My Mental Health App

Group 9

Alexander, Yonatan	**SN: 251097927**
Kien Tang Tran, Michael	**SN: 250735158**
Aziz, Mohammad Yasin	**SN: 251282726**

Please do NOT post our group's video to the course's YouTube Channel.

Contents

List of Figures

List of Tables

1 Problem Definition

Therapists and patients encounter many small yet persistent challenges during the therapeutic process concerning the patients' ability to self-manage between sessions.

While this project's scope is not large enough to address them all, the following stood out to us:

- Mood tracking between sessions.
    - Usually, at the start of a session therapists ask for an update about the time since the last session. However, self-reporting bias usually means that if the patient is having a bad day or is in a bad mood, their view on the time since the last session is skewed, sometimes heavily. Further, memory can be corrupted and events can easily be forgotten.
- Medication management
    - Inventory (to not run out of important medications, mood stabilizers are a good example in the context of mental health especially)
    - Persistence (taking medications as required and without missing a day)
- Symptoms
    - Medical (unrelated to the patients diagnosed)
- Other self care aspects, such as sleeping, eating, etc.

2 Project Objective

O1 Patients can add new entries.

O2 Patients can view existing entries.

O3 Patients can delete existing entries.

O4 User will be able to view the entries from the past week, at a glance, in a dedicated view.

O5 Learning the process of project development through application of knowledge acquired during the Computer Science 4471A Course.

O6 The creation of an intuitive and user friendly self-management mobile app (Android).

3   Stakeholders List

- Patients - Internal stakeholder

3.1   Internal Stakeholders

Therapists - External stakeholder

external - anyone who has an influence including people who write certain libraries and so on (include instructor and TA as external stakeholders)

internal - programmers, maintainable, customers, users, etc [double check this point's list]

1. Patients (main users and the target audience).

3.2   External Stakeholders

1. Therapists

2. Legislation and public policy (for example security of medical information)

3. Governing bodies (such as the CRPO - The College of Registered Psychotherapists of Ontario)

4. Instructor

5. TA

6. Libraries:

    (a) Android.

    (b) Google.

    (c) MPAndroid (to create the graphs).

    (d) JetBrains (makers of Android Studio).

    (e) Java Libraries

4 Success/Acceptance Criteria for each Stakeholder

- As a user, I want to be able to store my mood on a particular day, on a scale of 0 to 10.
- As a user, I want to be able to store my medication details such as the medication, dosage, etc.
- As a user, I want to be able to store my symptom details on a particular day.
- As a user, I want to be able to view my past mood reports for my therapy visit.
- As a user, I want to be able to view my past symptoms for my therapy visit.
- As a user, I want to be able to view my past medications for my therapy visit.

5 Use case diagram(s)

5.1 Use case Diagram 1: Mood
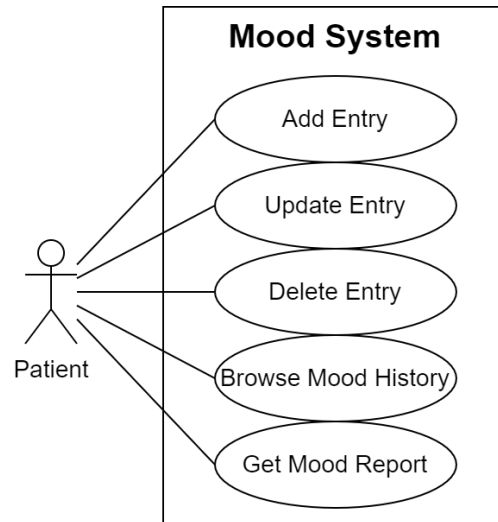
Figure 1: Use case Diagram 1: The Mood Subsystem

Contains the Mood subsystem.

5.2 Use case Diagram 2: Medication
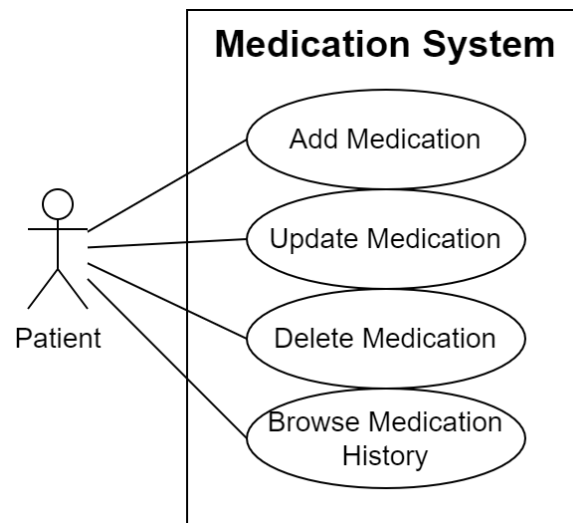
Figure 2: Use case Diagram 2: The Medication Subsystem

Contains the Medication subsystem.

5.3   Use case Diagram 3: Symptom
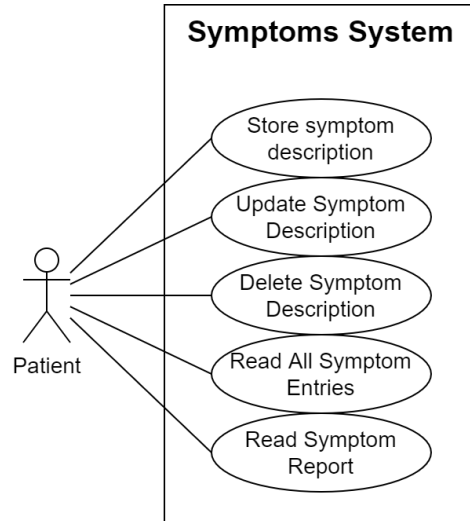
Figure 3: Use case Diagram 3: The Symptom Subsystem

Contains the Symptom subsystem.

6 Use case Descriptions

6.1 Use case Description: Add Medication

Use Case Name: Add Medication
Scenario: Add medication entry to database.
Brief Description: When a Patient (User) enters information for a new medication entry and presses "Submit", add the information to the medications' table in the database.
Triggering Event: Patient (User) enters the "Medication" view.
Actors: Patient (User)
Related Use cases: Update Medication, Delete Medication, Browse Medication History
Stakeholders: Patient
Preconditions: App is open on the "Medications" Screen.
PostConditions: A new entry was added to the medications' table in the database.
Flow of Activities:

Actor	System
1. User opens "Medications" screen	
2. User fills the "Enter the medication name" field	
3. (Optional) User fills the "Enter the brand name" field	
4. (Optional) User fills the "Enter the medication quantity" field	
5. (Optional) User fills the "Enter the medication quantity unit" field	
6. (Optional) User fills the "Enter the medication frequency" field	
7. User pressed "Add Medication Entry"	7.1 System confirms step 2 was completed 7.2 System updates the database with new entry

Table 1: Use case Description: Add Medication, Flow of Activities

Exception conditions:

The user did not fulfill step 2.

6.2 Use case Description: Browse Medication History

Use Case Name: Browse Medication History
Scenario: The user wants to view the medication entries.
Brief Description: When a Patient (User) enters opens the "Medication List" view, the list of medications should appear.
Triggering Event: Patient (User) enters the "Medication List" view.
Actors: Patient (User)
Related Use cases: Update Medication, Delete Medication, Add Medication
Stakeholders: Patient

Preconditions: App is open on any non-"Medications List" Screen.
PostConditions: None (can be empty).
Flow of Activities:

Actor	System
1. User opens the "Medication List" screen	1.1 The system propagates past entries into the view (if there are any).

