**JOB-AMIGO**

**A Complete Job Portal**

**Increment-3 (PG-6)**

**Kommineni, Siva Krishna**

**Thallapalli, Ravisha**

**Bandaru, Sarath Chandra**

**Yempalla, Suresh Reddy**

**Summary:**

Our application “**JOB-AMIGO**” is a complete job portal which deals with the job search using various parameters and various services are provided. Here the user basically registers in to the system and search the portal with his requirements. Authentication and registration services are hosted in the server. We have developed a service for searching based on the country codes and can view all the jobs in that specific job feeds. He can review through the retrieved results and apply to the desired position directly by redirecting to the company website. When he applied for a position an alert message is generated and sends to his registered mobile number. There are some existing services used for the weather and google maps which are helpful in finding out the destinations and weather in that particular region. Besides this there is QR & Instagram functionalities used for code scanning and sharing the images to friends and networks. Finally a resume service is built and submitted to the portal for the registered user which takes to the end of the job search task which is hosted on the UMKC server.

**Framework Specification:**

The frame work mainly consists of 3 stages which are Android GUI, User DB and the parsers for parsing the data. All the user operations are performed in between these 3 stages in various ways. The parsers help in parsing the data to the required format from JSON/XML format so that the user can understand easily from the UI. All the user validations and registrations are taken care using the DB and some hosted web Services. Finally data will be updated in the DB accordingly in the created tables which are received and validated from the services.



**System Architecture:**



The above diagram clearly depicts the system architecture of our application. Let us follow the architecture in a sequence manner. We use some of the existing services and API’s. Besides this some services are developed and hosted on the UMKC server using the SQL DB and finally REST/SOAP services are produced which are useful for resume building, User validations, Registration and the authentications. All these services parse the JSON data and retrieve the result to the UI. Some existing services like the google maps, Weather location services, Job feeds from some other job portals are used. The data from the feeds are in the XML format. These data is parsed using the XML parser and results are retrieved to the UI when the user search based on his requirement. We have data base tables stored in the SQL DB which gets updated whenever the particular service comes into picture.

Now coming to the Android GUI part First, the user login into the system after completion of his registration. There are basic validations and authentications provided to the system for security purpose so that only the registered user can perform his operations on the portal. For this there is a validation service and registration service provided. Later when the user logs into the system there are options provided with various services and functionalities developed. He searches the jobs online where the results are retrieved using the parsers on to the UI. When the user is interested in a particular job posting based on his search criteria, he can apply it directly from the portal as it redirects to the company profile. At the moment when he applies for a job post he receives an alert message to his registered mobile number notifying that he applied for a particular position from the job portal. There is resume service provided to store the user profile in the DB.

**Domain Model:**

**Data Sources:-**

Data is collected from various job portals and surveys. We used only useful data which gives the XML/JSON data so that parsing of the data is easy using the XML and JSON parsers. This parsed data models are finally retrieved to the GUI.

1. <http://www.linkup.com/developers/>

2. <http://www.programmableweb.com/category/jobs/apis?category=20080>

3. <http://www.labor.ny.gov/jobs/regional.shtm>

4.[https://data.ny.gov/browse?Dataset- Information\_Agency=Labor%2C+Department+of&utf8=%E2%9C%93](https://data.ny.gov/browse?Dataset-%20%20Information_Agency=Labor%2C+Department+of&utf8=%E2%9C%93)

**Methodologies and Algorithms:**

The methodologies we used are the services hosted on the UMKC servers and some of the existing API’S that support our application.

We used XML/JSON Parsers, REST/SOAP services for various task validations and authentications.

DB for storing the data and retrieving required results for performing the validation checks.

Once we have all the available services and the parsers it is very easy for the implementation of the logic in the sequence manner as follows.

The below flow chart describes the basic methods and tasks employed in our application.

**Flow Chart:**



**Analytic Tools:**

The entire process goes on in a sequence manner.

