

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

MeetUp!
RAPORT COMPLEMENTAR DE PROIECTARE

Echipa 9
Enescu Irina Ștefania
Panait Ana-Maria

2025

Cuprins

1. Scopul aplicației.....	3
2. Aria de acoperire a aplicației.....	3
3. Preliminarii.....	4
3.1 Specificații de analiză.....	4
3.2 Specificații de proiectare.....	11
3.3 Specificații tehnologice.....	17
3.4 Componente de dezvoltare.....	18
3.5 Componente de testare.....	20
3.6 Componente de comunicare.....	21
3.7 Componente de învățare.....	22
4. Perspectiva de dezvoltare.....	22

1. Scopul aplicației

Aplicația MeetUp! este o platformă digitală interactivă al cărei scop este creșterea frecvenței întâlnirilor sociale reale între prieteni, într-o eră în care tehnologia a înlocuit contactul direct. Aplicația facilitează organizarea întâlnirilor față-în-față, oferind o experiență digitală simplă, motivantă și ludică.

Misiunea aplicației este de a transforma planificarea socializării într-un proces natural și plăcut: utilizatorii pot vedea poziția prietenilor pe hartă, pot câștiga puncte pentru întâlniri confirmate în viața reală, pot participa la misiuni și provocări, iar sistemul AI le sugerează activități sau evenimente potrivite. Prin gamificare – puncte, badge-uri, misiuni și recompense digitale – aplicația încurajează interacțiunile autentice și spontane.

Beneficiarii principali:

- Tineri și studenți care vor să păstreze legătura cu prietenii, dar au programe încărcate.
- Persoane cu viață socială redusă după pandemie care au nevoie de motivație pentru a ieși din mediul digital.
- Grupuri de prieteni, colegi sau familii care doresc să organizeze mai ușor ieșiri, activități sau evenimente.
- Organizatori de evenimente sau locații sociale (cafenele, săli de evenimente) – pot fi parteneri în aplicație pentru recomandări AI.

MeetUp! contribuie la reconstruirea conexiunilor sociale, transformând mediul digital într-un sprijin pentru relațiile umane, nu un substitut al acestora.

2. Aria de acoperire a aplicației

MeetUp! este o platformă digitală interactivă care facilitează organizarea și creșterea frecvenței întâlnirilor față-în-față între prieteni, într-un mod simplu, motivant și ludic. Aplicația nu este un simplu chat, nici o rețea socială clasică axată pe postări sau conținut public; accentul nu este pus pe interacțiuni virtuale, ci pe sprijinirea contactelor reale și pe transformarea planificării întâlnirilor într-o experiență atractivă și ușor de inițiat.

Roluri în aplicație:

- Utilizator obișnuit – își vede prietenii pe hartă, acceptă/organizează întâlniri, câștigă puncte, finalizează misiuni și poate cheltui punctele pentru beneficii digitale.
- Organizator de evenimente – poate crea evenimente publice sau private în aplicație (ex: brunch, seară de boardgames), invita utilizatori și oferi recompense suplimentare pentru participare.
- Deținători de locații (cafenele, săli, spații sociale) – pot înregistra locația lor în aplicație, să promoveze oferte speciale pentru întâlniri și să devină puncte de interes pentru misiuni sau evenimente.

Obiective principale prin care aplicația își atinge scopul de creștere a frecvenței interacțiunilor:

- Asigurarea funcționalităților esențiale necesare pentru facilitarea întâlnirilor între utilizatori, precum vizualizarea prietenilor pe hartă, propunerea rapidă de ieșiri, organizarea grupurilor și confirmarea întâlnirilor în aplicație.
- Creșterea frecvenței întâlnirilor prin utilizarea tehnicilor de gamification – puncte pentru întâlniri confirmate, misiuni, provocări și recompense digitale care stimulează interacțiunile față-în-față.

3. Preliminarii

3.1 Specificații de analiză

3.1.1. Specificații funcționale

F1: Ca utilizator, vreau să pot vedea prietenii mei pe o hartă în aplicație, pentru a identifica oportunități de întâlnire în apropiere și a crește frecvența întâlnirilor față în față.

- **F1.1:** Ca utilizator, vreau ca aplicația să îmi ceară explicit acordul pentru partajarea locației, astfel încât să pot decide dacă apar sau nu pe hartă.
- **F1.2:** Ca utilizator, vreau să văd pe hartă poziția mea curentă, pentru a mă orienta mai ușor în raport cu prietenii.
- **F1.3:** Ca utilizator, vreau să văd pe hartă prietenii mei care își partajează locația, pentru a identifica rapid cine este în apropiere.
- **F1.4:** Ca utilizator, vreau să pot vedea, numele și statusul de disponibilitate al unui prieten de pe hartă, pentru a decide dacă propun o întâlnire.

F2: Ca utilizator, vreau să pot propune rapid o întâlnire selectând prieteni, locația și ora, pentru a reduce efortul de coordonare și a crește frecvența întâlnirilor față în față.

- **F2.1:** Ca utilizator, vreau să pot selecta unul sau mai mulți prieteni din listă sau de pe hartă pentru o întâlnire, astfel încât să trimit invitația direct celor vizați.
- **F2.2:** Ca utilizator, vreau să pot alege o locație pentru întâlnire cautând-o pe hartă, pentru a propune mai ușor un loc cunoscut.
- **F2.3:** Ca utilizator, vreau să pot seta detaliile întâlnirii (data, oră, mesaj) înainte de a trimite invitația, pentru a clarifica contextul întâlnirii.
- **F2.4:** Ca utilizator, vreau să primesc notificări că invitația a fost acceptată sau nu de către prieteni, pentru a fi sigur cine vine la întâlnire.
- **F2.5:** Ca utilizator, vreau să o văd o listă cu întâlnirile propuse de mine, pentru a ține evidența lor.

F3: Ca utilizator, vreau să pot accepta sau respinge invitațiile la întâlniri și să pot confirma participarea, pentru a organiza mai ușor timpul și a crește frecvența întâlnirilor față în față.

- **F3.1:** Ca utilizator, vreau să văd o listă a invitațiilor primite cu detalii (organizator, locație, dată, ora), pentru a decide dacă particip.

- **F3.2:** Ca utilizator, vreau să pot accepta sau respinge o invitație foarte ușor și rapid, pentru a economisi timp.
- **F3.3:** Ca utilizator, vreau să primesc notificări la primirea unei invitații, pentru a fi mereu la curent.

F4: Ca utilizator, vreau să pot organiza grupuri pentru ieșiri și coordonare, pentru a planifica mai ușor activitățile sociale și a crește frecvența întâlnirilor față în față.

- **F4.1:** Ca utilizator, vreau să pot crea un grup de prieteni, pentru a organiza mai ușor ieșiri recurente.
- **F4.2:** Ca utilizator, vreau să pot adăuga sau elimina membri dintr-un grup creat, pentru a menține grupul actualizat.
- **F4.3:** Ca utilizator, vreau să pot vedea o listă cu toate grupurile din care fac parte, pentru a le gestiona mai ușor.

F5: Ca utilizator, vreau să pot câștiga puncte pentru întâlnirile confirmate, pentru a fi motivat să particip la mai multe întâlniri față în față.

- **F5.1:** Ca utilizator, vreau să văd în profilul meu numărul total de puncte acumulate, pentru a-mi urmări progresul.
- **F5.2:** Ca utilizator, vreau să văd un jurnal al punctelor câștigate (data, tip activitate, număr puncte), pentru transparență.

F6: Ca utilizator, vreau să pot finaliza misiuni și provocări și să primesc recompense digitale, pentru a-mi crește motivația și frecvența întâlnirilor față în față.

