CS6310 – Software Architecture & Design
Assignment #1 [100 points]: Grocery Express Project – Analysis & Design (v2)
Spring Term 2023 – Instructor: Mark Moss

Submission

- This assignment must be completed as an individual, not as part of a group.
- You must submit:
  (1) A UML Class Model Diagram named **class_diagram.pdf**;
  (2) Answers to the provided Design Questions as **question_answers.pdf**.

- You may divide your UML (v2.0) Class Model Diagram into multiple diagrams/files if needed. For example, if the initial diagram is so large that dividing it into multiples makes it significantly easier to read, then you may submit multiple files with reasonable name extensions such as **class_model_1.pdf**, **class_model_2.pdf**, etc. You must provide enough context for the multiple diagrams to allow us to connect them together to understand the singular/completed diagram.
- Formats other than PDF must be pre-approved an Instructor or TA.

- You must notify the Instructors and TAs via a private post on EdDiscussion BEFORE the Due Date if you are encountering difficulty submitting your project. Send the message to "Instructors" when posting your message in EdDiscussion, as opposed to addressing a message to just one individual, to ensure that it is acknowledged as quickly as possible.

- You will not be penalized for situations where Canvas is encountering significant technical problems. However, you must alert us before the Due Date – not well after the fact. You are responsible for submitting your answers on time in all other cases.

- Please consider that uploading files to Canvas might occasionally take a long time, even in the case of seemingly "relatively small" submissions. Please plan accordingly, as submissions outside of the Canvas Availability Date will most likely not be accepted. You are permitted to make unlimited submissions, and we recommend that you save, upload and submit often.

Problem Scenario

You are being asked to design and develop a system to monitor deliveries of grocery items to customers. This system will support a "third party" grocery service. Customers will place orders with various stores using the service. The service will coordinate with grocery stores to find the items (at variable – but hopefully the lowest – prices) and arrange for a drone to deliver the items to the customer. On delivery, the store will be paid electronically by the customer.

Your task in this assignment is to review the problem description below, and then provide design artifacts – namely, a UML class model – that includes enough detail to support the requirements below. You will also provide answers to questions to elaborate on how your design addresses the problem requirements. The answers to your questions should be consistent with your design.

Project Timeline

This is the first phase of the course project. Your goal for this phase of the project is to review the provided problem description, and to identify the entities, attributes, operations, methods and relationships that are key to the successful design and development of the system. You'll review the system requirements and specifications and provide design artifacts (e.g., UML structural and/or dynamic diagrams) and other responses that represent your proposed design and how it addresses the problem's challenges. This is similar to the Object-Oriented Analysis discussed in Udacity videos P2L1 – P2L3 and demonstrated in P2L5.

In the second phase of the course project, you'll be asked to provide peer reviews for some of your fellow classmates. You'll be granted access to their submissions and design artifacts from the first phase of the project, and then you'll provide feedback on their designs. Reviewing other designs – especially those of your peers – can help you make you more aware of alternative design approaches. The actual grade for each student's "first phase" submission will be determined solely by a TA's evaluation of their design – not your feedback – so please be candid and professional when providing your review. Your grade for this second phase of the project will be based on the quality, correctness, and thoroughness of the feedback that you provide to your peers.

In the third phase of the course project, you'll be asked to implement these requirements. It's often the case that implementing a system can help you identify aspects of the problem description that are unclear, ambiguous or possibly even contradictory. It can assist you in identifying aspects of your architecture and design that are faulty and/or incomplete. You will develop, build and test a "lightweight version" of the core system to help you further think through (and possibly improve) the quality of your design. We will evaluate the correctness of your system from a source code quality and design standpoint, and by how well your system actually passes a set of test cases.

The first three phases of the course project are all performed individually. In the final phases of the course project, you'll work in teams to achieve two major goals:
(A) Select three distinct improvements to the system, each of which might require some significant change to the current architecture and/or design; and,
(B) Implement your changes, and then provide design artifacts that document your design changes.

You'll work together as a group of three, four or five designers to determine a single unified design using some of the techniques mentioned in Udacity videos P2L4, P3L5, and others. Then, you'll work together to weigh the various architectural tradeoffs and build a single, core system that represents (and improves upon) your concepts from the earlier phases. You will also provide a video demonstration of how your system addresses the requirements, along with a description of how your final "as-is" system and design artifacts compare to the "to-be" artifacts you developed earlier.

