

CHAPTER 1

INTRODUCTION

1.1 Introduction

Project document including Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the developed system. The aim of this document is to gather and analyze and give an in-depth insight of the complete NSTU Hall Management System by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by users and their needs while defining high-level desired features. The detailed requirements of the NSTU Hall Management System are provided in this document.

1.2 Purpose

The project "NSTU Hall Management System" is an online web-based management system that provides a simple, easy and beautiful interface for handling management of all activities in a16 university hall. The student hall of a university is one of the main places in a student's daily life, so hall management is a vital part of the university management. But it's a matter of regret that we have no digitalized management system available right now. Here everything is controlled manually. Currently the hall management authority uses handwritten paper for documentation and other works. For this the authorities and students face a lot of problems. We don't get enough information to allocate a seat for a student in the hall and manage room distribution. We have no digital record of payments of the student. In our current system, we solve our problems manually but we have no strong record, that's why some problems are not solved in time. In this case, we want to propose an online hall management system, which may help the student and authority to some extent.

1.3 Project Scope

The main objective of the development of NSTU Hall Management System is to manage the detailed information of the students, distributing rooms to the students and managing meals in dining, managing students' monthly payment and to manage hall programs using the web portal.

This SRS is also aimed at specifying requirements of the web application to be developed but it can also be applied to assist in the selection of in-house and commercial software products. The standard can be used to create software requirements specifications directly or can be used as a model for defining an organization or project specific standard. It does not identify any specific method, nomenclature or tool for preparing an SRS.

1.4 Objectives

The objectives of the “NSTU Hall Management System” are:

- To provide a digital management system for NSTU halls.
- To reduce time for solving a problem.
- To increase collaboration between students and authority.
- To digitalize the problem gathering and solving system.
- To digitalize the seat distribution system.
- To digitalize and reduce hassle for meal management.

1.5 Expected Outcome

The expected outcomes of the NSTU Hall Management System are:

- A complete digital management system for NSTU halls.
- Digital means of storing student information.
- Digitalized and Secure communication with authority and students.
- Reduction of human resource and additional cost.

1.6 Glossary

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document.

Terms and concepts from the application domain are defined.

- API – Application Programming Interface
- SRS – Software Requirement Specifications
- UI – User Interface
- SDLC – Software Development Life Cycle
- DBMS – Database Management System

1.7 References

Software & Systems Requirements Engineering: In Practice forewords by Manfred Broy, technical University, Munich, and Erik Simmons, Intel Corporation.

CHAPTER 2 OVERVIEW

2.1 Overview

To completely digitalize a university hall and provide an online-based management system in this project, every single thing will be automated (digital) and total activities will be completed using new technology. In our country, there are several universities. Our university is developing day by day, increasing with a number of students and hall rooms every semester.

Our hall management authority faced troubles in sharing rooms for students. Whereas, this was a lengthy procedure, so this spoils a lot of time, diligent is not always capable of making better use of resources. While we are living in the era of modern technology, we need to apply to solve this problem. We endeavor to solve the matter by executing a digital system and successfully we have done this. This project will help Hall authorities and students to locate rooms for students.

2.2 User Classes and Characteristics

There are 4 kinds of stakeholders in this NSTU Hall Management System.

1. User: any visitor on the website.
2. Student: residents of hall.
3. Dining Manager: an employee who manages meals for the residents.
4. Admin: the hall authority who will have the complete control of the system.

2.3 Design and Implementation Constraints

2.3.1 Interfaces

There are many types of interfaces as such supported by the web application system namely - User Interface, Software Interface and Hardware Interface.

The protocol used shall be HTTP.

The Port number used will be 80.

There shall be a logical address of the system in IPv4 format.

2.3.1.1 User Interfaces

The user interface for the web application shall be compatible with any smart device which has internet browsing feature. Also, the user interface shall be implemented in such a way that it will be displayed to any internet browsing enabled device smoothly and without missing any feature or option.

2.3.1.2 Hardware Interfaces

Since the application must run over the browser in the internet, all the hardware required to connect to the internet browser will be hardware interface for the system.

