

UNIVERSITATEA POLITEHNICA BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE



PROIECT DE DIPLOMĂ

Task Planner
Agenda digitala pentru programari

Gabriel Nicușor Orzață

Coordonator științific:
Prof. dr. ing. Florin Radulescu

BUCUREȘTI

2024

UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE DEPARTMENT



DIPLOMA PROJECT

Task Planner
Digital agenda for appointments

Gabriel Nicușor Orzață

Thesis advisor:
Prof. dr. ing. Florin Radulescu

BUCHAREST

2024

CUPRINS

Sinopsis	3
Abstract.....	3
1 Introducere	4
1.1 Context.....	4
1.2 Problema.....	4
1.3 Obiective	4
1.4 Structura lucrării	5
2 Analiza și specificarea cerințelor.....	5
2.1 Clientul	5
2.2 Motivație.....	5
2.3 Cerintele aplicatiei	6
2.3.1 Cerințe funcționale	6
2.3.2 Cerințe nefuncționale	6
3 Abordări existente	7
3.1 Square Appointments	7
3.2 Setmore.....	9
3.3 Discuție comparativă	10
4 Soluția propusă	11
4.1 Descrierea soluției	11
4.2 Schema bloc	12
4.2.1 Navigation Bar.....	12
4.2.2 Vizualizare în format lunar.....	13
4.2.3 Vizualizare în format săptămânal	14
4.2.4 Vizualizare în format de trei zile	14
4.2.5 Vizualizare în format zilnic	15
4.2.6 Gestionarea evenimentelor	15
4.2.7 Servicii	18
4.2.8 Setări	20
4.2.9 Tabel pentru clienți vechi.....	21
4.2.10 Tabel pentru totalul plăților.....	22

4.2.11	Promoții	23
5	Detalii de implementare	24
5.1	Tehnologii utilizate.....	24
5.1.1	Android	24
5.1.2	Java.....	24
5.2	Platforma pentru dezvoltare	24
5.3	Structura proiectului	25
5.3.1	Configurația proiectului	26
5.4	Descrierea modulelor aplicației	27
5.4.1	Pachetul „activities”	27
5.4.2	MainActivity	27
5.4.3	ApiAsyncTask	28
5.4.4	AddEventActivity.....	29
5.4.5	MessageClientsActivity	29
5.4.6	Resources	30
5.5	Baza de date.....	31
6	Evaluarea rezultatelor.....	34
7	Concluzii si dezvoltări ulterioare.....	37
7.1	Concluzii	37
7.2	Dezvoltări ulterioare	38
8	Bibliografie	38
9	Anexe	40
9.1	Anexa 1: Formular pentru analiza cerințelor	40

SINOPSIS

Viața cotidiană în societatea modernă a devenit din ce în ce mai aglomerată și solicitantă în ultimii ani, iar, în paralel cu aceasta, tehnologia a cunoscut o evoluție exponențială în ceea ce privește impactul acesteia asupra omului de rând. Părerea mea este că scopul tehnologiei este acela de a ne simplifica viața și de a ne permite să ne bucurăm cât mai mult de momentele în care suntem deconectați de la aceasta.

Din aceste considerente, am ales să dezvolt o aplicație care să facă organizarea timpului cât mai ușoară și intuitivă: Task Planner. Aceasta își propune să ofere utilizatorului o metodă simplă și eficientă prin care să își gestioneze evenimentele din viața profesională, dar și personală, prin intermediul unei interfețe grafice intuitive și ușor de folosit.

ABSTRACT

Everyday life in modern society has become increasingly busy and demanding in recent years, and in parallel with this, technology has seen an exponential evolution in terms of its impact on the common man. My view is that the purpose of technology is to simplify our lives and allow us to enjoy the moments when we are disconnected from it as much as possible.

For these reasons, I have chosen to develop an application that makes time organization as easy and intuitive as possible: Task Planner. It aims to provide the user with a simple and efficient way to manage the events of his professional and personal life through an intuitive and easy to use graphical interface.

1 INTRODUCERE

1.1 Context

Industria aplicațiilor mobile este în plină dezvoltare, fapt datorat de creșterea utilizării smartphone-urilor. Acestea au devenit o parte integrantă a vieții oamenilor, nu doar datorită capacității lor de a efectua apeluri telefonice. Smartphone-urile au devenit aparate de fotografiat, GPS, o modalitate prin care se pot afla știri și informații de ultimă oră, dar și o sursă de divertisment.

La momentul actual, există aproape 7 miliarde de utilizatori de telefoane smart [1], iar media timpului petrecut de către oameni online pe aceste dispozitive este de 4 ore pe zi [2]. Așadar, putem spune cu ușurință că telefonul a devenit un aspect indispensabil din viețile noastre. Aceste valori sunt în continuă creștere, iar în rândul tinerilor sunt considerabil mai mari. Trendul înspre care ne îndreptăm în ceea ce privește dependența de smartphone-uri, social media și nu numai, nu este unul benefic pe termen lung și sunt de părere că întotdeauna ar trebui să existe o limită la cât de mult face parte din viața noastră (personală) tehnologia.

Din acest motiv, am decis să creez „Task Planner”, o aplicație mobilă care să ajute utilizatorul să economisească timp prin organizarea agendei profesionale, cât și personale. Aceasta oferă totodată posibilitatea de a rămâne la curent cu ecosistemul tehnologic folosit.

1.2 Problema

Aplicația „Task Planner” a fost dezvoltată cu scopul de a rezolva problema gestionării programărilor și evenimentelor de orice fel. Această aplicație se adresează persoanelor cu o agendă foarte încărcată, fie că este vorba despre cei care au o afacere ce necesită lucrul constant cu oameni, prestarea de servicii sau organizarea de evenimente. Versatilitatea aplicației permite utilizatorului să o folosească în diverse moduri, eliminând astfel necesitatea de a avea o aplicație specifică pentru fiecare nevoie.

1.3 Obiective

Aplicația a fost proiectată pentru a automatiza gestiunea timpului în cadrul afacerii utilizatorului și pentru a facilita interacțiunea dintre el și clienți. Aceasta își propune să devină principala modalitate de organizare a agendei profesionale a utilizatorului, oferindu-i totodată posibilitatea de a-și integra și aspectele personale ale vieții sale.

Așadar, „Task Planner” a fost creată pentru a rezolva problema unui program de lucru dezordonat și pentru a menține un echilibru între viața personală și cea profesională a utilizatorului.

1.4 Structura lucrării

În continuare, voi descrie într-un mod sumar următoarele secțiuni ale lucrării, pentru a expune cât mai clar procesul de dezvoltare al aplicației:

- Analiza și specificarea cerințelor: determină publicul țintă, funcționalitățile și scenariile de utilizare ale aplicației.
- Abordări existente: descrie soluțiile existente pe piață împreună cu o discuție comparativă între acestea și „Task Planner”.
- Soluția propusă: descrie în detaliu componentele aplicației, oferind o vedere de ansamblu asupra acesteia.
- Detalii de implementare: expune aspectele tehnice ce țin de implementarea bazei de date, a interfeței grafice și a funcționalităților aplicației.
- Evaluarea rezultatelor: centralizarea feedback-ului primit de la potențialii clienți.
- Concluzii și dezvoltări ulterioare: oferă o sumarizare a întregii lucrări și include sugestii pentru dezvoltările viitoare ale aplicației.

2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Acest capitol își propune să prezinte tiparul clienților pentru care este dedicată aplicația, să descrie cerințele funcționale și nefuncționale ale acesteia, cât și scenariile de utilizare.

2.1 Clientul

Determinarea publicului țintă pentru o aplicație mobilă este o acțiune ce trebuie făcută încă din fazele incipiente ale proiectului și are o importanță majoră în determinarea cerințelor, funcționalităților și particularităților ce vor urma să fie implementate.

Un studiu recent [3] a demonstrat că mai puțin de 1 din 5 persoane folosesc o metodă de gestionare a timpului, iar, din cauza lipsei acesteia, sunt copleșiți de programul de muncă și nu pot avea o productivitate sporită. Cercetarea a inclus participanți cu vârste între 18 și 65 de ani, iar în urma acesteia s-a constatat că toți se confruntă cu dificultăți în organizarea eficientă a timpului. Așadar, utilizatorul aplicației „Task Planner” este o persoană cu un program încărcat și care are nevoie de o soluție simplă pentru a-și organiza programul.

2.2 Motivație

Motivația principală a proiectului „Task Planner” derivă din dorința de a rezolva problemele întâmpinate de persoanele cu un program foarte încărcat, în special a celor care dețin o afacere de dimensiuni reduse.

În urma discuțiilor mele cu potențialii utilizatori (vezi Anexa 1), am observat că majoritatea dintre ei nu folosesc o aplicație de gestiune a timpului. Principala cauză este complexitatea ridicată și limitările aplicațiilor existente, care le fac dificil de utilizat pentru organizarea unei mici afaceri. Acest lucru m-a motivat să creez o aplicație versatilă și intuitivă, în care

utilizatorul să găsească soluții pentru orice aspect legat de gestionarea agendei sale profesionale.

Pe lângă aceste motive, personal, am fost mereu interesat de modul în care tehnologia poate să ne facă viața mai ușoară. Cred cu tărie că tehnologia ar trebui folosită în acest scop și de aceea am ales să dezvolt „Task Planner”.

2.3 Cerințele aplicației

Ținând cont de obiectivele aplicației, au fost propuse cerințele principale ale acesteia, care au fost apoi prezentate unui grup de persoane care respectau tiparul de client pentru „Task Planner”. În urma feedback-ului și sugestiilor primite, s-au definitivat cerințele finale ale produsului, dar și identificat o parte nișată de clienți.

2.3.1 Cerințe funcționale

- Crearea și editarea evenimentelor într-un calendar principal.
- Vizualizarea calendarului pe zile, săptămâni sau luni.
- Gestionarea diferitelor tipuri de servicii, cu posibilitatea salvării acestora pentru a putea fi refolosite ulterior.
- Trimiterea de mesaje de reminder personalizate pentru clienți.
- Posibilitatea promovării unor evenimente viitoare sau oferte speciale.
- Generarea de statistici privind activitatea și comportamentul clienților.
- Sincronizarea cu Google Calendar: pentru îmbinarea agendei personale cu cea profesională a utilizatorului.
- Asigurarea unei interfețe grafice intuitive.

2.3.2 Cerințe nefuncționale

- Cerințe hardware: telefonul pe care rulează aplicația are nevoie de minim 2GB RAM, procesor de 1 GHz și 2GB spațiu disponibil pe disc pentru a funcționa corespunzător.
- Cerințe software: aplicația necesită Android 9 (Pie) sau o versiune ulterioară ca sistem de operare. De asemenea, este nevoie de o conexiune stabilă la Internet.
- Cerințe de performanță: aplicația trebuie să răspundă rapid la comenzi și să actualizeze datele în timp real.

3 ABORDĂRI EXISTENTE

Aplicația "Task Planner" a fost concepută pornind de la două surse importante de inspirație: ideile potențialilor clienți și cele mai populare abordări existente pe piață.

Mai precis, în urma unor interviuri cu potențialii utilizatori ai produsului, au fost determinate cerințele cheie ale aplicației, care se remarcă printr-un design minimal și funcționalități simple (vezi cerințele funcționale de la capitolul 2). În acest sens, clientul și-a dorit o modalitate extrem de facilă pentru a-și gestiona rapid programările de orice tip, atât pentru afacerea proprie, cât și pentru viața personală.

Un al doilea punct de vedere a fost preluat în urma analizei soluțiilor existente deja pe piață. O astfel de analiză pornește de la un clasament apărut într-un articol al revistei Forbes [4], prin care sunt evidențiate cele mai apreciate aplicații folosite pentru organizarea timpului în cadrul unei afaceri de dimensiuni relativ mici, și anume: Square Appointments și Setmore. În consecință, aplicația Square Appointments este nominalizată ca fiind cea mai bună în general, iar Setmore este desemnată cea mai bună aplicație pentru programări nelimitate. Acestea sunt aplicații complexe și sunt menite să ofere un pachet cât mai complet pentru administrarea relației afacere - clienți.

Pentru a depista motivele pentru care aplicația "Task Planner" ar putea fi preferată de un anumit grup de utilizatori, voi începe prin a descrie caracteristicile esențiale ale produselor menționate anterior, precum și anumite limitări observate.

