Software Requirements Specification

for

Habit Tracker

Prepared by:

Group Name: AAAAS

Name	Student ID	Email
Srachet Rai	BT22GCS155	srachet.rai22@st.niituniversity.in
Abhi Bhardwaj	BT22GCS108	abhi.bhardwaj22@st.niituniversity.in
Anmol Ranjan	BT22GCS082	anmol.ranjan22@st.niituniversity.in
Adarsh Bajpai	BT22GCS055	adarsh.bajpai22@st.niituniversity.in
Avijith Manikandan	BT22GCS140	avijith.manikandan22@st.niituniversity.in

Instructor: Mr. Manish Hurkat
Course: Capstone 1

Date: 15/02/2025

Table of Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Initial Draft v1	Abhi Bhardwaj	Started the initial draft of the SRS document, outlining the major headings and subheadings to be discussed	02/02/2024
Initial Draft v2	Abhi Bhardwaj, Anmol Ranjan	Discussed the dependencies and updated the assumptions along with Functional Requirements	15/02/2024
Mid Submission 1	Abhi Bhardwaj, Srachet Rai, Anmol Ranjan, Avijith Manikandan	Added the modules along with their coupling and cohesion. Went over the final document, made relevant changes to product scope and functionality.	20/02/2024

Introduction

➢ Document Purpose

The following document provides a detailed specification of the habit tracker application, outlining its functionality, design constraints, and specific requirements. It may serve as a guide for developers, designers, stakeholders, and testers throughout the development process.

➢ Product Scope

The habit tracker is a web-based and mobile-friendly application that helps users track, and analyze their daily habits, goals, and tasks. It integrates an AI-powered chatbot for personalized recommendations and insight, supports speech-to-text input and ensures secure data storage using MongoDB. Key features include -

- Habit and Goal Tracking (with informative dashboard)
- Pomodoro Timer
- AI Assistant (LLM API) and Progress Map
- In application Note-Taking
- Speech-to-Text Functionality
- Motivational Quotes and load time tips

➢ Intended Audience and Document Overview

Developers: Require technical specifications and instructions for implementation.
Project Managers: Consider the project's objectives, schedule, and essential specifications.
Marketing Staff: attracted by the features, advantages, and positioning of the app in the market.
Users: Need guidance on how to utilize the app and get its advantages.
Testers: Require acceptance criteria, use cases, and testing protocols.
Documentation Writers: To build user manuals and support materials, comprehensive information is required.
System Administrators: System requirements are necessary for setup and maintenance.
Designers: To create interfaces and experiences, information about user requirements is required.

➢ Definitions, Acronyms, and Abbreviations

Here is a short list of abbreviations and acronyms used in the SRS document, sorted alphabetically:

API: Application Programming Interface

DP-DP: Data protection and digital privacy law

LLM: Large Language Model

SRS: Software Requirements Specification

URL: Uniform Resource Locator

MERN: MongoDB, Express,js, React, Node.js

UI/UX: User Interface/ User Experience

➢ Document Conventions

In general, this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single-spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

Formatting Conventions:

- Font: To ensure readability on a variety of devices and formats, the main text is written in Arial at an 11-point size.
- Headings and Subheadings: To make it easy to identify the important sections, headings (such as section titles) are bolded and set in a 14-point font size.
- Paragraphs: To improve readability, a space is inserted between each paragraph and the body content is left-aligned with a single line spacing.
- Lists: To convey information in an ordered and succinct manner, bullet points are utilized for unordered lists.
- Sequential instructions are easier to follow when presented in ordered lists, which are utilized for organized tasks or procedures.

➢ References and Acknowledgments

- https://api-docs.deepseek.com/guides/reasoning_model
- https://platform.openai.com/docs/api-reference/introduction
- https://cloud.google.com/vision/docs/ocr
- https://cloud.google.com/speech-to-text/docs

Overall Description

➢ Product Overview

This product is a responsive website that has been designed to make it easier for people to monitor their habits and get quick, reliable guidance when required. Users will be able to easily sign up and login into the application. After they have been logged in once, they will be able to create and track their habits, set goals, and also use AI-powered suggestions for further improvement. This application will securely store this information for future reference.

By integrating advanced LLM models, the following application will provide immediate recommendations for habit improvements shaped according to the user's needs. If the user is found struggling to maintain consistency, the app can suggest some modifications and motivational quotes to keep them engaged. This is how this application will provide an interactive and intelligent platform to help users build positive habits and achieve their goals.

