

NETime Planner
Logo placeholder



Joshua Kanagasabai: 11
Kevin DeMars: 15.5
Trenton Strickland: 10
Samuel Kim: 11
Eric Jaroszewski: 9
Max Cadeddu: 11

Git Link (Issue tracker): <https://github.com/KevinDeMars/netime-planner>
Website Link: <https://KevinDeMars.github.io/netime-planner>

What is NETime Planner?

NETime (pronounced "any time") Planner is an social planner focused on students.

In addition to the basic actions of managing personal deadlines and activities, it supports sharing schedules and collaborating on schedules with groups. Our goal is to make managing your schedule simple and convenient, with helpful features such as automatically appending events to your next free time.

The features don't stop as soon as you've entered your info into your schedule. NETime Planner provides features to analyze your schedule, giving you feedback on what percentage of time you have left for deadlines, how much free time you have each week, and more.

If you don't want to spend much time entering information, this application is still for you. Advanced features that require entering more data are entirely optional, and you can make your schedule as minimal as you want.

NETime Planner

Feb 9, 2020

CSI 3471

<http://>

Project manager

Project dates

Jan 28, 2020 - Feb 12, 2020

Completion

0%

Tasks

18

Resources

6

Gantt Diagram for first iteration of software project.

Tasks

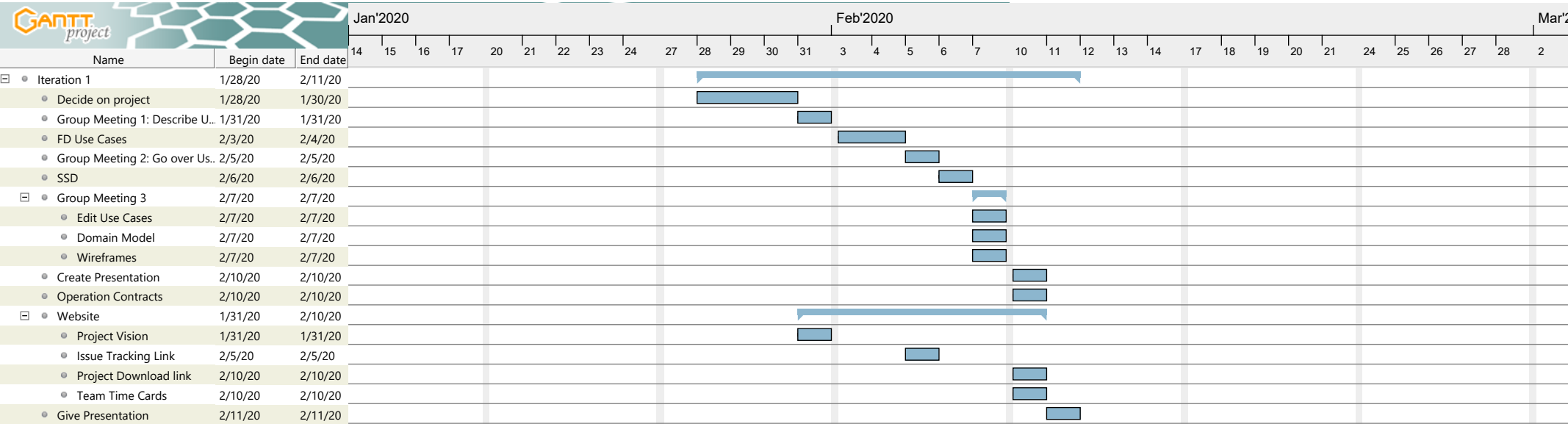
2

Name	Begin date	End date
Iteration 1	1/28/20	2/11/20
Decide on project	1/28/20	1/30/20
Group Meeting 1: Describe Use Cases	1/31/20	1/31/20
FD Use Cases	2/3/20	2/4/20
Group Meeting 2: Go over Use Cases	2/5/20	2/5/20
SSD	2/6/20	2/6/20
Group Meeting 3	2/7/20	2/7/20
Edit Use Cases	2/7/20	2/7/20
Domain Model	2/7/20	2/7/20
Wireframes	2/7/20	2/7/20
Create Presentation	2/10/20	2/10/20
Operation Contracts	2/10/20	2/10/20
Website	1/31/20	2/10/20
Project Vision	1/31/20	1/31/20
Issue Tracking Link	2/5/20	2/5/20
Project Download link	2/10/20	2/10/20
Team Time Cards	2/10/20	2/10/20
Give Presentation	2/11/20	2/11/20

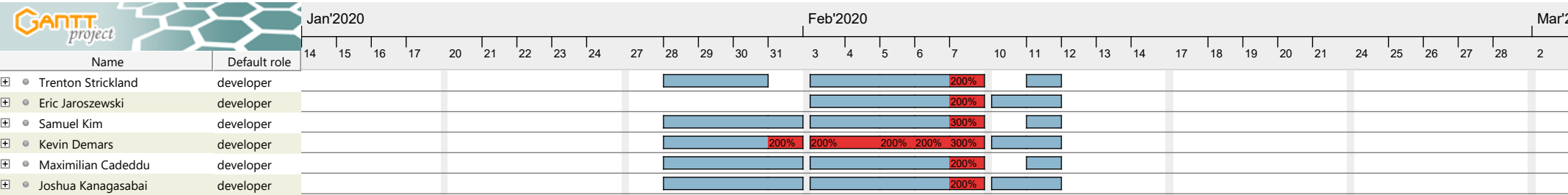
Resources

Name	Default role
Trenton Strickland	developer
Eric Jaroszewski	developer
Samuel Kim	developer
Kevin Demars	developer
Maximilian Cadeddu	developer
Joshua Kanagasabai	developer

Gantt Chart



Resources Chart



UC01
Create Deadline

UC02
Edit Deadline

UC03
Remove Deadline

UC20
Create Activity

UC21
Edit Activity

UC22
Remove Activity

UC04
Assign Work Time

UC05
Auto-Append

UC06
Generate To-Do List

UC07
Create Group

UC08
Delete Group

UC09
Edit Group

UC10
Edit Group Permissions

UC11
View Schedule

UC12
Create Profile

UC13
Edit Profile

UC14
View time spent on specific task

UC15
Show Time Left for Specific Task

UC16
Calculate Free Time

UC17
Share Schedule

UC18
Share Event

UC19
Edit Privacy Setting



Assign Work Time

Use case: Assign Work Time

Scope: NETime Planner Application

Level: User goal

Primary Actor: User

Stakeholders and Interests:

- User: Wants accurate emplacement of Work Time and pre-existing events in his/her schedule to remain untampered.
- System: Wants precise and correctly formatted data to perform time computations and apply schedule operations.

Preconditions: User has a created schedule.

Postconditions: Work Time is assigned to schedule, Start and End Times are saved and emplaced.

Basic Flow:

1. User navigates to schedule
2. User requests schedule to 'Assign Work Time'.
3. Schedule prompts User to input a 'Start Time' and 'End Time'.
4. User inputs 'Start Time' and 'End Time'.
5. Schedule prompts User to input the 'Day' in which the User would like to assign this time frame.
6. User inputs 'Day'.
7. Schedule checks if 'Start Time' to 'End Time' overlaps any existing event

•

8. Schedule records and emplaces 'Start Time' and 'End Time' on the specified 'Day' within the User's current week schedule

Extensions:

- a.) If System recognizes an event is already emplaced within Assigned Work Time
- i.) User can cancel assignment
 - ii.) User can auto emplace the events(s) within the requested time.
- b.) If assigned work time already exists for given day
- i.) System reports existing Work Time frame.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Auto-Append

Use case: Auto - Append

Scope: NETime Planner Application

Level: User goal

Primary Actor: User

Stakeholders and Interests:

- User: Wants accurate emplacement of events and pre-existing events in his/her schedule to remain untampered.
- System: Wants precise and correctly formatted data to perform time computations and apply schedule operations.

Preconditions: User has a created schedule and assigned Work Time to it.

Postconditions: Event is inserted into the User's schedule at the next available time within the selected day's Assigned Work Time time-frame.

Basic Flow:

1. User navigates to schedule
2. User selects an existing event OR a newly created event
3. User requests to 'Auto-Append' the event.
4. Schedule prompts User to input the 'Day' in which the event should be appended.
5. User inputs 'Day'.
6. Schedule navigates to specified 'Day', then calculates the difference between the day's Assigned Work Time 'End Time' and the time of the last occurring event between 'Start Time' and 'End Time'.

6a. The schedule adds 15 minutes to the calculated difference for spare time / travel time.

6b. If the Event to be auto-appended is less than (the difference + 15 minutes), the event is emplaced in the schedule at (the difference + 15 minutes).

Extensions:

a.) If the System recognizes Assigned Work Time is full:

i.) The user can cancel operation

ii.) The user can request to extend Assigned Work Time to ending time of Event to be appended

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Calculate Free Time

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

· User A (logged in user): Wants to check how much free time they have scheduled on their planner

Preconditions: User A has logged into their profile

Postconditions: User A is shown much free time they have

Basic Flow:

1. User A indicates they want to view their free time
2. User A enters Time period
3. User A submits request to the Planner System.
4. System searches parses the given time period
5. System adds up anytime that is not otherwise occupied by an event or task
6. System returns data to User A.

Extensions:

1a. Invalid time period(not formatted correct or not in planner)

1. System signals the time period is invalid and for what reason.
2. Operation is canceled.

Frequency of Occurrence: Medium- useful mostly for heavy users as much data has to be entered in order for free time calculation to be accurate.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Create Activity

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

- User A (User changing schedule): Wants User B's schedule updated for organizational/planning purposes
- User B (User whose schedule is changed): Wants their schedule to be updated for organizational/planning purposes. May want to be notified if different person than User A.
- Other users: Want to see whether user B is available at a given time, if permitted.

Preconditions: User A is authenticated and has permissions to modify User B's schedule.

Postconditions: Activity is created and saved.

Basic Flow:

1. User A requests to add an activity to User B's schedule.
2. System prompts User A for activity details.
3. User A fills in the form with activity details.
4. User A submits the data to System.
5. System records the new activity.
6. System shows User A the updated schedule (see View Schedule UC).

Extensions:

- 1a. User B isn't found
 1. System signals error to User A.
 2. Event creation is cancelled.
- 4a. The submitted data is invalid.
 1. System signals error to User A.
 2. Event creation is cancelled.
- 5a. User A is different than User B, and User B wants to be notified when others change their schedule.
 1. System notifies User B.

Frequency of Occurrence: Could happen concurrently with many users.

Special Requirements:

- Must work locally if the remote server is unavailable.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Create Deadline

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

- User A (User changing schedule): Wants User B's schedule updated for organizational/planning purposes
- User B (User whose schedule is changed): Wants their schedule to be updated for organizational/planning purposes. May want to be notified if different person than User A.
- Other users: Want to see whether user B is available at a given time, if permitted.

Preconditions: User A is authenticated and has permissions to modify User B's schedule.

Postconditions: Deadline is created and saved.

Basic Flow:

1. User A requests to add a deadline to Schedule
2. System prompts User A for deadline details.
3. User A fills in the form with deadline details.
4. User A submits the data to System.
5. System records the new deadline.
6. System shows User A the updated schedule (see View Schedule UC).

Extensions:

- 1a. User B isn't found
 1. System signals error to User A.
 2. Event creation is cancelled.
- 4a. The submitted data is invalid.
 1. System signals error to User A.
 2. Event creation is cancelled.
- 5a. User A is different than User B, and User B wants to be notified when others change their schedule.
 1. System notifies User B.

Frequency of Occurrence: Could happen concurrently with many users.

Special Requirements:

- Must work locally if the remote server is unavailable.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Create Group

Use Case:

Create a Group

- Scope- NERTime Planner website
- Level: User Goal
- Primary Actor: User
- Stakeholders and Interests:
 - User - Wants reliable, fully functional calendar for planning, and wants to be able to share important tasks with groups to help better plan for how to handle said tasks.
 - Peers/Friends - Want to stay up to date with important tasks being created by user, and be able to contribute in meaningful ways to the tasks at hand.
- Preconditions: The user has created and logged into their profile.
- Success Guarantee (or Postconditions): A group is created. The user is within the group, and has all available permissions to edit the group.
- Main Success Scenario (Basic Flow):
 1. User wishes to create a new group.
 2. User creates a new group.
 3. User provides a name for the new group.
 4. User provides profile information for a peer to add to the group.
User repeats step 5 until they indicate they are done.
 5. The group is created.
 6. Invitations are sent to all peer profiles provided to join the group.
- Extensions (Alternate flows)
 - a. At any time, the user decides to cancel the group.
 - i. The User presses the "Cancel" button.
 - ii. The group creation is canceled.
 - b. At any time, the system fails.
 - i. The system is restarted.
 - ii. The user restarts the group creation process.
 - c. At any time, the user wishes to go back and modify data in a previous step.
 - i. The user presses the in-site "back" button.
 - 4a. The user does not provide a name for the group.
 - i. The system informs the user a name has not been entered.
 - ii. The system asks the user if they wish to name the group.
 1. The user declines.
 2. The group is named "Group[num]" depending on the number of generic groups that exist.

- 5a. The user does not provide any profiles to the group.
 - i. The system asks if the user wishes to add any profiles to the group at the current time.
 - 2.
 - 1. The user declines.
 - 2. The group is created and no invitations are sent.
- 5b. The user wishes to remove from the invite list a peer they planned to invite.
 - i. The user removes the peer's profile from the list.
- Special Requirements:
 - Group creation and invites by system should be handled within 30 seconds.
- Technology and Data Variations List:
 - Servers to permanently hold user inputted group information and requests.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Create Profile

Scope: NETime Planner Application

Level: User goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to create a profile.

Preconditions: A user without a profile exists.

Postconditions: A profile unique to the User has been created. The profile will be usable even after the application has been exited. User has been briefed on the UI following profile creation.

Basic Flow:

1. User indicates they want to create a new profile.
2. User is prompted to enter login credentials
3. User enters their login credentials (username, email, password)
4. A new profile has been created and registered.
5. Application directs User to edit their profile.
6. User finishes editing their profile

→ T H E E N D O F T H E P R O J E C T

7. Tooltips are displayed to introduce the UI to the User.
8. The User is notified of a means to seek a documentation/usage guide.
9. User dismisses the notice.
10. Application reverts to default behavior

Extensions

- 2a. The entered email is already attached to an existing profile or is invalid.
 1. An error message is displayed, prompting the User to enter a different email address.

