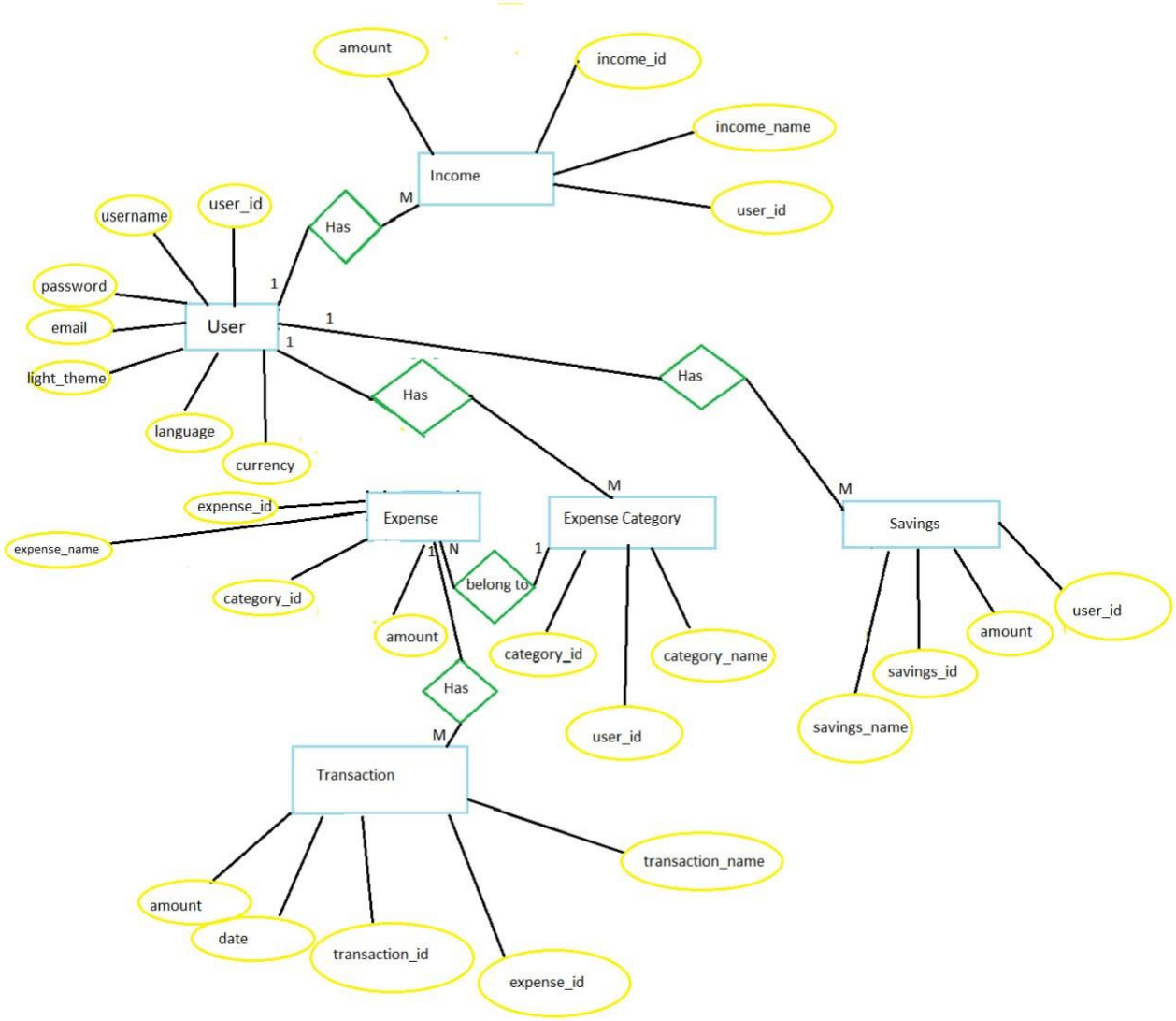
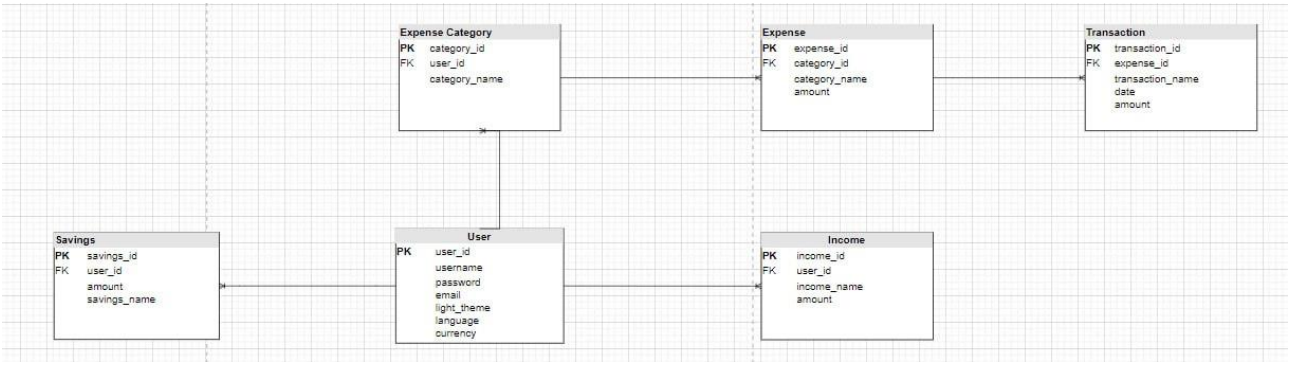
ER-diagram



