Online Clothing Retailer

Part 1

1.1Investigate

Project Development Timeline

1. **Requirements, Research and Consultation (1 week)**
    - Talk with the family member and collaboratively identify the consumer trends relative to potential store features for clothing retail.
    - Via conducting market research → obtain knowledge of customer preference and how to stand out with the competitors.
    - Request knowledge of the project's scope.
    - Use the known knowledge of the project scope to create a flowchart of the system being used, this will allow for identifying the gaps.
    - Propose solutions for gaps in the system to create an ideal environment for running the retailer.
2. **Design (2 weeks)**
    - Use the flowchart developed from the "**Requirements, Research and Consultation**" phase and create a system architecture (includes a retailer website, database, and industrial computer system setup). Make sure the system architecture has explicitly accounted for the database implementation in the system architecture. → Use dataflow diagrams for the design process for a more standardized application (and for future reference).
    - Use the system architecture to design the database schema.
    - Use wireframes to design the user interface.

3. **Development (4 months)**

- Set up a development environment (use Visual Studio code, DB-browser and google cloud)
- Develop the backend; obtain the API keys from google cloud and prepare to implement the database into the frontend).
- Develop the frontend.
- Integrate google cloud (backend) into the frontend.

4. **Testing (2 months)**
    - Perform testing in units to test individual components.
    - Test the components together in a system.
    - Report tests and findings.
5. **Deployment and final tests (2 months)**
    - Deploy the solutions to a live environment to simulate user interaction.
    - Perform user-acceptance testing to gain feedback on the implemented system.
    - Report tests and findings.
6. **Documentation (1 week)**
    - Showcase the final product
    - Create a user manual for troubleshooting and document the development process (done via a private GitHub repository) to review commits made to the system.
7. **Future Support (ongoing)**
    - Be available for future inquiries or hire a contractor for future troubleshooting.

Problem Outline

The objective is to create a software solution → comprehensive online platform for an online clothing retailer. The database in this solution will contain customer information and sales data.

- Provide an easy access point for the customer to browse and order clothing on the website.
- Have proper records on customer information and sales data.

Problem Description

The new online business offers various clothes to customers, according to the requirements of the family member. Customers should be able to browse a catalogue of various types of clothing and view extra information on each piece; add items to their shopping cart; and securely purchase their item.

General Requirements

- **User compartmentalisation**: User registration, login and profile customisation capabilities.
- **Product Catalogue**: Display products with description, pricing and images.
- **Shopping cart**: Users must be able to add/remove items from their cart.
- **Checkout Process**: Secure checkout process with credit card and service payment options.

Database Requirements

- **User Entity**: Stores user information (username, email, password, delivery address)
- **Products Entity**: Store details about each product (name, description, price, type, stock availability)
- **Orders Entity**: Holds information about each order (order ID, user ID, total amount and status)
- **Product Details Entity**: Stores information about each product (product ID, quantity)

Ethical, Legal and/or Security Issues

- **Data privacy**: Compliance with Australia's Privacy Act 1988 and ensure the Australian Privacy Principles are met.
- **Security**: As a part of the Privacy Act 1988, user information must be encrypted via SSL (Secure Sockets Layer) secure shopping, this encrypts data transmitted between users and the website; therefore, ensuring sensitive information can't be accessed by attackers.
- **User Consent**: Communicate terms and services to the user during registration and prior to usage of the website.
- **Accessibility**: Design the platform to include website feel/appearance options, this includes font option and size adjustment for the visually impaired and website themes including high contrast light/dark, general light, general dark and options for tritanopia and protanopia/deuteranopia.

Data Quality Factors

- **Consistency**: Integrity is maintained by ensuring data validity and adheres to defined constraints.
- **Completeness**: Ensures all data fields are populated.
- **Up to Date**: Ensures that data is updated to maintain relevance and allows for data to be easily updated.

