

**Report of Project Management**

**Texmart : Inventory Management System**

**Course:** Project Management

**Course code:** CSE 469/PMG 4101

#### Section: B

**Course faculty:** Mr. Tahmid Mosaddeque

|  |  |
| --- | --- |
| Mahmud Hasan Sadi | 11181075 |
| Md. Abdul Haque Rahid | 11181142 |
| Nur Nobi Nazmul Islam | 11181032 |
| Sakhawat Khan Sourav | 11183087 |

#### INDEX

1. **Introduction ……………………………………..**
   1. Vision……………..
   2. Problem …………
   3. Solution …………
   4. Motivation ……….

#### Methodology ………………………………………

* 1. Waterfall
  2. Prototype
  3. Agile
  4. Scrum
  5. Spiral
  6. Why Choose Agile Methodology?
  7. The steps of developing a comprehensive stray pet adoption system

#### Work Breakdown Structure…………………………

* 1. What is WBS?
  2. What are the rules and principles of WBS?
  3. Three levels, Level 2 is the most important
  4. The assignment rule
  5. WBS of our project

#### Wideband Delphi …………………………

* 1. What is Wideband Delphi
  2. Delphi Wideband Process
  3. Assemble Table

#### Risk Management………………….

* 1. What is Risk Management?
  2. Risk Mitigation, Monitoring, and Management (RMMM)
  3. Risk Plan

### Introduction:

This project aims to develop a functional prototype of a web-based Inventory Management platform that track the inventory, stock products etc.

### Vision:

**What is Mission of a project?**

The mission of a project defines the purpose and scope of the project, outlining the specific activities, deliverables, and objectives that need to be accomplished to fulfill the project's vision.

##### Mission

These objectives provide a clear roadmap and milestones for the development of the lending and borrowing web application, ensuring that the project addresses the goals of user-friendliness, security, functionality, and sustainability.

1. Developing an user-friendly and intuitive web application interface that enriches the user experience and facilitates efficient to track the inventory.
2. LC will be the key.
3. There is two warehouse one for stock product and another for expiry product.
4. Develop a communication and notification system that enables seamless interaction between lenders and borrowers, facilitating coordination, inquiries, and updates on the status of items.
5. Design and implement a distance and range filtering system to search products where the user can put a distance and the platform will show all the available items from user’s path distance.
6. Implement a rating and feedback wall that allows users to provide reviews and ratings for their borrowing and lending experiences, collecting trust and credibility within the lending and borrowing community.
7. Establish a secure and efficient payment system that enables users to make transactions for rental fees or deposits reliably and conveniently.
8. Conduct user testing and evaluation to gather feedback on the usability, functionality, and overall user experience of the lending and borrowing platform, identifying areas for improvement and validating the effectiveness of the developed web application.
9. Assess the sustainability impact of the lending and borrowing platform by analyzing potential environmental benefits, such as resource utilization, waste reduction, and carbon footprint reduction, resulting from promoting a sharing economy.

### Scope

The scope of the application is to automate the followings

* Aging report
* Import process
* RM storing - LC is the key
* Issue of materials from store based on w/o
* Receiving the FG in FG W/H based on w/o
* Dispatching FG to buyer
* Submission of Doc.
* Getting acceptance
* Realization of money at the end of maturity

# Methodology

Methodology refers to a structured set of principles, practices, and procedures used to conduct activities systematically. It provides guidelines and frameworks for achieving goals, solving problems, or carrying out specific tasks efficiently and effectively. Methodologies help standardize processes, ensure consistency, and facilitate communication and collaboration among individuals or organizations. They serve as blueprints for planning, executing, monitoring, and evaluating activities across various domains.

### Waterfall

The Waterfall Model is a linear application development model that uses rigid phases: when one phase ends, the next begins. Steps occur in sequence, and, if unmodified, the model does not allow developers to go back to previous steps.

##### Advantages of the Classical Waterfall Model:

* Easy to Understand: The classical waterfall model is very simple and easy to understand.
* Individual Processing: Phases in the classical waterfall model are processed one at a time.
* Properly Defined: In the classical waterfall model, each stage in the model is clearly defined.
* Clear Milestones: The classical waterfall model has very clear and well-understood milestones.
* Properly Documented: Processes, actions, and results are very well documented.
* Reinforces Good Habits: The classical waterfall model reinforces good habits like define-before-design and design-before-code.
  + Working: The classical waterfall model works well for smaller projects and projects where requirements are well understood.

##### Disadvantages of the Classical Waterfall Model:

* + In the classical waterfall model, software progresses linearly, akin to a waterfall, assuming error-free development without feedback mechanisms for error correction across phases.
  + The waterfall model struggles with change as it assumes fixed initial requirements, while evolving needs hinder integrating changes after the specification phase.
  + In contrast to the waterfall model's sequential phases, real projects often require overlapping phases to enhance efficiency and reduce costs.
* The Waterfall Model's rigid, linear approach to software development makes it unsuitable for projects with evolving or uncertain requirements, as it makes adjustments challenging and hinders backtrack ability.
  + The Waterfall Model is a structured, sequential approach that restricts stakeholder involvement in early phases like requirements gathering and analysis, reducing participation in later stages.

## Agile

Agile methodology is arguably one of the most popular software development methodologies in recent days. It takes a different approach from the conventional, linear method. Agile focuses on how to satisfy the users instead of emphasizing documentation and rigid procedures.

## Scrum

Scrum is an agile team collaboration framework commonly used in software development and other industries. Scrum prescribes for teams to break work into goals to be completed within time-boxed iterations, called sprints. Each sprint is no longer than one month and commonly lasts two weeks.

**Externally, the main advantages of adopting the Scrum methodology are:**

* + Scrum's adaptability and flexibility make it suitable for diverse environments and situations lacking clearly defined requirements, necessitating a flexible approach.
  + Scrum fosters creativity by promoting collaboration among team members, facilitating the analysis of various ideas, and encouraging the emergence of new creative concepts.
* It involves low costs: Adopting the Scrum approach can be cost-effective for an organization, as it usually requires less documentation and control.
  + Scrum's emphasis on team-wide responsibility and ownership often results in a productive environment conducive to high-quality work outcomes.
  + Scrum's focus on maximizing team performance and adapting to feedback loops often enhances customer satisfaction by delivering products and solutions that resonate well with their needs and preferences.
* The Scrum framework, by promoting individual responsibility and ownership, generally leads to higher employee satisfaction as team members are more motivated and engaged in their work.

**Some of the main disadvantages of using the Scrum framework are:**

* The Scrum framework's successful implementation relies on a skilled team that understands its benefits and complexities, necessitating comprehensive pre-adoption training.
  + Scaling Scrum for large projects requires extensive training and coordination, and complex implementation processes, despite available methods designed for such purposes.
* The Scrum framework necessitates significant organizational transformations, requiring effective collaboration among departments and the company's management and organization to ensure successful implementation.
  + Hybrid solutions can combine Scrum's long-term planning benefits with the flexibility of the Scrum framework, despite challenges in aligning with projects requiring predictability and a defined plan.
* Scrum methodology sets smaller deadlines for team productivity, but it doesn't address the project's overall deadline, so project managers and stakeholders must ensure project completion.
* Scrum methodology thrives in small teams with three to ten members, promoting collaboration and teamwork, but some organizations may face challenges in reallocating their workforce.
* The Scrum methodology requires experienced personnel to efficiently perform tasks within intense work cycles, providing expert input and informed feedback on outcomes and the process.

## Spiral

The spiral model is a systems development lifecycle (SDLC) method used for risk management that combines the iterative development process model with elements of the waterfall model. The spiral model is used by software engineers and is favored for large, expensive, and complicated projects.

**Model Advantages of the Spiral Model:**

* Software is produced early in the software life cycle.
* Risk handling is one of the important advantages of the spiral model, It is the best development model to follow due to the risk analysis and risk handling at every phase.
  + Flexibility in requirements. In this model, we can easily change requirements at later phases and incorporate them accurately. Also, additional functionality can be added at a later date.
* It is good for large and complex projects.
* Encouraging customer involvement in software development and production early in the software life cycle can enhance customer satisfaction.
* Strong approval and documentation control.
* It is suitable for high risk projects, where business needs may be unstable. A highly customized product can be developed using this.

**Disadvantages of the Spiral Model:**

* It is not suitable for small projects because it is expensive.
* It is much more complex than other SDLC models. the process is complex.
* Too much relies on risk analysis and requires highly specific expertise.
* Difficulty in time management. As the number of phases is unknown at the start of the project, time estimation is very difficult.
* spiral may go on indefinitely, the end of the project may not be known early.
* It is not suitable for low risk projects.
* It may be hard to define objective, verifiable milestones. Large numbers of intermediate stages require excessive documentation.

**Extreme Programming (XP):**

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software and a higher quality of life for the development team.

XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

**Advantages of XP Programming:**

* Extreme Programming is a cost-effective approach that prioritizes timely product delivery, minimizes documentation, and encourages team discussions for problem-solving in software development companies.
* Simplicity is one more advantage of Extreme Programming projects. The developers who prefer to use this methodology create extremely simple code that can be improved at any moment.
* Constant feedback is also a strong side. It is necessary to listen and make any changes needed in time.
* XP helps create software faster thanks to regular testing at the development stage.
* Extreme programming contributes to increasing employee satisfaction and retention.

**Disadvantages of XP:**

* Extreme programming may prioritize code over design, potentially causing issues in software applications and potentially causing future bugs due to poor defect documentation in XP projects.
* One more disadvantage of XP is that this methodology does not measure code quality assurance. It may cause defects in the initial code.
* XP is not the best option if programmers are separated geographically.

**Rational Unified Process (RUP):**

Rational Unified Process (RUP) is an agile software development method, in which the life cycle of a project, or the development of software, is divided into four phases. Various activities take place during these phases: modeling, analysis and design, implementation, testing, and application.

##### RUP can provide software development or design teams with an array of advantages, including:

* Offering thorough documentation: The RUP process involves carefully documenting each step, which can be highly beneficial for collaborative projects.
* Enhancing risk management practices: RUP can help software individuals proactively respond to potential software challenges. This can improve risk management and troubleshooting efforts.
* Consistent feedback to project stakeholders, including software individuals, company leaders, and vendors, is crucial for the successful execution of the RUP process.
* Reducing total project time: RUP may allow the software development team to lower their time in both the development and integration stages.
* RUP allows stakeholders to identify potential software issues early in the project, enabling easier mitigation or resolution of challenges before they become more complex.

**Potential drawbacks of RUP:**

* + RUP is a complex procedure requiring expertise from software team members, making it easier to choose a different development process for newer team members.
* RUP documentation can be time-consuming and costly, making it beneficial for software teams with smaller budgets to opt for a cost-efficient approach.
  + RUP may cause confusion during testing for larger projects with multiple components and software teams, prompting users to slow down the process or explore alternative development procedures.

## Incremental

The incremental model in software engineering is a process where the system is developed and delivered in small parts or increments. Each increment adds some functionality to the previous one, until the complete system is achieved. The increments are usually developed using an iterative waterfall model, which means they go through the phases of requirements, design, implementation, and testing.

##### Some advantages of the incremental model are:

* + It allows early feedback from the customer and users, which can improve the quality and satisfaction of the system.
  + It reduces the risk of failure, as each increment can be tested and verified before moving on to the next one.
* It provides flexibility and adaptability, as the requirements and design can be modified based on feedback and changes in the environment.
* It lowers the initial delivery cost, as the customer can start using the system with the core features and pay for the additional ones later.

