

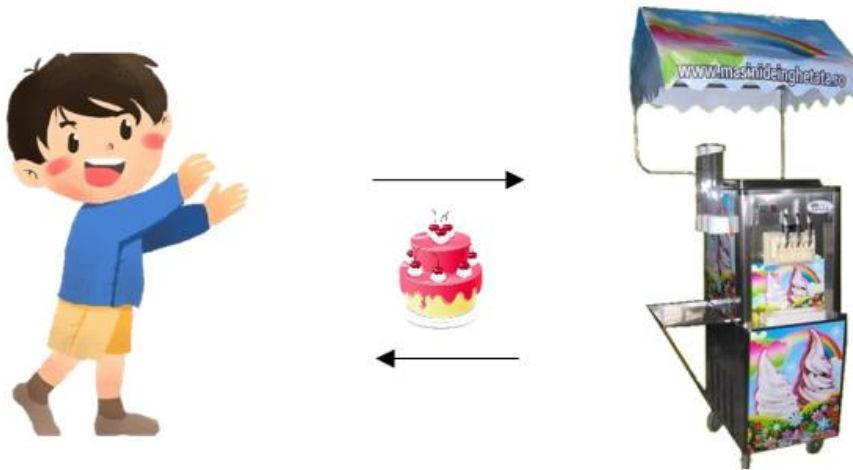
PROIECT 1 – Cake Maker

Cake Maker este un proiect care presupune dezvoltarea unei **aplicații C++ tip consolă** care **simulează** un automat de făcut prăjituri.

Rolul principal al aplicației este să servească clientul cu prăjitura dorită. Servirea se va face prin intermediul unui meniu principal care îi va oferi clientului anumite posibilități de selecție.

- Afișarea meniului de prăjituri care pot fi comandate
- Afișarea prăjiturilor existente
- Comandarea unui prăjituri

Toate aceste funcționalități vor fi disponibile numai după înregistrarea clientului, facilitând astfel crearea unui istoric cu toți clienții și prăjiturile comandate de aceștia.



1. Componentele aplicației:

Componentele aplicației sunt prezentate în figura 1 și acestea sunt:

- a) Panoul de comandă
- b) Panoul de control
- c) Depozit de prăjituri
- d) Preparatorul de prăjituri

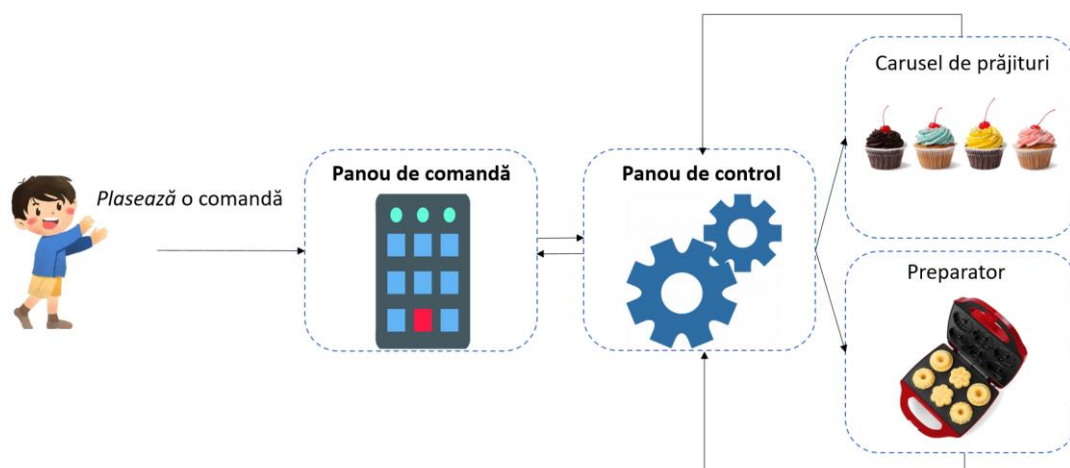


Figura 1 – Componentele aplicației

2. Descrierea aplicației:

Pentru simularea acestui automat, programul C++ va dispune de un meniu care prezintă comenzile posibile pentru un utilizator. Aceste comenzi sunt:

- 1) Înregistrare client nou
- 2) Autentificare client existent
 - Posibilitate resetare parolă
 - Posibilitate modificare nume client
- 3) Afișare prăjituri posibile
- 4) Comandă prăjitură
- 5) Afișare prăjituri existente
- 6) Afișare istoric pentru un anumit interval orar

Pentru un client se va cunoaște numele, adresa de email și parola.

