CSCE 156: Project Overview

Fall 2021

1. Introduction

Over the course of this semester, you will incrementally build a substantial database-backed application in Java. In each phase of the project, you will focus on a particular component, which will have deliverables that you must hand in by a certain date. These deliverables may include a Java Archive (JAR) file, source code, non-trivial test cases, database schemas and a well-written technical design document.

Each phase builds upon prior phases and may also require updates and modifications to prior phases. It is important that you understand the entire scope of the project. You should read all the assignment and the project overview to get a better understanding of where the project will be going.

The iterative nature of this project means that it is vital that you do not fall behind. In each phase, it is important that you have a good, well-thought design to make subsequent phases easier to design, extend and implement. Poor designs, bad implementations, bugs, and broken code will mean subsequent phases of this project will suffer. Investing your time and resources upfront will minimize your technical debt and mitigate the need to update or refactor your design later on. Remember one of the Golden Rules of coding: only code that which you would not mind having to maintain.

2. Problem Statement

Local investor **Thomas Anderson** has just acquired a chain of regional Computer stores, **The Matrix**. Unfortunately, the previous owner was a dinosaur and didn't have a good technology infrastructure. All sales records were maintained by a collection of Excel sheets and ancient mysterious macros that produce sales reports. In an effort to update business practices and to compete in the market, Anderson wants to modernize The Matrix by designing a new database backed sales system. He has hired you to design and implement this system.

It will be your responsibility to design a new system from scratch that is Object-Oriented, written in Java, and supports The Matrix's business model by implementing their business rules and providing the functionality as stated below.

The Matrix stores selling and renting PCs, Laptops and systems as well as accessories, gift cards, subscriptions and related products. All sales items fall under three main categories: products, services, and subscriptions. All items are identified with a unique alphanumeric code and a name.

- **Products:** includes PCs, Laptops, computer equipment, apparel, etc. as well as gift cards. Each item has a specific base price. However, depending on the subcategory of item, this price may be modified:
    - All new items' cost is the base price
    - All used items' cost is 80% of the base price
    - Gift card prices are determined by how much a customer specifies for a particular sale.

There are no additional fees or other direct costs for equipment.

- **Services:** are services that are performed by trained Matrix staff and may include repair, clean up, in-home system setup, PC system builds, game training, etc. Each service is associated with a particular employee and are billed on a per-hour basis. When a purchase is made, a fixed number of hours is charged to the customer. Thus, the total cost of a service is the hourly rate multiplied by the number of hours. Another service provided by the Matrix stores is renting services, where the customer can rent a PC, Laptop, computer equipment, …etc. The cost of the renting service will be calculated based on the device rented and the number of days to rent.

- **Subscriptions:** include service subscriptions, memberships, licenses, or third-party services. Subscriptions have an annual fee. A purchased subscription has an effective begin and effective end date to determine billing. The total cost of a subscription is the number of Months included in the effective dates divided by 12 multiplied by the annual fee.

A sale is a collection of items purchased by a customer at a particular Matrix store. Each sale includes:

- A unique alphanumeric code identifying the sale
- The customer (and their info) that the sale has been made to
- The store (and its info) that the sale was made at
- A number of items made for that particular sale

Depending on the customer and items on the sale, various fees and taxes are also applied.

- All products have a 7.00% sales tax
- All services have a 5.75% service sales tax
- No tax is applied to Memberships

The Matrix also has a membership program that recognizes Silver and Gold customers.

- For Silver customers, there is a 3% discount on all purchases
- For Gold customers, there is a 7% discount on all purchases
- If a customer is not a member of the program, no discount is applied
- Employees receive a generous 10% discount

Notes:

- Taxes are applied to each sale item independently (since they may have different rates)
- Any discount is applied to the subtotal of all items after taxes.
- Every figure is rounded to the nearest cent.

3. Project Outline

Over the course of this semester, you will iteratively design an application to support this business model. Development has been broken down into 6 phases:

- **Phase I:** Data Representation & Electronic Data Interchange (EDI) - in the first phase you will design and implement the objects that will form a basis for the system and create parsers to read data from at flat files, generate instances of your objects and export them to an interchange format (XML and/or JSON).

