Object Oriented Programming

Project

“E-Banking App”

By

Noman Shoukat (66008)

And

Syed Afnan Ali (65830)

Table of Contents

* Introduction
* Project Overview
* Features
* User Interface
* Project Summary
* Technologies and Usage
* Installation, Setup, and Code

Introduction

This report details the development of a C# console application for managing user accounts and financial transactions. The project was created as part of the Object-Oriented Programming course to demonstrate practical applications of OOP principles.

Project Overview

The application allows users to create accounts, manage their finances, and track transactions through a console interface. Key features include account creation, login, dashboard view, money management, and transaction history.

Features

**User Accounts:**

* **Signup Process:** Users can create a new account by providing their personal details and choosing a secure password.
* **Login:** Existing users can log in using their credentials to access their account.

**Dashboard:**

* **Account Overview:** The dashboard displays the user's current account balance.
* **Recent Transactions:** A list of recent transactions is shown for quick reference.

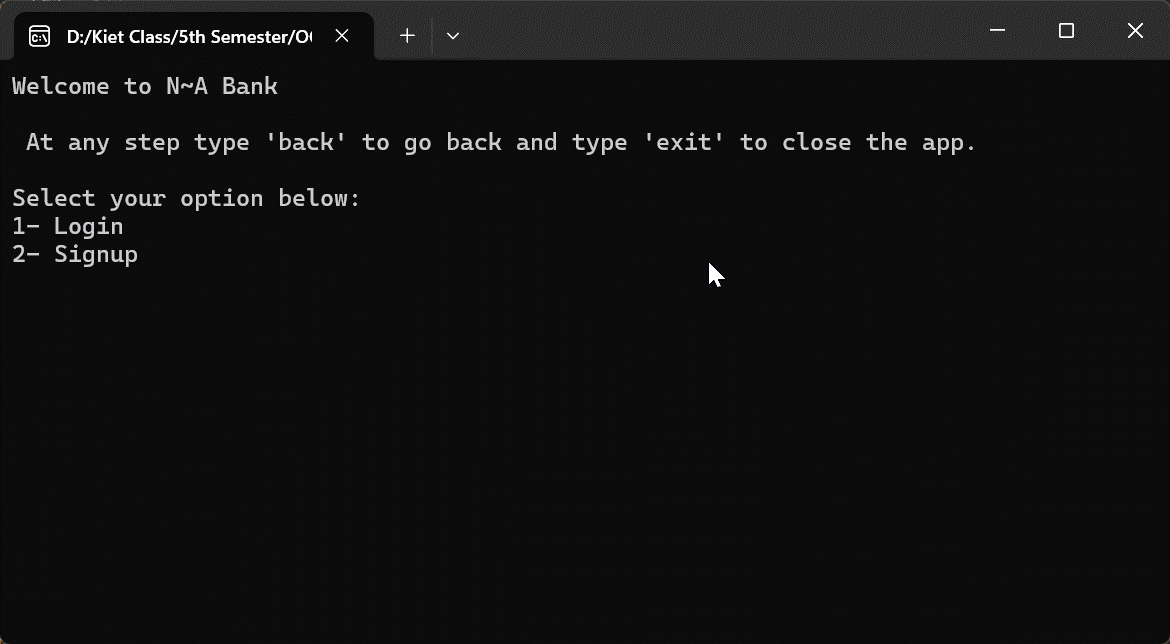
**Money Management:**

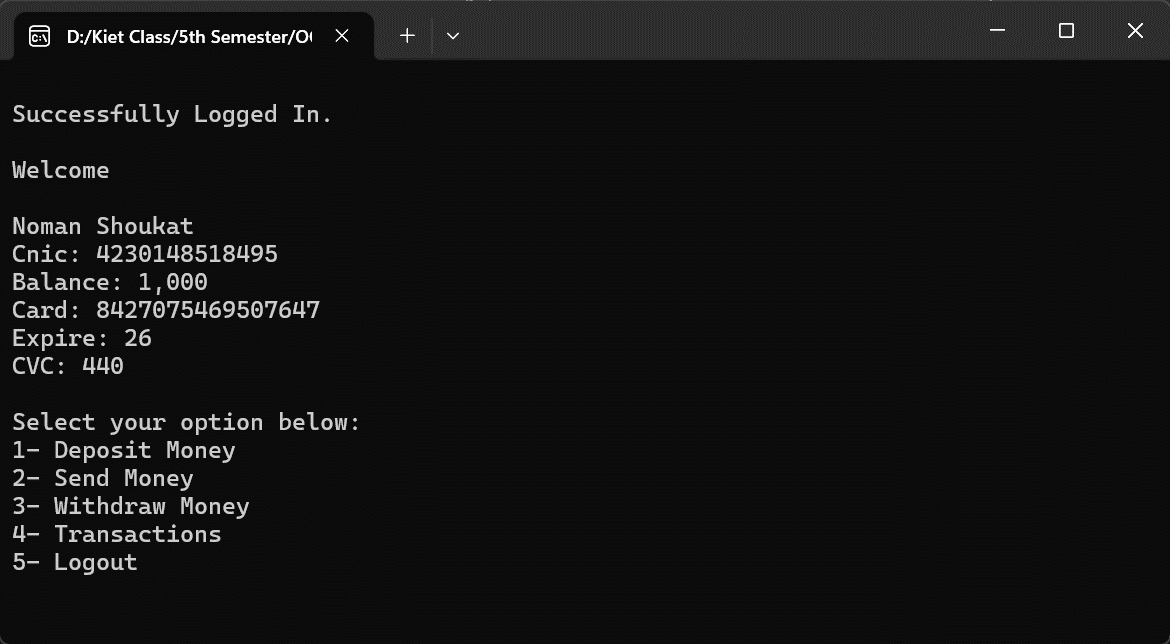
* **Add Money:** Users can add funds to their account.
* **Send Money:** Users can transfer money to other users within the application.
* **Transfer Between Accounts:** Users can transfer money between their own accounts.

