

PSSC

Tema 1

Petruşan Radu, gr. 3.1

TEMA 1 curs - Analiza unui sistem software la alegere

- prezentarea cerintelor funcționale si non-functionale si identificarea acelor cerinte care influenteaza arhitectura
- descompunere in componente, definirea responsabilitatilor componentelor si a relatiilor dintre ele; argumentare
- prezentarea sistemului software din doua perspective (o diagram pentru fiecare perspectiva + explicatii)
- identitificarea celor mai importanți 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate si argumetarea alegerii
- identificarea tehnologiilor middleware folosite pentru a comunica intre componente, argumentarea alegerilor
- identificarea pincipalelor modele și stiluri arhitecturale folosite, argumentarea alegerilor
- prezentarea scenariilor de validare a arhitecturii

Sistem software ales:

Sistem de programare la salon de infrumusetare

1. Prezentarea cerințelor

Cerințe funcționale:

- Sistemul trebuie să ofere posibilitatea clienților unui salon de înfrumusețare să își facă programări online
- Sistemul trebuie să permită angajaților salonului să accepte/respingă comenzile făcute de către clienți. În cazul în care un client nu respectă o programare, angajații pot bloca accesul clientului de a face noi comenzi, până la deblocare. Dacă un angajat nu respectă ora progamării, având întârzieri de mai mult de 15 minute, clientul poate cere un discount de până la 30%

- Sistemul trebuie să ofere un total de plată clientului, la confirmarea programării de către un angajat al salonului.
- Sistemul trebuie să dea posibilitatea clienților să vizualizeze toate ofertele și serviciile oferite de către salon
- Sistemul trebuie să permită managerilor de salon să adauge/modifice/șterge datele angajaților, cât și a clienților

Cerințe non-funcționale:

- Acest sistem va implementat în Visual Studio Code, bazat pe AngularJS 4
- Baza de date va fi de tip nerelaționată (Google Firebase)

Cerințe care influențează arhitectura:

- Se dorește ca sistemul să ofere atât aplicație Web, cât și aplicație mobile
- Implementare sistemului se bazează pe o arhitectură client-server.

2. Descompunere în componente

A. Aplicație Web

Aplicația Web va oferi utilizatorilor posibilitatea să se logheze în aplicație, și în funcție de drepturile acestora, vor avea acces la diferite pagini.

Clienții vor putea să vadă profilul lor și să îl actualizeze, să facă cereri pentru programări și să vadă statusul acestora

Angajații salonului vor putea să vadă cererile pentru programări, să le gestioneze și să trimită mesaje clienților.

Managerii salonului pot gestiona toți utilizatorii care folosesc aplicația.

B. Aplicație Mobile

Pe lângă funcționalitățile pe care le oferă aplicația Web, aplicația de mobile va trimite notificări atât clienților cât și angajaților salonului, cu date referitoare la programări.

C. API

API-ul este interfața care este folosită atât de aplicația WEB, cât și de cea mobile, fiind componenta care face legătură între aplicații și baza de date

D. Baza de date

Este component unde sunt stocate datele utilizatorilor.

