CS3216Assignment 3:
Mobile Cloud Application

Group 4

Inian Parameshwaran - A0077498H
Meng Kaizhi - A0077840Y
Liu Zenan - A0077897B
Ngo Hoang Gia - A0085188R

Milestone 0: *Describe the problem that your application solves.*

Have you ever worked late into the night, all alone and frustrated over the daunting task that you are embarking on? Would you doubt then if what you were doing is worthy, if you were insane pursuing those fleeting dreams? Would you wish that someone can tell you "It's alright. You are ok", rather than reading your teacher's email nagging you on the deadline?

Have you ever had a good friend, whom because of so many things in life, has drifted away from your life? You would no longer talk to him or her the way you used to. But you may chance upon something in life that reminds yourself so much of him or her. Would you message the person on Facebook and yet fear that the initial excitement would soon die off and your conversation would drone into an awkward "how's life" talk? Would you email the friend and yet felt the subtle message would be lost in the cold, bland look of the email? Would you just want something else to condense the emotions, the moment into one single package to be presented to your friend?

Have you ever lived far away from home? Would you just want to chronicle some moments of your life and send back to your mom? But have your mom ever told you that she did not know how to open the attachment that you sent her? That she forgot where the photos are when she just wants to browse through them?

The way people connect with each other has drastically changed over the years. Postal mails have been replaced by email, postcards have been replaced by Facebook update. It is arguable that instant messaging has partially replaced how people "talk" to each other.

No doubt the new modes of communication have brought about numerous merits. On the other hand, the anticipation, the human touch laid within a letter or a postcard have not been much taken care of.

From that inspiration, we aim to bring back the "love" that is missing in virtual messages. We want to create a separate space where users can keep and treasure the pieces of love they receive, as well as to send their messages to another person or share to the public in a nicely wrapped form.

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?*

Users use our application to send e-postcards to other people's email addresses. User also receive and keep the postcards sent by other people in our application. Users can also view a public gallery of postcards shared by other people.
The application is implemented as a mobile cloud application since the data, i.e. the postcards are handled on the server, so as to reduce the hassle caused for the users in managing the postcards. Furthermore, as the application aims to be used anywhere and anytime in order to capture the emotions and thoughts of the users, the application should be able to be used across different platforms and the data should be able to be accessible on multiple devices. That fits right in the strength of mobile cloud framework.

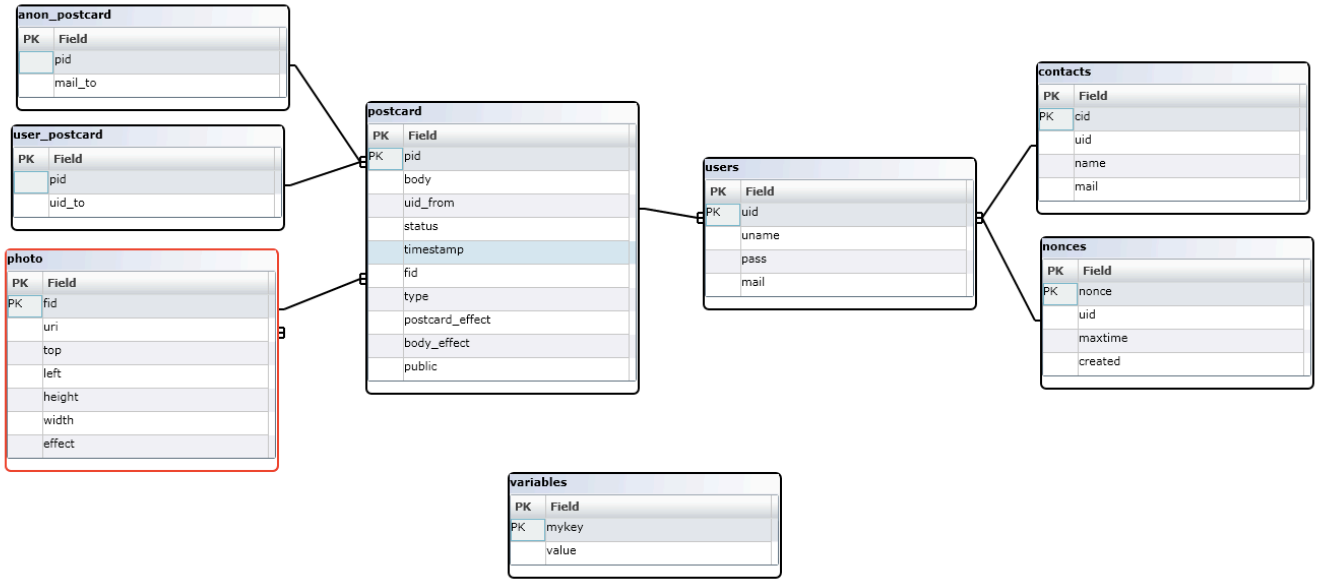
Milestone 2: *Describe your target users. Explain how you plan to promote your application to attract your target users.*

