

Milestone 1
E-Shop Web Application

Name	ID
Fareeda Mohamed Ali Abouzed	19016154
Rowan Nasser Edrees	19015686
Basant Yasser Salah Beshir	19017262
Mayar Ayman Mahmoud Ali	19016744
Nada Mohamed Ibrahim	19016782
Toka Ashraf Abo Elwafa Ahmed	19015539

Table of Contents:

1. [Functional and non-functional requirements](#)
2. [Detailed Use case diagrams](#)
3. [The flow of events for each of the use cases using activity diagrams](#)
4. [Jira report:](#)
 - Current Jira Roadmap
 - Milestone sprint listing the included requirements Jira Stories
 - Stories, tasks, and subtasks estimates
5. [Git & GitHub report showing:](#)
 - branches diagram showing the main branch, milestone branch, and feature branches
 - Pull Requests
 - Code Reviews comments

Functional and Non-functional:

a) Functional Requirements:

1. **Register:**

- This method is done only when the user isn't registered before the system and makes a new account. He only needs to fulfill the following fields to register in the system: (First name- Last Name - Password- Telephone Number - Email - Date of birth (he must be 15 or over)).

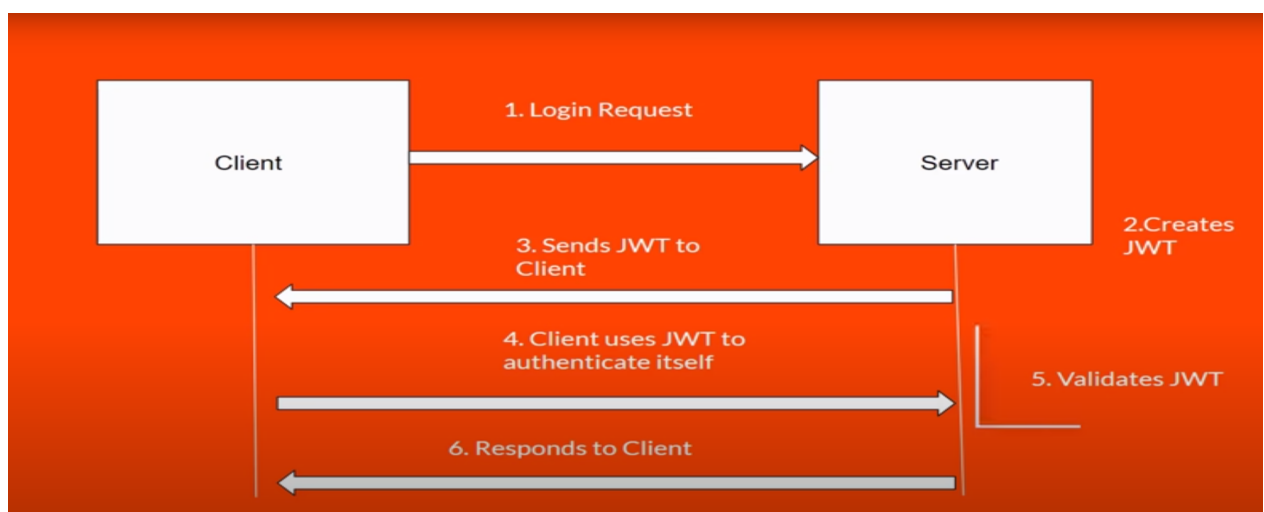
2. **Login:**

- when the user was registered before and wants to log in to his account. He only wants to provide his email and password. the data sent to the backend to be verified (whether it was registered before or not) and give the proper authentication to the user (detailed explanation in Authentication and Authorization part).

b) Non-Functional Requirements:

3. **Authentication and Authorization:**

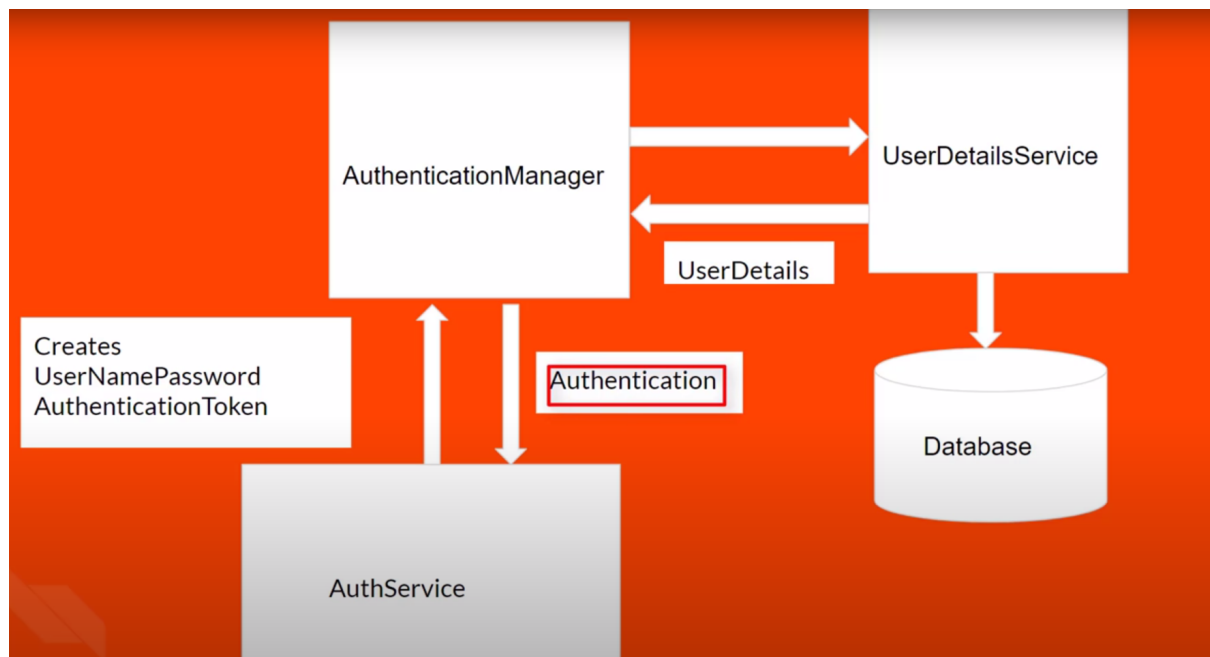
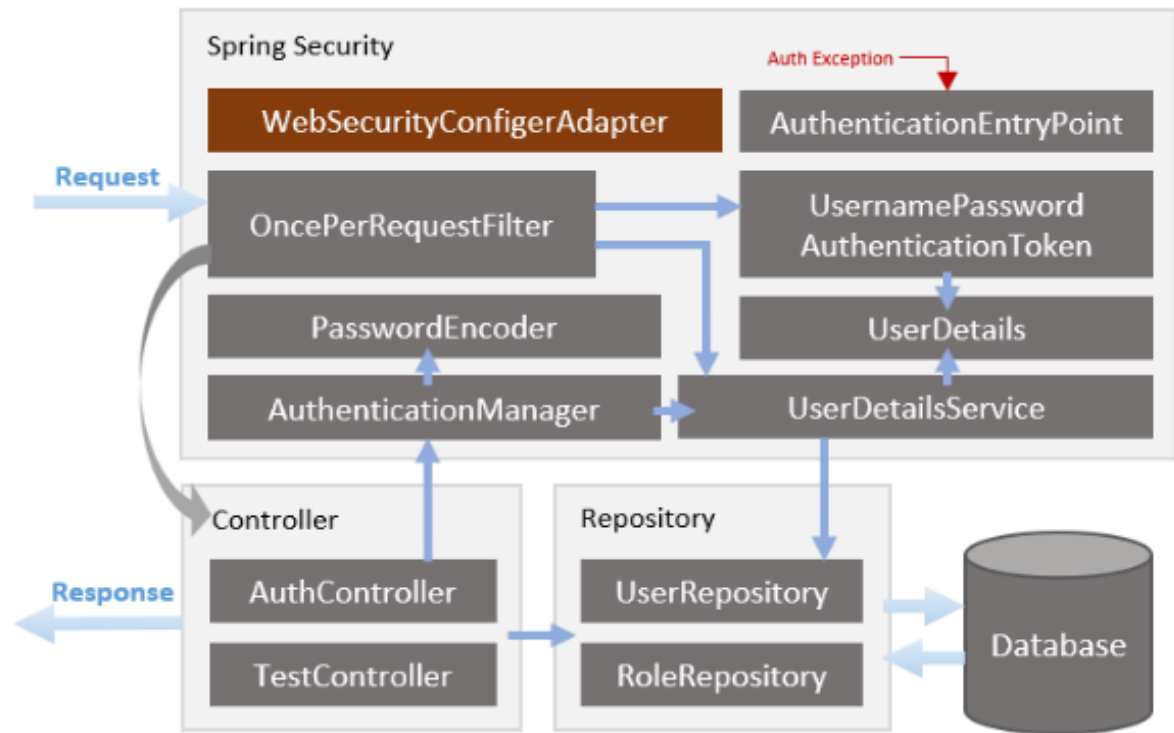
- Authentication is a process that verifies that someone or something is who they say they are. The authentication compares the username and password entered with a record it has on its database. If the information submitted matches, the system assumes you are a valid user and grants you access.
- Authorization which is the function that gives suitable permissions for the authenticated user.



The flow of the Authentication process:

- the user will make a request to the backend with his registered password and Gmail.
- the server will take this data given and validate whether it is in the database or not.
- After finding the data in the database the server generates a Jwt token to send it to the client again which is the user that requests the data (JWT token which is **JSON web Token** that is used to share data between the client and server in a secure way).
- This generated data will be sent back to the client to be able to get authorized to make changes and to have suitable permission in the web application.
- the client will use this token to authenticate his data and authorize his sign-in session, this will happen by sending the token again and the system will make sure either the token sent by the client is the one generated by the server that validates his data or not.

The previous explanation was an overview of the flow of the authentication process, we are going to explain more about authentication using **Spring Security** in spring boot.



How it works in the backend:

```
Authentication authentication = authenticationManager.authenticate(  
    new UsernamePasswordAuthenticationToken(loginRequest.getEmail(), loginRequest.getPassword())  
);
```

a) Generates UsernamePassword:

- First of all, we take from the front the data entered by the user which is the (Email and password) and generate a token to send to AuthenticationManager to send it to UserDetailsService which is a class that extracts the data of the user from the given token and finds whether it exists in the database or not.

b) AuthenticationManager: is the main strategy interface for authentication.

