

Assignment 3

Topicus Team 1

Umair Aamir Mirza (s2928558)

Martin Demirev (s2965046)

Condu Alexandru-Stefan (s2769549)

Narendra Setty (s2944200)

Alexandru Lungu(s3006301)

Teodor Pintilie (s2920344)

Introduction

This report covers the progress made on the User Stories, particularly their functionality testing and assessing how they reached the requirements that they were intended to meet. It will include the necessary details regarding the development and testing wherever possible, including ways that testing was achieved.

User Story Testing: Guardian-Related

1. As a parent, I want to be able to see my registration(s) so that I can check whether the details I have submitted are correct.

During the testing phase, a group of fellow students was engaged to assess the application's functionality and validate the user story. The specific focus was to ascertain whether accessing the "View Registration" feature consistently led to the presentation of the corresponding registration received from the back-end. In all diverse scenarios explored, the transmitted data was received successfully and reflected accurately within the application. It should be noted, however, that while the data integration and display mechanisms performed effectively, the visual elements of the application are still undergoing refinement. req works 2nd time

2. As a parent, I want to be able to see the status of my registration form so that I can check whether my child has been accepted.

In order to conduct the testing of the aforementioned functionality, it was determined that external perspectives were not required due to the simplicity of the feature. The testing process was carried out by the group colleagues, and it was observed that the status is accurately displayed at each stage of the registration process. However, it should be noted that the full implementation of the color-changing feature based on the registration status is still pending.

3. As a parent, I want to be able to delete my child's registration in case my child is no longer attending a particular school.

To validate the functionality related to the deletion of registrations, our team enlisted the assistance of fellow students who participated in testing procedures. In order to facilitate the testing process, a set of "dummy" registrations was created to

assess the functionality's performance. It should be emphasized that specific registrations were intentionally configured with the deletion property denied, enabling us to verify whether parents are indeed restricted from deleting unauthorized registrations. The testing phase yielded successful results, as the functionality performed as expected across all chosen scenarios evaluated by our testers.

4. As a parent, I want to be able to submit a registration form in order to register my child with a desired school.

The feature in question was tested in conjunction with the previously mentioned functionality. The creation of "dummy" registrations was carried out within the registration system itself, eliminating the need for external testers. The testing was conducted internally within the group, and the registration process proceeded smoothly without encountering any issues.

The user story regarding the Apply for Registration procedure has 6 components:

1) At first when a guardian navigates to Apply for Registration, they may see two different formats, as it depends on whether they are already logged in or not. They can log in either with an account or their Guardian Code that they obtained after applying before if they still do not want to create an account. However, there is a case in which a guardian might be applying for the first time and not want to create an account. This is possible, but it will also affect the steps 3) and 4). After a guardian has applied successfully, they will be able to log in with their Guardian Code in the future. It is crucial to mention that this code will be displayed to them only once, therefore they will have to save it, as it is quite long to ensure security.

2) Then the guardian will have to choose a school they want to apply to. All schools which are using the platform will be automatically displayed and the guardian can either type the name of the school they are looking for or click on its card. After that, they have to submit their choice which will redirect them to the first page of the particular school's form.

The only thing which is left to be implemented and tested for this step is the correct connection between the chosen school (and its registration form) and the actual loading of the corresponding form components, as they are still hardcoded.

3) On the first page of the Registration Form, the guardian will have to fill out their data. As stated in 1), there will be two cases which are yet to be implemented:

1. If they are applying for the second (or more) time(s) with an account or just Guardian Code - the mandatory Guardian Details fields that no school can change will be omitted, as the data is already stored in the database. Hence, only the **extra** fields will be displayed to them.

2. If a guardian has just created an account and logged in **or** they are applying for the first time without logging in - they will see all of the fields of the Guardian Details part.

4) When they proceed, on the second page the guardian will have to input their child's data. The first option is still not implemented, as the Account Management will have to be finished in a sufficient form for it to be realised. As explained in 1) and 3), there are two cases again:

1. When a guardian has already sent an application for their child to a school and wants to send another one or more to other schools, then there will be an option to choose if the application will be about the same child. In this case, the mandatory fields will be omitted again and only the **extra** ones will be displayed.
2. If the application is about a child for whom they have not applied before, then subsequently, they will have to fill out all fields. The same holds for all guardians who apply without an account or Guardian Code.

