COMP1661 Logbook Upload Template
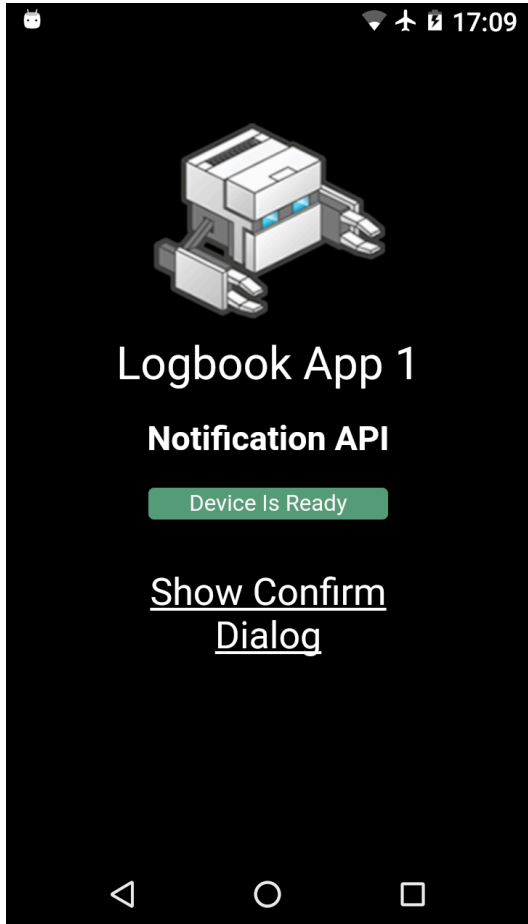
1.   Basic Information
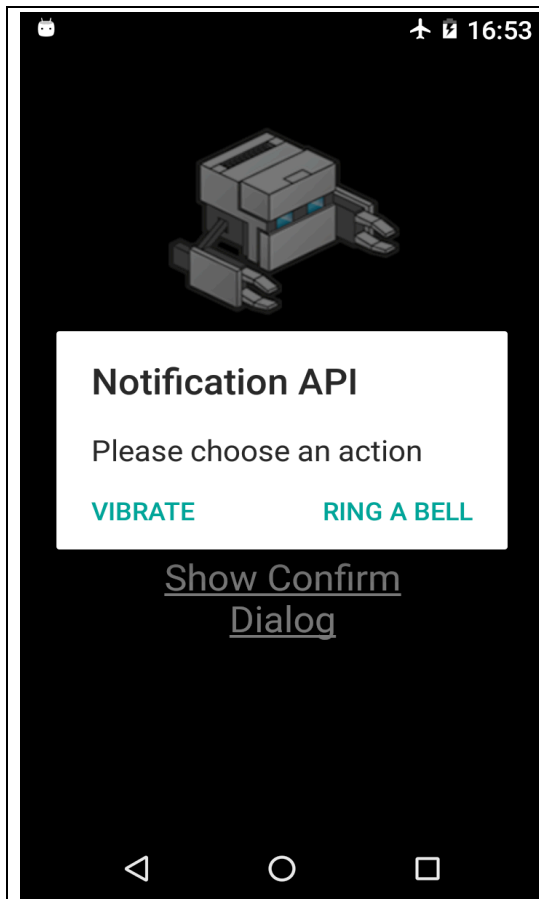
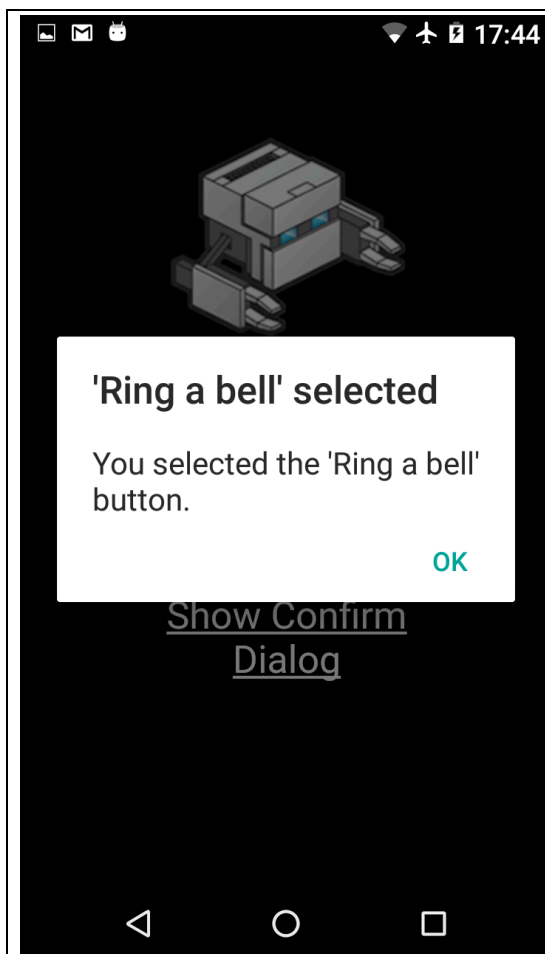| 1.1	Student name	**Myles Findlay**	
1.2	Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work.	**Name:** **Login id:**	
1.3	Which Exercise is this? Tick as appropriate.	1 - Create a PhoneGap App utilizing the Notification API	☒
		2 - Create a PhoneGap data entry screen RentalZ App	☐
		3 - Create an SQLite database for RentalZ App	☐
		4 - Create Android data entry screen for RentalZ App	☐
1.4	How well did you complete the exercise? Tick as appropriate.	• I tried but couldn't complete it	☐
		• I did it but I feel I should have done better	☐
		• I did everything that was asked	☐
		• I did more than was asked for	☒
1.5	Briefly explain your answer to question 1.4 e.g. what you could have done better?	• Changed the title of the dialog • Added custom message & custom button names to the dialog • Design of the app page could have been better	

