

**UNIVERSITATEA DIN  
BUCUREȘTI**

**FACULTATEA DE  
MATEMATICĂ ȘI  
INFORMATICĂ**



**SPECIALIZAREA INFORMATICĂ**

**Lucrare de licență**

# **APLICAȚIE DE GESTIONARE DE EVENIMENTE**

**Absolvent**

**Tudoroiu Simona**

**Coordonator științific**

**Lect.dr. Natalia Ozunu**

**București, iulie 2024**

## **Rezumat**

„EventApp” este o aplicație mobilă destinată gestionării evenimentelor. Această aplicație are ca scop îmbunătățirea procesului de promovare a evenimentelor, oferind un sprijin semnificativ administratorilor de local, având șansa să își evidențieze localul într-un loc destinat acestui lucru.

Administratorii de local pot promova evenimentele prezente în localurile sale prin postarea lor, editarea lor și trimiterea de notificări pentru toți utilizatorii participanți la eveniment. Totodată, pot vedea numărul de persoane care își rezervă loc pentru un anumit eveniment și recenziile primite după realizarea acestuia. Pot facilita procesul de planificare a evenimentelor prin calendarul prezent în aplicație, în care sunt introduse și gestionate toate evenimentele care vor avea loc.

Utilizatorii acestei aplicații beneficiază de un spațiu special dedicat găsirii de evenimente. Aceștia pot viziona toate evenimentele prezente în aplicație, să rezerve locuri, să dea recenzii, să își gestioneze timpul prin calendarul prezent, unic pentru fiecare utilizator și să vizioneze evenimentele la care a făcut rezervare.

Un bonus al aplicației este harta interactivă de care se vor bucura utilizatorii, având ca scop localizarea evenimentelor pe hartă, astfel utilizatorul are o privire de ansamblu asupra tuturor evenimentelor și unde sunt acestea situate, cu posibilitatea de a vedea un drum aproximativ până la locație.

## **Abstract**

“EventApp” is a mobile application for event management. This application aims to improve the event promotion process, providing significant support to venue administrators, and having the chance to highlight their venue in a designated place.

The venue administrators can promote events that are taking place in their venues by posting them on the application, editing them and sending notifications to all users participating in the event. At the same time, they can see the number of people who reserved seats for a certain event and the reviews received after the event. They can facilitate the process of planning events through the calendar present in the application, where all the events that will take place can be entered and managed.

Users of this application benefit from a space dedicated to finding events. They can view all the events present in the application, reserve seats, give reviews, manage their time through the present calendar, unique to each user, and view the events they have reserved.

A bonus of the application is the interactive map that users will enjoy, with the aim of locating the events on the map, so the user has an overview of all the events and where they are located, with the possibility of seeing an approximate route to the location.

# Cuprins

CAPITOLUL 1.....	7
Introducere.....	7
1.1. Preliminarii.....	7
1.2. Motivația alegerii temei.....	8
1.3. Scopul aplicației.....	9
1.4. Stadiul temei la momentul actual.....	9
1.5. Contribuția adusă.....	11
1.6. Structura lucrării de licență.....	11
CAPITOLUL 2.....	13
Tehnologii folosite.....	13
2.1. Node.js.....	13
2.1.1. NPM - Node Packet Manager.....	13
2.1.2. JWT - JSON Web Token.....	14
2.1.3. Funcționalități importante folosite în proiect.....	14
2.2. React Native.....	15
2.2.1. Expo.....	16
2.2.2. NPM.....	16
2.3. MySQL.....	17
2.3.1. Baza de date a aplicației.....	17
CAPITOLUL 3.....	19
Structura tehnică a aplicației.....	19
3.1. Viziunea generală asupra arhitecturii.....	19
3.2. Arhitectura interfeței grafice.....	19
3.3. Design-ul backend-ului.....	20
CAPITOLUL 4.....	23
Descrierea aplicației.....	23
4.1. Pagina de început.....	23
4.2. Login și register.....	23
4.3. Home Page.....	30
4.4. Detaliile evenimentului.....	32
4.5. Harta evenimentelor.....	35
4.6. Calendar.....	38
4.7. Pagina de evenimente.....	39
4.8. Pagina de locații.....	44
4.9. Pagina contului.....	46
4.10. Pagina de contact.....	47
CAPITOLUL 5.....	49
Concluzii și îmbunătățiri viitoare.....	49
5.1. Concluzii.....	49

5.2. Îmbunătățiri viitoare.....	50
BIBLIOGRAFIE.....	51

# Listă de figuri

Figura 2.1. - Diagrama ERD a bazei de date.....	18
Figura 2.2. - Paleta de culori a aplicației.....	20
Figura 2.3. - Exemplu de rută folosită în dezvoltarea aplicației.....	20
Figura 2.4. - Exemplu de funcție definită în controller.....	22
Figura 4.1. - Pagina de început.....	23
Figura 4.2. - Paginile de autentificare aferente utilizatorului, respectiv administratorului de local.....	24
Figura 4.3. - Pagina de creare de cont a utilizatorului.....	25
Figura 4.4. - Primul pas din procesul de creare a contului administratorului, introducerea email-ului.....	25
Figura 4.5. - Al doilea pas din procesul de creare a contului administratorului, fotografierea documentelor.....	26
Figura 4.6. - Paginile destinate administratorului general.....	27
Figura 4.7. - Email-urile de acceptare și respingere.....	27
Figura 4.8. - Funcția de trimitere a email-ului.....	29
Figura 4.9. - Paginile de completare a datelor contului de administrator de local.....	29
Figura 4.10. - Pagina principală a utilizatorului.....	30
Figura 4.11. - Efectuarea evenimentelor asincrone „fetchHotEvents” și „fetchCategories” simultan.....	31
Figura 4.12. - Pagina principală a administratorului de local.....	32
Figura 4.13. - Pagina detalierei evenimentului.....	33
Figura 4.14. - Pagina destinată locației evenimentului.....	34
Figura 4.15. - Pagina destinată vizionării rezervărilor aferente unui eveniment.....	35
Figura 4.16. - Codul folosit în frontend pentru randarea hărții.....	37
Figura 4.17. - Pagina hărții interactive.....	37
Figura 4.18. - Agenda și calendarul utilizatorului.....	38
Figura 4.19. - Pagina de evenimente active a utilizatorului.....	39
Figura 4.20. - Funcția de salvare a unei intrări în tabela Reviews.....	41
Figura 4.21. - Pagina evenimentelor inactive și pagina de postat o recenzie.....	42
Figura 4.22. - Pagina cu toate recenziile a administratorului.....	42
Figura 4.23. - Formularul de adăugare a unui eveniment.....	43
Figura 4.24. - Pagina locațiilor.....	43
Figura 4.25. - Paginile destinate adăugării unei noi locații.....	44
Figura 4.26. - Pagina locației administratorului de local.....	45
Figura 4.27. - Pagina contului de utilizator.....	46
Figura 4.27. - Pagina de contact.....	47
Figura 4.28. - Funcția de gestionare a contactării administratorului aplicației.....	48

# Capitolul 1

## Introducere

### 1.1. Preliminarii

Cum a fost prezentat și în rezumat, aplicația „EventApp” reprezintă o aplicație de gestionare de evenimente. Astfel, aplicația are un număr de funcționalități majore, care stau la baza acesteia. Ele sunt următoarele: postarea unui eveniment de către un administrator de local, vizionarea evenimentelor de către un utilizator, rezervarea de locuri pentru un anumit eveniment și vizionarea evenimentelor pe harta interactivă.

Pentru fiecare funcționalitate în parte, va fi descris în detaliu scopul acesteia în aplicație, astfel reieșind importanța fiecăreia.

#### **Postarea unui eveniment de către un administrator de local**

Cea mai importantă funcționalitate a proiectului este cea de postare a evenimentelor. Fără ea, aplicația nu și-ar putea îndeplini scopul de gestionare de evenimente. Aceasta se poate realiza doar de administratorii de local, deoarece aceștia sunt organizatorii evenimentelor.

Un administrator de local va putea lista un eveniment completând un formular ce se va afla pe o pagină specială pentru postare, unde toate câmpurile vor trebui completate, existând constrângeri în baza de date în acest caz. Aceste câmpuri sunt următoarele: numele evenimentului, data și ora de începere, respectiv de terminare, descrierea evenimentului, locurile disponibile și prețul intrării. Câmpul asociat locurilor rezervate va fi completat automat, acesta fiind 0 la postarea evenimentului, ulterior fiind schimbat în funcție de câți utilizatori rezervă locuri până la începerea evenimentului.

Postarea unui eveniment aduce la rândul ei noi funcționalități, și anume: completarea calendarului localului cu evenimentul respectiv și posibilitatea de a posta notificări tuturor utilizatorilor participanți la eveniment.

#### **Vizionarea evenimentelor de către un utilizator**

Un utilizator obișnuit al aplicației va dori să poată vedea toate evenimentele listate din aplicație. Acest lucru este posibil, pagina aferentă vizionării fiind chiar pagina principală,

acolo unde utilizatorul poate alege la ce evenimente vrea să participe.

În pagina principală, utilizatorul poate vedea toate detaliile pe care un administrator le înregistrează în formularul de postare pentru fiecare eveniment în parte. Această funcționalitate exprimă scopul aplicației, și anume găsirea de noi evenimente.

### **Rezervarea de locuri pentru un anumit eveniment**

După ce un utilizator a găsit un eveniment pe placul său, acesta are posibilitatea de a realiza o rezervare pentru a fi sigur că are un loc asigurat. În pagina dedicată detaliilor fiecărui eveniment în parte, există un buton special destinat rezervării de locuri în cadrul acestuia, urmând ca utilizatorul să completeze un formular cu următoarele câmpuri: numărul de locuri rezervate, și o descriere, dacă este cazul (de exemplu, pentru persoanele cu nevoi speciale). În baza de date va exista și un câmp pentru data la care a fost făcută rezervarea, însă acesta va fi completat automat cu data la care utilizatorul va face cererea.

Ca și în cazul postării de eveniment, și rezervarea locurilor aduce funcționalități suplimentare, și anume: completarea calendarului utilizatorului cu evenimentul la care urmează să participe și posibilitatea de a da o recenzie evenimentului după terminarea acestuia, astfel reflectând și recenzia asupra localului.

### **Vizionarea evenimentelor pe harta interactivă**

Pentru a ușura procesul de căutare a evenimentelor, aplicația „EventApp” vine cu soluția unei hărți interactive. Această hartă are ca scop venirea în ajutorul utilizatorului, facilitând astfel procesul de vizionare.

Această facilitate se realizează prin faptul că utilizatorului îi va fi mult mai ușor să vadă unde se situează locația evenimentului, astfel aproximând posibilitatea acestuia de a ajunge. Prin acest fapt, utilizatorul va putea decide mai rapid dacă va putea participa la eveniment, așadar evitând să facă o rezervare la care nu va putea ajunge.

## **1.2. Motivația alegerii temei**

Ca tânără studentă venită din alt oraș în București, am întâmpinat multe situații în care mi-aș fi dorit să pot socializa cu oamenii, însă să mă aflu într-un impas deoarece nu cunoșteam orașul atât de bine încât să știu unde pot afla de desfășurarea diferitelor



evenimente sociale din București. După câteva discuții cu colegi veniți din alte orașe, la fel ca mine, am ajuns la concluzia că o soluție lipsește cu desăvârșire.

În ciuda faptului că București este un oraș mare și plin de atracții turistice diverse, lipsește organizarea în ceea ce privește promovarea evenimentelor sociale, lucru care ar aduce un plus atât turiștilor, studenților, cât și orașului în sine.

