# Project Proposal

## Project Introduction

*MindfulMood is a simple and intuitive mood diary and tracking app. By tracking your mood and thoughts throughout the day, MindfulMood aims to provide insights into one’s mood and help discover one’s mindfulness.*

There are many similar apps already on the Android Market place, however, a lot of these apps are full of extra features and cause them to be difficult to use and understand. Additionally, most of them have the drawback of only being able to record your mood as an overall summary of the day. This project proposal outlines an approach that simplifies the experience and removes the unnecessary complexity compared to other apps.

A person’s mood can change throughout the day. MindfulMood helps the user track these changes by being able to record your mood multiple times a day. To keep it simple, the user’s mood is tracked on a five-point scale, 1 being to the worst and 5 being the best. When a user rates their mood they can then add a description to help identify factors of cause.

There are many factors they can influence a person’s mood. Once the user has regular input tracking of their mood, the graph visualisation feature will enhance their experience. The user will be able to view how their mood has fluctuated over the day, week, and month. The user will be able to see notable events about the data points as to help explain why they felt a certain way. The trends will become valuable to support the user with improving control over their mood and increase their mindfulness.

To help get the user in a routine of recording their mood, the app will be customisable with reminder notifications. Reminders can be scheduled for when the user has missing data, such as not recording when they woke up, and scheduled at the end of their day asking the user how their day was. More proactive, or more forgetful people, may choose to schedule reminders every hour.

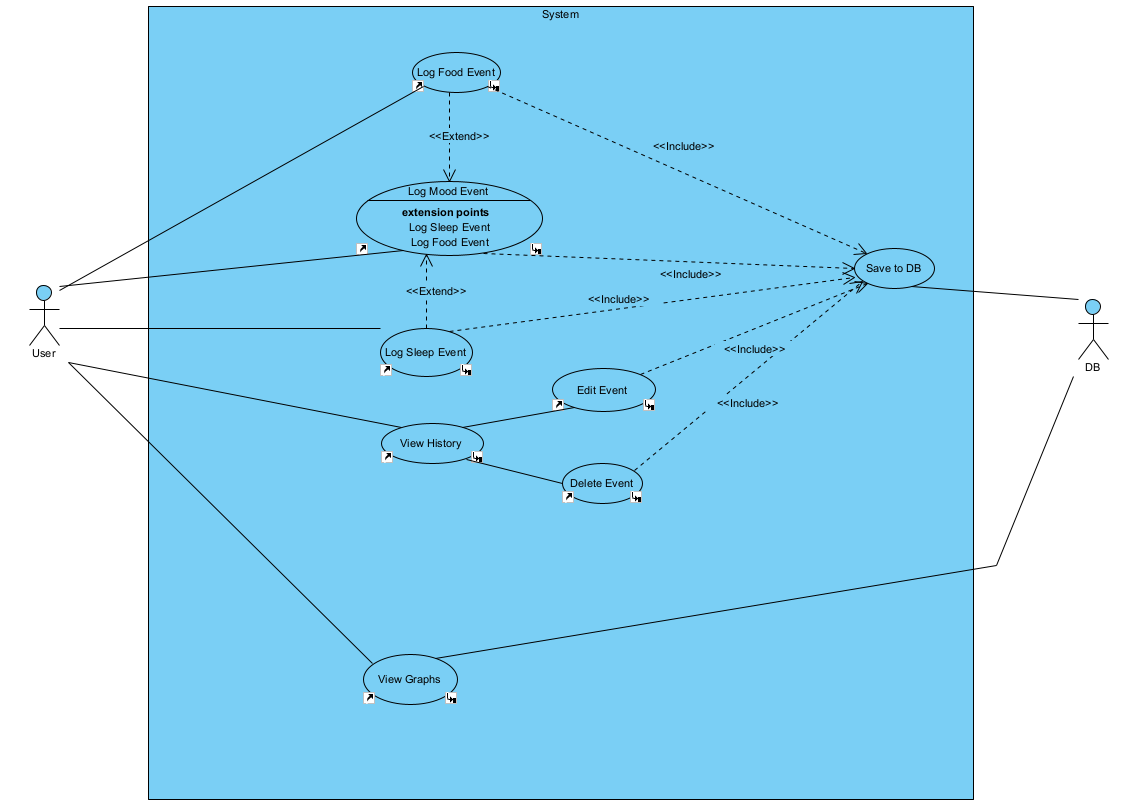
Initially, they app will support events such as when a person eats and sleeps. When a person’s blood sugar drops, they may become tired, irritable, and depressed. Knowing if the user has eaten can help explain and improve insights for why certain situations made them feel high or low. The same is also relevant with sleep. Either too little or too much can have negative effects on a person’s mood.