If the principal of the input authentication is valid and verified, AuthenticationManager returns an instance with the authenticated flag = true. It throws an AuthenticationException in case of invalidation of the token given. For the last case, it returns null if it can't decide. It delegates the authentication process to a list of AuthenticationProvider instances. We can set up a global or local AuthenticationManager if we create a SecurityFilterChain bean. For a local AuthenticationManager, we could create an AuthenticationManager bean, accessing AuthenticationManagerBuilder through HttpSecurity.

c) AuthService: after authentication of the user from the database. A Jwt token will be generated in order to determine the authority and give specific responsibility, expire at (session), etc.

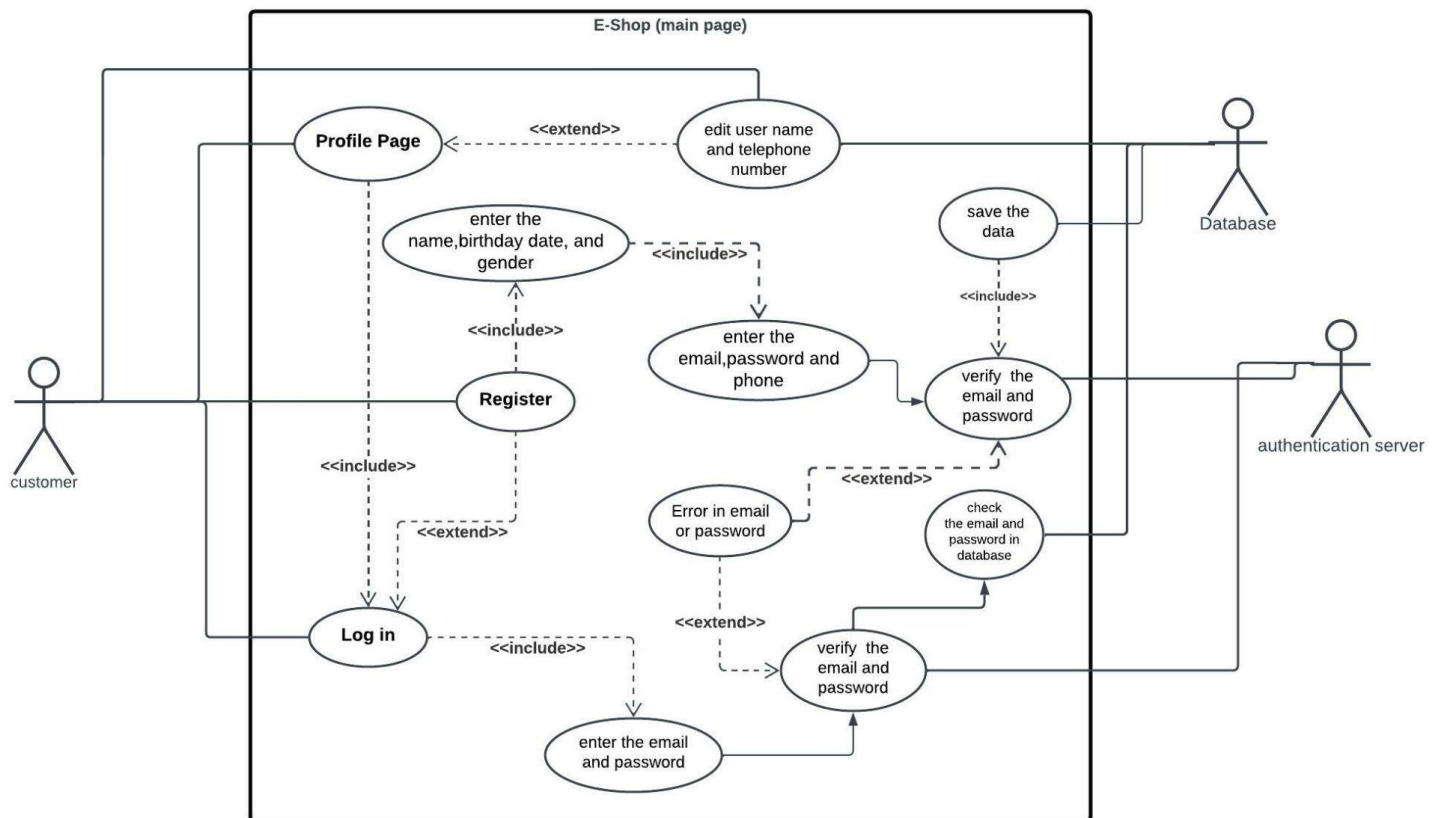
d) UserDetailsService: an interface is a core interface in Spring Security framework, which is used to retrieve the user's authentication and authorization information.

4. Password encoding: instead of storing the password of the user in plain text which can be easily hacked. There is a generator to take the provided password to encode it using BCryptPasswordEncoder which the spring security library supports in spring-boot.

