|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Logo_FPT_University_doc** |  | | | **FPT UNIVERSITY** | | |
| Capstone Project |
| Software Requirements Specification |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | **Group 8** | | | | | **Group Members** | Lê Khôi Phong |  | 60473 | | Đào Như Tùng |  | 60408 | | Công Minh Hiếu |  | 60535 | | Nguyễn Thanh Tùng  Đặng Quốc Duy |  | 60513  00276 | | **Supervisor** | Lâm Hữu Khánh Phương | | | | **Capstone Project code** | DropIT | | | | |
| * Ho Chi Minh City, june/ 2013 - |

# Table of Contents

[Table of Contents 2](#_Toc353754554)

[3.1 User Requirement Specification 3](#_Toc353754555)

[3.1.1 Common Features 3](#_Toc353754556)

[3.1.2 Tiktak staff Management 3](#_Toc353754557)

[3.1.3 Item Management 3](#_Toc353754558)

[3.1.4 Hub’s Order Tracking 3](#_Toc353754559)

[3.1.5 Order info checking 4](#_Toc353754560)

[3.1.6 Order Management 4](#_Toc353754561)

[3.2 System Requirement Specification 4](#_Toc353754562)

[3.2.1 External Interface Requirements 4](#_Toc353754563)

[3.2.1.1 User Interfaces 4](#_Toc353754564)

[3.2.1.2 Hardware Interfaces 4](#_Toc353754565)

[3.2.1.3 Software Interfaces 5](#_Toc353754566)

[3.2.1.4 Communications Protocol 5](#_Toc353754567)

[3.2.2 System Features 6](#_Toc353754568)

[3.2.2.1 Manage Plan 7](#_Toc353754569)

[3.2.2.2 Customer 14](#_Toc353754570)

[3.2.2.3 Manage Item 27](#_Toc353754571)

[3.2.2.4 Manage Request (not implement yet) 31](#_Toc353754572)

[3.2.3 Software System Attributes 33](#_Toc353754573)

[3.2.3.2 Reliability 33](#_Toc353754574)

[3.2.3.3 Availability 33](#_Toc353754575)

[3.2.3.4 Security 34](#_Toc353754576)

[3.2.3.5 Maintainability 34](#_Toc353754577)

# User Requirement Specification

## Common Features

* Only authenticated users can access the system. Users can log in and log out using their own accounts.
* Users can change their password.
* All guest or user can view event, artist. And they can search all event or artist, and view ticket or request ticket.
* Only authorized users can use specific functions of the system.

## Authorized User

* User can follow category(they’re interested in/ of their own preference).
* User can view history their transaction, and claim for transaction (in 5 days from transaction’s day).
* User want to sell a ticket, user have to provide bank account, and was tangible by admin.
* Show the amount of money belongs to the seller after subtracting 7% for website spending and round up/ down amount of money.
* When user creates a request or ticket, they can choose default event of website or they can create new event.
* User is rate prestige.

## Admin

* Admin have to manage category, event, user, transaction, venue, request, tickets.

## Guest

* They only can to view and search but they can’t implement authorized user’ function.

# System Requirement Specification

## External Interface Requirements

### User Interfaces

* The design should be simple and user-friendly. White and dark will be 2 main colors of the website while charts may have more color to visually express data more effectively.
* The menu bar should be on the left and not take too much space of the screen. It consists of the list of main functions of the system that users can access.
* The design should be responsive. It means that the web components should be scaled according to a range of resolutions and devices to provide a consistent experience, no matter what.

### Hardware Interfaces

* To access to the system, users only need any type of computer, tablet, or mobile phone with a fair internet connection.

### Software Interfaces

* At the server side, the system should run on top of Windows 7, Windows Server 2008 or later versions of Windows Server. Besides, Microsoft .NET Framework 4 and MVC 3 should be installed on the server. The database management system use for SMDH is SQL Server 2008 R2.
* At the client side, users can use any modern browser that supports JavaScript and HTML 5 to access to the system.

### Communications Protocol

HTTP is the protocol used for loading the web site in browsers.