Această problemă se regăsește peste tot, însă am ales să iau drept exemplu București datorită experienței mele personale.

### **1.3. Scopul aplicației**

Aplicația „EventApp” are ca scop ajutarea atât acestei industrii de evenimente, cât și a omului comun.

Pentru administratorii de local, „EventApp” își dorește să ajute prin promovarea evenimentelor într-o platformă special destinată acestui lucru. Aceștia nu ar mai fi nevoiți să caute diverse surse de promovare în diferite locuri, existând șansa ca locul de promovare să nu se adreseze publicului potrivit. Astfel, aplicația oferă un spațiu de promovare special destinat publicului care este în căutare de astfel de servicii.

Pentru omul obișnuit, „EventApp” își dorește a promova timpul liber de calitate, socializarea, dar și educația. Aplicația este destinată oamenilor de toate vârstele, tipurile de evenimente postate fiind la latitudinea administratorilor de local. Există diferite tipuri de evenimente, de la lansări de cărți, la petreceri, cinema în aer liber, sau pictură pe față în parc, astfel incluzând toate tipologiile de vârstă.

### **1.4. Stadiul temei la momentul actual**

La momentul actual, pe piața aplicațiilor mobile există soluții asemănătoare cu tema aleasă. Mai jos vor fi prezentate o parte din aplicațiile existente.

#### **Whova**

„Whova” este o aplicație mobilă disponibilă atât pe Android, cât și pe iOS, care are ca scop gestionarea evenimentelor din toată lumea, fie ele fizic, online sau hybrid. La prima interacțiune cu aplicația, aceasta necesită crearea de cont.

Ca și utilizator al aplicației, poți vedea evenimentele la care participi chiar pe pagina

principală, acolo unde poți selecta să vezi toate evenimentele. La majoritatea evenimentelor, ca să poți participa la ele ai nevoie să te înscrii, ulterior primind un ID care trebuie introdus pentru a avea acces la eveniment, sau poți avea un cod de invitație. Pe pagina evenimentului poți vedea doar o descriere a evenimentului, cu detalii cuprinzătoare.

Ca administrator, poți introduce evenimente contra cost, acest cost fiind stabilit în urma completării unui formular pe site în care trebuie răspuns la mai multe întrebări despre un anumit eveniment pe care îl vrei publicat în aplicație. Poți trimite email-uri drept notificări persoanelor care participă la eveniment, să aflii câte persoane au venit, să promovezi, să adaugi voluntari și să vezi rapoarte legate de eveniment. Sunt existente și alte tool-uri de promovare a evenimentului, cum ar fi o galerie foto comună, jocuri tip Trivia, sponsori etc.

## **SpotMe**

„SpotMe” este o aplicație mobilă disponibilă atât pe Android, cât și pe iOS, care are ca scop gestionarea evenimentelor organizate de o anumită firmă. Această aplicație este intenționată folosirii doar în cadrul firmelor, deoarece, ca un simplu utilizator, nu poți accesa nicio funcționalitate a aplicației.

Această aplicație permite gestionarea rapidă și eficientă a evenimentelor, fie ele fizic, online sau hybrid. Pentru a participa la un eveniment trebuie să te înscrii la acesta, și totodată să faci check-in-ul online. Promovarea se face prin diverse funcționalități, precum un loc în care utilizatorii pot discuta, un organizator al evenimentelor, Q&A prezent, poll-uri, funcția de aplaudare, jocuri și întrebări pe care utilizatorul le poate folosi să interacționeze, astfel stârnind interesul.

Pentru a beneficia de funcționalitățile unui administrator, și anume să postezi evenimente și să le gestionezi, trebuie plătit un cost în funcție de ce plan alegi, costul fiind stabilit după completarea unui formular.

## **Bizzabo**

„Bizzabo” este o aplicație mobilă disponibilă atât pe Android, cât și pe iOS, care are ca scop gestionarea de evenimente atât fizice, cât și online sau hybrid. Ca utilizator al aplicației, nu poți accesa funcționalitățile acesteia dacă email-ul de logare nu este asociat unui eveniment.

Aplicația este intenționată folosirii de către organizatori pentru gestionarea mai

eficientă a evenimentelor, existând diferite planuri pe care le poți achiziționa, prețul fiind stabilit ulterior, astfel încât să te poți bucura de beneficiile de administrator.

Funcționalitățile pe care această aplicație le aduce sunt destul de asemănătoare cu cele de mai sus. Pentru a participa la un eveniment trebuie să te înregistrezi, ca utilizator ai parte de servicii onsite, există tool-uri speciale pentru a menține atenția participanților, precum Q&A, poll-uri și live chat, posibilitatea de a schimba datele de contact cu altă persoană folosind funcționalitatea de „SmartBadge” și posibilitatea de a promova evenimentul sunt și ele prezente, și nu în ultimul rând, este prezentă funcționalitatea de a crea rapoarte cu privire la eveniment, participanți etc.

Aceste 3 aplicații sunt printre cele mai folosite pentru gestionarea de evenimente.

## **1.5. Contribuția adusă**

„EventApp” combină o mare parte din funcționalitățile aduse de aplicațiile prezentate în secțiunea anterioară, astfel aplicația are ca obiectiv definirea unui spațiu în care pot fi postate evenimente de toate tipurile, fie ele de divertisment sau de muncă.

Funcționalitatea principală pe care „EventApp” o aduce în plus este harta interactivă. Această hartă localizează toate evenimentele, fiind semnalate cu un marcaj, astfel existând posibilitatea de accesare a evenimentului direct de pe hartă. Totodată, această hartă oferă posibilitatea utilizatorului de a vedea drumul pe care ar trebui să îl parcurgă, cu mașina, până la un eveniment de interes.

Scopul hărții interactive este simplificarea procesului de căutare, astfel utilizatorul fiind capabil să găsească evenimentele din apropierea lui într-un timp mult mai scurt.

## **1.6. Structura lucrării de licență**

Lucrarea de licență prezentă este împărțită în 5 capitole ce prezintă în detaliu funcționalitatea aplicației, utilitatea acesteia și totodată cum a fost implementată. În cele ce urmează, vor fi descrise scopurile fiecărui capitol în parte, pentru a oferi o imagine de ansamblu asupra prezentei lucrări.

**Capitolul 1**, capitolul actual, prezintă informații de bază despre lucrarea de licență. Acesta oferă o perspectivă generală asupra prezentei lucrări, detaliind informații precum scopul și motivația alegerii temei, o introducere în funcționalitățile aplicației și ilustrează alte

aplicații pe aceeași temă, subliniind contribuția pe care aplicația „EventApp” o aduce.

**Capitolul 2** expune toate tehnologiile folosite în realizarea acestei lucrări. Totodată, în acest capitol se intră în detalii în ceea ce privește fiecare tehnologie în parte, incluzând și atribuțiile pe care acestea le-au avut în aplicația mobilă.

**Capitolul 3** oferă informații cu privire la cum a fost structurată prezenta lucrare. În acest capitol se pot observa detalii despre interfața grafică, structura și caracteristicile backend-ului, dar și un sumar despre arhitectura client-server adoptată în cadrul acestui proiect.

**Capitolul 4** prezintă în detaliu proiectul propriu-zis și utilitatea acestuia. Fiecare subcapitol al acestei secvențe reprezintă o pagină importantă din aplicația mobilă „EventApp”, ilustrând atât funcționalitatea principală pe care pagină o descrie, cât și funcționalități minore, dar care aduc aplicației un plus prin atenția la detalii. Pe parcursul acestui capitol se vor regăsi imagini cu paginile importante, așa cum sunt văzute de către utilizator, pentru a putea transmite și ilustra produsul finit.

**Capitolul 5** este reprezentat de atât de concluziile pe care le-am dobândit după finalizarea prezentei lucrări, cât și de un număr semnificativ de viitoare îmbunătățiri de care ar putea beneficia aplicația mobilă „EventApp” pentru a întări și dezvolta funcționalitățile majore deja implementate.

# Capitolul 2

## Tehnologii folosite

### 2.1. Node.js

Node.js [1] este folosit pentru executarea codului de JavaScript în ceea ce privește partea de server, fiind o platformă open-source ce se ocupă cu această problemă. Scopul acestei platforme a fost de a servi dezvoltării aplicațiilor web scalabile, fiind capabil de a gestiona mai multe operațiuni de tip I/O (Input/Output) fără a bloca procesele. Este construit pe motorul V8 ce aparține de Google Chrome, astfel fiind eficient și rapid.

O caracteristică cheie a Node.js este modul de gestionare a operațiunilor I/O, bazat pe evenimente. Această caracteristică este descrisă de executarea asincron a acestor operațiuni, astfel oferindu-i server-ului ocazia de a nu aștepta finalizarea operațiunilor anterioare până la executarea următoarelor. Astfel, Node.js se diferențiază de alte platforme de gestionare a operațiunilor din server, tradiționale și sincrone, ce blochează firele de execuție până la terminarea operațiunii anterioare.

#### 2.1.1. NPM - Node Packet Manager

Node Packet Manager (NPM) [2] este un sistem de gestionare a pachetelor, făcut special pentru JavaScript. Acest sistem permite dezvoltatorilor instalarea, partajarea dar și gestionarea dependențelor specifice dintr-un proiect Node.js, fiind creat pentru a spori productivitatea, lăsând la dispoziția dezvoltatorului un acces vast la pachete care pot fi reutilizabile.

În proiectul de licență, acest sistem de gestionare a pachetelor a fost folosit pentru instalarea anumitor pachete importante dezvoltării server-ului, pachete ce vor fi numite în subsecțiunea „Funcționalități importante folosite în proiect”, dar și pentru gestionarea dependențelor.

Într-un proiect Node.js, există fișierul „package.js”, fișier ce conține metadata relevantă pentru acesta, precum: numele, versiunea, descrierea, autorul, licențele, script-uri de rulare, dar și cel mai important, dependențele proiectului. Acestea au scopul de a stoca atât pachetele folosite în proiect, cât și versiunea necesară a pachetului, astfel încât proiectul să ruleze în același mod pe dispozitive diferite, oferind un mediu de dezvoltare constant.

NPM mai are ca obiectiv și automatizarea altor sarcini de dezvoltare, cum ar fi compilarea sau testarea codului.

### **2.1.2. JWT - JSON Web Token**

JSON Web Token [3] este un „open standard” folosit pentru transmiterea datelor între două instanțe, în siguranță. Token-urile JWT sunt reprezentate printr-un șir de caractere, despărțit prin câte un punct „. ”, fiind codificat și reprezentat de:

- header, care ilustrează atât tipul token-ului, cât și algoritmul de semnare,
- payload, în care apar anumite date relevante, precum date despre utilizator, sau rolul acestuia,
- semnătura, care are ca scop verificarea alterării mesajului, fiind generată de header-ul codificat, payload-ul codificat, și o cheie secretă aleasă la momentul generării token-ului.

Acest token este folosit ca modalitate de accesare a rutelor protejate, astfel server-ul fiind capabil să verifice dacă utilizatorul are drept de accesare a respectivei rute, sau nu.

Avantajele folosirii a token-urilor JWT sunt:

- mărimea relativ mică a token-urilor, astfel fiind posibilă trimiterea acestora prin metode standard HTTP,
- securitatea oferită de către semnătura din token, astfel nefiind posibilă alterarea acestuia,
- menținerea a unui număr suficient de date despre utilizator, astfel încât să nu fie nevoie de o ulterioară interogare a bazei de date.

