

# Team Cronus - Product Description

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

## **What is your product?:**

“TeamPlayer” is a group based task manager. Deployed as a web app, users can create and join teams where they can schedule task and manage bills between everyone on their teams online. TeamPlayer’s task management system is focused on a monthly main calendar where all team tasks can be created, viewed, and modified with a filtering system to help the users view and manage their individual tasks. Another main feature of TeamPlayer is the flexible bill management system, where groups can split and manage their bills between the users in the teams with integration to the main calendar through due dates and notification. Finally, teams can create tasks that automatically rotate between members after they are completed.

## **Target audience?**

We have rescoped our target audience past roommates to teams who need to organize tasks. Our target audience are groups who need an application that can help them organize and manage tasks and bills between their members. These functions are important to target groups such as roommates, who need to share household tasks and shared bills. Our target audience expands past roommates as a user base, as our product can be used in project groups who need to organize individual tasks, in planning groups that need to plan events and tasks prepared at specific times, and groups of friends who need to organize who owes what.

## **What problem does it solve?**

This application will help solve communication problems between groups of people. For the most part, it will help stop communication problems created by poor scheduling, poor reminders on due dates, or miscommunications over finances. It will help groups reduce the time needed to communicate with multiple people, reduce misunderstandings on who is assigned what task, and reduce to remember how much each person has spent and owes. Overall, it will help cleanly facilitate group bills and tasks by providing a clean interface to view and assign tasks.

## **What alternatives exist, and what are their strengths and weaknesses?**

Alternatives to these problems already exist. A representative application, HomeSlice, manages a social network between roommates to help with communication. HomeSlice seeks to provide incentive for good behavior and transparency in group dynamics. They provide a whiteboard to help with communication between roommates, a supplies manager to help with who needs to buy what and how much is left, and a chores and bills manager to help facilitate tasks and finances. Their app provides a clean interface for a small community, being roommates, to manage basic tasks. Its advantages are that it provides push notifications to notify community members of tasks they need to do. However, users have complained that the application does not provide the ability to submit one bill for multiple members. Additionally, there’s no way to snooze events.

Another cliché alternative is shared calendars, like Google Calendar. It provides an event based approach in a calendar format. Its advantages include already being widely used, and an integration with other Google software like Gmail. You can create a shared calendar, as well as subscribe to multiple calendars (e.g. a personal one). However, it is weak for groups because it is only task based--there is no concept of “bill management”. It also does not allow for tasks that do not have a specific due date.

## **How will your system be different, from the user's point of view? Be specific.**

Our application will differentiate itself by providing a calendar view of events to facilitate time-dependant tasks. While HomeSlice provides a lot of services for roommates to organize their group, it does not provide a calendar view that could make the application difficult for paying bills on time or completing tasks on time. Furthermore, HomeSlice specifically targets roommates as a user base and not just groups who need to organize events. Seeing as our application works with general tasks and bills and not specifics like supplies, we can reach a wider audience.

Our application will differentiate itself from Google Calendar by being geared toward groups rather than events. We will differentiate between items such as tasks, bills, events, and todo's (tasks without a due date). We will also provide a richer set of features such as calculating who owes whom, cycling tasks between a group, and allowing repeat events.

## **What are its major features?**

### Main Features:

- view all tasks and bills on the calendar
  - users can filter these tasks and bills and view only certain items, whether it be from only specific groups, only your bills or only your tasks
- task creation for individual, group, subgroup
  - tasks contain a name, description, and members assigned to it.
  - able to assign the task a day of completion the date that will appear on the calendar
- bill input
  - asks like a task, with specific users, name, description, and due date
  - will have a total cost and ability to split bill evenly
  - members can be assigned specific amount they owe on the bill
- view running list of user's bills and payments
  - can view bills for selected groups users are a part of
  - can see how much they owe another member
- ability for users to cycle through tasks and todo's
  - users will have a specific task that will be passed to other users in the group after they have completed the task.

### Stretch Features:

- (email) notifications
  - notifications could be sent on the day of an event to remind the user what events they have due that day
- group management
  - Currently, groups can be created and new members can be added by anyone in the group, but can be extended to management permissions
- Snooze/Move events
  - Currently, the creator of an event must remove, and create a new event. However, we discussed the ability for anyone whom the task applies to "snooze" the event. This may require additional thought, as the task often dictates when it can be snoozed. For example, a user probably does not want to snooze a rent payment.

## **What are its non-functional requirements?**

Our application will need to require that the user display is up to date when other users have entered new information. This means that our application will need to require consistency and accuracy among databases and updates from other users. Additionally, since our product will require each user to have their own username and password to access their calendar, our product needs to require security. Another non-functional requirement would be performance with a short response time, especially in the calendar and bill

viewing interfaces. Selecting and deselecting to filter tasks and bills, as well as calculating payment totals may need to be efficient.

**What external documentation will you provide that will enable users to understand and use your product?**

Our product will provide new users an optional step-by-step tutorial on the features of our product when they log in for the first time. From then on, the user will be able to provide integrated text on each feature if the user needs more help. Our main targets for understanding our product will be on the different functions, so when a user first uses one of the functions they will be informed how it works.