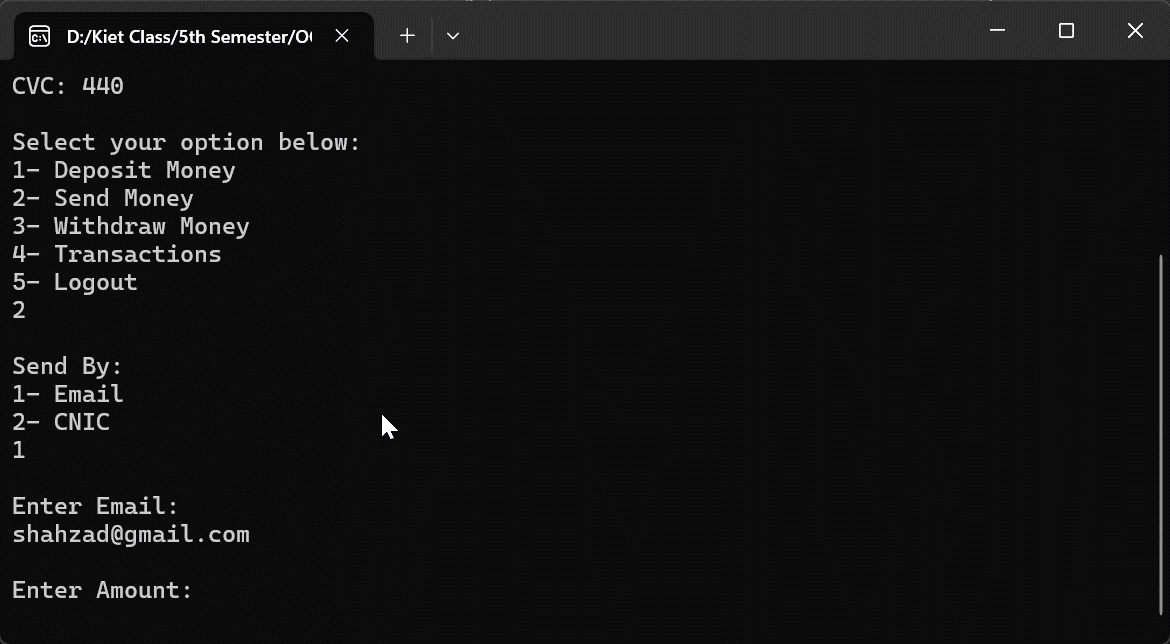
**Transaction History:**

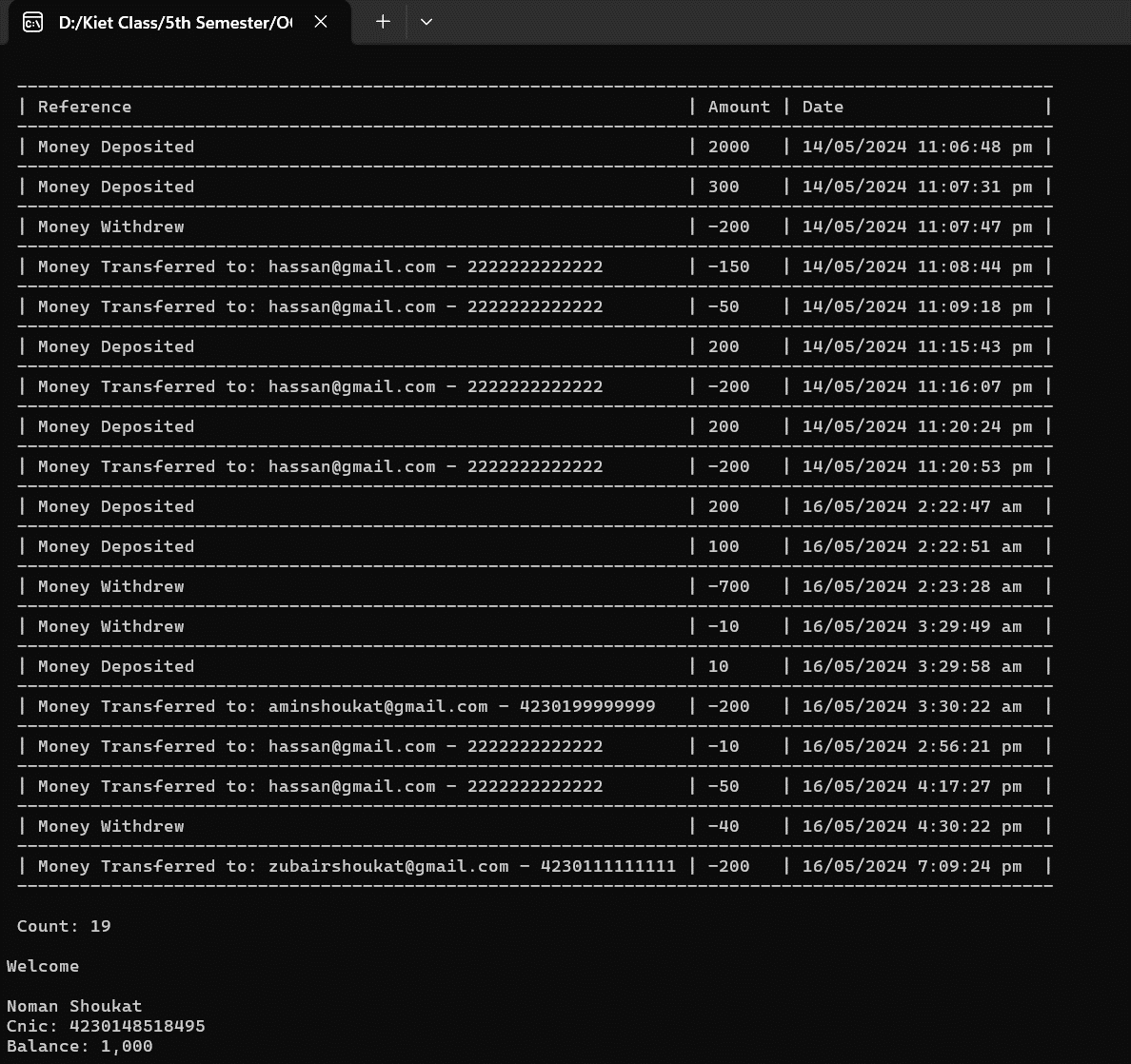
* **Detailed Tracking:** Users can view a comprehensive history of all their transactions.

User Interface









Project Summary

**User Registration:**

* During signup, users will be prompted to enter their email address, password, full name, and CNIC (Computerized National Identity Card).
* The system will perform robust validation to ensure:
  + **Unique email address:** The entered email address does not already exist within the system.
  + **Valid CNIC:** The CNIC format is correct and doesn't belong to an existing user.

**User Login:**

* Upon choosing login, users will be required to enter their registered email address and password.
* The system will validate the entered email address against the internal database to ensure it exists.

**User Dashboard:**

* Logged-in users will be directed to a comprehensive dashboard featuring various options:
  + **Deposit Money:** Allows users to deposit funds into their account.
  + **Withdraw Money:** Enables users to withdraw funds from their account.
  + **Transfer Money:** Facilitates transferring funds to other users' accounts using either their email address or CNIC.
  + **Transaction List:** Provides a detailed breakdown of past transactions.
  + **Logout:** Allows users to securely end their session and exit the application.

**System Architecture:**

* The application utilizes a FileSystem class for managing user data, including functionalities like adding new users, retrieving existing user information, updating user details, and recording transactions.
* Additionally, the system employs a set of specialized validator classes to ensure the accuracy and validity of user input during various actions.
* Upon successful login, user information will be securely stored in a session, making it accessible throughout the application.

Technologies and Usage

* C# version 12.0
* .NET version 8.0

Upon launching the application, you'll be prompted to either log in or create a new account.

Once logged in, you'll be directed to the dashboard.

Use the menu options to navigate between features and perform actions by providing inputs.

Installation, Setup and Code

Ensure you have Visual Studio with .NET Framework or JetBrains Rider installed on your system. Open the solution file (.sln). Build the project. Run the application.