Milestone 0: Describe the problem that your application solves.

There are many times when people urgently need a little help from others nearby, such as borrowing a hand phone charger, drawing crowds for events, pushing a heavy cart for Rag & Flag uphill and other dire situations where physical presence of the helper is valued. Whilst there are always people willing to lend a helping hand, it may take some effort and courage to find the right person to help you, which is especially a problem for introverts.

Milestone 1: Describe your application and explain how you intend to exploit the characteristics of mobile cloud computing to achieve your application's objectives, i.e. why does it make most sense to implement your application as a mobile cloud application?

Description:

1. **Hi NUS** application allows users to see their current location and interact with other nearby users inside NUS.

2. Users can choose to talk to a particular person nearby or chat with a nearby crowd.

Exploit characteristics of mobile cloud computing:

1. Mobile cloud provides enhanced service availability; this characteristics enable our users to access our application without limitation of time and location as long as they have an Internet connection.

2. By exploiting information about a user's location and context, we are able to inform users about their specific location in NUS and hence push them information about the nearby context and nearby users, thereby considerably personalizing and improving user experience.

Milestone 2: Describe your target users. Explain how you plan to promote your application to attract your target users.

Target users:

In the short term, Hi NUS will be targeting people who are physically inside NUS campus, mostly students and faculty staff members.

Currently NUS has 26,742 part-time and full-time students. Including 7,142 staff members, our potential initial market size is 33,884.

In the long term, if our app is able to facilitate geo-location communication well enough in NUS, expansion to other institutions, schools and business areas will be considered

Promoting Strategy:

1. **Collaborate with Student Organizations**

   Student organizations such as NUSSU strive to improve students' interest and welfare in NUS.

   NUSSU is the main student warfare organization. Collaborating with NUSSU will be a win-win strategy for both parties. **Hi NUS** is able to help NUSSU to achieve their goal of promoting students' interest and welfare by collecting student voices, whilst **Hi NUS** gains popularity and awareness amongst NUS students by serving as NUSSU's platform to broadcast and receive immediate feedback on the many events they host. Whilst participating in an event, **Hi NUS would be the audiences' best platform of choice for sharing their views on the event with the nearby NUS community concurrently as well as drawing their nearby friends to join in.**

2. **Broadcasting on IVLE**

   Posting blogs on the IVLE home page can act as an advertisement to access to teaching staffs and students, as IVLE is a main academic website that every student and teaching staff needs to login everyday. This will increase the rate of target market penetration

3. **Demonstration**

   Other than just simply raise awareness of **Hi NUS,** demonstration will be necessary to show the how **Hi NUS** can facilitate mass communication and hence benefit target users. These demonstrations shall be done in some hot locations such as lecture rooms, engineering canteen and etc to get attention from a mass crowd. At the same time, potential users will be educated of how to use this app and we also can get direct feedback from users.

4. **Flash Mob dance**

   Why flash mob dances? This will be one of the most fun ways to surprise nearby people in stressful NUS campus. Previous flash mob dances in arts canteen and business canteens have successfully attracted crowd and discussions. This will be an unconventional way for us to present **Hi NUS** to our target users.

