

UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI”, IAȘI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

# UniHub

Platformă de socializare pentru studenți

## Arhitectura sistemului

Budeanu Radu-Andrei

Codău Eduard-Alexandru

Nichiforeac Cristina

Șecman Alexandru

Ștefan Alexandra

# Cuprins

<b>Introducere</b>	<b>4</b>
Scopul documentului	4
Scurtă descriere a proiectului	4
Definiții și abrevieri	4
<b>Prezentarea sistemului</b>	<b>4</b>
Background	4
Design-ul de bază al aplicației	5
2.4 Decizii de design	7
2.4.1 Framework-uri	7
2.4.3 Model-View-Controller	8
<b>Funcționalitate</b>	<b>9</b>
Identificarea componentelor	9
Relațiile și comunicarea dintre componente	10
Caracteristici adiționale	10
<b>Descrierea componentelor</b>	<b>11</b>
<b>Concluzii</b>	<b>12</b>

## Definirea cerințelor

<b>Nume proiect</b>	UniHub
<b>Nume document</b>	Arhitectura sistemului / Architectural Design Document
<b>Autori</b>	Budeanu Radu-Andrei, Codău Eduard-Alexandru, Nichiforeac Cristina, Șecman Alexandru, Ștefan Alexandra
<b>Reviewed by</b>	-
<b>Status</b>	În lucru

### *Versiuni*

<b>Data</b>	<b>Versiune</b>	<b>Modificări</b>
<b>20.10.2021</b>	0.1	Structura documentului
<b>25.10.2021</b>	0.2	Adăugare specificații
<b>26.10.2021</b>	0.3	Adăugare diagrame și completări

# 1 Introducere

---

## 1.1 Scopul documentului

Documentul “Arhitectura sistemului” pune la dispoziție o descriere detaliată a arhitecturii aplicației și a modului în care interacționează componentele între ele. Acest document este necesar în vederea dezvoltării propriu zise a aplicației.

## 1.2 Scurtă descriere a proiectului

Proiectul UniHub este o platformă de socializare pentru mediul academic, disponibilă atât în versiune web cât și ca aplicație de mobil. Aceasta facilitează interacțiunea dintre utilizatori, indiferent de statutul lor - studenți sau profesori - punând la dispoziție modalități de creare și distribuire de mesaje și conținut.

## 1.3 Definiții și abrevieri

- UniHub – University hub. Platformă de socializare pentru studenți.
- MVC - Model View Controller. Model de proiectare al aplicațiilor.
- FTP - File Transfer Protocol. Protocol pentru transferul de fișiere.
- UI - User Interface. Interfața cu utilizatorul.
- API - Application Programming Interface. Interfața de acces pentru aplicații.

# 2 Prezentarea sistemului

---

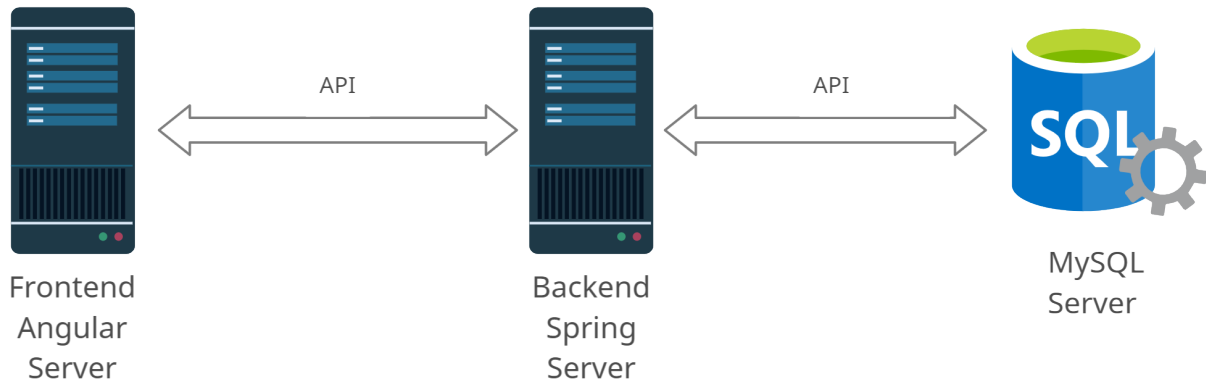
UniHub este o aplicație de tip platformă de socializare destinată interacțiunii studenților din cadrul unui campus. Sistemul conține atât o interfață web, cât și o aplicație mobilă destinată utilizatorilor terminali pentru a putea interacționa, în acest mod oferindu-se libertate în utilizare.