### **2.1.3. Funcționalități importante folosite în proiect**

Node.js a fost folosit în cadrul proiectului de licență ca platformă de dezvoltare a unui REST API în ceea ce privește partea de backend a aplicației mobile. Astfel, am dezvoltat aplicații de tip CRUD (Create, Read, Update, Delete) pentru fiecare tabelă a bazei de date, plus alte funcționalități importante.

O funcționalitate importantă care este gestionată folosind Node.js este trimiterea de email. În „EventApp”, este folosită această funcționalitate în pasul de autentificare a administratorilor de local. Astfel, după un pas intermediar de verificare a autenticității

administratorului de local, acesta primește un mail în care este furnizat un cod pe care îl poate folosi să intre în aplicație, sau un mail în care îi este specificat faptul că cererea sa de creare a contului a fost respinsă. Tot acest proces este gestionat folosind pachetul „nodemailer” [4], furnizat de NPM, scopul acestuia fiind exact trimiterea de mail-uri direct din aplicație într-un mod cât mai simplu. Pentru utilizarea acestui serviciu, trebuie furnizate host-ul, portul și credențialele de autentificare a contului de pe care vor fi trimise mail-urile. Folosind această librărie, este posibilă atât trimiterea de content HTML, cât și de text simplu (plain text), trimiterea de atașamente și imagini incorporate direct în conținutul HTML al mail-ului, și multe alte opțiuni care pot fi configurate direct din cod.

Alte pachete NPM folosite în proiect sunt:

- bcrypt [5], folosit pentru a crea hash-ul parolelor utilizatorilor,
- dotenv [6], care ajută la încărcarea variabilelor din fișierul .env în process.env, care conține diferite credențiale ale aplicației, astfel fiind posibilă accesarea lor pe parcursul proiectului,
- express [7], un framework web care oferă o diversitate de funcționalități pentru aplicațiile web sau mobile, precum rutarea, folosirea de middleware, ajutor în gestionarea răspunsurilor și cererilor, de la și spre client, integrarea cu bazele de date, etc.,
- multer [8], pachet ce ajută cu manipularea elementelor multipart/form-data, fiind folosit în special pentru manipularea fișierelor,
- mysql2 [9], client de MySQL pentru Node.js,
- sequelize [10], un Object Relational-Mapping (ORM) framework pentru Node.js, folosit pentru a scrie cod JavaScript care să interacționeze cu baza de date,
- nodemon [11], un ajutor în dezvoltarea server-ului de Node.js, oferind automatizarea procesului prin restartul proiectului la fiecare schimbare ce are loc în cadrul oricărui fișier ce aparține de acesta.

## 2.2. React Native

React Native [12] este un framework open source, dezvoltat de către cei de la Facebook, care permite dezvoltarea de aplicații mobile native. React Native a extins capabilitățile framework-ului React, care este destinat dezvoltării aplicațiilor web, însă în loc să utilizeze componente web, acesta folosește componente native platformei. React Native

este o alternativă mai simplă pentru dezvoltatorii de aplicații mobile care sunt obișnuiți cu Node.js, sau ca în cazul proiectului de licență, au deja o platformă de backend ce este obținută cu ajutorul Node.js, deoarece React Native este un framework ce folosește tot JavaScript, ca în cazurile prezentate anterior.

O caracteristică importantă a React Native este posibilitatea de dezvoltare a aplicațiilor mobile destinate atât pentru Android, cât și pentru IOS, utilizând același cod sursă.

Alte caracteristici demne de menționat sunt următoarele:

- folosirea componentelor native, astfel React Native include seturi de componente predefinite ce se mapează la componentele native ale sistemului de operare,
- hot reloading, permițând dezvoltatorilor să vadă modificările aduse interfeței grafice aproape instantaneu de la salvarea fișierului afectat de schimbarea propriu-zisă,
- accesul la API-uri native, prin utilizarea unor module speciale, ce oferă accesul la funcționalități specifice dispozitivului, precum camera foto, locația GPS, senzori, și altele, fiind prielnic dezvoltării aplicațiilor mobile complexe.

### **2.2.1. Expo**

Expo [13] este un framework open source, folosit ca ajutor în dezvoltarea aplicațiilor mobile ce folosesc React Native. Acesta oferă o gamă largă de API-uri extinse, ce acoperă multe funcționalități a dispozitivelor, astfel eliminând nevoia de scriere de cod nativ unde nu este cazul.

O caracteristică interesantă a acestui framework este Expo Go, o aplicație mobilă destinată atât pentru Android, cât și pentru IOS, care permite utilizatorului să testeze aplicația direct pe dispozitiv, nefiind nevoie de un emulator, astfel fiind mai realist tot procesul, atât de dezvoltare, cât și de testare.

### **2.2.2. NPM**

React Native Expo folosește ca manager de pachete tot NPM, precum Node.js. O diferență semnificativă între folosirea NPM pentru Node.js și folosirea lui pentru React Native Expo este faptul că, în a doua variantă, NPM este mult mai folosit, deoarece acest framework se bazează pe instalarea de module specifice pentru fiecare acțiune necesară, fie ea un anumit buton, integrarea camerei foto a dispozitivului, sau folosirea unei anumite iconițe predefinite



într-un anumit pachet.

## 2.3. MySQL

MySQL [14] este un sistem de gestionare a bazelor de date relaționale, care este open source, cunoscut pentru eficiența oferită. Acest sistem utilizează limbajul de interogare structurat SQL pentru a face interogații pe baza de date.

Caracteristicile importante ale acestui sistem sunt următoarele:

- performanța și scalabilitatea, fiind potrivit atât pentru aplicații de dimensiuni mari, cât și pentru cele mai mici,
- portabilitatea, MySQL este disponibil pe diverse versiuni de Windows, Linux și UNIX,
- rezistență la eșec, astfel fiind posibilă recuperarea datelor dintr-o bază de date compromisă.

### 2.3.1. Baza de date a aplicației

Aplicație dispune de o bază de date de mărime medie, ce cuprinde 12 tabele care sunt definite în cod cu ajutorul pachetului „sequelize” din NPM, însă aparând și un tabel de legătură.

În figura de mai jos este reprezentată diagrama ERD a bazei de date, unde pot fi observate toate tabelele, cu tot cu atributele lor și cu relațiile dintre ele.

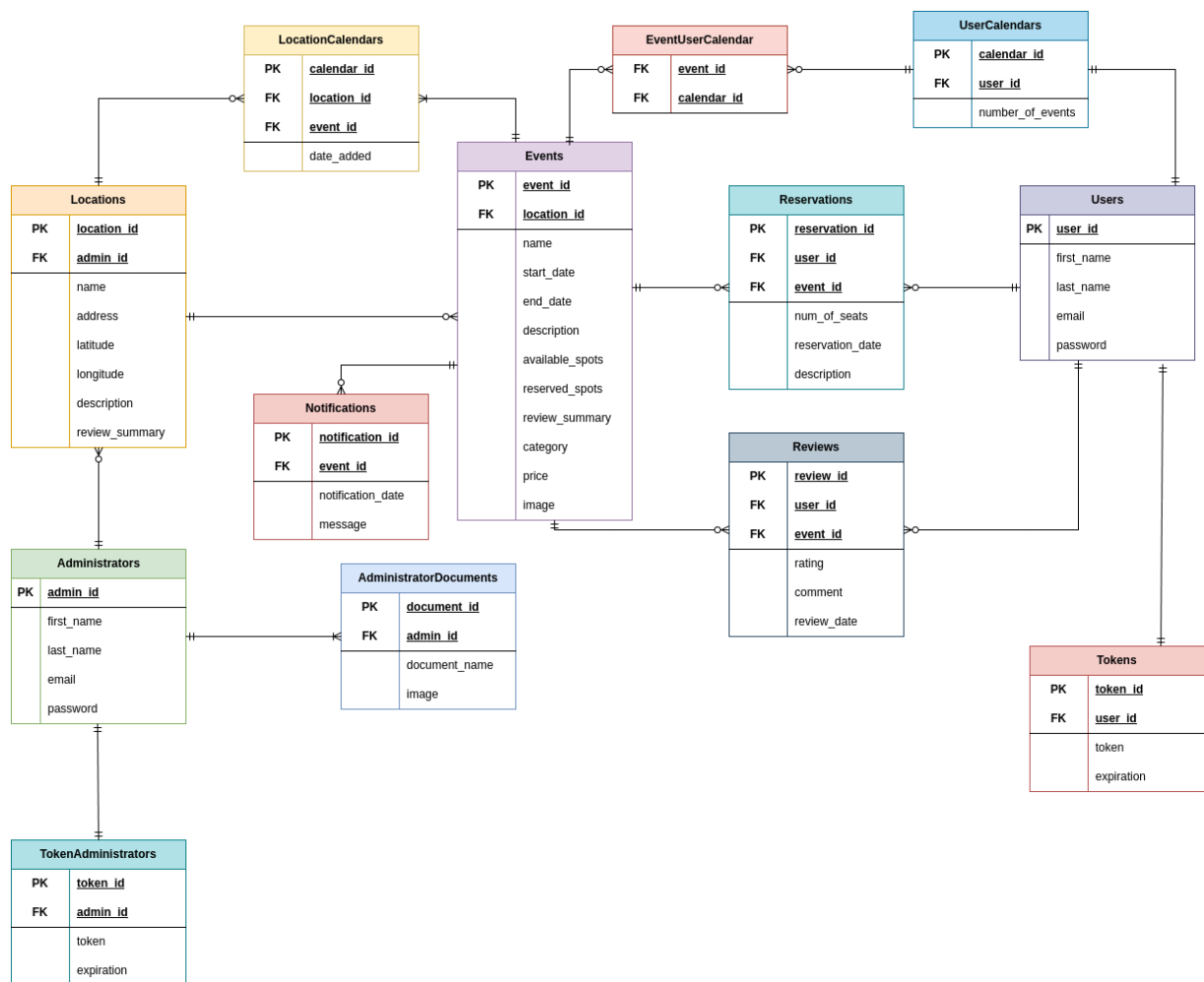


Figura 2.1. - Diagrama ERD a bazei de date

# Capitolul 3

## Structura tehnică a aplicației

### 3.1. Viziunea generală asupra arhitecturii

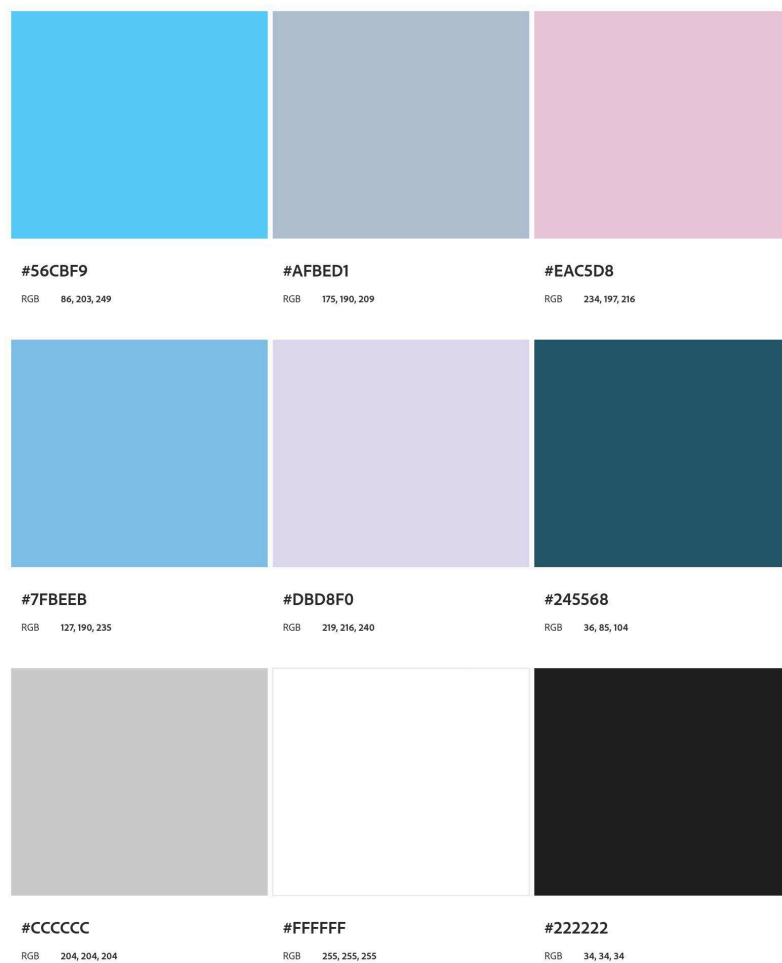
Aplicația mobilă „EventApp” este structurată pe o arhitectură client-server [15]. Această arhitectură are rolul de a eficientiza sarcinile pe care o aplicație trebuie să le îndeplinească, astfel repartizându-le între furnizorii de resurse, server-ul, și solicitanții de resurse, clienții. Este cea mai comună structură arhitecturală, deoarece oferă beneficii semnificative, precum:

- optimizarea resurselor și a sarcinilor, prin descentralizarea acestora în mai multe noduri
- specializarea fiecărui nod ce ia parte la arhitectură, astfel oferind performanțe superioare datorită faptului că sunt optimizate pe o anumită sarcină
- scalabilitate, deoarece aplicația poate fi extinsă cu ușurință doar prin adăugarea unui nou nod, fără a fi nevoie de schimbări majore la restul nodurilor deja existente
- flexibilitate, fiind posibilă inovația cu implementare rapidă, astfel fiind afectat doar nodul care reprezintă nevoia specifică a utilizatorului.

Așadar, aplicația are la bază cele două mari componente menționate mai sus, serverul și clientul. Serverul gestionează logica aplicației, interacționează cu baza de date, și trimite date rafinate către client, folosind un REST API. Clientul se ocupă atât cu manevrarea datelor primite de la server, cât și cu oferirea unei interfețe grafice utilizatorului pentru interacțiuni interactive.

### 3.2. Arhitectura interfeței grafice

Interfața grafică a aplicației „EventApp” are ca scop oferirea utilizatorului a unei experiențe cât mai minimaliste, intuitive, dar care să redea în detaliu atribuția principală a acesteia, și anume gestionarea de evenimente. Astfel, design-ul minimalist a fost combinat cu afișarea unui număr relativ mare de date și cu o paletă de culori care atrage atenția utilizatorului, rezultatul fiind o aplicație eficientă din punctul de vedere al datelor, atractivă din punctul de vedere al culorilor, dar și clară în ceea ce privește folosirea acesteia.



*Figura 2.2. - Paleta de culori a aplicației*

### 3.3. Design-ul backend-ului

Backend-ul este structurat din mai multe componente:

- modelele, care definesc, folosind cod JavaScript, entitățile prezente în baza de date a aplicației, cu tot cu atributele lor și tipul lor de date ,
- rutele, implementate folosind framework-ul Express, reprezentând câte o acțiune specifică a aplicației, precum crearea unui cont, logarea, postarea unui eveniment, etc.,

```
router.post('/login', userController.loginUser);
```

*Figura 2.3. - Exemplu de rută folosită în dezvoltarea aplicației*

- controller-ele, care preiau datele transmise de rute, pentru a face operații de tip CRUD (Create, Read, Update, Delete), adresându-se bazei de date MySQL folosind ORM-ul Sequelize,

```
const uploadImages = async (req, res) => {
  try {
    const files = req.files;
    const images = files.map(file => ({
      filename: file.originalname,
      content: file.buffer
    }));

    const administrator = await Administrator.create({
      first_name: "pending",
      last_name: "pending",
      email: req.query.email,
      password: "pending"
    });

    for (let image of images) {
      const document = await AdministratorDocument.create({
        document_name: image.filename,
        image: image.content,
        admin_id: administrator.admin_id
      });
    }

    res.status(200).json({
      success: true,
      message: "Pending Administrator created successfully",
      data: administrator
    });

  } catch (err) {
    console.error(err);
    res.status(400).json({
      success: false,
      message: "Error uploading images",
    });
  }
}
```

```
        data: err
    });
}
```

*Figura 2.4. - Exemplu de funcție definită în controller*

- `middleware-uri`, care preiau datele înainte de a ajunge în rute, și le validează, verificând, în cazul proiectului prezentat, dacă utilizatorul este logat, ce rol are utilizatorul în această aplicație, și dacă datele trimise sunt de tipul celor așteptate, reprezentând un strat intermediar de securitate, protejând astfel furnizorul de date de diferite atacuri.

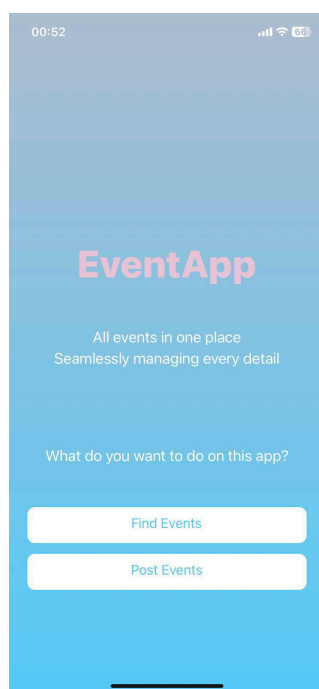
# Capitolul 4

## Descrierea aplicației

### 4.1. Pagina de început

Prima interacțiune a oricărui utilizator cu aplicația „EventApp” este dată de o pagină inițială, ce îl introduce pe acesta în scopul aplicației. Pagina prezintă numele aplicației, un motto ce descrie clar și concis ce dorește a oferi aceasta, și posibilitatea de a alege în ce mod ar vrea persoana respectivă să se folosească de ea: ca simplu utilizator pentru găsirea evenimentelor, sau ca administrator de local, ce primește beneficiul de a posta evenimentele prezente.

După selectarea uneia dintre cele două opțiuni prezente, utilizatorul este îndreptat către pagina de „Login” aferentă rolului pe care l-a ales.

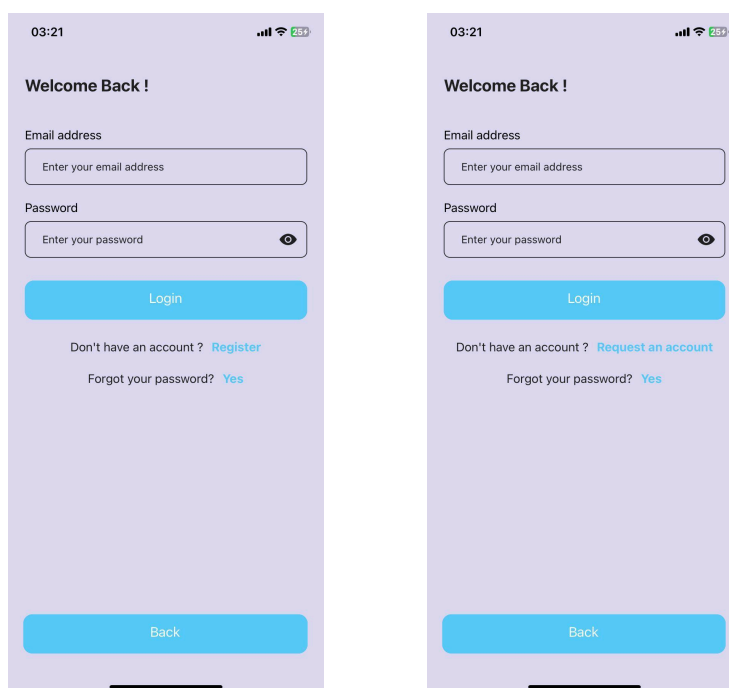


*Figura 4.1. - Pagina de început*

### 4.2. Login și register

Pentru ca utilizatorul să poată intra în aplicație, este nevoit să își creeze un cont, sau să se autentifice cu un cont deja existent în baza de date. În acest caz, există următoarele două procese, cel de login și cel de register.

Procesul de logare este clasic, utilizatorul trebuie să introducă credențialele puse la înregistrare astfel încât să acceseze funcționalitățile aplicației. Așadar, chiar dacă persoana este un utilizator obișnuit, sau un administrator de local, aceasta trece prin același proces de autentificare. Totodată, în cazul în care utilizatorul își uită parola, există posibilitatea de recuperare a contului prin cererea de schimbare a parolei prezentă tot în pagina de autentificare.



*Figura 4.2. - Paginile de autentificare aferente utilizatorului, respectiv administratorului de local*

În ceea ce privește crearea unui cont, procesul stă diferit în funcție de rolul pe care utilizatorul vrea să îl ia în aplicație.

Pentru utilizatorul normal al aplicației, cel care vrea să găsească evenimente, procesul este simplu. Trebuie doar să introducă date personale, precum numele, prenumele, o adresă de email și o parolă. După introducerea acestor date și apăsarea butonului „Sign Up”, datele utilizatorului sunt introduse în tabela „Users”, și acesta este redirecționat la pagina de autentificare.

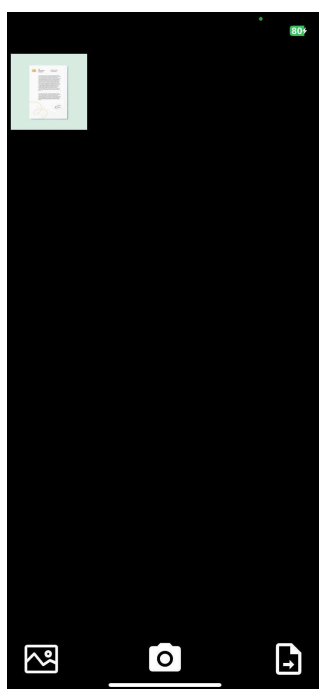


*Figura 4.3. - Pagina de creare de cont a utilizatorului*

Pentru administratorul de local, procesul este mai complex. Având în vedere că nu orice persoană deține un spațiu care administrează evenimente, a fost nevoie de o soluție pentru a valida acest lucru. Așadar, soluția pe care am găsit-o a fost ca procesul de creare de cont să conțină trimiterea unor poze cu documentele care atestă că persoana care vrea să posteze evenimente deține un astfel de spațiu. În primă fază a procesului, acesta trebuie să își introducă adresa de email, pe care urmează să primească răspunsul la cererea de creare de cont.