- **F6.1:** Ca utilizator, vreau să văd o listă de misiuni, pentru a ști ce pot face pentru a câștiga recompense.
- **F6.2:** Ca utilizator, vreau să primesc o notificare când finalizez o misiune și primesc recompensa, pentru a fi motivat să continui.
- **F6.3:** Ca utilizator, vreau să pot vedea istoricul misiunilor completate și al recompenselor primite, pentru a urmări implicarea mea în timp.

F7: Ca utilizator, vreau să pot cheltui punctele acumulate pentru beneficii digitale sau acces la funcționalități și evenimente, pentru a participa mai ușor la activități sociale potrivite și a crește frecvența întâlnirilor față în față.

- **F7.1:** Ca utilizator, vreau să văd un magazin de beneficii digitale, pentru a decide cum folosesc punctele.
- **F7.2:** Ca utilizator, vreau să pot cheltui puncte pentru a debloca anumite funcționalități, pentru a beneficia de avantajele implicării mele.
- **F7.3:** Ca utilizator, vreau să văd clar câte puncte costă fiecare beneficiu, pentru a lua decizii informate.
- **F7.4:** Ca utilizator, vreau să primesc o confirmare la achiziția unui beneficiu, pentru a ști că tranzacția a reușit.

F8: Ca utilizator, vreau să pot marca anumiți prieteni ca „Apropiati”, pentru a controla mai bine vizibilitatea informațiilor personale și a crește frecvența întâlnirilor față în față.

- **F8.1:** Ca utilizator, vreau să pot marca sau demarca un prieten ca „Apropiat” din lista de prieteni, pentru a controla relațiile rapid.
- **F8.2:** Ca utilizator, vreau să văd în lista de prieteni un indicator pentru cei marcați ca fiind „Apropiati”, pentru a-i identifica rapid.

F9: Ca utilizator, vreau ca doar prietenii marcați ca Apropiati să îmi poată vedea locația curentă, pentru a mă simți în siguranță și a crește frecvența întâlnirilor față în față.

- **F9.1:** Ca utilizator, vreau ca locația mea să fie vizibilă pentru prietenii marcați ca Apropiati, pentru a mă simți în siguranță.
- **F9.2:** Ca utilizator, vreau să văd o listă cu cine îmi poate vedea locația în funcție de setările curente, pentru a evita confuziile.

F10: Ca utilizator, vreau să pot dezactiva partajarea locației oricând, temporar sau permanent, pentru a păstra controlul asupra confidențialității și a crește frecvența întâlnirilor față în față.

- **F10.1:** Ca utilizator, vreau să am o opțiune simplă pentru a dezactiva temporar partajarea locației, pentru situațiile în care vreau confidențialitate.
- **F10.2:** Ca utilizator, vreau să pot alege între dezactivare temporară și dezactivare permanentă a locației, pentru mai mult control.
- **F10.3:** Ca utilizator, vreau să văd clar când partajarea locației este activată, pentru a nu uita că este pornită.

F11: Ca utilizator, vreau să pot alege intervalul de actualizare a locației (5, 15, 30 minute sau manual), pentru a echilibra acuratețea cu consumul de baterie și a crește frecvența întâlnirilor față în față.

- **F11.1:** Ca utilizator, vreau să pot selecta intervalul de actualizare a locației dintr-o secțiune de setări, pentru a optimiza consumul de baterie.

F12: Ca utilizator, vreau să pot seta un status de disponibilitate (Liber, La cursuri, La muncă, Nu pot acum), pentru a comunica mai ușor disponibilitatea mea și a crește frecvența întâlnirilor față în față.

- **F12.1:** Ca utilizator, vreau să pot alege un status de disponibilitate, pentru a comunica rapid disponibilitatea mea.
- **F12.2:** Ca utilizator, vreau ca prietenii să poată vedea statusul meu, pentru a ști dacă e momentul potrivit să propună o întâlnire.

F13: Ca utilizator, vreau să pot seta intervale orare de disponibilitate pentru întâlniri, pentru a accelera potrivirea și a crește frecvența întâlnirilor față în față.

- **F13.1:** Ca utilizator, vreau să pot defini intervale orare recurente de, pentru ca prietenii să știe dinainte când pot ieși.

F16: Ca utilizator, vreau să pot vedea statistici și insight-uri bazate pe inteligență artificială despre activitatea mea socială (persoane întâlnite, locuri preferate, evoluția în timp), pentru a lua decizii mai informate și a crește frecvența întâlnirilor față în față.

- **F16.1:** Ca utilizator, vreau să văd câte întâlniri față în față am avut într-o perioadă, pentru a evalua activitatea socială.
- **F16.2:** Ca utilizator, vreau să văd topul locurilor în care mă întâlnesc cel mai des, pentru a identifica preferințele mele de locații.
- **F16.3:** Ca utilizator, vreau să primesc insight-uri generate automat, pentru a vedea statistici interesante.

F17: Ca utilizator, vreau să pot vedea topul prietenilor apropiați în funcție de timpul petrecut împreună sau întâlnirile confirmate, pentru a fi motivat să interacționez mai des și a crește frecvența întâlnirilor față în față.

- **F17.1:** Ca utilizator, vreau să văd un clasament al prietenilor mei în funcție de numărul întâlnirilor confirmate, pentru a ști cu cine petrec cel mai mult timp și cu cine ies cel mai des.
- **F17.2:** Ca utilizator, vreau să pot filtra acest clasament doar la prietenii Aproțiați, pentru a mă concentra pe cercul intim.

F21: Ca organizator, vreau să pot crea evenimente publice sau private (ex. brunch, board games), pentru a aduna comunitatea și a crește frecvența întâlnirilor față în față.

- **F21.1:** Ca organizator, vreau să pot crea un eveniment nou cu titlu, descriere, locație, dată și oră, pentru a invita oamenii la o activitate.
- **F21.2:** Ca organizator, vreau să pot seta dacă evenimentul este public sau privat, pentru a controla cine îl poate vedea.
- **F21.3:** Ca organizator, vreau să pot limita numărul de participanți la eveniment, pentru a preveni supra-aglomerarea.
- **F21.4:** Ca organizator, vreau să văd lista de evenimente pe care le-am creat, pentru a le putea gestiona ulterior (modifica, anula).

F22: Ca organizator, vreau să pot invita utilizatori la evenimente, pentru a facilita coordonarea și a crește frecvența întâlnirilor față în față.

- **F22.1:** Ca organizator, vreau să pot selecta prieteni pentru a-i invita la eveniment, pentru a simplifica procesul.
- **F22.2:** Ca organizator, vreau ca utilizatorii invitați să primească notificări despre eveniment, pentru a crește rata de răspuns.
- **F22.3:** Ca organizator, vreau să văd statusul invitațiilor (acceptat, respins, în așteptare), pentru a estima participarea la eveniment.

F23: Ca organizator, vreau să pot acorda puncte suplimentare pentru participarea la evenimente, pentru a stimula prezența și a crește frecvența întâlnirilor față în față.

- **F23.1:** Ca organizator, vreau să pot configura un număr de puncte bonus pentru participarea la eveniment, pentru a stimula înscrierile.
- **F23.2:** Ca utilizator, vreau să primesc punctele bonus automat după ce particip la evenimentul la care m-am înscris, pentru a fi recompensat corect.

F24: Ca organizator, vreau să pot configura evenimente speciale cu criterii de eligibilitate (de exemplu, pentru cei mai sociali utilizatori), pentru a recompensa implicarea și a crește frecvența întâlnirilor față în față.

- **F24.1:** Ca organizator, vreau să pot defini criterii de eligibilitate pentru un eveniment, pentru a-l face exclusiv.
- **F24.2:** Ca utilizator, vreau să văd clar dacă sunt eligibil sau nu pentru un eveniment și de ce, pentru transparență.

F25: Ca deținător de locație, vreau să îmi pot înregistra locația în aplicație, pentru a facilita planificarea întâlnirilor și a crește frecvența întâlnirilor față în față.

