

Senior Project, 2017, Spring

GoDutch v1.0

Student: Zhenglin Pan, Florida International University

Mentor: Dr. Deng Pan, Mohsen Taheri, Florida International University

Instructor: Masoud Sadjadi, Mohsen Taheri, Florida International University

Problem

- Calculating group trip cost per person is a boring and time consuming process.
- Different individual purchase and consumption situations make fair cost distribution computation fallible.
- Manual calculation can result in great amount of unnecessary transactions.

Solution

- To solve the problems, we need to automate the cost distribution and in-group transaction calculation process.
- A web application is needed to collect participants' purchase and consumption records.
- Minimum-Cost-Max-Flow algorithm is a good fit in this case to analyze and compute cost distribution and transaction.

Requirements

- UI must be clean and easy to use.
- Users must be able to edit the same trip simultaneously and see changes in real time.
- All calculation must happen on the server side.
- Algorithm must produce the correct result in 0.1 second for any group within 1000 participants.

System Design

Two architectural patterns are used (Figure 1).

- Model-View-ViewModel (MVVM)
 - View and ViewModel are connected through data binding using Angular.js.
 - ViewModel propagates changes to Model through routing and saving services implemented in Javascript on back end.
- Client/Server (C/S)
 - Users send requests from Client to Server to utilize services such as save/load changes and calculate transaction.
 - Client and Server communicate through socket messages handled by Socket.io.

