Online Ticket System

Requirements Specification and Analysis

<1.0>

<10.11.2019>

Team: Gizem Gümüşçekiçci, Dilara Ünbay, Özay Ezerceli, Mert Mısırlıoğlu

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc496873294)

[1.1. Purpose of the System 1](#_Toc496873295)

[1.2. Scope of the System 1](#_Toc496873296)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc496873297)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc496873298)

[1.5. Overview 1](#_Toc496873299)

[2. Current System 1](#_Toc496873300)

[3. Proposed System 1](#_Toc496873301)

[3.1. Overview 1](#_Toc496873302)

[3.2. Functional Requirements 2](#_Toc496873303)

[3.3. Nonfunctional Requirements 2](#_Toc496873304)

[Usability 2](#_Toc496873305)

[Reliability 2](#_Toc496873306)

[Performance 2](#_Toc496873307)

[Supportability 2](#_Toc496873308)

[Implementation 2](#_Toc496873309)

[Interface 2](#_Toc496873310)

[Packaging 2](#_Toc496873311)

[Legal 2](#_Toc496873312)

[3.4. System Models 2](#_Toc496873313)

[Scenarios 2](#_Toc496873314)

[Use case model 2](#_Toc496873315)

[Object model 2](#_Toc496873316)

[Dynamic model 2](#_Toc496873317)

[User interface—navigational paths and screen mock-ups 3](#_Toc496873318)

[3.5. Project Schedule 3](#_Toc496873319)

[4. Glossary 3](#_Toc496873320)

[5. References 3](#_Toc496873321)

REQUIREMENTS ANALYSIS DOCUMENT[1]

# Introduction

## Purpose of the System

There are several different systems which provide online tickets to the user. Our purpose is to create a better solution which provides simpler and more effective management to the both users and the event hosts.

We are working to create a safe and reliable environment for the website. In this website users are able to find and buy tickets much easier. The interface of the website will be simplified to the needs of all kind of users.

## Scope of the System

Our website is an online ticket selling system in which consists of three different user types as the end users, visitors, admins and hosts as operators. The users can explore different events hosted by the operators, admins can manage host requests to create, edit, cancel functions about the events and operators can use these functions as stated above. Users can buy tickets with multiple choices to purchase. After the purchase the user can choose either to ship their tickets to a specified address or they can view their ticket online. The admins can choose if a host is eligible to become an operator upon hosts request.

## Objectives and Success Criteria of the Project

* The design of the website should be usable by any kind of end user such as old people.
* Users can request a refund upon cancellation of the event.
* Works swiftly and efficiently.
* Users information will remain private.
* Reliable information display for the end user.
* System functions work properly.

## Definitions, Acronyms, and Abbreviations

*Event* A visual and a written information in which enables users to view.

*Buy* Function to get ticket.

*Publish* Admins making events publicly available for the user on the website.

*Consumer* An individual using the system.

*Operator* An individual which hosts events through the system.

## Overview

The rest of the document contains further information about the functionality of the system. It contains differences from current working systems. It contains non-functional requirements as special properties to customize the system.

# Current System

No system is in place, but you can visit Biletix. If you examine Biletix website, you can get the idea of our system.

# Proposed System

## Overview

This section provides a general view on the functionality aspect of the system and it will be covered in two parts.

## Functional Requirements

* Users should be able to enter their first names, middle names and last names.
* Users should be able to enter their birth dates.
* Users should be able to login by using their emails and passwords.
* Users should be able to perform searches by event location, event date and event type.
* Users should be able to buy tickets.
* Users should be able to cancel their purchases.
* Users should be able to logout.
* Users should be able to select the ticket amount.
* Users should be able to select the ticket category.
* Only operator type of users should be able to add events.
* Only operator type of users should be able to cancel events.
* Only operator type of users should be able to perform changes on events.
* Only admins should be able to confirm event requests.
* Only admins should be able to decline event requests.
* The system should reserve tickets for 5 minutes upon purchases.
* The system should be able to send emails to the users.
* Only admins should be able to register operator type of users to the system.

## Nonfunctional Requirements

### 3.3.1. Usability

* Users who have basic knowledge of using computer can use this system.
* Users who are familiar with biletix.com user interface can be familiar with this website.
* Users who have valid email can be signup for website.
* System will provide status message to the users to inform them about their login state.
* System will provide verification email to users to complete registration to website.
* System will provide different searching filters to users make searching easier and specific.
* System will provide to review ticket for users who want to review his/her ticket.
* System will provide message to Operator about his/her event if it’s accepted or not.
* System will provide message to Operator about his/her request to make changes to the event if it’s accepted or not.
* System will notify Admin if he/she has any event in the waiting event list to be examined.
* System will provide notification message for the user who uses forgot password function to let her/him know that sent an email to reset password.
* System will provide status message for buying ticket if he/she is completed to buy the ticket.

