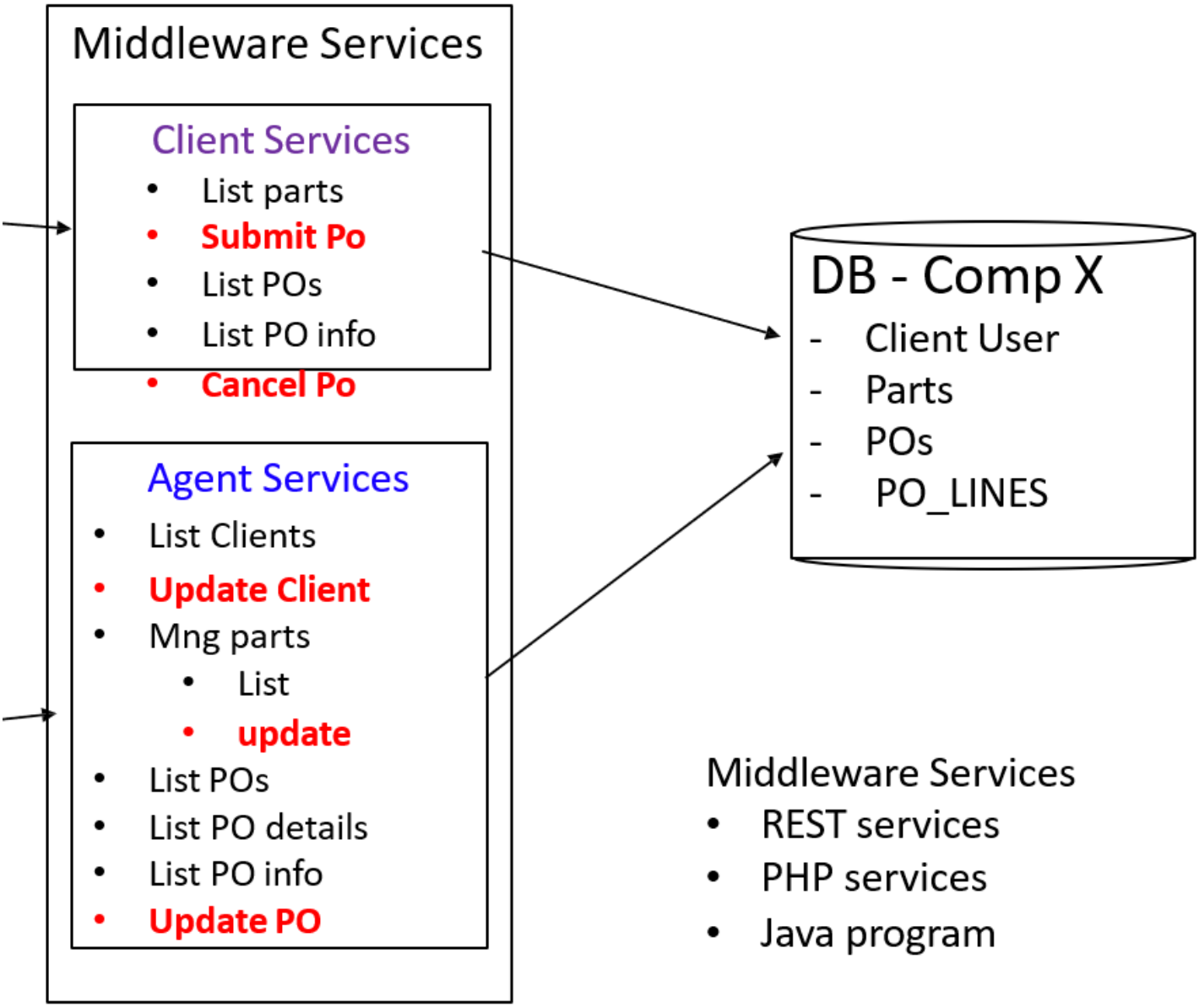
CSCI 4140 Advanced Database Systems

Assignment 3
REST (Mock) API with Procedural SQL

Context for Assignments and Project

Assignments and projects are structured so that they build upon each other:

- Assignment 1 is asking students to demonstrate that students can good understanding of DB design. And students can utilize at least one DB management tool to create ER/EER diagram, then convert diagram to the actual tables in a database.
    - You can use a simple database management to model the DB.
    - You need to generate DB modeling diagram by reverse engineering.
    - You need to generate SQL (DDL and DML) by forward engineering.
- Assignment 2 expands the functionality of the Assignment 1 and makes preparation for the web application implementation of database connectivity. Topics: REST API and JSON.
- Assignment 3 further expands the functionality of your software in Assignment 2 and it is requiring that you use web services (Update-delete-insert) for back-end processing. Besides, you need to utilize trigger or procedure to achieve the data manipulation cross different DB.
- Assignment 4 makes progress further to pick up the REST API constructed by A2 AND A3, THEN CONTRUCT UI BASED ON THE REQUIREMENTS.
    - Again, although you may develop your software as a simple application, it is recommended that you use web services (assistance will be provided)
- Assignment 5 integrate the implementations of A1-A4, and expands functionality requirements of implement 2-phase committing in distributed environment in which a transaction requires updates performed on independent DBs.



- Assignment 3 further expands the functionality of your software in Assignment 2 and it is requiring that you use web services (**Update-delete-insert**) for back-end processing. Besides, you need to utilize **trigger or procedure** to achieve the data manipulation cross different DB.

Learning Outcomes

- Design middleware(Update-Delete-Insert-Trigger-Procedure) to connect with DB of A1.
- **Know how to manipulate data with triggers and stored procedures to ensure the data integrity for the possible business events (you design and resolve them).**
- Know to how to utilize one of **REST services client tools to test services call**. (SOAPUI, POSTMAN, et al.)
- Know at least one of the languages, such as JavaScript(Node), PHP, Python, et al. to connect with DB.