1.2 Design

ER Diagram

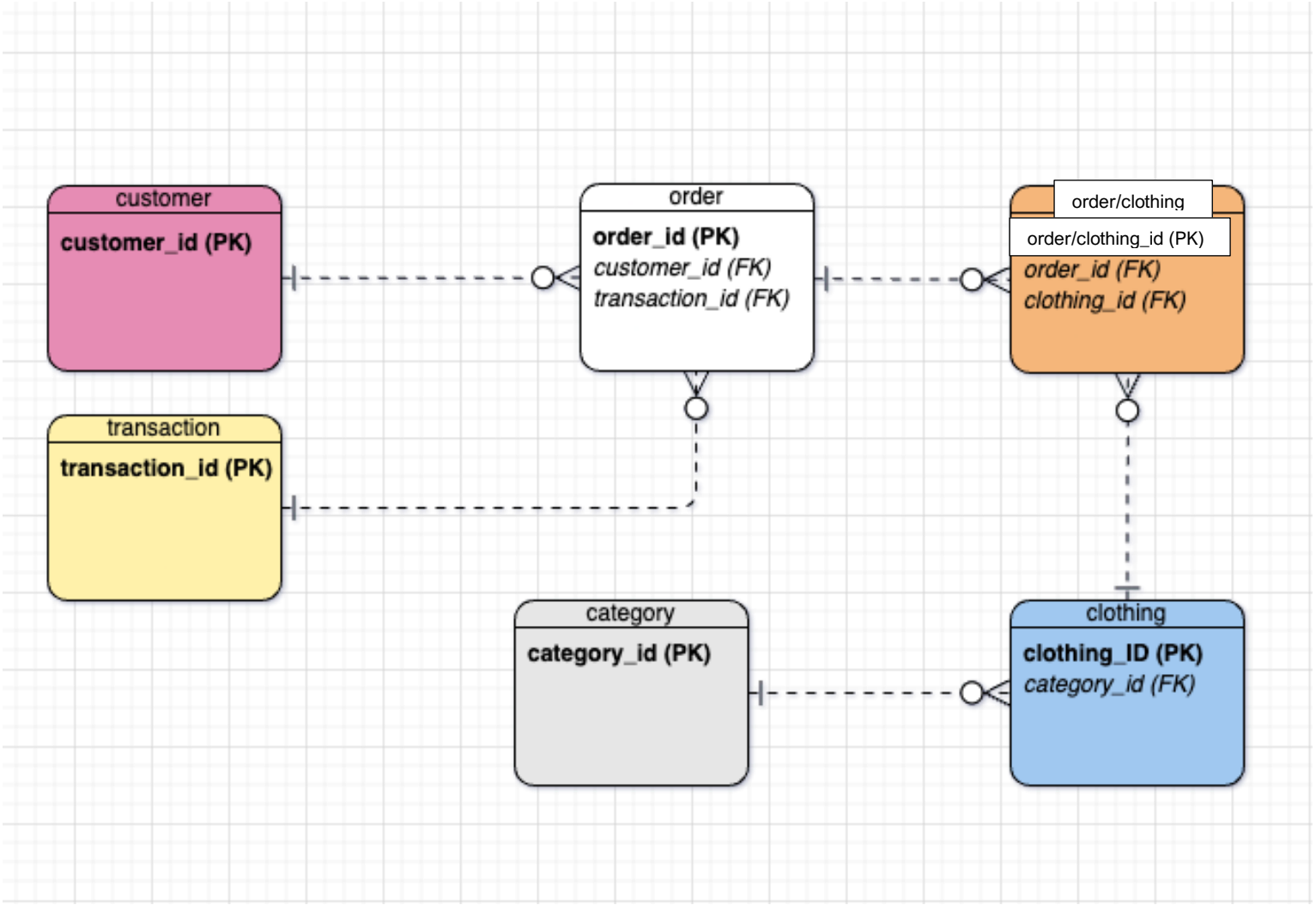

Figure 1 - Made with Visual Paradigm | PK = Primary Key; FK = Foreign Key

Relational Notation

CUSTOMER (customer_id, name, sex, address, phone number)

PAYMENT (payment_id, service provider, cardholder name, card number, expiry date, CVV)

ORDER (order_id, customer_id, payment_id, order_date)

CLOTHING (clothing_id, fabric, colour, category_id)

ORDER/CLOTHING (order/clothing_id, order_id, clothing_id)

CATEGORY (category_id, type)

Data Dictionary

CUSTOMER

Element Name	Data Type	Size	Description	Constraints
customer_id	Integer	10	Unique identifier for customer (auto generated).	Primary Key, not null
name	Text	20	Name of customer.	not null
sex	Text	1	Gender of customer.	not null
address	Text	20	Home/billing address of customer.	not null
phone number	integer	15	Mobile number of customer.	not null

PAYMENT

Element Name	Data Type	Size	Description	Constraints
payment_id	Integer	10	Unique identifier for payment (auto generated)	Primary Key, not null
service provider	Text	20	Credit/debit card provider/bank	not null
cardholder name	Text	20	Customer name on Credit/debit card	not null
card number	integer	20	Credit/debit card number	not null
Expiry date	date	5	Credit/debit expiry date	not null
cvv	integer	5	Credit/debit cvv	not null

ORDER

Element Name	Data Type	Size	Description	Constraints
order_id	Integer	10	Unique identifier for order (auto generated)	Primary Key, not null
customer_id	Integer	10	Unique identifier for customer (auto generated).	Primary Key, not null
payment_id	Integer	10	Unique identifier for payment (auto generated)	Primary Key, not null
order_date	date	10	Date of order	not null

CLOTHING

Element Name	Data Type	Size	Description	Constraints
clothing_id	Integer	10	Unique identifier for clothing (auto generated).	Primary Key, not null
fabric	Text	15	Type of Fabric	not null
colour	Text	20	Colour of Clothing/fabric	not null
category_id	Integer	10	Unique identifier for category (auto generated).	Primary Key, not null

ORDER/CLOTHING

Element Name	Data Type	Size	Description	Constraints
order/clothing_id	Integer	10	Unique identifier for relationship between clothing and order (auto generated).	Primary Key, not null
order_id	Integer	10	Unique identifier for order (auto generated)	Primary Key, not null
clothing_id	Integer	10	Unique identifier for clothing (auto generated).	Primary Key, not null

CATEGORY

Element Name	Data Type	Size	Description	Constraints
category_id	Integer	10	Unique identifier for category (auto generated).	Primary Key, not null
type	text	15	Unique identifier for type of clothing (auto generated)	not null

SQL Queries in Plain English

- Count Total Customers from the Customer Table.
- Find the Total Cost of the fabrics in the Clothing Table.
- Find the Average Cost of the fabrics in the Clothing Table.
- Find the Minimum Cost of the fabrics in the Clothing Table.
- Find the Maximum Cost of the fabrics in the Clothing Table.
- Number of customers under a service provider can be grouped.
- Cost of fabrics can be ordered in ascending order.

