

1	Introducere.....	3
1.1	Contextul proiectului.....	3
1.1.1	Contextul general.....	3
1.1.2	Contextul specific.....	3
1.2	Structura lucrării pe capitole	4
2	Obiectivele proiectului	5
3	Studiu bibliografic	7
3.1	Rețelele sociale.....	7
3.2	Studiul promovării unui eveniment.....	8
3.3	Sisteme similare	9
4	Analiză si fundamentare teoretică	12
4.1	Scenariile aplicației	12
4.2	Cerințele aplicației.....	13
4.2.1	Cerințele funcționale.....	13
4.2.2	Cerințele non-funcționale	14
4.3	Cazuri de utilizare	14
4.3.1	Diagrama Use Case pentru utilizatorul aplicației	15
4.4	Tehnologii utilizate	19
4.4.1	Platforma Android	19
4.4.1.1	Descrierea platformei	19
4.4.1.2	Arhitectura platformei Android.....	20
4.4.1.3	Componentele unei aplicatii Android.....	23
4.4.2	Google Maps API	26
4.4.3	JSON.....	26
4.4.4	GSON	27
4.4.5	Stripe referinte bibliografice !!!1.....	28
4.4.6	OrmLite	28
4.5	Pattern-uri arhitecturale.....	29
5	Proiectare de detaliu și implementare.....	31
5.1	Arhitectura sistemului	31
5.1.1	Atributiile componentelor sistemului	32
5.1.1.1	Aplicatia server.....	32

5.1.1.2	Aplicatia client.....	32
5.2	Arhitectura aplicatiei client	33
5.2.1	Nivelul de comunicatie si servicii	34
5.2.2	Nivelul de conversise.....	34
5.2.3	Nivelul de modele si entitati.....	34
5.2.4	Nivelul de management al bazei de date	35
5.2.5	Nivelul de prezentare.....	35
6	Testare și validare.....	37
7	Manual de instalare și utilizare	40
7.1	Resurse hardware	40
7.2	Resurse software	40
7.3	Instalarea aplicației.....	40
7.4	Utilizarea aplicației	42
7.4.1	Lansarea aplicației	42
7.4.2	Prima utilizare a aplicației	42
7.4.3	Înregistrare și logare	42
7.4.4	Utilizarea meniului lateral	44
7.4.5	Vizualizarea evenimentelor	44
7.4.6	Vizualizarea detaliilor despre un eveniment	45
7.4.7	Cumpărarea biletelor	46
7.4.8	Vizualizarea biletelor achiziționate	46
7.4.9	Postarea de comentarii pe pagina unui eveniment.....	48
7.4.10	Căutarea evenimentelor	48
7.4.11	Vizualizarea și modificarea categoriilor de evenimente de interes	49
7.4.12	Vizualizarea evenimentelor adaugate in „wishlist”	50
7.4.13	Vizualizarea propriilor evenimente	50
7.4.14	Setările aplicației	51
8	Concluzii.....	53
8.1	Realizari	53
8.2	Dezvoltari ulterioare.....	53

1 Introducere

1.1 Contextul proiectului

Acest capitol prezintă contextul general al tehnologiilor mobile, continuând cu expunerea contextului specific, care constituie și motivarea dezvoltării unei aplicații mobile de tip „event planner”.

1.1.1 Contextul general

Astăzi, mai mult că oricând, asistăm la o evoluție impresionantă a tehnologiilor lansate pe piața IT și putem fi beneficiarii multor dintre inovațiile aduse de companiile de profil. Despre noutățile aduse de-a lungul ultimilor ani în domeniul IT se poate spune că au încurajat transpunerea multor dintre facilitățile existente anterior doar pe mediile desktop, într-o zonă mobile, care a revoluționat modul în care se realizează comunicația.

Observând oportunitatea oferită de acest sector, s-au detașat pe piață 3 mari competitori, cu sisteme de operare proprii pentru dispozitive mobile: compania Apple cu sistemul de operare iOS, compania Google cu sistemul de operare Android și nu în ultimul rând, compania Microsoft cu sistemul de operare Windows Phone. Platforma mobilă Android este cea mai populară fiind utilizată în sute de milioane de dispozitive mobile în mai mult de 190 de țări din întreaga lume, iar printre principalele avantaje ale sale se regăsesc portabilitatea, extensibilitatea, posibilitatea integrării serviciilor oferite de Google.

Zona mobile oferă flexibilitate, datorită portabilității și posibilității de accesare a informației de oriunde și oricând. Mai mult, este o zonă care își dovedește zilnic capabilitățile prin modul în care ne ajută să rezolvăm sarcini de la cele mai ușoare la cele mai complicate: organizarea cumpărăturilor, organizarea mail-urilor, orientare în timp real prin GPS, plăți online, Internet banking și multe altele. Prin urmare, aplicațiile mobile sunt de un real ajutor, asistând utilizatorul în multe dintre activitățile realizate de acesta.

Ca și orice sistem existent, și tehnologia mobilă ascunde unele imperfecțiuni, contrabalansate de avantajele menționate mai sus. Printre dezavantajele mediilor mobile, trebuie menționate următoarele: puterea de calcul mai redusă a acestor dispozitive, durata mai scurtă de viață a bateriei încorporate în dispozitivele mobile, dimensiunile reduse ale ecranului. Acești factori sunt hotărâtori în proiectarea unei aplicații mobile, deoarece impun anumite limitări care trebuie respectate, însă reprezintă o provocare pentru dezvoltatorii software, care își propun o continuă optimizare a produselor software.

Pornind de la aceste premize, proiectul actual își propune să fie unul de actualitate, care să capteze tendițele tehnologice din momentul de față și să consitutie o aplicație utilă pentru utilizatorii țintă.

1.1.2 Contextul specific

În continuarea acestui capitol va fi expusă motivarea alegerii temei pentru proiectul actual, cât și modalitatea prin care aceasta se integrează în necesitățile omului modern, dinamic, conectat la informație, pentru care resursa temporală este vitală.

Trăim într-o societate în care avem posibilitatea de a explora lumea într-un mod creativ și lipsit de bariere, în care ne putem exprima părerile cu privire la tematicile ce ne interesează și vrem să fim la curent cu ceea ce se întâmplă în jurul nostru. Dinamismul, ritmul alert în care se desfășoară lucrurile în societatea actuală ne-a făcut să acordăm o prioritate mare organizării cât mai eficientă a timpului, menținerii unui echilibru între viața profesională și cea personală, dar și interacțiunii cu prietenii, nu doar în mediul virtual, cât și în mediul real. Așadar, se poate identifica un scenariu major în care tehnologia ne-ar putea îmbunătăți experiențele zilnice, interacțiunea, contactul interuman: dorința de a organiza timpul liber, prin participarea la evenimente sociale.

Având în vedere cele enunțate anterior, s-a decis crearea unei aplicații, sub forma unei rețele sociale, care să ruleze pe platforma Android. Aplicația oferă posibilitatea vizualizării de evenimente, participarea la aceste evenimente, căutarea evenimentelor aplicând diferite filtre, cumpărarea de bilete pentru evenimentele respective, prin integrarea unui sistem de plată online de ultimă generație, precum și vizualizarea locației evenimentului pe hartă, prin intermediul Google Maps. Sistemul este gândit similar unei rețele sociale deoarece pune la dispoziția utilizatorilor oportunitatea de a comenta evenimentele la care au participat, la care au cumpărat bilete sau la care sunt gazde, pot fi notificați de acțiunile unui alt utilizator pentru care a realizat acțiunea de tip „follow”. În plus, popularitatea rețelei de socializare Facebook a condus la decizia de a oferi utilizatorilor posibilitatea să utilizeze aplicația cu un cont valid de Facebook, având astfel foarte multe beneficii, printre care: posibilitatea de a opera „share” pentru un eveniment, adică de a răspândi informații despre evenimente și prietenilor de pe rețeaua socială Facebook.

1.2 Structura lucrării pe capitole

Documentul curent este structurat în 9 capitole, după cum urmează:

- capitolul 1 constituie introducerea și detalierea contextului problemei
- capitolul 2 prezintă obiectivele proiectului
- capitolul 3 descrie studiul bibliografic necesar familiarizării cu domeniul temei având drept scop identificarea și analiza de sisteme similare
- capitolul 4 realizează o analiză și fundamentare teoretică în ceea ce privește cerințele sistemului, cât și tehnologiile necesare implementării
- capitolul 5 abordează etapele de proiectare ale proiectului
- capitolul 6 prezintă aspectele referitoare la testarea și validarea produsului software
- capitolul 7 urmărește întocmirea unui manual de utilizare al sistemului
- capitolul 8 prezintă concluziile deduse din urma realizării proiectului propus.

2 Obiectivele proiectului

În acest capitol se vor detalia aspecte legate de obiectivele sistemului.

Kweekweek, sistemul proiectat și implementat, este un produs care își propune să vină în ajutorul utilizatorului, profilul căruia îl putem asocia cu o persoană interesată de organizarea eficientă a timpului liber, de participarea la evenimente alături de prietenii săi, și care dorește totodată ca pregătirea acestor acțiuni să nu fie „time-consuming”. Omofonia denumirii produsului „KweekWeek” cu sintagma „quick week” nu este întâmplătoare, pentru că numele produsului software își are originea chiar în ideea de planificare rapidă, dinamism.

În continuare sunt prezentate un set de obiective ale aplicației:

- **Vizualizarea evenimentelor**

Sistemul pune la dispoziția utilizatorului vizualizarea unei liste de evenimente după anumite filtre, care îl vor ajuta să descopere, de pildă, evenimente ce au loc în ziua accesării aplicației, în ziua următoare, în funcție de categoria în care se încadrează evenimentul sau în funcție de popularitate.

- **Vizualizarea detaliilor despre un eveniment**

Utilizatorul interesat de un eveniment afișat în lista evenimentelor poate accesa acel eveniment pentru a vizualiza detalii despre acel eveniment. Detaliile evenimentului includ: una sau mai multe poze aferente evenimentului, o descriere a acestuia, prețul participării la acel eveniment, data și ora începerii evenimentului, data și ora încheierii evenimentului, locația desfășurării evenimentului, informații despre gazda evenimentului, categoriile din care face parte evenimentul. Tot în această funcționalitate se încadrează și vizualizarea numărului de comentarii postate în cadrul evenimentului, precum și numărul de participanți la acel eveniment.

- **Vizualizarea comentariilor postate pe pagina unui eveniment și postarea de comentarii**

Aplicația este interactivă, permițând utilizatorului să acceseze comentariile unui eveniment și în plus să posteze comentarii în cadrul evenimentelor la care a cumpărat bilet, sau la care este gazdă. Această funcționalitate permite utilizatorului să vizualizeze comentariile utilizatorilor, însoțite de pozele acestor utilizatori și data la care aceștia au postat comentariile. Atunci când utilizatorul postează un comentariu, lista este actualizată. Comentariile sunt afișate în funcție de data postării lor, astfel că primul comentariu afișat va fi cel mai recent pentru acel eveniment.

- **Achiziționarea de bilete pentru evenimente**

O altă funcționalitate a aplicației KweekWeek este aceea de plată a biletelor pentru evenimentele preferate, prin intermediul unui sistem stabil de plăți online. Sistemul integrat este Stripe, care oferă suport și pentru dispozitivele mobile iOS sau Android. Stripe este un sistem sigur, ușor de utilizat, care face managementul plăților online, asigurând totodată securitatea tranzacțiilor și a datelor implicate în acestea. Această funcționalitate este accesibilă de pe pagina unui eveniment.

- **Vizualizarea locației evenimentelor prin intermediul hărții interactive**

Pentru a avea o imagine și mai exactă asupra locației unde se desfășoară evenimentele, s-a decis integrarea serviciilor oferite de Google, mai exact Google Maps API, cu ajutorul căruia se pot afișa pe baza coordonatelor evenimentului locația acestuia pe hartă. Astfel, se vor afișa

markeri, care indică locația evenimentelor, și prin intermediul cărora se vor putea oferi detalii despre eveniment.

- **Căutarea de evenimente în funcție de anumite filtre**

Evenimentele de interes pentru utilizator pot fi căutate prin intermediul submeniului destinat acestei funcționalități și pune la dispoziție căutarea după locație, în proximitatea utilizatorului curent, după data la care are loc evenimentul sau după categoria în care se încadrează evenimentul. Prin intermediul acestei funcționalități se realizează o filtrare a tuturor evenimentelor disponibile în datele stocate pe server.

- **Vizualizarea notificărilor**

Aplicația KweekWeek își dovedește calitatea de rețea de socializare și prin intermediul funcționalității de vizualizare a notificărilor, care sunt actualizate pe baza acțiunilor utilizatorului și a prietenilor săi. Notificările țin la curent utilizatorii cu evenimentele adăugate de prietenii lor, permit opțiunea de a accepta sau respinge cereri de prietenie din partea altor utilizatori și informează utilizatorul asupra creării evenimentelor și publicării lor.

- **Scanarea codului QR al participanților la eveniment**

În momentul achiziționării unor bilete, utilizatorul este notificat prin email de succesul tranzacției realizate și cu ajutorul aplicației poate accesa biletele generate. Acestea conțin un cod QR, care este scanat în momentul în care participantul se prezintă la eveniment. Utilitatea acestora este una deosebit de importantă, deoarece va ajuta la întocmirea de statistici cu privire la persoanele care au participat la eveniment.

- **Distribuirea evenimentului pe alte rețele sociale**

Această funcționalitate are o importanță majoră, deoarece este un beneficiu ca și alte persoane, conectate la alte rețele sociale, să afle de evenimentele care au loc, prin intermediul unui utilizator al aplicației KweekWeek.

- **Definirea categoriilor de evenimente de care este interesat utilizatorul**

Utilizatorul își poate exprima preferințele în ceea ce privește categoriile de evenimente puse la dispoziție de aplicație. Astfel, acesta poate bifa sau debifa categoriile, pentru ca pe viitor, să primească sau nu recomandări pentru acele categorii de evenimente.

- **Obținerea de bonusuri promoționale pe baza codurilor**

Funcționalitatea permite utilizatorului să folosească coduri promoționale pentru a primi reduceri pentru cumpărare de bilete la diverse evenimente, și să distribuie propriul cod, generat de sistem, prietenilor săi.

- **Actualizarea profilului utilizatorului**

Cu ajutorul acestei funcționalități, utilizatorul își poate actualiza profilul, își poate schimba poza, username-ul, emailul asociat contului de Kweekweek.

Sumarizând cele prezentate în cadrul acestui capitol, aplicația KweekWeek își dorește să fie una care să înglobeze aspecte similare unei rețele sociale capabile să asigure managementul evenimentelor, oferind totodată o interacțiune ridicată cu utilizatorul.

3 Studiu bibliografic

Acest capitol este dedicat descrierii studiului realizat înainte de a se contura produsul software. Studiul a fost realizat pentru a putea identifica care este potențialul ideii care stă la baza sistemului Kweekweek, și anume o aplicație de management a participării la evenimente, sub forma unei rețele sociale.

3.1 Rețelele sociale

Societatea ultimilor ani s-a dovedit a fi una dinamică, înconjurată de tehnologie, în care lucrurile se desfășoară cu o viteză impresionantă, informația fiind transmisă foarte rapid prin canalele de comunicație existente. Alături de multe alte inovații de business care s-au remarcat de-a lungul ultimei perioade în domeniul IT, au avut loc dezvoltări acerbe a site-urilor și aplicațiilor de tip rețea socială, care încurajează comunicarea și împărtășirea de informație între utilizatorii rețelei. Informația este consumată în diverse forme, cum ar fi: limbaj natural, informație audio sau video, poze.

Observând potențialul acestor modele de interacțiune virtuală, numeroase institute și persoane calificate în domeniul psihologiei au analizat avantajele și dezavantajele pe care le oferă aceste rețele sociale, în mod special în ce măsură ne influențează capacitatea de relaționare cu cei din jur.

În referința [1], lucrare realizată de Societatea Australiană de Psihologie, autorii prezintă un studiu realizat în anul 2010, a cărui rezultate pot fi sumarizate, așa cum reiese și din paragraful intitulat „Key Finding of Survey” al documentului, astfel:

- Rețelele sociale online sunt utilizate de 81% dintre adulții cu vârstă între 31 și 50 de ani, și într-o proporție de 56% de către persoanele cu vârstă de peste 50 de ani
- Timpul petrecut pe site-urile rețelelor sociale a fost evaluat de 70% dintre participanți că fiind mai puțin de 2 ore zilnic
- O proporție de 51% dintre persoanele intervievate au declarat că accesează și că au tendința de a se loga pe aceste site-uri de câteva ori pe zi
- În ceea ce privește timiditatea sau abilitățile de socializare ale participanților la studiu, acestea s-au încadrat în limite normale, cu puține persoane clasându-se în una dintre extreme. Mai mult, sociabilitatea persoanelor a crescut odată cu petrecerea unei perioade mai îndelungate de timp interacționând cu alți utilizatori, dovedindu-se așadar că modelul relaționării în mediul virtual este similar celui din viață reală.

Un alt studiu [2] realizat cu scopul enunțat mai sus, a fost condus de Asociația Americană de Psihologie, în anul 2012, prin care autorii au răspuns unor întrebări referitoare la rețeaua socială Facebook. Rezultatele obținute, și care sunt relevante în contextul aplicației Kweekweek, sunt următoarele:

- Persoanele care evită contactul sau interacțiunea „face-to-face” percep utilizarea rețelelor sociale ca fiind o metodă mai bună de a-și dezvoltă relațiile interpersonale.
- Motivele pentru care persoanele utilizează Facebook sunt: informația, prietenii, socializarea/comunicarea, având în vedere faptul că majoritatea utilizatorilor sunt interesați să țină legătură cu prietenii vechi sau noi, sau pentru scopuri cum ar fi: studiu, întâlniri, evenimente sociale.

În cele din urmă, se poate deduce faptul că rețele de socializare ocupă un rol important în viața omului modern, ajutându-l să interacționeze cu cei din jur, să păstreze legătura cu persoanele cunoscute, aflând noutăți și informații despre ceea ce se întâmplă în viețile lor. Rețelele de socializare reprezintă o modalitate de a comunica virtual, de a împărtăși cu restul persoanelor din cercul de prieteni experiențele noastre.

3.2 Studiul promovării unui eveniment

Aplicația Kweekweek are la bază ideea de promovare de evenimente într-un cadru social, constituit sub forma unei rețele sociale, și din acest motiv unul dintre aspectele cele mai relevante pentru sistemul de față este managementul evenimentelor.

O prima referință bibliografică întrebuințată în scopul informații cu privire la acest domeniu este referința numărul [3]. În capitolul 5 al acestei cărți, intitulat „Marketing Association Meetings, Conferences, Events, and Expositions”, subcapitolul „Promotion Methods for Association Events”, se amintesc elementele esențiale ale unei promovări de succes:

- existența unui titlu marcant, care să capteze atenția cititorului
- întrebuințarea unui limbaj cu expresii ușor de recunoscut de către publicul țintă
- realizarea unui design care să corespundă atmosferei evenimentului
- formularea clară a mesajului, astfel încât să se distingă segmentul de utilizatori căruia i se adresează
- includerea de modalități de contactare a celui care organizează evenimentul, în cazul în care utilizatorul are nelămuriri.

