



<https://github.com/dannyyi/410coffee>

Work Breakdown

The components of this project are fairly independent of each other, can be split into small iterations, and is required for release in three months time. As such, agile methodologies will be used instead of the standard SLDC approach. Below is a breakdown of tasks required to complete this project.

✓ = Complete

● = To do

! = To do (more difficult)

1. Server

 Apache 2 with mod-wsgi, MySQL

Amazon Web Services (AWS)



✓ Set up server and development environment: AWS instance, security policies, SSH key pair, Linux 32-bit, Python and all required modules, Apache 2

✓ Hello world Django test.

● Register SSL certificate to implement HTTPS

Time required:



Database



✓ Schema design

✓ Implement database with sqlite3 for development

✓ Create Django models based on schema, including dynamically named tables

● Migrate to MySQL for production

● Load Testing

Time required:

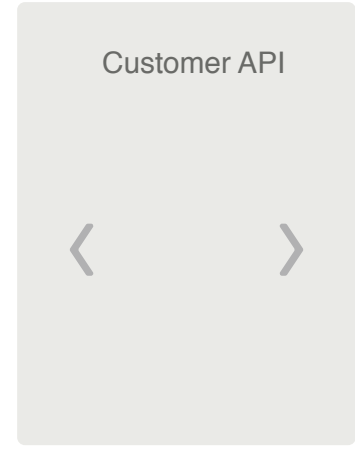






2. Web framework

 Python 2.7 (32-bit)

 Django 1.7

Customer View



 Paypal API
 Handle credit cards
 QR code redirecting
 Location filtering

● Create user group to limit customer permissions

● Design and create customer API to POST/GET JSON data

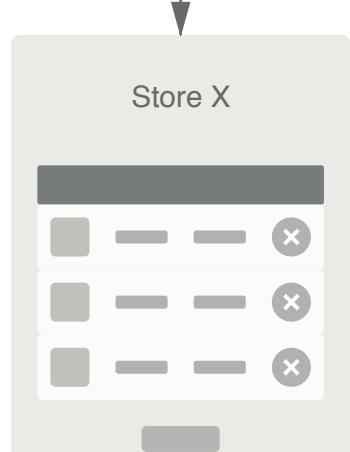
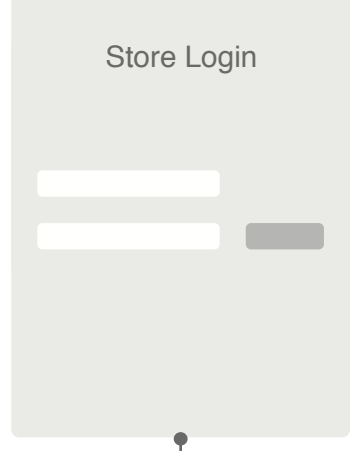
● Create automation scripts to simulate customer ordering in a real environment.

! Implement PCI compliant credit card payment handling, or PayPal API.

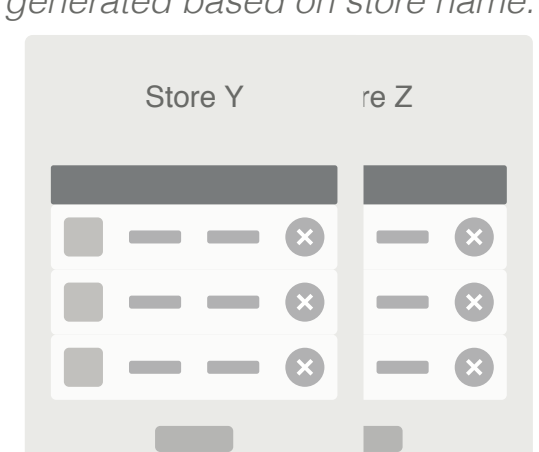
Time required:





Store Views



One page per store, dynamically generated based on store name.



 Location filtering
 Real-time order handling

✓ Generate test data into database for testing (used Tim Hortons menu, locations, and people names from NNDB).

✓ Create the store view, dynamically listing inventory based on store name.

● Create user group to grant Store permissions.

! Dynamically display customer orders in real-time. (ie. Twisted + pyopenssl)

● Create functions to handle POST/DELETE requests.

