**Research and Development Team** Technical Test **Documentation**

**LL20-2 –** Lionel Ritchie

* **Application**: Nationality Test App
* **Entity Relationship Diagram**

Diagram

Description automatically generated

Figure . ERD

* **Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Email** | **Password** | **Role** |
| 1 | [admin@mail.com](mailto:admin@mail.com) | Admin123! | Admin |
| 2 | [brandon@mail.com](mailto:brandon@mail.com) | Bran123! | Participant |
| 3 | [stanley@mail.com](mailto:stanley@mail.com) | Stan123! | Participant |
| 4 | [clarissa@mail.com](mailto:clarissa@mail.com) | Clar123! | Participant |
| 5 | [gaby@mail.com](mailto:gaby@mail.com) | Gaby123! | Participant |
| 6 | [cleo@mail.com](mailto:cleo@mail.com) | Cleo123! | Participant |
| 7 | [vincent@mail.com](mailto:vincent@mail.com) | Vincent123! | Participant |
| 8 | [jepe@mail.com](mailto:jepe@mail.com) | Jepe123! | Participant |
| 9 | [onel@mail.com](mailto:onel@mail.com) | Onel123! | Participant |

* **Features**

1. **Sidebar**

* If the logged in user is **guest**, then the application will show **nothing**
* If the logged in user is **admin**, then the application will show a link to **Home**, **Manage Questions**, **Manage Schedules**, and **Manage Participant**
* If the logged in user is **participant**, then the application will show a link to **Home**, and **Take Test**.
* If the link is **clicked**, then user will be **redirected** to the associated page.

Graphical user interface, application

Description automatically generated

Figure . Navigation bar for guest (left), admin (middle), participant (right)

1. **Home page**

* If the logged in user is **guest**, then the application will **show a message for guest.**

Background pattern

Description automatically generated

Figure . Home page for guest

* If the logged in user is **admin**, then the application will **show a message for admin.**

Graphical user interface, application, Teams

Description automatically generated

Figure . Home page for admin

* If the logged in user is **participant**, then the application will **show message for participant** and his/her **schedule status**.
* The application will show information about the schedule such as **start date**, **end date**, **taken on**, and **status**.

Graphical user interface, application

Description automatically generated

Figure . Home page for participant (already taken the test)

Graphical user interface, application, background pattern, Teams

Description automatically generated

Figure . Schedule summary for participant (Test not yet started)

Background pattern

Description automatically generated

Figure .Schedule summary for participant (Test is on going)

Background pattern

Description automatically generated

Figure . Schedule summary for participant (missed the test)

1. **Manage question page**

* This page is only accessible by **admin**
* The page will show **all questions** in grid list, **question preview**, and a button to **add new question**.
* Each question will show the question information such as **type of the question**, **question text**, and a **button to show its preview**.
* If the admin **click add new question** button, then admin will be **redirected to add question page.**

A screenshot of a computer

Description automatically generated

Figure . Manage question page (with no question to be previewed)

* If the admin click the show preview button, then the **preview of the question** will be shown along with a button to **update** or **delete** the question.

Graphical user interface, application

Description automatically generated

Figure . Question preview (Multiple Choice)

A picture containing graphical user interface

Description automatically generated

Figure . Question preview (Boolean)

Background pattern

Description automatically generated

Figure . Question preview (Select from list)

Background pattern

Description automatically generated

Figure . Question preview (Essay)

Graphical user interface, text, application

Description automatically generated

Figure . Question preview (File upload)

* If the admin **click update** button, then the admin will be **redirected to update question page**.
* If the admin **click delete** button, the application will show a prompt. If admin **accepted the prompt**, then **the selected question will be deleted.**

**Optimization**: after the selected question is deleted, the app will not fetch the updated data from database, instead it will only remove the deleted question from the list and update the UI with *StateHasChanged* method.

1. **Add question page**

* This page is only accessible by **admin**.
* This page will show a form to **input question** text and **question type**.

Background pattern

Description automatically generated

Figure . Add question form (no type selected)

* If the selected type is **multiple choice** or **select from list**, then an **input field to add possible answers** will be shown.

**Easy to use**: This form ease the admin when adding new question because it has different state for different question type. I am using Blazor’s EditForm to make me easier to validate the dynamic possible answers input.

