

# **EVENT IT!**

## **- AN EVENT REPOSITORY**



**OOAD PROJECT ITERATION 1**

**SUBMITTED BY**

**TEAM: Group9.getEvent()**

**Date: 02/22/2016**

**Team Members:**

Niveditha Varadha Chandrasekaran

Naveenraj Palanisamy

Shravya Kuncha

Navya Vandanapu

Namrata Singh

## INDEX

1. Introduction	3
1.1 Vision	
1.2 Overview	
2. Project scope for iteration 1	3
3. Domain Model	4
4. Design Specification	5
4.1 Functional requirements	
4.2 Non Functional requirements	
4.3 Constraints and Limitations	
5. Use Case Diagram	6
6 Use Case Text	7
6.1 Create Event	
6.2 Manage Worklist	
6.3 Browse Events	
7.System Sequence Diagram	9
7.1 Main Scenario	
8. Interaction Diagrams	10
8.1 Create Event	
8.2 Manage Worklist	
8.3 Browse Events	
9.Operation Contract	13
10. Supplementary Specification	13
11. Testing	13
12. UI Screenshots	15
13. Plan for Iteration-II	18
14. Glossary	19

## **1. INTRODUCTION**

### **1.1 Vision**

To create a web application – EventIt!, a one stop event repository that helps people to know about the various events happening in and around their neighborhood. It also serves as a platform for the event hosts to promote their ticket free events.

### **1.2 Overview**

The following functionalities are included in this system.

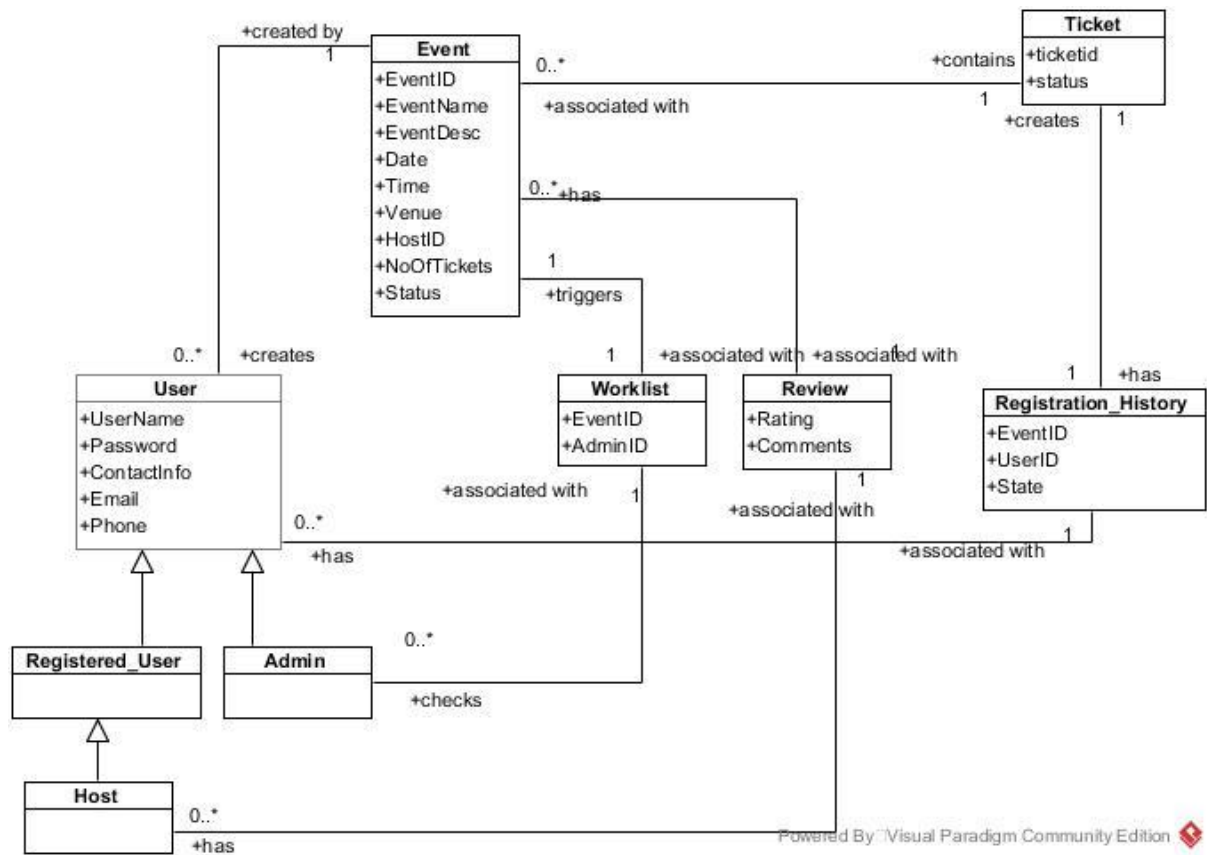
- Provides a platform to create events and promote them on the website
- Helps people to get to know the various events taking place
- Allows users to browse the events based on different categories
- Provides login to users to host or to register for events
- Allows users to reserve tickets for events based on the availability
- Allows the event hosts to update or delete events
- Allows users who have reserved tickets to rate the events once the event has happened
- Ensures that notifications are sent for every update
- Eliminates any inappropriate content present

## **2. PROJECT SCOPE FOR ITERATION 1**

The following core functionalities will be addressed in this iteration.

