**Name : Shweta Bangad CWID: A20392438**

Contents

[1. TOPIC: 2](#_Toc481519333)

[2. TEAM MEMBERS: 2](#_Toc481519334)

[3. PROJECT DESCRIPTION: 2](#_Toc481519335)

[Host your Events at Town Events: 2](#_Toc481519336)

[Description about the application: 3](#_Toc481519337)

[Login Credentials for the application: 3](#_Toc481519338)

[Specifications of the Project Implementation: 4](#_Toc481519339)

[Screenshots of the application: 7](#_Toc481519340)

[Screenshots of Database Table Structures: 22](#_Toc481519341)

[Executable Files: 25](#_Toc481519342)

# 1. TOPIC:

Host your event at **Town Event –** Event Management System

# 2. TEAM MEMBERS:

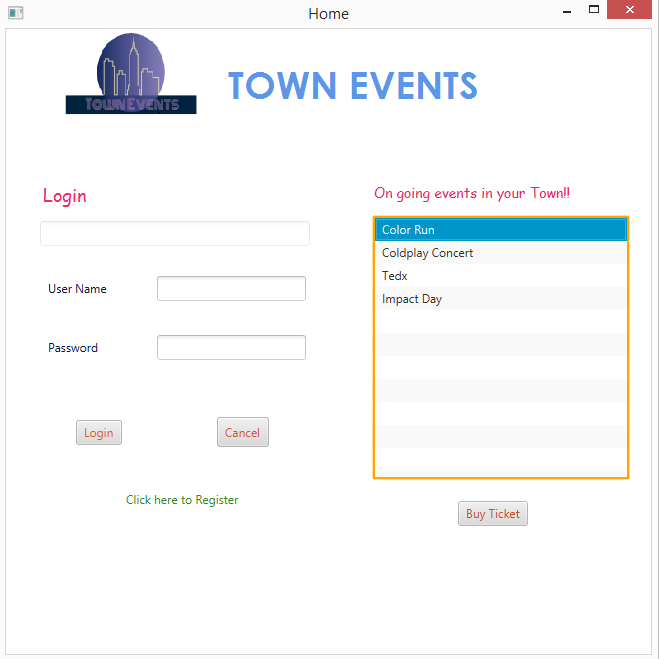
1. Shweta Bangad (A20392438)

2. Priya Mehta (A20391957)

# 3. PROJECT DESCRIPTION:

## Host your Events at Town Events:

Home Page of the application looks as below:



## Description about the application:

Town Events helps you to create events and sell tickets for the same.

1. The Town Events application has 4 modules: Login Module, Event Module, Buy Ticket Module and Payment Module.

* **Login Module** – This module allows to create new Registration, and Log In into the system.
* **Event Module** – This module allows to create, view, update and delete the event. (CRUD)
* **Buy Ticket Module** – This module allows customers to buy tickets for the event created.
* **Payment Module** – This module allows customers to make payment for the tickets purchased.

2. The application allows the Event Managers to Login and create Events hosted by them.

3. The Event Manager enters details about the event and the number of tickets he wants to sell.

4. This application has another role of Admin. He has privileges of deleting any event on request of the Event Manager.

5. Another role is of Customer who enters the number of tickets he wants to buy and makes payment for the same.

6. After the payment, ticket pass will be displayed to the customer.

## Login Credentials for the application:

**Admin Role**

Login Name: Admin

Login Password: \*Admin123

**Event Manager Role**

Login Name: Kathey

Login Password: Kathey123

Login Name: Hannah

Login Password: Hannah123

## Specifications of the Project Implementation:

**1. List of Entities:**

**Townevents\_user\_table**: This table stores details about the users of the system (Admin and Event Manager details)

**Townevents\_event\_bookings:** This table stores details about the events created.

**Townevents\_event\_tickets**: This table stores details about the number of tickets and fare for the event created.

**Townevents\_ticket\_booking:** This table stores details about the tickets bought by the customer.

**Townevents\_customer:** This table stores the customer details.

**Townevents\_event\_type:** This is a Reference table that stores types of events.

**2. List of Roles:**

|  |  |
| --- | --- |
| Role | Privileges |
| Admin | View Events  Delete Events |
| Event Manager | View Events  Create Events  Edit Events |
| Customer | Buy Tickets  View Event Details  Make Payment |

