SafeDriving Android Application

- Nandinii Yeleswarapu & Shivam Gupta

- May 21st, 2021

Project Statement

➔ Application purpose and a summary of features:

The SafeDriving application we developed in this project is an android based Blood Alcohol Content (BAC) calculator which uses the Widmark formula to calculate a user's BAC levels based on their weight, the drinks they have consumed and the time elapsed since they began drinking. SafeDriving allows users to use the application as guests, thereby discarding any information they enter into the application while using it once their results have been displayed. SafeDriving also allows users to sign in using Google SignIn, allowing the application to store recurrent information such as the user weight and gender into a database, thus saving the user the effort and time of reentering such information whenever they use the app.

Once base information has been collected from the user or retrieved from the user database, SafeDriving allows the user to choose the type and number of drinks they've consumed from a list of options. The users are also prompted to enter the time elapsed since they started drinking. The application then calculates the user's BAC and displays the results along with a conclusion on whether it is safe for the user to drive or not based on the legal driving limit for BAC in Massachusetts.

- ➔ Motivation behind developing the SafeDriving App:

Given that alcohol consumption is a part of most people's lifestyle including as part of regular meals, oftentimes people are unable to keep track of exactly how much alcohol they have consumed. Given alcohol has different behavioral effects on different people, sometimes people are unable to make an informed decision as to whether their judgement is too impaired or whether they are in a condition to safely drive. The objective and motivation for this application is rooted in an effort to reduce the number of unintentional and unaware people who have consumed more alcohol than the legal limit for safe driving from driving in effect reducing the number of DUIs sanctioned and hopefully the number of road traffic accidents caused by DUIs.

Application Design:

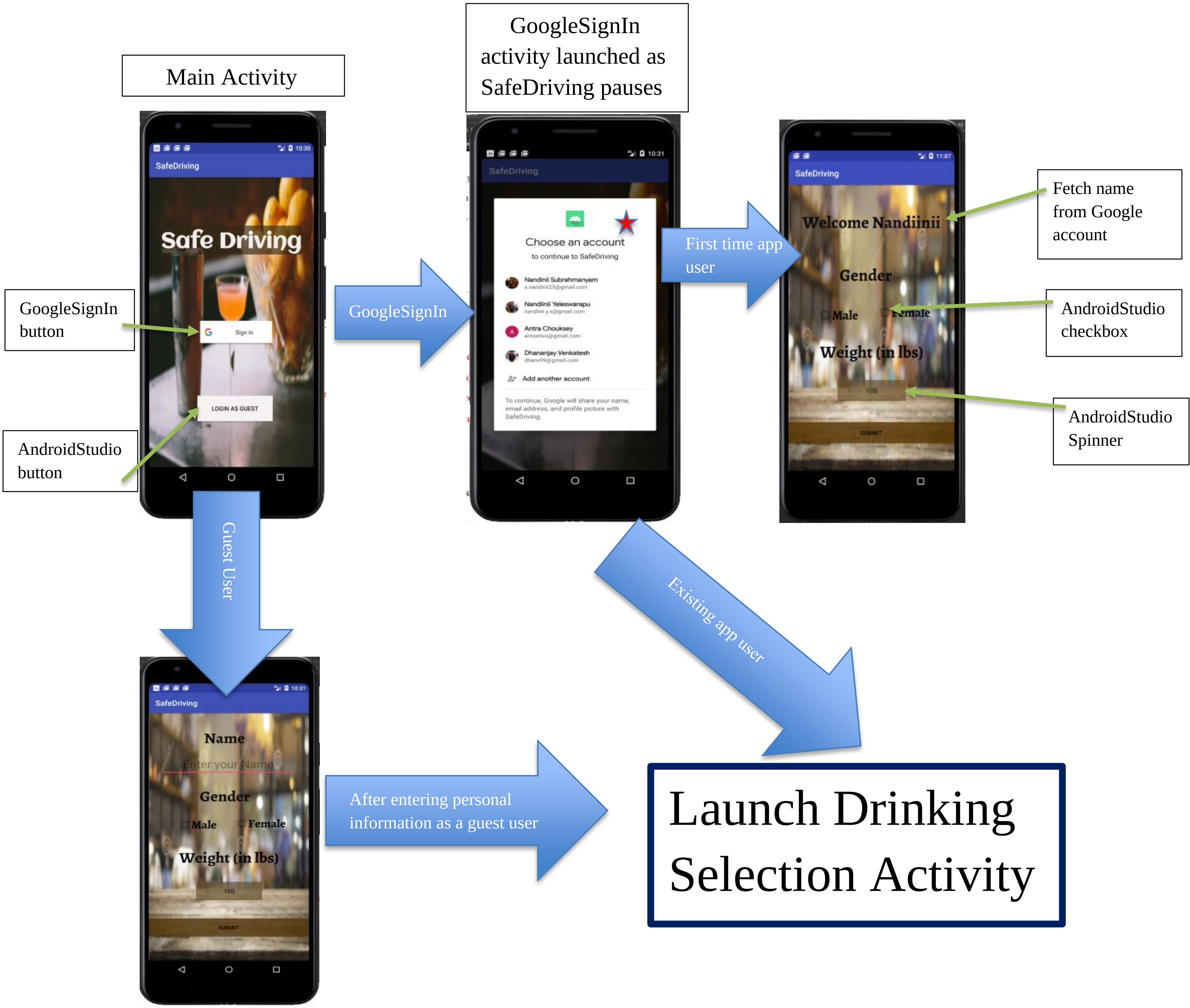
- ➔ Application User Interface:

The application, designed to be used on smartphones, can be categorized into four major aspects:

1. Welcoming the user:

   The first aspect of the application UI is welcoming the user. SafeDriving allows a user to use the application either as a signed in user (sign in using their Google account credentials) or as a guest user.

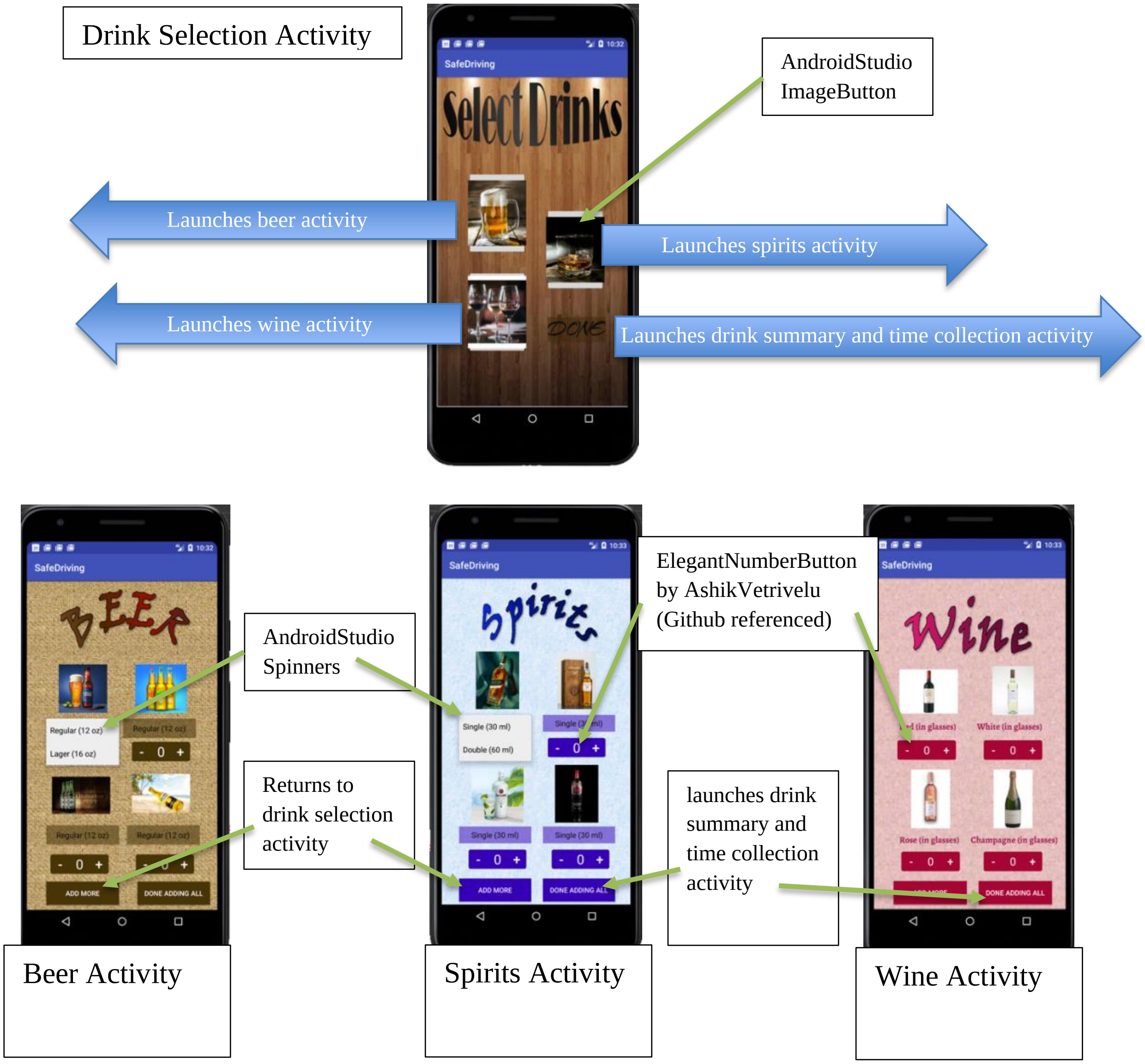
   If the user signs in using Google SignIn, the app uses the user's google account id to check if the user database has their information. If yes, the user's weight and gender is queried from the database and used for the remainder runtime of the application. If no, their google account id is collected and an activity is launched to collect their name, gender and weight information. This information is stored in an SQLlite powered database for the next time the user uses the application using their Google account. If the user chooses to use the app as a guest, then the app launches an activity which collects their name, gender and weight. This information is used until the results are calculated and displayed for the user and then discarded.



★ Permissions have been obtained from the various google account holders to display their emails in this report.

2. Drink Selection:

The second aspect of the application UI is the drink selection for the user. The user is provided with a screen of three images, one corresponding to wine, one to beer and one to spirits. Clicking any of the images launches an activity which provides a selection of drinks corresponding to the type of alcohol they selected in the previous screen. Under each image there is a selection of customization options such as the size of the drink and the number of said drink and size consumed. The user is given a choice to return to the previous menu to keep adding more drinks and also an option to click when the user is done adding all the drinks.



Drink Selection Activity

AndroidStudio ImageButton

Launches beer activity

Launches spirits activity

Launches wine activity

Launches drink summary and time collection activity

AndroidStudio Spinners

ElegantNumberButton by AshikVetrivelu (Github referenced)

