QUESTION 1 (199/200 Words)

A)

Additional Functional Requirements:

|  |  |  |  |
| --- | --- | --- | --- |
| **FR** | **NAME** | **DESCRIPTON** | **OBSTACLE** |
| FR4 | Payment Integration | The app should allow users to make payments for their taxi-sharing requests or offers within the app.- ✓- | Integrating a secure and reliable payment gateway that supports various payment methods and complies with data privacy and security regulations. /\*\* M. Comment – You describe a state of the FR in action. The need is to say what would stop, make it difficult, to meet the needs and expectations of its intended use, so what is incapacitating the FR from working \*\*/ |
| FR5 | Rating and Review Features | The app should allow users to rate and review their taxi-sharing experience with other users.- ✓- | Implementing a robust rating and review system that can effectively handle user-generated content and maintain the integrity of the feedback*. /\*\* M. Comment a*s above*\*\*/* |
| FR6 | Push Notifications | The app should send notifications to users for important updates, such as when a taxi-sharing request is accepted.- ✓- | Ensuring that the notification system is efficient, reliable, and can handle a large number of notifications in real-time. |
| FR7 | Camera Functionality | The app should allow users to take photos for identification purposes, such as for profile pictures.- ✓- | Ensuring that images are stored securely, while also considering challenges related to privacy and data security. |
| FR8 | User Verification | The app should have a user verification process to ensure the authenticity and trustworthiness of users.- ✓- | Implementing a secure and reliable user verification process, while also considering potential challenges related to user privacy, data security, and fraud prevention*. /\*\* M. Comment s*ecurity and privacy are usually considered as a given*\*\*/* |

|  |
| --- |
| Comments Some of the FR descriptions are ambiguous |
| Note 1: Some of your OB have been very generic general – FR and OB needs to be quite specific in the remit of its function or non-Function against which ambiguity can be removed.  Note 2: You describe what needs to be done rather than an obstacle. Perhaps, it needs to be reworded to be an obstacle.  Needed a caption ----   |  |  |  |  | | --- | --- | --- | --- | | FR | Name | Description | Obstacle | | Note: all tables – screenshots etc.… always need to have a label, Caption | the set of functional requirements, they need to be distinctive |  | The obstacles must be operational (runtime) conditions rather than design time conditions |   Table 1: Additional functional requirements and obstacles for a taxi-sharing app. |
| *The aim of Q1a is to elicit FRs that are implementable and to ensure that the developer is aware of their possible failures, something that might stop it from working. The obstacles are something that the developer will have to consider. The TMA03 brief explains, "...a functional requirement is a set of tasks we want the app to carry out for the user, and an obstacle is something we can think of that might stop it from working" because of the various technical and non-technical conditions that the users are in. Consequently, the app is unable to carry out a task for the user. Therefore. for the obstacles, we are looking for these technical and non-technical conditions that the users are in.* |
| Comments - obstacle for a functional requirement |
| Obstacles give a valuable perspective when defining functional requirements as it can help ensure that the system is designed to meet the needs and expectations of its intended users. |

B)

Screenshot showing app running on emulator

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidenceScreenshot showing app running in browser

Body of www/index.html below (I have kept VS code formatting for improved readability, although there are cases where code goes onto the next line due to the limited horizontal space). CSS was not a requirement for the question, but I have used some basic styling to allow me to fit everything on one page when running on the android emulator and have thus included it below for reference.- ✓-

HTML CODE:

<html>

  <head>

    <!--

  /\*\*\*\*\*\*\*\*\*\*\*

   \* OU TM352 Block 3, TMA03: index.html

   \*

   \* To function correctly this file must be placed in a Cordova project and the appopriate plugins installed.

   \* You need to complete the code which is commented with TODO.

   \*

   \* Released by Chris Thomson / Stephen Rice: Dec 2020

   \* Modified by Chris Thomson: March 2023

   \* Modified and submitted by (Luke Bowman)

   \*\*\*\*\*\*\*\*\*\*\*\*/

 --->

    <meta charset="utf-8" />

    <meta name="format-detection" content="telephone=no" />

    <meta name="msapplication-tap-highlight" content="no" />

    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width" />

    <script src="js/jquery-3.5.1.min.js"></script>

    <script src="https://js.api.here.com/v3/3.1/mapsjs-core.js" charset="utf-8"></script>

    <script src="https://js.api.here.com/v3/3.1/mapsjs-service.js" charset="utf-8"></script>

    <script src="https://js.api.here.com/v3/3.1/mapsjs-ui.js" type="text/javascript" charset="utf-8"></script>

    <link rel="stylesheet" type="text/css" href="https://js.api.here.com/v3/3.1/mapsjs-ui.css" />

    <link rel="stylesheet" type="text/css" href="css/style.css"/>

    <script src="https://js.api.here.com/v3/3.1/mapsjs-mapevents.js" type="text/javascript" charset="utf-8"></script>

    <title>Taxi Sharing</title>

  </head>

  <body>

    <h2>Taxi Sharing</h2>

    <div>

      <!--- TODO: add text and form entry methods to capture required data -->

      <form>

      <label for="userid">Please Enter your ID:<br><i>(click <b>Register</b> if you are a first-time user)</i><label><br>

      <input type="text" id="userid" name="userid">

            <button type="button" onclick="registerUser();">REGISTER</button>

      </form>

      <!-- Offer Ride form -->

      <div id="offer">

        <h3>Offer to share a ride details:</h3>

          <form>

            <label for="sdatetime">Date and Time:</label>

              <input type="datetime-local" id="sdatetime" name="sdatetime"><br>

              <label for="sduration">How long will you wait?: </label>

              <select id="sduration" name="sduration">

              <option value="0">Exact time only</option>

              <option value="1">1 Hour</option>

              <option value="2">2 Hours</option>

              <option value="3">3 Hours</option>

              </select><br>

              <label for="spickup">Pickup:</label>

              <select id="spickup" name="spickup">

              <option value="Open University Campus, Milton Keynes">Open University Campus, Milton Keynes</option>