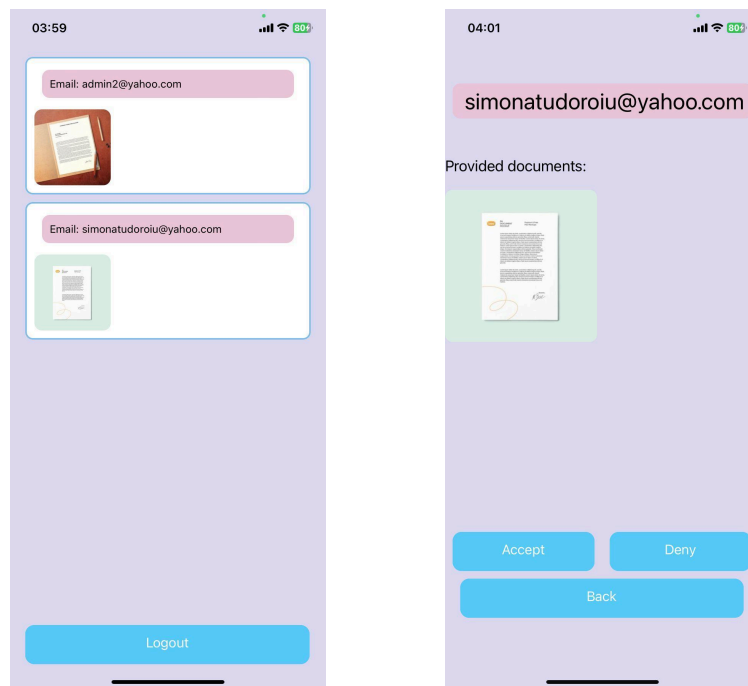
*Figura 4.4. - Primul pas din procesul de creare a contului administratorului, introducerea email-ului*

Următorul pas este realizarea fotografiilor. Integrarea camerei telefonului a fost făcută folosind librăria „expo-camera” [16], care a facilitat atât aspectul camerei, cât și cererea permisiunii de folosire a acesteia în aplicație. Există și posibilitatea încărcării fotografiilor din galeria telefonului. Pentru aceasta, a fost folosită librăria „expo-image-picker” [17], care a ajutat atât cu funcționalitatea de încărcare a pozelor din galeria telefonului, cât și cu cererea permisiunii pentru accesul la galeria foto. Logica acestei funcționalități este simplă, a fost luată poza de la client și stocată în tabela „AdministratorDocuments”, alături de alte informații relevante, și totodată a fost stocată în tabela „Administrators” o instanță care are completat doar email-ul administratorului, iar la celelalte câmpuri este regăsit șirul de caractere „pending”, pentru a determina faptul că acesta nu are încă contul aprobat.

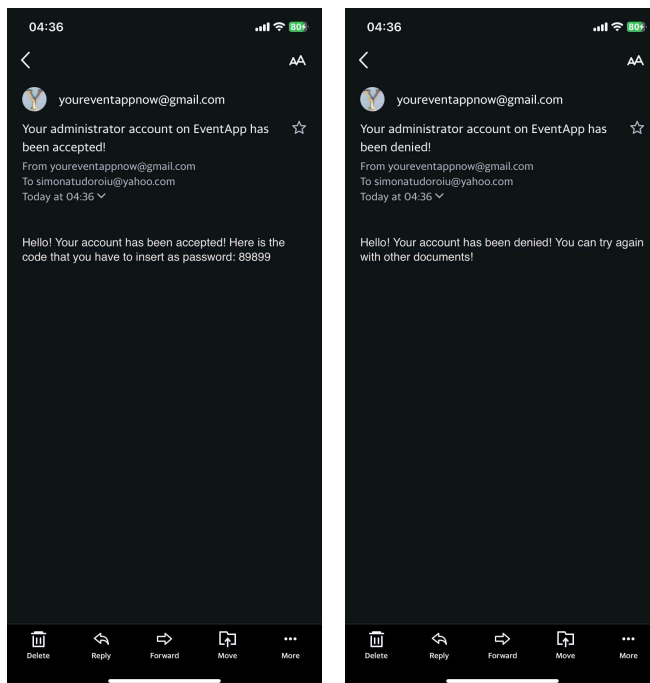


*Figura 4.5. - Al doilea pas din procesul de creare a contului administratorului, fotografierea documentelor*

Aceste documente sunt mai apoi validate de către un utilizator special, numit în cadrul acestui proiect drept administrator general. Acesta se autentifică în pagina de logare a utilizatorilor normali cu un set de credențiale speciale, pe care aplicația le recunoaște ca fiind credențialele acestui administrator general. Astfel, acesta este redirecționat către o pagină dedicată strict acceptării sau respingerii cererilor de cont cu drept de administrator de local.



*Figura 4.6. - Paginile destinate administratorului general*



*Figura 4.7. - Email-urile de acceptare și respingere*

Decizia pe care administratorul general o ia cu privire la conturile în așteptare are ca urmare trimiterea unui email pe adresa furnizată în procesul de înregistrare. În cazul respingerii contului, persoana v-a primi un email simplu în care îi este transmisă vestea. În perspectiva opusă, dacă este acceptat contul, v-a fi trimis un email în care este furnizat un cod, pe care utilizatorul v-a trebui să îl introducă în locul parolei.

Pentru trimiterea de email, a fost folosită librăria „nodemailer”, a cărei scop este fix acela de a trimite email-uri folosind Node.js. A fost nevoie de crearea unui cont de email pentru aplicație, astfel am ales gmail, datorită simplității utilizării contului în cod, fiind nevoie de generarea unei parole speciale, folosită strict în cadrul acestei aplicații.

```
function sendEmail (email, status, cod) {
  const transport = nodemailer.createTransport({
    service: 'gmail',
    auth: {
      user: 'youreventappnow@gmail.com',
      pass: 'umbj bdyd ibzm hjjm'
    }
  });
  let data;
  if(status === 'accept'){
    data = {
      from: 'youreventappnow@gmail.com',
      to: email,
      subject: 'Your administrator account on EventApp has been
accepted!',
      html: 'Hello! Your account has been accepted! Here is the
code that you have to insert as password: ' + cod
    };
  }
  else{
    data = {
      from: 'youreventappnow@gmail.com',
      to: email,
      subject: 'Your administrator account on EventApp has been
denied!',
      html: 'Hello! Your account has been denied! You can try
again with other documents!'
    };
  }

  transport.sendMail(data, (error, info) => {
    if (error) {
      console.error(error);
      res.status(500).json({
```

```

        success: false,
        message: "Error sending email",
        data: error
    });
} else {
    console.log('Email sent: ' + info.response);
    res.status(200).json({
        success: true,
        message: "Email sent successfully",
        data: info.response
    });
}
});
}

```

*Figura 4.8. - Funcția de trimitere a email-ului*

Dacă contul a fost acceptat, și utilizatorul se loghează folosind codul furnizat prin email, acesta este redirecționat către procesul de introducere a detaliilor din cont care nu au fost furnizate, și anume nume, prenume, și o altă parolă. În spate, se realizează de fapt un Update al contului, înlocuind toate câmpurile care au „pending” cu datele oferite în aceste formulare.

*Figura 4.9. - Paginile de completare a datelor contului de administrator de local*

Ultimul pas al acestui proces este dat de redirecționarea utilizatorului către pagina principală aferentă rolului pe care acesta îl deține în aplicație.

### 4.3. Home Page

După logare, utilizatorul este redirecționat către pagina principală a aplicației. Aceasta redă funcționalitatea esențială a aplicației, și anume aceea de a găsi evenimente. Astfel, în această pagină sunt redate toate evenimentele, împărțite pe categorii, pentru a fi mai ușor utilizatorului să găsească un eveniment la care ar vrea să participe, după tipul acestuia. Totodată, există și secțiunea „Hot”, unde sunt regăsite evenimentele a căror număr de locuri rezervate sunt mari, mai exact primele 5.



*Figura 4.10. - Pagina principală a utilizatorului*

Pentru a putea realiza acest lucru, am avut nevoie de mai multe cereri către server:

- o cerere în care iau primele 5 evenimente în ordine descrescătoare după variabila „reserved\_spots”, pentru evenimentele din categoria „Hot”,
- o cerere în care extrag toate categoriile regăsite printre evenimentele existente în baza de date,
- câte o cerere în care iau toate evenimentele, filtrate după fiecare categorie regăsită în

cererea anterioară.

Numărul de cereri este relativ mare, ceea ce face ca pagina să se încarce mai greu. Soluția pe care am abordat-o pentru a rezolva această problemă și pentru a scurta semnificativ timpul de încărcare a paginii este efectuarea evenimentelor asincrone simultan. Astfel, voi avea 2 funcții, „fetchHotEvents” și „fetchCategories”. Prima funcție se ocupă cu extragerea evenimentelor „hot”, iar a doua cu cererea pentru categorii, și cu cererile pentru evenimentele filtrate după aceste categorii.

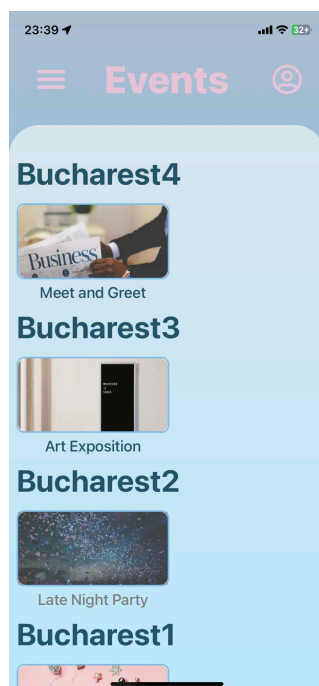
```
useEffect(() => {
    const unsubscribe = navigation.addListener('focus', async () => {
        {
            setLoading(true);
            try {
                const [hotEventsResponse, categoriesResponse] = await
Promise.all([
                    fetchHotEvents(),
                    fetchCategories()
                ]);
                setHotEvents(hotEventsResponse);
                setCategories(categoriesResponse);
            } catch (error) {
                console.error('Error loading data:', error);
            }
            setLoading(false);
        });
    });

    return unsubscribe;
}, [navigation]);
```

*Figura 4.11. - Efectuarea evenimentelor asincrone „fetchHotEvents” și „fetchCategories” simultan*

În ceea ce privește administratorul de local, pagina principală nu este foarte diferită de cea a utilizatorului. Aspectul paginii principale este identic cu cel al utilizatorului, însă lui nu îi apar toate evenimentele existente în aplicație, ci toate evenimentele pe care acesta le-a postat. Un alt aspect diferit al acestei pagini este și faptul că evenimentele sunt împărțite în

funcție de locația în care au loc, spre deosebire de pagina utilizatorului unde sunt împărțite după categoria evenimentului. Totodată, categoria „Hot” a evenimentelor nu există în pagina principală a administratorului de local, deoarece nu am găsit această funcționalitate ca una de interes pentru acesta.



*Figura 4.12. - Pagina principală a administratorului de local*

În ambele cazuri, când un eveniment s-a terminat, acesta apare diferit față de restul, pentru a se putea diferenția de evenimentele care au loc sau care vor avea loc. Astfel, acesta apare cu titlul scris într-o nuanță de gri, și poza având o peliculă gri peste ea, cum se poate observa în figura 4.12. În cazul evenimentului „Late Night Party”, pentru a se putea înțelege faptul că evenimentul nu mai este activ.

#### **4.4. Detaliile evenimentului**

La apăsarea unui eveniment de pe pagina principală, utilizatorul este redirecționat către o pagină dedicată detalierii acestuia. Astfel, pe lângă elementele prezentate în pagina principală, și anume numele, banner-ul evenimentului și categoria din care face parte, apar și alte detalii precum descrierea, data de început și de final, cu tot cu ore, numele locației unde are loc împreună cu rating-ul acesteia, numărul de locuri disponibile și prețul intrării.





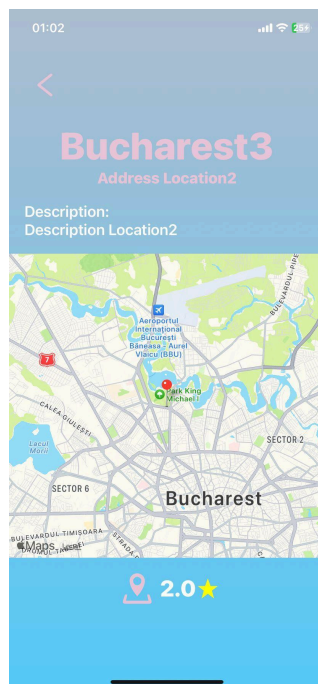
*Figura 4.13. - Pagina detalierii evenimentului*

Se pot observa în figura 4.13. două butoane care nu au fost menționate mai sus, mai precis butonul de rezervare și butonul de notificări.

