PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

Format 3

Project Plan Document

Instructions:

1: Prepare a detailed plan for your project which comprises of the below mentioned details.

2: Upload pdf document through the given link.

3: The name of the document should be your Project ID.

Things to be included as part of the project plan.

1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model. (PES2UG20CS402, PES2UG20CS405)

We will be using **Extreme Programming (XP) lifecycle** for the execution of the project.

What is Extreme Programming(XP)?

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

Features of XP programming to choose this model are

- **Test-Driven Development:** Priority is given to testing, enabling us to create an error-free code.
- **Communication:** Meetings are kept every 1-2 weeks with a project report to discuss the development of the project and to plan accordingly.
- **Implementation:** Helps in creating a good design, eliminating dependencies within the system.
- **Pair Programming:** means all production software is developed by two people sitting at the same machine. The idea behind this practice is that two brains and four eyes are better than one brain and two eyes. You effectively get a continuous code review and quicker response to nagging problems that may stop one person dead in their tracks.
- **Feedback Loops:** The charm of Extreme Programming is continuous feedback that keeps everyone focused and development continues in the right direction without any delays. In Extreme Programming, feedback is accomplished at different levels, to the

required and sufficient detail. This is done continuously and constantly across the iterations and releases as well.

- **Project Management:** In Extreme Programming, Project Management is not given emphasis and the manager role performs the minimal and the most essential management activities**.**

Values of XP:

The five values of XP are communication, simplicity, feedback, courage, and respect which are described in more detail below.

- **Communication**
  - Software development is inherently a team sport that relies on communication to transfer knowledge from one team member to everyone else on the team. XP stresses the importance of the appropriate kind of communication – face-to-face discussion with the aid of a whiteboard or other drawing mechanisms.
- **Simplicity**
  - Simplicity means "what is the simplest thing that will work?" The purpose of this is to avoid waste and do only necessary things such as keep the design of the system as simple as possible so that it is easier to maintain, support, and revise. Simplicity also means addressing only the requirements that you know about; don't try to predict the future.
- **Feedback**
  - Through constant feedback about their previous efforts, teams can identify areas for improvement and revise their practices. Feedback also supports simple design. Your team builds something, gathers feedback on your design and implementation, and then adjusts your product going forward.
- **Courage**
  - Preference for action based on other principles so that the results aren't harmful to the team. The team needs the courage to raise organizational issues that reduce your team's effectiveness. The team needs the courage to stop doing something that doesn't work and try something else. You need the courage to accept and act on feedback, even when it's difficult to accept.
- **Respect**
  - The members of the team need to respect each other to communicate with each other, provide and accept feedback that honors your relationship, and work together to identify simple designs and solutions.

2: Identify the tools which u want to use it throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool. (PES2UG20CS403, PES2UG20CS405)

- **JIRA Software:**
  - Jira Software is part of a product family that helps teams of all kinds to manage their jobs. Jira was originally designed as a tracker for problems and bugs. Today, however, Jira has become a powerful work management tool for all types of applications, from requirements and test case management to agile software. You will learn in this guide what Jira's features and features can help your team with your unique needs.
  - It makes workflow easy by tracking issues and bugs related to software and mobile applications. The products developed on the Jira platform helps the teams in planning, assigning task, tracking project issues, and managing the work.
- **Draw.io Software:**
  - The Draw.io platform is a free-to-use online diagram app and editor. By using Draw.io you can also create high-quality designs such as custom flow charts, complex network diagrams, and Unified Modelling Language (UML) system layouts. This makes Draw.io an excellent way to create visual aids when you are on a tight budget.