Table 2: Use case Description: Browse Medication History, Flow of Activities

7 Sequence diagram(s) for the selected use case for descriptions

7.1 Sequence Diagram 1: Add Medication
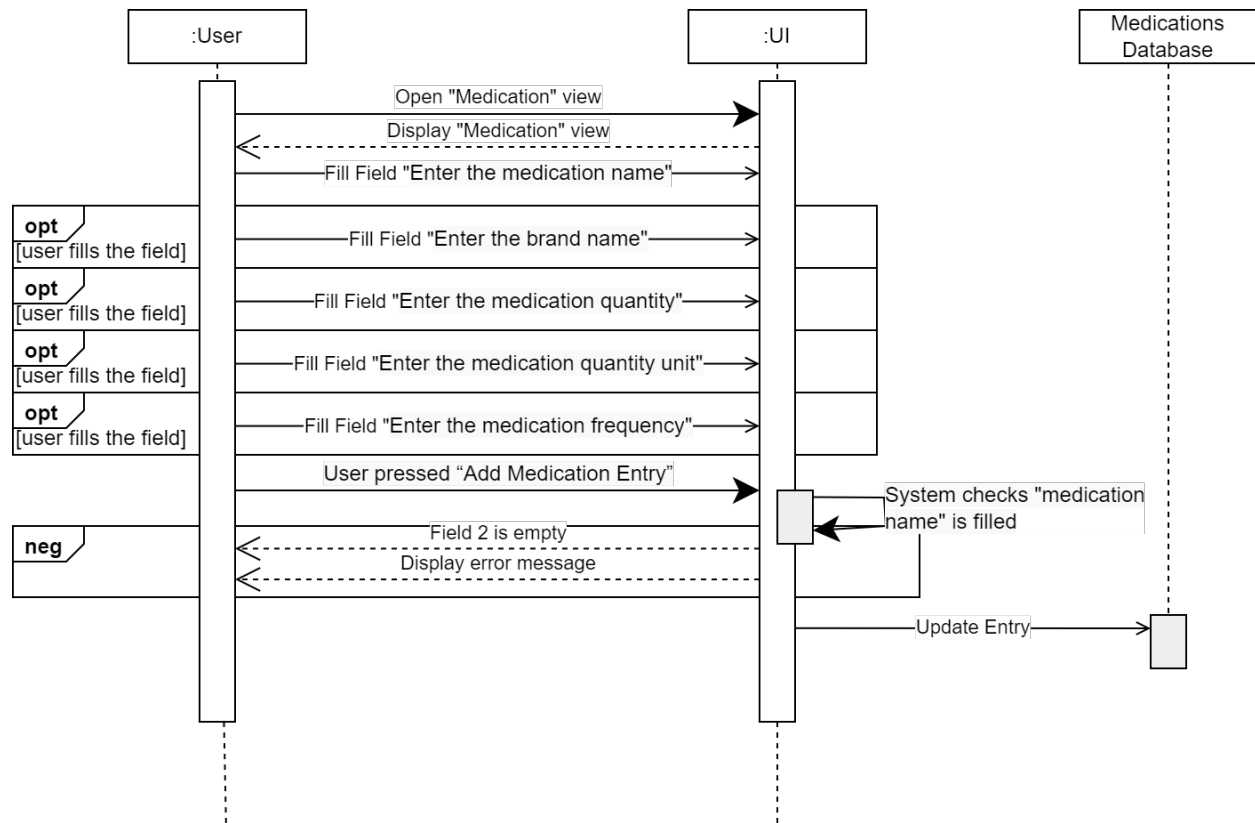
Figure 4: Class Diagram: Relationships.

7.2 Sequence Diagram 2: Browse Medication History
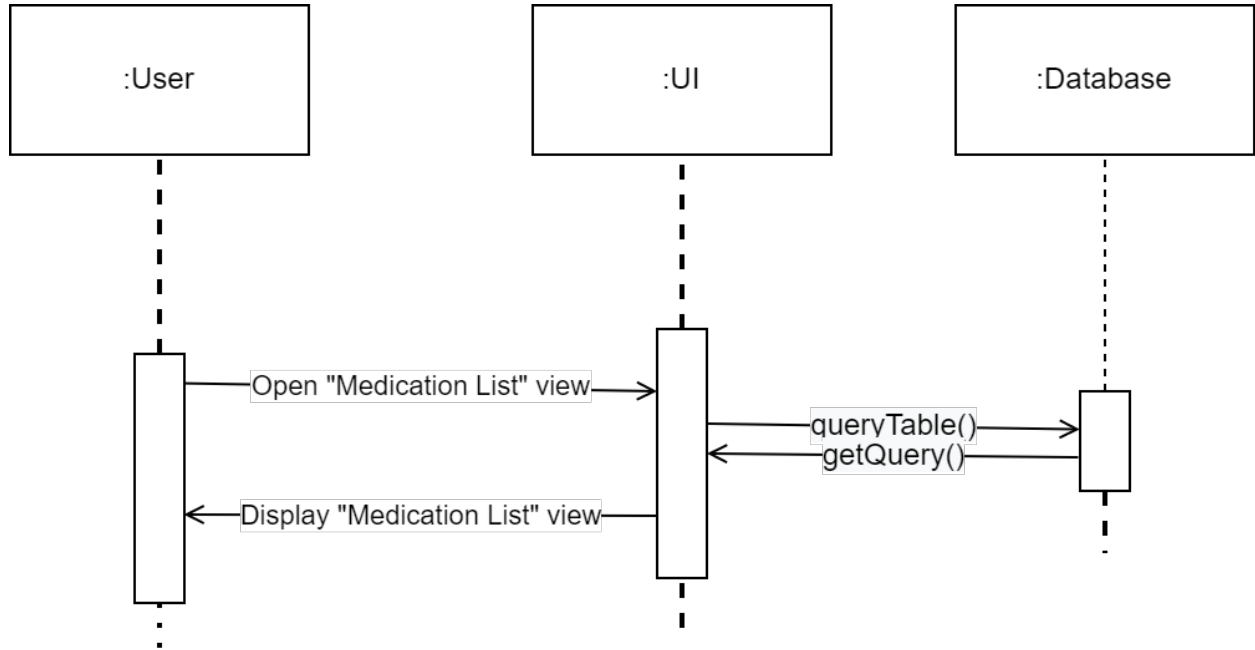
Figure 5: Class Diagram: Relationships.

8   System Architecture

Unfortunately, writing the code in Java meant that our chosen architecture (Model-View-ViewModel) was not feasible, resulting in needing to change the System Architecture to Model View Controller very late in development.

System Architecture: Model View Controller.

9   Detailed Class diagram(s)
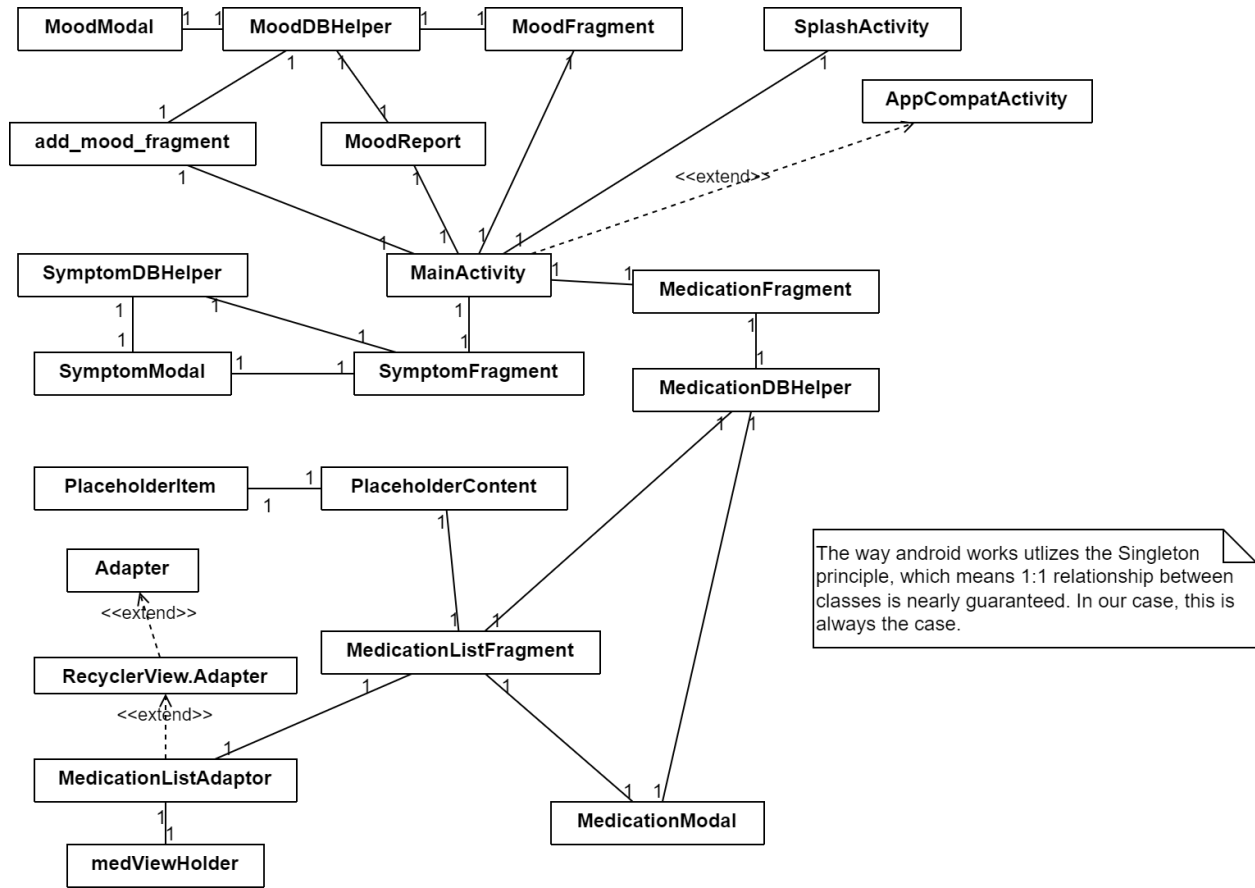
Figure 6: Class Diagram: Relationships.

Contains **all** classes in the system, and the relationships between them. The other class diagrams contain the respective methods.