Loop steps 2-3 until a working email address has been entered.

- 2a. The entered password is less than 8 characters long.
 1. An error message is displayed, prompting the User to enter a new password at least 8 characters in length.

Loop steps 2-3 until a working password has been entered.

- 2a. The entered username contains invalid characters
 1. An error message is displayed, prompting the User to enter a new username that does not contain invalid characters. Violating character is labeled as such for the user to see

Character is labeled as such for the user to see.

Loop steps 2-3 until a working username has been entered.

Special Requirements:

- If there is going to be a usage guide, it will need to be intelligible.

Frequency of Occurrence:

- Could be nearly continuous across many different machines.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Delete Group

Delete a Group

- Scope - NETime Planner Website
- Level - User Goal
- Primary Actor - Group Owner
- Stakeholders and Interests -
 - Group Owner - Wants easy access to all functions of a group, and wants to make sure rest of group is fully informed of his decisions.
 - Peers/Group Members - Want to be up to date with all information regarding the group, and wish to contribute to tasks quickly and efficiently.
- Preconditions - Group Owner has created and logged into their profile, and has created at least one group.
- Success Guarantee(Postconditions) - The group selected by the owner is removed.
- Main Success Scenario -
 1. The group owner decides he no longer needs a particular group.
 2. Group owner clicks on the group he wishes to remove.
 3. Group owner goes to the delete group option.
 4. The system asks the owner if he wishes to delete the group.
 5. The group owner confirms he wishes to delete the group.
 6. All tasks within the group are deleted.
 7. All peers and the group owner are removed from the group.
 8. The group is deleted.
- Extensions -
 - a. At any time, the system fails.
 - i. The system restarts and the group deletion process is cancelled.
 - 1-4a. At any time, the owner decides to go back one step.
 - i. The owner clicks the in-site “back” button.
 - 6a. The owner declines the request to delete the group.
 - i. The group deletion is canceled.
 - 10a. Within 48 hours of deletion, the group owner wishes to restore the group.
 - i. Group owner clicks on the deleted group.
 1. Group owner decides against restoring the group.
 2. Group owner clicks in-site “back” button.
 - iii. Group owner restores the group.

- Special Requirements:
 - Group deletion by system should be handled in 48 hours for user convenience.
- Technology and Data Variations List:
 - Servers to permanently hold user inputted group information and requests.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Edit Activity

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

- User A (User changing schedule): Wants User B's schedule updated for organizational/planning purposes
- User B (User whose schedule is changed): Wants their schedule to be updated for organizational/planning purposes. May want to be notified if different person than User A.
- Other users: Want to see whether user B is available at a given time, if permitted.

Preconditions: User A is authenticated and has permissions to modify User B's schedule.

Postconditions: Activity is modified and saved.

Basic Flow:

1. User A requests to edit Activity in User B's schedule.
2. System displays form containing Activity's details.
3. User A edits the form as needed.
4. User A submits data to System.
5. System shows User A updated schedule (See View Schedule UC).

Extensions:

1a. User B or Activity doesn't exist.

1. System signals error to User A.
2. Event editing is cancelled.

4a. The submitted data is invalid.

1. System signals error to User A.
2. Event editing is cancelled.

5a. User A is different than User B, and User B wants to be notified when others change their schedule.

1. System notifies User B.

Frequency of Occurrence: Could happen concurrently with many users.

Special Requirements:

- Must work locally if remote server is unavailable.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Edit Deadline

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

- User A (User changing schedule): Wants User B's schedule updated for organizational/planning purposes
- User B (User whose schedule is changed): Wants their schedule to be updated for organizational/planning purposes. May want to be notified if different person than User A.
- Other users: Want to see whether user B is available at a given time, if permitted.

Preconditions: User A is authenticated and has permissions to modify User B's schedule.

Postconditions: Deadline is modified and saved.

Basic Flow:

1. User A requests to edit Deadline in User B's schedule.
2. System displays form containing Deadline's details.
3. User A edits the form as needed.
4. User A submits data to System.
5. System shows User A updated schedule (See View Schedule UC).

Extensions:

1a. User B or Deadline doesn't exist.

1. System signals error to User A.
2. Event editing is cancelled.

4a. The submitted data is invalid.

1. System signals error to User A.
2. Event editing is cancelled.

5a. User A is different than User B, and User B wants to be notified when others change their schedule.

1. System notifies User B.

Frequency of Occurrence: Could happen concurrently with many users.

Special Requirements:

- Must work locally if remote server is unavailable.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Edit Group

Use Case:

Edit a Group

- Scope- NERTime Planner website
- Level: User Goal
- Primary Actor: User
- Stakeholders and Interests:
 - User - Wants easy access to all functions of a group, and wants to make sure rest of group is fully informed of his decisions.
 - Peers/Group Members - Want to be up to date with all information regarding the group, and wish to contribute to tasks quickly and efficiently.
- Preconditions - User has created and logged into their profile, is in at least one group, and has the permissions necessary to edit parts of the group.
- Success Guarantee - The group is modified to the user's liking.
- Main Success Scenario -
 1. The User wishes to modify member list of the group.
 2. User clicks on desired group.
 3. User provides profile information for a peer to add or remove.
User repeats step 3 until they indicate they are done.
 4. Added group members are sent invitations.
 5. Members specified to be deleted are removed from the group.
 6. User wishes to modify group's task list.
 7. User clicks on desired group.
 8. User creates a new task, or selects task to remove.
 9. User indicates he is done.
 10. The specified task is created or removed, and the assigned group members are notified.
- Extensions -
 - a. At any time, the system fails.
 - i. The system restarts, and the last unsaved edit is canceled.
 - b. At any time, the user wishes to go back, either to cancel an unsaved action or to modify data entered in a previous step.
 - i. The user clicks on the in-site "back" button.
- Special Requirements:
 - Once saved, edits done by system should be handled within 30 seconds.
- Technology and Data Variations List:

- Servers to permanently hold user inputted group information and requests.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Edit Group Permissions

Use Case: Edit Group Permissions

- Scope- NERTime Planner website
- Level: User Goal
- Primary Actor: User
- Stakeholders and Interests:
 - Group Owner - Wants easy access to all functions of a group, and wants to make sure rest of group is fully informed of his decisions.
 - Peers/Group Members - Want to be up to date with all information regarding the group, and wish to contribute to tasks quickly and efficiently.
- Preconditions - Group Owner has created and logged into their profile and is in at least one group, and does not wish to modify his own permissions.
- Success Guarantee - The permissions of individuals within the group are modified to the owner's liking.
- Main Success Scenario -
 1. The Owner wishes to modify the permissions for group members.
 2. Owner clicks on desired group.
 3. Owner goes to Permissions setting.
 4. Owner finds the group member whose permissions he wishes to modify.
 5. Owner wishes to modify the group member's permissions to either add other group members, or to edit group tasks.
 6. Owner checks off the boxes for approved permissions, and clears the boxes for denied permissions.
 7. The owner indicates he is done.
 8. The Owner wishes to transfer ownership of the group to another member.
 9. Owner performs steps 2-3.
 10. System asks Owner if he wishes to transfer ownership to the selected user.
 11. Owner confirms.
 12. Notification is sent to specified group member for request to transfer ownership.
 13. Other group member confirms.
 14. Ownership is transferred from old owner to group member.
 15. Former owner loses permission to set permissions.

16. New Owner gains all available permissions to modify the group.
- Extensions-
 - a. At any time, the system fails.
 - i. The system restarts, and the permissions are reset to their last known state.
 - b. At any time, the user wishes to go back, either to cancel an unsaved action or to modify data entered in a previous step.
 - i. The user clicks on the in-site “back” button.
 - 1 or 11a. The Owner is the only member of the group.
 - i. The system will state that there are no other group members with modifiable permissions.
 - 19a. The group member denies.
 - i. System asks group member if he does not want ownership of the group.
 - 2. 1. Group member denies. Back to 19a.
 - iii. Group member confirms.
 - iv. Ownership of the group remains with original owner.
 - Special Requirements:
 - Once saved, edits done by system should be handled within 30 seconds.
 - Technology and Data Variations List:
 - Servers to permanently hold user inputted group information and requests.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Edit Privacy Setting

Scope: NETime Planner Application

Level: User Goal

Primary Actor: User A

Stakeholders and Interests:

User A

- A user of the application that would like to change the privacy settings of one of their schedules.

User B

- A user that views the profile of User A after the change in the privacy setting of one of their schedules.

Precondition: The user has navigated to the list of schedules in their profile.

Postcondition: The privacy setting of the schedule has been set to one of four possible states: View/Edit, Only View, Only View Free/Not Free, and Private.

Basic Flow:

1. User A right clicks the schedule of which they would like to change the privacy setting.
2. The system displays the context menu, where "Privacy" is one of the options that appears.
3. User A clicks on "Privacy" and the context menu is extended to the right to show the four privacy settings. By default, the schedule is set to Private, with a checkmark existing to the right of that option. The three other options displayed

are View/Edit, Only View, and Only View Free/Not Free.

4. User A clicks on the desired option to change the privacy setting of the schedule to that option.
5. The system updates the privacy setting of the schedule to that option.
6. User B views the profile of User A.
7. What User B sees of the schedule is changed depending on the option that was chosen by User A. (See the extensions for the differences between different options.)

Extensions

- 7.a. If the privacy setting was changed to “Only View”,
 1. User B is able to open and view the schedule, as well as see updates.
- 7.b. If the privacy setting was changed to “Only View Free/Not Free”,
 1. User B is able to open and view the schedule and see updates, but they can only see blocks of time when User A is having an event, which are labeled “Not Free”.
- 7.c. If the privacy setting was changed to “View/Edit”,
 1. If the schedule is not shared with User B,
 - 1.1. The privacy setting behaves identically to the “Only View” option. User B is able to open and view the schedule, as well as see updates.
 2. If the schedule is shared with User B,
 - 2.1 User B is able to open, view, and see updates with the schedule, as well as edit the schedule.

Frequency of Occurrence:

This happens frequently, since it is often used when sharing schedules.

Special Requirements:

The remote server must be available to update the setting for other users.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Edit Profile

Scope: NERTime Planner Application

Level: User goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to edit an existing profile.

Preconditions: User and a profile exists.

Postconditions: Profile has been edited and pertinent information has been updated server-side.

Basic Flow:

1. User indicates they want to edit their profile.
2. Profile is displayed.
3. Edit-able text fields are displayed with their current value, next to a text box where the user can enter a new value.
4. The user changes their email field.
5. The option to delete specific categories or add new ones is supplied.
6. User deletes a category.

7. User is prompted to confirm they wish to delete the category.
8. User clicks a button labeled "Yes".
9. The category is marked to be removed *iff* Step 15 occurs.
10. User adds a category.
11. User inputs a name for the category.
12. The category is temporarily created, but not added to the profile until Step 15 occurs.
13. User clicks a button labeled "finish".
14. User is prompted to confirm they wish to make these changes.
15. User clicks a button labeled "Yes".
16. The changes are verified.
17. The profile window is dismissed.
18. Application reverts to default behavior.

Extensions

- 6a. User is prompted to confirm they wish to delete this category.
 1. A User clicks a button labeled "No".
 2. Confirmation prompt is dismissed.

3. Category is not marked for deletion.

10a. User clicks a button labeled “cancel”

1. User is prompted to confirm they wish to cancel. All unsaved changes will be lost.

1a. User clicks a button labeled “Yes”

1. Profile is not changed.

Jump to step 14

1b. User clicks a button labeled “No”

1. Profile is not changed.

2. “Cancel” Prompt is dismissed.

15a. User clicks a button labeled “No”

1. Changes are not made.

2. Any categories marked for deletion/creation remain as such until step 15 occurs.

3. Prompt is dismissed.

4. User remains on the Profile page in edit mode.

5. User continues to make changes.

or user continues to make changes.

16a. One or more text fields contain invalid characters.

1. User is directed back to the edit page as they left it.
2. Error message is displayed.
3. Offending fields are highlighted.

Special Requirements:

- While editing. Of the Editable fields; none, some, or all can be changed, and the process should complete successfully regardless.

Frequency of Occurrence:

- Could be nearly continuous across many different machines.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Generate To-Do List

Use case: Generate Today's To-Do List

Scope: NETime Planner Application

Level: User goal

Primary Actor: User

Stakeholders and Interests:

- User: Wants accurate list of events and assignments with their times and pre-existing events in his/her schedule to remain untampered.
- System: Wants precise and correctly formatted data to perform time computations and apply schedule operations.

Preconditions: User has a created schedule with events emplaced upon it.

Postconditions: User is presented with List of events with their associated times, in the order in which they occur on the schedule.

Basic Flow:

1. User navigates to Profile.
2. User requests Generate Today's To Do List.
3. System navigates to the current day within schedule.
4. System generates an empty list of events.
5. From clock-time 00:00:00 to 23:59:59 of the calculated date, System inserts each event emplaced on schedule into the List.
6. System displays the list.

Extensions:

EXTENSIONS.

a.) User wants a day other than the current day

i.) System requests specified day

ii.) User inputs day

ii.) System follows basic flow steps 3-5

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Remove Activity

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

- User A (User changing schedule): Wants User B's schedule updated for organizational/planning purposes
- User B (User whose schedule is changed): Wants their schedule to be updated for organizational/planning purposes. May want to be notified if different person than User A.
- Other users: Want to see whether user B is available at a given time, if permitted.

Preconditions: User A is authenticated and has permissions to modify User B's schedule.

Postconditions: Activity is removed from User B's schedule.

Basic Flow:

1. User A requests to delete Activity in User B's schedule.
2. System deletes Activity.
3. System shows User A the updated schedule (see View Schedule UC).

Extensions:

- 1a. User B or Activity doesn't exist.
 1. System signals error to User A.
 2. Event deletion is cancelled.
- 2a. User A is different than User B, and User B wants to be notified when others change their schedule.
 1. System notifies User B.

Frequency of Occurrence: Could happen concurrently with many users.

Special Requirements:

- Must work locally if the remote server is unavailable.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Remove Deadline

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

- User A (User changing schedule): Wants User B's schedule updated for organizational/planning purposes
- User B (User whose schedule is changed): Wants their schedule to be updated for organizational/planning purposes. May want to be notified if different person than User A.
- Other users: Want to see whether user B is available at a given time, if permitted.

Preconditions: User A is authenticated and has permissions to modify User B's schedule.

