

# CSCE 156H/RAIK 184H: Project Overview

Dr. Chris Bourke

Spring 2016

## 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 will include self-executing JAR (Java Archive) files, ZIP archive files of your source code, well-written 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 of the assignments to get a better understanding of where the project will be going.

The iterative nature of this project means that it is important that you do not fall behind. In each phase it is also important that you have a good, well-thought design to make subsequent phases easier to design 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

Cinco Computer Consultants (CCC) is a medium sized corporation that provides computer services to regional companies including equipment and services such as training, consultations and licenses. Though the services that CCC supplies are state-of-the art, its invoice system is not. CCC has hired you to design and implement a simple invoice system to replace the aging AS400 green-screen system that they currently use. It will be your responsibility to design a new system from scratch that is Object-Oriented, written in Java, and supports CCC's business model by implementing their business rules and providing the functionality

as stated below. CCC offers several products: equipment, consultations, and licenses. Each product consists of various pieces of data. All products have a unique alphanumeric product code and a name.

- **Equipment** includes various computer and electronic products that CCC sells to its clients as a reseller. All equipment has a price-per-unit cost. When a piece of equipment appears on an invoice, the total cost is the price-per-unit times the number of units on the invoice. There are no additional fees or other costs for equipment.
- **Consultations** are services that are offered to clients by CCC such as training, system evaluations, etc. Consultation services are associated with a consultant (a person) and are billed to the client on a per-hour basis. In addition to the hourly cost, there is a flat \$150 service fee for all consultations. Consultations have an hourly fee and when they appear on an invoice, a total number of billable hours. Thus, the total cost of a consultation is the hourly fee multiplied by the number of billable hours plus the \$150 service fee.
- **Licenses** are also products that include things like software, server hosting, or third-party services. Licenses also carry a service fee, but the amount is different for each license. In addition, each license also has an annual license fee. When a license is included in an invoice, it has effective dates: a beginning and end date to determine billing. The total cost of a license is the number of days included in the effective dates divided by 365 (we ignore leap years) multiplied by the annual service fee plus the flat service fee.

Invoices are individual sales that include:

- A unique alphanumeric code identifying the invoice
- The customer that the sale has been made to
- A salesperson that was responsible for the sale
- A number of products made for that particular order

Depending on the customer and product on an invoice, various fees and taxes are also applied.

- For government customers, a flat state and federal compliance fee is applied of \$125.00 per invoice. For corporate customers, there is no compliance fee.
- Equipment carries a 7% sales tax rate while services (licenses and consultations) carry a 4.25% sales tax rate for corporate customers. For licenses and consultations, tax is assessed only on the hourly fees, not the \$150 service fee (consultations) nor the flat service fee (licenses)
- Government customers have no sales tax on any products as they are tax-free institutions

### 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 5 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 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 and persist your objects to your database rather than from flat files.
- Phase V: Sorted List ADT – In this phase you design and implement a sorted list ADT and integrate it into your application.

## A Creating Runnable JARs and Archive Files

These instructions are for creating runnable JAR file in Eclipse (Indigo); instructions may differ for other versions or other IDEs.

### A.1 Creating a Runnable JAR file

1. Run your program at least once, this creates a “Launch Configuration” in Eclipse
2. Right click on your project, select “Export”
3. Under the Java folder, select “Runnable JAR file”, click Next
4. Select the “Launch Configuration” corresponding to your main method
5. Under “Export destination:” click Browse and select a director location and file name (ending in .jar) where you want the JAR file to be exported to
6. Under “Library handling:” be sure that “Package required libraries into generated JAR” is selected
7. Click Finish
8. The JAR file should now be in the directory you indicated

### A.2 Creating a ZIP archive of your source code

1. Right click your project, select “Export”
2. Under the “General” folder select “Archive File”
3. Click Next
4. Click Browse and select a directory/file name to export to
5. Under “Options” make sure that “Save in zip format” is selected
6. Click Finish; the ZIP file should now be in the directory/file you indicated

## **B Adding External JAR Libraries to an Eclipse Project**

These instructions are for creating runnable JAR file in Eclipse (Kepler); instructions may differ for other versions or other IDEs.

There are many ways to import external JAR libraries (examples: gson-1.7.1.jar, joda-time-2.0.jar). The following instructions will be most compatible with how we expect you to build your runnable JAR file.

1. Create a folder in your project: Right click project → New → Folder
2. Name your folder `lib` (short for library)
3. Drag and drop your JAR file to this folder, be sure to select “copy files”
4. Right click the new JAR file in your lib folder and select Build Path → Add to Build Path

## C Partner Policy

If you choose, you may work in groups of *up to three* people for each phase of the project (you can change partners with prior permission from the instructor). If you choose to work in groups, you must adhere to the following:

- All work must be the result of an equal collaborative effort by each member
- Turn in only one copy of the design document with all names on it
- You must turn in only one electronic copy under the first author's login
- 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 SVN or 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.
  - For more information on Github, see their bootcamp: <https://help.github.com/articles/set-up-git/>
  - For more information on setting up an SVN on CSE, see: <http://cse.unl.edu/systems-faqs/using-cvs-or-svn-version-control>