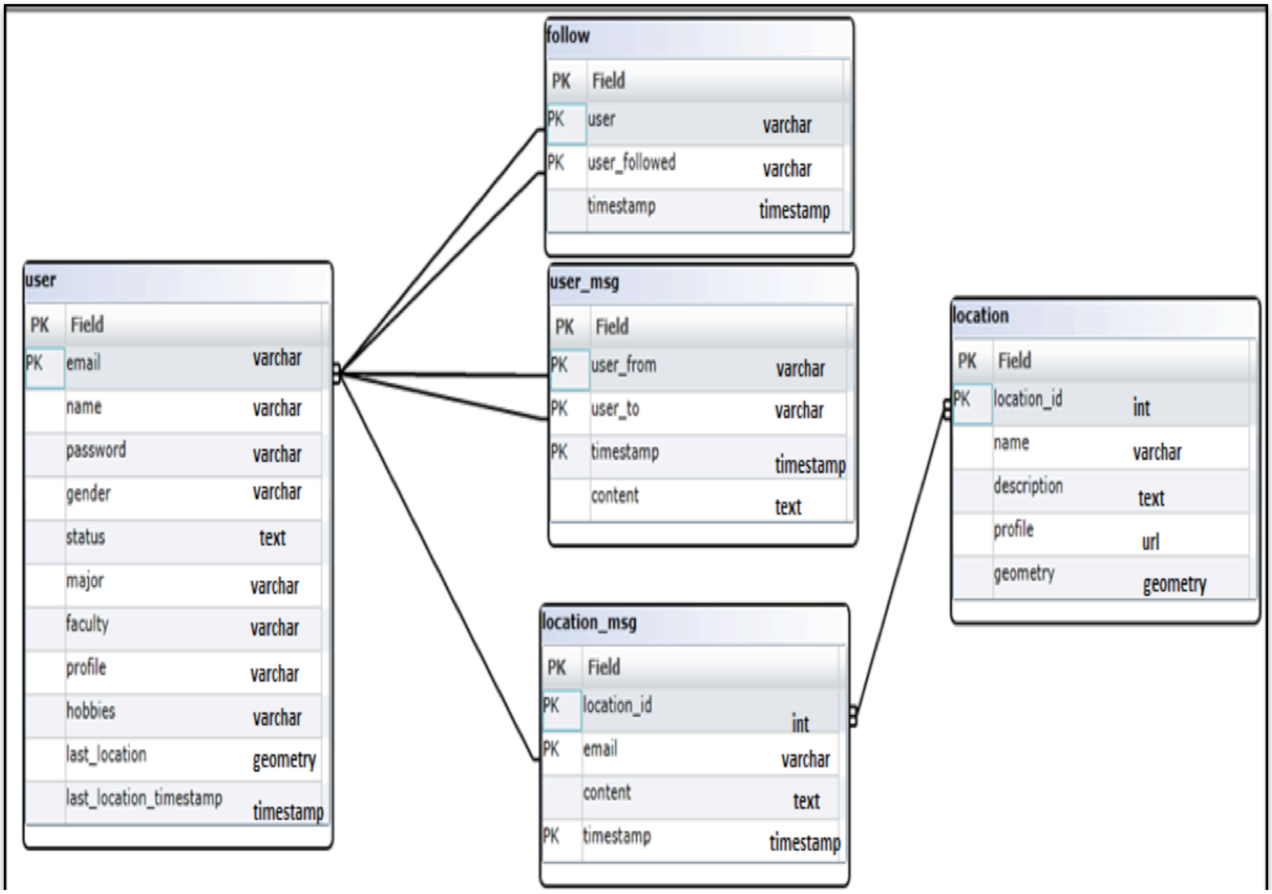
5. **Market through Facebook**

   Almost every NUS student and staff member has a Facebook account, and their Facebook connections usually consist of a large percentage of other NUS students. Connecting one NUS user can easily allow one to reach hundreds of other NUS users, and eventually Kevin Bacon (the idea of six degrees of separation). However, due to time constraints, **Hi NUS** will have Facebook login only in future.

Milestone 3: Pick a name for your mobile cloud application.

