

A Mobile Application for Technical Interview Success

Brooke Kelsey and Rachel Stein
Swarthmore College
{bkelsey1, rstein2}@cs.swarthmore.edu

1. INTRODUCTION

The technical interview process is daunting for even the most experienced computer science students. The breadth of information that may be required at anytime, from the first phone interview to that last on-site interview with the company's founder, forces aspiring software engineers to sift through piles of notes from their previous courses. Bulky textbooks such as [4] and [2] are a significant help in preparing for technical interviews, but require setting aside blocks of time that many college students rarely have. These books state in their introductions that the readers should review the technical 'trivia' that underlie many of their questions elsewhere. We propose moving the fact-based part of the review process onto a mobile platform (see code at [1]) that can be accessed at any time. While 'whiteboard interviews' are often cited as a rite of passage in the tech industry, many of questions posed are short-answer or conceptual questions, which can be practiced without writing code. By providing a platform that gives users easy access to practice questions that can be answered in 5 minutes or less, we intend to make use of the pockets of time a college student has in-between classes or waiting for a train. Initially, we will implement this as a mobile website, but we hope to move to an iOS application. We believe this platform will fill a void within the interview practice process and make practicing questions more amenable to students.

At a minimum, we hope to provide a basic question-and-answer platform, in which a single user attempts to answer questions in a manner similar to [3], or other flashcard applications. Our end goal is to support two modes: a 'game mode' in which questions are formatted as multiple choice, and a 'flashcard mode' in which both the questions and answers are provided for the user. This means that our game question pool is limited to those that have one-word or phrase answers, such as Big-O analysis or 'name-the-term' questions. Questions will be stored on Amazon Web Services and tagged with their relevant topics. Users can sort questions selecting categories of questions that they would like to see (e.g. trees or parallel threading). After searching

the database for relevant questions, questions will be posed to the user one at a time. Within game mode, users will be able to answer multiple choice questions. While it may not be possible within the short project timeline, we would eventually like users to be able to connect and compete with each other within this mode - similar to [5]. We hope that this feature would motivate users to play more frequently. Within flashcard mode, we intend for users to be able to read brief answers to conceptual questions. While we will be creating the initial database ourselves, we intend for users to be able to submit sets of their own questions. This will increase the depth of material covered, and allow users to share knowledge about areas that the authors do not have significant exposure to. For example, if the reach of this application were to expand to upper-level course material, students from the various electives could contribute questions from courses they have taken, and that other students - us included - may not have.

We plan to evaluate a simplified version of our application by distributing it as a study tool for students in the intermediate classes of Swarthmore College's computer science department, "CS35: Data Structures and Algorithms" and "CS31: Introduction to Computer Systems". Our rationale for this testing environment is three-fold. First, the scope of these classes is comparatively narrow - we could quickly create a database of questions from preexisting study material. Second, the classes' topics are common interview questions, so the application could relatively easily scale up to include a broader range of questions. Finally, because we are current Swarthmore students, we expect to be able to effectively encourage students to use the application and collect their feedback. Since the goal is to provide an easily-accessible end-user product, having as many unbiased users as possible using it and giving feedback is going to be our most valuable test mechanism.

2. RELATED WORK

The clearest motivation for our platform derives from the current resources available for job seekers in the tech industry. Books such as [4] and [2] are seen as the canonical guides to performing well in any interview involving a knowledge of computer science, whether it be a coding interview, brainteasers, or a broader technical discussion with the interviewer. These books are designed as all-in-one guides to scoring a dream job, and so contain an overwhelming amount of material. It can be difficult to pick a starting point, or to identify precisely what topics you should review further. Further, because they are published books, these

materials are not as dynamic as an online platform could be (e.g. the question bank is not updated until the next edition). In contrast, this platform can be updated at any point with corrections and new material. Additionally, it will be portable - rather than needing a book, laptop, or tablet physically with you that has the book on it, all that is needed is access to the application. By only seeing one question at a time, the amount of material the user sees at any given time is not overwhelming.

Multiple web and mobile based applications have been developed with the goal of improving the study experience. Quizlet [3] is a browser-based and mobile application flashcard tool allowing users to create their own sets of flashcards or search for sets created by other users. Users can have the website generate quiz games with the flashcard set (such as fill-in-the-blank or matching) to make the experience more interactive. During these games, Quizlet keeps track of cards the user gets wrong, building the next round of the game entirely from the cards they answered incorrectly during the current round. When a user is flipping through the set as they would with physical cards, they have the option to flag cards they are having trouble with. Ideally, our platform to have a similar function in which users can flag questions they would like to come back to. This saved application state

would require a database look up for the flagged questions every time the user returns.

The potential player-vs.-player feature that we hope to implement is inspired by QuizUp [5]. Quizup a popular mobile trivia application that allows users to challenge each other, earn badges, and choose from over 400 categories. We would like to draw from the competitive nature of this app for our own application, as we believe that adding a social aspect will increase collaboration and motivation when studying.

3. REFERENCES

- [1] "<https://github.com/rstein66/CS91-Final-Project>".
- [2] A. P. Adnan Aziz, Tsung-Hsien Lee. *Elements of Programming Interviews: The Insider's Guide*. CreateSpace Independent Publishing Platform, 2012.
- [3] Q. LLC. Quizlet - simple free learning tools for students and teachers, 2014. <http://quizlet.com/>.
- [4] G. L. McDowell. *Cracking the Coding Interview: 150 Programming Questions and Solutions*. Career Cup, CA, USA, 2013.
- [5] P. Vanilla. Quizup: The biggest trivia game in the world, 2013. <http://quizup.com>.