Postconditions: Deadline is removed from User B's schedule.

Basic Flow:

1. User A requests to delete Deadline in User B's schedule.
2. System deletes Deadline.
3. System shows User A the updated schedule (see View Schedule UC).

Extensions:

- 1a. User B or Deadline doesn't exist.
 1. System signals error to User A.
 2. Event deletion is cancelled.
- 2a. User A is different than User B, and User B wants to be notified when others change their schedule.
 1. System notifies User B.

Frequency of Occurrence: Could happen concurrently with many users.

Special Requirements:

- Must work locally if the remote server is unavailable.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Share Event

Scope: NETime Planner Application

Level: User Goal

Primary Actor: User A

Stakeholders and Interests:

User A

- A person that is interested in sharing their event with another user or a group of users.

Event Recipient(s)

- The user or group of users that the event will be shared with.

Precondition: The user has opened the schedule containing the event that he or she intends to share.

Postcondition: The event has been added to the schedules of one or more selected users, allowing them to view the event on that schedule when they use the application.

Basic Flow:

1. User A right-clicks an event which appears on his or her schedule.
2. In the context menu, an option called "Share Event" appears.
3. The user clicks the "Share Event" option.
4. The system displays a pop-up window for selecting the user or group of

users with whom User A would like to share the event.

5. User A types the name of a recipient in the search bar and presses enter or clicks “Search”.

6. The drop-down menu below the search bar, in response, lists existing profiles which are retrieved from the search query.

7. The profile of the recipient appears in the drop-down menu.

8. User A selects the profile to be a recipient of the event. The system indicates this in the popup menu, showing a list of all of the recipients added so far.

9. User A repeats steps 5 - 8 until they have selected all of the intended recipients.

10. User A clicks the “Share” button at the bottom of the popup menu to share the event with all of the selected recipients.

11. The system adds the event to the schedule of each recipient.

12. All recipients, when viewing their personal schedules, see the event that was added by User A.

Extensions:

1-2.a. The user can instead navigate to the schedule’s event list to share an event.

1. After finding the schedule’s event list, the User A right-clicks the

desired event to open the context menu.

6.a. If User A could not find the recipient from the search query,

1. The recipient may not have the application installed.

- 1.1. In that case, User A can email the recipient letting them know of the event.

2. The setting of the recipient's profile may be set to private.

- 2.1. In that case, User A can ask for the recipient's username and manually enter it to add that person to the list of recipients.

3. The search query may have failed to associate the entered name with the profile that User A was looking for.

- 3.1 In that case, User A can ask for the recipient's username and manually enter it to add that person to the list of recipients.

5-8.a. If the user would like to remove one of the profiles from the list of recipients,

1. The user hovers their mouse over the profile icon.

2. The user clicks the "X" button which appears to the top right of the profile icon.

Frequency of Occurrence:

This happens frequently, as this is one of the core aspects of the software's social element.

Special Requirements:

The remote server must be available.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Share Schedule

Scope: NETime Planner Application

Level: User Goal

Primary Actor: User A

Stakeholders and Interests:

User A

- A person that is interested in sharing their schedule with another user or a group of users.

Schedule Recipient(s)

- The user or group of users that the schedule will be shared with.

Precondition: The user has opened the schedule that he or she is intending to share.

Postcondition: The schedule has been shared with one or more selected users, allowing them to view the most updated version of the schedule from their computers when they open the application.

Basic Flow:

1. User A, while viewing the editing page after opening their schedule, navigates to the upper righthand corner of the screen and clicks a button labeled "Share".

2. The system displays a pop-up window for selecting the user or group of users with whom User A would like to share the schedule.

3. User A types the name of a recipient in the search bar and presses enter or clicks "Search".
4. The drop-down menu below the search bar, in response, lists existing profiles which are retrieved from the search query.
5. The profile of the recipient appears in the drop-down menu.
6. User A selects the profile to be a recipient of the schedule. The system indicates this in the popup menu, showing a list of all of the recipients added so far.
7. User A repeats steps 3 - 6 until they have selected all of the intended recipients.
8. User A clicks the "Share" button at the bottom of the popup menu to share the schedule with all of the selected recipients.
9. The system gives each recipient access to the schedule created by User A.
10. All recipients, when exploring their list of schedules in the application, are able to view the schedule that was sent from User A.

Extensions:

- 1.a. The user can instead navigate to their list of schedules, right click the desired schedule, and select "Share" from the context menu.
- 5.a. If User A could not find the recipient from the search query,
 1. The recipient may not have the application installed.

1.1. In that case, User A can email a pdf file of the schedule to the intended recipient.

2. The setting of the recipient's profile may be set to private.

2.1. In that case, User A can ask for the recipient's username and manually enter it to add that person to the list of recipients.

3. The search query may have failed to associate the entered name with the profile that User A was looking for.

3.1 In that case, User A can ask for the recipient's username and manually enter it to add that person to the list of recipients.

3-6.a. If the user would like to remove one of the profiles from the list of recipients,

1. The user hovers their mouse over the profile icon.

2. The user clicks the "X" button which appears to the top right of the profile icon.

Frequency of Occurrence:

This happens frequently, as this is one of the core aspects of the software's social element.

Special Requirements:

The remote server must be available.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

Show Time Left for Specific Task

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

· User A (logged in user): Wants to check how much time they have left to work on a specific task based on their planner

Preconditions: User A has logged into their profile

Postconditions: User A is displayed the time left on a specific task

Basic Flow:

1. User A indicates they want to view the time left to finish a task
2. User A enters
 1. Task name
3. User A submits request to the Planner System.
4. System searches for given task.
5. System finds task and calculates the time to due date.
6. System returns data to User A.

Extensions:

1a. Invalid task name(not found in Planner System)

1. System signals this task does not exist.
2. Operation is canceled.

1a. Task has included time schedule

1. System detects task has scheduled time allocation.
2. In addition to the time to due date, the System also returns total remaining time allocated before the task ends.

5a. Alter return format based on if start date is included

Frequency of Occurrence: Heavy- Many users will check their time left on tasks as a central part of the planner.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

[View Schedule](#)

Scope: NETime Planner Application

Level: User goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to create an application-generated view of the upcoming week or a specific day.

Preconditions: A user with a profile exists.

Postconditions: A view of the schedule specific to the User's profile has been generated within the specified time range (week/day) and displayed.

Basic Flow:

1. User indicates they want to view their schedule
2. The schedule is displayed to the User.
3. The User views the schedule.
4. The User terminates the display and the view of the schedule is destroyed.
5. The application reverts to default behavior

Extensions

- 1a. For the next week.

1. A less detailed “birds-eye-view” of the upcoming week is created.

- 1a. For a specific day.

1. A more detailed, daily schedule organized by hourly increments is created.

- 3a. The User has a profile, but has not inputted any information

1. A blank view is displayed alongside a prompt to the User to enter events or add events to their schedule.

- 4a. The User leaves the view open.

1. The application continues performing as usual, but some functionality may be inaccessible until the view is closed.

Special Requirements:

- Whatever functionality is “covered” by the view either needs to be relocated or auxiliary as far as the schedule is concerned. Users should be able to add/remove events while the schedule is open, and the schedule should update to reflect those changes.
- Group events should exist locally in the event the user is offline. They will need to be updated when the User goes back online to reflect any changes that may have occurred.

Frequency of Occurrence: A localized event that could happen sporadically.

Information

Rank Unspecified

ID

Status Unspecified

Justification

Primary Actors

Supporting Actors

View time spent on specific task

Scope: NETime Planner Application

Level: User goal

Primary Actor: User A

Stakeholders and Interests:

- User A (logged in user): Wants to check how much time they have worked on a specific task based on their planner

Preconditions: User A has logged into their profile

Postconditions: User A is displayed the time spent on a specific task

Basic Flow:

1. User A indicates they want to view their time spent on a task
2. User A enters
 1. Task name
 2. Time frame to check over, can also select global to get entire time spent.
3. User A submits request to the Planner System.
4. System searches for given task.

5. System displays time spent on task to User A

5. System finds task and calculates time spent within the given time frame.
6. System returns data to User A.

Extensions:

1a. Invalid task name(not found in Planner System)

1. System signals this task does not exist
2. Operation is canceled

1a. Invalid time frame(incorrectly formatted time frame entered)

1. System signals this task does not exist
2. Operation is canceled

1b. No time frame is entered

1. System assumes global time frame

Frequency of Occurrence: Medium- Mostly for users that heavily use the planner as it requires for users to have logged/planned their tasks heavily for accurate results.

