ABSTRACT

In any University, career services helps students identify their strengths and thereby possible career choices. It plays a vital role in deciding the student’s future career. Hence, students consider it as an important attribute while deciding the college. It conducts various counseling sessions, seminars which would make it easier for them to decide upon their career. Having a constant access to these services and regular updates about the sessions and seminars would be really helpful to the student. When we talk about constant access and regular updates mobile phone is the only thing that comes to our mind. It is one such device which is being widely used now-a-days and every student definitely has at least one. Our app would help the students have better access to the services currently provided. The services can be availed at anytime from anywhere avoiding the need to always have a computer/laptop. Whenever any new job is posted on the site, a notification can be sent to the student. They can update their profile, upload/download documents, apply for jobs, get notifications and check for event updates.
# TABLE OF CONTENTS

Abstract .................................................................................................................. ii

Table of Contents .................................................................................................. iii

List of Figures ......................................................................................................... v

List of Tables .......................................................................................................... vi

1. Background and Rationale ................................................................................. 1

1.1. Introduction ..................................................................................................... 1

1.1.1. Android ..................................................................................................... 2

1.1.2. Parse .......................................................................................................... 3

1.2. Existing Applications ..................................................................................... 3

1.2.1. University specific applications ................................................................. 3

1.2.2. College Career Center app ......................................................................... 3

2. Narrative ............................................................................................................ 4

2.1. Problem Statement ......................................................................................... 4

2.2. Motivation ....................................................................................................... 4

2.3. Product Description ....................................................................................... 4

2.4. Product Scope ................................................................................................ 5

2.5. System Requirements .................................................................................... 5

3. Application Design ........................................................................................... 6

3.1. Application Design and Architecture ......................................................... 6

3.2. Design Flow ................................................................................................... 7

3.3. Use Case Diagram ......................................................................................... 8
3.4. Class Diagram ................................................................................................................. 9

3.5. User Interface ................................................................................................................... 10

3.5.1. Login ............................................................................................................................. 10

3.5.2. Register ........................................................................................................................ 11

3.5.3. Reset Password ........................................................................................................... 11

3.5.4. Home ............................................................................................................................. 12

3.5.5. My Profile ..................................................................................................................... 14

3.5.6. Job Search .................................................................................................................... 16

3.5.7. My Documents ............................................................................................................ 18

3.5.8. My Activity .................................................................................................................. 20

3.5.9. Events .......................................................................................................................... 20

3.5.10. Query .......................................................................................................................... 21

3.5.11. Action Bar ................................................................................................................... 22

3.5.12. Settings ....................................................................................................................... 23

3.5.13. Push Notifications ..................................................................................................... 25

4. Implementation of Application Modules ............................................................................ 26

4.1. Parse Connection ............................................................................................................. 26

4.2. Login ............................................................................................................................... 27

4.3. Notification ...................................................................................................................... 28

4.4. Logout ............................................................................................................................... 30

5. Testing and Evaluation ....................................................................................................... 31

5.1. Registration ...................................................................................................................... 31

5.2. Reset Password ............................................................................................................... 33
5.3. Login ................................................................................................................. 34

5.4. Notifications .................................................................................................. 35

5.5. My Profile ....................................................................................................... 36

5.6. Apply Job ......................................................................................................... 38

5.7. User Feedback ................................................................................................ 39

6. Conclusion and Future Work .............................................................................. 40

Bibliography and References .................................................................................. 41

LIST OF FIGURES

Figure 3.1. Islander Career Services Architecture ................................................. 6

Figure 3.2. End User Design Flow ......................................................................... 7

Figure 3.3. Use Case Diagram .............................................................................. 8

Figure 3.4. Class Diagram ..................................................................................... 9

Figure 3.5. Initial Screen ...................................................................................... 10

Figure 3.6. Login Screen ...................................................................................... 10

Figure 3.7. Registration Screen ........................................................................... 11

Figure 3.8. Reset Password Screen ..................................................................... 12

Figure 3.9(a) Home Screen .................................................................................. 13

Figure 3.9(b) Home Screen .................................................................................. 13

Figure 3.10(a) My Profile ..................................................................................... 14

Figure 3.10(b) My Profile ..................................................................................... 15

Figure 3.10(c) My Profile ..................................................................................... 15