- Through the usage of draw.io software, we have made the following design to date:
    1. Use-Case Diagram
    2. E-R Diagram
    3. WBS diagram

- **GIT/GitHub:**
    - GitHub is a web-based hosting service for Git repositories that allows you to create a remote copy of your local version-controlled project. This can be used as a backup or archive of your project or make it accessible to you and to your colleagues so you can work collaboratively.
    - We will use it to keep a check on the versions of our software and it can also be used for bug tracking as well.
    - We will also use it to divide works using branches allowed in git-tree for a cooperative/coordinating working environment.
- **VS code/Extensions:**
    - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, and preferences, and install extensions that add additional functionality.
    - We are using this software as a project development tool
- **GanttPRO Software**:
    - GanttPRO is online project management software that facilitates project scheduling and implementation with the help of Gantt charts.
    - for creating a plan to work around deadlines and properly allocate resources. They depict, among other things, the relationship between the start and end dates of tasks, milestones, and dependent tasks.
- **Figma:**
    - Figma is a cloud-based design and prototyping tool for digital projects. It's made so that users can collaborate on projects and work pretty much anywhere.
    - Figma is a digital design and prototyping tool. It is a UI and UX design application that you can use it to create websites, apps, or smaller user interface components that can be integrated into other projects.
    - We will be using the following for designing UI/UX for our software.

3: Determine all the deliverables and categorise them as reuse/build components and justify the same. (PES2UG20CS402, PES2UG20CS403)

Since we are making a small project, we will mainly use a reuse method as it is time effective and easy to deliver.

Reuse:

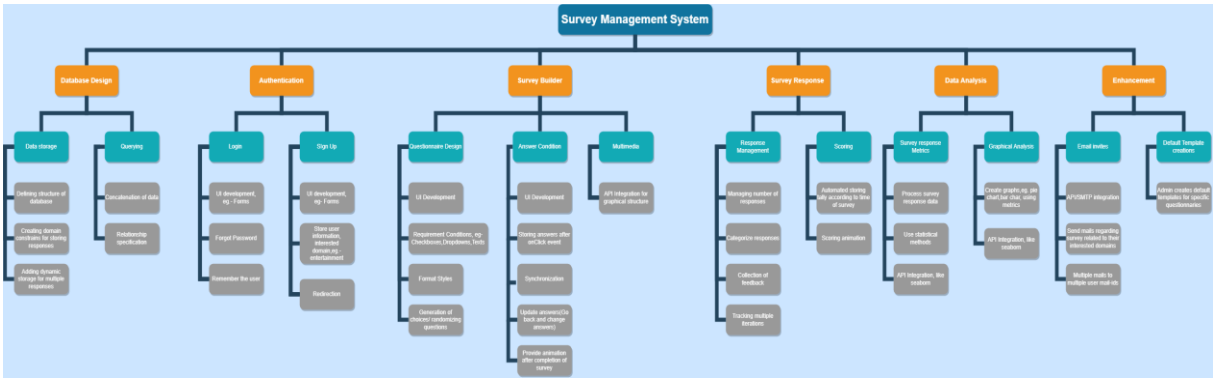
- **System Specification**: a technical document that provides a framework for the software development process.
- **User Interface**: We will be using Product Lines for similar designs in our software. Reusable design patterns will be created as a part of UI-UX designing in Build Components
- **Backend Development**: Involves reusability of build components for database interaction, Synchronization, Authentication, etc.
- **Data Analysis**: We will be integrating API calls to obtain the required analysis and statistics for each survey data.
- **Setting up of environment**: We will be using the default operating system's environment.

Build Components:

- **UI-UX design**: We will be designing the front-end ourselves from scratch as per our requirements.

- **Database Management**: It's easy to implement and we won't be getting the required attributes and relations to reuse.
- **Backend Development**: The build components will include authentication development from scratch, creating API calls to our database, making a synchronization function, etc.

4: Create a WBS for the entire functionalities in detail. (PES2UG20CS403, PES2UG20CS402)



5: Do a rough estimate of effort required to accomplish each task in terms of person months. (PES2UG20CS403)

To do a rough estimate of the Effort required as well as a rough estimate of time, we make use of the **CoCoMo model-**

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e. number of Lines of Code. It is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time, and quality. It was proposed by Barry Boehm in 1981 and is based on the study of 63 projects, which makes it one of the best-documented models. The key parameters which define the quality of any software products, which are also an outcome of Cocomo are primarily **Effort & Schedule**:

Effort: Amount of labour that will be required to complete a task. It is measured in person-months units.

Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks, and months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of the constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below. Boehm's definition of organic, semidetached, and embedded systems:

Organic – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

Semi-detached – A software project is said to be a Semi-detached type if the vital characteristics such as team size, experience, and knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. E.g.: Compilers or different Embedded Systems can be considered of Semi-Detached type.

Embedded – A software project requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two

models and also the developers need to be sufficiently experienced and creative to develop such complex models.

There are three types of COCOMO models:

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

We are going to use the **Organic Basic COCOMO model** to estimate the effort required to accomplish each task in terms of person-months: -

Effort= a(KLOC)^b person-months

time= c(Effort)^d months

Person required = Effort/ time

Basic COCOMO Model Values:

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Therefore,

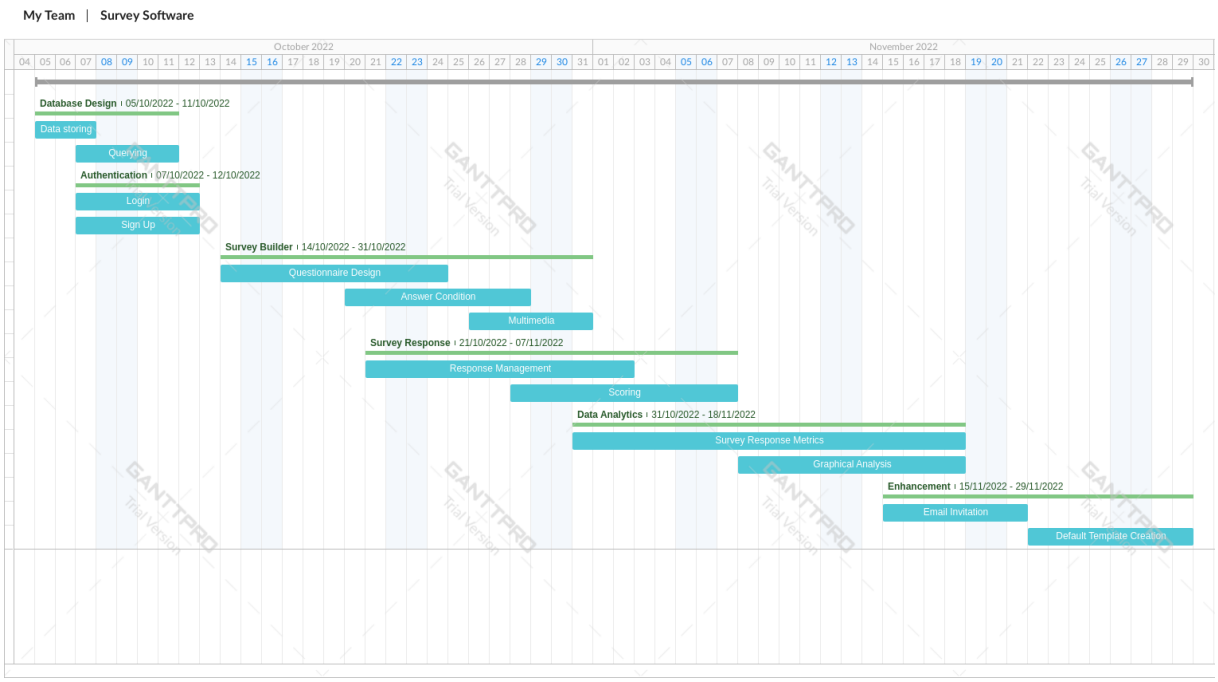
Assuming we are going to write 1,500 lines of code, i.e., 1.5 KLOC, then:

Effort(E) = 2.4 * (1.5) ^1.05 = 3.67 person-months

Time(T) = 2.5 * (3.67) ^0.38 = 4.097 Months

Person required = 0.8957 Person ≈ 1 Person

6: Create the Gantt Chart for scheduling using any tool. (PES2UG20CS416)



What is **Gantt-Chart:** Generalized Activity Normalization Time Table?

GanttPRO is online project management software that facilitates project scheduling and implementation with the help of Gantt charts.
GanttPRO allows making Gantt charts for simple and complex projects, tracking their progress, organizing tasks and subtasks in a way you need, etc. This displays the work distribution and the deadline for each task.