
Software Design Document

Want2Remember Project

Final Documentation – May 6, 2025

Kevin Bayona-Galindo & Nikolazi Tartinsky

May 10, 2025

Contents

1	Introduction	2
2	System Architecture	2
3	Functional Requirements	2
4	Non-functional Requirements	3
5	External Interfaces	3
6	Assumptions and Constraints	3
7	User Interface Overview	3
8	Deployment Notes	3
9	References	4
10	Workflow Diagram Image / System Architecture Image	4

Version History

Version	Date	Description	Authors
v1.0	04/15/2025	Initial requirement draft	Kevin & Niko
v2.0	04/22/2025	Added persistence and validation logic	Kevin & Niko
v3.0	04/29/2025	Included delete features and UI polish	Kevin & Niko
v4.0	05/05/2025	Finalized functional requirements and added test planning	Kevin & Niko

1 Introduction

Want2Remember is a lightweight web application that enables users to create, view, and delete personal memory entries. It is built entirely using front-end technologies (HTML, CSS, JavaScript) and leverages 'localStorage' to persist user data. This SDD outlines the system architecture, design decisions, component structure, and deployment approach of the application.

2 System Architecture

- **Frontend:** HTML, CSS, JavaScript
- **Storage:** localStorage
- **Deployment:** Docker (NGINX)
- **Testing:** TestRail (Snapshots 2-4)
- **Project Tracking:** Jira (Sprint-based)

Workflow Diagram Description

User accesses a form → submits memory → JavaScript stores it in `localStorage` → display updates dynamically → memory reloaded on refresh → user can delete memory → `localStorage` updates.

3 Functional Requirements

- The system must allow users to add a memory entry that includes a title and note.
- The system must display all previously saved memory entries below the form.
- The system must allow users to delete individual memory entries.

- The application must persist user data using ‘localStorage’ between sessions.

4 Non-functional Requirements

- The app must function correctly in all modern browsers.
- The UI should be responsive and accessible.
- The app should not require internet access after initial loading.
- Page reloads must not erase saved data.

5 External Interfaces

- HTML5 and CSS for UI
- JavaScript for logic and DOM manipulation
- Browser localStorage API for persistence

6 Assumptions and Constraints

- No login or authentication is required.
- Data is stored on a per-browser basis.
- The app is primarily for desktop use, but responsive design supports mobile.

7 User Interface Overview

- Clean, minimalist design with a form at the top and saved memories listed below.
- Confirmation message appears when a memory is saved.
- Each memory has a delete button for individual removal.
- Fully responsive layout for mobile and desktop use.

8 Deployment Notes

- The application can be deployed using Docker with NGINX to serve static files.
- Developers can also run the app locally by opening ‘index.html’ in a modern browser.
- A local development server such as Live Server can be used for real-time testing.

9 References

- <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>
- https://hub.docker.com/_/nginx
- <https://www.docker.com/products/docker-desktop>
- <https://overleaf.com>
- <https://www.atlassian.com/software/jira>
- <https://github.com/kbthepioneer/Want2Remember-Final-Project>

