CoCoA Admin Progress Report

Introduction

CoCoA Admin is a mobile web application that help school teachers to manage their CCA events. This report will present the work that is done, the progress compared to the schedule, the difficulties we've encountered and the changes made since the initial proposal, some feedbacks we gather from our potential customers, as well as our updated timeline.

Application prototype

A minimal version has been deployed to http://54.169.115.238/www/index.html and the features that are currently available are:

- create a new CCA (only with a name, you can't import members yet)
- create a new event in a CCA (only with a name, you can't add in details yet)
- select participants (from the currently hardcoded members) for the event in the event detail view; you can search for the members by name
- add new task (e.g. submit form A) for the event
- enter a task and
    - mark whether a participant has complete the task
    - view participant detail (currently it's only a blank view)

Progress compared to schedule

Generally speaking, our project is on schedule. We are supposed to finish UI prototyping and Google api exploration by Oct 19th and we've managed to do that. However, at this point the app is only a prototype and not really functioning since it's not connected to the backend (and Google drive), so we are going to put more efforts to make sure that we finish the first round of integration to make the app functioning by Oct 26th.

Difficulties encountered

1. Unfamiliarity of the user's actual situation

    This is caused by the nature of our project, one that is proposed actually by the one of the targeted users, Yanjie, and it has been a serious issue since the beginning of our project.

Our targeted users are the pre-college school teachers who are also CCA managers at the same time, and we aim to facilitate their work in CCA administration. However, none of us has been through the JC or Secondary life in Singapore, and thus not familiar with how these whole procedure works. This requires us to have massive communication with Yanjie and some other teachers from the same school in order to thoroughly understand the problems. We actually designed surveys to further broaden the user base to other schools such as HCI. Currently, we are trying to consolidate a few interested teachers to be our consultants, from whom we can get feedbacks consistently. This is still in progress.

2. Misunderstanding the Google Drive and Google Spreadsheet API

   This has been considered as one of the biggest challenges of our project. None of us has any prior experience with these two APIs. However, as we tried dividing ourselves into two groups in the early phase - one working with front-ends without Google API and the other working focusedly on Google API - with the assumption that Google API works directly on the front end, serious problems occurred when we actually found out that the assumption was not so correct. The situation is, although it is possible to put everything in the front-end, it is something that would significantly affect the performance and responsiveness of the product. Fortunately, we discovered this not too long after the project started, and hence still had enough time to fix it. We currently already started building a server that works with all the Google API related jobs, and revising the front-end architecture so that it fits better to the new communication mechanism.

3. Inconsistency in official Google Drive and Google Spreadsheet API

   This has been a notorious “feature” of Google API. The documentations for Google’s APIs are usually outdated. What happens actually is that even though Google has updated its API for long, no one would bother to change the documentation on developers’ site. We have been following a few examples on Google Developer’s page, and none of them works. In the end, we have to rely on external libraries that abstracts away some of the troublesome procedures. Currently we have found this library called edit-google-spreadsheet on npm and are still testing its APIs.

Changes

We do have a significant change in the design philosophy of our product. Previously our product was designed to be something similar to an interface of the Google Spreadsheets that only performs partial functionalities of the Google Spreadsheet. However, as we go on with the design, we gradually realised that this is too restrictive and would seriously affect the usability of our product. Therefore, we decided to change the positioning of our product. Rather than being a

simple user interface of Google spreadsheets, we redesign our app in such a way that the users can do on our app whatever they need to do with Google Spreadsheet for CCA administration. More specifically, we allow users to take full control of the Google Spreadsheet via our app by allowing them to add CCA, to add events inside CCA and even edit participants to some certain events, all of which are loading from Spreadsheet files in the previous design without the ability to CRUD.

Feedbacks from potential customers

We've asked our potential customers to try out our prototype, and we've received some feedbacks from them such as

- allow users to create their own tasks rather than having a set of predefined tasks
- have a default task named "attendance" (haven't implemented)

However, they cannot give us more suggestions since the app has not integrated with Google drive and the workflow is not complete yet. We decided to finish the integration first by this weekend and ask them to try out the whole workflow so that we can gather more useful feedbacks.

updated timeline

Duration	Tasks
20 Oct - 26 Oct	Implement project skeleton & basic connection with Google Drive
27 Oct - 2 Nov	Account Management in Google and Google Drive Syncing
3 Nov - 9 Nov	Account Management in Google and Google Drive Syncing
10 Nov - 16 Nov	Testing and Refinement
17 Nov - 21 Nov	Project Wrap-up and Report preparation