

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



UNIVERSITY OF ILLINOIS
AT CHICAGO

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

Student Name: _____ **UIN:** _____

Student Name: _____ UIN: _____
Last _____ First _____ M. _____

Last

First

M.

UIN:

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.)

AUTHORIZED APPROVAL

Major Advisor:

Date:

Committee:

Date:

Date:

Date:

Date:

Department DGS:

Date:

Graduate College:

Abstract

Here is abstract.

Contents

1	Introduction	1
2	Background	2
2.1	Motivation	2
2.2	Restaurants Review Apps	3
2.2.1	Yelp	3
2.3	Recipe Related Apps	4
2.3.1	Evernote Food	4
3	Project Design	5
3.1	System Requirement	5
3.2	Design Philosophy	6
3.2.1	Singleton Design Pattern	6
3.2.2	Model-View-Controller	6
3.3	Architecture	7
4	Implementation	10
4.1	Data Model	10
4.1.1	Database Schema	10
4.1.2	Settings Property List	10
4.2	Workflow	13
4.2.1	Insert New Event	13
4.2.2	Update Event	13
4.2.3	Delete Event	13
4.2.4	Searching Events	13
4.3	User Interface Design	14
4.3.1	Home Tab	14
4.3.2	Food List Tab	14
4.3.3	Stats Tab	17
4.3.4	Map Tab	17
4.3.5	Setting Tab	17

5 Future Work	21
6 Conclusion	22
Acknowledgements	23
References	24

List of Figures

3.1	Storyboard of Overview Fancy Foodie	7
3.2	Storyboard of Home Tab View	8
3.3	Storyboard of Food List Tab View	8
3.4	Storyboard of Statistics View	8
3.5	Storyboard of Settings View	9
3.6	Storyboard of Map View	9
4.1	Database Schema	11
4.2	Home Tab View	15
4.3	Foodie List Tab View	16
4.4	Statistics Tab View	18
4.5	Map Tab View	19
4.6	Setting Tab View	20

Chapter 1

Introduction

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.

This project is a food review application called “Fancy Foodie” for iOS. 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. ?

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.

Chapter 2

Background

2.1 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 courses 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 want to do such kind of thing. A few of my friends tells 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.

2.2 Restaurants Review Apps

2.2.1 Yelp

Yelp is used by tremendous people to get rate of restaurants in order to pick a place to eat. The searching filter for Yelp app is really good since they have a very rich database of different kinds of restaurants as well as different reviews from many people. The app provide a way to check with map and give direction the restaurant which is helpful for people to find the place. On the other hand, since they have a really large database, it is not easy to find what exactly you want to go. And most of the reviews are from people you don't know, it is highly possible that the taste of the people might be different from yours. Let alone there are people paid to make fake good reviews or bad reviews for some restaurants.

In this app, you're not using this looking for suggestions of where you're going to eat, but you need to get suggestions from your social media where your friends published restaurants reviews. In this way, fake reviews will be avoided.

2.3 Recipe Related Apps

2.3.1 Evernote Food

Food app from Evernote did a good job showing the meals. The user interface is really friendly. Adding tags to the meal is easy and choosing place information is convenient in Food. The app also provided functionality to add a cuisine recipe as well. But people likes to take pictures of their food not always into making food. So in our project, we won't do any recipe recording.

Chapter 3

Project Design

3.1 System Requirement

- Operating System: iOS 6
- Compile with Automatic Reference Counting (ARC)
- Hardware Used: iPhone 5 and iPod Touch 4th Generation