- **F25.1:** Ca deținător de locație, vreau să pot introduce adresa, numele și tipul locației, pentru a fi afișată corect pe hartă.
- **F25.2:** Ca deținător de locație, vreau ca locația mea să fie afișată pe hartă, pentru ca utilizatorii să o poată alege pentru întâlniri.

F26: Ca deținător de locație, vreau să pot promova oferte speciale legate de întâlniri, pentru a atrage utilizatorii și a crește frecvența întâlnirilor față în față.

- **F26.1:** Ca deținător de locație, vreau să pot crea campanii, pentru a atrage cât mai mulți utilizatori.
- **F26.2:** Ca utilizator, vreau să pot vedea ofertele asociate unei locații atunci când o consult, pentru a decide dacă merg acolo.

F27: Ca deținător de locație, vreau ca locația mea să devină un punct de interes (POI) pentru misiuni sau evenimente, pentru a încuraja vizitele și a crește frecvența întâlnirilor față în față.

- **F27.1:** Ca deținător de locație, vreau să pot solicita ca locația mea să fie folosită ca POI în misiuni sau evenimente, pentru a atrage utilizatori.
- **F27.2:** Ca administrator al aplicației, vreau să pot aproba sau respinge astfel de cereri, pentru a controla calitatea POI-urilor.

F28: Ca deținător de locație, vreau să pot gestiona un profil public al locației (cu descriere, imagini și recenzii), pentru a oferi informații relevante și a crește frecvența întâlnirilor față în față.

- **F28.1:** Ca deținător de locație, vreau să pot adăuga o descriere, imagini și programul de funcționare al locației, pentru a informa utilizatorii.
- **F28.2:** Ca utilizator, vreau să pot lăsa o recenzie și un rating unei locații, pentru a împărtăși experiența mea cu alți utilizatori.
- **F28.3:** Ca utilizator, vreau să pot vedea rating-ul mediu și recenziile unei locații, pentru a alege mai informat locul întâlnirii.

3.1.2. Specificații de calitate

NF1: Ca echipă de dezvoltare, vreau ca aplicația să fie dezvoltată cross-platform în Flutter (Android/iOS) dintr-o singură bază de cod, pentru a accelera livrarea și a facilita disponibilitate largă.

- **NF1.1:** Ca dezvoltator, vreau să implementez proiectul Flutter de bază cu structura de foldere agreată, pentru a avea un punct de pornire unitar.

NF2: Ca echipă de dezvoltare, vreau ca backend-ul aplicației să fie implementat în Python folosind FastAPI sau Flask și să expună un API REST standardizat, pentru a facilita evoluția și extinderea rapidă a serviciilor.

- **NF2.1:** Ca dezvoltator, vreau să implementez partea de backend utilizând Python cu framework-ul FastAPI, pentru a implementa API-ul REST.
- **NF2.3:** Ca dezvoltator, vreau să implementez mecanisme standard de tratare a erorilor și răspunsuri consistente, pentru a simplifica integrarea frontend-ului.

NF3: Ca administrator de baze de date, vreau ca datele aplicației să fie stocate într-o bază de date relațională MariaDB, pentru a asigura consistența datelor, fiabilitate operațională.

- **NF3.1:** Ca dezvoltator, vreau să definesc schema bazei de date, pentru a asigura structurarea coerentă a datelor.
- **NF3.2:** Ca dezvoltator, vreau să folosesc tranzacții în operațiile critice precum creare întâlniri, confirmări, acordare puncte, pentru a evita inconsistențele.

NF4: Ca arhitect software, vreau ca soluția să fie construită pe o arhitectură pe trei niveluri (frontend, backend și bază de date), pentru a simplifica mentenanța aplicației și a susține fiabilitatea acesteia.

- **NF4.1:** Ca arhitect software, vreau să definesc clar separarea între frontend (Flutter), backend (API) și baza de date, pentru a simplifica mentenanța.
- **NF4.2:** Ca dezvoltator, vreau să implementez serviciul de backend pe baza unor layer-e separate (logică, acces bază de date), pentru a susține fiabilitatea.

NF5: Ca echipă de dezvoltare, vreau ca arhitectura sistemului să asigure scalabilitate, securitate și ușurință în extinderea funcționalităților, pentru a susține dezvoltările viitoare.

- **NF5.1:** Ca dezvoltator, vreau să folosesc practici de codare modulare (feature-

based, repository pattern), pentru a facilita adăugarea de noi funcționalități.

- **NF5.2:** Ca dezvoltator, vreau să folosesc practici avansate de gestionare a securității, pentru a respecta standardele de securitate.

NF6: Ca echipă de dezvoltare, vreau ca sistemul de gamification (misiuni, puncte, recompense, clasamente) să nu aibă impact asupra performanței sistemului, pentru a menține motivația utilizatorilor de a folosi aplicația.

- **NF6.1:** Ca dezvoltator, vreau să calculez punctele și statusul misiunilor în mod asincron acolo unde e posibil, pentru a nu bloca interacțiunile utilizatorului.

NF7: Ca echipă de dezvoltare, vreau ca aplicația să respecte principiile confidențialității prin design, cu controale clare pentru partajarea și vizibilitatea locației, pentru a spori încrederea și siguranța utilizatorilor.

- **NF7.1:** Ca dezvoltator, vreau ca opțiunile de partajare a locației și statutului să fie dezactivate implicit, pentru a nu expune utilizatorii fără consimțământ.
- **NF7.2:** Ca utilizator, vreau să pot vedea și modifica oricând din setări cine îmi vede locația, pentru a menține controlul.

NF8: Ca echipă de dezvoltare, vreau ca actualizarea locației să funcționeze fiabil în background, cu un consum optim de baterie, pentru a păstra buna funcționare a celorlalte funcționalități ale aplicației.

- **NF8.1:** Ca dezvoltator, vreau să implementez actualizarea locației în background, pentru a păstra funcționalitatea fără a consuma excesiv bateria.

NF9: Ca echipă de dezvoltare, vreau ca aplicația să respecte cerințele de securitate (autentificare sigură, protejarea datelor în tranzit și în repaus), pentru a proteja informațiile personale ale utilizatorilor.

- **NF9.1:** Ca dezvoltator, vreau ca toate comunicațiile dintre aplicație și server să fie făcute prin HTTPS, pentru a proteja datele în tranzit.
- **NF9.2:** Ca dezvoltator, vreau să implementez un mecanism de autentificare sigur (ex: JWT cu expirare), pentru a controla accesul la resurse.
- **NF9.3:** Ca administrator, vreau ca datele sensibile din bază - parole, token-uri să fie stocate criptat sau hash-uit, pentru a reduce riscul în caz de breach.

NF10: Ca echipă de dezvoltare, vreau ca procesul de verificare a identității utilizatorilor să fie sigur și să protejeze datele personale, pentru a preveni abuzurile și fraudele.

- **NF10.1:** Ca utilizator, vreau să pot parcurge un proces de verificare minimal (ex: email și număr de telefon), pentru a confirma că sunt o persoană reală.
- **NF10.2:** Ca dezvoltator, vreau să stochez doar datele strict necesare pentru verificare, pentru a respecta principiul minimizării datelor.

NF11: Ca echipă de dezvoltare, vreau ca implementarea sistemului de notificări să asigure

livrarea unor mesaje fiabile și la timp, pentru a putea coordona eficient evenimentele.

- **NF11.1:** Ca dezvoltator, vreau să integrez un serviciu de push notifications (ex: Firebase Cloud Messaging), pentru a trimite notificări în timp real.

NF13: Ca echipă de dezvoltare, vreau ca interfața aplicației să fie intuitivă și rapidă pentru task-urile frecvente (proponere de întâlniri, confirmări, setarea statusului), pentru a oferi utilizatorilor o experiență cât mai plăcută de utilizare a aplicației.

