Project Management Document

for

<To-do it Web-App>

Version 1.0 approved

Prepared by <Nguyen Duy Nguyen>

Group 12, 2324II INT2208E 23,
VNU-UET

May 16, 2024

Table of Contents

I. Introduction

1. Purpose

The purpose of the project management documentation for To-do it web-app is not only to provide a detailed guide on implementing, managing and completing the To-do it web-app, but also to create a detailed and comprehensive document that guides project team members on specific tasks, goals, and standards. The ultimate goal is to ensure that the project is implemented effectively, all work is completed on time and the desired results are achieved in the highest quality manner.

2. Scope

The To-do it web-app project aims to develop a web application to manage personal tasks and time effectively. The scope of the project includes the design and implementation of basic features such as creating, editing, and deleting tasks, displaying task lists, marking tasks as completed, and categorizing them by list or category. In addition, the project also focuses on providing a user-friendly interface, reminder features, data sharing with synchronization capabilities, work progress statistics and reports, along with customized and personalized settings. This aims to ensure that the To-do it web-app will provide a flexible, convenient and secure task management experience for users.

3. Audience

The project management documentation for the To-do it web-app will cater to a wide range of subjects and audiences, including:

- Development Team: Software developers, software engineers, and those involved in application development will be the primary audience for this document, as they need detailed guidance on technical requirements, source code, project structure, and software development processes.

- UX/UI Design Team: User interface (UI/UX) designers must clearly understand user experience requirements, information

architecture, and interface design to ensure the app looks and feels Friendly and easy to use.

- Project Management: Project managers will use this document to plan, track progress, and ensure that the project is delivered on schedule and on budget.

- End Users: For end users, including individuals or organizations with personal time and task management needs, project management documents provide guidance on how to use and take advantage of Make the most of the features of the To-do it web-app.

- Stakeholders and Support Teams: Other stakeholders such as investors, partners or support teams may also need to understand the development process and key features of the application to be able to provide support or integration. integrate the product into their system.

In short, project management documentation for To-do it web-app caters to a diverse range of audiences, from those directly involved in the development process to those who will ultimately use the product. Products.

II. Project Scope

1. Detail

The To-do it web-app project aims to provide a broad scope, including many basic features and functions to provide a flexible and convenient task-management experience for users. Each feature is designed and implemented to meet a specific part of your personal work management needs. This range will be described in detail below.

Basic features:

- Users can create, edit and delete tasks.
- Task lists are displayed by criteria such as time, priority, or label.
- Ability to mark tasks as completed.

- Categorize tasks by list or category.

User interface:

- The interface will be designed to be friendly and easy to use.
- User experience will be optimized to create a smooth and convenient experience.

User management:

- User registration and login system will be implemented.
- Users can manage their personal information and account settings.

Reminder Feature:

- Provides the ability to send reminders about upcoming tasks.
- Users can customize reminder settings according to personal preferences.

Share and sync:

- Allows users to share tasks with others.
- Synchronize data across different devices to ensure consistency and convenience.

Statistics and reports:

- Provide reports on task progress and activities.
- Statistics on time and completed work.

Customization and Settings:

- Provides customization options and personalization settings so users can tailor the app to their needs.

Security:

- Ensure confidentiality and security of personal information and user data.

2. Requirements and constraints

The To-do it web-app project must comply with a series of requirements and constraints to ensure that it meets specific goals and customer expectations. Below is a detailed description of the project requirements and constraints:

Functional Requirements:

As for functionality, the To-do it web-app must have the ability to create, edit, and delete tasks. These basic yet important functions ensure users can manage their to-do lists flexibly and conveniently. Furthermore, the application also needs to display task lists by

criteria such as time, priority, or labels so users can organize their work effectively.

Performance Requirements:

To ensure smooth and hassle-free use, To-do it needs to have a quick response time. Users need to feel comfortable and uninterrupted when working on the application. At the same time, the stability and reliability of the system is also a factor that needs to be ensured, especially when many users access it at the same time.

Security Requirements:

In terms of security, To-do it must protect personal information and user data from any risk of intrusion or breach. The system needs to comply with legal standards and regulations related to personal data protection, including data encryption and the implementation of strict security measures.

Compatibility Requirements:

As for compatibility, To-do it needs to work smoothly on a variety of devices and platforms. The user interface must display properly on different screen types, from computers to smartphones and tablets.

Customization and Personalization Requirements:

To-do it also needs to provide customization options so users can tailor the app to their personal preferences and needs. Interface and settings personalization options will help users feel comfortable and create the best user experience for them.

