

A report on

## FANCY FOODIE : A FOOD REVIEW APP FOR IPHONE

Done by

---

Full Name	Xinjiang Shao
UIN	675469866

---

Advisor	Jakob Eriksson
Secondary Committee	Ugo Buy

---



Department of Computer Science  
UNIVERSITY OF ILLINOIS AT CHICAGO

Spring 2013

# CERTIFICATE OF APPROVAL: MASTER'S PROJECT

**UIC GRADUATE  
COLLEGE**  
UNIVERSITY OF ILLINOIS  
AT CHICAGO

601 S. Morgan Street (MC 192)  
<http://grad.uic.edu/>

Student Name: \_\_\_\_\_ UIN: \_\_\_\_\_  
Last \_\_\_\_\_ First \_\_\_\_\_ M. \_\_\_\_\_

Degree Program: \_\_\_\_\_ Degree Sought: \_\_\_\_\_ Program Code: 20FS

Project Title: \_\_\_\_\_

Results: \_\_\_\_\_

**SATISFACTORY**

Project meets the standards of scholarly performance expected of master's candidates in the field.

**UNSATISFACTORY**

Project lacks the minimum criteria for approval.

Grade Change:

(Indicate below any research courses requiring a grade change upon completion of the project.)

Subject	Course #	Course Ref#	Credit Hours	Term	Current Grade	Final Grade

## AUTHORIZED APPROVAL

Major Advisor: \_\_\_\_\_ Date: \_\_\_\_\_

Committee: \_\_\_\_\_ Date: \_\_\_\_\_  
\_\_\_\_\_ Date: \_\_\_\_\_  
\_\_\_\_\_ Date: \_\_\_\_\_  
\_\_\_\_\_ Date: \_\_\_\_\_  
\_\_\_\_\_ Date: \_\_\_\_\_

Department DGS: \_\_\_\_\_ Date: \_\_\_\_\_

Graduate College: \_\_\_\_\_ Date: \_\_\_\_\_

## **Abstract**

Food review is become trendy among young people since smart phone get more and more popular. However, not many tools available for people to review a cuisine and publish their comment to social media conveniently in the market. This project is meant to develop an iOS App called *Fancy Foodie* that can take picture of food, attach meta data such as location, rate, comment, date, tags to the picture and sharing with friends via social media including Facebook, Twitter, Weibo etc.. The App also provides ways to search events by address and review statistic of all events.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Scope of Project . . . . .	1
1.2	Motivation . . . . .	1
1.3	Deliverable . . . . .	2
1.4	Intended Audience . . . . .	2
1.5	Platform for Deployment . . . . .	2
1.6	Development Platform . . . . .	3
1.7	Roadmap . . . . .	3
<b>2</b>	<b>The Graphic User Interface</b>	<b>4</b>
2.1	Home Tab . . . . .	4
2.2	Food List Tab . . . . .	4
2.3	Stats Tab . . . . .	6
2.4	Map Tab . . . . .	6
2.5	Setting Tab . . . . .	6
<b>3</b>	<b>Design</b>	<b>12</b>
3.1	Software Architecture . . . . .	12
3.1.1	Relationship with Cocoa Touch Framework . . . . .	12
3.1.2	Database Schema . . . . .	12
3.1.3	Settings Property List . . . . .	15
3.2	Modules . . . . .	15
3.2.1	View Controllers . . . . .	15
3.2.2	Models . . . . .	15
3.2.3	Third party libraries and Utilities . . . . .	15
3.3	Key Methods . . . . .	15
3.3.1	Insert New Event . . . . .	15
3.3.2	Update Event . . . . .	15
3.3.3	Delete Event . . . . .	15
3.3.4	Searching Events . . . . .	15

<b>4 Testing</b>	<b>18</b>
4.1 Unit Test, Integration Test and System Test . . . . .	18
4.2 Beta Testing . . . . .	18
4.3 iTunes . . . . .	18
4.4 Future Testing . . . . .	19
<b>5 Conclusion</b>	<b>20</b>
<b>Acknowledgements</b>	<b>21</b>
<b>References</b>	<b>22</b>

# List of Figures

2.1	Home Tab View . . . . .	5
2.2	Foodie List Tab View . . . . .	7
2.3	Statistics Tab View . . . . .	8
2.4	Map Tab View . . . . .	9
2.5	Setting Tab View . . . . .	11
3.1	Relationships with Cocoa Touch Framework . . . . .	13
3.2	Database Schema . . . . .	14
3.3	Class-diagram of the project . . . . .	16
4.1	TestFlight Dashboard . . . . .	19

# **Chapter 1**

## **Introduction**

### **1.1 Scope of Project**

Mobile Apps are making our life more and more interesting than ever before. A lot of young people love to take photo with their smart phone, making comments to it and sharing with their friend about what they had done. I personally find it would be useful if there is a tool for anyone to create food event and share with friends.

The particular aim of the project is to be able to review food and sharing food event with friends. Before you start eating, the user would take a picture of the food they are going to eat and give some basic information about location, tags, date etc. Tags are used to describe the type of the food such as “Chinese, Bun” or “Tea, Classic, Jasmine Green”. After finish eating, the user would make comments of the food. This application also lets user search nearby location or location defined by user, which will show a list of pins on the map where the user had been there before, to decide what the user want to eat. When the user saves a food event, he or she may also share the review and foodie’s photo to Facebook or Twitter or to an email address. For user to see the history records, the application also provide a way to view statistics data such as how many places the user has been, how the rates are, how many tags the user used etc. ?

