Project Planning Document

for

Co-Lancer

Version 1.0 approved

Prepared by

PES2UG21CS374- POOJA SATHEESH

PES2UG21CS396 – PREETHI M.

PES UNIVERSITY

19/11/2023

Project Planning Document

Lifecycle Method

The **incremental model** is used as the life cycle method for the Co-Lancer project.

The incremental model is beneficial for a freelancer collaboration website project because it allows for the early delivery of core features, enables quick response to changes in the dynamic freelance industry, facilitates continuous feedback from stakeholders, mitigates overall project risks, supports parallel development by different teams or freelancers, enhances client engagement through tangible progress, and streamlines testing and debugging processes. This approach ensures a flexible and adaptive development process, aligning the evolving project with user expectations and market needs.

Features of incremental model

Early Delivery of Core Features: With the incremental model, you can deliver essential features in the early stages of development. This allows users (freelancers and clients) to start using and benefiting from the system sooner rather than having to wait until the entire project is complete.

Quick Response to Changes: The freelance industry and collaboration requirements can evolve rapidly. The incremental model allows for flexibility and adaptability, making it easier to respond to changing market needs, user feedback, or emerging trends during the development process.

Continuous Feedback: Each increment is a deliverable version of the product. This allows stakeholders to provide feedback on the implemented features early in the development process. This iterative feedback loop can help ensure that the final product aligns closely with user expectations and requirements.

Risk Management: By breaking the project into smaller increments, the overall risk is reduced. If issues arise in one increment, they can be addressed before moving on to the next.

Parallel Development: Different parts of the system can be developed concurrently by different teams or individuals. This parallel development can accelerate the overall development process and is particularly useful in a collaborative environment where multiple freelancers may be contributing to various aspects of the project.

Client Engagement: Clients or end-users can see tangible progress with each increment, which can enhance client satisfaction and engagement. It also helps in building trust between the development team and clients as they can witness the project evolving in real-time.

Easier Testing and Debugging: Testing and debugging can be performed incrementally, making it easier to identify and fix issues in specific parts of the system. This approach streamlines the testing process and ensures a more reliable final product.

Iterations:

Iteration 1:

Planning and initial requirements gathering:

Define the high level overview of required features and objectives. Identify all the functional and non-functional features and formulate a SRS document that can benefit all stakeholders.

Iteration 2:

Design Phase:

Design the ER diagram, convert ER Diagram to Relational Schema. Create Data Flow Diagrams (DFDs) along with Use Case diagrams and Class diagrams.

Iteration 3:

Database formation:

Create a database schema based on the relational schema and grant necessary permissions to user roles based on the principle of Least Access Privileges.

Iteration 4:

Start project implementation

- Create a user registration page along with the client and freelancer registrations. Also build the Login page.
- Include password hashing and security constraints and validation of all form inputs to avoid sql injections
- Do unit testing to ensure it works as expected

Iteration 5:

Create, View and Join Projects

- Build the create project feature for clients and create the view and join project features for the freelancers.
- Search feature to search by domain of projects.
- Build a recap of completed and in progress projects and their monthly earnings.
- Unit testing

Iteration 6:

Payment Portal

A payment portal for clients to pay compensation to the freelancer when finalising the project submitted by the freelancer.

Iteration 7:

Chat feature

- Option for freelancers to chat with other freelancers in near real-time when working on a collaborated project.
- Chat should be deleted once the project is submitted.
- Unit test all possibilities.

Iteration 8:

Refinements and system testing:

Make any refinements in UI, usability and features,
System testing for validation of all features.

Iteration 9:

Add any additional features if required
Get user feedback and make changes.

Tools to use in the entire lifecycle

Design:
Tool: Canva, DrawIO
Use: Wireframe of user interfaces and design of ER diagrams and other design diagrams

Development
Tool: Visual Studio Code, NeoVim or any IDE
Use: To write source code and perform any basic debugging and testing.

Version Control and Collaboration:
Tool: Git and Github
Use: Seamless collaboration and versioning of codebase

Testing:
Tool: Postman
Use: Test backend API calls through customised request frames.

