1.

Technical Interview Preparation Platform

Software Requirements Specification

CSCE 247-002 : Software Engineering

Dylan Cobb, Tyler Bouldin, Chris Feuchter, Blake Franks, Alex Marinov

January 24, 2026

Version 1.0

Table of Contents

1. Introduction

Purpose

The purpose of this project is to define the requirements for a software platform designed to help college students prepare for technical coding interviews through structured, hands-on practice. Technical interviews are a major barrier for students pursuing internships and full-time roles in software engineering and related technical fields. These interviews typically require strong problem-solving skills, familiarity with data structures and algorithms, and the ability to perform under time constraints.

Many students struggle with interview preparation due to a lack of guided practice, unclear performance feedback, and difficulty identifying which skills to focus on. While existing platforms provide large collections of problems, they often lack personalization, structured progression, and clear indicators of readiness. According to the National Association of Colleges and Employers (NACE), technical competency and problem-solving ability are consistently ranked among the most important skills employers seek in computing-related roles, yet many students feel underprepared for technical interviews despite completing relevant coursework.

This project aims to create a centralized practice platform that simulates technical interview conditions, provides targeted feedback, and helps students build confidence and competence over time. The system will allow students to practice coding problems, track progress, and identify strengths and weaknesses relevant to real-world interview scenarios.

Source:
National Association of Colleges and Employers (NACE) – Job Outlook & Career Readiness Competencies
https://www.naceweb.org/career-readiness/competencies/career-readiness-defined/

Scope

This document will define the functional and non-functional requirements of the Technical Interview Preparation Platform. Specifically, it will cover:

- The primary users and stakeholders involved in or affected by the system.
- The overall goals and intended use of the platform.
- Key system features and capabilities required to support technical interview preparation.

4.

- Constraints and assumptions that impact system design and usage.
- A competitive analysis to outline the purpose of the project

2. Stakeholders

Primary Stakeholders

- **College Students (Primary Users)**
  - Undergraduate students pursuing computer science or related technical degrees
  - Undergrad/Graduate students preparing for technical roles
  - Career-switchers enrolled in technical programs or bootcamps
    These users rely on the platform to improve technical interview performance and increase employability.
- **Hiring-Focused Students and Job Seekers**
  Students actively preparing for internships, co-ops, and full-time technical roles who require structured, interview-relevant practice.

Secondary Stakeholders

- **Universities and Academic Programs**
  Institutions that may recommend or integrate the platform as a supplemental learning or career preparation resource.
- **Career Services Departments**
  Offices that support students in job placement and interview readiness.
- **Employers and Recruiters**
  Organizations indirectly impacted by the platform through improved candidate preparedness and interview performance.
- **Development Team**
  The software engineering team is responsible for gathering requirements, designing, and implementing the system.

5.

MEREDITH MOORE

PROFILE

Gender	:	Female
Age	:	20
Education	:	UnderGraduate
Occupation	:	Student
	:	

Brands/Influences

- Stack overflow
- Youtube
- Reddit

BIOGRAPHY

As a computer science student that goes to the university of South Carolina she is wanting to start practicing for internships as she is in her sophomore year and is taking it more seriously. She does not know what to prioritize when going into the interview.

GOALS

- Be able to answer follow up questions quickly
- Be prepared with multiple skills
- Increase problem solving speed
- know how to tie concepts together

FRUSTRATIONS

- gets stuck on follow up questions
- has tried other websites but they are too hard
- Does not know what to spend their time on

PERSONALITY

Introvert ——— Extrovert

Thinking ——— Feeling

Judging ——— Perceiving

Sensing ——— Intuition

MOTIVATION

Job sercurity

Efficiency

Skill mastery

ALEX MARTINEZ

PROFILE

Gender	:	Male
Age	:	21
Education	:	Undergraduate Comp Sci
Occupation	:	Part time IT Assistant
Home Town	:	Raleigh NC

BIOGRAPHY

Alex is a junior computer science student actively applying for software engineering internships. He understands core concepts like arrays, hash maps, and recursion from his coursework, but struggles to apply them quickly during technical interviews. Alex wants consistent, realistic practice that mirrors real interview conditions so he can improve his confidence and performance.