### **1.2 Motivation**

Seeking for places to eat is a thing in my genuine. I always want to keep track of my food adventures so that next time I could have better sense of what kind of cuisines I should order. I could also give recommendations to

my friends about the adventures. Luckily, it turns out that I am not the only one who want to do such kind of thing. A few of my friends tell me that people nowadays love to take photos of the food before they eat, and post their photos to all kinds of social media such as Facebook, Twitter, Weibo (Chinese Twitter) etc.. Especially asian students have really strong motivation to take pictures of food when they eat.

After searching the apps available in apple app store, I didn't find any suitable choice for my need. So I came up with an idea that I need to make best use of skills and build a good app for people who like to explore food world with friends.

Currently, various of apps related to foodie are available in apple's app store. But most of them focused on the whole store/restaurant review. This could be inaccurate sometimes because you might just hate one dish. And there are also a few recipe-related apps. But the project want to have a better tool to publish what you eat instead of how that dish is made.

### 1.3 Deliverable

An iOS app called *Fancy Foodie* is built and submitted to apple's app store via iTunes connect. The version 1.0 is released. The category of the app are "Food Drinks" and "Lifestyle". The default language for *Fancy Foodie* is English. Another website( <http://soleo.github.io/Fancy/> ) is built to support the app.

### 1.4 Intended Audience

The target users for *Fancy Foodie* is people who love to try different kind of food and enjoy using social media to publish their experience of food adventures.

### 1.5 Platform for Deployment

- Operating System: iOS 6.0 or later
- Hardware: iPhone 4, 4S , 5 or iPod Touch 4th Generation

## **1.6 Development Platform**

- Operating System: Mac OS X 10.8.3
- Processor: 2.9 GHz Intel Core i7
- Memory: 8 GB 1600 MHz DDR3
- IDE: Xcode 4.6 with ARC
- Version Control: git

## **1.7 Roadmap**

Fancy Foodie 1.0

### **Core Features**

- Taking picture of food, stored all meta data in local database
- Listing all the food events in a table view and detailed view
- Showing basic statistics information of the events
- Searching by using current user location or user input address
- Configuration of fancy foodie

### **User Interface Elements**

- Created App Logo, Navigation Bar, Icons and Backgrounds.
- Used FontAwesome and Droid Font as default in App

### **Testing**

- Using TestFlight do beta Testing
- Tested in iPhone 5, iPod Touch 4th Generation

# **Chapter 2**

## **The Graphic User Interface**

### **2.1 Home Tab**

As shown in Figure 2.1, home tab is created for adding new event to the app. The default view for user entering the app is Figure 2.1(a).

Figure 2.1(a) gives a short tutorial for user. If the user chooses the plus sign in navigation bar, a another blank view should show up. After tapping on the camera icon, Figure 2.1(b) will display an action list including “Use Last Photo Taken”, “Take Photo”, “Choose from Library” and “Cancel”. For instance, we choose “Take Photo”, the app should pop up a modal and let user take picture of food. A modal like Figure 2.1(c) should appear for user to move and scale the photo to a right position and size. Figure 2.1(d) will show up after scaling step. If no photo is chosen, a warning will show up to alert user add a photo first. Tap on next to move to next view Figure 2.1(e). In this view, a form is created for user to fill in location information, date, tags, comment and rate. Since the user just starts to eat, comment and rate field can be empty for now. Other fields should be filled correctly in this view because the user won’t be able to edit all the other fields. In Figure 2.1(g), the location list is generated through Foursquare API. Foursquare API is chosen here because it has a good reputation in both academic and industry world. By passing current longitude and latitude, we’re able to have a venue list based the distance.

### **2.2 Food List Tab**

In food list tab, it fetches a list of food events ordered by creation date. Figure 2.2(a) shows some events. On each table cell, it shows a thumb-

