



TRƯỜNG ĐẠI HỌC FPT



ASSIGNMENT FRONT SHEET <No.1>

Qualification	BTEC Level 5 HND Diploma in Computing and Systems Development		
Unit number and title	Unit 04: Database Design & Development		
Assignment due		Assignment submitted	2
Learner's name	Tran Quang Huy	Assessor name	
Learner's ID	GCD18457	Submission number	

**Learner declaration:**

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Learner signature	Huy	Date	

**Grading grid**

P2	P3	P4	P5	M2	M3	M4	M5	D2	D3

Assignment title	Understand databases and data management systems
------------------	--



TRƯỜNG ĐẠI HỌC FPT



In this assignment, you will have opportunities to provide evidence against the following criteria.

Indicate the page numbers where the evidence can be found.

Assessment criteria	Expected evidence	Task no.	Assessor's Feedback
<b>LO2</b> Develop a fully functional relational database system, based on an existing system design. <b>LO3</b> Test the system against user and system requirements. <b>LO4</b> Produce technical and user documentation			
<b>P2</b> Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables.	<ul style="list-style-type: none"><li>- Code snippets to create each table &amp; to insert some sample data for each table</li><li>- Generated Database Diagram</li><li>- Explanations about any changes comparing to your design</li><li>-Use the range of tools and techniques used to design (working or prototyping) user interfaces for these above system's functionalities</li></ul>		

<b>P3</b> Implement a query language into the relational database system.	<ul style="list-style-type: none"> <li>- Queries to support the functionalities (insert, update, delete)</li> <li>- Queries to support the enhancement of the interface to the above functionalities</li> </ul>		
<b>P4</b> Test the system against user and system requirements.	Produce a test plan and execute it. You must include: <ul style="list-style-type: none"> <li>• Normal scenario</li> <li>• Data validation</li> <li>• Extremes of data</li> <li>• Query execution (the queries to execute the test cases)</li> </ul>		
<b>P5</b> Produce technical and user documentation.	<ul style="list-style-type: none"> <li>- Technical documentation</li> <li>- User documentation</li> </ul>		
<b>M2</b> Implement a fully functional database system which includes system security and database maintenance.			

<b>M3</b> Assess whether meaningful data has been extracted through the use of query tools to produce appropriate management information.			
<b>M4</b> Assess the effectiveness of the testing, including an explanation of the choice of test data used.			
<b>M5</b> Produce technical and user documentation for a fully functional system, including ER Diagram and normalization statements and describing how the system works.			
<b>D2</b> Evaluate the effectiveness of the database solution in relation to user and system requirements, and suggest improvements.			



TRƯỜNG ĐẠI HỌC FPT



<b>D3</b> Assess any future improvements that may be required to ensure the continued effectiveness of the database system.			
---	--	--	--



TRƯỜNG ĐẠI HỌC FPT



Summative feedback

Assessor's Signature

Date



# Table of Contents

<b>INTRODUCTION .....</b>	<b>1</b>
<b>PART 1: DATABASE DESIGN .....</b>	<b>2</b>
1. Database system architecture .....	2
2. Entity Relationship Diagram (ER Diagram):.....	5
3. Database creation tables.....	6
4. Diagram .....	15
<b>PART 2: INTERFACE DESIGN, QUERY, TEST CASE .....</b>	<b>16</b>
1. Login.....	16
2. Register.....	18
3. User main menu .....	20
4. Salesperson main menu.....	22
5. Management main menu. ....	24
6. Function “Invoice” .....	26
7. Function “Customer” .....	28
8. Function “Supplier” .....	32
9. Function “Salespersons” .....	36
10. Function “Items” .....	40
11. Function customer report .....	44
12. Function Sales report .....	47
13. Function Items report .....	50
<b>CONCLUSION .....</b>	<b>53</b>
1. How Interface Design and Database Design assessments meet user requirements.....	53
2. Advantages and Disadvantages .....	57
3. Remedial solutions.....	59
<b>APPENDIX: Survey Form .....</b>	<b>61</b>
<b>References .....</b>	<b>62</b>

# TABLE OF TABLES

Table 1- The third normal form (3NF).....	3
Table 2 - Fully Database system normal form .....	4
Table 3 - Customer table.....	6
Table 4 - Example Customer table .....	7
Table 5 - Salespersons table .....	7
Table 6 - Example Salespersons table.....	8
Table 7 - Supplier table .....	9
Table 8 - Example Supplier table.....	9
Table 9 - Items table .....	10
Table 10 - Example Items table.....	10
Table 11 - Orders table .....	11
Table 12 - Example Orders table.....	12
Table 13 - Order detail table .....	13
Table 14 - Example Order detail table .....	13
Table 15 - Example Customer Login Table.....	14
Table 16 - Customer Login Table .....	14
Table 17 - Customer Login Table matching ID and password.....	17
Table 18 - Test case function login .....	17
Table 19 - Result of query for register .....	19
Table 20 - Test case of Register .....	19
Table 21 - Show information about customer is using .....	21
Table 22 - Test case User main menu .....	21
Table 23 - Test case salespersons main menu.....	23
Table 24 - Test case Management main menu.....	25
Table 25 - Create Invoice table .....	27
Table 26 - Test case function Invoice.....	27
Table 27 - Result of function Search customer.....	30
Table 28 - Result of function update .....	30
Table 29 - Test case function customers .....	31
Table 30 - Result of function Search supplier .....	34
Table 31 - Test case function supplier .....	35
Table 32 - Test case function salesperson .....	39
Table 33 - Test case function Items .....	43
Table 34 - Result of function loyal customer in customer report.....	45
Table 35 - Result of function Customers buy the most .....	45
Table 36 - Test case function customer report.....	46



Table 37 - Result of function bonuses for employees .....	48
Table 38 - Result of function sales employees .....	48
Table 39 - Result of function sales report.....	49
Table 40 - Test case function sales report .....	49
Table 41 - Result of function List of items and price .....	51
Table 42 - Result of function Min, Max price.....	51
Table 43 - Test case function items report .....	52
Table 44 - Evaluate the responsiveness of Database with user requirements.....	54
Table 45 - Interface Design and Database Design assessments user requirements .....	56

## TABLE OF FIGURES

Figure 1 - Purchase invoice ElectroShop.....	2
Figure 2 - ElctroShop Entity Relationship Diagram .....	5
Figure 3 - Database System Diagram .....	15
Figure 4 - Interface login .....	16
Figure 5 - Interface register .....	18
Figure 6 - Interface User main menu .....	20
Figure 7 - Interface salesperson main menu .....	22
Figure 8 - Interface Management main menu.....	24
Figure 9 - Interface function Invoice.....	26
Figure 10 - Interface function Customers .....	28
Figure 11 - Interface function Customer - Add New.....	29
Figure 12 – Interface function Supplier .....	32
Figure 13 - Interface function Supplier - Add New .....	33
Figure 14 - Interface function Salespersons .....	36
Figure 15 - Interface function Salespersons - Add New .....	37
Figure 16 - Interface function Items .....	40
Figure 17 - Interface function Items - Add New .....	41
Figure 18 - Interface Function Customer Report.....	44
Figure 19 - Interface function Sales report.....	47
Figure 20 - Interface function Items report .....	50
Figure 21 - Evalute the responsiveness of Database with user requirements in 3 days .....	53

# INTRODUCTION

Use an appropriate design tool to design a relational database system for substantial problem. Base on Assignment Brief that ElectroShop company sell electronic devices and they want to create a database to store system.

This system will store the store's necessary data, which are: Customer Information, Seller (Employee) Information, Product & Provider Information, Invoice Information

Designed by using Relational Database and 3NF Normalization Process, this System will provide a fully functional Database which includes: Data validations, Data normalization Data interfaces for different situations, Data modification, Calculate Seller's (Employee) bonus

Requirement of ElectroShop' application:

- The database will store the information and security data b. This system will store the store's necessary data, which are:
  - Seller information
  - Customer information
  - Invoice
  - Product information
  - Supplier information
- Designed by using Relational Database and 3NF Normalization Process, this System will provide a fully functional Database which includes:
  - Data validations
  - Data normalization
  - Data modification
  - Calculate Seller's (Employee) bonus

# PART 1: DATABASE DESIGN


## 1. Database system architecture

The Database system's requirement additional steps should be taken so that no attribute can be transitively dependent on the primary key. That means they cannot be dependent on a non-primary key attribute in the same table.

Upon inspection, a name of supplier is actually retrievable from supplier information and similarly to item information, customer information and salesperson information are dependent on item\_id, customer\_id, salesperson\_id.

One customer can buy many items and one item can be bought by many customers so the Database system need to have 1 table order\_detail include item\_id, order\_id and order\_quantity to solve that problem.

In the beginning, ElectroShop has a small database system and is limited to paperwork by getting information from this Invoice:



### PURCHASE INVOICE

Date: \_\_\_\_\_

InvoiceID: \_\_\_\_\_

CustomerID: \_\_\_\_\_ SellerID: \_\_\_\_\_

Customer name: \_\_\_\_\_

Customer address: \_\_\_\_\_

Customer phone number: \_\_\_\_\_

Customer E-mail: \_\_\_\_\_

STT	Product name	Supplier name	ProductID	Calculation Unit	Sold quantity	Price	Total
A	B	C	D	E	1	2	3
1							
2							
3							
4							
5							
In total (by words):							

Seller \_\_\_\_\_ Customer \_\_\_\_\_

Figure 1 - Purchase invoice ElectroShop

3NF					
<b>order_id</b> order_day customer_id salesperson_id	<b>Order ID</b>	<b>Order Day</b>	<b>Customer ID</b>	<b>Salesperson ID</b>	
	1	16/01/2018	1	3	
	2	16/01/2018	2	4	
	3	15/03/2018	3	5	
	4	20/06/2018	4	6	
<b>item_id</b> <b>order_id</b> order_quantity	<b>Item ID</b>	<b>Order ID</b>	<b>Order Quantity</b>		
	1	2	2		
	2	3	2		
	3	4	3		
	4	5	4		
<b>item_id</b> item_kind item_price	<b>Item ID</b>	<b>Kind of Item</b>	<b>Price</b>		
	1	Ipad	300		
	2	Iphone	400		
	3	Television	300		
	4	Laptop	500		
<b>supplier_id</b> supplier_name supplier_mail supplier_adress supplier_phone	<b>Supplier ID</b>	<b>Supplier Name</b>	<b>Supplier mail</b>	<b>Supplier Address</b>	<b>Supplier Phone</b>
	1	HCT	qwe@gmail.com	23 HCM	1234578
	2	Zero9	rty@gmail.com	22 Hue	45786541
	3	ST MTP	yui@gmail.com	54 HN	12458212
	4	XV	iop@gmail.com	87 Dong Ha	2134578
<b>customer_id</b> customer_name customer_mail customer_adress customer_zipcode customer_phone	<b>Customer ID</b>	<b>Customer Name</b>	<b>Customer Mail</b>	<b>Customer Phone</b>	
	1	Trần Quang Huy	qweas@gmail.com	1231234578	
	2	Nguyễn Quang Ngọc	rtyasd@gmail.com	45345786541	
	3	Nguyễn Quang Thắng	yuxzxi@gmail.com	1244258212	
	4	Phạm Nhật Vượng	iopdas@gmail.com	21423434578	
<b>salesperson_id</b> salesperson_name salesperson_age salesperson_mail salesperson_phone salesperson_adress	<b>Salesperson ID</b>	<b>Salesperson Name</b>	<b>Salesperson Mail</b>	<b>Salesperson Phone</b>	
	1	Nguyễn Hà My	qweas1@gmail.com	31234578	
	2	Nguyễn Công Phượng	rtyasd2@gmail.com	453456541	
	3	Trương Văn Hiếu	yuxzx3i@gmail.com	44258212	
	4	Nguyễn Anh Tuấn	iopda4s@gmail.com	214234578	
<b>login_id</b> login_name login_passoword fk_customer_id	<b>Login ID</b>	<b>Login Name</b>	<b>Login Password</b>		
	1	supermido	haha123		
	2	shogunminh	123hahah		
	3	kaze04	duongminh		

Table 1- The third normal form (3NF)

Conclusively, the Database system will have:

UNF	1NF	2NF	3NF
customer_name customer_mail customer_address customer_zipcode customer_phone salesperson_name salesperson_age salesperson_mail salesperson_phone salesperson_adress order_day order_quantity item_kind item_price supplier_name supplier_mail supplier_adress supplier_phone	<b>order_id</b> item_kind item_price supplier_name supplier_mail supplier_adress supplier_phone	<b>item_id</b> item_name item_price supplier_name supplier_mail supplier_adress supplier_phone	<b>order_id</b> order_day customer_id salesperson_id
	<b>order_id</b> order_day order_quantity customer_name customer_mail customer_address customer_zipcode customer_phone salesperson_name salesperson_age salesperson_mail salesperson_phone salesperson_adress	<b>order_id</b> <b>item_id</b> item_name item_price	<b>item_id</b> <b>order_id</b> order_quantity
		<b>order_id</b> order_day order_quantity customer_name customer_mail customer_address customer_zipcode customer_phone salesperson_name salesperson_age salesperson_mail salesperson_phone salesperson_adress	<b>item_id</b> item_kind item_price
			<b>supplier_id</b> supplier_name supplier_mail supplier_adress supplier_phone
			<b>customer_id</b> customer_name customer_mail customer_adress customer_zipcode customer_phone
			<b>salesperson_id</b> salesperson_name salesperson_age salesperson_mail salesperson_phone salesperson_adress
			<b>login_id</b> login_name login_passoword fk_customer_id

