

CSCE 4350.001 – INTRODUCTION TO DATABASE SYSTEMS DESIGN (summer 2020)

Project Title:

A database-driven Logo Store System.

Objective:

Design and implement a standalone software product to handle the entire operation of a Lego store like Bricks and Minifigs. The software is based on a relational database to access and manage the data. The major tasks of this project include identifying information requirements, conceptual data modeling, logical database design, and database implementation.

Project Phases

The two phases of the project cover the following work processes.

Phase	Description	Output	Due
Phase I	System Implementation	All Code for functioning system uploaded to Canvas	Jul. 20
Phase II	Presentation	PowerPoint presentation with audio uploaded to canvas.	Jul. 27

Groups

Groups of 3 or fewer students are required.

Deliverables

1. A report shall be generated for **Phase I** and must contain a short description of the purpose of this phase. It must describe the problems encountered and justify the solutions. It must contain all documentation produced, ER Diagrams, Relational schema, and abstract code with SQL. The relational schema should include integrity constraints, foreign keys, primary keys, and any constraint keys. The report plus any code should be uploaded to Canvas by the due date. It should include both scripts SQL scripts to create the DB as well as a Script to populate the database so that testing can be done on both an initial database and one with enough data to resemble a small setup. It should contain system manual, all sources of programs.
2. A PowerPoint presentation shall be created for **Phase II**. It must contain audio recorded audio as part of the PowerPoint presentation that describes the slides. It must describe the problems encountered in the and justify the solutions. Also, it must contain the results of creating database schema on the database you

choose, e.g., SQL script, sample data, or captured screen shots. This should represent your demo to the customer.

Available Platforms and Languages

You are restricted to a few programming languages and databases. This is to allow consistent grading across all projects and be supportable by the instructor and TA. Your team can choose the program language, relational database and DB modeling tool in the Phase I, based on members' skill and system availability. The available options are as follows:

- Database
 - (1) Microsoft SQL Server
 - (2) MySQL: you can download from internet (Freeware).
- Program Language
 - (1) PHP
 - (2) Python
- Database Connectivity
 - (1) Should be an embedded SQL. These are readily available for both SQL Server and MySQL
- User Interface
 - (1) Command line interface like DOS

* Feel free contact to TA if you have any concern for choosing your options of the project.

Lego Store System – General Functionality

**** Important!!! The following specifications are intended as a guide; they are NOT the complete, nitty-gritty specifications. These are intended to get you started in the right direction in designing your system. You as the designer must analyze and decide what other details or features should be specified for your system. Thus, individual group implementations will differ in terms of design and implementation styles.*

A store that sells Legos has employed your team to computerize their entire operation. They are currently managing all of their activities in a paper-driven environment with some rudimentary use of a computer. Their business is in the process of expansion and they realize that computerizing the operations is a necessity at this point. They would like to have the system up and operational before the end of this summer. In addition to the set of minimum requirements listed below, they are open to any recommendation in functionality as well as look & feel of the product. Legos are sold as both Sets and individual Bricks. Sets contain multiple

bricks of various types. Prices Bricks should all have prices. Set prices should not be stored for sets but derived from individual brick prices.

Example Set:

Set 40367 Lady Liberty

Quantity	Part No.	Description
1	6175968	Square
2	300328	2x2 Brick
2	6227073	4x4 Plate

1. The software system must handle their current two stores as well as seamless expansion to multiple stores potentially in different locations.
2. Two modes of operation must be supported, “Online mode” and “Store mode”.
3. “*Online mode*” suggested operations:
 - Allow placing an **order** from the internet
 - Support credit card **payment** only (AMEX, MC, Visa, etc.)
 - **Customer**’s personal information (e.g. name, address, phone, e-mail, etc.) must be kept in the system for billing and future purchases.
 - Customers and the system should be password protected.
(**authentication**)
 - Customers may **browse** through all existing sets and bricks or **search** for particular items.
 - When a set or brick is selected, various details should be **displayed** (e.g. price, available quantity.)
 - **History** of customer’s previous activities should be readily available.
4. “*Store mode*” suggested operations:
 - Handle store **employee** information (i.e. add/change/delete employee, and their access level assignment)
 - All employee must have logon id and password to access the system
(**authentication**)
 - The software should have appropriate sections to handle the following day-to-day activities:
 - **Sale**: Sell sets or bricks to customers. Customers should be able to use credit cards as well as cash. Should also handle returns.
 - **Order Management**: Order sets or bricks, reorder, stop orders, returns.
 - **Database Management**: Update database with all personnel and parts information
 - **Report Management**: Generate various reports on the parts and employee status
 - **Delivery Management**: For internet orders only. Should minimally include: Order date, delivery date, payment method, cancellations, etc.

Note: You can find a lot of reference sites to this project, such as <https://www.lego.com/en-us> or <https://bricksandminifigs.com>

5. The system should be menu-driven system. The following is an example of Database Management menus.

**** The followings are just examples for command line interface.
Also, you can add more menus depending on your requirements. Be sure to capture all of the requirements in the menus that you create.*

Login

1. Username
2. Password

Order Sets or Bricks

1. Set or Brick number
2. Name
3. Price
4. Quantity

Payment

1. Payment Option (Cash, Card)
2. Card Type
3. Card Number
4. Billing Address
5. Amount

Customer Information

1. Name
2. Address
3. Store Preference
4. Username
5. Password

Search Item (Set or Brick)

1. Part Number
2. Description

Employee Management

1. Employee ID
2. Employee Name
3. Store Preference

Order Management

1. Order new parts
2. Reorder parts
3. Stop Order

Reports

1. Daily Reports
2. Weekly Reports
3. Monthly Reports

Grading

The project will be assessed according to the following criteria:

1. Functionality: Does the project provide all the required functionalities and desirable others? Does it capture the domain in sufficient detail?
2. Design Quality: To what extent does the database design measure up to the design principles presented in the course?
3. Documentation: How well are the design and implementation documented?
4. Demonstration: How clearly is the work presented to the Instructor and TA?

Recommendations

1. Database
 - SQL Server <http://www.microsoft.com/sql/default.mspx>
 - MySQL <http://www.mysql.com/>
2. Database Connectivity
 - SQL Server: <https://docs.microsoft.com/en-us/sql/connect/sql-connection-libraries?view=sql-server-ver15>
 - MySQL
 - Python: https://www.w3schools.com/python/python_mysql_getstarted.asp
 - PHP: https://www.w3schools.com/php/php_mysql_intro.asp
3. Language
 - PHP: <http://www.php.net/>
 - Python: <https://www.python.org>
4. Design and Project Management tools
 - DBDesigner: <http://www.fabforce.net/dbdesigner4>
 - DataGrip: <https://www.jetbrains.com/datagrip/>
 - DBVis: <https://www.dbvis.com>