



Universitatea
Transilvania
din Braşov

FACULTATEA DE MATEMATICĂ
ŞI INFORMATICĂ

Chat App for students

Team: Lagarul ∞

Members:

- Apriotese Andreea
- Arhip Florin
- Călăvie Adrian-Constantin
- Dinu Ionuţ-Alin

Cuprins

1. INTRODUCERE.....	3
1.1. Functionalitatea aplicatiei.....	3
1.2. Etapele proiectarii aplicatiei.....	3
2. Tehnologii folosite.....	4
2.1. Limbaj de programare.....	4
2.2. Mediu de dezvoltare.....	4
2.3. Versionare.....	4
2.4. Managementul task-urilor.....	5
2.5. Crearea diagramelor.....	5
2.6. Server baza de date.....	5
3. Diagrame.....	6
3.1. Diagrama cazurilor de utilizare.....	6
3.2. Diagrama de clase.....	7
3.3. Diagrama bazei de date.....	8
3.4. Diagrama de stare.....	9
3.5. Diagrama de colaborare.....	9
3.6. Diagrama de activitati.....	10
3.7. Diagrama de interactiuni.....	11
3.8. Diagrama de componente.....	12
3.9. Diagrama de pachete.....	13
3.10. Diagrama de desfasurare.....	14

1. INTRODUCERE

1.1. Functionalitatea aplicatiei

Scopul creării acestei aplicații a fost acela de a facilita comunicarea dintre studenții ai Universitatii Transilvania din Brasov, creand un canal oficial de comunicare dintre studenții aceleiași facultăți, aceleiași specializari sau chiar aceleiași grupe. Astfel, anunturile oficiale, mesajele de interes la nivel de specializare, sau grupa fiind intr-un singur loc si accesibil oricui are o adresa de e-mail institutionala.

1.2. Etapele proiectării aplicației

- **Identificarea problemei de rezolvat** - Problema de care ne-am lovit aproape fiecare dintre noi la începutul studiului universitar a fost sa găsim diverse informații cu privință la facultate, materii, acte sau chiar sa ne gasim colegii. Asadar, ne-am gandit la o aplicație oficială a universității unde studenții pot comunica mai ușor, fiind create în prealabil și grupuri pentru fiecare grupa, specializare sau facultate.
- **Specificarea cerințelor** - Una dintre cele mai importante cerințe și cele mai esentiale este aceea prin care doar utilizatorilor cu adresa de e-mail instituțională poate folosi aplicatia noastra, fiind un canal oficial de comunicare între studenții și profesorii Universității Transilvania din Brașov.
- **Analiza** - În etapa de analiză am identificat caracteristicile sistemului existent și am identificat elementele ce trebuie îmbunătățite. Exista deja un sistem de mesaje integrat in platforma e-learning, dar nu este specializat pe grupuri și pe facultăți, motiv pentru care am ales sa facem o aplicatie dedicata pentru comunicare.
- **Proiectare** - În faza de proiectare am stabilit structura aplicației până la cel mai mic detaliu. De la tehnologiile folosite, la structura bazei de date, structura claselor model, functiile, procesele și designul aplicației.
- **Implementare** - În etapa de implementare ne-am impartit task-urile folosind Trello, pe back-end si pe front-end, urmand ca fiecare din echipa sa se ocupe de implementarea acestora. Am început cu partea de login si register urmand ca apoi sa ne ocupăm de restul aplicației luand datele din Firestore specifice fiecarui fragment.

2. TEHNOLOGIILE FOLOSITE

2.1. Limbajul de programare

În realizarea proiectului ne-am folosit de limbajul de programare cross-platform **Kotlin**, care a fost proiectat pentru a interopera pe deplin cu Java, iar versiunea JVM a bibliotecii standard a lui Kotlin depinde de Java Class Library, dar inferența de tip permite sintaxa să fie mai concisă. Costurile de dezvoltare a limbajului sunt suportate de JetBrains, în timp ce Fundația Kotlin protejează marca Kotlin. La 7 mai 2019, Google a anunțat că limbajul de programare Kotlin este limbajul preferat de programare al dezvoltatorilor de aplicații Android.

