Project Plan Document

For

COVITEAM
Collaborative Virtual Teammate Network

Prepared by

PES2UG21CS175 | GAURAV B V
PES2UG21CS172 | GAGAN R
PES2UG21CS140 | CHARAN S GOWDA

INDEX

1.LIFECYCLE MODEL

We will be using ITERATIVE lifecycle for the execution of our project in which continuous refinement and incremental development are essential.

Reasons why we feel that iterative lifecycle is best suited for our project:

- **Continuous Improvement**: The iterative model emphasizes continuous improvement through repeated cycles. This is best suited with our goal of creating a dynamic and evolving educational platform. Each iteration allows for the incorporation of user feedback and the refinement of features.

- **Incremental Development:** Coviteam can be developed in increments, focusing on specific features or modules in each iteration. This approach enables us to deliver functional components of the platform early and receive feedback.

- **User-Centric Approach:** An iterative model encourages strong user involvement throughout the development process. This applies particularly in our project, where user feedback and needs drive feature enhancements.

- **Risk Mitigation:** By dividing the project into manageable iterations, risks can be identified and addressed early in our project.

2.TOOLS

The following tools will be used throughout the lifecycle of our project

- Planning Tool:

  Jira is a versatile project management and issue tracking tool that can be used for project planning, task assignment, and Agile project management.
- Design Tools:

  Figma is a cloud-based design and prototyping tool that supports collaborative design and wireframing, which is valuable for creating the user interface of Coviteam.
- Version Control:

  Git is a distributed version control system, and **GitHub** is a web-based platform that provides Git repository hosting. This combination allows for version control, collaboration, and code review.
- Development Tools:

Visual Studio Code is a versatile code editor with a wide range of extensions for different programming languages.

- Bug Tracking:

  Besides its planning capabilities, **Jira** also provides robust issue tracking, making it suitable for tracking and managing bugs throughout the development process.

- Testing Tools:

  Selenium is an open-source testing framework for web applications. It's suitable for automated testing of the Coviteam web-based user interface.

.3.DELIVERABLES

- **Reuse Components**:
  - Third-Party Libraries and Frameworks:

    Justification: Many software projects, including Coviteam, rely on third-party libraries and frameworks.These components are considered "reuse" as they are not developed from scratch but incorporated to provide specific functionality. For example, the use of a web framework like Flutter (front-end) and Go (back-end) can be considered reuse.

  - Database Management System (DBMS):

    Justification: The Coviteam project utilizes a DBMS, such as MySQL 8.0, for data storage and retrieval. The DBMS

is a fundamental component for most software applications and can be categorized as "reuse" because it is an existing technology used as is.

- Operating Systems:
  Justification: The project specifies compatibility with Linux and Windows Server as operating systems. These are standard, existing platforms that are not developed but leveraged for system deployment.
- Web Servers:
  Justification: The Coviteam system relies on web servers like Apache or Nginx. These are mature, existing solutions that are reused for hosting the application.
- User Authentication Mechanisms:
  Justification: User authentication is a crucial component, and many projects leverage existing solutions for this purpose. Coviteam may integrate with third-party authentication providers or use industry-standard authentication mechanisms like OAuth. These are considered reuse components.

- **Build Components:**

- User Interface (UI):

  Justification: The user interface of Coviteam, including the web-based interface, is a custom design and build component. It is tailored to the project's unique requirements and user experience goals.

- Business Logic:

  Justification: The business logic of the Coviteam application, which includes features like project management, task assignment, and document collaboration, is a custom-built component. It is developed specifically for the project's unique functional requirements.

- Project-Specific Code:

  Justification: Any code, scripts, or components that are designed to meet the project's unique requirements and functionalities are considered "build" components. This encompasses all the custom code developed exclusively for Coviteam.

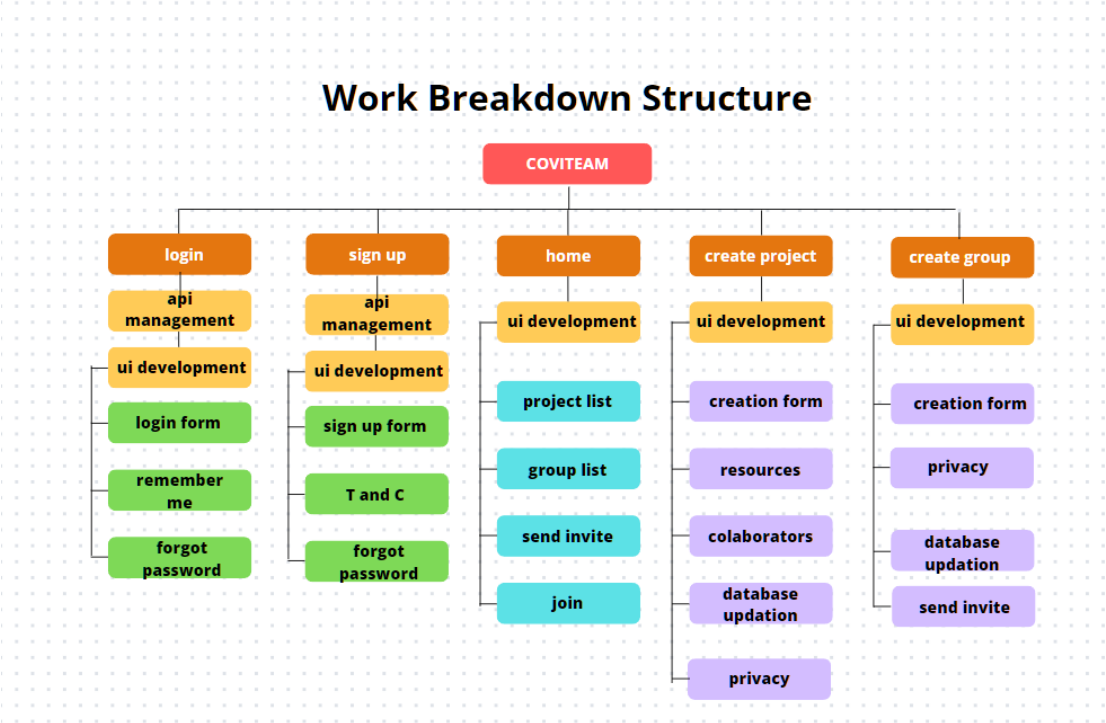
- Custom Reports and Analytics:

Justification: Coviteam's reporting and analytics features are customized for the platform's specific needs. This code is developed from scratch and is tailored to generate insights for improving team efficiency.

- Unique Documentation:

  Justification: While there may be references to existing educational platforms, the documentation for Coviteam, such as user guides, tutorials, and project-specific documents, is unique and built to cater to the platform's users.

4.WBS



5.EFFORT ESTIMATION

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

Using Cocomo model (semi-detached),

a=3.0
b=1.12
c=2.5
d=0.35
$Effort = \mathrm{a} \times (KLOC)^{\mathrm{b}}$

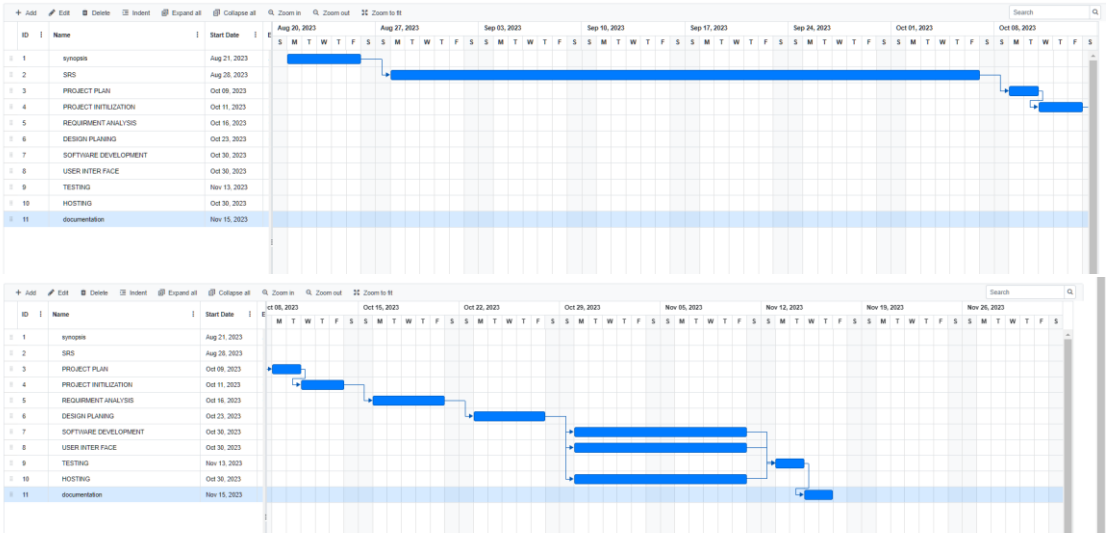
$Time = \mathrm{c} \times (Effort)^{\mathrm{d}}$

