**Girl Scout Cookie Tracker**

**(GSCT)**

Albert Li

Jack Ye Yuan

Long Pham

Luke Walker

Rudy Duarte

**Welcome!**

**Table of Contents**

**Cookie Mom:**

***Overview***……………………………………………………………………………………………………………………………………………………………………………………2

***Android Application***………………………………………………………………………………………………………………………………………………………………2

***Website***………………………………………………………………………………………………………………………………………………………………………………………4

**Girl Scout:**

***Overview***…………………………………………………………………………………………………………………………………………………………………………………… 8

***Android Application***……………………………………………………………………………………………………………………………………………………………… 8

**SELECT ‘Programmer’ FROM ‘Hello, World!’;**

***Overview***…………………………………………………………………………………………………………………………………………………………………………………. 10

***Website***……………………………………………………………………………………………………………………………………………………………………………………. 10

***Database***…………………………………………………………………………………………………………………………………………………………………………………… 10

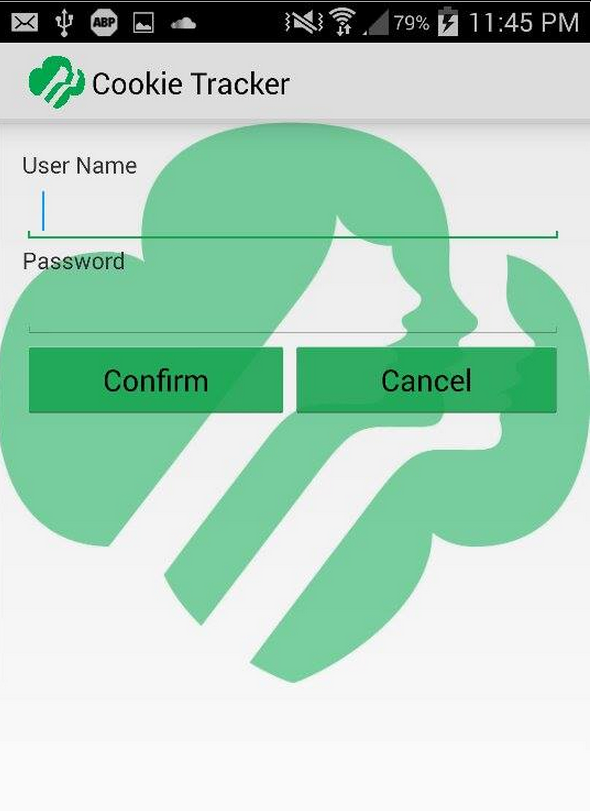
***Android Application***……………………………………………………………………………………………………………………………………………………………. 10

***Migration***……………………………………………………………………………………………………………………………………………………………………………….. 10

**Cookie Mom**

***Overview:*** The cookie tracker utilizes an Android app that tracks the girl scouts’ cookie order and sends it to the customized website (<http://nodejs-gscookiesales.rhcloud.com/index.html>). The website is simplified and only shows the cookie mom the number of orders each Girl Scout has made (Dashboard tab). Each tab is unique so that the Cookie mom can take advantage of time and get all her necessary orders ready to go. The Android application has a local database in which the Girl Scouts’ orders will be saved until they are connected to Wi-Fi and they can sync their information onto the external database. These orders are then simplified for the Cookie Mom on the website.

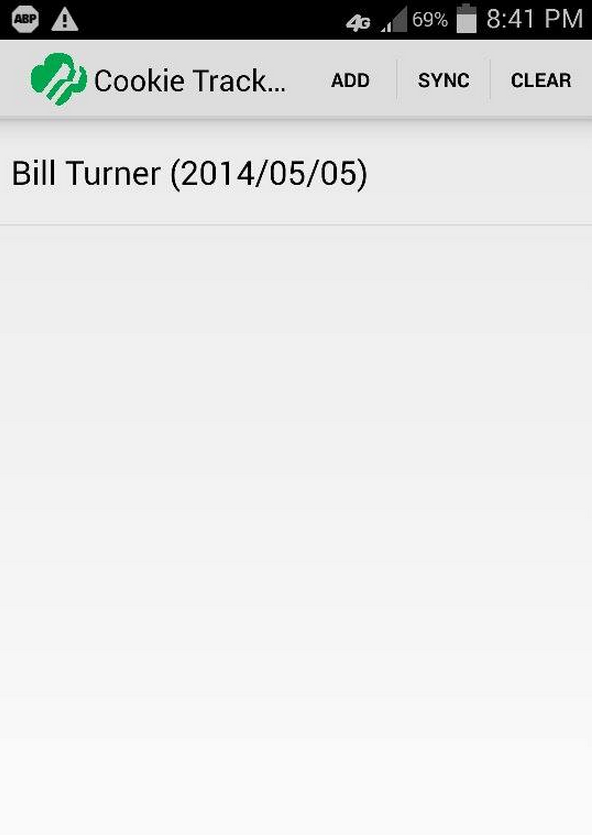
***Android Application:*** The Android Application uses an app in collaboration with a local database that stores the Girl Scouts’ cookie information directly on the phone when she is taking orders on the road. The Girl Scout will download the GSCT from Google Play Store by searching Girl Scout Cookie Tracker or GSCT. When the Girl Scout opens the program, she will be prompted to create a username and password so that the application can sync her orders on the external database (which is then shown on the website). The app is able write all the necessary information for each customer including: contact information, cookie types they are ordering, number of cookies, if the customer has paid for their items and whether or not they have picked up their order.