Aceleași concepte se regăsesc și în cea de-a doua referință bibliografică [4], care se referă la managementul evenimentelor. Capitolul 13, „Communications and sales”, menționează faptul că mesajele formulate în scopul promovării, trebuie să conducă la cel puțin una dintre următoarele situații:

- Crearea sau îmbunătățirea imaginii acelui eveniment
- Obținerea unei poziții mai bune comparativ cu competitorii
- Informarea cu privire la detaliile evenimentului
- Convertirea cererilor în vânzări efective
- Reamintirea segmentului țintă de existența evenimentului și de modalitățile de achiziționare a билетelor pentru eveniment.

Tot această referință bibliografică propune integrarea de bonusuri, reduceri, promoții în cadrul evenimentelor, deoarece stimulează o primă participare la eveniment, încurajează alte participări viitoare și generează recenzii pozitive pentru acel eveniment. Acest aspect este unul deosebit de important, deoarece aplicația Kweekweek oferă utilizatorilor posibilitatea de a beneficia de reduceri, primite de la prieteni, dar și obținute pe baza codurilor promoționale oferite de Kweekweek.

Așadar, sistemul reprezintă suma unor funcționalități atractive și interesante alături de psihologia utilizatorului care devine parte a unei rețele de socializare atât „indoor”, prin intermediul aplicației, încurajându-l să interacționeze cu ceilalți utilizatori, cât și „outdoor”, datorită noțiunii de evenimente organizate, la care participă un colectiv.

3.3 Sisteme similare

Realizarea unui proiect de calitate și care să fie competitiv pe piața aplicațiilor mobile Android a presupus un studiu al sistemelor similare existente pe piață și evaluarea atât a avantajelor, cât și a dezavantajelor acestora. Au fost identificate două aplicații Android care se încadrează, la fel că și aplicația KweekWeek, în categoria „event planners”: Nappoca și YPlan.

În cele ce urmează, se va face o analiză a celor două sisteme, pentru a determina funcționalitățile fiecăreia.

Prima aplicație menționată mai sus, intitulată Nappoca[referinta], a fost lansată la dată de 7 Februarie 2014 și este destinată vizualizării de evenimente ce au loc în orașul Cluj-Napoca. Din punctul de vedere al utilizatorului care intră în contact pentru prima dată cu aplicația, se pot afirma următoarele lucruri referitoare la această aplicație: are un nivel al interacțiunii cu utilizatorul destul de scăzut, iar funcționalitățile cele mai importante ale aplicației nu pot fi accesate decât prin conectarea aplicației la contul de Facebook al utilizatorului. Aceast ultim aspect este unul destul de relevant: o aplicație trebuie să poată fi folosită independent de existența unui cont de Facebook deținut de persoana care interacționează cu aplicația, și constituie un minus în cadrul acesteia.

În plus, vizualizarea evenimentelor nu beneficiază de filtre după care să se selecteze evenimentele pe baza unor criterii, cum ar fi locația, data, prețul, categoria din care face parte evenimentul. Accesând profilul unui eveniment, se vor vizualiza titlul evenimentului, detaliile sale, o poză reprezentativă pentru acesta, costul participării la respectivul eveniment, precum și data și locația la care are loc acesta.

Tot în cadrul paginii dedicate vizualizării detaliilor despre un eveniment se poate observa posibilitatea postării de comentarii, pe care restul utilizatorilor să le poată citi, pentru a participa și ei la aprecierea evenimentului. Aplicația pune la dispoziția utilizatorului opțiunea de adăugare de prieteni, fapt care înlesnește comunicarea între utilizatori. O altă funcționalitate accesibilă de pe pagina unui eveniment este adăugarea acestuia în calendarul telefonului, pentru că utilizatorul să fie notificat de apropierea desfășurării evenimentului. Un dezavantaj îl reprezintă imposibilitatea de a accesa harta interactivă de pe pagina destinată vizualizării detaliilor despre un eveniment, deoarece utilizatorul ar putea fi interesat de găsirea pe harta doar a locației acelui eveniment. Per ansamblu, Nappoca este o aplicație utilă, dar căreia îi lipsesc elemente mai interactive, care să ușureze navigarea utilizatorului prin aplicație.

Cea de-a doua aplicație folosită ca termen de comparație din soluțiile existente pe piață este YPlan, o aplicație disponibilă pentru orașele Londra, New York, San Francisco și Las Vegas. Ultima versiune a aplicației a fost publicată în Google Play Store la dată de 1 Mai 2014 și oferă mai multe funcționalități utilizatorului, comparativ cu Nappoca.

YPlan este o aplicație mai complexă, iar acest lucru este vizibil încă de la prima interacțiune cu aplicația: permite logarea cu un cont valid de Facebook, dar și cu un cont obișnuit, realizat cu un cont valid de email. Acest lucru este foarte util, deoarece pune la dispoziția utilizatorului 2 opțiuni, având totodată deschiderea către rețeaua socială atât de răspândită, Facebook.

O caracteristică importantă a aplicației YPlan este filtrarea evenimentelor în funcție de componentă temporală: se afișează evenimente din ziua curentă, din ziua următoare, sau evenimente viitoare. Acest lucru ajută utilizatorul în organizarea timpului liber, care este defapt și scopul aplicațiilor de acest gen.

Vizualizarea evenimentelor este însoțită, odată cu logarea utilizatorului, și de posibilitatea de adăugare a evenimentelor într-un „wishlist” sau de achiziționarea de bilete online. În cadrul paginii destinate afișării detaliilor despre un eveniment se pot regăsi următoarele: meniul populat cu un buton dedicat acțiunii de share, o galerie foto cu pozele caracteristice aceluia eveniment, titlul evenimentului, nota acordată care se situează în intervalul zero stelute – cinci stelute, categoria din care face parte evenimentul și prețul unui bilet. În plus, sunt afișate detalii despre eveniment, dar și recenzii oficiale, fapt care ajută utilizatorul în aprecierea evenimentelor. Totodată, pagina cuprinde și posibilitatea de adăugare în calendarul telefonului a evenimentului, alături de redirecționarea către Google Maps, care va prelua locația evenimentului și o va afișa individual. Achiziționarea biletelor pentru un eveniment se poate realiza din pagina destinată acestuia, și se realizează ușor, introducându-se numărul biletelor dorite, urmând ca apoi să se completeze datele cardului prin intermediul căruia se va efectua plata online.

Aplicația dispune și de căutarea de evenimente după cuvinte cheie, afișare colectivă de evenimente pe harta interactivă, dar și opțiunea de vizualizare a biletelor cumpărate.

Având în considerare cele spuse mai sus despre aplicația YPlan, se poate afirma că este o aplicație complexă, care poate fi utilizată cu succes pentru achiziționarea de bilete, informarea cu privire la activitățile sociale desfășurate în zona de interes a utilizatorului.

Ca și o concluzie vizuală a celor enunțate mai sus, în tabelul 3.1 este ilustrată o analiză comparativă a celor 2 sisteme similare, alături de sistemul de față.

Funcționalități	Nappoca	YPlan	Kweekweek
Vizualizare evenimente prin intermediul meniului navigabil, care presupune aplicarea unor filtre	✗	✓	✓
Accesarea unui eveniment și vizualizarea detaliilor despre acesta	✓	✓	✓
Vizualizarea comentariilor postate pe pagina unui eveniment și postarea de comentarii	✓	✗	✓
Follow pentru un utilizator	✓	✗	✓
Achiziționarea de bilete în cadrul evenimentelor	✗	✓	✓
Vizualizarea locației evenimentelor prin intermediul hărții interactive	✓	✓	✓
Vizualizarea locației unui eveniment individual prin intermediul hărții interactive	✗	✓	✓
Căutarea de evenimente în funcție de anumite filtre	✗	✗	✓
Vizualizarea notificărilor	✗	✗	✓
Scanarea codului QR al participanților la eveniment	✗	✗	✓
Adăugarea evenimentelor în calendarul dispozitivului mobil	✓	✓	✓

Utilizarea de bonusuri obținute din coduri	✗	✓	✓
Adăugarea evenimentelor într-o listă de preferințe viitoare	✗	✓	✓
Definirea categoriilor de care este interesat utilizatorul	✗	✗	✓

Tabelul 3.1 Analiza comparativă a sistemelor similare

Așadar, se observă că sistemul dezvoltat este unul care stăpânește multe funcționalități în plus față de sistemele analizate, și își propune să fie un sistem competitiv, care să stârnească interesul utilizatorilor prin noutățile aduse.

În plus, se observa ca studiul bibliografic a cuprins atat aspectele legate de psihologia necesitatii unei astfel de aplicatii, cat si studiul sistemelor similare existente pe piata, pentru a realiza o comparatie realista intre ceea ce a fost deja implementat de alte persoane si ceea ce urmeaza a fi implementat de catre dezvoltatorul sistemului actual.

4 Analiză si fundamentare teoretică

În acest capitol se vor analiza cerințele funcționale și non-funcționale ale sistemului, urmând ca apoi să fie identificate cazurile de utilizare pentru actorul sistemului. În ultimul subcapitol vor fi identificate tehnologiile folosite la crearea sistemului.

4.1 Scenariile aplicației

În acest subcapitol vor fi descrise 2 scenarii mai interesante ale aplicației, și etapele de realizare a acestora.

Cumpararea unui bilet

Primul scenariu poate fi descris prin următoarele etape:

- Un utilizator, denumit generic utilizatorul A, accesează site-ul www.kweekweek.com și demarează procesul, prin care completând datele cerute (locația desfășurării evenimentului, data începerii, data finalizării evenimentului, ora începerii evenimentului, tipurile de bilete disponibile, prețul lor, dar și categoria din care face parte evenimentul), va posta evenimentul și îl va face public și celorlalți utilizatori Kweekweek.
- Utilizatorii aplicației accesează aplicația Kweekweek și în cazul în care sunt prieteni cu utilizatorul A (cel care a postat evenimentul) vor fi notificați de faptul că utilizatorul A a realizat acțiunea de publicare a unui eveniment nou, iar în caz contrar, evenimentul poate fi regăsit în secțiunea „Discover”.
- Utilizatorii accesează evenimentul și vizualizează detaliile acestuia, decizând să cumpere bilete
- Utilizatorii aleg tipurile de bilete dorite, sistemul calculează suma totală de plată
- Utilizatorii încep plata online, introducând datele cardului cu care vor realiza plata
- Sistemul răspunde prin a trimite un mail în care se regăsesc atașate biletele în cadrul evenimentului ales
- La o interogare a notificărilor, utilizatorul va observa că printre notificări se regăsește și notificarea de achiziționare a unor bilete
- Utilizatorul poate accesa biletele achiziționate .

„Check in”ul participanților

Cel de-al doilea scenariu se realizează urmărind pașii de mai jos:

- Utilizatorul se prezintă la eveniment,
- Utilizatorul accesează secțiunea „My calendar” din cadrul meniului lateral, categoria „Booked” si acționează butonul „See ticket”
- Persoana responsabilă de realizarea acțiunii de „check in” a participanților la eveniment scanează codul QR al biletului.

Primul dintre scenarii este unul complex și asupra lui se mai pot interveni alte acțiuni precum: la accesarea unui eveniment, utilizatorul poate adăuga evenimentul în „wishlist”, poate

realiza acțiunea de „share” în cadrul rețelelor sociale Facebook, Twitter, sau prin Email/SMS, sau poate lăsa comentarii pe pagina evenimentului.

4.2 Cerințele aplicației

Cerințele sistemului reprezintă exprimarea serviciilor care trebuie să fie puse la dispoziție de către sistem. Acestea au o importanță majoră în construirea aplicației, deoarece întemeiază suportul de la care se pornește analiza și designul sistemului, și oferă o perspectivă a obiectivelor impuse de „stakeholder”-ii sistemului.

Cerințele funcționale exprimă funcțiile pe care trebuie să le îndeplinească sistemul, cum trebuie să răspundă anumitor intrări, care trebuie să fie reacția în anumite situații.

În schimb, cerințele non-funcționale reprezintă constrângeri ale serviciilor oferite de sistem, și definesc capacitatea sistemului.

4.2.1 Cerințele funcționale

Cerințele funcționale urmăresc descrierea funcțiilor pe care trebuie să le implementeze sistemul și a serviciilor oferite. Tabelul 4.1 expune cerințele funcționale ale aplicației, grupându-le în funcție de categoria logică în care se încadrează.

CF 1	Vizualizarea evenimentelor
CF 1.1	Vizualizarea evenimentelor prin intermediul meniului navigabil (aplicându-se diverse filtre): evenimente care se desfășoară în ziua curentă, în ziua următoare, evenimente „Trending”, „Featured”, „Redeemable”, sau publicate de prietenii utilizatorului
CF 1.2	Vizualizarea evenimentelor prin intermediul hărții interactive și meniului navigabil (aplicându-se diverse filtre): evenimente care se desfășoară în ziua curentă, în ziua următoare, evenimente „Trending”, „Featured”, „Redeemable”, sau publicate de prietenii utilizatorului
CF 2	Funcționalități specifice unui eveniment
CF 2.1	Accesarea unui eveniment și vizualizarea detaliilor despre acesta
CF 2.2	Vizualizarea comentariilor postate pe pagina unui eveniment și postarea de comentarii
CF 2.3	Achiziționarea de bilete pentru evenimente
CF 2.4	Distribuirea evenimentului pe alte rețele sociale
CF 2.5	Achiziționarea de bilete pentru evenimente
CF 3	Căutarea de evenimente în funcție de anumite filtre: locație, preț, categorie
CF 4	Vizualizarea notificărilor
CF 5	Scanarea codului QR al participanților la eveniment
CF 6	Obținerea de bonusuri promoționale pe baza codurilor Kweekweek
CF 7	Actualizarea profilului utilizatorului

Tabelul 4.1 Cerințe funcționale

4.2.2 Cerințele non-funcționale

Cerințele non-funcționale definesc proprietăți și constrângeri pe care produsul software trebuie să le îndeplinească alături de cele funcționale. Aceste cerințe au scopul de a defini performanțele sistemului, de a face o evaluare a sistemului din punct de vedere al calității, al securității, al rapidității, al comportamentului în situații extreme sau a ușurinței în întreținere. Ele pot fi mai critice decât cerințele funcționale, deoarece dacă nu sunt îndeplinite, sistemul nu va fi util scopului în care a fost dezvoltat. În tabelul 4.2 sunt expuse principalele cerințe non-funcționale ale aplicației dezvoltate.

CNF 1 - Extensibilitate	Sistemul este ușor de extins, datorită faptului că s-a respectat o arhitectură stabilă, de tip MVC, cu elementele specifice platformei mobile Android.
CNF 2 - Mentenabilitate	Aplicația este ușor de întreținut.
CNF 3 - Performanța	Aplicația este rapidă, pentru că s-au folosit tehnici de programare avansate, concepte noi din tehnologia Android pentru a optimiza comportamentul aplicației.
CNF 4 - Reutilizare	Sistemul a fost conceput după o arhitectură MVC, cu caracteristicile tehnologiei Android, astfel că este o aplicație modulară, în care fiecare componentă este ușor de refolosit.
CNF 5 - Disponibilitate	Aplicația funcționează continuu. Nu sunt prevăzute cazuri speciale, în care sistemul să eșueze.
CNF 6 - Securitate	Funcționalitatea de tip plată online este securizată, prin natura componentei integrată în sistem. Datele despre cardul utilizatorului sunt criptate, la fel și username-ul și parola.
CNF 7 - Încredere	Sistemul își păstrează performanțele de-a lungul timpului.
CNF 8 - Uzabilitate	Aplicația are o interfață prietenoasă, ușor de înțeles, folosindu-se de simboluri grafice ușor de recunoscut pentru orice tip de utilizator.

Tabelul 4.2 Cerințe non-funcționale

4.3 Cazuri de utilizare

Actorul principal al sistemului proiectat și implementat este utilizatorul obișnuit. Acesta poate să fie sau nu logat în aplicație pentru a utiliza aplicația, însă în cazul în care nu este logat, anumite operații nu vor putea fi realizate. Spre exemplu: utilizatorul care nu este logat, nu va putea să adauge un eveniment în „wishlist”, deoarece nu are asociată o astfel de listă. În principiu, dacă utilizatorul care nu este logat în aplicație încearcă să acceseze operații specifice, care sunt legate logic de existența unui utilizator, utilizatorul va fi redirectionat către pagină introductivă de înregistrare.

Astfel, analiza cazurilor de utilizare se va realiza luând în considerare un utilizator autentificat și un utilizator neautentificat.

4.3.1 Diagrama Use Case pentru utilizatorul aplicației

În diagramele de use case, reprezentate de figurile 4.1 și 4.2, sunt ilustrate acțiunile pe care le poate realiza atât utilizatorul logat în aplicație, cât și cel neautentificat. Se observă că între cele două tipuri de utilizatori există o relație de moștenire, deoarece acțiunile puse la dispoziția utilizatorului neautentificat se regăsesc în acțiunile pe care le poate realiza utilizatorul autentificat.

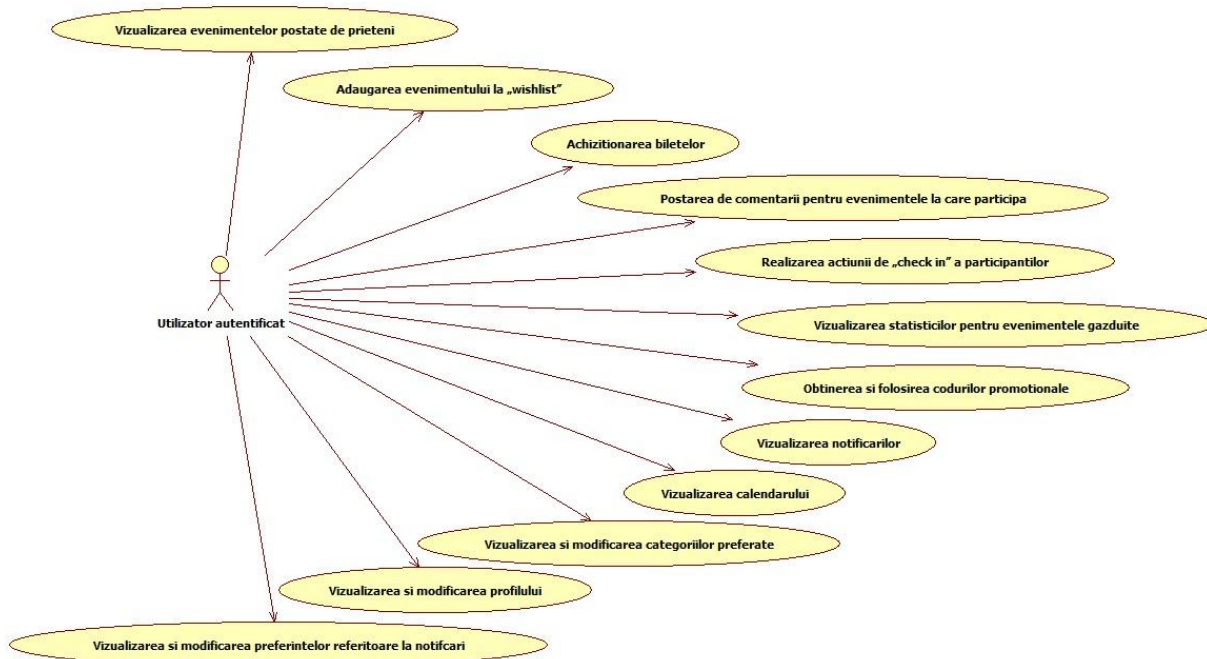


Figura 4.1 Diagrama use case pentru utilizatorul autentificat

În continuare, vor fi analizate câteva modele de use case mai interesante pentru utilizatorul obișnuit, pentru fiecare dintre acestea specificându-se precondițiile, postcondițiile, dar și scenariul de succes, alături de scenariul de eșec.

