|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Gunpla World Application |
|  |
| |  |  | | --- | --- | | **Team 15** | | | **Group Members** | Hoàng Ngô Minh Tùng – Role 1 – SE62231  Nguyễn Bảo Khánh – Role 2 – SE61978  Lê Nguyễn Ngọc Sang – Role 3 – SE61647 | | **Supervisor** | Phạm Công Thành | | **Ext Supervisor** |  | | **Capstone Project code** |  | |
|  |
|  |

– TP.HCM, 09/2018 –

**E. System Implementation & Test**

**1. Introduction**

**1.1. Overview**

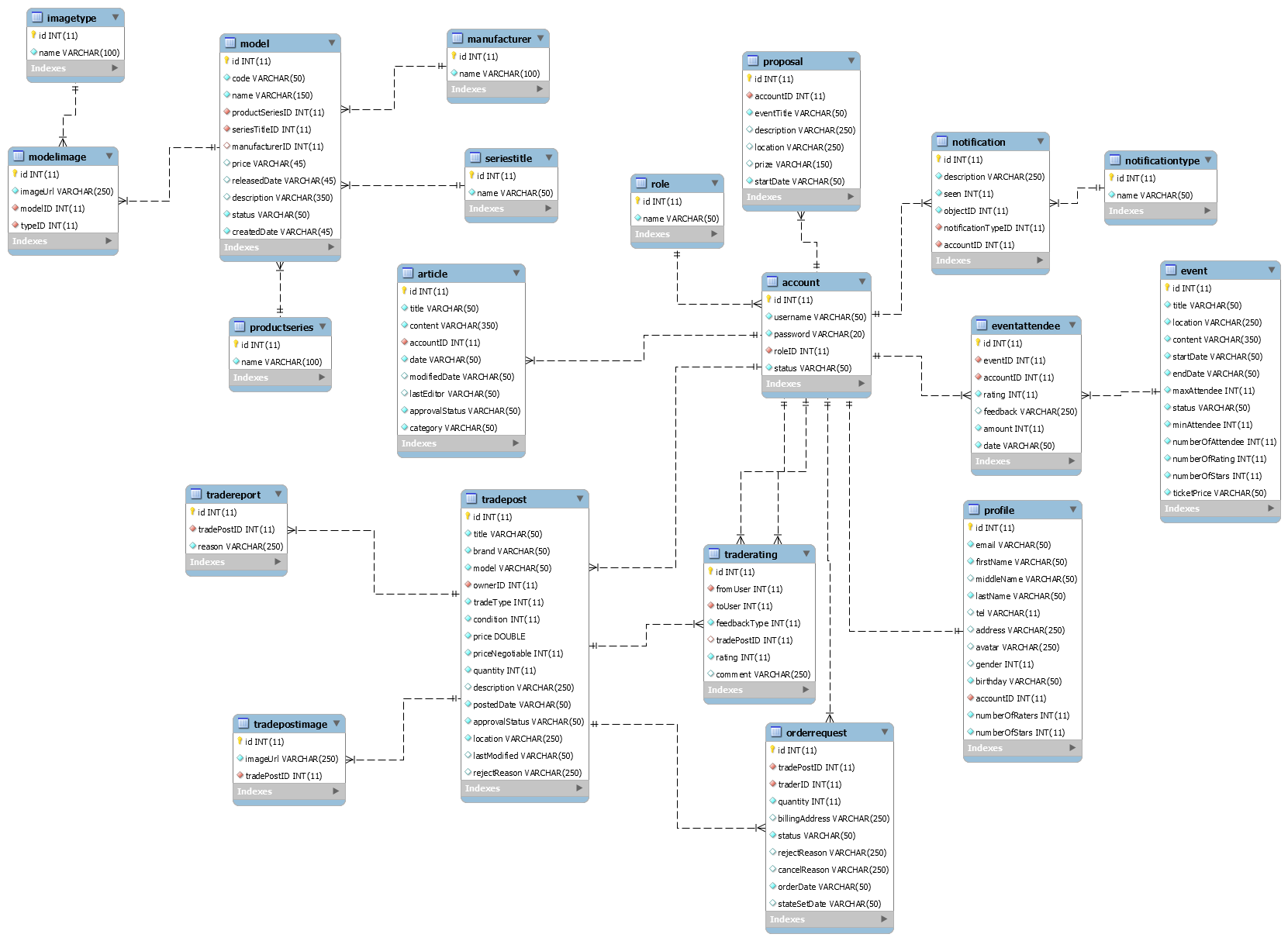
This section provides in detail all necessary information about implementation information of **GWA** which includes physical database design, system state machine diagram and testing procedures which includes test plans, test cases and test result.

