COMP 1531 2018, S1 Group Project
Event Management System

Aim:

The aim of this group project is to enable students to consolidate their knowledge in the fundamental principles of Software Engineering and apply the theoretical concepts to a "hands-on" software engineering problem. The project will enable students to:

- Develop problem-solving skills to solve 'real-world' software engineering problems; analyse the problem domain, design and develop a solution to the problem
- Learn to work effectively as part of a team by managing your project, planning, and allocation of responsibilities among the members of your team
- Gain experience in collaborating through the use of a source control
- Apply appropriate design practices and methodologies in the development of their solution

Background:

Your consultancy firm has been called in to the School of Computer Science and Engineering for a new project. The staff and students at UNSW frequently hold courses and seminars. These events are attended by both staff and students at the university. Currently, the trainer of each event is responsible for advertising the event and handling registrations of trainees. The university is interested in making this process more efficient and would like to develop an online event management system (EMS) that streamlines the process of planning an event and registering attendees to the event. The university feels that such a system would help to market and increase the likelihood of attracting more attendees.

The centralised event management system (EMS) should enable course convenors to post different kinds of events and also enable interested attendees to register online for these events. The system will also allow attendees registered for a particular event to de-register. The system will be accessible by all UNSW staff and students. Following an initial discussion with the client, your team has elicited the following preliminary requirements for the event management system.

Preliminary Requirements from UNSW

The events organised in the university can be grouped broadly into two categories namely courses and seminars. Each event is scheduled at a particular venue and can run for a single day or over a period of days. Each event has a maximum attendee capacity. Each scheduled event has a de-register window period up and until which a registered attendee is allowed to de-register from the event, e.g., a course 'photo-shop' can be cancelled up to 24 hours before the scheduled date'.

A course is delivered by a single presenter who will also post the event on the EMS. A seminar consists of multiple sessions and each session can be presented by a UNSW academic or a non-UNSW guest-speaker e.g., "7th Australasian Symposium on Big Data & Analytics" is a seminar which runs from 2nd of October, 2018 to 3rd of October, 2018 and consists of sessions such as 'Using Machine Learning to analyse financial data', 'Semantic ontologies for financial data analysis', 'Predictive Analytics'. A seminar is posted on the

EMS by a convenor, who may or may not be a speaker at one of the sessions. The system should provide different forms to handle the posting of the above two types of events.

The EMS is only accessible by members (staff and students) of UNSW. Users should be able to log into the system with their zID and password. Once a user has logged into the system, they should see the two categories of events and list of all the 'open events' under each category. Clicking on a particular event should provide more details about the event. A user can select a particular event and register for this event. Once the user has successfully registered, a confirmation is displayed to the user. A registered user can cancel their registration up and until the de-registration window specified for that particular event.

Only UNSW staff are allowed to post (i.e., convene) events. Both UNSW staff and student can register for any event provided the event capacity has not exceeded. However, a UNSW staff cannot register for an event for which they are the convenor. Both UNSW staff and students can register for an event. Every logged in user must also have a link to a dashboard. If the user is a student, then the dashboard should display all current and past events that they have registered for. If the user is a staff, then their dashboard will display (i) current and past events that they have registered for (ii) current and past events that they have posted (iii) any cancelled events. Additionally, when a convenor clicks on an event they have organised, they should be able to see a list of attendees for that event. An unauthenticated user cannot access the website.

At any point in time, an event can be 'open', 'closed' or 'cancelled'. Any event which is scheduled to run in the future or currently running is said to be 'open'. Once an event has been completed, the course convenor will change the status of the event to 'closed'. If an event is cancelled before the scheduled-date, then it is said to be 'cancelled'.

The customer has also advised your team that they are still unsure of all their requirements, but these are their preliminary requirements. They would refine their requirements after seeing an initial version of the software system. Keeping this in mind, your consultancy firm has decided that this project will be delivered adopting an agile software development methodology to give the team flexibility to be able to adapt to the customer changes.

Group Project Requirements

Teams

1. For this project, you will need to organise yourself into teams of 3 (no more than 3), and all the team members must be from the same lab session.
2. Nominate one person from your team as the team master. This person will be responsible for registering the team on GitHub and the other members of the team will join the team. Instructions to register your team have been uploaded to webcms3 under Assignment 2 folder. Please ensure that your team is registered by week 6.

Implementation Guidelines

1. Keeping mind that an Agile Software Development style has been chosen for this project, your team will be required to build and deliver the project in iterations. Each iteration will deliver a part of the requirements of the project during which the team members are expected to carry out all the SDLC

activities, namely analysis, design, coding and testing. At the end of the iteration, you (as a team) will demonstrate to your lab class the functionality implemented during that iteration cycle. Your team must bear in mind that project requirements may be subject to change and enhancements to functionalities may be made at the end of the iteration. You will need to carefully design the solution for your current iteration, such that the solution is extensible to accommodate these changes. Deliverables for each iteration will be outlined at the start of each iteration cycle.

