UNIVERSITATEA BABES-BOLAY

# Facultatea de Matematica si Informatica

# Sectia Informatica

Monitorizare angajati -proiect final ISS

Nume student: Man Maria

Grupa: 224

Profesor indrumator: Triban Paul

Cerinte generale:

Se cere realizarea unei aplicatii (desktop sau web) cu arhitectura stratificata (presentation, business logic, data access), conform cerintelor problemei asignate/alese. Limbajul de implementare va fi un limbaj orientat obiect, la alegere dintre cele deja studiate. Interfata cu utilizatorul se recomanda a fi realizata cu un instrument vizual (ex. NetBeans, WindowBuilder, Visual Studio). Se cere aplicarea sablonului arhitectural MVC (Model-View-Controller). Aplicatia trebuie sa exploateze o baza de date relationala prin stratul de data access. Se impune ca interactiunea cu baza de date sa se faca prin intermediul unei biblioteci ORM (Object-Relational Mapping, ex. Hibernate/EclipseLink, NHibernate/Entity Framework). Proiectul trebuie sa ofere functii de tip CRUD (Create-Read-UpdateDelete) pentru cel putin o entitate din domeniul problemei (clasa din Model). Obligatoriu, toate modelele UML ale aplicației (functional, structurale, comportamentale) vor fi construite cu ajutorul unui instrument CASE (Computer Aided Software Engineering, ex. StarUML) și vor fi incluse în cadrul aceluiasi proiect, structurat adecvat, conform etapelor ciclului de viața al unui sistem soft (pentru a putea urmări aspecte legate de sincronizare). Implementarea claselor din pachetul Model poate fi realizata partial folosind o operatie de tip forward/direct engineering, aplicata cu ajutorul instrumentului CASE folosit.

FAZA I (Colectarea si analiza cerintelor)

Laboratorul 1

• momente organizatorice, stabilirea temelor individuale

Laboratorul 2

• modelul functional (diagrama UML a cazurilor de utilizare + descrierea textuala/tabelara a cazurilor de utilizare, cu scenariul normal si scenariile alternative/de exceptie posibile)

• planificarea cazurilor de utilizare pe 3 iteratii

Laboratorul 3

• modelul conceptual (reprezentat folosind o diagrama UML de clase) ◦ !!! modelul conceptual reda exclusiv entitatile din domeniul problemei si relatiile aferente

• prototipul interfetei grafice cu utilizatorul

FAZA II (Proiectare si implementare iteratia 1)

Laboratorul 4

• reprezentarea scenariilor aferente cazurilor de utilizare din iteratia 1, prin diagrame de interactiune (diagrame de secventa si de comunicare/colaborare)

• rafinarea diagramei de clase initiale, cu includerea entitatilor din domeniul solutiei aferente iteratiei 1

Laboratorul 5

• implementare iteratia 1: prima versiune functionala a aplicatiei

FAZA III (Proiectare si implementare iteratii 2, 3)

Laboratorul 6

• reprezentarea scenariilor aferente cazurilor de utilizare din iteratia 2, prin diagrame de interactiune (diagrame de secventa si de comunicare/colaborare)

• rafinarea diagramei de clase initiale, cu includerea entitatilor din domeniul solutiei aferente iteratiei 2

• implementare iteratia 2: a doua versiune functionala a aplicatiei

Laboratorul 7

• reprezentarea scenariilor aferente cazurilor de utilizare din iteratia 3, prin diagrame de interactiune (diagrame de secventa si de comunicare/colaborare)

• rafinarea diagramei de clase initiale, cu includerea entitatilor din domeniul solutiei aferente iteratiei 3

• iteratia 3: versiunea finala a aplicatiei

Cerinte specifice:

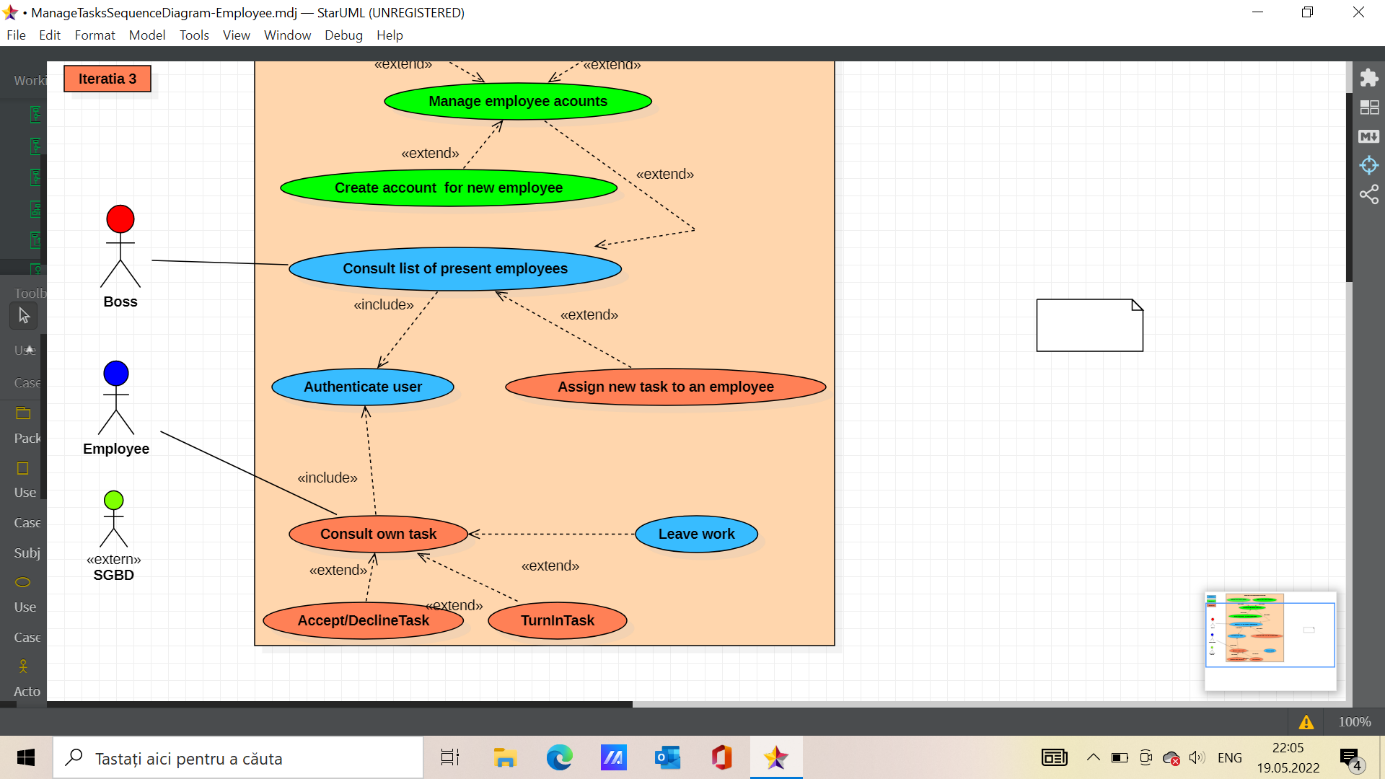
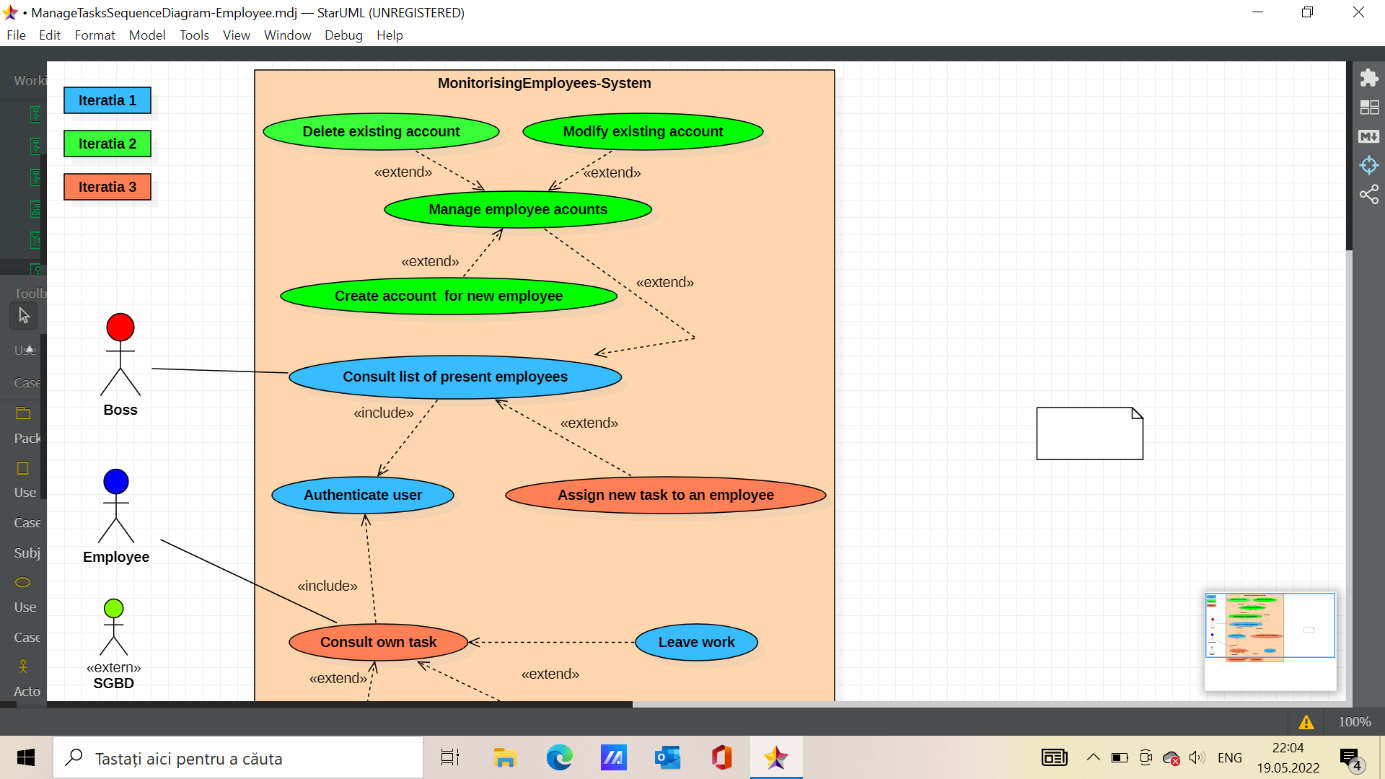
4. MONITORIZARE ANGAJATI