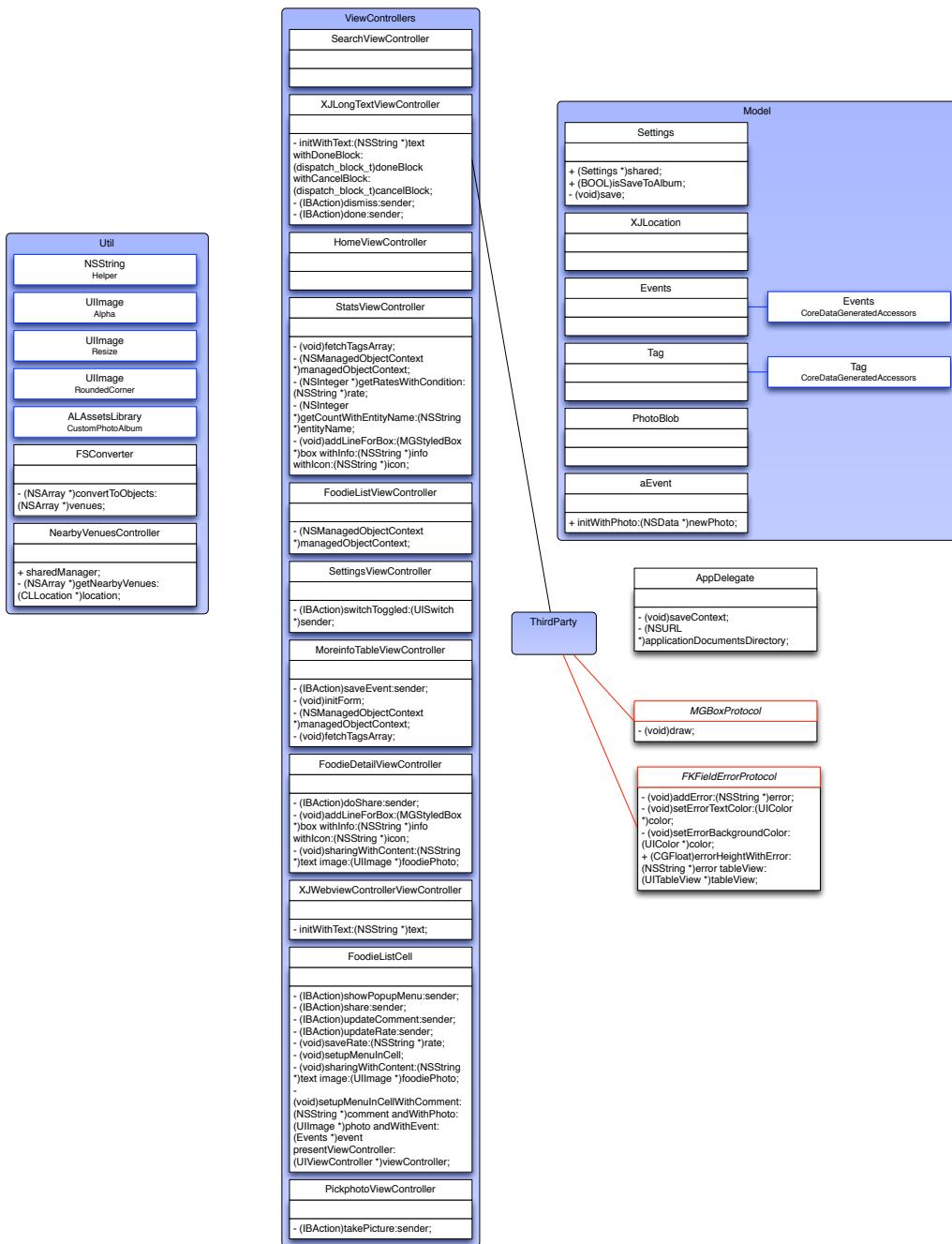


Figure 3.1: Database Schema

3.2 Design Philosophy

3.2.1 Singleton Design Pattern

Design Pattern is used in this project. In software engineering, the Singleton Pattern aims at creating the instance of a class only one time so that we don't need to create twice. It is especially useful when you need to use a method of the class while the class is just a toolkit for you. In this project, we could use this pattern to create a Settings instance in order to keep Settings globally shared inside of the app. There are a bunch of other places use this pattern as well, such as NearbyVenueController, LocationManager etc.

3.2.2 Model-View-Controller

Another popular design philosophy in software engineering field is Model-View-Controller. This separate each parts so that we could focus on one thing instead of all together. We could call it as a divide and conquer method. The benefit to use it is that if we changed one Controller function, sometimes we don't need to update the view for it. It is a time and life saver for app designing.

