# Busybee: Job Application Management System

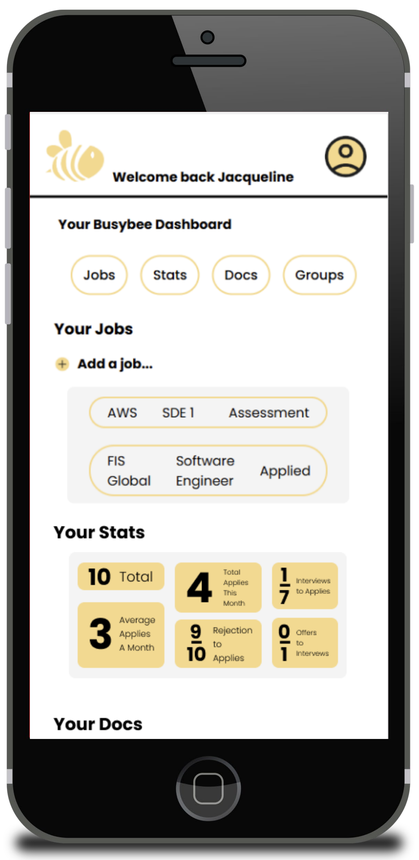
Mason Krause, Mason Luna, Jacqueline Justice, Jason Moss

Dept. Of Computer Science and Information Technology

## Abstract

This project will improve the job-seeking experience by providing a dashboard where job seekers can monitor and manage their job applications. Many platforms are involved when looking for a job (Indeed, LinkedIn, Monster, Glassdoor, etc.), and a job seeker is likely using some or all of them. Our project will help users track all the applications they've completed across platforms in one consolidated list. They can keep that list up to date as the statuses of applications change.

The dashboard will be divided into a few main sections. One section will allow the user to store a copy of their resume and cover letter. Another section will contain all applications submitted throughout the job search, with a filtering/sorting mechanism to organize applications as desired (by company, date applied, etc.). The final section will show some statistics about the current job search, such as the number of applications per month, the application-to-interview ratio, and the interview-to-offer ratio. These tools will empower users to structure their job search in a deliberate, efficient, and data-driven way.




## Technology

The system is made up of the following key elements: the frontend, backend, and database. These elements work together to provide users with a structured job management experience. The frontend is built using React.js, HTML, CSS, and TypeScript, enabling a modular, maintainable interface with dynamic rendering and smooth routing through React Router. It communicates with the backend via the Supabase JavaScript client, sending queries and receiving JSON response objects. Supabase, which serves as both the backend service and cloud database host, handles authentication, database operations using PostgreSQL, and file storage through Supabase Storage. This integration allows for real-time updates, secure session management, and efficient handling of user data and files throughout the application.

## Frontend Design

Busybee's frontend is developed with React and TypeScript, using a modular structure that makes the interface dynamic, scalable, and easy to maintain. The main user experience is centered around an intuitive dashboard that allows smooth navigation through the app's features. Style is done through React.js, HTML, and CSS. Authentication and session management are handled by Supabase. Users can also request email or password changes directly from the login form using Supabase's built-in services. App.js serves as the entry point of the app and uses React Router to manage navigation. It routes users from login to the main dashboard, which loads several child components: navigation, header, footer, and a dashboard view. The dashboard view displays one of five sections based on user selection: jobs, groups, stats, documents, or main. Three static pages—Contact Us, FAQ, and About the Team—are accessible through footer links and represent the only navigation outside the main dashboard. Apart from these, Busybee functions as a single-page application, with all core interaction taking place within the dashboard.