- **NF13.1:** Ca dezvoltator, vreau să optimizez ecranele principale pentru încărcare rapidă, pentru ca utilizatorii să nu aștepte mult.

NF16: Ca echipă de dezvoltare, vreau ca principiile GDPR (minimizarea datelor, consimțământul explicit și controlul asupra informațiilor personale) să fie aplicate, pentru a respecta legile referitoare la protecția datelor personale.

- **NF16.1:** Ca utilizator, vreau să văd o secțiune clară de „Confidențialitate și date personale” în aplicație, pentru a înțelege ce date se colectează.
- **NF16.2:** Ca utilizator, vreau să am posibilitatea de a-mi șterge contul și datele asociate, pentru a avea control deplin asupra informațiilor mele.

3.2 Specificații de proiectare

P1.F1: Vizualizarea prietenilor pe hartă

- **P1.F1.1:** Ca dezvoltator frontend, vreau să implementez un modul MapView folosind Google Maps SDK, cu suport pentru poziția utilizatorului, pentru a afișa locațiile utilizatorilor pe hartă.
- **P1.F1.2:** Ca dezvoltator frontend, vreau să afișez marker-e pentru prietenii autorizați, consumând date din API, pentru vizualizarea prietenilor apropiați.
- **P1.F1.3:** Ca dezvoltator backend, vreau un endpoint REST GET /friends/locations care returnează locația prietenilor vizibili, pentru a popula harta.
- **P1.F1.4:** Ca dezvoltator backend, vreau un endpoint PATCH /location/update pentru actualizarea locației utilizatorului, pentru ca poziția să fie actualizată.

P2.F2: Propunerea unei întâlniri

- **P2.F2.1:** Ca dezvoltator frontend, vreau o componentă MeetingCreateView pentru selectarea prietenilor, locației și datei, pentru a crea o invitație.
- **P2.F2.2:** Ca dezvoltator backend, vreau un endpoint POST /meetings care salvează detaliile întâlnirii, incluzând participanții, pentru a trimite invitațiile.
- **P2.F2.3:** Ca dezvoltator backend, vreau un endpoint GET /meetings pentru a returna lista întâlnirilor unui utilizator, pentru afișare în UI.
- **P2.F2.4:** Ca dezvoltator backend, vreau un endpoint GET /meetings/{id} pentru a returna detaliile unei întâlniri, pentru afișare în UI.

- **P2.F2.5:** Ca dezvoltator frontend, vreau o componentă MeetingListView pentru a vizualiza lista întâlnirilor unui utilizator.
- **P2.F2.6:** Ca dezvoltator frontend, vreau o componentă MeetingView pentru a vizualiza detaliile unei întâlniri a unui utilizator.

P3.F3: Acceptarea și respingerea invitațiilor

- **P3.F3.1:** Ca dezvoltator frontend, vreau o componentă InvitationsListView care afișează invitațiile primite, pentru gestionarea participanților.
- **P3.F3.2:** Ca dezvoltator frontend, vreau implementarea unor butoane rapide Accept/Decline, pentru răspuns imediat.
- **P3.F3.3:** Ca dezvoltator backend, vreau un endpoint GET /invitations care returnează invitațiile utilizatorului, pentru a le afișa într-o pagină separată.
- **P3.F3.4:** Ca dezvoltator backend, vreau un endpoint PATCH /invitations/{id} pentru actualizarea statusului invitației, pentru a comunica decizia fiecăruia.
- **P3.F3.5:** Ca dezvoltator backend, vreau un endpoint GET /invitations/{id} care returnează detaliile unei invitații, pentru a le afișa într-o pagină separată.
- **P3.F3.6:** Ca dezvoltator frontend, vreau o componentă InvitationView care afișează detaliile unei invitații primite, pentru gestionarea sa.
- **P3.F3.7:** Ca dezvoltator backend, vreau un eveniment push (WebSocket/Push Notification) către organizator când invitația este răspunsă, pentru actualizare.

P4.F4: Organizarea grupurilor

- **P4.F4.1:** Ca dezvoltator frontend, vreau un ecran GroupsView pentru a afișa grupurile din care face parte un utilizator, pentru gestionarea lor ușoară.
- **P4.F4.2:** Ca dezvoltator frontend, vreau să implementez butoane Add/Remove Members pentru administratorul unui grup, pentru modificarea grupului.
- **P4.F4.3:** Ca dezvoltator backend, vreau să implementez un endpoint POST /groups pentru crearea unui grup, pentru a salva structura socială.
- **P4.F4.4:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /groups/{id}/members pentru gestionarea membrilor unui grup.
- **P4.F4.5:** Ca dezvoltator backend, vreau să implementez un endpoint GET /groups pentru a afișa grupurile din care face parte utilizatorul.
- **P4.F4.6:** Ca dezvoltator frontend, vreau să implementez un buton Leave Group pentru a oferi utilizatorilor posibilitatea de a ieși dintr-un grup.
- **P4.F4.7:** Ca dezvoltator backend, vreau să implementez un endpoint GET /groups/{id} pentru a afișa detaliile unui grup din care face parte utilizatorul.
- **P4.F4.8:** Ca dezvoltator frontend, vreau un ecran GroupView pentru a afișa detaliile unui grup din care face parte un utilizator.

P5.F5: Sistemul de puncte bonus pentru întâlniri

- **P5.F5.1:** Ca dezvoltator backend, vreau un serviciu PointsEngine care acordă puncte în funcție de acțiunile utilizatorilor, pentru gamificare automată.

- **P5.F5.2:** Ca dezvoltator backend, vreau să implementez un endpoint GET /points/summary pentru a întoarce totalul punctelor utilizatorului.
- **P5.F5.3:** Ca dezvoltator backend, vreau să implementez un endpoint GET /points/history pentru vizualizarea tranzacțiilor cu puncte.
- **P5.F5.4:** Ca dezvoltator frontend, vreau o interfață în profil pentru totalul și istoricul punctelor unui utilizator.

P6.F6: Misiuni și provocări

- **P6.F6.1:** Ca dezvoltator frontend, vreau o componentă MissionsListView cu lista misiunilor disponibile pentru gamificare.
- **P6.F6.2:** Ca dezvoltator backend, vreau să implementez un endpoint GET /missions pentru returnarea misiunilor active și stării fiecăreia.
- **P6.F6.3:** Ca dezvoltator backend, vreau să trimit notificări atunci când o misiune este finalizată, pentru a motiva utilizatorul.
- **P6.F6.4:** Ca dezvoltator frontend, vreau MissionsHistoryView pentru afișarea misiunilor finalizate și a recompenselor primite.
- **P6.F6.5:** Ca dezvoltator backend, vreau să implementez un endpoint GET /missions/history pentru a oferi informații referitor la istoricul misiunilor finalizate de utilizator.

P7.F7: Cheltuirea punctelor în magazinul digital

- **P7.F7.1:** Ca dezvoltator frontend, vreau o componentă StoreView cu listă beneficii și costuri în puncte, pentru o selecție rapidă.
- **P7.F7.2:** Ca dezvoltator backend, vreau să implementez un endpoint GET /store/items pentru returnarea listei de beneficii disponibile.
- **P7.F7.3:** Ca dezvoltator backend, vreau să implementez un endpoint POST /store/purchase pentru procesarea tranzacțiilor în puncte.
- **P7.F7.4:** Ca dezvoltator frontend, vreau un popup StorePurchaseConfirmation pentru confirmarea achiziției reușite.

P8.F8: Prieteni apropiați

- **P8.F8.1:** Ca dezvoltator frontend, vreau un toggle în componenta FriendsListView pentru marcarea unui prieten ca Apropiat.
- **P8.F8.2:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /friends/{id}/close-status pentru actualizarea relației.
- **P8.F8.3:** Ca dezvoltator frontend, vreau un indicator vizibil pentru prietenii apropiați în listă.