Deployment:
Services: Vercel, AWS RDS
Use: Database hosting and website deployment

Deliverables categorised as reuse/build components

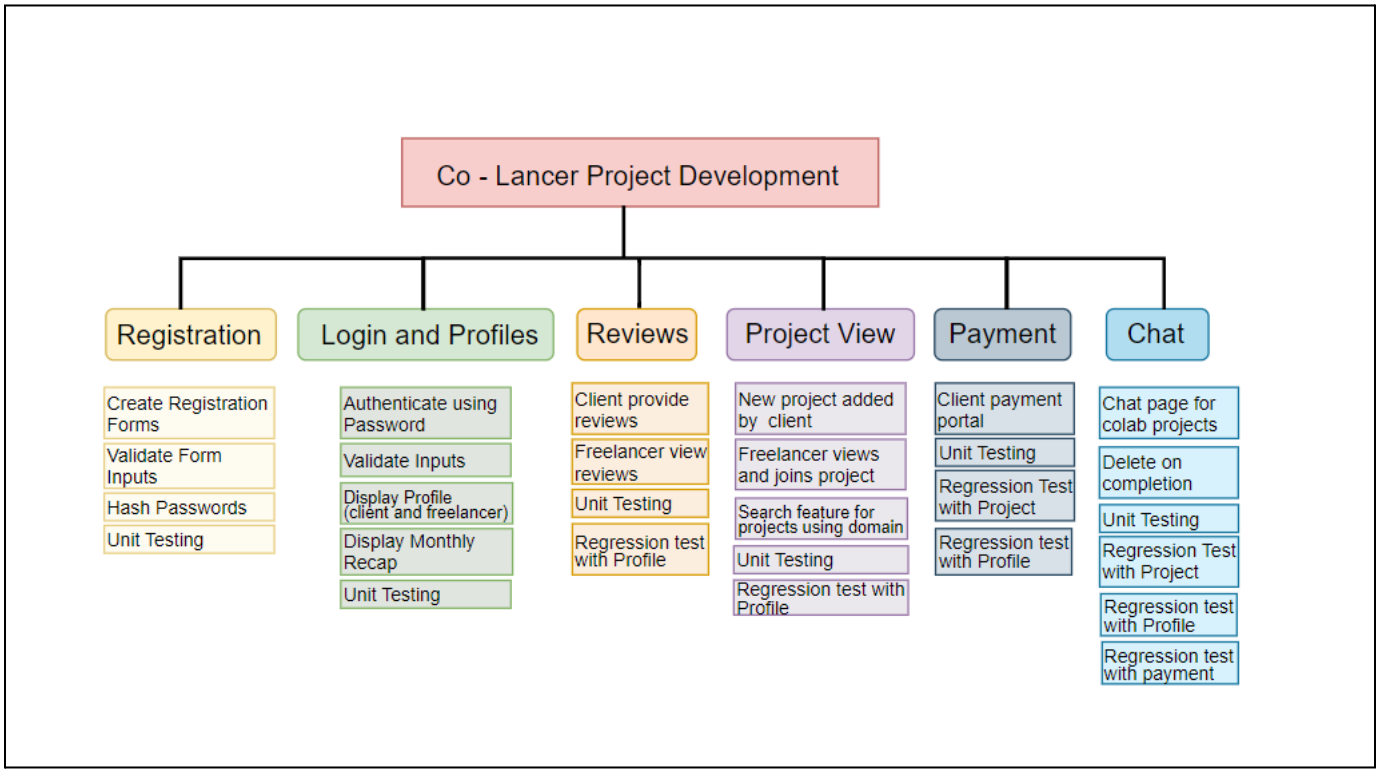
Reuse Components:

1. **User Login**
    - Categorising: Reuse component
    - Justification: The User Login component is used across various websites aiming to personalise their site.
2. **Chat Feature**
    - Categorising: Reuse component
    - Justification: A chat feature is an integral part of multi user websites that require user interactions.
3. **Review Feature**
    - Categorising: Reuse component
    - Justification: Review feature is commonly used across various projects that provide services or products.

Build Components:

1. **Profile Pages**
    - Categorising: Build Component
    - Justification: The profile pages may vary across websites according to the services provided.
2. **Monthly Recap**
    - Categorising: Build Component
    - Justification: A monthly recap feature is not frequently used and differs in terms of what is displayed and considered.
3. **Payment Feature**
    - Categorising: Build Component
    - Justification: Payment gateway is specific to the project and API used.
4. **Project Display**
    - Categorising: Build Component
    - Justification: This feature is unique to this project and requires customised components
5. **Project Creation**
    - Categorising: Build Component
    - Justification: Project creation page is limited to only a few applications. Hence this component requires additional features that have to be built.