Technical Constraints:

Technically, To-do it must use appropriate and effective technologies and programming languages to ensure flexibility and ease of maintenance of the source code. At the same time, the project also needs to comply with software development standards and regulations to ensure the professionalism and reliability of the application.

These requirements and constraints will guide the development and design of the To-do it web-app, ensuring that the final product will best meet all customer expectations and needs. .

III. Planning

1. Project Schedule

Task Description	Start Date	End Date
Requirements Gathering		(2 weeks)
- Define target audience and user needs	March 26, 2024	March 4, 2024
- Identify key features and functionalities	March 5, 2024	March 10, 2024
Design & User Interface (UI) Development		(3 weeks)
- Develop user personas and user stories	March 11, 2024	March 17, 2024
- Wireframe and prototype the app interface	March 18, 2024	March 24, 2024
- Design visual elements and ensure a user-friendly experience	March 25, 2024	March 31, 2024
Backend Development		(3 weeks)
- Choose and set up a web development framework	April 1, 2024	April 7, 2024
- Develop core functionalities: task creation, management, and tracking	April 8, 2024	April 14, 2024

- Integrate data storage and retrieval mechanisms	April 15, 2024	April 21, 2024
Testing & Deployment		(3 weeks)
- Conduct internal testing for functionality and usability	April 22, 2024	April 28, 2024
- Address bugs and refine the app	June 29, 2024	May 5, 2024
- Deploy the app to a web hosting platform	May 6, 2024	May 12, 2024

2. Resource Allocation

2.1. Model and resource management

Main model: Agile and Scrum

With this model, We agree that gathering resources, managing and using them is the duty and right of everyone on the team. This means that everyone in the team is responsible for:

- Researching tools and requesting access to data or systems.
- Identifying experts or collaborators who can provide specific knowledge or datasets.
- Drafting data access requests, outlining the purpose, intended use, and data security measures.
- Monitoring for new resources, attending workshops, or subscribing to relevant publications.
- Maintaining documentation, including user guides, data dictionaries, and access protocols.
- Version control for data and code to track changes and ensure reproducibility.

- Implementing access controls and permissions to safeguard sensitive information.
- Training users on proper data handling practices and usage guidelines.
- Regularly evaluating resource needs and optimizing utilization.
- Archiving outdated resources and ensuring proper disposal of sensitive data.

2.2. Software

Version control systems: Git allows teams to track changes in code, collaborate on projects and revert to previous versions if necessary.

Project management tool: Notion helps the team plan sprints, assign tasks, track progress and identify and resolve issues.

Communication tools: Messenger facilitates communication between team members, as well as creates online meetings

Source code editor: Visual Studio Code because of the flexibility and including many powerful tools for coding, testing and debugging.

Technology:

- **Frontend:** Bootstrap for fast development, consistency, compatibility, and easy customization. Being open-source, the framework is free to use, with extensive documents
- **Backend:** Firebase is a powerful, easy-to-use application development platform that saves developers time and effort.
- **Server-side:** NodeJS is a powerful, efficient, and easy-to-use web application development platform suitable for many different types of projects.

2.3. Hardware

Because this is a light and simple web application, the application's hardware requirements are an electronic device with a user interface such as a smartphone, laptop, or PC with network access.

3. Task Breakdown and Roles:

3.1. Roles:

Based on Agile and Scrum Model, we divide our team into each roles:

Scrum Master: Bùi Đức Anh

Project Owner: Co-owner by the team

Development Team

- **Backend Developer:** Nguyễn Tuấn Anh
- **Frontend Developer:** Nguyễn Công Minh
- **Tester:** Nguyễn Văn Bản
- **Designer:** Nguyễn Công Minh

Document Writer: Nguyễn Duy Nguyên

3.2. Task Breakdown:

Task Description	Assigned To	Estimated Time
Project Planning		
- Define project scope and user stories	Scrum Master, Project Owner	2 days
- Create a product backlog with prioritized tasks	Scrum Master, Project Owner	1 day
- Establish development methodology (e.g., Agile Sprints)	Scrum Master, Team	1/2 day
Design		
- Design user interface (UI) mockups for core functionalities	Designer	3 days
- Create styling guide for consistent UI elements	Designer	1 day