##### Some disadvantages of the incremental model are:

* + It requires good planning and design, as the system needs to be divided into coherent and independent increments that can be integrated smoothly.
  + It increases the total cost, as each increment may need rework and maintenance to ensure compatibility and consistency with the other increments.
  + It needs well-defined module interfaces, as communication and coordination between the increments are crucial for the system's functionality and performance.

### Methodology

##### Development Approach

The web application will be developed by following an agile methodology, specifically the Scrum framework. This approach allowed for iterative development, continuous feedback, and flexibility in responding to changing requirements.

##### Requirements Gathering

To gather requirements for the web application, we conducted user interviews and adminis- tered surveys to potential users. The interviews provided valuable insights into the desired features and functionalities, while the surveys helped in understanding user preferences and expectations.

##### System Design

The system architecture and overall structure of the web application will be designed using the Model-View-Controller (MVC) design pattern. This separation of concerns allowed for modular development and easier maintenance of the code base. We utilized the usage of UML diagrams, such as class diagrams, the sequence diagrams, to create a mind map, visualize and flow of the design.

##### Technology Stack

The web application will be built using the following technologies:

* 1. Programming Languages: HTML, Bootstrap CSS, JavaScript
  2. Front end Frameworks: React.js
  3. Back end Frameworks: Laravel
  4. Database: MySql
  5. Version Control: Git

These technologies have been chosen based on their compatibility, community support, and scalability.

##### Development Process

The development process will follow an iterative approach with regular sprints. We will utilized project management tools like Jira for task management, Slack for communication, and GitHub for version control. Continuous integration and deployment (CI/CD) pipelines is setting up to ensure smooth development and deployment of the web application.

##### Testing and Quality Assurance

For the unit testing we will be using the popular Jest which is pre-configured framework for testing react based apps. Another popular solution is React Testing Library. In component testing we are going to use the Enzyme which helps to testing utilities for shallow rendering, mounting, and manipulating React components.

##### Data Collection

As part of the web application, user feedback and usage data were collected. To protect user privacy and ensure data security, all data collection processes adhered to relevant regulations and best practices. The data collected were anonymized and stored securely in a separate database.

##### Evaluation

Evaluation metrics met user satisfaction, , security assessments, system performance and adherence to regulatory standards. The platform demonstrated high levels of user satisfaction and engagement, robust security features, and compliance with relevant legal and regulatory frameworks. Performance testing indicated the platform’s scalability to accommodate a growing user base.

##### Ethical Considerations

Throughout the development process, we considered ethical considerations related to user privacy, data protection, and informed consent. Measures

were implemented to obtain informed consent from users, protect personal information, and adhere to applicable data protectionregulations.

# Use Case

A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request.

**Actor:** In a use-case diagram, users of the system are depicted as actors. Each actor performs a specific role in the system, which is called a use-case. More than one actor can perform a single use-case. An actor can be a person, such as a customer, or a computer, such as a database system or server.

|  |  |
| --- | --- |
| Name | UC- 03:Stock a new product to the warehouse. |
| Summary | This use case describes the process of stock a product from LC to Inventory. |
| User | The primary user of this use case is the Employee, who seeks to stock item from LC to Inventory |
| Precondition | 1. The employee must have a registered account on the IMS. 2. The employee must have listed the item they wish to stock, including necessary details such as availability and terms. |

|  |  |
| --- | --- |
| Basic Course of Events | 1. Login: The employee logs into their account on the stock item platfom. 2. Search for Item: The borrower navigates to the search functionality and enters keywords or selects categories to find the desired item. 3. View Item Details: The borrower reviews the item details, including description, availability, and terms set by the lender. 4. Initiate Borrow Request: If the item meets the borrower's requirements, they initiate a borrow request. 5. Confirmation: The lender receives the borrow request notification and reviews the borrower's profile and request details. 6. Accept/Reject Request: The lender accepts or rejects the borrow request based on their discretion. 7. Arrange Pickup/Delivery: Upon request approval, the borrower and lender coordinate the pickup or delivery logistics. 8. Borrowing Period: The borrower utilizes the item within the agreed-upon timeframe. 9. Return Item: At the end of the borrowing period, the borrower returns the item to the lender. |

|  |  |
| --- | --- |
|  |  |
| Alternation Path | If the item is unavailable or the lender rejects the borrow request, the borrower may search for alternative items or adjust their borrowing criteria. |
| Postcondition | 1. The borrower successfully borrows the item, and the lender receives it back in the agreed condition. 2. Feedback and ratings contribute to building trust within the lending and borrowing community. 3. Transaction details are recorded for future reference and analysis by the lending and borrowing platform. |

|  |  |
| --- | --- |
| Name | Use Case 04: Conferred as Rewarder |
| Summary | This use case outlines the process of conferring rewards to users who actively participate and contribute positively to the lending and borrowing platform. It involves recognizing and rewarding users for their engagement, reliability, and positive interactions within the community. |
| Rationale | Rewarding users encourages active participation, fosters a sense of community, and incentivizes desirable behavior such as timely returns, providing helpful feedback, and maintaining item quality. This use case aims to enhance user engagement, trust, and overall satisfaction with the platform. |
| User | The primary user of this use case is the Rewarder, typically an administrator or moderator responsible for managing and overseeing community engagement on the lending and borrowing platform. |
| Precondition | 1. The lending and borrowing platform must have a system in place for tracking user activity, contributions, and interactions. 2. The Rewarder must have the necessary permissions and access to confer rewards to users. |

|  |  |
| --- | --- |
| Basic  Course of Events | 1. Identify Deserving Users: The Rewarder identifies users who have demonstrated positive behavior, such as timely returns, providing helpful feedback, or actively engaging with the community. 2. Select Reward Criteria: The Rewarder establishes criteria or guidelines for conferring rewards, considering factors such as frequency of participation, helpfulness, reliability, and adherence to platform guidelines. |
|  | 3. Review User Contributions: The Rewarder reviews the user's activity, contributions, and interactions to ensure they meet the established reward criteria. |
|  | 4. Confer Reward: If the user meets the criteria, the Rewarder confers a reward, which may include virtual badges, points, discounts, or special privileges within the platform. |

|  |  |
| --- | --- |
|  | 1. Notify User: The user is notified of the reward conferred, along with a message expressing appreciation for their positive contributions to the platform. 2. Update User Profile: The user's profile is updated to reflect the conferred reward, allowing other users to view their achievements and contributions. |
| Alternation Path | If a user is not eligible for a reward, the Rewarder may provide constructive feedback or guidance on how they can improve their participation and contributions to become eligible in the future. |
| Post conditions | 1. The deserving user receives recognition and reward for their positive contributions to the lending and borrowing platform. 2. User engagement and satisfaction are enhanced, leading to a stronger sense of community and trust among users. 3. The lending and borrowing platform maintains a positive   environment conducive to sustainable consumption and collaborative sharing. |

##### Functional and Nonfunctional Requirements

**Functional Requirements** describe the expected functionality of the system: what the system should do. They are usually more explicit and can be directly linked to specific user stories or use cases.

**Non-Functional Requirements** define how the system should behave. They are more about the system’s qualities or attributes rather than its functionality.

##### Functional Requirements

1. **User Registration and Authentication**:
   1. Users can register on the platform by providing necessary details.
   2. Users can log in to the platform using credentials.
2. **Profile Management**:
   1. Users can view and edit their profile details.
   2. Users can view their stock item history.
3. **Listing Items for store**:
   1. Employee can list items they want to lend.
   2. Lenders can specify the duration for which the item is available for lending.
4. **Searching and Borrowing**:
   1. Borrowers can search for items.
   2. Borrowers can send requests to lenders to borrow an item.
   3. Lenders can approve or decline borrowing requests.
5. **Notifications and Messaging**:
   1. Users receive notifications for borrowing requests, approvals, etc.
6. **Feedback and Ratings**:
   1. Borrowers can leave feedback and ratings for lenders and vice versa.
7. **Payment and Security Deposit Management**:
   1. Borrowers can make online payments for borrowed items.
   2. Security deposits can be held and refunded upon return of the item.

##### Non-Functional Requirements

1. **Performance**:
   1. The platform should be able to handle a high number of concurrent users.
   2. Search results should be displayed in less than 2 seconds.
2. **Usability**:
   1. The interface should be user-friendly and intuitive.
   2. The platform should be accessible on multiple devices (responsive design).
3. **Security**:
4. User data should be encrypted and stored securely.
5. Financial transactions should be secured using encryption standards.
6. The platform should have measures to prevent SQL injection, XSS, CSRF, and other common web vulnerabilities.
7. **Reliability**:
   1. The platform should have an uptime of 99.9%.
   2. Regular backups of the database should be taken.
8. **Maintainability**:
9. The code base should be modular and well-documented.
10. Regular updates and bug fixes should be rolled out without affecting user experience.
11. **Scalability**:
12. The platform should be scalable to accommodate an increasing number of users.
13. It should be possible to add new features without major changes to the existing system.

### Work Breakdown Structure

A Work Breakdown Structure (WBS) is a project management tool that breaks down a project into smaller components.

1. **What is a Work Breakdown Structure (WBS)?**

* A WBS is a visual, hierarchical, and deliverable-oriented deconstruction of a project.
  + It allows project managers to break down their project scope and visualize all the tasks required to complete their projects.
  + The WBS outlines project work steps, and each level subdivides the project scope into tasks, deliverables, and work packages.

1. **Why use a WBS in project management?**

16

* + **Project Planning:** The WBS is the first step in developing a project schedule. It defines all the work that needs to be completed (and in what order) to achieve the project goals and objectives.
  + **Resource Allocation:** By visualizing your project in this manner, you can allocate resources effectively for all your project tasks.
  + **Risk Management:** A well-constructed WBS helps manage risks by identifying critical components.
* **Avoiding Common Issues:** It helps prevent missed deadlines, scope creep, and cost overruns.

1. **Benefits of Using a WBS:**

* **Clear Framework:** Provides a framework for project planning, execution, and control.
* **Cost Estimation**: Helps estimate costs for each component.
* **Task Scheduling:** organizes tasks in a logical order.
* **Resource Allocation:** Allocates resources efficiently.
* **Keeps Projects on Track:** Ensures alignment with project goals.

**Work Breakdown Principles**

##### 100% Rules

The 100% rule is a fundamental principle in Work Breakdown Structure (WBS) development, which states that the WBS should include all the work necessary to complete the project scope, and it should encompass 100% of the project scope. In other words, the WBS captures all deliverables, including internal, external, and interim ones, in terms of the work to be completed. This ensures that nothing is left out and that the entire project scope is accounted for.

1. **Planned Outcomes, Not Planned Actions**

##### Level 2 is the Most Important

Level 2 represents the second tier in the WBS hierarchy. It is a critical level because it provides a balance between a high-level overview and a detailed breakdown. At this level, major components or sub-projects are further subdivided into manageable work packages.

1. **The Four Elements in Each WBS Element**

The four elements in each Work Breakdown Structure (WBS) element typically include:

* 1. **Description:** A brief description or title that identifies the specific task or deliverable.
  2. **Work Package**: This outlines the work to be completed, typically at a level where it can be easily managed and tracked.
  3. **Assigned Resources**: This specifies the individuals or teams responsible for completing the work package.
  4. **Timeline**: An estimated duration or timeframe for completing the work package,

17

providing a schedule for the project's activities.