- Creating the events and submitting it for approval.
- Management of worklist by the admin to approve or reject the event publish request.
- Browsing the events that are approved by the admin and published on the web site.

### 3. DOMAIN MODEL



## 4. DESIGN SPECIFICATION

### 4.1 Functional requirements:

- Admin should be able to approve or reject event publish requests.
- Registered users should be able to create the events and submit them to the admin for approval.
- Registered users should be able to browse events, reserve tickets if needed and rate the events.
- General users should be able to browse events and sign up for the website if needed.

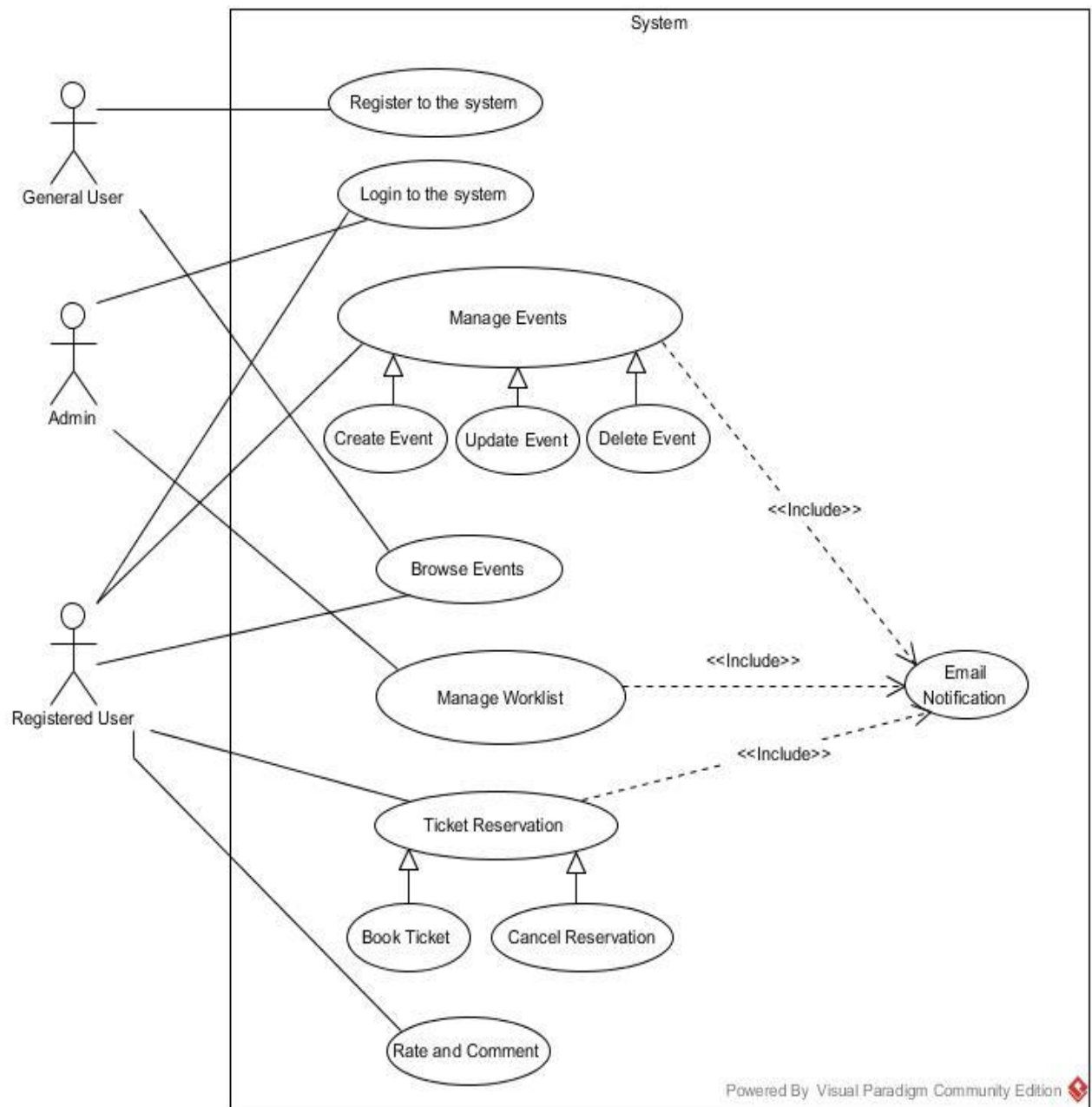
### 4.2 Non Functional requirements:

- *Usability:* This application provides usability by having a user friendly user experience. As the application is hosted in the web, anyone with Internet can access the website.
- *Security:* The application is password restricted for registered users and admin, so that general users and other anonymous users will not be able to see the controls that are available for the admin and registered users.
- *Deployment:* The application is deployed in web to get worldwide access.
- *Changeability:* The application changes its layout based on the screen size.