* Admin could **add** or **delete** the possible answer input, **but there will be minimal of 1 possible answers input remain.**

Background pattern

Description automatically generated

Figure . Add question form (select from list / multiple choice)

Background pattern

Description automatically generated

Figure . Add question form (boolean / essay / file upload)

* **Error messages will be shown** if admin didn’t fill all the available input field

Background pattern

Description automatically generated

Figure . Validation for add question form

* If all input field is filled, the **next section of the form will be shown**.
* if the **selected type is multiple choice, select from list, or boolean**, then admin must **choose the correct answer from the possible answers list.**

Background pattern

Description automatically generated

Figure . Choose correct answer (boolean / multiple list / select from list)

* If **no correct answers is selected**, then the application will **show an error alert.**
* If the **selected type is essay or file upload**, then **a message to inform that will be shown.**

**Ease of use:** the correct answer to be chosen list is binded to the list of possible answers input in the first section form using two way data-binding. If the user added new possible answer in the first section, then it will be also reflected when choosing the correct answer.

Background pattern

Description automatically generated

Figure . Choose correct answer (essay / file upload)

* If the admin **click add question**, then the **question will be added to the database** and admin will be **redirected to manage question page.**

1. **Update question page**

* This page is only accessible by **admin**.
* This page will **show all fields as in the add question page**. The **input field will be filled by the question data.**

Background pattern

Description automatically generated

Figure . Update question (multiple choice / select from list)

Background pattern

Description automatically generated

Figure . Update question (essay / boolean / file upload)

* Admin could also **change the type of the question to any type he/she want.**
* The **validation** will also **behave like in add question page** (validate empty).
* If the admin **click update**, then the **question and possible answers (if exists) will be updated to the database.**

1. **Manage schedule page**

* This page is only accessible by **admin**.
* This page will **show all user schedules in list**, and **add new schedule button.**
* Each **schedule** will show information such as **the participant user name, start date, end date**, **status, update button,** and **delete button.**
* The **update** and **delete** **button** will only show **if the schedule is not started yet** or **the schedule is on going but the participant has not taken the test yet.**
* If **update button is clicked**, than **admin will be redirected to update page.**
* If **delete button is clicked**, then a prompt will be shown. If the prompt is accepted, then the **schedule will be deleted from the database.**

**Optimization**: same as delete question, after delete is done, the deleted participant is removed from the list in the UI instead fetching the updated data from database.

A screenshot of a computer

Description automatically generated

Figure . Manage schedule

Graphical user interface, application

Description automatically generated

Figure . Manage schedule (passed deadline)

* There are 5 possible statuses for each schedule, which are:

|  |  |  |
| --- | --- | --- |
| No | Status | Description |
| 1 | Passed deadline and has been taken | Test has already taken and the end date is greater than now. |
| 2 | Not yet started | Start date is less than now. |
| 3 | On going and not yet taken | Test is not taken and start date <= now <= end date |
| 4 | Passed deadline but not yet taken | Test is not taken but the end date is greater than now |
| 5 | On going and has been taken | Test is taken and start date <= now <= end date |

1. **Add schedule page**

* This page is only accessible by **admin**
* This page will **show a form for the admin to add a new schedule**. The form will consists of **select input to choose participant**, **start date**, and **end date**.
* **Participants** that can be **selected are only participants** that **do not have schedule** associated with him/her.
* The form **will validate that a participant is chosen** and **the start date time must be less than end date time.**
* If admin click **add schedule button**, then a **new schedule will be added to the database** and admin will be **redirected to manage schedule page.**

Background pattern

Description automatically generated

Figure . Add Schedule

1. **Update schedule page**

* This page is only accessible by **admin**
* This page will **show a form to update the selected schedule**. The form is exactly the same as in the add schedule page, except the **select input to select participant is disabled.**
* The **participant selected, start date**, and **end date input** will be **filled with the selected schedule.**
* The validation will also be the same as in the add schedule page.
* If admin click **update schedule button**, then **the schedule will be updated to the database** and admin will be **redirected to the manage question page.**

Background pattern

Description automatically generated

Figure . Update schedule

1. **Manage participant**

* This page is only accessible by **admin**
* This page will show **all participants** in list and **a delete button for each participant.**
* If the **delete button is clicked**, then a promp will be shown. If admin accepted the prompt, then the **selected participant will be deleted** from database.