2.3.1.4 Communications Interfaces

NSTU Hall Management System shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

2.3.2 Licensing Requirements

Not Applicable

2.3.3 Legal, Copyright, and Other Notices

NSTU Hall Management System should display the disclaimers and copyright.

2.3.4 Applicable Standards

It shall be as per the industry standard.

2.3.5 Supporting Information

Following documents will be referred:

1. Vision document for NSTU Hall Management System
2. Use case analysis
3. Structural models
4. Behavioral models
5. Non-functional requirements model
6. Project Plan

2.3.5 Web Server

A web server is a system that delivers content or services to end users over the internet. A web server consists of a physical server, server operating system (OS) and software used to facilitate HTTP communication. A web server is also known as an internet server. The simplest definition is that a web server runs a website by returning HTML files over an HTTP connection. This definition may have been true in the early days of the internet, but the line has blurred between websites, web applications and web services, etc. For example, a server that delivers an XML document to another device can be a web server. A better definition might be that a web server is any internet server that responds to HTTP requests to deliver content and services.

2.3.6 Database Server

The term database server may refer to both hardware and software used to run a database, according to the context. As software, a database server is the back-end portion of a database application, following the traditional client-server model. This back-end portion is sometimes called the instance. It may also refer to the physical computer used to host the database. We will use the MYSQL database server to store all the information of this system.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the developed system. The aim of this document is to gather and analyze and give an in-depth insight of the complete NSTU Hall Management System by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by users and their needs while defining high-level desired features. The detailed requirements of the NSTU Hall Management System are provided in this document.

3.1 Requirement Collection and Analysis

3.1.1 Functional System Requirement:

This section gives a functional requirement that is applicable to the NSTU Hall Management System. These are sub modules in this phase.