- **Phase II:** Summary Report - In this phase you will further refine your objects and define relationships between them to generate a summary report that aggregates pieces of data together.

- **Phase III:** Database Design - This phase focuses on designing a relational database to model your objects and support your application.

- **Phase IV:** Database Connectivity - You will refactor your code to load your objects to your database rather than from flat files.

- **Phase V:** Database Persistence - You will implement and use an API to persist (save) data to your database.

- **Phase VI:** Sorted List ADT - In this phase you design and implement a sorted list ADT and integrate it into your application.

Appendix A: Project Setup Requirements

To ensure that your projects will run on the webgrader you must adhere to the following requirements and procedures. Failure to do so may make it impossible to grade your project and you will not receive credit.

A.1 Setup Requirements (Eclipse)

- You must include a readme.md (a markdown file) file at the root of your project with your name(s), cse login(s), NUIDs and preferred email. If you choose to work in pairs, both names/contact info should be included.
- All data files must be included in a directory named data at the root of your project.
- Any external JAR libraries should be placed in a directory named lib at the root of your project. Your project should look something like the following.

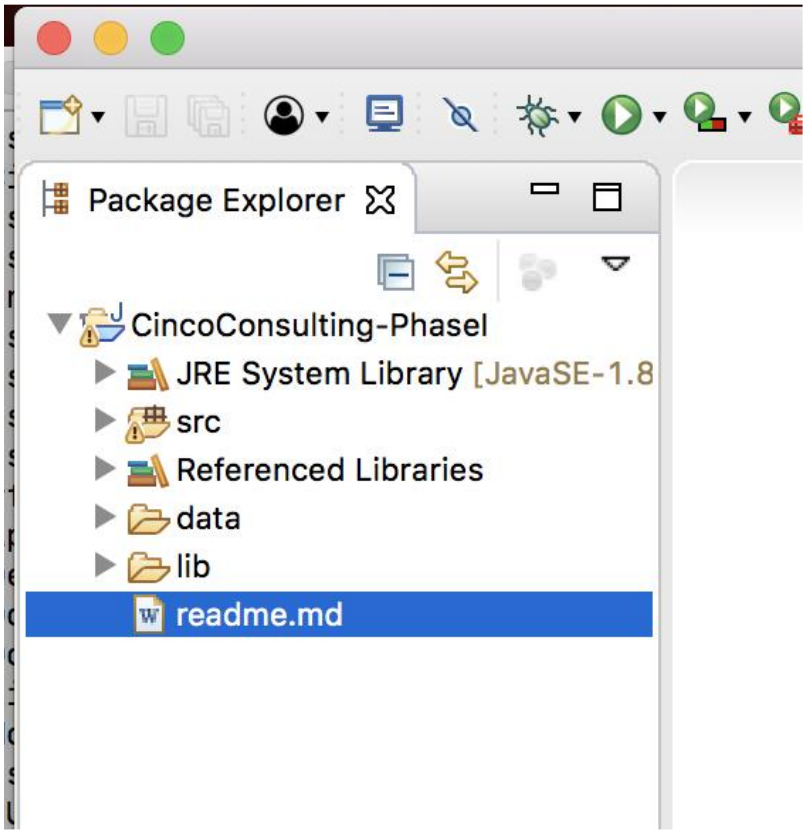


Figure 1: Your project setup should look something like this figure.

