Project Vision:

We will be making a Planner application that can support event interactions, recognize event proposals, recognize resources, allow booking event times, and disallow two groups from booking to the same resources. The Planner should be able to create/delete events on the Planner, archive events, delete events that have already passed, and edit pre-existing events. We may add or change some features in the future, but this is the baseline we have so far.

Requirements

Functional

1. Events should be able to interact with each other
2. The planner should be able to recognize separate events
3. The planner should prevent multiple events from booking the same time
4. The planner should automatically delete events with no information provided
5. The existing users should have access to the planner
6. The planner should allow the addition of new users
7. The users should be able to delete events they own
8. The users should be able to create new events
9. The users should be able to book times
10. The user should be able to un-archive events
11. The user should be able to add notes onto events
12. The user should be able to complete events
13. The user should be able to archive or delete completed events
14. The user should be able to change the details of their own events
15. The user should be able to delete notes

Non-Functional

1. The planner, and its features, will be available in the english language.
2. Documentation for the planner will be written in english
3. All event dates are posted in the format of the Gregorian calendar
4. The planner assumes Central Standard Time

Use Cases

Use Cases	req 1	req 2	req 3	req 4	req 5	req 6	req 7	req 8	req 9	req 10	req 11	req 12	req 13	req 14	req 15	Owner
UC Add Event		X	X	X				X	X							BK
UC Archive Event		X						X	X	X		X	X			BK
UC Edit Event		X	X	X	X		X	X	X	X		X	X	X		BK
UC Delete Event		X		X	X		X		X			X	X		X	YD
UC Resolve Event Conflicts	X	X	X					X		X						JW
UC Event Notes		X									X					JW
UC Delete Notes	X	X														YD
UC Event Notification		X									X					BC
UC Toggle Notes		X									X					YD
UC Toggle Notifications																JW
UC Create New User						X										BC
UC List Events	X	X						X								YX
UC Login																YX
UC Complete Event	X	X	X			X		X	X							YX
UC Set starting location and time	X	X	X			X		X	X							BC

ID UC Add Event

Created By B K

Scope Travel planner

Stakeholders and interests

User

- The person/persons who will be traveling on the trip who wants transport and accommodations

Transportation Businesses

- Airlines, Taxi services etc. That will be relocating the traveler from one place to another at a specific time

Accommodations Businesses

- Restaurants, Hotels, entertainment etc. That will be providing accommodation for the traveler at a specific location but will not be relocating the traveler

Precondition

The traveler wants to attend an event.

Postcondition

The event is added to the traveler's planner.

Main success scenario

1. Traveler wants to add an event
2. Traveler inputs start and end locations and times
3. System checks for conflicts
4. Traveler confirms event

Extensions

a.* anytime the traveler wishes to stop adding event

1. No event is added

3.a there is a conflict in time or locations

a. the traveler edits the current event

1. The event is edited

2. Go back to step 3

b. the traveler edits the conflicting event

1. The conflicting event is edited

2. The conflicting event is checked for conflicts created by edit

3. Go back to step 3

c. the traveler allows the conflicts to exist unchanged

1. Conflicts are added to a list of conflicts and are clearly displayed so that the traveler can change them later.

2. Proceed to confirmation in step 4

4.a The traveler does not confirm the event

1. The event is not saved

ID UC Archive Event

Created By B K

Scope Travel Planner

Stakeholders and interests

User

- The person/persons who will be traveling on the trip who wants to save an event for later use

Precondition

There is an event that the traveler wants to archive

Postcondition

The event is stored for later use in the archive.

Main success scenario

1. The traveler selects an event to archive
2. The traveler confirms that they want to archive the event
3. The event is archived

Extensions

a.* At any time the traveler no longer wishes to archive the event

    1. No event is archived

2.a The traveler does not confirm that they want to archive the event

    1. No event is archived

ID UC Edit Event

Created By B K

Scope Travel Planner

Stakeholders and interests

User

- The person/persons who will be traveling on the trip who wants to change their transport and accommodations

Transportation Businesses

- Airlines, Taxi services etc. That will be relocating the traveler from one place to another at a specific time

Accommodations Businesses

- Restaurants, Hotels, entertainment etc. That will be providing accommodation for the traveler at a specific location but will not be relocating the traveler

Precondition

There is an event that the traveler wishes to edit

Postcondition

The event is edited

Main success scenario

1. The traveler selects an event to edit
2. The traveler edits the event
3. The event is checked for conflicts in location and space
4. The traveler confirms the edit

Extensions

a.* at any time the traveler wishes to not edit

1. No event is saved

3.a The event conflicts with another event

a. the traveler edits the current event

1. The event is edited

2. Go back to step 3

b. the traveler edits the conflicting event

1. The conflicting event is edited

2. The conflicting event is checked for conflicts created by edit

3. Go back to step 3

c. the traveler allows the conflicts to exist unchanged

1. Conflicts are added to a list of conflicts and are clearly displayed so that the traveler can change them later.

2. Proceed to confirmation in step 4

4.a The traveler does not confirm the edit

1. No event is saved

ID UC Delete Event

Created By Y D

Scope Travel Planner

Stakeholders and interests

User

- The person/persons who will be traveling on the trip who wants to cancel their transport and accommodations

Transportation Businesses

- Airlines, Taxi services etc. That will be relocating the traveler from one place to another at a specific time

Accommodations Businesses

- Restaurants, Hotels, entertainment etc. That will be providing accommodation for the traveler at a specific location but will not be relocating the traveler

Precondition

An event exists that the traveler wants to delete

Postcondition

The event is removed

Main Success Scenario

1. The traveler selects an event they want to delete
2. The system checks if removing the event will cause any conflicts
3. The traveler confirms that they want to remove the event
4. The event is removed

Extensions

a.* At any time the traveler decides not to remove the event

    1. The event is not removed

2.a There is a conflict with removing the event

    a. The traveler adds a filler event

        1. The traveler begins to add a filler event with UC Add Event

    b. The traveler edits a neighboring event to get to the next event

        1. The traveler begins the UC Edit Event

ID UC Resolve Event Conflicts

Created By J W

Scope Travel planner

Stakeholders and interests

User

- The person/persons who will be traveling on the trip who wants transport and accommodations

Transportation Businesses

- Airlines, Taxi services etc. That will be relocating the traveler from one place to another at a specific time

Accommodations Businesses

- Restaurants, Hotels, entertainment etc. That will be providing accommodation for the traveler at a specific location but will not be relocating the traveler

Precondition

System found conflict with event being added to the planner.

Postcondition

The conflict no longer exists.

Main success scenario

1. System finds a conflict
2. User chooses to edit the program
3. Event is added to the list
4. Conflict is resolved

Extensions

2a. User closes prompt

1. Deletes Event

2. Returns to Planner Screen

2b. User Edits Event with valid input

1. Adds event to planner

2. Returns to planner

2c. User Edits Event w/ invalid input

1. Deletes event

2. Returns to Planner

ID UC Event Notes

Created by J W

Scope Travel Planner

User

- The person/persons who will be traveling on the trip who wants to add a note to an event

Precondition

An event exists that the traveler wants to add a note to.

Postcondition

A note is added to the event

Main Success Scenario

1. The traveler selects an event that they want to add a note to
2. The traveler inputs a note
3. The traveler confirms that they want to add a note
4. The note is added to the event

Extensions

a.* At any time the traveler no longer wishes to add the note

    1. No note is added to the event

3.a The traveler does not confirm note addition

    1. No note is added to the event

ID UC Delete Notes

Created by Y D

Scope Travel Planner

User

- The person/persons who will be traveling on the trip who wants to add a note to an event

Precondition

A note exists that the traveler wants to delete

Postcondition

A note was deleted

Main success Scenario

1. The traveler selects an event that they want to remove a note from
2. The traveler hits delete
3. The traveler confirms that they want to delete a note
4. The note is deleted

Extensions

a.* At any time the traveler no longer wishes to delete the note

    1. No note is deleted

ID UC Event Notification

Created by B C

Scope Travel Planner

User

- The person/persons who will be traveling on the trip, would like to be reminded/notified about an upcoming event

Precondition

An event that has already been created

Postcondition

A reminder/notification is sent to the user

Main Succes Scenario

1. An event is about to happen
2. A notification about an upcoming event is sent in advance

Extensions

1.a

ID UC Toggle Notes

Created by Y D

Scope Travel Planner

User

- The person/persons who will be traveling on the trip who wants to toggle the notes on or off

Precondition

The traveler is viewing the events and wants to change whether they can see the notes

Postcondition

The visibility of notes is changed

Main Success Scenario

1. The traveler toggles the notifications

Extensions

1.a the notes are visible

    1. The notes are not visible

1.b the notes are not visible

    1. The notes are visible

ID UC Toggle Notifications

Created by J W

Scope Travel Planner

User

- The person/persons who will be traveling on the trip who wants to toggle the notifications on or off

Precondition

The traveler wishes the state of the notifications to change

Postcondition

The notification state is changed

Main Success Scenario

2. The traveler toggles the notifications

Extensions

1.a the notifications are off

    1. The notifications are now on

1.b the notifications are on

    1. The notifications are now off

ID UC Create New User

Created by B C

Scope Travel Planner

User

- The person/persons who will be traveling would like to create a new profile that tracks their future trips

Precondition

Postcondition

A new user is created

Main Succes Scenario

1. User selects “create new user”
2. User fills in information
3. A new user is created

Extensions

1a. User can select cancel if they do not want to create new user

ID UC List Events

Created by Y X

Scope Travel Planner

User

- The person/persons who will be traveling on the trip who wants to see the events

Precondition

There are events that the traveler wants to see

Postcondition

The traveler can view the events

Main Success Scenario

1. The traveler queries the events
2. The events are displayed

Extensions

1.a The traveler wants to see Archived events

1. The traveler is shown archived events

ID UC Login

Created by Y X

Scope Travel Planner

User

- The person/persons who will be logging into the application to access all the user's travel plans

Precondition

The user wants to login

Postcondition

The user is logged in to the application.

Main Success Scenario

1. The user will input their username and password.
2. The system will check that both the username exists, and the password is correct.
3. The user will be allowed to access their plans.

Extensions

2a. The username does not exist within the system

1. Call Creating User

2. Automatically sign in after creation.

2b. The password is not correct

1. Prompt the user to enter the password again.

ID UC Complete Event

Created by Y X

Scope Travel Planner

User

- The person/persons who will be traveling on the trip who wants to mark an event as completed

Precondition

The traveler has input an event that they have subsequently completed

Postcondition

The event is saved as a completed event

Main Success Scenario

4. An event is selected
5. The traveler confirms completion of the event
6. The event is stored as completed

Extensions

1.a The event's timebox has run out

1. The event is selected

1.b The traveler has reached the desired destination

1. The user is prompted to enter a new end time.

2. The event is selected

1.c The traveler selects an event to complete early

1. The traveler is prompted to add a new end time

2. The event is selected

2.a The traveler does not confirm completion of the event

1. The event is stored as a failed event instead of a completed event

ID UC Set starting location and time

Created by B C

Scope Travel Planner

User

- The person/persons who will be traveling on the trip, would like to note the where and when they started.

Precondition

An event is being created.

Postcondition

The time and place for an event has been set.

