4.3 Die Webapp—Das Interface zur Einsicht und Konfiguration aller Daten

4.3.1 Erste Konzeption

Zur Kontrolle der Daten und aller Benutzerkonten haben wir uns entschieden, eine Web Applikation zu entwickeln, die Zugriff auf die Datenbank nehmen kann, um deren Informationen anzuzeigen / bearbeiten / löschen und ggf. neue Datensätze hinzufügen kann. Zudem soll dies dadurch vereinfacht werden indem die Applikation für einige Einträge Vorschläge für Werte macht bzw. ausfiltert, falls ungültige Eingaben getätigt werden. Die App soll verschiedene Sicherheitsbereiche besitzen, die nur Erreichbar sind, falls ein Nutzer angemeldet ist. Administrative Bereiche sollen bei gegebener Sicherheitsfreigabestufe (Eintrag ‚Rights‘ in der ‚Users‘ Tabelle der Datenbank) in die vorhandene Website eingebettet werden um unnötige Doppelungen von Bereichen zu vermeiden bzw. die Benutzung der App weiter zu erleichtern.

4.3.2 Ziele zur Bedienung und der Benutzeroberfläche

Der User soll ohne Anmeldung Zugriff auf die Bereiche ‚Home‘ und ‚Menu‘ haben. Dabei beinhaltet Home allgemeine Informationen und Neuigkeiten zur Seite, sowie sonstige Infos. Der Menübereich beinhaltet lediglich aktuelle Angaben zu verfügbaren Speisen ohne weiteren Zugriff in Form von Bestellungen oder Ähnliches. Sobald sich der User angemeldet hat, verschwindet der Login-Button und wird von einem User-Dropdown abgelöst, in dem der User Links zu den Bereichen ‚Notifications‘, ‚Leaderboard‘, ‚Settings‘, ‚Favorites‘ und den Logout erhält. Es wird sichergestellt, dass der Client keinen Zugriff auf Daten bekommt bevor dieser die benötigten Userdaten zum Anmelden abgeschickt hat. Im Notifications-Bereich kann der User seinen Kontostand, Nachrichten und letzte Einkäufe einsehen. Das Leaderboard dient nur Kosmetischen Unterhaltungszwecken und zeigt die Top 5 der Kilokalorien / Koffein Konsumenten, die dieses Feature aktiviert haben. In den Settings kann der User seine persönlichen Daten bearbeiten. Im Favorites-Tab kann er Favoriten aussuchen, die in der Schnellauswahl am Raspberry Pi angezeigt werden. Der Logoutbereich entfernt alle gespeicherten Userdaten vom Client und trennt die aktive Sitzung. Der Admin hat folgende zusätzliche Features: Es werden im Menü Admin-Bedienelemente eingebettet, er hat einen separaten Bereich, um Userstatistiken einzusehen (Layout ist ähnlich dem Menü), wo er diese auch bearbeiten / löschen, oder neue User hinzufügen kann.

4.3.3 Umsetzung

Zu den Bereichen der Entwicklung gehören zum einen die Grundsprachen HTML5, CSS, JavaScript PHP und SQL, jedoch werden weitere Frameworks, Plugins und Erweiterungen benutzt, wie: jQuery-1.11.2, Mustache v2.1.2, Bootstrap-3.3.4, AngularJS v1.4.3-build.4096, Angular-UI-Router v0.2.15 und UI-Bootstrap-tpls-0.13.0.

In der ersten Version der Webapp, wurde ein Grundgerüst mit HTML5 CSS und Bootstrap erstellt. Der Fokus lag dabei beim Design und der Anordnung aller Elemente, die leichte Bedienbarkeit und eine selbsterklärende Grafische Benutzeroberfläche. Da der RaspberryPi sich möglicherweise innerhalb eines Intranetzes, ohne Internetzugriff befindet, muss dafür gesorgt werden dass die Libraries auch offline geladen werden können.

Hier ein Code-Auszug:

\begin{frame}

\begin{

<html ng-app="app">

<head>

<title>Tally C&amp;S</title>

<!-- Latest compiled and locally saved CSS bootstrap -->

<link rel="stylesheet" href="styles/bootstrap-3.3.4-dist/css/bootstrap.css">

<!-- Optional theme (locally saved) -->

<link rel="stylesheet" href="styles/bootstrap-3.3.4-dist/css/bootstrap-theme.css">

</head>

<body class="color-bg">

<!-- Plugins and Services -->

<!-- jQuery (necessary for Bootstrap's JavaScript plugins. Saved on Drive) -->

<script src="scripts/jquery-1.11.2.js"></script>

<!-- Latest compiled and locally saved JavaScript from Bootstrap -->

<script src="styles/bootstrap-3.3.4-dist/js/bootstrap.js"></script>

<!-- <script src="mustache/mustache.js"></script> -->

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

<script src="angular-ui-router/build/angular-ui-router.js"></script>

<script src="ui-bootstrap/ui-bootstrap-tpls-0.13.0.js"></script>

<!-- End Plugins and Services -->

</body>

</html>

Um ein breiteres Spektrum zur DOM Traversal zu bekommen wurde jQuery in das Projekt eingebunden. Event-Listeners und Callbacks für z.B. Buttons sind somit einfach zu realisieren. Hier ist ein Codebeispiel für einen einfachen Button und ein Event-Listener in jQuery:

4.3.4 Probleme / Schwierigkeiten

4.3.5 Verbesserungen / Änderungen für die Zukunft