

Control Satisfaction Problem

A constraint satisfaction problem is defined by a collection of variables (x_1, x_2, \dots, x_n), a domain D_i with all possible values for each variable (x_i), and a set of assumptions about the relationships between the variables' values. If a problem is a control satisfaction problem then that problem needs to have three main properties.

- Variables: may store a variety of different values but it should only ever store one value at a time.
- Domain: values that variables can hold
- Constraints: conditions that must satisfy while assign value for variables from the domain

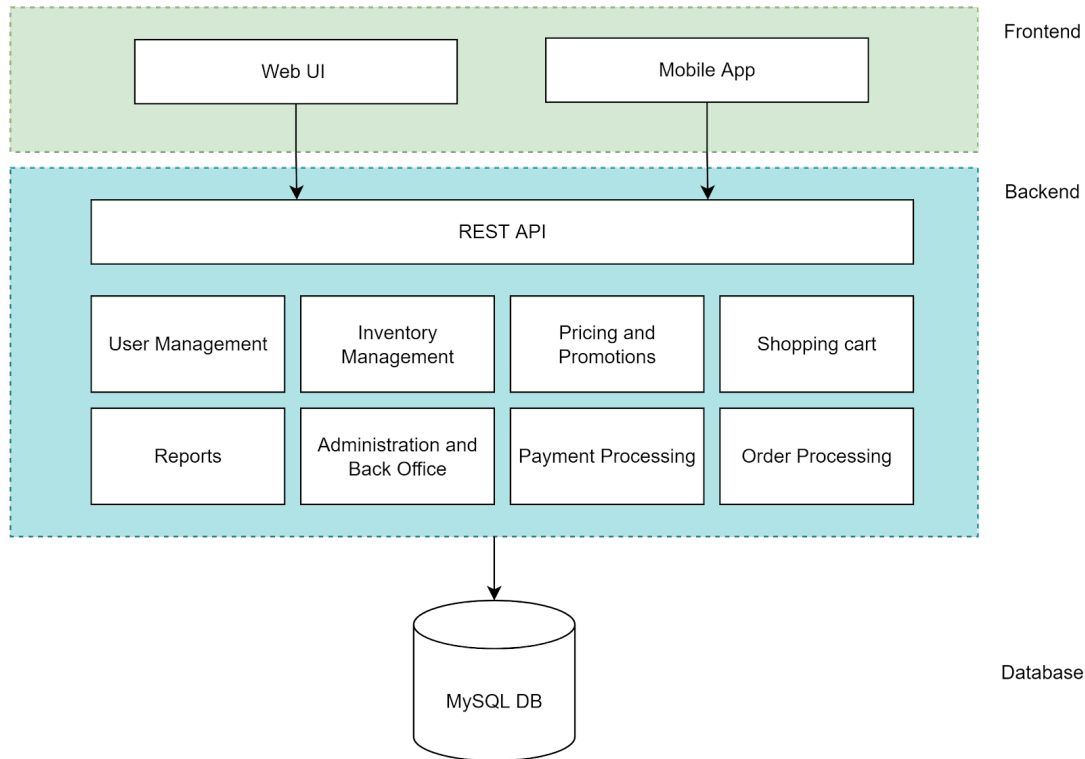
Timetabling Assignment problem also has above mentioned three properties.

- Variables – subjects
- Domain – possible time ('M1' , 'M2' , ...) and room ('R1' , 'R2' , ...) pairs for a subject
- Constraints
 - A given subjects can be assigned only to one of the possible time slots given for that subject.
 - Two compulsory subjects cannot be in the same time slot (optional subjects may).
 - Two subjects cannot be assigned to the same room if they are assigned to the same time slot.

So this problem also a Constraint Satisfaction Problem. Here we can define the initial state as empty list (initially any class has not assigned to a time slot), successor function as backtrack search function and the goal test will be the final answer which contains time slots (got from each subject domain) for every subject and each time slot satisfies all constraints.

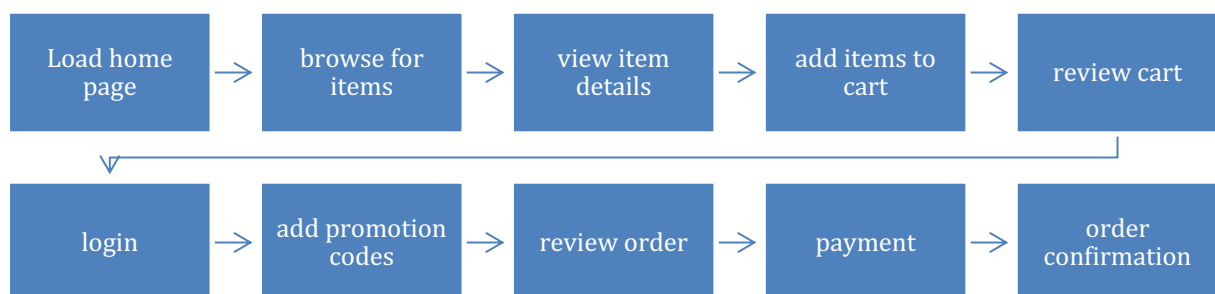
Current System State -

Customer has an existing e-commerce application with website and mobile apps. Following is a high-level component architecture of the existing system commerce platform which he can list and sell various products.

