Urban Beats

Project Report

CIS 450/550 Database & Information Systems - Spring 2015

Team Members :

Aakriti Singla

Krishna Chaitanya Daliparthy

Parul Bhalla

Vamsi Jandhayala

http://urbanbeatsbeta.herokuapp.com/

Table of Contents

Abstract (the idea and motivation behind the application)

Below is the detailed description of the idea and key features of the Application that we have implemented as part of CIS 550 final project:

Motivation behind:

On a daily basis we come across the following situations:

- Clogged mailbox with pamphlets showing the deals on pizzas or any restaurants
- Every company (not just restaurants or coffee shops) have their custom websites and mobile apps to publicize their offers for customers. It becomes increasingly difficult for us to check all the websites to find the best deals
- Businesses like Yelp only give the business info to the user and help take a decision, the other way round?
- From a user perspective, there is a need for a good hangout planner

Enter UrbanBeats:

- Now we let the businesses around you know about a potential customer around them!
- This means more customized and dynamic deals for the users
- A neat way of organizing a hangout with friends or family
- No pain of visiting various sites to find offers of the day, let the businesses take the pain for a customer

Modules and Architecture

Broadly, speaking Urban Beats can be divided into two modules: The User Module, and The Business Module. Before actually going into the details of these modules, here is the brief outline of the control flow in our application:

The User can log in using his/her email-id and password. According to his choice, he can select one of the categories offered by our website: Restaurants, Food, NightLife, Shopping, Bars. We shortlisted these five categories from the Yelp Dataset, on the basis of the amount of information provided by Yelp on different categories. By analyzing the Yelp Dataset, we found out that these five are the most popular categories provided by Yelp. Once the user selects his category, he is displayed all the businesses related to that Category and located near his location that are open at that time. Thus this initial list of businesses is filtered on the basis of location,category, and time. After the initial filter, the user can further refine his research on the basis of other filters like: Ratings of Business, Ambience, Alcohol,Take Out, Delivery, Outdoor Seating, Parking etc. As soon as the user clicks on a particular checkbox the list of businesses

displayed on the page get refreshed. The user can also view the images of each of these businesses. On the same page, the user can also see all the premium flyers that are sent on the fly by the businesses who are registered as premium members with our website.Once he is satisfied with the results, he can proceed ahead to view the flyers/deals offered by these businesses, by clicking on the Notify All button. Moving ahead, now the user can see the deals offered by these different businesses. He can also check the reviews of these businesses, then he can make his choice and select one of the given deals. Once the user selects the flyer and the business, the user is shown the driving route from his current location to the location of the business. He is also sent the flyer coupon on his email. There is also an option to send the flyer coupon on SMS to the user, if the user wishes so.

The Business on the other hand can also log in to our website using their business-id. The businesses can see all the users who are currently logged into our website. The businesses are also given the opportunity to create new flyers. They can also view all of the flyers associated with that business till now, along with the success rate of that flyer. So with all this information provided and with the information of the active users, the business can decide which out of all the flyers associated with that business, will act as the active flyer. Each business will also have a Personal Information page, where it can see its name, address, ratings etc. The business is also given the facility to invalidate the flyer coupon. Once the user avails his flyer coupon, the business can invalidate that flyer coupon, so the same coupon cannot be used again.

The User Module

The User Module can be further divided into the following various sub modules:

- **Sign-up**: The User can register with our site using the sign-up option. On the sign-up page, the user needs to fill in details like First Name, Last Name, Email-id, Password, and three Security Questions. The Security Questions are an important feature since it will help the user retrieve the password in case the user forgets the password.
- **Login**: The user can login using his/her email-id and password. Once the user is authenticated, user is directed to a page where he/she can select the category.
- **Refine Businesses**: Once the user selects, the category from one amongst: Food, Nightlife, Shopping, Bars etc. he is directed to a page which displays the businesses associated with that category, near the user's location which are open at that time. Further the search can be refined on the basis of filters like Rating,Ambience, Take-Out, Delivery, Alcohol etc.
- **Select Flyer**: The user gets the list of all the flyer coupons provided by different businesses along with the reviews associated with that business. The user can select any one from the list.
- **Driving Route/Email**: Once the user finally makes up his/her mind on the business, his flyer coupon is sent to him in his mail, and also can be sent to him

through SMS. The user is also shown the driving route to his/her final destination.

- **Forgot Password**: In case the user forgets the password, the user can answer the security questions that he would have choosen at the time signup. If he answers the questions correctly, his/her password is sent to him/her on mail.

The Business Module

The Business Module can be divided into the following sub modules:

- **Login**: The business can login using its business-id and can view all the logged in users.
- **Prepare Flyers**: The business is provided an interface where it can prepare a new flyer. The new flyer will get added to the list of flyers associated with that business.
- **Success Rate of Flyers**: The business can also see the list of all the flyers associated with that business. It can also view the success rate of each flyer and select one of them to behave as an active flyer.
- **Invalidate Flyer Coupon**: Once the customer avails the offer provided by the business, the business can invalidate the flyer coupon, so that the user cannot avail the same flyer coupon again.
- **Personal Info Page**: Each business has their own personal information page, where it can view its name, address, rating, image etc.

