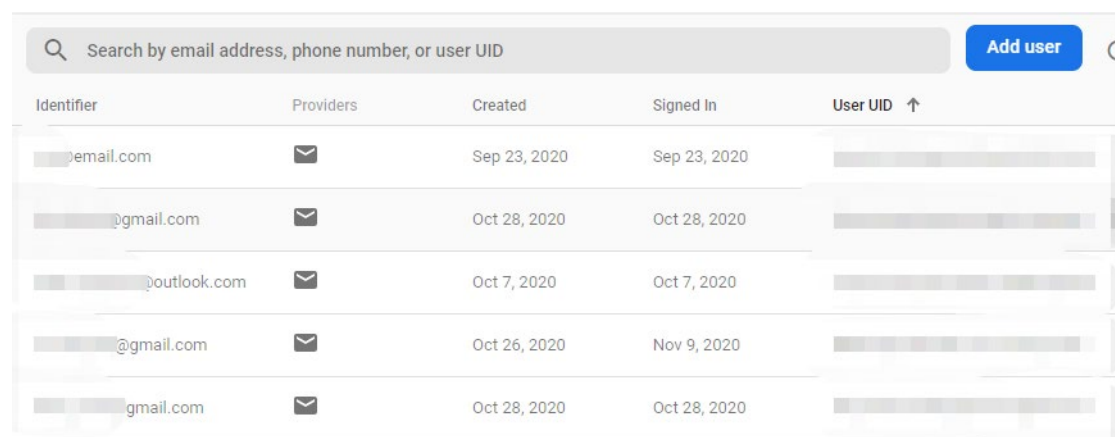


Description of database structure

This project use firebase as the database including:

1. Authentication
2. Realtime database
3. Storage

For the Authentication part, it contains all email user registered and their UID as shown below



The screenshot shows the Firebase Authentication console. At the top, there is a search bar with the placeholder text "Search by email address, phone number, or user UID" and a blue "Add user" button. Below the search bar is a table with the following columns: Identifier, Providers, Created, Signed In, and User UID (with an upward arrow icon). The table contains five rows of user data, with some fields redacted with grey boxes.

Identifier	Providers	Created	Signed In	User UID ↑
[redacted]@email.com	📧	Sep 23, 2020	Sep 23, 2020	[redacted]
[redacted]@gmail.com	📧	Oct 28, 2020	Oct 28, 2020	[redacted]
[redacted]@outlook.com	📧	Oct 7, 2020	Oct 7, 2020	[redacted]
[redacted]@gmail.com	📧	Oct 26, 2020	Nov 9, 2020	[redacted]
[redacted]@gmail.com	📧	Oct 28, 2020	Oct 28, 2020	[redacted]

For Realtime database sector, it stores all users' information which contains







1. About: Description of the user (User's overall introduction)
2. Intro: User's self-writhed Introduction (Including specific education backgroups··)
3. First_Name: User's first name
4. Last_Name: User's last name
5. City: City where user live in (for search purpose)
6. Country: Country where user is from (for search purpose)
7. University: University where user is from (for search purpose)
8. Faculty: User's Faculty (for search purpose)
9. Occupation: User's job
10. Id: User's UID
11. Other_id: ePortfolio the user wants to check
12. Other_ids: a list of all ePortfolio the user wants to check (searched result)



```


xDTToq8wzIjbHkZbbDUJFyUJwodH2
  About: "Shengze from Melbourne"
  City: "melbourne"
  Country: "china"
  Faculty: "commerce"
  First_Name: "Shengze"
  Intro: "Shengze from Unimelb"
  Last_Name: "zhu"
  Occupation: "student"
  University: "unimelb"
  id: "xDTToq8wzIjbHkZbbDUJFyUJwodH2"
  other_id: "mxLvUm1JE3dh4v1fgEnSy4Lj1FI2"
  other_ids
    0: "mxLvUm1JE3dh4v1fgEnSy4Lj1FI2"
    1: "xDTToq8wzIjbHkZbbDUJFyUJwodH2"

```

Storage has all user's UID as individual files each user's media (including image, pdf video) will be stored in their individual folders