## System Features

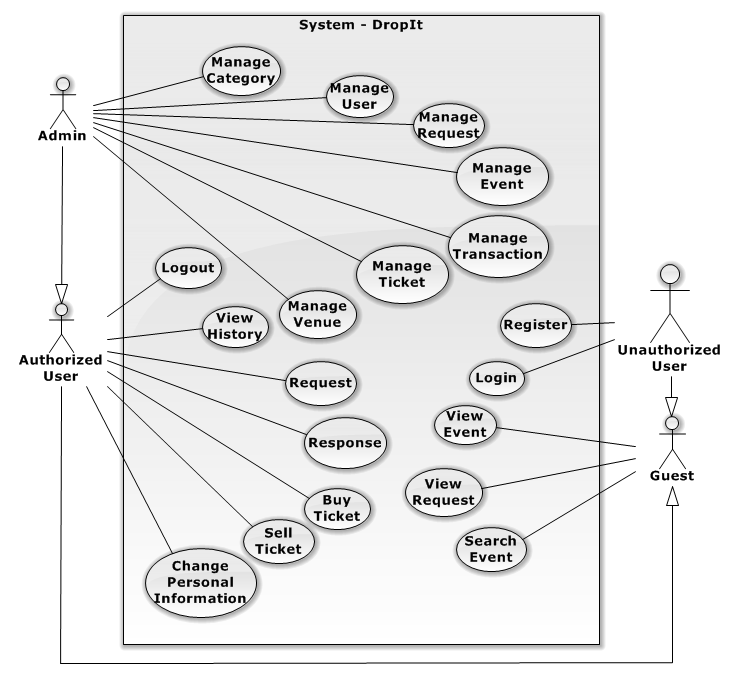


Figure 1 - General Use Case Diagram

### Admin

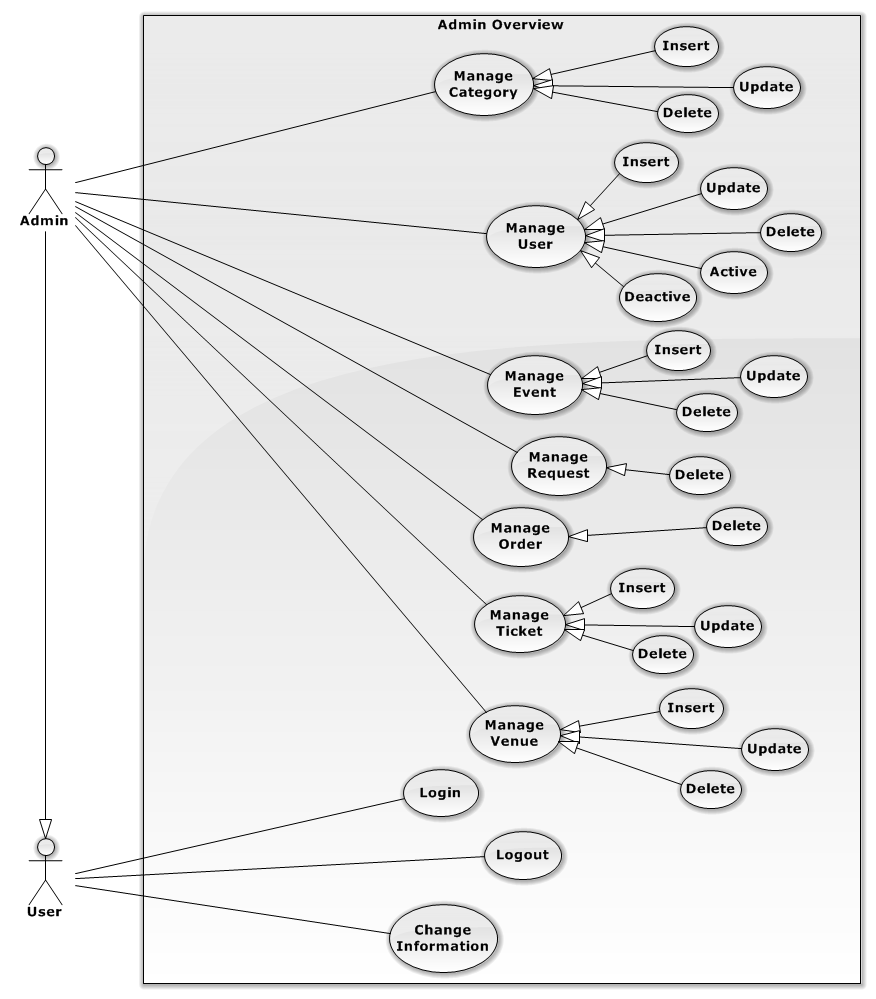


Figure 2 – Admin

#### Insert Category



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – INSERT CATEGORY | | | |
| Use-case No. | UC01.1 | Use-case Version | 1.0 |
| Use-case Name | Insert category | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to insert category  **Goal:** Insert category successful  **Triggers:** N/A  **Pre-conditions:** N/A  **Post-conditions:** Category is created  **Main Success Scenario:**   1. Open Admin page, choose Category and insert category 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Update Category



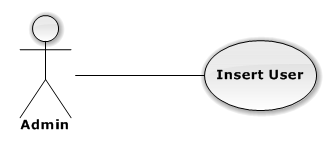
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UPDATE CATEGORY | | | |
| Use-case No. | UC01.2 | Use-case Version | 1.0 |
| Use-case Name | Update category | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to update category after creating a category  **Goal:** Category is updated  **Triggers:** N/A  **Pre-conditions:** Category is already created  **Post-conditions:** Update category successful  **Main Success Scenario:**   1. Open Admin page, choose Category and update category 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Delete Category



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – DELETE CATEGORY | | | |
| Use-case No. | UC01.3 | Use-case Version | 1.0 |
| Use-case Name | Delete category | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to delete category after creating a category  **Goal:** Category is deleted  **Triggers:** N/A  **Pre-conditions:** Category is already created  **Post-conditions:** Delete category successful  **Main Success Scenario:**   1. Open Admin page, choose Category and delete category 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Insert User



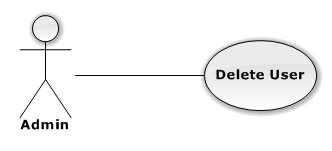
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – INSERT CATEGORY | | | |
| Use-case No. | UC01.4 | Use-case Version | 1.0 |
| Use-case Name | Insert user | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to insert user  **Goal:** Insert user successful  **Triggers:** N/A  **Pre-conditions:** N/A  **Post-conditions:** User is inserted  **Main Success Scenario:**   1. Open Admin page, choose User and insert user 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Update User