Extra Credit Modules

- **Login with Facebook:** The user is allowed to login to our website by logging through facebook.
- **Google API:** We have used **Google Maps** to show the driving route from the user's current location to the business location. We have also used the Google Image Search API, to fetch the images of the different businesses that the user can view on the basis of the category, location, and time.
- **No-SQL Component:** We used **Reddis** as our No-SQL component, to store the reviews associated with all the businesses.
- **Invite Friends and Friend Groups**: Once the user logs in through the facebook, he/she can invite friends and later when he avails the flyer coupon, he can share his flyer coupon with his friends. He can also view all his logged in friends.

Architecture

We have made use of the following Relational Database schemas at different stages of our application. The following are the schemas that we used in our application.

- **Business**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
business_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
name	text		NO	latin1	latin1_swedish_ci	select,insert,update,references
full_address	varchar(400)		NO	latin1	latin1_swedish_ci	select,insert,update,references
city	varchar(40)		NO	latin1	latin1_swedish_ci	select,insert,update,references
state	varchar(40)		NO	latin1	latin1_swedish_ci	select,insert,update,references
stars	decimal(10,2)		NO			select,insert,update,references
review_count	int(11)		NO			select,insert,update,references
categories	varchar(2000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
open	int(11)		NO			select,insert,update,references
email	varchar(1000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
Date	datetime		YES			select,insert,update,references
is_premium	varchar(40)	no	YES	latin1	latin1_swedish_ci	select,insert,update,references
Longitute	float(30,20)	0.00000000000000...	YES			select,insert,update,references
Latitude	float(30,20)	0.00000000000000...	YES			select,insert,update,references

- Business_Attributes

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
business_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references	
price_range	int(100)		NO			select,insert,update,references	
take_out	tinyint(1)		NO			select,insert,update,references	
parking_lot	tinyint(1)		NO			select,insert,update,references	
parking_street	tinyint(1)		NO			select,insert,update,references	
parking_garage	tinyint(1)		NO			select,insert,update,references	
parking_valet	tinyint(1)		NO			select,insert,update,references	
delivery	tinyint(1)		NO			select,insert,update,references	
waiter_service	tinyint(1)		NO			select,insert,update,references	
drive_thru	tinyint(1)		NO			select,insert,update,references	
ambience_divey	tinyint(1)		NO			select,insert,update,references	
ambience_classy	tinyint(1)		NO			select,insert,update,references	
ambience_touristy	tinyint(1)		NO			select,insert,update,references	
ambience_hipster	tinyint(1)		NO			select,insert,update,references	
ambience_trendy	tinyint(1)		NO			select,insert,update,references	
ambience_intimate	tinyint(1)		NO			select,insert,update,references	
ambience_casual	tinyint(1)		NO			select,insert,update,references	
ambience_romance	tinyint(1)		NO			select,insert,update,references	
ambience_upscale	tinyint(1)		NO			select,insert,update,references	
outdoor_seating	tinyint(1)		NO			select,insert,update,references	
alcohol	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references	
byob	tinyint(1)		NO			select,insert,update,references	
byob_corkage	tinyint(1)		NO			select,insert,update,references	

- **Categories**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
category_id	int(11)		NO			select,insert,update,references
category_name	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references

- **Business_Categories**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
business_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
category_id	int(11)	0	NO			select,insert,update,references

- **Business_Hours**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
business_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
day	enum('Monday','Tue...		NO	latin1	latin1_swedish_ci	select,insert,update,references
open	varchar(100)		YES	latin1	latin1_swedish_ci	select,insert,update,references
close	varchar(100)		YES	latin1	latin1_swedish_ci	select,insert,update,references

- **Flyer**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
flyer_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
business_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
flyer_coupon	varchar(6000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
is_accepted	varchar(40)	no	NO	latin1	latin1_swedish_ci	select,insert,update,references
No_of_views	int(11)		YES			select,insert,update,references
No_of_selects	int(11)		YES			select,insert,update,references
date_modified	date		YES			select,insert,update,references

- **Friends**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
login_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
friend_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references

- **User_Flyer**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
coupon_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
flyer_id	varchar(100)		YES	latin1	latin1_swedish_ci	select,insert,update,references
user_id	varchar(100)		YES	latin1	latin1_swedish_ci	select,insert,update,references
is_valid	varchar(10)	yes	YES	latin1	latin1_swedish_ci	select,insert,update,references

- **User**

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
user_id	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
review_count	int(11)		NO			select,insert,update,references
member_since	varchar(100)		NO	latin1	latin1_swedish_ci	select,insert,update,references
password	varchar(100)		YES	latin1	latin1_swedish_ci	select,insert,update,references
first_name	varchar(1000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
last_name	varchar(1000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
Sec_Ques1	varchar(5000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
Sec_Ques2	varchar(5000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
Sec_Ques3	varchar(5000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
Sec_Ans1	varchar(1000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
Sec_Ans2	varchar(1000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
Sec_Ans3	varchar(1000)		YES	latin1	latin1_swedish_ci	select,insert,update,references
is_loggedIn	int(11)	0	YES			select,insert,update,references

