Terms of Reference

HYPERSCREEN

University of British Columbia

CPSC 319 - Software Engineering Project

Date of Submission: January 13th, 2025

Instructor: Jerry Jim

By Group: Daine Yip, Tirth Choksey, James Reinhardt, Advika Kumari, Muneeba Ashiq, Steve Paul, Ki Bum Kim, Advay Rajguru

Table of Contents

Project Information

Background

Problem Statement: As AWS postings continue to garner a large number of applicants, the team at AWS Vancouver would like to speed up the process and allow recruiters and hiring managers to become more effective in filling positions in the organization.

- **What is it?**
  - HyperScreen is a web application allowing hiring managers to quickly and effectively identify top-talent.
- **Why build it?**
  - Recruiters and hiring managers lose valuable time searching through high volumes of applicants.
- **Why now?**
  - With the wide adoption of LLM tools and applications incentivizing mass application (Simplify/AIApply/ChatGPT/etc.) there is a growing need for optimal screening strategies.
- **Who is it for?**
  - AWS Hiring managers and recruiters receiving >750 applications per posting.

Goals and Success Metrics

Goal	Measure	Target
Accurate scoring system	% of high-scored candidates also green-lit through first-round manual screening	75% of high-scoring applicants (>80) also green-lit from manual processes
Hire high-value candidates	% of candidates hired through HyperScreen still employed after 1 year	50% of employees hired through HyperScreen maintain their employment after 1 year.
Reduce applicant screening time	Average time required to screen a candidate application	Reduce applicant screening time by 50% compared to manual process. Response time for all user actions < 2s

Scope and Release Planning

Approach: Implement waterfall methodology and work in pairs to ensure there is cover up for any unforeseen circumstances.

Module	Tasks	Pre-plan Done	In Design Date	BE Dev. Date	FE Dev. Date	Target Release Date	Cost
In-scope deliverables							
Applicant Module	Overarching Summary: - - Access to this module is open to any users with the URL and no authentication is needed. - Allow applicants to search job postings by reference number, title, or keywords. - Enable applicants to complete a basic form (e.g., name, address, contact info). - Upload and submit resumes/CVs in Word/text format.	Jan 15	Jan 15	Jan 15	Jan 22	Feb 17	$32K
Hiring Manager Module	- Scan current database of users for new posting - Set & edit scoring criteria with prioritization. - Create, delete, and update job postings. - Evaluate and rank applications based on criteria. - Search the applicant database for past applicants.	Jan 15	Jan 22	Jan 15	Feb 3	Feb 17	$24K
Recruiter/ Admin Module	- Hiring Manager onboarding - Hiring Manager login - Set up hiring manager accounts. - Configure score criteria dynamically (e.g., years of experience, education, skills). - Launch evaluation process for applications.	Jan 15	Jan 29	Jan22	Feb 17	Mar 14	72K
Reporting Module	- Show number of applicants to a posting - Show the ranking of all the applicants - Show a detailed report on specific applicants.	Jan 15	Feb 5	Jan29	Feb 28	Mar 14	60.8K
License	- All software created in this project will fall under MIT creative license.						
Out-of-scope deliverables							
ML Module	- Apply AI model beyond the keyword matching in the core requirements - generate candidate-specific questions for interview - capture interview notes from interviews into the database - Automation for notification of candidate - Ability to mask interview notes until all interviews						

Others	- Handling users beyond the specified 500 participants or 10 concurrent users is excluded. - End-to-end notifications and dynamic job descriptions. - Integration with Non-Google authentication - Mobile application development - Real-time collaboration features						

Constraints, Assumptions, and Risks

Constraints & Assumptions

Concern	Risk Level	Description
Constraints		
Response time	HIGH	- Response times for searches/filters less than 2 seconds.
Concurrent users	HIGH	- System can support up to 10 concurrent users
Assumptions		
Maximum users	HIGH	- Assume maximum user base of 500 participants (recruiter, HR and hiring manager, applicants, etc)

Risks and Mitigation Measure

Concern	Risk Level	Risk	Mitigation
Scoring accuracy with poorly formatted resumes	ME…	Some resumes may not be easily text parsable, potentially harming score	Guards to prevent image files submitted in the applicant portal
Unfamiliarity with AWS	HIGH	Project team's little experience in AWS may stall progress at times	Uncover early which functions are reliant on AWS features, so we can allocate additional time
Team member commitments	ME…	Bottlenecks and blockers may arise from team members adhering to other obligations	Two members assigned to each task to add redundancy in case one person fails to follow through.

