



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FIRST SEMESTER SESSION 2017/2018

SCSD2523 DATABASE

FINAL REPORT PROJECT DATABASE

LECTURER: DR SHARIN HAZLIN BTE HUSPI

(SECTION 01)

PREPARED BY:

- 1. LIM HUI TING**
- 2. NURAINA SYAZA BINTI AZMAN**
- 3. VIVEKAANAN A/L SIRGUNAN**
- 4. YAP SIN YEE**

DATA BASE PROJECT SAMPLE

SUBMISSION DATE: 3rd JANUARY 2018

Table of Contents

1.0 Database Planning and System Definition	1
1.1 Overview of the Current System	1
1.2 Database Planning	1
1.2.1 Mission Statement	1
1.2.2 Mission Objective.....	2
1.3 System Definition	2
1.3.1 Systems Boundary	2
1.3.2 Major User View	3
1.4 Gantt Chart	4
2.0 User's Requirement Specification.....	5
2.1 Data Requirement	5
2.2 Transaction Requirements	6
2.3 Cross-Reference Analysis.....	7
3.0 Conceptual Database Design	8
3.1 Entity Relationship Design.....	8
4.0 Logical Database Design.....	9
4.1 Logical Entity Relationship Diagram (Logical ERD).....	9
4.2 Relation Database Schema	10
4.3 Data Dictionary for Logical Database Design.....	13
5.0 Project Implementation.....	16
5.1 Structured Query Language.....	16
5.2 System Prototype Interface.....	23
5.3 Set of Queries for each transaction.....	35
5.4 Data Flow Diagram (DFD).....	41
5.4.1 Context Diagram.....	41
5.4.2 DFD Level 0	42
5.4.3 Child Diagram for Process 2	43
5.4.4 Child Diagram for Process 3	44
5.4.5 Child Diagram for Process 4	45
5.4.6 Child Diagram for Process 6	46
Appendix	47
Interview Output.....	47
Other Supporting	51

PROJECT PHASE1, PHASE 2 AND PHASE 3

1.0 Database Planning and System Definition

1.1 Overview of the Current System

Iddin Enterprise is a bicycle shop opened in September 2014, located in Universiti Teknologi Malaysia (UTM) Johor Bahru campus. The bicycle shop provides services such as buy, service and rent bicycles. Since then, the shop has grown steadily and now has more and more regular customers especially the students in UTM. Furthermore, in order to provide more choices of bicycle and accessories, the shop has purchased more products and the categories of product had also increased. According to the shop owner, the current system which is vastly based on handwritten record is not effective and the data are either being forgot to record or update leading to lack of data integrity. Below are some of the sample records from the bicycle shop.

DATE	NAME & PHONE NUMBER	MODEL
1/1/15	Sherif Khalid Abuzelmaiz 01137080924	FONIX 29-06 BLUE COLOUR ✓
-	AN DONG 011-21678098	26-118 OPEN COLOUR SOTHE
1/2/15	Khalid Suwan Buranen 01131745877	22-06 BLACK COLOUR ✓
1/2/15	Ar.Rainbow 0102687647	Rainbow Upgrade. - 26X23 - 26X24
10/2	Siti Hajar 01135531923 Norparham 011-28579391	26-OSA (8) ✓ 26-OSA (m) ✓
11/2	Nury Zik 016-4034356	26-OSA(R) ✓
12/2	Mukimah binti Ossman 989-7715026	07-06(8) ✓
26/2	Nuramni Ashikin binti Anuar 013-2893559	URATA BIRU Y frame

Figure 1 Customer & Sales Record

IDDIN ENTERPRISE (JMO65241-H)
LOT 105 STUDENT MALL UNIVERSITI TEKNOLOGI MALAYSIA
81300 SKUDAI JOHOR
HP: 011-11949 817 kedaibasikalutm@gmail.com
www.kedaibasikalutm.blogspot.com