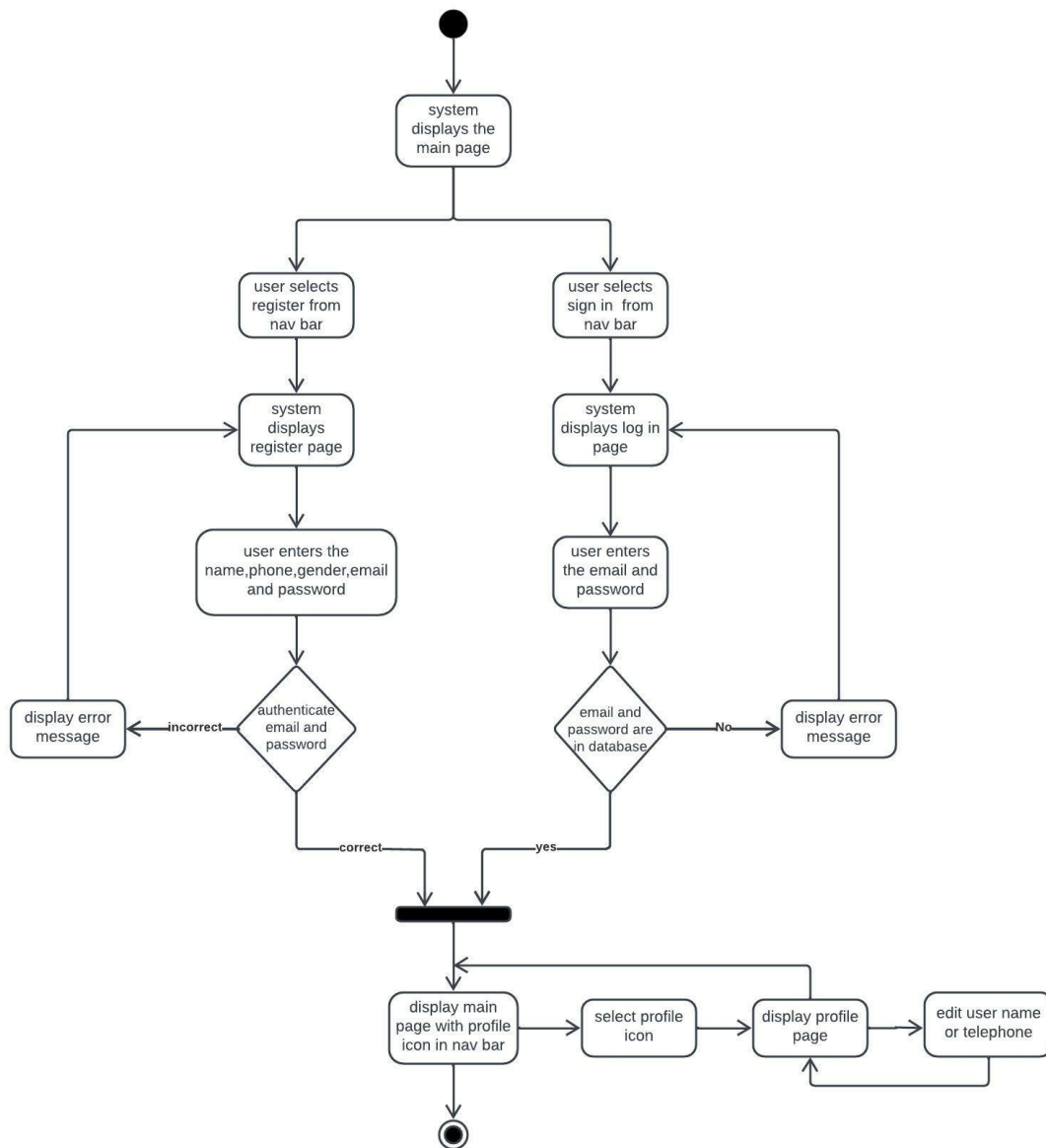
The security configuration is done in simple steps:

- injection of the implementation of the users' details service
- Definition of the authentication provider that references the details service.
- Finally, enable the password encoder.

Detailed Use case diagrams:

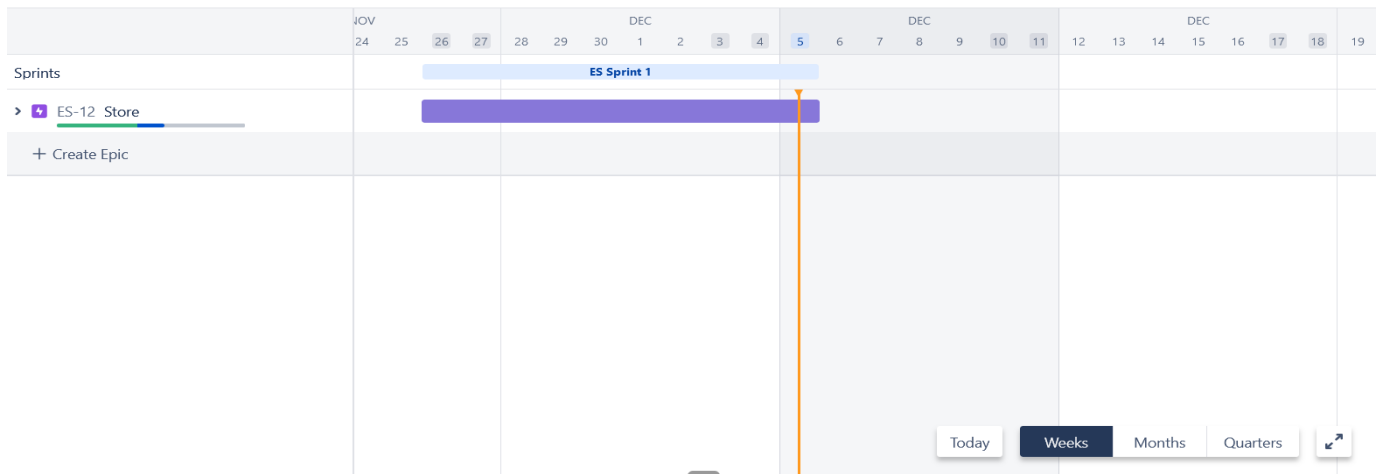


The flow of events for each of the use cases using activity diagrams:



Jira report:

- Current Jira Roadmap:



- Milestone sprint listing the included requirements Jira Stories:

▼ **ES Sprint 1** 26 Nov – 5 Dec (4 issues) 0 3 6 Complete sprint ...

Building the essential features of the application which are sign in, sign up and main page which will make the user able to surf the application.

ES-15 Sign Up Feature	STORE	2	DONE ✓	BY
ES-16 Sign in Feature	STORE	2	DONE ✓	
ES-14 Main Page	STORE	3	IN PROGRESS ▾	
ES-17 Profile Page	STORE	2	DONE ✓	TA

+ Create issue

▼ **Backlog** (4 issues) 0 0 0 Create sprint






ES-18 Cart Page	STORE	TO DO ▾	
ES-19 Product Page	STORE	TO DO ▾	
ES-20 Purchase Page	STORE	TO DO ▾	
ES-32 Sign Out Feature		TO DO ▾	

+ Create issue

- Stories, tasks, and subtasks estimates:

Projects /  E-Store /  ES-12 /  ES-15

Sign Up Feature










 Attach  Add a child issue  Link issue  

Description

The user needs to enter his meta-data such as the name, the email, the password for his account etc.






Child issues

Order by   

<div></div> 100% Done			
	ES-22	The Form on the Front end	1  DONE 
	ES-24	Append the data to the database in the back-end	1  DONE 
	ES-30	Link the Front-end with the Back-end	1  DONE 

Projects /  E-Store /  ES-12 /  ES-16

Sign in Feature










 Attach  Add a child issue  Link issue  

Description






The user needs to enter the email and password which will allow the back-end to authenticate the user.

Child issues

Order by   

<div></div> 100% Done			
	ES-24	The Front login page.	1  DONE 
	ES-23	Authenticating the user in the backend	1  DONE 
	ES-25	Link the front login page with the backend.	1  DONE 

Main Page

 Attach  Add a child issue  Link issue  

















Description

A complete page with a navigation bar, side bar includes the filters and the categories, and the products.






Child issues

Order by   

 50% Done

	ES-26	Navigation Bar	0.5		DONE 
	ES-27	Side Bar	0.5		DONE 
	ES-28	Product Box	0.5		IN PROGRESS 
	ES-29	Products Database	2		IN PROGRESS 
	ES-34	Routing Feature	1	TA 	DONE 
	ES-36	Linking the back-end with the front-end	1		IN PROGRESS 

Profile Page

 Attach  Add a child issue  Link issue  










Description

A page containing the full name, the email, the phone number, the date of birth and the gender of the user.

