

ICT239

End-of-Course Assessment – July Semester 2020

Web Application Development

INSTRUCTIONS TO STUDENTS:

1. This End-of-Course Assessment paper comprises **SEVEN (7)** pages (including the cover page).
2. You are to include the following particulars in your submission: Course Code, Title of the ECA, SUSS PI No., Your Name, and Submission Date.
3. Late submission will be subjected to the marks deduction scheme. Please refer to the Student Handbook for details.

IMPORTANT NOTE

ECA Submission Deadline: Sunday, 15 November 2020, 12 noon

Please Read This Information before You Start Working on your ECA

This ECA carries 70% of the course marks and is a compulsory component. It is to be done individually and not collaboratively with other students.

Submission

You are to submit the ECA assignment in exactly the same manner as your tutor-marked assignments (TMA), i.e. using Canvas. Submission in any other manner like hardcopy or any other means will not be accepted.

Electronic transmission is not immediate. It is possible that the network traffic may be particularly heavy on the cut-off date and connections to the system cannot be guaranteed. Hence, you are advised to submit your assignment the day before the cut-off date in order to make sure that the submission is accepted and in good time.

Once you have submitted your ECA assignment, the status is displayed on the computer screen. You will only receive a successful assignment submission message if you had applied for the e-mail notification option.

ECA Marks Deduction Scheme

Please note the following:

*a) Submission Cut-off Time – Unless otherwise advised, the cut-off time for ECA submission will be at **12:00 noon** on the day of the deadline. All submission timings will be based on the time recorded by Canvas.*

*b) Start Time for Deduction – Students are given a grace period of 12 hours. Hence calculation of late submissions of ECAs will begin at **00:00 hrs** the following day (this applies even if it is a holiday or weekend) after the deadline.*

*c) How the Scheme Works – From 00:00 hrs the following day after the deadline, **10 marks** will be deducted for each **24-hour block**. Submissions that are subject to more than 50 marks deduction will be assigned **zero mark**. For examples on how the scheme works, please refer to Section 5.2 Para 1.7.3 of the Student Handbook.*

Any extra files, missing appendices or corrections received after the cut-off date will also not be considered in the grading of your ECA assignment.

Plagiarism and Collusion

Plagiarism and collusion are forms of cheating and are not acceptable in any form of a student's work, including this ECA assignment. You can avoid plagiarism by giving appropriate references when you use some other people's ideas, words or pictures (including diagrams). Refer to the American Psychological Association (APA) Manual if you need reminding about quoting and referencing. You can avoid collusion by ensuring that your submission is based on your own individual effort.

The electronic submission of your ECA assignment will be screened through a plagiarism detecting software. For more information about plagiarism and cheating, you should refer to the Student Handbook. SUSS takes a tough stance against plagiarism and collusion. Serious cases will normally result in the student being referred to SUSS's Student Disciplinary Group. For other cases, significant marking penalties or expulsion from the course will be imposed.

Answer all questions. (Total 100 marks)

Question 1

The food services of a hospital play a critical role to assist patients' fast recovery by providing the right food at the right time. To better facilitate the food service process, heated food trolleys are used to maintain the portioned food trays at a desirable temperature. A web application is designed accordingly to record the trolley temperature every day.

Since the web application should only be hosted by hospital staff, the authentication is necessary. According to your understanding of the Dashboard realized in your TMA, re-use your TMA codes and update the Flask application to include proper user login functions.

- (a) **Apply** proper MVC structure and **employ** the Flask Framework to build necessary UI elements. Here, two pages are required as shown in Figure Q1(a) below:
- **Register page:** a new user can register with his/her email and set up a password. Once the 'Register' button is clicked, the user profile should be sent to the backend and stored in a proper data model;
 - **Login page:** an existing user can login to the system by key in his/her email and the corresponding password. Once the 'Login' button is clicked, the user profile should be checked via backend controllers. An 'Error Message' is required to remind if a user not exist.

The figure displays two side-by-side web application mockups for 'DCS Smart Trolley'. Each mockup has a header bar with the application name and logo. The left mockup is the 'Register' page, with a 'Login' tab and an active 'Register' tab. It contains two input fields labeled 'Enter your email:' and 'Enter your password:', followed by a blue button labeled 'REGISTER'. The right mockup is the 'Login' page, with an active 'Login' tab and a 'Register' tab. It also contains two input fields labeled 'Enter your email:' and 'Enter your password:', followed by a blue button labeled 'LOGIN'.