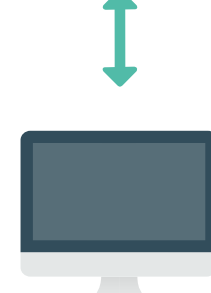
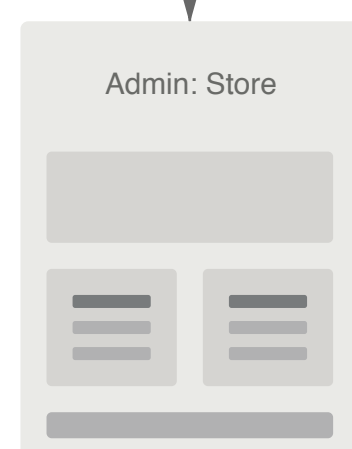
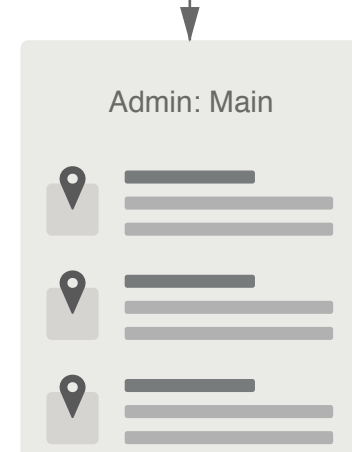
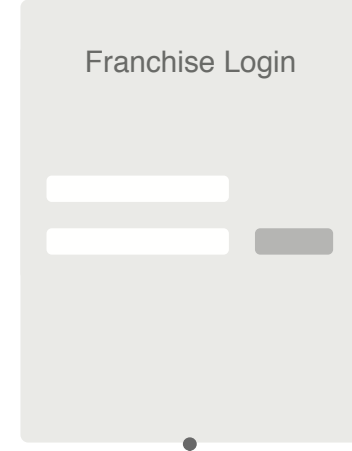
● Make some automation scripts to simulate a full franchise operating environment.

● Finalize a Django app for one store into an app template. This will be used to generate apps for every store added.

Time required:



Franchise View



● Create user group to grant Franchise permissions.


● Create admin pages to 1) add/delete stores and 2) add/edit/delete inventory for each store.

● Automation script to generate new a Django app whenever a new store is added.

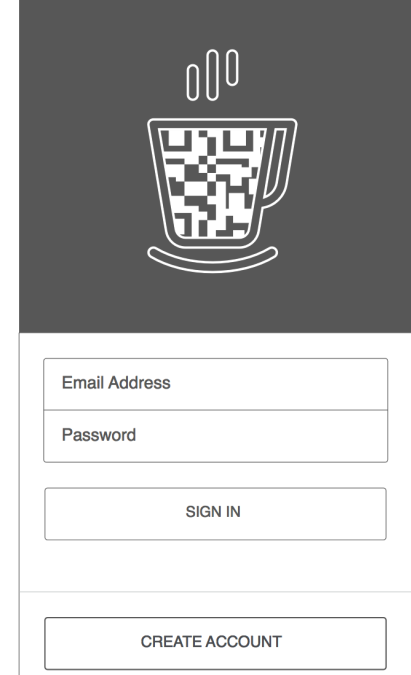
Time required:



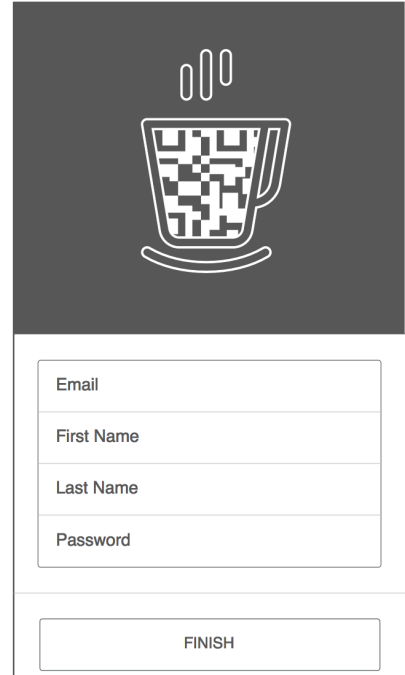
3. Mobile App

 Android: API 15+, target 19
Language Level: 7.0 (instead of standard 6.0 due to zxing library requirement)
IDE: IntelliJ IDEA 13 with Java 1.7 compiler.
UI: Sketch 3 SVG drawables