M/S: SYARIF SELIM		SERIAL NO.: 00203	
Contact No: 016-8189062		DATE / TARikh: 30-9-2017	
	PARTICULARS BUAH-BUAH	QUANTITY KUANTITI	UNIT PRICE HARGA SEJUNIT
1	NEW BIKECYCLE	01	280
2	SEVEN UP 26'		280.00
3			
4			
5	A17 KHO159		
6			
7			
8			
9			
10			
11			
12			
13			
* All O.E. Goods sold are not returnable			
		TOTAL / JUMLAH (RM)	280.00
		DEPOSIT (RM)	
		BALANCE (RM)	
Received by / Diterima oleh		Signature / Tandatangan	

Figure 2 Receipt

1.2 Database Planning

1.2.1 Mission Statement

The purpose of the o7o database system is to maintain the data that is used and generated to support the bicycle shop e-commerce services such as online bicycle and accessories shopping for the customers. Besides, the o7o database system should ease the transaction of bicycle shop as it acts as an online interconnected database between shop owner, staff and customers.

1.2.2 Mission Objective

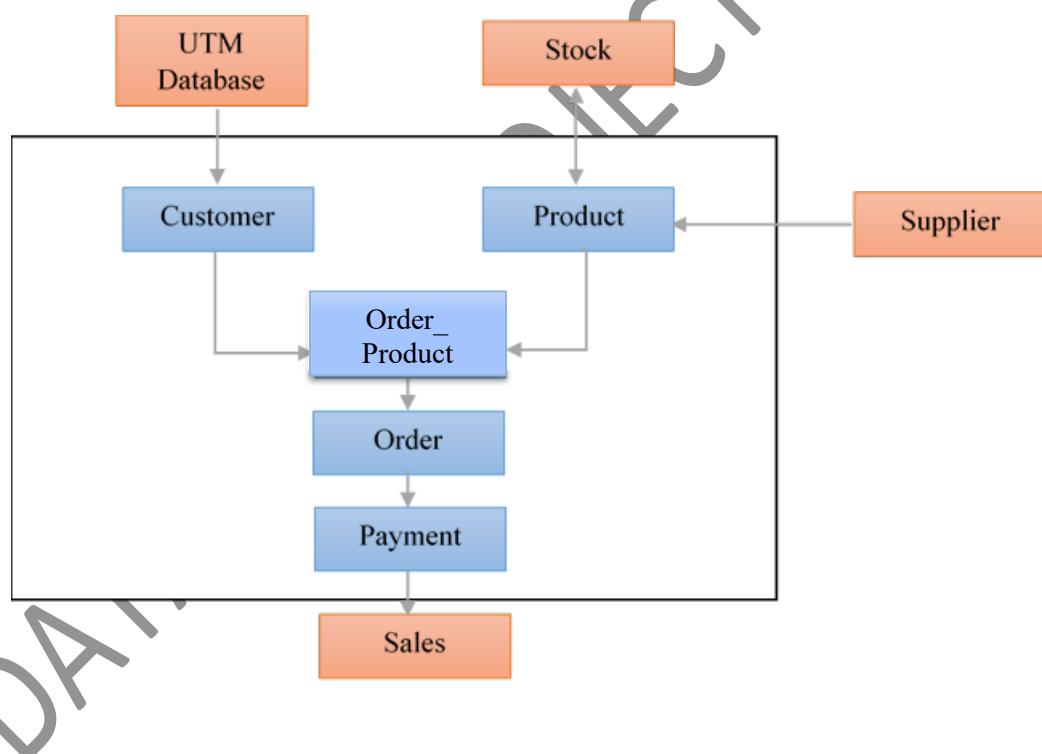
The mission objectives of our database planning are:

- a) To maintain (enter, update and delete) data on customer, product, cart, order and payment.
- b) To perform searches on customer, product, cart, order and payment.
- c) To track the status of product availability, the status of payment and the status of order.
- d) To report on customer, product, cart, order and payment.

1.3 System Definition

1.3.1 Systems Boundary

The figure below shows the system boundary for our project. The entities insides the black line box should be included in our system planning. Our focus for the project is to create a database system for online bicycle and accessories shopping system.