UML



Code

```
1 reference
static public void CreateDatabase()
{



    using (MySqlConnection connection = new MySqlConnection(connectionString))
    {
        connection.Open();


        string createTableUser =
        @"CREATE TABLE IF NOT EXISTS User(
            id INT AUTO_INCREMENT NOT NULL,
            username VARCHAR(30) NOT NULL,
            CHECK (LENGTH(username) >= 5),
            password VARCHAR(40) NOT NULL,
            CHECK (LENGTH(password) >= 8),
            email VARCHAR(100),
            CHECK(email LIKE '%@%.%'),
            light_theme bool DEFAULT(0),
            language VARCHAR(2) DEFAULT('uk'),
            CHECK (language IN ('ua', 'en')),
            currency VARCHAR(3) DEFAULT('uan'),
            CHECK (currency IN ('uan', 'usd')),
            PRIMARY KEY (id)
        );";

        string createTableIncome =
        @"CREATE TABLE IF NOT EXISTS Income(
            id INT AUTO_INCREMENT NOT NULL,
            income_name VARCHAR(100) NOT NULL,
            CHECK(LENGTH(income_name) >= 5),
            amount INT,
            CHECK(amount > 0 AND amount < 100000000),
            user_id INT,
            PRIMARY KEY (id),
            FOREIGN KEY (user_id) REFERENCES User(id)
        );";