              <option value="Milton Keynes Central Station">Milton Keynes Central Station</option>

              </select>

          <button type="button" onclick="offerTaxi()">OFFER</button>

        </form>

      </div>

      <!-- Request Ride form -->

      <div id="request">

        <h3>Request to share a ride details:</h3>

          <form>

            <label for="rdatetime">Date and Time:</label>

              <input type="datetime-local" id="rdatetime" name="rdatetime"><br>

              <label for="rpickup">Pickup:</label>

              <select id="rpickup" name="rpickup">

              <option value="Open University Campus, Milton Keynes">Open University Campus, Milton Keynes</option>

              <option value="Milton Keynes Central Station">Milton Keynes Central Station</option>

              </select>

              <button type="button" onclick="controller.requestTaxi()">REQUEST</button>

          </form>

      </div>

      <!--- Cancel offers/requests -->

      <!--- TODO: add buttons to allow the various functional requirements to be enacted -->

      <div id="cancel">

        <p><b>Click to cancel any offers or requests: </b></p>

        <button type="button" onclick="cancel()">CANCEL</button>

      </div>

      <!--- Map options-->

      <div id="update">

        <p><b>Click to edit Map display:</b></p>

        <button type="button" onclick="updateMap()">UPDATE</button>

        <button type="button" class="btn btn-info mt-3" onclick="centreMap()">CENTRE</button>

        <button type="button" class="btn btn-info mt-3" onclick="clearMarkersFromMap()">CLEAR</button>

      </div>

      <!--- TODO: add a placeholder for the HERE Map -->

      <div id="mapContainer">

      <!--Buttons for map functionality are above, styling is located in the CSS file.-->

      </div>

       <!-- Added a window for displaying results-->

      <div id="results">

        <div id="table">

        </div>

      </div>

    <!-- Scripts -->

    <script type="text/javascript" src="cordova.js"></script>

    <script type="text/javascript" src="js/helpers.js"></script>

    <script type="text/javascript" src="js/index.js"></script>

  </body>

</html>

CSS CODE:

/\* Heading styles \*/

  h2, h3 {

    font-size: 17px; /\* Adjust the font size \*/

    margin: 3px;

  }

  /\* Input field styles \*/

  input[type="text"], input[type="datetime-local"], select {

    font-size: 12px; /\* Adjust the font size\*/

    padding: 2px; /\* Add padding around the input fields \*/

  }

  /\* Button styles \*/

  button {

    font-size: 11px; /\* Adjust the font size \*/

    padding: 2px 2px; /\* Add padding around the buttons \*/

  }

  #mapContainer {

    width: 320px; /\* Adjust the width \*/

    height: 150px; /\* Adjust the height \*/

    border: 1px solid #000; /\* Add a thin border \*/

    margin: 0 left;

    overflow: hidden; /\* Hide any overflow \*/

  }

  #cancel p, #update p {

    margin-bottom: 3px; /\*  margin bottom to reduce vertical space \*/

  }

  /\* Button styles \*/

  #cancel button, #update button {

    margin-top: 2px; /\* margin top to reduce vertical space between buttons and paragraphs \*/

  }

  #update button{

    margin-bottom: 4px;

  }

|  |
| --- |
| *Comments* |
| The required input elements - ✓-  The buttons, such that they satisfy the functional requirements. - ✓-  The map code - ✓-  For a screenshot of the test - ✓- |
| - Well done; all the GUI elements corresponding to the needs of inputs and outputs of the app have been implemented.  - Correct call-back functions and syntax indicated on the buttons  - Good initial layout having the right initial layout of the UI elements. Correct rendering of HTML  - Correctly labelled screenshot of the app without interaction.   * Could have commented code with the FR   some of the fields for Offering and Requesting a Taxi could have been shared. This would have helped reduce space and allowed all elements to appear on a single screen without scrolling. |
| The functional requirements  1. Offer to share a ride by specifying when the car will be available in the offer, i.e. starting date and time plus the duration of the period they are prepared to wait.  The pick-up address of the sharing offer also needs to be specified as a string, this will be ‘Milton Keynes Central station’, or ‘Open University, Milton Keynes’. (FR1)  2. Request to share a ride by specifying the date and time when the ride is needed along with the pickup location. (FR1)  3. Cancel all offers or requests associated with their userid. (FR1)  4. Request that a map will be displayed with the current offers. (FR2)  5. Request that the map be centred on the current location. (FR2)  6. Register to use the taxi-sharing service using their Open University Computer Username (userid) e.g. ‘abc123’. (FR3) |

Question 2

Index\_2.js code (I have left out some of the comments at the top of the file):

// Execute in strict mode to prevent some common mistakes

"use strict";

// Declare a TaxiShare object for use by the HTML view

var controller;

document.addEventListener("deviceready", onDeviceReady, false);

function onDeviceReady() {

    console.log("Running cordova-" + cordova.platformId + "@" + cordova.version);

    // Create the TaxiShare object for use by the HTML view

    controller = new TaxiShare();

    //Camera

    console.log(navigator.camera);

}

// JavaScript "class" containing the model, providing controller "methods" for the HTML view

function TaxiShare() {

    console.log("Creating controller/model");

    // PRIVATE VARIABLES AND FUNCTIONS - available only to code inside the controller/model

    // Note these are declared as function functionName() { ... }

    var BASE\_GET\_URL = "http://137.108.68.13/openstack/taxi/";

    // NOTE HTTPS: at the time of writting we are investigating the creating a https service

    //             this will increase compatibility with Android. If we get it working we

    //             will ask you to modify the line above.

    var BASE\_URL = BASE\_GET\_URL;

    // NOTE CORS: if you get a CORS error we may recommend you insert code at this point.