- Using multiple joins customer name, cost of bought fabric in clothing and service provider can be selected.
- Cost of 1000x units of the fabric can be displayed using a calculated field.
- Customer name and phone number can be displayed in a single column value using a concatenated field.

2.1 Develop

***Database.py in "Main" Repository**

2.2 Evaluate

Does the solution meet the requirements from Part A?

1. **User Compartmentalisation**: The developed system does not implement such functions. Therefore, security and authentication of user information is not present.
2. **Product Catalogue**: As the developed product is still a terminal application without a proper interface and dummy products, the requirement has not been met.
3. **Shopping Cart**: No shopping cart for saving orders has been implemented.
4. **Checkout Process**: There is no authentication nor secure payment procedure in place, there is the 'payment' entity which lists out dummy payment information shared along the tested customers.

Therefore, none of the general requirements have been met. This is due to the product still being at a very preliminary stage of development and has only been developed to produce a rough idea for the database structure to be developed into the final product.

5. **User Entity**: 'Customer' entity is present and functions accordingly, but the attributes like email and password have not been implemented.
6. **Products Entity**: 'Clothing' entity is present but lacks description and stock availability.
7. **Orders Entity**: 'CustomerOrder' entity is present, although status and total amount are not implemented.
8. **Product Details Entity**: There is an association been orders and products, but there is no system component in place for storing specific quantities of the purchased/available product.

ER-diagram comparison

The program follows the general structure of the ER diagram, although some of the entity names have been modified to not clash with the program and improves the ability to keep track of them as the program gets bigger.

Extra Features Implemented

- Ability for user to insert, delete and update entries.
- A terminal number-based selection panel for user to select options and see their function.
- All required SQL commands are implemented into the program.
- Saves SQL commands to a plain-text file.

Known Bugs and Limitations

1. **Missing executed SQL commands in plain text file**: Not all SQL commands get recorded into the file.
2. **Plain text file**: non-encrypted file which poses a security vulnerability.
3. **No shopping cart**: No ability for the user to dynamically shop.
4. **No user account/authentication**: No ability for the user to make purchases or have a private purchase history.

Developer Retrospective

What worked well?

- **Basic Structure**: straightforward to implement entity relationships
- **Insertion of Data**: Mock data population was seamless and integrated well with the rest of the database.
- **Aggregate, group, order, join, calc and conc SQL queries**: Almost all the SQL queries worked as expected.

What didn't work well?

- **Limited Flexibility**: Pre-coded SQL commands reduce user flexibility.
- **Security Features**: No encryption or user authentication resulting in database having a major security vulnerability.
- **Bugs**: A lot of bugs present in software

What to do differently?

- **More Flexibility**: Allow user to manipulate the SQL commands prior to execution, in a user-friendly manner.
- **Add security measures**: store queries in encrypted file, enable encryption and user authentication.

- **Perform more research and study**: Complete more research and improve understanding of programming and SQL skills to produce a better product in a similar timespan.
- **Be more organised**: Improve labels for classes and definitions and include more detailed comments to make program more readable and allow user to understand how it works.

Bibliography

1.

Dy L. Do Agencies Need an SSL Secure Shopping Certificate for Their Client's Websites [Internet]. Vendasta Blog. 2023 [cited 2024 Aug 7]. Available from: https://www.vendasta.com/blog/ssl-secure-shopping-client-websites/#:~:text=SSL%20secure%20shopping%20refers%20to

2.

Chapter 14 Australian Privacy Principle 11-security of Personal Information [Internet]. Parliament of Australia. 2014 Mar [cited 2024 Aug 7]. Available from: https://www.aph.gov.au/~/media/wopapub/senate/committee/fapa_ctte/completed_inquiries/2010-13/priv_exp_drafts/report_part1/c14.ashx#:~:text=14.1%20Australian%20Privacy%20Principle%2011

3.

OpenAI. ChatGPT [Internet]. ChatGPT. 2024. Available from: https://chatgpt.com/

4.

w3schools. SQL Tutorial [Internet]. W3schools.com. 2019 [cited 2024 Aug 25]. Available from: https://www.w3schools.com/sql/

5.

Stack Overflow. Stack Overflow - Where Developers Learn, Share, & Build Careers [Internet]. Stack Overflow. 2024 [cited 2024 Aug 25]. Available from: https://stackoverflow.com/