Team Cronus - Meeting Reports

* Karthik Palaniappan(karth295)
* Ki Ming (Keith) Yeung(kyeungk)
* Linsen Wu(linsenwu)
* Tiffany Lin(lintiff)
* Micaela Tolliver(mictol42)
* Panji Wisesa(panjiw)
* Tsz Fung Cheung(tfcheung)

**Meeting with our customers**

Panji and I (Karthik) met with OneDrinkAway (our customers) on Friday April, 11th at 1:30 to discuss their requirements/expectations for our product. We first explained the concept of our project—we are creating an app for individuals and groups to organize both tasks and bills. We plan to mainly represent these with a calendar view with both tasks and bills, and a separate detailed view with all bills. This is a combination of the proposals “AgreeMates” and “Community Costs”.

They first asked several clarifying questions on our summary. How would you resolve events between multiple groups? Your own personal events? We discussed allowing users to “filter” what they see on the calendar by group. What’s unique about your product? There are clearly multiple systems out there for basic task management. Google calendar is a popular one, and it also includes having group-based calendars. However, no popular system that we could find integrated tasks (events) and bills (managing costs) for groups in a calendar view.

Then, we discussed extra requirements that they found important. Here are the main ones: Not all tasks have a “due-date”. How would we create an task such as “take out the trash”, or “do the dishes”, which just need to be done when they come up? We discussed creating a separate list of “TODOs”. If it were my turn to “take out the trash”, it would be on my list until I indicate that I completed it. Then, the system would assign that “TODO” to the next person in the rotation. How are we going to know if a new task is added? We discussed creating a notification system that would indicate new tools were available (think Catalyst). We also talked about an option to receive email notifications.

Other features such as debt reduction or paying people back online came up during the meeting, but they are now on our “stretch” feature list.

**Meeting with TA**

During our meeting with our TA, Isaac, we discussed issues with him as a customer and a TA. First, we discussed our product purpose: an application to split tasks and bills, helping with communication to specific duties. We clarified that our project will differentiate itself from available methods like post-it note bulletin boards by allowing users to access the calendar from their own computers instead of a central location. As a customer, our TA said that rescoping this would include a wider set of customers but could have competition with other group management applications and needed some differentiation. We suggested a couple of features for our application. One was the ability to join multiple groups and display tasks on the same calendar, then filter these tasks to our user needs. Another feature was the ability to have rotating tasks, which would differentiate our product from avaliable calendar applications. As a customer, our TA suggested that we need one key killer feature that would differentiate this product from other available products. Additionally, we asked the TA as a TA what languages would be best for backend. We gave the suggestion of PHP and Javascript as backend, and the TA brought up that the most common backend development framework used at the moment is Ruby on Rails, and he would not suggest using PHP for this project. We decided based off of this judgement to work on Ruby on Rails during this week.

**Meeting as a customer with GeoPost**

Keith and Micky were the customers for GeoPost at 1:30 on Friday, April 11th to discuss their product. Their project sought to allow users to unlock location-based messages on a map. In the meeting, we suggested a few extended features that would improve the user experience and usability of their product. We suggested that they could add timestamps to the messages for organization. Additionally, we raised issues with how a user would unlock a message. It would be required that a user be at location to unlock a message, so they do not collect messages to unlock at their leisure. This better fits their app’s purpose of having location-based messages by forcing the user to be on location. We also raised some issues such as when a lot of locations are pinned, it can be clustered and hard to select pins individually, some user might spam pins all over the place, so we suggested some ranking system that allows the user to “unlock” amount of pins they can put down. We have also suggested filterings to allow users to filter between public posts and friend’s posts or types of posts, and we suggested using google map features like signifying if the user could find the message from biking, driving, or walking.