3.1 Square Appointments

Square Appointments este o aplicație mobilă pentru programări, care oferă un sistem complet pentru vânzări și rezervări și este destinată afacerilor pentru prestare de servicii. [5]

Fiind o aplicație complexă, Square Appointments dispune de numeroase funcționalități, pe lângă rolul principal dedicat stabilirii de programări simplu și rapid.

O primă funcționalitate utilă este reprezentată de controlul asupra calendarului, prestatorul de servicii având posibilitatea de a exclude anumite intervale de timp pentru viitoarele programări. Similar, utilizatorul are libertatea de a controla aspectul paginii de rezervare care ajunge să fie folosită de către clienți.

O altă caracteristică importantă o constituie integrarea unui sistem de plată securizat direct în aplicație. În acest fel, furnizorul de servicii nu are nevoie de instrumente separate pentru tranzacțiile bancare din partea clienților.

Tot din zona plăților electronice, putem remarca o funcționalitate cu adevărat valoroasă ce asigură respectarea programărilor, și anume perceperea unei taxe pentru clienți în cazul anulării și neprezentării. Această taxă este preluată garantat în astfel de cazuri datorită mecanismului prin care aplicația stochează, pentru o durată limitată de timp, cardul clientului.

În plus, Square Appointments pune la dispoziție sincronizarea cu Google Calendar într-o singură direcție. Cu alte cuvinte, aplicația va prelua programările din Google Calendar, fără să modifice conținutul aplicației sursă. Mecanismul enunțat realizează astfel interconectarea sferei personale cu cea profesională și asigură o separare între aceste două lumi prin excluderea vieții profesionale din Google Calendar.

Pe lângă asta, aplicația permite efectuarea de rezervări în mod direct prin intermediul rețelelor sociale: Facebook și Instagram. Mai mult, Square Appointments oferă integrarea cu 20 de aplicații auxiliare cu scopul de a-și extinde funcționalitatea, și anume: aplicații specifice pentru marketing, contabilitate, întâlniri video, gestionarea timpului, și altele.

Nu în cele din urmă, aplicația se descurcă excepțional la transmiterea automată de mesaje de confirmare și reamintire prin SMS sau e-mail, atât personalului, cât și clienților, asigurând astfel buna desfășurare a întregului proces. În mod similar, se pot trimite mesaje personalizate legate de oferte și promoții către clienți direct din aplicație.

Cu toate că Square Appointments posedă o multitudine de funcții adecvate pentru afacerile de prestare de servicii, există și câteva limitări ale acestei aplicații.

Mai precis, aplicația impune folosirea sistemului de procesare a plăților Square, sistem care prevede taxe fixe per tranzacție. Datorită acestei flexibilități limitate în ceea ce privește procesarea plăților, clientul ajunge să plătească tarife mai mari pe termen lung.

Totodată, un alt dezavantaj îl constituie întocmai integrarea platformei Square Appointments cu alte aplicații externe, enumerată ca beneficiu mai sus, întrucât unele dintre aplicațiile respective pot fi necunoscute sau nepreferate de către utilizator.

De asemenea, utilizatorii neexperimentați găsesc interfața grafică destul de dificilă și complexă, în timp ce utilizatorii experimentați sunt nemulțumiți de informațiile minimale rezultate în urma analizei de date.

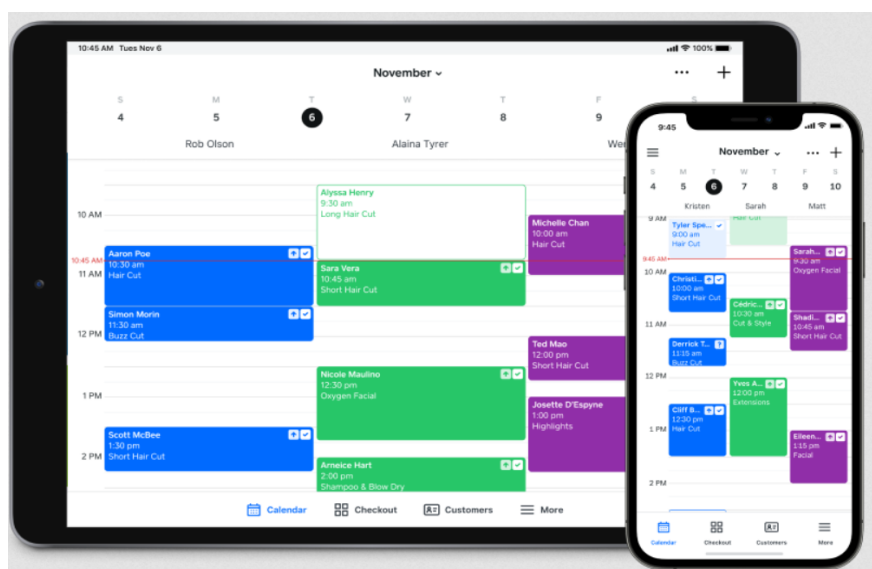


Figura 1: Square Appointments [6]

3.2 Setmore

Setmore este un software pentru programări, disponibil pentru platformele Desktop și Mobile, care permite o relaționare simplă și rapidă între întreprinderi și clienți, deoarece acoperă o gamă completă de funcționalități necesare micilor întreprinderi specializate în prestarea de servicii. [7]

În primul rând, Setmore permite personalizarea aspectelor ce țin de procesul de rezervare, precum: definirea propriilor politici de anulare a programărilor, setarea anumitor intervale de timp între programări, dar și crearea unei pagini personalizate pentru rezervări.

În al doilea rând, aplicația realizează integrarea cu platforme deja cunoscute pentru procesarea securizată a plăților electronice: Square, Stripe și PayPal, oferind astfel diversitate în alegerea platformei preferate de fiecare utilizator în parte.

În plus, Setmore se poate sincroniza cu alte aplicații de timp calendar (Google Calendar, Office365 Calendar) într-un mod bidirecțional și acceptă un număr nelimitat de programări. Astfel, modificările făcute în Setmore vor fi reflectate automat în calendarele externe și invers. În acest sens, utilizatorul poate să își gestioneze mult mai ușor domeniile personal și profesional.

Totodată, aplicația oferă o bună integrare cu platformele de socializare, Instagram și Facebook, pentru extinderea punctelor de acces în aplicație, respectiv platformele pentru întâlniri video, Zoom, Microsoft Teams, Google Meet și Teleport.video, cu scopul de a permite și servicii online, nu doar în persoană. De fapt, în cadrul aplicației sunt incorporate mult mai multe instrumente externe, printre care amintim platformele dedicate pentru business, Slack și Gmail.

De asemenea, Setmore reușește să mențină o experiență plăcută atât pentru clienți, cât și pentru angajați, datorită gamei variate de mesaje care se trimit automat în diferite momente cheie ale întregului proces. Aceste mesaje pot fi trimise pe e-mail sau prin SMS și pot fi de multe tipuri: mesaje de confirmare, de amintire, mesaje de întârziere, notificări interne, mesaje pentru anulări și reprogramări, dar și mesaje de mulțumire și promovare.

În altă ordine de idei, afacerile cu mulți angajați pot folosi Setmore pentru a atribui programările membrilor echipei, îmbunătățind astfel gestionarea resurselor umane și planificarea întâlnirilor. Mai mult, utilizatorii pot vizualiza performanța după fiecare sesiune și activitatea clienților prin intermediul rapoartelor și analizelor realizate de aplicație.

În ciuda faptului că aplicația Setmore prezintă numeroase beneficii, putem enumera și câteva neajunsuri.

Prin urmare, se pot preciza probleme ocazionale în ceea ce privește sincronizarea informațiilor cu alte calendare, dar și un anumit grad de dificultate și dezordine legat de interfața grafică.

Pe de altă parte, unii utilizatori consideră că mecanismul pentru raportare și analiză al aplicației nu conferă informații suficient de detaliate, în comparație cu alte instrumente existente pe piață.

Mai mult, Setmore nu oferă o prea mare flexibilitate cu privire la compatibilitatea cu diferite aplicații de afaceri.

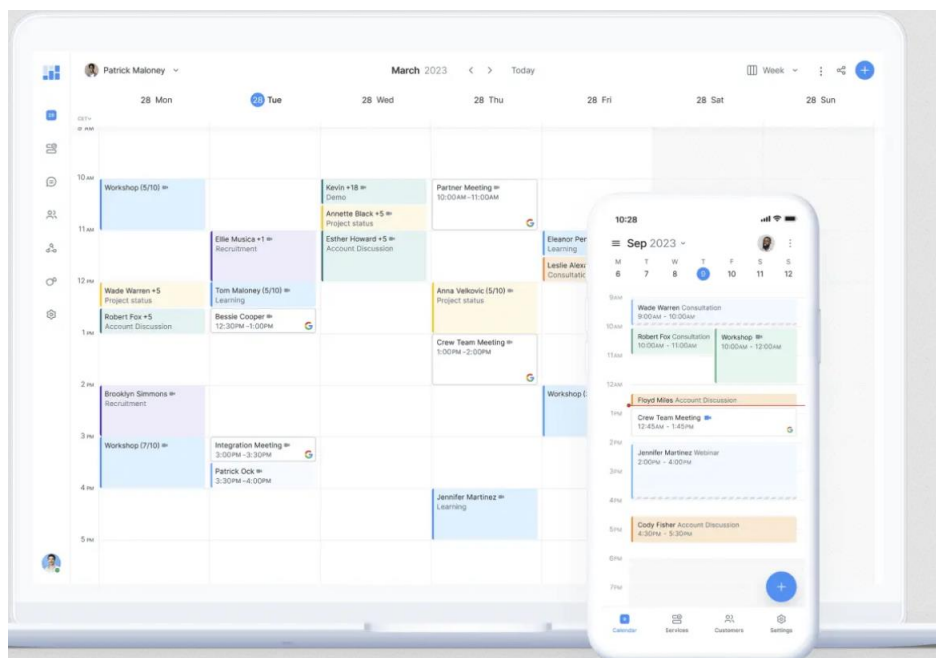


Figura 2: Setmore [8]

3.3 Discuție comparativă

Multitudinea de funcționalități existente în această industrie competitivă poate părea copleșitoare la o primă vedere, dar limitările descoperite au generat câteva oportunități pentru care aplicația "Task Planner" să fie preferată de anumiți utilizatori.

Drept urmare, "Task Planner" îndeplinește funcționalitățile esențiale care contează cel mai mult pentru utilizatori, păstrează un design cât mai intuitiv, ușor de utilizat chiar și pentru începători și oferă opțiuni de personalizare și flexibilitate în ceea ce privește tipurile de evenimente din viața utilizatorului. Cum aplicația este una generică, aceasta poate fi folosită și adaptată unui număr mare de afaceri și cât mai diverse.

4 SOLUȚIA PROPUȘĂ

4.1 Descrierea soluției

Aplicația propusă rezolvă problema gestionării eficiente a programului de lucru pe care clientul îl are. Aceasta își propune să îi ofere o modalitate prin care să își poată organiza toate aspectele legate de afacerea sa, oferindu-i un mediu simplu și intuitiv.

„Task Planner” constă într-o aplicație mobilă bazată pe un calendar principal care oferă posibilitatea utilizatorului de a vizualiza evenimentele în patru moduri diferite: lună, săptămână, trei zile și o zi. Cu ajutorul acestor patru moduri de vizualizare se pot adăuga, șterge sau modifica evenimente din viața profesională a clientului, dar și personală. Fiecare eveniment are asociat un serviciu cu un aspect generic, care poate fi creat de către utilizator, respectând următoarele criterii obligatorii: nume, durată, preț și o culoare specifică. Tot în cadrul unui eveniment este inclus numele și numărul de telefon al persoanei care beneficiază de serviciul asociat, o zonă unde se poate adăuga o mică descriere, cât și opțiunea de a trimite mesaje de reminder persoanei respective. În acest fel utilizatorul se poate asigura că beneficiarii serviciilor sale nu vor uita de programarea făcută, reducând riscul de neprezentare.

O altă secțiune importantă din cadrul aplicației este secțiunea de statistici despre clienți. Există un tabel destinat totalului de plăți făcute de clienți de-a lungul timpului, cât și un tabel în care sunt trecuți clienții care au depășit un prag de număr de zile de la ultima programare. Cu ajutorul acestor statistici utilizatorul poate analiza comportamentul clienților, pe baza căruia își poate îmbunătăți calitatea serviciilor.