Figure Q1(a)

(15 marks)

- (b) For a successful login, the application will render the Dashboard page by default as shown in Figure Q1(b) below. As shown in the red circle, please enable **TWO (2)** UI element to support the logout request.

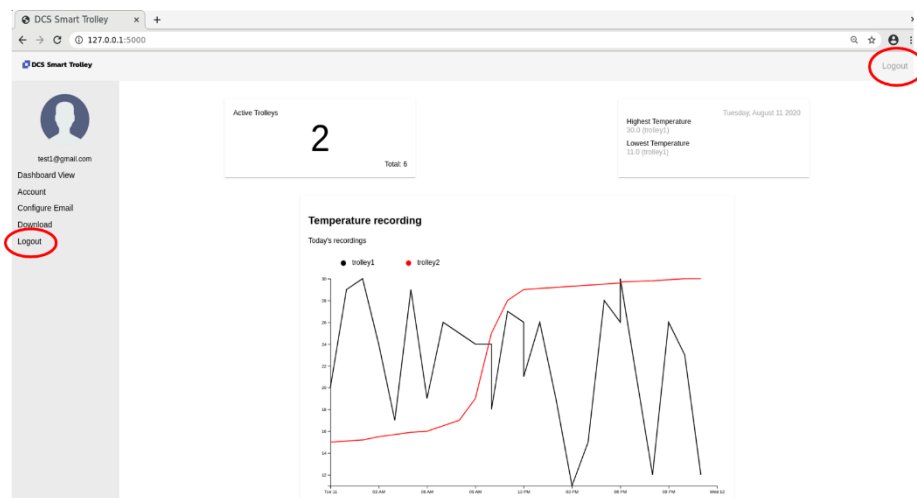


Figure Q1(b)

(12 marks)

- (c) Update the 'Account' tab in the side menu. Once the 'Account' tab is selected, display the following web page accordingly.
- Display a 'Change password' box and an 'email account' box;
 - Design a controller to assist the password changing requirements. Once the 'Change Password' button is clicked, the new password will be updated in the backend data model;
 - Once the password updating is complete, show a success message as circled in red.

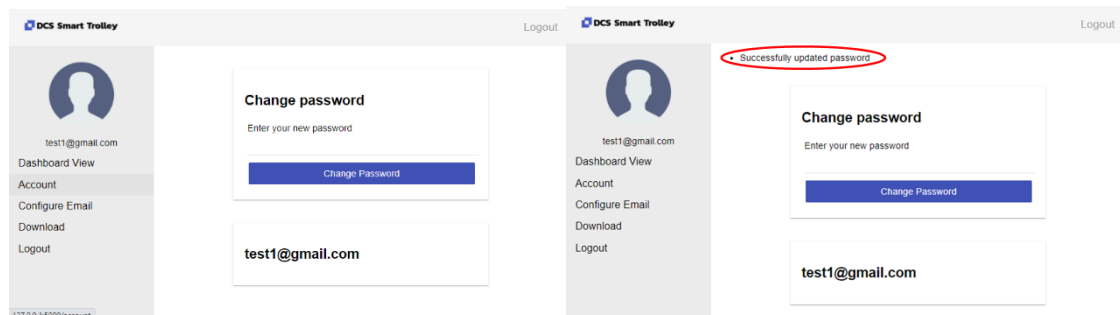


Figure Q1(c)

(15 marks)

Question 2

The designed web application targets on **analysing** the trolley temperature information. Thus, it should support the users to check and export the historical temperature recordings for any given date. Here, a download controller should be designed accordingly. Before download, please upload a demo dataset through the webpage created by your TMA Q3 code. The demo dataset is available in the following URL link: https://raw.githubusercontent.com/LiuFang00/ICT239/master/seed_demo_ECA.csv.

- (a) Please update the 'Download' tab in the side menu. Once the 'Download' tab is selected, display the following web page accordingly.
- Display a 'Download Data' box which allows a user to choose a date by either key in or from a calendar;
 - Create **TWO (2)** buttons ('Download' and 'Preview'). The buttons will trigger a popup window with reminder if no date is selected.