CU1. Titlu: Vizualizarea evenimentelor din ziua curentă/din ziua următoare/după categorie/„Trending”/„Featured”/„Redeemable”

Descriere: utilizatorul accesează ecranul „Discover”, dorind vizualizarea evenimentelor

Actor primar: utilizatorul neautentificat sau autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: nu există

Principiu: utilizatorul accesează aplicația, nu se loghează, și navighează la pagina „Discover” responsabilă de afișarea evenimentelor

Scenariu de succes: utilizatorului i se afișează evenimentele care corespund interogărilor aplicate, însă dacă nu există evenimente care să corespundă acestor interogări, utilizatorul este informat de acest lucru printr-un mesaj de tipul: „There are no events”.

Scenariu de eșec: utilizatorul nu dispune de conexiune la internet, caz în care sistemul afișează un „loader” care sugerează faptul că se încearcă încărcarea datelor de pe server, dar și un

mesaj exprimat astfel: „No internet connection. Please check the internet connection and try again.”

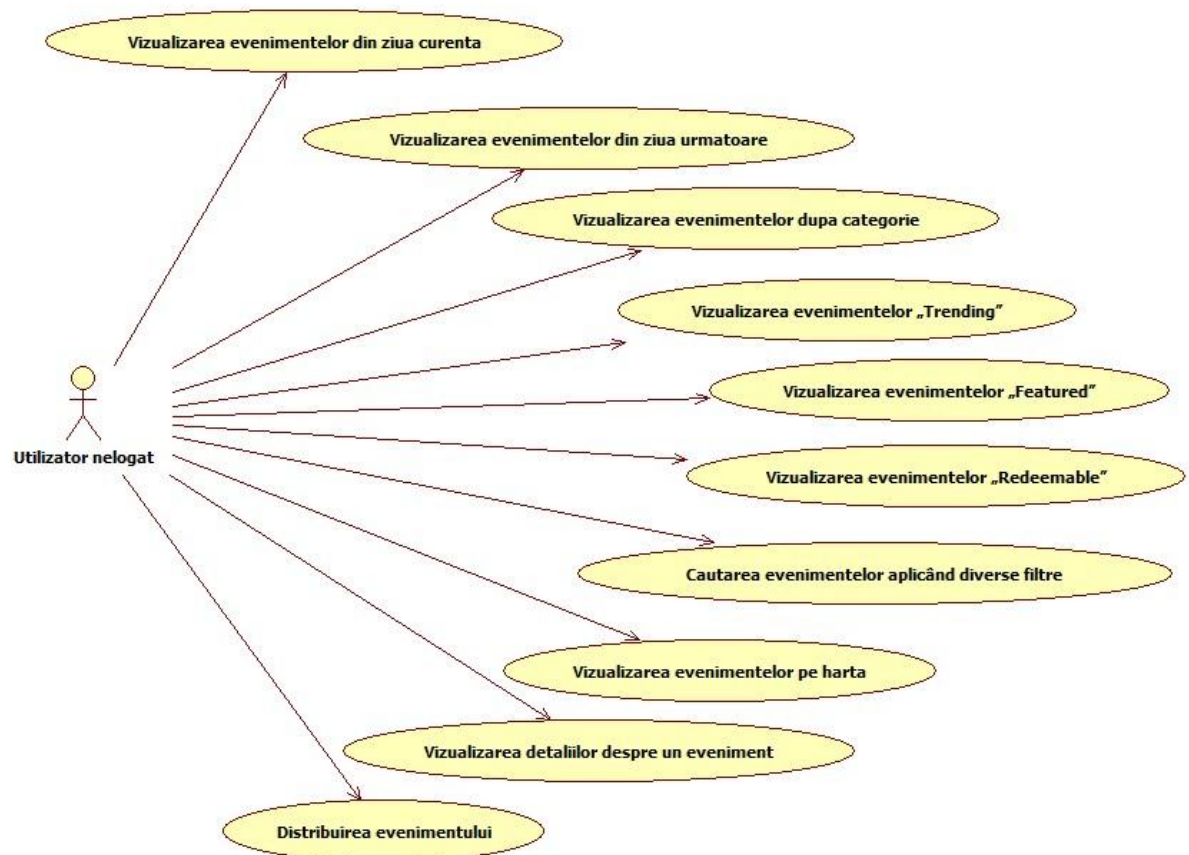


Figura 4.2 Diagrama use case pentru utilizatorul neautentificat

CU2. Titlu: Căutarea evenimentelor aplicând diverse filtre

Descriere: utilizatorul accesează simbolul grafic „lupă” din cadrul meniului principal cu intenția de a realiza acțiunea de căutare de evenimente

Actor primar: utilizatorul neautentificat sau autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: nu există

Principiu: utilizatorul accesează aplicația și navighează la pagina „Search events” responsabilă de aplicarea filtrelor

Scenariu de succes: sistemul afișează evenimentele care corespund interogărilor aplicate, respectiv filtrelor după care se dorește să se facă căutarea, în cadrul unei alte pagini „Search results”, însă dacă nu există evenimente care să corespundă acestor interogări, sistemul afișează un mesaj de tipul: „There are no events”.

Scenariu de eșec: utilizatorul nu dispune de conexiune la internet, caz în care sistemul afișează un „loader” care sugerează faptul că se încearcă încărcarea datelor de pe server, dar și un

mesaj exprimat astfel: „No internet connection. Please check the internet connection and try again.”

CU3. Titlu: Vizualizarea evenimentelor pe hartă

Descriere: utilizatorul accesează simbolul grafic „marker geografic” din cadrul meniului principal cu intenția de a vizualiza evenimente pe harta interactivă

Actor primar: utilizatorul neautentificat sau autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: nu există

Principiu: utilizatorul accesează aplicația și navighează la pagina „Map” responsabilă de afișarea evenimentelor pe harta interactivă

Scenariu de succes: sistemul afișează evenimentele conform interogărilor aplicate, respectiv filtrelor după care se dorește să se facă căutarea, fiecare dintre aceste evenimente primite ca rezultat de la server având asociat un „marker geografic” specific categoriei din care face parte

Scenariu de eșec: utilizatorul nu dispune de conexiune la internet, caz în care sistemul nu va încarca harta interactivă, afișându-se și un mesaj exprimat astfel: „No internet connection. Please check the internet connection and try again.”

CU4. Titlu: Vizualizarea detaliilor despre un eveniment

Descriere: utilizatorul accesează un eveniment vizualizat fie în cadrul secțiunii „Discover”, fie în cadrul unei liste de evenimente rezultate în urma unei căutări de evenimente, cu intenția de a vizualiza detaliile despre respectivul eveniment

Actor primar: utilizatorul neautentificat sau autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: nu există

Principiu: utilizatorul accesează aplicația și intră pe pagina „Event details” a unui eveniment

Scenariu de succes: sistemul afișează detaliile evenimentului, care includ: o scurtă descriere a evenimentului, data, ora, locația desfășurării evenimentului, categoriile din care face parte, gazda evenimentului

Scenariu de eșec: utilizatorul nu mai dispune de conexiune la internet în momentul accesării evenimentului, caz în care sistemul afișează un mesaj exprimat astfel: „No internet connection. Please check the internet connection and try again.”, iar pagina „Event details” nu este populată cu date despre eveniment.

CU5. Titlu: Distribuirea unui eveniment prin intermediul rețelelor sociale Facebook sau Twitter, respectiv prin Email sau SMS

Descriere: utilizatorul accesează un eveniment vizualizat fie în cadrul secțiunii „Discover”, fie în cadrul unei liste de evenimente rezultate în urma unei căutări de evenimente, cu intenția de a distribui evenimentul și altor persoane

Actor primar: utilizatorul neautentificat sau autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: nu există

Principiu: utilizatorul accesează aplicația și intră pe pagina „Event details” a unui eveniment, mai apoi acționează simbolul grafic „share” din meniul acestei pagini și alege dintre opțiunile puse la dispoziție de sistem, adică distribuirea pe Facebook, Twitter, Email sau SMS.

Scenariu de succes: sistemul realizează acțiunea de distribuire a evenimentului cu succes, iar postarea evenimentului este vizibilă în cadrul aplicațiilor Facebook, respectiv Twitter, Emailul/SMS-ul este trimis cu succes

Scenariu de eșec: în cazul în care pe device-ul utilizatorului nu există instalată aplicația de Facebook, Twitter, Email sau SMS, utilizatorul nu va putea realiza distribuirea evenimentelor

CU6. Titlu: Adăugarea evenimentului la „wishlist”

Descriere: utilizatorul accesează un eveniment vizualizat fie în cadrul secțiunii „Discover”, fie în cadrul unei liste de evenimente rezultate în urma unei căutări de evenimente, cu intenția de a adăuga evenimentul la „wishlist”

Actor primar: utilizatorul autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: evenimentul este adăugat în lista de „wishlist”

Principiu: există 2 moduri prin care utilizatorul poate adăuga un eveniment la „wishlist”: utilizatorul vizualizează o listă de evenimente și poate acționa asupra simbolului grafic „înmormântare” care apare în colțul dreapta-sus al fiecărui eveniment afișat sau utilizatorul accesează aplicația și intră pe pagina „Event details” a unui eveniment, iar mai apoi acționează simbolul grafic „înmormântare” din meniul acestei pagini

Scenariu de succes: sistemul realizează acțiunea de adăugare a evenimentului la „wishlist”, iar evenimentul respectiv se poate vizualiza în cadrul secțiunii „Calendar”, categoria „Wishlisted”.

Scenariu de eșec: nu există

CU7. Titlu: Achiziționarea biletelor

Descriere: utilizatorul accesează un eveniment vizualizat fie în cadrul secțiunii „Discover”, fie în cadrul unei liste de evenimente rezultate în urma unei căutări de evenimente, cu intenția de a achiziționa bilete pentru evenimentul respectiv

Actor primar: utilizatorul autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: utilizatorul a achiziționat bilete pentru evenimentul la care dorește să participe și le poate vizualiza în secțiunea „Calendar”, categoria „Booked”

Principiu: utilizatorul accesează aplicația și intră pe pagina „Event details” a evenimentului la care dorește să participe și acționează butonul „Book now”

Scenariu de succes: sistemul realizează acțiunea de achiziționare a biletelor selectate de către utilizator, iar biletele cumparate pentru evenimentul respectiv se pot vizualiza în cadrul secțiunii „Calendar”, categoria „Booked”.

Scenariu de eșec: utilizatorul nu mai dispune de conexiune la internet în momentul realizării tranzacției de plată a biletelor online, caz în care sistemul afișează un mesaj exprimat astfel: „Something went wrong, can you please try again?”.

CU8. Titlu: „Check in”-ul participantilor la eveniment

Descriere: utilizatorul accesează secțiunea „Event Management” din cadrul meniului lateral, cu intenția de a realiza „check in”-ul participantilor pentru evenimentul respectiv

Actor primar: utilizatorul autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: numărul persoanelor care s-au prezentat la eveniment este incrementat

Principiu: utilizatorul accesează aplicația și intră pe pagina „Event Management”, apasă pe butonul intitulat „Manage” corespunzător evenimentului pentru care dorește să realizeze acțiunea de „check in” a participantilor, după ce pagina cu statisticile evenimentului a fost încărcată, apasă pe butonul „Check in guests”. Utilizatorul este redirecționat către pagina de „check in”, în cadrul căreia dacă este apăsat butonul „Scan QR code”, camera nativă a dispozitivului va porni, iar utilizatorul poate scana codul QR regăsit pe biletele participantilor la eveniment.

Scenariu de succes: sistemul realizează acțiunea de scanare a biletelor participantilor la eveniment, iar numărul persoanelor care s-au prezentat la eveniment este incrementat.

Scenariu de eșec: nu există

CU8. Titlu: Postarea de comentarii pe pagina evenimentelor

Descriere: utilizatorul accesează secțiunea „Comments” a unui eveniment în cadrul căruia a achiziționat bilete sau pentru care este gazda

Actor primar: utilizatorul autentificat

Precondiții: utilizatorul dispune de conexiune la internet

Postcondiții: numărul comentariilor pentru respectivul eveniment este incrementat

Principiu: utilizatorul accesează aplicația și intră pe pagina „Event details” a unui eveniment, apoi navighează către sfârșitul paginii, apasă secțiunea „Comments”. În cazul în care nu există nici un comentariu, se va afișa un mesaj de tipul: „There are no comments yet. Be the first to post one.”. În cazul în care utilizatorul a achiziționat cel puțin un bilet în cadrul acelui eveniment sau este gazda acestuia, va putea posta comentarii.

Scenariu de succes: sistemul realizează acțiunea de postare a comentariilor pentru acel eveniment

Scenariu de eșec: utilizatorul nu mai dispune de conexiune la internet în momentul realizării cererii de postare a comentariului, caz în care sistemul afișează un mesaj exprimat astfel: „ No internet connection. Please check the internet connection and try again”.

4.4 Tehnologii utilizate

4.4.1 Platforma Android

4.4.1.1 Descrierea platformei

Android este o platformă software și un sistem de operare pentru dispozitive mobile bazat pe nucleul Linux, dezvoltat de compania Google, iar apoi de consorțiul comercial Open Handset Alliance.

Câteva dintre avantajele pe care platforma Android le detine comparativ cu celelalte platforme mobile sunt:

- **Are la baza o arhitectură bazată pe componente**

Părți dintr-o aplicație pot fi concepute și refolosite cu ușurință și în alte aplicații. Se pot înlocui și construi componentele existente cu versiuni proprii îmbunătățite. Aceasta va oferi dezvoltatorului posibilitatea exploatării creativității în spațiul mobil.

- **Constituie o platformă open-source**

Producătorii de telefoane mobile pot folosi și customiza platforma fără a plăti bani suplimentari pentru licențe. Dezvoltatorii au avantajul ca platforma este independentă și nu este blocată în nici un furnizor care poate ajunge să fie achiziționat de către alte companii.

- **Incorporează o multitudine de servicii predefinite**

Serviciul de localizare bazat pe GPS permite utilizatorului să afle tot timpul informații despre locul în care se găsește la un moment dat. Pentru stocarea datelor Android pune la dispoziția dezvoltatorului un sistem de gestiune de baze de date bazat pe SQLite. Pentru navigarea pe Web, Android are încorporat un browser.

- **Permite o gestionare automată a ciclului de viață a aplicației**

Programele sunt izolate unele de altele prin mai multe straturi de securitate, care vor oferi un nivel de stabilitate superior oricăror platforme mobile existente. Utilizatorul nu va trebui să-și mai facă griji despre programele ce rulează în sistem la un moment dat și nici nu va trebui să închidă anumite aplicații pentru a putea face loc altora. Platforma Android este optimizată pentru consumul redus de energie și de memorie.

- **Oferă o grafică și sunet de înaltă calitate**

Pentru a reda conținutul 3D dezvoltatorul are la dispoziția sa librăria Open GL. De asemenea animațiile pot fi redare cu ajutorul flash player-ului încorporat. Pentru sunete, Android oferă codec-uri pentru o varietate mare de formate de fișiere incluzând H.264 (AVC), MP3 și AAC.

- **Sustine portabilitatea rulării pe o gamă largă de hardware curente și viitoare**

Toate programele sunt scrise în Java și executate pe mașina virtuală Dalvik, existând posibilitatea portării codului pe ARM, x86 și alte arhitecturi.

De asemenea, platforma Android pune la dispoziția dezvoltatorilor o serie de unelte utile pentru dezvoltarea aplicațiilor precum un emulator, unelte pentru debugging, pentru măsurarea performanțelor aplicațiilor, și posibilitatea de integrare cu Eclipse IDE. Android este dezvoltat continuu, fiecare versiune lansată aducând îmbunătățiri la diverse componente deja existente precum și elemente și funcționalități noi care să utilizeze cât mai bine resursele fizice ale dispozitivelor.

4.4.1.2 Arhitectura platformei Android

Arhitectura după care este structurată platforma Android este ilustrată în figura 4.3. Nivelele ierarhice indică faptul că fiecare nivel este dependent de funcționalitățile oferite de nivelul inferior acestuia.

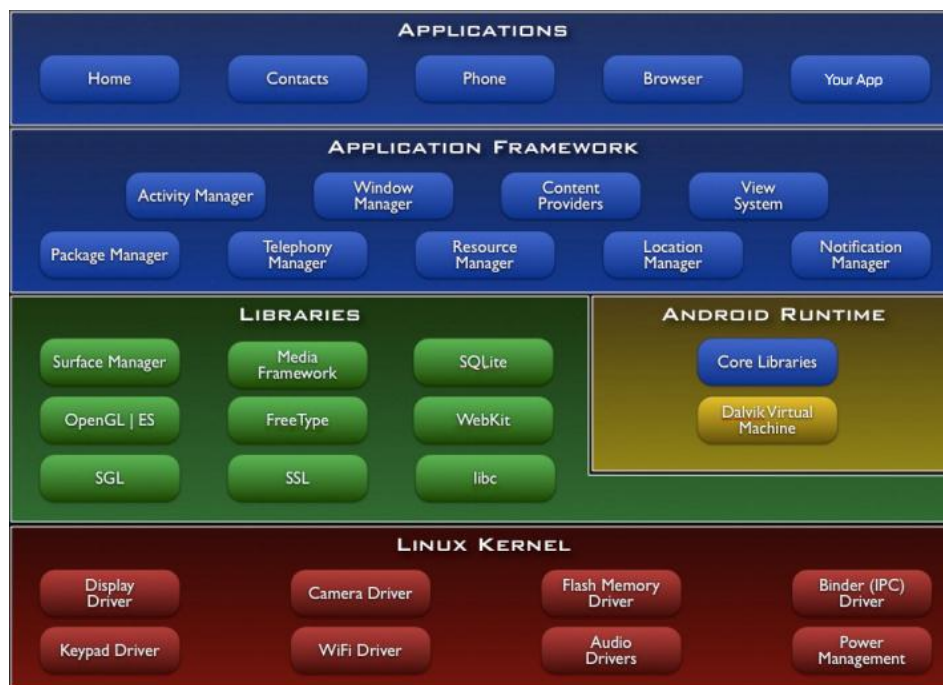


Figura 4.3 Arhitectura platformei Android^[ref]

În continuare, vor fi prezentate componentele acestei arhitecturi:

- **Kernel-ul Linux**

Android este construit pe o fundație solidă: kernel-ul Linux. Creat de Linus Torvalds în 1991, în timp ce era student la Universitatea din Helsinki, Linux poate fi găsit astăzi pe o multitudine de dispozitive, de la ceasuri de mână până la supercalculatoare. Linux oferă nivelul de abstractizare hardware pentru Android care să permită portarea acestuia pe diferite arhitecturi. Pe plan intern, Android folosește Linux pentru gestionarea memoriei, managementul de procese, crearea de rețele, precum și alte servicii predefinite. Utilizatorul nu are habar de existența acestui nivel, dezvoltatorul de aplicații nu va face apeluri directe către el, dar va trebui să ia în considerare existența lui la baza arhitecturii.