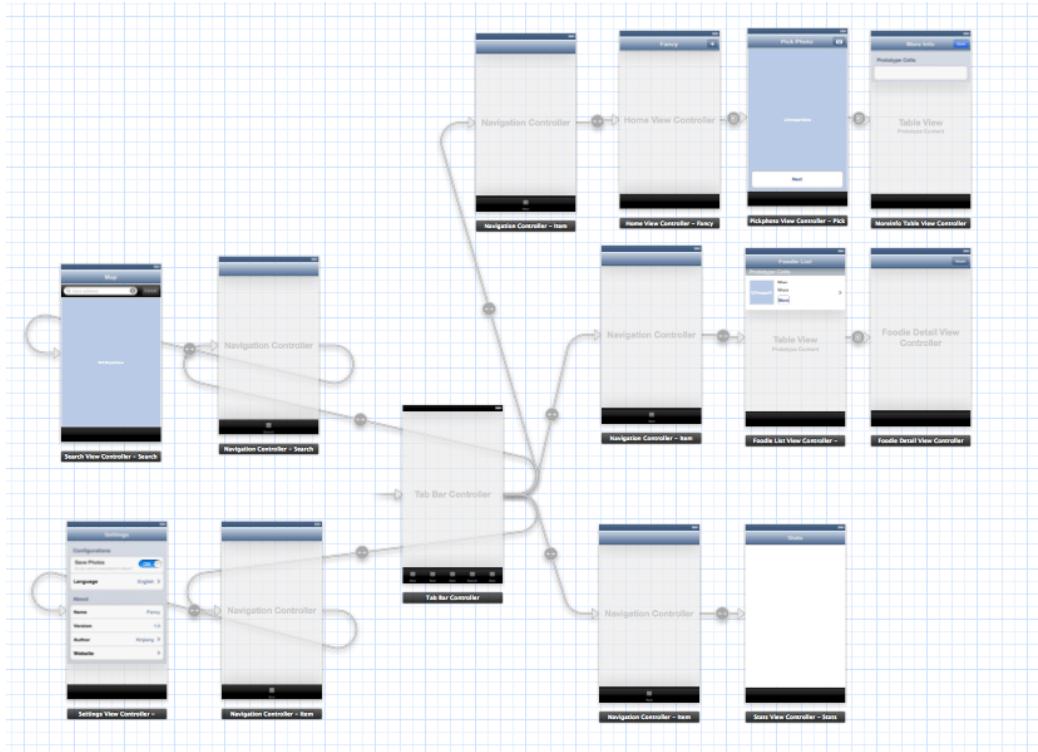


Figure 3.2: Storyboard of Overview Fancy Foodie

3.3 Architecture

Figure 3.1 shows main views in storyboard. In this project, we build it as a tab based application. Five tabs are created for different purpose. Home tab plays as a guide to create a foodie event; Foodie list tab shows all the events created before; Statistics tab shows all aggregated data; Searching tab is used to show all past events and searching by address to find the events locations; Setting tab is for configurations.