**3. CRUD operations performed by different Roles:**

For the extra credit work, we have implemented CRUD operations on total 6 Entities as shown below.

|  |  |
| --- | --- |
| Roles | Entities |
| Event Manager | **Entity 1:**  Insert into Townevents\_event\_bookings; **(C)**  Edit into Townevents\_event\_bookings; **(U)**  Retrieve from Townevents\_event\_bookings; **(R)**  Retrieve from Townevents\_event\_type; **(R)**  **Entity 2:**  Insert into Townevents\_event\_tickets; **(C)**  Edit into Townevents\_event\_tickets; **(U)**  Retrieve from Townevents\_event\_tickets; **(R)**  **Entity 3:**  Insert into Townevents\_user\_table; **(C)**  Retrieve from Townevents\_user\_table; **(R)** |
| Admin | **Entity 1:**  Retrieve from Townevents\_event\_bookings; **(R)**  Delete from Townevents\_event\_bookings; **(D)**  Retrieve from Townevents\_event\_type; **(R)**  **Entity 2:**  Retrieve from Townevents\_event\_tickets; **(R)**  Delete from Townevents\_event\_tickets; **(D)**  **Entity 3:**  Insert into Townevents\_user\_table; **(C)**  Retrieve from Townevents\_user\_table; **(R)** |
| Customer | **Entity 1:**  Insert into Townevents\_ticket\_bookings; **(C)**  Retrieve from Townevents\_ticket\_bookings; **(R)**  Delete from Townevents\_ticket\_bookings; **(D)**  **Entity 2:**  Insert into Townevents\_customer; **(C)**  Retrieve from Townevents\_customer; **(R)**  Delete from Townevents\_customer; **(D)**  **Entity 3:**  Retrieve from Townevents\_event\_bookings; **(R)**  Retrieve from Townevents\_event\_type; **(R)** |

**4. Coding Architecture:**

The architecture followed is MVC Architecture: Models, Views and Controllers

**Data Access Object (Dao package) –** We have created a separate package called dao. It contains all the data access methods. For every entity, separate DAO file is created.

dB connection used 🡪 **jdbc: mysql://www.papademas.net:3306/fp**

**Remaining code is written as per MVC Architecture.**

The source code is attached with this document.

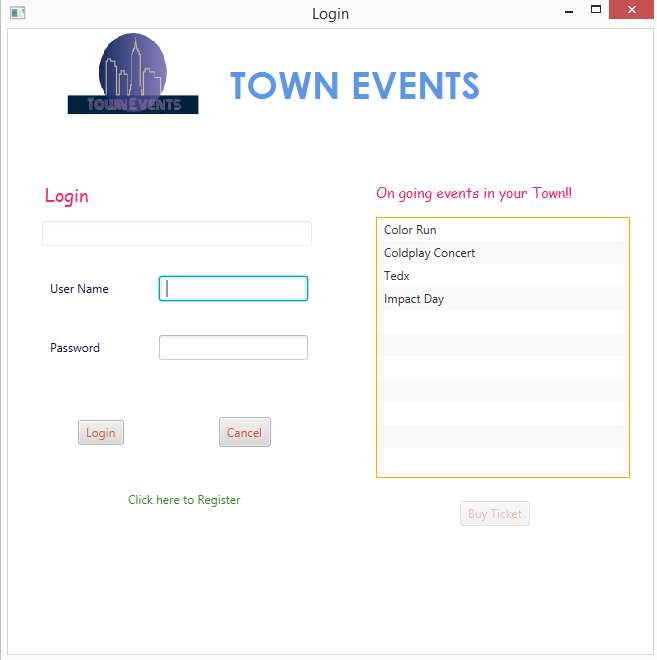
**5. Exception Handling:**

Validations are applied for all the fields in different modules. Different type of validations used are: Email Validation, Phone Validation, Character Input validation, Numeric Validation, Date Validations.