Child issues

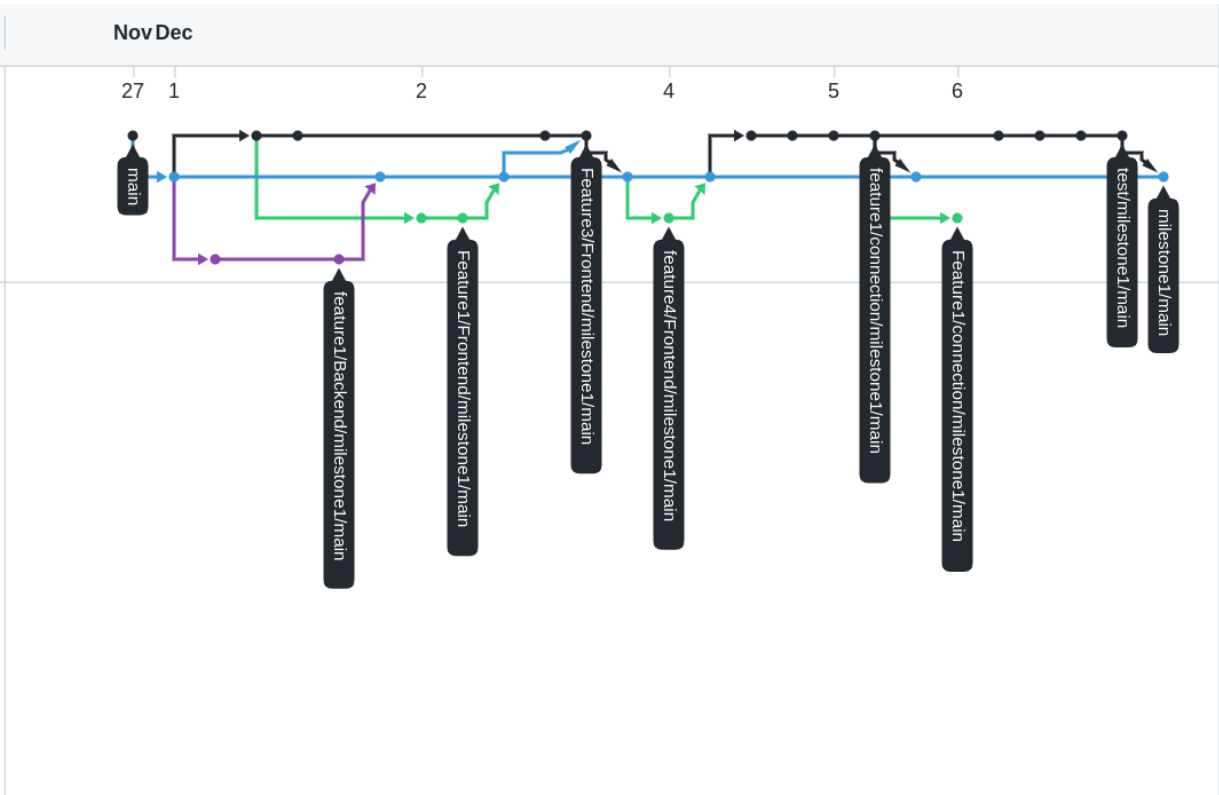
Order by   

 100% Done

	ES-33	UI in the Front end	1	TA 	DONE 
	ES-34	The database in the backend	1		DONE 
	ES-35	Linking the front-end with the backend	1		DONE 

Git & GitHub report showing:

- branches diagram showing the main branch, milestone branch, and feature branches:



- Pull Requests:

<input type="checkbox"/>	1 Open	<input checked="" type="checkbox"/> 10 Closed	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>			Test For MileStone 1 Services						
			#11 by farida52369 was merged 6 minutes ago						
<input type="checkbox"/>			Solving Issue of Register Routing and Empty Password						
			#10 opened 4 hours ago by TokaAshraf12						
<input type="checkbox"/>			Add Connection Between Client and Server	enhancement					2
			#9 by farida52369 was merged 8 hours ago • Approved						
<input type="checkbox"/>			Components and Routing between them	enhancement					1
			#8 by TokaAshraf12 was merged 2 days ago • Approved						
<input type="checkbox"/>			Feature2/frontend/milestone1/main						
			#7 by rowannasser was closed 3 days ago						
<input type="checkbox"/>			Adding Register Page						1
			#6 by MayarElkhouly was closed 2 days ago						
<input type="checkbox"/>			Adding Register Page						
			#5 by MayarElkhouly was closed 3 days ago						
<input type="checkbox"/>			Profile Page						
			#4 by TokaAshraf12 was merged 3 days ago						
<input type="checkbox"/>			Adding Login Page and making it be directed to the milestone1/main						3
			#3 by TokaAshraf12 was merged 4 days ago • Changes requested						
<input type="checkbox"/>			Adding Login Authentication Page						2
			#2 by TokaAshraf12 was closed 5 days ago						
<input type="checkbox"/>			Authentication and Authorization for Server Side	enhancement					4
			#1 by farida52369 was merged 5 days ago						

- Code Reviews comments:

1. Branch “[feature1/Backend/milestone1/main](#)” ← the branch for the Authorization and Authentication feature in the backend.

The screenshot displays a GitHub pull request interface. At the top, a navigation bar shows 'Conversation' (4), 'Commits' (2), 'Checks' (0), and 'Files changed' (22). A status bar on the right indicates '+796 -0' with a green progress bar.

A comment from user **farida52369**, posted 4 days ago, is highlighted. The comment contains a bulleted list of tasks:

- Adding `User`, `RefreshToken` models
- Adding Controller for
 - register
 - login
 - logout
 - refreshToken
- Adding `AuthenticationManager` in `AuthService`
- Adding `JwtProvider` for Authorization (security purpose)
- Adding `dto` package for lightweight objects transfer
- Adding `exception` package to handle exceptions
- Adding `PasswordEncoder` in `SecurityConfig` to encode user password before insert DB

Below the comment, a section titled 'Dependencies Used' lists the following:

- SQL/DB
 - spring-boot-starter-data-jpa
 - mysql
- Security
 - spring-boot-starter-security
- Setters/Getters
 - lombok
- Validation
 - spring-boot-starter-validation
- Jwt
 - spring-boot-starter-oauth2-resource-server
 - spring-boot-starter-thymeleaf

On the right side of the interface, a sidebar contains metadata for the pull request:

- Reviewers:** **Bassantyasser043**
- Assignees:** No one—assign yourself
- Labels:** **enhancement**
- Projects:** None yet
- Milestone:** No milestone
- Development:** Successfully merging this pull request may close these issues. None yet
- Notifications:** Unsubscribe (You're receiving notifications because you modified the open/close state.)
- 2 participants**
- Lock conversation**

At the bottom, a commit history shows a commit by **farida52369** titled 'authentication pages for backend' with commit hash **19e0287**, and a request for review from **Bassantyasser043** 4 days ago.