Figure 3.11 Job Search .......................................................................................... 16
| Figure 3.12 | Job Details | 17 |
| Figure 3.13 (a) | Apply Job | 18 |
| Figure 3.13 (b) | Apply Job | 18 |
| Figure 3.14 | My Documents | 19 |
| Figure 3.15 | My Activity | 20 |
| Figure 3.16 | Events | 21 |
| Figure 3.16(a) | Events List | 21 |
| Figure 3.16(b) | Employer Details | 21 |
| Figure 3.17 | Query | 22 |
| Figure 3.17(a) | Query in app | 22 |
| Figure 3.17(b) | Gmail app | 22 |
| Figure 3.18 | Action Bar | 23 |
| Figure 3.19 | Settings | 23-24 |
| Figure 3.20 | Push Notifications | 25 |
| Figure 4.1 | Code to connect with Parse | 26 |
| Figure 4.2 | Code for Login | 27 |
| Figure 4.3 | Code for setting alarm for job notifications | 28 |
| Figure 4.4 | Code to send push notifications for jobs | 28 |
| Figure 4.5 | Code for setting alarm for event notifications | 29 |
| Figure 4.6 | Code to send push notifications for events | 29 |
| Figure 4.7 | Code for logout | 30 |
| Figure 5.1 | Registration | 31 |
| Figure 5.1(a) | Positive Test Case | 31 |
Figure 5.1(b) Negative Test Case ................................................................. 32
Figure 5.1(c) Invalid Email and phone number ........................................ 32
Figure 5.1(d) Student is already registered ............................................. 32
Figure 5.2 Reset Password ........................................................................ 33
Figure 5.2(a) Positive Test Case ............................................................... 33
Figure 5.2(b) Negative Test Case .............................................................. 33
Figure 5.3 Email with link to Reset Password ........................................... 34
Figure 5.4 Login page Negative Case ....................................................... 34
Figure 5.5 Positive Test Case for Enabling/Disabling Notifications .......... 35
Figure 5.6 Positive Case for My Profile .................................................... 36
Figure 5.7 Negative Test Cases for My Profile .......................................... 37
Figure 5.8 Apply Job .................................................................................. 38
  Figure 5.8(a) Positive Test Case .............................................................. 38
  Figure 5.8(b) Negative Test Case ............................................................ 38

LIST OF TABLES

Table 4.1 Flags ......................................................................................... 30
BACKGROUND AND RATIONALE

1.1 Introduction

Finding jobs suitable for one’s capabilities has always been a challenge to every individual. In the distant past, people had a difficult time finding jobs because there was no proper platform wherein they can apply for jobs. Most of the times the jobs were descended to a son from his father and so on, it never went out of the family. Any new job posting would be known only if anyone informed it to you in person or sent a letter mentioning about that and then the applying for the job was also done in the same manner. When telephones were invented, this task became a little easier as talking on the phone solved the problem to a certain extent as it made the news spread faster and also many people could know about the job posting. When the internet came into picture, job application process evolved, there were many job portals which helped people in finding the jobs of their interest.

One of the concerns of the students in today’s world is to have a proper placement when they come out of college. Thus, the career services department of any university plays a major role when students finalize the college in which he/she wants to pursue his/her education. Career Services department of the University plays an active role in helping students decide their future career. It conducts various counselling sessions and seminars which would help the student decide easily their future career path. The entire journey of helping student identify his interests and strengths, helping him to find a job that suits his interest, guiding him with the resume writing, interview preparation till he finds a job is all taken care by the career services department. Hence, it is necessary for every student to have a good interaction with career services.
Career services of Texas A & M University Corpus Christi works hard to keep up to the expectations of the students. They try to be in contact with the student by conducting different career counselling sessions, seminars and career events. They show different career opportunities which the student can have based on his/her interests and strengths and help them to identify the best career option for themselves.

1.1.1. Android

Different mobile operating systems have been developed, but an Android is one of the mostly used mobile operating systems in today’s world. It is a free and open source operating system which is developed and maintained by Google. Because of it being an open source, many companies have adopted it and is widely used in different mobile devices in contrast with iOS which is specific only to the IPhone users. Many versions of android have been released till date, each having several updates. Each new version is named after a dessert, going in alphabetical order like: cupcake (Version 1.5) followed by Donut (Version 1.6), Éclair (Version 2.0 – 2.1), Froyo (Version 2.2), Gingerbread (Version 2.3), Honeycomb (Version 3.0 – 3.2), Ice Cream Sandwich (Version 4.0), Jelly Bean (Version 4.1- 4.3), KitKat (Version 4.4) and the latest one Lollipop (Version 5.0 – 5.1). The system as such comes with many built-in apps and because of it being open source, various apps are being developed depending on the needs and are being put on the play store for everyone to use [1].