These elements help organize and define the scope of the project, allowing for better planning, execution, and monitoring.

1. **Mutually-exclusive Elements**

Mutually-exclusive elements are distinct components within a system that do not overlap, ensuring clear categorization and avoiding ambiguity in classification.

1. **How Far Down?**

##### The 40-Hour Rule of Decomposition

The 40-Hour Rule of Decomposition is a guideline suggesting that a task or work package should ideally be decomposed into smaller subtasks or components that can be completed within 40 hours or less, facilitating better management and tracking of progress.

* **The 4% Rule of Decomposition**

The 4% Rule of Decomposition is a guideline advising that tasks or work packages should ideally be decomposed into smaller subtasks or components that can be completed within 4% of the total project duration, aiding in effective project management and progress tracking.

1. **WBS Identification Numbering**
2. **Set co-ordinate:**

**\*\*Scope of Work (Deliverables):\*\***

* 1. Develop a user-friendly web interface for the lending and borrowing platform.
  2. Implement features to set and display coordinates for item locations.
  3. Ensure prominently featured items on the homepage.
  4. Create intuitive browsing functionality for users to easily find items based on coordinates.

**\*\*Beginning and End Dates:\*\***

* Beginning Date: January 17, 2024
* End Date: January 19, 2024

**\*\*Budget for the Scope of Work:\*\***

* Total Budget: 1000$

Person Responsible: Shahriar Hossain

**Search item:**

##### \*\*Scope of Work (Deliverables):\*\*

1. Develop a search functionality allowing users to find items based on categories, keywords, or locations.
2. Implement filters to refine search results, including distance and range filtering.
3. Ensure search results display relevant and accurate information about available items.
4. Create an intuitive user interface for the search feature, ensuring ease of use and navigation.

**\*\*Beginning and End Dates:\*\***

* Beginning Date: January 10, 2024 - End Date: January 12, 2024

**\*\*Budget for the Scope of Work:\*\***

- Total Budget:

1200$

Responsib le: Dipa Akter

**Live chat:**

##### \*\*Scope of Work (Deliverables):\*\*

1. Develop a live chat feature for real-time communication between users.
2. Implement user-friendly interface for seamless interaction within the live chat.
3. Ensure integration of chat functionality across all platform pages.
4. Provide customization options for users to personalize their chat experience.
5. Include features such as emoji support, file sharing, and message formatting.

**\*\*Beginning and End Dates:\*\***

* Beginning Date: January 11, 2024
* End Date: January 13, 2024

**\*\*Budget for the Scope of Work:\*\***

* Total Budget:1300$

Person Responsible: Sadman Sakib

**Borrow item:**

##### \*\*Scope of Work (Deliverables):\*\*

1. Develop a user-friendly interface for borrowing items on the lending and borrowing platform.
2. Implement features to browse and search for available items based on categories, keywords, and locations.
3. Ensure prominently featured items on the borrowing section of the platform.
4. Create intuitive functionality for users to request to borrow items and coordinate with lenders.
5. Develop a notification system to facilitate communication between borrowers and lenders regarding item availability, pickup/delivery arrangements, and return schedules.

**\*\*Beginning and End Dates:\*\***

* + Beginning Date January 03,2024
  + End Date: January 05,2024

**\*\*Budget for the Scope of Work:\*\***

- Total Budget: [Insert Total Budget Amount] Person Responsible: MD. Easin

**Add to cart system:**

##### \*\*Scope of Work (Deliverables):\*\*

1. Implement an "Add to Cart" feature for the lending and borrowing platform.
2. Develop functionality to add multiple items to the cart from different categories.
3. Ensure seamless integration of the "Add to Cart" feature into the user interface.
4. Enable users to view and manage items in their cart easily.
5. Implement validation checks to prevent adding unavailable items to the cart.

* Beginning Date: January 03,2024
* End Date: January 05,2024

**\*\*Budget for the Scope of Work:\*\***

Total Budget: 1000$

20

Person Responsible: MD. Easin

**Deposit money:**

##### \*\*Scope of Work (Deliverables):\*\*

1. Develop a secure deposit system for the lending and borrowing platform.
2. Implement user-friendly interface for depositing money into user accounts.
3. Ensure encryption and protection of users' financial information.
4. Enable users to view their deposit balance and transaction history.

**\*\*Beginning and End Dates:\*\***

* Beginning Date: [Insert Beginning Date]
* End Date: [Insert End Date]

**\*\*Budget for the Scope of Work:\*\***

* Total Budget: - Beginning Date January 03,2024
* End Date: January 05,2024

**\*\*Budget for the Scope of Work:\*\***

Total Budget: 1000$

Person Responsible: MD. Easin

**Receive reward money:**

##### \*\*Scope of Work (Deliverables):\*\*

1. Design and develop a feature for users to receive reward money.
2. Implement a secure transaction system for transferring reward money to users' accounts.
3. Create a user interface for users to view and manage their reward money.
4. Ensure compatibility with various payment methods for receiving reward money.

* Beginning Date January 03,2024
* End Date: January 05,2024

**\*\*Budget for the Scope of Work:\*\***

Total Budget: 1000$

Person Responsible: MD. Easin

**Admin-à**

##### Remove post

Scope of Work: Remove post feature from the lending and borrowing platform. Beginning Date: February 20, 2024

End Date: March 5, 2024 Budget:

$1000

Person Responsible: Shahriar Hossain

* 1. **Approve post**

Scope of Work:

* + 1. Review the proposed post feature implementation on the lending and borrowing platform.
    2. Test the functionality and user experience of the post feature.
    3. Approve the post feature for deployment if it meets all requirements and standards. Beginning Date: March 10, 2024 End Date: March

25, 2024 Budget:

$1500 Person Responsible:

Dipa Akter

* 1. **Troubleshoot for issue**

Scope of Work:

* + 1. Identify the root cause of the issue affecting the lending and borrowing platform.
    2. Implement necessary fixes and updates to resolve the issue.
    3. Conduct thorough testing to ensure the issue is completely resolved and no new issues arise. Beginning Date: March 28, 2024

End Date: April 11, 2024 Budget:

$2500

Person Responsible: Sadman Sakib

Scope of Work:

* + - 1. Develop a feature for replaying chat conversations within the lending and borrowing platform.

22

* + - 1. Implement functionality to navigate through chat history and replay specific messages.
      2. Ensure compatibility and seamless integration with existing chat interface. Beginning Date: June 1, 2024 End Date: June

15, 2024 Budget:

$2000

Person Responsible: Mahmudur Rahman Tushar

Scope of Work:

* + - * 1. Design the user interface for uploading items within the chat feature of the lending and borrowing platform.
        2. Develop backend functionality to support uploading images and descriptions of items. 3.Implement security measures to ensure safe and secure item uploads.

Beginning Date: September 10, 2024 End Date:

September 24,

2024 Budget:

$2500

Person Responsible: Lutfar Rahman Khan Hafez

### Delphi Wideband Method

The Wideband Delphi estimation process is a crucial tool for project managers as it yields several vital components of the project plan. Foremost among these is the collection of estimates that serve as the foundation for constructing the project schedule. Additionally, the project team collaborates to develop a work breakdown structure (WBS), an essential aspect of the plan's framework. Furthermore, the team compiles a list of assumptions, which can supplement the vision and scope document.

The discussions held among team members during both the kickoff meeting and the estimation session are invaluable outcomes of the Delphi process. These conversations often unveil previously overlooked project priorities, assumptions, and tasks, enhancing the overall understanding of the project's scope and requirements.

Upon completing the Wideband Delphi process, the team emerges with a heightened familiarity with the upcoming work, setting a solid groundwork for successful project execution.

#### Wideband Delphi Script:

|  |  |
| --- | --- |
| **Name** | **Wideband Delphi script** |
| **Purpose** | A project team generates estimates and a work breakdown structure. |
| **Summary** | A repeatable process for estimation. Using it, the project team can generate a consensus on estimates for the scope of the project. |
| **Work Product** | * **Input:** project requirements document, or other documentation that defines the scope of the newspaper website. * **Output:** Work breakdown structure (WBS), effort estimates for each of the tasks in the WBS. |

|  |  |
| --- | --- |
| **Entry Criteria** | * The project requirements document has been agreed to by stakeholders (e.g., project manager, engineering team). * The kickoff meeting date has been scheduled by stakeholder agreement. * The project manager and moderator agree on the goal of estimation session(s), methodology defining the scope of the WBS at each session. |
| **Basic course of**  **event** | * **Choosing the team:** The project manager selects the estimation team and moderator. The team should consist of 3 to 7 project team members. The team should include representatives from every engineering group that will be involved in the development of the newspaper website. * **Kickoff meeting:** The moderator prepares the team and leads a discussion to brainstorm assumptions, generate WBS, etc. * **Individual Preparation:** Each team member independently estimates the effort required for each task in the WBS. * **Estimation Session:** The team discusses their estimates, with the moderator facilitating the discussion to ensure it stays constructive and on-topic. * **Assemble Task:** The project manager works with the team to collect all the estimations from the team members at the end of the meeting and compile the final task list , estimates, and assumptions. * **Reviewing results:** The project manager reviews the estimation with the estimation team. |
| **Alternative Path** | **Step 1:** If the team determines that not enough information is known about the project to perform the estimations , the script ends. Before the script can begin again, the project manager must document the missing information by creating or modifying the vision or scope of the project.  **Step 2:** If the team determines that there are outstanding issues, the script ends. |
| **Exit Criteria** | The script ends after the team has either generated a set of estimates or has agreed upon a plan to resolve the outstanding issues. |

##### Estimating the team:

**Project Manager:** Mohammed Shahariar Hossain

**Project Moderator:** Sadman Sakib Pias

**Development Team:** Lutfar Rahman Khan Hafez,Dipa Akter,Mahmudur Rahman Tushar,Md.Easin.

### Individual Estimation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name:** Mahmudur Rahman Tushar **Date:20-3-23** | | | **Estimation Form** | |
| **Goal Statements:** To estimate the time to develop a prototype for a Lend Borrow WebPlatform**.** | | | **Unit: Days** | |
| **Category Goal Task** | | | **Quality Task** | |
| **Name of the WBS or Priority** | **Task Name** | **Estimation** | | **Assumptions** |
| **1.** | **Develop Home page** | **8** | |  |
| **2.** | **Implement a secure authentication system** | **7** | |  |
| **3.** | **Implement a comprehensive item listing** | **10** | |  |
| **4.** | **Develop a communication and notification system** | **12** | |  |
| **5.** | **Design and implement a distance filtering system** | **16** | | **Integration with third-party services** |
| **6.** | **Implement a rating and feedback system** | **9** | |  |
| **7.** | **Establish a secure payment system** | **11** | |  |
| **Total:** | | **73** | | |

