PRACTICAL:- 02

Practical - 2.2:

Aim: To perform the system analysis task of your system:

Prepare Software Project Management Plan (SPMP) Document.

Program:

SPMP –Student Fees Management System.

Table Of Content :

SPMP – Student Fees Management System

Software Project Management enables a group of Software developers to work efficiently towards the successful completion of the Project.

1. Introduction:

This document is a formal plan for an Student Fees Management System. The following section give a basic background of the project including an overview, deliverables, milestones and a glossary of terms and acronyms.

1.1 Overview:

The purpose of student fees management system simplifies trivial tasks like fee submission, generate and issue receipts, online transaction, record maintenance and many more. In short, it simplifies task of fee payment and makes easier task for student to pay fees at any time.

For Example, suppose Student is using the SFMS, he/she will get an integrated fee structure and allows online payment through different gateways. The system will also collect a late payment, fines and provide required function as well. With payment integration, he/she will submit their fees online without any hassle. The reports generated via online payment includes all the important details of the transaction which student can access any time you need.

Following is the high-level functionality that the Student Fees Management will

provide.

Create a User-Experience that binds the data fetched from the server in a flawless manner. The user-friendly experience of the web-app will help the student and admin perform the following functions:

Student –

- **Registration:** If the Students wants to pay fees, he/she must be registered. Unregistered Students can't go to pay fees.
- **Login:** Student logins through the valid user id and password for Paying Online Fees.
- **View Details:** After login Student are able to see their personal details and Fees payment details.
- **Payment:** For Student there are many types of payment gateway such as online payment, debit/credit card, net banking, etc.
- **Logout:** After the payment student will logout form the application/Web.
- **Report Generation:** After all the transaction the system can generate the portable document file (.pdf) and then send one copy to Student user id and one to the system database.

Admin –

- **Student information Management:** Admin should be able to manage the students info. at any time i.e., admin should be able to change the Fees Structure and other necessary operations like adding or removing the student details.
- **Payment Monitoring**: Admin can monitor the transactions made by the Students on web/app.

• The team members have agreed that the SFMS will be developed as a Web-App. A web-app solution will provide many benefits including the following:

-The application doesn't need to be installed and maintained on any personal device. Instead it works on any web browser irrespective of any operating system.

-The web-app will be made operational 24x7 after the first deployment and shall be privy to maintenance break at the time of the deployment cycle for the updates to the platform.

1.2 Deliverables:

The table below lists the deliverables that will be provided as part of the system. Both product and project deliverables are included. Product deliverables are those items that are created to produce the system. Project deliverables are those items are created to support the project.

Deliverables	Description	Dates
Software Project Management Plan (CRMS)	A complete formal project plan, including technical and managerial processes that will be implemented in the development and delivery of the system	
Software Requirements Specification (SRS)	A formal document detailing the functional and non-functional requirements of the system	
Test Scenarios	Formal documentation detailing scenarios that must be followed in order to ensure that the product software is satisfactorily tested	
Final Presentation	A demonstration of the product software and a presentation of the project experience	

Table 1.2.1 Deliverables

1.3 Evolution of the SPMP:

All members of the project team have agreed upon this SPMP. Any team member may make changes to the SPMP, but the entire project team must approve all changes. This document

may be periodically updated during the lifecycle of the project to reflect changes in the project schedule. All changes will be documented in order to keep the SPMP current.

1.4 Reference Materials

- IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes.
- IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans
- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications
- https://www.geeksforgeeks.org/software-engineering-software-project- management-plan-spmp/

1.5 Definitions and Acronyms:

This section provides a quick reference for the acronyms used throughout this document.

SFMS→ Student Fees Management System

SQL → Structured Query Language

GUI →Graphical User Interface

SRS →Software Requirement Specification

SPMP →Software Project Management Plan

2. Project Organization

The following sections explain the organizational structure for the project, the specific roles and responsibilities for each team member, and the process model that will be used.

2.1 Process Model:

The main process model for this project will be an agile model, Agile SFMS models is a combination of iterative and incremental process models with focus on process adaptability and customers satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds, or we can say small parts. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements At the end of the iteration, a working product is displayed to the customer and important stakeholders. The time to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change.