Hi NUS

Milestone 4: Draw the database schema of your application



follow

PK	Field	
PK	user	varchar
PK	user_followed	varchar
	timestamp	timestamp

user_msg

PK	Field	
PK	user_from	varchar
PK	user_to	varchar
PK	timestamp	timestamp
	content	text

user

PK	Field	
PK	email	varchar
	name	varchar
	password	varchar
	gender	varchar
	status	text
	major	varchar
	faculty	varchar
	profile	varchar
	hobbies	varchar
	last_location	geometry
	last_location_timestamp	timestamp

location

PK	Field	
PK	location_id	int
	name	varchar
	description	text
	profile	url
	geometry	geometry

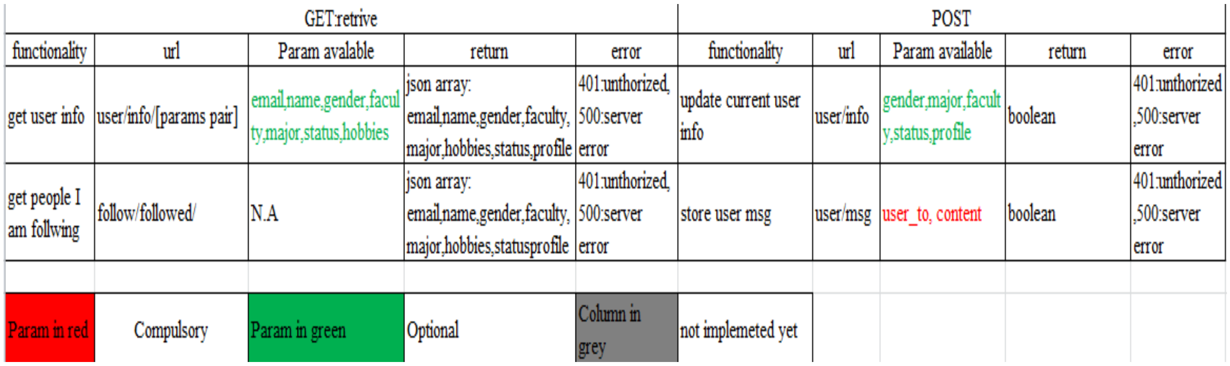
location_msg

PK	Field	
PK	location_id	int
PK	email	varchar
	content	text
PK	timestamp	timestamp

Milestone 5: Design and document (at least 3) most prominent requests of your REST API. The documentation should describe the requests in terms of the triplet mentioned above. Do provide us with a brief explanation on the purpose of each request for reference. Also, explain how your API conforms to the REST principles and why you have chosen to ignore certain practices (if any).

The diagram below displays some of part of our REST server APIs. It describes 4 REST API in total, and their respective triplets. For example, to get user information, you can call URL ‘user/info’ with ‘GET’ appended with parameter pairs, ‘key/value’ to narrow the query. The keys can be anything in the ‘Param available’ box. The return value will be a JSON array containing email, name etc. There would be 2 possible errors for this request. One is ‘401 un Authorized’ and ‘500 Server error’. Similarly, to get people I followed, Call ‘follow/followed’ with GET request. No parameter is needed. To update the user info, client needs to call ‘user/info’ with ‘POST’ method.

The full document can be found on Git repository named ‘REST API Guide’ which is with more detail.



GET:retrive					POST				
functionality	url	Param avalable	return	error	functionality	url	Param available	return	error
get user info	user/info/[params pair]	email,name,gender,faculty,major,status,hobbies	json array: email,name,gender,faculty, major,hobbies,status,profile	401:unthorized, 500:server error	update current user info	user/info	gender,major,faculty,status,profile	boolean	401:unthorized ,500:server error
get people I am follwing	follow/followed/	N.A	json array: email,name,gender,faculty, major,hobbies,statusprofile	401:unthorized, 500:server error	store user msg	user/msg	user_to, content	boolean	401:unthorized ,500:server error
🟥 Param in red	Compulsory	🟩 Param in green	Optional	▨ Column in grey	not implemeted yet				

Our REST API follows reasonably close to the principles of REST.

