# Assignment 1 - COIT20269 Mobile Web Apps

| | | |
|---|---|---|
| **Due date:** | 11:00pm AEST, Friday Week 6 | **ASSIGNMENT** |
| **Weighting:** | 20% | |
| **Length:** | Less than 50 MB | **1** |

## Objectives

This assessment item relates to the learning outcome 1, 2, 3 and 4. More specifically to design and implement a complex mobile application.

The objective of this assignment is for students to:

- Develop, test and maintain a mobile internet application using an integrated suite of mobile software development tools. More specifically on the client side *jQuery Mobile*, *JavaScript* and *HTML5* and *CSS* are used. On the server side we use *JavaScript* and the *node* server with various *node* packages. The *Mongo* database is used as a data store and that data is also stored locally on the mobile device.

## Introduction

You are assigned the task of creating a data logger to capture experimental data in a mobile application that stores data in a local database. The app has fields to record data for each of five days of use of drone aircraft. When a day is selected, a page is shown to record data values for drones that are recorded by the app. A drone entry consists of a **date** and **log** data. When the **Save Log Entry** button is pressed these values are saved locally in the devices' **localStorage**. When the **Show Log Entries** button is pressed a related page is shown that lists all the date/time and drone entries. More details of these pages will be given in the sections below.

We will refer to our app as **DroneLogs**. The specification of this app will be further refined in Assignment 2. This app is to be tested using the **Safari**, **FireFox** or **Chrome** browser and tested on an Android or iPhone mobile device.

## Client Side HTML / CSS / JavaScript Mobile Application

You are to implement this app using HTML5, CSS and JQueryMobile. The app is used by a company who has to collect data on who is using their drones over rolling 5 day periods. A number of individual drones exist and for each drone a unique ID number is used to identify it. For each individual drone the company needs to record data that can be used to document who flies the drone and from where the drone is flown.

The app should be based on a multi-page template structure. The client side application home page is shown in Fig. 1. The **Day 1, Day 2, Day 3, Day 4** and **Day 5** buttons should link to pages in a JQuery-mobile multi-page structure.

Images for the page/views required to implement the assignment are given in the following figures. Please note that the illustrations are for reference only, and your actual pages will be based on the CSS style-guide in use on your actual mobile device.

### Home page view

The home page view is shown in Fig. 1. This is the opening page for the app. A user clicks on a page link and is transferred to that page.

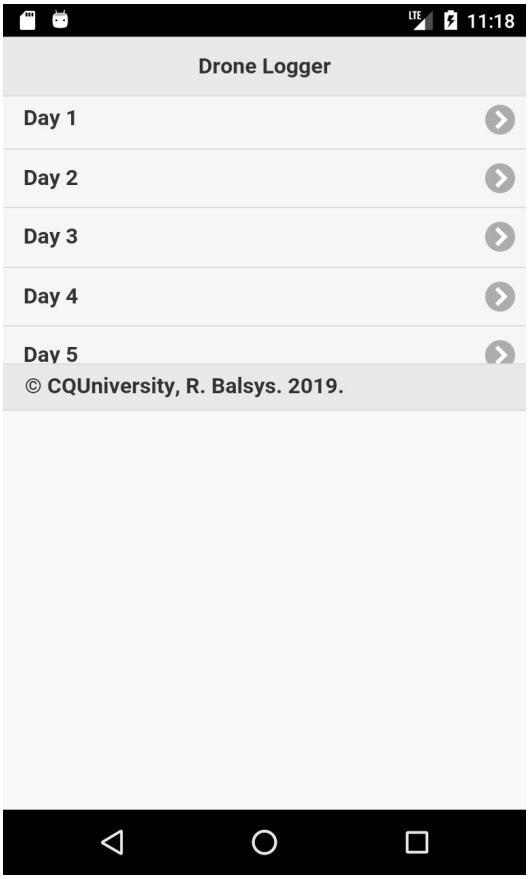



Fig 1: Home page view

## Drone view

The pages for entering the drone data are all the same so it makes sense to use one html page to show the basic structure and then use Javascript to customise the page based on what day we clicked on the home page. The view shown in Fig. 2 is for the **Day 1** page.

The header has titles from the variety sequence {Day 1, Day 2, Day 3, Day 4, Day 5} based on what button is pressed on the Home page. It also has a **Clear** and **Show logs** buttons visible in the header bar. The **Clear** button will clear the values in all the text entry fields on the page.

The footer has 3 buttons. The **Next** button will take you to the next element in the sequence Day 1, Day 2, Day 3, Day 4, Day 5}. The **Previous** button will take you to the previous element. The **Home** button will take you back to the home page.