According to International Data Corporation Android dominates the smartphone market with a share of 82.8%, iOS being 13.9% and the rest is shared by Windows, Blackberry and other operating systems [2]. Thus developing an Android application for the career services would be the best way to make sure the services properly reach to the student community.
1.1.2. Parse

Parse provides backend services to mobile apps. It provides Software Development Kits (SDKs) for devices running iOS, Android, Windows (Phone) 8, OS X and JavaScript. With the help of the SDK, a connection to the database on the cloud can be established and also other services like push notifications can be used [3].

1.2 Existing Applications

There are few existing android applications discussed below.

1.2.1. University specific applications

University of Missouri-Kansas City has an android application “UMKC Career Services App” [4] designed for its career services department. This app is specific to their university and would require login credentials to go into the app for more details.

1.2.2. College Career Center App

Another application “College Career Center” [5] is developed by the Snagajob. It needs the career service department of the university to register with them and then can be customized to generate notifications, reminders etc. While registering the university with the app, it asks to provide the domain name used by the students for the university email address. This information is used to provide specific and authorized access of the University to its students. We have a specific job database for our University and all the jobs are not available to all the students, based on the profile, interests and requirements each student is shown limited number of jobs. The main problem using this application is that it doesn’t have access to our jobs database, thus completely sideling the idea of the finding jobs by the career services for the students.
2. NARRATIVE

2.1 Problem Statement

As career services play an important role in deciding student’s career prospects, it is very important for every student to have a constant access to it. The career services at Texas A & M University Corpus Christi currently operates in two ways, through office or through the web application. The students can directly go and interact with the professionals/mentors at the office or they could take the help of the web application to find jobs, know about events. The web application requires a web browser to be used, thus it becomes a bit difficult to access it at anytime and anywhere when the student doesn’t have a computer/laptop with data connection. Mobile phone browsers can also be used to open the website, but it is not convenient to use because of the restrictions on the screen size. Also many jobs are added on a daily basis to the website, and there is no proper notification system which can help the students know immediately about it. The only way would be to regularly open the website and check for any new job posts and event notifications [6].

2.2 Motivation

To help students overcome this problem of having better access to the University Career Services, an android application can be created, which gives a better user experience and reduces the gap between the University Career Services and the students.

2.3 Product Description

This Android application helps students to complete the initial registration process with the university career services, update profile information, search for any jobs matching their profile and requirements, check for events, get some tips related to resume writing and interview preparation and also post any queries for the mentors/career
professionals. Also notifications are sent to the students if any new job postings matching their profile and reminders for the upcoming events.

2.4 Product Scope

This app is developed for students who have android phones and internet connection is needed to access it. The current students who are registered with the university can use this app using their University ID and the password. The Alumni may use this app using their alumni user name and password. New students can register themselves first with the career services, their identity is validated by and thus provided with the temporary password, which can be used to access the services. This app is compatible with different versions of android starting from Honeycomb (Version 3.0 – 3.2) to Lollipop (Version 5.0 – 5.1).

2.5 System Requirements

The following are required to develop this android application.

- Operating System : Windows
- Database : Parse
- IDE : Android Studio
- Software : Android SDK (Software Development Kit), Java SE 7
- Android Device : Android phone
3. APPLICATION DESIGN

3.1 Application Design and Architecture

Figure 3.1 shows an overview of the architecture of the proposed application. The end user, in this case student, downloads the app on his/her android mobile. To start using the app and have access to the data the student needs to first login to the system using his university ID. Only students at the university can have access to the app as university ID is needed for authentication purpose. The admin on his/her part enters the new job postings and event details and the student can have an access to them using the app from anywhere at any time. All the data is stored on the parse cloud server which acts as an intermediate between the student and the career services admin.

The user must have an internet connection, to have access to the data on the parse server, to get latest data and to make any changes to their profile.
3.2 Design Flow

Figure 3.2 shows the basic design flow of all the functionalities the student has. The student after installing the app, logs into the app using his university id and password. If the student is using the career services for the first time i.e., if he/she is a new student, he/she needs to first register, if the registration is successful he is redirected to the home page with a pending status.

![Figure 3.2 End User Design Flow](image-url)
If the student forgets his/her password, he/she can choose “forgot password?” option and input his/her university id, if the university id is valid, then a mail is sent to him/her with a link to reset his/her password. Once the login credentials are validated, the student is taken to the home page where he/she can view/edit his/her profile, find jobs, documents uploaded by him/her, events and also can leave any queries for the mentors. At the end, the student can logout of the app using the logout option [7].

3.3 Use Case Diagram

The student register himself/herself if he/she is using the services for the first time or logs in using his university id and password or uses forgot password if he/she forgets his/her password. The students can check their profile, any new jobs, and upcoming events and also post any queries for the mentors.