### 3.3.2. Reliability

* Components of the project code will be tested alongside the implementation phase to ensure that they are functional.
* System will cover handling of time out, connection lost, wrong email or password exception cases.
* System should not lose %99 of data. (There is no system which is perfect.)
* System won’t need to be restarted in the event of a failure.

### 3.3.3. Performance

* Login form must be presented within 5 seconds.
* The Usermust receive the response email for activation within 2 minutes.
* The Operatormust receive a response in 30 seconds after the admin response.
* System must be running functional up to 1000 users.
* Users could wait up to 3 minutes while waiting completion of buying ticket in worse case.
* Login session expires after 24 hours of idle.
* System must respond to any kind of user within 3-5 seconds after clicking button which is showing message to that user.
* System must show results of searching to user in 5 seconds.
* System must create ticket in 3 seconds for the users who want to buy ticket.

### 3.3.4. Supportability

* It should be able to run on any pc that runs any web browser.
* System can provide user to reserve ticket for a while to buy it.
* Admin maintains the system.
* System can connect with such software like Spotify which shows the singer’s concerts. From singer’s profile, users can buy ticket for his/her concert through link to this website.

### 3.3.5 Implementation

* Project will be implemented in Django Web Framework which uses Python.
* Sublime editor and Atom IDE’s will be used for implementation.

### 3.3.6 Interface

* The System will be independent by itself which means does not interact with any existing systems.
* Data exporting and importing will be provided by Django Web Framework.
* Standards such used in Biletix.com should be supported by the system.

### 3.3.7 Packaging

* The System is not proper to be installed. Thus, in this system packaging is not a concern.

### 3.3.8 Legal

* System should be licensed as freeware (it’s free to use, but the creator holds the copyright).
* There can be liability issues such refund to user in system failure while processing of payment.

## System Models

### **3.4.1. Scenarios**

**1)**

*Scenario name* **CouldNotLogin (Gizem)**

*Participating actors*Alice:The user

*Flow of events*

*1)* Alice opens the website to buy a concert ticket and clicks the login button.

2)Alice fills up the login form and clicks the submit button.

3) System responds “wrong information failed to login” message.

4)Then Alice clicks the Forgot Password button and fills up the necessary form to reset her password.

5) Alice receives an email to reset her password.

6) Alice opens the email and clicks the link in the email.

7) Alice fills up the form in the links address and creates her new password and submits the form.

8)Alice uses her new password to login to the website and buys the concert ticket.

**2)**

*Scenario name* **RockConcert (Gizem)**

*Participating actors*Jack:The Operator

Daniel:The Admin

*Flow of events*

*1)*Jack owns a night club and he host events using the website.

2)Jack wants to host a rock concert in his night club.

3)Jack creates this rock event using the create event function. He completes and submits the event request.

4)Admin Daniel receives the rock concert request from Jack.

5)Then Daniel declines Jacks rock event request due to containing inappropriate content in the event.

6) After Daniel declines the request Jack receives the response from the admin Jack.

**3)**

*Scenario name* **AcceptChange (Gizem)**

*Participating actors*Jack:The Operator

Daniel:The Admin

*Flow of events*

1. Jack is hosting a Halloween party in his club, but he wants to change the opening time of the Halloween party.
2. Jack uses the Edit Event function to change the opening time of the event. He edits the event and changes the opening time of the party 9pm to 10pm.
3. Admin Daniel receives the editing request from Jack.
4. Then Daniel accepts and confirms the changes Jacks made.
5. After Daniel accepts the request, Daniel receives the response from the admin Daniel that Jack confirmed the changes about the Halloween party.
6. The opening time of the event changed 9pm to 10pm.

**4)**

*Scenario name***EditMyAccount (Gizem)**

*Participating actors*Alice:The user

*Flow of events*

1. Alice is exploring the website. She sees the Edit Account button in her profile, and she wants to change some information and wants to add a profile (address, birthday etc.) for her account.
2. Alice uses the edit account button in the website profile area to edit her account.
3. After she finished changing the information about her profile, she adds a profile picture and she clicks the save changes button.
4. After she clicked the button, she saw a response message that the changes made successfully.
5. Then Alice checks her profile again to make sure that the data is correct and continues to explore the website.