The content of a drone page has text entry fields with labels, **Serial #:**, **Pilot:**, **Key:, Contract #:** and a drop down **Category** field (“”, “Civil”, “Government”, “Military”) as shown in Fig. 2.

When valid data is entered into all the text entry fields the **Save log entry** button can be pushed to save a log for the drone. If any of the data values are out of range or nothing is in a text entry then an alert should be shown to indicate the problem, as in Fig. 3.

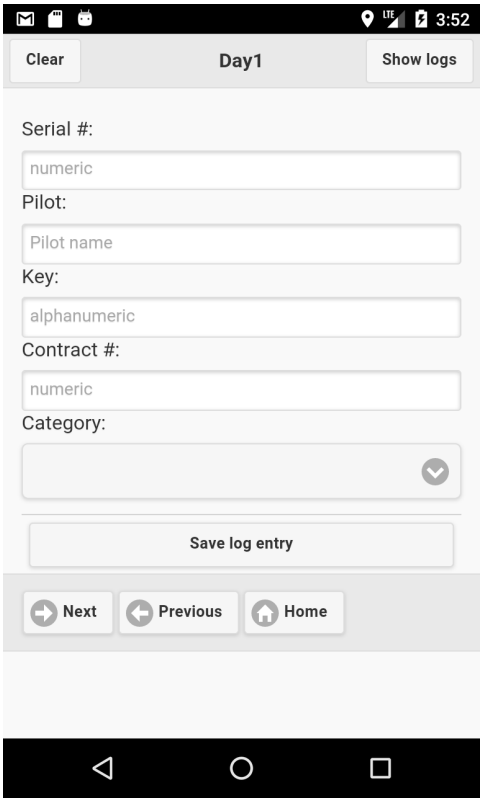



Fig 2: Drone page view

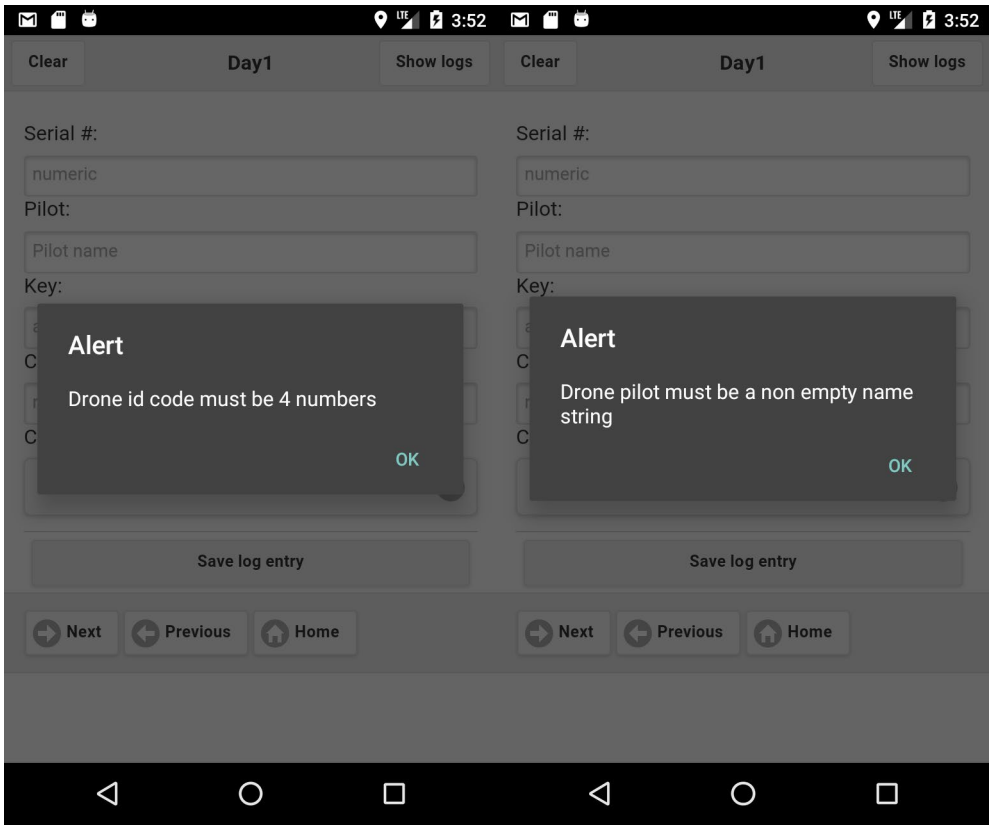


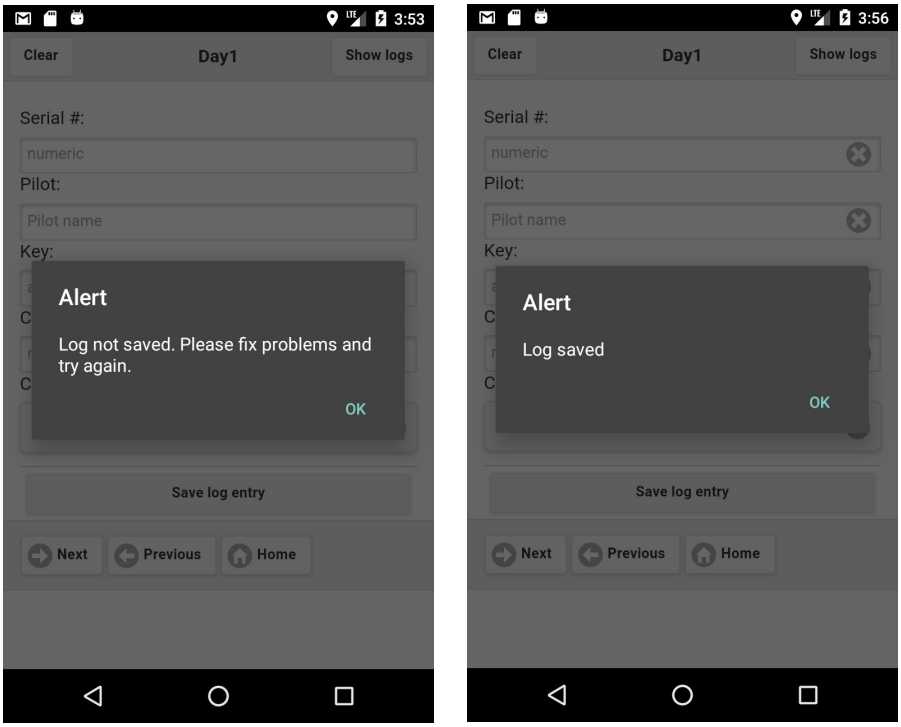

Fig 3. Sample error dialogs.

A second dialog must also be shown when there were problems saving the log (Fig. 4a), alternatively a success dialog should be shown (Fig 4b).

Any number of log entries can be made to record any number of drones being flown on a day in this manner.

## DroneLogs view

When the **Show logs** button in the drones's page header is pressed, the current date/time should be added to the data structure used to store the drone