E. Utilizatori

- a. Angajat salon
- b. Client
- c. Manager salon

3. Prezentarea sistemului software

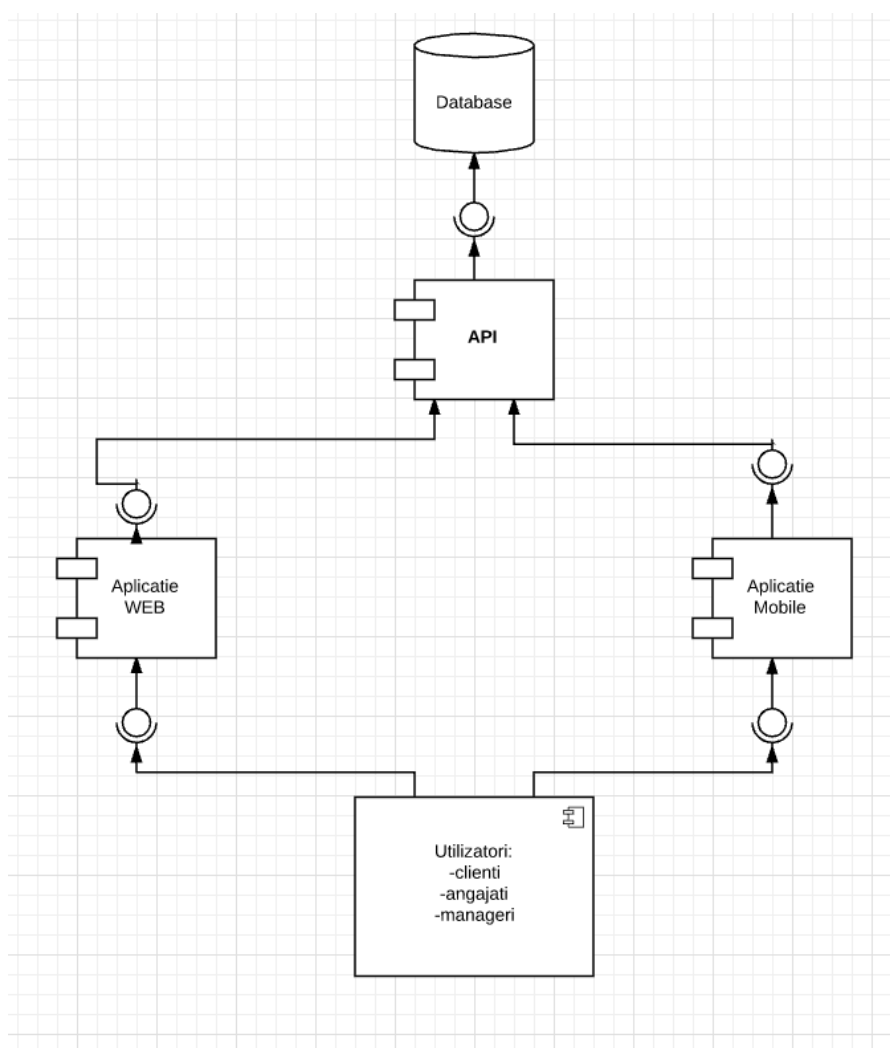


Fig. 1 – Diagramă de componente

Se observă că toate componentele descrise la punctul 2 au fost acoperite în această diagramă.

Utilizatorii au acces la aplicația mobile și aplicația WEB. La rândul lor, cele 2 aplicații comunică cu API-ul, care este în legătură directă cu baza de date.

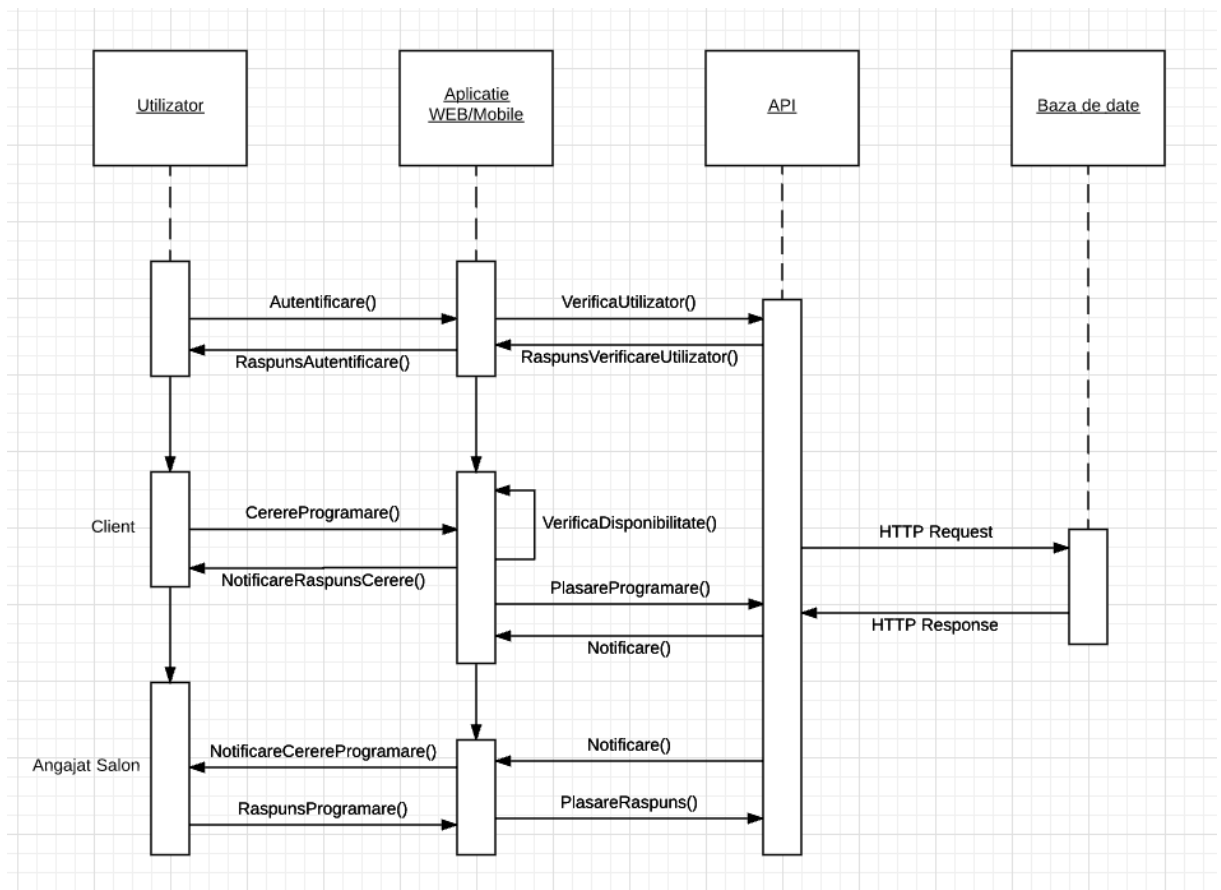


Figura 2 – Diagrama de secvență

Utilizatorii se vor autentifica în aplicația mobile sau aplicația WEB. Aplicația trimite mai departe această cerere către API, care, prin intermediul requesturilor HTTP comunică cu baza de date, unde se verifică dacă utilizatorul există și parola este corectă.