- **Librăriile native**

Următorul strat deasupra kernel-ului îl reprezintă librăriile predefinite. Aceste librării sunt scrise în totalitate în C sau C++, compilate pentru arhitectura hardware specială folosită de telefon și preinstalate de către vânzătorul telefonului.

Unele dintre cele mai importante librării implementate sunt:

- **Surface Manager** - Android utilizează un sistem de manager de ferestre similar cu cel prezent în Vista sau Compiz, dar mult simplificat. În loc să fie desenate direct pe ecran, bitmap-urile sunt administrate local și combinate pentru a reda utilizatorului interfața finală. Acest sistem permite crearea unei game largi de efecte vizuale interesante, cum ar fi ferestrele transparente sau tranziția animată între ferestre

- **Grafică 2D si 3D** - Elementele 2D și 3D pot fi combinate într-o singură interfață în Android. Bibliotecile vor folosi accelerația hardware dacă aceasta este prezentă pe dispozitiv sau un renderer software în caz contrar
- **Browser** - Pentru afișarea rapidă a conținutului HTML, Android folosește WebKit. Aceasta folosește același motor folosit în browser-ul Google Chrome, browser-ul Apple Safari, Apple iPhone, și platforma Nokia S60
- **Baza de date SQL** - Android include o versiune ușoară de sistem de gestiune a bazelor de date: SQLite, aceeași bază de date utilizată în Firefox și iPhone. Dezvoltatorul poate folosi acest sistem pentru stocarea persistentă a datelor

Aceste biblioteci nu sunt aplicații de sine stătătoare, ele există doar pentru a oferi suport pentru nivelele superioare.

- **Android Runtime**

Android include un set de biblioteci care susțin o mare parte din funcționalitatea pusă la dispoziție de limbajul de programare Java. Prin implementarea lor a rezultat crearea mașinii virtuale Java intitulată Dalvik, care a fost special creată pentru Android de către Dan Bornstein și echipa sa de la Google. Fiecare aplicație Android rulează într-un proces propriu și are o instanță separată de mașină virtuală Dalvik. Dalvik a fost astfel scrisă încât să permită rularea mai multor instanțe de mașină virtuală pe același dispozitiv hardware.

Diferențele dintre Dalvik și mașina virtuală Java standard:

- Dalvik VM rulează pe fișiere .dex care sunt obținute prin convertirea fișierelor .class și .jar în momentul compilării. Fișierele .dex sunt mai compacte și mai eficiente decât fișierele .class pentru a optimiza consumul de memorie și de energie.
- Bibliotecile de bază Java care vin cu Android sunt diferite atât față de bibliotecile din Java Standard Edition (Java SE) și de cele din Java Mobile Edition (Java ME). Există totuși un subset comun de API-uri.

- **Application Framework**

Acest nivel se găsește peste nivelul bibliotecilor native și a runtime-ului Android. Punând la dispoziție o platformă de dezvoltare deschisă, Android oferă dezvoltatorilor posibilitatea de a crea aplicații extrem de variate și inovative și acces la hardware-ul telefonului, la informații de localizare, posibilitatea rulării de aplicații în fundal, setarea de alarme, notificări și multe altele. Dezvoltatorii au acces complet la același API care este folosit și de aplicațiile native Android. Arhitectura este astfel concepută încât să ușureze reutilizarea componentelor: orice aplicație poate să-și publice capabilitățile și orice aplicație poate apoi să folosească aceste capabilități, sub anumite constrângeri făcute de framework.

Sub toate aplicațiile se găsește un set de servicii și sisteme, incluzând:

- un set bogat de componente UI ce pot fi folosite de utilizator pentru a crea o aplicație. Acest set include grid-uri, listview-uri, textbox-uri, butoane și chiar și un browser embeddable
- Activity Manager - controlează ciclul de viață al aplicațiilor și modul de comunicare dintre acestea
- Resource Manager - controlează accesul la resursele non-cod, cum ar fi: string-uri,

layout-uri, imagini

- Content providers - aceste obiecte incapsulează datele si informațiile care trebuie împarțite între aplicații
- Notification Manager - permite notificarea utilizatorului sub diferite forme.

4.4.1.3 Componentele unei aplicatii Android

Componentele sunt blocuri esențiale ale aplicației Android. Fiecare componentă reprezintă un punct distinct prin care sistemul poate accesa aplicația, și joacă un rol specific ajutând astfel la definirea comportamentului aplicației. Există șase tipuri diferite de componente. Fiecare tip are un scop specific și un ciclu de viață care definește cum este creată și distrusă respectiva componentă.

Activități

Activitățile reprezintă partea de prezentare într-o aplicație. Fiecare ecran din aplicație va fi o extensie a clasei Activity. O aplicație poate fi alcătuită dintr-o singură activitate sau poate să conțină mai multe în funcție de design-ul conceput de dezvoltator. De obicei, una dintre activități este marcată ca fiind prima activitate care să fie prezentată utilizatorului.

Trecerea de la o activitate la alta se realizează prin apelul explicit în cadrul primei activități a rutinei care o pornește pe cea de a doua. Cele mai multe activități sunt proiectate pentru a ocupa întregul ecran, dar se pot crea activități care sunt semitransparente, plutitoare, sau dialog-uri. Fiecare activitate este alcătuită dintr-o structură de vederi.

Vederile reprezintă elementele de interfață de bază în Android. De exemplu, o vedere poate prezenta o imagine și să reacționeze când utilizatorul apasă pe ea. Pentru a crea o activitate, dezvoltatorul trebuie să extindă clasa de bază Activity și să constituască în interiorul ei interfața și comportamentul dorit. Componentele unei aplicații au un ciclu de viață : un început când Android le instanțiază pentru a răspunde la anumite intenții și un sfârșit atunci când instanțele sunt distruse.

Starea fiecărei activități este determinată de poziția sa cu privire la stiva de activități, o colecție FIFO (first in first out) ce conține toate activitățile care rulează la un moment dat în sistem. Când o nouă activitate este începută, aceasta este mutată în capul stivei. În cazul în care utilizatorul navighează înapoi folosind butonul dedicat sau când activitatea este închisă, activitatea imediat următoare este împinsă pe stivă și devine activă. Această stivă este folosită și de manager-ul de memorie pentru determinarea priorității aplicației și eliberarea resurselor de memorie.

O activitate are, în esență, trei stări:

- Este activă sau se execută atunci când este în prim planul ecranului (se situează în capul stivei de activități). Aceasta este activitatea cu care interacționează utilizatorul la un moment dat.
- Este întreruptă în cazul în care și-a pierdut focus-ul, dar este încă vizibilă pentru utilizator. O altă activitate se află peste activitatea întreruptă dar o parte a activității întrerupte este încă accesibilă utilizatorului. O activitate întreruptă este menținută în viață (își menține starea în întregime, informațiile despre membrii și rămâne atașat la managerul de ferestre), dar poate fi oprită de sistem în situații de utilizare excesivă a memoriei.

- Este oprită în cazul în care este complet acoperită de o altă activitate. Informațiile despre stare sunt menținute, dar activitatea respectivă nu mai este vizibilă pentru utilizator și va fi distrusă când va fi nevoie să se elibereze memoria.

Ciclul de viața al unei activități este ilustrat în figura 4.4.

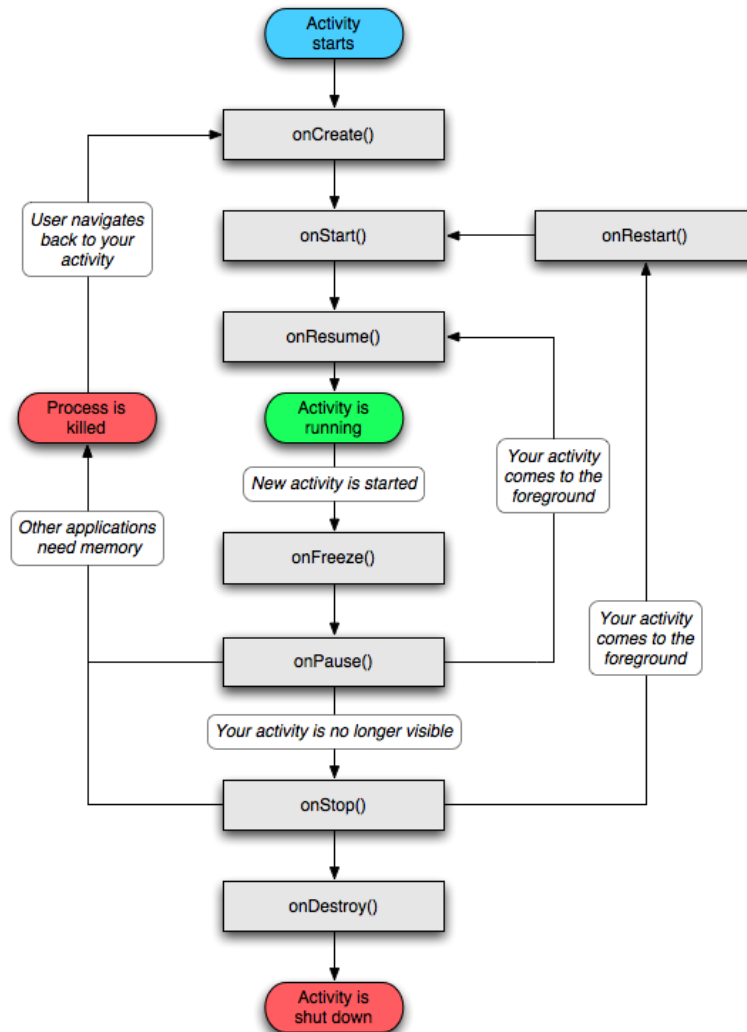


Figura 4.4

Intenții

Intent-urile reprezintă un mecanism de mesaje ce permite declararea intenției ca o acțiune de efectuat, de obicei, cu un anumit set de date. Intențiile suportă interacțiunea dintre două componente diverse aflate în aceeași aplicație sau în aplicații diferite. Ele sunt responsabile cu transformarea unei colecții de componente independente într-un singur sistem interconectat.

Una dintre cele mai comune utilizări ale intențiilor îl reprezintă pornirea unei activități, fie în mod explicit (prin specificarea clasei activității care se dorește a fi încărcată), fie implicit (prin cererea ca o acțiune să fie efectuată pe un set de date). O aplicație poate înregistra un broadcast

receiver care să asculte și să reacționeze la aceste Intent-uri. Android folosește intent-uri broadcast pentru a anunța evenimente de sistem, cum ar fi schimbarea stării conexiunii de internet sau nivelul bateriei telefonului.

Aplicațiile native Android , cum ar fi SMS manager-ul sau manager-ul de telefonie, înregistrează componente care ascultă aceste intent-uri broadcast și reacționează corespunzător prin alertarea utilizatorului cu privire la schimbările din sistem. Pentru a începe în mod explicit o activitate trebuie creat un nou intent specificând contextul curent al aplicației, precum și clasa de activitate ce trebuie lansată. Această intenție trebuie trimisă ca parametru metodei `startActivity`, așa cum se arată în următorul fragment de cod:

```
Intent intent = new Intent(MyActivity.this, MyOtherActivity.class);
startActivity(intent);
```

Fragmente

Servicii

Un serviciu este o componentă care rulează în background pentru efectuarea de operații de durată sau pentru a permite accesul la distanță al proceselor. Un serviciu nu dispune de o interfață utilizator. De exemplu, un serviciu poate efectua un proces în background, în timp ce utilizatorul se află în cadrul altei aplicații. O altă componentă, precum o activitate, poate începe serviciul, îl poate lăsa să ruleze sau îl poate îngloba pentru a interacționa cu acesta. Un serviciu este implementat ca o subclasă a clasei `Service`.

Receptorii de broadcast

Un receptor de broadcast este o componenta care raspunde anunturilor de broadcast ale sistemului. Multe broadcasturi isi au originea in cadrul sistemului, de exemplu, un anunt broadcast legat de inchiderea ecranului, terminarea bateriei, sau efectuarea unei fotografii. Aplicatiile pot initia si ele broadcast-uri, precum anuntarea altei aplicatii de faptul ca anumite date au fost descarcate si se afla pe dispozitiv disponibile spre utilizare. Cu toate ca receptorii de broadcast nu afiseaza o interfata utilizator, pot crea o bara de notificare a status-ului pentru a instiinta utilizatorul de momentul aparitiei unui eveniment de broadcast. Un receptor de broadcast este implementat ca si subclasa a clasei `BroadcastReceiver` si fiecare broadcast este livrat ca un obiect `Intent`.

Fișierul manifest Android

Fiecare proiect Android include un fișier manifest, `AndroidManifest.xml`, stocat în directorul rădăcină al proiectului. În acest fișier sunt descrise componentele folosite în aplicație și alte informații referitoare la permisiuni sau librării ce trebuie legate de aplicația curentă. Se poate defini versiunea de Android folosită de aplicație, precum și o temă preferată pentru afișarea interfeței.

Structura acestui fișier impune prezența unui tag rădăcină `<manifest>` urmat de tag-ul `<application>` în interiorul căruia vor fi descrise componentele care alcătuiesc aplicația. Pentru definirea de permisiuni se va folosi tag-ul `<uses-permission>`, urmat de numirea permisiunii dorite.

Deoarece sistemul rulează fiecare aplicație într-un proces separat cu fișiere de permisiuni care restricționează accesul la alte aplicații, aplicația nu poate acționa direct o componentă din cadrul altei aplicații. Sistemul Android, în schimb, poate. Astfel, pentru a activa o componentă a altei aplicații

trebuie trimis un mesaj sistemului care să specifice intenția (*Intent*) de a începe o anumită componentă. Sistemul activează apoi componenta respectivă.

4.4.2 Google Maps API

Google Maps[ref] este o tehnologie si aplicatie desktop si mobila care ofera serviciul de „web mapping”. Acest concept se refera la procesul de folosire a hartilor oferite prin sistemele de informatie geografica. Google Maps ofera imagini satelitare, harti stradale, vizualizare stradala, precum si functii de planificare a rutelor alegand modalitati de transport precum: mersul, masina, bicicleta, transport public.

Compania Google a lansat Google Maps API in anul 2005 pentru a permite dezvoltatorilor sa integreze Google Maps in cadrul site-urilor web. Este un serviciu public, gratuit, fara reclame, insa termenii si conditiile Google specifica faptul ca isi rezerva dreptul ca pe viitor sa poata afisa reclame.

Google Maps API este gratuit pentru uz comercial, in conditiile in care aplicatia care se foloseste de aceste servicii este publica, nu taxeaza utilizarea sa si nu depaseste pragul de 250 000 de accesari ale hartilor pe zi. Aplicatiile care nu se incadreaza in aceste aspecte pot achizitiona pachetul Google Maps API pentru afaceri.

Maps Engine API este un serviciu API RESTful, toate „request”-urile catre API sunt „request”-uri HTTP, astfel incat orice limbaj de programare cu o librerie HTTP poate fi folosita pentru a interoga si modifica date in API.

In plus, resursele sunt reprezentate in formatul JSON, care va fi descris in subcapitolul urmator.

4.4.3 JSON

Acronimul JSON provine de la JavaScript Object Notation si desemneaza un format standard public, care utilizeaza text usor de interpretat de catre oameni pentru a transmite obiecte alcatuite din perechi de tipul cheie-valoare. Acest format este adesea preferat in detrimentul formatului XML.

Tipurile definite in acest format sunt urmatoarele:

- Number - un numar zecimal, cu semn, care poate contine parte fractionara si se poate folosi de notatia E, pentru notarea exponentilor. JSON nu permite utilizarea non-numerelor precum NaN si nici nu face distinctie intre un numar real si un numar intreg
- String - o secventa de 0 sau mai mult caractere Unicode. In formatul JSON, stringurile sunt delimitate prin ghilimele, iar apostroful trebuie insotit de caracterul backslash, pentru a evita confuziile care ar putea interveni la nivelul compilarii
- Boolean - valorile „True” sau „False”
- Array - o lista ordonata de 0 sau mai multe valori, fiecare dintre acestea putand fi de orice tip. Array-urile folosesc notatia cu paranteze acolade, iar elementele se separa prin virgula
- Object – un array asociativ de tip (cheie, valoare). Obiectele sunt delimitate de paranteza acolada si folosesc virgula pentru a delimita fiecare pereche. Cheia si

valoarea sunt delimitate de caracterul „:”. Toate cheile trebuie sa fie de tipul String si trebuie sa fie distincte intre ele in cadrul aceluiasi obiect.

- Null - nu exista o valoare asociata.

