**Project README and Log - Orbital – Tay Guo Qiang & Won Jun Ru Daphne (Documentation for Milestone #2)**

**Project Title: Schedule Comparator**

Table of Contents

**Readme 2-5**

Overview2

Intended features 2

User stories2

Project status2

Features finished up to now3

Planned features after current milestone 5

Proposed level of achievement 5

References used 6

**Project Log6-7**

Time spreadsheet 7

**README**

|  |
| --- |
| **Overview**: Our project is to come up with a schedule comparison web application that helps groups of people to compare schedules and find out a common time available for a meet-up. We plan to have both the manual entry and the timetable-comparison option in our web application.    Slide for project idea  **Intended features:**   * Temporary sessions created using MySQL database with PHP, where groups of users can come together to collate the availability schedules of each person without requiring logins * Date-picker input for users to enter inputs * NUS Timetable URL upload comparison option (to implement in due time)   **User stories:**   * As a general user, I want to use this web application to know INSTANTLY which date everyone can meet up for projects/gatherings using the manual entry option. * As a general user, I want to be able to adjust my input schedules flexibly when I need to change my schedules in the midst of the comparison. * As an NUS student, I want to be able to use my NUSMods timetable together with my fellow NUS friends’/project mates’ timetables to find a common date where we can all meet up together, using the timetable-comparison option.   This is somewhat different from Doodle Poll, which is contrary to general opinion. For Doodle Poll, a date is set and the group of people indicate in Doodle Poll whether they are available on that specified date or not. However, for our Schedule Comparator, everyone’s schedules are collated together and the comparator attempts to find a common date that everyone can meet up together. If there is not a single common date where everyone can meet up, certainly a compromise is needed in the schedules.  **Project status:**   * Link to video: <https://www.youtube.com/watch?v=ZiJUVi7jKXE> * Link to application is currently not available since we have not intended to deploy the product yet, since even one of the 2 sections (the manual comparison option) is not yet complete. However, the demonstration video as well as screenshots of our application shown in the next pages should offer a clear idea, mostly for the manual comparison section, on the features. * In summary, the first section of the product is currently not ready for acceptance testing.   **Features finished up to now:**   * The general application interface has been completed, but may be further polished over time. * Sessions system using PHP-MySQL has been implemented.   For clearer elaboration, screenshots of our website are shown below.    Home page with help dialog    Sessions page with input validation and exception handling    Manual option with date-picker interface    Timetable option (currently a placeholder)  Elaboration about our application:  The sessions system:   * When one member in the group creates a session, effectively he/she is creating a table with table name set as the input session ID in a database that has been pre-created for our comparator. * Other members in the same group that joins that particular session would be effectively accessing the same table in the database. * Inputs from both the session-creator and the other members would simply be appended to the table. * When every single person in the group is done, the common dates can be computed from the table data. * When the group no longer wishes to use this comparator, the creator can destroy the session, which effectively deletes the table so that it is rendered inaccessible anymore.   Manual comparison system:   * For this comparison, every group member will gain access to the same page (based on their group session ID). All members in the group are to fill in their schedules with the interface and append their entries to the table. * The intention of the page is to fetch the updated table each time an entry is appended to the table. It should also allow each user to edit their own entry’s schedule if there happens to be a change while in the midst of the comparison process.   Timetable comparison system:   * For this comparison, every group member will gain access to the same page (based on their group session ID). All members in the group are to input their NUSMods timetable URLs. * The intention of the page is to make use of the NUSMods API to retrieve the common available dates and times in the users’ timetables.   Application exit:   * When the group is done using the application, the members who joined the session would leave the session upon clicking the exit button whereas the one who created the session would destroy the session as well as exit the session upon clicking the exit button.   Input validation & Exception handling:   * First scenario would be SQL injection attacks, which would totally ruin the functionality of our application’s session system if SQL queries in the form of ‘DROP TABLE’ or ‘DROP DATABASE’ manage to slip into our application. Therefore, our application does not permit sensitive SQL keywords such as ‘TABLE’, ‘DROP’, ‘DATABASE’, etc. * Second scenario would be when a particular session ID is already in use. To prevent different groups from accessing the same table via the session ID, we check to see if a table of the same session ID name exists before allowing the session-creator to proceed.   **Planned Features after current milestone:**   * We intend to use Ajax to update table entries in the session via MySQL queries without having to reload the page, for the manual option. * We intend to allow users to modify their own inputs during the comparison process. * We intend to adjust the interface such that users can now also select time ranges and account for this section of their own schedules during the comparison process. * We intend to make use of NUSMods’ API for NUS students to compare their timetable schedules with each other simply by inputting their NUSMods timetable URL. * The Schedule Comparator would only be deployed after full completion.   **Proposed level of achievement:**  We are aiming for the Gemini level of achievement for our project. We propose why we should be granted the Gemini level of achievement below:  We have used different platforms as listed below:   1. We use a WAMP stack architecture for our Schedule Comparator instead of using GAE 2. We use multiple Javascript components for our web application (i.e. PHP, Ajax, jQuery) 3. Our sessions system serve as an administrative front end for our Schedule Comparator 4. We have input validation and exception handling to guard against SQL injections (since we are using MySQL for our tables-implemented sessions system) 5. We use Git for version control, moving on from the Google Drive phase   We are also active in our application development and a significant portion of our application is already completed as of this point.  **References Used:** *For self-learning:*   * W3Schools (<http://www.w3schools.com/>) – things for web development can be learnt here * Codecademy ([http://www.codecademy.com/)](https://developers.google.com/appengine/docs/python/%29) * Code School (<https://www.codeschool.com/>) – however to complete the full course, payment subscription is needed * Free Code Camp (<http://www.freecodecamp.com/>) * Meteor.js tutorial (<http://meteortips.com/first-meteor-tutorial/projects/>)   *Features that we used in our Schedule Comparator so far that we managed to find:*   * Multi Date-Picker (<http://multidatespickr.sourceforge.net/>) for the date-picker feature we used * Bootstrap (<http://getbootstrap.com/>) for the Bootstrap we used * jQuery (<https://jquery.com/>) for the jQuery used * PHP (<https://secure.php.net/>) for the PHP used * MySQL (<https://www.mysql.com/>) for MySQL used |