## Backend Design

Busybee's backend is powered by Supabase, using PostgreSQL to store structured data for users, jobs, groups, and documents. When a user logs in, all related data is fetched from the database and stored in JobModel, GroupModel, and DocumentModel instances, which keep both local and remote data in sync as users interact with the app. Users can log in, register, manage job applications, organize groups, upload documents, and view statistics. Jobs and groups can be created, edited, and deleted, while documents can be uploaded, viewed, or removed, though not edited. Documents can exist independently or be linked to specific job entries, depending on whether a job_id is assigned. The dashboard serves as the main user interface, offering quick access to all key features. Jobs can be filtered and grouped, documents are stored in Supabase Storage, and groups allow users to organize job searches by category. The statistics system provides insight into the user's progress by calculating metrics like total applications, monthly activity, and conversion ratios at login or after changes to job data. By combining a responsive frontend with real-time data handling on the backend, Busybee helps users track their job search with structure and clarity. Whether adding a job, uploading a resume, or checking interview ratios, each feature is designed to support users in staying organized and improving their outcomes.

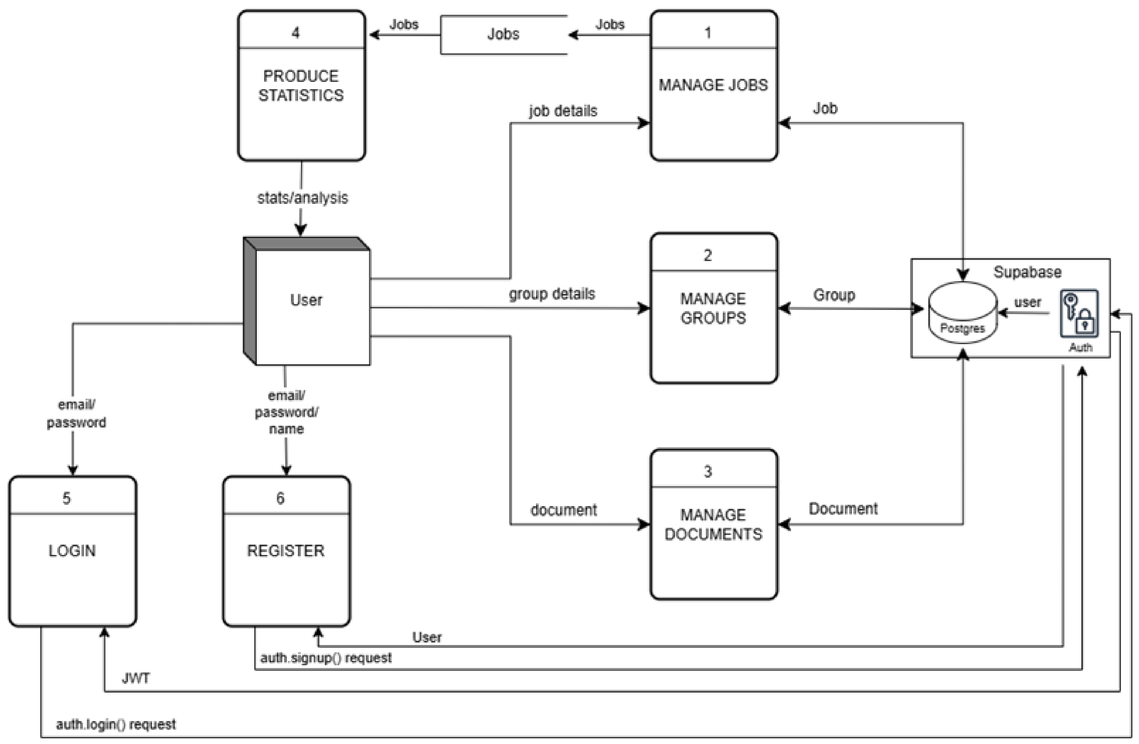



**Figure 1:** **Shows** the DFD0 of Busybee, outlining how users interact with the system. Users can log in, register, manage jobs and groups (create, view, update, delete), handle documents, and view statistics. This diagram illustrates the main data flows between the user and the system.

## References

1. Supabase - https://supabase.com/docs
2. PostgreSQL - https://www.postgresql.org/docs/
3. React - https://react.dev/

## Acknowledgements

We would like to thank Dr. John Nicholson for her support of students in the College of Science, Technology, Engineering & Mathematics, and Dr. Leong Lee for his support of students in the Department of Computer Science and Information Technology.