Iata un exemplu de informatie structurata in formatul JSON:

```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to create markup languages such as
DocBook.",
            "GlossSeeAlso": ["GML", "XML"]
          },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

Anterior s-a precizat faptul ca acest format este din ce in ce mai des utilizat in locul formatului XML. Iata cateva dintre avantajele formatului JSON in comparatie cu XML:

- JSON are o gramatica mai simpla decat XML si se mapeaza mai bine pe structurile de date folosite in limbajele de programare moderne
- JSON este mult mai „human readable”, dar este si mai usor de scris
- JSON este mult mai usor de procesat de catre calculator
- Necesita software mult mai putin specializat, fiind o notatie mult mai simpla.

4.4.4 GSON

GSON este o biblioteca Java „open source” utilizata pentru a serializa si deserializa obiecte Java, sau asa numitele POJO („Plain Old Java Object”) in si din formatul JSON, prezentat in subcapitolul anterior. Libraria a fost initial dezvoltata de compania Google pentru utilizarea in cadrul companiei, insa apoi a fost facuta publica in anul 2008.

Cateva caracteristici ale acestei librarii ar fi urmatoarele:

- Poate manipula colectii, tipuri generice si clasele „nested”
- La deserializare, parcurge arborele pentru obiectul de deserializat. Implicatia acestui lucru este ca campurile care sunt prezente in formatul JSON, dar nu si in obiectul in care se doreste deserializarea, vor fi ignorate.
- Utilizatorul poate crea serializatoare si deserializatoare proprii, pentru a controla procesul si mai bine, si chiar si pentru a serializa/deserializa instante ale claselor pentru care codul sursa nu este accesibil.
- Se pot specifica cerinte referitoare la: „pretty printing”, cum sa se manipuleze campurile „null” dintr-un obiect, care campuri sa fie prezente in serializare/deserializare.

Asadar, aceasta librarie este una foarte utila, deoarece face conversia intre obiecte Java si formatul JSON si invers.

4.4.5 Stripe **referinte bibliografice !!!1**

Stripe este un sistem nou, aparut in 2010, care permite plati online securizate. Avantajele acestui sistem sunt faptul ca asigura un flux al platilor personalizabil, care functioneaza impecabil atat in zona desktop, cat si in zona mobile, permite plati online de pretutindeni in lume, se pot vizualiza grafice referitoare la tranzactiile realizate, iar interfata pusa la dispozitia dezvoltatorilor software permite scalabilitate: sistemul ofera suport pentru foarte multe tipuri de cerinte. Este un sistem securizat, protejand persoanele implicate in tranzactii de posibile plati frauduloase si monitorizeaza tranzactiile suspecte.

Stripe API este organizat pe modelul REST, este conceput astfel incat sa aiba adrese URL orientate pe resursa, si cat mai previzibile, si foloseste coduri de eroare HTTP pentru a indica erori. Se utilizeaza caracteristici „built-in” ale protocolului HTTP, cum ar fi autentificarea HTTP sau metodele HTTP. Stripe utilizeaza si notiunea de „Cross-origin resource sharing” ceea ce inseamna ca resursele pot fi interogate din cadrul unui alt domeniu, dinafara celui din care provin resursele, ceea ce permite interactiunea cu API-ul, pe baza unor chei private care nu trebuie expuse. Aceste chei sunt disponibile atat in modul „live”, cat si in modul de testare. Modul de testare asigura faptul ca sumele de bani implicate in tranzactii nu vor fi retrase din conturile bancare. Mai mult, unicitatea cheilor generate asigura un permanent control al platilor si o securitate crescuta. Formatul raspunsului returnat de catre apelurile la API este JSON, inclusiv in cazul erorilor.

4.4.6 OrmLite

OrmLite[] este un framework software „open source” care pune la dispozitie maparea relationala obiectuala „lightweight” intre clasele Java si bazele de date SQL. OrmLite are urmatoarele caracteristici:

- Permite configurarea claselor Java care trebuie sa fie persistate prin intermediul anotarilor
- Pune la dispozitie clase DAO („Database Access Object”) foarte puternice
- Are in componenta sa un „Query Builder” prin intermediul caruia se pot realiza atat interogari simple, cat si complexe
- Manipuleaza „statement”-uri SQL compilare pentru actiuni de interogare repetitive

- Suporta MySQL, Postgres, Microsoft SQL Server, H2, Derby, HSQLDB si Sqlite
- Suporta obiecte „foreign” cu campul definit de obiectul respectiv, dar cu un id stocat in baza de date
- Oferă suport standard pentru tranzactii
- Autogeneraza SQL pentru a crea si sterge tabele din baza de date
- Integreaza apeluri native catre Android SQLite API.

4.5 Pattern-uri arhitecturale

Pattern-ul arhitectural care se mapeaza cel mai bine pe arhitectura Android si componentele sale este cel bazat pe „layere”, datorita faptului ca permite organizarea logica si ierarhica a componentelor care indeplinesc actiuni specifice.

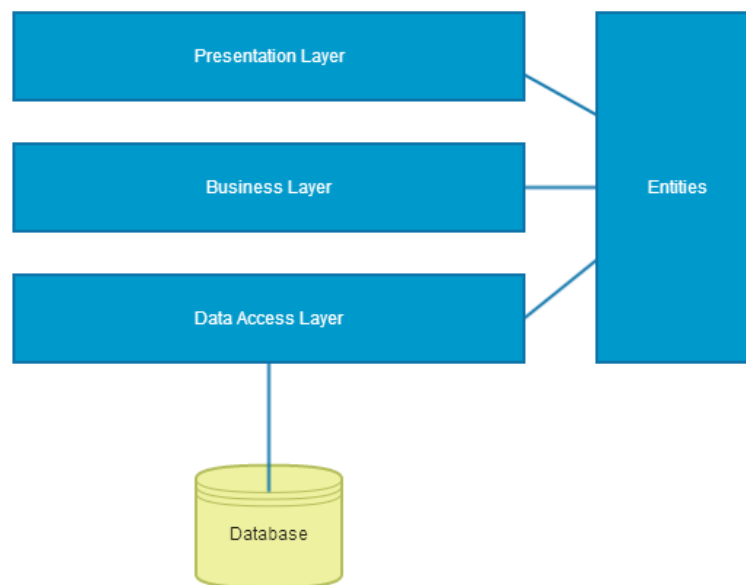


Figura 4.5 Arhitectura pe layere

Acest pattern arhitectural isi propune sa impuna o delimitare clara asupra componentelor, prin implementarea de pachete si entitati care sa comunice intre ele, insa doar intr-o anumita ordine. Prin acest lucru se intelege faptul ca un nivelele comunica intre ele, insa nici un nivel nu acceseaza informatii care provin de la mai mult de un nivel inferior lui.

Figura 4.5 prezinta maniera in care se realizeaza comunicarea si interactiunea intr-un sistem care are la baza acest pattern. Utilizatorul interactioneaza strict cu componentele interfeței utilizator, specifice tehnologiei in care se realizeaza implementarea. Acestea din urma comunica cu nivelul logic de business, cel care implementeaza logica sistemului, si care se poate folosi si de serviciile puse la dispozitie de „third parties”. Ultimul layer realizeaza interactiunea sistemului cu baza de date, pentru a persista date local.

Astfel, analiza sistemului si fundamentarea teoretica au acoperit analiza cerintelor functionale si non-functionale, realizarea unor cazuri de utilizare care sa poata fi utilizate cu

usurinta in cadrul iteratiilor proiectului, dar si analiza tehnologiilor disponibile si modul in care acestea se mapeaza pe obiectivele proiectului.

5 Proiectare de detaliu și implementare

Capitolul de fata prezinta o detaliere a sistemului, alaturi de solutiile de implementare, arhitectura, componentele implicate.

5.1 Arhitectura sistemului

Sistemul actual este implementat utilizand o arhitectura client-server, asa cum se poate observa si din figura 5.1, care ilustreaza arhitectura conceptuala a sistemului. Aceasta este menita sa ofere o viziune de ansamblu asupra modului in care realizeaza comunicatia intre client si servere, deci intre client si furnizorii de servicii, ilustrand protocolul folosit pentru comunicatie, dar si formatul datelor care se partajeaza intre cele doua componente.

Arhitectura client-server presupune existenta unei aplicatii distribuite care partajeaza sarcinile intre furnizorii de servicii numiti servere si cei care solicita servicii, numiti clienti. Acest model este alegerea cea mai populara printre dezvoltatorii de aplicatii software, astfel ca s-a impus ca fiind unul dintre cele mai fiabile modele pentru implementarea aplicatiilor distribuite.

De cele mai multe ori, clientii si serverele comunica prin intermediul Internetului, la fel cum se intampla si in cazul de fata, datorita faptului ca acesta ne ofera beneficiul partajarii de informatie intr-un mod rapid si sigur.

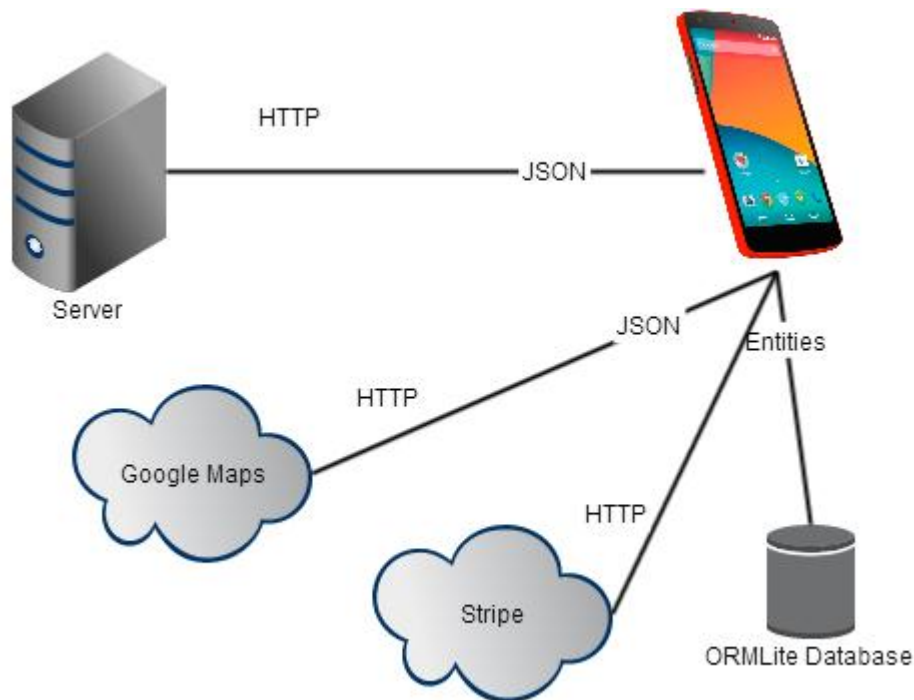


Figura 5.1 Arhitectura sistemului

Componenta server isi partajeaza resursele cu clientii, insa un client nu isi partajeaza nici o resursa, ci doar solicita partajarea de resurse din partea serverului. Astfel, clientii initiaza sesiuni de comunicare cu serverul si asteapta raspunsul acestuia.

Partajarea resurselor semnifica faptul ca serverele raspund cu informatie utila clientului, iar clientul primeste informatia pe care o cere. In plus, serverul nu expune informatii precum specificatii software sau hardware ale masinilor pe care ruleaza, ci doar o interfata transparenta pentru clienti. Acest lucru se obtine in cadrul sistemului actual cu ajutorul unui API (Application Programming Interface), prin care serverul si clientul schimba informatie utila in formatul JSON.

Din faptul ca serverul pune la dispozitia clientului informatie utila deducem faptul ca serverele au o putere mare de calcul, fac procesari intense si au resurse hardware puternice pentru a realiza aceste operatiuni. Serverele ofera clientilor raspunsuri pentru cererile lor, iar acestia afiseaza informatia. Se observa un beneficiu major al arhitecturii client-server, care imparte responsabilitatea in mod echilibrat: atribuie procesari mai complexe serverului, iar procesarile mai „light” sunt realizate de partea clientului, care dispune de resurse hardware si software mai limitate.

5.1.1 Atributiile componentelor sistemului

In cele ce urmeaza se vor prezenta rolurile componentelor server si client in cadrul sistemului actual, pentru a clarifica cum a fost conceput acesta.

Asa cum s-a precizat anterior, serverul este o componenta cu o putere de calcul mai mare si suporta mai multe operatii pe secunda, acest lucru permitand o procesare mai buna. Coomponenta client dispune de resurse hardware mult mai modeste si realizeaza operatii simple.

5.1.1.1 Aplicatia server

Aplicatia server este responsabila de realizarea unor operatii complexe, precum:

- Gestionarea utilizatorilor
- Gestionarea relatiilor dintre utilizatori (de tipul „follow”)
- Gestionarea preferintelor utilizatorilor
- Realizarea de cautari rapide
- Gestionarea informatiilor despre evenimente
- Trimiterea raspunsurilor la clienti conform cererilor initiale de acestia

Toate acestea se rezuma la gestionarea eficienta a bazelor de date, a resurselor hardware, care sunt limitate, asigurarea unui „load balancing” intre masinile server.

5.1.1.2 Aplicatia client

Aplicatia client are atributii mai simple si implementarea clientului trebuie sa fie la fel de eficienta ca si in cazul aplicatiei server, deoarece resursele dispozitivului mobil sunt limitate: bateria cu durata de viata mai scurta, conexiunea la Internet este, de multe ori, costisitoare.

Astfel, aplicatia client realizeaza urmatoarele operatii:

- Parsarea informatiilor primite de la server
- Afisarea informatiilor in urma parsarilor
- Persistarea locala a informatiilor despre utilizator

Se observa faptul ca atributiile sunt repartizate uniform si pe masura capacitatilor fiecarei componente. In plus, verificarea rezultatelor este mult mai usor de realizat, atat pe partea de server, cat si pe partea de client, deoarece se pot urmari rezultatele provenite de la fiecare componenta.

5.2 Arhitectura aplicatiei client

În figura 5.2 este reprezentată arhitectura aplicației mobile. Aceasta este constituită din câteva componente, structurate pe 5 layer: nivelul de prezentare, nivelul de comunicare și servicii, nivelul de conversie, nivelul de modele și entități și nivelul de management al bazei de date. Se observă că aceste niveluri corespund arhitecturii generale pe layeruri descrisă anterior, la aceasta adăugându-se nivelul de comunicare, necesar datorită nevoii de interacțiune a serverelor furnizoare de servicii cu clientul Android, și nivelul de conversie.

Aceste 4 niveluri se regăsesc mapate în pachetele implementării propriu-zise și au următoarele roluri:

- Nivelul de comunicare: asigură realizarea conexiunii la servere, prin intermediul protocolului HTTP, stabilind timpi de time-out, modalități de tratare a succesului sau erorii la finalizarea request-urilor.
- Nivelul de conversie: convertește informația utilă provenită din răspunsurile serverelor în modele de date
- Nivelul de modele și entități: reprezintă obiecte POJO, modelări ale vieții reale și obiecte mapate pe baza de date prin intermediul anotărilor
- Nivelul de management al bazei de date: permite persistarea datelor și realizarea de operații de tip CRUD asupra datelor din baza de date

În continuare vor fi detaliate fiecare dintre aceste niveluri și modul în care acestea interacționează.

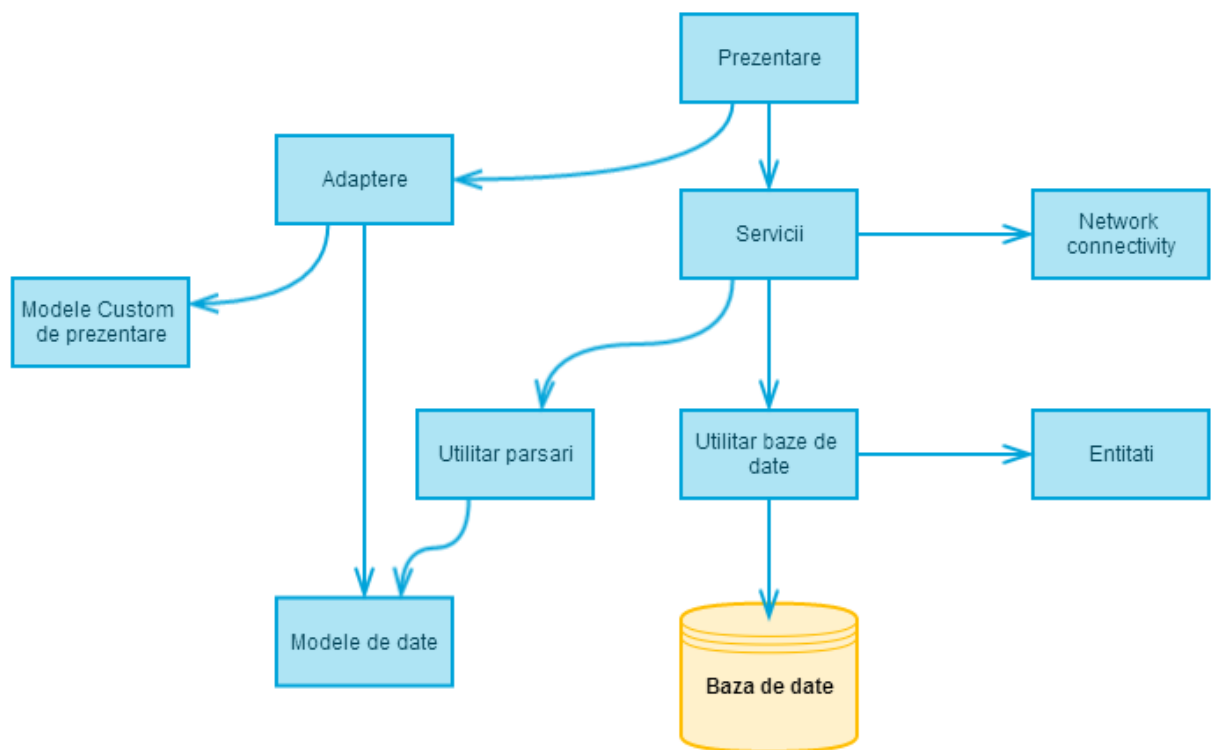


Figura 5.2 Arhitectura aplicației mobile

5.2.1 Nivelul de comunicare si servicii

Toate functionalitatile majore ale aplicatiei se bazeaza pe existenta unei conexiuni stabile la Internet. Comunicatia intre client si server se realizeaza pe baza unor cereri exprimate de client, prin intermediul protocolului HTTP, urmate de un raspuns al serverului. Acest lucru este efectuat la nivelul implementarii prin existenta unor obiecte create la momentul intemeierii requestului si care primesc ca si parametri in constructor urmatoarele informatii:

- Metoda HTTP invocata
- URL-ul la care se face requestul
- Parametrii requestului
- Un Listener personalizat, care specifica ce actiuni trebuie sa se realizeze in caz de succes sau eroare
- Tipul obiectului returnat
- Clasa corespondenta pentru tipul de informatie returnata.

Mai apoi, exista implementata o coada FIFO de requesturi in care se adauga requesturi pentru a fi trimise catre server. Astfel, ca la fiecare creare de request este necesara adaugarea acestuia in coada pentru a fi inregistrata si tratata. Requesturile se iau in ordinea sosirii in coada FIFO si se executa.

Un aspect important de mentionat este acela ca requesturile se executa intr-un thread separat de main thread (Ui thread) pentru a nu bloca interfata utilizator. In timpul executiilor requesturilor, sunt afisate mesaje de asteptare sau animatii care sa simbolizeze progresul executiei.

5.2.2 Nivelul de conversie

Comunicatia intre server si client se realizeaza prin intermediul protocolului HTTP, componentele interschimband informatie in formatul JSON. Aceste date trebuie sa se translateze in obiecte JAVA, necesare procesarilor ulterioare ale sistemului. Acest nivel tocmai cu acest aspect se ocupa: conversia datelor din format JSON in obiecte POJO, adica de o deserializare. Pentru aceasta implementare a fost necesara utilizarea bibliotecii GSON, care a fost amintita si n cadrul capitolului 4, Analiza si fundamentare teoretica.

5.2.3 Nivelul de modele si entitati

Modelele reprezinta obiecte POJO, unele dintre ele fiind chiar modelari ale lumii reale. Spre exemplu, un obiect de tip Event are ca si attribute caracteristici care se regasesc transpuse in viata reala: descrierea evenimentului, numarul de bilete vandute, nota, numarul de persoane participante, numar de comentarii, sumar, o lista de imagini asociate evenimentului si multe altele. Aceste modele doresc sa surprinda cat mai mult dintre proprietatile reale ale unor concepte si sa le translateze cu succes in implementare.

Entitatile sunt similare modelelor, insa vor fi supuse persistarii in baza de date. Obiectele Java vor fi mapate unor intrari din baza de date, acest lucru realizandu-se prin intermediul anotarilor din framework-ul OrmLite.

5.2.4 Nivelul de management al bazei de date

În cadrul acestui nivel intra așa numitele DAO (Data Access Object), care sunt responsabile de implementarea unor metode CRUD personalizate pe tipul fiecărei entități. Clasele suprascriu metode deja existente în cadrul framework-ului OrmLite și adaugă în plus metode specifice, care îndeplinesc anumite operații asupra bazei de date: actualizări, ștergeri asupra tabelor în funcție de interacțiunea dintre obiectele aparținând nivelului de entități.

5.2.5 Nivelul de prezentare

Nivelul de prezentare este poate cel mai complex nivel deoarece modelează ecranele aplicației, asigură popularea acestora cu informație necesară din obiectele Java, obținute în urma parșării descrise mai devreme.

Un aspect important este acela că nivelul de prezentare include foarte multe aspecte legate de validarea datelor înainte ca requesturile să fie transmise către server. Sistemul nu va permite trimiterea unui request cu niște parametrii HTTP necompletați, în condițiile în care aceștia sunt necesari, doar pentru ca mai apoi răspunsul să fie că acele câmpuri trebuie completate. Sistemul tratează aceste cazuri înainte ca operații mai costisitoare, precum realizarea de trafic, consumarea de baterie a dispozitivului, să fie inițiate. Acest lucru înseamnă că validările vor avea loc la nivelul prezentării, astfel încât datelor să nu ajungă prin request până la componenta de server a sistemului.

Nivelul de prezentare cuprinde:

- Activități
- Fragmente
- Adaptare
- Resurse XML pentru layouturi cu id-uri asociate elementelor de interes
- Resurse drawable: definiții de forme personalizate
- Resurse animație: definiții de animații personalizate
- Resurse string: definiții de string-uri folosite în resursele XML ale layout-urilor
- Resurse dimensiune: definiții de dimensiuni în dp și sp
- Resurse imagini pentru următoarele densități ale ecranelor dispozitivelor: mdpi, hdpi, xhdpi, xxhdpi.

Punctul de start în a concepe interfața utilizator a unei aplicații Android o constituie crearea layout-urilor corespunzătoare viitoarelor ecrane. Acest lucru înseamnă că ecranele vor fi modelate prin intermediul unor fișiere XML și cu ajutorul resurselor de tip imagine. Layout-urile aplicației de față au fost realizate exclusiv prin intermediul LinearLayout, cel mai simplu tip de layout. Aceasta contrângere a provenit din faptul că prin acest tip de layout poate fi modelat orice ecran, deoarece promovează ordonarea elementelor pe verticală și orizontală, dar și datorită faptului că este foarte ușor de procesat de către telefon, deci foarte ușor de randat în comparație cu alte tipuri de layout-uri mai complexe, cum ar fi RelativeLayout. În plus, câteva dintre ecranele aplicației au presupus folosirea FrameLayout-ului, care presupune că elementele existente în cadrul acestuia sunt afișate unul deasupra celuilalt, primul dintre copiiii săi în ierarhie fiind cel mai de jos afișat. Acest lucru a fost util, spre exemplu, în cazul ecranului „How it works”, în care butoanele din partea de jos a ecranului rămân pe loc, iar background-ul culisează în funcție de mișcarea tactilă aplicată de utilizator asupra ecranului.

Resursele layout obtinute prin intermediul fisierelor XML sunt folosite atat in cadrul activitatilor, cat si in cazul fragmentelor create de activitati. Fragmentele sunt portiuni de UI reutilizabile si care se folosesc in interiorul activitatilor. Activitatile se folosesc de resursele XML pentru a seta un layout prin intermediul metodei `protected void onCreate(Bundle savedInstanceState)` si apeland in cadrul acesteia metoda `setContentView(Layout layout)`, ce primeste ca parametru layout-ul de afisat. In general, resursele se obtin prin intermediul clasei `R`, clasa generata automat de SDK-ul Android, astfel: `R.tipul_resursei.numele_resursei`.

Anterior s-a precizat ca resursele layout sunt utilizate si in cadrul fragmentelor. Acest lucru se realizeaza tot pentru a afisa un layout-ul unui ecran, prin intermediul metodei publice `View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)`. In cadrul acestei metode dezvoltatorul se foloseste de inflater pentru a incarca layout-ul dorit. Iata un exemplu provenit din cadrul fragmentului `InterestFragment.java`:

```
return inflater.inflate(R.layout.interest_main, container, false);
```

Dupa ce s-a obtinut view-ul principal, se pot extrage elementele de UI care il intereseaza pe dezvoltator cu ajutorul metodelor de tipul `findViewById(R.id.some_id)`. In cazul activitatilor aceste apeluri se pot realiza in metoda `onCreate`, iar in cazul fragmentelor acest lucru este recomandat sa se faca in cadrul metodei publice `void onViewCreated(View view, Bundle savedInstanceState)`, apeland metoda `findViewById` asupra obiectului de tip `View` primit ca parametru.

Asadar, metoda `findViewById` cauta in cadrul view-ului care este generat pentru activitate/fragment id-ul trimis ca parametru si returneaza obiecte specifice platformei Android, care constau in elemente de UI: `Button`, `TextView`, `EditText`, `ImageView` si altele. Acestea vor fi utilizate apoi pentru a accesa datele introduse de utilizator sau pentru a modifica textul acestor elementele cu scopul de a schimba continutul ecranelor.

Comunicarea intre activitati si fragmente se face prin intermediul interfetelor. Acestea permit fragmentelor sa notifice activitatea de schimbarile produse in timp ce utilizatorul interactioneaza cu aplicatia. Spre exemplu: in cazul de utilizare in care utilizatorul posteaza un comentariu pe pagina unui eveniment, activitatea „`CommentsActivity`” care este responsabila de managementul comentariilor despre un eveniment este notificata de fragmentul „`CommentsListFragment`” ca a fost postat un comentariu prin intermediul interfetei „`PostCommentInterface`”.

6 Testare și validare

În acest capitol se vor descrie elementele de testare luate în considerare pentru două scenarii de test din cadrul sistemului.

Testarea sistemului s-a realizat la finalul fiecărei etape de proiectare. Respectându-se maniera iterativă de dezvoltare software, fiecare componentă nou introdusă a fost testată înainte de a fi aprobată și adăugată în etapa iterativă curentă a proiectului. Acest lucru îmbunătățește calitatea produsului software și reduce viitoarele posibile erori apărute în cadrul unui modul.

Aplicația a fost testată în mod manual, urmărind scenariile de test întocmite pe baza funcționalităților sistemului. Cu toate acestea, printre obiectivele viitoare ale aplicației se regăsește și integrarea unor componente de testare automată a interfeței utilizator.

În cele ce urmează vor fi descrise două scenarii de test relevante din punct de vedere al complexității elementelor testate.

TC1. Titlu: Creare cont pe baza adresei de email

Descriere: Utilizatorul trebuie să își poată crea un cont cu ajutorul unei adrese de email valide însoțită de o parolă sigură

Condiție de test: Utilizatorul care realizează scenariul de test este utilizatorul neautentificat

Scenariul 1 (Scenariul de succes)

Pasul 1: Utilizatorul accesează aplicația și este întâmpinat de ecranul introductiv de logare

Pasul 2: Utilizatorul apasă butonul „Sign up”

Pasul 3: Utilizatorul apasă butonul „Sign up with email” din cadrul ecranului introductiv de înregistrare

Pasul 4: Utilizatorul completează câmpurile pentru nume, prenume, adresa de email, parola, selectează sexul și data nașterii. Utilizatorul introduce o parolă care are lungimea de minim 6 caractere și o adresă de email validă, adică corespunde formatului xyz@domeniu.ro sau alte top level domain

Pasul 5: Utilizatorul apasă butonul „Sign up” și un nou cont a fost creat.

Pasul 6: Utilizatorul verifică corectitudinea emailului pentru care a fost generat contul, și confirmă „username”-ul generat de către sistem.

Rezultatul așteptat: Utilizatorul apasă butonul „Get started” și contul este salvat.

Scenariul 2

Pasul 1: Utilizatorul accesează aplicația și este întâmpinat de ecranul introductiv de logare

Pasul 2: Utilizatorul apasă butonul „Sign up”

Pasul 3: Utilizatorul apasă butonul „Sign up with email” din cadrul ecranului introductiv de înregistrare

Pasul 4: Utilizatorul selectează un câmp, însă nu îl completează și trece la completarea unui altul câmp.

Rezultatul așteptat: Afisarea unui mesaj de tipul „Value required” în dreptul câmpului care a fost selectat și nu a fost completat.

Scenariul 3

Pasul 1: Utilizatorul acceseaza aplicatia si este intampinat de ecranul introductiv de logare

Pasul 2: Utilizatorul apasa butonul „Sign up”

Pasul 3: Utilizatorul apasa butonul „Sign up with email” din cadrul ecranului introductiv de inregistrare

Pasul 4: Utilizatorul introduce o parola care nu are lungimea de minim 6 caractere

Rezultatul asteptat: Afisarea unui mesaj de tipul „Minimum 6 characters” in dreptul campului destinat introducerii parolei.

Scenariul 4

Pasul 1: Utilizatorul acceseaza aplicatia si este intampinat de ecranul introductiv de logare

Pasul 2: Utilizatorul apasa butonul „Sign up”

Pasul 3: Utilizatorul apasa butonul „Sign up with email” din cadrul ecranului introductiv de inregistrare

Pasul 4: Utilizatorul nu introduce textul in nici unul dintre campurile destinate introducerii de informatii necesare crearii unui cont

Pasul 5: Utilizatorul apasa butonul „Sign up”

Rezultatul asteptat: Afisarea unui mesaj de tipul „Value required” in dreptul tuturor campurilor.

TC2. Titlu: Achizitionarea de bilete

Descriere: Utilizatorul trebuie să își poată crea un cont cu ajutorul unei adrese de email valide însoțită de o parolă sigură

Scenariul 1 (Scenariul de succes)

Conditie de test: Utilizatorul care realizeaza scenariul de test este utilizatorul autentificat si cotul biletului pentru evenimentul selectat este „FREE”

Pasul 1: Utilizatorul acceseaza aplicatia

Pasul 2: Utilizatorul acceseaza un eveniment

Pasul 3: Utilizatorul apasa butonul „Book now”

Pasul 4: Utilizatorul selecteaza un numar de bilete pe care vrea sa-l achizitioneze, numar diferit de zero

Pasul 5: Utilizatorul apasa butonul „Continue”

Rezultatul asteptat: Afisarea unui mesaj de tipul „Press OK to confirm the order or Cancel to cancel the order”.

Scenariul 2

Conditie de test: Utilizatorul care realizeaza scenariul de test este utilizatorul autentificat si cotul biletului pentru evenimentul selectat este „FREE”, iar dispozitivul nu mai este conectat la Internet in momentul in care utilizatorul incearca sa achizitioneze biletul

Pasul 1: Utilizatorul acceseaza aplicatia

Pasul 2: Utilizatorul acceseaza un eveniment

Pasul 3: Utilizatorul apasa butonul „Book now”

Pasul 4: Utilizatorul selecteaza un numar de bilete pe care vrea sa-l achizitioneze, numar diferit de zero

Pasul 5: Utilizatorul apasa butonul „Continue”

Rezultatul asteptat: Afisarea unui mesaj de tipul „No internet connection. Please check the internet connection and try again”.

Scenariul 3

Conditie de test: Utilizatorul care realizeaza scenariul de test este utilizatorul autentificat si cota biletului pentru evenimentul selectat este „FREE”

Pasul 1: Utilizatorul acceseaza aplicatia

Pasul 2: Utilizatorul acceseaza un eveniment

Pasul 3: Utilizatorul apasa butonul „Book now”

Pasul 4: Utilizatorul nu selecteaza un numar de bilete pe care vrea sa-l achizitioneze, acest numar fiind zero in mod implicit

Pasul 5: Utilizatorul apasa butonul „Continue”

Rezultatul asteptat: Afisarea unui mesaj de tipul „Please select at least one ticket to purchase”.

7 Manual de instalare și utilizare

În cadrul acestui capitol sunt detaliate resursele software și hardware necesare pentru instalarea și rularea aplicației Kweekweek, alături de o enumerare a pașilor care trebuie parcurși pentru instalarea aplicației. Mai apoi, este descrisă utilizarea aplicației, din punctul de vedere al utilizatorului, însoțită de imagini relevante pentru fiecare etapă parcursă.

Instalarea aplicației Kweekweek, și implicit utilizarea acesteia, sunt posibile în cazul în care dispozitivul fizic, adică telefonul mobil, al utilizatorului îndeplinește anumite cerințe hardware și software. În următoarele două secțiuni sunt descrise aceste cerințe, iar apoi este descrisă instalarea și utilizarea aplicației.

7.1 Resurse hardware

Aplicația Kweekweek poate fi instalată pe orice dispozitiv cu sistem de operare Android, cu un minim de memorie de 25 de MB, aplicația având la momentul downloadării din Google Play Store dimensiunea de 20 MB. O altă cerință hardware pentru utilizarea aplicației la capacitate maximă ar fi ca dispozitivul să aibă acces la o conexiune la Internet, deoarece astfel se realizează traficul de date dinspre clientul Android și server, respectiv între clientul Android și serviciile Google sau Stripe.

7.2 Resurse software

O resursă software deosebit de importantă pe care dispozitivul pe care se dorește să se realizeze instalarea aplicației Kweekweek trebuie să o îndeplinească este că sistemul de operare Android să înceapă de la API LEVEL 11 (Android HONEYCOMB), până la maxim API LEVEL 19 (Android KITKAT). Acest lucru impune o constrângere asupra dispozitivelor pe care se poate instala această aplicație, tocmai datorită faptului că sistemul utilizează resurse și concepte introduse în „core-ul” sistemului de operare Android începând cu versiunile mai noi.

7.3 Instalarea aplicației

Ultima perioada a adus o serie de îmbunătățiri în ceea ce privește posibilitățile de instalare a aplicațiilor din Google Play Store. O astfel de îmbunătățire este aceea că un utilizator își poate defini dispozitivele deținute, și poate acționa instalarea aplicațiilor pe dispozitivul mobil dintr-un mediu desktop, alegând pe care dintre dispozitivele mobile să se realizeze instalarea. În cazul de față, se va prezenta situația în care utilizatorul dorește instalarea aplicației Kweekweek direct de pe dispozitivul mobil, neimplicând un sistem desktop intermediar.

O necesitate pentru downloadarea aplicației este aceea că dispozitivul să aibă conexiune la Internet, astfel că înainte de a continua cu pașii de mai jos, utilizatorul trebuie să se asigure că telefonul este conectat la Internet.

Instalarea aplicației Kweekweek se realizează prin câțiva pași interactivi, primul dintre aceștia fiind accesarea aplicației Google Play Store, dintre aplicațiile instalate „by default” în telefon. Utilizatorul realizează acțiunea de „tap” (apăsare) pe iconița aplicației Google Play Store, iar aplicația Google Play Store se va deschide.

În meniul principal, situat în partea de sus a ecranului, utilizatorul poate vizualiza o iconiță ce indică o lupă, așa cum reiese și din figura 7.1. Apăsând pe această iconiță, se va deschide tastatură virtuală, în zona meniului va apărea un câmp editabil, iar utilizatorul va putea introduce în câmpul editabil textul „Kweekweek”.

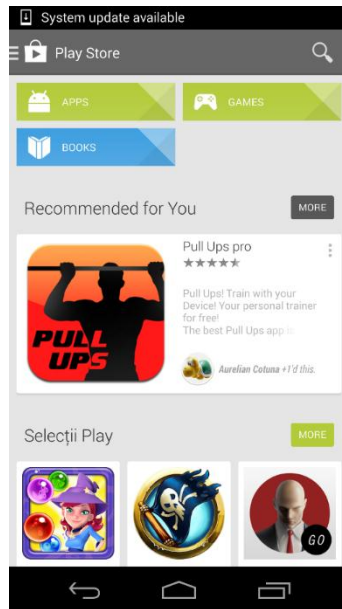


Figura 7.1

În funcție de viteza conexiunii la Internet, pe ecranul dispozitivului vor apărea rezultatele căutării după textul „Kweekweek”. Rezultatele sunt afișate în mod individual, și prin apăsarea asupra rezultatului dorit, utilizatorul va fi redirecționat pentru a vedea informații despre aplicația asupra căruia a acționat prin apăsare. În această pagină se pot vizualiza informații precum: data la care a fost realizată ultima actualizare a aplicației, dimensiunea aplicației, numărul de download-uri, rating-ul obținut în urma voturilor acordate de utilizatori, recenzii ale utilizatorilor, dar și screenshot-uri din cadrul aplicației, pentru a avea un preview al acesteia.

