SCAV: Scav Collaboration Application for Victory
Ekaterina Shkrabova and Anton Melnikov

INTRODUCTION

SCAV set out to address a need presented by University of Chicago students, in the form of the annual Scavenger Hunt. The event is rather unique to our campus and while there have been mobile apps made especially for UChicago students, Scav has not been included in such specializations. We see an opportunity to provide UChicago students with an application that aids them in participating in Scav. A mobile app can significantly ease communication and help organisational efforts for individual Scav teams and aid individual participants in having a greater control over their involvement and efforts. Currently, there is no single infrastructure provided for the teams to take advantage of for organising and communicating amongst each other, resulting in a medley of communication methods of various technological complexity – websites, Google spreadsheets, email lists, Facebook groups. We imagine that it would be beneficial to provide a unified tool that incorporates some of these approaches and accounts for Scav’s specific needs. An app could also take full advantage of the particulars of the devices on which it would run – mobility, that is ease of transportation and pervasiveness of such devices allows convenient access for the users in the chaotic and fast-paced environment of Scav. Additionally, incorporating capabilities, cameras, and alarm and calendar systems that most smartphones are equipped with would be more useful for Scav participants than if one simply used a website, even one formatted for mobile devices.

OBJECTIVES

We wanted to make an app that is as flexible as possible, to fit Scav’s creative format and not to restrict teams in the way they could use this tool. To do this we’ve had to primarily adopt the structure that Scav organises themselves – there are always items in a List, Teams and at least one Captain per Team.

With this loose format in mind, we implemented a structure centered around Items, Teams, and Members. We also needed to use some sort of a registration and validation system even though one is not inherent in the Scav structure. Lacking resources for a more sophisticated system, we decided to validate based on Cnnet ID email access, even though this might exclude some participants who are not actually UChicago students who come in to help regardless.

Upon registration, a user picks his team and after that he can see other members of his team and browse the item list, sorted by original order or by completion status (available, in progress, or done), and, of course, view and edit his own information. Available items can be claimed by the user as ones he is working on.

We ran into some time-consuming trouble when focusing on implementing server-client communications so vital to the multi-device interaction that the app is based on. As the result, we did not have much time left to implement all the features we are planning on, and will have to continue working on the app to reach full functionality. We still have to implement map, calendar, messaging and a couple of other APIs for the app to be most useful to be run on a mobile device. For this year’s Scav, in two-months time, we will have a testable version that can perhaps be used by a particular team to give us proper feedback.

CONCLUSIONS

While we were not as successful as we hoped to be in completing the app in the allotted timeframe, the experience was nonetheless important and quite educational. Naturally, we gained knowledge about how servers, databases, and client apps can work together for small-scale projects that do not require many resources. We understand that some of the architecture we built will have to be re-worked and altered for the actual demands of short-term high activity that happens during Scav. This was a unique opportunity to work on such a project with constraints that one would have to deal with when working on a large-scale project while still having control over all aspects of the project.

The most important lesson we gained from this experience is that sometimes a lot of work and time goes into processes that might not yield tangible or useful results. It is quite possible that most of the work for a project might have to be discarded quite close to the deadline in favor of quickly adopting other tools and technologies that perhaps allow more progress. It is an important lesson in being flexible enough to know when to discard progress in favor of other approaches, even though it is important to study new technologies and approaches that might provide valuable usage.

Despite this knowledge, it seems that an approach of working with most customisable and editable tools available is best in the case of problems and troubleshooting. While some tools might ease a lot of work and seem to be rather universal, it can be hard to work with these tools without great knowledge of how exactly these tools work, something that might not be available.

REFERENCES

We extensively used Google’s documentation for Android developers and App Engine, Cloud Messaging, and Endpoints users. Additionally, we used some Flask framework documentation, JSON documentation, and various troubleshooting sites.

There was not any other academic work available that we could use for the development and building of this app.

Acknowledgements

This project was completed as part of CS 234 Mobile Computing (Winter 2013), taught by Professor Chien, with TA support by Tanakorn L. and Falcon Dai. Additionally, it was completed as part of CS 162, taught by Professor Kurtz. We would also like to thank Anil Vaitla for pointing us in the right direction at a desperate time.