P9.F9: Vizibilitate locație doar pentru prietenii apropiați

- **P1.F9.1:** Ca dezvoltator backend, vreau o regulă de filtrare a locațiilor, astfel încât doar prietenii marcați Apropiati să poată vedea locația utilizatorului.

- **P1.F9.2:** Ca dezvoltator backend, vreau să implementez un endpoint GET /privacy/location-visibility pentru returnarea listei celor care pot vedea locația.
- **P1.F9.3:** Ca dezvoltator frontend, vreau o secțiune în PrivacySettingsView care afișează cine are acces la locație.

P10.F10: Dezactivarea partajării locației

- **P10.F10.1:** Ca dezvoltator frontend, vreau un control vizibil în componenta PrivacySettingsView pentru dezactivarea locației temporar sau permanent.
- **P10.F10.2:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /location-sharing pentru actualizarea setării de partajare locație.
- **P10.F10.3:** Ca dezvoltator frontend, vreau un indicator vizibil în UI când partajarea locației este dezactivată.

P11.F11: Intervalul de actualizare a locației

- **P11.F11.1:** Ca dezvoltator frontend, vreau o setare prestabilită în componenta PrivacySettingsView pentru selectarea frecvenței de actualizarea locației.
- **P11.F11.2:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /location-update-interval pentru salvarea opțiunii selectate.
- **P11.F11.3:** Ca dezvoltator backend, vreau scheduling configurabil pentru actualizarea coordonatelor pe intervalul ales de utilizator.

P12.F12: Status de disponibilitate

- **P12.F12.1:** Ca dezvoltator frontend, vreau un dropdown în componenta ProfileView pentru setarea statusului.
- **P12.F12.2:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /availability-status pentru actualizarea statusului curent.
- **P12.F12.3:** Ca dezvoltator frontend, vreau afișarea statusului în componenta FriendsListView și pe hartă.

P13.F13: Interval orar de disponibilitate

- **P13.F13.1:** Ca dezvoltator frontend, vreau un modul AvailabilityScheduleView unde utilizatorul poate defini intervale orare recurente.
- **P13.F13.2:** Ca dezvoltator backend, vreau să implementez un endpoint POST /availability-schedule pentru salvarea intervalelor configurate.
- **P13.F13.3:** Ca dezvoltator frontend, vreau vizibilitate vizuală (UI chips/calendar slots) pentru intervalele setate.

P16.F16: Statistici și insight-uri social AI

- **P16.F16.1:** Ca dezvoltator frontend, vreau un dashboard SocialInsightsView cu grafice privind numărul întâlnirilor față în față pe perioade.

- **P16.F16.2:** Ca dezvoltator backend, vreau să implementez un endpoint GET /analytics/meetings-history pentru statistici agregate pe timp.
- **P16.F16.3:** Ca dezvoltator backend, vreau să implementez un endpoint GET /analytics/top-places pentru lista locațiilor frecventate.
- **P16.F16.4:** Ca dezvoltator backend, vreau un modul AI/ML pentru generarea de insight-uri personalizate și endpoint GET /analytics/ai-insights.

P17.F17: Top prieteni în funcție de socializare

- **P17.F17.1:** Ca dezvoltator frontend, vreau o componentă LeaderboardView cu clasament prieteni pe baza întâlnirilor confirmate.
- **P17.F17.2:** Ca dezvoltator backend, vreau să implementez un endpoint GET /leaderboard/friends pentru calcul și returnare clasament.

P21.F21: Crearea și gestionarea evenimentelor publice/private

- **P21.F21.1:** Ca dezvoltator frontend, vreau EventCreateView pentru introducerea detaliilor (titlu, descriere, locație map picker, dată, oră).
- **P21.F21.2:** Ca dezvoltator frontend, vreau un toggle Public/Privat în formularul de creare eveniment.
- **P21.F21.3:** Ca dezvoltator backend, vreau să implementez un endpoint POST /events pentru creare eveniment cu validări.
- **P21.F21.4:** Ca dezvoltator backend, vreau să implementez un endpoint GET /events/created pentru listarea evenimentelor create de utilizator.
- **P21.F21.5:** Ca dezvoltator frontend, vreau EventDetailsView pentru vizualizarea detaliilor cu posibilitatea modificării/anulării.
- **P21.F21.6:** Ca dezvoltator backend, vreau să implementez două endpoints PATCH /events/{id} și DELETE /events/{id}, pentru gestionarea evenimentului.
- **P21.F21.7:** Ca dezvoltator backend, vreau posibilitatea configurării unui număr maxim de participanți.

P22.F22: Invitarea utilizatorilor la evenimente

- **P22.F22.1:** Ca dezvoltator frontend, vreau selecție prieteni în EventDetailsView pentru trimiterea invitațiilor.
- **P22.F22.2:** Ca dezvoltator backend, vreau să implementez un endpoint POST /events/{id}/invite pentru trimiterea invitațiilor.
- **P22.F22.3:** Ca dezvoltator backend, vreau un sistem de trimitere notificări push la primirea invitațiilor la eveniment.
- **P22.F22.4:** Ca dezvoltator backend, vreau să implementez un endpoint GET /events/{id}/participants-status pentru returnarea statusului fiecărui invitat.

P23.F23: Puncte suplimentare pentru participarea la evenimente

- **P23.F23.1:** Ca dezvoltator frontend, vreau un câmp Bonus Points în EventCreateView pentru stabilirea recompensei.
- **P23.F23.2:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /events/{id}/bonus pentru configurarea punctelor bonus.
- **P23.F23.3:** Ca dezvoltator backend, vreau un job automat ce acordă punctele bonus după participare confirmată.
- **P23.F23.4:** Ca dezvoltator frontend, vreau o confirmare vizuală în ProfileView când utilizatorul primește bonusul.

P24.F24: Evenimente speciale cu criterii de eligibilitate

- **P24.F24.1:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /events/{id}/eligibility pentru configurarea criteriilor (ex: nivel puncte, top social).
- **P24.F24.2:** Ca dezvoltator backend, vreau un serviciu de validare automată a eligibilității la accesarea unui eveniment.
- **P24.F24.3:** Ca dezvoltator frontend, vreau un indicator vizibil cu mesajul eligibil sau neeligibil și motivul de (ne)eligibilitate în EventDetailsView.

P25.F25: Înregistrarea locațiilor de către deținători

- **P25.F25.1:** Ca dezvoltator frontend, vreau LocationCreateView cu câmpuri pentru nume, adresă, categorie și coordonate de pe hartă.
- **P25.F25.2:** Ca dezvoltator backend, vreau să implementez un endpoint POST /locations pentru înregistrarea unei locații.
- **P25.F25.3:** Ca dezvoltator backend, vreau să implementez un endpoint GET /locations/{id} pentru afișarea locației pe hartă și în detalii.

P26.F26: Promovarea ofertelor speciale în locații

- **P26.F26.1:** Ca dezvoltator frontend, vreau CampaignsManagementView pentru gestionarea ofertelor speciale la o locație.
- **P26.F26.2:** Ca dezvoltator backend, vreau să implementez un endpoint POST /locations/{id}/campaigns pentru creare campanie promoțională.
- **P26.F26.3:** Ca dezvoltator frontend, vreau afișarea campaniilor active în LocationDetailsView, pentru a atrage utilizatorii.

P27.F27: Locații ca puncte de interes (POI) pentru misiuni / evenimente

- **P27.F27.1:** Ca dezvoltator backend, vreau endpoint POST /locations/{id}/poi-request pentru ca deținătorii să poată solicita transformarea unei locații în POI.
- **P27.F27.2:** Ca administrator, vreau o componentă AdminPOIApprovalView pentru aprobarea sau respingerea cererii.
- **P27.F27.3:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH

/locations/{id}/poi-status pentru actualizarea statutului.