The initial screen that the Girl Scout is prompted. She will input her credentials every time she wants to sync her orders to the external database (Wi-Fi is needed to prompt that connection, she will receive an error if she is not connected).



This is the order form for Girl Scout to use when inputting orders.

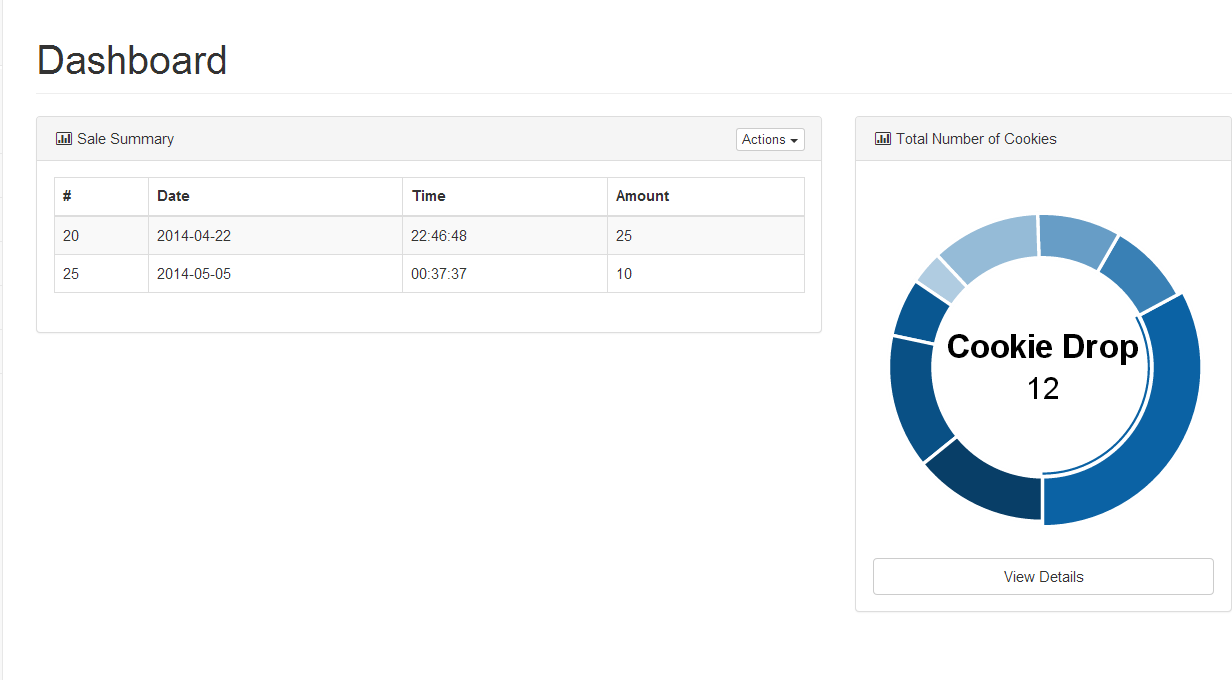


Once the order is placed it will show a box of the most recent orders. She will be able to tap on the order to make changes, sync it or clear the order once she has fully fulfilled the order (the customer has paid and picked up the order).

