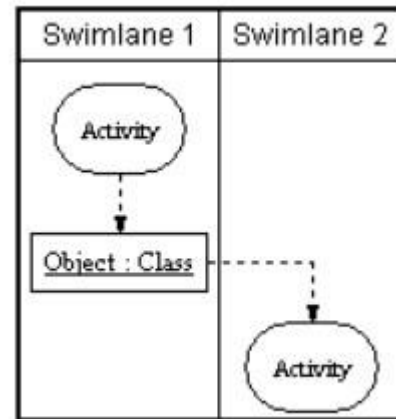


Synchronization

A synchronization bar helps illustrate parallel transitions. Synchronization is also called forking and joining.

Swimlanes

Swimlanes group related activities into one column.



Where to use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages. The activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues or any other system. Now we will look into the practical applications of the activity diagram. From the above discussion it is clear that an activity diagram is drawn from a very high level. So it gives high level view of a system. This high level view is mainly for business users or any other person who is not a technical person. This diagram is used to model the activities which are nothing but business requirements. So the diagram has more impact on business understanding rather implementation details.

Following are the main usages of activity diagram:

- ☐ Modelling work flow by using activities.
- ☐ Modelling business requirements.
- ☐ High level understanding of the system's functionalities.
- ☐ Investigate business requirements at a later stage.

A.2 Appendix 2- StateChart Diagram manual

UML Statechart 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 Statechart 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 Statechart diagram. As Statechart diagram defines states it is used to model lifetime of an object.

Purpose:

Statechart 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.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of Statechart diagram is to model life time of an object from creation to termination.

Statechart diagrams are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

Following are the main purposes of using Statechart 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.

Basic Statechart Diagram Symbols and Notations

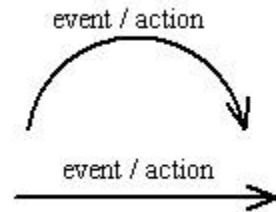
States

States represent situations during the life of an object. You can easily illustrate a state in SmartDraw by using a rectangle with rounded corners.



Transition

A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it.



Initial State

A filled circle followed by an arrow represents the object's initial state.



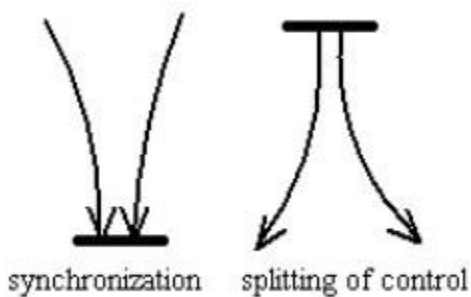
Final State

An arrow pointing to a filled circle nested inside another circle represents the object's final state.



Synchronization and Splitting of Control

A short heavy bar with two transitions entering it represents a synchronization of control. A short heavy bar with two transitions leaving it represents a splitting of control that creates multiple states.



How to draw Statechart Diagram?

Student Management System

Statechart diagram is used to describe the states of different objects in its life cycle. So the emphasis is given on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.

Statechart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs. Before drawing a Statechart diagram we must have clarified the following points:

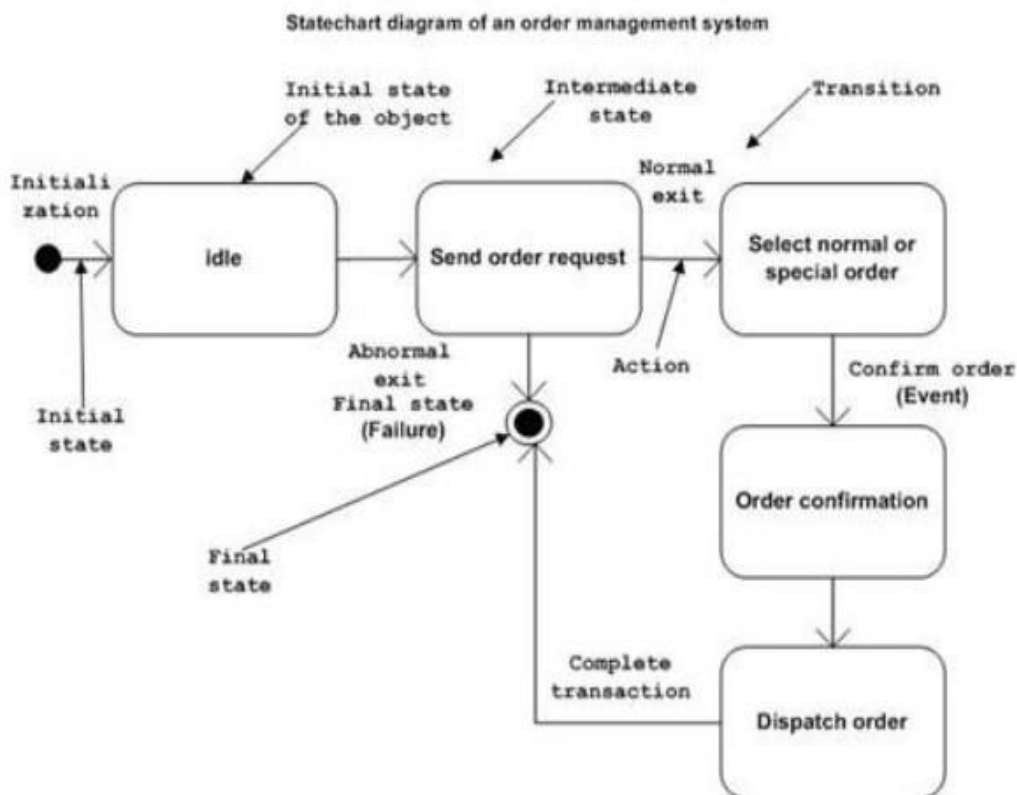
- ☐ Identify important objects to be analyzed.
- ☐ Identify the states.
- ☐ Identify the events.

The following is an example of a Statechart diagram where the state of *Order* object is analyzed.

The first state is an idle state from where the process starts. The next states are arrived for events like *send request*, *confirm request*, and *dispatch order*.

These events are responsible for state changes of order object. During the life cycle of an object (here order object) it goes through the following states and there may be some abnormal exists also. This abnormal exit may occur due to some problem in the system. When the entire life cycle is complete it is considered as the complete transaction as mentioned below.

The initial and final state of an object is also shown below.



Where to use Statechart Diagrams?

From the above discussion we can define the practical applications of a Statechart diagram. Statechart 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.

Statechart 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.

Statechart 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 Statechart diagram then it is mainly used to analyze the object states influenced by events. This analysis is helpful to understand the system behaviour 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.