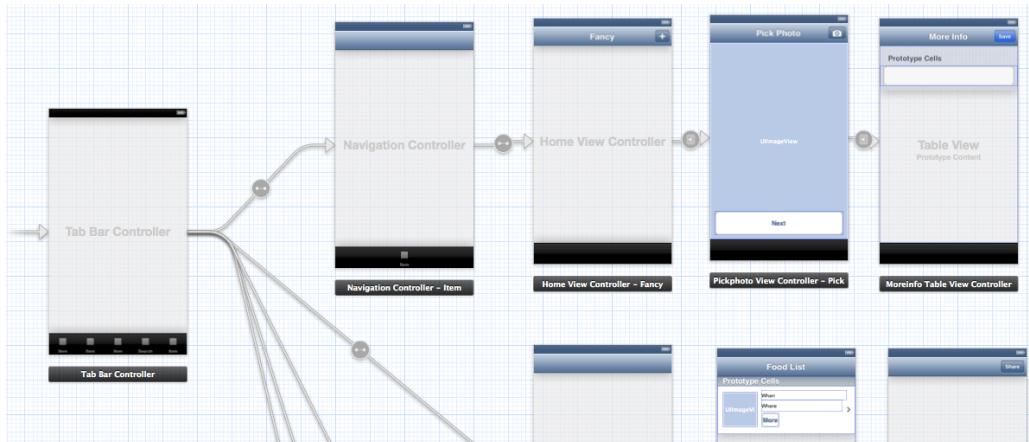


Figure 3.3: Storyboard of Home Tab View

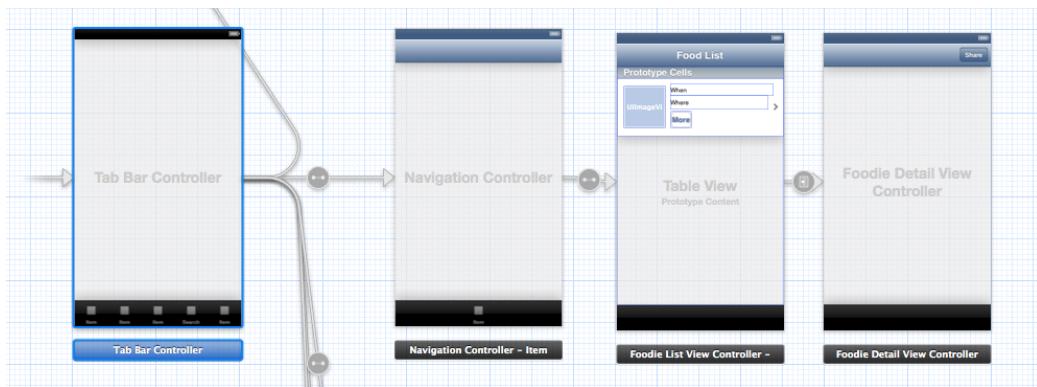


Figure 3.4: Storyboard of Food List Tab View

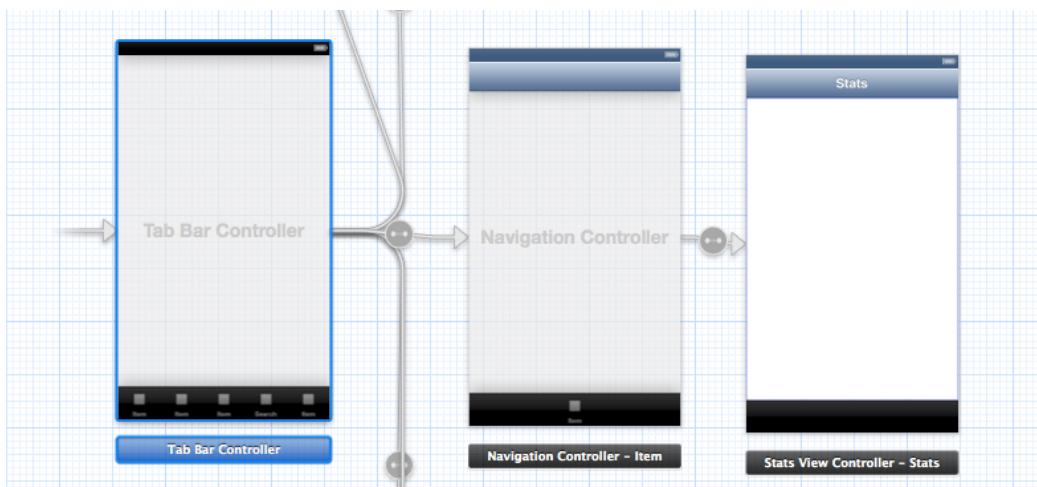


Figure 3.5: Storyboard of Statistics View

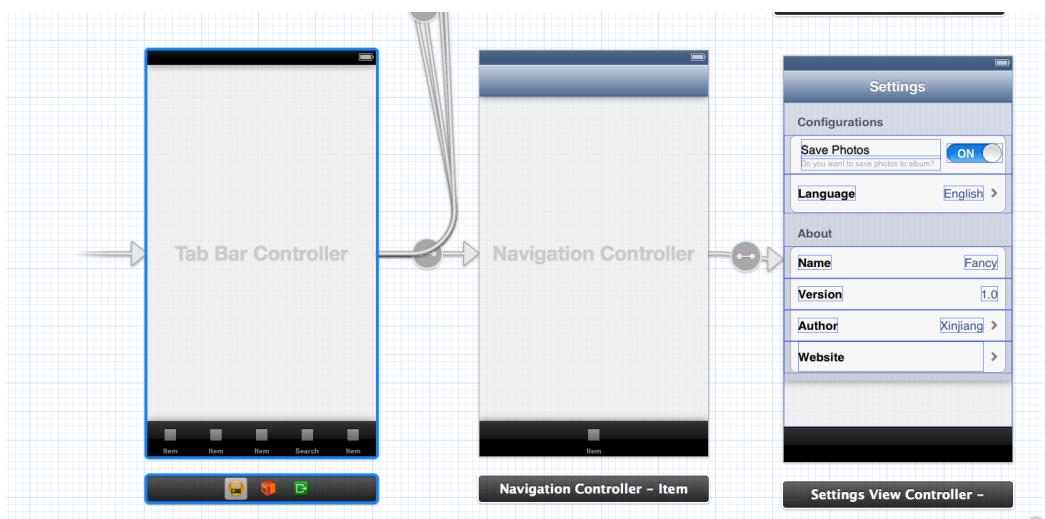


Figure 3.6: Storyboard of Settings View

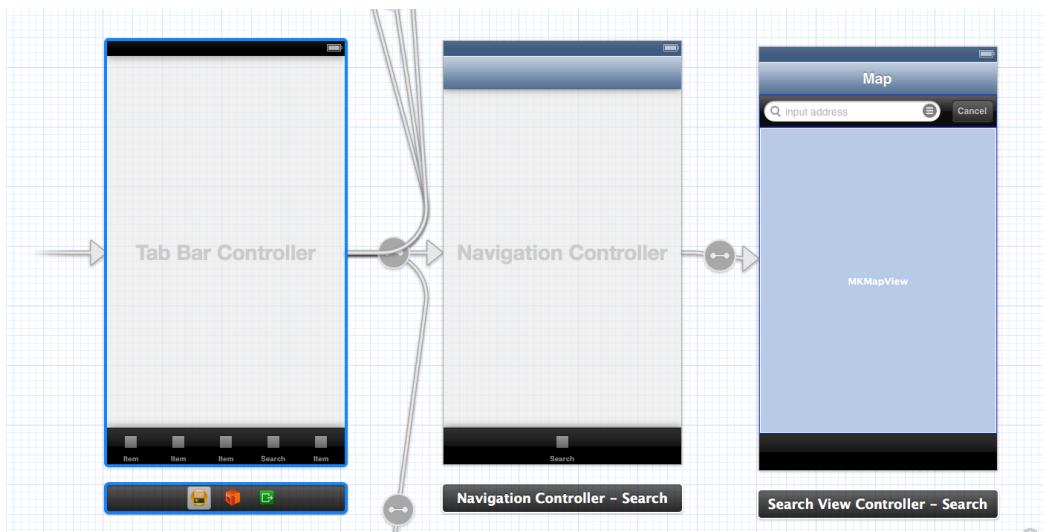


Figure 3.7: Storyboard of Map View

Chapter 4

Implementation

4.1 Data Model

4.1.1 Database Schema

In this project, we used three table for storing all the information from users. As Figure 4.1 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.