### 4.3 Constraints and Limitations

- The application will not let any user to create and promote their events until they sign up for the website.
- The application will not let any registered users to modify or delete the events which are not created by them.

## 5. USE CASE DIAGRAM



## 6. USE CASE TEXT

**ACTORS:** General User, Registered User, Admin

**SYSTEM:** The EventIt System

### 6.1 USE CASE UC1: CREATE EVENT

**Scope:** EventIt application

**Level:** User goals

**Primary Actor:** Registered user

**Stakeholders and Interests:**

**Registered user:** Wants to create and promote their events. They also want to know about the upcoming events and book tickets.

**General user:** They can know about the events happening.

**Preconditions:** User is registered to the system with a username and password.

**Post conditions:** An event is created and a publish request is sent to the admin for approval.

#### Main Success Scenario (Basic Flow):

1. Login to the system using username and password.
2. Click on create events to retrieve the event creation form.
3. Enter the event details such as event name, description, date, time, venue, total number of tickets in the event creation form.
4. Click on submit to send event publish request to the admin.
5. Send email notification to the admin about the event publish request and update the admin worklist.
6. Send email confirmation to the user saying that a publish request has been sent to the admin and will be notified once a decision is made.

#### Extensions (or Alternative Flows):

- 4a. If the event name given is already present.
  1. Prompt the message to the user about the duplicate event name.
    - 1a. If the user still wants to go with the same event name, then submit the publish request to the admin.
    - 1b. If the user decides to change the event name given, then do not submit the publish request.
- 4b. If the admin creates an event and clicks on submit.
  1. The event is auto approved by the system and it is published on the site so that it can be browsed by other users visiting the website.

## 6.2 USE CASE UC2: MANAGE WORKLIST

**Scope:** EventIt application

**Level:** User goals

**Primary Actor:** Admin

**Stakeholders and Interests:**

**Registered user:** Wants to create and promote their events. They also want to know about the upcoming events and book tickets.

**General user:** They can know about the events happening.

**Preconditions:** events are created by the registered users and event publish request is submitted.

**Post conditions:** Event will be published on the site once it is approved by the admin and the user is notified about the approval.

### Main Success Scenario (Basic Flow):

1. Log in to the system using admin username and password.
2. Click on worklist to get the list of all event publish requests to be approved.
3. Click on the event to be approved to get the event details.
4. Go through the event details and approve the event by clicking "Approve Event".
5. Send email notification to the user about the event approval.
6. Publish the event on the site so that it can be browsed by other users visiting the website.

### Extensions (or Alternative Flows):

- 4a. Admin goes through the event details and reject the event by clicking "Reject Event"
  1. Send email notification to the user about the rejection of their event publish request.
  2. Do not Publish the event on the site.

## 6.3 USE CASE UC3: BROWSE EVENTS

**Scope:** EventIt application

**Level:** User goals

**Primary Actor:** General user, Registered user

**Stakeholders and Interests:**

**Registered user:** Wants to create and promote their events. They also want to know about the upcoming events and book tickets.

**General user:** They can know about the events happening.

**Preconditions:** events are created by the registered users, approved by the admin and is published on the site.

**Post conditions:** List of all upcoming events is retrieved.



### Main Success Scenario (Basic Flow):

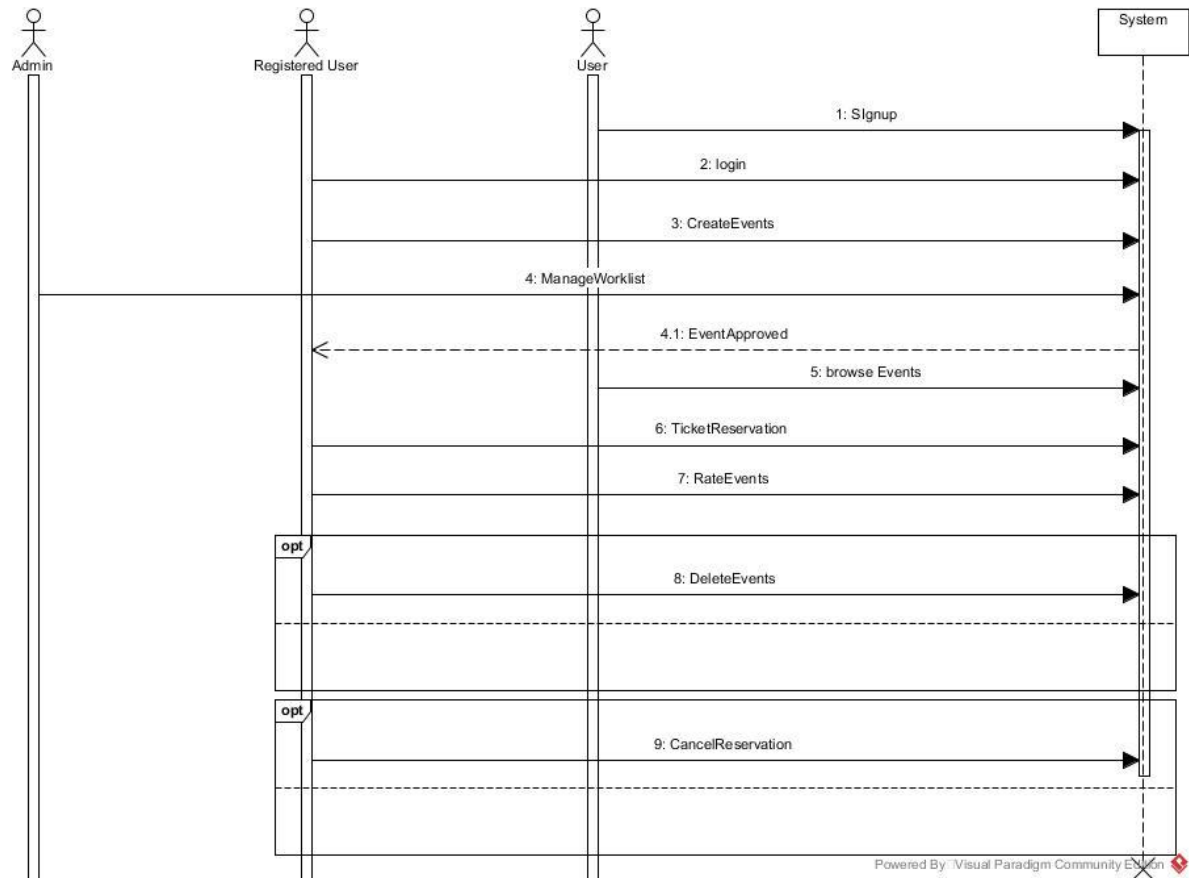
1. Visit the web site and click on browse events.
2. Retrieve the list of all upcoming events and display it to the user.
3. Click on the event of your interest.
4. The event page is retrieved with the event details.