***Website:*** We understand the Cookie Mom’s time is value which is why we created a simplified website that shows the important information that the Cookie Mom to fulfill her tasks (<http://nodejs-gscookiesales.rhcloud.com/>). We have broken down the Cookie Mom’s tasks into 6 tasks: Dashboard, Scouts, Orders, New Order, New Scout, and New Cookie.

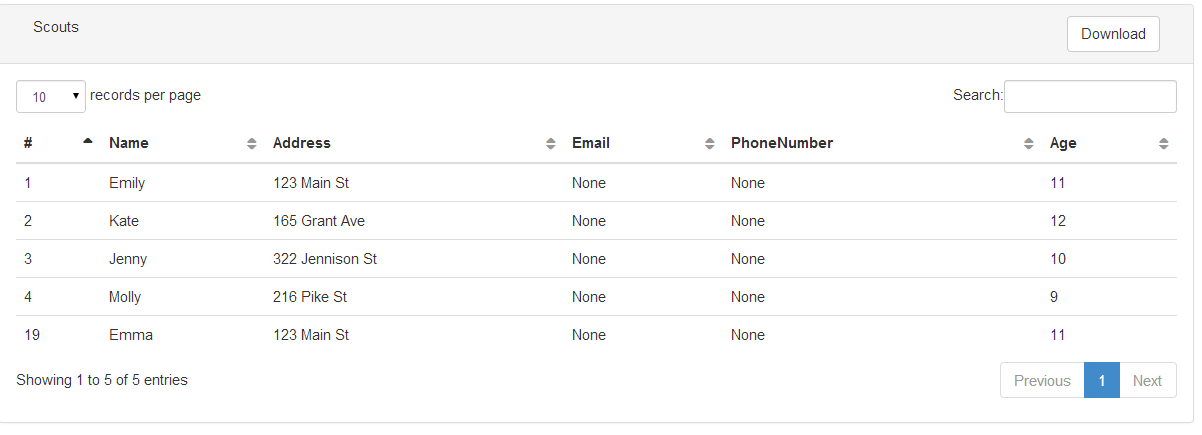
*Dashboard Tab:*





The Dashboard tab quickly shows the Cookie Mom the amount of orders that were recently placed by the Girl Scouts. The graph on the right shows the total amount of cookie that have been ordered and it breaks it down by each cookie time.

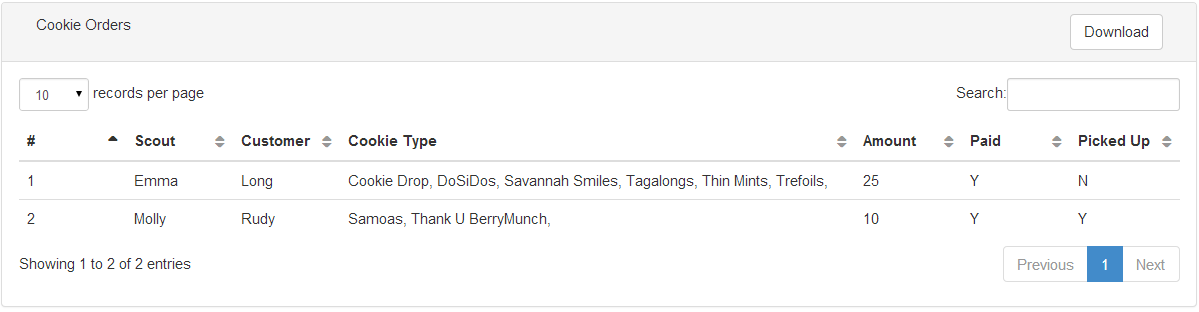
*Girl Scout Tab:*

****

Under the Scouts tab, the Cookie Mom is able to view the current girl scouts, as well as all the important information corresponding to each Girl Scout (i.e., name, address, age, etc.). On the right top right, the Cookie Mom is also able to search on any terms that she is looking for (as long as it corresponds to a Girl Scout).

