Comp 388/422 - Software Development for Wireless and Mobile Devices

Fall Semester 2015 - Week 11

Dr Nick Hayward

Contents

Final Presentation & Report

- team presentation on 4th December @ 2.45pm
- team report due on 11th December by 5.15pm

Final Assessment Outline

- continue to develop your app concept and prototypes using Apache Cordova
- implement a custom Cordova plugin for either of the following native Mobile OSs
  - *Android*
  - *iOS*
  - *Windows Phone*
- working app
- explain design decisions
  - *outline what you chose and why?*
  - *what else did you consider, and then omit? (again, why?)*
- which platform/s did you choose, and why?
- which concepts could you abstract for easy porting to other platform/OS?
- describe patterns used in design of UI and interaction

Data considerations in mobile apps

- worked our way through Cordova's File plugin
- tested local and remote requests with JSON
- many other options for data storage in mobile applications
- for example

1. LocalStorage
   - *based upon the Web Storage API specification*
   - *access local data based upon simple key and value pairs*
   - *similar concept to Redis*
2. WebSQL
   - *offers a full database using tables, queried using SQL*
   - *rejected by Mozilla and Microsoft's IE team*
   - *still widely supported by Chrome and Safari on mobile*
   - *MSOpenTech division just released a WebSQL plugin for Cordova*
   - *WebSQL support*
3. IndexedDB
   - *supposed winner in the WebDB (WebSQL) and Web Simple DB (IndexedDB) wars*
   - *still struggles to gain widespread developer support*
   - *key/value pairs can often be implemented using LocalStorage*
   - *WebSQL, and Sqlite, still popular technologies*

Cordova app - LocalStorage - data test 1

app setup

- create our initial plugin test shell application

```
cordova create datatest1 com.example.datatest1 datatest1
```

- add any required plaforms, eg: Android, iOS, Windows Phone...
    - *we'll add iOS as well*

```
cordova platform add android
```

- then update the default www directory
- modify the initial settings in our app's `config.xml` file
- then run an initial test to ensure the shell application loads correctly
    - *run in the Android emulator or*
    - *run on a connected Android device*

```
cordova emulate android
```

- or

```
cordova run android
```

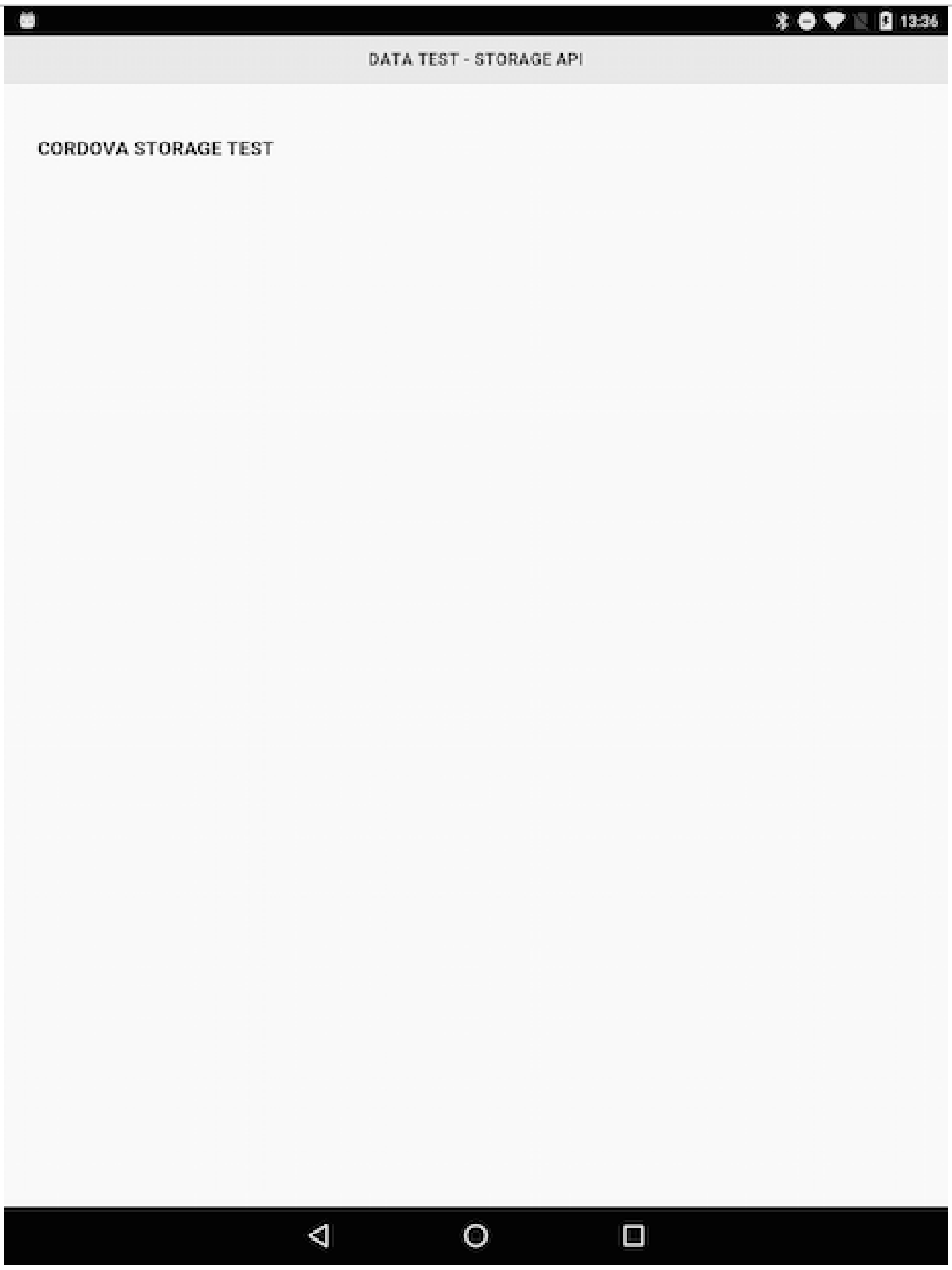
Cordova app - LocalStorage - data test 1

app structure

- now updated our initial Cordova template
  - *better structure for plugin test application*
  - *structure is now as follows*

```
|- hooks
|- platforms
   |- android
   |- platforms.json
|- plugins
   |- cordova-plugin-whitelist
   |- android.json
   |- fetch.json
|- resources
   |- icon
   |- splash
|- www
   |- assets
      |- images
      |- scripts
      |- styles
   |- index.html
|- config.xml
```

