Project Part 2

Personal Activity Assistant

Team: Bharat Nallan Chakravarthy
Megha Sree Yadla
Swati Patil

Title: Personal Activity Assistant

Table of Contents

1. Project Summary

Personal Activity Assistant is a web based application that enables a user to organize their activities. A user can create an account and manage their daily activities as well as track their expenses. A user can maintain an address book with name, email ID and phone number. A person using this application will also be able to schedule appointments such as personal, professional, etc. Users can also make notes and set reminders for any important tasks.

2. Project Requirements

2.1 Business Requirements

ID	Requirement	Topic Area	User	Priority
BR-01	All signups require a gmail ID.	Signup	User	High
BR-02	Each account corresponds to a unique mail ID.	Signup	User	High

2.2 User Requirements

ID	User	Description	Priority
UR-01	User	As a user, I want to see a dashboard listing all the tasks that can be performed using the Personal Activity Assistant.	High
UR-02	User	As a user, I want to be able to view, add, delete or update an entry to the address book.	Medium
UR-03	User	As a user, I want to be able to view, add, delete, undo or update an entry to the expense list.	Medium
UR-04	User	As a user, I want to be able to view, add, delete or update a note.	Low
UR-05	User	As a user, I want to be able to view, add, delete or update an appointment.	Medium
UR-06	User	As a user, I want to be able to view, add, delete or update a reminder.	Medium

2.3 Functional Requirements

ID	Description	Priority
FR-01	Multiple appointments for the same time is not allowed.	High
FR-02	In the addressbook, duplicate entries for name, email ID and phone number is not allowed.	High

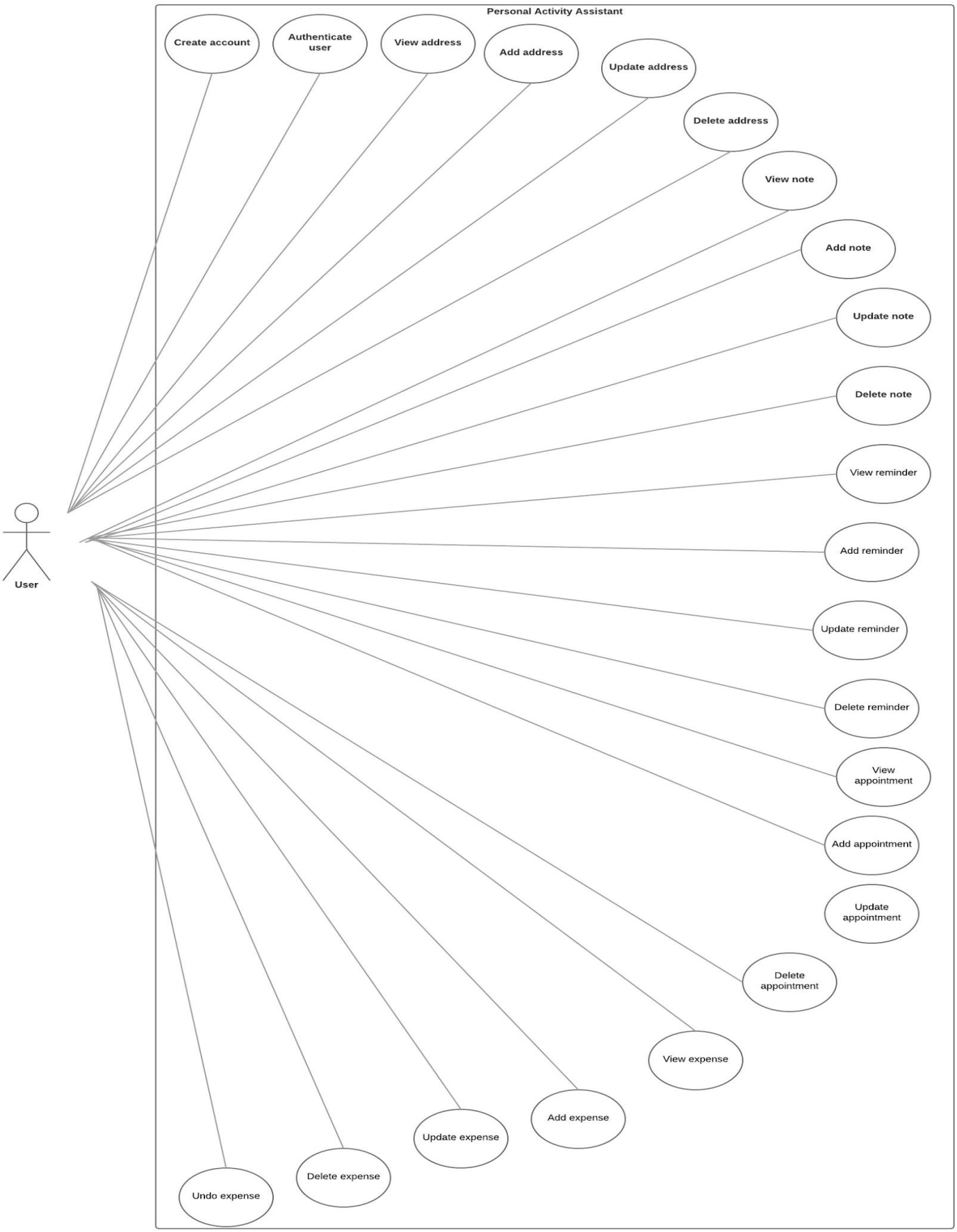
2.4 Non-Functional Requirements

ID	Description	Priority
NFR-01	Security: All passwords should be hashed before saving in the database.	Critical
NFR-02	Performance: When user logs in from a new device, a mail should be sent to the user's mail ID.	Medium
NFR-03	Platform Constraints: The functionality should be supported across various browsers.	High