2.   Exercise answer

2.1   Screenshots demonstrating what you achieved

Paste screenshots in here. Add explanation of what each screenshot demonstrates
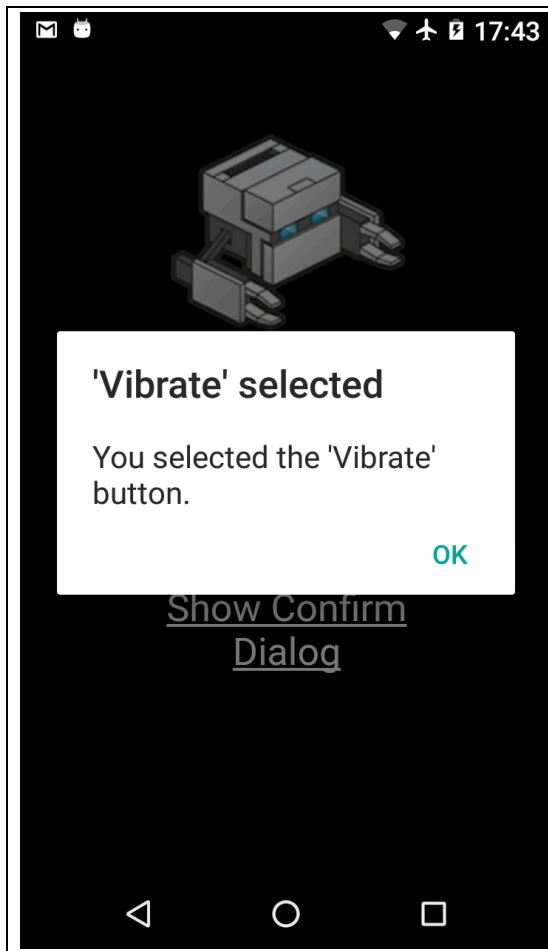
	This is the first screen/page that the user will see when the app is launched. A button (disguised as a link) is available near the bottom of the screen, which the user can click/tap to open the dialogue.

	When the user clicks/taps the 'Show Confirm Dialog' button, a confirm dialog will appear along with a message instructing the user on what they should do, and 2 buttons (provided as options) informing the users what actions can be performed by the phone. If the user selects the 'Vibrate' button, the dialog will disappear & the phone will vibrate for a certain amount of time. On the other hand, if the user selects the 'Ring a bell' button, the dialog will disappear & the phone will play the sound of a bell being rung.
	In addition, when the user selects the 'Ring a bell' button, a new dialog will appear letting them know that the 'Ring a bell' button was selected. An 'OK' button will also be shown to close the dialog.

