



institute for  
**SOFTWARE  
RESEARCH**

## Engineering Data Intensive Scalable Systems

**17-648**

Project 3

### Objectives:

This project is designed to give you hands on experience with various aspects of data intensive distributed systems (the scalable part comes later). Much of the complexity of both dealing with data and the distribution is often abstracted away from the application developer. The issue is that as systems grow in scale these abstraction mechanisms start to break down. As the systems scale the burden on the application developer increases.

The individual projects are designed to force you to deal with these complexities by removing the abstractions prior to building a system at scale. Specifically this assignment is designed to:

- Gain hands on experience with issues related to concurrency
- Reason about what is required in order to ensure correct operations given multiple simultaneous users
- Think about things like data integrity
- Worry about consistency
- Prepare the system to support scalability

### Background:

It's now clear that your business is successful. In fact you're so successful that you've started to stock your own inventory as well as sell bikes from external suppliers. You accomplish this by buying two of your suppliers. The systems are now completely under your control but continues to live on site at the warehouses (locations that are physically separate from your offices).

In order to manage the increased volume you are now allowing customers to browse inventory, order bikes, and look at their history directly via a web client.

The primary objective for your company is to be able to sell as many bikes as possible. When making design decisions consider the impact of the decision on the ability of the system to facilitate the sale of bikes.

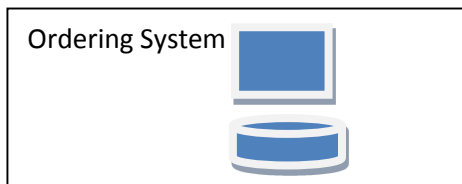
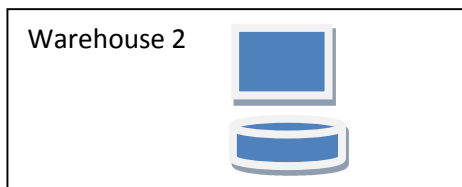
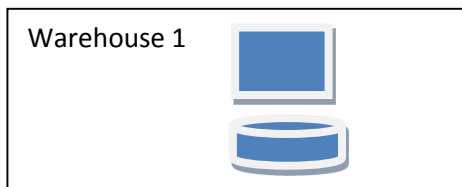
## Project:

The project is to add a web interface to the ordering system. The previous requirements for the system have been updated to support direct access by the customer. The system should work correctly with multiple users. This means that you are going to have to figure out a way to manage multiple users accessing the system at the same time and ensure the data is correct and consistent.

The functional requirements for the system are:

- Create an account that includes:
  - A first name
  - Last name
  - Address
  - a username and password
- Browse inventory
  - Order inventory by
    - Price
    - Bike name
- Purchase bikes (must be logged in). **The customer can buy more than one bike (potentially different models).** For each order you should store (the customer information can be gotten from the account):
  - Customer first and last name
  - Shipping address
    - Address
    - City
    - State
    - Zipcode
  - Items purchased
    - Item number(s)
    - Item name(s)
    - Price(s)
    - Total price
    - Quantity
- Search order history (must be logged in)
- Send notification that the bike has shipped
  - The warehouse should indicate when the bike has shipped (this can be some random timeout after the order is placed)
  - The system should send a notice to the user (we can assume that this would be an email but in this case it can just be a pop up notification)

You will continue to have access to the data sets for the two warehouses. The warehouses have separated the inventory data (the current bikes in stock) from the catalog data (the details of the bikes that the supplier typically stocks). You should make sure that the inventory across the system is as consistent as possible.



You are constrained to using Java SE 7 or later and Apache Tomcat 7 or later. You can use javascript, JSP, or Servlets for the processing of data on the web browser. Chrome will be the target browser.

### **Deliverables:**

You should deliver your source code and execution instructions. The code and instructions should be zipped together and submitted via blackboard by the due date. Late assignments will be penalized one letter grade per day late.

The instructions should describe exactly what we need to do in order to compile and execute your software. If we are not able to execute your software based on the instructions provided points will be deducted. You will not be given an opportunity to fix issues (so be sure to test your software and

instructions on a new machine to make sure no assumptions are made on the configuration of the environment).

### **Evaluation:**

You have a wide range of latitude with respect to how you implement this system. You will be evaluated along several dimensions:

- **Quality of your implementation:** It's expected that you adhere to acceptable coding standards. The more you use generally acceptable practices as opposed to just "hacking" the solution together the better you will do along this dimension. It's also expected that the system works. A fully working system is better than a partially working system, a partially working system is better than a system that builds but doesn't execute, and so forth.
- **Understanding:** You'll be evaluated based on your level of understanding with respect to what you've implemented and the impacts of these solutions. These will be discussed orally after you submit your assignment.