

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

2. UML diagram of relationships

