

# **INFS3202 / INFS7202 - Group Proposal**

Arun Harish Balasubramonian

44340326

Tze Thong Khor

44260394

Saisrikar Paruchuri

44751575

P01

---

## Introduction:

This document proposes a To-do task manager web application called "*StudentTM*", with few features targeting UQ students, and includes the mechanism for both server-side and client-side technologies of the website, following Model-View-Controller (MVC) architecture. The application is designed to organise students' tasks based on their learning session, course period and timetable, which requires the layout of the website to be intuitive for the mental model of the users.

Hence, the application includes side panels listing options; notes with drag and drop interaction for priority and arrangement; user authentication prompts and other familiar interfaces. These features entail the use of PHP, as both Model and Controller, and JavaScript, as Controller sending commands to PHP and manipulating Views such as HTML5 and CSS3 for user presentations, which follows web standards.

## Web standards

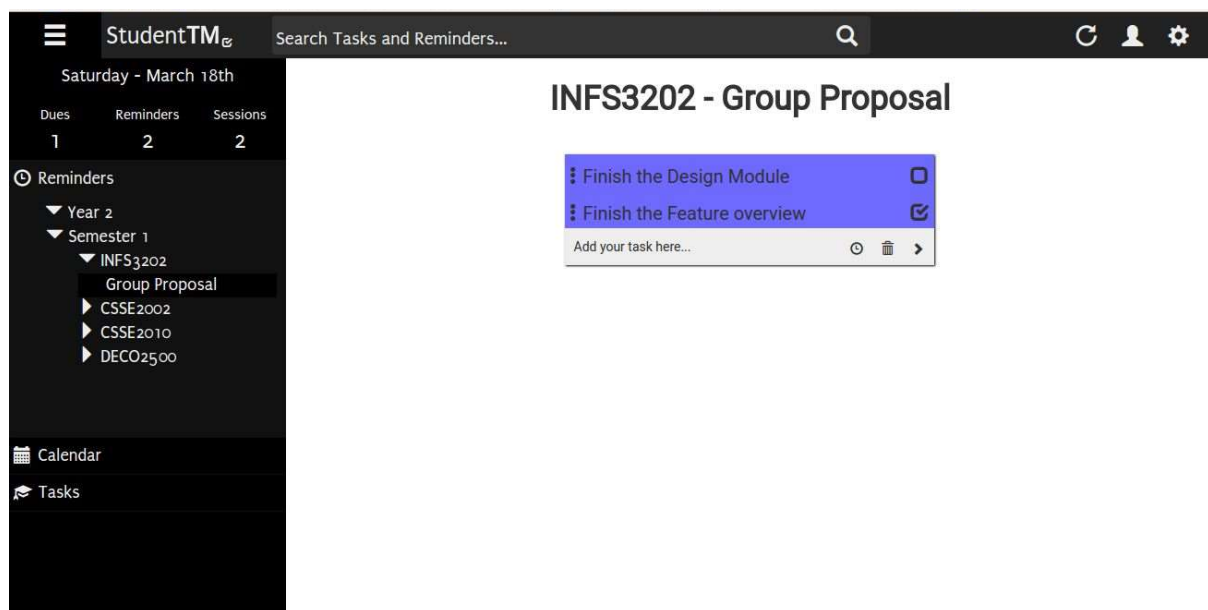
Web standards enable a programmer to create contents that are browser compatible, and device compatible, allowing them to reach their target audience. Moreover, certain browsers might experiment their technologies that may not be supported by other diverse vendors. Following web standards ensures a web developer to use technologies that are consistent across devices, helping their content to be correctly rendered and functional.

Further, adherence to web standards increases the accessibility of the website. Blind users use text-to-speech software that is designed to work perfectly when the parsing websites follow the web standards, making possible for the user to understand the content of the website.