*Orders Tab:*

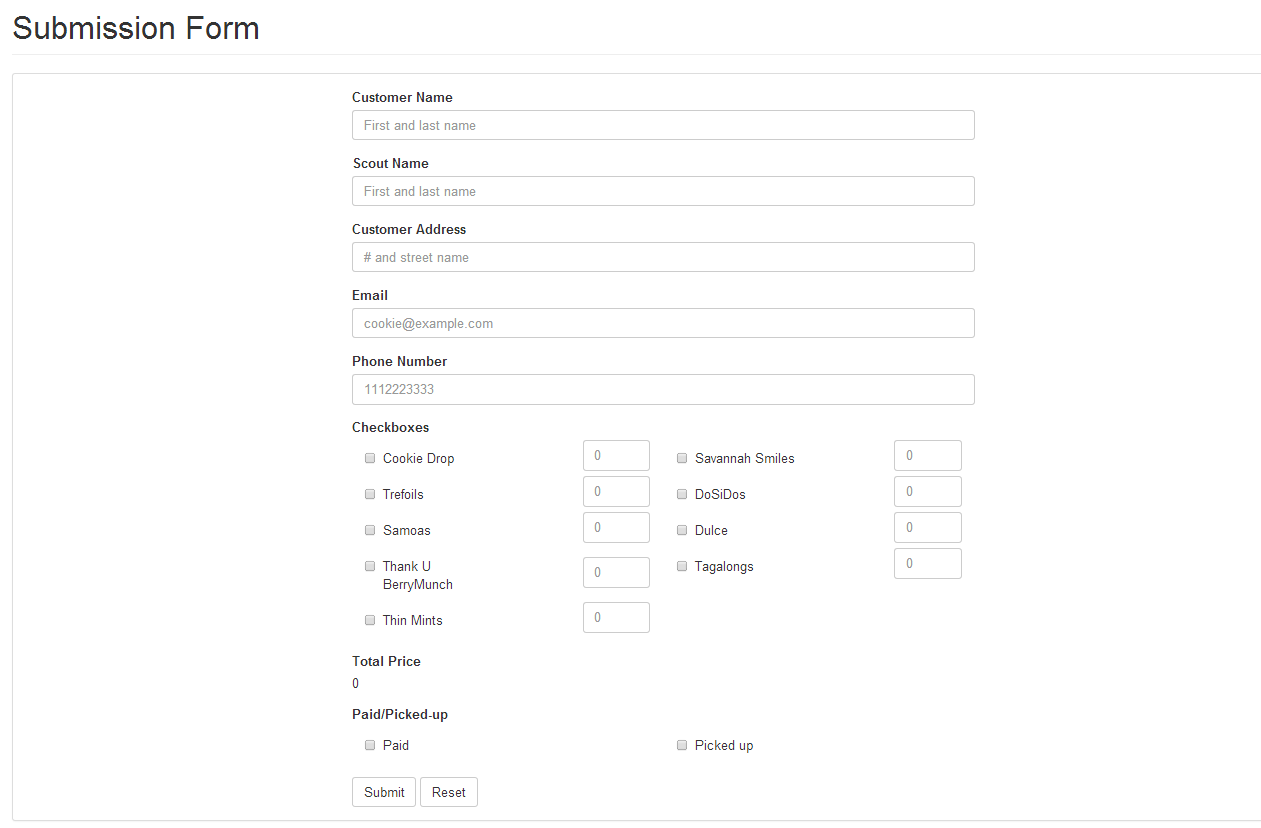
**



In the Orders tab, the Cookie Mom is able to view detailed information of the orders the Girl Scout has been placing. The Cookie Mom is also able to click in the order and edit the order or delete it. The Cookie Mom is also able to download all the orders into a CSV file.

