

Prototyping:

Before we start:

The way we imagined the program will work is that we will deliver the web application with a premade admin account. From there, admins will be able to create user profiles for the organizations. We did not make users create an account themselves because a user could make himself an admin account, then he would have access to files he shouldn't see.

1. Login page

Figure 1 is our draft version of the login page of the program. As in any application at first there is always a login page, so basically this is also the initial page of our program. In here there are 2 textboxes and 1 button. The 2 text boxes are for users to enter their username and password. Next, there is a login button. When the login button is pressed the application will check whether that username exists in the program's database, then it will check whether the user has entered the correct password. Next, after all checks are completed successfully, it will show the user the main search bar page.

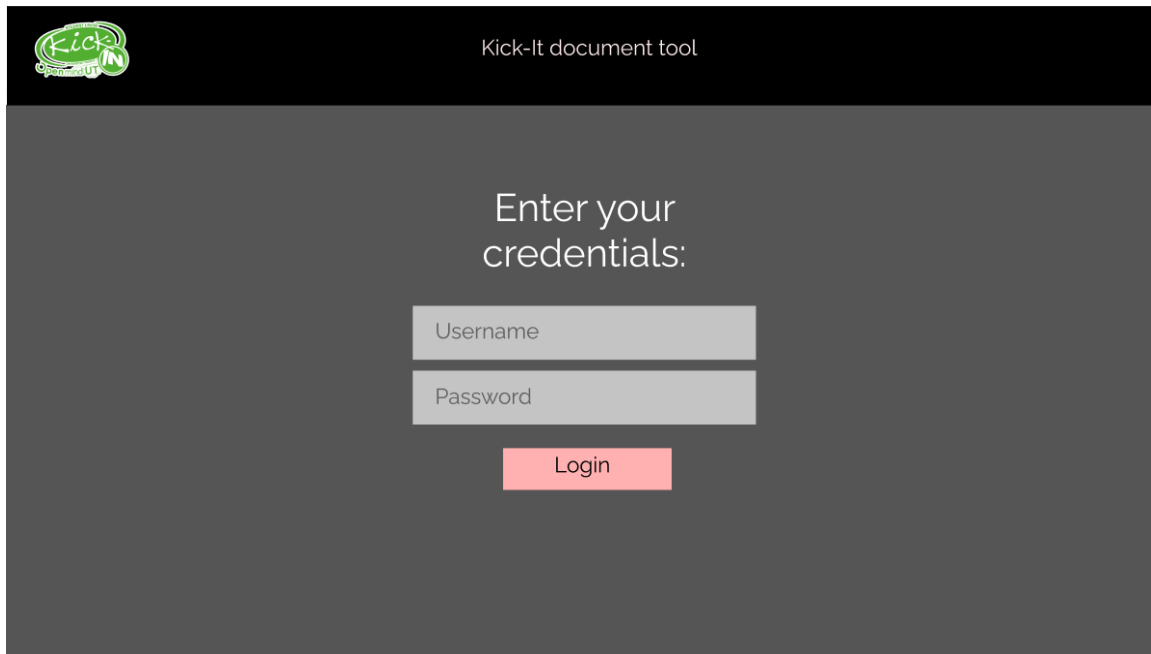
The image shows a web application login page. At the top, there is a black header bar. On the left side of the header is a logo for 'Kick-It' with 'Gponit-IT' written below it. On the right side of the header is the text 'Kick-It document tool'. Below the header, the main background is a dark gray. In the center of the page, the text 'Enter your credentials:' is displayed in a light gray font. Below this text are two light gray rectangular input fields. The first field is labeled 'Username' and the second is labeled 'Password'. Below these two fields is a red rectangular button with the text 'Login' in white.

Fig. 1: Login page

2. Search / Main page

On this page (see *Figure 2*) there are a couple of tools. First, there is a search bar where the user will type a document name or an attachment name. Next, after the name to search is chosen, the user will press the “search” button and the results will appear below it. In the result, the application will show a table, in which the application will show sender names and subjects of the messages. It will also show attachments that some messages could have. For every message there are: a “view”, a “download” and an “edit access” button.