In the future, the app can be expanded to include a menstrual cycle and medication calendar. As these events fluctuate a person’s hormones and mood, they user can become more aware of how they might be feeling on those days and take action to support their mood.

## Project Design

The app will be developed for android devices in mind, using the java language. This is partly because Android has the largest share of smartphone users globally as compared to Apple. Because of this, Android has a large number of beta testers compared to the platforms, meaning bugs are fixed before the stable version of an app is released. Android also supports the java language, which I am familiar coding with. To create an app for Apple, I would need to be familiar with Objective-C and Swift languages.

### Use Cases Diagram:



# Full Use Case Descriptions:

# Delete Event

## Brief Description

User End-Goal Story

*When the user*

*Wants to remove an event*

*They select the event to delete*

*So that the data can be removed from the database*

## Triggers

User

*The user has noticed duplicate data.*

*The user wishes to delete an event.*

## Scenarios

### Normal Flow

1. User wants to delete an event

2. SYSTEM Displays date range available

3. User selects the day of the event

4. SYSTEM Displays all the events of that day

5. User selects the event they want to delete.

6. SYSTEM Prompts to confirm deletion

7. User selects response

7.1. if yes

7.1.1. SYSTEM Deletes events

7.1.2. Save to DB

end if

7.2. if no

7.2.1. exitNormal Flow

end if

## Preconditions

1) The user has noticed duplicate data.

2) The user wishes to delete an event.

## Post-conditions

Save to DB

## Assumptions

User can add mood, food, and sleep events to previous days.

User can delete mood, food, and sleep events to previous days.

User can modify mood, food, and sleep events to previous days.

# Edit Event

## Brief Description

User End-Goal Story

*The the user*

*Wants to append the data of an event*

*They edit the data of the event*

*So that their data can be more accurate*

## Triggers

User

*Wants to add more data to the event.*

*Notices data missing from an event.*

*Wants to edit the data of an event.*

## Scenarios

### Normal Flow

1. User wants to edit an event

2. SYSTEM Displays date range available

3. User selects the day of the event

4. SYSTEM Displays all the events of that day

5. User selects the event they want to edit.

6. SYSTEM Displays the data of the event

7. User appends the changes

8. SYSTEM Save to DB

## Preconditions

1) The user wants to add more data to the mood event.

2) The user notices data missing from a mood event.

3) The user wants to edit the data of a mood event.

## Post-conditions

Save to DB

## Assumptions

User can add mood, food, and sleep events to previous days.

User can delete mood, food, and sleep events to previous days.

User can modify mood, food, and sleep events to previous days.