In addition, when the user selects the 'Vibrate' button, a new dialog will appear letting them know that the 'Vibrate' button was selected. An 'OK' button will also be shown to close the dialog.

2.2 Code that you wrote

File name: index.html

```
<!DOCTYPE html> <!-- this declaration informs the web browser which
version of HTML the page was written in - in this case it is HTML5 -->
<html> <!-- informs the web browser that the document is a HTML one -->

<head> <!-- conatiner for all 'head elements'  -->
    <meta charset="utf-8" /> <!-- type of character encoding used for
this document i.e. uft-8  -->
    <meta name="format-detection" content="telephone=no" />  <!--
prevent phone numbers appearing as hyperlinks -->
    <meta name="msapplication-tap-highlight" content="no" />  <!--
prevent grey highlighting when objects are tapped -->
    <meta name="viewport" content="user-scalable=no, initial-scale=1,
maximum-scale=1, minimum-scale=1, width=device-width, height=device-
height" />  <!-- prevent user scaling/zooming (user-scalable=no),  100%
zoom level on page load (initial-scale=1), 100% zoom level maximum on
page load (maximum-scale=1), 100% zoom level minimum on page load
(minimum-scale=1), use maximum width and height of device (width=device-
width) and (height=device-height") respectively -->

    <!-- This is a wide open CSP declaration. To lock this down for
production, see below. -->
    <meta http-equiv="Content-Security-Policy" content="default-src *
```

```html
'unsafe-inline' gap:; style-src 'self' 'unsafe-inline'; media-src *" />
    <!-- Good default declaration:
    * gap: is required only on iOS (when using UIWebView) and is needed
for JS->native communication
    * https://ssl.gstatic.com is required only on Android and is needed
for TalkBack to function properly
    * Disables use of eval() and inline scripts in order to mitigate
risk of XSS vulnerabilities. To change this:
        * Enable inline JS: add 'unsafe-inline' to default-src
        * Enable eval(): add 'unsafe-eval' to default-src
    * Create your own at http://cspisawesome.com
    -->
    <!-- <meta http-equiv="Content-Security-Policy" content="default-src
'self' data: gap: 'unsafe-inline' https://ssl.gstatic.com; style-src
'self' 'unsafe-inline'; media-src *" /> -->


    <link rel="stylesheet" type="text/css" href="css/index.css" /> <!--
defines a link between a document and an external resource i.e. CSS file
-->
    <title>Logbook App 1</title> <!-- title of the document -->
</head>

<body> <!-- defines the document's body (containing all HTML document
content i.e. text, images, links, etc.) -->
    <div class="app"> <!-- defines a division/section with the HTML
document -->
        <h1>Logbook App 1</h1> <!-- defines a HTML heading - h1 is the
most important heading -->
        <h2>Notification API</h2> <!-- defines a HTML heading - h2 is
not as important as the h1 heading -->
        <div id="deviceready" class="blink">
            <p class="event listening">Connecting to Device</p>
            <p class="event received">Device Is Ready</p>
        </div>
        <br> <!-- new line -->
        <br> <!-- new line -->
        <p><a href="#" onclick="showConfirm(); return false;" data-
role="button">Show Confirm Dialog</a></p>
    </div>
    <script type="text/javascript" src="cordova.js"></script> <!-- start
of client-side script (JavaScript) linking to cordova.js file -->
    <script type="text/javascript" src="js/index.js"></script> <!--
start of client-side script (JavaScript) linking to index.js file
(within the js folder) -->
    <script type="text/javascript"> <!-- start of client-side script
(JavaScript) linking to index file (within the js folder) -->
        app.initialize();
    </script> <!-- end of client-side script (JavaScript) -->
    <script type="text/javascript">
        // variable consisting of array of names used for button names
        var options = ["Ring a bell", "Vibrate"];

        // function called when 'Show Confirm Dialog' button is clicked
```

```
        function showConfirm() {
            navigator.notification.confirm(
                //dialog message
                "Please choose an action",

                //dialog callback
                onDoConfirm,

                //dialog title, option names for buttons
                "Notification API", options
            );
        }

        // this callback will be performed when the user clicks a dialog
button
        // if user clicks first button, doorbell sound will be played;
otherwise the phone will virbate
        // better to use 'else if (btnNum == "2")' instead of 'else'
because if the latter is used,
        // the phone will still vibrate if the hardware back button is
tapped
        function onDoConfirm(btnNum) {
            if (btnNum == "1") {
                // ring the bell once when 'Ring a bell' button is
selected
                navigator.notification.beep(1);


                // alert to let user know the 'Ring a bell' button was
selected
                navigator.notification.alert(
                //dialog message
                "You selected the 'Ring a bell' button.",

                //no dialog callback required
                null,

                //dialog title, name of dialog button
                "'Ring a bell' selected", "OK"
                );
            } else if (btnNum == "2") {
                // 500 millisecond/0.5 second delay before phone
vibrates
                navigator.notification.vibrate(500);


                // alert to let user know the 'Vibrate' button was
selected
                navigator.notification.alert(
                //dialog message
                "You selected the 'Vibrate' button.",

                //no dialog callback required
                null,
```

```
                    //dialog title, name of dialog button
                    "'Vibrate' selected", "OK"
                    );
                }
            }
    </script> <!-- end of client-side script (JavaScript) -->
</body> <!-- end of document's body -->

</html> <!-- end of HTML document -->
```