10 Workflow Diagram Image / System Architecture Image

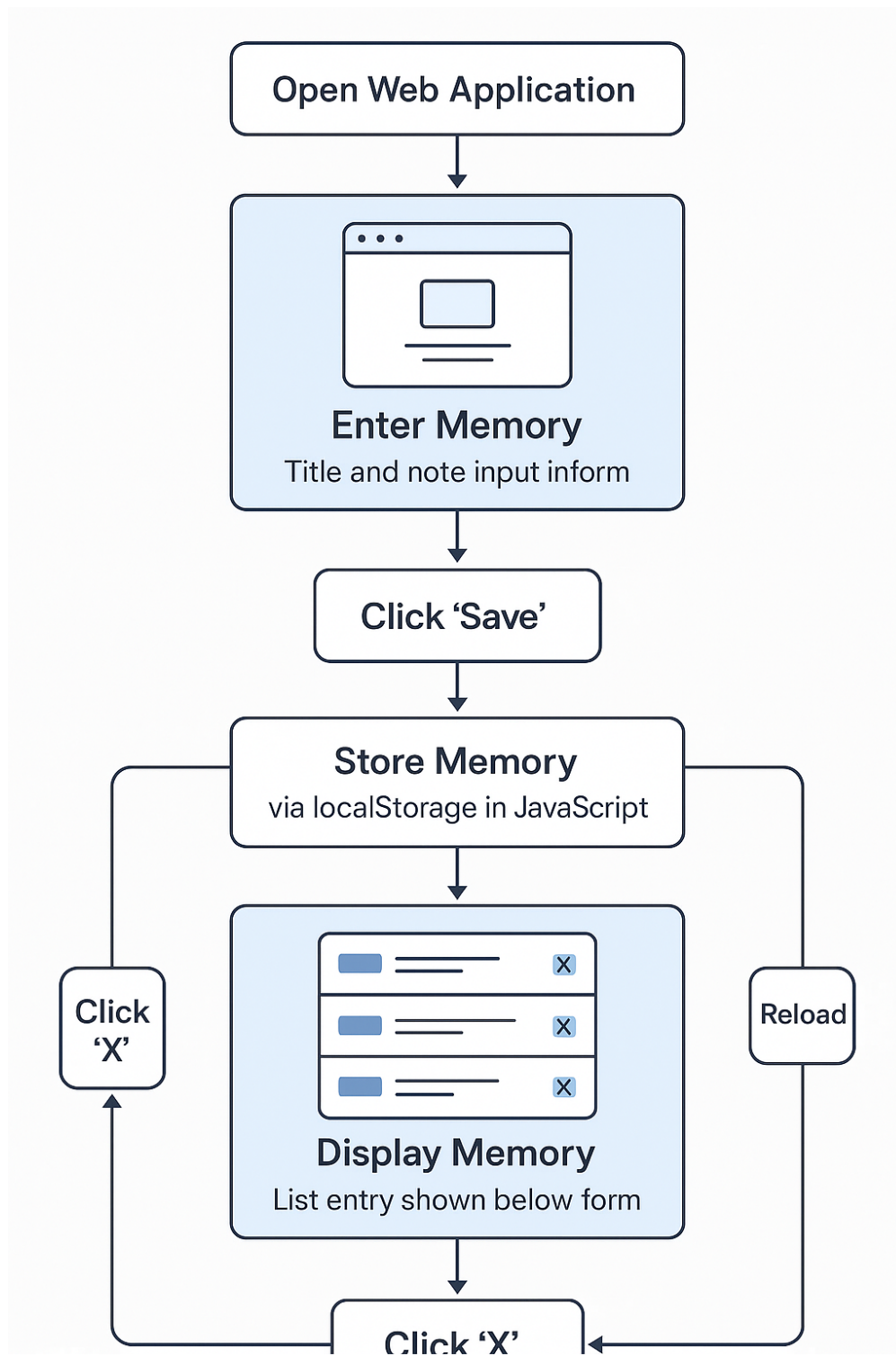


Figure 1: Final Workflow Diagram

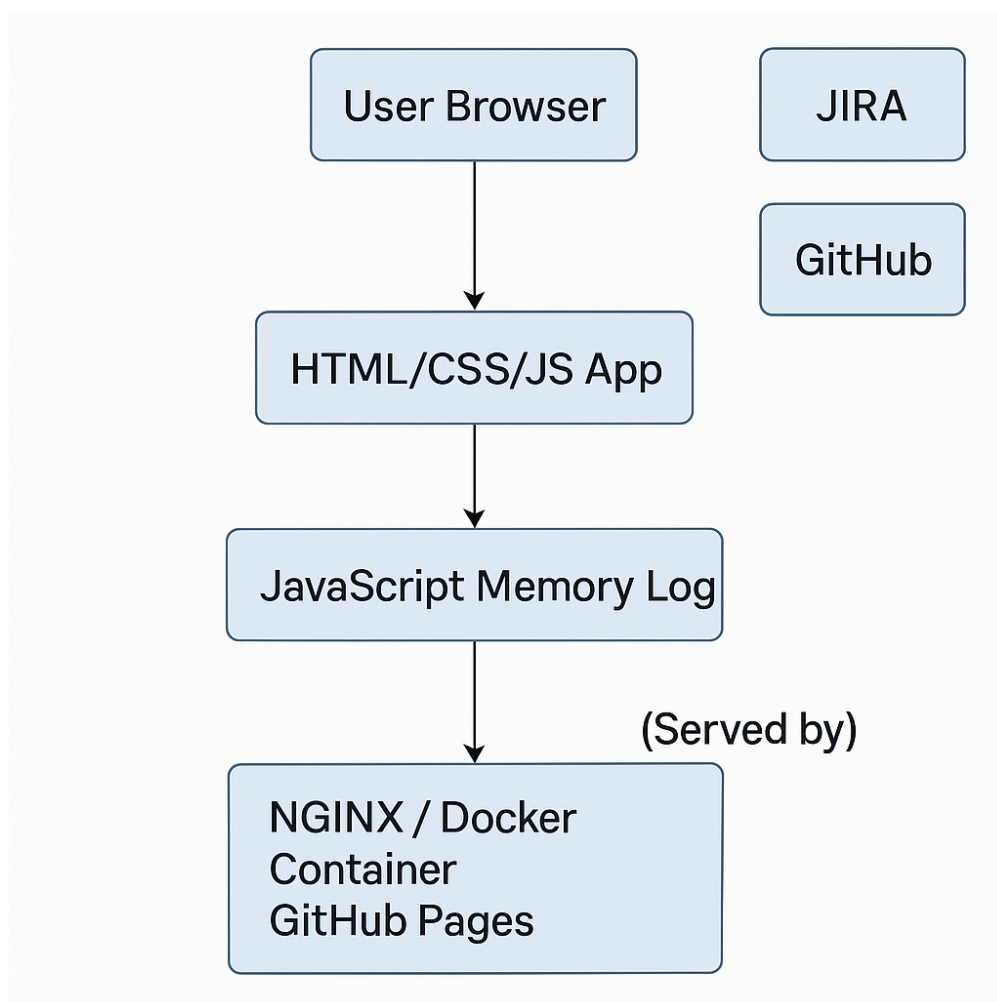


Figure 2: System Architecture Diagram