Team Cronus - Use Cases

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

**Use Cases Considered**

The following use cases are the most important ones and they cover the important scenarios.

* Login/Create Account

Every user must go through the Login/Create Account process to use this application.

* Home Page View

The home page is the first view after the user logs in. It contains summaries of user’s information and it will be displayed every time a user is logged in.

* Group Management

Since our app is a group based task and bill management application, joining and creating groups is an important use case as it is how our task and bills are organized.

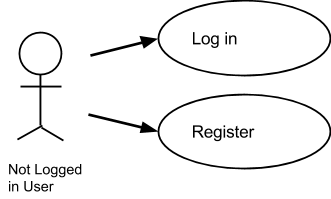
* Task Management and Bill Management

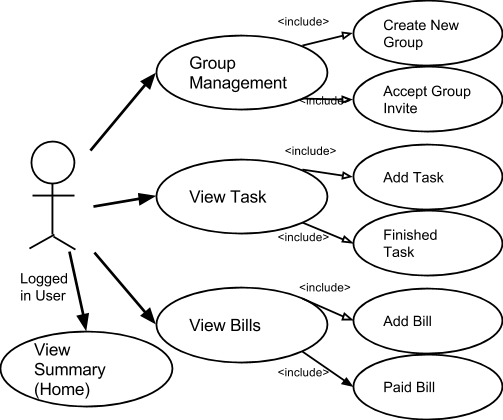
Task and bill management are the two core functionality of the application and will be frequently used.

**Failure handling**

The most frequent failure that would occur would be cause by concurrency issue: when two users of a group modified the same information. The general idea of solving this problem is to accept the first submitted change, and notify the other users that the information has been modified. If an unexpected failure happens, we would ask the user to provide feedback on what happened. We will then try to reproduce the failure, and start to troubleshoot to fix the problem.

**Use Cases Diagrams**





**Detailed Use Cases**

**Use Case: Login**

Actor: Not logged in User

Precondition: User is not logged in

Triggers: User, who have not logged in, come to the webpage

Success Condition: The user logged in and get redirected to home page

Failure End Condition: The user failed to login, back to login/register page

Use Case Basic Flow:

1) The user comes to the product website

2) The system displays the login input box and the create account input box

3) The user input his/her username and password in the login input box then click login

4) Login successful: the user becomes a “logged in user” and the system displays the application home page

Alternative Flow:

1. In step 3), if the user input a wrong username-password combination:

2. The system displays an error message and ask the user to try again

3. Back to step 2)

**Use Case: Create Account**

Actor: Not logged in User

Precondition: User is not logged in

Triggers: User, who have not logged in, come to the webpage

Success Condition: The user creates an account get redirected to home page

Failure End Condition: The user failed to create an account , back to login/register

Use Case Basic Flow:

1) The user comes to the product website

2) The system displays the login input box and the create account input box

3) The user input his/her name, email, and desired password in the create account input box then click “Create Account”

4) An account is successfully created: the user becomes a “logged in user” and the system displays the application home page

Alternative Flow:

1. In step 3), if the user input an email address that has been used before:

2. The system displays an error message and ask the user to try again

3. Back to step 2)

**Use Case: View Summary (Home)**

Actor: Logged in User

Precondition: User is logged in

Triggers: User come to the webpage or clicked on “Home” Tab

Success Condition: The user see the summary of tasks and bills in all his/her groups

Failure End Condition: Page fails to load

Use Case Basic Flow:

1) The user come to the webpage or clicked on “Home” Tab

2) The system displays a summary of user’s tasks list, a calendar that contains tasks and bills information on all the user’s group, and a notification bar displaying the most recent news

3) The user may continue to use case “Group Management”, “View Tasks” or “View Bills”

**General Use Case: View Group/Group Management**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on “Group Management”

General Use Case Basic Flow:

1) The user selected the “Group Management” Tab

2) The system displays the current groups that the user is in, and displays groups that the user being invited to

3) The user may choose to continue to a sub use case

**Sub Use Case: Create New Group**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on “Create New Group”

Success Condition: The user defined group is created

Failure End Condition: Nothing changed from the start

Use Case Basic Flow:

1) The user selected “Create New Group” on Group Management page

2) The system displays a popup window, prompt for user input on group name, description, and invite group members

3) The user inputs desired group name, associated description, invite members to the group, and clicked “Add”

4) The popup window disappears and the system displays the Group Management page, with the new group created displayed

5) The system stores the new group information

6) Step 1-5 may be repeated

Alternative Flow:

1. in step 3), if the desired group name is already a group name of one of the user’s groups:

2. The system displays “Group Name Exist” message when user clicks “Add”

3. The user clicks “OK” and changed a group name not used before

4. Continue to Step 4)

**Sub Use Case: Accept Group Invite**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on “Accept Invite”

Success Condition: The user join the group inviting him/her

Failure End Condition: Nothing changed from the start

Use Case Basic Flow:

1) The user checked the checkboxes next to the groups inviting him/her then clicked “Accept Invite”

2) The system displays a success message and display the Group Management page with the inviting groups moved to the “My groups” section

Alternative Flow:

1. in step 1), if the user did not check any group then clicked “Accept Invite”:

2. The system displays an error message and ask the user to try again

3. Back to step 1)

**General Use Case: View Task**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on “My Tasks”

General Use Case Basic Flow:

1) The user selected the “My Tasks” Tab

2) The system displays the list of tasks, with brief info of each, needs to be done by the user

3) The user may choose to continue to a sub use case

**Sub Use Case: Add Task**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on “Add Task”

Success Condition: The user defined task is created

Failure End Condition: Nothing changed

Use Case Basic Flow:

1) The user selected “Add Task” on “My Tasks” page

2) The system displays a popup window, prompt for user to select group that the task belongs to, name, date and description of task, and members responsible for the task

3) The user input the requested information

4) The system displays a preview of what it will look like on the calendar

5) The user select “Add”

6) The pop up window disappears and the system displays the “My Task” page

7) The system stores the new task information

8) Step 1 - 6 may be repeated

Alternative Flow:

1. in 5), if the user put in a task name already exist on that day:

2. The system displays an error message and ask the user to try again

3. Back to step 2

**Sub Use Case: Finish Task**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on a task name

Success Condition: The user defined task is marked as “Finished”

Failure End Condition: Nothing changed

Use Case Basic Flow:

1) The user selected one of the tasks from his “My Tasks” list

2) The system displays a popup window, displaying detailed information about the task including task name, description, participating members, and date.

3) The user clicks “Finished”

4) The pop up window disappears and the system displays the “My Task” page

5) The task will be marked as “finished”

6) Step 1-5 may be repeated

**General Use Case: View Bills**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on “My Bills”

General Use Case Basic Flow:

1) The user selected the “My Bills” Tab

2) The system displays the list of bills related to the user, with brief info of each and a bill summary

3) The user may choose to continue to a sub use case

**Sub Use Case: Add Bills**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on “Add Bills”

Success Condition: The user defined bill is created

Failure End Condition: Nothing changed

Use Case Basic Flow:

1) The user selected “Add Bill” on “My Bills” page

2) The system displays a popup window, prompt for user to select group that the bill belongs to, name, date, amount and description of bill, members involved and which member paid

3) The user input the requested information

4) The system displays a preview of what it will look like on the calendar

5) The user select “Add”

6) The pop up window disappears and the system displays the “My Bill” page

7) The system stores the new bill information

8) Step 1 - 6 may be repeated

**Sub Use Case: Paid Bill**

Actor: Logged in User

Precondition: User is logged in

Triggers: User clicked on a bill name

Success Condition: The user defined bill is marked as “Paid”

Failure End Condition: Nothing changed

Use Case Basic Flow:

1) The user selected one of the tasks from his “My Bills” list

2) The system displays a popup window, displaying detailed information about the bill including bill name, description, participating members, amount and date.

3) The user clicks “Paid”

4) The pop up window disappears and the system displays the “My Bills” page

5) The bill will be marked as “Paid”

6) Step 1-5 may be repeated