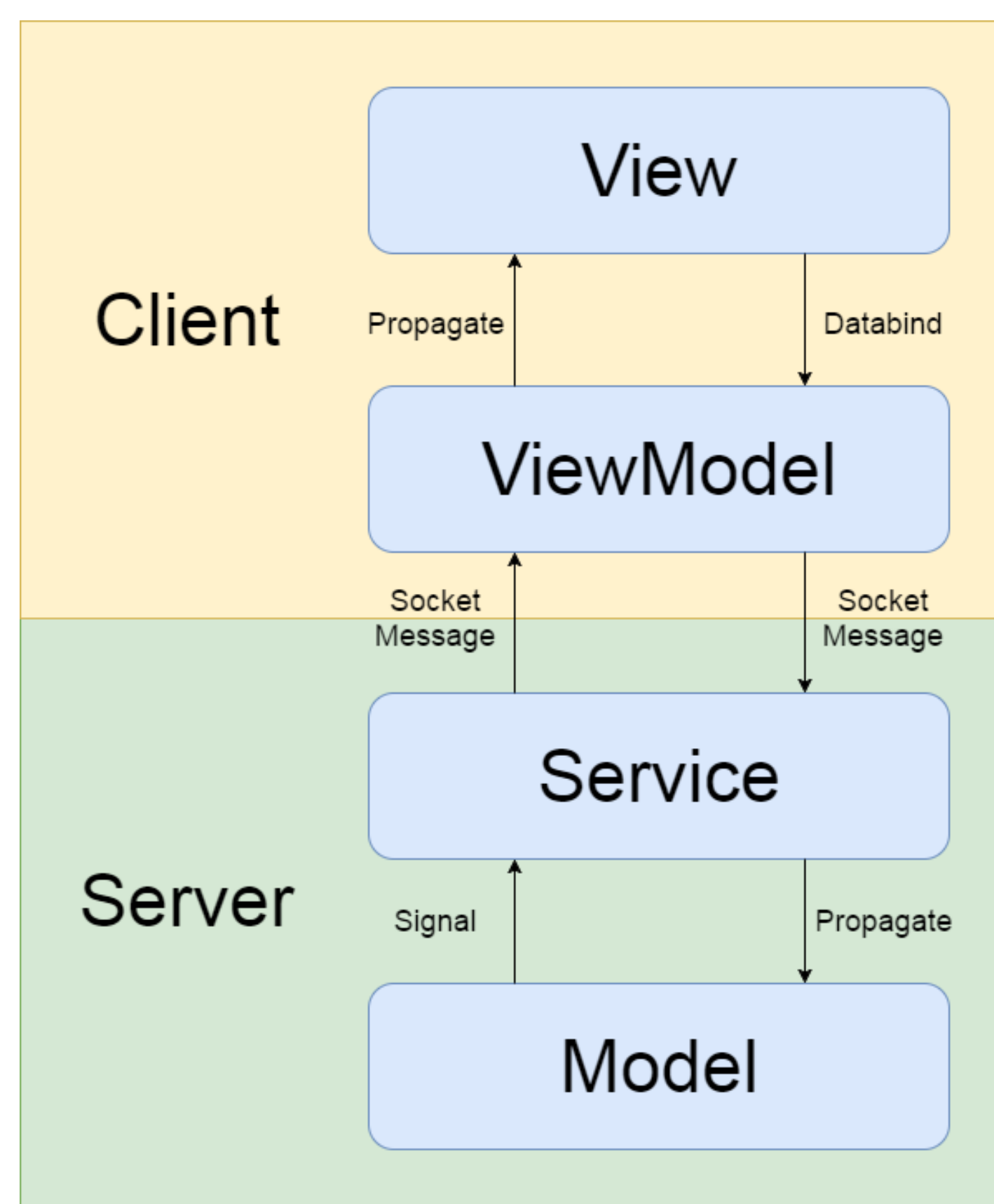


Figure 1. GoDutch leverages both MVVM and C/S architectural patterns.

Object Design

- Front Controller
 - All front requests are sent through Socket and handled by Route (Figure 2).

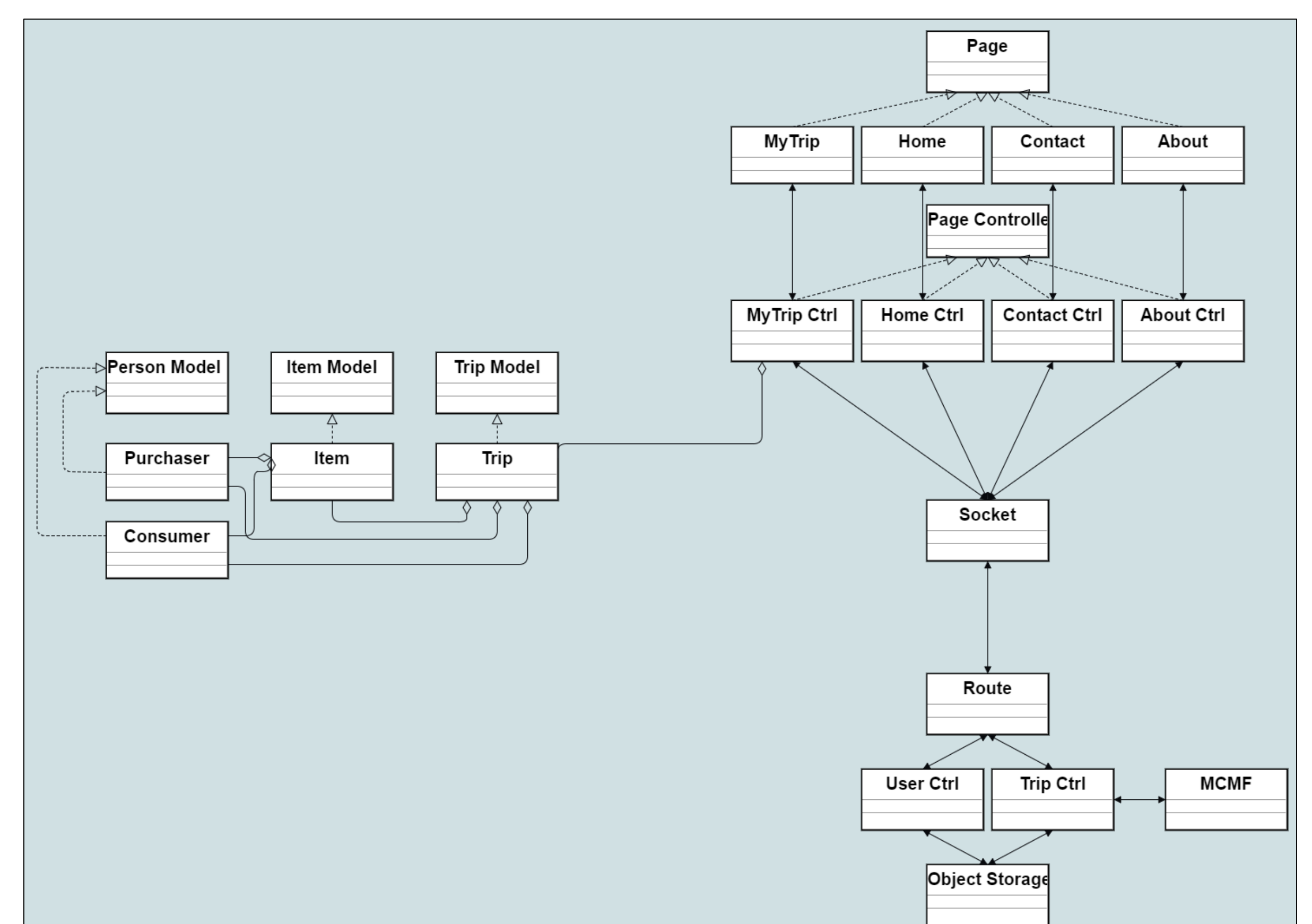


Figure 2. GoDutch uses front controller design patterns.

