**Documentatie Proiect**

-Restaurant Management App-

Aceasta aplicatie reprezinta o platforma de management a unui restaurant care doreste sa faca mult mai usoara administrarea localului.Ne-am bazat pe o arhitectura MVVM (Model,View,ViewModel) pentru a avea un cod ordonat si pentru a separa partea logica de UI.

Este foarte usor de utilizat,dupa ce se realizeaza inregistrarea,se deschide un meniu pentru a selecta ce dorim sa administram:

O imagine care conține text, captură de ecran, Dreptunghi, Font

Descriere generată automat

1. Employees permite adaugarea,modificarea si stergerea angajatiilor:
2. Products permite adaugarea,modificarea si stergerea preparatelor
3. Table permite adaugarea,modificarea si stergerea meselor din local
4. Order Ticket reprezinta partea in care se adauga/sterg comenzile din restaurant si care in care,ulterior,se va genera chitanta pentru client.

Urmeaza cateva screenshot-uri din aplicatie in care se prezinta fiecare optiune:

1.

O imagine care conține text, captură de ecran, software, Pictogramă computer

Descriere generată automat

2.

O imagine care conține text, captură de ecran, software, diagramă

Descriere generată automat

3.

O imagine care conține text, captură de ecran, software, afișaj

Descriere generată automat

4.

O imagine care conține text, captură de ecran, software, Pictogramă computer

Descriere generată automat

Pe partea de back-end,am folosit MSSQL(Microsoft SQL Server Management Studio) ca si baza de date.Pentru a avea o legatura mai buna cu aceasta si mai safe,am folosit proceduri stocate care au fost apelate in functiile CRUD din layer-ul de DataAcces.

O imagine care conține text, captură de ecran, software

Descriere generată automat

Acest screenshot reprezinta o functia prin care se obtin toate produsele din tabela products in momentul in care este rulata procedura stocata “GetProducts”.Daca acest proces nu functioneaza cum ne asteptam,se va returna o exceptie.

Aceste operatii CRUD sunt integrate in functiile din BusinessLogic care ne ajuta sa legam logica din DataAcces cu cea din ViewModel,responsabila pentru pregatirea si formatarea elementelor UI.

O imagine care conține text, captură de ecran, software

Descriere generată automat

-Secventa din ProductViewModel-

O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automat

-Secventa din ProductBLL-

Ca si design pattern-uri,am folosit Factory si Observer.Observer ne-a ajutat mult pe partea de formatare a elementelor UI deoarece aveam nevoie sa stim cand apar modificari.Design pattern-ul Factory ne-a ajutat in stratul de access la date pentru a creea obiecte care au dependente complexe sau care necesita o logica de creare speciala.

O imagine care conține text, captură de ecran, software, Sistem de operare

Descriere generată automat O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automat

-Initializarea interfetelor si utilizarea acestora-

Pentru realizarea unit testelor am folosit framework-ul MSTest,cu care am facut unit teste simple ,cat si unit teste cu Mocking:

O imagine care conține text, captură de ecran, software

Descriere generată automat

-Unit test simplu-

O imagine care conține text, captură de ecran, software

Descriere generată automat

-Unit test cu Mocking-

In cele simple am ales sa testam proprietatile de set si get ale fiecare entitati,iar in cele cu mocking am ales sa testam functiile de get.Tinem sa mentionam ca au fost facute unit teste pentru fiecare unitate si pentru fiecare procedura stocata de Get.

O imagine care conține text, diagramă, linie, captură de ecran

Descriere generată automatDiagramele realizate:

- Use Case diagram-

-Aici este reprezentata o imagine in ansamblu asupra proiectului si a functionalitatiilor

acestuia.Sunt 3 personaje:administrator,client si chelner/angajat.Administratorul gestioneaza partea din spate a restaurantului,chelnerul preia si inchide comenzi,iar clientul comanda si plateste consumatia.

O imagine care conține text, captură de ecran, Font, Paralel

Descriere generată automat

-Class diagram-

Aici avem organizata structura claselor care stau la baza acestui proiect,impreuna cu getteri si setter.

O imagine care conține text, captură de ecran, Paralel, diagramă

Descriere generată automat

-Sequence Diagram-

In aceasta diagrama se arata legatura dintre ospatar si sistem prin intermediul interfetei.

O imagine care conține text, captură de ecran, diagramă, linie

Descriere generată automat

-Activity diagram-

In aceasta diagrama se prezinta procesul prin care se realizeaza comanda,se finalizeaza si se plateste,actiunea fiind facuta de catre client si chelner.

O imagine care conține diagramă, Plan, Desen tehnic, schematic

Descriere generată automat

-State Chart diagram-

In aceasta diagrama este cuprinsa functionalitatea aplicatiei prin starile pe care le poate avea aplicatia in momentul in care o optiune este selectata.

O imagine care conține text, diagramă, captură de ecran, linie

Descriere generată automat

-Collaboration diagram-

In aceasta diagrama se prezinta colaborarea dintre aplicatie si cei 2 actori in care unul dintre ei realizeaza modificarile necesare pentru a oferii servicii bune,iar al doi -lea actor realizeaza o comunicare intre aplicatie si client.