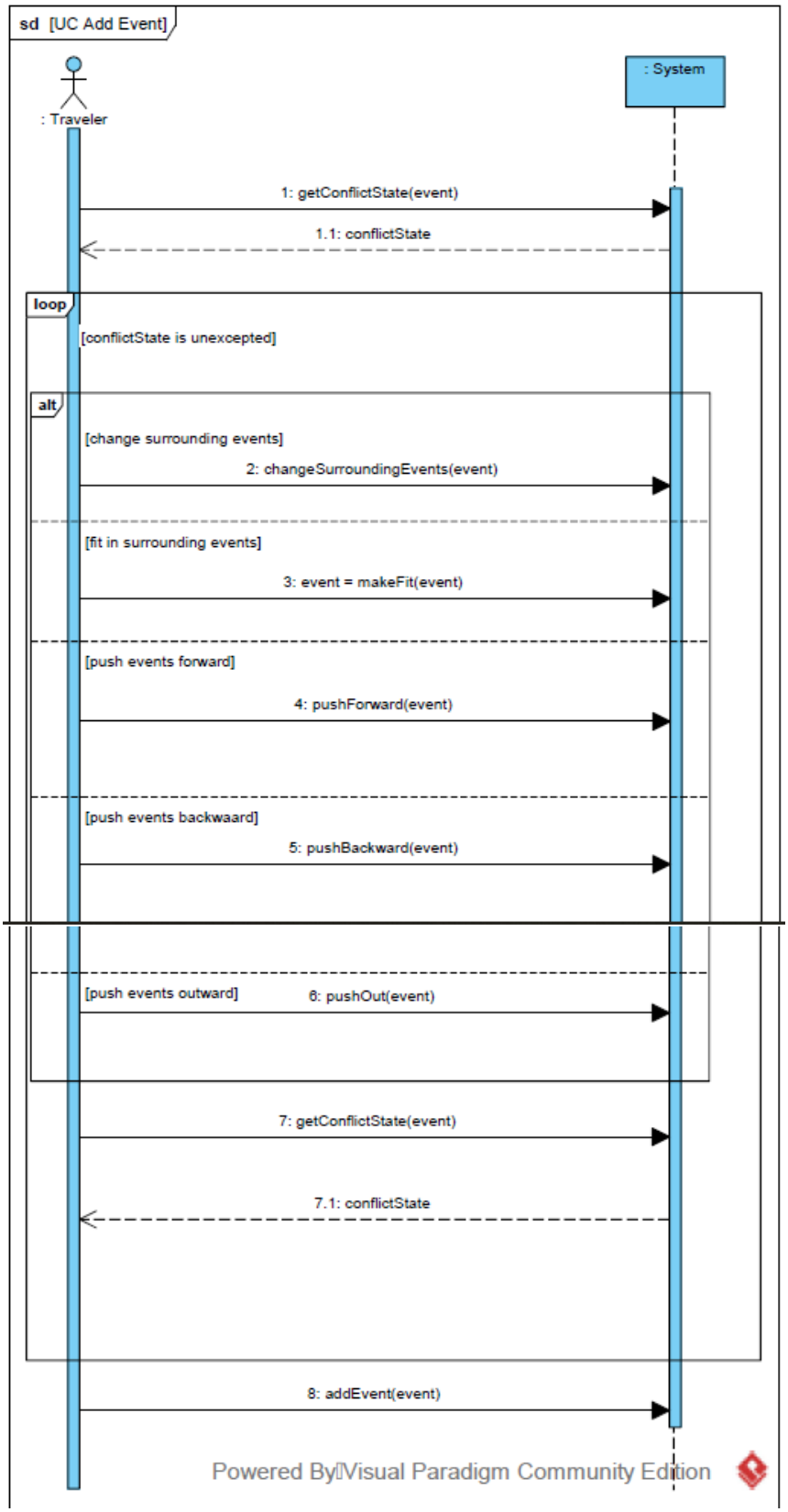
Main Succes Scenario

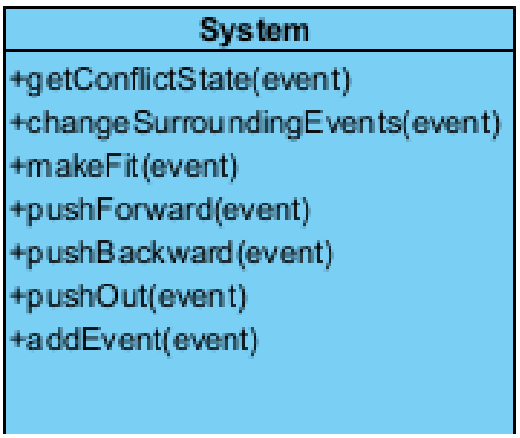
1. An event is being created
2. The event's time and place are set

Extensions

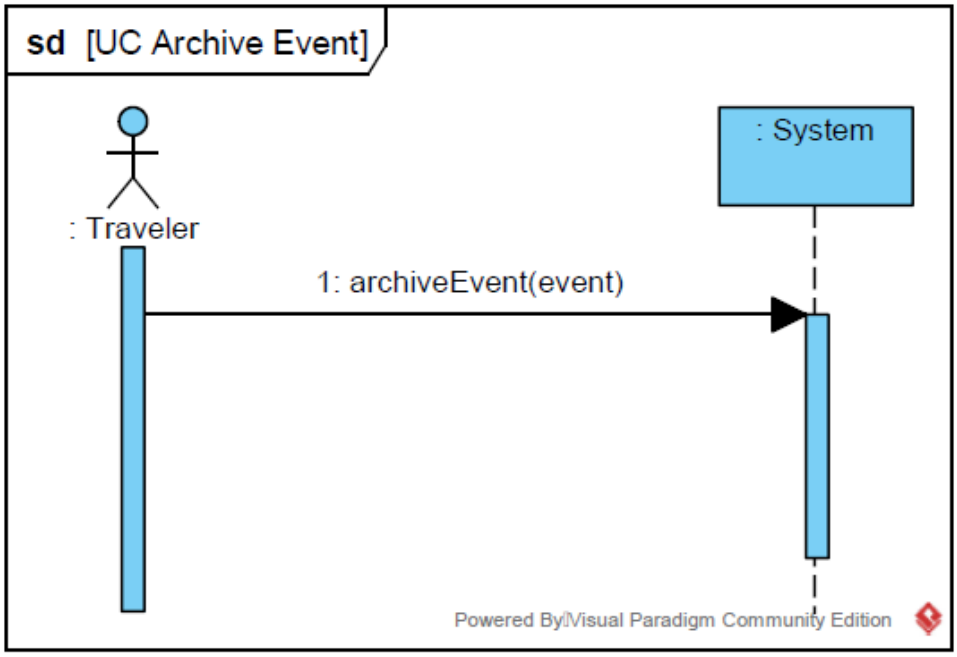
1a. The event's time and place has not yet been set

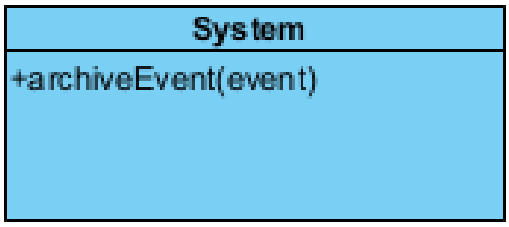
SSD and Operation Contracts



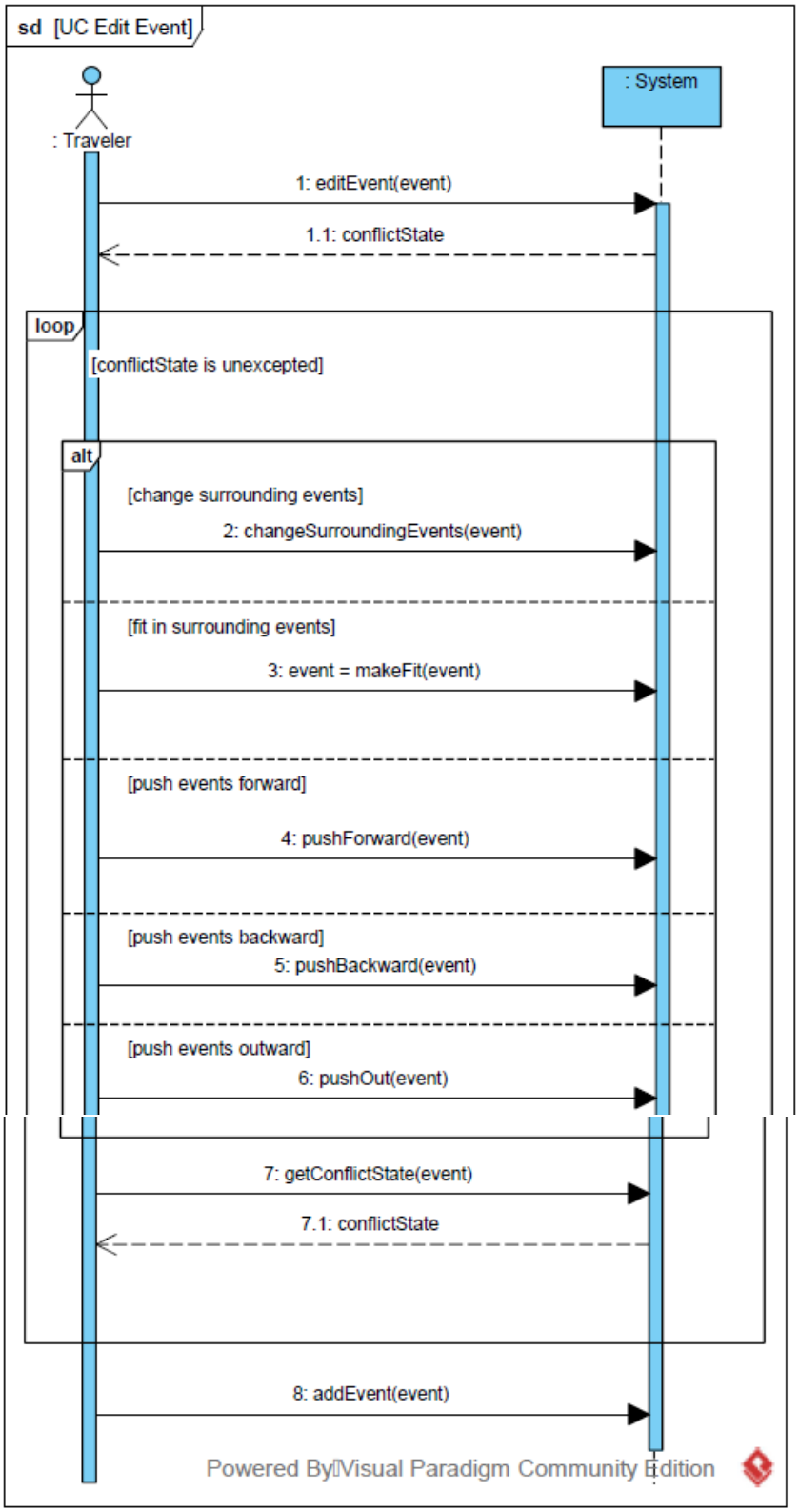


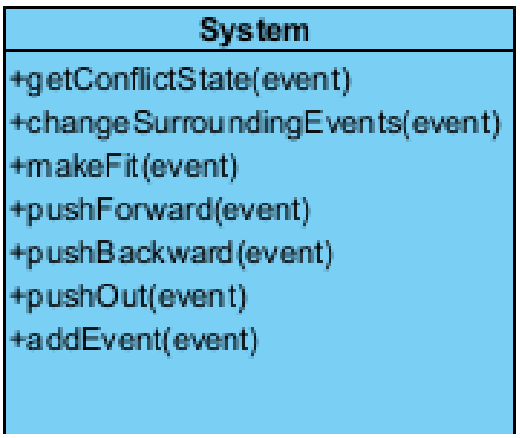
Contract: Add Event (event)
Operation: Adds the event Event Cross Reference: Use Cases: Add Event Pre-conditions: Prompts to add a new event. Post-condition: A new event is created
Contract: push out (event)
Operation: removes the event Cross Reference: Use Cases: Add Event Pre-conditions: an event is being removed. Post-condition: the event is removed
Contract: push backward (event)
Operation: pushes event further pack along the planner Cross Reference: Use Cases: Add Event Pre-conditions: An Event being inserted in front of event . Post-condition: event is pushed back
Contract: push forward(event)
Operation: brings event forward in the planner Cross Reference: Use Cases: Add Event Pre-conditions: An Event being inserted behind event. Post-condition: event is brought forward
Contract: makeFit(event)
Operation: makes the event fit into the planner Cross Reference: Use Cases: Add Event Pre-conditions: inserts event at a certain position. Post-condition: A new event is inserted
Contract: change surroundings events (event)
Operation: changes the Events around event Cross Reference: Use Cases: Add Event Pre-conditions: an event is going to be added. Post-condition: planner is prepped for new event
Contract: get Conflicted State (event)
Operation: checks the date of the new event with the dates of events on the planner Cross Reference: Use Cases: Add Event Pre-conditions: no time conflict Post-condition: event times are checked with the new event for conflicts