WBS for functionalities



Rough Estimate of effort to accomplish each task in person months

1. **Initial Planning and SRS formulation**
   Participants: Pooja, Preethi
   Time allocated: 2 weeks
   Total: 1 Person Month

2. **Design and architecture**
   Participants: Pooja, Preethi
   Time allocated: 1 Week
   Total: 0.5 Person Month

3. **Database schema creation and access rights**
   Participants: Preethi
   Time allocated: 1 week
   Total: 0.25 Person Month

4. **Registration**
   Participants: Preethi
   Time allocated: 2 weeks
   Total: 0.5 Person Month

5. **Login and Profile**
   Participants: Preethi
   Time allocated: 3 weeks
   Total: 0.75 Person Months

6. **Project View with Search feature**
   Participants: Preethi
   Time allocated: 2 weeks
   Total: 0.5 Person Months

7. **Create Project**
   Participants: Preethi
   Time allocated: 2 weeks
   Total: 0.5 Person Months

8. **Monthly Recap**
   Participants: Pooja
   Time allocated: 1 week
   Total: 0.25 Person Months

9. **Reviews**
   Participants: Preethi
   Time allocated: 1 week
   Total: 0.25 Person Months

10. **Payment**
    Participants: Pooja
    Time allocated: 1 week
    Total: 0.25 Person Months

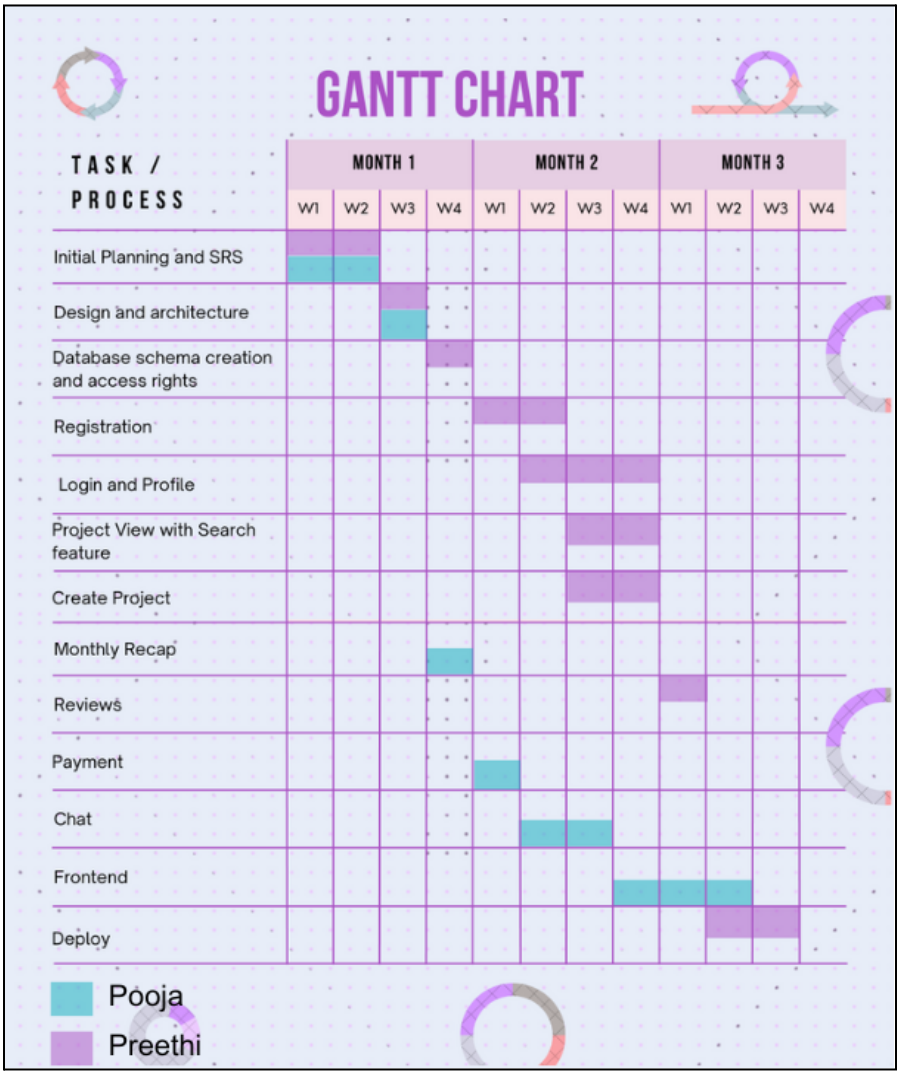
11. **Chat**
    Participants: Pooja
    Time allocated: 2 weeks
    Total: 0.5 Person Months

12. **Frontend**
    Participants: Pooja
    Time allocated: 3 weeks
    Total: 0.75 Person Months

13. **Deploy**
    Participants: Preethi
    Time allocated: 2 weeks
    Total: 0.25 Person Months

TOTAL ESTIMATE OF EFFORT: 6.25 Person Months

Gantt Chart


GANTT CHART
TASK / PROCESS
MONTH 1
MONTH 2
MONTH 3
W1
W2
W3
W4
Initial Planning and SRS
Design and architecture
Database schema creation and access rights
Registration
Login and Profile
Project View with Search feature
Create Project
Monthly Recap
Reviews
Payment
Chat
Frontend
Deploy
Pooja
Preethi