**Justification: Product include BIKE and ACCESSORIES table.

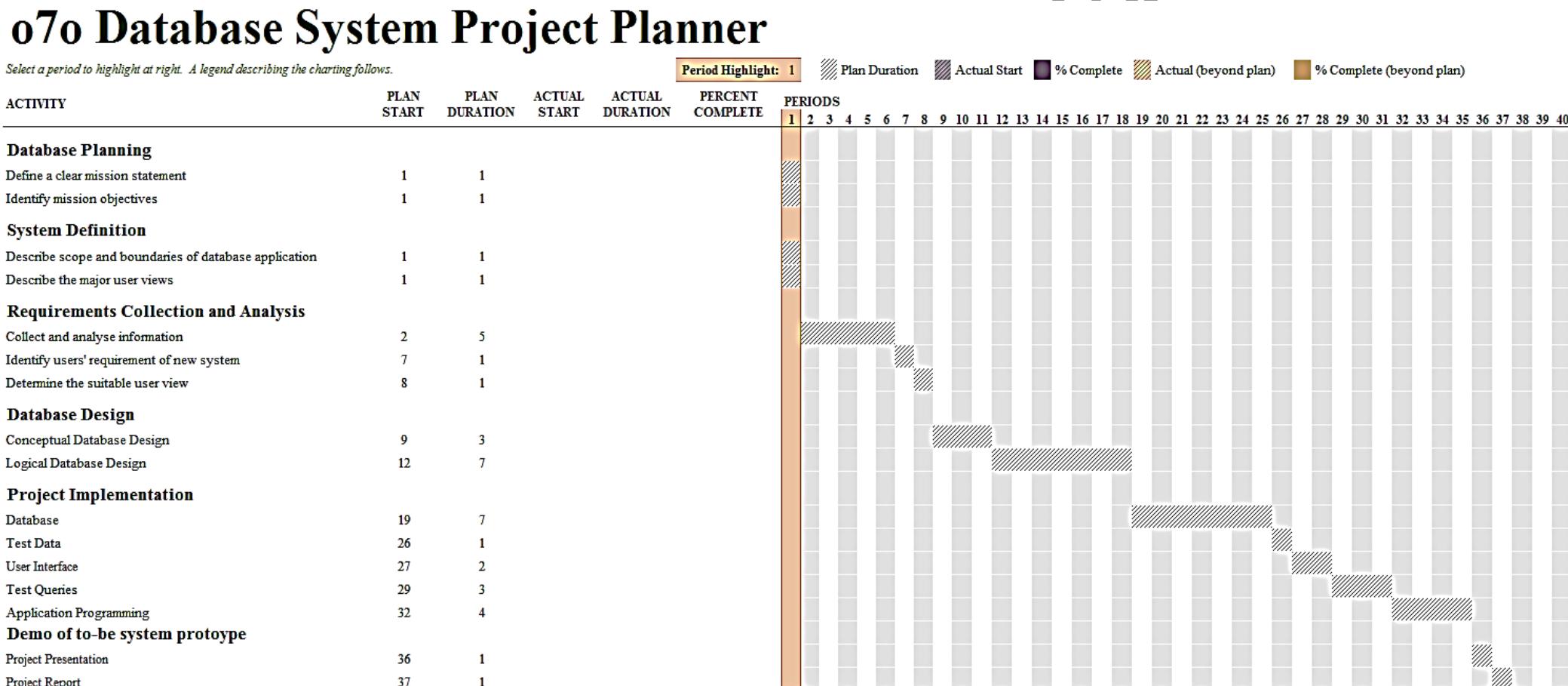
1.3.2 Major User View

The users are Manager, Staff and Customer. The major user views for each of the user group are shown in the table below:

Data	Access Type	Manager	Staff	Customer
Customers	Maintain Query Report	x x		x
Bike	Maintain Query Report	x x x	x x	x
Accessories	Maintain Query Report	x x x	x x	x
Order_Product	Maintain Query Report	x		x x
Order	Maintain Query Report	x x	x x	x
Payment	Maintain Query Report	x x x	x x	x

1.4 Gantt Chart

Below is the Gantt chart produced for the whole database system project.



