

Name: Kristoffer Snopestad Søderkvist

Student Number : 191284

Candidate number: 618

Title: Random Fact + Documents page

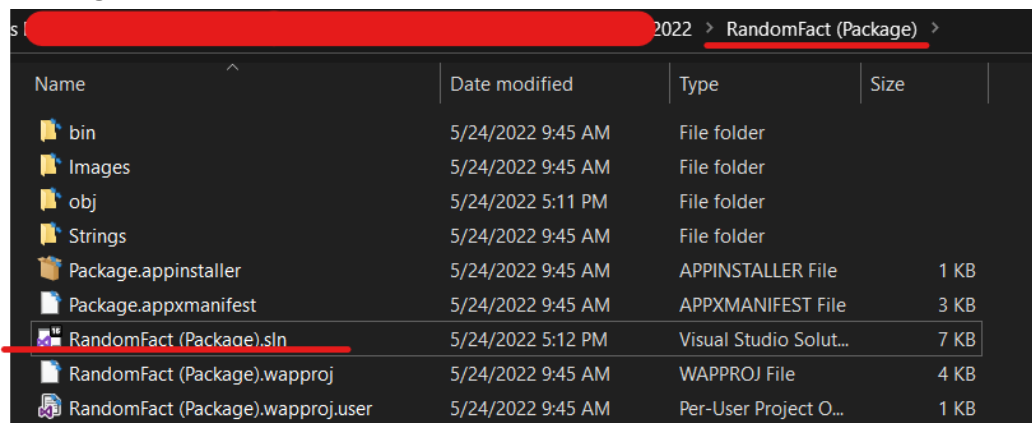
Any requirements

- Internet for online database.
- Connection to HIOF network. (For a partly “local” database) - If Donau connection is working. (Create, Update and Delete is working)
- And the obvious ones
 - Visual Studio 2019
 - .Net5
 - Possible to run WinUi 3

How to start the application

Open Solution

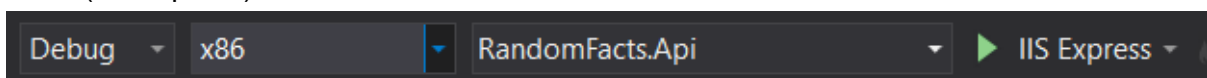
1 . Click on the RandomFact (Package).sln - solution file in the folder **RandomFact (Package)**.



Don't know if VS2019 has a way to run two applications simultaneously. If so skip step 2.

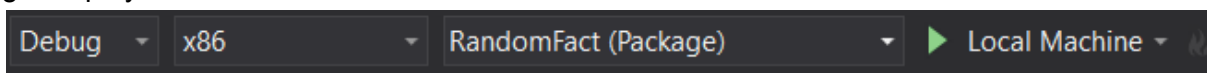
2. Launch it again, so you have the project open in two VS2019. One will be for the server side, and the other will be for the client.

3.Next make sure it is using RandomFact.Api and launch it by clicking on the green play button(IIS Express).



This will launch the server so it will be ready for the client/application.

4. Next in the other VS2019. Make sure it is using RandomFact (Package) and click on green play Local Machine to make it run.



Now it should be ready to go.

Recommended

When launching the application I recommend clicking on the New Document button and adding data, since the database is not connected with get/Read to donau, so it will not show any documents before adding at least a document. And close the app and start it again. I wrote more about that problem later in the text.

Information about parts not fully completed yet.

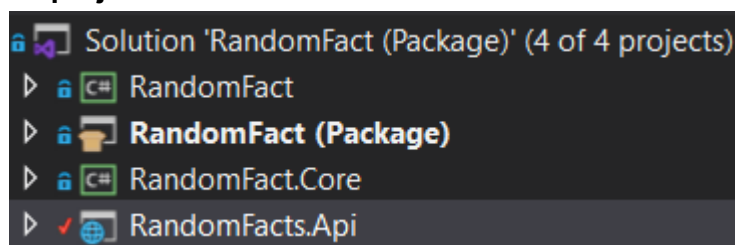
- Real random fact, did not have any time to implement it.
 - Spent too much time on connecting Asp.net Rest Api to SQL Donau database.
 - The database should contain titles on wiki pages, and use them to get a full link. That link would be used to get all data from that wikipedia page as Json format. With that it would convert Json over to data that it would show on that page.
 - The plan was if the fact database had a lot of Titles, it would choose one at random and show that page in the app.
- Get data from Donau.
 - Have made the Sql connection to Donau for Get and ASP.net core, but did not find a way to connect those two together.

