Dokumentation: Evermind App

Inhalt

Einleitung

Innerhalb des Moduls 335 in der Studienwoche war es das Ziel eine Mobile-Applikation zu realisieren. Unsere Umsetzung ist eine Hybrid-App, welche anhand den gängigen Web-Standards entwickelt wurde. Da wir in diesem Bereich der Entwicklung nur wenig Erfahrung haben, konnten wir viel Neues lernen.

Idee

Sehr oft kommt es vor, dass eine Person einen wichtigen Einfall hat, der nicht vergessen werden sollte. Leider fehlt bei solchen Einfällen aber oft die Zeit diese aufzuschreiben. Mit der Applikation Evermind wird diesem Problem Abhilfe geschafft. Mit einem Klick auf einen Knopf werden die Uhrzeit, das Datum und der Standort gespeichert. Zusätzlich lässt sich eine Beschreibung zum Eintrag erfassen. Solch ein Eintrag stellt einen gedanklichen Anker dar. Diese Anker Einträge lassen sich dann in einer Liste abrufen, archivieren oder löschen. Die Benutzung soll intuitiv geschehen, darum bietet das App ein simples Benutzererlebnis.

Anforderungsdefinition

Fachlicher Inhalt

Mobile App zum Festhalten von Mikro-Erinnerungen (sog. Anker).

Verwaltung:

- Erfassen von Zeitpunkt
- Erfassen von Standort
- Aufnahme von Beschreibung
- Alte Items können archiviert werden
- Aufbereitete Liste mit den Items (Momentan & Archiv)

User Interface:

- Hauptseite mit grossem Button um Ankerpunkt zu erstellen und kleiner Button um zur Liste zu gelangen
- Liste mit den Ankerpunkte

Funktionelle Anforderungen

Muss-Kriterien

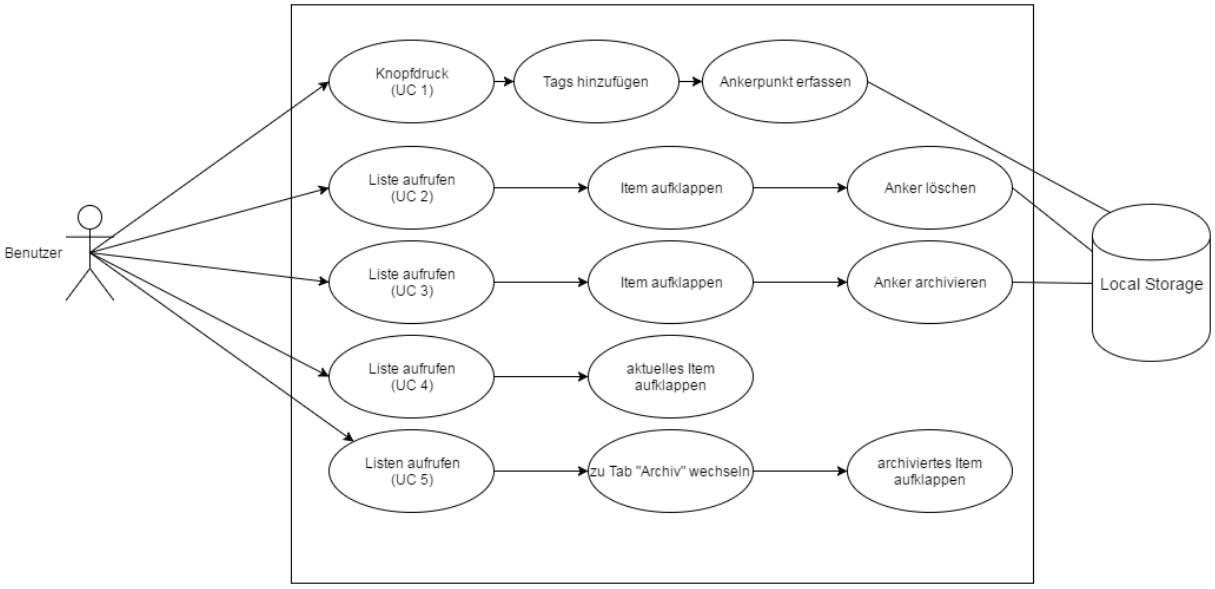
- Ankerpunkt mit Zeitpunkt und Standort erfassen
- Liste mit den aktuellen Ankerpunkten
- Liste mit den archivierten Ankerpunkten