Pentru o prăjitură se cunoaște numele acesteia și rețeta. Iar o rețetă este caracterizată prin nume rețetă și timp de preparare exprimat în minute. Există două tipuri de prăjituri – prăjituri obișnuite și prăjituri pentru diabetici. Pentru prăjiturile diabetice se mai cunoaște și cantitatea de zahăr.

Depozitul de prăjituri ai automatului este caracterizat printr-o listă de prăjituri, capacitate maximă.

Panoul de control dispune de următoarele funcționalități:

- Preluare comandă, în cazul în care prăjitura comandată există în depozit ea va fi servită imediat utilizatorului. În cazul în care prăjitura nu există în depozit, se va crea o comandă pentru preparatorul de prăjituri folosind rețeta aferentă prăjiturii comandate. După așteptarea timpului conform rețetei, prăjitura va fi servită utilizatorului
- Returnarea numelor prăjiturilor existente din depozit
- Umplerea automată a depozitului cu ultima prăjitură comandată, dacă depozitul ajunge la 0, în urma unei comenzi.

Preparatorul de prăjituri dispune de următoarea funcționalitate: preia o comandă de preparare (o rețetă) și simulează un timp de așteptare aferent rețetei primite. După scurgerea timpului, acesta returnează prăjitura creată.

3. Inițializare automat:

Înainte de punerea în funcțiune a automatului el trebuie inițializat cu lista de prăjituri/rețete pe care le poate pune la dispoziția utilizatorului.

În programul C++, această inițializare se produce înainte de afișarea meniului prin citirea unei liste de prăjituri (fiecare cu rețeta aferentă) dintr-un fișier numit rețete.txt, prin intermediul panoului de control.

4. Afișare prăjituri posibile:

La primirea acestei comenzi, automatul va afișa toate prăjiturile pe care le va putea procesa. În programul C++, pe ecran va fi afișată lista de prăjituri cu care a fost inițializat automatul, prin intermediul panoului de control.

5. Afișare prăjituri existente:

La primirea acestei comenzi, automatul va afișa toate prăjiturile care se află în depozit. În programul C++, pe ecran vor fi afișate toate prăjiturile care se află în depozit, prin intermediul panoului de control.

6. Comandă prăjitură:

La primirea acestei comenzi, automatul, prin intermediul panoului de control primește numele unei prăjituri și opțional numărul de bucăți. Apoi, el verifică dacă prăjitura comandată există în depozit, dacă da va returna prăjitura găsită, dacă nu, prin intermediul preparatorului se va crea o prăjitură folosind rețeta asociată. De asemenea, dacă în urma comenzii, depozitul nu este plin, acesta se va umplea cu ultima prăjitură comandată, până la capacitate maximă.

În programul C++, comanda introdusă va fi preluată de panoul de control care face verificarea existenței prăjiturii în depozit. Dacă aceasta există, prăjitura va fi afișată pe ecran, dacă nu există, se va afișa un mesaj de informare pentru utilizator că prăjitura nu există și trebuie să aștepte un timp – indicat de rețeta asociată prăjiturii comandate, apoi se va simula crearea unei prăjituri, folosind rețeta asociată, prin intermediul preparatorului. Preparatorul va simula așteptarea tipului necesar pentru gătire. După acest timp de așteptare, prăjitura va fi afișată pe ecran prin intermediul panoului de control. De asemenea se va verifica dacă depozitul este plin și se va acționa în consecință.

7. Afișare istoric:

La primirea acestei comenzi, automatul va afișa un istoric al comenzilor efectuate pentru un anumit interval orar. Se va afișa numele clientului, adresa de email, și prăjiturile comandate.

Observații:

În cazul introducerii unei comenzi greșite se va arunca o excepție corespunzătoare, iar mesajul de eroare va fi afișat pe ecran și în același timp va fi stocat într-un fișier log.txt.

Simularea timpului de așteptare se poate face prin intermediul bibliotecii <Windows.h>, funcția „Sleep()”.