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UPDATE CATEGORY | | | |
| Use-case No. | UC01.5 | Use-case Version | 1.0 |
| Use-case Name | Update user | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to update user after creating a user  **Goal:** User is updated  **Triggers:** N/A  **Pre-conditions:** User is already created  **Post-conditions:** Update user successful  **Main Success Scenario:**   1. Open Admin page, choose User and update user 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Delete User



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – DELETE USER | | | |
| Use-case No. | UC01.6 | Use-case Version | 1.0 |
| Use-case Name | Delete user | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to delete user after creating a user  **Goal:** User is deleted  **Triggers:** N/A  **Pre-conditions:** User is already created  **Post-conditions:** Delete user successful  **Main Success Scenario:**   1. Open Admin page, choose User and delete user 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Active User



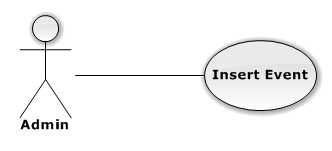
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – INSERT CATEGORY | | | |
| Use-case No. | UC01.7 | Use-case Version | 1.0 |
| Use-case Name | Active user | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to active a user  **Goal:** Active user successful  **Triggers:** N/A  **Pre-conditions:** User is deactivate  **Post-conditions:** User is activate  **Main Success Scenario:**   1. Open Admin page, choose User and active user 2. Click Next button 3. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Deactive User



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – INSERT CATEGORY | | | |
| Use-case No. | UC01.8 | Use-case Version | 1.0 |
| Use-case Name | Deactive user | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to deactive a user  **Goal:** Deactive user successful  **Triggers:** N/A  **Pre-conditions:** User is activate  **Post-conditions:** User is deactivate  **Main Success Scenario:**   1. Open Admin page, choose User and deactive user 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Insert Event