Dacă autentificarea a avut loc cu succes, utilizatorii vor putea face cereri de programări (clienții) și să le gestioneze pe acestea (angajații salonului).

4. Identificarea celor mai importanți 3 indicatori de calitate

a. Timpul de răspuns

Trăim în era vitezei și de aceea este foarte important ca utilizatorii care folosesc această aplicație să aibă parte de cea mai bună experiență.

Requesturile trebuie să primească răspuns repede, iar partea de UI trebuie să fie cât mai fluidă

b. Securitatea

Deoarece vorbim despre o aplicație WEB, trebuie avut în vedere că astfel de aplicații pot fi foarte ușor ținta unor atacuri cibernetice. Asta înseamnă că securitatea reprezintă un aspect esențial.

Datele trebuie criptate pentru a nu exista riscul atacurilor de tip "man-in-the-middle". Utilizatorii trebuie să aibă datele în siguranță.

Autentificarea/autorizarea este alt factor important în securitatea aplicației. Accesul la aplicație trebuie să poată fi verificat și limitat.

c. Disponibilitatea

Aplicația dezvoltată va trebui să fie accesibilă utilizatorilor 24h din 24. În caz contrar, salon riscă să piardă clienți și asta aduce automat o scădere a profitului firmei.

5. Identificarea tehnologiilor middleware folosite pentru a comunica între componente

Se vor folosi tehnologii middleware bazate pe mesaje. Avantajele acestui tip de tehnologie software sunt multiple:

- sistemele software care sunt interconectate prin aceste tehnologii pot rula pe diferite mașini și software, fără a se impune o anumită platformă
- se pot conecta componente deja existente, fără ca acestea să fie rescrise

Modul în care aceste tehnologii vor fi implementate este "cozi de mesaje", un model de comunicare slab-cuplat și asincron.

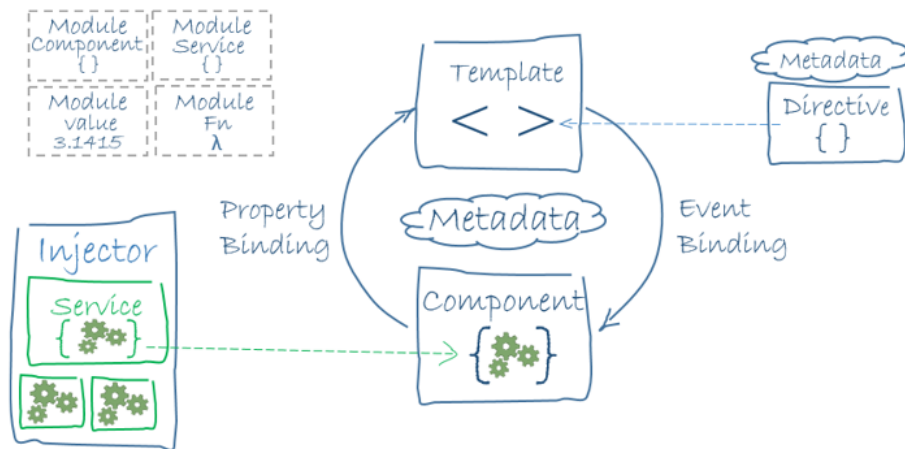
6. Identificarea principalelor modele și stiluri arhitecturale

Având în vedere tehnologiile utilizate pentru dezvoltarea acestui system software, arhitectura este specifică aplicațiilor WEB, construite cu Angular 4.

Aplicația va avea un comportament modular, bazat pe module care la rândul lor sunt formate din componente.

Aplicația WEB va fi de tipul client-server, clientul comunicând cu serverul prin intermediul requesturilor HTTP.

Aplicația mobilă va avea aceeași arhitectură ca și cea WEB, datorită tehnologiei cu ajutorul căreia este dezvoltată. Ionic Framework permite dezvoltarea de aplicații native



7. Prezentarea scenariilor de validare a arhitecturii

- Clientul își crează cont
- Utilizatorul se loghează în aplicație
- Clientul plasează o cerere pentru programare la salon.
- Angajatul primește o notificare despre cererea clientului și o validează/invalidează
- Clientul primește mesaj cu confirmarea/infirmarea cererii

Având în vedere modul în care este dezvoltată baza de date Firebase, folosind WebSockets, se asigură o funcționare foarte bună a aplicației, gestionând cu succes un volum mare de date (în cazul în care mai mulți client sunt conectați concomitant).