1. Project Objective

The purpose of this project is to develop a simple yet functional desktop-based accounting management system that allows users to track their income and expenses efficiently. With separate interfaces for administrators and regular users, the system ensures personalized data access and manipulation. It aims to help individuals monitor their financial transactions, generate summary reports, and visualize spending habits through charts, all within a user-friendly GUI environment built using Java Swing and MySQL.

2. Technologies and Components Used

This project was developed using **Java Swing** for the graphical user interface and **MySQL** as the database management system. The system uses **JDBC (Java Database Connectivity)** for database interaction. Core Swing components such as JTable, JComboBox, JRadioButton, JSpinner, JList, and JTabbedPane were used to build interactive user screens. For visualizing financial data, the **JFreeChart** library was integrated to display dynamic pie charts. The application ensures modularity with proper use of object-oriented programming principles such as inheritance, method overloading, and generic methods.

3. User Roles and Responsibilities

The system distinguishes two user roles with different levels of access and functionality: Admin and Standard User. Each role has specific responsibilities and interacts with different parts of the application.

- Administrator (Admin)

- Has full access to all system functionalities.
- Can manage all registered users:
    - Add new users
    - Update existing user information
    - Delete users from the system
- Has access to all users' transactions.
- Can:
    - View, add, update, and delete any transaction
    - Generate and export detailed financial reports for any user
    - View pie chart comparisons of income vs. expenses for any user
- Has access to theme customization options (light, dark, default).
- Navigates through:

- User Management Panel
- Transaction Management Panel
- Financial Report Panel

➢ Standard User

- Can only access their own data.
- Functionalities include:
  - Add new transactions (income or expense)
  - Update or delete only their own transactions
  - View their transaction history
  - Generate personal monthly financial reports
  - Export reports to .txt format
  - View personal income vs. expense pie chart
- Cannot access other users' data or admin-specific panels.
- Has theme customization options similar to the admin.

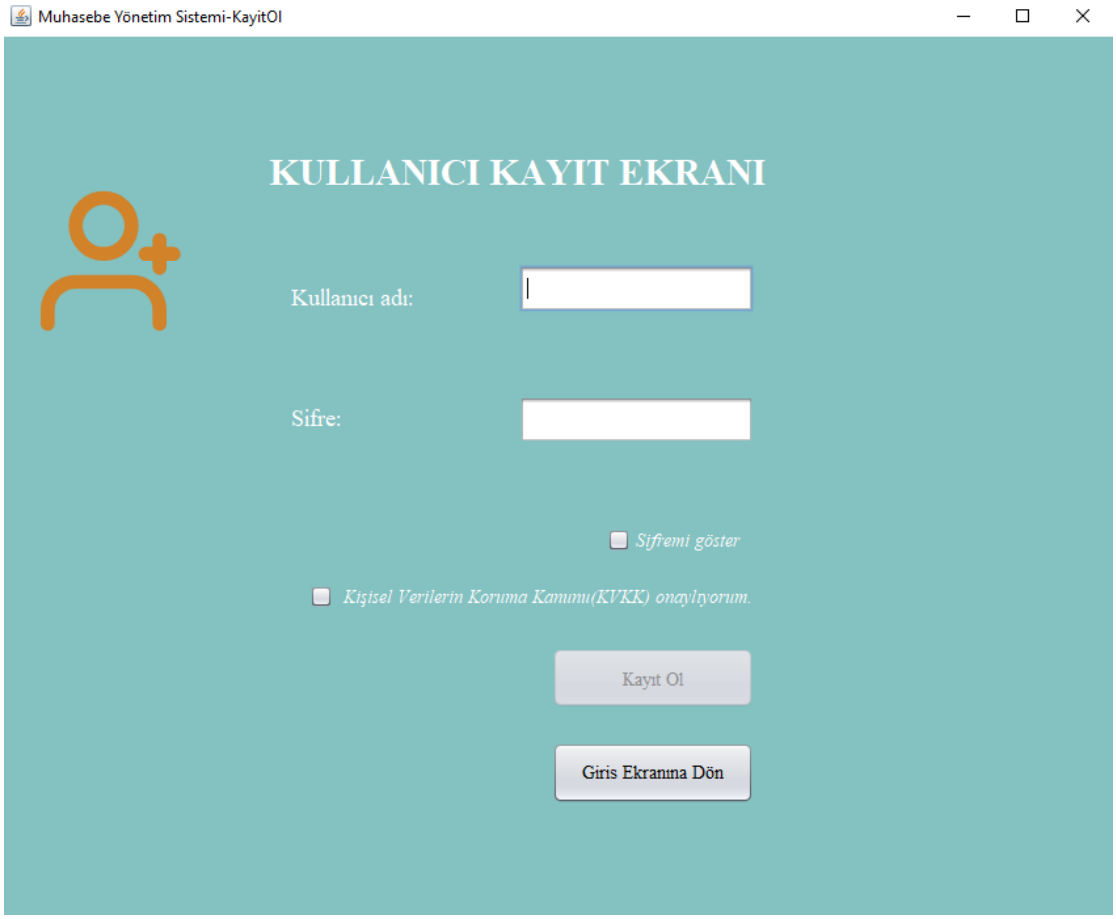
4. System Design

4.1 Login Frame

UI Components and Functionalities:

- **JTextField – Username Input (Kullanıcı Adı):**Accepts the user's username for authentication.
- **JPasswordField – Password Input (Şifre):** Secure password field for user entry.
- **JButton – Login (Giriş):**
    - Validates user credentials against the database.
    - If successful, navigates the user to the corresponding interface:
        - If role = "admin" → AdminPanel
        - If role = "user" → UserPanel
    - If login fails, an error message is displayed.
- **JButton – Register (Kayıt Ol):**
  Redirects the user to the registration screen.
- **Additional Security:**
    - Users are allowed up to 5 login attempts.
    - If the credentials are entered incorrectly 5 times, the login function is temporarily disabled for 15 seconds.

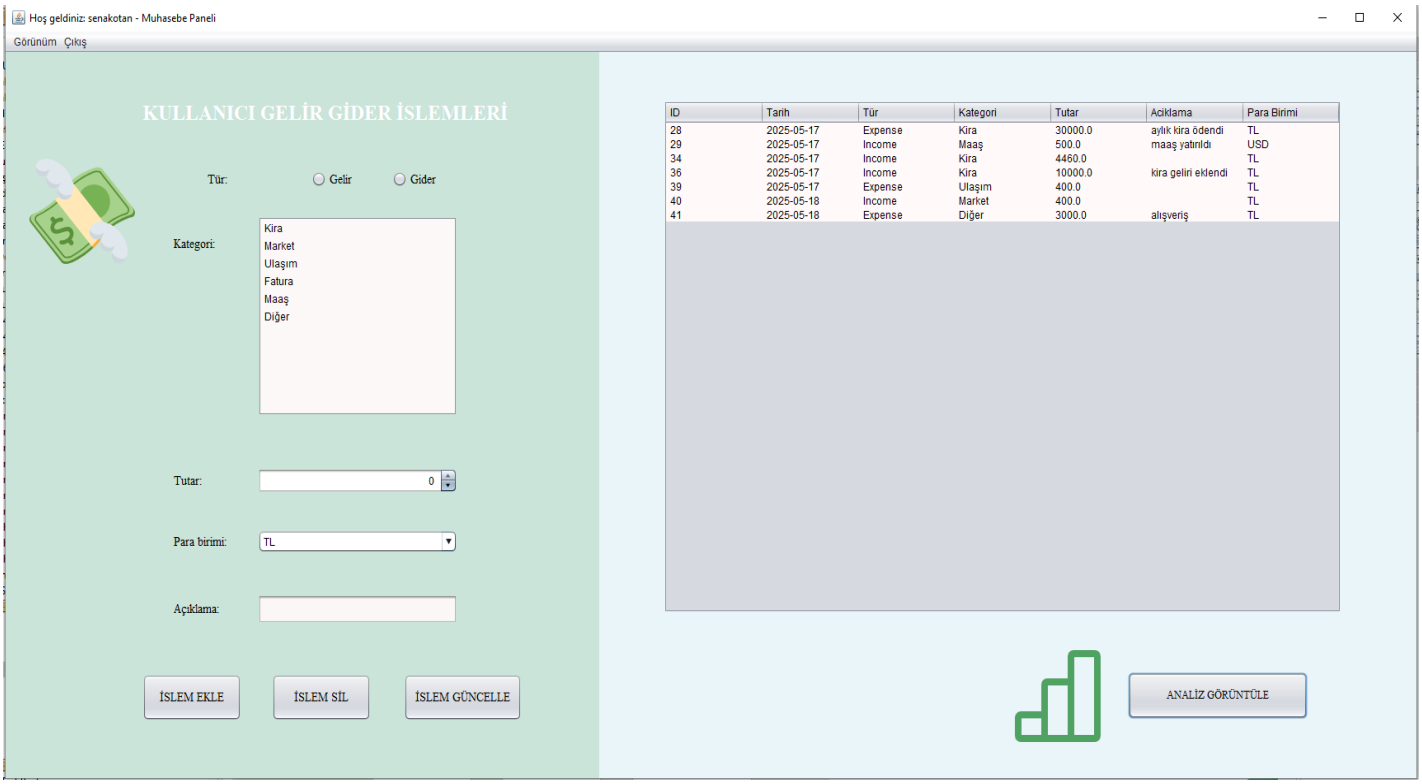
4.2 Register Frame



UI Components and Functionalities:

- **JTextField – Username Input (Kullanıcı adı):** Input field for the user to enter a unique username.
- **JPasswordField – Password Input (Şifre):** Secure field where the user sets a password.
- **JCheckBox – Show Password (Şifremi Göster):** Enables users to view the entered password as plain text.
- **JCheckBox – KVKK Consent:**
  Text: *"Kişisel Verilerin Koruma Kanunu (KVKK) onaylıyorum."*
    - Must be checked before the "Register" button is enabled.
    - Ensures that the user agrees to data processing and storage rules.
- **JButton – Register (Kayıt Ol):**
    - Inserts new user credentials into the database after validating the inputs.
    - Disabled unless KVKK checkbox is selected.
- **JButton – Back to Login (Giriş Ekranına Dön):**
  Navigates back to the login screen without performing any action.

4.3 User Frame

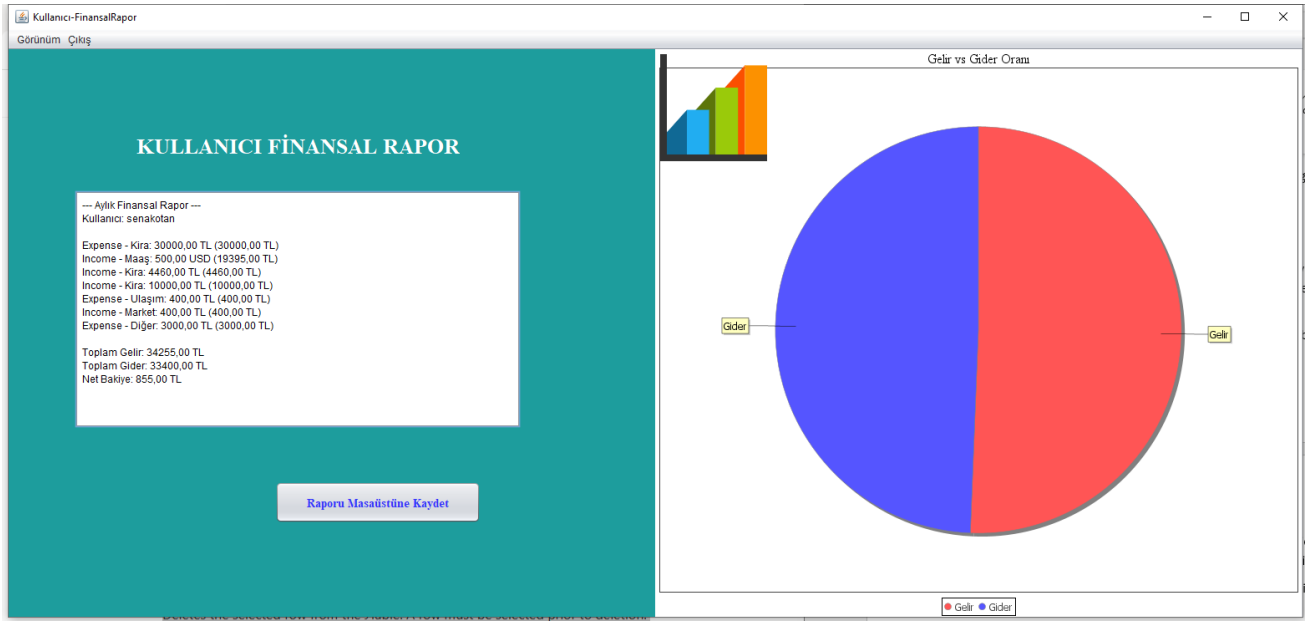


UI Components and Actions:

- **RadioButtons(Income-Expense)** :Allows the user to select the transaction type as either income or expense.
- **JList (Kategori):** A predefined category list (e.g., Kira, Market, Maaş) for the user to select from.
- **JSpinner (Tutar):** Numeric spinner that allows input of the transaction amount only as valid numeric data.
- **JComboBox(ParaBirimi):** Dropdown menu for selecting the transaction currency (TL, USD, EURO).
- **JTextField(Açıklama):** Free text input for providing a brief description of the transaction.
- **JButton- Add Transaction ("İşlem Ekle"):** Validates all fields and inserts a new record into the database. It performs regex checks on the category name and verifies that the amount is numeric.
- **JButton- Delete Transaction ("İşlem Sil"):** Deletes the selected row from the JTable. A row must be selected prior to deletion.

- **JButton - Update Transaction ("İşlem Güncelle"):** Updates the selected transaction based on current form input. Changes are saved back to the database after validation.
- **JTable ("İşlem Listesi"):** Displays the user's transaction history. Allows row selection for editing or deletion.
- **JButton- View Analysis ("Analiz Görüntüle"):** Navigates to the financial analysis screen that visualizes income vs. expense data using charts.