Implementation

- GoDutch follows top-to-bottom and left-to-right reading habit with soft color combinations (Figure 3).
 - The purchasers and consumers of selected item are highlighted.
 - Transaction list clearly displays payment records and in-group transaction result.
- Transaction result is generated using Min-Cost-Max-Flow algorithm (Figure 4).
 - Purchasers are connected from the Source and all users are connected to the Sink.
 - All users are fully connected to each other.

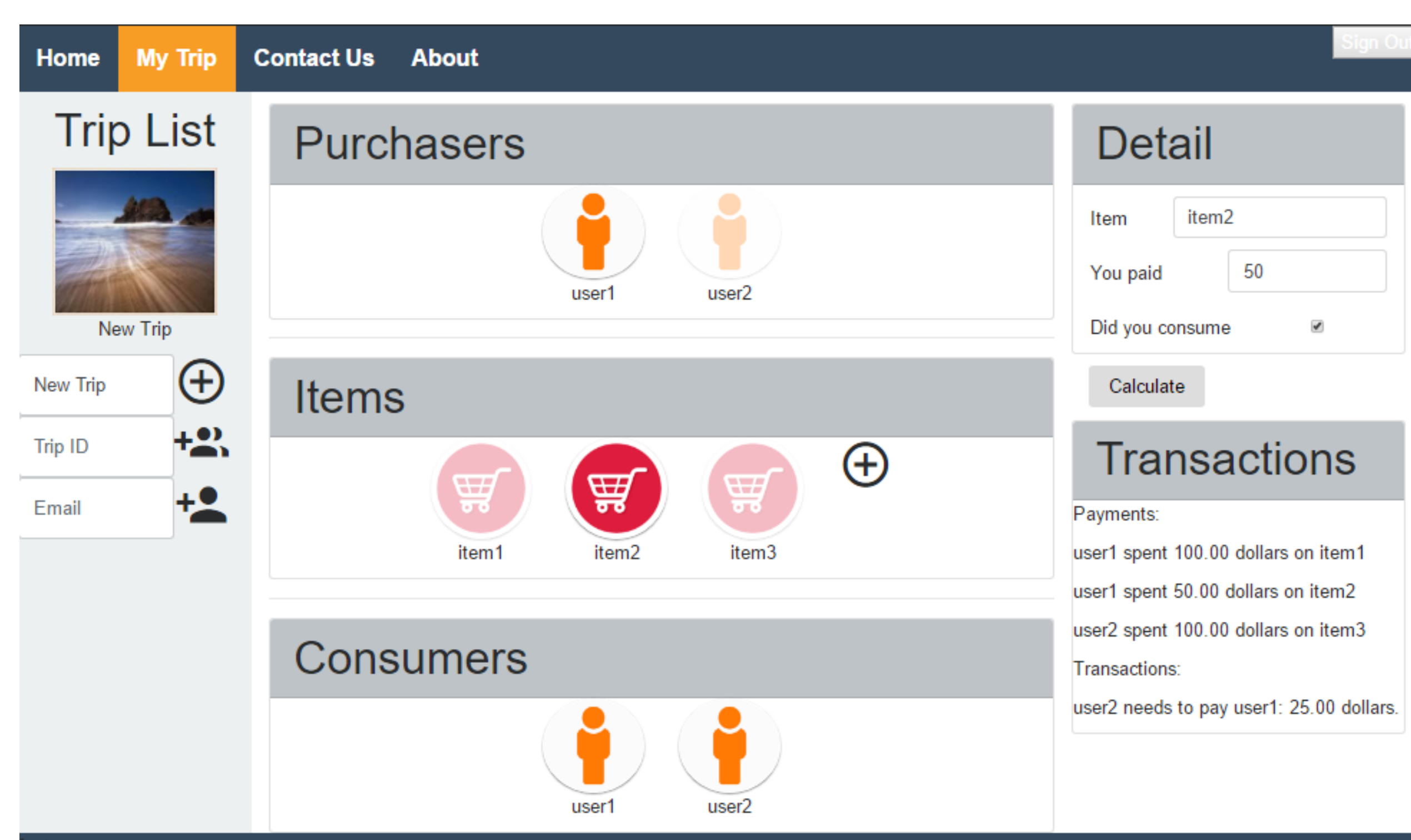


Figure 3. GoDutch User Interface.

