

A
PROJECT REPORT
ON

“Online Project Tasks Scheduling & Planning”

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

MASTER OF COMPUTER APPLICATIONS (MCA)

Under Guidance of

Mr. Chandra Prakash Pandey

Project Manager

A.P.V. India Pvt. Ltd.

Submitted By

Amrish Sinha



A.P.V. INDIA PVT. LTD.

LUCKNOW (INDIA)

2016-17

Student's Declaration

I, **Amrish Sinha**, do hereby declare that the project work titled “**Online Project Tasks Scheduling & Planning**” for partial fulfillment of the requirement for the award of degree of **Master of Computer Applications (MCA)** submitted in the Department of Computer Science, **Aligarh Muslim University, Aligarh** is an authentic record of my own work carried out during the period from **23-Jan-2017 to 23-May-2017** at **A.P.V. India Pvt. Ltd.**

The Project report has not been submitted by me for award of any other degree/diploma to any university.

Date:

(Amrish Sinha)

Place:

CONTENTS

Certificate.....	(i)
Acknowledgement.....	(ii)
Abstract.....	(iii)
Chapter 1: INTRODUCTION	1
1.1 Objective	2
1.2 Introduction	2
1.2.1 Introduction about technology	2
1.2.2 Relational database (MySQL)	9
1.2.3 Application Language	10
1.3 Project description	10
1.4 Benefits	12
1.5 About the company	13
1.5.1 A.P.V. India Pvt. Ltd.	13
1.5.2 Services	13
1.5.3 Technologies	14
Chapter 2: PROBLEM FORMULATION	15
2.1 Introduction	16
2.2 Problem definition detailed description	16
2.3 Existing System	17
2.3.1 Drawbacks of existing system	17
2.4 Scientific novelty and need of work	18
2.5 Proposed system/method solution	18
Chapter 3: SYSTEM ANALYSIS	20
3.1 System Analysis	21
3.2 Objectives	22
3.3 System development tools	22
3.3.1 System Flowcharts	22
3.3.2 Data Flow Diagrams	23
3.3.3 Program Flow Charts	23
3.3.4 Document Flow Charts	23
3.3.5 Prototyping	24
3.3.6 Entity Relationship Modelling	24
3.3.7 Case Tools	25
3.3.8 Top-Down Analysis	26
3.3.9 Decision Tables	26
3.4 Information Collection Interview	26
3.4.1 Interviews	26
3.4.2 Questionnaires	29

Chapter 4: SOFTWARE REQUIREMENT SPECIFICATION	30
4.1 Introduction	31
4.1.1 Purpose	31
4.1.2 Scope	31
4.1.3 Intended audience and reading suggestions	31
4.1.4 Product feature	32
4.2 Overall Description	32
4.2.1 Product perspective	32
4.2.2 Product function	32
4.2.3 Initial settings	36
4.2.4 Query processing	36
4.2.5 Reporting	36
4.2.6 User classes and characteristics	37
4.2.7 Operating environment	38
4.2.8 Design and implementation constraints	39
4.2.9 Assumption and dependencies	39
4.2.10 Time constraints	39
4.3 Requirements and legends	39
4.4 Requirements	40
4.4.1 Functional requirement	40
4.4.2 Security requirement	41
4.4.3 Performance requirement	41
4.5 Data Management	42
4.6 Deployment requirement	42
 Chapter 5: SYSTEM DESIGN	 43
5.1 System design	44
5.2 Design model	45
5.3 Input design	47
5.4 Output design	48
5.5 Database design	48
5.5.1 Objectives of database design	49
5.5.2 Relational schema design	49
5.6 Architectural design	50
5.7 Interface design	50
5.8 Procedural design	50
5.9 Analysis and development of actual solution	51
5.10 Description of application user interface	51
5.10.1 Interface for admin	51
5.10.2 Interface for user	53
5.11 Detailed design	54
5.11.1 Unified Modelling Language (UML) diagrams	54
5.11.2 Data flow diagram	63
5.12 Database design	65
5.12.1 Data dictionary	65
5.12.2 DBA	65

5.12.3 End user	65
5.12.4 Structure of tables use	66
5.12.5 Entity-Relationship diagram	70
5.13 Choice of language	72
5.13.1 Front End design (presentation layer)	72
5.13.2 Business logic (business layer)	74
5.13.3 Back end design (database layer)	76
5.14 Aims and objectives to be fulfilled	77
Chapter 6: SYSTEM IMPLEMENTATION	79
6.1 Introduction	80
6.1.1 Aspects of implementation	80
6.2 Hardware requirement	82
6.3 Software requirement	82
Chapter 7: SYSTEM TESTING	83
7.1 Software testing	84
7.1.1 Overview	84
7.2 Software Quality Assurance (SQA)	85
7.3 Testing levels	85
7.3.1 Unit testing	85
7.3.2 Integration testing	86
7.3.3 System testing	86
7.3.4 Regression testing	86
7.3.5 Acceptance testing	86
7.3.6 Smoke testing	87
7.3.7 Alpha testing	87
7.3.8 Beta testing	87
7.3.9 Manual testing	87
7.4 Testing Methods	88
7.4.1 White Box testing	88
7.4.2 Black Box testing	88
7.5 PASS/FAIL criteria	89
7.6 Suspension criteria and resumption requirement	89
7.6.1 Suspension criteria	89
7.6.1 Resumption requirements	90
7.7 Test Deliverables	90
7.8 Test cases and results	90
7.8.1 User login test case	91
7.8.2 Project Management test case	92
7.8.3 Component Management test case	93
7.8.4 Functionality Management test case	95
7.8.5 User Management test case	97
7.8.6 Team Management test case	99
7.8.8 Add requirement test case	100
7.8.9 Search requirement test case	101

7.8.10 Setup management test case	102
7.8.11 Schedule management test case	104
Chapter 8: Result, Discussion and Conclusion	108
8.1 Interpretation of results	109
8.2 Application areas and scope of the work	109
8.3 Conclusion	109
8.4 Advantages and special feature of the system	110
8.5 Limitation	111
8.6 Future extensions	111
8.7 Scope of enhancements	112
APPENDIX	
Data Flow Diagram	114
Context Level Diagram	114
First Level DFD	115
Second Level DFD-Requirement management	116
Second Level DFD-Setup management	116
Second Level DFD-Schedule management	117

CERTIFICATE

This is to certify that the dissertation/project work entitled “**Online Project Tasks Scheduling and Planning**” being submitted by **Amrish Sinha** in partial fulfillment of the requirements for the award of the degree of Master of Computer Science and Application, Aligarh Muslim University, Aligarh, is a record of the students own work, carried under my supervision and guidance.

Signature of the supervisor

ACKNOWLEDGEMENT

Successful accomplishment of any task is possible only with the cooperation of the people at various levels. It is very grateful to thank each of them individually. A sincere effort is being made to thank each of them.

“I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and my University. I would like to thank Department of Computer Science and all the faculty members for being with me throughout these three years. Last but not least, **Ms. Priti Bala (My mentor and supervisor at AMU) & Prof. Jamshed Siddiqui (Chairman, Department of Computer Science, AMU, Aligarh)** deserve special credit for his ubiquitous support and encouragement. I would also like to express gratitude to **all faculty members** and **office staff** for their sincere cooperation. I am also thankful to my friends and classmates for their valuable suggestions.

Next, I would like to express gratitude and indebtedness to **Mr. Chandra Prakash Pandey (Project Manager) & Mr. Ashish Varman (Sr. Project Leader)** for being my mentor at **A.P.V. India Pvt. Ltd.** throughout the completion of this project, and all my colleague for his valuable advice and guidance without which this project would not have seen the light of the day.”

(Amrish Sinha)

ABSTRACT

“Online Project Tasks Scheduling & Planning” is a web application, designed to maintain the list of various projects, its component, functionality of component, various requirements(tasks) to achieve functionality of component and planned the requirement in prioritize manner to develop a complete product.

This application maintains a record of the various project, component (module) in a project, and functionality of component in an organized way. This application keep the requirements of project which is collected during requirement gathering phase in SDLC, planned the setup version from requirements which is at high priority, prepare a weekly schedule of task from planned undispatched setup versions (those requirement which is planned to implement but not complete), track the status of task which is closed (complete), In Progress, In Testing, planned and unplanned and generate the report accordingly.

This application is implemented using a 3-tier approach with a backend database, a middle tier of J2EE and user interface in HTML as the front end client. This application is based on request-response model.

In order to develop this web based application, a number of technologies must be studied and understood. These include multi-tiered architecture, implementation technologies such as servlet, java server pages (jsp), java bean as business logic, JavaScript, Ajax, JQuery as a scripting language and MySQL as a relational database.

CHAPTER 1

Introduction

1.1 OBJECTIVE:

“Online Project Task Scheduling and Planning” is a web application through which we can maintain all the records and requirement of a developing project at one place. The objective of this project is to help the organization and employee working in an organization by maintaining the requirements of any project, planned accordingly to achieve goal in stages and update the status where we are? And next what to do?

This project act as a central repository where we can maintain all the information related the project which is developed or under developing and user get relevant information through this.

The basic objective of this project is are as follows:

- Maintain the record of all the project, its component and functionalities.
- Keep all the requirement (task) of projects in organized manner.
- Eliminate manual process of keeping record of all the project planned setup version accordingly.
- Reduce time to generate weekly schedule form requirement to develop a project.

1.2 INTRODUCTION:

1.2.1 Introduction about Technology

Project category: RDBMS

Tools/Platform:

Platform	:	Windows
Front-end	:	JSP and SERVLET
Back-end	:	MYSQL Database
Language	:	JAVA Language
Connectivity	:	JDBC

There are the following technologies used for develop this application:

- J2EE
- MYSQL Database
- Eclipse Mars

J2EE (Java Enterprise Edition)

In many ways the **Java Enterprise Edition (j2ee)** platform is a natural extension from the standard java platform into the network environment. Enterprise java is specially designed for servers, providing many advantages, including the ability to run distributed applications across different JVMs (and hence different physical servers) .To accomplish this mammoth task, it require a whole new programming architecture which alleviates some of the less interesting task from the server programmer. This allows application developers to run application on one server, or across an entire cluster of load balanced server. Additionally, the java EE platform provides ‘out of the box’ support for all standard network protocols. There are two main uses of the Java EE platform in industry:

- To provide business services such as database storage and transaction processing for client operating on a network.
- To provide Web front-ends and business services to clients using the biggest network of all the Internet.

Terminology

In Java EE there are following key terms are defined:

Web Components

A **Web component** is defined in the Java EE specification to be either a servlet or a JSP. Static resources such as HTML documents, images and other file available to the client such as PDFs and executable are not considered Web components, although they may be deployed alongside components. Additionally, metadata about the application itself, as well as all supporting classes, JAR libraries and private configuration files are not considered Web components.

Web Application

A Web application is a collection of Web components, chosen or designed to fulfill the requirements of a particular website or online services, along with other deployment or configuration data. A Web application is also known as a “**Web module**”, ‘module’ here refers to Java EE modules which collectively form a Java EE application.

Web Containers

The server handles many of the menial task of a developer for us. In order to provide these extra services, and to implement additional features such as security and filtering, the container has to manage application in a "sandbox". On the server these sandboxes are called containers. Each Java EE module type has its own container, so there are separate ones for EJBs, applet, application clients and particular focus on our studies, the Web container. The Web container is where all web applications are managed and executed. All communication within or into a Web application are always moderated by Web container. This allows the container to transpose its own services (such as protocol management and security) on top of those created by the application developer.

Features of J2EE:

J2EE (Java 2 -Enterprise Edition) is a basket of **12** inter-related technologies, which can be grouped as follows for convenience:

➤ Group-1(Web-Server & support Technologies)

- 1) JDBC (Java Database Connectivity)
- 2) Servlets
- 3) JSP (Java Server Pages)
- 4) Java Mail

➤ Group-2(Distributed-Objects Technologies)

- 5) RMI (Remote Method Invocation)
- 6) Corba-IDL (Corba-using Java with OMG-IDL)
- 7) RMI-IIOP (Corba in Java without OMG-IDL)
- 8) EJB (Enterprise Java Beans)

➤ Group-3(Supporting & Advanced Enterprise technologies)

- 9) JNDI (Java Naming & Directory Interfaces)
- 10) JMS (Java Messaging Service)
- 11) JAVA-XML (such as JAXP, JAXM, JAXR, JAX-RPC, JAXB, and XML-

WEB SERVICE)

12) Connectors (for ERP and Legacy systems).

The explanation some of the features are as follows:

JSP (Java Server Pages)

JSP technology is object-oriented programming language and is based on Java language. **Java Server Pages (JSP)** is a server side Java technology that allows software developers to create dynamically generated web pages, with HTML, XML, or other document types, in response to a Web client request to a Java Web Application container (server). JSP pages are Web pages coded with an extended HTML that makes it possible to embed Java code in a Web page. Java Server Pages (JSP) is a technology that lets you mix regular, static HTML with dynamically generated HTML. JSP page can call custom Java Classes, called *taglibs*, using HTML like tags. The JSP compiler translates JSP pages into Servlet. Server automatically compile JSP pages if the Servlet class file not present or is older than the JSP source file. You can also pre-compile JSP pages and package the Servlet class in the 'Web Archive' to avoiding compile the server. JSP pages may depend upon additional helper classes that must also be deployed with in the Web application.

The features of JSP (Java Server Pages) are as follows:

- An **Expression Language (EL)** which allows developers to create Velocity-style templates (among other things).
- It simplifies the process of development (It allows programmers to insert the Java code directly into the JSP file, making the development process easier).
- Reusability (JSP allows the component reuse by using JavaBeans and EJBs).
- Robust (JSP allows the robust platform for web development).
- Independency of Layers. There is a clear separation between presentation and implementation layers. The HTML on the web browser of the client is displayed as a presentation layer. The JSP on the server is displayed in the implementation layer.

Servlet

Servlet are module of Java code that run in a server application to answer client requests. The name "Servlets" come from the "Applets", which run on the client side whereas Servlet execute at the server side. In other words, the Servlet is the web front-end of a large Web application. Java servlets, a widely accepted technology for building dynamic content for web-based applications.

Servlets are Java programming language objects that dynamically process requests and construct responses. The **Java Servlet API** allows a software developer to add dynamic content to a Web server using the Java platform

A Servlet is a Java class that runs in the Server machine on an HTTP request from client and creates an html as response to the client. Servlets are stored in a Servlet Container in the Web Server and are mapped to URLs, for which the servlet constructs response. The features of the Servlets are as follows:

- **Ease of development** (Ease of development is one of the key success mantras of any technology. This means that you can swiftly develop a servlet or a filter class by simply annotating the class with appropriate annotations like `@Servlet` or `@Servlet Filter`).
- **Plug ability and extensibility.**
 - The ability to receive data from a client without blocking even if the data is slow arriving (non-blocking input).
 - The ability to send data to a client without blocking, even if the client or network is slow (non-blocking output).
 - The ability to handle delayed response close; i.e., the response will be held open to allow additional data to be sent when asynchronous events occur.
 - The ability to notify blocking or non-blocking events.
- **Several web.xml convenience** (Servlet 2.5 introduces several small changes to the web.xml file to make it more convenient to use.
- **A Handful of removed restrictions** (Servlet 2.5 removed a few restrictions around error handling and session tracking.

HTML:

Without HTML, the World Wide Web wouldn't exist. HTML allows the individual elements on the Web to be brought together and assented as a collection. Text, images, multimedia, and other files can be packed together using HTML. This section explains the basic principles behind the interaction between HTML and the World Wide Web.

JAVA SCRIPT:

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

JQUERY:

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto – **Write less, do more.**

jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery –

- **DOM manipulation** – The jQuery made it easy to select DOM elements, traverse them and modifying their content by using cross-browser open source selector engine called **Sizzle**.

- **Event handling** – The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- **AJAX Support** – The jQuery helps you a lot to develop a responsive and feature-rich site using AJAX technology.
- **Animations** – The jQuery comes with plenty of built-in animation effects which you can use in your websites.
- **Lightweight** – The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).

AJAX:

AJAX stands for Asynchronous **JavaScript** and **XML**. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and JavaScript.

Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

Conventional web applications transmit information to and from the server using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.

With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.

AJAX is a web browser technology independent of web server software.

A user can continue to use the application while the client program requests information from the server in the background.

1.2.2 Relational Database (MySQL):

Introduction: MySQL is the world's most popular open source database software. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

Features: MySQL offered MySQL 5.5 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server include the following features:

- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- True Varchar support
- Information schema
- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Embedded database library
- ACID compliance when using transaction capable storage engines
- Partitioned tables with pruning of partitions in optimizer

- Hot backup under certain conditions.

1.2.3 Application Language:

The Java Enterprise language is an application development language that allows you to solve a range of application problems with much less effort than earlier language technologies.

1.3 PROJECT DESCRIPTION

This project deals with automation of development operations required to convert project into product and keeping all the task and record related to project, and scheduling the task and generate report accordingly.

This project can be classify into three basic module.

1. Data Administrative
2. Project Management
3. Project Development

Data Administrative:

Admin performs major data management or administrative functionalities which are as follows:

- Add project details
- Update project details
- Delete project details

- Add project component details
- Update project component details
- Delete project component details
- Add component functionality
- Update component functionality

- Delete Component functionality
- View project, project component and functionality details
- View projects tasks
- View Schedule details
- View feedback
- Generate report
- Perform User management and team management for a project.

Project Management:

Project Manager, Project Leader and Team Leader of project can perform operations related to project management activity which are as follows:

- Add requirements (task) in project gathered from client or during requirement gathering.
- Update all the task if necessary.
- Planned the setup version.
- Schedule the task to developer accordingly.
- Generate the status report of task.

Project Development:

Developer of the project perform activities related to project development which are as follows:

- View project requirements.
- View Schedule.
- Send feedback to client if necessary.

1.4 BENEFITS

While our customers are our best references when it comes to listing out the numerous advantages and benefits of deploying a CLI Software Solution, we've documented some of them below.