MedicationFragment
- ARG_PARAM1:String {readOnly} - ARG_PARAM2: String {readOnly} - medicationNameEdt: EditText - brandNameEdt: EditText - dosageEdt: EditText - dosageUnitEdt: EditText - frequencyEdt: EditText - oldSystemName: EditText - addMedicationBtn: Button - readMedicationBtn: Button - deleteMedicationBtn: Button - updateMedicationBtn: Button + MedicationdbHelper: MedicationDBHelper + adapter: Adapter
+ MedicationFragment(): + newInstance: MedicationFragment (String, String) + onCreate(Bundle): void + onCreateView(LayoutInflater, ViewGroup, Bundle): View + onClick(View): void

MedicationModal
- id: long - brandName: String - commonName: String - dosage: String - frequency: String - dosageUnit: String
+ MedicationModal(String, String, String, String, String) + getDosage(): String + setDosage(String): void + getFrequency(String): String + setFrequency(): void + getId(): long + setId(long): void + getBrandName():String + setBrandName(String):void + getDosageUnit():String + setDosageUnit(String): void + getCommonName(): String + setCommonName(String)

MedicationListAdaptor
+ list: ArrayList <MedicationModal> + context:Context
+ MedicationListAdaptor(Context, ArrauList <MedicationModal>): + onCreateViewHolder (@NonNull ViewGroup, int): medViewHolder + onBindViewHolder(@NonNull medViewHolder, int): void + getItemCount: int + update(ArrayList< MedicationModal>): void + onAttachedToRecyclerView: void

medViewHolder
+ medCommonName: TextView + medBrandName: TextView + frequency: TextView + dosage: TextView + dosageUnit: TextView
+ medViewHolder(View):

MedicationListFragment
-ARG_COLUMN_COUNT: String {readOnly} - mColumnCount: int = 1 {readOnly} + medicationsrc: RecyclerView + adapter: MedicationListAdaptor
+ MedicationListFragment(): + newInstance (int): MedicationListFragment + onCreate(Bundle): void + onCreateView(LayoutInflater, ViewGroup, Bundle): View + onClick(View): void

MedicationDBHelper
- DB_Name: String {readOnly} - DB_version: int {readOnly} - Table_Name: String {readOnly} - ID_COL: String {readOnly} - medname_COL: String {readOnly} - brandName_COL: String {readOnly} - dosageQuantity_COL: String {readOnly} - dosageUnit_COL: String {readOnly} - frequency_COL: String {readOnly}
+ onCreate(SQLiteDatabase): void + addNewMedication (String, String, String, String, String): void + deleteRecord(String): void + updateMedication(String, String, String, String, String, String): void + readMedications(): ArrayList< MedicationModal> + onUpgrade(SQLiteDatabase): void

Figure 7: Class Diagram: Medication

Contains **all** classes relating directly Medications. Relationships included only as part of *Class Diagram: Medication.*

add_mood_fragment

- <u>ARG_Param1: String {readOnly}</u>
- <u>ARG_Param2: String {readOnly}</u>

- textProgress: TextView
- descriptionProgress: TextView
- myCalendar: Calendar {readOnly}
- mParam1: String
- mParam2: String
- moodvalue: int
-color: int
- submit_button: Button
- <u>update_button: Button</u>

+ add_mood_fragment()
+ newInstance:(String, String): add_mood_fragment
+ onCreate(Bundle): void
+ onCreateView(LayoutInflater, ViewGroup container, Bundle): View
+ onProgressChanged (SeekBar, int, boolean): void
+ onStopTrackingTouch(SeekBar): void
+ onStartTrackingTouch(SeekBar): void
+ onDateSet(DatePicker, int, int, int): void
+ onClick(View): void
+ setMoodValue(): void
+ getMoodValue(int): int
+ setColor(int): void
+ getColor: int
+ updateView(int)

MoodDBHelper

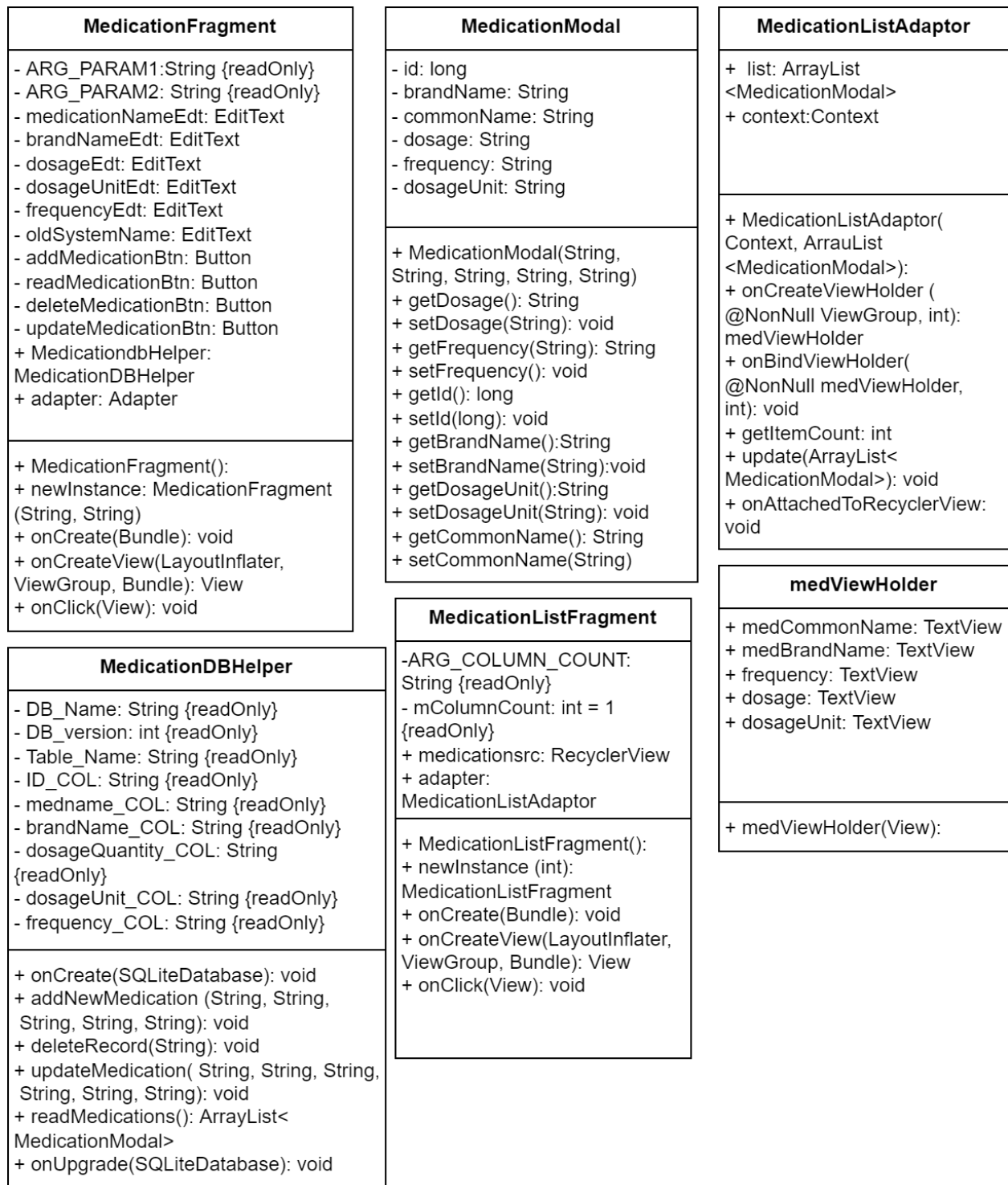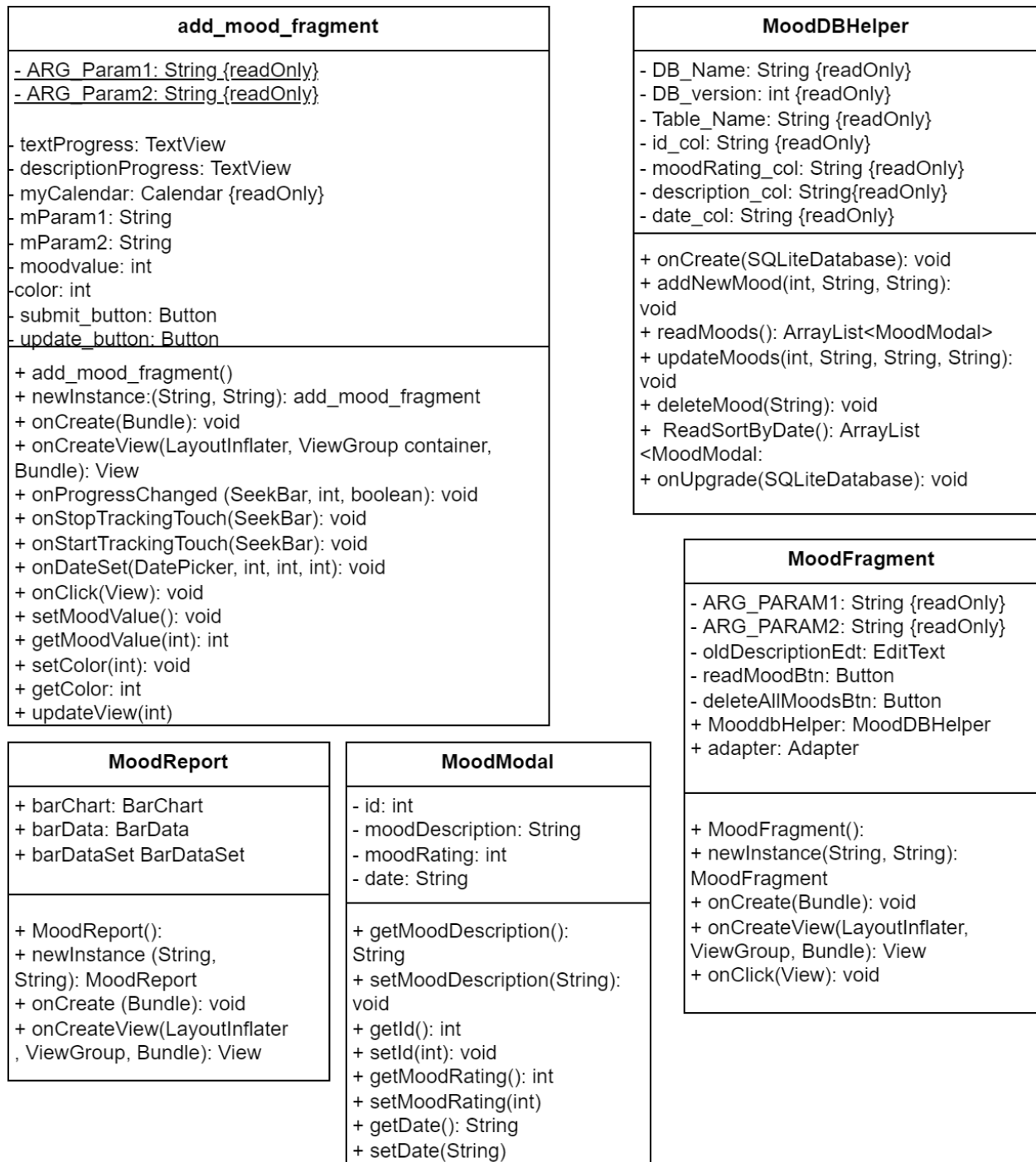- DB_Name: String {readOnly}
- DB_version: int {readOnly}
- Table_Name: String {readOnly}
- id_col: String {readOnly}
- moodRating_col: String {readOnly}
- description_col: String{readOnly}
- date_col: String {readOnly}

