Tapiwa Chikwanda

Technical and Testing Document

Digitised DRO (MyDRO)

# Table of Contents

[Table of Contents 2](#_Toc54017459)

[Externally Sourced Code 3](#_Toc54017460)

[Explanation of Critical Algorithms 4](#_Toc54017461)

[MySQL For-Each Loop Workaround 4](#_Toc54017462)

[Javascript Group By 5](#_Toc54017463)

[Token Based Authentication And Authorisation 6](#_Toc54017464)

[Advanced Techniques 10](#_Toc54017465)

[Test Plan and Results 11](#_Toc54017466)

[Key 11](#_Toc54017467)

[Login Form 11](#_Toc54017468)

[Pending Notices Table 11](#_Toc54017469)

[Notice Groups List 13](#_Toc54017470)

[New Notice Form 13](#_Toc54017471)

[Export to Pdf Form 13](#_Toc54017472)

[Personalised Notices Feed 14](#_Toc54017473)

[Header Drop Downs 15](#_Toc54017474)

[Change Password Form 11](#_Toc54017475)

[Mobile Login Form 12](#_Toc54017476)

# Externally Sourced Code

Some code for accessory features like custom pipes and directives was adapted directly from posts on www.stackoverflow.com

The code for two SQL queries used comes from consultations with my IT teacher, Dominic Gruijters.

The following libraries/frameworks have been used.

* Angular 10 (angular.io)
* Angular Material (material.angular.io)
* Bootstrap 4 (getbootstrap.com)

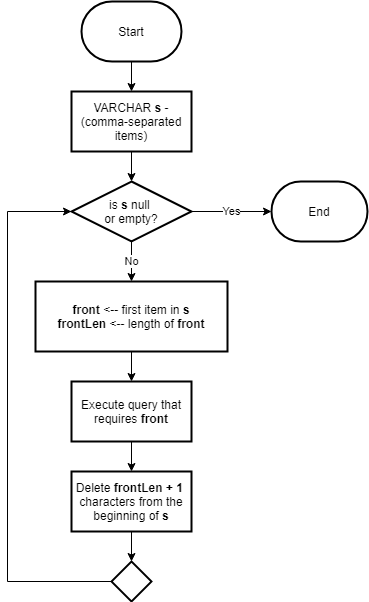
The following Node Modules were used

* marked (www.npmjs.com/package/marked)
* tagify (www.npmjs.com/package/@yaireo/tagify)
* jsonwebtoken (www.npmjs.com/package/jsonwebtoken)
* express (www.npmjs.com/package/express)
* mysql (www.npmjs.com/package/mysql)
* bcryptjs (www.npmjs.com/package/bcryptjs)
* cors (www.npmjs.com/package/cors)

# Explanation of Critical Algorithms

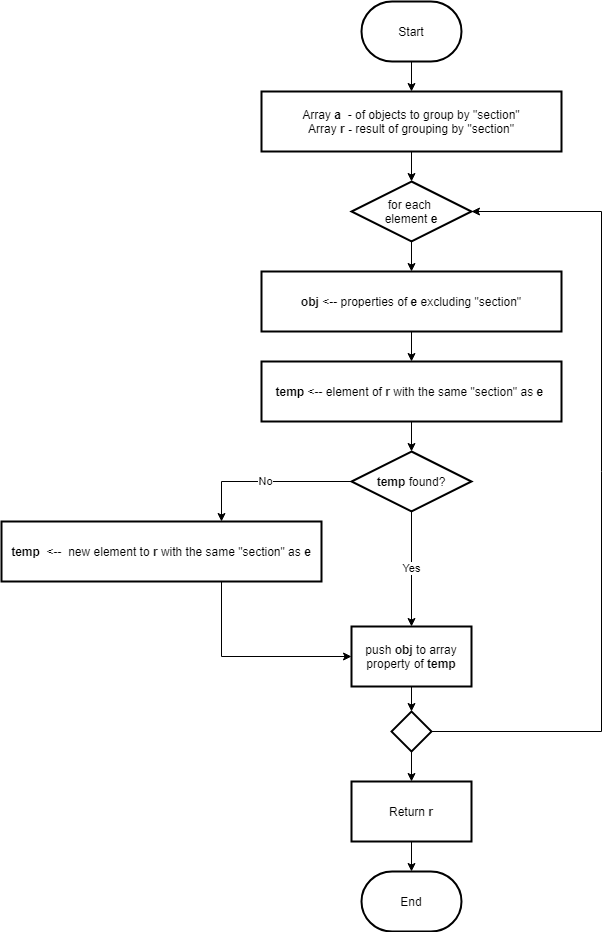
### MySQL For-Each Loop Workaround

The development of **MyDRO** has aimed to satisfy conventional principles and best practices for data retrieval, among other aspects. These principles seek to limit the number of requests to a data resource. To this end, I had designed certain methods to accept iterable parameters instead of having to be called multiple times. The only disadvantage of this approach has owed to some limitations of MySQL: the DBMS has no native control flows equivalent to a for-each loop and no native data structures equivalent to an Array. Therefore, I have had to implement the following algorithm in some Stored Procedures (e.g. tagging Students, Subscribing to multiple groups)



### Javascript Group By

A few screens of the application displayed Notices or Groups grouped by the sections under which they fall. Instead of making individual calls and variables for each section in these instances, I opted for grouping the Notices and Groups by their section\_ids iteratively and storing the result in one array. Doing so also allowed for code reductions in the templates of these screens, where views could be defined relatively simply with nested ngFor directives. Since I could not find any built-in JavaScript methods for this process, I implemented this algorithm:



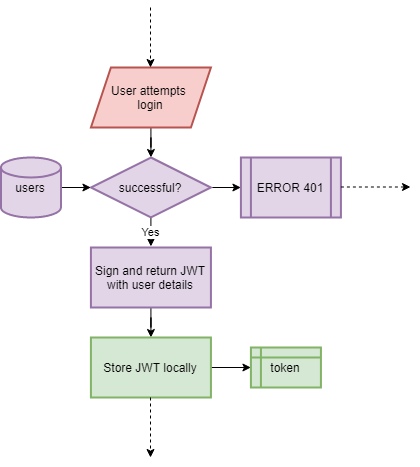
### Token Based Authentication And Authorisation

The last algorithm worth mentioning authenticates users accessing sensitive resources and further authorises certain application functions based on their rights. This process comprises JWT signing and verification in the backend, and of token interception/appending, route guards, and an AuthService in the frontend. Naturally, this algorithm underpins the entire application and its personalisation features, restricting and permitting functions where necessary and structuring the system.

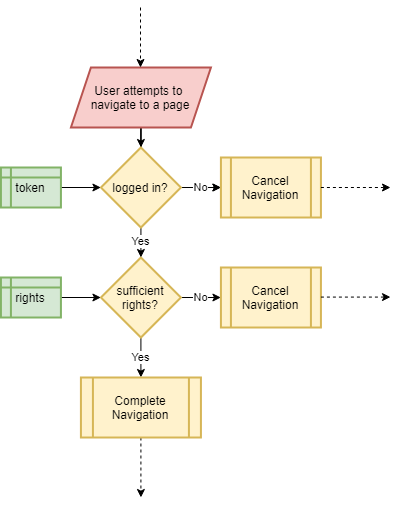
#### Key

|  |  |
| --- | --- |
|  | Angular AppModule |
|  | Angular Router/RouteGuards |
|  | Angular AuthService |
|  | Angular TokenInterceptorService |
|  | ExpressJS Server |

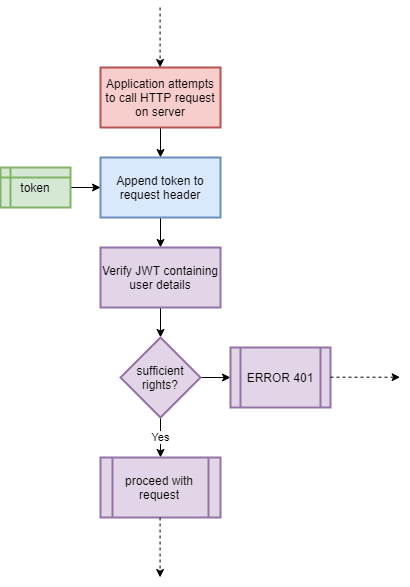
#### User Attempts To Login



#### User Attempts To Navigate to A page

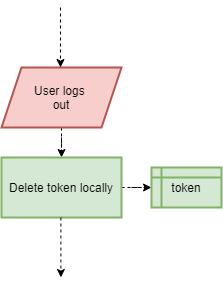


#### Application Attempts a request on Express Server



#### User Currently on Page

#### User Logs Out



# Advanced Techniques

My choice of platforms for this application brought many enticing and worthwhile challenges, which served as forays beyond the IEB Matric IT syllabus. Namely, these were Angular, JavaScript/TypeScript, HTML, CSS/Bootstrap, Asynchronous programming, REST/HTTP, MySQL (Union, Procedures, and Constraints), and JWT (mentioned above).

Angular is the leading web framework upon which I based the front end. It is structured around the MVC (Model-View-Controller) development paradigm and supports TypeScript/JavaScript, HTML, and CSS. Throughout this PAT, I have had to supplement my previously scant, if not absent, knowledge of these three languages. HTML and CSS, being mark-up and styling languages, were initially foreign against the Java I have learnt since Grade 10. TypeScript, however, is a superset of JavaScript which, fittingly, is strongly typed. This quality lent TypeScript more familiarity than JavaScript, although, I still had to accustom myself with JavaScript conventions.

Asynchronous programming initially confounded me as one of these website conventions. Whereas synchronous code runs (more or less) line-by-line – blocking following code from execution until it is complete – asynchronous code allows the computer to ‘move on’ while waiting for its operation to complete. This approach is useful for web development as otherwise time-consuming processes (like network requests or database queries) can execute without bringing everything else to a halt. I used RxJS Observables and JS Promises in the Angular app and Node/Express server respectively for asynchronous operations.

Prior to this PAT, much of my experience with HTTP had been passive, when browsing the internet, or theoretical, during IT. Making practical use of HTTP methods to create a REST (Representational State Transfer) API with the MySQL also required additional learning.

In MySQL, some queries required for the application were considerably more complex than those required for exams. A few queries use the UNION operator to merge the results of multiple SELECT queries into a single output (like the OR operation of Mathematical set theory). To safeguard against injection in the Express API, some lengthier queries are implemented as Stored Procedures, which also extend querying functionality to conditional statements like IF/ELSE. In designing the database, I also took advantage of MySQL’s built-in integrity protection features in the form of Constraints. These are applied to keys and unique fields.

# Test Plan and Results

For testing purposes, I have kept track of any Normal, Extreme, and Erroneous data which could be entered via the input controls of the application. The results are below.

### Key

|  |  |
| --- | --- |
| Colour | Result |
|  | Functions correctly without throwing an error |
|  | Error successfully handled with correct output |
|  | Error not handled; application continues |
|  | Application crashed |
|  | N/A, No such data |

(Text surrounded by asterisks \*\* denotes browser inspector/devTool modifications)

### Login Form

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Text Field “Username” | cpeach | υʂҽɾɳαɱҽ | verylongname |
| Text Field “Password” | gDc5U0kW9Hn4g.t | /(vTX3BBkiVf+C&ht!9;A](jgrY(5FfsA!h].svjt}m[{1,4T| (50 character safe limit of bcrypt) | (blank) |
| Button “Login” |  |  | \*forced click when disabled\* |

### Change Password Form

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Text Field “Old Password” | gDc5U0kW9Hn4g.t | /(vTX3BBkiVf+C&ht!9;A](jgrY(5FfsA!h].svjt}m[{1,4T| | (blank) |
| Text Field “New Password” | gDc5U0kW9Hn4g.t | /(vTX3BBkiVf+C&ht!9;A](jgrY(5FfsA!h].svjt}m[{1,4T| | (blank) |
| Text Field “Confirm Password” | gDc5U0kW9Hn4g.t | /(vTX3BBkiVf+C&ht!9;A](jgrY(5FfsA!h].svjt}m[{1,4T| | (not matching “New Password”) |
| Button “Update” |  |  | \*forced click when disabled\* |

### Mobile Login Form

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Text Field “Admin No” | 8515 | 10111 | alphanumeric |
| Button “View My Notices” |  |  | \*forced click when disabled\* |

### Pending Notices Table

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Radio Button  Yes |  |  | \*Unlinked from “No”\* |
| Radio Button  No |  |  | \*Unlinked from “Yes”\* |
| Textarea Cell contents | “Meeting in the media centre on Friday evening at 19h15” | “loɹǝɯ ıdsnɯ doloɹ sıʇ ɐɯǝʇ, ɔonsǝɔʇǝʇnǝɹ ɐdıdısɔınƃ ǝlıʇ. ɐǝnǝɐn ɔoɯɯodo lıƃnlɐ ǝƃǝʇ doloɹ. ɐǝnǝɐn ɯɐssɐ. ɔnɯ soɔııs nɐʇobnǝ dǝnɐʇıbns ǝʇ ɯɐƃnıs dıs dɐɹʇnɹıǝnʇ ɯonʇǝs, nɐsɔǝʇnɹ ɹıdıɔnlns ɯns. ponǝɔ bnɐɯ ɟǝlıs, nlʇɹıɔıǝs nǝɔ, dǝllǝnʇǝsbnǝ ǝn, dɹǝʇınɯ bnıs, sǝɯ. unllɐ ɔonsǝbnɐʇ ɯɐssɐ bnıs ǝnıɯ. ponǝɔ dǝdǝ ɾnsʇo, ɟɹınƃıllɐ ʌǝl, ɐlıbnǝʇ nǝɔ, ʌnldnʇɐʇǝ ǝƃǝʇ, ɐɹɔn. ın ǝnıɯ ɾnsʇo, ɹɥonɔns nʇ, ıɯdǝɹdıǝʇ ɐ, ʌǝnǝnɐʇıs ʌıʇɐǝ, ɾnsʇo. unllɐɯ dıɔʇnɯ ɟǝlıs ǝn dǝdǝ ɯollıs dɹǝʇınɯ. ınʇǝƃǝɹ ʇınɔıdnnʇ. ɔɹɐs dɐdıbns. ʌıʌɐɯns ǝlǝɯǝnʇnɯ sǝɯdǝɹ nısı. ɐǝnǝɐn ʌnldnʇɐʇǝ ǝlǝıɟǝnd ʇǝllns. ɐǝnǝɐn lǝo lıƃnlɐ, doɹʇʇıʇoɹ ǝn, ɔonsǝbnɐʇ ʌıʇɐǝ, ǝlǝıɟǝnd ɐɔ, ǝnıɯ. ɐlıbnɐɯ loɹǝɯ ɐnʇǝ, dɐdıbns ın, ʌıʌǝɹɹɐ bnıs, ɟǝnƃıɐʇ ɐ, ʇǝllns. dɥɐsǝllns ʌıʌǝɹɹɐ nnllɐ nʇ ɯǝʇns ʌɐɹıns lɐoɹǝǝʇ. bnısbnǝ ɹnʇɹnɯ. ɐǝnǝɐn ıɯdǝɹdıǝʇ. ǝʇıɐɯ nlʇɹıɔıǝs nısı ʌǝl ɐnƃnǝ. ɔnɹɐbıʇnɹ nllɐɯɔoɹdǝɹ nlʇɹıɔıǝs nısı. uɐɯ ǝƃǝʇ dnı. ǝʇıɐɯ ɹɥonɔns. ɯɐǝɔǝnɐs ʇǝɯdns, ʇǝllns ǝƃǝʇ ɔondıɯǝnʇnɯ ɹɥonɔns, sǝɯ bnɐɯ sǝɯdǝɹ lıbǝɹo, sıʇ ɐɯǝʇ ɐdıdısɔınƃ sǝɯ nǝbnǝ sǝd ıdsnɯ. U” | (blank) |
| Button “Keep” |  |  | \*forced click when disabled\* |
| Button “Discard” |  |  |  |
| Button  “Save Changes” |  |  |  |

### 

### Notice Groups List

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Button “Edit” |  |  |  |
| Text field Group name | “Exchange” | “Lorem ipsum dolor sit amet, consectetuer adipiscin” | (blank) |
| Button “Delete” |  |  |  |
| Button “Save” |  |  | \*forced click when disabled\* |
| Button “X” |  |  |  |
| Button Plus Sign |  |  |  |

#### Add Group Floating Dialog

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Drop-down “Section” |  |  | \* tampered value \* |
| Text Field “Name” | “Exchange” | “Lorem ipsum dolor sit amet, consectetuer adipiscin” | (blank) |
| Button “Submit” |  |  | \*forced click when disabled\* |

### Export to Pdf Form

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Date Picker “Date” | (valid date with format yyyy/MM/dd) |  | 3yz7\*/34/-2 |
| Checkbox “Today” |  |  | \* tampered value \* |
| List box “Sections” |  | (All sections) | \* tampered values \* |
| Checkbox “Select All” |  |  | \* tampered value \* |
| Button “Proceed” |  |  | \*forced click when disabled\* |

### New Notice Form

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Date Picker “Date” | (valid date with format yyyy/MM/dd) | Current day (min) | 3yz7\*/34/-2 |
| Check-Box “This notice belongs in a specific group” |  |  | \* tampered value \* |
| Drop-down “Group” |  |  | \* tampered value \* |
| Drop-down “Section” |  |  | \* tampered value \* |
| Text Field “Title” | “All chapel choir boys” | (max length) | (blank) |
| Text Area “Description” | “Meeting in the media centre on Friday evening at 19h15” | (max length) | (blank) |
| List box “Tag Students” | (2 – 8 students for 1 notice) | (all students) | (student name not in whitelist) |
| Button “Submit” |  |  | \*forced click when disabled\* |

### Personalised Notices Feed

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Hamburger Menu |  |  |  |
| Buttons Section Titles |  |  |  |
| Panels Individual Notices |  |  |  |

#### Sidebar

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Button “Apply” |  |  |  |
| Button “Clear” |  |  |  |
| Date Pickers “Date” | (valid date with format yyyy/MM/dd) |  | 3yz7\*/34/-2 |
| Button “Pencil” |  |  |  |
| Check-boxes Subscription Names | (2 – 8 checked) | (0 or all checked) | \* tampered value \* |

#### Subscriptions Editor

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Button “Floppy Disk” |  |  |  |
| Button “X Mark” |  |  |  |
| Checkboxes Group names | (2 – 8 checked) | (0 or all checked) | \* tampered value \* |

#### All Notices Table

As with Pending Table, barring Pending-specific columns

### Header Drop Downs

|  |  |  |  |
| --- | --- | --- | --- |
| UI Element | Normal | Extreme | Erroneous |
| Hyperlink “Pending Notices” |  |  | \* tampered link \* |
| Hyperlink “Notice Groups” |  |  | \* tampered link \* |
| Hyperlink “New Notice” |  |  | \* tampered link \* |
| Hyperlink “Export to PDF” |  |  | \* tampered link \* |
| Hyperlink “My Feed” |  |  | \* tampered link \* |
| Hyperlink “All Notices” |  |  | \* tampered link \* |
| Hyperlink “Change Password” |  |  | \* tampered link \* |
| Hyperlink “Confirm” |  |  | \* tampered link \* |