

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Journal Flow \(3\)](#)

[Journal Overview \(4\)](#)

[Home Screen \(5\)](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services or other external services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Functionality](#)

[Task 4: Settings](#)

[Task 5: Notification Service](#)

[Task 6: Add Widget](#)

[Task 7: Quiet please!](#)

[Task 8: Add Pdf Convert](#)

[Task 9: Add Flavor](#)

[Task 10: Add Google Play Services](#)

[Task 11: Add Guides](#)

[Task 12: UI polish](#)

[Task 13: Code polish](#)

**GitHub Username:** [atschpe](#)

# C.MI. - Meditation Journal

## Description

C.MI. (Certified Meditation Instructor) is an online certification I took a few years ago. I have been supporting it ever since as a Faculty Assistant ever since, helping to field student questions and keeping an eye on the online discussion. Over the course of the years I repeatedly noted on recurring questions, confusion and problems arising around the meditation log students are required to keep and submit.

The course currently provides a starter word-document where students log prep, meditation and review for their daily meditation. Some struggle with handling the document (being less versed with computers), others haven't fully grasped certain guidelines pertaining to time limitations, length of logging and similar. Hence I would like to propose my capstone project to solve this problem, by creating an app that will enable the user/student to log their meditations, which supports both the course guidelines and the students. It will thus combine the functionality of a meditation timer and journaling system into one.

## Intended User

- C.MI. students (the course has been constantly growing – currently with about 1000 students enrolling every year)
- non-students who want to use the app to log their personal meditation practice

## Features

- Student and non-student users
- Meditation journaling system:
  - self-assessment to be filled in by the user after each meditation
  - times (preparation, meditation & review) will be automatically calculated by the app (ensuring they meet the guidelines of the course)
  - each entry will be stored, and accessible to the user. Once the target for each log has been met (20 hours per course description) the user can export a pdf of the log to be submitted (currently the course requires additional material to be submitted so an auto-submit would be further down the line when the app is fully implemented and successful)
- The user can personalise the app use, by choosing from provided app-own sounds for the callback, sounds already provided by the phone-package, or even audio provided by the course for this purpose (course-own material would not be openly available to the non-student user)
- In-app purchase of course-owned audio & subtle ads for non-students users

# User Interface Mocks

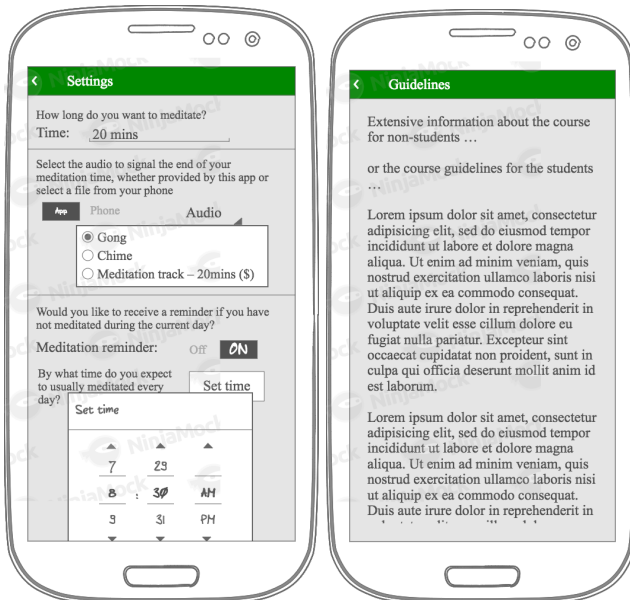
## Screen 1



On first opening the app the user will be met with the left screen. A C.M.I. student can click the appropriate button and enter their activation code. Non-students can directly move forward to the overview activity. On consecutive launches the user will immediately be directed to the The Overview activity gives access to the settings and guidelines, offers a quick overview where the user currently is in their journaling process. Clicking the “add new Journal entry” button will start the journaling process. Clicking “View Journal” takes user to a full overview of the entries, where they can edit, delete and – once over 20 hours accumulated – create a pdf ready for submission (only students). In the overflow menu:

- **Meditation Settings** opens the Settings screen
- **Guidelines** opens C.M.I. guidelines for students or more information for free users
- **Enter Activation Code** takes user back to the first entry screen enabling them to activate the code at a later date

