Capstone I App Requirements

Samiksha Mailarpwar and Alex Lambrecht

Advisor: Dr. Goldwasser

For our Senior Capstone project, we are collaborating with a professor in the School of Public Health and Social Justice, Dr. Enbal Shacham, who is conducting a study involving alcohol use and sexual consent on SLU’s campus. To appeal to the participant pool, we named the app “Boozymeter” which addresses the main goal of the app. For the researcher, the app is a tool to analyze alcohol use on campus. At a high level, the app basically logs participants’ answers into a secure database which will later be used to provide the researcher with a web-facing portal for the analyzed data in Capstone II. The platform used for the development and testing of this app is Android Studio.

The first issue we addressed was the requirement for a secure database. We had three options to choose from: Amazon Web Services, a local server, and Firebase. For the purposes of this app, we chose Firebase for the purposes of easy access, easy use, and secure storage of data. Firebase provides us with a separate authentication database and another database that stores real-time data. This cloud service also provides Dr. Shacham with an easy way of adding or removing users - one that would require a simple login to the Google account.

Participants will be provided with a unique username and password. We are not including a “Sign-up” feature to prevent unknown participants from downloading and using the app if published on the Google Play Store. There will be a sign-out method, which is a basic feature in any app. When the user opens up the app, there are two possibilities. If it is the first time the participant is using the app on their phones, they will be directed to a Login screen where they have to enter their username and password. If they have already logged into the app, they will be directed to the Main Screen of the app, which will serve as the starting point for the participant everyday. To protect the security of the participants, the data will be stored under a unique userID which is only visible to Dr. Shacham in the authentication database. The JSON tree (Firebase’s database) will record the data as child nodes of the userID. Ideally, the emails of the participants and the userIDs will be stored in separate databases or even in separate locations. This protects the integrity of the data and will prevent any potential leak of participant data. To achieve this goal, we will be generating a list of unique strings that will serve as the participant identifier and will be used as the username while logging into the app. This way there will be layers to the data storage and tracing back to the participant’s email from the long string of letters (generated by Firebase) will be extremely difficult.

The Main Screen will have a “Start” button which the participant will have to press if they are drinking that day. Once the button is pressed, the app will passively start logging the location and timestamp every x minutes. The frequency of this passive activity is still to be determined by Dr. Shacham. Regardless, the idea is to track how frequently the participant travels during the night. The user will have to enable location services the first time they login to the app, which allows us to ensure that we have the proper permissions to passively track their location. The app will have to communicate with Google Play Services as well as communicate with the phone settings. Once location services is enabled, the user will not be involved in the tracking of their location throughout the night.

The next core component of this app is after the participant touches the “Start” button on the Main Screen. Pressing the button will start a timer for 30 minutes. The participant is supposed to log every drink they have in the app. However, if the app does not hear back from the participant in 30 minutes, the app sends out notifications to the participant. If the phone is locked, the notification will take up the lock screen (using the pre-built capabilities in an Android phone). If the phone is unlocked, the app sends push notifications at the top of the screen to remind the user to log their activity. After the timer goes off 3 times (90 minutes later) and there is no user-activity, the app will stop sending notifications, assuming that the user is no longer continuing their consumption of alcohol.

Every time the user logs a drink, the app will walk the participant through screens that will ask the user to specify the type of drink and size of drink they just consumed. Keeping in mind that the participants are under the influence of alcohol, these screens will be clutter-free and user friendly. There will be no scrolling or typing - just clicking buttons. All of this requires the user to actively participate and open the app. The app will passively log all of this data in the database under the specific userID. This process will continue for 1-2 weeks, depending on however long Dr. Shacham wants to run the study per participant.

The last part of the app, but of a lower priority, is a morning questionnaire and daily report. The morning questionnaire will contain questions regarding the participant’s previous night in terms of alcohol intake as well as sexual activity. We are not stressing the sexual consent portion of this app as of now because Dr. Shacham is unsure of whether or not she will be able to have approval to ask questions specific to that topic. The answers to these questions will be logged in the database along with the timestamp to indicate when the questionnaire was taken. The app will send a notification to remind the user to complete the questionnaire. The timing of the notification will be dictated by the timer and Dr. Shacham will tell us an exact time she would like the notification to pop up on their login screen. After the participant completes the post-night survey, if we have time, we will create a mini-report that highlights the previous night in terms how many times they moved, how many drinks they had, how many calories they consumed, etc. The report, although a nice feature to have, is the lowest task on our backlog primarily because we need to first focus on the core components of the app before we add other features.