Backend Development		
User Management	Backend Developer	
- Develop user registration and login functionalities	Backend Developer	2 days
- Implement user roles and permissions (e.g., admin, user)	Backend Developer	1 day
Task Management	Backend Developer	
- Create database structure to store tasks, deadlines, and statuses	Backend Developer	1 day
- Develop API endpoints for adding, editing, and deleting tasks	Backend Developer	2 days
Frontend Development		
- Implement user interface (UI) based on design mockups	Frontend Developer	4 days
- Integrate backend API calls for user management and task functionalities	Frontend Developer	2 days
- Implement responsive design for different screen sizes	Frontend Developer	1 day
Testing		
- Develop test cases for core	Tester	2 days

functionalities		
- Perform manual and automated testing of user interface and features	Tester	3 days
- Report bugs and defects to development team	Tester	1 day
Documentation		
- Create user guide for navigating and utilizing the web app	Document Writer	2 days
- Develop technical documentation for developers and future maintenance	Document Writer	1 day
Deployment		
- Deploy the web app to a hosting platform	Scrum Master, Team	1 day

4. Communication Plan

Project Management Platform: We will use a project management platform Github and Notion to track tasks, deadlines, and progress. This will provide a central location for all team members to access information and collaborate.

Messenger (Facebook Messenger): We will utilize Messenger for quick updates, questions, and informal discussions. It allows for real-time communication and easy file sharing.

Google Meet: We will conduct regular video meetings using Google Meet for:

Weekly Stand-up Meetings (15 minutes): Every week, our team will have a brief stand-up meeting to discuss progress on assigned tasks, identify any roadblocks, and plan for this week.

Sprint Planning Meetings (1 hour): At the beginning of each development sprint, our team will meet to define goals, assign tasks, and answer any questions.

Sprint Review Meetings (1 hour): At the end of each sprint, our team will sit down together to review the project and receive feedback from other team members.

Ad-hoc Meetings: Additional meetings can be scheduled via Google Meet as needed to address specific issues or brainstorm solutions.

Decision-Making Process:

For minor technical decisions: The development team (backend and frontend developers) will have the autonomy to make decisions related to their area of expertise.

For major technical decisions or those impacting user experience: The Scrum Master will facilitate discussions and ensure all perspectives are considered before a final decision is made.

For overall project direction and prioritization: The Project Owner (co-owned by the team) will have the final say on project direction and task prioritization, considering feedback from the Scrum Master and development team.

5. Risk Assessment and Mitigation Strategies

Risk	Potential Impact	Mitigation Strategies

Scope Creep: Unforeseen features or functionalities are added during development.	Delays, increased development costs, and potential feature bloat.	* Clearly define project scope and user stories at the outset. * Prioritize tasks in the product backlog and resist incorporating unapproved features. * Conduct regular project reviews to ensure adherence to scope.
Technical Challenges: Encountering unexpected technical hurdles or limitations with chosen technologies.	Delays, potential need for additional resources, and frustration among developers.	* Carefully evaluate technology stack during planning to ensure suitability for project requirements. * Allocate time for prototyping and proof-of-concept development. * Encourage collaboration and knowledge sharing among developers.
Communication Issues: Miscommunication or lack of clarity between team members.	Delays, rework, and potential conflict.	* Utilize a central project management platform for clear task communication and progress tracking. * Hold regular meetings to discuss progress, roadblocks, and decisions. * Encourage open communication and active listening among team members.
Resource Availability: Team members may have competing deadlines or unforeseen personal commitments.	Delays and potential impact on project quality.	* Plan development sprints realistically considering team member availability. * Establish clear expectations and communication channels for managing workload. * Identify potential backup resources within the team for unforeseen situations.
Bug Detection and Resolution: Unidentified bugs or defects impacting functionality after launch.	Negative user experience, reputational damage, and potential rework.	* Implement a rigorous testing process using both manual and automated testing tools. * Encourage exploratory testing to uncover potential edge cases. * Establish a clear bug reporting and resolution process.

IV. Cost management

Cost management is an important part of a To-do it web-app project, ensuring that the project is completed within the expected budget without compromising on quality or schedule. Below are the detailed steps on cost management for this project:

Determine Initial Budget:
First, it is necessary to determine the overall budget for the To-do it web-app project. This budget will include costs for factors such as software development, user interface (UI/UX) design, testing, implementation and maintenance. Determining the initial budget should be done through analyzing project requirements and scope, and consulting similar projects for accurate estimates.

Budget Allocation:
The overall budget needs to be allocated in detail to specific items such as personnel costs, technology and infrastructure costs, project management costs, marketing and promotion costs, and project costs. room. Each item needs to have a specific cost allocated, to ensure that every part of the project has enough financial resources to implement.