Table 2 - Fully Database system normal form

## 2. Entity Relationship Diagram (ER Diagram):

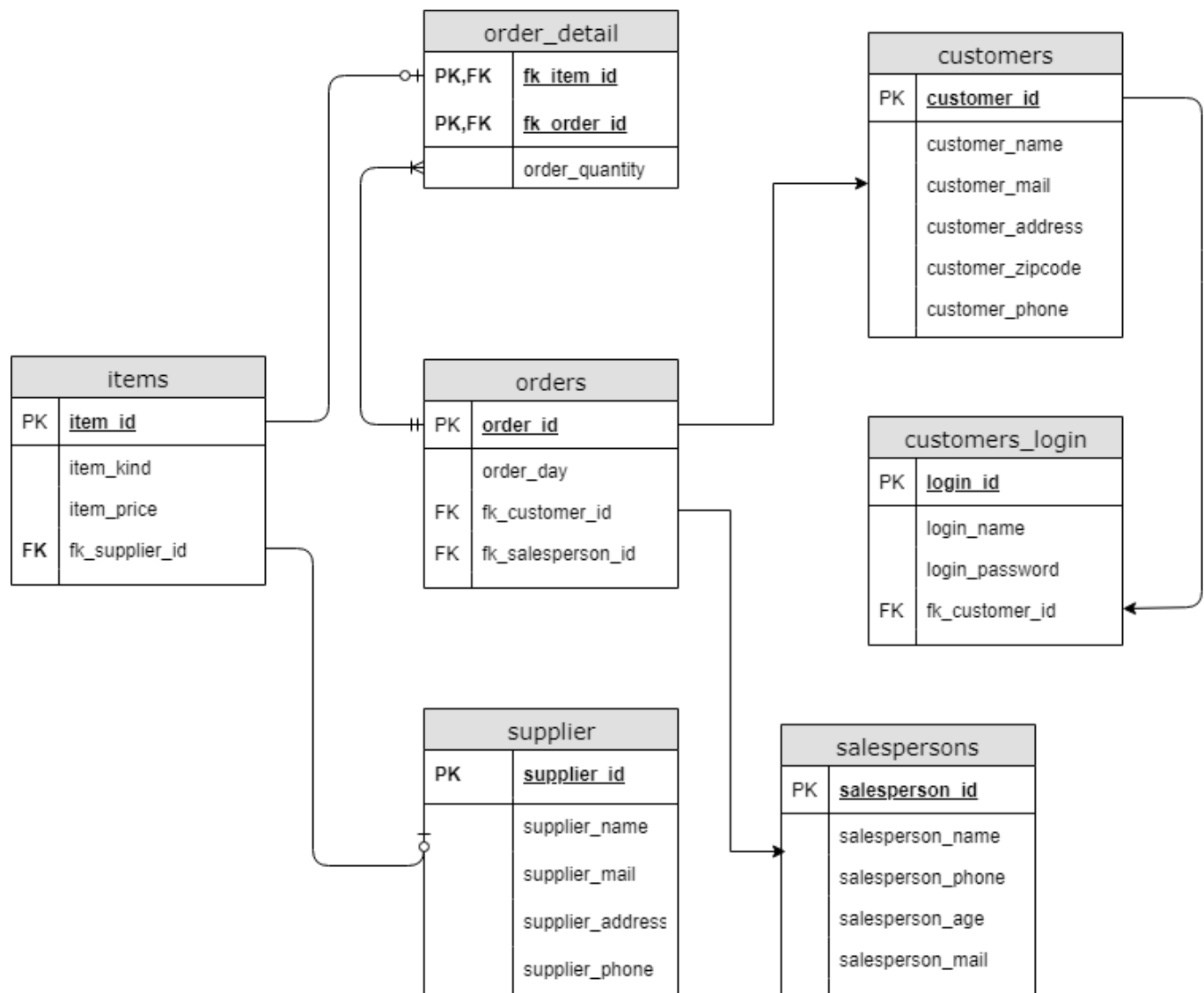


Figure 2 - ElctroShop Entity Relationship Diagram

### 3. Database creation tables

#### 3.1. Customer table:

This table used to store information of ElectroShop's customer information, customer's email will be used to send offers, product or promotions.

```
CREATE TABLE customers(  
    customer_id INT PRIMARY KEY,  
    customer_name VARCHAR(100),  
    customer_mail VARCHAR(100),  
    customer_address VARCHAR(100),  
    customer_zipcode VARCHAR(100),  
    customer_phone VARCHAR(100),  
);
```

customers *			
	Column Name	Data Type	Allow Nulls
🔑	customer_id	int	⌏
	customer_name	varchar(100)	⌏
	customer_mail	varchar(100)	☑
	customer_address	varchar(100)	☑
	customer_zipcode	varchar(100)	☑
	customer_phone	varchar(100)	☑
			⌏

Table 3 - Customer table

This table includes:

- Customer's identity (**customer\_id** - PK): This is primary key of customer table to make sure that a customer has a unique and non-duplicate ID.
- Customer's Name (**customer\_name**): Store customer name information.
- Customer's Email (**customer\_mail**): Store customer mail information and send product information as well as the completion process when customers buy products.
- Customer's Address (**customer\_address**): Store customer address information.
- Customer's Zip code (**customer\_zipcode**): Store customer zip code information.
- Customer's Phone number (**customer\_phone**): Easily contact customers when needed.

	customer_id	customer_name	customer_mail	customer_address	customer_zipcode	customer_phone
1	1	Allen Ayers	posuere.cubilia.Curae@omare.edu	294-4580 At Rd.	IC5 8NW	0819 171 8803
2	2	Dolan Lowery	sociis@nuncac.net	728 Pede, Avenue	UK7R 9NE	(0113) 072 1649
3	3	Aaron Martinez	lobortis.tellus@sempercursus.net	Ap #577-6187 Ultrices Ave	WV3 7QG	0827 814 4921
4	4	Gannon Potter	fringilla.euismod@magnaCrasconvallis.co.uk	P.O. Box 215, 4902 Nullam Av.	LD7 0JD	0845 680 8907
5	5	Stuart McLaughlin	amet@maurissit.com	3248 Imperdiet Rd.	X6S 7JI	056 7531 6531
6	6	Elliott Cunningham	vel@arcuacorci.edu	928-9736 Semper Rd.	GW29 0YT	(014962) 24309
7	7	Clayton Compton	montes@tellus.edu	P.O. Box 208, 6020 Tortor, Rd.	A82 7XW	0845 46 48
8	8	Moses Pearson	vel.arcu.eu@augueacipsum.com	Ap #289-6765 Id St.	OO0J 3XU	07624 737901

Table 4 - Example Customer table

### 3.2.Salespersons table:

This table is used to store information of ElectroShop's employee information would help to determine which salesperson is in charge of selling which order. Many orders can be sold by one salesperson so it will have many relationships.

```
CREATE TABLE salespersons(
  salesperson_id INT PRIMARY KEY,
  salesperson_name VARCHAR(100),
  salesperson_phone VARCHAR(100),
  salesperson_age INT,
  salesperson_mail VARCHAR(100),
);
```

salespersons *			
	Column Name	Data Type	Allow Nulls
🔑	salesperson_id	int	<input type="checkbox"/>
	salesperson_name	varchar(100)	<input type="checkbox"/>
	salesperson_phone	varchar(100)	<input checked="" type="checkbox"/>
	salesperson_age	int	<input checked="" type="checkbox"/>
	salesperson_mail	varchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table 5 - Salespersons table



This table includes:

- Salesperson's identity (**salesperson\_id** - PK): Each salesperson just only has one ID that make sure that salesperson has a unique and non-duplicate ID so this must be primary key.
- Salesperson's name (**salesperson\_name**): Store salesperson name information.
- Salesperson's Age (**salesperson\_age**): Store salesperson age information.
- Salesperson's phone number (**salesperson\_phone**): Store salesperson phone number information.
- Salesperson's email (**salesperson\_mail**): Store salesperson email information.

	salesperson_id	salesperson_name	salesperson_phone	salesperson_age	salesperson_mail
1	1	Bert Washington	(016977) 5226	45	dapibus.gravida.Aliquam@nonenim.edu
2	2	Keane Rivas	07470 645211	52	velit.Aliquam.nisl@maurisd.ca
3	3	Elton Kaufman	0303 014 7073	45	non.nisi@risus.co.uk
4	4	Ferris Riggs	(01958) 800854	36	massa.Suspendisse@sedduiFusce.net
5	5	Julian Dominguez	(01089) 10436	51	ullamcorper.velit@Nulla.net
6	6	Flynn Randall	0800 1111	40	amet.ultrices@Nuncommodoactor.org
7	7	Orlando Frank	07624 853883	33	molestie.pharetra.nibh@aceleifendvitae.co.uk
8	8	Malik Pruitt	(018750) 41060	55	nunc.Quisque@omarelectus.ca

*Table 6 - Example Salespersons table*

### 3.3. Supplier table

This table is used to store information of ElectroShop's supplier.

```
CREATE TABLE supplier(  
    supplier_id INT PRIMARY KEY,  
    supplier_name VARCHAR(100),  
    supplier_mail VARCHAR(100),  
    supplier_adress VARCHAR(100),  
    supllier_phone VARCHAR(100),  
);
```

supplier			
	Column Name	Data Type	Allow Nulls
🔑	supplier_id	int	<input type="checkbox"/>
	supplier_name	varchar(100)	<input checked="" type="checkbox"/>
	supplier_mail	varchar(100)	<input checked="" type="checkbox"/>
	supplier_adress	varchar(100)	<input checked="" type="checkbox"/>
	supllier_phone	varchar(100)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table 7 - Supplier table

This table includes:

- Supplier's Identity (**supplier\_id** - PK): This is primary key of supplier table to make sure that a supplier has a unique and non-duplicate ID.
- Supplier's name (**supplier\_name**): Store supplier name information.
- Supplier's mail (**supplier\_mail**): Store supplier mail information.
- Supplier's address (**supplier\_address**): Store supplier address information.
- Supplier's phone number (**supplier\_phone**): Store supplier phone information.

	supplier_id	supplier_name	supplier_mail	supplier_adress	supplier_phone
1	1	Silas Brock	molestie.Sed.id@ullamcorpernislarcu.net	547-6530 Lacus, St.	0800 1111
2	2	Chadwick Daniels	ridiculus.mus.Proin@vulputate.net	P.O. Box 422, 8223 Amet Street	(017099) 57031
3	3	Aladdin Gould	ligula.Donec@Praesentinterdumligula.com	608-9434 Neque. St.	0868 860 6721
4	4	Amir Petersen	magna@iaculisquispede.org	625-6903 Aenean St.	055 5617 1458
5	5	Dominic Dale	nibh@atpedeCras.net	P.O. Box 651, 6906 A Rd.	0845 46 49
6	6	Xander Bright	adipiscing.fringilla.porttitor@Aliquam.co.uk	601-5875 Feugiat Av.	0800 1111
7	7	Mark Richard	blandit.at@odioEtiam.org	2264 Odio. St.	(01656) 945906
8	8	Nathaniel McDonald	a.dui@et.com	Ap #477-6616 Eu St.	(0113) 828 1200

Table 8 - Example Supplier table

### 3.4. Item table.

This table is used to store information of ElectroShop's orders. That also contains information about items that are being sold. The items are shown in the items table and in order details.

```
CREATE TABLE items(  
    item_id INT PRIMARY KEY,  
    item_kind VARCHAR(100),  
    item_price INT,  
    fk_supplier_id int,  
    CONSTRAINT fk_supplier_id_items  
        FOREIGN KEY (fk_supplier_id)  
        REFERENCES supplier (supplier_id),  
);
```

items			
	Column Name	Data Type	Allow Nulls
?	item_id	int	<input type="checkbox"/>
	item_kind	varchar(100)	<input checked="" type="checkbox"/>
	item_price	float	<input checked="" type="checkbox"/>
	fk_supplier_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table 9 - Items table

This table includes:

- Item identity (**item\_id** – PK): This is primary key of items table to make sure that item has a unique and non-duplicate ID.
- Kind of Items (**item\_kind**): Store kind of items information.
- Price (**item\_price**): Store price of items information.
- Supplier Identity (**fk\_supplier\_id**): This is foreign key to link with **supplier\_id** at supplier table.

This allow people can know what specific items originated.

	item_id	item_kind	item_price	fk_supplier_id
1	1	Laptop	8115	1
2	2	VGA	4202	2
3	3	Camera	7373	3
4	4	Laptop	3516	4
5	5	VGA	7119	5
6	6	CPU	2097	6
7	7	Mobliephone	1519	7
8	8	VGA	3389	8
9	9	Printer	1785	9

### 3.5. Order table

This table is used to store information of ElectroShop's orders. It is an important table to connect and reference by many others: A customer who order that items, a salesperson who sold that items, specially that connect with that

```
CREATE TABLE orders(  
  order_id INT PRIMARY KEY,  
  order_day date,  
  fk_customer_id int,  
  CONSTRAINT fk_customer_id  
    FOREIGN KEY (fk_customer_id)  
      REFERENCES customers (customer_id),  
  order_salesperson_id INT,  
  CONSTRAINT fk_order_salesperson_id  
    FOREIGN KEY (order_salesperson_id)  
      REFERENCES salespersons (salesperson_id),  
);
```

orders			
	Column Name	Data Type	Allow Nulls
🔑	order_id	int	<input type="checkbox"/>
	order_day	date	<input checked="" type="checkbox"/>
	fk_customer_id	int	<input checked="" type="checkbox"/>
	order_salesperson_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table 11 - Orders table

Order detail table as know as invoice.

This orders table includes:

- Order's identity (**order\_id** – PK): This is primary key to confirm that when customer order something this order's identity exist only and non-duplicate ID.
- Order day (**order\_day**): Store the day that customer order information.
- Customer's identity (**fk\_customer\_id** - FK): This is foreign key references to **customer table**. It will allow people know information about customer is ordering.
- Salesperson identity (**fk\_salesperson\_id** - FK): This is foreign key references to **supplier table**. That indicates employee who sell items to customer.

	order_id	order_day	fk_customer_id	fk_salesperson_id
1	1	2019-09-17	1	1
2	2	2019-05-20	2	2
3	3	2019-12-19	3	3
4	4	2019-07-01	4	4
5	5	2019-01-20	5	5
6	6	2019-11-21	6	6
7	7	2018-09-17	7	7
8	8	2018-03-30	8	8

*Table 12 - Example Orders table*

### 3.6. Order detail table

This order detail table know as invoice and this dataset stores all the invoices that are made through sale. One order will generate one invoice.

```
CREATE TABLE order_detail(  
    fk_item_id INT,  
    CONSTRAINT fk_item_id  
        FOREIGN KEY (fk_item_id)  
        REFERENCES items (item_id),  
  
    fk_order_id int,  
    CONSTRAINT fk_order_id  
        FOREIGN KEY (fk_order_id)  
        REFERENCES orders (order_id),  
    PRIMARY KEY (fk_item_id, fk_order_id),  
    order_quantity INT,  
);
```

order detail			
	Column Name	Data Type	Allow Nulls
🔑	fk_item_id	int	<input type="checkbox"/>
🔑	fk_order_id	int	<input type="checkbox"/>
	order_quantity	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table 13 - Order detail table

Order detail table includes:

- Item's identity (**fk\_item\_id** -PK, FK): This is one of two primary key with order identity that allow customer can offer more than one item. And it also foreign key references with **item\_id** in **items table** to show what items customer order.
- Order's Identity (**fk\_order\_id** – PK, FK): this is a last primary key with item identity that allow person can offer more than one item and foreign key references with **order\_id** in **orders table** this is the main of function let customer can offer more than one item.
- Order quantity (**order\_quantity**): Store a number of each items that customer order information.

	fk_item_id	fk_order_id	order_quantity
1	1	1	8
2	2	2	4
3	3	3	8
4	4	4	9
5	5	5	10
6	6	6	8
7	7	7	10
8	8	8	6

