Milestone 1

Version 1.0

Mental Health App

By:

Daniel Bornemann
Joshua Breininger
Phi Duong

1. **Team Information**

1.1 Names and Emails of Project Members

- Daniel Bornemann - dbornemann2018@my.fit.edu
- Joshua Breininger - jbreininger2018@my.fit.edu
- Phi Duong - pduong2018@my.ft.edu

1.2 Faculty Advisor

- Dr. Bernhard - pbernhar@cs.fit.edu

1.3. Client

- Calvin Schwartz
- Maya Lindseth

2. **Project Details**

2.1 Progress of Milestone 1

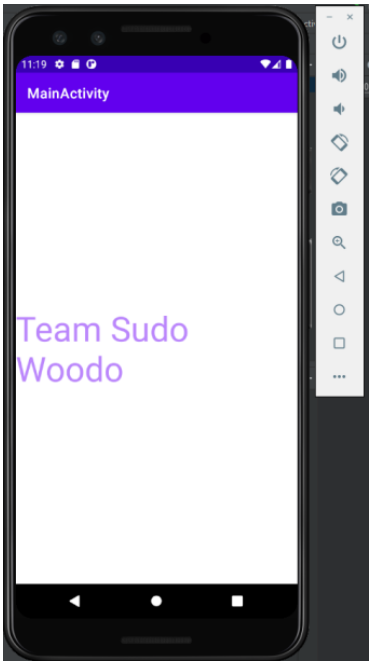
Task	Progress	Daniel	Josh	Phi	To Do
1. ("hello world") demo(s)	100%	16%	16%	66%	None
2. Resolving technical challenges	100%	16%	16%	66%	None
3. Compare and select collaboration tools	100%	16%	66%	16%	None
4. Create a Requirements Document	100%	50%	25%	25%	None
5. Create a Design	100%	25%	50%	25%	None

Document					
6. Create a Test Plan	100%	50%	25%	25%	None

2.2 Discussion of Milestone 1

Task 1: “hello world” demo

For the demos, we successfully installed and used Android Studio to emulate an Android Device, which we ran code on. We also successfully wrote a hello world program for java to ensure our installations of the language were correct. Android Studio presented some challenges for us by being rather confusing to understand, and having to select a good specific device to emulate. Android Studio also crashed multiple times, even crashing one of our team member’s entire PC. This could serve as a severe roadblock in the future if precautions are not taken. A screenshot of the emulation running is contained below.



Task 2: Resolving technical challenges

All of us spent some time familiarizing ourselves with Java, although to an extent we will have to learn as we go, which will be a future challenge. We also came up with a database to use, being an SQLite relational database as Android has SQLite naturally installed and highly recommends it. Our database design was remade after its initial creation to have attributes that allow us to “clean” the database by removing data associated with a past year and leaving the information about the year as simply averages in the year entity’s attributes rather than having to save 365 entries, which should save space on the devices.

Task 3: Compare and select collaboration tools

As we examined the various collaboration tools, we compared and contrasted them. Visual Studio Code is professionally used and has a massive number of features, and one of our teammates has experience in using it. Codeshare.io had chat abilities which although useful is not a particularly important attribute to have as our communication has already been through Discord, a messaging and calling app we are used to. Codecollab.io like Codeshare.io did not have the same level of feature depth as VSC and required internet access to work on code, which is limited in usefulness. Furthermore, we ended up using Android Studio for our code writing uses as it has an inbuilt emulator for Android, which means we don't have to separately install it. Since the above tools had limited integration we could find with Android Studio, we decided to simply use GitHub as our collaboration tools to upload and view code, and continue using the video and messaging app Discord as we could share our screen to show our emulations running which is more useful for this project than simply sharing code.

Task 4: Create Requirements Document

Our requirement document is meant to describe the requirements as specified by the client to fulfill the goal of our project. It contains descriptions of all features we need to have, and what those features have to do as well as what kind of behavior is correct for them. After meeting with our clients, we had to partially revise our document to include the option for the user to pick a name for the avatar in the daily game. They suggested some future features/requirements we could include if we have time, such as making more avatars. They also mentioned that special care should be taken in designing the theme and graphical elements of the app to not make them unpleasant or exhausting to look at.