KLOC =1

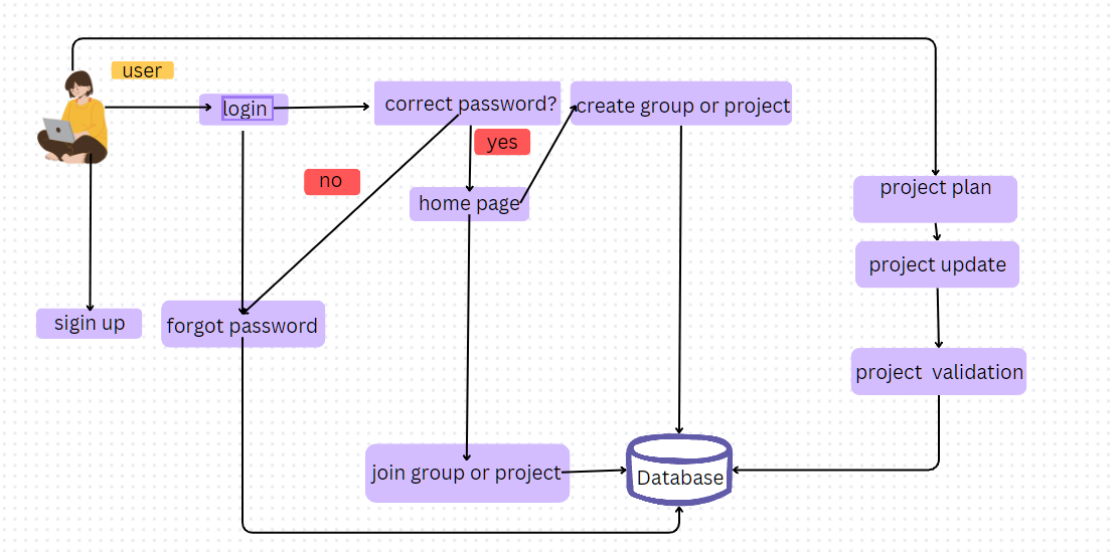
Effort = $3 \times (1)^{1.12}$
= 3 Person months
Time = $2.5 \times (3)^{0.35}$
= 3.6 months
With 3 people in team,
Time taken = 3.6722 /3

= 1.22 months

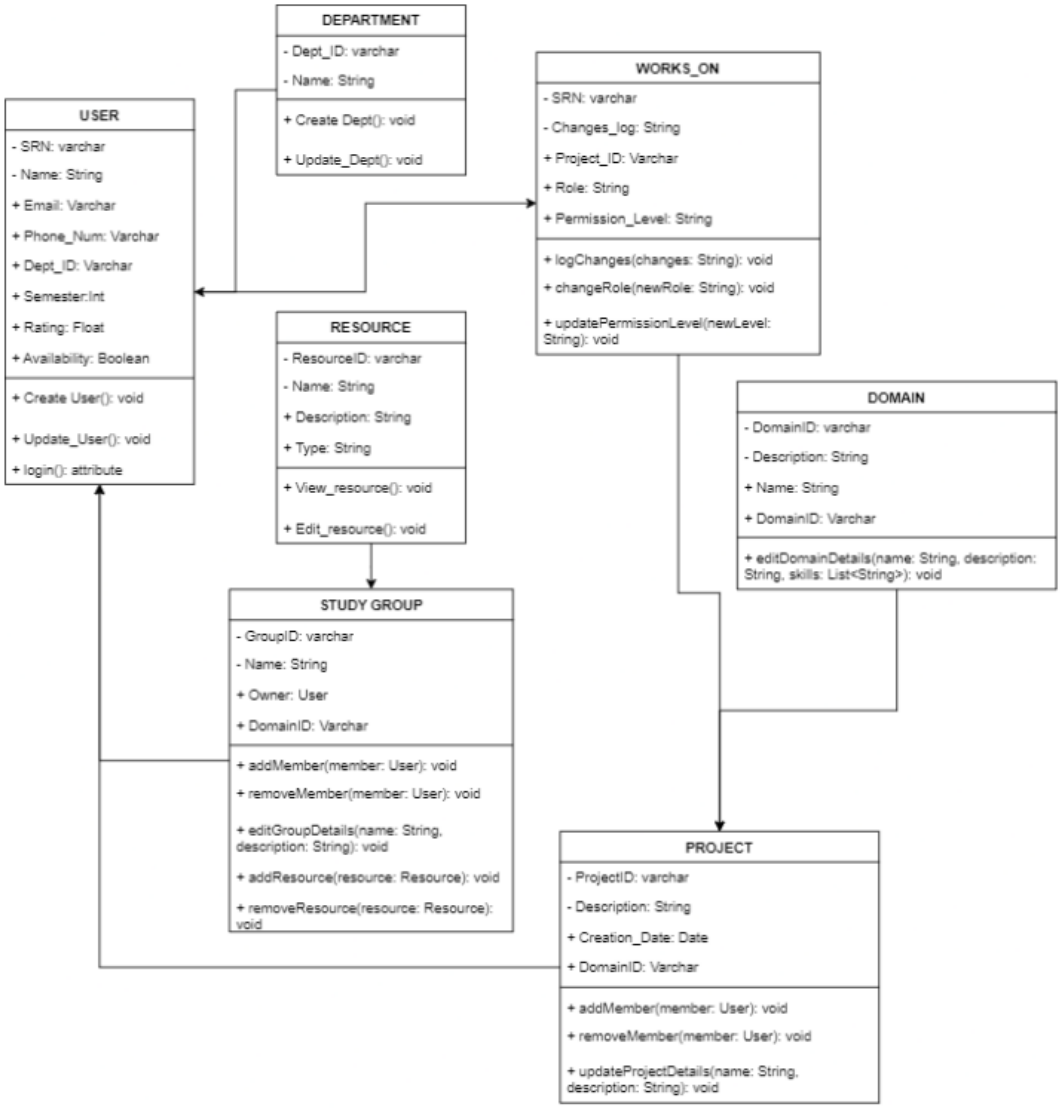
6.GANTT CHART



7.ARCHITECTURE DIAGRAM



8.CLASS DIAGRAM



*****