Kann-Kriterien

- Farbliche Rückmeldung zu den Anzahl der Ankerpunkte im Hauptbildschirm
- Hinzufügen von Beschreibuzng zu den Einträgen
- Suchfunktion in den Listen
- Kartenausschnitt für die Standortinformation.

Analyse

Use Case Diagramm

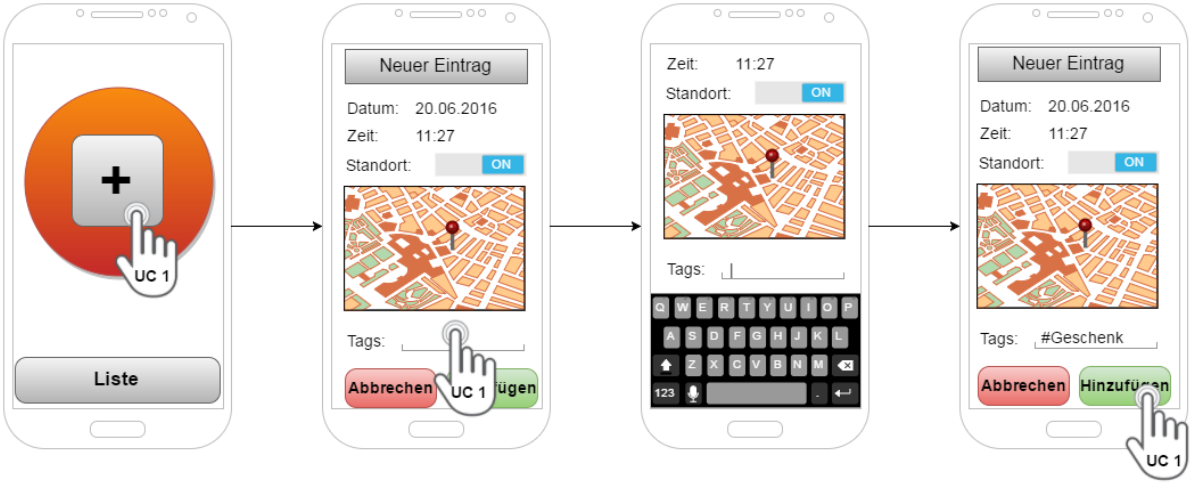


Beschreibung

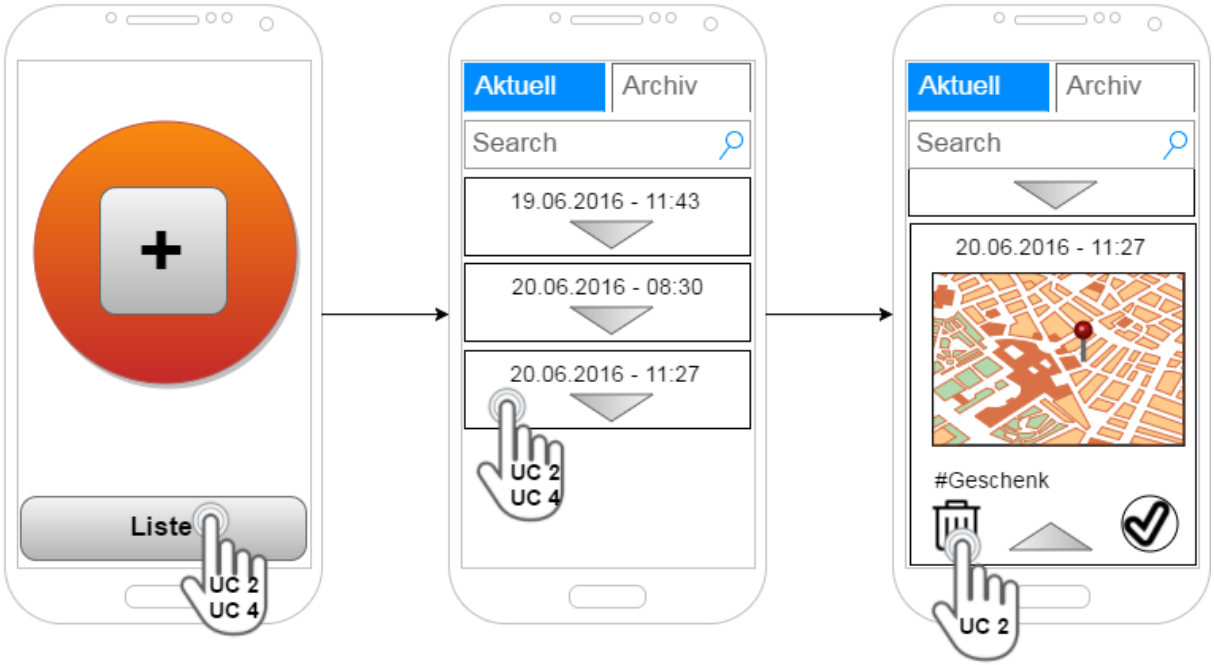
- UC1:
  - Ein neuer Eintrag wird über das Hauptmenü erstellt und mit eine Beschreibung versehen.
- UC2:
  - Ein Eintrag wird von der Liste aus gelöscht.
- UC3:
  - Ein Eintrag wird von der Liste aus archiviert.
- UC4:
  - Die Liste mit den aktuellen (nicht archivierten) Einträgen wird aufgerufen und ein Eintrag geöffnet.
- UC5:
  - Liste mit den archivierten Einträgen wird aufgerufen und ein Eintrag betrachtet.