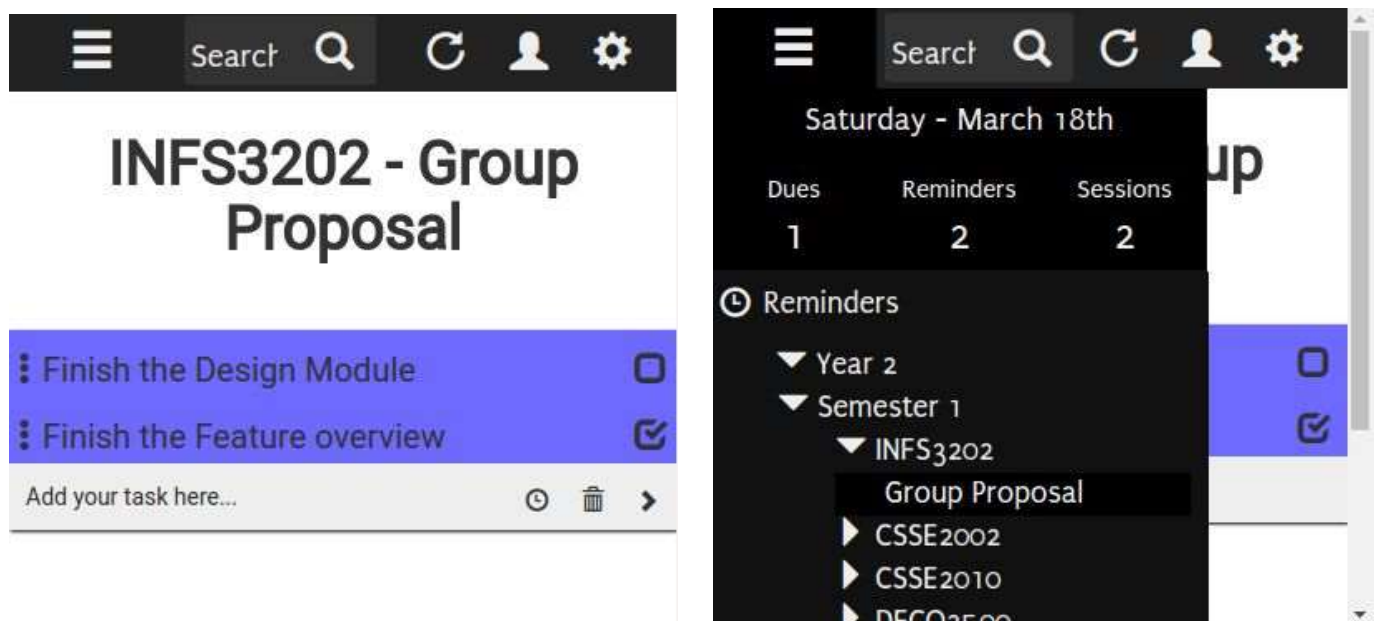
Besides, the standards also provide backward-compatibility for old browsers that are not compliant with modern standards but still render other contents properly along with fallback content of the modern technology. For example, HTML5 canvas element is incompatible with old standards but renders the child elements that are compatible. The application uses HTML5 standard targeting modern browsers that are compatible. Although few browsers may not support modern APIs, the interfaces would be rendered intact if followed to the web standards. Also, the users of the application being students, who often use diverse devices with different screen resolution, it is the design requirement for the website to be responsive.

## Design:

The UI requirement of the application is that it should be natural to the mental model of the user. Given the target audience of the application are University students familiar with app interfaces, the following designs are produced.



**Figure 1: Shows the interface of the application in a standard desktop environment**



**Figure 2: Shows the interface of the application in a mobile environment**

From the above images, the application uses layout, and colours along with interface metaphors to enable the user an intuitive understanding of the website. For example, the magnifying glass implies the action of the find, and its orientation near to the search bar relates its functionality and gives an integrated design enhanced by the same colour choice.

Besides, the application encompasses most common interfaces such as tabs, tree structures and icons. The tab interface inside the side panel uses colours to emphasise the selected tab and its content: the background of the unselected tabs have the same colour as the side