The steps are as follows:

1. User registers into the system
2. Search for the required skill set
3. Receive SMS alerts
4. Resume service provider
5. Existing service support to identify the climatic condition of the location

**Analytical Tasks:**

Various analyzing tasks are predefined in some of the existing API’S and helpful for searching the jobs easier for the user. They can identify easily by their search criteria and retrieval is made using some XML/JSON parsers.

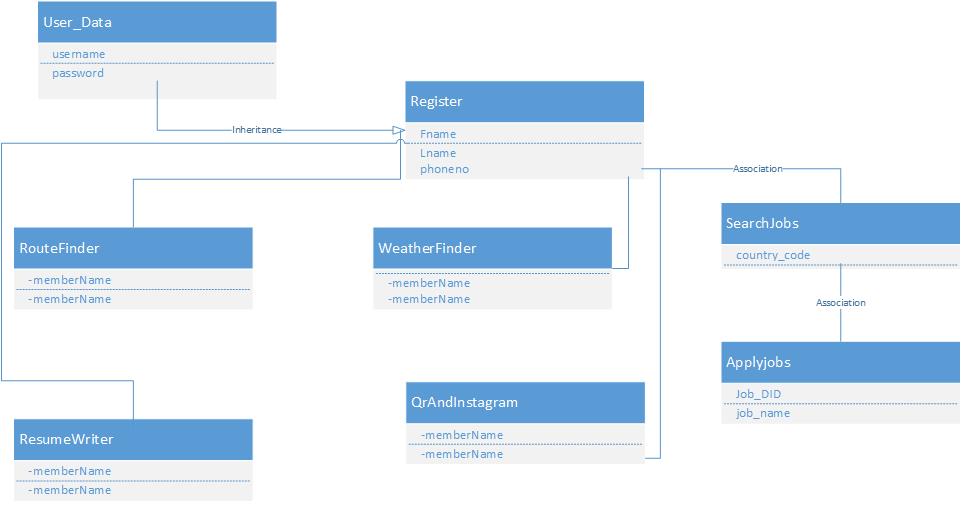
**Application Specifications:**

**Software Specifications:**

* Tools: Android Development Kit, Microsoft Visual Studio
* Operating System: Android, Windows
* Development Operating System: Windows 8
* Programming Language: Java 7.0, C#,ASP.NET
* Databases: MySQL, SQLite

**Class Diagram:**

The below screen shows all the classes and their functionalities present in the application.



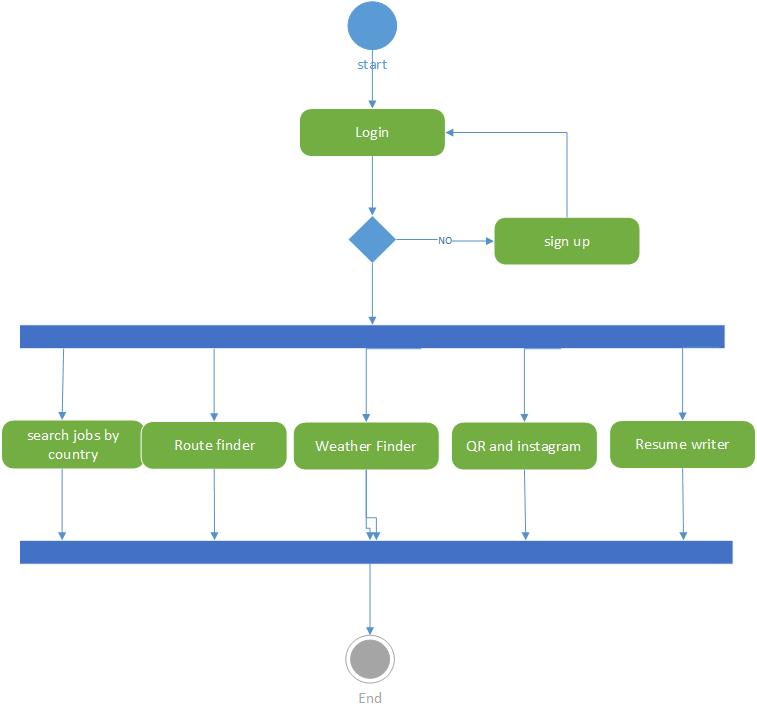
**Sequence Diagram:**



The above diagram depicts the sequence diagram of our application which clearly explains all the steps in a sequence manner

**Activity Diagram:**

Various activities from the initial start to the final end state is clearly depicted using the activity diagram.