**5)**

*Scenario name* **KalbenGetsSick (Dilara)**

*Participating actor* Leyla:Operator *instances* Gizem:Admin

*Flow of events*

* + - 1. Leyla receives a call from Kalben’s manager. Kalben’s manager informs Leyla that Kalben got sick and she is unable to perform for two weeks. Leyla is asked to reschedule the next weeks concert. So, she activates the “Edit Event” function.
      2. Leyla changes the event date and event start time. Also, she adds a brief sorry message to the event description for Kalben’s fans. After checking the form, she submits it.
      3. Gizem, the Admin, receives the new Kalben event form.

**6)**

*Scenario name* **forgotToLogout (Dilara)**

*Participating actor* Emin:User *instances*

*Flow of events*

* + - 1. Emin buys a Manga ticket for his girlfriend and runs out of his house leaving his account logged in.
      2. Emin decides to stay at a friend’s house for a night. His account stays logged in for over 24 hours while he is away. So, the **SYSTEM** expires his session and logs him off.

**7)**

*Scenario name* **ShakiraDied (Mert)**

*Participating* Mert:Admin

*actor instances*

*Flow of events*

1. Mert is going home but suddenly he got a call from Shakira’s manager. Shakira dead by a car accident. So that he must delete the event from their site.
2. Mert find the event page on administrator panel and deleted the event.

**8)**

*Scenario name* **PostMaloneConcert (Özay)**

*Participating* Özay: User

*actors instances*

*Flow of events*

1. Özay learned that there will be Post Malone concert in Istanbul and he really likes his songs and stage performance. Thus, he decided to buy a ticket for his concert from online. He noticed that he didn’t have an account to enter the website where he’s going to buy a ticket. Thus, he activates “SignUp” function at the login page of the website.
2. Özay enters his name, surname, email and birthday and submit the signup form and waits for an acknowledgment.
3. Özay got an email to verify his email and login the website. He

clicks the link in the email and goes to website says, “email verification is completed”.  Then, to login the website he activates login function.

1. After he logged in the website he activates “Search Event”

function by typing Post Malone into search bar to see if his concert is still available to buy a ticket.

1. According to what Özay searched, available event list of Post Malone is showed to him. He found which concert he’s going to get a ticket and luckily, he had a chance to buy the last ticket for the Post Malone concert.

**9)**

*Scenario name* **ThraceFestival (Özay)**

*Participating* Ömer: Operator

*actors instances* Yeliz: Admin

*Flow of events*

*1*. Ömer is one of the organizators of Thrace Festival and he wants to add this festival as an event to website. He has an account as an operator. So, he logged into website and activates “AddEvent” function by clicking add event button.

2. Ömer fills out address of festival, date, time, celebrities that will be in the concert and rules in the form presented by the system. Then, He submits the form he filled and waits for an acknowledgment.

3. Yeliz is notified that an operator requested for an event to be accepted. She sees the waiting event submitted by Ömer and reviews event information to check if there is a problem about event.

4. Yeliz activates “AcceptEvent” function by clicking accept button to accept the event and make event to be added to available event list.

5.Ömer receives an acknowledgment that his request is accepted, and event is added successfully.

**10)**

*Scenario name* **TheManWhoWonTheLottery (Mert)**

*Participating* Mert : Admin

*actor instances*

*Flow of events*

* + - 1. Some guy wins the lottery and decided to be a big party organizer. He calls to Mert and said, “I want to be an operator in your site”. Mert sends him the agreement and he signed it.
      2. Mert opens the operator list and add operator account for him and send information to him. He is an operator now.

**11)**

*Scenario name* **Belieber (Mert)**

*Participating* Gizemsu : User

*actor instances*

*Flow of events*

1. Gizemsu is a big fan of Justin Bieber. And the online ticket site is announced new Justin Bieber concert in Istanbul. Gizemsu freaks out and runs to computer for buying a ticket.
2. Gizemsu had already an account at online ticket site. She directly goes to the event page and click “Buy Ticket”.
3. She chooses 2 ticket to buy and clicked next.
4. She choose online ticket and clicked next.
5. Gizemsu fill the payment details and clicked to complete order.
6. Gizemsu bought the ticket successfully and have unforgettable times with Justin Bieber.

### **3.4.2. Use-case definitions**

**1)**

*Use case name* **Logout (Mert)** *Participating* Initiated by**User** *actorsFlow of events*

1. The **User** activates the “Logout” function of the website.  
   1. **SYSTEM** responds by renewing the page and presenting the login form.

*Entry condition* The **User** activates the logout.*Exit conditions* The **User** is logged out of the **SYSTEM**.  
*Quality* Login form is presented within 5 seconds. *requirements*

**2)**

*Use case name* **ForgotPassword (Gizem)**

*Participating* Initiated by **User**

*actors*

*Flow of events*

*1.* **The user** activates the “Forgot Password” function by clicking Forgot password button in the website.