O firma si-a creat o infrastructura prin care seful monitorizeaza angajatii prezenti la lucru si le traseaza sarcini individuale. Firma are o aplicatie care ofera:

• **o fereastra pentru sef**, cu ajutorul careia seful vede lista angajatilor prezenti în firma, un element din lista precizând numele angajatului si ora la care s-a logat în sistem. De asemenea, seful poate transmite o sarcina unui angajat prezent astfel: selecteaza angajatul din lista, introduce o descriere a sarcinii si declanseaza un buton "transmite sarcina". Imediat dupa transmiterea unei sarcini, aceasta poate fi consultata de catre angajatul respectiv.

**• câte o fereastra pentru fiecare angajat**: Atunci când angajatul vine la serviciu, introduce ora sosirii si declanseaza un buton "prezent". Imediat dupa declansarea butonului, seful vede în lista lui ca angajatul este prezent. În continuare, cât timp angajatul sta la serviciu, el primeste si, în consecinta, vede în fereastra lui, sarcinile transmise de sef. La plecare, angajatul închide fereastra, moment în care seful este notificat de delogarea acestuia din sistem.

Diagrama **cazurilor de utilizare**:



Descrierea **cazurilor de utilizare**:

**Monitorizare angajați**

**Descrierea cazurilor de utilizare**

Proiectant:Man Maria-224/2

**Autentificare**

|  |  |
| --- | --- |
| **Nume** | Autentificare |
| **Participanți** | Angajat  Șef |
| **Condiții de intrare** | Cont de angajat existent |
| **Condiții de ieșire** | Angajatul primește un răspuns de stadiu al autentificării (reușită, nereușită) |
| **Cerințe de calitate** | Răspunsul este trimis în maxim 10 secunde |

|  |  |
| --- | --- |
| **Flux de evenimente** | |
| **Pas** | **Descriere** |
| **1** | Angajatul introduce ora sosirii la servici și acționează butonul prezent |
| **2** | Lista de angajați prezenți din fereastra șefului este actualizată |
| **3** | Angajatul închide fereastra și pleacă de la servici |
| **4** | Lista de angajați prezenți din fereastra șefului este actualizată |

**Adăugare cont**

|  |  |
| --- | --- |
| **Nume** | Adăugare cont |
| **Participanți** | Șef |
| **Condiții de intrare** | Șeful este logat |
| **Condiții de ieșire** | Contul a fost adăugat sau adăugarea a fost anulată |
| **Cerințe de calitate** | Adăugarea unui cont se realizează în maxim 5 secunde și lista conturilor este actualizată |

|  |  |
| --- | --- |
| **Flux de evenimente** | |
| **Pas** | **Descriere** |
| **1** | Șeful selecteaza opțiunea de adăugare cont |
| **2** | Sistemul afișează câmpurile obligatorii de completat |
| **3** | Șeful completează câmpurile |
| **4** | Șeful apasă butonul de finalizare |
| **5** | Sistemul validează datele și afișează un mesaj corespunzător |

**Modificare cont**

|  |  |
| --- | --- |
| **Nume** | Modificare cont |
| **Participanti** | Șef |
| **Conditii de intrare** | Șeful este logat și există un cont selectat |
| **Conditii de iesire** | Contul a fost modificat sau modificarea a fost anulată |
| **Cerinte de calitate** | Modificarea unui cont se realizează în maxim 5 secunde și lista conturilor este actualizată |

|  |  |
| --- | --- |
| **Flux de evenimente** | |
| **Pas** | **Descriere** |
| **1** | Șeful selectează opțiunea de modificare cont |
| **2** | Sistemul afișează câmpurile care pot fi modificate |
| **3** | Șeful completează câmpurile pe care dorește să le modifice |
| **4** | Șeful apasă butonul de finalizare |
| **5** | Sistemul validează datele, modifică atributele ale căror câmpuri au fost completate și le păstrează pe celelalte, iar în urma evaluării se afișează un mesaj corespunzător |

**Șterge cont**

|  |  |
| --- | --- |
| **Nume** | Șterge cont |
| **Participanți** | Șef |
| **Condiții de intrare** | Șeful este logat și există un cont selectat |
| **Condiții de ieșire** | Contul a fost șters |
| **Cerințe de calitate** | Ștergerea unui cont se realizează în maxim 5 secunde și lista conturilor este actualizată |

|  |  |
| --- | --- |
| **Flux de evenimente** | |
| **Pas** | **Descriere** |
| **1** | Șeful selectează un cont din lista curentă |
| **2** | Șeful selectează opțiunea de ștergere cont |
| **3** | Sistemul șterge contul selectat |
| **4** | Sistemul afișează un mesaj corespunzător |
| **5** | Sistemul actualizează lista conturilor |

**Transmite o sarcină nouă**

|  |  |
| --- | --- |
| **Nume** | Transmite o sarcină nouă angajatului selectat |
| **Participanți** | Șef |
| **Conditii de intrare** | Șeful este logat și există un cont selectat |
| **Conditii de ieșire** | Angajatul selectat primește noua sarcină |
| **Cerințe de calitate** | Actualizarea listei de sarcini se realizează în maxim 5 secunde |

|  |  |
| --- | --- |
| **Flux de evenimente** | |
| **Pas** | **Descriere** |
| **1** | Șeful selectează opțiunea de autentificare |
| **2** | Șeful consultă lista angajaților |
| **3** | Șeful selectează un anagajat prezent din listă |
| **4** | Șeful introduce descrierea unei noi sarcini în căsuța de text aferentă |
| **5** | Șeful trimite noua sarcină anagajatului selectat prin apăsarea butonului “Send task” |

**Părăsirea locului de muncă**

|  |  |
| --- | --- |
| **Nume** | Părăsirea locului de muncă |
| **Participanți** | Angajat |
| **Conditii de intrare** | Angajatul este logat |
| **Conditii de ieșire** | Angajatul selectat nu mai este prezent |
| **Cerințe de calitate** | Actualizarea listei de angajați prezenți se realizează în maxim 5 secunde |

|  |  |
| --- | --- |
| **Flux de evenimente** | |
| **Pas** | **Descriere** |
| **1** | Angajatul selectează opțiunea de autentificare |
| **2** | Angajatul introduce ora sosirii la servici și acționează butonul prezent |
| **3** | Angajatul închide fereastra și pleacă de la servici |

Diagrama de clase a modelului conceptual:

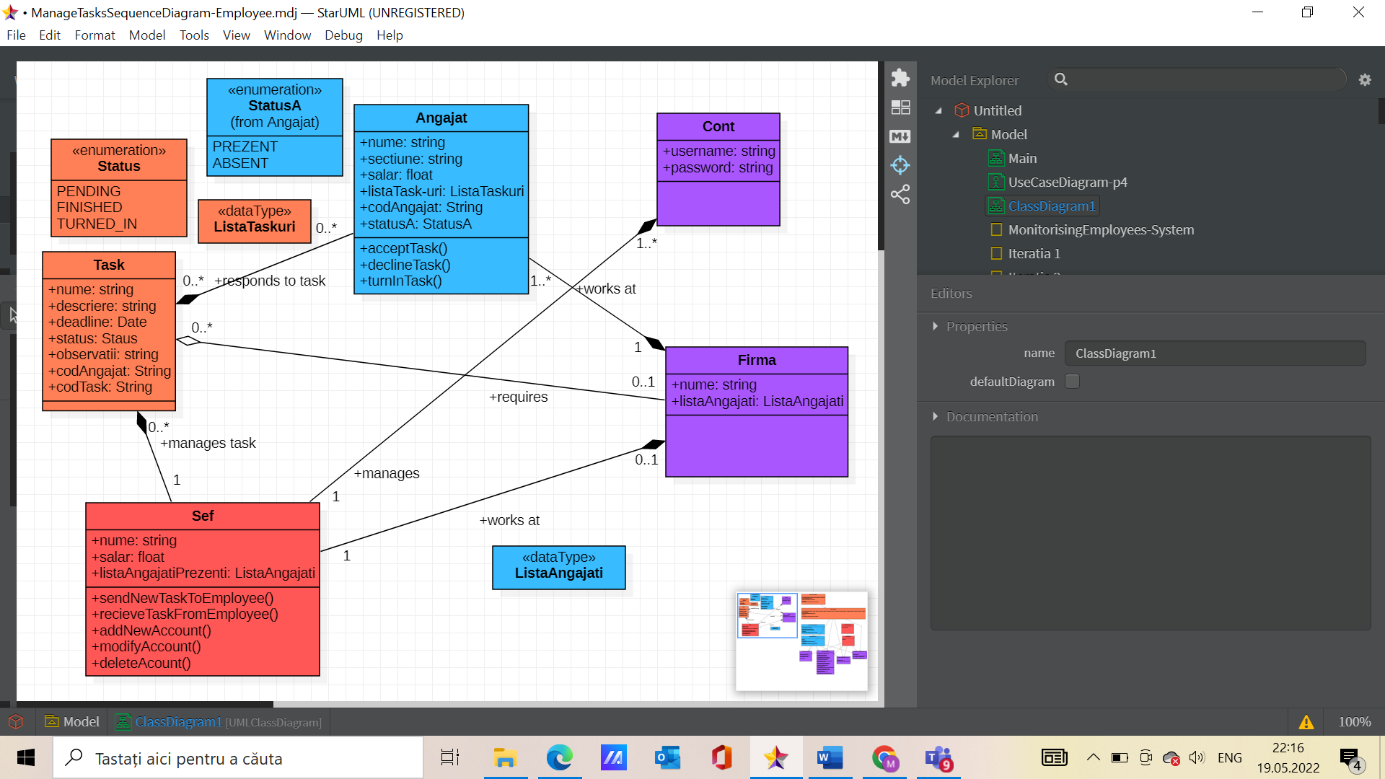


Diagrama de clase completata cu clasele de Repository,Service,Controller:

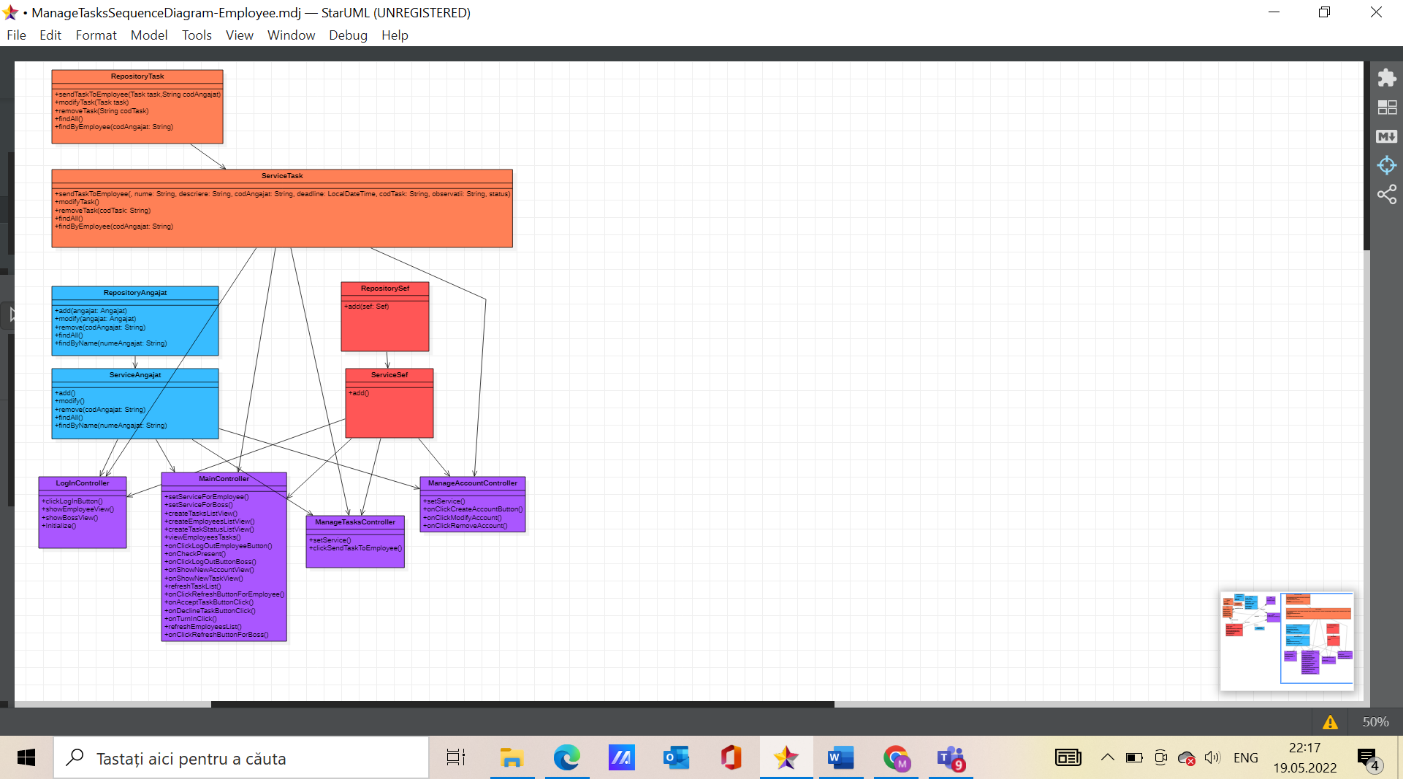
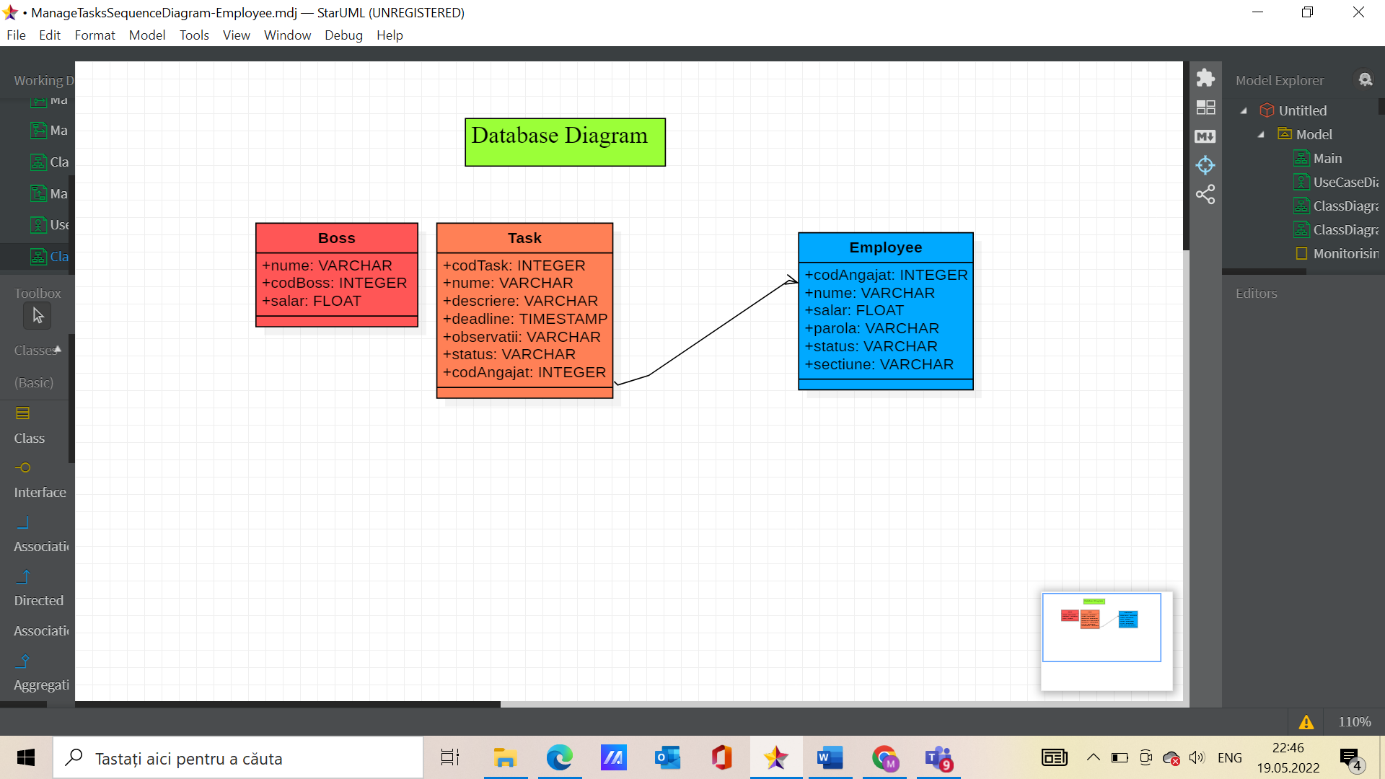
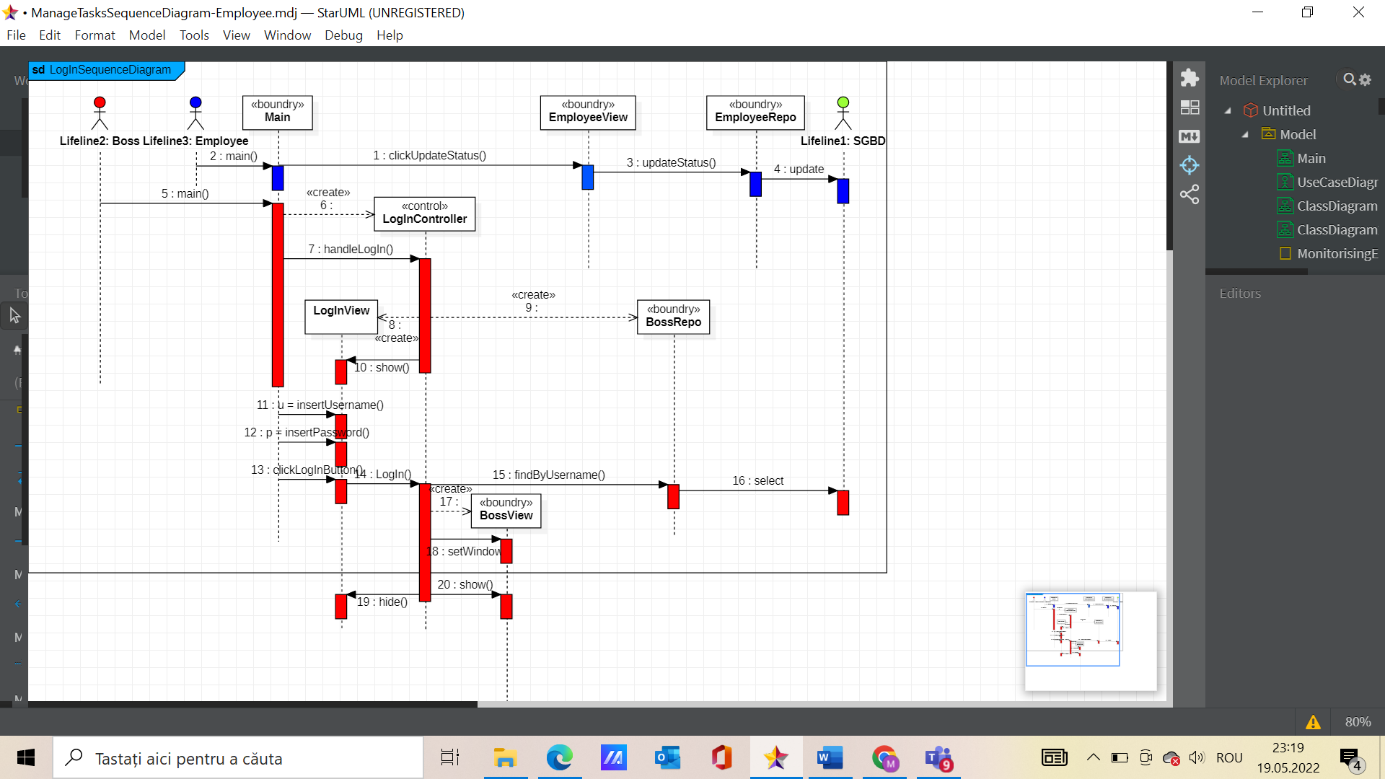