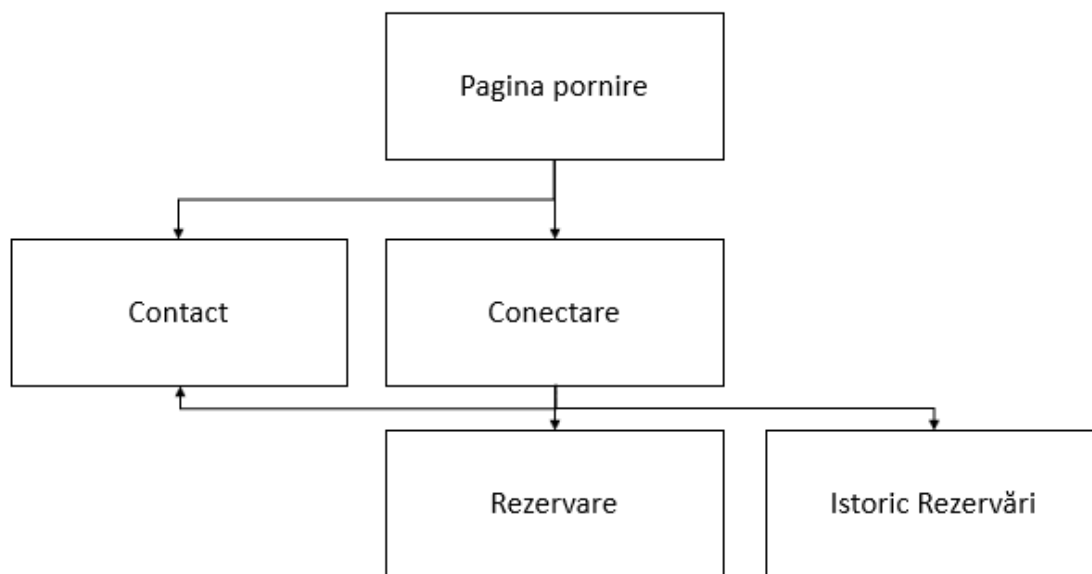
PROIECT 2 - Bookstore App

Bookstore App este un proiect care presupune dezvoltarea unei **aplicații C++ tip consolă** care **simulează** o aplicație de tip web.

Programul permite realizarea achiziționarea respectiv împrumutul de cărți dintr-o librărie.

1. Componentele aplicației:

Componentele aplicației sunt prezentate în schema 1.



Schema 1 – Componentele aplicației

2. Pagina de pornire (a unui utilizator neautentificat)

Prima pagină a website-ului (aplicației C++) permite utilizatorului două opțiuni:

- 1) *Conectare*, care presupune deschiderea unui alt meniu care va prezenta alte trei opțiuni:
 - 1) *Înregistrare* (dacă este un utilizator nou)
 - 2) *Autentificare* (dacă este un utilizator existent)
 - 3) *Resetare parolă*
- 2) *Contact*, care permite utilizatorului să transmită un email către echipa de suport (funcționalitatea va fi descrisă în secțiunea de contact a unui utilizator autentificat)

BookstoreApp.

Alegeți o opțiune:

1. Conectare
2. Contact

3. Fereastra de conectare

La accesarea opțiunii de conectare pe ecran va fi afișată o fereastră care permite utilizatorului să se conecteze la aplicație, să se înregistreze sau să își reseteze parola.

Conectare

Alegeți o opțiune:

1. Autentificare
2. Înregistrare
3. Resetare parolă
4. Înapoi

La selectarea opțiunii *Înregistrare*, utilizatorul va trebui să introducă: **numele**, **prenumele**, **CNP-ul** (unic), o adresă validă de **e-mail** (unic) și o **parolă** (trebuie să conțină cel puțin 3 caractere).

Aceste date vor fi stocate într-un fișier text, după care va fi afișat un mesaj de succes.

La selectarea opțiunii *Autentificare*, utilizatorul va trebui să introducă o adresă de e-mail validă și o parolă. În cazul în care adresa specificată și parola corespund unui utilizator înregistrat, autentificarea are succes, iar pe ecran va apărea un nou meniu, un meniu al unui utilizator autentificat. În toate etapele, vor fi afișate mesaje corespunzătoare.

La selectarea opțiunii *Resetare parolă*, utilizatorul va trebui să introducă adresa de e-mail. Dacă un utilizator cu această adresă de email există, atunci se va introduce o nouă parolă care va înlocui parola actuală pentru utilizatorul identificat. După care se va afișa un mesaj corespunzător de succes.