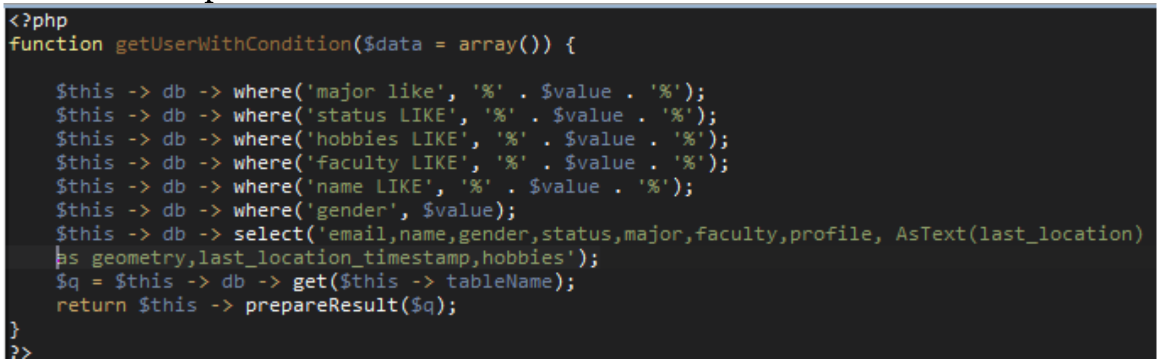
First of all, the request URL is clean as in no ‘&’ or ‘?’ are used for GET. Then, all the resources needed to process the request are retrieved via calling URL or data comes with POST, DELETE or PUT. Secondly, interface separates clients from servers. Clients are not concerned with data storage. Moreover, the purpose of each API matches the HTTP request method. For example, GET will only handle retrieving of data; DELETE methods are related removing of data. However, you may notice we do not have PUT API. The reason that we did not differentiate much between PUT and POST as they are quite similar and POST data is easier to handle.

Milestone 6: Tell us some of the more interesting queries (at least 3) in your application that requires database access. Provide the actual SQL queries you used.

1. GET: user/info/[param pairs]

This is the REST API used to retrieve user data. Client can provide 'key/value' pairs to reduce the result. For example, calling 'user/info/name/tan' will only return people who has name similar to 'Tan'.

Below is a simplified version of DB select code for this method.



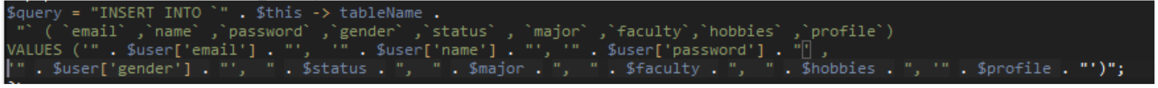
```
<?php
function getUserWithCondition($data = array()) {

    $this -> db -> where('major like', '%' . $value . '%');
    $this -> db -> where('status LIKE', '%' . $value . '%');
    $this -> db -> where('hobbies LIKE', '%' . $value . '%');
    $this -> db -> where('faculty LIKE', '%' . $value . '%');
    $this -> db -> where('name LIKE', '%' . $value . '%');
    $this -> db -> where('gender', $value);
    $this -> db -> select('email,name,gender,status,major,faculty,profile, AsText(last_location)
    as geometry,last_location_timestamp,hobbies');
    $q = $this -> db -> get($this -> tableName);
    return $this -> prepareResult($q);
}
?>
```

2. POST : user/info

This API is for Client to call to update user's information. Client provides some optional parameters in POST data section.

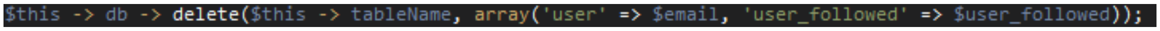
Below is the SQL query code



```
$query = "INSERT INTO `" . $this -> tableName .
 "` ( `email` ,`name` ,`password` ,`gender` ,`status` , `major` ,`faculty`,`hobbies` ,`profile`)
VALUES ('" . $user['email'] . "',  '" . $user['name'] . "', '" . $user['password'] . "' ,
'" . $user['gender'] . "',  " . $status . ",  " . $major . ",  " . $faculty . ",  " . $hobbies . ", '" . $profile . "')";
```

3. DELETE : follow/followed

Calling this API will end up remove an entry from 'follow' table. For users, it means 'un-following' someone.



```
$this -> db -> delete($this -> tableName, array('user' => $email, 'user_followed' => $user_followed));
```

Milestone 7: Find out and explain what [QSA,L] means. Tell us about your most interesting rewrite rule.

