Milestone 2

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**

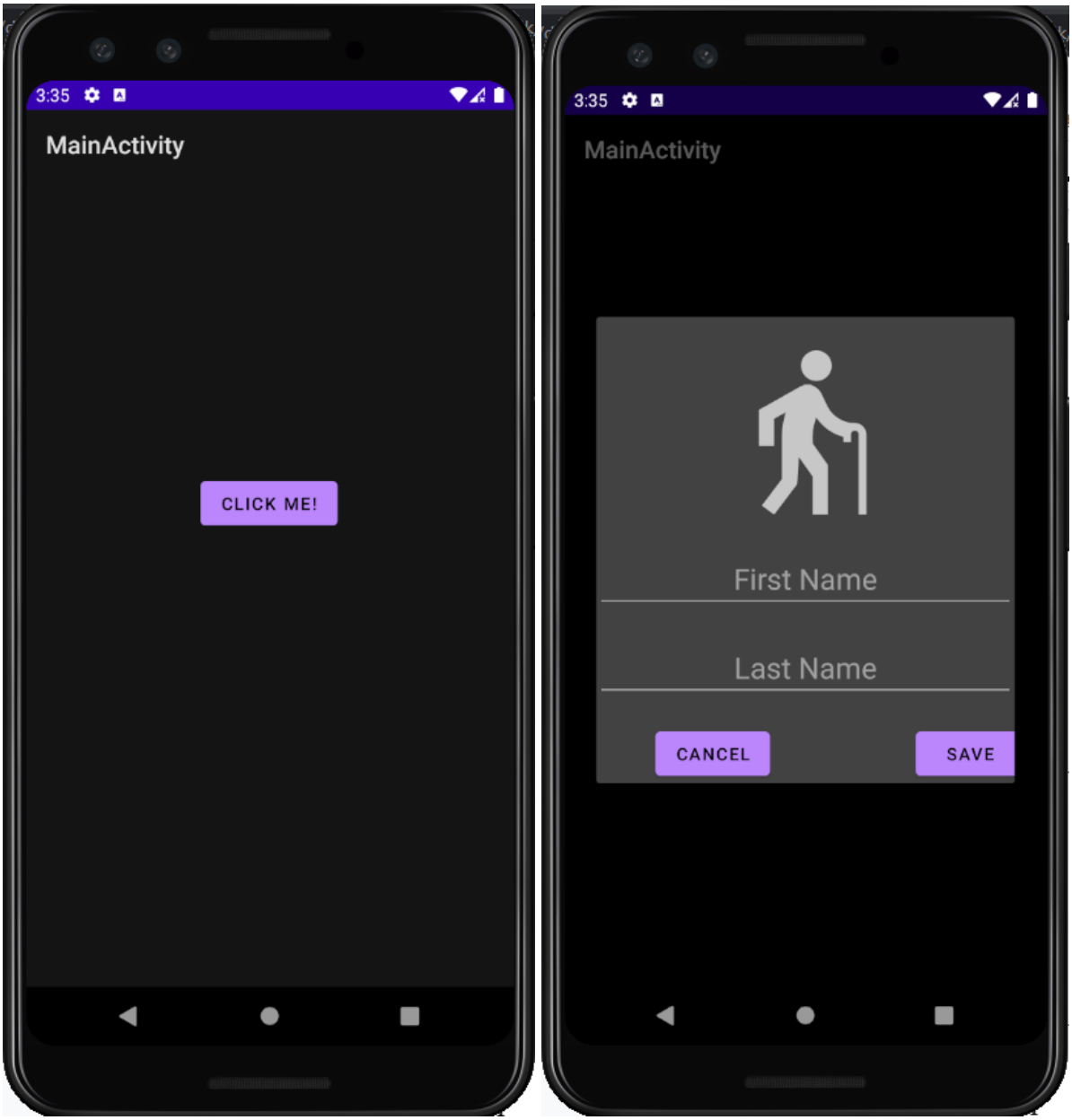
2.1 Progress of Milestone 2

Task	Progress	Josh	Phi	Daniel	Todo
1. Implement, test, and demo a basic Android UI.	100%	50%	25%	25%	None
2. Implement, test, and demo an interactive pop-up with an input field.	100%	25%	25%	50%	None
3. Implement, test, and demo saving inputted information	100%	33%	33%	33%	None

