

Software Engineering Project

Project Report

Todoit

A Real-time Based To-do List

CLASS
INT2208E 23

IMPLEMENTORS
BUI DUC ANH
NGUYEN TUAN ANH
NGUYEN VAN BAN
NGUYEN CONG MINH
NGUYEN DUY NGUYEN

Navigation

PART 1. Customer Problem Statement & Requirements/ User Stories

1. Custom Problem Statement
 - A. Problem Statement
 - B. Glossary of Terms
2. System Requirements
 - A. Enumerated Functional Requirements
 - B. Enumerated Non-Functional Requirements
 - C. User Interface Requirements
 - D. User Stories

PART 2. Planning

1. Libraries & Frameworks
 - A. Online Server
 - B. Desktop Client
2. Approaching methods
 - A. System Architecture & Development Progress
 - a) From Requirements & User Stories to System Design
 - b) System Architecture Decisions
 - c) Implementing Technologies to System Architecture
 - d) Refactoring
 - e) Deploying & Testing
 - B. Project Management
 - a) Agile & Scrum
 - b) Detailed of Sprints
3. Implementers' roles

PART 3. System Modeling & Architecture / Scrum Sprints

1. System modeling
 - A. Context model

B. System Architecture

2. Scrum Sprints

A. Sprint 1

- a) Product Backlog & Development
- b) Testing

B. Sprint 2

- a) Product Backlog & Development
- b) Testing

PART 1

Customer Problem Statement & Requirements User Stories

1. Custom Problem Statement

A. Problem Statement

In today's fast-paced and digitally-driven world, individuals face challenges in managing their tasks and priorities effectively. Despite the availability of numerous productivity tools and applications, many users still struggle with organizing their daily activities, tracking progress, and prioritizing tasks efficiently. Existing to-do list applications often lack certain features or user-friendly interfaces, making it difficult for users to stay organized and focused amidst their busy schedules. First, we must begin with the question of what.

- What is a Todo list?

A Todo list, short for "To-Do list," is a simple and commonly used tool for organizing tasks, assignments, or activities that need to be completed. It serves as a reminder and a way to keep track of what needs to be done, helping individuals or teams stay organized and productive.

- What is in a Todo list?

- **Task Description:** A brief description or title of the task that needs to be completed. This description should be clear and concise, providing enough information to understand what needs to be done.
- **Priority:** Some Todo lists include a priority level or ranking for each task to indicate its importance or urgency. This can help users prioritize their tasks and focus on high-priority items first.
- **Due Date:** The deadline or due date by which the task should be completed. Including due dates helps users stay on track and ensures timely completion of tasks.
- **Status:** The current status or progress of each task, such as "Not Started," "In Progress," or "Completed." Tracking task status allows users to monitor their progress and identify tasks that require attention.
- **Notes or Details:** Additional information or notes related to the task, such as instructions, resources, or relevant details. Including notes can provide context and clarification for each task.

- **Subtasks:** For complex tasks, users may break them down into smaller subtasks or action steps. Subtasks help users manage large projects more effectively and track progress at a granular level.
- **Tags or Categories:** Some Todo lists use tags or categories to organize tasks into different groups or themes. This can help users filter and prioritize tasks based on specific criteria, such as project, context, or location.
- **Reminders or Notifications:** Todo list apps or software often include features for setting reminders or notifications to alert users of upcoming deadlines or overdue tasks. This can help users stay organized and ensure that important tasks are not forgotten.