➢ Product Functionality

1. Provide a bulleted list of all the major functions of the system

- User Registration & Login:

- Register using personal details or third-party accounts (Google).
- Secure login to access user profiles and tracked goal history.

- Habit and Goal Tracking

- create, modify and track habits with detailed analytics and an interactive dashboard displaying the amount of days the habit has been followed for, if any streaks are maintained.

- AI Chatbot Integration

- suggest improvements and modifications based on the user activity.

- Pomodoro timer

- Set timers for particular tasks based on the pomodoro technique.

- Speech-to-text

- converts voice input into tasks and habits on the main page.

- Note taking

- Allow users to create notes within specific tasks

.

- Motivational quotes and load time tips

- Show a motivational quote on every application cold start. Show tips while the user is waiting for a response from the AI agent.

Design and Implementation Constraints

Hardware Limitations:

Database:

- Utilize MongoDB(NoSQL) for storing user habits, progress, and chatbot interaction.
- It also supports indexing for faster retrieval of data.
- Implementation of data scaling and redundancy.
- Optimized query performance using aggregation pipelines for habit-tracking analytics.

Security:

- Data encryption uses AES-256 encryption for storing sensitive user data.
- Secure authentication implements OAuth authentication
- Access contro**l**: Role-based permissions for restricting admin and user-level functionalities.
- HTTPS enforcement to prevent attacks.
- Regular security audits and penetration testing to ensure system integrity.

Performance:

- Ensures real-time habit tracking with efficient database queries.
- AI chatbot responses optimized for sub-2-second latency.
- Implements caching strategies (Redis) to reduce load on the database.
- Uses background processing for non-essential tasks (e.g., analytics generation, chatbot learning updates).

Compatibility:

- Web application built with React.js for responsive UI on desktop and mobile.
- Supports progressive web app (PWA**)** features for offline usage.
- Mobile compatibility-tested across major browsers (Chrome, Firefox, Safari, Edge).

Assumptions and Dependencies

Assumptions:

Stable API Availability:

It is assumed that the ChatGPT/LLaMA API and third-party authentication services (e.g., Google, Facebook) will be consistently available and maintain high uptime. Any outages or changes in these services could disrupt key functionalities such as AI based habit recommendations, chatbot interaction and user authentication.

User Device Compatibility:

The system is assumed to be used primarily on modern smartphones and browsers that support the latest web standards. If a significant number of users access the app with outdated devices or browsers, it may require additional development efforts to ensure compatibility, impacting project timelines.

Sufficient Network Connectivity:

The system assumes that users will have reliable internet connectivity for syncing habits, accessing AI recommendations, and storing progress data. Poor network conditions could affect the app's usability and response times, particularly in rural or underdeveloped areas.

Compliance with Data Protection Regulations:

It is assumed that the legal and regulatory environments (such as DPDP) will remain consistent throughout the development and operation of the app. Significant changes in data protection laws could necessitate redesigns to ensure compliance, affecting both development time and cost.

Availability of Skilled Development Team:

The project assumes the availability of a skilled development team familiar with the chosen technologies (e.g., React, Node.js, MongoDB, Express, HTML, CSS and JS) and design methods (e.g., COMET). Any delay in securing the necessary expertise could impact the project timeline and quality of the final product.

Dependencies:

Third-Party APIs:

The system is dependent on the availability and performance of the specific LLM API for chatbot functionality. Any changes in API pricing, terms of service, or technical specifications could affect the project's feasibility and cost.

Dependency on third-party authentication services (e.g., Google) for user login. If these services change their API or discontinue support, the authentication mechanism would need to be reworked.

Legal and Regulatory Compliance:

The system's design is dependent on current data protection regulations (e.g., GDPR, DPDP). Any change in these regulations would require updates to how data is stored, processed, and shared, potentially leading to redesigns and additional costs.

Hardware and Software Standards:

The system relies on the assumption that users will have access to devices that support the latest software and hardware standards. This dependency could affect the design if a significant portion of the target audience uses outdated technology.

Specific Requirements

External Interface Requirements

■ User Interfaces

❖ **Responsive UI for Web and Mobile**: The interface must dynamically adjust based on device screen size, ensuring seamless usability across desktops, tablets, and smartphones.

❖ **Intuitive Navigation and Interactive Elements**: The UI should be user-friendly with easy-to-access menus, buttons, and shortcuts for habit tracking, AI interactions, and goal setting.

❖ **Dashboard Visualization**: A user dashboard that provides real-time habit-tracking analytics, goal progress, and AI recommendations.

