Objective

In this project, each group needs to design and implement an online retail database application.

Basic Requirements

- There are many shops in an online retail website. Each *shop* has a shop id, shop name, rating, location, and a list of items on sale in the shop. Each *item* has an item id, name, price, and at most 3 keywords to describe the item, which depend on the shop.
- For every *customer*, we must capture customer id, telephone number, and address.
- Each *order* is made by a customer and must contain at least an item of a shop. An order can contain items of different shops.

Required Functionalities

- **Shop management:** You should be able to 1) show all shops, 2) add a new shop to the database.
- **Item management:** You should be able to 1) show all items of a shop, 2) add a new item to the shop.
- **Item search:** You can search items by keywords. For simplicity, you can only type a keyword. If the name of an item or a keyword of an item fully matches the keyword in searching, the item should be outputted.
- **Item purchase:** You should support item purchase. Record in database which customer purchases which item.
- **Order canceling:** You should support order canceling. You can cancel some item(s) in an order or cancel the entire order.

System Implementation

- Write a Java or Python program (command line interface or GUI) to implement the online retail application and required functions. For version, Java is Java11 and Python is Python3.5.
- Use JDBC and PyMySQL to access MySQL from the Java and Python program, respectively. MySQL is open source and can be freely download here.

Project Schedule

The project should be carried out in groups of 1-4 members, which includes two phases.

- **Phase 1: Group forming**

  Each group should have a group leader and the group leader should send the list of group members to Ms. jingjing Wang csjjwang@comp.hkbu.edu.hk, by 11:59pm, 17th March 2023. Kindly note that each group will be graded using the same standard regardless of the number of members. If you do not send the email to Ms. Wang before 11:59pm, 17th March 2023, you will be in a single-person group.

  Each member of the same group will receive the same marks, so you need to form your group carefully and learn how to collaborate with your group members. Note that it is not necessary to be good if there are many members in a group.

- **Phase 2: Database design & System implementation**

  Each group must design an ER diagram and convert the diagram to relational tables.

  Each group must implement the required functionalities.

  Each group needs to submit a soft copy of the final report and a soft copy of your code of implementation.

  Each group needs to write a readme file to specify how to run your code.

  Each group needs to record a video, which must be less than or equal to 6 minutes, for selling the database system to the instructor (customer), which includes the design of ER diagram, the design of relational tables, and the demonstration of functionalities.

Grading Criteria

- **Correctness (40%)**: You will get full marks if your implementation is correct. Partial credit will be given to a partially correct submission.
- **Video demonstration (40%)**: You will be graded based on the following criteria.
  1. Clearness of the presentation (for explaining the ER diagram, relational tables, and functionalities) (20%)
  2. Language proficiency (10%)
  3. Number of functionalities in the demonstration (10%)

  Note that the total marks of this component for each group will be multiplied by the factor $(3 - x) \div 3$, where x is the number of minutes that the video time exceeds 6 minutes. If the video is more than or equal to 9 minutes, we will give this group zero mark in this component.
- **Documentation (20%)**: The report should be short, clear, concise, and informative.

Submission (due by 11:59pm, 14th April 2023)

- Each group should submit a compressed file named **groupX.zip**.

The groupX.zip file should include the following items:

1. **groupX_project.zip** (the Java or Python project source files, which include comments of all your implemented functions.)
2. **groupX_insert_sql.txt** (the SQL command file for creating your tables (with constraints) and inserting sample data)
3. **groupX_report.pdf** (the project report should include: 1. group members, including student IDs and names; 2. ER diagram, table designs, normalization (if any), and the corresponding explanation for helping the instructor/TAs understand your design; 3. a readme description for running your code.)
4. **groupX_video.mp4** (the video)

Note: Replace '**X**' with your **group no.** in the above-mentioned files.

- Upload your compressed file **groupX.zip** to BUMoodle.
- Only ONE submission is required for each group, for multiple submissions, only the latest one will be collected.

Notes

- **Plagiarism (complete/partial copying of other people's work or sharing your own work with other groups): Those groups that involve in this case will get a zero mark.**

- **Late Penalty: The total marks of the late project will be multiplied by the factor (100 - 50x)%, where x is the number of days it is late (0 mark if $x \geq 2$). Any exceptions to this rule must be made prior to when the project is due, and the excuse needs to be a good one – just too busy won't cut it. Individual exceptions are unfair to other groups. Hence, they won't be made unless the circumstances are truly exceptional.**