Exceptions are also handled using appropriate TRY CATCH blocks.

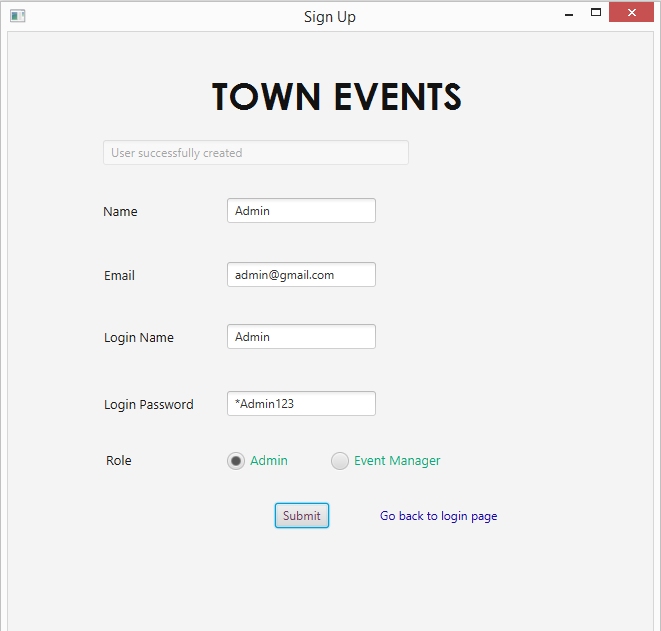
## Screenshots of the application:

1. **Login Screen:**

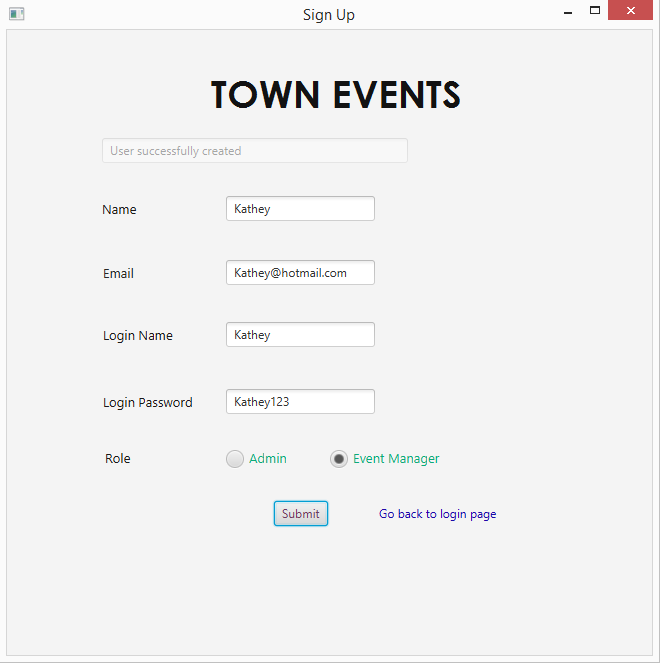


1. **Sign up Screen:**

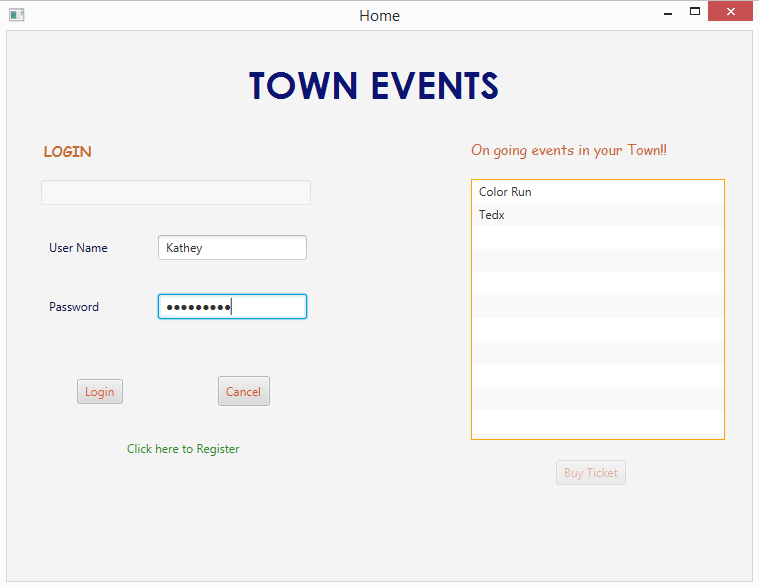
Admin:



Event Manager:

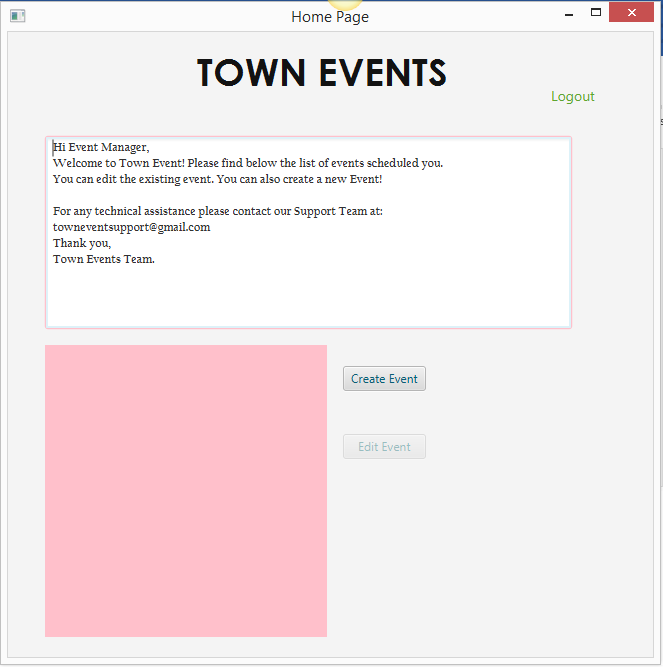


1. **Event Manager Login:**

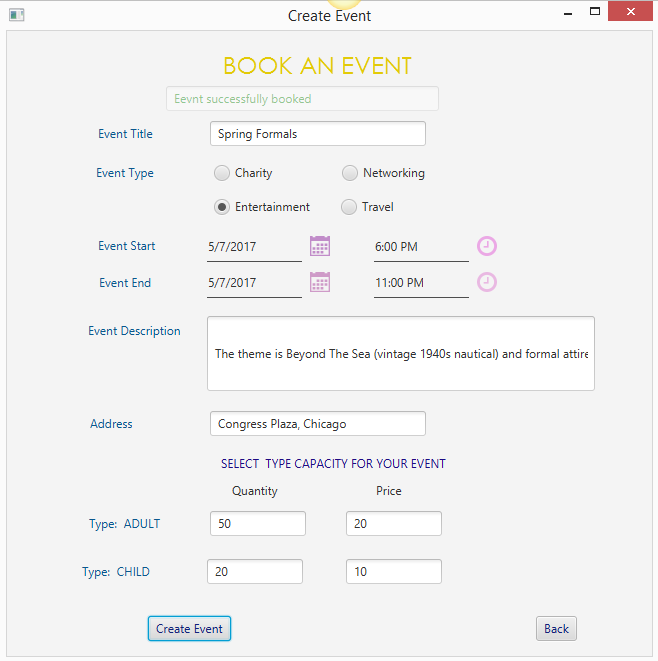


4. **Home page for event manager:**

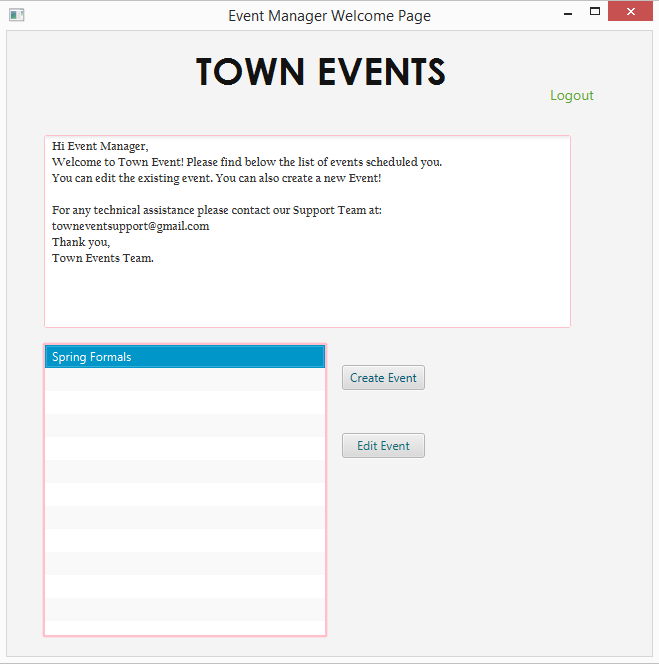