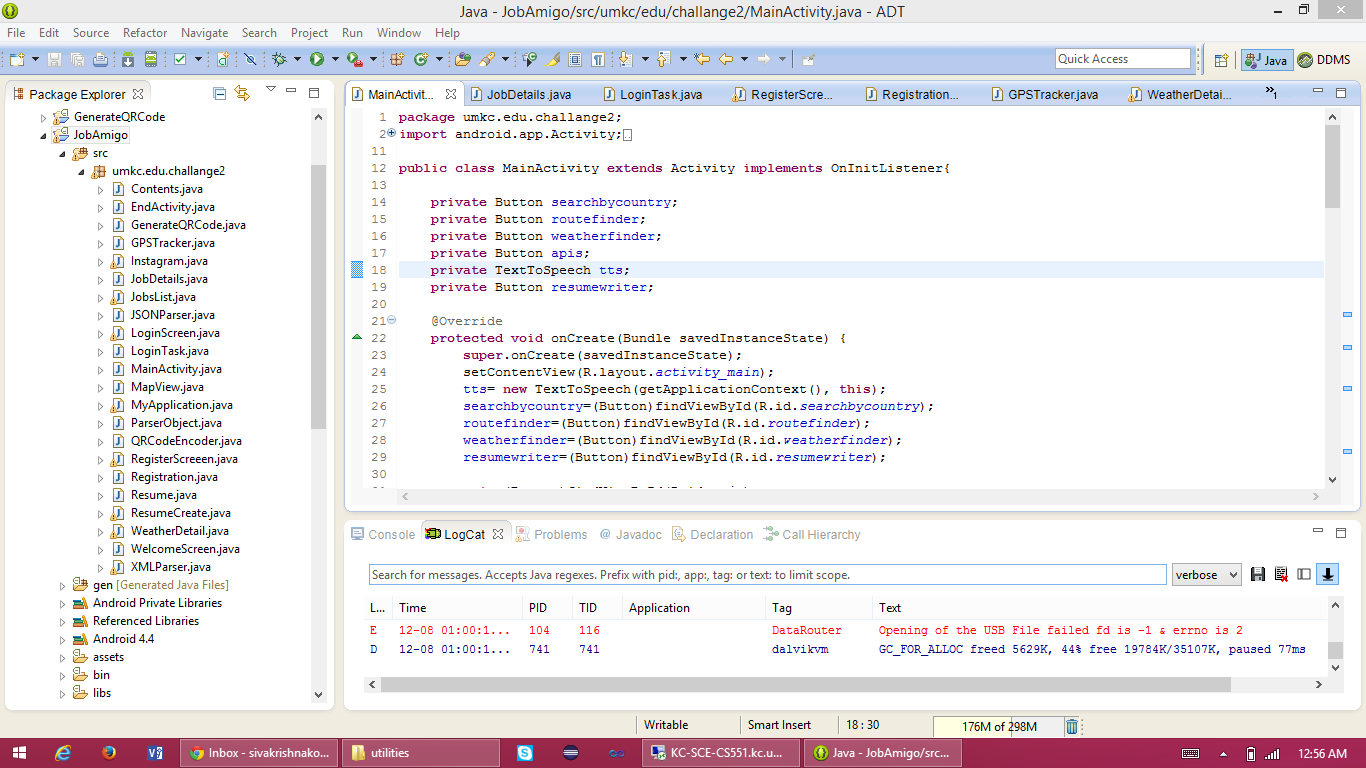
****

**Design of Mobile Interface:**

The entire mobile interface design is combination of both service and API’S. There are some of the existing API’S used and user defined services are also used in GUI at certain levels.