    var icon = new H.map.DomIcon("<div>&#x1F695;</div>");

    var marker;

    // HERE Maps code, based on:

    // https://developer.here.com/documentation/maps/3.1.19.2/dev\_guide/topics/map-controls-ui.html

    // https://developer.here.com/documentation/maps/3.1.19.2/dev\_guide/topics/map-events.html

    // Initialize the platform object:

    var platform = new H.service.Platform({

        // TODO: Change to your own API key or map will NOT work!

        apikey: "IX5xI7ari4wCYqSANvz5l9HRXpaD9ZuhF7wva47GXNQ",

    });

    // Obtain the default map types from the platform object:

    var defaultLayers = platform.createDefaultLayers();

    // Instantiate (and display) a map object:

    var map = new H.Map(

        document.getElementById("mapContainer"),

        defaultLayers.vector.normal.map,

        {

            zoom: 15,

            center: { lat: 52.5, lng: 13.4 },

        }

    );

    // Create the default UI:

    var ui = H.ui.UI.createDefault(map, defaultLayers);

    var mapSettings = ui.getControl("mapsettings");

    var zoom = ui.getControl("zoom");

    var scalebar = ui.getControl("scalebar");

    mapSettings.setAlignment("top-left");

    zoom.setAlignment("top-left");

    scalebar.setAlignment("top-left");

    // Enable the event system on the map instance:

    var mapEvents = new H.mapevents.MapEvents(map);

    // Instantiate the default behavior, providing the mapEvents object:

    new H.mapevents.Behavior(mapEvents);

    var markers = []; // array of markers that have been added to the map

    var details = [];//Empty array of offer details

    // TODO Lookup an address and add a marker to the map at the position of this address

    function addMarkerToMap(address) {

        if (address) {

            // Hint: If you call the OpenStreetMap REST API too frequently, your access will be blocked.

            //       We have provided a helper function to prevent this however if you open the app

            //       on several browser windows at once you may still run into problems.

            //       Consider hardcoding locations for testing.

            // Hint: To ensure a marker will be cleared by clearMarkersFromMap, use:

            //       markers.push(marker);

            //       to add it to the markers array

            clearMarkersFromMap();

            // Convert the coordinates from the OSM API into point variables recognisable by HERE maps.

            var onSuccess = function (data) {

                // TODO 2(a) FR2.2

                // You need to implement this function

                // See the TMA for an explanation of the functional requirements

                // Hint: If you can't see the markers on the map if using the browser platform,

                //       try refreshing the page.

                var point = {

                    lng: data[0]["lon"],

                    lat: data[0]["lat"]

                };

                // For each location in the markers array, place a marker on the map.

                markers.push(point);

                map.setCenter(markers[0]);

                for (var i = 0; i < markers.length; i+=1){

                    marker = new H.map.DomMarker(markers[i], { icon: icon });

                    map.addObject(marker);

                }

            };

            // Hint: We have provided the helper function nominatim.get which uses

            //       the OpenStreetMap REST API to turn an address into co-ordinates.

            //       It does this in such a way that requests are cached and sent to

            //       the OpenStreetMap REST API no more than once every 5 seconds.

            nominatim.get(address, onSuccess);

        }

    }

    // Clear any markers added to the map (if added to the markers array)

    function clearMarkersFromMap() {

        // This is implemented for you and no further work is needed on it

        markers.forEach(function (marker) {

            if (marker) {

                map.removeObject(marker);

            }

        });

        markers = [];

    }

    // Obtain the device location and centre the map

    function centreMap() {

        // This is implemented for you and no further work is needed on it

        function onSuccess(position) {

            console.log("Obtained position", position);

            var point = {

                lng: position.coords.longitude,

                lat: position.coords.latitude,

            };

            map.setCenter(point);

        }

        function onError(error) {

            console.error("Error calling getCurrentPosition", error);

            // Inform the user that an error occurred

            alert("Error obtaining location, please try again.");

        }

        // Note: This can take some time to callback or may never callback,

        //       if permissions are not set correctly on the phone/emulator/browser

        navigator.geolocation.getCurrentPosition(onSuccess, onError, {

            enableHighAccuracy: true,

        });

    }

    // TODO Update the map with addresses for orders from the Taxi Sharing API

    function updateMap() {

        // TODO adjust the following to get the required data from your HTML view

        // TODO 2(a) FR2.1

        // You need to implement this function

        // See the TMA for an explanation of the functional requirements

        // Hint: You will need to call addMarkerToMap and clearMarkersFromMap.

        // Hint: If you cannot complete FR2.1, call addMarkerToMap with a fixed value

        //       to allow yourself to move on to FR2.2.

        //       e.g. addMarkerToMap("Milton Keynes Central");

        function matchSuccess(data) {

            var results = data.data;

            markers = [];

            var locations = [];

            details = [];

            alert("you have " +results.length +" match(es)");

            //Create an array of match results

            for(var i = 0; i < results.length; i+=1){

                locations.push(results[i]);

            }

            //Create an array of match detail objects

            for (var i=0; i<results.length; i+=1){

                    var offer = {

                    userid: results[i]['userid'],

                    time: results[i]['start'],

                    location: results[i]['offer\_address']

                };

                details.push(offer);

            }

            //populate and display a table of matches

            tableResults(details);

            // Send each address location to the map

            for (var i = 0; i < locations.length; i+=1){

                var address = (locations[i]['offer\_address']);

                addMarkerToMap(address);

            }

        }

        var userid = getInputValue("userid", "user1");

        var url = BASE\_URL+"matches?userid="+ userid;

        console.log("matches: sending GET to url", url);

        $.ajax({

            url: url,

            type: "GET",

            dataType: "json",

            success: matchSuccess,

            error: function(xhr, textStatus, errorThrown) {

                console.log(xhr.responseText);

                console.log(textStatus);

                console.log(errorThrown);

            }

        });

