JOB ANALYTICS 

4th Year Project

Functional Specification

Name: Daniel O'Sullivan

Student Number: 14307881

Supervisor: Suzanne Little

0. Table of contents

1. Introduction

Job Analytics is a web application for employers and applicants to post and find jobs. Employers can post jobs on the app and applicants can apply for open positions. When an employer gets a CV from an applicant, the application will examine an applicant's CV and return how qualified that person is for the specific role and will also predict how well that person will perform in the role. When an applicant registers an account with the application, their CV will be parsed and mined, and jobs are returned that match their skills and experience. The idea behind this is to make applying and recruiting for an open vacancy easier for both parties by letting the application do all the work for them. All the application needs is a CV.

This idea was developed during my INTRA placement. I was working in a team that specialized in predictive analytics and this peaked my interest in this topic along with machine learning. Learning these topics made me start to think about my 4th year project. I wanted to make an application that would be useful to people and make a task or function easier to carry out and it's result more accurate.

1.2 Business Context

With much of hiring services being carried out online or by software recruitment tools, Job Analytics has the potential to be a great asset to this sector. From a business point of view, subscription services could be sold to companies to recruit new employees. For example, companies could pay a subscription depending on how much they want to use the application. They could pay for a monthly or yearly subscription or could pay for a certain amount of curriculum vitae examinations. For an employee, there could be a free to use and premium accounts. A free to use account can only let the applicant apply for a certain number of jobs a day or allow the applicant to use the application to examine their curriculum vitae only a certain number of time. A paid premium account would allow users to carry out these tasks as much as they want. There could also be advertising in the application for additional profit.

1.3 Glossary

Python: Is a widely used high-level, general-purpose, interpreted, dynamic programming language.

JavaScript: Is an object-oriented computer programming language commonly used to create interactive effects within web browsers.

HTML: Hypertext Markup Language is a standardized system for tagging text file to achieve font, colour, graphic and hyperlink effects on World Wide Web pages.

CSS: Is the language for describing the presentation of web pages, including colours, layout and fonts. It allows one to adapt the presentation to different types of devices, such as larger screens or printers.

Django Framework: Is a powerful and flexible toolkit for building Web APIs. It used to implement backend services in web applications.

ReactJS: Is a JavaScript library for building user interfaces. Used for front-end services.

Google Firebase: Is a realtime database that stores data using JSON (JavaScript Object Notation)

Elasticsearch: Is a search engine that provides full text search along with other features. It can also be used a database.

CV Parser: This is the tool that will examine an applicant's CVs and retrieve relative information.

Machine Learning: Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed.

2. General Description

2.1 Product / System Functions

- **Advertise a job:** Employers will be able to advertise jobs and set specific requirements for the role. A short form with brief questions will be automatically generated based on the constraints set by the employer that applicants will have to answer when they apply for the vacancy.

- **Apply for open vacancies:** Applicants will be able to apply for open vacancies that are advertised on the application.

- **Examine curriculum vitaes:** Both employers and applicants will be able to use this function. Employers will be able to parse and analyse an applicant's CVs to see if they fit the role in question. Applicants will be able to use the application to analyse their CVs to see what available jobs their skills and experience match.

- **Predict an applicant's performance:** The application will be able to predict how an applicant will perform in a certain role based on their CV and questions they answer in the generated form. This will be done using a dataset and a machine learning algorithm.

- **User creation:** Users will be able to create accounts based on whether they are an employer posting a job or an applicant looking for a new job.

- **Return information from CV examinations:** Employers will be able to see a percentage based on how qualified an applicant is for a role and how well they are expected to perform in that role. All applicants that apply for a role will be ranked based on the result returned from the examination.

2.2 User Characteristics and Objectives

The target user demographic ranges from people in their mid-20s to people in their early 50s. The user base will consist of employers looking to fill an open vacancy and applicants seeking a new job. The applicants will be people who are either already employed seeking a new job or recently unemployed and have previous work experience. The application aims to help people to find the preferred role that will aid their career. The application isn't aimed towards people looking for part time work.

Expertise with software systems will be varied depending on the user. Employers should be familiar with similar HR or recruitment software and applicants could be very experienced or have little to no experience with using web applications. From the user's perspective, the application's UI should be clear and simple to use. The goal of the application is to post a job

and recruit new employees with minimal effort from the user's side and allow users to apply for a job by just uploading your curriculum vitae and then clicking on the open vacancy the user wishes to apply for.

Desirable characteristics for an employer user, would be someone who is familiar with using web applications as well other software platforms. For an applicant user, it would be someone who is also familiar with web software and who will also have a properly formatted curriculum vitae. If a user didn't have such experience with such technologies, they should have a good understanding of following instructions and good patience. The employer characteristics seem to be more realistic to that of an applicant, as the demographic of applicants in terms of software skills can vary greatly. With an employer, the majority of the user demographic will be familiar with HR recruitment software or something similar.

