

Department of Accounting & Information Systems

ACCT/INFO GROUP PROJECT SUBMISSION

CASE STUDY/PROJECT TITLE tServer Web Interface

Please complete all sections of this sheet, sign the declaration and attach the sheet to your project.

The next panel must be completed by all team members, **including** the agreed proportion of work done on the project. (For example, if all members of a team of four made equal contributions then enter 25% for each team member.)

	Student ID No.	User ID e.g. afg21	Student Names: (Surname first & alphabetical order please)	Proportion % (Agreed by group)
1)	<u>75400547</u>	<u>hdd19</u>	<u>Dacillo Hezekiah</u>	<u>25 %</u>
2)	<u>68697438</u>	<u>kll60</u>	<u>Luu Khanh Linh</u>	<u>25 %</u>
3)	<u>25082165</u>	<u>bvv10</u>	<u>Vu Viet Bach</u>	<u>25 %</u>
4)	<u>72004622</u>	<u>ezh15</u>	<u>Zhang Engang</u>	<u>25 %</u>
5)				

Honesty Declaration

- I declare that this is an original assignment and is entirely my own work.
- Where I have made use of the ideas, words or work of others, I have acknowledged the source in every instance.
- Where I have used any diagrams (including modifications) prepared by others, I have acknowledged the source in every instance.
- I have read and understood the Dishonest or Improper Practices Statement overleaf.
- I am aware of what constitutes cheating, and the penalties for plagiarism and cheating as described in University publications.
- I am aware that the content of this written work may be checked against an electronic database.
- I have supplied the correct word count and have taken no steps to cause disclosure of an incorrect word count for the assessment.

I have read and fully understand the Honesty Declaration above, and hereby certify that this item of work submitted for assessment is entirely the work of the members of the group, in the proportions stated.

Signed . . .

1) <u>[Signature]</u>	3) <u>[Signature]</u>	5)
2) <u>[Signature]</u>	4) <u>[Signature]</u>	

Under the University Regulations, evidence of any of these or other forms of dishonest practice by any student(s) represents grounds for disciplinary action and may result in penalties ranging from denial of credit for the item or work in question, to exclusion from the University.

Dishonest or Improper Practices

It is recognised that students will discuss course work and assignments with others, and such discussion is an important part of the learning process. However, any work presented by a student for credit in a course must be that student's own original work. If students are directed to complete work submitted for credit in groups, the work submitted must be the original work of the group. Work submitted in breach of these requirements or which fails to comply with other instructions contravenes the University's Dishonest Practice and Breach of Instruction Regulations. Such work will either not be marked, and all credit for the work in question forfeited, or the matter will be referred to the University's proctor for investigation and possible referral to the University's Disciplinary Committee.

Penalties which may be imposed in the event of a finding of dishonest or improper practice include loss of credit for a course or an item of assessment and, in serious cases, suspension or expulsion from the University. A record is kept of all instances of dishonest conduct.

Instances of dishonest or improper practice in coursework and assignments include but are not limited to:

- ❖ Plagiarism. Plagiarism means the dishonest presentation of work that has been produced by someone else as if it is one's own. Please note that the presentation of someone else's work as one's own, even without dishonest intent, may still constitute poor academic practice, and this may be reflected in the mark awarded. There are academic conventions governing appropriate ways to acknowledge the work or part of the work of another person, including the APA and Harvard citation styles. For further information see the UC Library website, under "Citations and Referencing".
- ❖ Submitting for credit in a course without the prior consent of the Course Coordinator for an essay, research paper or any other written work which, although it is the student's own work, is substantially the same as work which has already been (or will be) submitted for credit in another course, whether in the Department of Accounting and Information Systems (ACIS Department) or some other department or academic institution.
- ❖ Copying the work of another student. This includes copying the work submitted by another student for credit for a course in the ACIS Department or some other department or academic institution.
- ❖ Knowingly allowing another student to copy work which that other student then submits for credit for a course in the ACIS Department.
- ❖ Arranging for another person to complete work which is then submitted for credit for a course in the ACIS Department. An example falling in this category is work submitted for credit which has been obtained from a commercial assignment completion service. Care must be taken when using editing services as it is **only** assistance with grammar, punctuation and expression that is permissible and does **not** include the addition or amendment of content.
- ❖ Completing work for another student which is then submitted by that other student for credit for a course in the ACIS Department.
- ❖ Including made up or fabricated material in work submitted for credit for a course in the ACIS Department.
- ❖ Collaborating in the preparation of answers for take home or online tests unless advised otherwise in the take home test instructions.

If you are in doubt about any of the above with respect to a particular course, you should discuss the matter with the lecturer or course co-ordinator concerned.

See also the University Discipline Regulations, Dishonest Practice and Breach of Instructions Regulation, and Academic Integrity Policy – refer to UC Calendar and UC web.

INFO263 PROJECT

tServer Web Interface

Introduction

tServer Web Interface is an INFO263 project with the purpose of scheduling events such as exams, tests, tutorials, and so on that bring more manageable for the teaching staff. Students are to form a group of three or four students and produce a web application that would help with the scheduling.

Teamwork

Our team started the project by identifying user tasks and type of users, of which we decide not to implement sign-up features since the organization should be responsive for who get access to the database. Every week, we have a meeting to plan for the following tasks and create a backup to the master branch. This process helps all members stay aware of their duty to finish their task. We also used a GitHub repository with each member's branch so everyone can work in parallel.

Login Page

The login page is a template from Colorlib[1] with a slight modification to our team's likeness.

We assume that a table of users (teaching staff) already exists in the university database. Therefore, we create a table of users that contains each team member's credentials and manually adding a hashed password using a built-in PHP function of `password_hash`.

