The Nest Habit Tracker

Software Requirements Specification

1.0

3/4/2025

Matt Bennett, Abigail Booker, Bosit Akhmadjonov

Lead Software Engineer

Prepared for

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>
4/23/2025	Version 1.0	Matthew Bennett, Abby Booker	

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
Matthew Bennett	Matthew Bennett	Systems Engineer	4/23/2025
Abigail Booker	Abigail Booker	Test Engineer	4/23/2025
Akhmadjonov Abdulbosit	Akhmadjonov Abdulbosit	Software Engineer	04/23/2025

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirement Specification document is to define the functional and non-functional requirements for The Nest, an interactive calendar habit tracking system. This document serves as a reference for software engineers, project managers, and stakeholders to ensure the proper design, implementation, and deployment of the system.

1.2 Scope

The Nest is designed to enhance user wellness and productivity by allowing users to create, track, and manage habits efficiently. The system provides featured such as:

- User authentication: Secure login/logouts
- Calendar navigation: Interactive calendar with navigation between months
- Habit Management: Adding, deleting, editing and tracking habits
- Data storage and syncing: Firestore based storage with real time updates
- UI/UX performance: A seamless interface for users

The Nest is designed as a comprehensive motivational system focused on enhancing user wellness. The website will allow users to enhance their wellness through habit tracking and goal accomplishment. The website will provide an interactive calendar where users can set, edit, and achieve development goals. The key goals include:

- Customizable Habit Tracking: Users can define and track habits over time, receiving insights into their progress.
- Goal Achievement Visualization: The system will highlight completed goals, offering motivation and reinforcing positive behaviors.
- Interactive Calendar: A user-friendly interface allows seamless navigation through months, enabling easy habit tracking and progress review.

The Nest will function as a web-based platform, ensuring users can track habits and goals anytime. It will integrate features aligned with higher-level system requirements, such as secure user authentication, integration with Firestore with real-time updates, and an interactive interface for goal-setting and progress tracking. The software will be designed for scalability, allowing possible future enhancements such as social features and integration with external health apps.

1.3 Definitions, Acronyms, and Abbreviations

- Streak: Consecutive days a habit is to be completed
- Habit: a user-defined activity that is tracked regularly
- API:Application programming interface
- UI: User interface

1.4 References

- Eagle Tech Software Design Document (SWDD)
- *The Nest Test Procedures*

1.5 Overview

This SRS will provide the functional and non-functional requirements for the Nest Habit Tracker. It includes a description of the system, its functionality, performance expectations, and constraints.

2. General Description

2.1 Product Functions

The software product described in this document will offer the following key features:

User Authentication and Account Management

- Account Creation: Users can create accounts using email and password authentication through the Clerk API. Invalid email addresses or weak passwords will trigger error messages. Duplicate emails will also result in an error.
- Login and logout: Registered users can log in with valid credentials and log out successfully, redirecting to the login page post-logout. Invalid credentials will trigger error messages, while network failures will notify users of connectivity issues.
- Error Handling: The system will display appropriate error messages for incorrect credentials or network failures.

Calendar Navigation and Display

- Month Navigation: Users can navigate between months using clickable arrow buttons on the calendar, with month updates reflecting the correct days and abbreviations.
- Calendar Display: The calendar will display the correct day abbreviations and dates for the selected month.

Habit Management and Interaction

- Adding, Editing, and Deleting Habits: Users can add new habits, modify existing habits, and delete them with confirmation prompts. Habit limits will be enforced, allowing no more than 10 habits per month.
- Marking Completion: Users can check/uncheck habit completion, with three distinct colored checks for easy readability between habits.
- Goal Achievement Highlighting: Once the achieved count for a habit matches or exceeds the set goal, the goal box will be highlighted green.

Data Storage and Synchronization:

- Firestore Integration: Habit data, including progress, will be stored and retrieved from Firestore.
- Real-Time Updates: Habit updates will be saved and synchronized in real time with Firestore.
- Handling Firestore Downtime: In case of Firestore unavailability, users will be notified with an error message.

Error Handling and security:

- Invalid Input Handling: The system will prevent invalid entries.
- Security and Authentication: User authentication will be secure, and sessions will not be exposed to unauthorized users.
- Database Security: Only authenticated users will have permission to modify their own habit data

2.2 User Characteristics

This section outlines the characteristics of the intended users of the system, which will influence the design and functionality of the software:

User demographics:

- Age range: The software is intended for users 18 and up. The user interface will be designed to be simple and accessible for users with different levels of technical knowledge. This range will allow for a diverse user base, from young adults to older individuals.
- Technical knowledge: The target audience includes people with both limited technical skills and individuals with lots of technical experience. The application will prioritize a user-friendly design, with simple designs and intuitive navigation.
- Language proficiency: The application will be based in English.

User needs and requirements:

- Ease of use
- Flexibility
- Reliability
- Security

2.3 General Constraints

This subsection of the SRS deals with the limitations that are imposed by the hardware, software, or any other factors that restrict design options.

- The system must operate as a web-based application and requires a modern web browser.
- The application depends on continuous access to Firestore and the Clerk API
- Users must have a stable internet connection for real-time data to sync.
- The system is developed using Firestore and Clerk API, limiting the ability to have flexibility in switching technologies without major reconstruction.
- Maximum of 10 habits per user per month.