2.2. Mediu de dezvoltare

Android Studio este un mediu de dezvoltare (engl. software development environment, sau integrated development environment - "mediu integrat de dezvoltare) pentru colaborarea cu platforma Android, anunțată pe data de 16 mai 2013 în cadrul conferinței I / O Google. Android Studio este bazat pe software-ul IntelliJ IDEA de la JetBrains, este instrumentul oficial de dezvoltare a aplicațiilor Android. Acest mediu de dezvoltare este disponibil pentru Windows, OS X și Linux. Pe 17 mai 2017, la conferința anuală Google I / O, Google a anunțat asistență pentru limbajul Kotlin utilizat de Android Studio ca limbaj de programare oficial pentru platforma Android, pe lângă Java și C ++.

2.3. Versionare

GitHub este un serviciu de gazduire web pentru proiecte de dezvoltare a software-ului care utilizează sistemul de control al versiunilor Git. GitHub oferă planuri tarifare pentru depozite private, și conturi gratuite pentru proiecte open source. Site-ul a fost lansat în 2008 de către Tom Preston-Werner, Chris Wanstrath, și PJ Hyett. În 2018 Microsoft a cumpărat Github pentru 7.5 miliarde de dolari.

2.4. Managementul Task-urilor

Pentru un management mai bun al task-urilor ne-am folosit de **Trello**, o aplicație web cu un design intuitiv pentru crearea task-urilor, împărțirea pe categorii a acestora între membrii echipei, pentru a facilita întreg procesul de creare a unei aplicații, dar poate fi folosit și pentru alte scopuri precum to do list, planificarea lectiilor sau a temelor, contabilitate, web design si multe altele.

2.5. Crearea diagramelor

Pentru desenarea diagramelor am folosit **draw.io**, un software de desenare open source gratuit dezvoltat in HTML5 si JavaScript. Interfața sa poate fi folosită pentru a crea diagrame, organigrame, wireframes, diagrame UML, și diagrame de rețea. De asemenea, se integrează cu servicii cloud pentru stocare, inclusiv Dropbox, OneDrive, Google Drive, GitHub și GitLab.com.

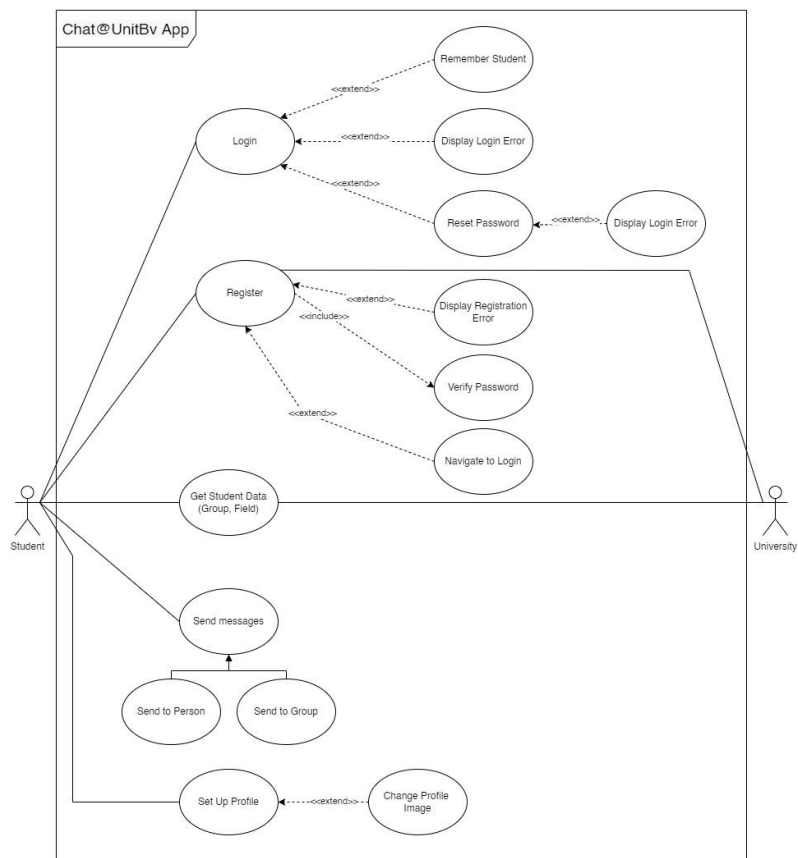
2.6. Server baza de date

Pentru stocarea datelor aplicației noastre ne-am folosit de **Cloud Firestore**, o bază de date NoSQL găzduită în cloud pe care aplicațiile Apple, Android și web o pot accesa direct prin SDK-uri native. Cloud Firestore este disponibil și în SDK-urile native Node.js, Java, Python, Unity, C++ și Go, pe lângă API-urile REST și RPC.