3. Use Cases

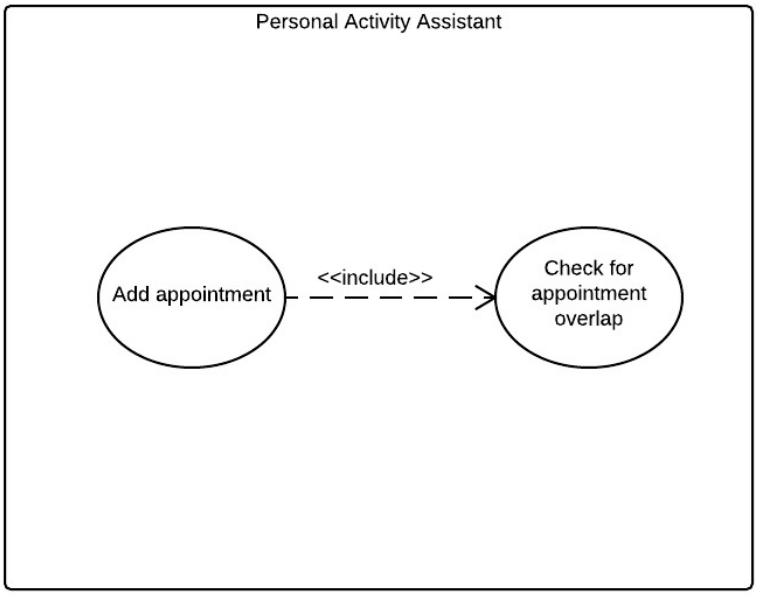
3.1 Actors: Users

3.2 Use Case Overview:

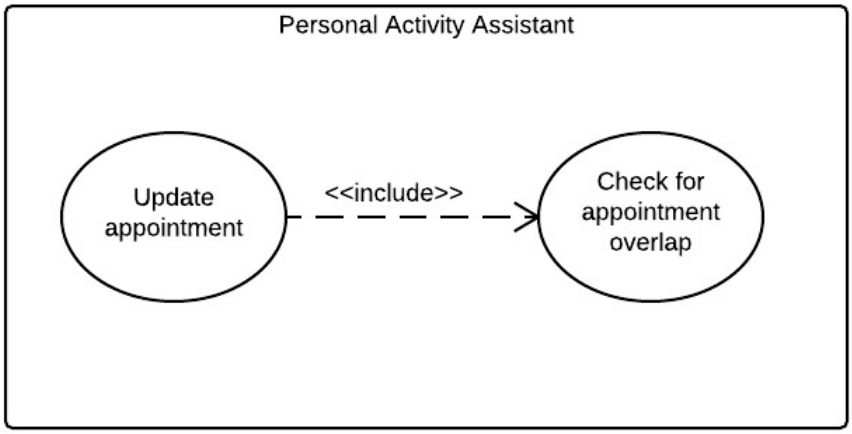


3.3 Sub-diagrams:

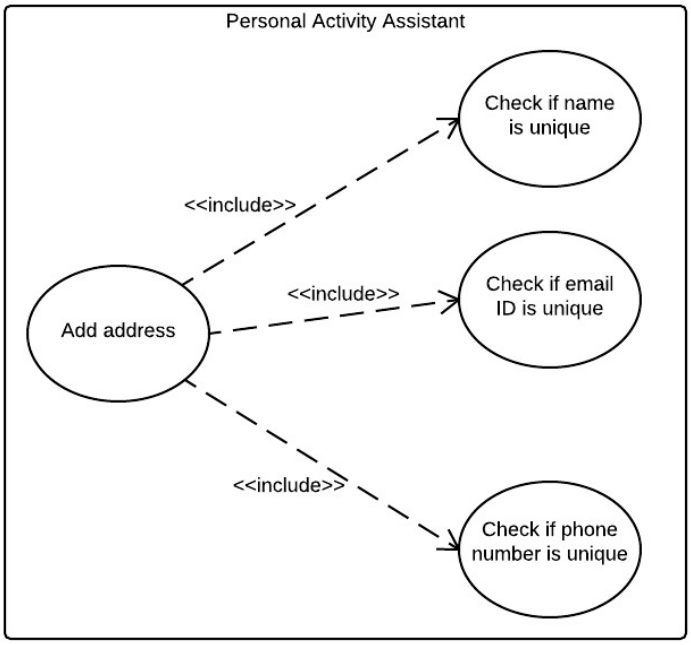
Use Case UC 16: Add appointment



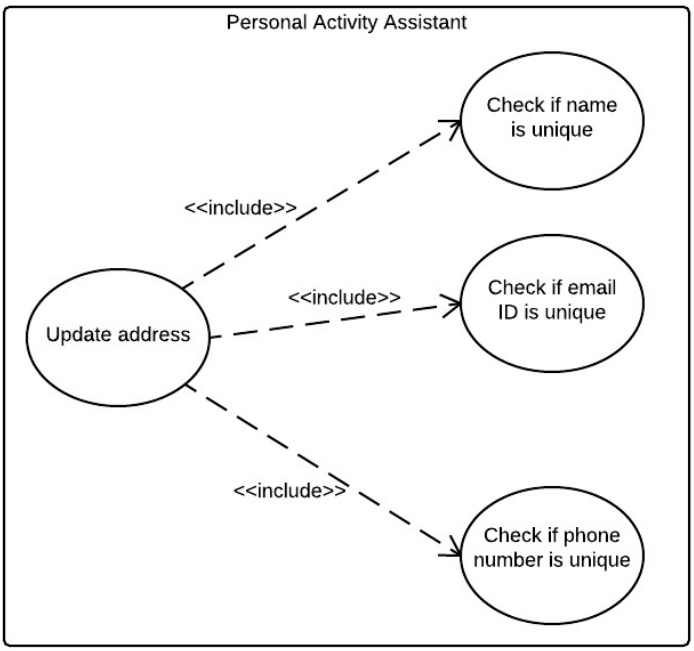
Use Case UC 17: Update appointment



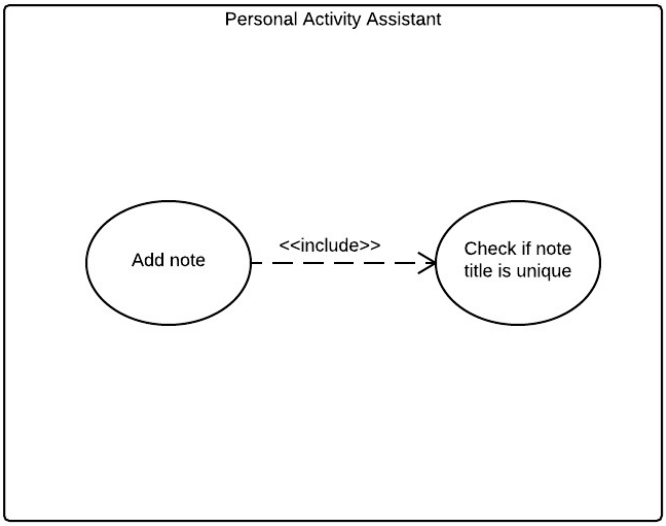
Use Case UC 04: Add address



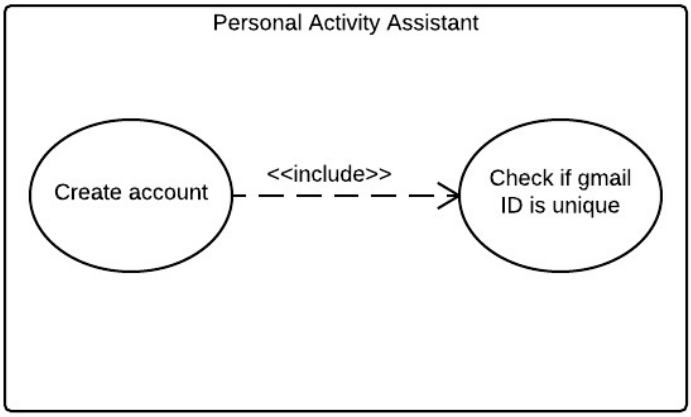
Use Case UC 05: Update address



Use Case UC 08: Add note



Use Case UC 01: Create account



3.4. Use Case Documents:

Use Case ID:	UC 01
Use Case Name:	Create account
Description:	User creates an account on the personal activity assistant.

Actors:	User
Pre-Conditions:	User opens the web application.
Post-Conditions:	User account is created.
Frequency Use:	Frequently when a new user visits the page and wants to have access to their own personal activity assistant.
Flow of Events:	

<table>
<tr><td></td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Enter an email ID.</td><td>System checks if it is a unique gmail ID.</td></tr>
<tr><td>2.</td><td>Enter a password.</td><td>System creates an account for the user and saves the user credentials in the database.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>1. The email ID entered by the user is not a gmail ID.
2. The gmail ID entered already exists.</td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

Use Case ID:	UC 02
Use Case Name:	Authenticate user
Description:	User logs in to the web application.

<table>
<tr><td>Actors:</td><td>User</td></tr>
<tr><td>Pre-Conditions:</td><td>1.User opens the web application.
2.User has a Personal Activity Assistant account.</td></tr>
<tr><td>Post-Conditions:</td><td>User is logged in to their account.</td></tr>
<tr><td>Frequency Use:</td><td>Frequently when the user visits the page and wants to access their account.</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Enter the user ID.</td><td>System accepts the ID entered.</td></tr>
<tr><td>2.</td><td>Enter the password.</td><td>System accepts the password.</td></tr>
</table>
</td></tr>
</table>

	<table><tr><td>3.</td><td>Click on the login button.</td><td>System checks the user credentials and logs the user into their Personal Activity Assistant Application. It also displays the user's dashboard.</td></tr></table>
Variations:	1. The ID or password entered by the user are invalid. The user is directed to the login page with an error message.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 03
Use Case Name:	View address
Description:	User accesses the address book and views an entry in it.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User accesses the address book and views an entry in it.
Frequency Use:	Frequently when the user wants to view an entry in the address book.
Flow of Events:	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1.</td><td>Enter a name.</td><td>System returns the entry with the name.</td></tr></table>
Variations:	1. An entry for the name searched for by the user does not exist in the address book.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 04

Use Case Name:	Add address
Description:	User accesses the address book and adds an entry to it.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User accesses the address book and adds an entry to it.
Frequency Use:	Frequently when the user wants to add an entry to the address book.
Flow of Events:	

	Actor Action	System Response
1.	Enter a name.	System checks if the name already exists and creates an entry with the name.
2.	Enter an address.	System accepts the address and adds it to the entry.
3.	Enter an email ID.	System checks if the email ID already exists and adds it to the entry.
4.	Enter a phone number.	System checks if the phone number already exists and adds it to the entry.

Variations:	1. The name, email ID or phone number already exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 05
Use Case Name:	Update address
Description:	User accesses the address book and edits an entry in it.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User accesses the address book and edits an entry in it.
Frequency Use:	Frequently when the user wants to edit an entry in the address book.
Flow of Events:	

	Actor Action	System Response
1.	Search for the entry to be edited by the name.	System searches for the name and returns the entry of that name.
2.	Edit the entry.	System checks for duplicate name, email ID or phone number.

Variations:	1. The name searched for by the user does not exist in the address book. 2. The edited name, email ID or phone number entries already exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 06
Use Case Name:	Delete address
Description:	User accesses the address book and deletes an entry in it.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User accesses the address book and deletes an entry in it.
Frequency Use:	Frequently when the user wants to delete an entry in the address book.
Flow of Events:	

	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1.</td><td>Search for the entry to be deleted by the name.</td><td>System searches for the name and returns the entry of that name.</td></tr><tr><td>2.</td><td>Delete the entry.</td><td>System deletes the entry from the address book.</td></tr></table>
Variations:	1. The name searched for by the user does not exist in the address book.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 07
Use Case Name:	View note
Description:	User views a note in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User views a note in their personal activity assistant.
Frequency Use:	Frequently when the user wants to view a note.
Flow of Events:	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1.</td><td>Enter note title.</td><td>System returns the note with the title.</td></tr></table>
Variations:	1. A note with the title entered does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 08

Use Case Name:	Add note
Description:	User adds a note in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User adds a note in their personal activity assistant.
Frequency Use:	Frequently when the user wants to add a note.
Flow of Events:	

	Actor Action	System Response
1.	Enter note title.	System checks if the title is unique and creates a note with the title.
2.	Enter the content of the note.	System adds the content to the note created.

Variations:	1. A note with the same title already exists.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 09
Use Case Name:	Update note
Description:	User updates a note in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User updates a note in their personal activity assistant.
Frequency Use:	Frequently when the user wants to update a note.
Flow of Events:	

	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1.</td><td>Enter note title.</td><td>System searches for the title and returns the note with that title.</td></tr><tr><td>2.</td><td>Update the note.</td><td>System updates the contents of the note.</td></tr></table>
Variations:	1. Note with the title searched for by the user does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 10
Use Case Name:	Delete note
Description:	User deletes a note from their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User deletes a note from their personal activity assistant.
Frequency Use:	Frequently when the user wants to delete a note.
Flow of Events:	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1.</td><td>Enter note title.</td><td>System searches for the title and returns the note with that title.</td></tr><tr><td>2.</td><td>Delete the note.</td><td>System deletes the note.</td></tr></table>
Variations:	1. Note with the title searched for by the user does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 11

Use Case Name:	View reminder
Description:	User views a reminder in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User views a reminder in their personal activity assistant.
Frequency Use:	Frequently when the user wants to view a reminder.
Flow of Events:	

	Actor Action	System Response
1.	Enter reminder title.	System returns the reminder with the entered title.

