

Internet Programming

Final Project

Issa Haddad – 47102

Firas Sardast – 47500

Abdulrahman Osoble – 49975

Osama Al-Madani - 44877

May 24, 2016

The purpose of our project was to propose an IoT system that provides a form of service to the public. Our service was to measure the foot fall of certain malls in Dubai.

* Dubai Mall
* Emirates Mall

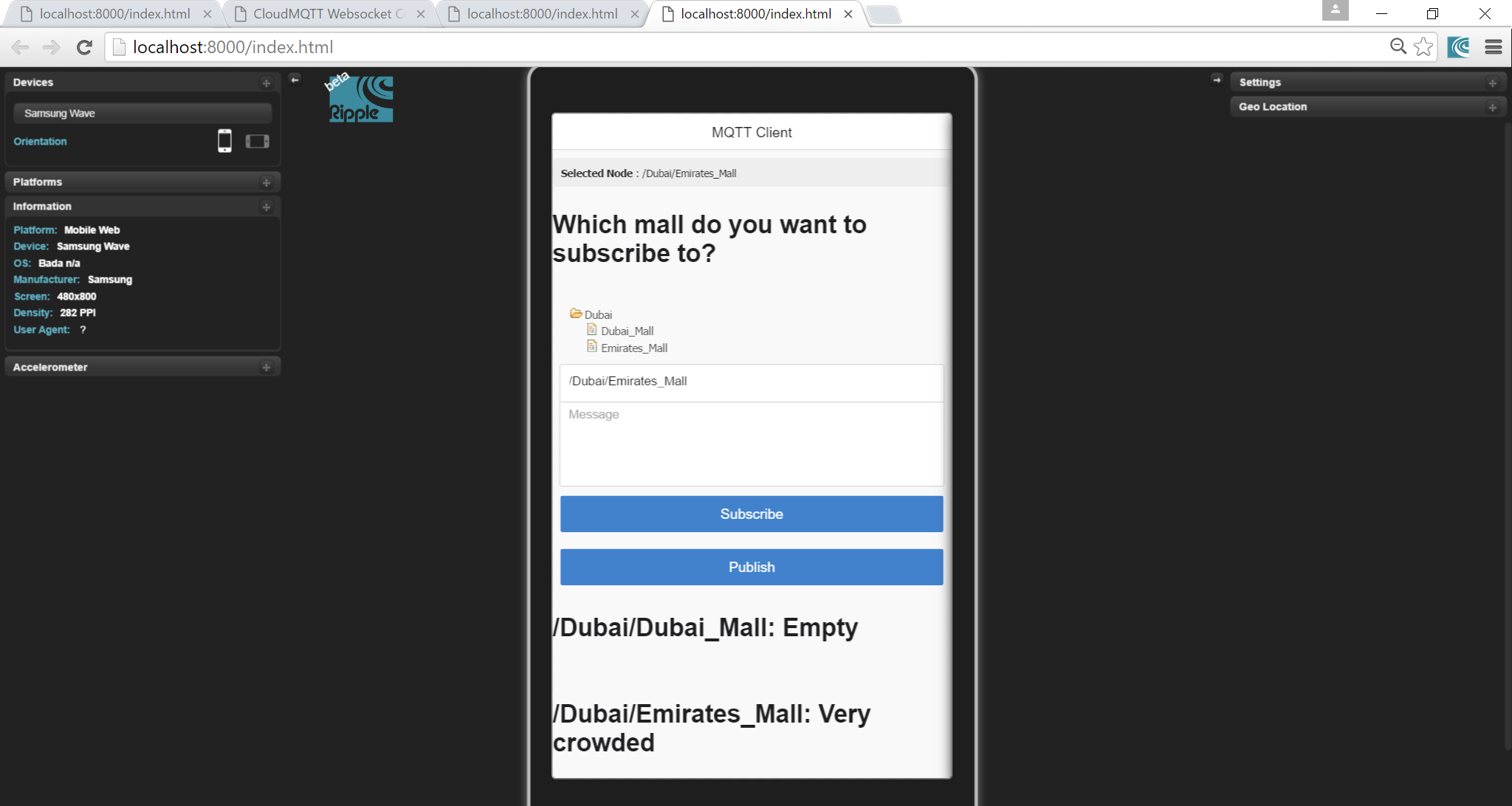
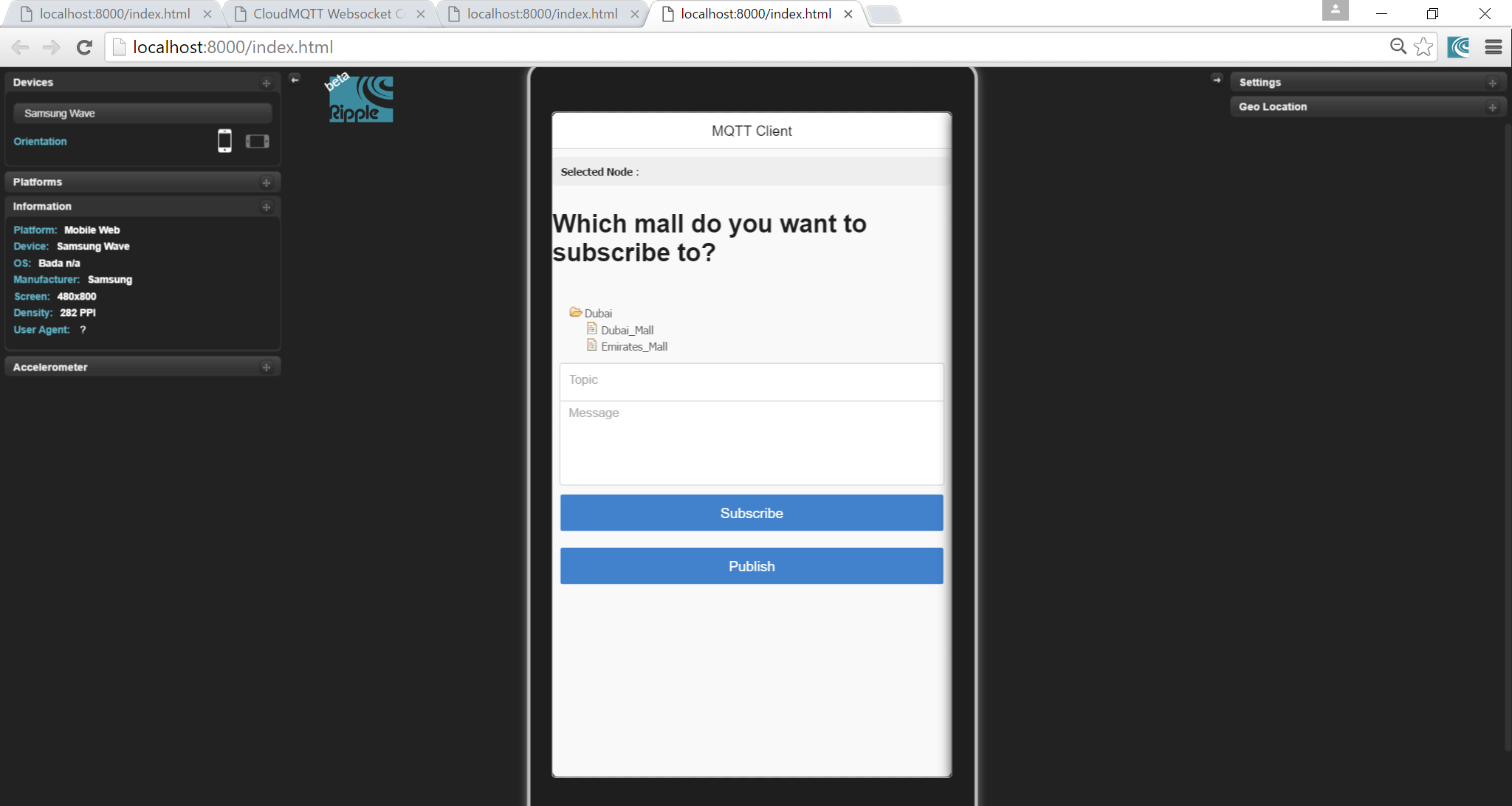
Our system first measures the amount of motion in a certain period of time. The data we get is processed to notify the clients of the crowd measure in a specific mall.