Known Issues

- When adding/deleting/Updating or any data changing methods, it will not refresh the data shown in the app. Must restart the app to see the changes happen.
 - When I looked into it, it looked like x:Bind data will not update when the data changes.
 - Also tried to figure out a way to refresh the page, but everything with refresh was for UWP, and those I tried didn't work in WinUi 3.
- RandomFact page is not always showing all data.
 - And it is not Random, since the issue above.
- **NOTE ON LAST PAGE**


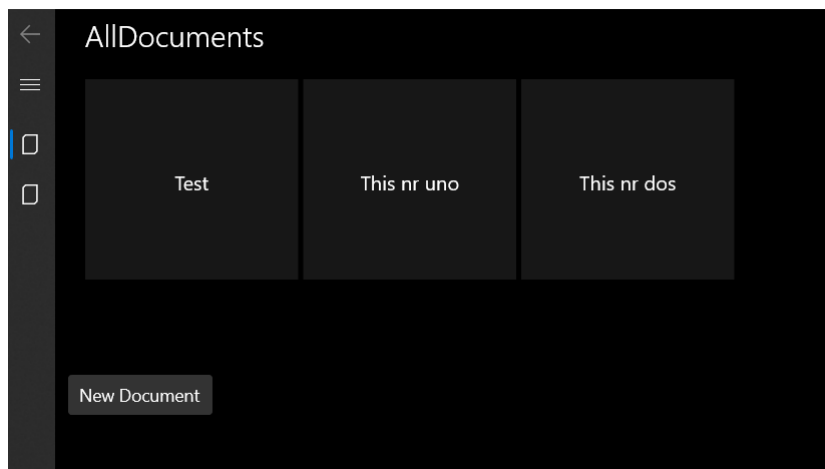
Description of the application.

All projects in Solution




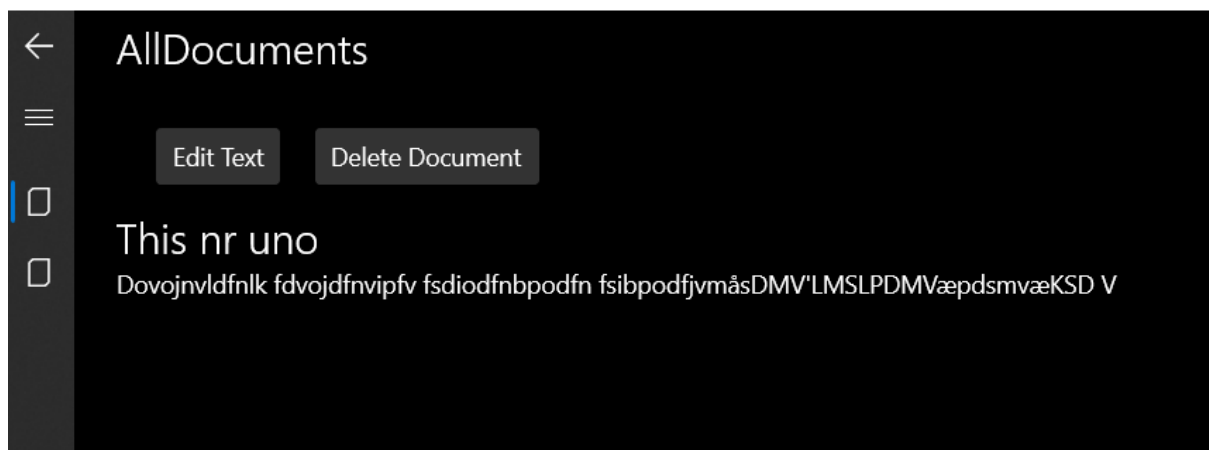
StartPage => All Documents

Choose to use this as a frontpage to show off all CRUD. Here it is possible to see all documents stored in the database, and it is also possible to create a new document. In this page it is possible to click on each document for getting more detail on each document.

 RandomFact

When clicking on a document

When clicked on a document it will show buttons for edit and delete the document, it will also show data it has received and put it on the screen. When clicking on "Edit document" it will set all data's visible status to collapsed, and all other that was hidden to visible. When the edit field is visible it will also bring two buttons, one for saving the changes made, and one to return to how it was before, changing all visible status to back as it was.