Variations:	1. Reminder with the title entered does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 12
Use Case Name:	Add reminder
Description:	User adds a reminder in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User adds a reminder in their personal activity assistant.
Frequency Use:	Frequently when the user wants to add a reminder.
Flow of Events:	

	Actor Action	System Response
1.	Enter reminder title.	System creates a

	<table><tr><td></td><td></td><td>reminder with the entered title.</td></tr><tr><td>2.</td><td>Enter reminder date and time.</td><td>System adds the date and time to the reminder.</td></tr><tr><td>3.</td><td>Enter a description of the reminder.</td><td>System adds the description to the reminder.</td></tr></table>
Variations:	
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 13
Use Case Name:	Update reminder
Description:	User updates a reminder in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User updates a reminder in their personal activity assistant.
Frequency Use:	Frequently when the user wants to update a reminder.
Flow of Events:	<table><tr><td></td><td>Actor Action</td><td>System Response</td></tr><tr><td>1.</td><td>Enter the title of the reminder.</td><td>System searches for a reminder with the entered title and returns it.</td></tr><tr><td>2.</td><td>Update the reminder.</td><td>System updates the reminder.</td></tr></table>
Variations:	1. Reminder with the title searched for by the user does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 14
Use Case Name:	Delete reminder
Description:	User deletes a reminder from their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User deletes a reminder from their personal activity assistant.
Frequency Use:	Frequently when the user wants to delete a reminder.
Flow of Events:	

	Actor Action	System Response
1.	Enter the title of the reminder.	System searches for a reminder with the entered title and returns it.
2.	Delete the reminder.	System deletes the reminder.

Variations:	1. Reminder with the title searched for by the user does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 15
Use Case Name:	View appointment
Description:	User views an appointment in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User views an appointment in their personal activity assistant.

<table>
<tr><td>Frequency Use:</td><td>Frequently when the user wants to view an appointment.</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Enter the title of the appointment.</td><td>System returns the appointment with the entered title.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>1. Appointment with the title searched for by the user does not exist.</td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

Use Case ID:	UC 16
Use Case Name:	Add appointment
Description:	User adds an appointment to their personal activity assistant.

<table>
<tr><td>Actors:</td><td>User</td></tr>
<tr><td>Pre-Conditions:</td><td>1.User opens the web application.
2.User has a Personal Activity Assistant account.
3.User is logged into their account.</td></tr>
<tr><td>Post-Conditions:</td><td>User adds an appointment to their personal activity assistant.</td></tr>
<tr><td>Frequency Use:</td><td>Frequently when the user wants to add an appointment.</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Enter appointment title.</td><td>System creates an appointment with the entered title.</td></tr>
<tr><td>2.</td><td>Enter appointment date and time.</td><td>System checks if the date and time of the appointment are unique and adds them to the appointment.</td></tr>
<tr><td>3.</td><td>Enter a description of the appointment.</td><td>System adds the description to the appointment.</td></tr>
</table>
</td></tr>
</table>

Variations:	1. The date and time of the appointment overlap with that of another appointment.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 17
Use Case Name:	Update appointment
Description:	User updates an appointment in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User updates an appointment in their personal activity assistant.
Frequency Use:	Frequently when the user wants to make changes to an appointment.
Flow of Events:	

	Actor Action	System Response
1.	Enter the title of the appointment.	System searches for an appointment with the entered title and returns it.
2.	Update the appointment.	System checks if the date and time of the appointment are unique and updates the appointment.

Variations:	1. Appointment with the title searched for by the user does not exist. 2. The date and time of the appointment overlap with that of another appointment.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 18

Use Case Name:	Delete appointment
Description:	User deletes an appointment from their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User deletes an appointment from their personal activity assistant.
Frequency Use:	Frequently when the user wants to delete an appointment.
Flow of Events:	

	Actor Action	System Response
1.	Enter the title of the appointment.	System searches for an appointment with the entered title and returns it.
2.	Delete the appointment.	System deletes the appointment.

Variations:	1. Appointment with the title searched for by the user does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 19
Use Case Name:	View expense
Description:	User views an expense in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User views an expense in their personal activity assistant.
Frequency Use:	Frequently when the user wants to view an expense.

<table>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Enter expense category.</td><td>System returns the expenses in the entered category.</td></tr>
<tr><td>2.</td><td>Enter expense date and time.</td><td>System return the expense with the date and time.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td>1. Expense with the category entered does not exist.
2. Expense with the date and time entered does not exist.</td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

Use Case ID:	UC 20
Use Case Name:	Add expense
Description:	User adds an expense to their personal activity assistant.

<table>
<tr><td>Actors:</td><td>User</td></tr>
<tr><td>Pre-Conditions:</td><td>1.User opens the web application.
2.User has a Personal Activity Assistant account.
3.User is logged into their account.</td></tr>
<tr><td>Post-Conditions:</td><td>User adds an expense to their personal activity assistant.</td></tr>
<tr><td>Frequency Use:</td><td>Frequently when the user wants to add an expense.</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Enter expense date and time.</td><td>System creates an expense with the date and time.</td></tr>
<tr><td>2.</td><td>Enter the expense category.</td><td>System adds the category to the expense.</td></tr>
<tr><td>3.</td><td>Enter the expense description.</td><td>System adds the description to the expense.</td></tr>
<tr><td>4.</td><td>Enter the expense amount.</td><td>System adds the amount</td></tr>
</table>
</td></tr>
</table>

---	---	---
		to the expense.
Variations:		
Notes and Issues:		
Developer Notes:		

Use Case ID:	UC 21
Use Case Name:	Update expense
Description:	User updates an expense in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User updates an expense in their personal activity assistant.
Frequency Use:	Frequently when the user wants to update an expense.
Flow of Events:	

	Actor Action	System Response
1.	Enter the expense category.	System returns the expenses in that category.
2.	Enter the expense date and time.	System returns the expense with the date and time.
3.	Update the expense.	System updates the expense.

Variations:	1. Expense with the category entered does not exist. 2. Expense with the date and time entered does not exist.
Notes and Issues:	
Developer Notes:	

Use Case ID:	UC 22

Use Case Name:	Delete expense
Description:	User deletes an expense in their personal activity assistant.

Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.
Post-Conditions:	User deletes an expense in their personal activity assistant.
Frequency Use:	Frequently when the user wants to delete an expense.
Flow of Events:	

	Actor Action	System Response
1.	Enter expense category.	System returns the expenses in that category.
2.	Enter the expense date and time.	System returns the expense with the date and time.
3.	Delete the expense.	System deletes the expense.

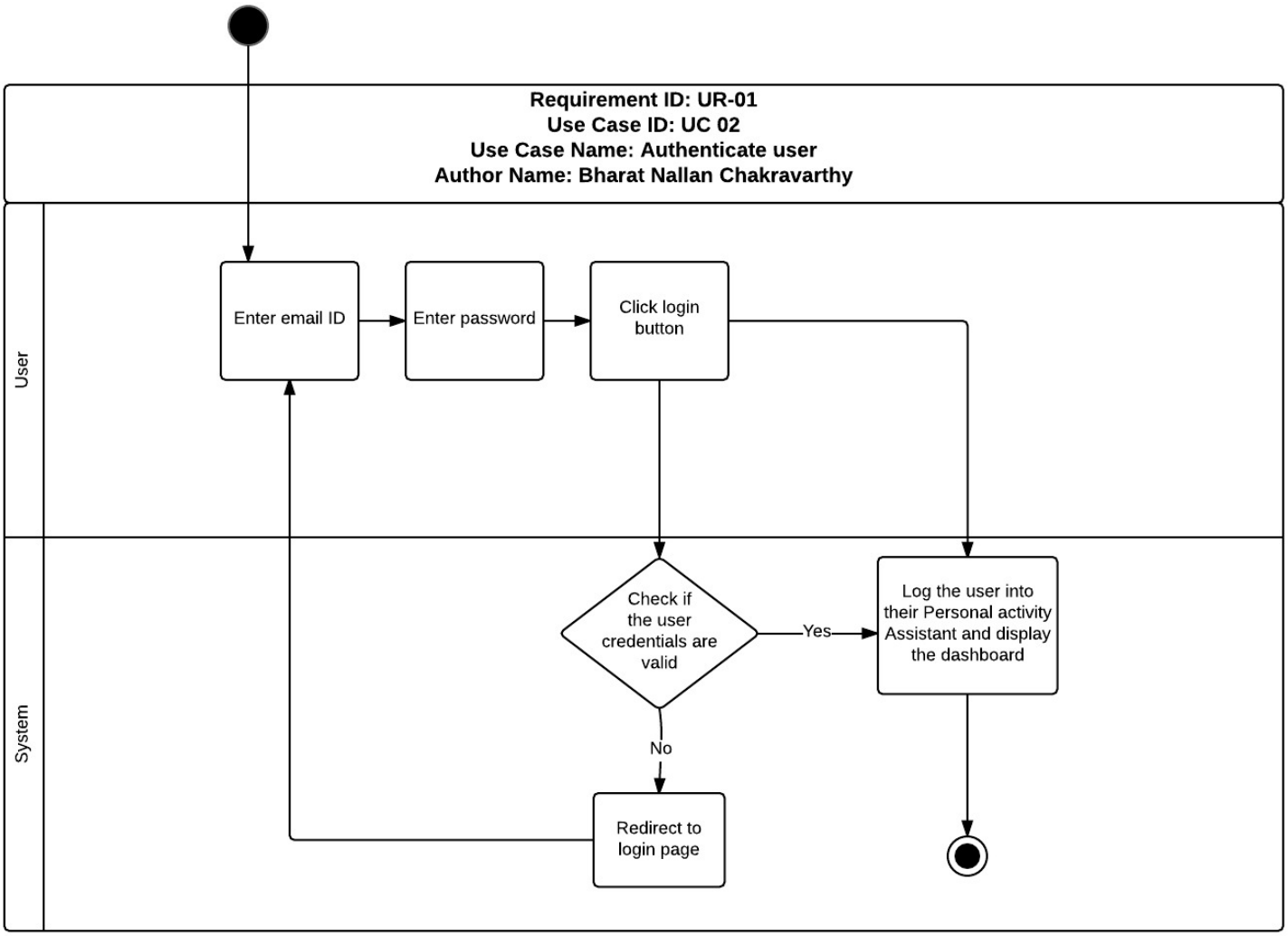
Variations:	1. Expense with the category entered does not exist. 2. Expense with the description entered does not exist.
Notes and Issues:	
Developer Notes:	

