

High Level Design

# Little Wins

version 1.0

December 07th 2025

## Document history

| <b>Version</b> | <b>Status</b>                  | <b>Date</b> | <b>Responsible person</b>                         | <b>Reason for change</b>  |
|----------------|--------------------------------|-------------|---|---|
| 1              | Came across the ideas          | 15.11.2025  | Theresa Hartmann<br>Arooj Shahzadi<br>Junu Rahman | Project started and modified  |
| 2              | More Modified version          | 22.11.2025  | Theresa Hartmann<br>Arooj Shahzadi<br>Junu Rahman | Incorporated feedback on SRS  |
| 3              | Added Diagrams                 | 26.11.2025  | Junu Rahman<br>Arooj Shahzadi<br>Theresa Hartmann | Class diagram, activity diagram, and sequence diagram   |
| 4              | Reviewed document and diagrams | 29.11.2025  | Theresa Hartmann<br>Arooj Shahzadi<br>Junu Rahman | Discussed session sequence, logout modalities and options; worked on section about components of the system               |
| 5              | Reviewed document              | 06.12.2025  | Theresa Hartmann<br>Arooj Shahzadi<br>Junu Rahman | Clearer definition of "session" and required changes connected with that issue, incorporate feedback on last presentation |
| 6              | Finalize document              | 07.12.2025  | Theresa Hartmann<br>Arooj Shahzadi<br>Junu Rahman | Review diagrams, finalize document  |

## Abbreviations

|             |   |
|-------------|---|
| <b>HLD</b>  | <b>High level Design Document</b>         |
| <b>UI</b>   | <b>User Interface</b>                     |
| <b>API</b>  | <b>Application Programming Interface</b>  |
| <b>GDPR</b> | <b>General Data Protection Regulation</b> |
| <b>MVP</b>  | <b>Minimum Viable Product</b>             |

## Table of contents

|   |          |
|---|----------|
| <b>1. INTRODUCTION</b>                              | <b>1</b> |
| 1.1. Purpose  | 1        |
| 1.2. System Overview                                | 1        |
| 1.3. Definitions                                    | 1        |
| <b>2. Components of the System</b>                  | <b>2</b> |
| 2.1. Web Client                                     | 2        |
| 2.1.1 Role in the Overall System                    | 2        |
| 2.1.2 Functionality                                 | 2        |
| 2.2 Backend Server                                  | 3        |
| 2.2.1 Role in the Overall System                    | 3        |
| 2.2.2 Functionality                                 | 3        |
| 2.3 Database  | 4        |
| 2.3.1 Role in the Overall System                    | 4        |
| 2.3.2 Functionality                                 | 4        |
| 2.4 External Activity Services                      | 4        |
| 2.4.1 Role in the Overall System                    | 4        |
| 2.4.2 Functionality                                 | 5        |
| <b>3. DETAILED DESIGN</b>                           | <b>5</b> |
| 3.1. Architecture                                   | 5        |
| 3.1.1. Class Diagram                                | 6        |
| 3.1.2. Activity Diagram - Run a Little Wins Session | 7        |
| 3.1.3. Sequence Diagram                             | 8        |

# 1. Introduction

## 1.1. Purpose

The purpose of this High-Level Design (HLD) document is to add technical detail to the existing Software Requirements Specification (SRS) of Little Wins – Meaningful Micro-Moments App.

It serves as a model for implementing the system, especially for structuring the backend, database and client, and for understanding how the main components interact. The HLD can also be used as a reference during implementation and testing.

## 1.2. System Overview

Little Wins is a lightweight web application that helps users make meaningful use of short waiting periods instead of doomscrolling.

It provides:

- **User registration and login** for personalised access and saved session history.
- **Mode selection** (Mood Booster, Brain Booster, Relax & Reset, Kindness & Connection) and **time selection** (3, 5, 10, 15 minutes).
- **Automatic activity suggestions** based on the chosen mode and available time.
- **Activity content from internal sources or optional external APIs** e.g., jokes, riddles.
- **A completion summary** after each activity to reinforce positive behaviour.
- **Session storage in the database**, allowing users to keep track of their Little Wins.
- A system architecture consisting of a **Web Client, Backend Server, Database**, and **optional external activity services**.

