

2102: WebOS / Web / Web browser

1. Case 11. Not notes 12. Notes 13. Notes 14. Notes 15. Notes

Project 0 requirements

customer order management app - 1.0 version

Project 0: store application

Feb 3 2021 10:00pm MDT / Web browser

functionality

- place orders to store locations for customers
- add a new customer
- search customers by name
- display details of an order
- display all order history of a store location
- display all order history of a customer
- input validation
- exception handling
- passwords: salt, no plain, customer, order history, etc. (hardcoded in code)
- optional: order history can be sorted by status, date, request, most requested
- optional: get suggested order for a customer based on his order history
- optional: display store statistics based on order history
- optional: asynchronous network & use (VCL)
- optional: logging of exceptions, if SQL commands, and other events to a log (JSON)
- optional: descriptive data from data (JSON optional)
- optional: multiple data to add (JSON optional)

design

- use IT Core (database first approach) (JSON)
- use an Azure SQL DB in cloud (normal form JSON)
- include a .NET script that can create the database schema and initial data (JSON)
- don't use public fields
- define and use a first class interface
- core / domain / business logic
- class library
- create an all business logic
- create domain classes (customer, order, store, product, etc.)
- document with Swagger? (API comments on all public types and members (optional))
- (recommended) have no dependencies on UI data access, or any third-party dependencies

order

- has a store location
- has a customer
- has an order type (when the order was placed)
- can contain multiple items of product in the same order
- rejects orders with unreasonably high product quantities
- optional: some additional business rules, like special deals

customer

- has an inventory
- inventory decreases when orders are accepted
- rejects orders that cannot be fulfilled with existing inventory
- (optional) for at least one product, more than one inventory item decreases when ordering that product

user interface

- interactive console application
- has only display and input related code
- low priority component, will be replaced when we move to project 1

data access (JSON)

- class library
- separate project for data access code using IT Core
- contains data access logic, but no business logic
- use repository pattern for separation of concerns

test

- at least 70% test coverage
- focus on unit testing business logic, testing the console app is very low priority

(MVC) Business Logic Model with

Customer Expressions as Business (Tree) Data

Order Screen First Name Last Name Account ID < PK Unique Address Secondary Info [Zipcode] < Unique PK as Default Store Location

Order Data

Order ID < PK Unique

Store location as Zipcode

Account ID

Date Time as Unix Timestamp

Product List code as File Content

Create

Product List code

Order with Product Transactions

General Description on classes and its members

Some other Add -> Customer

Search by Field Name < PK Unique

State City

Speech pattern for Feb 19 - 5th Ed

Differences in Specifications - Application as SQL with JSON

OP Code Interface Construction, Op Code JSON with JSON