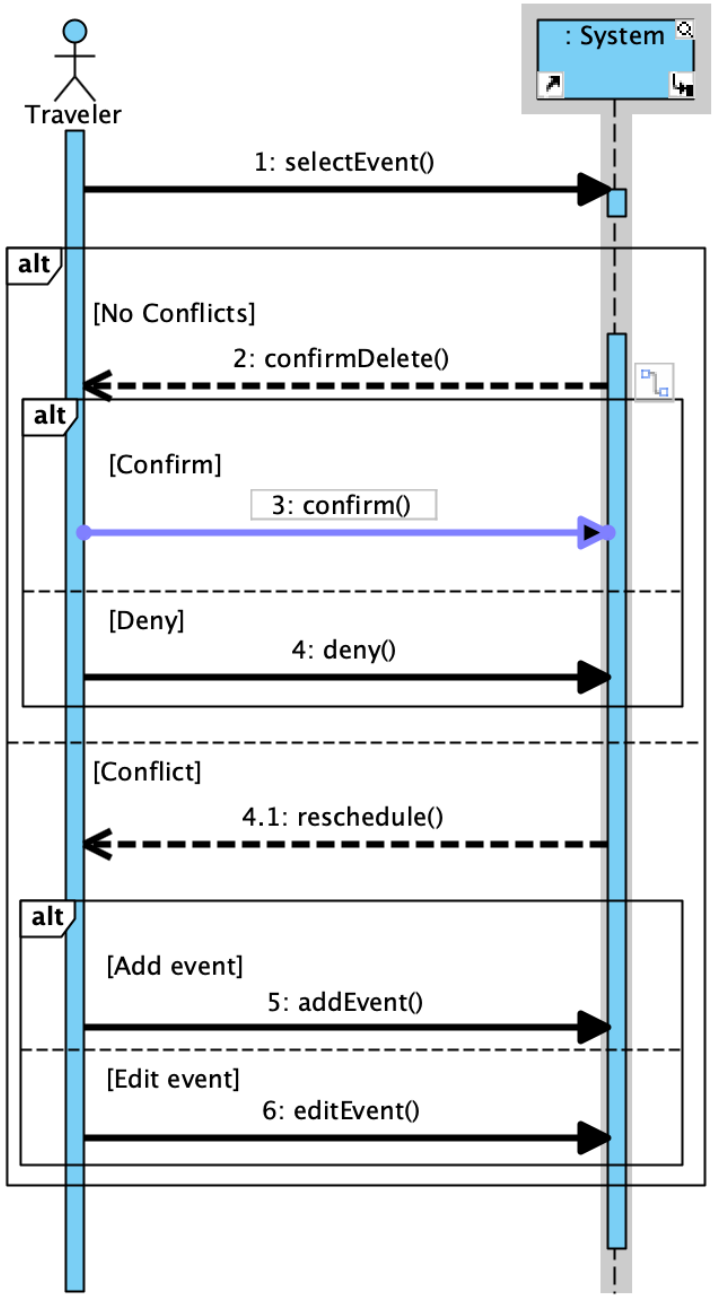
Contract: Archive Event (event)
Operation: Archives the Event, event Cross Reference: Use Cases: Archive Event Pre-conditions: User wants to archive an existing event. Post-condition: The existing event is archived

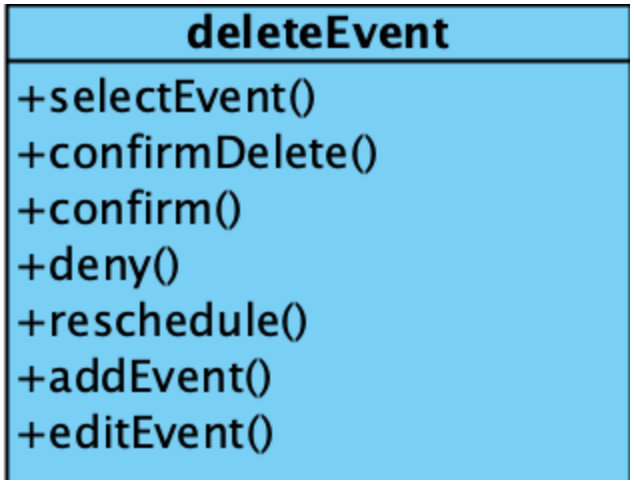

sd [UC Edit Event]
: Traveler
: System
1: editEvent(event)
1.1: conflictState
loop
[conflictState is unexcepted]
alt
[change surrounding events]
2: changeSurroundingEvents(event)
[fit in surrounding events]
3: event = makeFit(event)
[push events forward]
4: pushForward(event)
[push events backward]
5: pushBackward(event)
[push events outward]
6: pushOut(event)
7: getConflictState(event)
7.1: conflictState
8: addEvent(event)
Powered By Visual Paradigm Community Edition



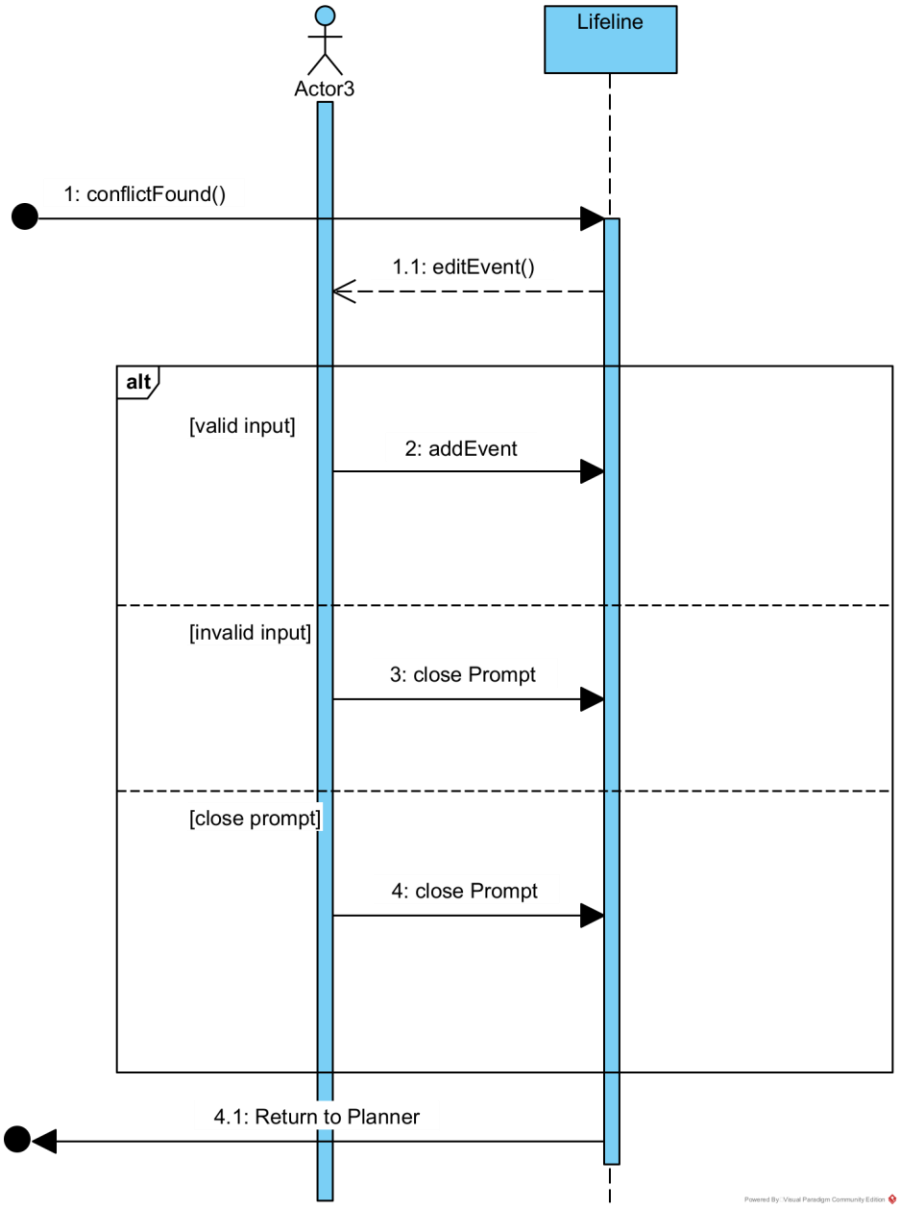
Contract: Add Event (event)
Operation: Adds the event Event Cross Reference: Use Cases: Edit Event Pre-conditions: Prompts to add a new event. Post-condition: A new event is created
Contract: Edit Event (event)
Operation: Edits the event Event Cross Reference: Use Cases: Edit Event Pre-conditions: Prompts to add a new event. Post-condition: A new event is created
Contract: push out (event)
Operation: removes the event Cross Reference: Use Cases: Edit Event Pre-conditions: an event is being removed. Post-condition: the event is removed
Contract: push backward (event)
Operation: pushes event further pack along the planner Cross Reference: Use Cases: Edit Event Pre-conditions: An Event being inserted in front of event . Post-condition: event is pushed back
Contract: push forward(event)
Operation: brings event forward in the planner Cross Reference: Use Cases: Edit Event Pre-conditions: An Event being inserted behind event. Post-condition: event is brought forward
Contract: makeFit(event)
Operation: makes the event fit into the planner Cross Reference: Use Cases: Edit Event Pre-conditions: inserts event at a certain position. Post-condition: A new event is inserted
Contract: change surroundings events (event)
Operation: changes the Events around event Cross Reference: Use Cases: Edit Event Pre-conditions: an event is going to be added. Post-condition: planner is prepped for new event
Contract: get Conflicted State (event)
Operation: checks the date of the new event with the dates of events on the planner Cross Reference: Use Cases: Edit Event

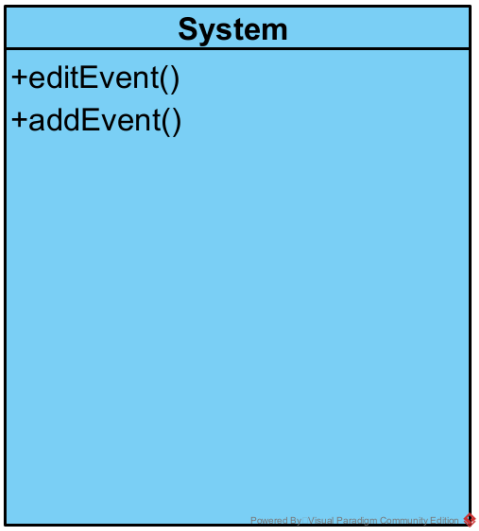
Pre-conditions: no time conflict Post-condition: event times are checked with the new event for conflicts