Storyboard

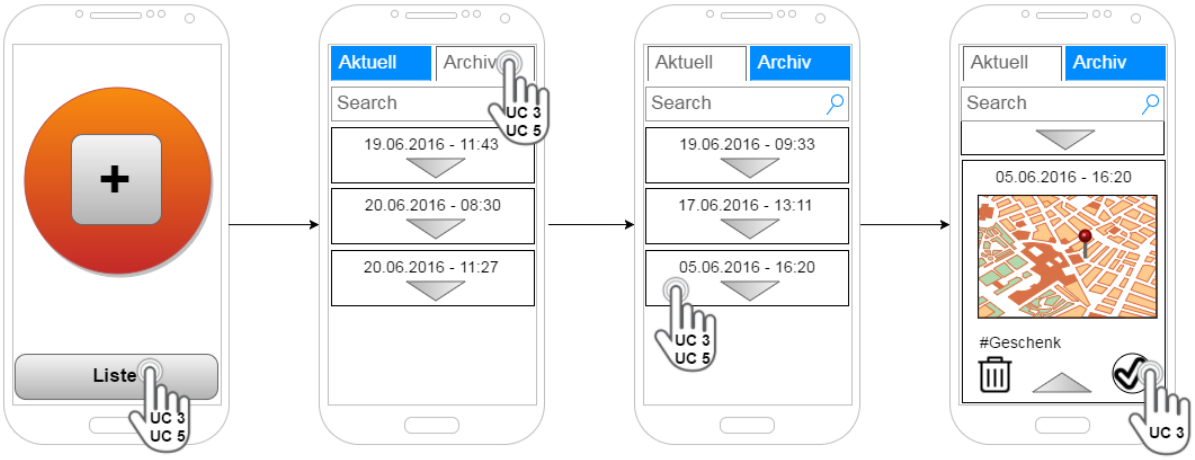
UC1



UC2 & UC4



UC 3 & UC 5

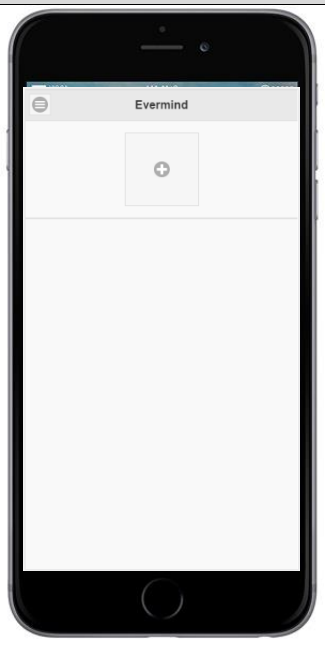


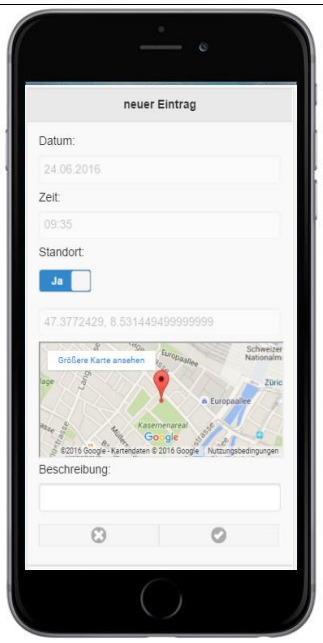
Implementierung

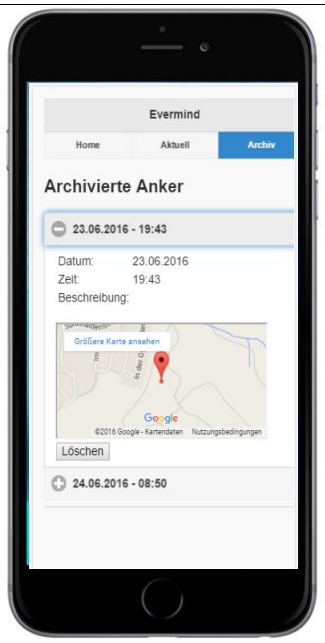
Die Implementierung erfolgte in HTML5 und JavaScript mit der jQuery Mobile Bibliothek. Als Entwicklungsumgebung dienten Brackets von Adobe und die Entwickler-Tools in Google Chrome.

Design

Das Design konnte grösstenteils gemäss Storyboard realisiert werden.

Ansicht	Beschreibung
	Einstiegsseite: Mit dem grossen Plus-Button besteht die Möglichkeit einen neuen Ankerpunkt zu erstellen. Mit der kleineren Schaltfläche oben links gelingt man zur Listenansicht.

	Neuen Eintrag erstellen: Datum, Zeit sowie den Standort wird automatisch übernommen. In einer kleinen Karte wird der Standort angezeigt. Eine Beschreibung kann optional hinzugefügt werden. Mit Betätigung der «X-Schaltfläche» wird der Anker ohne zu speichern verworfen. Mit der Taste mit dem Häkchen wird der Anker in die aktuelle Liste gespeichert.
	Listenansicht «Aktuell»: Die einzelnen Ankerpunkte werden als aufklappbare Liste mit allen Details angezeigt. Beim oberen Beispiel ist der Anker ausgeklappt. Der untere Anker blieb jedoch geschlossen. Mit den Schaltflächen unterhalb der Karte kann ein Anker gelöscht oder archiviert werden. Hier kann auch einfach wieder auf den Hauptbildschirm gewechselt werden oder zur Liste der archivierten Anker.

	Listenansicht «Archiv»: Hier wurde das Design gleich gehalten wie bei der Liste Aktuellen Elemente. Jedoch wurde im Element logischerweise der Button «Archivieren» entfernt.