Specifications

Data Requirements

The database schema is based on the outcomes of A1, plus, **students need to trigger the events through the REST services of A3.**

Functional Requirements (A3)

Assume that the application that you are developing is for a company that offers parts for sale . Your task is to develop an application program that will enable a company agent to manage parts and client information. More specifically, your task is to create a list of (REST) services , that enables a agent to:

- Update Part. (especially part price and quantity, you might consider the possibility of utilizing triggers and procedures.)
- Update Customer (especially balance, you might consider the possibility of utilizing triggers and procedures.)
- Update Purchase Order (especially order status, you might consider the possibility of utilizing triggers and procedures.)

And it enables the clients to:

- **Post new order (keep in mind: it could fire your defined triggers and procedures)**
- **Cancel the existing order (keep in mind: it could fire your defined triggers and procedures)**

Client Service API

i. Post New Order

Client's post new order event will trigger the defined the procedure to update other tables (Please consider it carefully).

ii. Cancel the Progressing PO

Client's cancellation of progressing order event will trigger the defined the procedure to update other tables (You might want to add additional considerations, for instance, free cancellation is not allow for a progressing PO before confirmation from seller; after that, there will be small charge as cancellation fee).

Company Agent Service API

iii. Update parts

Update information of parts, especially for adjusting the price and replenishing the parts.

iv. Update Customer

Update customer information. Agents can adjust the customer balance.

v. Update PO status

The agents are able to directly change the PO status. (keep in mind, changing the PO status to cancel might fire the trigger to refund to customer.

Additional Requirements

- MySQL DBMS should be used, either using the FCS DB server or cloud-based one.
- Naming Conventions
  - Your DB should be such that the names of your tables and of the columns must end with the last three digits of your group ID, such as 004 (if your group number is 4).
  - In your software, the names of your methods/functions/procedures should end with the last three digits of your group ID and the same applies about any parameter of a method/procedure/function.

Submission requirements

Your submission should have the usual packaging (front page, TOC (Table of

Contents), etc.) and contain the following information:

- Section describing your **improved** ER model developed with MySQL Workbench, including
  - ER model/diagram and a brief listing of your entities/relationships/subtypes.
  - SQL scripts to modify DB tables.
  - **Procedural SQL scripts, such as stored procedures and triggers.**

- Section describing your services call implementation
  - Describe how you implemented database connectivity.
  - List of implemented web services, and their endpoints for testing.
  - Describe how you utilize the web services client tools, such as POSTMAN or SOAPUI etc., to validate your web services.
  - The screenshots of the results of the database before and after firing the triggers and executing the stored procedures.

- Section that identifies under which situation/condition your program does not work (known bugs … if any)

- Your code and how to run it, where the code includes
  - Your SQL script files, including **Procedural SQL scripts, such as stored procedures and triggers.**
  - Your code for the program/ web-services
  - Your should submit your code
    - either as a text file(s), "compressed/zipped" using some open-software product (so we could un- zip)
    - preferred - stored in your repo (and provide your TA and course instructor access to it)
  - Brief description of what your TA/instructor will have to do to access your code and execute it (readme file in the repo)

For an appeal of your grade, see the last paragraph in the below section on rubric.

Assignment 3 – Rubric

Rubric Categories

- Improved ER diagram and DB design for A2 … 5% (if there is not any improvements, resubmit the ER design of A1 as part of A2)
- Software (e.g., middleware, procedural SQL) … 35%
  - Functional requirements satisfied
  - Procedural SQL triggered by updating, deleting and inserting operation via web services.
  - Software can be organized using a middleware-DB backend services (web-services + DB-related)
- Submission … 50%
  - Format (e.g., format (e.g., title page, section headings, use of white space))
  - Submission organization (structure of the submission, logical flow of material)
  - Dal FCS Git submission instructions will be here (including Readme file)

- Evaluator impression … 10%
    - Software organization
    - Supporting documentation
    - Notable assignment features over and above requirements

Rubric

The rubric is presented by first describing requirements to obtain a C grade range (satisfactory), meaning that the requirements were mostly satisfied, however, the Good performance (required for a B range) has not been met. Then, B range is described, following by the A (Excellent) range. Clearly, any submission that is not satisfactory is below C.

Excellent: A- to A+ range

As above and software works correctly with submission quality being classified between very good to excellent. Your marker is impressed with your work. As above but the functional requirements are satisfied, and all evaluation categories must be classified as good and variations (range of "mostly good" -> "good" -> "very good" determine which of B-, B, and B+ applies.

Good: B- to range B+ range

As above but the functional requirements are satisfied, and all evaluation categories must be classified as good and variations (range of "mostly good" -> "good" -> "very good" determine which of B-, B, and B+ applies.

Satisfactory: C- to C+ range

The submission has sufficient information to convince the marker that the main objectives of the assignment were more or less satisfied. Documentation organization, presentation and completeness will affect the variations with the range …

Appeal process

Although we do strive for consistency and correctness in marking, we are all humans and mistakes can be made. If you feel that your assignment has not been marked appropriately, you may appeal it within a certain time frame. In short, when appealing your mark, whether for an assignment or a quiz/test, you need to describe specifically which part you feel was marked wrong and why it was marked wrong while making references to the parts of your answer (e.g., identifying statements or code parts) that were marked wrong and stating why it was marked wrong. Please see instructions on Brightspace re appealing your grades by submitting a request for appeal on the assignment box labelled *Grade-Review-Request-Assignment* (*Assessments -> Assignments -> Grade-Review-Request*).