Technology

- Enterprise level software
- Cohesive platform and database
- A strong & robust infrastructure
- Low maintenance technology
- Easy to integrate to legacy and other 3rd party applications
- Quick & responsive system performance
- Designed to run on a variety of operating systems
- Mouse driven with, in many cases, right-click and drag & drop functionality

Methodology

- A formal methodology for system implementation
- Native data integrity from the time it is written to the time it is referenced
- Product development services to tailor modules to meet specific needs
- Continuous application development for product expansion

Application

- Vibrant, Visual and Multi-Faceted screens facilitate the user experience
- Real-time system updating across all departments
- Real-time visibility through our robust web modules
- A dynamic report writer for fast on demand custom reporting
- Simple techniques available to assist in the import of data

Pricing

- Low start-up costs with our subscription pricing model
- Low IT infrastructure costs with our secure hosting option

- No charge annual system upgrades
- Traditional on-premise deployments available
- Tailor-made pricing plans to meet your budget
- User based licensing models with no transaction fees
- Competitive rates for professional services

1.5 ABOUT THE COMPANY:

1.5.1 A.P.V. India Pvt. Ltd.

A.P.V. India Pvt. Ltd. is ISO 9001: 2008 Certified Company, Specializes in the design, development, operation, supply, support and Total management of computer software.

It provide high quality solutions and support by way of software products and services to the customers after understanding their needs and to continually review our systems to achieve more and better, develop a set of values and attitudes, a positive team work and a mindset that enable innovation and successful outcome.

1.5.2 Services

We provide products and solutions ranging from website development to complex web based portals and desktop based software applications.





























We deliver comprehensive products, services and support to meet needs in various sectors: Oil & Gas, Education, Finance etc.

We base our solutions on a critical understanding of clients' business requirements.

We combine business knowledge with technology competence and proven methodologies to deliver high quality results in a cost effective manner to maximize your competitive advantage and productivity.

We support a full cycle development process from requirements definition and specifications, architecture, design coding, testing & validation to product maintenance and support.

1.5.3 Technologies

Platforms	Languages	Web technologies
 Windows	 Java	 J2ee
 Linux	 C++	 Node.js
 Mac X OS	 .net	 .net
 Android	 C++	 HTML5 ,  CSS3 ,  Bootstrap
	 C# .net	 Aptana Studio ,  Inkscape
Database	Cloud Solutions	Testing
 SQLite	 Azure	 Hawkeye
 Mariadb	 IBM	 Bugzilla
 Mysql	 Amazon	 Dottrace
 Sql Server		 JvisualVM

CHAPTER 2

Problem Formulation

PROBLEM FORMULATION

2.1 INTRODUCTION

Problem introduction or problem statement is the starting point of the software development activity. The objective of this statement is to answer the question: Exactly what must the project do? The software project is initiated by the user's needs. In the beginning, these needs are in the minds of various people in the user's organization. The analyst has to identify the requirements by interviewing these people and understanding their needs. It goes without saying that an accurate and through understanding of software requirements is essential to the success of a software development effort. All further development like System Analysis, System Design, and Coding will depend on how accurate and well understood the requirements are. Poorly analyzed and specified software will disappoint the user and bring grief to the developer, no matter how well designed and well coded the software is.

Software requirement appears to be a relatively simple task, but appearances are often deceiving. Chances of misinterpretation are very high, ambiguity is probable and communication gap between customer and developer is bound to bring confusions. Requirement understanding begins with a clear and concise heading stating in a sentence the task to be performed. Then requirements are described in a technical manner in precise statements.

2.2 PROBLEM DEFINITION DETAILED DESCRIPTION

Almost every activity in the world is controlled by computer driven software programs. This trend was first accommodated by engineering applications in the past. However, as the life style become more and more complex, every area of human interactions was invaded by various software systems, such as real time, simulation, embedded, web based, personal and more recently, artificial intelligence software etc.

According to the above facts, managing and maintaining the various aspects of developing project could also be controlled by efficient software. This project focuses attention on designing efficient and reliable software which controls the operations of rising project.

In real world, there are various IT organizations whose main area of focus is software development. Before any product is developed, requirements are gathered from various methods like discussion with client, questionnaire, interviews, users review etc. It is very difficult to store and maintain all these requirements by manual process and convey this information to software developer timely and what is the status of task whether it is completed successfully or in progress. There are various problems occur which are as follows:

How to maintain and store a large scale of requirement?

How to plan the requirement which is taken to first?

How to schedule the requirement for developers and convey to developer?

How to track status of requirement daily basis?

How to prepare report?

Manually processing above mentioned aspects are time consuming, less secure and increase the development time of the software product. This is a very big problem for IT Organizations to deliver the product timely. Computerizing all the above process reduced some amount of time and eliminate the manual process.

2.3 EXISTING SYSTEM

The existing system is doing some processes manually. There was a need of adding functionality of new divisions of organization. The personnel should refer all the records kept for years ago to simply know details. This is tedious and time consuming. This process is so difficult when the number of products increases and to search information about a particular project.

2.3.1 Drawbacks of Existing System:

- Sluggish search of details
- Large amounts of requirement of various project are to be kept in a separate file.
- Manually scheduling and planning on excel.
- Complicated procedure

- Information mishandling
- Tedious report Generation
- Time constraints were ignored

2. 4 SCIENTIFIC NOVELTY AND NEED OF WORK

Although there was no existing system, if we supposed to do it manually, they had to face a lot of problem to manage the system. Manual system use human beings as processors. Human beings are valuable processors because human beings can think, analyze and learn. They have the capability to put together all the information they have and draw inferences. But in spite of these qualities we cannot fully rely on human for the functioning of the system because of the following disadvantages.

- **Time Consuming:-** Doing very large amount of work manually is a very time consuming task as the speed of the human is limited, they are not machine so that can't do a large amount of work for long hours without rest. Even the most ideal working situations, the speed of human being is insufficient for processing large amount of data.
- **Tedious:-** Doing lot and lots of paper work is very tedious task as it is very slow and doing same thing every day looks very dull and boring.

2.5 PROPOSED SYSTEM/METHOD OF SOLUTION:

If we form a computerized system with a proper planning and analysis then this problem can be solved. All would be done at the click of mouse. No need to keep records on papers as all the information would be stored in the database. Developers can view the requirements and schedule online and not need to go to the file or excel to view. Management of project will become very easy by storing the requirement of particular project in a relational database and generating the schedule for developer, generating the report status of task is done on mouse click and searching of information is very fast.

FEATURES

- Computerization results in a faster, reliable and efficient way of satisfying the formation needs of user as well as the organization.
- This website gives error free and accurate information.
- The system is user friendly and built in web application environment
- It reduces data redundancy.
- At a click of mouse/button we can get complete information of books and users
- Updating is very much user friendly and easy.
- The burden of manual work is reducing because of computerization of the system.

CHAPTER 3

System Analysis

SYSTEM ANALYSIS

3.1 SYSTEM ANALYSIS

System Analysis is the examination of the problem. It is concerned with identifying all the constraints and influences. It deals with the data collection and a detailed evolution of present system. The process of the system analysis in project development is started with “**Problem Analysis**”

PROBLEM ANALYSIS

Before we start trying to solve a problem it's important to study the existing system before embarking on major changes.

As according to the existing system, the IT Organizations maintains files to store information related to development of project such as storing requirements of project, planning the requirements how to implement it on excel or create manually project plan on excel, planned the setup versions which setup is released on which date, scheduling the task according to planned setup versions and finally generating task status report. A lot of time is wasted in manually maintaining these files. With the increase in clients requirements and increase the demands of product, manual maintenance of files will be too difficult. Therefore, the management is planning to develop an automated system that will ease the IT Organizations software development procedure and ensure easy maintenance of information.

The IT Organizations needs user friendly Web/GUI based interface which provide faster data processing, proper management of information related to project development and proper report generation whenever it required.

The management of **IT Organizations** has decided to develop an “**Online project task scheduling and planning**” application, which will change the entire software development process to an automated Web-based system. Using **Online project task scheduling and planning** searching information related to project management and project development became very easy and there is no communication gap between developers and project manager, project leader etc.

3.2 OBJECTIVES

- To enhance automation of data management operations of software development providing speed and cost benefits.
- To increase operational efficiency.
- To integrate with existing systems.
- Maintains project requirement task and project setup date and weekly plan.
- Proper schedule generation and report generation.
- Proper maintenance of task status.

3.3 SYSTEM DEVELOPMENT TOOLS

3.3.1 System Flowcharts

3.3.2 Data Flow Diagrams

3.3.3 Program Flowcharts

3.3.4 Document Flowcharts

3.3.5 Prototyping

3.3.6 Entity Relationship Modeling

3.3.7. Case tools

3.3.8 Top-Down Analysis

3.3.9 Decision Tables

Some of the most widely and not-so-widely used tools for uses by auditors in documenting systems are briefly described below.

3.3.1 System Flowcharts

Systems flowcharts are graphic illustrations of the physical flow of information through the entire system. A systems flowchart is commonly used in analysis and design. Flow lines represent the sequences of processes, and other symbols represent the inputs and outputs to a process. User use system flowcharts to describe the computerized processes, manual operations, and inputs and outputs of an application system. ‘

3.3.2 Data Flow Diagrams

Data flow diagrams present the logical flow of information through a system in graphical or pictorial form. Data flow diagrams have only four symbols, which makes useful for communication between analysts and users. Data flow diagrams (DFDs) show the data used and provided by processes within a system. DFDs make use of four basic symbols.

1. Open-ended rectangles: Data collection points (e.g. accounts receivable master file)
2. Ovals: Processes (e.g. order entry)
3. Open ended rectangles with “horns”: Interfaces with other applications (e.g. a billing interface with the accounts receivable processing system).
4. Rectangles: Processing reports (e.g. schedule of A/R)

The purpose of data flow diagrams is to track the flow of data through an entire system by representing data and processes that are critical to the understanding of a system.

3.3.3 Program Flowcharts

Program flowcharts illustrate how individual computer programs work. That is, a program flowchart will show in detail each processing step of a computer program. Exhibit 4 depicts the master file update process in a batch processing system. As shown, the transaction record number is compared to the master file record number. When the two numbers match, then the master file account balance (MF_AMT) is updated. This process continues until all master file transaction file records are read.

3.3.4 Document Flowcharts

A document flowchart displays the flow of documents between organizational units. The chart is divided into several columns separated by vertical lines. Each column represents an organizational unit, such as a department, section, or employee. The flowchart shows the movement of a document from one department to another by a flow line connecting the document symbol in each department.

3.3.5 Prototyping

Prototyping is the creation of a shell template of a system. In prototyping an information system, only sections of the system are modeled with emphasis on user interfaces such as screens, menus, source documents, and reports. This emphasis ensures that the user approves of the output. It is important that users understand that the modeling and building of the data underlying the shell is a time-consuming and critical portion of system development.

3.3.6 Entity Relationship Modeling

The Entity-Relationship Diagram (ERD) is used to illustrate the logical components of a database schema (complete enterprise level data) or subschema (application view of the data). *An ERD contains* entities, relationships, attributes.

Entities are objects (persons, places, things, or events) which comprise the data of the database. Entities must be unique, such as a vendor. Relationships are links that exist between or among entities. An example of a relationship between vendors and invoices is inclusion (e.g., vendors include invoices with shipments). The vendors are entities, and the invoices are entities, and “include” describes the relationship between vendors and invoices. Attributes modify or describe both entities and relationships. Attributes are the information about an entity that we are interested in capturing and processing. Examples of attributes of a vendor include: name, address, account number, accounts payable balance due. An example of an attribute of the “include” relationship is date of shipment.

Symbols used in the ERDs are: (1) rectangles to represent entities; (2) diamonds to represent relationships; (4) lines connecting the entities and the relationships; and (4) 1, M, or N on the lines represent the entity occurrences. Entity occurrences can be one to one (1 to 1), one to many (1 to M), and many to many (M to N). The 1s, Ms, and Ns are placed on straight lines between the diamonds and the rectangles. Exhibit 7 illustrates a generic ERD along with an ERD for a bank. It can be read as follows: each customer is the registered owner of one auto, the bank holds titles on many autos, the bank has a creditor relationship with many customers, the customers have a debtor relationship with many banks. In addition to entities, relationships, and entity occurrences, the attributes associated with each entity may be added to the ERD.

3.3.7 Case Tools

The automation of methodology that assists in systems development and software engineering is called computer aided software engineering (CASE). The term “software engineering” is associated with a few well-known methodologies such as Warnier-Orr Diagrams and data flow diagrams. Systems analysis and design of business application systems, including accounting systems, once used methodologies that were primarily based on pencils and templates. Recently developed methodologies, such as CASE, are employing design automation techniques linked to code generators as well as computer-aided planning and analysis. CASE has been defined as a corporate philosophy that imposes the engineering discipline on the development of application software. CASE focuses on the entire systems development life cycle (SDLC) from preliminary design to analysis to implementation to maintenance. Creating application software involves numerous stages that overlap. Stages of software development include many different activities and require different tools and techniques from the point of conception through the maintenance of the finished system. The features of CASE tools assist the developer in all stages of the SDLC: planning, system analysis, system design, system implementation, operation, and maintenance.

An ideal CASE system would have an integrated set of tools and features to perform all aspects in the life cycle. The following are features that various CASE products possess:

1. Repository/Data Dictionary
2. Computer Aided Diagramming Tools
4. Word Processing
4. Screen and Report Generator
5. Prototyping
6. Project Management
7. Code Generation
8. Reverse Engineering

3.3.8 Top-Down Analysis

In top-down analysis, the analyst begins with an overview of the entire system and gradually progresses until details at the lowest level are understood. This is an interactive process such that the analysis, design, coding, testing, and installation steps occur at each level. The greatest benefit to top-down analysis is that the difficult interface bugs are found early in the development process rather than at the end.

3.3.9 Decision Tables

Decision tables help a person understand a system process by separating a complex problem into its underlying conditions and actions. Each decision table has four sections; the upper two presenting the conditions, and the lower two presenting the actions. The benefit provided by the decision table is that all possible conditions and their actions can be considered.

3.4 INFORMATION COLLECTION-INTERVIEW

The specific methods that are used for collecting data about the requirements of the new system are called fact finding techniques. These techniques include interviews, questionnaires, record inspection and observations. Here in both the techniques were used for finding out whether there is a need of converting the existing manual handling in to an online computerized system.

3.4.1 Interviews

Interviews are a fact finding technique where interviewer collects information from individuals. The respondents are generally current users of the existing system or potential users of the proposed system. In our case, the respondents were the managers or employees who provided data for the proposed system and who would be affected by it. Specifically structured technique was followed for the interview. In structured technique, the interviewer has the predefined set of questions of which the user has to answer.

3.4.1.1 Questions That Need To Be Asked During Analysis

There's not much good getting heavily into a project if the whole thing is a silly idea to start with. The preliminary investigation is an early test of whether the project should even be started.

3.4.1.2 Is There Really A Problem?

Imagine owner thinking his organization was losing sales because employees cannot work with computerized system. He spends thousands of dollars and hundreds of man-hours to implement computerized system. After it's launched, no-one actually uses it because they were happy with the old system. This is only waste of time and money.

3.4.1.3 If There Is A Problem, Is It Worth Fixing?

Assuming a valid problem has been identified, you must consider a "cost/benefit" analysis. In other words: is fixing the problem worth it?

A manager decides the company logo and letterhead looks old fashioned. He spends thousands on graphic design consultants, market research, psychological testing of consumer reactions; he recalls and destroys all existing stationery and company publications and reprints them with the new logo and letterheads; he has sign writers repaint all company vehicles and buildings. It cost millions and it simply not worth it.

Determining whether a problem is worth fixing involves a **feasibility study**. The aim of the feasibility study is to understand the problem and to determine whether it is worth proceeding.

There are five main factors to be considered:

- **Technical feasibility** - investigating whether the technology exists to implement the proposed system, or whether this is a practical proposition. For the design and development of the system, several software products have been accommodated.

Database design- Maria DB/MySQL

Interface design- HTML, CSS

Coding- Java and other supported Language like JavaScript, JQuery, Ajax.

This software's have the enough efficiency in producing the system. Therefore the project is technically feasible.

- **Economic feasibility** – According to the resources available and the project scheduling process it is estimated that the expenses allocated for the software to be developed, by the customer is sufficient enough. If the benefits do not outweigh the costs, then it is not worth going ahead.
- **Legal feasibility** - determines whether there is any conflict between the proposed system and legal requirements - for example, will the system contravene the Information Privacy Act?
- **Operational feasibility** - Operational feasibility is concerned with whether the current work practices and procedures are adequate to support the new system. It is also concerned with social factors - how the organizational change will affect the working lives of those affected by the system.

By automating the process of data management of software development process, IT Organizations will get very quick service by reducing the manual maintenance of data. Developers will feel comfortable by reduction of time of getting information of their work. Recording errors and bugs will be reduced and locating errors, bugs easily. Easy to handle large database. Loosing of records will be avoided.

Considering all these factors we can conclude that all the users and end users will be satisfied by the system.

- **Schedule feasibility** – The duration of time required for the project has been planned appropriately and it is the same as the duration of time expected by the customer. Therefore the product can be delivered to the customer within the expected time duration, satisfying the customer. Hence the project is feasible in scheduling.

To answer these questions, data has to be collected about the system. You might need to:

- Measure things, like how long different processes take, how much output is produced in a given time, how many staff are required.
- Count things, like the numbers of errors or system failures in an existing system.
- Survey or interview workers, management, customers, and corporate partners to discover what these people know about the system's requirements, strengths or weaknesses from their specialist perspectives
- Observe processes in action to see where problems lie and improvements can be made in work-flow, and consider how procedures should be changed to accommodate the new or changed

system. If processes take place outside your organization, be sure to include them too - you never know: the problem might not be in your organization at all!

