Section 1 – General Explanation

* the project is an event-based logistics control and management system.
* it contains a registration and login system based on session and database.
* the system has three user roles (admin, manager, user).
* system flow works according to the event status :  
  opening an event by an admin → admin creates a number of responsibilities → admin assigns managers to each responsibility → each responsible person may edit a list of equipment within the responsibility given to him by the admin → any regular user can enter the event and choose which equipment he wants from any list of responsibilities → the manager of each list gives approval or cancels a user's request → after approval by the manager → any user who requested equipment receives it to his list and registers that he owns the same equipment → admin decides on a change of status (return of equipment) → administrators and users receive a notification that the equipment has begun to be returned → each user marks the equipment he wants to return → each manager checks that the equipment has indeed been returned and marks that he approves the return of each equipment → only after all the equipment of the responsibility has been returned can each manager confirm that everything has been returned → when all the equipment of all the responsibilities has been returned → only then can the admin finally close the event.

Section 2 – Main Pages and their Goals

* **Login & Register Page**

Authenticate existing users and allow new users to create an account. It ensures secure access to the system by verifying user credentials and storing new user information safely.

* **Event List\***  
  CRUD operations on events (admin role), viewing current active and closed events (manager + user roles).
* **Selected Event**

CRUD operations on Responsibilites (admin role), viewing responsibility list (manager + user roles).

* **Selected Responsibility**

Edit Responsibilty title and managers (admin role), CRUD operations on responsibility list (manager role), viewing , asking from responsibility list (user role).

* **User List (User Role)\***each user can view in a single page all the items he asked (waiting for manager approval status) and all items he holds (after manager approval).   
  also user can ask to return item from that list without going into the specific responsibilty list.
* **Manager List (Manager Role)\***

each manager can view his responsibilities in a list and access his responsibility from that page.

* **Roles List (Admin Role)\***

a list of all the users in the systems. admin can view and set a role of (manager or user[default]) each user registered.

**\***- pages can be accessed through the navbar.

Section 3 – Description of Database Beans : Classes and Relations

* **Event** | Represents events in the system  
  Attributes :
  + eventId (long) – primary key, auto-generated.
  + eventName (string) – name of event.
  + dateOfIssue (date) – timestamp of creation.
  + status (string) – current status of event.

Business Role: Central entity that groups related responsibilities

* **Role** | defines user role and permissions in the system  
  Attributes :
  + roleId (long) – Primary Key, auto-generated.
  + roleName (string) – role name (“admin”, “manager”, “user”).

Business Role: controls user access levels.

* **User** | users in the system with authentication and profile information

Attributes :

* + userId (long) – primary key, auto-generated.
  + emailAddress (string) – unique email for login.
  + phoneNumber (string) – unique phone number.
  + encryptedPassword (string) – for authentication.
  + dateOfIssue (date) – timestamp when user first registered.

Business Role: used for authentication, authorization and activity tracking.

* **Responsibility** | Represents a duty that has a list of items to manage

Attributes :

* + responsibilityId (long) – primary key, auto-generated.
  + responsibilityName (string) – responsibility name.
  + status (string) – current status of responsibility.

Business Role: links responsibilities with managers.

* **Item** | represents physical items that are managed within responsibilities.  
  Attributes :
  + itemId (long) – primary key, auto-generated.
  + itemName (string) – name of item.

Business Role: tracks resources and equipment that required management and can be requested.

* Request | user requests for items in the system (asking, returning)  
  Attributes :
  + requestId (long) – primary key, auto-generated.
  + dateOfIssue (date) – creation request timestamp.
  + requestType (string) – type of the request.

Business Role: tracks user interactions with items.

Relations between databases

* **User : Role** - Many to One ( many users can have the same role )
* **User : Responsibility** - One to Many ( if user is a manager he can be signed to multiple responsibilities )
* **Event : Responsibility**- One to Many ( event can have multiple responsibilities )
* **Responsibility : Item** - One to Many ( responsibility can have multiple items )
* **User : Item**- One to Many ( user can hold multiple items )
* **Item : Request**- One to Many ( item can have multiple requests )
* **User : Request**- One to Many ( one user can make multiple requests )