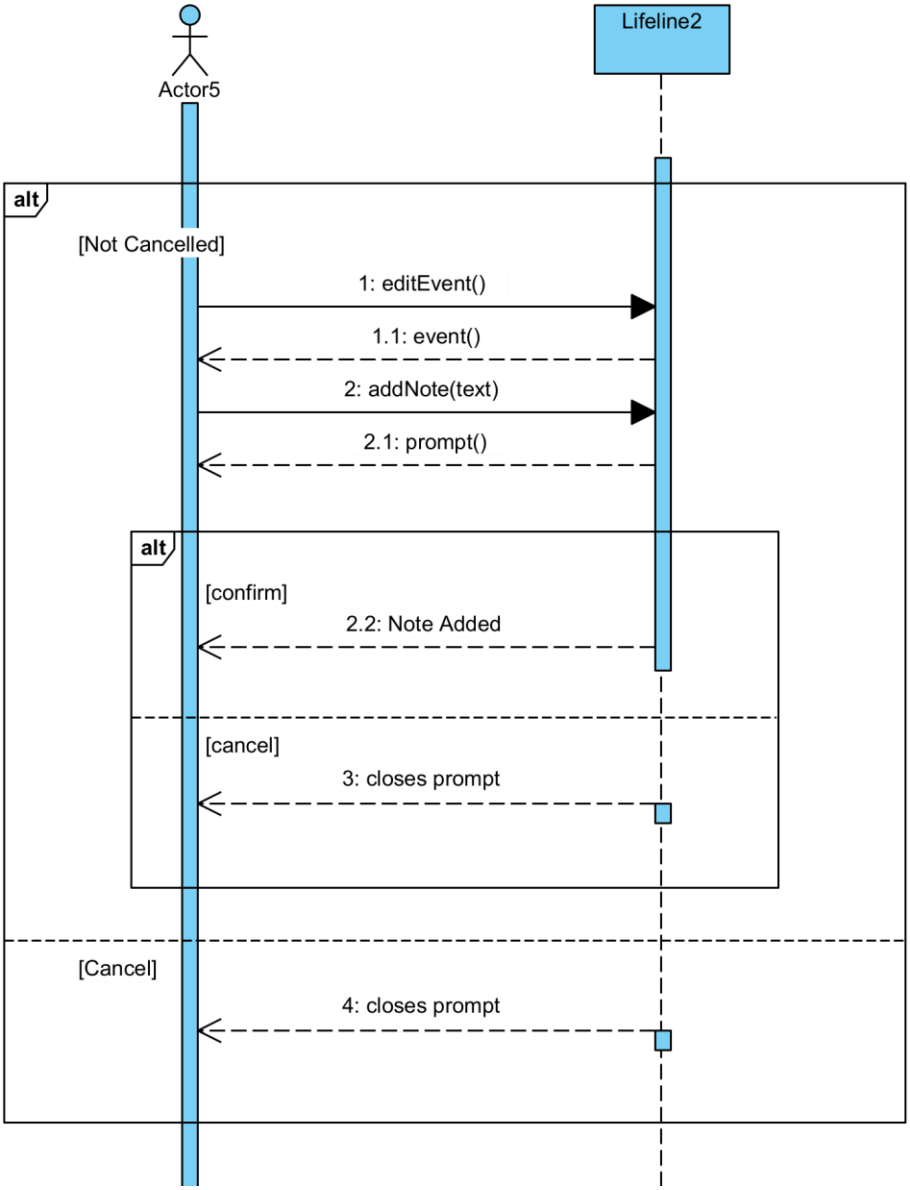


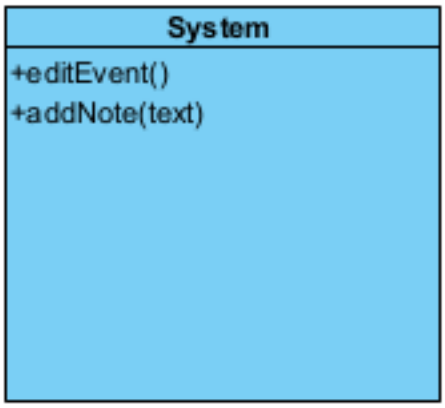
Contract: select Event ()
Operation: Selects an Event to delete Cross Reference: Use Cases: deleteEvent Pre-conditions: No event selected. Post-condition: An event is selected
Contract: confirm Delete()
Operation: Deletes selected event Cross Reference: Use Cases: deleteEvent Pre-conditions: Event selected. Post-condition: Selected event is deleted
Contract: confirm()
Operation: Confirms deletion of an event Cross Reference: Use Cases: deleteEvent Pre-conditions: Selected event . Post-condition: An event is deleted
Contract: deny ()
Operation: Cancels event deletion process Cross Reference: Use Cases: deleteEvent Pre-conditions: Event is selected. Post-condition: No event is selected
Contract: reschedule()
Operation: Reschedules selected event to a different time Cross Reference: Use Cases: deleteEvent Pre-conditions: Event is selected Post-condition: Selected event has new rescheduled
Contract: Add Event ()
Operation: Adds a new Event Cross Reference: Use Cases: delete Event Pre-conditions: Prompts to add a new event. Post-condition: A new event is created
Contract: Edit Event ()
Operation: Edits “existing” Event Cross Reference: Use Cases: delete Event Pre-conditions: Prompts to edit an existing event. Post-condition: An existing event’s valuse are changed


Actor3
Lifeline
1: conflictFound()
1.1: editEvent()
alt
[valid input]
2: addEvent
[invalid input]
3: close Prompt
[close prompt]
4: close Prompt
4.1: Return to Planner
Powered By: Visual Paradigm Community Edition

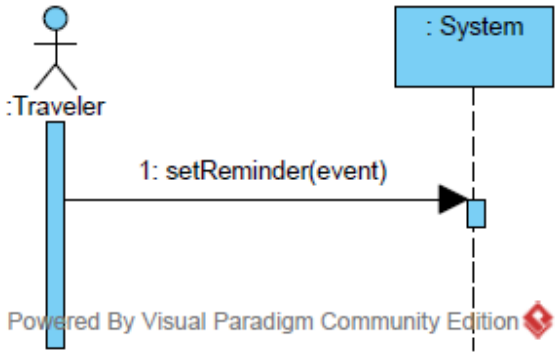

System
+editEvent()
+addEvent()
Powered By: Visual Paradigm Community Edition

Contract: Add Event
Operation: Adds a new Event Cross Reference: Use Cases: Resolve Event Conflicts Pre-conditions: Prompts to add a new event. Post-condition: A new event is created
Contract: Edit Event
Operation: Edits “existing” Event Cross Reference: Use Cases: Resolve Event Conflicts Pre-conditions: Prompts to edit an existing event. Post-condition: An existing event’s valuse are changed

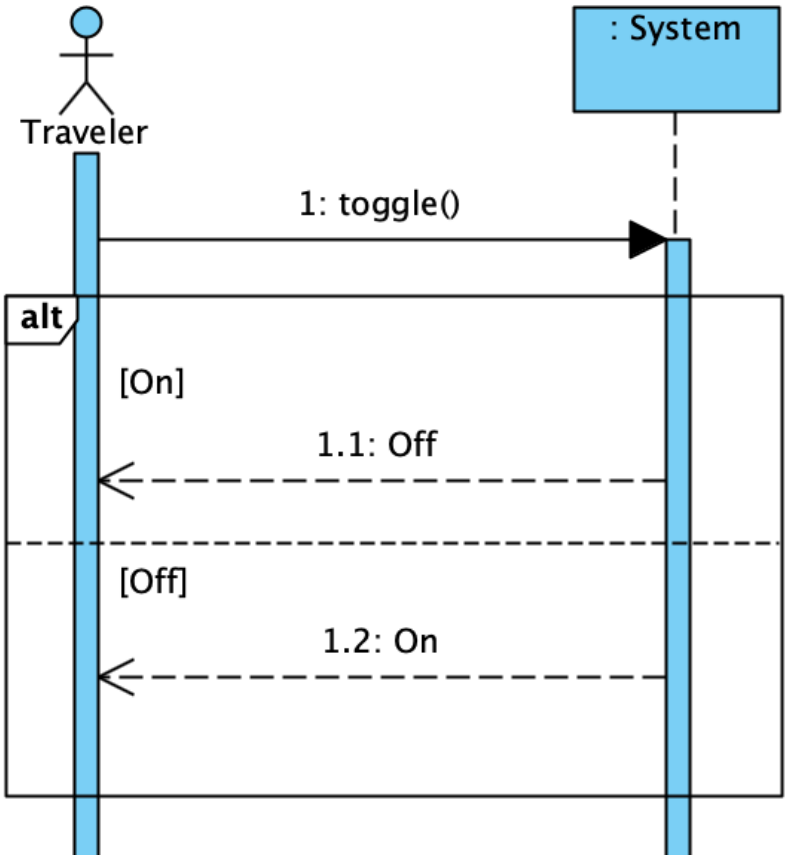


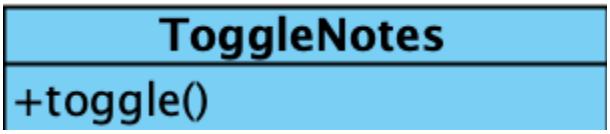


Contract: Add Note (Text)
Operation: Adds a new note to the event with the input text Cross Reference: Use Cases: Event Note Pre-conditions: Prompts to add a new event note Post-condition: A new event note is created
Contract: Edit Event
Operation: Edits “existing” Event Cross Reference: Use Cases: Event Note Pre-conditions: Prompts to edit an existing event. Post-condition: An existing event’s valuse are changed

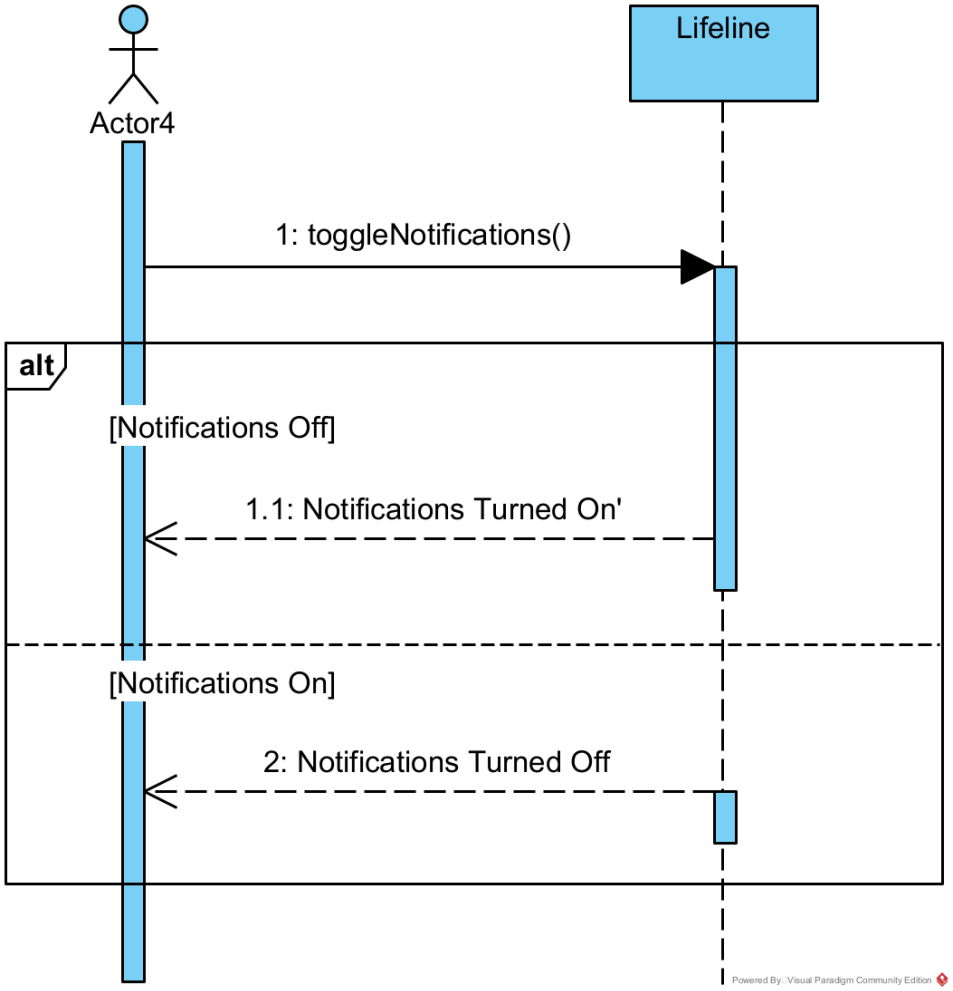


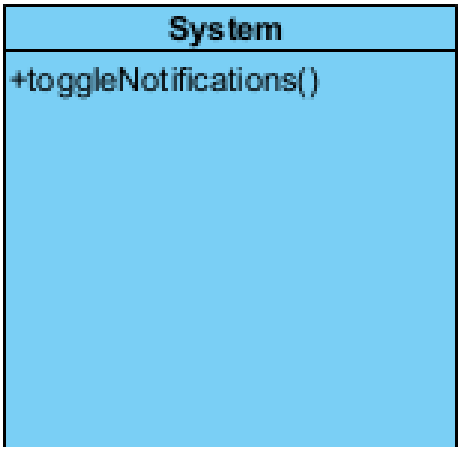
Contract: Event Notification
Operation: set notification of event Cross Reference: Use Cases: Create Event Pre-conditions: Event exists Post-condition: Notification for event is set