2.3 Operational Scenarios

- **User Creation:** User can create accounts with the application based on if they're an employer posting an open vacancy or an applicant seeking a new job.

- **CV Examinations:** The application will be able to perform examinations on CVs and return scores based on the examinations. For employers, the examinations will be to see how qualified an applicant is for a role. For applicants, the examinations will be to see what posted jobs their skills and experience meet.

- **Predict an applicant's performance:** The application will be able to predict how an applicant will perform in a certain role. This will be done using a machine learning algorithm based on the information retrieved from an applicant's CV and their answers in the generated text forms.

- **Apply for a job:** Applicant users will be able to apply for jobs. When the user first creates an account with the application, they will be prompted to upload their curriculum vitae. The application will then examine their CV and return the jobs they best match based on their skills and experience. If they don't like the looks of the vacancies returned from the examination they can search for other ones.

- **Post open vacancies:** Employer users will be able to post jobs. They will be able specify requirements.

- **Automatically generate text forms:** Based on a job listing and its requirements, a text form will be generated for the applicant to fill out when they apply for the role. The employer will have the option to modify the text form if they do not find the generated one satisfactory.

2.4 User Scenarios

Sample Scenario 1
Margaret works in Human resources for a medium sized international courier company based in Ireland. She needs to find a new van driver for the monthly deliveries to France. She ideally wants a candidate with a minimum of 4 year's experience as a courier driver and fluent in French. Several applications are received for the vacancy all of which are ranked from best to worst candidates based on their score from CV examinations. The top candidate is Brian who meets the requirements in full. The reason he is ranked top because of his well-structured CV and the machine learning algorithm predicts he will have the best evaluation rate for the role. The second top ranked candidate is Steven, who does not meet the requirements fully as he only as 2 years' experience as a courier. However, he is fluent

in French and the prediction algorithm states he would excel in the role. Margaret can then decide who to contact for an interview based on the ranking of applicants.

Sample Scenario 2
Ben works as a software engineer and is looking for a new job. He has spent the last few years as a front-end developer and is proficient in JavaScript and frameworks such as ReactJS and Ember. He creates an account with Job Analytics and uploads his CV to the application. His CV is examined and posted jobs are returned based on his skills and experience in the software development sector. The majority of the jobs returned are front-end jobs because of his previous experience. Ben decides to apply for one job as a senior developer for a new start-up company. When he goes to apply for the job, he sees that he meets the requirements stated. He is then asked to fill out a short form that has questions related to his experience, skills and the role he is applying for. He also has the option to upload or input a cover later for the role. Once he has completed the form, he has applied for the job. He can then choose to apply for another vacancy if he wishes or wait to hear back from recruiters for the company he applied for.

Sample Scenario 3
Niamh is an accountant who is looking for a career change. She decides that she wants to work in a sales department as she feels that she's a good people person. She creates an account with Job Analytics and uploads her CV. From the first CV examination, the application returns accounting jobs because of her skills and previous experience. She decides not to apply for any of the jobs returned from the examination and searches for jobs in sales that have been posted to the application. She applies for a role in a sales team. Despite not meeting the requirements, the prediction algorithm could give her a good evaluation score based on her answers from the generated form and the information retrieved from her CV. Once she has completed the form, she has applied for the job. She can then choose to apply for another vacancy if she wishes or wait to hear back from recruiters for the company she applied for.

2.5 Constraints

- **Curriculum Vitae format:** The goal of the application is to predict a person's performance for role by analysing the job requirements, question form answers and the structure of an applicant's CV. Analysing the structure of the CV will be challenging, as there is a wide variety of templates.

- **Grammar of job requirements:** How I allow an employer define job requirements for a role could be problematic. Grammar could be constrained to just key words which would be a simple solution, but this could leave users feeling like they can't advertise a job the way they want or need to. Allowing the use of free grammar would require some thesaural software to check the format of the job requirements are correct.

- **Speed of examinations:** The speed of CV examinations will need to be quick and not compromise the accurateness of the examinations. The examinations will have to find the job requirements specified, analyse the structure of the CV and find any other relative information that relates to the vacancy in question. There may have to be a compromise between speed and accurateness of score results from examinations.

3. Functional Requirements

User Creation

- **Description:** User can create accounts with the application based on if they're an employer posting an open vacancy or an applicant seeking a new job.

- **Criticality:** Creating a user profile is essential. An anonymous approach would mean applicant users would have to upload their CVs for every time they open the application and employer users wouldn't be able to track applicants for posted jobs. This function is an essential part of the application.