Sponsors are not available	HIGH	Over-engineering or unnecessary resource spend	Creation of messaging channels with TA
Security	ME…	User data may be leaked	Security practices around cloud permissions, environment variables, and version control.
Difference of opinions and ideas between team members	LOW	Disagreements within project team may lead to slower progress	Regular meeting cadence and formal planning process to remove ambiguity

Project Governance

Role	Assignee	Responsibilities
Project Manager	Daine Yip	Deliverable scheduling
Site Reliability Engineer	James Reinhardt	Devops/Security/Testing - Assist with DB/backend
Front End	Advika Kumari	Frontend - Assist with backend API integration and ML
Reporting Operations and MLEngineer	Muneeba Ashiq	Stretch Goals - Assist with AI integration in to the project
Backend	Steve Paul	Design & develop API endpoints.
Frontend	Ki Bum Kim	Frontend- UI and UX
Database	Tirth Choksey	DB design, administration and SQL
Backend	Advay Rajguru	Assist with backend

Team Charter

- **Meeting Date and Time:** Mondays and Wednesday: In person during lecture, Saturday: 12pm - 1pm (Virtual)
- **Communication Medium and Turnaround:** Discord - 24 hours

Appendix

Open Questions and Discussions

1. Will we get an evaluating set (example resumes and their rankings)?
    a. Nope
2. Are there AWS-specific processes and documents you'd like completed this term?
    a. not needed
3. What cloud and Gen AI resources will be available to us?
    a. sign up for the AWS free tier account
4. How does the Admin scoring modifications affect the hiring manager module scoring criteria?
    a. Recruiter is like HR, can set default for HM, has access to all jobs
    b. HM responsible for subset of jobs only their team
    c. HM can add additional categories
    d. HM has say in criteria over recruiter
    e. If HM leaves the Recruiter should be able to assign postings to new HM
5. What platforms to include in a candidate's social media review.
    a. Not that important, probably the other sponsor added it
    b. Maybe look at more open-source profiles such as GitHub
6. What problems do you currently have with the recruitment process? What is slowing you down the most?
7. Who are the main applicants here? Students/new grads/full-time/seniors? Would you like separate scoring criteria for each?
    a. dependent on the job
8. Would you like separated scoring based on Job Description or would you like general scoring applied across all applicants?
    a. scoring based on the job posting and by demographic
9. What belongs in the hiring manager module, and what belongs in the admin recruiter module?
10. Should we be persistently storing scores of each applicant for every scoring criteria, or is a generated report enough
11. Should we store saved searches?
    a. valuable to store it
    b. if evaluation criteria is changed then the previous ones are not valid
12. What is meant by a system process to evaluate applications?
13. For security:
    RDS relation database service does support encryption which should be good enough
14. Recruiter should not override the hiring manager's criteria. Hiring manager has the final say
15.

Proposed Technology

1. Frontend and Backend: JavaScript/TypeScript (React) with Node.js, Express, Java, or Python.
2. SQL-based databases (MySQL, Postgres).
3. Selective use of AWS services (e.g., EC2, Lambda).
4. AI tooling, leveraging open-source models for cost-effectiveness.
5. Authentication via Google OAuth.
6. Version Control using Github
7. CI/CD using Github Actions
8. Project backlog maintained with Github Issues

Cost Framework

Module-wise cost rationale: We assume each team member will work 10 hours a work which means a total of 80 hours of team work. Based on our expected module development durations we multiplied the duration by the number of hours per week and the hourly rate of $100/hour.

Module 1 - Applicant Module = 20 days = 4 weeks (4 weeks * 80 hours/week * 100)

Module 2 - Hiring Manager Module = 15 days = 3 weeks (3 weeks * 80 hours/week * 100)

Module 3 - Recruiter/Admin Module = 45 days = 9 weeks (9 weeks * 80 hours/week * 100)

Module 4 - Reporting Module = 38 days = 7.6 weeks (7.6 weeks * 80 hours/week * 100)

Future Costs

We plan on using free services using Render or AWS to prototype our solution. Scaling to a larger user base, Amazon will likely have to migrate to paid accounts.