Figure 3.3 Use Case Diagram for Student
3.4 Class Diagram

Figure 3.4 shows the class diagram of the “myprofile” and “events” functionalities of the student. Student and Career Fair are created using the addClass() method of the Parse class.

When the student logs in to the app he has options like view/edit profile, find job, view upcoming events etc. In the “myprofile”, the user can view his profile details like the name, university id, email, phone number, mail id etc. He has an option to edit the data and save it. In the “events”, the user can view information about the upcoming events, their location, start date and end date details etc. All the data processed by the student is stored into Student class of the parse cloud server.
3.5 User Interface

3.5.1 Login

Figure 3.5 shows the screen which is shown when the app opens. Figure 3.6 shows the login screen in which the student needs to give in his credentials i.e. University ID and password to login to the app.

If the student is new to the university and doesn’t have an account with the career services department, he/she needs to themselves by clicking on the “Click here to register!” link in Figure 3.6. Every new student must register himself/herself with the career services to avail the services provided by the department. If the student forgets his/her password, he/she can click on the “Forgot your Password?” link to reset his/her password.
3.5.2 Register

The screens in Figure 3.7 are used by the new students for registering themselves with the career services department. The student has to enter all the details asked for to complete the registration process successfully. As soon as the registration is done, the student is logged in to the app.

![Figure 3.7 Registration Screen](image)

When the students register themselves, they are put in a pending status, till they upload their resume and their details are validated by the admin who then changes the status to active.

We can go back to the login page by clicking on the arrow in the action bar before the app logo and the screen name.

3.5.3. Reset Password

Figure 3.8(a) shows the screen to reset the password if the student forgets it. The
The student needs to give his/her University ID as input.

![Figure 3.8 Reset Password Screen](image)

Once the student enters his/her University ID and clicks on the reset password button, the screen in Figure 3.8(b) is shown. As mentioned in Figure 3.8(b) a link is sent to the mail ID he/she is registered with in the system. The mail contains a link, which when clicked leads the student to a page where he/she can enter their new password. After this, the student can use the new password to login to the app.

### 3.5.4. Home

The screens shown in Figure 3.9 are the home screens which the student sees once he/she logs in to the app successfully. The screen in Figure 3.9(a) is shown for the students whose status is “Pending” i.e. the students whose account is still under review. When the student is in this status he/she doesn’t have access to the jobs database i.e. they
cannot see the “Job Search” and “My Activity” options.

![Home Screen](image)

**Figure 3.9 Home Screen**

When the student has his/her status “Active”, he can work with all the options like “My Profile” is used to update his/her profile, “Job Search” to search for new jobs available, “My Documents” to upload/view/delete the related documents, “My Activity” to check the job status for the jobs which he/she applied, “Events” to check the latest events, “Query” to send any queries to the department.

The images at the bottom of the screen are used to access the career services pages on their websites. The first one, Facebook, is to access the career service’s Facebook page, second one is for the twitter page, third for the Instagram page, and fourth LinkedIn page followed by the Pinterest and YouTube pages. The student can like the pages and also
view them to get any latest updates.

3.5.5. My Profile

Figure 3.10 shows the My Profile Screen in which by default all the fields are disabled and most of them filled by the students while registering themselves.

![My Profile Screen](image)

Figure 3.10 (a) My Profile (Cont’d)

There are two buttons in this screen, the “Edit” button which helps the student edit the information and the “Save” button to save the updated information.

When the student clicks on the “Edit” button, all the fields are enabled, ready to make changes, once the changes are done the “Save” button needs to be clicked which updates the profile. When the changes are updated successfully, “The profile is updated” message is shown as an alert as shown in Figure 3.10 (b).
All the fields that have “*” before them are mandatory and must be filled before saving.

To go back to the Home screen to access other options, the students can either click on the arrow icon before the logo and the screen name, or they can click on the three dots present in the right end of the action bar which has three options as shown in Figure 3.10(c), “Home” which takes the student back to the home screen to access other options and “Settings” to either change password or set notifications for jobs and events. The back button option which is normally used to go to previous screen is disabled and hence cannot be used to go back to the Home screen.
3.5.6. Job Search

Figure 3.11 shows the Job Search screens of the app. The student can search the job based on the two parameters “Position Type” and “Job Function” as shown in Figure 3.11(a). The student chooses the values from the drop down list and then clicks on the “Search” button.