To validate the user, a procedure that takes the username and outputs the hashed value called. This function uses a built-in PHP function of `password_verify` that verifies if the entered password of the user matches the hashed password stored in the database.

We also decided to use sessions for user logins and timeouts. The user will automatically be logged out after 10 mins of inactivity.

Note that login credentials information for testing are all mentioned in README.

Home page

After the user has logged in, they are redirected to the home page. The home page/dashboard shows up to six upcoming events from `tomorrow` to next month. If there are no future events, there will be a message informing the user that they have no incoming events. In the dashboard, it also displays up to 6 current events that are happening and 5-10 previous events. We decided to divide the page into three sections for future events, current events, and past events. This format is to increase readability, provide quick and concise information to the user without using the extended search feature.

Defining Events

Given the database, our team decided to interpret and represent an Event as follow:

Events with cluster 'Labs' should not be visible on the UI. Those records are for the network daemon to reserve the physical lab for a test. Also, Events with `activate=0` is not shown on

the UI, since the user would be interested in the starting time of an event, not when that event ends.

Event with the same name (same event_id in front_event) but has different cluster name and time shall be display as separated events.

Each of such event may appear as multiple records in 'vw_front_event' due to having different Locations (group_machine), so we shall merge these records to a single event with multiple locations.

In this way, it is easier to understand than using a direct result from 'vw_front_event' (i.e. front_action is dependent on event_id, which as described in bullet point 2, can be different events according to our team definition). This solution allows for more robust for future improvements.

The Create New Event page follow concepts above, allows the user to define the new event to the database. The event name is divided into three parts: course name, year-term and referred name. By completing these three, the user can define the name in the form of 'XXXX-XXSX-XXXX' (i.e. STAT101-20S2-test), yet year-term and referred name are optional.

Cluster name and datetime must be provided to set up the event in the database correctly. Default time offset is zero minutes, but the user can provide the desired offset time for the event. If the event involves to labs, the user has the option to set labs closed by providing offset time onto labs.

The user needs to provide machine group(s) to locate where the event will happen. They can add or delete multiple machine groups for one event. Otherwise, the user can edit the machine group afterwards.

Errors checking is implemented for required information (exclude year-term and referred name) for the event. If the user does not provide the needed information, a pop-up window will appear. This window pops up after the event preview, which is implemented for the user to check if all information is correct before creating the event. And we also assume that the user can create event that has location and time same to exist events. That is due to consideration of some test may be held in same room and at same time.

Event Search

The search engine is got the idea from a template from Colorlib[2] with some modifications to fit our team's likeness.

There are two ways for the user to search events information, either its event name or the range of date. With the event name search, we have implemented an autocomplete function with the AJAX method to retrieve all the event names from the database (connect to PHP API within the source code) and limited by typing at least three symbols to get a dropdown of matching events. Unfortunately, the list of results only shows up if the name of the event is exact; otherwise, there would be no result found. Hence, the user is highly recommended to choose the event name from the dropdown. With the date range search, we have used an online daterangepicker element for getting help building a date range interface.

Searching an event can have a result with a lot of events (especially with a broad date range) which might cause slow speed due to the massive data workload on the webpage. Therefore, we use pagination to divide into ten results per page. Also, the result represents a table with the purpose of providing a clear vision for the user.

Extra Features

1. User Privileges

Our team came up with an idea of implementing multiple user privileges to boost database security. We have narrowed down to 3 levels; level 0 is an Exam Officer, level 1 is a course coordinator, and level 2 is a tutor.

All users should be able to view defined events of any subject (even if they are not in charge). This decision was to help the course coordinator to be able to see a conflict event when creating a new one, and tutors can plan their schedule.

The exam officers can create/edit new event of all subjects whereas the course coordinator can only create/edit event of their course. The tutor should not be able to modify/create new events to maintain data integrity. Deleting an event via tServer interface is not permitted, so they must contact the database owner to resolve the problem.

2. Editing an event

As explained above, only exam officers and course co-ordinator can edit an event permitted by their privileges. From the search page, they need to find the event they wish to change and click on `edit` button of the corresponding record. If unable to edit a record, the edit button will be disabled in the interface.

Editing an event is made simple for the user by using preloaded data to the pop-up modal as a preview, resulting to any changes are updated as soon as the user click on `save changes` button, no preview window as in creating the event.

3. User Profile

We also create a page dedicated to the user profile. In this page, it displays their name, their contact details, additional information and addresses. This page also allows the user to edit their information except for their name and user level. User profile at this time of development is unnecessary since more resource of the host server is needed.

Navigation/User Interaction (UI)

We believe it would be appropriate to create four main pages; the home page, create a new event page, profile page, and the search page for the server. All these pages except for search page are accessible through the left navigation bar. The search page is accessible through the search button on the top navigation bar. The reason we went in this route because it is the most appealing approach, the user can get started immediately, easy to navigate between pages, and the user will not be confused about the functionality and features.

Code Components and Libraries

For this project, we have integrated several open-source libraries such as PHP and AJAX for dynamic page content, acting as the middleman between the server and client machine. As for the front end, we have Bootstrap3/4 as a main responsive stylesheet for appealing PC and mobile view.

Reference List

[1] <https://colorlib.com/wp/template/login-form-v3/>

[2] <https://colorlib.com/etc/searchf/colorlib-search-14/>

Default avatar image: <https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTLuox6vatPBS6w8edvrLbqXzHimyKXOVejMQ&usqp=CAU>

Sidebar menu navigation template: <https://bootstrapious.com/tutorial/sidebar/index2.html>