Butonul „Reserve Now” prezintă, cum este și intuitiv, un formular de realizare a unei rezervări. Apăsarea acestui buton redirecționează utilizatorul către o pagină dedicată completării numărului de locuri pe care acesta dorește să îl facă, și dacă este cazul, o descriere a rezervării, în cazul în care persoana are nevoie de un tratament special.

Butonul cu iconița sub formă de clopoțel redirecționează utilizatorul către pagina de notificări. Acolo vor fi regăsite toate anunțurile pe care administratorul de local le va face cu privire la eveniment.

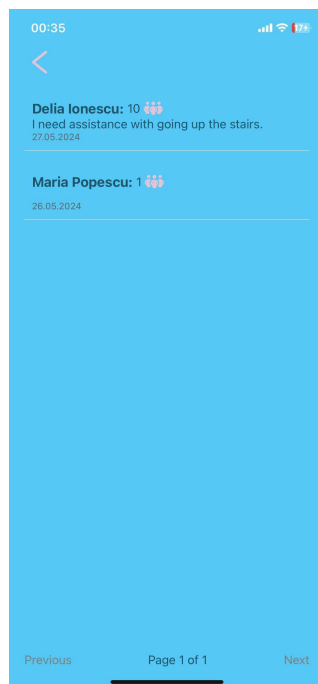
Un alt aspect important al acestei pagini este dat de faptul că la apăsarea pe numele locației, utilizatorul este redirecționat către pagina locației, unde pe lângă numele și rating-ul locației, apar adresa, descrierea și localizarea pe hartă a locației.



*Figura 4.14. - Pagina destinată locației evenimentului*

Administratorul de local are drepturi în plus față de un utilizator normal. Astfel, în ceea ce privește pagina destinată detalierii evenimentului, aceasta arată exact la fel ca cea a utilizatorului normal, cu următoarele lucruri diferite:

- la apăsarea pe câmpurile evenimentului, administratorul le poate redefini, aceste schimbări având loc doar după ce este apăsat butonul „Update Event”, care ia locul butonului „Reserve Now” al utilizatorului normal,
- butonul de adăugare a notificării, care este prezent sub forma unui clopoțel care are și un plus, fiind redirecționat către pagina aferentă, unde trebuie completat mesajul notificării,
- butonul de ștergere a evenimentului, reprezentat de butonul sub forma unui coș de gunoi,
- butonul de inbox, care redirecționează administratorul de local către o pagină în care sunt prezentate toate rezervările făcute la evenimentul respectiv, incluzând cine a făcut rezervarea, câte persoane sunt și, dacă este cazul, cerințele speciale în ceea ce privește rezervarea.



*Figura 4.15. - Pagina destinată vizionării rezervărilor aferente unui eveniment*

## 4.5. Harta evenimentelor

Pentru utilizatorul normal, în secțiunea „Menu” a aplicației se poate observa opțiunea „See Events Map”, care are ca scop vederea evenimentelor sub o altă formă, și anume după localizarea acestora pe hartă. Această pagină vine în ajutorul utilizatorilor prin găsirea eficientă a evenimentelor în funcție de distanța de la locația acestora. Astfel, pe lângă marcarea locațiilor evenimentelor, este marcată și locația utilizatorului curent. Pentru acest lucru, a fost nevoie de cererea permisiunii de preluare a locației curente a dispozitivului de pe care aplicația rulează, acest lucru fiind realizat folosind plugin-ul din „expo-location” [18], și anume „locationAlwaysAndWhenInUsePermission”. La randarea paginii, în cazul în care utilizatorul nu și-a dat acceptul deja, acestuia îi este cerut printr-un pop-up.

Harta a fost realizată cu ajutorul bibliotecii „react-native-maps” [19], de unde a fost folosit MapView pentru randarea hărții propriu-zise, și Marker, pentru a marca locațiile importante pe aceasta. Tot din această librărie a fost folosit și Polyline, pentru a marca drumul de la locația utilizatorului, până la locația evenimentului.

```

<View style={styles.container}>
  {location && mapReady ? (
    <>
      <View style={styles.titleContainer}>
        <TouchableOpacity
          onPress={() => navigation.goBack()}
          hitSlop={{ top: 10, bottom: 10, left: 10, right: 10 }}
          style={[styles.iconTouch, styles.leftIcon, { padding: 30
}}}
        <
          <AntDesign name="left" size={40} color={COLORS.pink} />
        </TouchableOpacity>
        <Text style={styles.titleText}>Events Map</Text>
      </View>
      <MapView
        style={styles.map}
        initialRegion={location}
        showsUserLocation={true}
        followUserLocation={true}
        showsMyLocationButton={true}
      >
        {markers.map(marker => (
          <Marker
            key={marker.id.toString()}
            coordinate={{ latitude: marker.latitude, longitude:
marker.longitude }}
            title={marker.title}
            onPress={() => {
              getDirections(location, { latitude: marker.latitude,
longitude: marker.longitude });
            }}
            onCalloutPress={() =>
navigation.navigate('EventDetailed', marker.id)}
          />
        ))}
        {routeLines.length > 0 && (
          <Polyline
            coordinates={routeLines}
            strokeWidth={5}
            strokeColor="red"

```

```

        />
    })
    </MapView>
</>
) : (
    <ActivityIndicator size="large" color={COLORS.pink}
style={styles.spinner} />
    )
</View>

```

Figura 4.16. - Codul folosit în frontend pentru randarea hărții

O funcționalitate importantă a acestei hărți este marcarea drumului de la locația utilizatorului până la locația evenimentului, pe care acesta ar trebui să îl parcurgă cu mașina. Acest lucru se întâmplă când se apasă o dată pe marcajul evenimentului, iar dacă marcajul este apăsat de două ori, utilizatorul este redirecționat către pagina evenimentului detaliat. Pentru această funcționalitate a fost folosit API-ul de la OpenRouteService [20], deoarece acesta oferă un număr mare de cereri pe zi pe gratis, și anume 7000.

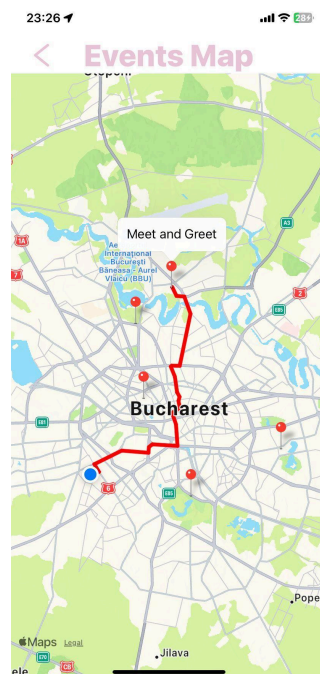


Figura 4.17. - Pagina hărții interactive

Această funcționalitate nu există și în cazul administratorului de local, dat fiind faptul că acesta nu are nevoie să vadă unde se află toate evenimentele, el având în grijă aceste locații.

## 4.6. Calendar

În aplicația „EventApp”, fiecare utilizator are propriul său calendar personalizat. Personalizarea reiese din faptul că acest calendar se schimbă odată cu realizarea unei noi rezervări, care mai apoi este stocată în cadrul acestuia, alături de zilele evenimentului, numele, orele de început și de final și locația aferentă.

Calendarul a fost realizat folosind Agenda din librăria „react-native-calendars” [21]. Aceasta, cum îi spune și numele, randează un calendar în stil agendă, adică pe lângă funcționalitatea principală a calendarului, și anume aceea de a arăta zilele, săptămânile, lunile și anii, pentru fiecare zi în parte există o agendă separată unde, în cadrul „EventApp”, utilizatorul își organizează mai ușor timpul, fiind mereu la un click distanță de a afla ce evenimente are pregătite în ziua respectivă. Dacă utilizatorul vrea să vadă mai multe detalii despre evenimentul ce are loc sau urmează să aibe loc, acesta poate apăsa pe căsuța destinată evenimentului din agendă, și este redirecționat către pagina evenimentului.

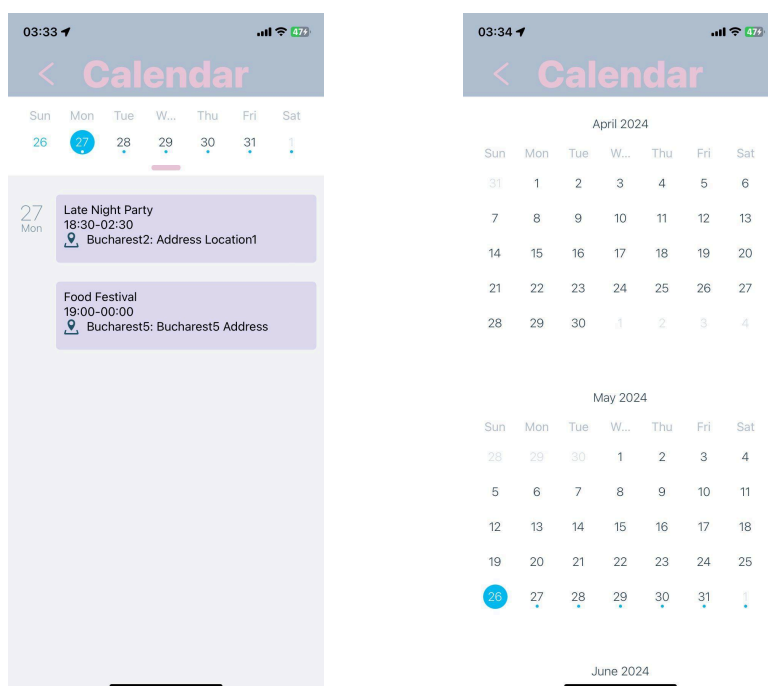


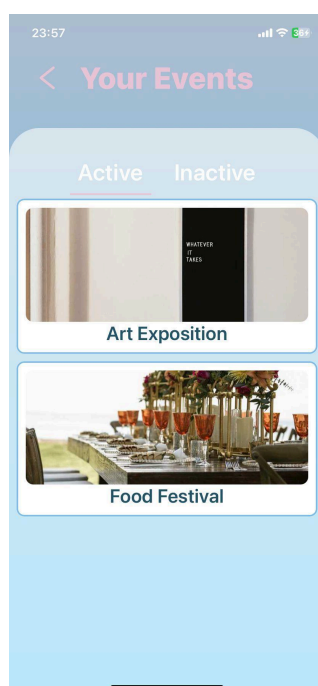
Figura 4.18. - Agenda și calendarul utilizatorului

Acest tip de calendar este prezent și în cazul administratorilor de local. Din punctul meu de vedere, pentru aceștia este mai de ajutor câte un calendar separat pentru fiecare locație, facilitând organizarea evenimentelor bazate pe fiecare local în parte. Așadar, în meniul aplicației, la secțiunea „See Calendars”, administratorul este nevoit să aleagă locația pentru care dorește să vadă calendarul, iar în agendă apare doar numele evenimentului și orele între care se desfășoară, la apăsarea evenimentului acesta fiind redirecționat către pagina evenimentului.

## 4.7. Pagina de evenimente

Pentru eficiența găsirii evenimentelor la care utilizatorul urmează să participe, sau a participat deja, „Event App” oferă această facilitate prin pagina „Your Events”, care se poate găsi în meniul aplicației. Această pagină distinge evenimentele active de cele inactive, adică cele care vor avea sau au deja loc, de cele care s-au terminat.