Finally, you might also have to account for some relatively minor changes in the requirements from the earlier phases. Changes occur often in many projects, and this is an opportunity to consider strength of your original designs based (at least in part) on the amount of effort needed to update your most recent designs to accommodate the changed requirements.

Deliverables

For this assignment, you must analyze the Grocery Express Project Description that has been provided and then submit the following deliverables that accurately reflect the system. You must submit the following items:

(1) **Design Class Diagram [70 points] –** You must provide a Design Class Diagram as a file named **class_diagram.pdf** that includes:
   - classes (a minimum of 3 classes, a maximum of 10 classes)
   - attributes with basic types
   - operations (methods, as the implementations of the operations, will be required later during the development process)
   - relationships with proper cardinalities (e.g., [0..*]) that accurately reflect the problem space
   - visibility notation (e.g., public, private, etc.)

   To make your class diagram more readable, you may omit very basic "setters and getters" [e.g., set_() and get_() based operations and methods], especially those of the one-line variety that only provide simple and direct read or write access for an object attribute. Further details about what Is required in your Class Diagram can be found in the UML Guidelines in Canvas' Welcome Module.

(2) **Question Answers [30 points] –** The following is a set of questions designed to help ensure that your design meets the level of detail we are looking for in this assignment. Please submit your answers as a file named **question_answers.pdf**. Provide a relatively short (3-8 sentences) answer for each that explains which components of your design artifacts address the issues discussed in the question.

- How does your design address the need to ensure that drones can lift/carry a new order line?
- How does your design address the need to ensure that a customer can afford a new order line?
- How does your design address the need to display the incoming revenue for each store?

Vague answers that fail to reference your design artifacts appropriately (e.g., statements like "…it should be obvious that…") and avoid addressing the significant aspects of the question will lose points. By the same token, these answers should refer to and highlight the aspects of your class model that actually address the question, so you shouldn't need more than a few sentences to answer each question.

Problem Description, Requirements, Specifications, etc.

The Groceries Express Project focuses on a system that simulates the interactions between various customers and grocery stores, along with the other entities used by the third-party service to facilitate the placement and delivery of orders. The application that is developed must allow the users to:

- Create objects needed to represent the customers, stores, and other entities.
- Update details about the objects over time as specified.
- Support interactions between customers, stores, and other entities (e.g., a customer adding another line item to an existing order from a certain store); and,
- Display reports about the current state of the system (e.g., order price totals for each customer.)

The next few paragraphs will describe the key concepts that must be represented in your Design Class Diagram for this phase of the assignment. The concepts might be expanded or otherwise modified in later phases of the assignment as the clients, stakeholders, etc. clarify their requirements.

We must support various types of users, including customers and employees of the grocery stores (i.e., drone pilots). All users must be either independent customers or employees of a store - we will not keep track of any other types of users. We must maintain the first name, last name, and phone number for each user. The first and last name will be represented as simply strings, and the phone number will be represented with a ten-digit "xxx-xxx-xxxx" format. We will use the phone number to contact users only in cases where there are unexpected issues with one or more orders.

Each store will have a distinct name. Each store will also keep track of the revenue it has earned from delivering orders successfully. Stores will be supported by various employees, but the only types of employees that we will be concerned with at this phase are the drone pilots.

We must track the unique tax identifier (e.g., Social Security Number for some people) for each employee for legal purposes. The tax-identifier will be stored using a "xxx-xx-xxxx" format. We will also keep track of the number of months (as a whole number) that the employee has been working for the store the company, along with the employee's current salary.

Drone pilots are employees hired by stores to control the drones as they carry groceries back and forth between the stores and the customer's homes. Each drone pilot must have a valid license to signify that they have received the proper training to operate the drone safely. Each license will have a unique ID within the system for tracking purposes. Flight skills tend to improve with experience, so we must also keep track of the number of successful deliveries for each pilot.

Stores can purchase many drones to deliver orders to customers in a timely manner. Each drone can only be controlled by one pilot at a time - having multiple pilots for a single drone would eventually lead to conflicts and crashes. Also, it's unsafe for a pilot to control more than one drone at a time. Each drone has been purchased by and serves a single store, and is used to deliver orders for that store alone.