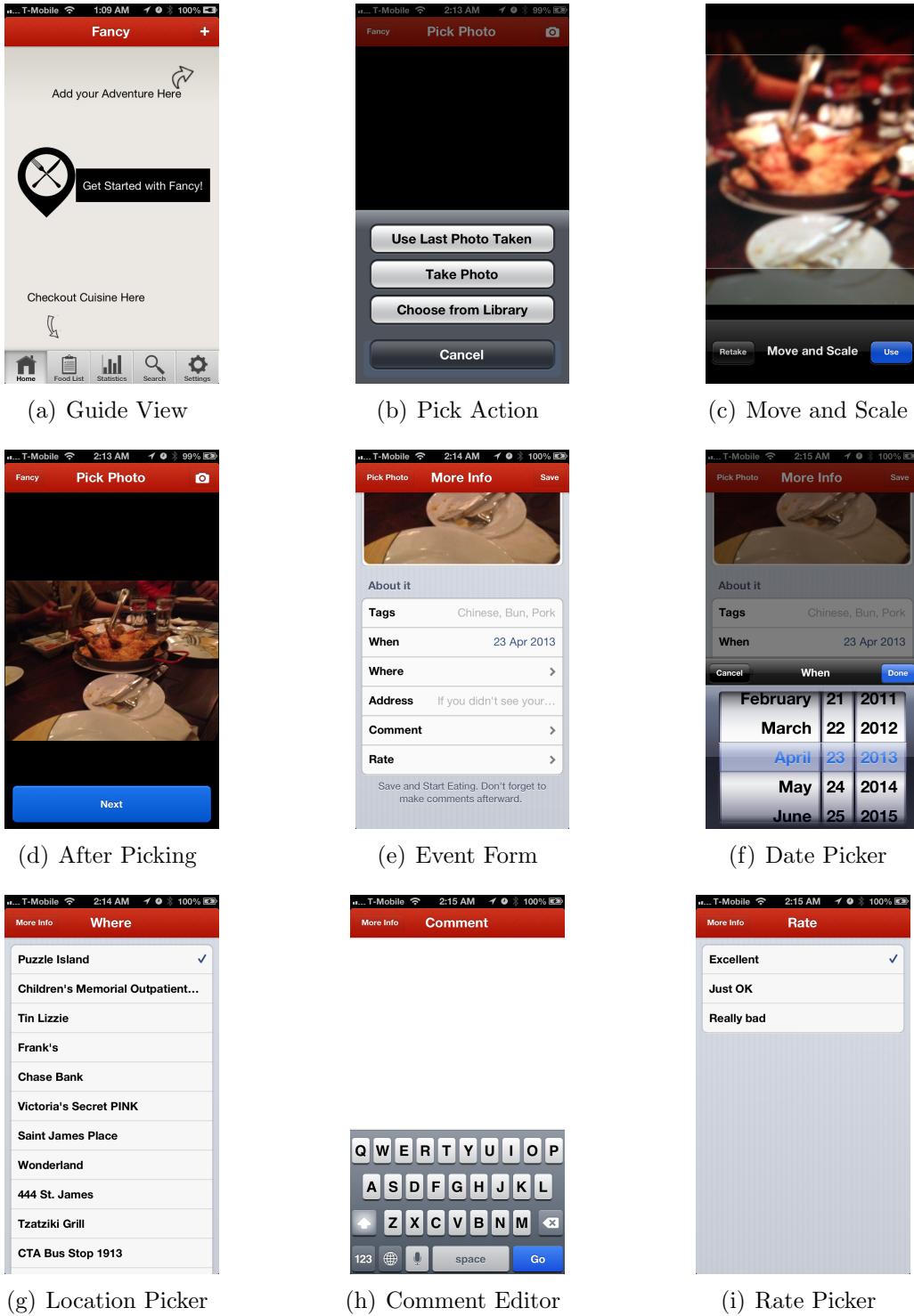


Figure 2.1: Home Tab View

nail, time string and string of the event. And a green menu pop up button is created for sharing with social media(Figure 2.2(e)) and updating comment(Figure 2.2(c)) or rate(Figure 2.2(d)).

After choosing one event, a detail view of the event will appear as shown in Figure 2.2(f) and Figure 2.2(g). You could see all the detailed information when you create the event. In the top navigation bar, a share button is displayed for user to share the event with friends. If the user chooses twitter, it'll check if he is logged in or not. After making sure the user have a twitter account, Figure 2.2(h) will appear. The comment and photo is generated by the app. Simply tap on send, and the event will be shared.

## 2.3 Stats Tab

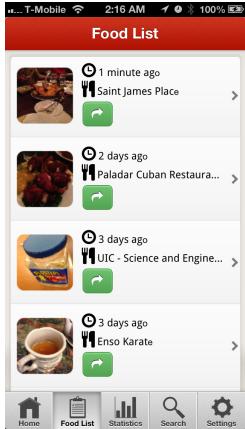
Statistics tab is supposed to give the user an overview of all events he created. As shown in Figure 2.3, “Rates” are clustered in different categories. All the tags are counted and shown in the table. The total number of places are shown in the table as well. In this way, the user can easily get an idea of what kind of places and food he likes.

## 2.4 Map Tab

In Figure 2.4(a), all the event locations are labeled with red pins. By default, it is showing the events around your current location. But you could also change your view by passing address in Figure 2.4(b). After choosing a location in the list, the tableview will be dismissed and the new central region will be the location you chose.

## 2.5 Setting Tab

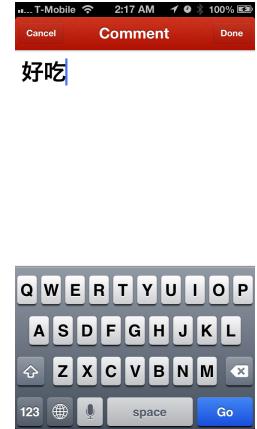
Setting tab provides a view to setup configurations of the app (Figure 2.5). The view in this tab is a static table view built in Storyboard. A web-view controller is created to show the app website and author introduction. By turning on the save photo option, it will save the photo to a album. The feedback option is used for user to submit feedback through TestFlight. TestFlight is a online tool to do open beta testing on the fly. By hooking with TestFlight, we will be able to see all the crash reports, time durations



