



UE20CS303 - Software Engineering
PROJECTPLANDOCUMENT
POS SYSTEM

Team Member 1	PES1UG21CS680	Team Member 3	PES2UG21CS621
Team Member 2	PES1UG21CS728	Team Member 4	PES1UG21CS685

Life-cycle followed

Identify the lifecycle to be followed for the execution of your project and provide justification for your choice of the model. Iterative Model the Iterative Model is a development approach where a project is divided into smaller iterations, each encompassing phases of requirements, design, implementation, testing, and deployment. This methodology offers several advantages, including the ability to manage complexity by breaking it into more manageable parts, early delivery of working functionality for feedback, adaptability to changing requirements, risk mitigation through incremental development, and clear progress tracking through iterative milestones. These characteristics make the Iterative Model particularly suitable for projects where flexibility and continuous improvement are crucial.

Tools Used for this Project

Identify the tools you want to use throughout the lifecycle, such as planning tools, design tools, version control, development tools, bug tracking, and testing tools.

- 1) Integrated Development Environment (IDE):
Visual Studio Code, Python IDLE
 - 2) Programming Languages:
HTML, CSS, PHP, NODE JS, REACT JS
 - 3) Database Management System (DBMS):
MYSQL OR MONGO DB
 - 4) Front-End Development:
HTML, CSS, REACT JS
 - 5) Back end:
PHP
 - 6) Project Management and Collaboration Tools:
Jira
 - 7) Testing and Quality Assurance:
Selenium and Jira
-

Build components

- 1) User Requirements Document (URD)
- 2) Documentation and User Manuals

Reusable Components:

System Architecture Design

Database Schema

User Interface (UI) Templates

Reporting Module

Testing Framework

Deployment Scripts

Work Breakdown Structure:

Create a Work Breakdown Structure (WBS) detailing all the functionalities.

- 1) Requirements gathering:
Collecting and documenting the project's functional and non- functional requirements.
- 2) Initiation of the project:
Kicking off the project, defining goals, and setting initial timelines.
- 3) Defining the scopes:
Clearly outlining the project's boundaries and deliverables. Project Manager
- 4) Team Roles (Developer, Designer and Tester):
Assigning roles and responsibilities to team member.
- 5) System design:
 - Wireframing Creating visual representations of the user interface for initial design.
 - Database schema Designing the structure and relationships within the project's database.
 - System architecture is based on client server architecture.
- 6) Implementation:
 - Backend development Building the server-side components and functionality.
 - Frontend development Creating the user-facing interface and client-side functionality.
- 7) Testing:
 - Testing by Conducting thorough testing of the software to identify defects and issues.
 - Fixing Bugs: Addressing and resolving identified bugs and issues.
- 8) Deployment:
 - Install the system on production servers
 - Configure the system for production use
 - Train users on how to use the system

Effort Estimation (in person-months)

Provide a rough effort estimate for each task in person-months

You may use the assumption that 1 year = 260 working days, then 1 month = $260/12 = 21.66$ working days. So 7 full working days for one person would be $7/21.66 = 0.323$ person-months.

We have 1 month for our Software engineering project that is 21.66 working days. Our team has 4 people so $5.415/21.66 = 1.292$ person months.

Gantt Chart

Create a Gantt Chart for scheduling using any scheduling tool

