Vlad Predovic

CS 496

Due 11/11/2015

Assignment 4

http://web.engr.oregonstate.edu/~predoviv

For my application I decided to go a different route than the Windows 8 option. After researching for a while I decided I wanted to deliver a hybrid application. Hybrid applications primarily consist of web applications wrapped in order to be delivered like native applications. Additionally hybrid applications can interact with the hardware specific to different devices thanks to open source movements such as Apache Cordova.

I developed my application primarily using the Intel XDK IDE. I was drawn to this 'IDE' because it provides much of the drag and drop functionality similar to natural Android and Iphone environments but allows you to build an application that will work on both as it is built off Cordova.

My application is a timestudy app that interfaces with the API I built last week. Although many details of the implementation aren't where I want from a relational standpoint, the functionality is there.The phone app lets you see existing data for timestudies that are currently in place and it lets you add additional timestudies, procedures, and time study instances to the database. I will now describe the features of these three objects within the mobile application.

Timetudies:

For timestudies all that is required is the name of the study you want to add. This can be any combination of symbols. However, in order to add procedures to it you have to use the procedure keys. Adding anything but a set of numbers will result in error.

TimeStudy Instances:

For these you need to include the timestudy name, the name of the user conducting the study, and the iteration number. Failure to specify an iteration number or the name of an existing study will result in error. The system will not let you add an instance.

Procedures:

These require that you add the iteration number corresponding with the study instance and a name. Adding non-number units to the iteration number field will cause the system to reject the implementation.

All data is displayed using 'GET' requests. However, the data is changed using POST requests. This is modeled like the API assignment a few weeks ago.

I implemented geolocation because ultimately I would want my users to specify where the study was conducted. This information would be important if a company wanted to know where the studies were coming from, especially if they had factories in different parts of the world as the best results could then be compared and streamlined.

NOTE: I could not get the recording software to portray alerts in the video, but I would gladly show this in person if it is an issue.

```
$(document).on("click", ".uib_w_64", function(evt)
{
    /* your code goes here */
    var Tkeys = document.getElementById("studyKeys").value;
    var Tname = document.getElementById("studyName").value;
    if (isNaN(Tkeys))
    {
        alert("Must input numbers");
        return false;
    } else
    {
        intel.xdk.device.getRemoteData("http://proj2-1095.appspot.com/TimeStudies",
"POST", "name=" + Tname + "&procedures[]=" + Tkeys, "POSTsuccess_handler",
"POSTerror_handler");

        document.getElementById("studyResults").innerHTML = "TimeStudy: " + Tname +
"<br>was added succesfully with the following keys:" +

                                    "<br><br>" + Tkeys;

    }
});

/* button  Add a Procedure */
$(document).on("click", ".uib_w_68", function(evt)
{
    /* your code goes here */
    var Pname = document.getElementById("procName").value;
    var Piter = document.getElementById("procIter").value;
    if (isNaN(Piter))
    {
        alert("Must input numbers");
        return false;
    } else
    {
        intel.xdk.device.getRemoteData("http://proj2-1095.appspot.com/studyProcedures",
"POST", "name=" + Pname + "&iter=" + Piter, "POSTsuccess_handler", "POSTerror_handler");

        document.getElementById("studyResults").innerHTML = "Procedure: " + Pname +
"<br>was added succesfully running the following iteration:" +

                                    "<br><br>" + Piter;

    }
});

/* button  Start a Study */
$(document).on("click", ".uib_w_71", function(evt)
{
    /* your code goes here */
    var Iname = document.getElementById("instName").value;
    var Iiter = document.getElementById("instIter").value;
    var Iuser = document.getElementById("instUser").value;
    if (isNaN(Iiter))
    {
        alert("Must input numbers");
        return false;
    } else
```

```
    {
        intel.xdk.device.getRemoteData("http://proj2-1095.appspot.com/TSinstance",
"POST", "name=" + Iname + "&iter=" + Iiter + "&user=" +
                                        Iuser, "POSTsuccess_handler",
"POSTerror_handler");

        document.getElementById("studyResults").innerHTML = "The Instance: " + Iname +
"<br>was added succesfully running the following iteration:" + "<br><br>" +Iiter;

    }

});

}
```