[QSA]

When the replacement URI contains a query string, the default behavior of Rewrite Rule is to discard the existing query string, and replace it with the newly generated one. Using the [QSA] flag causes the query strings to be combined.

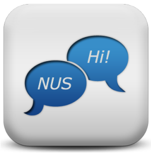
The [L] flag causes mod_rewrite to stop processing the rule set. Which means if the rule matches, no further rules will be processed

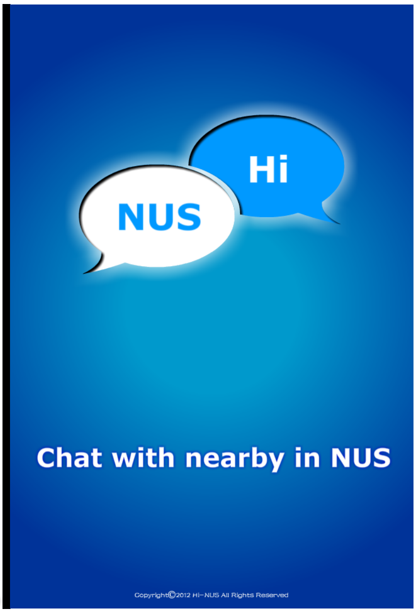
```
RewriteEngine on
RewriteBase /

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?/$1 [L]
```

Our rewrite rule will remove “index.php” from URL, and keep the others same.

Milestone 8: Create an attractive icon and splash screen for your application. Try adding your application to the home screen to make sure that they are working properly.





Milestone 9: Style different UI components within the application using CSS in a structured way (i.e. marks will be deducted if you submit messy code). Explain why your UI design is the best possible UI for your application.

Navigating between any two panes will require no more than two clicks in Hi NUS. This is because the user will frequently navigate between the map and the chat room. In order to provide convenience of switching panes, we provided buttons at the edge of the screen where it can be easily reachable with the thumbs on a touch screen device. Our application is easily accessible on touchscreens as well as traditional devices. We decided to put content before chrome in order to optimize usage of screen on smaller mobile device. There are screen transitions provided to guide the user along the way in the simplest manner.

Milestone 10: Implement and explain briefly the offline functionality of your application. Make sure that you are able to run and use the application from the home screen without any in- ternet connection. State if you have used Web Storage, Web SQL Database or both in your application. Explain your choice.

Even there is no Internet access, our app allows users to browse and search of latest static data in the application, which includes the user's friend list, historical conversation, friends' profile data and the map data.

Since data has been well organized in Web SQL Database, we used it to store all the data need to be browsed.

Milestone 11: Explain how you will keep your client in sync with the server. Elaborate on the cases you have taken into consideration and how it will be handled.

When user has Internet access, he is able to view the latest information pulled from the server. When the user is offline, the past data can still be seen, because the information is cached locally and made available offline. When the user reconnets to the internet, the local data is synchronized with the server so that the information is updated to the latest version in the local storage.

For example, when the user is chatting online, chatting data is stored locally. Every time when users view the logs, our app will pull the latest logs data and store to local storage. So if app becomes offline, user can view the data when they view last time.

Milestone 12: Compare the pros and cons of basic access authentication to other schemes such as using cookies, digest access authentication or even OAuth. Justify your choice of authentication scheme.