DIAGRAM:

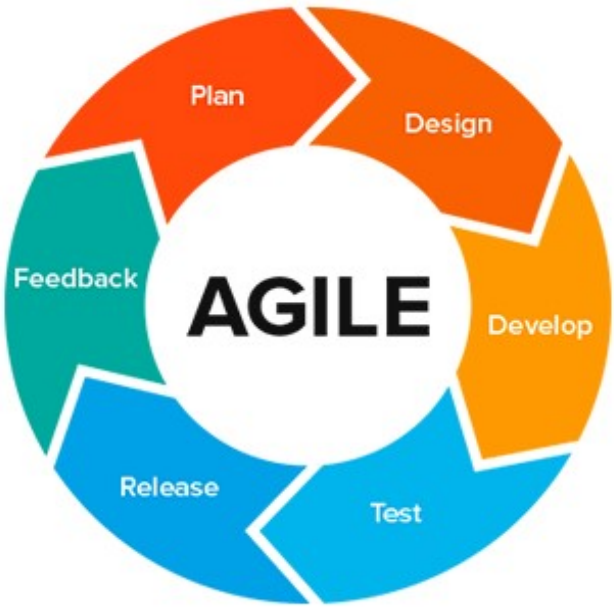

Figure 2.1.1 Agile Model

2.2 Work break down:

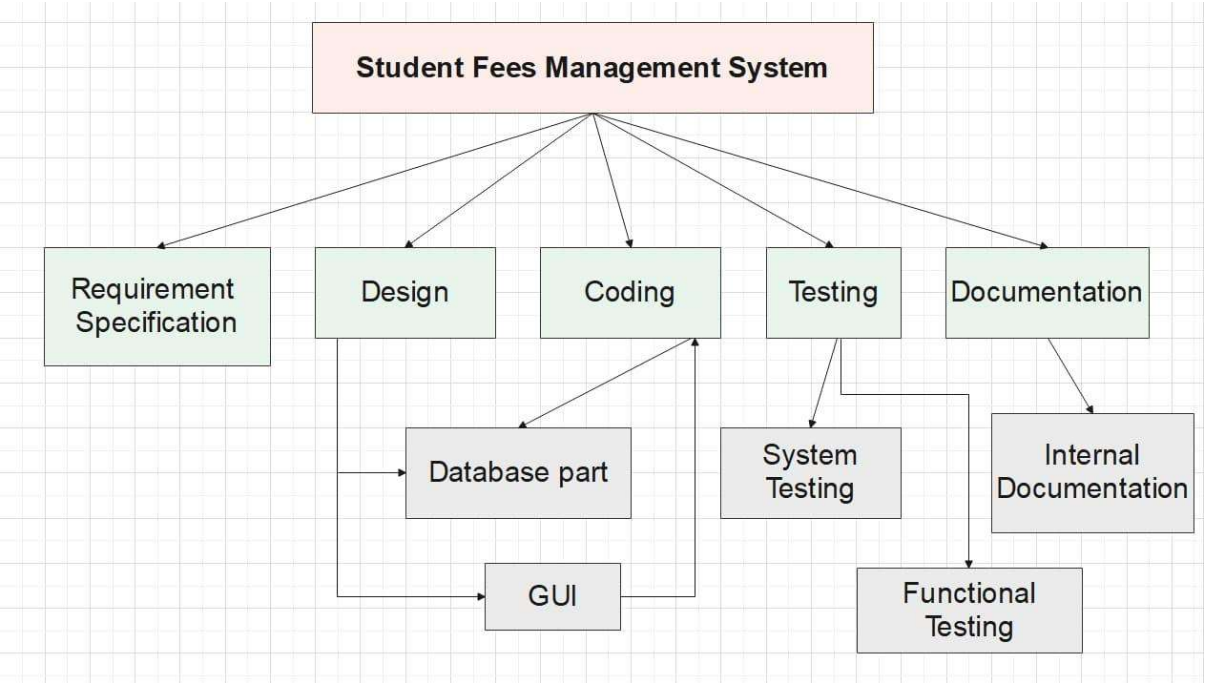

Figure 2.2.1 Work break down

2.3 Roles and responsibility:

The following table lists the roles and responsibilities necessary to complete the project.