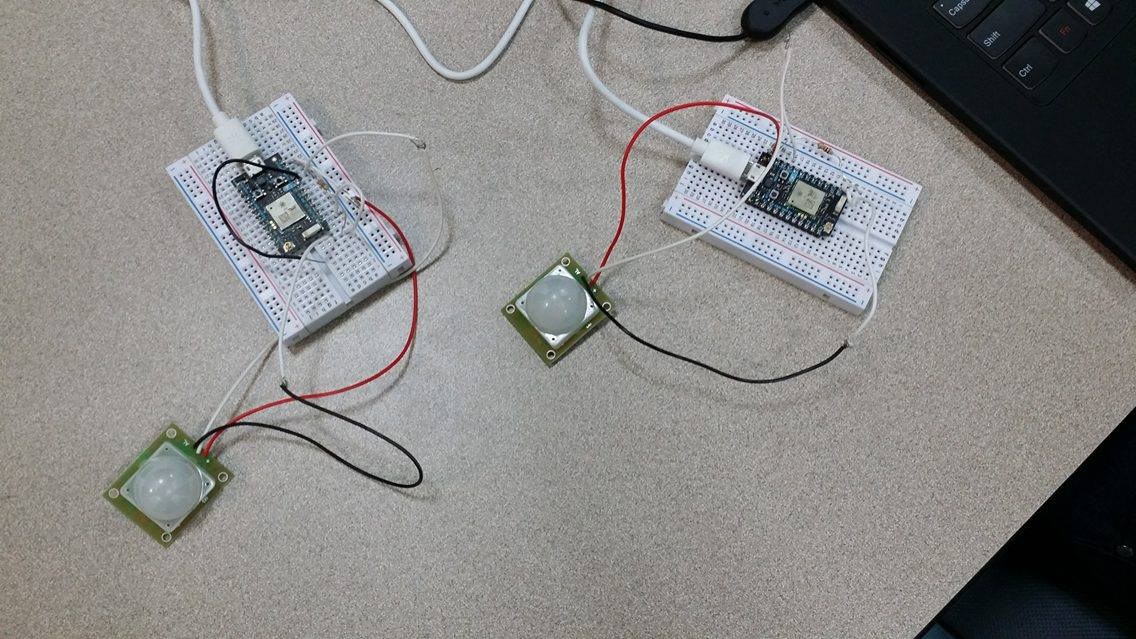
* Empty
* Little crowded
* Crowded
* Very crowded

