|  |  |
| --- | --- |
| Project name | IT events meetups management system |

|  |
| --- |
| 1. Domain object description |

4.1 All **Users** should have following common attributes:

package UserManagment

* id - (generated automatically) - long number;
* first\_name - string 2 to 70 characters long;
* last\_name - string 2 to 70 characters long;
* email - should be valid email address, unique within the system, cannot be changed;
* bio - string 2 to 255 characters long;
* is\_active – bool default True

4.2 Each **Role** has the following structure: .

package UserManagment

* id - (generated automatically) - long number;
* name - string 2 to 100 characters long;
* description - string 2 to 255 characters long;
* role\_type – enumeration including “Admin”, “Host”,” Guest”

4.3 Each **UserRole** has the following structure:

package UserManagment

* has\_a (relationship) user\_id long number;
* has\_a (relationship) role\_id long number;

4.4 Each **Group** has the following structure:

package GroupManagment

* id - (generated automatically) - long number;
* name - string 2 to 100 characters long;
* description - string 2 to 255 characters long;

4.5 Each **GroupUser** has the following structure:

package GroupManagment

* has\_a (relationship) group\_id long number;
* has\_a (relationship) user\_id long number;

4.6 Each **Event** has the following structure:

package EventManagment

* id - (generated automatically) - long number;
* name - name - string 2 to 70 characters long;
* description - string 2 to 255 characters long;
* creation\_date - date
* creation\_user\_id – long number
* registration\_end\_date – registration end date of the event
* start\_datetime – start date of the event
* end\_datetime – end date of the event
* place - string 2 to 255 characters long or url format;
* is\_public - bool
* capacity - int
* price – float value
* status\_id binded with event status

4.7 Each **AllowedEventGroup** has the following structure:

package GroupManagment

* has\_a (relationship) event\_id long number;
* has\_a (relationship) group\_id long number;

4.8 Each **EventInvitation** has the following structure:

package EventManagment

* has\_a (relationship) event\_id long number;
* has\_a (relationship) user\_id long number;
* sent\_date – invitation sent date
* has\_a (relationship) invitation\_response\_id long number;
* text\_response - string 2 to 255 characters long
* response\_date – invitation response date

4.9 Each **InvitationResponseType** has the following structure:

package EventManagment

* id – long number
* name - string 2 to 70 characters long
* response\_type – enumeration including “Accept”, “Reject”, “”Maybe”
* invitation\_response\_id
* text\_response
* response\_date

4.10 Each **EventTicket** has the following structure:

package EventManagment

* id – long number
* has\_a (relationship) event\_id long number;
* has\_a (relationship) owner\_id long number;
* is\_paid – bool default False
* paid\_date timestamp

4.11 Each **EventStatus** has the following structure:

package EventManagment

* id – long number
* name - string 2 to 70 characters long
* event\_status – enumeration including “Draft”, “Open to registration”, “ClosedToRegistration”, “Ongoing”, “Past”, “Cancelled”

4.12 Each **EventPost** has the following structure:

package EventManagment

* id– long number
* has\_a (relationship) event\_id long number;
* text- string 2 to 70 characters long
* creation\_date – date of the event post creation
* has\_a (relationship) creation\_user\_id long number – user id of the organizer

|  |
| --- |
| 1. UML diagram of relationships |