- **Technical Issues:** With multiple users comes the necessity for protection. It is important that users can only access their data. Users should also be able to delete their accounts if they please.

- **Dependencies:** One dependency could be how much storage there is in the database for all users. This shouldn't be an issue with most databases offering substantial amount of storage. Security for users will also be needed.

CV Examinations

- **Description:** The application will be able to perform examinations on CVs and return scores based on those examinations. For employers, the examinations will be to see how qualified an applicant is for a role. For applicants, the examinations will be to see what posted jobs their skills and experience meet.

- **Criticality:** The purpose of this application is to make the hiring process for both the applicant and the recruiter easier. The CV examinations are the main feature of this application and will make users fell like that the work is being done for them. Therefore, implementing this functionality is of the upmost importance.

- **Technical Issues:** The examinations will need to be quick and accurate. The Elasticsearch framework will be used to implement this functionality. The examinations must not only just return the specified requirements but retrieve any other necessary information that is relative to the role in question. The examinations must also be accurate for applicant users when they first scan their CVs to find jobs they match.

- **Dependencies:** Structure of curriculum vitaes and grammar specified in job requirements.

Predict an applicant's performance

- **Description:** The application will be able to predict how an applicant will perform in a certain role. This will be done using a machine learning algorithm based on the information retrieved from an applicant's CV and their answers in the generated text forms.

- **Criticality:** This feature will allow employers to get an idea of how an applicant will perform in a role. It is important that these predictions are accurate and consistent. If the algorithm is not implemented properly, applicants who should be scoring high could not be and therefore the right candidate will not be picked for the role. Employer users should feel comfortable and confident knowing that this functionality will perform to its full potential.

- **Technical issues:** The algorithm will be using a dataset of employee evaluations based on attributes such as years at previous company, salary (high, medium or low), average monthly working hours and more. The algorithm will be created using the clustering technique. As mentioned previously, the algorithm must be consistent. Two candidates who have similar attributes should return similar scores.

- **Dependencies:** The information needed for the prediction will mainly be retrieved from the answers filled out in the generated text form. Information will also be retrieved from the CV examinations. The algorithm will be written in python and will be easier to implement into the application with the use of the Django backend framework.

Apply for a job

- **Description:** Applicant users will be able to apply for jobs. When the user first creates an account with the application, they will be prompted to upload their curriculum vitae. The application will then examine their CV and return the jobs they best match based on their skills and experience. If they don't like the looks of the vacancies returned from the examinations they can search for other ones.

- **Criticality:** This feature is essential for applicant users in particular. If this feature is not implemented there would be no need to have applicant users. The application would just be a tool for employers to parse CVs they received from other job posting websites and applications, which could be one way of designing the application. Despite this, implementing this feature is still crucial as without it, applicants wouldn't fill out the generated text forms that are needed for the prediction algorithm. CV examinations would have to do a lot more and CVs would have to be in a certain format and include specific information for the application to perform some of its other functionality.

- **Technical issues:** Posted jobs must be visible for applicant users. When searching for jobs, the application will be able to return jobs based on key words and what industry the jobs are in. When applying for job the system must keep track of who applied for job, their CV and information from the generated forms and information retrieved from examinations.

- **Dependencies:** How much information a user gives for a posted job will be a dependency as this will be needed for searching for open vacancies. How the system will handle an applicant's CV and application form when they apply for a job will also be a dependency.

Post open vacancies

- **Description:** Employer users will be able to post jobs. They will be able specify requirements.

- **Criticality:** This functionality is critical only if the “Apply for a job” feature is implemented as although this feature is essential for applicant users, it is different for the employer users. Without this functionality implemented into the application, employers would have to post open vacancies using other websites and applications. However, they could still use Job Analytics for CV examinations using CVs they received from other job posting software. They would just need to quickly specify the requirements. However, as specified previously in this document, implementing this feature is a dependency for other features for this application. Thus the “post open vacancies” functionality is essential also.

- **Technical issues:** This will be implemented using a job board package within the Django framework. Job posting will allow the employer to specify how much information is given for a job.

- **Dependencies:** Dependent on Django framework for implementing job board package.

Automatically generate text forms

- **Description:** Based on a job listing and its requirements, a text form will be generated for the applicant to fill out when they apply for the role. The employer will have the option to modify the text form if they do not find the generated one satisfactory.