Table 14 - Example Order detail table

### 3.7. Customer login table

This table used to store information about login id and login password of each customer. Base on that customer can log in to the ElectroShop system.

```
CREATE TABLE customer_login(  
    login_id INT PRIMARY KEY,  
    login_name VARCHAR(100),  
    login_passowrd VARCHAR(100),  
    fk_customer_id INT,  
    CONSTRAINT fk_customer_id_login  
        FOREIGN KEY (fk_customer_id)  
        REFERENCES customers (customer_id),  
);
```

#### customer login

	Column Name	Data Type	Allow Nulls
🔑	login_id	int	<input type="checkbox"/>
	login_name	varchar(100)	<input checked="" type="checkbox"/>
	login_passowrd	varchar(100)	<input checked="" type="checkbox"/>
	fk_customer_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table 16 - Customer Login Table

	login_id	login_name	login_password	fk_customer_id
1	1	egestas.	felis,	1
2	2	tincidunt,	Quisque	2
3	3	Sed	commodo	3
4	4	venenatis	scelerisque	4
5	5	nonummy	facilisis	5

Table 15 - Example Customer Login Table

## 4. Diagram

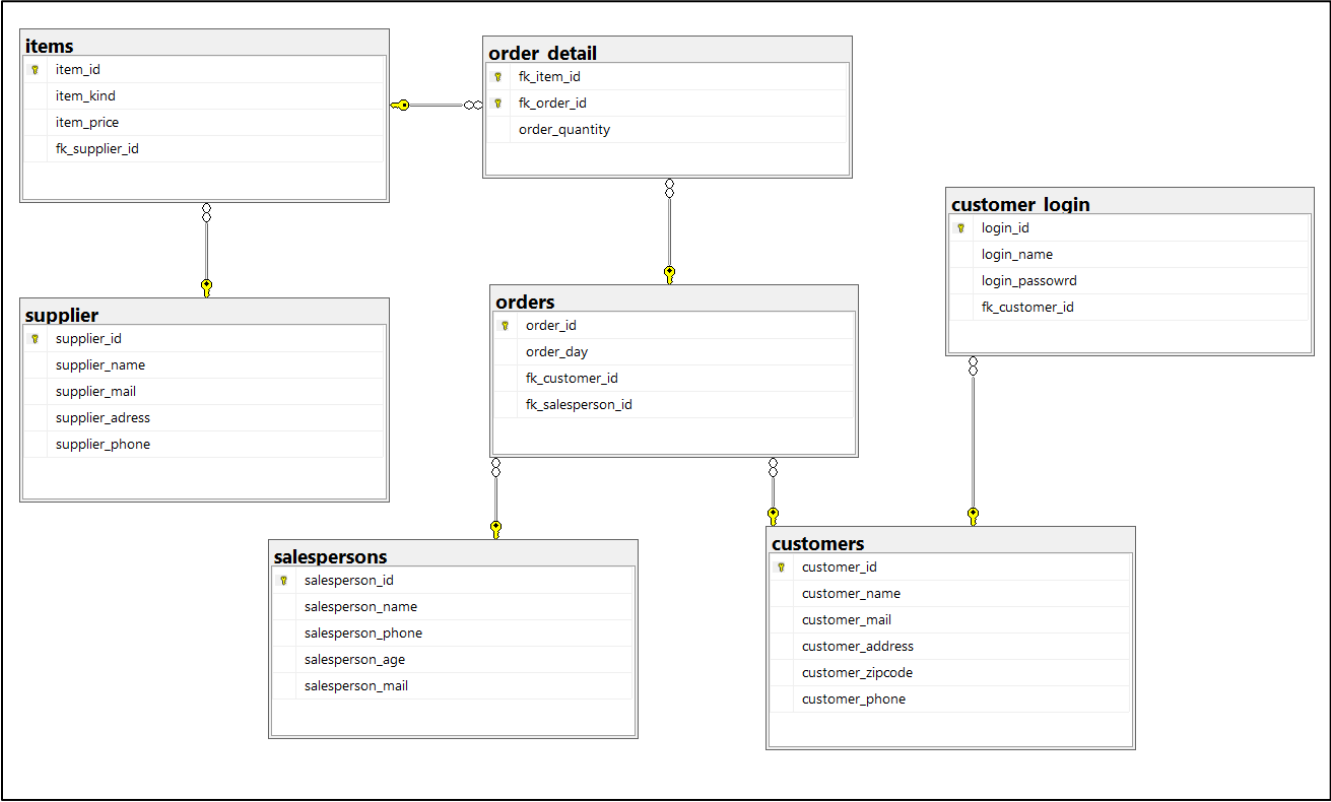


Figure 3 - Database System Diagram



## PART 2: INTERFACE DESIGN, QUERY, TEST CASE

### 1. Login

#### 1.1. Interface design

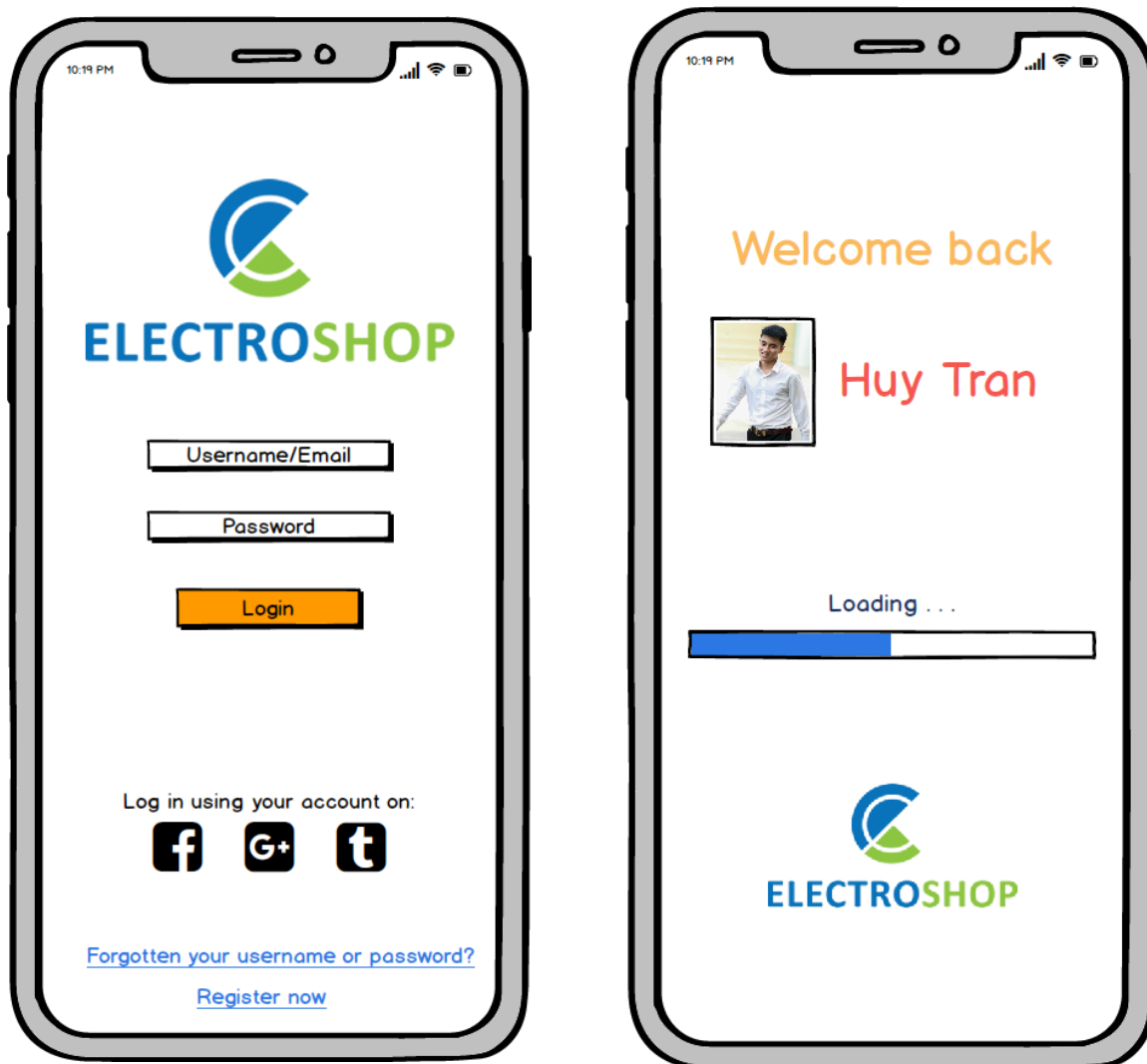


Figure 4 - Interface login

## 1.2. Query

```
SELECT customers.customer_id AS Customer_ID, customers.customer_name AS
Customer_name
FROM customers, customer_login
WHERE (customer_login.login_name = 'tincidunt,')
      AND (customer_login.login_password='Quisque')
      AND (customer_login.fk_customer_id=customers.customer_id);
```

	login_id	login_name	login_password	fk_customer_id
1	1	egestas.	felis,	1
2	2	tincidunt,	Quisque	2
3	3	Sed	commodo	3
4	4	venenatis	scelerisque	4
5	5	nonummy	facilisis	5
6	6	turpis	a,	6
7	7	posuere	Morbi	7
8	8	nulla.	elementum	8
9	9	libero	vitae	9
10	10	amet	est	10
11	11	In	leo	11

	Customer_ID	Customer_name
1	2	Dolan Lowery

Customer_ID	Customer_name
-------------	---------------

Table 17 - Customer Login Table matching ID and password

## 1.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Enter valid ID and incorrect Password and hit login button	ID: tincidunt Password: Quiqueeeeeeee	The ID or password that user have entered is incorrect.	The password that user have entered is incorrect.	Fail
2	Enter valid ID and Password but ID have some capital letter	ID: tiNciDuNt Password: Quisque	ID and Password are corrected move to Main Application	The ID that user have entered is incorrect.	Fail

Table 18 - Test case function login

## 2. Register

### 2.1. Interface design



Figure 5 - Interface register

## 2.2. Query

```
INSERT INTO
customers([customer_id],[customer_name],[customer_mail],[customer_address],[customer_zipcode],
[customer_phone])
VALUES(1,'Allen Ayers','posuere.cubilia.Curae@ornare.edu','294-4580 At Rd.','IC5 8NW','0819
171 8803')
```

	customer_id	customer_name	customer_mail	customer_address	customer_zipcode	customer_phone
1	1	Allen Ayers	posuere.cubilia.Curae@ornare.edu	294-4580 At Rd.	IC5 8NW	0819 171 8803
2	2	Dolan Lowery	sociis@nuncac.net	728 Pede, Avenue	UK7R 9NE	(0113) 072 1649
3	3	Aaron Martinez	lobortis.tellus@sempercursus.net	Ap #577-6187 Ultrices Ave	WV3 7QG	0827 814 4921
4	4	Gannon Potter	fringilla.euismod@magnaCrasconvallis.co.uk	P.O. Box 215, 4902 Nullam Av.	LD7 0JD	0845 680 8907
5	5	Stuart McLaughlin	amet@maurissit.com	3248 Imperdiet Rd.	X6S 7JI	056 7531 6531
6	6	Elliott Cunningham	vel@arcuacorci.edu	928-9736 Semper Rd.	GW29 0YT	(014962) 24309
7	7	Clayton Compton	montes@tellus.edu	P.O. Box 208, 6020 Tortor, Rd.	A82 7XW	0845 46 48
8	8	Moses Pearson	vel.arcu.eu@augueacipsum.com	Ap #289-6765 Id St.	OO0J 3XU	07624 737901

Table 19 - Result of query for register

## 2.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Enter full of requirement of register but let "Name" empty.	Name: Mail: <a href="mailto:abc@gmail.com">abc@gmail.com</a> Address: 12 BVC Zip Code: 8777 Phone: 098765421	Retry again and fill in Name	The NULL at Name is acceptable in database.	Fail
2	Enter ID and Password with special characters	ID: tincidunt@\$#! Password: Quisque@#	ID is illegal and Password is acceptable	ID and Password are acceptable	Fail

Table 20 - Test case of Register

### 3. User main menu

#### 3.1. Interface design

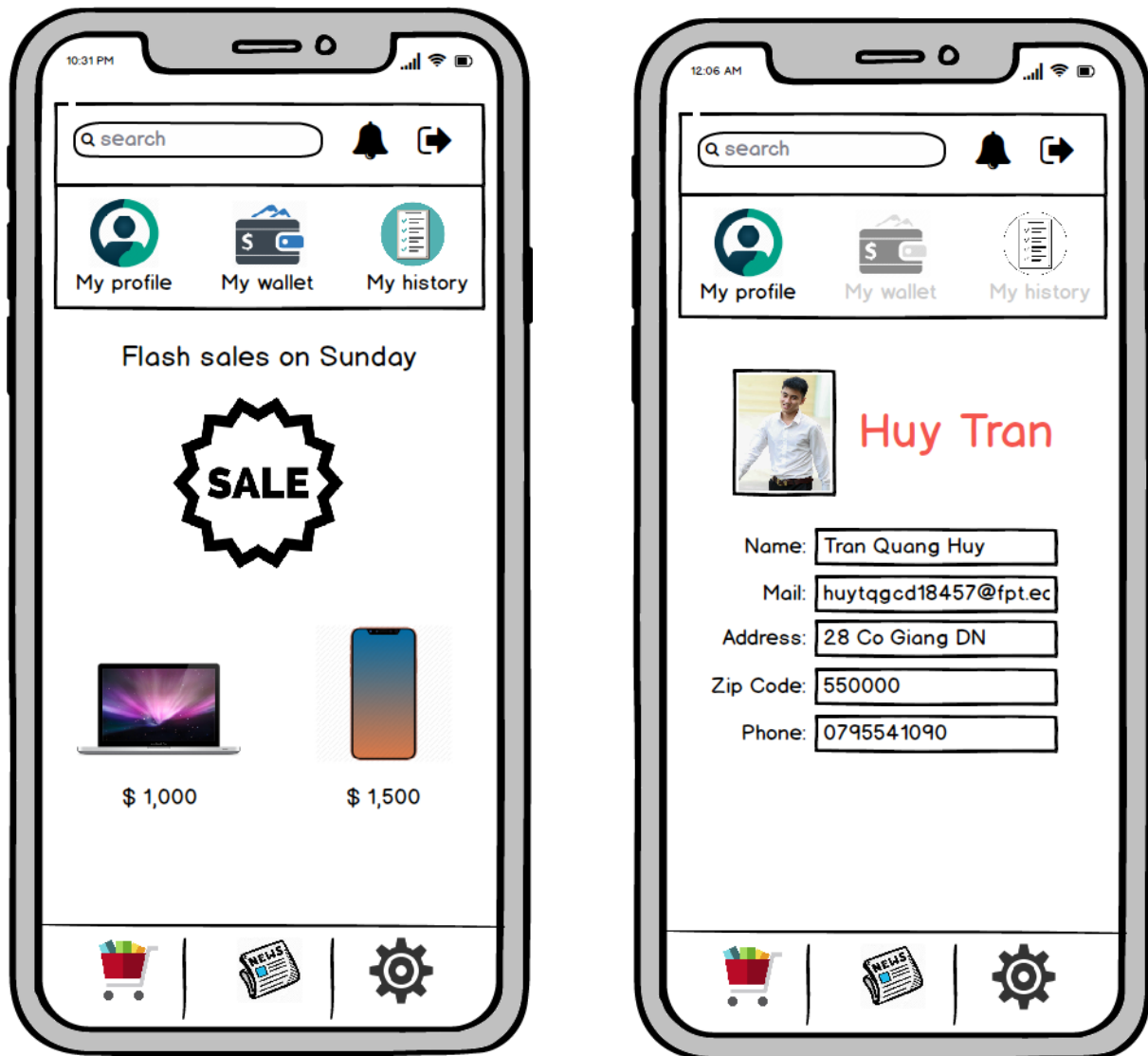


Figure 6 - Interface User main menu

### 3.2. Query

```
SELECT * FROM customers, customer_login
WHERE (customer_login.login_id = 5)
      AND (customers.customer_id = customer_login.fk_customer_id);
```

customer_id	customer_name	customer_mail	customer_address	customer_zipcode	customer_phone	login_id	login_name	login_password	fk_customer_id
5	Stuart McLaughlin	amet@maurissit.com	3248 Imperdiet Rd.	X6S 7JI	056 7531 6531	5	nonummy	facilisis	5