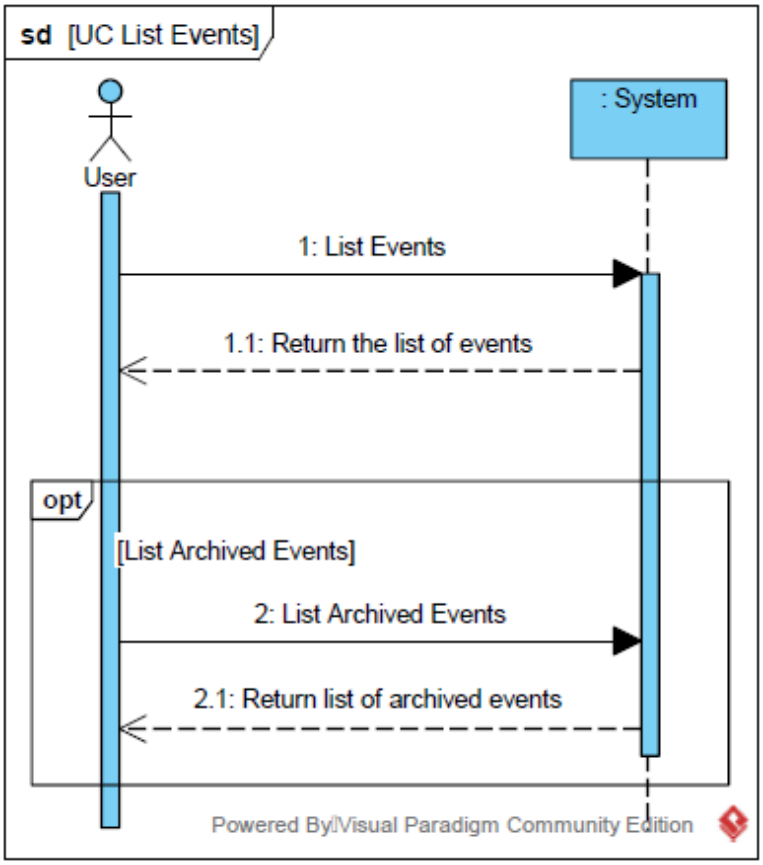


Contract: Toggle Notes
Operation: toggles the notes visible or invisible Cross Reference: Use Cases: Toggle Notes Pre-conditions: The Notes are in a particular state Post-condition: The notes are in the opposite state

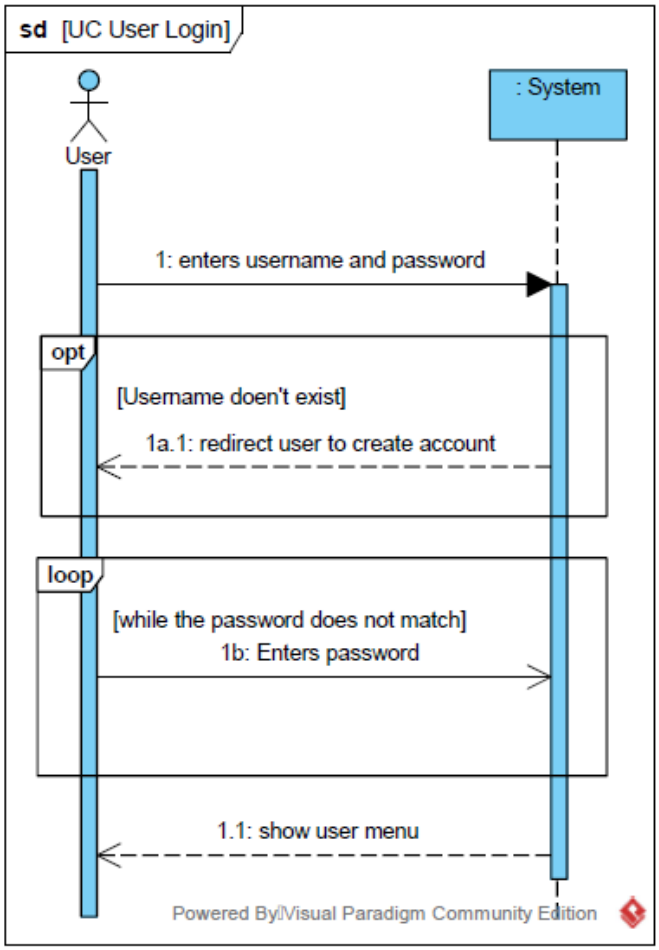




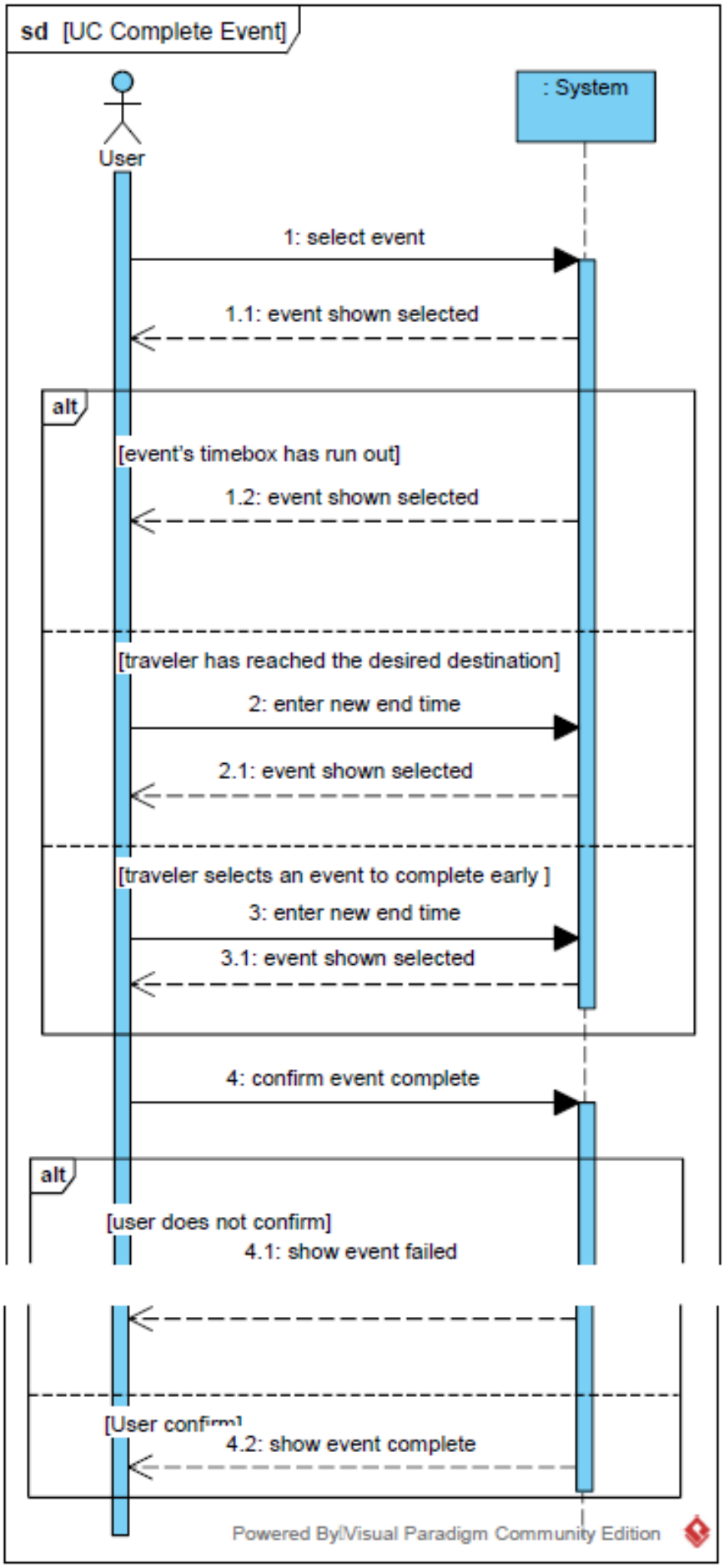
Contract: Toggle Notifications
Operation: Edits “existing” Event Cross Reference: Use Cases: Toggle Notifications Pre-conditions: Prompts to edit an existing event. Post-condition: An existing event’s valuse are changed



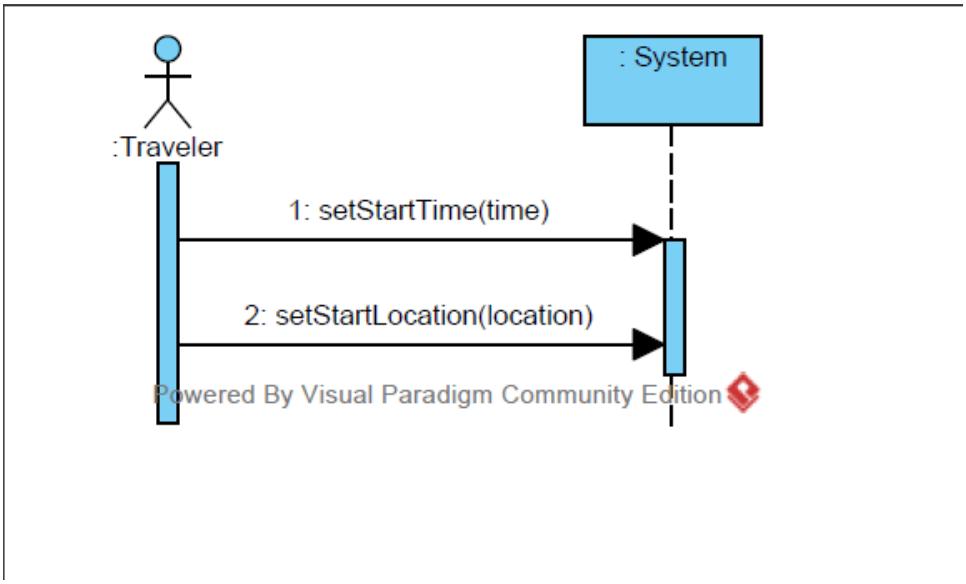
Contract: List Events
Operation: list events Cross Reference: Archive events Pre-conditions: The user is logged in and accessible to the menu Post-condition: the event list is presented to the events.



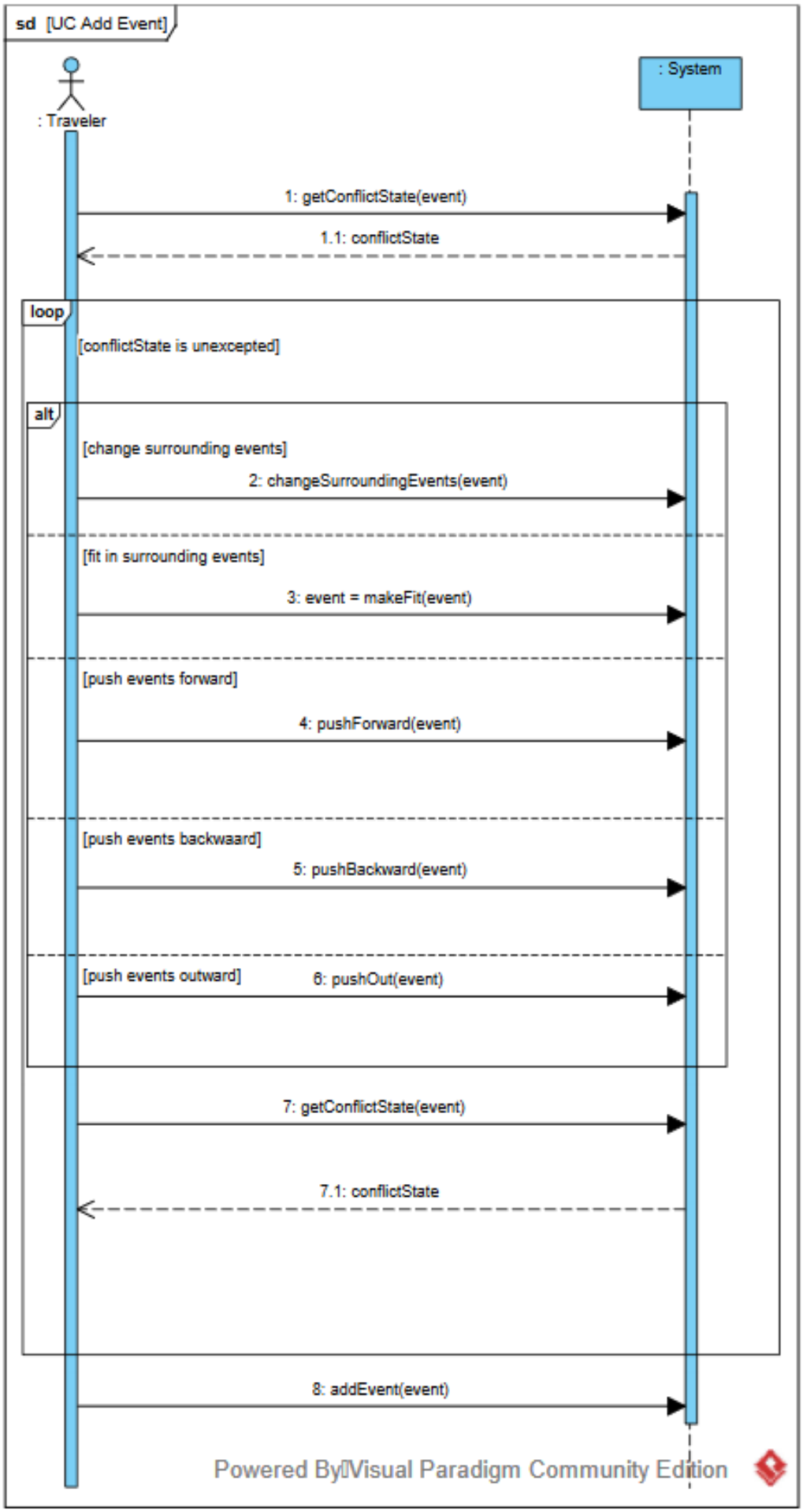
Contract: User Login
Operation: User Login Cross Reference: Use Cases: Create User Pre-conditions: Post-condition: User is logged in