2.0 User's Requirement Specification

2.1 Data Requirement

Based on interview and some record samples that we get from the bicycle shop (refer to Appendix 1), the data requirements are identified as below:

Entity	Data to be stored	Requirements of Data
Customer	1. Username 2. Password 3. Name 4. Phone number 5. Shipping Address	<ul style="list-style-type: none">[1] For username, UTM students and staff may use their matric number or staff ID as username.<ul style="list-style-type: none">The data of 3-5 will be taken from UTM Database
Bike	1. Product ID 2. Product Name 3. Price 4. Stock Quantity 5. Feature <ul style="list-style-type: none">a. Brandb. Modelc. Heightd. Weighte. Shifterf. Frameg. Tyreh. Chain Wheeli. Color	<ul style="list-style-type: none">[3] Price shown should be included with tax such as GST. No further add on after that.
Accessories	1. Product ID 2. Product Name 3. Price 4. Stock Quantity 5. Feature <ul style="list-style-type: none">a. Brandb. Modelc. Sized. Color	<ul style="list-style-type: none">[3] Price shown should be included with tax such as GST. No further add on after that.
Order_Product	1. Identity Number 2. Price 3. Quantity	<ul style="list-style-type: none">[1] Identity Number should be unique for each other.
Order	1. Order Number 2. Total price 3. Order date 4. Order status 5. Subtotal 6. Grand Total 7. Delivery Date	<ul style="list-style-type: none">[1] Order Number should be automatically generated by system.[5] Subtotal = (price *quantity) which refer to the price and quantity from Order_Product entity[6] Grand total = subtotal + shipping/handling fee

		<ul style="list-style-type: none"> • [7] Delivery Date is the date which the product is delivered •
Payment	<ol style="list-style-type: none"> 1. Transaction number 2. Bank 3. Payment Date 4. Total Price 	<ul style="list-style-type: none"> • [6] Payment Date is the date which the payment is confirmed

2.2 Transaction Requirements

The transaction requirements are identified based on three types of transaction (Data entry, Data update, Data deletion, Data queries):

Entity	Data	Data Entry	Data Update	Data Deletion	Data Queries
Customer	Username	Sign up by customer	Update info by customer	Delete Account by customer	Query on Customer data by Manager
	Password				
	Name				
	Phone number				
	Shipping Address				
Product	Product ID	Enter bike information by manager/staff	Update bike information by manager/staff	Delete bike information by manager/staff	Search for bike by manager, staff, customer
	Product Name				
	Price				
	Stock Quantity				
	Brand				
	Model				
	Height				
	Weight				
	Shifter				
	Frame				
	Tyre				
	Chain Wheel				
Accessories	Color	Enter accessories information by manager/staff	Update accessories information by manager/staff	Delete accessories information by manager/staff	Search for accessories by manager, staff, customer
	Product ID				
	Product Name				
	Price				
	Stock Quantity				
	Brand				
	Model				
	Weight				
	Size				
	Color				
Order_Product	Identity Number	Add product into cart by customer			
	Price				

	Quantity				
Order	Order Number	Confirm order by customer	Update order status by staff or manager	Cancel order by staff, manager	Query order by staff, manager
	Order date				
	Order status				
	Quantity				
	Subtotal				
	Grand Total				
	Delivery Date				
Payment	Transaction Number	Enter bank, transaction number & payment date by customer	Confirm payment by manager or staff	Delete payment by manager	Query payment by manager, staff
	Bank				
	Payment Date				
	Total Price				

