Requirements and Analysis Document (RAD) for Barcode Scanner Project - Group 30

Christian Svensson Olle Andreasson Olof Karlsson
Rasmus Letterkrantz

August 26, 2013

Contents

1 Introduction

The Barcode Scanner is an android application that is meant to be used by small-time shop owners, potentially replacing conventional barcode scanners. Since most people own a smartphone, we believe this could also be useful for anyone wanting to keep track of items that carry conventional barcodes, and store them in a database.

1.1 Purpose of Application

The project aims towards creating an application that can scan barcode and store product information related to each barcode in a database, and provide methods for handling and accessing these products, for Android devices (with a camera) running Android 4.0+.

1.2 General Characteristics of Application

The application will be an Android application with a graphical user interface. It will take use of the camera to take snapshots. Once the application have an image containing a barcode, it will be analyzed to generate an identification number. This will be compared to a database to result in a product as output.[3][1]

1.3 Scope of Application

The application will not be guaranteed to work on every android platform, it will be tested on version 4.1.1 and this will be the only version that is guaranteed to work. The application will not necessarily generate standard barcode IDs, as barcode scanning is a very complex problem that we cannot reasonably solve completely in the given time.

1.4 Objectives and Success Criteria of the Project

1. The application will be able to take a picture using the back-facing camera on your phone.
2. The application will then scan that image for a possible barcode.
3. The application will be able to generate a unique id from that barcode.
4. The application will be able when given an id, compare it to the database content, and if the id exists, return a product. If no product matches the given barcode, the user shall then be given the ability to create a product to match that barcode ID.

1.5 Definitions, Acronyms and Abbreviations

- GUI, Graphical User Interface.
- Android, open source operating system for mobile units (smart-phones, tablets, etc).
- Java, platform independent programming language.
- XML, Extensible Markup Language is used for structuring the elements in the GUI.
- SQLite, the database used to store the barcodes.
- Locator, class used to locate the barcode on a given image.
- Generator, class used for analyzing the pattern extracted by the Locator and generating the actual key.
- MVC, a way of partition an application with a GUI into distinct parts avoiding a mixture of GUI-code, application code and data spread all over.
- ADT, the Java Runtime Environment. Additional software needed to run an Java application.
- .apk, application package file, a file format use to distribute and install application software and middleware onto Android.

2 Requirements

2.1 Functional Requirements

1. Take a picture.
2. Scan the picture for any barcode (parse).
3. Match (read) or add (write) any result to a database.
4. End the application (turn of camera and exit the process).

2.2 Non-Functional Requirements

2.2.1 Usability

Aim for simplicity! The application should be simple to use, the GUI should be clean and follow the standard Android design guidelines[2] and the flow should be straightforward for any user. The scanner should pass tests from users who are not, by any means, an advanced android user.

2.2.2 Reliability

Each combination of numbers that we generate from each barcode should match the general combination that the barcode represents (as long as the barcode is in the correct EAN format).

2.2.3 Supportability

The application Gui should be clean and scalable with the screen size of the given device. There should be automated test verifying most use cases. Code related to the

GUI could be tested manually. GUI test should be recorded and included in the final documentation.

2.2.4 Performance

The scanning should be quite fast and the application should not weigh down the time for the camera to take and present a picture.

2.2.5 Implementation

The application should be using the standard Android-specific Java environment.

2.2.6 Packaging and installation

Installation should be done using a standard .apk file containing the application, that the user can then install on their Android device.

2.2.7 Legal

Android is a open platform, freely available to developers everywhere. The barcode standard our algorithm will detect is EAN-13, which is an international standard freely available.

3 | Application models

This is the UML diagram for the project:

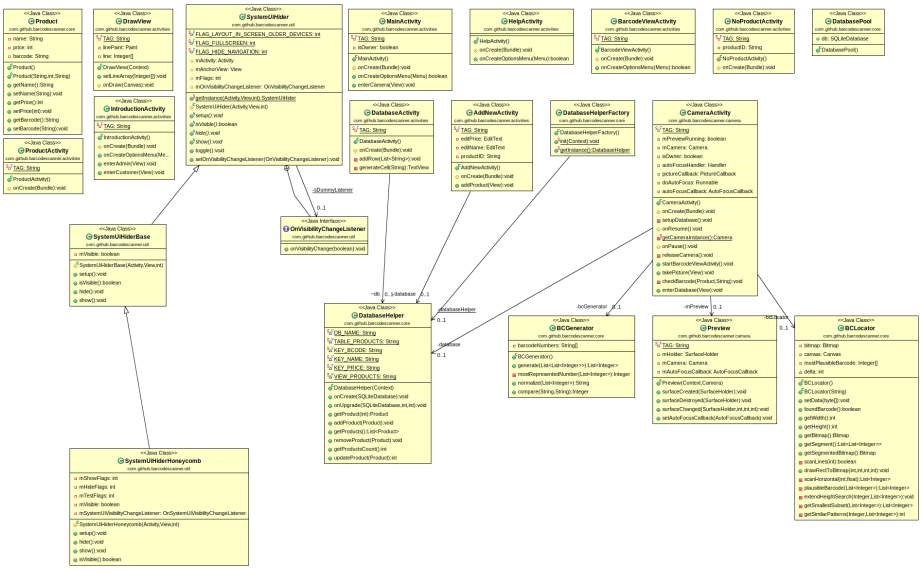

Figure 3.1: UML

3.0.8 Use cases

Here follows the use cases for the barcode scanner project:

1. Scan an image containing a new barcode.

   Summary: This is how the user actually scans a barcode. Behaves differently depending on whether or not a barcode is found.
   Priority: High
   Extends: -
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   A barcode is found in the image.