Testen

Anforderungs-Tests

Muss-Kriterien

Nr	Testfall	Erwarteter Output	Status
1	Ankerpunkt mit Zeitpunkt und Standort erfassen	Korrekte Anzeige des aktuellen Datums und der aktuellen Zeit	OK
2	Liste mit den aktuellen Ankerpunkten	Übersicht über alle aktuellen Ankerpunkte in einer aufklappbaren Liste	OK
3	Liste mit den archivierten Ankerpunkten	Übersicht über alle archivierten Ankerpunkte in einer aufklappbaren Liste	OK
4	Erfassen von Zeitpunkt	Aktuelle Uhrzeit - automatisch	OK
5	Erfassen von Datum	Aktuelles Datum – automatisch	OK
6	Erfassen einer Beschreibung	Textfeld für Beschreibung	OK
7	Alle Items können archiviert werden.	Über Schaltfläche in der Liste	OK

Kann-Kriterien

Nr	Testfall	Erwarteter Output	Status
1	Hinzufügen von Beschreibung zu den Einträgen	Textfeld für Beschreibung	OK
2	Kartenausschnitt für die Standortinformation.	Kartenausschnitt in Liste und beim Erfassen	OK
3	Suchfunktion in den Listen	Suchfeld oberhalb der Listen	NOK
4	Farbliche Rückmeldung zu der Anzahl der Ankerpunkte im Hauptbildschirm	Je mehr Ankerpunkte je dunkler die Farbe um den Hinzufügen-Knopf	NOK

Verbesserungspotential

Vor allem die Struktur des Source-Codes weist noch einige Mängel auf. Für diese Applikation bestehen lediglich zwei Files – eine HTML-Datei und eine JavaScript-Datei. In einem nächsten Projekt sollte dies nicht mehr so geregelt werden. Es sollte mehr auf die Technik «MVC» geachtet werden. Dies ermöglicht eine grössere Individualität jeder Datei.

Auch bezeichnet das Projektteam den Start der Implementierung als happig. Zu Beginn wurden die Software-Teile aufgeteilt und von da an arbeitete jedes Projektmitglied für sich. Hier hätte man mehr zusammenarbeiten können und das neu erkannte wissen zu JavaScript und JQuery Mobile regelmässiger austauschen.

Fazit

Abschliessend können die Autoren sehr zufrieden auf ihr Projekt zurückschauen. Die Applikation wurde implementiert und ist ausführbar. Einige Elemente wurden programmtechnisch sauber und elegant gelöst. Die Teamarbeit funktionierte ebenfalls einwandfrei in wie auch neben der Schule. Bei Schwierigkeiten konnte man stets auf die Unterstützung seiner Teamkollegen zählen.