4.4 User Report Frame

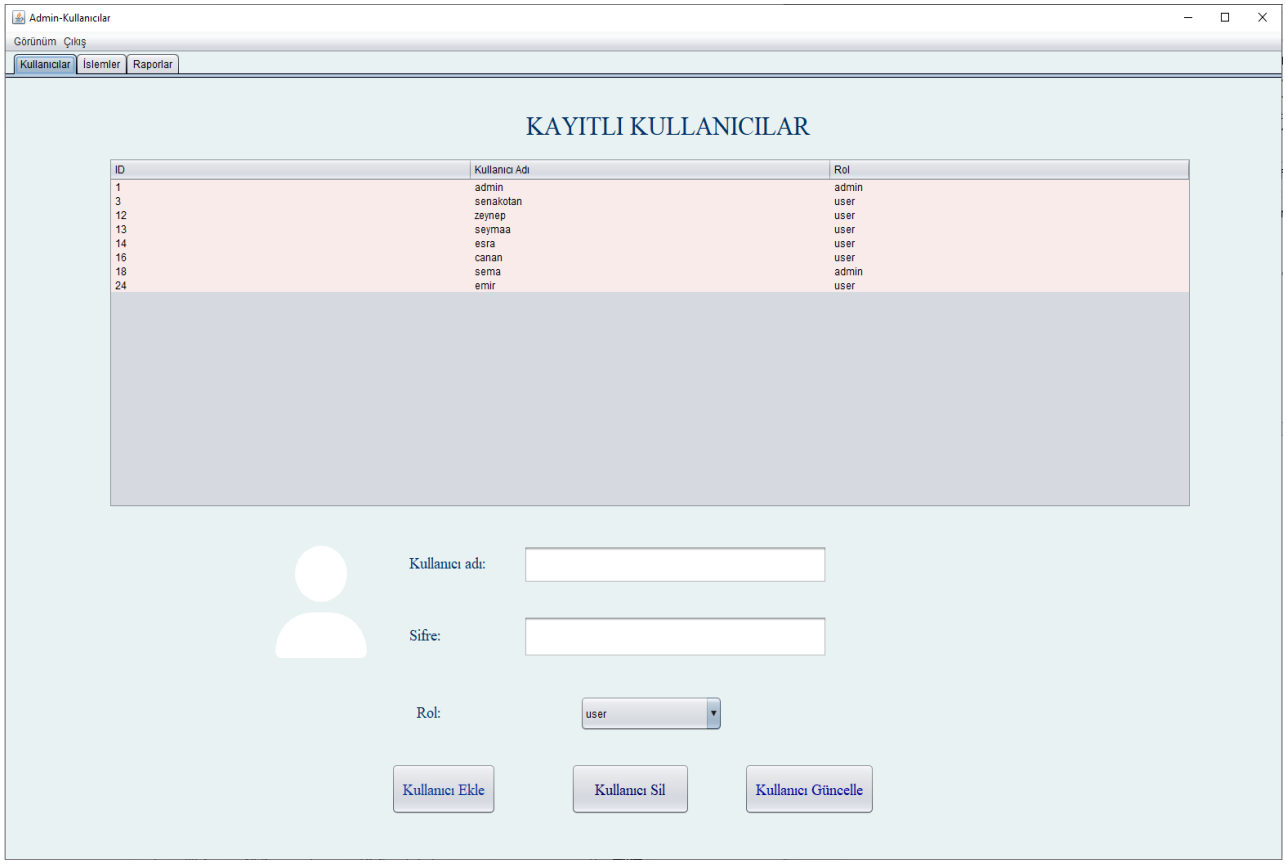


UI Components and Functionalities:

- **JTextArea (Report Display):** Displays a detailed monthly financial report for the selected user. This area is **non-editable** and generated directly from the database. Includes both categorized transaction breakdowns and total income, total expense, and net balance. User cannot modify this field manually.
- **JButton – Export Report to Desktop ("Raporu Masaüstüne Kaydet"):** Saves the text report in .txt format to the user's desktop. Includes all calculated totals and breakdowns.
- **JPanel – Pie Chart Panel (Gelir vs. Gider Oranı):** A chart panel that visualizes the proportion of income vs. expense using JFreeChart.
  - Red represents "Gelir (Income)"

- Blue represents “Gider (Expense)”
-Legend is displayed below the chart
-Chart is centered and slightly enlarged for readability.