To add external JAR files to your project, do the following:

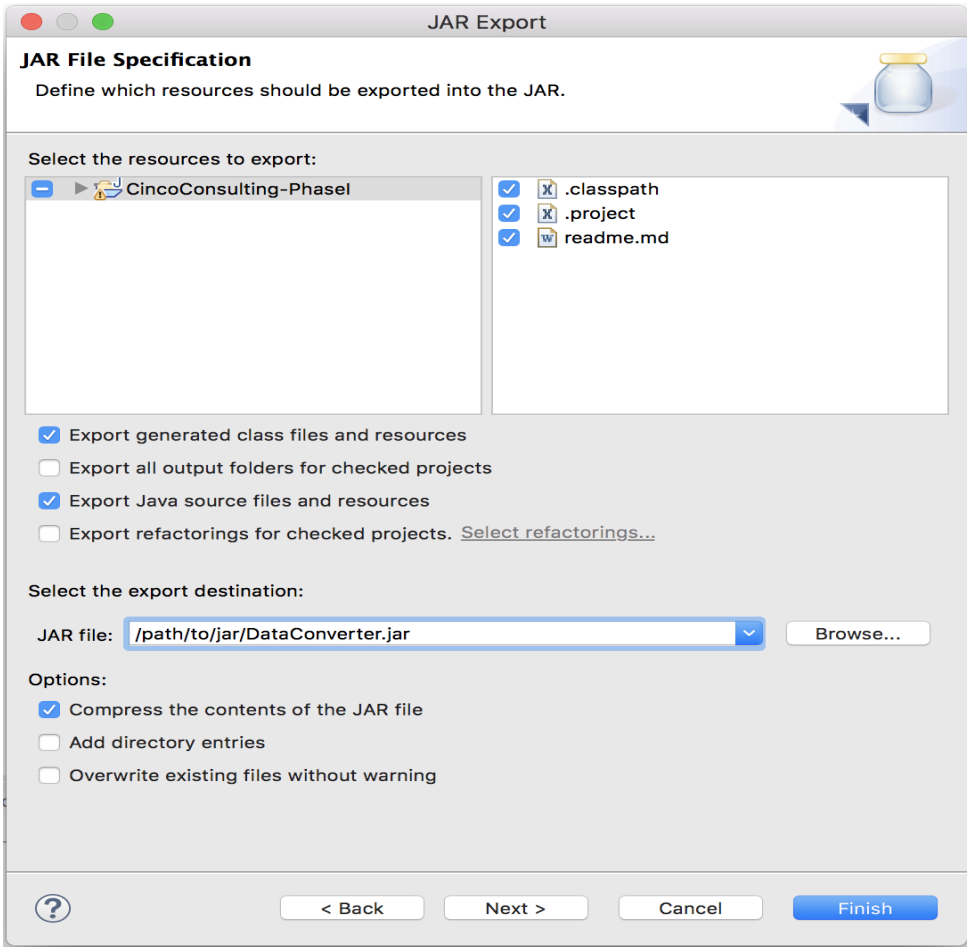
1. Drag and drop your JAR file to this folder, be sure to select “copy files”
2. Right click the new JAR file in your lib folder and select Build Path -> Add to Build Path

- Your main method must be in the package and class name specified by the assignment.

A.2 Exporting Your Project (Eclipse)

To export your project for submission to webgrader, do the following:

1. Run your program at least once, this creates a “Launch Configuration” in Eclipse
2. Right click your project and select Export...
3. Under the Java folder, select JAR file, click Next
4. Be sure to check Export Java source files and resources and select the location where you want the JAR file saved. It should look something like the following.



5. You can now click Finish and it should save the project as a JAR file which you will then turn in. The JAR file should contain all libraries, classes, source files and other data files necessary, but you should be **sure it works by running the webgrader and addressing any issues.**

Appendix B: Partner Policy & Procedure

You may choose to work alone or with a single partner (*i.e., pairs*) for each phase of the project. You may change partners between each phase if you choose. If you do choose to work in pairs, you must adhere to the following guidelines:

- All work must be the result of an equal collaborative effort by each partner. You may not simply partition the work between you

- Turn in only one copy of the design document with both of your names on it

- You must turn in only one electronic copy under the login of the partner whose last name comes first alphabetically

- You must follow any additional policies regarding late passes or other items as described in the syllabus

- You are highly encouraged to use some sort of revision control system such as Git. However, you must ensure that your team's codebase and artifacts are not publicly accessible. Failure to do so will be considered a violation of the department's academic integrity policy.

- In order to work in pairs, you must join a group together in Canvas under the “Project Partners” group set. **Do not create your own group set, use an existing one**.