- Administrator module
- User Module
- Hostile Module
- Registration Module

3.1.2 Non-Functional System Requirements:

- Performance Requirements
- Security Requirements

3.2 Use Case Modeling

3.2.1 Use Case Diagram

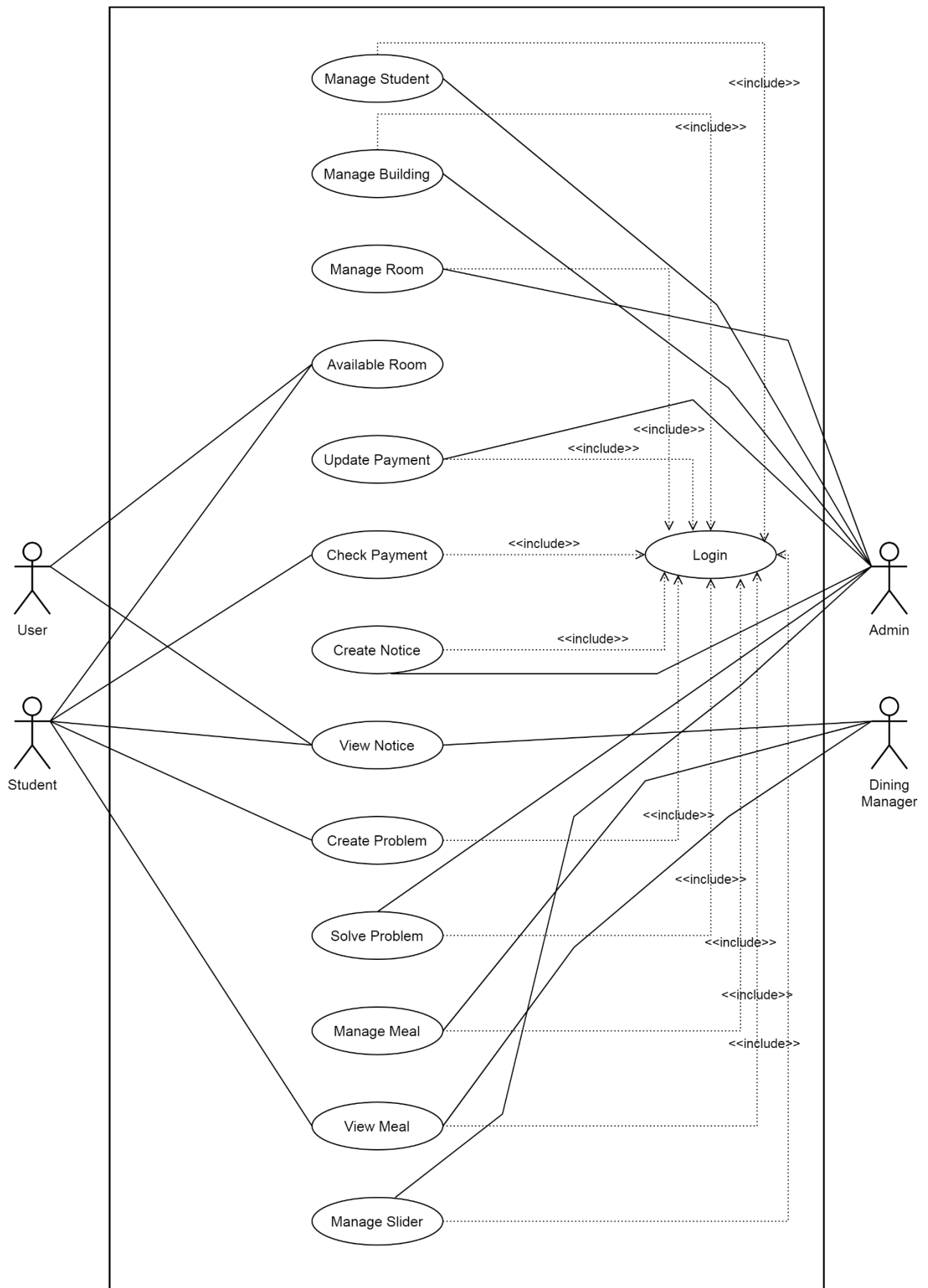


Figure 3.2.1 Use Case Diagram

3.3 Use Case Description

Table 3.3.1 Manage Student

Use Case Term	Manage Student
Actors	Admin
Flow of Events	1. Affix to Student 2. Remove Student 3. View Student Details
Substitute Flows	1. No student found 2. Do not add new student 3. Invalid Information
Pre-Condition	Login
Post-Condition	Confirm Student, Delete Student

Table 3.3.2 Manage Building

Use Case Term	Manage Building
Actors	Admin
Flow of Events	1. Affix to Building 2. Update Building 3. Remove Building
Substitute Flows	1. Chosen the wrong building 2. Building not found
Pre-Condition	Login
Post-Condition	Chosen right building

Table 3.3.3 Manage Room

Use Case Term	Manage Room
Actors	Admin
Flow of Events	1. Affix on Room 2. Update Room 3. Remove Room
Substitute Flows	1. Chosen false room 2. Delete incorrect room 3. Invalid Input
Pre-Condition	Login
Post-Condition	Select the right room

Table 3.3.4 Available Room

Use Case Term	Available Room
Actors	Admin, Student, User
Flow of Events	1. Check Available Room 2. Change/UpdateRoom
Substitute Flows	1. Chosen false room 2. Don't updated available room 3. Invalid Input
Pre-Condition	Visit Website/No Pre-condition
Post-Condition	Select Building, Select Room

Table 3.3.5 Update Payment

Use Case Term	Update Payment
Actors	Admin
Flow of Events	1. Affix to Payment 2. Update Payment 3. Remove Payment
Substitute Flows	1. Update wrong student payment 2. Don't delete running student payment 3. Invalid Input
Pre-Condition	Login
Post-Condition	Select correct student

Table 3.3.6 Check Payment

Use Case Term	Check Payment
Actors	Student
Flow of Events	1. Check current payment 2. Check payment list
Substitute Flows	1. Incorrect student id 2. Invalid Input
Pre-Condition	Login
Post-Condition	Enter Login Information

Table 3.3.7 Create Notice

Use Case Term	Create Notice
Actors	Admin
Flow of Events	1. Affix to Notice 2. Remove Notice
Substitute Flows	1. Affix wrong notice 2. Invalid Information
Pre-Condition	Login
Post-Condition	Enter notice title and description