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – INSERT EVENT | | | |
| Use-case No. | UC01.9 | Use-case Version | 1.0 |
| Use-case Name | Insert event | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to insert event  **Goal:** Insert event successful  **Triggers:** N/A  **Pre-conditions:** N/A  **Post-conditions:** Event is created  **Main Success Scenario:**   1. Open Admin page, choose Event and insert event 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Update Event



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UPDATE EVENT | | | |
| Use-case No. | UC01.10 | Use-case Version | 1.0 |
| Use-case Name | Update event | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to update event after creating an event  **Goal:** Event is updated  **Triggers:** N/A  **Pre-conditions:** Event is already created  **Post-conditions:** Update event successful  **Main Success Scenario:**   1. Open Admin page, choose Event and update event 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Delete Event



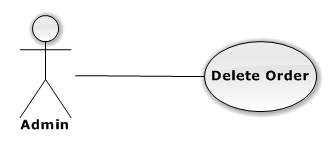
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – DELETE EVENT | | | |
| Use-case No. | UC01.11 | Use-case Version | 1.0 |
| Use-case Name | Delete event | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to delete an event after creating event  **Goal:** Event is deleted  **Triggers:** N/A  **Pre-conditions:** Event is already created  **Post-conditions:** Delete event successful  **Main Success Scenario:**   1. Open Admin page, choose Event and delete event 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Delete Request



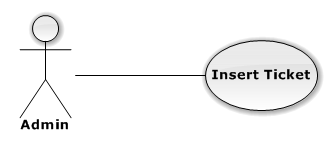
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – DELETE REQUEST | | | |
| Use-case No. | UC01.12 | Use-case Version | 1.0 |
| Use-case Name | Delete request | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to delete request after creating request  **Goal:** Request is deleted  **Triggers:** N/A  **Pre-conditions:** Request is already created  **Post-conditions:** Delete request successful  **Main Success Scenario:**   1. Open Admin page, choose Request and delete request 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Delete Order



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – DELETE ORDER | | | |
| Use-case No. | UC01.13 | Use-case Version | 1.0 |
| Use-case Name | Delete order | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to delete order after creating order  **Goal:** Order is deleted  **Triggers:** N/A  **Pre-conditions:** Order is already created  **Post-conditions:** Delete order successful  **Main Success Scenario:**   1. Open Admin page, choose Order and delete order 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Insert Ticker



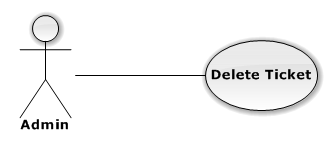
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – INSERT TICKET | | | |
| Use-case No. | UC01.14 | Use-case Version | 1.0 |
| Use-case Name | Insert ticket | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to insert ticket  **Goal:** Insert ticket successful  **Triggers:** N/A  **Pre-conditions:** N/A  **Post-conditions:** Ticket is inserted  **Main Success Scenario:**   1. Open Admin page, choose Ticket and insert ticket 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Update Ticket



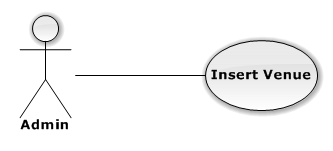
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UPDATE EVENT | | | |
| Use-case No. | UC01.15 | Use-case Version | 1.0 |
| Use-case Name | Update ticket | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Date | 07/31/2013 |
| **Actor:** Admin  **Summary:** This use case is about how to update ticket  **Goal:** Update ticket successful  **Triggers:** N/A  **Pre-conditions:** Ticket is already created  **Post-conditions:** Ticket is updated  **Main Success Scenario:**   1. Open Admin page, choose Ticket and update ticket 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Delete Ticket



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – DELETE TICKET | | | |
| Use-case No. | UC01.16 | Use-case Version | 1.0 |
| Use-case Name | Delete ticket | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to delete ticket after creating ticket  **Goal:** Ticket is deleted  **Triggers:** N/A  **Pre-conditions:** Ticket is already created  **Post-conditions:** Delete ticket successful  **Main Success Scenario:**   1. Open Admin page, choose Ticket and delete ticket 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Insert Venue



