SJSU GO

Rod Fatoohi, Sheryl Ehrman, Lisa Francesca, Chanip Chong, Joseph Khoury, Anthony Minaise, Dan Pham, Manas Abhyankar, Jia Chen, Raymond Chen, Antonio Corona-Sanchez

Charles W. Davidson College of Engineering, San Jose State University, San Jose, California 95192, USA

Abstract—The Davidson College of Engineering at San Jose State University (SJSU), San Jose, California, offers a rewards program for students to get points for attending professional and academic events called GO. Currently, students submit events for the program through a Google form with many limitations. In this project, we have developed mobile applications (for both iOS and Android systems) and a website to submit events, to view points and prizes, and to navigate the system. The goal is to increase student participation in these events and ease of use of the system.

Keywords—mobile applications, software development, professional events.

I. INTRODUCTION

The Davidson College of Engineering (CoE) at San Jose State University (SJSU) recently initiated a program called GO [1]. It promotes networking, career-building, building skills through industry contacts, and hosting events at the school. Through these events, students learn important skills for their future career and meet with industry leaders. Students are encouraged to attend these events and can submit proof of their attendance to earn points, which can be converted to prizes awarded by the college. However, the program seems to lack participation partly due to the current submission process, which is based on a Google form, and sometimes students are unaware of these events. Also, verifying and managing a large body of students’ submissions in such a system is quite a challenge. Currently, there are over 200 students using the GO program in a college with over 6000 students.

Just as technology is becoming more integrated into our daily lives, researchers are finding ways to use technology to help promote student involvement and learning [2]. Byron and Khazanchi [3] show that rewards help stimulate student creativity and motivation in school. They analyzed whether and how rewards affect students to be more creative. Their design was to test the extent to which rewards are given to see students’ creative outcomes. When students are satisfied with their work, they are more likely to be motivated by rewards [4] and more likely to stay active in the reward program [5]. There is also a link between rewards in the workplace and performance management. Lawler [6] shows that employees who were given rewards after a positive review report tended to perform better in the workplace.

The development of the GO Program mobile app in iOS and Android as well as the overhaul of the desktop website is primarily focused on increasing the participation of the college students in campus events to better prepare them for their future careers. Based on the work by Wong [7], a desktop application will be necessary as not all users will be inclined to use the mobile app. For mobile applications, both Android and iOS implementations might be needed since it is hard to determine which mobile OS the user will use. As found by Götz [8], there are negligible personality trait differences between iOS and Android users, making it hard to draw any conclusions about which OS the user’s phone will operate on.

II. SYSTEM DESIGN

Fig. 1 shows a top-level view of the design architecture. A similar architecture is used for all three applications: Android, iOS, and the web. For the current implementation, we used Firebase, a Google developed backend database solution that provides cross-platform support for the three applications. Users, as students, and admins, as program administrators, interact with the frontend to request and submit data. The backend serves data to the frontend based on client requests and stores client data. Fig. 2 shows the database entry diagram of the application.

Fig. 3 shows a flow diagram of the overall application. A client accesses the GO program by navigating to the main page and either logging in or signing up if the client does not already have an account. After successful client login, the client is directed to their dashboard where aggregate information such as total points accumulated, recent event activity, and available prizes are shown.

Fig. 4 shows the system functional requirements. A user, a qualified student, should be able to login or sign up for an account. Also, a user should be to choose an event type, from a drop-down menu, or insert an event being attended if it is not listed in the menu. In addition, a user should be able to upload a document of proof or a photo of an attended event. Moreover, a user should be able to view their accumulated points and claim a prize. On the other hand, an administrator should be able to add and revise possible events in the list, review the proof that a user has uploaded and assign corresponding points, and to pull statistical data in different formats (such as percentage of departments that actively engaged, highest scores of individuals, number of participated students, eligibility, and student feedback).

III. IMPLEMENTATION

For the iOS implementation, we designed and created the Minimum Viable Product (MVP) as an iOS app. We also used the programming language Swift and the Integrated Development Environment (IDE) Xcode as well as Firebase (as it was used in all our implementations). We used both
focused on specific unit tests, regression testing, and integration testing in order to validate the entire application. For example, in regression testing, we tested many functionalities including sign up, login, event submission, claim prize, and log out.

V. CONCLUSION & FUTURE WORK

In this project, we overhauled the GO program website and submission form by creating two mobile applications (for iOS and Android) and a new website that would enable more students to benefit from the program. This is only the first phase of the project where we were able to develop, implement and test prototypes of the three applications. More debugging and testing are needed as well as adding more features.

Next steps for this project are to properly deploy these applications so students can use them. Currently, we are working to deploy the iOS app in Apple App store and use the Apple Developer TestFlight for beta testing. Additionally, we plan to integrate these applications with the San Jose State University database to properly assess student emails for verification.

REFERENCES

Figure 1. Design architecture

Figure 2. Database entry diagram of the application
Figure 3. Flow diagram of the application

Figure 4. User case diagram of requirements.
Figures 5 and 6. UI mockup of iOS app login and dashboard screens