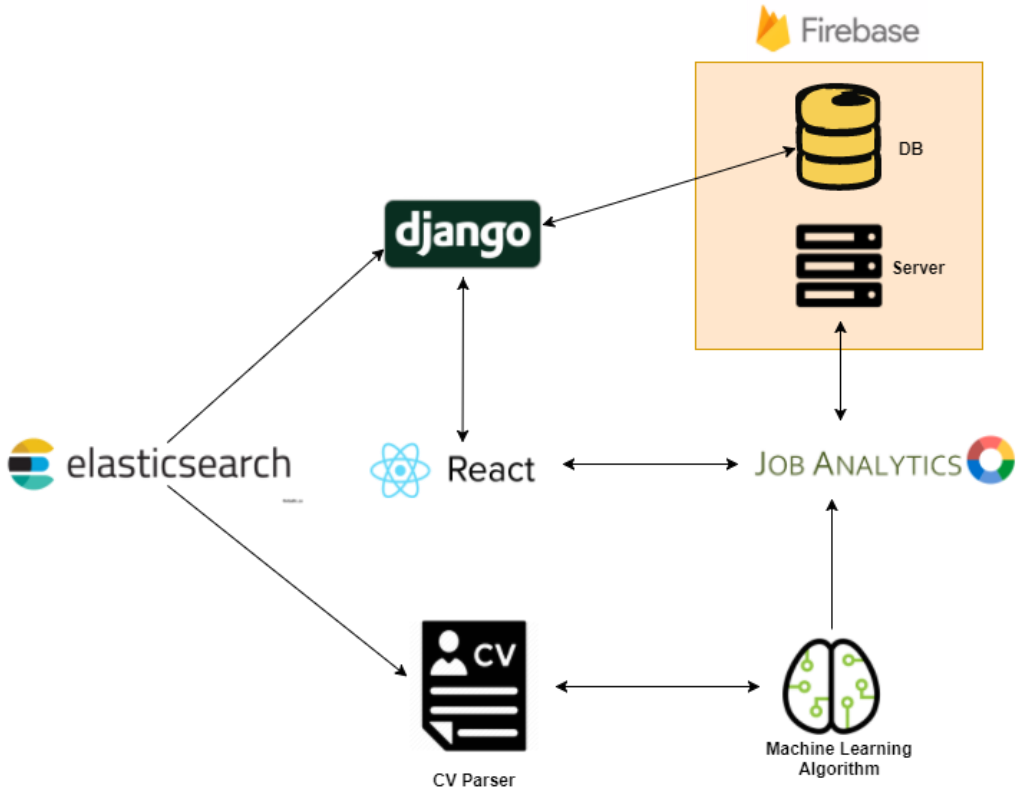
- **Criticality:** This functionality is critical for the machine learning prediction algorithm. The dataset used for the algorithm needs information that may or may not be on a curriculum vitae so the answers on the forms from applicants will provide this information. Implementing this feature will also make it easier for employers posting jobs as the form are automatically generated based on the job in question.

- **Technical issues:** The forms will be generated using the Django framework. Django allows forms to be generated based on the models used in the framework.

- **Dependencies:** Dependent on the Django framework and information from job postings.

4. System Architecture

4.1 Component Architecture



This diagram shows the technologies I will be using to implement my application and how each of them interact with each other.

- **Google Firebase** can be used as either a real-time database and/or a hosting web server. The application will be using both as the firebase platform is both powerful and easy to implement.

- **The Django Web Framework** will be used to implement backend services. It will manage services for the application such as connection to the database, job posting board, notifications for users and more.

- **ReactJS** will be used for the front-end of the application. It will control the view of the application and will be connected to the back-end.