<input type="checkbox"/>	Name
<input type="checkbox"/>	 7pOWDTel14gFlKsaWdrPp0iBGzI2/
<input type="checkbox"/>	 AupW2YKkpndveMdVvkFRF0QjxlK2/
<input type="checkbox"/>	 BMlrxBjQRXadFCX1GJfI4SyfJD3/
<input type="checkbox"/>	 CbfoCFQiNEb3BZfmW3Xlwpqu9CG3/
<input type="checkbox"/>	 DTIpKCZxArdlZppAt7FecgbAuWw1/
<input type="checkbox"/>	 EusGqj5fdog2kS94aYsk7GhwPro1/

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	 profile.jpg	27.98 KB	image/jpeg	Oct 21, 2020
<input type="checkbox"/>	 research.pdf	39.1 KB	application/pdf	Oct 21, 2020



Name
[profile.jpg](#)

Size
 28,654 bytes

Type
 image/jpeg

Created
 Oct 21, 2020, 4:58:12 PM

Updated
 Oct 21, 2020, 4:58:12 PM

File location ▼

Other metadata ▼

Description of key algorithms

script.js

used to log in/ out and show display log in page

1. signUp()
Take userEmail and userPass obtained from sing up box and construct a new account using that email if the email already exist, report error.
2. login()
Take userEmail and userPass obtained from log in box and log in that account using that email and password if fail to authentication \, report error.
3. resetPass()
Take userEmail obtained from log in box and send reset password email to that email address if the email is invalid, report error otherwise.
4. logOut()
Log out current user
5. verifyEmail()
Called while construct a new account send email to that email address if the email is invalid, report error otherwise.
6. update()
Update the user's information obtained from setting page. Report error if the form is incomplete.
7. containerShow use to switch log in and sign up box

view.js

used to view other user's ePortfolio

upload.js

used to upload files and text to construct one's ePortfolio in editing page

1. UploadButtonPressed()
Listen to a file Upload the image file to firebase Cloud 'profile.jpg'
2. UploadpdfButtonPressed()
Listen to a file Upload the image file to firebase Cloud ' research.pdf'
3. SubmitButtonPressed()
Using the text obtained from textbox and stored it in firebase realtime database Intro.
4. AboutButtonPressed()
Using the text obtained from textbox and stored it in firebase realtime database About.

search.js

used in search page to search for all users satisfied the requirement

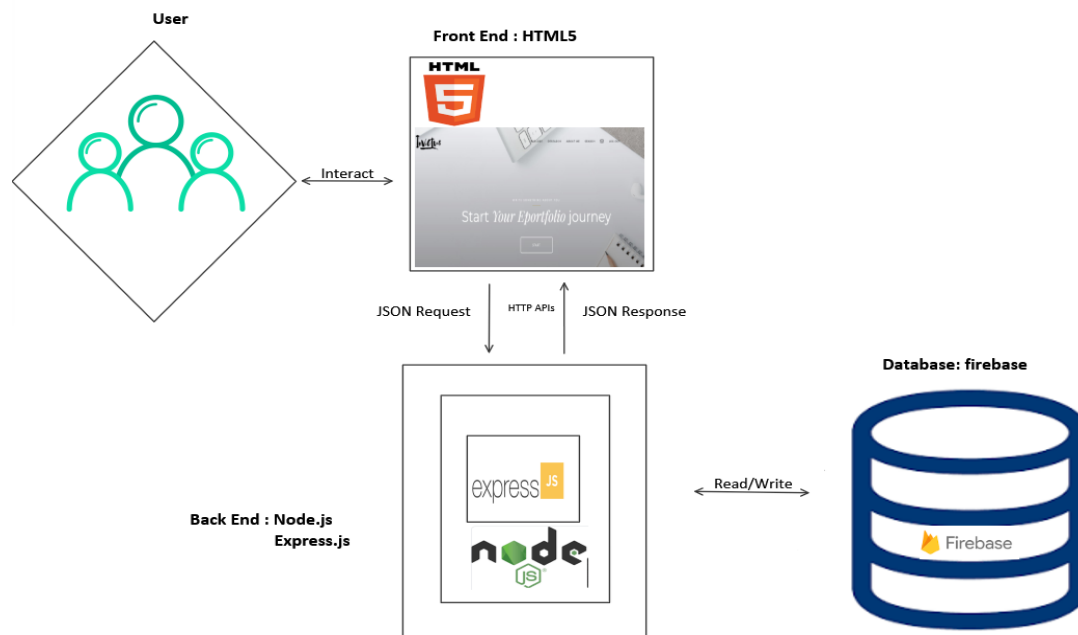
1. SearchButtonPressed()
Use the filter and key word obtained from search page to filter out all matched result.
2. show()
3. display searched result.