- Research similar systems elsewhere to see how similar problems have been addressed
- Test the existing system to determine whether suspected points of weakness are real or imagined
- Study the workers in the organization and list the types of information the system needs to produce for each type of worker

Such operations often create more questions that also need to be answered. Analysis often turns up issues that need to be investigated with further analysis. By the time you are finished analyzing the problem, you should have a clear idea of:

The context of the problem - How the system fits into its surroundings, which includes people (in your organization and perhaps in other organizations) and other systems which interact with the system. You need to understand how input, processing and output data are managed in the organization - including data and information destinations. An IPO chart and workflow diagram would be good to document these relationships.

- **The processes** - you need to know how data is transformed into information. What tasks are performed? A top-down approach makes sense: identify major processes, and keep subdividing them up into smaller processes until individual tasks are listed.

When an existing system is well understood, and the needs of the new or changed system have been defined, a design for the solution is not far away

3.4.2 Questionnaires

Questionnaires are special purpose documents that allow the analyst to collect information and opinions from respondents. When it becomes difficult because of time, distance and cost constraint, to interview all the desired people involved in the system,

The specific methods that are used for collecting data about the requirements of the new system are called fact finding techniques. These techniques include interviews, questionnaires, record inspection and observations. Here in both the techniques were used for finding out whether there is a need of converting the existing manual handling in to an online computerized system.

CHAPTER 4

Software Requirement

Specification

4.1 INTRODUCTION

4.1.1 Purpose

This document represents the various technical and functional requirements analysis of Online project task scheduling and planning system and business process requirements of client. This document is produced prior to detailed design and development of the application. It will be used by the design team as the baseline for establishing systems design and ultimately the development of the system.

4.1.2 Scope

The scope of online project task scheduling and planning system is to provide efficient data management and faster searching technique of requirement related to project. It also providing a common global platform for Clients, Administrator (Data Administrator) and Operator (Project Management and Project Development Level User).

The requirement specified in this document cover the entire necessary and mandatory feature of application required by the client. The data captured in this process will aid the testing process during unit testing and deployment. The software is required to fulfill all the functional requirements to identify by this document.

4.1.3 Intended Audience and Reading Suggestions

Developers-

A Software developer is a person or organization concerned with facts of the software development process wider than design and coding, a somewhat broader scope of computer programming or a specialty of project managing including some aspects of software product management.

Project Managers-

The person with authority to manage a project. Responsibilities usually include managing the budget and work plan, planning, performance and all project management procedures. Responsibilities may vary widely depending on the company or organization and may be specified in the project charter.

Marketing Staff

Users

Testers

Documentation Writers

4.1.4 Product feature

- To enhance automation of data management operations of software development providing speed and cost benefits.
- To increase operational efficiency.
- To integrate with existing systems.
- Maintains project requirement task and project setup date and weekly plan.
- Proper schedule generation and report generation.
- Proper maintenance of task status.

4.2 OVERALL DESCRIPTION

4.2.1 Product Perspective

The IT Organizations needs user friendly Web/GUI based interface which provide faster data processing, proper management of information related to project development and proper report generation whenever it required.

The management of **IT Organizations** has decided to develop an “**Online project task scheduling and planning**” application, which will change the entire software development process to an automated Web-based system. Using **Online project task scheduling and planning** searching information related to project management and project development became very easy and there is no communication gap between developers and project manager, project leader etc.

4.2.2 Product Function

The main functionality of this software solution namely “Online Project Task Scheduling and Planning” are classified into three parts:

Data Management

Project Management

Project Development

Data Management

Admin of the organizations performs data management functionality which includes project data management, project component data management, component functionality data management, user data management, team management, feedback analysis etc.

Project Data Management

Add Product Details: Admin add the product details information like project name, project code, project description, purpose etc.

Update Product Details: Admin can update the product details except project name and project code.

Delete Product Details: If necessary, admin can remove the particular product.

Project Component Data Management

Add Component Details: Admin add the component/module details in any project like component name, description etc.

Update Product Component Details: Admin can update the component details.

Delete Product Details: If necessary, admin can remove the particular component from product.

Component Functionality Data Management

Add Functionality Details: Admin add the functionality details in component/module of any project like functionality name, description, functionality etc.

Update Functionality Details: Admin can update the functionality details of any module.

Delete Functionality Details: If necessary, admin can remove the particular from component/module of product.

User Data Management

Add user: Admin can add the user (employee) details working in an organizations like username, user first name, user last name, userid, password, user type, user designation, address, email id, user working status etc.

Update user: Admin can update the particular user details and change user status according to his present employment means he is working employee or switched employee.

Delete user: Admin can delete the particular user record.

Team Data Management

Add User To Team: Admin can form the project team from the user information. He/She can able to add user to any project team.

Update Team Details: Admin can update the team details.

Delete Team Details: Admin can remove the user from particular team.

View Data

Admin can view the following data:

Product details

Component details

Functionality details

Product requirement details

Schedule details

Product Setup version details (dispatched/un-dispatched)

User details

Team details

Feedback Analysis

Admin can analysis the feedback sent by user (PM, PL, TL, Developer) in particular product or particular module of project etc.

Project Management

Project Manager (PM), Project Leader (PL), Team Leader (TL) can perform project management functionalities such as requirement gathering of project and add requirement in project, planned the requirement in setup version, generate the schedule from planned setup version, generate the report accordingly etc.

Project Requirement Management

Add requirement: PM, PL, TL can add the requirement in any project gathered from some technique which include Task Id, Task, module, functionality, start date, date of close, planned setup version, planned setup date, user, reported by, task status, comments etc.

Update requirement: PM, PL, TL can update the requirement details and can change the task status whether it unplanned, planned, in progress, in testing or closed.

Delete requirement: PM, PL, TL can delete the unnecessary requirement from functionality if added.

Project Setup Plan Management

Add setup version: PM, PL, TL can add the setup version in any project which include id, setup version name, setup date, status (dispatched/un-dispatched), requirement etc.

Update requirement: PM, PL, TL can update the setup version details.

Delete requirement: PM, PL, TL can delete the setup version if wrongly added by mistake.

Project Schedule Management

Create Schedule: PM, PL, TL can create weekly schedule for project from the un-dispatched setup version task. If any schedule task is not complete, it can carry forward in next weekly schedule. In creation of weekly schedule, task is assigned to user (Team member of project) in particular week and this way requirement is implemented by making proper schedule. Schedule entries involve id, name, task id, task, planned start and end date of task, actual start and end date of task, comments and task status.

Update Schedule: PM, PL, TL can update the schedule details.

Delete schedule task: PM, PL, TL can delete the particular task from a schedule.

Project Report Management

PM, PL, TL can generate report of task status whether it is completed successfully or it is in progress.

Project Development

Software developer of the organization can perform development activities. For this all the necessary information is easily available on the application developer's account. It can be easily retrieved by efficient searching technique like requirement filter, rows template, schedule template etc.

4.2.3 Initial Settings

Even before the entry of project requirement, Task Id is initially created and it is auto generated from product code concatenated with auto incremented number.

Product Id and Product Name is initially set at login time.

User Id and password is automatically fetched by system at login process.

4.2.4 Query Processing

Project is fetched by product id.

Requirement is fetched by setup version or start date and end date of setup version or requirement id.

Schedule is fetched by schedule id.

4.2.5 Reporting

This application generates two types of report.

Task Status Report- Distinguished by color coding according to task.

Schedule Report- An excel sheet is generated of a particular project weekly schedule. It can be classified into two types.

Weekly Schedule Report: An individual product weekly schedule.

Consolidated Weekly Schedule: A whole product weekly schedule along with project weekly feedback and setup plan.

4.2.6 User Classes and characteristics

This project can be classify into three basic user module.

4. Data Administrative
5. Project Management
6. Project Development

Data Administrative:

Admin performs major data management or administrative functionalities which are as follows:

- Add project details
- Update project details
- Delete project details

- Add project component details
- Update project component details
- Delete project component details
- Add component functionality
- Update component functionality
- Delete Component functionality

- View project, project component and functionality details
- View projects tasks
- View Schedule details
- View feedback

- Generate report
- Perform User management and team management for a project.

Project Management:

Project Manager, Project Leader and Team Leader of project can perform operations related to project management activity which are as follows:

- Add requirements (task) in project gathered from client or during requirement gathering.
- Update all the task if necessary.
- Planned the setup version.
- Schedule the task to developer accordingly.
- Generate the status report of task.

Project Development:

Developer of the project perform activities related to project development which are as follows:

- View project requirements.
- View Schedule.
- Send feedback to client if necessary.

4.2.7 Operating Environment

Server Side

Operating System: Windows 8.1 or higher

Application Language: Enterprise java/jdk-1.7

Web Server: Glassfish server

Client Side

Operating System: Any

Browser: Google Chrome/Mozilla Firefox

4.2.8 Design and Implementation Constraints

- For a web based application its design depend on the browser so the CSS should be cross browser in nature.
- Must have windows in deployment platform.

4.2.9 Assumptions and Dependencies

- Every user of this system should have internet connection.
- System should be connected on LAN to use in an Organization to share frequent information.
- The speed of work depends upon speed of internet connection. For better performance LAN network topology is required.
- Every user is familiar to internet and computers.

4.2.10 Time Constraints

The final deployment of this website will not take place until all functional features has tested. This is estimated to be done by 29 May 2017.

4.3 REQUIREMENTS NOTES AND LEGENDS

Priority of Requirement:

C	Conditionally required
R	Required
O	Optional
SM	System Maintained
EF	External Feature
PM	Project Manager
PL	Project Leader
TL	Team Leader

4.4 REQUIREMENTS

4.4.1 Functional Requirements

S.No.	Req.No	Priority	Requirement
1	R1	R	User interface should be web based.
2	R2	R	Website should be user friendly and secure.
3	R3	R	Project sub module in admin section
4	R4	R	Component sub module in admin section
5	R5	R	Interface for functionality in component
6	R6	R	Interface for requirement management
7	R7	R	User management module
8	R8	R	Project team requirement
9	R9	R	Project planned setup version requirement
10	R10	R	Project weekly plan and schedule
11	R11	R	Some searching technique like filter, template etc.
12	R12	R	Facility to export schedule to excel
13	R13	R	Facility to import requirement from excel
14	R14	R	Unique id for each product
15	R15	R	Unique auto generated task id for product
16	R16	R	Schedule reports in form of excel sheet
17	R17	R	Task status report in form of pdf.
18	R18	R	Project setup management
19	R19	R	Users feedback management
20	R20	O	Persistent data storage in cookie
21	R21	R	Provision to display requirement, setup plan, schedule and other records.

22	R22	R	Remember password in login process
23	R23	R	Responsive Graphical User Interface

4.4.2 Security Requirements

S.No.	Req.No	Priority	Requirement
1	R1	R	There should be log maintained for login and logout of administrator.
2	R2	R	All the important information of database must be secure and protected from the malicious user
3	R3	R	All users of site must have separate login and authentication, and would be provided different feature in application based on their privilege level.
4	R4	R	Hashed password is stored in database using MD5 algorithm so that even database administration is not able to get password.

4.4.3 Performance Requirements

S.No.	Req.No	Priority	Requirement
1	R1	R	The system should be easy to learn and understand so that new user can also use the system effectively, without any difficulty.
2	R2	R	The system should meet user expectations.
3	R3	R	The response time of all the operations should be low. This can be made possible by careful programming.
4	R4	R	Response to user errors and the undesired situations should be taken care of to ensure that the system operates without halting.
5	R5	R	The system should be able to avoid or tackle catastrophic behavior.

6	R6	R	The system should recover from undesired events without human intervention.
---	----	---	---

4.5 DATA MANAGEMENT

Data should be stored in such a way that it is properly and quickly accessed. Separate folder should be maintained for each and every reports and excel sheets which is uploaded and downloaded by the client side. The name of report files are such that they can reside in same folder even client upload the same name file. For each report and excel files there must be an entry in database so old reports and uploaded can be traced out. All the data should be stored and organized in relational database management system.

4.6 DEPLOYMENT REQUIREMENT

4.6.1 Software Requirement

Client on Internet: Web Browser, Operating System (any)

Web Server: Apache Tomcat server

Application Language: Enterprise Java/jdk-1.7

Database: MySQL/Maria DB

IDE: Net beans

4.6.2 Hardware Requirement

Client Side

Browser: Google Chrome/Mozilla Firefox

Processor: Core i5 1.4GHz

RAM: 1GB

Disk Space: 150GB

Server Side

Browser: Google Chrome/Mozilla Firefox

Database: MySQL Server 2005

Processor: Core i5 1.4GHz

RAM: 2GB

Disk Space: 250GB

CHAPTER 5

System Design

SYSTEM DESIGN

5.1 SYSTEM DESIGN

Once the software requirements have been analyzed and specified, software design is the first of the three technical activities - design, code generation, and testing- that are required to build and verify the software. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. The designer's goal is to produce a model or representation of an entity that will later be built. The design step produces a data design, an architectural design, an interface design, and a procedural design.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software-engineering steps that follow. Without a strong design we risk building an unstable system -one that will fall when small changes are made; one that will be difficult to test; one whose quality can not be assessed until the last stage.

The data design transform the information domain model created during analysis into data structures that will be required to implement the software. The objects, relationships and detailed data content provided the basis for the data design activity.

The architectural design defines the relationship among major structural elements of the program. The interface design describes how the software communicates within itself, to systems that inter-operate with it and with humans who use it. An interface implies a flow of information.

The procedural design transforms structural elements of the program architecture into a procedural description of software components.

During design, we make decisions that will ultimately affect the success of software construction. The importance of design can be stated with a single word "quality".

5.2 DESIGN MODEL:

This model, illustrated in figure below, derives its name from the way in which the so system is built. More specifically, the model is designed, implemented and tested as a series of incremental builds until the product is finished. A build consists of pieces of code from various modules that interact together to provide a specific function.

At each stage of the IM – Incremental Model, a new build is coded and then integrated into the structure, which is tested as a whole. Note that the product is only defined as finished when it satisfies all of its requirements.

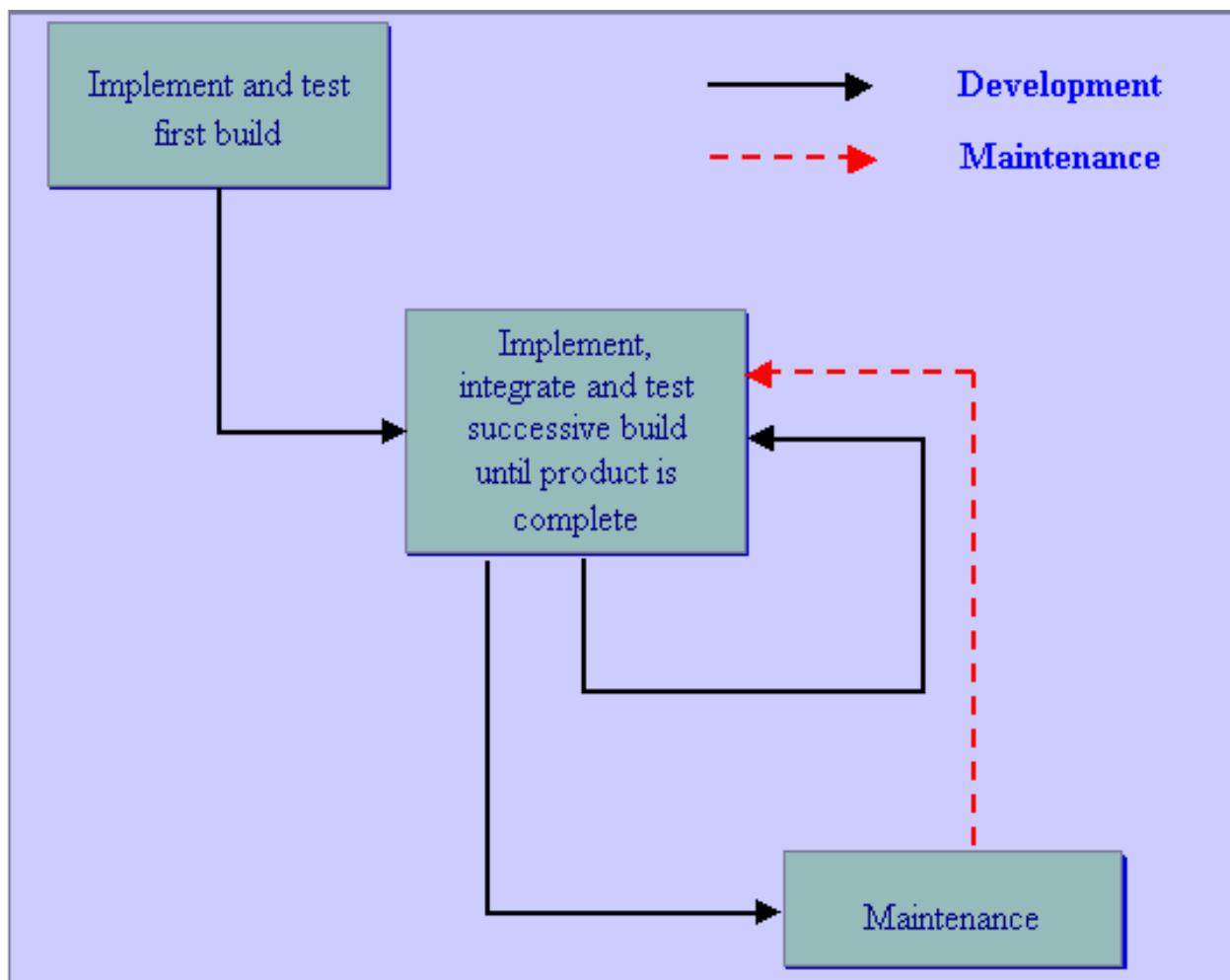


Figure 5.1.1: Increment Model

This model combines the elements of the waterfall model with the iterative philosophy of prototyping. However, unlike prototyping the IM focuses on the delivery of an operational product at the end of each increment.

The first increment is usually the core product which addresses the basic requirements of the system. This may be either be used by the client or subjected to detailed review to develop a plan for the next increment. This plan addresses the modification of the core product to better meet the needs of the customer, and the delivery of additional functionality.

More specifically, at each stage:

- The client assigns a value to each build not yet implemented
- The developer estimates cost of developing each build
- The resulting value-to-cost ratio is the criterion used for selecting which build is delivered next

Essentially the build with the highest value-to-cost ratio is the one that provides the client with the most functionality (value) for the least cost. Using this method the client has a usable product at all of the development stages.