Pentru a oferi mai mult control în ceea ce privește combinarea cu viața personală a utilizatorului, aplicația oferă posibilitatea de a activa sau dezactiva sincronizarea cu Google Calendar. În acest fel, „Task Planner” poate fi folosit într-un mod strict profesional. O altă funcționalitate făcută în scopul creșterii nivelului de control asupra interacțiunii cu beneficiarii serviciilor este opțiunea de a formata mesajul de reminder care poate fi trimis către aceștia.

Un ultim feature adus utilizatorului este opțiunea de a își promova serviciile în rândul bazei de clienți acumulate pe parcursul timpului. Posibilitatea de a menține un contact constant cu clienții existenți prin mesaje promoționale ajută utilizatorul să consolideze loialitatea acestora. De asemenea, utilizatorul poate anunța noi servicii sau evenimente, aspect ce face această funcționalitate un înlocuitor minimal pentru marketing, potrivit unei afaceri de dimensiuni reduse.

4.2 Schema bloc

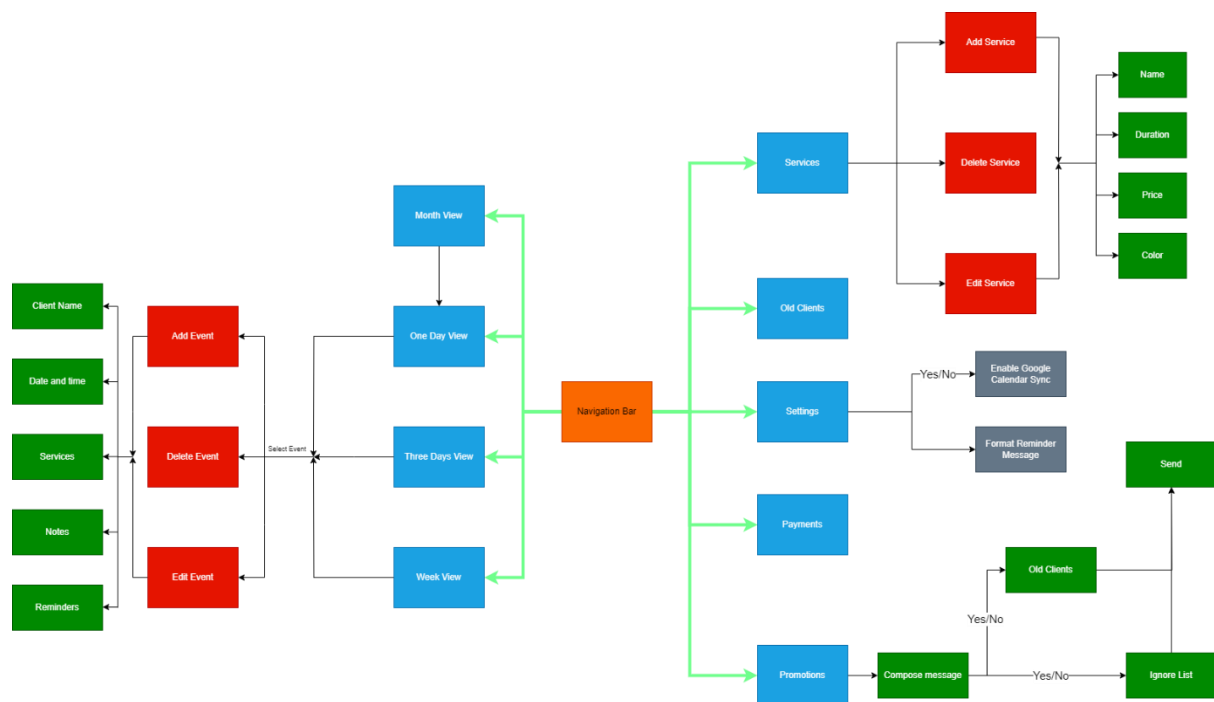


Figura 3: Schema bloc

După cum se poate observa în diagrama de mai sus, aplicația este împărțită în două părți principale: vizualizarea și editarea programărilor utilizatorului și gestionarea aspectelor tehnice. Organizarea aplicației într-un mod cât mai intuitiv a fost una dintre necesitățile principale, acest lucru oferind posibilitatea utilizatorului de a-și gestiona timpul într-un mod cât mai rapid și eficient.

În cele ce urmează voi face o prezentare detaliată a fiecărei componente, punând accent pe funcționalitățile acestora și conexiunile între ele.

4.2.1 Navigation Bar

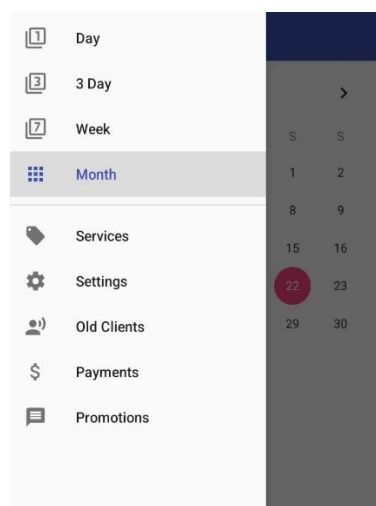


Figura 4: Bară de navigare

Am ales să folosesc o bară de navigare ca metodă de comutare între componentele aplicației pentru că oferă o imagine clară asupra celor două secțiuni în care este împărțită aceasta.

În partea superioară este prezentă secțiunea pentru vizualizarea evenimentelor programate, care pot fi organizate în funcție de perioada de timp în cele patru moduri prezente. Am ales ca în fiecare dintre aceste patru componente să plasez un buton pentru expandarea sidebar-ului, în partea de sus a ecranului, astfel făcând comutarea între ele cât mai facilă.

A doua secțiune a barei de navigare este despărțită de o linie orizontală pentru a face o delimitare clară a aplicației în două părți. Aceasta este destinată gestionării aspectelor tehnice ce țin de analiza statisticilor despre clienți, modificarea setărilor și a listei de servicii, cât și trimiterea de mesaje promoționale. Pentru componentele din această secțiune am ales să plasez în partea de sus a ecranului un buton de „Înapoi” care să permită accesul rapid la vizualizarea principală în format lunar.

4.2.2 Vizualizare în format lunar

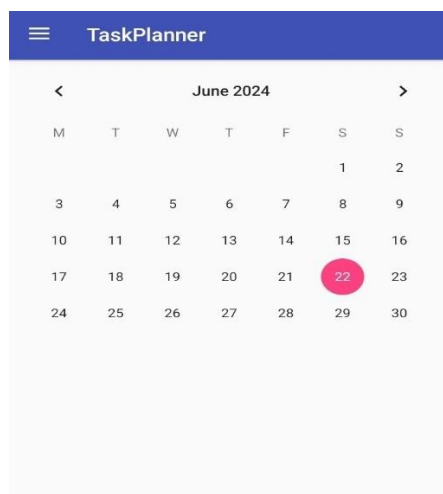


Figura 5: Calendar principal

Meniul principal al aplicației este vizualizarea în format lunar, astfel la deschiderea aplicației utilizatorul are o privire de ansamblu asupra lunii curente și poate selecta ziua unde vrea să își organizeze agenda. Fiecare zi este interactivă, iar prin apăsarea ei se deschide componenta dedicată vizualizării în format zilnic.

De asemenea, este posibilă navigarea între lunile trecute sau viitoare pentru a examina programările deja efectuate sau pentru a planifica noi evenimente din timp.

4.2.3 Vizualizare în format săptămânal

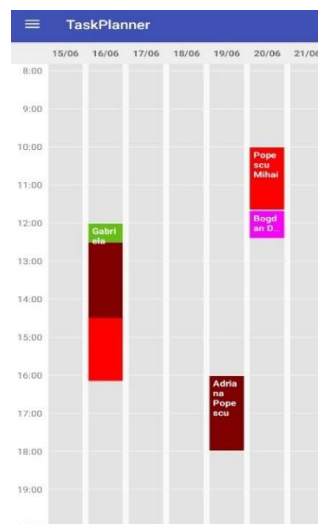


Figura 6: Vizualizare pe săptămână

În modul de vizualizare săptămânal sunt prezente și evenimentele programate, cu opțiunea de a adăuga noi evenimente, de a le șterge sau edita pe cele curente. Acest format a fost gândit pentru a oferi utilizatorului posibilitatea să își vizualizeze întreaga săptămână într-un mod cât mai clar și eficient, permițându-i să își organizeze timpul pentru săptămâna viitoare într-o manieră cât mai structurată.

4.2.4 Vizualizare în format de trei zile

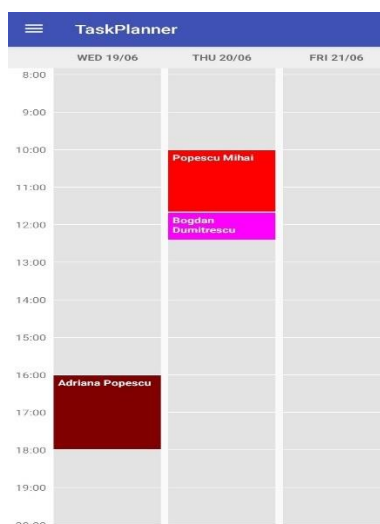


Figura 7: Vizualizare pe trei zile

Acest format a fost creat pentru a aduce un plus de flexibilitate în modul de afișare a agendei utilizatorului. Este ideal pentru cei care preferă o vizualizare detaliată a programului pe o perioadă de timp scurtă.

4.2.5 Vizualizare in format zilnic

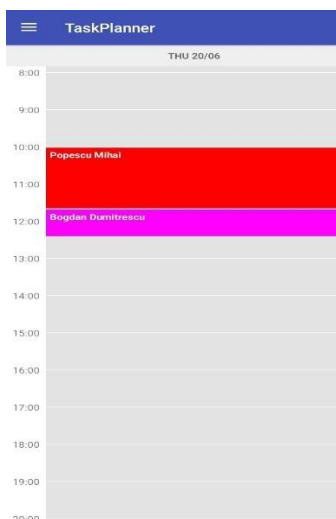


Figura 8: Vizualizare pe o zi

În formatul zilnic, evenimentele sunt afișate mai clar și inteligibil. Intervalele orare între care se desfășoară evenimentele sunt mai ușor de urmărit, lucru ce oferă utilizatorului o imagine de ansamblu mult mai detaliată asupra zilei curente.

Se poate observa că există o limită a orelor între care se pot adăuga evenimente, limită care reflectă programul de lucru al utilizatorului. De asemenea, fiecare eveniment este marcat cu culoarea specifică a serviciului asociat pentru a îmbunătăți vizibilitatea și claritatea agendei utilizatorului. Astfel, acesta poate identifica rapid diversele activități planificate pentru ziua respectivă.

Similar celorlalte moduri de vizualizare, navigarea între zilele adiacente este posibilă printr-un simplu gest de glisare la stânga sau la dreapta.

4.2.6 Gestionarea evenimentelor

Evenimentele reprezintă obiectul principal din aplicația „Task Planner”. Acestea populează calendarul utilizatorului și îl ajută să își organizeze programul.

Fiecare mod de vizualizare, cu excepția celui lunar, este structurat în intervale de o oră interactive, care oferă două funcționalități distincte la apăsare: adăugarea de noi evenimente și editarea sau ștergerea unui eveniment existent.

În cazul în care un interval orar este gol, utilizatorul îl poate selecta și o componentă pentru completarea datelor despre eveniment va fi afișată.

Figura 9: Ecran adăugare eveniment

În cadrul acestei componente există mai multe câmpuri care trebuie completate, după cum urmează:

- **Numele clientului:** este un input de text care odată cu completarea a două caractere afișează o listă cu opțiunile posibile de completare. În cazul în care utilizatorul vrea să vadă lista completă de persoane, poate apăsa încă o dată pe căsuța de text și o listă cu toate persoanele din agenda sa de contacte va fi afișată. Prezența clientului în lista din agenda de contacte este obligatorie, deoarece numărul de telefon este un parametru necesar funcționalității de mesaje de reminder. Pentru a trata cazul în care utilizatorul introduce un nume care nu se află în agenda sa de contacte, am ales să afișez un mesaj informativ în acest sens.

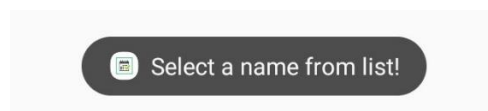


Figura 10: Eroare nume

- **Data și ora:** Aceste câmpuri sunt completate automat în urma selectării unui interval orar din calendar, însă am ales să dau utilizatorului opțiunea de a le modifica. Pentru selectarea valorilor din aceste câmpuri am adăugat câteva selectoare simple și intuitive pentru a da un plus de eficiență întregului proces de creare a unui eveniment.