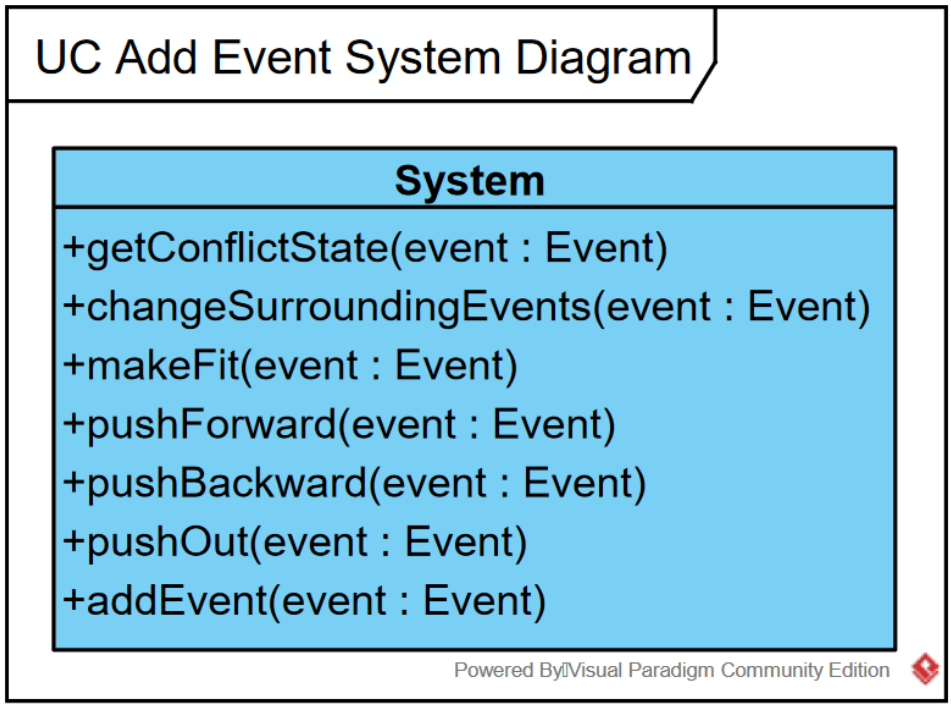

sd [UC Complete Event]
User
: System
1: select event
1.1: event shown selected
alt
[event's timebox has run out]
1.2: event shown selected
[traveler has reached the desired destination]
2: enter new end time
2.1: event shown selected
[traveler selects an event to complete early]
3: enter new end time
3.1: event shown selected
4: confirm event complete
alt
[user does not confirm]
4.1: show event failed
[User confirm]
4.2: show event complete
Powered By Visual Paradigm Community Edition

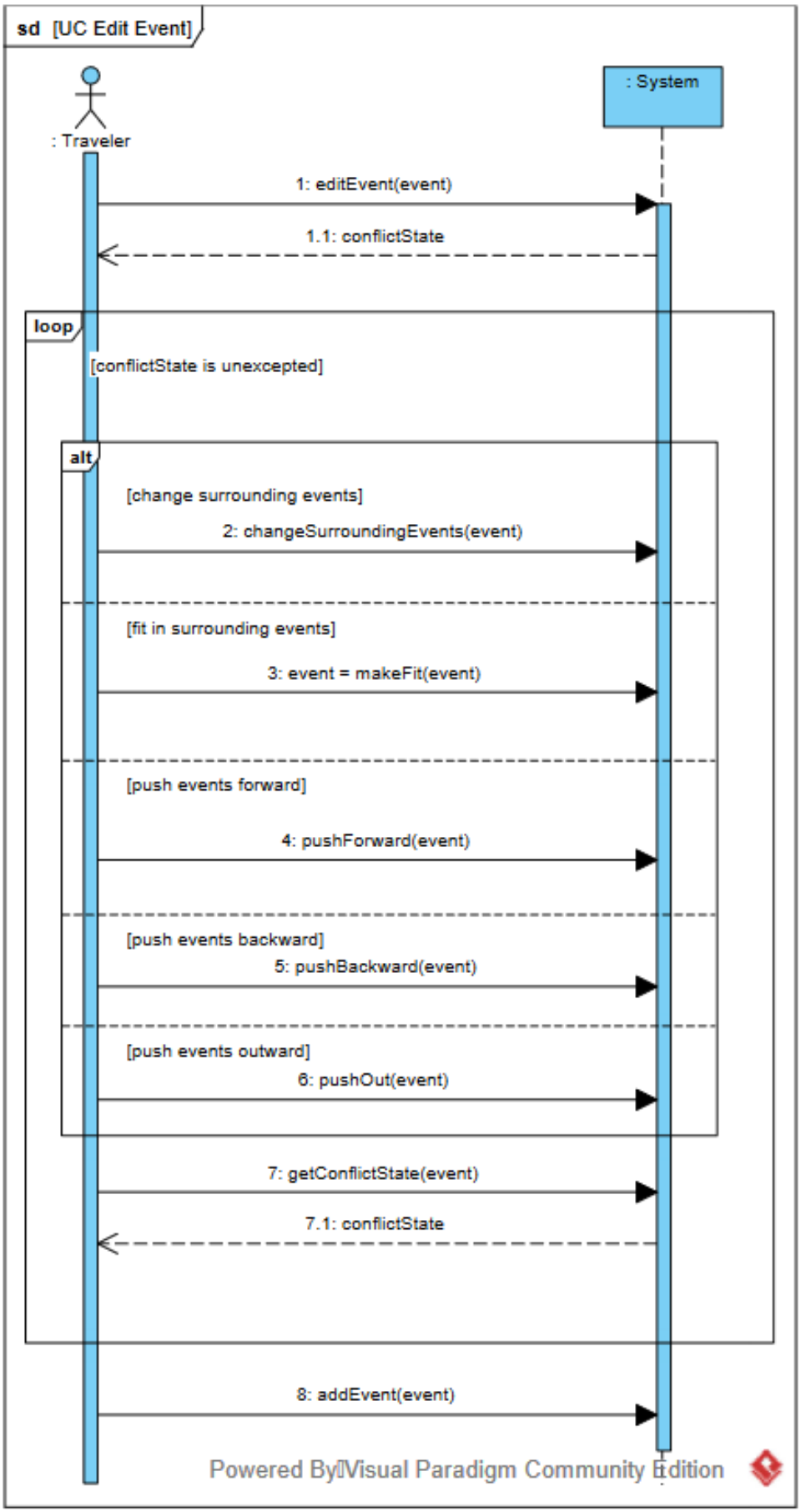
Contract: Complete Events
Operation: Complete events Cross Reference: List events Pre-conditions: The traveler has input an event that they have subsequently completed Post-condition: The event is saved as a completed event

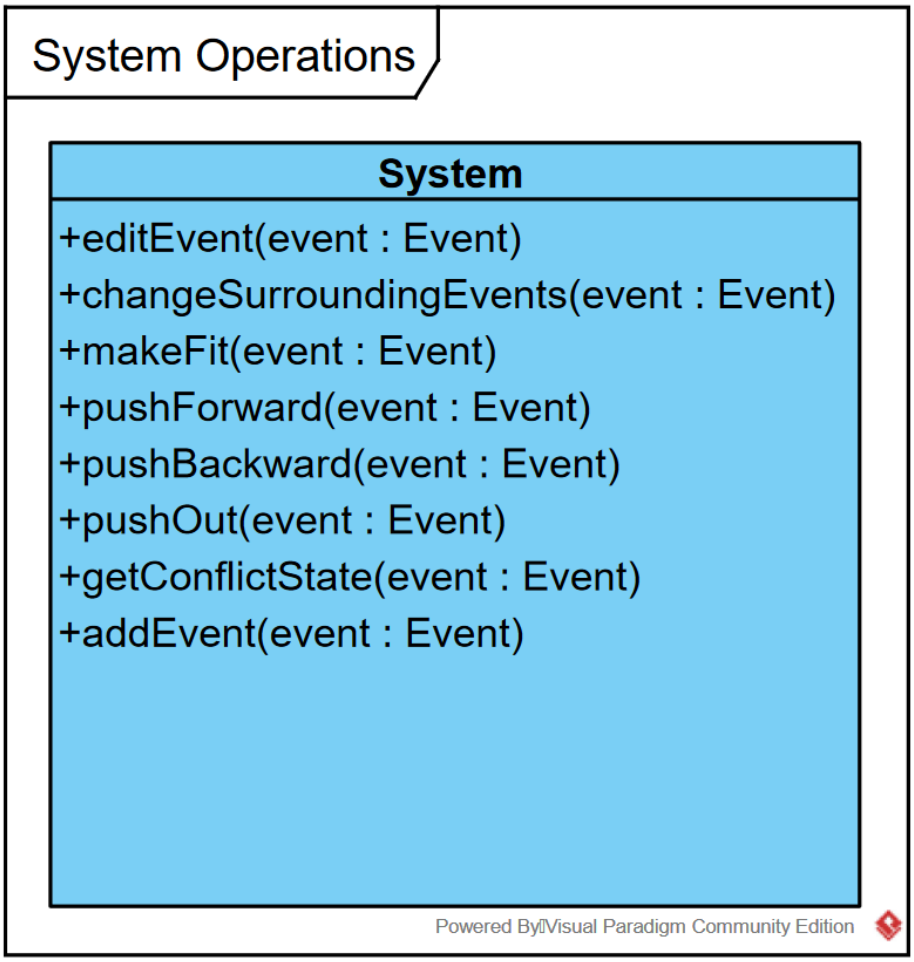


Contract: Set Starting Time and Location
Operation: set time and location Cross Reference: Use Cases: Create Event Pre-conditions: An event is currently being created Post-condition: The time and place has been set for the event


sd [UC Add Event]
: Traveler
: System
1: getConflictState(event)
1.1: conflictState
loop
[conflictState is unexcepted]
alt
[change surrounding events]
2: changeSurroundingEvents(event)
[fit in surrounding events]
3: event = makeFit(event)
[push events forward]
4: pushForward(event)
[push events backwaard]
5: pushBackward(event)
[push events outward]
6: pushOut(event)
7: getConflictState(event)
7.1: conflictState
8: addEvent(event)
Powered By Visual Paradigm Community Edition


UC Add Event System Diagram
System
+getConflictState(event : Event)
+changeSurroundingEvents(event : Event)
+makeFit(event : Event)
+pushForward(event : Event)
+pushBackward(event : Event)
+pushOut(event : Event)
+addEvent(event : Event)
Powered By Visual Paradigm Community Edition


sd [UC Edit Event]
: Traveler
: System
1: editEvent(event)
1.1: conflictState
loop
[conflictState is unexcepted]
alt
[change surrounding events]
2: changeSurroundingEvents(event)
[fit in surrounding events]
3: event = makeFit(event)
[push events forward]
4: pushForward(event)
[push events backward]
5: pushBackward(event)
[push events outward]
6: pushOut(event)
7: getConflictState(event)
7.1: conflictState
8: addEvent(event)
Powered By Visual Paradigm Community Edition



Contract: Get Conflict State
Operation: Get Conflict State Cross Reference: Use Cases: Add Event, Edit Event Pre-conditions: Event exists that may be in conflict with other events. Post-condition: Conflicting Events are known

Contract: Change Surrounding Events
Operation: Change Surrounding Events Cross Reference: Use Cases: Add Event, Edit Event Pre-conditions: There is an Event that has conflicts with other events Post-condition: Conflicting events start and stop time are changed to reflect the start and stop time of the given event.

