**Digistrux**

Final Project Design — CS 4962-001

Rob Johansen (u0531837)

Summary

Many packaged products, such as electronics, furniture, and other household items, include some form of paper documentation that provides assembly instructions. Such documentation is often folded or crumpled, and typically contains the same set of instructions in a variety of different languages. Digistrux is a mobile application that aims to eliminate the need for paper instructions in packaged products.

The primary use case for Digistrux is as follows:

1. Manufacturers subscribe to the Digistrux service and provide:
   * Product documentation in digital form.
   * The barcodes and/or QR codes of their products.
2. Using the Digistrux mobile application, consumers scan the bar codes and/or QR codes of the manufacturer's products.
3. The corresponding digital documentation is downloaded directly to the consumer's device.

Digistrux provides a number of benefits over traditional paper documentation:

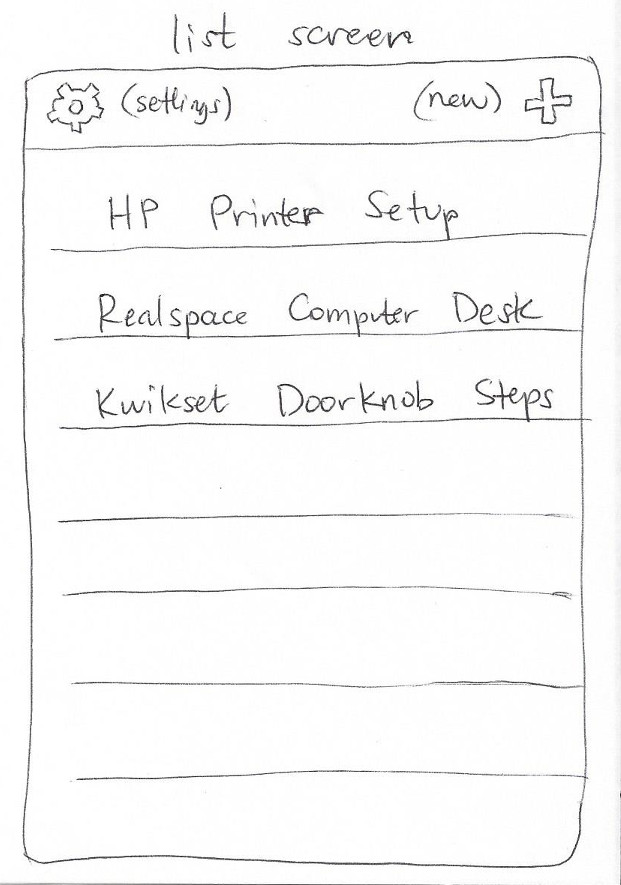
* *Reduced documentation costs* — Manufacturers no longer pay to print paper documentation in multiple languages; a wasteful practice considering that consumers only read a small portion of such documentation. Instead, manufacturers pay of fraction of that cost for a Digistrux subscription.
* *Instant revision* — If paper documentation is shipped with errors of any kind, or needs to be changed for any reason, it is virtually impossible to do so. The Digistrux platform allows manufacturers to correct mistakes or update product documentation and make such revisions available to consumers immediately.
* *Consumer convenience* — Paper documentation is often lost or discarded prematurely. Moreover, diagrams and font sizes in such documentation are typically small and difficult to read. Digistrux allows consumers to store documentation directly on their mobile devices, and facilitates zooming text and images for better clarity and detail. And, of course, Digistrux downloads documentation in the appropriate language, based on the user's locale.