**Implementation:**

There is navigation between the UI and the services developed. All these are handled using the Android SDK environment. The implemented classes and services are shown as below.



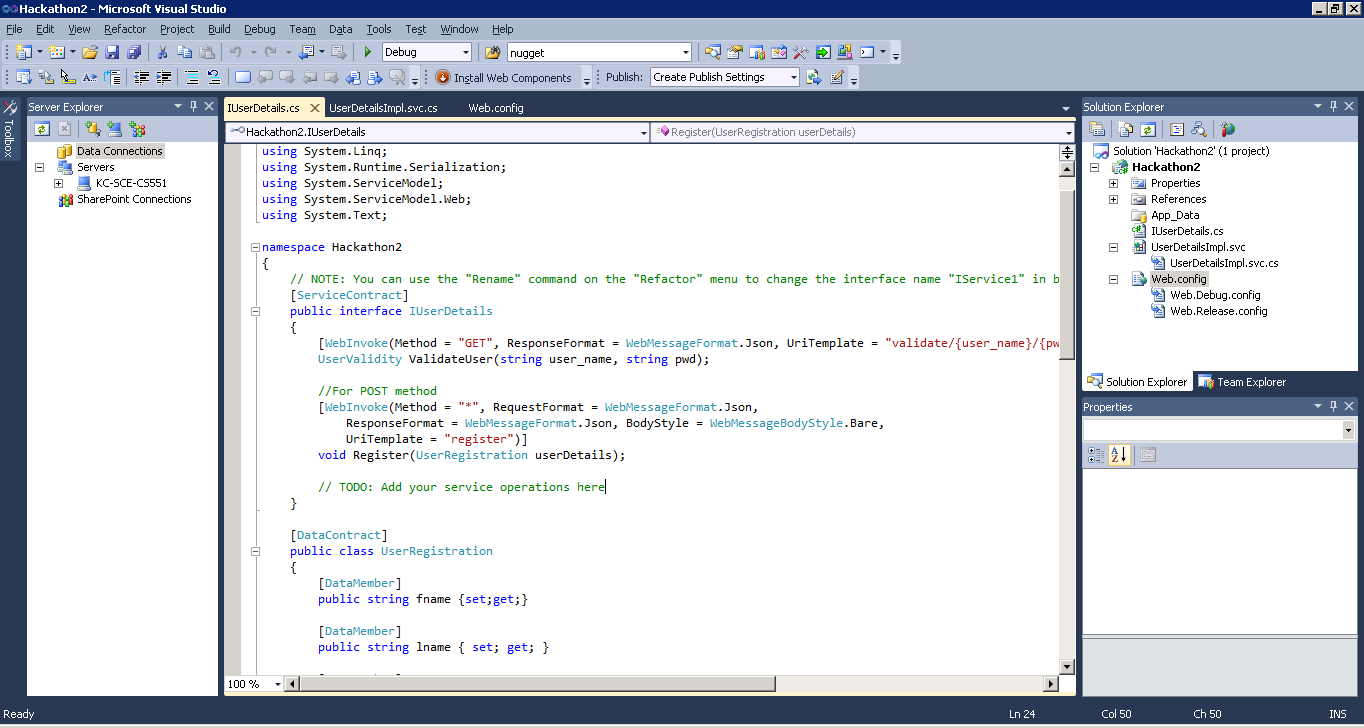
**Implementation of services:**

1. **Your own services**