Contract: Make Fit
Operation: Make Fit Cross Reference: Use Cases: Add Event, Edit Event Pre-conditions: There is an Event that has conflicts with other events Post-condition: Given event's start and stop time is changed to reflect the start and stop time of theconflicting events.

Contract: Push Forward
Operation: Push Forward

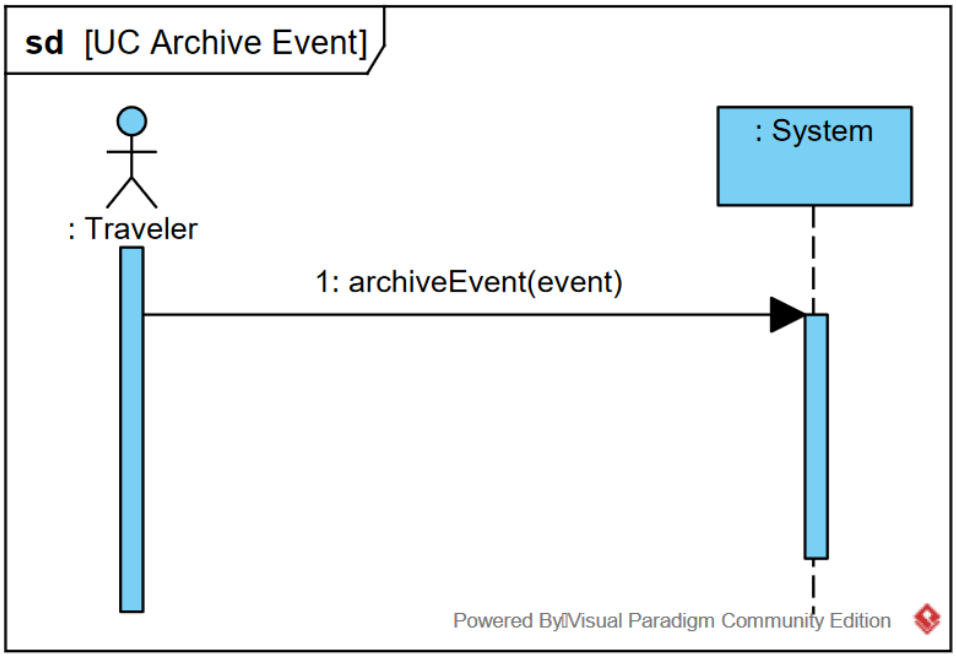
Cross Reference: Use Cases: Add Event, Edit Event
Pre-conditions: There is an Event that has conflicts with other events
Post-condition: All conflicting future events start and stop time are increased so that the given event does not have conflict with them.

Contract: Push Backward
Operation: Push Backward Cross Reference: Use Cases: Add Event, Edit Event Pre-conditions: There is an Event that has conflicts with other events Post-condition: All conflicting past events start and stop time are decreased so that the given event does not have conflict with them.

Contract: Push Out
Operation: Push Out Cross Reference: Use Cases: Add Event, Edit Event Pre-conditions: There is an Event that has conflicts with other events Post-condition: All conflicting future events start and stop time are increased so that the given event does not have conflict with them and all conflicting past events start and stop time are decreased so that the given event does not have conflict with them.

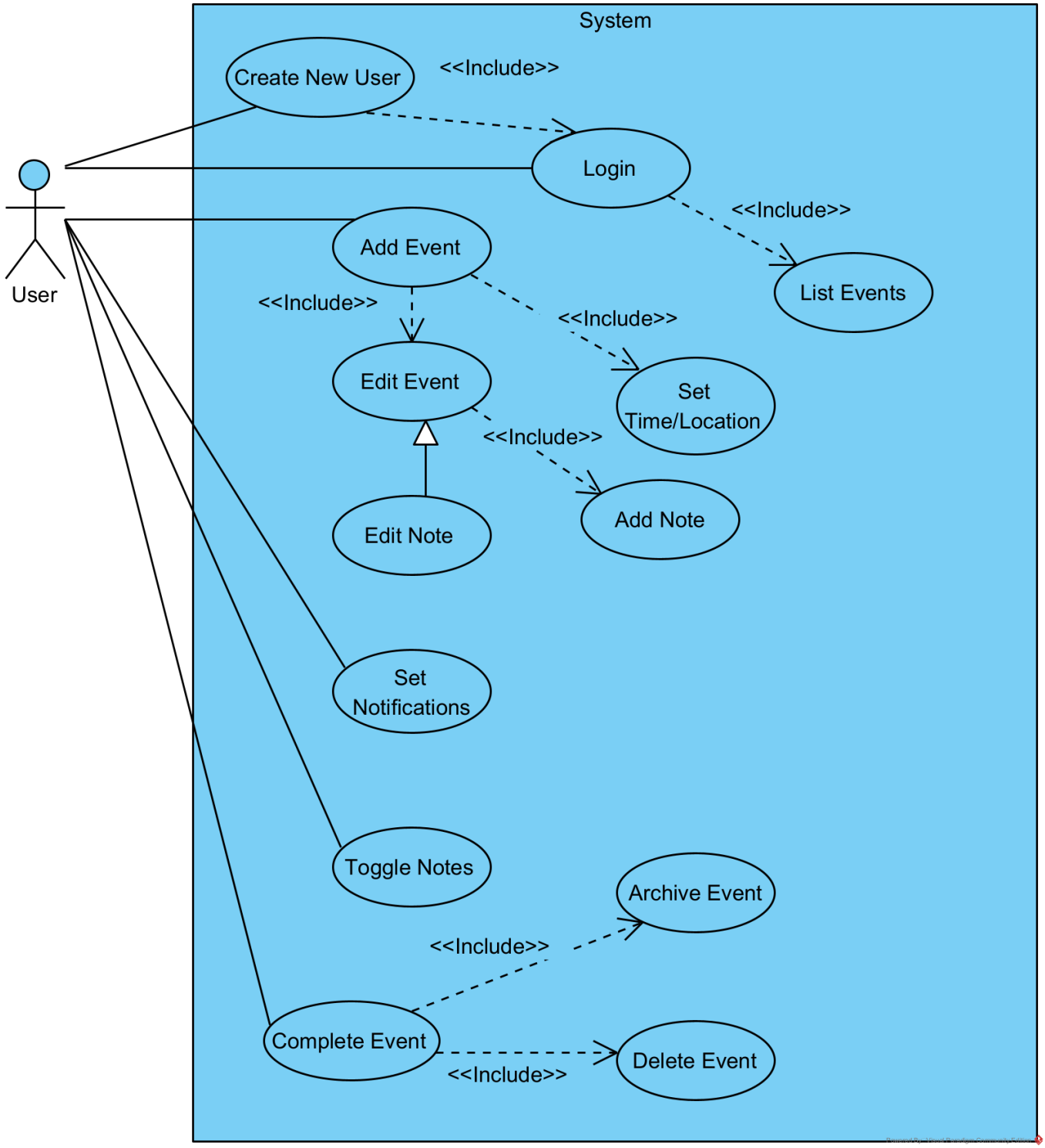
Contract: Add Event
Operation: Add Event Cross Reference: Use Cases: Add Event, Edit Event Pre-conditions: There is an event that is not in the Planner Post-condition: The event is added to the planner and conflicting events are identified and referenced.

Contract: Edit Event
Operation: Edit Event Cross Reference: Use Cases: Edit Event Pre-conditions: There is an event that is in the planner Post-condition: The the event is removed from the planner for editing and all conflict links are removed between this and other conflicting events.

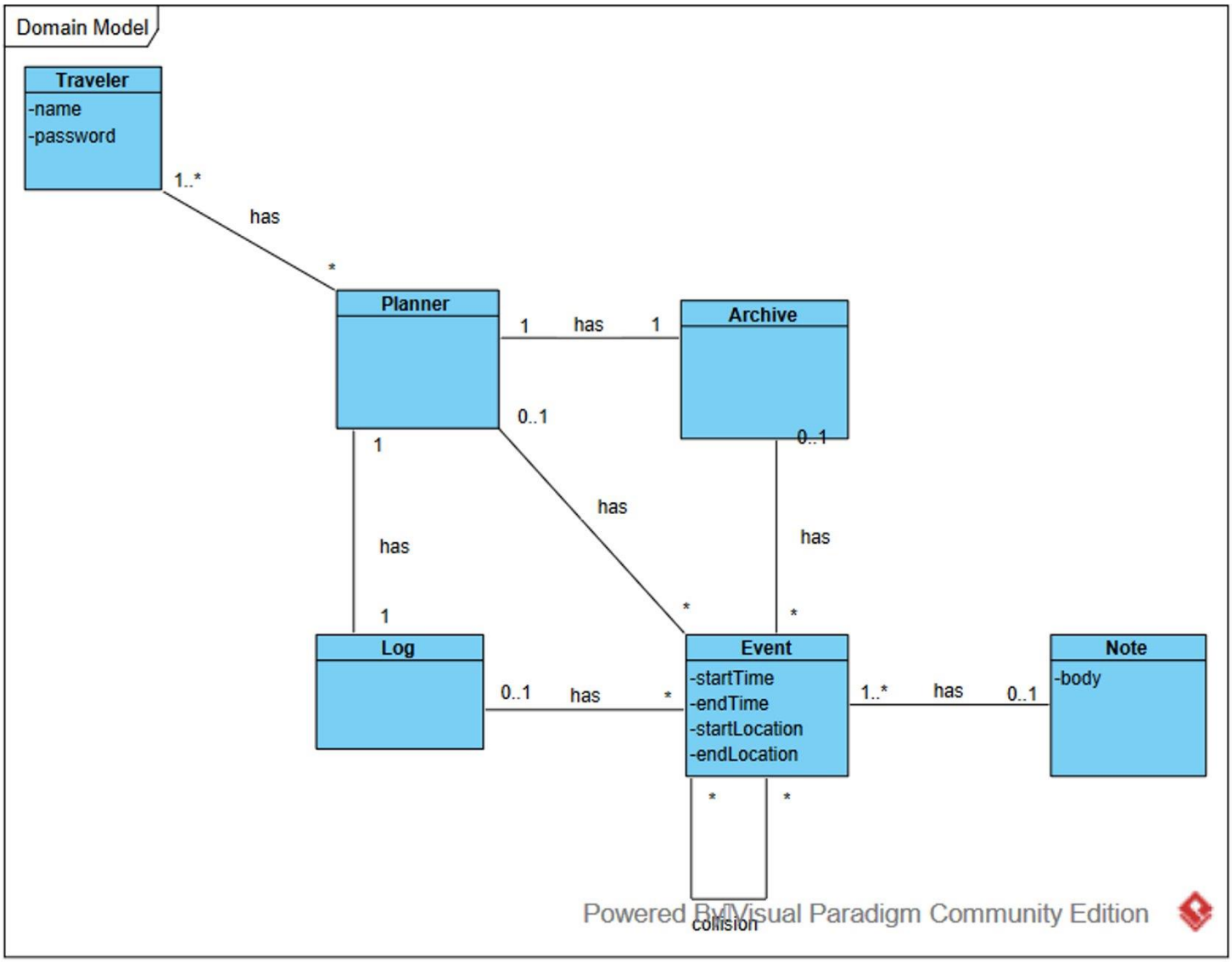


Contract: Archive Event
Operation: Archive Event Cross Reference: Use Cases: Archive Event Pre-conditions: There is an event that is in the planner or log Post-condition: The event is stored in the archive.