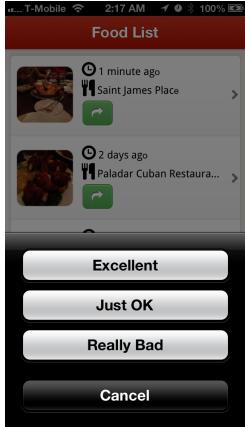
(a) Food List



(b) Menu Popup



(c) Comment Editor



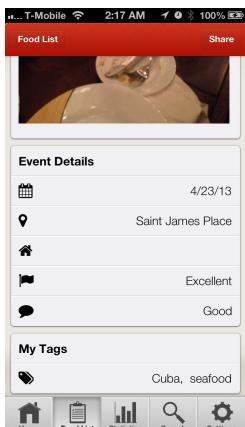
(d) Rate Action Sheet



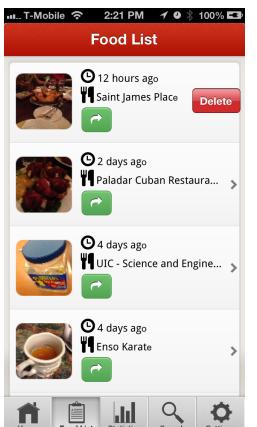
(e) Sharing Activity Controller



(f) Detail View

(g) Detail View  
Cont'd

(h) Twitter



(i) Delete View

Figure 2.2: Foodie List Tab View

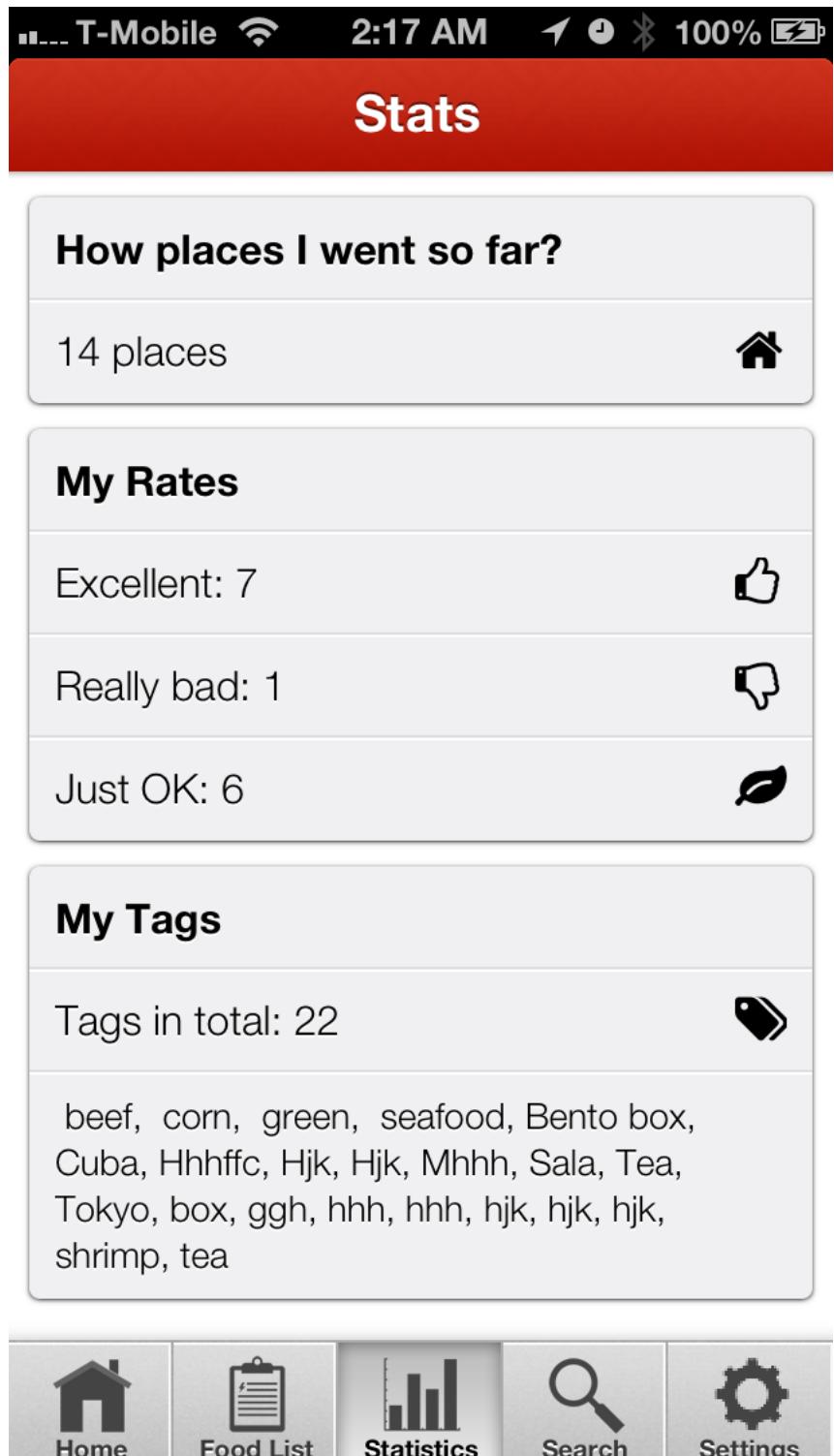
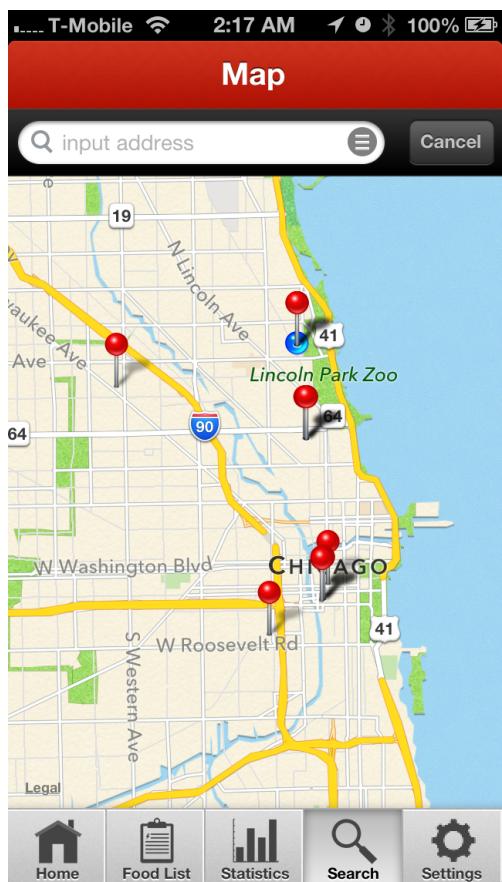
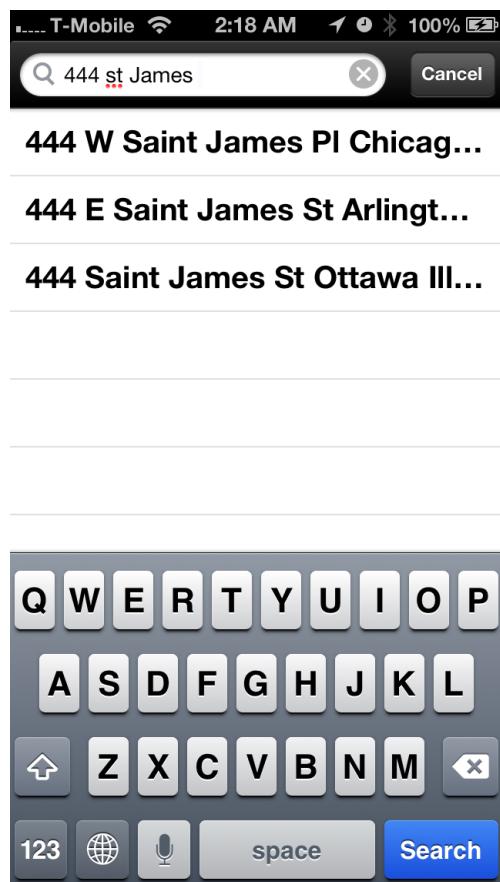