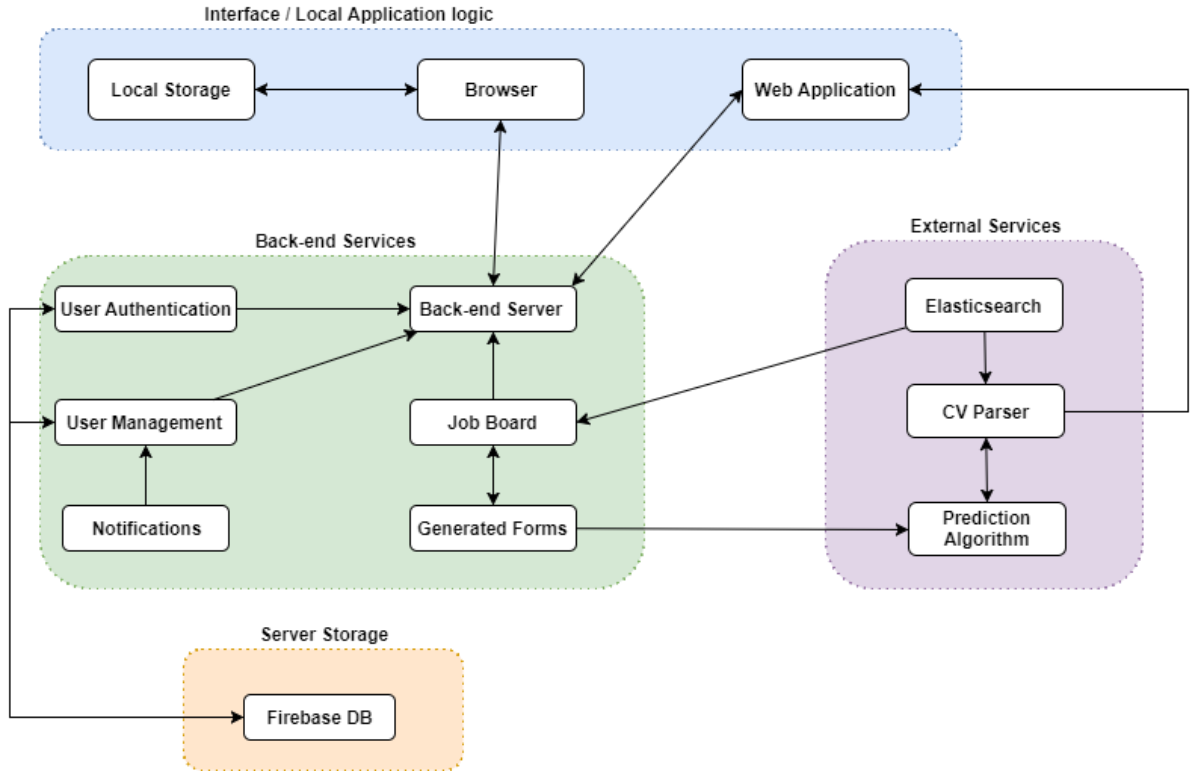
- **Elasticsearch** will provide the search and information retrieval for the application. It will be used for searching jobs on the job board and will also be used in making the CV Parser.

- **The CV Parser** will be made using Elasticsearch and will retrieve information for the machine learning prediction algorithm to use.

- **The machine learning algorithm** will be made using separate libraries and packages that aren't shared with other components of the application. It will use information retrieved from the CV Parser for its predictions.

All the above components will be integrated together to form the Job Analytics web application. The application will be hosted on a Google Firebase server.

4.2 System Architecture



The above diagram shows how the different components and services interact with each other.

The interface / Local Application logic layer consists of the web application, a browser and browsers local storage. Aspects of application data will be stored in local storage for quick use. The web application is created using a combination of components and external services. The external services are the CV Parser, the prediction algorithm and Elasticsearch. The prediction algorithm uses information retrieved from CV examinations and from answers filled out by applicants on the generated forms. The scores for applicants who applied for a role is calculated from both the CV examination results and the prediction algorithm. The CV Parser will be created using the Elasticsearch component. Elasticsearch is also used in the back-end of the application.

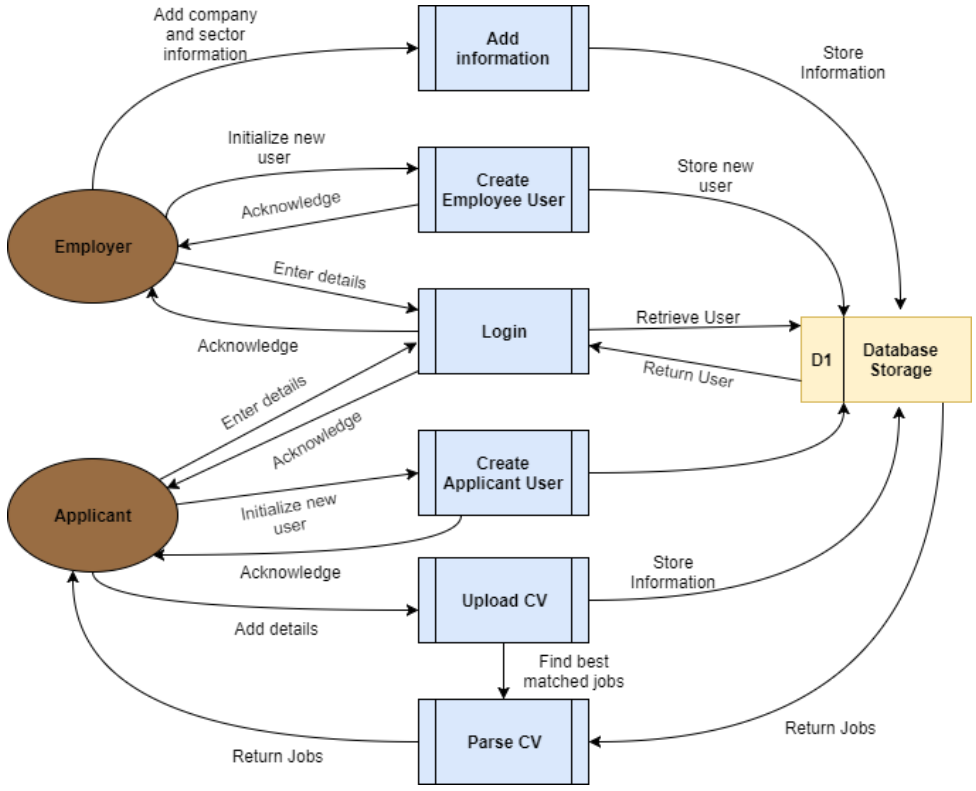
The back-end services include components such as the back-end server, user authentication and management, the job board, notification system and the generated forms. The job board, notification system and the generated forms are all implemented by the Django framework. The generated forms are created using information specified in posted jobs on the job board. Notifications are handled in the user management component. User authentication and management will get and store data in the Firebase Database. All the back-end components are implemented in the Job Analytics application which is hosted on the back-end server. A browser connects to the back-end server to display the application on a client's machine.