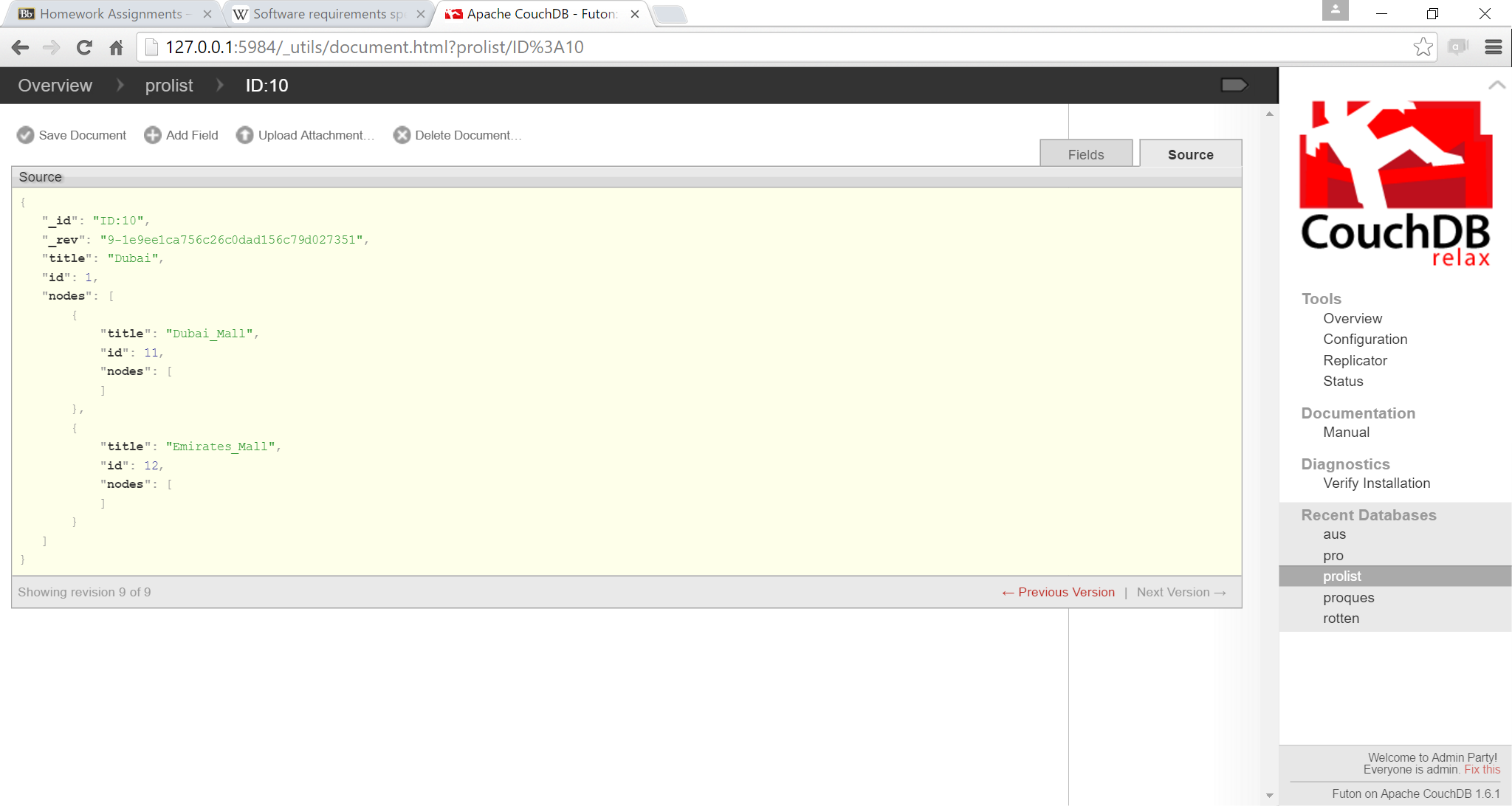
Next we send this data to an online cloud MQTT broker. Clients can then subscribe to their mall of choice from any platform. They will receive their data directly from the cloud broker, hence no localhost broker is being used. Clients can then pick the least crowded mall to shop in.