### Extensions (or Alternative Flows):

- 2a. If the list of upcoming events retrieved has no events.
  1. Display the message "No Upcoming events".

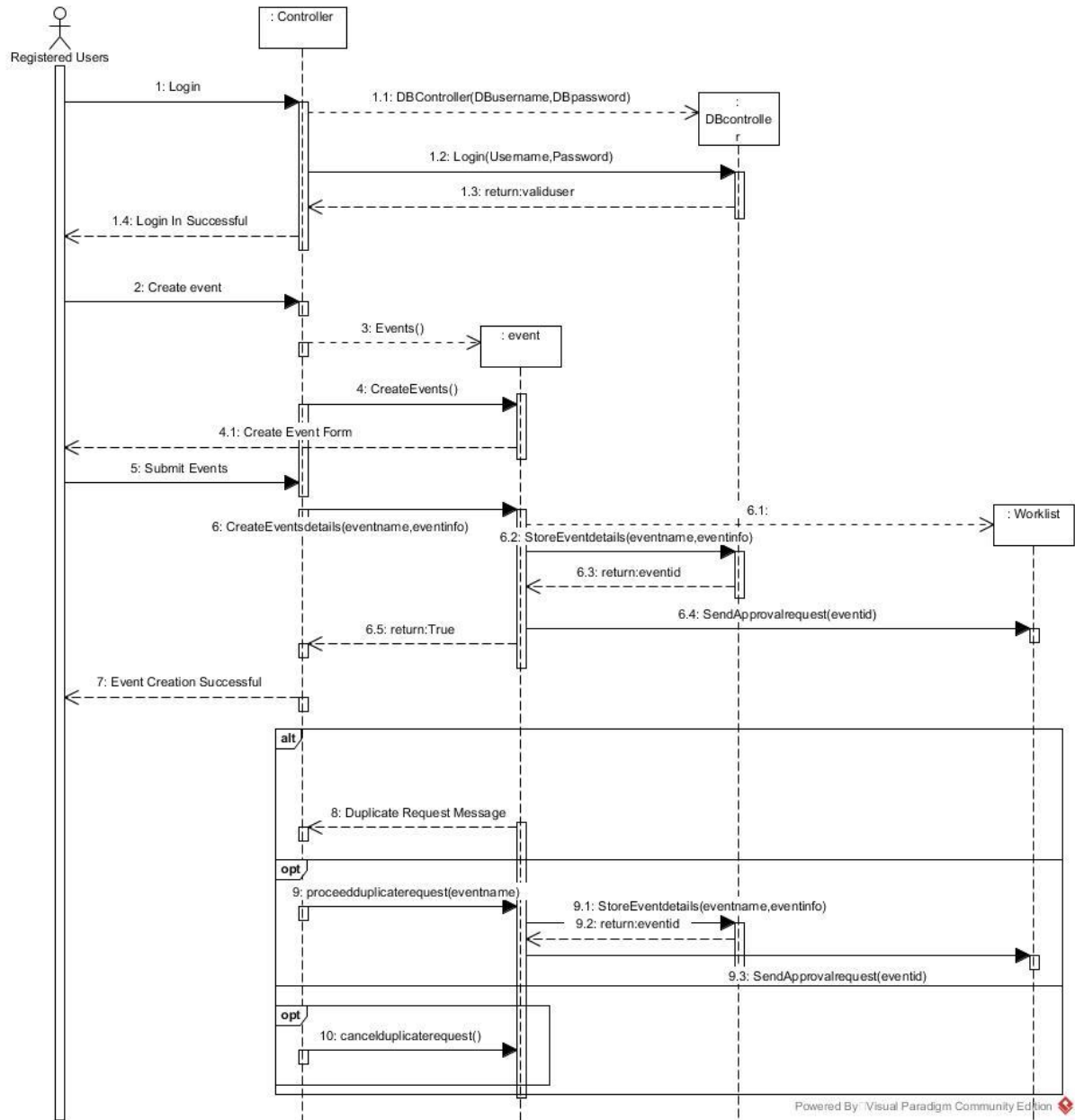
## 7. SYSTEM SEQUENCE DIAGRAM

### 7.1 MAIN SCENARIO

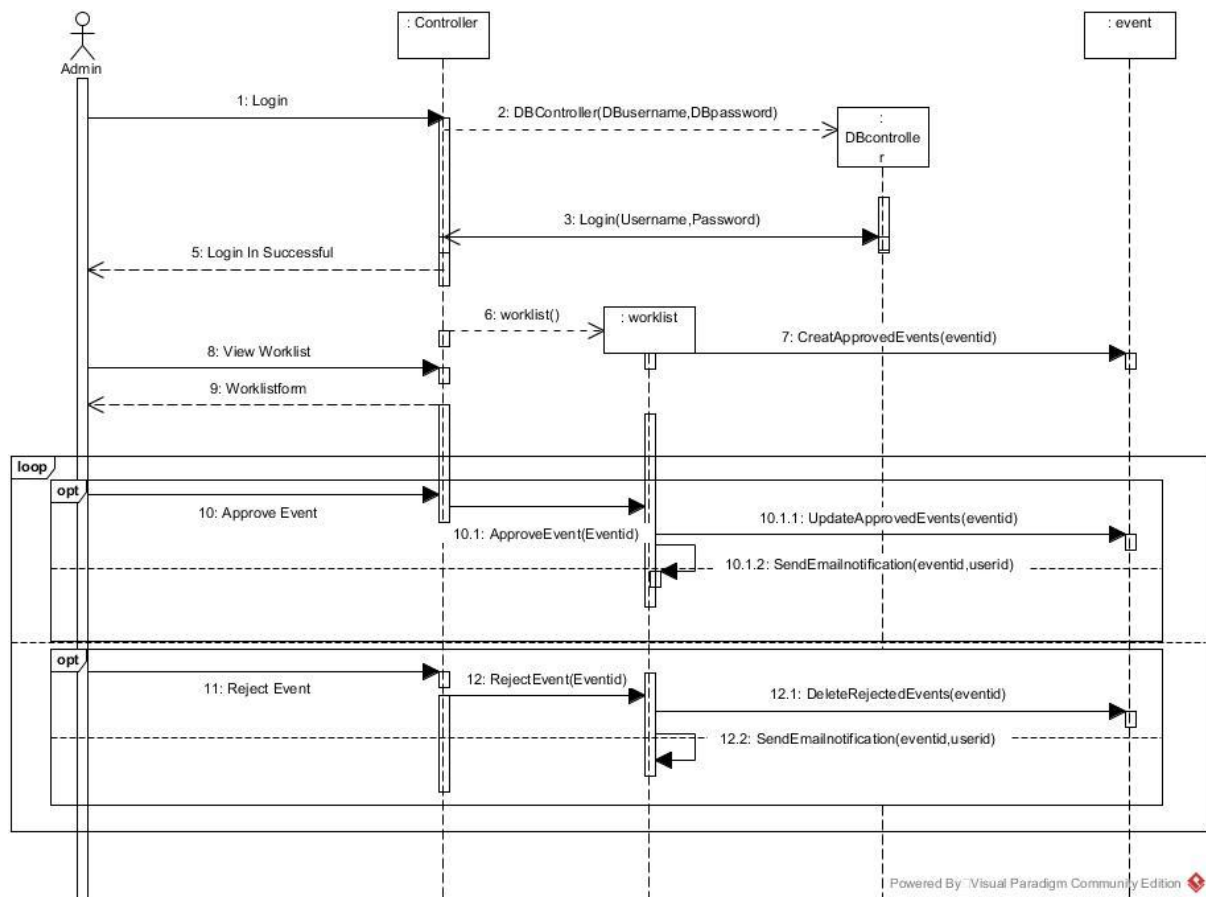


## 8. INTERACTION DIAGRAMS

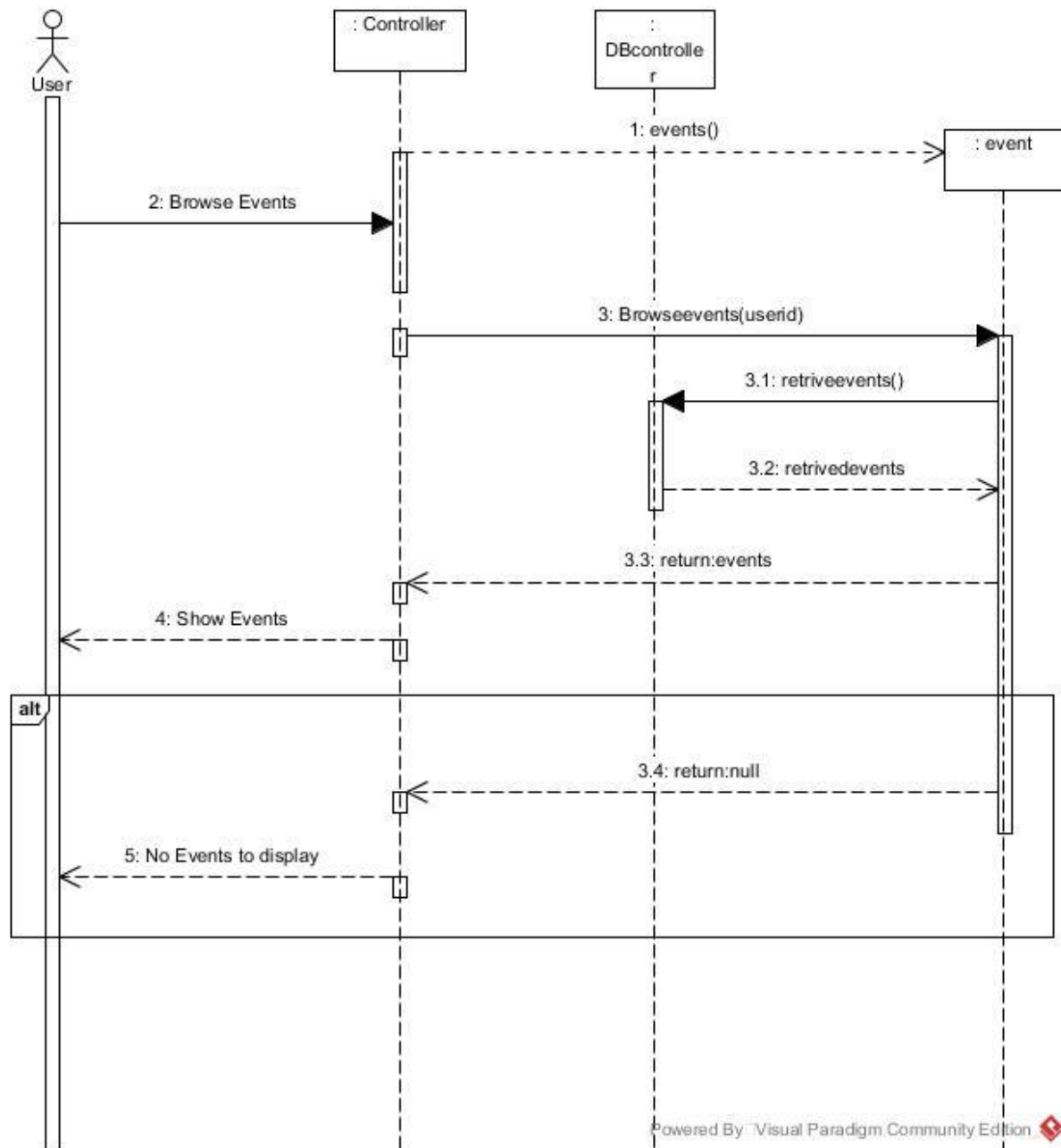
### 8.1 CREATE EVENT



## 8.2 MANAGE WORKLIST



### 8.3 BROWSE EVENTS



Powered By Visual Paradigm Community Edition

## 9. OPERATION CONTRACT

<b>Contract 1:</b>	Create Event
<b>Operation:</b>	EventCreate (Name: varchar, Description: varchar, Location: varchar, Date: date)
<b>Cross Reference:</b>	Event is sent for approval
<b>Preconditions:</b>	User is logged in
<b>Post conditions:</b>	Event instance EV is created Work List instance W is created EV and W are associated
<b>Contract 2:</b>	Manage worklist
<b>Operation:</b>	ManageWorkList( Name: varchar, Description: varchar, Location: varchar, Date: date)
<b>Cross reference:</b>	Use case: Approve/Reject event request
<b>Preconditions:</b>	Event instance and Work instance must be created
<b>Postconditions:</b>	Event object status is changed

## 10. Supplementary Specification

- This program must run on all the platforms.
- This program must run on Internet Explorer, Google Chrome, Safari and Firefox.
- Developer will give a manual that explains how to use this program.

## 11. Testing

Testing is done on the software to check if the quality is maintained and implementation is according to requirements stated earlier.

- Meets the requirements which guided design and development
- Works for correct inputs and handles incorrect inputs
- Get system response within time frame.
- Achieves the Stakeholder goal.

### 11.1 Test Plans

The following activities and test cases are used to check the quality of code and to map the implementation with the requirements of the system.