File name: index.css

```
body {
    -webkit-touch-callout: none;                /* prevent callout to
copy image, etc when tap to hold */
    -webkit-text-size-adjust: none;             /* prevent webkit from
resizing text to fit */
    -webkit-user-select: none;                  /* prevent copy paste,
to allow, change 'none' to 'text' */
    background-color:#000;
    background-image:linear-gradient(top, #A7A7A7 0%, #E4E4E4 51%);
    background-image:-webkit-linear-gradient(top, #A7A7A7 0%, #E4E4E4
51%);
    background-image:-ms-linear-gradient(top, #A7A7A7 0%, #E4E4E4 51%);
    background-image:-webkit-gradient(
        linear,
        left top,
        left bottom,
        color-stop(0, #A7A7A7),
        color-stop(0.51, #E4E4E4)
    );
    background-attachment:fixed;
    font-size:12px;
    height:100%;
    margin:0px;
    padding:0px;
    width:100%;
    color: white;
}

/* Portrait layout (default) */
.app {
    background:url(../img/logo.png) no-repeat center top; /* 170px x
200px */
    position:absolute;              /* position in the center of the
screen */
```

```css
    left:50%;
    top:25%;
    height:50px;                        /* text area height */
    width:225px;                        /* text area width */
    text-align:center;
    padding:180px 0px 0px 0px;     /* image height is 200px (bottom 20px
are overlapped with text) */
    margin:-115px 0px 0px -112px;  /* offset vertical: half of image
height and text area height */
                                    /* offset horizontal: half of text
area width */
}

/* Landscape layout (with min-width) */
@media screen and (min-aspect-ratio: 1/1) and (min-width:400px) {
    .app {
        background-position:left center;
        padding:75px 0px 75px 170px;  /* padding-top + padding-bottom +
text area = image height */
        margin:-90px 0px 0px -198px;  /* offset vertical: half of image
height */
                                       /* offset horizontal: half of
image width and text area width */
    }
}

h1 {
    font-size:24px;
    font-weight:normal;
    margin:0px;
    overflow:visible;
    padding:0px;
    text-align:center;
}

p {
    font-size:20px;
    font-weight:normal;
    margin:0px;
    overflow:visible;
    padding:0px;
    text-align:center;
}

a {
    color: white;
}
```

```css
.event {
    border-radius:4px;
    -webkit-border-radius:4px;
    color:#FFFFFF;
    font-size:12px;
    margin:0px 30px;
    padding:2px 0px;
}

.event.listening {
    background-color:#333333;
    display:block;
}

.event.received {
    background-color:#4B946A;
    display:none;
}

@keyframes fade {
    from { opacity: 1.0; }
    50% { opacity: 0.4; }
    to { opacity: 1.0; }
}

@-webkit-keyframes fade {
    from { opacity: 1.0; }
    50% { opacity: 0.4; }
    to { opacity: 1.0; }
}

.blink {
    animation:fade 3000ms infinite;
    -webkit-animation:fade 3000ms infinite;
}
```