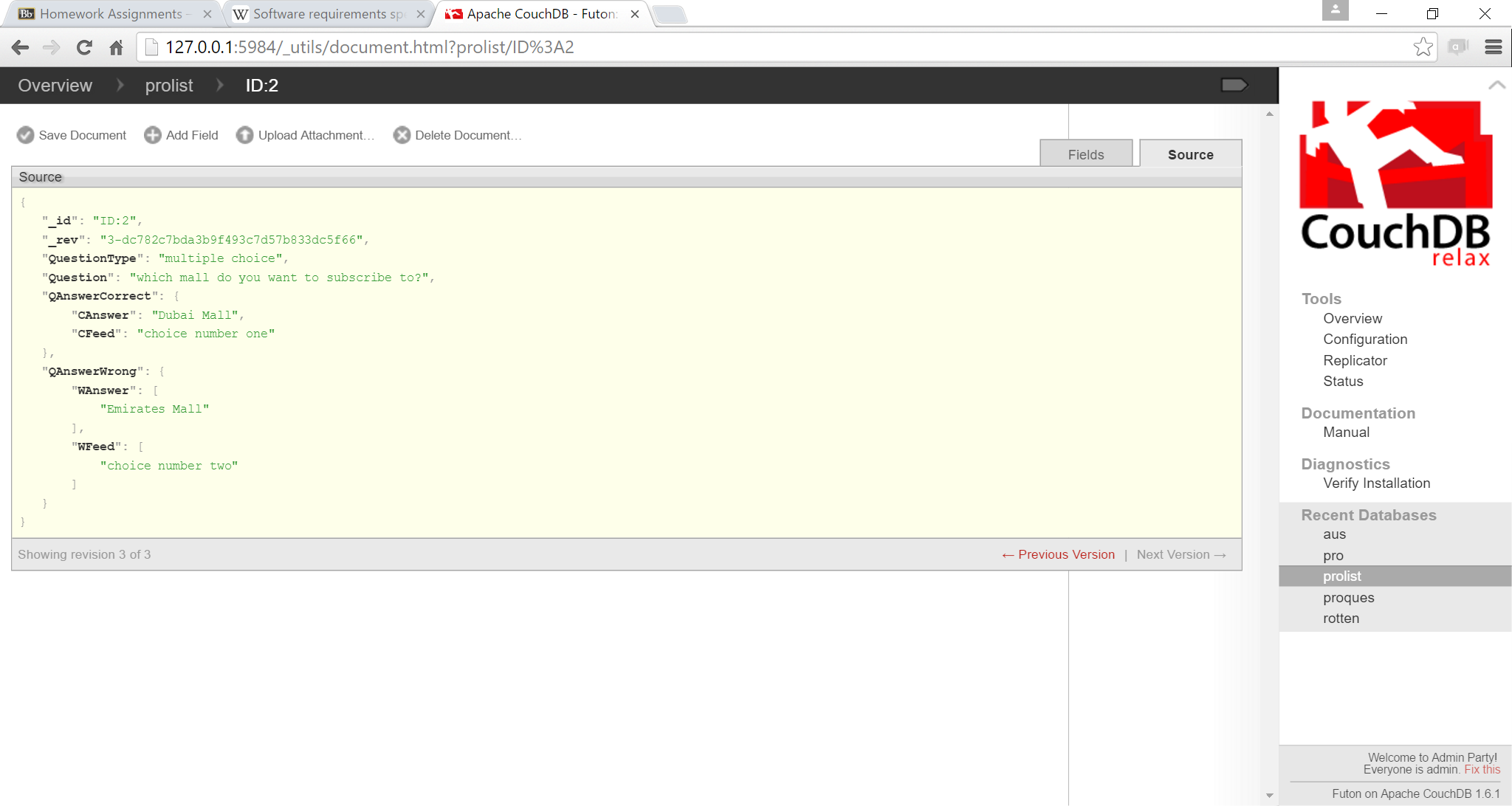
We used couch dB to store and retrieve questions related to the mall, that are displayed to the user on his/her platform. Cordova was our choice of framework, because it cross platform and its compatible with OnsenUI and Angularjs.

Furthermore, we used angular tree view to display the nodes because its dynamic and works well with any JSON file you feed. We have used nodeJS to store the data about the JSON file and the question onto couchDB. Retrieving data is done through Ajax









Code:

-------------------------index.html---------------------------------------------------------------------------------------------------

<!DOCTYPE html>

<!-- CSP support mode (required for Windows Universal apps): https://docs.angularjs.org/api/ng/directive/ngCsp -->

<html lang="en" ng-app="app" ng-csp>

<head>

<meta charset="utf-8" />

<meta name="apple-mobile-web-app-capable" content="yes" />

<meta name="mobile-web-app-capable" content="yes" />

<!-- JS dependencies (order matters!) -->

<script src="scripts/platformOverrides.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<script src="lib/angular/angular.min.js"></script>

<script src="lib/onsen/js/onsenui.js"></script>

<script src="lib/jquery.js"></script>

<!-- CSS dependencies -->

<link rel="stylesheet" href="lib/onsen/css/onsenui.css" />

<link rel="stylesheet" href="lib/onsen/css/onsen-css-components-blue-basic-theme.css" />

<link rel="stylesheet" href="lib/onsen/css/ex.css" />

<!-- CSP support mode (required for Windows Universal apps) -->

<link rel="stylesheet" href="lib/angular/angular-csp.css" />

<!-- --------------- App init --------------- -->

<script src="js/app.js"></script>

<script src="mqttws31.js"></script>

<script src="cordova.js"></script>

<script src="scripts/index.js"></script>

<script >

</script>

<title>My App</title>

<style>

.item {

padding: 10px;

line-height: 1;

}

.item-thum {

background-color: #ccc;

width: 50px;

height: 50px;

border-radius: 4px;

}

.item-title {

font-size: 15px;

font-weight: 500;

}

.item-desc {

font-size: 14px;

color: #666;

line-height: 1.3;

margin: 4px 0 0 0;

padding: 0 30px 0 0;

}

.item-label {

font-size: 12px;

color: #999;

float: right;