2.4 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

- Users have consistent internet access for cloud features
- Daily reset check-ins at 12:00 AM local time
- All account management functions within Clerk API are functional

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 The user interface will allow account creation and login through Clerk API
3.1.1.2 A navigation bar will be displayed at the top of the page with clickable arrows (left and right) for month navigation
3.1.1.3 An interactive calendar will display habits with checkable boxes corresponding to each day
3.1.1.3.1 There will be three separate colors of checkmarks to increase readability for the user for completed tasks (green, blue, and yellow)
3.1.1.4 A clickable new habit button will be centered below the calendar

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.3.1 Integration with Clerk API for user authentication
3.1.3.2 Integrates with the Firestore Database for habit tracking and storage

3.1.4 Communications Interfaces

3.1.4.1 Real-time communication with Firestore for data storage and retrieval

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 User Authentication

3.2.1.1 Introduction

Users must be able to create an account or log in to their account using their email and password

3.2.1.2 Inputs

- Email
- Password

3.2.1.3 Processing

- Clerk API for both account creation and login validation

3.2.1.4 Outputs

- Success message, and then enter into the users dashboard
- An error message

3.2.1.5 Error Handling

- Invalid information entered will prompt an error message

3.2.2 Calendar Navigation

3.2.2.1 Introduction

Users can navigate through months using clickable arrows

3.2.2.2 Inputs

- Click on left/right arrows

3.2.2.3 Processing

- Update calendar view to previous/next month

3.2.2.4 Outputs

- Updated Calendar display

3.2.2.5 Error Handling

3.2.3 Habit Management

3.2.3.1 Add New Habit

3.2.3.1.1 Introduction

Users must have the ability to add up to 10 habits

3.2.3.1.2 Inputs

- Habit name
- Goals (wanted habit completion number)
- Save button
- Cancel button

3.2.3.1.3 Outputs

- A new habit is added with a goal amount
- Error message

3.2.3.1.4 Error Handling

- If the goal set is greater than days of the month than an error message is displayed

3.2.3.2 Edit Habit

3.2.3.2.1 Introduction

Users must be able to hover over the name of the habit in the dashboard and be shown an edit symbol to click.

3.2.3.2.2 Inputs

- Habit Name
- Goals
- Save button
- Cancel button

3.2.3.2.3 Outputs

- New updated habit is added with goal amount
- Error message

3.2.3.2.4 Error Handling

- If the goal set is greater than days of the month than an error message is displayed

3.2.3.2 Delete Habit

3.2.3.2.1 Introduction

Users must be able to hover over the name of the habit in the dashboard and be shown a trash can icon to click. The user will then be prompted with a pop-up to make sure they want to delete it.

