

FINAL PROJECT

DONATION TRACKER

About the Customer

CS+Social Good is an organization that focuses on the intersection of computer science and social impact. Striving to build a community of passionate computer scientists and engineers who are applying their technical knowledge towards solving the world's biggest problems, CS+Social Good is changing the way students understand the value behind a computer science education and its application worldwide.

Our mission: To empower students to leverage technology for social good by inspiring action, facilitating collaboration, and forging pathways towards change.

This semester, CS2340 will build an Android application for CS+Social Good to help non-profit organizations track donations of goods. The overall goal of this project is to create an internal resource planning app for non-profit donation focused organizations (e.g. Goodwill, Lost-n-found Youth) to efficiently track and categorize donations. Being able to effectively manage donations allows these services to maximize their ability to distribute donated goods to people in need, as well as track donation and distribution metrics to further optimize their services. The app would also connect donation centers allowing them to redistribute certain goods (if saturated at one location and/or needed at another) and to better help people in need find an item they are looking for.

The top five projects selected by the instructor and TA's will be forwarded for further development. Teams may opt to continue their projects to full functionality in conjunction with the CS+Social Good organization.

Final Project General Information

Projects are usually somewhat open-ended. If we do not specify how you have to do something, then you are free to use your own imagination. I have put some information in the description and grading criteria, but you may design and organize your application as you wish, as long as it meets the basic requirements.

A few extra credit opportunities are suggested in M12. You may propose other ideas by posting them to Piazza in the appropriate topic. Throughout the semester, you will be designing the full application as described here. You will be required to implement a subset of the requirements as detailed in the coming feature slices. Any functionality described here that is not in one of the milestone descriptions can be completed for extra credit.

Donation Tracking Project

This application concept was created by CS+Social Good, an organization that focuses on the intersection of computer science and social impact. You may actually name your application whatever you want, I will be calling it BuzzTracker.

We want to help answer questions like:

- What is the donation rate at a particular location?
- What is the distribution rate at a particular location?
- What is the inventory turnover rate?
- What were donations and distributions for a particular day?
- What items are available at a particular location?

That is where this app comes in.

User Categories

Admin : An admin can add and remove users and locations as well as unlock/lock accounts.

User : A user is anyone who wants to use the system to view inventory or locations. They cannot actually alter any information.

Location Employee : A location employee is someone who works at one of the dropoff sites and can update the current inventories for his or her location.

Manager: Someone who can view/add/modify information at all locations.

All user types contain the same basic information:

- login name (in recognition of current popular trends, this can be the email address)
- password
- account state (locked or unlocked)
- contact info (email address)

In addition to this information, a Location Employee should be linked to their location (Employees should only be able to edit information for their specific location).

Location Data Handling

You will be provided with a location list (a .csv file) that gives you information about the known locations for the organization. This file will be located in the Files section of Canvas (locations.csv).

The information for each location includes the following:

- Unique key (These numbers are meaningless outside of this app and should not be visible to the users. Depending on implementation, they may or may not be useful to your team.)
- Location Name
- Location (latitude, longitude, and street address)
- Type: Dropoff-only, Store (drop-off and sales), Warehouse (inventory storage only)
- Phone Number (in (000) 000-000 format)
- Website link

Donation Drop-Off

When a donation is made at the location, the following information should be entered:

- Time stamp of donation
- Location of donation
- Short Description of item (for display purposes)
- Full Description of item
- Value (in dollars)
- Category (Clothing, Hat, Kitchen, Electronics, Household, Other)
- Comments (optional)
- Picture (optional)

The provided categories are the only ones your application must support. The ability of employees / locations to create custom categories, or to have even more categories (or even a hierarchical structure) is extra credit.

Security

A person must login to the application in order to access its features. At login, the user type is determined, and the appropriate rights granted.

If a login attempt is unsuccessful, the person is allowed a total of 3 failures before they are locked out. An administrator must unlock their account once it is locked.

Logs will be maintained to track usage of the system and to log critical events to persistent storage. Critical events include: Altering/Deleting Information, Successful/Unsuccessful login, logout.

A security entry in the log file consists of a timestamp (when the event happened), what type of event, and the user id of the active user that caused the event.

Searches

A user may conduct a search for information about locations. The following searches should be supported:

- Item - show all locations with a particular item
- Category - show all locations that have items in the desired category
- Location - show all Locations close to a certain given location

Map Displays

The location of Locations should be shown on google maps. Clicking on a pin should show some details of the Location at that pin.

Graphs

The application should provide graphs of data analysis for the Locations and donations.