+ onCreate(SQLiteDatabase): void
+ addNewMood(int, String, String): void
+ readMoods(): ArrayList<MoodModal>
+ updateMoods(int, String, String, String): void
+ deleteMood(String): void
+ ReadSortByDate(): ArrayList <MoodModal:
+ onUpgrade(SQLiteDatabase): void

MoodFragment

- ARG_PARAM1: String {readOnly}
- ARG_PARAM2: String {readOnly}
- oldDescriptionEdt: EditText
- readMoodBtn: Button
- deleteAllMoodsBtn: Button
+ MooddbHelper: MoodDBHelper
+ adapter: Adapter

+ MoodFragment():
+ newInstance(String, String): MoodFragment
+ onCreate(Bundle): void
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View
+ onClick(View): void

MoodReport

+ barChart: BarChart
+ barData: BarData
+ barDataSet BarDataSet

+ MoodReport():
+ newInstance (String, String): MoodReport
+ onCreate (Bundle): void
+ onCreateView(LayoutInflater , ViewGroup, Bundle): View

MoodModal

- id: int
- moodDescription: String
- moodRating: int
- date: String

+ getMoodDescription(): String
+ setMoodDescription(String): void
+ getId(): int
+ setId(int): void
+ getMoodRating(): int
+ setMoodRating(int)
+ getDate(): String
+ setDate(String)

Figure 8: Class Diagram: Mood

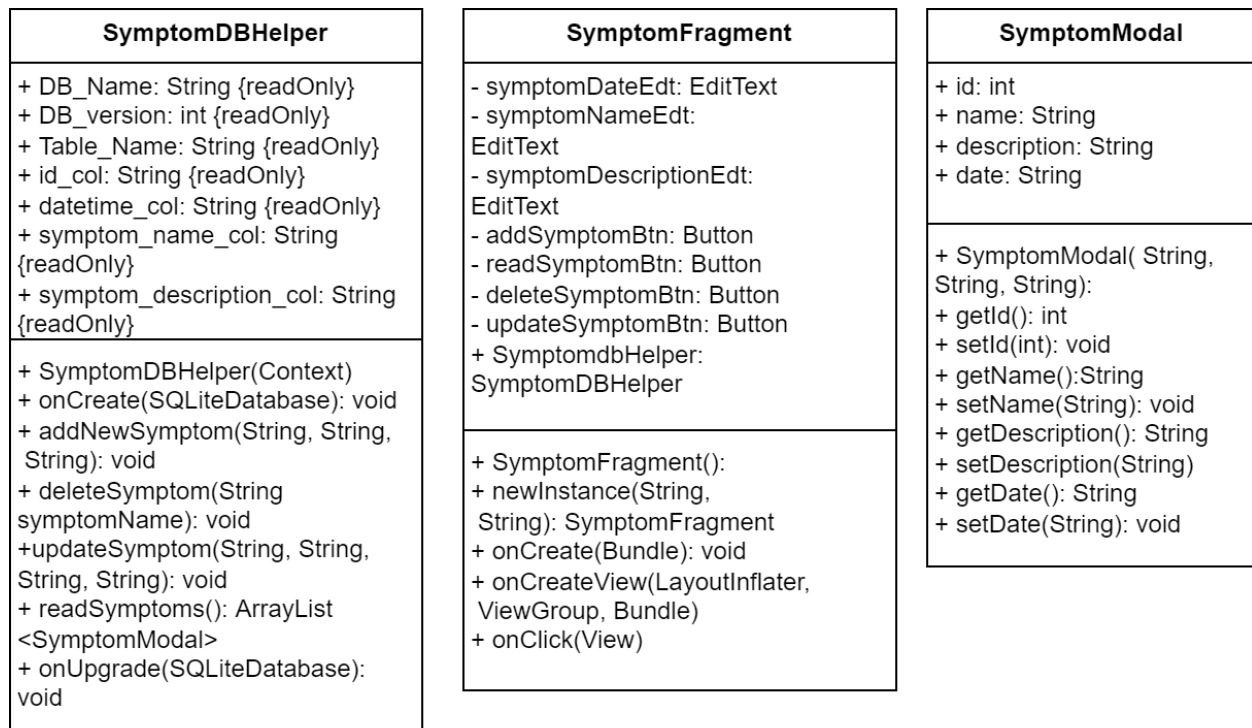Contains **all** classes relating directly Mood. Relationships included only as part of *Class Diagram: Medication.*

SymptomDBHelper	SymptomFragment	SymptomModal
+ DB_Name: String {readOnly} + DB_version: int {readOnly} + Table_Name: String {readOnly} + id_col: String {readOnly} + datetime_col: String {readOnly} + symptom_name_col: String {readOnly} + symptom_description_col: String {readOnly}	- symptomDateEdt: EditText - symptomNameEdt: EditText - symptomDescriptionEdt: EditText - addSymptomBtn: Button - readSymptomBtn: Button - deleteSymptomBtn: Button - updateSymptomBtn: Button + SymptomdbHelper: SymptomDBHelper	+ id: int + name: String + description: String + date: String
+ SymptomDBHelper(Context) + onCreate(SQLiteDatabase): void + addNewSymptom(String, String, String): void + deleteSymptom(String symptomName): void +updateSymptom(String, String, String, String): void + readSymptoms(): ArrayList \<SymptomModal> + onUpgrade(SQLiteDatabase): void	+ SymptomFragment(): + newInstance(String, String): SymptomFragment + onCreate(Bundle): void + onCreateView(LayoutInflater, ViewGroup, Bundle) + onClick(View)	+ SymptomModal(String, String, String): + getId(): int + setId(int): void + getName():String + setName(String): void + getDescription(): String + setDescription(String) + getDate(): String + setDate(String): void

Figure 9: Class Diagram: Symptoms

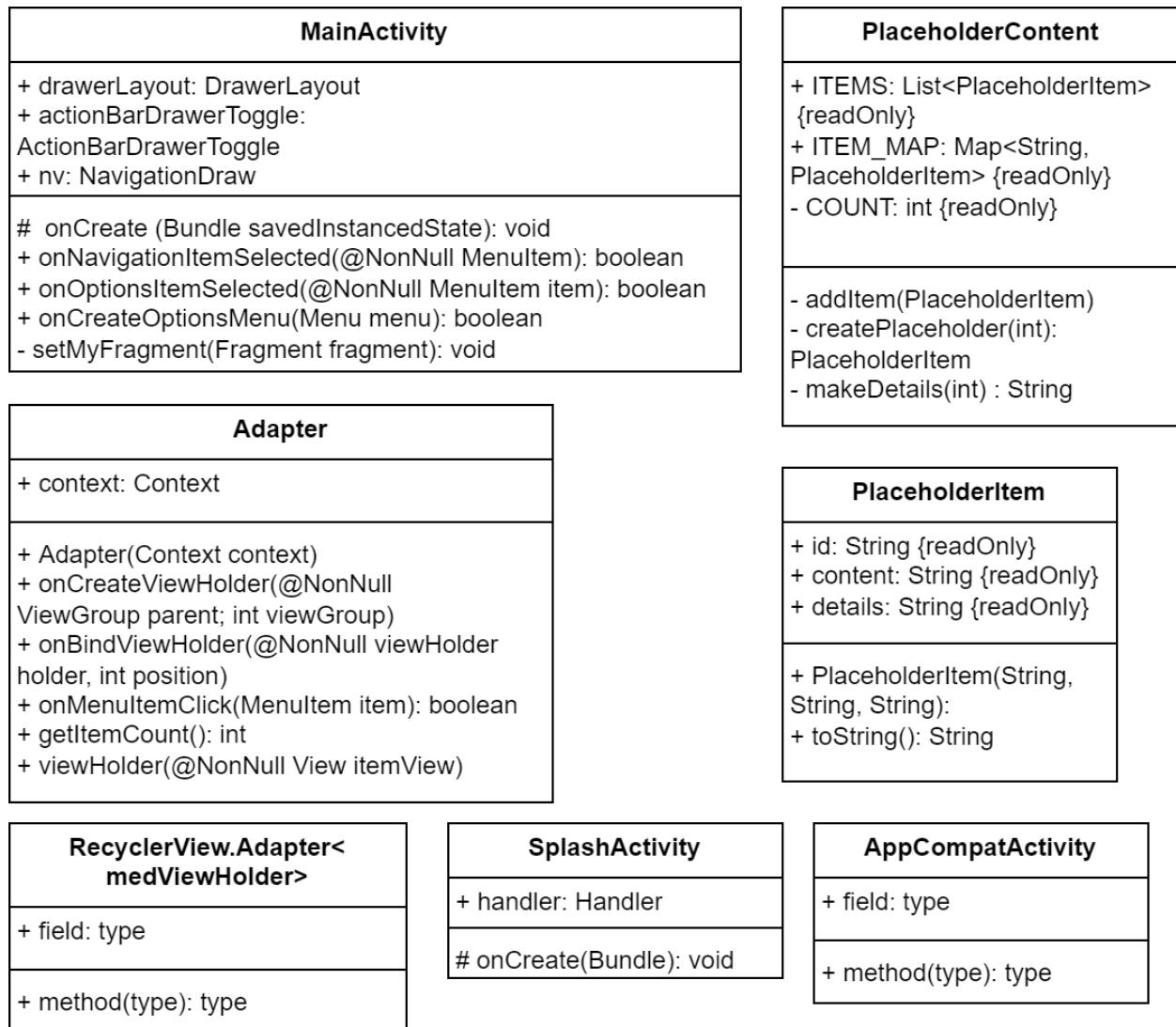Contains **all** classes relating directly Symptoms. Relationships included only as part of *Class Diagram: Medication.*