# Log Food Event

## Brief Description

User End-Goal Story

*When the user*

*Wants to monitor their food-mood cycle*

*They record their food intake*

*So that their food-mood correlation can be graphed*

*Event-Response Story*

*When no food event has been logged AND current time is now 6PM*

*It causes the system to prompt the user to record their food intake*

*By sending a push notification*

*So that their food-mood correlation can be graphed*

## Triggers

By User

*User has wants to log their food*

By System

*The locale time is passed 6PM and their is no food logged.*

*Push Notification sent*

## Scenarios

### Normal Flow

1. User wants to log their food.

2. SYSTEM Asks if the entry relates to Breakfast, Lunch, Dinner, or Snacks

3. User has entered all food events they wish to log

4. User chooses either Breakfast, Lunch, Dinner, or Snacks

5. SYSTEM Ask what time the food was eaten.

6. User enters the time.

7. SYSTEM Save to DB

8. SYSTEM Asks for a description/notes of what was eaten.

9. User enters notes.

10. SYSTEM Save to DB

11. SYSTEM Prompts to record mood.

12. User Log Mood Event

### Alternate Flow

1. User wants to log their food.

2. SYSTEM Asks if the entry relates to Breakfast, Lunch, Dinner, or Snacks

3. User has entered all food events they wish to log

4. User chooses either Breakfast, Lunch, Dinner, or Snacks

5. SYSTEM Ask what time the food was eaten.

6. Enters the time.

7. SYSTEM Save to DB

8. SYSTEM Asks for a description/notes of what was eaten.

9. User ignores

### Exception Flow

1. User wants to log their food

2. SYSTEM Asks if the entry relates to Breakfast, Lunch, Dinner, or Snacks

3. User ignores.

4. SYSTEM Systems discards food event.

## Preconditions

1) The user has eaten and wishes to log their food.

2) A push notification has reminded to the user to log their food intake from eary today.

## Post-conditions

Log Mood Event

Save to DB

## Assumptions

Logging food events are optional.

Can log multiple food events during the day.

## Requirements

Can log multiple food events during the day.

Logging food events are optional.

# Log Mood Event

## Brief Description

User End-Goal Story

*When the user*

*Wants to log their mood*

*They choose their mood on a scale and provide a description*

*So that their mood is recorded and can be viewed graphically*

## Scenarios

### Normal Flow

1. User wants to log their mood

2. SYSTEM Displays mood scale

3. User selects their mood on the scale

4. SYSTEM Saves data

5. SYSTEM Asks to enter a description

6. User enters a description related to their mood

7. SYSTEM Saves data

### Alternate Flow

1. User wants to log their mood

2. SYSTEM Displays mood scale

3. User selects their mood on the scale

4. SYSTEM Saves data

5. SYSTEM Asks to enter a description

6. User ignores

7. SYSTEM Discards description

### Exception Flow

1. User wants to log their mood

2. SYSTEM Displays mood scale

3. User ignores

4. SYSTEM Discards mood event

## Preconditions

1) User wants to log their mood.

2) User wants to log their mood associated with a Log Food Event

3) User wants to log their mood associated with a Log Sleep Event

## Post-conditions

Save to DB

## Assumptions

User can log their mood multiple times a day

## Requirements

User can log their mood multiple times a day

# Log Sleep Event

## Brief Description

User End-Goal Story

*When the user*

*Wants to monitor their sleep-mood cycle*

*They record their sleep time*

*So that their sleep-mood correlation can be graphed*

Event-Response Story

*When no sleep event has been logged AND current time is now 10AM*

*It causes the system to get the user to record their sleep time*

*By sending a push notification*

*So that their sleep-mood correlation can be graphed*

## Triggers

By User

*User has woken up and wants to log their sleep*

By System

*The locale time is passed 10AM and their is no sleep logged.*

*Push Notification sent*

## Scenarios

### Normal Flow

1. User wakes up and want to log their sleep.

2. User records the time they went to sleep and the time they woke up.

3. SYSTEM Displays the total time slept. Data Saved.

4. SYSTEM Prompts to rate sleep quality.

5. User records sleep quality.

6. SYSTEM Save to DB

7. SYSTEM Prompts to record mood.

8. User Log Mood Event

### Alternate Flow

1. User wakes up and want to log their sleep.

2. User records the time they went to sleep and the time they woke up.

3. SYSTEM Displays the total time slept.

4. SYSTEM Save to DB

5. SYSTEM Prompts to rate sleep quality.

6. User ignores.

### Exception Flow

1. User wakes up and want to log their sleep.

2. User records only the time they went to sleep.

3. SYSTEM Prompts for their wake up time.

4. User ignores.

5. SYSTEM Systems discards sleep event.

## Preconditions

1) User has not already logged a sleep event for the day.