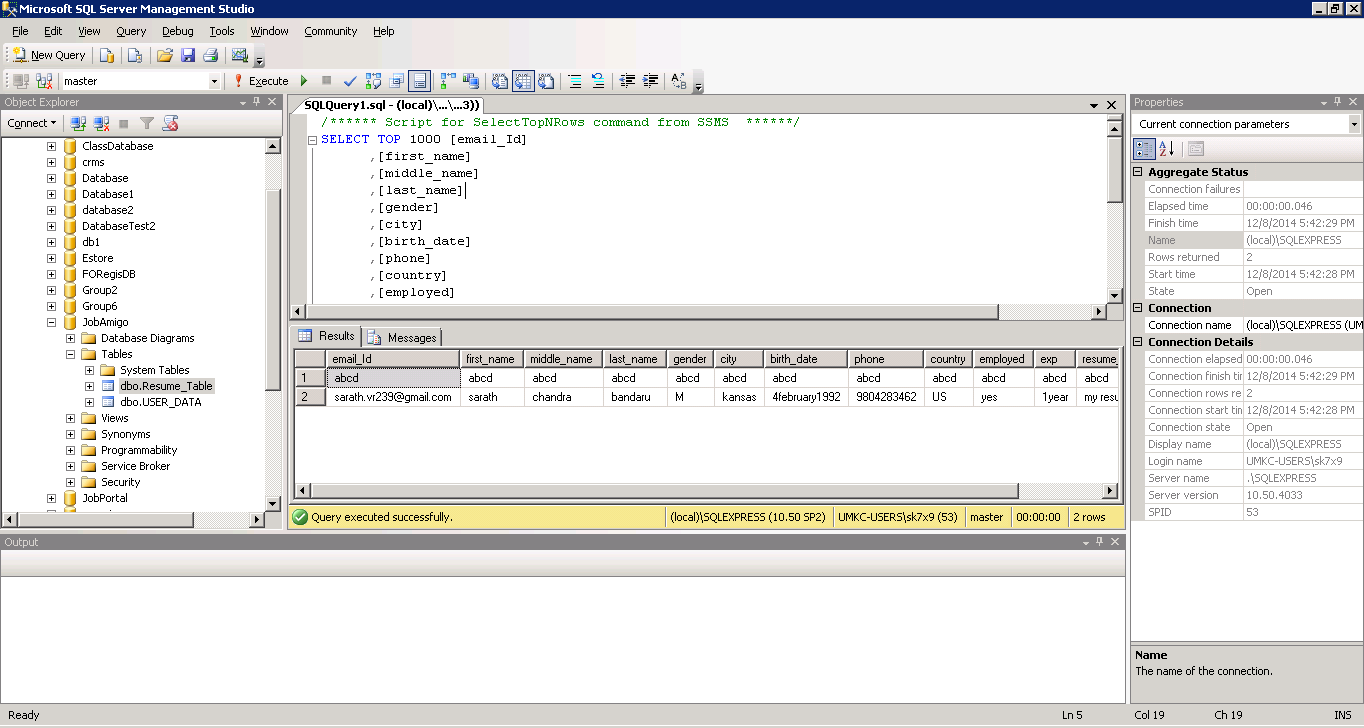
<http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/T8-Restful/Service1.svc/insertInfo/siva/24/umkc/student>

Various operations and services like insertion, deletion and updating of the user data into the data base is been performed.