![Job Search Screenshots](image)

(a) (b)

*Figure 3.11 Job Search*

Based on the student’s profile and the values given for the Position type and Job Function, the jobs that suite the profile and the values, are chosen and displayed as a list as shown in Figure 3.11(b). Each item in the list displays the job details such as job ID, job title, job location, position type and the date when the job was posted. For further details the student can click on the job details and he is redirected to another screen.
shown in Figure 3.11(c). This screen gives more details about the job, such as number of openings, pay per period, start date, job description, qualifications, application instructions, online application address, wage/salary and the hours per week.

The student can apply for the job if he/she is interested by clicking on the Apply button which is at the end of the screen as shown in Figure 3.12. When the student clicks on the apply button, the screen in Figure 3.13 is displayed. The student can add any message for the employer and also he/she needs to include his/her resume when applying the job. Once the application is done, an alert dialog with message “Done” is displayed and the submit button is disabled to avoid the re-application. The user can go back and verify the application in the “My Activity” screen.
Once the student is done with the application, the user can click on the arrow before the app icon on the action bar to go to the previous screen that is the job details. The student can also click on the three dots at the right corner of the action bar to open a dropdown which has the home button which takes the student to the home screen.

3.5.7 My Documents

The screen shown in Figure 3.14 is the My Documents screen. The student uploads his/her documents such as resume, cover letter and class schedule. All the files must be uploaded only in the pdf format. When a new student registers himself, he would be in “Pending status”, he needs to upload his resume, which would be reviewed and then the status is changed to “Active”. Initially when no file is uploaded, the view and delete buttons in all the three sections are disabled.
The “Add/Update” button is used to add a new document or update the one which is already uploaded by uploading a new document. The “View” button is used to view the file which is uploaded to the database and the “Delete” button is used to delete the file from the database.

Once the student uploads the file, the file name is displayed in italics and the view and delete buttons are enabled. And when the student deletes the file, the view and delete buttons are again disabled and the file name in italics is shown as “** No File**”. Resume is very much necessary initially to get an active status and also later on for applying to different jobs, hence it must be always uploaded.
3.5.8 My Activity

Figure 3.15 shows the “My Activity” page, which has details about the jobs the student has applied for. The details for each job such as the job ID, Job title are shown. Also the date of application and the employer response which tells about the status of the job is displayed or if the employer has a message for the student.

![My Activity Screen](image)

Figure 3.15 My Activity

When the student applies for a job, a new record is added to the My Activity screen. The student can check for any updates about the job here.

3.5.9 Events

The screen shown in Figure 3.16(a) shows the list of upcoming events, the location of the event, category, event type, the start date and the end date. The student can click on the career event in the list to see the list of employers who would be attending the event as shown in Figure 3.16(b).
The employer details include the name of the employer, web link and the category. The value under website column is a link which when clicked on redirects to the employer’s website.

### 3.5.10 Query

Whenever the student has any questions he/she can use this option to send a mail to the career counsellors at the career services. The screen for this option is as shown in Figure 3.17(a). It has two fields to enter the subject and the body for the mail.
When the student gives the values and clicks on the send, the Gmail app opens as shown in Figure 3.17(b), a new mail is composed with the details given in the app and the student must click on the send button in the Gmail app to send the mail to the department. When the student clicks on the send button in the Gmail app, that is once he/she sends the mail, he/she is redirected to the home screen in the app.

3.5.11 Action Bar

Action Bar has options on both sides. The right side of the action bar has three dots which when clicked on has three options: Home, Settings and Logout. The left side has an arrow which directs to the previous screen. The “Home” option when clicked on redirects the student to the Home screen of the app from anywhere in the app.
The Settings option has internally two types: General and Notifications. General settings has one option, “Change password” and Notifications Settings has two options, “Allow Notifications for Jobs” and “Allow Notifications for events”. Logout option on the action bar allows the student to logout from the app from any screen in the app.

3.5.12. Settings

Settings has two options General settings and Notifications settings as shown in Figure 3.19(a). The General settings currently has one option “Change Password” as shown in Figure 3.19(b).
This option is used to change the password from inside the app after the student is logged into his/her account. This redirects to the change password screen in Figure 3.8. A link is sent to the email, where a new password can be given by the student.

The Notification Settings has two options “Allow Notifications for jobs” and “Allow notifications for events”. These two options can be enabled when they are checked and disabled when unchecked respectively. The notification settings for jobs and events can be done from three places. Initially the student enables or disables the notifications in the registration screen as in Figure 3.7 and then in the My Profile screen while updating his/her profile and lastly in the Notification Settings under the Settings option in Action Bar.
3.5.13 Push Notifications

Push Notifications are sent to the student whenever a new job which matches the student’s profile is posted and any new career event is posted. The student receives the link depending on the notifications are enabled or not. If the notification for both jobs and events are enabled he/she receives both else only the one which he/she enabled.