Cost Control:
Cost control is important to ensure that the project does not exceed the expected budget. This includes tracking and monitoring actual costs against planned budgets, early detection of deviations, and timely adjustments. Using expense management tools and accounting software can help track costs effectively.

Human Resources Cost Management:
Personnel costs often make up a large portion of the project budget. Therefore, it is necessary to clearly define the roles and responsibilities of each team member, as well as the salary and associated costs. Effective human resource management will help optimize costs and ensure that human resources are used in the most effective way.

Managing Technology and Infrastructure Costs:
The To-do it web-app project will need to use appropriate technologies and infrastructure to ensure performance and reliability. The costs of procuring and

maintaining development tools, hosting services, and supporting software need to be estimated and closely managed.

Provisions and Risk Management:
Part of the budget should be reserved for contingency costs to deal with unforeseen risks. Identifying potential risks and creating a risk management plan will help the project avoid unexpected costs and keep the budget under control.

Cost Reporting and Evaluation:
Finally, periodic cost reporting and evaluation will help evaluate the effectiveness of project cost management. Cost reports need to be generated regularly to monitor the financial progress of the project, and make timely adjustment decisions if necessary.

Effective cost management not only ensures that the To-do it web-app project is completed within budget but also helps optimize resource usage and achieve business goals in the most effective way.

V. Stakeholder

1. Stakeholder Summary

Name	Description	Responsibilities
Bùi Đức Anh	Scrum Master	- Facilitate Scrum ceremonies (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective) - Manage the Scrum board and ensure team adheres to Scrum practices - Remove impediments for the development team - Coach the team on Agile methodologies
Project Owner (Team)	Product owner responsibilities are collectively held by the	- Define product vision and roadmap - Prioritize user stories and backlog items - Accept or reject completed user stories - Represent the voice of the customer

	development team.	
Development Team	Developers responsible for building the To-do it web app.	- Design, develop, and test the web application features - Work collaboratively in sprints to deliver working software increments - Continuously integrate and deploy code
Nguyễn Hữu Hoàng	Data analyst	- Collects, processes, and cleans data from different sources - Conducts statistical analysis to derive insights from data - Creates data visualizations and reports to present findings - Collaborates with team members to implement data-driven solutions - Continuously improves data collection and analysis processes
Nguyễn Công Minh	Backend Developer	- Develops server-side logic and functionalities of the web app - Manages databases and data storage - Ensures secure and scalable backend infrastructure
Nguyễn Tuấn Anh	Frontend Developer & Designer	- Designs the user interface (UI) for the web app - Develops the front-end code to deliver an interactive and user-friendly experience - Collaborates with the designer to implement visual elements
Nguyễn Văn Bản	Tester	- Creates and executes test cases to identify bugs and ensure application functionality - Reports defects and works with developers to resolve them - Contributes to improving the overall quality of the web app
Nguyễn Duy Nguyên	Document Writer	- Creates user manuals, technical documentation, and other written materials - Maintains clear and up-to-date documentation for the

		development team and future reference

2. Stakeholder profile

2.1 Busy Individual (Primary User)

Description	A professional, student, entrepreneur, or anyone juggling a demanding schedule.
Type	User
Responsibilities	- Creates and manages to-do lists - Prioritizes tasks - Sets deadlines and reminders - Tracks progress and completion
Success Criteria	- Increased productivity - Reduced stress and overwhelm - Improved time management
Involvement	- Uses the web app - Provides feedback on features and usability
Deliverables	- None directly
Comments/Issues	- User testing feedback

2.2 Team Member(Secondary User)

Description	An individual working collaboratively on projects within a team.
Type	User
Responsibilities	- Creates and manages to-do lists - Prioritizes tasks - Sets deadlines and reminders - Tracks progress and completion

Success Criteria	- Improved team collaboration - Increased project efficiency - Clearer communication and task ownership
Involvement	- Uses the web app with team-based features
Deliverables	- None directly
Comments/Issues	- User testing feedback

2.3 Team Leader/Project Manager (Secondary User)

Description	An individual responsible for overseeing and managing a team's work.
Type	User
Responsibilities	- Assign tasks and delegate responsibilities - Set deadlines and track team progress - Facilitate communication and collaboration - Utilize features for team task management and reporting - Prioritizes tasks - Sets deadlines and reminders - Tracks progress and completion
Success Criteria	- Improved team organization - Streamlined project workflows - Enhanced team communication and visibility
Involvement	- Uses the web app with team management features
Deliverables	- None directly
Comments/Issues	- User testing feedback