Figure 2.3: Statistics Tab View



(a) Map



(b) Address Searching

Figure 2.4: Map Tab View

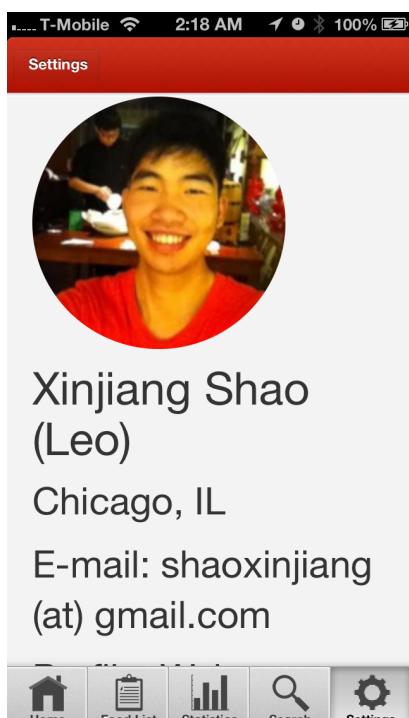
for each session, and even ask questions to users after a checkpoint is reached.



(a) Settings



(b) Save to Album



(c) Author Web-view



(d) Website View

Figure 2.5: Setting Tab View  
11

# Chapter 3

## Design

### 3.1 Software Architecture

#### 3.1.1 Relationship with Cocoa Touch Framework

Core Data, Core Location, WebKit, MapKit, UIKit are used in this projects. As shown in Figure 3.1, NSManagedObject, NSObject, UITableViewController, UIViewController are inherited by different class in the project.

#### 3.1.2 Database Schema

In this project, we used three table for storing all the information from users. As Figure 3.2 shows, Events is the main table in the app. It provides fields “address”, “comment”, “creationDate”, “latitude”, “longitude”, “locationName”, “rate”, “thumbnail”, “photoBlob” and “tags”. “address” field is used when user didn’t find their “locationName” in location List fetching from Foursquare API v2. “Latitude” and “longitude” is used for adding annotations in Map View. A 80\*80 resolution thumbnail is stored for each event in order to accelerate loading in food list table.

“photoBlob” has one to one relationship with “photo” in PhotoBlob table. Using a separate table should also help speeding up when we don’t need to load photo while we still need to get the meta data of the event.

Field “tags” has many to many relationship with “photos” in Tag table since one photo can labeled with many tags and one tag can relate to many photos.