Diagrama **bazei de date:**

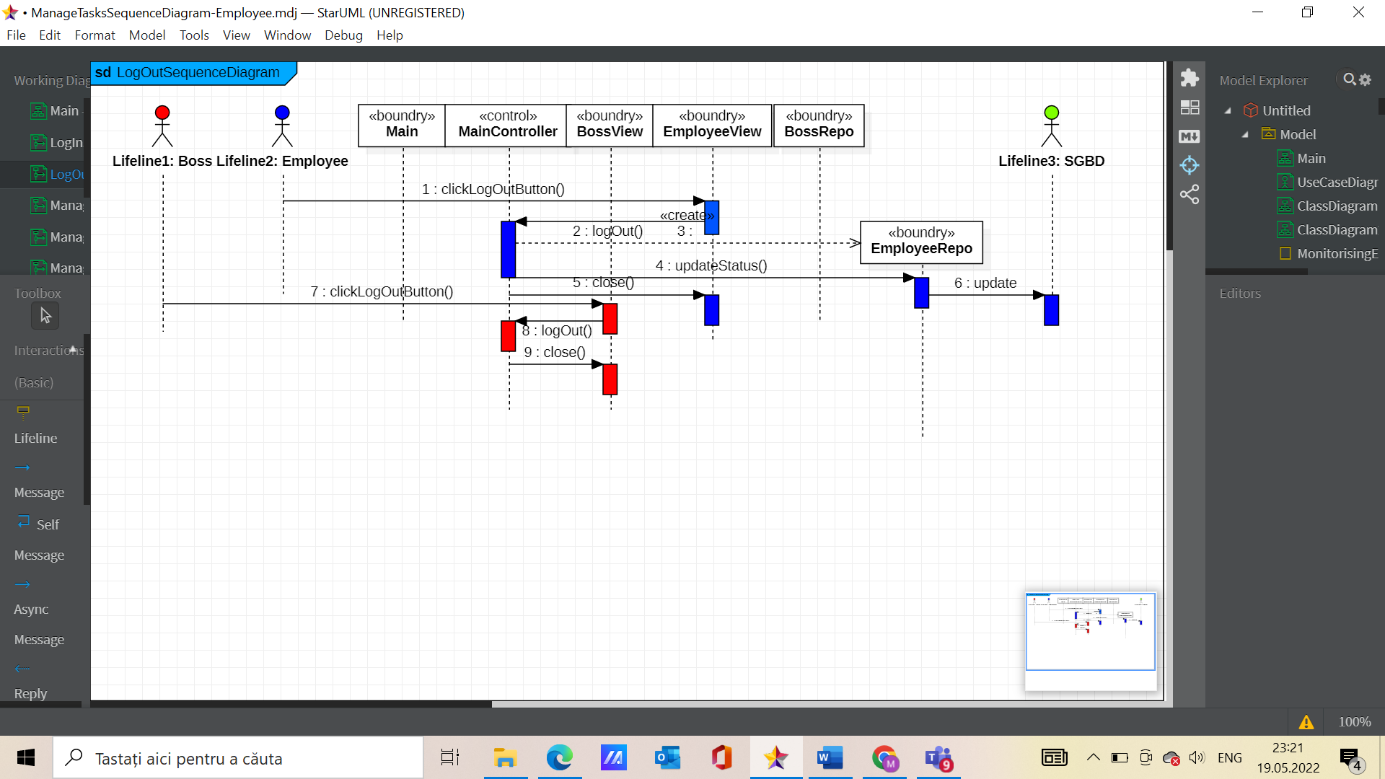


Modelul dinamic-SEQUENCE DIAGRAMS:

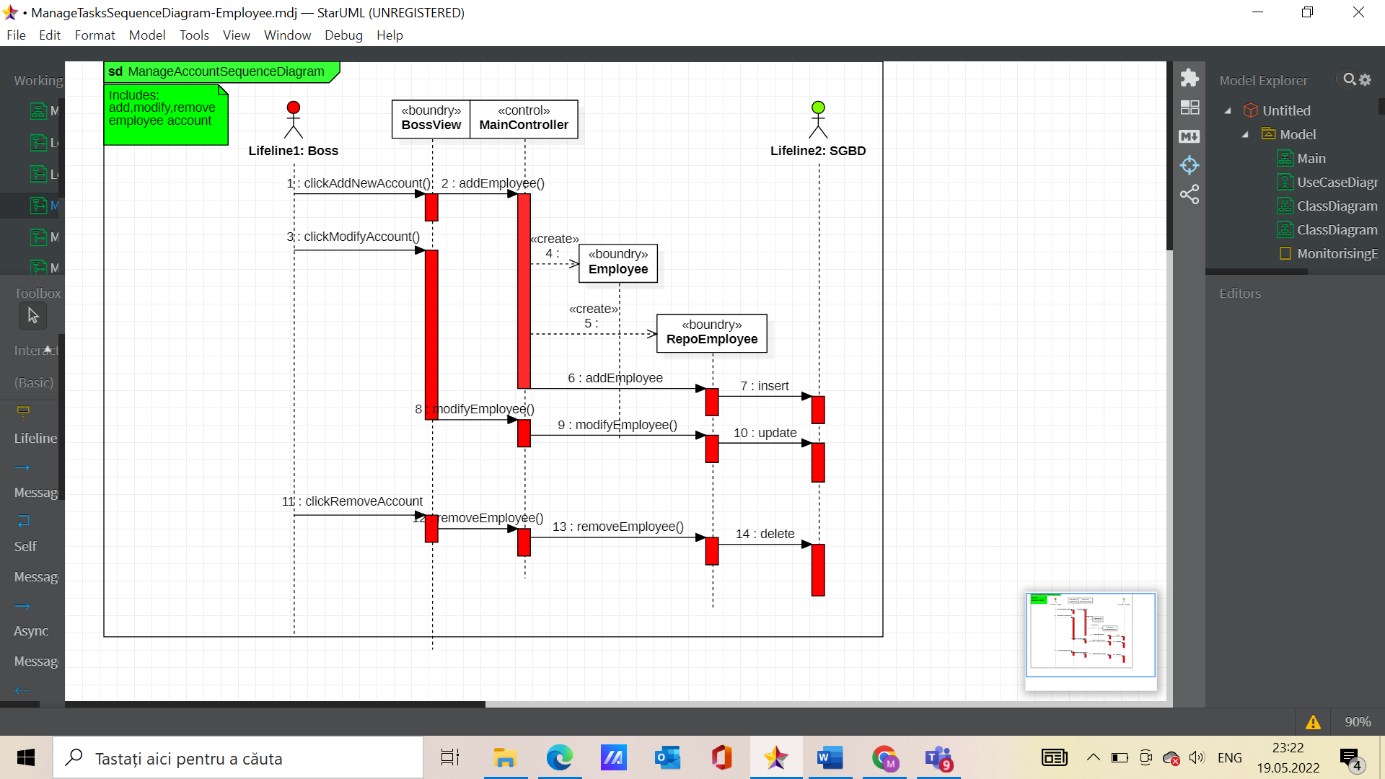
a)LogInSequenceDiagram



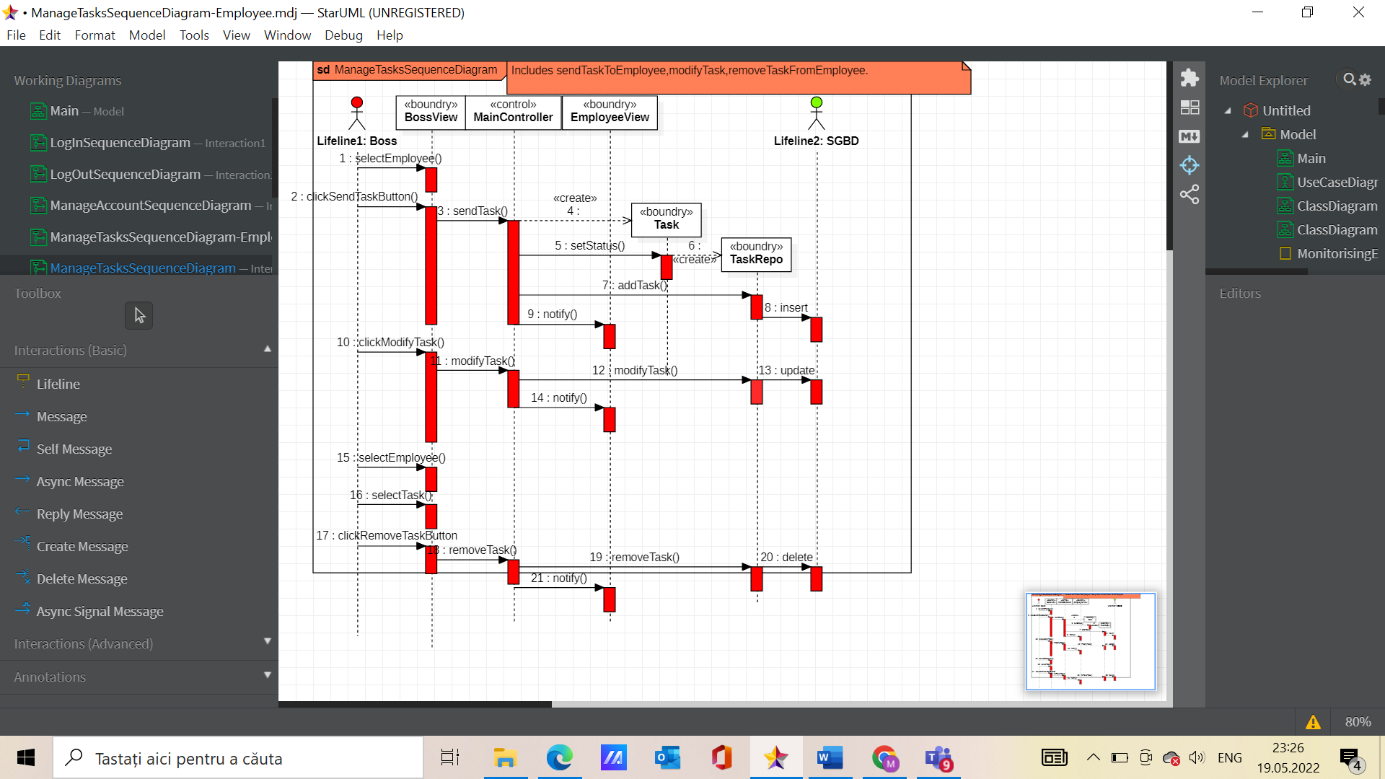
b)LogOutSequenceDiagram



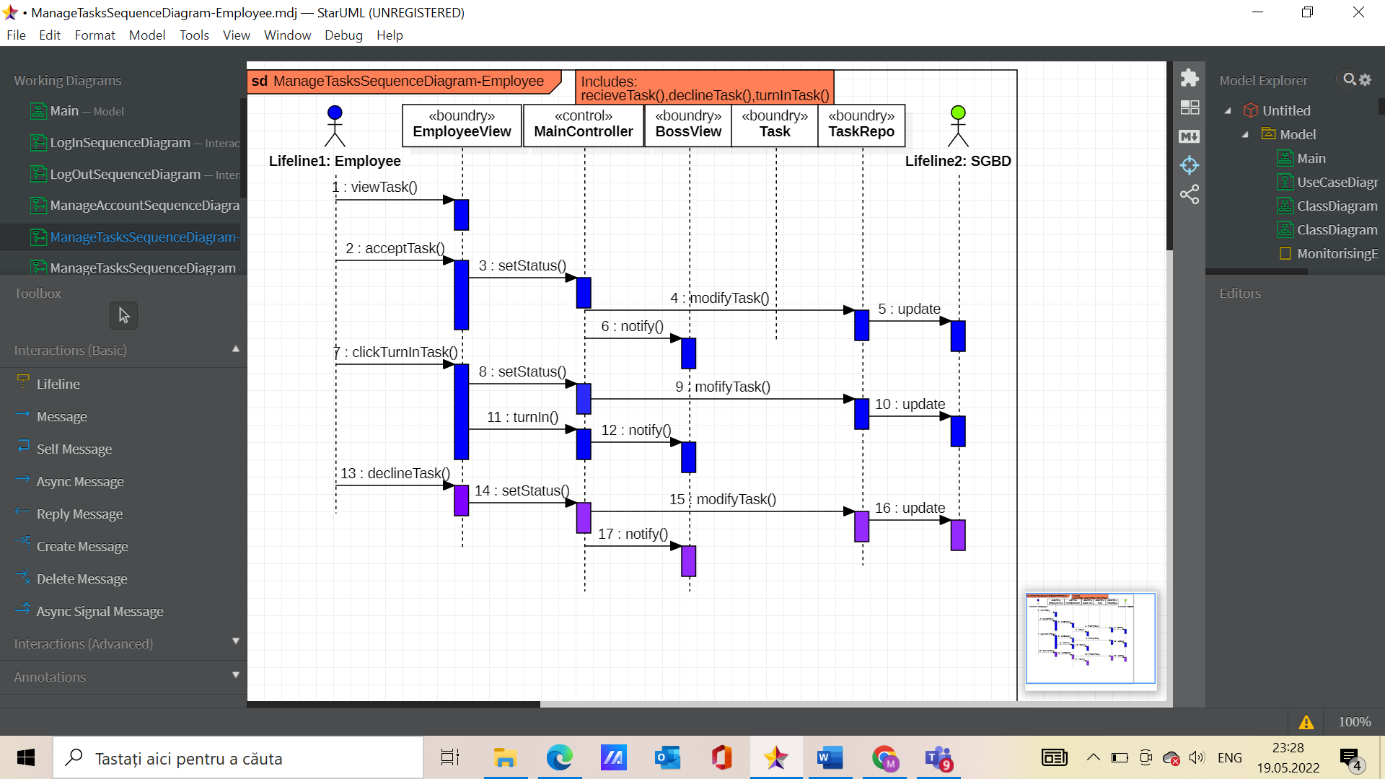
c)ManageAccountDiagram(Create/Update/Delete account)