3.2.3.2.2 Inputs

- Delete
- Confirm delete
- Cancel button

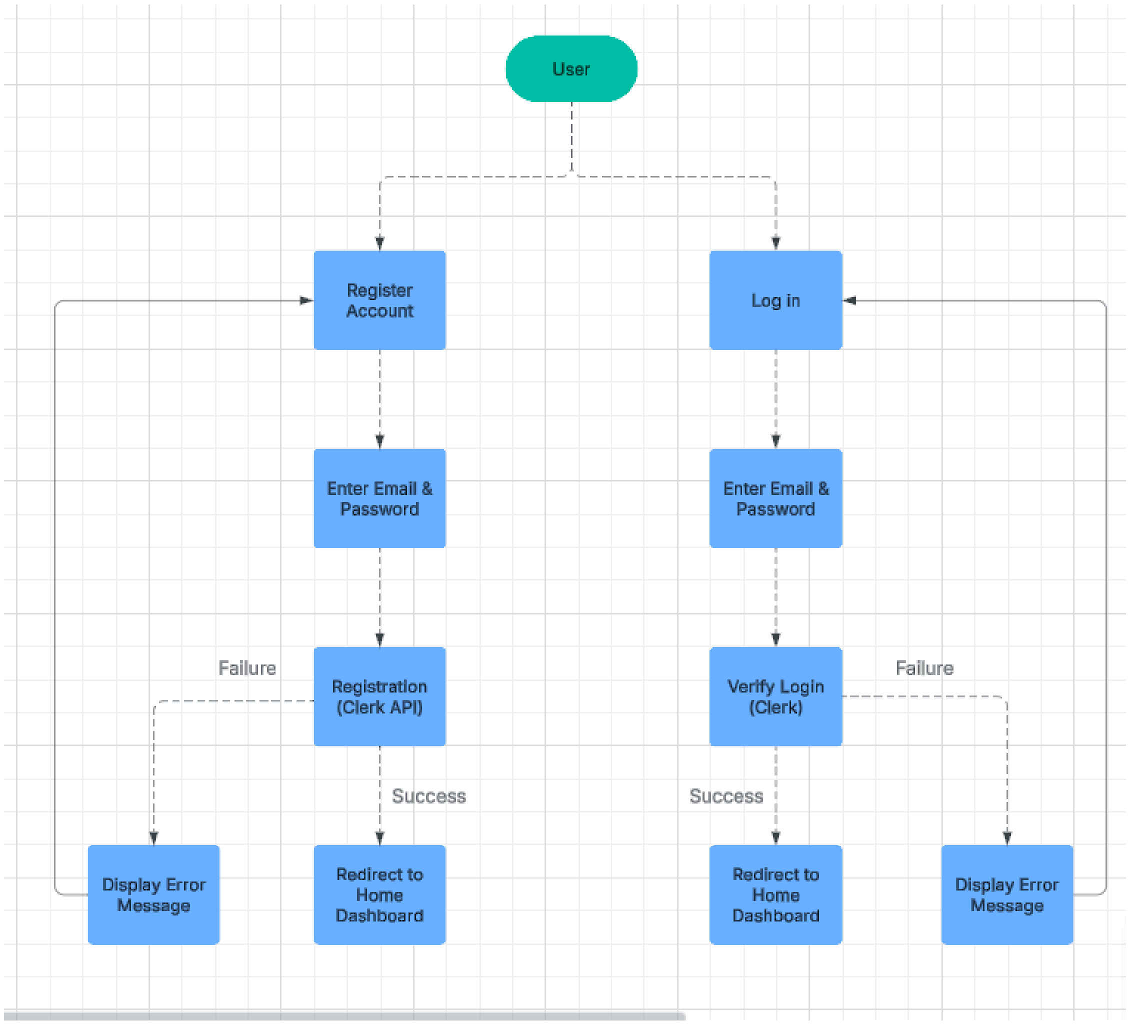
3.2.3.2.3 Outputs

- Habit is deleted

3.3 Use Cases

3.3.1 Use Case #1

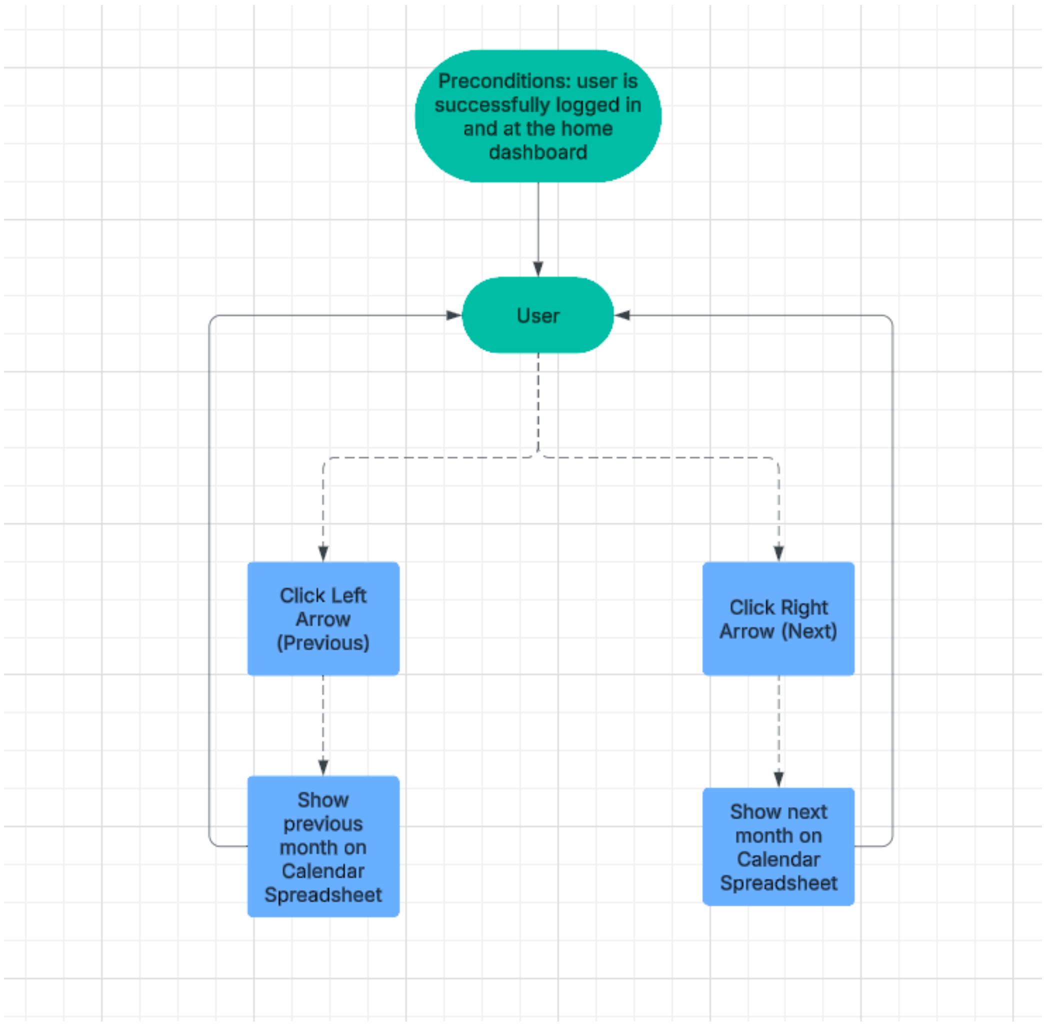
Use case: Registration & login	
Actor	User
Preconditions	The system is on and running. User has navigated onto the Nest website.
Flow	1. User navigates to the Nest website 2. User is directed to the Nest log in page 3. If they don’t have an account, they click on the "Sign Up" button. 4. The system prompts the user to enter an email address and password. 5. User enters email and strong password. 6. The system processes the registration with Clerk API and stores the user’s data. 7. If successful, the user is redirected to the dashboard. 8. If unsuccessful, an error message is displayed.
Alternative flow:	● If the user already has an account, they log in by entering their email and password, which is verified using Clerk API. ● Upon successful login, the user is redirected to the dashboard.