MainActivity

+ drawerLayout: DrawerLayout
+ actionBarDrawerToggle:
ActionBarDrawerToggle
+ nv: NavigationDraw

onCreate (Bundle savedInstancedState): void
+ onNavigationItemSelected(@NonNull MenuItem): boolean
+ onOptionsItemSelected(@NonNull MenuItem item): boolean
+ onCreateOptionsMenu(Menu menu): boolean
- setMyFragment(Fragment fragment): void

PlaceholderContent

+ ITEMS: List<PlaceholderItem>
 {readOnly}
+ ITEM_MAP: Map<String,
PlaceholderItem> {readOnly}
- COUNT: int {readOnly}

- addItem(PlaceholderItem)
- createPlaceholder(int):
PlaceholderItem
- makeDetails(int) : String

Adapter

+ context: Context

+ Adapter(Context context)
+ onCreateViewHolder(@NonNull
ViewGroup parent; int viewGroup)
+ onBindViewHolder(@NonNull viewHolder
holder, int position)
+ onMenuItemClick(MenuItem item): boolean
+ getItemCount(): int
+ viewHolder(@NonNull View itemView)

PlaceholderItem

+ id: String {readOnly}
+ content: String {readOnly}
+ details: String {readOnly}

+ PlaceholderItem(String,
String, String):
+ toString(): String

RecyclerView.Adapter<
medViewHolder>

+ field: type

+ method(type): type

SplashActivity

+ handler: Handler

onCreate(Bundle): void

AppCompatActivity

+ field: type

+ method(type): type

Figure 10: Class Diagram: Other

Contains all classes **not** already included in the other diagrams above. Relationships included only as part of *Class Diagram: Medication*.

17

10 State-machine diagram
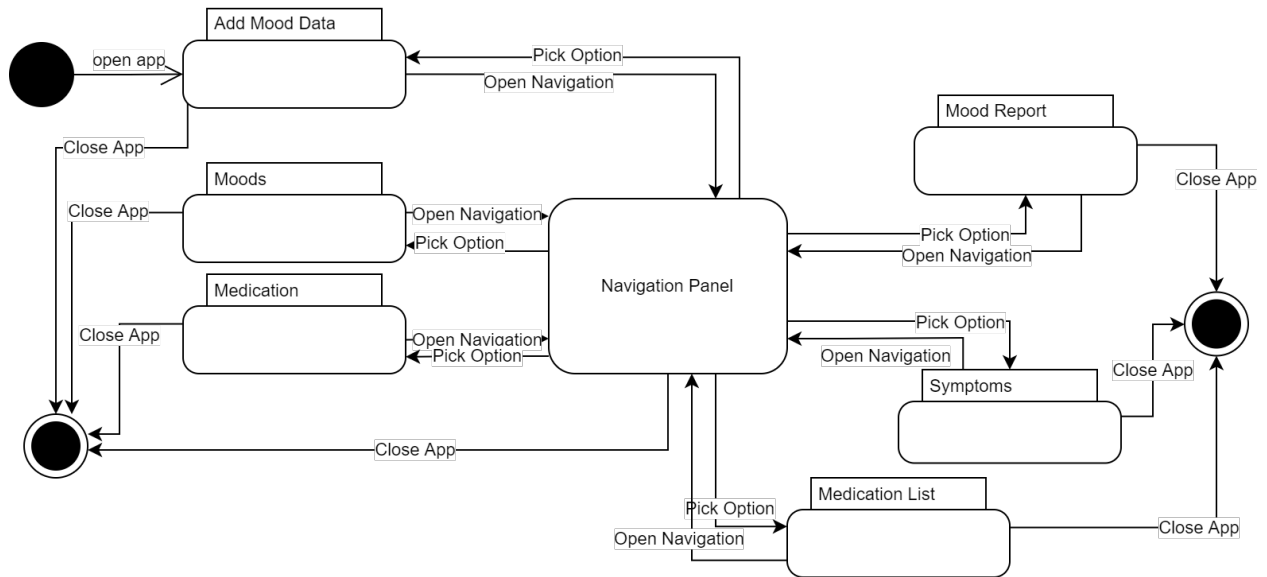
Figure 11: State Diagram: Whole System

Contains the state diagram of the whole system. Each specific screen in the UI interface being its own composite state. Each composite state has substrates, however they all follow the user interaction → system reaction pattern.

11 Entity Relationship Diagram (Data modelling)
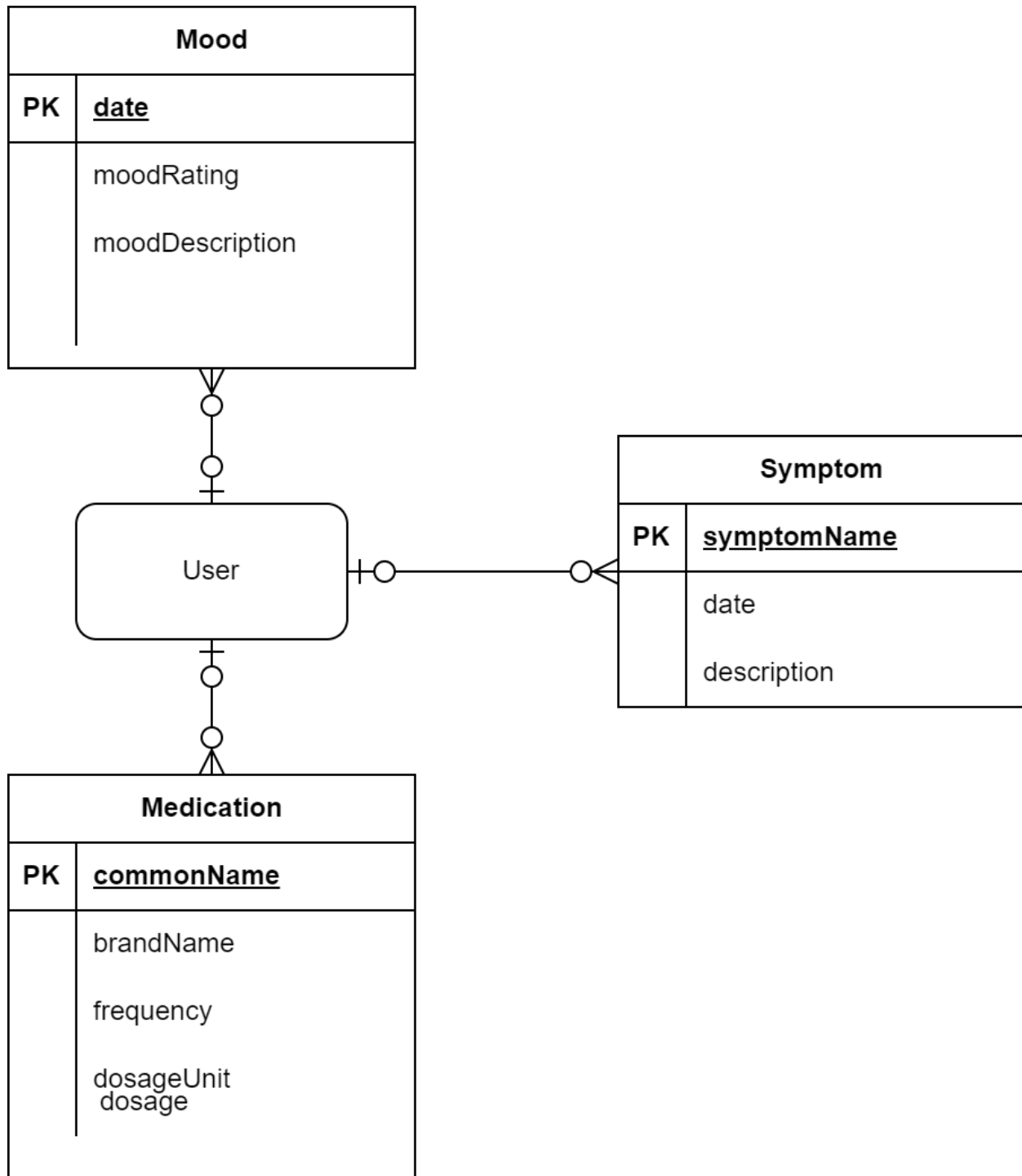
Figure 12: Entity Relationship Diagram

The entity relationship diagram (Data modelling) for the system.

12 GitHub Link

You MUST ask Yonatan Alexander for access, as this repository is set to private.

Group Therapy-Support-App

13 Conclusion

In many ways, the project did not go according to plan. We made a number of mistakes, but we believe the biggest was not making the project in Kotlin. Making the project in Kotlin would have allowed us to make the app's UI programmatically, which would have been a huge improvement. Additionally, this project turned out to be far larger in scope than we anticipated, simply because we were learning as we went.