Pentru o descriere mai amănunțită a platformei UniHub, a se vedea documentul cu specificarea cerințelor utilizatorului.

## 2.1 Background

Platforma UniHub are ca scop îmbunătățirea modului de comunicare între persoanele din același sistem de învățământ prin punerea la dispoziție a unei interfețe web/aplicații mobile intuitive și ușor de înțeles.

## 2.2 Design-ul de bază al aplicației



*Fig 1 - Diagrama generală a aplicației*

Platforma de socializare UniHub va fi structurată în 3 componente principale reprezentate de: un server realizat prin intermediul framework-ului Angular pe partea de front-end, rulat în NodeJs și un server MySql ce va deservi bazele de date, între care se va face legătura cu ajutorul unui server implementat prin framework-ul Spring pe partea de back-end. Acest design este specific modelului MVC, fiecare componenta implementand cate un modul din model:

- Server-ul de NodeJs cu Angular reprezintă componenta **View**.
- Server-ul care rulează în Spring reprezintă componenta **Controller**.
- Server-ul care rulează MySql reprezintă componenta **Model**.

## 2.3 Diagrama de activitate

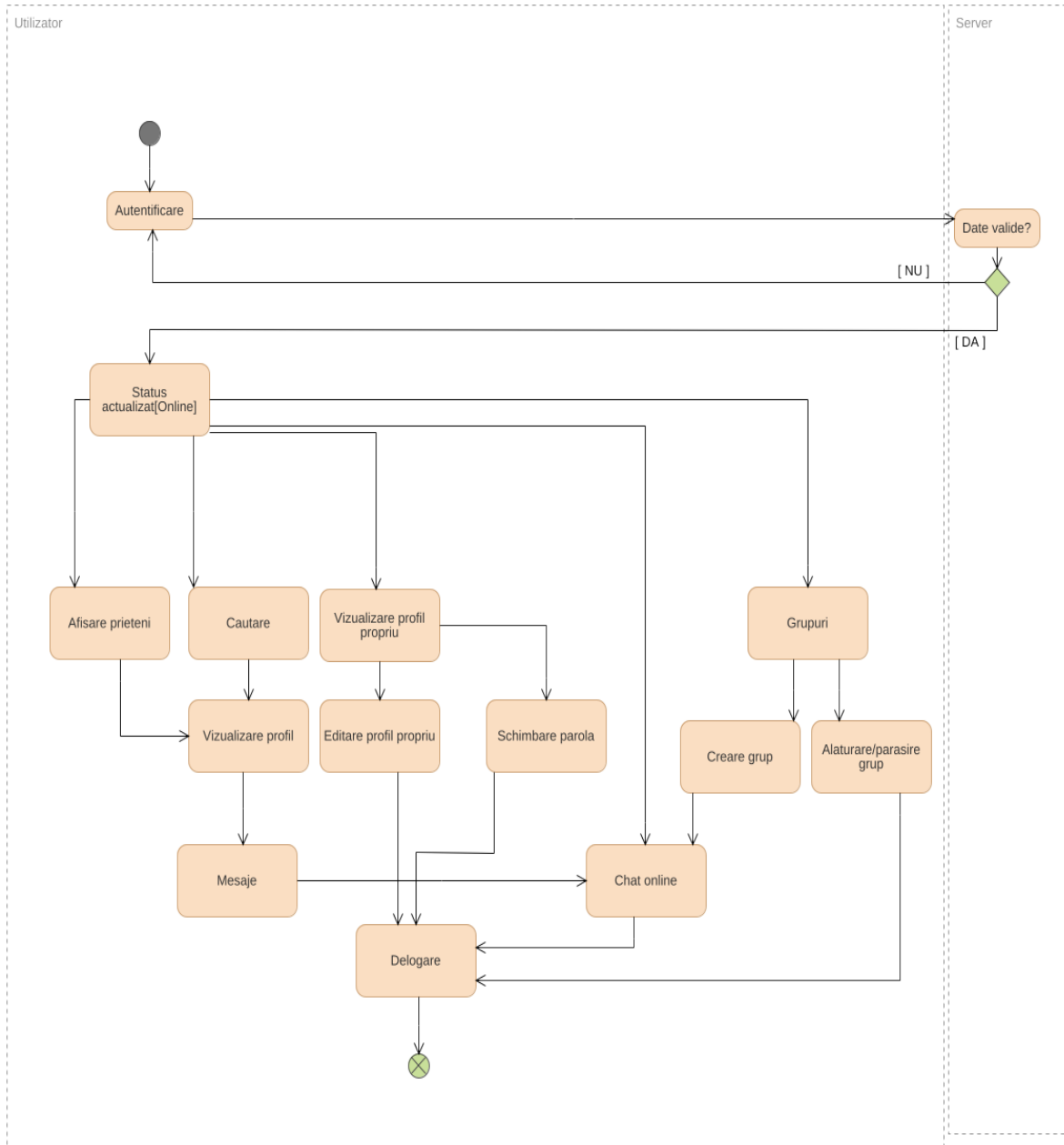


Fig. 2 - Diagrama de activitate pentru utilizator

Diagrama de mai sus prezintă principalele activități pe care utilizatorul le poate desfășura folosind aplicația. Aceste activități țin strict de partea de client, flow-ul și relațiile dintre componentele de back-end, urmând să fie descrise în următoarele paragrafe.

## 2.4 Decizii de design

Tehnologiile și șabloanele de proiectare care au fost alese în dezvoltarea aplicației vor fi detaliate în următoarele subcapitole.