4.5 Admin - User Management Frame



UI Components and Functionalities:

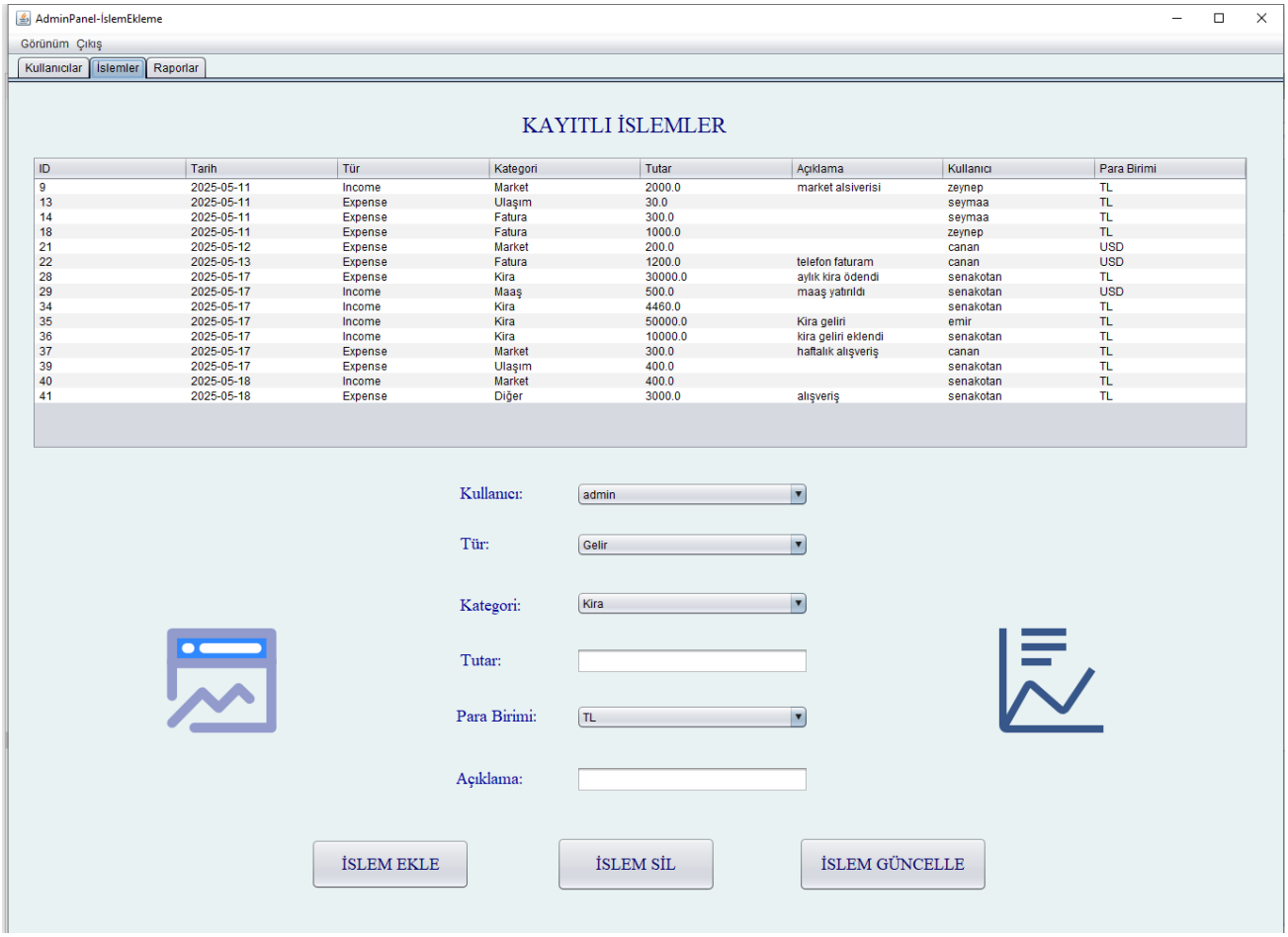
- **JTable – User List Table:**
  Displays all registered users along with their IDs, usernames, and roles.
  Automatically refreshed after insert/update/delete operations.
  Red background (ID rows) helps differentiate entries visually.
- **JTextField – Username & Password Inputs:**
  Used to enter or update user credentials.
  Plain text for demonstration, but can be replaced with JPasswordField.
- **JComboBox – Role Selector:**
  Allows selection between user and admin roles.
  Dynamically sets user permissions.

- **JButtons – Action Controls:**
    - **Add User (Kullanıcı Ekle):** Inserts a new user after input validation.
    - **Delete User (Kullanıcı Sil):** Removes the selected row from the user list.
    - **Update User (Kullanıcı Güncelle):** Updates selected user's information.
- **Icon (User Silhouette):**
  Decorative visual representing user management functionality.