2) It is the start of the day and the user has woken up.

## Post-conditions

Log Mood Event

Save to DB

## Assumptions

Can only log sleep once a day

## Requirements

Can only log sleep once a day

Every morning the user expects the app to remind them to enter their sleep.

# View Graphs

## Brief Description

User End-Goal Story

*When a user*

*Wants to view their graphs*

*They select the data range they want to view*

*So that the system can display the relevant data*

## Triggers

User

*The user has logged at least a days worth of data and wants to see trends.*

## Scenarios

### Normal Flow

1. User wants to view graphs

2. SYSTEM Display date ranges available

3. User selects date range

4. SYSTEM Graphically displays all events

5. while User is not finished

5.1. User selects a data point

5.2. for each More data points

5.2.1. SYSTEM Displays the event notes for that data point

end for each

end while

## Preconditions

1) The user has data logged.

2) The user wants to inspect for any trends.

## Post-conditions

n/a

## Assumptions

User can view graphically view their logged events

## Requirements

User can view graphically view their logged events

# View History

## Brief Description

User End-Goal Story

*When the user*

*Wants to edit the entries of a previous day*

*They can search for their mood, sleep, and food events from previous days*

*So they can add, update, or delete extra details*

## Triggers

User has remembered details from a previous day and wishes to log them.

User has noticed incorrect details and wishes to change them.

User has noticed duplicate data and wishes to delete them.

## Scenarios

### Update Event Flow

1. User selects a previous day to edit.

2. SYSTEM Displays the events for the day

3. User chooses and event to edit

4. SYSTEM Displays the events current details

5. Edit Event

5. SYSTEM Save to DB

### Delete Event Flow

1. User selects a previous day to edit.

2. SYSTEM Displays the events for the day

3. while Wants to delete events

3.1. User chooses and events to delete

3.2. SYSTEM Displays the events current details

3.3. User presses delete

3.4. SYSTEM Asks for deletion confirmation

3.5. User chooses

3.5.1. if Yes

3.5.1.1. Delete Event

end if

3.5.2. if No

3.5.2.1. SYSTEM Displays the events for the day

end if

end while

## Preconditions

1) User has remembered details from a previous day and wishes to log them.

2) User has noticed incorrect details and wishes to change them.

3) User has noticed duplicate data and wishes to delete them.

## Post-conditions

Delete Event

Edit Event

Save to DB

## Assumptions

User can add mood, food, and sleep events to previous days.

User can delete mood, food, and sleep events to previous days.

User can modify mood, food, and sleep events to previous days.

Logging food events are optional.

Can log multiple food events during the day.

## Requirements

User can add mood, food, and sleep events to previous days.

User can modify mood, food, and sleep events to previous days.

User can delete mood, food, and sleep events to previous days.

|  |
| --- |
| Sequence Diagram Delete Event Normal Flow |
| Sequence Diagram Edit Event Normal Flow |
| Sequence Diagram Log Food Event Normal Flow |
| Sequence Diagram Log Food Event Alternative Flow |
| Sequence Diagram Log Food Event Exception Flow |
| Sequence Diagram Log Mood Event Normal Flow |
| Sequence Diagram Log Mood Event Alternative Flow |
| Sequence Diagram Log Mood Event Exception Flow |
| Sequence Diagram Log Sleep Event Normal Flow |
| Sequence Diagram Log Sleep Event Alternative Flow |
| Sequence Diagram Log Sleep Event Exception Flow |
| Sequence Diagram View Graphs Normal Flow |
| Sequence Diagram View History Delete Event Flow |
| Sequence Diagram View History Edit Event Flow |