Returns to drink selection activity

launches drink summary and time collection activity

Beer Activity

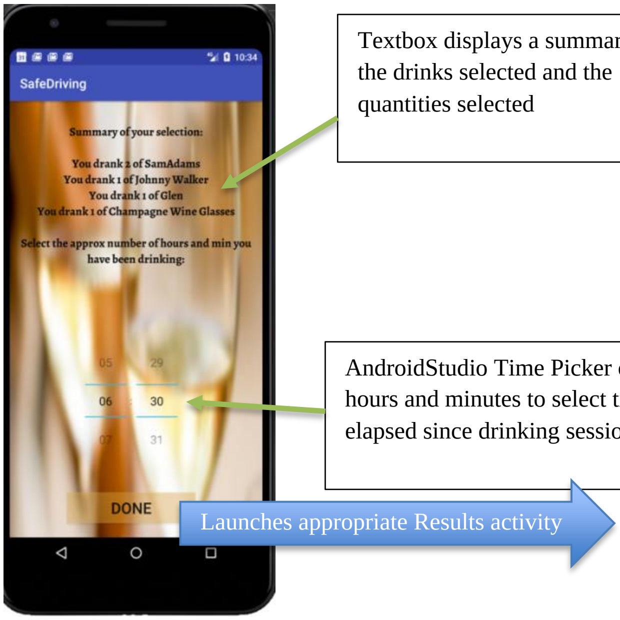
Spirits Activity

Wine Activity

3. Drink summary and time collection:

The third aspect of the app's UI launched after drink selection is complete is the drink summary and time collection activity. In this activity, a summary of all the drinks the user selected are summarized and the user is asked to enter the time elapsed in hours and minutes since the user began drinking. Once complete, the results are ready to be displayed.

Drink Summary and Time Collection Activity



Summary of your selection:

You drank 2 of SamAdams
You drank 1 of Johnny Walker
You drank 1 of Glen
You drank 1 of Champagne Wine Glasses

Select the approx number of hours and min you have been drinking:

Textbox displays a summary of the drinks selected and the quantities selected

AndroidStudio Time Picker on hours and minutes to select time elapsed since drinking session

Launches appropriate Results activity

4. Results and further options:

Once the third aspect is complete, the results are calculated on the backend using all the collected information from the previous aspects. For the BAC calculation, the Widmark formula (referenced below) is used:

$$BAC = \frac{AlcConsumed(volume\ \&\ abv\ considered) \times 5.14}{weight \times genderConst} - 0.015 \times timeDrinking$$

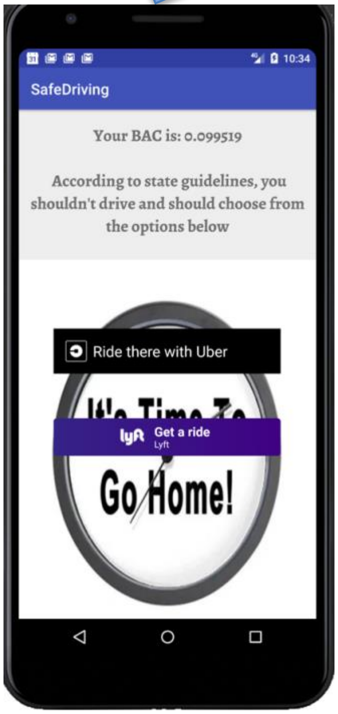
According to Massachusetts state laws, the safe BAC limit for driving is 0.08%. Thus, if the user's BAC is calculated as higher than 0.08%, the app displays the BAC calculated, a conclusion stating that the user shouldn't drive at the moment and links to launch ride-share apps such as Lyft and Uber.

If the user's BAC is calculated as lower than 0.08%, the app displays the BAC calculated, a conclusion stating that the user is within the legal safe limit to drive at the moment and a GoogleMap view allowing them to search their destination and launch the GoogleMaps app for navigation.

Legal MA safe driving BAC level = 0.08%

If BAC calculated is higher than 0.08%

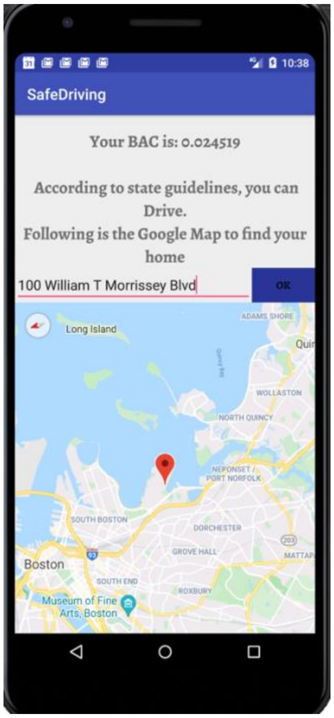
If BAC calculated is less than 0.08%



SafeDriving

Your BAC is: 0.099519

According to state guidelines, you shouldn't drive and should choose from the options below