Fig. 1 'Commits of the branch'

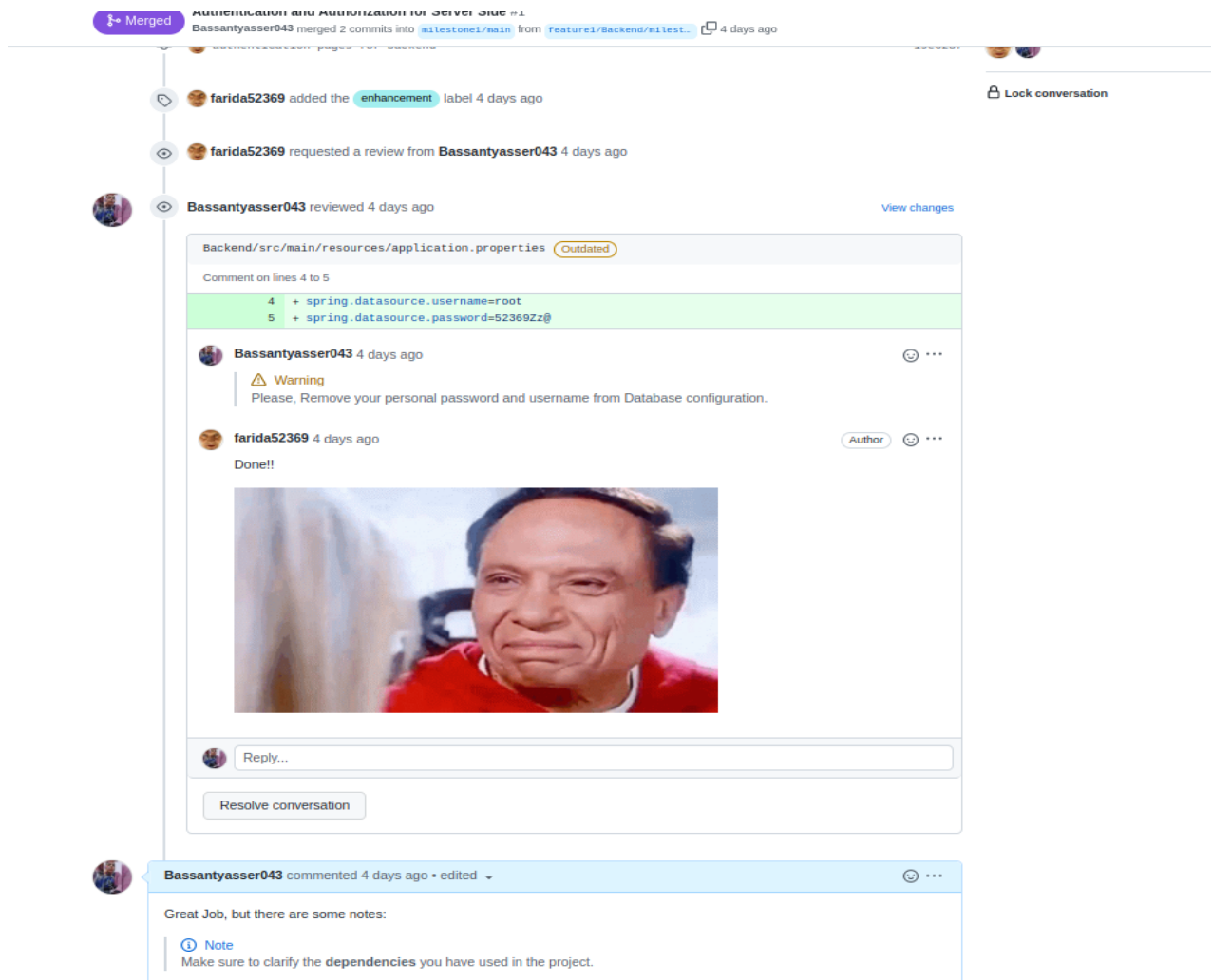


Fig. 2 'Code review for Authentication and Authorization'

2. Branch “[Feature1/Frontend/milestone1/main](#)” ← Branch for committing the sign-in component.

Adding Login Page and making it be directed to the milestone1/m

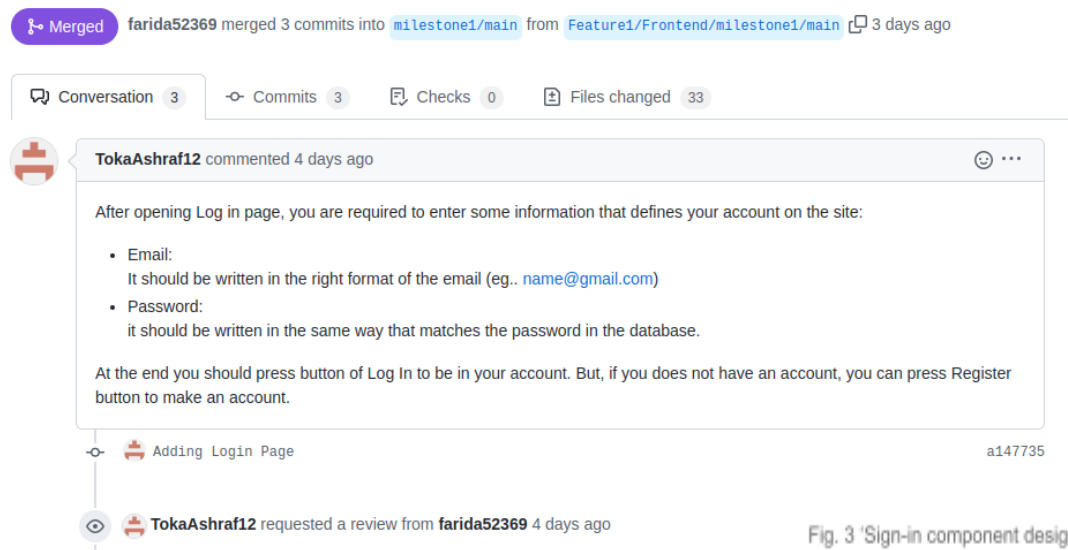


Fig. 3 'Sign-in component design'

Merged

Adding Login Page and making it be directed to the milestone1/main #3

farida52369 merged 3 commits into [milestone1/main](#) from [Feature1/Frontend/miles...](#) 3 days ago



farida52369 commented 4 days ago • edited



Great work, Toka.