Data Instance Use

We have extensively used the Yelp Dataset provided to us throughout our application. Database acts as a backbone for our application. As shown above, we have used different schemas according to the requirement of the application. For instance as soon as any User registers with our website, the user's information like the user's first name, last name, email-id, password, security questions are added to my User table. Next time when the user has to log in to the website, his credentials are authenticated from the database. Then to display the business located in a particular location, we refer to the Business table. To filter the table on the basis of category, rating, other business attributes like Ambience, Take-out, Delivery, Alcohol, Outdoor Seating etc, we refer to the Business_attributes table and Business table. We also refer to the Business_hours table to find out which all businesses are open at a particular time. Furthermore, once the user refines his search of the businesses, he is directed to a page that displays all the flyers associated with the given business. To do that, we make use of the Flyer table. In addition, we have stored all the friends of a particular user in the friends table. So if the user wants to share with his friends the flyer coupon that he has availed, we will send an email to his friends, by retrieving the user's friend's email-id from the Friends table. We have also used a User_Flyer table, where we have kept record of the flyer coupons availed by a particular user.

On the Business side as well, the information about the existing and new businesses are stored in the Business table. The business can create new flyers, which get stored

in the Flyer table. The business can also view the success rate of its flyers which can give them an idea as to whether to go ahead with the same flyer or not.

Thus at every stage our application interacts with the database extensively.

Data Cleaning and Import Mechanism

We used PYTHON scripts to transfer data line-wise from JSON format to MYSQL Database created in XAMPP server on local machines. After proper clean up the data was moved on AWS instances. For data clean up, we handled dirty data like removing NULL, empty values or values which were not consistent in various related tables. We created various integrity constraints like Primary (simple and composite) Keys, Foreign Keys to keep data compatible across various tables. We added Unique, Default constraints, BTree and Hash Indexes wherever necessary. Like for Business Hours where the comparisons were like >= / <= we implemented BTree indexes where as for exact matches like current location, current day etc Hash Indexes were used.

Also, we ensured to add only the data related to Top 5 categories in YELP database as those were the only required businesses.

Algorithms and Communication Protocols

We have used algorithm to determine the success rate of the flyers generated by the Business.

For every flyer associated with the business

find the no_of_views associated with that flyer

find the no_of_selects associated with that flyer

success_rate = No_of_selects/No_of_views

Communication Protocols : We used various different modules for communication purposes like for instance we used ‘Node-mailer’ to send the emails, ‘Passport’ for facebook login, and ‘twilio’ to send SMS to the users.

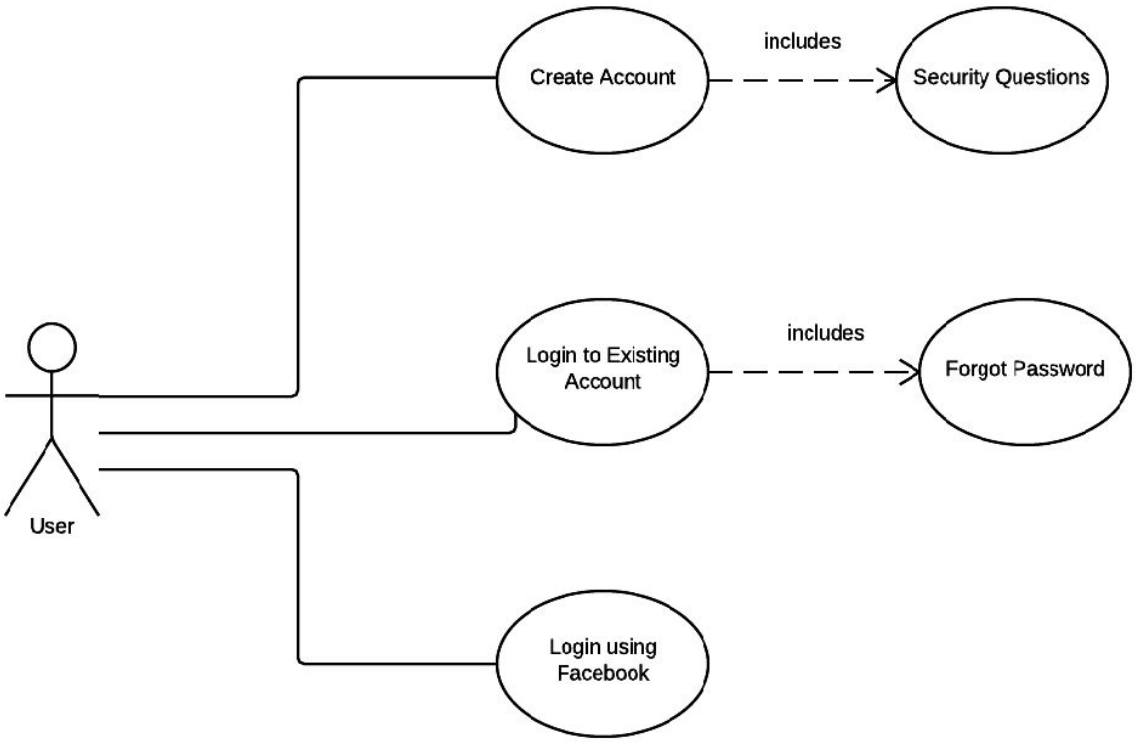
Use Cases

There are primarily two modules for which we have defined a set of use-cases:

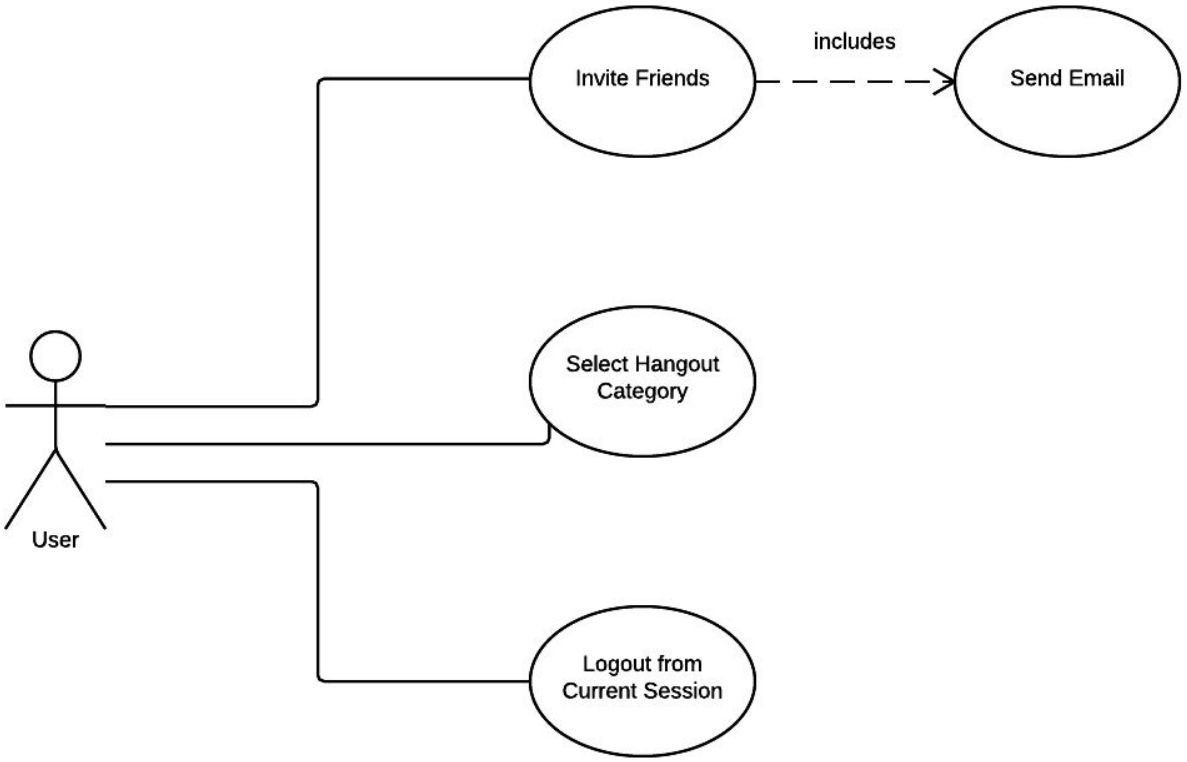
- User
- Business

User

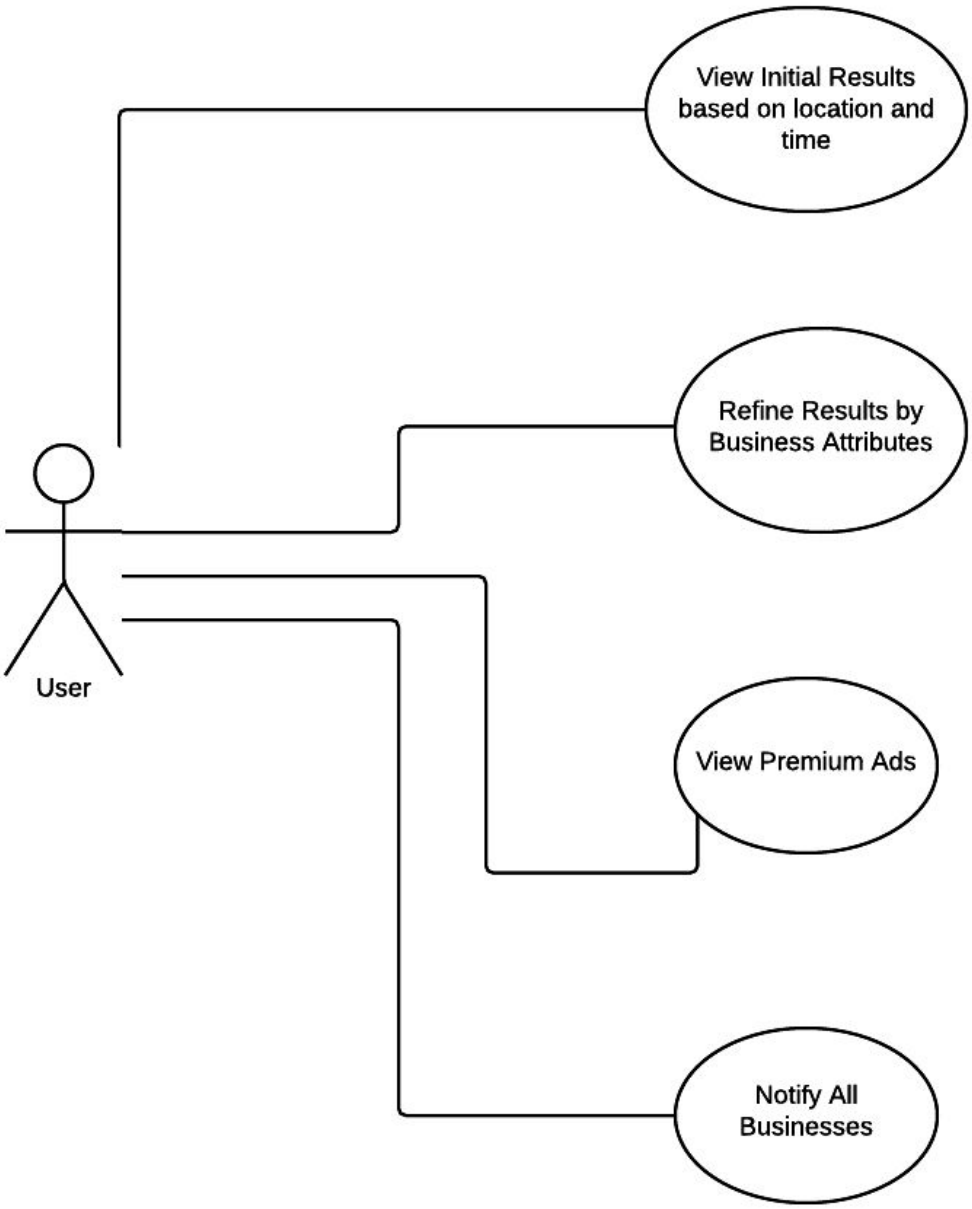
1. The User must be able to create an account, login to an existing account or login through Facebook.



