

Design

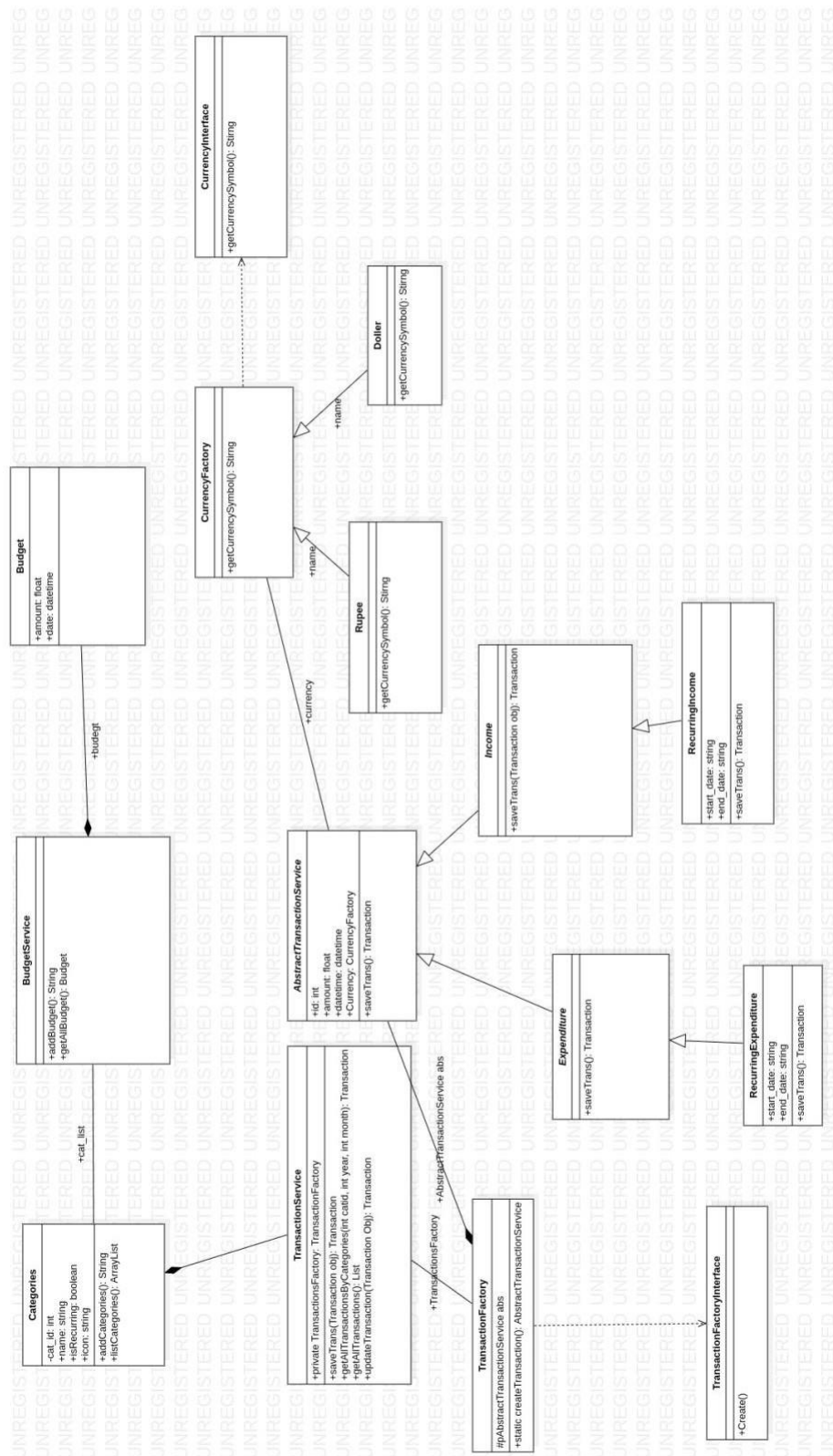


Figure 1 Overall Backend Class Diagram of the Application

Implementations

A couple of places with tight coupling were discovered while evaluating the initial class diagram for the implementation. The following design patterns were used to maintain a loose coupling, high cohesion and improve the code readability:

1. Factory pattern

- To get the transaction type object such as Income or expenses or recurring expense or recurring income. **TransactionFactory** class creates the necessary objects.
- To generate the currency symbol such as rupee and dollar. **CurrencyFactory** class was used to create the relevant currency symbol.