P28.F28: Gestionarea profilului public al locației

- **P28.F28.1:** Ca dezvoltator frontend, vreau LocationProfileView cu descriere, imagini și program afișate clar.
- **P28.F28.2:** Ca dezvoltator backend, vreau să implementez un endpoint PATCH /locations/{id} pentru actualizarea profilului unei locații.
- **P28.F28.3:** Ca dezvoltator frontend, vreau ReviewsSection cu posibilitatea de a adăuga evaluări și comentarii.
- **P28.F28.4:** Ca dezvoltator backend, vreau să implementez un endpoint POST /locations/{id}/reviews și unul GET /locations/{id}/reviews pentru gestionarea recenziilor.
- **P28.F28.5:** Ca dezvoltator frontend, vreau afișarea ratingului mediu și numărului total de recenzii în LocationDetailsView.

3.3 Specificații tehnologice

Flutter: framework utilizat pentru dezvoltarea interfeței aplicației tip mobile, permițând realizarea versiunilor pentru Android și iOS dintr-o singură bază de cod și contribuind la reducerea timpului de dezvoltare.

Python (FastAPI): utilizat pentru construirea service-ului de tip backend al aplicației și pentru implementarea endpoint-urilor necesare comunicării prin API-uri de tip REST.

MariaDB: sistemul de baze de date utilizat pentru stocarea și gestionarea datelor aplicației, oferind stabilitate și performanță ridicată.

Google Maps SDK & Geolocator: integrat pentru funcționalități de afișare a hărții și localizare, accesul la GPS realizându-se prin plugin-ul Flutter Geolocator.

Firebase Cloud Messaging (FCM): folosit pentru trimiterea notificărilor de tip push către utilizatori (invitații, confirmări, misiuni).

Sistem de autentificare securizat (Email / Parolă, OAuth Google / Facebook, JWT): implementate pentru gestionarea identității utilizatorilor, securizarea sesiunilor și criptarea comunicațiilor prin HTTPS, parolele fiind stocate în format hash.

Instrumente de dezvoltare software (VS Code, Android Studio, PyCharm, Git, GitHub): utilizate pentru implementarea codului, gestionarea versiunilor și colaborare în procesul de dezvoltare.

Instrumente de testare (Flutter Test, Pytest, Postman): folosite pentru verificarea funcționalităților din frontend, backend și API-uri, asigurând calitatea aplicației.

3.4 Componente de dezvoltare

Specificații		Componente
P1.F1	Vizualizarea prietenilor pe hartă	Frontend: MapView (Google Maps SDK, marker-e pentru prieteni), LocationMarker, Harta UI. Backend: LocationService, FriendService, GET /friends/locations (returnează locațiile prietenilor), PATCH /location/update (actualizează locația utilizatorului în timp real).
P2.F2	Propunerea unei întâlniri	Frontend: MeetingCreateView (selectare prieteni, locație, dată), MeetingListView, MeetingView (detalii întâlnire). Backend: MeetingsService, POST /meetings (creare întâlnire), GET /meetings (listare întâlniri), GET /meetings/{id} (detalii întâlnire).
P3.F3	Acceptarea și respingerea invitațiilor	Frontend: InvitationsListView, InvitationView, butoane Accept/Decline. Backend: InvitationService, GET /invitations (listare), PATCH /invitations/{id} (actualizare status), GET /invitations/{id} (detalii), FCM notificare organizator.
P4.F4	Organizarea grupurilor	Frontend: GroupsView, GroupView, butoane Add/Remove Members, Leave Group. Backend: GroupService, POST /groups (creare grup), PATCH /groups/{id}/members (modificare membri), GET /groups (listare), GET /groups/{id} (detalii grup).
P5.F5	Sistemul de puncte pentru întâlniri	Frontend: ProfileView (vizualizare total și istoric puncte promoționale). Backend: PointEngine, GET /points/summary, GET /points/history (istoric tranzacții).
P6.F6	Misiuni și provocări	Frontend: MissionsListView, MissionsHistoryView. Backend: MissionService, GET /missions (listă activă), GET /missions/history (istoric finalizări), FCM notificări la finalizare.
P7.F7	Cheltuirea punctelor bonus în magazinul digital	Frontend: StoreView (listă beneficii și costuri), StorePurchaseConfirmation (popup). Backend: StoreService, GET /store/items (listă beneficii), POST /store/purchase (procesare tranzacție).
P8.F8	Prieteni apropiați	Frontend: FriendsListView (toggle Apropiat, indicator vizual). Backend: FriendsService, PATCH /friends/{id}/close-status (actualizare relație).

P9.F9	Vizibilitate locație doar pentru Prieteni apropiați	Frontend: PrivacySettingsView – secțiune vizibilitate locație utilizator. Backend: LocationService (filtrare vizibilitate), GET /privacy/location-visibility (listă persoane care văd locația).
P10.F10	Dezactivarea partajării locației	Frontend: PrivacySettingsView – switch ON/OFF, indicator vizual status. Backend: LocationService, PATCH /location-sharing (actualizare setare), logică actualizare coordonate când locația e dezactivată.
P11.F11	Intervalul de actualizare a locației	Frontend: PrivacySettingsView – setare interval actualizare. Backend: LocationService, PATCH /location-update-interval (salvare opțiune), scheduler configurabil pentru actualizarea locației.
P12.F12	Status de disponibilitate	Frontend: ProfileView – dropdown status, afișare FriendsListView și MapView. Backend: AvailabilityService, PATCH /availability-status (actualizare status).
P13.F13	Interval orar de disponibilitate	Frontend: AvailabilityScheduleView (definire intervale recurente, vizual UI chips/calendar). Backend: AvailabilityService, POST /availability-schedule (salvare intervale).
P16.F16	Statistici și insight-uri social AI	Frontend: SocialInsightsView – dashboard grafice întâlniri și top locuri. Backend: AnalyticService, GET /analytics/meetings-history, GET /analytics/top-places, AI/ML module, GET /analytics/ai-insights (insight personalizat).
P17.F17	Top prieteni în funcție de socializare	Frontend: LeaderboardView (clasament prieteni). Backend: LeaderboardService, GET /leaderboard/friends (calcul și returnare clasament).
P21.F21	Crearea și gestionarea evenimentelor publice/private	Frontend: EventCreateView, EventDetailsView (vizualizare și editare), toggle Public/Privat. Backend: EventService, POST /events (creare), GET /events/created (listare), PATCH /events/{id}, DELETE /events/{id} (gestionare), setare limită de participanți.
P22.F22	Invitarea utilizatorilor la evenimente	Frontend: EventDetailsView – selecție prieteni. Backend: EventService, POST /events/{id}/invite (trimitere invitații), GET /events/{id}/participants-status (status invitați), notificări push FCM
P23.F23	Puncte suplimentare pentru participarea la evenimente	Frontend: EventCreateView – câmp Bonus Points, ProfileView confirmare primire puncte. Backend: EventService, PATCH /events/{id}/bonus, job automat acordare puncte.

P24.F24	Evenimente speciale cu criterii de eligibilitate	Frontend: EventDetailsView – indicator eligibilitate/motiv. Backend: EventService, PATCH /events/{id}/eligibility, EligibilityService – validare automată eligibilitate.
P25.F25	Înregistrarea locațiilor de către deținători	Frontend: LocationCreateView – câmpuri nume, adresă, categorie, coordonate. Backend: LocationService, POST /locations (înregistrare), GET /locations/{id} (detalii locație).
P26.F26	Promovarea ofertelor speciale în locații	Frontend: CampaignsManagementView, LocationDetailsView – afișare campanii. Backend: LocationService, POST /locations/{id}/campaigns (creare campanie).
P27.F27	Locații ca puncte de interes (POI) pentru misiuni / evenimente	Frontend: AdminPOIApprovalView (aprobări/respingeri). Backend: LocationService, POST /locations/{id}/poi-request, PATCH /locations/{id}/poi-status (actualizare statut POI).
P28.F28	Gestionarea profilului public al locației	Frontend: LocationProfileView (descriere, imagini, program), ReviewsSection, LocationDetailsView (rating mediu + număr recenzii). Backend: LocationService, PATCH /locations/{id} (actualizare profil), POST /locations/{id}/reviews, GET /locations/{id}/reviews (gestionare recenzii).