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – INSERT VENUE | | | |
| Use-case No. | UC01.17 | Use-case Version | 1.0 |
| Use-case Name | Insert venue | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to insert venue  **Goal:** Insert venue successful  **Triggers:** N/A  **Pre-conditions:** N/A  **Post-conditions:** Venue is inserted  **Main Success Scenario:**   1. Open Admin page, choose Ticket and insert ticket 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Update Venue



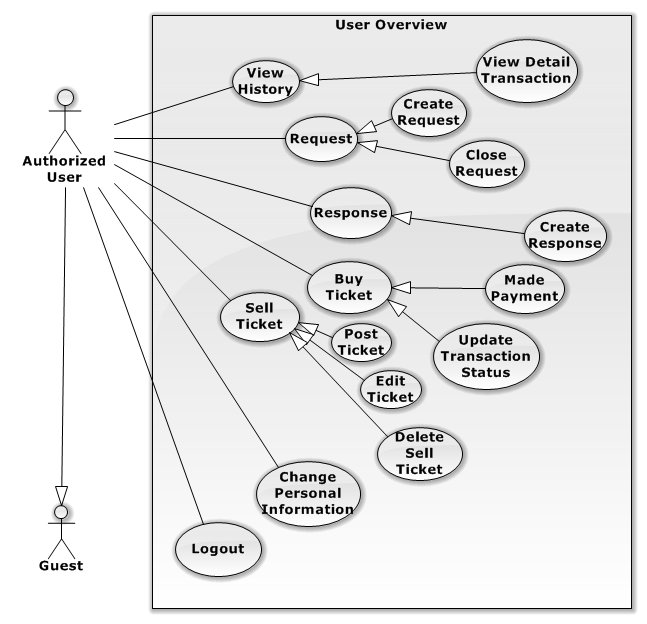
|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – UPDATE VENUE | | | |
| Use-case No. | UC01.18 | Use-case Version | 1.0 |
| Use-case Name | Update venue | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Date | 07/31/2013 |
| **Actor:** Admin  **Summary:** This use case is about how to update venue  **Goal:** Update venue successful  **Triggers:** N/A  **Pre-conditions:** Venue is already created  **Post-conditions:** Venue is updated  **Main Success Scenario:**   1. Open Admin page, choose Venue and update venue 2. Click Next button 3. Fill in receiver information 4. Click OK button   **Alternative Scenario:**  N/A  **Exceptions:** 3. In step 3, if some required fields are not filled in, admin click *OK* button, validation message will be shown.  **Relationships:**  N/A  **Business Rules:** N/A | | | |

#### Delete Venue



|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE – DELETE VENUE | | | |
| Use-case No. | UC01.19 | Use-case Version | 1.0 |
| Use-case Name | Delete venue | | |
| Author | TungNT | | |
| Date | 07/31/2013 | Priority | Normal |
| **Actor:** Admin  **Summary:** This use case is about how to delete venue after creating venue  **Goal:** Venue is deleted  **Triggers:** N/A  **Pre-conditions:** Venue is already created  **Post-conditions:** Delete venue successful  **Main Success Scenario:**   1. Open Admin page, choose Venue and delete venue 2. Click Next button 3. Click OK button   **Alternative Scenario:**  2. When Admin click Next button, system shows confirm message  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | |

### User

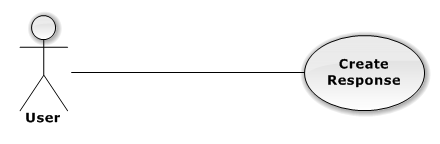


#### View Detail Transaction

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – View Detail Transaction | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | View Detail Transaction | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user view history detail transaction.  **Goal:** User can view detail transaction successfully.  **Triggers:** At the left side, click on “lịch sử giao dịch”.  **Pre-conditions:**  User login into system already.  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “lịch sử giao dịch” at the left side | System will display list of transaction of user in database. | | 2 | Click on “chi tiết” button of a transaction | System will show detail transaction. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Create Request



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Create Request | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Create Request | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user make a new request  **Goal:** User can make a new request successfully  **Triggers:** At the left side, click on “rao mua vé”.  **Pre-conditions:**  User login into system already.  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “rao mua vé” at the left side | System will go to detail request page. | | 2 | Input information your request. | System will validate input. | | 3 | Click on “đăng lời rao” button. | System will validate, and change notification for admin. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Close Request



