E-commerce website

Carl Frisenstam - carfri-5@student.ltu.se
Jacob Eklund - jacekl-5@student.ltu.se

January 3, 2018

1 Introduction

This is the final report for a software development project that started in early November and was finished late December 2017. The project group consist of 2 people, Carl Frisenstam and Jacob Eklund. The project was part of a course in databases (D0018E) and as such the focus of the project was mainly on the back end of software development, database design and database application programming. The purpose of this report is to provide a comprehensive overview of the development process without getting bogged down too much in technical details.

2 Project Description

2.1 Project Overview

The product aimed to be produced was a simple E-commerce website where customers could create user accounts and then go on to browse the inventory of the web store. The customers should be able to add products into shopping bags. The customer should then be able to purchase the products placed in the shopping bag. A rating system should also be implemented so that customers can rate specific products on a scale from $1 - 5$ as well as leave comments on said product.

Tools for store owners should also be implemented so that store owners can administrate their web store without having to modify database tables directly. The ability to modify the store inventory and prices are an example of such tools.

2.2 The Purpose of the Project

Seeing as this was a project created as a part of a course in databases the main purpose was development of skills. To familiarize with and to learn some of the tools available for web-development. Other than technical skills there was also a significant focus on skills related more to the development process and how you go about managing a software engineering project. A large focus of this project was to familiarize yourself with agile methodologies and then apply these methodologies to the development process.

3 System Architecture - Implementation Overview

3.1 The Database

The database schema was created with MySQL workbench and hosted localy on LTU servers. phpMyAdmin is used to handle administration of the database. The database have undergone plenty of changes during development, the current database schema can be seen in the figure below.

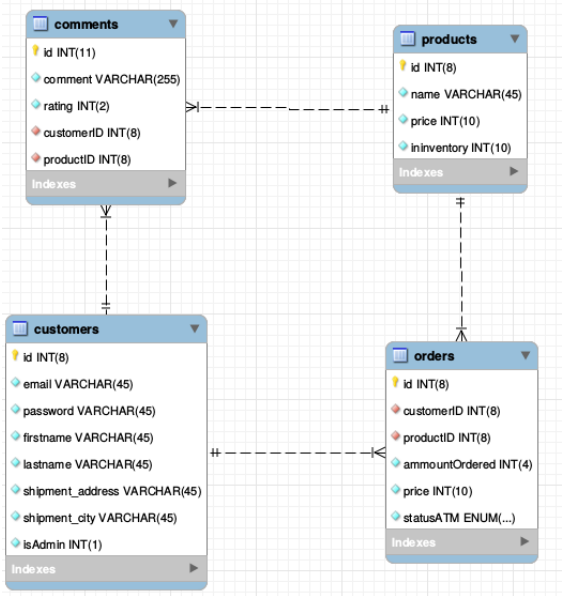

Figure 1: The database schema generated in MYSQLworkbench.

The 'id' variable in each table have been set to primary keys. For the 'customers' table the 'email' variable was set to unique since this will also function as a username for the login portal. In the 'orders' table the 'customerid' and productID have a foreign keys that references the 'id' variable in the 'customer' and 'product' table. Similarly the 'productID' and 'customerID' variables in the 'comments' table each have a foreign key that references the 'id' variables in the 'product' and 'customer' tables. These 4 tables are sufficient to achieve what we wanted to do in this project. By using 'join' and nested SQL queries we managed to implement desired functionality without creating unnecessary tables and rewriting data that already exist in other tables.

3.2 Web page Functionality

The current features have all been implemented with php. When visiting the webpage the user is directed to a index.php file that present the user with a login form. He can the choose to login or click the register button. This will direct him to a registerform.php file which will offer up the opportunity to register a new user. When a new user have been registered there will be a redirection to the index file which will take the user back to the login screen. When the user have logged in he will be directed to a 'userwelcome' screen on which he can view the store inventory and place items into his shopping basket. He can also click on a link that will direct him to the shopping basket, there he can view any items currently in the shopping basket, as well as remove remove those items from the shopping basket. If an admin user is logged in he we instead be redirected to a 'adminWelcome' page on which he can access various admin tools. The admin can see all items to ship in the orders.php page.

3.3 Version Control and Link to the Code

Github have been used for version control and it has worked well. Link to the code on github

3.4 Backlog

1	Backlog				
2	Description	Priority	Time cost	Dependancy	#
3	Create database	1	3h	-	1
4	Store shelves	2	2h	1	2
5	Login system	3	4h	1	3
6	Web page (UI)	4	1h	1	4
7	link login with web page	5	2h	3 4	5
8	Add items to shelves	6	1h	1 2	6
9	display shelves on web page	7	2h	1 2 4 6	7
10	shoping basket	8	8h	1 3 4 5	8
11	rating system for wares	9	8h	7 8	9
12	increase web page functionality	10	5h	4	10
13	item options	8	3h	6	11
14	store management tools	5	4h	2	12
15	mySQLworkbech schedule	1	5h		13
16	comments on items	9	5h	7 8	14
17	improve on visual design	11	-	4	15
18					
19	sprint #				
20	1	#1-4, 13			
21	2	#6, 7, 12, 11, 8			
22	3	#5, 9, 10, 14, 15			

Figure 2: Development backlog.

