**Summary**

My high school currently employs an ineffective notice system between staff members and students. Abbreviated to DRO, the Daily Routine Order has sprouted a variety of complaints: some of which prompting, and others resulting from, an interview I had held with the School's Pastoral Administrator. Currently, DRO operation is as follows. Before a daily cut-off time of 08h30, staff members may send via email to the Administrator a notice for the day's DRO. After the cut-off, the Administrator spends 30 minutes at the shortest, manually sifting through these notices for problematic formatting. Once ready, a notice falls under one of six broader categories – To See Staff; [General-]Notices; Sport; Clubs and Societies; Sanatorium; [Detention] Restrictions) – which are visually separated on the final .pdf of the DRO. In time for mid-morning breaks, hard-copies of this file are printed and distributed for display at high-traffic areas on campus - such as the Boarding Houses, Media Centre, and Dining Halls - until removed the following weekday.

Unfortunately, strategic placement does not guarantee that a DRO message will reach its recipients, let alone the student body at large. This is the foremost access shortcoming of the DRO. Moreover, while the current DRO system is in some respect digital, its processing shortcoming is the required manual collation. I intend to address both shortcomings with my proposed application, automating collation by staff and providing electronic access to students.

**Specifications of Program Function**Users will belong to one of three categories, Students, Staff, and Administrators. These will grant cumulative levels of privileges to the DRO System i.e. all Admins have Staff privileges; all Staff have Student privileges. Only the minimum authorisation required is shown.

The application functions comprise 3 broader categories, falling primarily to the class of users in parentheses:

* **Management** of the DRO, its structure, and its users   
   (**Admin**)
* **Publishing** content to be shown   
   (**Staff**)
* **Viewing** notices as required   
   (**Students**)

1. Authentication and Authorisation (Students) – Allow users to login using some form of unique identification and password, and gain their relevant access to the DRO system

* Show login fields (ID and Password) and allow user to type into both
* If both valid, fetch and apply user rights
  + Progress to respective landing pages
* Otherwise, inform user of error

1. Notice Request (Staff) – Add to, edit or, delete from the DRO, subject to Admin approval

* Allow Staff Member to supply the notice’s Description, Dates, Times, DRO Group, Specific Students, and Recurrence
* Notify user of declining

1. Notice Approval (Admin) – Allow approval/declining of DRO publishing requests

* Show pending requests
* If approved, give option to edit formatting before publishing
* If declined, exclude the notice from the DRO

1. Export to PDF (Admin) – Optional formatting for a single day’s DRO

* Allow selection of Date and Groups to display
* Output a .pdf file with the required information

1. View Notices (Students) – Allow user to view and filter notices

* A categorised list of the day’s notices shows
* The list may be filtered by Group/Subscription

1. Configure Notice Groups (Admin) – Allow admin to perform CRUD on notice groups

* Show, Edit, Add, and Remove Notification Groups

1. Subscribe to Groups (Students) – Allow user to specify which groups’ notifications they receive

* Select/Deselect groups of interest to associate with profile

1. Alert on Notice (Students) – Ping user about notices which they have either subscribed to or have been mentioned in

* Once such a notice is posted for within the following week, show an alert- or reminder-based notification to the user
* If the user clicks the notification, show the full notice

**Specifications of UI**

The application will consist of two distinct frontends; each interfacing, however, with a single backend which is further specified on page \_:

1. A Desktop Web-Based GUI   
    **(Admin and Staff)**
2. A Mobile Progressive Web App (PWA) or Native Apps for iOS and Android **(Students)**

Accordingly, the specific user interfaces lend themselves to either of the two frontends.

Website (Admins and Staff)

Specifications of Help

Specifications of Data Storage

Hardware and Software Requirements

(Based on Chrome Requirements)

Hardware: Intel Pentium 4 with SSE2 compatibility

Software: Browser which supports HTML5, JS ECMAScript 5, CSS3 -- Google Chrome v50+

Windows -- version: 7 or Later; Secondary Storage: 500 MB(Chrome), OS Size

Mac --