Top Menu Bar – Navigation:

- **Users Tab (Kullanıcılar):**
  Opens the current user management screen.
- **Transactions Tab (İşlemler):**
  Navigates to the screen for managing user income and expenses.
- **Reports Tab (Raporlar):**
  Navigates to the admin report analysis screen.
- **Appearance Menu (Görünüm):**
  Contains theme customization options:
    - Light Mode (Açık Mod)
    - Dark Mode (Koyu Mod)
    - Default Mode (Varsayılan Görünüm)
      All themes adjust background, text, and table colors accordingly.
- **Exit Menu (Çıkış):**
  Contains options to:
    - Exit the application
    - Return to the login screen

4.6 Admin- Transaction Management Frame



UI Components and Functionalities:

- **JTable – Registered Transactions Table:** Displays all financial transactions across users with detailed columns:
  ID, Date, Type, Category, Amount, Description, Username, and Currency.
  Automatically refreshed after each operation (add, delete, update).
  Enables selection for update/delete actions.
- **JComboBox – User Selector:** Lists all registered users. The admin can choose a user to assign the transaction to.
- **JComboBox – Transaction Type (Tür):** Allows selection of transaction type:
  - Income (Gelir)
  - Expense (Gider)
- **JComboBox – Category Selector:** Lists predefined transaction categories such as:

    - Rent (Kira), Market, Transportation (Ulaşım), Utility (Fatura), Salary (Maaş), Other (Diğer)

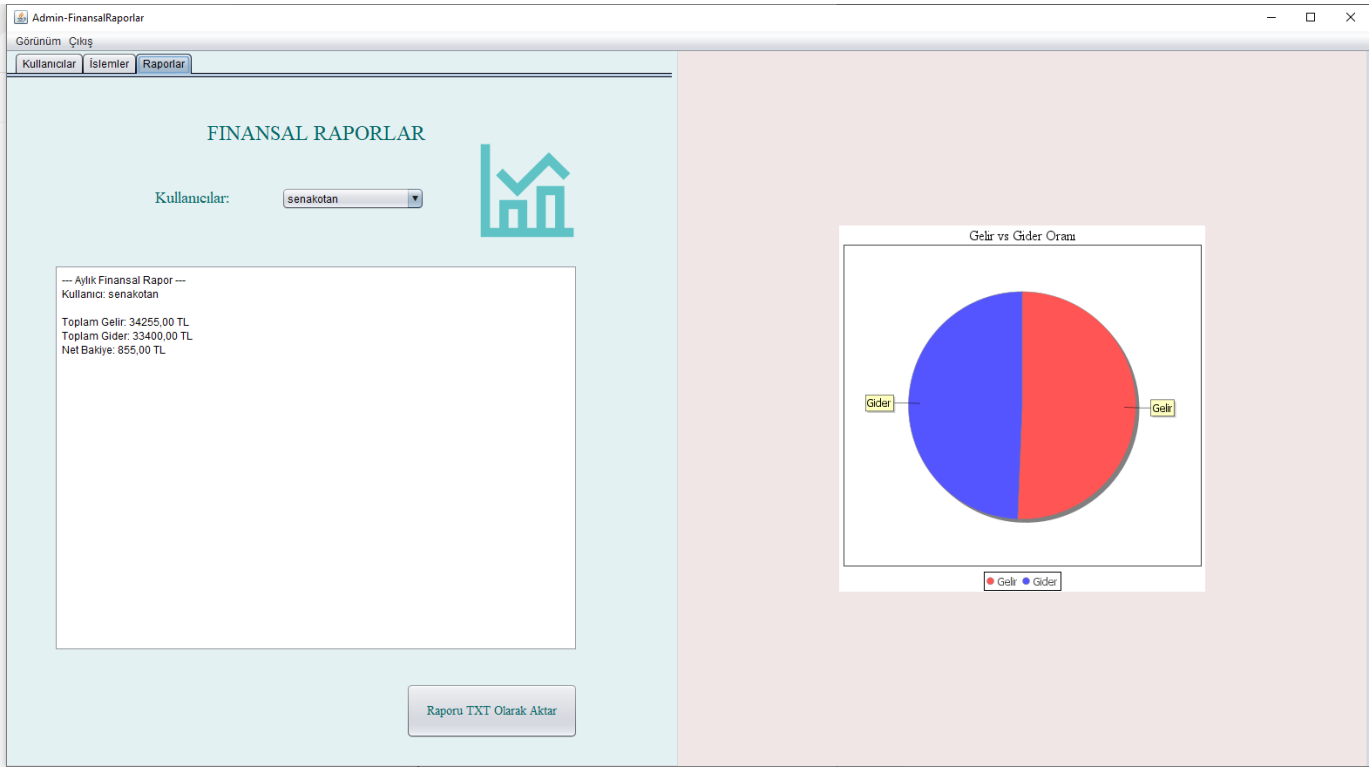
- **JTextField – Amount Field (Tutar):** Takes the numeric amount for the transaction.
  Validated as a number with optional decimal places.
- **JComboBox – Currency Selector:** Lists supported currencies: TL, USD, EURO.
- **JTextField – Description Field (Açıklama):**
  Allows entering brief notes or details about the transaction.
- **JButtons – Operation Controls:**
    - İŞLEM EKLE (Add Transaction): Adds a new record after input validation.
    - İŞLEM SİL (Delete Transaction): Deletes the selected row.
    - İŞLEM GÜNCELLE (Update Transaction): Updates selected row with new data.
- **Icons (Left & Right Panels):**
    - Left: Represents file input or transaction
    - Right: Represents financial reporting or analytics