Information

Rank Unspecified

ID

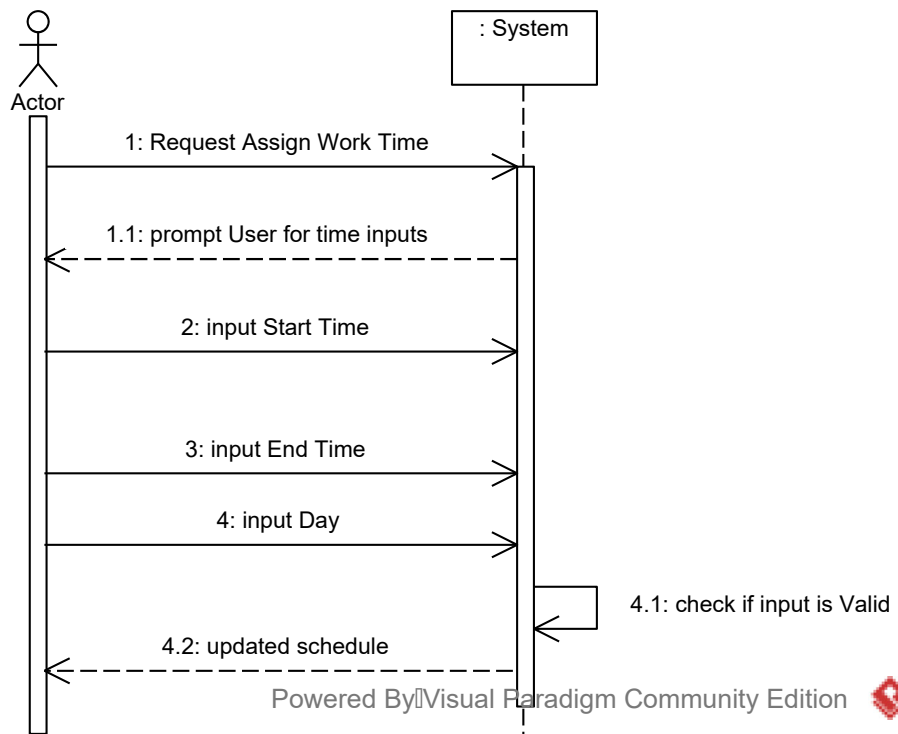
Status Unspecified

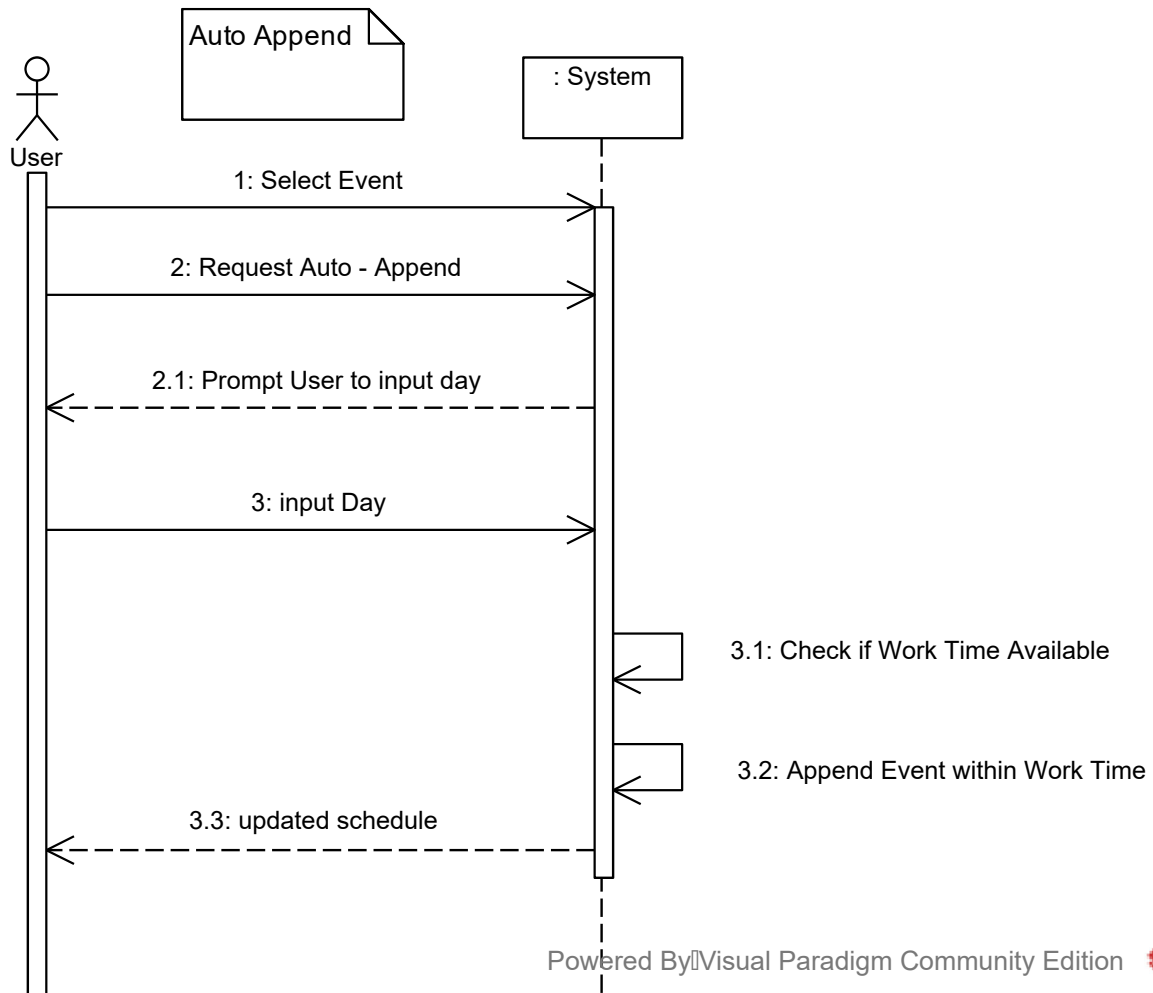
Justification

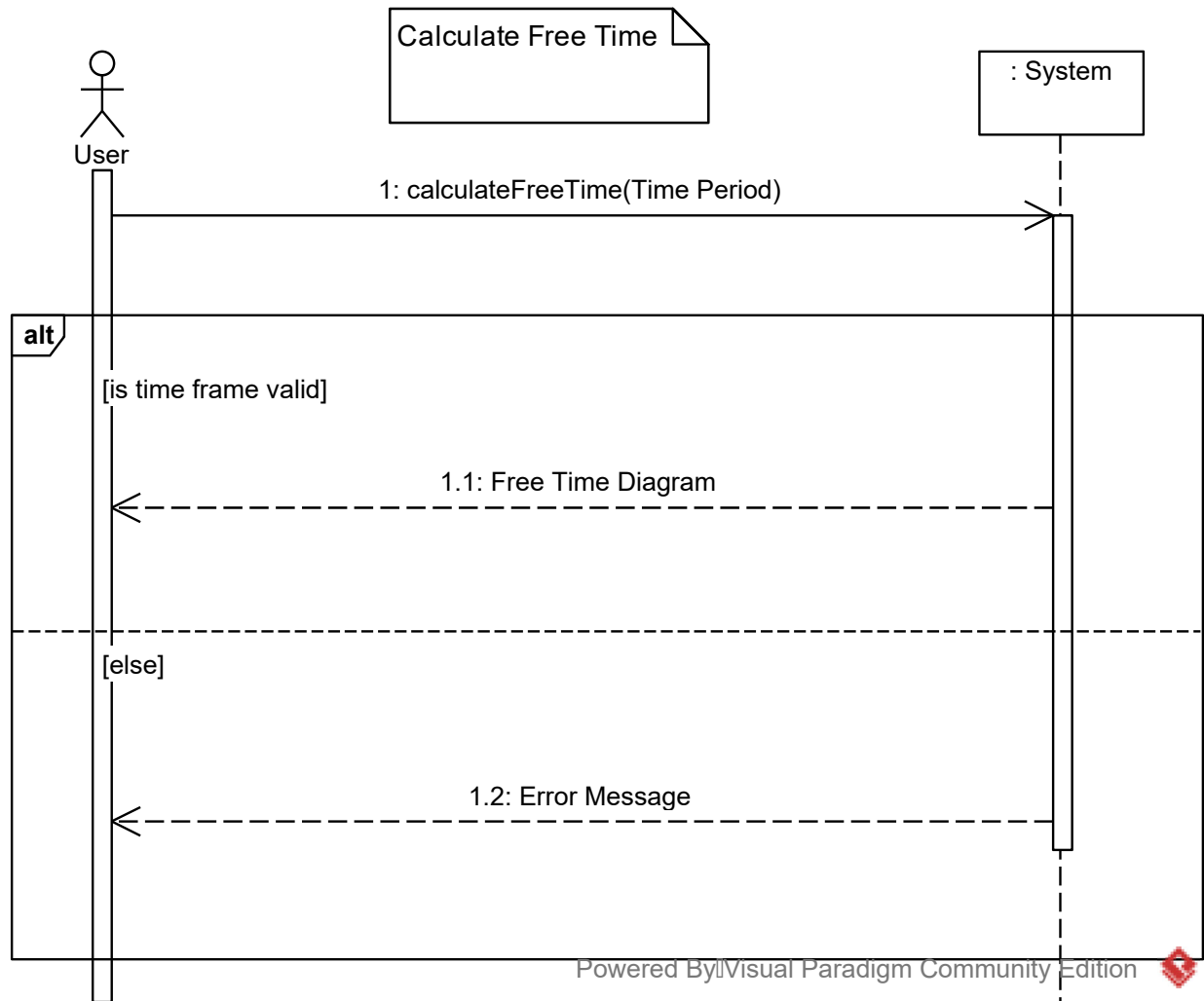
Primary Actors

Supporting Actors

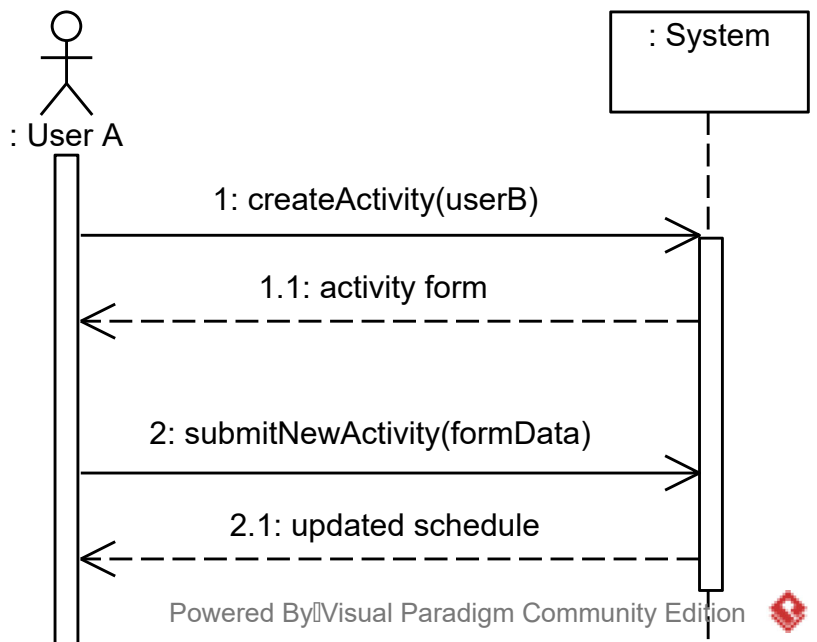
Assign Work Time

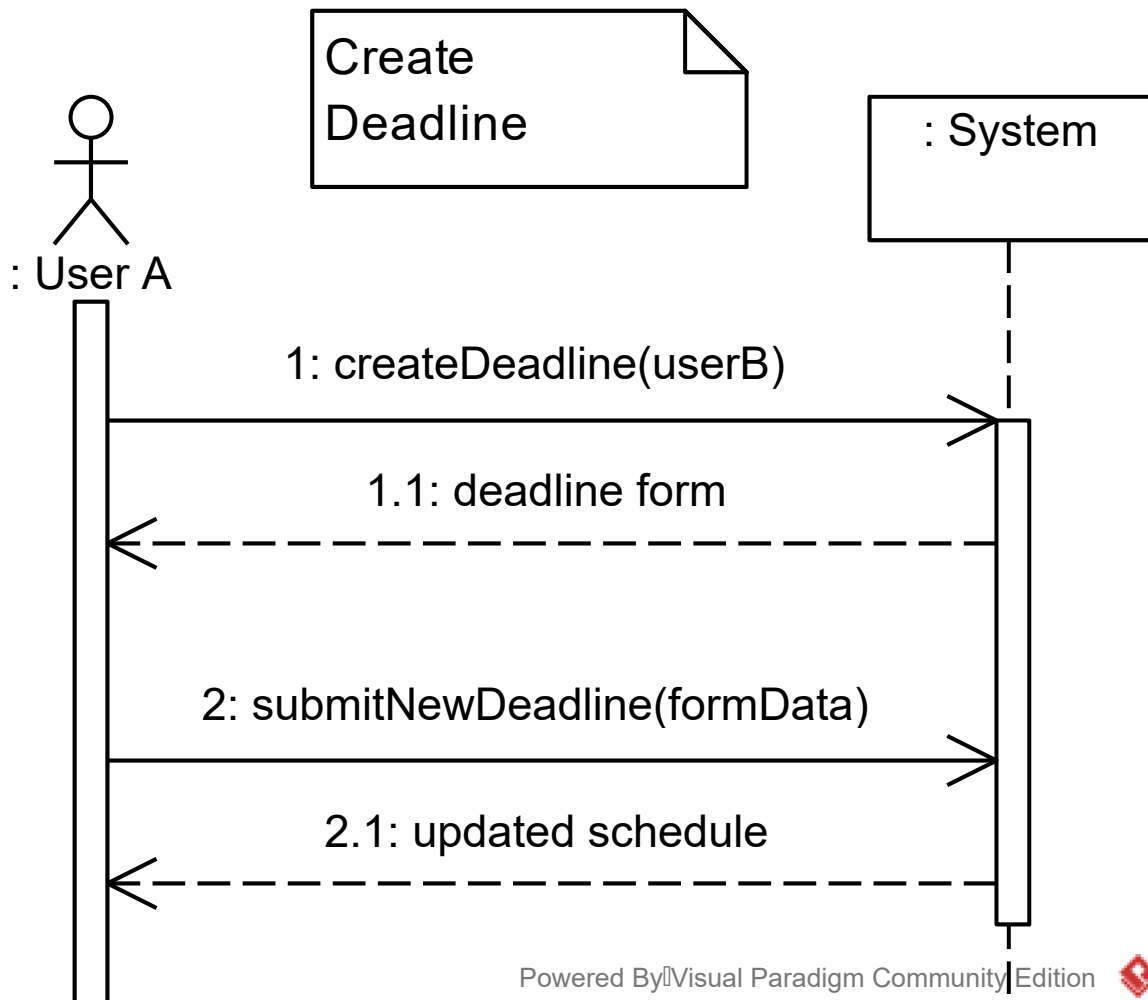


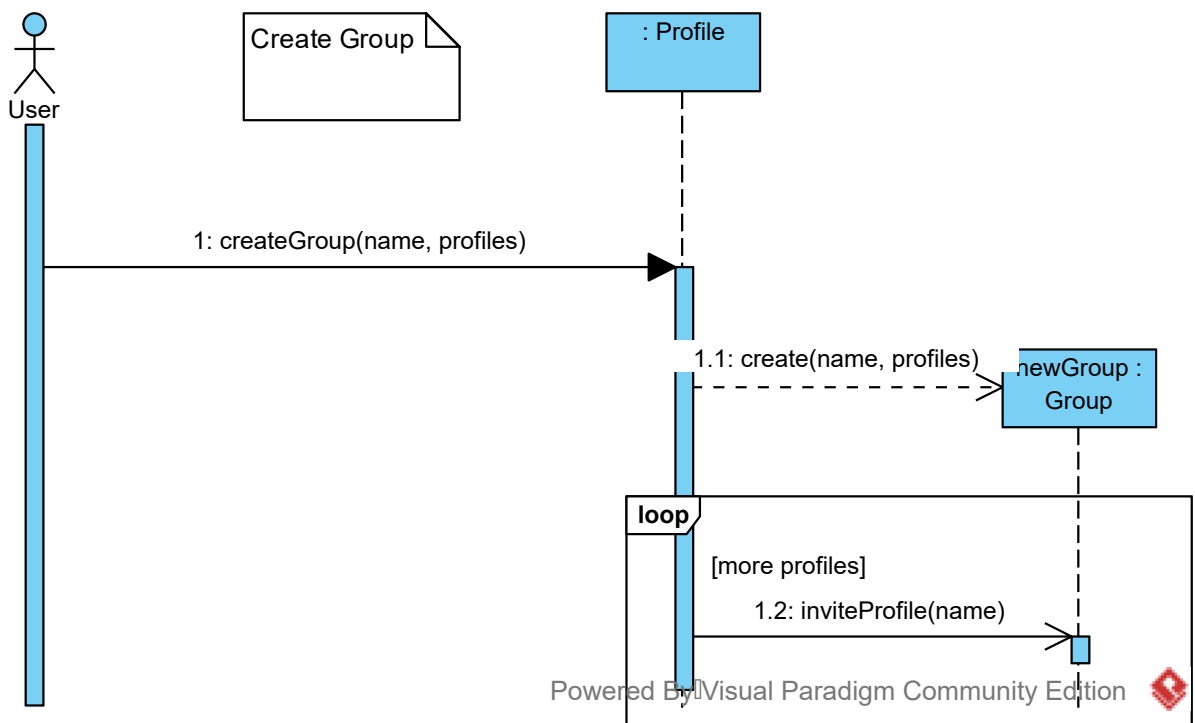


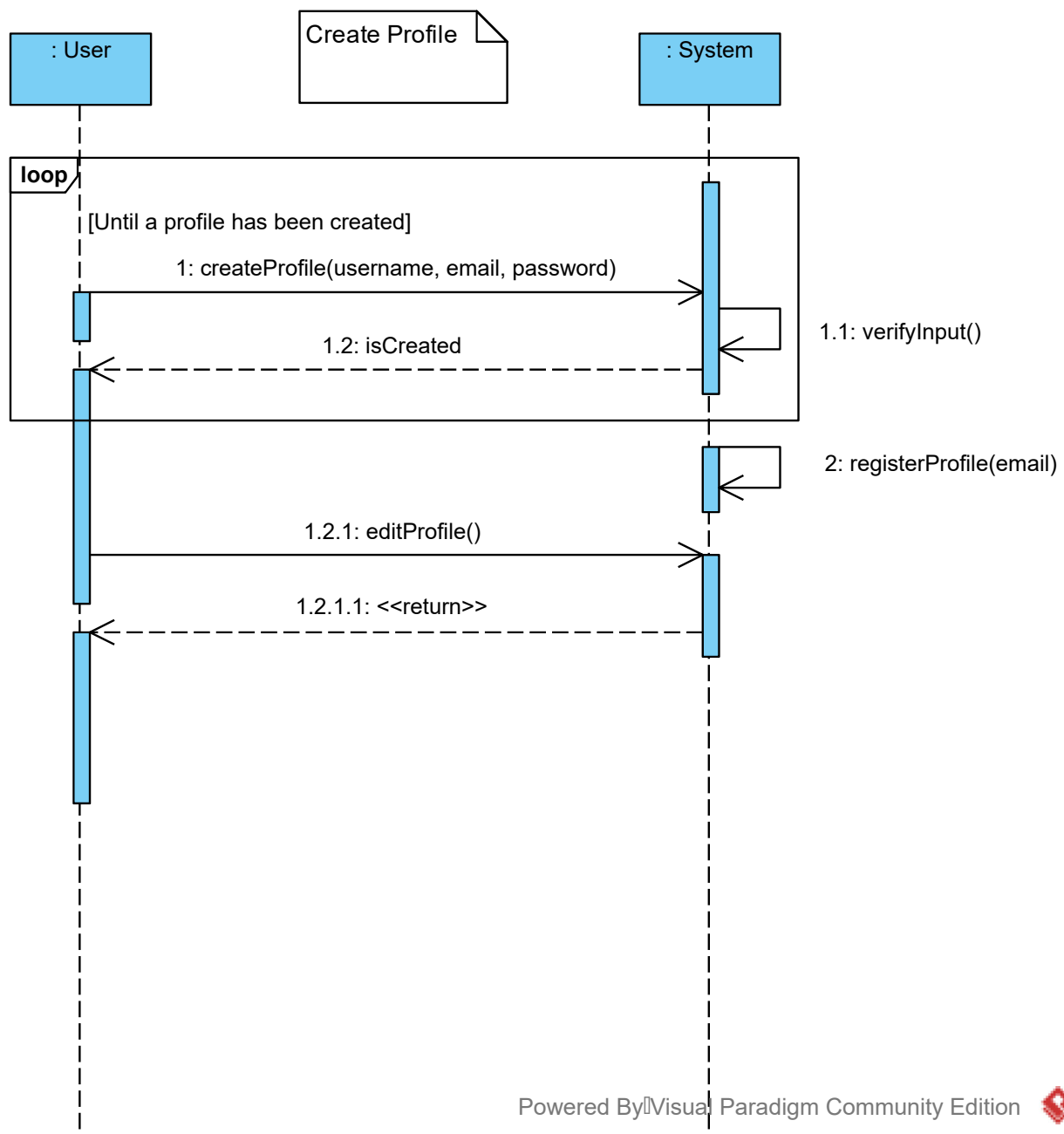


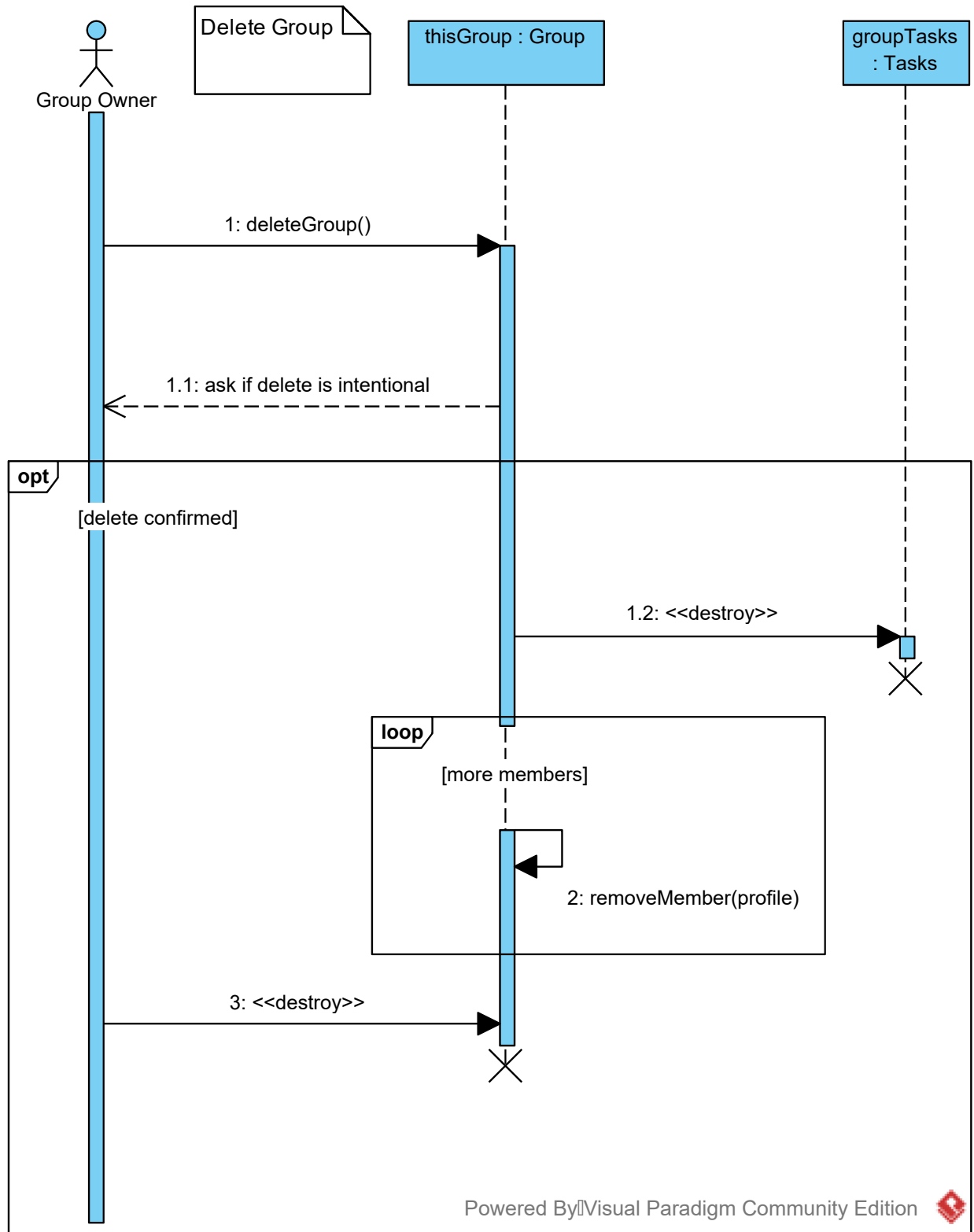
Create Activity



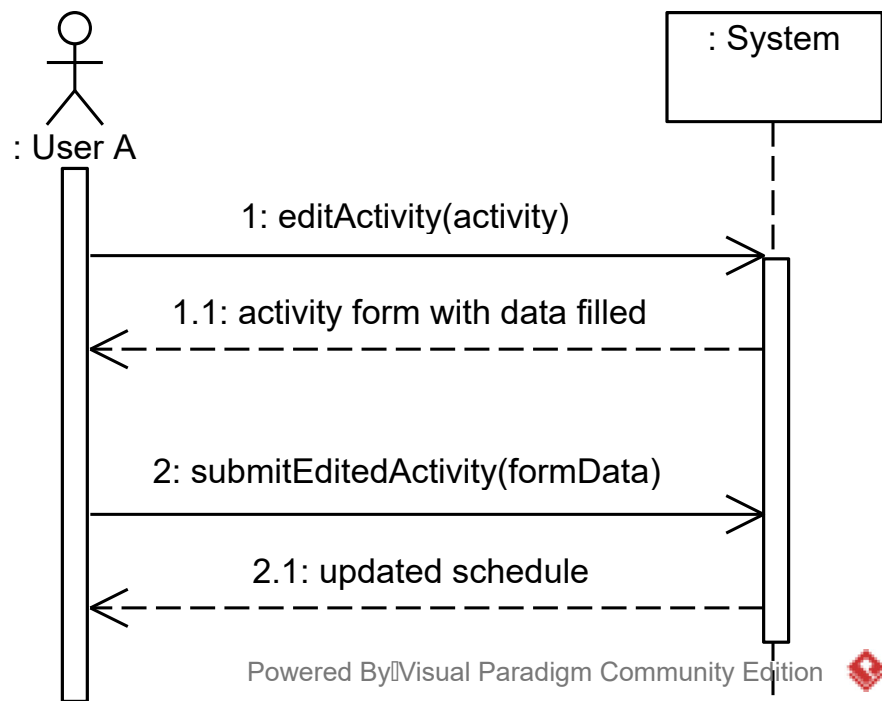




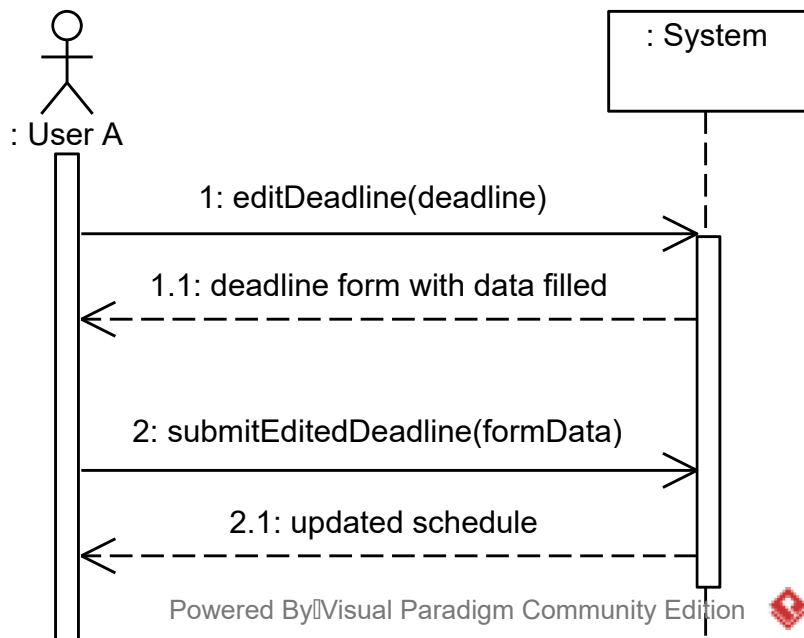


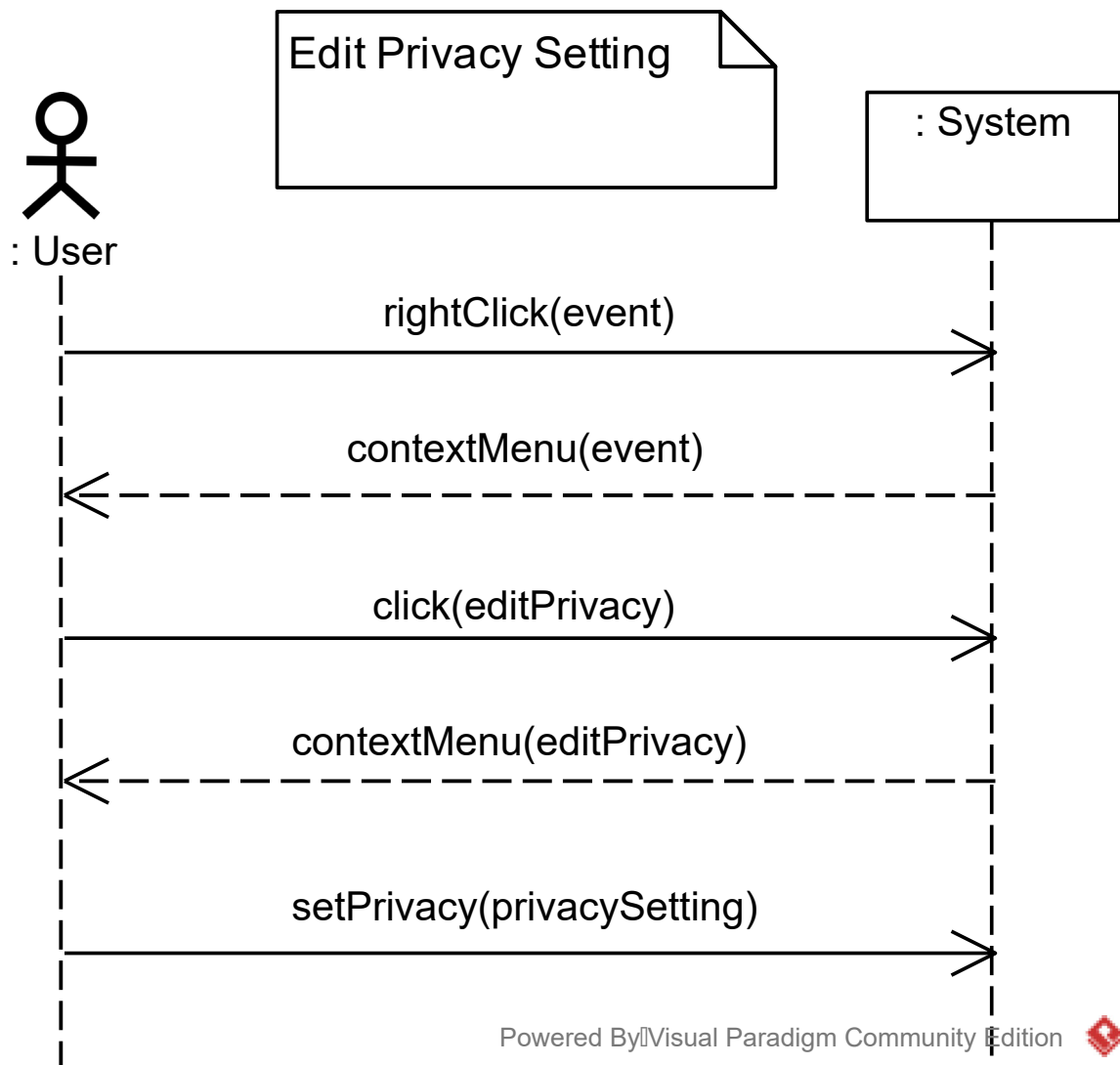


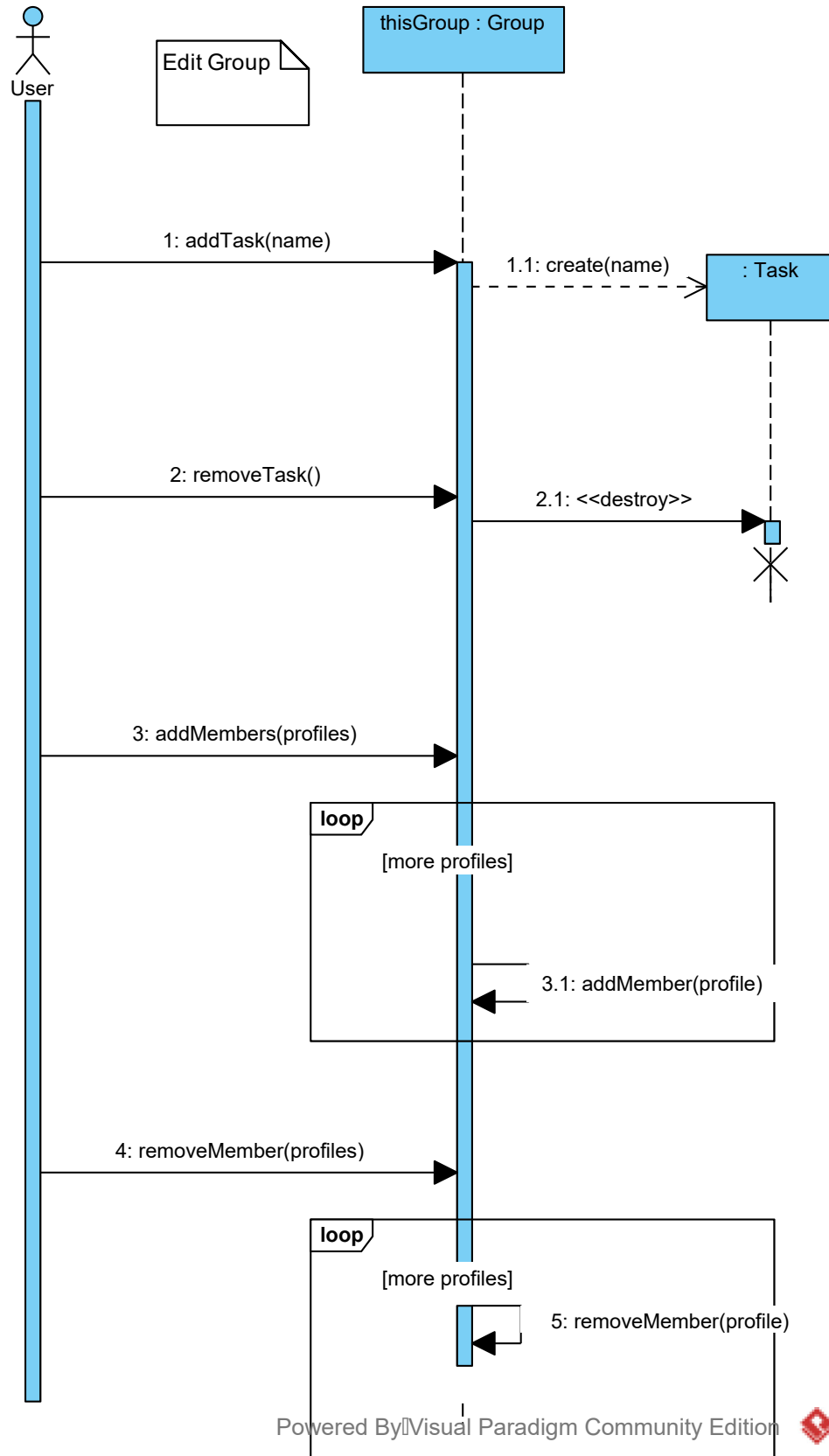
Edit Activity

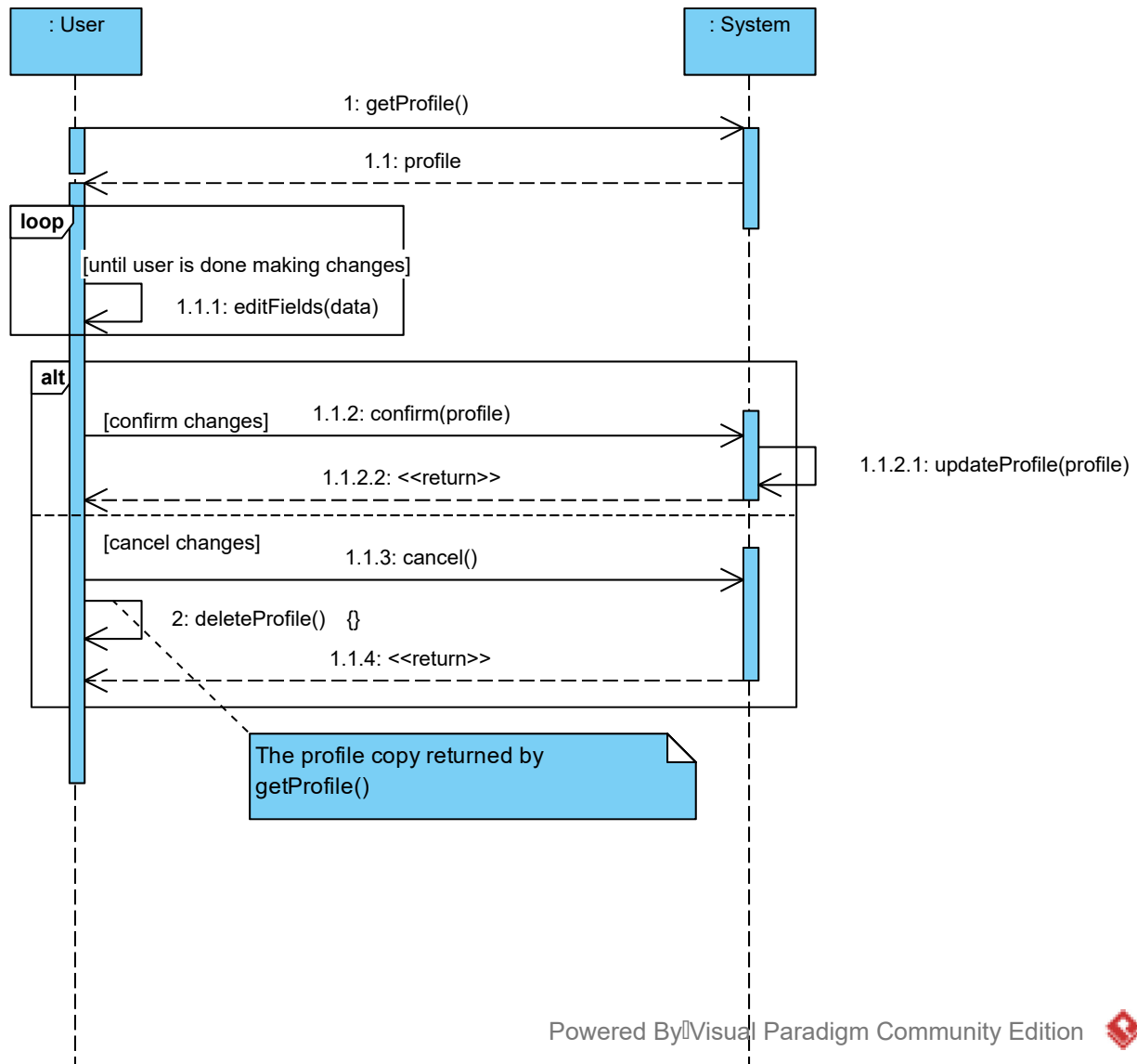


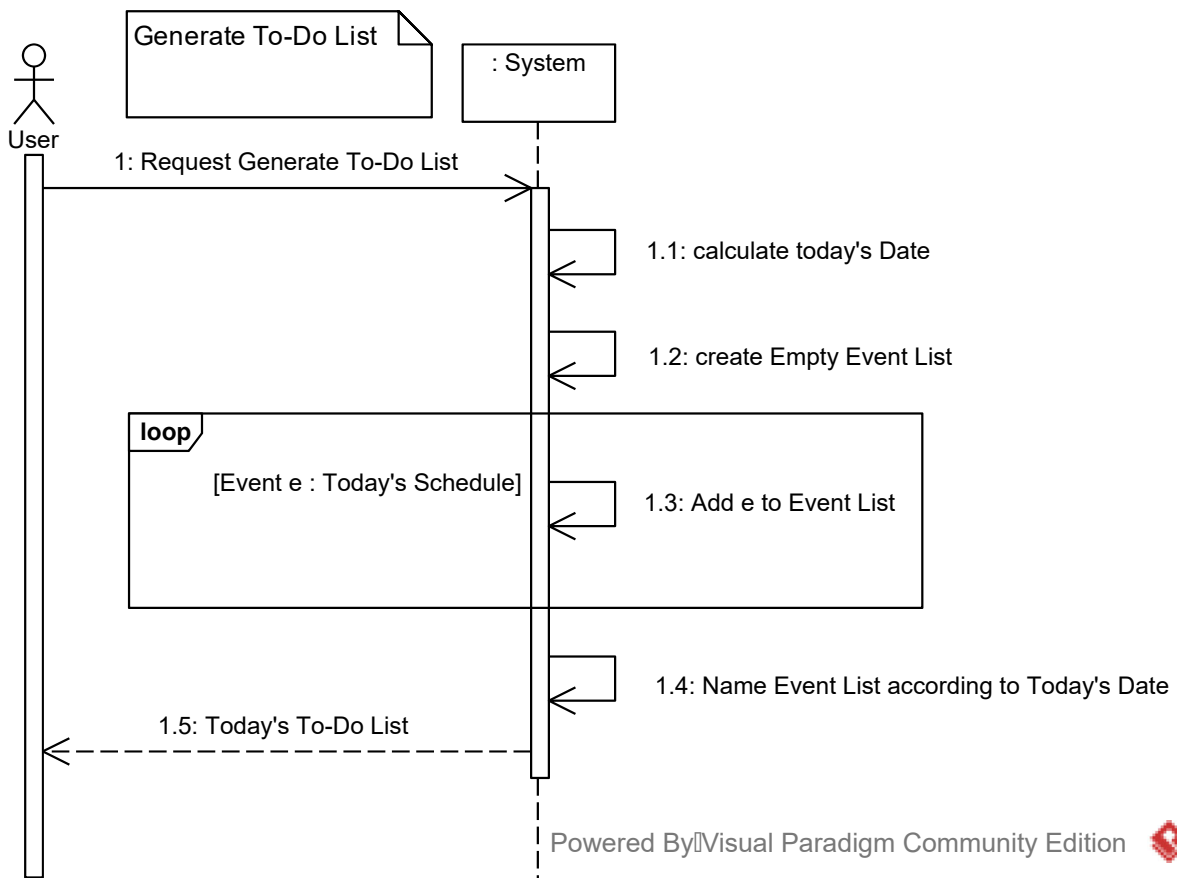
Edit Deadline



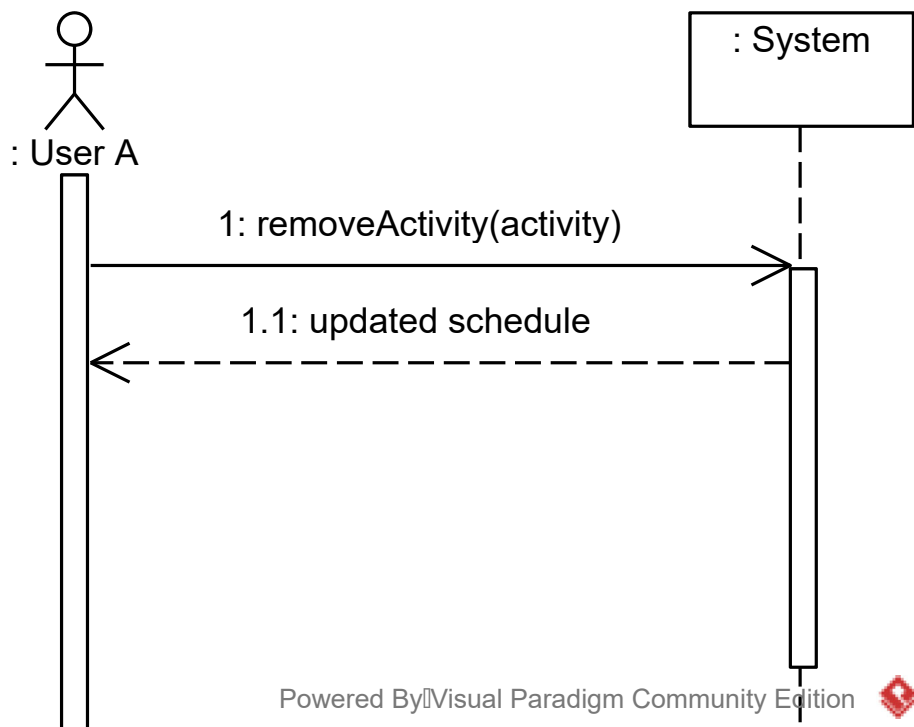




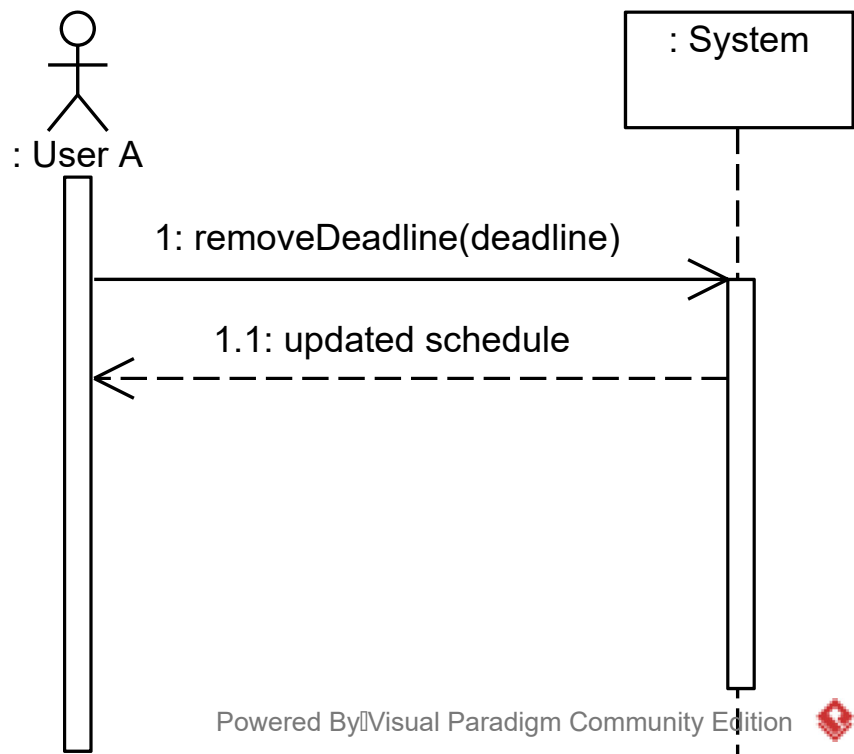


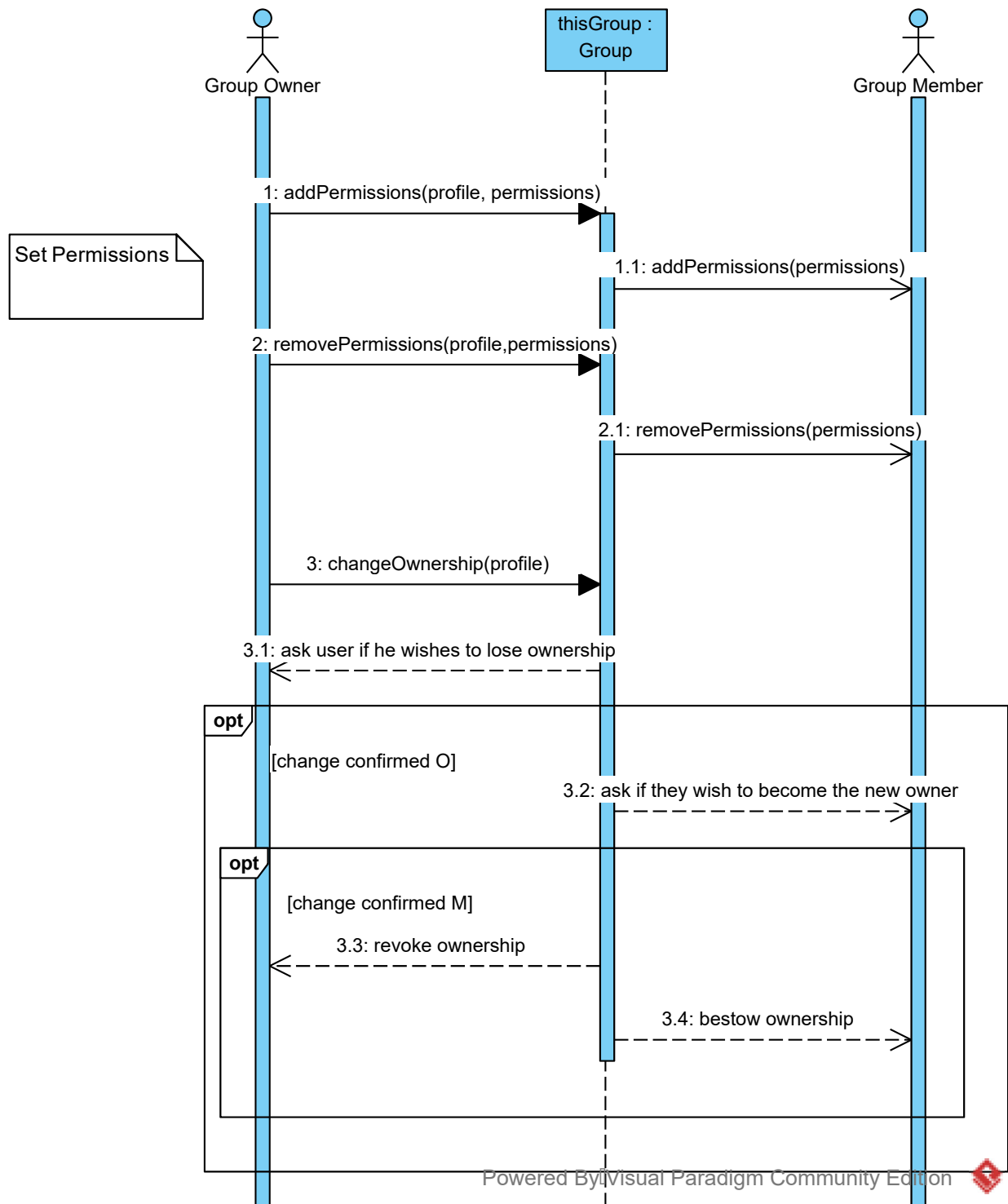


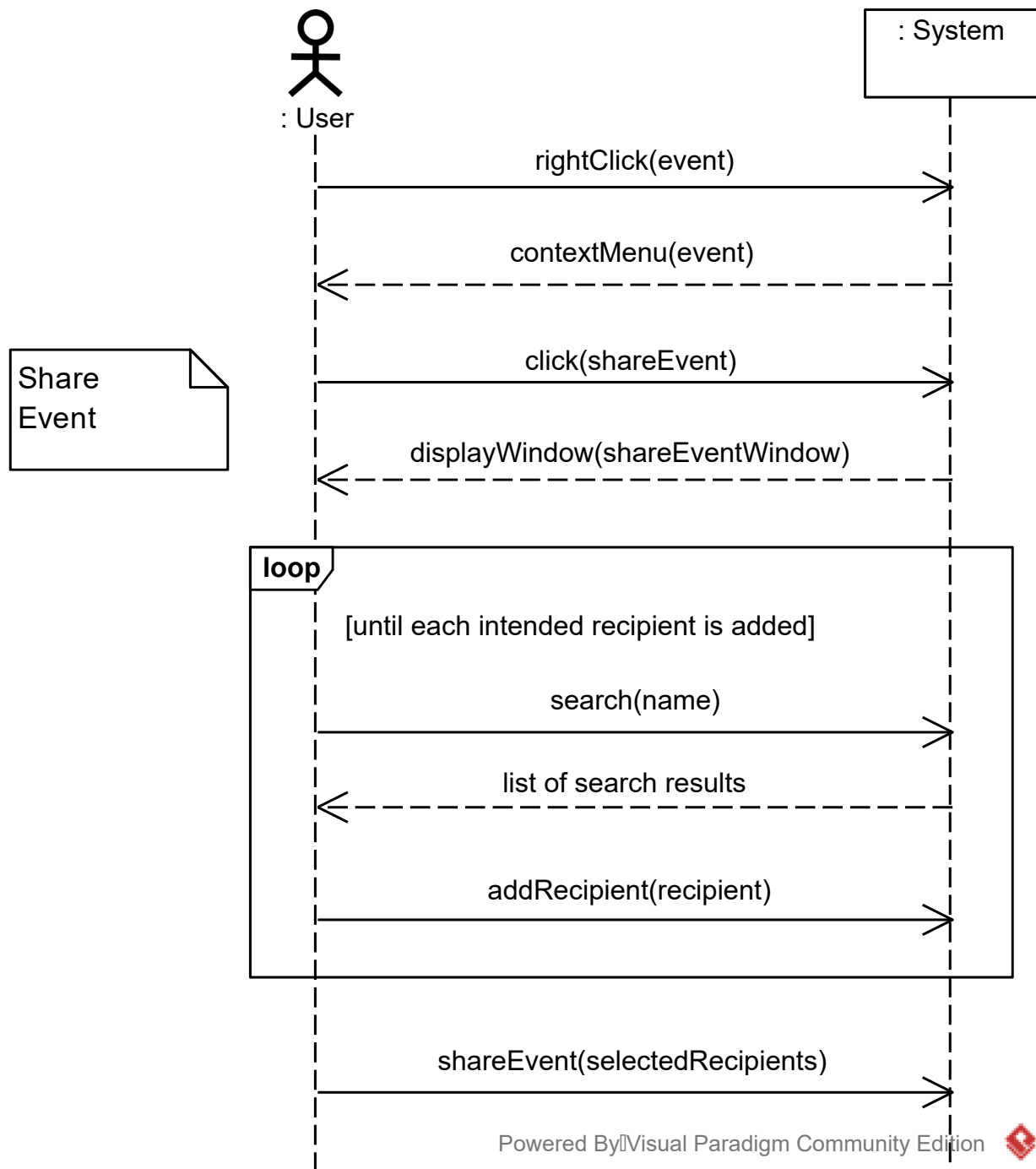
Remove
Activity



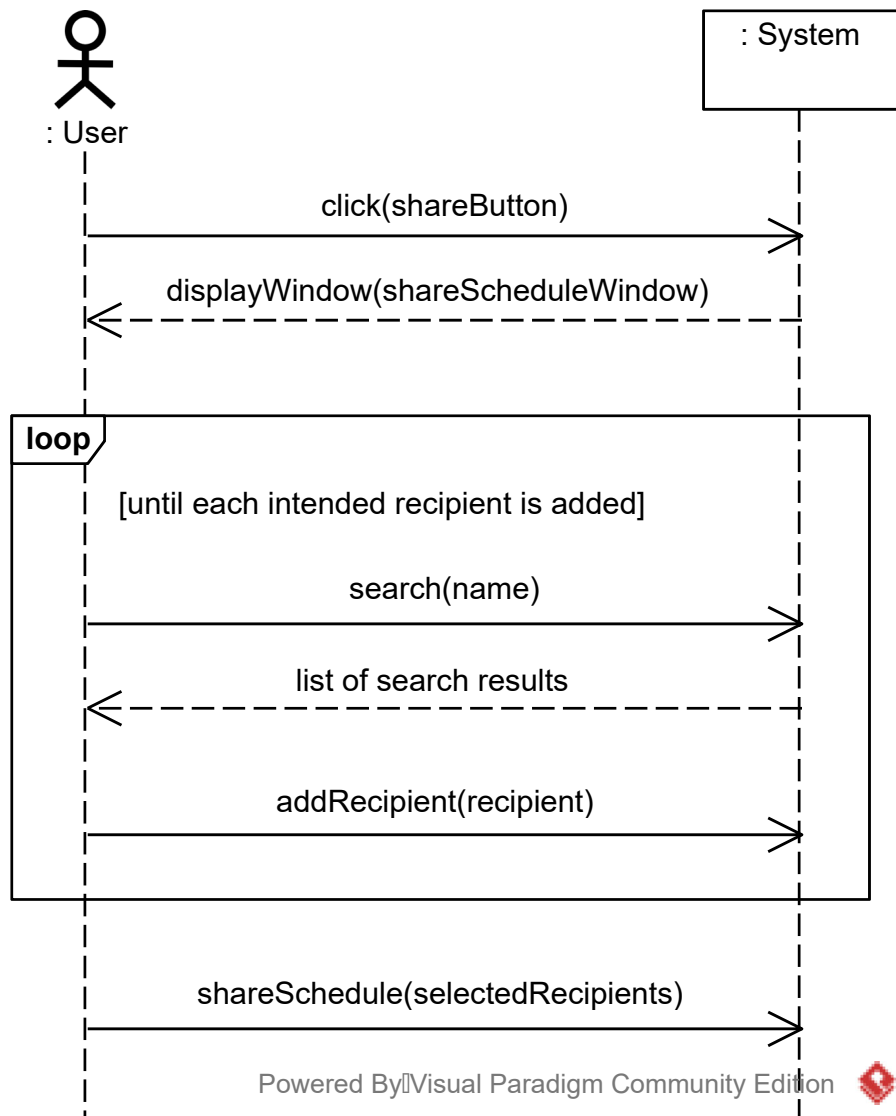
Remove
Deadline

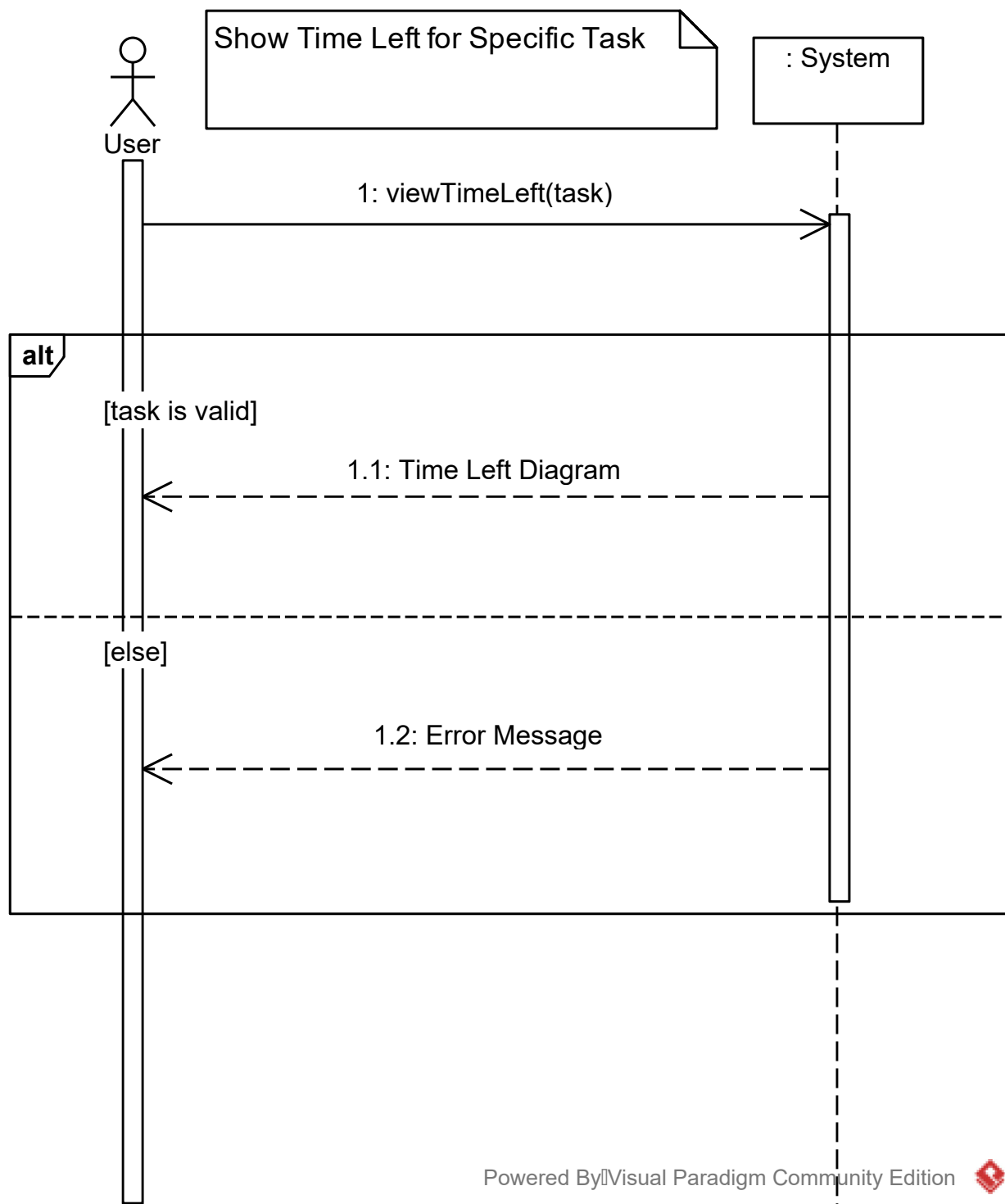


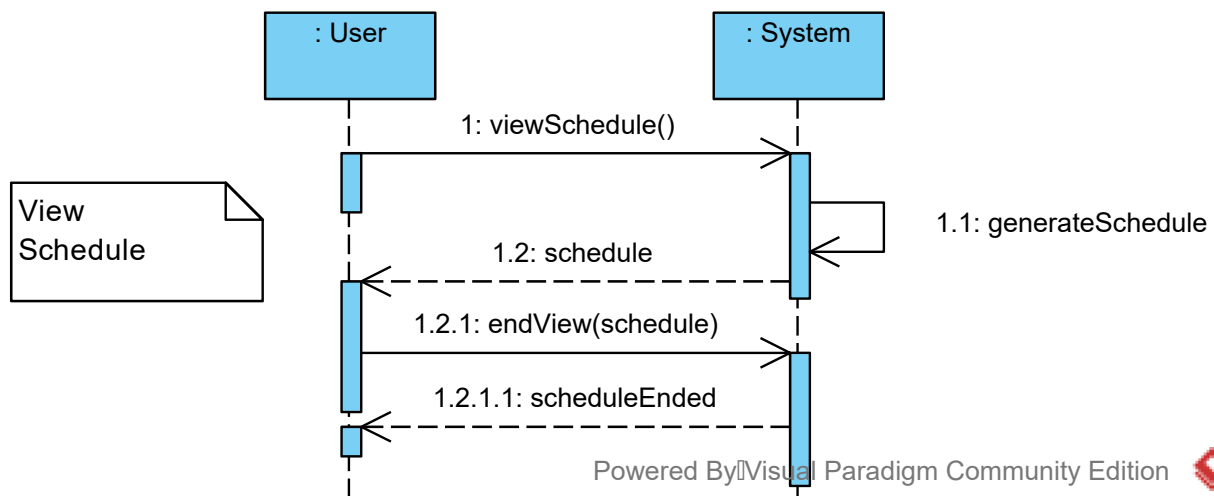


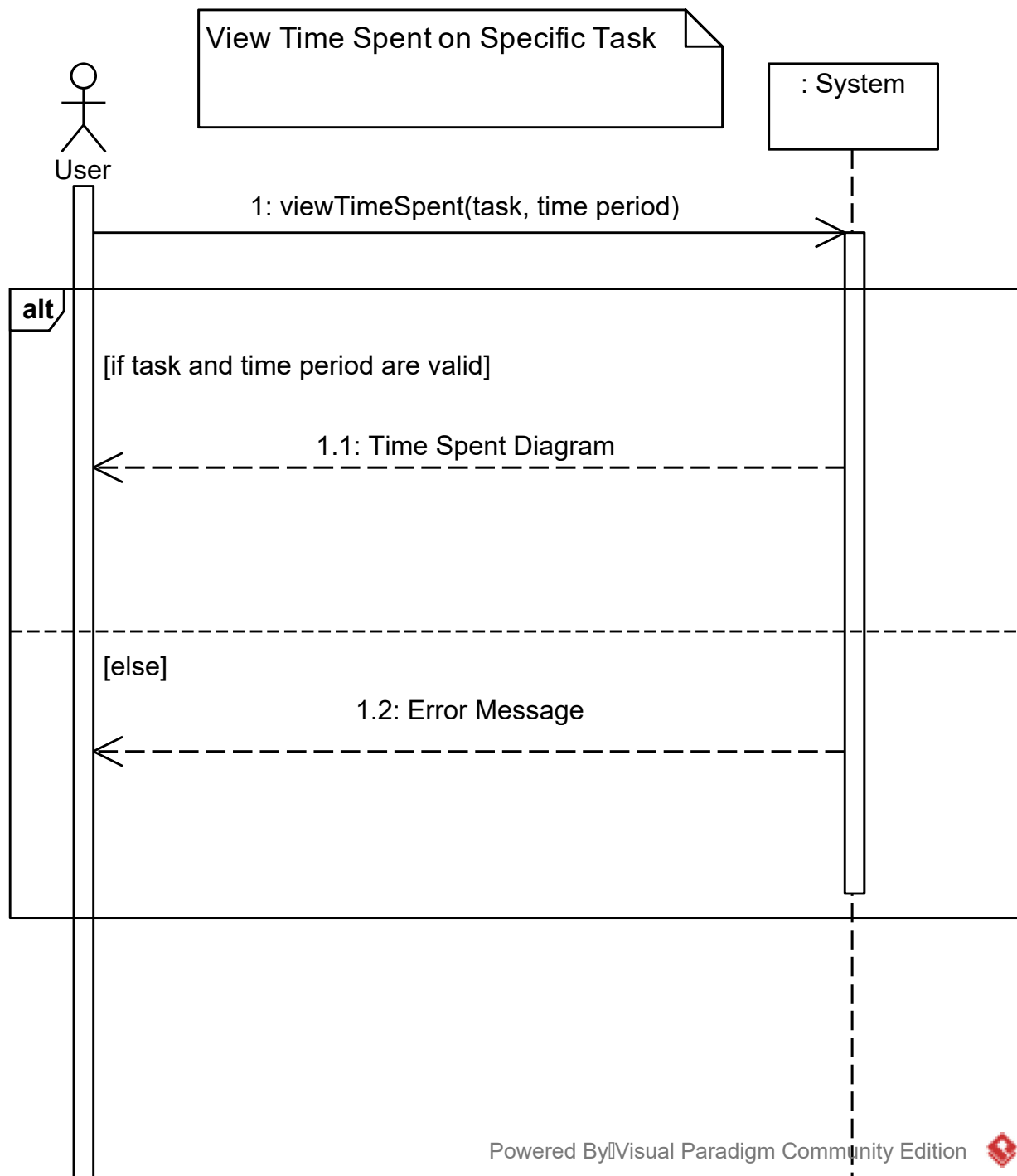


Share
Schedule









<<requirement>> Deadline Creation
Text = "A user should be able to create a deadline on their schedule." ID = "REQ001" source = "" kind = ""

<<requirement>> Deadline Editing
Text = "A user should be able to change the name time, date, etc. of a deadline once it's on their schedule." ID = "REQ002"

