LinkedHU_CENG
System-Wide Requirements Specification

1. Introduction

This article presents the LinkedHU CENG web project's system-wide needs. To create a better and more efficient mechanism for networking with the Hacettepe University Society, LinkedHU CENG is requested. In this paper, the functionality and requirements for this web project will be outlined.

2. System-Wide Functional Requirements

- Functional
    - Auditing
        - Keeping track of who and when is utilizing the system. It guarantees that the company's varied objectives are met. The capacity to provide campaigns during the hours when the system is used the most, for example, necessitates keeping track of who utilized the system in this project.
    - Authentication
        - The user must use the e-mail address and hashed password that are stored in the database to get access to the system. Logging into the system requires this credentials.
    - Printing
        - There is no requirement for the system to be able to print. Everything will be done over the internet.
    - Reporting
        - Which users are utilizing the system, which users are posting job ads, scholarship announcements, or internship announcements, and the ability to report on the system are all required to discover answers to inquiries and provide information on these topics.
    - Scheduling
        - Online meetings and presentations will be planned using the system. The system will arrange an online meeting or online presentation when a user makes one.
    - Security
        - Some data in the system and database will be restricted, and access must be kept safe. When a user registers with the system, personal information is protected.

3. System Qualities

3.1 Usability

Ease of learning:

- *The system can easily learn and operate by each University student*

Task Efficiency:

- *Users should be able to access certain scenarios in a short time*

Ease of Remembering:

- *We will try to convey certain scenarios in the simplest way in the application we created,so that the use is memorable*

3.2 Reliability

Availability:

- *Unless there is severe failure or menantance system will be available always.*

Recoverability:

- *Mean time to recovery wil between 12-24 hours.*

Frequency and Severity of Failures:

- *Critical defects:*
  - *Inability of use app totally or partially. For example, crashing of the meeting events.*
- *Significant :*
  - *Bugs in operation.For example taking a long time for a message to be sent*
- *Minor :*
  - *Bugs which is simpler to solve and has less effects to usage, For example Mistakes of interface or notification problems*

3.3 Performance

- *The maximum response time between user and system should not takes more than 5 seconds*
- *The start-up and shutdown operations will takes between 1 and 5 seconds .*
- *The system will be capable of dealing with different user request at the same time.*
-

3.4 Supportability

Adaptability:

- *The system can easily adapt to new environments and can be upgraded.*

Compatibility:

- *There are no special requirements regarding this system.*

Configurability:

- *The System will be configured at maintenance times*

Installation:

- *There are no special requirements regarding this system.*

Scalability:

- *The system will increase its database capacity and network when the volume goes up*

Testability:

- *There are no special requirements regarding this system.*

Maintainability:

- General maintenance will be carried out at six-month intervals.

Level of Support:

- We will prepare frequently asked questions and documentation.For a more specific problem we will take the questions via email

4. System Interfaces

Our system interface is a simple HTTP-based API. It is also a monolithic web application, which means that all the code is in one place. It is easy to add new features, and to make changes to the code. This web application has front-end, back-end, and database layers. Backend is written in Spring, and front-end is written in React.

4.1 User Interfaces

Our user interface is a web application, which is a single page application. It is written in React. For the state management, Context API is used.