Astfel, pentru evenimentele active, utilizatorul va vrea să revadă informațiile acestuia, deci la apăsarea pe eveniment acesta este redirecționat la pagina evenimentului respectiv.



*Figura 4.19. - Pagina de evenimente active a utilizatorului*

În ceea ce privește un eveniment inactiv, la care utilizatorul a participat, acesta poate da o recenzie, care se extinde ca recenzie la locația respectivă. Astfel, la apăsarea pe unul dintre evenimentele inactive, utilizatorul este redirecționat către pagina de oferire a unei recenzii, unde este rugat să atribuie un număr de stele de la 1 la 5, și un comentariu cu privire la organizarea evenimentului. Acest lucru are ca scop informarea celorlalți utilizatori cu privire la organizarea unui eveniment la o anumită locație, din acest motiv recenzia se transpune locației. O medie a recenziilor poate fi găsită în pagina unui eveniment, la secțiunea destinată locației, unde pe lângă numele propriu-zis, apare și recenzia, ca în figura 4.13. prezentată în subcapitolul 1.4. al capitolului prezent.

Mai jos este prezentată funcția din controller-ul tabelii Reviews care se ocupă cu salvarea unei noi intrări în tabel, și totodată actualizarea mediei recenziilor pentru intrarea aferentă din tabela Locations.

```
const saveReview = async (req, res) => {
  const corpRecenzie = req.body;
  try {
    const recenzie = await Review.create({
      user_id: req.user.id,
      ...corpRecenzie});
    const { avg } = await Review.findOne({
      where: {
        event_id: recenzie.event_id
      },
      attributes:
[[Review.sequelize.fn('AVG', Review.sequelize.col('rating')), 'avg']],
      raw: true
    });

    await Event.update({review_summary: avg}, {
      where: {
        event_id: recenzie.event_id
      }
    });

    const eveniment = await Event.findOne({
      where: {
        event_id: recenzie.event_id
      }
    })
  }
}
```



```

    });

    const { avg_event } = await Event.findOne({
      where: {
        location_id: eveniment.location_id
      },
      attributes:
        [[Event.sequelize.fn('AVG', Event.sequelize.col('review_summary')),
'avg_event']],
      raw: true
    });

    await Location.update({review_summary: avg_event}, {
      where: {
        location_id: eveniment.location_id
      }
    });

    res.status(200).json({
      success: true,
      message: "Review created successfully",
      data: recenzie
    });
  }
  catch (err) {
    console.log('err', err);
    res.status(400).json({
      success: false,
      message: "Error creating review",
      data: err
    });
  }
}

```

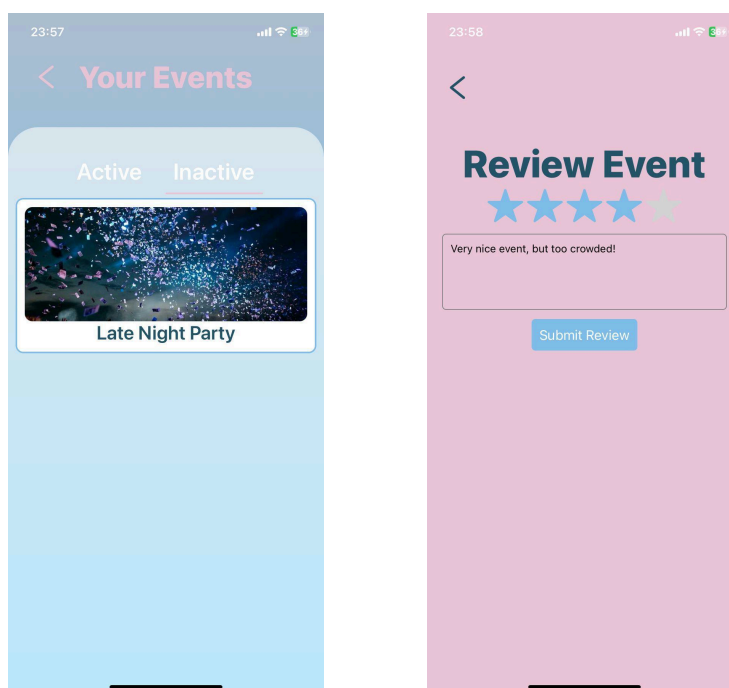
*Figura 4.20. - Funcția de salvare a unei intrări în tabela Reviews*

Așa cum am specificat și mai sus, codul atașat are ca scop salvarea unei noi recenzii. Așadar, primul lucru pe care codul îl face este de a salva intrarea în tabela Reviews.

Următorul pas este de a găsi media tuturor recenziilor ce sunt aplicate aceluiași eveniment ca recenzia nou introdusă. Astfel, este selectată media aritmetică a numerelor ce se află în coloana rating, reprezentând numărul de stele pe care utilizatorul l-a acordat

evenimentului.

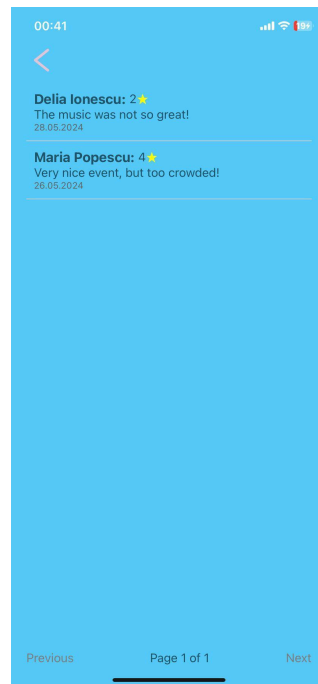
Recenziile introduse în tabela Reviews trebuie să se reflecte și în tabela Locations, în intrările corespunzătoare. Așadar, în final, este actualizată intrarea din tabela Locations ce corespunde cu evenimentul la care utilizatorul a oferit o recenzie cu o nouă medie aritmetică, obținută prin coloana review\_summary corespunzătoare tuturor evenimentelor ce au loc în locația respectivă.



*Figura 4.21. - Pagina evenimentelor inactive și pagina de postat o recenzie*

Pentru administratorul de local lucrurile nu stau foarte diferit. Această pagină prezintă toate evenimentele postate de acesta, și la fel ca la utilizatorul normal sunt împărțite în două categorii: cele active și cele inactive. La apăsarea unui eveniment activ, administratorul este redirecționat către pagina detaliilor acestuia, unde poate fi modificat.

Una dintre diferențe se regăsește la evenimentele inactive, unde la apăsarea unuia dintre ele, administratorul este redirecționat către pagina tuturor recenziilor date la respectivul eveniment. Astfel, în această pagină este prezentat numele utilizatorului care a lăsat recenzia, numărul de steluțe pe care l-a oferit și comentariul dat. Această pagină are ca scop luarea la cunoștință de către administrator de posibilele probleme apărute în cadrul unui eveniment gestionat de acesta, dar totodată și lucrurile bune care a atras oamenii către acest eveniment.



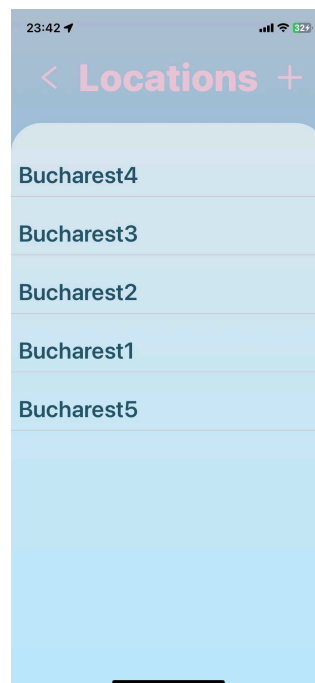
*Figura 4.22. - Pagina cu toate recenziile a administratorului*

*Figura 4.23. - Formularul de adăugare a unui eveniment*

O altă diferență este dată de butonul de adăugare de eveniment, prezent în colțul din dreapta sus al paginii evenimentelor, sub forma unui plus. Acesta este prezent doar pentru administratorii de local, oferind posibilitatea postării unui eveniment, fiind nevoiți să completeze un formular cu detaliile acestuia, și anume numele, descrierea, categoria din care face parte, locurile disponibile, prețul intrării, data și ora de început și de final, locația unde are loc, iar în următoarea pagină fiind nevoiți să încarce o poza cu banner-ul evenimentului.

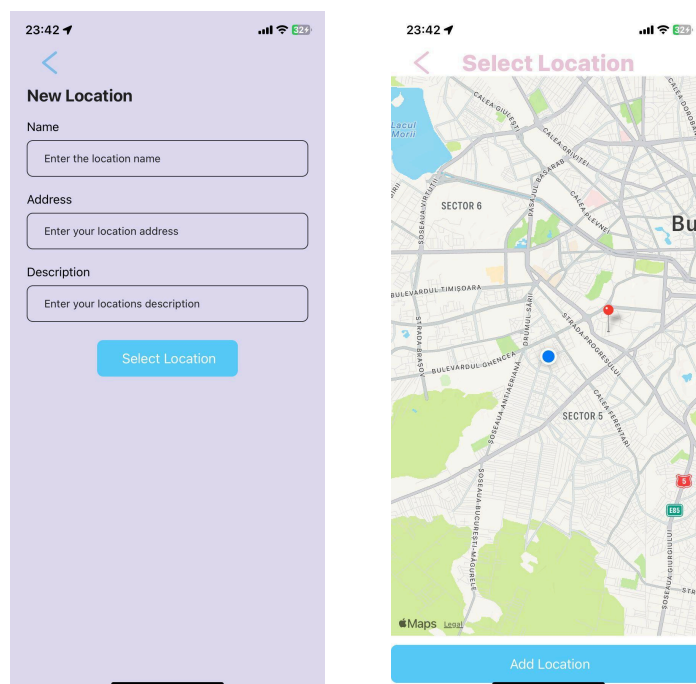
## 4.8. Pagina de locații

Pagina de locații este destinată administratorilor de local, fiind locul unde pot găsi toate locațiile pe care le dețin, să schimbe anumite câmpuri de care este nevoie, să adauge o locație sau să o șteargă.



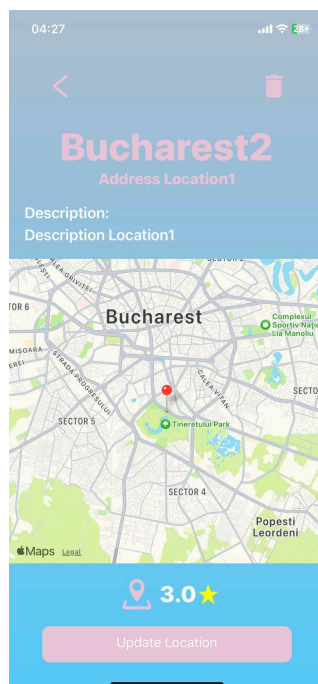
*Figura 4.24. - Pagina locațiilor*

Pentru adăugarea unei noi locații, administratorul de local este nevoit să apese pe butonul din colțul din dreapta sus, care este sub forma unui plus, unde este nevoit să introducă numele, adresa și descrierea locației, iar în pagina următoare să selecteze punctul de pe hartă unde se află această locație.



*Figura 4.25. - Paginile destinate adăugării unei noi locații*

La apăsarea unei locații din pagina destinată acestora, administratorul este redirecționat către pagina de detaliere a locației, acolo unde poate redefini anumite câmpuri care dorește a fi schimbate.



*Figura 4.26. - Pagina locației administratorului de local*

## 4.9. Pagina contului

În pagina principală se poate observa în colțul din dreapta sus o iconiță, care la apăsare, redirecționează utilizatorul către pagina contului. Această pagină este dedicată atât schimbării detaliilor contului, cât și ștergerii lui sau deconectarea din aplicație. Funcționalitatea este prezentă atât pentru utilizatorul normal, cât și pentru administratorul de local.