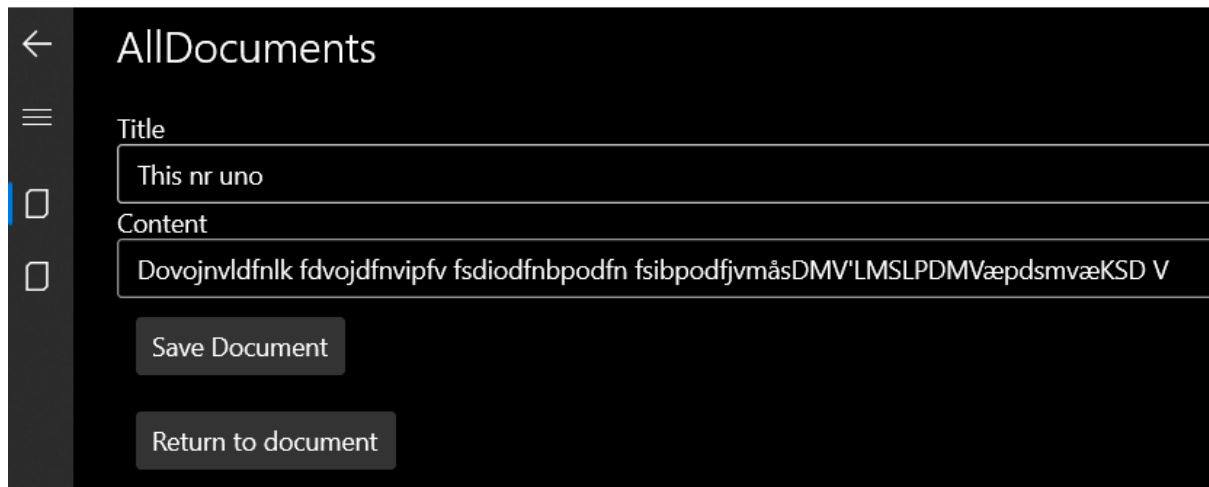
 RandomFact

Clicking on Edit Text

This will set all data from above to not visible (Collapsed), and show instead where you want to change data(Update/Put). It will take the data and set it already in the textbox field, so it will be easier to change it. When work is done you can either click on button "Return to document" if you want to discard all changed, or you can click on button "Save Document"

for it to send the data changes (with Put to db and so Sql Update command to Donau) to the Database. To see the changed content, you must restart the WinUi app.

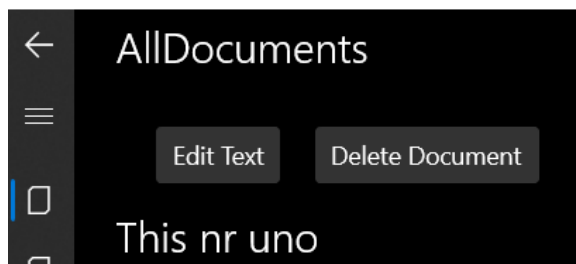
RandomFact



When clicking on the Delete Document button.

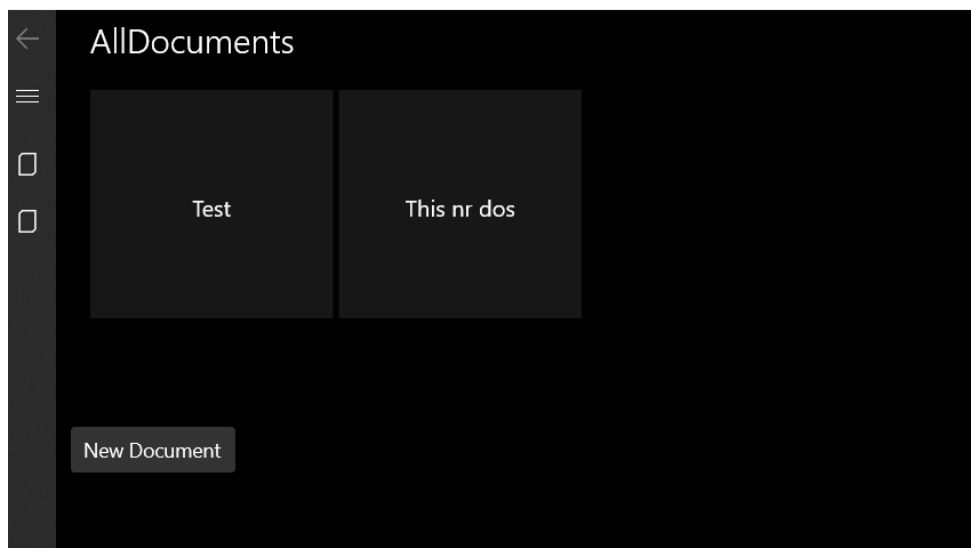
When clicking on the “Delete Document” button it will delete the document from the database. Please restart the app.

RandomFact



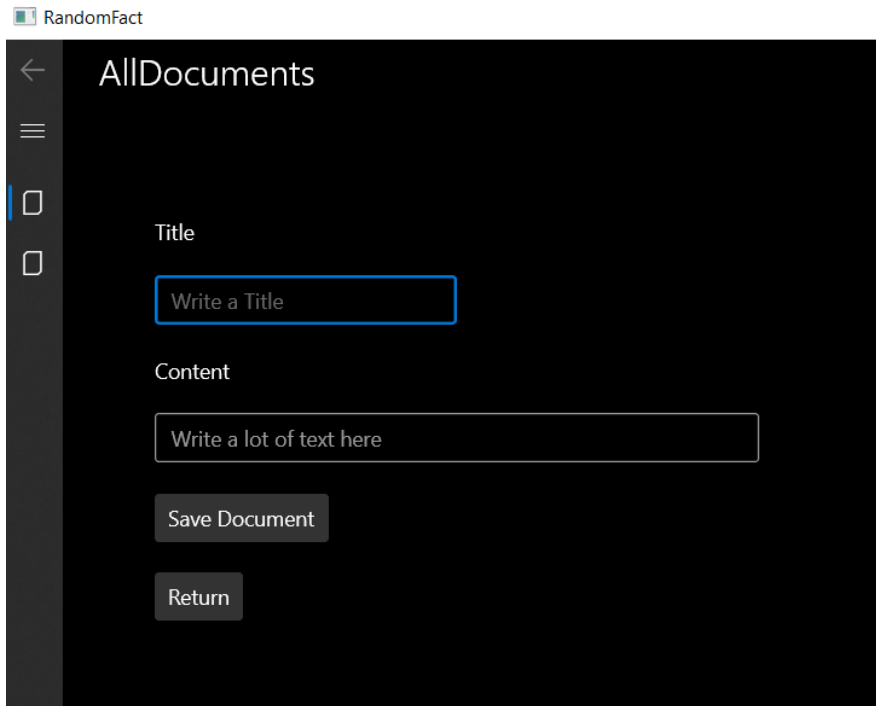
After a quick restart of the application, now it shows one less document because of the deleting.

RandomFact



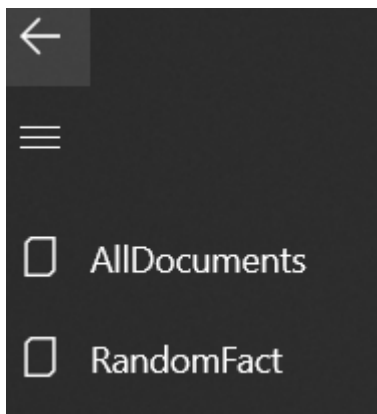
Clicking on New Document button

When clicking on the “New Document” button it will remove all existing documents, and show two input fields instead. Clicking on “Return” will show all documents again and remove the input fields, can also click on the top left arrow for return. Next button “Save Document” will create a document and add it to the database. It is using Rest api for sending data over to the server and turning the data to sql command and then over to Donau server. But still, to see changes in the app it must restart.



Icons on the left.

Possible to change between pages on the left side in the application.



Note. (Just to repeat information mention in the text)

The application has two major issues, and some not implemented features that I wished I had planned enough time to do.

First major issue is that if the data changes, the application needs to be restarted for the possibility to see those changes.

Second major issue is that it won't get the data from my donau database.

CRUD - RestApi with Asp.net core

Create => Using it to create a document and sending it over with POST so convert it over to a SQL command (INSERT INTO) and then over to donau.

Read => Tried to implement a method that takes a sql select command and adding data to a list, and then add it to the REST database. This is not the way to do it so it did not work.

Update => Using it to edit a document. Taking its id and what changes you want to make, and sending it over with PUT so convert it over to a SQL command (UPDATE) and then over to donau.

Delete => Delete a document. Taking Id from that document and sending a DELETE request over Rest, and then a DELETE with WHERE Id to the SQL donau database.