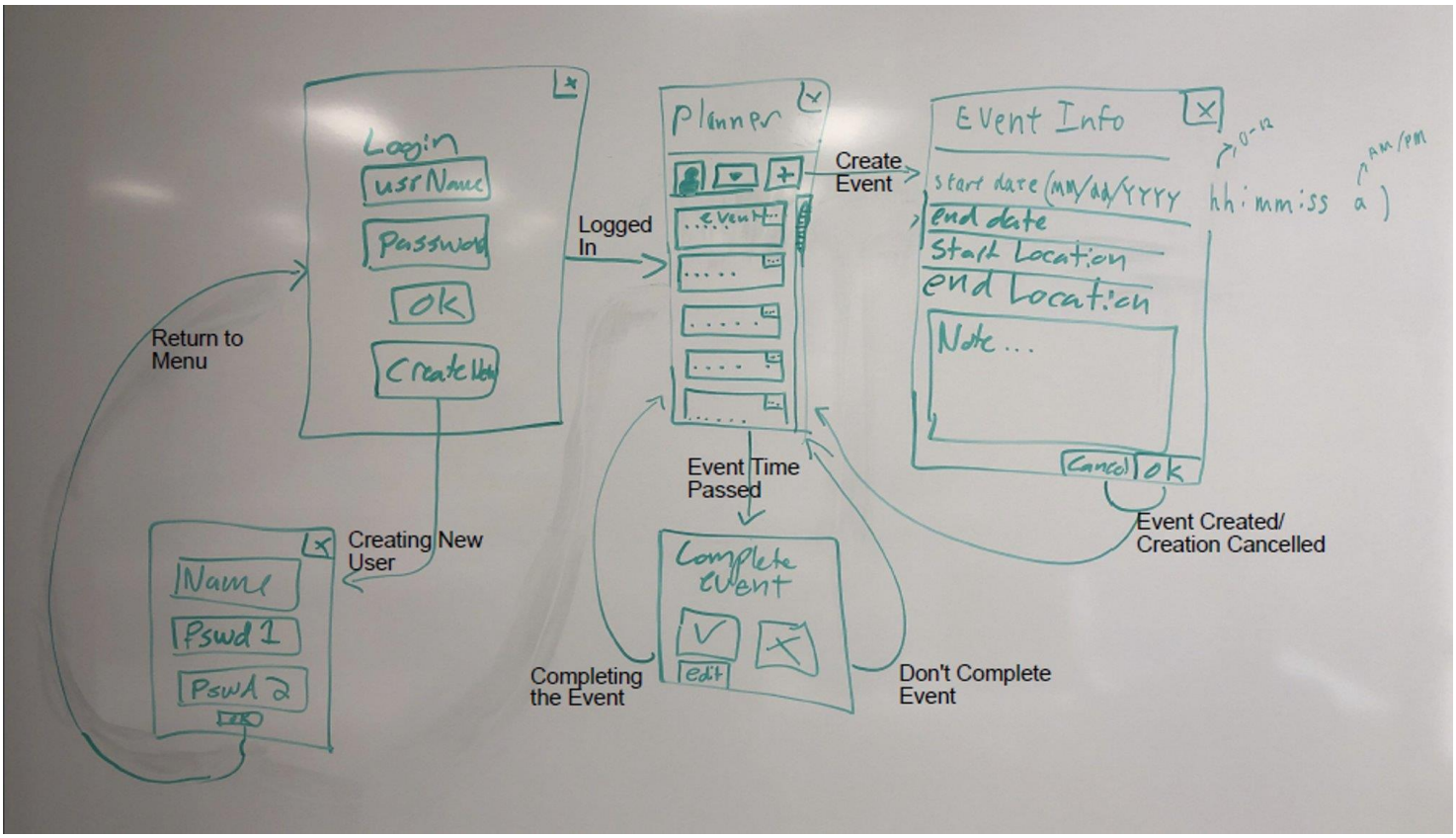
Use Case Diagram



Domain Model



Wire Frame


Login
usrName
Password
ok
Create Usr
Return to Menu
Logged In
Planner
Create Event
Event Info
start date (MM/dd/YYYY
hh:mm:ss
end date
Start Location
end Location
Note...
Cancel
ok
Event Time Passed
Event Created/ Creation Cancelled
Name
Pswd 1
Pswd 2
Creating New User
Complete Event
edit
Completing the Event
Don't Complete Event

Gantt Diagram

Untitled Gantt Project

Aug 22, 2022

http://

Project manager	
Project dates	Aug 22, 2022 - Dec 30, 2022
Completion	0%
Tasks	28
Resources	4

Tasks

3

Name	Begin date	End date
presentation and reporting	12/29/22	12/29/22

Tasks

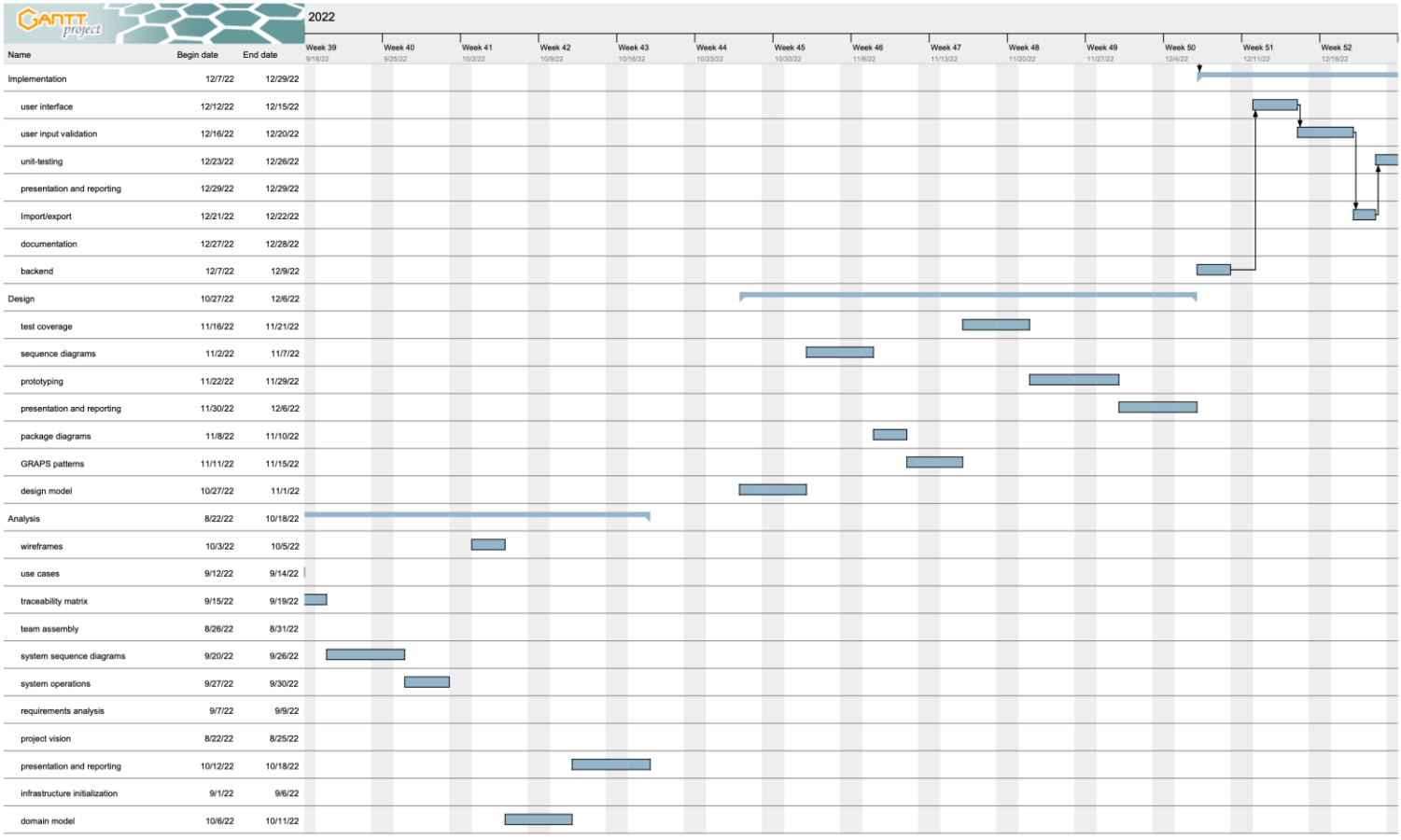
2

Name	Begin date	End date
Analysis	8/22/22	10/18/22
project vision	8/22/22	8/25/22
team assembly	8/26/22	8/31/22
infrastructure initialization	9/1/22	9/6/22
requirements analysis	9/7/22	9/9/22
use cases	9/12/22	9/14/22
traceability matrix	9/15/22	9/19/22
system sequence diagrams	9/20/22	9/26/22
system operations	9/27/22	9/30/22
wireframes	10/3/22	10/5/22
domain model	10/6/22	10/11/22
presentation and reporting	10/12/22	10/18/22
Design	10/27/22	12/6/22
design model	10/27/22	11/1/22
sequence diagrams	11/2/22	11/7/22
package diagrams	11/8/22	11/10/22
GRAPS patterns	11/11/22	11/15/22
test coverage	11/16/22	11/21/22
prototyping	11/22/22	11/29/22
presentation and reporting	11/30/22	12/6/22
Implementation	12/7/22	12/29/22
backend	12/7/22	12/9/22
user interface	12/12/22	12/15/22
user input validation	12/16/22	12/20/22
Import/export	12/21/22	12/22/22
unit-testing	12/23/22	12/26/22
documentation	12/27/22	12/28/22

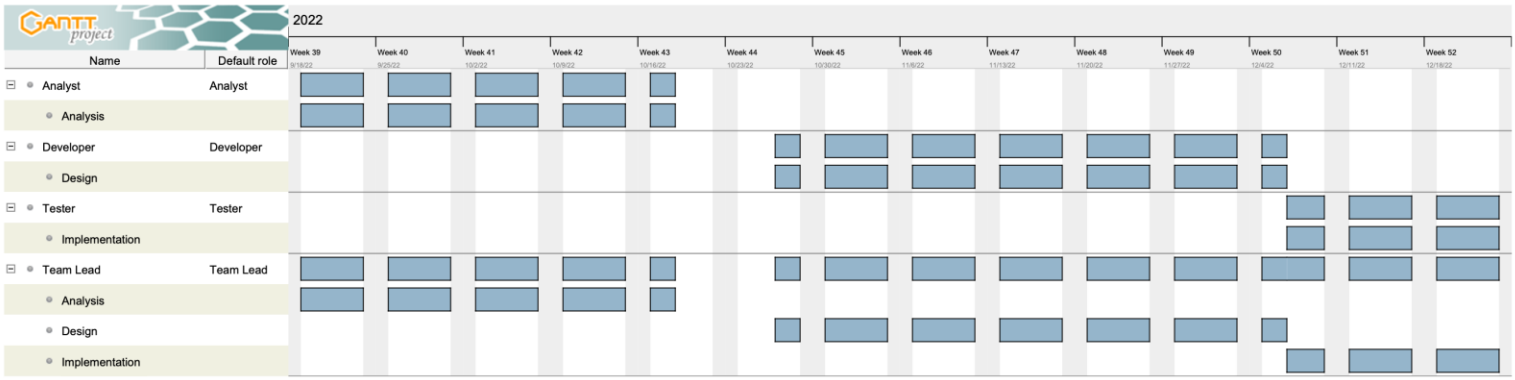
Resources

Name	Default role
Analyst	Analyst
Developer	Developer
Tester	Tester
Team Lead	Team Lead

Gantt Chart



Resources Chart



Time-Card

Benjamin: hours
- Domain Diagram
- SSD
- Fully Dressed Use Case
- Wire frame
- Glossary
- Group Avatar

Bryce: hours
- Requirements
- Use Case
- SSD
- Fully Dressed Use Case

Jason: 10 hours
- **System Operations**
- **Traceability matrix**
- **Fully Dressed Use Case**
- **Requirements**
- **Wire frame**
- **Project Planning**
- **Project Time Management**
- **SSD**
- **Fully Dressed Use Case**
- **Trello**
- **Project vision**
- **Use Case Diagram**

Yutai: hours
- Gantt Diagram
- Use Case
- SSD
- Fully Dressed Use Case

Yi: 10 hours
- Website
- PPT
- Use Case
- SSD
- Fully Dressed Use Case
- Trello