The view button will view the document online, while the download button will download the document to the user’s local device. The edit access button will only be accessible to admin users. They will be able to change any documents access levels. Then, there is a menu option. After a user presses the menu option, a pop-up bar will be shown. There is an upload database button in the Menu which will be available to use only for users who have access. When the upload button is pressed it will ask for a message to upload. There will be a “Log Out” button and an “Admin page” button. The log out button will make the user go back to the login page because it will log out the user. The Admin page button is only for people who have access to use that button. After a person with proper access presses the Admin page access button it will forward the user to the Admin page.

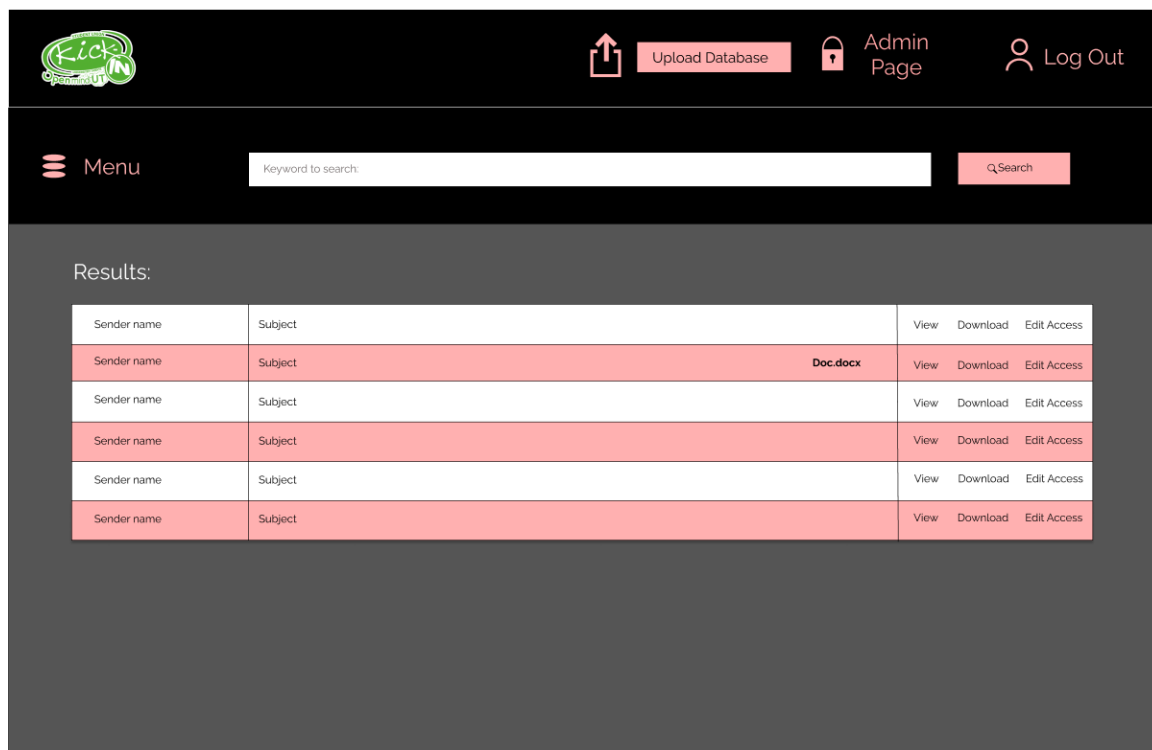


Fig. 2: Main/search page.

3. Admin page

This page (see *Figure 3*) is for stakeholders to search a user from the list of users in the database and to create new users. When a username is chosen to search in the results it will show the persons username, association and his access level. From there, a user with admin rights can edit the other user's access level to documents. The admin can also delete the user from the database. There is also a menu option, which also triggers a toolbar to be shown on screen. From there, a user can also log out with the "Log Out" button, and they could also go back to the search page through pressing the "Search Documents" button. Lastly, there is a "Register New Person" button. When an admin wants to create a user he presses this button, and it will take him to the "register page".