**1.2. Test Approach**

We use black box test approach to estimate our system quality. It’s the most appropriate testing method for the system to test all functions from a user’s point of view and help in exposing discrepancies in the specifications. Our testers don’t need to know how the software has been implemented, tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.

**2. Database Relationship Diagram**

**2.1. Physical Diagram**

****

**2.2. Data dictionary**

|  |  |
| --- | --- |
| **Data dictionary: describe content of all tables** | |
| **Table name** | **Description** |
| model | Table which contains model information |
| seriesitle | Table which contains series title information, this is a type of model’s category |
| productseries | Table which contains product series information, this is a type of model’s category |
| manufacturer | Table which contains model’s manufacturer information |
| modelimage | Table which contains model’s images |
| imagetype | Table which contains model’s image type |
| article | Table which contains article information which is posted by authenticated users |
| proposal | Table which contains proposal information which is posted by members |
| account | Table which contains account information |
| profile | Table which contains user’s profile information |
| role | Table which contains role information |
| event | Table which contains event information, which is created by administrator |
| eventattendee | Table which contains information about event’s attendee, we also use this table to contain user’s evaluation about the events |
| notification | Table which contains notification information |
| notificationtype | Table which contains types of notification |
| tradepost | Table which contains trade post information, which is created by buyer/seller |
| traderating | Table which contains user’s evaluation about the exchange |
| tradepostimage | Table which contains trade post’s images |
| tradereport | Table which contains information about user’s report on exchange |
| orderrequest | Table which contains member’s order request to trade posts |

