Morning Star ScanTrack v0.5

Morning star scantrack is an android based application to scan barcodes into batches and assign gps coordinates to a batch or groups of batches as a backup means of locating a batch or container. The application is written in java, was developed using android studio, and the google mobile vision API. The activities for the android application are each explained in detail below as well as current features, a roadmap for future development, and any known bugs in the application.

Java Activity Files:

AddToBatchActivity.java

-This activity is designed to allow a user to add to an existing batch from the batch management screen. It opens the scanner with the batch to be added to, its data is populated to section, row, and batch ID. This section is unfinished.

BatchManagmentActivity.java

-This activityImplements the ability for the application to grab gps coordinates.

-Batch management also displays a checklist of the current batches showing batchID and the number of containers in the batch.

-Checkboxes are used to assign gps coordinates to one or more batches.

-They are also used to select a single batch for editing, sending that batch data and the user to the EditBatchActivity screen.

-New batch takes the user back to the scanning screen using the next available batchID.

-This activity is unfinished.

EditBatchActivity.java

-This activity opens a detailed list of containers in the batch to be edited, and allows the user to remove individual or multiple containers as well as add new containers.

This activity is unfinished.

MainActivity.java

-This activity runs the main menu. Implements button functionality on the main screen, sends users to appropriate activity, and implements an exit button to fully close the application.

PictureBarcodeActivity.java

-This activity was developed to be a method for scanning more than one barcode from a photo.

-It only works when barcodes are in close proximity, at the same angle, and in good light.

-It also requires loading the built in camera app on the android device as such it added extra steps to the workflow.

-The button to access it was removed from the current version for this reason, though the code and xml were left here in case further development was requested.

ScannedBarcodeActivity.java

-This is the primary scanning activity. It creates a real time scanner using the built in back camera of an android device. Barcodes are detected in realtime and scanned very quickly. Scanned barcodes are added to the current batch and displayed on the left hand side of the screen below the batchID number. Upon starting a new batch via the listed button this list is cleared and the ibatchID is increased. Upon pressing clear batch any scanned codes are cleared front the current batch and the displayed codes are reset. The code also detects any attempt to scan a duplicate code to a batch, warns the user of the duplicate scan, and does not add the code to the batch.

SearchActivtiy.java

-This activity is designed to display batches or containers in the database based on a variety of search parameters. This section is unfinished.

SyncActivity.java

-This activity is meant to contain the code to engage syncing with the main database, it is unfinished at present.

Additional Java Files:

Batch.java

-A class to contain batch data transferred between activities via an arraylist of objects.

Conatiner.java

-A class to contain container data transferred between activities via an arraylist of objects.

CustomAdapter.java

-A custom adapter used to display lists for data in batch management and other screens.

DataModel.java

-A data type for checklist implementation.

DataWrapper.java

-A class to enable sending of custom object array lists between android activities.

Current Version v0.5 Features:

v0.1

-Scans code-39 barcodes as well as others. Testing was done with code-39 barcodes.

-Displays the current batchID and all codes scanned into that batch.

-Has a batch system that allows barcodes to be scanned into groups of batches. A batch being a group of containers.

v0.2

-A functioning local database that is updated with batches and containers as they are scanned.

v0.3

-Scanning done sends the user to the batch management screen.
-Batchmanagement Menu Skeleton Built
-Quit Button closes current activity return user to last activity

v0.4

-Clear batch button clear all scanned codes in the current batch
-New batch inserts the current batch, increments the batchID, and clears the displayed code list.
-A basic GPS implementation

v0.5

-Exit button built for main screen
-Scanning Animation Added
-A few Bug Fixes
-Data Transfer between activities built and tested

Planned Development Roadmap:

v0.6

-Build a batch detail screen that shows the number of containers in each batch and the batchID.
-Enable update batches in the database via checklist by adding GPS data to batches.

v0.7

-Build a container detail screen accessed from the batch detail screen by clicking on a container in the batch list.

v0.8

-Build a main search functionality that displays results by one or more search parameters

v0.9

-Implement a google map based gps display of the location of a container based on GPS coordinates.
-Implement the ability to sync with a server for database updates.

v1.0

-Organize and test all existing Functionality - Minimum functionality state

Known Bugs:

-Duplicate barcodes are prevented from being scanned to the same batch and a prompt warns the user that a duplicate code scan was attempted. Due to the rapidity with which the application scans codes if one holds the scanner over a duplicate code for too long a time it can

be scanned many times, this can cause the message warning the user to hang and not disappear for a long time or until the application is reloaded.

-Back Vs Quit, The quit button in each activity should be used to end current activities vs the back button on the home bar of android. Using back can cause the exit button on the main screen to malfunction as activities are still open in memory. This causes exiting the application to require the exit button be pressed several times before the app will properly close.

Contact Info:
If you have any questions regarding this project or would like it developed into a finished product that meets the specifications please contact the developers listed below.

Lead Programmer: Ada Pluguez - 209-608-6998 - adapluguez@gmail.com