3.5 Componente de testare

Strategia generală de testare implică un proces continuu, incremental, ce va avea loc în fiecare SPRINT în paralel cu dezvoltarea funcționalităților.

Testare unitară: Atât pe backend, cât și pe frontend vor fi implementate teste unitare care să valideze buna funcționare a metodelor, a claselor și a comportamentelor logice definite. Fiecare dezvoltare nouă va fi însoțită de cel puțin un test unitar pentru a putea fi integrată în aplicației. Procentul final de code coverage trebuie să fie cel puțin egal cu 60%. Ca instrumente de testare se pot folosi flutter_test, mockito, mocktail pentru partea de frontend și pytest, unittest pentru partea de backend.

Testare de integrare: Sistemul de backend al aplicației va include teste de integrare care să verifice interacțiunea nivelurilor arhitecturale. Se va folosi o baza de date diferită de cea utilizată în implementarea sistemului, conținând date de test. Ca instrumente de testare se pot folosi pytest și HTTPX/requests, precum și o instanță de test MariaDB.

Testare funcțională: Fiecare funcționalitate nouă va fi testată imediat după integrarea sa. Pentru a marca un user story ca fiind finalizat, testarea funcțională a sa trebuie să fie finalizată cu succes. În cazul în care sunt identificate bug-uri, acestea trebuie rezolvate în funcție de prioritate. Dacă o funcționalitate este dezvoltată de către un dezvoltator, ceilalți

doi dezvoltatori sunt responsabili cu testarea acesteia. Ca instrumente de testare se poate folosi Postman pentru verificarea răspunsurilor API-urilor, restul testării fiind manuală.

Testare de acceptanță: Se va efectua la finalul fiecărui sprint, validându-se faptul că funcționalitățile implementate corespund specificațiilor de analiză. Aceasta va include verificarea corectitudinii interfeței, a comunicației cu backend-ul, a integrării serviciilor externe și a experienței utilizatorului final.

3.6 Componente de comunicare

Comunicare sincronă	
Sprint Planning	La începutul fiecărui Sprint se va ține o ședință de maxim o oră pentru definirea sarcinilor și responsabilităților fiecărui membru din echipă pentru următoarele două săptămâni. Aceasta include estimarea efortului pentru fiecare task și prioritizarea lor în backlog-ul sprintului. Când va avea loc: în prima săptămână a sprint-ului, în ziua de luni, ora 20:00
Weekly Stand-up	În timpul săptămânii se va ține o întâlnire scurtă de 10–15 minute în care se verifică progresul fiecărui membru din echipă și se rezolvă eventualele blocaje care au apărut pe parcurs. Scopul este menținerea transparenței și sincronizarea echipei. Când va avea loc: în prima săptămână a sprint-ului, în ziua de duminică, ora 12:00
Sprint Review	La final de Sprint se va ține o ședință de maxim 45 minute în care echipa dezvoltatorilor va prezenta un demo al aplicației mobile, echipa coordonatoare oferind un feedback rapid. Aceasta ajută la validarea obiectivelor sprintului și la ajustarea cerințelor pentru următorul sprint. Când va avea loc: în a doua săptămână a sprint-ului, în ziua de duminică, ora 20:00
Sprint Retrospective	Tot la final de Sprint se va ține o ședință de maxim 30 minute în care se vor identifica problemele apărute pe parcursul sprint-ului și se vor propune sugestii de îmbunătățire a colaborării. Scopul ședinței este creșterea eficienței și calității procesului de dezvoltare în echipă. Când va avea loc: în a doua săptămână a sprint-ului, în ziua de duminică, ora 21:00
Call-uri ad-hoc	La nevoie vor avea loc call-uri ad-hoc pentru a se rezolva diverse probleme urgente și a se debloca situațiile critice. Acestea asigură continuitatea lucrului și evitarea întârzierilor majore în sprint. Când vor avea loc: ori de câte ori este nevoie în timpul sprint-ului curent, în funcție de disponibilitatea membrilor

În ceea ce privește **comunicarea asincronă**, se va utiliza **Discord** pentru discuții rapide, anunțuri și clarificări. Monitorizarea task-urilor se va face prin intermediul **GitHub**. Fiecare membru va actualiza zilnic progresul task-urilor sale pe board.

3.7 Componente de învățare

În ceea ce privește învățarea individuală, fiecare membru are responsabilitatea de a studia documentație oficială și de a urmări tutoriale pentru utilizarea tehnologiilor de care are nevoie pentru finalizarea unui task. Acesta va împărtăși cu colegii resursele studiate, astfel încât fiecare membru al echipei să cunoască conceptele abordate în cadrul dezvoltării. În acest fel se asigură inclusiv calitatea code review-ului.

În fiecare sprint va avea loc o sesiune scurtă de maxim 30 – 40 minute în care mentorii vor clarifica conceptele noi studiate de echipa de dezvoltare. Toți colegii dezvoltatori vor pune întrebări și vor cere feedback referitor la codul scris. În cazul în care întâmpină dificultăți, se vor realiza la cerere sesiuni de pair-programming. Mentorii vor fi disponibili pe grupul de Discord pentru a răspunde la orice fel de nelămurire care apare în timpul sprint-ului și care blochează dezvoltările curente.

4. Perspectiva de dezvoltare

EPIC 1 (17 nov – 23 dec)	
Scop	Implementarea unui scenariu complet de interacțiune socială de bază între utilizatori (vizualizare prieteni pe hartă, localizare, propunere întâlniri, acceptare invitații).
SPRINT 1 (17 nov – 30 nov)	
Scop	Configurarea infrastructurii și vizualizarea prietenilor pe hartă.
Obiective specifice	<ul style="list-style-type: none"> • (P1.F1) Setup pentru proiectele de frontend Flutter și backend FastAPI, integrare MariaDB și Google Maps SDK. • (NF9) Implementarea autentificării de bază (email + parolă). • (P1.F1, P1.F1) Implementare endpoint-uri de bază pentru locație. • (P1.F1) Implementare componentă de frontend MapView cu marker-e pentru a afișa prieteni și locația lor curentă. • (P1.F1) Implementare componente de backend, incluzând FriendsService, endpoint-urile GET /friends/locations și PATCH /location/update.

Activități	<ul style="list-style-type: none"> • Dezvoltare: Crearea și configurarea proiectelor de frontend și backend, integrarea bazei de date MariaDB și a API-urilor de localizare. Implementarea modulului MapView și a serviciilor backend asociate. • Testare: Verificarea afișării corecte a locațiilor prietenilor pe hartă, actualizarea în timp real a poziției și testarea comunicației API între frontend și backend. • Învățare: Familiarizarea echipei cu Google Maps SDK, comunicarea REST API în Flutter, bune practici pentru gestionarea locației în timp real și fluxurile de date între servicii.
-------------------	---

SPRINT 2 (1 dec – 13 dec)	
Scop	Crearea întâlnirilor și gestionarea invitațiilor.
Obiective specifice	<ul style="list-style-type: none"> • (P2.F2) Implementare componentă de frontend MeetingCreateView pentru selectarea prietenilor, a locației și a datei întâlnirii. • (P2.F2, P3.F3) Implementare componente de frontend MeetingListView și MeetingView pentru vizualizarea listelor de întâlniri și detalii pentru întâlniri. • (P3.F3) Implementare componente de frontend InvitationsListView și InvitationView pentru vizualizarea, acceptarea și respingerea invitațiilor primite. • (P2.F2, P3.F3) Implementare componente de backend, incluzând: MeetingsService și InvitationsService cu endpoint-urile POST /meetings, GET /meetings, GET /meetings/{id}, GET /invitations, PATCH /invitations/{id} și WebSocket / FCM pentru notificări.
Activități:	<ul style="list-style-type: none"> • Dezvoltare: Crearea componentelor UI pentru întâlniri și invitații, integrarea backend-ului și a notificărilor push/WebSocket. • Testare: Validarea creării și vizualizării întâlnirilor, acceptarea și respingerea invitațiilor, testarea notificărilor către organizator. • Învățare: Schimb de bune practici privind UX-ul listelor și detaliilor de întâlnire, manipularea statusurilor invitațiilor și sincronizarea frontend-backend.

