CS5200: DATABASE MANAGEMENT SYSTEMS

Project Report

By
Group Name: MahantMPereiraR
Roydon Pereira
Malhar Mahant

Topic: Job Search and Application Portal

Description

We intend to create a job search portal, that allows a *User* to register or log in as job *Applicants*. Another role for the *User* is *Recruiter* that works for an organization. The *Recruiter* can create *Job* postings on behalf of the *Organization*. An *Organization* and each *Job* also have a *Location* associated with them. *Applicants* can view all job listings on the web portal. Users can also search, and filter based on various attributes of the *Job*. The web portal allows *Applicants* to create *Applications* to apply for a *Job*. *Documents* related to an *Application* are also stored in the database. Each user can do the following actions:

- Applicant
    1. Log in on the web portal
    2. Search all job postings, with features like filtering based on location, position. etc.
    3. Apply to a job posting by creating an Application
    4. View/Delete their Applications
- Recruiter
    1. Log in on the web portal
    2. Create job postings
    3. Update job postings
    4. View applicant applications for their job postings
- System admin
    1. Add new Organizations (It is an invite-only web portal for organizations)

Software & Technical Specifications

Database: MySQL
Back-end: NodeJS
Front-end: HTML, CSS, Javascript
Software: MySQL Workbench, Git, Github

CS5200: DATABASE MANAGEMENT SYSTEMS

README

Prerequisites: Back-end => NodeJS, Database => MySQL Server

Setup steps:

1) Install NodeJS & NPM: To install Node.js and NPM, use any of the official Node.js installers provided for your operating system from Node.js (nodejs.org)
2) Run the installer. That's it, you have installed Node.js. Now check the version using the command node -v && npm -v to verify Node.js and NPM has installed correctly
3) Extract the application source code to a folder. Navigate to the directory that contains 'package.json' file and run 'npm install' in a terminal to install necessary dependencies for the application. Verify all dependencies have installed by checking the logs and a new file `package-lock.json` should be generated in the directory.
4) Now open MySQL Workbench and import the provided dump into your database.

Starting the Application

1) In the root directory of the source code, open db.js file and update the username and password of your local database into the appropriate fields.
2) To start the application, navigate to the directory that contains 'package.json' file and run 'npm start' in a terminal to start the application.
3) Open a browser window and navigate to http://localhost:3000/ to view the application web page.