Top Menu Bar – Navigation:

- **Users Tab (Kullanıcılar):**
  Navigates to the Admin - User Management screen.
- **Transactions Tab (İşlemler):**
  This is the current screen — allows full control over financial entries.
- **Reports Tab (Raporlar):**
  Opens the Admin - Financial Reports screen with graphical summary.
- **View Menu (Görünüm):**
  Contains menu items:
    - Açık Mod (Light Mode)
    - Koyu Mod (Dark Mode)
    - Varsayılan Görünüm (Reset theme): Changes the appearance of the entire UI.
- **Exit Menu (Çıkış):**
  Contains:
    - Exit Application

- Return to Login Screen

4.7 Admin – Financial Report Frame



UI Components and Functionalities:

- **JComboBox – User Selector (Kullanıcılar):**
  Allows the admin to select any registered user from the dropdown. The financial report updates dynamically according to the selected user.
- **JTextArea – Monthly Financial Summary:**
  Displays the selected user's monthly financial statistics:
  - Total Income (Toplam Gelir)
  - Total Expense (Toplam Gider)
  - Net Balance (Net Bakiye)
    This area is non-editable; the content is automatically generated based on user data.
    Cannot be manually changed by the user or admin.
- **JButton – Export to TXT: (Raporu TXT Olarak Aktar)**
  - Saves the report content (text only) into a .txt file on the user's desktop.
  - This allows archiving or printing of monthly reports.

- **JFreeChart – Pie Chart Panel:**
    - Visually represents the ratio of income to expense for the selected user.
    - Red slice: Income
    - Blue slice: Expense
    - Automatically updates when a new user is selected from the dropdown.
- **Chart Legend and Labels:**
    - Labels (Gelir, Gider) appear around the pie chart.
    - Color-coded and balanced in proportion to the amounts.

Top Menu Bar – Navigation Overview:

- **Users Tab (Kullanıcılar):**
  Takes the admin to the User Management screen.
- **Transactions Tab (İşlemler):**
  Opens the Transaction Management screen for reviewing or editing user transactions.
- **Reports Tab (Raporlar):**
  This is the current screen. Provides detailed financial reporting and analysis.
- **View Menu (Görünüm):**
  Includes:
    - Light Mode
    - Dark Mode
    - Default View
- **Exit Menu (Çıkış):**
  Includes:
    - Exit system
    - Return to login screen

5. Database Design

The system uses a relational database designed in **MySQL**, consisting of two main tables:

1. users – for managing user credentials and roles.
2. transactions – for storing individual financial records linked to users.

5.1 Users Table Description

This table is used to store registered users' identity information and their roles within the system. Each user is assigned a unique ID, and the username must be unique for login purposes. This table is essential for authentication and role-based access control.

Columns and Explanations:

- **id (INT)**
  A unique identifier automatically assigned to each user. It acts as the **primary key** of the table.
- **username (VARCHAR)**
  The unique username chosen by the user. It is used during the login process and must not be duplicated.
- **password (VARCHAR)**
  The user's password stored as plain text for demonstration purposes.
- **role (VARCHAR)**
  Specifies the role of the user. It can be either 'admin' for administrators or 'user' for standard users. This field determines access privileges throughout the application

	id	username	password	role
▶	1	admin	1234	admin
	3	senakotan	Kotan1234.	user
	12	zeynep	Zeynep123.	user
	13	seymaa	Seyma123.	user
	14	esra	Esra123.	user
	16	canan	12345	user
	18	sema	Kotan123.	admin
	24	emir	Emir12345.	user
*	NULL	NULL	NULL	NULL

5.2 Transactions Table Description

This table stores all financial transactions recorded by users. Each entry represents a single income or expense operation and is directly linked to a specific user through the user_id foreign key. This structure allows the system to generate personalized reports and support admin-level oversight.

Columns and Explanations:

- **id (INT)**
  A unique identifier automatically assigned to each transaction. It is the primary key of the table.
- **date (DATETIME)**
  The exact date and time when the transaction was recorded. Useful for filtering and monthly reporting.
- **type (VARCHAR)**
  Indicates whether the transaction is an "Income" or an "Expense". This field is used for categorizing financial data.
- **category (VARCHAR)**
  Specifies the type of the transaction, such as "Market", "Rent", "Utilities", "Salary", etc. Categories help in organizing and analyzing user behavior.
- **amount (DECIMAL)**
  The monetary value of the transaction. It can include decimal values and is validated before insertion.
- **description (VARCHAR)**
  A short explanation or note about the transaction. This is optional but helps provide context for reports.
- **user_id (INT)**
  A foreign key that links the transaction to a user in the users table. It ensures that each transaction belongs to a valid user.
- **currency (VARCHAR)**
  The currency type of the transaction, such as "TL", "USD", or "EURO". It allows for multi-currency financial tracking.