SPRINT 3 (14 dec – 23 dec)	
Scop	Organizarea grupurilor și implementarea sistemului de puncte bonus pentru întâlniri.
Obiective specifice	<ul style="list-style-type: none"> • (P4.F4) Implementare componente de frontend GroupsView și GroupView, butoane Add/Remove Members și Leave Group pentru gestionarea grupurilor utilizatorilor.

	<ul style="list-style-type: none"> • (P4.F4) Implementare componente de backend, incluzând: GroupsService cu endpoint-urile POST /groups, PATCH /groups/{id}/members, GET /groups, GET /groups/{id} pentru gestionarea grupurilor utilizatorilor. • (P5.F5) Implementare afișare puncte în profilul utilizatorului prin actualizarea UI-ului ProfileView. • (P5.F5) Implementare componente de backend, incluzând: PointsEngine, GET /points/summary și GET /points/history pentru gestionarea punctelor și a istoricului acestora.
Activități:	<ul style="list-style-type: none"> • Dezvoltare: Crearea interfețelor pentru grupuri și puncte bonus, implementarea serviciilor backend și integrarea cu bazele de date. • Testare: Verificarea gestionării membrilor grupurilor, afișarea corectă a punctelor și istoricului tranzacțiilor, validarea endpoint-urilor backend. • Învățare: Discuții privind logica de gamificare, strategii de afișare a datelor de profil și bune practici pentru managementul grupurilor.

EPIC 2 (8 ian – 6 feb)	
Scop	Extinderea funcționalităților sociale și introducerea elementelor gamificate și a evenimentelor publice/private.

SPRINT 4 (8 ian – 16 ian)	
Scop	Implementare misiuni, magazin digital și prieteni apropiați.
Obiective specifice	<ul style="list-style-type: none"> • (P6.F6) Implementare componente de frontend MissionsListView și MissionsHistoryView pentru afișarea misiunilor active și istoricul misiunilor finalizate. • (P6.F6) Implementare componente de backend, incluzând: MissionsService cu endpoint-urile GET /missions și GET /missions/history și notificări push la finalizarea misiunilor. • (P7.F7) Implementare componente de frontend StoreView și StorePurchaseConfirmation pentru vizualizarea itemelor digitale și confirmarea achiziției acestora. • (P7.F7) Implementare componente de backend, incluzând: StoreService cu endpoint-urile GET /store/items și POST /store/purchase pentru gestionarea tranzacțiilor pe puncte. • (P8.F8) Implementare UI în FriendsListView pentru activarea modului „apropiat” și afișarea indicatorului vizual. • (P8.F8) Implementare componente de backend, incluzând: FriendsService cu endpoint-ul PATCH /friends/{id}/close-status.

Activități	<ul style="list-style-type: none"> • Dezvoltare: Crearea componentelor pentru misiuni și magazin, integrarea backend-ului și logica de puncte pentru prietenii apropiați. • Testare: Validarea funcționalităților ce includ misiuni, tranzacții digitale și statutul de prieten apropiat. • Învățare: Familiarizarea cu strategii de gamificare.
-------------------	--

SPRINT 5 (17 ian – 24 ian)	
Scop	Setări de confidențialitate și disponibilitate.
Obiective specifice	<ul style="list-style-type: none"> • (P9.F9, P10.F10) Implementare componente de frontend PrivacySettingsView pentru configurarea vizibilității locației, dezactivarea partajării și indicator vizual de stare. • (P11.F11) Implementare opțiuni în PrivacySettingsView pentru setarea intervalului de actualizare a locației. • (P12.F12, P13.F13) Implementare componente de frontend ProfileView și AvailabilityScheduleView pentru setarea statusului de disponibilitate și a intervalelor orare. • (P9.F9 – P13.F13) Implementare componente de backend, incluzând: LocationService și AvailabilityService pentru controlul vizibilității locației, intervalului de actualizare și statusului utilizatorului.
Activități:	<ul style="list-style-type: none"> • Dezvoltare: Crearea interfețelor de setări, integrarea serviciilor backend și logica de filtrare a vizibilității și programului de disponibilitate. • Testare: Validarea modificării vizibilității locației, a intervalelor de actualizare, a statusului și a afișării acestora în UI și pe hartă. • Învățare: Discuții despre securizarea datelor de locație, gestionarea preferințelor utilizatorilor și bune practici pentru sincronizarea frontend-backend.

SPRINT 6 (25 ian – 6 feb)	
Scop	Evenimente publice/private, bonusuri și profiluri de locații.
Obiective specifice	<ul style="list-style-type: none"> • (P16.F16) Implementare componente de frontend SocialInsightsView pentru afișarea dashboard-ului cu statistici și insight-uri sociale. • (P16.F16) Implementare componente de backend, incluzând: AnalyticsService cu endpoint-urile GET /analytics/meetings-history, GET /analytics/top-places și GET /analytics/ai-insights pentru statistici și insight-uri AI personalizate. • (P17.F17) Implementare componente de frontend LeaderboardView pentru afișarea clasamentului prietenilor după activitate socială.

	<ul style="list-style-type: none"> • (P17.F17) Implementare componente de backend, incluzând: LeaderboardService cu endpoint-ul GET /leaderboard/friends pentru calcularea și returnarea clasamentului. • (P21.F21–P23.F23) Implementare componente de frontend EventCreateView, EventDetailsView pentru crearea și vizualizarea evenimentelor, inclusiv setarea bonusurilor și invitațiilor. • (P21.F21–P23.F23) Implementare componente de backend, incluzând: EventsService cu endpoint-urile POST /events, GET /events/created, PATCH /events/{id}, DELETE /events/{id}, POST /events/{id}/invite, GET /events/{id}/participants-status și PATCH /events/{id}/bonus pentru gestionarea completă a evenimentelor, invitațiilor și punctelor bonus. • (P24.F24) Implementare backend, incluzând EventsService cu endpoint-ul PATCH /events/{id}/eligibility și un serviciu de validare automată pentru criteriile de eligibilitate. • (P25.F25–P28.F28) Implementare componente de frontend LocationCreateView, LocationProfileView și ReviewsSection pentru înregistrarea locațiilor, vizualizarea profilului și recenzii. • (P25.F25–P28.F28) Implementare componente de backend, incluzând LocationService cu endpoint-urile POST /locations, GET /locations/{id}, POST /locations/{id}/campaigns, POST /locations/{id}/poi-request, PATCH /locations/{id}/poi-status, PATCH /locations/{id}, POST /locations/{id}/reviews și GET /locations/{id}/reviews pentru gestionarea completă a locațiilor, campaniilor, POI-urilor și recenziilor.
Activități:	<ul style="list-style-type: none"> • Dezvoltare: Finalizarea interfețelor pentru evenimente și locații, integrarea completă a serviciilor backend și afișarea datelor AI/social insights. • Testare: Validarea fluxului complet de creare și participare la evenimente, aplicarea criteriilor de eligibilitate, acordarea bonusurilor și gestionarea recenziilor. • Învățare: Împărtășirea experienței privind integrarea funcționalităților complexe, folosirea insight-urilor AI și afișarea datelor agregate în dashboard.