Note: Both 1. in 3) and 4) are to prevent data redundancy in the database, hence, no more than one object is related to either a guardian or a child. However, this still depends on the guardians, as they will have to follow the steps and might apply every time without an account or Guardian Code. Furthermore, "**extra**" are called the fields which every school can add to their registration form and will (possibly) be different from any other school's.

5) On the third page of the registration form, the guardian will have to fill out all remaining sections with fields (if any) that a school has added. If there are none, this page might be automatically skipped.

6) The last step which is presently in development, is the Preview page on which the guardian can preview all of their input on a single page and edit it if needed. When they click Apply and it is successful, a Guardian Code will be displayed to them which they will have to save possibly for later use. If they are already logged in, this will be omitted.

User Story Testing: School Administrator

1. **As a school administrator, I want to be able to view all registrations submitted for this school, so that I can stay updated about the latest developments.**
 - a. This feature has been successfully implemented on both the back-end and front-end. The back-end contains the necessary resources which produce RegistrationViews for a particular school of the school administrator. These views are then processed and presented on the

screen in the list format, enabling the school administrator to view all the registrations.

- b. Testing for this user story was quite simplistic; different requests with the front-end were made to ensure that the list was produced in a legible and appropriate manner, further design changes are awaiting development.
- 2. As a school administrator, I want to be able to modify/delete registrations to respond appropriately to registrations.**
 - a. The functionality for a school administrator to be able to conduct such modifications is in place on the back-end, however, the front-end JavaScript still requires some testing to ensure that the functionality is working.
- 3. As a school administrator, I want to be able to accept or reject registration requests from parents to inform parents if their child has been accepted or not.**
 - a. The functionality for a school administrator to be able to make these changes is in place on the back-end, however, the front-end JavaScript still requires some testing to ensure that the functionality is working.
- 4. As a school administrator, I want to receive requests for updating or deleting a student's registration before any final action is taken.**
 - a. On further inspection, the statuses are more so used to manage what parents can do with a particular registration. Hence, there is no need for the requests, as a Parent can contact an administrator if they would like to explicitly revoke the application if it is in a state that the school administrator is still investigating it. Otherwise, a parent can delete a Registration with no consequences.
- 5. As a school administrator, I want to be able to close the registration form based on dates so that the school can more easily accommodate the appropriate number of students.**
 - a. This is possible during the construction of the Registration Form itself, which will enable the administrator to hide the form when they so choose.

User Story Testing: General User

1. As a user, I want to be able to log in securely to my account to perform my desired tasks.

For security purposes when the user creates an account and sets a password, we utilize the Argon2 hashing algorithm to convert the password into an irreversible and unique hash value. This process ensures that even if the stored hashes were somehow obtained, it would be computationally infeasible to retrieve the original password.

In order to further improve security we are using salt which we add to the password before hashing it. This salt is stored securely alongside the hash in our

database and significantly increases the complexity of potential attacks, such as rainbow table or dictionary attacks.

Therefore, when the user inputs their password in the application on the login page, that password is hashed with the same salt and we compare the hash of the entered password with the stored hash in the database. This process verifies the authenticity of the provided credentials without ever storing or transmitting the actual password.

On the Front-End, as set on Back-End, both username and email are taken alongside the password, so that the uniqueness of the user's account is increased.

In the other case - of logging in with just a Guardian Code - it was needed to migrate the ID-s in the database to the "cryptographically strong" UUID-s. They ensure that guessing a Guardian ID is nearly impossible. Moreover, a check has been added to prevent the UUID-s from generating the same ID for more than 1 row.

Conclusion

To conclude, the efforts made on the User Story Testing were extremely useful in assessing what work remains to be done. The most pressing matter to deal with is the management of the interconnectedness of the application, as the components are functionally independent of one another. However, once requests and tokens are sent appropriately, management of the information and linking the pages together with their functionalities shall happen without any unforeseen issues.