

Alternative Banking System

A Client-Server Java Desktop Application

Client Side:

1. Admin Operations:

- There can be only one Admin in the system, so I've allowed only one such entry.
On the main screen, there are two tables with information about all the loans and customers in the system, as mentioned in the exercise document.
- The admin can increase the "Yaz" (יחידת זמן) using the **"Increase Yaz" button**.
- We've added a new feature in this exercise called timeline navigation, which is found under the "Timeline" tab. With this feature, you can navigate through all the days from the system's inception until the present day.
Clicking the **"Rewind"** button will put the system in this mode. The admin can pick a "Yaz" value to rewind the system to using a slider and then click the "Change view" button.
After doing this, the whole system enters the "Rewind" mode, and all \ buttons are locked for both customers and the admin.
It allows the admin to observe (and only observe) the status of loans and finances for all customers for each "Yaz".
To stop the "Rewind" mode, the admin presses **"Stop."**

2. Customer Operations:

- When opening the system as a customer, you need to type in your name and log in. Each customer can be logged in once.
- **Deposit\ Withdraw money from one's account:**
A pop-up window will appear prompting you to input the deposit or withdrawal amount, along with displaying your current account balance. If you enter a non-numeric value or an amount exceeding your available funds, an error message will be displayed.
- **Loans Info:**
To see information about loans you've either given or received, just head to the "Information" tab on your screen. You'll find neatly organized tables with all the details.
- **Account Transactions:**
For a closer look at your account transactions, go to the same "Information" tab. Here, you can click on the deposit or withdrawal buttons to see a record of your financial activity.
- **Notifications Hub:**
To stay updated on payments and reminders, visit the "Payment" tab. This is where you'll find the Notifications Center, which keeps you informed with messages regarding your financial matters.
- **Making Payments:**
If you need to make payments for active loans or those in the "Risk" state, you can do so within the "Payment" tab as well. Here, you'll see a list of your active loans and those that are considered "Risky." To make a payment, simply select the loan you'd like to pay, then choose between **"Pay single payment"** or **"Pay full loan,"** depending on your preference.
- For active loans, you can only pay the set amount on the specified "Yaz" date.

- If the loan is in the "Risk" state, you have the flexibility to choose the payment amount, much like depositing or withdrawing money. In this state, you can make payments on any "Yaz" date you prefer.
- Additionally, you can use the "**Pay full loan**" option whenever it suits you to pay off the entire loan balance.

Information

Scramble

Payment

Loans as a borrower

ID	Owner	Category	Principal	Loan Term	Interest rate	Payment Interval	Starting Date	Termination Date	Next f
I don't have mo...	Tikva	Overdraft cover	4000.0	10	3.0	1	1	0	2

share of an existing loan from a lender who is selling their loans. The loans displayed in the table are those in which the customer is not already participating as a lender and have not been listed for sale by the customer. Nevertheless, it's conceivable that the same loan the customer is offering for sale may appear if another lender is selling their portion of it. If there isn't enough money in their balance, an error message will be displayed. Otherwise, the loan will be added to their assets.

- **Create A New Loan:**

The customer has the ability to initiate the creation of a new loan by clicking on a designated button and then proceeding to complete the required form. However, it's important to note that the option to create a new loan becomes available only after uploading a minimum of one XML file. This XML file serves as the means through which the system gains access to and populates the available loan categories.

- **Load A new Loan:**

Customers have the option to import an XML file that includes data regarding loans and categories. In cases where an uploaded XML file doesn't adhere to the exercise instructions or contains a loan already existing in the system, the system will refuse the entire file upload and provide an error message that corresponds to the issue.

Server Side:

For every data request from the logical part (the Engine) or the execution of actions that require updates, the requests pass through the server. Here are some examples of server classes we've implemented:

- **Transaction Servlet** – This class is responsible for executing deposit/withdrawal transactions and loan payments.
- **Login Servlet** – This class ensures that a customer can log in only once with a unique username.
- **Promote Yaz Servlet** – This class handles notifying the engine about advancing the timeline (Yaz) and performing associated tasks such as saving the current engine state for Rewind, sending payment notifications to customers, changing loan statuses in some cases, and more.
- **Create New Loan Servlet** – This class acts as an intermediary between the UI component, where an XML file path is received, and the system engine, where the file is loaded and a new loan is created.
- **Admin View Servlet** – This class is responsible for retrieving all information about customers and existing loans in the system, which is displayed in the admin interface.

Main Classes:

Our project consists of several modules:

Client Server :

As previously detailed, this module is responsible for handling communication between clients and the server.

Customer , Admin:

There are two separate applications (and two separate modules) - one for customers and one for administrators (admins). We've provided details above about each of these applications.

UI:

This module contains code shared between both the customer and admin applications. One of the key shared components is the header, which includes:

- A greeting message with the customer's or admin's name for identification.
- The system's mode (regular/rewind).
- The current timeline (Yaz).

Engine:

Engine - The class responsible for the logic of the alternative bank. This class contains public methods that provide all the necessary services for the bank as specified in this exercise. These services include loading an XML file containing information about loans and customers in the system, displaying information about loans/customers, depositing/withdrawing money from a bank account, matching lenders with loans, and handling loan repayments.

Customer - This class consolidates all the relevant information for a customer. The class holds data members of type Account (see details below) for each customer, as well as two lists of loans: one where the customer is a borrower (borrower's loans) and another where the customer is a lender (lender's loans).

Account - A class for managing a customer's bank account. It contains the customer's capital (balance) and a list of transactions performed within the system. This class actually performs actions related to withdrawing and depositing money for various system needs.

Loan - A class for managing loans. Each loan has a unique identifier (loanId). Additionally, this class contains the following data members:

- owner: The customer who is the loan's borrower.
- participants: A list of lenders participating in this loan.
- category: The loan category.
- principal with interest: The total loan amount (principal + interest) that the borrower needs to repay.
- interest rate: The loan's interest rate.
- loan term: The expected loan duration (how many Yaz it takes for the payer to complete all payments; exceptions may occur if the payer fails to pay on time).
- payment interval: How often payments are collected from the borrower.
- starting date: The date when the loan became active.
- next repayment: The next date when payment is due.
- status: The loan's status, which can be one of the following: new, pending, active, risk, finished.
- history: A list of payments made for the loan.
- paid principal/interest: The sum of principal and interest paid from the loan.

- unpaid principal/interest: The remaining sum of principal and interest to be paid from the loan.
- unpaid payments: A list of payments that haven't been made on time. As long as there are items in this list, the loan is in the "Risk" state. After the customer pays all overdue payments, the contents of this list are deleted (the debt is settled).
- pending principal: The amount required to turn a loan from pending to active.
- termination date: The actual end date of the loan (not the original term + Yaz but the date when all loan payments have been made).
- new previous engine: A list holding the "past engines" of the application, i.e., the historical records of customers and loans.

Loan Action / Action - These classes contain the necessary information to be stored in the "history" of actions performed in a customer's account or a loan.

Message - This class aggregates information for notifying a customer about loan payments they need to make. It includes the loan name, payment date, and payment amount.