d)ManageTaskSequenceDiagram-BOSS

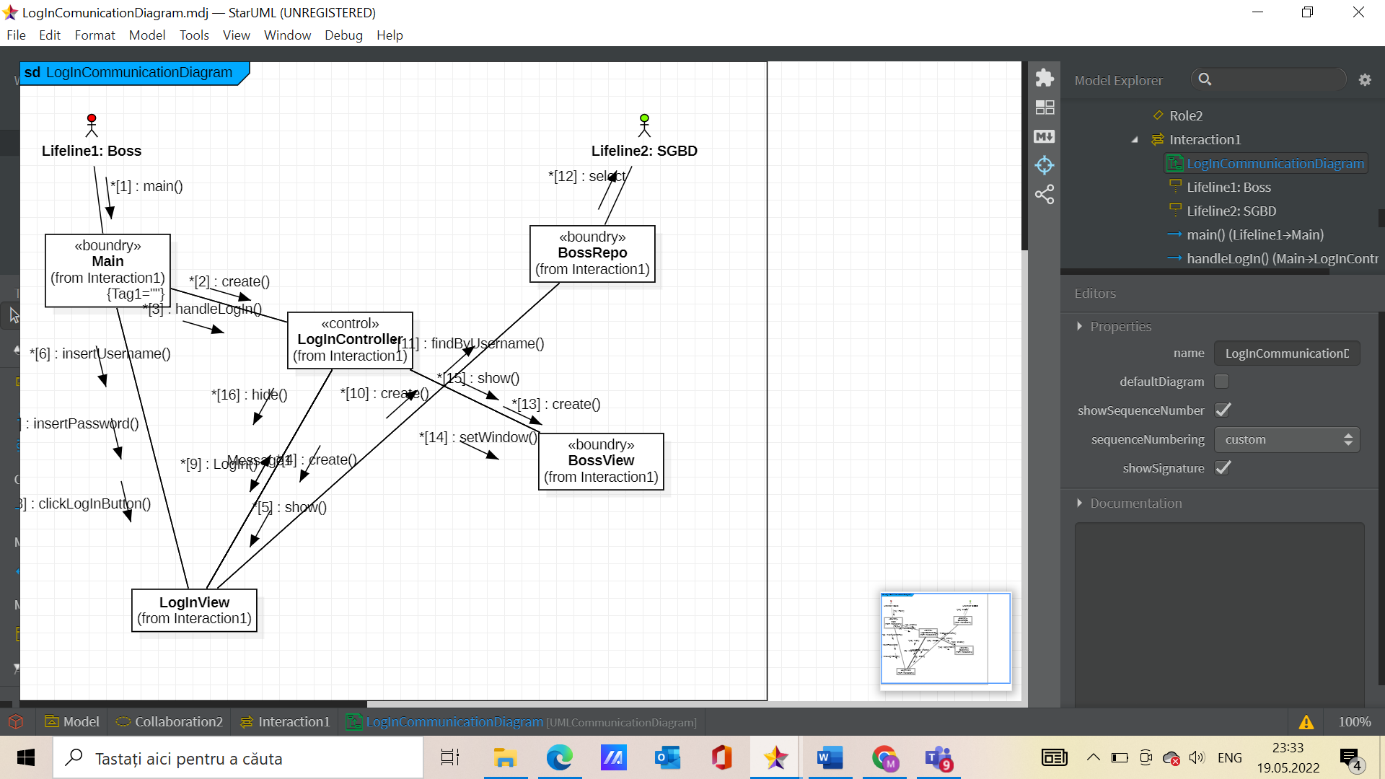


e)ManageTaskSequenceDiagram-EMPLOYEE

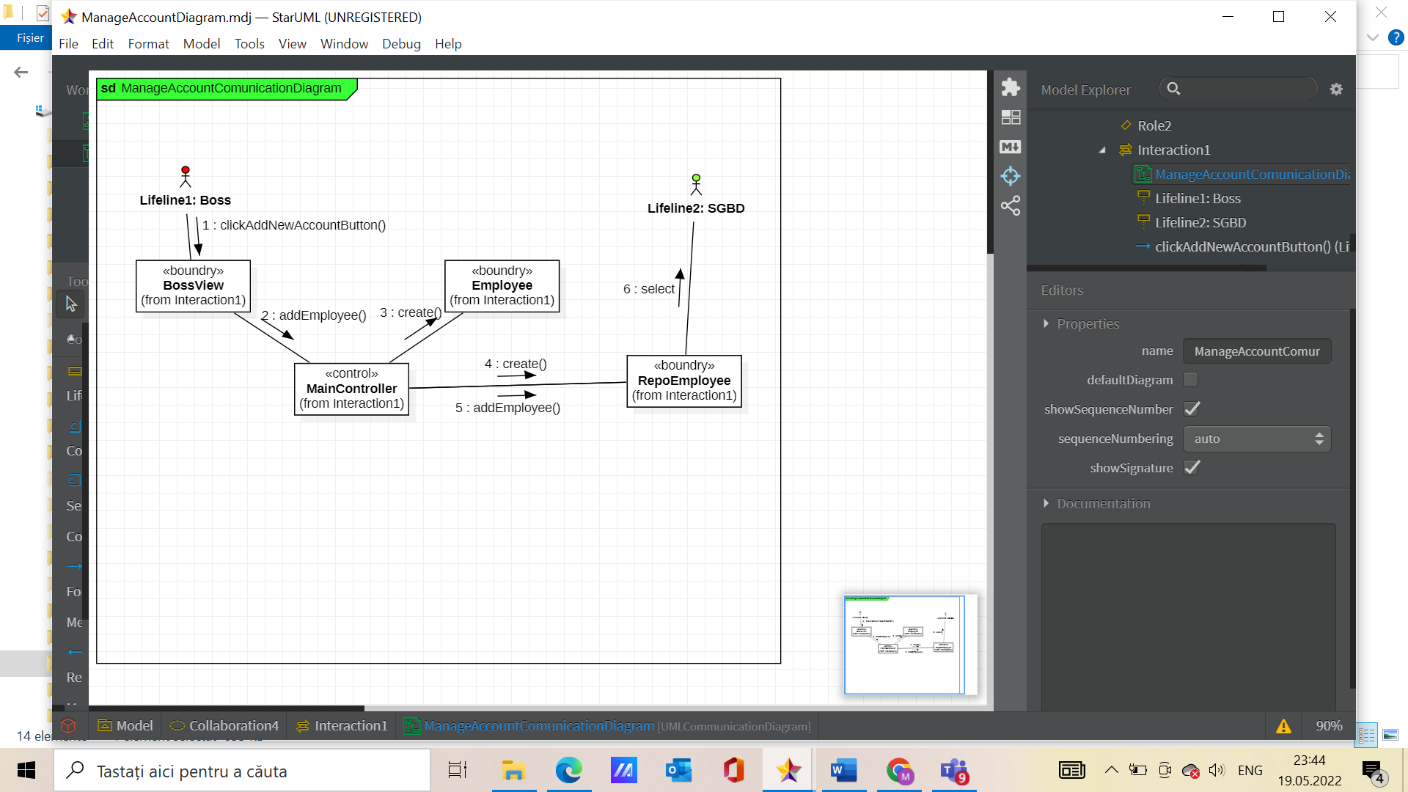


Modelul dinamic-COMUNICATION DIAGRAMS

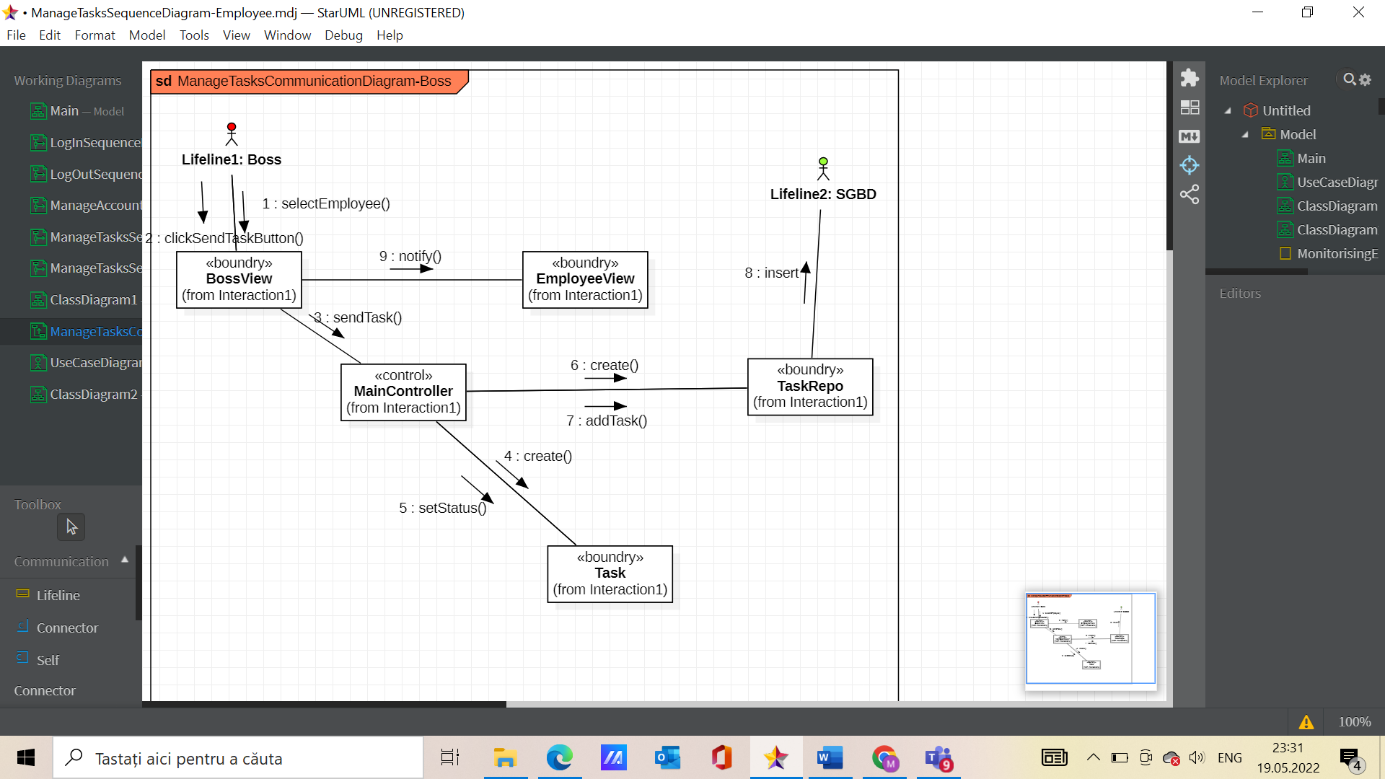
a)ITERATIA 1🡺LogInCommunicationDiagram



b)ITERATIA 2🡺ManageAccountCommunicationDiagram



c)ITERATIA 3🡺ManageTasksCommunicationDiagram



Detalii tehnice

Am folosit:

* **IntelliJ IDEA** pentru a implementa actiunile descrise in diagramele de mai sus.
* **SceneBuilder** pentru a creea designul interfetei grafice.
* **StarUML** pentru a creea diagramele mai sus mentionate si a reprezenta interactiunile care au loc intre clase,obiecte si baza de date.
* **Gradle** pentru a importa dependentele necesare rezovarii problemei.
* **JavaFX -tipul aplicatiei**
* **SQLite** pentru reprezentarea bazei de date care sta in spatele programului.

Instructiuni de utilizare **All@bout:**

1. **All@bout** este o aplicatia personalizata, a unei reviste online, unde seful trimite angajatilor task-uri care au legatura cu un anumit articol care ar urma sa fie publicat in revista.
2. **All@bout** dispune de o interfata de LogIn comuna pentru sef si angajat. Se introduce username-ul ca in imaginea de mai jos,parola si apoi se selecteaza unul din cele doua checkbox-uri(drept cine dorim sa ne conectam),iar in cele din urma se apasa butonul de LogIn. Atentie!!!! ID trebuie sa fie un numar!!!!

**O imagine care conține text

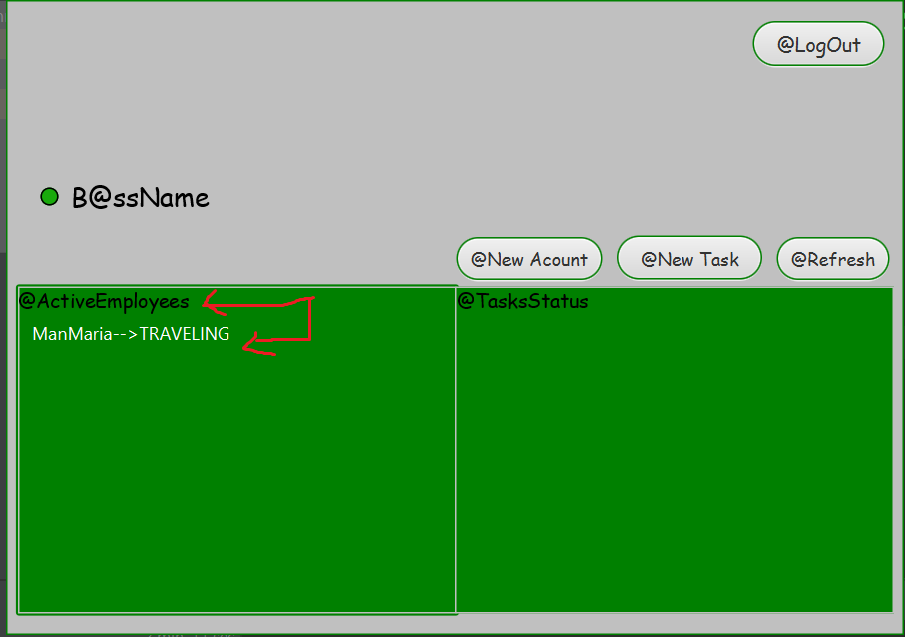
Descriere generată automat**

3) Mai intai deschidem fereastra destinata angajatului, si activam butonul de notificare care sa anunte prezenta angajatului.

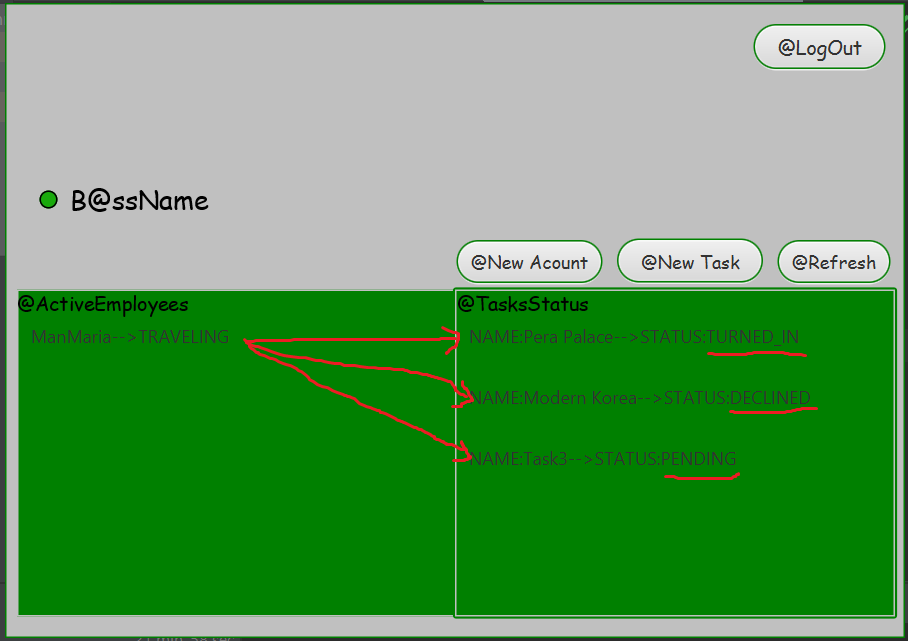
O imagine care conține text

Descriere generată automat

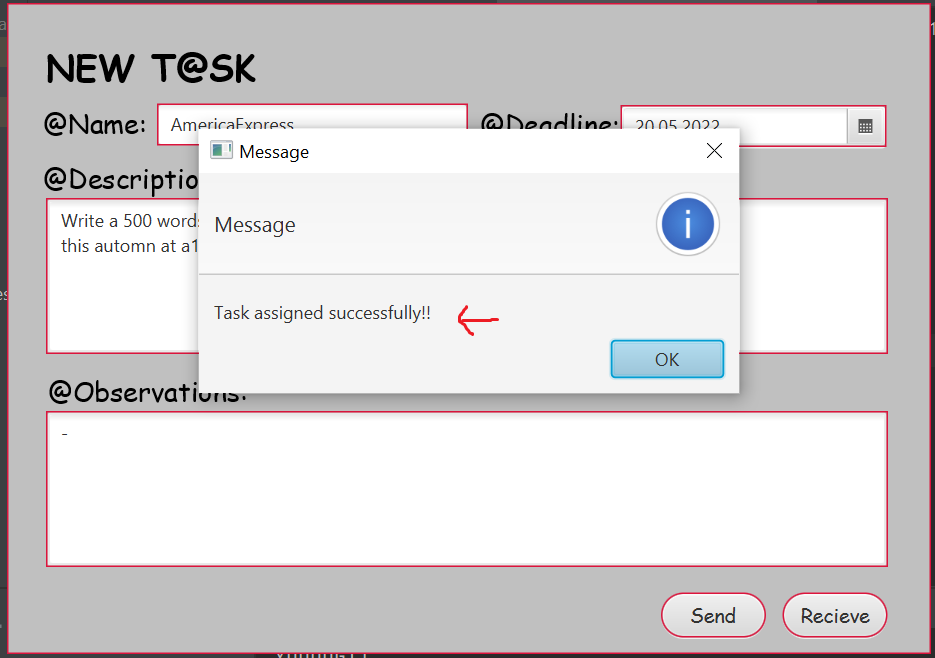
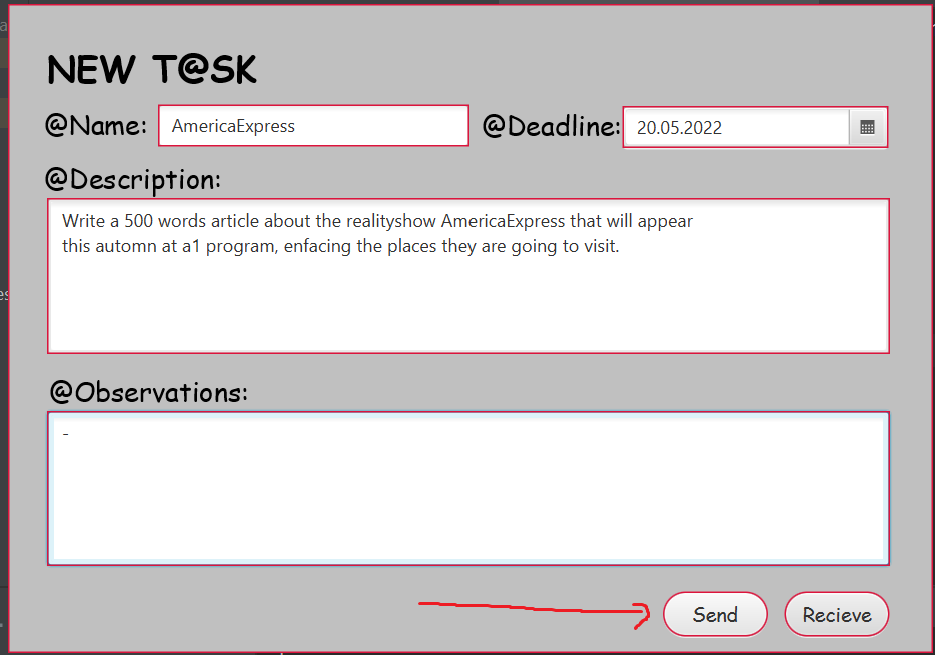
4) Apoi,printr-un alt thread se repeta procedura de logare ,dar pentru sef. El va vedea lista angajatilor prezenti si in dreptul lor, departamentul in care acestia lucreaza(astfel incat sa stie ce fel de task sa ii atribuie angajatului).