#### Estimation Session:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name:** Mahmudur Rahman Tushar **Date:20-3-23 Estimation Form** | | | | | | | |
| **Goal Statements:** To estimate the time to develop a prototype **Unit: Days**  for a Lend Borrow web platform**.** | | | | | | | |
| **Category Goal Task Quality Task** | | | | | | | |
| **Name of the WBS or Priority** | **Task Name** | **Est.** | **Delta- 1** | **Delta- 2** | **Delta-3** | **Total** | **Assumption** |
| **1.** | **Develop Home page** | **8** | **+2** | **-1** | **+2** | **11** |  |
| **2.** | **Implement a secure authentication system** | **7** | **+3** | **+4** | **+1** | **15** | **Including servers, databases, and development tools.** |
| **3.** | **Implement a comprehensive item listing** | **10** | **+1** | **+3** | **+3** | **17** |  |
| **4.** | **Develop a communication and notification system** | **12** | **+4** | **+3** | **-1** | **18** |  |
| **5.** | **Design and implement a distance filtering system** | **16** | **+5** | **+2** | **-3** | **20** | **Integration with third-party services** |
| **6.** | **Implement a rating and feedback system** | **9** | **+1** | **+4** | **+2** | **16** |  |
| **7.** | **Establish a secure payment system** | **11** | **+4** | **-1** | **+1** | **15** |  |
|  | **Delta** | **73** | **20** | **14** | **5** |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total:** | **93** | **107** | **112** |  |

**Assemble Tasks:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Goal Statements:** To estimate the time to develop a prototype for a Lend Borrow WebPlatform**.** | | | | | | | | | |
| **Estimators:**Mahmudur Rahman Tushar,Md.Easin,Dipa Akter,Lutfar **Unit: Days**  Rahman Khan Hafez,Sadman Sakib Pias. | | | | | | | | | |
| **Category Goal Task Quality Task** | | | | | | | | | |
| **Name of the WBS**  **or Priorit y** | **Task Name** | **MRT** | **ME** | **DA** | **LRK** | **SSP** | **Worst Case** | **Best Case** | **Avg. Case** |
| **1.** | **Develop Home page** | **8** | **9** | **7** | **10** | **11** | **11** | **7** | **9** |
| **2.** | **Implement a secure authentication system** | **7** | **6** | **8** | **9** | **8** | **9** | **6** | **7.5** |
| **3.** | **Implement a comprehensive item listing** | **10** | **11** | **9** | **13** | **12** | **13** | **9** | **11** |
| **4.** | **Develop a communication and notification system** | **12** | **10** | **9** | **11** | **8** | **12** | **8** | **10** |
| **5.** | **Design and implement a distance filtering system** | **16** | **15** | **16** | **14** | **13** | **16** | **13** | **22.5** |
| **6.** | **Implement a rating and feedback system** | **9** | **8** | **7** | **10** | **11** | **11** | **7** | **9** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **7.** | **Establish a secure payment system** | **11** | **12** | **11** | **9** | **10** | **12** | **9** | **10.5** |
| **Total:** | | **73** | **71** | **67** | **76** | **73** | **84** | **59** | **79.5** |

##### Alternative Path:

An alternative path for the project could involve prioritizing environmental sustainability measures in the early stages of development. This approach would emphasize integrating features such as carbon footprint tracking, eco-friendly product recommendations, and incentivized usage for sustainable practices. By placing a stronger emphasis on environmental impact assessment and eco-conscious user engagement strategies, the project aims to foster a community deeply committed to sustainable consumption. This alternative path ensures that the lending and borrowing platform not only facilitates resource sharing but also actively contributes to mitigating environmental challenges associated with over-consumption and waste generation. **Exit Criteria:**

Upon project completion, success will be determined by meeting the following exit criteria:

The web application must be fully operational, fulfilling all outlined objectives. User testing feedback should reflect exceptional usability, functionality, and overall positive experience. The authentication and payment systems must be secure and reliable. The item listing, search, and communication/notification features should function seamlessly. The distance and range filtering system must enhance user convenience effectively. A robust rating and feedback mechanism should foster community trust. Sustainability analysis should demonstrate positive environmental impacts. Timelines and budgets must be adhered to. Stakeholder approval is essential, confirming alignment with project goals.

### Risk Analysis

A risk plan serves as a comprehensive inventory of potential threats to the project, accompanied by strategies to alleviate or prevent their impact. It's often said that uncertainty poses a significant challenge to planning processes. In an ideal scenario devoid of uncertainty, project plans would flawlessly unfold, and every endeavor would proceed seamlessly. However, reality often disrupts these expectations, typically at the most inconvenient moments. The risk plan functions as a safeguard against such uncertainties, akin to an insurance policy.

|  |  |
| --- | --- |
| **Name** | **Risk planning script** |
| **Purpose** | To assess risks and create a risk plan. |
| **Summary** | The risk planning meeting happens in three parts: a brainstorming session to identify risks; a discussion in which the probability and impact of each risk is estimated; and a discussion to identify actions that can mitigate risks. The end result is a risk management plan, which should be included verbatim in the final project plan. |
| **Work Product** | **Input**  Any project documentation that has been developed so far.  **Output**  Risk plan.  Assumptions generated by the Delphi process. Assumptions in the vision and scope document |
| **Entry Criteria** | The project manager has gathered the project team for a two-hour meeting to assess the  project’s risks. |
| **basic course of event** | 1. **Brainstorm potential risks:** The project manager leads a brainstorming session to identify risks. Team members suggest every risk they can think of; the project manager writes the risks on a whiteboard as they come up. Brainstorming should be reminiscent of microwave popcorn: a few ideas should “pop” at first, followed by a large number being fired rapidly,slowing down to a final few “pops.” The team will generally be able to judge when the risk identification is over. 2. **Estimate the impact of each risk:** The team assigns a number |

|  |  |
| --- | --- |
|  | from 1 (highly unlikely) to 5 (very likely to occur) to represent the estimated probability of each risk. Similarly, impact should be estimated by assigning a number from 1 (for a risk with low impact) to 5 (for a  risk which, if it occurs, will require an enormous effort to clean up).  **3. Build the risk plan**: The team identifies actions to be taken to mitigate high-priority risks and creates a risk plan that documents these actions |
| **Exit Criteria** | The risk plan is finished. |

##### Risk Plan Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Plan for Project:** Lend Borrow WebPlatform | | | | |
| **Assessment team members:**Lutfar Rahman Khan Hafez,Mahmudur Rahman Tushar,Sadman Sakib Pias,Dipa Akter ,Mohammed Shahariar Hossain,Md.Easin | | | | |
| **Risk** | **Prob.** | **Impact** | **Priority** | **Actions** |
| Limited User Adoption | **5** | **5** | **25** | Conduct extensive market research and user testing to ensure the platform meets users' needs and preferences. |
| Security Breach or Data Privacy Issues | **5** | **4** | **20** | Implement robust security measures such as encryption, secure authentication, and regular security audits. |
| Technical Issues or System Downtime | **4** | **4** | **16** | Develop a robust and scalable infrastructure with redundancy and failover mechanisms. Implement regular maintenance and updates to address technical issues promptly. Have backup systems in place to minimize downtime. |
| Lack of Trust and Credibility in the Platform | **3** | **3** | **9** | Implement a comprehensive rating and feedback system to build trust among users. Provide clear terms of service and guidelines for safe lending and borrowing. Address user concerns |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | promptly and transparently. |

**RMMM - Risk Management**

A risk mitigation, monitoring, and management (RMMM) is a systematic framework for recognizing, evaluating, and controlling risks within a project, program, or organization. It delineates the methods and protocols for reducing the effects of probable risks and continuously overseeing and handling them throughout the duration of the project or program.

##### Risk mitigation involves:

* Identifying potential risks and assessing their probability of occurrence.
* Evaluating the potential impact or consequences of each risk.
* Formulating strategies to minimize or prevent risks.
* Executing the planned risk mitigation measures.

**Limited user adoption:**Limited user adoption poses a significant risk to the success of the project, potentially resulting in underutilization of the platform and failure to achieve its objectives. It may stem from factors such as lack of awareness, resistance to change, or inadequate user experience. Addressing this risk requires targeted marketing efforts,

user-friendly design, and effective communication to drive engagement and adoption.

##### RMMM - Risk Monitoring:

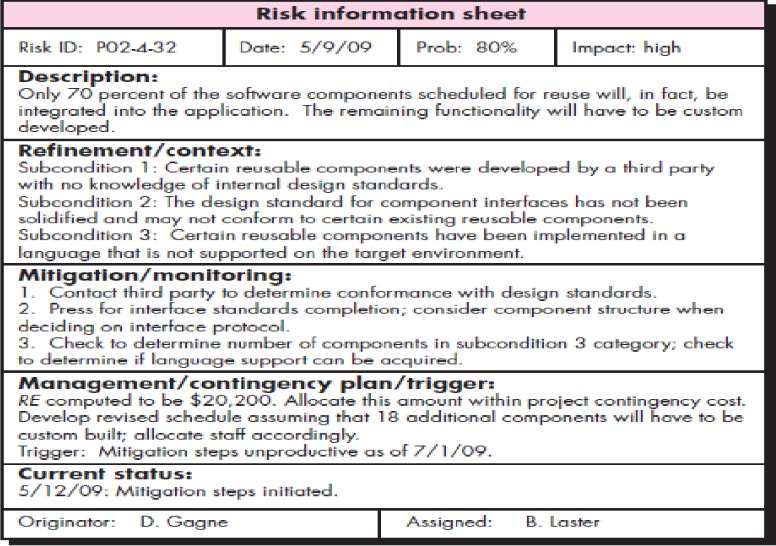
* Constant vigilance for emerging risks.
* Monitoring the effectiveness of risk mitigation tactics.
* Gathering input and feedback from stakeholders.
* Analyzing collected data to detect patterns or potential problems.
* Disseminating findings to pertinent parties.
* Implementing corrective measures as needed.

**Security Breach or Data Privacy Issues:**Security breaches or data privacy issues pose serious threats, compromising sensitive information and eroding user trust. They may result from vulnerabilities in systems or malicious attacks. Mitigating these risks demands robust security measures, including encryption, access controls, and regular audits, to safeguard data and maintain user confidentiality.

##### RMMM - Risk Management:

* Formulate a comprehensive risk management plan.
* Allocate resources and designate responsibilities for risk handling.
* Identify and evaluate potential risks.
* Create actionable strategies to mitigate risks.
* Execute the devised risk management tactics.
* Continuously monitor and assess the efficacy of risk management strategies.

**Technical Issues or System Downtime:**Technical issues or system downtime represent critical disruptions that can impede project progress and user experience. These challenges may arise from software bugs, hardware failures, or network issues. Addressing them necessitates prompt troubleshooting, effective contingency plans, and proactive measures to minimize downtime and ensure system reliability.



|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Information Sheet** | | | |
| Risk ID:P05-4-22 | Date: 3/20/2024 | Probability: 3 | Impact: High |
| **Description:** Limited user adoption refers to the risk associated with the failure of users to adopt a new system, technology, or process as intended. This could result from various factors such as inadequate training, resistance to change, lack of perceived benefits, or poor user experience. | | | |

|  |  |
| --- | --- |
| **Mitigation:** Involve users in the development process to ensure their needs are addressed. Provide tailored training sessions to ensure users understand the new system's benefits and functionalities. | |
| **Monitoring:** Monitor active users and time spent on the system to gauge adoption levels. Establish surveys to gather qualitative insights into user satisfaction and pain points. | |
| **Management:** Continuously update the system based on user feedback and evolving requirements. Proactively address resistance to change by providing support and highlighting success stories. | |
| **Current status:** 20/03/2024 Mitigation steps initiated | |
| Originator: Sadman Sakib | Assigned: Md. Easin |

### Function Point

##### What is Function Point?

Function points measure the size of an application system based on the functional view of the system.