Deployment guidelines

Follow this link to deploy the project

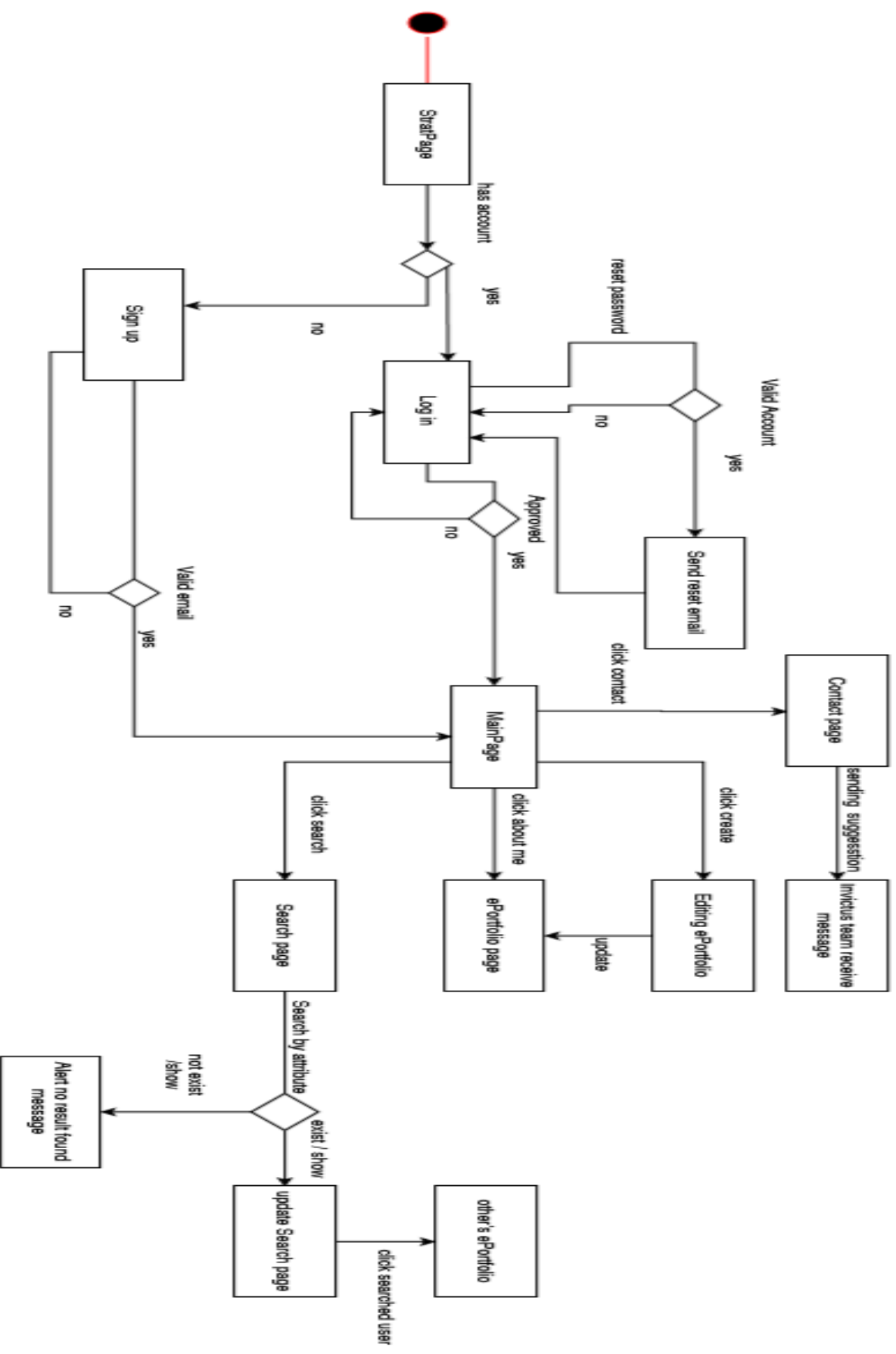
<https://medium.com/@aleemuddin13/how-to-host-static-website-on-firebase-hosting-for-free-9de8917bebf2>