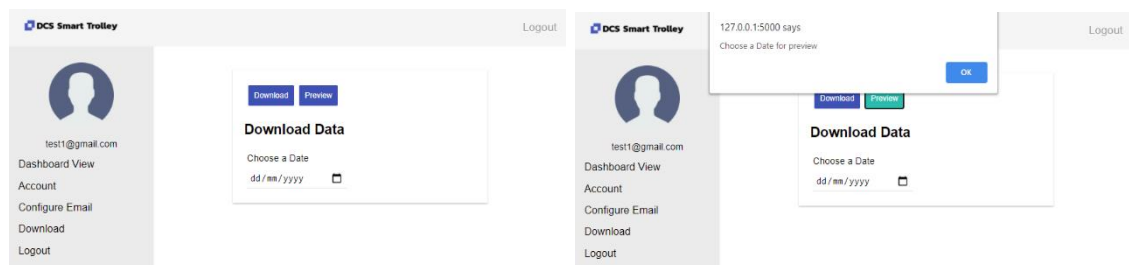


Figure Q2(a)

(10 marks)

- (b) Let the user continue working on the 'Download' page. Once the 'Download' button is clicked, a .csv file with temperature recordings on the selected date will be downloaded to the local computer. The sample format saved in the .csv file should like "trolley1,2020-06-30T22:19,24".

(15 marks)

- (c) Once the 'Preview' button is clicked with specified date, display the following webpage accordingly.
- Besides the 'Download Data' box, a chart is created to **demonstrate** the temperature recordings retrieved from the backend data models filtered by the selected date.
 - The 'Download' button is still available to download the previewed results to a .csv file.

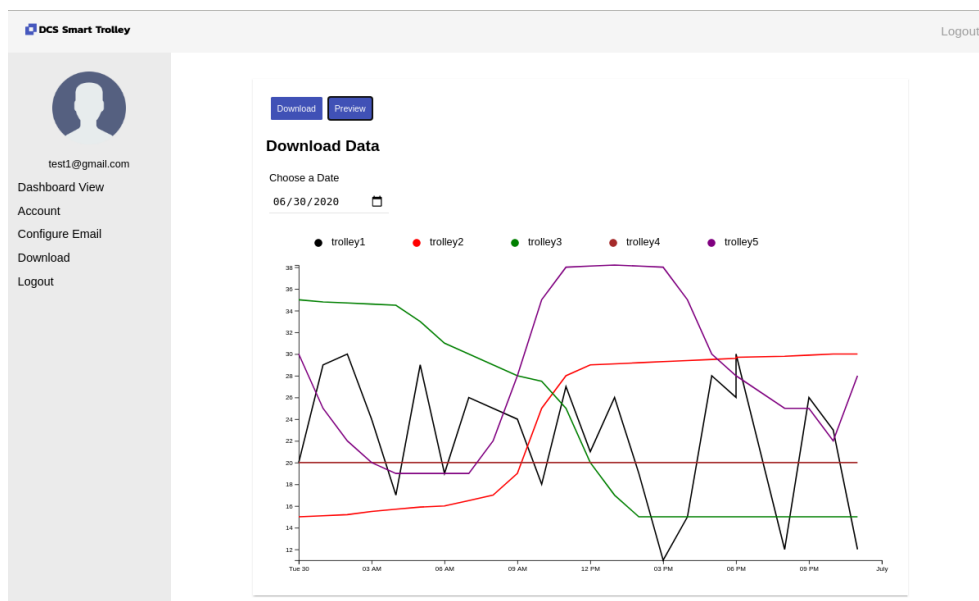


Figure Q2(c)

(13 marks)

Question 3

Suppose the hospital would like other departments to **analyse** the trolley temperature dataset jointly. Thus, an authorized user may want to share their data with other users by providing an email address. Please design a corresponding controller to realize this requirement in the web application.

- (a) Please update the 'Configure Email' tab in the side menu. Once the tab is selected, display the following web page accordingly.
- Display an 'Email' box which allows a user to input an email address;
 - The user is allowed to choose a date by either key in or from a calendar.

The screenshot shows a web application interface for 'DCS Smart Trolley'. On the left is a side menu with a user profile (test1@gmail.com) and navigation links: 'Dashboard View', 'Account', 'Configure Email' (highlighted), 'Download', and 'Logout'. The main content area displays an 'Email' configuration form. This form includes a 'Choose a Date' section with a text input (dd/mm/yyyy) and a calendar icon, followed by an 'Email Address' text input field. At the bottom of the form is a blue 'Send email' button. The top right of the page has a 'Logout' link, and the bottom left shows the URL '127.0.0.1:5000/email'.

Figure Q3(a)

(7 marks)

- (b) Once the user clicks the 'Send email' button, the historical temperature recordings should be retrieved from the backend data model with a specified date. Then, a .csv file will be formed and send to the corresponding email address.

(13 marks)

----- END OF ECA PAPER -----