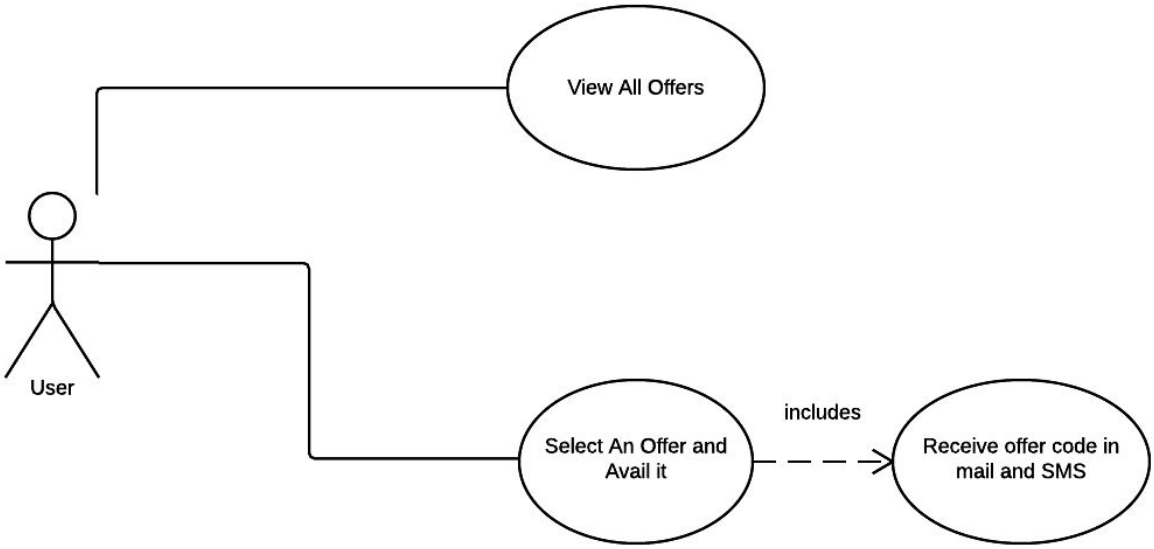
2. User must be able to invite friends and choose a category to hangout



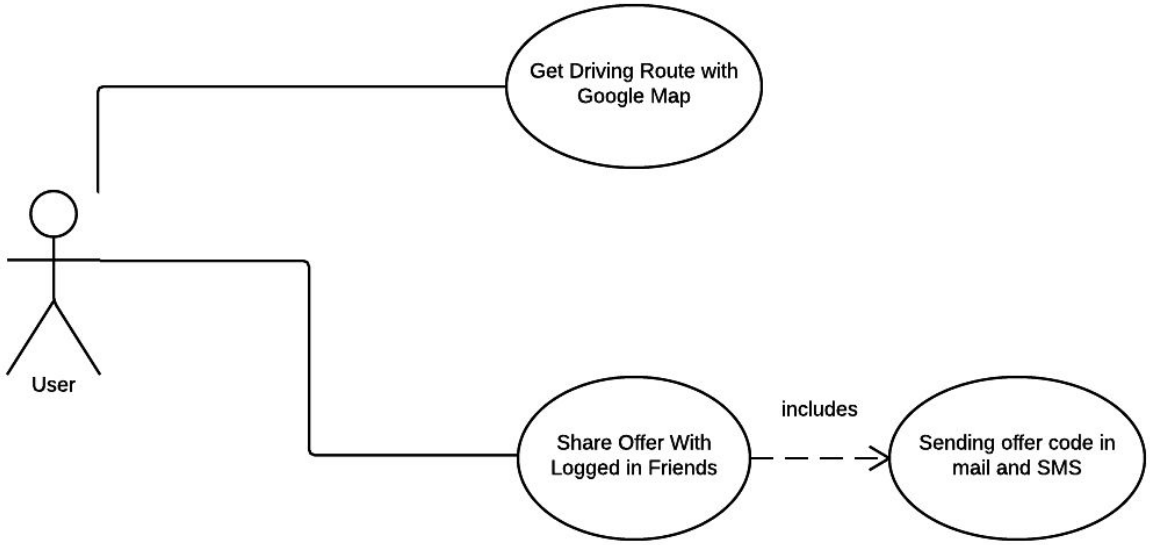
3. User refines tastes by options and views corresponding results. User additionally views Premium Flyers and clicks “Notify All” to get offers specific to results