Young people who use and carry mobile devices around like Ipad, Iphone…
People who stay away from family.
Students who want to send post card to teachers.
Lonely people.
Couple.
Promotion Plan:
- A public page is meant for postcards shared by random users. This page is to arouse some interest in first-time users when they have not received many postcards yet.
- When a user sends a postcard to a non-user, an email is sent to the receiver, prompting the receiver to sign up for an account with us. This is to get new users try our application.

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

Love, me
The name is inspired by a song of the same name by Colin Raye. This is also an affectionate signature written at the end of a letter. The name conveys the purpose of the application as well as the sentiments that we try to enclose in the application.

Milestone 4: *Draw the database schema of your application.*



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 pur-pose 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).*

1.

Request Method + relative URL

GET user/uid

URL parameters - token

The token associated with the user with uid. This is the token that is got when the user logs into the application. This ensures that the unauthorised requests are not made to the REST server. This token is unique to every user and also has an expiry time.

Return Value

This is a GET request to retrieve all possible details of the user as a JSON object. This request also returns returns information about the user's postcards (categorised as read, unread, archived, drafts, sent and public), photos attached to the postcards and the contacts of the user. This makes sure that the client does not have to make repeated requests to the server for retrieving information about the user.

A sample request for user with uid 1 is shown below

GET http://54.251.37.19/api.php/user/1?token=1_5064a6f2bfb848.08495996


```
{
    uid: "1",
    uname: "Inian",
    mail: "inian1234@gmail.com",
    draft:
    [
        {
            pid: "9",
            body: "new postcard",
            uid_from: "1",
            status: "0",
            timestamp: "1347797550",
            type: "0",
            mail_to: "inifdjohn@doe.com"
        }
    ],
    sent:
    [
        {
            pid: "2",
            body: "this is a note2",
            uid_from: "1",
            status: "1",
            timestamp: "1347642255",
            type: "0",
            mail_to: "inian@yahoo.com"
        },
        {
            pid: "6",
            body: "haha",
            uid_from: "1",
            status: "1",
            timestamp: "1347736910",
            type: "1",
            uid_to:
            {
                uid: "2",
                uname: "John",
                mail: "john@doe.com"
            }
        },
        {
```

```
                    pid: "8",
                    body: "haha",
                    uid_from: "1",
                    status: "1",
                    timestamp: "1347737213",
                    type: "0",
                    mail_to: "inijohn@doe.com"
            }
],
archived:
[
            {
                    pid: "4",
                    body: "note4",
                    uid_from: "1",
                    status: "2",
                    timestamp: "1347734471",
                    type: "1",
                    uid_to:
                    {
                            uid: "2",
                            uname: "John",
                            mail: "john@doe.com"
                    }
            }
public:
        {

pid: "17",
body:
{
            content: "
            There are only two ways to live your life.
            One is as though nothing is a miracle.
            The other is as though everything is a miracle.",
            body_effect: "0"
},
uid_from: "28",
status: "1",
timestamp: "1348830899",
type: "0",
postcard_effect: "0",
public: "1",
photo:
```

```
        {
                **fid**: "16",
                **uri**: "image/16.png",
                **top**: "0",
                **left**: "0",
                **height**: "200",
                **width**: "300",
                **effect**: "0"
        }
        },
        ],
        **received**:
        [
                {
                        **pid**: "3",
                        **body**: "note3",
                        **uid_from**: "2",
                        **status**: "1",
                        **timestamp**: "1347734471",
                        **type**: "1",
                        **uid_to**:
                        {
                                **uid**: "1",
                                **uname**: "Inian",
                                **mail**: "inian1234@gmail.com"
                        }
                }
        ],
        **contacts**:
        [
                {
                        **cid**: "1",
                        **uid**: "1",
                        **name**: "Raj",
                        **mail**: "raj@gmail.com"
                }
        ]
}
```