Figure 3.20 Push Notifications

When the student clicks on the notification, he/she is redirected to the app, where he/she can check the update.
4. IMPLEMENTATION OF APPLICATION MODULES

4.1 Parse Connection

This android app is connected to the cloud database server using the Parse REST API [3]. The code in Figure 4.1 is used to connect to the Parse. All the Parse Objects defined in the database, have an entity class which extends Parse Object i.e. subclass of Parse Object defined in the code, which is registered. Student.class, CareerFair.class, Job.class, Employer.class and MyActivity.class.

```java
public class GlobalClass extends Application {

    private String username;

    public void onCreate() {
        super.onCreate();

        Parse.enableLocalDatastore(this);
        //Registering entity class objects which are subclasses of the ParseObject class
        ParseObject.registerSubclass(Student.class);
        ParseObject.registerSubclass(CareerFair.class);
        ParseObject.registerSubclass(Job.class);
        ParseObject.registerSubclass(Employee.class);
        ParseObject.registerSubclass(MyActivity.class);
        //Application ID and Client Key used for initialization
        Parse.initialize(this, "OfQXpDfDU6uqrwLp4ZYSbIyaIC00jmXF4DN8DxbV", "CpDQRCQhHssD4SB7iNL7iiGkDUszIGEcgfAnfag");
        ParseInstallation.getCurrentInstallation().saveInBackground();
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
}
```

Figure 4.1 Code to Connect with Parse
4.2. Login

The student when gives the login credentials in the login page, with the help of ParseUser.logInInBackground (uid, pwd, new LogInCallback ()); the user is logged into the app, if the credentials are valid, else “Invalid Login Parameters” message is displayed in the alert dialog box. When the student is successfully logged in the details are stored in SaveSharedPreferences to maintain the session i.e., to keep the User logged in. Also the notification permissions are stored in the shared preferences.

```java
public void onClick (View v) {
    Student s = new Student();
    final GlobalClass globalvar = (GlobalClass) getApplicationContext ();
    switch(v.getId ()){
        case R.id.bLogin://When Login button is clicked
            uid = etUniversityID.getText().toString ();
            pwd = etPassword.getText().toString();
            ParseUser.logInInBackground(uid, pwd, new LogInCallback() {
                public void done(ParseUser user, ParseException e) {
                    if (user != null) {// Successful login
                        globalvar.setUsername(user.getUsername());
                        SaveSharedPreference.setUserName(MainActivity.this,
                        user.getUsername());
                        SaveSharedPreference.setPassword(MainActivity.this, pwd);
                        startActivity(new Intent(MainActivity.this, HomeActivity.class).
                        putExtra("sharedpref", false));//start a home page activity
                    } else {// Signup failed
                        new AlertDialog.Builder(MainActivity.this).setTitle("Oops").setMessage("Invalid Login Parameters").setNeutralButton("Ok", null).show();
                    }
                }
            });
    ParseQuery<Student> q = ParseQuery.getQuery("Student");
    q.whereEqualTo("StudentUniversityID", uid);
    q.selectKeys(Arrays.asList("notificationsforjobs","notificationsforevents"));
    try { List<Student> studentList = q.find();
        if(studentList !=null) {   for(Student s1 :studentList){
            SaveSharedPreference.setNJobs(MainActivity.this,
            String.valueOf(s1.isNotificationsforjobs()));
            SaveSharedPreference.setNEvents(MainActivity.this,
            String.valueOf(s1.isNotificationsforevents()));
    } } } catch (ParseException e) {
        e.printStackTrace();
    } break;
}
```

Figure 4.2 Code for Login
4.3 Notification

Push Notifications are generated based on if any new job or event is posted. An Alarm Manager is used to set an alarm twice a day which runs a service as in Figure 4.3, used to check if any new job/ event is posted and accordingly a new push notification is generated as in Figure 4.4. Push notifications in parse are sent using google cloud messaging, internally a setup is done on the parse server, which enables the push notifications to be generated with the help of the code.

```java
AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
if (s.isNotificationsforjobs()) {
    Calendar c = Calendar.getInstance();
    c.set(Calendar.HOUR_OF_DAY, 0);
    c.set(Calendar.MINUTE, 0);
    c.set(Calendar.SECOND, 0);
    Intent intent = new Intent(this, JobReceiver.class);
    intent.putExtra("uid", uid);
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, c.getTimeInMillis(),
    TimeUnit.HOURS.toMillis(12), PendingIntent.getBroadcast(this, 0, intent,
    PendingIntent.FLAG_UPDATE_CURRENT));
}
```