    }

    // Register userid with the taxi sharing service

    function register(userid) {

        // 2(a) FR3

        // This is implemented for you and no further work is needed on it

        // Note we have pre-registered your userid so using this is only required

        // should you want to add an additional (fictional) userid for testing.

        function onSuccess(obj) {

            console.log("register: received obj", obj);

            // Inform the user what happened

            if (obj.status == "success") {

                alert("User " + userid + " has been successfully registered.");

            } else if (obj.message) {

                alert(obj.message);

            } else {

                alert("Invalid userid: " + userid);

            }

        }

        // Post the userid to register with the Taxi Sharing API

        var url = BASE\_URL + "users";

        console.log("register: sending POST to " + url);

        $.ajax(url, { type: "POST", data: { userid: userid }, success: onSuccess });

    }

    // TODO Offer a taxi for the given userid

    function offer(userid, address, startTime, endTime) {

        // TODO 2(a) FR1.1

        // You need to implement this function

        // See the TMA for an explanation of the functional requirements

        function offerSuccess(data) {

            console.log("offer: received data ", data);

            if (data.status === "success") {

            alert("Your offer has been successfully received");}

        }

        var url = BASE\_URL + "orders";

        console.log("offer: sending POST to " + url);

        $.ajax(url, { type: "POST", data: {userid: userid, type: 0, address: address, start: startTime, end: endTime}, success: offerSuccess });

    }

    // TODO Request an offered taxi for the given userid

    function request(userid, address, startTime) {

        // TODO 2(a) FR1.2

        // You need to implement this function

        // See the TMA for an explanation of the functional requirements

        function requestSuccess(data){

            console.log("request: received data ", data);

            if (data.status === "success") {

                alert("Your request has been successfully received");}

            }

            var url = BASE\_URL + "orders";

            console.log("request: sending POST to " + url);

            $.ajax(url, { type: "POST", data: {userid: userid, type: 1, address: address, start: startTime}, success: requestSuccess });

    }

    // Cancel all orders (offers and requests) for the given userid

    function cancel(userid) {

        // 2(a) FR1.3

        // This is implemented for you and no further work is needed on it

        function onDeleteSuccess(obj) {

            console.log("cancel/delete: received obj", obj);

        }

        function onListSuccess(obj) {

            console.log("cancel/list: received obj", obj);

            if (obj.status == "success") {

                // Orders are in an array named "data" inside the object returned

                var orders = obj.data;

                // Inform the user what is happening

                alert("Deleting " + orders.length + " orders");

                // Loop through each one and delete it

                orders.forEach(function (order) {

                    // Delete the order with this ID for the given userid

                    var deleteUrl = BASE\_URL + "orders/" + order.id + "?userid=" + userid;

                    console.log("cancel/delete: Sending DELETE to " + deleteUrl);

                    $.ajax(deleteUrl, {

                        type: "DELETE",

                        data: {},

                        success: onDeleteSuccess,

                    });

                });

            } else if (obj.message) {

                alert(obj.message);

            } else {

                alert(obj.status + " " + obj.data[0].reason);

            }

        }

        // Get all the orders (offers and requests) for the given userid

        var listUrl = BASE\_GET\_URL + "orders?userid=" + userid;

        console.log("cancel/list: Sending GET to " + listUrl);

        $.ajax(listUrl, { type: "GET", data: {}, success: onListSuccess });

    }

    // Add table of match results to the HTML view

    function tableResults(data) {

        var col = [];

        for (var i = 0; i < data.length; i++) {

            for (var key in data[i]) {

                if (col.indexOf(key) === -1) {

                    col.push(key);

                }

            }

        }

        // Create a  results table.

        var table = document.createElement("table");

        var tr = table.insertRow(-1);

        //Add a table header

        for (var i = 0; i < col.length; i++) {

            var th = document.createElement("th");

            th.innerHTML = col[i];

            tr.appendChild(th);

        }

        // add results data to the table as rows.

        for (var i = 0; i < data.length; i++) {

            tr = table.insertRow(-1);

            for (var j = 0; j < col.length; j++) {

                var tabCell = tr.insertCell(-1);

                tabCell.innerHTML = data[i][col[j]];

            }

        }

        //Add the table to the view container

        var divShowData = document.getElementById('table');

        divShowData.innerHTML = "";

        divShowData.appendChild(table);

    }

    // Set initial HERE Map position

    centreMap();

    // PUBLIC FUNCTIONS - available to the view

    // Note these are declared as this.functionName = function () { ... };

    // Controller function to update map with matches to request or offer

    this.updateMap = function () {

        // 2(a) FR3

        // This is implemented for you and no further work is needed on it

        // Update map now

        updateMap();

    };

    // Controller function to centre map with matches to request or offer

    this.centreMap = function () {

        // 2(a) FR2.3

        // This is implemented for you and no further work is needed on it

        // Update map now

        centreMap();

    };

    // Controller function to register a user with the web service

    this.registerUser = function () {

        // 2(a) FR3

        // TODO adjust the following to get the userid from your HTML view

        var userid = getInputValue("userid", "user1");

        // Call the model using values from the view

        register(userid);

    };

    // Controller function for user to offer to share a taxi they have booked

    this.offerTaxi = function () {

        var defaultStartTime = convertToOrderTime(new Date());

        // TODO adjust the following to get the required data from your HTML view

        var userid = getInputValue("userid", "user1");

        var address = getInputValue("spickup", "Milton Keynes Central Station");

        var startTime = getInputValue("sdatetime", defaultStartTime); // eg. 2020:12:18:14:38

        var hours = getInputValue("sduration", "1"); // duration in hours

        // The following code is very sensitive to the formatting of the date/time.

        // The code above automatically populates a defaultStartTime of the correct

        // format based on the current date and time. If you have problems we

        // recommend you leave the input blank on the HTML and use the default.