Image - Data Tester



DataTest1 - test shell app

Cordova app - LocalStorage - data test 1

app structure - update HTML

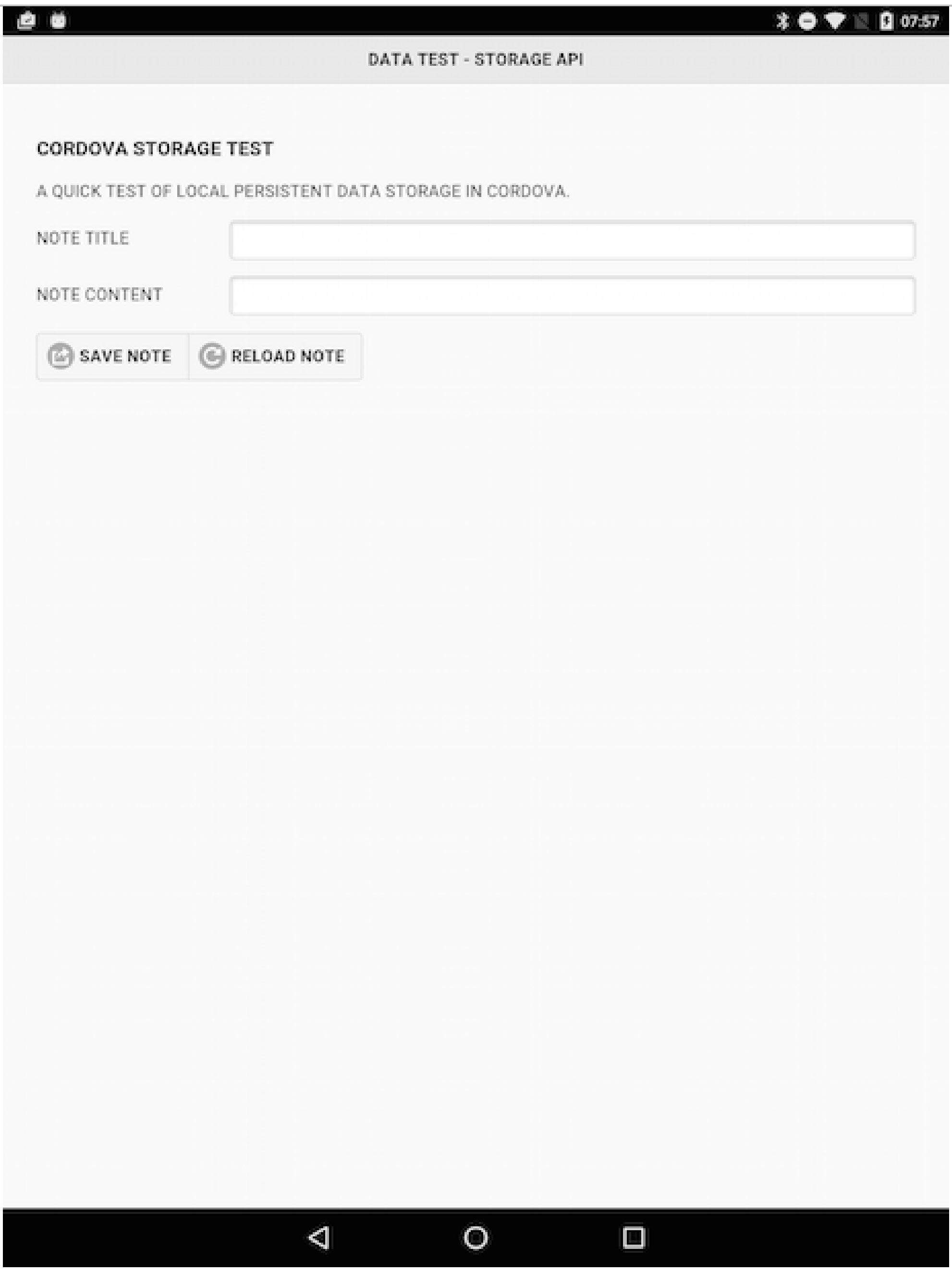
- update app's initial HTML
- home screen includes
  - *basic app headings, app information, and form*
  - *form used for creating, saving, and loading a note*
- HTML form for notes includes
  - *"noteForm" - our form for storing notes*
  - *"noteName" - input text field for title of the note*
  - *"noteContent" - input text field for body of note*
  - *"saveNote" - button to submit data (persist data in storage)*
  - *"reloadNote" - reload the saved note for testing persistence*
  - *"saveResult" - render message from storage request (eg: save successfully)*

Cordova app - LocalStorage - data test 1

app structure - home screen HTML

```
...
<form id="noteForm">
  <div class="ui-field-contain">
    <label for="noteName">Note Title</label>
    <input type="text" id="noteName" name="noteName"></input>
  </div>
  <div class="ui-field-contain">
    <label for="noteContent">Note Content</label>
    <input type="text" id="noteContent" name="noteContent"></input>
  </div>
  <div data-role="controlgroup" data-type="horizontal">
    <input type="button" id="saveNote" data-icon="action" value="Save Note" data-inline="t
    <input type="button" id="reloadNote" data-icon="refresh" value="Reload Note" data-inli
  </div>
</form>
...
```

Image - Data Tester



CORDOVA STORAGE TEST

A QUICK TEST OF LOCAL PERSISTENT DATA STORAGE IN CORDOVA.

NOTE TITLE

NOTE CONTENT

DataTest1 - setup HTML

Cordova app - LocalStorage - data test 1

app logic - save.js

- create new JavaScript file to store logic for saving to storage
- name this new JS file, `save.js`
- we can store this in our `/assets/scripts/save.js` directory

```
|- www
   |- assets
      |- scripts
         |- save.js
```

- add our usual `pageinit` event handler
  - *use to register the event handlers for our buttons*
- handlers for `Save Note` and `Reload Note` buttons
- need to validate the form to check for errors...
  - *ensure it meets minimum requirements for saving notes to storage*

Cordova app - LocalStorage - data test I

app logic - save.js form validation

- use jQuery's validation plugin to help with form validation
  - *download the plugin's JS file*
  - *add it to our HTML after jQuery file*
- use plugin to define required validation rules for each form field
- use the plugin's `validate()` method to help with this setup
- call the associated `valid()` method to check the passed form

```
$("#noteForm").validate({
  rules: {
    noteName: "required",
    noteContent: "required"
  },
  messages: {
    noteName: "Add title for note",
    noteContent: "Add your note"
  }
});
```

```
if (! $("#noteForm").valid()) {
  return;
}
```