Basic access has obvious security flows. The authentication is attached in the header and is sent over secure networks. If the data is sent over an insecure network, malicious hackers are able to detect the credentials using sniffing tools. Digest access authentication does not send credentials directly. Instead it sends a challenge to users, and then compares the user-computed hash with its own computed hash. However, HTTP Basic authentication implementation is the easiest methods comparing to cookies, digest or OAuth. Comparing to cookies or Session, no confidential data is stored, which is more RESTful. Comparing to digest, it generates less HTTP request over network since no handshake, and hence is faster.

We choose the basic authentication together with HTML local storage as the solution. We use this scheme because of its simplicity and high speed. Hi-NUS is real time chatting application developed over wireless work, therefore speed is important. We append the hash of user credential to every REST request. HTML local storage is used to cache the credential so users do not need to type them every time. Also it overcomes the weakness of using Cookies which requires network connection.

Milestone 13: Describe 1-3 user interactions within the application and explain why those interactions help make it better.

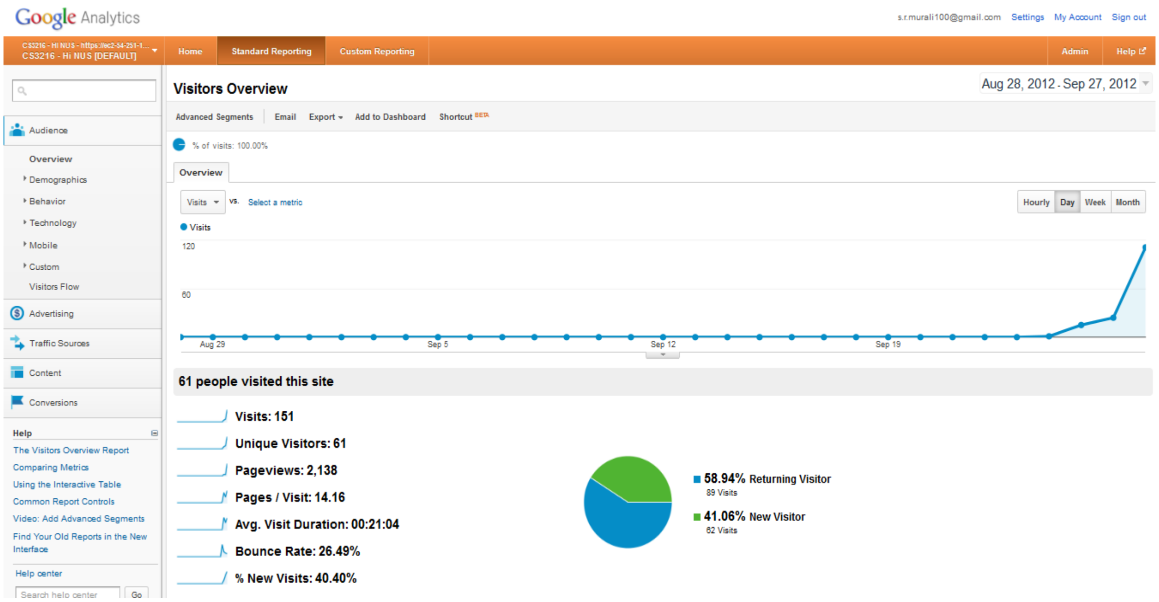
- If user A wants to follow user B, user A only needs to tap the icon of user B on map. Quick information about user B pops over next to the user B's icon on the map. If user A wants to view user B's full profile, he/she can just simply tap on the user B's profile photo in the information box. This interaction is simpler compared to selecting a person from a list.

- Once the users login, they can see their current location and nearby users directly on the map without a hassle to refresh or search for nearby users .

- When in a chat room, it is obviously a necessity to know who will be the recipients of your chat message. A list of nearby users is present in the "People" option. This list automatically gets out of the way when the user is on a smaller device, or when he focuses on something else. This is more user friendly than showing a list all the time - which may waste screen space, or distract the users.

