Web Services

Personal Finance Management Tool

E. F. Tebebu[1]

[1] Faculty of Informatics
Université Toulouse Capitole

April 19, 2024

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

Functional perimeter

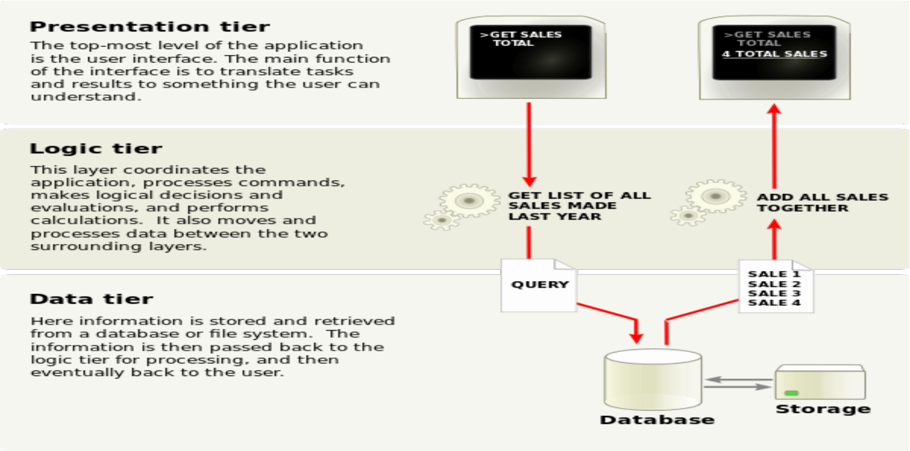
This project aimed to create a personal finance management tool that can

- Let users sign-in and sign-up.
- Let users view their expenses and incomes *vis-à-vis* their budgeting and savings goals.
- Let users modify, delete, add their expenses and incomes
- And, finally, set their budget and savings goals.

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

Three-Layer Architecture



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Figure: © Wikipedia

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

Modelling

This project is structured following the three-layer architecture

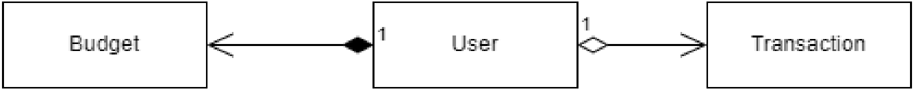


Figure: UML Class Diagram

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

API

The API performs all CRUD operations on the database.

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

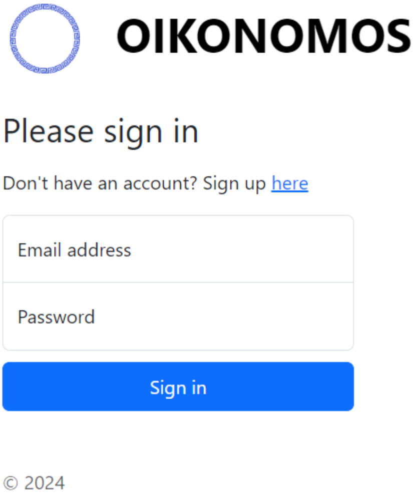
Open API Specification

The Open API Specification can be found after running the project at <url>:swagger-ui/index.html

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

Sign-in



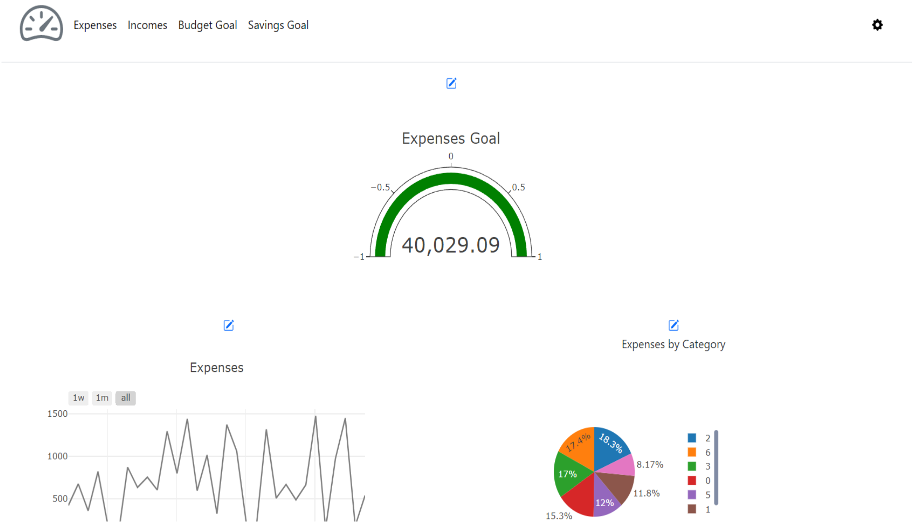
OIKONOMOS

Please sign in

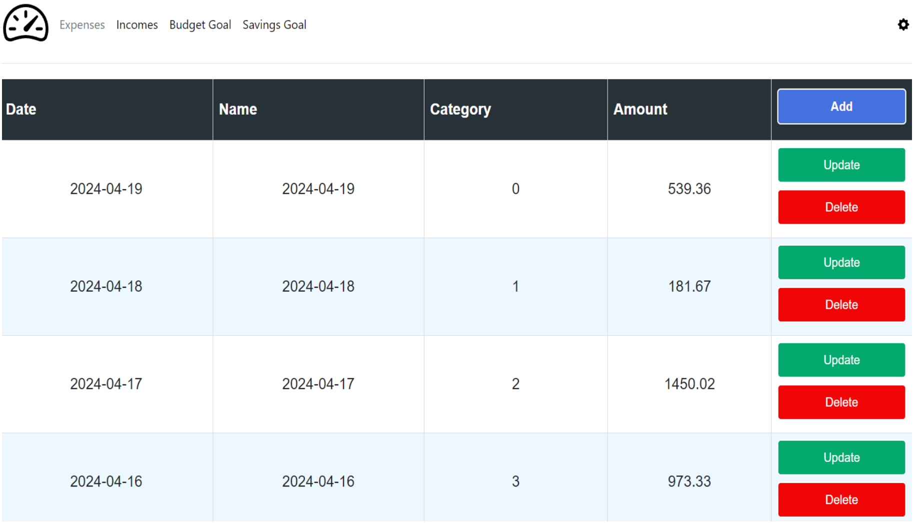
Don't have an account? Sign up here

© 2024

Dashboard


Expenses
Incomes
Budget Goal
Savings Goal
Expenses Goal
0
−0.5
0.5
−1
1
40,029.09
Expenses
1w
1m
all
1500
1000
500
Expenses by Category
17.4%
18.3%
8.17%
17%
11.8%
12%
15.3%
🟦 2
🟧 6
🟩 3
🟥 0
🟪 5
🟫 1

CRUD Table(s)



Date	Name	Category	Amount	Add
2024-04-19	2024-04-19	0	539.36	Update Delete
2024-04-18	2024-04-18	1	181.67	Update Delete
2024-04-17	2024-04-17	2	1450.02	Update Delete
2024-04-16	2024-04-16	3	973.33	Update Delete

Table of Contents

1. Functional perimeter
2. Architecture
3. Modelling
4. API
5. Open API Specification
6. Examples
7. Launching the server

Launch

To run this app,

- Make sure to have a running MongoDB server on mongodb://localhost:27017/oikonomos with MongoDB Compas or the provided url in Mongo Atlas.
- Then launch mvn clean spring-boot:run.