Table 21 - Show information about customer is using

### 3.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Check customer is using information	Select function “My profile” in main menu	Show all information about customer is using.	Show all information about customer	Pass
2	Update customer is using information after checked.	Phone: 123456712	Customer's phone number is updated	Customer's phone number remains the same	Fail

Table 22 - Test case User main menu

## 4. Salesperson main menu.

### 4.1. Interface design

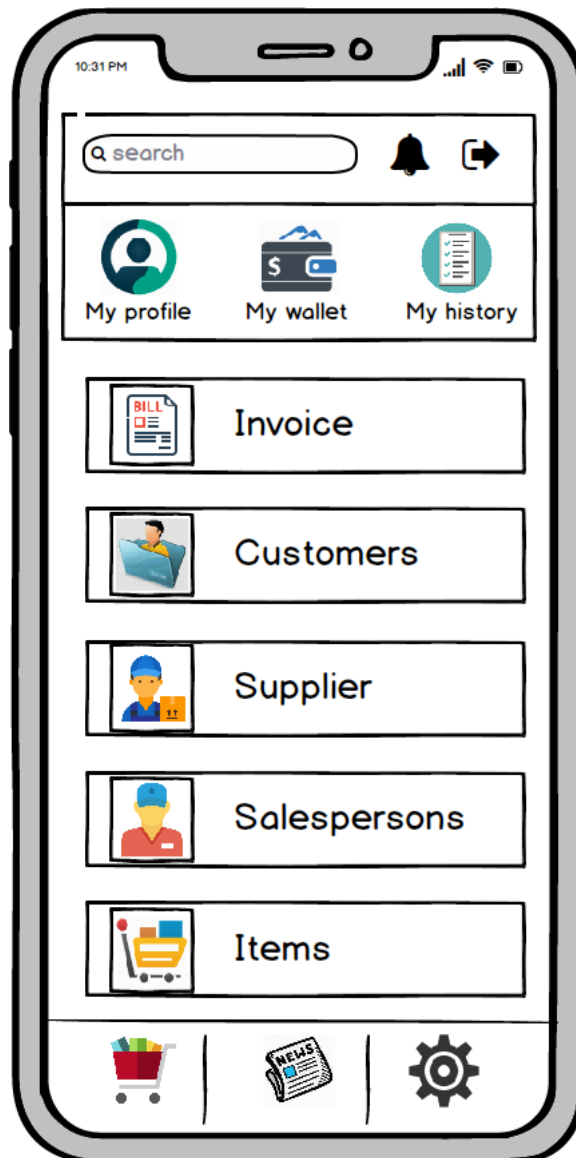


Figure 7 - Interface salesperson main menu

### 4.2. Query

```
SELECT * FROM customers;  
SELECT * FROM supplier;  
SELECT * FROM salespersons;  
SELECT * FROM items;  
SELECT * FROM orders, order_detail;
```

#### 4.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	List all information of Customer	Select function “Customers”	Show all information about customer	Show all information about customer	Pass
2	List all Supplier and Salesperson in same time.	Select function “Supplier” and “Salespersons”	Show all information about Supplier and Salespersons	Just only show all information about Supplier or Salesperson	Fail

*Table 23 - Test case salespersons main menu*



## 5. Management main menu.

### 5.1. Interface design

This interface only used by manager and manager have full function.

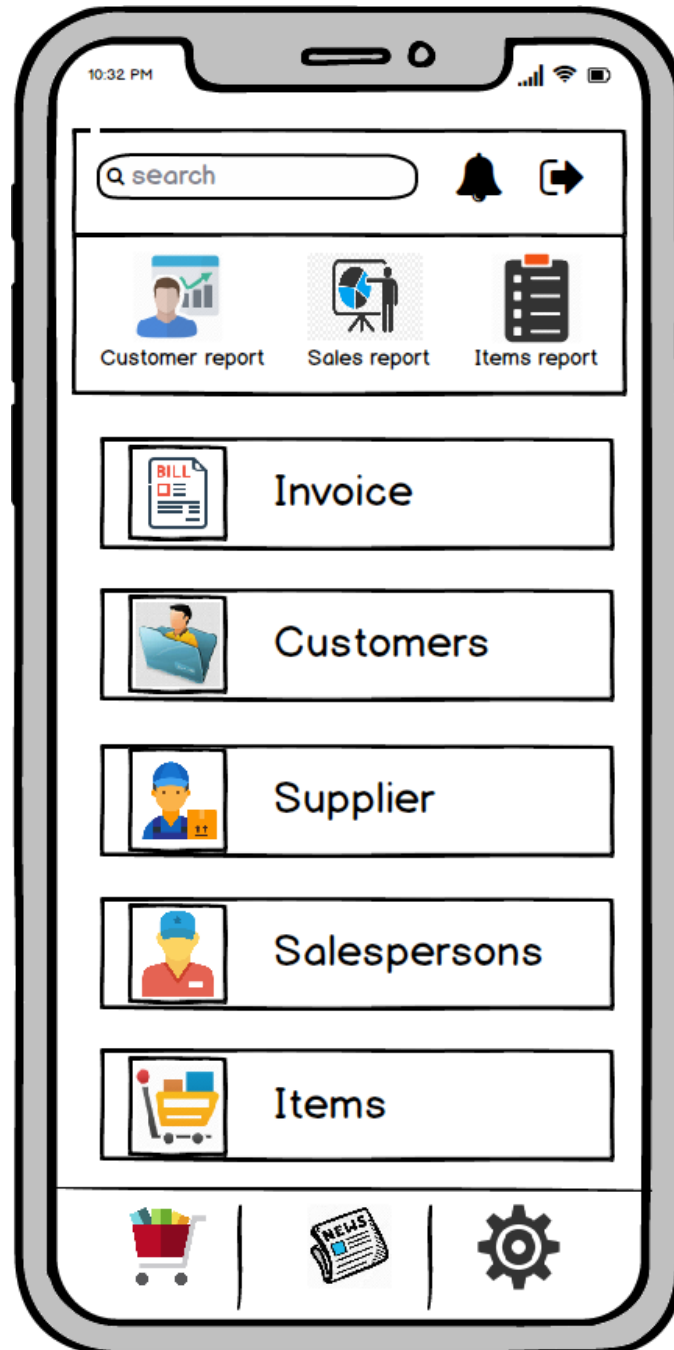


Figure 8 - Interface Management main menu

## 5.2. Query

```
SELECT * FROM customers;  
SELECT * FROM supplier;  
SELECT * FROM salespersons;  
SELECT * FROM items;  
SELECT * FROM orders, order_detail;
```

## 5.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	List all information of Customer	Select function “Customers”	Show all information about customer	Show all information about customer	Pass
2	List all Supplier and Salesperson in same time.	Select function “Supplier” and “Salespersons” in same time	Show all information about Supplier and Salespersons	Just only show all information about Supplier or Salesperson	Fail

Table 24 - Test case Management main menu

## 6. Function “Invoice”

### 6.1. Interface design

This interface is used by Salesperson and Manager to create a Invoice.

ID	Product	Kind	Quantity	Price
03	Chip i7-8700K	CPU	1	\$450
04	Giga 1080Ti	VGA	2	\$900
05	AW2518HF	LCD	1	\$550
06	Z370	Main CPU	1	\$250
07	G502	Gears	2	\$55
08	G633	Gears	2	\$120
09	G512	Gears	2	\$135

Mobilephone

- Tivi
- CPU
- VGA
- LCD
- Main CPU
- Gears

Items: Catalogue

Items

Select

Quantity: 3

Salesperson: Huyen My

Confirm

iPhone XS Max

- Tivi 42inch, Tivi
- Chip i7-8700K
- Giga 1080Ti
- AW2518HF
- Z370, Main CPU
- G502, Gears
- G633, Gears
- G512

Huyen My

- Ngoc Han
- Phuong
- Ngoc Trinh

Figure 9 - Interface function Invoice

## 6.2. Query

```
INSERT INTO orders([order_id],[order_day],[fk_customer_id],[fk_salesperson_id])  
VALUES(1, '09/17/2019',1,1)  
  
INSERT INTO order_detail([fk_item_id],[fk_order_id],[order_quantity]) VALUES(1,1,8)
```

	fk_item_id	fk_order_id	order_quantity	order_id	order_day	fk_customer_id	fk_salesperson_id
1	1	1	8	1	2019-09-17	1	1
2	2	2	4	1	2019-09-17	1	1
3	3	3	8	1	2019-09-17	1	1
4	4	4	9	1	2019-09-17	1	1
5	5	5	10	1	2019-09-17	1	1
6	6	6	8	1	2019-09-17	1	1

Table 25 - Create Invoice table

## 6.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Out of stock when ordering	Select item with quantity: 1000000000	Show message (“Out of stock”)	Accept order	Fail
2	Duplicate customer name when entering	Input Name of customer: ‘Tran Quang Huy’	Check if the customer already exists or create a new customer.	Create a new customer with same name	Fail

Table 26 - Test case function Invoice

## 7. Function “Customer”

### 7.1. Interface design

This interface allows salesperson and manager can list customers of ElectroShop with these functions: Search, Add new, update and delete. But salesperson can use 2 functions such as: Search and Add new.