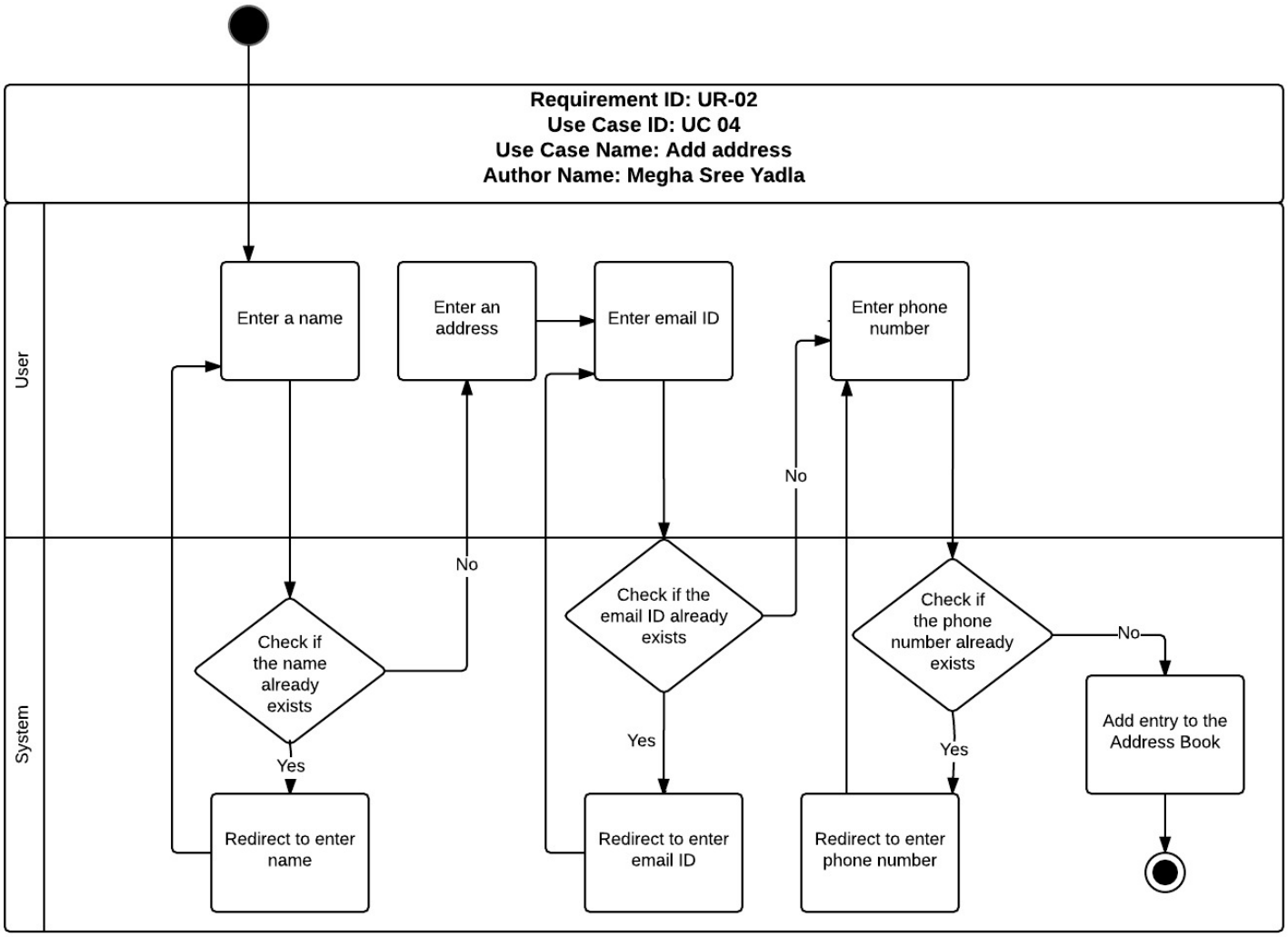
Use Case ID:	UC 23
Use Case Name:	Undo expense
Description:	User can undo the last deleted expense in their personal activity assistant.

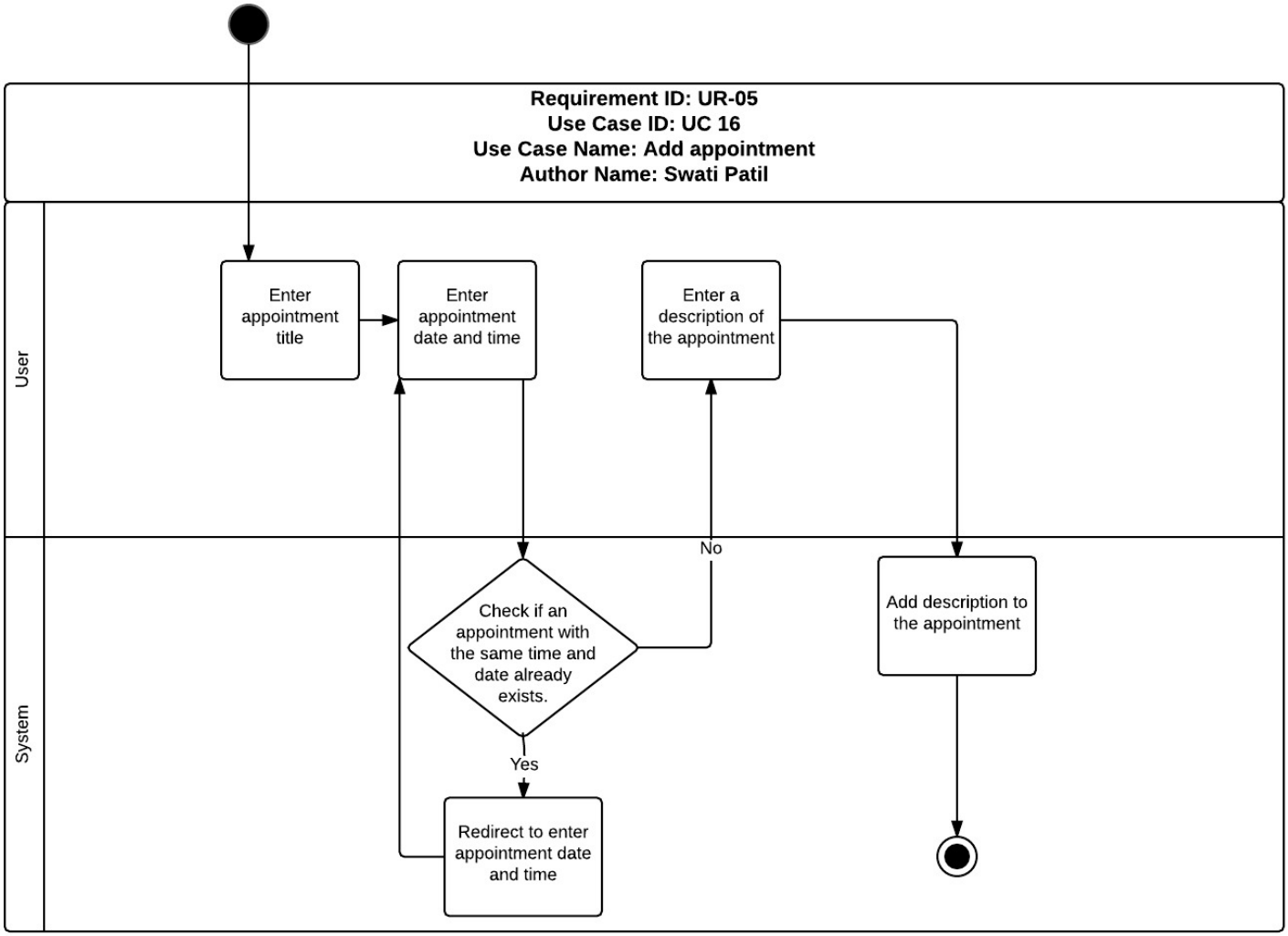
Actors:	User
Pre-Conditions:	1.User opens the web application. 2.User has a Personal Activity Assistant account. 3.User is logged into their account.