Table 3.3.8 View Notice

Use Case Term	View Notice
Actors	User, Student
Flow of Events	1. View current Notice 2. View previous Notice
Substitute Flows	1. Selected invalid notice 2. Can't view future notice
Pre-Condition	Visit Website/No Pre-condition
Post-Condition	View notice panel

Table 3.3.9 Create Problem

Use Case Term	Create Problem
Actors	Student
Flow of Events	1. Affix to Problem 2. Update Problem 3. Cancel Problem 4. Pending Problem 5. Remove Problem
Substitute Flows	1. Incorrect student id 2. Invalid Information
Pre-Condition	Login
Post-Condition	View problem list, View reply from admin

Table 3.3.9 Solve Problem

Use Case Term	Solve Problem
Actors	Admin
Flow of Events	1. Reply 2. Delete
Substitute Flows	1. Can't ignore problem 2. Invalid reply
Pre-Condition	Login
Post-Condition	View solved problem list

Table 3.3.10 Manage Meal

Use Case Term	Manage Meal
Actors	Dining Manager
Flow of Events	1. Create Meal 2. View Meal List
Substitute Flows	1. Invalid number of meals
Pre-Condition	Login
Post-Condition	View Meal Dashboard

Table 3.3.11 View Meal

Use Case Term	View Meal
Actors	Dining Manager
Flow of Events	1. View Total Meal
Substitute Flows	Invalid Number of Meals
Pre-Condition	Login
Post-Condition	View Meal Dashboard

Table 3.3.11 Manage Slider

Use Case Term	Manage Slider
Actors	Admin
Flow of Events	1. Create New Slider 2. Delete Slider 3. Edit Slider
Substitute Flows	Can't create slider
Pre-Condition	Login
Post-Condition	View slider dashboard

3.4 Database Schema

Admins (id, f_name, l_name, email, contact, email_verified_at, password, is_active, remember_token)

Applies (id, st_application, st_id)

Buildings (id, building_name)

Members (id, name, email, position, contact, image)

Notices (id, title, file, created_at, updated_at)

Payments (id, st_id, st_semester, hall_fee)

Problems (id, p_description, st_id)

Replies (id, st_reply, problem_id)

Rooms (id, room_num, quantity, booked, building_id)

Meal (sl, lunch_meal, dinner_meal, total_meal)

Sliders (id, image)

Students (id, st_id, st_name, email, password, st_dept, image, room_id, semester_id, st_contact, remember_token)

User (id, f_name, l_name, email, password, address, contact)