Advantages

- Delivers an operational quality product at each stage, but one that satisfies only a subset of the client's requirements.
- A relative small number of programmers/developers may be used.
- There is a working system at all times.
- Clients can see the system and provide feedback.

- Most importantly, it breaks down the problem into sub-problems, dealing with reduced complexity, and reduced the ripple effect of changes by reducing the scope to only a part of the problem at a time.
- Distributes feedback throughout the whole development cycle, leading to more stable artifacts.

Disadvantages

- Each additional build has somehow to be incorporated into the existing structure without degrading the quality of what has been built to date.
- Addition of succeeding builds must be easy and straightforward.
- The more the succeeding builds are the source of unexpected problems, the more the existing structure has to be reorganized, leading to inefficiency and degrading internal quality and degrading maintainability.
- The incremental models can easily degenerate into the build and fix approach.
- Design errors become part of the system and are hard to remove.
- Clients see possibilities and want to change requirements.

5.3 Input Design

The design part of the project mainly deals with putting the data in right perspective. Data and its proper manipulation form the basis of this project. For the purpose of achieving a high level of control over data, it is imperative that we are able to put related data in the right context so that the retrieval of the data becomes not just convenient but also effective in terms of results that it can produce.

Our project concerns itself with various entities, both at the giving end and the receiving end. It is always important to cater the needs and requirements of all entities involved in data transfer, so that the specific purpose of producing data for the right party is achieved through a process of

simple querying. System design involves first logical design and then physical design of the system. When system analysts formulate a logical design they write the detailed specifications for the new system. They describe its feature, the outputs, the inputs, files and databases and procedures. The statements of these features are termed as the Design specifications.

5.4 Output Design

Computer output is the most important and direct source of information to the administrator. Efficient, intelligible output design should improve the systems relationship with the appraisal. A major form output, reports, is a hard copy from printer. When designing output, system analyst must accomplish the following:

- Determine what information should be present
- Decide whether to display, print the information and select the output medium
- Arrange the presentation of information in an acceptable format
- Decide how to distribute the outputs

5.5 DATABASE DESIGN

Data base is designed to manage large bodies of information. The management of data involves both the definitions of structures for the storage of information. In addition the data base system must provide for the safety of the information stored, despite system crashes or due to attempts at unauthorized access. For developing an efficient database we have to fulfill certain conditions such as controlled redundancy.

- a) Defining the data
- b) Inputting the data
- c) Locating the data
- d) Accessing the data
- e) Communicating the data
- f) Revising the data

5.5.1 Objectives of Data base design:

In this data base design several objectives are designed such as:

- a) Ease of use
- b) Control of data integrity
- c) Control of redundancy
- d) Control of security
- e) Data independence (logical & physical)
- f) Data storage protection
- g) System performance
- h) System functions
- i) System compatibility

For achieving the abovementioned criteria's we have to make use of various features that are available with the RDBMS by enforcing integrity constraints, we can ensure data integrity and reduce data inconsistency to a great extent.

5.5.2 Relational Schema Design

5.6 ARCHITECTURAL DESIGN

The primary objective of architectural-design is to develop a modular program structure and represent the control relationships between modules. In addition architectural design melds program structure and the data structure, defining an interface that enables the data to flow throughout the program.

5.7 INTERFACE DESIGN

Interface design focuses on the three areas of concern:

1. The design of interface between software modules.
2. The design of interfaces between the software and other non-human producers and consumers of information.
3. The design of interface between a human and the computer.

The interface is very helpful, friendly and easy, with respect to the error warning and the message. On each action the system responds, with warning and the error messages, where necessary.

5.8 PROCEDURAL DESIGN

Procedural design occurs after database, architectural, and interface designs have been established. The procedural design specifies the algorithmic detail of each of the function (module) in PDL (Process Design Language).

5.9 ANALYSIS AND DEVELOPMENT OF ACTUAL SOLUTION

The analysis phase is the detailed understanding of all important facts of the business area under investigation. The relationships of the various system components among themselves and with environment are studied and understood. This requires data collection from a variety of sources. For this, questionnaires, forms, interviews, study of existing documents, records etc. are used. The analysis must try to answer the following set of questions:

- What is being done in the organization?
- How it is being done?
- What is the volume of transactions?
- How frequently do the transactions occur?
- What are the problems that may arise?
- If a problem arises, how will it be solved?
- What could cause such a problem?

To answer all these questions, the system analyst must consult a variety of persons. He has to understand about the whole details about the problem. He should also identify the reasons for the problems that have occurred and the preventive measures to avoid them. The analyst must have a detailed study of the manuals and reports about the organization.

Further he should have a direct observation of the activities in the organization and collect a sample of the forms and documents to understand the whole system.

5.10 DESCRIPTION OF APPLICATION USER INTERFACE

We divided the interface of proposed system in two categories:

1. Interface for admin
2. Interface for user

5.10.1 Interface for admin

Product Interface

Purpose: If admin can want to add, update or delete the project, he/she can able to perform the task.

Component Interface

Purpose: If admin can want to add, update or delete the component in the project, he/she can able to perform the task.

Functionality Interface

Purpose: If admin can want to add, update or delete the functionality in the component of project, he/she can able to perform the task.

View Requirement Interface

Purpose: Admin can view the requirement list of any project and export as a excel sheet of all the requirement whose task status is maintained.

Setup Management Interface

Purpose: Admin can view the planned setup versions of project and can add, update or delete the particular versions and also view the graphical view of setup version task status.

User Management Interface

Purpose: Admin can manage all the user account and user (employee) working in an organization through this interface. He/ She can able to add user details, modify user details and delete the particular user record.

Team Management Interface

Purpose: Admin can manage all the project teams. He/She can able to add any user to a project team and if necessary make changes and also able to remove the user from the team.

Schedule Management Interface

Purpose: Admin can view all the project development weekly schedule through this interface.

Report Generation Interface

Purpose: Admin can generate the task status report through this interface.

Logout

Purpose: If the admin wants to end his session and sign out of the website then he can use the logout option.

5.9.2 Interface for User

Product Interface

Purpose: User can view his project name and details through this interface.
(Allowed to all user)

Requirement Interface

Purpose: User can add requirement details of functionality of project through this module and also able to create new functionality in module at add requirement time.
(It is only for PM, PL, TL user).

Search Requirement Interface:

Purpose: User can search requirement details of project through this module and also able to perform multiple searching.
(Allowed to all user).

Setup Management Interface

Purpose: User can add setup versions in project and can also able to modify the versions details. (Addition is allowed to PM, PL, TL user and view is allowed to all user)

Schedule Management Interface

Purpose: User can create project weekly schedule from an un-dispatched setup versions for a particular week through this interface. (Creation, modification is allowed to PM, PL, TL user and view is allowed to all user)

Report Generation Interface

Purpose: User can generate the task status report and schedule excel sheet through this interface.

Logout

Purpose: If the user wants to end his session and sign out of the website then he can use the logout option.

5.11 DETAILED DESIGN

5.11.1 Unified Modeling Language

UML is the international standard notation for object-oriented analysis and design. The Object Management Group defines it. The heart of object-oriented problem solving is the construction of a model. The model abstracts the essential details of the underlying problem from its usually complicated real world. Several modeling tools are wrapped under the heading of the **UML**, which stands for Unified Modeling Language.

UML combines techniques from data modelling (entity relationship diagrams), business modelling (work flows), object modelling, and component modelling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modelling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modelling language. UML aims to be a standard modelling language which can model concurrent and distributed systems. UML is a de facto industry standard, and is evolving under the auspices of the Object Management Group (OMG). OMG initially called for information on object-oriented methodologies that might create a rigorous software modelling language. Many industry leaders have responded in earnest to help create the UML standard.

An Overview of UML:

The UML is a language for

- Visualizing

- Specifying
- Constructing
- Documenting

The vocabulary of UML encompasses three kinds of building blocks:

- Things
- Relationships
- Diagrams

Things are the abstractions that are first-class citizens in a model.

Relationships tie these things together.

Diagrams group the interesting collection of things.

Relationships in the UML:

There are four kinds of relationships in the UML:

1. Dependency
2. Association
3. Generalization
4. Realization

1. Dependency:

This is relationship between two classes whenever one class is completely dependent on the other class. Graphically the dashed line represents it with arrow pointing to the class that it is being depended on.

2. Association:

It is a relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other in order to perform its work. In a diagram, an association is a link connecting two classes. Graphically it is represented by line.

3. Generalization:

An inheritance is a link indicating one class is a super class of the other. A generalization has a triangle pointing to the super class. Graphically it is represented by line with a triangle at end.

3. Realization:

It is relationship between two entities , where one entity guarantee that it will perform some task on behalf of other entity Graphically it is represented by dash line/broken lines with the hollow arrow pointing to the entity that specify the contract .

Diagrams in UML:

Diagrams play a very important role in the UML. There are nine kinds of modeling diagrams as follows:

1. Use Case Diagram
2. Class Diagram
3. Object Diagram
4. Sequence Diagram
5. Collaboration Diagram
6. State Chart Diagram
7. Activity Diagram
8. Component Diagram
9. Deployment Diagram

USE CASE DIAGRAM

A **use case diagram** in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

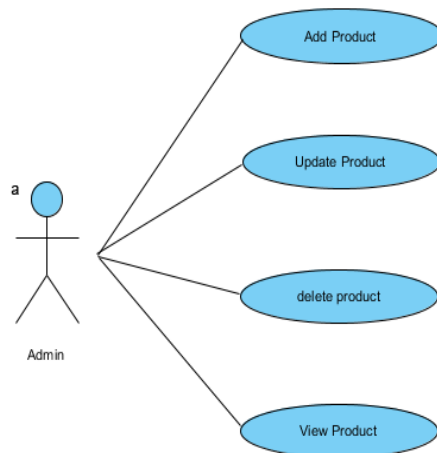
ELEMENTS OF A USE CASE DIAGRAM:

- **Actors**: An external entity (person or machine) that interacts with or uses the system.
- **Sequence of events description**: This describes a high level process of what an actor will do with a system. An actor may perform an event to start the system.

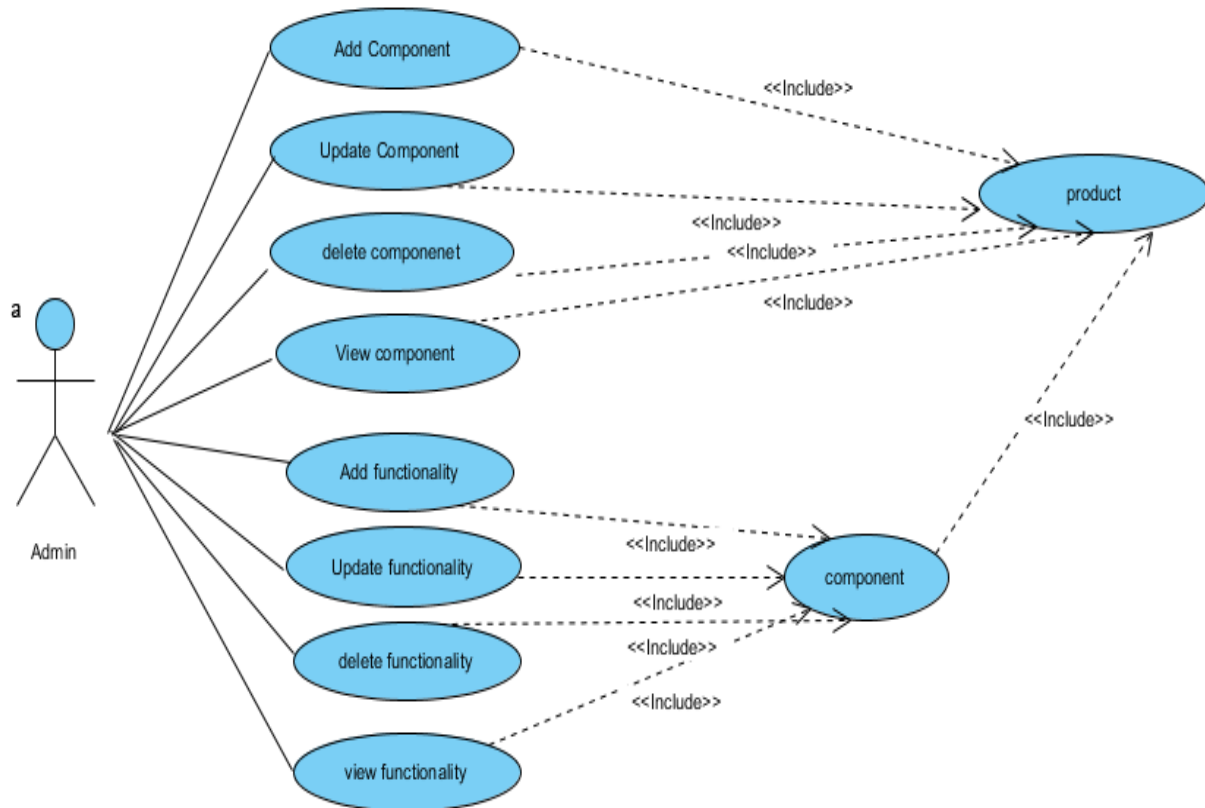
Use Case Diagram for Admin



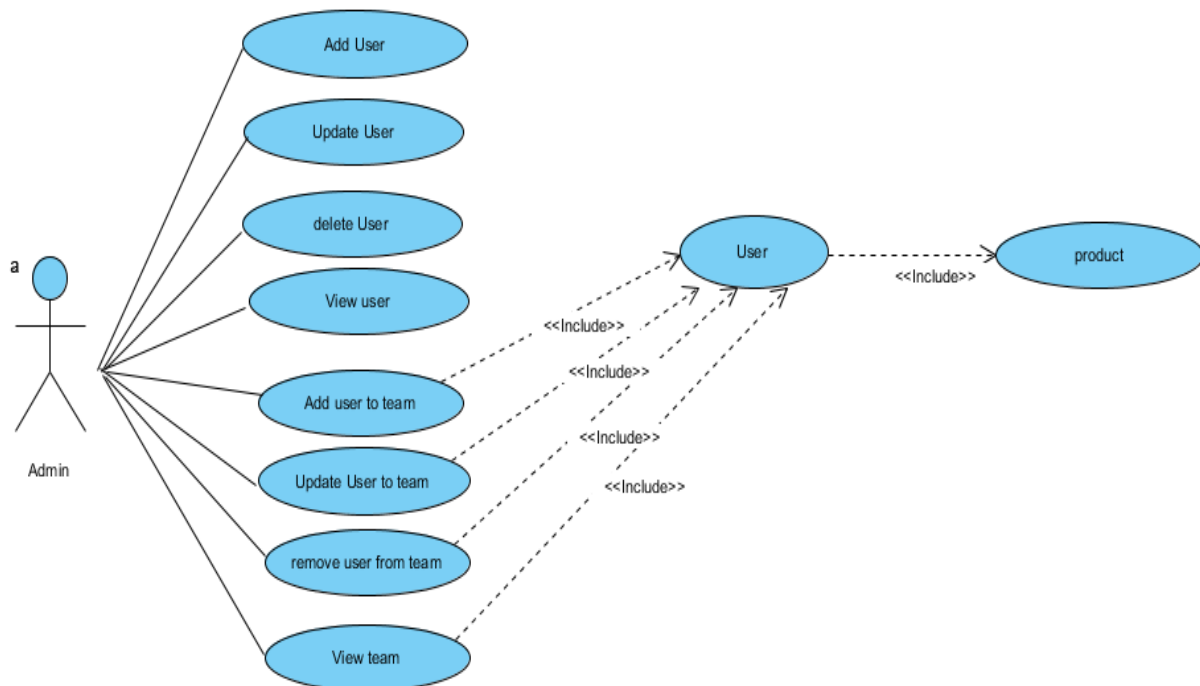
Use case diagram-Product management



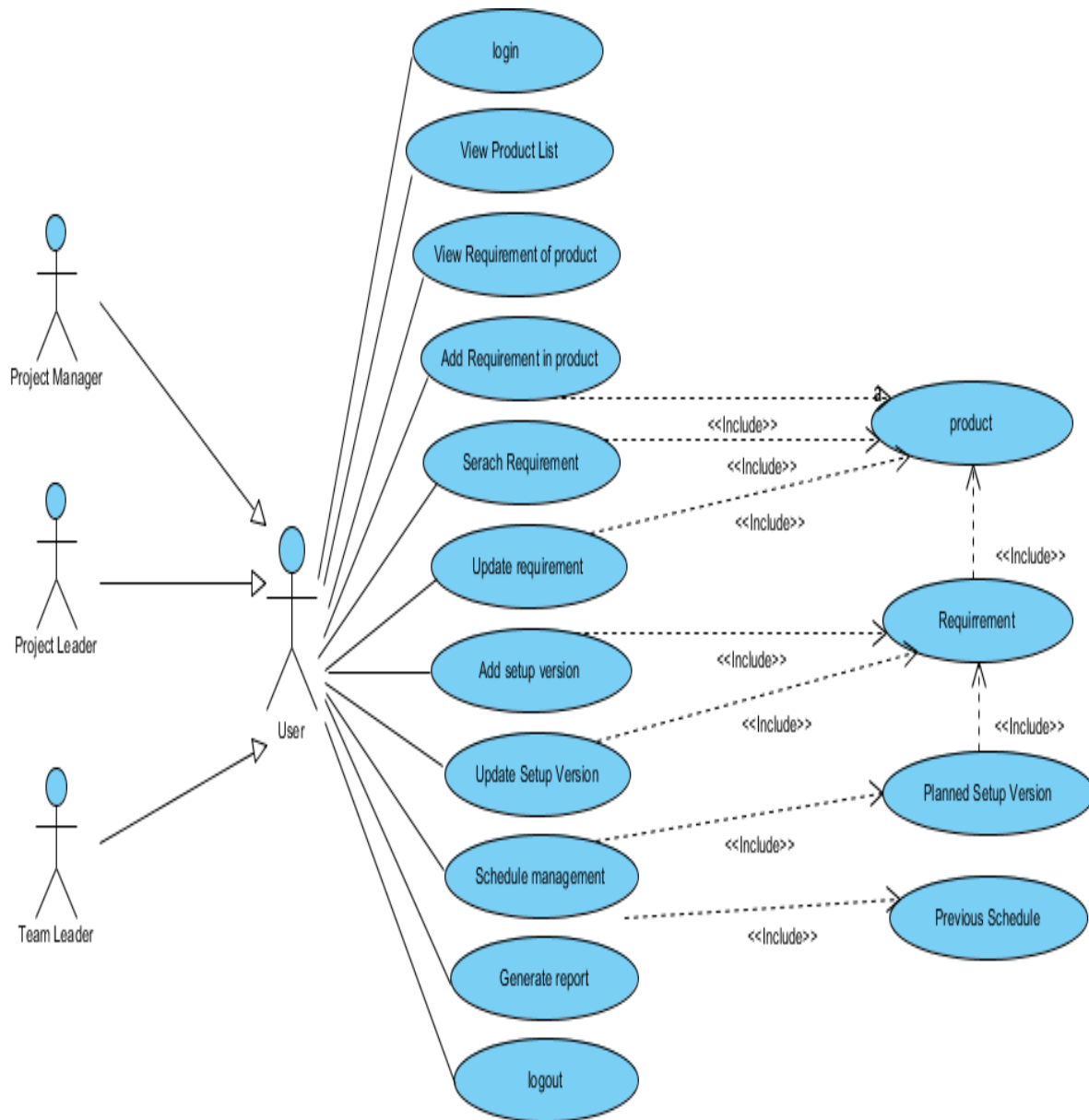
Use case diagram-Component and Functionality management