|  |
| --- |
| Activity Diagram Delete Event Normal Flow |
| Activity Diagram Edit Event Normal Flow |
| Activity Diagram Log Food Event Normal Flow |
| Activity Diagram Log Food Event Alternative Flow |
| Activity Diagram Log Food Event Exception Flow |
| Activity Diagram Log Mood Event Normal Flow |
| Activity Diagram Log Mood Event Alternative Flow |
| Activity Diagram Log Mood Event Exception Flow |
| Activity Diagram Log Sleep Event Normal Flow |
| Activity Diagram Log Sleep Event Alternative Flow |
| Activity Diagram Log Sleep Event Exception Flow |
| Activity Diagram View Graphs Normal Flow |
| Activity Diagram View History Delete Event Flow |
| Activity Diagram View History Edit Event Flow |

Figure Class Diagram

## 

## Project Requirements

* Android 7.0 Nougat (API version 24) – 92% Compatibility
* 128MB RAM)
* Screen resolution; 1080 x 1920 pixels minimum
* Screen size; 5.0 inches minimum
* Storage; 100mb
* SQLite Database
* On-screen keyboard

MindfulMood should not be a CPU/RAM intensive app as most functionality is inputting data and displaying. Data will be saved to the internal SQLite database and thus does not require Wi-Fi/Cellular connectivity.

Android 7.0 was released in 2016, making it 6 years old today. The average user replaces their phone every 2 to 3 years, thus Android 7.0 should be reasonable expectation. It also has a 92% compatibility for current user’s phones.

## GUI Mock-ups

|  |  |
| --- | --- |
| Home Screen:  The home screen will host the functions of the app. The user will be able to record and entry for their mood, food, and sleep, as well as being able to generate graphs of their data and view their input history.  The three record buttons will also host information for quick access, such as when the user last made an entry. The record food button will also have indicators for what means have been entered out of breakfast, lunch, dinner, and snacks.  The data search bar at the top will allow the user to view previous days to add missing data. |  |
| Record Mood Screen:  Here the user will be able to enter a time and rate their mood on a scale. The time will default to the current locale time but able to be changed if the entry was earlier.  The scale displays a scroll bar for visualisation and lights up the number above it when the user changes the scroll bar progress.  Next the user can enter an option description of why they are feeling a certain way. The description is encouraged as this will help link moods and emotions with external events.  Finally, the user will press save and return to the home screen. The return arrow at the top will also return to the home screen but it will not save the entry. |  |
| Record Food Screen:  This screen displays tabs along the top that correspond to different meals. For each tab, the user can select the time (also defaults to the current time) and a description of what they ate.  The bottom of the screen has two progression buttons, one to save and return to the home screen, and the add will save and navigate to the mood screen. This provides and option mood entry to related to the same time the user ate.  Once again, the return arrow at the top will also return to the home screen but it will not save the entry. |  |
| Record Sleep Screen:  This screen helps the user enter the time that they went to bed and woke up the next day. It also has the option for the user to record the quality of their sleep. These descriptions can help the user identify factors to why they are feeling a certain way.  The bottom of the screen has two progression buttons, one to save and return to the home screen, and the add will save and navigate to the mood screen. This provides and option mood entry to related to the same time the user ate.  Once again, the return arrow at the top will also return to the home screen but it will not save the entry. |  |
| View History Screen:  This screen displays all the entries that user has entered today. They can filter entries by different rating and also exclude food and sleep. The sort button can help sort the results by time, or by mood rating.  Each entry has a corresponding Edit or Delete button. This is useful for appending entries and deleting duplicate data.  At the top of the screen the user will be able to navigate to past days and view their history.  Once again, the return arrow at the top will also return to the home screen. |  |
| View Graphs Screen:  This is where the user will be able to view their history visually. The graph will be plotted mood rating against time and highlight when the user has made an entry. A trend line can then be drawn between data points. |  |
| Clicking on a data point will display the description that is associated with that entry. The user can then edit if need be or close the pop up.  At the top of the screen the user will be able to navigate to past days and view their history.  Once again, the return arrow at the top will also return to the home screen. |  |