### 2.4.1 Framework-uri

În această secțiune sunt prezentate principalele framework-uri și tehnologii ce vor fi utilizate în construirea aplicației.

Spring Boot este un framework bazat pe microservicii care oferă dezvoltatorului numeroase avantaje precum ușurința și rapiditatea dezvoltării unei aplicații de sine stătătoare sau dependențele injectate automat care duc la o configurare ușoară. Arhitectura orientată pe microservicii accelerează procesele de dezvoltare și implementare prin utilizarea setărilor implicite intuitive pentru testele unitare și de integrare. Din aceste considerente, partea de backend va fi realizată în Java, pe baza framework-ului Spring Boot.

Frontend-ul aplicației va fi dezvoltat în Angular, un framework open-source întreținut de Google. Angular este construit cu TypeScript. Principalul avantaj al acestui limbaj este că îi ajută pe dezvoltatori să-și păstreze codul curat și ușor de înțeles. Bug-urile sunt mai ușor de detectat și eliminat, acest lucru fiind benefic atunci când vine vorba de depanare. Lizibilitatea codului Angular facilitează înțelegerea proiectului, iar componentele pot fi ușor decuplate și înlocuite cu implementări îmbunătățite, aducând un plus în ceea ce privește dezvoltarea aplicației.

Structura conținutului web va fi realizată cu ajutorul tehnologiei HTML, iar descrierea aspectului poate fi realizată folosind CSS. Funcționalitatea este întreținută folosind TypeScript, un superset sintactic al limbajului JavaScript și asigură un sistem de tipuri opțional.