## 1.3. Definitions

**Web Client:** Browser-based frontend (HTML/CSS/JavaScript) used by the end user.

**Backend Server:** Server-side application exposing a REST API and containing business logic.

**Database:** Persistent storage for users, sessions, activities, etc.

**Mode:** User intention category like Mood Booster, Brain Booster, Relax & Reset, Kindness & Connection.

**Activity Type:** Template for an activity e.g. mood-lifting, riddle-style, movement-based, sensor mini-game.

**Activity:** Concrete task shown to the user in a session, based on a mode and an activity type.

**Session:** One complete cycle starting with mode selection and time selection and ending with the summary screen. A session always includes exactly one activity: mode selection → time selection → display of one activity → activity completion → summary. If the user wants to do another activity afterwards, this is started as a new session.

## 2. Components of the System

### 2.1. Web Client

#### 2.1.1 Role in the Overall System

The Web Client is the main user interface of Little Wins. It runs inside the user's browser and allows the user to interact with the system by selecting a mode, requesting activities, completing sessions and viewing summaries and statistics. The Web Client communicates exclusively with the Backend Server via HTTP/REST.

#### 2.1.2 Functionality

The Web Client provides the following functionality:

- Display registration and login UI and submit user credentials.
- Allow the user to select a mode and a duration for a new session.
- Request and display an activity based on mode and duration.
- Provide controls to mark an activity as completed or request another activity within the same session.
- Display the summary screen containing mode, duration, activity type, and optional media or sensor outputs.
- Display the statistics screen (sessions per day, sessions per mode, optional streaks).
- Optionally show local browser notifications (e.g., inactivity reminders), if the user grants notification permissions.

## **2.2 Backend Server**

### **2.2.1 Role in the Overall System**

The Backend Server is the central processing component of the Little Wins architecture. It implements all business logic, exposes REST API endpoints, manages sessions and activities, and communicates with the Database and optional External Activity Services. The Backend Server is responsible for ensuring consistent and secure data access. Each session in the system represents exactly one activity for a given mode and duration; the Backend Server ensures that one activity is created, completed and summarised per session.

### **2.2.2 Functionality**

The Backend Server performs the following tasks:

- Register users and authenticate login requests.
- Create and manage user sessions (mode, duration, start/end time).
- Select appropriate activities for each session based on the mode–activity mapping defined in the SRS.
- Handle the “request another activity” option and return additional matching activities.
- Mark activities as completed and generate the final summary.
- Store and retrieve session data to support statistics (sessions per day, sessions per mode).
- Communicate with external APIs (e.g., Joke API, Riddle API) for optional advanced activities.
- Provide secure access to persistent data stored in the Database.

## **2.3 Database**

### **2.3.1 Role in the Overall System**

The Database provides persistent storage for all essential data in the Little Wins system. It stores user accounts, sessions, activity information and optionally external content. The Backend Server is the only component that directly interacts with the Database. Each session record is linked to exactly one activity record, which simplifies storage and querying of session history.

### **2.3.2 Functionality**

The Database supports:

- Storing user account data (ID, email, password hash).
- Storing session records (user, mode, duration, timestamps, status).
- Managing activity definitions and activity types.
- Storing completed activities, including optional media references or sensor results.
- Providing data needed for summary generation and statistics (session history, activity usage patterns).
- Optionally caching external content such as jokes or riddles to reduce API calls.

## **2.4 External Activity Services**

### **2.4.1 Role in the Overall System**

External Activity Services provide dynamic or sensor-based content for optional activity types in Little Wins. These services are used only when the user selects a challenge that requires external data or device capabilities. They are accessed exclusively by the Backend Server, ensuring that all external communication is centralised and secure.

Although the MVP uses internally stored activities, these external services allow the system to offer more engaging challenges such as jokes, riddles, GPS-based tasks and sensor mini-games. If an external service is not available, the Backend Server automatically falls back to internal content so the app remains functional.

## **2.4.2 Functionality**

Depending on configuration and device permissions, the system may use the following external services:

- **Jokes API:** Provides joke text for the Joke Challenge. Used to fetch new jokes or provide alternatives when the user requests another activity.
- **Riddles API:** Provides riddles and their solutions for the Riddle Challenge. Supports “get new riddle” functionality.
- **GPS / Geolocation Service (via Browser API):** Used for the GPS Shape Challenge. Enables the system to request the user’s location coordinates during movement, track the GPS path, compare the path to a target shape (circle, square, heart), compute an accuracy score. This service is available only if the user grants location permission.
- **Device Sensor API (Orientation, Motion, Gyroscope):** Used for Sensor Mini-Games. Provides real-time sensor data that allows simple interaction tasks such as balancing, tilting or rotating the device. Requires device and browser support (e.g. modern mobile browsers).