Urmând modelul de date NoSQL al Cloud Firestore, datele sunt stocate în documente care conțin câmpuri asociate cu valori. Aceste documente sunt stocate în colecții, care sunt containere pentru documente, pe care le vom utiliza pentru a ne organiza datele și a crea interogari. Documentele acceptă multe tipuri de date diferite, de la șiruri și numere simple până la obiecte complexe, imbricate. De asemenea, putem crea subcolecții în documente și putem construi structuri de date ierarhice care să se extindă pe măsură ce baza de date crește.

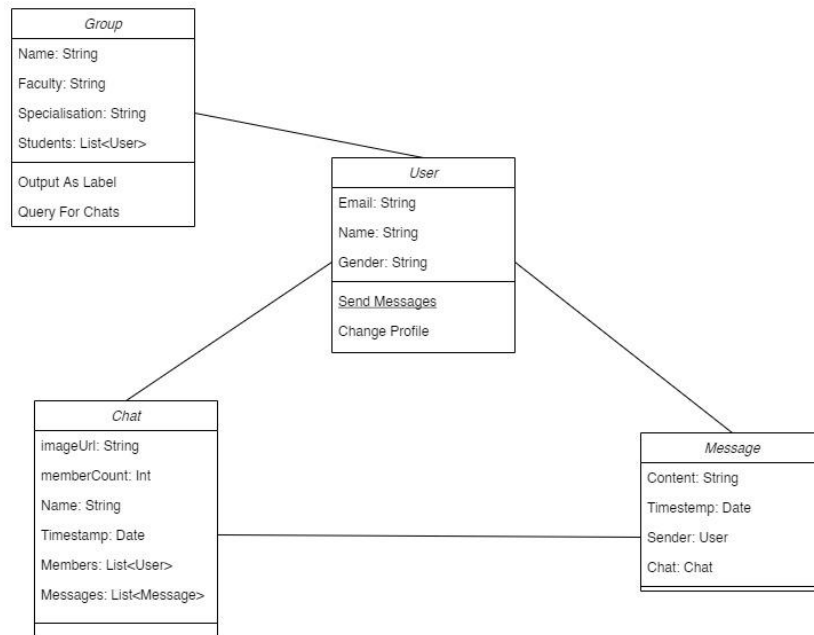
3. DIAGRAME

3.1. Diagrama cazurilor de utilizare

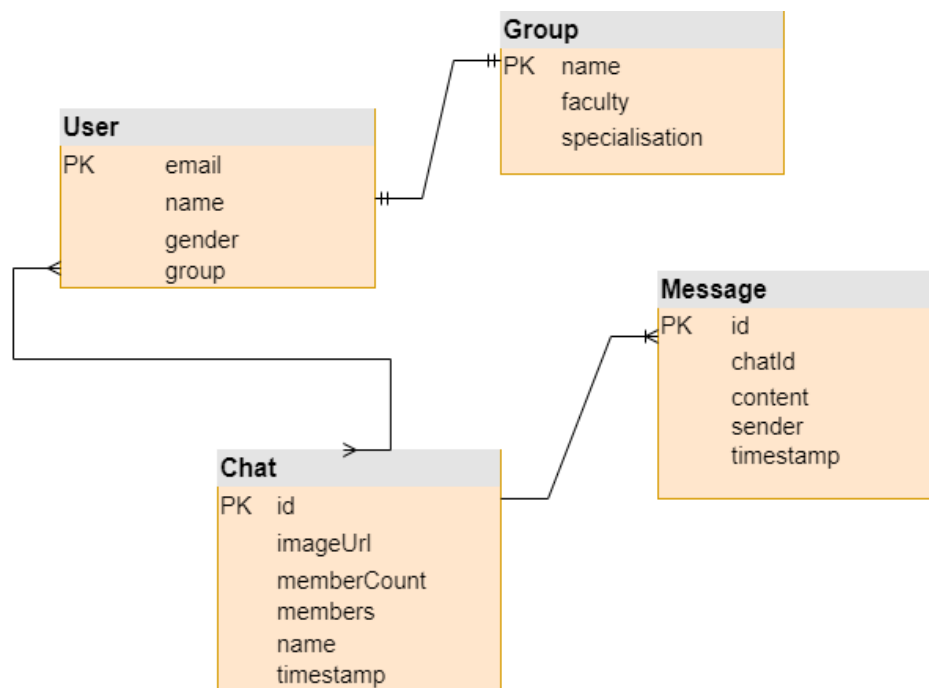


3.2. Diagrama de clase

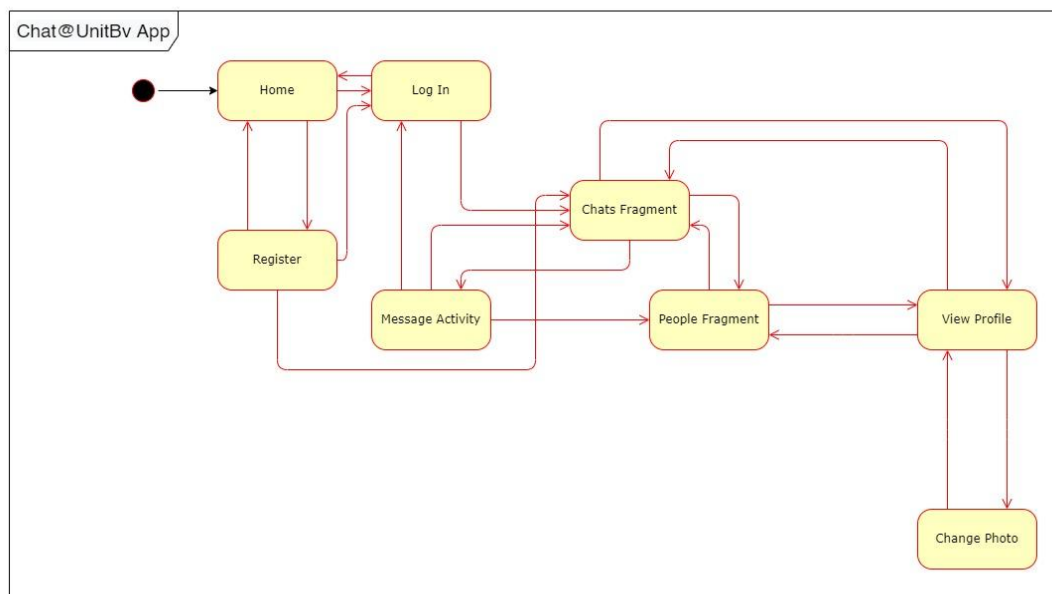
Class Diagram