2. Bridge pattern

- **AbstractTransactionService** class and **TransactionController** have an attribute “Currency” that can be used to generate the relevant symbol based on the parameters.

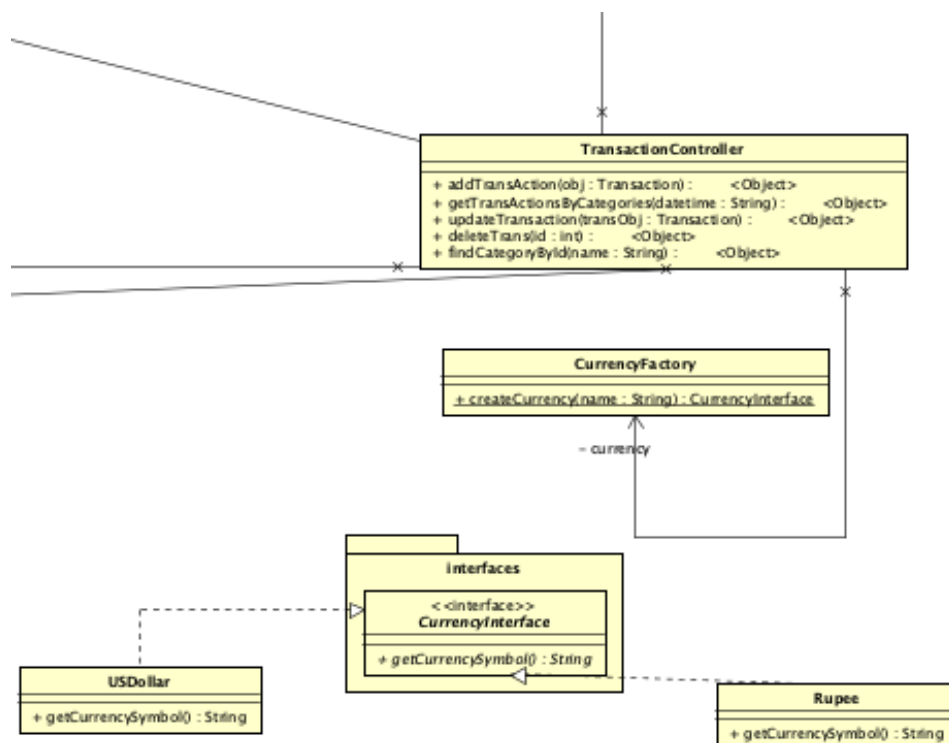


Figure 2 Example use of Bridge and Factory Pattern in the Application

3. Abstract pattern

- **AbstractTransactionService** class was implemented to abstract the two transaction types found throughout the system namely Income and Expenses.

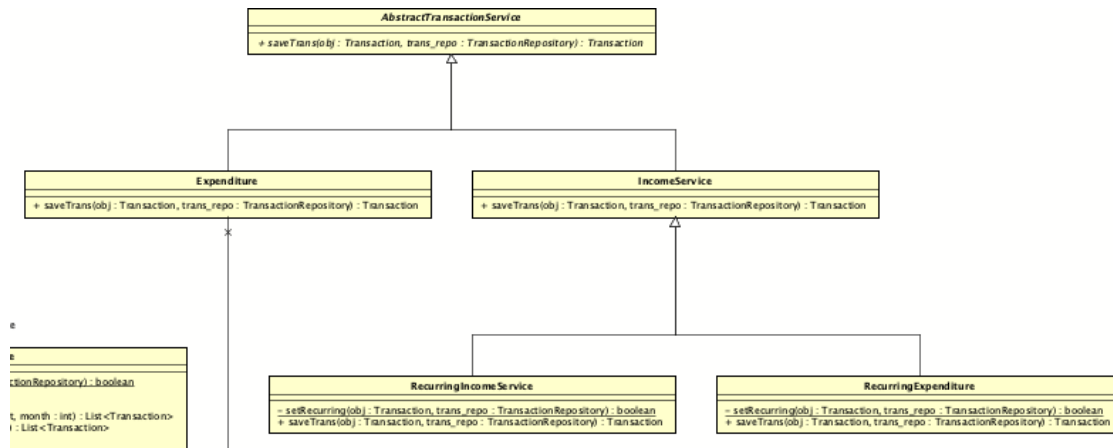


Figure 3 Example use of Abstract Pattern in the Application

Testing

Two types of testing were performed to ensure that the software worked as expected and that the new feature did not interfere with existing functionalities.