data values being recorded. Pressing this button also triggers a request for the location (latitude/longitude) to be added to the drone data. The log entries should then be saved in the device's **localStorage**. A drones logs page is then shown with all the saved logs for the drone listed, as shown in Fig. 5. Fig. 5 shows the display when more than 1 entry has been saved. Pressing the Back button on the footer should take you to the previous page.




(a) (b)

Fig 4. Save log completion dialog.

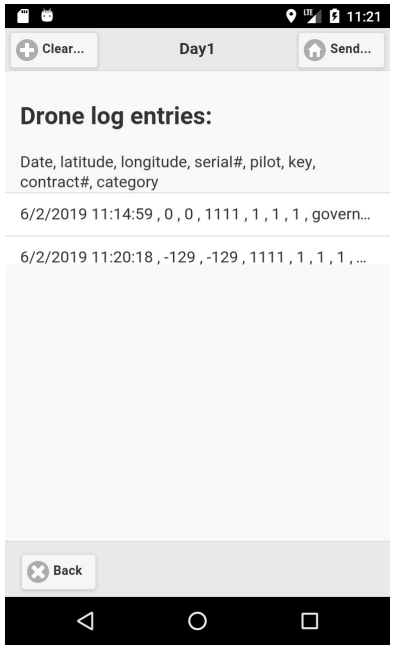



Fig. 5: Drone logs page.

### Send button

The **Send** button on the drone logs header bar is used to send the data over the WWW and clear the particular drone logs. This will be done in the second assignment. For now we will use this to clear all this drone data in the **localStorage**, and thus clear the existing drone logs. As this deletes data you must present a dialog box to the user with the choice to either send the logs or not, as shown in Fig. 6

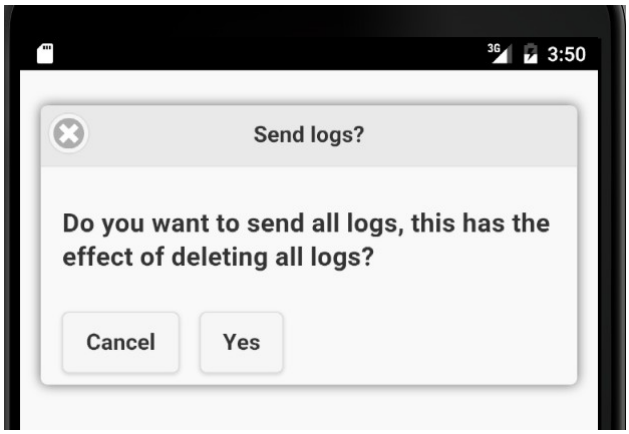



Fig. 6: Send logs confirmation dialog.