3.5 Diagram of Entity Relationship

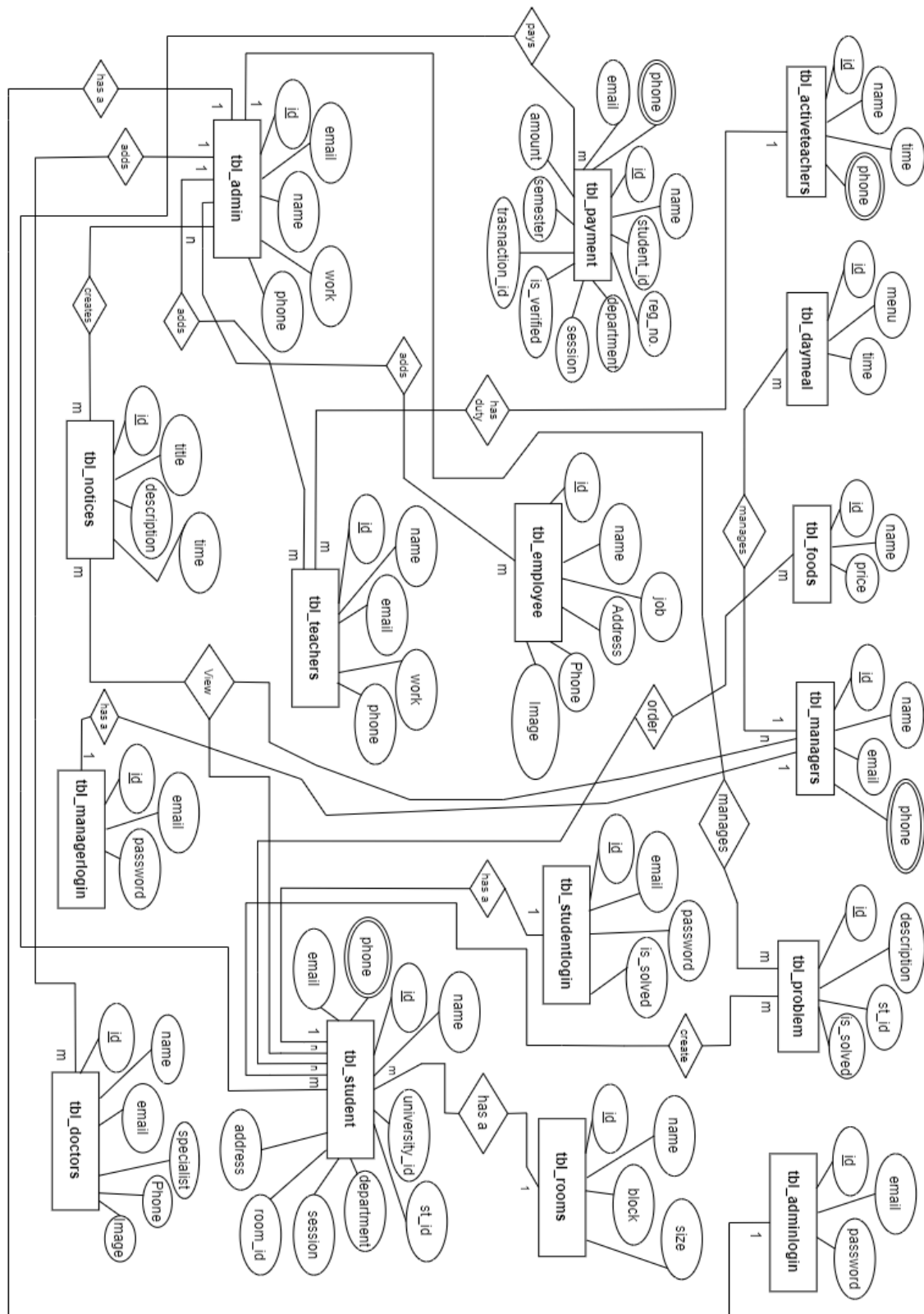


Figure 3.5.1: Entity Relationship Diagram

3.7 Data Requirements

Data requirements are prescribed directives or consensual agreements that define the content and/or structure that constitute high quality data instances and values. Here are requirements for Data:

- The system shall input, process and output data types of integers, strings, characters and real due to the variation of information that will be stored
- The system shall have accurate and up-to date information
- The system shall handle information from various sources

Firstly, we need to build a data model. We need to focus on the entity of data, quantity of data, capacity of data resource, data availability etc. For collecting data, we are following these steps:

- **Data Object Selection:** A data object is a representation of information which has different properties or attributes that must be understood by software. Firstly we find all Attributes of our System.
- **Data Objects and Attributes:** we find necessary attribute for each Entity
- **Relation between Data objects:** Now we define all relation between two entities and its attributes.
- **E-R Diagram:** Entity relationship diagram displays the relationships of entity sets stored in a database.
- **Schema Diagram:** Using E-R diagram, we draw Schema of our database
- **Create table:** Now create table and set data types it's attribute.
- **Normalization:** Normalization is the process of reorganizing data in a database so that it meets two basic requirements:
 - I. There is no redundancy of data (all data is stored in only one place), and
 - II. Data dependencies are logical

At last, we do Normalization.

3.8 Performances Requirements:

Performances are a very Important issue of our application. Performance requirements define how well the system performs certain functions under specific conditions. The Performance requirements are often based on Supporting our System end User Tasks. To ensure our System Performance, we need to maintain some issues. Here we describe the issues by which we are going to enhance the performance of our Web Application.