GOALS

Improve problem-solving speed under time pressure

Gain confidence in live coding interviews

Identify weak areas in algorithms and data structures

FRUSTRATIONS

Freezes during timed coding problems

Doesn't know if he is "interview ready"

Practices randomly without clear progress indicators

PERSONALITY

Introvert ——— Extrovert

Thinking ——— Feeling

Judging ——— Perceiving

Sensing ——— Intuition

TECHNOLOGY

Software

Social Media

Mobile App

BRANDS

GitHub
Leetcode
Youtube

6.

JORDAN LEE

PROFILE

Gender   : Male
Age   : 23
Education   : Senior in college
Occupation   : None
Home Town   : San Jose CA

BIOGRAPHY

Jordan is a senior preparing for full-time software engineering roles. While academically strong, Jordan feels overwhelmed by the volume of interview topics and struggles to prioritize what matters most. Jordan wants a structured preparation path that helps identify weaknesses and maximize readiness in a limited time frame.

GOALS

Prioritize interview topics efficiently
Track readiness across multiple skill areas
Perform consistently in technical interviews

FRUSTRATIONS

Unsure which topics employers care about most
Wastes time on problems that don't improve performance
Lacks clear feedback beyond "right" or "wrong"

PERSONALITY

Introvert — Extrovert
Thinking — Feeling
Judging — Perceiving
Sensing — Intuition

TECHNOLOGY

Software
Social Media
Mobile App

BRANDS

HakersRank
Glassdoor
Blind

TAYLOR REYNOLDS

PROFILE

Gender   : Female
Age   : 38
Education   : Bachelors degree
Occupation   : Uni Career services coordinator
Home Town   : Austin TX

BIOGRAPHY

Taylor works in a university career services office and supports students preparing for technical interviews. Taylor is responsible for recommending resources that help improve student job placement outcomes. Scalability and effectiveness are critical, as one-on-one coaching is not always feasible.

GOALS

Improve student interview performance at scale
Recommend credible, structured prep tools
Support job placement metrics

FRUSTRATIONS

Limited scalable interview prep resources
Inconsistent student engagement
Difficulty measuring readiness outcomes

PERSONALITY

Introvert — Extrovert
Thinking — Feeling
Judging — Perceiving

TECHNOLOGY

Software
Social Media
Mobile App

BRANDS

LinkedIn
Coursera
University Career Platform

3. Constraints

Time Constraints

- The project must be completed within the timeline defined by the course schedule for CSCE 247.
- Development and documentation are limited to the duration of the academic semester.

Monetary Constraints

- No external funding is allocated for this project.

Technical Constraints

- The platform must be accessible through standard web browsers.
- The system must support use on common devices, including laptops and desktop computers.
- The code must be done in Java

Operational Constraints

- The system will be developed and evaluated in an academic environment.