■ Software Interfaces

❖ **MongoDB for Data Storage**:

➢ Stores user profiles, habit logs, progress tracking, and chatbot interactions.
➢ Implements indexing for faster queries and retrieval.
➢ Uses encryption to secure sensitive user data.
➢ Supports replication for backup and disaster recovery.

❖ **LLM API for Chatbot Responses**:

➢ The chatbot provides habit suggestions, motivational messages, and personalized insights based on user behavior.
➢ Supports NLP (Natural Language Processing) \for interactive conversations.
➢ Ensures API rate limits are handled to prevent overuse.

❖ **OAuth API for Secure Authentication**:

➢ Supports third-party login via Google OAuth and other social platforms.
➢ Uses JWT-based authentication for secure session management.
➢ Implements Multi-Factor Authentication (MFA) for enhanced security.

❖ **WebSockets for Real-Time Updates**:

➢ Enables instant updates for habit-tracking progress and chatbot interactions.
➢ Supports bidirectional communication between the client and server.

➢ Functional Requirements

F1: User Authentication

- Secure sign-up/login with email or third-party providers.
- Password encryption and session management.

F2: Habit and Goal Tracking

- Create, update, delete, and visualize habits.
- Set reminders and progress-based notifications.
- Interactive user interface that visualizes progress

F3: AI Chatbot Integration

- Provides habit-building suggestions and feedback.
- Answers any queries the user may have
- Create structured plans for user specific goals that can be modified based on user preference

F4: Pomodoro timer

- Defined timers for tasks and

F5: Speech-to-Text Input

- Converts voice commands into habits and tasks.
- Integrates with a chatbot for easier interaction.

F7: Note taking

- Allows users to jot down insights.
- Supports rich text formatting.

F8: Motivational Quotes and loading time tips

- Displays motivational quotes on every application cold start and loading time tips while the user is waiting for a response from the AI agent.

■ Modules Used:

- ❖ **NLP Module:** Processes user input (text or speech) and extracts meaning to generate appropriate responses.

- ❖ **Speech Recognition Module:** Converts spoken user input into text. Can be implemented using a Google speech engine.

- ❖ **UI Module:** Provides an interactive user experience for habit tracking. It will include React.js(Frontend), TailwindCSS for styling, interactive chatbot AI

- **Backend Logic/Service Module:** Implements core logic such as managing conversations, processing responses, and handling requests. Can be implemented Node.js for backend and API endpoints for habit tracking & AI chatbot.

- **Authentication Module:** Manages user authentication and login processes. Can be implemented via Google's OAuth authentication as well as by storing the user's Sign Up credentials.

- **Data Storage Module:** Stores user data, habits logs, conversation history and insights. It will include MongoDB for habit logs and analytics. locally stored, session data will be encrypted using an asymmetric encryption algorithm.

- **Security Module:** Ensures the system's security by handling data encryption, secure authentication, and also by ensuring data integrity. The data security can be ensured by using encryption algorithms and secure SSL/TLS certificates for communication. As for the part of data integrity, we can use data encryption to prevent man-in-the-middle attacks.

- **Habit and GoalTracking Module:** It will manage habit creation, tracking and completion. Where users will be able to add, edit, delete and track habits.

Coupling between the different modules:

- **UI Module and Backend Logic Module: Data Coupling**
  Reason: The UI module sends user actions (such as habit tracking inputs) to the backend via API requests. The backend processes these requests and returns relevant responses. Since only necessary data (habit logs, analytics) is exchanged, this interaction represents data coupling.

- **Speech Recognition Module and NLP Module: Data Coupling**
  Reason: The Speech Recognition module converts spoken input into text and passes this text to the NLP module. Since only the necessary transcribed text is exchanged, this represents data coupling.

- **NLP Module and Backend Logic Module: Data Coupling**
  Reason: The NLP module processes user input and sends structured data (like extracted intent or user queries) to the backend for response generation. As only essential processed data is exchanged, this interaction involves data coupling.

- **Backend Logic Module and Data Storage Module: Data Coupling**
  Reason: The backend logic interacts with the database to store and retrieve habit logs, user preferences, and analytics. Since only necessary structured data (habit tracking details, AI interactions) is passed, this represents data coupling.

- **Backend Logic Module and Authentication Module: Control Coupling**
  Reason: The backend controls authentication flow by verifying user login status before processing habit tracking or chatbot requests. If authentication status (tokens) determines the backend's actions, this represents control coupling.

- **Authentication Module and Security Module: Common Coupling**
  Reason: The Security module handles encryption, token validation, and secure communication. Since both authentication and backend logic share security mechanisms (such as encrypted tokens or SSL/TLS protocols), they exhibit common coupling.