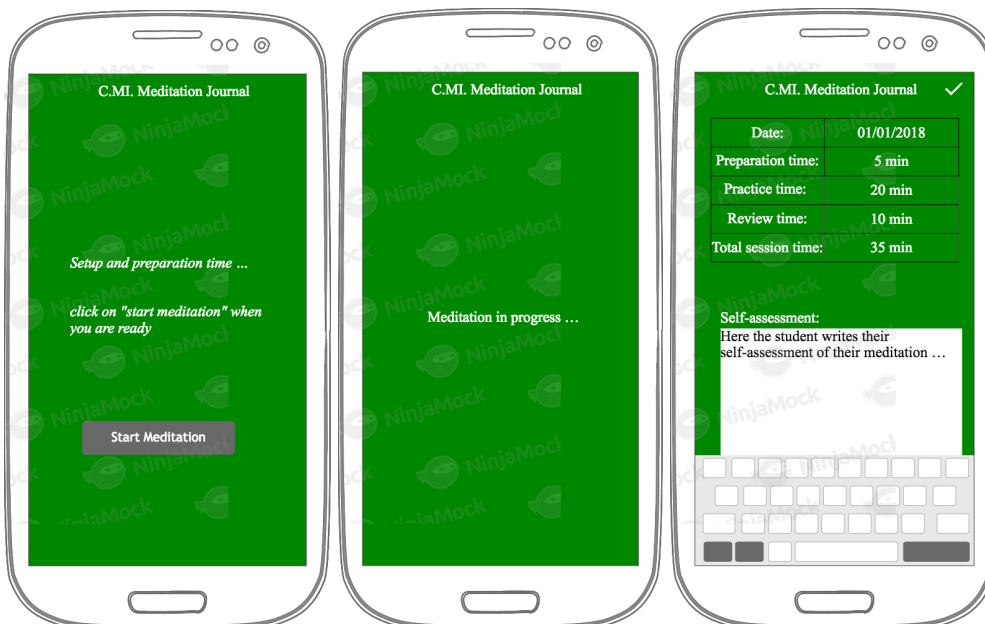
## Screen 2



The settings screen offers the user the chance to set the length of the meditation and the audio callback. If they wish to receive a reminder the “set time” button appears allowing them to specify when to alert them if they have not meditated that day.

The guidelines offer the C.M.I. student a run down of the guidelines for their meditation practice, or the non-student more information about the course and the chance to sign up.

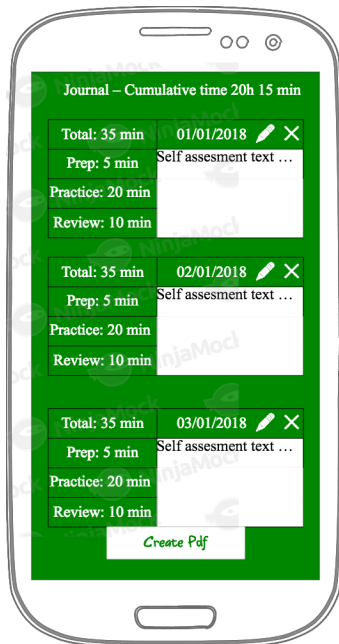
## Journal Flow (3)



The journal flow consists of three fragments, representing the three stages outlined in the guidelines: preparation, meditation and review. Timers are consciously not shown, to encourage

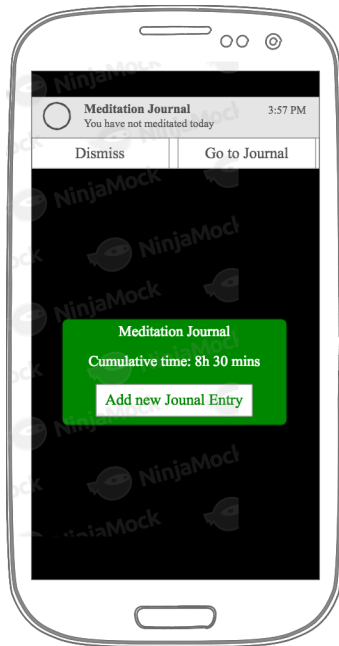
the meditator to fully focus on their meditation. Likewise, as long as the user is in this flow all alerts from the phone will be muted but for the signal that ends the meditation. The meditation fragment will hold an exoplayer, to playback the callback or full-length audio, as selected by the user.

## Journal Overview (4)



The journal overview displays all entries. The user can edit an entry (the review-fragment will be used for this) or even delete one. Once 20 hours or more have been reached the “create pdf” button will appear enabling the user to export to pdf ready for submitting.