        // The format of the date and time should be exactly like:

        // 2020:12:18:14:38

        // YYYY:MM:DD:HH:MM

        // OR like

        // 2021-04-01 12:00:00

        // YYYY-MM-DD HH:MM:SS

        // You may change the way this code works if you wish.

        // Please take care with the formatting of the dates for the API call!

        // The model requires an end time, but the view provides a duration, so...

        // ...convert the start time back to a Date object...

        var endDate = convertFromOrderTime(startTime);

        // ...add on the hours (ensuring the string is an integer first)...

        endDate.setHours(endDate.getHours() + parseInt(hours));

        // ...convert back to an end time string

        var endTime = convertToOrderTime(endDate);

        // Call the model using values from the view

        offer(userid, address, startTime, endTime);

    };

    // Controller function for user to request to share an offered taxi

    this.requestTaxi = function () {

        // TODO adjust the following to get the required data from your HTML view

        var userid = getInputValue("userid", "user1");

        var address = getInputValue("rpickup", "Open University, Milton Keynes");

        var startTime = getInputValue("rdatetime", convertToOrderTime(new Date()));

        // Call the model using values from the view

        request(userid, address, startTime);

    };

    // Controller function for user to cancel all their offers and requests

    this.cancel = function () {

        // TODO adjust the following to get the required data from your HTML view

        var userid = getInputValue("userid", "user1");

        // Call the model using values from the view

        cancel(userid);

    };

}

Screenshots:

Matches showing on map:

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generatedSuccessful Offer and Request:

Graphical user interface, application

Description automatically generated

|  |  |
| --- | --- |
| *Question 2 Business logic for mobile web app (20 marks)* | |
| *FR1.1 Offering a taxi for a duration of availability with a pick-up address (the campus or train station), starting date, time and duration (in hours) for which the offering user is prepared to wait using the taxi/orders API* | *3/3.5* |
| |  | | --- | | *• assembles the data for the post call appropriately.- ✓-* | | *• makes a correct post call to the service - - ✓-* | | *• handles errors appropriate l- ☒* | | *• handles the success case appropriately- - - ✓-* |     *Perhaps better for error handling rather than alert()*  *if (obj.status == "success") {*  *alert("Your taxi offer has been recorded");*  *} else if (obj.message) {*  *alert(obj.message);*  *} else {*  *alert(obj.status + " " + obj.data[0].reason);*  *}*  *anonymous function to handle the success of the POST, perhaps names F*  *the logic of the functions seems sound* |  |
| *FR1.2 Requesting a taxi at a pick-up address for a specific date and time using the taxi/orders API* | *3 /3.5* |
| |  | | --- | | *assembles the data for the post call appropriately.- - ✓-* | | *makes a correct post call to the service- - ✓-* | | *handles errors appropriately- ☒* | | *handles the success case appropriately- - ✓-* |   *syntax function looks correct, and there are no obvious syntax errors.* |  |
| *FR2.1 The matching offers and requests should be found via the matches API* | *6/6* |
| |  | | --- | | *removes any existing markers from the map (e.g. by calling clearMarkersFromMap - ✓-* | | *makes a correct get call to the service- - ✓-* | | *deals with at least 1 result appropriately- - ✓-* | | *deals with more than 1 result appropriately--* | | *makes attempt to add marker using value returned by service (eg by calling addMarkerToMap)- - ✓-* |   *the logic of the functions seems sound* |  |
| *FR2.2 - The map location should be found via the OpenStreetMap API and then shown on the map.* | *5/5* |
| |  | | --- | | *Handles errors appropriately – address not resolved -* | | *deals with cases where there is more than one match for the address, for example by selecting the first item -* | | *creates the marker-* | | *adds the marker to the map-* | | *centres the map, so the marker can be viewed-* |   the logic of the functions seems sound |  |
| *Q2 - For building and testing the app and including a screenshot in the solution document* | 2/2 |
| *The TMA brief clearly stated*   1. *Comment your code with appropriate TODO comments to identify where each FR is implemented so your tutor can mark your work.* 2. *Copy the code in www/js/index.js* ***that you have written for each requirement*** *into your Solution Document.*  * *Not the whole \*. js file – JUST your additional code lines.* |  |

QUESTION 3 (750/750 Words):

A)

Functional Requirement: **FR 7 Camera Functionality- ✓-**

Market Research:

Two very popular apps that have similar functionality to the taxi sharing app are Uber and Lyft, both of which have camera functionality.- ✓- In this case the purpose is to allow users to take and upload a photo of themselves for authentication purposes (so that their driver can verify they are collecting the correct passenger, and so passengers can ensure they are getting into the correct vehicle). This is important for both passenger and driver safety.- ✓-

Feature Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Uber** | **Lyft** | **Bolt** |
| In-app camera | Yes | Yes | Yes |
| Document uploading using camera | Yes | Yes | Yes |
| Verification using camera | Yes | Yes | Yes |
| Visual identification of driver and passenger | Yes | Yes | Yes |

*/\*\* M. Comment - All tables need a caption here – the title of what the table is and then referred to in the body of the text - Table 1: Possible plugins for FR ‘x’ \*\*/*

These camera features are a standard feature in taxi-sharing applications, so it is a must have for my taxi-sharing application in order for it to meet the standard that has been established.- ✓-

I have included another feature table with other features that could be considered with the camera integration.