If the **Yes** button is pressed the **Send logs** dialog shown in Fig. 7 is presented and we return the drones view shown in Fig. 2. If the **No** button is pressed we just return to the drones view page shown in Fig. 2.

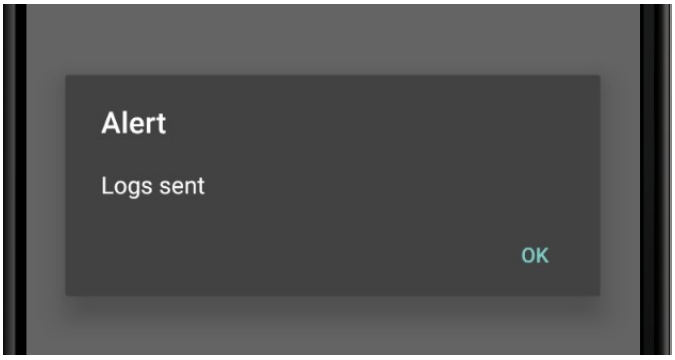



Fig. 7: Logs sent confirmation.

## Required Documentation

You are also to prepare a Word document. Your document should include an appropriate title page. Your document should have two sections that address the hardware and software requirements and application commentary given below.

## Hardware and Software Requirements

Your document should:

- Create a list of mobile devices targeted for the mobile web app. State what devices you tested the web app on.

## Application Commentary

Your document should:

- List the features you successfully implemented and those you were unable to successfully implement; you should describe the problem in a few sentences and also briefly describe anything you attempted to do to get it to work. Your approach to identifying and attempting to fix these bugs may gain you some partial credit for those features you were unable to implement.
- Include a description of any additional functionality you believe would be useful to add to this App. Explain what these new features are and how they would help to improve the web app.

## Submission

You are required to submit your assignment electronically via the Moodle course website. The deliverable is a rar or zipped directory containing all the code and resources needed for testing. You must tar, rar or zip together:

- The directory containing all the files, folders, images required to test your *mobile* application
- Your Word document

The resulting gzip, rar or zip file should be submitted on the course website. Please note that you should use your student number as the name for your gzip, rar or zip file when uploading to Moodle so that all students work can be linked back to the author.

# Assessment criteria – Ass1 - COIT20269 Mobile Web Apps

**Student name :**

**Student Number :**

| Assignment Component | Criteria | Marks | Total |
|---|---|---|---|
| **Client Side application** | | | |
| **DroneLogs.html / drone.css** | - The required scripts are all correct and available<br>- The multi-page logic is correct<br>- The home page displays and works as intended<br>- The navigation header/footers of the pages are correct and functions as expected<br>- The date, latitude, longitude, Serial #, Pilot, Key, Contract # and Category fields are displayed correctly and have the right hints<br>- The Show log page shows all the drones' data logs, and has the required functionality<br>- The date in the show logs page is formatted correctly | **7** | |
| **DroneLogs.js** | - Click on a day's button on home page takes you to the correct drone page<br>- Drone values range checked and all save log entries dialogs shown based on contents of the drones' fields<br>- All required dialogs are displayed<br>- The Clear button on the page header clears all fields<br>- Clicking on the Show logs button takes you to a new page where the current logs are all listed in required form<br>- The drones' page pageinit and pageshow methods are implemented correctly<br>- All navigation buttons have the required effect on the page view<br>- The drone data is saved in localStorage so when the application quits and restarts the values are retained (unless the Send logs button is used).<br>- The user interface of the drone app meets the guidelines given in the assignment | **9** | |
| **Hardware/Software & commentary** | | | |
| | **Hardware / Software requirements** | **1** | |
| | **Application commentary** | **2** | |
| **General** | | | |
| | - Feedback given as required<br>- Use appropriate naming conventions<br>- Adequate commenting<br>- Correct grammar<br>- Citation of references, copyright use | **1** | |
| **Penalties** | | | |
| | **Total** | **20** | |

**Lecturer Comments**

**Lecturer's Signature**_____ **Date:**