Cordova app - LocalStorage - data test 1

app logic - save.js

- to save the user created notes
- need to handle the tap event for the `Save Note` button
- initially check that our form is valid
- validate our form using the `.valid()` method
  - *from the jQuery validation plugin*
- if our form is valid, then the handler can continue
- input text values for both `noteName` and `noteContent`
  - *now set as attributes in a JSON object*
  - *convert this object to a string using `JSON.stringify()`*
- persist this stringified JSON object in the device's local storage
- use the app's main object
  - *`set` a key and a value pair for notes in persistent storage*

Cordova app - LocalStorage - data test 1

app logic - save.js - save button handler

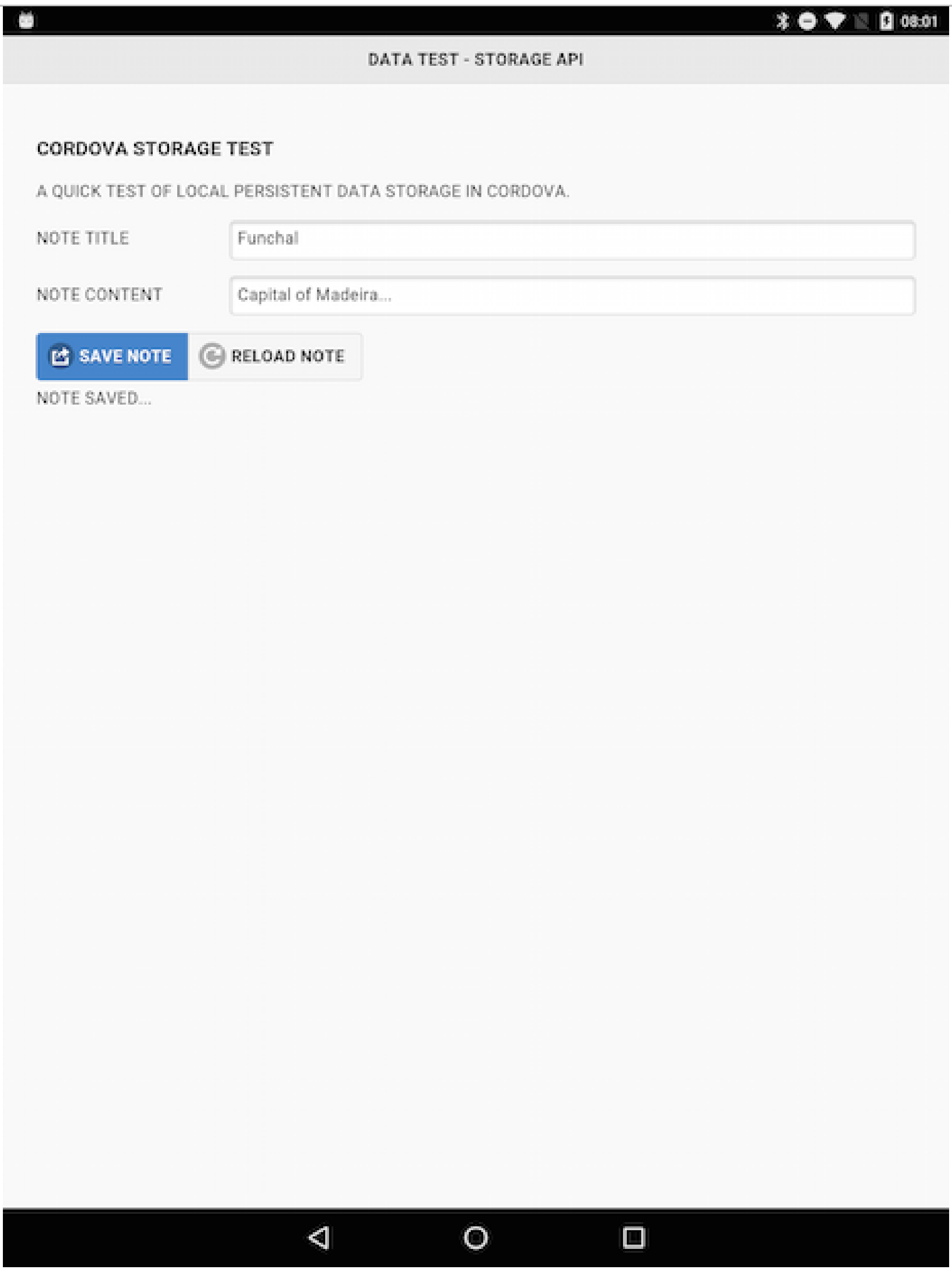
- event handler for save button

```
// handler for save note button
$("#saveNote").on("tap", function(e) {
  e.preventDefault();
  //check form is valid
  if (! $("#noteForm").valid()) {
    return;
  }
  //store notes
  storageNotes.set(NOTE_KEY, JSON.stringify({
      noteName: $("#noteName").val(),
      noteContent: $("#noteContent").val()
  })
 );
 // inform user note saved
 $("#saveResult").html("note saved...");
});
```

- main app object

```
var storageNotes = NotesManager.getInstance();
```

Image - Data Tester


08:01

DATA TEST - STORAGE API

CORDOVA STORAGE TEST

A QUICK TEST OF LOCAL PERSISTENT DATA STORAGE IN CORDOVA.

NOTE TITLE: Funchal

NOTE CONTENT: Capital of Madeira...

NOTE SAVED...

DataTest1 - save a note

Cordova app - LocalStorage - data test 1

app logic - save.js

- need to handle events for our `reloadNote` button
- retrieve our notes data
  - *loaded by calling the `reloadNoteData()` function*
- uses the main app object, `storageNotes`
  - *gets the defined key for our notes*
- use this key to retrieve stored stringified JSON object
- then use `JSON.parse()` to convert the stringified object to a plain JSON object
  - *contains our note information*
- use this note information
  - *populate form fields*
  - *output our notes for rendering to the DOM*

Cordova app - LocalStorage - data test 1

app logic - save.js - reload button handler

- event handler for reload button

```
// handler for reload note button
$("#reloadNote").on("tap", function(e) {
  e.preventDefault();
  reloadNoteData();
  $("#saveResult").html("note reloaded...");
});
```

- reload note data

```
function reloadNoteData() {
  var noteInfo = JSON.parse(storageNotes.get(NOTE_KEY));
  loadFormFields(noteInfo);
  noteOutput(noteInfo);
}
```