*Fig. 3 - Principalele tehnologii utilizate*

## 2.4.2 Baza de date

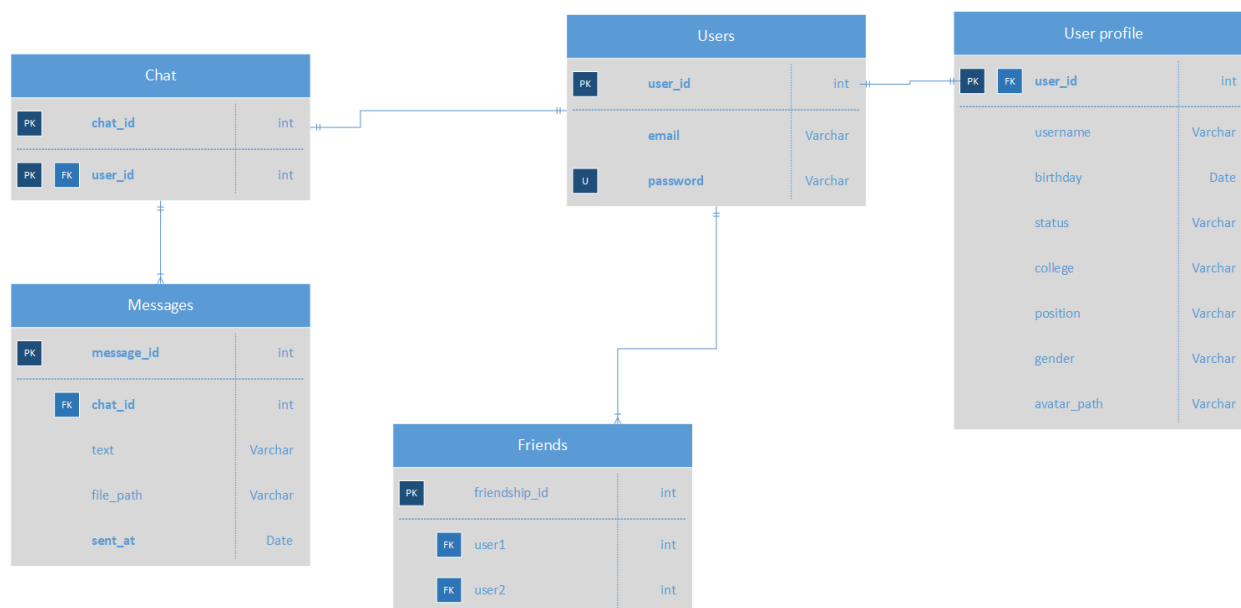


Fig. 4 - Diagrama ER pentru baza de date

Platforma UniHub va utiliza un server de baze de date MySQL. Se va apela la această variantă întrucât reprezintă o opțiune open-source, compatibilă cu majoritatea furnizorilor de găzduire, rentabilă și ușor de gestionat.

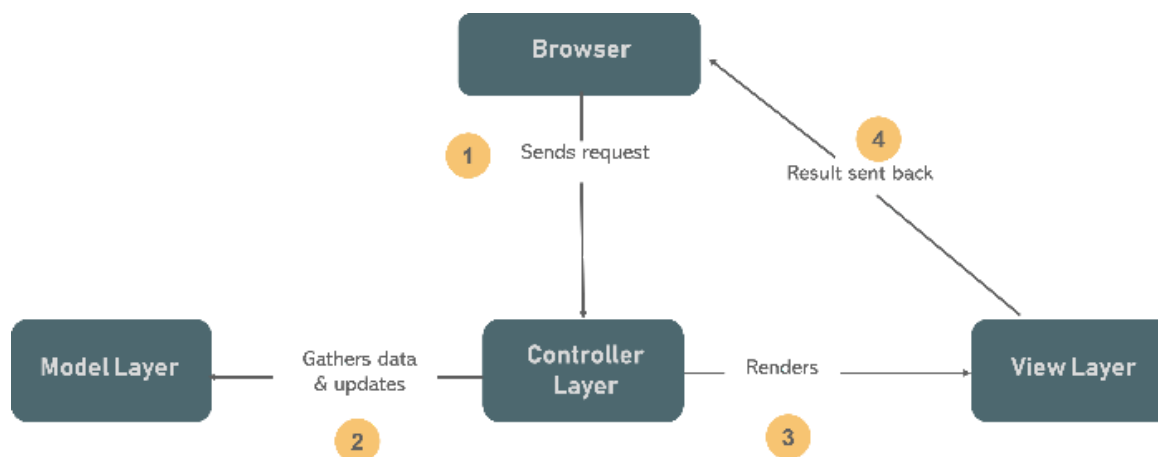
Multe organizații folosesc suportul oferit de MySQL legat de securizarea datelor și a tranzacțiilor atât pentru securizarea acțiunilor desfășurate online cât și pentru îmbunătățirea interacțiunilor cu clienții.

## 2.4.3 Model-View-Controller

Modelul arhitectural Model-View-Controller (Model-Vizualizare-Controlor) este utilizat în ingineria software întrucât izolarea logicii de business în cele trei module sugestive facilitează dezvoltarea, modificările putând fi făcute fără a afecta celelalte nivele.