* Create Event
* Edit Event



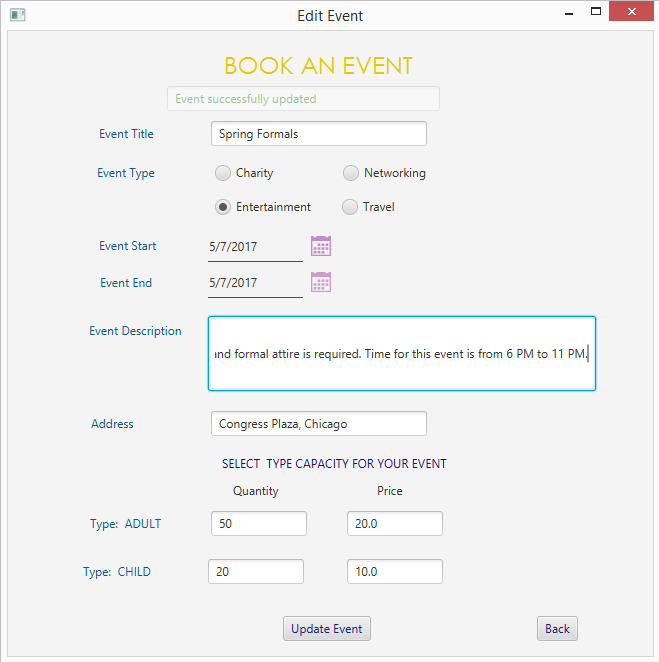
Create Event:



Edit Event:



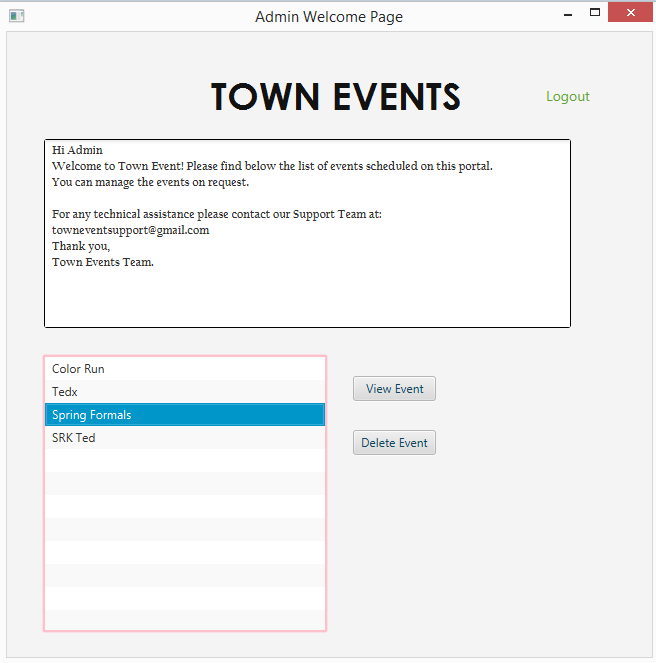
Here we have updated information in the event description fiels.