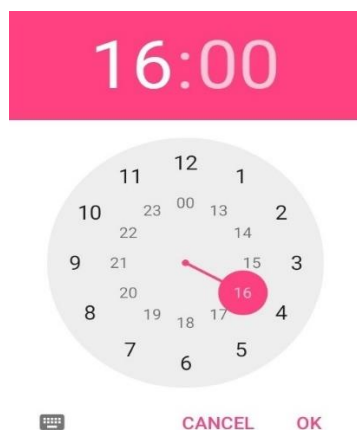


Figura 11: Selector oră

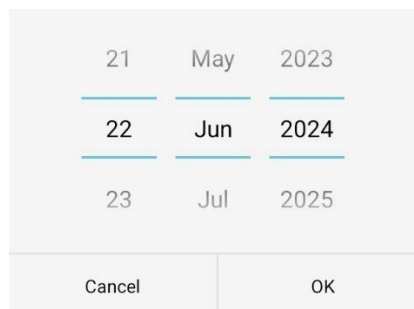


Figura 12 selector dată

În cazul în care ora de început a evenimentului nu se încadrează în intervalul orar al programului utilizatorului, un mesaj informativ în acest sens va fi afișat.

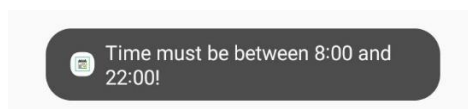


Figura 13: Eroare interval orar

- **Serviciul:** Pentru adăugarea serviciului de care va beneficia clientul am ales să folosesc o listă simplă de unde utilizatorul poate selecta unul sau mai multe servicii.

Add Services

- ☒ Serviciu 1
- ☒ Serviciu 2
- ☐ Serviciu 3

CANCEL OK

Figura 14: Dialog selectare servicii

În momentul în care acestea sunt adăugate, va fi calculat prețul total, iar ora la care se va încheia evenimentul va fi actualizată. Cu toate acestea, am ales să dau opțiunea utilizatorului de a modifica prețul total, pentru a-i oferi cât mai mult control în folosirea aplicației.

- **Notițe:** Această secțiune este folosită pentru a adăuga anumite comentarii sau detalii specifice programării.
- **Mesaje de reminder:** Formatul acestui mesaj este prestabilit de către utilizator și este trimis clientului care beneficiază de serviciul oferit. Am ales

să implementez o listă cu alegere multiplă de unde utilizatorul poate alege să trimită mesaj fie în momentul inițializării programării, fie cu câteva ore înainte de eveniment. Mesajele sunt trimise de pe numărul de telefon al utilizatorului și necesită integrarea cu un furnizor de servicii de telefonie mobilă terț pentru a asigura livrarea acestora.

Pentru a exemplifica cât mai clar procesul de gestionare a evenimentelor, am realizat o diagramă de secvențe:

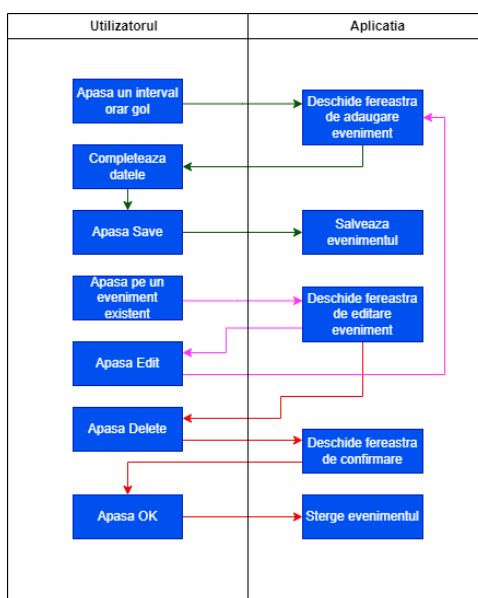


Figura 15: Diagramă de secvențe pentru evenimente

4.2.7 Servicii

Această componentă oferă utilizatorului o interfață completă pentru gestionarea serviciilor sale. Acesta poate vizualiza lista de servicii existente, după cum se poate observa în imaginea de mai jos:

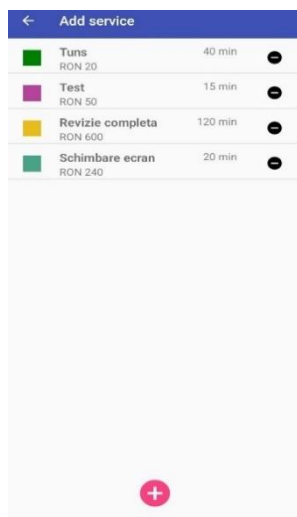


Figura 16: Ecran de gestionare a serviciilor

O altă funcționalitate inclusă în această componentă este adăugarea de servicii noi. Prin apăsarea butonului din partea de jos a ecranului este deschisă o nouă fereastră în care există patru câmpuri de completat: nume, durată (exprimată în minute pentru simplitatea designului), preț și o culoare specifică.

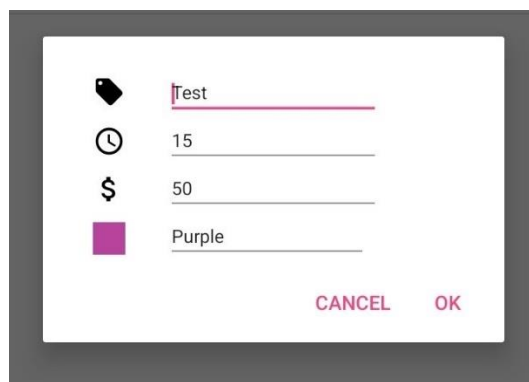
A screenshot of a dialog box for adding a new service. It has a white background with a dark gray border. On the left, there are four icons: a black tag, a clock, a dollar sign, and a purple square. To the right of each icon is a text input field. The first field contains 'Test', the second '15', the third '50', and the fourth 'Purple'. At the bottom right, there are two buttons: 'CANCEL' and 'OK', both in pink text.

Figura 17: Dialog de adăugare a unui serviciu

Am ales să asociez fiecărui serviciu o culoare pentru a se putea diferenția atunci când este adăugat în calendar. Există o listă predefinită de culori contrastante pentru ca utilizatorul să le poată identifica cu ușurință.

De asemenea, fiecare element din listă este interactiv, iar atunci când este apăsat deschide fereastra din imaginea [nr_poză], prezentată anterior. În această fereastră, utilizatorul poate modifica proprietățile serviciului selectat, iar aceste modificări se vor actualiza și în calendarul său de evenimente.

O altă funcționalitate esențială este posibilitatea de a șterge servicii existente. Pentru simplitate, am ales să plasez în dreptul fiecărui serviciu un buton de delete, astfel încât utilizatorul să poată elimina cu ușurință serviciile pe care nu le mai prestează. Pentru a rezolva problema apăsărilor accidentale pe buton, am adăugat un dialog de confirmare pentru ștergerea serviciului înainte ca acesta să fie șters definitiv. În acest fel, este oferită o gestionare mai sigură și controlată a listei de servicii.

Remove service?

CANCEL OK

Figura 18: Dialog de confirmare ștergere serviciu

Pentru a exemplifica cât mai clar procesul de gestionare a serviciilor, am realizat o diagramă de secvențe:

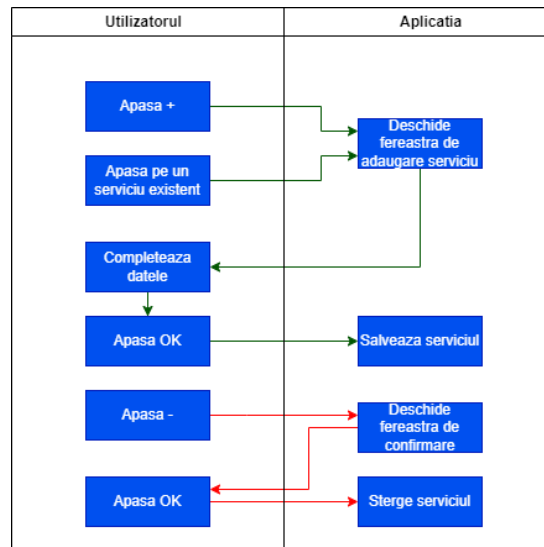


Figura 19: Diagramă de secvențe pentru servicii

4.2.8 Setări

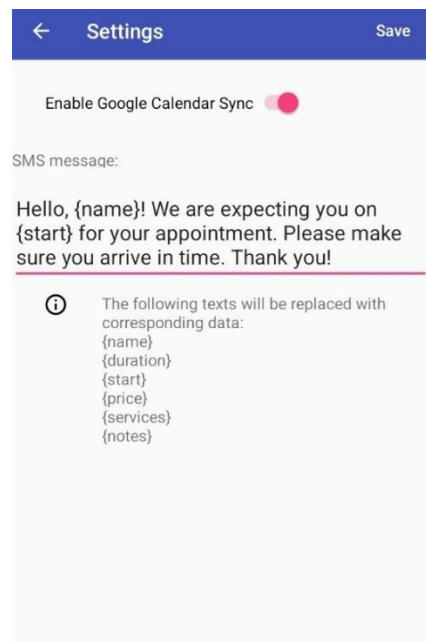


Figura 20: Ecran pentru setări

Componenta de setări oferă posibilitatea utilizatorului de a personaliza funcționalitățile aplicației în așa fel încât să se potrivească nevoilor proprii.

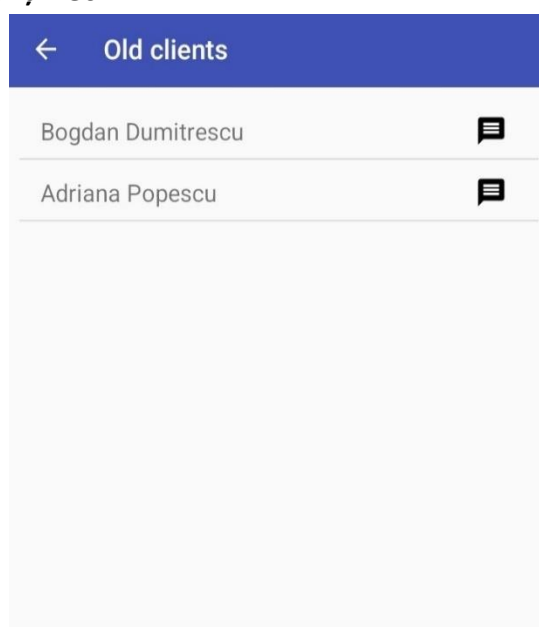
Una din funcționalitățile care este destul de posibil ca utilizatorii să nu o folosească este sincronizarea cu Google Calendar. Deși aceasta poate combina agenda profesională cu cea personală a utilizatorului, este necesar să existe opțiunea de a nu face acest lucru. De aceea, am adăugat un switch prin care să poată fi dezactivată această funcționalitate.

Un alt aspect care este lăsat la latitudinea utilizatorului este forma mesajului de reminder care se trimite către clienți în momentul în care este făcută o programare. În funcție de tipul de afacere pe care utilizatorul îl are sau de tipul comunității căreia se adresează, acesta

poate alege să folosească un limbaj formal sau informal, o descriere detaliată a modului de operare al acestuia sau anumite detalii despre situații neprevăzute care necesită ajustări în modul de lucru.

În compunerea mesajului, utilizatorul are opțiunea de a introduce cuvinte cheie, de exemplu {name}. Înainte ca un mesaj de reminder să fie trimis, acesta trece printr-un proces de prelucrare unde toate cuvintele cheie sunt extrase din datele despre evenimentul aferent programării și înlocuite în template-ul inițial conceput de utilizator, pentru a obține un mesaj final personalizat pentru fiecare client.