**Optimization**: same as delete question, after delete is done, the deleted participant is removed from the list in the UI instead fetching the updated data from database.

Background pattern

Description automatically generated

Figure . Manage participant

* For each participant card, there will be 3 possible state:

|  |  |  |
| --- | --- | --- |
| No | State | UI |
| 1 | Participant do not have schedule associated with him/her | Show ‘Not scheduled yet’ and set schedule button. If this button is clicked, the admin will be redirected to add schedule page. |
| 2 | Participant missed the test or his/her test is not started | Show ‘Test not yet started or has already passed deadline’ message. |
| 3 | Participant has already taken the test. | Show view status button and view result button. |

1. **View participant result**

* This page is only accessible by **admin**
* This page will **show the result of the participant** such as **test result**, **correct answer**, **wrong answer**, **not verified answer** (for essay or file upload), and **final score**.
* There will be **3 possible result**, which are:
* **Passed**: final score >= 60
* **Failed**: final score < 60
* **Not verified**: there are some answers that had not been verified by admin.

A screenshot of a computer

Description automatically generated

Figure . Answer status (passed)

Background pattern

Description automatically generated with medium confidence

Figure . Answer status (not verified)

Rectangle

Description automatically generated with low confidence

Figure . Answer status (failed)

1. **View participant answer status**

* This page is only accessible by **admin**.
* This page will **show the participant answer in each question.**
* Each participant answer will show information such as the **question**, **answer**, and **status.**
* There will be **3 possible status** : **correct**, **wrong**, **need manual verification** (essay or file upload).

A screenshot of a computer

Description automatically generated

Figure . View answer status

* If the file **status** is **need manual verification**, then **mark correct** and **mark incorrect button** will be shown.
* If the **mark correct button is clicked**, then the answer will be **marked as correct**. Else if the **mark incorrect button is clicked**, then the answer will be **marked as incorrect**.
* If the **answer type is file upload**, then a **download button will be shown**. If **clicked**, the answer file will be **downloaded** from the server.

**Ease of use**: this make verifying participant answer easy to use because admin could verify user answer directly below their answer.

1. **Test page**

* This page is only accessible by **participant that his/her schedule is on going** (start date <= now <= end date).
* **Server time** will be shown in this page with interval every one second.
* This page will **show all questions** from database to be answered by the participant.
* Each **question** will show information such as **question type**, **number**, **question text**, and **input field according to the question type.**

A picture containing text, screenshot, monitor, screen

Description automatically generated

Figure . Test page

Graphical user interface

Description automatically generated

Figure . Test page (continue)

* Question with type **select from list**, **multiple choice**, or **boolean** will be **auto-corrected by the system.**
* Question with type **essay** or **file upload** will need to **manually corrrected by admin** in answer status page.

**Efficiency**: I use the same component used by question preview in manage question page to show answer input field according to the question type. This make me didn’t have to duplicate the code.

* If user **uploaded a file** in file upload, the **file will be saved to the database** and **information about the file will be displayed.**

Graphical user interface, application

Description automatically generated

Figure . Uploaded file upload input

* If the participant **click submit answer**, then the answer will be **added to the database** and participant will be **redirected to home page.**

1. **Login page**

* **Admin** and **participant** will login from this page.
* I am using **ASP. Net Identity** to **ease the auth process** and it also **make role-based authentication much easier and efficient.**

Graphical user interface

Description automatically generated

Figure . Login page

1. **Register page**

* I am also using **ASP. Net Identity** for register
* New user that **register** to this page will **automatically assigned a role as participant.**

Background pattern

Description automatically generated

Figure . Register page

* **Other Efficiency**
* Almost all DB operations are using async / await to ensure that fetching data does not block the UI thread.

Text

Description automatically generated

* Using @key directive when showing component in looping, this will let blazor only update the needed element when change occurs to the list . Ex: when I delete one of the schedule, blazor will only remove the element from the container, not affecting the whole list.

Graphical user interface, text, website

Description automatically generated

* Using blazor dependency injection for operations that need to be used in many components (example: DB operations) to make it easier for component to get data from DB operations.



A screenshot of a computer

Description automatically generated with medium confidence