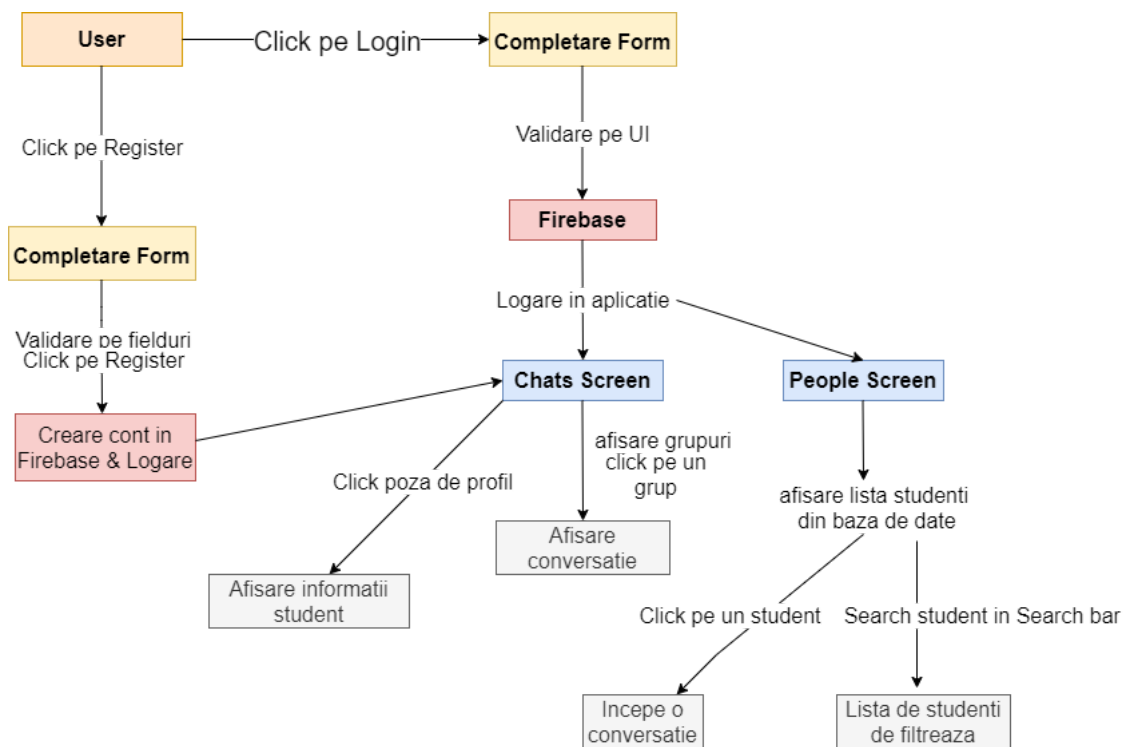
3.3. Diagrama bazei de date



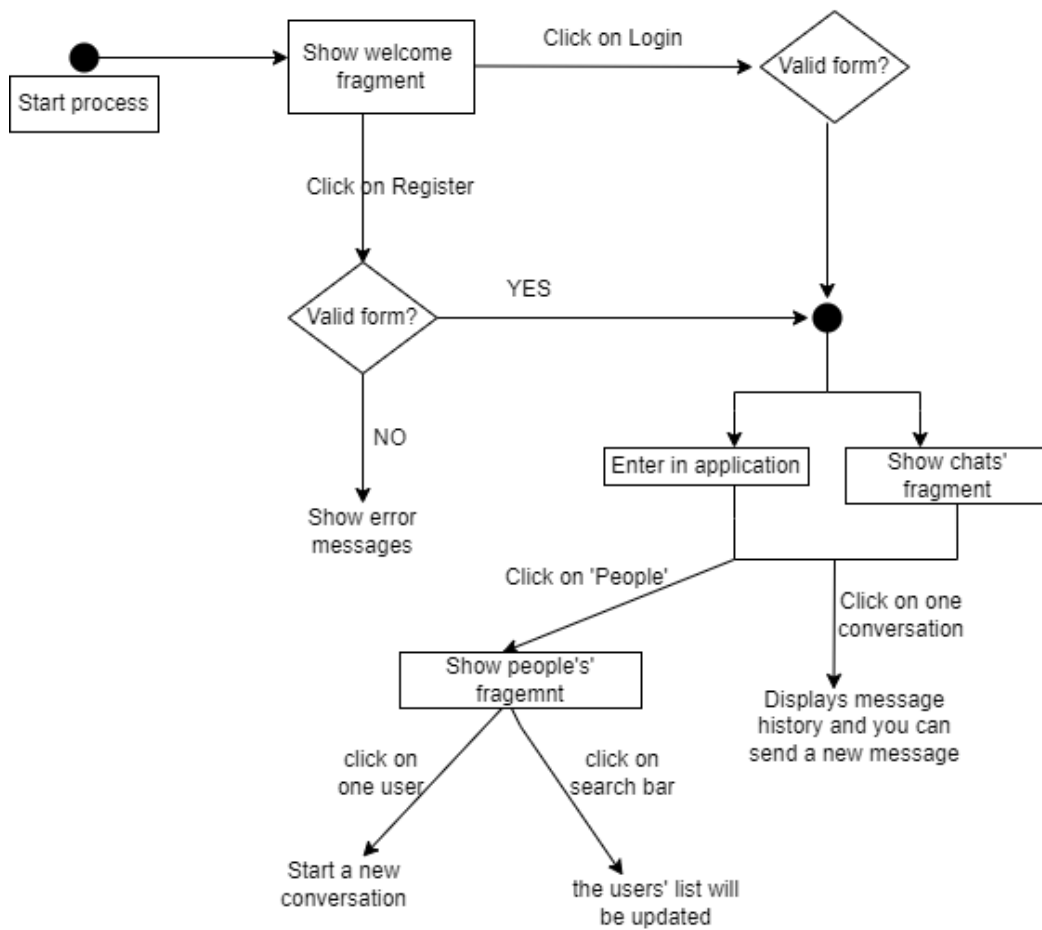
3.4. Diagrama de stare



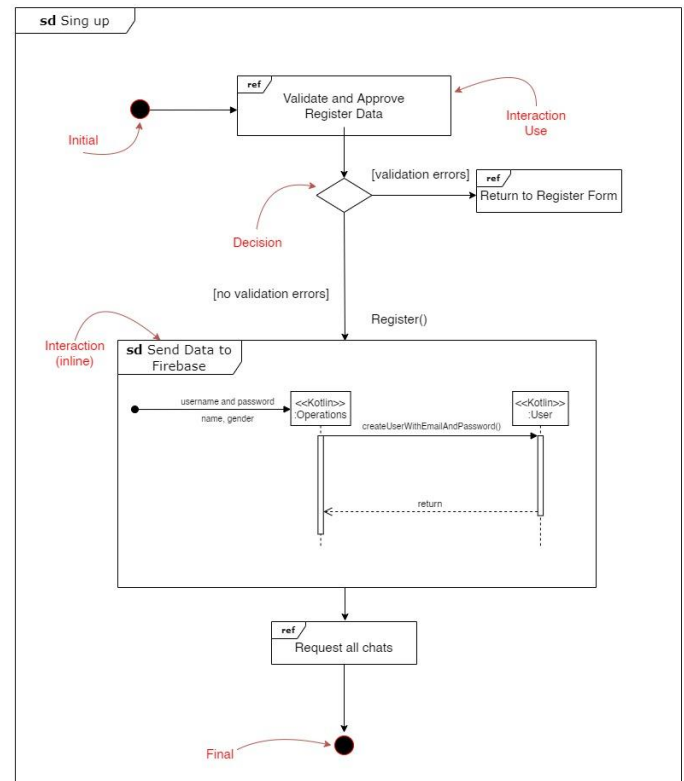
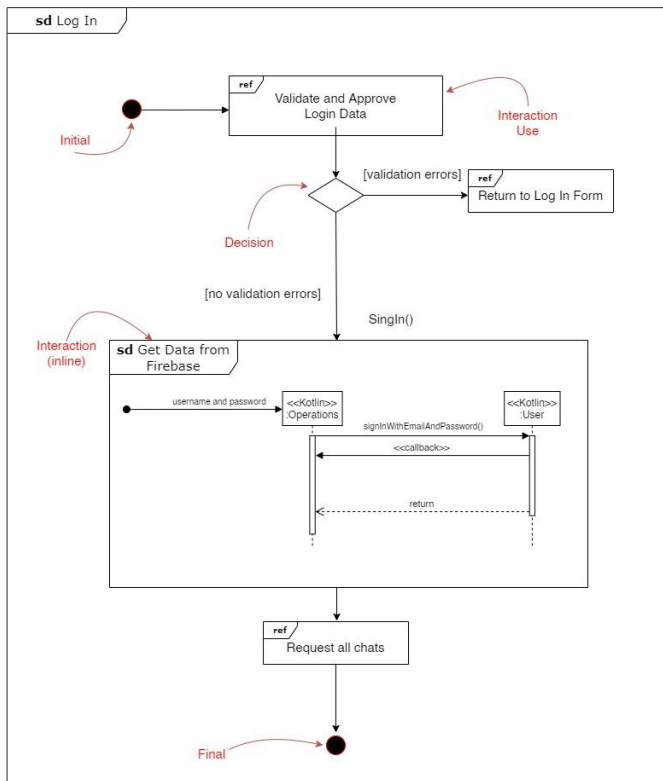
3.5. Diagrama de colaborare