4. User sees all offers and selects an offer by email and SMS

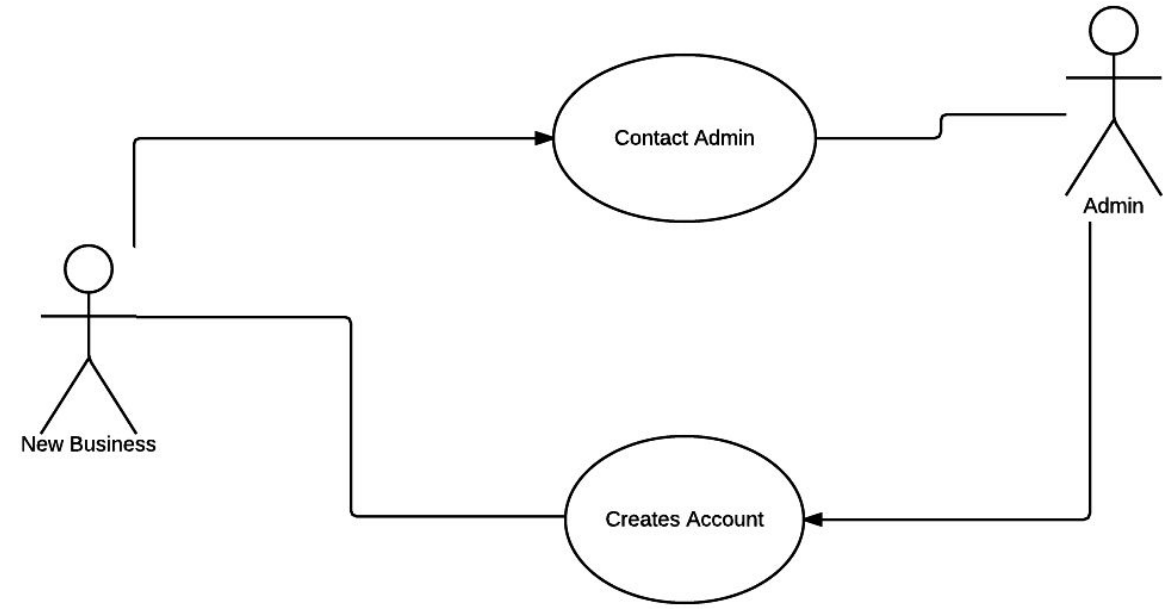


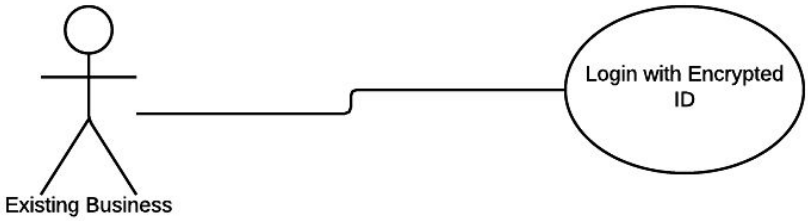
5. User can get Driving route to the business he/she availed offer with and also the option to share the offer with friends