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity name** | **Attributes** | **Description** | **Null** |
| model | id {pk} | Auto increment unique identifier | No |
| code | Model code | No |
| name | Model name | No |
| productSeriesID{fk} | Model’s product series | No |
| seriesTitleID {fk} | Model’s series title | No |
| manufacturerID | Model’s manufacturer | Yes |
| price | Model’s price | Yes |
| releasedDate | Model’s released date | Yes |
| description | Model’s description | Yes |
| status | Model’s status | No |
| createdDate | Model’s created date | No |
| productseries | id {pk} | Auto increment unique identifier | No |
| name | Unique name of product series | No |
| seriestitle | id {pk} | Auto increment unique identifier | No |
| name | Unique name of series title | No |
| manufacturer | id {pk} | Auto increment unique identifier | No |
| name | Unique name of manufacturer | No |
| modelimage | id {pk} | Auto increment unique identifier | No |
| imageUrl | Url of images | No |
| modelID {fk} | Model which images belong to | No |
| typeID {fk} | Type of images | No |
| imagetype | id {pk} | Auto increment unique identifier | No |
| name | Unique name of image type | No |
| account | id {pk} | Auto increment unique identifier | No |
| username | Unique username of user | No |
| password | Account password to login | No |
| roleID {fk} | Account’s role | No |
| status | Account status | No |
| role | id {pk} | Auto increment unique identifier | No |
| name | Unique name of role | No |
| proposal | id {pk} | Auto increment unique identifier | No |
| accountID {fk} | Proposal owner | No |
| eventTitle | Proposal event’s title | No |
| description | Proposal description | Yes |
| location | Proposal event’s location | Yes |
| prize | Proposal event’s prize | Yes |
| startDate | Proposal event start date | No |
| article | id {pk} | Auto increment unique identifier | No |
| title | Article title | No |
| content | Article content | No |
| accountID {fk} | User who posts the articles | No |
| date | Article posted date | No |
| modifiedDate | Article modified date | Yes |
| lastEditor | Article’s last editor | Yes |
| approvalStatus | Article’s approval status | No |
| category | Article’s category | No |
| notification | id {pk} | Auto increment unique identifier | No |
| description | Notification description | No |
| seen | Status seen | No |
| objectID | To save id of content object | No |
| notificationTypeID {fk} | Type of notification | No |
| accountID {fk} | User whom the notification is sent to | No |
| notificationtype | id {pk} | Auto increment unique identifier | No |
| name | Unique name of notification type | No |
| event | id {pk} | Auto increment unique identifier | No |
| title | Event title | No |
| location | Event location | No |
| content | Event’s content | No |
| startDate | Event start date | No |
| endDate | Event end date | No |
| maxAttendee | Event’s max attendee | No |
| status | Event’s status | No |
| minAttendee | Event’s min attendee | No |
| numberOfAttendee | Event’s number of attendee, this attribute is to improve query performance | No |
| numberOfRating | Event’s number of raters, this attribute is to improve query performance | No |
| numberOfStars | Event’s number of star ratings, this attribute is to improve query performance | No |
| ticketPrice | Event’s ticket price | No |
| eventattendee | id {pk} | Auto increment unique identifier | No |
| eventID {fk} | ID of event | No |
| accountID {fk} | Event’s attendee | No |
| rating | Event’s rating which is evaluated by user | No |
| feedback | Feedback of user about event | Yes |
| amount | Number of tickets that a user register | No |
| date | Date when user registers for event | No |
| profile | id {pk} | Auto increment unique identifier | No |
| email | User’s email | No |
| firstName | User’s firstname | No |
| middleName | User’s middleName | Yes |
| lastName | User’s lastname | No |
| tel | User’s mobile number | Yes |
| address | User’s address | Yes |
| avatar | User’s avatar image | Yes |
| gender | User’s gender | Yes |
| birthday | User’s birthday | No |
| accountID {fk} | Profile owner | No |
| numberOfRaters | Number of users who rate this user, this attribute is to improve query performance | No |
| numberOfStars | Number of star ratings from users, this attribute is to improve query performance | No |
| tradepost | id {pk} | Auto increment unique identifier | No |
| title | Trade post title | No |
| brand | Sale product’s brand | No |
| model | Sale product name | No |
| ownerID {fk} | User who posts this trade post | No |
| tradeType | Type of trade post: buyer/seller | No |
| condition | Condition of trade post | No |
| price | Sale product’s price | No |
| priceNegotiable | Price that may be negotiable between traders | No |
| quantity | Sale product quantity | No |
| description | Trade post description | Yes |
| postedDate | Date when trade post is created | No |
| approvalStatus | Trade post approval status by administrator | No |
| location | Trade post location | No |
| lastModified | Trade post’s last modified | Yes |
| rejectReason | Trade post’s reject reason by admin | Yes |
| tradereport | id {pk} | Auto increment unique identifier | No |
| tradePostID {fk} | ID of the trade post which is reported by user | No |
| reason | Report reason | No |
| tradepostimage | id {pk} | Auto increment unique identifier | No |
| imageUrl | Url of image | No |
| tradePostID {fk} | Trade post which images belong to | No |
| traderating | id {pk} | Auto increment unique identifier | No |
| fromUser {fk} | User who evaluates | No |
| toUser {fk} | User who is evaluated | No |
| feedbackType | To check whether the evaluations is from trade owner to user or users to trade owner | No |
| tradePostID {fk} | This attribute is used when user evaluates a trade post | Yes |
| rating | Trade post rating | No |
| comment | User’s feedback about trade post | Yes |
| orderrequest | id {pk} | Auto increment unique identifier | No |
| tradePostID {fk} | The trade post which user request order to | No |
| traderID {fk} | User who requests order to trade post | No |
| quantity | Product order quantity | No |
| billingAddress | Address that is negotiable for exchange | Yes |
| status | Order request status | No |
| rejectReason | This attribute is used when trade post owner reject traders after accepted their order request | Yes |
| cancelReason | This attribute is used when trade post owner cancel user’s order request to trade post | Yes |
| orderDate | Date of order request | No |
| stateSetDate | This attribute is used when trade post owner reject or cancel order request, if reject, this attribute will belong to reject concept. If cancel, this attribute will belong to cancel concept | Yes |

**3. Performance Measures**

Website can locate the user’s location in 2 seconds max

**4. Test plan**

The purpose of this section is to verify and ensure that GWA meets its requirements and design specification. The following part will describe which features to be tested and which will not.

**4.1. Features to be tested**

**Web and mobile application:**

* Search models and view detail
* Crawl models, articles
* Proposal to open event
* Register for event
* Post for trade/exchange
* Request order to a trade post

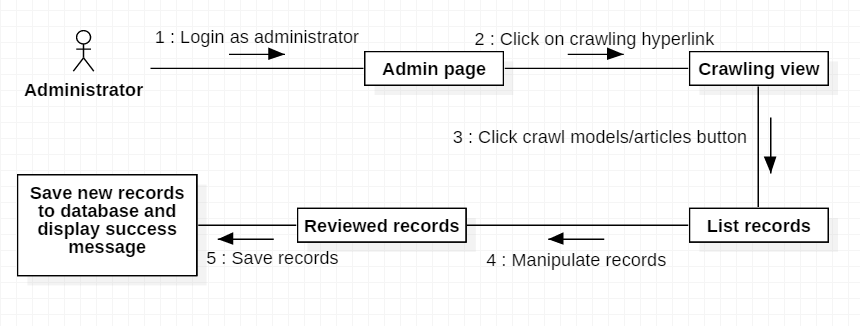
**4.2. Features not to be tested**

Functions are not listed above won’t be tested.