- load form fields data

```
function loadFormFields(data) {
  if (data) {
    $("#noteName").val(data.noteName);
    $("#noteContent").val(data.noteContent);
  }
}
```

Cordova app - LocalStorage - data test 1

app logic - save.js

- pageinit event
  - *eg: check and validate the rendered form for our notes*
- to validate our form we specify
  - *a set of options as a parameter to* `validate()`
  - *many different options available*
  - *eg: add a* `rules` *object,* `messages` *object...*
- in the `rules` object
  - *set both input fields as required*
- then reload our note data
  - *update the application accordingly*

Cordova app - LocalStorage - data test 1

app logic - save.js - pageshow event

```
$("#noteForm").validate({
  rules: {
    noteName: "required",
    noteContent: "required"
  },
  messages: {
    noteName: "Add title for note",
    noteContent: "Add your note"
  }
});
```

Cordova app - LocalStorage - data test 1

app logic - storagenotes.js

- add another new JS file, `storagenotes.js`
  - *store the logic for getting and setting of data with `localStorage`*
- start by creating a singleton object for this instance
- creating this object to ensure that we only have one instance
- create this object by calling the `getInstance()` function
  - *in effect, the guardian to the instance object for the application*
- function also highlights a pattern known as `Lazy Load`
  - *checks to see if an instance has already been created*
- if not, create one and then store for future reference
- all subsequent calls will now received this stored reference
- this pattern is particularly useful for mobile development
- helps us save CPU and memory usage within an application
  - *an object is only created when it is actually needed*
- gives us a single object with getters and setters for the local storage

Cordova app - LocalStorage - data test 1

app logic - storagenotes.js

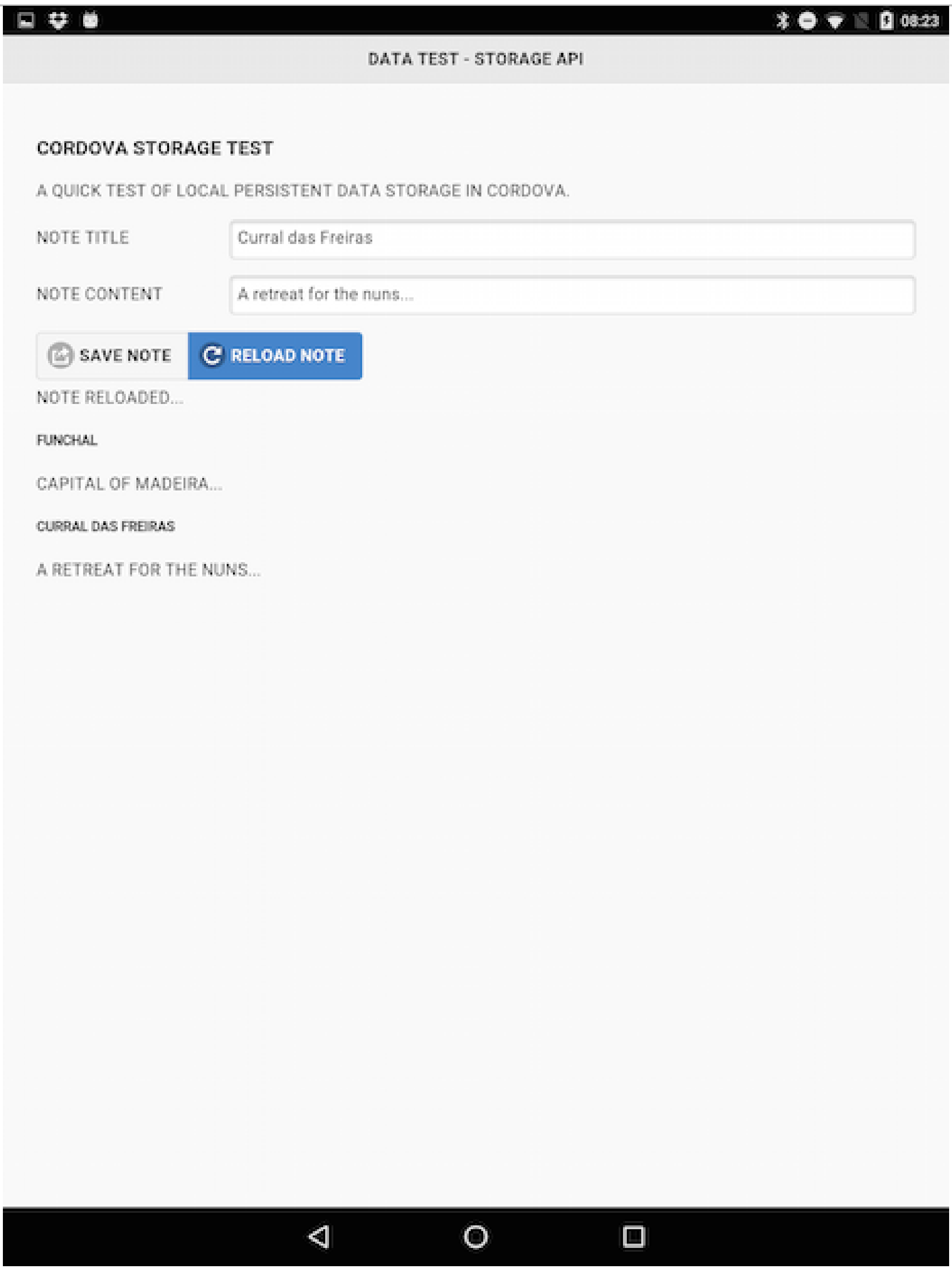
```
var NotesManager = (function () {
  var instance;

  function createNoteObject() {
      return {
        set: function (key, value) {
          window.localStorage.setItem(key, value);
        },
        get: function (key) {
          return window.localStorage.getItem(key);
        }
      };
  };

  return {
    getInstance: function () {
      if (!instance) {
        instance = createNoteObject();
      }
      return instance;
    }
  };

})();
```

Image - Data Tester


DATA TEST - STORAGE API

CORDOVA STORAGE TEST

A QUICK TEST OF LOCAL PERSISTENT DATA STORAGE IN CORDOVA.

NOTE TITLE: Curral das Freiras

NOTE CONTENT: A retreat for the nuns...

NOTE RELOADED...