- **API Testing** – postman client was used to test each endpoint after adding a new feature. Postman collection is included in the attached achieve file.
- **Manual Functional Testing** – while leveraging deployed remote application and end-to-end manually regression test was conducted before submission.

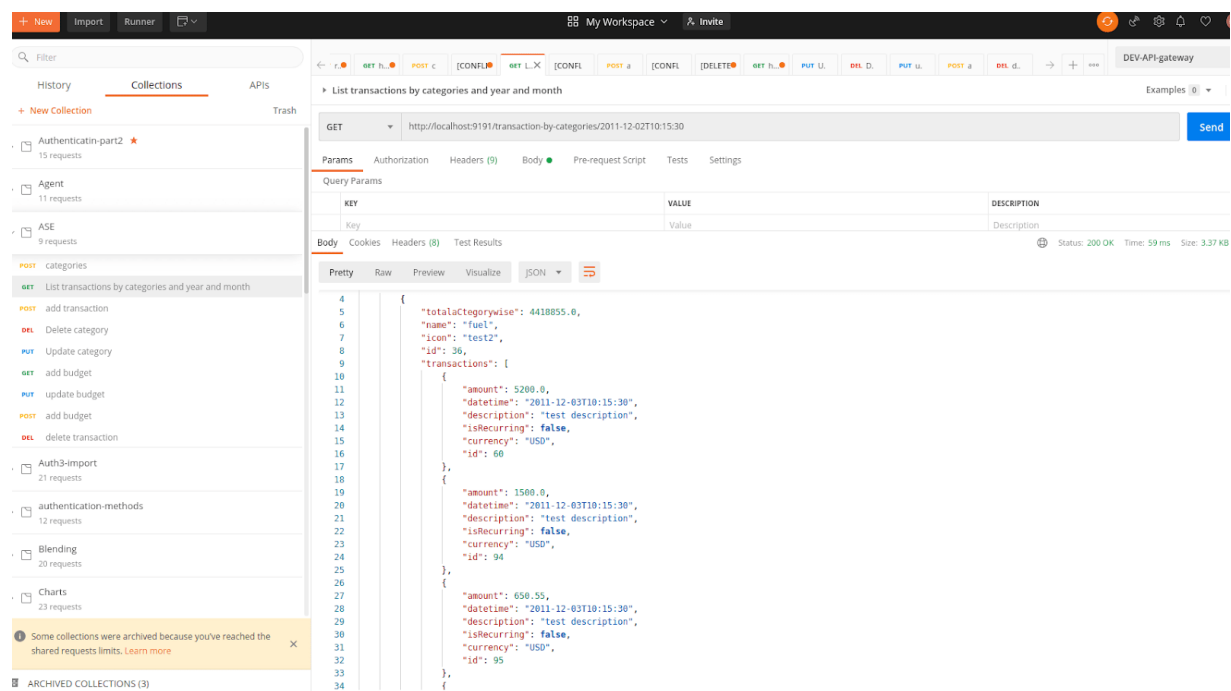


Figure 4 Screenshot of the API tests

Contributions

The responsibilities were divided among the team members as follow.

- **Project Manager –**
 - Murshid oversaw the project's leadership and management for the first four meetings.
 - Azeem oversaw the rest of the meetings and the team's leadership.
- **Designer –**
 - **Use case, Class Diagram and ERD:** Murshid and Azeem
 - **Design Pattern and Programming Logics:** Murshid and Azeem
- **Developer –**
 - **Frontend Development:** Murshid
 - **Backend Development:** Azeem
 - **CI/CD:** Murshid