4.1.2 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.

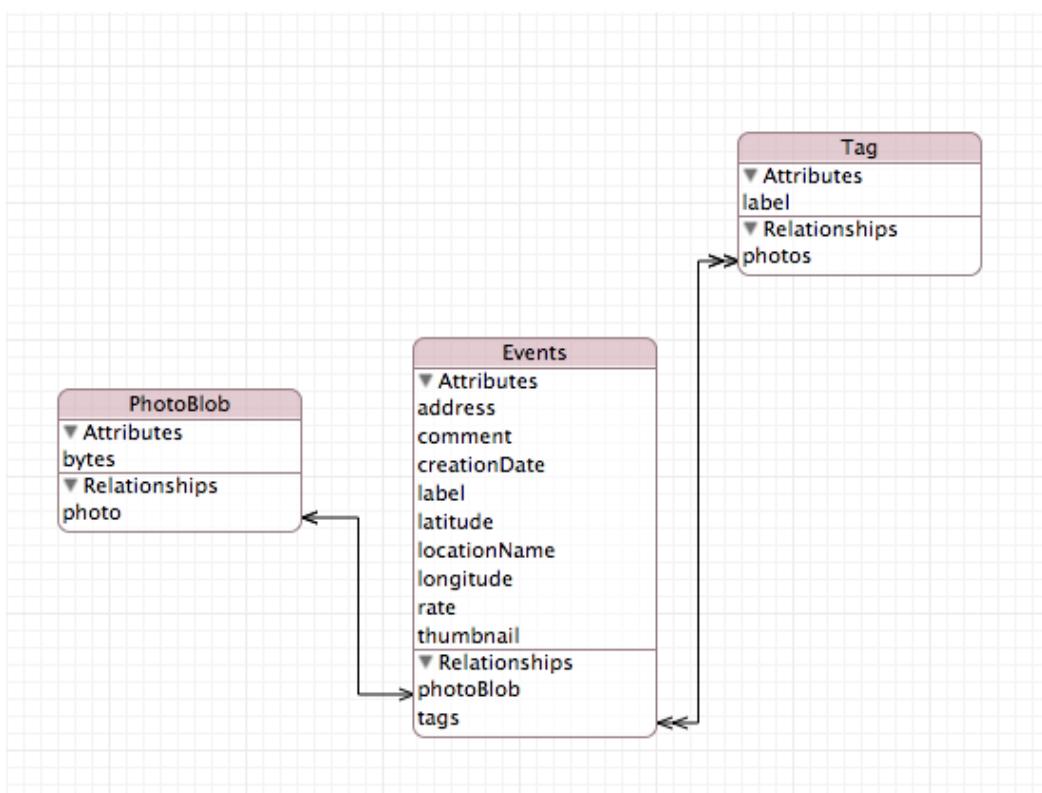


Figure 4.1: Database Schema

“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 4.6 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 4.6(b).

4.2 Workflow

4.2.1 Insert New Event

Figure ?? shows the

4.2.2 Update Event

4.2.3 Delete Event

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

4.2.4 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 4.5(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.

4.3 User Interface Design

4.3.1 Home Tab

As shown in Figure 4.2, home tab is created for adding new event to the app. The default view for user entering the app is Figure 4.2(a).

Figure 4.2(a) gives a short tutorial for user. If the user chooses the plus sign in navigation bar, another blank view should show up. After tapping on the camera icon, Figure 4.2(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 4.2(c) should appear for user to move and scale the photo to a right position and size. Figure 4.2(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 4.2(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 4.2(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.

4.3.2 Food List Tab

In food list tab, it fetches a list of food events ordered by creation date. Figure 4.3(a) shows some events. On each table cell, it shows a thumbnail, time string and string of the event. And a green menu pop up button is created for sharing with social media(Figure 4.3(e)) and updating comment(Figure 4.3(c)) or rate(Figure 4.3(d)).

After choosing one event, a detail view of the event will appear as shown in Figure 4.3(f) and Figure 4.3(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 4.3(h) will appear. The comment and photo is generated by the app. Simply tap on send, and the event will be shared.