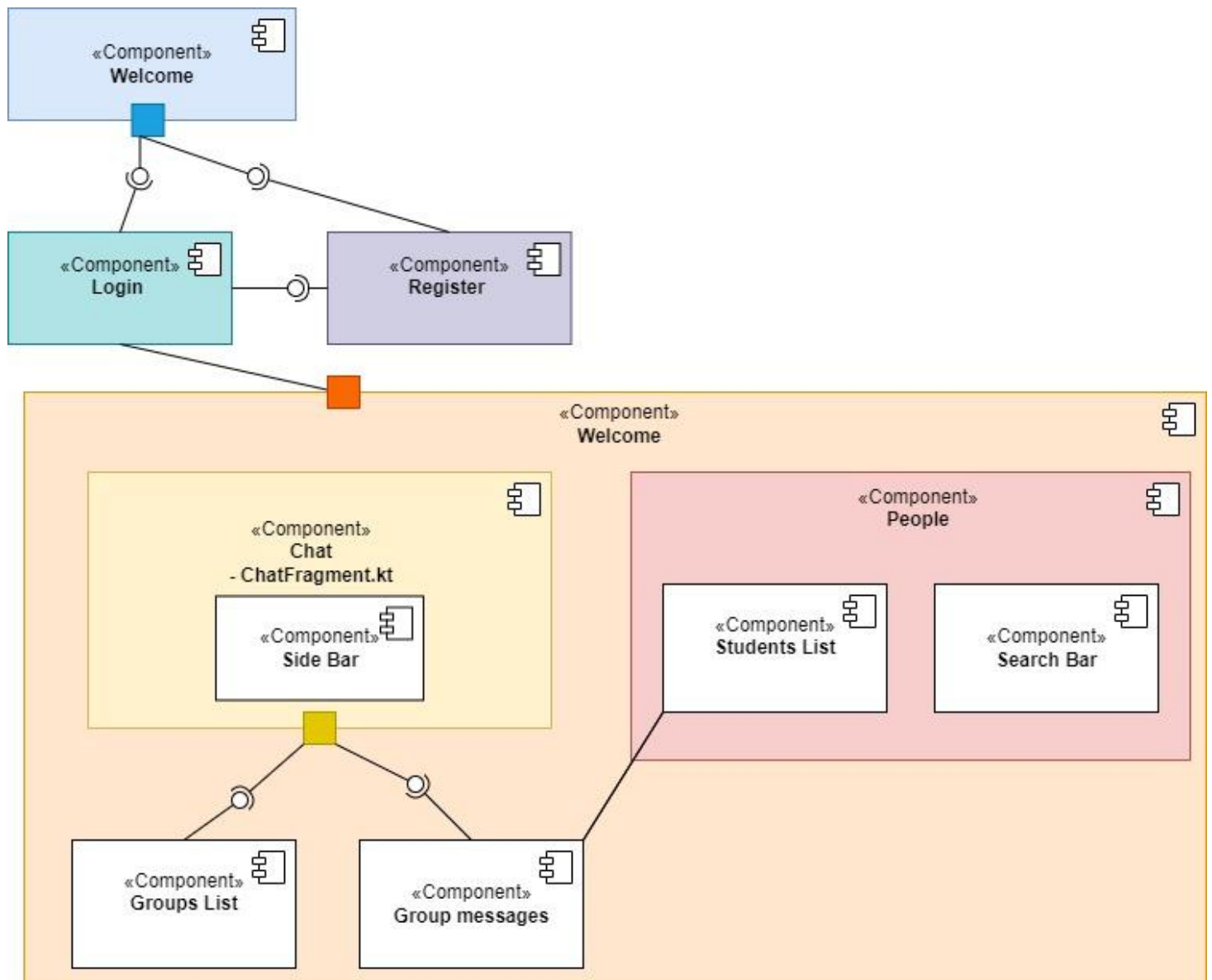
3.6. Diagrama de activități



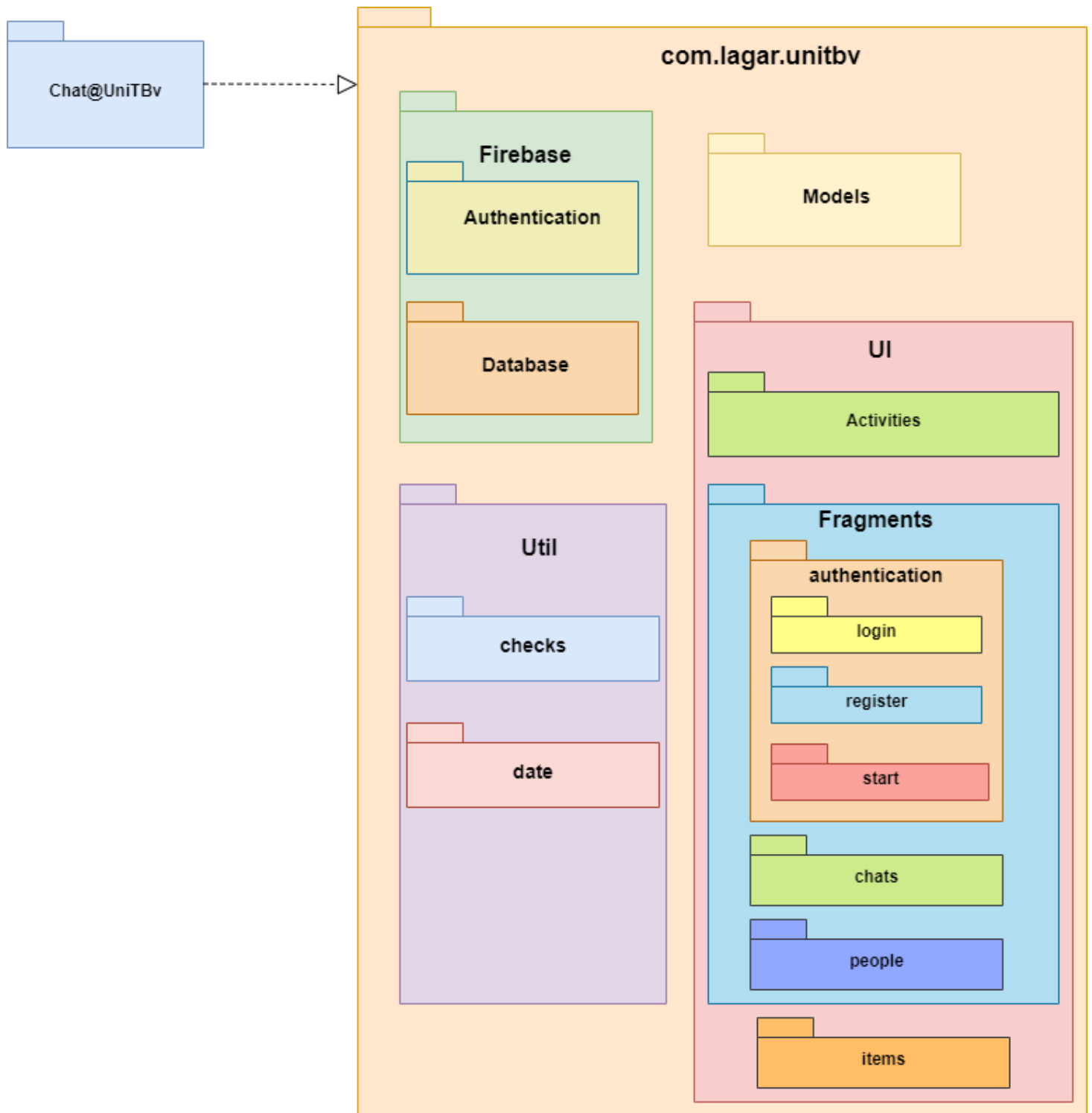
3.7. Diagrama de interacțiune



3.8. Diagrama de componente



3.9. Diagrama de pachete



3.10. Diagrama de desfășurare

Chat@UnitBv
Deployment Diagram