Role	Responsibility
Project Manager	Plans, organizes, and coordinates the team project. Schedules and prepares team meetings. Resolves conflicts. Works as a liaison between team members. Monitors and reports the weekly status of the team. Ensures that project deliverables are met.
Requirements Analyst	Identifies, prioritizes, and documents requirements that satisfy the customer's needs. Clearly communicates the requirements to the Application Developers. Defines acceptance criteria for the completion of the solution.
Application Designer	Designs a solution to the problem that satisfies the requirements. Determines the data needs for the solution. Determines what hardware and tools are necessary. Assists the Technical Writers in documenting the design.

Application Developer	Develops the software solution. Determines the data needs for the solution. Determines what hardware and tools are necessary. Fixes bugs found by the Quality Assurance Testers. Assists the Technical Writers in creating online help, and writing the user manual.
Quality Assurance Engineer	Tests the software solution to identify bugs and ensure that the requirements have been met. Communicates problems found with the solution to the Application Developers. Retests bug fixes. Determines whether the software solution is ready for delivery.
Technical Writer	Coordinates the project documents and their review by all team members. Collects, proofreads, and integrates document parts. Generates the final version of all the documents.

3. **Project estimates:**

3.1 Estimation Technique Used:

We have used Constructive Cost Model (COCOMO) and it is procedural software cost estimation model and also used for estimating effort and schedule for software projects.

3.2 Efforts, Resource Cost and Project Duration Estimation:

(Assumption: Our project falls in the category of organic mode software project)
INPUT: The no. of lines: 14000
Suppose LOC (lines of code) = 20,000
Effort $E = a (KLOC)^{b}$
$= 2.4 (14)^{1.05}$
= 38 person – month

Where E = Effort in person – month

D = Development time in months

kLOC = Estimated numbers of Line of Code

Time $= c (Effort)^{d}$
$= 2.5 (38)^{0.38}$
= 10 months(approx)

Person estimation P = (E/D)
= (38/10)
= 4 person-month(approx)

4. Managerial Process:

4.1 Project Dimensions:

Project Dimension	Fixed	Flexible
Resources	×	
Hours Worked		×
Schedule	×	
Scope		×

Table 4.1.1 Project Dimension

The number of resources for the project is fixed. The scope and the hours worked are the dimensions that can be adjusted if necessary.

4.2 Assumptions, Dependencies and Constraints:

The project will be developed under the following assumptions and Dependencies:

- It is assumed that our website is compatible to every web browser.
- It is also responsive in different devices.
- It is assumed that server down time is zero.

The project will be developed under the following constraints:

- All the inputs should be checked for validation and message should be given for the improper data. The invalid data are to be ignored and error messages should be given to the user.
- Fees information provided by Admin are accurate.

4.3 Risk Management:

The project doesn't contain any major risk which may crash the system but though there are some risks which may affect the system. The risk which may affect the system are as below:

1. Project Risk:

Project risks concern differ forms of budgetary, schedule, personnel, resource, and customer-related problems. It is very tough to control something which cannot be identified.

2. Technical Risk:

Technical risks concern potential method, implementation, interfacing, testing, and maintenance issue. It also consists of an ambiguous specification, incomplete specification, changing specification, technical uncertainty, and technical obsolescence.

❖ **Risk Analysis**:

1. Security
2. Maintenance
3. Access Loading
4. Database Crash

4.4 Project Tracking and Control Plan:

Sr. No.	Person	Task	Target
1.	Vishakha Kharvasiya	Searching Functionality Implement	3 Days
2.	Dinkel Mistry	UX- Design	10 Days
3.	Vishakha Kharvasiya	Testing	15 Days
4.	Nidhi Gupta	Coding	3 Week
5.	Dinkel Mistry	GUI part	4 Week
6.	Nidhi Gupta	Integration	3 Week
7.	Nidhi Gupta	Database	3 Week

Table 4.4.2 Monitoring and Controlling

5. Technical Process:

Student Fees Management System will be developed using an Agile model. It is very flexible, Speedy and its processes follows a continuous improvement life cycle. In it each iteration involves a team working through a full software life cycle including planning, requirement analysis, design, coding and testing.