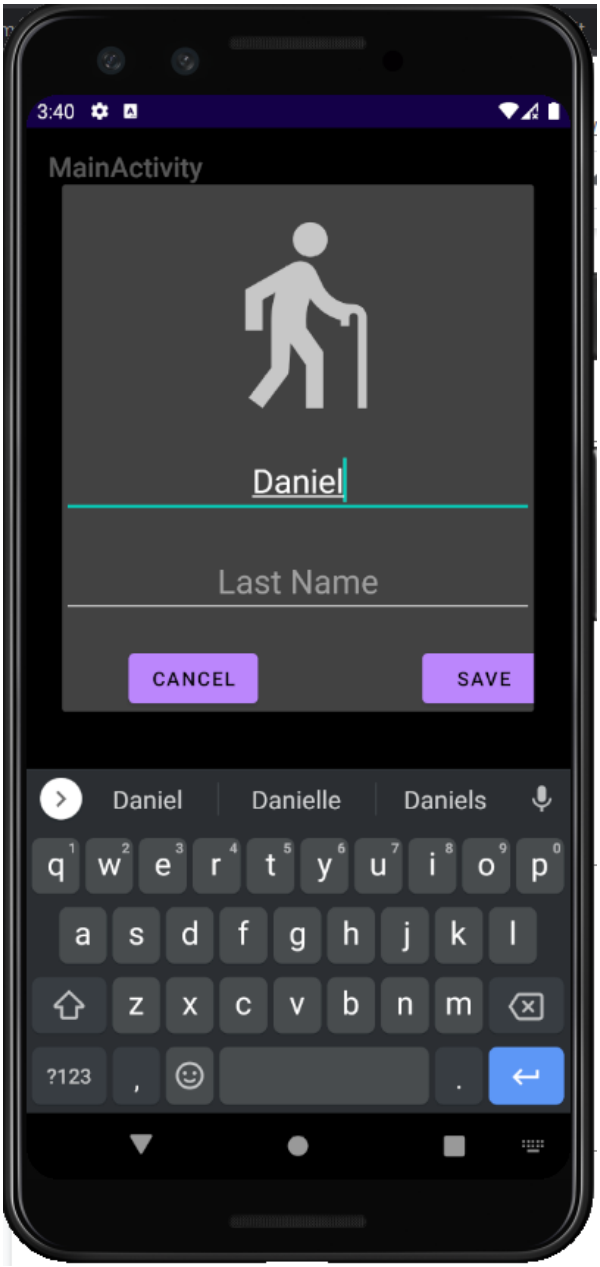
into an in-app database.					
4. Implement, test, and demo an in-app database structured around a calendar.	100%	25%	50%	25%	None
5. Implement, test, and demo retrieving information from the database.	70%	25%	20%	25%	We still need to write actual queries to obtain information through SQL and not indirectly through the handler. This is incorporated into Milestone 3.

2.2 Discussion of Milestone 2

Task 1: Implement, test, and demo a basic Android UI.



The above two images show the basic UI that we implemented. We figured out how to add vector images as shown by the graphic in the second image. The “Cancel” button is used to close the pop-up and the “Save” button saves whatever the user types into “First Name” into the database and displays it on the same screen as “Click Me!” This is not adapted to what our UI will actually look like, but it is a good starting point as our app will have many pop-ups similar to this one.


3:40
MainActivity
Daniel
Last Name
CANCEL
SAVE
Daniel
Danielle
Daniels


3:41
MainActivity
Daniel
CLICK ME!

Task 2: Implement, test, and demo an interactive pop-up with an input field.

Task 2 is covered in Task 1.

Task 3: Implement, test, and demo saving inputted information into an in-app database.

We created a database designed to match a calendar, being an implementation of the tables in the design document. We used a separate table for average food attributes so that both month

