PES UNIVERSITY
BENGALURU
Department of Computer Science and Engineering

LOST AND FOUND SYSTEM

Group Number: 13
Team Members:

SAMYUKTA B	PES2UG21CS470
SANJANA NAIK L	PES2UG21CS475
SARAYU THAMPAN	PES2UG21CS480
SHRAVYA N	PES2UG21CS496

Section: H

1. Model Used for Development

"Extreme Agile Programming"

1. Collective Code Ownership: In XP, any developer can modify any piece of code. This collective ownership promotes knowledge sharing, reduces bottlenecks, and ensures that the code quality remains consistent.
2. Frequent Feedback: The frequent feedback loop implemented in XP ensures that the product being developed closely aligns with customer needs and that any discrepancies are detected and rectified early on.
3. Iterative Development: The Lost and Found System can benefit from an iterative development approach, starting with important functionalities and gradually adding enhancements based on user feedback and changing needs.
4. Adaptability: XP allows our development team to adapt quickly to any unexpected challenges, changes in requirements or opportunities that may arise during the project.
5. Refactoring & Simplicity: Over time, as features are added and changes are made, code can become unnecessarily complex. Refactoring would involve simplifying this code, removing redundancies, making it more straightforward and easier to modify in the future.

2. Project Tools to be Used

Project Management and Planning Tools:

For efficient project planning, task management, and progress tracking, we will utilize specialized software tools such as Jira. Additionally, we will employ Lucidchart to create visual representations of our project workflows and diagrams.

Project Scheduling and Gantt Charts:

For project scheduling, we chose Asana's Timeline view, which provides us with a clear visual of our project phases, task durations, and dependencies.

Design Tools:

For wireframing and design purposes, we will employ Figma and Canva, industry-recognized tools known for their robust capabilities in creating visual design prototypes.

Version Control:

To maintain version control and streamline collaborative development efforts, we will utilize Git in conjunction with GitHub, ensuring the integrity and traceability of our codebase.

Development Tools:

The choice of our development tool is Visual Studio Code, selected for its versatility, compatibility with various programming languages and it offers a customizable development environment which aligns with our project's diverse needs.

Database Management:

To manage our database schema and data effectively, we will rely on tools like MySQL Workbench and Line Client. These tools are instrumental in ensuring the stability and performance of our database system.

Bug Tracking and Issue Management:

For meticulous tracking and management of bugs and feature requests, we will utilize industry-leading issue tracking systems such as Jira, Bugzilla, or Redmine.

Testing Frameworks:

To maintain code quality and reliability, we will employ reputable unit testing frameworks such as Pytest and Jasmine.

3. Deliverables

1. User Interface (UI):
Build Component: The user interface is a critical component of the system and should be custom-designed to meet the specific needs and branding of the organization or venue implementing the system.
2. Database:
Build Component: The database is a core component that stores information about lost items, including item descriptions, locations, and contact details. It needs to be designed and built according to the system's requirements.
3. User Registration and Authentication:
User registration and authentication components can often be reused to ensure security and user management.
4. Search:
Build Component: It is specific to the lost and found system's requirements, such as searching for items based on keywords, location, or category.
5. Notification System:
Custom notification logic may be needed to inform users when their lost items are found or when they find a matching item in the system.
6. Item Entry and Editing Forms:
Custom forms are usually required for users to enter details about lost items or edit their listings.
7. Item Management Dashboard:

A dashboard for administrators and staff to manage lost item listings, mark items as found.

8. Reporting:

Custom reporting and analytics tools may be needed to track system usage, success rates, and other relevant metrics.

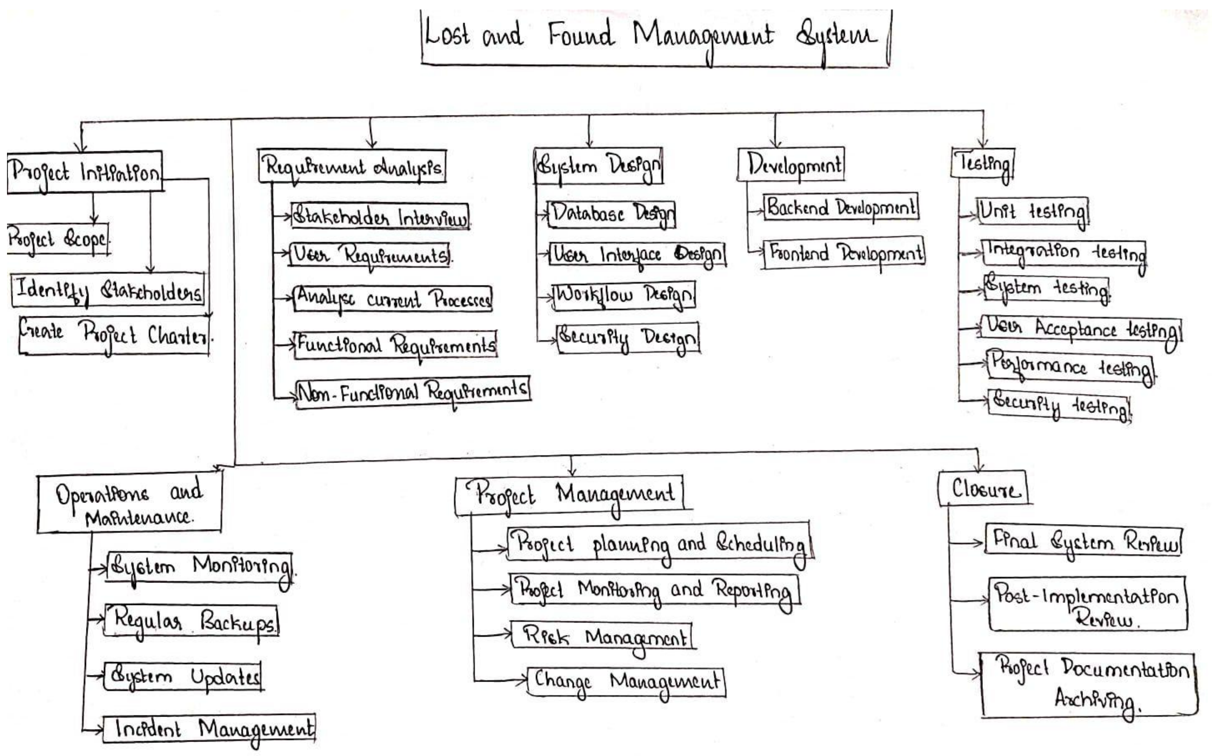
9. Documentation:

Comprehensive documentation is crucial for users and developers to understand how the system works, including user guides, and system architecture documentation.

10. Testing and Quality Assurance:

A dedicated testing and QA phase is necessary to ensure the system works correctly, is secure, and meets user expectations.

4.Work Breakdown Structure.



5. Effort Estimation

Our project, estimated at 7.6 KLOC (including git commands), is managed using the organic COCOMO model strategy, which is well-suited for our small-sized team.

a= 2.4, b=1.05, KLOC=7.6

Effort = $a(KLOC)^{b}$ = $2.4\ (7.6)^{1.05}$ = 20.18 PM $\approx 20\ PM$ (person months)

Task 1: Database implementation

KLOC: 2

Effort: $2.4(2)^{1.05} = 4.96 \approx 5PM$

Task 2: Web development

KLOC: 4

Effort: $2.4(4)^{1.05} = 10.289 \approx 10PM$

Task3: Matching lost and found items

KLOC: 0.5

Effort: $2.4(0.5)^{1.05} = 1.15 \approx 1PM$

Task 4: Interfacing website with database

KLOC: 0.9

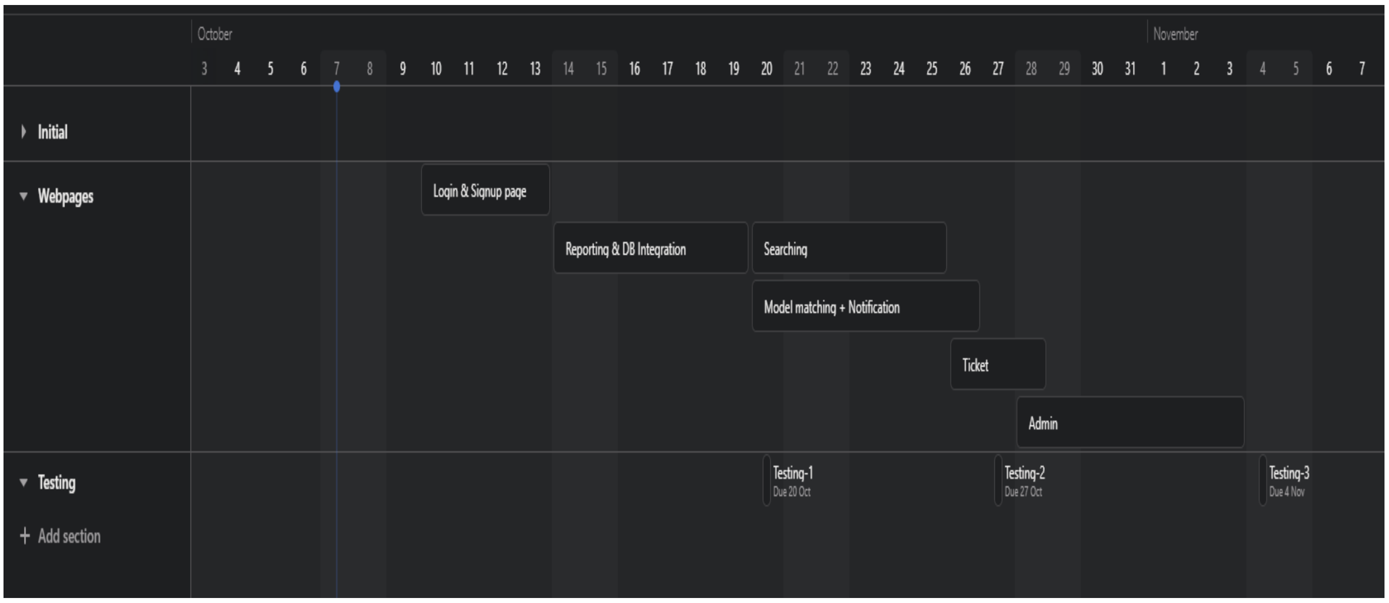
Effort: $2.4(0.9)^{1.05} = 2.14 \approx 2PM$

Task 5: Git and GitHub (These would be used alongside other tasks)

KLOC: 0.2

Effort: $2.4(0.2)^{1.05} = 0.44\ PM$

6. Gantt Chart


October
November
Initial
Webpages
Login & Signup page
Reporting & DB Integration
Searching
Model matching + Notification
Ticket
Admin
Testing
Testing-1
Due 20 Oct
Testing-2
Due 27 Oct
Testing-3
Due 4 Nov
Add section