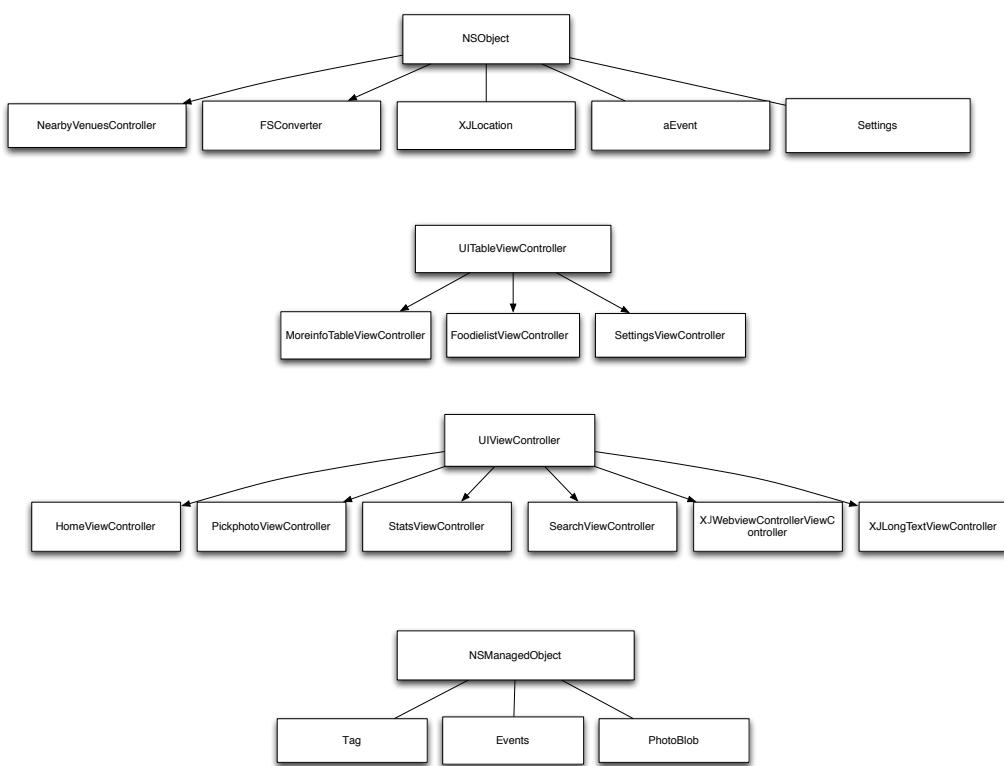


Figure 3.1: Relationships with Cocoa Touch Framework

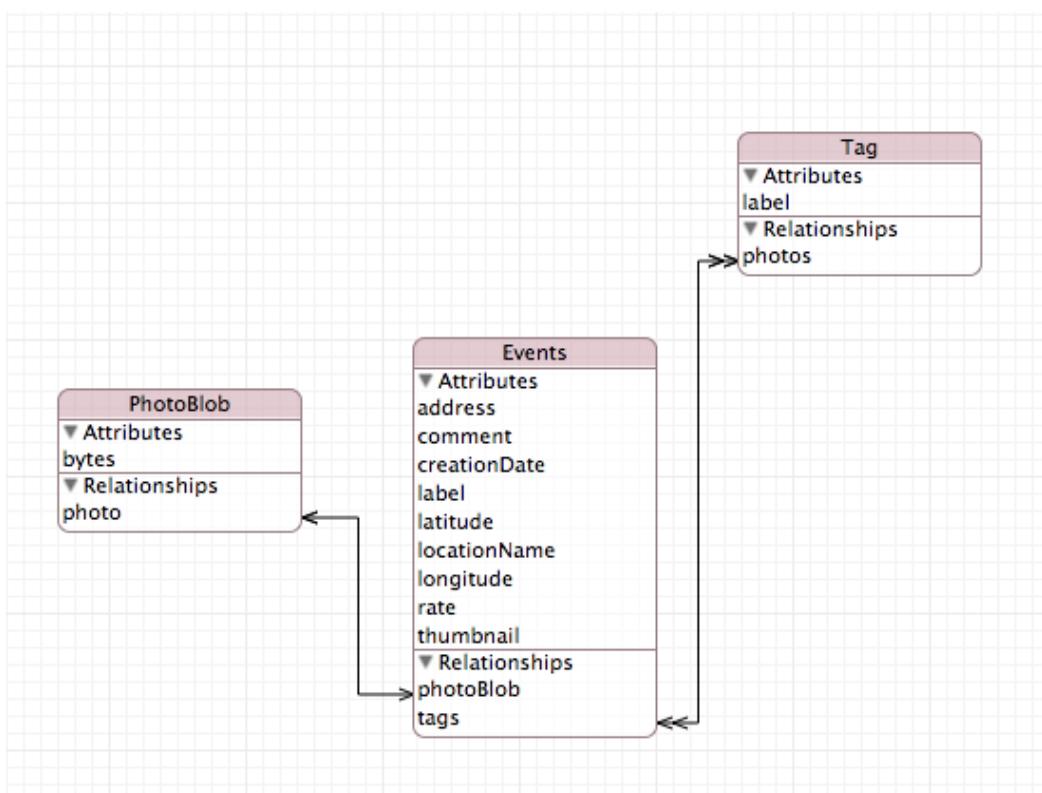


Figure 3.2: Database Schema

### **3.1.3 Settings Property List**

When developing iOS, we can also use another way to store data we need. Property List is used to store all information related to the app itself. For instance, using Property List to store App Display Name, App Version are commonly used in all apps for iPhone.