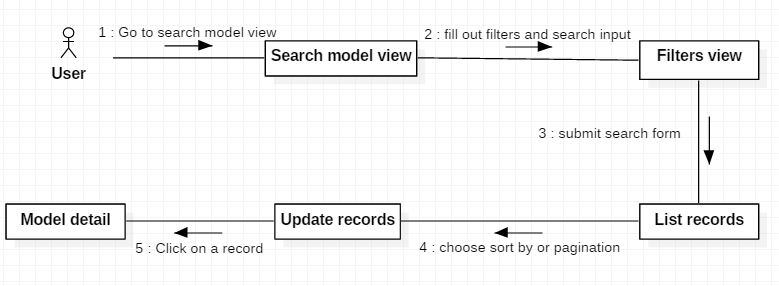
**5. System Testing Test Case**

**5.1. Communication Diagram**

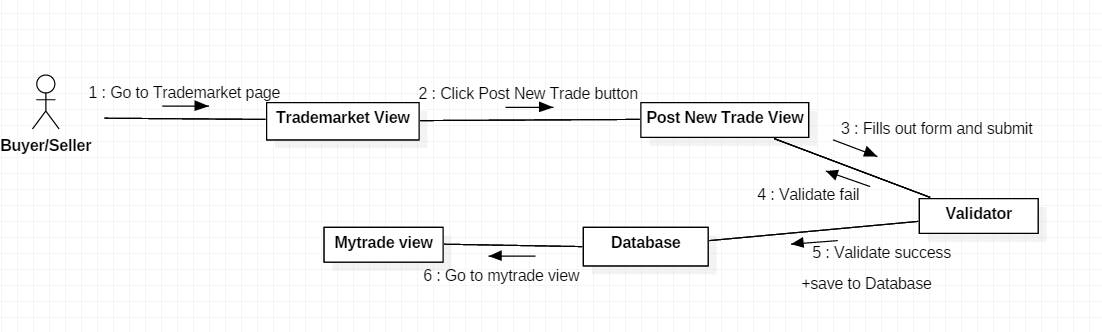
**5.1.1. Manipulate crawling**



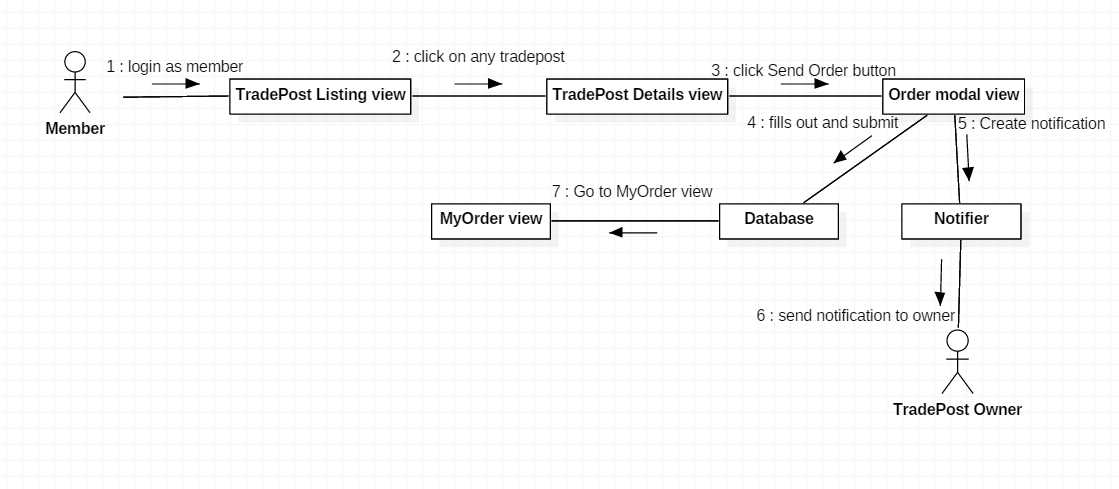
**5.1.2. Search for model information**



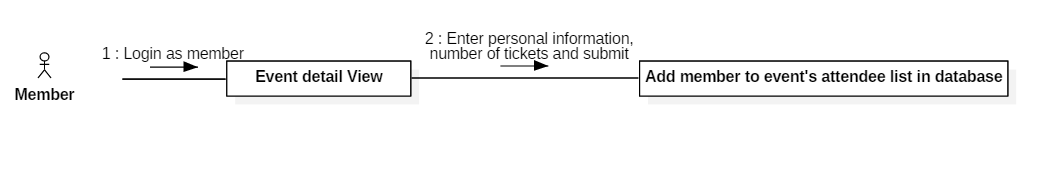
**5.1.3. Manipulate trade post**

****

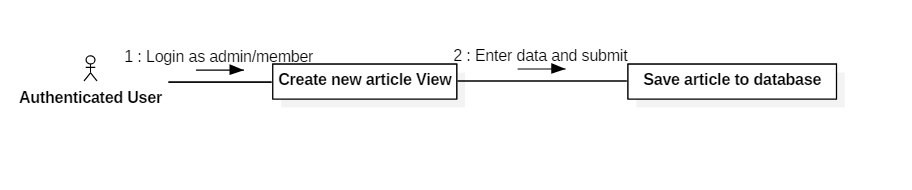
**5.1.4. Request order to trade post for exchange**

****

**5.1.5. Register for event**

****

**5.1.6. Create articles**

****

**5.2. Web and Mobile Application Test case**

**5.2.1. Manipulate crawling**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Result** | **Test Date** |
| AMC1 | Crawl models | 1. Login with role “Admin” 2. Go to “Crawling” view. 3. Click “crawl model” or “crawl articles” button. | List of crawling models/articles records is displayed |  |  |
| AMC2 | Review records | 1. CRUD records | The selected record information is updated when admin submit |  |  |
| AMC3 | Save records | 1. Click “SAVE” button | Save new records to database and display success message on the view |  |  |

**5.2.2. Search for model information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Result** | **Test Date** |
| SFM1 | Search for model information with filters | 1. Go to “Search Model” view. 2. Fill out the search form with filters 3. Click search button | List of matching records are shown with filters sort by and pagination |  |  |
| SFM2 | Choose filter sorts by and pagination | 1. Choose sort by options or pagination | Updated list records follow the filters and pagination |  |  |
| SFM3 | Show model detail information | 1. Click on a record’s “More information” button | The selected model detail page is shown |  |  |

**5.2.3. Manipulate trade post**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Result** | **Test Date** |
| MTP1 | Post New Trade with valid data. | 1. Click “Post new trade” button. 2. Fills out form with valid data and submit 3. Click “Post your trade” button. | Go to My trade view. |  |  |