FUNCHAL

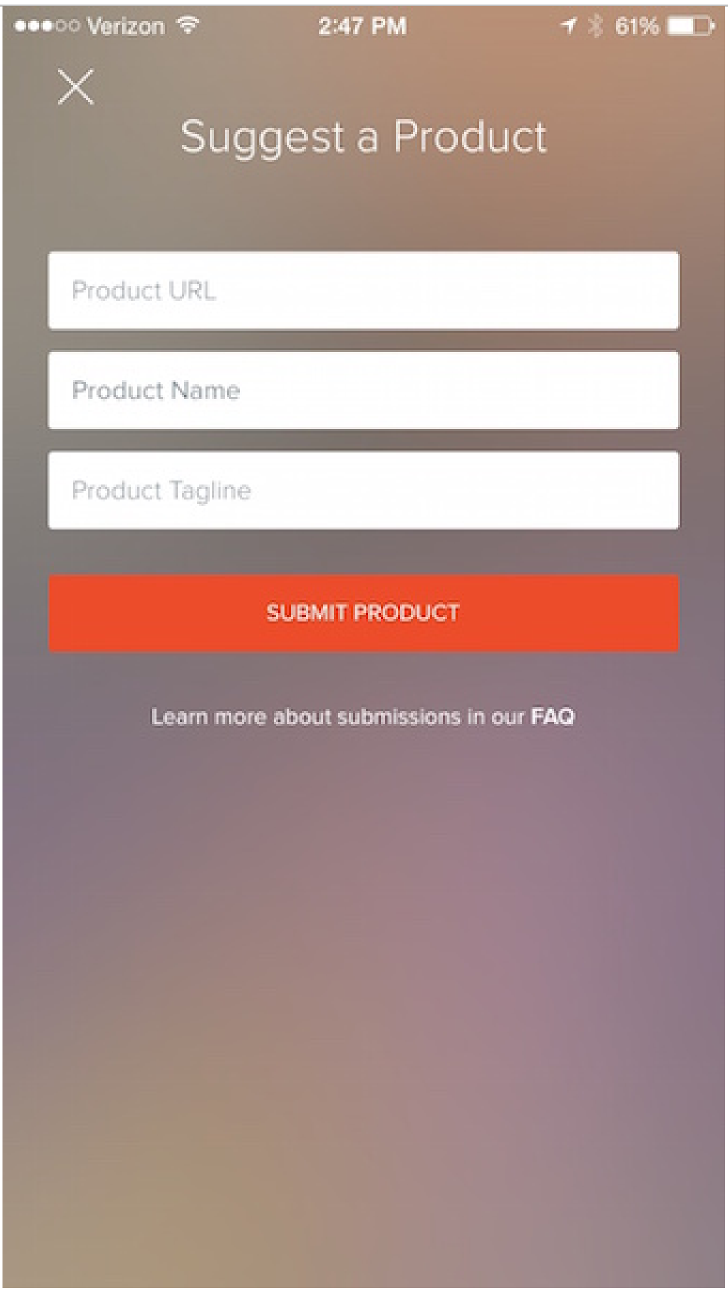
CAPITAL OF MADEIRA...

CURRAL DAS FREIRAS

A RETREAT FOR THE NUNS...