**Figure 4.3 Code for Setting Alarm for Job Notifications**

```java
ParseQuery pushQuery = ParseInstallation.getQuery();
pushQuery.whereEqual("user", ParseUser.getCurrentUser());
ParsePush push = new ParsePush();
push.setQuery(pushQuery); // Set our Installation query
if(j.getJob_id() != null)
push.setMessage("New job posted id:"+j.getJob_id());
else push.setMessage("New job Posted");
push.sendInBackground();
```

**Figure 4.4 Code to Send Push Notifications for Jobs**
The code snippets for the event notifications are shown in Figure 4.5 and Figure 4.6 respectively.

```java
AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
if (s.isNotificationsforevents()) {
    Calendar c = Calendar.getInstance();
    c.set(Calendar.HOUR_OF_DAY, 0);
    c.set(Calendar.MINUTE, 0);
    c.set(Calendar.SECOND, 0);
    Intent intent = new Intent(this, EventReceiver.class);
    intent.putExtra("uid", uid);
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, c.getTimeInMillis(),
                             TimeUnit.HOURS.toMillis(12), PendingIntent.getBroadcast(this, 1, intent,
                             PendingIntent.FLAG_UPDATE_CURRENT));
}
```  

**Figure 4.5 Code for Setting Alarm for Event Notifications**

```java
ParseQuery<CareerFair> query = ParseQuery.getQuery("CareerFair");
query.whereLessThanOrEqualTo("createdAt", new Date());
query.whereGreaterThanOrEqualTo("createdAt", new Date(new Date().getTime() -
TimeUnit.HOURS.toMillis(12)));
try{
    if(query.count() >0) {
        query.findInBackground(new FindCallback<CareerFair>() {
            @Override
            public void done(List<CareerFair> list, ParseException e) {
                if (e == null) {
                    if (list != null) {
                        ParseQuery pushQuery = ParseInstallation.getQuery();
                        pushQuery.whereEqualTo("user", ParseUser.getCurrentUser());
                        ParsePush push = new ParsePush();
                        push.setQuery(pushQuery); // Set our Installation query
                        push.setMessage("New event Posted");
                        push.sendInBackground();
                    }
                }
            }
        });
    }catch(ParseException e){
        e.printStackTrace();
    }
}
```  

**Figure 4.6 Code to Send Push Notifications for Events**
4.4 Logout

The student can use the option in the action bar to logout of the app. The following flags are set.

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAG_ACTIVITY_CLEAR_TASK</td>
<td>If set in an intent passed to Context.startActivity(), this flag will cause any existing task that would be associated with the activity to be cleared before the activity is started.</td>
</tr>
<tr>
<td>FLAG_ACTIVITY_CLEAR_TOP</td>
<td>If set, and the activity being launched is already running in the current task, then instead of launching a new instance of that activity, all of the other activities on top of it will be closed and this Intent will be delivered to the (now on top) old activity as a new Intent.</td>
</tr>
<tr>
<td>FLAG_ACTIVITY_NEW_TASK</td>
<td>If set, this activity will become the start of the new task on this history stack</td>
</tr>
</tbody>
</table>

Table 4.1 Flags

```java
Intent myIntent = new Intent(LogoutActivity.this, MainActivity.class);
myIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);// clear back stack
myIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);// If set, this activity will become the start of the new task on this history stack
myIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);// this flag will cause any existing task that would be associated with the activity to be cleared before the activity is started.
ParseUser.logOut();
ParseUser currentUser = ParseUser.getCurrentUser();
if(currentUser == null) {
    System.out.println("User logged out");
}
SharedPreferences.Editor editor =
SaveSharedPreference.getSharedPreferences(this).edit();
editor.clear(); //clear all stored data
```

Figure 4.7 Code for Logout
5. TESTING AND EVALUATION

This application is tested on a Motorola Moto X 1st Generation mobile device with Android version 5.1. The following are the test cases that were tested.

5.1 Registration

The student needs to give all the fields in the form provided when he/she is registering himself/herself for the career services.

**Positive Test Case:** If all the fields are filled out in the form, the student is logged into the app with the pending status mentioning the account is under review and asking to upload the resume under My Documents as shown in Figure 5.1(a).

![Figure 5.1 (a) Positive Test Case](image)

**Negative Test Case:** If the student misses out on any fields an error message is displayed asking to enter all the fields as shown in Figure 5.1(b).