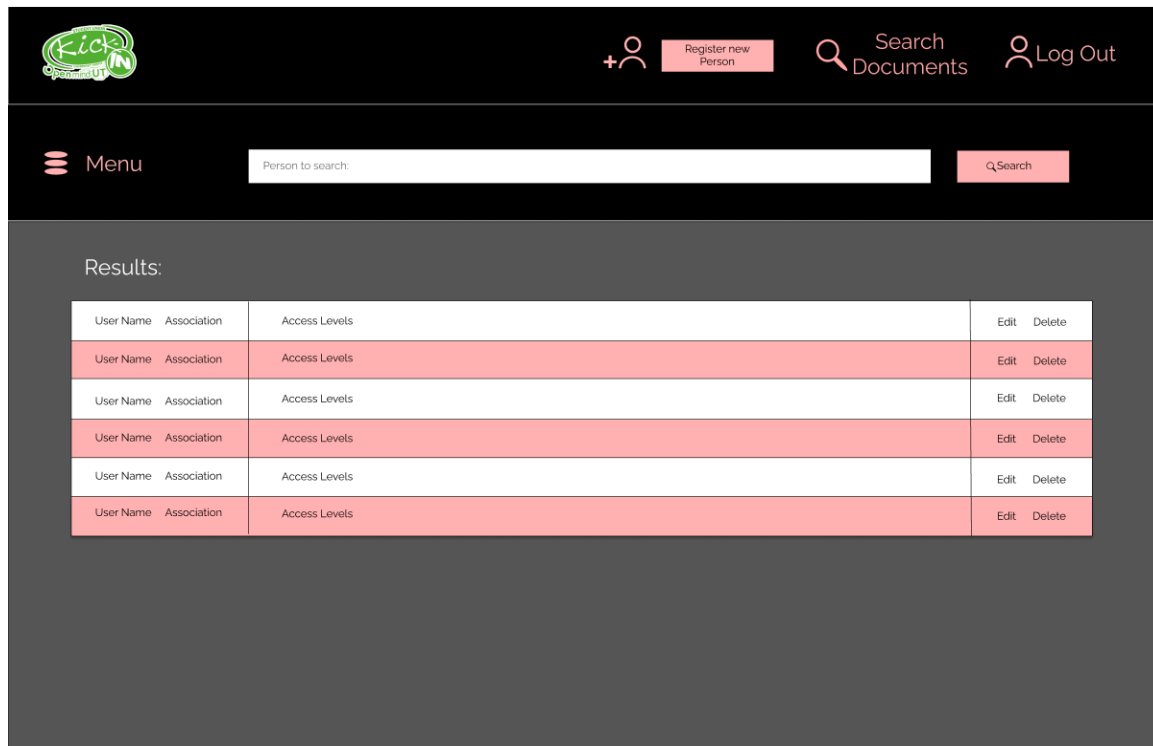
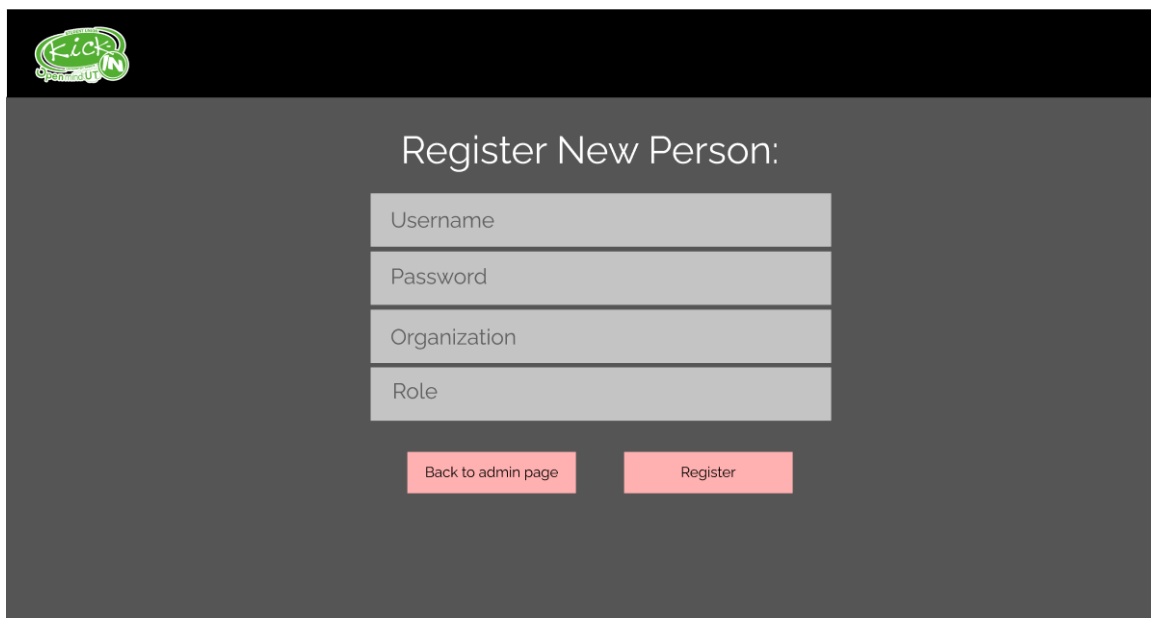