tables and year tables can have their own individual linked table to contain their average food items. We had two special attributes to serve as our keys for those tables, where a -1 is default for the month_num so that the food tables associated with a year in general can be signified as such rather than having a table with the exact same attributes but for months and years separately. Although Android and Java prefer to use SQLite programmatically, and so we implemented our database in that way, contained below is the contents of the ddl file that could be used to load up our schema. The currently implemented database is slightly simpler than this schema, where our daytable's only primary key is Day_Num and it has no foreign keys as we prioritized making the groundwork over the specifics, and changing the database to its full point will be done in Milestone 3.

```
PRAGMA foreign_keys = ON;
CREATE TABLE averagefood (
  Average_Fruits REAL,
  Average_Vegetables REAL,
  Average_Sugar REAL,
  Year_Num INTEGER,
  Month_Num INTEGER DEFAULT -1,
  FOREIGN KEY(Year_Num) REFERENCES yeartable(Year_Num),
  PRIMARY KEY(Year_Num, Month_Num));
CREATE TABLE daytable (
  Day_Num INTEGER,
  Year_Num INTEGER,
  Month_Num INTEGER,
  Mood INTEGER,
  Fruit INTEGER,
  Vegetable INTEGER,
  Sugar INTEGER,
  FOREIGN KEY (Year_Num) REFERENCES yeartable (Year_Num),
  FOREIGN KEY (Month_Num) REFERENCES monthtable (Month_Num),
  PRIMARY KEY(Day_Num, Month_Num, Year_Num));
CREATE TABLE monthtable (
  Month_Num INTEGER,
  Av_Exercise REAL,
  Av_Mood REAL,
  Year_Num INTEGER,
  FOREIGN KEY (Year_Num) REFERENCES yeartable (Year_Num),
  PRIMARY KEY (Month_Num, Year_Num));
CREATE TABLE yeartable (
  Year_Num INTEGER,
  Av_Exercise REAL,
  Av_Mood REAL,
  PRIMARY KEY(Year_Num)); (edited)
```

Task 4: Implement, test, and demo an in-app database structured around a calendar.

This task is covered in Task 3, where we show and describe our implemented database.

Task 5: Implement, test, and demo retrieving information from the database.

We didn't fully complete this task as writing the handlers for the SQL integration was harder and more extensive than we expected. We do have a handler devoted to reading from the database, however it does it by reading each line rather than using an SQL query as it was part of our testing to ensure we properly made our database. The handler is more of an issue

than the SQL queries as it is what is unfamiliar to us rather than SQL logic itself, so having a handler that will look similar to the query handler is substantial progress to completing reading from the database.We have shifted using queries to Milestone 3 as a large part of that milestone will be making efficient queries to use for our graphs.
Sample Potential Query:

```
SELECT Day_Num, Month_Num, Year_Num FROM daytable
 INNER JOIN monthtable ON daytable.Month_Num=monthtable.Month_Num
 WHERE monthtable.Av_Mood>5 (edited)
```

2.3 Discussion of Team Contribution

- Joshua Breininger worked on the progress report, wrote the sqlite logic and database as well as helped program the database handling in android/java.
- Daniel Bornemann worked on the UI implementation and the database handling.
- Phi Duong worked on the UI implementation and the database handling.

2.4 Plan for Milestone 3

Task matrix for Milestone 3

Task	Josh	Phi	Daniel
Create a basic UI with buttons and pop ups corresponding to what will be worked on in the future.	30%	35%	35%
Create functionality for food, exercise and mood pop ups, store inputted information into the database.	40%	30%	30%
Prepare for notifications by testing	40%	30%	30%

queries we would use to check for notifications on the database.			
Produce a pop up graph that corresponds to mood and can have the x axis changed by the user through buttons.	30%	35%	35%
Create settings menu which change settings in the app, although they won't have any current functionality.	30%	35%	35%

2.4 Discussion for Milestone 3

1. Given our groundwork in Milestone 2, this should be fairly simple as it will be using what we learned and wrote to make all the menus we will need. We will hold off on making it look graphically good and stick to getting the basics done.
2. This will use Task 1's pop ups to store information in the database. Like Task 1, this should be relatively simple given that we have created essentially samples and basic code reflecting what this would look like in Milestone 2.
3. For notifications, we won't be making them yet, but will test and develop the queries to obtain the relevant information to be time and space efficient. Some testing will be performed on MySQL to test the queries themselves.
4. The pop up graph will be the main challenge of this milestone - it requires querying the database for a lot of information and use of dependencies not natively in the installation.
5. The settings menu will define numbers based on input to change our notification tests, replicating what it would do in the final project.

2.5 Dates of Meetings with Clients

1. October 12th, 2021
2. October 20th, 2021

2.6 Client Feedback

- Seems to accomplish the goal, database makes sense.
- Not any substantial feedback due to this milestone being mostly groundwork.

2.7 Dates of Meetings with Faculty Advisor

- October 29th, 2021 - Email sending documents and progress report for feedback.

2.8 Faculty Advisor Feedback

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: _____