<<requirement>> Deadline Removal
Text = "A user should be able to remove a deadline from their schedule." ID = "REQ003"

<<requirement>> Work Time Assignment
Text = "User should be able to specify what times they are available to work." ID = "REQ004" source = ""

<<requirement>> Auto-Append
Text = "The System should be able to find a free space to put a new or existing event." ID = "REQ005" source = ""

<<requirement>> Generating To-Do Lists
Text = "The system should be able to generate and display a to-do list using information from a user's schedule."

<<requirement>> Group Creation
Text = "A user should be able to create groups with other users." ID = "REQ007" source = "" kind = ""

<<requirement>> Group Deletion
Text = "A user should be able to remove groups that they own." ID = "REQ008" source = "" kind = ""

<<requirement>> Group Editing
Text = "A user should be able to edit their groups to change the name or members." ID = "REQ009" source = ""

<<requirement>> Changing Group Permissions
Text = "A user should be able to change the permissions of a group they own." ID = "REQ010" source = ""

<<requirement>> View Schedule
Text = "A user should be able to view their schedule in a convenient manner." ID = "REQ011" source = ""

<<requirement>> Create Profile
Text = "The system should support creating user profiles." ID = "REQ012" source = "" kind = ""

<<requirement>> Edit Profile
Text = "A user should be able to change their email, password, etc." ID = "REQ013" source = "" kind = ""

<<requirement>> View Time Spent on Specific Task
Text = "The system should be able to calculate the time a user has spent on a task using the user's schedule and active work times." ID = "REQ014"

<<requirement>> Show Time Left for Specific Task
Text = "The system should be able to calculate and show the time left on a task using the start date and due date."

<<requirement>> Calculate Free Time
Text = "The system should be able to calculate the amount of free time a user has using user-defined"

<<requirement>> Share Schedule
Text = "A user should be able to share a schedule with groups." ID = "REQ017" source = "" kind = ""

<<requirement>> Share Event
Text = "A user should be able to share individual events with groups." ID = "REQ018" source = ""

<<requirement>> Edit Privacy Settings
Text = "A user should be able to edit the privacy settings for their profile." ID = "REQ019" source = "" kind = ""

<<requirement>> Create Activity
Text = "A user should be able to add an activity to their schedule." ID = "REQ020" source = "" kind = ""

<<requirement>> Edit Activity
Text = "A user should be able to edit the time, name, location, etc. of activities on their schedule." ID = "REQ021"

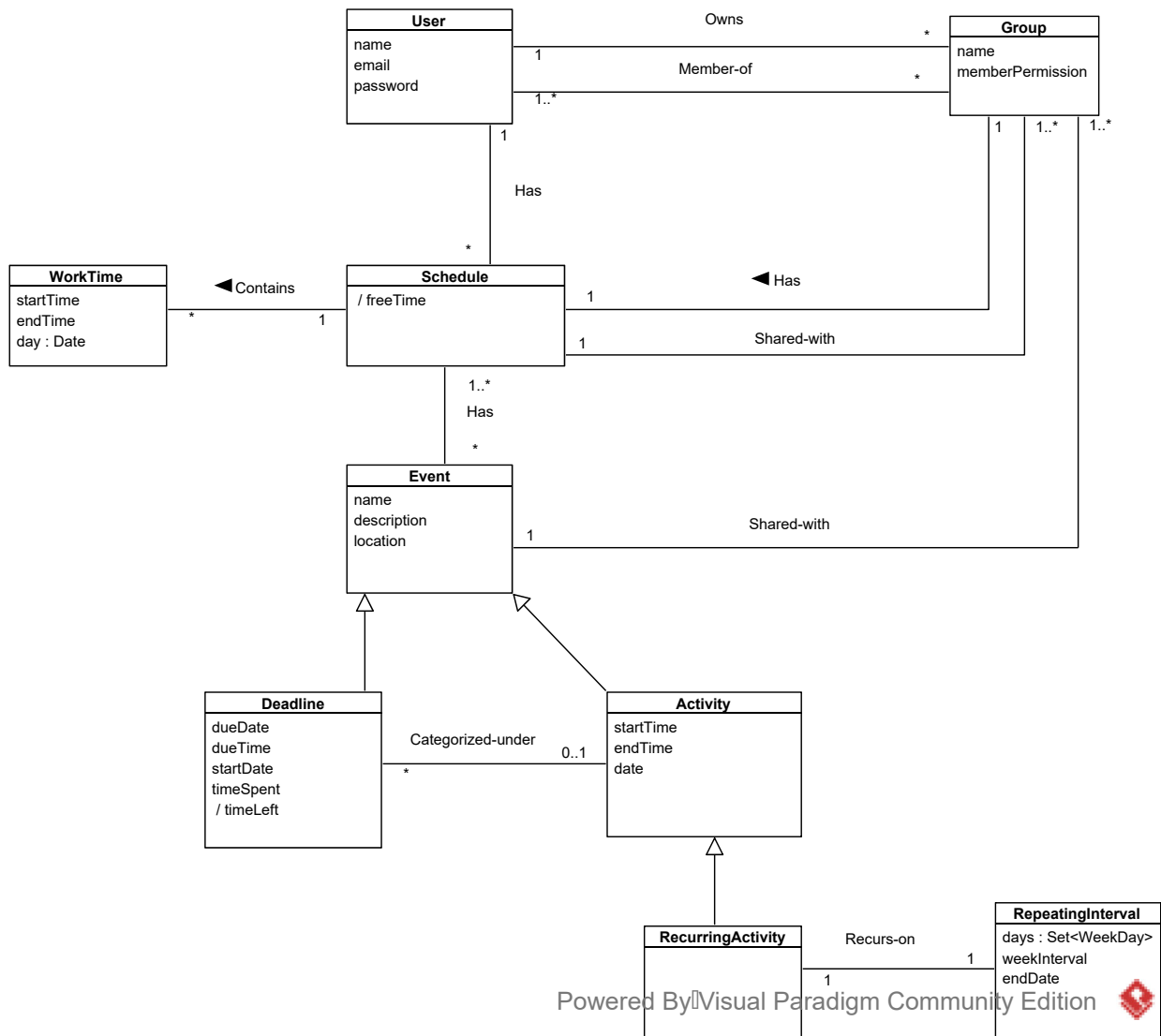
<<requirement>> Remove Activity