3.3.2 Use Case #2

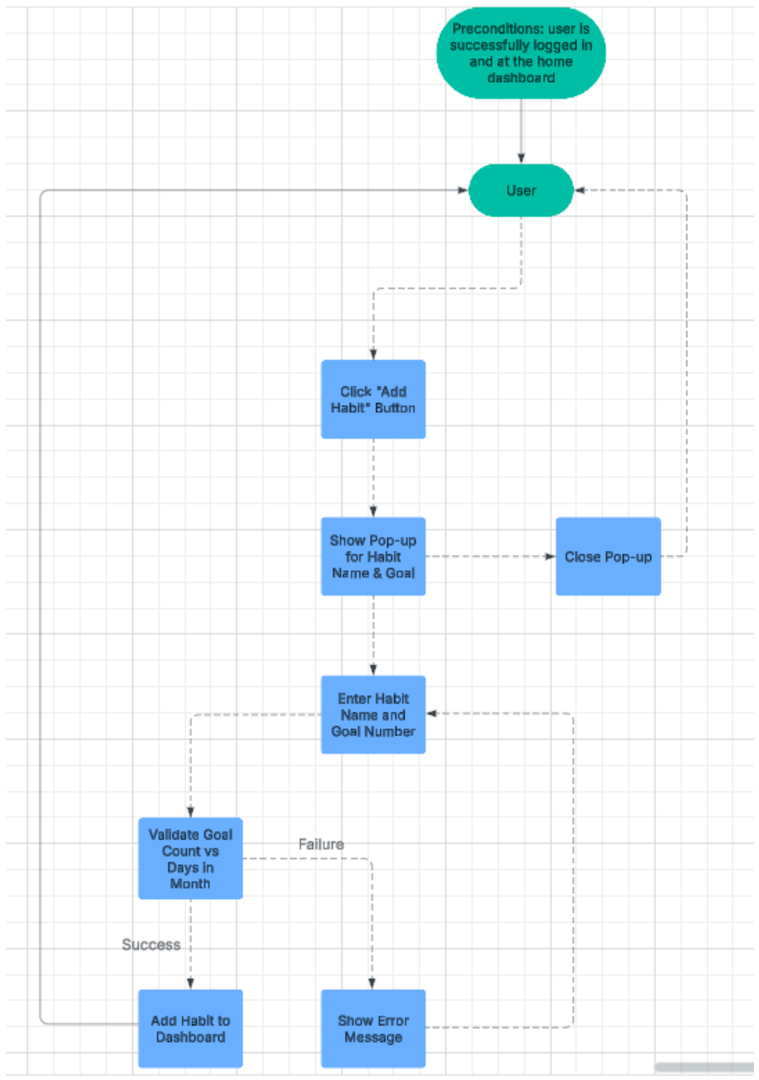
Use case: Navigating between months	
Actor	User
Preconditions	The system is on and running. User has navigated onto the Nest website. User has logged in.

Flow	1. User is shown the home screen after logging in 2. The user clicks the arrow pointing to the right from the month 3. The interactable calendar spreadsheet will show the month immediately following the current one
Alternative flow:	1. User is shown the home screen after logging in 2. The user clicks the arrow pointing to the left from the month 3. The interactable calendar spreadsheet will show the month immediately preceding the current one


Preconditions: user is successfully logged in and at the home dashboard
User
Click Left Arrow (Previous)
Click Right Arrow (Next)
Show previous month on Calendar Spreadsheet
Show next month on Calendar Spreadsheet