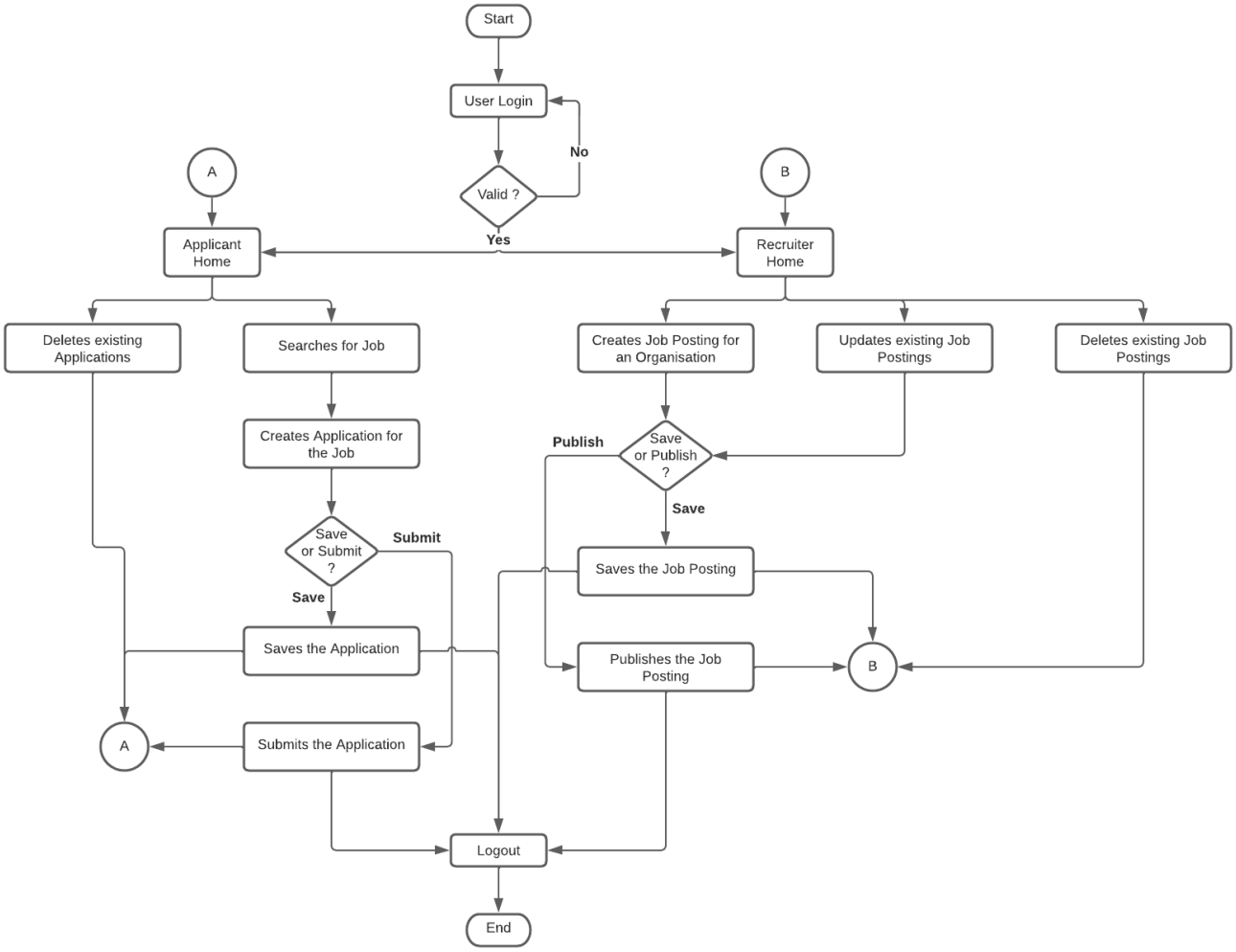
Using the Application

1) We have 2 user roles, a recruiter and an applicant
2) Here are the credentials for recruiter role: ck@kforce.com , password: abc12345
3) Here are the credentials for applicant role: dr@gmail.com , password: abc12345
4) There are multiple users added in the database which can be used to test different functionalities.

User Interaction

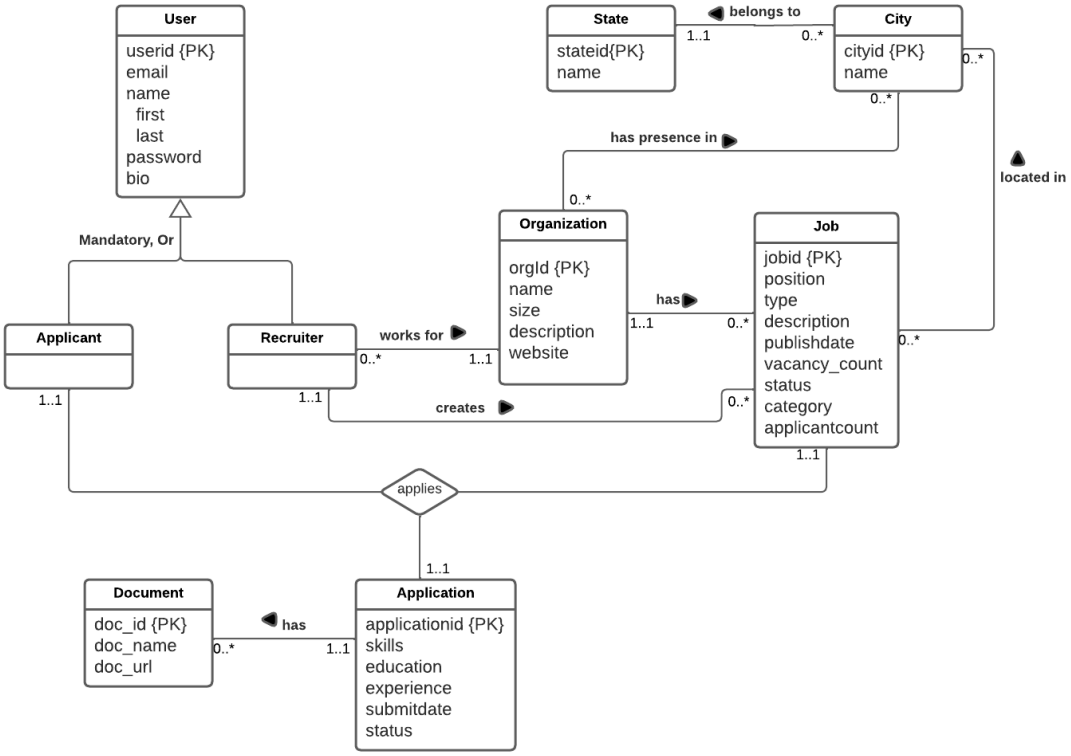
1) The user can log into their account with an email and password which can result in a valid or invalid result.
    1. If Valid, the user is sent to their Home screen based on their role.
    2. If Invalid, the user is asked to reenter the credentials.
2) If the user is of Applicant role, he/she can perform the following operations:
    1. Search jobs based on location, position.
    2. Apply to published job postings by filling in the required details.
    3. Attach documents like Resume, Cover Letter, etc. to their applications.
    4. Submit the unsubmitted job applications.
    5. View/Withdraw submitted applications.
3) If the user is a Recruiter, he/she can perform the following operations:
    1. Create job postings.
    2. View all job of their postings.
    3. View applications to the job postings.
    4. Update/Publish/Delete existing job postings.
4) Users can then log out from the website.