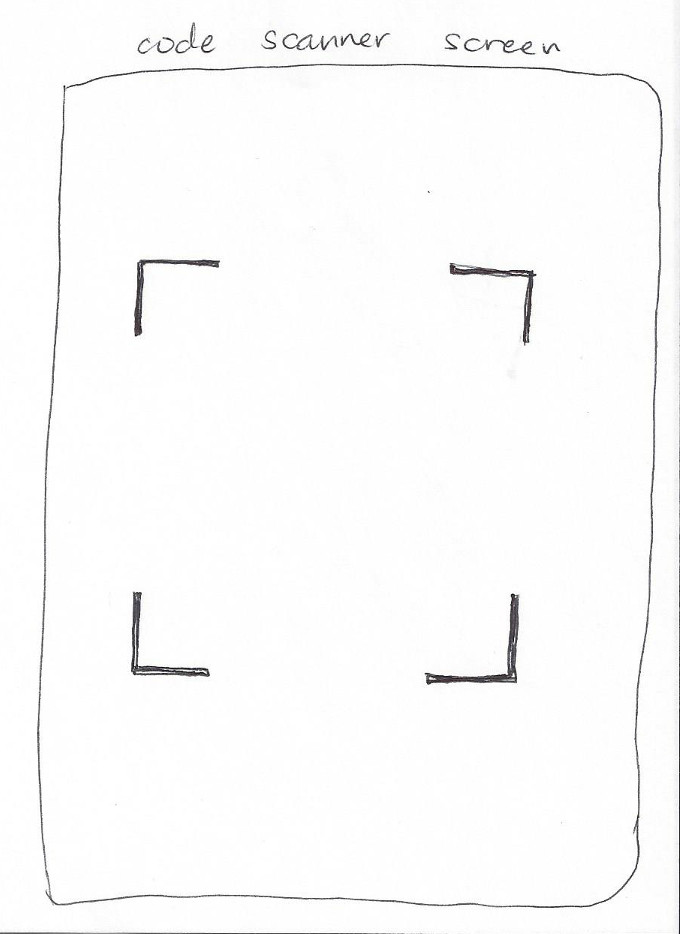
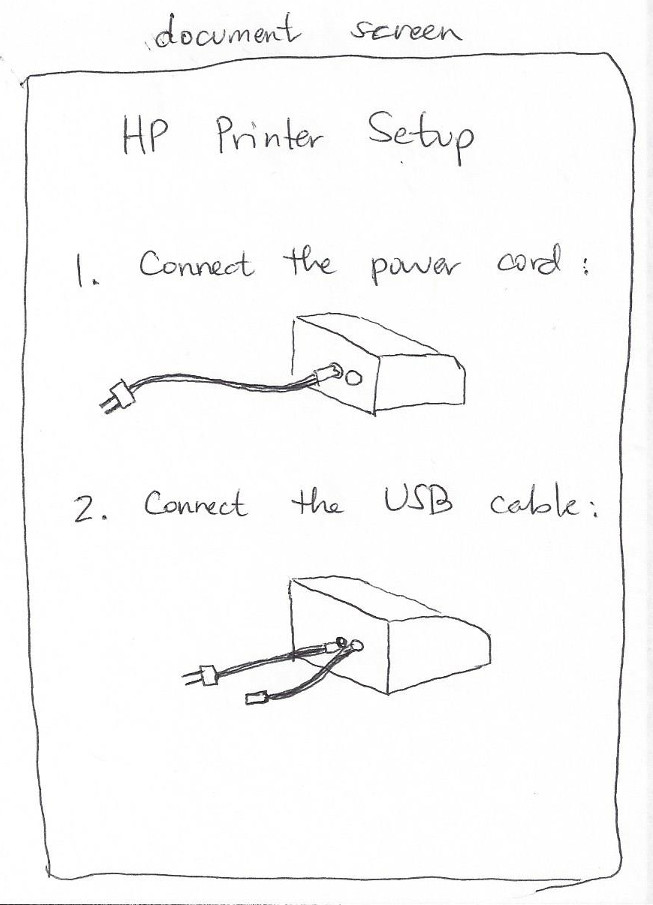
Screens