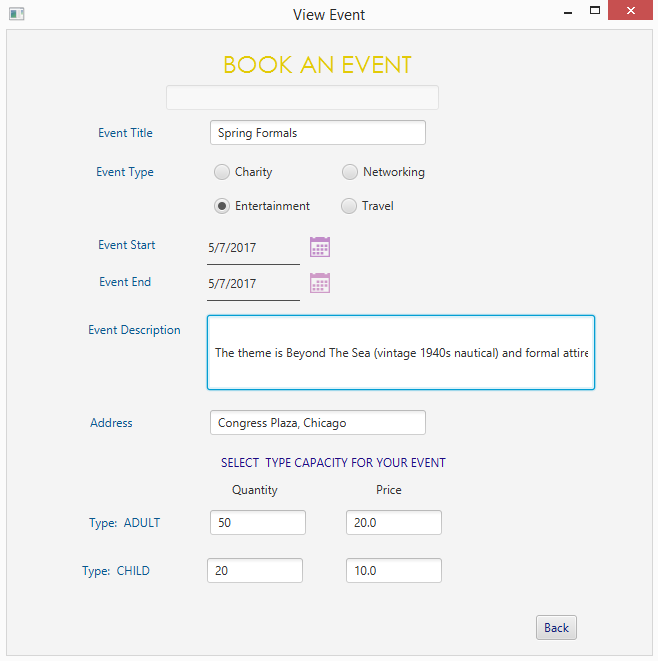
In the same way, other Event Managers can create their events and all the events will be listed in Admin Portal.

5**. Home Page for Admin:**

* View Event
* Delete Event

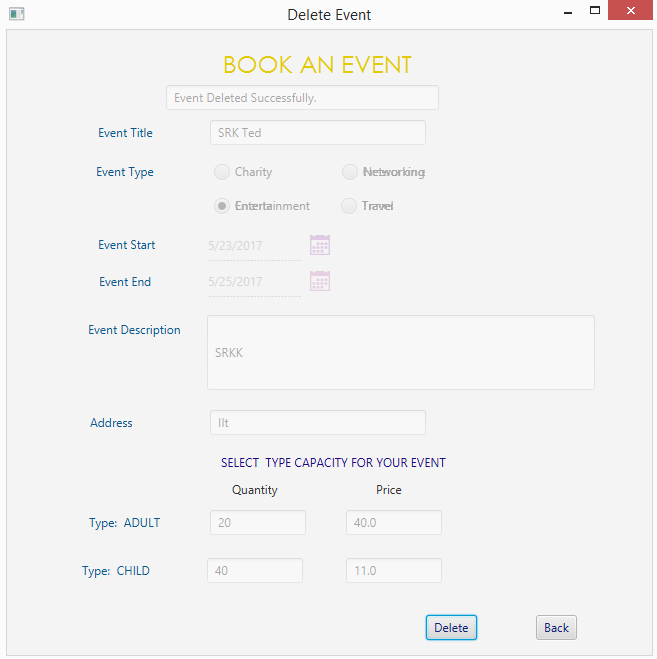


View Event:



Delete Event:

After selecting SRK TED from the list, we can delete the event. Screen shot of deleted event.