## Home Screen (5)



A home screen widget may provide the user a quick overview of the meditation practice and provides a button that takes them directly into the journaling process. If wished a notification will pop up if the user has not meditated/journaled after a set time of day.

## Key Considerations

### How will your app handle data persistence?

As these journals are personal to the user (even when they later submit them for grading) I aim to store the data locally, so they have full control over the data. I.e. DB with ContentProvider.

### Describe any edge or corner cases in the UX.

- As per the C.MI. guidelines the student may only submit one meditation per day which may not exceed 60 minutes. Likewise there is a cap on the amount of preparatory time and review time. The app will enforce these limits providing alerts as needed to explain to the student why the time has been reduced, or why they cannot start a new journal on the same day of another one.
- As the user is adding information to be stored there will be checks to ensure all fields are completed and meet criteria.

### Describe any libraries you'll be using and share your reasoning for including them.

- [Android-PdfMyXml](#): (v. 1.0.1) Currently, the journals are to be submitted as pdfs for the C.MI. course. This library uses an xml file to first display the data and create a pdf of the xml.
- I am currently using Gradle 4.4 / Android Studio 3.1.2 and will solely code in Java.

**Describe how you will implement Google Play Services or other external services.**

- Google Ads: for the non-student flavor. A banner at the bottom of the MainActivity, to keep it non-intrusive.
- Google Pay: for non\_student interested in using the course-provided audios (meditation audio, or similar)

## Next Steps: Required Tasks

### Task 1: Project Setup

- Setup Database & ContentProvider
- Select audios and store in assets ready for selection. This will also include a test-audio of a minute or so in length, to ensure this too can be tested without lengthy waits.
- Set up an Android & Java Library. Goal being that the app can later be expanded to iOS and/or web

### Task 2: Implement UI for Each Activity and Fragment

Create the Activities and Fragments

- MainActivity which will hold the overview fragment of the journal and first time setup fragment
- MeditationActivity which will hold the fragments preparation, meditation and review.
- JournalActivity that will display a recyclerview of all the entries and once over 20 hours the button to create the pdf.

### Task 3: Functionality

Build the UI functionality including interaction with the db.

- The content provider will enable all standard functionalities, enabling the user to create, view, edit and delete entries. A Cursorloader will retrieve the data for displaying purposes.
  - Add checks to ensure that time guidelines are being upheld (prep- & review-time cap, not exceeding 60 mins, only once a day) along with alerts that explain why this is happening.
  - Add checks to ensure that all fields are populated correctly.
  - RecyclerView adapter for the JournalActivity.
- Write tests to ensure UI functionality and guideline adherence work

### Task 4: Settings

Add the settings and their functionality

- Audio selection: setup the selection fro app- and phone provided audio
  - Time selection: if user selects a full guiding audio, then this setting will be disabled
  - Meditation reminder: the user can choose to be reminded. If they activate the reminder a button will appear to set the time of day by when they intend to have meditated.
- Write test to ensure settings work

### **Task 5: Notification Service**

Implement the service that will create and deploy the notification to remind the user, if they have not meditated by the time of day they have set in the settings.

### **Task 6: Add Widget**

Set up the widget that will display the current cumulative time meditated and a button that takes the user directly to the preparatory fragment of the journal

### **Task 7: Quiet please!**

Implement functionality that turns off alerts, sounds and similar so the user is not distracted during the meditation. The only “sound” allowed is the signal that ends the meditation or sounds provided by guided meditation audio/video.

→ Write test to ensure all but the callback sound/alerts are turned off

### **Task 8: Add Pdf Convert**

Set up Android-PdfMyXml. Enable the user to view the full journal and then convert it to a pdf.□

→ Test UI and pdf conversion

### **Task 9: Add Flavor**

Add flavors distinguishing students from free users.

- On first launch
- An option to switch if the user decides to become a course student or wants to continue using the app after course completion
- Add code activation logic

### **Task 10: Add Google Play Services**

Add the play services for free users:

- Ads to the Main Activity (overview)
- Google Pay to the selection of audio (settings)

### **Task 11: Add Guides**

- Create a walkthrough so the user knows how to use app
- Add further information about the course (for non-student), or the full guidelines (for students) accessed from the menu dropdown on an extra item

### **Task 12: UI polish**

- Polish transitions, animation and overall look
- Ensure consistency of colors, borders etc.

### **Task 13: Code polish**

- Review that comments and code are actively used and correctly formatted
- Run lint to double check for any hidden problems





