

System Requirements Specification (SRS)

Ticktrak App

Version: 1.1

Last Updated: June 17, 2025

Author: Carlo Merin

I. Introduction

A. Purpose

The Ticktrak App is a task-tracking platform designed to help users manage their tasks and track progress. The main objective of this is to enable the users to create, view, update, delete and complete tasks while retain the view of their progress.

B. Scope

This Application includes:

- Welcome/Landing Page
- User Authentication (Login/Register)
- Dashboard View
- Side and Top Navigation Menu
- User Profile/Setting
- Task List (Table with search and pagination controls)

C. Tech Stack

This Application uses:

1. PHP
 - Laravel v.10
 - Composer
2. JavaScript
 - Node JS
 - Alpine JS
 - Rest API / Fetch
3. CSS
 - Tailwind CSS
 - Flowbite
4. Icons
 - Heroicons
 - Untitledui
5. Graphs
 - Apexcharts

II. User Role and Permissions

- A. User – Create, View, Update and Delete
- B. Admin – Manage All the users.

III. Functional Requirements

- A. User Authentication
 - 1. Register and Login
- B. User Profile/Setting
 - 1. Add Profile Picture
 - 2. Update Information
 - 3. Change Password
 - 4. Delete Account
- C. Task Management
 - 1. Create new Tasks
 - 2. View List of Tasks
 - 3. Update / Delete Tasks

IV. Non-Functional Requirements

- A. Security: Password hashing
- B. Performance: Response should be 500ms minimum
- C. Validation: Proper input sanitation and error message

V. Database Schema (Tables)

- A. Table: Users
 - 1. id – Primary Key
 - 2. name – String | max:255
 - 3. profile – String | Null
 - 4. email – String | Email | Unique
 - 5. password – String | Hash
- B. Table: Tasks

1. `id` – Primary Key
2. `title` – Required | String | max:255
3. `details` – Null | text
4. `priority` – Required | String | enum ['low', 'medium', 'high'] | default = low
5. `status` – Required | String | enum ['pending', 'completed'] | default = pending
6. `due_date` – Null | date
7. `user_id` – Foreign Key | exists: users, id

VI. Routes (API)

A. Task Api Endpoints

Name	Method	Url	description	authorization
<code>index</code>	<code>GET</code>	<code>/api/tasks</code>	View task list	auth/logged
<code>store</code>	<code>POST</code>	<code>/api/tasks</code>	Create new tasks	auth/logged
<code>update</code>	<code>UPDATE</code>	<code>/api/tasks/{id}</code>	Update selected tasks	auth/logged
<code>destroy</code>	<code>DELETE</code>	<code>/api/tasks/{id}</code>	Delete selected tasks	auth/logged

VII. Models

A. User

1. Fillable Attributes
 - `name` | `profile` | `email` | `password`
2. Relationship
 - `hasMany(Task)`

B. Task

1. Fillable Attributes
 - `title` | `details` | `priority` | `status` | `due_date` | `user_id`
2. Relationship
 - `belongsTo(User)`

VIII. User Interface

A. Top Navbar

1. Search Function
2. Dark/Light Mode

3. Profile (Name and Email)

B. Side Navbar

- Default View: Shows only Icons (Menu and Others)
- Toggled View: Shows both Icons and label for each items in the side panel

1. Application Logo

2. Menu (placeholder)

- Dashboard
- My Task

3. Others (placeholder)

- Setting
- Logout

C. Task List UI (CRUD)

1. Header:

- Label: Task List
- Button: Add Tasks (Modal)
 - Title: Add Task
 - Input Fields: Title, Priority, Due Date, Details
 - Close Button: Close
 - Submit Button: Add Task

2. Main:

- Search Box
- Table
 - # | Title | Description | Priority | Due Date | Action
 - Action:
 - Button: Update Task (Modal)
 - Title: Update Task
 - Input Fields: Title, Priority, Due Date, Details
 - Close Button: Close
 - Submit Button: Update Task
 - Button: Delete Task (Modal)
 - Title: Delete Task
 - Description: Are you sure you want to delete this task?

- Close Button: Close
- Submit Button: Delete Task
- Pagination Controls
 - Limit per page (10, 25, 50, etc.)
 - Previous, Next

Change Logs:

1. v.1.1

Add Collapsible sidebar to maximize the space of the content of the main page.

1. Default: showing sidebar icons only.
2. Toggled: shows both icons and label.

2. v.1.2

Implementing CRUD API for Tasks.

1. Add Models and Relationship
2. Add Routes API / Rest API