# **3. Detailed Design**

## **3.1. Architecture**

The architecture follows a simple three-layer model:

- Presentation Layer: Web Client in the user’s browser.
- Application Layer: Backend Server exposing REST endpoints and using internal services (e.g. ActivityService, StatisticsService).
- Data Layer: Database for persistent storage plus optional External Activity Services.

The Web Client communicates only with the Backend Server via HTTP/REST. The Backend Server coordinates all requests, reads and writes data in the Database, and optionally calls External Activity Services (e.g. Joke and Riddle APIs) when a corresponding activity type is selected. This separation ensures that the system remains modular, testable and easy to extend.

# High Level Design

## Little Wins

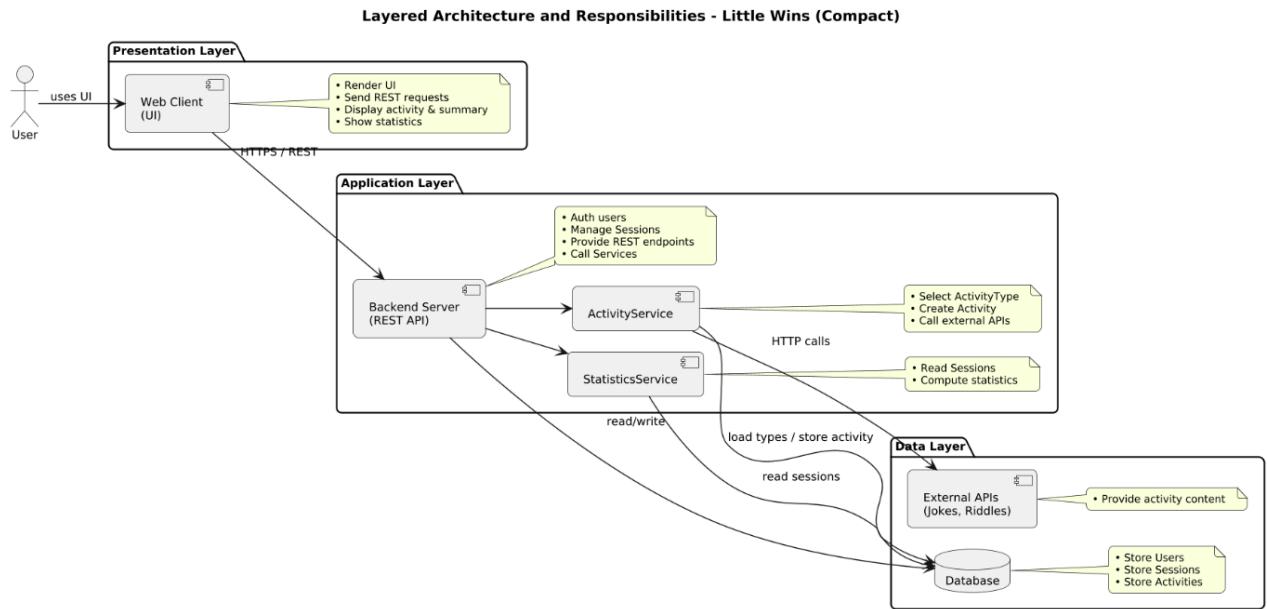