*New Order Tab:*

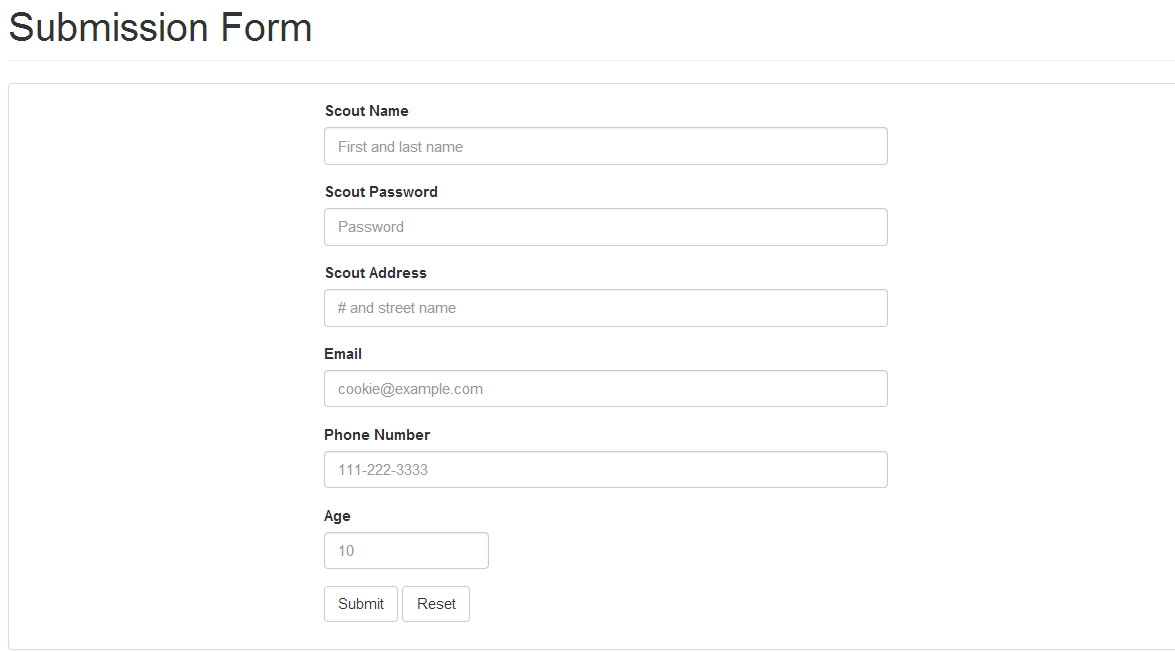
**

**

In New Order tab, the Cookie Mom is able to place new orders that she is given or place orders for Scouts that may have forgotten to submit orders. The submission form is similar to that of the Android app.

*New Scout Tab:*

**

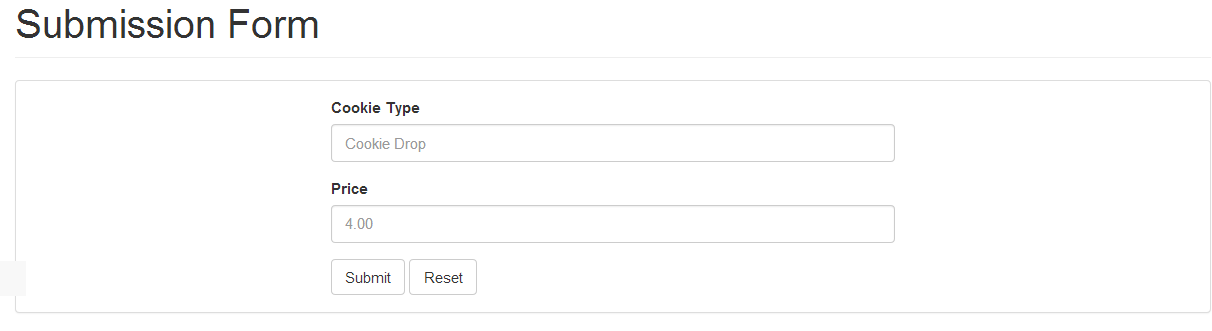
**

In the New Scout tab, the Cookie Mom is able to add new Girl Scouts into the troop if needed.

*New Cookie Tab:*

**

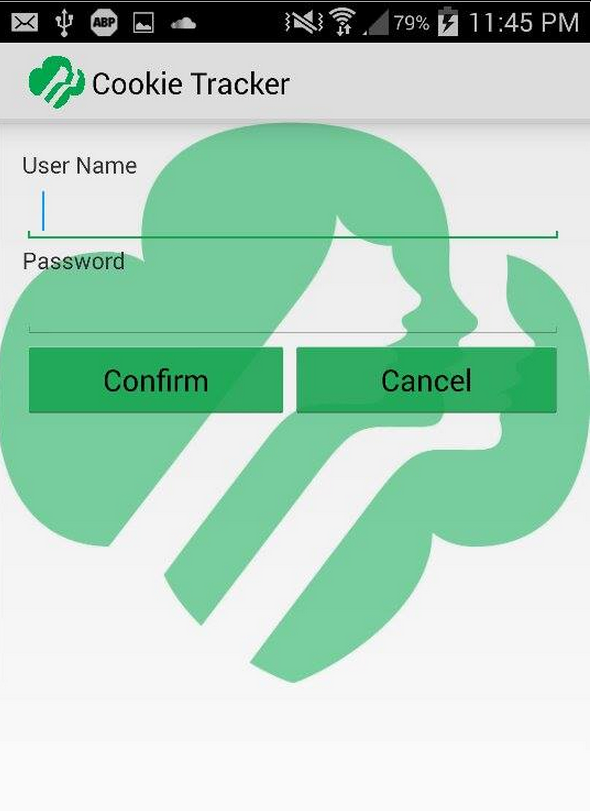
If the cookie types were to ever change, the Cookie Mom is able to submit the new cookie type into the website so that Girl Scouts are able to place orders for that cookie type.