- Model - responsabil pentru logica legată de datele cu care lucrează dezvoltatorul. Aceasta poate reprezenta fie datele care sunt transferate între componentele View și Controller, fie orice alte date legate de logica de afaceri.
- View - utilizată pentru toată logica UI a aplicației.
- Controller - reprezintă o legătură între Model și View, pentru a procesa toată logica de afaceri și cererile primite. Manipulează datele folosind componenta Model și interacționează cu View pentru a reda un rezultat final.





*Fig. 5 - Diagrama MVC*

## 3 Funcționalitate

---

### 3.1 Identificarea componentelor

Așa cum am ilustrat și în secțiunea 2.1, componentele aplicației sunt:

- Serverul de servicii web (back-end): furnizează API-ul principal al aplicației, și realizează legătura cu celelalte componente.
- Serverul de baze de date: constituie unitatea de persistență a aplicației, făcând posibilă stocarea datelor pe termen lung.
- Serverul front-end: se ocupă de livrarea interfeței grafice în browser, la cererea clientului.
- Serverul FTP: păstrează stocate fișiere precum conținutul video, documente etc.
- Serverul de control al accesului: face posibilă gestionarea utilizatorilor pe platformă.

### 3.2 Relațiile și comunicarea dintre componente

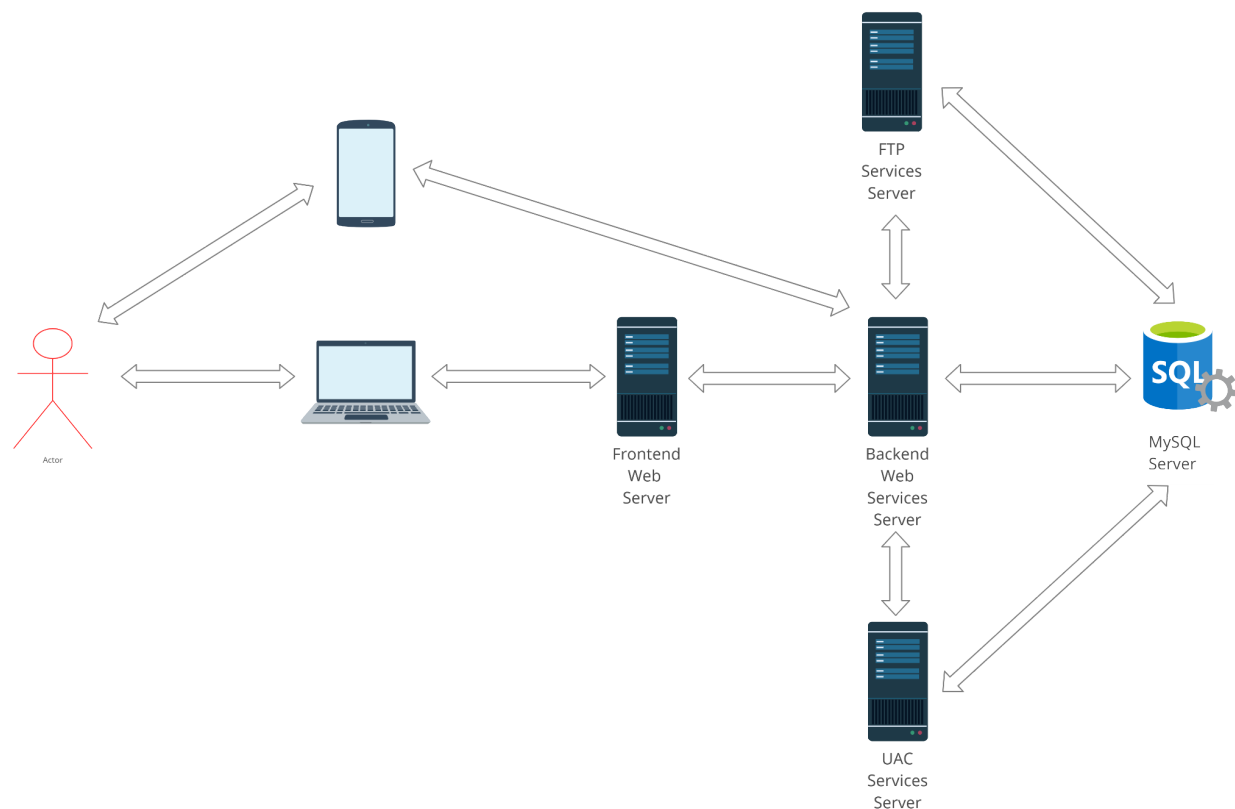


Fig. 6 - Diagrama pentru relațiile între componente

Diagrama prezentată mai sus sugerează modul în care vor relaționa componentele platformei UniHub, din momentul în care un utilizator accesează aplicația din browser sau de pe telefonul mobil până la momentul în care informațiile ajung la server-ul de baze de date, unde sunt stocate și procesate.

Rolul central va fi jucat de server-ul de servicii web, care va face legătura dintre server-ul web, server-ul de front-end al aplicației, server-ul de baze de date și server-ul de control al accesului.

Întregul design este gândit pentru a putea procesa și a livra în final răspunsuri, facilitând interacțiunea utilizatorilor indiferent de browser-ul de pe care se accesează aplicația.

### 3.3 Caracteristici adiționale

Utilizatorii aplicației vor beneficia de anumite opțiuni cu privire atât la profilul personal cât și la funcționalitatea chat-ului, spre exemplu posibilitatea de a accesa setări de profil și de a raporta anumiți utilizatori pentru încălcarea standardelor comunității.

## 4 Descrierea componentelor

---