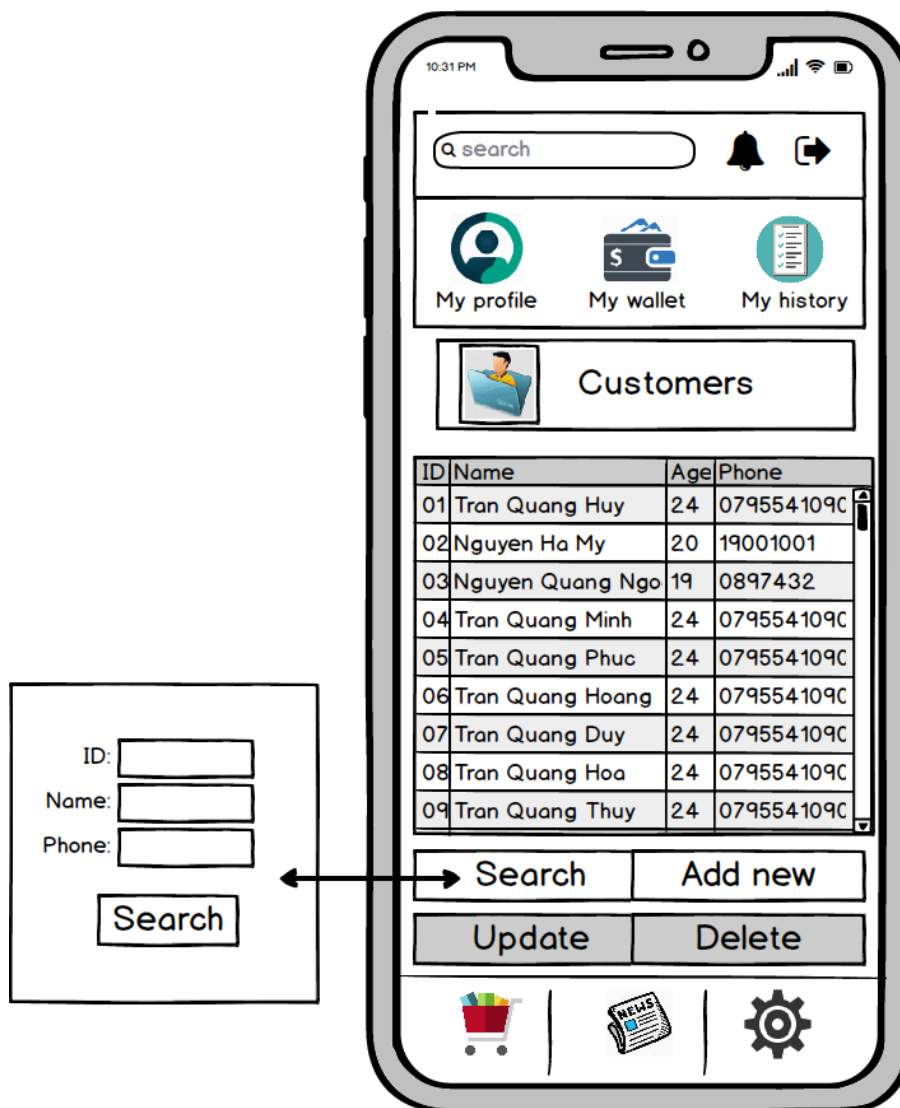


Figure 10 - Interface function Customers

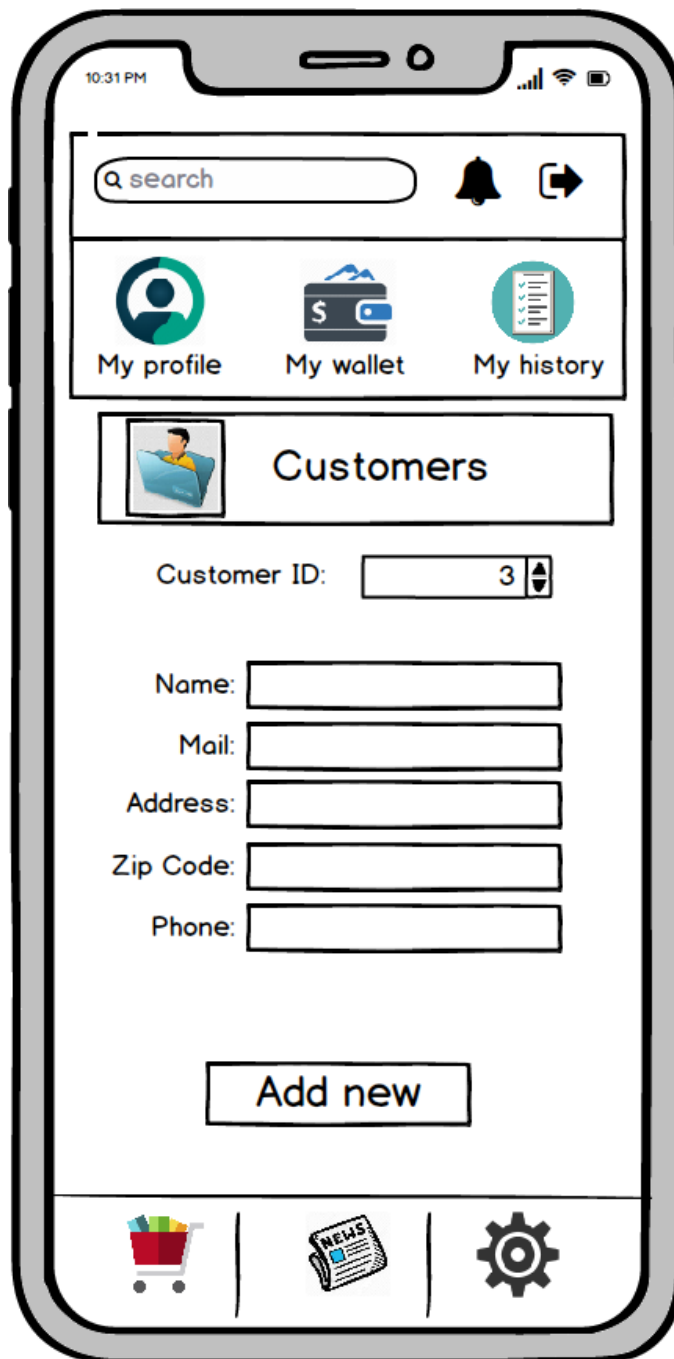


Figure 11 - Interface function Customer - Add New

## 7.2. Query

- Function Search:

```
SELECT * FROM customers
WHERE (customer_id = 10)
      OR (customer_name = 'huy')
      OR (customer_phone = '113');
```

	customer_id	customer_name	customer_mail	customer_address	customer_zipcode	customer_phone
1	10	Ronan Barlow	eu.metus.In@liberoMorbiaccumsan.net	9350 Torquent Road	JF5 7XZ	076 1219 2404

Table 27 - Result of function Search customer

- Function Add new:

```
INSERT INTO
customers([customer_id],[customer_name],[customer_mail],[customer_address],[customer_zipcode],
[customer_phone]) VALUES(1,'Allen Ayers','posuere.cubilia.Curae@ornare.edu','294-4580 At
Rd.','IC5 8NW','0819 171 8803')
```

- Function Update:

```
UPDATE customers
SET customer_name = 'huy', customer_phone = '0795541090'
WHERE customer_id = 10;
```

	customer_id	customer_name	customer_mail	customer_address	customer_zipcode	customer_phone
1	10	huy	eu.metus.In@liberoMorbiaccumsan.net	9350 Torquent Road	JF5 7XZ	0795541090

Table 28 - Result of function update

- Function Delete:

```
DELETE FROM customers
WHERE customer_id=10;
```

### 7.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Update customer name longer than 100 letters	Select function Update and input customer name: 'Tran Quang Huyyyyyyyyyy...yyyyy'	Database system does not accept this customer name	Database system does not accept this customer name	Pass
2	Delete customer does not exist in the system	Select function Delete	Database system does not perform this action	Database system does not perform this action	Pass

*Table 29 - Test case function customers*



## 8. Function “Supplier”

### 8.1. Interface design

This interface allows salesperson and manager can list customers of ElectroShop with these functions: Search, Add new, update and delete. But salesperson can use only function: Search.

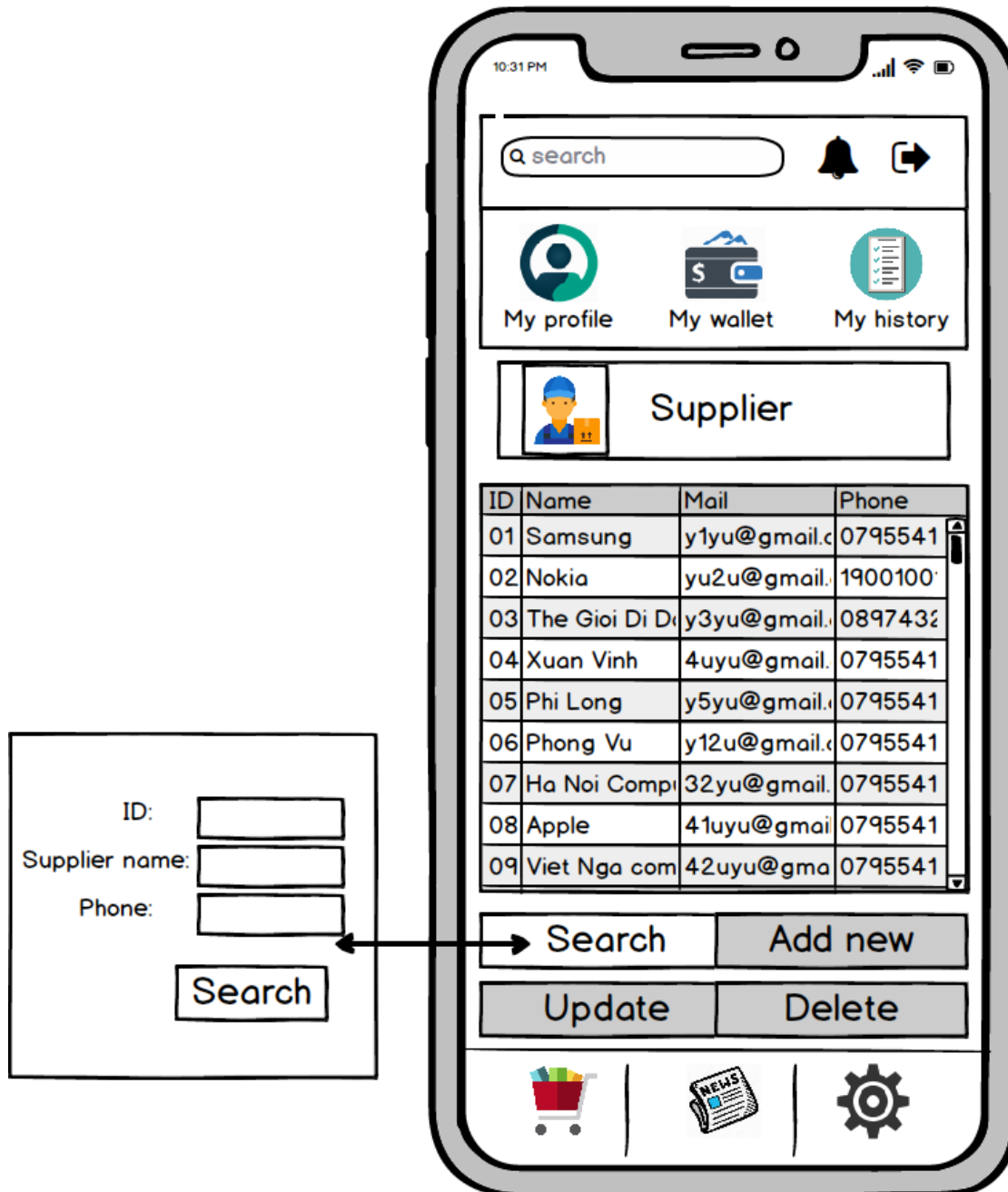


Figure 12 – Interface function Supplier

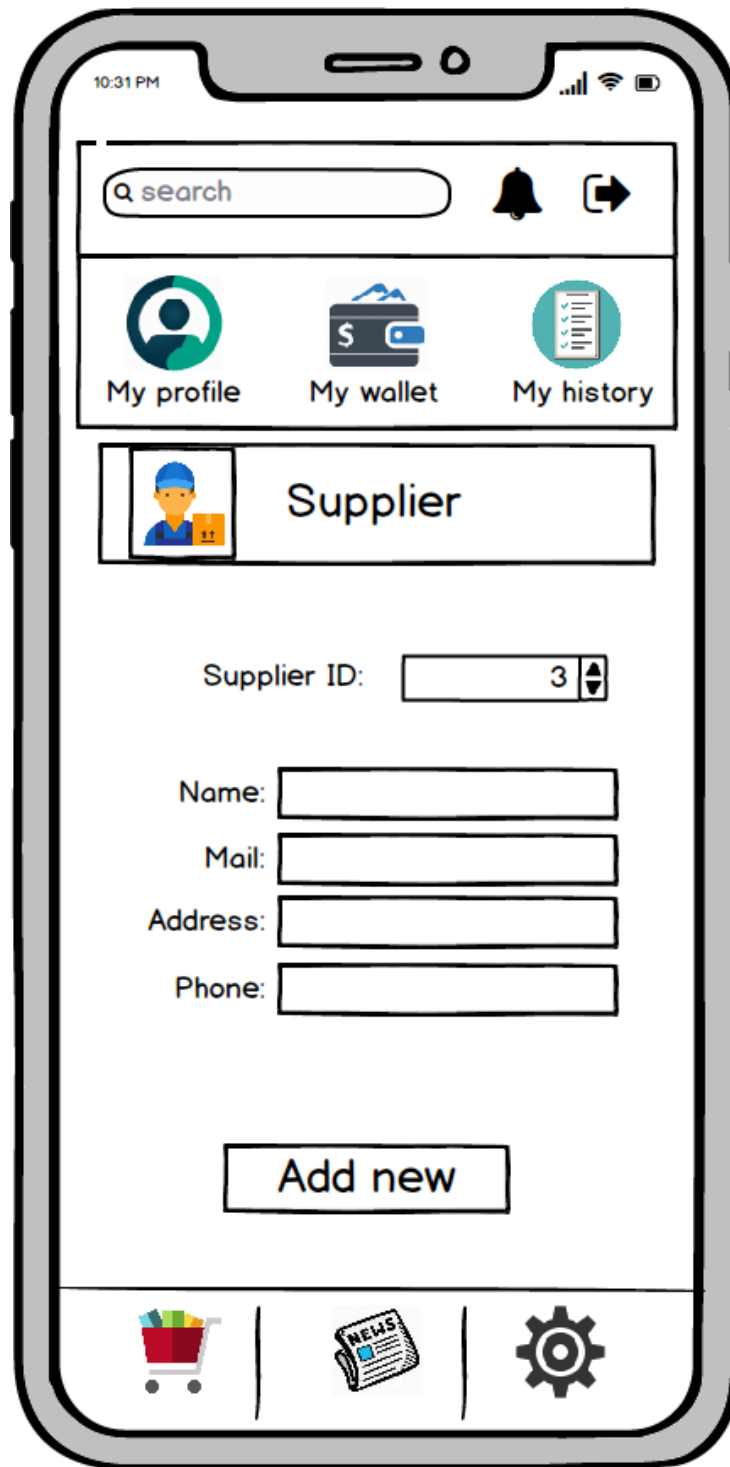


Figure 13 - Interface function Supplier - Add New

## 8.2. Query

- Function Search:

```
SELECT * FROM supplier
WHERE (supplier_id = 10)
      OR (supplier_name = 'huy')
      OR (supplier_phone = '113');
```

	supplier_id	supplier_name	supplier_mail	supplier_adress	supplier_phone
1	10	David Shannon	a.sollicitudin.orci@vestibulumloremsit.ca	Ap #562-8467 Pellentesque Av.	0952 569 8611

Table 30 - Result of function Search supplier

- Function Add new:

```
INSERT INTO
supplier([supplier_id],[supplier_name],[supplier_mail],[supplier_adress],[supplier_phone])
VALUES(1,'Silas Brock','molestie.Sed.id@ullamcorpennislarcu.net','547-6530 Lacus, St.','0800
1111')
```

- Function Update:

```
UPDATE supplier
SET supplier_name = 'huy', supplier_phone = '0795541090'
WHERE supplier_id = 10;
```

- Function Delete:

```
DELETE FROM supplier
WHERE supplier_id=10;
```

### 8.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Add same supplier exist in the database system	Select function Update and Supplier name: 'Apple'	Database system does not accept this same Supplier name	Database system accept this same supplier name	Fail
2	Add supplier mail improper standards of the system	Supplier Email: aa@bb#cc\$bb%bb.com	Database system does not add this supplier email and request add supplier again.	Database system add the improper standards email	Fail

