Grant Farnsworth - grantfar@gmail.com

David Hatten - dhatten87@gmail.com

Price Tracker

Our initial goal with the project was to create a price tracker that would store multiple data points for products that a user would input. It would then offer the user the ability to upload that data to an online server for secure storage and tracking across devices using a user account. The server would use a JSON string and an MD5 hash to verify data sent from user devices. In the end, we were not able to implement the server, so all user inputted data will be stored locally on their device.

Upon opening the app, the user is greeted with two main buttons, edit data and add product. The add product button will pull up an edit screen which will have the user input the Universal Product Code (UPC), Price of the product, Product Brand, and Product Name. The user also has the ability to use their camera to scan the barcode. The barcode scanner uses a bitmap barcode decoder to pull the raw data in string format from an image taken with the camera, and input that into the UPC field. The data the user inputs into the UPC field becomes the primary key for that product within the local SQLite database.

The second button, edit data, pulls up a recycler view of all products currently stored in the database and allows the user to click any product to pull up its edit page. Once a product has been saved to the database, the user will not be able to change the UPC they originally entered.

There is an additional options menu with username and password entry fields and the ability for users to log in or sign up for an account. The username database is currently deactivated as the feature is not needed in the apps current, local database only state.

In future development, we would like to enable the online database feature and allow users to upload their local SQLite databases to the online database. Additionally, we would like to enable a web UPC lookup API (such as UPCitemdb) to auto populate the price, brand, and description fields for the user. We would also like to add geolocation services to attempt to auto populate a store/location field within the database. Finally, we would also look to add a general notes field for the users to add any notes about the product (such as, limited quantity).

The final goal of this future development, in addition to what has been done already, would be to create a local app that would allow users to track pricing of products at various stores in their area, then upload that data to a central server that all users could access to create a crowd-sourced price tracking solution that offers localized results for consumers. In addition, the manual upload feature of the app would allow users to choose when, if at all, to upload their own databases, potentially eliminating the need for customers to use wireless data as a result.

**Responsibility Matrix**

|  |  |  |
| --- | --- | --- |
|  | David Hatten | Grant Farnsworth |
| Create App UI | X | X |
| Create App with local data |  | X |
| Document Code | X | X |
| Final Report | X |  |

**Original Proposal**

Price Tracker

Our project is to create a price gathering app. It will store the price, date, store location, Universal Product Code, product name, and user account.

The app can read in the UPC from the barcode using the camera. Everything will be stored locally on a SQLite database until they are uploaded. Once the app is set to upload the data, everything is put into JSON format including a md5 checksum. The JSON file is posted to the server which calculated the md5 hash. If the md5 matches the data will be entered into the MySQL database. The server will send back the matching md5 hashes to the app. The app will then delete the matching records from the local database. If some of the md5 hashes are missing or different, an error will be displayed.

Android Libraries

* [Zxing](https://github.com/zxing/zxing): a library for reading bar codes.
* [Sugar](http://satyan.github.io/sugar/): a library for SQLite

Server Software

* MySQL
* PHP7
* Laravel
* Apache

To do

|  |  |  |
| --- | --- | --- |
| Create databases and PHP | Grant | October 30 |
| Create app UI | David | October 30 |
| Create app version with local data gathering completed | Grant & David | November 21 |
| Create code to interface app and server | Grant & David | December 3 |