Ryan Abel
Jason Terpstra
Ray Oesch
CIS 467-01

Design Document

Languages:

The primary languages we will use are Ruby on Rails for a web application that supports a database backend and Objective-C for an iOS application.

Frameworks/APIs

The main iOS framework we will use is MapKit. This framework is the iOS framework for utilizing maps in an application. Another iOS library we will use is called allseeing-i. This library defines communication via HTTP requests. We will need this in order to communicate with our Rails API. We will also be using various ruby gems as our server requires them.

Code Repository

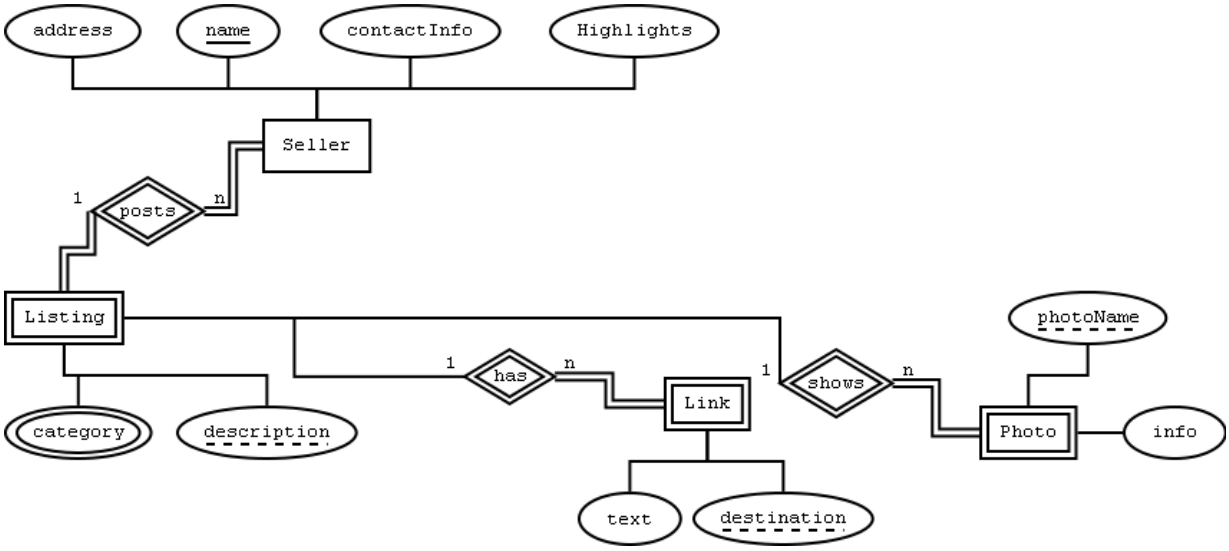
Our code will be stored in a public repository at github.com. This will allow each of our team members to look at what parts of the project other members have completed, as well as make contributions of their own. There are two separate repositories we will be pushing changes to. We will have one for our rails application and one for our iOS application. Both code bases will be separated into source and test, with their own obvious responsibilities.

Basic Organization

This project will utilize a client/server organization. All of the information on sellers and their listings will be stored in a database on the server. The clients view the database information through the iOS application, and they can make modifications or additions to the information through a web page.

Database

This project will use a database to store information. This database will be implemented in SQLite. The front end of this database system will be a web app. We will write the front end web app in Ruby on Rails.



Possible ER Diagram

User Interface

There are two scenarios for this project. In the first scenario a user will download the iOS application from the iTunes store. When they open the application they will see a map. The map will show their current location with a roughly 1 mile radius. Within the visible map will be pins for sellers and/or items. If a user selects a pin a short description of the item or place will appear. A user can select more information to see detailed information for a specific item or place. The user can also filter the pins they see based on a category or enter their own search.

In the second scenario the user will navigate to a web page. When they land at the page they will be able to log in or sign up for an account. After they sign up, they will be able to add an item to sell or a location. Adding an item will consist of a name, description, and an optional photo. The user can also view/delete their current items. See wireframes at the end of the document for sample workflows.

Alternative Approaches

We have gone through several different approaches as to what database solution we will use to keep track of each listing. The first one we looked at was Core Data, the database offered along with Xcode. We weren't sure how well this would interact with having to interact with a website as well, so we looked at alternatives. Parse.com looked promising, in that it had a very easy to use structure, as well a team member already having experience with it. Again, though, we were unsure as to how well it would integrate the mobile and internet branches. That, along with a possible fee structure that didn't quite fit our intended use led us to look elsewhere, culminating is using a Ruby on Rails database backend

Our first thought for the app was that everything you could do would be on the mobile app. However, this seemed like it might pose problems when it came to adding or editing listings. Any sort of listing management would be quite cumbersome on a mobile interface, as well as making it more difficult than necessary to upload files and edit descriptions. Having a web portal through which listings can be maintained and edited makes it much more user friendly for the

sellers.