2. **System** responds by showing a form to the user.

3.**The user** receives the form sent by the system and fills it with e-mail information which used by the user to sign up to website and the user sends the form to the **system** by clicking the submit button.

4. **System** gets the form and checks the validity of the information and the **system** sends the reset password form link to the user’s email address.

5. **The User** clicks the link in the email sent by the system, ResetPassword function will be activated.

*Entry condition* **The User** failed to login to the website and activated the Forgot

Password action.

*Exit conditions* **The User** has received an email sent by the system to reset

password.

*Quality* **The User** will receive the response email in 2 minute.

*requirements*

**3)**

*Use case name* **ResetPassword (Gizem)**

*Participating* Initiated by **User**

*actors*

*Flow of events*

1.**The user** requests the reset password form from system by forgot password action or edit account action.

2.**The System** sends reset password form to the user.

3. **The** **User** fills up the reset password form and submits the form.

4. **The** **System** receives the form, processes the information and sends an acknowledgment to the user.

*Entry condition* **The User** receives the reset password form

*Exit conditions* **The User** receives an acknowledgement from the system.

*Quality* **The User** will receive the acknowledgment in 30 seconds.

*requirements*

**4)**

*Use case name* **DeclineEvent (Gizem)**

*Participating*

*actors*  **Initiated** by **The Admin**

*Flow of events*

1)**The Admin** receives an event request from **the** **Operator** and opens event approval page from **System**.

2) **System** responds by showing event approval page to **The Admin.**

3) **The Admin** selects the event and sees the details about the event then declines event by clicking the decline button in the event approval page.

4) **System** sends an acknowledgment to the operator to inform the operator about the event request has been declined by the admin.

*Entry condition* **The Admin** receives the create event request from the Operator.

*Exit conditions* **The Operator** receives an acknowledgement from the admin

*Quality* **The Operator** will receive a response in 30 seconds after the admin

*requirements* response.

**5)**

*Use case name* **DecideEventChanges (Gizem)**

*Participating* Initiated by **The Admin**

*actors*

*Flow of events*

1. **The** **Admin** receives the edit event request from the operator.

2)**SYSTEM** responds by presenting an event form to the **Admin**.

3**) The Admin** views theedited events page, confirms or declines the changes made by the Operator.

4**)The System** sends the response to the operator by the admin.

*Entry condition* **The Admin** receives the edit event request from the operator.

*Exit conditions* **The Operator** receives the response from the admin.

*Quality* **The Operator** will receive a response in 30 seconds after the

*requirements* adminresponses.

**6)**

*Use case name* **EditAccount (Gizem)**

*Participating* Initiated by **The User**

*actors*

*Flow of events*

1) **The user** activates the Edit Account button to change or add or remove information about the account and clicks the save changes button and completes the edit account function.

2)**System** receives the information and make the changes and send user an acknowledgment that the changes are made successfully.

3)**The user** receives the acknowledgment from the system.

*Entry condition* **The User** activates the edit account function.

*Exit conditions* **The User** receives an acknowledgment from the system.

*Quality* **The User** will receive a response in 5 seconds after clicking the

*requirements* save changes button.

***7)*** *Use case name* **TimeOut (Dilara)***Participating   
actorsFlow of events*

1. **User** is logged in for a long period of time and didn’t activate the “Logout” function of the website.

1. **SYSTEM** expires the session and informs the **User**.

*Entry condition* The **User** didn’t activate the logout.*Exit conditions* The session expires.  
*Quality* Session expires after 24 hours. *requirements*

**8)**

*Use case name* **EditEvent (Dilara)***Participating* Initiated by**Operator** *actors* Communicates with **Admin***Flow of events*

1. The **Operator** activates the “Edit Event” function of the website.  
   * 1. **SYSTEM** responds by presenting an event form to the **Operator**.
2. The **Operator** edits the form by changing or updating the event name, artist name, event date, event location, event hall, event description, ticket types, ticket prices and event start time. After the changes are completed, the **Operator** submits the new form.
   * 1. **SYSTEM** sends the new form to **Admin** for confirmation and notifies the **Operator**.

*Entry condition* The **Operator** hasclicked on the edit event.*Exit conditions* The **Admin** has received the form changed by the **Operator**.  
*Quality   
requirements*

**9)**

*Use case name* **SearchEvent (Özay)**

*Participating* Initiated by **User**

*actors*

*Flow of events* 1. The User activates **“Search Event”** function by writing the event s/he wants to search and clicking the search button in the page.

2. **System** responds by showing events to **The Operator**

    according to user’s search keys and/or filter.