In summary, many mistakes were made, but making such mistakes in a project where the intention is learning is exactly the right time to make them - and therefore we believe that those mistakes were (overall, at least) a positive.

14 References

1 **Smartpatient** (nd) *MyTherapy* (app): "Your personal pill reminder and medication tracker app", MyTherapy

2 **Bipolar UK** (nd) *Bipolar UK's Mood Tracker* (app): "Our new Mood Tracker app can make it much easier to record your daily mood, medications, emotions and how much sleep you've had", Bipolar UK

Appendices

A Project WBS and Gantt Chart

A.1 Project WBS

Spread over a number of pages due to Project Libre's many limitations
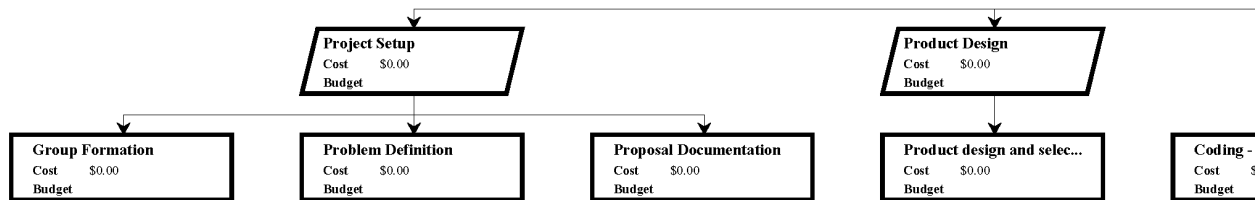
Figure 13: WBS Part 1 out of 5
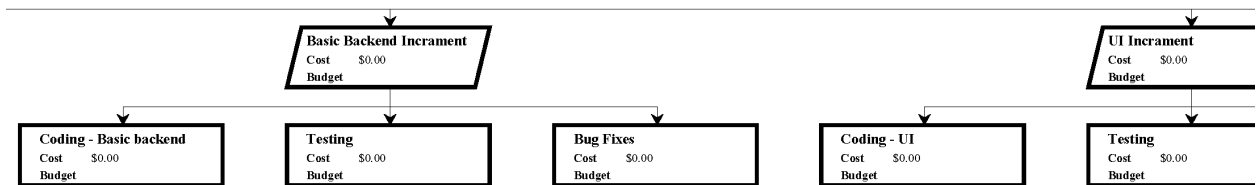
Figure 14: WBS Part 2 out of 5
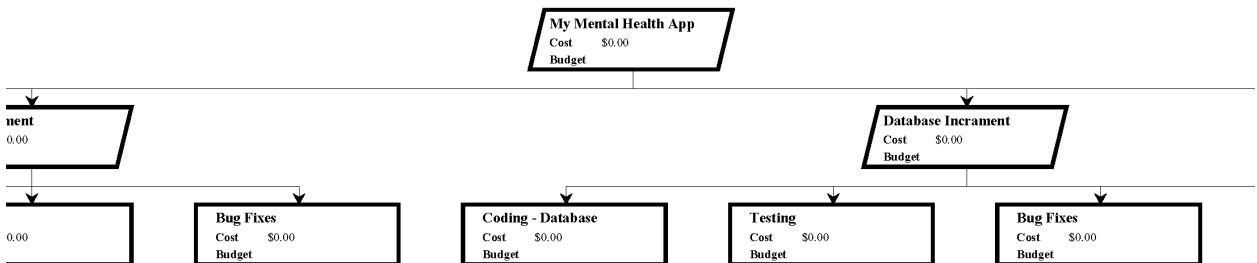
Figure 15: WBS Part 3 out of 5
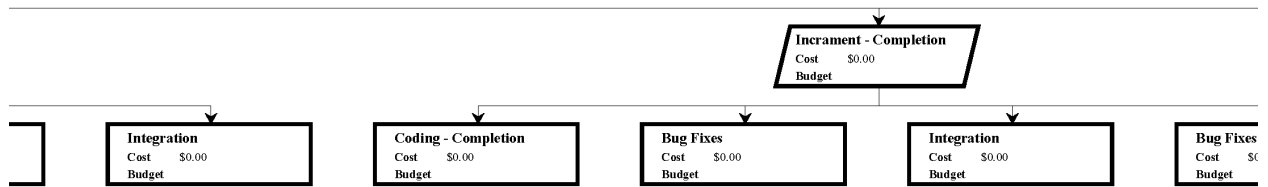
Figure 16: WBS Part 4 out of 5
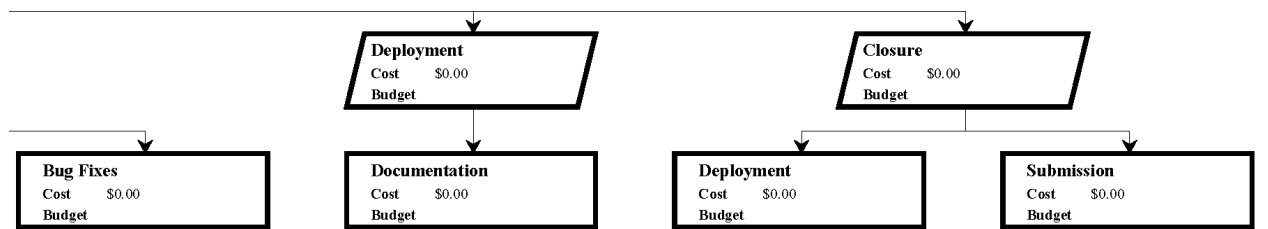
Figure 17: WBS Part 5 out of 5

A.2   Project Gantt Chart

Figure 18: Gantt Chart
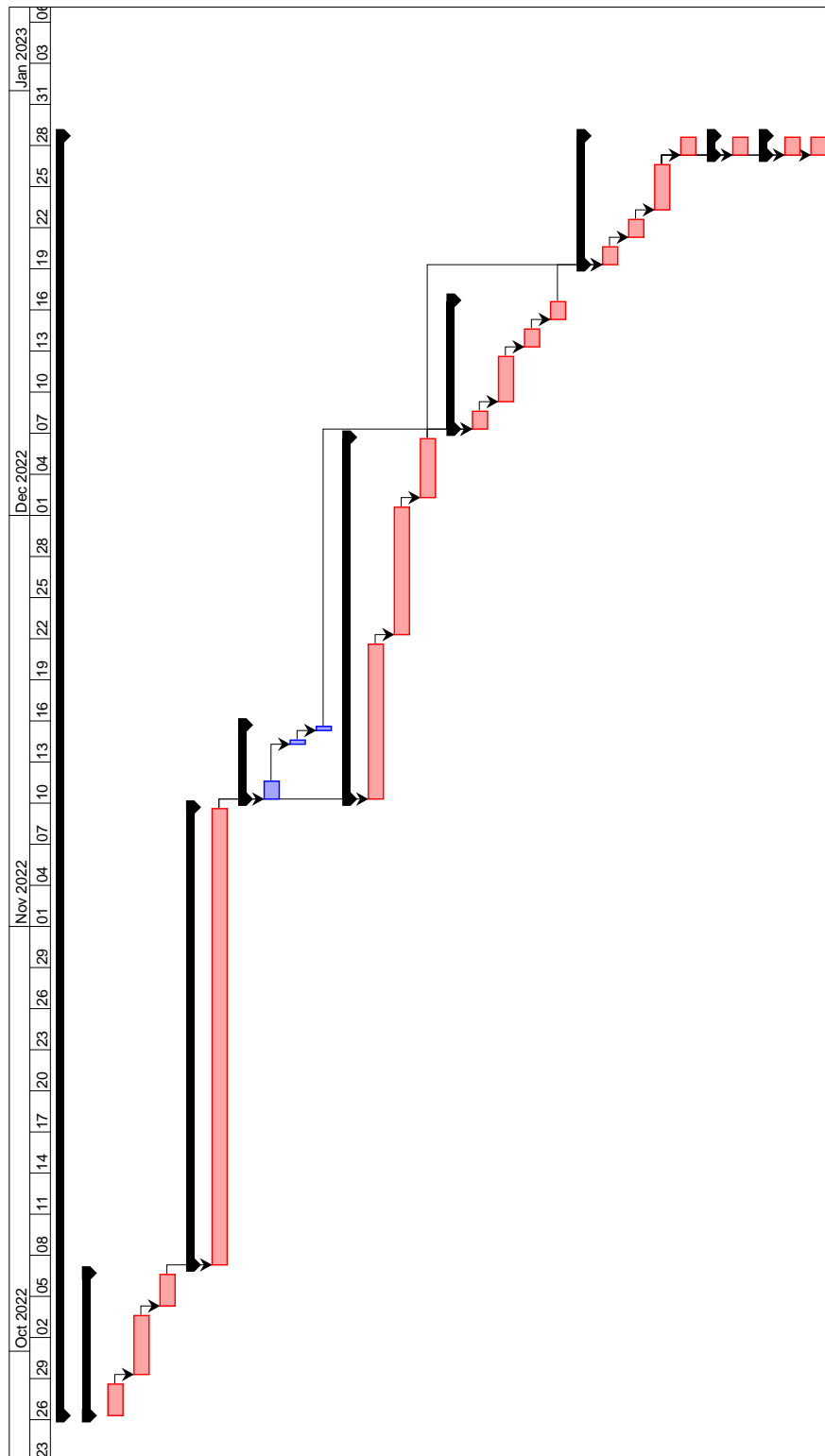
B   Task Assignment Matrix

Task	Task Owner	Support
Basic UI User Interfaces	Yasin	Yonatan, Michael
User Stories for Project UI	Yasin	Yonatan, Michael
Advanced UI User Interfaces	Michael	Yonatan
Front end (UI) completion	Michael	Yonatan
WBS	Yonatan, Michael	
Gantt Chart	Michael, Yonatan	
Task Assignment Matrix	Yonatan, Michael	
Testing Matrix	Michael	Yonatan
Skeleton Code	Yonatan	Michael
Database Integration with Backend	Yonatan, Michael	
Database Execution	Michael	Yonatan
Front end Integration with Backend	Michael	Yonatan
Use Case Diagrams	Yonatan, Michael	
Use Case Descriptions	Michael	Yonatan
Sequence Diagrams	Yonatan, Michael	
Class Diagram	Yonatan, Michael	
Data Entity Relationship Diagram	Yonatan, Michael	
State Diagrams	Michael	Yonatan
Video	Yonatan	Michael
Slideshow	Michael, Yonatan	
Report Editing	Yonatan	Michael

Table 3: Task Assignment Matrix

This table reflects what happened in practice, not what the plan was initially.

