**FINAL REPORT**

**1. Project Statement**

**WHAT APP DO YOU WANT TO DEVELOP?**  


I came up with this idea as our supervisor made a group to report each activity happening in the parking lot and report him on daily basis during the heavy flow of traffic in the Bayside Parking lot. There after I came up with other problem as the new commuters of UMass Boston were experiencing trouble to find spaces in the parking lot during the head start of the semester.

Considering all these points I came up with an idea of designing a User Interface which will be very helpful for my supervisor(ADMIN) and the UMass commuters (GENERAL USERS).

**WHY DO YOU WANT TO DEVELOP SUCH AN APPLICATION?**

Students/faculty/visitors of the University have been experiencing problems while parking. As a result, have been getting late to their classes. This application notifies them about available spots in the lot. Allowing users to park their cars during rush hours. The application navigates the user to the nearest available parking lot.

**MOTIVATION AND OBJECTIVES**UMass Parking Lot it’s an application which helps UMass commuters/students park their cars using the real-time user interface, depending on the spots available in the respective parking lots in and around the UMass campus.

**PROJECT STATEMENT**This will be an app which keeps track of which UMass Boston parking lots are full. Specially designed for UMass commuters so that they can plan their commute accordingly and find a parking space using the real-time user interface, depending on the spots available in the respective parking lots in and around the UMass campus. The app presents a sign-in interface through which users may log in, which lists the status of the lots including the capacity of the lot, number of parking spaces available in a lot at the time, among others. The control of managing the status of each lot lies with the admin**.**

**IS THERE ANY SIMILAR APPLICATION ALREADY AVAILABLE ON THE ANDROID MARKET?**My main aim was to work and develop an android application for UMass commuters as the rising popularity of applications is increasing day by day. Apps such as Track the T, Best parking ,Parker were used by many Bostonians. Universities such as UMass Amherst already have an app, Park mobile, which can tell you available spaces on campus

**IS THERE ANY SPECIAL REQUIREMENT FOR USING THE APP?**

Lots in UMass has variations in their parking lots depending upon the number of spots each parking lot. I want to make an app that would mainly focus on two issues mainly ADMIN and GENERAL USER

User – Login in to application checks the spots available in the nearest parking area

Admin – Status of the parking lot is set by admin to organize spots in the parking lot.