	Actor	System
1.1	Clicks the "Scan" button	
1.2		Displays different views depending on whether or not the user is a shop owner or a customer.

   Alternate flow
   Flow 2
   No barcode is found in the image.

	Actor	System
2.1		Nothing happens.

   Exceptional flow
   There is no exceptional flow.

2. Save a scanned barcode into the database as a new product.

   Summary: This is how the user saves information about a barcode to the database.
   Priority: High
   Extends: Use cases where a barcode is found.
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   The user enters information and then clicks the "Save" button

	Actor	System
1.1	Enters information in the information fields.	
1.2		The information is temporarily saved in the fields.
1.3	Clicks the "Save" button	
1.4		Moves back to the scan view, in order for the user to scan more barcodes.

   Alternate flow
   Flow 2
   The user decides she doesn't want to save the barcode, so she backs out of the view.

	Actor	System
2.1	Clicks the back button on the android device	
2.2		The system moves back into the scan view.

   Exceptional flow
   There is no exceptional flow.

3. Customer finds a barcode but no product matching the barcode is in the database.

   Summary: This is when a customer scans a barcode but there is no matching entry in the database. She is shown a view that tells her that there is no matching product.
   Priority: High
   Extends: Use case where the user scans a barcode
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   The user has scanned a barcode.

	Actor	System
1.1	User scans a barcode	
1.2		The user is shown a view that informs her that there is no product matching the barcode in the database.
1.3	User backs out of the view	
1.4		The system moves back to the scan view.

   Exceptional flow
   There is no exceptional flow.

4. Remove a product from the database.

   Summary: This is how the user removes a product from the database.
   Priority: High
   Extends: Use case where the database is shown.
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   The user long-presses the given item she wants to remove and then clicks the "Delete" button in the top-right of the application..

	Actor	System
1.1	Clicks the "Delete" button	
1.2		Deletes the product from the database, and updates the view to shown that it is no longer there.

   Exceptional flow
   There is no exceptional flow.

5. Find a scanned barcode in the database.

   Summary: This is how the user scans a barcode that already exists in the database. Behaves differently whether or not a barcode is found in the image.
   Priority: High
   Extends: Use Case 1
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   A barcode is found in the image.

	Actor	System
1.1	Clicks the "Scan" button	
1.2		Displays a view where the user can view the information that the database stores about the matching barcode.

   Alternate flow
   Flow 2
   No barcode is found in the image.

	Actor	System
2.1		Nothing happens.

   Exceptional flow
   There is no exceptional flow.

6. The user edits a product in the database view.

   Summary: This is how the user edits an already existing product in the database.
   Priority: High
   Extends: -
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   The user clicks on an item in the database and is taken to a product view for that item. Here the user clicks the "Edit" button in the actionbar at the top of the screen.

	Actor	System
1.1	Clicks the "Edit" button	
1.2		Takes the user to a separate edit view where the user can edit whatever aspect of the product she wants.

   Exceptional flow
   There is no exceptional flow.

7. View the database.

   Summary: This is how the user navigates to the view that shows the database.
   Priority: High
   Extends: -
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   The database view is shown

	Actor	System
1.1	Clicks the "Database" button	
1.2		Displays a view that shows each stored product in the database.

   Exceptional flow
   There is no exceptional flow.

8. View the help view.

   Summary: This is how the user navigates to the view that explains how to use the application.
   Priority: High
   Extends: -
   Includes: -
   Participators: Actual user
   Normal flow of events
   Flow 1
   The help view is shown.

	Actor	System
1.1	Clicks the "Database" button	
1.2		Displays a view that shows the user how to use the application.

   Exceptional flow
   There is no exceptional flow.

3.0.9 Analysis Model

There will be unique id's for every product in database.

3.0.10 User interface

Application will use a fixed (non skinable, non themeable) GUI following standard conventions. The GUI must take into account different screen sizes, possible very small (minimum size: 320 x 480 (HVGA) at 163 ppi). See APPENDIX for screens and navigational path's.

Bibliography

[1] Barcode scanner- how to. http://www.ski.org/Rehab/Coughlan_lab/General/Publications/barcodes-TekinCoughlan.pdf, Mars 2009.

[2] Android design guidelines. http://developer.android.com/design/index.html, June 2013.

[3] Barcode scanner. http://en.wikipedia.org/wiki/Barcode_Scanner_(application), April 2013.