SafeDriving

Your BAC is: 0.024519

According to state guidelines, you can Drive.
Following is the Google Map to find your home

100 William T Morrissey Blvd

➔ The implementation of technical app features:

1. GoogleSignIn:

   We used the Google API and the template code on the Google developer's website to integrate GoogleSignIn into our app. We had to add the relevant Firebase and PlayServices dependencies into our build file and configure our application to be able to obtain a clientID. This clientID is used by the app to connect to Google to be able to facilitate interaction with user information stored in Google accounts. From the user's Google account, we obtain and record the account ID associated with the account. This serves as the unique identifier we use to record user weight and gender in our database. We also record the given name associated with the Google account for first time users. This name is displayed in a welcome message on the screen that collects their weight and gender.

2. User profile database:

   We use a simple database with four columns powered by SQLLite. The four columns are: an auto-incrementing entry number primary key column, an account ID column, a gender column and a weight column. The database provides functionality for adding a user, updating a user's information, displaying all the users in the database and to query a given user whose account ID matches the one supplied to the function.

Application Evaluation:

The SafeDriving application was evaluated on two devices: the Pixel 3a with Android 5.1 running on the Android Virtual Device (AVD) emulator and the mobile device Samsung S10 with Android 9.0. We tested the application on the AVD multiple times using the Logcat to print various test messages to ensure proper running of the application. We allowed several of our friends to try using the application while it was running on the AVD to test various functionalities and to obtain feedback on the user-friendliness of the application. We also created an Android Application Package (APK) file and used it to install the application on our Samsung S10 device to have a more realistic understanding of how the application runs on a mobile device. We also allowed our friends to use the application once installed on the Samsung device.

One of the major obstacles we encountered while testing the application across various platforms was with the GoogleSignIn feature. Since the feature requires the configuration of a clientID into the application. Since the app is not published, each of our individual clientIDs were working under the debug mode only on our own but not on any other. Thus, we were able to test GoogleSignIn on the AVD and used various Gmail accounts to ensure log in was successful and that the user's information was successfully retrieved from our user database, but we weren't able to test GoogleSignIn on the Samsung device. We were able to test all other features on the Samsung device by using the application as a guest.

Project experience:

The project was fun and we had a great time doing the project. We had many ideas to add to the functionality of the app, but some required a subscription or for the app to be published and others couldn't be done due to the time constraint. A few features that could be included into the application implementation would be:

1. Implementing a radial menu:

One of the features we wanted to incorporate was to try and implement our own Android widget, inspired by the Radial Menu design. This widget would be used for each category or type of alcohol we have where the circular menu opened would hold various brands or companies of alcohol. This would have made UI extra engaging, and we could also limit the space usage and add a large variety of drink options in the app.

2. An animation associated with the ImageButtons:

Another interesting feature would be adding animation to the app in which when a user clicks an ImageButton, the button does some animation while the activity loads in which the user gets to know that button has been clicked and the process has started and needed time to load.

3. User data updates:

As of now, the user has no interaction with the database directly and once added a user doesn't have the option to delete or alter their information stored in the database. A possible improvement for the future would be to allow users to both disconnect their Google account from the app and delete or update their stored information in the database.

References:

"Developers." *Uber*, developer.uber.com/docs/riders/ride-requests/tutorials/button/android.

Google, Google, developers.google.com/maps/documentation/android-sdk/start.

Head, William C. "Blood Alcohol Content Formula." *Find the Best DUI & DWI Lawyers Nationwide - TeamDUI.com*, 24 Feb. 2021, www.teamdui.com/bac-widmarks-formula/.

"Lyft Button." *Lyft*, developer.lyft.com/docs/android-lyft-button.

"MapQuest Development and Documentation - Getting Started." *MapQuest API Documentation*, developer.mapquest.com/documentation/.

"Start Integrating Google Sign-In into Your Android App." *Google*, Google, developers.google.com/identity/sign-in/android/start-integrating.