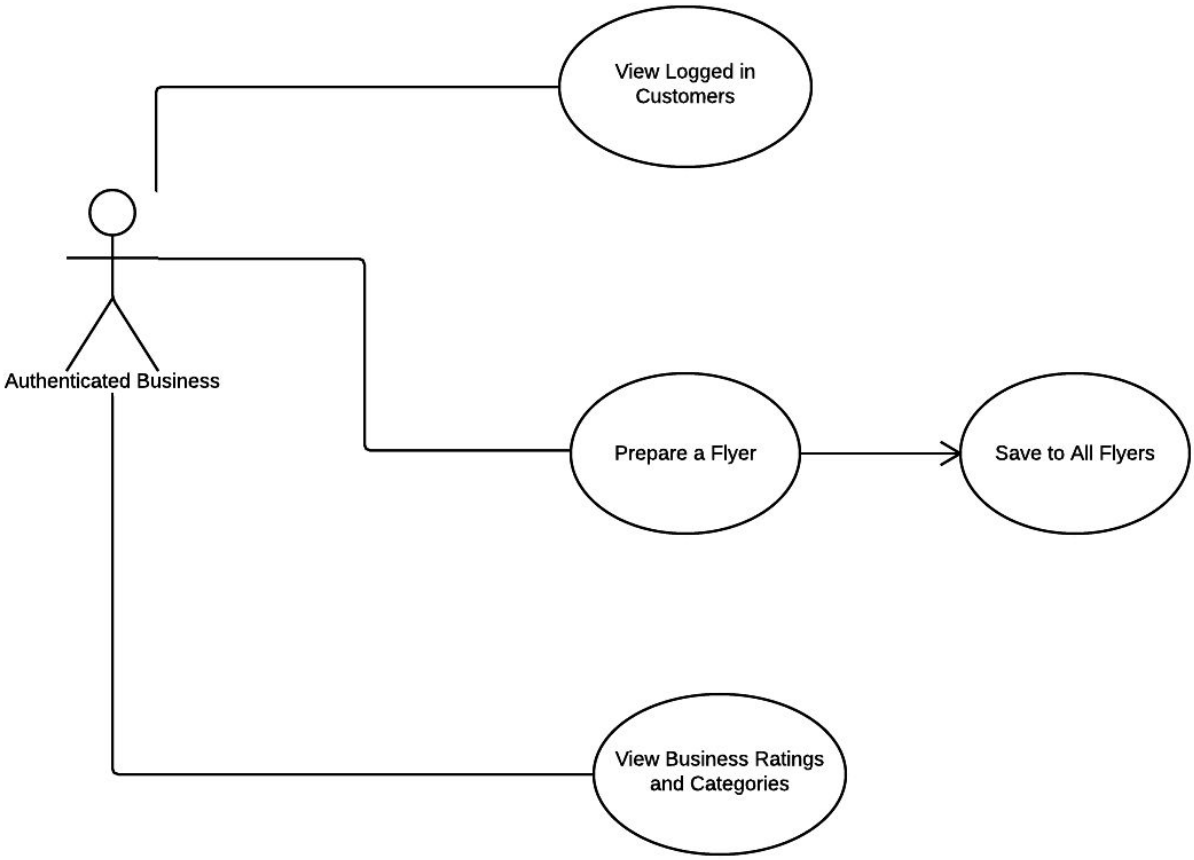
Business

1. The Business owner will login with Encrypted ID, he/she contacts administrator for creating account

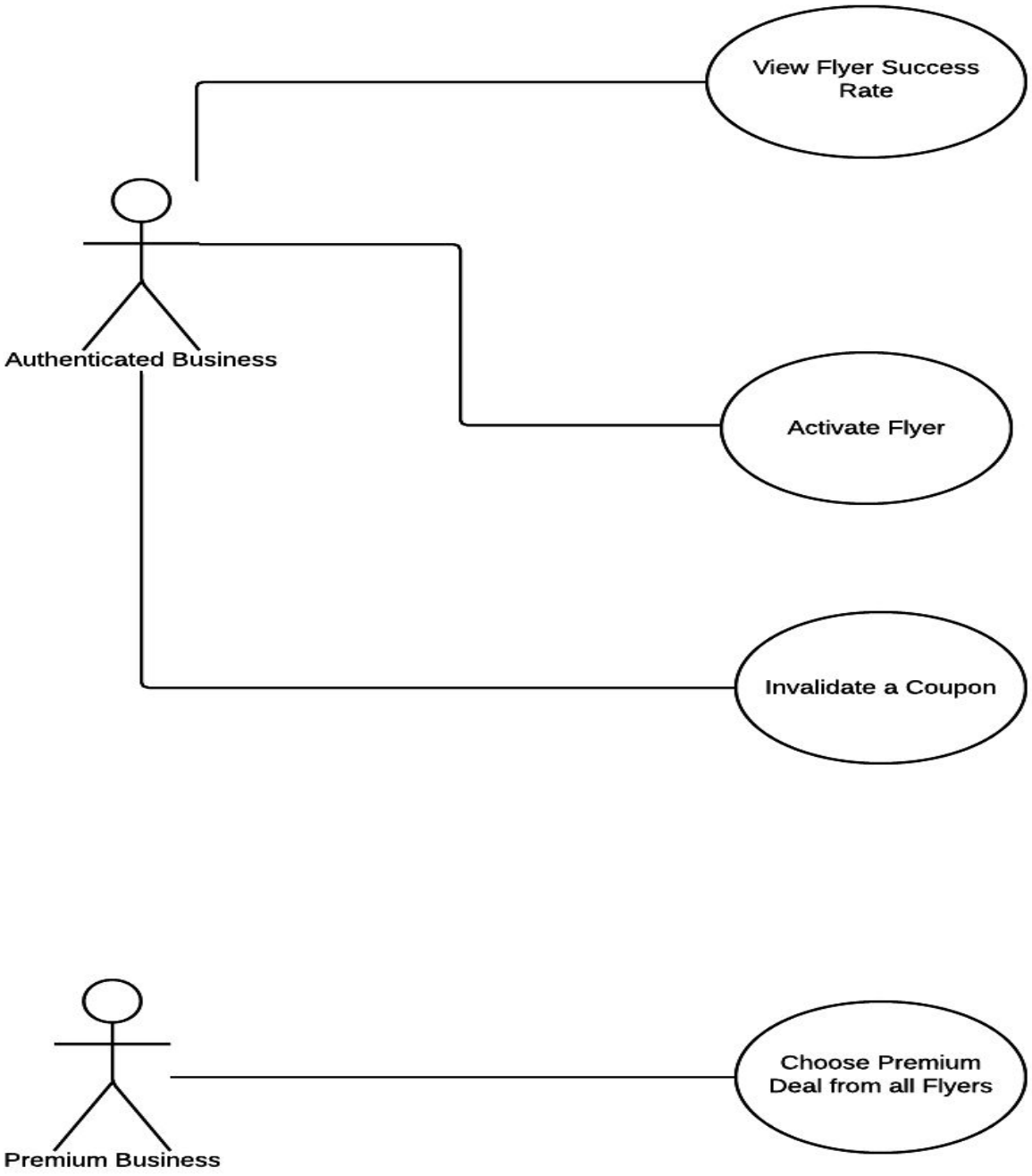




2. Business owner will be able to see all logged in customers, prepare a flyer and save it to All flyers. Check his/her business ratings and categories it belongs to.



3. Business owner can see the success rate of his/her flyer and choose one among the flyers to activate and send to user. One can also invalidate a coupon the Customer avails. The premium business can make premium deals from existing flyers


View Flyer Success Rate
Authenticated Business
Activate Flyer
Invalidate a Coupon
Premium Business
Choose Premium Deal from all Flyers

Optimization Techniques Employed & Performance Evaluation

Query/Database optimizations were done step by step in the entire application based on the retrievals required in each file. We used Full Joins initially to map multiple tables which took longer to access the dataset. Later on Joins were applied on temp tables created with certain where clauses (using where clause on columns that had indexes) that made joins faster. The effect of using such optimization was observed in the User Category/Option Selection page where the join was applied on five different Business tables to extract data based on the attributes selected. The same join technique was used in various pages. Also, Case statements were used to handle division by zero exceptions. The run time for the query went down from **0.234s** to **0.126s.**
Also, initially, cartesian product was performed on various tables in a random order which was optimized later by applying the same in the decreasing order of size which reduced time about **6-10%**. Further optimizations involved using subqueries instead of cartesian products which reduced it further by **6-10%**

Few of the example queries are listed below

```
SELECT name,business_id,full_address,stars,review_count FROM Business WHERE Business.city='Las Vegas') TEMP INNER JOIN (SELECT business_id FROM Business_Hours WHERE day='"+day+"' and open<= '"+time+"' and close>= '"+time+"')TEMPHOURS ON TEMP.business_id=TEMPHOURS.business_id INNER JOIN Business_Categories ON TEMP.business_id=Business_Categories.business_id INNER JOIN Categories ON Business_Categories.category_id=Categories.category_id INNER JOIN Attributes ON TEMP.business_id=Attributes.business_id WHERE Categories.category_name='"+request.newSession.hasCategory+"'" and stars>="+request.newSession.hasRating and Attributes."+column_name+"=1" and Attributes.take_out=1" and Attributes.delivery=1" and Attributes.alcohol NOT IN ('','none') and (Attributes.parking_lot=1 or Attributes.parking_street=1 or Attributes.parking_garage=1 or Attributes.parking_valet=1) and Attributes.outdoor_seating=1