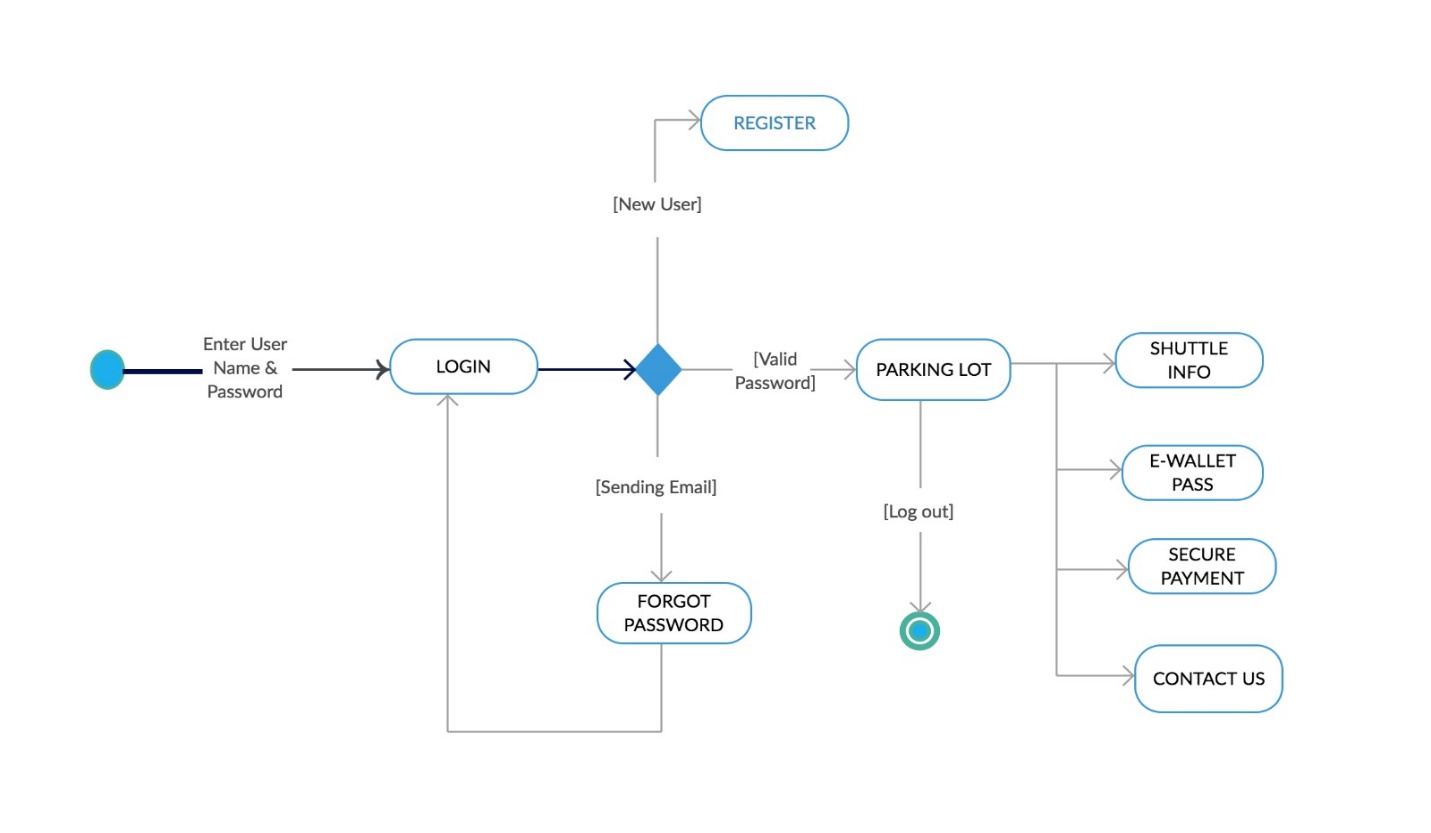
**2. Application Design**

**HIGH LEVEL DESIGN – UML (USE CASE DAIGRAM)**

**UMASS PARKING LOT PARKING SYSTEM**

****

**STATE DAIGRAM OF UMB PARKING LOT AND ITS FUNCTIONING**

****

MODULES:

MODULE 1 – Menu   
This page contains the contents about the menu which is placed on the top right corner of the application when the user logins to the account.  
onCreateOptionsMenu(Menu menu) method overrides in the activity class this creates menu and returns Boolean value . Inflater is used to create a new view or layout object from one of XML layouts. Inflate inflates a menu hierarchy from XML resource.   
**public boolean** onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.***menu\_splashscreen***, menu);  
 **return super**.onCreateOptionsMenu(menu);  
}

MODULE 2 – DBHelper (Local Database)

I am creating table Register and Location.

**public void** onCreate(SQLiteDatabase db) {  
 String query = **"create table Register("**+*FULL\_NAME*+**" varchar(40),"**+*USER\_NAME*+**" varchar(40),"**+*PASSWORD*+**" varchar(40),"**+*Gmail*+**" varchar(20),"**+*AGE*+**" varchar(20))"**;  
 String parkTable = **"create table Location(LocationName varchar(30),Lat varchar(20),Lon varchar(20),seekCount varchar(20))"**;

To insert values into the Location table we sue the syntax insert into tablename values(“”)

SQL(**"Insert into Location values('Boston Globe','42.3155805','-71.0532585','60')"**);

@Override  
**public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 onCreate(db);  
}

The onupgrade method is called when the database is needed to be upgraded.This is achieved by checking the value defined in the DATABASE\_VERSION constant.

