*Event Horizon*

Class Description Document

*Team 7*

*American University of Sharjah*

College of Engineering

Computer Science and Engineering

Date: 15-03-2018

Jayroop Ramesh 57412

Saad Bashar 61382

Zahra Sadat 59068

Version 1.0: EH-CD-15032018

**Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Author(s)** | **Change Description** | **Date** | **Notes** |
| 1.0 | Jayroop, Zahra, Saad | Initial Class Diagrams and Descriptions | 15/03/2018 | 16 Classes |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Content**

[1 Overview 4](#_Toc508889373)

[2 Classes 5](#_Toc508889374)

[2.1 Class Diagram 5](#_Toc508889375)

[2.2 Class Descriptions 10](#_Toc508889376)

# Overview

This document outlines the initial developed classes responsible for the core logic of the Event Horizon event management web application. The application allows organizers, suppliers and attenders to set up, provide services and go to events respectively. Firstly, for organizers, the application will have a listing of all the locations, vendors, and other necessary factors to host an event, along with each amenities’ pricing. Secondly, suppliers can post their items, human resources and venues which is indexed by the search engines available to the organizers filtered by location and type of service. Lastly, attendees simply open the platform and browse available events, filtered by their interest. The functionalities focused on by these classes are based on the previously formulated use-cases, and functional and non-functional requirements. The user interface, database and external services are not included in this version as for architectural integrity and modularity they are isolated into various layers to be covered in the subsequent documentation.

# Classes

## Class Diagrams

Figure 1: Class Diagram View

A close up of a map

Description generated with very high confidence

Figure 2A close up of a map

Description generated with very high confidence: User Associations

![A close up of a map

Description generated with high confidence]()

Figure 3 User Subclasses

![A close up of text on a white background

Description generated with very high confidence]()Figure 4 Interrelationships between User subclasses and their respective associations

Figure 5![A close up of a map

Description generated with very high confidence](): Supplier associations and Service subclasses

## Class Descriptions

|  |  |
| --- | --- |
| **Class Name** | 1. User |
| **Class Description** | The user class encapsulates all data related to an account on the system. |
| **Use Case Affiliation** | Buy Tickets, Create Event, Registration, Login, Customize Profile, Logout |
| **Attributes** | username, email, number, password, userID, user\_type |
| **Operations** | *Constructor: User(username, email, password, user\_type)*  This operation is called during “sign up” for account. Returns Boolean to indicate the success of an account creation. |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 6 User Class

|  |  |
| --- | --- |
| **Class Name** | 1. Supplier |
| **Class Description** | The Supplier class is a subclass of User. It encapsulates additional information related to the suppliers of services and equipment. Every supplier has a unique supplier ID that is used to identify items belonging to a specific supplier. |
| **Use Case Affiliation** | Registration, Login, Customize Profile, Logout, Buy Items, Create Item Inventory |
| **Attributes** | service\_type, company\_name, supplier\_ID, bank\_account\_no |
| **Operations** | *Boolean addItem(Item item)*  This function adds an item to items provided by the supplier and returns true if the operation has succeeded.  *Boolean removeItem(int item\_ID)*  This function removes the item with the specified ID from the supplier’s itemlist.  *Resultset show()*  This function returns all items and services provided by the supplier.  *Boolean addService(Service)*  This function adds a service to the list of services provided by the supplier and returns true if the operation has succeeded.  Mutators and Accessors |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 7: Supplier Class