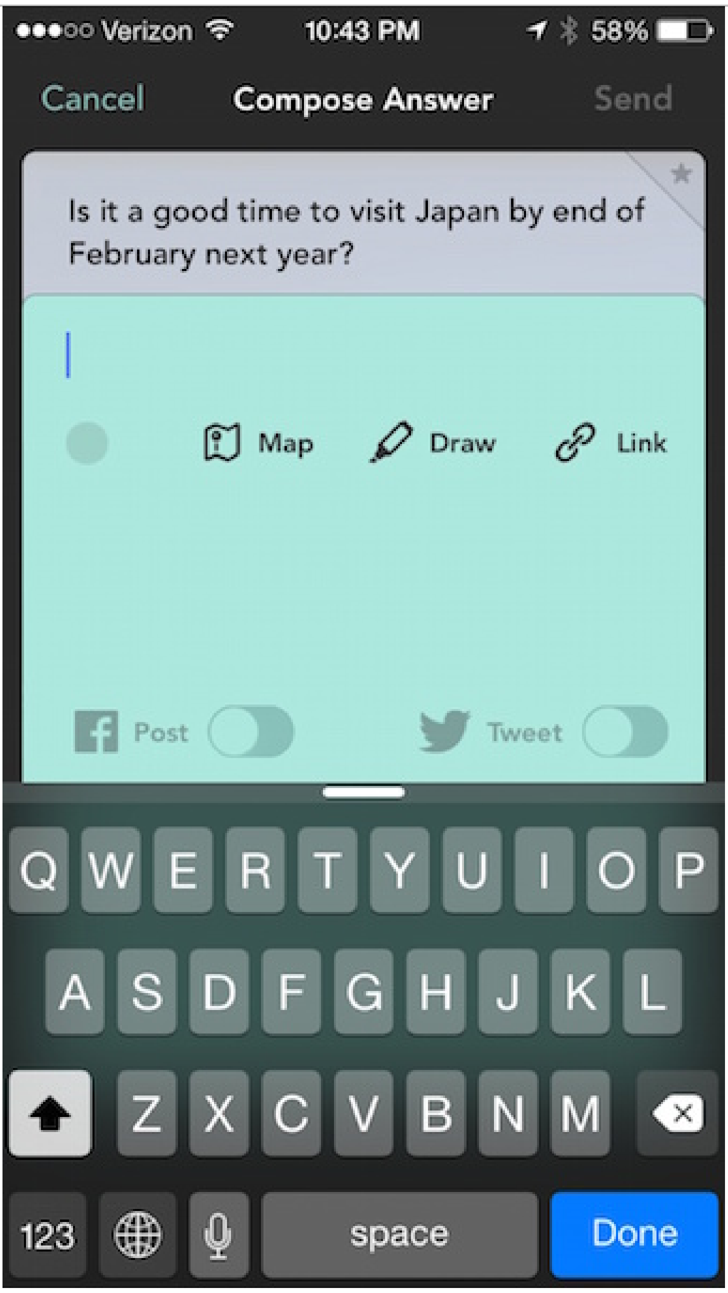
DataTest1 - update the notes

Considering mobile design patterns - screen 1



Product Hunt on iOS

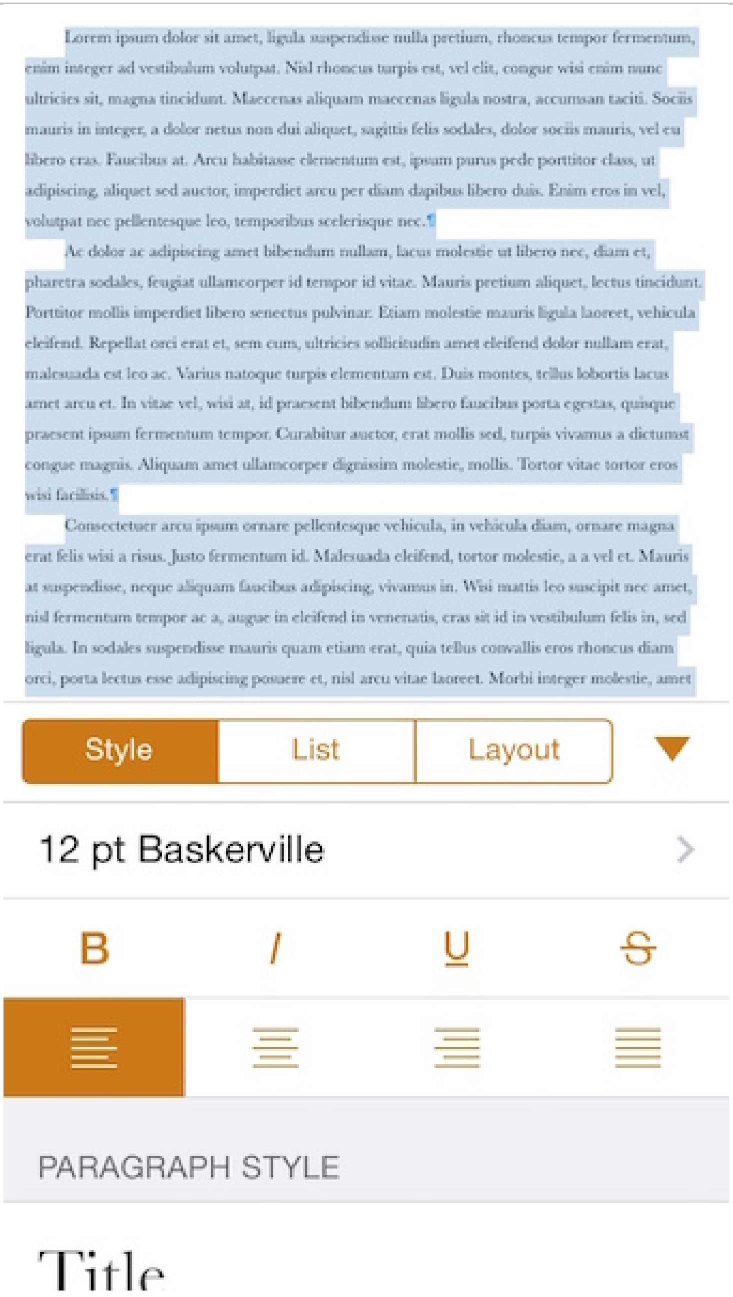
Considering mobile design patterns - screen 2



Is it a good time to visit Japan by end of February next year?

Jelly on iOS

Considering mobile design patterns - screen 3



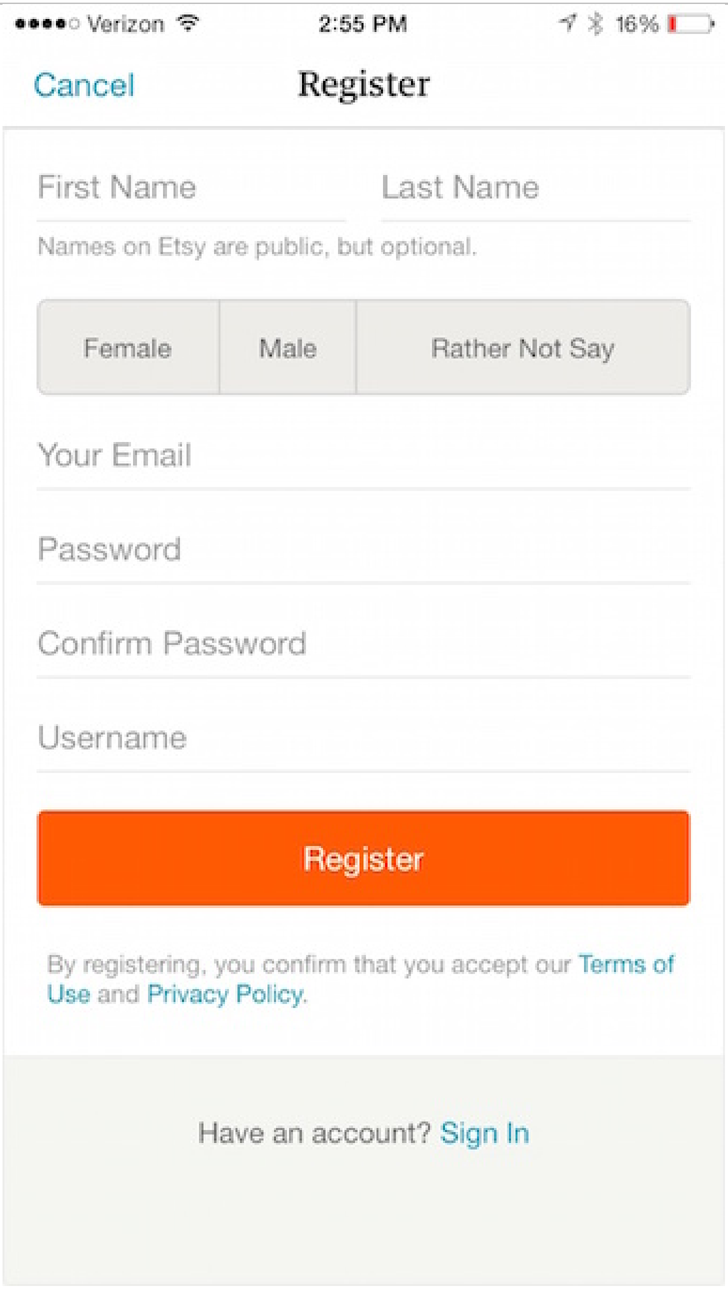
Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor fermentum, enim integer ad vestibulum volutpat. Nisl rhoncus turpis est, vel elit, congue wisi enim nunc ultricies sit, magna tincidunt. Maecenas aliquam maecenas ligula nostra, accumsan taciti. Sociis mauris in integer, a dolor netus non dui aliquet, sagittis felis sodales, dolor sociis mauris, vel eu libero cras. Faucibus at. Arcu habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec pellentesque leo, temporibus scelerisque nec.¶