Anhang

Source-Code «evermind.html»

```
<!DOCTYPE html>
<html>
<head>
        <title>Page Title</title>
        <meta name="viewport" content="width=device-width, initial-scale=1" charset="utf-8">
        <link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"
/>
        <script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
        <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
    <script src="evermind_v1_0.js"></script>
</head>

<body>
    <!-- Start Hauptseite -->
    <div data-role="page" id="main">

        <!-- Header -->
        <div data-role="header">
            <h1>Evermind</h1>
            <!-- Button to acces list -->
            <a href="" class="ui-btn ui-icon-bars ui-btn-icon-notext" id="listview"></a>
        </div>

        <!-- Main -->
        <div role="main" class="ui-content">
            <!-- Button for new anchor -->
            <a href="#newAnchor" class="ui-btn ui-icon-plus ui-btn-icon-notext" id="newAnchorButton"
style="width:100px;height:100px;margin:0 auto;"></a>

        </div>

        <!-- Footer -->
        <div data-role="footer">
        </div>

    </div><!-- /Hauptseite -->

    <!-- Start Seite für neuer Eintrag -->
    <div data-role="page" id="newAnchor">

        <!-- Header -->
        <div data-role="header">
            <h1>neuer Eintrag</h1>
        </div>

        <!-- Main -->
        <div role="main" class="ui-content">

            <!-- Form for anchor data -->
            <form method="post">
                    <!-- Date -->
```

```
                    <label for="date">Datum:</label>
                    <input type="text" disabled="disabled" name="date" id="date">

                    <!-- Time -->
                    <label for="time">Zeit:</label>
                    <input type="text" disabled="disabled" name="time" id="time">

                    <!-- Location -->
                    <label for="location">Standort:</label>
                    <!-- Location: switch -->
                    <select name="locationSwitch" id="locationSwitch" data-role="slider" data-
mini="true">
                        <option value="no">Nein</option>
                        <option value="yes" selected>Ja</option>
                    </select>

                    <!-- Location: text -->
                    <input type="text" disabled="disabled" name="location" id="location">

                    <!-- Map goes here -->
                    <div id="mapholder"></div>

                    <!-- Description text -->
                    <label for="description">Beschreibung:</label>
                    <input type="text"  name="description" id="description">

            </form>

            <!-- Button to cancel anchor entry -->
            <a href="#main" class="ui-btn ui-icon-delete ui-btn-icon-notext" id="cancelAnchorButton"
style="float:left; width: 49%;"></a>

            <!-- Button to confirm anchor entry -->
            <a href="#main" class="ui-btn ui-icon-check ui-btn-icon-notext" id="confirmAnchorButton"
style="width:49%; margin-top:16px;"></a>

        </div>

        <!-- Footer -->
        <div data-role="footer">
        </div>

    </div><!-- /Page for reviewing new anchor entry -->

    <div data-role="page" class="ui-content" id="pageone">
        <div data-role="header">
            <h1>Evermind</h1>
        </div>
   <!-- Navigation Bar -->
   <div data-role="navbar">
    <ul>
        <li><a href="#main">Home</a></li>
        <li><a href="#pageone" class="ui-btn-active ui-state-persist">Aktuell</a></li>
        <li><a href="#pagetwo" id="archive">Archiv</a></li>
    </ul>
   </div>

   <h2>Aktuelle Anker</h2>
```

```
   <!-- table for Output in list element-->
   <div data-role="collapsibleset" data-content-theme="a" data-iconpos="left" id="set">
  </div>

  <!-- Footer -->
  <div data-role="footer">
  </div>
</div>
<!-- Second Page -->
  <div data-role="page" class="ui-content" id="pagetwo">
   <div data-role="header">
        <h1>Evermind</h1>
   </div>

   <!-- Navigation Bar -->
   <div data-role="navbar">
    <ul>
        <li><a href="#main">Home</a></li>
        <li><a href="#pageone">Aktuell</a></li>
        <li><a href="#pagetwo" class="ui-btn-active ui-state-persist" id="archive">Archiv</a></li>
    </ul>
   </div>

   <h2>Archivierte Anker</h2>
   <div data-role="collapsibleset" data-content-theme="a" data-iconpos="left" id="setArch">
   </div>

   <!-- Footer -->
   <div data-role="footer">
   </div>

  </div>
</body>
</html>
```