Protocols

The main protocol we will use in this application is HTTP. We will be using HTTP requests to send and receive information from our Rails server to our iOS client application and vice versa.

Testing

We will strive to test every part of our code. We will add unit tests for any class possible to test and integration tests to test interaction between separate parts of code. We prefer the test driven development methodology when it comes to testing. This means we will write a test, make it pass, then refactor (also known as red-green-refactor).

Development Environment

Our development environment will have to be in Objective-C on a Mac for our iOS application. For the Rails application the environment is a little more flexible as Ruby can be installed on both Windows and Linux. Ruby can also be written in multiple editors including vim, TextMate, and Sublime Text.

Division of Labor

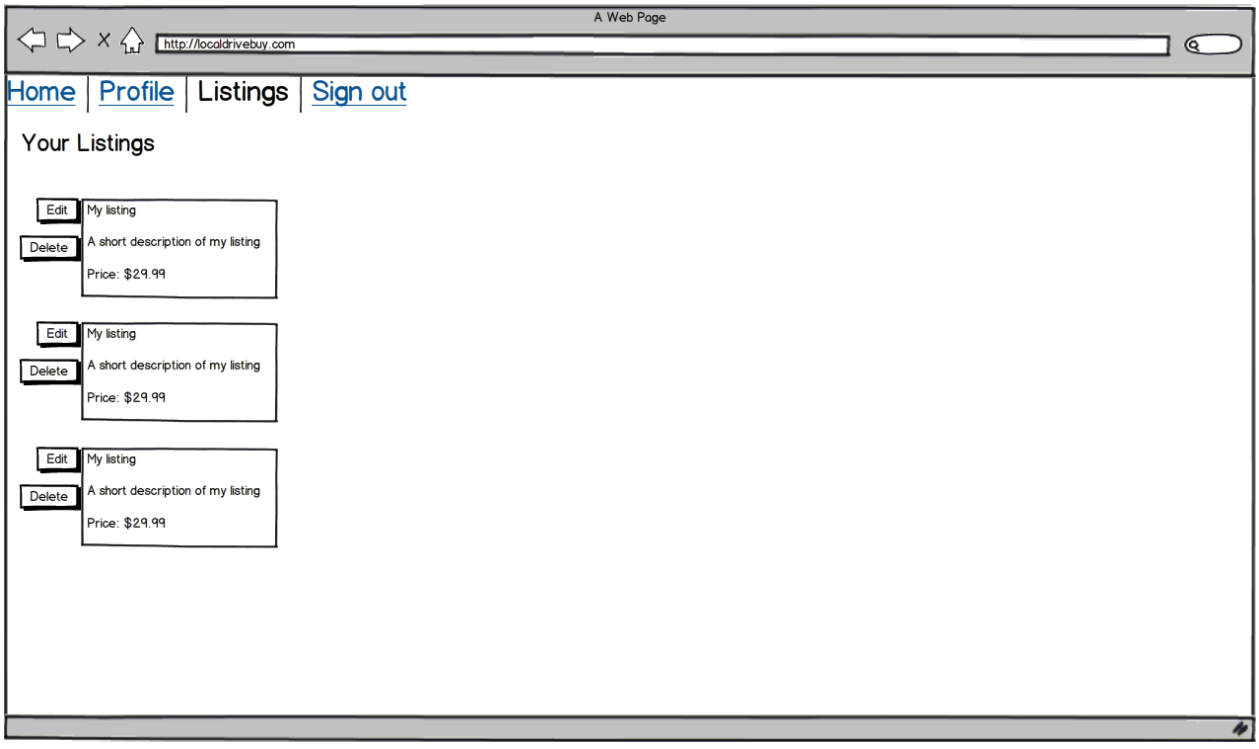
We will have all three group members working on all parts of the project. However, we also will place a single group member in charge of each individual piece of the project, and that member will be expected to make sure that the piece they are in charge of functions correctly. For example, one member will be assigned the responsibility of creating and testing a search feature within the iOS application. The other members will be able to critique and possibly refactor changes as they see necessary.