Text = "A user should be able to remove an activity from their schedule." ID = "REQ022" source = ""

<<requirement>> Support Private Mode
Text = "User should be able to hide their schedule from others." ID = "REQ023" source = "" kind = "" verifyMethod = ""

<<requirement>> Support View-Only Mode
Text = "A group owner should be able to make their group's schedule viewable by other group members but not editable." ID = "REQ024" source = ""

<<requirement>> Works Offline
Text = "Basic functionality should be available offline." ID = "REQ025" source = "" kind = "" verifyMethod = "" risk = "" status = ""





System

calculateFreeTime(timePeriod)
createDeadline(userB)
submitNewDeadline(formData)
editDeadline(deadline)
submitEditedDeadline(formData)
createActivity(userB)
submitNewActivity(formData)
removeDeadline(deadline)
editActivity(activity)
submitEditedActivity(formData)
removeActivity(activity)
search(name)
shareSchedule(selectedRecipients)
shareEvent(selectedRecipients)
addProfile(profile)
addTask(taskName)
removeTask(task)
removeMember(profile)
setPrivacy(privacySetting)
addPermissions(profile, permissions)
removePermissions(profile, permissions)
changeOwnership(profile)
deleteGroup()
createGroup(name, profiles)
inviteProfile()
assignWorkTime(startTime, endTime)
createProfile(username, email, password)
getProfile()
viewSchedule()
generateSchedule()
endView()