Source-Code «evermind.js»

```
function setUpEvents(){

    // Object for Date
    var d = new Object();

    // Date for output
    var outputDate = "";
    // Time for output
    var outputTime = "";
    // Location data for output
    var outputLocation = "";
    // Description for anchor
    var description = "";

    // Location data
    var latitude = "";
    var longitude = "";

    /**
      * Clear button for description input field
      */
    $("#description").textinput({clearBtn: true});

    /**
      * Click event of button to generate new anchor
      */
    $("#newAnchorButton").click(function(){

        // Get current date object
        d = new Date();

        // Get date in output format
        getOutputDate();

        // Get time in output format
        getOutputTime();

        // Check if location is on
        if("yes" == $("#locationSwitch").val()){
            // Get location
            getLocation();
        }

        // Show outputDate in the view
        document.getElementById("date").value = document.getElementById("date").defaultValue =
outputDate;

        // Show outputTime in the view
        document.getElementById("time").value = document.getElementById("date").defaultValue =
outputTime;

        /*document.getElementById("location").value = document.getElementById("location").defaultValue
= outputLocation;*/

        // Check if location is on
```

```
        if("yes" == $("#locationSwitch").val()){
            // Show map in the view
            getMap();
        }
        });

    /**
      * Formats date for output
      */
    function getOutputDate(){
        outputDate = ("0" + d.getDate()).slice(-2) + "." + ("0" + (d.getMonth() + 1)).slice(-2) + "."
+ d.getFullYear();
    }

    /**
      * Formats time for output
      */
    function getOutputTime(){
        outputTime = ("0" + d.getHours()).slice(-2) + ":" + ("0" +d.getMinutes()).slice(-2);
    }

    /**
      * Detect change in location setting
      */
    $("#locationSwitch").change(function(){
        // Location off
        if("no" == $("#locationSwitch").val()){
            // Hide elements with location data
            $("#location").hide();
            $("#mapholder").hide();
        // Location on
        } else {
            // Show elements with location data
            $("#location").show();
            $("#mapholder").show();
            // Update location data
            getLocation();
            getMap();
        }
    });

    /**
      * Click event of anchor confirmation button
      */
    $("#confirmAnchorButton").click(function(){

        // Get the data from description field
        getDescription();

        // Create object for anchor item
        var ankerItem = new Object();

        // Save date data in object
        ankerItem.timestamp = d.getTime();
        ankerItem.dateString = outputDate;
        ankerItem.timeString = outputTime;

        // Save location data in object
```

```
        if("yes" == $("#locationSwitch").val()){
            ankerItem.latitude = latitude;
            ankerItem.longitude = longitude;
        }

        // Save description in object
        ankerItem.description = description;
        // Set item not archived
        ankerItem.archived = false;

        // Save object in local storage
        localStorage.setItem(d.getTime(), JSON.stringify(ankerItem));

        // Reset values from input fields
        resetValues();
    });

    /**
      * Click event of anchor cancel button
      */
    $("#cancelAnchorButton").click(function(){
        // Reset values from input fields
        resetValues();
    });

    /**
      * Reset value of input fields
      */
    function resetValues() {
        // Set description field blank
        $("#description").val("");
    }

    /**
      * Read out data from description field in view
      */
    function getDescription() {
        description = $("#description").val();
    }

    /**
      * Accesing geo location API and gets relevant data
      */
    function getLocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showPosition);

            // Format location data for ouput
            getOutputLocation();

        } else {
            // Alert if there's an error with accesing the geo location
            window.alert("Error Geo Location");
        }
    }

    /**
      * Get location data from geo location object
```

```
      */
    function showPosition(position) {
        latitude = position.coords.latitude;
        longitude = position.coords.longitude;
    }

    /**
      * Format location data for output
      */
    function getOutputLocation(){
        outputLocation = latitude + ", " + longitude;

        document.getElementById("location").value = document.getElementById("location").defaultValue =
outputLocation;
    }

    /**
      * Get map from google maps with location data and show in view
      */
    function getMap(){
        $("#mapholder").html('<iframe
src="http://maps.google.com/maps?q='+latitude+","+longitude+'&z=15&output=embed"