2.4 Scrum Master

Description	Facilitates Scrum ceremonies and ensures smooth development process.
Type	Internal
Responsibilities	- Manages the Scrum board and facilitates Scrum ceremonies - Removes impediments for the development team - Coaches the team on Agile methodologies
Success Criteria	- Timely delivery of working software increments - High team morale and productivity - Adherence to Agile principles
Involvement	- Regular meetings with the development team - Scrum board management

Deliverables	- Requires strong communication and facilitation skills
Comments/Issues	

2.5 Project Owner (Team)

Description	Represents the product vision and prioritizes development efforts. (Responsibilities shared by development team)
Type	Internal
Responsibilities	- Define product vision and roadmap - Prioritize user stories and backlog items - Accept or reject completed user stories
Success Criteria	- Successful product development aligned with user needs - Delivery of valuable features within budget and time constraints
Involvement	- Provides input and prioritizes backlog items - Approves completed features
Deliverables	- Requires clear communication and collaboration within the development team
Comments/Issues	

2.6. Development Team

Description	Developers responsible for building the To-do it web app.
Type	Internal
Responsibilities	- Design, develop, and test the web application features - Work collaboratively in sprints to deliver working software increments

	- Continuously integrate and deploy code
Success Criteria	- Functional and high-quality web application - On-time delivery of features according to sprint commitments
Involvement	- Develops and tests features - Deploys the web app
Deliverables	- Requires strong technical skills, collaboration, and ownership of the product
Comments/Issues	

2.7. Tester

Description	Ensures the functionality and quality of the web app.
Type	Internal
Responsibilities	- Creates and executes test cases to identify bugs and ensure application functionality - Reports defects and works with developers to resolve

	them - Contributes to improving the overall quality of the web app
Success Criteria	- Bug-free and reliable web application - High user satisfaction with app functionality
Involvement	- Executes test cases and reports bugs - Provides feedback on testing process
Deliverables	- Needs to stay updated on new testing methodologies and tools.
Comments/Issues	

2.8. Documents Writer

Description	Creates user manuals, technical documentation, and other written materials.
Type	Internal
Responsibilities	- Creates user manuals, technical documentation, and other written materials - Maintains clear and up-to-date documentation for the

	development team and future reference
Success Criteria	- Clear and concise user documentation - Easy access to technical information for the development team
Involvement	- Develops and maintains documentation
Deliverables	- User manuals, technical guides
Comments/Issues	- Documentation needs to be updated regularly to reflect changes in the app.

VI. Quality standard

To ensure that the To-do it web-app project achieves high quality, it is necessary to establish a clear set of quality standards. These standards will guide every phase of the project, from software development to testing and deployment. Below are the detailed quality standards for the To-do it web-app project:

Functional Standards:

Completeness: All functionality described in the project scope must be fully implemented and working as expected.
Accuracy: Features must function as required and described in the functional specification document without error.

Performance Standards:

Page load speed: Apps must have quick response times, with page load times no longer than 2 seconds.
Load capacity: The application must be able to handle large numbers of concurrent users without affecting performance.

Security Standards:

Data protection: Personal information and user data must be encrypted and protected against security attacks.

Authentication and authorization: The system must ensure that only authorized users can access sensitive functions and data.

User Interface (UI) Standards:

Friendliness: The interface must be easy to use, intuitive and not require the user to have technical knowledge to use.

Consistency: All parts of the interface should have a consistent design style, including colors, fonts, and layout.

User Experience (UX) Standards:

Convenience: In-app processes should be easy and fast, helping users complete their tasks efficiently.

User feedback: The system must provide clear feedback to users about the actions they performed, including error messages and success confirmations.

Stability Standards:

Error-free: The app must operate stably without errors or serious crashes.

Resilience: The system must be able to recover quickly from errors and crashes without data loss.

Maintainability Standards:

Ease of maintenance: Source code should be clearly written, easy to understand, and easy to maintain, including the use of good programming practices and complete source code documentation.

Scalability: The system must be designed so that it can be easily expanded and upgraded in the future without changing the underlying structure.

Testing Standards:

Functional Testing: All features must be thoroughly tested to ensure correct and complete operation.

Performance testing: The system must be tested to ensure that it meets performance standards under various load conditions.

Security testing: Security tests must be performed to detect and fix security vulnerabilities before deployment.

Implementation Standards:

Implementation process: There must be a clear implementation process and contingency plans to ensure the implementation is carried out smoothly.
Instructional documents: Provide detailed user manuals and support users during use.
By adhering to these quality standards, the To-do it web-app project will ensure that the final product is of high quality, meets user and stakeholder expectations, and works stable in real environments.