|  |  |  |
| --- | --- | --- |
| Feature | Description | Market Evidence |
| Capture Photos | Allow users to capture photos using their device's camera. | Most social media apps, photo editing apps, and online marketplaces allow users to capture and upload photos directly from their device's camera. |
| Image Preview and Review | Show a preview of captured photos before saving or uploading. | Most apps provide a preview of captured photos/videos before saving or uploading, this allowing users to review and make necessary adjustments if the photo isn’t of the required quality. |
| Image Quality Settings | Allow users to adjust image quality settings, such as resolution and compression. | Certain apps allow users to adjust image quality settings, such as resolution, to optimize the captured photos based on their preferences and device capabilities. Although this primarily found in apps focused specifically on photography and editing. |
| Camera Switching | Allow users to switch between front and rear cameras (if they are on a mobile device). | Many apps allow users to switch between front and rear cameras, enabling them to capture different types of photos with ease (Imagine only being able take selfies with a rear-facing camera). |
| Image Gallery Integration | Integrate with the device's image gallery for access to captured photos. | Apps typically integrate with the device's image gallery, allowing users to easily access their captured photos. |
| Geolocation Tagging | Automatically tag captured photos with geolocation information | Many camera apps automatically tag captured photos with geolocation information, allowing users to know where the photo was taken. |

- ✓-

Proposed Solution:

In order to implement Camera functionality in the taxi-sharing app, I plan to use a Cordova plugin (please see question B). This will allow for the integration of the device’s camera into the app, enabling users to take photos which can be used authentication purposes.- ✓-

|  |
| --- |
| *Q3(a). Research ( #/6 marks)* |
| *Comments  - a brief feature table - evidence of relevant marketing research copied from Q1a, - A new requirement - is not described well enough* |
| *Very Good  The source(s) should be clearly stated/referenced.  Two basic questions frame this question.*  *a, What are you looking for to enhance the taxi share app in this FR?   b, What functionality are you looking for to enhance the taxi share app?  the first question will give us the research remit, as you need to tell us what you were searching for and why? How many are there that are similar? What are the specific features you searched for and why?   the second question, b, gives you the remit of the feature table that justifies your FR inclusion into the taxi-sharing app* |

B)

The Cordova plugin I have selected */\*\* M. Comment How many? What were the search criteria \*\*/what was that selection thinking…? \*\*/* is the “[com.matrixgz.cordova-plugin-camera](https://www.npmjs.com/package/com.matrixgz.cordova-plugin-camera" \t "_self) *[/\*\* M. Comment How?\*\*/](https://www.npmjs.com/package/com.matrixgz.cordova-plugin-camera" \t "_self)*

**”.** As described in the plugins readme file, “this plugin defines a global navigator.camera object, which provides an API for taking pictures and for choosing images from the system's image library.” *(npm, 2018)- ✓-*

I chose this plugin for a few reasons:

* It has the highest quality rating for “Cordova-plugin-camera “”
* It can run on both Android and browser (I have been using both whilst working on the TMA as I find it easier working with the browser development tools).
* It is well documented and being used by others, as seen by the “Weekly Downloads” tab.
* It can be installed on the version of Cordova I am using.
* It was first published 5 years ago and has been updated constantly.- ✓-

|  |
| --- |
| *Q3(b). Select a Cordova plugin (6 marks)* |
| *Comments identifying a relevant Cordova plugin. describing the possibilities that were considered giving a rationale for the choice made* |
| *Note 1: - Were several alternatives considered? What were the criteria of your comparison why did you select the one you did? what was important for the choice of your selection? You could have also justified it by the number of downloads. Or the number of recent updates… et cetera et cetera  Selection criteria and the reviewing process are important discussions, the aspects of which need to have been developed to answer this question.* |

C) JS Code:

// Define global variables

var imageElement = document.getElementById('myImage');

var options = {

    quality: 50,

    destinationType: Camera.DestinationType.DATA\_URL

};

// Define the success callback function for taking photo

function onSuccess(imageData) {

    // Set the source attribute of the image element to the image data

    imageElement.src = "data:image/jpeg;base64," + imageData;

}

// Define the failure callback function for taking photo

function onFail(message) {

    alert('Failed because: ' + message);

}

// Function to open camera

function openCamera() {

    if (navigator.camera) {

        // Enable the take photo button

        document.getElementById("take-photo-btn").disabled = false;

        // Call the camera function to open the camera

        navigator.camera.getPicture(onSuccess, onFail, options);

    } else {

        alert('Camera object not defined.');

    }

}

// Function to take photo

function takePhoto() {

    if (navigator.camera) {

        // Call the camera function to take a photo

        navigator.camera.getPicture(onSuccess, onFail, options);

    } else {

        alert('Camera object not defined.');

    }

}

|  |
| --- |
| *Comments* |
| *commenting good and a brief discussion in your text to say where and how this plugin was implemented* |

To use the camera feature, the user is required to click on the “Open Camera” button, this will then open the camera and the user will be able to prepare to take the photo. The photo is captured when the user decides to click on the “Take Photo” button. The photo is then displayed to the user next to the registration section.- ✓-

|  |
| --- |
| *Q3(c). Implement coding (6 Marks)* |
| *Comments Code - Solution Document & zip file (index\_3.html / index\_3.js) (3 marks) explanation of how the code addresses the FR. (2 marks) How the user would use the feature (1 mark)*  *The code needed to make use of the plug-in to implement the FR (\*more than copying and pasting example code from the plugin documentation)* |
| *Comments*  *explanation of how the code addresses the FR* |
| *User Instructions A good description of how a user would use the new feature* |

D)

I tested on my browser and eventually the app worked as expected. I decided to use two separate buttons as I was having an issue when just using one “Take Photo” button. Specifically, the camera would open successfully but I was then unable to actually take a photo. Adding two sperate buttons helped me resolve this issue, and in my opinion, makes it clearer to the users how to use the camera functionality.

Ideally, I would have liked the captured photo to be displayed better but from a purely functional standpoint I was satisfied with the implementation and didn’t see the need to worry about further CSS styling.

Screenshot of when “OPEN CAMERA” is clicked:

Graphical user interface, application

Description automatically generated

Screenshot 3 showing captured image (of my textbook):- ✓-

Graphical user interface

Description automatically generated

|  |
| --- |
| *Q3(d). Build and test your project (2 marks)* |
| *Comments Build and test the app (You can get marks for this part even if the build fails.)*  *Explain app worked as expected //List the obstacles // screenshot (screenshot\_3. Png) in your Solution Document. - good* |

QUESTION 4 (860/900 Words)

A)

I have identified a gap for an app called “HockeyHub”. - ✓-

HockeyHub is a mobile app aimed at connecting field hockey enthusiasts, including players, umpires, clubs, and coaches, on a single platform. The app allows users to find nearby hockey clubs, join teams, view training days and times, communicate with fellow players, and track statistics.