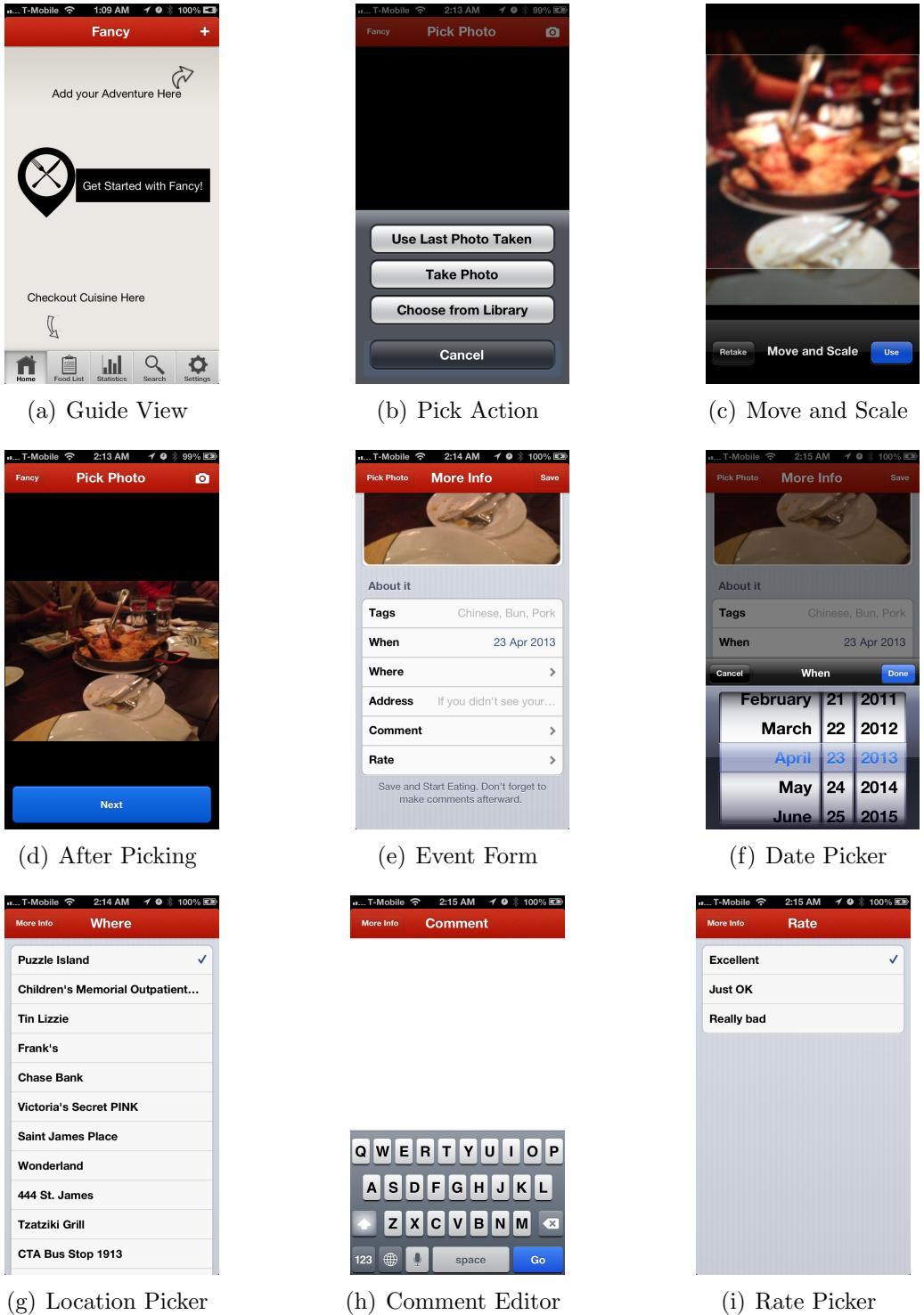
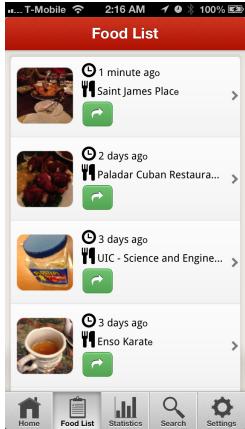