Fig. 3: Admin/ User management page.

4. Register page

On this page (see *Figure 4*) an admin will be able to sign up users. On this page there are 4 text bars. The first 2 are for writing down a username and password. Then there is a text bar for writing down which organization the new user belongs to (this might become a dropdown menu in the actual app). And finally, there will be a text bar for a role. These 2 information are necessary to determine the access level of the user. Because, for example, if a user role is a student, then that user will have limited things they can do on the program. And if a user role is for example, a Kick-IN committee member then they will have access to all the programs features. After the sign-up form is filled the user will press the “Register” button, and a new user with a password will be added to the database. If the admin wants to go back to the admin page he will just press the admin button.



The screenshot shows a web interface for the 'Kick-IN' app. At the top left is a logo with the text 'Kick-IN' and 'Open to All' below it. The main heading is 'Register New Person:'. Below this heading are four stacked text input fields labeled 'Username', 'Password', 'Organization', and 'Role'. At the bottom of the form are two buttons: 'Back to admin page' and 'Register'.

Fig. 4: Registration page.

Usability testing

For our prototype, we asked a fellow student named Duru, who is a member of Inter-Actief and Kick-IT. We gave her small tasks such as: finding the “upload database” button and “edit user access” button. Throughout the process she gave feedback on the details of the design. The very first thing she mentioned was that the register page looked unprofessional for reasons she couldn’t explain and that it looked too similar to the login page which could cause some issues. She said she liked the login page, but that we should probably change the color scheme to something that fits kick-in more. She also mentioned we have not added any button to go to the register page.

After she successfully reached the main page, she mentioned that we repeated the search word too many times and advised us to minimize the use of them, if possible. She also implied that most websites have their menu buttons on the left and told us that we might want to change it. She liked how the results looked, and said that we should maybe add a button to grant specific access, not only per person, but also adding specific access to documents. She liked the design of the menu and its buttons.

Finally, when she reached the current last page of the website, she said that the “upload file” button was confusing, and at an awkward place. Other than that she also implied that more information about a person should be shown on the admin page, so that it is easier to distinguish between users and manage them. She also mentioned that a “delete user” button might be necessary here. She also told us to be careful on the default search bar text as it might confuse users otherwise.

Conclusions and changes

The first thing she mentioned was the design of the register page looking too similar to the login page. The first change that we did was to change the design of the register page we put a heading in that would clearly indicate to any user that this is a register page and not a login page. Next, as from the feedback we got, the place of the upload button was in an awkward place, so we decided to put the upload button to the menu bar in the search page next to the other tools. Our third change was to add a register button in the admin page since we forgot to do that before. Now an admin will be able to go to register page through admin page. Since she also mentioned that besides user access level editing we should also have individual document access editing we added an edit button to the main page where an admin will be able to change the access level of any document he searches. One of the things that was mentioned was that search buttons were looking different in admin and search pages, so we put the same design to both search buttons. The last thing she mentioned was the color of the prototype, we discussed the color problem within our team and decided that since this is just a prototype we will be able to change colors when we will start implementing the web app.