This request returns a 401 Unauthorised Error if the token is not valid

2. **Request Method + Relative URL**

POST /user/

Parameters

uname- The user name of the new user
mail - The mail id of the new user
pass - The base64 encoded password of the user

Return

This request creates a new user based on the values provided. Since the email id of the user has to be unqiue, this request returns a 500 "Internal Server Error" if a email id that has already been taken is requested again. This response is taken care of by the client.
On success, this returns a json object with the uid and the access token of the newly created user which can be used to log the user into our application.
For example,
{"uid":"1","token":"2_5063cbfb5cafc9.20832393"} is a json object that could be returned on when a new user is successfully created.

3.**Request Method + Relative URL**

PUT /postcard/pid

Parameters

token - The access token of the person who is sending the postcard or recieving the postcard
uid_from - The user id of the person sending the postcard
top - The top coordinate of the photo in the postcard
left - The left coordinate of the photo in the postcard
width - The width of the picture
height - The height of the picture
photo_effect - The effect applied to the photo
body - The actual content of the postcard
body_effect - The styling applied to the body of the postcard
status - The status of the postcard (draft, unread, read, archived)
data_url - The data-url of the image to be saved
mail - The mail of the person to whom the postcard has to be sent to
public_card - A value to indicate if the postcard is public/private

Based on these values, all the corresponding values of the postcard are updated. Thus any modification to postcard can be achieved through this PUT request.

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. $query = mysql_query("SELECT * from user_postcard inner join postcard on user_postcard.pid = postcard.pid and user_postcard.uid_to = '$uid' and postcard.status =

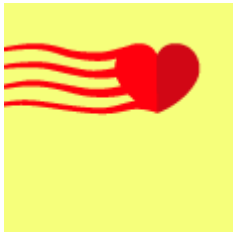
'$type' ORDER BY postcard.timestamp desc");
This query is used to find the details of a postcard of a particular user and a particular type (read, unread, archived, sent). This makes the query very flexible as well, giving you just what you need. It returns the postcards in reverse chronological order so that it can be easily displayed by the client.
The INNER JOIN is also used in this query. Since the information about the postcard is spread across two tables, INNER JOIN effeciently retrieves all the infomation about a given set of postcards.

2. while (!$query) {
    //check if a valid token exists and return that token, and increase the expiry time
    $created = time();
    $maxtime = $created + 3 * 24 * 60 * 60;
    $new_token = uniqid($uid . "_", true);
    $query = mysql_query("INSERT into nonces (nonce, uid, maxtime, created) VALUES ('$new_token', $uid, $maxtime, $created)");
  }
This code fragment makes interesting use of the mysql error mechanism in PHP. Since a unique token has to be returned to the client and it should not clash with other access tokens, we make the nonce field of the table as the primary key and try inserting the random token. If the token has already been taken, this will result in an error and will try inserting the token again with the help of a while loop.