##### What is Function Point Analysis (FPA)?

Function Point Analysis (FPA) is a technique used in software development and

project management to measure the size and complexity of a software system based on the functionality it delivers to users. It's primarily used for estimating the effort and resources required for developing or maintaining software projects. The idea is that a product with more functionality will be larger.

##### Types of FPA :

Function Point are mainly divided into five terms (Input, Outputs, Queries, Internal Files, External files) and these five terms are also divided into two parts.

1. Data Functions, 2.Transaction Functions.

##### Data Functions:

**Internal Logical Files (ILF):** These files are the master or transaction files that the system interacts with during its session.

**External Interface Files(EIF):** Represent the data that the application will use/reference, but data that is not contained within the system.

##### Transaction Functions :

**External Inputs (EI):** these are end-user actions such as putting in a login or executing a mouse click.

**External Outputs (EO):** the system provides the end-user output or interface such as a GUI display or items in a report.

**External Inquiries (EQ):** this function is initiated by the end-user. For example, the end-user wishes to submit a query to a database or requests online help. In any case, the developer provides a means for the end-user to "search" for answers.

##### Advantages and Disadvantages of FPA:

**Advantages of FPA:** The advantages of FPA are given below standardized Measurement: FPA provides a standardized and objective measure of software size, which allows for consistent estimation and comparison across different projects and organizations.

**Focus on User Requirements:** By categorizing software functionalities based on user interactions, FPA encourages a focus on the essential requirements and features that delivers value to users.

**Estimation Accuracy:** When used correctly and with appropriate adjustment factors, FPA can provide relatively accurate estimates of effort, resources, and project duration, which can help in project planning and management.

**Resource Allocation:** FPA helps in allocating resources more effectively by quantifying the size and complexity of software systems.

**Disadvantages of FPA:** The disadvantages of FPA are given belowComplexity: Function Point Analysis can be complex and time-consuming, requiring trained analysts with expertise in both the FPA methodology and the domain of the software being analyzed.

**Subjectivity:** Despite attempts to standardize the process, there can still be subjectivity in assigning function points, especially for ambiguous or complex functionalities. This subjectivity can lead to inconsistencies and inaccuracies in the estimates.

**Limited Coverage:** FPA primarily focuses on measuring the functional size of software based on user interactions, which may not capture other important aspects such as performance, scalability, and maintainability.

**Not Suitable for Agile Development:** FPA is less suited for Agile development environments where requirements are constantly evolving and emphasis is placed on delivering working software over comprehensive documentation. FPA's reliance on upfront analysis may not align well with Agile principles.

7.2 COCOMO Model Estimation 7.1 What is COCOMO Model The COCOMO (constructive Cost Model) model is an empirical and algorithmic-based model derived from data from many software projects. These data were analyzed to discover formulae that best fit the observations. These formulae link the size of the system and product, project and team factors to the effort to develop the system. 7.2 Types of COCOMO There are three types of COCOMO Models:

* Model 1: The basic COCOMO model computes software development effort (and cost) as a function of program size expressed in estimated lines of code ( LOC ). ●Model 2: The Intermediate COCOMO model computes software development effort as a function of program size and a set of "cost divers" that include subjective assessments of product, hardware, personnel, and project attributes. ●Model 3: The Advanced COCOMO model incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step ( analysis, design, etc.) of the software engineering process. There are four steps that the user needs to follow in COCOMO, which are: Step 1: The user should fill in the Information Domain table in which he/she can get the Count Total (CT) which will be used in the FP equation. The Information Domain is defined in the following manner: ●Number of user inputs: Each user input that provides distinct application-oriented data to the software is counted. Inputs should be distinguished from inquiries, which are counted separately. ●Number of user outputs: Each user output that provides application-oriented information to the user is counted. In this context, output refers to reports, screens, error messages, and so on. Individual data items within a report are not counted. ●Number of user inquiries: An inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output. Each distinct inquiry is counted. ●Number of files: Each logical master file (i.e., a logical grouping of data that may be one part of a large database or a separate file), is counted.
* Number of external interfaces: All machine-readable interfaces ( e.g., data files on tape or disk) that are used to transmit information to another system are counted. Once the above data have been collected, a complexity value is associated with each count. Once all the information is entered, the Count Total (CT) is calculated. Following is an example of this step. Step 2: The end user should calculate the "Complexity/ Value Adjustment Factor" ( ∑ Fi where i = 1 to 14). The user will give a value between 0 to 5. Once Step 1 and Step 2 are calculated, then the end user can calculated the Function Points ( FP) which is : FP = CT \* [0.65 + 0.01 \* ∑ Fi] Step 3: The end user should select a programming language from the Table found in step 3 on the main page that provides a rough estimate of the average number of lines of code required to build one function point in various programming languages. Once the programming language is selected, then the end user can calculate the Line Of Code (LOC). Thus the required KLOC is determined for the software project. Step 4: This is the final step of the basic COCOMO model. Here we will find Efforts and Duration of the project. The end user has to select one of the three

(3) types of modes, which are organic, semi-detached, and embedded. Organic Model: Relatively small, simple software projects in which small teams with good application experience work to a set of less than rigid requirements. The equation for the Effort (E) and Development time (D) for this model are: E = 2.4 \* (KLOC)^1.05 D = 2.5 \* (E)^0.38 Semi-Detached Model: An intermediate (in size and complexity) software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements. The equation for the Effort (E) and Development time (D) for this model are: E = 3.0 \* (KLOC)^1.12 D = 2.5 \* (E)^0.35 Embedded Model: A software project that must be developed within a set of tight hardware, software and operational constraints. The equation for the Effort (E) and development time (D) for this model are: E = 3.6 \* (KLOC)^1.20 D = 2.5 \* (E)^0.32 Once the end user selects his/her model, he/she calculates the Effort and the Development time. Note: ●Effort is in person-month.

* Duration is in month. The relationship between the number of staff working on a project, the total effort required and the development time is not linear. As the number of staff increases, more effort may be needed. The reason for this is that people spend more time communicating and defining interfaces between the parts of the system developed by other people. Therefore, doubling the number of staff (for example) does not mean that the duration of the project will be halved. 7.3 COCOMO Model based on our project For our project, we are using the Organic Cocomo Model based on our project criteria. The reasons are discussed below: · Project Size: Between 2 - 50k LOC. · Nature of Project: Small Scale · Deadline of the Project: Not necessary Calculation For our project, The estimated LOC will be 35,000. a = 2.4, b = 1.05, c = 2.5, d =

0.38 Effort Applied: Effort = a(KLOC)b Effort = 2.4(35)1.05 Effort = 100.341 ≈ 100 Development Time: Development time = c(Effort)d Development Time = 2.5(100.341)0.38 = 14.404 ≈ 14 People Required: Required People = Effort / 𝑇𝑖𝑚𝑒 Required People = 100.341/ 14.404 Required People = 6.9 ≈ 7 persons Productivity: Productivity = KLOC/ Effort = 35 / 100.341 = 0.348 ≈ 0.35

1. **Project Management Tools**
   1. What is project Management Tool?

Project management tools are software applications or platforms designed to assist project managers and teams in

planning, executing, monitoring, and controlling projects effectively.

These tools provide various features and functionalities to streamline project workflows, improve collaboration,

and ensure project success. Some common project management tools include Trello, Asana, Microsoft Project,

Jira, and Monday.com.

The purposes of project management tools include:

* Planning and Scheduling: Tools help in creating project plans, defining tasks, setting deadlines, and

allocating resources efficiently.

* Task Management: They enable breaking down projects into manageable tasks, assigning responsibilities,

and tracking task progress.

* Collaboration: Project management tools facilitate communication and collaboration among team members,

allowing them to share files, exchange messages, and collaborate on tasks in real-time.

* Resource Management: Tools assist in resource allocation, managing budgets, and tracking expenses to

ensure optimal resource utilization throughout the project lifecycle.

* Monitoring and Tracking Progress: They provide dashboards, Gantt charts, and other visualizations to

monitor project progress, identify bottlenecks, and track key performance indicators (KPIs).

* Risk Management: Project management tools help in identifying, assessing, and mitigating risks associated

with the project, ensuring that potential issues are addressed proactively.

* Documentation and Reporting: Tools allow for centralizing project documentation, generating reports, and

documenting project milestones, decisions, and changes for future reference.

* Integration and Automation: Many project management tools offer integrations with other software

applications and allow for automating repetitive tasks, thereby increasing efficiency and productivity.

Overall, project management tools play a crucial role in facilitating project planning, execution, and monitoring,

leading to improved project outcomes and stakeholder satisfaction.

* 1. Project Management Tools Notion

Notion is an all-in-one productivity tool that combines note-taking, task management, document collaboration,

and knowledge management into a single platform. It offers users a versatile workspace to organize and manage

various aspects of their personal and professional lives. Notion is known for its flexibility and customization

options, allowing users to tailor their workspace according to their specific needs and preferences.

Features of Notion:

* Flexible Workspace: Notion provides a blank canvas where users can create and organize content in the form

of notes, documents, databases, boards, and more.

* Blocks: Notion uses a block-based system, allowing users to add different types of content blocks such as

text, images, videos, checklists, tables, and embeds within their documents.

* Templates: It offers a wide range of pre-designed templates for various use cases, including project

management, meeting notes, personal goals, and more, making it easy to get started with different types of

documents.

* Database and Relations: Notion allows users to create databases with customizable properties and

relationships between different databases, enabling powerful organization and filtering capabilities.

* Task Management: Users can create to-do lists, kanban boards, and calendars to manage tasks and projects

efficiently. Notion supports task prioritization, deadlines, and reminders.

* Collaboration: Notion facilitates real-time collaboration by allowing multiple users to work on documents

simultaneously. Users can leave comments, mention collaborators, and track changes.

* Integration: Notion integrates with popular third-party tools such as Google Drive, Slack, Trello, and more,

enabling seamless workflow integration.

* Cross-Platform Availability: Notion is available on multiple platforms, including web browsers, desktop

applications (Windows, macOS), and mobile apps (iOS, Android), ensuring accessibility across devices.

Pros of Notion:

* Versatility: Notion's flexible workspace can adapt to various use cases, making it suitable for personal

organization, project management, team collaboration, and knowledge management.

* Customization: Users have the freedom to customize their workspace with different templates, layouts, and

content blocks to suit their specific needs and preferences.

* Collaboration: Notion's collaborative features enable seamless teamwork, allowing users to work together on

documents, share feedback, and stay updated in real-time.

* Organization: With its hierarchical structure, databases, and relational capabilities, Notion helps users

organize and manage large amounts of information effectively.

* Integration: Integration with other productivity tools enhances workflow efficiency by centralizing

information and streamlining processes. Cons of Notion:

* Learning Curve: Notion's extensive feature set and customization options may have a steep learning curve for

new users, requiring time to explore and understand its functionalities fully.

* Performance: Some users have reported occasional performance issues, such as slow loading times,

especially when working with large databases or documents.

* Mobile Experience: While Notion offers mobile apps, the user experience on mobile devices may not be as

polished compared to the desktop version, leading to usability challenges for some users.

