COGNITIVE DEFUSION MOBILE APPLICATION

Developing a single page application for cognitive defusion exercises

Bachelorthesis
Submitted to Prof. Oscar Nierstrasz
7.5.2015

Pascal Zaugg
pascal.zaugg@students.unibe.ch

Table of contents

1 Technologies

(overview illustration created by myself)

1.1 Sinatra

Sinatra is “a DSL for quickly creating web applications in Ruby with minimal effort” (Sinatra, kein Datum). For our case this was ideal because the only thing we wanted to do was to create a very simple form of a questionnaire and store the answers into the CouchDB. There was no need for a more heavyweight framework like RubyOnRails.

(illustration of Sinatra working with client and ruby code, created by myself)

1.2 CouchDB

CouchDB is a NOSQL database developed by Adbobe which stores data as JSON-Documents. A big advantage of this database is that there exist a client called PouchDB which abstracts away the automatic synchronization between the local database on smartphones and the database on the server. Thus its possible to store data on the local machine and as soon as it connects to the internet synchronize it with the database on the remote side.

1.3 PouchDB

1.4 Ionic

Ionic is a framework to build mobile apps with HTML5. Ionic heavily depends on AngularJS. By its developer it is called “a powerful HTML5 native app development framework that helps you build native-feeling mobile apps all with web technologies like HTML, CSS, and Javascript.” (Ionic, n.d.) and its goal is “to make it easier to develop native mobile apps with HTML5, also known as Hybrid apps” (Adam, 2013).

A big advantage of the Ionic Framework is that is firstly provides a large set of prebuild css styles which make your app immediately look good. And implements concepts widely used in mobile applications like side menus and header menus.

1.5 AngularJS

AngularJS is a JavaScript Framework developed by Google™. It enhances static HTML to create dynamic websites.

Key features of AngularJS

- Declarative HTML approach
- Two way data binding
- Encourages MVC Design Pattern
- Easy unit testing
- Templating
- Routing
- Concepts like injector, directive and scope

Slightly altered from (Shekhawat, 2013, p. 5)

AngularJS creates a highly dynamic html webpage. At page startup it waits for the DOM content loaded event. As soon as this events is emitted an injector is created. Then the webpage is compiled and displayed in the browser.

(Startup UML image) (Eyal, 2013, p. 2)

1.5.1 MVC Design Pattern

The model is defined in plain JavaScript. AngularJS encourages people to use the build-in service constructor to create and access the model

(downsized example from my code: User Model)

1.5.2 The view is defined in HTML5 with a powerful addition, the declaratives (see 1.5.5 Routing

In standard AngularJS routing is done with the build in $routeProvider . Because Ionic uses a more advanced routing called UI-Router that uses a state machine the $routeProvider will not be covered here. Instead we'll describe the concepts of the UI-Router.

Directives).

Controller are defined via the angular variable

(example)

After a controller was defined AngularJS lets you assign controllers to parts of the view. Controllers maybe reused for other parts.

(example)

1.5.3 Injector

The injector provides a lookup for services and objects. Dependencies are resolved by the injector when using the invoke method on it.

```
//create a module
var myModule = angular.module(‘myModule’, [])

//create a factory in that module
myModule.factory(‘serviceA’, function() {
      return “Hello World!”
});

//write a function
function hello(serviceA) {
      console.log(serviceA)
}

//retrieve injector
var $injector = angular.injector([‘myModule’]);

//let angularjs do the work
angular.invoke(hello)
```

altered from (Eyal, 2013, pp. 8-10)

1.5.4 Double binding

In AngularJS variables are watched. If a change happen to a variable in the model those changes are reflected in the view and vice versa. To make AngularJS clear that such a change happened one uses the $apply method.

(example from my code)

($apply, $digest life cyle from) (Eyal, 2013, p. 28)

As soon the $apply method is called. The $digest cycles begins. It means that all watched variables are reevaluated and associated functions are called. One $apply cycle may consist multiple $digest cycles. (Eyal, 2013, pp. 28-33)

Bindings in the view are denoted by the double curly brakets

```
{{name}}
```

Name refers here to the name variable in the $scope variable of the controller. (AngularJS, n.d.)

```
$scope.name = “Pantalaimon”
```

