**Team “Back Five”**

**Riley Horton, Darryl Murray, Hammad Khan, Camden Rodgers, Robert Jones**

## INTRODUCTION

Our team chose to design a product that aimed at helping both applicants and companies in the early stages of the hiring process. We concluded that the best way to do this would be to incorporate AI into the process to help in two big areas: matching applicants to prospective jobs and the resume/applicant profile. This would be done by having applicants submit a resume and going through a mock interview. This information would be put through an algorithm and given a score based on qualifications, skills, and responses in the interview. The applicant would then be given feedback on places they excel in and where they fall short. The companies looking to hire them would also be able to see the score given, speeding up the process of looking through a large number of applicants. Our software aims to have the following qualities:

- AI gives notes (pros-cons) of applicant.
- Use web cam to record the mock interview and add to the applicants file sent to the employer.
- AI gives feedback to help applicant improve ("score").
- AI takes/gives notes as to what employers are looking for/have hired for in the past. Or what they may be actively looking for (Data driven feedback).
- AI matches jobs to applicants

## REQUIREMENTS

1. User Requirements:

    1.1. Functional Requirements:

        1.1.1. Upload Resume- Users must be able to upload the resume to the system.

        1.1.2. Job Matching - Users expect the application to match them with job positions that align with their skills and qualifications.

        1.1.3. Improvement Suggestions - Users desire feedback on areas they can improve upon to enhance their interview skills.

        1.1.4. Internet Access Requirement- Users should have access to the internet and its functionality.

1.1.5. Webcam Integration - Users want the application to use their webcam to record mock interviews and attach them to their application files.

1.2. Non-functional Requirements:

1.2.1. Usability - The application should have an intuitive user interface to facilitate easy navigation and interaction.

1.2.2. Performance - The application must provide timely feedback and match users with job positions efficiently.

1.2.3. Privacy - User data, including interview recordings and feedback, must be securely stored, and protected.

1.2.4. Reliability - The application should be available and reliable for users to access whenever needed.

2. System Requirements:

2.1. Mock Interview Module - The system must have a module to conduct mock interviews and generate questions.

2.2. Matching Algorithm - The system needs an algorithm to analyze user profiles and job descriptions for effective job matching.

2.3. Scheduler Module: The system should include the scheduler module to facilitate the scheduling of interviews between users and potential employers.

2.4. AI Feedback System- The system must incorporate AI capabilities to provide detailed feedback to users after mock interviews.

2.5. Data Analysis Module - The system requires a module to analyze hiring trends and employer preferences for data-driven feedback.

2.6. Webcam Integration Feature- The system needs functionality to integrate with users' webcams for recording mock interviews.

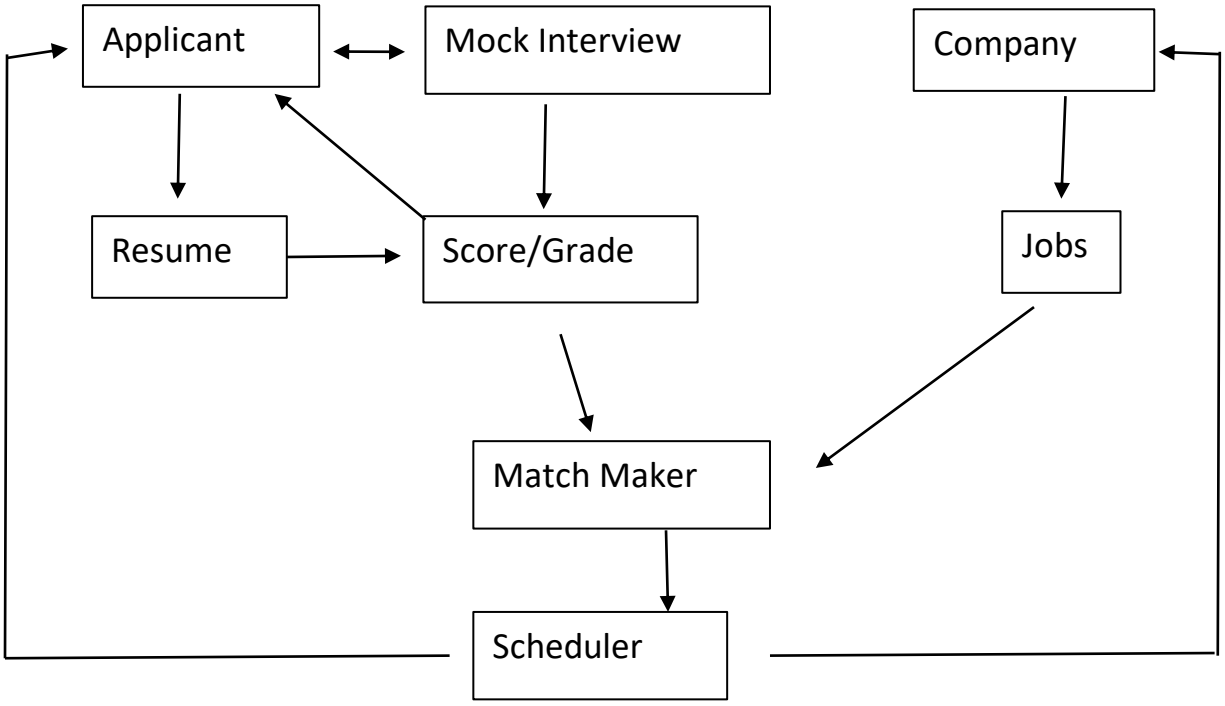
2.7. Secure Data Storage - The system must ensure secure storage of user data, including interview recordings and feedback.