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Close Request | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Close Request | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how close a request  **Goal:** User can close request successfully  **Triggers:** At the left side, click on “quản lí rao mua”.  **Pre-conditions:**  User login into system already.  **Post-conditions:** N/A.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “quản lí rao mua” at the left side | System will display list of request in database. | | 2 | Click on “hủy bỏ” button | System will validate, and change status to “đã hủy”. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Create Response



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Create Response | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Create Response | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user make a new response.  **Goal:** User can make a new response successfully.  **Triggers:** N/A  **Pre-conditions:**  User login into system already.  **Post-conditions:** N/A.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “xem yêu cầu cần vé” tab at the left side | System will display list of request in database. | | 2 | Choose a request, and click on “rao vé” button | System will go to this Request page. | | 3 | Input information of ticket or choose a post ticket | System will validate input. | | 4 | Click “trả lời” button | System will validate and post a direct link your post ticket in “chi tiết lời rao” page of request. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Made Payment



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Made Payment | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Made Payment | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user made a payment  **Goal:** User can make a new payment successfully  **Triggers:** click “sự kiện” tab on the top header  **Pre-conditions:**  User login into system already.  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “sự kiện” tab on header, and choose the event in category | System will go to direct manage event page. | | 2 | Click “đặt vé” button | System will show detail sell ticket page. | | 3 | Input information for buy | System will validate input. | | 4 | Click “xác nhận” button | System will validate and save to database. And status transaction will change ‘đang giao dịch’. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.2.7 Update Transaction Status



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Update Transaction Status | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Update Transaction Status | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user update status of transaction.  **Goal:** User can change status transaction successfully.  **Triggers:** click “lịch sử giao dịch” tab at the left side  **Pre-conditions:**  User login into system already.  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “lịch sử giao dịch” tab at the left side | System will display list of transaction of user in database. | | 2 | Click “chi tiết” button of a transaction | System will show detail transaction | | 3 | Choose status in text box | System will validate input. | | 4 | Click “Lưu” | System will validate, save to database, and change status of transaction. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.2.8 Post Ticket



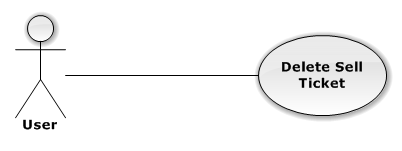
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Post Ticket | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Post Ticket | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user post a ticket.  **Goal:** User can make a new post ticket successfully.  **Triggers:** Click “bán” tab on the top header  **Pre-conditions:**  User login into system already.  **Post-conditions:** The request is created successfully.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “bán” tab on header | System will go to direct home seller page. | | 2 | Click “tạo mới” button at the left side | System will show “đăng vé” page. | | 3 | Input information for sell | System will validate input. | | 4 | Click “đăng vé” button | System will validate and save to database. |   **Alternative Scenario:**  **Exceptions:**  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.2.9 Edit Sell Ticket



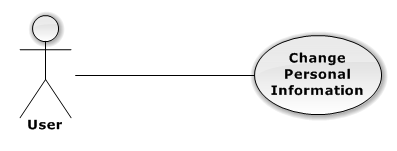
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Edit Sell Ticket | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Edit Sell Ticket | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user edit information for sell ticket  **Goal:** User can edit information to successfully  **Triggers:** click on “vé đang bán” tab at the left side  **Pre-conditions:**  User login into system already, and don’t person to order this ticket  **Post-conditions:** N/A.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “vé đang bán” tab at the left side | System will display list of sell ticket. | | 2 | Click on”chi tiết” button of a transaction | System will show detail transaction. | | 3 | Input your change | System will validate input. | | 4 | Click “thay đổi” | System will validate and save to database. And show messege “đã lưu thay đổi”. |   **Alternative Scenario:** N/A  **Exceptions: N/A**  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.2.10 Delete Sell Ticket



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Delete Sell Ticket | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Delete Sell Ticket | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User  **Summary:** This use case is about how user delete a sell ticket  **Goal:** User can delete a sell ticket successfully  **Triggers:** click on “vé đang bán” tab at the left side  **Pre-conditions:**  User login into system already, and don’t have person to buy this ticket.  **Post-conditions:** N/A.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “vé đang bán” tab at the left side | System will display list of sell ticket. | | 2 | Click on “xóa” button of this sell ticket | System will validate, and save to database. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Change Personal Information