3.3.3 Use Case #3

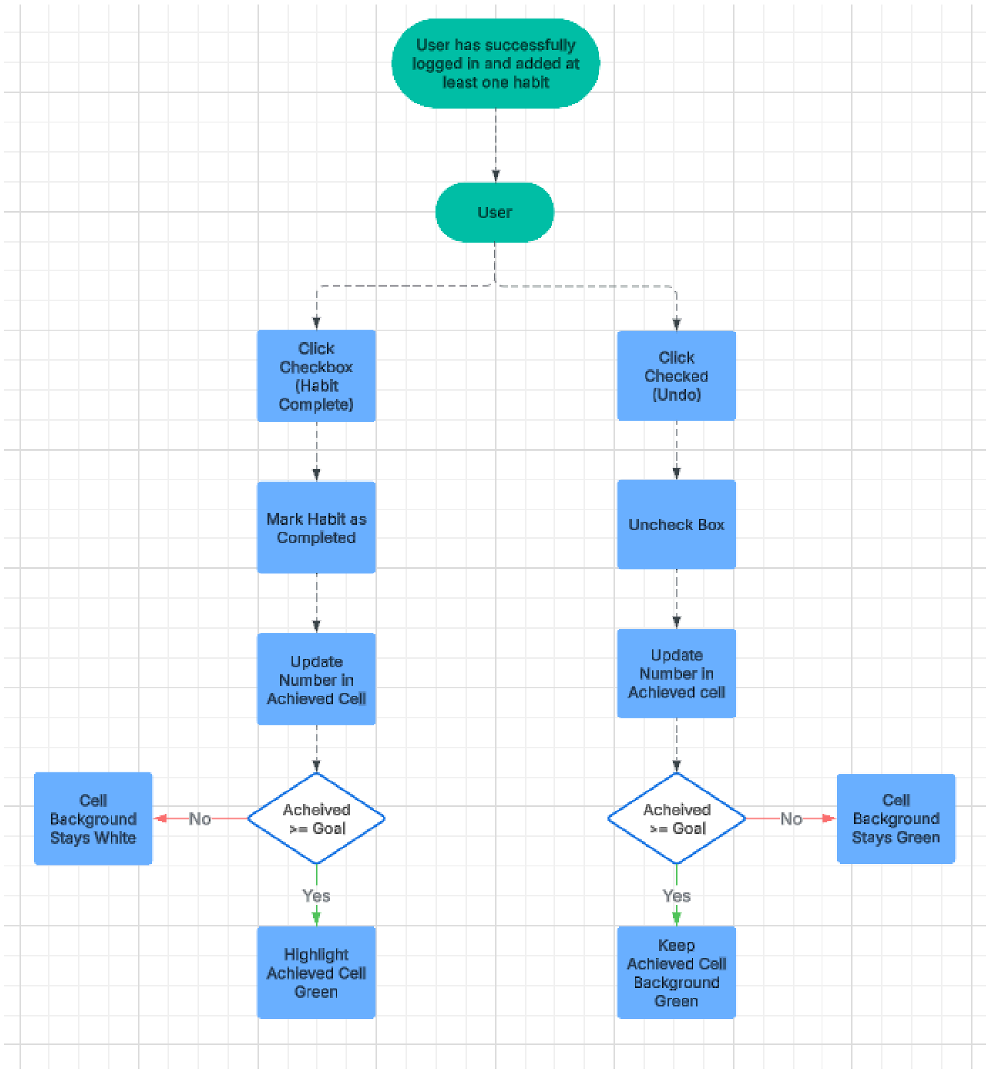
Use case: Adding a new habit	
Actor	User
Preconditions	The system is on and running. User has navigated onto the Nest website. User has logged in. User is in the desired month spreadsheet.
Flow	1. The user clicks the add habit button 2. The user is shown a pop up window asking for the name of the habit and the number of times they want to achieve that goal for that month 3. The information is entered for both the habit name and goal 4. If the number of times the user wants to achieve that goal is less than the number of days in the selected month, the habit is added to the calendar 5. If not, an error message is displayed
Alternative flow:	1. The user clicks the add habit button 2. The user is shown a pop up window asking for the name of the habit and the number of times they want to achieve that goal for that month 3. The user changes their mind and closes out the window


Preconditions: user is successfully logged in and at the home dashboard
User
Click "Add Habit" Button
Show Pop-up for Habit Name & Goal
Close Pop-up
Enter Habit Name and Goal Number
Validate Goal Count vs Days in Month
Failure
Success
Add Habit to Dashboard
Show Error Message

3.3.4 Use Case #4

Use case: Checking Off a Completed Habit	
Actor:	User
Preconditions:	The system is on and running. User has navigated onto the Nest website. User has logged in. User is in the desired month spreadsheet, and already has a habit added.