MODULE 3 – Forget Password   
Intent email = **new** Intent(Intent.***ACTION\_SEND***);  
email.putExtra(Intent.***EXTRA\_EMAIL***, **new** String[] { to });  
A string array – To recipient email addresses  
email.putExtra(Intent.***EXTRA\_SUBJECT***, **"Recovery password"**);  
string with email subject   
email.putExtra(Intent.***EXTRA\_TEXT***, **"The last password id :"**+pass);

string with the body of the email   
email.setType(**"message/rfc822"**);*//email set type//*

setting type “message/rfc822” for intent to send e-mail   
startActivity(Intent.*createChooser*(email, **"Choose an Email client"**));

MODULE 4 – ListAdapter

**inflater** = (LayoutInflater) **mContext**.getSystemService(Context.***LAYOUT\_INFLATER\_SERVICE***);

*//Layout inflater is used to create a new view object from XML layouts//*

Layout Inflater class is used to initialize layout android.xml file into its corresponding view objects

It takes as input an XML file and builds the View object from it (or)  
Instantiate a place-holding view, which later will be used to display a row. If you want to instantiate the row view from a xml file, you should inflate that file which gives a view. The inflation process is done through a LayoutInflater object.   
convertView= **inflater**.inflate(R.layout.***list\_adapter***,**null**);

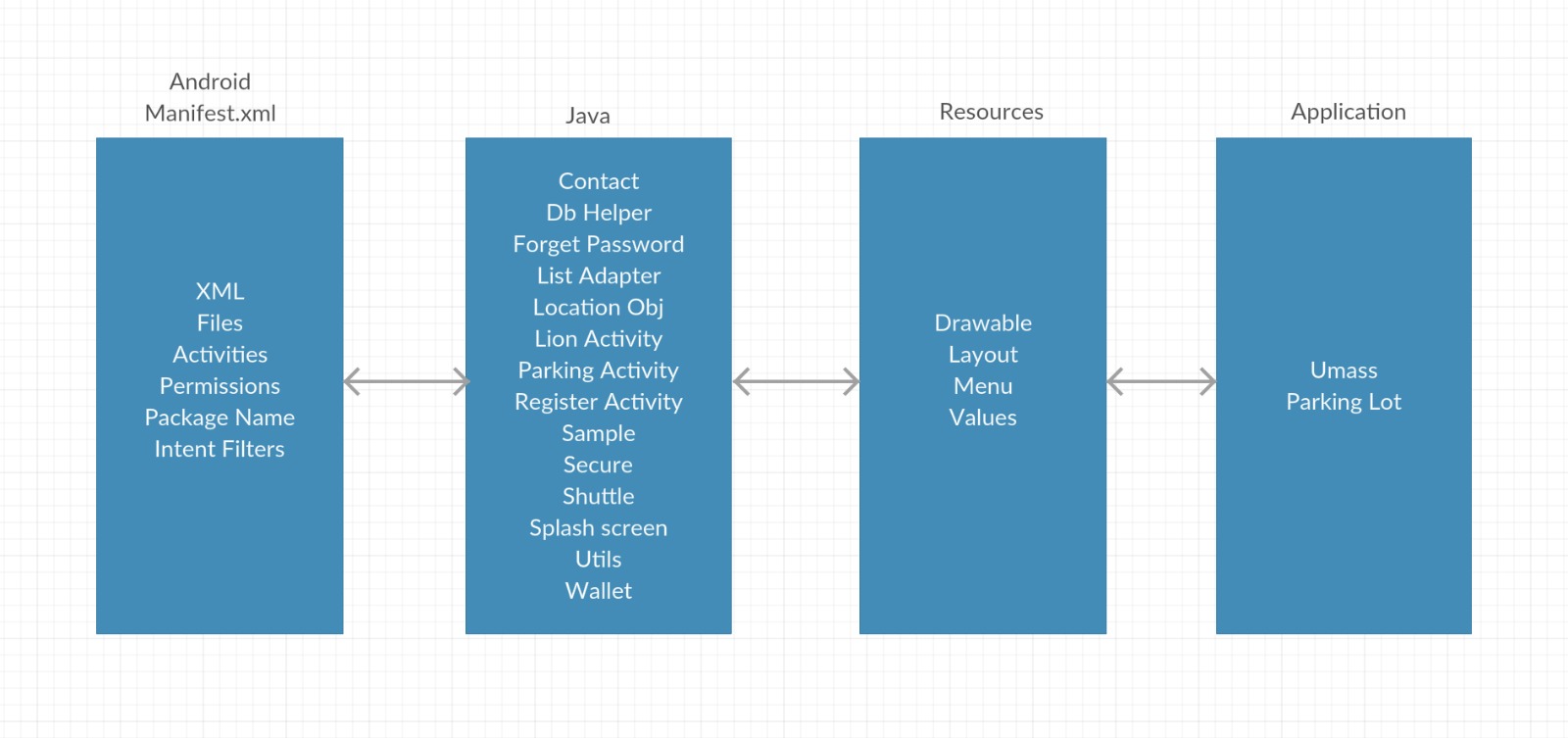
MODULE 5 (Local Database)