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Change Personal Information | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Change Personal Information | | | |
| Author | DuyDQ | | | |
| Date | 06/01/2013 | Priority | Normal | |
| **Actor:** User, Admin  **Summary:** When actor wants to change detail, user can use change personal information function to change user account’s information  **Goal:** User can change detail information successfully  **Triggers:** On Main page, click on ”thay đổi thông tin”, then the change detail page will be showed to use can change information  **Pre-conditions:**  User login into system already.  **Post-conditions:** User’s information is changed.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on “Thay đổi thông tin” on User Main Page. | System will show the change detail form. | | 2 | Useredit information | System validate new data | | 3 | Click “Hoàn tất” button. | System will show the success page and transfer to User Profile main page. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Logout



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Logout | | | | |
| Use-case No. | UC03.1 | Use-case Version | | 1.0 |
| Use-case Name | logout | | | |
| Author | DuyDQ | | | |
| Date | 01/06/2013 | Priority | Normal | |
| **Actor:** Admin, User, Authorized User  **Summary:** When user want to login with other account, or user want to get out from system, user can use “Logout” function.  **Goal:** Get out from system successfully.  **Triggers:** At the top of right, user click on “đăng xuất”  **Pre-conditions:** User login into system already.  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on the “Đăng Xuất” button at the top of right | System will remove user’s session and transfer user to main page that guest can see. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

### Guest

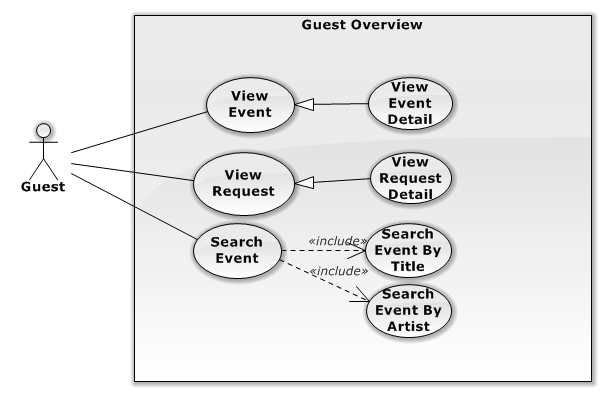
****

Figure 5 – Tiktak staff

#### 3.2.2.4.1 View Event Detail



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – View Event Detail | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | View Event Detail | | | |
| Author | DuyDQ | | | |
| Date | 01/06/2013 | Priority | Normal | |
| **Actor:** Admin, User, Authorized User, Guest, Unauthorized User.  **Summary:** This use case is about how actor view event detail.  **Goal:** User can view event detail successfully.  **Triggers:** N/A  **Pre-conditions:** N/A  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Respons | | 1 | Select category at the top header | System will display list of event in database. | | 2 | Click on a event | System will show event detail. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.4.2 View Request Detail



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – View Request Detail | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | View Request Detail | | | |
| Author | DuyDQ | | | |
| Date | 01/06/2013 | Priority | Normal | |
| **Actor:** Admin, User, Authorized User, Guest, Unauthorized User.  **Summary:** This use case is about how actor view request detail**.**  **Goal:** Actor can view request detail.  **Triggers:** At the left side, click on “quản lí rao mua”  **Pre-conditions:** N/A.  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “quản lí rao mua” tab at the left side | System will display list of request. | | 2 | Click on”chi tiết” button of a request | System will show detail request. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### 3.2.2.4.3 Seach Event by Title



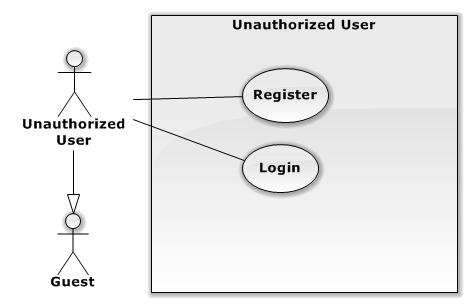
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Seach Event by Title | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Seach Event by Title | | | |
| Author | DuyDQ | | | |
| Date | 01/06/2013 | Priority | Normal | |
| **Actor:** Admin, User, Authorized User, Guest, Unauthorized User.  **Summary:** actor can seach a event that they want to find.  **Goal:** Actor can seach to successfully.  **Triggers:** At the top input information.  **Pre-conditions:** N/A  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Input your title that you want to find in text box | System will validate input. | | 2 | Click on”tìm kiếm” button | System will implement, and will show result. |   **Alternative Scenario:**  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Search Event By Artist