	REQ001	REQ002	REQ003	REQ004	REQ005	REQ006	REQ007	REQ008	REQ009	REQ010	REQ011	REQ012	REQ013	REQ014	REQ015	REQ016	REQ017	REQ018	REQ019	REQ020	REQ021	REQ022	REQ023	REQ024	REQ025
UC01	X																								X
UC02		X																							X
UC03			X																						X
UC04				X																					X
UC05					X																				X
UC06						X																			
UC07							X																		
UC08								X																	
UC09									X																
UC10										X															X
UC11											X														X
UC12												X													X
UC13													X												X
UC14														X											X
UC15															X										X
UC16																X									
UC17																	X								X
UC18																		X							
UC19																			X				X	X	
UC20																				X					X
UC21																					X				X
UC22																						X			X

Username

Password

Login

Sign Up

Profile

My Schedule

Name

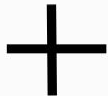
Email:

My Group Schedules

> SE Group

> Algorithm's Group

> Fun Activities Group



Tuesday, 2/11/2020

My Categories

> Software Engineering

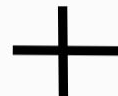
> Intro to Algorithms



Upcoming Deadlines

> SE Project -- in 2 days

> SE Midterm I -- in 4 days



Schedule

To Profile...

2/9/2020 - 2/15/2020

Today's To-Do List

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Calculate Free Time

Share this schedule

Generate To Do List

Assign Work Time