“Fancy Foodie” uses release number as main version string and git hash tag of the release as build string, so “1.0 (build 98a9e84)” is shown in Figure 2.5 which means that the main version number is 1.0 and build hash tag is 98a9e84.

This app also use property list to store whether we need to save photo to album locally. If this option is enabled, the app will create a album named “Fancy Foodie Photos” and put photos in this album as shown in Figure 2.5(b).

## **3.2 Modules**

### **3.2.1 View Controllers**

### **3.2.2 Data Models**

### **3.2.3 Third party libraries and Utilities**

The following is a list of third party libraries used in the project.

- TestFlightSDK: Beta Testing on the fly.
- BaseKit: Tools to create singleton class and better Location Manager.
- QBPopupMenu: Popup Menu User Interface
- CZPhotoPickerController: Preview of photos and better photo picker.
- FormKit: Form style tableview creation.
- BButton: Button with twitter bootstrap color schema.
- Foursquare2: Foursquare API version 2.
- MGBox: Table style boxes creation.
- SORelativeDateTransformer: Convert date to relative date such as “One day ago”.

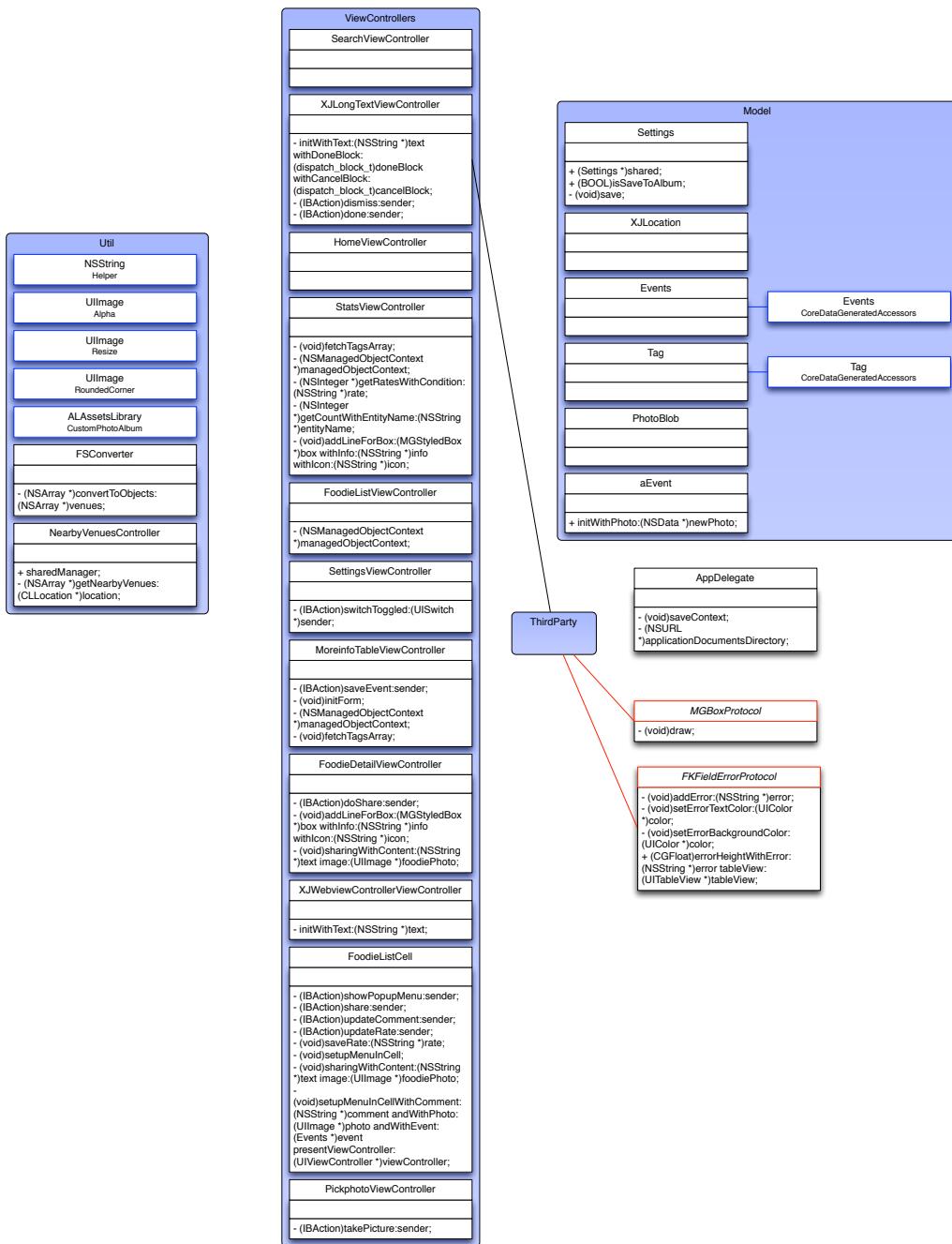


Figure 3.3: Class-diagram of the project

- FontAwesome: Adding icons to user interface.
- SVProgressHUD: Notification messages display.

## 3.3 Key Methods

### 3.3.1 Insert New Event

Figure ?? shows the

### 3.3.2 Update Event

### 3.3.3 Delete Event

Delete event is very simple in this app. In food list tab, you'll see a list of

### 3.3.4 Searching Events

Searching events is done with MapKit API. All the events are fetched in during loading time. The events are annotated with red pins when the tab is loaded.