|  |  |
| --- | --- |
| **Class Name** | 1. Organizer |
| **Class Description** | The Organizer class is a subclass of User. It encapsulates additional information related to the organizers of events. Every organizer has a unique org\_ID which is used to identify events organized by a specific organizer. |
| **Use Case Affiliation** | Buy Tickets, Create Event, Registration, Login, Customize Profile, Logout |
| **Attributes** | org\_ID |
| **Operations** | *Resultset* showEvents()  This function returns a list of all events organized by this organizer.  Resultset showActiveEvents()  This function returns a list of all upcoming events organized by this organizer.  *Event createEvent()*  The organizer to create a new event uses this function; it creates a new Event object with the organizer’s org\_ID. It returns a Boolean to indicate the success of the operation.  *Boolean cancelEvent(int event\_ID)*  The organizer to cancel an event uses this function. It changes the status of the event to “cancelled”. It returns a Boolean to indicate the success of the operation.  *Bool modifyEvent(int event\_ID, String detail, String newdet)*  This function is used to modify a certain detail (such as title) of an event by inputting the event ID and the new detail. |

![A screenshot of a cell phone

Description generated with high confidence]()Figure 8: Organizer Class

|  |  |
| --- | --- |
| **Class Name** | 1. Customer |
| **Class Description** | The Customer class is a subclass of User. It encapsulates additional information related to the customer. |
| **Use Case Affiliation** | Registration, Login, Customize Profile, Logout, Search for Events, Buy Tickets |
| **Attributes** | ticketlist: an arrayList<Ticket> that contains all tickets purchased by the customer.  c\_card: an object of type CreditCard which the customer uses to buy tickets. The customer can choose to save their credit card info or input it at every purchase. |
| **Operations** | *Resultset showTickets()*  This function returns a resultset of all tickets purchased by the customer.  *Ticket buyTickets(Event)*  This function is used by the customer to buy tickets for an event. |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 9: Customer Class

|  |  |
| --- | --- |
| **Class Name** | 1. Login |
| **Class Description** | This class is used by a user to log into their account by establishing connection with the database and validating their credentials. |
| **Use Case Affiliation** | Login |
| **Attributes** | inputEmail, inputPass |
| **Operations** | *Boolean loginAttempt(inputEmail, inputPass)*  This function is used to login by inputting the email and password. It returns a Boolean to indicate the success of the login attempt. |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 10: Login

|  |  |
| --- | --- |
| **Class Name** | 1. Event |
| **Class Description** | This class encapsulates all details related to an event. |
| **Use Case Affiliation** | Search for Events, Buy Tickets, Create Event |
| **Attributes** | itemlist: an arraylist of type Item which contains all items used for this event.  servicelist: an arraylist of type Service which contains all services used for this event.  Venue, org\_ID, ticket\_price, date, type, title, event\_ID, status |
| **Operations** | *Boolean buyItem(Item)*  This function is used to buy an Item for the event. It returns a Boolean to indicate the success of the operation.  *Boolean buyService(Service)*  This function is used to buy a Service for the event. It returns a Boolean to indicate the success of the operation.  *Boolean returnItem(Item)*  This function is used to return an Item to the supplier. It returns a Boolean to indicate the success of the operation.  *Boolean returnService(Service)*  This function is used to return a Service to the supplier. It returns a Boolean to indicate the success of the operation.  Mutators and Accessors for attributes. |

**![A screenshot of a cell phone

Description generated with very high confidence]()**

Figure 11: Event Class

|  |  |
| --- | --- |
| **Class Name** | 1. Ticket |
| **Class Description** | This class encapsulates information regarding the tickets of an event. |
| **Use Case Affiliation** | Buy Tickets |
| **Attributes** | Ticket\_id, Holder\_Name, Price, Seat\_No, Date, Event\_Title, event\_ID |
| **Operations** | N/A |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 12: Ticket

|  |  |
| --- | --- |
| **Class Name** | 1. Services(Human Resources) |
| **Class Description** | This class encapsulates information about the different services that a supplier can provide. The supplier of the service is identified by the supplier\_ID. |
| **Use Case Affiliation** | Buy Items, Search for Supplies, Create Item Inventory. |
| **Attributes** | Supplier\_ID, Service\_ID, Price |
| **Operations** | *Void Specify(int num, String type)*  This function permits the selection of a specific type of services, based on its subclasses and the number of them needed. |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 13: Service/HR Class