}

</style>

</head>

<body>

<ons-navigator var="navi">

<ons-page>

<ons-toolbar>

<div class="center">MQTT Client</div>

</ons-toolbar>

<div ng-controller="DetailController">

<div style="margin:10px 0 30px 0; padding:10px; background-color:#EEEEEE; border-radius:5px; font:12px Tahoma;">

<span><b>Selected Node</b> : {{parentone}}</span>

</div>

<h2>Which mall do you want to subscribe to?</h2><br>

<div

data-angular-treeview="true"

data-tree-model="roleList"

data-node-id="id"

data-node-label="title"

data-node-children="nodes"

ng-click="printParent($event);" >

</div>

<ons-list modifier="inset" style="margin-top: 10px">

<ons-list-item>

<input type="text" class="text-input text-input--transparent" placeholder="Topic" style="width: 100%" ng-model="topic">

</ons-list-item>

<ons-list-item>

<textarea class="textarea textarea--transparent" placeholder="Message" style="width: 100%; height: 100px;" ng-model="message">

</textarea>

</ons-list-item>

</ons-list>

<div style="padding: 10px 9px">

<ons-button modifier="large" style="margin: 0 auto;" ng-click="doSubscribe()">

Subscribe

</ons-button>

<br>

<ons-button modifier="large" style="margin: 0 auto;" ng-click="doPublish()">

Publish

</ons-button>

</div>

<h2>{{question1}}</h2><br>

<h2>{{question2}}</h2><br>

</div>

</ons-page>

</body>

</html>

--------------------------end of index.html ----------------------------------------------------------------------------------------

----------------------------app.js-------------------------------------------------------------------------------------------------------

(function(){

'use strict';

var module = angular.module('app', ['onsen','angularTreeview']);

module.controller('DetailController', function($scope, $mqtt) {

$.ajax({

type: "GET",

url: "http://localhost:5984/prolist/ID:10/?jsonp=callback",

dataType : 'jsonp',

async: false,

contentType: "application/json",

jsonpCallback: 'callback',

success : function(data){

console.log(data.title);

// assigning scope variables inside the user interface

$scope.$apply(function() {

$scope.roleList = [data];

});

}

});

$scope.printParent = function ($event) {

var root = $scope;

var currentScope = angular.element($event.target).scope();

var clicked = currentScope.node.title;

console.log('selected Node details: ', currentScope.node);

currentScope = currentScope.$parent;

console.log('parents::')

$scope.parentone ="";

while(currentScope.$id !== root.$id) {

console.log(currentScope.node);

$scope.parentone = currentScope.node.title + "/" + $scope.parentone ;

currentScope = currentScope.$parent;

}

$scope.parentone = "/" + $scope.parentone + clicked;

$scope.topic = $scope.parentone;

}

$scope.doSubscribe = function() {

console.log($scope.topic);

$mqtt.subscribe($scope.topic);

};

$scope.doPublish = function() {

console.log($scope.topic);

console.log($scope.message);

$mqtt.publish($scope.topic,$scope.message);

//$scope.question = $scope.message;

};

$scope.submit = function(){

var grade= "False";

if($scope.truefalse=="T"){

grade="Correct";

}

window.alert("your answer is: " + grade);

};

});

module.factory('$mqtt', function() {

var wsbroker = "m12.cloudmqtt.com"; //mqtt websocket enabled broker

var wsport = 37554 // port for above

var client = new Paho.MQTT.Client(wsbroker, wsport,

"myclientid\_" + parseInt(Math.random() \* 100, 10));

client.onConnectionLost = function (responseObject) {

console.log("connection lost: " + responseObject.errorMessage);

};

client.onMessageArrived = function (message) {

console.log(message.destinationName, ' -- ', message.payloadString);

var appElement = document.querySelector('[ng-app=app]');

var $scope = angular.element(appElement).scope();

$scope = $scope.$$childHead;

$scope.$apply(function() {

if (message.destinationName == "/Dubai/Dubai\_Mall")

$scope.question1 = message.destinationName + ": " + message.payloadString;

else

$scope.question2 = message.destinationName + ": " + message.payloadString;

});

};

var options = {

timeout: 3,

useSSL: true,

userName: "vjtovbxj",

password: "\_3au3PZ\_zfNK",

onSuccess: function () {

console.log("mqtt connected");