4.2.9 Tabel pentru clienți vechi



← Old clients	
Bogdan Dumitrescu	💬
Adriana Popescu	💬

Figura 21: Ecran pentru clienți vechi

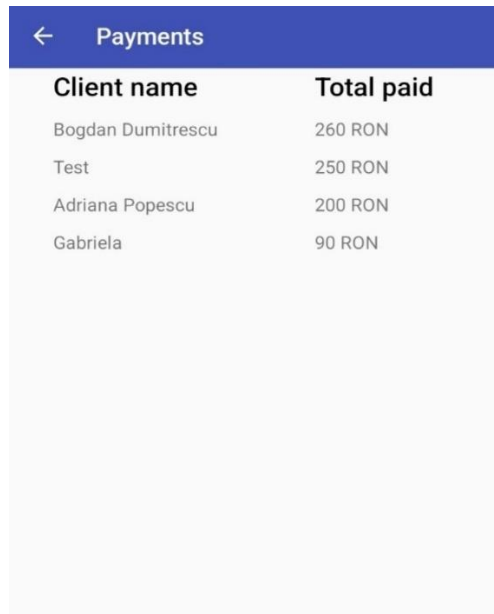
În acest tabel sunt afișați clienții care nu au mai avut o programare de o perioadă îndelungată de timp. Popularea acestuia se face în momentul în care este accesată componenta în cadrul aplicației, iar informațiile legate de ultima programare sunt extrase din baza de date. Aceasta conține o tabelă dedicată clienților, iar una dintre coloanele acesteia reprezintă data când a avut loc ultima programare. Având această configurație, am putut realiza ușor un query pentru a extrage datele necesare pentru această componentă:

```
@Query("SELECT * FROM client WHERE lastEventDate <= :date")
public abstract List<Client> findClientsOlderThan(Calendar date);
```

Figura 22: Query pentru selectarea clienților vechi

Cu ajutorul acestui tabel, utilizatorul poate avea o imagine clară și organizată asupra clienților inactivi, permițându-i să ia măsuri în momentul în care acest lucru devine o problemă.

4.2.10 Tabel pentru totalul plăților



Client name	Total paid
Bogdan Dumitrescu	260 RON
Test	250 RON
Adriana Popescu	200 RON
Gabriela	90 RON

Figura 23: Ecran pentru totalul plăților

În acest tabel sunt afișați toți clienții și suma totală de bani pe care au plătit-o de-a lungul timpului. Această componentă are ca scop expunerea activității financiare a fiecărui client pe care utilizatorul îl are pentru ca acesta să aibă o vedere amplă asupra clienților fideli.

La fel ca în cazul tabelului pentru clienți vechi, popularea acestui tabel se face în momentul în care componenta este accesată, iar informațiile legate de totalul de plăți făcute de un anumit client cât și numele acestuia sunt extrase dintr-o tabelă de evenimente așa cum se poate vedea în query-ul de mai jos:

```
@Query("SELECT client.name as name, SUM(event.price) as totalPaid " +  
        "FROM client " +  
        "INNER JOIN event ON client.id = event.clientId " +  
        "WHERE event.endDate <= :currentDate " +  
        "GROUP BY client.name " +  
        "ORDER BY SUM(event.price) DESC")  
public List<ClientTotal> getTotalForEveryClient(Calendar currentDate);
```

Figura 24: Query pentru extragerea totalului plăților facute de clienți

4.2.11 Promoții

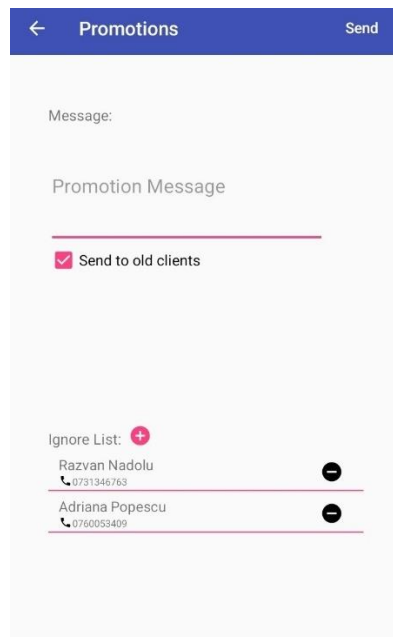


Figura 25: Ecran pentru promoții

Componenta de promoții a fost concepută pentru a stabili un mod facil de comunicare între utilizator și clienții săi. Aceasta oferă posibilitatea utilizatorului să trimită mesaje către baza sa de clienți, mesaje care sunt compuse în totalitate de către el. Acest lucru aduce un plus de genericitate aplicației, astfel această funcționalitate poate fi folosită în diferite moduri.

Cu toate acestea, scopul principal pentru care am dezvoltat această componentă a fost pentru a permite utilizatorului să gestioneze mici campanii de marketing. Din acest motiv, am ales să adaug opțiunea de a trimite mesajul doar clienților care nu au mai avut o programare de o perioadă îndelungată de timp, deoarece un mesaj promoțional poate să reînnoiască interesul și să îi încurajeze să revină.

În plus, am inclus și o listă de ignorare, unde utilizatorul poate alege clienții pe care dorește să îi excludă din campaniile de promovare. În acest fel, campaniile de marketing pot fi direcționate către publicul țintă relevant, având un impact și o eficiență mai mare.

5 DETALII DE IMPLEMENTARE

5.1 Tehnologii utilizate

Unul dintre criteriile principale de care am ținut cont în momentul alegerii platformei pe care am dezvoltat aplicația „Task Planner” a fost nivelul de răspândire în rândul clienților țintă pentru care este dedicată această aplicație.

5.1.1 Android

Având în vedere că Android este cea mai mare platformă mobilă la nivel mondial, fiind prezent pe aproximativ 75% din toate dispozitivele mobile de pe piață [9], alegerea acestuia a fost evidentă. De asemenea, Android este utilizat pe o gamă variată de dispozitive, de la smartphone-uri de buget până la variante premium. Acest lucru permite aplicației mele să fie disponibilă unei game largi de utilizatori, lucru extrem de important având în vedere publicul variat căruia i se adresează.

Un alt aspect important care m-a determinat să aleg Android a fost ecosistemul foarte complex de dezvoltare oferit, anume Android Studio, dar și accesul la o multitudine de API-uri oferite de Google care au simplificat implementarea unor funcționalități cheie din cadrul aplicației, precum: sincronizarea cu Google Calendar și serviciul de mesaje de reminder.

5.1.2 Java

Atunci când Java fost inventat în anul 1991 de către James Gosling de la Sun Microsystems (achiziționat ulterior de Oracle), scopul său principal a fost rezumat prin expresia „write once, run anywhere”[10]. Acesta a devenit un limbajul de programare orientat pe obiect principal în rândul dezvoltatorilor software și este utilizat pe scară largă și în prezent. Acest lucru se datorează eficienței sale și portabilității pe diverse tipuri de dispozitive.

Am ales Java 8 ca limbaj de programare pentru dezvoltarea aplicației mele Android, deoarece aduce un echilibru optim între funcționalitățile incluse și compatibilitatea extinsă.

Java 8 este bine integrată cu ecosistemul Android și compatibilă cu ultimele versiuni de SDK ale acestuia. Astfel, am putut să folosesc compileSDK 34 și minSDK 28. În acest fel, aplicația mea poate rula eficient atât pe cele mai noi dispozitive, cât și pe cele mai vechi.

5.2 Platforma pentru dezvoltare

Am ales Android Studio ca platformă de dezvoltare pentru aplicația mea datorită integrării sale oficiale și complete cu ecosistemul Android [11]. Fiind IDE-ul dezvoltat de Google, acesta oferă suport nativ pentru SDK-ul Android și biblioteca AndroidX, care aduce o modularitate extinsă, compatibilitate îmbunătățită și acces la API-uri moderne.

Un alt avantaj important este Layout Editor, un instrument folosit pentru crearea rapidă și intuitivă de interfețe grafice. Cu ajutorul acestei funcționalități, dezvoltarea ecranelor din aplicație a fost foarte facilă, având la dispoziție o interfață „drag-and-drop” pentru poziționarea elementelor, dar și o mulțime de iconițe și resurse predefinite.

Android Studio include și un debugger integrat care face procesul de identificare și rezolvare a problemelor din cod foarte facil. Acest debugger oferă posibilitatea de a examina starea aplicației în timpul rulării prin setarea de breakpoint-uri în zonele de cod problematice.

O ultimă componentă importantă a Android Studio este Device Manager. Acesta reprezintă o interfață completă cu ajutorul căreia am putut să îmi configurez emulatoare cu versiuni specifice de Android, dar și să pot conecta propriul dispozitiv Android pentru a putea testa aplicația în condiții reale de funcționare.

În faza de planificare a proiectului am luat în considerare și alte platforme de dezvoltare Android prezente pe piață, însă doar două mi-au atras atenția.

- **Unity:** Deși este o platformă dedicată dezvoltării de jocuri 2D sau 3D, aceasta include o gamă largă de funcționalități care pot aduce un plus de eficiență și productivitate, în funcție de tipul de aplicație pe care o poți face [12]. Totuși, pentru aplicația „Task Planner”, partea de design și interactivitate nu a fost esențială, așadar am ales Android Studio pentru capacitatea sa de a dezvolta interfețe grafice eficiente pentru aplicații cu utilitate generală.
- **Eclipse:** Folosit pe scară largă înainte de apariția Android Studio, această platformă a reprezentat modul principal de dezvoltare a aplicațiilor Android [13]. Împreună cu plugin-ul ADT (Android Development Tool) [14], a fost soluția de bază recomandată de Google pentru dezvoltatorii de aplicații Android. Cu toate acestea, în prezent nu mai oferă suport actualizat pentru cele mai noi tehnologii Android, fiind în contrast cu Android Studio, care este IDE-ul dezvoltat de Google.

5.3 Structura proiectului

Pentru dezvoltarea aplicației „Task Planner” am împărțit proiectul în două părți principale:

- Componenta funcțională (backend), care conține codul logic scris în limbajul de programare ales: Java. Această parte cuprinde prelucrarea datelor, comunicarea cu baza de date, cât și efectuarea operațiunilor necesare pentru funcționarea aplicației.
- Interfața de utilizator (frontend), care cuprinde toate elementele vizuale și resursele necesare pentru interfața grafică și interacțiunea cu utilizatorul.

Pentru a oferi o perspectivă mai detaliată asupra structurii proiectului, voi prezenta ierarhia de fișiere din cadrul acestuia, ilustrată în imaginea de mai jos:

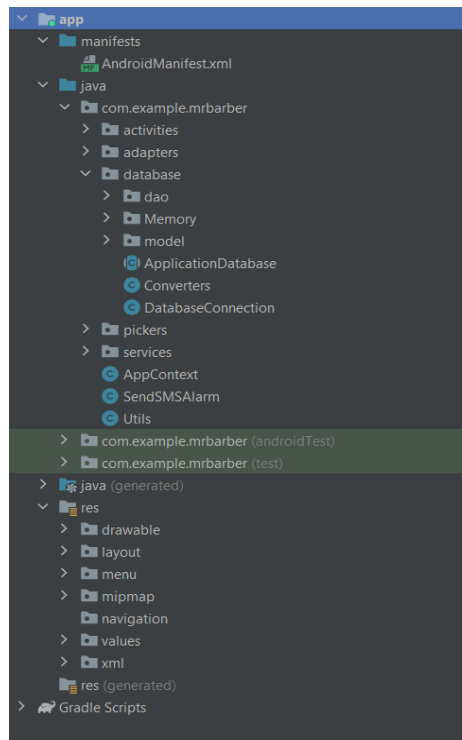


Figura 26: Structura de fișiere

5.3.1 Configurația proiectului

Așa cum am menționat anterior, aplicația „Task Planner” este gândită pentru a putea fi disponibilă unei baze mari de clienți, lucru care necesită o configurare specifică a fișierelor Gradle și AndroidManifest.

Fișierul **Gradle** gestionează sursele de biblioteci externe, dependențele și plugin-ul necesar pentru procesul de compilare al aplicației. Pe lângă SDK-ul Android 34 și versiunea de Java 8, acesta include biblioteci precum AndroidX, Google Play Services, necesară pentru integrarea serviciilor Google, Room Database, care se ocupă de gestionarea eficientă a datelor locale și Alamy [15], biblioteca pentru calendarul principal utilizat pentru afișarea programărilor.

Fișierul **AndroidManifest** definește structura aplicației și declară activitățile principale, care se traduc prin componentele afișate în cadrul interfeței grafice. De asemenea, fișierul specifică permisiunile necesare pe care aplicația trebuie să le obțină de la utilizator pentru a activa toate funcționalitățile disponibile. Printre acestea se enumeră: accesul la internet, accesul la agenda de contacte, permisiunea de a trimite SMS-uri.

5.4 Descrierea modulelor aplicației

În continuare, voi furniza descrieri cu aspect tehnic pentru fiecare componentă și pachet Java relevant.