4. Overall Description
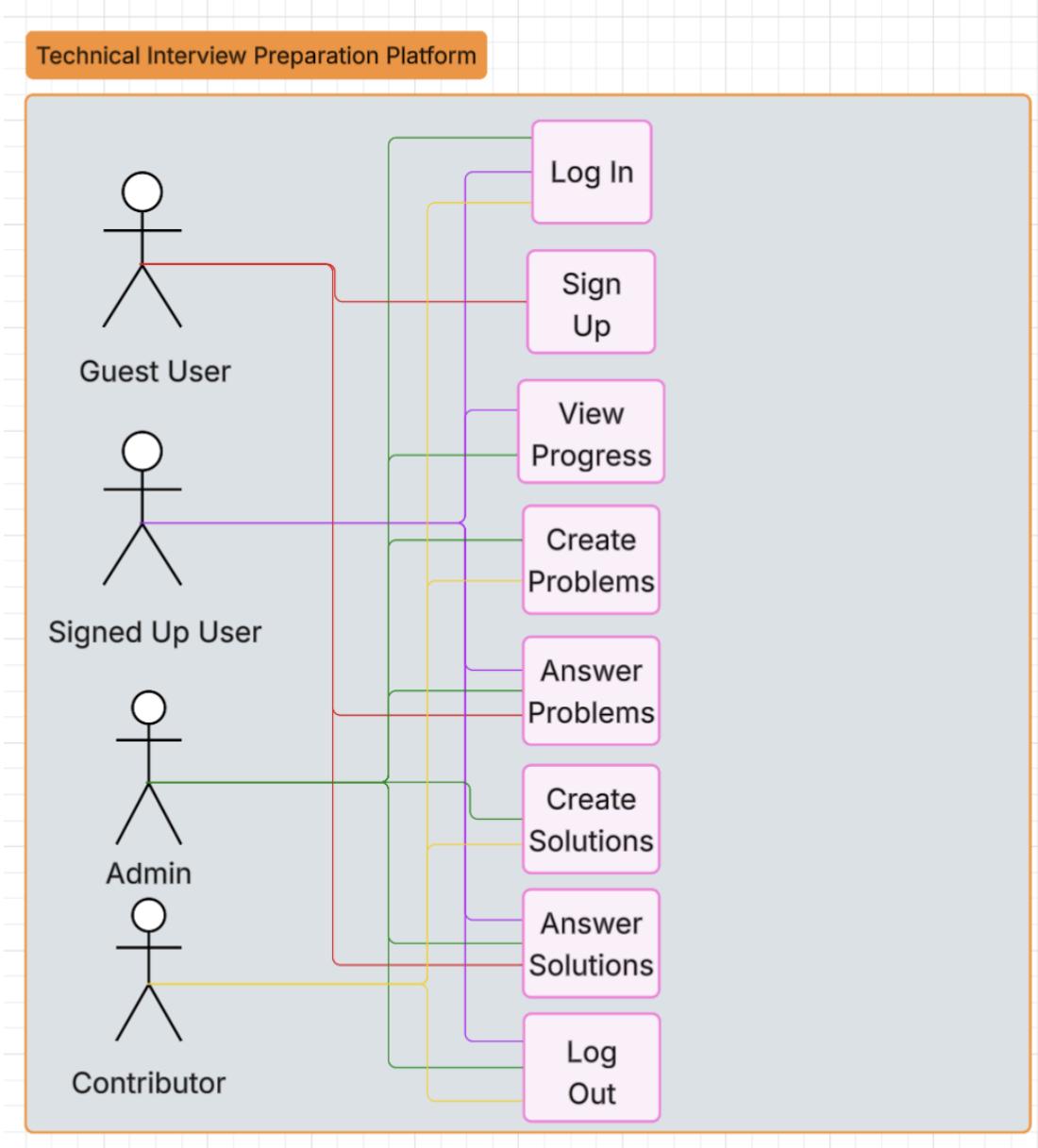
The Technical Interview Preparation Platform is intended to operate as a web-based system that provides users with tools to practice coding problems commonly encountered in technical interviews. Users will access the platform through a web browser and interact with a collection of interview-style programming problems organized by topic and difficulty.

The system will allow users to engage in practice sessions that simulate real interview conditions, such as timed problem-solving. It will also provide feedback on user performance and track progress over time to help users identify strengths and areas for improvement. The platform is designed to support independent use by students preparing for internships or full-time technical roles.

8.

The system will operate independently and does not require direct integration with external academic systems or employer platforms. Any data used by the system, such as practice problems or performance metrics, will be managed internally for the purpose of supporting interview preparation.

5. Business Use Cases

6. Functional Requirements

https://docs.google.com/spreadsheets/d/14ZgKG2nNnwpdBlPOBQXicxyN5a
QuH4y-vbdvqaTujWY/edit?gid=1972461568#gid=1972461568

7. Non-Functional Requirements

Look and Feel Requirements

● The user interface shall be simple and readable for beginner users.
● The system shall present problem statements and result in a consistent layout.

Usability Requirements

● The system shall allow a new user to start a practice session within 3 minutes of creating an account.
● The system shall clearly display why a submission failed (e.g., which test case category failed) without confusing jargon.

Performance Requirements

● The system shall load the problem list within 3 seconds on a standard broadband connection.
● The system shall return submission results within 5 seconds for typical problem sizes (prototype scope).

Maintainability and Support Requirements

● The system shall be modular enough that new problems can be added without changing the core practice workflow.
● The system shall provide basic error messages when services fail (e.g., submission evaluation unavailable).