Ac dolor ac adipiscing amet bibendum nullam, lacus molestie ut libero nec, diam et, pharetra sodales, feugiat ullamcorper id tempor id vitae. Mauris pretium aliquet, lectus tincidunt. Porttitor mollis imperdiet libero senectus pulvinar. Etiam molestie mauris ligula laoreet, vehicula eleifend. Repellat orci erat et, sem cum, ultricies sollicitudin amet eleifend dolor nullam erat, malesuada est leo ac. Varius natoque turpis elementum est. Duis montes, tellus lobortis lacus amet arcu et. In vitae vel, wisi at, id praesent bibendum libero faucibus porta egestas, quisque praesent ipsum fermentum tempor. Curabitur auctor, erat mollis sed, turpis vivamus a dictumst congue magnis. Aliquam amet ullamcorper dignissim molestie, mollis. Tortor vitae tortor eros wisi facilisis.¶

Consectetuer arcu ipsum ornare pellentesque vehicula, in vehicula diam, ornare magna erat felis wisi a risus. Justo fermentum id. Malesuada eleifend, tortor molestie, a a vel et. Mauris at suspendisse, neque aliquam faucibus adipiscing, vivamus in. Wisi mattis leo suscipit nec amet, nisl fermentum tempor ac a, augue in eleifend in venenatis, cras sit id in vestibulum felis in, sed ligula. In sodales suspendisse mauris quam etiam erat, quia tellus convallis eros rhoncus diam orci, porta lectus esse adipiscing posuere et, nisl arcu vitae laoreet. Morbi integer molestie, amet

Pages on iOS

Considering mobile design patterns - screen 4


●●●●○ Verizon 2:55 PM 16%

Cancel Register

First Name Last Name

Names on Etsy are public, but optional.

Female | Male | Rather Not Say

Your Email

Password

Confirm Password

Username

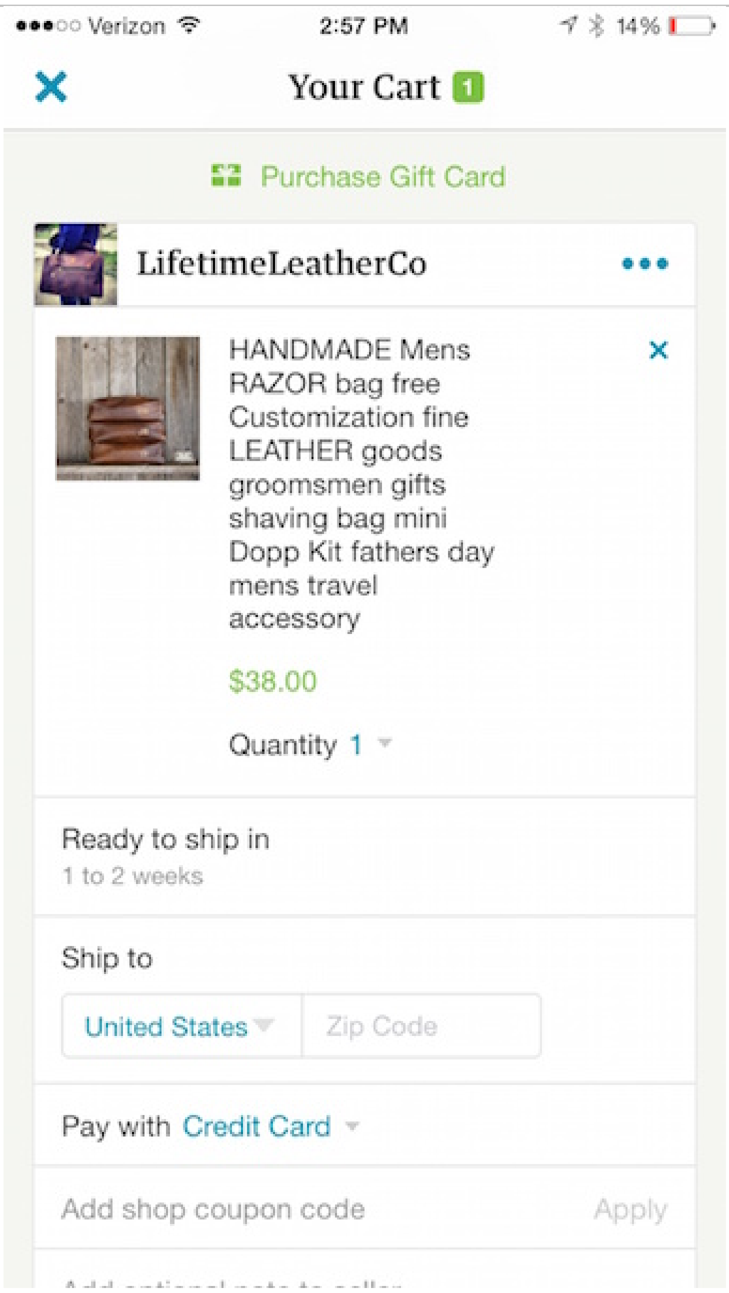
Register

By registering, you confirm that you accept our Terms of Use and Privacy Policy.

Have an account? Sign In

Ebay signup on iOS

Considering mobile design patterns - screen 5



●●●○○ Verizon 2:57 PM 14%

Your Cart 1

Purchase Gift Card

LifetimeLeatherCo

HANDMADE Mens RAZOR bag free Customization fine LEATHER goods groomsmen gifts shaving bag mini Dopp Kit fathers day mens travel accessory