5.4.1 Pachetul „activities”

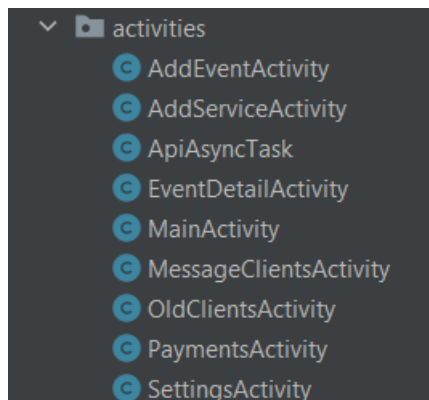


Figura 27: Structura activităților

Din acest pachet fac parte clase Java personalizate care moștenesc clasa **AppCompatActivity** [16] și gestionează logica și comportamentul fiecărui ecran al aplicației.

5.4.2 MainActivity

Cea mai importantă componentă este MainActivity, care este rulată la deschiderea aplicației. Atunci când utilizatorul folosește pentru prima dată aplicația, îi sunt cerute permisiunile menționate mai sus pentru citirea contactelor și trimiterea de mesaje SMS de pe numărul propriu de telefon. Tot în momentul în care acesta creează pentru prima dată un eveniment, este afișat un dialog unde i se cere să selecteze contul în care să fie sincronizat calendarul Google. De asemenea, în această componentă este instanțiată alarma de SMS, care se ocupă de trimiterea mesajelor de reminder periodic.

În plus, această componentă gestionează logica pentru afișarea celor patru moduri de vizualizare a calendarului: lunar, săptămânal, pe trei zile și zilnic și configurează bara de navigare între celelalte componente din aplicație.

Pentru o înțelegere mai ușoară a funcționalităților acestei clase, am optat pentru prezentarea sa sub formă de pseudocod:

```

public class MainActivity extends AppCompatActivity {

    onCreate() {
        super.onCreate();
        checkPermissions(); // Verifică și solicită permisiunile necesare
        configureToolBarAndDrawer(); // Configurează bara de navigare și meniul lateral
        configureCalendarView(); // Setează modul de vizualizare
        configureGoogleCalendarAPI(); // Configurează autentificarea cu Google Calendar API
    }

    onActivityResult() // Gestionează rezultatele specifice:
        // crearea/editarea evenimentelor, contul Google

    checkPermissions() // Verifică permisiunile și, dacă nu sunt acordate, le solicită

    readContacts() // Accesează și stochează contactele în memorie

    startsMSAlarm()
        // Configurează AlarmManager pentru alarmele SMS
        // Se verifica odata la 5 minute daca exista remindere care trebuie trimise

    onNavigationItemSelected(element selectat din bara de navigare)
        // Modifică vizualizarea calendarului și navighează către alte ecrane

    // Sincronizează datele cu Google Calendar
    syncWithGoogleCalendar() {
        if (nu a fost selectat un cont google) {
            chooseAccount(); // Alege contul Google
        } else if (device-ul este online) {
            new ApiAsyncTask(this).execute(); // trimite evenimentul catre Google Calendar
        }
    }
}

```

Figura 28: Pseudocod pentru MainActivity

5.4.3 ApiAsyncTask

Clasa ApiAsyncTask este destinată sincronizării calendarului local cu cel asociat contului Google selectat de utilizator. Trimiterea cererilor către Google pentru actualizarea evenimentelor necesită un timp considerabil de procesare, iar de aceea trebuie făcute asincron, pentru a preveni încetinirea aplicației și pentru a menține un flux de utilizare cursiv al aplicației, fără întreruperi. Această clasă gestionează operațiuni esențiale precum adăugarea, ștergerea și editarea evenimentelor din calendarul Google. [17]

Implementarea acestei logici este relativ simplă, după cum se poate observa în pseudocodul ilustrat mai jos:

```

class ApiAsyncTask extends AsyncTask {

    // Execută trimiterea evenimentului catre Google în fundal
    doInBackground() {
        // Verifică modul evenimentului și alege acțiunea corespunzătoare:
        // Dacă utilizatorul a ales să salveze un nou eveniment
        if (modul de eveniment este "save") {
            insertEventInGoogleCalendar(); // Inserează un eveniment în Google Calendar
        }
        // Dacă utilizatorul a ales să editeze un eveniment existent
        else if (modul de eveniment este "edit") {
            editEventInGoogleCalendar(); // Editează un eveniment existent în Google Calendar
        }
        // Dacă utilizatorul a ales să ștergă un eveniment existent
        else if (modul de eveniment este "remove") {
            removeEventInGoogleCalendar(); // Șterge un eveniment din Google Calendar
        }
    }
}

```

Figura 29: Pseudocod pentru sincronizarea cu Google Calendar

5.4.4 AddEventActivity

Această clasă gestionează completarea datelor necesare pentru programarea unui eveniment. Aceasta include verificarea existenței persoanei care beneficiază de serviciu în agenda de contacte a utilizatorului și asigurarea că intervalul orar selectat respectă programul de lucru.

În plus, clasa aduce toate serviciile definite anterior de utilizator din baza de date, permițându-i să selecteze unul sau mai multe. În funcție de serviciile selectate, clasa calculează automat prețul total și ajustează ora de sfârșit a evenimentului.

AddEventActivity este declanșată din clasa MainActivity atunci când utilizatorul apasă pe un interval orar din calendar. După completarea tuturor informațiilor necesare, evenimentul este salvat în baza de date.

O imagine simplificată a funcționalității acestei clase este prezentată mai jos:

```
class AddEventActivity extends AppCompatActivity {  
  
    // Configurarea activității și UI la creare  
    onCreate() {  
        initializeFields(); // Inițializează câmpurile de dată, oră și completare automată pentru clienți  
        setupServiceDialog(); // Configurează dialogul pentru selectarea serviciilor asociate evenimentului  
        if (modul de eveniment este "edit") {  
            loadEventDetails(); // Încarcă detaliile evenimentului pentru editare  
        }  
    }  
  
    // Salvează evenimentul curent în baza de date  
    saveEvent() {  
        validateClient(); // Validează dacă numele clientului este selectat din lista de contacte  
        setEventDetails(); // Setează detaliile evenimentului (dată, oră, servicii, preț, notițe)  
        if (modul de eveniment este "edit") {  
            updateExistingEvent(); // Actualizează evenimentul existent în baza de date  
        } else {  
            saveNewEvent(); // Salvează un nou eveniment în baza de date  
        }  
        sendInstantReminderIfChecked(); // Trimite un SMS de reamintire instant dacă opțiunea este bifată  
    }  
  
    // Afișează selectorul de oră pentru ora de început și sfârșit  
    showTimePickerFromDialog(view)  
  
    // Afișează selectorul de dată pentru ziua evenimentului  
    showDatePickerDialog()  
}
```

Figura 30: Pseudocod pentru adăugarea unui eveniment

5.4.5 MessageClientsActivity

Clasa MessageClientsActivity este responsabilă pentru gestionarea procesului de trimitere a mesajelor de reminder către clienți. Această funcționalitate se execută periodic, la intervale de timp fixe, extrăgând toate evenimentele relevante din baza de date și verificând dacă este necesar să se trimită un mesaj de reminder către client. [18]

Un aspect esențial al acestei clase este personalizarea mesajului pentru fiecare client. Acest lucru este posibil utilizând un șablon definit de utilizator în meniul de setări. Acest șablon este stocat în baza de date și conține câmpuri care pot fi înlocuite cu informații specifice unei programări, cum ar fi numele clientului, ora de început și serviciile incluse.

Pentru a genera aceste mesaje personalizate, am implementat o metodă care formatează mesajul SMS. Această metodă preia șablonul din baza de date și valorile specifice fiecărui eveniment, compunând astfel un mesaj coerent și adaptat fiecărui client.

O reprezentare simplificată a acestei clase este ilustrată mai jos:

```
class SendSMSAlarm extends BroadcastReceiver {  
    // Metoda apelată atunci când se primește semnalul de alarmă pentru trimiterea SMS-urilor  
    onReceive() {  
        initializeSmsManager(); // Inițializează managerul SMS pentru trimiterea mesajelor  
  
        // Setează intervalul de timp pentru evenimentele care vor fi verificate  
        setTimeRangeForEvents();  
  
        // Obține lista de evenimente din baza de date în intervalul de timp specificat  
        eventList = getEventsWithinTimeRange();  
  
        for (fiecare eveniment din eventList) {  
            reminderList = parseRemindersFromEvent(event); // Extrage lista de reamintiri din eveniment  
  
            currentDate = getCurrentDate(); // Obține data și ora curentă  
  
            for (fiecare reminder al programarii) {  
                startDate = calculateReminderTime(event, reminder); // Calculează ora reamintirii pentru eveniment  
  
                if (isCurrentTimeNearReminder(startDate, currentDate)) {  
                    smsText = formatSMS(event, context); // Formatează textul SMS-ului folosind  
                    sendReminderSMS(smsText, event.getClientPhoneNumber()); // Trimite SMS-ul de reamintire  
                    removeReminderFromList(reminder); // Elimină reamintirea care tocmai a fost trimisă  
                }  
            }  
  
            // Actualizează evenimentul în baza de date cu lista de reamintiri modificată  
            updateEventWithNewReminders(event, reminderList);  
        }  
    }  
}
```

Figura 31: Pseudocod pentru alarma de SMS-uri

5.4.6 Resources

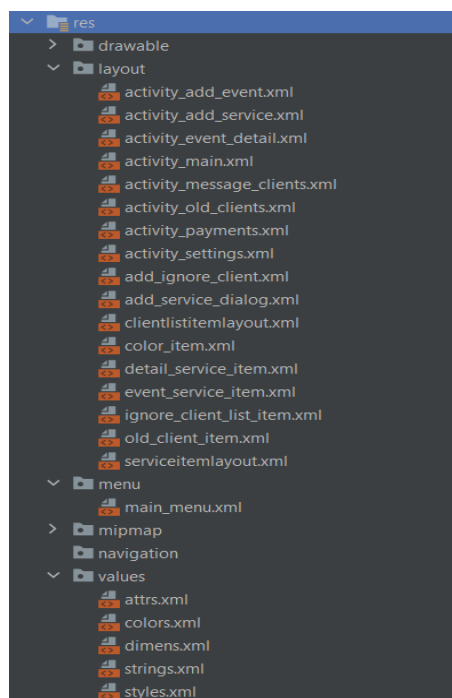


Figura 32: Structura directorului resources

Componentele din Resources reprezintă frontend-ul în dezvoltarea aplicațiilor Android. Aceste resurse sunt esențiale pentru definirea aspectului și comportamentului vizual al aplicației.

Cea mai importantă componentă o reprezintă fișierele de tip layout, care definesc structura și aranjamentul vizual al ecranelor aplicației prin XML. Layout-urile stabilesc modul în care sunt aranjate elementele în interfața grafică, cum ar fi butoanele, textele și imaginile.

Fișierele care au în nume prefixul „activity” reprezintă echivalentul claselor din backend care definesc funcționalitatea acestora. Clasele interacționează cu elementele din layout folosind metoda „findViewById”, care permite obținerea și manipularea componentelor și a datelor prezente în layout. În acest fel, clasele „Activity” și layout-urile colaborează pentru a oferi utilizatorului o experiență coerentă și interactivă.

De asemenea, există layout-uri secundare, care fac parte dintr-un ecran mai complex, precum: **add_service_dialog**, un dialog pentru adăugarea unui serviciu, **color_item**, un element pentru afișarea culorilor, și **old_client_item**, un element listat pentru clienți vechi.

5.5 Baza de date

Pentru implementarea bazei de date a aplicației mele, am ales să utilizez biblioteca **Room Database** [19]. Aceasta este o bibliotecă inclusă în Android, care simplifică gestionarea bazelor de date locale SQLite [20].

Room se remarcă prin faptul că oferă o abstractizare de nivel înalt în interacțiunea cu baza de date, permițând dezvoltatorilor să folosească obiecte pentru a defini structura unui tabel, să definească metode care execută query-uri și pot fi apelate cu ușurință în cadrul proiectului, dar și să gestioneze relațiile între tabele într-un mod intuitiv. Încă unul din motivele pentru care am ales să folosesc Room este faptul că acesta validează schema bazei de date și interogările SQL la momentul compilării, reducând astfel riscul de a întâmpina erori la runtime.

