Meilenstein III und IV

Projektverlauf

Die Umsetzung unserer Vorstellung des Projekts zu Beginn ist uns größtenteils gelungen. Die Klassenaufteilung ist beinahe komplett übernommen worden, mit Ausnahme der Benutzereinstellungen, des Services und der Notification. Ansonsten wurden die Klassen um einzelne Methoden erweitert. Außerdem wurde für eine bessere Übersicht in der MainActivity eine “Helper”-Klasse erstellt, in die (unter anderem) die Hilfsmethoden der OnClick-Methoden verschoben worden sind.

Benutzereinstellungen

Da der Aufwand höher als erwartet war, entschlossen wir uns dafür, Kürzungen am Projekt vorzunehmen. Das einzige Feature, das zunächst geplant aber nicht gefordert war, sind die Benutzereinstellungen. Daher wurde auf die Laustärkereglung des Alarms, Alarmtonauswahl und Farbeinstellung der LED zu den Alarmzeiten verzichtet.

setAlarm

Um den Alarm zusetzen, der den User daran erinnern soll, erneut eine Frage zu beantworten, wurde der AlarmManager genutzt. Dieser lässt zu den entsprechenden Alarmzeiten einen Ton erklingen und starten die App.

Startbildschirm

Mit der Einarbeitung in Android konnte der Startbildschirm zum Feststellen, ob ein Proband bereits initialisiert wurde oder nicht, weggelassen werden. Stattdessen übernimmt dies die onStart() Methode der MainActivity, die immer beim Aufrufen der App gestartet wird.

Control

Die Klasse Control wurde um folgende Methoden erweitert:

- changeLastAlarm(Time): void
- getUserTimes(): int[]
- wasLastQuestionAnswered():boolean
- userANsweredAQuatio(): boolean
- probandenCodIsEmpty(): boolean
- setNextAlarmAt(long): void
- isAlarmTime(long): boolean
- getTimeSinceLastAnswer(): long
- isQuestionInputOK(String): boolean
- generateQuestionText(): String
- createCSV(): boolean

Richtlinien

- CamelCase-Notation
- Funktionen und Methoden kurz halten
- ToDos zur Dokumentation von noch umzusetzenden Features/offenen Problemen
- Umsetzung mit Code Reviews sichergestellt

Unit Tests

Da es leider Probleme mit der Durchführung von JUnit TestCases gab, konnten die erstellten Tests nicht durchgeführt und bewertet werden. Entweder fehlte die Source-Datei für die JUnit Testcases oder die Tests wurden nicht durchgeführt(überprüft mit 2 Tests, einer sollte funktionieren und der andere sollte eine Fehlermeldung schmeißen).
Bei den Unit Tests wurden folgende Module/Methoden getestet:

- onClick() beim button_probanden_code
    - Es wurden unterschiedliche Probandencode-Eingaben überprüft. Dabei wurden Leerzeichen, Ziffern und entweder längere oder kürzere Codes bei der Eingabe simuliert. Anschließen wurde überprüft, ob das Bestätigen des Codes auf das nächste Layout verwies oder nicht.
- Control.wasLastQuestionAnswered()
    - Hier wurde erst ein Eintrag einer Abfrage simuliert und anschließend nach unterschiedlichen Zeitintervallen diese Methode getestet. Die Zeitintervalle wurden so bestimmt, dass noch kein weiterer Alarm, min. ein Alarm und ein Tag verstrichen sind.
- Control.getTimeSinceLastAnswer()
    - Weil auch diese Methode von einem Zeitunterschied abhängt, wurde die ausgabe nach unterschiedlich langen Abständen überprüft.
- Control.isQuestionInputOK(String)
    - Diese Methode wurde mithilfe unterschiedlicher Parametereingaben auf den erwünschten Output kontrolliert.
- Control.generateQuestionText()
    - Bei dieser Methode wurden die unterschiedlichen Outputs nach bestimmten Zeitintervallen auf deren Inhalt untersucht. Dabei wurde möglichst jeder unterschiedliche Fall getestet.

Acceptance Testing

funktional

- Ändern der Zeiten ist möglich
- Das Schreiben in die Datei Probandencode.csv funktioniert
- Die Eingabe zur Anzahl und Dauer der sozialen Kontakte wird zu den eingegebenen Zeiten gefordert

nicht funktional

- Probandencode wird erklärt und die Eingabe wird kontrolliert
- Erstellen eines neuen Probanden ist einfach, es muss nur die Probandencode.csv vom Speicher entfernt werden
- Es ist sichergestellt, dass die Zeiten ohne Fehler bei der Eingabe gewählt werden können.

Code Review

Pair Programming

- besseres Verständnis untereinander
- schnelle Lösungsfindung bei Problemen
- besserer Überblick über das Gesamtprojekt
- gefundene Fehler: unter anderem in der Zeitbestimmung für den Alarm und in der Eingabekontrolle beim Probandencode

Walkthroughs

- besserer Überblick über das Gesamtprojekt
- Finden von Denkfehlern
- gefundene Fehler: unter anderem in der Speicherung der letzten und nächsten Alarmzeit bzgl. der Ausgabe in die Probandencode.csv

Aufgabenverteilung

Implementierung

- Andreas Petrow: Views, Inputkontrolle, Tests
- Fabian Broeg: Alarm
- Anne-Lena Simon: Datenverwaltung, CSV-Output

Dokumentation

- Andreas Petrow: Projektverlauf, Testcases
- Fabian Broeg: Acceptance Testing, Umsetzung der Richtlinien, Code Review
- Anne-Lena Simon: Codedokumentation, ReadMe