

UNIVERSITATEA POLITEHNICA TIMIȘOARA

LUCRARE DE LICENȚĂ

**Bogart - Aplicație hibridă pentru
rezervarea online a meselor din
restaurante**

Autor:

Laura-Iasmina NEAGA

Coordonator Științific:

conf. dr. ing. Oana
AMARICAI-BONCALO

Facultatea de Automatică și Calculatoare

Departamentul de Calculatoare și Tehnologia Informației



October 27, 2020

Contents

1	Aplicații hibride. Analiza stadiului actual în domeniul problemei	1
1.1	Context	1
1.2	Aplicații existente pe piață	1
1.2.1	Stadiul actual	1
1.3	Structura aplicației	1
1.3.1	Arhitectura	1
1.4	Scopul și obiectivul lucrării	1
2	Tehnologii utilizate în dezvoltarea aplicației	3
2.1	Frameworks	3
2.1.1	Ionic	3
2.1.2	Angular	4
2.1.2.1	Componente	4
2.1.2.2	Servicii	4
2.2	Limbaje de programare	4
3	Implementare	5
3.1	Navigation	5
3.1.1	Lazy-routing	5
3.1.2	Tabs	5
3.2	Main Section 2	5
A	Frequently Asked Questions	7
A.1	How do I change the colors of links?	7
	Bibliography	9

List of Figures

List of Tables

1.1	Compararea specificațiilor pentru cele mai populare aplicații din domeniu	2
-----	---	---

List of Abbreviations

LAH List Abbreviations **Here**
WSF What (it) Stands For

Chapter 1

Aplicații hibride. Analiza stadiului actual în domeniul problemei

1.1 Context

1.2 Aplicații existente pe piață

1.2.1 Stadiul actual

În tabelul 1.1 ... Wieman and Hollberg, 1991

1.3 Structura aplicației

1.3.1 Arhitectura

SPA - Single Page Application

why? <https://blog.angular-university.io/why-a-single-page-application-what-are-the-benefits-what-is-a-spa/>

1.4 Scopul și obiectivul lucrării

TABLE 1.1: Compararea specificațiilor pentru cele mai populare aplicații din domeniu

Specificații	OpenTable	Tock	Resy	I Know the Chef	Yelp Reservations	ialoc
Gestionarea angajaților	✓	-	✓	-	✓	-
Inventariat	✓	-	✓	-	-	-
Gestionarea bucătăriei	-	-	-	-	-	-
Gestionarea meniului	-	-	✓	-	-	-
Gestionarea rezervărilor	✓	✓	✓	✓	✓	✓
Gestionarea meselor	✓	✓	✓	✓	✓	✓
Înscrierea pe o listă de așteptare	✓	✓	✓	✓	✓	-
Integrare POS	✓	✓	✓	✓	-	-
Client Web	✓	✓	✓	✓	✓	✓
Client Android	✓	✓	✓	-	✓	✓
Client iOS	✓	✓	✓	✓	✓	✓
Crearea unui profil personalizat	-	-	-	✓	-	-
Gratuitate pentru utilizatori	✓	✓	✓	-	✓	✓
Cupoane cadou	✓	✓	✓	-	✓	-
Notă	4,8	3,5	4,3	4,7	3,9	4,8

Chapter 2

Tehnologii utilizate în dezvoltarea aplicației

2.1 Frameworks

=e un set de reguli și tool-uri de utilitate pe care le putem folosi în codul JS sau TS pt a ne face viața mai simplă și care se vor ocupa de lucruri ca cereri HHTP, rutare sau redarea paginilor

2.1.1 Ionic

Ionic se referă la a avea un codebase pentru o aplicație web, folosind HTML, JavaScript și CSS -deci ceea ce folosești pt a crea un UI normal-, și folosești aceste tool-uri și tehnici pentru a crea aplicații web progresive, aplicații Android sau iOS și chiar aplicații desktop electron. Electron este o tehnologie care îți permite să împachetezi o aplicație web într-una desktop. Deci poți crea o aplicație pentru platforme diferite cu un singur cod de bază, cu mici adaptări.

Aplicație Web Progresivă

o aplicație web care se comportă ca o aplicație mobile nativă. Nu e una, nu e distribuită prin app store, încă e un website pe care îl poți descoperi pe Google, dar e optimizat pentru a fi: reliable, rapid, engaging.

- de încredere(reliable): adică funcționează offline sau componente ale ei funcționează offline
- rapidă: și oferă reacție în timp rapid utilizatorului
- atrăgătoare: permite utilizarea unor trăsături native cum ar fi folosirea locației sau camera.

