Building a complex online shop system from scratch with MySQL and Django

Team Members : Zexiang Chen[1,*], River Liu[1,*], Shiyuan Zhang[1,*], and Siyuan Zhao[1,*]

[1]Dept. of Computer Science, University of Illinois at Urbana-Champaign
[*]{zexiang5, ll24, sz54, siyuan6}@illinois.edu
[+]these authors contributed equally to this work, and are ordered alphabetically

ABSTRACT

Today, with the prevalence of e-commerce, studying the details of how to build complex e-commerce systems become increasingly important for every computer science students, especially those aspiring to become a database expert. In CS411, a class focusing on database management systems, we built one such system from scratch using MySQL [2] and the Django [1] framework of Python. In this process, we faced many important design decisions and made a few changes on our initial design. This reports seek to be a detailed discussion of these design decisions we made and the rationales behind them.

Overview

In this report, we seek to answer the following 10 questions:

1. Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).
2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.
3. Discuss if you changed the schema or source of the data for your application.
4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?
5. Discuss what functionalities you added or removed. Why?
6. Explain how you think your advanced database programs complement your application.
7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.
8. Are there other things that changed comparing the final application with the original proposal?
9. Describe future work that you think, other than the interface, that the application can improve on.
10. Describe the final division of labor and how well you managed teamwork.

Organization

Answer to each question will be organized into its own section, with the title of the section being the question itself.

1 Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission)

There are indeed some differences between our project and the project proposed by stage1. The first is that the target users we originally envisaged were the residents of Champagne. However, during the design process, we found that by designing the account login function, consumers, store owners and managers can use this database system together. And compared with the original proposal, the functions we currently implement are more realistic. For example, our statistics on consumer orders are not included in the original proposal, and we have also added the function of calculating discounts.

2 Discuss what you think your application achieved or failed to achieve regarding its usefulness

During the initial design of this program, we originally planned to design a fun function that randomly recommends pets by keywords. However, in the process of later implementation, it is found that it is relatively difficult to add this function to the entire application, and it is not a necessary component of the entire program. Later we chose to cancel this feature.

3 Discuss if you changed the schema or source of the data for your application

Compared with the data source provided in stage3, we have made some changes to it. But this change does not add or change existing data information. This change is related to a change in our database. Mainly because we added a new attribute discount on customer. Then for this data that we think cannot be empty, we add a default value of 1 to it. So in fact, it can't actually be said that the data source has been changed, it's just a new column of attributes that were forced to be added to the database design. The application schema is basically the same as we originally proposed.

4 Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

For the ER diagram we originally proposed, some changes have taken place in our final result area. The first is that we removed the user entity and replaced it with customer. Because the user of our program is store owners, consumers, and managers. So this change is closer to reality. In addition to this, we also changed the customer's primary key. Because we found that the original design of the primary key is not reasonable. Using the name as the primary key will cause duplication problems. Next is relative to the er diagram in stage4, we also added something in the final demo. In the final demo, we added two properties, discount, and recommender, to customer. This is to facilitate our ability to implement discounts and referrer offers.

5 Discuss what functionalities you added or removed. Why?

As we mentioned before, I think the functionality of this program is increasing. Although we didn't implement the creative functionality that was proposed in the original design, we changed the object orientation of the program. And added functions such as statistics of the total order price and calculation of discounts. I believe such changes can help improve the functionality of our programs.

6 Explain how you think your advanced database programs complement your application

In order to make our project more realistic and better-fit people's real needs, we added a discount for each customer to our database (the default discount is 1, which means the person will buy at the original price. If the discount corresponds to a value of 0.9, it means that the person will buy at a 10% discount). New registrants have the opportunity to fill in the name of the referrer, and both the referrer and the referee have the opportunity to receive the discount. Customers will also receive discounts if their total orders reach a certain amount. Because of this design, the trigger + stored procedure is the most suitable for our use cases, and it is an insert event that the process of filling in the referrer information when a new user registers. Moreover, we want discounts to be ranked by the total amount they purchased and the number of other people they referred. After we insert these new order information and new customer information, old customers may refer more people and it is possible that the total consumption of old customers will increase. Therefore, the store procedure is suitable for this use case. When we call this procedure, the program will update the current consumer's discount based on the number of other people he referred and the total consumption. As an advanced database program, they implement the complex data integrity rules of our database, automatically calculate data values, and replicate data from tables in real-time and synchronously.

7 Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project

(a) In the early stage of the project, how to design rationally tables, attributes, and define integrity constraints is a major difficulty we encountered, because we have a lot of different categories of data, for example, the 'pet' table has nearly 30,000 rows of data. Integrity constraints must ensure the stability of the database and the rationality of data.

(b) The front-end application is the most challenging part of the project since no one has experience in the front-end application. How to select, learn, and deploy a front-end application in a short period of time is a test of our teamwork, and we must find a tool that best suits our pace and coding language. We used Django as our tool since the front-end and back-end coding languages are the same.

(c) When we connect your database to the front-end application. We encountered a challenge that the response time of the search function of the page is very slow. We think we need a well-designed paging feature to improve it, otherwise, the cost will be extremely high.

(d) An advanced database program is a vital part of the database system and a difficult part of the project. The database needs many advanced database programs to ensure its consistency and integrity, so we need to design more triggers and procedures for the database. For example, we need to "hide" the pets that have found a home.

8 Are there other things that changed comparing the final application with the original proposal

Due to time limits, we are unable to complete some of the designed features. For example, "Special Events" is a creative function, which will automatically generate a "mysterious box" for customers who do not have particular preferences. Moreover, we've made cool plans for the interface, but the UI design of the final application is not good enough, we only designed simple pages and layout for our application. Also, we planned to add picture for each product in the original proposal, but we just contained the price and some basic information.

9 Describe future work that you think, other than the interface, that the application can improve on

In future work, we plan to add more triggers and stored procedures to ensure the integrity of our advanced database. We also need to lower the searching cost since it's very slow, which can reduce the user's real experience. Furthermore, we will try different database systems to store our data, such as Oracle database, Neo4J, and MongoDB, it is essential to investigate whether different databases will improve efficiency and other positive effects to our application; different cloud servers may also have different experiences, such as Amazon Cloud.

10 Describe the final division of labor and how well you managed teamwork

In this project, Siyuan Zhao and River Liu are mainly responsible for the front-end application; Zexiang Chen and Sean Zhang are mainly responsible for the database. In general, our group divided and collaborated efficiently in every collaboration. The group leader carefully assigned each assignment based on each group member's situation before each stage. For example, we have a meeting on Zoom before each assignment, and every group member would talk about his insights and challenges. In the process of project, if someone encountered a technical problem, they will also be time to raise the issue and work together to solve it, so that we can achieve the project milestones efficiently and completed it with high quality. The key point to the success of our teamwork is that we follow a way of communication based on mutual respect and mutual trust, therefore, it brings together group wisdom to our project effectively.

Acknowledgement

We would like to thank our wonderful TA, Sophia Yang, for providing quick and useful feedback throughout the process and guiding us to the right way.

References

1. Django Software Foundation. Django. https://djangoproject.com.

2. Oracle Corporation. Mysql. https://www.mysql.com.