As shown in Figure 2.4(b), as the user is typing in the address, the searching string is passed to search controller, so at the same time, we're using geo coder to guess the possible locations for that string. The possible locations are displayed in a table view. After the user choose one possible location, the central region will be focused on that area. At this time, the nearby events stored in database will appear in front of the user.

# **Chapter 4**

## **Testing**

### **4.1 Unit Test, Integration Test and System Test**

All the modules are tested individually in separate Xcode project, and integrated into our main project *Fancy Foodie*. After making sure the features are integrated with the main project, we build the project nightly, and test the function systematically with both stimulators and real cellphone which is iPhone 5.

### **4.2 Beta Testing**

*Fancy Foodie* use TestFlight to do beta testing on the fly. TestFlight provides a SDK for us to integrate it to the App. After integration, crash reporting will happen automatically. By logging in the dashboard of TestFlight, we could see all the information for each session of users as shown in Figure 4.1.

### **4.3 iTunes**

*Fancy Foodie* is created in iTunes Connect on April 17, 2013. First submission was uploaded on April 17, 2013. But minor changes are made after that, so another submission was upload on April 23, 2013. Currently, *Fancy Foodie* is waiting for review. After review, <https://itunes.apple.com/us/app/fancy-foodie/id638036832?ls=1mt=8> should be available to U.S. users. Anyone could download the App via that link.

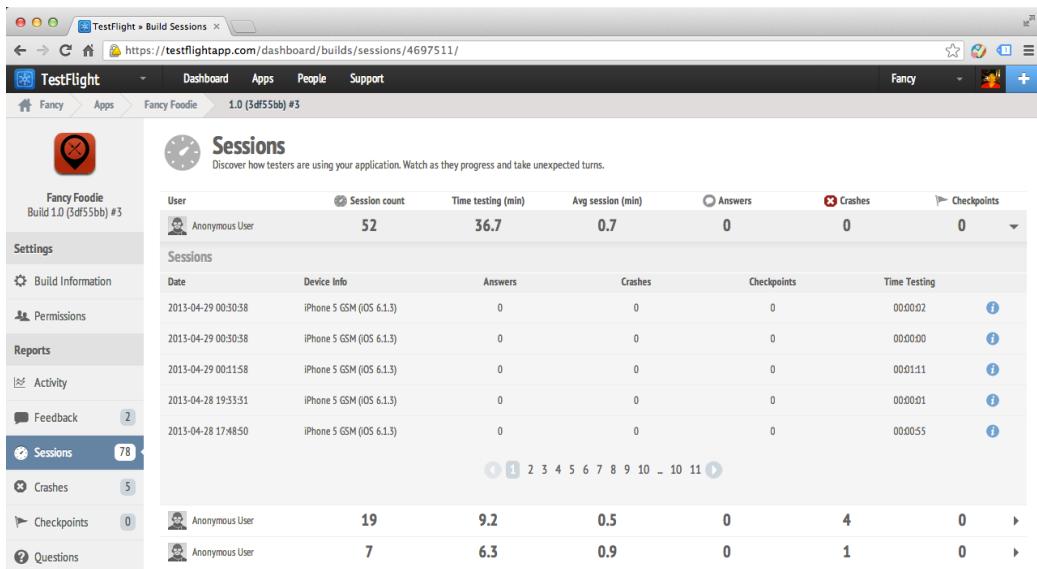


Figure 4.1: TestFlight Dashboard

## 4.4 Future Testing

Since it is ongoing project, future testing will be done by using TestFlight as well. We would set up checkpoint and create questionnaire to get better feedback from users.

# **Chapter 5**

## **Conclusion**

This project is successfully finished in time with good functionalities overall. It could provide users to take picture of food they like and create tags, location information, rates, and comments to the food. At the time, it provides convenient methods for users to share those food with their friend by social media such as Facebook, Twitter, Weibo etc.. A map view and statistics view are created for users to have better understanding of their preferences.

There are still more features could be done in future. Since Yelp is a good resource of restaurant information, the App could pull more information from Yelp in order to give users more about the place they eat at. A central database is needed to making users communicate with each other such as making comments to friends food inside of the app. Chinese is my mother tongue, so I'm hoping to adding multi-language feature for the App. In this way, more people could be able to use the App. Last but not least, a larger pool of testers should be included in beta testing process.

# Acknowledgments

I would like to thank open source community. Without them, I probably need much more time to finish this project.

I would also like to thank Jakob Eriksson and Ugo Buy. Prof. Eriksson gave me a lot of valuable suggestion for my project. Prof. Eriksson and Prof. Buy both give a good evaluation of my work.

Special thanks to all my friends who made tremendous comments about my app and all my colleagues in EXACT Sports LLC who slow me down in the right way so that I can make the project better.

Xinjiang Shao  
April 2013  
University of Illinois at Chicago

# References

- [1] iOS Cocoa Touch, <https://developer.apple.com/technologies/ios/cocoa-touch.html>
- [2] Apple Developer Center, <https://developer.apple.com/devcenter/ios/index.action>
- [3] Cocoa Controls, <https://www.cocoacontrols.com/>
- [4] AppCoda: iOS Programming Course, <http://www.appcoda.com/ios-programming-course/>
- [5] Code for App, <http://code4app.com/>