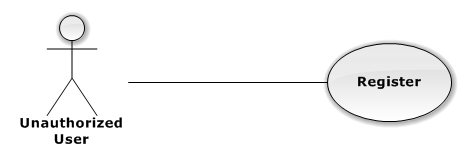


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Seach Event By Artist | | | | |
| Use-case No. |  | Use-case Version | | 1.0 |
| Use-case Name | Seach Event by Artist | | | |
| Author | DuyDQ | | | |
| Date | 01/06/2013 | Priority | Normal | |
| **Actor:** Admin, User, Authorized User, Guest, Unauthorized User.  **Summary:** actor can seach a event that they want to find by artist.  **Goal:** Actor can seach to successfully.  **Triggers:** At the top input information.  **Pre-conditions:** N/A  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Input your artist that you want to find in text box | System will validate input. | | 2 | Click on”tìm kiếm” button | System will implement, and will show result. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

### Unauthorized User



#### Register



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Register | | | | |
| Use-case No. | UC03.1 | Use-case Version | | 1.0 |
| Use-case Name | Register | | | |
| Author | DuyDQ | | | |
| Date | 01/06/2013 | Priority | Normal | |
| **Actor:** Unauthorized User  **Summary:** this usecase is about how unauthorized user can register in system  **Goal:** Unauthorized User register to successfully.  **Triggers:** At the top of right, user click on “đăng ký”  **Pre-conditions:** N/A  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “đăng ký” button at right on the top header | System go to ‘đăng ký tài khoản mới’ page. | | 2 | Input information in text box | System will validate input. | | 3 | Click “đăng ký” button | System will send a mail for active account. |   **Alternative Scenario:**  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

#### Login



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE – Logout | | | | |
| Use-case No. | UC03.1 | Use-case Version | | 1.0 |
| Use-case Name | logout | | | |
| Author | DuyDQ | | | |
| Date | 01/06/2013 | Priority | Normal | |
| **Actor:** Admin, User, Guest  **Summary:** Actor can get in system.  **Goal:** Get in to system successfully.  **Triggers:** On main Page, user can enter username and password into the text field at top of right then click “Đăng Nhập”.  **Pre-conditions:** Actor have to account of system.  **Post-conditions:** N/A  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Select “đăng nhập” at right on the top header | System will display screen for login. | | 2 | Input your account and password into the text field at the top of right | System will validate input. | | 3 | Click “đăng nhập” button | If valid user, show homescreen. Else, navigate back to login form. |   **Alternative Scenario:**  **Exceptions:** N/A  **Relationships:**  N/A  **Business Rules:** N/A | | | | |

## Software System Attributes

### Reliability

* The database can be backed up easily and recovered if necessary.
* The system should never crash or hang, except for the cause from an operating system or network error.
* Mean Time Between Failures (MTBF): The acceptable failure is once a year. The failure should not be because of the database, or else the data may be lost and cannot be recovered.
* Mean Time To Repair (MTTR): When the failure occurs, it should take as little time as possible to repair. The acceptable mean time for a particular failure must be less than 8 hours.
* Maximum Bugs or Defect Rate: 05 bugs / KLOC

### Availability

* The uptime percentage should be at least 99.95%.
* The acceptable time for maintenance or backup should not be more than 8 hours per month.
* When the system goes in under-maintenance, the website should display a message to inform that.

### Security

* Only users with proper account can access certain information of the system. All the information of users must not be available for anyone or software that is not part of the system. User password is also encrypted and not available to the system administrators.

### Maintainability

* The code must follow C#.NET coding and naming convention.
* There should be comments in code files that explain the functions of each code segment.
* All the errors should be logged, which supports bug fixing and maintenance.

# Entity Relationship Diagram