Shared preference allows us to store and retrieve, private primitive application data in the form of Key- Value pair .Shared preference data is lost when the user performs either of this task   
uninstall of the application (or) clearing the application data through settings  
Interface used for modifying data values in a shared preferences object and automatically commit those changes back using   
editor.commit (Permanent changes),  
 editor.apply(temporary changes)

SharedPreferences **sharedPreferences**;creating an shared preference object   
SharedPreferences.Editor **editor**;  
To save a value in share preference we use sharedpreferences.editor class  
**sheredPrefarence** = getSharedPreferences(Utils.***SH\_NAME***, Context.***MODE\_PRIVATE***);  
This method gets shared preference from a specified file, where preference name is the name of the file and context .mode private is the operating mode   
**editor** = **sheredPrefarence**.edit();

**editor**.apply();apply the editor object **editor**.commit();Commit the editor object   
  
I have targeted my application on moto e, HTC and Nexus android mobiles.

**Components Organized and Interacted with one another**



**3. Application Implementation and Evaluation**

**CLASSES IMPLEMENTED IN THE APP:**App: As usual like all the other apps my application is designed to work on a phone screen. Click on the UMBParking Lot icon and app starts functioning.   
Splash screen: This activity is displayed as the user clicks the icon UMBParking Lot (splash screen runs first)**.** If the user is already login in to the account it takes the user to the parking page or else it takes the user to the login page.  
LoinActivity:  
This page allows user to login, considering the username and password if the person is already registered as a user in our database. Otherwise the user can register and createaccount by clicking the register button. Incase if the user forgot his password, he can retrieve it through forgot password button and an email will be sent to user with his password to his email ID which is registered with the account. IF username = admin then login as admin (To login as admin one must mandatorily fill the username as “admin” as to avail the control over parking lot.

ELSE General user

The loinactivity is designed to interface between the rest of the app on the one hand, and a database storing username and passwords on the other. Storing data in the shared preference temporarily to access them is essential, otherwise all the data will be lost when the app is closed.

Shared preference interface **is** used for modifying data values in a shared preference object and uses commit those changes back using editor.commit() to change their data into permanent mode .This would make accessing and searching the data more efficient. If the details are not matched with the database it displays an error message “wrong details”.

RegisterActivity:  
This activity allows the user to register in to the app in order to access the parking lot info by choosing fullname , username , password , confirm password, email address, age and a submit button.  
I made some constraints as per USA rules the driver’s age should not be less than 18

Displays Toast message when the “age is lesser”

Email address – “Empty fields are not allowed”   
Email address – In order to consider the email ID unique, the app request the user by prompting the toast message “Email ID already exists” by cross checking it with the database.

ForgetPassword:   
This activity allows user to retrieve their password through email address registered by the user while creating an account.

DBHelper:

I have used two tables Register and Location.