B. Glossary Of Terms

2. System Requirements

A. Enumerated Functional Requirements

Identifier	Priority	Requirement
R1	5	The app shall allow users to create accounts
R2	2	The app should allow login with Google accounts
R3	2	The system shall allow users to reset their forgotten passwords
R4	5	Users can add tasks
R5	4	Start time and end time for tasks
R6	4	Tasks can be edited
R7	5	Data will be stored locally when the user is offline and synced into the database when the user is online
R8	1	Users can set avatar
R9	4	The app will allow users to create new tasks
R10	3	Tasks can be deleted
R11	3	Filter and search for tasks
R12	3	Task priority

B. Enumerated Non-Functional Requirements

Identifier	Priority	Requirement
R13	3	The application shall be responsive and usable on various devices (desktop, tablet, mobile).
R14	3	The application shall ensure data security and privacy for user tasks.
R15	4	The application shall function reliably and be available to users most of the time (high uptime).
R16	3	The application shall provide a user interface that is intuitive and easy to learn.
R17	2	The application shall load and display tasks quickly, even with a large number of tasks.
R18	3	System maintenance should be done regularly
R19	4	Synchronized data
R20	3	Logging and monitoring of the system and application must be in place.

C. User Interface Requirements

D. User Stories

Individual User - Basic Task Management:

- As a busy professional, I want to create new tasks quickly and easily to stay on top of my workload.
- As a forgetful person, I want to set due dates and reminders for tasks so I don't miss important deadlines.
- As someone who gets overwhelmed, I want to categorize my tasks to organize my workload and prioritize effectively.
- As a list maker at heart, I want to be able to check off completed tasks for a sense of accomplishment.

Individual User - Advanced Features:

- As a power user, I want to be able to add notes and attachments to tasks for detailed information.
- As someone who juggles multiple projects, I want to create recurring tasks to ensure routine actions don't get missed.
- As someone who likes to filter and search, I want to be able to find specific tasks based on various criteria (title, due date, category).
- As someone who likes to customize, I want to be able to personalize the app interface with themes or layouts.

Collaborating User (Optional):

- As a team leader, I want to be able to share task lists with my team members for better project coordination.
- As a team member, I want to be assigned tasks by my team leader and track my progress within the app.
- As a collaborator, I want to see real-time updates on task completion and changes made by other team members.

PART 2

PLANNING

1. Libraries & Frameworks:

A. Online Server / Backend:

- **Cloud Backend as a Service (BaaS):**

A rapidly growing cloud computing solution for tech enthusiasts and businesses seeking to save costs on building and maintaining their own backend infrastructure. A prominent player in the BaaS market is Google's Firebase product.

Firebase:

- Initially a real-time database, it has now grown to encompass 18 services (4 of which are in beta) and dedicated APIs.
- Offers a comprehensive Backend-as-a-Service solution for building, testing, and managing both mobile and web applications.
- **Reasons for choosing Firebase for the project:**
 - Suitable for both small-scale initial development and future scalability.
 - Secure and easy-to-implement platform.
 - Key features relevant to the project:
 - Realtime Database (Firestore)
 - Push Notifications
 - Authentication (Email & Password, Google, Facebook, Github)
 - Built-in Data Security with Node-Level Access Control
 - Stream-based Data Handling for Highly Scalable Applications

B. Desktop Client / Frontend:

1. Building Cross-Platform Applications with Electron and Node.js:

- **Node.js:**
 - The underlying platform for Electron, offering benefits such as a vast plugin ecosystem.
 - The npmJS package repository provides the world's largest collection of open-source libraries, further expanding development capabilities.

2. Streamlining UI Development with Bootstrap:

- **Bootstrap:**
 - A powerful toolkit composed of HTML, CSS, and JavaScript tools designed for building web pages and applications.
 - **Key benefits for the project:**

- **Responsive Design:** Ensures optimal user experience across various screen sizes.
- **Extensive Browser Compatibility:** Guarantees our application functions properly on a wide range of browsers.
- **Consistent Design with Reusable Components:** Promotes a cohesive user interface and simplifies development.
- **Rich Extensibility with JavaScript and jQuery Support:** Allows for customization and integration with existing JavaScript plugins.
- **Ideal choice for front-end development in this project** due to the lack of user-friendly tools for UI design and visual manipulation of HTML and CSS code.