4. Pagina unui utilizator autentificat

După ce utilizatorul se autentifică, acestuia îi va fi afișat următorul meniu:

- a) Rezervare
- b) Istoric achiziții
- c) Contact (același meniu de contact de care dispune un utilizator neautentificat)

BookstoreApp (Nume, Prenume)

Alegeți o opțiune:

1. Achiziție/Împrumut
2. Istoric

- 3. Contact
- 4. Înapoi

1. La selectarea opțiunii **Achiziții/Împrumut**, pe ecran va fi afișat un nou meniu cu 2 opțiuni: Achiziție sau Împrumut.

BookstoreApp (Nume, Prenume)

Alegeți opțiunea:

- 1. Achiziție
- 2. Împrumut
- 3. Înapoi

La selectarea oricărei dintre cele 2 opțiuni vor fi afișate cărțile **disponibile**. Pentru fiecare carte se va cunoaște: titlul, numele autorilor, editura, prețul. Există două categorii de cărți: ficțiune pentru care se cunoaște și numărul de pagini și cărți non-ficțiune pentru care se cunoaște și anul publicării. Prețul se calculează în funcție de tipul cărții, dacă este ficțiune va fi preț de bază + numărul de pagini, iar dacă este carte non-ficțiune prețul de bază va fi preț bază – (25 * anul publicării).

Toate cărțile existente, vor fi citite dintr-un fișier.

De asemenea, pentru fiecare carte se cunoaște și numărul din stoc.

- a) Dacă se alege achiziție, numărul din stoc trebuie să fie diminuat cu 1 și cartea va fi marcată ca și **vândută**
- b) Dacă se alege împrumut, cartea va fi marcată ca și **împrumutată**.

2. La selectarea opțiunii **Istoric**, pe ecran se vor afișa toate achizițiile efectuate de utilizatorul autentificat, cât și totalul cheltuit pe achiziții.

3. La selectarea opțiunii **Contact**, la fel, ca și în meniul unui utilizator neautentificat.

5. Pagina de contact

Afișarea acestei pagini se realizează prin afișarea următorului meniu.

BookstoreApp (Nume, Prenume)

Suport tehnic

În cazul în care întâlniți dificultăți echipa de suport vă va sta la dispoziție.

Tot ce trebuie să faceți este să completați următorul formular în care trebuie descrisă problema.

Alegeți o opțiune:

- 1. Trimitere sesizare
- 2. Înapoi

La selectarea opțiunii Trimitere sesizare, va trebui introdusă o problemă. Sesizarea va conține astfel: numele și prenumele, adresa de e-mail, CNP-ul și conținutul problemei. Se va transmite un e-mail către adresa support@bookstoreapp.com (pentru testare se va utiliza adresa unui prieten). Adresa care va transmite e-mailul de mai sus va fi cea a programatorului împreună cu datele de conectare ale acestuia (se va utiliza numai la testare, de asemenea credențialele se pot utiliza dintr-un fișier text local – **datele private nu trebuie expuse în proiect**).

Forma unui e-mail poate arăta după cum urmează:

Buna ziua,
Subsemnatul Nume Prenume, avand CNP: 1250122202020
Nu pot accesa website-ul, va rog sa ma ajutati in aceasta privinta.
Multumesc,
Nume Prenume

Observații

În toate etapele, atunci când utilizatorul va introduce date greșite, mesajele de eroare se vor stoca în fișierul *log.txt*.

Se vor utiliza liste, supraîncărcarea operatorilor, scrierea în fișiere, tratarea excepțiilor.

Transmitere e-mail

Pentru transmiterea unui email dintr-o aplicație c++ este necesară conectarea la o rețea de internet fără proxy (se pot utiliza date mobile în acest caz).

Pași necesari:

1. Descărcare cURL <https://curl.se/windows/>
2. Extragere arhivă și copiere **conținut** folder *bin* în același folder cu proiectul C++
3. Includere bibliotecă <Windows.h>, creare comandă *curl* și lansarea curl.exe prin comanda **WinExec("comanda trimitere email", SW_HIDE);** sau prin utilizarea CreateProcess.
4. Informații creare comanda pentru trimitere e-mail utilizând cURL: <https://ec.haxx.se/usingcurl/usingcurl-smtp>

În ceea ce privește adresa de e-mail a programatorului există și posibilitatea înregistrării unei aplicații pentru a putea permite server-ului de e-mail (ex. gmail) să se autentifice și să transmită un e-mail, însă calea cea mai simplă este modificarea setării "Allow less secure apps: ON" - pentru un cont **gmail**. Mai multe detalii la: <https://myaccount.google.com/lesssecureapp>.