Flowchart

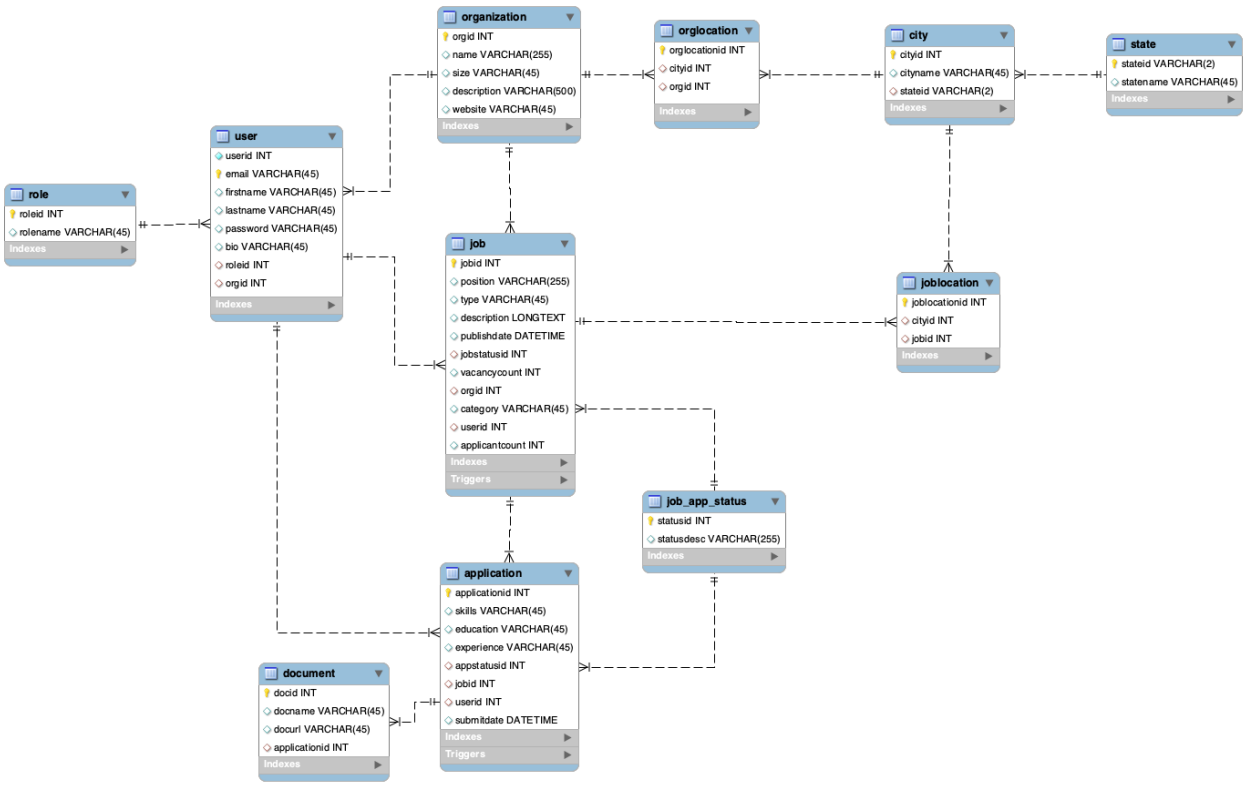

Start
User Login
No
Valid ?
Yes
A
B
Applicant Home
Recruiter Home
Deletes existing Applications
Searches for Job
Creates Job Posting for an Organisation
Updates existing Job Postings
Deletes existing Job Postings
Creates Application for the Job
Publish
Save or Publish ?
Save
Save or Submit ?
Submit
Save
Saves the Job Posting
Saves the Application
Publishes the Job Posting
B
A
Submits the Application
Logout
End

CS5200: DATABASE MANAGEMENT SYSTEMS

UML



Logical Design



Lessons Learned:

- Due to the complex nature of expected operation results, we had to write queries using loops. Maintaining atomicity by performing error handling in such database objects helped us gain in-depth knowledge of how the database system works.

- We had to change the approach of maintaining location details for different objects as our initial approach had introduced duplicate data mapping.

- The domain for Job Search services could include several additional features which we realized after we started working on the Project. However, these features would increase the scope and size of the project significantly and hence could be implemented in future.

Future Work:

- Planned Uses:
    - Store multiple Job postings at a single location.
    - Ability to Update Job Postings as per requirement.
    - Provides smaller company access to larger pool of applicants that match their requirements.
    - Provides applicants access to larger pool of jobs that match their profile

- The following features could be added in this project in the future:
    - User Registration
    - Ability to Save Job Applications as Drafts and resume later.
    - More comprehensive Search functionality based on multiple fields present in an application and job posts.
    - Improvements in the UI to provide functionality to map multiple locations to a job while creating a Job Post.
    - Database tables like State, City and Organization could be made more robust to include larger set of existing data.
    - A new login page that provides special access privileges to the Organization.