5. High-Level Design

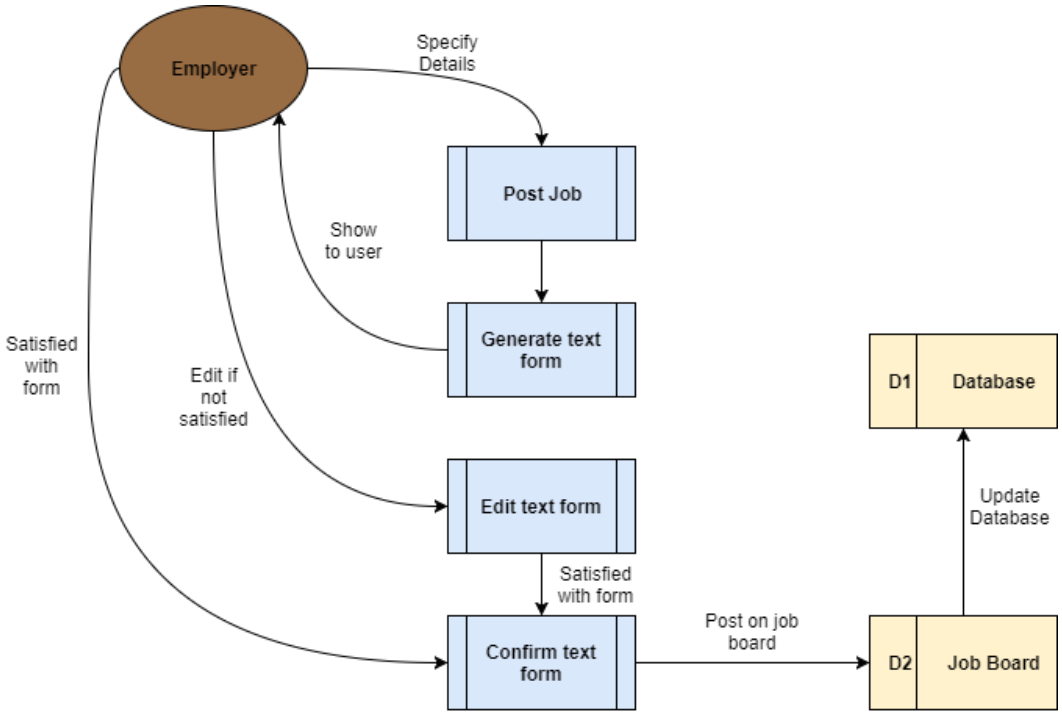
Below are Data flow diagrams which illustrate how some of the applications functions work.