},

onFailure: function (message) {

console.log("Connection failed: " + message.errorMessage);

}

};

// will connect only once

client.connect(options);

return{

subscribe : function(topic){

client.subscribe(topic, {qos: 1});

},

publish: function(topic, msg) {

var message = new Paho.MQTT.Message

(msg);

message.destinationName = topic;

client.send(message);

}}

});

})();

(function(l){l.module("angularTreeview",[]).directive("treeModel",function($compile){return{restrict:"A",link:function(a,g,c){var e=c.treeModel,h=c.nodeLabel||"label",d=c.nodeChildren||"children",k='<ul><li data-ng-repeat="node in '+e+'"><i class="collapsed" data-ng-show="node.'+d+'.length && node.collapsed" data-ng-click="selectNodeHead(node, $event)"></i><i class="expanded" data-ng-show="node.'+d+'.length && !node.collapsed" data-ng-click="selectNodeHead(node, $event)"></i><i class="normal" data-ng-hide="node.'+

d+'.length"></i> <span data-ng-class="node.selected" data-ng-click="selectNodeLabel(node, $event)">{{node.'+h+'}}</span><div data-ng-hide="node.collapsed" data-tree-model="node.'+d+'" data-node-id='+(c.nodeId||"id")+" data-node-label="+h+" data-node-children="+d+"></div></li></ul>";e&&e.length&&(c.angularTreeview?(a.$watch(e,function(m,b){g.empty().html($compile(k)(a))},!1),a.selectNodeHead=a.selectNodeHead||function(a,b){b.stopPropagation&&b.stopPropagation();b.preventDefault&&b.preventDefault();b.cancelBubble=

!0;b.returnValue=!1;a.collapsed=!a.collapsed},a.selectNodeLabel=a.selectNodeLabel||function(c,b){b.stopPropagation&&b.stopPropagation();b.preventDefault&&b.preventDefault();b.cancelBubble=!0;b.returnValue=!1;a.currentNode&&a.currentNode.selected&&(a.currentNode.selected=void 0);c.selected="selected";a.currentNode=c}):g.html($compile(k)(a)))}}})})(angular);

------------------------------- end of app.js -----------------------------------------------------------------------------------------

---------------------finalpro.js--------------------------------------------------------------------------------------------------------

var fs = require("fs");

// handle of database couch\_db

var nano = require('nano')('http://localhost:5984');

var test\_db = nano.db.use('prolist');

var MultiQ;

fs.readFile(\_\_dirname+"/projectQ.txt", function(error, data) {

console.log();

var file = data.toString();

file=file.split("\r\n");

// counter to keep track of different questions

var counter = 0;

// these will be used to store all data of questions, ex: Qtype[0] has questions[0] --> answers[0]

var newlnLoc=[];

var Qtype=[];

var questions=[];

var answers=[];

// Travers through all the lines in the text file

for(i = 0; i < file.length; i++)

{

// get in if it is a new line to update counter or question number

if (file[i].match("^\s\*$"))

{

newlnLoc[counter]=i;

counter +=1;

}

// get in if line is for question type and update Qtype[]

if(file[i].indexOf("// ") > -1)

{

var Qt = file[i].substring(3);

Qtype[counter]=Qt;

}

// get in if line is for question and update questions[]

if(file[i].indexOf("::") > -1)

{

// in fill-in-the-blank, the answer is within the question, we need to remove that

if (Qtype[counter]=="fill-in-the-blank")

{

var ind = file[i].lastIndexOf("::");

var beg = ind +3;

var blank = " \_\_\_\_\_\_\_ ";

OpenBr = file[i].indexOf("{");

CloseBr = file[i].indexOf("}");

// get in if the blank is in the beginning of the question

if (OpenBr == beg)

{

var str = CloseBr + 2;

var Qu = file[i].substring(str);

var Q = blank + Qu;

questions[counter]=Q;

}

// get in if the blank is not in the beginning of the question

else

{

var str = CloseBr + 2;

var Qu1 = file[i].substring(beg,OpenBr);

var Qu2 = file[i].substring(str);

var Q = Qu1 + blank + Qu2;

questions[counter]=Q;