3. $query = mysql_query("UPDATE photos set `uri` = '$uri',`top` = $top, `left` = $left, `height` = $height, `width` = $width, `effect` = $effect where `fid` = $fid");
This is a simple update query which updates the photos table based on the new values received. However the interesting part of the query is that the columns have to be escaped before passing it into the mysql_query function since left (the left coordinate of the photo) is a PHP keyword which can not be directly used in the query.

Milestone 7: *Find out and explain what [QSA,L] means. Tell us about your most interesting rewrite rule.*
QSA is a flag in the Apache mod_rewrite module. It expands to Query String append. The default behaviour of the RewriteRule is to usually discard the exisiting query string and replace it with the new query string. Using the QSA flags causes the query strings to be appended instead of being discarded.

Milestone 8: *Create an attractive icon and splash screen for your appli-cation. 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.*

The UI is kept clean and simple since we hope anyone can find the application easy to use.
The major focus should be on the postcard itself. So the majority of screen size is allocated to postcards.
In order to give a more native application feeling. We implemented touch move scrolling for the postcard.
We use responsive design for all pages in order to optimise for devices of all screen sizes
Navigation and postcard creating functions are intuitive and easy to use.

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 internet connection. State if you have used Web Storage, Web SQL Database or both in your application. Explain your choice.*

If a user cannot connect to the Internet after logging in, the user can only view the stored postcards. We decided not to allow postcard editing/uploading due to the possibility of having multiple users using the apps (security issues would occur if a new user logged in on the same device).

To store the postcard data, we have made use of Web Storage and cookie. Web Storage is for holding postcard information while cookie is used to hold more sensitive data like user email and hashed password (if "remember me" is checked). We chose to use web storage because what needs to be stored is just one list of data and there is no relational data to be worried about.

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.*

The client side will actively retrieve data from the server each time a navigation triggered. This makes sure that client would always possess the latest version of data. Additionally, all user action will trigger the client to send a request to server and to make corresponding changes on local data. This ensures that same changes would be applied on both client and server end. Lastly, when user uses the application offline, all actions that requires data modification would be disabled, making sure that no inconsistency would occur.

For example, once user is logged in, data of all postcards (inbox and sent) would be retrieved from server. Whenever user navigates to view the "sent" folder, another request would be sent to server to retrieve all data needed, making sure every page user is viewing contains the latest version of information.

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.*

We have used used a better form of Basic Authentication for our application. When the user logs in, he sends his emailid and password which is base64 encoded. This may not be the best

way to transmit the password given that the conncetion between the server and the client is not secure. Once the server recieves the password, it is immediately hashed (SHA-256) with a random salt. The salt is ensured to be random by appending random strings to the username, current time and hashing the resultant string. This password is then appended to this salt and hashed again before storing in the database. The random salt is also stored along with the password so that it can be verified when the user logins again. Storing the hash of the passwords ensures that the passwords of the user is not compromised even when the server is compromised.