Register Table has the following contents Fullname, Username, Password, Gmail, Age.

Location Table has the following contents Locationname, Lat-(Latitude),Lon-(Longitude),SeekCount

To Insert content in the table

Insert into Location values ('','','','')

DBHelper(context)  
onCreate(SQLiteDatabase):void  
onUpgrade(SQLiteDatabase,int,int):void  
updateContact(String,String):boolean

ListAdapter:

We use inflator in this class. Layout inflater is used to create a new view or layout object from one of theXML layouts. To display in the form of list I have used listadapter class.

Parking Activity: This activity is used to display Location Name, Latitude, Longitude, seek count . The parking activity uses ListAdapter in order to display it in list.  
ListAdapter adapter = **new** ListAdapter(**this**,**list**);  
 **lv**.setAdapter(adapter);

To display in the list we use adapter   
Users can access more information about each parking lot which provides users with all the basic information about each parking lot, and the user can navigate to the respective parking lot by clicking on the location (this activity is to be implemented)  
Page displays both printed and graphical representation of each parking lots ratio of cars to spaces. For now, the number of cars in each parking lot is a projection based on records kept at the UMass Boston Transportation Department. The temporary memory is stored in the shared preference which analyzes whether the user is admin or general user depending on its credibility admin = admin it proceeds ahead to perform the seek bar on start and on stop which works with respect to their location resulting admin to take control over all the umass parking lots.

LocationObj:  
This activity contains POJO class LocationObj to set and get LocationName, Lat, Lon, seekcount  
POJO - POJO is a Java object not bound by any restriction other than those forced by the Java Language Specification.  
  
Contact: This page provides the details of the UMass parking department contact info. Still working on to implement direct email, Secure Payment, Wallet

Secure: Secure Payment page accepts credit/debit card details to initiate the online mode of payment. If UMass wants to adopt this application this will be a most advance feature to consider online payment for UMass commuters. (due to time constraint, it is not implemented)

Wallet: E-Wallet Pass displays all the semester passes which will be an advance feature in the app (due to time constraint it is not implemented).

Shuttle:  
This page provides the basic schedule of the three UMass Boston shuttle busses.  
I usually run the application on android emulator and android mobile which is connected via USB. The app works perfectly without any errors. Due to time constraint, I was unable to implement the google maps. Even after implementing it I faced many problems as it says unexpected error due to google maps stopped working. I was unable to fix this issue. This link helped me a lot but it is not implemented in the application yet.

/\*\*

\* Method to decode polyline points

\* Courtesy : http://jeffreysambells.com/2010/05/27/decoding-polylines-from-google-maps-direction-api-with-java

\*/

**4. References**

<https://developer.android.com/guide/components/intents-common.html#Email>  
<https://developer.android.com/reference/android/view/LayoutInflater.html>  
<https://developer.android.com/training/data-storage/sqlite.html>

**5. Experiences and Thoughts**

**Future goals:** This app is designednow to easily add the following features:

1. Use real data from the UMass Transportation department to update the number of cars in real time. The parking lots at UMass have electronic gates which runs under AMANO system that can keep track of the exact number of cars in each lot so this feature would not be difficult to implement.
2. A point value is added whenever the vehicle enters the parking lot and -1 points is deducted whenever the vehicle passes through the exit gate.
3. Allow users to navigate using google Maps as in real time the user navigates from the origin and destination with display of miles, and duration of the journey.
4. Allow users to purchase semester passes through UMB parking lot. A complete online payment mode to ease of purchasing online semester passes. Moreover, this app has security bounded constraints in terms of secure password as it’s an most important feature related to online payment.

**Conclusion:**

I felt that implementing Admin/General user features on my app was an absolute necessity. If UMass were to adopt our app, I would want them to allow users to view the parking lots which is near by their current location to save their ample amount of time. As a result, commuters don’t get late to their classes using this real-time user interface application. Implementation of google maps and Purchasing parking passes through our app. This would be the most advance feature to be implemented in the app.