3.**The User** sees the page of event list that is available to

preview or buy a ticket for an event among events in that list.

*Entry condition* **The User** is logged into the system.

*Exit conditions* Event list is displayed to **The User.**

*Quality* **The User** is acknowledged in at most 5 seconds.

*Requirements*

***10)***

*Use case name* **SignUp (Özay)**

*Participating* Initiated by **Visitor**

*actors*

*Flow of events* 1. **The Visitor** activates **“SignUp”** function by clicking the Signup

button at the login page.

2. **System** responds by presenting a signup form to **The**

**Visitor.**

3.**The Visitor** completes the form by typing his/her name, surname,

email and selecting his/her birthday. Once the form is completed,

 The User submits the form.

4. **System** receives the form and after validating his/her

email, system sends verification email to the user’s mail to

complete user’s signing process.

5.  **The Visitor** gets the verification email and by clicking the link in

the email

6. **System** directs user to page and shows the message that email verification is approved and signing process is completed.

*Entry condition* **The Visitor** is clicked the Signup button at the login page.

*Exit conditions* **The Visitor** is signed up.

*Quality* **The Visitor** gets verification email in 2 minutes.

*Requirements*

**11)**

*Use case name* **AddEvent (Özay Ezerceli)**

*Participating* Initiated by **Operator**

*actors*        Communicates with **Admin**

*Flow of events* 1. **The Operator** activates **“Add Event”** function by

clicking add event button inside the cite.

2. **System** responds by showing a form to **The Operator**.

3.**The Operator** receives the form sent by **System** and fills it with

information like event place, date, ticket price, type, artist who takes the stage and rules and send the form to the system by clicking the submit button.

4. **System** gets the form and adds the request to the queue that will be examined by **the Admin**.

5. **Admin** reviews the submitted event request form and accept the request by invoking **AcceptEvent** use case and acknowledges the event.

6. **System** receives the acknowledge and adds the event to the available event list. System notifies **The Operator** that adding event request is accepted by **Admin** and event is added to event list.

*Entry condition* **The Operator** is logged into the system.

*Exit conditions* **The Operator** has notified with a message that system has

                                 added the event to available event list.

*Quality* **Operator**’s submission is acknowledged in 5 seconds.

*requirements*

**12)**

*Use case name* **AcceptEvent (Özay)**

*Participating* Initiated by **Admin**

*actors*

*Flow of events* 1. **The Admin** opens the page of waiting events requested by **The**

**Operator** to be waiting for accepting by clicking “waiting events”

   button.

2. **System** responds by showing waiting events to **The Admin.**

3.**The Admin** selects the event and sees the details about

    event then accepts event by clicking accept button.

4. **System** receives acknowledgment and notifies **The Admin** that event is added to available event list.

*Entry condition* **Admin** is logged into the system.

*Exit conditions* **Admin** has notified that event is accepted and added to event list

                                by System.

*Quality* **Admin** has acknowledged in 3 seconds.

*requirements*

**13)**

*Use case name* **ViewEvent (Özay)**

*Participating* Initiated by **User**

*actors*

*Flow of events* 1. **The User** opens the page of event by clickingover the specific event to see details about that event such as artist name, event date, event location, event hall, event description, ticket types, ticket prices and event start time.

2. **System** responds by showing the page of event to **The User.**

*Entry condition* **User** is logged into the system.

*Exit conditions* **User** has seen the event page.

*Quality* **User** has acknowledged in 3 seconds.

*requirements*

**14)**

*Use case name* **ViewAccount (Özay)**

*Participating* Initiated by **User**

*actors*

*Flow of events* 1. **The User** opens the page of account by clickingto “MyAccount” button to see tabs about his/her account such as profile, addresses and tickets and by selecting between one of these three tabs user can see details about what the tab related with (Profile has information such as name, birthday, phone number. Addresses has user’s addresses. Tickets has user’s tickets since the account was created).

2. **System** responds by showing the page of account to **The User.**

*Entry condition* **User** is logged into the system.

*Exit conditions* **User** has seen the account (profile) page.

*Quality* **User** has acknowledged in 3 seconds.

*requirements*

**15)**

*Use case name* **PreviewTicket (Özay)**

*Participating* Initiated by **User**

*actors*

*Flow of events* 1. **The User** opens the page of my tickets by clickingover the “My Tickets” tab at the account page to see the ticket details (artist name, event date, event location, event hall, event description, seat no, ticket types, ticket prices and event start time) he/she bought.

2. **System** responds by showing the ticket page to **The User.**

*Entry condition* **User** is logged into the system.

*Exit conditions* **User** has previewed the ticket.