```

```
select flyer_id, flyer_coupon, (selects/views * 100) as success_rate from (select flyer_id,flyer_coupon,No_of_selects as selects,Case when No_of_views = 0 then 1 else No_of_views end as views from Flyer where business_id='" + business_id + "' ) temp
```

Technical Specifications & Key Challenges Faced

The technology stack trace used by our application is as follows:

Front-end	Bootstrap(includes HTML,CSS, and Javascript), Jade Templating Engine
Back-end	My-SQL
Middleware	Node.js, Python
Additional	Google Image Search API, Redis, Google Maps,Twilio

Amongst the technical challenges faced, the first and foremost was that since we were using Node.js for the first time, understanding the asynchronous way in which Node.js functions, took some time since it is very different from the usual Sequential flow that most languages follow. Apart from that using Facebook API to implement login with facebook also took some time. Dynamically refreshing the page with new search results on the basis of the filter applied by the user on click of every checkbox also was a challenge since we had to make a dynamic and optimized query. Data cleaning and making it compatible to suit our requirements was also challenging.

Special Features

a. Mobile no verification and sending message for coupons availed
b. Email Verifications for login, registered businesses, invite friends, sending and sharing coupons, forgot password etc
c. Facebook Login
d. Robust Login/ Registration System
    i. Check for existing users
    ii. Password Authentication
    iii. Mandatory fields in registration
    iv. 3 security questions for registering users
e. Google Image Search API for displaying images of filtered businesses.
f. Google Map API
g. REDIS for storing reviews and review ids for the businesses
h. Invite friends and friends group, view logged-in friends and share coupons.

Division of Work

Aakriti Singla	Relational schema design and Middleware design
Krishna Chaitanya Daliparthy	User Interface Designer and Middleware design
Parul Bhalla	Relational schema design and Middleware design
Vamsi Jandhayala	Middleware design

Potential Future Extensions

a. Chat system implementation to let user and their friends communicate. Also implementation of group chat system
b. Recommendation system for user to help him decide on the filtered out businesses
c. Implementation for loyal customers (business wise)
d. Implementation for letting friend groups choose a group coupon