**Project Log**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| We have met about 2/3 of the requirements for Orbital, which is calculated from our current progress so far. We will continue our development of the Schedule Comparator from this point as specified in the planned features section, in which we would be meeting up more often for our web application development.  Specified here are the accumulated time figures. For more in-depth elaboration of how we used these hours, please see the spreadsheets in the next page.  Accumulated hours for Guo Qiang: 109  Accumulated hours for Daphne: 96  Total hours together: 205   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **S/N** | **What** | **Date** | **Guo Qiang (Duration in hours)** | **Remarks** | **Daphne (Duration in hours)** | **Remarks** | | 1 | Liftoff Day 1 | 11/5/2015 | 8 | - | 8 | - | | 2 | Liftoff Day 2 | 12/5/2015 | 8 | - | 8 | - | | 3 | Self-learning (HTML/CSS/Bootstrap/jQuery) + Website creation | 14/5/2015 to 22/5/2015 | 30 | Self-learning and coding was done simultaneously during this period. Self-learning took up 6 hours and the coding took up 24 hours. | 30 | Self-learning from Codecademy (HTML/CSS/Javascript /Python) | | 4 | Attending MC #2 | 27/5/2015 | 2 | Meteor.js | - | - | | 5 | Self-learning (Ruby) | 30/5/2015 to 31/5/2015 | 5 | - | - | - | | 6 | Self-learning (SQL) | 31/5/2015 to 5/6/2015 | 7 | - | - | - | | 7 | Self-learning (Git) | 31/5/2015 to 2/6/2015 | 3 | Some basics of Git have been learnt. Learning phase is still incomplete though. | - | - | | 8 | Self-learning (jQuery) | 1/6/2015 to 23/6/2015 | - | - | 12 | - | | 9 | Self-learning (PHP) | 5/6/2015 to 21/6/2015 | - | - | 10 | - | | 10 | Self-learning (PHP) | 7/6/2015 | 2 | - | - | - | | 11 | Self-learning (Ruby) | 10/6/2015 to 13/6/2015 | - | - | 12 | - | | 12 | Website coding (adding on PHP-MySQL functionality) | 18/6/2015 to 28/6/2015 | 33 | See next ‘Remarks’ | - | Difficulties with PHP and its installation/configuration was faced here, thus resulting in much time spent in this phase | | 13 | Meet-up 1 | 24/6/2015 | 6 | See next ‘Remarks’ | 6 | To debug PHP issues | | 14 | Mission Control: Android App | 24/6/2015 | - | - | 2 | - | | 15 | Edit design for webpage | 25/6/2015 | - | - | 3 | - | | 16 | Meet-up 2 | 28/6/2015 | 5 | See next ‘Remarks’ | 5 | For milestone 2 | |  | **Accumulated Hours** |  | **109** |  | **96** | **Total: 205** | |