Dustin Lantz

5/20/18

Meal Record Write-Up

When I began pondering how to create an app that “benefits society” I quickly realized I would have to narrow my scope a bit. I considered issues that our society faces today, as well as issues I have personal experience with. An issue that sees large amounts of coverage among the population at large is obesity, or being overweight in general. Having been trying multiple dieting strategies these past 5 months myself, I sought to think of a way I could create an application to help aid those who wished to do something about weight. When I was first starting out with my dieting, I found it prudent to keep track of what I ate and when, and then began eliminating unnecessary meals and calories from my diet, or substituting them where necessary. I realized that an easy to manage, quick to change and update list always available in my pocket, even when eating out or in public, would be an excellent way to address this need.

In addition to the aforementioned basic function, I also recognized an opportunity to benefit society in a more niche way. Working at a food service establishment myself (Bridgewater Retirement Dining Services) I’ve spoken with the chefs there on multiple occasions about foodborne illnesses. As it turns out, some of the most dangerous foodborne illnesses are not only deadly, but incredibly difficult to trace back to their origin due to having long hibernation periods after the ingestion of the tainted food, such as certain seafood borne illnesses which can take up to a month before symptoms develop. If the application provided not only a place to store health information, but also a datafield in which to store information about where/by whom food was prepared/bought, and a space where additional information regarding the meals could be stored, then in the case of food poisoning or worse, the patient would have a tangible record of past meals to pull from in examining the likely causes.

In attempting to address all of these concerns, the Meal Record app was created. The Meal Record android application is capable of saving meal data to an internal sqlite database directly on the phone. These recorded meals consist of 4 data fields that the user has great freedom to track or ignore, in the interest of not forcing people to use the app for functions they do not want to. The only datafield required for the application to function is the name/date field, and its only stipulation is that all meal entries have unique names. The app calls attention to this fact both in the information screen navigated to from the home screen, but also by preventing the addition of meals to the database or the editing of their contents if an existing meal in the database already possesses the same name/date. The other three data fields that make up any entry in the database, Preparer, Calories, and Description, are all similarly explained in the information screen, but are not required fields. If the user so wishes, they may leave any or all of these fields blank when creating meals. While this would not be optimal use of the app, I believe that giving users flexibility is more important than forcing them to use the app for functions they do not want it for. If someone solely wishes to keep track of calories of meals for dietary purposes, then they may do so. If they wish to keep record of where they got certain meals and their thoughts on them, without worrying about the calories of the meal, such is their choice.

The application upon first being launched displays a simple login screen to the user. This login screen merely requires that the phone’s user input the password to continue, and the user is informed at the login screen that first time users have a default password of “password” and are encouraged to change this in the settings screen later. While I considered adding a list of users with matching passwords, the scope of the app as a personal list on a personal device, using a non-transferable sqlite database, made such an addition seem superfluous, and additionally there were some technical difficulties early on in implementing such a feature. Still, it is a feature I would/may revisit for the application in the future.

After the user has entered their password, they are taken to the home screen. The home screen has four points of navigation: About the App in the top left, Settings in the top right, Add Meals in the bottom left, and View Meals in the bottom right. This screen is the main navigation hub for the app, and most activities return to this screen after completing their functions.

The About App screen is reached by touching the question mark in the upper left of the home screen. This screen is fairly straightforward. Within is a short write-up that describes the application’s intended uses and functions to the user. It also explains the fields used to create meal objects. While not a technically impressive screen, it nevertheless felt like an important addition. The app needs to be able to provide for users of all familiarity with technology, and centralizing information seems like a good way to accomplish this goal.

The Add Meal button navigates to the MealCreator screen, where users are prompted to fill in the information necessary to add new meal objects to the database. As previously mentioned, this entry format is very flexible, allowing for any entry with a unique name/date. In the event a non-unique name/date entry is attempted, the screen will reject the submission, alert the user, and bring the screens focus back to the Name/Date field. The screen adds legal entries to the database when the “Confirm Meal” button is pressed, and returns the user to the home screen.

The View Meals button navigates to the ObjectList screen, where all of the databases current meal entries are displayed in a object list textfield. From here, users can scroll through entries, select entries by touching them, and then interact with the selected entries using the bottom buttons, “Edit” and “Delete”. If the edit button is pressed after a meal has been, it will launch the edit activity. If no meal was selected, the button does nothing. If the delete button is pressed after a meal has been selected, it will create an alert dialogue that warns the user, and prompts them to confirm their selection. If they confirm the deletion, the entry is wiped from the database, and the list is refreshed. If no meal was selected, the delete button does nothing.

The edit button navigates to the edit screen, where the information of the edited meal object is filled into appropriate editText fields for the user to make changes too. When they are finished, they can hit the Confirm Changes button to save their changes, and return to the ObjectList screen. If they attempt to confirm changes while the Name/Date entry is the same as an existing entry (other than itself) the application will deny the request, notify the user, and redirect the focus to the Name/Date field.

The Settings screen (named TrueSettings in the java files) is reached by pressing the wrench icon in the top right of the home screen. This screen displays two editText fields, and two buttons. The editText fields and the Confirm button can be used to change the password on the application. The two entries must be matching. If they are not, pressing the button will notify the user that they must be the same, and redirect focus to the New Password editText. The Delete All Entries button, when pressed, creates an alert dialogue that warns the user that this action is permanent, and prompts them for confirmation. If the user confirms the dialogue, the entire database has its entries cleared, the user is notified of the success, and is returned to the homescreen. This function is meant to be of use to those who have created a large amount of entries and want to clear the entire list of Meals at once instead of one at a time through the ObjectList screen.

Using the screens and options listed above, it is possible to keep a long running cache of organized and personalized information about one’s meals purchased/consumed for a myriad of purposes for the user, be it personal or medical. Additionally, there are a few features that I would like to add in the future, but were not working in this version of the application. Chief among these features is a way to search the database for a specific entry based on Name/Date, or perhaps even other criteria. At current, entries can only be examined by scrolling through the ObjectList screen, and may become difficult to find if there are many entries. Additionally, the ObjectList currently displays the oldest entered data first, and while it does not take long to scroll down to the newest additions, a setting to reverse their priority/weight in the list would be a welcome addition in the future, but was not working in time for this version of the app. Still, even in its current state this application has the potential to provide great benefits to society, provided people are willing to put in the effort and allow the application to aid them on their way.