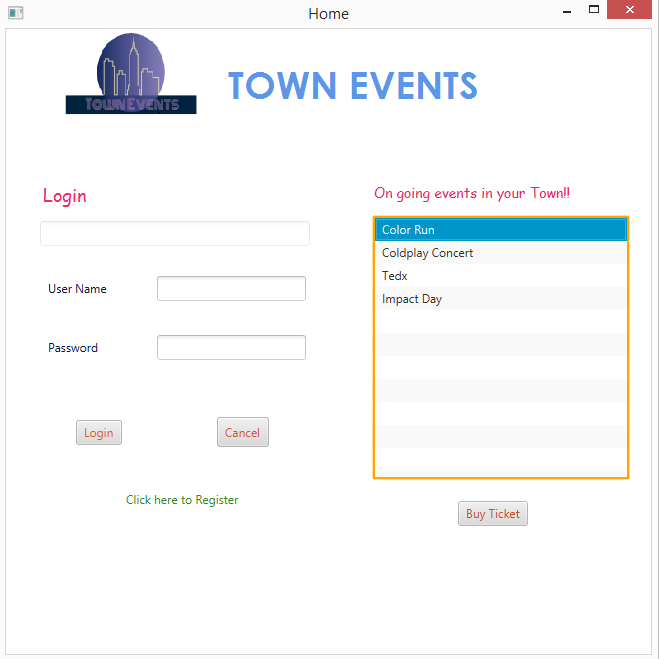


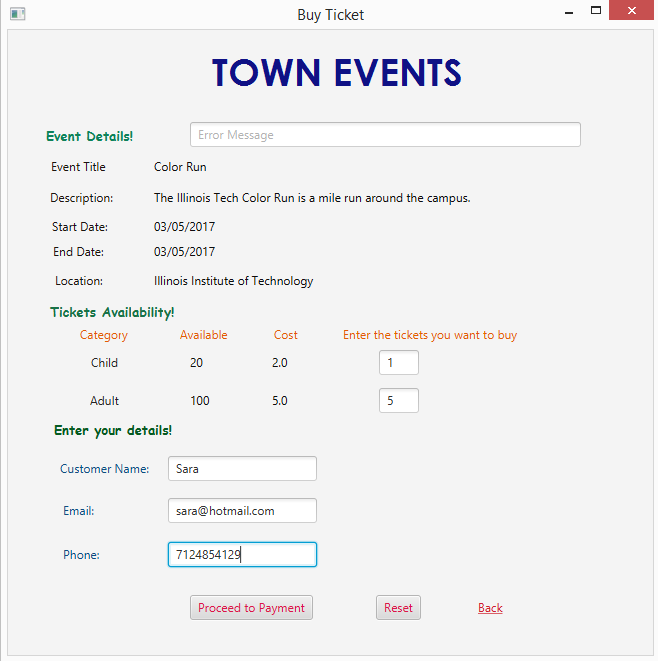
After deleting the event, event is deleted from list too.



**6. Customer:**

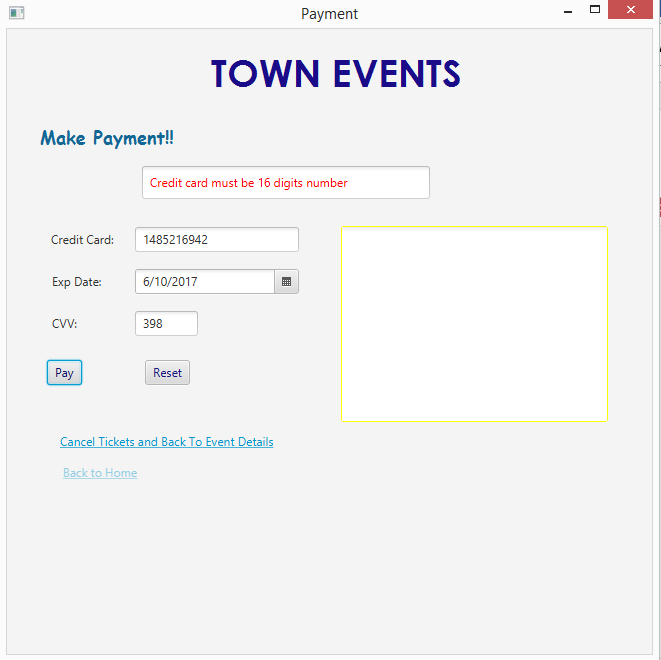
Customer will buy tickets from the Town Event Portals main page from the list of ongoing events.