Drones need to be taken out of service for maintenance after making a certain number of trips, where a trip is equivalent to delivering one order. We must keep track of the number of trips that a drone has remaining before it needs maintenance. Also, each drone has a limited lifting capacity - measured in pounds – that must be tracked. The lifting capacity of a drone allows it to carry multiple orders up to a maximum weight.

Each store offers various items, where each item has a unique name (for that store) along with a weight measured in pounds. Customers who wish to purchase items can place an order. An order must be initiated by a specific customer and must also be immediately assigned to a specific drone for eventual delivery. Customers are allowed to place multiple orders concurrently.

Each order will consist of one or more lines, where each line represents a certain quantity (i.e., one or more) of an item being ordered at a given unit price. Each item can be listed at most once on a given order, but the quantity for that item can be one or more. The price and the quantity for an item can vary between different orders. A new line cannot be added to an existing order unless the drone assigned to deliver that order has enough lifting capacity to carry its all of its current pending orders. An order must be delivered in its entirety by a single drone - it cannot be split across multiple drones.

Customers have a rating as an integer from one (1) to five (5), where a higher number indicates that the customer has been more reliable in ordering items over time. We must also track the customer's credit as the number of dollars that they have to request orders. A customer's credit must always be greater than or equal to the total cost of all of the orders for which they are currently waiting. A new line cannot be added to an existing order unless the customer requesting the order has enough credit to cover all of the customer's current pending orders (i.e., orders waiting to be delivered).

We must be able to calculate and display the cost of an order as the total cost of each line, which is the cost of the item as listed on that order multiplied by the quantity purchased. We must also be able to calculate and display the total cost for all of the outstanding orders for each customer.

We must be able to calculate and display the weight of an order as the total of the weight of each line, which is the weight of the individual item multiplied by the quantity purchased. We must also be able to calculate and display the total weight (i.e., payload) for all of the orders being delivered by each drone.

Finally, we must be able to calculate and display the incoming revenue for each store as the total cost of all pending orders currently waiting to be delivered by drones on behalf of that store.

Evaluating/Grading Your Submissions

- You must label all diagrams with a header that includes your name, class name & semester and the UML version being used – for example:
  `Matthew Carter : CS6310 Spring 2023 Assignment 1`
- You must generate your diagrams using an automated tool (e.g. Argo UML, draw.io, Lucid Charts, Microsoft Visio, etc.) so that they are as clear & legible as possible. Even PowerPoint is allowed, though this is an excellent opportunity to use a tool that is more appropriately designed for UML as opposed to a general drawing tool like PowerPoint. The choice of tool(s) is yours; however, you should do a "sanity check" to make sure that your final diagrams are readable when exported to PDF; or, if necessary, some reasonable graphical format (PNG, JPG or GIF).
- By "automated tool", we don't mean a system or application that will generate the elements of the UML Design Class Diagram for you. For example, we don't mean a tool or system that will reverse engineer a codebase and propose the classes, objects and relationships for you. These are the skills that you should be developing as part of this assignment.
- We prefer that you submit your diagrams in dark, rich colors (particularly black). Diagrams submitted in certain colors, especially pastel or lighter shades, can be difficult to read.
- ~~You must designate which version of UML you will be using – either 1.4 (the latest ISO-accepted version) or 2.0 (the latest OMG-accepted version).~~ You must use UML version 2.0 (or later) as a

minimum. There were significant differences between versions 1.4 and 2.0, so moving to 2.0 will improve consistency during evaluations.
https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=oum-naming-differences-between-uml-14-later-uml-versions

Closing Comments & Suggestions

This is the information that has been provided by the customer so far. We (the OMSCS 6310 Team) will likely conduct an Office Hours where you will be permitted to ask us questions in order to clarify the client's intent, etc. We will answer some of the questions, but we will not necessarily answer all of them. One of your main tasks will be to ensure that your architectural documents and related artifacts remain consistent with the problem requirements – and with your system implementations – over time.

Quick Reminder on Collaboration

Please use EdDiscussion for your questions and/or comments and post publicly whenever it is appropriate. If your questions or comments contain information that specifically provides an answer for some part of the assignment, then please make your post private first, and we (the OMSCS 6310 Team) will review it and decide if it is suitable to be shared with the larger class. Best of luck on to you this assignment, and please contact us if you have questions or concerns.