*Quality* **User** has acknowledged in 3 seconds.

*requirements*

**16)**

*Use case name* **BuyTicket (Mert)**

*Participating* Initiated by **User**

*actors*

*Flow of events*

1. The **User** activates “Buy ticket” function by clicking

                                    Buy ticket button inside the cite.

  2.**System** sends the ticket details on the ticket review page. Creates a ticket buy session.

1. **User** confirm and clicks next button.

5.**System** receives this form and sends the payment form.

6. **User** fills the payment form with credit card or money order option.  Clicks complete order button.

    7.**System** receives this payment information and sends to the bank for processing. If the bank approvals the payment

**System** creates a ticket object which is binded with this user.

If the bank does not approval the payment,

**System** create PaymentFail exception.

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

*Exit conditions* **User** has received a message shows buying ticket is completed.

*Quality* **System’s** creates the ticket is in 3 seconds.

*Requirements*

**17)**

*Use case name* **AddOperator (Mert)**

*Participating* Initiated by **Admin**

*actors*

*Flow of events*

1.**The Admin** activates “AddOperator” function by clicking Add operator button inside the cite.

2. **System** responds by showing a form to the Admin.

3.  **The Admin** receives the form sent by system and fills it with

information (email, password) and send the form to the system by clicking the submit button.

4. **System** adds the operator.

*Entry condition* **Admin** authorization is required

*Exit conditions* After adding operator operation System redirect Admin to Operators Page.

*Quality* **The Admin’s** adding operator feedback time is in 3 seconds.

*Requirements*

**18)**

*Use case name* **CancelEvent (Mert)**

*Participating* Initiated by **Admin**

*actors*

*Flow of events*

1. **The Admin** activates “DeleteEvent” function by clicking delete event button inside the cite.

2. **System** responds by showing a confirmation page.

3.**The admin** confirms.

4. **System** deletes the event.

*Entry condition* **Admin** authorization is required

*Exit conditions* After delete operation System redirect Admin to Events Page.

*Quality* **The Admin’s** deleting feedback time is in 3 seconds.

*Requirements*

**19)**

*Use case name* **Login (Mert)**

*Participating* Initiated by **User**

*actors*

*Flow of events*

1. **The User** activates “Login” function by clicking login button inside the cite.

2. **System** responds by showing a form to the User.

3.**The Operator** receives the form sent by system and fills it with information (email, password) and send the form to the system by clicking the submit button.

4. **System** gets the form and validates.

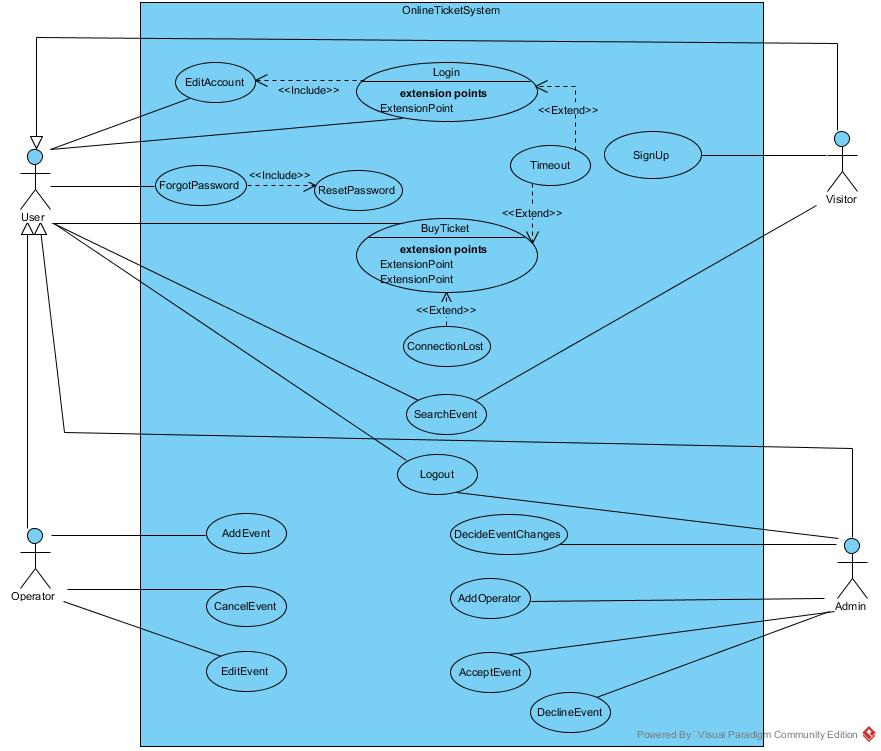
*Entry condition* **User** is signed up to the website.