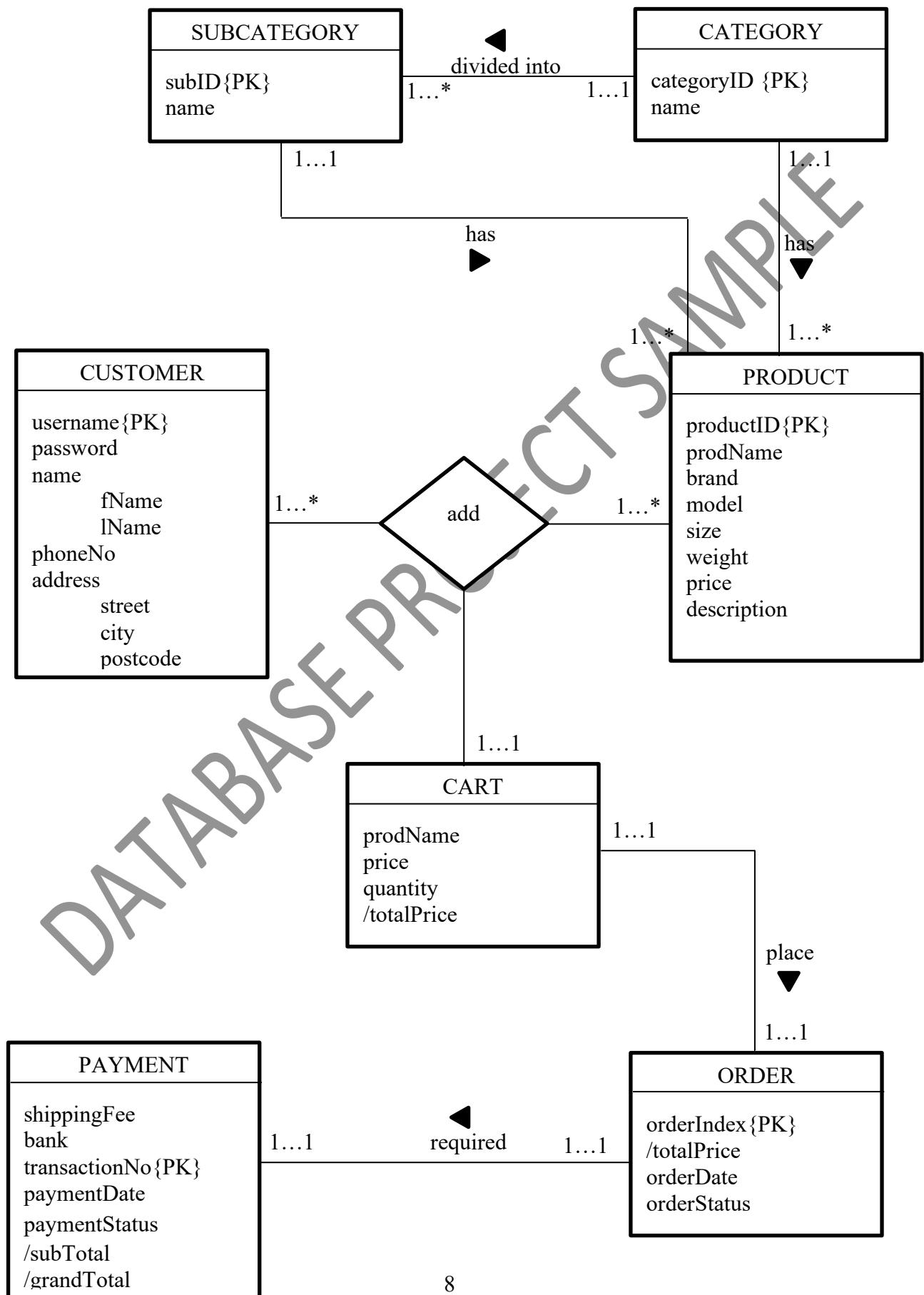
2.3 Cross-Reference Analysis

The table below cross-references the Manager, Staff and Customer user views with the main types of data used by each user view.