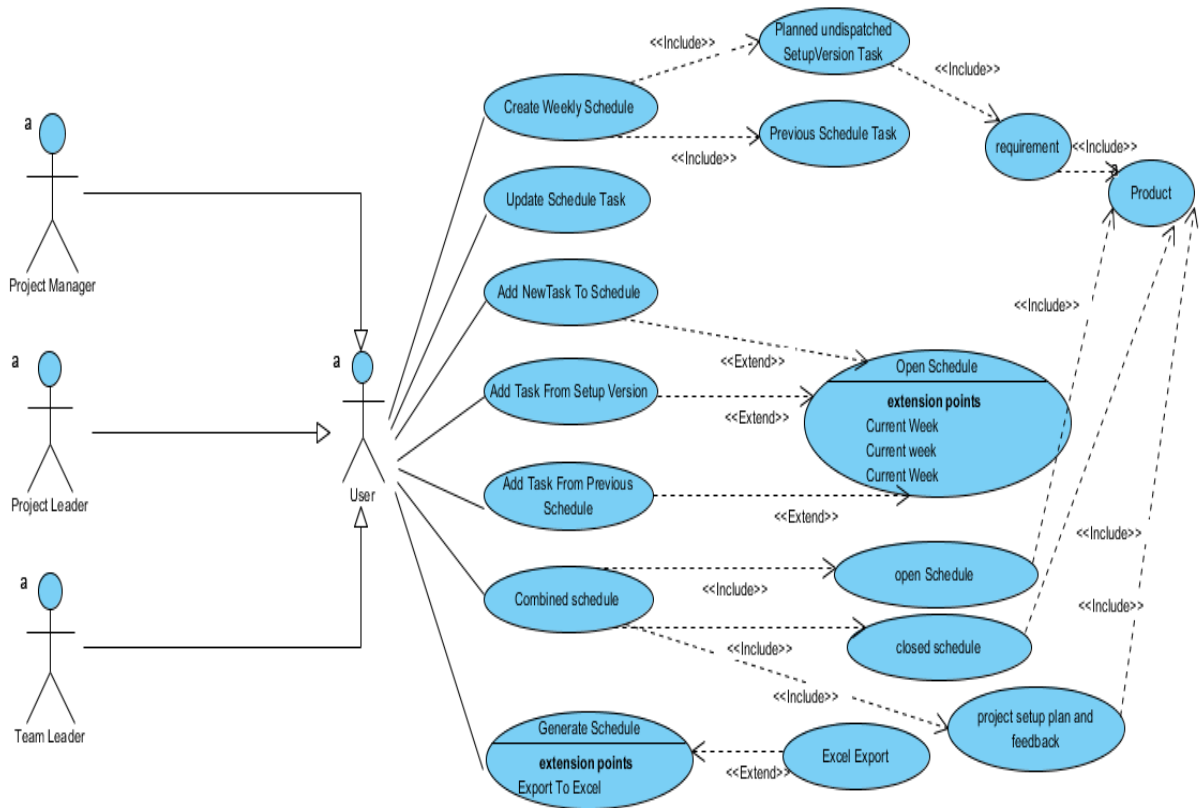
Use case diagram-User management and Team management



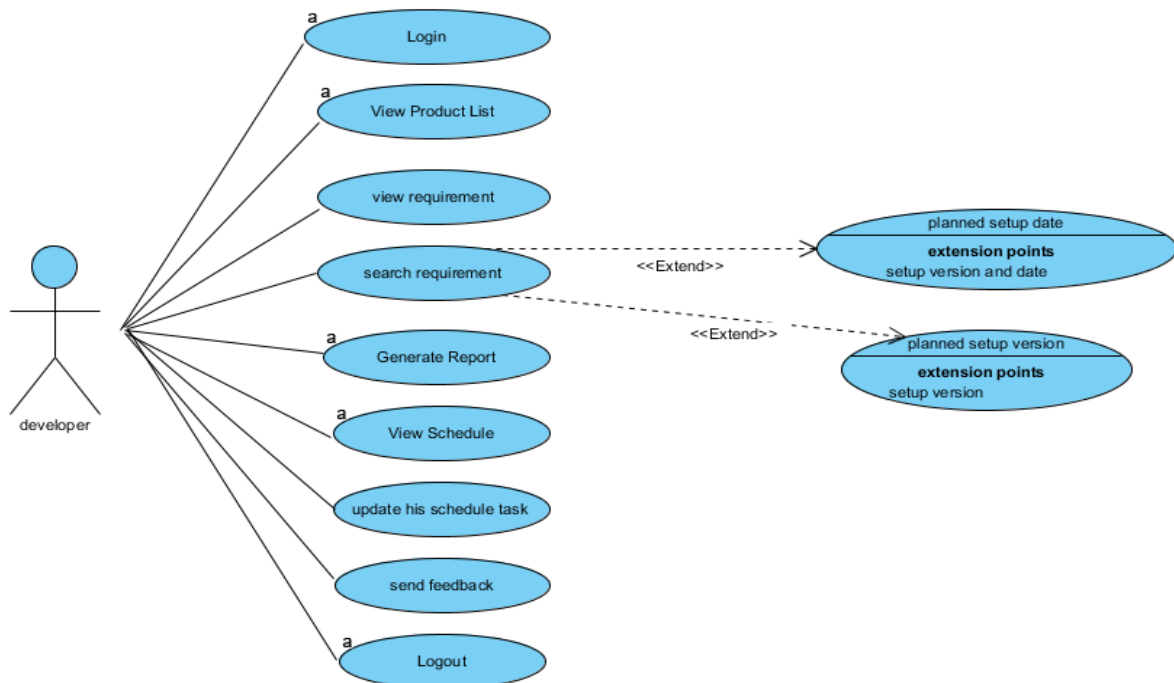
Use Case Diagram for Users of the application



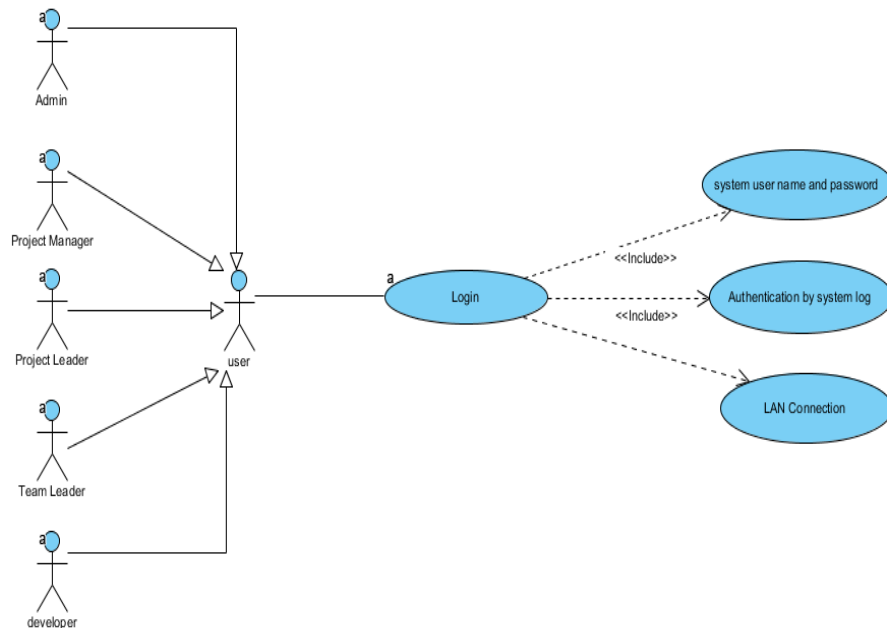
Use Case Diagram-Schedule Management



Use Case Diagram for developer



Use Case Diagram-Login



CLASS DIAGRAM

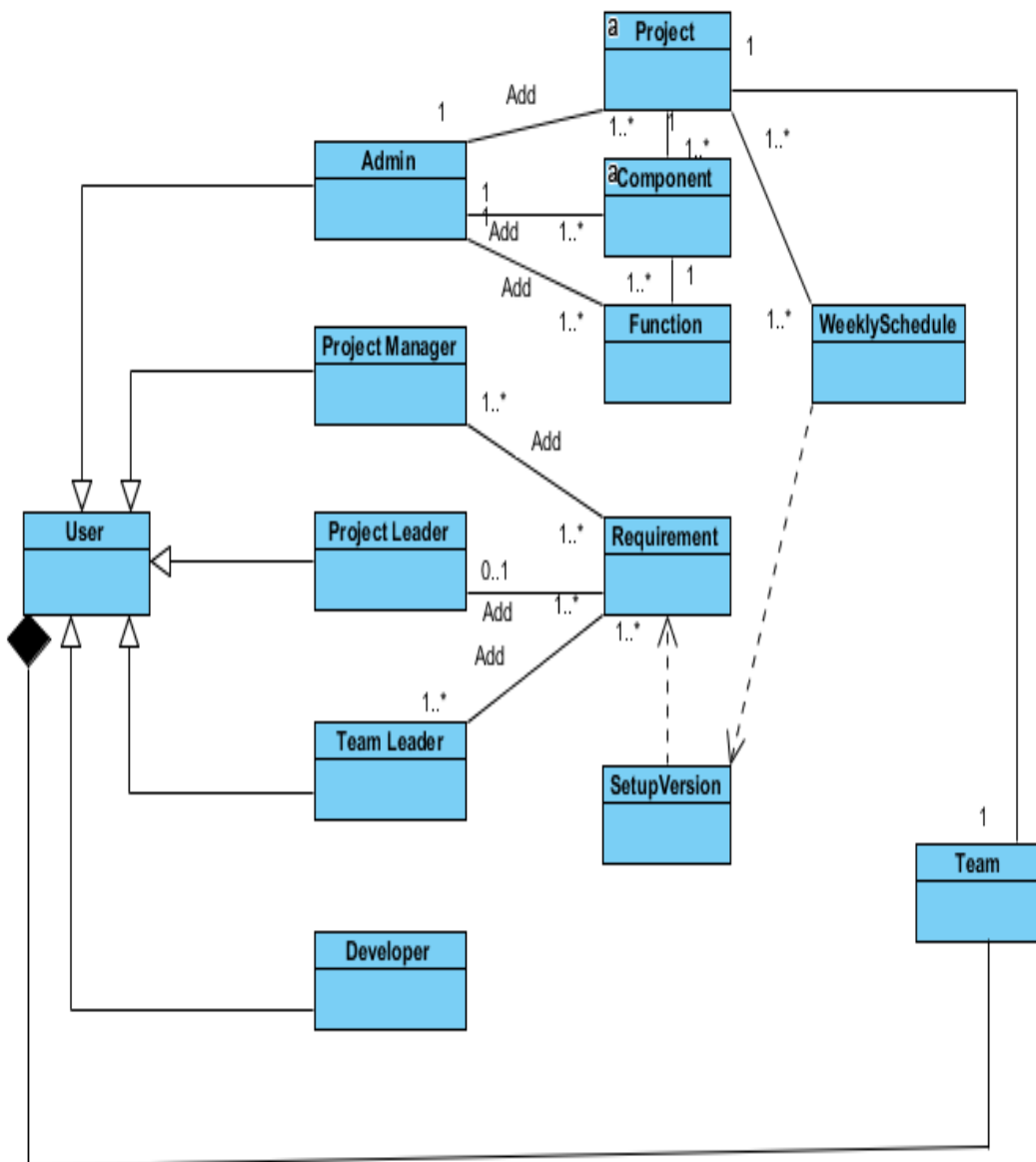
A **class diagram** in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

The class diagram is the main building block in object oriented modelling. They are being used both for general conceptual modelling of the systematic of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts:

A class with three sections:

- The upper part holds the name of the class
- The middle part contains the attributes of the class, and
- The bottom part gives the methods or operations the class can take or undertake

In the conceptual design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the statically relations between those objects. With detailed modeling, the classes of the conceptual design are often split in a number of subclasses.



SEQUENCE DIAGRAM

A **Sequence diagram** is an [interaction diagram](#) that shows how processes operate with one another and what is their order. It is a construct of a [Message Sequence Chart](#). A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner

5.11.2 Data Flow Diagram

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

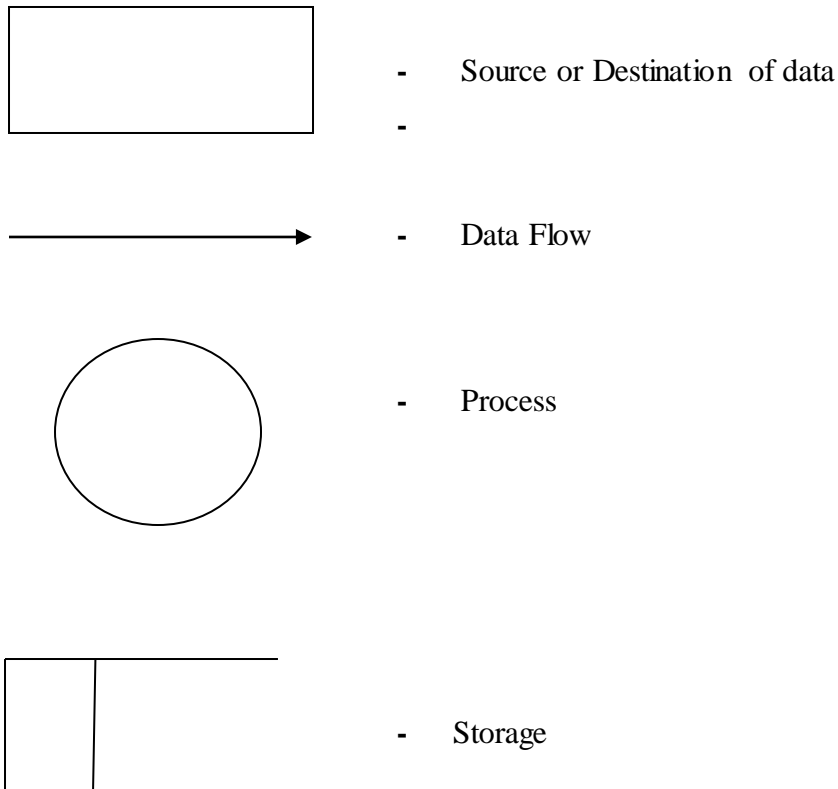
The Data flow Diagram shows the flow of data. It is generally made of symbols given below:-

- A square shows the Entity.
- A Circle shows the Process
- An open Ended Rectangle shows the data store.
- An arrow shows the data flow.

The DFD can be up to several levels. The 0 level DFD states the flow of data in the system as seen from the outward in each module.

The first level DFD show more detail, about the single process of 0 levels DFD

Data Flow Diagram Symbols:-



Steps to Construct Data Flow Diagrams

Four steps are commonly used to construct a DFD

- Process should be named and numbered for easy reference. Each name should be representative of the process.
- The destination of flow is from top to bottom and from left to right.
- When a process is exploded in to lower level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram

- Arrows should not cross each other.
- Squares, circles and files must bear names.
- Decomposed data flow squares and circles can have same names.

5.12 DATABASE DESIGN

A database is a set of data, organized for easy access. The database is an actual data; it is the database that you will be accessing when you need to retrieve data.

5.12.1 Data Dictionary

The data dictionary is a set of tables used to maintain information about the database. The data dictionary contains information about tables, indexes, clusters and so on.

5.12.2 DBA (Database Administrator)

The DBA is the person responsible for the operation, configuration and performance of the database. The DBA is charged with keeping the database operating smoothly, ensuring that backups are done on regular basis (and that backups work), and installing new software. Other responsibilities might include planning for future expansion and disk space needs, creating databases and table spaces, adding users and maintaining security, and monitoring the database and retuning it as necessary. Large installations might have teams of DBA's to keep the system running smoothly; alternatively, the task might be segmented among the DBA.

5.12.3 End-user

End users are those who access the database from a terminal and may employ query language provided as an integral part of the system. The user may invoke a user written program that accepts a command from the terminal and in this issue requests to the DBMS on end User's behalf.

(a) Native user: Users who need not be aware of the presence of the database system or any other system supporting their usage, are considered naïve users. These users work through a menu oriented application program.

(b) On line user: These users may communicate with the database directly via an online terminal or directly via a user interface and application program. They use DML to manipulate database directly.

5.12.4 STRUCTURE OF THE TABLES USE

Project: This table contains the project details.