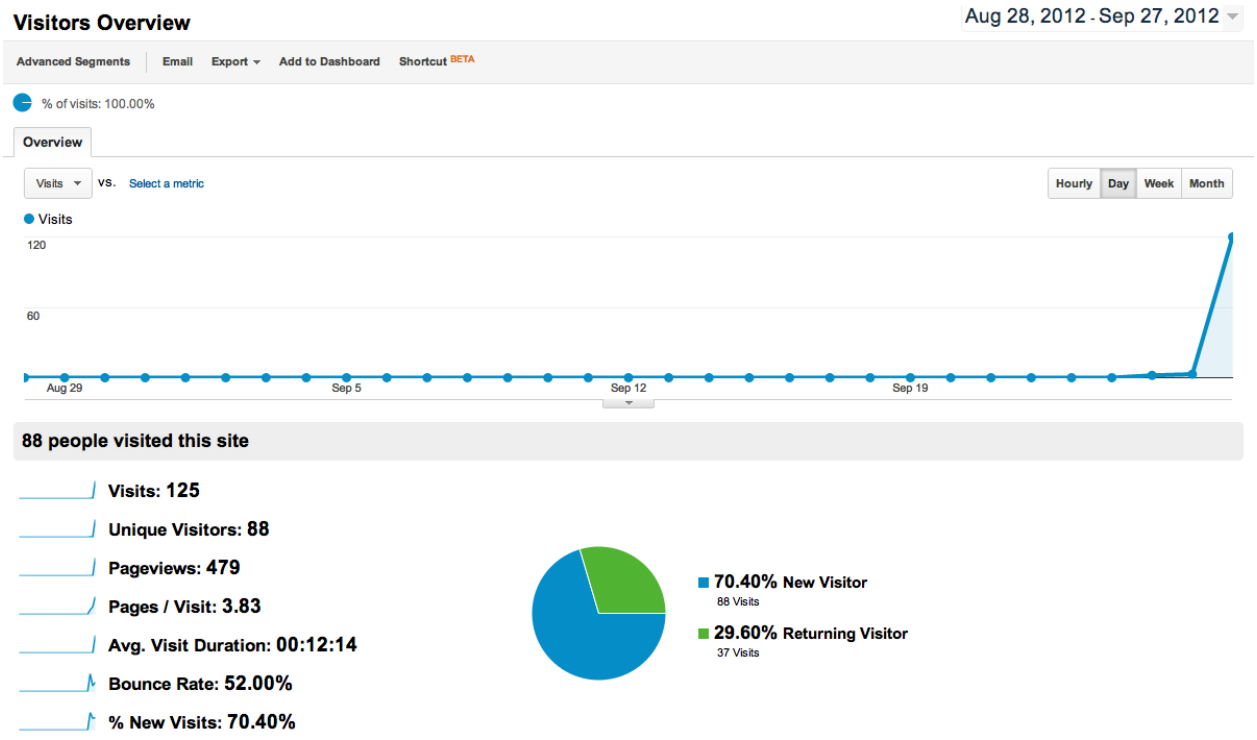
We decided not to use OAuth because at present we would not be offering our API to third parties. Therefore, we did not find the necessity to implement OAuth for our application.

Request to the REST API also required a access token, which is a pseudo-random number generated by a PHP function. These tokens are given to the user when he logs in and must be attached to each request that he makes to the REST server. These token have an expriry time which gets updated each time the user logs in. This also prevents replay attacks to some extent. Since each user gets a different random access token, even if the access token of one user is compromised, the safety of the other users is not compromised.

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

- Swiping gesture to navigate in a collection: since the application mainly aims at mobile device, the swiping gesture is the most intuitive way for users to navigate within a collection.
- Flipping of postcards when clicked or touched: this helps the user switch quickly between the image and the content of the postcard.

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.*



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

Our application makes use of email, which is the most popular and widely-spread electronic communication tool, to connect people. Not only would users log in using their email addresses, every postcard has an existing email account as its receiver's address. It is also used to connect existing users with non-users since non-users would receive an invitation email if a postcard is sent to them.

Moreover, the core idea behind "Love,Me" is to simulate real-life human interaction with rich emotion, which is the reason why we decided not to rely too heavily on any existing virtual social networking services.

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)*

We have incorporated Geolocation API in our application so that the current location of the user would be recorded in the postcard. We also planned to make use of Google Map API to provide more accurate and meaningful location information.

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

Backbone JS

Backbone.js gives structure to web applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions,views with declarative event handling, and connects it all to your existing API over a RESTful JSON interface. It works perfectly with RESTful server and also allows local storage in JSON format.

Slim Framework for Rest Server - Since the backend is a predominantly a REST server, we used a light-weight framework (as the name suggests!) called Slim to implement our backend. This framework gave a more structured and easy way for implementing the REST API. The powerful routing of requests by the framework was also very useful during development.