The following graphs should be provided:

- a. Total items by category
- b. Value of inventory by month
- c. Income (sell prices – value) per month.
- d. Donations per month per location

Management Information

The following queries should be supported:

- a. Donations per location per day for a given month
- b. Sales per location per day for a given month
- c. Average time from a donation to a sale for each category of item

M12 Extra Credit ideas

- You may implement an extra option at the login/register stage of your application. The third option is to proceed without logging in. This would allow a user to search the locations but not report any other details. Essentially, this creates a fourth user type (Guest) who has less privileges than any of the other three in return for being completely anonymous.
- Email-based password recovery (in lieu of or in addition to Admin unlock after 3 unsuccessful attempts)
- Use Google Maps to show directions to a selected location
- Push notifications (for item donations)
- Alternative platform implementations: for example, Native iOS or a web app

Warm-Up Milestones (Not related to features)

M1 – Team Setup

1. Create a team page on Canvas
2. Complete a team charter
3. Setup Zenhub and enter some initial data

M2 – Version Control and Automated Builds

1. Create a team github repository
2. Edit the Person files and conduct merges as required
3. Create and manage Branches
4. Write a gradle build file

M3 – Individual Android Program

1. Install and setup Android Studio
2. Import the existing project
3. Make required code changes
4. Test completed application

Functional Milestones

M4 - Project Setup/Login/Logout

1. Create your team's Android Project and Import into Git
2. Create an opening or welcome screen which has options for Login and Registration. You may have some other method of differentiating login and registration, if you do not want an opening screen.
3. If someone selects Login, then you should display a way to enter the user name and password.
4. Selecting Registration is a no op for this milestone and can have whatever effect the team wants.

5. If someone requests login, you should check for the correct user name and password. For this first milestone, you may have a hard-coded user with name "user" and password "pass" to check against.
6. If the login matches user name / password then go to your application. Otherwise, notify of the bad login attempt.
7. Canceling the login (pressing Cancel or going back a screen for example) will close out the login attempt. No information is recorded, and the application does not start up. Note: here you are not explicitly required to have a cancel button, just a way to back out of the login attempt.
8. Once in the application, there should be a way to logout. After logging out, the application should return to the welcome / opening page.

NOTE: Security logging, password encryption, or account lockout is not required for implementation and would be extra credit for later in M12.

M5 - Registration

Add the registration screen to your app. A new user should be able to register by entering their information (like name, email, and password). If accepted, the new user should be added to the system. If cancelled, the user should not be added.

During registration, the new user's account type (User or Location Employee) should be specified. In a real application, we would have controls for determining this, but for this class project, we will have registration designate accounts as basic Users. You may implement creation of other account types for extra credit.

When registration is working, you may remove the hard-coded user/pass from the app.

NOTE THAT PERSISTENCE IS NOT REQUIRED AT THIS POINT. IT IS OK TO HAVE TO RE-ENTER USERS FROM SCRATCH EACH TIME THE APPLICATION IS STARTED.

M6 - Read Location Information

Note: Databases are extra credit, therefore, you can assume for testing of this app all data will fit into the device memory. Since persistence is not yet required, entering users and other data can be done on each run of the application.

In this milestone, we implement the ability to read in location data. You should have some kind of input screen for selecting the option to read in the data. Optionally, you can automatically read in the location data on program startup. The data after loading should be stored somewhere in the model. You should provide a list view of the data (this is a conceptual list, not a requirement to use any particular widget).

The location data is in a text .csv (comma separated) file in the Files section of Canvas.

Opening the file in a spreadsheet will allow you to see the order of the data elements. The following fields should be handled by your app (the other fields can be ignored)

- Location Name
- Location Type
- Longitude
- Latitude
- Address
- Phone Number

What fields you show in the main list view is up to you, but should at least include the Location's name. Selecting an item in the main list should take you to a details page with all the information about that specific location.

M7 – Add a donation item

In this milestone, we implement the ability for location employees to add a donation to the location. Your application should have some way to enter all the information about an item for a given location (see the Donation Drop-Off section). Location employees should be able to view a list of all items at their location and selecting any item should take you to the relevant details. Note that photos and comments are optional extra credit items.

M8 – Search Items and Persistence

We are going to support two different search filters through items. A user should be able to search through all locations, or a selected location. How a location is selected is left to the team to design (you may just use the list from the previous milestone if desired).

Search 1: Search by category. The user selects a specific category, and your application should display all the donation items available that have that category.

Search 2: Search by item name. The user enters a specific item name and your application should display all items with that name. Note: Fuzzy searches and partial matches are extra credit.

We are also going to now persist the data on the phone. In class we looked at ways to save the data onto the phone. You can review my sample code if desired. After this milestone, it should no longer be necessary for you to re-enter data each time you run the application.

NOTE: If you are doing the database extra credit, you are already persisting the information. You do not have to have both a database AND read/write to the phone.

M9 - Google Map Display

We will now add the ability to show on a Google map the locations of shelters. You should allow the user at a minimum to select which restrictions to show. For example, show only shelters for women or only shelters for families.

Pins should appear on the map for each shelter that meets the requirements. Clicking on a pin should give some basic information about the shelter. Information must include the shelter's name and their phone number, but feel free to include more.