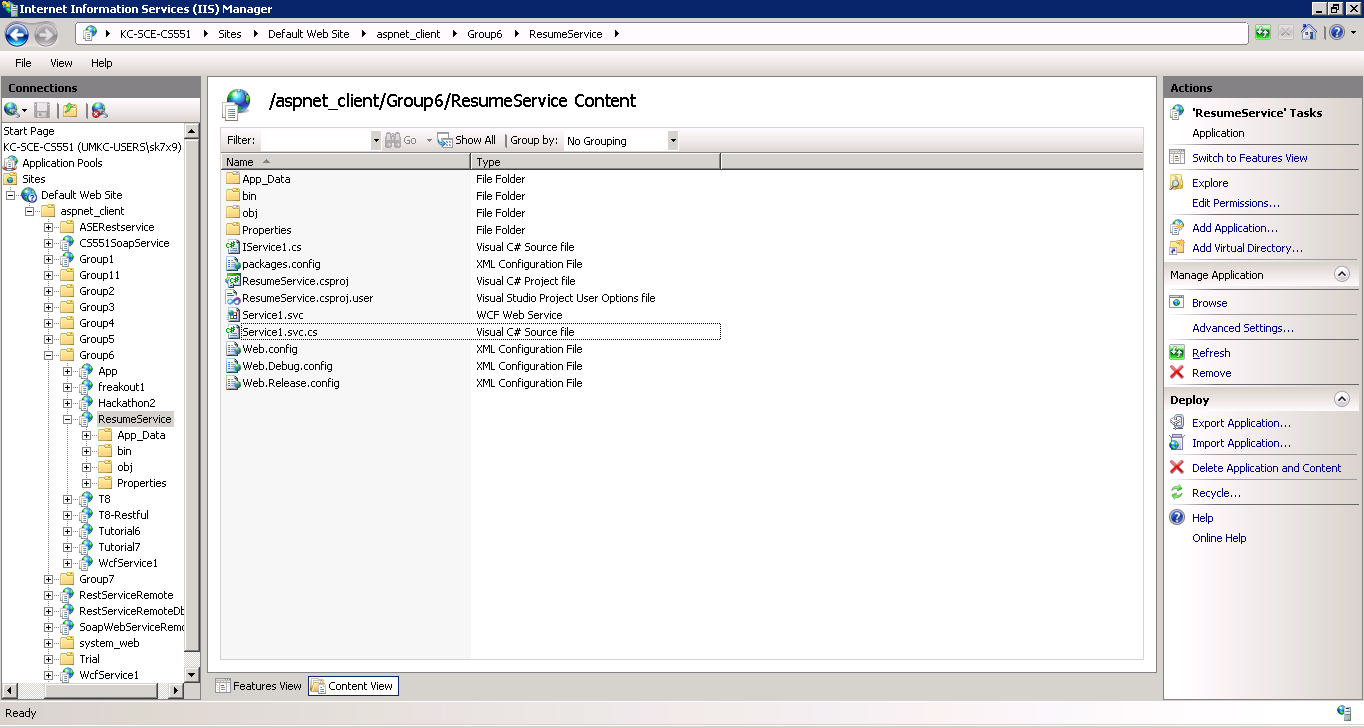
The following Screen shot shows the service developed in Visual Studio.



The below screen shot describes DB creation in SQL Server.



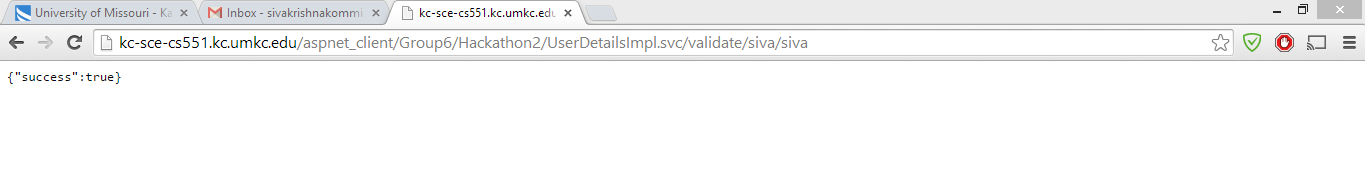
Hosting the application on to the remote IIS server for browsing.