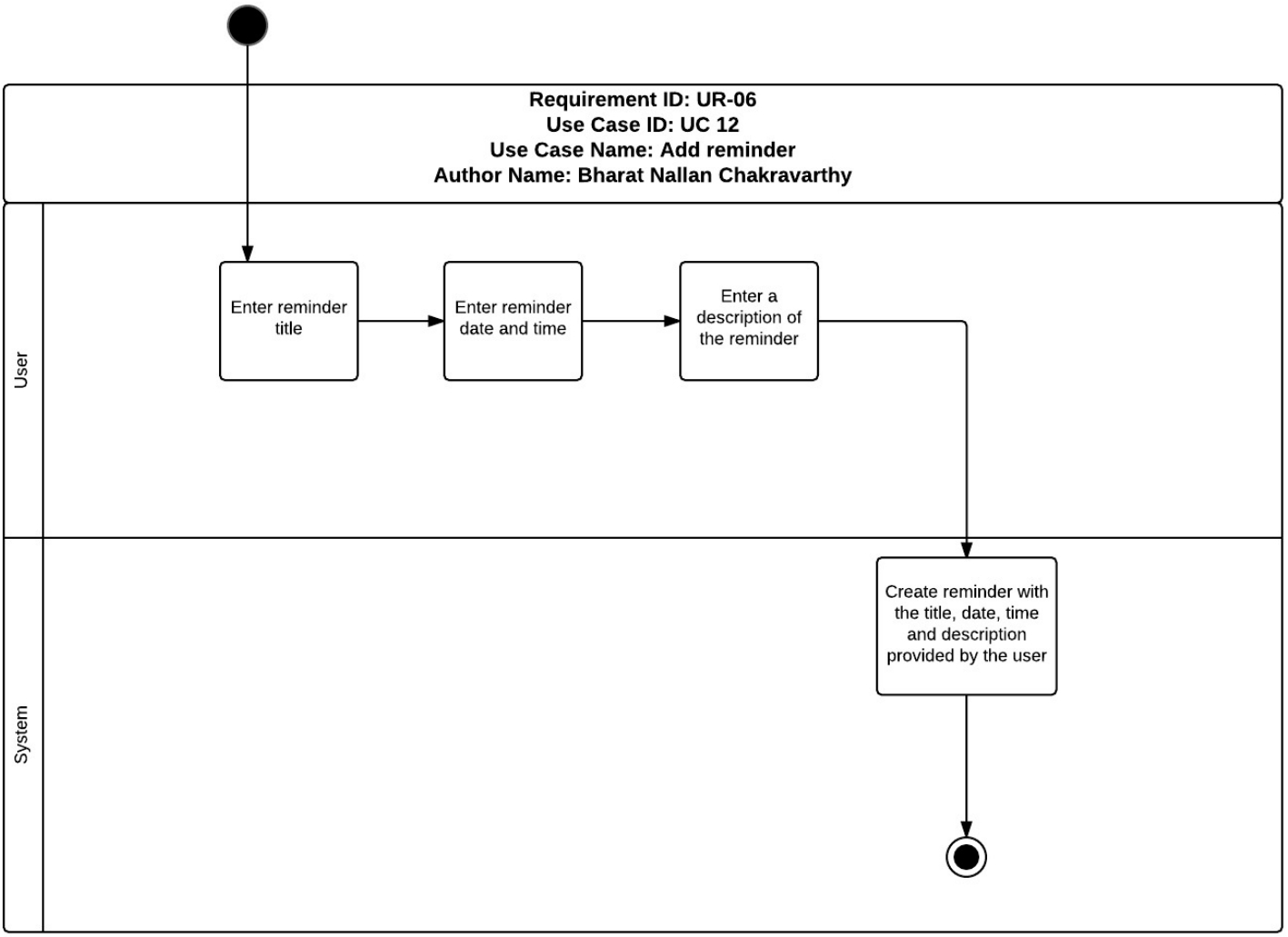
<table>
<tr><td>Post-Conditions:</td><td>User can undo the last deleted expense in their personal activity assistant.</td></tr>
<tr><td>Frequency Use:</td><td>Frequently when the user wants to undo a deleted expense.</td></tr>
<tr><td>Flow of Events:</td><td>
<table>
<tr><th></th><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.</td><td>Undo</td><td>System restores the last deleted expense.</td></tr>
</table>
</td></tr>
<tr><td>Variations:</td><td></td></tr>
<tr><td>Notes and Issues:</td><td></td></tr>
<tr><td>Developer Notes:</td><td></td></tr>
</table>

4. Activity Diagram




Requirement ID: UR-02
Use Case ID: UC 04
Use Case Name: Add address
Author Name: Megha Sree Yadla
User
System
Enter a name
Enter an address
Enter email ID
Enter phone number
Check if the name already exists
No
Yes
Redirect to enter name
Check if the email ID already exists
No
Yes
Redirect to enter email ID
Check if the phone number already exists
No
Yes
Redirect to enter phone number
Add entry to the Address Book


Requirement ID: UR-05
Use Case ID: UC 16
Use Case Name: Add appointment
Author Name: Swati Patil
User
System
Enter appointment title
Enter appointment date and time
Enter a description of the appointment
Check if an appointment with the same time and date already exists.
No
Yes
Redirect to enter appointment date and time
Add description to the appointment



5. Data Storage

The following are the tables used:

5.1 User Credentials Table

This table contains the login credentials and details of the user.

Example User Credentials Table:

ID	Name	Email ID	Password Hash
01	Bharat NC	bharatnc@gmail.com	XXXXX
02	Swati Patil	swatpat7@gmail.com	XXXXXXX
03	Megha Sree Yadla	meghuyadla@gmail.com	XXX

5.2 Address Table

This table contains the address and details of a contact that the user enters.

Example Address Table:

Name	Address	Email ID	Phone No.
Bharat NC	University Heights, Boulder, CO	bharatnc@gmail.com	(720) 662-8853
Swati Patil	1707 22nd Street, Boulder, CO	swatpat7@gmail.com	(720) 639-0248
Megha Sree Yadla	3161 Madison Avenue, Boulder, CO	meghuyadla@gmail.com	(720) 876-8599

5.3 Expenses Table

This table contains the details of the expenses of the user.

Example Expenses Table:

S.No.	Date	Time	Category	Description	Amount
01	02/27/2016	10:00	Grocery	Milk, Fruits and vegetables	$35
02	02/28/2016	12:30	Food	Pizza	$20

5.4 Notes Table

This table contains the notes saved by the user.

Example Notes Table:

Date	Time	Title	Note
02/27/2016	22:00	Username amazon	activity_assistant
02/28/2016	09:00	Office hours	Tuesday: 1-1:45 pm Wednesday: 11:30-12:30

5.5 Appointments Table

This table contains the details of the appointments set by the user.

Example Appointments Table:

Appointment Title	Appointment Date and Time	Appointment Description
Dentist appointment	02/29/2016- 3:00 pm	Dentist follow up appointment at wardenburg
Career services	03/02/2016- 10:00 am	Career services session at C4C

5.6 Reminders Table

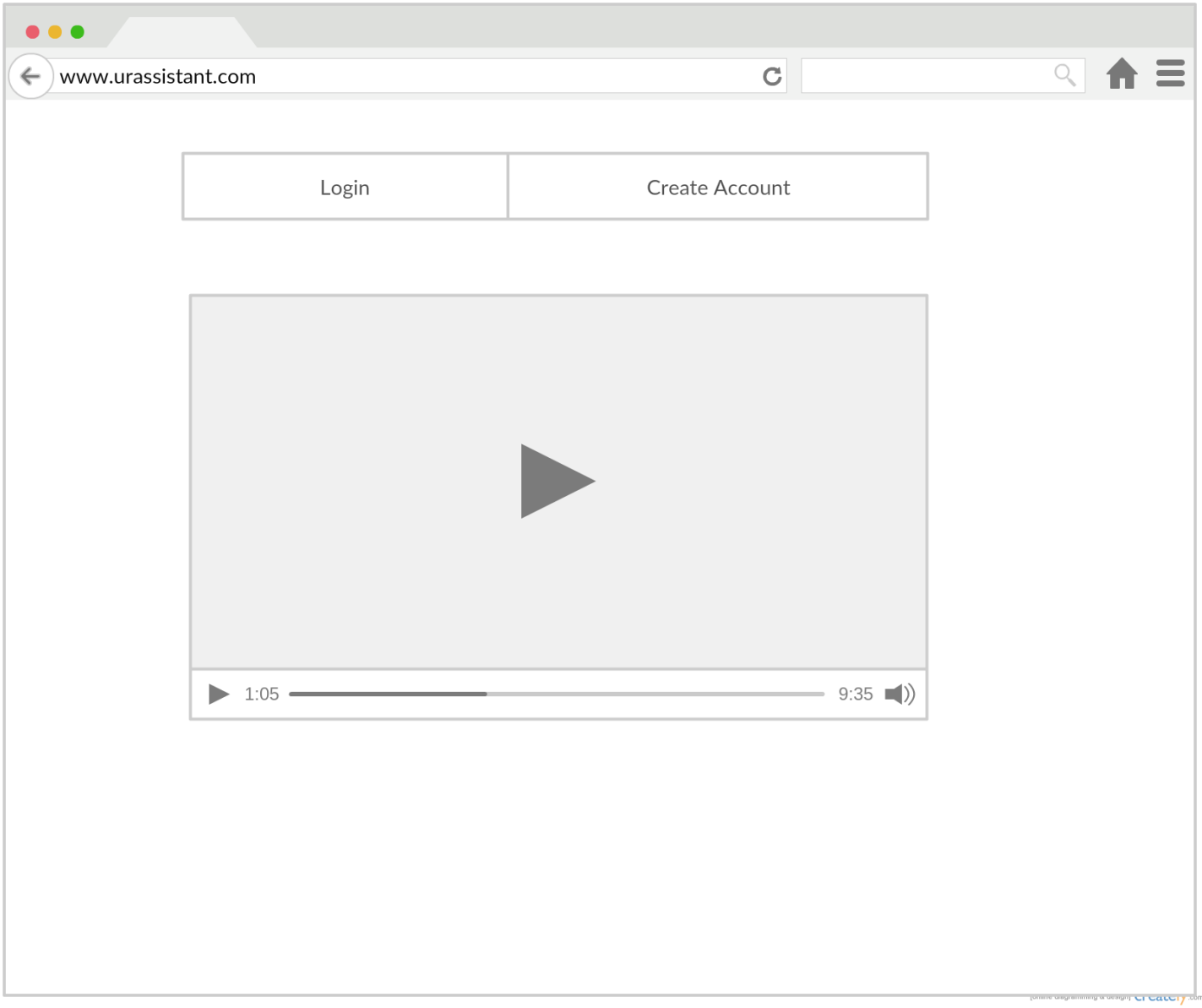
This table contains the details of the reminders set by the user.