Prin apăsarea butonul „Install” afișat în cadrul acestei pagini, utilizatorului îi vor fi afișate permisiunile pe care trebuie să le accepte dacă dorește să instaleze aplicația. Printre aceste permisiuni se numără: permisiunea de a accesa aplicația de mesagerie (utilizată în cadrul funcționalității de distribuire a informațiilor despre evenimente prin SMS), permisiunea de a accesa USB Storage (utilizată pentru a stoca în baza de date informații despre utilizator, pentru a stoca poza utilizatorului, în cazul în care acesta vrea să își actualizeze poza de profil), permisiunea de a accesa locația utilizatorului, permisiunea de a accesa aplicația de fotografiat (utilă pentru scanarea codului QR), permisiunea de a modifica informațiile personale (se referă la modificarea calendarului telefonului), permisiunea de a realiza apeluri telefonice (utilă pentru funcționalitatea de contactare a utilizatorului care găzduiește un eveniment), permisiunea de a folosi conexiunea la Internet.

Acceptarea condițiilor amintite mai sus conduce la downloadarea fișierului cu extensia .apk și instalarea propriu-zisă a aplicației pe dispozitivul mobil.

7.4 Utilizarea aplicației

7.4.1 Lansarea aplicației

După instalarea aplicației, utilizatorul accesează aplicația Kweeweek, apăsând iconița acesteia. Iconița este ușor de recunoscut, este un simbol grafic personalizat, ilustrând un „background” sub forma unui calendar și litera „K”.

7.4.2 Prima utilizare a aplicației

Utilizatorul este întâmpinat la prima lansare a aplicației cu șase ecrane introductive, care descriu pe scurt funcțiile îndeplinite de aplicație, cu scopul de a explica beneficiile majore ale sistemului. Unul dintre aceste șase ecrane este prezentat și în figura 7.2.

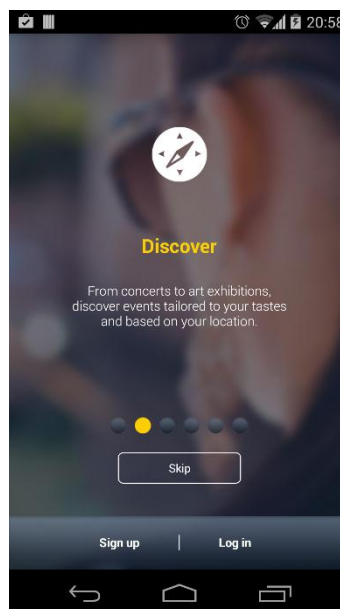


Figura 7.2

În partea de jos a ecranului se pot observa șase buline, culoarea galbenă a unei buline indicând progresul în cadrul celor șase ecrane introductive, un buton cu textul „Skip” și două butoane separate de o linie verticală, prin care utilizatorul poate realiza acțiunile de înregistrare (prin intermediul butonului „Sign up”) sau logare (prin intermediul butonului „Log in”). Acțiunea asociată butonului „Skip” este aceea de a închide acest ecran introductiv, și de a ajunge pe un nou ecran, ecranul de „Discover”.

7.4.3 Înregistrare și logare

Apăsând butonul „Sign up”, utilizatorul este redirecționat către un ecran introductiv de înregistrare, care îi oferă posibilitatea de a se înregistra cu un cont de email, cu ajutorul butonului „Sign up with email”, de a fi redirecționat către un ecran care conține termenii și condițiile sistemului, prin apăsarea textului „Terms&Conditions”, sau către un ecran de logare, apăsând butonul „Log in”. Acest lucru poate fi vizualizat în figura 7.3.

Presupunem că utilizatorul dorește să se înregistreze și apasă butonul „Sign up with email”. Utilizatorul este redirecționat către un ecran în care i se cer informațiile necesare creării unui cont Kweekweek: nume, prenume, adresa de email, parolă, sex, dată nașterii. Utilizatorul completează în câmpurile cerute informații și apoi apasă butonul „Sign up”, care are ca și efect crearea unui cont. Neintroducerea datelor în câmpurile amintite anterior este semnalată de mesajul: „Value required!”, iar introducerea unei parole mai scurtă de șase caractere este semnalată prin mesajul: „Minimum 6 characters!”. Figura 7.4 ilustrează cele amintite mai sus referitoare la ecranul de înregistrare.

Înregistrarea utilizatorului îl va redirecța pe acesta către ecranul „Discover”, iar utilizatorul este logat în aplicație.

În cazul în care utilizatorul are deja un cont Kweekweek, acesta poate alege din secțiunea „How it works” opțiunea de a se autentifica, prin apăsarea butonul „Login”. Utilizatorul va fi redirecționat către un ecran introductiv de logare, care pune la dispoziție două opțiuni de logare în aplicație: prin intermediul unui cont de Facebook sau prin intermediul unui cont Kweekweek, dar și un buton „Sign up”, care redirecțează utilizatorul către pagină introductivă de înregistrare descrisă anterior. În figura 7.5 se poate vizualiza acest ecran.

Ecranul de logare prin contul Kweekweek afișează două câmpuri destinate introducerii adresei de email asociate contului Kweekweek și parola corespunzătoare lui, textul „Forgot password?” și un button cu textul „Login”. Butonul „Login” realizează logarea utilizatorului în aplicație, dacă datele introduse sunt corecte. Logarea cu succes îl va conduce pe utilizator către ecranul „Discover”.

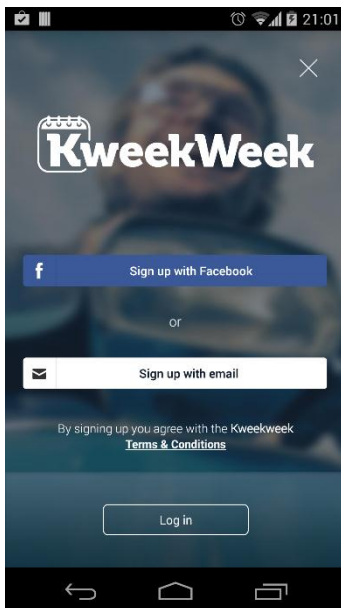


Figura 7.3

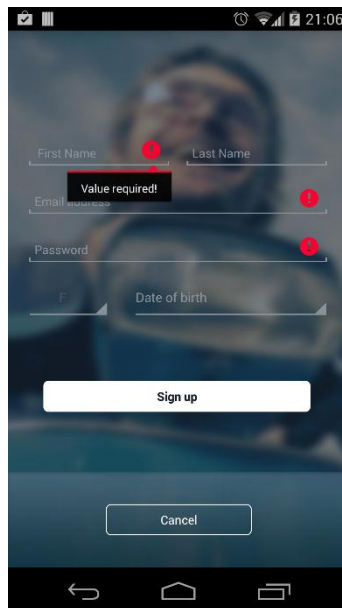


Figura 7.4

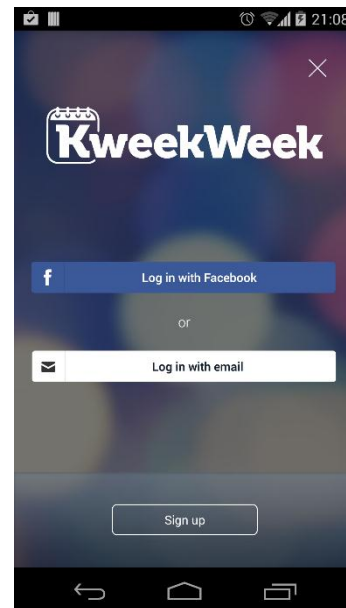


Figura 7.5

Dacă utilizatorul și-a uitat parola contului, poate apăsa textul „Forgot password?” și apoi, este redirecțat la un ecran de recuperare a parolei. Aici poate introduce adresa de email a contului,

și apăsând butonul „Reset password”, sistemul trimite un mail la adresa introdusă, astfel că utilizatorul o să poată să își schimbe parola.

Aplicația are o structură circulară, astfel că din ecranul introductiv de logare se poate ajunge la ecranul introductiv de înregistrare și viceversa, cu ajutorul butoanelor din partea de jos a ecranelor enumerate.

7.4.4 Utilizarea meniului lateral

Meniul lateral poate fi accesat din momentul în care utilizatorul ajunge în ecranul principal al aplicației, cel de „Discover”. Meniul aplicației este accesibil din toate ecranele aplicației în care în stânga-sus a ecranului apar 3 linii orizontale, alături de sigla aplicației. Acest meniu este alcătuit din următoarele elemente:

- Un prim element care redirecțiază utilizatorul către pagina de „Log în”, dacă utilizatorul nu este logat în aplicație, sau la un ecran destinat afișării informațiilor despre utilizator, în caz contrar
- „Discover” care conduce utilizatorul către ecranul „Discover”
- „Calendar” prin intermediul căruia utilizatorul o să poată vizualiza evenimentele pentru care a cumpărat bilete
- „Interests” care permite utilizatorului să-și definească tipurile de categorii de evenimente preferate
- „Rewards” cu ajutorul căruia utilizatorul va beneficia de promoțiile Kweekweek sau va invita prieteni să se alăture rețelei Kweekweek
- „Event management” datorită căruia utilizatorul o să poată să contorizeze statusul aplicațiilor publicate de el
- „Settings” care oferă diferite facilități: log out, editarea profilului, vizualizarea termenilor și condițiilor aplicației, editarea unor setări.

7.4.5 Vizualizarea evenimentelor

Un alt ecran, poate cel mai semnificativ pentru descoperirea evenimentelor, îl constituie ecranul „Discover” la care se poate ajunge prin mai multe modalități: din ecranul „How it works”, acționând asupra butonului „Skip”, din ecranele introductive „Sign up” sau „Login”, prin intermediul cruciuliței din partea dreapta-sus a ecranului, dar și dacă utilizatorul se loghează sau se înregistrează. La prima accesare a acestui ecran, se vor afișa evenimentele din ziua curentă, aspect indicat și de textul elementului selectat din meniul navigabil din partea de sus a ecranului: Today. Mai mult, se observă că în meniul din partea dreapta-sus a ecranului există trei iconițe: o lupă, un marker specific localizărilor geografice și un simbol grafic care desemnează notificările utilizatorului. Acestea au rolul de a realiza căutări, de a realiza tranziția către ecranul în cadrul căruia se pot vizualiza evenimente de harta interactivă, respectiv de a afișa notificările utilizatorului logat. Dacă nici un utilizator nu este logat, ultima iconiță nu este vizibilă. Figura 7.6 captează cele spuse până acum despre ecranul „Discover”.

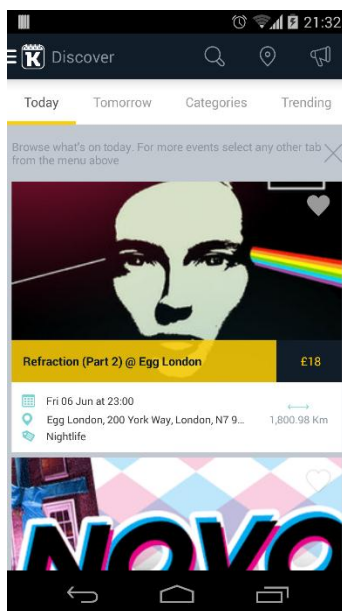


Figura 7.6

Acest ecran permite navigarea pe orizontală, pentru a descoperi evenimente din diferite categorii: Today, Tomorrow, în funcție de categorie, Trending, Following, Redeemable. Fiecare dintre celulele afișate în cadrul acestui ecran reprezintă un eveniment, și așa cum se poate observa și în figura 7.6, fiecare eveniment are asociată o poză, un titlu, prețul biletului, data și locația la care se desfășoară evenimentul, categoria din care face parte și distanța de la locația curentă până la locația evenimentului. Simbolul grafic inimioară din partea dreapta-sus a evenimentului indică faptul că utilizatorul poate adăuga evenimentul în „wishlist”. Dacă acest simbol este apăsat, însă nu utilizatorul nu este logat, aplicația îl va redirecta către pagina de „Sign up”. Vizualizarea evenimentelor adăugate în „wishlist” va fi prezentată în subcapitolul 7.4.12.

7.4.6 Vizualizarea detaliilor despre un eveniment

Acest ecran are ca scop prezentarea detaliilor despre un eveniment ales din ecranul „Discover”. Meniul din partea dreapta-sus a ecranului conține de data această inimioara, simbolul grafic menționat și în cadrul ecranului „Discover”, ceea ce înseamnă că utilizatorul poate adăuga evenimentul în „wishlist” și din acest context, precum și simbolul grafic pentru „share”. Acesta din urmă permite utilizatorului să distribuie informațiile despre eveniment și altor persoane, prin intermediul rețelelor sociale Facebook, Twitter, sau prin Email sau SMS. Figurile 7.7 și 7.8 prezintă conținutul ecranului „Event details”.

Continuând cu informațiile despre pagina „Event details”, se va descrie ce se poate vizualiza pe acest ecran. Așa cum se poate observa și în figura 7.7, evenimentul are asociat una sau mai multe poze. Acestea sunt prezentate într-o galerie foto, care schimbă poză vizualizată la fiecare 2 secunde, însă pozele pot fi vizualizate și dacă utilizatorul intervine și realizează acțiunea de „swipe” asupra galeriei. În continuarea galeriei foto, sunt afișate detalii despre eveniment, cu posibilitatea de a expanda sau restrânge această căsuța.

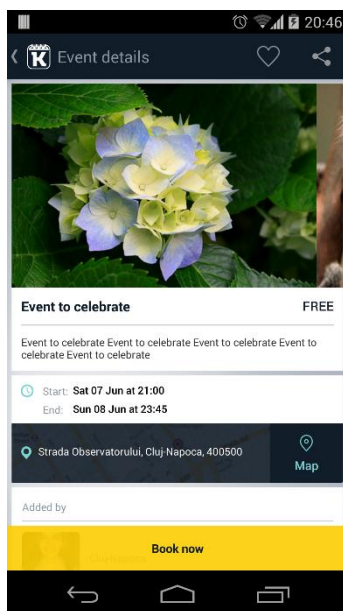


Figura 7.7

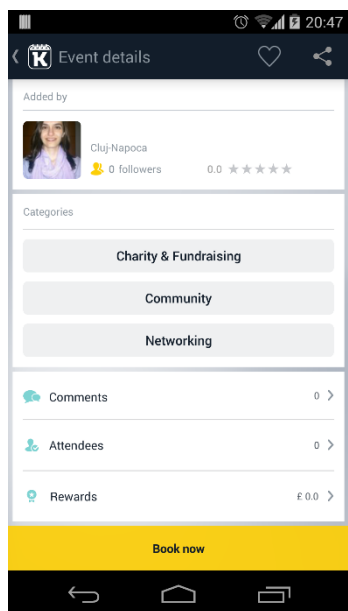


Figura 7.8



Figura 7.9

Mai apoi, sunt afișate data începerii și data finalizării evenimentului și locația acestuia, cu posibilitatea de a selecta vizualizarea evenimentului pe hartă. Figura 7.8 reprezintă continuarea imaginii 7.7, și ilustrează alte trei mari aspecte importante prin care se poate caracteriza un eveniment: persoana de care a fost adăugat evenimentul, categoriile din care face parte evenimentul și o secțiune dedicată comentariilor, persoanelor care participă la eveniment și bonusurilor evenimentului. În figura 7.9 este exemplificată vizualizarea unui eveniment pe hartă.

7.4.7 Cumpărarea biletelor

În figura 7.7 se poate identifica un buton cu textul „Book now” cu ajutorul căruia utilizatorul este redirecționat către ecranul „Buy tickets” unde utilizatorul poate alege ce tipuri de bilete dorește să achiziționeze și cantitatea lor. În timp ce utilizatorul selectează bilete pe care intenționează să le cumpere este afișat prețul final al tranzacției. Figura 7.10 descrie cele amintite mai sus în cadrul acestui subcapitol, iar în figura 7.11 ilustrează etapă finală pentru a termina o tranzacție, la care se poate ajunge apăsând butonul „Continue” din partea dreapta-jos a ecranului „Buy tickets”. Utilizatorul introduce datele referitoare la cardul cu care se va realiza plata online și apoi acționează butonul „Pay now”. Utilizatorul poate alege să îi fie salvate datele despre cardul curent, bifând opțiunea „Save card”.

7.4.8 Vizualizarea biletelor achiziționate

Vizualizarea biletelor cumpărate de un utilizator se poate realiza din cadrul notificărilor utilizatorului sau din cadrul meniului, accesând secțiunea „Calendar”.

Pentru început, va fi detaliată prima situație, cea a vizualizării biletelor din cadrul notificării. O notificare de tipul: „You bought ticket(s)”, continuată de numele unui eveniment, semnalează faptul că utilizatorul a cumpărat bilete pentru acel eveniment, iar aceste bilete pot fi vizualizate apăsând pe butonul „See tickets”, lucru ilustrat și în figura 7.12.