Example UI Views

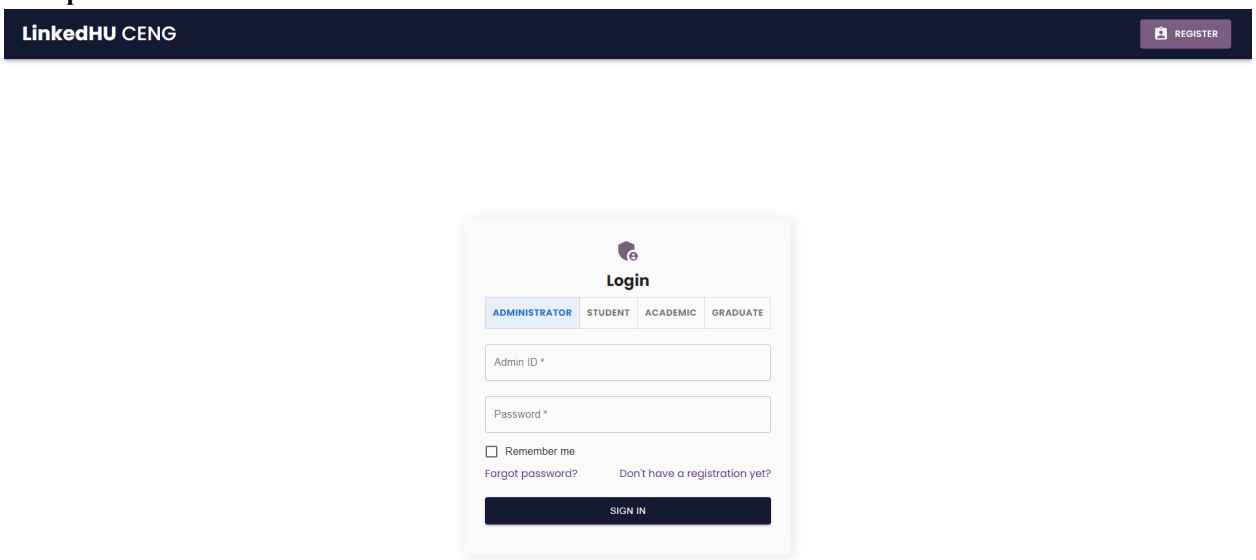


Image 1: LinkedHU_CENG Login View

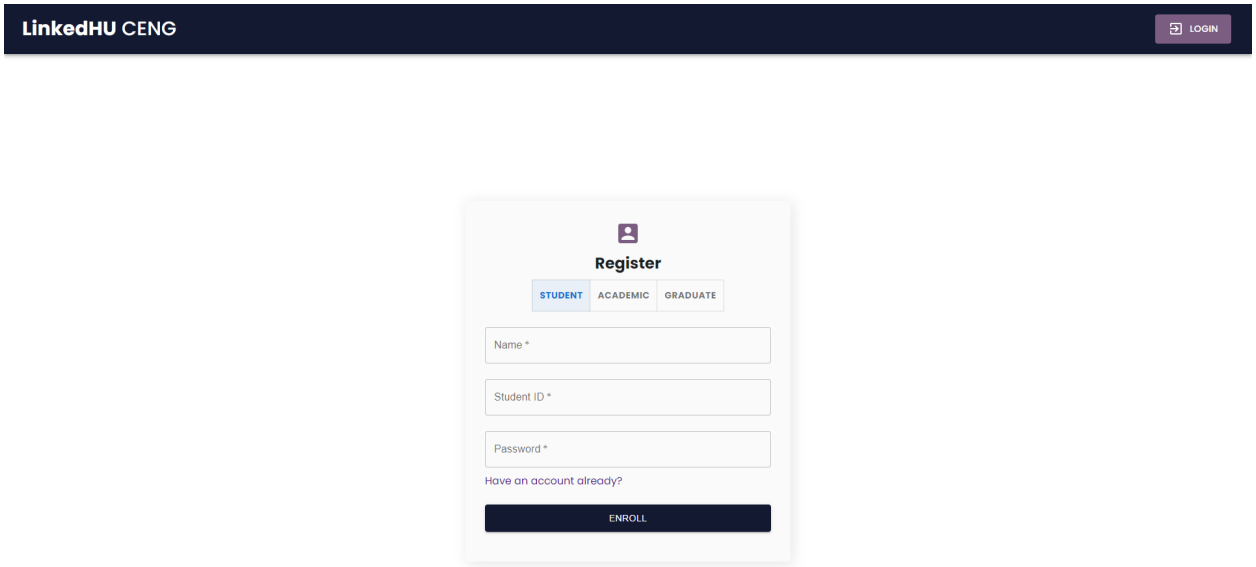
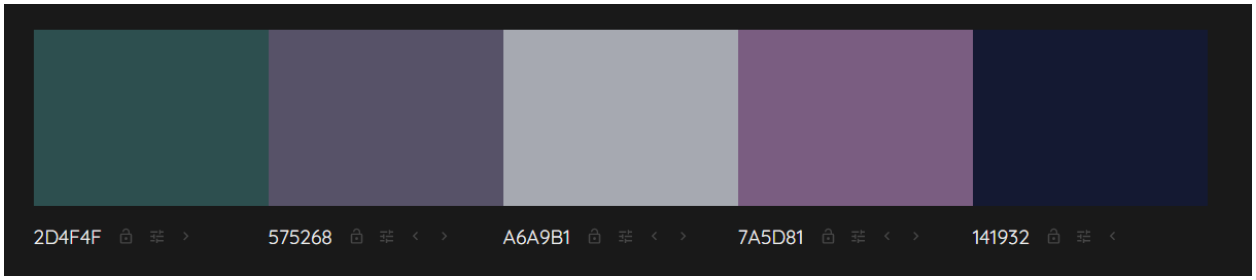