SFMS will be a web-based application supporting the following browsers:

- Google chrome
- Mozilla Firefox
- Internet Explorer
- Opera

It will be developed using the following technologies.

- Visual Studio Code (VS Code)
- HTML5, CSS3, JavaScript.
- It will run on MySQL Database.
- For Backend php.

6. Task Dependency, Scheduling and Other Plans:

The following sections explain what tasks are to be done and by whom. This plan may become more detailed as the project progresses and the risk share better understood.

6.1 Dependencies:

The dependencies for the project are outlined below.

Task	Dependencies
Software project Management Plan (SPMP)_	None
Software Requirements Specification (SRS)	SPMP
Coding	SPMP, SRS
Test Scenarios	SPMP, SRS
User Guide	SPMP, SRS
Final Presentation	SPMP, SRS

Table 6.1.1 Task Dependency

6.2 Resource Requirement:

The project team will be do all of the work on the project. No additional resources will be required. The following table details the expected person hours for each major task.

Task	Efforts in person hours
Software project Management Plan (SPMP)	40
Software Requirements Specification (SRS)	72
Coding	100
User Guide	30
Final Presentation	24
Total Effort	266

Table 6.2.1 Resource Requirement

6.3 Quality Assurance Plan:

Sr. No.	Person	Module	Duration	Report	Submit
1.	Vishakha Kharvasiya	1	2 Days	Quality Report of modules	Nidhi Gupta
		2	5 Days		
2.	Nidhi Gupta	3	10 Days		Dinkel Mistry
		4	2 Days		
3.	Dinkel Mistry	5	3 Days		Vishakha Kharvasiya
		6	1 Day		
4.	Vishakha Kharvasiya	7	2 Days	Quality Report of Functionality and modules	Nidhi Gupta
		8	4 Days		
5.	Nidhi Gupta	9	5 Days		Dinkel Mistry
		10	6 Days		
6.	Dinkel Mistry	11	7 Days		Vishakha Kharvasiya
		12	10 Days		

Table 6.3.1 Quality Assurance Plan

6.4 Schedule:

	Start Date	Duration in days
Project Initiation	3 Jul	2
SPMP	6 Jul	8
SRS	13 Jul	15
Coding	28 Jul	26
Test Scenarios	23 Aug	18
User guide	11 Sep	15
Final Presentation	25 Sep	10

Table 6.4.1 Schedule

Gantt Chart:

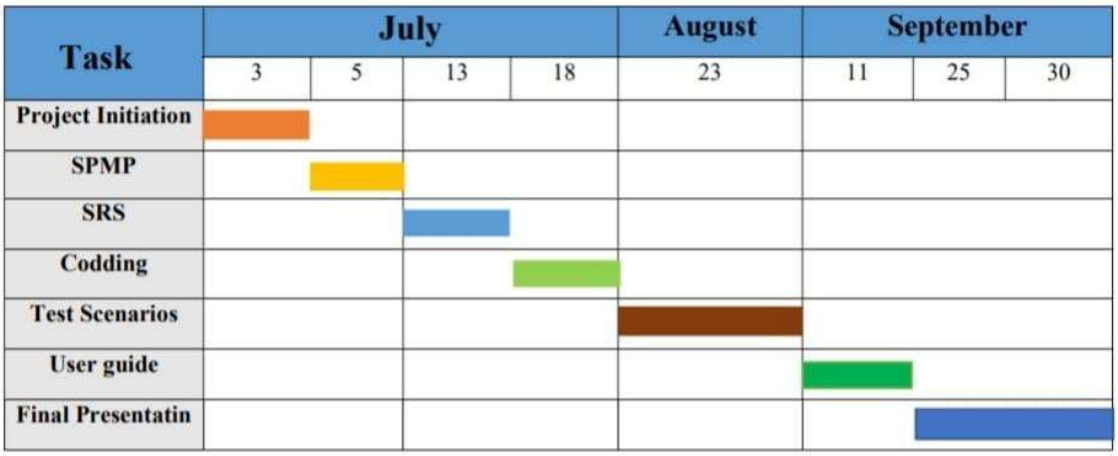

Figure 6.4.2 Gantt Chart