Description of key classes and the application's layers



Main class

1. Start page
The index of the project
2. Log in page
Page used to log in, contains a log in box which has multiply functions
3. Account Setting page
A user first enter main page will be redirected to the account setting page where the user can enter their information
4. Main page
Main page of the ePortfolio system contains a navigation bar and templet to guide user
5. Edit page
User can use edit page to create or modify their e portfolio
6. ePortfolio page
User's own page of ePortfolio
7. Search page
A page used to search other user by name, university, city, country...



Testing

Unit test

Project Name	Invictus ePortfolio
Ver. Tested	Ver.0.10.15.3
Testing Start Date	2020/10/19
Testing End Date	2020/10/20

Input Number	Input	Expected result	Pass	Fail	Defect/ Comments
Part 1: Authentication and User info					
Function 1: signUp() function					
1	Existing email	Report error	✓		
2	Valid email	creat an account	✓		
3	Bad formatted email	Report error	✓		
Function 2: login() function					
1	Valid Account	Log in	✓		
2	Invalid Account	Report error	✓		

Function 3: resetPass() function					
1	Valid email	Send email	✓		
2	Bad formatted email	Report error	✓		
Function 4: logOut() function					
1	none	Report error	✓		
2	an Account	log out	✓		
Function 5: verifyEmail() function					
1	current user	send email	✓		
2	none user	report error	✓		
Function 6: update() function					
1	Valid list	store in database	✓		glitch spotted: will remove all previous data of the current user (as this function will only be used once while the user create the account it still pass the test)
2	Invalid list (incomplete info)	Print error message	✓		
Part 2: ePortfolio and File manipulation					
Function 1: SubmitButtonPressed()					
1	Valid String	store in database	✓		
2	None	Print error message	✓		

Function 2: AboutButtonPressed()					
1	Valid String	store in database	✓		
2	None	Print error message	✓		
Function 3: UploadButtonPressed()					
1	Valid file	store in database	✓		
2	Invalid file (too big >30mb)	throw	✓		
3	Invalid file (type mismatch)			✗	function can not handle file with a wrong type
Function 4: UploadpdfButtonPressed()					
1	Valid file	store in database	✓		
2	Invalid file (too big >30mb)	throw	✓		
3	Invalid file (type mismatch)			✗	function can not handle file with a wrong type

UAT

UAT Preparation

The software to be tested:

Invictus ePortfolio Ver.0.10.7

The date to hold the test:

2020/10/7

The hardware that test used:

The Desktop with Windows 10 as operating system

UAT Scope:

UAT in Scope

1. Authentication
2. Main page and ePortfolio page
3. Editing Page
4. Search Page
5. Account Setting Page

UAT not in Scope

1. Mobile testing

UAT Assumptions and Constraints

Assumptions :

Test environment:

- Via Zoom

Test document:

- All UAT test cases documented within google drive

Error reporting:

- All error reported in Test result report

Constraint:

- tested operating system Windows 10
- Browser test Chrome (Ver.85.0.4183.121)

UAT Risks

Description	Probability	Impact	Mitigation
Not properly trained UAT testers	Low	Med	Specified test cases and well formatted description for tester
Incompleted test environment	Med	Med	Realistic time and resources
Error Handling	Low	High	Easy to report bugs Solution provided in System
UAT test Failure	Low	Med	Completed description done before UAT start

UAT Role

Name	Roles	Responsibility
Shengze Zhu	Tester	Testing on Windows 10
Shengze Zhu	UAT Test Manager	Managing UST Test
Invictus Team	Product Owner	Project ownership

UAT Report

User Acceptance Testing

Project Name	Invictus ePortfolio
Ver. Tested	Ver.0.10.7
Testing Start Date	2020/10/07
Testing End Date	2020/10/12
Name of Tester/s	Shengze Zhu

Test Number	Description	Steps to Execute	Expected Results/ Acceptance Criteria	Pass	Fail	Defect/ Comments
Part 1: Authentication						
1	Sign up	Use the sign up box to sign up a new account	A new account should be created and a verified email should be sent to the email user entered. The animation of switching boxes shown correctly and buttons worked as clicked.	<input checked="" type="checkbox"/>		
2	Log in	Use a registered account to log in	Users should be able to log in if they entered their account and	<input checked="" type="checkbox"/>		

			password correctly. Error messages displayed otherwise.			
3	Reset Password	Use reset password button in login box to reset password	a reset password email should be sent to the email user entered. User can follow that email to reset password	✓		
4	Auto redirect	Log in a new and a existed account	After log in a new user should be redirected to the Account setting page and all other users should be redirected to the main page.	✓		
Part 2: Main page and ePortfolio page						
1	Create my ePortfolio	Click Create on navigation bar	Users can create their own eportfolio by clicking create and entering the editing page. Follow the instruction their, they can construct their own eportfolio	✓		
2	View Template	Click all three of	User should be able to view 3		✗	Pony's eportfolio

		templet to check if they work	template for inspiring to create their own eportfolio			redirect error (as it previously used for debug, can be fixed immediately after change href)
3	Slides Display/ function	Click slide button to check if it works	Slides can guide users on what to do next. It is displayed as entered in the main page and can change automatically. There are three of them and each has a function.	✓		
4	View my ePortfolio	Click view on navigation bar	Users should be able to view his/her ePortfolio where it has their Name, University, Occupation, Introduction, Skills, and Work.	✓		
Part 3: Editing Page						
1	Upload photo	On editing Introduction page click upload and choose jpg file	Users should be able to Upload jpg files as their photo displayed on ePortfolio	✓		Note: the function currently cannot check file type which

			<p>page. The function should be able to upload the image to the database . And the photo can be viewed by the user after being uploaded. Users can change their photo if they do not look good on the page.</p>			<p>can lead to display error, this can be improved .</p>
2	Submit About me Introduction	Use textbox and submit button on about editing page	<p>The user can write their own 'About' introduction on the editing page. After submit The page can displayed it to let the user view their page can modified it if needed.</p>	✓		<p>Note: the function currently cannot check String length which can lead to display error, this can be improved .</p>
3	Submit Skills information	Use textbox and submit button on skill editing page	<p>The user can write their own 'Skills' introduction on the editing page. After submit The page can displayed it to let the user view their page can modified it if needed.</p>	✓		<p>Note: the function currently cannot check String length which can lead to display error, this can be improved .</p>

4	Upload pdf work	On editing Work page click upload and choose pdf file	Users should be able to Upload jpg files as their photo displayed on ePortfolio page. The function should be able to upload the image to the database .	✓		Note: the function currently cannot check file type which can lead to display error, this can be improved .
Part 4: Search Page						
1	Access to Search page	On Main page, click search	Users will be redirected to the search page where there is a search bar and search option. Users can enter information there to search	✓		
2	Search function	Search page use search bar to check if it works	Choose the information sector user wants to search and enter key work. The System will display results matches where users can click to view the other's eportfolio.		✗	Yet to complete. Can only search names now. An intermediate page should be added to view the search result.
Part 5: Account Setting Page						

1	Access to Account Setting page	Use a new Account log in to enter main page	A new user will be redirect to this page where they provide information needed to construct their eportfolio	<input checked="" type="checkbox"/>		
2	Submit function	Use the Account setting box to submit information	Users should be able to submit information they entered into Account setting box. All information is needed and should be entered. Error messages displayed otherwise.	<input checked="" type="checkbox"/>		
Tester general comments			Currently have all basic function of a eportfolio system, only few minor issues to improve			