1. **Proposed System**

We are going to try to develop a social network that is completely free. This web site is user friendly , and useful for everyone. Everyone can register in web site for free. The point is creating a communication between the school’ clubs and students and everyone. You do not have to study at the same school. You can still register web site.We will try give all clubs information and clubs events information to visitors and registered users.

**3.1 Overview**

Clubs Events is a online platform for communication between the Işık University clubs and students and everyone. The system also has an admin, whom manages the system and has full authentication to the system. The admin also sets who will be moderator and who will add comment on web site. The next section( Functional Requirements) contain a list of all the functions about what any actor will be able to do in the system . Nonfunctional Requirements decribes the systems behavior, which will affect the system success criteria and performance , and includes system , interface and implementation details. Basically, a functional requirement describes what our system should do , while non-functional requirements place constraints on how our system will do so.

**3.2 Functional Requirements**

This section contains what each of our actors in the system must be able to do in the system.

Functional Requirements describe the high-level functionality of the system. They are the requirements the system must have, and corresponding actors are able to perform. Each actor has different functional requirements, and there are some common requirements for all actors. These common functionalities are associated with account-related functionalities.

**Functional Requirements shared by all actors:**

* All actors in the system can see all clubs’ information and clubs’ events and details.

**Functional Requirements for the Normal User:**

* Normal User shall be able to see all clubs and events
* Normal User shall be able to comment clubs’ events.
* Normal User must have a password recovery option in case they forget their password. And are unable to authenticate, with their password reset and sent to them via e-mail.
* Normal User can create own favorite clubs list and can show favorite clubs’ events.

**Functional Requirements for the Moderator:**

* Moderator shall be able to can access Moderator Panel.
* Moderator can add comment clubs’ events.
* Moderator can add club on web site.
* Moderator can edit club on web site
* Moderator can disable club on web site
* Moderator can create event on web site
* Moderator can edit event on web site
* Moderator can disable event on web site

**Functional Requirements for the Admin:**

* Admin shall be able to login
* Admin shall be able to create a new admin
* Admin shall be can access Moderator Panel and Django Admin Panel.
* Admin shall be able to display personal information and account information of Clubs, Events and all users
* Admin shall be able to add, edit or delete clubs information from the system.
* Admin shall be able to add, edit or delete event information from the system.
* Admin shall be able to add, edit or delete user (both Normal User and Moderator) information from the system.
* Admin shall be able to delete comments.
* Admin can add comments clubs’ events.
* Admin can check events and clubs status.

**3.3 Nonfunctional Requirements**

This section describes user-level requirements those are not directly related to functionality.

**3.3.1 Usability**

Clubs & Events will include just one interface. Every users use same interface. This interface will include all the functions an actor must perform. For usability, this interface should be easy to use and understand, with the actor being able to perform their tasks without explanation. They should also be appealing to the eye while being simple in design. Users should be able to reach their goals with minimal number of clicks.

**3.3.2 Reliability**

Clubs & Events shall be secure, and will not allow any allow any unauthorized user to enter the system. Only users permitted and defined by the admin in the database shall be able to authenticate to the system, using their account information. Visitors should be able to do some stuff. Yet, registered visitors which are normal user and moderator , should be provided with unique e-mails and password that are appropriate for password criteria. The password should be hit as hashed password in database. Also visitors can register the system. We are designing the web site for everyone. All information will keep Django Framework database. . In addition, for it to be reliable, the system should not be down or crash in case of errors, and should be running 90% of the time.

.

**3.3.3 Performance**

Clubs & Events is designed to be used by multiple numbers of users. And also, should be responsive in different devices which are mobile phones, tablet and computers. The web-site is going to be dynamic content, so there should not be complicated queries in back en to not decrease performance. Responsive time should be fast, with a maximum of 4 seconds waiting time in regular site traffic. HTTP GET or POST request may be used to receive information and feedback, and task completions of users without reloading the page.

**3.3.4 Supportability**

The system should be managed by admin. The system should be reachable over any browser (maybe with the exclusion of older versions of internet explorer) in a standard computer. Management and maintenance of Clubs & Events belongs to the system administrator. Maintenance of Clubs & Events should be able to be done easily, and in certain time intervals. Interfaces for database altering, tables and instances should be shown in administrator panel.

**3.3.5 Implementation**

The system will be implemented on Django Frameworks. Python and HTML will be used as the programming language and Pycharm will be used as a framework in the process. . We will be using a “code-first” approach, which means we will construct the Models first and the virtual relations between them, and the database will be created based on these models. Model-view-controller design methodology is going to be implemented in this project. We will check input over controller, our model is satisfied object relation model over SQL.HTML5, CSS and Boostsrap should be used to implement. User Interface should be web-based (accessible via WWW Browser).

**3.3.6 Interface**

The interface should be easy to use for all actors, as usability is the main concern. There should be no guide to instruct the actor about what and how to do their tasks. The interface shouldn’t be too colorful and eye-tiring, instead a simple color scheme should be used. The colors of school might be used. Menu navigation should change according to actor. There will be a side menu navigation in our interfaces. Similar tasks and functions should be put together.

**3.3.7 Packaging**

Admin should install the system. Also, the system is a web site so the site should uploaded on the server. There should no time constraints on the installation. We are planning for the project to be used by the school clubs and SKS(Department of Health, Culture and Sports), with the project fulfilling the functional requirements of entering clubs information , events information, but other , but other constraints on the actual delivery of the system are not determined. Thus, packaging requirements will be decided in the future.

**3.3.8 Legal**

The Clubs & Events does not use any license or licensed stuff. It is a social network. The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and no infringement.