|  |  |
| --- | --- |
| **Class Name** | 1. Transaction |
| **Class Description** | This class performs the buying and selling processes between users by establishing connection with the bank. |
| **Use Case Affiliation** | Buy Items, Buy Tickets |
| **Attributes** | Trans\_no, amount, buyer\_id, seller\_id, date, |
| **Operations** | *Boolean execute(buyer\_id, seller\_id, amount)*  This function performs the transaction between a buyer and a seller. Returns a Boolean to indicate success of transaction. |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 14: Transaction

|  |  |
| --- | --- |
| **Class Name** | 1. Venue |
| **Class Description** | This class encapsulates information about venues where events take place. |
| **Use Case Affiliation** | Search for Supplies, Buy Items |
| **Attributes** | Address, no\_seats, area, type, image |
| **Operations** | N/A |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 15: Venue Class

|  |  |
| --- | --- |
| **Class Name** | 1. Item |
| **Class Description** | This class encapsulates information about items and equipment. |
| **Use Case Affiliation** | Create Item Inventory, Search for Supplies, Buy Items |
| **Attributes** | Item\_ID, Service\_ID, Price, Type, Description, No\_Stock, Company\_make |
| **Operations** | N/A |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 16: Item Class

|  |  |
| --- | --- |
| **Class Name** | 1. Search |
| **Class Description** | This class encapsulates the searches performed by user and associated results. |
| **Use Case Affiliation** | Search for Events, Search for supplies |
| **Attributes** | N/A |
| **Operations** | *Resultset searchEvents(String keyword)* This function returns a resultset of events matched by the keyword parameter.  *Resultset searchEventsbydate(Date startdate, Date enddate)*  This function returns a resultset of events between the start and end dates.  *Resultset searchServices(String keyword)*  This function returns a resultset of services matched by the keyword.  *Resultset searchItems(String keyword)*  This function returns a resultset of items matched by the keyword.  *Resultset allEvents()*  This function returns all events stored on the database.  *Resultset allItems()*  This function returns all items stored on the database.  *Resultset allServices()*  This function returns all services stored on the database. |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 17: Search Class

|  |  |
| --- | --- |
| **Class Name** | 1. Caterer |
| **Class Description** | This is a subclass of Service. This class encapsulates information about the catering services relating to food for the event. |
| **Use Case Affiliation** | Create Item Inventory, Search for Supplies, Buy Items |
| **Attributes** | Cusine, Menu, Head\_count |
| **Operations** | N/A |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 18: Caterer Class

|  |  |
| --- | --- |
| **Class Name** | 1. Entertainers |
| **Class Description** | This is a subclass of Service. This class encapsulates information about the entertainment providing artists for the event. |
| **Use Case Affiliation** | Create Item Inventory, Search for Supplies, Buy Items |
| **Attributes** | Artist\_name, genre\_type |
| **Operations** | N/A |

![A screenshot of a cell phone

Description generated with very high confidence]() Figure 19: Entertainer Class

|  |  |
| --- | --- |
| **Class Name** | 1. Staff |
| **Class Description** | This is a subclass of Service. This class encapsulates information about the staff for the event. |
| **Use Case Affiliation** | Create Item Inventory, Search for Supplies, Buy Items |
| **Attributes** | Specialty, work\_hours. |
| **Operations** | N/A |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 19: Staff Class

|  |  |
| --- | --- |
| **Class Name** | 1. CreditCard |
| **Class Description** | This class encapsulates information about the credit card used by the user to make purchases. |
| **Use Case Affiliation** | Buy Items, Buy Tickets |
| **Attributes** | holder\_name, expiry, cvv, card\_number, type |
| **Operations** | *Boolean Validate()*  This function validates the card with the bank. |

![A screenshot of a cell phone

Description generated with very high confidence]()Figure 20: CreditCard Class