* Pricing: While Notion offers a free tier with basic features, advanced functionalities such as unlimited file

uploads and enhanced collaboration tools are only available with paid plans, which may be a deterrent for

budget-conscious users or large teams.

Overall, Notion's comprehensive feature set, customization options, and collaboration capabilities make it a

versatile tool for individuals and teams looking to streamline their workflow and stay organized.

However, users

should be mindful of the learning curve and potential performance limitations when adopting Notion for their

productivity needs. Zoho

Zoho is a suite of cloud-based software applications designed to help businesses streamline their operations,

manage their workflow, and improve productivity. It offers a wide range of applications for various business

functions, including CRM (Customer Relationship Management), project management, email marketing,

accounting, HR, and more. Zoho aims to provide integrated solutions that cater to the needs of small and

medium-sized enterprises (SMEs) as well as larger organizations. Features of Zoho:

* CRM (Customer Relationship Management): Zoho CRM helps businesses manage their interactions with

customers, track sales, automate marketing campaigns, and analyze customer data to improve engagement and

retention.

* Zoho Projects: Zoho Projects is a project management tool that allows teams to plan, track, and collaborate

on projects. It offers features such as task lists, Gantt charts, time tracking, and document sharing to streamline

project execution.

* Zoho Mail: Zoho Mail is an email hosting service that provides business users with a professional email

address, along with features such as email filtering, spam protection, calendar integration, and mobile access.

* Zoho Books: Zoho Books is an accounting software solution for small businesses, freelancers, and

entrepreneurs. It helps users manage invoices, expenses, inventory, and financial reports, as well as track

payments and taxes.

* Zoho People: Zoho People is a human resources management system (HRMS) that automates HR processes

such as employee onboarding, attendance tracking, performance appraisal, and leave management.

* Zoho Campaigns: Zoho Campaigns is an email marketing platform that enables businesses to create, send,

and track email campaigns. It offers features such as email templates, A/B testing, segmentation, and analytics to

optimize campaign performance.

* Zoho Desk: Zoho Desk is a customer support software that helps businesses manage customer inquiries,

tickets, and support channels from a centralized platform. It offers features such as ticket management,

knowledge base, live chat, and customer satisfaction surveys.

* Zoho Creator: Zoho Creator is a low-code application development platform that allows users to build

custom business applications without coding knowledge. It offers drag-and-drop interface, pre-built templates,

and integrations with other Zoho apps. Pros of Zoho:

* Integrated Suite: Zoho offers a comprehensive suite of applications covering various business functions,

allowing users to centralize their operations and data management.

* Affordability: Zoho's pricing plans are competitive and affordable, making it accessible to small and

medium-sized businesses with limited budgets.

* Customization: Many of Zoho's applications offer extensive customization options, allowing users to tailor

the software to their specific requirements and workflows.

* Scalability: Zoho's solutions are scalable and can grow with businesses as they expand, offering flexibility to

add new features and users as needed.

* Security: Zoho prioritizes data security and offers robust security features such as encryption, user

authentication, role-based access control, and regular data backups. Cons of Zoho:

* Learning Curve: Some users may find Zoho's interface and features complex, especially when transitioning

from other software platforms, leading to a steep learning curve.

* Integration Limitations: While Zoho offers integrations with popular third-party applications, the range of

integrations may be limited compared to other platforms, potentially restricting workflow flexibility.

* Customer Support: Users have reported mixed experiences with Zoho's customer support, with some

encountering delays or difficulties in resolving technical issues.

* Performance: Occasionally, users may experience performance issues such as slow loading times or system

downtime, impacting productivity and user experience.

Overall, Zoho offers a comprehensive suite of cloud-based business applications with a focus on affordability,

customization, and scalability. While it may have some drawbacks in terms of complexity and integration

limitations, Zoho remains a popular choice for businesses looking for integrated solutions to streamline their

operations and improve efficiency.

Trello Description:

Trello is a popular project management tool that utilizes a visual Kanban-style board to help teams organize and

prioritize tasks. It's designed to be intuitive and flexible, catering to a wide range of project management needs.

Features:

1. Kanban Boards: Tasks are organized into customizable boards, with columns representing different stages of a

project workflow (e.g., To-Do, In Progress, Done).

1. Cards: Each task is represented by a card, which can contain detailed information, checklists, attachments, due

dates, and comments.

1. Drag-and-Drop Interface: Tasks can be easily moved between columns by dragging and dropping cards,

allowing for seamless task management.

1. Collaboration: Team members can collaborate in real-time, assign tasks to individuals, and add comments or

attachments to cards.

1. Integration: Trello integrates with various other tools and platforms, such as Google Drive, Dropbox, Slack,

and more, enhancing its functionality.

1. Customization: Users can customize boards, cards, and columns to suit their specific project management

needs.

1. Mobile App: Trello offers a mobile app for iOS and Android, allowing users to manage tasks on the go.

Pros:

1. User-Friendly: Trello's intuitive interface makes it easy for teams to adopt and use without extensive training.
2. Visual Organization: The Kanban-style boards provide a clear visual representation of tasks and project

progress, aiding in better organization and prioritization.

1. Flexibility: Trello's customizable features allow teams to adapt the tool to their unique workflows and

preferences.

1. Real-Time Collaboration: Teams can collaborate in real-time, fostering communication and transparency.
2. Cross-Platform Accessibility: Trello is accessible via web browsers and mobile apps, ensuring users can

manage tasks from anywhere.

Cons:

1. Limited Functionality: While Trello is versatile, it may lack some advanced features required for complex

project management needs.

1. Scalability: Larger teams or projects may find Trello's simplicity limiting in terms of scalability and handling

large volumes of tasks.

1. Dependency on Internet Connectivity: Trello's cloud-based nature requires a stable internet connection for

access, which may pose challenges in certain environments.

1. Integration Complexity: While Trello offers integrations with various tools, setting up and managing

integrations may require technical expertise.

1. Cost for Advanced Features: Some advanced features, such as automation and custom fields, are only available

in paid plans, potentially adding to the overall cost for users with specific needs.

Monday Description:

Monday.com is a versatile project management tool designed to streamline workflow and collaboration across

teams. It offers a highly customizable interface with a range of features to manage tasks, projects, and processes

efficiently.

Features:

1. Customizable Workflows: Monday.com allows users to create custom boards and workflows tailored to their

specific project needs, including Kanban boards, Gantt charts, and more.

1. Task Tracking: Users can create tasks, set due dates, assign responsibilities, and track progress in real-time.
2. Communication: Built-in communication features such as comments, @mentions, and file sharing facilitate

collaboration and feedback among team members.

1. Automation: Monday.com offers automation features to streamline repetitive tasks, such as sending

notifications, updating statuses, and triggering actions based on predefined rules.

1. Integration: It integrates with popular tools and services like Slack, Google Drive, Jira, and more, enabling

seamless data exchange and workflow integration.

1. Time Tracking: Users can track time spent on tasks and projects directly within Monday.com, helping with

resource allocation and project budgeting.

1. Reporting and Analytics: Monday.com provides customizable dashboards and reporting tools to track project

progress, identify bottlenecks, and analyze performance metrics.

Pros:

1. Versatility: Monday.com offers a wide range of features and customization options, making it suitable for

various project management needs across different industries.

1. User-Friendly Interface: Its intuitive interface and drag-and-drop functionality make it easy for teams to

collaborate and manage tasks without extensive training.

1. Collaboration: The platform facilitates seamless communication and collaboration among team members,

promoting transparency and accountability.

1. Automation: Automation features help streamline workflows, reduce manual efforts, and improve efficiency.
2. Integration: Integration with third-party tools and services enhances functionality and allows for seamless data

exchange between platforms. Cons:

1. Learning Curve: While Monday.com is user-friendly, mastering its advanced features and customization

options may require some learning time.

1. Cost: The pricing plans for Monday.com can be relatively high, especially for larger teams or organizations

requiring advanced features.

1. Dependency on Internet Connectivity: Like many cloud-based tools, Monday.com relies on internet

connectivity, which may pose challenges in offline environments.

1. Complexity for Simple Projects: For small or straightforward projects, the extensive features and

customization options of Monday.com may feel overwhelming or unnecessary.

1. Integration Complexity: Setting up and managing integrations with external tools may require technical

expertise, adding complexity to the implementation process.

Smartsheet Description:

Smartsheet is a cloud-based project management and collaboration tool that combines the familiarity of

spreadsheet software with powerful project management features. It offers a flexible platform for planning,

tracking, automating, and reporting on projects and tasks.

Features:

1. Grid View: Users can create and manage tasks, projects, and workflows in a familiar spreadsheet-like

interface, allowing for easy data entry and organization.

1. Gantt Charts: Smartsheet enables users to create dynamic Gantt charts to visualize project timelines,

dependencies, and milestones, aiding in project planning and scheduling.

1. Collaboration: Team members can collaborate in real-time, share files, add comments, and receive

notifications, fostering communication and transparency.

1. Automation: Smartsheet offers automation features such as alerts, reminders, and conditional formatting to

streamline repetitive tasks and improve workflow efficiency.

1. Integration: It integrates with popular tools and services like Microsoft Office, Google Drive, Slack, and more,

enabling seamless data exchange and workflow integration.

1. Resource Management: Users can allocate resources, track progress, and manage workloads effectively with

resource management features.

1. Reporting and Dashboards: Smartsheet provides customizable dashboards and reporting tools to track project

performance, monitor KPIs, and generate insights.

Pros:

1. Flexibility: Smartsheet's spreadsheet-like interface and customizable features make it adaptable to a wide range

of project management needs and industries.

1. Ease of Use: Its familiar interface and intuitive design make it easy for users to get started quickly with

minimal training.

1. Collaboration: Real-time collaboration features promote teamwork, communication, and accountability among

team members.

1. Integration: Integration with various tools and services enhances functionality and allows for seamless data

exchange between platforms.

1. Scalability: Smartsheet can scale to accommodate projects of all sizes, from small teams to large enterprises,

with features to support growth and complexity.

Cons:

1. Limited Advanced Features: Compared to specialized project management tools, Smartsheet may lack certain

advanced features and functionalities required for complex project requirements.

1. Learning Curve for Advanced Features: While basic functionalities are easy to grasp, mastering advanced

features and automation capabilities may require additional training or expertise.

1. Cost: The pricing plans for Smartsheet can be relatively high, especially for organizations requiring advanced

features or large user counts.

1. Dependency on Internet Connectivity: Like most cloud-based tools, Smartsheet relies on internet connectivity,

which may pose challenges in offline environments or areas with poor connectivity.

1. Complexity for Simple Projects: For small or straightforward projects, the extensive features and

customization options of Smartsheet may feel overwhelming or unnecessary. Hive

Description:

Hive is a comprehensive project management and collaboration platform designed to streamline workflows,

enhance team communication, and improve productivity. It combines project management features with

communication tools to facilitate seamless collaboration within teams.

Features:

1. Task Management: Hive offers robust task management features, allowing users to create, assign, prioritize,

and track tasks across projects.

1. Kanban Boards: Users can visualize project workflows using Kanban boards, enabling teams to easily track the

progress of tasks through different stages.

1. Gantt Charts: Hive provides interactive Gantt charts to visualize project timelines, dependencies, and

milestones, aiding in project planning and scheduling.

1. File Sharing and Collaboration: Team members can collaborate in real-time, share files, add comments, and

receive notifications within the platform, fostering communication and transparency.