*Table 31 - Test case function supplier*

## 9. Function “Salespersons”

### 9.1. Interface design

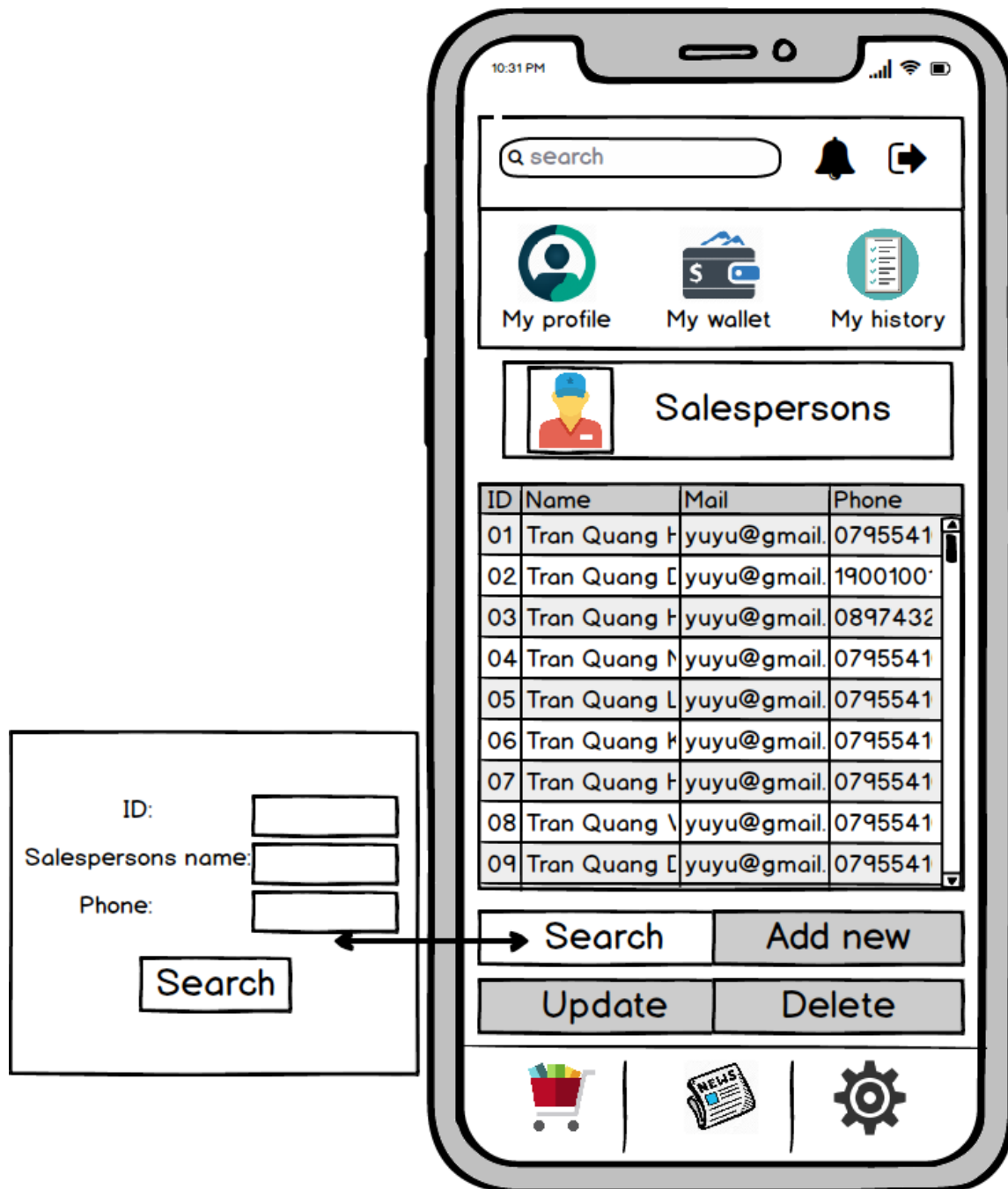


Figure 14 - Interface function Salespersons



Figure 15 - Interface function Salespersons - Add New

## 9.2. Query

- Function Search:

```
SELECT * FROM salespersons
WHERE (salesperson_id = 10)
      OR (salesperson_name = 'huy')
      OR (salesperson_phone = '113');
```

- Function Add new:

```
INSERT INTO
salespersons([salesperson_id],[salesperson_name],[salesperson_phone],[salesperson_age],[salesp
erson_mail]) VALUES(1,'Bert Washington','(016977)
5226',45,'dapibus.gravida.Aliquam@nonenim.edu')
```

- Function Update:

```
UPDATE salespersons
SET salesperson_name = 'huy', salesperson_phone = '0795541090'
WHERE salesperson_id = 10;
```

- Function Delete:

```
DELETE FROM salespersons
WHERE salesperson_id=10;
```

### 9.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Add same phone number of salespersons	Select function Add new and input phone number already has in database: '0795541090'	Database system does not accept this same Phone number	Database system accept this same Phone Number	Fail
2	List all salespersons information	Select function "salespersons"	Show all information of salespersons	Show all information of salespersons	Pass

*Table 32 - Test case function salesperson*



## 10. Function “Items”

### 10.1. Interface design

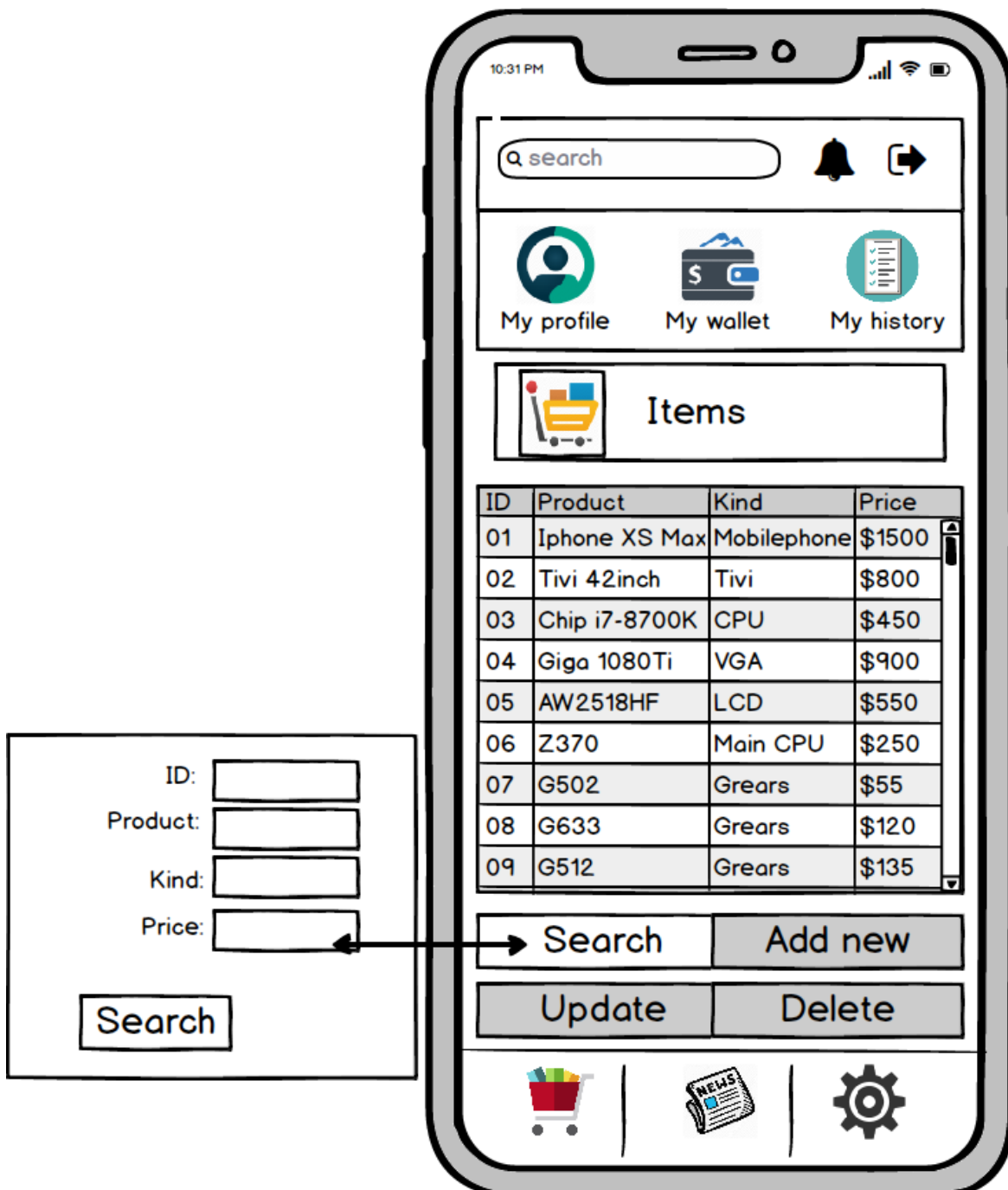


Figure 16 - Interface function Items

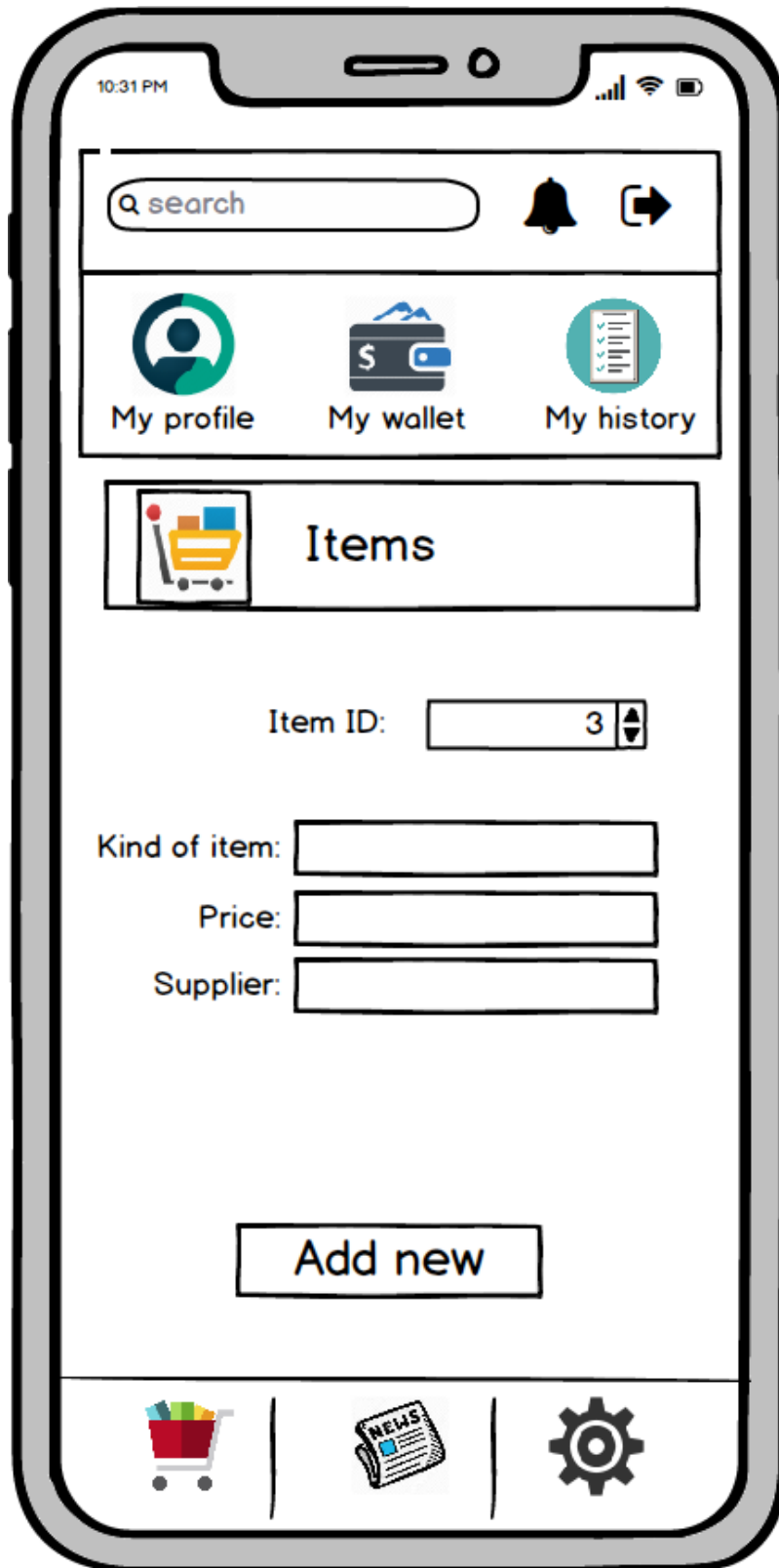


Figure 17 - Interface function Items - Add New

## 10.2. Query

- Function Search:

```
SELECT * FROM items
WHERE (item_id = 10)
      OR (item_kind = 'VGA')
      OR (item_price = '1134');
```

- Function Add new:

```
INSERT INTO items([item_id],[item_kind],[item_price],[fk_supplier_id])
VALUES(1,'Laptop',8115,1)
```

- Function Update:

```
UPDATE items
SET item_price = '41414141'
WHERE item_id = 10;
```

- Function Delete:

```
DELETE FROM items
WHERE item_id=10;
```

### 10.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Enter the price that is identical to another product already in the system	Item Price: '\$1500'	Database system accept this same Price	Database system accept this same Price	Pass
2	Delete one supplier	Select function "Delete"	Can not delete Supplier in "items" function	Can not delete Supplier in "items" function	Pass

*Table 33 - Test case function Items*

## 11. Function customer report

### 11.1. Interface design

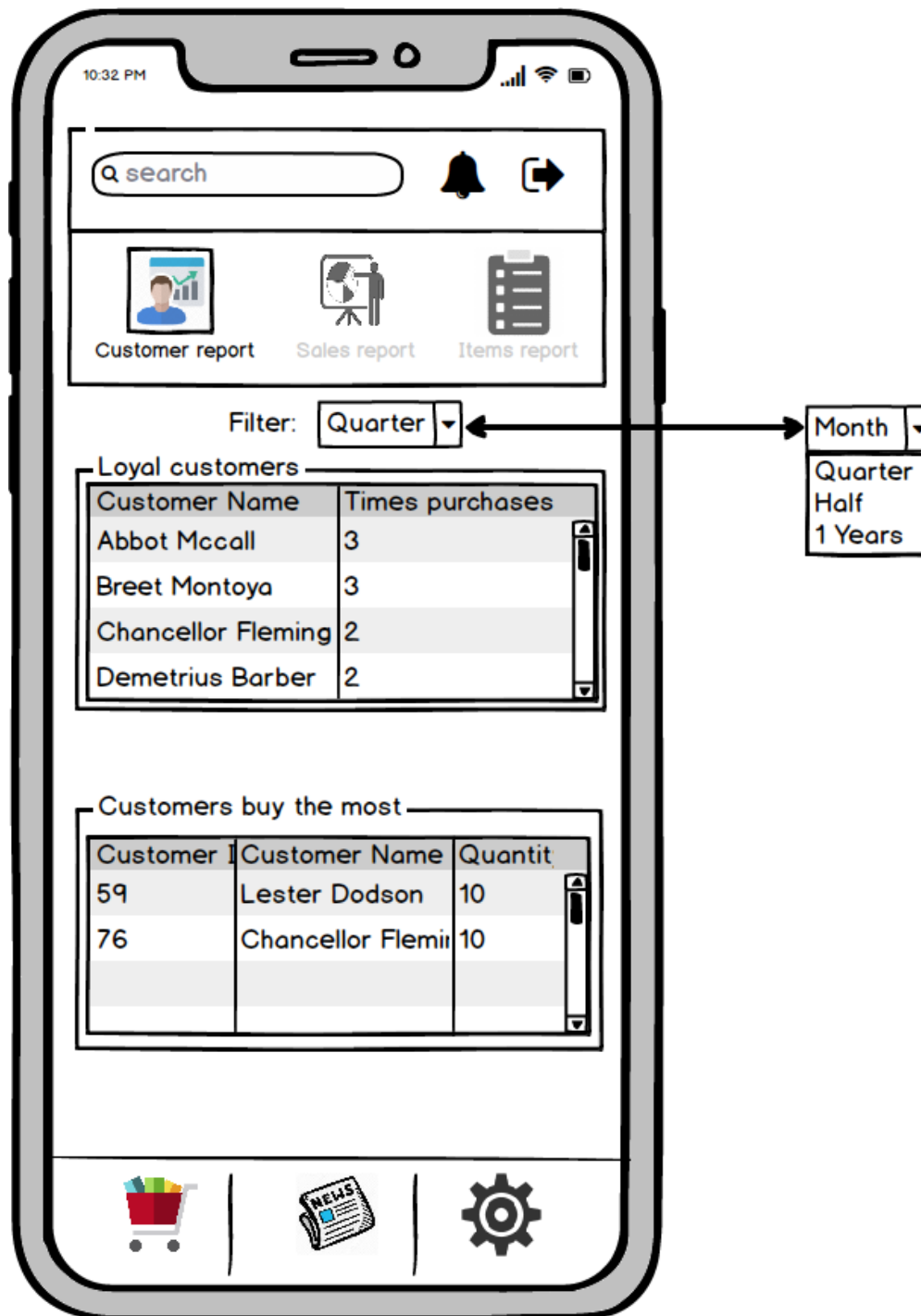


Figure 18 - Interface Function Customer Report

## 11.2. Query

- Loyal customers:

```
SELECT customers.customer_name AS CustomerName,  
       COUNT(orders.fk_customer_id) AS TimesPurchases  
FROM customers, orders  
WHERE (orders.order_day BETWEEN '4/1/2019' AND '7/1/2019')  
      AND (customers.customer_id = orders.fk_customer_id)  
GROUP BY customers.customer_name  
HAVING COUNT(orders.fk_customer_id) >=2;
```