The PR performs its functionality perfectly but there are some notes to be considered:

- logo position
- background coloring (size)
- centralized box of `Log In`

They aren't responsive in the design as they differ depending on screen size.

As shown in the screenshots for two different screen sizes:



Merged

Adding Login Page and making it be directed to the milestone1/main #3

farida52369 merged 3 commits into [milestone1/main](#) from [Feature1/Frontend/miles...](#) 3 days ago



farida52369 requested changes 3 days ago

[View changes](#)

Frontend/src/app/login/login.component.ts Outdated

Comment on lines 20 to 23

```
20 + if (this.password === '' || this.email === '') {
21 +   alert("Please Enter Your Email and Password");
22 +   return;
23 + }
```



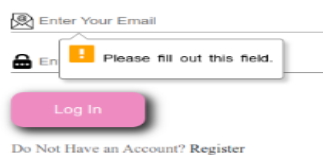
farida52369 3 days ago



`Alert` isn't a good choice to handle the emptiness of `Password` and `Email` fields as it stops the App from doing any action until clicking `OK` so it has a special cases where `Alert` is useful but ours isn't one of them.

You already handled the case of emptiness through this note:

Log In



Frontend/src/app/app-routing.module.ts
Outdated

```

4 +
5 +
6 + const routes: Routes = [
7 +   {path: 'Login', component: LoginComponent}

```

farida52369 3 days ago

Following the conventional naming for `ts`:

- `routing` has to be in lowercase letters
 - Login → login
 - MainPage → main-page
- `method` has to be in lowerCamelCase letters
 - Login_Req_and_Res → loginReqAndRes in login.component.ts
- Paragraphs shouldn't be in lowerCamelCase
 - Do Not Have an Account? → Don't have an account? in login.component.html

You can find more about conventional naming for `ts` [here](#)

Reply...

Resolve conversation

TokaAshraf12 added 2 commits 3 days ago

- Login page after modify the background and conventional naming
93090e7
- After correct some conflicts
c175cb2

farida52369 merged commit `7dcb3f4` into `milestone1/main` 3 days ago

Revert

3. Branch “[feature4/Frontend/milestone1/main](#)” ← Routing between different components in frontend.

Components and Routing between them #8

Merged
farida52369 merged 1 commit into `milestone1/main` from `feature4/Frontend/milestone1/main` 2 days ago

Conversation 1
Commits 1
Checks 0
Files changed 18

TokaAshraf12 commented 2 days ago • edited by farida52369

- add all components together and make routing between pages:
 - clicking on link of `register` in `login` page makes routing to the `register` page .
 - clicking on `log in` in home page makes routing to `login` page and `sign up` makes routing to `register` page .
 - when logging in successfully with validation from database that this account information is stored after making register, routing will be to `home` page and links to `log in` and `sign up` will disappear and `profile` link will appear instead.

add all components together and some routing between pages
ae15f78

farida52369 added the `enhancement` label 2 days ago

farida52369 self-requested a review 2 days ago

farida52369 approved these changes 2 days ago

View changes

farida52369 left a comment

Looks like the enhancements are fine So i'll merge them.
Great Work @TokaAshraf12 @rowannasser @MayarElkhouly 🙌🙌

4. Branch “[feature1/connection/milestone1/main](#)” ← Connection between frontend and backend.

Add Connection Between Client and Server #9

Merged Bassantyasser043 merged 4 commits into [milestone1/main](#) from [feature1/connection/milestone1/main](#) 5 hours ago

Conversation 2 Commits 4 Checks 0 Files changed 39



farida52369 commented 15 hours ago



- Adding HTTP connection between Client and Server using `HttpClient` API in:
 - Login Component
 - Register Component
- Change PK in `user` model → email to have a unique key for each user
- Routing between Different Components `Login`, `Register`, `Home`

Warning

Register routing doesn't function properly

farida52369 added 4 commits yesterday

- adding environment prod and some services 65c4575
- remove personal info f7bac46
- add connection between components 4b43bd9
- remove personal info ffd7bb8

farida52369 added the [enhancement](#) label 15 hours ago

farida52369 requested a review from Bassantyasser043 15 hours ago



Bassantyasser043 approved these changes 5 hours ago

[View changes](#)

Bassantyasser043 left a comment



- The functionality is good enough.
- The connection between login form in the `front` and `backend` is working well.
- Authentication method are working well and return to the front whether it is registered or not (`routing to login in case of registred before otherwise return in the same page`) . well done 🌟🌟🌟🌟🌟💙