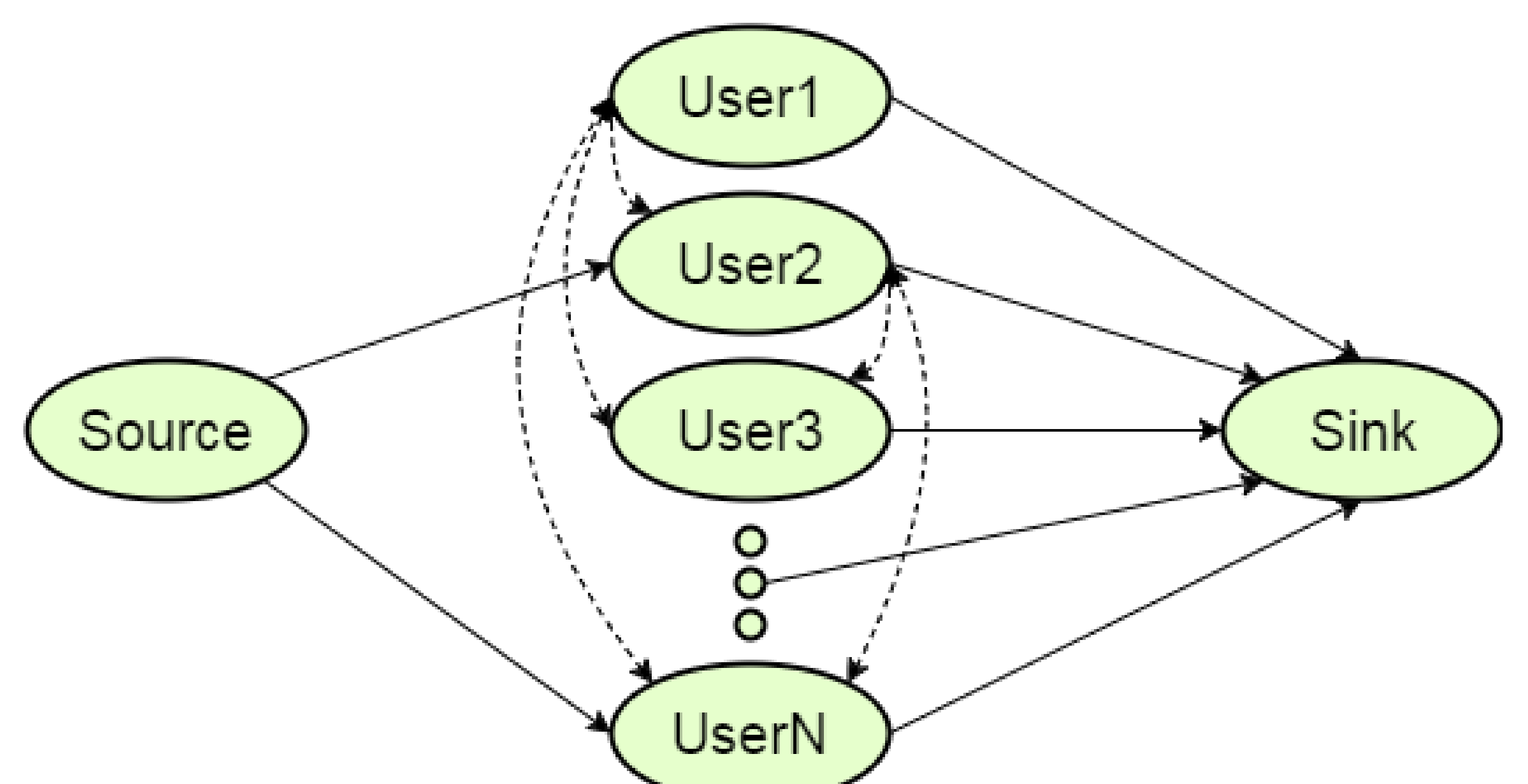


Figure 4. Min-Cost-Max-Flow Graph Model.

Acknowledgement

The material presented in this poster is based upon the work supported by Zhenglin Pan and Jonathan Beltran. I am thankful to the help that I received from my product owner Dr. Deng Pan and instructor Mohsen Taheri.