1. Integration: Hive integrates with popular tools and services like Google Drive, Slack, Zoom, and more,

allowing for seamless data exchange and workflow integration.

1. Time Tracking: Users can track time spent on tasks and projects directly within Hive, helping with resource

allocation and project budgeting.

1. Analytics and Reporting: Hive offers reporting tools and analytics dashboards to track project performance,

monitor KPIs, and generate insights for informed decision-making.

Pros:

1. Comprehensive Features: Hive provides a wide range of project management features, including task

management, collaboration tools, and reporting capabilities, all within a single platform.

1. User-Friendly Interface: Its intuitive interface and user-friendly design make it easy for teams to adopt and use

without extensive training.

1. Integration: Integration with various third-party tools and services enhances functionality and allows for

seamless data exchange between platforms.

1. Real-Time Collaboration: Hive facilitates real-time collaboration among team members, promoting teamwork,

communication, and accountability.

1. Customization: Users can customize workflows, task boards, and reports to suit their specific project

management needs and preferences.

Cons:

1. Learning Curve: While Hive is user-friendly, mastering its advanced features and customization options may

require some learning time.

1. Cost: The pricing plans for Hive can be relatively high, especially for organizations requiring advanced

features or large user counts.

1. Dependency on Internet Connectivity: Like most cloud-based tools, Hive relies on internet connectivity, which

may pose challenges in offline environments or areas with poor connectivity.

1. Limited Advanced Features: Compared to specialized project management tools, Hive may lack certain

advanced features and functionalities required for complex project requirements.

1. Scalability: Some users may find that Hive's scalability is limited compared to other enterprise-level project

management tools, especially for large-scale projects or organizations with complex needs.

ClickUp Description:

ClickUp is a versatile project management platform designed to streamline workflows, enhance collaboration,

and increase productivity across teams. It offers a wide range of features and customization options to meet the

diverse needs of different industries and project types.

Features:

1. Task Management: ClickUp provides robust task management features, allowing users to create, assign,

prioritize, and track tasks efficiently.

1. Hierarchy: Users can organize tasks into hierarchical structures, including spaces, folders, lists, tasks, and

subtasks, providing flexibility and scalability for managing projects of any size or complexity.

1. Kanban Boards: ClickUp offers customizable Kanban boards to visualize project workflows, track progress,

and manage tasks through different stages.

1. Gantt Charts: Interactive Gantt charts enable users to create and visualize project timelines, dependencies, and

milestones, aiding in project planning and scheduling.

1. Collaboration: Real-time collaboration features such as comments, mentions, file sharing, and chat facilitate

communication and teamwork among team members.

1. Automation: ClickUp provides automation features to streamline repetitive tasks, trigger actions based on

predefined rules, and improve workflow efficiency.

1. Integration: It integrates with popular tools and services like Google Drive, Slack, Trello, and more, allowing

for seamless data exchange and workflow integration.

1. Time Tracking: Users can track time spent on tasks and projects directly within ClickUp, helping with resource

allocation, project budgeting, and performance analysis.

Pros:

1. Comprehensive Features: ClickUp offers a wide range of project management features, including task

management, collaboration tools, automation, time tracking, and more, all within a single platform.

1. Customization: Its highly customizable interface and flexible structure allow users to tailor workflows, task

boards, dashboards, and reports to suit their specific project management needs and preferences.

1. User-Friendly Interface: Despite its extensive capabilities, ClickUp maintains an intuitive and user-friendly

interface, making it easy for teams to adopt and use without extensive training.

1. Integration: Integration with various third-party tools and services enhances functionality and allows for

seamless data exchange between platforms, maximizing productivity and efficiency.

1. Scalability: ClickUp is scalable and adaptable to projects of any size or complexity, from small teams to large

enterprises, with features to support growth and changing requirements.

Cons:

1. Learning Curve: While ClickUp is user-friendly, mastering its advanced features and customization options

may require some learning time, especially for new users.

1. Cost: The pricing plans for ClickUp can be relatively high, especially for organizations requiring advanced

features or large user counts.

1. Dependency on Internet Connectivity: Like most cloud-based tools, ClickUp relies on internet connectivity,

which may pose challenges in offline environments or areas with poor connectivity.

1. Complexity for Simple Projects: For small or straightforward projects, ClickUp's extensive features and

customization options may feel overwhelming or unnecessary.

1. Support: Some users may experience challenges with customer support responsiveness or documentation

clarity, particularly when encountering technical issues or seeking assistance with advanced features.

Basecamp Description:

Basecamp is a popular project management and collaboration platform designed to simplify team

communication, centralize project information, and enhance productivity. It provides a straightforward approach

to managing projects and fostering collaboration among team members.

Features:

1. Project Organization: Basecamp allows users to organize projects into separate workspaces, each containing

tools and features specific to the project, such as to-do lists, discussions, files, and schedules.

1. To-Do Lists: Users can create to-do lists to track tasks, assign responsibilities, set due dates, and monitor

progress within each project workspace.

1. Messaging: Basecamp offers built-in messaging tools, including group chat and direct messaging, to facilitate

communication and collaboration among team members.

1. File Sharing: Team members can share files, documents, and attachments within project workspaces, ensuring

everyone has access to the latest information and resources.

1. Schedule: Basecamp provides a shared calendar to schedule events, deadlines, and milestones, helping teams

stay organized and aligned with project timelines.

1. Check-Ins: Users can set up automatic check-ins to gather status updates, feedback, or progress reports from

team members at predefined intervals.

1. Client Collaboration: Basecamp offers client-friendly features, such as client access, message forwarding, and

project summaries, to facilitate collaboration with external stakeholders. Pros:

1. Simplicity: Basecamp's straightforward interface and minimalist design make it easy for teams to navigate and

use without extensive training or technical expertise.

1. Centralized Information: By consolidating project-related discussions, tasks, files, and schedules within project

workspaces, Basecamp helps teams stay organized and ensures everyone has access to essential information.

1. Collaboration: Built-in messaging, file sharing, and scheduling tools promote collaboration and

communication among team members, fostering teamwork and transparency.

1. Client-Friendly: Basecamp's client collaboration features make it easy to involve external stakeholders in

project discussions, share progress updates, and gather feedback.

1. Cost-Effective: Basecamp offers a flat-rate pricing model with no per-user fees, making it a cost-effective

option for teams of all sizes.

Mastertask Description:

Master task is a project management tool which is designed to assist teams in prioritizing tasks, tracking progress,

and effectively managing the projects. It offers a wide range of features tailored towards helping teams prioritize

tasks effectively.

Features:

1. Priority Matrix: MasterTask utilizes a priority matrix to help users categorize tasks based on urgency and

importance. The tasks are organized into quadrants (Critical, High, Medium, Low) to provide a clear visual

representation of priorities.

1. Target Audience: MasterTask is suitable for teams and organizations that prioritize the tasks, time tracking, and

structured project management methodologies. It may appeal to users who prefer a more structured approach to

task management compared to more flexible, card-based systems.

1. Collaboration and Communication: Facilitates collaboration and communication among team members

through features like comments, mentions, and notifications. Users can discuss tasks, share updates, and

coordinate efforts more efficiently.

1. Task Management: Provides features for creating, assigning, and tracking tasks. Supports task dependencies,

allowing users to define relationships between tasks and ensure that tasks are completed in the correct order.

Users can create subtasks within tasks to break down complex tasks into smaller, manageable steps.

.

1. Time Tracking and Reporting: Includes built-in time tracking features, allowing users to track the time spent

on tasks and projects. Users can generate reports to analyze productivity, identify bottlenecks, and make informed

decisions.

1. Customization: Offers customization options within its structured framework. Users can customize task

priorities, deadlines, and other attributes to suit their specific needs and workflows.

1. Integration: While MasterTask may not have as many integrations available as some other tools, it likely offers

integrations with common tools used in project management and team collaboration.

1. Timeline View: MasterTask offers a timeline view that allows users to visualize tasks and deadlines over time.

This feature helps users plan and schedule tasks more effectively, especially for projects with complex timelines.

Pros:

1. Structured Task Prioritization: MasterTask's priority matrix offers a clear and structured way to prioritize tasks

based on urgency and importance. This helps teams focus on the most critical tasks first.

1. Visual Timeline View: The timeline view provides a visual representation of tasks and deadlines over time,

allowing teams to better plan and schedule their projects.

1. Built-in Time Tracking: The built-in time tracking feature allows teams to track the time spent on tasks and

projects accurately. This helps in analyzing productivity and resource allocation.

1. Collaboration Tools: MasterTask facilitates collaboration among team members through features like

comments, mentions, and notifications. This improves communication and coordination within the team.

1. Customization Options: While structured, MasterTask offers customization options to tailor the tool to the

specific needs and workflows of different teams and projects.

1. Focus on Task Management: MasterTask's focus on task prioritization and tracking makes it suitable for teams

and organizations that prioritize structured project management methodologies.

Cons:

1. Learning Curve: MasterTask's unique priority matrix and structured approach may have a steeper learning

curve for users accustomed to more flexible, card-based systems like Trello.

1. Limited Integrations: While MasterTask likely offers integrations with common project management and

collaboration tools, it may not have as extensive an ecosystem of integrations as some other tools.

1. Cost: MasterTask primarily offers paid plans, which may be a deterrent for smaller teams or individuals

looking for free or more affordable options.

1. Complexity for Simple Projects: For smaller projects or teams, MasterTask's structured approach and feature

set may be overkill, leading to unnecessary complexity.

1. Dependency Management Overhead: While task dependencies can prevent project bottlenecks, managing them

effectively requires careful planning and oversight, which may add overhead to project management

1. Limited User Base: MasterTask may not be as widely adopted or have as large a user base as more established

project management tools, potentially limiting community support and resources.

1. Updates and Support: Depending on the development cycle and support structure, users may experience delays

in receiving updates or addressing issues compared to larger, more established platforms.

Jira Description:

Jira is a robust project management and issue tracking tool designed to help software development teams plan,

track, and release high-quality software efficiently. It offers a wide range of features tailored to the needs of agile

and DevOps teams, enabling seamless collaboration and workflow management throughout the software

development lifecycle. Features:

1. Agile Boards: Jira provides customizable agile boards, including Scrum and Kanban boards, to visualize and

manage project workflows. Teams can track user stories, tasks, bugs, and epics as they progress through different

stages.

1. Issue Tracking: Users can create, prioritize, and assign issues within Jira, allowing teams to track bugs,

features, improvements, and other tasks throughout the development process. Each issue can be customized with

fields, attachments, comments, and workflow statuses.

1. Roadmaps: Jira offers roadmaps to help teams visualize their project timelines, dependencies, and milestones.

Roadmaps provide a high-level overview of project progress and help stakeholders understand the project's

trajectory.

1. Sprint Planning: Jira facilitates sprint planning by allowing teams to create and manage sprint backlogs,

prioritize user stories, estimate effort, and track sprint progress. Teams can also conduct sprint retrospectives to

reflect on their performance and identify areas for improvement.

1. Integration: Jira integrates with a wide range of development tools and services, including version control

systems like Git, continuous integration servers like Jenkins, collaboration platforms like Slack, and more. This

seamless integration enhances workflow efficiency and data synchronization across tools.

1. Customization: Jira offers extensive customization options, allowing teams to tailor their workflows, issue

types, fields, and notifications to fit their unique requirements and processes. Custom workflows can be designed

to reflect specific development methodologies and team structures.