Ionic este la bază un set de componente web. Pe de altă parte, componentele web sunt o tehnică, o tehnologie suportată de browser-ele(navigatoarele de internet) moderne care-ți permit construirea propriilor elemente HTML cu o logică complexă în spate.

Înainte puteai face asta construindu-ți propriul HTML, adăugând propriul CSS și logica JavaScript; Ionic oferă această funcționalitate împachetată deja într-o componentă web, alături de alte componente care pot fi folosite. Asta presupune ceea ce vezi pe ecran, dar ecosistemul Ionic e mai mult de atât.

Capacitor - un tool care îți ia aplicație web existentă și o împachetează într-un "web view" într-o aplicație nativă. În final ai o aplicație nativă care-ți rulează aplicația web înăuntru. (o versiune modernă a tool-ului deja existent Cordova; construit de la pământ de Ionic)

Ionic CLI(command line interface) care poate lucra cu alte CLI-uri -aici cu Angular CLI pe care îl folosesc în spate; ceea ce ajută la întregul build workflow; Ca rezultat, dezvoltarea va fi mai rapidă, ușoară și permite ca într-un final aplicația să fie creată și împachetată corespunzător.

2.1.2 Angular

Angular este un framework JavaScript care permite crearea aplicațiilor pe o singură pagină (Single-Page-Application (SPAs)). Diferența dintre o aplicație web standard și o aplicație pe o singură pagină constă în faptul că navigarea de la o pagină la alta necesită o generare completă a acesteia. În cazul celei din urmă întreaga aplicație va fi încărcată o singură dată.

- un framework folosit pentru a crea interfețe cu utilizatorul.

- E important de menționat că rulează în browser, nu pe server. (CLIENT-SIDE JS framework). Nu folosește NodeJs sau alt libaj de programare server-sided. Folosește JS așa cum rulează pe browser.

- De asemenea, folosește TypeScript, care e un superset JavaScript, dar asta se întâmplă deoarece ca developer/dezvoltator e mai convenabil. În cele din urmă vom avea o grămadă de cod JS, chiar dacă nu-l scriem direct.

SPA - asta în principiu înseamnă că avem o singură pagină HTML pe care o creăm și o servim serverului web pentru clienți; utilizatorilor care vizitează pagina din browser și pe această pagină încărcăm script-ul Angular: index.html

- totul e controlat de JS, ceea ce îi conferă rapiditate, deoarece JS rulează pe browser și chiar dacă pierdem conexiunea la internet, nu e nevoie să descarce nimic de pe server în afară de pagina inițială, deci totul poate fi recuperat rapid; e foarte plăcut pentru utilizator deoarece e foarte reactiv

2.1.2.1 Componente

- angular gândește în module. și un modul e pur și simplu un pachet de caracteristici, un pachet de componente

2.1.2.2 Servicii

- instantiate o singura data pe durata de viata a unei aplicatii

- organizeaza logica din spate

- services handle certain functions on behalf of an application's components.

2.2 Limbaje de programare

Chapter 3

Implementare

3.1 Navigation

Un dintre cele mai importante librării dintr-o aplicație Angular. <https://ionicframework.com/docs/angular>

3.1.1 Lazy-routing

-lazy loading inside → if we don't press something the code for the page won't be downloaded unnecessarily

3.1.2 Tabs

-sunt un concept de navigare obișnuit

-am → Stive de Pagini Separate (Separate Navigation Stacks)

-ține minte unde ești pe fiecare stivă; nu se amestecă informația

3.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

```
\hypersetup{urlcolor=red}, or  
\hypersetup{citecolor=green}, or  
\hypersetup{allcolor=blue}.
```

If you want to completely hide the links, you can use:

```
\hypersetup{allcolors=.}, or even better:  
\hypersetup{hidelinks}.
```

If you want to have obvious links in the PDF but not the printed text, use:

```
\hypersetup{colorlinks=false}.
```


Bibliography

- Arnold, A. S. et al. (Mar. 1998). "A Simple Extended-Cavity Diode Laser". In: *Review of Scientific Instruments* 69.3, pp. 1236–1239. URL: <http://link.aip.org/link/?RSI/69/1236/1>.
- Hawthorn, C. J., K. P. Weber, and R. E. Scholten (Dec. 2001). "Littrow Configuration Tunable External Cavity Diode Laser with Fixed Direction Output Beam". In: *Review of Scientific Instruments* 72.12, pp. 4477–4479. URL: <http://link.aip.org/link/?RSI/72/4477/1>.
- Wieman, Carl E. and Leo Hollberg (Jan. 1991). "Using Diode Lasers for Atomic Physics". In: *Review of Scientific Instruments* 62.1, pp. 1–20. URL: <http://link.aip.org/link/?RSI/62/1/1>.