Figure 4.2: Home Tab View



(a) Food List



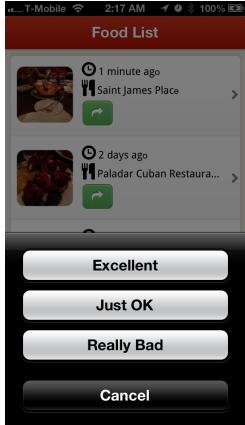
(b) Menu Popup



好吃



(c) Comment Editor



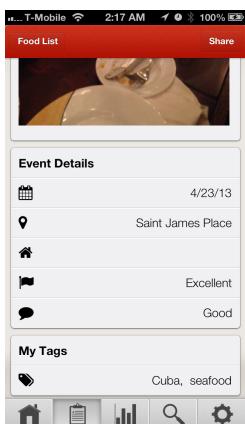
(d) Rate Action Sheet



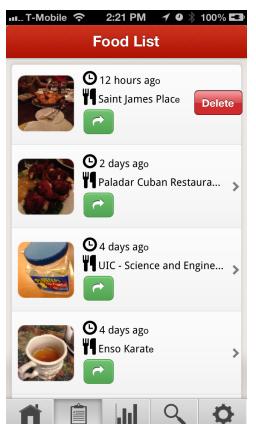
(e) Sharing Activity Controller



(f) Detail View

(g) Detail View
Cont'd

(h) Twitter



(i) Delete View

Figure 4.3: Foodie List Tab View

4.3.3 Stats Tab

Statistics tab is supposed to give the user an overview of all events he created. As shown in Figure 4.4, “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.

4.3.4 Map Tab

In Figure 4.5(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 4.5(b). After choosing a location in the list, the tableview will be dismissed and the new central region will be the location you chose.

4.3.5 Setting Tab

Setting tab provides a view to setup configurations of the app (Figure 4.6). 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 for each session, and even ask questions to users after a checkpoint is reached.