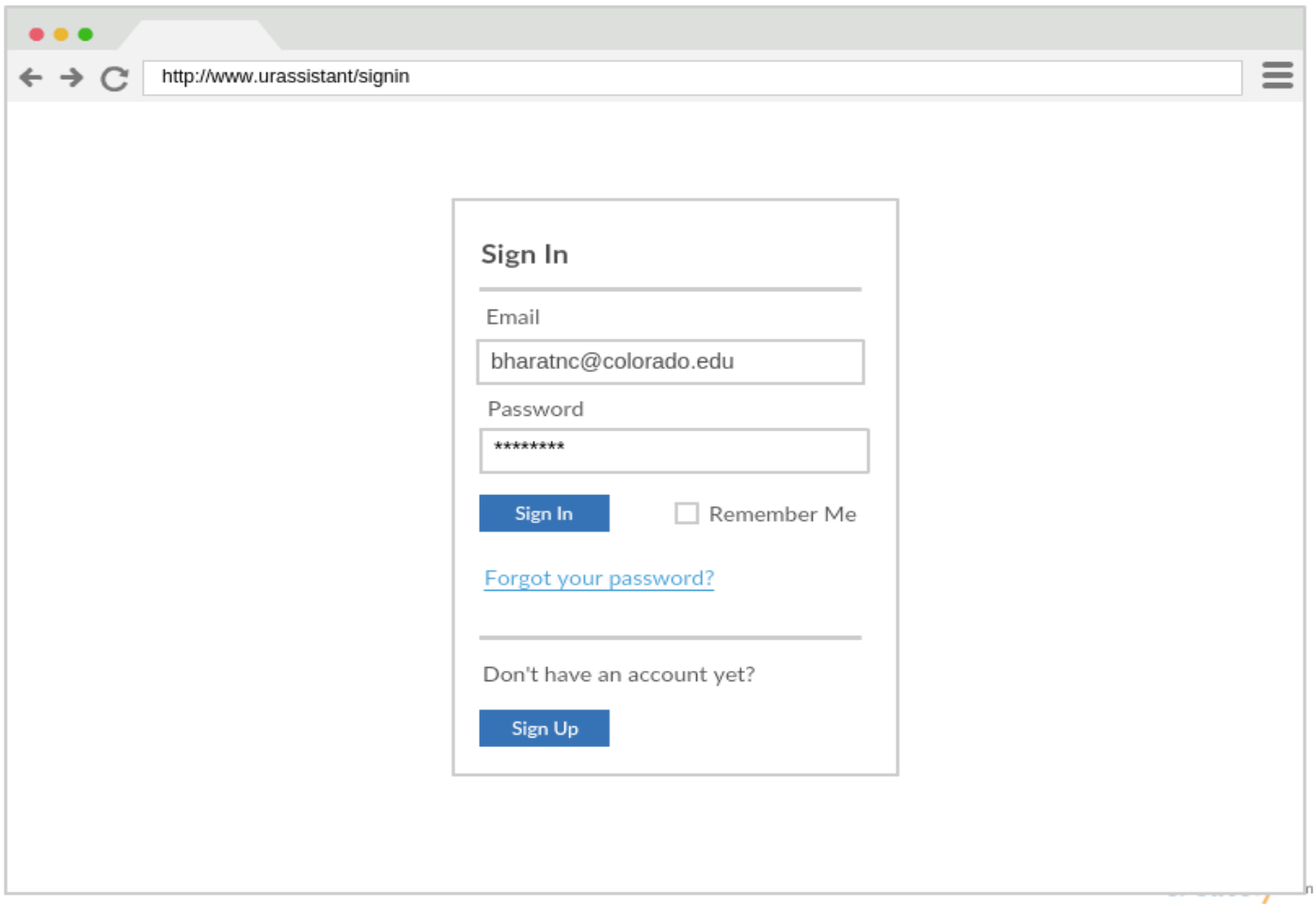
Example Reminders Table:

Reminder Title	Reminder Date and Time	Reminder Description
Email	03/04/2016- 12:00 pm	Send an email to Swati Patil about project meeting.
Rent	04/01/2016- 09:00 am	Pay rent to Sherry.

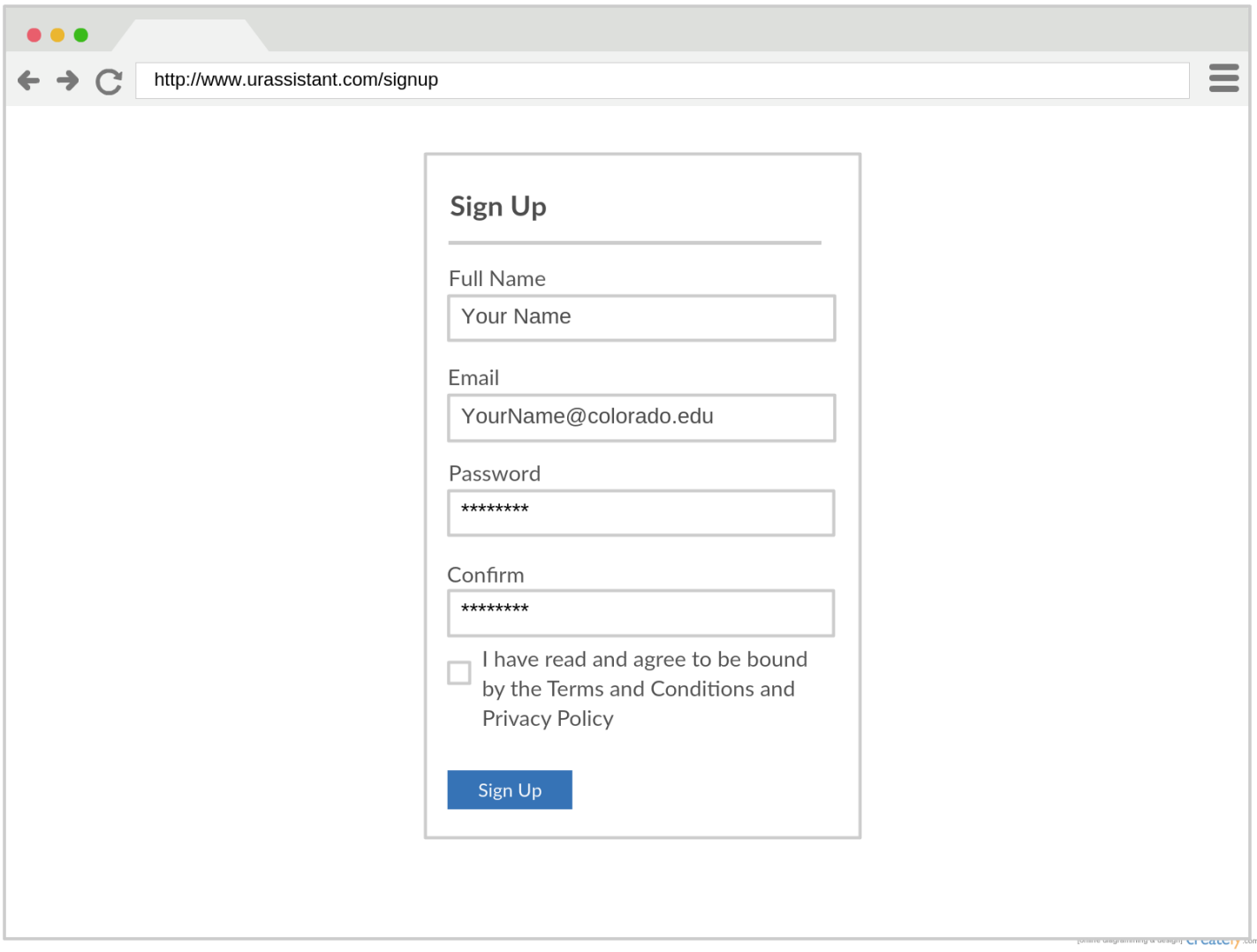
6. User InterFace Mockup

The returning users login page:

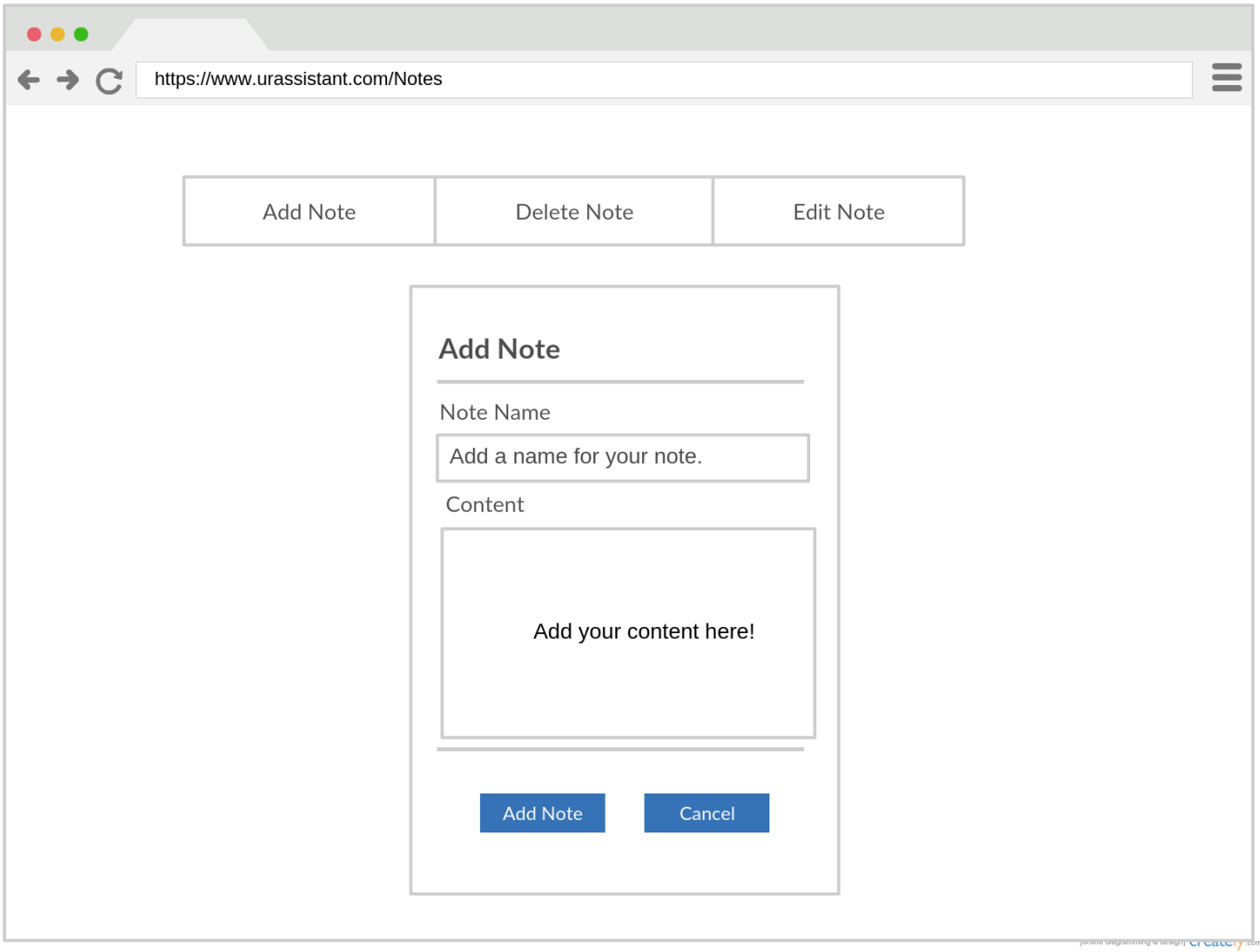



http://www.urassistant/signin
Sign In
Email
bharatnc@colorado.edu
Password
****
Sign In
Remember Me
Forgot your password?
Don't have an account yet?
Sign Up

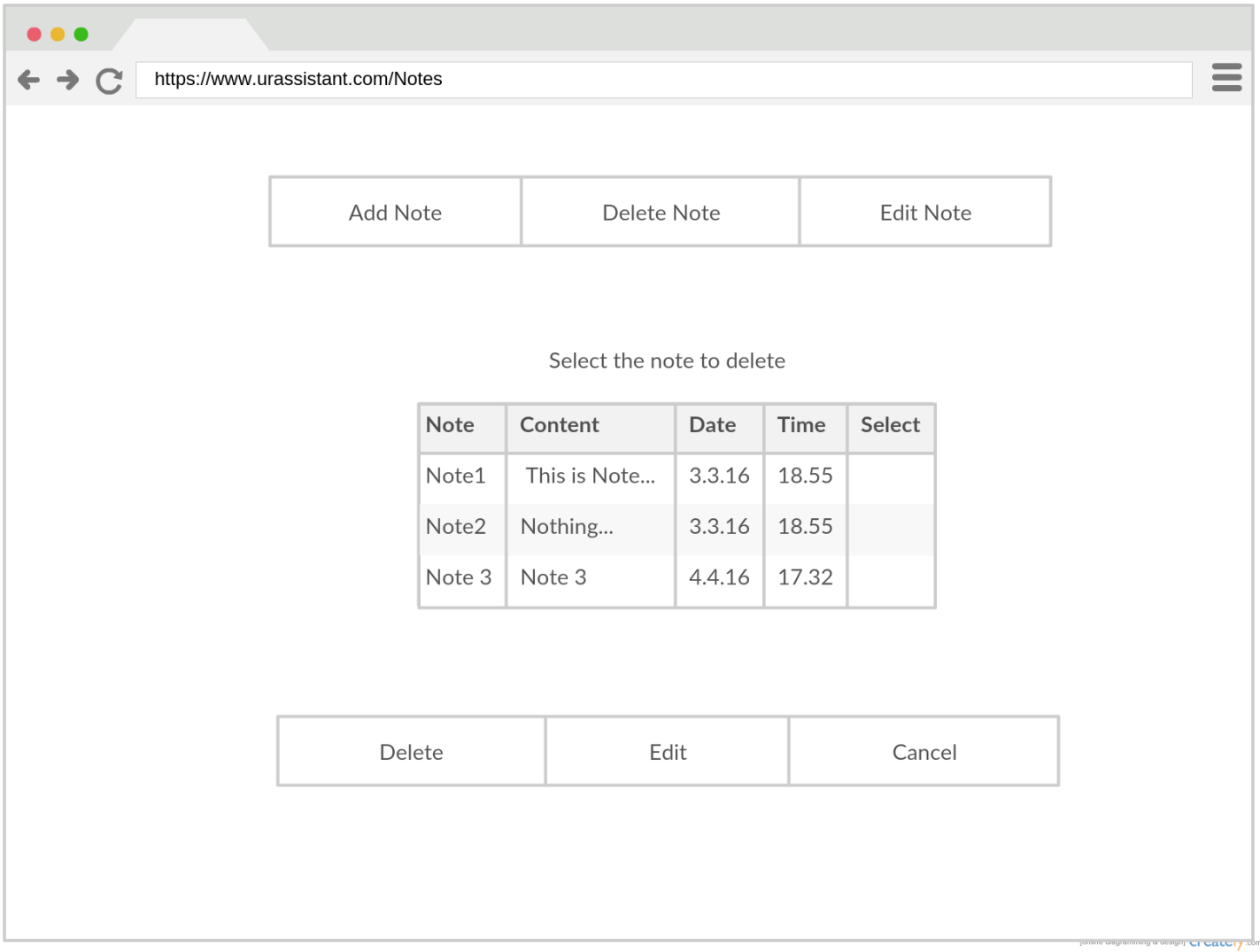
The create account page for new users:



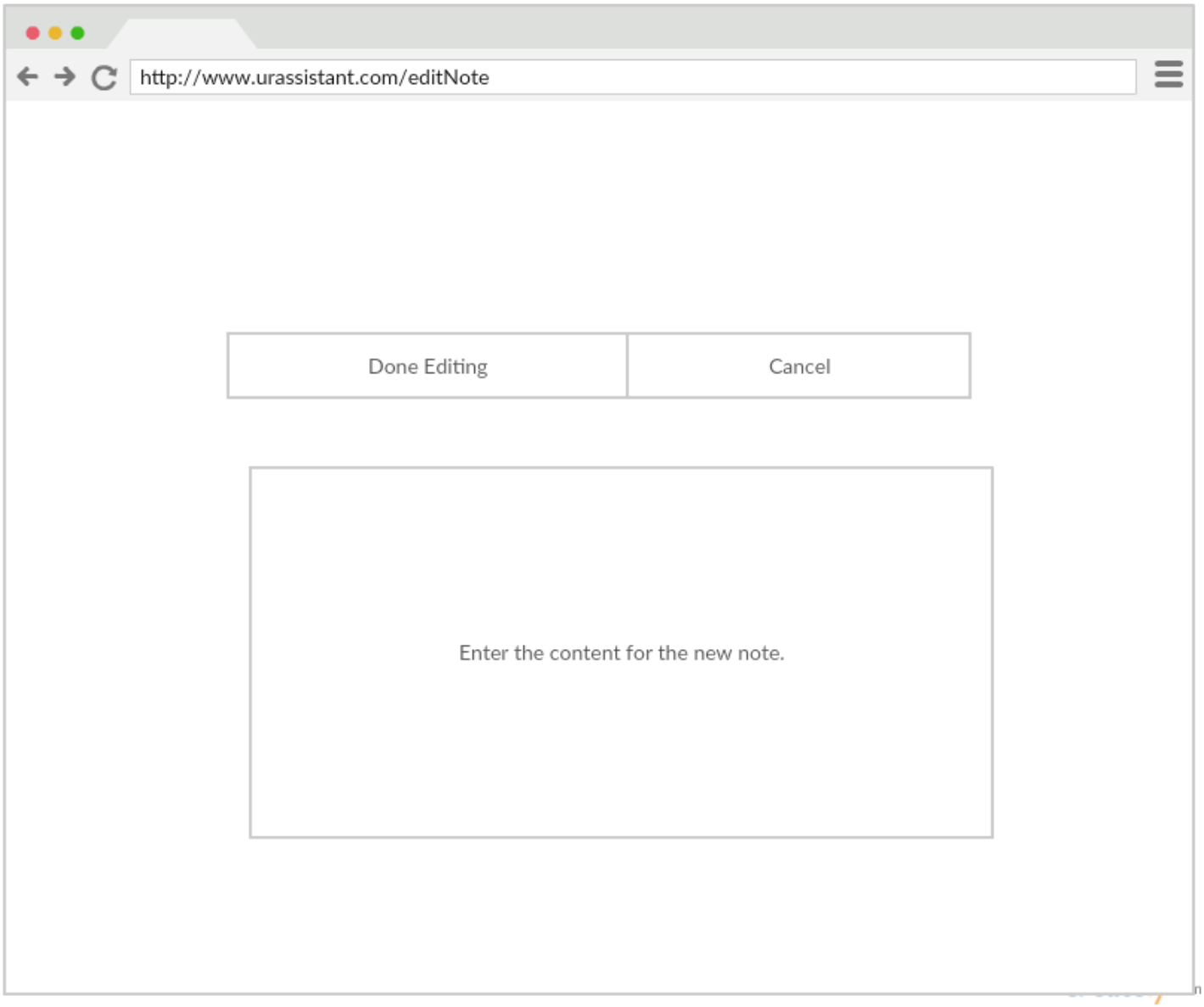
Adding Notes:



Deleting Notes:



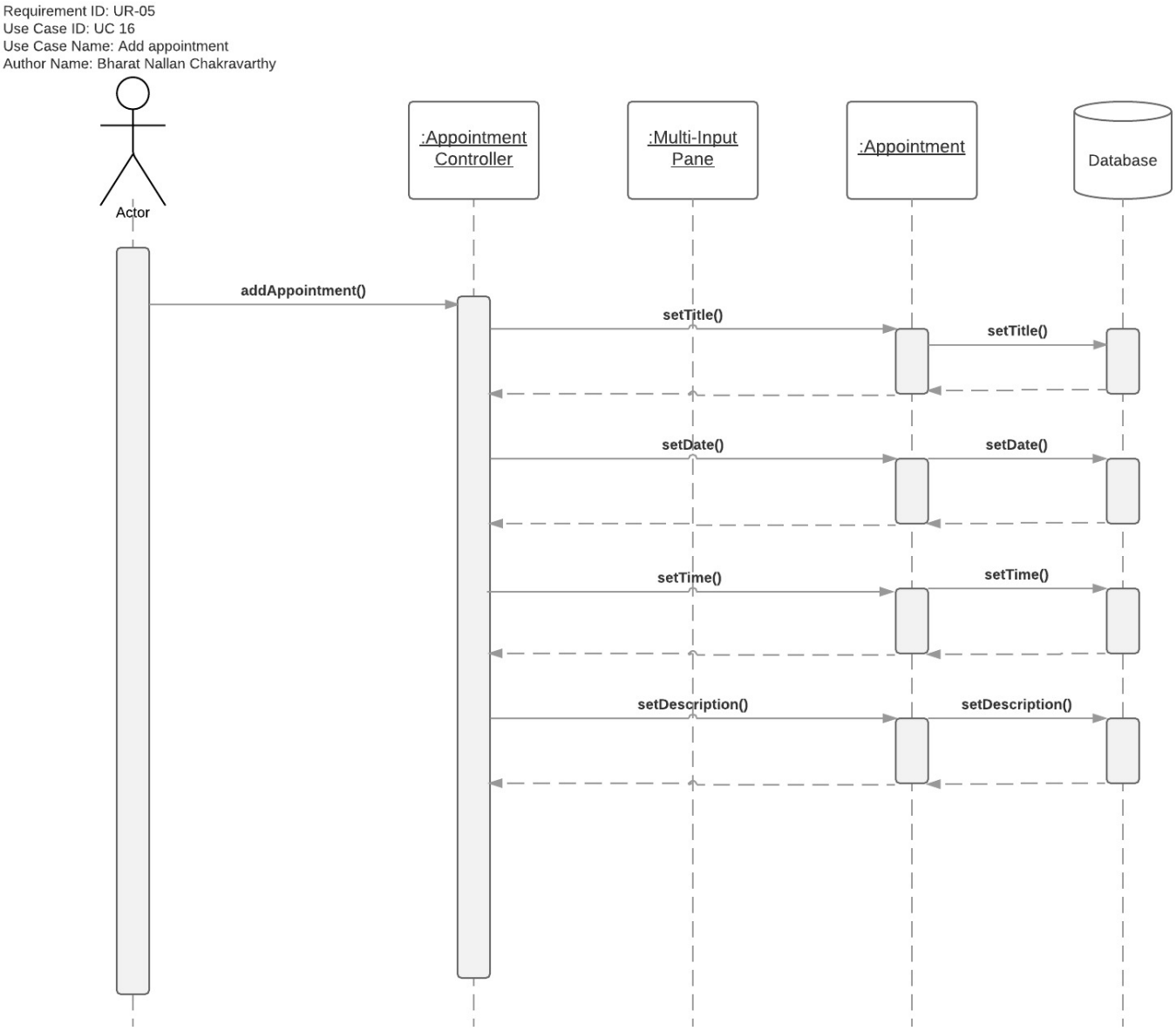
Editing a Note:



7. User Interactions

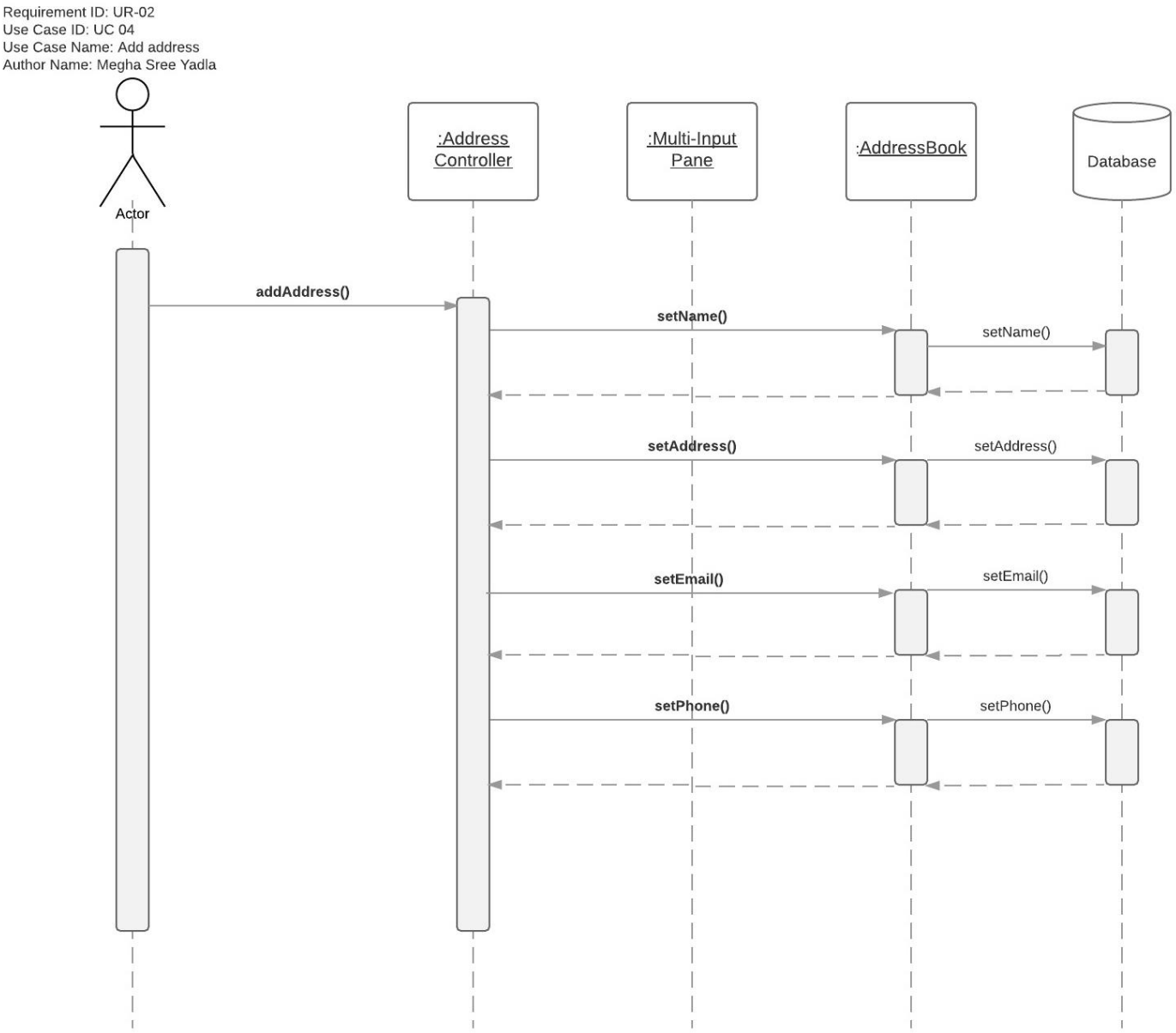
7.1 Sequence Diagram: User adding an appointment

Use Case Name: Add appointment



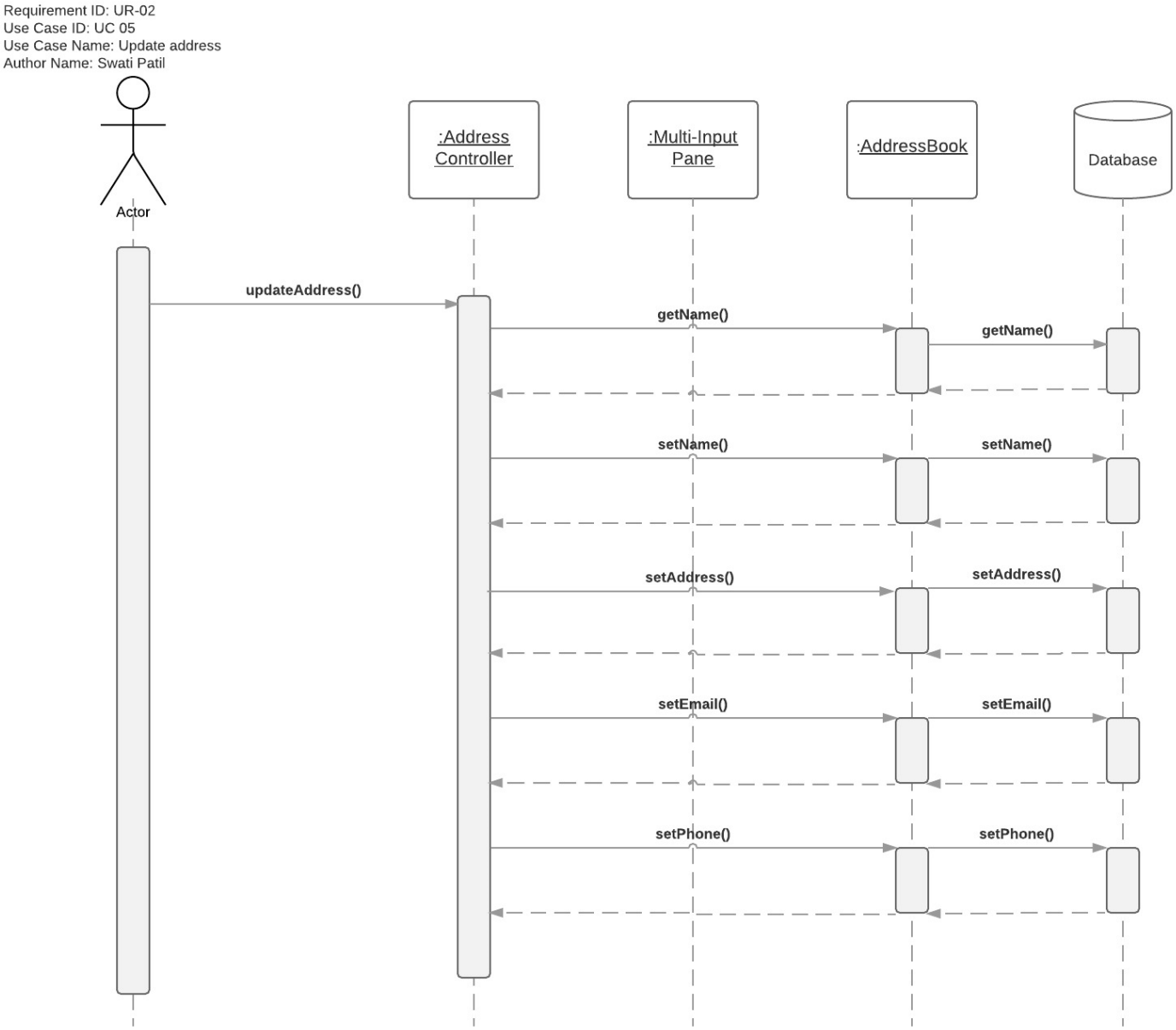
7.2 Sequence Diagram: User adding an address

Use Case Name: Add address



7.3 Sequence Diagram: User updating an address

Use Case Name: Update address



8. Class Diagram