CustomerName	TimesPurchases
Abbot Mccall	3
Brett Montoya	3
Chancellor Fleming	2
Demetrius Barber	2

Table 34 - Result of function loyal customer in customer report

- Customers buy the most:

```
SELECT customers.customer_id AS CustomerID,  
       customers.customer_name AS CustomerName,  
       order_detail.order_quantity AS Quantity  
FROM customers, order_detail, orders  
WHERE (order_detail.order_quantity = (SELECT MAX(order_quantity) FROM  
order_detail))  
      AND (order_detail.fk_order_id = orders.order_id)  
      AND (orders.fk_customer_id = customers.customer_id)  
      AND (orders.order_day BETWEEN '4/1/2019' AND '7/1/2019');
```

	CustomerID	CustomerName	Quantity
1	59	Lester Dodson	10
2	76	Chancellor Fleming	10

Table 35 - Result of function Customers buy the most

### 11.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Edit report of “customers report”	Time purchases: 1000	Database system does not accept this action	Database system does not accept this action	Pass
2	Move to function “sales report”	Select function “sales report”	Show interface of “sales report”	Show interface of “sales report”	Pass

*Table 36 - Test case function customer report*

## 12. Function Sales report

### 12.1. Interface design

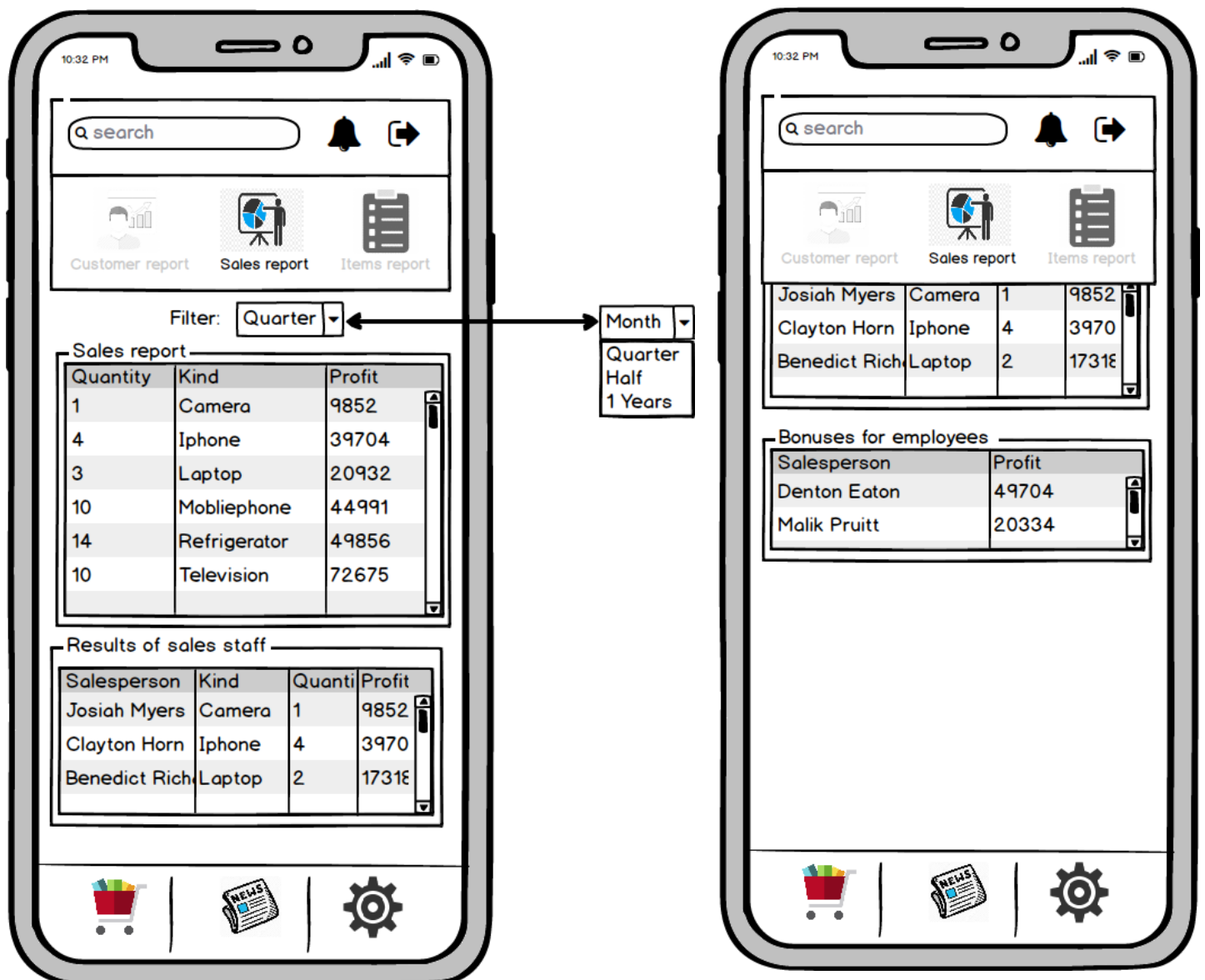


Figure 19 - Interface function Sales report



## 12.2. Query

- Bonuses for employees:

```
SELECT salespersons.salesperson_name AS SalespersonName,  
       SUM(order_detail.order_quantity*items.item_price) AS Profit  
FROM salespersons,items,order_detail,orders  
WHERE (salespersons.salesperson_id = orders.fk_salesperson_id)  
      AND (orders.order_id = order_detail.fk_order_id)  
      AND (items.item_id = order_detail.fk_item_id)  
      AND (orders.order_day BETWEEN '1/1/2018' AND '4/1/2018')  
      AND ((order_detail.order_quantity*items.item_price) >= 15000)  
GROUP BY salespersons.salesperson_name;
```

	SalespersonName	Profit
1	Denton Eaton	49704
2	Malik Pruitt	20334

Table 37 - Result of function bonuses  
for employees

- Result of sales employees:

```
SELECT salespersons.salesperson_name AS SalespersonName,  
       items.item_kind AS KindOfItem,  
       SUM(order_detail.order_quantity) AS Quantity,  
       SUM(order_detail.order_quantity*items.item_price) AS Profit  
FROM salespersons,items,order_detail,orders  
WHERE (salespersons.salesperson_id = orders.fk_salesperson_id)  
      AND (orders.order_id = order_detail.fk_order_id)  
      AND (items.item_id = order_detail.fk_item_id)  
      AND (orders.order_day BETWEEN '4/1/2018' AND '7/1/2018')  
GROUP BY salespersons.salesperson_name, items.item_kind;
```

	SalespersonName	KindOfItem	Quantity	Profit
1	Josiah Myers	Camera	1	9852
2	Clayton Horn	Iphone	4	39704
3	Benedict Richards	Laptop	2	17318
4	Dustin Frye	Laptop	1	3614
5	Evan Ferguson	Mobliephone	3	651
6	Ian Abbott	Mobliephone	1	4104
7	Neil Rios	Mobliephone	6	40236

Table 38 - Result of function sales employees

- Sales report:

```
SELECT items.item_kind,
       SUM(order_detail.order_quantity) AS TotalQuantity,
       SUM(order_detail.order_quantity*items.item_price) AS Profit
FROM items, order_detail, orders
WHERE (orders.order_day BETWEEN '4/1/2018' AND '7/1/2018')
      AND (items.item_id = order_detail.fk_item_id)
      AND (orders.order_id = order_detail.fk_order_id)
GROUP BY items.item_kind;
```

	item_kind	TotalQuantity	Profit
1	Camera	1	9852
2	Iphone	4	39704
3	Laptop	3	20932
4	Mobliephone	10	44991
5	Refrigerator	14	49856
6	Television	10	72675
7	VGA	2	1068

Table 39 - Result of function sales report

### 12.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Select filter option “2 years”	Select function filter and pick option “2 years”	Can not pick option “2 years”	Can not pick option “2 years”	Pass
2	Log out	Select function with icon log out	Log out of system	Log out of system	Pass

Table 40 - Test case function sales report

## 13. Function Items report

### 13.1. Interface design

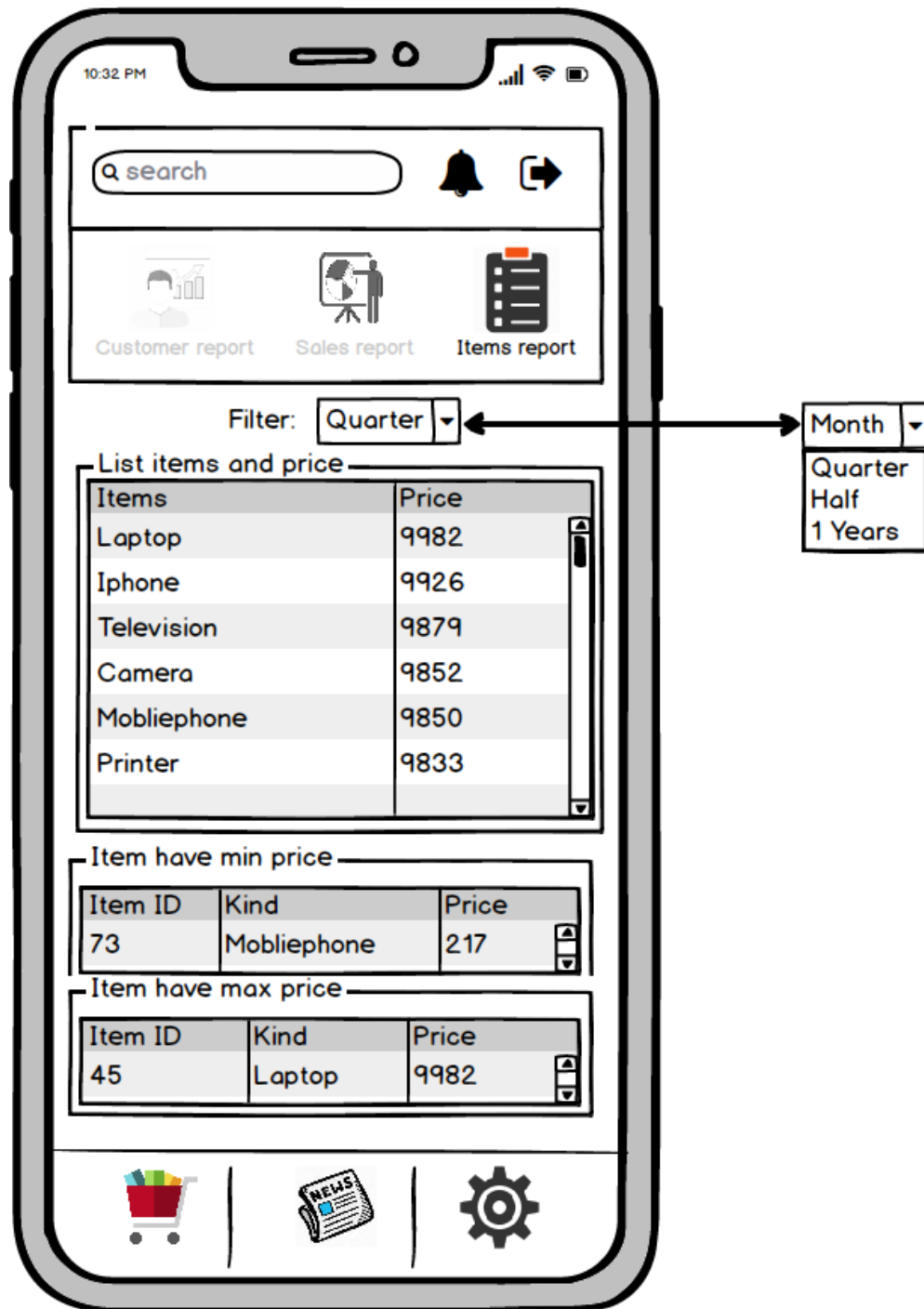


Figure 20 - Interface function Items report

### 13.2. Query

- List of items and price

```
SELECT items.item_kind AS KindOfItem,  
       items.item_price AS Price  
FROM items  
ORDER BY items.item_price DESC;
```

	KindOfItem	Price
1	Laptop	9982
2	Iphone	9926
3	Television	9879
4	Camera	9852
5	Mobliephone	9850
6	Printer	9833
7	Printer	9822
8	VGA	9576
9	Printer	9490
10	Mobliephone	9465
11	VGA	9340

*Table 41 - Result of  
function List of items and  
price*

- Min, Max price of items:

```
SELECT * FROM items  
WHERE item_price = (SELECT MIN(item_price) FROM items);  
  
SELECT * FROM items  
WHERE item_price = (SELECT MAX(item_price) FROM items);
```

	item_id	item_kind	item_price	fk_supplier_id
1	45	Laptop	9982	45

	item_id	item_kind	item_price	fk_supplier_id
1	73	Mobliephone	217	73

*Table 42 - Result of function Min, Max price*

### 13.3. Test case

S.No	Action	Inputs	Expected Output	Actual Output	Test result
1	Select an item type and show what the supplier of this item type	Select items: VGA	Show supplier of item that selected	Show not thing	Fail
2	Move to function "Customer report"	Select function "Customer report"	Display "customer report" interface	Display "customer report" interface	Pass

*Table 43 - Test case function items report*

# CONCLUSION

## 1. How Interface Design and Database Design assessments meet user requirements.

### 1.1. Evaluate the responsiveness of Database with user requirements

Base on result of evaluate the responsiveness of Database with user requirements in survey form in three days. That we can know what important to improve function for system.