}

}

else

{

var ind = file[i].lastIndexOf("::");

var beg = ind +3;

var Q = file[i].substring(beg);

questions[counter]=Q;

}

}

// get in if line has answer and update answer[]

if(file[i].indexOf("{") > -1)

{

var OpenBr = file[i].indexOf("{");

var CloseBr = file[i].indexOf("}");

var start = OpenBr + 1;

var A = file[i].substring(start,CloseBr);

answers[counter]=A;

}

}

// get the number of questions which is number of empty lines + 1

Qnum = counter+1;

var ID=1;

// handle each question accordingly

for(i=0;i<Qnum;i++)

{

if(Qtype[i]=="true/false")

{

var Questiontype = Qtype[i];

var Quest = questions[i];

var Ans = answers[i];

// create the json file for true of false questions

var data = {

QuestionType: Questiontype,

Question: Quest,

QAnswer: Ans

};

// insert json file into database

test\_db.insert(data,'ID:'+ID, function(err, body){

if(!err)

{

console.log("document Added");

}

else

{

console.log(err);

}

});

ID +=1;

}

if(Qtype[i]=="multiple choice")

{

var Questiontype = Qtype[i];

var Quest = questions[i];

var MultiQ = questions[i];

var numWr = (answers[i].match(/~/g) || []).length;

var equone = answers[i].indexOf("=");

var eq = equone + 1;

var hashone = answers[i].indexOf("#");

var ha1 = hashone - 1;

var ha2 = hashone + 2;

var tildeone = answers[i].indexOf("~");

var ti1 = tildeone - 1;

var ti2 = tildeone + 1;

var ansone = answers[i].substring(eq,ha1);

var ansoneFeed = answers[i].substring(ha2,ti1);

var wr = answers[i].substring(ti2);

var res = wr.split("~");

var Wrong = [];

var WrongFeed = [];

for(i=0;i<numWr;i++)

{

var hash = res[i].indexOf("#");

var hashm = hash - 1;

var hashp = hash + 2;

Wrong[i] = res[i].substring(0,hashm);

WrongFeed[i] = res[i].substring(hashp);

}

// create the json file for multiple choice questions

var data = {

QuestionType: Questiontype,

Question: Quest,

QAnswerCorrect: {CAnswer:ansone,CFeed:ansoneFeed},

QAnswerWrong: {WAnswer:Wrong, WFeed:WrongFeed}

};

// insert json file into database

test\_db.insert(data,'ID:'+ID, function(err, body){

if(!err)

{

console.log("document Added");

}

else

{

console.log(err);

}

});

ID +=1;

}

if(Qtype[i]=="fill-in-the-blank")

{

var Questiontype = Qtype[i];

var Quest = questions[i];

var numAns = (answers[i].match(/=/g) || []).length;

var equone = answers[i].indexOf("=");

var eq = equone + 1;

var wr = answers[i].substring(eq);

var res = wr.split(" =");

var Ans = [];

for(i=0;i<numAns;i++)

{

Ans[i] = res[i];

};

// create the json file for fill-in-the-blank questions

var data = {

QuestionType: Questiontype,

Question: Quest,

QAnswer: Ans

};

// insert json file into database

test\_db.insert(data,'ID:'+ID, function(err, body){

if(!err)

{

console.log("document Added");

}

else

{

console.log(err);

}

});

ID +=1;

}

if(Qtype[i]=="essay")

{

var Questiontype = Qtype[i];

var Quest = questions[i];

var Ans = "";

// create the json file for essay questions

var data = {

QuestionType: Questiontype,

Question: Quest,

QAnswer: Ans

};

// insert json file into database

test\_db.insert(data,'ID:'+ID, function(err, body){

if(!err)

{

console.log("document Added");

}

else

{

console.log(err);

}

});

ID +=1;

}

}

});

data2 = { "title" : "Dubai", "id" : 1, "nodes" : [{ "title" : "Dubai\_Mall", "id" : 11, "nodes" : []},{ "title" : "Emirates\_Mall", "id" : 12, "nodes" : []}]};

test\_db.insert(data2,'ID:'+10, function(err, body)

{

if(!err)

{

console.log("document Added");

}

else

{

console.log(err);

}

});

---------------------------------end of code------------------------------------------------------------------------------------------