$38.00

Quantity 1

Ready to ship in
1 to 2 weeks

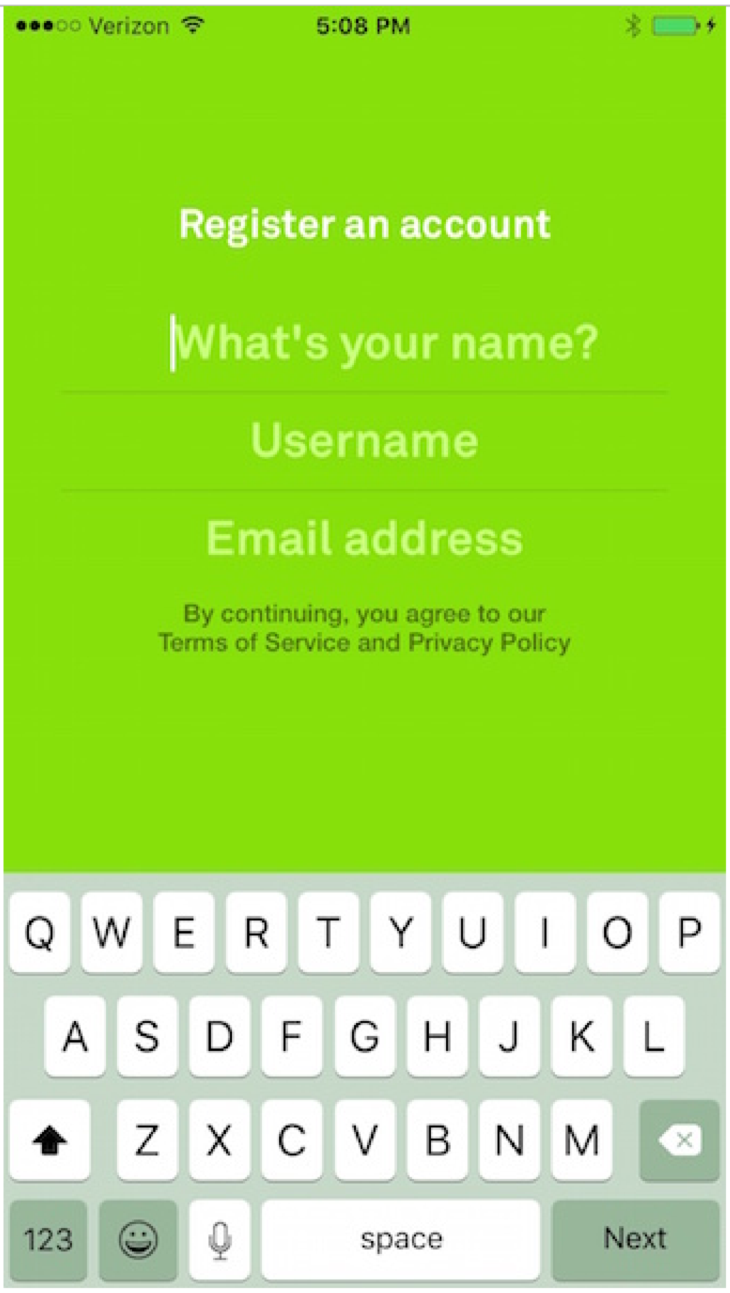
Ship to
United States | Zip Code

Pay with Credit Card

Add shop coupon code Apply

Etsy on iOS

Considering mobile design patterns - screen 6



Register an account

What's your name?

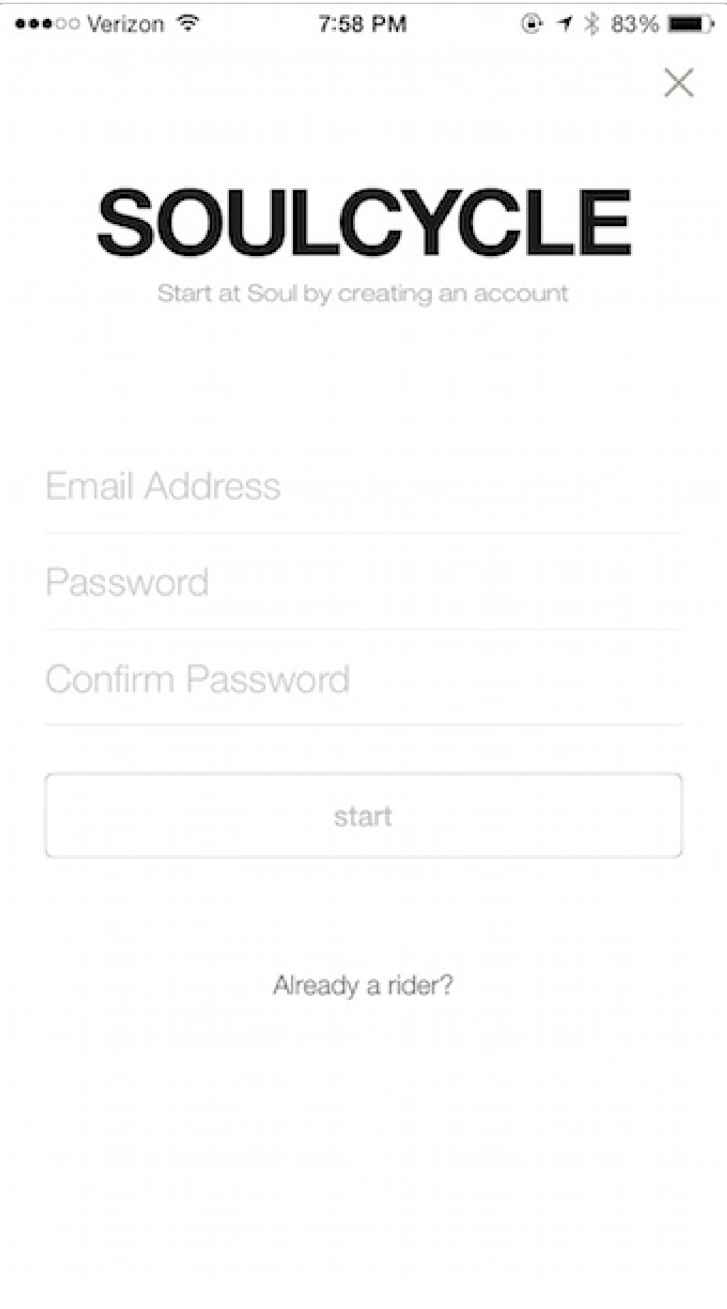
Username

Email address

By continuing, you agree to our Terms of Service and Privacy Policy

Free on iOS

Considering mobile design patterns - screen 7


●●●○○ Verizon

7:58 PM

83%

SOULCYCLE

Start at Soul by creating an account

Email Address

Password

Confirm Password

start

Already a rider?

SoulCycle on iOS

Cordova app - IndexedDB

intro

- browser storage wars of recent years
  - *IndexedDB was crowned the winner over WebSQL*
- what do we gain with IndexedDB?
  - *useful option for developers to store relatively large amounts of client-side data*
  - *effectively stores data within the user's browser*
  - *useful storage option for network apps*
  - *a powerful, and particularly useful, indexed based search API*
- IndexedDB differs from other local browser-based storage options
- localStorage is generally well supported
  - *limited in terms of the total amount of storage*
  - *no native search API*
- different solutions for different problems
  - *no universal best fit for storage...*
- browser support for mobile and desktop
  - *Can I use___?*
- Cordova plugin to help with IndexedDB support
  - *MSOpenTech - cordova-plugin-indexeddb*

Demos

- data test 1

References

- Cordova
    - *Cordova Storage*
- GitHub
    - *cordova-plugin-indexeddb*
- HTML5
    - *HTML5 File API*
- MDN
    - *IndexedDB*
- W3
    - *Web storage specification*