Apsarea butonului „See tickets” va conduce către un ecran în care se detaliază informațiile despre eveniment, alături de un QR code. Dacă utilizatorul a cumpărat mai multe bilete, utilizatorul se poate vizualiza pe rând, realizând acțiunea de „swipe”. Asa cum s-a menționat anterior, vizualizarea biletelor se poate realiza și prin accesarea secțiunii „Calendar”, categoria „Booked”, iar apoi apăsând pe butonul „Ticket” din dreptul evenimentului pentru care se dorește vizualizarea biletului.

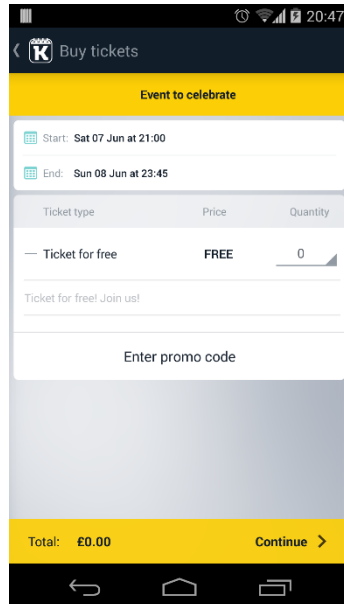


Figura 7.10

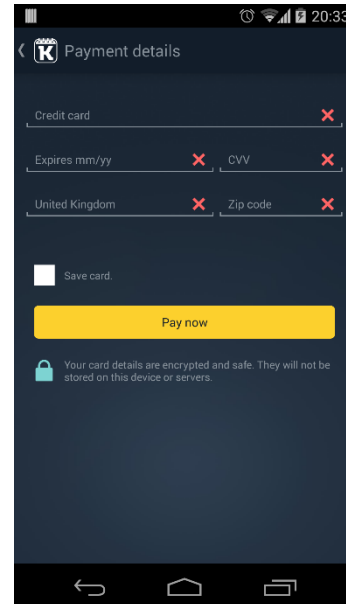


Figura 7.11

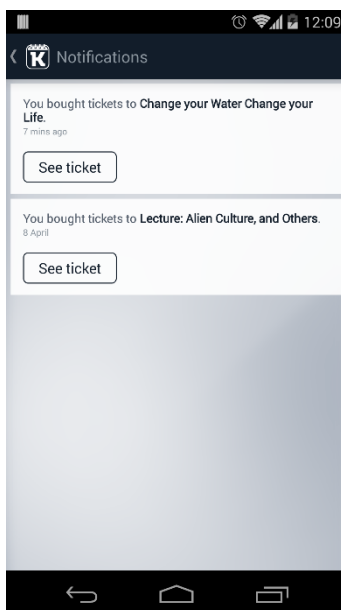


Figura 7.12

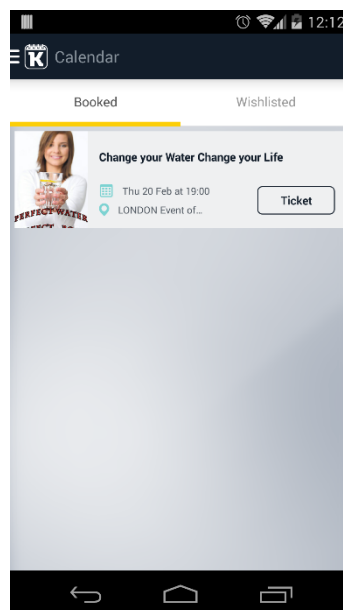


Figura 7.13

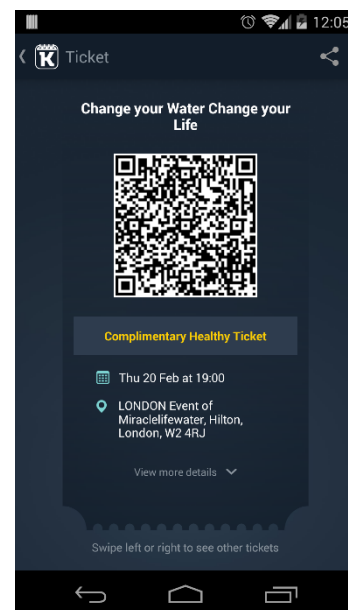


Figura 7.14

7.4.9 Postarea de comentarii pe pagina unui eveniment

Utilizatorul poate accesa evenimentele pentru care a achizitionat bilete sau pentru care este gazda si poate comenta pe pagina evenimentului, pentru a-si exprima opinia cu privire la acel eveniment. Acest lucru se realizeaza accesand sectiunea „Comments” de pe pagina in care se face afisarea detaliilor despre evenimente. Apoi utilizatorul introduce comentariul in campul editabil care are indiciul „Reply to user” si apasa butonul „Reply”. Utilizatorul poate observa ca lista comentariilor este actualizata si comentariul a fost postat. Aceste informatii se pot vizualiza si in figura 7.15 si figura 7.16.

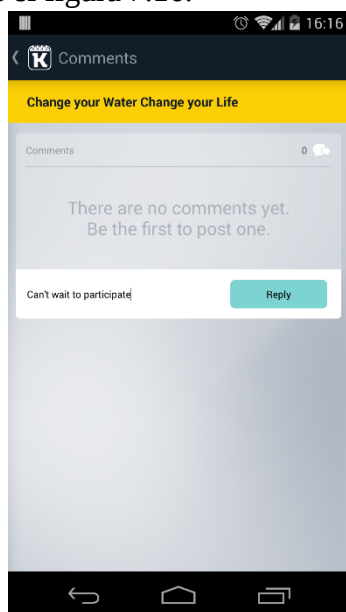


Figura 7.15

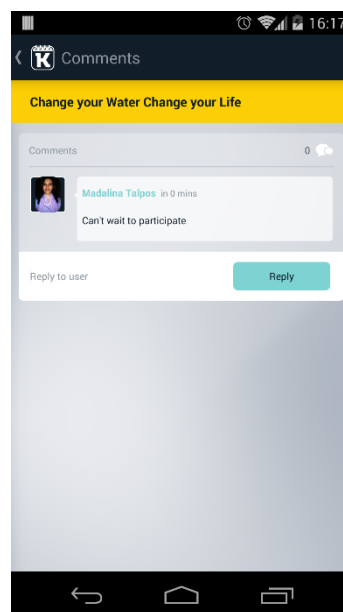


Figura 7.16

7.4.10 Căutarea evenimentelor

Căutarea evenimentelor se poate realiza din cadrul meniului navigabil prezentat în cadrul ecranului „Discover”, dar și apăsând simbolul grafic lupă, din meniul ecranului „Discover”. Cea mai simplă căutare care se poate realiza este căutarea după nume, care se realizează introducând textul după care dorește să se facă căutarea în câmpul editabil cu „placeholder”-ul „Find events”.

Căutarea evenimentelor se poate realiza și aplicând filtre asupra prețului. Există trei tipuri de filtre referitoare la prețul biletelor: afișarea tuturor evenimentelor indiferent de preț, afișarea evenimentelor pentru care biletul este gratuit și afișarea evenimentelor a căror bilet trebuie plătit. Asupra evenimentelor se poate aplica și o sortare, una dintre următoarele disponibile: după distanță, după preț sau după dată. În plus, se pot selecta evenimente în funcție de locația lor (locația curentă sau o locație introdusă de utilizator, asistată de „autocomplete”) dar și după dată. Pe lângă aceste filtre, se poate adăuga și filtrarea după categoria din care face parte evenimentul. Toate aceste aspecte sunt exemplificate în figurile 7.17 și 7.18, cea din urmă ilustrând rezultatul unei căutări.

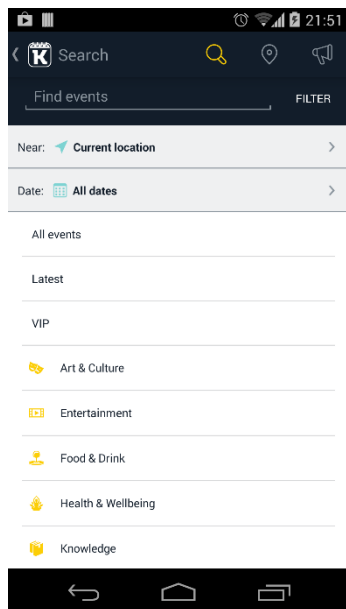


Figura 7.17

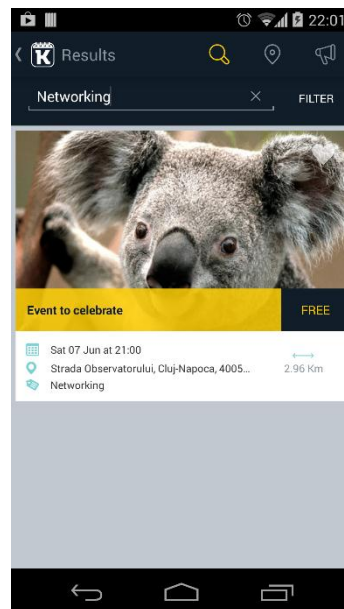


Figura 7.18

7.4.11 Vizualizarea și modificarea categoriilor de evenimente de interes

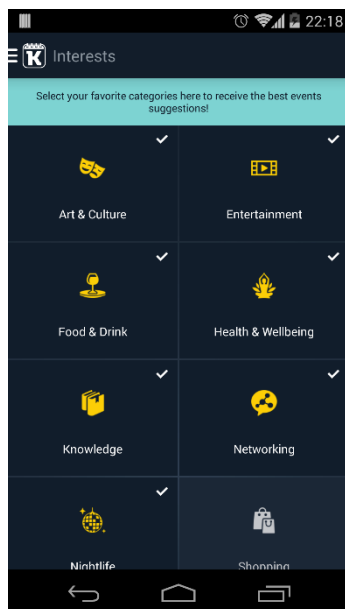


Figura 7.19

Utilizatorul poate modifica preferințele sale în ceea ce privește categoriile de evenimente de care este interesat. Acest lucru poate fi realizat accesând secțiunea „Interests” din meniul lateral al aplicației. Utilizatorul poate realiza acțiunea de „swipe” vertical, pentru a vizualiza întregile categorii, poate activa sau dezactiva o categorie care îl interesează apăsând asupra respectivei categorii. O categorie este activă dacă imaginea care o însoțește este galbenă și are în partea dreapta-sus o bifa, iar o categorie inactivă poate fi recunoscută după imaginea gri care o însoțește.

si lipsa bife in coltul din dreapta-sus a categoriei. Toate aceste aspecte pot fi observate in figura 7.19.

7.4.12 Vizualizarea evenimentelor adaugate in „wishlist”

Evenimentele pe care utilizatorul le-a adaugat in „wishlist” se pot vizualiza accesand sectiunea „Calendar” din meniul lateral al aplicatiei, iar apoi categoria „Wishlisted”. Evenimentele vor fi afisate in o lista de tipul celei din figura 7.20. Din cadrul acestei liste, utilizatorul poate accesa detalii despre un eveniment din lista, actionand asupra evenimentului de care este interesat, sau poate merge direct catre pagina de achizitionare de bilete pentru un eveniment din lista, apasand butonul „Book” pentru evenimentul la care doreste sa participe.

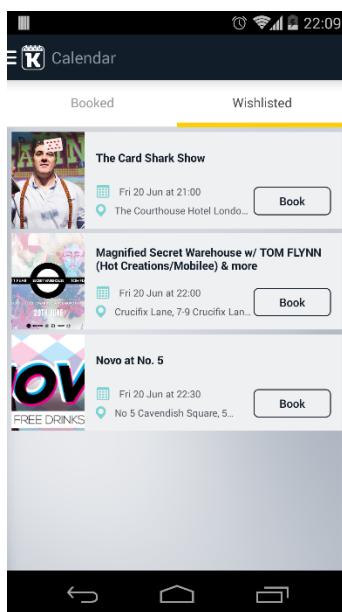


Figura 7.20

7.4.13 Vizualizarea propriilor evenimente

Evenimentele postate de utilizatorul logat se pot monitoriza accesând secțiunea “Event management” a meniului aplicației. Secțiunea permite vizualizarea evenimentelor care urmează să aibă loc și a celor care au avut deja loc, însă comportamentul lor este similar. Acest ecran este afișat așa cum reiese și din figura 7.21, iar mai departe în figura 7.22 este prezentat ecranul la care este condus utilizatorul dacă apasă pe butonul „Manage”. În figura 7.22 se observă că sunt furnizate informații precum: suma încasată din vânzarea biletelor, numărul biletelor vândute, tipul biletelor și câteva statistici referitoare la vânzări.

În plus, în figura 7.23 este ilustrat cum se realizează „check in”-ul unui participant la eveniment: scanarea codului QR regăsit pe biletul achiziționat de persoana participantă, apăsând butonul „Scan QR code”. Tot pe acest ecran sunt afișate informații despre câte persoane au realizat acțiunea de „check in”, dar și care sunt participanții înregistrați pentru acel eveniment.

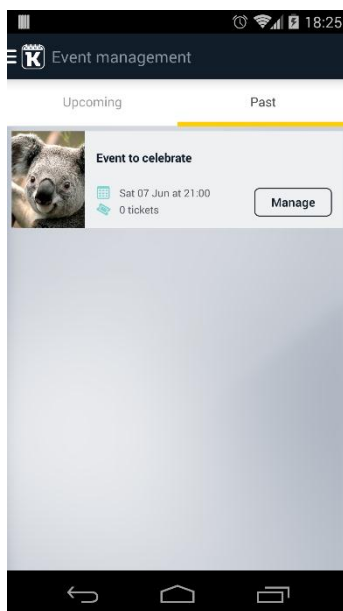


Figura 7.21

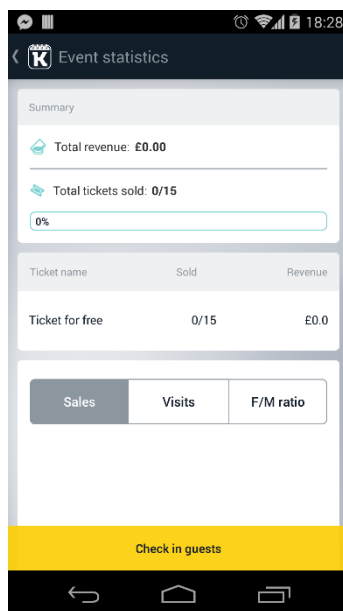


Figura 7.22

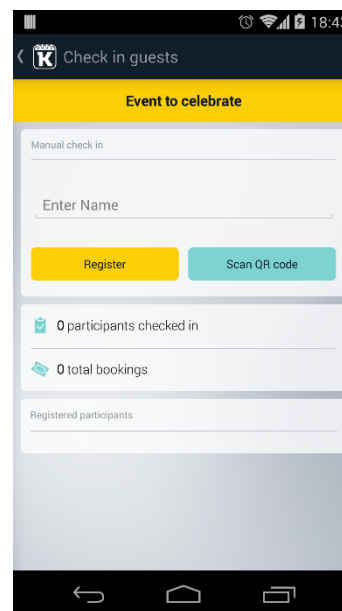


Figura 7.23

7.4.14 Setările aplicației

Acest ecran poate fi accesat prin intermediul meniului lateral, din secțiunea „Settings”. În continuare se vor detalia ecranele disponibile în cadrul setărilor aplicației. Figura 7.24 prezintă acest ecran, și se poate observa că din acest ecran utilizatorul poate realiza acțiunea de „Log out”.

În primul rând, utilizatorul își poate actualiza profilul prin intermediul secțiunii „Profile”, în cadrul căreia sunt afișate informațiile utilizatorului: o poză a utilizatorului, numele, prenumele, username, email, sex, dată nașterii, și o scurtă prezentare a utilizatorului. Poza utilizatorului poate fi schimbată apăsând pe iconița care indică poza actuală a utilizatorului sau poza prestabilită, dacă utilizatorul nu are la momentul actual o poză. Apoi utilizatorul poate alege o poză dintre cele existente în memoria telefonului sau poate face o nouă poză. De altfel, utilizatorul își poate modifica datele prezente în cadrul acestui ecran, și le poate salva, apăsând butonul „Save”. Toate cele amintite referitoare la secțiunea de „Profile” pot fi urmărite în figura 7.25.

Secțiunea „Past orders” afișează evenimentele la care utilizatorul a participat în trecut, și are o afișare identică cu cea întâlnită și în cadrul secțiunii „Calendar”.

Cea de-a treia secțiune regăsită în ecranul setărilor este cea de vizualizare a setărilor active pentru notificările pe telefon sau email. Utilizatorul are posibilitatea de a activa sau dezactiva aceste notificări pentru a reduce informația care este redirecționată către el atunci când anumite acțiuni au loc. Astfel, utilizatorul poate alege să fie sau nu notificat de acțiunile în rețelele sociale ale prietenilor din cadrul aplicației Kweekweek, de acțiunile care au loc atunci când un eveniment cumpărat este anulat, de faptul că un utilizator pentru care utilizatorul curent a dat „follow” postează un eveniment, și de noutățile aplicației. Figura 7.26 prezintă cele amintite în cadrul acestei secțiuni.

A patra secțiune este cea de „How it works” care prezintă încă o dată tutorialul urmărit de utilizator la prima interacțiune cu aplicația, și a cărui comportament a fost descris în cadrul

subcapitolului 7.4.2, precum și termenii și condițiile, care sunt accesibile și din cadrul ecranului de „Sign up”, descris în subcapitolul 7.4.3.

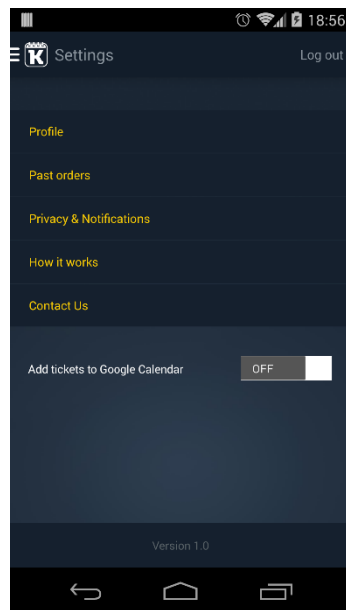


Figura 7.24

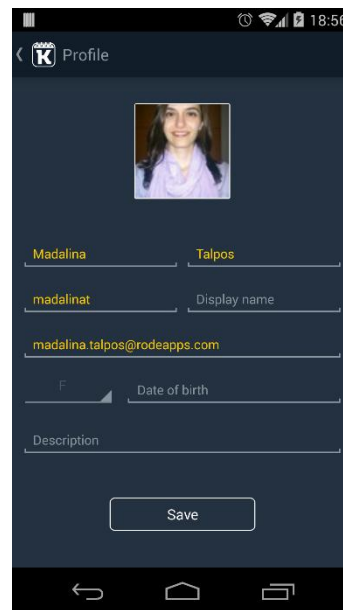


Figura 7.25

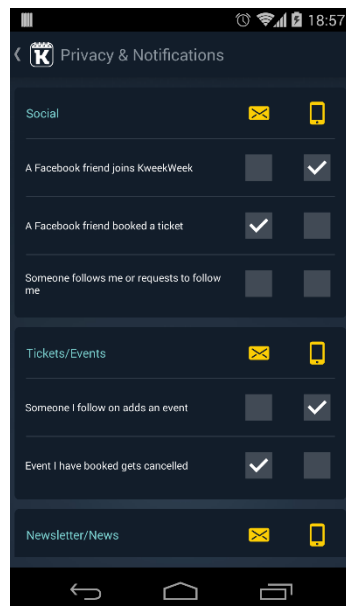


Figura 7.26

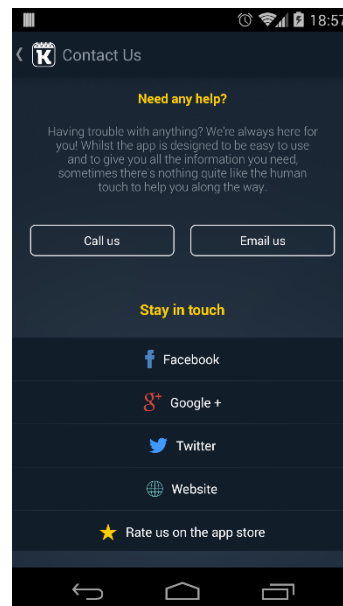


Figura 7.27

În cele din urmă, secțiunea „Contact us” pune la dispoziția utilizatorilor posibilitatea de a intra în contact cu persoanele care se ocupă de managementul aplicației, dar și de a interacționa cu aceștia prin intermediul rețelelor sociale Facebook, Google+, Twitter, pe Website. Această secțiune se poate vizualiza în figura 7.27.

8 Concluzii

În acest capitol vor fi prezentate principalele realizări actuale ale proiectului alături de idei pentru dezvoltările ulterioare.

8.1 Realizări

Proiectul a constituit o oportunitate de dezvoltare tehnică pentru dezvoltatorul software. Astfel, acesta din urmă a parcurs următoarele etape:

- a dobândit cunoștințe solide despre platforma Android, despre conceptele cu care se operează în cadrul acesteia, cât și despre „good practice”-urile aferente tehnologiei mobile
- a participat activ în etapa de analiză și proiectare a proiectului, fapt care a format o viziune de ansamblu asupra proiectului, a conturat o perspectivă asupra posibilelor implementări
- a implementat sistemul proiectat într-o manieră iterativă, care a permis parcurgerea tuturor etapelor dezvoltării software pentru fiecare iteratie în parte
- a căutat soluții pentru diversele provocări tehnice
- a ales soluțiile optime pentru a rezolva „task”-urile fiecărei iteratii.

Proiectul a dus la îndeplinire toate obiectivele impuse și enunțate în cadrul capitolului 2, astfel ca proiectul final permite utilizatorului să:

- își creeze un cont corelat cu o adresă validă de email
- se logheze în cadrul aplicației prin intermediul rețelei sociale Facebook
- vizualizeze evenimente aplicând diverse filtre
- afle detalii despre evenimentele de care este interesat
- cumpere bilete pentru evenimente și să participe la evenimente
- facă acțiunea de „share” asupra evenimentelor favorite
- comenteze pe pagina evenimentelor la care participă sau pentru care este gazda
- vizualizeze evenimentele pe harta interactivă
- beneficieze de bonusurile promotionale
- caute evenimente după anumite filtre
- vizualizeze statistici pentru evenimentele proprii
- realizeze acțiunea de „check in” pentru participanții la un eveniment organizat de acesta
- adauge evenimente în „wishlist”
- își modifice profilul.

Mai mult, proiectul a încurajat dobândirea de cunoștințe noi în ceea ce privește diferitele „framework”-uri disponibile la momentul actual.

8.2 Dezvoltări ulterioare

Orice sistem poate fi îmbunătățit și actualizat pentru a se mapa pe cerințele și necesitățile utilizatorilor țintă, pentru a fi pe măsura așteptărilor acestora. Printre îmbunătățirile care ar putea fi aduse sistemului actual pentru o versiune viitoare, se enumerează:

- implicarea mai multor rețele sociale în procesul de logare/„share”
- posibilitatea vizualizării în mod „offline” a biletelor, deoarece la momentul actual acestea se pot vizualiza doar dacă dispozitivul mobil este conectat la Internet
- realizarea unei componente de testare automată a interfeței utilizator, folosind chiar „tool”-ul pus la dispoziție de „SDK” -ul de Android, „uiautomatorviewer”.
- oferirea ca alternativă la introducerea manuală a informațiilor despre cardul de plată, scanarea automată a acestuia. Acest lucru presupune integrarea unei componente de recunoaștere a formelor, mai precis a literelor și a cifrelor, utilizând camera nativă a dispozitivului mobil.