Task 5: Create a Design Document

The design document was made to show how our system designs worked and how the program interacts with the user and itself. Since the database user needs to be small and not take up much space, the effort was put in to keep entities and attributes to a minimum while also providing functionality in a relatively efficient way. A possible challenge in the future may be the storage of the graphical files for the UI that would change every month, although as there would not be a particularly large amount this will probably not be an issue. If it becomes one we would have to find a new method of storing those elements.

Task 6: Create a Test Plan

This will include our test cases that we should be able to implement and test. This will be our guide to show our progress with the app. The user's amount of input is minimized to prevent mistakes and make the process easier which reduces the number of tests we need to do. However, the primary function of the app relates to a database, and so a relatively large

number of test cases need to be made to ensure none of the user's input could potentially mess up the database which could result in other functionalities being broken, such as the graphing functionality.

2.3 Discussion of Team Contribution

- Joshua Breininger focused mostly on working on descriptive and planning elements of the milestone. He was the primary communicator with the clients and participated heavily in writing the documents. Part of the planning involved designing the database and system architecture diagram which he did while researching how Android would be involved as a language and system in those aspects of the project. He also took most of the responsibility for researching collaboration tools.
- Daniel Bornemann was in charge of the Software Test Plan and the Software Requirements Document, as well as communicating with the Faculty Advisor. He focused mainly on finding a format the team agreed upon for those two documents and keeping it consistent. He also helped Phi with the "hello world" demo and has a working Android emulator on his device.
- Phi Duong took part in the document writing and checked and edited them for format and errors. He also focused on the technical aspects of the project, ensuring and working on Android Studio and testing its emulator, resolving most of the technical issues we foresaw.

2.3 Plan for Milestone 2

Task matrix for Milestone 2

Task	Josh	Phi	Daniel
1. Implement, test, and demo a basic Android UI.	50%	25%	25%
2. Implement, test, and demo an interactive pop-up with an input field.	25%	25%	50%
3. Implement, test, and demo saving inputted information into an in-app database.	33%	33%	33%
4. Implement, test, and demo an in-app database structured around a calendar.	25%	50%	25%
5. Implement, test, and demo retrieving information from the database.	33%	33%	33%

2.4 Discussion for Milestone 2

1. We will need to implement a basic functioning UI into the Android Emulator. This is beyond the emulation we did previously and will be a significant step up in the challenge. We plan on implementing a simple button with a label for the user to press, which will be used in other parts of the milestone. We also will need to start looking into how to change how UI elements look to have an app theme.
2. We need to implement a pop-up with an input text field. This will be connected to the button in Task 1 to create a basic plan for how our apps will function considering most of the functionality relies on these pop-ups with saved information. The pop-up will also have an element to include a label to state what the text field is for.
3. Next, we will save the information saved in the above text field into a small in-app SQLite database. This will require using the SQLite libraries and will take some time to understand the Java API for SQLite. We will not be concerned with saving the database yet, as being able to manage and store information in the database is needed beforehand.
4. We will make a database similar to what we will be using in the actual app except fairly simpler, only having day entities, although they will be made here in the demo as strong and not weak entities. This database will use SQLite and use the information saved from the previous tasks to save multiple different pieces of information and thus different entities to ensure the database itself works. We will include multiple attributes to better understand and have a better template for the database we will be using in the future.
5. Finally, we will test retrieving information from the database. This will require the information we obtained in earlier tasks to be correctly saved and will hardcode some SQLite queries to verify it works. We will have to research and begin optimizing these queries to maximize efficiency, especially considering most of the app will be interacting with the database in some way.

2.5 Dates of Meetings with Clients

1. September 16th, 2021
2. September 26th, 2021

2.6 Client Feedback

- Add option for the user to decide a name for their avatar (added to documents.)
- The UI looks understandable but plain. (Addresses with themes)
- Themes should be made specifically to be appealing to look at and not use exhaustive or unpleasant imagery.
- Aside from that, it appears to fulfill the requested functionality.

2.7 Dates of Meetings with Faculty Advisor

- September 21st, 2021 - Email conversation discussing database choice about SQLite.
- October 1st, 2021 - Email sending documents and progress report for feedback.

2.8 Faculty Advisor Feedback

- Please create and provide an SQL DDL file for the database design.

Faculty Advisor Signature: _____ Date: _____

Evaluation by Faculty Advisor

Faculty Advisor: detach and return this page to Dr. Chan (HC 214) or email the scores to pkc@cs.fit.edu

Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Joshua Breininger	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Daniel Bornemann	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Phi Duong	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: _____ Date: _____