Field	Data Type	Length	Constraint	Description
projectid	BIGINT	10	Primary Key	Auto-increment, not null
Projectname	VARCHAR	100		Not null
Description	VARCHAR	500		Null
Nick	VARCHAR	100		Not null

Component: This table contains the project component details.

Field	Data Type	Length	Constraint	Description
Id	BIGINT	10	Primary Key	Auto-increment, not null
Componentname	VARCHAR	100		Not null
Description	VARCHAR	500		Null
Project code	BIGINT	10	Foreign key	Null

Functionality: This table contains the project component functionality details.

Field	Data Type	Length	Constraint	Description
Id	BIGINT	10	Primary key	Auto-increment, not null
cid	BIGINT	10	Foreign Key	Null
Name	VARCHAR	100		Not Null
Details	VARCHAR	500		Null
Pid	BIGINT	10	Foreign Key	Null

User: This table contains the user details.

Field	Data Type	Length	Constraint	Description
Id	BIGINT	10	Primary Key	Auto-increment, not null
Username	VARCHAR	100		Not Null
User id	VARCHAR	100		Not null
Password	VARCHAR	50		Not null
First name	VARCHAR	50		Null
Last name	VARCHAR	50		Null
Email	VARCHAR	100		Not null
User type	VARCHAR	50		Not null
User designation	VARCHAR	50		Not null

Team: This table contains the team details.

Field	Data Type	Length	Constraint	Description
Uid	BIGINT	10	Foreign Key	Null
Pid	BIGINT	10	Foreign Key	Null

Requirement: This table contains the requirement details of project.

Field	Data Type	Length	Constraint	Description
Id	BIGINT	10	Primary Key	Auto-increment, not null
Task ID	VARCHAR	50	Unique Key	Not null
Task	VARCHAR	500		Not null
Planned_setup_version	VARCHAR	100	Foreign Key	Not Null
Planned_setup_date	DATE	50		Not Null
Resource	VARCHAR	100		Null

Start date	DATE	50		Null
Close Date	DATE	50		Null
Status	VARCHAR	50		Null
Reported by	VARCHAR	100		Null
Pid	BIGINT	10	Foreign Key	Null

Setup version: This table contains the setup version details of project.

Field	Data Type	Length	Constraint	Description
Version_id	BIGINT	10	Unique key	Auto-increment, not null
Pid	BIGINT	10	Primary key, unique key, foreign key	Not null
Planned_setup_version	VARCHAR	100	Primary key, unique key	Not null
DISPATCHED	Boolean	1		Null
Planned_setup_date	DATE	50		Not null

Weekly Schedule: This table contains the schedule details.

Field	Data Type	Length	Constraint	Description
Id	BIGINT	10	Primary Key	Not null
Pid	BIGINT	10	Foreign Key	Not null
Weekly-schedule	VARCHAR	100		Not null
Start date	DATE	50		Not null
End date	DATE	50		Not null
Status	VARCHAR	50		Not null

Weekly Schedule Task: This table contains the schedule task details.

Field	Data Type	Length	Constraint	Description
id	BIGINT	10	Primary Key	Not Null
Task Id	VARCHAR	50	Foreign Key	Not null
Task	VARCHAR	500		Not Null
Planned start date	DATE	50		Not null
Planned end date	DATE	50		Not null
Actual Start date	DATE	50		Not Null
Actual End date	DATE	50		Not null
Comments	VARCHAR	500		Null
Status	VARCHAR	10		Null
Wsid	BIGINT	10	Foreign Key	Null
Uid	BIGINT	10	Foreign Key	Null

Schedule template: This table contains the schedule template details.

Field	Data Type	Length	Constraint	Description
Userid	BIGINT	10	Primary Key	Not null
Template name	VARCHAR	50	Primary Key	Not null
Project template	VARCHAR	500	Primary Key	Not null

Feedback: This table contains the feedback details.

Field	Data Type	Length	Constraint	Description
Id	BIGINT	10	Primary Key	Auto-increment, not null
Product	VARCHAR	50	Key	Not null
Requirement	VARCHAR	500	Key	Not null
Email	VARCHAR	50	Key	Not null
Subject	VARCHAR	100		Null
Message	VARCHAR	500	Key	Null
Date	DATE			Null
Time	DATETIME			Null

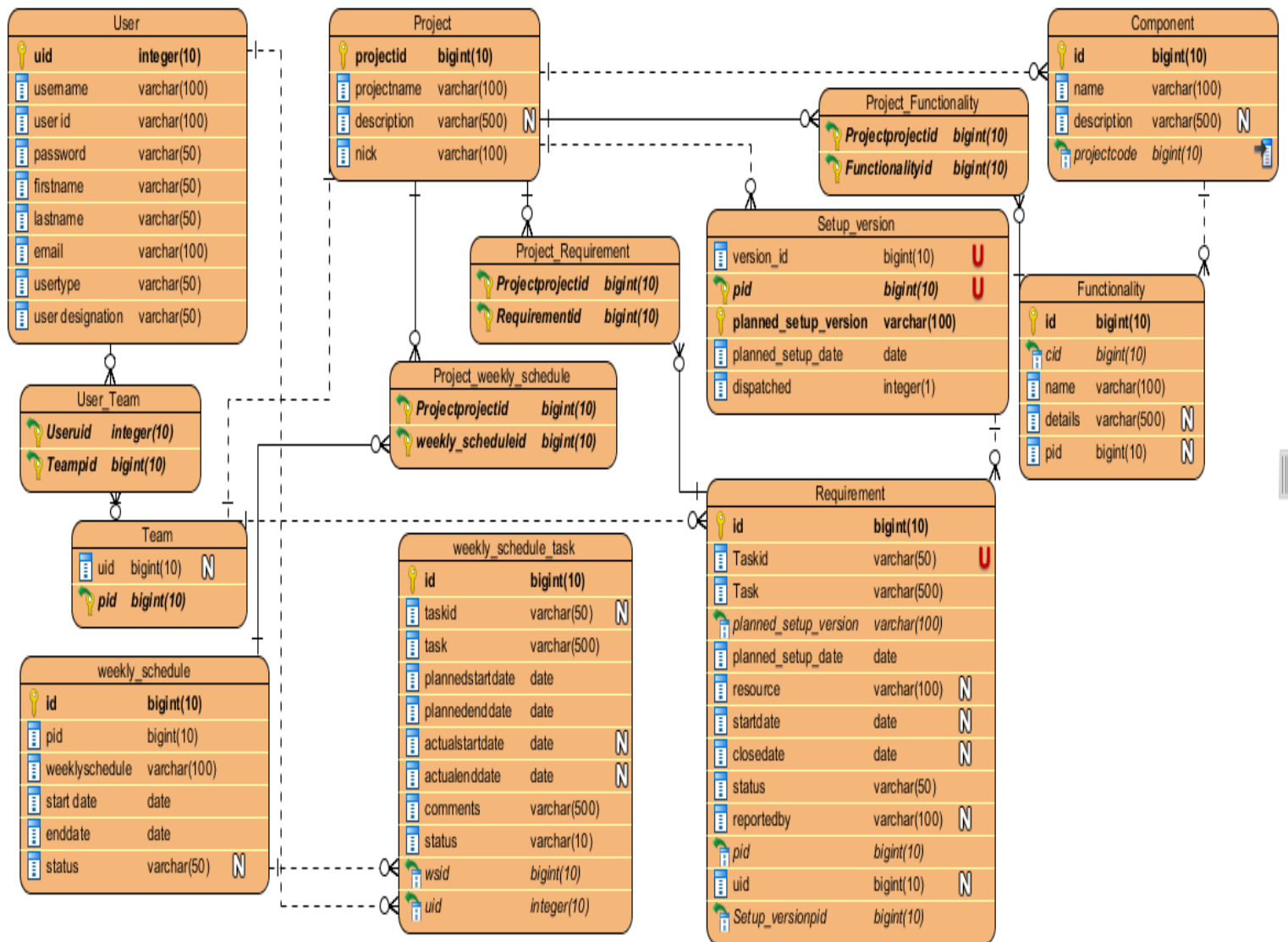
5.12.5 ENTITY-RELATIONSHIP DIAGRAM

Entity: The basic object that the ER model represents is an entity, which is a “thing” in the real world with an independent existence. An entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

Entity relationship diagram depicts the various relationships among entities, considering each object as entity. Entity is represented as rectangle shape and relationship is represented as diamond shape. It depicts the relationship between the data objects. ER diagram is a notation that is used to conduct the data modelling activity.

Relationship: A relationship is named connection or associated between entities or used to relate two or more entities with some common attributes or meaningful interaction between the objects.

Attributes: Each entity has attributes – the particular properties that describe it. A particular entity will have a value for each of its attributes.



5.13 CHOICE OF LANGUAGE

5.13.1 Front End design (Presentation Layer)

HTML: Without HTML, the World Wide Web wouldn't exist. HTML allows the individual elements on the Web to be brought together and assented as a collection. Text, images, multimedia, and other files can be packed together using HTML. This section explains the basic principles behind the interaction between HTML and the World Wide Web.

CSS: Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

- **Pages load faster** – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

JAVA SCRIPT: JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JQuery: jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto – **Write less, do more.**

jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery –

- **DOM manipulation** – The jQuery made it easy to select DOM elements, traverse them and modifying their content by using cross-browser open source selector engine called **Sizzle**.
- **Event handling** – The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- **AJAX Support** – The jQuery helps you a lot to develop a responsive and feature-rich site using AJAX technology.
- **Animations** – The jQuery comes with plenty of built-in animation effects which you can use in your websites.
- **Lightweight** – The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).

AJAX:

AJAX stands for Asynchronous **JavaScript** and **XML**. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

Conventional web applications transmit information to and from the server using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.

With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.

AJAX is a web browser technology independent of web server software.

A user can continue to use the application while the client program requests information from the server in the background.

5.13.2 Business Logic (Business Layer)

JSP (Java Server Pages)

JSP technology is object-oriented programming language and is based on Java language. **Java Server Pages (JSP)** is a server side Java technology that allows software developers to create dynamically generated web pages, with HTML, XML, or other document types, in response to a Web client request to a Java Web Application container (server). JSP pages are Web pages coded with an extended HTML that makes it possible to embed Java code in a Web page. Java Server Pages (JSP) is a technology that lets you mix regular, static HTML with dynamically generated HTML. JSP page can call custom Java Classes, called taglibs, using HTML like tags. The JSP

compiler translates JSP pages into Servlet. Server automatically compile JSP pages if the Servlet class file not present or is older than the JSP source file. You can also pre-compile JSP pages and package the Servlet class in the 'Web Archive' to avoiding compile the server. JSP pages may depend upon additional helper classes that must also be deployed with in the Web application.

The features of JSP (Java Server Pages) are as follows:

- An **Expression Language** (EL) which allows developers to create Velocity-style templates (among other things).
- It simplifies the process of development (It allows programmers to insert the Java code directly into the JSP file, making the development process easier).
- Reusability (JSP allows the component reuse by using JavaBeans and EJBs).
- Robust (JSP allows the robust platform for web development).
- Independency of Layers. There is a clear separation between presentation and implementation layers. The HTML on the web browser of the client is displayed as a presentation layer. The JSP on the server is displayed in the implementation layer.

Servlet

Servlet are module of Java code that run in a server application to answer client requests. The name "Servlets" come from the "Applets", which run on the client side whereas Servlet execute at the server side. In other words, the Servlet is the web front-end of a large Web application. Java servlets, a widely accepted technology for building dynamic content for web-based applications. Servlets are Java programming language objects that dynamically process requests and construct responses. The **Java Servlet API** allows a software developer to add dynamic content to a Web server using the Java platform

A Servlet is a Java class that runs in the Server machine on an HTTP request from client and creates an html as response to the client. Servlets are stored in a Servlet Container in the Web Server and are mapped to URLs, for which the servlet constructs response. The features of the Servlets are as follows:

- **Ease of development** (Ease of development is one of the key success mantras of any technology. This means that you can swiftly develop a servlet or a filter class by simply annotating the class with appropriate annotations like @Servlet or @Servlet Filter).
- **Plug ability and extensibility.**
 - The ability to receive data from a client without blocking even if the data is slow arriving (non-blocking input).
 - The ability to send data to a client without blocking, even if the client or network is slow (non-blocking output).
 - The ability to handle delayed response close; i.e., the response will be held open to allow additional data to be sent when asynchronous events occur.
 - The ability to notify blocking or non-blocking events.
- **Several web.xml convenience** (Servlet 2.5 introduces several small changes to the web.xml file to make it more convenient to use.
- **A Handful of removed restrictions** (Servlet 2.5 removed a few restrictions around error handling and session tracking.

5.13.3 Back End Design (Database Layer)

MySQL

Introduction: MySQL is the world's most popular open source database software. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

Features: MySQL offered MySQL 5.5 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server include the following features:

- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- True Varchar support
- Information schema
- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Embedded database library
- ACID compliance when using transaction capable storage engines
- Partitioned tables with pruning of partitions in optimizer
- Hot backup under certain conditions.

5.14 AIMS AND OBJECTIVES TO BE FULFILLED

Provides Right Security:

In the proposed system, we have kept User Login facility for the user. Because of this only authorized person can operate it and access to database. This is helpful to stop unauthorized access so that sensitive information can be kept secured

Time Effective:

This system is very fast for retrieving information from the database. It makes the information available on the click of the mouse. This saves the user's time and it stops it as being time consuming process.

Reduces Paper Work:

This system reduces the paper work that is the user has not to do entry in each and every register only one entry needed to be done by this computerized project.

Saves time:

This system also saves time of the user by scheduling the task for each and every project on the web rather than paper.

Provides Security:

This system provides security to the information of system as only valid operator or authorized person can access it.

Providing information:

This system will provide more and more information about the firm. That is the system will provide information regarding the product, its component, functionality, user, team, requirement, schedule, setup version etc.

Backup and Recovery:

The system also provide facility for back and recovery in the case of manual system if any information about project got miss there is no way to get the useful information again but in this computerized system there would be the chance to collect the useful information again.

CHAPTER 6

System Implementation

SYSTEM IMPLEMENTATION

6.1 INTRODUCTION

System implementation is the stage when the user has thoroughly tested the system and approves all the features provided by the system. The various tests are performed and the system is approved only after all the requirements are met and the user is satisfied.

The new system may be totally new; replacing an existing manual or automated system, or it may be a major modification to an existing system. In either case, proper implementation is essential to provide a reliable system to meet organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system (that is a design question), but improper will prevent it. Implementation is the process of having systems personnel check out and put new equipment into use, train users, install the new application and construct any files of data needed to use it. This phase is less creative than system design. Depending on the size of the organization that will be involved in using the application and the risk involved in its use, systems developers may choose to test the operation in only one area of the firm with only one or two persons. Sometimes, they will run both old and new system in parallel way to compare the results. In still other situations, system developers stop using the old system one day and start using the new one the next.

6.1.1 ASPECTS OF IMPLEMENTATION

The two aspects of implementation are:

- Training Personnel
- Conversion Procedures

Training

Even well designed and technically elegant systems can succeed or fail because of the way they are used. Therefore the quality of the training received by the personnel involved with the system in various ways helps or hinders, and may even prevent, the successful implementation of an information system.

Since, Human Resource Recruitment Process is web-based and user friendly, not much effort was required in training process.

Conversion:

Conversion is the process of changing from the old system to the new system. There are two methods of handling systems conversion:

- Parallel Run
- Immediate cut-off

Parallel Run

In this approach, the old system and the new system are used simultaneously for some period of time so that the performance of the new system can be monitored and compared with that of the old system. Also in case of failure of the new system, the user can fall back on the old system.

The risk of this approach is that the user may never want to shift to new system.

Immediate cut-off

In this method, the use of the old system ceases as soon as the new system is implemented and brought in to place. The old system becomes redundant from the day of implementation of the new system. There is the high risk involved in this approach if the new system is not tested rigorously. This is because of the fact that if the new system fails, then there will not be anything to fall back upon. The advantage of this approach is that both the systems need not be used simultaneously

6.2 HARDWARE REQUIREMENT:

It is recommended that the minimum configuration for this project is as below:

Client Side

Browser: Google Chrome/Mozilla Firefox

Processor: Core i5 1.4GHz

RAM: 1GB

Disk Space: 150GB

Server Side

Browser: Google Chrome/Mozilla Firefox

Database: MySQL Server 2005

Processor: Core i5 1.4GHz

RAM: 2GB

Disk Space: 250GB

6.3 SOFTWARE REQUIREMENT

Client on Internet: Web Browser, Operating System (any)

Web Server: Apache Tomcat server

Application Language: Enterprise Java/jdk-1.7

Database: MySQL/Maria DB

IDE: Net beans

CHAPTER 7

System Testing

SYSTEM TESTING

7.1 SOFTWARE TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software testing can also be stated as the process of validating and verifying that a software program/application/product:

1. meets the business and technical requirements that guided its design and development;
2. works as expected; and
3. Can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

7.1.1 OVERVIEW

Testing can never completely identify all the defects within software. Instead, it furnishes a criticism or comparison that compares the state and behavior of the product against oracles—principles or mechanisms by which someone might recognize a problem. These oracles may include (but are not limited to) specifications, contracts, comparable products, past versions of the same product, inferences about intended or expected purpose, user or customer expectations, relevant standards, applicable laws, or other criteria.

Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. Software testing is the process of attempting to make this assessment.

7.2 SOFTWARE QUALITY ASSURANCE (SQA)

Though controversial, software testing may be viewed as an important part of the software quality assurance (SQA) process. In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in the delivered software: the so-called *defect rate*.

What constitutes an "acceptable defect rate" depends on the nature of the software; a flight simulator video game would have much higher defect tolerance than software for an actual airplane. Although there are close links with SQA, testing departments often exist independently, and there may be no SQA function in some companies.

Software testing is a task intended to detect defects in software by contrasting a computer program's expected results with its actual results for a given set of inputs. By contrast, QA (quality assurance) is the implementation of policies and procedures intended to prevent defects from occurring in the first place.