width="100%"></iframe>');
    }

    // delete button in list element

    $(document).on('click', '.delete', function () {
        $(this).closest('[data-role=collapsible]').remove();
        var key = this.id;
        localStorage.removeItem(key);
    })

    //delete button in archive list element

    $(document).on('click', '.deleteArch', function () {
        $(this).closest('[data-role=collapsible]').remove();
        var key = this.id;
        localStorage.removeItem(key);
    })

    // archive button in list element //

    $(document).on('click', '.archive', function (){
        $(this).closest('[data-role=collapsible]').remove();
        var key = this.id

        var ankerItem = new Object();
        var outKey = "";
        var outDate = "";
        var outTime = "";
        var outLatitude = "";
        var outLongitude = "";
        var outDescription = "";
        var outArchived = "";

        /** get Datarecord you want to change */
        ankerItem = JSON.parse(localStorage.getItem(key));
```

```
        outKey = ankerItem.timestamp;
        outDate = ankerItem.dateString;
        outTime = ankerItem.timeString;
        outLatitude = ankerItem.latitude;
        outLongitude = ankerItem.longitude;
        outDescription = ankerItem.description;
        outArchived = true;
        ankerItem.archived = true;

        // write the modified data

        localStorage.setItem(outKey, JSON.stringify(ankerItem));

        var contentArch = "<div data-role='collapsible' id=" + outKey + "'><h3>"+ outDate + " - "+
outTime +"</h3><table><tr><td>Datum: </td><td>"+ outDate +"</td></tr><tr><td>Zeit: </td><td>"+ outTime
+"</td></tr><tr><td>Beschreibung: </td><td>"+ outDescription +"</td></tr></table><br><iframe
src='http://maps.google.com/maps?q="+outLatitude+","+outLongitude+"&z=15&output=embed'></iframe><form><
input type='button' data-inline='true' value='Löschen' id=" + outKey + " class='deleteArch' data-
icon='delete'></form></div>";
        $( "#setArch" ).append( contentArch );//.collapsibleset( "refresh" );

    });

    // read data from local storage

    $(document).on('click', '#listview', function () {

        // change page
        $.mobile.changePage("#pageone");

        //clear already existing list to avoid double entrys
        $('[data-role=collapsibleset]').empty();

        var ankerItem = new Object();
        var outKey = "";
        var outDate = "";
        var outTime = "";
        var outLatitude = "";
        var outLongitude = "";
        var outDescription = "";
        var outArchived = "";
        var nextId = 1;
        var nextIdArch = 1;

        // get data from local storage

        for (var key in localStorage){
            ankerItem = JSON.parse(localStorage.getItem(key));
            outKey = ankerItem.timestamp;
            outDate = ankerItem.dateString;
            outTime = ankerItem.timeString;
            outLatitude = ankerItem.latitude;
            outLongitude = ankerItem.longitude;
            outDescription = ankerItem.description;
            outArchived = ankerItem.archived;


            // List entry is not archived -> write in activ list
```

```
            if (outArchived == false) {

                var content = "<div data-role='collapsible' id=set" + outKey + "'><h3>"+ outDate + " -
"+ outTime +"</h3><table><tr><td>Datum: </td><td>"+ outDate +"</td></tr><tr><td>Zeit: </td><td>"+
outTime +"</td></tr><tr><td>Beschreibung: </td><td>"+ outDescription +"</td></tr></table><br><iframe
src='http://maps.google.com/maps?q="+outLatitude+","+outLongitude+"&z=15&output=embed'></iframe><form><
input type='button' data-inline='true' value='Löschen' id=" + outKey + " class='delete' data-
icon='delete'><input type='button' data-inline='true' value='Archivieren' id=" + outKey + "
class='archive' data-icon='check'></form></div>";
                $( "#set" ).append( content );
                }

            // List entry is archived -> write in archived list
            else {

                var contentArch = "<div data-role='collapsible' id=" + outKey + "'><h3>"+ outDate + "
-  "+ outTime +"</h3><table><tr><td>Datum: </td><td>"+ outDate +"</td></tr><tr><td>Zeit: </td><td>"+
outTime +"</td></tr><tr><td>Beschreibung: </td><td>"+ outDescription +"</td></tr></table><br><iframe
src='http://maps.google.com/maps?q="+outLatitude+","+outLongitude+"&z=15&output=embed'></iframe><form><
input type='button' data-inline='true' value='Löschen' id=" + outKey + " class='deleteArch' data-
icon='delete'></form></div>";
                $( "#setArch" ).append( contentArch );
                }
        }
        $( "#set" ).collapsibleset( "refresh" );
    });

    // read data from local storage

    for (var key in localStorage){
        var outputList = JSON.parse(localStorage.getItem(key));
        console.log(key);
    };

    $(document).on('click','#archive', function() {
        $( "#setArch" ).collapsibleset( "refresh" );
    });


}

window.onload = function(){

    setUpEvents();

}
```