C The History of Commits

Does not contain the very latest commits, only the almost very latest commits.

Table 4: **Almost Complete Commit History**

Date	Time	Committer	Note
03-12-2022	03:13	Jon-AL	This time, really the Last Update :)
03-12-2022	02:32	Jon-AL	Last Update :)
03-12-2022	02:14	Jon-AL	tiny updates
03-12-2022	02:12	Jon-AL	deleted uneeded files (WBS png and svg)
03-12-2022	02:11	Jon-AL	main.tax updated to contain the Gantt chart and the WBS in a more accessible manner. File organization (preparing for submission), README.md update
03-12-2022	01:02	Jon-AL	README.md update
03-12-2022	01:01	Jon-AL	README.md update
03-12-2022	01:00	Jon-AL	README.md update
03-12-2022	00:51	Jon-AL	README.txt-¿README.md
03-12-2022	00:49	Jon-AL	Organizing the files
03-12-2022	00:46	Jon-AL	Starting organizing the files
03-12-2022	00:44	Jon-AL	Revert "Starting organizing the files"
03-12-2022	00:44	Jon-AL	Starting organizing the files
03-12-2022	00:40	Jon-AL	Starting organizing the files
03-12-2022	00:34	Jon-AL	Re-committed LaTeX
03-12-2022	00:33	Jon-AL	Merge branch 'main' into michael_workingbranch
02-12-2022	23:16	Jon-AL	Updated LaTeX in branch
02-12-2022	22:57	Michael	Updated comments.
02-12-2022	22:09	Michael	Merge branch 'main' of https://github.com/Jon-AL/4471—Therapy-Support-App
02-12-2022	22:06	Michael	pushed testcases
02-12-2022	21:55	Michael	Update README.txt
02-12-2022	21:49	Michael	Update README.txt
02-12-2022	21:44	Michael	Update README.txt
02-12-2022	20:59	Michael	Changed the text entered times.
02-12-2022	20:49	Michael	Changed the text entered times.
02-12-2022	19:57	Michael	Merge remote-tracking branch 'origin/main'
02-12-2022	19:57	Michael	Massive comment push.
02-12-2022	19:43	Jon-AL	Merge remote-tracking branch 'origin/michael_workingbranch' into michael_workingbranch
02-12-2022	19:43	Jon-AL	Project Report Updated (Nearly done), added class diagrams
02-12-2022	18:51	Michael	Added a build readme file.
02-12-2022	18:15	Michael	Had to change the gradle file.
02-12-2022	15:38	Michael	Bugs were patched after the introduction of test case file.
02-12-2022	11:11	Michael	Bug fixes have been fixed, and app modifications have been changed. The home screen is now the add screen. Defensive statements have been added to reject non-valid empty fields.
			Continued on next page

Table 4 – continued from previous page

Date	Time	Committer	Note
02-12-2022	00:21	Michael	Merge branch 'michael_workingbranch'
02-12-2022	00:20	Michael	fragment bug fix.
01-12-2022	23:32	Jon-AL	Merge remote-tracking branch 'origin/michael_workingbranch' into michael_workingbranch
01-12-2022	23:32	Jon-AL	Project Report Updated, added class diagrams
01-12-2022	23:24	Michael	Merge branch 'michael_workingbranch'
01-12-2022	23:15	Michael	rcmenu was forgotten. it needs to be placed inside there.
01-12-2022	23:14	Michael	Cleaned up code and commented some left behind functions. More comments can possibly happen tomorrow or tonight.
01-12-2022	20:48	Michael	Merge branch 'michael_workingbranch' of https://github.com/Jon-AL/4471—Therapy-Support-App into michael_workingbranch
01-12-2022	20:48	Michael	Revised Detailed Class Diagram
01-12-2022	20:18	Jon-AL	Project Report Updated
01-12-2022	20:01	Jon-AL	Project Report Updated
01-12-2022	19:08	Michael	Updated comments with partially finished UML diagram.
01-12-2022	18:06	Michael	storing current changes so nothing is lost.
01-12-2022	15:01	Michael	Updated use case and sequence diagrams.
01-12-2022	11:16	Michael	Updated comments for mooddb helper and fragment had its refresh button resized to make the UI cleaner.
01-12-2022	11:07	Michael	Updated all the medication views to make it look much nicer.
01-12-2022	09:30	Michael	Updated medication view. delete and update methods are buggy atm.
01-12-2022	01:44	Michael	Fixed medicaTION VIEW
30-11-2022	22:07	Michael	Cleaned up portions of code.
30-11-2022	19:34	Michael	A lot of code except reports has been commented.
30-11-2022	18:16	Michael	UI rebuild.
30-11-2022	17:31	Michael	Suggestions of removing bar and adding comments added in this version.
29-11-2022	01:09	Michael	A simple bar graph has been pushed into the mood report code. ty yonatan
28-11-2022	23:07	Jon-AL	added gradle support for a couple of external libraries
28-11-2022	22:45	Jon-AL	small modifications to mood addition frag
28-11-2022	22:36	Michael	Mood had its update method split into the addition component. The read and delete methods are two different components.
28-11-2022	22:28	Michael	Mood had its update method split into the addition component. The read and delete methods are two different components.
28-11-2022	21:57	Michael	med list updated.
28-11-2022	21:53	Michael	med list updated.
28-11-2022	21:51	Michael	med list updated.
28-11-2022	21:19	Michael	Separated the add button.
28-11-2022	19:53	Michael	Updated data to revert crash. Pushed to save current work.
28-11-2022	17:31	Jon-AL	+ to mood

Table 4 – continued from previous page

Date	Time	Committer	Note
27-11-2022	21:32	Michael	Seek bar color is dynamically changing as the seek bar motion is changing.
27-11-2022	10:10	Michael	Seek bar is now working with the short description, rating, and long description. This progress is being committed to avoid the app crashing during the report development.
27-11-2022	06:48	Michael	Merge branch 'michael_workingbranch'
27-11-2022	06:46	Michael	Seek bar is now working with the short description, rating, and long description. This progress is being committed to avoid the app crashing during the report development.
27-11-2022	06:25	Michael	Temporary fix for mood.
26-11-2022	23:58	Michael	Mood has been significantly updated. The slider works with the new addition method. Read and delete are now functional. Delete has an updated primary key. Update is not working at this point in time.
26-11-2022	21:43	Michael	Updated the current plus to have a working a visual sliding bar as requested. Code and descriptions was pulled from Yonatan's code.
26-11-2022	20:38	Michael	Merge branch 'michael_workingbranch'
26-11-2022	20:29	Michael	This is protect what work we have from changes.
26-11-2022	12:26	Jonathan	added seek bar, minor updates
25-11-2022	10:17	Jon-AL	PPTX
24-11-2022	22:06	Jon-AL	Corrected a color
24-11-2022	22:03	Jon-AL	Corrected a color
24-11-2022	20:28	Jon-AL	Merge branch 'jonathan'
24-11-2022	16:48	Jon-AL	New Logo New color scheme
24-11-2022	12:20	Jon-AL	Switching branch, requires commit first.
23-11-2022	17:59	Michael	Updated Use Case diagrams
21-11-2022	19:26	Michael	Medication code has been created. Update, add and delete seem to work but the view does not fit.
21-11-2022	19:12	Michael	Medication code has been created. Update, add and delete seem to work but the view does not fit.
21-11-2022	16:57	Michael	Medication Modal and Medication DB code modified.
21-11-2022	15:41	Michael	Symptom base feature has been established, created, and implemented db as the prof suggested.
21-11-2022	14:20	Michael	Fragments medication and symptom have been started. No connection as of yet.
21-11-2022	14:12	Michael	Modals have been created for base features Medication and Symptom.
21-11-2022	13:24	Michael	All initial features are done for mood.
18-11-2022	09:24	Michael	VCiew and delete method being worked on.
18-11-2022	08:31	Michael	View method being worked on.
18-11-2022	00:37	Michael	Adding data is possible now. Reading the data is crashing. Delete method is not up yet.