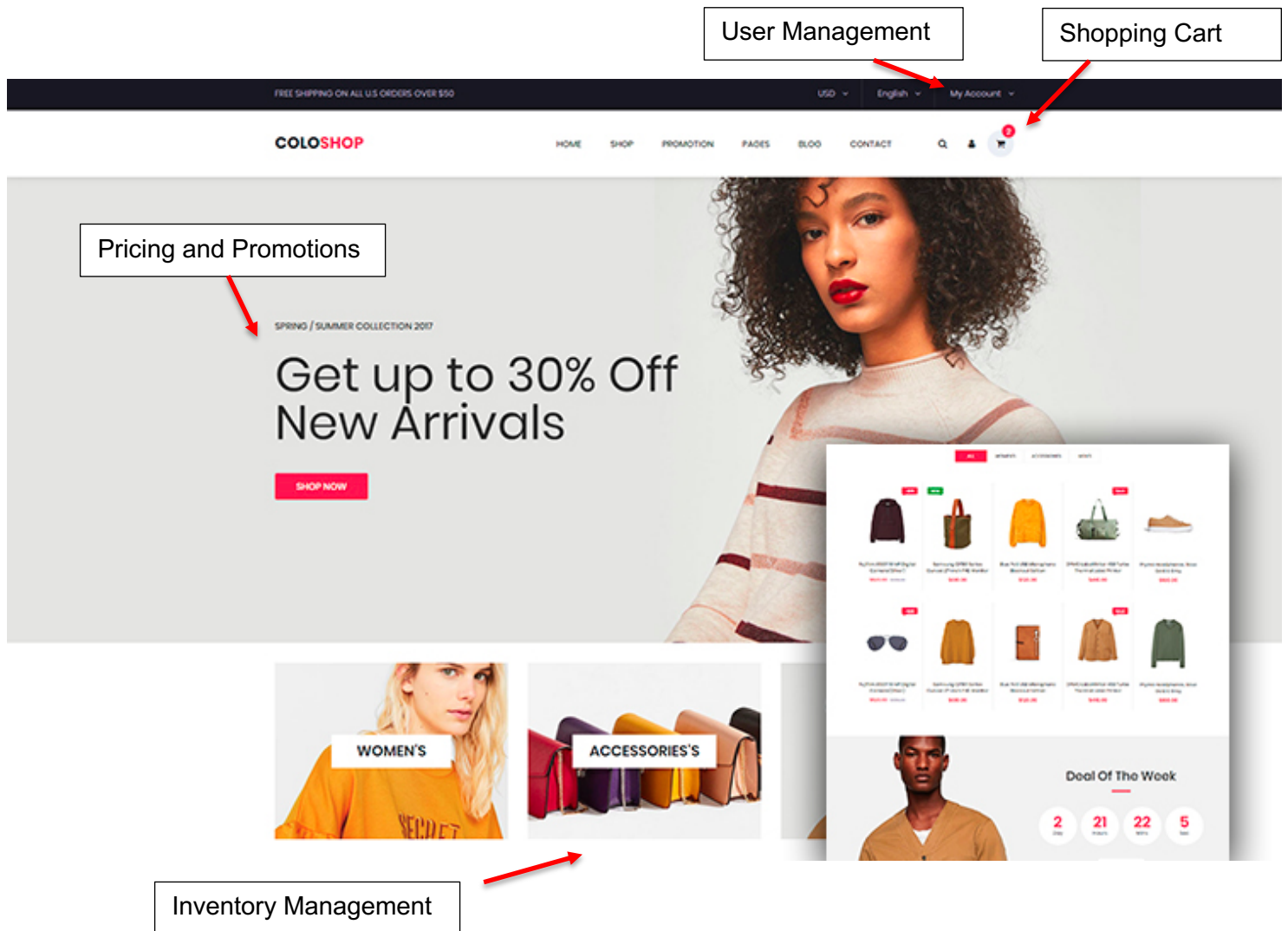


- Backend is written as a single deployable component (monolith)
- Currently system must be run on single server (no state management)

Simple user journey in the system will be as follows



Following is a sample view from web page outlining functionality mapping between presentation and different components of the backend



Change Request -

Client has requested to change the system to meeting following requirements

- Backend needs to be changed into a microservices
- System should be able to deploy in multiple servers to handle increasing load from growing business
- System needs to be integrated with external supplier systems to be able to connect to their system and sell external inventories

Ask –

- Re- Design the system to meet first two requirements and keep provision to implement third requirement as a future enhancement
- You are required to done the architecture changes while maximizing reuse of existing components