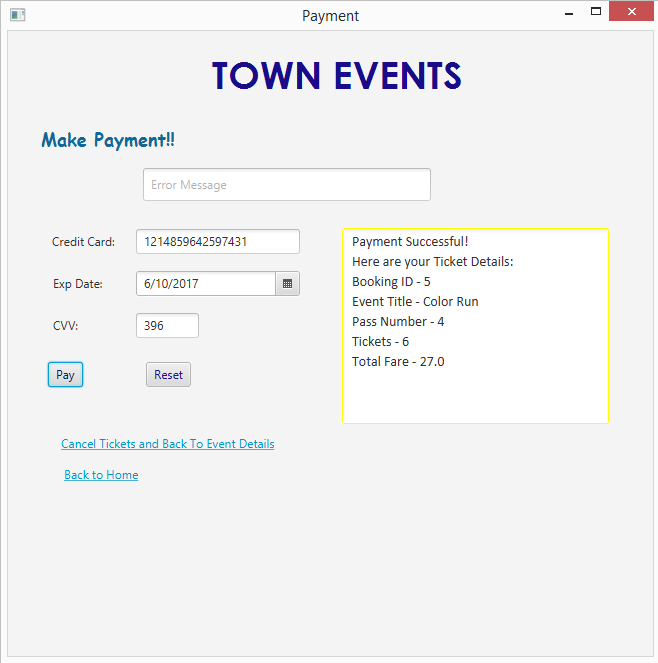
After clicking on proceed to payment we are redirected to payment where there are validations like credit card number must be 16 digits, expiry date must be in future, cvv is a 3 digit number.

**7. Payment Screen:**



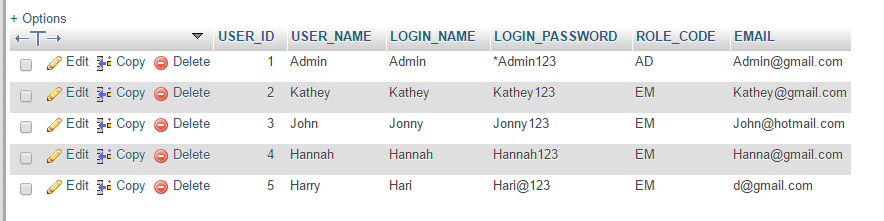
After clicking on Pay button, Payment Module generates receipt on the screen displaying the ticket details.

The hyperlink Cancel Tickets and Back to Event Details deletes the ticket and the payment done.

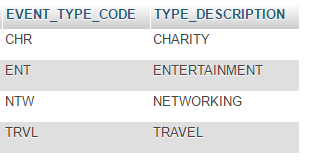


## Screenshots of Database Table Structures:

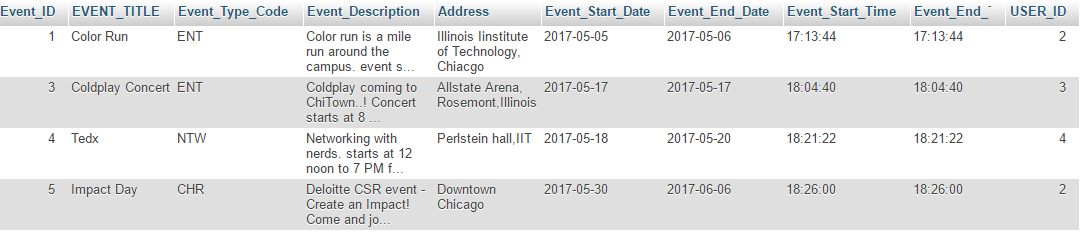
**Townevents\_user\_table:**



**Townevents\_event\_type:**



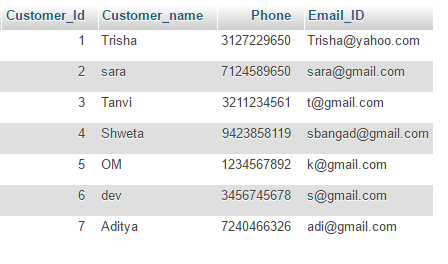
**Townevents\_events\_booking:**



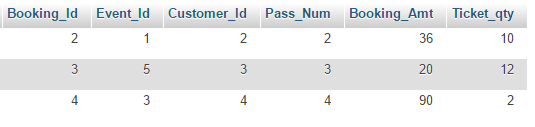
**Townevents\_event\_tickets:**



**Townevents\_customer:**



**Townevents\_ticket\_bookings:**



## Executable Files:

**The executable Jar of the application is attached along with this document. Also, a video is made for the application capturing the flow and working of the system.**

Please find below, the Video Links of the modules

**Admin and event manager module:** <https://www.screencast.com/t/PzZtFhlE48ie>

**Customers Module:** <https://www.screencast.com/t/M1FBgtp5bT>

**Executable JAR file**