Table 4 – continued from previous page

Date	Time	Committer	Note
16-11-2022	21:45	Michael	App is no longer crashing. Fragment layout is displaying but stuck on how to change the activity into a fragment.
14-11-2022	19:31	Michael	Something keeps shutting the app down when the database is installed. Do not run at this time.
14-11-2022	14:21	Michael	Merge branch 'main' into michael_workingbranch
12-11-2022	20:41	Mohammad Yasin Aziz	First Commit. Commit No 1. Created skeleton UI so that the backend dev can start working. 12 Nov 2022
07-11-2022	20:50	Michael	Database has an updated delete method for a single record, and a method for dropping the table. Experimental.
07-11-2022	11:20	Jon-AL	Added the 'loading' screen
07-11-2022	09:33	Jon-AL	Large commit; changed organization of files slightly (androidTest-¿main). Added new activity for main menu.
06-11-2022	22:41	Mohammad Yasin Aziz	Add files via upload
06-11-2022	20:43	Michael	Updated use case and entity relationship diagram.
06-11-2022	20:42	Michael	Previous upload was not correctly updated. My bad.
06-11-2022	20:41	Michael	Updated use case diagram. An addition of the entity relationship diagram (rough copy).
06-11-2022	16:37	Michael	Important line for writing to a database had to be written in AndroidManifest.xml
06-11-2022	16:36	Michael	Updated a sample interface on how the user would interact with the app.
06-11-2022	16:01	Michael	Use Case diagram change
06-11-2022	15:59	Michael	Cleaned up mood db and symptom db to remove db table ambiguity. An additional variable was created to make symptom db more representative of current database labelled inside diagrams.
05-11-2022	18:43	Michael	Mood db for now. Functionality will come later.
05-11-2022	18:34	Michael	MedicationDBHelper.java has removed a useless library and variable comment.
05-11-2022	16:19	Michael	Further comments at the start.
05-11-2022	16:18	Michael	Initial creation of MedicationDBHelper.java. This class will support medication for now. Base feature of creating and adding the database has been established.
04-11-2022	14:26	Jon-AL	Medication - draft class to be used by graphical interface to display medications.
04-11-2022	14:26	Jon-AL	NumericMoodCheckIn - class to be used by graphical interface. Does it's own conversion from number to descriptors (short, long). To be made into a UI component for self reporting. Text take from the app "Bi-Polar UK", may need to be changed/updated according to the current psychological standards.
02-11-2022	17:57	Jon-AL	re-synced patterns
02-11-2022	17:55	Jon-AL	Updated visuals in title
			Continued on next page

Table 4 – continued from previous page

Date	Time	Committer	Note
02-11-2022	10:18	Jon-AL	Updated visuals in title
02-11-2022	09:31	Jon-AL	Merge remote-tracking branch 'origin/main' into jonathan
02-11-2022	09:30	Jon-AL	Updated visuals in title
01-11-2022	23:05	Michael	Merge pull request #2 from Jon-AL/michael_workingbranch
01-11-2022	23:04	Michael	Merge pull request #1 from Jon-AL/jonathan
01-11-2022	17:01	Jon-AL	Updated visuals in title
01-11-2022	16:55	Jon-AL	Updated visuals in title
01-11-2022	12:31	Jon-AL	design pattern templates for future reference (src: refactoring_guru)
01-11-2022	11:50	Jon-AL	App bones commit
01-11-2022	11:49	Jon-AL	Main file
01-11-2022	11:27	Jon-AL	Main file
01-11-2022	11:23	Jon-AL	Main file
01-11-2022	10:59	Jon-AL	Main file
07-10-2022	13:05	Jonathan	Updated sections, TOC, started on Bib
07-10-2022	13:04	Jonathan	Updated sections, TOC, started on Bib
07-10-2022	10:33	Jonathan	Main file
07-10-2022	10:32	Jonathan	Merge branch 'michael_workingbranch' of https://github.com/Jon-AL/4471—Therapy-Support-App into jonathan
07-10-2022	10:28	Jonathan	Main file
05-10-2022	17:37	Michael	Add files via upload
05-10-2022	17:34	Michael	Add files via upload
04-10-2022	13:38	Jon-AL	Main file
04-10-2022	13:26	Jon-AL	Merge remote-tracking branch 'origin/main'
04-10-2022	13:26	Jon-AL	Deleted duplicate file
04-10-2022	13:25	Jon-AL	Deleted duplicate file
04-10-2022	13:20	Jon-AL	Main file
04-10-2022	13:19	Jon-AL	Main file
04-10-2022	13:17	Jon-AL	Add files via upload

D   Minutes

D.1   26-Sep-'22

Present:

- Yonatan
- Michael

Not Present:

- Yasin

Discussed:

- Development proposal and development details.
- The project will be a therapy application that helps users gather, store and present their information to be used by patients.
- Suggested so far:
    - Incremental development.
    - Smartphone app.
    - Web app.
- Suggested Programming languages:
    - Dart.
    - Python.
    - Javascript.
    - C++.
    - Java.
- Stakeholders:
    - Clients.
    - Therapists and other mental health professionals.
- Features:
    - Mood self-reported values.
    - Medication tracking.

D.2   30-Sep-'22

Present:

- Yonatan
- Michael

Not Present:

- Yasin

Discussed:

- Project proposal was accepted.

- Instructor mentioned a number of points during lecture in the context of our project:
    - Patient data requirements with respect to privacy.
    - Identifying the stakeholders with who should be able to access data.
    - We need to identify the use cases carefully of what we are able to do.
    - We have to be able to clearly define the project objective and description.

- An aggressive timeline has been suggested to keep with coursework. This timeline is subject to change due to other courses requiring our attention.

- Github was setup by Yonatan.

- Report is in progress.

D.3 04-Oct-'22 and 05-Oct-'22

Discussed:

A timeline has been suggested to be finishing up to the user acceptance criteria. QT has been suggested as a platform to build an Android app.

The report has been started in LaTex by Yonatan. Proposed statements will be made by the group once we reviewed the work.

D.4 14-Oct-'22

Present:

- Yonatan
- Michael
- Yasin

Discussed:

Debate about the code is that we have no firm agreements on project libraries and frameworks. Suggestions include: salesforce as the backend. Flutter or ReactNative has also been suggested. Qt.

D.5 21-Oct-'22 to 28-Oct-'22

A lot of project will be developed on the reading week. Professor wants us to use Docker, and create a Gantt chart. He explicitly states that he wants us to assign backup roles to our task assignment matrix.

D.6   01-Nov-'22

Present:

- Yonatan

- Michael

Not Present:

- Yasin

Discussed:

Planned architecture:

  - repository style for the databases.

  - Event-driven for reminders.

  - Component-based.

- Today's objective:

  - Architecture and design patterns were discussed.

  - Get the backend done behind the scenes.

  - Another meeting: 9:30 am to 11:30 am tomorrow (November 2nd, 2022), want to start writing code.

D.7   02-Nov-'22

Present:

- Yonatan

- Michael

Not Present:

- Yasin

Discussed:

- We are switching the development process from incremental to a hybrid agile-iterative method.

- Scope and requirements changed significantly throughout the process.

- We are also incorporating agile methods to respond to changes over time.

- Kanban board has been created inside GitHub repository.

- User stories will be created for UI to make the interface clearer to write components and understand the problems.

- Drafts have been started but significant changes need to be created.

- An Android application will be created.

- The data will be stored locally with some encryption to get us started.

- Yonatan wants to push simple skeleton code, will require approval from Michael and Yasin.

- We want to finish baseline features executed and finished by November 6th.

- The list of things we want executed:

  - A basic UI - covered by Yasin (done).

  - Basic backend - covered by Yonatan

  - Database integration - covered by Michael

- We will need significant research need before implementations.

- No meeting will be scheduled for tomorrow.

- A few mini-meetings between Friday to Sunday.

D.8 02-Nov-'22

Present:

- Yonatan

- Michael

Not Present:

- Yasin

Discussed:

- Backend talk

- Significant modifications need to be executed for medication.

- We will have to split this into 3 main Tables:

  1. Medication info

  2. Medication inventory history

  3. Medication intake history

- Mood will be generated next.

- Symptoms will be generated last.

- Backend sample is expected to be done by Sunday late afternoon.

- Yasin will present a frontend demo on November 6th.

D.9 14-Nov-'22

Present:

- Yonatan

- Michael

- Yasin

Discussed:

- Front end and back end are being implemented. We are integrating today.

- Our approach has changed. We have recognized that the Model-View-ViewModel (MVVM) pattern was the best representation.

- Things to do:

  - Integrate front-end and back-end.

  - Revise databases.

  - Check on scope and extra features.

  - Testing and presentation notes to build.

D.10 24-Nov-'22

Present:

- Yonatan

- Michael

Not Present:

- Yasin

Discussed:

-

- Presentation is being worked on.

- Yonatan changed color scheme and logo.

- Things to do:

  1. He wants a better option for moods. A better option would be spinner or radio button. A perfect solution would be adding a slider [Seek bar].

  2. We need to focus more on accessibility.

  3. Add statistics.

D.11 27-Nov-'22

Present:

- Yonatan

- Michael

Not Present:

- Yasin

Discussed:

- Code must refactored to meet architectural standards.

- Code will be redone to the exact standard to the specified report

- Mood will be the focus.

- Medication and Symptoms are not the primary objective.

- Deadline to finish the code is tomorrow night.

- Deadline should be finished by Wednesday night.

- Deadline to finish the document components is Thursday.

D.12   28-Nov-'22

Present:

- Yonatan
- Michael

Not Present:

- Yasin

Discussed:

- Multiple meetings today.
- We have to go with MVC. MVVM was not feasible given the circumstances.
- Code is being redone to make the sections better.
- We want to get the code ready for deployment soon.
- Things to do:
    - Clean up redundant code
    - add comments
    - fix anything that seems broken.
    - Deploy on Docker.
    - Another addition of report documents and diagrams.

D.13   29-Nov-'22 until completion

Present:

- Yonatan
- Michael

Not Present:

- Yasin

Completed report, diagrams, and other loose ends.

Submitted on December 2nd, 2022.