HockeyHub aims to connect field hockey players with each other and with clubs based on their profile information and preferences (matching them in a way that is similar to most dating apps), all on one convenient central application.

By offering a comprehensive solution for the field hockey community, HockeyHub aims to fill the market gap for a dedicated and user-friendly app that brings together all aspects of the field hockey world in one place.- ✓-

Feature Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **App** | **Team Communication** | **Scheduling** | **Statistics Tracker** | **Player + Club Matchmaking** |
| CA Field Hockey | Yes | Yes | No | No |
| Sports Engine | No | Yes | Yes | No |
| Teamo | Yes | Yes | Yes | No |
| Activity Pro | Yes | Yes | No | No |

*/\*\* M. Comment references to the apps in your table are needed \*\*/*

Key words used to produce this table: Field Hockey, Communication, Social, Management, Sports Management

|  |
| --- |
| Q4(a). Identifying a market gap (4 marks) |
| * The idea - 1 * For a feature table and an analysis of potential gaps the app could fill. – 3 |
| Needed to identify a niche feature to distinguish the app from rivals, as such could have included your app in the table to assist in highlighting your features in comparison.  How can your reader tell your niche feature? Would like to have read one or two lines confirming your niche feature(s).  You should provide references to the apps in your table  - Good, |

B)

Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **FR** | **Name** | **Description** | **Obstacle** |
| FR1 | Training Scheduling | The app should allow captains and coaches to add and edit training dates, and provide a description (optional)- ✓- | Managing multiple teams and players: HockeyHub may need to manage schedules for multiple teams with different levels, age groups, or locations, as well as individual players with varying training needs. Coordinating and aligning the schedules of different teams and players while adhering to their unique requirements can be challenging, *- ? -*  particularly if there are conflicts or overlaps. |
| FR2 | Matchmaking | The app should have a player matching algorithm that matches players to clubs and teams based on their skill level, position, availability, and other relevant criteria.- ✓- | Data quality and accuracy:  The accuracy and quality of the data used for matchmaking, such as player skill levels, positions, and availability, can greatly impact the effectiveness of the algorithm. Inaccurate or incomplete data could lead to inaccurate matching results.- ✓- |
| FR3 | Team Communication | Team communication tools, such as messaging and notifications- ✓- | User adoption and engagement:  Encouraging team members to actively use the team communication feature can be challenging. If team members are not motivated or find it difficult to use the communication feature, it may not be effectively utilized. - ✓- |
| FR4 | User Registration | Users should be able to create accounts and register with HockeyHub using a valid email address and password.- ✓- | User abandonment and incomplete registrations: Users may start the registration process but abandon it before completion due to various reasons. Incomplete registrations may result in lost potential users and incomplete user profiles. Implementing measures to capture and follow up with incomplete registrations, such as reminder emails, can help reduce user abandonment and increase successful registrations. |
| FR5 | Player Management | Users should be able to manage players in their teams, including adding, editing, and deleting players, as well as assigning players to specific positions or roles.- ✓- | Player registration and eligibility: Player management in HockeyHub may involve player registration, eligibility verification, and compliance with league rules. Ensuring accurate registration, verification, and tracking of player eligibility based on league rules, age limits, or other criteria can be challenging. |

|  |
| --- |
| Q4(b). App functional requirements (10 marks)   * Five functional requirements for your app * A table with each row describing one functional requirement and unique obstacle |
| Need to identify both the function and the obstacles in achieving the function. |
| - Provided a good set of FRs however you have provided descriptions of some of these FRs rather than their obstacles. |

C.I)

About this application*: /\*\* M. Comment NAME…?\*\*/*

HockeyHub is a free application that allows for seamless organisation and communication between clubs and players. What makes HockeyHub unique is the personalized experience the app provides, matching Hockey players to other players, and clubs based on their profile information and preferences. This includes information such as “Playing experience”, “Preferred Positions”, “Highest Playing level”, location, and more.- ✓-

This allows players to connect with clubs most suited to them, saving them the hassle of browsing individual clubs’ websites in order to find the best club. HockeyHub also allows clubs the opportunity to match with specific players they may be looking for, for example (a Centre-mid who has played National League).- ✓-

The application aims to make Field hockey more accessible to players of all ability levels and provide assistance to those who may have just relocated and are searching for a new club.

|  |
| --- |
| Q4(ci). Mobile app proposal (#/3) |
| * Describing the app and its purpose – * describing the features it offers –  (Copy of answer in solution document.) |
| ‘Such a description would normally be the first paragraph shown in an app store, brief, informative, and attractive’.  - You have provided a good description of your app   - You have, perhaps, made it over complicated simplicity was key here KIS  - It does appear to have been rushed. It is good, but perhaps it needed an edit I had to read it three times – not good as a sales pitch |

II)

Graphical user interface

Description automatically generated

|  |
| --- |
| Q4(c.ii). App wireframe (5 marks) |
| * GUI elements cover the functional requirements - * Choosing appropriate UI elements for most of the input/output that was proposed – * Making good use of space. This is an important design imperative – * The wireframe takes into account the form factor of the intended app – |
| Taking that this is an initial proposal, as a first draft.  Needed to show functionality.  - I would like, need to, have seen you indicating where the access to that FR appears in this wireframe  - In this question, we were looking for the form factor of the intended app, selecting the correct type of UI element for the input/output purpose, having an appropriate number of GUI elements with respect to the functional requirements and making good use of space (design).  A wireframe is a low-fidelity way to present a product and can efficiently outline structures and layouts. Wireframe is the basic and visual representation of the design. Your wireframe design doesn’t need to focus too much on minutiae but must express design ideas and should not miss any important parts.  A wireframe is like a channel that helps team member understand their projects better.  - It is more like a mock-up of the app’s interface has a realistic overall appearance rather than what could be considered a wireframe, a conceptual model of functionality. |