*Exit conditions* After validation System redirect User to home page.

*Quality* **The User**’s login feedback time is in 3 seconds.

*requirements*

*3.4.2.1 Use-Case Diagram*

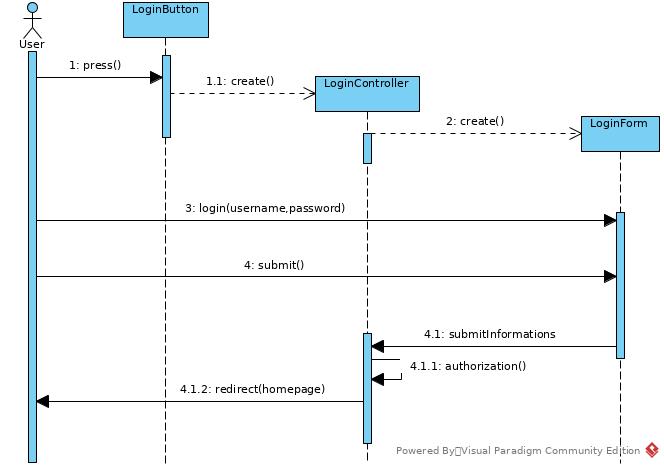
**

### 3.4.3. Object model

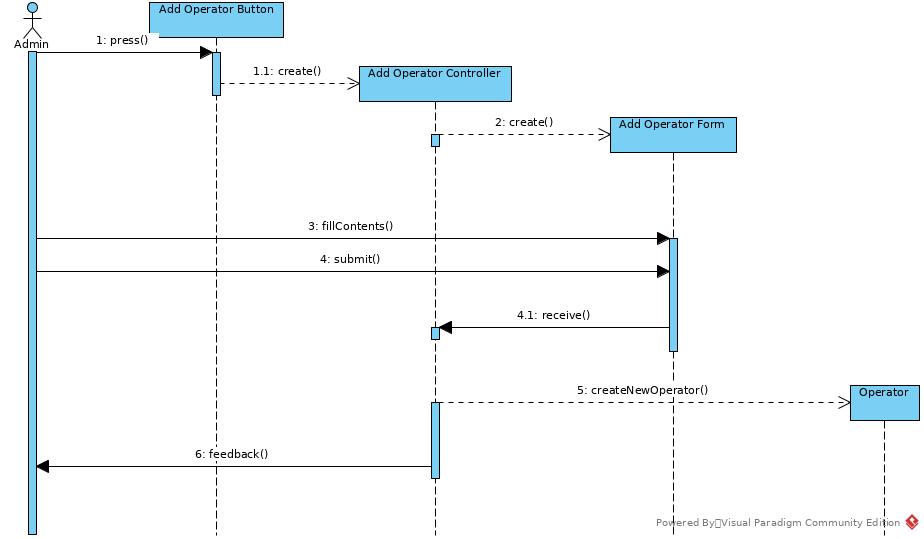
### 

### 3.4.4. Dynamic model

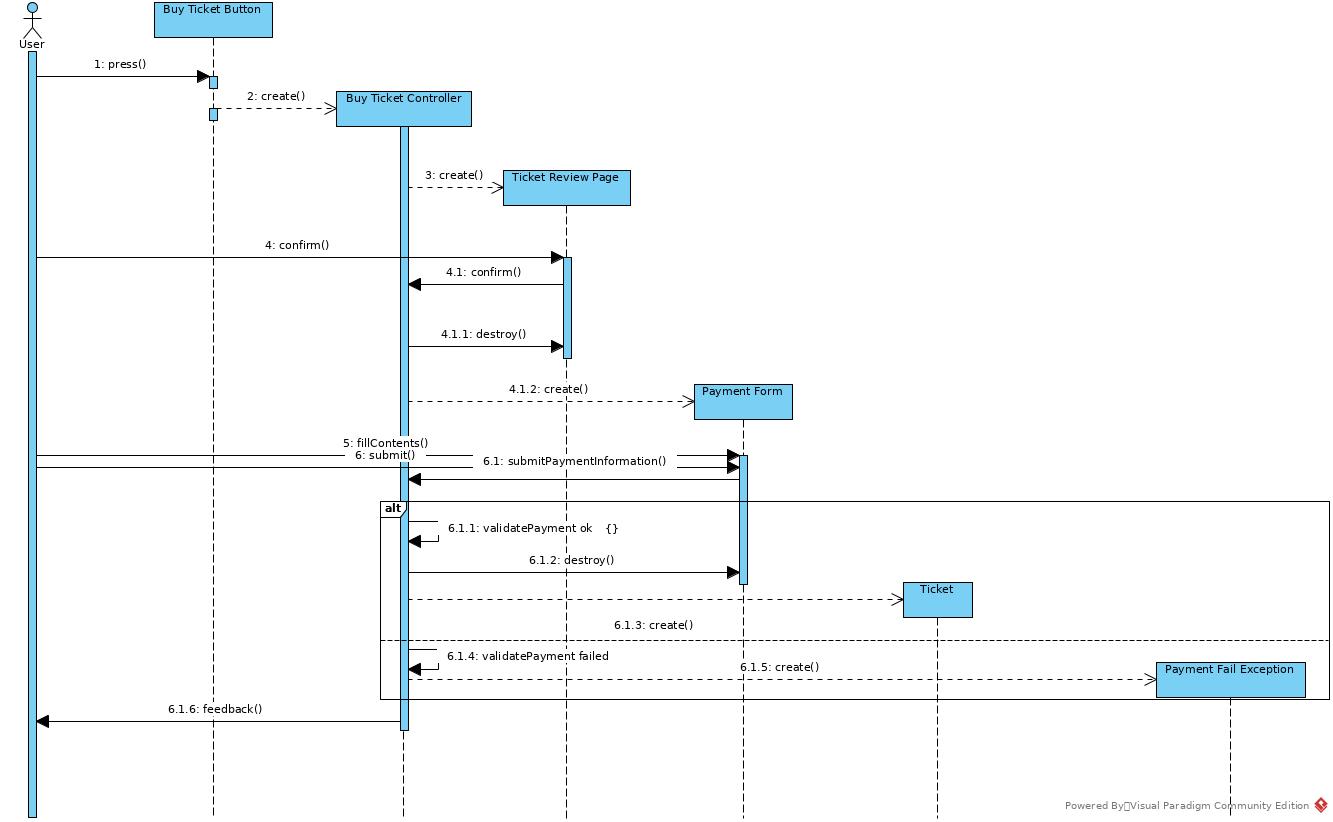
Login Sequence Diagram



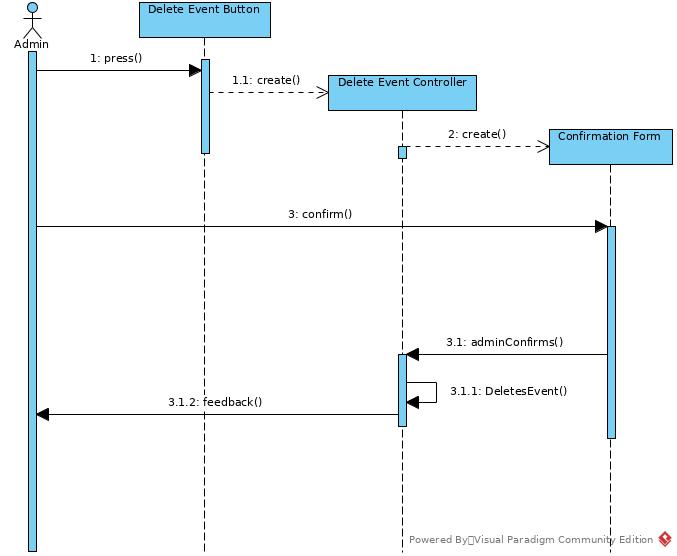
Add operator sequence diagram



Buy Ticket sequence diagram



Delete Event sequence diagram



## Project Schedule

Prepare Gannt Chart, and add it to this section.

# Glossary

**Admin**: One of the user types that has authorization to accept and decline events. Admin can add Operator to the system. Also, can accept event changes request.

**User**: One of the user types that already signed up and can search event and buy ticket.

**Operator**: One of the user types that can add, cancel and edit event. This user type can be only added to system by admin.

**Visitor**: One of the user types that needs to be signed up to login and buy ticket. Visitor can surf and search event in the website.

**Ticket**: Object that can be bought by users to go to event which ticket belongs to and has information such as event name, date, starts and ends times, place, seat number.

**Event**: Object that is created by Operator type of user and can be searched, viewed by any type of user.

**Stage**: Place that where event is happened.

**Form**: Object that is presented to different type of users whether who wants to sign up, add event, add operator, buy ticket or edit account and event.

**Request**: Sent by Operator to add or edit event to Admin to decide whether it’s accepted or declined.

**Website**: System that users can be able to use functions that website has.

**Concert, Theatre, Sport**: Type of event that is related with this area.

# References

## Bernd Bruegge, Allen H. Dutoit, Object Oriented Software Engineering Using UML, Patterns and Java, 3rd ed., Prentice Hall

## Biletix.com

## Visual Paradigm

## Balsamic