**1. Design and Implementation**

1.1 The REST API Specification

* Basic CRUD application. The web application focus is data management - Customers and Orders. The mobile application focus is operations – Sign pickup, delivery, and change scheduling.
* How to deal with Schedule – Placing an order will create an order document, but also needs to schedule the sign delivery, but Mongo doesn’t support transactions.
* How to deal with Order – Customer can contain OrderId and Mongoose will fetch the orders when the customer is requested, but it does so by executing a separate call to Mongo for each customer order. Order could be a sub-document of Customer, but would I have the ability to query all orders across all customers?

1.2 Database Schemas, Design and Structure

* Customer
  + Address { line1, line2, city, state, zip }
  + Phone { phone, type }
  + Contact { name, email, Address, [ Phone ] }
  + Customer { name, Address, [ Phone ], [ Contact ], [Order] }
* Order
  + LineItem { itemName, type, cost, geolocation }
  + Order { CustomerId, [ LineItem ], deliveryDate, rentalDuration }
* Inventory
  + Inventory { ProductId, geolocation, quantity }
* Product
  + Product { name, rentalCost }
* Schedule
  + Schedule { CustomerId, OrderId, LineItemId, date, action, geolocation }

1.3 Communication

* All JSON. It will support the standard REST API verbs – GET, PUT, POST, DELETE, and additionally, PATCH.

**2. Conclusions**

* I will probably not have time to complete the Web/Mobile UI with back-end integration. So for the purposes of this class, working via Postman will suffice.

**3. References**