1. Reporting and Analytics: Jira provides robust reporting and analytics tools to track project progress, monitor

team performance, and identify bottlenecks. Users can generate various reports, including burndown charts,

velocity charts, cumulative flow diagrams, and more, to gain insights into project health and productivity.

Pros:

1. Versatility: Jira's flexible architecture and extensive feature set make it suitable for a wide range of software

development methodologies, including agile, Scrum, Kanban, and hybrid approaches.

1. Customization: Jira offers extensive customization options, allowing teams to tailor the tool to their specific

workflows, processes, and terminology. This flexibility ensures that teams can adapt Jira to their unique needs

and preferences.

1. Integration: Jira's seamless integration with third-party tools and services enhances its functionality and

interoperability with the broader development ecosystem. This integration enables teams to leverage their

existing tools and streamline their workflow without disruption.

1. Scalability: Jira is highly scalable and can accommodate teams of any size, from small startups to large

enterprises. Its robust architecture and performance ensure that it can handle large volumes of data and users

without compromising speed or reliability.

1. Community Support: Jira benefits from a large and active user community, providing access to a wealth of

resources, knowledge, and best practices. Users can tap into community forums, user groups, online tutorials, and

documentation to learn from others and solve common challenges.

1. Security: Jira prioritizes data security and offers robust security features, including role-based access control,

data encryption, audit logs, and compliance certifications. This focus on security ensures that sensitive project

data remains protected from unauthorized access or breaches. Cons:

1. Learning Curve: Jira's extensive feature set and customization options may have a steep learning curve for new

users, requiring time and effort to master the tool's capabilities fully. Teams may need training or onboarding

support to effectively use Jira.

1. Complexity: Jira's rich functionality and configuration options can lead to complexity, especially for teams

with simpler projects or less experienced users. Teams must strike a balance between customization and

simplicity to avoid overwhelming users and slowing down productivity.

1. Cost: Jira's pricing plans can be relatively high, especially for large teams or organizations requiring advanced

features, add-ons, or enterprise support. Budget-conscious teams may need to evaluate the cost-effectiveness of

Jira compared to alternative solutions.

1. Maintenance Overhead: Managing and maintaining Jira instances, including upgrades, backups, and

performance optimization, can require significant time and resources. Organizations must allocate sufficient

resources to ensure the smooth operation of Jira and minimize downtime or disruptions.

1. Dependency on Internet Connectivity: Jira's cloud-based nature relies on stable internet connectivity for access

and operation. While this allows for remote access and collaboration, it also introduces dependencies on internet

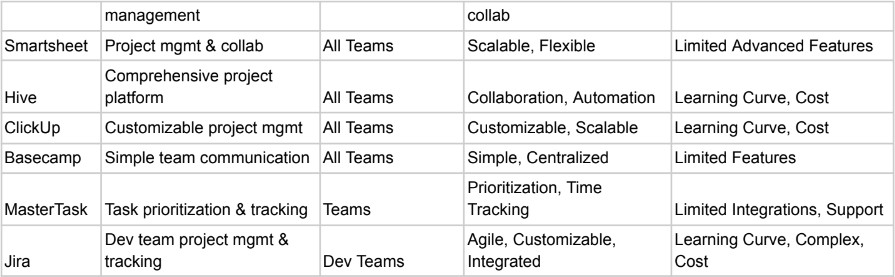
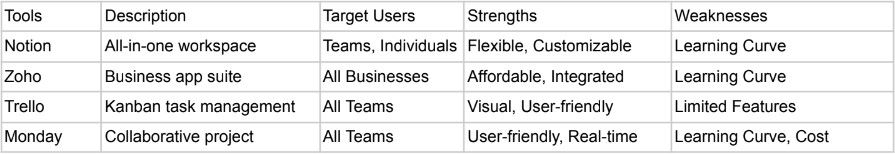
reliability and performance, which may impact productivity in certain environments.

1. Over-customization: While customization is a strength of Jira, over-customization can lead to complexity,

inconsistency, and maintenance challenges. Teams should carefully consider the trade-offs of customization and

strive for a balance between meeting specific needs and maintaining simplicity and usability.

* 1. Comparison and Benchmark Analysis:



#### Choosing the right project management tool depends on several factors, including:

* + - Project Size and Complexity: Smaller projects with fewer team members may require simpler tools, while

larger, more complex projects may need robust features such as resource management, Gantt charts, and

advanced reporting.

* + - Team Size and Collaboration Needs: Consider the number of team members, their locations (remote or

in-house), and how they will collaborate. Tools with features like task assignments, file sharing, and real-time

communication may be necessary for effective collaboration.

* + - Budget: Some project management tools are free or offer basic plans, while others require a subscription fee

for advanced features. Consider your budget constraints and whether the tool's features justify the cost.

* + - Ease of Use and Learning Curve: Choose a tool that is intuitive and easy for your team to learn. Complex

tools may require extensive training, which could impact productivity.

* + - Integration Capabilities: Consider whether the tool integrates with other software your team uses, such as

email clients, calendars, or specific project management methodologies (e.g., Agile, Scrum).

* + - Customization Options: Depending on your project's unique requirements, you may need a tool that offers

customization options to tailor workflows, fields, and notifications to your specific needs.

* + - Reporting and Analytics: If you need to track project progress, analyze data, and generate reports, ensure the

tool provides robust reporting and analytics features.

* + - Security and Compliance: For projects handling sensitive information, security and compliance features such

as data encryption, access controls, and compliance certifications may be crucial.

* + - Support and Documentation: Evaluate the level of customer support provided by the tool's vendor, as well as

the availability of user guides, tutorials, and community forums for assistance.

* + - Scalability: Consider whether the tool can scale with your project as it grows in size or complexity. You don't

want to switch tools frequently as your needs change.

By considering these factors and prioritizing this project's specific requirements, we can choose a project

management tool that best fits our needs.

Considering our team size of 5 members and your beginner level, a tool that is user-friendly, intuitive, and easy to

set up would be the most suitable choice. Based on these criteria, I would recommend Trello for our team.

#### Wideband Delphi Estimation:

Level 1: Develop Website Level 2:

Design & Planning Level 3:

-> Information Gathering

-> User Interface (UI) Design

-> User Experience (UX) Design

Development Level 3:

-> Front-End Development

-> Back-End Development

-> Content Management System (CMS) Setup

Testing & Deployment Level 3:

-> Functionality Testing

-> Usability Testing

-> Website Launch

#### Scheduling

Week 1 - Design & Planning

Day 1-2: Information Gathering - Meet with stakeholders to understand website goals, target audience, and desired functionalities.

Day 3-5: User Interface (UI) Design - Develop initial mockups for website layout and visual style. Day 4-5 (Overlaps UI Design): User Experience (UX) Design - Plan user journeys and site architecture to optimize user experience.

Week 2 - Development

Day 6-10: Front-End Development - Code the website's user interface based on approved UI mockups.

Day 8-12: Back-End Development - Develop website functionalities and integrate with databases or

APIs (if needed).

Day 10-12 (Overlaps Back-End Dev): Content Management System (CMS) Setup - Install and configure a CMS for content editing.

Week 3 - Testing & Deployment

Day 13-15: Functionality Testing - Test all website features to ensure they work as intended. Day 14-16: Usability Testing - Conduct user testing sessions to identify and address usability issues.

Day 17: Website Launch - Deploy the website to a live server and make it publicly accessible.

#### Budget

Development | $5,000 | - Front-end development (coding website interface)

* + - Back-end development (website functionalities)
    - CMS setup and configuration

Testing & Deployment | $750 | - Functionality testing (manual and automated)

* + - Usability testing with a small group of users
    - Website hosting for the first month

Content Creation | $1,000 | - Writing website copy (text content)

* + - Creating basic images or graphics (if needed)

Contingency | $1,750 | - Unexpected costs, revisions, or additional development needs Total Estimated Budget: $10,000

#### Estimate

Estimation includes budget estimation , resource estimation,time estimation, and lastly the risk estimation.

Budget estimation:

The total budget estimation is $10000 for this project. Resource estimation:

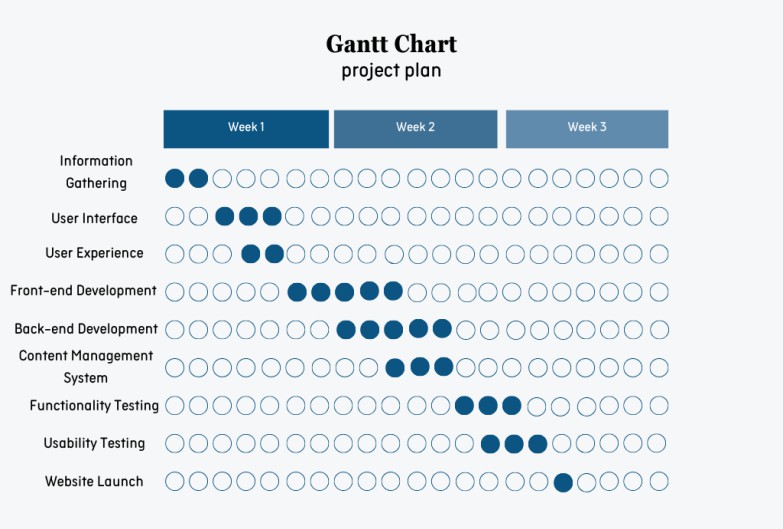
From the function it is estimated that 7 people will be needed for this project with the productivity of

0.35 kilo lines of code by effort. Time estimation:

It will take 17 days overall to complete the project. Risk estimation:

From the risk management sheet we have to prioritize the data security breach and scope creep issues over other risks to avoid major problems.

#### Gantt Chart



##### 8.8 Here's why Trello might be the best fit:

* Simplicity: Trello's simplicity makes it easy for beginners to get started quickly without a steep learning

curve. Its visual interface, consisting of boards, lists, and cards, allows you to organize tasks and workflows

intuitively.

* Flexibility: Trello's flexibility allows you to customize your boards and workflows to match your team's

preferences and project requirements. You can create lists for different stages of your project and move cards

between lists as tasks progress.

* Collaboration: Trello facilitates collaboration among team members by allowing you to assign tasks, add

comments, attach files, and set due dates directly within cards. This ensures everyone is on the same page and

aware of their responsibilities.

* Integration: While Trello itself is simple, it offers integration with various other tools and services, allowing

you to extend its functionality as your project grows. You can integrate Trello with communication tools like

Slack or file storage services like Google Drive for added convenience.

* Free Plan: Trello offers a free plan with basic features, which is suitable for small teams like yours. As

beginners, you can start with the free plan and upgrade to premium plans later if needed as your team and project

requirements evolve.

* ​

Individual Contribution:

|  |  |  |
| --- | --- | --- |
| Team | Individuals Assigned task | Completed |
| Shahariar (Team leader) | COCOMO Model Estimation , selection of PMG tool, | Yes |
| Pias | Function Point Analysis | Yes |
| Hafez | WBS, scheduling, budget assign, estimate, Gantt chart, BenchMark analysis | Yes |
| Tushar | Function Point Analysis | Yes |
| Dipa | WBS, scheduling, budget assign, estimate, Gantt chart, | Yes |
| Easin | reasoning and Update DOC, BenchMark analysis | Yes |