Structura bazei de date este una relativ simplă, însă mai mult decât suficientă pentru cerințele funcționale ale aplicației:

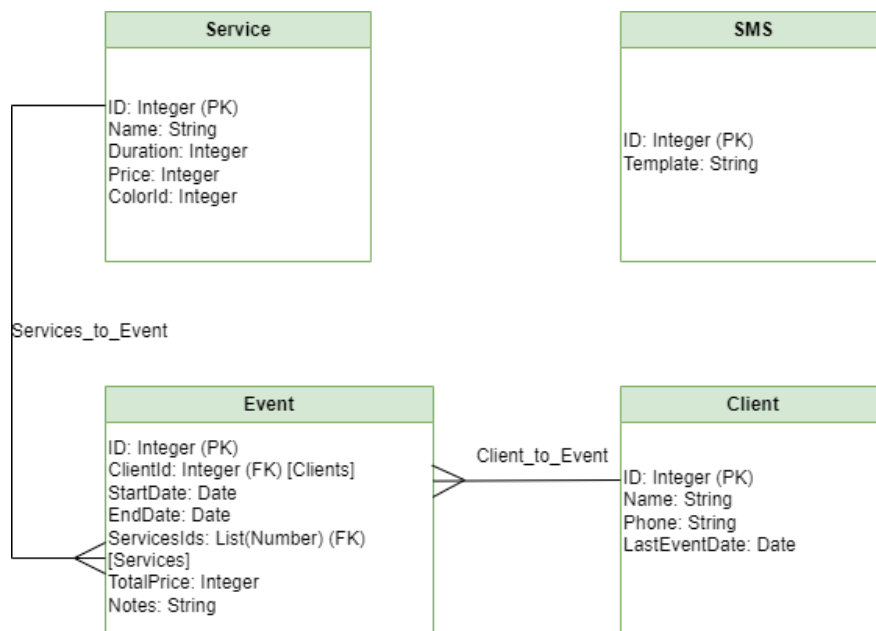


Figura 33: Diagrama bazei de date

Principalele tabele utilizate în baza de date sunt: Service, Event, Client.

Relațiile de legătură între acestea asigură o structură clară a datelor și sunt bine definite, după cum urmează:

- Tabela Event este legată de tabela Client prin intermediul id-ului de client, având o relație ONE-TO-ONE, ceea ce înseamnă că fiecare eveniment este asociat unui singur client.
- De asemenea, Event este legat de tabela Service printr-o relație ONE-TO-MANY, adică un eveniment poate include mai multe servicii.

În plus, este prezentă și o tabelă dedicată pentru stocarea șabloanelor pentru mesaje de reminder, utilizate pentru trimiterea de notificări personalizate către clienți.

Se poate observa că tabelul Service conține o coloană denumită „ColorId”, însă nu este prezent un tabel separat pentru culori. Din motive de simplificare, am ales să definesc paleta de culori în directorul **assets** și să declar o listă de valori statice care să fie salvate în memoria RAM a dispozitivului în momentul rularii aplicației. În acest fel, aceste culori sunt mai ușor de accesat în cadrul aplicației, eliminând necesitatea unei interogări suplimentare a bazei de date, care ar fi fost costisitoare din punct de vedere al timpului de procesare.

După cum am menționat anterior, Room oferă suport pentru gestionarea relațiilor între tabele, ceea ce face ca implementarea acestora să fie intuitivă și bine structurată. Pentru a ilustra acest proces, voi detalia în continuare relația ONE-TO-MANY dintre tabelele Event și Service.

Pentru început, am definit structura celor două tabele utilizând două clase adnotate cu **@Entity**. Această adnotare indică faptul că fiecare clasă corespunde unei tabele din baza de date. Pentru o mai bună înțelegere, am inclus o parte din codul necesar definirii clasei Event:

```
@Entity
public class Event {

    @PrimaryKey(autoGenerate = true)
    private int id;

    private String eventId;
    private String clientId;
    private Calendar startDate;
    private Calendar endDate;
    private String reminders;
    private int price;
    private String notes;
```

Figura 34: Modelul de date pentru tabela Event

Adnotarea **@PrimaryKey** are ca rol generarea automată a id-ului de către bibliotecă.

În continuare, a fost necesar să definesc relația dintre cele două tabele. Acest lucru a fost realizat prin crearea unui tabel auxiliar de legătură (join table) care conține coloanele după care se face identificarea între tabele:

```
@Entity(primaryKeys = {"eventId", "serviceId"})
class EventServiceCrossRef {
    public int eventId;
    public int serviceId;
}
```

Figura 35: Tabela de join pentru Event si Service

Un ultim pas a fost definirea unei clase care să conțină atributele celor două tabele. Această clasă permite stocarea datelor rezultate în urma query-urilor care extrag informații din ambele tabele.

```

public class EventWithServices {
    @Embedded public Event event;

    @Relation(
        parentColumn = "id",
        entityColumn = "id",
        associateBy = @Junction(
            value = EventServiceCrossRef.class,
            parentColumn = "eventId",
            entityColumn = "serviceId"
        )
    )
    public List<Service> services;
}

```

Figura 36: Model de date pentru tabela de join

Adnotarea **@Embedded** este utilizată pentru a include toate câmpurile din clasa Event în noua clasă.

@Relation definește legătura dintre tabele, specificând că un eveniment poate avea mai multe servicii asociate. Iar **@Junction** este folosit pentru a specifica tabelul de legătură utilizat.

6 EVALUAREA REZULTATELOR

Cu scopul de a valida și analiza calitatea produsului final, a fost realizat un chestionar pe baza căruia potențialii clienți ai aplicației să își exprime punctul de vedere per ansamblu, dar și punctual asupra anumitor aspecte funcționale și nefuncționale.

Drept urmare, concluziile extrase în urma rezultatelor au fost următoarele:

1. Utilitatea Modurilor de Vizualizare:

Cat de utila gasesti optiunea de a vizualiza calendarul in mai multe moduri (zi, 3 zile, saptamana, luna)?

34 responses

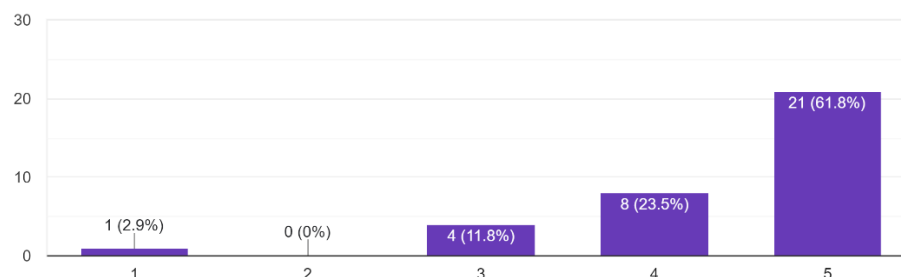


Figura 37: Utilitatea Modurilor de Vizualizare

2. Importanța sincronizării cu Google Calendar:

Cat de importanta este pentru dvs. integrarea cu Google Calendar?

34 responses

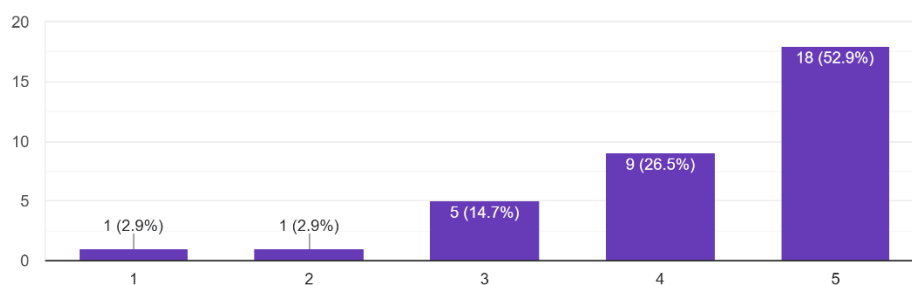


Figura 38: Importanța sincronizării cu Google Calendar

3. Complexitatea opțiunilor de configurare a evenimentelor:

Esti multumit de optiunile disponibile pentru configurarea evenimentelor (ora, data, nume client, serviciu, pret, etc)?

34 responses

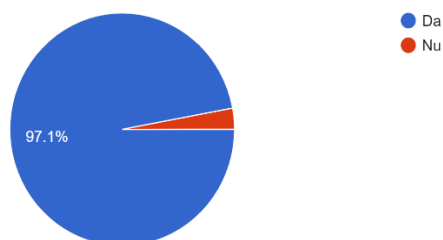


Figura 39: Complexitatea opțiunilor de configurare a evenimentelor

4. Complexitatea opțiunilor de configurare a serviciilor:

Ti se par suficiente optiunile disponibile pentru configurarea unui serviciu (nume, durata, pret, culoare)?

33 responses

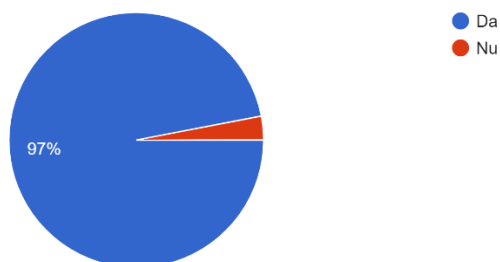


Figura 40: Complexitatea opțiunilor de configurare a serviciilor

5. Importanța mesajelor de reminder:

Ti se par eficiente mesajele de reminder trimise clientilor?

34 responses

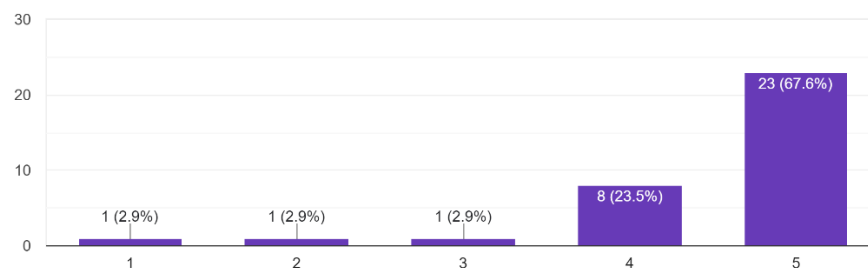


Figura 41: Importanța mesajelor de reminder

6. Importanța posibilității promovării serviciilor:

Ti se pare utila functia de promovare a evenimentelor viitoare sau a ofertelor?

33 responses

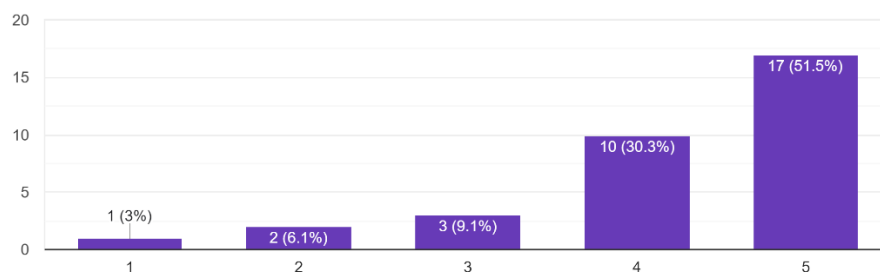


Figura 42: Importanța posibilității promovării serviciilor

7. Utilitatea statisticilor despre clienți:

Crezi ca te-ar ajuta sa cunosti ce clienti nu au mai dorit vreun serviciu in ultima perioada?

34 responses

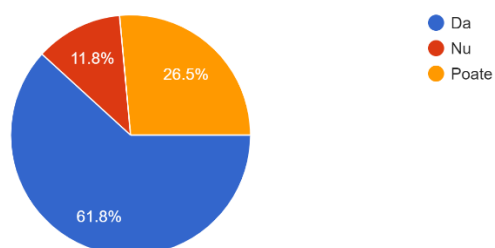


Figura 43: Utilitatea statisticilor despre clienți vechi

Cat de folositor ti se pare clasamentul celor mai profitabili clienti?
34 responses

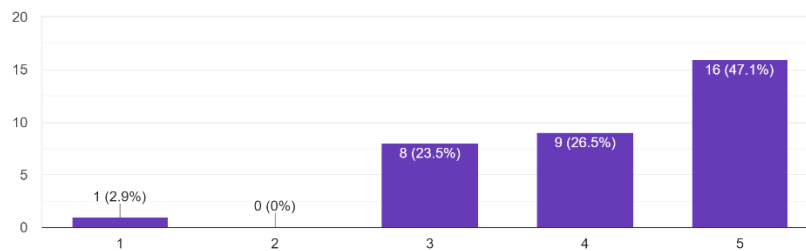


Figura 44: Utilitatea statisticilor despre totalul plăților

8. Experiența per ansamblu:

Cum apreciezi întreaga experiența oferită de aplicație?
34 responses

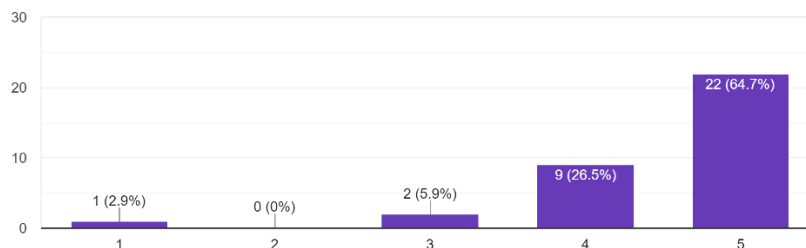


Figura 45: Experiența per ansamblu

„Task Planner” a reușit să ofere utilizatorilor un mediu eficient și intuitiv, prevăzut cu funcționalități specifice, menite să facă gestionarea afacerii lor cât mai facilă. Așadar, pot spune că obiectivele propuse inițial au fost îndeplinite cu succes.