7.3 TESTING LEVELS

Tests are frequently grouped by where they are added in the software development process, or by the level of specificity of the test.

7.3.1 Unit testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

Unit testing is also called *component testing*.

7.3.2 Integration testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

7.3.3 System testing

System testing tests a completely integrated system to verify that it meets its requirements.

7.3.4 Regression testing

Regression testing focuses on finding defects after a major code change has occurred. Specifically, it seeks to uncover software regressions, or old bugs that have come back. Such regressions occur whenever software functionality that was previously working correctly stops working as intended. Typically, regressions occur as an unintended consequence of program changes, when the newly developed part of the software collides with the previously existing code. Common methods of regression testing include re-running previously run tests and checking whether previously fixed faults have re-emerged. The depth of testing depends on the phase in the release process and the risk of the added features. They can either be complete, for changes added late in the release or deemed to be risky, to very shallow, consisting of positive tests on each feature, if the changes are early in the release or deemed to be of low risk.

7.3.5 Acceptance testing

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. before integration or regression.

2. Acceptance testing performed by the customer, often in their lab environment on their own hardware, is known as user acceptance testing (UAT). Acceptance testing may be performed as part of the hand-off process between any two phases of development.

7.3.6 Smoke Test

This is also called as sanity testing. This is mainly used to identify environmental related problems and is performed mostly by test manager. For any application it is always necessary to have the environment first checked for smooth running of the application. So in this testing process the application is run in the environment technically called as dry run and checked to find that the application could run without any problem or amend in between.

7.3.7 Alpha Testing

The above different testing process described takes place in different stages of development as per the requirement and needs. But a final testing is always made after a full finished product that is before it released to end users and this is called as alpha testing. The alpha testing involves both the white box testing and black box testing thus making alpha testing to be carried out in two phases.

7.3.8 Beta Testing

This process of testing is carried out to have more validity of the software developed. This takes place after the alpha testing. After the alpha phase also the generally the release is not made fully to all end users. The product is released to a set of people and feedback is got from them to ensure the validity of the product. So here normally the testing is being done by group of end users and therefore this beta testing phase covers black box testing or functionality testing only.

7.3.9 Manual Testing

Manual Testing will be done to ensure the correctness of various parts of the code using test cases generated by the tester.

7.4 TESTING METHODS

7.4.1 White Box Testing

Structural testing or White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc.

In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.

Advantages of White box testing are:

- i) As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
- ii) The other advantage of white box testing is that it helps in optimizing the code
- iii) It helps in removing the extra lines of code, which can bring in hidden defects.

Disadvantages of white box testing are:

- i) As knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, which increases the cost.
- ii) And it is nearly impossible to look into every bit of code to find out hidden errors, which may create problems, resulting in failure of the application.

7.4.2 Black Box Testing

Black Box Testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name "black box" suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the

work product / software application. **Black box testing is sometimes also called as "Opaque Testing", "Functional/Behavioral Testing" and "Closed Box Testing".**

In order to implement Black Box Testing Strategy, the tester is needed to be thorough with the requirement specifications of the system and as a user, should know, how the system should behave in response to the particular action.

Various testing types that fall under the Black Box Testing strategy are: functional testing, stress testing, recovery testing, volume testing, User Acceptance Testing (also known as UAT), system testing, Sanity or Smoke testing, load testing, Usability testing, Exploratory testing, ad-hoc testing, alpha testing, beta testing etc.

7.5 PASS/FAIL CRITERIA

The system should satisfy all the functional requirements in the system analysis. Each feature to be tested will be evaluated against its requirement as stated in the system analysis. The pass or fail of a test depends on whether the system meets with all the particular post conditions.

Test cases executed on the Online Book Store will pass if they meet the specific requirements as mentioned in the system design.

7.6 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

7.6.1 Suspension criteria

If the system contains one or more critical defects like the defects in the GUI editor which provides the editing features for one line diagrams and database locking, unlocking and sharing features which provides the environment for multiple users to work in parallel, the entire system should be suspended. The testing may also be suspended if the hardware and software components required are not available on time. The failed test cases should be recorded along with the description for failure.

7.6.2 Resumption requirements

When a new version of the system is transmitted to the test group after a suspension of testing has occurred, all previous tests will be rerun to ensure program changes have not inadvertently affected other portions of the program.

7.7 TEST DELIVERABLES

The following documents are the available test deliverables:-

- Test plan
- Test case specifications
- Test input and output data
- Test procedure specifications
- Test logs

7.8 TEST CASES AND RESULTS

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases - manual and automated. A manual test case is executed manually while an automated test case is executed using automation. In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger errors.

Manual testing has been done for the developed application. The test cases along with their execution status are as below.

7.8.1 USER LOGIN TEST CASE

Test Case	TC-01
Test Description	Leaves all fields empty when logging in
Test Data	Blank User Id and password fields
Expected Result	Message should be displayed "Please enter valid User Id and Password"
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-02
Test Description	Enter Invalid User Id
Test Data	Invalid User Id
Expected Result	Message should be displayed "User Id or Password is incorrect"
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-03
Test Description	Enter Invalid password
Test Data	Invalid password
Expected Result	Message should be displayed "User Id or Password is incorrect"
Actual Result	The message is displayed correctly
Status	Pass
Test Case	TC-04
Test Description	Enter Valid User Id and Password
Test Data	User Id: correct user id and Password: correct
Expected Result	Allows to proceed

Actual Result	Login Successful
Status	Pass

7.8.2 PROJECT MANAGEMENT TEST CASE

Test Case	TC-05
Test Description	Leaves all non-mandatory fields empty when adding a new project details
Test Data	Blank Project Name, project code, description and purpose fields
Expected Result	Message should be displayed "Please enter project name and project code"
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-06
Test Description	Enter invalid project name and project code
Test Data	Project name=123 and project code=123
Expected Result	Message should be displayed "Please enter project name and project code in valid format."
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-07
Test Description	Enter valid project details
Test Data	Project name="Document Management System", Project code="DMS", description="for managing document", Purpose=Blank
Expected Result	Message should be displayed "Project added successfully" and project details show in tabular form.

Actual Result	The message is displayed correctly and project details is visible in tabular form.
Status	Pass
Test Case	TC-08
Test Description	Click on edit link and update the project details.
Test Data	Project name="Document Management System", Project code="DMS", description="for managing document", Purpose="to reduce time to manage document in file"
Expected Result	After click on edit button, project details should be visible in editable format and after saving display message "Project details updated successfully."
Actual Result	Same as expected result
Status	Pass
Test Case	TC-09
Test Description	Click on delete link.
Test Data	
Expected Result	After click on delete button, a message should be displayed for ask permission for delete and after pressing ok button in confirmation window, display message "Project details deleted successfully."
Actual Result	Same as expected result
Status	Pass

7.8.3 COMPONENT MANAGEMENT TEST CASE

Test Case	TC-10
Test Description	Leaves all non-mandatory fields empty when adding a new project component details

Test Data	Blank component Name, description and purpose fields
Expected Result	Message should be displayed "Please enter component name."
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-11
Test Description	Enter invalid project component name.
Test Data	Component name=123
Expected Result	Message should be displayed "Please enter valid project component name."
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-12
Test Description	Enter valid project component details.
Test Data	Component name="Document data manager", description="managing all document data in an organization", Purpose=Blank
Expected Result	Message should be displayed "Component added successfully" and component details show in tabular form.
Actual Result	The message is displayed correctly and component details is visible in tabular form.
Status	Pass
Test Case	TC-13
Test Description	Click on edit link and update the project component details.
Test Data	Component name="Document data manager", description="managing all document data in an organization", Purpose="managing document data"

Expected Result	After click on edit button, project component details should be visible in editable format and after saving display message “Component details updated successfully.”
Actual Result	Same as expected result
Status	Pass
Test Case	TC-14
Test Description	Click on delete link.
Test Data	
Expected Result	After click on delete button, a message should be displayed for ask permission for delete and after pressing ok button in confirmation window, display message “Project component details deleted successfully.”
Actual Result	Same as expected result
Status	Pass

7.8.4 FUNCTIONALITY MANAGEMENT TEST CASE

Test Case	TC-15
Test Description	Leaves all non-mandatory fields empty when adding a new functionality of a component in project.
Test Data	Blank functionality Name, description.
Expected Result	Message should be displayed “Please enter functionality name.”
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-16
Test Description	Enter invalid functionality name.
Test Data	functionality name=123#\$\$%

Expected Result	Message should be displayed “Please enter valid functionality name.”
Actual Result	The message displayed correctly
Status	Pass
Test Case	TC-17
Test Description	Enter valid functionality details.
Test Data	Functionality name=”Check In saved document”, description=Blank
Expected Result	Message should be displayed “functionality added successfully” and functionality details show in tabular form.
Actual Result	The message is displayed correctly and functionality details is visible in tabular form.
Status	Pass
Test Case	TC-18
Test Description	Click on edit link and update the functionality details.
Test Data	Functionality name=”Check In saved document”, description=”viewing all saved document”.
Expected Result	After click on edit button, functionality details should be visible in editable format and after saving display message “functionality details updated successfully.”
Actual Result	Same as expected result
Status	Pass
Test Case	TC-19
Test Description	Click on delete link.
Test Data	
Expected Result	After click on delete button, a message should be displayed for ask permission for delete and after pressing ok button in

	confirmation window, display message “functionality details deleted successfully.”
Actual Result	Same as expected result
Status	Pass

7.8.6 USER MANAGEMENT TEST CASE

Test Case	TC-20
Test Description	Leaves all fields empty in user management page
Test Data	All fields are blank
Expected Result	Message to be displayed “All fields are required”
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-21
Test Description	Enter details in wrong format
Test Data	Username: 1@342, User Id: 234!@, password: 145, first name: 123, last name:456, user type:” ”, user designation:456, status: “select”
Expected Result	Message should be displayed “Please enter user details in correct format”
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-22
Test Description	Enter details in correct format
Test Data	Username: Amrish Sinha, User Id: amrish.123, password: amr@1234, first name: Amrish, last name: Sinha, user type: “developer ”, user designation: developer, status: “Active”

Expected Result	Message should be displayed “User details added successfully” and user details is displayed in tabular form with edit and delete link.
Actual Result	Same as expected result.
Status	Pass
Test Case	TC-23
Test Description	Click on edit link and update the user details.
Test Data	Username: Amrish Sinha, User Id: amrish.123, password: amr@1234, first name: Amrish, last name: Sinha, user type: “developer ”, user designation: developer, status: “Non Active”
Expected Result	After click on edit button, user details should be visible in editable format and after saving display message “user details updated successfully.”
Actual Result	Same as expected result
Status	Pass
Test Case	TC-24
Test Description	Click on delete link.
Test Data	
Expected Result	After click on delete button, a message should be displayed for ask permission for delete and after pressing ok button in confirmation window, display message “user details deleted successfully.”
Actual Result	Same as expected result
Status	Pass

7.8.7 TEAM MANAGEMENT TEST CASE

Test Case	TC-25
Test Description	Leaves all fields empty in team management page
Test Data	All fields are blank
Expected Result	Message to be displayed “All fill (*) required fields.”
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-26
Test Description	Enter details in wrong format
Test Data	Project name: “Document Management System” user name=” ”
Expected Result	Message should be displayed “Please select user name”
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-27
Test Description	Enter details in correct format
Test Data	Project name: “Document Management System” user name=” Amrish Sinha”
Expected Result	Message should be displayed “User added to team successfully” and details is displayed in tabular form with edit and delete link.
Actual Result	Same as expected result.
Status	Pass
Test Case	TC-28
Test Description	Click on edit link and update the team details.
Test Data	Project name: “Document Management System” user name=” Arpit Singh”

Expected Result	After click on edit button, team details should be visible in editable format and after saving display message “team updated successfully.”
Actual Result	Same as expected result
Status	Pass
Test Case	TC-29
Test Description	Click on delete link.
Test Data	
Expected Result	After click on delete button, a message should be displayed for ask permission for delete and after pressing ok button in confirmation window, display message “user details deleted from team successfully.”
Actual Result	Same as expected result
Status	Pass

7.8.8 ADD REQUIREMENT TEST CASE

Test Case	TC-30
Test Description	Leaves all fields empty
Test Data	All fields are blank
Expected Result	Message should be displayed “Please fill (*) required fields”
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-31
Test Description	Enter Invalid requirement details
Test Data	Invalid requirement details
Expected Result	Message to be displayed “Please enter details in valid format”

Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-32
Test Description	Enter Valid requirement details
Test Data	Task Id: DMS-01, Task: “check in process of saved data”, setupversion:”2017.8.1”, planned date:”28 April 2017”, Resource: ”Amrish Sinha”, status: “in progress”
Expected Result	Message to be displayed “requirement added Successfully”
Actual Result	Same as expected.
Status	Pass

7.8.9 SEARCH REQUIREMENT TEST CASE

Test Case	TC-33
Test Description	Leaves all fields empty and click get requirements.
Test Data	All fields are blank
Expected Result	Message should be displayed “Please select setup version or planned setup date.”
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-34
Test Description	Search requirement by setup version
Test Data	Select setup version.
Expected Result	Requirement of selected setup version displayed.
Actual Result	Same as expected.
Status	Pass

Test Case	TC-35
Test Description	Search requirement by setup date
Test Data	Planned start date="21 Jan 2017" planned end date:"22 May 2017"
Expected Result	Requirement between two date are displayed
Actual Result	Same as expected.
Status	Pass

7.8.10 SETUP MANAGEMENT TEST CASE

Test Case	TC-36
Test Description	Leaves all fields empty in add setup version page
Test Data	All fields are blank
Expected Result	Message to be displayed "please fill (*) required fields"
Actual Result	Message is displayed correctly
Status	Pass

Test Case	TC-37
Test Description	Enter details in wrong format
Test Data	Setup version: "adcjbkj", planned setup date:"123456", status:"45678"
Expected Result	Message should be displayed "Please enter details in correct format"
Actual Result	Message is displayed correctly
Status	Pass

Test Case	TC-38
Test Description	Enter details in correct format

Test Data	Setup version: "2017.6.3", planned setup date:"28 April 2017", status: "un-dispatched"
Expected Result	Message should be displayed "setup version added successfully" and setup version details is displayed in tabular form with edit and delete link.
Actual Result	Same as expected result.
Status	Pass
Test Case	TC-39
Test Description	Click on edit link and update the setup versions details.
Test Data	Setup version: "2017.6.3", planned setup date:"28 April 2017", status: "dispatched"
Expected Result	After click on edit button, setup version details should be visible in editable format and after saving display message "setup version details updated successfully."
Actual Result	Same as expected result
Status	Pass
Test Case	TC-40
Test Description	Click on delete link.
Test Data	
Expected Result	After click on delete button, a message should be displayed for ask permission for delete and after pressing ok button in confirmation window, display message "setup version details deleted successfully."
Actual Result	Same as expected result
Status	Pass

7.8.11 SCHEDULE MANAGEMENT TEST CASE

Test Case	TC-41
Test Description	Leaves week fields empty and click get schedule.
Test Data	Week=""
Expected Result	Message to be displayed "please select week date."
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-42
Test Description	Select week one week before from current week
Test Data	Week="27 March 2017- 01 April 2017" if current week is "03 April 2017-08 April 2017"
Expected Result	Message should be displayed "Schedule does not exist."
Actual Result	Message is displayed correctly
Status	Pass
Test Case	TC-43
Test Description	Enter current week date and click on get schedule.
Test Data	Current week date.
Expected Result	If schedule exist , open schedule displayed and if schedule not exist , task from un-dispatched setup version and previous schedule in progress task is displayed to create a schedule
Actual Result	Same as expected result.
Status	Pass
Test Case	TC-44
Test Description	Create schedule from un-dispatched setup version task and previous schedule in progress task.

Test Data	Select task and click on save task to schedule.
Expected Result	Selected week schedule created.
Actual Result	Same as expected result
Status	Pass
Test Case	TC-45
Test Description	Leaves all field empty in adding new task in schedule
Test Data	All field blank.
Expected Result	Message displayed "Please select (*) required fields."
Actual Result	Same as expected result
Status	Pass
Test Case	TC-46
Test Description	Fill all fields in valid format.
Test Data	Name: Amrish Sinha , Task Id=" ", Task: "update schedule", planned start date:"27 March 2017", Planned end date:"01 April 2017", Actual Start date:" ",Actual end date:" " status:" in progress"
Expected Result	Message displayed "Task added successfully."
Actual Result	Same as expected result
Status	Pass
Test Case	TC-47
Test Description	Edit task in schedule. Click on edit button in a row.
Test Data	Name: Amrish Sinha , Task Id=" ", Task: "update schedule", planned start date:"27 March 2017", Planned end date:"01 April 2017", Actual Start date:" 03 April 2017",Actual end date:"26 March 2017 " status:" complete"

Expected Result	Message displayed “Actual start date should be within a range from planned start date, planned end date and schedule start date and schedule end date”
Actual Result	Same as expected result
Status	Pass
Test Case	TC-48
Test Description	Edit task in schedule. Click on edit button in a row.
Test Data	Name: Amrish Sinha , Task Id=” ”, Task: “update schedule”, planned start date:”27 March 2017”, Planned end date:”01 April 2017”, Actual Start date:” 28 March 2017”,Actual end date:”29 March 2017 ” status:” complete”
Expected Result	Message displayed “Task updated successfully.”
Actual Result	Same as expected result
Status	Pass
Test Case	TC-49
Test Description	Enter week and select template. Click on getallschedule.
Test Data	Week:”23 May 2017 – 28 May 2017”
Expected Result	All project schedule displayed and shows project name in a message whose schedule is not created.
Actual Result	Same as expected result
Status	Pass
Test Case	TC-50
Test Description	Enter week and select template. Click on getallschedule and then click export to excel button.
Test Data	Week:”23 May 2017 – 28 May 2017”
Expected Result	Excel report of all schedule is generated with project feedback and setup plan.