	Manager	Staff	Customer
Customer	x		x
Bike	x	x	x
Accessories	x	x	x
Order_Product	x		x
Order	x	x	x
Payment	x	x	x

3.0 Conceptual Database Design

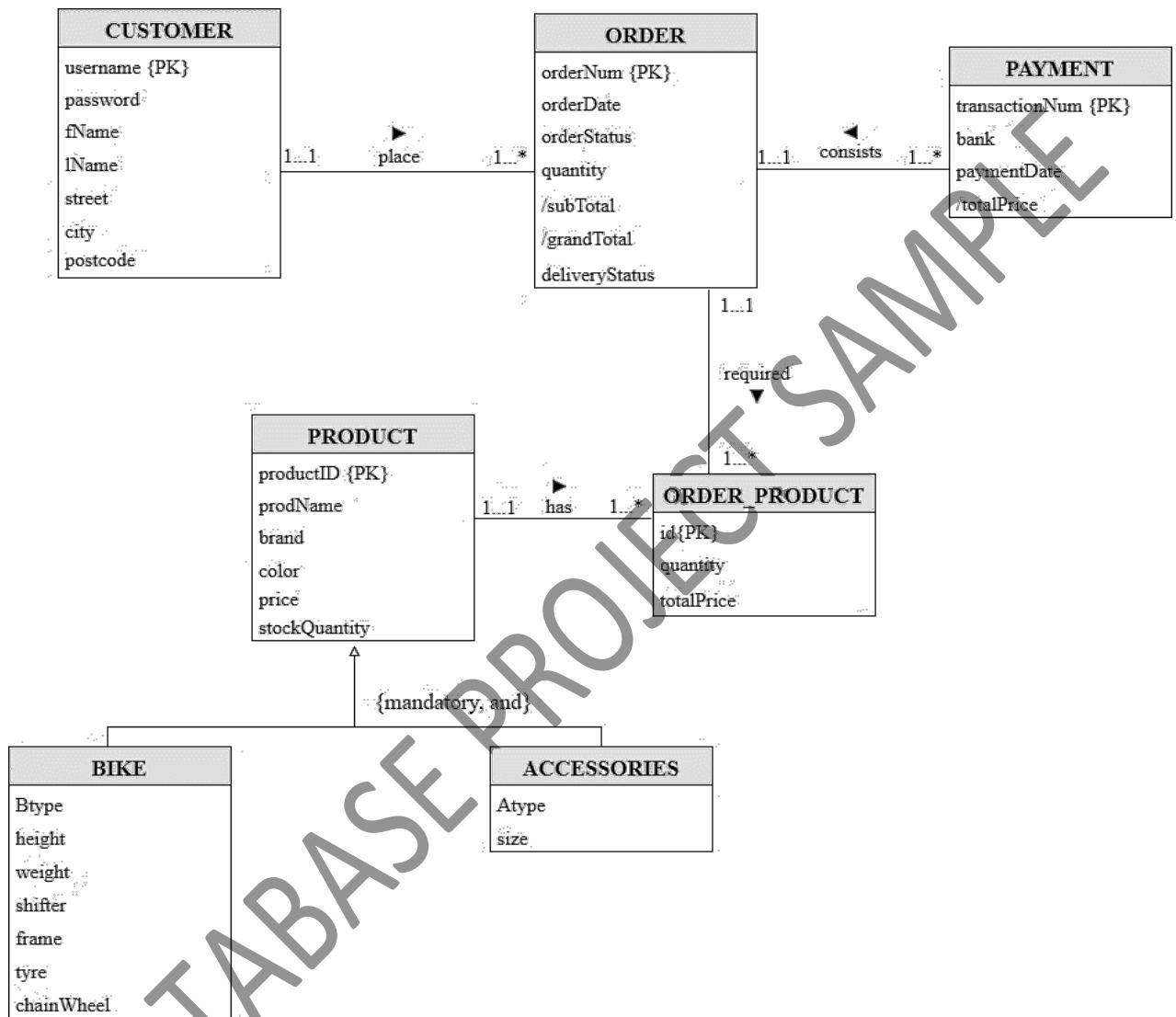
3.1 Entity Relationship Design



PROJECT PHASE 4

4.0 Logical Database Design

4.1 Logical Entity Relationship Diagram (Logical ERD)



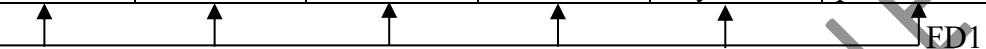
4.2 Relation Database Schema

Since there is no partial dependency (PD) and transitive dependency (TD) in all relation, therefore the relations **CUSTOMER**, **ORDER**, **PAYMENT** and **ORDER_PRODUCT** are already in 3NF relation.

i. **CUSTOMER (username, password, fName, lName, street, city, postcode)**

CUSTOMER

username	password	fName	lName	street	city	postcode



FD1: $\text{username} \rightarrow \text{password, fName, lName, street, city, postcode}$

FD1 is full functional dependency. Relation CUSTOMER is in 3NF.

ii. **ORDER (orderNum, orderDate, orderStatus, quantity, subTotal, grandTotal, deliveryStatus, username)**

ORDER

orderNum	orderDate	orderStatus	quantity	subTotal	grandTotal	deliveryStatus	username



FD1: $\text{orderNum} \rightarrow \text{orderDate, orderStatus, quantity, subTotal, grandTotal, deliveryStatus, username}$

FD1 is full functional dependency. Relation ORDER is in 3NF.

iii. **PAYMENT (transactionNum, bank, paymentDate, totalPrice, orderNum)**

PAYMENT

transactionNum	bank	paymentDate	totalPrice	orderNum



FD1: $\text{transactionNum} \rightarrow \text{bank, paymentDate, totalPrice, orderNum}$

FD1 is full functional dependency. Relation PAYMENT is in 3NF.

iv. **PRODUCT_BIKE_ACR** (productID, prodName, brand, color, price, stockQuantity, Btype, height, weight, shifter, frame, tyre, chainwheel, Atype, size)

1NF

PRODUCT_BIKE_ACR