5) Seful selecteaza din lista un angajat caruia ii afiseaza in sectiunea **TasksStatus,**printr-un click pe aceasta**,** toate taskurilecurente impreuna cu statusul acestora. Astfel va putea urmari progresul angajatului,disponibilitatea acestuia, dar si stadiul in care se afla task-ul.



6)Printr-o noua selectie urmata de apasarea butonului **@NewTask** se va deschide o noua fereastra in care seful va introduce detaliile unui nou task pentru angajatul selectat. Prin apasarea butonului se va activa o alerta informative care va anunta trimiterea cu success a taskului catre angajat.

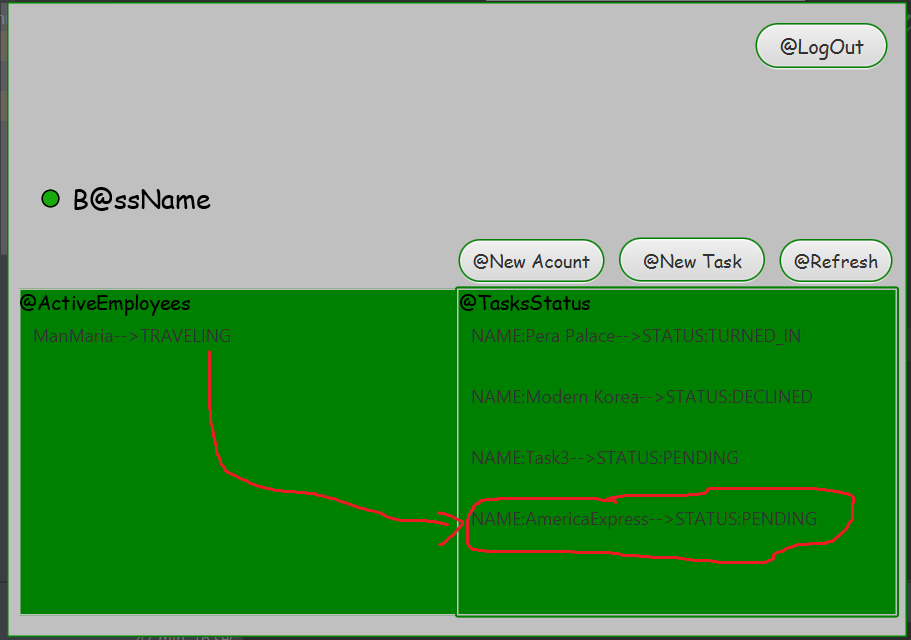


7)Prin apasarea butonului de refresh din fereastra angajatului, lista de task-uri ale acestuia se va actualiza.

De asemenea, prin repetarea pasului 5, se va actualiza si lista de Task-uri disponibila in fereastra sefului.

**O imagine care conține text

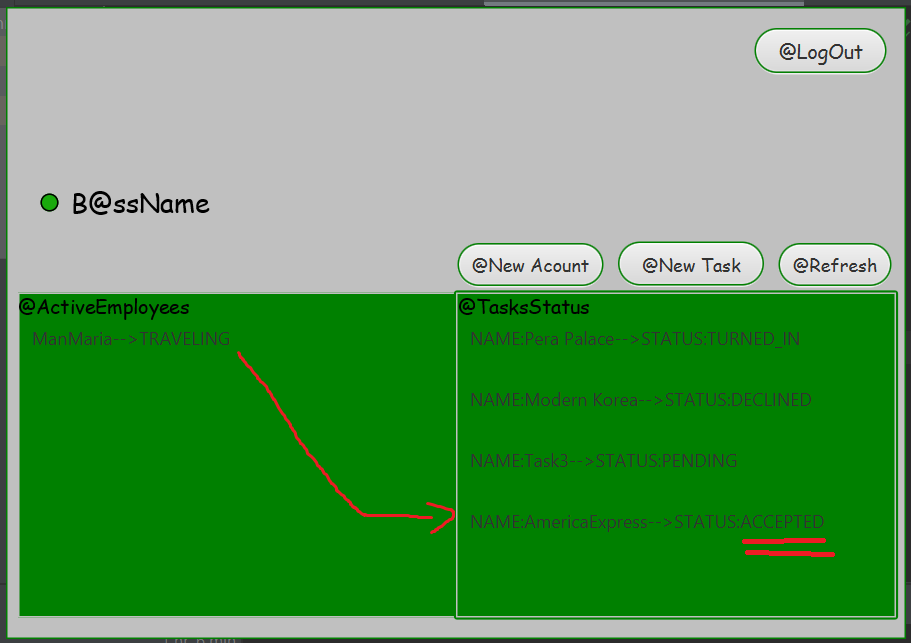
Descriere generată automat**

****

**8)** Angajatul selecteaza task-ul primit si decide ce va face cu el.Daca are timp si nu are alte task-uri in asteptare, va apasa butonul **Accept**. Daca decide ca nu are timp de el, sau nu ii surade task-ul respectiv, poate alege sa il refuze apasand butonul **Decline**. Daca task-ul a fost acceptat si este gata, atunci va apasa **TurnIn** pentru a-l preda superiorului. Un task refuzat sau unul terminat, ramane atat in lista sefului, cat si a angajatului, afisand statusul corespunzator. Se vor sterge doar atunci cand seful va primi si verifica task-ul(adica dupa afisarea statusului Terminat). La fiecare modificare a statusului unui task, se va declansa o alerta informativa.

**O imagine care conține text

Descriere generată automat**

****

9) Prin apasarea butonului **@NewAccount** din fereastra sefului, se va deschide o noua fereastra in care, seful va completa field-urile necesare pentru adaugarea in baza de date a unui nou angajat(fapt ce are loc in urma apasarii butonului **AddEmployee**). Acesta declanseaza de asemenea o alerta informative.

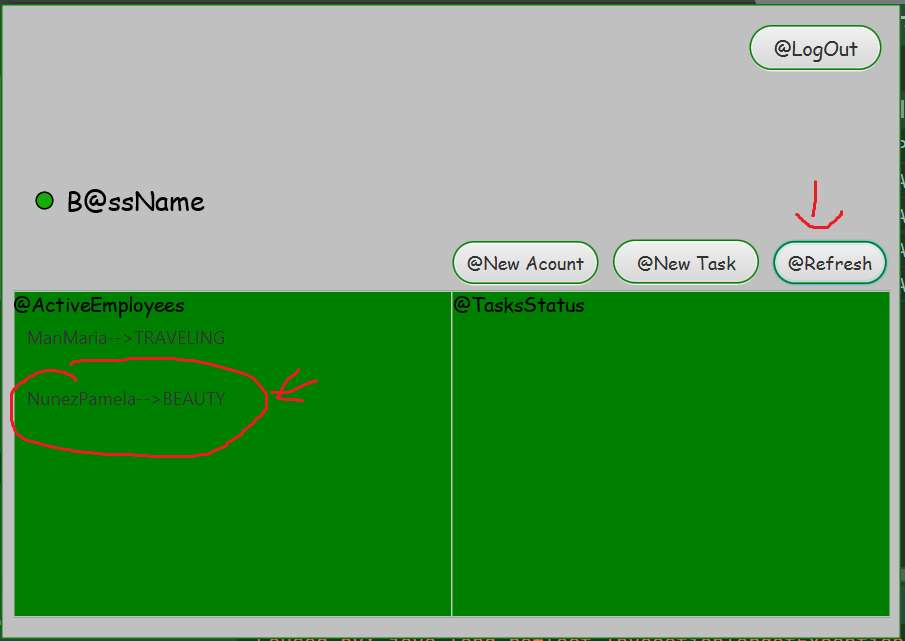
O imagine care conține text

Descriere generată automat

O imagine care conține text

Descriere generată automat

10)Prin apasarea butonului de **@Refresh** din fereastra sefului, dupa logarea noului angajat, se va actualiza lista angajatilor prezenti.



**Documentatie:**

* StackOverflow
* Cursuri ISS
* Seminarii ISS
* Cursuri MPP
* MavenRepositories
* Web site-profesor coordonator de la laborator