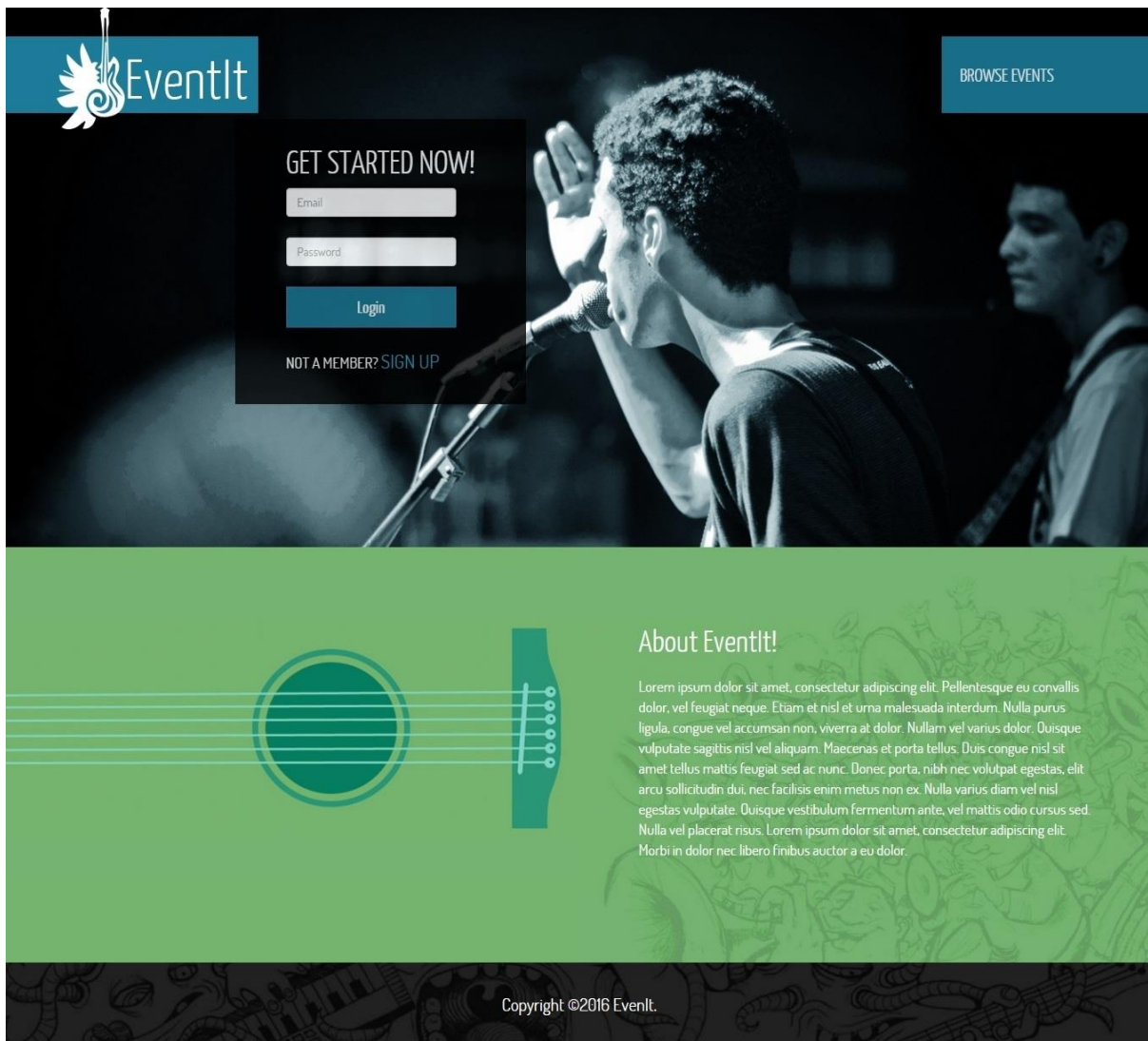
### 11.2 Unit Testing

- Unit testing is done for each use cases.
- On this iteration testing is done by giving basic inputs.