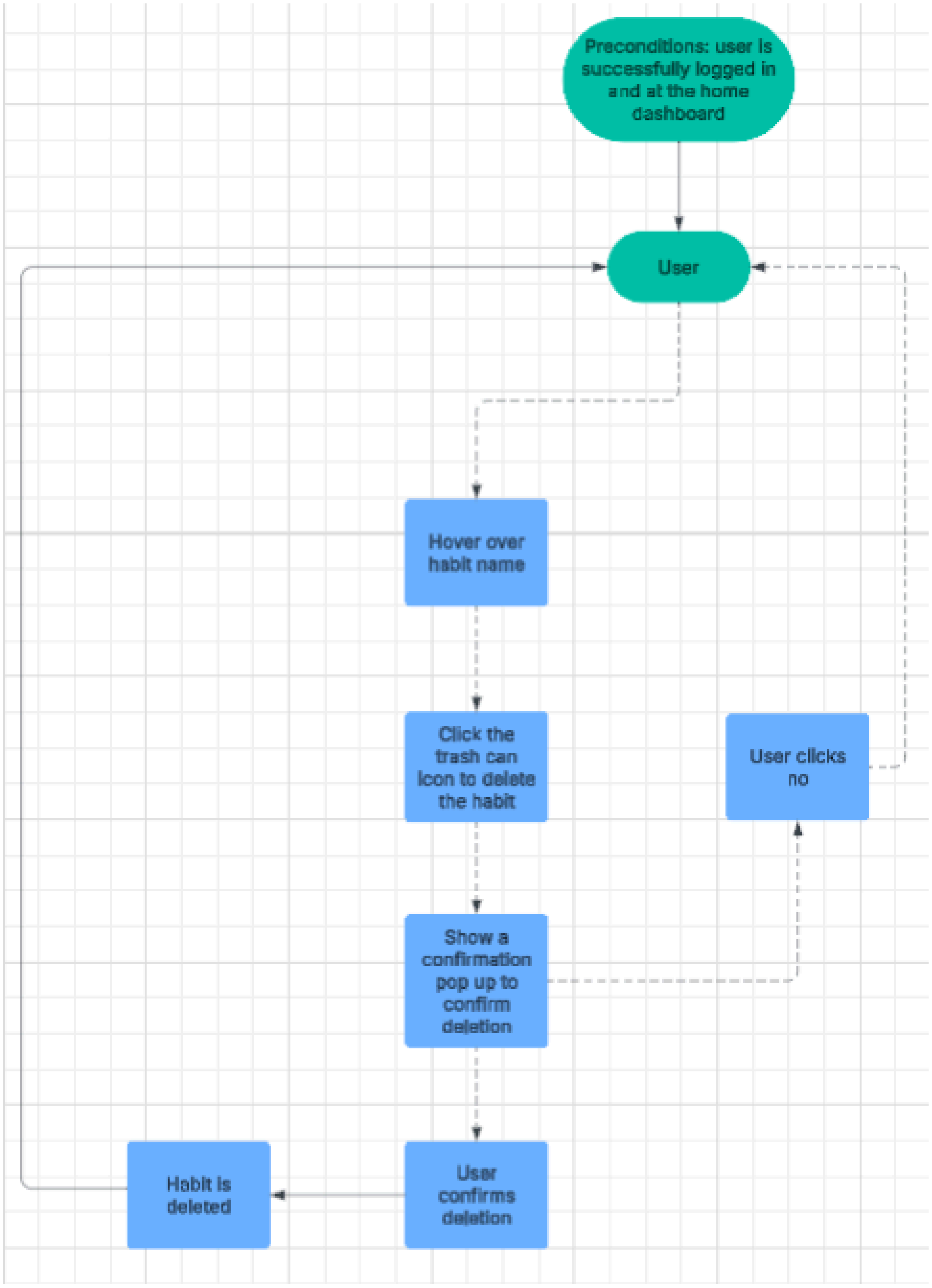
Flow:	1. The user clicks the checkbox of the day the habit was completed 2. The checkbox is then checked, symbolizing that day the habit was completed 3. The amount of completions is compared to the goal set, and will update the number in the achieved cell 4. If the value of habits completed is at or greater than the goal, then the achieved cell will be highlighted green, indicating that the goal has been reached 5. If not, the number achieved will simply update
Alternative flow:	1. The user clicks the already checked box 2. The box is unchecked, symbolizing the habit was not completed 3. The amount of completions is compared to the goal set, and will update the number in the achieved cell 4. If the value of habits completed drops below the goal, then the achieved box will turn back white, indicating the goal is no longer achieved


User has successfully logged in and added at least one habit
User
Click Checkbox (Habit Complete)
Mark Habit as Completed
Update Number in Achieved Cell
Acheived >= Goal
No
Cell Background Stays White
Yes
Highlight Achieved Cell Green
Click Checked (Undo)
Uncheck Box
Update Number in Achieved cell
Acheived >= Goal
No
Cell Background Stays Green
Yes
Keep Achieved Cell Background Green

3.3.5 Use Case #5

Use case:Deleting a habit	
Actor	User
Preconditions	The system is on and running. User has navigated onto the Nest website. User has logged in. User is in the desired month spreadsheet. User has at least one existing habit