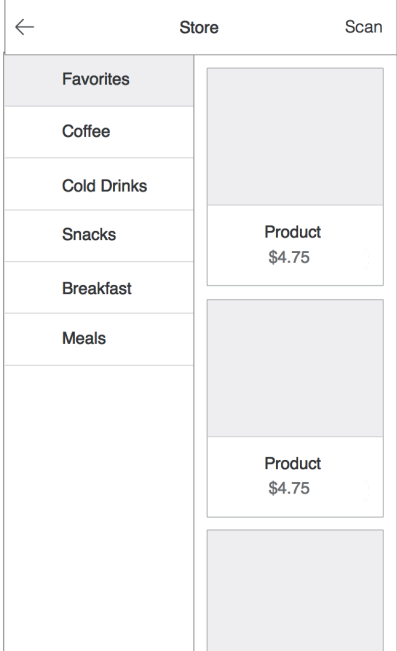
Home/Login



Register



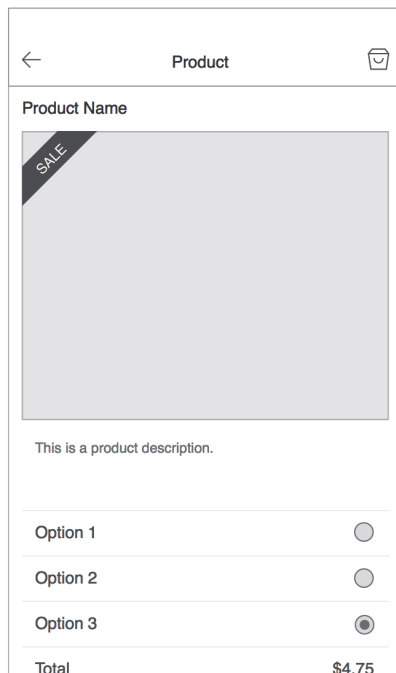
Store



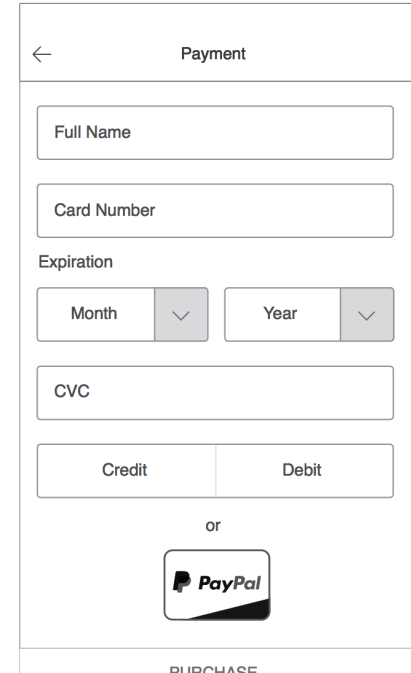
QR Scanner



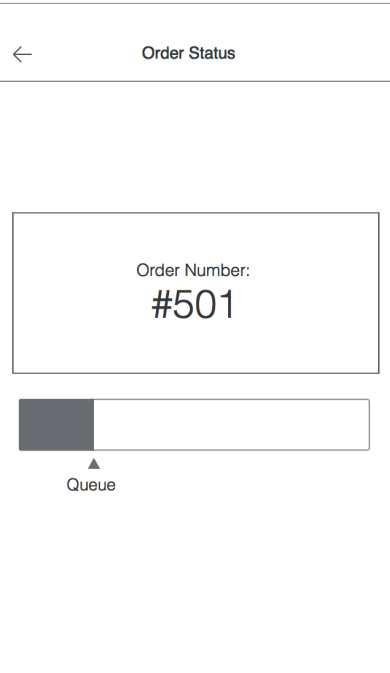
Product



Payment



Order Status



✓ Implement QR scanner

● All app pages with correct transitions.

● Test some pages with cloud

● Create a QR generating script, and test QR scanner.

● Fully functional app without payment

! Implement credit card or Paypal API

● Fully link with cloud API

Time required:

