

Biletixx

Requirements Specification and Analysis

Final Version

28.10.2021

Sezin Eliçalışkan
Nazlı Zeynep Uysal
Betül Akgül
Can Saygılı
Altan Öztürk

Prepared for
SE301 Software Engineering



IŞIK UNIVERSITY
COMPUTER
SCIENCE AND
ENGINEERING

Table of Contents

1.	Introduction	1
1.1.	Purpose of the System	1
1.2.	Scope of the System	1
1.3.	Objectives and Success Criteria of the Project.....	1
1.4.	Definitions, Acronyms, and Abbreviations	1
1.5.	Overview	2
2.	Current System	2
3.	Proposed System.....	2
3.1.	Overview	2
3.2.	Functional Requirements.....	2
3.3.	Nonfunctional Requirements.....	3
	Usability.....	3
	Reliability	3
	Performance.....	4
	Supportability	4
	Implementation.....	4
	Interface	4
	Packaging.....	4
	Legal	4
3.4.	System Models	5
	Scenarios.....	5
	Use case model	17
	Object model.....	41
	Dynamic model.....	42
	User interface—navigational paths and screen mock-ups.....	53
3.5.	Project Schedule	59
4.	Glossary	60
5.	References	60

REQUIREMENTS ANALYSIS DOCUMENT[1]

1. Introduction

The following section provides an introductory overview of the RAD (Requirement Analysis Document) for the “Biletixx “ ticket reservation system. The purpose of this document is to summarize Biletixx system in an easy, understandable way for the clients. This document includes Functional and non functional requirements, system models, glossary and references. Purpose, scope, objectives, success criteria and definitions are shown below in this section of the document.

1.1. Purpose of the System

The main purpose of the system is to provide our clients with a functioning, quality and easy to use ticket purchasing system. Our system helps users easily book a ticket without going to a ticket office. Doesn't matter if you're making a last minute purchase or you just don't want to leave your house, our users can easily search for an event that they want to go to and book their ticket in few simple clicks online from the comfort of their own home.

1.2. Scope of the System

Biletixx is a ticket reservation system where users can purchase event tickets online. In our system there are four roles such as, visitors, users, event holders and admin roles. Visitors could create their regular user or event holder account and become a Biletixx user. Visitors and users can search and display available events, but since being an user is required to purchase tickets, only users can buy tickets, see their purchased tickets and also manage their profile. Event holders can manage the events and admins can manage event holders and users accounts all together.

1.3. Objectives and Success Criteria of the Project

Our objective is to present our users with a high quality, easy to use ticket purchasing website. And our succes criteria depends on the meeting the following set of objectives:

- Project meets the functional requirements.
- Project meets the non-functional requirements.
- Project documentations are delivered on time.
- Project provides the items described on the scope.
- Positive feedbacks.

1.4. Definitions, Acronyms, and Abbreviations

User: Customer who signs up on the website.

<Biletixx>

Visitor: User who can browse the website without signing up.

Event Holder: Person or an organization who holds the event.

Admin: The controller of the system.

RAD: Requirement Analysis Document

1.5. Overview

The document is about system information, functional requirements, nonfunctional requirements, system models that includes scenarios, use case diagrams, class diagrams and sequence diagrams. All of them are explained in this document. High-level functionalities of the system is described in functional requirements, user-level functionalities of the system is described in nonfunctional requirements.

2. Current System

Biletixx is a website that you can buy tickets online. The website has users, visitors, admins and event holders. Visitors can view the website as they wish without logging in. They can access all events and performances and learn all information about them. Users can save their information such as address, payment information, personal informations etc. In order to purchase tickets which is the most important part, it is necessary to become a user. Event holders main task are adding new events and deleting events. Event holders can also display personal information. Admin is the most authorized person on the website. His powers are different everyone else. They control everything including eventholders for example they can view event holder information, user information delete user accounts and delete event holder accounts.

3. Proposed System

Documents the requirements elicitation and the analysis model of the new system

3.1. Overview

In this section, we explain the functional requirements and the description of the actors who will use the functions. There are four actors in our system. These are Visitor, User, EventHolder and Admin.

Visitor; S/he can visit our website and browse the appropriate events. These functions do not require membership. And finally, you can register on our site.

Users; They can access everything visitors have access to and also users can log out, change their password, and buy tickets. S/he can see how much the tickets will cost. It can register its own address in the system. You can view and make changes to this address. S/he can view purchased tickets. S/he can update her/his personal information.

Eventholder; S/he can log out of his accounts just like users. S/he can change his password. S/he can view her/his personal information. Unlike users, they can add and delete events in the system.

Finally Admin. Admin can see eventholders and users information. S/he can delete user account and also can delete eventholder account. The logged in user must be an administrator.

3.2. Functional Requirements

- 1) **Visitor** should be able to create account.
- 2) **Visitor** should be able to create event holder account.
- 3) **Visitor/ User** should be able to search events that are available.
- 4) **Visitor/ User** should be able to display event from the available events list.
- 5) **Visitor** should be able to forget password.
- 6) **User/ EventHolder** must be able to log in.
- 7) **User/ EventHolder** should be able to log out.
- 8) **User/ EventHolder** should be able to change password.
- 9) **User** should be able to buy ticket
- 10) **User** should be able to pay ticket cost.
- 11) **User** should be able to create address.
- 12) **User** should be able to display addresses they created.
- 13) **User** should be able to update address they created.
- 14) **User** should be able to delete address they created.
- 15) **User** should be able to display tickets they have bought.
- 16) **User** should be able to update personal information.
- 17) **User/ EventHolder** should be able to display personal information .
- 18) **EventHolder** should be able to add new events.
- 19) **EventHolder/Admin** should be able to delete events.
- 20) **Admin** should be able to viewEventHolderInformation.
- 21) **Admin** should be able to viewUserInformation.
- 22) **Admin** should be able to delete user accounts.
- 23) **Admin** should be able to delete event holder accounts.

3.3. Nonfunctional Requirements

Usability

- 1) User should be able to log out from their account at most in 2 clicks.

Reliability

- 1) The system should continue functioning even if the user enters wrong password.
(Robustness)
- 2) The system must be running 98% of the time.

<Biletix>

Performance

- 1) User should be able to login 3 seconds after clicking login. (Respond time)
- 2) User should be provided feedback after buying ticket in less than 3 seconds. (Respond time)
- 3) The system should not be down more than 10 mins per week. (Availability)

Supportability

- 1) EventHolder should be able to add new events without modifications to the existing system.

Implementation

- 1) All related software will be written using Python, HTML, CSS, JavaScript.

Interface

Packaging

- 1) System must be available on desktop and perform properly on a browser.

Legal

- 1) The system is free to use.

3.4. System Models

Scenarios

Scenario name [newBiletixAccount](#)

Participating Actor Instances [Jack:Visitor,](#)

Flow of events

1. Jack sees a post about a music event while checking his social media. He decides to buy a ticket for this event. He opens his laptop and enters biletix website from his web browser. He sees the main page of the website.
2. He clicks the “CreateAccount” button on the main page of the website. He clicks create account as an event holder button He sees a blank form. He fills his name, surname, gender, birthday, phone number to the form. He finishes filling the form and clicks okay button at the end of the form.
3. Jack receives feedback about account creation conformation.

Scenario name [newBiletixEventHolderAccount](#)

Participating Actor Instances [James:EventHolder,](#)

Flow of events

1. Jack is an event organitazot and he wants to create a biletix account to organize a new event and sell tickets. He enters the biletix website.
2. He clicks the “CreateAccount” button on the main page of the website. He clicks create account as an event holder button. He sees a blank form. He fills his name, surname, gender, birthday, phone number to the form. He finishes filling the form and clicks okay button at the end of the form.
3. James receives feedback about account creation conformation.

Scenario name lookingForNewEvent

Participating Actor Instances Sarah:User,Alice:Visitor

Flow of events

1. Sarah wants to attend an event for the weekend. She opens her laptop and enters biletix website from his web browser. She gets directed to the main page of the website by the system.
2. She chooses the event category, date, location from the menu on the main page. She clicks the "Search Event" button. She sees a list of events according to the information entered by her.
3. She sees a Music event that she likes and clicks display event button to buy a ticker for it.

Scenario name viewAnEvent

Participating Actor Instances Jack:Visitor,Sarah:User

Flow of events

1. Sarah wants to attend an event for the weekend. She opens her laptop and enters biletix website from his web browser. She gets directed to the main page of the website by the system.
2. She chooses the event category, date, location from the menu on the main page. She clicks the "Search Event" button. She sees a list of events according to the information entered by her.
3. She sees a Music event that she likes and clicks "Display Event" button to buy a ticket for it. She sees the information of the event she choose.

Scenario name [loginToAccount](#)

Participating Actor Instances [Sarah:Visitor](#)

Flow of events 1. Sarah wants to go to a theater event. She opens her laptop and enters biletix website from her web browser. She gets directed to the main page of the website by the system.

2. She clicks "[Log In](#)" button to login to her account and buy a ticket for the event.

Scenario name [logoutFromAccount](#)

Participating Actor Instances [Sarah:User](#)

Flow of events 1. Sarah wants to go to a theater event. She opens her laptop and enters biletix website from her web browser. She gets directed to the main page of the website by the system.

2. She clicks "[Log In](#)" button to login to her account and buy a ticket for the event. After she has bought the ticket she goes to her account page on the website and clicks "[Log Out](#)" but and logouts from the website.

Scenario name *forgetUserPassword*

Participating Actor Instances Jack:Visitor,James:EventHolder

Flow of events

1. Jack/James wants to check his biletix account. Jack/James tries to login to the website but he realizes forget his password.
2. He opens the login page. He clicks the “Forget Password” button on the login page of the website. He sees reset password form. He fills his e-mail. He finishes filling the form and clicks send button at the end of the form.
3. Feedback is given by the website that the reset password link has been sent to Jack/Jame’s e-mail.
4. Jack/James logins his e-mail and opens the corresponding e-mail. He clicks The “Reset Password” button. He creates new password.

Scenario name *changeUserPassword*

Participating Actor Instances Jack:User,James:EventHolder

Flow of events

1. Jack/James thinks his password was insecure and he wants to change his password.
2. He opens the “My Account” menu and clicks my profile.
3. He fills the change password form and clicks change password button.
4. Website gives feedback about password change status.

Scenario name *buyingEventTicket*

Participating Actor Instances Jack:User

Flow of events

1. Jack thinks he hadn't been to the theater for a long time and decides to buy a theater ticket.
2. He opens the main page of the website and he clicks theater button at the top of the page.
3. He selects the one he likes from the available shows and clicks on it.
4. He views the event rules and ticket prices he fills the fields in the buy ticket form and he clicks the next button.
5. He chooses his seat and ticket delivery option and he clicks the next button.
6. He fills up the payment information and clicks buy button.
7. He sees the feedback that the ticket was purchased successfully or not.

Scenario name payTicketCost

Participating Actor Instances John:User

Flow of events

1. John wants to buy a ticket for an event. He logs in to his biletix account.
2. He chooses the event category, date and location then searches events. He picks a music event he likes then displays the event.
3. He clicks the buy ticket button on the event page, he sees the price and clicks the "Pay Ticket".

Scenario name [createAddress](#)

Participating Actor Instances [John:User](#)

Flow of events

1. John just created a biletix account. He wanted to buy a ticket for an event He wants to attend. Before buying the ticket he realized he didn't add his address.
2. He clicks on the My Account button and then on the My Addresses button.
3. Then, he writes his address and presses the [create address](#) button. After that he clicks my addresses button again. He can see his newly created address on the list of addresses.

Scenario name [displayAddress](#)

Participating Actor Instances [John:User](#)

Flow of events

1. John logins to his biletix account. He wants to buy a ticket for an event he wants to attend. Before buying the ticket he wants to confirm that his address is correct.
2. He clicks the My Account button and then My Addresses button.
3. He sees his addresses being displayed on the website.

Scenario name updateAddress

Participating Actor Instances John:User

Flow of events 1. John moved from his old house to a new house. Because of that he wants to update his address information on the biletix website.

2. He logins to his account.He clicks on the My Account button and then clicks on the My Addresses button. He sees his addresses being displayed on the website.

4. He clicks the update my address button and updates his address.

Scenario Name: deletingAddress

Participating Actor Instances: Jack: User

Flow of Events:

1. Jack has recently moved to a new location and wants to delete his address he created before. He opens his computer and goes to the biletix website from his web browser. He sees the main page of the website.
 2. He then clicks on the "My Account" button that is on the main page. He chooses "My addresses" then he sees his current address on the page. He clicks the delete button underneath his address.
 3. Jack's receives a feedback that his address is succesfully deleted.
-

Scenario Name: viewingTickets

Participating Actor Instances: Jack: User

Flow of Events:

1. Jack has bought a ticket to a concert a week ago and wants to check his ticket again to check if everything is alright. He opens his computer and goes to the biletix page then logs in.
 2. He clicks "My account" button that is on the main page. Then he chooses "My tickets" from the option.
 3. He sees his tickets being listed on the page with its details.
-
-

Scenario Name: cancellingTickets

Participating Actor Instances: Sarah: User

Flow of Events:

1. Sarah got the news that her friend cannot make it to the concert. She decides she doesn't want to go alone so she wants to cancel her ticket. She goes on her computer and logs into the Biletix website.
 2. She clicks on "My account" then chooses "My tickets" from the screen.
 3. Her tickets are being displayed so she chooses the correct ticket and then clicks "Cancel my ticket"
 4. "Ticket has been cancelled succesfully" message appears on the screen and she gets a money refund on her account.
-

Scenario Name: displayPersonalInfo

Participating Actor Instances: Sarah: User ,Jack:EventHolder

Flow of Events:

1. Sarah/Jack created a biletix account and now wants to see if she/he entered her/his personal information correctly. So she/he logs into the Biletix website and clicks on the “My Account” button that is on the main page.
 2. Then she/he chooses “My Profile” from the option.
 3. Website directs her/him to My Profile page and here he/she sees her/his personal information, such as name, surname, date of birth.
-

Scenario Name: updatePersonalInfo

Participating Actor Instances: Sarah: User

Flow of Events:

1. Sarah notices she entered her personal information wrong when she created her account and she wants to change it. So she clicks “My Account” and chooses “My Profile” option.
 2. She gets directed to her profile page, here she can edit her personal information. She sees that she entered her date of birth incorrectly so she clicks on the “Edit Profile” button and changes her birth date.
 3. Then she clicks the save button and system saves her new information.
 4. She sees the “Profile updated” message on the screen.
-

Scenario name CreateEvent

Participating Actor Instances Jake: EventHolder

Flow of events

1. Jake wants to promote and sell tickets for the concert his company is organizing. Jake contacts with the 'Biletix' customer services and creates an 'Event Holder' account with his email address and a password.
2. Jake enters the 'Biletix' website and logs in with his email and password. He sees the main page of the website, he sees the 'Events' button and clicks on it. He clicks create event button on the events page.
3. Jake sees the event creation form and enters the Event Title, Event Pictures, Event Description, Organizer Name, Event Category, Tags, Location, Date and Time, Ticket Quantity and Ticket Price information of the concert he wanted to promote. He finishes the form and clicks on the 'Publish' button.
4. Jake sees the event he created on the events page of the website.

Scenario name DeleteAnEvent

Participating Actor Instances Jake: EventHolder

Flow of events

1. Jake wants to delete the concert event he created. He opens his laptop and enters biletix website from his web browser. Then he logs in with his email and password and directed to the main page.
2. Jake clicks the events button. He sees the event he wants to delete on the events list on the event page and clicks on the 'Edit' button next to it. Then he sees the event information on the edit page and Scrolls through the end of the page then he clicks on the 'Delete Event' button.
3. Jake gets a feedback about the deletion and he saws that the event is not shown on the events list.

Scenario name **DeleteAnEventAsAdmin**

Participating Actor Instances Jake: EventHolder

Flow of events

1. Taylor is the admin of the 'Biletix' website. Taylor once scrolls through the events to see if there is any inaccuracy or fault in the events. Taylor sees one event that keeps on postponing for a year. She tries to reach the organizer but their information is no longer in use. She decides to delete the event.
2. She turns back to the main page and clicks on the event she decided to delete. Then she clicks 'Delete Event' button next to the event and gets a feedback about the deletion

Scenario name **ViewUsers**

Participating Actor Instances **Taylor: Admin**

Flow of events

1. Taylor is the admin of the 'Biletix' website. She wants to see all the user's information.
2. She enters the website and logs in as an Admin. Then at the main page she clicks on the 'Users' button and sees all the users profile information.

Scenario name [ViewEventHolders](#)

Participating Actor Instances [Taylor: Admin](#)

Flow of events 1. Taylor is the admin of the 'Biletix' website. She wants to see all the event Holder informations.

2. She enters the website and logs in as an Admin. Then at the main page she clicks on the 'EventHolders' button and sees all the information about event holders.

Scenario name [DeleteUser](#)

Participating Actor Instances [Taylor: Admin](#)

Flow of events 1. Taylor is the admin of the 'Biletix' website. Taylor is contacted by a user who wants their account deleted. Taylor logs to the website and display the user list.

2. Taylor selects the mentioned user and clicks on the 'Delete' button. Then she gets a feedback about the deletion.

Scenario name DeleteEventHolder

Participating Actor Instances Taylor: Admin

Flow of events

1. Taylor is the admin of the 'Biletix' website. Taylor is contacted by a event Holder wh owants their account deleted. Taylor logins to the website and Clicks event holders buttons and display the event holders list.
2. Taylor selects the mentioned eventholder and clicks on the 'Delete' button. Then she gets a feedback about the deletion.

Use case model

Use case name CreateAccount

Participating Actors Initiated by Visitor

Flow of events

1. **Visitor** clicks the CreateAccount button on the main page of the website.
2. **System** shows user two options if he wants to create a user account or event holder account.
3. **Visitor** clicks user account option.

<Biletix>

4. **System** shows account creation form to the visitor.
5. **Visitor** enters the required user information.
6. **System** validates the information and notifies user that the account has been created.

Entry condition **Visitor** enters the website.

Exit conditions **Visitor** receives feedback about account creation.

Quality Requirements 1) System should give feedback in 5 seconds.

Use case name **CreateEventHolderAccount**

Participating Actors Initiated by **Visitor**

Flow of events

1. **Visitor** clicks the CreateAccount button on the main page of the website.
2. **System** shows user two options if he wants to create a user account or event holder account.
3. **Visitor** clicks event holder account option.
4. **System** shows account creation form to the visitor.
5. **Visitor** enters the required user information.
6. **System** validates the information and notifies user that the account has been created.

<Biletix>

Entry condition **Visitor** enters the website.

Exit conditions **Visitor** receives feedback about account creation.

Quality Requirements 1) System should give feedback in 5 seconds.

Use case name **SearchEvent**

Participating Actors Initiated by **Visitor,User**

Flow of events 1. **Visitor,User** chooses the event category, time and location from the menu on the main page of the website. Visitor clicks SearchEvent button on the main page.
2. **System** shows the event list to the visitor,user.

Entry condition **Visitor,User** enters the website.

Exit conditions **Visitor,User** is shown the event list by system.

Quality Requirements 1) System should show the event list in 5 seconds.

Use case name **DisplayEvent**

Participating Actors Initiated by **Visitor,User**

Flow of events

1. **Visitor,User** chooses the event category, time and location from the menu on the main page of the website. Visitor clicks SearchEvent button on the main page.
2. **System** shows the event list to the visitor,user.
3. **Visitor,User** clicks DisplayEvent button next to the event they choose.
4. **System** shows the information of the event visitor,user choose.

Entry condition **Visitor,User** is shown the event list.

Exit conditions **Visitor,User** is shown the information of the chosen event..

Quality Requirements 1) System should display the event information in 5 seconds.

Use case name **Login**

Participating Actors Initiated by **Visitor**

Flow of events 1. **Visitor** clicks the Log In button on the main page of the website.

2. **System** shows the log in form to the visitor.

3. **Visitor** enters the required information.

4. **System** validates the information and directs user to the main page.

Entry condition **Visitor** has an user account.

Visitor entered the website.

Exit conditions **Visitor** succesfully logins to website.

Use case name **LogOut**

Participating Actors Initiated by **User,EventHolder**

Flow of events 1. **User,EventHolder** clicks the LogOut button on the main page of the website.
2. **System** logs out the **User,EventHolder**.

Entry condition **User,EventHolder** is logged in.

Exit conditions **User,EventHolder** succesfully logouts from the website.

Use case name *ForgetPassword*

Participating Actor Initiated by **Visitor**

Flow of events 1. **Visitor** clicks ForgetPassword button on the login page of the website.
2. **System** shows the reset password form.
3. **Visitor** enters his e-mail.
4. **System** sends reset password link to **Visitor**'s e-mail.
5. **Visitor** clicks the reset password link and creates new password.

Entry condition **Visitor** is on the log in page.

Exit conditions **Visitor** creates new password.

Quality Requirements 1) System should create password in 5 seconds.

Use case name *ChangePassword*

Participating Actor Initiated by **User,EventHolder**

Flow of events

1. **User,EventHolder** clicks My Account button on the main page of the website.
2. **System** shows the My account page.
3. **User,EventHolder** clicks change password button.
4. **System** shows change password form to the **User,EventHolder**
5. **User,EventHolder** enters his current and new passwords.
6. **System** changes the password and gives feedback about it.

Entry condition **User,EventHolder** is logged in.

Exit conditions **User,EventHolder** changes his password.
System gives feedback

Quality Requirements 1) System should change password in 5 seconds.

Use case name *BuyTicket*

Participating Actor Initiated by User

Flow of events 1. **User** chooses the number of tickets on the event's page.
2. **System** confirms the number of tickets.
3. **User** clicks buy ticket button on the page.
4. **System** directs User to the checkout page.

Entry condition This use case **extends** the displayEvent use case.

Exit conditions **User** is directed to checkout page.

Quality Requirements 1) System should feedback about ticket in 5 seconds.

Use case name **PayTicket**

Participating Actors Initiated by **User**

Flow of events

1. **User** clicks payTicket button on the checkout page.
2. **System** shows feedback message to user "You purchase has been complete".

Entry condition This use case **extends** the buyTicket use case.

Exit conditions **User** is shown a feedback message.
User paid for the ticket.

Use case name **CreateAnAddress**

Participating Actors Initiated by **User**

Flow of events

1. **User** clicks My Account on the main page of the website .
2. **System** shows the information about user account.
3. **User** clicks the My Addresses button.
4. **System** shows the My Addresses screen.
5. **User** clicks the create an address button and he writes his address.
6. **System** shows his address in address screen.

Entry condition **User** logged in.

Exit conditions **User** address is created.

Use case name **DisplayAddresses**

Participating Actors Initiated by **User**

Flow of events

1. **User** clicks My Account on the main page of the website .
2. **System** shows the information about user account.
3. **User** clicks the My Addresses button.
4. **System** shows the addresses to the user.

Entry condition **User** is logged in.
User has created at least one address.

Exit conditions **User** is shown his addresses.

Use case name **UpdateAddress**

Participating Actors Initiated by **User**

Flow of events

1. **User** clicks My Account on the main page of the website .
2. **System** shows the information about user account.
3. **User** clicks the My Addresses button.
4. **System** shows the My Addresses screen.
5. **User** clicks the update address button.
6. **System** shows the update screen of address.
7. **User** updates his address and clicks the save button.
8. **System** shows his new address.

Entry condition **User** is logged in.

Exit conditions **User** updated his address.

Use Case Name: DeleteAddress

Participating Actors: Initiated by User

Flow of Events:

1. User logs in and clicks on My Account button on the main page.
 2. System shows My Account page.
 3. User clicks on My Addresses button from the options.
 4. System shows users addresses.
 5. User clicks the delete address button.
 6. System deletes the address information and notifies user that their address has been deleted succesfully.
-

Entry Condition: User enters website.

Exit Condition: User receives feedback about their address being deleted.

Quality requiremenets: 1. System should give feedback within 5 seconds.

Use Case Name: **DisplayTickets**

Participating Actors: Initiated by **User**

Flow of Events:

1. **User** clicks on the log in button and logs in then clicks "My account".
 2. **System** shows My Account page.
 3. **User** clicks on My Tickets from the options.
 4. **System** displays users tickets.
-

Entry Condition: **User** enters website.

Exit Condition: **User** sees the ticket list being displayed.

Quality requiremenets: 1. System should display users tickets within 5 seconds.

Use Case Name: **displayPersonalInfo**

Participating Actors: Initiated by **User,EventHolder**

Flow of Events:

1. **User or EventHolder** logs in and chooses “My Account” from the main page.
 2. **System** shows “My Account” page.
 3. **User or EventHolder** clicks on “My Profile”
 4. **System** directs user to their profile page.
 5. **User or EventHolder** sees their personal information displayed.
-

Entry Condition: **User or EventHolder** enters website.

Exit Condition: **User or EventHolder** is shown their personal information on the screen.

Quality requiremenets: 1. System should display the required page within 5 seconds.

Use Case Name: **updatePersonalInfo**

Participating Actors: Initiated by **User**

Flow of Events:

1. **User** logs in and chooses “My Account” from the main page.
 2. **System** shows “My Account” page.
 3. **User** clicks on “My Profile”
 4. **System** directs user to their profile page.
 5. **User** sees their personal information displayed and clicks “Edit Profile” button.
 6. **System** becomes editable and lets user edit their profile.
 7. **User** changes their personal information and clicks “Save”.
 8. **System** displays “Profile Updated” message on the screen.
-

Entry Condition: **User** logs in.

Exit Condition: **User** succesfully updated her personal info.

Quality requiremenets: 1. System should display the “Profile Updated” message and save the new information within 5 seconds.

Use case name **AddNewEvent**

Participating Actors Initiated by **EventHolder**

Flow of events

1. **EventHolder** enters the website and logs in.
2. **System** directst him to the **EventHolder's** page of the website.
3. **EventHolder** clicks Events button on the page.
4. **System** shows a page consisting of the list of events that has been created by **EventHolder** before.
5. **EventHolder** clicks on addNewEvent button on the Events page.
6. **System** shows event creation form to the **EventHolder**.
7. **EventHolder** enters event information to the form and confirms.
8. **System** shows the created event to the **EventHolder**.

Entry condition **EventHolder** logins to website.

Exit conditions **EventHolder** has successfully created an event.

Use case name **DeleteEvent**

Participating Actors Initiated by **EventHolder,Admin**

Flow of events

1. **The Actor** logs in and clicks on Events button on the Actor's page.
3. **System** shows a list of events that has been created before.
4. **The Actor** clicks on the Edit button next to the event they wanted to delete.
5. **System** shows the event information to the **The Actor**
6. **The Actor** clicks on the Delete Event button at the bottom of the page.
7. **System** gives a feedback to the **The Actor** about the deletion.

Entry condition There is at least one event created before.

Exit conditions **The Actor** has successfully deleted an event.

Use case name **ViewInformationofUsers**

Participating Actors Initiated by **Admin**

Flow of events

1. **Admin** enters the website and logs in.
2. **System** directs Admin to his page.
3. **Admin** clicks on the Users button.
4. **System** shows the information of users.

Entry condition **Admin** logins to website.

There is an at least one user account.

Exit conditions **Admin** has successfully viewed user information.

Use case name **ViewInformationofEventHolders**

Participating Actors Initiated by **Admin**

Flow of events 1. **Admin** enters the website and logs in.
 2. **System** directs Admin to his page.
 3. **Admin** clicks on the EventHolders button.
 4. **System** shows the information of EventHolders.

Entry condition **Admin** logins to website.
 There is an at least one event holder account.

Exit conditions **Admin** has successfully viewed event holder information.

Use case name **DeleteUser**

Participating Actors Initiated by **Admin**

Flow of events

1. **Admin** logs in to website
2. **System** directs **Admin** to the admin page.
3. **Admin** clicks users button on the page.
 4. **System** shows the list of users and their personal information.
 5. **Admin** selects the user they want to delete and clicks on the “Delete” button.
 6. **System** gives feedback to **Admin** about the account deletion.

Entry condition **Admin** enters to website.

There is at least one user account.

Exit conditions **Admin** has successfully deleted user account.

Use case name **DeleteEventHolder**

Participating Actors Initiated by **Admin**

Flow of events 1. **Admin** logs in to website

 2. **System** directs **Admin** to the admin page.

 3. **Admin** clicks eventholders button on the page.

 4. **System** shows the list of eventholders and their
personal information.

 5. **Admin** selects the event holder they want to delete and
clicks on the "Delete" button.

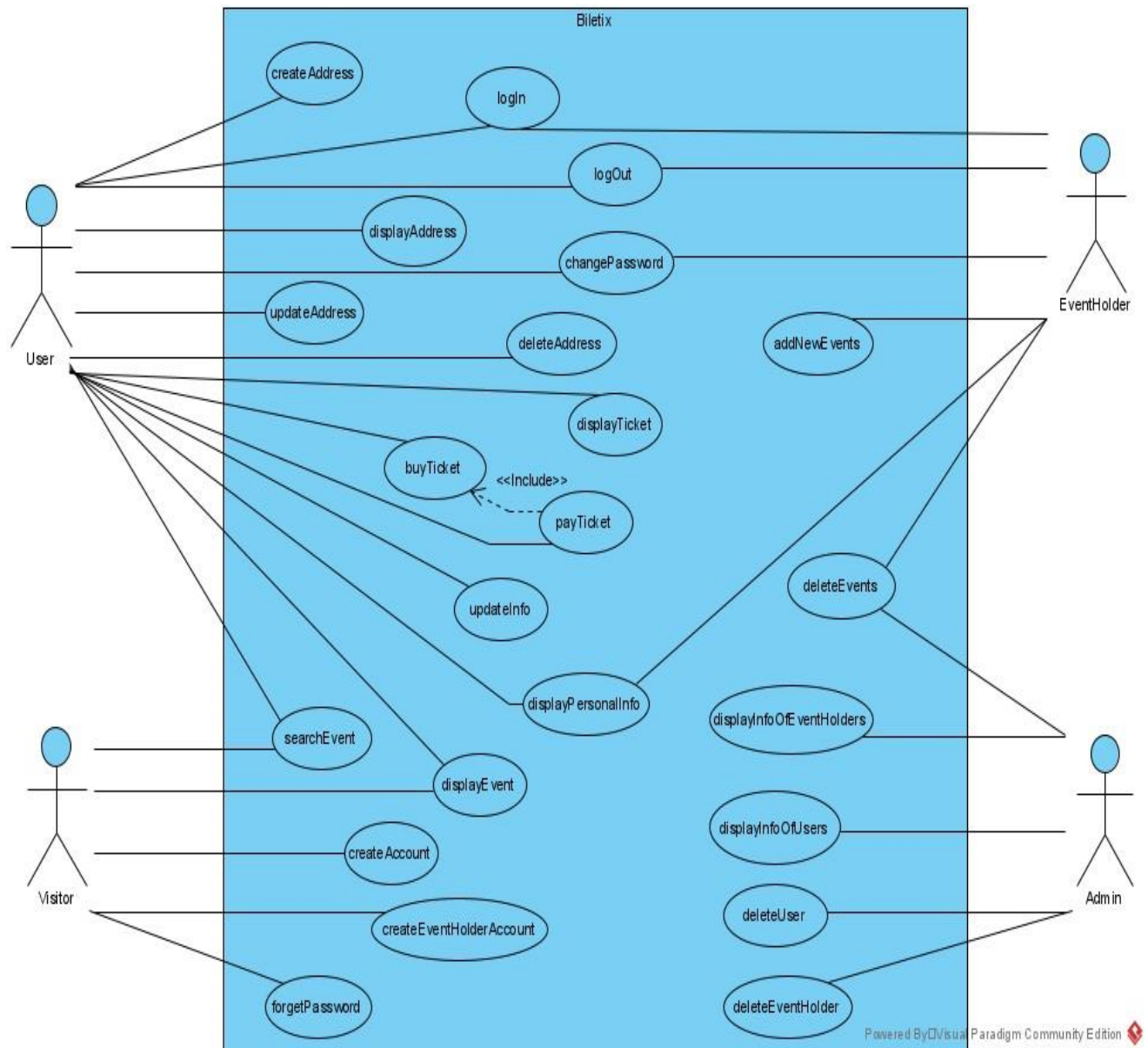
 6. **System** gives feedback to **Admin** about the account deletion.

Entry condition **Admin** enters to website.

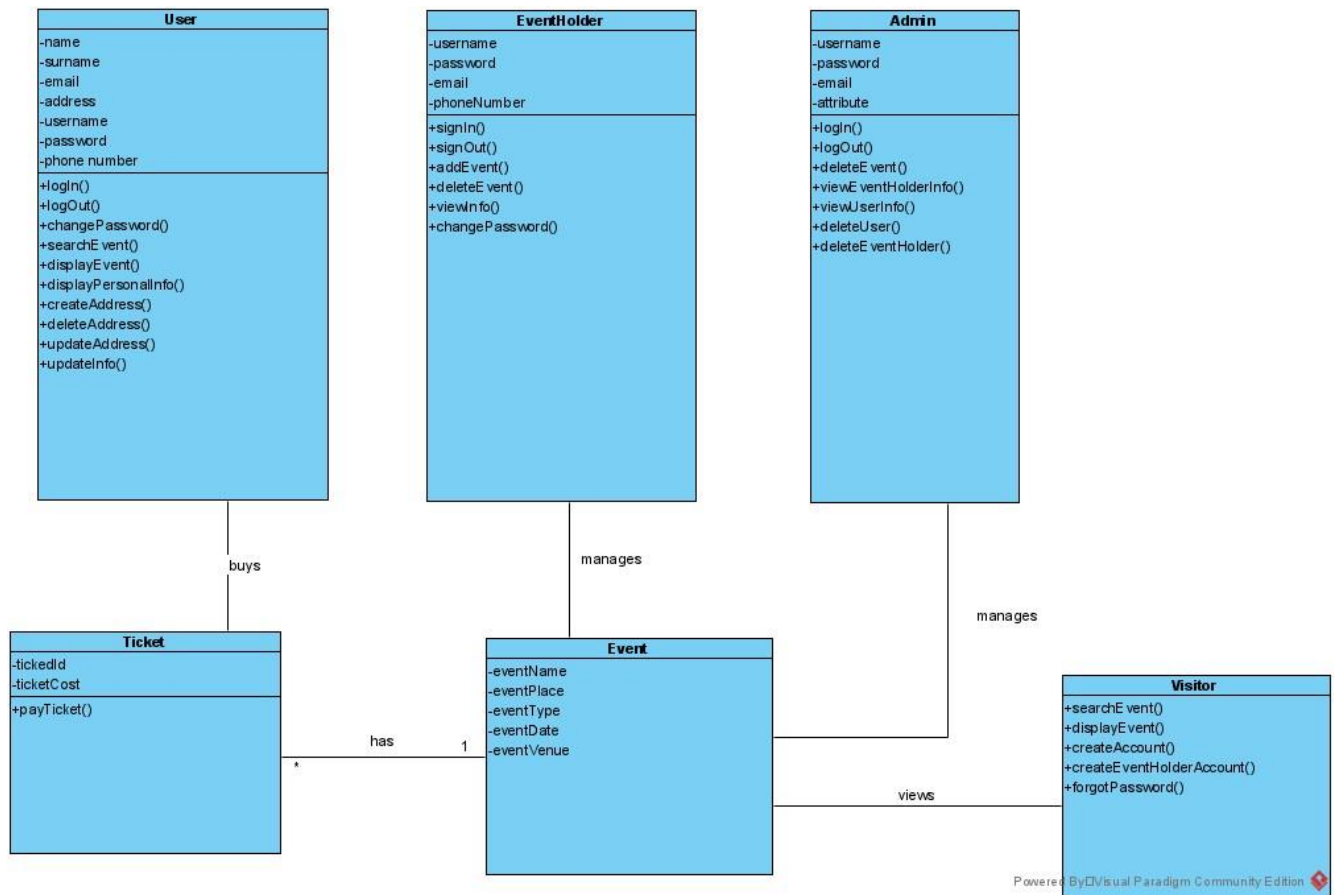
 There is at least one event holder account.

Exit conditions **Admin** has successfully deleted user account.

USE CASE DIAGRAM



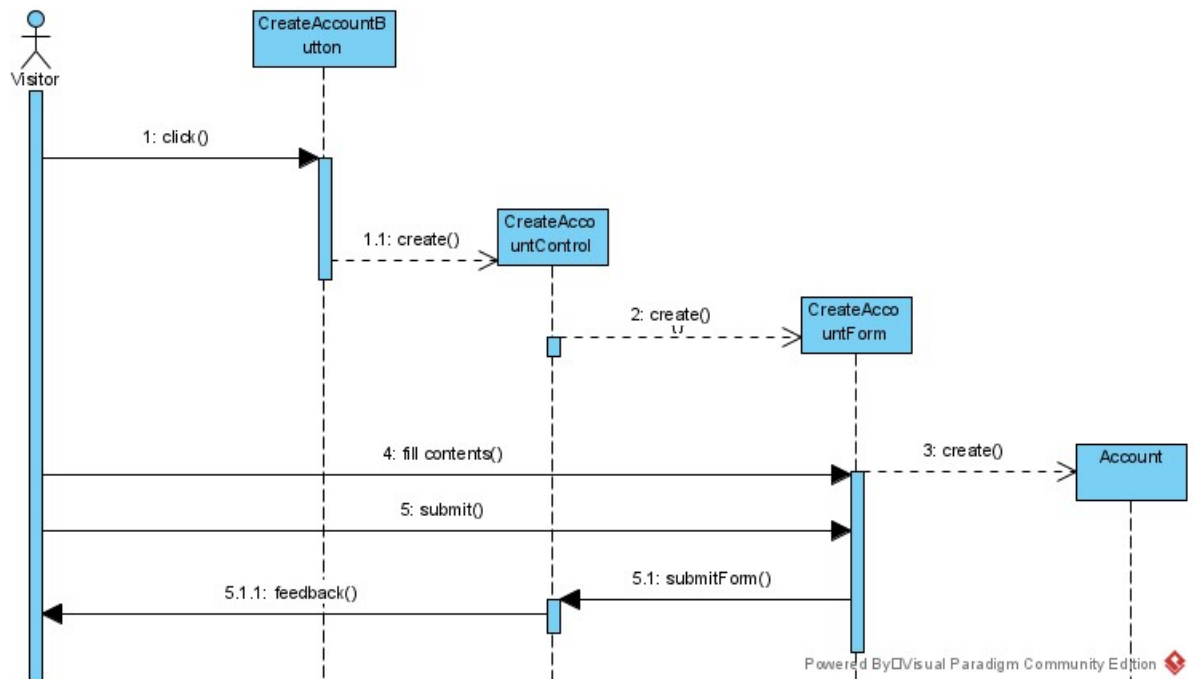
Object model



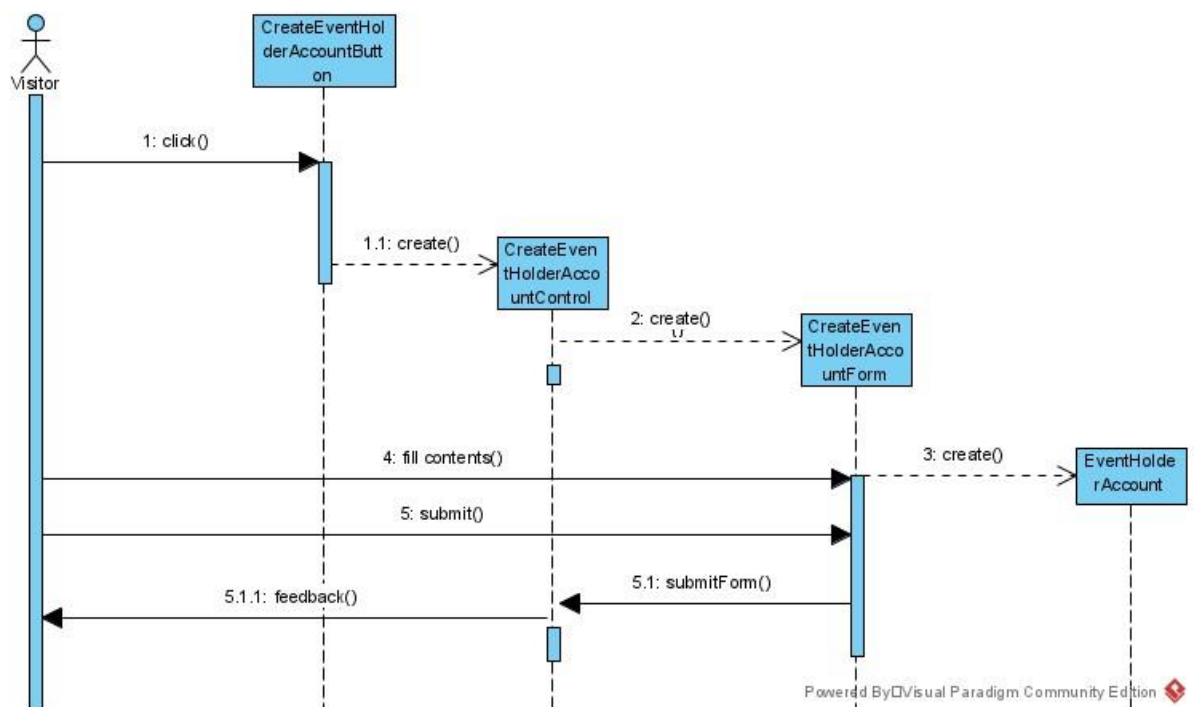
<Biletix>

Dynamic model

1) Create Account

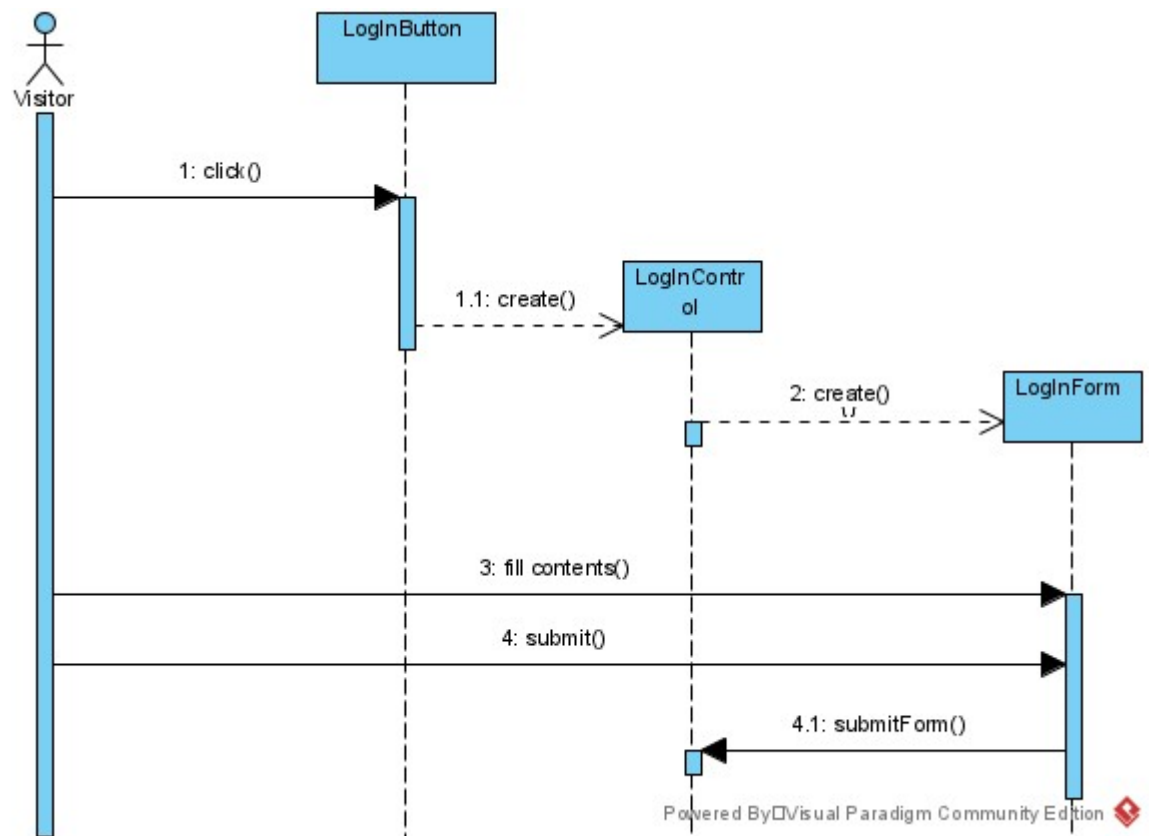


2) Create EventHolder Account

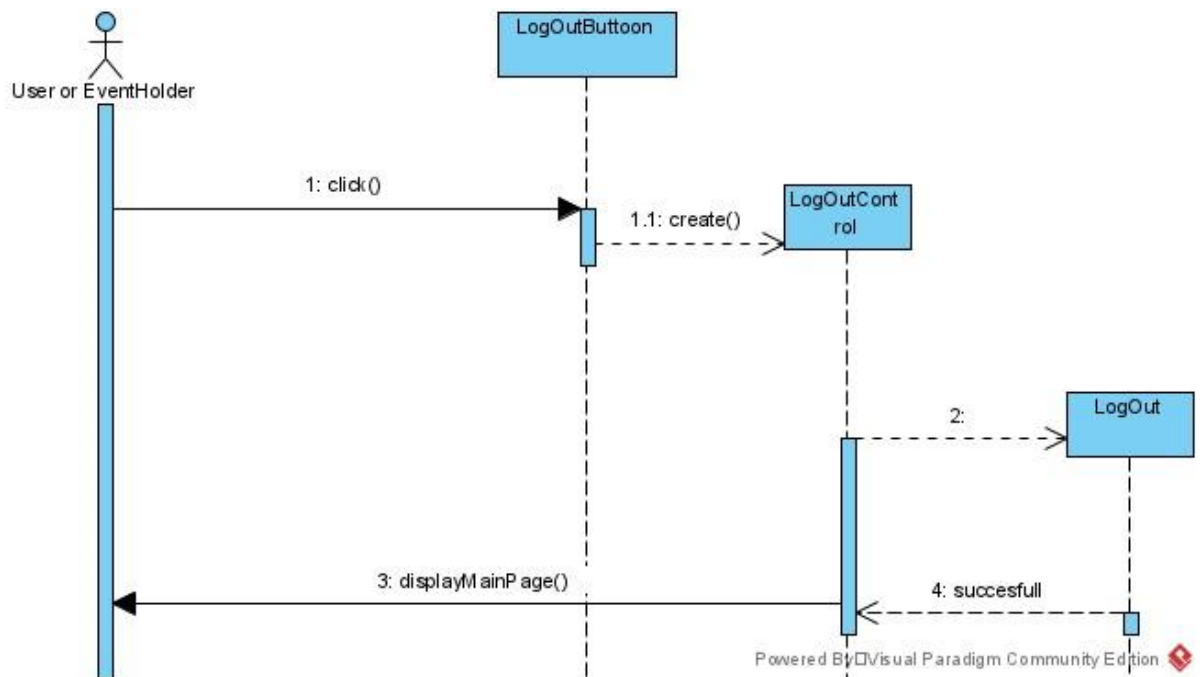


<Biletix>

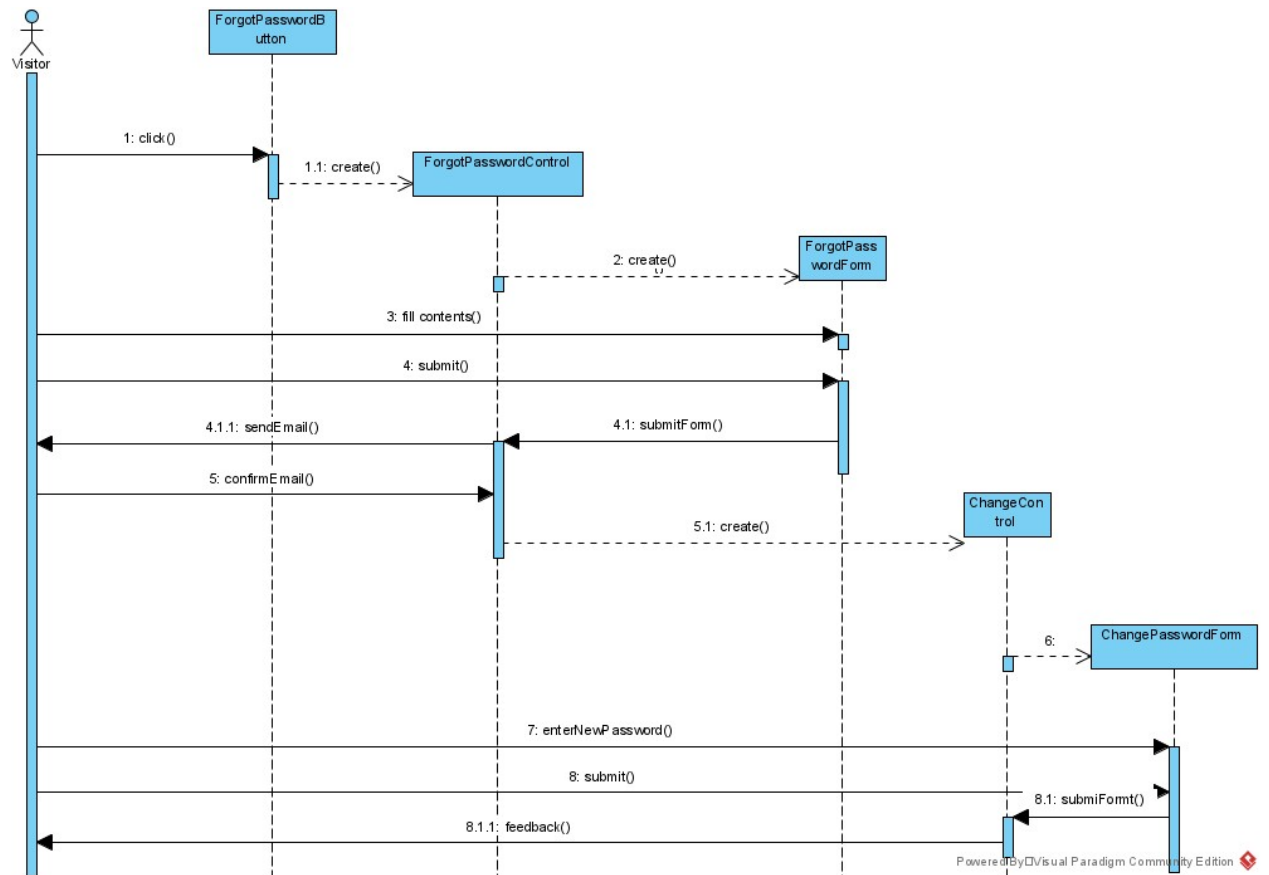
3) LogIn



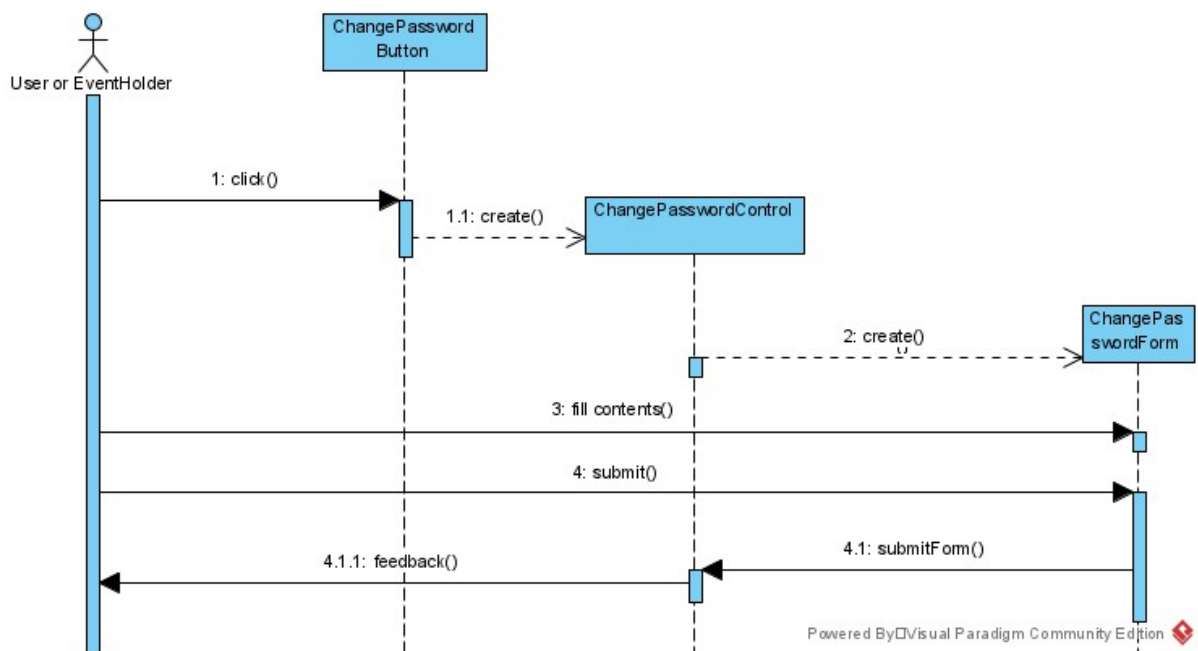
4) LogOut



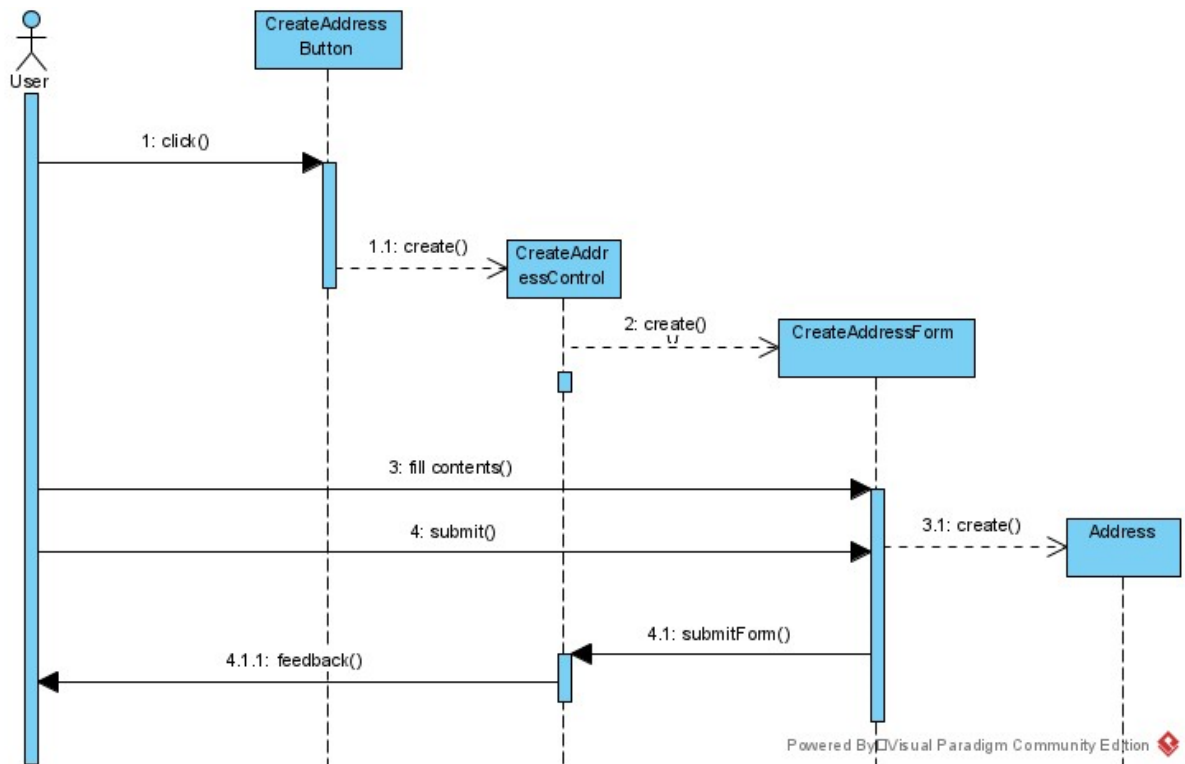
5) Forget Password



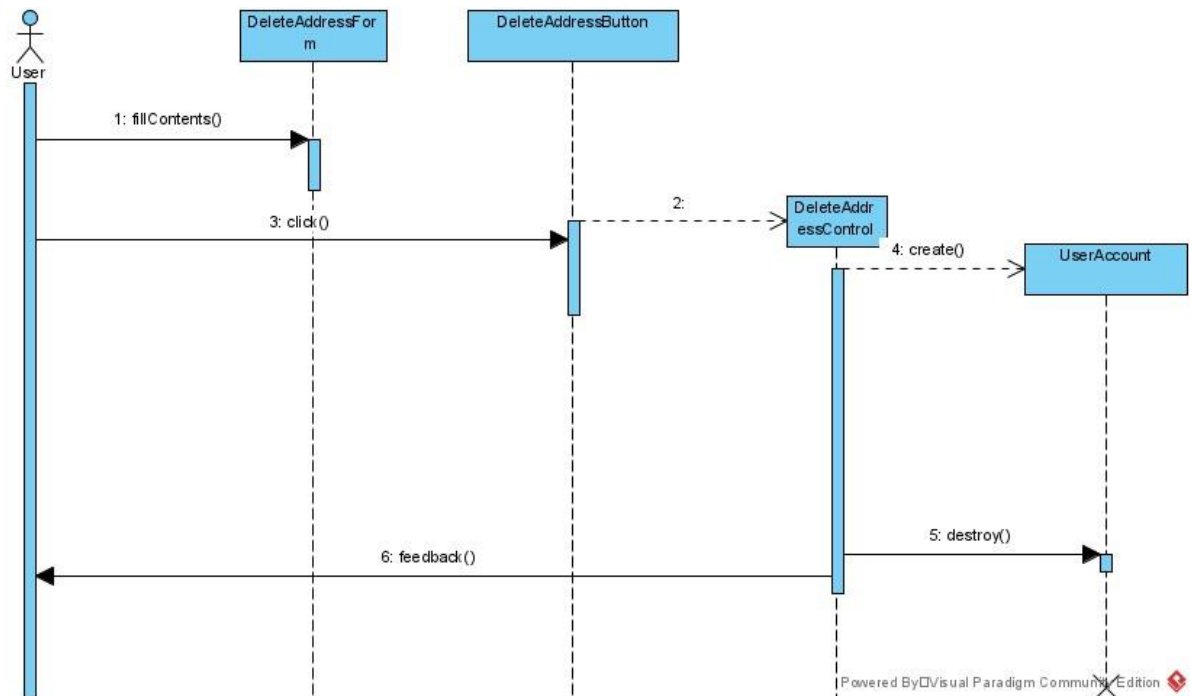
6) Change Password



7) Create Address

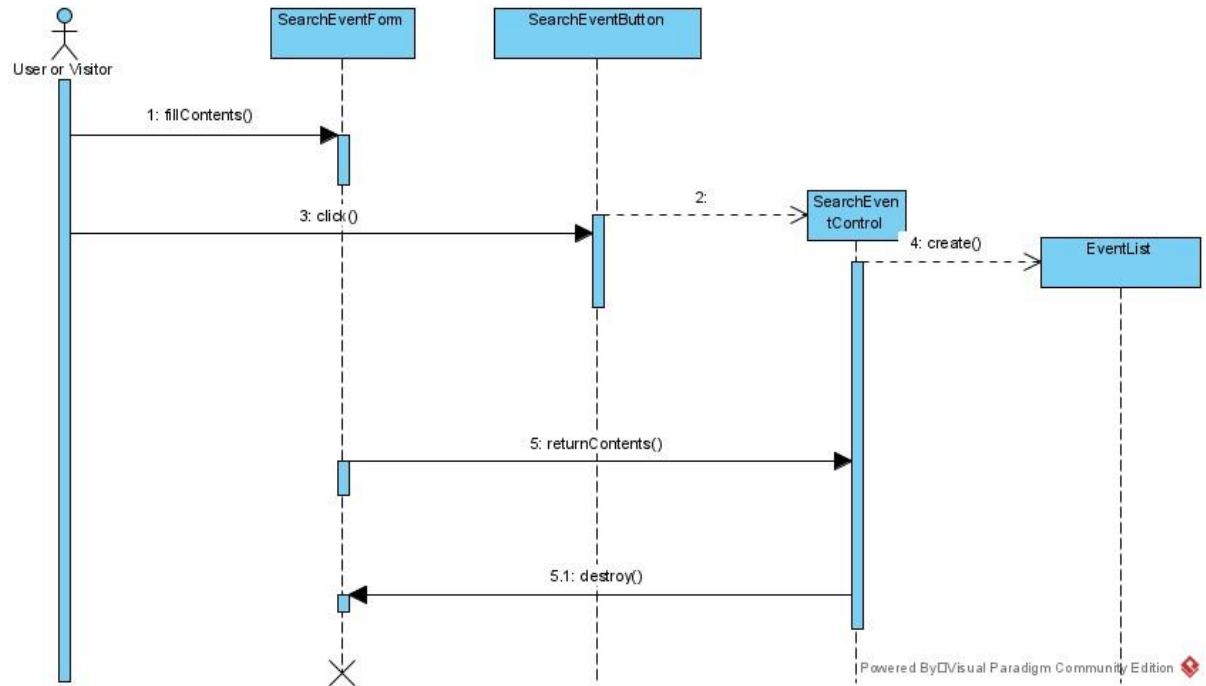


8) Delete Address

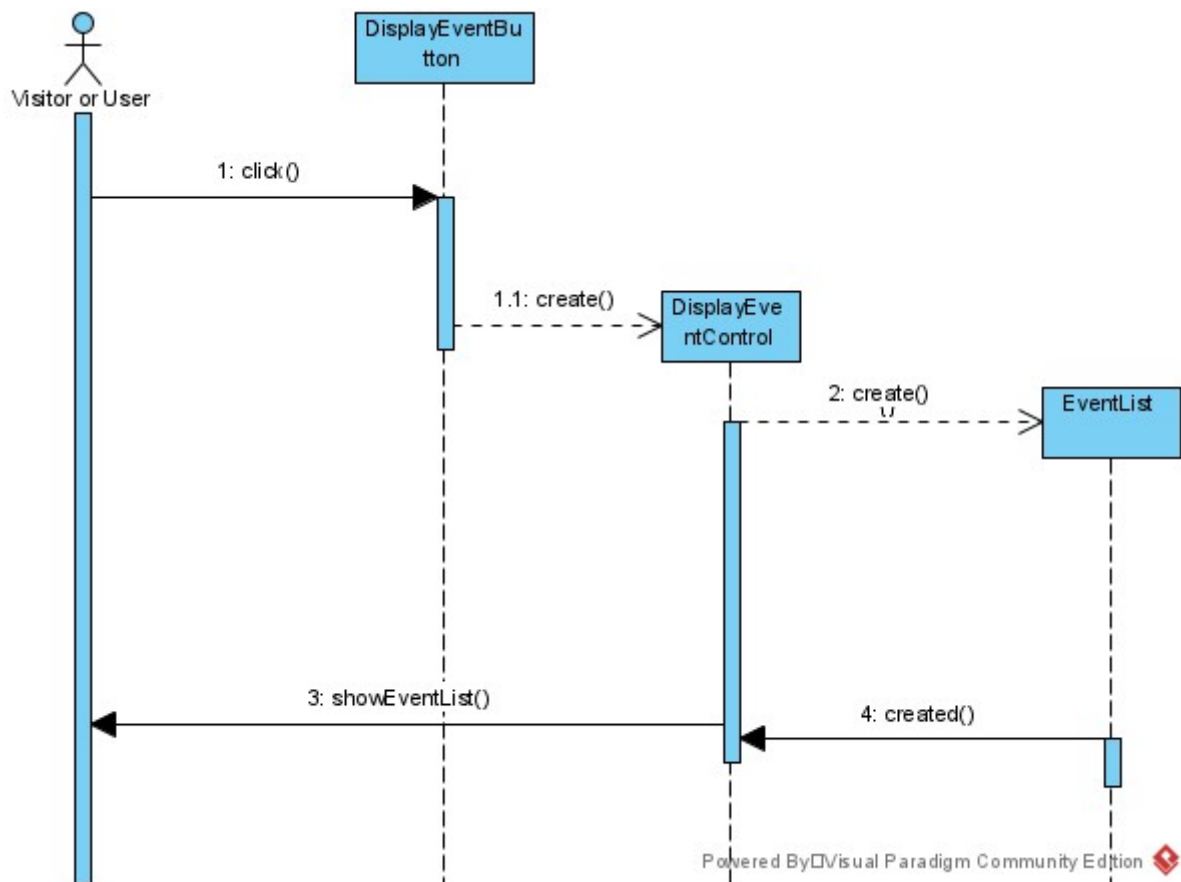


<Biletix>

9) Search Event

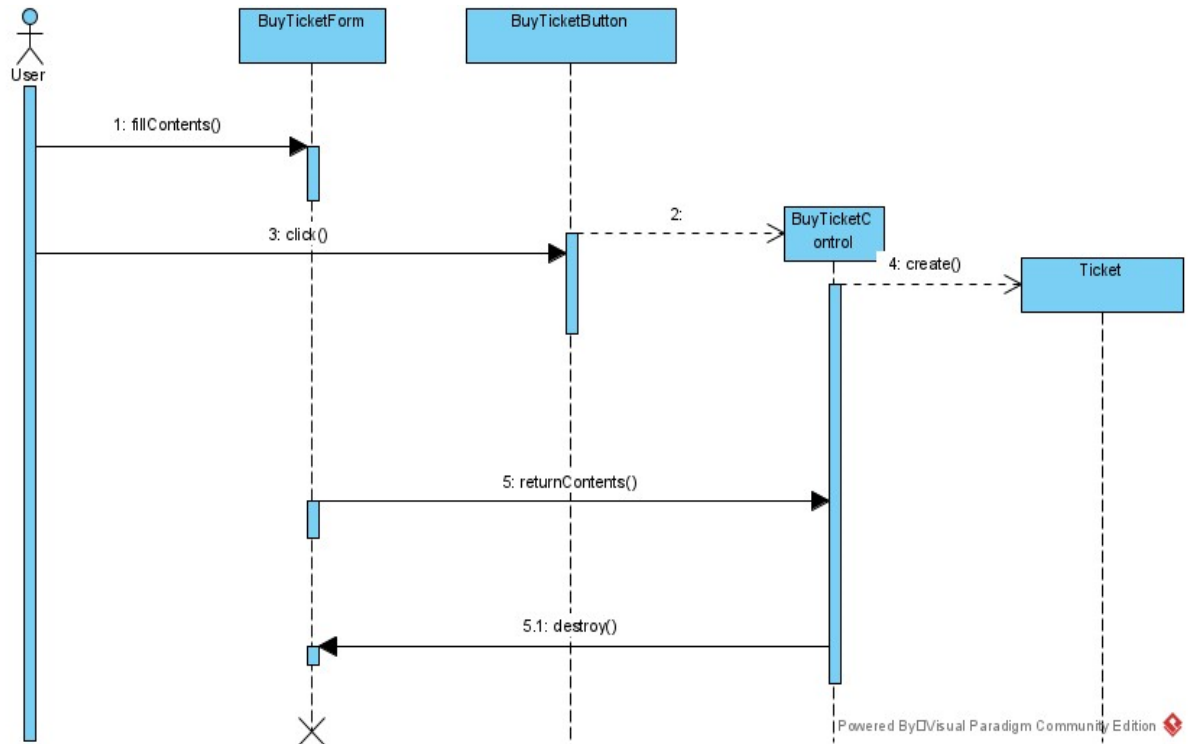


10) Display Event

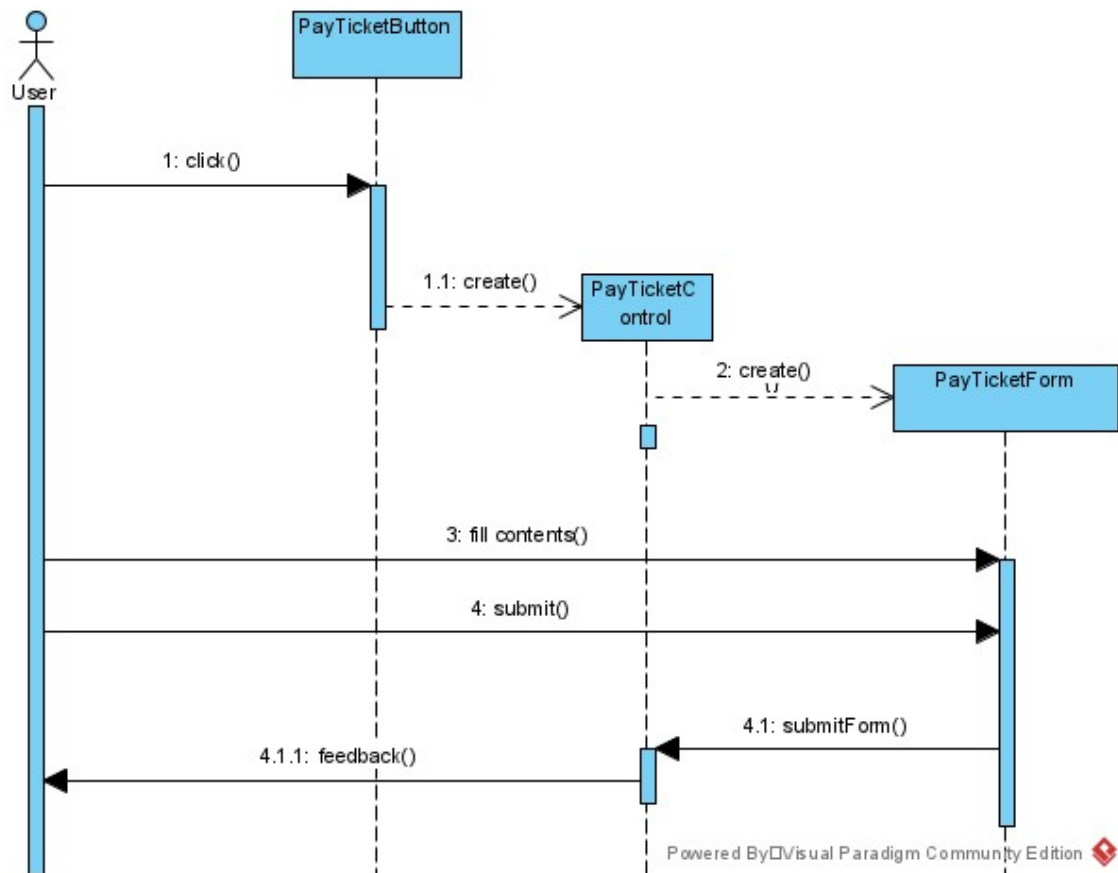


<Biletix>

11) Buy Ticket

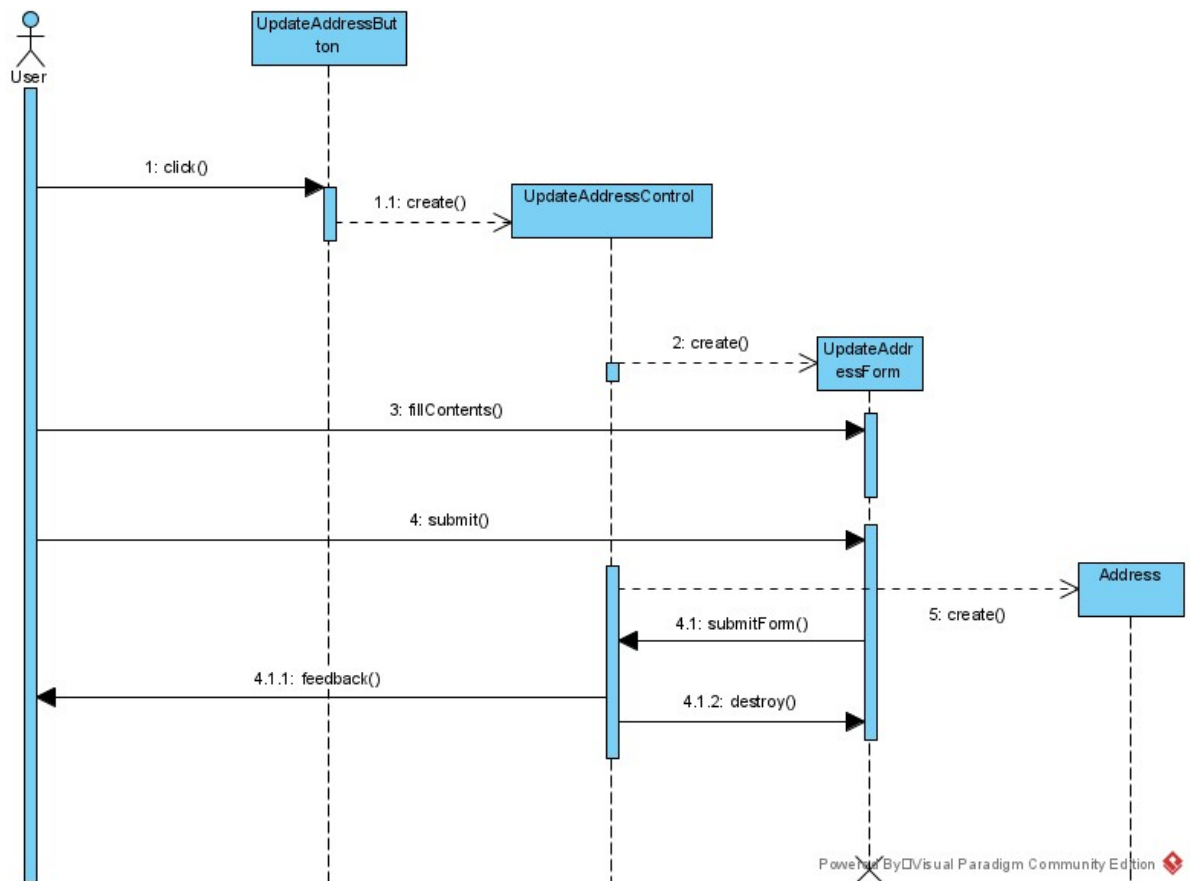


12) Pay Ticket

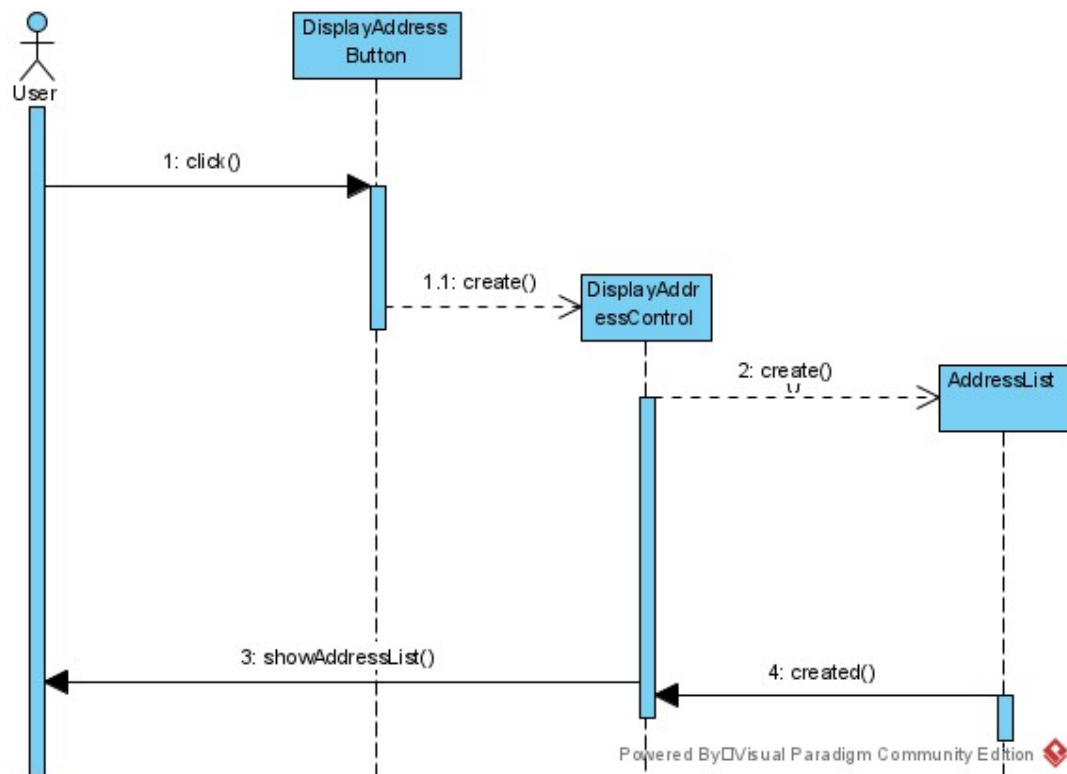


13) Update Address

<Biletix>

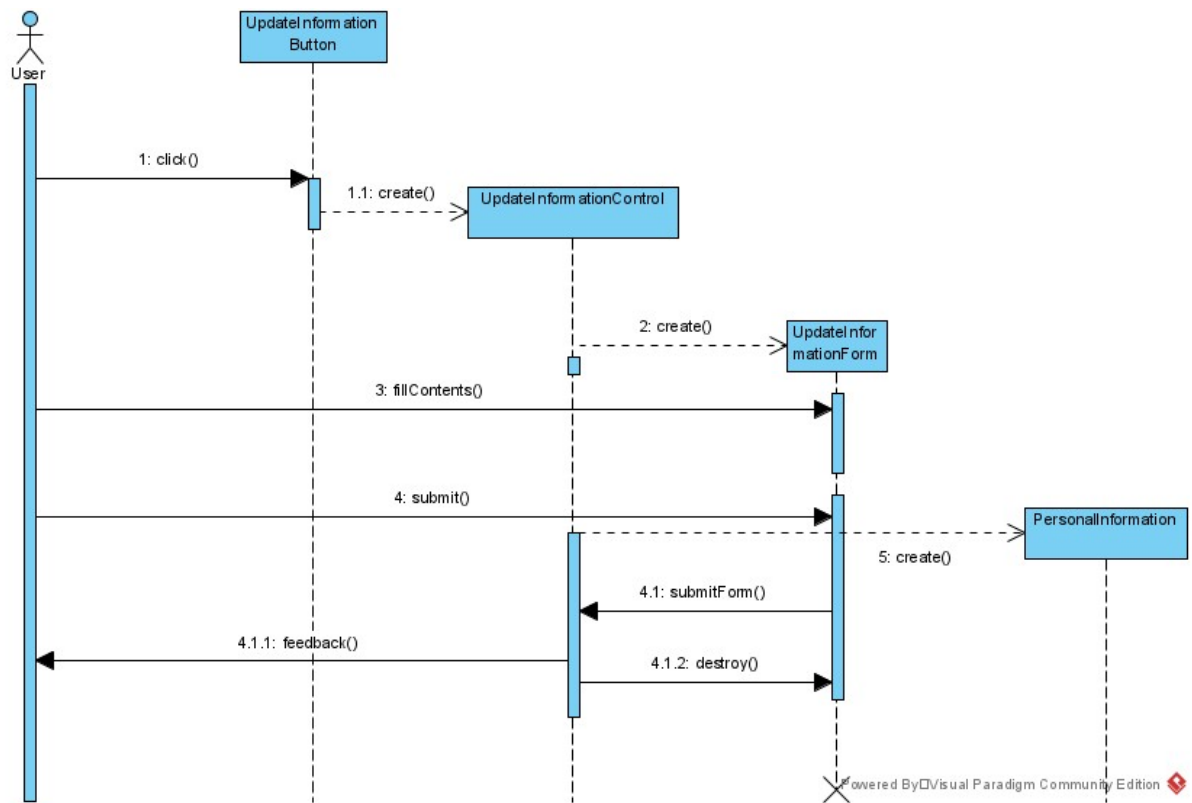


14) Display Address

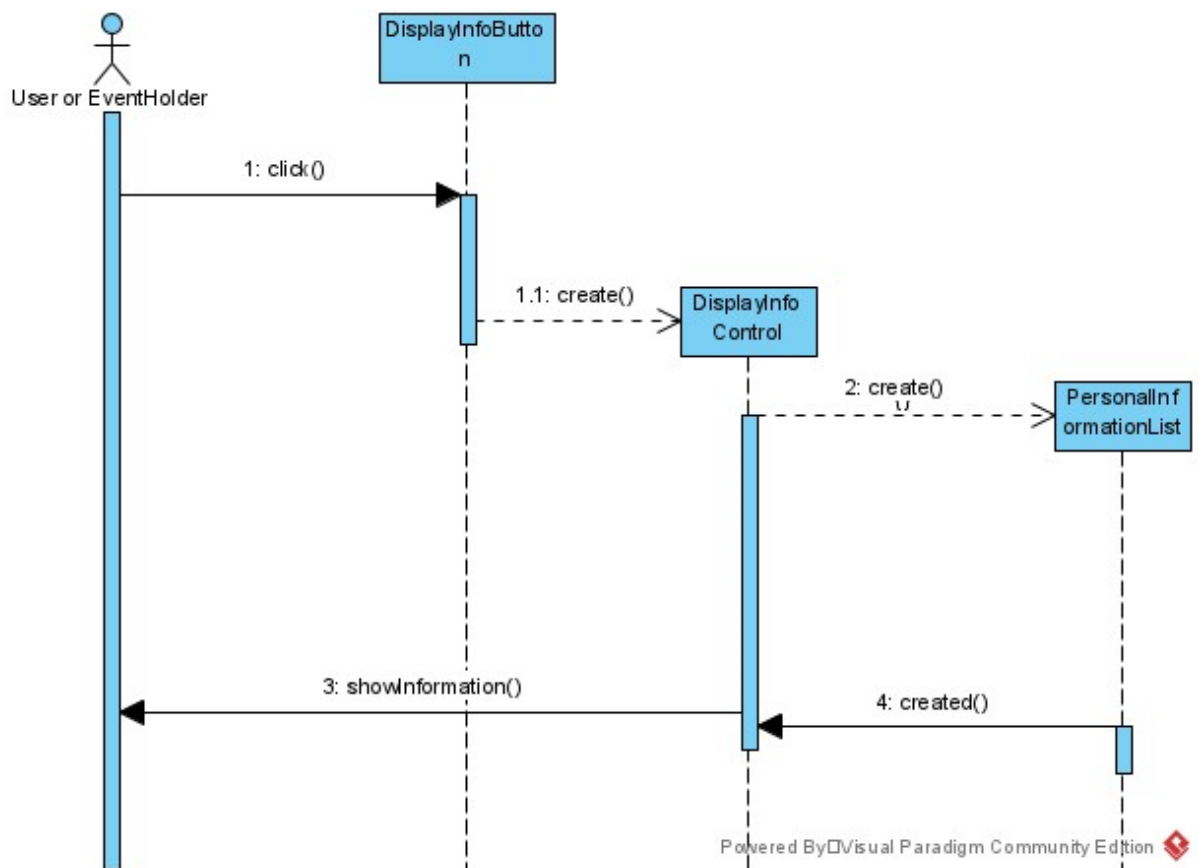


15) Update Information

<Biletix>

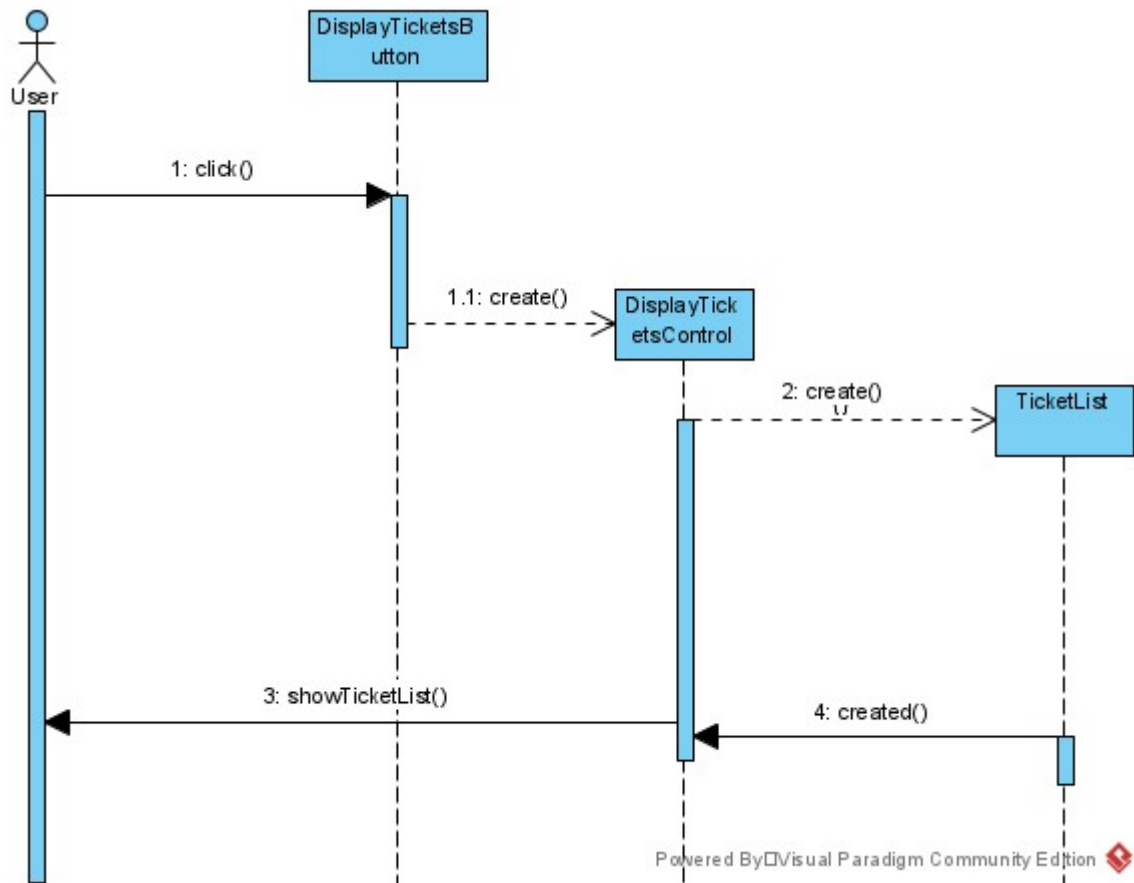


16) Display Personal Information

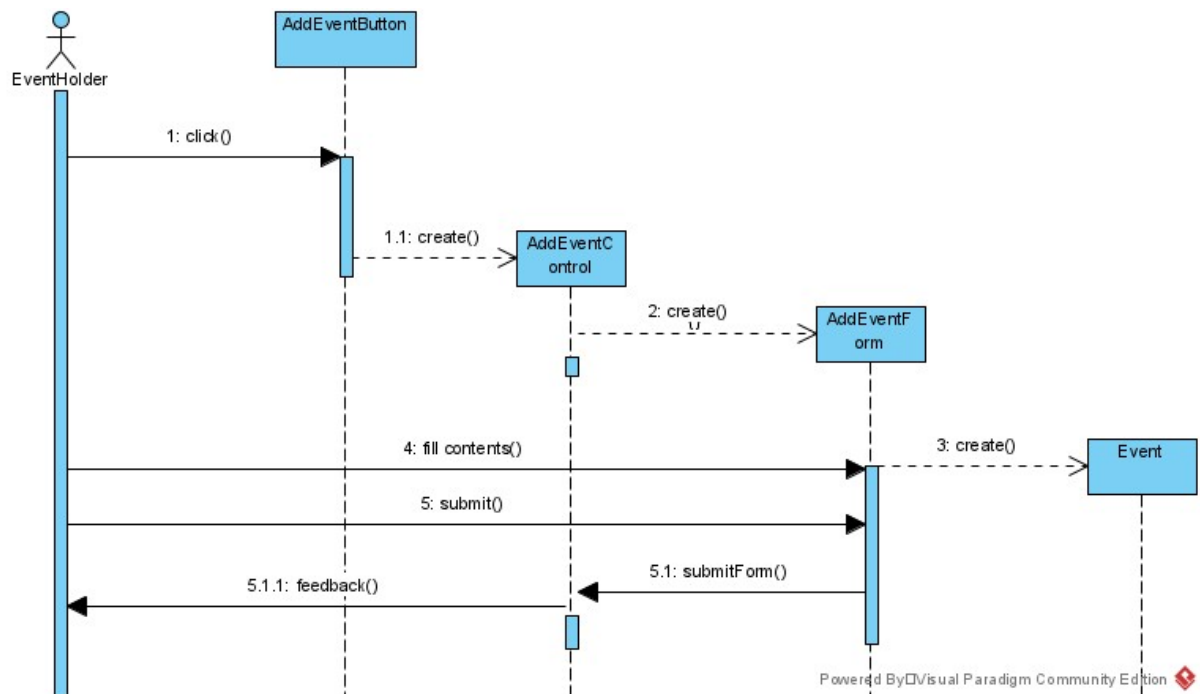


17) Display Ticket

<Biletix>

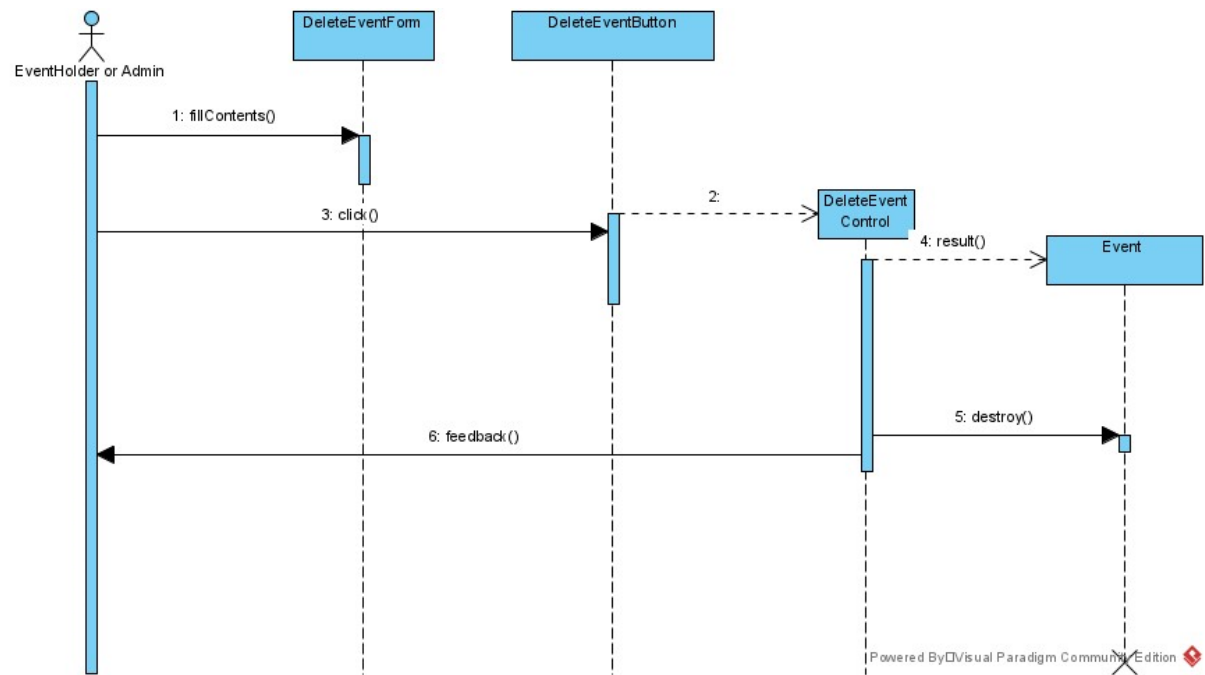


18) Add Event

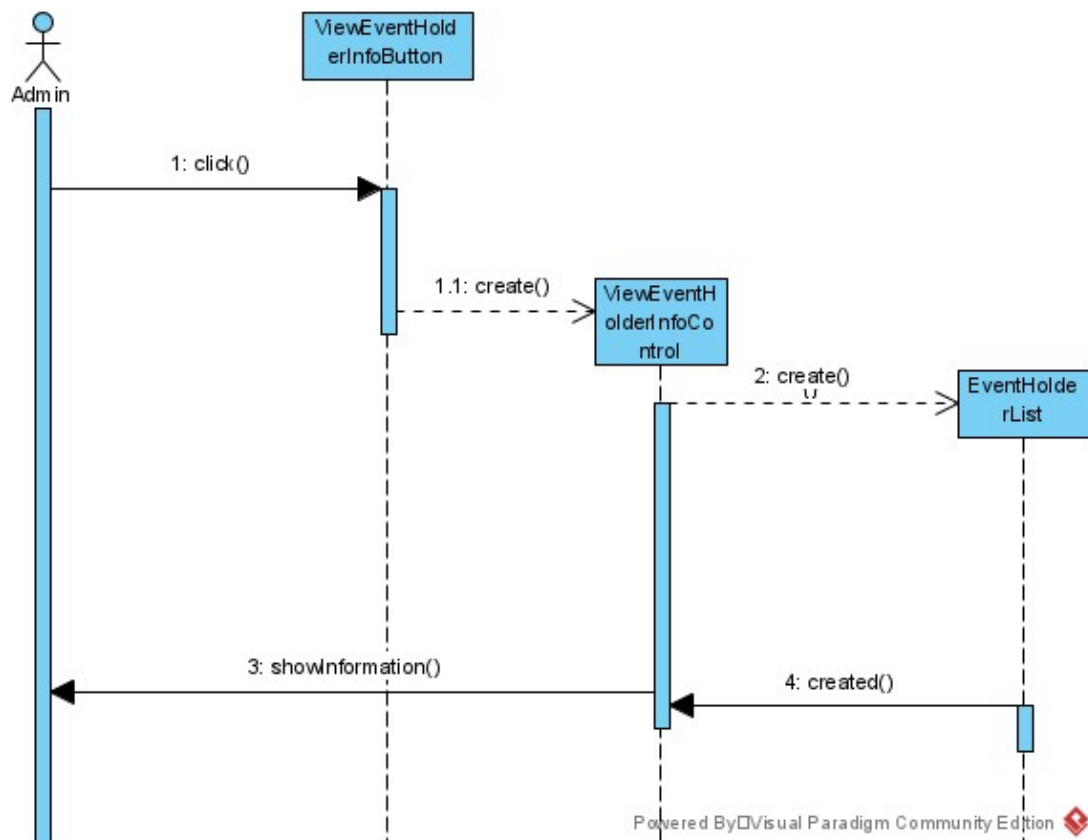


19) Delete Event

<Biletix>

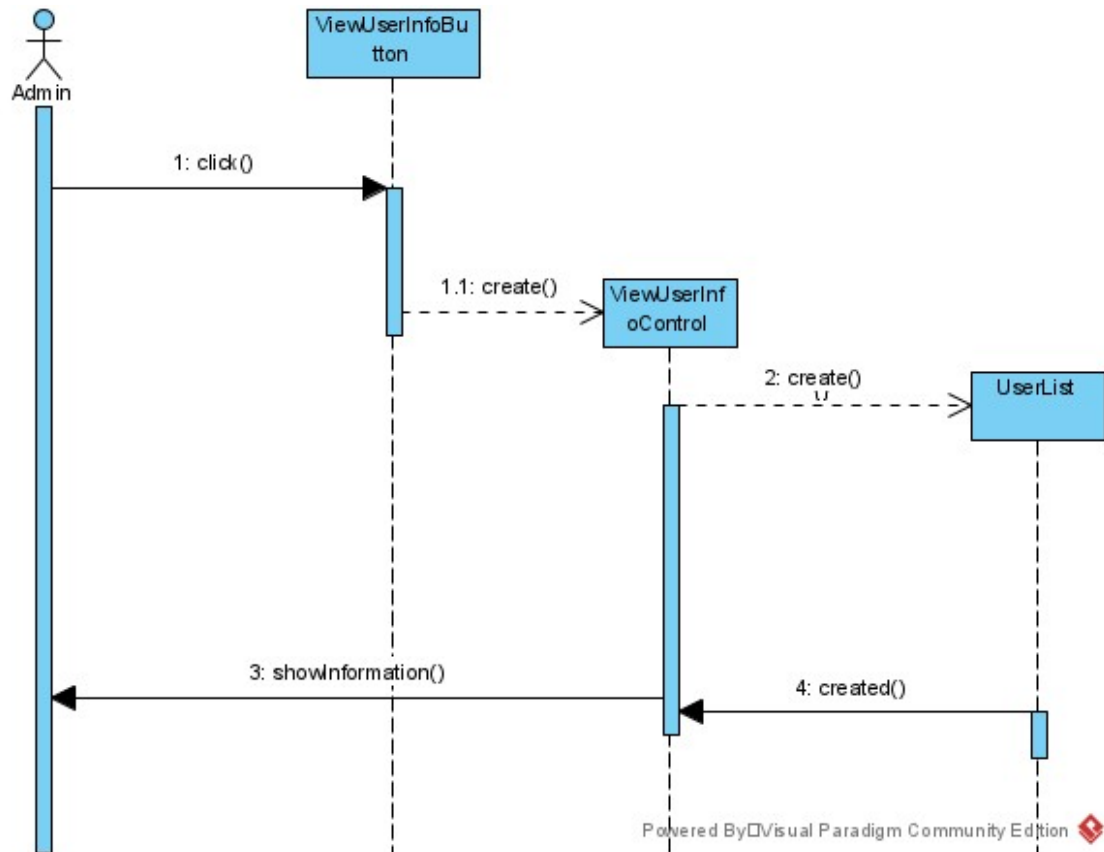


20) View EventHolder Info

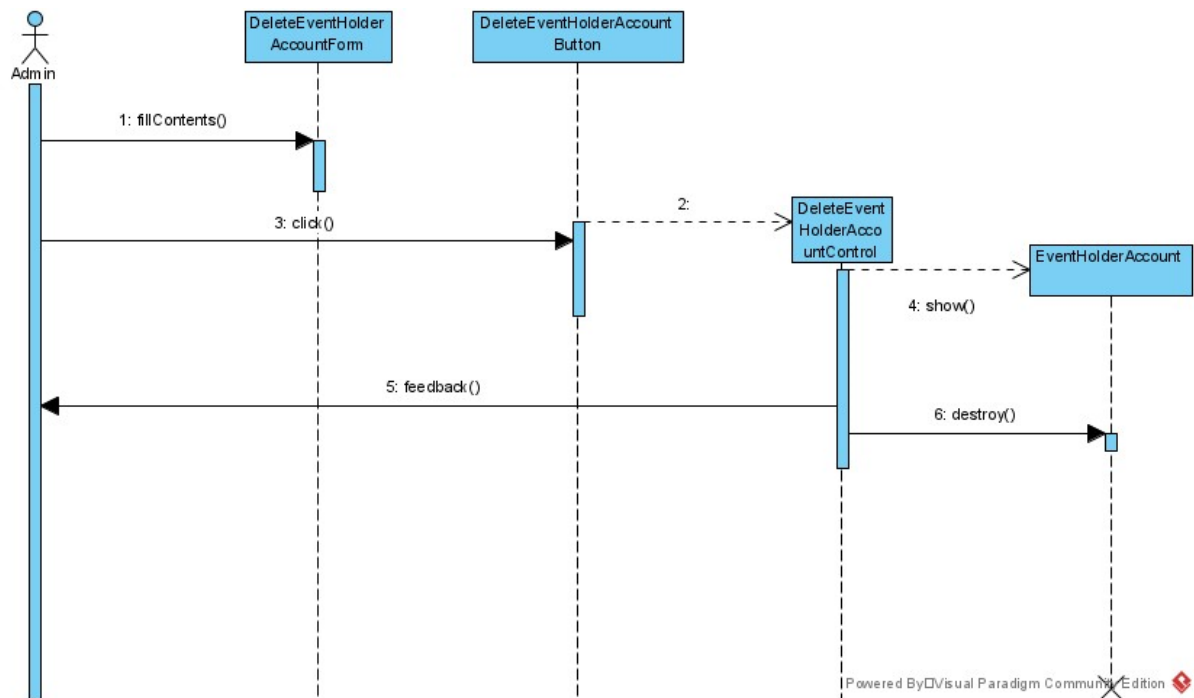


<Biletix>

21) View User Info

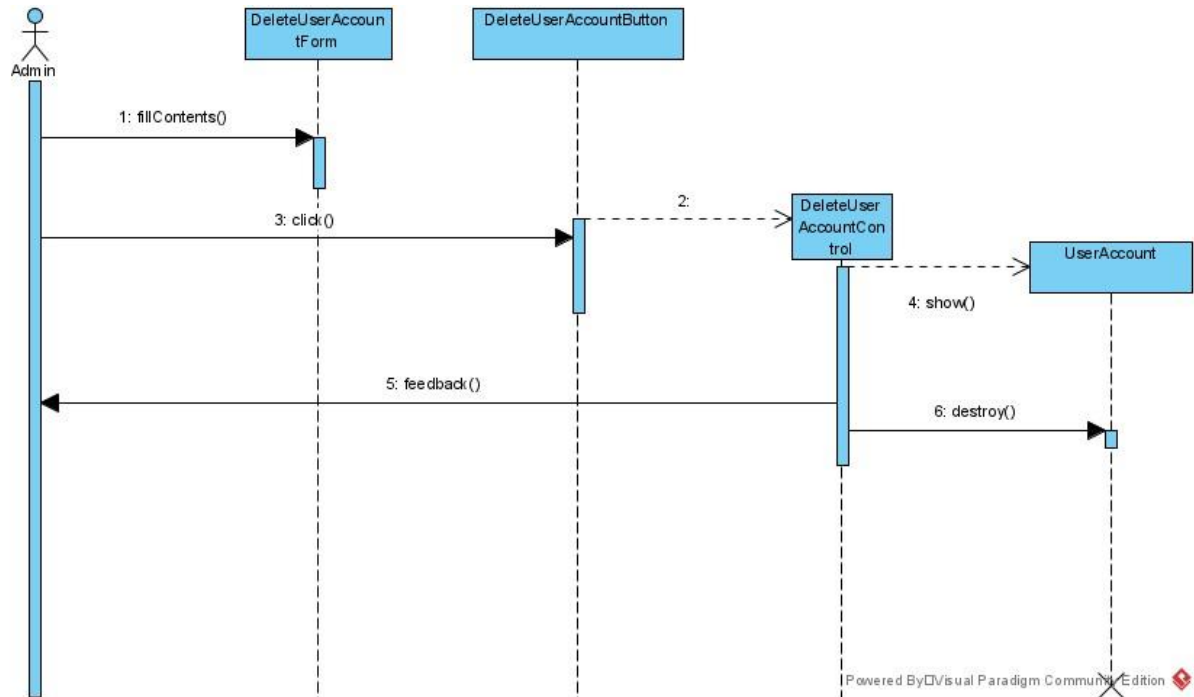


22) Delete EventHolder Account



<Biletix>

23) Delete User Account



User interface—navigational paths and screen mock-ups

The screenshot shows the Biletix web application interface. At the top, there is a navigation bar with the Biletix logo, a search bar, and links for Sign Up and Log In. Below the navigation bar, there is a section for event discovery with filters for Categories (All Categories, Art, Sport, Music, Stage, Education and More), Date (27/11/2021), and Location (All Turkey, Istanbul, Ankara, Izmir, Adana). The main content area features a large Sign Up form with fields for Name, Surname, Email address, Username, Phone number, and Password. A CREATE ACCOUNT button is located at the bottom of the form, and a SIGN IN link is positioned below it.

← → × 🏠

https://Biletixx.com

🔍

Biletixx

Art Sport Music Stage Education and More

[Sign Up](#) | [Log In](#)

Discover

All Categories

Art
Sport
Music
Stage
Education and

Date

27/11 /2021

Location

All Turkey

Istanbul
Ankara
İzmir
Adana

Search Event

Update Password

New Password:

Re-Type New Password:

Update Password

← → × 🏠

https://Biletixx.com

🔍

Biletixx

Art Sport Music Stage Education and More

[Profile](#)
[My Profile](#)
[My Addresses](#)
[My Tickets](#)
[Log Out](#)

Discover

All Categories

Art
Sport
Music
Stage
Education and

Date

27/11 /2021

Location

All Turkey

Istanbul
Ankara
İzmir
Adana

Search Event

My Profile

Name: User's Name

Surname: User's Surname

Gender: User's Gender

Birth Date: User's Birth Date

Phone Number: User's Phone Number

Update Info

Update Password

Old Password:

New Password:

Re-Type New Password:

Update Password

Biletixx

Art Sport Music Stage Education and More

Sign Up | Log In

Discover All Categories

Art
Sport
Music
Stage
Education and


Date 27/11 /2021


Location All Turkey

Istanbul
Ankara
Izmir
Adana
...

Search Event

Log In

 Username

 Password

☐ Remember Me

[Forgot Password ?](#)

Login

Biletixx

Art Sport Music Stage Education and More

Discover Music

Art
Sport
Music
Stage
Educa

Date 27/11 /2021

Location Istanbul

Istanbul
Ankara
Izmir
Adana
...

Search Event

Profile

My Profile
My Addresses
My Tickets
Log Out

< Event Name >

Event Name

Event Name

←

→

✕

🏠

https://Biletixx.com

🔍

Biletixx

Art Sport Music Stage Education and More

Profile

My Profile

My Addresses

My Tickets

Log Out

Discover

All Categories

Art

Sport

Music

Stage

Education and

Date

27/11/2021

📅

Location

All Turkey

Istanbul

Ankara

İzmir

Adana

...

Search Event

Category > Music

Event Name



Event Description

Event Date

Venue

Event Time

Find Tickets

Select Date

Event Date, Event Time

Ticket Type

Category 1- Price

Category 2- Price

...

Continue

←

→

✕

🏠

https://Biletixx.com

🔍

Biletixx

Art Sport Music Stage Education and More

Sign Up

Log In

Discover

All Categories

Art

Sport

Music

Stage

Education and

Date

27/11/2021

📅

Location

All Turkey

Istanbul

Ankara

İzmir

Adana

...

Search Event

Event Name



Event Description

Event Date

Venue

Event Time

Your Tickets

Event Name

Event Date Event Time

Venue

Type:

Category 1

Tickets Price:

50\$ x 2

Subtotal:

100\$

Payment

Name on the Credit Card

Credit Card Number

Expiry Date

01 2021

02 2022

03 2023

...

CVV

Buy Now

← → × 🏠

https://Biletix.com

🔍

Biletix

Art Sport Music Stage Education and More

Profile

My Profile

My Addresses

My Tickets

Log Out

Discover

All Categories

Art

Sport

Music

Stage

Education and

Date

27/11 /2021

📅

Location

All Turkey

Istanbul

Ankara

İzmir

Adana

...

Search Event

My Tickets

Reference Number	Transaction Date	Event Name	Seats	Total	Delivery Status
AX5FJ7	27-11-2021	Büyük Ev Ablukada	C14 C15	200TL	Ongoing

Cancel Ticket

← → × 🏠

https://Biletix.com

🔍

Biletix

Art Sport Music Stage Education and More

logo

Discover

All Categories

Art

Sport

Music

Stage

Education and

Date

27/11 /2021

📅

Location

All Turkey

Istanbul

Ankara

İzmir

Adana

...

Search Event

My Events

EventImage

Event Name

Venue Location

Event Date

🗑️

EventImage

Event Name

Venue Location

Event Date

🗑️

Add New Event

Biletixx

Art Sport Music Stage Education and More

Discover

All Categories
Art
Sport
Music
Stage
Education and

Date

27/11 /2021

Location

All Turkey
İstanbul
Ankara
İzmir
Adana
...

Search Event

Event Title:

Upload Event Image:

Upload JPG or PNG

Event Description:

Organizer Name:

Event Category:

Venue:

Date and Time:

/ /

Ticket Quantity:

100

Ticket Price:

Create Event

Biletixx

Art Sport Music Stage Education and More

Discover

All Categories
Art
Sport
Music
Stage
Education and

Date

27/11 /2021

Location

All Turkey
İstanbul
Ankara
İzmir
Adana
...

Search Event

Profile
My Profile
My Addresses
My Tickets
Log Out

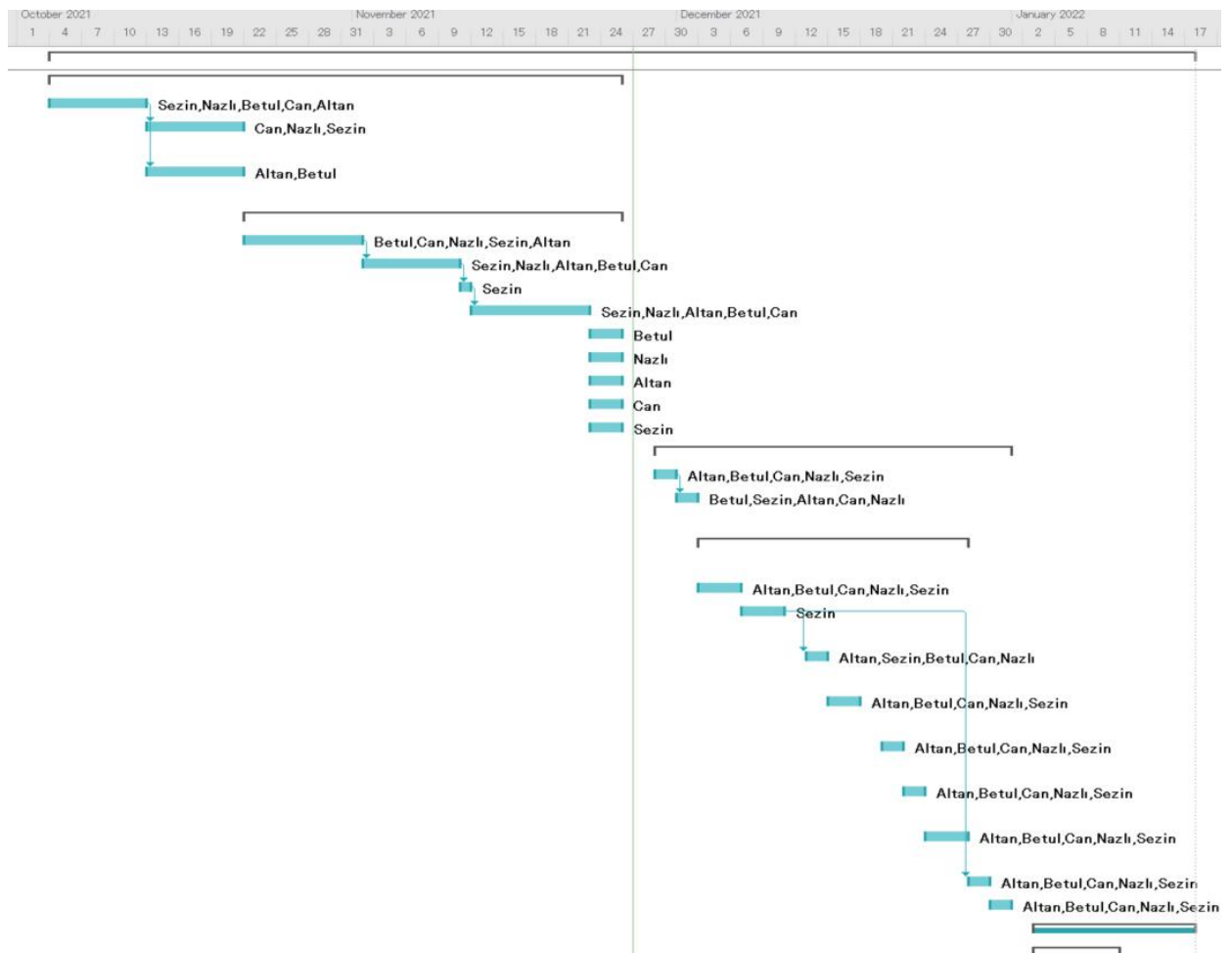
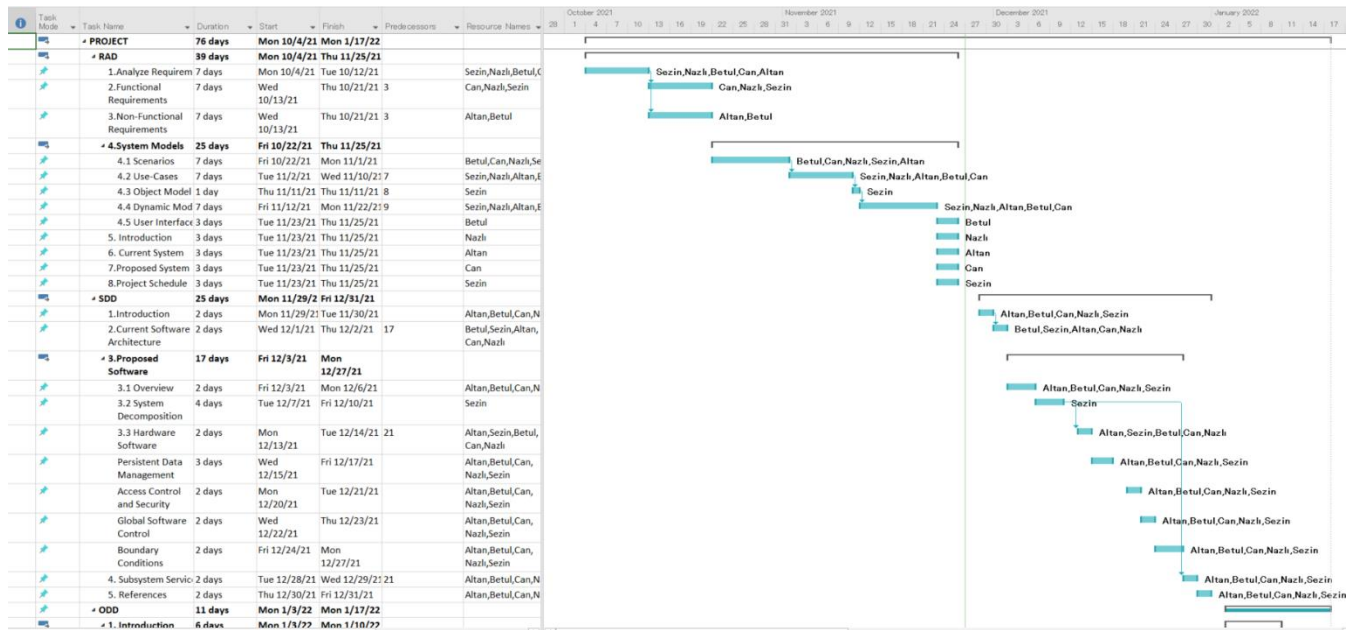
My Addresses

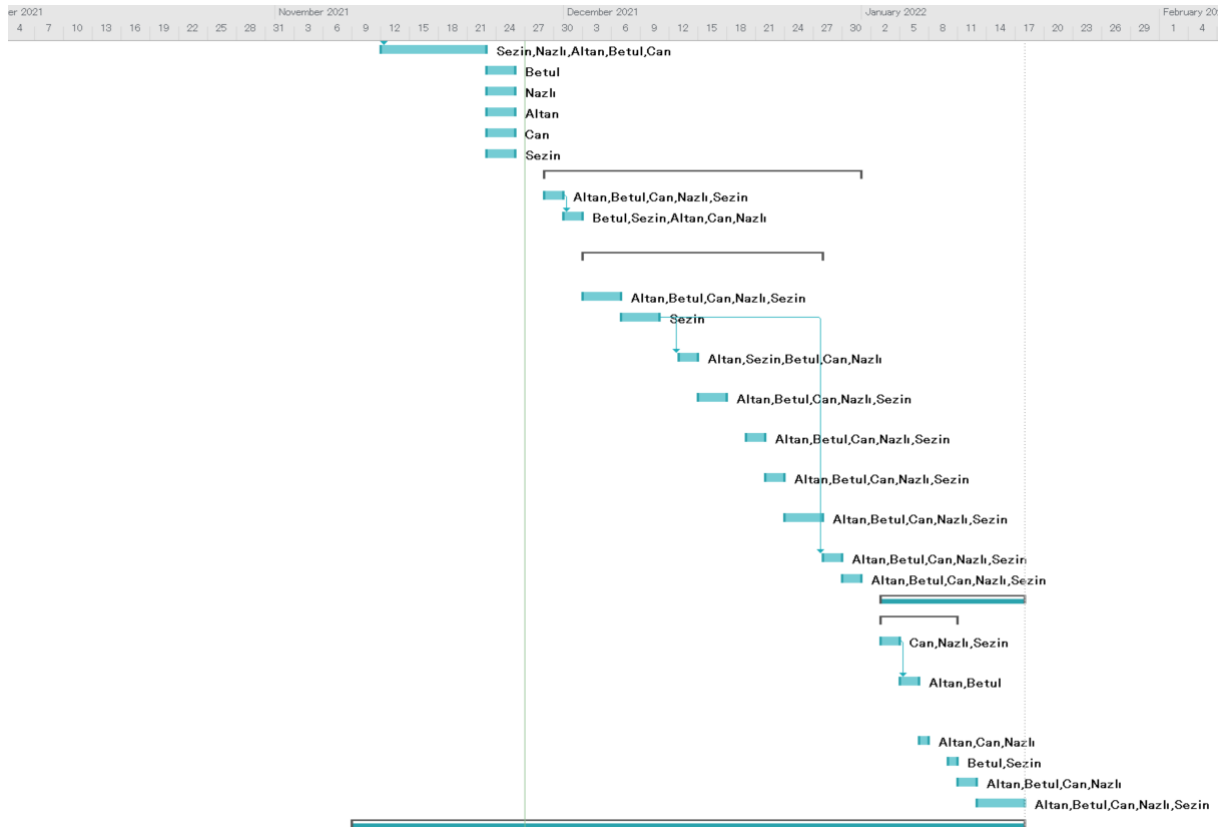
Address-1

Address Description

Create Address

3.5. Project Schedule





4. Glossary

Use Case: A specific situation in which a function could be used.

Entry Condition: The condition needed to activate a function.

Exit Condition: The condition needed to consider a function successfully done.

Visitor: A user who is viewing the but didn't log in to an account yet.

User: A user who is viewing the RBS and logged into an account.

Event Holder: A person who create the events.

Admin/Administrator: A person who manages all content about the Biletixx (users, database, events etc.)

System: A controller which operates all of the functions in the background.

Sign in Form: A form that contains "username" and "password" fields which will be filled by the user who wants to log into an account.

Sign up Form: A form that contains "first name", "last name", "username", "e-mail", "password" and "re-enter password" fields which will be filled by the user who wants to register.

Homepage: The initial screen of the Biletixx.

Profile: A page that contains user's information.

Ticket: Ticket to be purchased for the event.

Forgot Password Form: A form contains a text field for the recovery code and two password fields for the new password which need to be filled by the user who forgot his/her password in order to set a new password.

<Biletix>

Change Password Form: A form contains three password fields (one for the current password, two for the new password) which need to be filled by the user who wants to change password.

5. References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. <https://www.biletix.com/anasayfa/TURKIYE/tr>