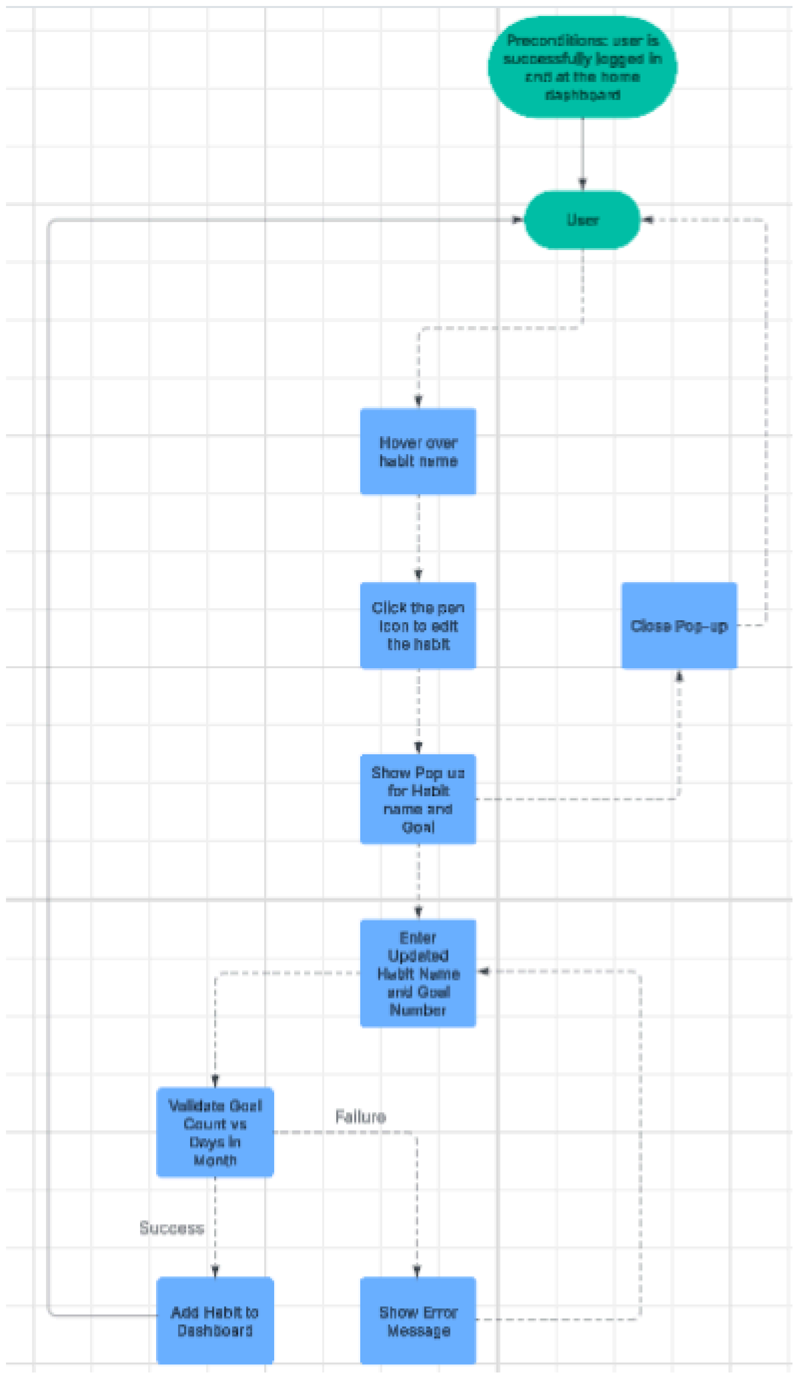
Flow	1. The user hovers over habit 2. When hovering, a clickable trash icon appears 3. The user clicks the icon 4. A pop up message is displayed and prompts them to verify deletion 5. The user clicks confirm 6. The habit is deleted
Alternative flow:	4. The user clicks the trash icon 5. The user is shown a pop up window asking if they want to delete it 6. The user changes their mind and clicks the cancel button


Preconditions: user is successfully logged in and at the home dashboard
User
Hover over habit name
Click the trash can icon to delete the habit
User clicks no
Show a confirmation pop up to confirm deletion
Habit is deleted
User confirms deletion

3.3.6 Use Case #6

Use case: Edit habit	
Actor	User
Preconditions	The system is on and running. User has navigated onto the Nest website. User has at least one existing habit.

Flow	1. User navigates to the Nest website 2. User navigates to the desired month and hovers over the habit 3. The pencil icon and trash icon appear and the user clicks the pencil icon for editing 4. A popup window will appear prompting for the edited habit name and monthly goal 5. The user will edit the name and monthly goal 6. The user will click save
Alternative flow:	● The user clicks the pencil icon ● The user is shown a pop up window asking for the name and month edits ● The user changes their mind and clicks the cancel button


Preconditions: user is successfully logged in and at the home dashboard
User
Hover over habit name
Click the pen icon to edit the habit
Close Pop-up
Show Pop up for Habit name and Goal
Enter Updated Habit Name and Goal Number
Validate Goal Count vs Days in Month
Failure
Success
Add Habit to Dashboard
Show Error Message

3.4 Classes / Objects

3.4.1 User

3.4.1.1 Attributes

- Email
- Password

3.4.1.2 Functions

- CreateAccount()
- Login()

3.4.2 Habit

3.4.2.1 Attributes

- Name
- Goal
- Progress

3.4.2.2 Functions

- AddHabit()
- EditHabit()
- DeleteHabit()
- TrackProgress()

3.5 Non-Functional Requirements

3.5.1 Performance

- All interactions other than login/create account, and loading time between changing months, must process in under 2 seconds

3.5.2 Reliability

- 99.95% uptime per month for Firestone/Clerk API (Firestone aims for 99.95%)
    - 22 minutes of downtime allowed per month

3.5.3 Availability

- System downtime must not exceed 2 minutes per day

3.5.4 Security

- Secure account authentication through Clerk API

3.5.5 Maintainability

- Modular code design allowing for easy updates

3.6 Inverse Requirements

- The system will not provide health, medical or psychological advice
- The Nest will not support offline functionality
- The system will not allow users to modify or access other user’s data
- No integration with external applications in the initial release

3.7 Design Constraints

- Must use Clerk API for authentication
- Must use Firebase for data storage

3.8 Logical Database Requirements

- Firestore structure: username -> habit name -> month -> dates -> true/false
- Data retention: Habit data must be stored for at least 12 months

3.9 Other Requirements

Future enhancements will include social features and health app integration.