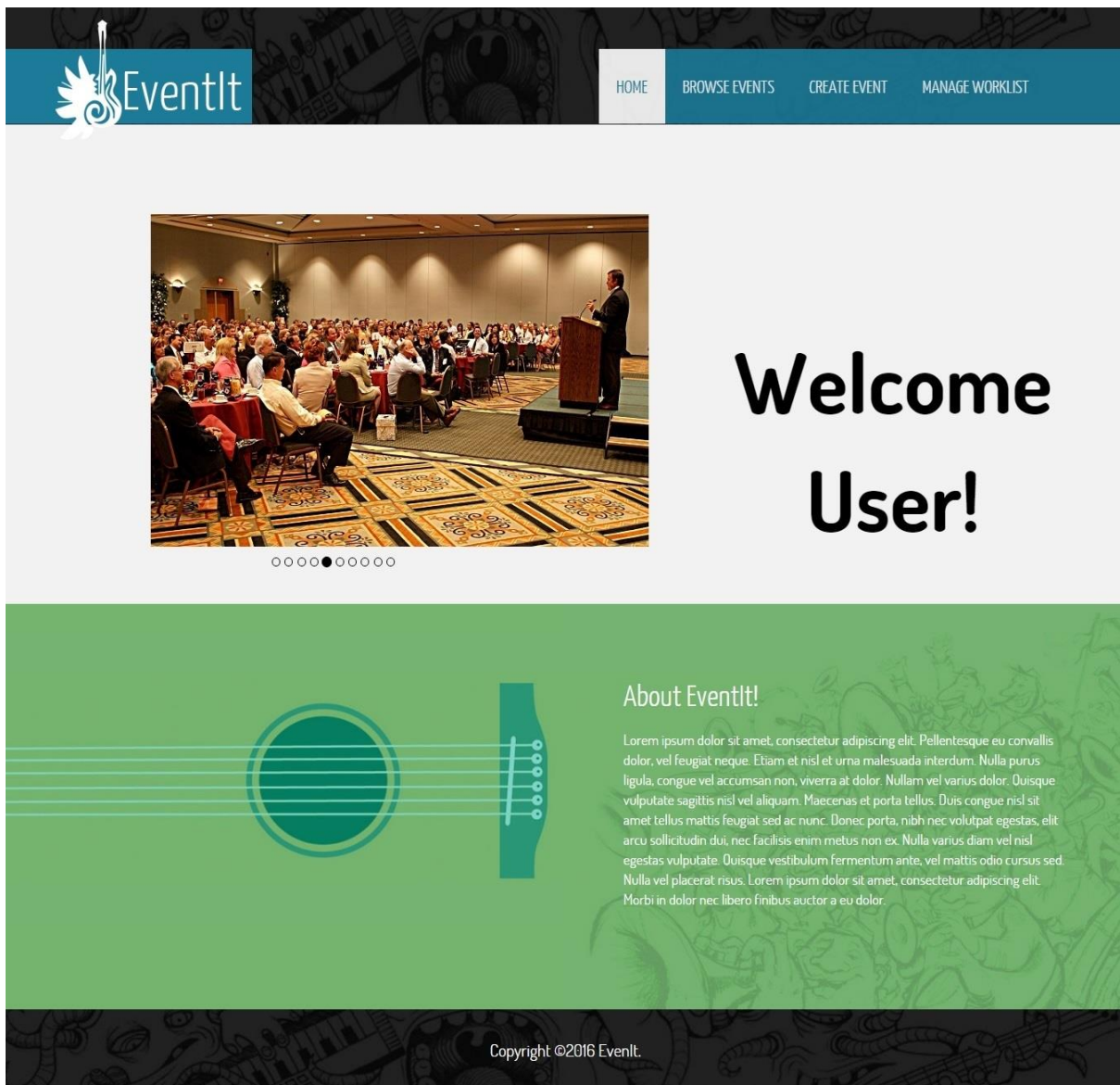
S.No	Test Case	Test Case Description	Expected Output	Observed Output
1	Login by using email id/user id	1.The user enters URL and clicks enter. 2.User provides login details and clicks login	The user is able to login in	The user is able to login successfully and home page is displayed
2	Sign up to the system	1.The user enters URL and clicks enter 2.User clicks sign up 3.Open Model page 4.User provides sign up details and clicks submit	The user is able to sign up	The user is able to sign up successfully and login in page is displayed
3	Create Event	1.The user enters URL and clicks enter 2.Open create event page 3.Provide event details 4.Clicks submit	The user is able to create event	The user is able to create event and event is sent for approval
4	Manage Worklist	1.The admin enters URL and clicks enter 2.Open worklist page 3.Clicks event to approval and clicks approve or reject	The admin is able to manage worklist	The user is able to approve event successfully and event is published.
5	Browse Events	1.The user enters URL and clicks enter 2.Click Browse events 3.Displays published events	The user is able to view events	The user is able to view event successfully

## 12.UI SCREENSHOTS

### Login Page




## Home Page





## Create Event

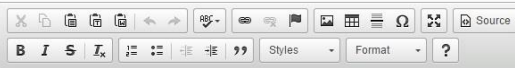
HOMEBROWSE EVENTSCREATE EVENTMANAGE WORKLIST

Create Event

Event Name:

Event name

Description:



Venue:

Address

Date:

Event date

Time:


Event time

Submit

Reset

Copyright ©2016 EventIt.

## Manage Worklist

HOMEBROWSE EVENTSCREATE EVENTMANAGE WORKLIST

Pending approvals

Big Texas Beer Fest 2016

Friday, Feb 26th 2016

Event host: Niveditha

View Event

Copyright ©2016 EventIt.

## Browse Events

## 13.PLAN FOR ITERATION-II

- Requirement analysis
- Ticket reservation module
- Cancel ticket reservation module
- Update event module
- Delete event module
- Refine the system
- Documentation and Report
- Backlogs from iteration 1

## 14. GLOSSARY

Glossary			
Revision History			
Version	Date	Description	Author
Iteration 1 document	2/22/2016	First draft to be refined in later iterations	Group 9
Definitions			
Terms	Definition and Information		
Admin	Admin is responsible for the credibility of the events being published on the website. Whenever a registered user creates an event, an event request is sent to the admin for approval, to check the authenticity of the event and its host, before being published on the site. This prevents the creation of fraudulent events on the site.		
Registered User	Registered user is the user who has signed up for the website. These users have the privilege to create events and promote them after the event is approved by the admin. They can also edit the event details or delete the event. Not only that, but they can also reserve tickets for the events they wish to attend, which are hosted by other users. They will also have permission to rate the events they have attended.		
General User	General users are the other users who visit the website. They can browse events based on different event types and look into the details but they will not be able to reserve tickets unless they sign up		
Username	User ID of the User to be used to login to the system.		
Worklist	The list of events to be approved by the admin to be published on the web site.		