| MTP2 | Post New Trade with invalid data. | 1. Click “Post new trade” button. 2. Fills out form with invalid data and submit 3. Click “Post your trade” button. | Stay at Post new trade view and show error message for invalid data. |  |  |

**5.2.4. Order to trade post for exchange**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Result** | **Test Date** |
| OTT1 | Send order with valid quantity | 1. Click “Send Order” button on trade post details view. 2. Enter valid quantity 3. fills out Trader information form 4. Click submit | Go to My order view |  |  |
| OTT2 | Send order with invalid quantity | 1. Click “Send Order” button on trade post details view. 2. Enter invalid quantity 3. fills out Trader information form 4. Click submit | Stay at send order modal view and show error message for invalid quantity.. |  |  |

**5.2.5. Register for event**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Result** | **Test Date** |
| RE1 | Register with a valid card | 1. Enter card number and all required information 2. Click on “Register” | Save information to database and a Popup shows up saying “Registration complete” |  |  |
| RE2 | Register with a invalid card | 1. Enter card number and all required information 2. Click on “Register” | Popup shows saying “Card is invalid!” |  |  |
| RE3 | Register with the amount of tickets exceeding the remaining tickets available | 1. Enter all required information with the number of tickets higher than the amount of remaining tickets for that event 2. Click on “Register” | Popup shows saying The number of tickets you’ve entered exceeds the remaining tickets amount! ” |  |  |

**5.2.6. Create articles**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Case Procedure** | **Expected output** | **Result** | **Test Date** |
| PA1 | Post an article | 1. Enter data in all fields 2. Click on “Submit” | Save information to database and a Popup shows up saying “Article submitted! Please wait for admin’s approval!” |  |  |