5.1 Data Flow Diagrams

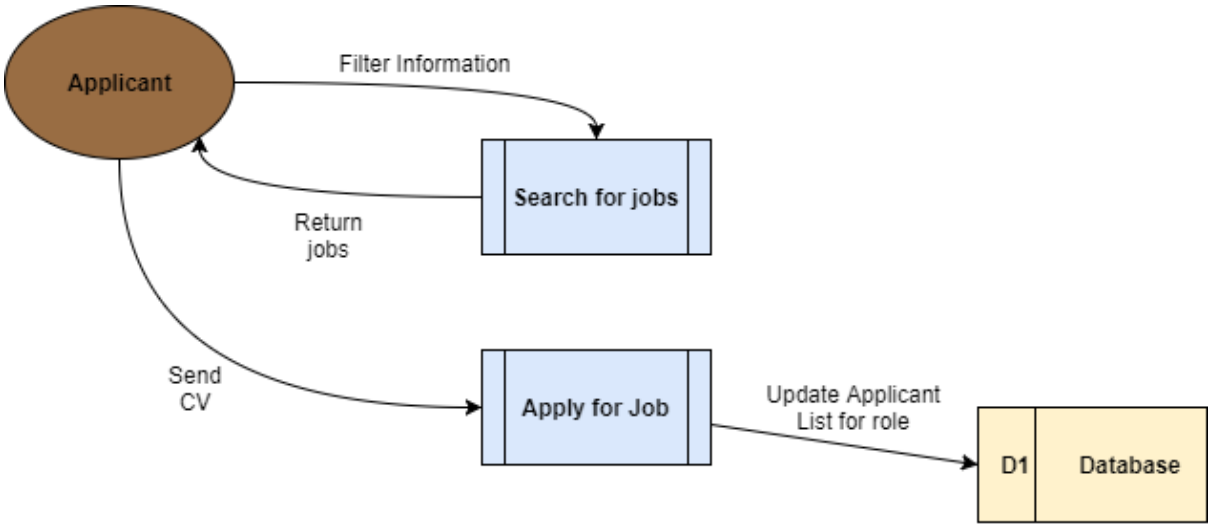
Login / User Creation



Posting a job

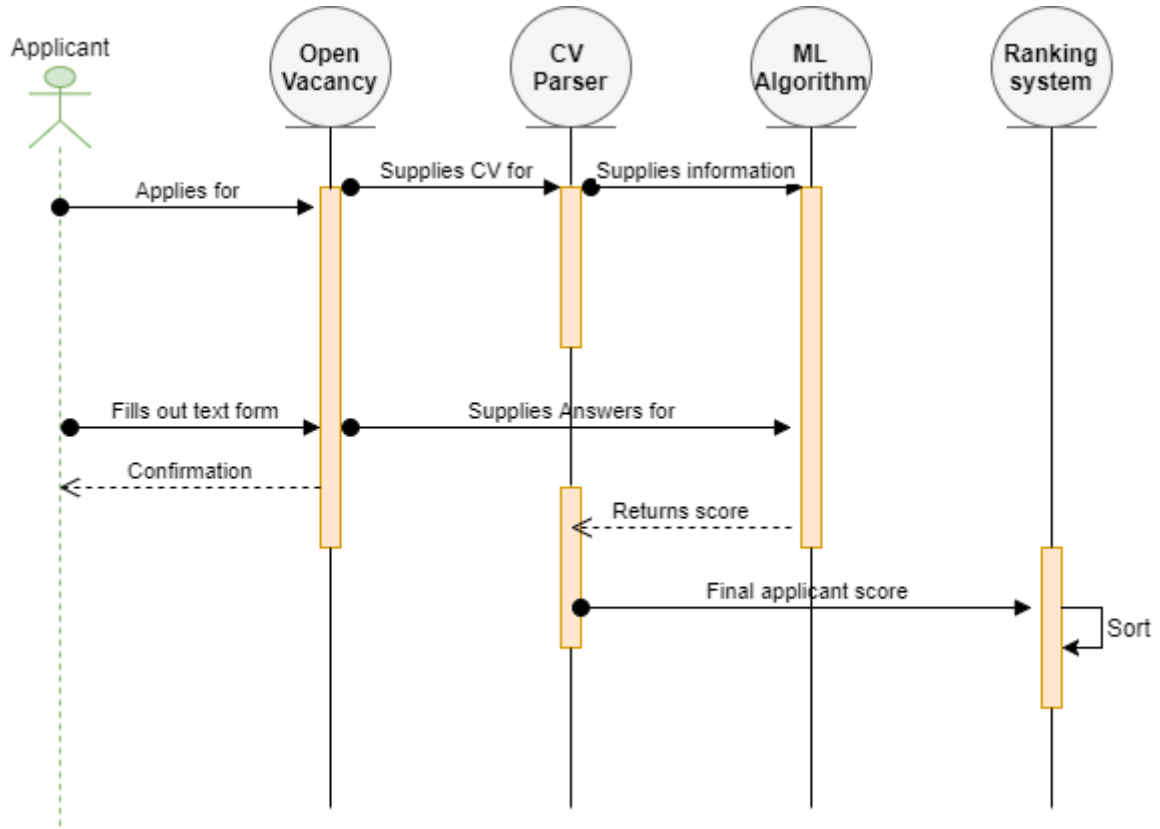


Applying for a job



5.2 Sequence Diagram

Processing a job application

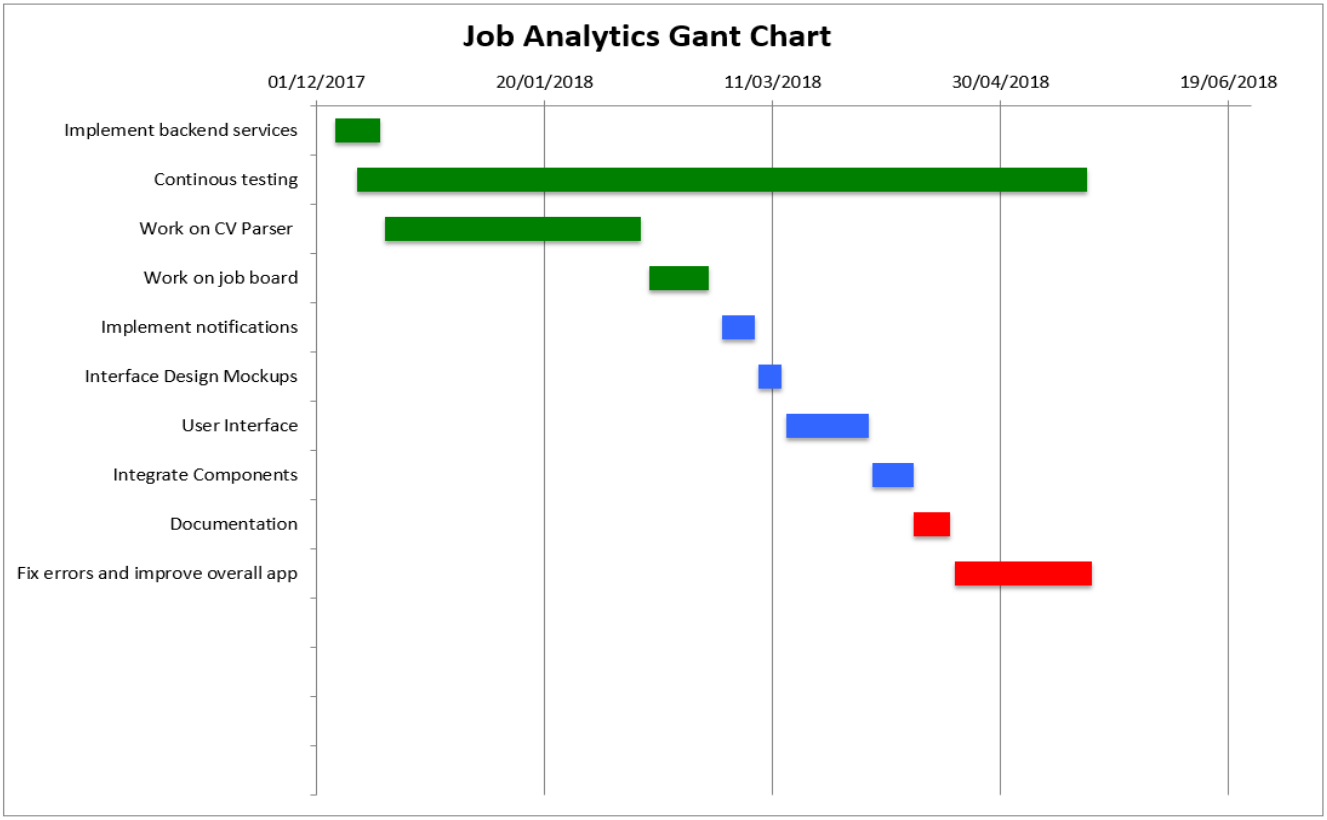


The diagram above shows the sequence of interactions between components of the application when a job application is being processed. It contains the following steps:

- The applicant applies for the role by first sending his/her CV.
- The applicant then fills out the generated text form to complete the application.
- A confirmation is sent to the applicant letting them know they've successfully applied for the vacancy.
- The CV is then supplied to the CV parser for examination and form answers supplied to the machine learning prediction algorithm.
- The role evaluation score from the prediction algorithm is then passed to the CV parser so the two scores (Machine learning score and CV examination score) can be combined to give a final applicant score.
- The final applicant score is then passed to the to ranking system of the open vacancy to be added in for the employer to see.
- The ranking system is then sorted to fit in the new applicant's score.

6. Preliminary Schedule

Task Name	Start	End	Duration (days)
Implement backend services	05/12/2017	15/12/2017	10
Continuous testing	10/12/2017	20/05/2018	160
Work on CV Parser	16/12/2017	10/02/2018	56
Work on job board	12/02/2018	25/02/2018	13
Implement notifications	28/02/2018	07/03/2018	7
Interface Design Mockups	08/03/2018	13/03/2018	5
User Interface	14/03/2018	01/04/2018	18
Integrate Components	02/04/2018	11/04/2018	9
Documentation	11/04/2018	19/04/2018	8
Fix errors and improve overall app	20/04/2018	20/05/2018	30



Please keep in mind that these are estimates, and that dates and durations are likely to change.

7. Appendices

7.1 Software referenced

- **Django:** https://www.djangoproject.com/
- **ReactJS:** https://reactjs.org/
- **Elasticsearch:** https://www.elastic.co/
- **Google Firebase:** https://firebase.google.com/