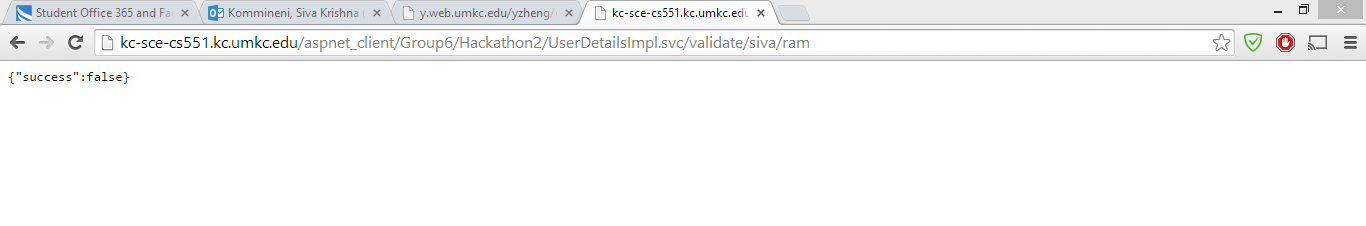


<http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/Hackathon2/UserDetailsImpl.svc>

This service is used for the user validation check and the registration of the user details in the DB

The below screens shows the validation showing success or failure whether the username and password of a particular uses matches or not.





<http://kc-sce-cs551.kc.umkc.edu/aspnet_client/Group6/ResumeService/Service1.svc>

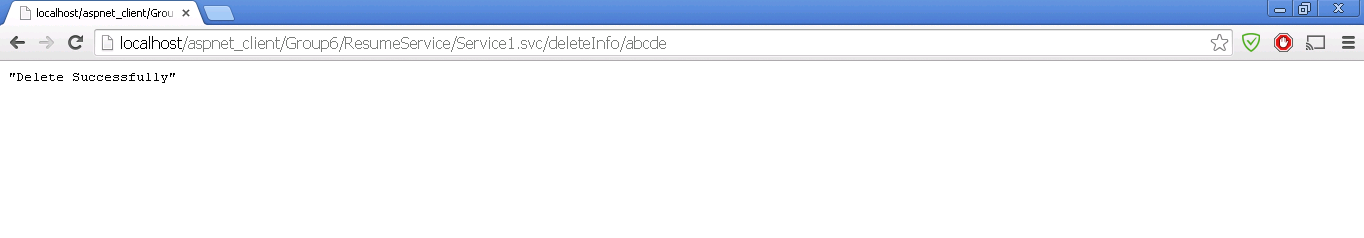
Data is submitted and stored in the DB using the remote service



We can insert and delete the data in this resume activity.

This is useful for building a resume and submitting it to the job portal in a precise manner.

All these services are hosted on the UMKC Server in project group folder and DB is designed accordingly.



1. **Existing services/APIs**

We used various existing API’s from the Job search portals like search by using various parameters.

We implemented searching jobs based on country codes and names which is very useful for the user and easily track them.

**Job API**

[**http://api.careerbuilder.com/Search/jobsearch/jobsearchinfo.aspx**](http://api.careerbuilder.com/Search/jobsearch/jobsearchinfo.aspx)

**Places API –**

[**https://developers.google.com/places/documentation/?csw=1**](https://developers.google.com/places/documentation/?csw=1)

Besides this we used Weather and Instagram Services for the comfort of the user in tracking the location and the climatic conditions.

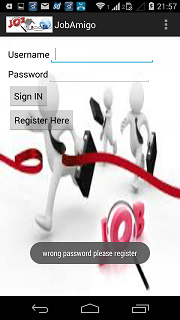
**Implementation of User Interface:**

The entire implementation of the UI is shown on an Android device which is flexible for the Android user. As emulator loading is time consuming. The whole process is clearly documented below n the form of screen shots and explained clearly.

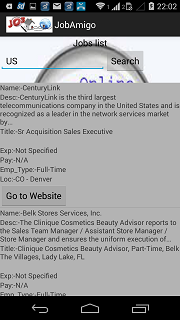
The screen shots are arranged in a sequence manner of operations based on the services used in the application.