-

Milestone 14: Embed Google Analytics in your application and give us a screenshot of the report. Make sure you embed the tracker at least 48 hours before submission deadline as updates are reported once per day.



Visitors Overview

Aug 28, 2012 - Sep 27, 2012

61 people visited this site

Visits: 151

Unique Visitors: 61

Pageviews: 2,138

Pages / Visit: 14.16

Avg. Visit Duration: 00:21:04

Bounce Rate: 26.49%

% New Visits: 40.40%

58.94% Returning Visitor
89 Visits

41.06% New Visitor
62 Visits

As the app is not launched yet, the Google Analytis does not show very fantastic statistics.

Milestone 15: Identify and integrate any social network(s) with users belonging to your target group. State the social plugins you have used. Explain your choice of social network(s) and plugins. (Optional) (facebook All)

Due to time constrain, we did not integrate any other social network plugins. However, our app itself is a small social network platform within NUS which is more geography based. Users within the same location can chat with each other and follow the other users. If we are given more time, we would like to integrate Facebook as our target users are young people who are active on facebook.

Milestone 16: Make use of the Geolocation API in your application. Plot it with Bing or Google Maps or even draw out possible routes for the convenience of your user. (Optional) (Han Yu)

Our Application fully utilizes the Geolocation API. First clients get their location data using HTML5 Geolocation API. The application will upload all the users' location data to the server and return them other users' data. Then all users within the same location will be plotted on the Google map. Moreover, we highlight location area on Google map so that it is able to detect and inform the users which part of NUS they are in like School of Computing, PGPP and etc. In order to build this function, we wrote our own algorithm to check which area the users are located. The areas are highlighted in polygons according to geographical shape of the areas. Further more, we customized the marker and infomation window. It will show profile photos on the map and user can click them to view basic information about that user or click on the photo in info window to redirect to the profile page of that user.

Milestone 17: Justify your choice of framework/library by comparing it against others. Explain why the one you have chosen best fulfils your needs.

Back-end

For backend development, we choose Codeigniter(CI) over the framework instead of Zen Framework, or Symfony. Considering the nature of this project and the context of our application, we need a framework that is easy to pick up, simple in terms of structure and easy to be implemented as REST. CI really fits in well.

CI is really easy to get started and has great documentation, which means its learning curve is less steep compared to Zen Framework and Sysmfony. Zen has a lot of libraries which are powerful yet complex. Sysmfony is too big scale for small project like ours.

CI has a clear MVC structure. When it comes to deploying an MVC project in as less as under one minute, the CI stand ahead since all you need to do with it is configure the respective files and next viola. That should settle it! Compared to this, there is the painful routine task of having to create your personal bootstrap file and then there is the job of choosing an appropriate directory structure in Zen.

CI has great controller-method architecture which is easy for developing a RESTful server. It also has an active community. We didn't want to pick a framework that has few active contributors and CI seemed well respected and established. The cool thing of CI is that you can post your doubts to the forum. It is especially handy for beginners like us.

Front-end

- We chose jQuery Mobile for our Frontend design. jQuery Mobile is easy to use as Sencha Touch is completely Javascript. Our application takes place in JavaScript files, dealing with JavaScript classes. On the other hand, jQueryMobile is markup-driven. Sencha Touch is an extension of the Ext JS framework.
- Another reason is because jQuery Mobile is loosely based on the jQuery Framework, we believed it would be a more steady stepping stone than Sencha and jQTouch. jQuery

Mobile also comes with different *themes with various choices of buttons, headers and colors*.

- jQuery Mobile also provides an easy way of implementing page transitions so that the panels hide or show in a more native way. Also, jQuery Mobile implements the data attributes provided by the HTML5 markup as we gained more understanding of HTML5 by just using the framework.
- From what we see, which providing an easy to debug interface, jQuery Mobile seems to be lighter which, in addition to which it allows many single purpose "pages" within the same HTML document which enables quicker transitions and lesser trips to the server.