III)

Uploaded to Open Studio

IV)

My comments to Adam L:

Hi Adam, I clearly understand the purpose of your application and your wireframes are clear and easy to understand. As you describe it, I think this is a useful all-in-one-app with a variety of useful but simple features. My initial thoughts are that there are already a fair number of apps with very similar features that are already established and popular, so that would be a reason why I personally might not download the app. What is appealing is the simplicity of the app, as sometimes these scheduling and productivity apps tend to be over complicated. I think another area of importance relates to privacy concerns around the accessing of payslips on the mobile app, as none of the other information is as sensitive.

I think the addition of a “back” button or “home” button might be useful as there doesn’t appear to be anything like that at the moment.

(150 Words)

My comments to Ina A:

Hi Ina, I understand the purpose of your application and your wireframe is understandable, although I think a few of the features could be improved by making some changes, such as adding a slider for location distance instead of using several checkboxes, and the same for price range. I can understand how to use the application just by looking at the wireframe, although it may be useful to think about how the design may look on a mobile device as I think more information may need to be placed vertically compared to horizontally, as it may be quite hard to read with a lot of information.

As the app may collect user data, such as location information and email addresses, it's important to prioritize user privacy and data protection. Implementing appropriate security measures and clearly communicating the app's data handling practices to users is necessary.

(145 Words)

|  |
| --- |
| Q4(C iv). Clear TMA instruction was … “proposals you have chosen” - missing |
| iv. Reviewing other proposals  “Copy both the proposals you have chosen (these do not count towards your word count) and your comments on them into your TMA document, labelling each of them clearly ‘My comments to [your fellow student’s name]’.” |

|  |
| --- |
| Q4(C iv). Reviewing other proposals (6 marks) |
| * Constructive and also specific * Assessment * Dialogue related to the possible alternatives |
| “proposals you have chosen” missing copy, but your comments are constructive and useful  One more revision would have helped with the structure of the feedback to be constructive |
| Suggestions are Positive |

V)

Comments from Emma:

Hi Luke,

I think your app is well thought out, and although I have not played hockey, I have been involved in other sports and know it can be difficult to find teams of the right level. As I do not play hockey, I would not download this app, but I can see its usefulness.

I think the overall layout is good, and it is clear where all the features are. However, it may be easier for users to differentiate the club and player connections if they were in different app sections.

As clubs can use the app already, it could be nice to include features for clubs to connect if they have players at similar levels to arrange matches. Additionally, rather than just showing the latest suggested connections, it may be nice for people to see all the potential club and player suggestions.

It is a great idea and would benefit small teams and local players.

Good luck,

Emma

My updated proposal:

I have chosen to keep my app proposal (from question C.I) the same as I believe it does a suitable job of explaining the purpose of the app, as well as the features it offers.  
  
I have, however, changed my wireframe to illustrate various changes based on some of the feedback I received.   
- Club connections and player connections have been separated, users can now see their suggested connections at the top of the screen, as well as all available connections at the bottom.  
- The "Profile" page has moved to the top right of the screen to allow for the separate "Clubs" and "Players" pages.  
- Each page heading will appear at the top left of the screen and will allow users to jump back to the top of the page when pressed.  
-The "Latest" page will now display any unread broadcast communications from other clubs or players (i.e. clubs advertising for new players of a certain position). All private messages will be displayed on the "Messages" page.  
  
These changes should hopefully improve the usability of the app by giving users more complete control over their connections.- ✓-

Updated Wireframe:

Graphical user interface

Description automatically generated

|  |
| --- |
| Q5(C v.) Your final proposal (5 marks) |
| * For the comments on how and why they changed (or did not change) * For changes made, the change extent, improved proposal, or did not change proposal with an explanation. |
| A structured approach is needed to answer this question. Firstly, you need to identify all the suggestions made on your proposal; secondly, discuss which suggestion(s) you have decided to accept or reject with justification. These should be clear, and presented in your answer. |

REFERENCES:

* npm. (2018). *com.matrixgz.cordova-plugin-camera*. [online] Available at: https://www.npmjs.com/package/com.matrixgz.cordova-plugin-camera?activeTab=readme#camera-getPicture-examples [Accessed 4 Apr. 2023].
* Uber. (2023). *Rider Safety - Is Uber Safe for Riders?* [online] Available at: https://www.uber.com/gb/en/ride/safety/?utm\_campaign=CM2180099-search-google-brand\_184\_-99\_GB-National\_o-d\_web\_acq\_cpc\_en\_T1\_Generic\_BM\_uber%20information\_kwd-298370892673\_610671926827\_137354978845\_b\_c&utm\_source=AdWords\_Brand [Accessed 6 Apr. 2023].
* ‌ Lyft.com. (2023). *How Lyft Works to Keep Riders Safe*. [online] Available at: https://www.lyft.com/safety/rider [Accessed 19 Apr. 2023].
* Bolt. (2023). *Safety for Riders | Bolt*. [online] Available at: https://bolt.eu/en-gb/rider/safety/ [Accessed 10 Apr. 2023].
* MarketWatch. (2023). *Global Mobile Phone Camera Module Market Size Growth Rate Analysis 2023 by Top Key Players, Product Types, End Users, Drivers and Forecast to 2027*. [online] Available at: https://www.marketwatch.com/press-release/global-mobile-phone-camera-module-market-size-growth-rate-analysis-2023-by-top-key-players-product-types-end-users-drivers-and-forecast-to-2027-2023-01-27 [Accessed 111 Apr. 2023].

‌