1.5.5 Scope

A crucial part of AngularJS is the concept of the scope. It lets model, controller and view talk with each other without binding them to strongly together. A scope can be inherited. (AngularJS, n.d.). Each controller has his own scope. If controllers are nested the nested controller inherits the scope of its parent but not vice-versa. Variables defined in the child are not accessible in the parent even if they are called the same.

```
angular.controller(..
      $scope.greeting = “hey parent”
      // Has no access to the $scope.child variable

      //waits for a message from its child

      //prints always “hey parent”
)
angular.controller(..
      console.log($scope.greeting) //prints “hey parent”
      $scope.greeting = “hey child”
      console.log($scope.greeting) //prints “hey child”

      //sends message to parent to print its variable
      //and set it to “hey parent” again
      console.log($scope.greeting) // prints “hey child”
)
```

1.5.6 Routing

In standard AngularJS routing is done with the build in $routeProvider (AngularJS, n.d.). Because Ionic uses a more advanced routing called UI-Router that uses a state machine the $routeProvider will not be covered here. Instead we’ll describe the concepts of the UI-Router.

1.5.7 Directives

One of the most important concepts is that of a directive. Directives are added as tags to the html code. AngularJS has many build-in directives for example the ng-repeat directive which lets you iterate over a list an repeat parts of your html code

```
<html ng-app="myApp">
<head>
  <script src="angular.js"></script>
```

```
</head>
<body ng-controller="PhoneListCtrl">
  <ul>
    <li ng-repeat="number in [3, 2, 1, “take off!”]">
      <p>{{number}}</p>
    </li>
  </ul>
</body>
</html>
```

altered from (AngularJS, n.d.)

The above example results in something similar as

```
<html ng-app="myApp">
<head>
  <script src="angular.js"></script>
</head>
<body ng-controller="PhoneListCtrl">
  <ul>
    <li>
      <p>3</p>
    </li>
    <li>
      <p>2</p>
    </li>
    <li>
      <p>1</p>
    </li>

    <li>
      <p>take off!</p>
    </li>
  </ul>
</body>
</html>
```

In this project AngularJS was used internally by the Ionic Framework. The Ionic framework used it to create his own directives.

1.6 calabash-android

Calabash allows readable tests written in Gherkin. It expresses the software to be build in real examples and brings stakeholders and developers together.

“By using real-world examples to describe the desired behaviour of the system we want to build, we stay grounded in language and terminology that makes sense to our stakeholders: *we’re speaking their language*” (Wynne, 2012, p. 25)

(example from one of my tests)

This allows customer and developer to develop an ubiquitous language and to exchange requirements in a way both sides understand. Gherkin can be described as a DSL to describe a requirement and scenario of the application. Because this is a high level of abstraction gherkin is then bootstrapped to ruby. Each step in gherkin is then processed in ruby where it is wired to the model.

(Figure 3, page 41, (Wynne, 2012, p. 41)

(example of a step definition of my tests)

What calabash does is providing possibilities to communicate through gestures (touch, double-tab, pinch, etc.) with the application. It also provides a way to inject java-script code into the mobile application which allows one to manipulate runtime java-script objects.

(illustration „calabash in android) from (Xamarin, n.d.))

2 Literature

Adam. (2013, 10 28). *Where does the Ionic Framework fit in | The Official Ionic Blog* . Retrieved 5 7, 2015, from http://blog.ionic.io/where-does-the-ionic-framework-fit-in/

AngularJS. (n.d.). *AngularJS Tutorial 2 - Angular Templates*. Retrieved 5 5, 2015, from https://docs.angularjs.org/tutorial/step_02

AngularJS. (n.d.). *AngularJS: API: $rootScope.Scope*. Retrieved 5 6, 2015, from https://docs.angularjs.org/api/ng/type/$rootScope.Scope

Eyal, V. (2013, 5 12). *AngularJS architecture.* Retrieved 5 4, 2015, from http://de.slideshare.net/EyalV/angularjs-architecture

Ionic. (n.d.). *Ionic Documentation Overview - Ionic Framework*. Retrieved 5 6, 2015, from http://ionicframework.com/docs/overview/

Shekhawat, M. (2013, 6 14). *Introduction to Angularjs.* Retrieved 5 4, 2015, from http://de.slideshare.net/manishekhawat/angularjs-22960631

Xamarin. (n.d.). *Introduction to calabash - Xamarin*. Retrieved 5 7, 2015, from http://developer.xamarin.com/guides/testcloud/calabash/introduction-to-calabash/

[Unattributed] : https://angularjs.org/ [4.5.2015]