Project Report

Team 103

December 8, 2022

Contents

1 Changes in Final Project

In our project proposal, we had more computer entities, such as Monitor, Keyboard, SSD, etc. We removed these entities in stage.2 when we designed our ER diagram. In our final application, we only have CPU, GPU, RAM, and MotherBoard.

In our project proposal, we should have the price information of different products. However, we found it difficult to collect the correct information of different products because the price information is not always fixed. We did not choose to fabricate data in stage 3 because we wanted to keep the realness of our data.

We did include more functionalities in our final application. We included realiseTime which can be automatically generated by triggers once the releaseDate of one product is specified.

2 Things Achieved and Failed

We have successfully encapsulated our adaptation check functionality in two advanced queries. In the original design, we can only check the adaptation of two specific components, which is not efficient. In our final version, we only need to type some keywords, and the advanced queries will return any adapted accessories. For example, we can type the name of one motherboard, and the advanced queries can display the information all adapted GPUs.

We have developed a login in interface which is better than our original thought. Our log in interface allows users to register their account, and the user information will be automatically inserted into our database. We will display error message if the user is already exist in our database. When correct account name and password are entered, the website will jump to our user interface.

We failed to provide the functionality to allow users to create their own PCs because we do not have enough information of the computer accessories. We failed to separate the authorities of administrators and regular users. In our final application, anyone who have successfully logined in can edit the database information through the button in our user interface.

3 Source Change

We fabricated RAM data because we cannot find a RAM dataset which contains different brands of RAMs along with the features we need. The ReleaseDate information was fabricated as well. When we inserted real data into our database, we did not have ReleaseDate information. This information was specified to meet the requirements of stage.5. All of the information we inserted into CPU, GPU, and MotherBoard datasets are real.

4 Change To ER Diagram

We removed some features from our entities. We removed userType, Name, and PhoneNumber from User entity; we removed Model from GPU entity. We removed CustomizedPC entity as well. The reason why we removed these information is that these information should not affect the functionality of our program, and removing them can reduce a lot of workload in our

implementation.

We do not have the CustomizedPC entity in our final application because we do not have enough computer accessories such as monitors and keyboards. Moreover, it is hard to get the correct price information; and there is no standard numerical information to specify the performance of different computer accessories. It is meaningless to include the CustomizedPC entity if we do not have any measurements of the performance or the price of different accessories.

5 Functionalities added and removed

5.1 Functionalities added

We added a releaseTime information to different products to finish the trigger requirements of stage.5. In addition, we believed that when the users want to find a motherboard that is adapted to the CPU they entered, the product information of all adapted motherboards should be display. Originally, there was no way for users to find the best product among all these information. Therefore, we labeled each motherboard a new feature called "Status" which is based on the support socket type of the motherboards. This new feature should show "advanced", "medium", and "low" as a measure of performance of the motherboards. This functionality is supported through stored procedures.

5.2 Functionalities removed

We removed the functionality that allows users to create their own computers, because we did not have enough entities such as SSD and monotor to build a customized computer. We removed the performance score of the products because we did not find a way to numerically define the performance of CPU, GPU, RAM, and MotherBoard.

6 Database Application Complement

The powerful database we created is critical to supporting the functionality of our application. For example, the application displays the product information when the users click the CPU or GPU buttons. In fact, this information is generated using a query that joins the CPU or GPU table with the Product table based on the productId information. In our advanced query designs, the users only need to type some keywords of the accessories. For example, if the users want to know the information of motherboards that are adapted to CPU "Ryzen 7 5700X", they only need to type "Ryzen 7" or "5700"; then our application will do a keyword match and return the result of the advanced query.

7 Technical Challenge

Jixuan Lu: Design a detailed and beautiful user interface first, then use HTML, CSS, and JavaScript to implement it. Use Ajax and Jinja2 to make the frontend HTML page dynamic, in which the elements can adjust based on backend data records. Ensure the transferring of commends from frontend to backend is through the correct route.

Zijian Pei: If multiple inputs need to be transferred from user interface to Flask in order to

execute queries, it's best to include these information in a JSON String. If a "LIKE" command needs to be executed in Flask, two "%" must be added instead of one.

8 Other Changes

No other things changed comparing the final application with the original proposal. All changes are included in section 1 to 4 of this report.

9 Future Work

We can definitely add more accessories, and include the price information of each product. Price is always a useful information for customers to select computer accessories. In that case, we can support the CustomizedPC entity, and therefore allow users to build their own computers.

We believe it is inappropriate to allow our customers to edit the product information entered by us. It's better to separate the data generated by users and administrators. In the user interface, when the user click the button, the database should display both the data generated by users and the administrators. For example, when the user click the "CPU" button, the user-generated CPU and the CPU information inserted by us should both be displayed.

10 Division of Labor

10.1 Stage 1

Juxuan Lu: Drew user interfaces
Zhanyu Feng: Designed projects, Wrote report
Zijian Pei: Wrote Report

10.2 Stage 2

Juxuan Lu: Drew ER diagram, Wrote report
Zhanyu Feng: Wrote relational schema
Zijian Pei: Wrote report
Zhexuan Yin: Drew ER Diagram, wrote report

10.3 Stage 3

Juxuan Lu: Wrote report, collected data
Zhanyu Feng: Designed advanced queries, created data
Zijian Pei: Wrote report, loaded data into database
Zhexuan Yin: Create tables in database

10.4 Stage 4

Juxuan Lu: Created the whole user interface
Zijian Pei: Fixed bugs in database and user interface

10.5 Stage 5

Juxuan Lu: Implemented interfaces to support triggers and stored procedures
Zhanyu Feng: Generated datasets to support stored procedures
Zijian Pei: Wrote triggers' and stored procedures' queries

10.6 Stage 6

Juxuan Lu: Recorded video
Zijian Pei: Wrote report