

Introduction.

Purpose: The purpose of this document is to define the requirements for a command-line application that interacts with the user about their daily mood, sleep quality, activities, and additional thoughts. This application will be built in C/C++ and will not use any external APIs apart from those built into C/C++. It will run in the command line on Windows.

Scope: The application will allow users to log daily mood, sleep quality, activities, and additional thoughts. It will analyze and correlate this data to provide insights into how activities and sleep quality affect mood.

General Description.

Product perspective: This application is a standalone tool that helps users track and analyze their daily mood, sleep quality, and activities.

Product functions:

- Interact with users through a series of questions.
- Record user responses about mood, sleep quality, activities, and additional thoughts.
- Analyze data to identify patterns between activities, sleep quality, and mood.
- Save journal entries to local device

User characteristics: The target users are individuals who are interested in understanding how their sleep and activity affects their mood and want an interactive way to wind down and reflect at the end of their day.

Constraints:

- The application will run on Windows command-line.
- No use of third-party libraries; use only standard libraries of C or C++.

Specific requirements.

Stakeholders

Internal: Developers (Hustling Hamsters)
External: Application users

Functional Requirements

	Motivation	Requirement
Dialogue	Questions need to be asked so that the application can collect uniform data from the user	The application <i>must</i> ask the user a series of questions

Type reply	The keyboard will be the only means of the user being able to communicate to the application	The user <i>must</i> be able to reply using keyboard
Data collection	Collecting and storing data about the users answers will allow the application to produce meaningful insights	The application <i>must</i> be able to store the users answers
Daily journal	It may be useful for the user to be able to see their daily journal that includes all of their answers from that day	The application <i>must</i> be able to compile the users daily answers into a single file
Save daily journal	The user may want to keep their own folder of their daily journal entries	The application <i>must</i> provide an option to export daily journal to local device
Statistics and comparisons	Collection of data should serve a purpose. Displaying what sleep quality and activities affect mood add value of the application to the user	The application <i>must</i> be able to display a weekly report on how activities and sleep affect their mood
Customisation	Everyone's days are different and contain different activities, the user should be able to reflect this in the application.	The user <i>should</i> be able to add and delete their own categories for activities.
Welcome/info display	The application will be unknown to new users and a command line interface may not be as intuitive as a GUI	The application <i>should</i> have a page entailing how to use the application and a list of possible commands the user can use
User ID/log in	The application may need to identify new users or multiple users may share a device	The application <i>may</i> take a username and password to identify users

Non-functional requirements

- The **application** *should* use encouraging and positive dialogue
- The **application** *may* use a Hamster theme throughout

Use cases

Actors: User (used in all use cases)

Use Case 1: Opening menu

Goal: The application directs the user to the part of the application the user would like to use

Main Scenario:

1. The application prompts the user "Welcome to the Hangout, so good to see you! What would you like to do today? [0] Welcome page [1] Settings [2] Data Report [3] Daily Logging"
2. The user enters the integer corresponding to menu
3. If use enters 0:
 - a. Use is directed to welcome page
4. If user enters 1:
 - a. User is directed to settings menu
5. If user eneter 2:
 - a. User is directed to reports menu
6. If user enters 3:
 - a. User is directed to journaling * Triggers Use Case 2

Use Case 2: Record sleep quality

Goal: The application learns users sleep quality

Main Scenario:

1. The application prompts the user "How did you sleep last night? [1] - Terrible, [2] - Not great, [3] - Average, [4], Pretty good, [5] Fantastic"
2. The user enters the integer corresponding to their sleep quality
3. The application records the response
4. * Triggers Use Case 4

Use Case 3: Record daily activity

Goal: The application learns the users activity from the day

Main scenario:

1. The application prompts the user "Did you complete *activity* today? [Y/N]"
2. The user answers Y/N
3. The application prompts the same question with all activities in activity list
4. The user answer Y/N to all
5. The application prompts "Did you do any other activities not in your existing list?" [Y/N]
6. User answers Y:
 - a. The application prompts "What activities did you do today? This activity will be added to your list of activities"
 - b. The user enters new activities
 - c. The application stores new activities
 - d. The application prompts "Amazing! Those activities have been added to your list. **some inspiration quote**"
 - e. The application records activities response
7. User answers N:
 - a. The application records activities response
8. Triggers Use Case 4

Use Case 4: Record daily mood

Goal: The application learns the users daily mood

Main scenario:

1. The application prompts the user "What was your general mood today? [1] - Down in the dumps :(, [2] - A bit flat :/ [3] - Calm : | [4] - Pretty good :), [5] - Felt amazing! :))))))"
2. The user enters the integer corresponding to their mood
3. The application records their response
4. *Triggers Use Case 5

Use Case 5: Record additional thoughts

Goal: The application gathers additional thoughts

Main scenario:

1. The application prompts the user "Any additional thoughts on your day? (max 250 character limit)"
2. The user responds with text
3. The application stores their response
4. *Triggers Use Case 6

Use Case 6: End of Conversation

Goal: Consolidate and display daily journal

Main Scenario:

1. The application prompts " Well done on another day of journaling! Here's how your day went..."
2. The application retrieves answers from today's journal prompts
3. The application displays all of the users answers as a daily journal
4. The application prompts "Do you want to save this daily journal? [Y/N]"
5. If user answers Y:
 - a. The application saves daily journal to users computer
 - b. The application prompts "That's all for today, I hope to talk to you tomorrow!"
6. If user answers N:
 - a. The application prompts "That's all for today, I hope to talk to you tomorrow!"