

Architectural Design Document

Arhitectura sistemului

Nume proiect	beFit
Nume document	Arhitectura sistemului / Architectural Design Document
Autori	Iurea Bogdan-Iulian, Matei Cosmin-Constantin

Versiuni

Data	Versiune	Modificări
22.10.2020	0.1	Structura documentului

1. Introducere

1.1. Scopul documentului

Acest document descrie design-ul arhitectural al aplicației 'beFit'. Scopul lucrării este acela de a acoperi arhitectura de nivel înalt și design-ul sistemului. De asemenea se urmărește înregistrarea informațiilor esențiale pentru a defini efectiv arhitectura și design-ul sistemului, cu rolul de a ghida echipa de dezvoltare în implementarea acestui proiect.

1.2. Scurtă descriere a proiectului

Prin acest proiect se urmărește ca utilizatorii să poată efectua operațiuni care în afara mediului online ar necesita timp și efort suplimentar, astfel prin implementarea aplicației web, aceștia își pot crea abonamente mult mai simplu și mai eficient. De asemenea, prin utilizarea aplicației, se oferă posibilitatea de a scuti clientul din a purta unul sau mai multe documente care atestă deținerea de abonament, făcând evidența asupra abonamentelor deținute mult mai simplă.

1.3. Definiții și abrevieri

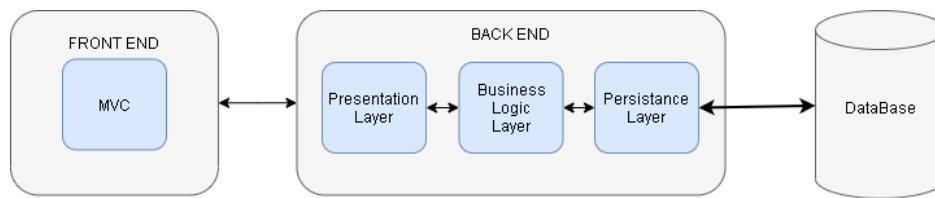
CRUD - Create Retrieve Update Delete

API - Application Programming Interface

REST - REpresentational State Transfer

MVC - Model View Controller

2. Prezentarea sistemului



3. Arhitectura top-level

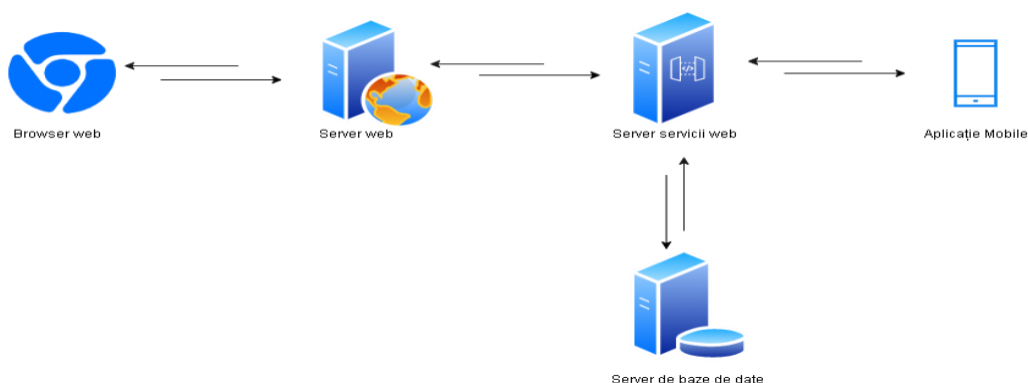
3.1. Identificarea componentelor

- Server de baze de date
- Server servicii web
- Server web
- Aplicație mobile

3.2. Relațiile și comunicarea dintre componente

Aplicația mobilă și serverul web vor comunica direct cu serverul de servicii web. Acesta, în funcție de tipul cererii va putea trimite la rândul său cereri către serverul de baze de date pentru a primi informații necesare pentru procesarea răspunsului cererii primite sau va putea răspunde direct, fără a mai face cereri către baza de date.

Serverul web va prelua informațiile primite de la serverul de servicii web și va crea la rândul său un alt răspuns pe care browserul web îl poate interpreta.



4. Descrierea componentelor

4.1. Serverul de baze de date - MySQL

Baza de date a proiectului va avea doi utilizatori, unul cu rol de administrator, cu control deplin asupra bazei de date iar unul cu rol de read-write care va putea modifica informațiile din tabele.

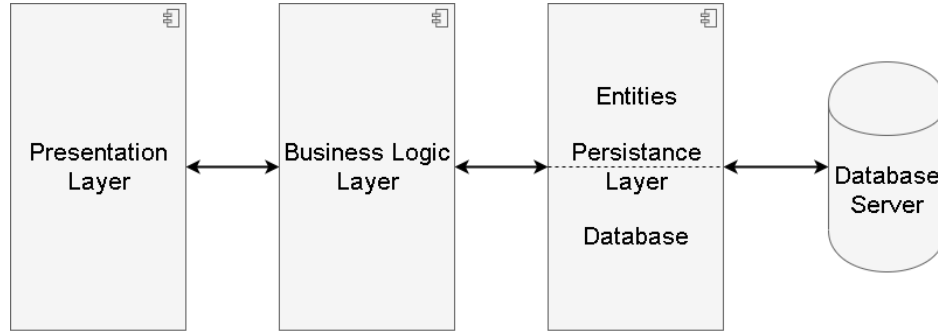
4.2. Serverul de servicii web

Reprezintă componenta principală a aplicației. Ca stil arhitectural se va folosi RESTful API deoarece cerințele aplicației se modelează perfect în raport cu principiile de proiectare REST.

Pentru a dezvolta serverul API se va utiliza framework-ul Spring, pentru că oferă o multitudine de funcționalități, este foarte flexibil, performant și de actualitate.

Acesta va fi împărțit în trei unități logice conform arhitecturii 3-Layer:

- Presentation Layer:
 - conține metodele de gestiune a mapărilor rutelor.
 - apelează serviciile din Business Logic.
- Business Logic Layer:
 - reprezintă unitatea logică ce se ocupă ca toate informațiile citite din baza de date sau primite de la un client să fie procesate și trimise mai departe către client/baza de date.
- Persistence Layer:
 - conține logica de conectare la baza de date precum și logica pentru operațiile CRUD peste aceasta.
 - împărțit în două module:
 - Entities - conține clasele pentru maparea cu tabelele din baza de date.
 - Database - conține clasa-context pentru realizarea conexiunii la baza de date și repository-urile pentru fiecare tabelă a bazei de date.



4.3. Serverul web

Framework-ul folosit pentru dezvoltarea componentei: **Angular**

Acesta va fi împărțit în două module:

- **MVC Layer** - Se va respecta arhitectura MVC
 - **Controller** - unitate de procesare
 - Trimite date către serverul de servicii web
 - Aduce date de la serverul de servicii web
 - Primește cererile de la useri
 - Trimite răspunsuri către useri
 - **Model**
 - Instituie modalitatea de reprezentare a datelor trimise/primate de către server
 - **View**
 - Reprezintă răspunsul format ce urmează a fi trimis către client

4.4. Aplicația mobilă

Pentru aplicația mobilă, framework-ul ales este **React Native**.

Dezvoltarea acesteia este făcută în principal prin apelarea serviciilor din serverul de servicii și afișarea răspunsului astfel încât panourile aplicației să fie asemănătoare cu "View-urile" trimise de serverul web.