**Girl Scout**

***Overview:*** The Girl Scout will be able to track her cookie order/sales through a customized app for Android. The Girl Scout will be able to download the app through Google Play Store onto their Android device.

*Login Screen:*



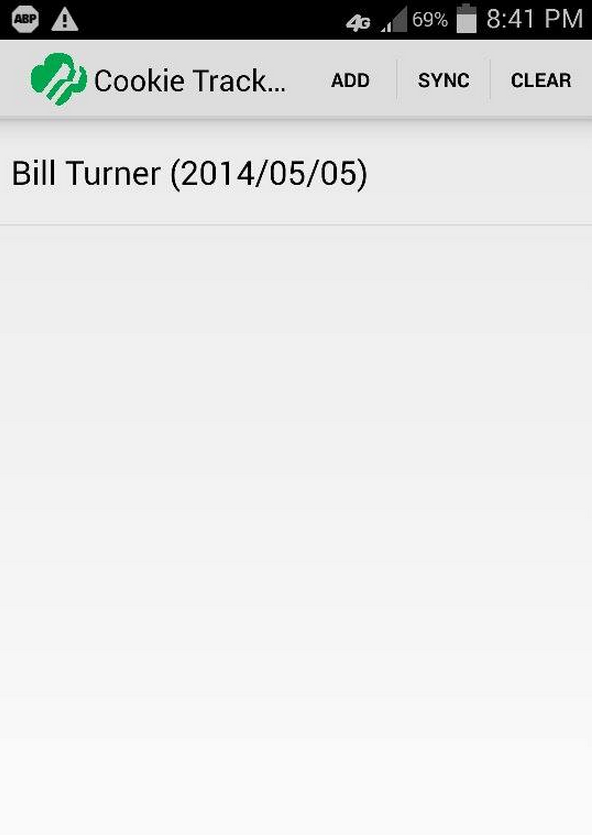
The initial screen that the Girl Scout is prompted. She will input her credentials every time she wants to sync her orders to the external database (Wi-Fi is needed to prompt that connection, she will receive an error if she is not connected).

*Order Form Screen:*



This is the order form for Girl Scout to use when inputting orders. She will be able to put in detailed information for each customer as well as keeping track if their customers have paid for their items and if they have picked up their items.

*Total Orders Screen:*

**

Once the order is placed it will show a box of the most recent orders. She will be able to tap on the order to make changes, sync it or clear the order once she has fully fulfilled the order (the customer has paid and picked up the order).

**Programmer**

***Overview:*** The GSCT uses Openshift’s free webhosting to store the database and host the website (<https://www.openshift.com/application-gallery>). To access the project, the username is: duarterudy125@yahoo.com and the password is: TurnerRocks1. Openshift uses what are called cartridges which are pre-installed programs to manage your project. We used the phpMyAdmin, MySQL and Nodejs cartridges. The android application was modified from Deitel book code (movie list). The app is written in Java code for all the app function and it utilizes SQLite3 on the phone. SQLite queries are written in JDBC; the app saves the content on the local database until it is synced to the external database, this uses a JSON parser to aggregate the data. The website was modified from Start-Up bootstrap web template. The website is built on HTML5 and JavaScript. We used a MySQL database for the external database.

***Website:***In order to view the source code for the website, you will need to fork (copy) the project on GitHub (version controller). The project can be found at <https://github.com/longpham91/Cookie-Sale-Project>. The website is hosted on open shift but it can be edited using an IDE that supports web development. The backend of the website is handled by Nodejs.

***Database:***We used a MySQL database to handle all the information from the website as well as the android information that is synced. phpMyAdmin (MySQL interface/database manager) was used in order to create the tables and manage the database. The credentials to use phpMyAdmin are on the application gallery (under cartridges).

***Android Application:***The app uses modified book code from Deitel that manages the app’s interface as well as the SQLite management. A JSON parser was used for the communication of the local database to the server. Source code for the app can be found on Github.

***Migration:*** In order to migrate the application (website, database) to another hosting site, the service must be able to handle MySQL database and it must support Nodejs/JS for the website management. The service must also allow remote connection to the database (not port forwarding). Typically, these services are paid services in order to allow remote connection to the server.