### 4.1 Server de servicii web

Server-ul de servicii web se va ocupa de procesarea cererilor venite de la server-ul de front-end și transmiterea acestora către componentele de gestionare specifice (FTP, control al accesului, baze de date etc). Acest server va rula un serviciu Aggregator care va trimite cereri și va returna răspunsuri venite de la celelalte servicii importante din sistem. Se dorește acest model de proiectare deoarece toate serviciile vor fi grupate astfel într-o singură entitate ce va fi tratată singular.

### 4.2 Server-ul de front-end

Server-ul aplicației (server-ul de front-end) gestionează solicitările clienților și acționează ca o interfață pentru aceștia. Acesta se ocupă de solicitările de accesare a aplicației, va apela serviciile necesare cererii prin intermediul server-ului de servicii web și va întoarce datele sub forma unor pagini web.

### 4.3 Server de baze de date

Server-ul de baze de date va reprezenta locul în care aplicația va colecta și organiza într-un mod structurat informațiile studenților și profesorilor sub forma unor baze de date complexe, la care vor avea acces doar utilizatorii autorizați (administratorii de sistem).

În ceea ce privește implementarea funcționalităților din spate ale platformei UniHub, se va face apel la interogări pentru regăsirea, validarea și gestionarea anumitor date din tabelele create.

Totodată, referitor la accesul utilizatorilor la datele personale - profil utilizator, conversații private și de grup - și datele publice, acesta va avea loc pe baza autentificării cu adresa de e-mail instituțională și parola setată de utilizatorul respectiv, urmată de confirmarea acestora de către serviciul specific din aplicație.

### 4.3 Server pentru controlul accesului

Server-ul pentru controlul accesului se va ocupa de sistemul de înregistrare și autentificare și, de asemenea, de gestionarea accesului utilizatorilor asupra unor componente, cum ar fi vizualizarea și rezolvarea tichetelor de report.

Acest serviciu va implementa un sistem de management al accesului pe baza unor token-uri, serviciu ce va ține cont de dispozitivele de pe care se conectează utilizatorii. În acest mod se dorește folosirea unui sistem de „stay logged” pentru ca utilizatorul să nu introducă de

fiecare data cand deschide aplicația datele de autentificare, ci doar atunci cand o folosește de pe un dispozitiv nou.

## 5 Concluzii

---

Arhitectura propusă pentru aplicația dezvoltată de echipa noastră este concepută spre a fi implementată conform metodologiei Agile, cu separarea componentelor în task-uri independente, în final furnizând produsul complet într-o manieră progresivă.

De asemenea, arhitectura MVC ne va ajuta la dezvoltarea rapidă a aplicației, dezvoltând și actualizând în paralel modulele necesare.