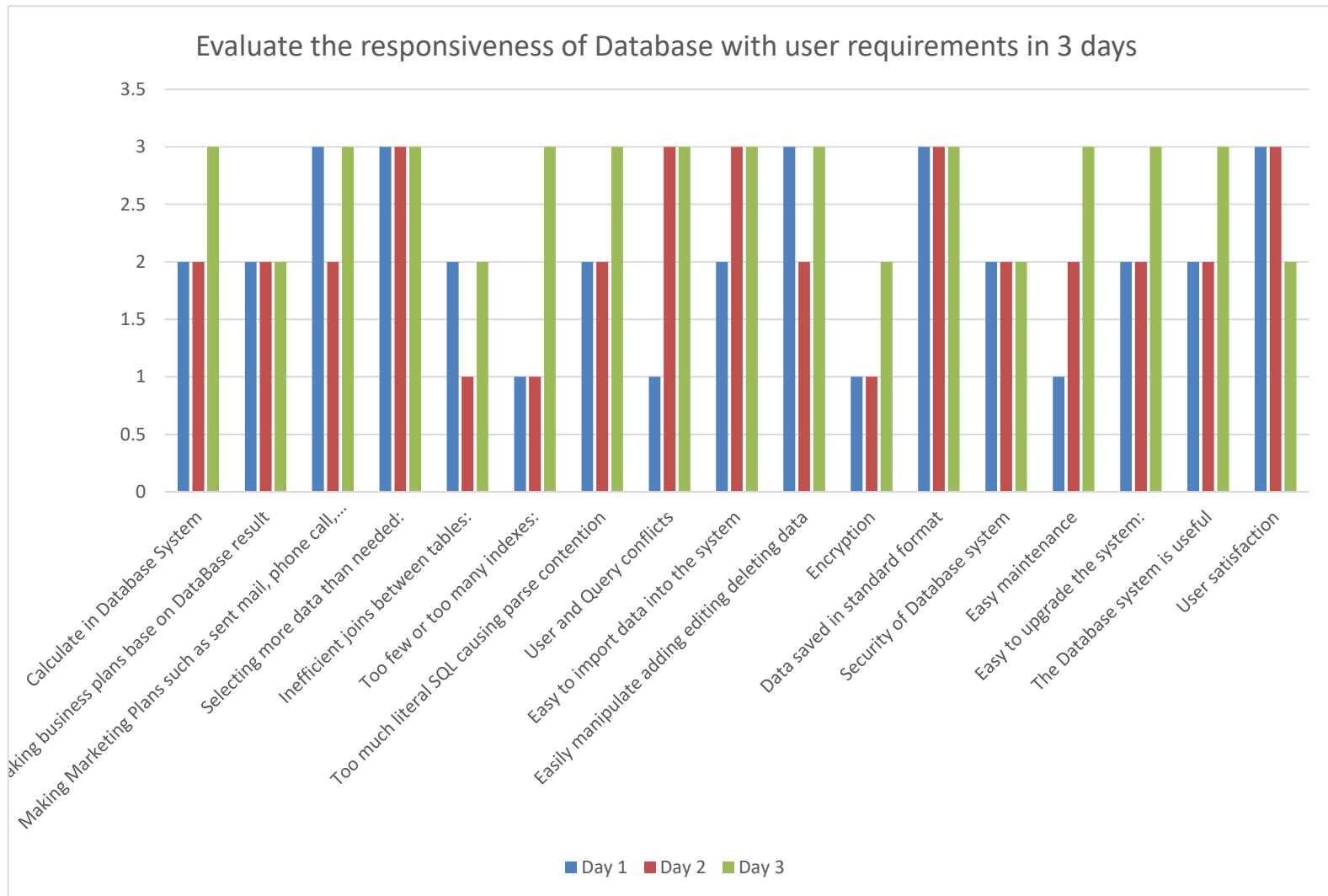


Figure 21 - Evaluate the responsiveness of Database with user requirements in 3 days

Evaluation Criteria	Reviews from users	Evaluate
<b>Business:</b> <ul style="list-style-type: none"> <li>- Calculate</li> <li>- Business plans</li> <li>- Making Marketing Plans such as sent mail, phone call, sent advertisement, etc.</li> </ul>	<ul style="list-style-type: none"> <li>- User almost satisfied with database calculate</li> <li>- The Database cannot work well in making business plans</li> <li>- Easily to contact customers and marketing</li> </ul>	High Medium High
<b>Query Performance:</b> <ul style="list-style-type: none"> <li>- Selecting more data than needed</li> <li>- Inefficient joins between tables</li> <li>- Too few or too many indexes</li> <li>- Too much literal SQL causing parse contention</li> <li>- User and Query Conflicts</li> </ul>	<ul style="list-style-type: none"> <li>- This system still acceptable to select more data</li> <li>- Tables in this database inefficient to join between</li> <li>- There are a few excess information</li> <li>- Database cause parse contention</li> <li>- Query is not clean enough and make user confusing</li> </ul>	High Medium Medium High High
<b>Data:</b> <ul style="list-style-type: none"> <li>- Easy to import data into the system</li> <li>- Easily manipulate adding, editing, deleting data</li> <li>- Encryption</li> <li>- Data saved in standard format</li> </ul>	<ul style="list-style-type: none"> <li>- User easy import any kind of data into database</li> <li>- User easy add, edit, delete data in database system</li> <li>- The data still not encryption in the database</li> <li>- Most data is saved in standard format</li> </ul>	High High Low High
<b>Other Issues:</b> <ul style="list-style-type: none"> <li>- Security of Database system</li> <li>- Easy maintenance</li> <li>- Easy to upgrade the system</li> </ul>	<ul style="list-style-type: none"> <li>- Security of Database system is acceptable</li> <li>- The maintenance problem is relatively good</li> <li>- Update the database system accordingly</li> </ul>	Medium Medium High
<b>General assessment:</b> <ul style="list-style-type: none"> <li>- The Database system is useful:</li> <li>- User satisfaction</li> </ul>	<ul style="list-style-type: none"> <li>- The user feels the database is useful for the job</li> <li>- Most users feel satisfied when using this database system</li> </ul>	High High

Table 44 - Evaluate the responsiveness of Database with user requirements

### 1.2.Interface Design and Database Design assessments user requirements.

User requirement	Interface Design	Database Design
<b>Business:</b>		
Calculate	Shows the results of user, employee and business analysis on the interface	Query statements are preceded to meet Interface calculations
Business plans	Not yet shown about business planning	The Database cannot work well in making business plans
Making Marketing Plans such as sent mail, phone call, sent advertisement, etc.	Easily to contact customers and marketing with customer, supplier, salesperson's email or phone number	Query statements are preceded to meet Interface list of email or phone number.
<b>Query Performance:</b>		
Selecting more data than needed	Interface can not select more data in some function	This system still acceptable to select more data
Inefficient joins between tables	Interface can not joins between tables at database in some function	Tables in this database inefficient to join between
Too few or too many indexes	Interface of the user indicates the necessary data	There are a few excess information
Too much literal SQL causing parse contention	Interface is working well without causing parse contention	Database cause parse contention
User and Query Conflicts		Query is not clean enough and make user confusing

**\*\* Continue in the following page**



User requirement	Interface Design	Database Design
<b>Data:</b>		
Easy to import data into the system	Interface has function add new can help user easy to import data into database system and function register can import data of customer	User easy import any kind of data into database
Easily manipulate adding, editing, deleting data	Interface has function Update, Delete that user easy add, edit, delete data in database system	Query statements are preceded to meet Interface function
Encryption	The data is protected but not high such as password and information about customer, salesperson	The data still not encryption in the database
Data saved in standard format	Also rely heavily on people who enter data	Most data is saved in standard format
<b>Other Issues:</b>		
Security of Database system	The system is currently not at fault for any security issues	Security of Database system is acceptable
Easy maintenance	Users cannot perform maintenance on the user interface	The maintenance problem is relatively good
Easy to upgrade the system	Users cannot perform upgrade the system on the user interface	Update the database system accordingly
<b>General assessment:</b>		
The Database system is useful:	Interface is used to input right data in right database	The user feels the database is useful for the job
User satisfaction	The interface is easy to use	Most users feel satisfied when using this database system

*Table 45 - Interface Design and Database Design assessments user requirements*

## 2. Advantages and Disadvantages

### 2.1. Advantages:

- Data is divided into clear tables, not duplicated or too dependent on each other
- Flexibility of the System: Since changes are often necessary to the contents of the data stored in any system, these changes are made more easily in a centralized database than in a conventional system and Applications programs need not to be changed on changing the data in the database. (vissicompcodder, 2015)
- Integrity can be improved: In the conventional systems because the data is duplicated in multiple files so updating or changes may sometimes lead to entry of incorrect data in some files wherever it is applicable.
- Standards can be enforced: Standards may relate to the naming of data, structure of data, format of the data etc.
- Easy to use: most functions that require less than 2 touches.
- Easy-to-use interface design for new users.

## **2.2. Disadvantages:**

- Security is not good: The security of information about customers, employees, suppliers and especially account and password is not good.
- Increased costs: One of the disadvantages is Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software, and personnel required to operate and manage a database system can be substantial. Training, licensing, and regulation compliance costs are often overlooked when database systems are implemented.
- Frequent upgrade/replacement cycles: Vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features.
- Maintain regularly: Because database technology advances rapidly, personnel training costs tend to be significant. To maximize the efficiency of the database system, you must keep your system current. Therefore, you must perform frequent updates and apply the latest patches and security measures to all components. (THIRU, 2018)
- Extensive conversion costs: In moving from a file-based system to a database system If you are currently working on file based system and need to upgrade it to database system, then large amount of cost is incurred in purchasing different tools, adopting different techniques as per the requirement.
- Damage to database affects virtually all applications programs: If one part of the database is corrupted or damaged because of the hardware or software failure, since we don't have many versions of the file, all the application programs which are dependent on this database are implicitly affected.
- Still has not met many test cases, it is necessary to have optimal measures in the interface and database.

### 3. Remedial solutions

#### 3.1. SQL Server:

Therefore, it is necessary to determine the appropriate DATA TYPE for each data attribute to ensure optimal memory during use.

In this Database system of ElectroShop used most data types “varchar(100)” so it can not include Unicode types so in the future, when someone input the value with Unicode types will make Database system have trouble.

Finally, we should change type of data from “varchar(100)” to “Nvarchar(100)” this is better solution.

A database consists of many tables, between the tables have a relationship with each other through the Primary Key & Foreign Key.

Primary Key, Foreign Key: In this ElectroShop, the Primary Key and Foreign Key is used for define Only unique, non-duplicate data exists but not contains non-empty values (NULL) and Foreign Key must have the same data type, the same number of fields that have the corresponding Primary Key sorted. This will cause the Database system have a lot of trouble, so contains non-empty values (NULL) should be add to Primary Key and Foreign Key in each tables of Database system

- Some notes in table initialization:
  - o Create a new Table in the current Database: Before create a new table, make sure that we are using right database, if not use syntax: “use <database>”.
  - o The Table has not been found: In the process of creating a Table with code, in some cases we cannot see the Table that we just created, so right-click Table and Refresh to update the Table list.
  - o Each Table and Column has only one name in the Table.
- In addition, one of the important things is to know relations (Relationship) between the tables:
  - o One-to-One Relationships or 1-1 relations: in this relationship, a row on table A cannot be linked to more than 1 row on Table B and vice versa.
  - o One-to-Many Relationships or 1-n relationships: in this relationship, a row on the table A can be linked to multiple rows on Table B.
  - o Many-to-Many Relationships or relationships: in this relationship, a row on Table A can be linked to multiple rows on Table B and a row on Table B can also link to multiple rows on Table A. As we can see in the example above a book can be written by many authors and an author can also write many books. Therefore the relationship between Books and Authors is Many to Many relations. In this case, an intermediate table is often used to solve the problem (AuthorBook table).
- Do not use multiple temporary tables when querying

### **3.2.Interface Design:**

- The way the similar functions are still in different interfaces, the need to simplify, avoid causing too many user interfaces will affect the time and memory of the database.
- Successfully enhance application:
  - Creative direction
  - Strong layout and graphic redesign
  - Custom graphics, logo design and identity/brand development
  - Content analysis and generation
  - Usability
  - Information architecture
  - Database integration
  - Online branding
  - Creative online advertising solutions
- Presentation Design: Whatever your need for a presentation, whether it be for corporate events or a conference. We can create presentations to suit all needs. We enhance these new presentations and existing presentations by adding: (evampsaanga, n.d.)
  - Animation
  - Graphics
  - Sounds
  - Music

### **3.3.Fix problems from test case:**

- Login: Distinguish lowercase letters and capital letters
- Input Data: Do not leave the fill information blank
- Register / Add new:
  - Do not accept special characters
  - Phone numbers are not duplicated
  - Email must fill in the correct standard form.
- Function: Functions can show results from different tables at the same time
- Supplier: Need to know if the product is still in stock.

## APPENDIX: Survey Form

Code:.....

Survey users on the database, the feasibility of the system on a scale: High - Medium – Low.

Evaluation Criteria	Evaluate		
	High	Medium	Low
<b>Business:</b>			
Calculate			
Business plans			
Making Marketing Plans such as sent mail, phone call, sent advertisement, etc.			
<b>Query Performance:</b>			
Selecting more data than needed			
Inefficient joins between tables			
Too few or too many indexes			
Too much literal SQL causing parse contention			
User and Query Conflicts			
<b>Data:</b>			
Easy to import data into the system			
Easily manipulate adding, editing, deleting data			
Encryption			
Data saved in standard format			
<b>Other Issues:</b>			
Security of Database system			
Easy maintenance			
Easy to upgrade the system			
<b>General assessment:</b>			
The Database system is useful:			
User satisfaction			

Google Form: <https://goo.gl/forms/kS3JUKCDD5jcs1e62>

## References

*evampsaanga*. (n.d.). Retrieved from evampsaanga: <https://evampsaanga.com/products-services/web-solutions/ui-ux-design/>

THIRU. (2018). *myreadingroom*. Retrieved from myreadingroom: <http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/462-advantages-and-disadvantages-of-dbms.html>

*vissicompcodder*. (2015, 2 3). Retrieved from vissicompcodder: <https://vissicompcodder.wordpress.com/category/advantages-and-disadvantages-of-database-systems/>