Security Requirements

● The system shall store user credentials securely (passwords shall not be stored in plain text).

- The system shall restrict access to a user's progress data to that user only.

Cultural Requirements

- The system shall avoid offensive or discriminatory language in problem statements and examples.
- The system shall use inclusive, neutral wording throughout the UI.

Legal Requirements

- The system shall not collect unnecessary personal data beyond what is required for basic account creation.
- The system shall provide a basic privacy notice describing what user data is stored (account + progress).

9. Competitive Analysis

	LEETCODE
Strengths	Leetcode has a massive problem library covering many topics for an array of learning, and has many playlists for interview topics (array, graphs, design, data structures). Common questions that match directly for big tech interviews (Amazon, Google, Meta), which allows users to be extremely prepared. Leetcode has strong brand recognition for many companies, which is why many users tend to use it.
Weaknesses	There is a paywall for premium services, which allows for company lists and better solutions. There is a learning curve for beginners, which may be overwhelming for users who have never learned some of the basics or tougher questions. Promotes "grind culture," meaning that some people may just learn memorization rather than truly learning what the

	problem entails.
Audience/ Focus	The primary audience and focus of LeetCode is Software engineers and students who are prepping for interviews. Leetcode focuses on data structures, algorithms, and patterns for coding interviews.

	HackerRank
Strengths	HackerRank offers a range of challenges for coding interview/ prep for similar topics like LeetCode. HackerRank is commonly used for early-stage interview prep questions rather than deep algorithm optimization. HackerRank is also well recognized by employers as a hiring assessment, which makes it very popular.
Weaknesses	HackerRank lacks many advanced, interview-specific problems, which makes it less effective for higher-level big tech interviews. Many of the questions/ problems on the site are more academic or assessment-based rather than interview-based. The platform's strongest features are geared towards companies and recruiters, rather than users,s so you don't really get your money's worth.
Audience / Focus	It's used by students and early-career developers who want to practice the fundamentals for coding and prepare for technical assessments rather than advanced interview rounds. HackerRank is focused on companies and recruiters who need to see the user's skills during the hiring process, typically for early-stage recruiting.

	CodeSignal
Strength	Focuses on realistic coding assessments that simulate real job

	tasks, including full-stack and role-specific problems, helping users practice skills for real-world scenarios. Many of the assessments are used by companies for hiring, allowing users to prepare for screening formats rather than algorithm- based questions. Known for its clean interface and structured assessments, making the testing feel professional and similar to real interviews.
Weaknesses	CodeSignal has a much smaller problem library compared to LeetCode, which limits the amount of practice users can do for data structures and algorithms. The platform is less focused on traditional algorithms and data structures, which could disadvantage users preparing for big-tech interviews. Many of the advanced features are designed for companies so that individual users may find fewer low-cost options for practice.
Audience / Focus	CodeSignal is focused on companies that want standardized assessments for hiring. Serves job seekers and developers who want to practice real-world practice rather than deep thinking questions. The platform emphasises skill validation and job-relevant questions over problem grinding.

Summary

	Strengths	Weaknesses	Focus
LeetCode	+Company Leader + Vast Question Library	Interview-based	A strategic approach to interviews
HackerRank	+widely used	- Fewer advanced interview prep	Skill assessment and early-stage

13.

	+Good for fundamentals	-Less focused on big-tech	interviews
CodeSignal	+realistic, real-world coding assessments +Good UI	-Small Question Library -Less algorithm-based	Real-world coding tasks

Based on the completed competitive analysis above, we have gained a lot of insight about similar interview-based sites. In order for our product to succeed, it must have a large problem bank with a clean and useful interface that doesn't have unnecessary problems. LeetCode is a strong contender, because their question bank is quite large and has high industry level recognition, so it can be really daunting for beginners. HackerRank is a user-friendly introduction to competitive programming. It's not as extensive, but a more user-friendly website compared to LeetCode. CodeSignal was good for real questions, but had a small question bank. Users preparing for interviews want a site that's organized and efficient, with a strong question bank, but they also don't want unnecessary problems that can feel frustrating and overwhelming at times. In order for us to have the strongest product on the market, we must blend the depth of LeetCode, HackerRank's approach, and CodeSignal's real-life approach.