Keeping a backlog during development turned out to be very helpful. The ability to visualize work tasks in such a manner helped to mitigate some of the stress that can be experienced when starting a new project. Being able to plan what work needed to be done during the upcoming week kept development from becoming messy. It was easy to keep in sync with the team when tasks could be condensed and put in a simple list. The planing stages became easier when you could refer to the backlog and in the end planning for this project turned out to be very accurate. The planning for the backlog have worked out well for all three sprints. All items have been achieved in each sprint in a timely fashion.

4 Applying newfound knowledge to future work

The team haven't run into any major problems. The only issues have been due to inexperience with the php programming language. We also had some git merger errors but they were easily fixed. In the future we think that a better use of tools would be very helpful. For example the SQL workbench have been very useful in this project. But all the code have been written in notepad, the development process would probably have been a lot smoother if a proper IDE was used instead.

4.1 Limitations

The scope of the project was limited due to the short time available. This is clearly noticeable in the layout and design of the web page. Graphical design wasn't in focus during development. Functionality had to be limited as well due to short development time. For example a system where users can reply and rate other users comments could be useful. Web page security is currently in a very poor state and vulnerable to SQL-injections due to the way user input is handled.

4.2 Possible Improvements

There is much room for improvement when it comes to security. Sanitizing user input is probably the first thing that should be done though. Then there is obvious room for improvement in website

design. Some functions are currently clunky to use. For example the shopping bag could be put in a drop down menu rather than a all together different page.

5 User stories

- As Customer, I want an easy to understand website layout so that I can make quick purchases.
- As Customer, I want to be able to search and use filter so that finding items is easier.
- As Customer, I want to be able to see ratings and comments so i can make a better purchasing decision.
- As Customer, I want to be able to leave ratings and comments so that other customers can gain information about a product.
- As Customer, I want to be able to add more than one product at a time to my basket for a smoother shopping experience.
- As Administrator, I want a control page so i can mange the store.
- As Administrator, I want to be able to remove inappropriate comments.
- As Administrator, I want to be able to see what items to ship.

6 Use-Cases

6.1 Use case specification - Register a new user

6.1.1 Brief Description

This use case allows a new user the registration of an account. It allows a user to login to an already existing account.

6.1.2 Actors

- Primary actor - Customer
- Secondary actor - login/register system

6.1.3 Flow of Events

- New user
  This use case starts when a new customer clicks on the register button on the front page. They are then redirected to a register form where they will input profile informations such as email, password, first name, last name etc. The system saves the information in the database.
- Login
  The user puts his information into the login form. The system validates the information by referencing with the database.
- Welcome screen
  If the user entered with a customer account they are redirected to the customer welcome screen. If the user entered with an admin account they are redirected to the admin welcome screen.

6.2 Use case specification - Purchase an item

6.2.1 Brief Description

This use case allows customers to browse a list of item available for purchase. The user can enter a product page for each item where they can then add a number of said item to their shopping bag. The customer can then purchase all items in the shopping bag buy clicking the buy button.

6.2.2 Actors

- Primary actor - Customer
- Secondary actor - Storefront system

6.2.3 Flow of Events

- enter storefront
  The system checks the database on what items are currently available for purchase. A list of products are then presented to the customer.
- add product to shopping bag
  The customer can add listed products to their shopping bag by clicking the buy button next to the desired item. An option to add a specific amount of a certain product is available by typing in the desired amount in the text box and then clicking buy. The products are then added to the database by the system with relevant customerID attached to the entry.
- check out shopping bag
  The customer can purchase all the items in the shopping bag by clicking on the casket link which redirects them to their personalized shopping bag page and then click on the buy button. When the customer clicks the buy button a flag is raised for each item and the status are set to 'bought'.

7 Assumptions

This web page can function assuming certain resources are available. First off it is assumed that the web page is hosted on a stable server which the user can connect to. It is also assumed that the customer have access to an email account since that is required to create an account. It is assumed that the store administrator as well as the customer possesses a certain amount of competence navigating web pages since there are no support or help resources available to the users.

8 Code and Testing

All code can be found github: https://github.com/carfri/D0018E

- Test scenario shopping basket: Check if relevant products are viewable in the basket.
- Test case add item to shopping basket: Check if relevant items appear in the basket when added from store shelves.
- Test data: productID: 1, customerID: 2, product name: onion, amount ordered: 1
- Expected result: When 'buy' button is pressed one onion should be added to the basket of the $id = 2$ customer.
- Test case remove item from shopping basket: Check if clicking the remove button removes relevant product.
- Test data: productID: 1, customerID: 2, product name: onion, amount ordered: 1'
- Expected result: When 'remove' button was pressed the onion product should be removed from the basket.