Actual Result	Same as expected result
Status	Pass
Test Case	TC-51
Test Description	Enter previous week from current week. Click on getallschedule.
Test Data	Week="27 March 2017- 01 April 2017" if current week is "03 April 2017-08 April 2017"
Expected Result	If schedule exist, schedule displayed in which no task is added and only update actual start and end date and status.
Actual Result	Same as expected result
Status	Pass

CHAPTER 8

Result, Discussion &

Conclusion

RESULT AND DISCUSSION

8.1 INTERPRETATION OF RESULTS

System is fulfilling the requirements what was expected from it. User can easily know about the details of project development related information, search to own requirement. User can perform the advance search on requirement of project and requirement status and view more details of project schedule. Administrator can easily update the records. This project reduced paper work a lot and may increase time efficiency.

8.2 APPLICATION AREAS AND SCOPE OF THE WORK

The application area for this software is in any IT organization whose main area of focus is software development. By automating the existing system user can generate different reports such as task status report sent to client, task schedule report how the development takes place in organized way and system helps to maintain complex database with security and free from redundancy. It is beneficial for the users also because user not need to be go to file to search requirement, planned and schedule the development process on paper. He/she search the requirement ,planned the development process and schedule to achieve the target in stages any time through this system .The scope of this project become more increase when user find that major software development process is automated.

8.3 CONCLUSION

“Online Project Task Scheduling and Planning” is a web application through which we can maintain all the records and requirement of a developing project at one place. The objective of this project is to help the organization and employee working in an organization by maintaining the requirements of any project, planned accordingly to achieve goal in stages and update the status where we are? And next what to do?

This project act as a central repository where we can maintain all the information related the project which is developed or under developing and user get relevant information through this.

The basic objective of this project is are as follows:

- Maintain the record of all the project, its component and functionalities.
- Keep all the requirement (task) of projects in organized manner.
- Eliminate manual process of keeping record of all the project planned setup version accordingly.
- Reduce time to generate weekly schedule form requirement to develop a project.

There is always a room for improvement in any software, however efficient the system may be. The important thing is that the system should be flexible enough for future modifications. The system has been factored into different modules to make system adapt to the further changes. Every effort has been made to cover all user requirements and make it user friendly.

- **Goal achieved:** The System is able provide the interface to the user so that he can search all the necessary information related product development..
- **User friendliness:** Though the most part of the system is supposed to act in the background, efforts have been made to make the foreground interaction with user as smooth as possible. Also the integration of the system with Inventory Management project has been kept in mind throughout the development phase.

8.4 ADVANTAGES & SPECIAL FEATURES OF THE SYSTEM

- The system is developed in enterprise java environment and IDE which provides a user friendly interface.
- It is user friendly and efficient one with increase in the speed and provides an efficient.
- It is completely menu driven and even the non-computer person can operate it.
- Verification at every point prevents the redundancy in the saved data.
- It allows for remote submission of information.
- As the system is developed using enterprise java technology and the code and data is maintained very easily.
- Follows 3-tier architecture.
- Automation of the Data Entry program at administrator side.
- Development of Reports at administrator side to analyze the growth of system.

8.5 LIMITATIONS

Although efforts have been made to make the system more and more efficient and user friendly, it is felt that some enhancements can be incorporated, even now, into the control to make it more usable. Some of the limitations, which were encountered, are listed below:

- The objects created at Application Level takes up considerable memory space that can be improved.
- Here predefined component that slow the process are used, one can speed up process by using self-developed component.
- Better technology might speed up the whole process starting from filling the application form up to generation of reports.
- Not supported MVC (Model View Controller) which speed up the processing and organized the code in well-structured format.
- Moderate maintenance and modification cost.

8.6 FUTURE EXTENSIONS

As it is obvious that there is always a reason for improvement in any software package however good and efficient it may be. But important thing is that the system should be flexible enough for further modifications. This system has been designed in such a way that the modification can be done without affecting the actual design of the system. Project for **ONLINE PROJECT TASK SCHEDULING AND PLANNING** can be extended to add some other modules without affecting the system.

Factors driving the organized online website include the following:

- This project can be extended to automate the testing phase of software development process.
- This project can be used in future as a testing tool.
- New technology and lifestyle trends creating replacement demand
- It can be transformed in MVC architecture in future demand.

8.7 SCOPE OF ENHANCEMENT

Here the programs are merged in order to decrease the number of programs and to increase the performance of the programs so that they take less time in execution. But it has increased the complexity of the programs. Also their performance can be increased so that these programs take much less time in execution. So the enhancements that can be done are:

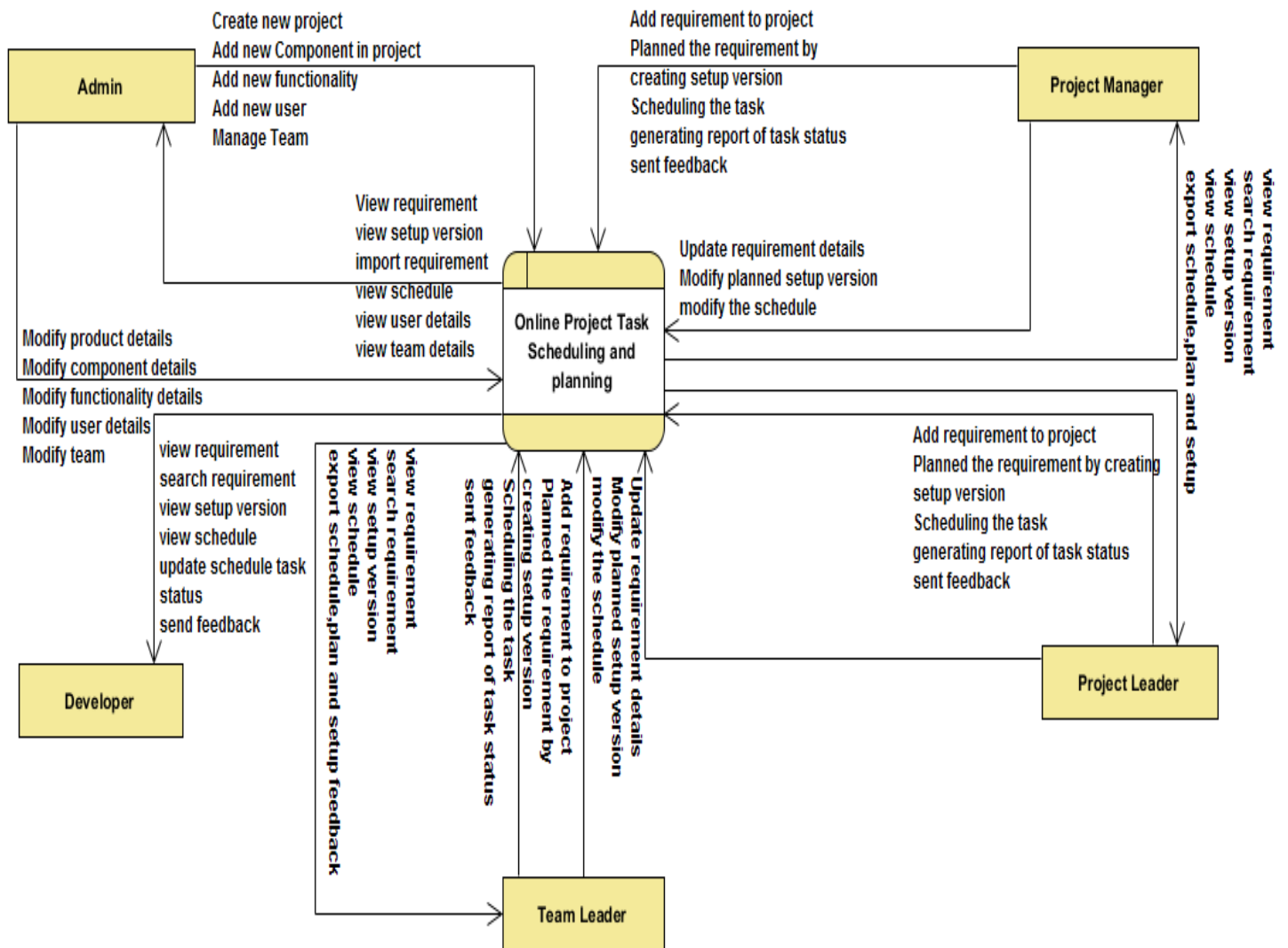
- The online website will require more secure encryption to provide more transaction.
- Live support should be enabled.
- Decrease the time of execution by tuning the programs.
- Various other modules can be integrated to this application to make it a real core entrepreneur solution.
- Decrease the complexity of the programs by improving the program logic and database design.
- This system is developed by using latest technology so it can be enhanced with an ease in near future as the requirement changes.

APPENDIX

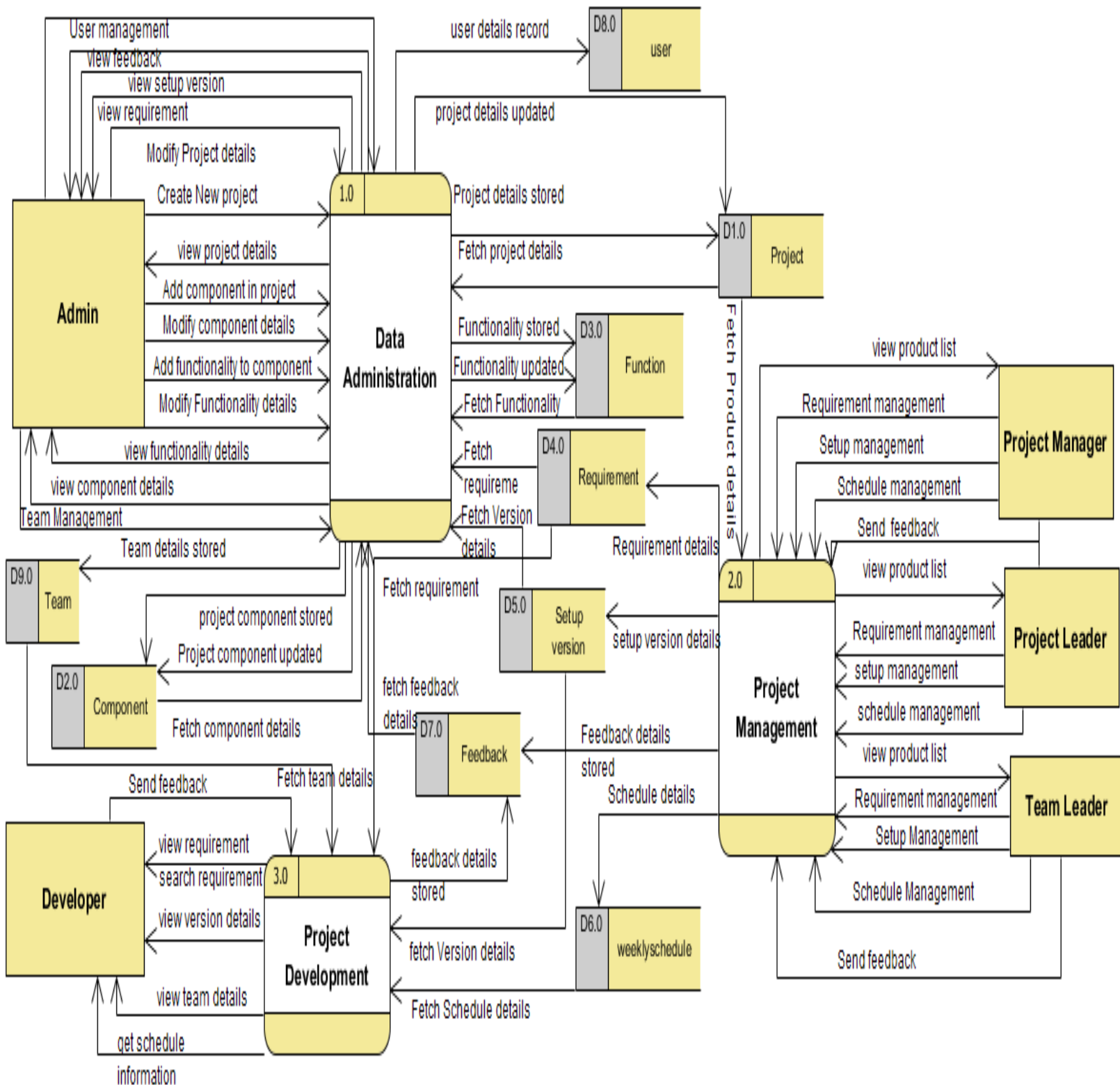
APPENDIX

DATA FLOW DIAGRAM

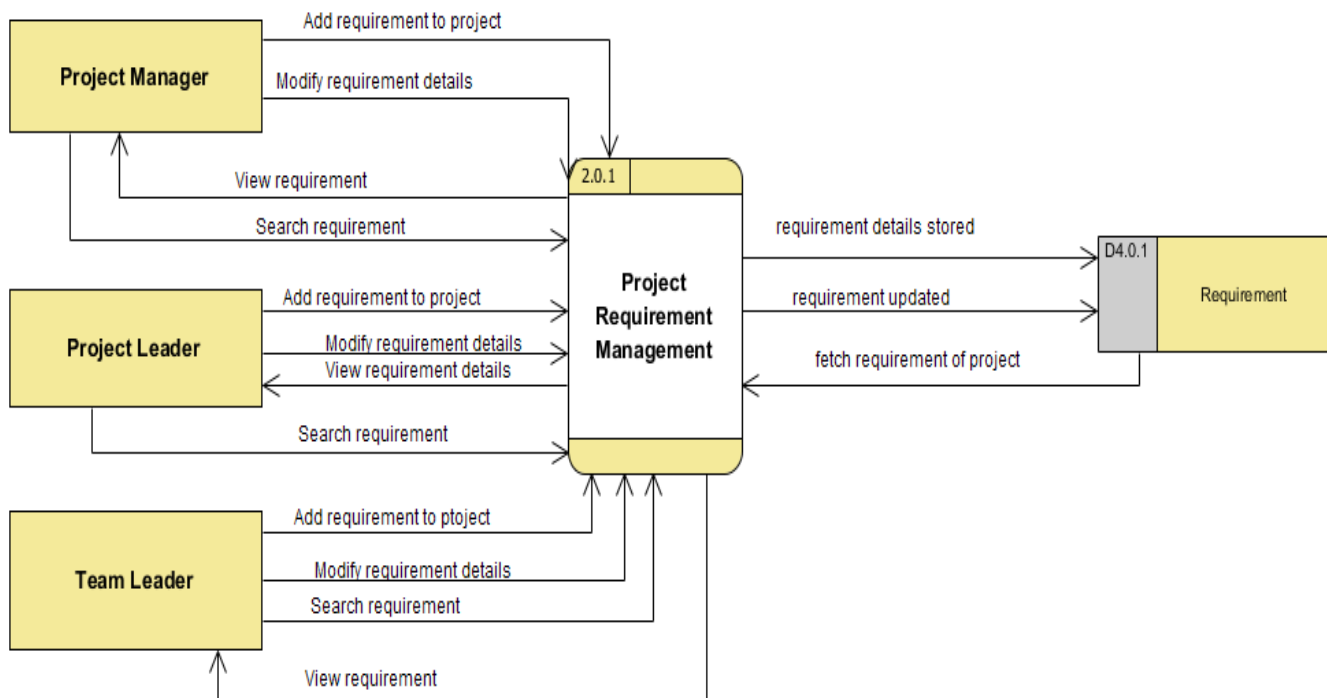
CONTEXT LEVEL DIAGARM



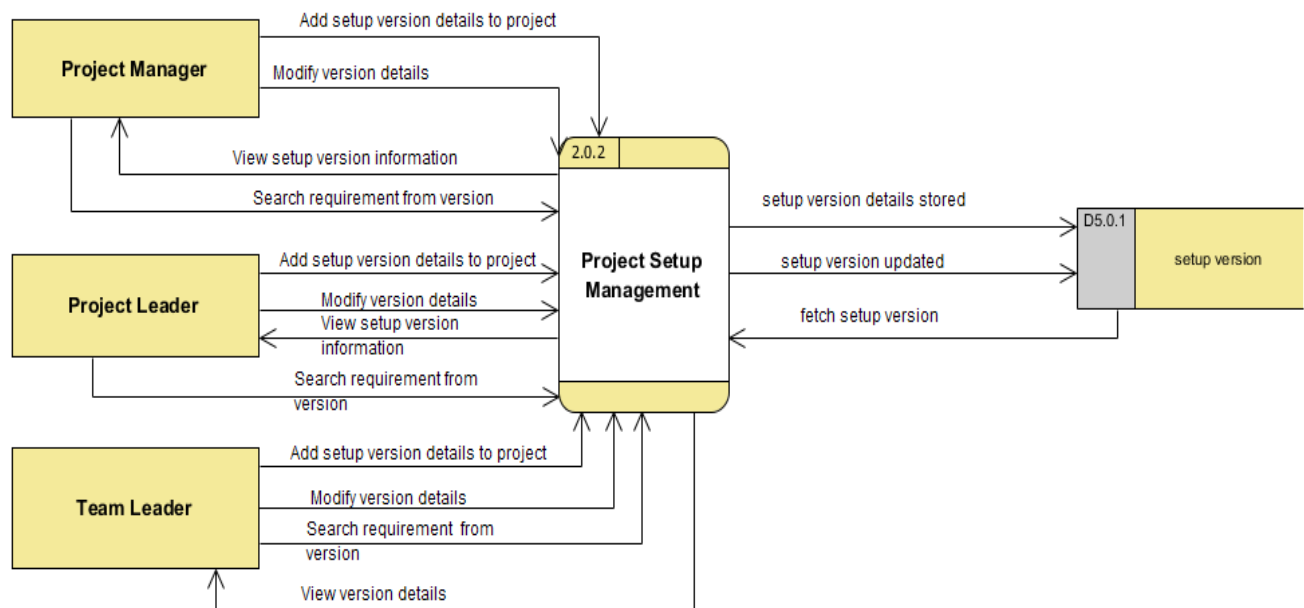
FIRST LEVEL DFD



SECOND LEVEL DFD-REQUIREMENT MANAGEMENT



SECOND LEVEL DFD-SETUP MANAGEMENT



SECOND LEVEL DFD-SCHEDULE MANAGEMENT