**Documentation:**

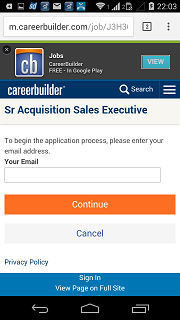
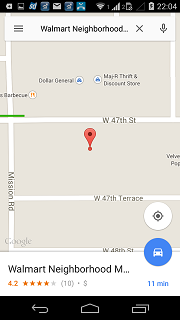
**Screen Shots of Home page, Login and Sign up with authentications and validations using the services hosted on the UMKC Server:**

**  **

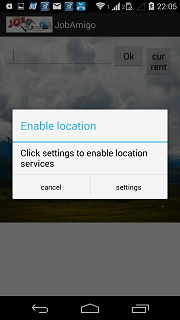
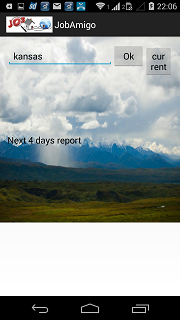
**Screen Shots of existing Job Search API and Implementation and Receive SMS Alerts when Applied for a particular position in a company:**

**  **

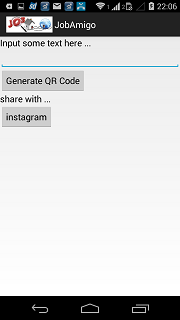
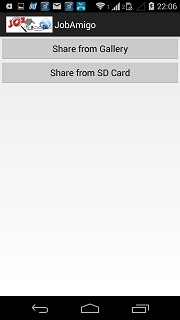
**Screen Shots of Google Map Services used for finding the Routes for the user**

**  **

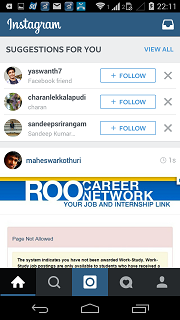
**Screen Shots of Weather Services used for Weather prediction of a location**

**  **

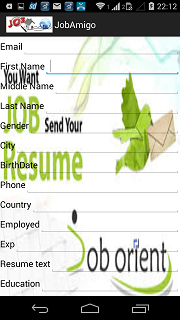
**Screen Shots of QR & Instagram Functionalities for Picture Sharing and Scanning**

**  **

**Sharing from the required place to the friends and networks**

** **

**Screen Shot of Resume Services hosted in UMKC Server and final phase of resume submission**

** **

This is the final page of the application which says that the job search for a particular user will end finally after selecting to the applied position.

**Project Management:**

In addition to the tasks defined in the previous increment for the third increment. We included the following tasks as it is the final increment for the project.

* Development of the Services for the user validation and registration. (Suresh & Ravisha)
* Implementing the resume service in a precise manner for job application. (Sarath & Suresh)
* Generating SMS alerts to the User once he applied to particular position.(Siva Krishna & Ravisha)
* GUI animations and design for all activities. (Ravisha)
* DB creation and validation for all the services created. (Sarath & Siva Krishna)
* Hosting of the services on to the UMKC server. (Sarath)
* Integrating all the activities to common platform. (Siva Krishna)
* Final GUI and data model testing. (Suresh)

All the above mentioned tasks are equally shared among all the team members in same weightage. It is clearly explained in the Scrum do. The link for the work sharing and the stories allocated are as follows.

* <http://www.scrumdo.com/projects/project/job-amigo/summary>

Some of us worked on GUI for few days and on services and development phase for the next few days. We shared work equally and each team member contributed in the maximum effort to complete the tasks allocated to them

In this way all of the 4 team members went through all the development life cycle of the project from Initial design to the final GUI testing and bug fixing.

**Future Work:**

* Implementing the Employer login and helps in guiding the user to the exact position
* Developing resume building and feedback system
* Tracking the status of Job application once the application process is completed by the user.

**Issues/Concerns:**

* Faced issues while generating the validation checks and developing exact DB for the user.
* Resume building and introduction of employer into the portal is really challenging.

**Deployment:**

The entire project repository is deployed in GitHub and the link is as follows

<https://github.com/sk7x9/CS551-Project>

There is a video production for the entire project which will be shared in the final report submission. The entire idea and implementation is recorded and shared on YouTube.