Image 2: LinkedHU_CENG Register View

4.1.1 Look & Feel

This system is practical to use. The interface is developed using HTML, CSS and Material UI Design. The responsive design that comes with Material UI helps users to use it on wider and narrower screen sizes. For user experience, many social media platforms are reviewed and an observation and elaboration is added accordingly.

Color Palette for Design



4.1.2 Layout and Navigation Requirements

This application will use a nav-bar and side-bar for navigation. On the nav-bar there will be a search, link to access user details, and several navigation links. In the side-bar section, there will be links and plugins found in social media applications, which are faster and generally easier for users to use. **react-router** and **react-hooks** dependencies will be used for navigation operations.

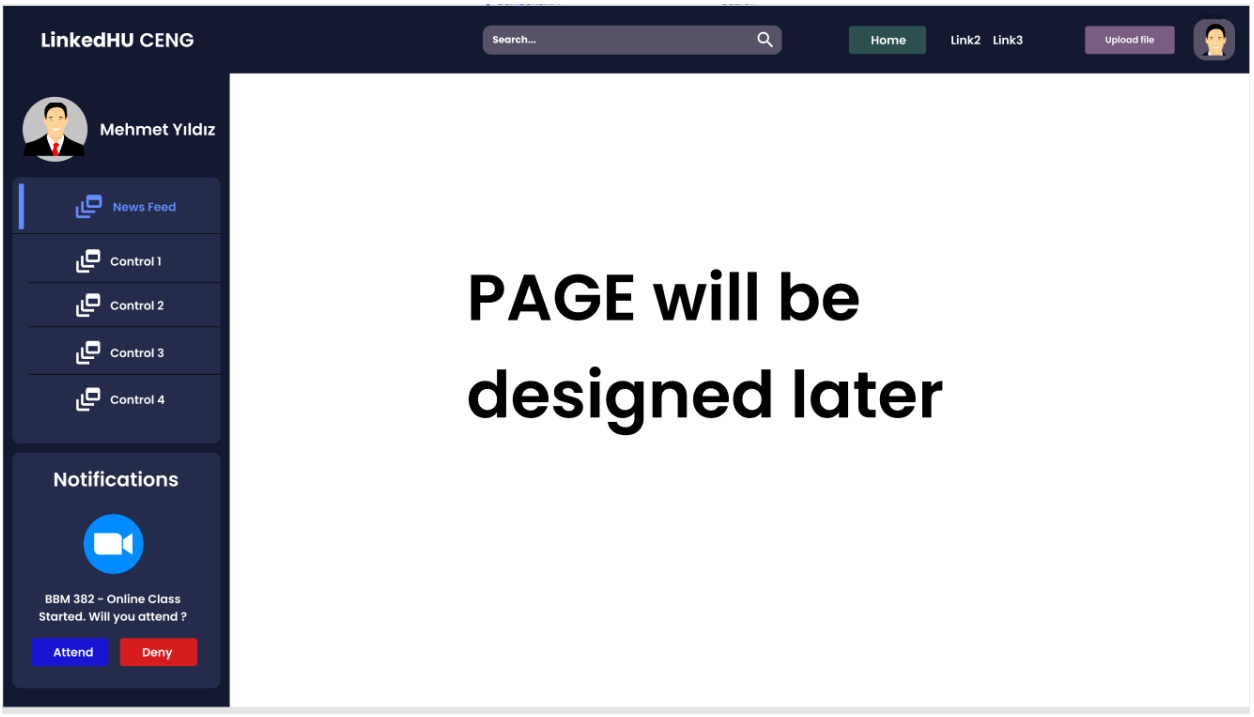


Image 3: LinkedHU_CENG Navigation Design View

4.1.3 Consistency

In this application, to improve the user experience, color selection, component designs, and animations have standards that the user is familiar with and that makes it easy to use. For this reason, the user will be able to use the system more quickly and practically. All navigation, colors, and components have been tried to be designed most appropriately as a result of examining many designs.

4.1.4 User Personalization & Customization Requirements

- Although there is a theme feature to change between dark and light that comes with Material UI, we are still in the decision stage to add it to the project. If we add, the user will be able to choose and customize both dark and light themes.

- The user can add a personal photo or avatar

4.2 Interfaces to External Systems or Devices

4.2.1 Software Interfaces

- Only **Material UI** (see: https://mui.com/) is used as an external interface. It is free to use and open-source. So we can implement it easily.

4.2.2 Hardware Interfaces

- No hardware interfaces needed.

4.2.3 Communications Interfaces

- No communications interfaces needed.

5. Business Rules

5.1 User Control

5.1.1 User Deletion: Deletes the students who are dismissed from the university and the academicians resigned from his/her faculty member profession.To delete an account, you must be an administrator.

5.2 Make an announcement, presentation vs.