Design Methodologies

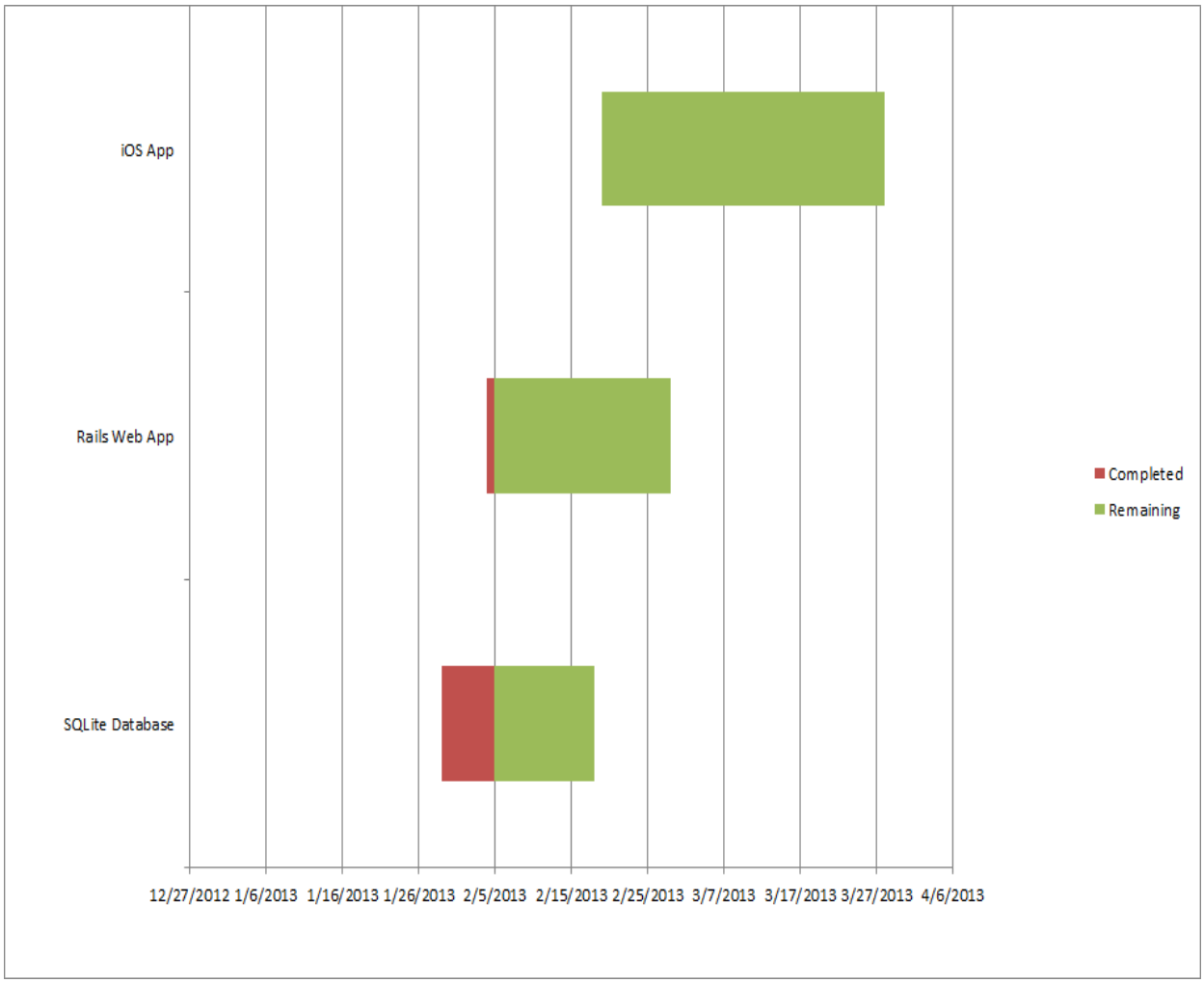
We will follow the agile design methodology. We will attempt to complete iterations every week. These iterations will contain several features that we hope to accomplish for that iteration. We will make these features available to our customer and send out a report of the previous iteration each week. This will allow our customer to stay informed and it will also give us a great roadmap to completing the project.

Wireframes


Some text
ABC 3G 03:25 PM
search
All
Edible
Goods
Free
ABC 3G 03:25 PM
search
Bob's Organic Foods
Click for more details
All
Edible
Goods
Free
ABC 3G 03:25 PM
Back
Bob's Organic Foods
100 Street Rd.
Hours: 9:00AM - 5:00PM
Bob's is an entirely organic market for customers that love fresh, homegrown food.
Reviews:
Bob's is great!
****
All
Edible
Goods
Free


A Web Page
http://localdrivebuy.com
Home
Profile
Listings
Sign out
Your Listings
Edit
Delete
My listing
A short description of my listing
Price: $29.99
Edit
Delete
My listing
A short description of my listing
Price: $29.99
Edit
Delete
My listing
A short description of my listing
Price: $29.99

Gantt Chart


iOS App
Rails Web App
SQLite Database
12/27/2012
1/6/2013
1/16/2013
1/26/2013
2/5/2013
2/15/2013
2/25/2013
3/7/2013
3/17/2013
3/27/2013
4/6/2013
Completed
Remaining