2. For this iteration, no sophisticated authentication is required to be implemented. When the URL of the EMS system is specified, e.g., http://localhost:8080/ems, this should launch a login page that prompts the user to enter their zID and password.

3. For this iteration, any kind of persistence mechanism may be used (flat-files or in-memory objects)

4. This project must be designed using **object-oriented design** principles and implemented using **Python/Flask/Jinja2**. It is recommended that students build their front-end using HTML and CSS only. If students choose to use other CSS frameworks (e.g., Material CSS, Bootstrap etc.,) to build the UI for the application then it must be kindly noted that support will be offered by course staff only on the core technology stack namely HTML, CSS, Python, Flask and Jinja2.

5. To implement authentication, we strongly recommend that the "Flask Login" extension is used.

6. Use of All necessary artifacts for this project e.g., CSV file containing the zID's and passwords can be downloaded from Webcms3 under the Assignments section.

7. What has been provided to you is a problem statement; a set of high-level requirements from the customer. Your team must analyse the problem statement, and go through a process of eliciting more formal requirements to develop the final set of user-stories and acceptance tests e.g., one the requirements in the specification states – "Clicking on a particular event should provide more details about the event." What details need to be displayed has not been elaborated. This must be elicited as part of your requirements analysis.

8. The model solution demonstrated in the lectures was only a guide to demonstrate "one" possible way of implementing the customer's requirements. You are required to design your solution and in week 13, during your final demo, you will be judged on how well your team can "sell" your solution to the customer.

9. At the end of each deliverable (weeks 7,9,11 & 13) tutors will check the team's GitHub repositories to ensure that all members of the team have contributed equally to the project and appropriate branches are created by each team member.

Scheduled Deliverables

The planned dates for the different stages of the project deliverables are outlined below.

- Week 7 Lab Session: Present Iteration 1 Design Deliverables to tutors (20%)
- Week 9 Lab Session: Demonstration of iteration 1 (25%)

- Week 11 Lab Session: Mile-stone check (5%)
- Week 13 Lab Session: Final Group Project Demonstration (50%)

Group Project Iteration 1 Deliverables

1. Week 7 Lab Session:

Prior to your lab session, your team must schedule collaboration sessions, to have initial high-level visioning discussions during which you will brainstorm to identify epic stories from the customer requirements and their key features, break-down high-level user stories to smaller user-stories and detail each user story to identify key conditions of satisfaction, error-conditions etc. Based on the requirements analysis, produce a domain model as a UML class diagram. Your team is expected to produce:

1. High Level Epic Stories from the problem statement
2. Each epic story broken into user stories – Each user-story must define:
   - a unique status identifier (e.g., UC1),
   - a short description of the feature based on Role-Goal-Benefit template (Refer to the RGB model described in the lectures)
   - an estimate for the implementation of the user story in user story points (e.g., UC1 = 2 User story points, where each point = 2.5 hours)
   - priority of implementation
   - acceptance criteria for each user story (Refer to the 3 C's model described in the lectures)

3. UML class diagram that clearly shows:
   - all classes with attributes, methods and right access modifiers
   - relationships (inheritance, association, aggregation and composition). Cardinality must also be indicated

4. Log book that records:
   - date of regular, stand-up meetings
   - summary of decisions made in stand-up meetings, requirements elicited and key design decisions (hand-written user-stories, CRC cards etc.)
   - responsibilities allocated to each team member and tasks to be accomplished for the next meeting
   - progress of tasks using a velocity chart (a hand-drawing will suffice, no sophisticated tool needed), summary of decisions made in stand-up meetings
   - reflection if assigned tasks (decided from last meeting) have been achieved
   - any obstacles

Please note, the above artifacts will need to be submitted using GIVE by week 7, Friday 11:59 pm.
The submission guidelines for week 7 deliverables will be outlined over the next week.

2. Week 9 Lab Session:

During this lab session, you will demonstrate the first iteration of your working software. For this iteration, the customer has requested that they be able to see an instance of working software that meets all the requirements outlined in the problem statement. However, they have agreed that the functionality to

cancel events by trainers and viewing cancelled events on the trainer's dashboard is low priority and can be viewed in the next iteration. The task for your team is to select the necessary user-stories from the backlog of user-stories developed in week 7 and implement them to meet the customer's goals. The additional document deliverable required for this iteration will be outlined later.