id	date	type	category	amount	description	user_id	currency
9	2025-05-11 00:00:00	Income	Market	2000	market alsiverisi	12	TL
13	2025-05-11 00:00:00	Expense	Ulaşım	30		13	TL
14	2025-05-11 00:00:00	Expense	Fatura	300		13	TL
18	2025-05-11 21:16:25	Expense	Fatura	1000		12	TL
21	2025-05-12 00:00:00	Expense	Market	200		16	USD
22	2025-05-13 00:00:00	Expense	Fatura	1200	telefon faturam	16	USD
28	2025-05-17 00:00:00	Expense	Kira	30000	aylık kira ödendi	3	TL
29	2025-05-17 00:00:00	Income	Maaş	500	maaş yatırıldı	3	USD
34	2025-05-17 00:00:00	Income	Kira	4460		3	TL
35	2025-05-17 00:00:00	Income	Kira	50000	Kira geliri	24	TL
36	2025-05-17 00:00:00	Income	Kira	10000	kira geliri eklendi	3	TL
37	2025-05-17 00:00:00	Expense	Market	300	haftalık alışveriş	16	TL
39	2025-05-17 00:00:00	Expense	Ulaşım	400		3	TL
40	2025-05-18 00:00:00	Income	Market	400		3	TL
41	2025-05-18 00:00:00	Expense	Diğer	3000	alışveriş	3	TL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. Code Architecture

The project follows an Object-Oriented Programming (OOP) structure in Java, emphasizing modularity, reusability, and maintainability. Each part of the application is separated into frames and helper classes based on responsibility.

Main Components and Their Roles:

- **DBHelper**
    - A utility class responsible for managing the database connection.
    - Provides a reusable getConnection() method for all database-related operations.
- **Login Frame**
    - Handles user authentication.
    - Directs the user to the appropriate panel (AdminPanel or UserPanel) based on their role.
- **Register Frame**
    - Allows new users to sign up.
    - Includes input validation and a mandatory data consent checkbox.
- **User Frame**
    - Enables users to:
        - Add, update, and delete their own financial transactions.

        - View their transaction list.
        - Access graphical financial analysis.
    - Connected directly to the transactions table.
- **Users Report Frame**
    - Displays the user's financial report in a text area.
    - Allows export of report to .txt format.
    - Shows income vs. expense as a pie chart using JFreeChart.
- **Admin Frame**
    - Acts as the main menu for admin users.
    - Provides tabbed navigation to:
        - User management
        - Transaction management
        - Report generation
    - Contains a theme switcher (Light, Dark, Default).
- **Admin Transactions Frame**
    - Allows the admin to view and manage all users' transactions.
    - **Admin can:**
        - Add a transaction to any user
        - Edit or delete any transaction
- **Admin Reports Frame**
    - Allows the admin to select any user and generate their financial report.
    - Contains both text summary and graphical (chart) analysis.
    - Supports export functionality to .txt format.

OOP Concepts Used:

- **Inheritance:**

  Common properties and methods shared across frames have been successfully moved into a BaseFrame superclass to ensure reusability and maintainability.

- **Method Overloading:**

Method Overloading is implemented in the AdminRaporlar frame to display user messages in different contexts. The overloaded method gosterMesaj has two versions:

  - gosterMesaj(String mesaj) – Displays a simple message dialog.
  - gosterMesaj(String mesaj, String kullanici) – Displays a personalized message including the user's name.

  This allows the same method name to be used with different parameter lists, improving code readability and reusability depending on the message context.

- **Generic Methods:**
  - The fillComboBox<T>() generic method was implemented within the project to populate JComboBox elements across different frames. This utility method allows dynamic and type-safe insertion of data into dropdowns, ensuring reusability and consistency throughout the application.

7. Error Handling and Testing Considerations

While the application implements a basic retry mechanism for login attempts (users are temporarily locked out for 15 seconds after 5 failed login attempts), a more comprehensive error-handling strategy is essential for robustness and maintainability.

Exception Handling with try-catch

Throughout the project, try-catch blocks are used to gracefully handle runtime exceptions, especially during database interactions and user input operations. Key areas where exceptions are caught include:

- Database Connectivity Errors using SQLException
- Invalid User Inputs such as non-numeric values in amount fields
- File Export Failures during report generation

This ensures that unexpected failures do not crash the program and that users receive meaningful feedback.

Validation Controls

To prevent invalid data entry:

- Regex patterns are used to validate category inputs.
- Numeric spinners and formatted fields restrict amount values to valid decimal numbers.
- Mandatory fields are checked before any database insertion or update.