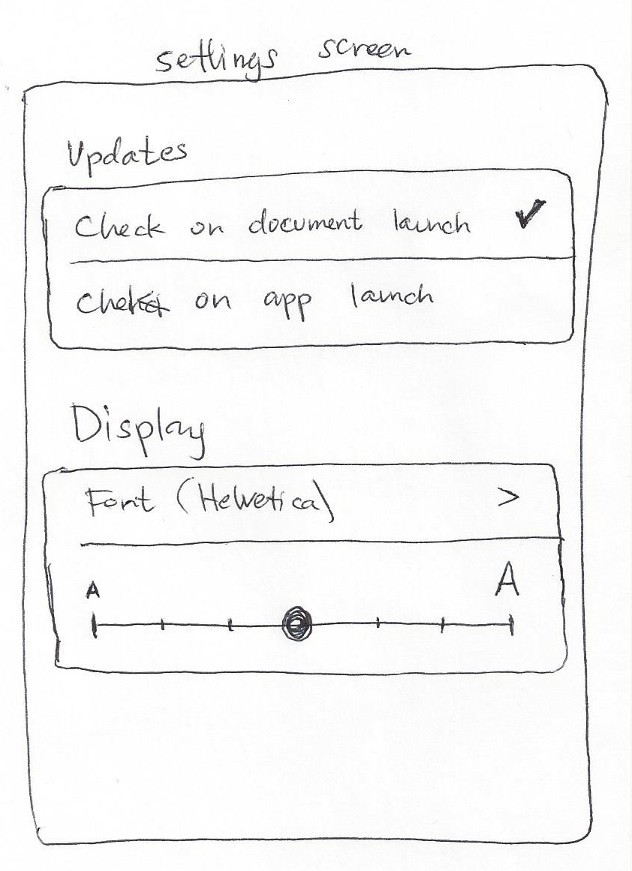
It is estimated that the Digistrux mobile application will contain the following screens:

* A splash screen with the Digistrux logo
* A list screen containing a list of all previously downloaded documents, as well as two buttons:
  + New - Launch the code scanner to download a new document.
  + Settings - Launch the settings screen for configuring the functionality of Digistrux (such as display settings, and when to check for documentation updates).
* A code scanner screen
  + This screen will activate the device's camera and provide an appropriate frame for capturing barcodes or QR codes
* A document screen
  + This screen will display the actual document downloaded from a scanned code
* Settings
  + The aforementioned settings screen

The following mock-ups illustrate what each screen might look like:







Additional Features

The Digistrux mobile application could have the following additional features:

* *Cloud storage* — This feature would allow users to identify themselves in some way, perhaps through a third-party account such as Google or Facebook, then synchronize their downloaded documents across multiple devices (or restore downloaded documents after a device failure).
* *Feedback* — This feature would allow users to submit errata or suggestions to manufacturers.

Breakdown

My approach to estimating time constraints is composed of two steps:

* Analyze the requirement in depth and calculate a best guess of the actual time
* Double it

|  |  |
| --- | --- |
| **Splash Screen** | |
| ***Task*** | ***Estimated Hours*** |
| Create view controller | 0.5 |
| Create container view | 0.5 |
| Create and center logo view | 1.5 |
| **List Screen** | |
| ***Task*** | ***Estimated Hours*** |
| Create table view controller | 1.0 |
| Create new and settings buttons | 1.0 |
| Create model for populating table view | 6.0 |
| **Code Scanner Screen** | |
| ***Task*** | ***Estimated Hours*** |
| Create view controller | 1.0 |
| Set up provisioning so I can install app | 3.0 |
| Create TestFlight project for deploying test builds | 2.0 |
| Activate camera on device | 6.0 |
| Draw frame corners | 1.0 |
| Write logic for processing bar/QA codes | 42 |
| Write logic for downloading document based on code | 9 |
| **Document Screen** | |
| ***Task*** | ***Estimated Hours*** |
| Create view controller | 1.0 |
| Write logic for parsing and displaying documents | 21 |
| **Settings Screen** | |
| ***Task*** | ***Estimated Hours*** |
| Create view controller and view | 6.0 |
| Update model logic to use settings | 1.5 |
|  | ***TOTAL*** |
|  | 104.0 |