Există câte o pagină special dedicată schimbării fiecărui câmp din baza de date asociat unui cont, astfel oferind o abordare simplă de actualizare a contului, nefiind nevoie de introducerea altor date pe care utilizatorul nu le vrea a fi schimbate.

În situația în care utilizatorul decide că nu îi mai este de folos această aplicație, există opțiunea ștergerii contului. La apăsarea butonului „Delete account” din pagina contului, utilizatorul este redirecționat către o pagină de confirmare, astfel excluzând posibilitatea ștergerii contului din greșeală, acesta fiind întrebat dacă este sigur că dorește să facă acest pas.

Butonul de deconectare din aplicație redirecționează utilizatorul către pagina de început, și șterge din AsyncStorage [22], care este un sistem de stocare local asincron folosit în aplicațiile React Native, toate datele stocate până atunci, cum ar fi token-ul de autentificare.

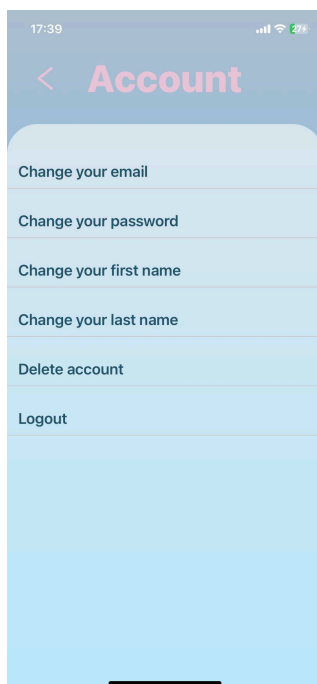


Figura 4.27. - Pagina contului de utilizator

## 4.10. Pagina de contact

În această aplicație am luat în vedere și posibila întâmplare a unor erori neprevăzute, pe care utilizatorii le pot întâmpina. Având în vedere faptul că nu orice problemă poate fi descoperită în perioada de dezvoltare a aplicației, sau nu este atât de comună încât să fie prevăzută, am dezvoltat și o pagină de contact, special concepută pentru astfel de momente. Așadar, în cazul în care un utilizator, fie el administrator de local sau utilizator normal, întâmpină o problemă în aplicație care nu este tratată, acesta poate intra pe pagina „Contact Us”, ce se găsește în meniul aplicației, unde este nevoit să completeze email-ul de pe care să continue discuția mai departe, și problema în speță.

*Figura 4.27. - Pagina de contact*

Având în vedere ca nu am putut să transpun ideea din formular și în cod, și anume să trimit un email de pe contul utilizatorului fără a dispune de credențialele acestuia, în backend se întâmplă următoarele: se creează un transport din librăria „nodemailer” folosind credențialele contului de gmail ale aplicației, se trimite un mail de la contul aplicației către el însuși, folosind funcția `sendMail()`, în care corpul mail-ului este mesajul scris de utilizator, iar în opțiunea `replyTo` este email-ul utilizatorului care a ridicat problema, astfel atunci când administratorul aplicației „EventApp” vrea să răspundă acestei probleme, să primească acest răspuns fix utilizatorul care a avut problema în primă fază.

```

const sendEmail = async (req, res) => {
  const { email, message } = req.body;
  const transport = nodemailer.createTransport({
    service: 'gmail',
    auth: {
      user: 'youreventappnow@gmail.com',
      pass: 'umbj bdyd ibzm hjjm'
    }
  });

  const optiuniEmail = {
    from: 'youreventappnow@gmail.com',
    to: 'youreventappnow@gmail.com',
    subject: 'New Contact Us Message',
    text: `Message: ${message}`,
    replyTo: email
  };

  transport.sendMail(optiuniEmail, function(error, info){
    if (error) {
      console.log(error);
      res.status(500).send('Error sending email');
    } else {
      console.log('Email sent: ' + info.response);
      res.send('Email sent successfully');
    }
  });
};

```

*Figura 4.28 - Funcția de gestionare a contactării administratorului aplicației*



# Capitolul 5

## Concluzii și îmbunătățiri viitoare

### 5.1. Concluzii

Proiectul de licență prezentat în această lucrare expune atât ideea din spatele aplicației mobile de gestionare de evenimente „EventApp”, cât și procesul de realizare a acesteia. Precum a fost prezentat și în introducerea lucrării, aplicația mobilă are 4 mari funcționalități, și anume postarea de evenimente de către administratorii de local, vizionarea acestora de către utilizatori, rezervarea de locuri la aceste evenimente, și nu în ultimul rând, harta interactivă, care dă o notă de unicitate aplicației.

Această aplicație are un potențial foarte mare, mai ales în rândul tinerilor, care vor să își petreacă timpul liber într-un mod calitativ, „EventApp” oferindu-le un spațiu destinat fix acestui lucru. Însă acest lucru nu este o limitare, aplicația fiind destinată persoanelor de toate vârstele ce vor un timp liber mai social, oferind o escapadă din viața cotidiană monotună.

„EventApp” este o aplicație specială pentru mine, fiind prima aplicație mobilă pentru care am lucrat de la 0, dezvoltând atât ideea, cât și partea practică, și anume backend-ul, frontend-ul și baza de date. Totodată, „EventApp” a fost o provocare dusă la bun sfârșit, având în vedere că până la începerea proiectului de licență nu am mai lucrat cu React Native. Pe tot parcursul acestor luni de lucru intens la prezentul proiect, mă pot afirma spunând că am dobândit o nouă cunoștință valoroasă în ceea ce privește dezvoltarea unei interfețe grafice de mobil, dar am aprofundat și totodată am pus în practică cunoștințele deja existente în Node.js și dezvoltarea efectivă a unui server pentru această aplicație, plus folosirea unei baze de date noi pentru mine, și anume MySQL.

Sunt mândră de produsul finit obținut în finalul acestor luni, iar următorul pas la care aspir să ajung în ceea ce privește acest proiect este publicarea aplicației „EventApp”, pentru ca tinerii, și nu numai, să se poată bucura de ea, astfel oferindu-le o platformă destinată timpului liber de calitate, în apropierea persoanelor ce dețin pasiuni comune.

## 5.2. Îmbunătățiri viitoare

Proiectul de licență reprezentat de aplicația mobilă de gestionare de evenimente „EventApp” se vrea a fi un „proof-of-concept”, faptul că ideea este realizabilă, astfel funcționalitățile principale ale aplicației fiind deja dezvoltate. Ca orice proiect, aplicația mobilă prezentată lasă loc de posibile dezvoltări viitoare, printre acestea numărându-se următoarele:

- logarea ar putea beneficia de posibilitatea realizării acestui lucru folosind un alt cont al utilizatorului, deja existent, din diferite rețele de socializare precum Facebook, Google sau altele,
- implementarea unui sistem de cumpărare a biletelor la eveniment, nu numai de rezervare a locului, astfel fiind o soluție pentru cozile mari la cumpărarea biletelor fizice, generându-se un cod QR unic pentru fiecare bilet care să fie scanat la intrarea în respectivul eveniment,
- extinderea hărții interactive prin afișarea tuturor modalităților posibile de a ajunge la un anumit eveniment, cu tot cu timpul de parcurgere a drumului,
- implementarea unui sistem de chat live, activ în timpul evenimentului, pentru ca participanții să poată împărtăși informații precum: unde sunt locurile mai aglomerate, când sosește artistul principal, prețurile la mâncare din cadrul evenimentului, pentru a crea o experiență cât mai confortabilă,
- testarea pe mai multe dispozitive. Având în vedere ca am avut la dispoziție doar un dispozitiv Apple, și anume un Iphone 11, se poate testa aplicația și pe alte dispozitive, cum ar fi dispozitive ce rulează Android, astfel făcându-se schimbări la interfață, dacă este cazul,
- implementarea unui sistem 100% automat în ceea ce privește verificarea eligibilității administratorului de local.

Prin lista de mai sus am demonstrat faptul că aplicația „EventApp” are un potențial mare de dezvoltare. Cum am specificat la începutul acestui subcapitol, starea actuală a aplicației prezintă ideea principală ce s-a vrut a fi realizată, însă cu micile dezvoltări ulterioare prezentate, ar avea un impact mai mare asupra utilizatorului.

# BIBLIOGRAFIE

- [1] “Index,” *Node.js v22.2.0 Documentation*. <https://nodejs.org/docs/latest/api/> (accessed May 30, 2024).
- [2] “About npm,” *npm Docs*. <https://docs.npmjs.com/about-npm> (accessed May 30, 2024).
- [3] “JWT.IO - JSON Web Tokens Introduction,” *Auth0*. <https://jwt.io/introduction> (accessed May 31, 2024).
- [4] A. Reinman, “Nodemailer :: Nodemailer.” <https://nodemailer.com/> (accessed May 31, 2024).
- [5] C. Hale, “How To Safely Store A Password,” *codahale.com*, Jan. 31, 2010. <https://codahale.com/how-to-safely-store-a-password/> (accessed May 31, 2024).
- [6] “Documentation,” *Dotenv*. <https://www.dotenv.org/docs/> (accessed May 31, 2024).
- [7] “Express,” *Node.js web application framework*. <https://expressjs.com/> (accessed May 31, 2024).
- [8] “Express multer middleware.” <https://expressjs.com/en/resources/middleware/multer.html> (accessed May 31, 2024).
- [9] “Quickstart,” *Quickstart*. <https://sidorares.github.io/node-mysql2/docs> (accessed May 31, 2024).
- [10] “Sequelize,” *Feature-rich ORM for modern TypeScript & JavaScript*. <https://sequelize.org/> (accessed May 31, 2024).
- [11] “nodemon.” <https://nodemon.io/> (accessed May 31, 2024).
- [12] “React Native · Learn once, write anywhere.” <https://reactnative.dev/> (accessed May 31, 2024).
- [13] “Expo Documentation,” *Expo Documentation*. <https://docs.expo.dev/> (accessed May 31, 2024).
- [14] “MySQL :: MySQL Documentation.” <https://dev.mysql.com/doc/> (accessed May 31, 2024).
- [15] J. Terra, “What is Client-Server Architecture? Everything You Should Know,” *Simplilearn*, Apr. 22, 2022. Available: <https://www.simplilearn.com/what-is-client-server-architecture-article> (accessed May 31, 2024).

- [16] “Camera,” *Expo Documentation*. <https://docs.expo.dev/versions/latest/sdk/camera/> (accessed May 31, 2024).
- [17] “ImagePicker,” *Expo Documentation*. <https://docs.expo.dev/versions/latest/sdk/imagepicker/> (accessed May 31, 2024).
- [18] “Location,” *Expo Documentation*. <https://docs.expo.dev/versions/latest/sdk/location/> (accessed Jun. 01, 2024).
- [19] G. Singhal, “Introduction to React Native Maps,” *LogRocket Blog*, Oct. 18, 2021. <https://blog.logrocket.com/react-native-maps-introduction/> (accessed Jun. 01, 2024).
- [20] “openrouteservice.” <https://openrouteservice.org/> (accessed Jun. 01, 2024).
- [21] “Intro,” *RNC*. <https://wix.github.io/react-native-calendars/docs/Intro> (accessed Jun. 01, 2024).
- [22] “Usage,” *Async Storage*. <https://react-native-async-storage.github.io/async-storage/docs/usage/> (accessed Jun. 01, 2024).
- [23] “JavaScript Guide - JavaScript,” *MDN Web Docs*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (accessed Jun. 01, 2024).