Bild 1: Architecture of Little Wins

### 3.1.1. Class Diagram

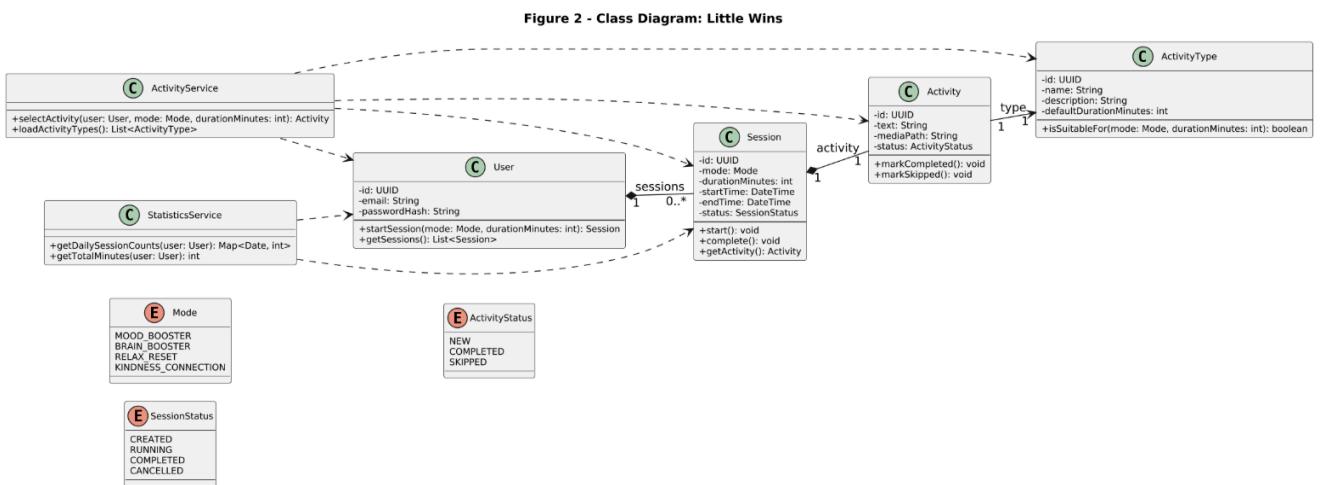


Bild 2: Class Diagram Little Wins

### 3.1.2. Activity Diagram - Run a Little Wins Session

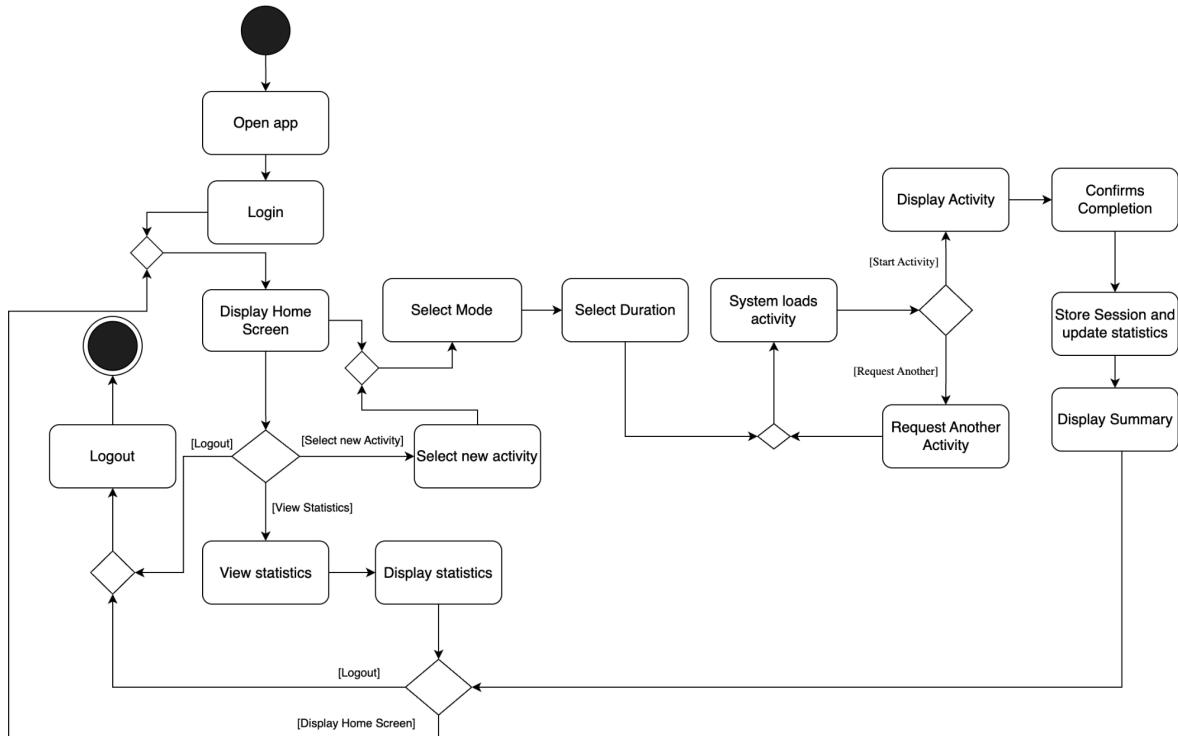


Bild 3: Activity Diagram Little Wins

### 3.1.3. Sequence Diagram

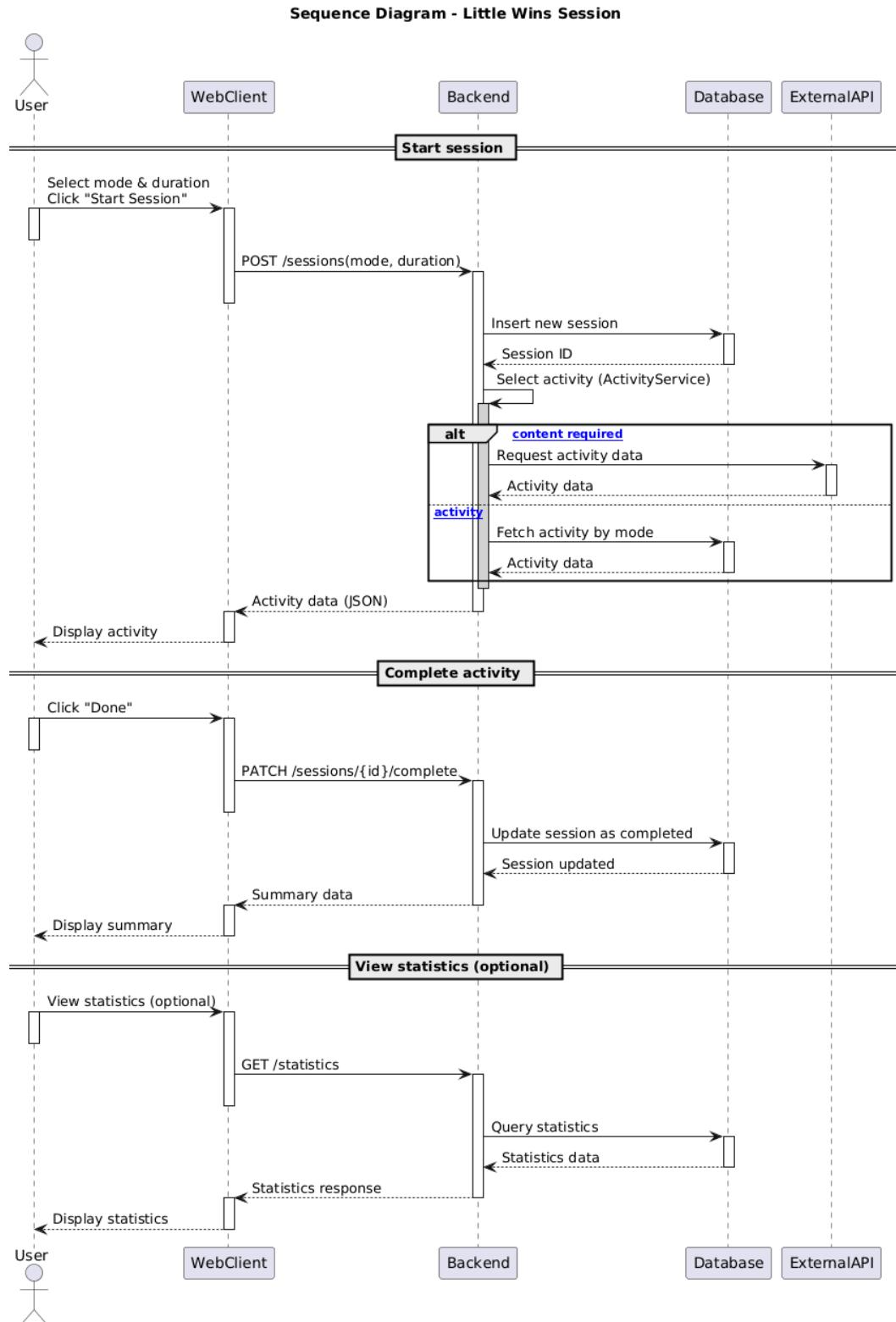


Bild 4: Sequence Diagram Little Wins