**Negative Test Case:** When the email id is not given properly, a message saying please enter a valid email id is shown in the alert dialog box. It is very important for the email id to be proper because some of the important communication like link to reset password
happens through email as shown in Figure 5.1(c).

Figure 5.1. Registration

**Negative Test Case**: If student is already registered, we get the following messages as shown in Figure 5.1(d).
5.2 Reset Password

When the student wants to reset his/her password, he/she needs to give a valid University ID.

**Positive Test Case:** If the student enters the proper University ID, he/she is directed to another screen, which says the student will receive an e-mail shortly with a link to reset the password and a button to return to login page is also present.

![Positive Test Case](image1)

![Positive Test Case](image2)

(a) Positive Test Case  (b) Negative Test Case

**Figure 5.2**

**Negative Test Case:** If the student doesn’t give his/her University ID and clicks on the reset password button an alert dialog box pops up asking to enter the University ID. The student then has to click “Ok” and then gives in the university ID.

If the student gives an invalid University ID, he/she gets a message asking to enter a valid University ID.
The email received when the action of reset password results in positive test case results in an e-mail as shown in Figure 5.3.

![Email with link to Reset Password](image1)

**Figure 5.3 Email with link to Reset Password**

### 5.3 Login

When the student tries to login to the system, he/she is supposed to enter correct login credentials. The student can login to the system in two ways, when the student registers for the first time he is logged in to the system directly and put in a pending status restricting access to few options. Other way is direct login, when the user gives the credentials in login page.

![Login page Negative Test Case](image2)

**Figure 5.4 Login page Negative Test Case**
**Negative Test Case:** If the student gives wrong credentials, invalid password or username, an alert dialog box with a message “Invalid Login Credentials” is shown as in Figure 5.4.

**Positive Test Case:** If the credentials given by the student are correct, the student is redirected to the home screen and based on his status, he will see all the options or has a restricted access.

**5.4 Notifications**

The student can enable/ disable notifications in three ways. When the student is new and is registering himself/herself with the services for the first time, he/she has an option in the registration page to enable or disable the notifications for jobs and events. He/ She can update it in the My Profile page or in the settings option available in the action bar.

![My Profile and Notifications Screenshots](image_url)

*Figure 5.5 Positive Test Case for Enabling/Disabling Notifications*
Negative Test Case: If the change is made to the notifications in the My Profile page, the changes must be reflected in the settings option in the action bar.

Positive Test Case: The changes made in the My Profile page is reflected in the settings option in action bar.

Positive Test Case: The student must receive the notifications only when he/she asks for it by enabling the allow notifications option.

5.5 My Profile

The student can edit his/her profile details using this screen. There are few fields which are mandatory and needs to be filled or cannot be left blank. The mandatory fields are marked with a star at the beginning.

Positive Test Case: All the mandatory fields are filled and the profile is saved as shown in Figure 5.6.

![Figure 5.6 Positive Test Case](image-url)
**Negative Test Case:** If the mandatory fields are not filled, alert dialogs with messages asking the user to give values to the mandatory fields are displayed.
If first name is not filled then, the message as shown Figure 5.7(a) is displayed, if last name is not filled then, the message as shown in Figure 5.7(b) is displayed, if email ids are not valid then, the message as shown in Figure 5.7(c) is displayed and if phone number is less than or greater than 10 digits then, the message as shown in Figure 5.7(d) is displayed.

5.6 Apply Job

The student can apply for the job using this screen. He / She can leave a message for the employer while applying for the job.

Positive Test Case: When resume is uploaded as shown in Figure 5.8(a).

Negative Test Case: When resume is not uploaded as shown in Figure 5.8(b).

![Figure 5.8. Apply Job](image)
5.7 User Feedback

Several students were contacted to receive feedback, on the usability of the application. The students liked the outlook of the app, also they found it user friendly and very helpful to keep in constant contact with the career services department of the University.
6. CONCLUSION AND FUTURE WORK

This main aim of developing this application was to provide students a better access to the career services. This application now allows the students

- To create a profile by registering themselves and be able to update their profile.
- Upload their documents such as resume, cover letter and class schedule.
- To stay up-to-date with the latest events and new jobs posted, by receiving notifications whenever a new job or event is posted and also constant reminders about upcoming events.
- To apply for new jobs and also see the already applied one’s in the activity.
- To utilize the services provided by them efficiently at any time and from anywhere.
- To stay connected with the career services department by being able post any queries for the mentors.

Some of the future enhancements that can be done are

- Extending it for iOS users.
- Having a mobile version for the admin side as well.
- Help section can be created, which can help students understand all the navigations better.