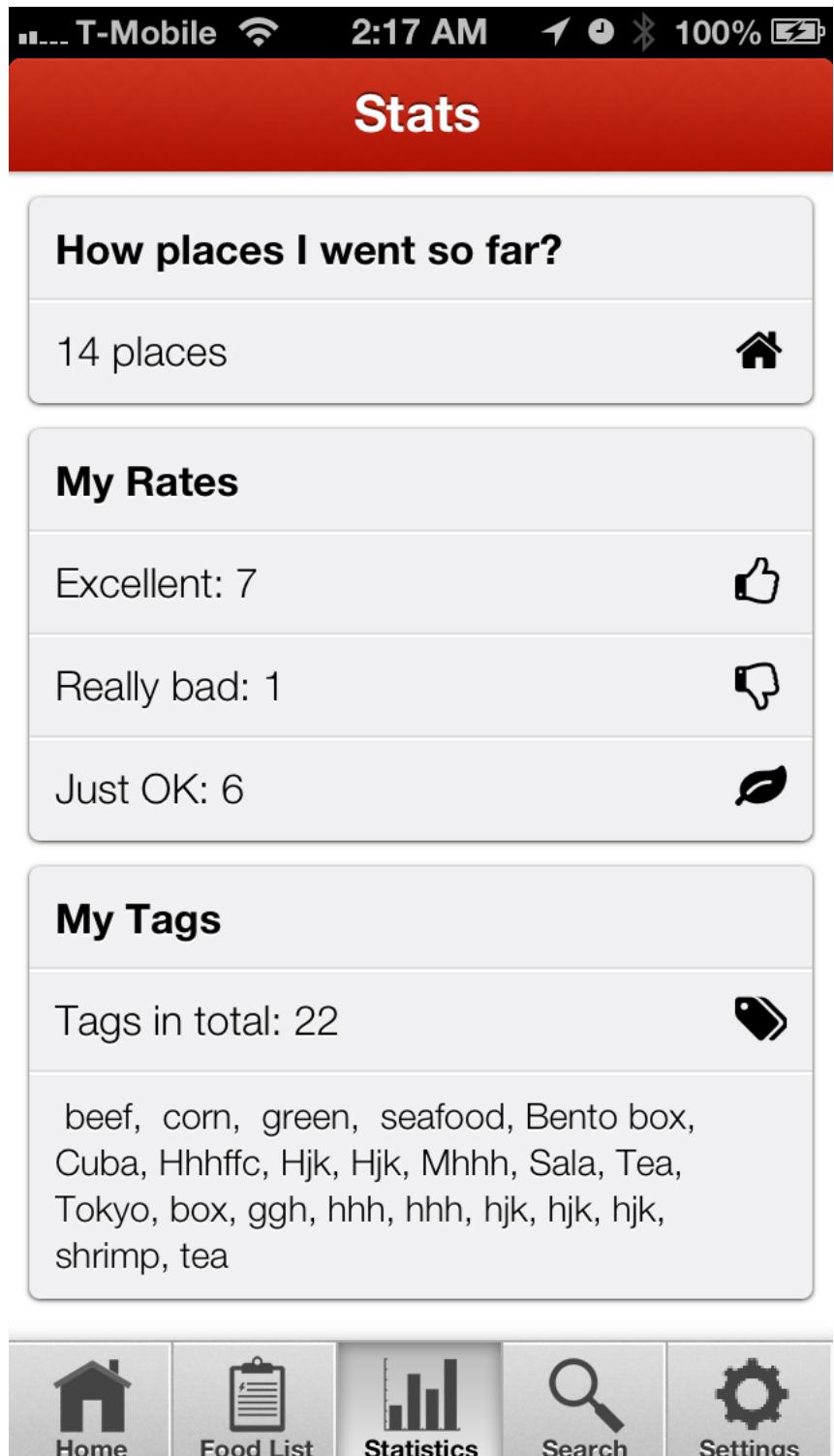


Figure 4.4: Statistics Tab View

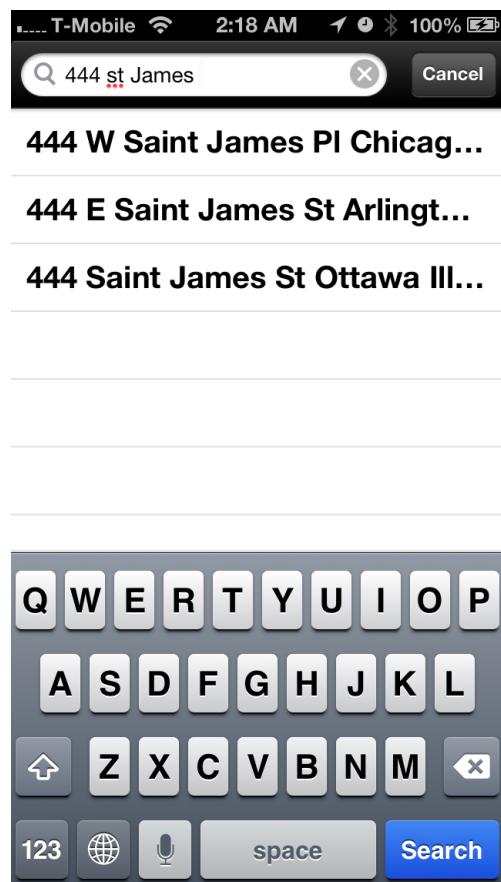
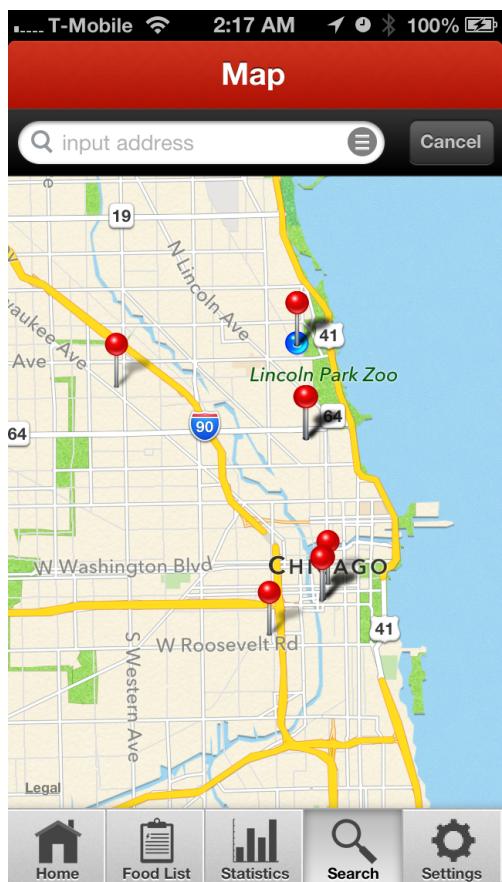


Figure 4.5: Map Tab View



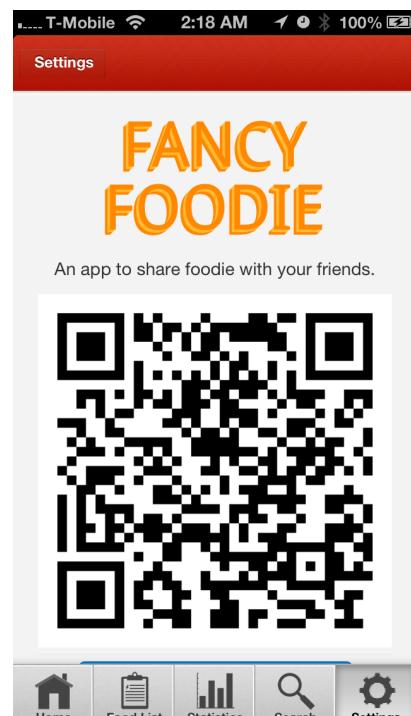
(a) Settings



(b) Save to Album



(c) Author Web-view



(d) Website View

Figure 4.6: Setting Tab View
21

Chapter 5

Future Work

Yelp Intergration
Centeral Database
Multi-languages
beta Test

Chapter 6

Conclusion

Conclusion here

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] ¡Name of the reference here¿, <urlhere>
- [2] ¡Name of the reference here¿, <urlhere>