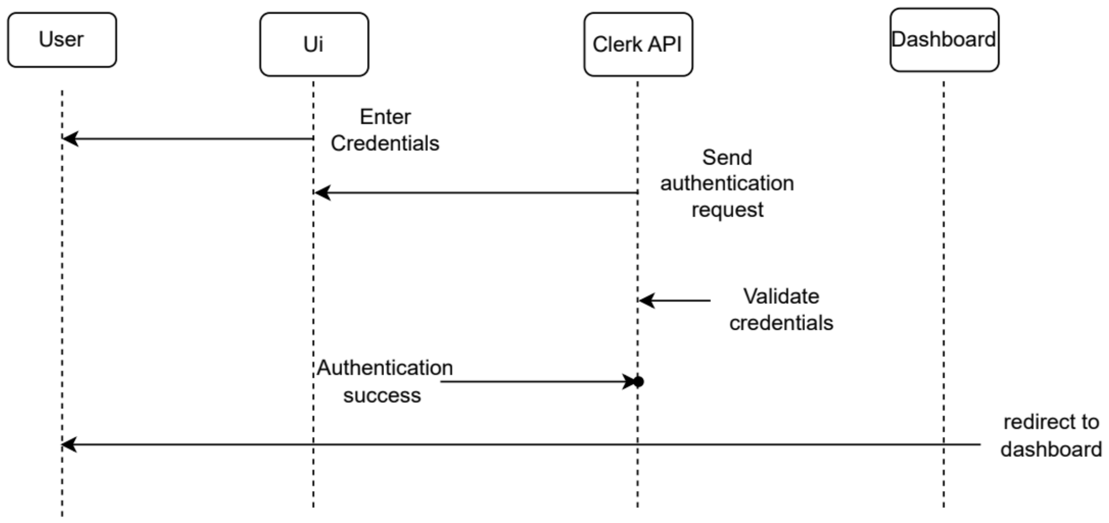
4. Analysis Models

4.1 Sequence Diagrams

The sequence diagram illustrates the interaction between users and system components over time for critical use cases:

Use Case: User Login Process
This diagram demonstrates the sequence of interactions when a user logs into The Nest platform:

1. User submits login credentials with the UI.
2. UI sends an authentication request to Clerk API.
3. Clerk API validates credentials.
4. Upon success, the UI redirects user to dashboard.

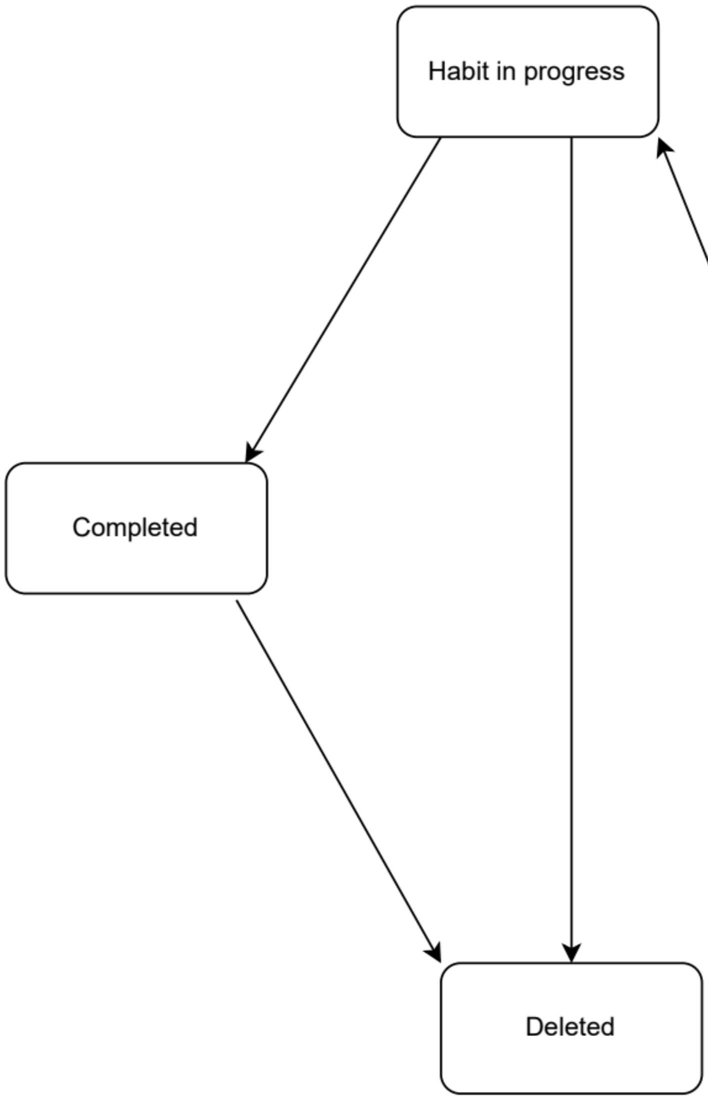


4.2 State-Transition Diagrams (STD)

State-transition diagrams represent the various states of the system or components and how they change in response to events.

The habit tracking component transitions between the following states:

- Habit Created: Initial state when a habit is created.
- Habit in progress : When at least one completion is logged.
- Completed :When the habit meets or exceeds its monthly goal.
- Deleted : If the user deletes the habit.



5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

All proposed changes to this SRS must follow a structured process to review and ensure consistency and alignment with the project goals.

Any project stakeholder can submit changes. The change request must be documented using a change request form and will be submitted to the project manager. The submission can be through email or a project management tool. The final approval will be required from both the lead software engineer and the project sponsors. Each approved change request will increment the SRS version number and update the document.

Review process:

- The project manager will log the request and schedule a meeting to review.
- The development team evaluates the impact on scope, timeline and resources
- Approved changes are versioned and will be appended to the SRS

A. Appendices

A.1 Appendix 1

References:

- IEEE Guide to SRS.
- Firestore Documentation.
- Clerk API Documentation.