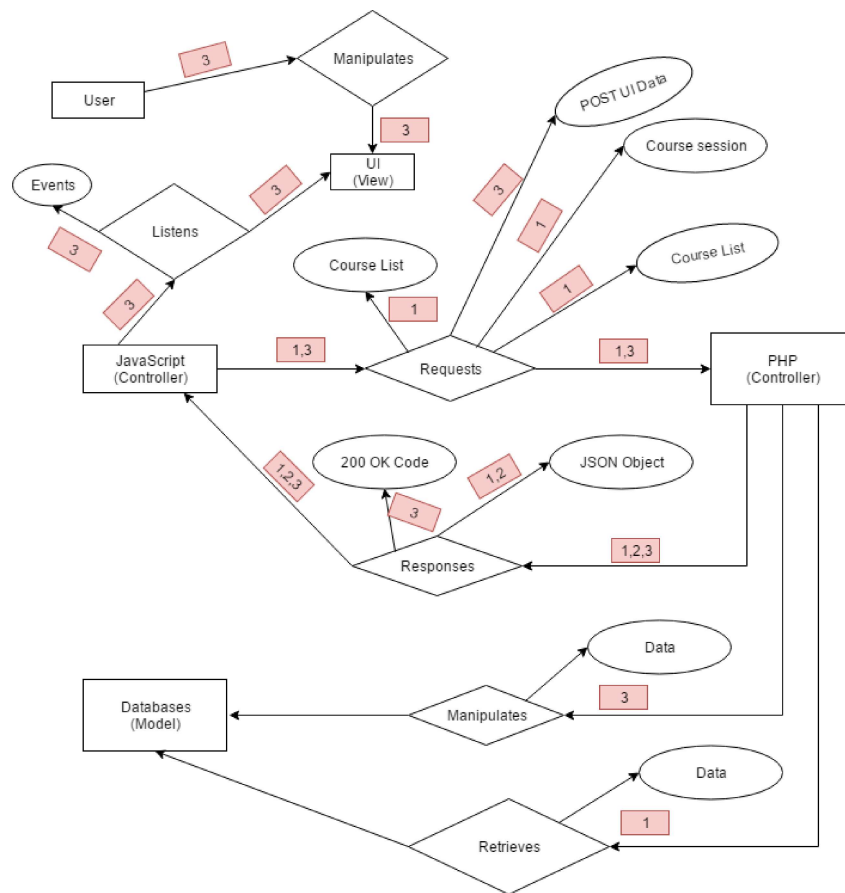
panel indicating them as inactive, while the selected tabs have light grey. Further use of layout is evident in the tree-structure widget.

The tree-structure widget uses CSS padding to differentiate parent and child elements, along with arrows indicating their collapsed status. The padded child elements provide a hierarchy that enables the user to locate their subject notes and reminders easily. In Figure 1 the Task list for Group Proposal are visible, and the tree-structure indicates its subject, which is particularly useful when there are same reminder titles relating to different subjects. Additionally, the application also uses relevant layout and icons even in mobile screen resolution to give users a seamless interaction.

Figure 2 shows the responsive design of the application changing layout according to the screen resolution. In small devices, it is necessary that the main interface is visible and well prominent than other menus, which might distract the user. Hence, the design removes the logo of the application while the user is well aware of the website they are visiting.

Also, the side panels are removed, but still accessible via the hamburger icon indicating this option: the user knows its functionality as it is commonly used in many applications, being natural to the mental model of the user. Thus, the UI of the application is logically designed, using layouts, contrasting colours, and icons to provide relevant information based on the user environment. Moreover, the use of client-side languages to interact with server-side is also pivotal, and the following outlines both frontend and backend mechanism.

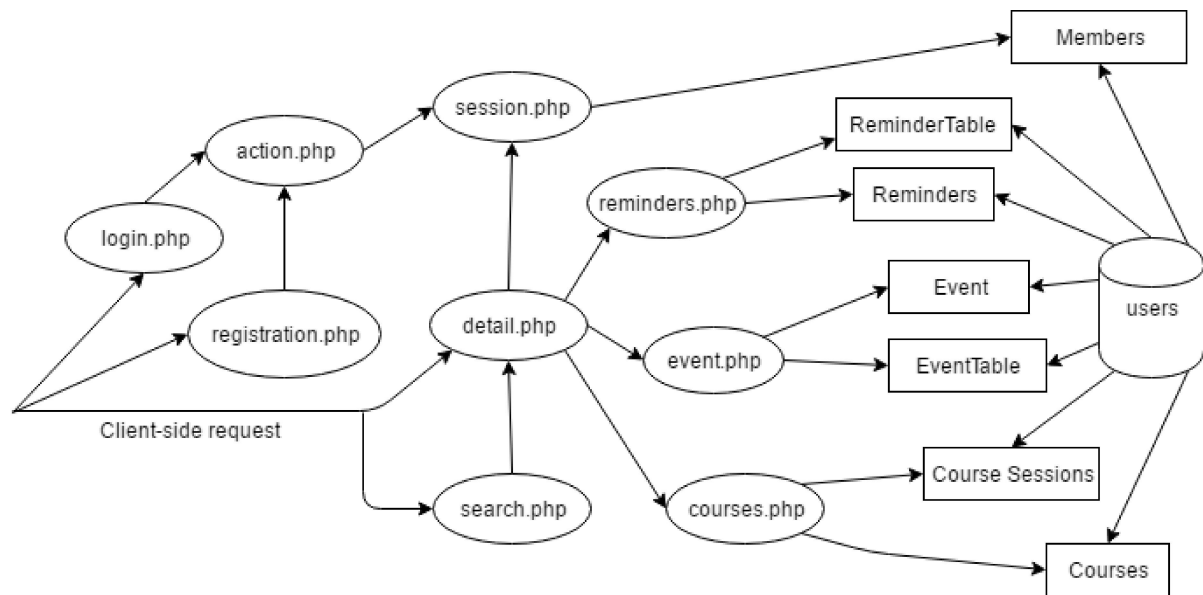
## Feature Coverage



**Figure 3: Shows the Entity Relationship diagram of the communication between technologies. Numbers represent the sequence of interaction.**

The application utilises the MVC architecture to develop and better organise the interaction between the server and the client. There are sequential processes using technologies, in which both client and server interact: the server-side uses MySQL framework for PHP and the client-side uses HTML5, CSS3 and JavaScript. The following describes Figure 3 showing possible communication from JavaScript, acting as a Controller sending commands to PHP for retrieving and manipulating data from the database (Model); assuming the user authentication is validated:

- 1) Dynamic web page requests information about user profile, courses, course sessions and reminders/notes and the server responds with JSON objects.
- 2) JavaScript parses JSON objects and controls the views.
- 3) The user manipulates the UI (view) using interactions, and JavaScript sends POST XMLHttpRequest to PHP, which uses MySQL framework to manipulate data.



**Figure 4: Shows the Model aspect of the application in detail: Cylinder represents the database; Rectangle represents tables and Oval represents server-side PHP pages.**

In addition to Figure 3, the backend of the application involves certain in-depth processes as outlined in Figure 4 above. The following clarifies those processes in detail:

- User registration and validation requires a lookup into Members table of the “users” database. The session.php looks into the database using MySQL and validates them by responding back to action.php and creating a session for the user. The login.php then redirects to users page.
- Further client-side requests are user validated via the session.php, to manipulate or retrieve data. The detail.php functions has one-way interaction getting client request, validating them using session.php and communicates with reminders.php or courses.php or event.php depending on the request. The search.php gets information from detail.php and filters them based on the query.
- The Reminders and ReminderTable have Foreign Keys with Reminders being the parent table. Also, Reminders have a Foreign key with Members table. The Reminders table contains Primary Keys for the unique set, a description column listing users reminder/note titles and Foreign Key to Courses table.
- The Course Sessions and Courses have Foreign Keys to each other with Courses and Members being the parent key, i.e. Courses table has a unique list of courses and Members table related to the Course Sessions.
- The Event and EventTable are similar to Reminder and ReminderTable respectively: EventTable has a primary key for the unique set, Foreign key to the courses table and a description of Event and Event table contains Foreign Key of the EventTable and members.

Thus, the application uses both the client-side and server-side interaction following MVC

architecture and creates a mechanism through which the backend features such as user authentication, database manipulation and update occurs based on the request of the UI. Moreover, such interactions could not be possible without the use of JavaScript and relevant HTML5 API as views.

Figure 1 showed the main reminders/notes interface, which contains the drag icon. Using HTML5 drag and drop API and JavaScript, users can either use touch or drag events to either rearrange using drag up and down or remove their notes by dragging sideways. Drag and drop function will also be used to move the events or reminders from the event panel to the user's calendar. Moreover, the application would also use Geolocation API along with Google Maps API to notify them of nearby reminders. Besides, jQuery will be used in the application to create sliding effect on side panel which shows all the task and information. The application page will become simple and clean with this feature as it will save a lot of space.

As mentioned before, the application will search information such as assignments, events and many others. This function can be achieved by searching the data and retrieving it from respective PHP modules, which uses MySQL frameworks. Further, the application would use AngularJS, as its main JavaScript framework, while it provides expressions that can translate data into UI easily. Also, AngularJS has the tendency to include and run parallel with JQuery library, which will be used for Ajax requests. JavaScript will also have functionalities for the interface provide by Bootstrap, which will be included in UI.