```

```
        string createTableSaving =
        @"CREATE TABLE IF NOT EXISTS Saving(
            id INT AUTO_INCREMENT NOT NULL,
            saving_name VARCHAR(100) NOT NULL,
            CHECK(LENGTH(saving_name) >= 5),
            amount int,
            CHECK(amount > 0 AND amount <= 10000000000000000),
            user_id INT,
            PRIMARY KEY (id),
            FOREIGN KEY (user_id) REFERENCES User(id)
        );";

        string createTableExpenseCategory =
        @"CREATE TABLE IF NOT EXISTS ExpenseCategory(
            id INT AUTO_INCREMENT NOT NULL,
            category_name VARCHAR(100),
            CHECK(LENGTH(category_name) >= 5),
            user_id INT,
            PRIMARY KEY (id),
            FOREIGN KEY (user_id) REFERENCES User(id)
        );";

        string createTableExpense =
        @"CREATE TABLE IF NOT EXISTS Expense(
            id INT AUTO_INCREMENT NOT NULL,
            expense_name VARCHAR(100),
            CHECK(LENGTH(expense_name) >= 5),
            amount INT,
            CHECK(amount > 0 AND amount <= 100000000),
            expense_category_id INT,
            PRIMARY KEY (id),
            FOREIGN KEY (expense_category_id) REFERENCES ExpenseCategory(id)
        );";
```

```
346                 string createTableTransaction =
347                 @"CREATE TABLE IF NOT EXISTS Transaction(
348                     id INT AUTO_INCREMENT NOT NULL,
349                     transaction_name VARCHAR(100),
350                     CHECK (LENGTH(transaction_name) >= 5),
351                     amount int,
352                     CHECK(amount > 0 AND amount <= 100000000),
353                     date TIMESTAMP,
354                     expense_id INT,
355                     PRIMARY KEY (id),
356                     FOREIGN KEY (expense_id) REFERENCES Expense(id)
357                 );";
358
359
360                 using (MySqlCommand command = new MySqlCommand(createTableUser, connection))
361                 {
362                     command.ExecuteNonQuery();
363                 }
364                 using (MySqlCommand command = new MySqlCommand(createTableIncome, connection))
365                 {
366                     command.ExecuteNonQuery();
367                 }
368                 using (MySqlCommand command = new MySqlCommand(createTableSaving, connection))
369                 {
370                     command.ExecuteNonQuery();
371                 }
372                 using (MySqlCommand command = new MySqlCommand(createTableExpenseCategory, connection))
373                 {
374                     command.ExecuteNonQuery();
375                 }
376                 using (MySqlCommand command = new MySqlCommand(createTableExpense, connection))
377                 {
378                     command.ExecuteNonQuery();
379                 }
380                 using (MySqlCommand command = new MySqlCommand(createTableTransaction, connection))
381                 {
382                     command.ExecuteNonQuery();
383                 }
384
385             }
386         }
```

```
0 references
158 static void FillDatabaseWithTestData()
159 {
160
161     using (MySqlConnection connection = new MySqlConnection(connectionString))
162     {
163         connection.Open();
164
165         string insertQuery = "INSERT INTO User (username, password, email,light_theme, language, currency) VALUES (@username, @password, @email,@light_theme,@language,@currency)";
166
167         for (int i = 0; i < 50; i++)
168         {
169
170             using (MySqlCommand insertCommand = new MySqlCommand(insertQuery, connection))
171             {
172
173
174                 insertCommand.Parameters.AddWithValue("@username", $"username {i}");
175                 insertCommand.Parameters.AddWithValue("@password", $"password {i}");
176                 insertCommand.Parameters.AddWithValue("@email", $"user{i}@gmail.com");
177                 insertCommand.Parameters.AddWithValue("@light_theme", false);
178                 insertCommand.Parameters.AddWithValue("@language", "ua");
179                 insertCommand.Parameters.AddWithValue("@currency", "uan");
180                 insertCommand.ExecuteNonQuery();
181             }
182
183             }
184
185         string insertQuery1 = "INSERT INTO Income (income_name, amount, user_id) VALUES (@income_name, @amount, @user_id)";
186
```

```
1 reference
38      static void DisplayUserData()
39      {
40
41
42          using (MySqlConnection connection = new MySqlConnection(connectionString))
43          {
44              connection.Open();
45
46              string query = "SELECT * FROM User";
47
48              using (MySqlCommand command = new MySqlCommand(query, connection))
49              using (MySqlDataReader reader = command.ExecuteReader())
50              {
51                  while (reader.Read())
52                  {
53                      Console.WriteLine($"ID: {reader["id"]}, Username: {reader["username"]}, Email: {reader["email"]}");
54                  }
55              }
56          }
57      }
```

```
0 references
static void Main(string[] args)
{

    CreateDatabase();
    //FillDatabaseWithTestData();
    Console.WriteLine("Displaying User Table Data:");
    DisplayUserData();
    Console.WriteLine();
    Console.WriteLine("Displaying Expense Table Data:");
    DisplayExpenseData();
    Console.WriteLine();
    Console.WriteLine("Displaying ExpenseCategory Table Data:");
    DisplayExpenseCategoryData();
    Console.WriteLine();
    Console.WriteLine("Displaying Income Table Data:");
    DisplayIncomeData();
    Console.WriteLine();
    Console.WriteLine("Displaying Saving Table Data:");
    DisplaySavingData();
    Console.WriteLine();
    Console.WriteLine("Displaying Transaction Table Data:");
    DisplayTransactionData();
    Console.WriteLine();

}
```

Висновок: ми зробили ER-діаграма та UML діаграма, які відображають такі таблиці:

- User
- Expense
- ExpenseCategory
- Income
- Savings
- Transaction

Також був створений проект Console App на платформі .NET Core, який надає можливість здійснювати доступ до цієї бази даних.