2.8. User Interface - The system should have a user-friendly interface for easy user interaction and navigation.

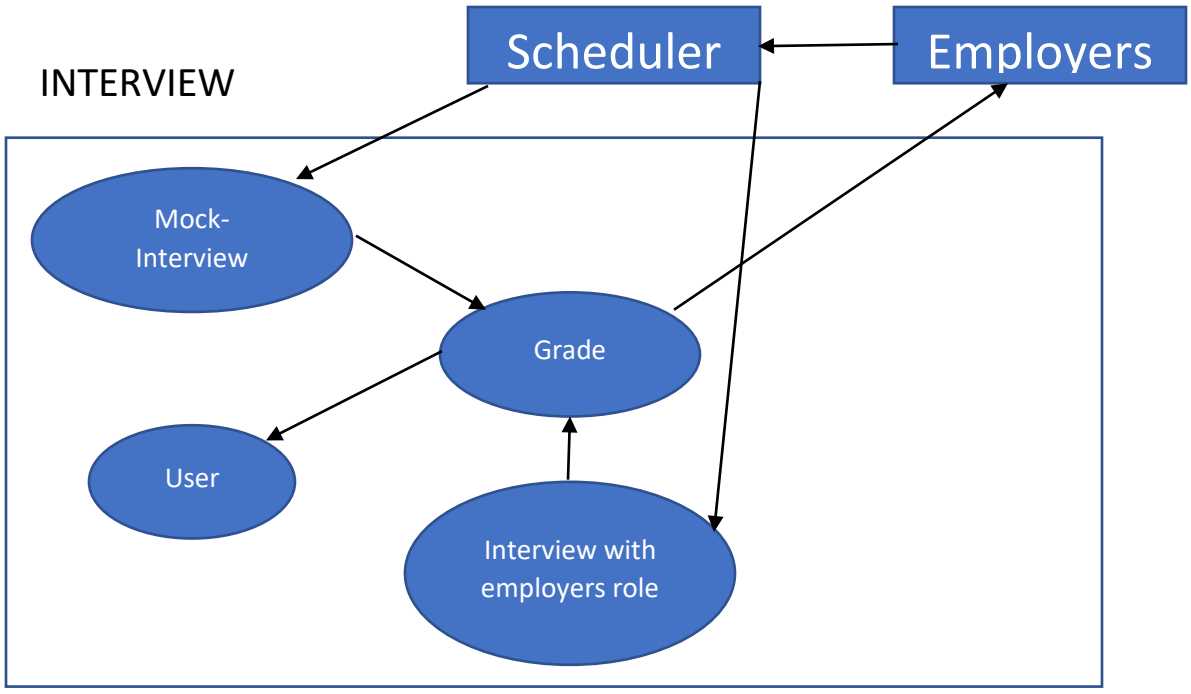
2.9. Performance Optimization - The system must be optimized for performance to provide timely feedback and match users with job positions efficiently.

2.10. Privacy Measures - The system should implement privacy measures to protect user data and ensure confidentiality.

## ARCHITECTURE




## DETAILED DESIGN




**TEST CASES**

Test Case 2.1.1:
Name: Mock Interview
Description: Test the AI Tool's ability to generate applicant's score.
Test Input Data: Responses

.1 Initiate a mock interview session with AI.
.2 Initiate Webcam usage.
.3 Respond to the first three interview questions by repeating the questions.
.4 Do not provide any responses as the interviewee for the rest of the questions.

Expected Results: AI should provide a low score to the applicant and provide feedback on the lack of responses. Interview should be recorded.

Test Case 2.3.2:
Name: Scheduler Match
Description: Tests the connection between our system and the one we are communicating with.
Test Input data: Resume, Criteria, Login

.1 Create an account as an applicant and another as an employer.
.2 Ensure that the applicant and employer have matching criteria.
.3 Schedule an interview using the scheduler feature.
.4 Verify that both applicant and employer can connect at scheduled time.

Expected Results: The applicant and employer should be able to connect at the scheduled time.

Test Case 2.2.3:
Name: Filter out Unlikely Employer
Description: Tests the matching algorithm for filtering jobs to the applicant's criteria.
Test Input Data: Resume, Criteria.

.1 Create an account as an applicant with specific preferences.

.2 Create an account as an employer with contrasting preferences from the applicant.

.3 Use the filter feature to search for employers.

Expected Results: The employer should be filtered out and not displayed in the search results.

Test Case 1.2.1.4:
Name: Usability Testing
Description: Tests the ease of use of our program to the common user.
Test Input Data: Optional: resume, login, criteria.

.1 Ask a group of volunteers unfamiliar with the application to perform basic tasks such as creating an account, signing in and out, uploading a resume, testing the filter feature, etc.

.2 Observe and note any difficulties that occur during navigation.

Expected Result: User should be able to navigate the application with ease.

## Conclusion

Throughout this project we were able to "hands on" experience of what we were learning in class. As we learned about each of the chapters, we were able to apply what we learned to where we were in the project. This helped us to solidify what we have learned and to look at it from a new perspective. Here are some of the big take aways we had during the project:

- It is easy to get carried away with what requirements we wanted to add. Getting over excited with the thought of creating something on our own can make our eyes bigger than our stomachs. We can see what you, professor, were talking about when saying not to bring along the engineers as we would just promise more features that we can handle.
- Setting up the architecture is pretty hard. There are so many different ways to show the process and each one shows different information so choosing which is best can be hard.
- Planning out what test we want is hard. There is so much to test, and we want tests that looking into every aspect of a requirement but often that can't be done without taking way to long in the testing phase. Knowing the border values really helps on some.

All in all, this was an enjoyable project that helped us to see what we will most likely be doing in the field and has given us more confidence for when we go into interviews in the future.