5.2.1 Create online Meeting: Academics and graduates can create an online meeting, but students cannot create an online meeting. Student representatives can manage the meeting program only with the permission of academicians and graduates

5.2.2 Upload video, presentation, lecture notes: In order to upload videos, presentations, lecture notes, papers, the user must be an academician or graduate

5.2.3 Make presentation: In order to make online/offline presentation, the user must be an academician or graduate

5.2.4 Make announcement: Academics and graduates can publish announcements, but a student must be a student representative in order to publish an announcement.

6. System Constraints

- The system we're working on may either be a web app or a mobile app!
- For our system, the Model-View-Controller (MVC) architectural pattern must be employed.
- In order to evolve, our system must have a minimum of 9 and a maximum of 12 use-cases.
- The system must be developed using the JAVA or C# programming languages. We were also set on using Java.
- There are no limitations when it comes to database administration. Any database management system, such as Oracle, MySQL, or SQL Server, can be used.
- To link the database and the software code, we'll have to utilize the JAVA or C# programming languages.

7. System Compliance

7.1 Licensing Requirements

We will only be using licensed open source applications and programming languages.We will pursue what the applications wants us to comply with.

7.2 Legal, Copyright, and Other Notices

Our aim is to do most of the work to be our own work, and in case we benefit from any outside product(such as a template), we will take necessary actions(such as paying for rights) to prevent any copyright problem from rising up. All the copyrights will belong to Teamscript.

7.3 Applicable Standards

Standards such as Java, React, Mysql are followed in our project. We highly value coding standards.

8. System Documentation

Uğur Can KUŞOĞLU - Introduction and System-Wide Functional Requirements
Cavit Bora ÖZTEKEŞİN - System Constraints and System Compliance
Meryem Ülkü KARA - Business Rules
Mehmet YILDIZ- System Interfaces

Yusuf Sercan TUNK- System Qualities
Use-Case Diagram; Yusuf Sercan tunk, Meryem Ülkü Kara
Context Diagram- Cavit Bora Öztekeşin
E-R Diagram-Uğur Can Kuşoğlu