- **NLP Module and Security Module: Data Coupling**
  Reason: The NLP module may pass processed text data to the Security module for validation (e.g., filtering sensitive information). Since only necessary processed data is exchanged, this represents data coupling.

- **Speech Recognition Module and Backend Logic Module: Data Coupling**
  Reason: The Speech Recognition module sends transcribed text data to the backend for further processing (habit logging, chatbot interaction). As only essential text data is exchanged, this interaction involves data coupling.

- **Backend Logic Module and Security Module: Data Coupling**
  Reason: The backend logic depends on the Security module for user authentication, data encryption, and validation. Since only relevant authentication tokens and encrypted data are exchanged, this represents data coupling.

- **Habit and Goal Tracking Module and Backend Logic Module: Data Coupling**
  Reason: The Habit and Goal Tracking module communicates with the backend to create, update, and retrieve user habits. Since only structured habit-related data is exchanged, this represents data coupling.

Other Non-functional Requirements

➢ Performance Requirements

- AI chatbot responses and habit tracking updates must be responsive in order to ensure that consumers get the information they need in a timely manner. This shows how responsive and quick the website loads, allowing users to log their habits, monitor their progress, and get insights from the AI assistant in only a few seconds.

- The platform can quickly access and evaluate user behavior data and provide real-time feedback thanks to efficient data processing and retrieval. This enables users to make well-informed decisions to maintain consistency with their goals by providing them with timely, accurate recommendations based on their progress.

➢ Safety and Security Requirements

- To protect user data, the platform employs encryption, which means that the information is transformed into a code that only authorized users can read. Additionally, we follow industry best practices for data protection, making sure that the information is handled and stored in the safest manner possible, maintaining its confidentiality, and guarding against unwanted access.

- In addition, every communication between the application, the server, and the chatbot is encrypted. This implies that information is sent and received by the app in a method that makes it impossible for third parties to intercept or alter the data, guaranteeing its security and privacy.

➢ Software Quality Attributes

- **Reliability -**
  Our goal is to make sure the habit tracker is reliable and accessible whenever customers need it. We are doing this by putting in place backup systems and automatic recovery in the event of an outage. The system will make use of cloud services that can support many users at once without experiencing performance problems. Frequent testing and monitoring will guarantee a 99.9% uptime for smooth habit tracking and AI interactions by assisting in the early detection and resolution of possible problems.
- **Usability -**
  Making the habit tracker user-friendly and entertaining is our aim. The platform will have an easy-to-use UI that is simple to use on both desktop and mobile devices. The experience will be improved through ongoing user feedback. In order to guarantee inclusivity, accessibility elements will also be included, adhering to accepted accessibility standards to provide accommodations for persons with disabilities.

Other Requirements

Database Requirements

- **Data Security and Privacy**: Sensitive user information must be encrypted in the database, both in transit and at rest. The security and privacy of user data must be guaranteed by adherence to GDPR and other applicable data protection laws. Strong encryption techniques, safe access restrictions, and frequent audits will be implemented to prevent unwanted access to user credentials and habit-tracking data.

- **Scalability**: The database should be designed to handle growth in the number of users and the volume of data, with options for horizontal and vertical scaling as needed.
- **Backup and Recovery**: Regular automated backups must be scheduled to prevent data loss. The system should support point-in-time recovery to minimize downtime in case of data corruption or system failure.

Reuse Objectives

- **Modular Architecture**: The system will be designed with a modular architecture to allow for the reuse of components in future projects. This includes reusable code modules for common functionalities such as authentication, data encryption, and API integrations.

Internationalization Requirements

- **Language Support**: The application should support multiple languages, starting with English, Spanish, and Mandarin. The system should allow easy addition of new languages as needed.
- **Date and Time Formats**: The system must adapt to different regional formats for displaying dates, times, and currencies based on the user's location or preferences.
- **Localization**: The app should also account for regional differences in medical practices and units of measurement (e.g., metric vs. imperial).

Legal Requirements

- **Compliance with Data Protection Regulations**: The system must comply with relevant healthcare laws and regulations, such as HIPAA in the United States and GDPR in the European Union. This includes data protection, user consent, and the right to access or delete personal data.
- **Licensing**: All third-party libraries and software components used in the project must be properly licensed, and the project must ensure compliance with open-source licensing terms.
- **Data Retention**: The application must have a clear data retention policy that aligns with legal requirements, specifying how long user data will be stored and the process for securely deleting it when no longer needed.