7 CONCLUZII SI DEZVOLTĂRI ULTERIOARE

7.1 Concluzii

Dezvoltarea proiectului „Task Planner” a fost un proces complex și plin de provocări, care a inclus momente de dificultate în ceea ce privește implementarea funcționalităților, cât și momente în care soluțiile gândite inițial au necesitat regândire și ajustare.

Rezultatele obținute au fost satisfăcătoare, deoarece aplicația mobilă a reușit să ofere utilizatorului un mediu prietenos organizării întreprinderii lor.

Experiența câștigată în urma acestui proces este inestimabilă. Am învățat să gestionez un proiect de amploare de la început până la sfârșit. Am căpătat cunoștințe noi din zona

dezvoltării unui produs software, precum: discuția cu publicul țintă, înțelegerea nevoilor acestuia și punerea în aplicare a acestora pe baza discernământului personal.

7.2 Dezvoltări ulterioare

Pe parcursul dezvoltării aplicației s-au ivit câteva idei care ar face experiența de utilizare a acesteia mult mai facilă pentru client, dar și care ar putea să o scaleze la nivel de companie.

Acestea sunt:

- Opțiunea de a apela clienții direct din cadrul aplicației.
- Implementarea unei logici pentru a notifica utilizatorul dacă evenimentele din calendarul aplicației se suprapun cu evenimentele din Google Calendar.
- Implementarea unui algoritm care să determine următorul interval orar potrivit pentru o programare, în cazul în care există conflicte.
- Notificări pop-up care să notifice utilizatorul despre începerea unui eveniment sau programare.
- Integrarea cu smartwatch.
- Dezvoltarea unui sistem de centralizare unde să fie stocate datele din aplicația fiecărui angajat din cadrul companiei.
- Extinderea funcționalităților de statistică pentru a include date la nivelul angajaților (firmei deținute de utilizator) care folosesc aplicația.

8 BIBLIOGRAFIE

[1] Smartphone time usage. [Online]. Available:

<https://backlinko.com/smartphone-usage-statistics> [Accesat: Iunie 2024]

[2] Smartphone users worldwide. [Online]. Available:

<https://whatsthebigdata.com/smartphone-stats/> [Accesat: Iunie 2024]

[3] Time management research. [Online]. Available:

<https://www.acuitytraining.co.uk/news-tips/time-management-statistics-2022-research/> [Accesat: Iunie 2024]

[4] Best scheduling apps. [Online]. Available:

<https://www.forbes.com/advisor/business/software/best-scheduling-apps/>
[Accesat: Mai 2024]

[5] Square Appointments. [Online]. Available:

<https://www.forbes.com/advisor/business/square-appointments-review/>

- [Accesat: Mai 2024]
- [6] Imagine Square Appointments. [Online]. Available:
<https://squareup.com/us/en/appointments> [Accesat: Iunie 2024]
- [7] Setmore. [Online]. Available:
<https://www.forbes.com/advisor/business/software/setmore-review/>
[Accesat: Mai 2024]
- [8] Imagine Setmore. [Online]. Available:
<https://www.setmore.com/> [Accesat: Iunie 2024]
- [9] Android. [Online]. Available:
https://insse.ro/cms/sites/default/files/field/publicatii/accesul_populatiei_la_tehnologia_informatiei_si_comunicatiilor_romania_2020.pdf [Accesat: Iunie 2024]
- [10] Java. [Online]. Available:
<https://www.ibm.com/topics/java> [Accesat: Martie 2024]
- [11] Android Studio. [Online]. Available:
<https://developer.android.com/studio/intro> [Accesat: Martie 2024]
- [12] Unity. [Online]. Available:
<https://docs.unity3d.com/Manual/index.html> [Accesat: Iunie 2024]
- [13] Eclipse. [Online]. Available:
<https://www.eclipse.org/home/whatis/> [Accesat: Iunie 2024]
- [14] Android Development Tools. [Online]. Available:
<https://marketplace.eclipse.org/content/android-development-tools-eclipse>
[Accesat: Iunie 2024]
- [15] Alamkanak calendar. [Online]. Available:
<https://github.com/alamkanak/Android-Week-View> [Accesat: Martie 2024]
- [16] AppCompatActivity. [Online]. Available:
<https://developer.android.com/reference/kotlin/androidx/appcompat/app/AppCompatActivity?hl=en> [Accesat: Februarie 2024]
- [17] Google Calendar. [Online]. Available:
<https://developers.google.com/calendar/api/guides/overview> [Accesat: Aprilie 2024]

[18] SMS Manager. [Online]. Available:

<https://developer.android.com/reference/android/telephony/SmsManager>

[Accesat: Aprilie 2024]

[19] Room Database. [Online]. Available:

<https://developer.android.com/reference/kotlin/androidx/room/RoomDatabase?hl=en> [Accesat: Martie 2024]

[20] SQLite. [Online]. Available: <https://www.sqlite.org/> [Accesat: Martie 2024]

9 ANEXE

9.1 Anexa 1: Formular pentru analiza cerințelor

Ce varsta ai?

33 responses

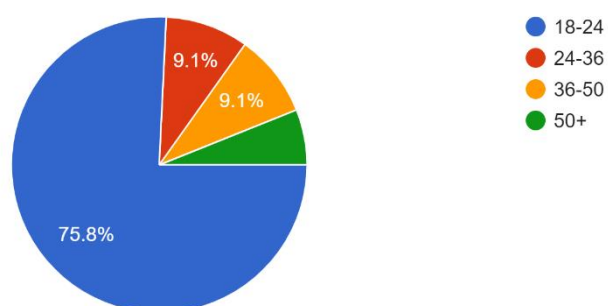


Figura 46: Vârsta clienților

În medie, cât timp petreci pe zi pe telefonul mobil?

37 responses

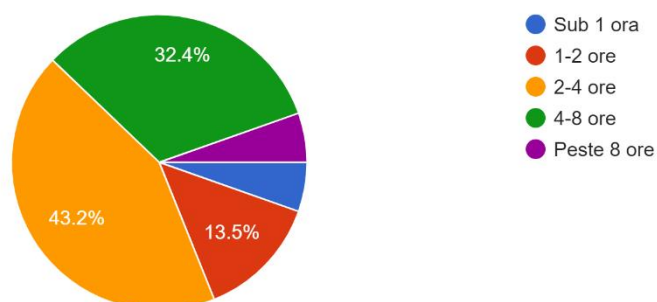


Figura 47: Media utilizării telefonului mobil

Cum te descurci cu gestionarea timpului, în general?

37 responses

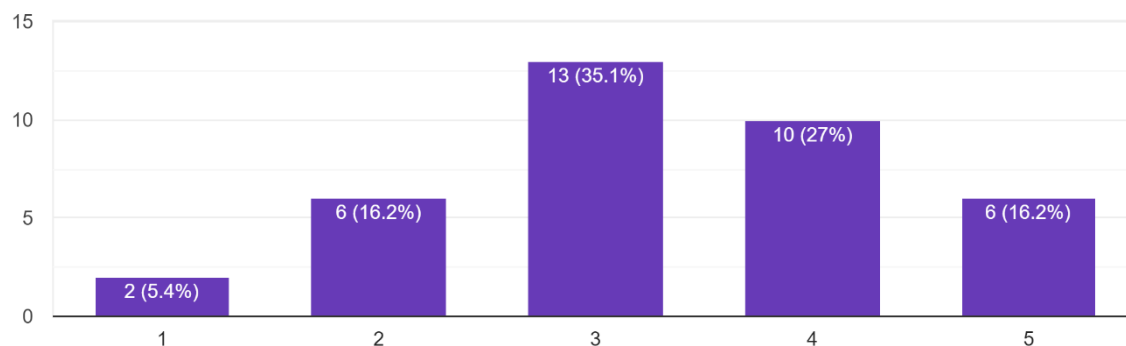


Figura 48: Gestionarea timpului

Folosești aplicații pentru administrarea timpului?

36 responses

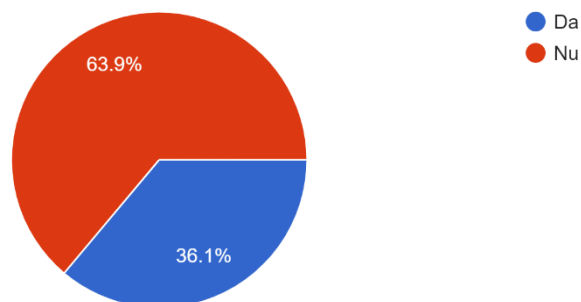


Figura 49: Procent de utilizare al aplicațiilor de gestiune a timpului

Cat de mult apreciezi aspectul vizual al unei aplicatii mobile?

37 responses

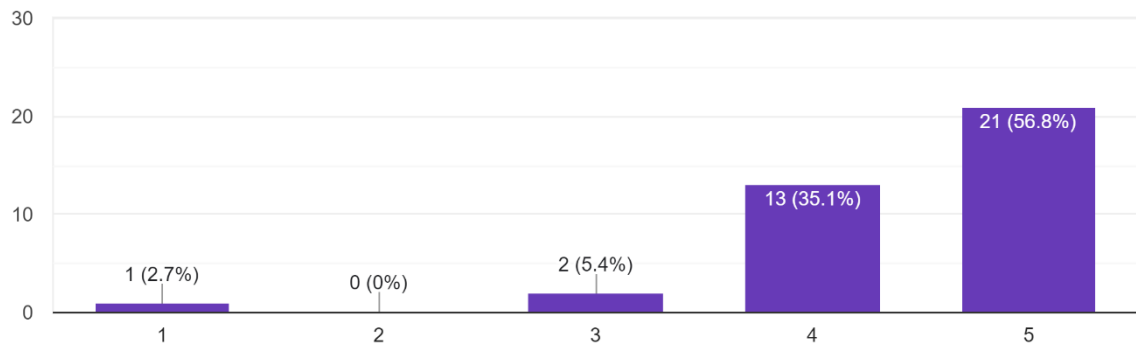


Figura 50: Importanța aspectului vizual

Care sunt cele mai importante functionalitati pe care le asteptati de la o aplicatie de gestionare a programarilor?

33 responses

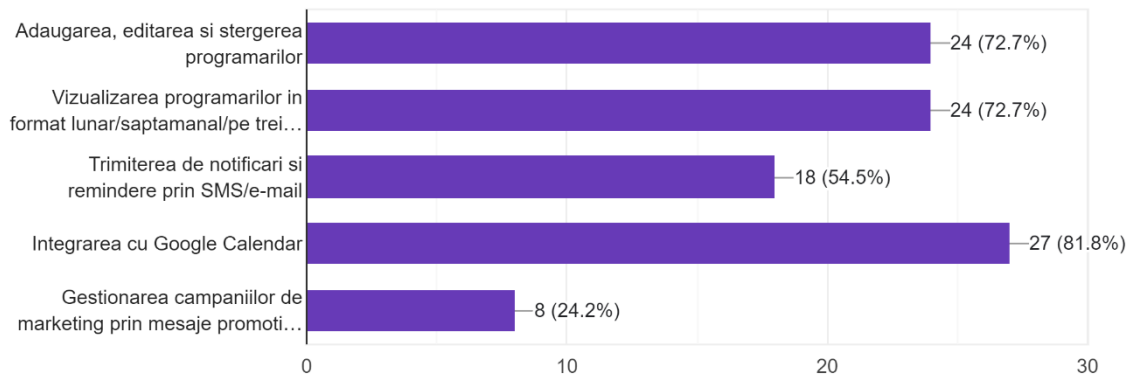


Figura 51: Cele mai importante funcționalități