productID	prodName	brand	color	price	stockQuantity	Btype	height	weight	shifter	frame	tyre	chainwheel	Atype	size
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	FD1
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑

FD1: productID, Btype → prodName, brand, color, price, stockQuantity, height, weight, shifter, frame, tyre, chainwheel (*partial dependency*)

FD2: productID, Atype → prodName, brand, color, price, stockQuantity, size (*partial dependency*)

2NF

BIKE (productID, Btype, prodName, brand, color, price, stockQuantity, height, weight, shifter, frame, tyre, chainwheel)

ACCESSORIES (productID, Atype, prodName, brand, color, price, stockQuantity, size)

BIKE

productID	prodName	brand	color	price	stockQuantity	Btype	height	weight	shifter	frame	tyre	chainwheel
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑

ACCESSORIES

productID	prodName	brand	color	price	stockQuantity	Atype	size
↑	↑	↑	↑	↑	↑	↑	↑

Since there is no transitive dependency (TD) in both BIKE and ACCESSORIES relation, therefore both relations are in 3NF.

v. ORDER_PRODUCT (**id**, quantity, totalPrice, orderNum, productID)

ORDER_PRODUCT

<u>id</u>	<u>quantity</u>	<u>totalPrice</u>	<u>orderNum</u>	<u>productID</u>
				↑ FD1

FD1: **id** → quantity, totalPrice, orderNum, productID

FD1 is full functional dependency. Relation ORDER_PRODUCT is in 3NF.

These are our final of normalized relation in 3NF:

- i. CUSTOMER (username, password, fName, lName, street, city, postcode)
 - ii. ORDER (orderNum, orderDate, orderStatus, quantity, subtotal, grandTotal, deliveryStatus, username)
 - iii. PAYMENT (transactionNum, bank, paymentDate, totalPrice, orderNum)
 - iv. BIKE (productID, Btype, prodName, brand, model, color, price, stockQuantity, height, weight, shifter, frame, tyre, chainwheel)
 - v. ACCESSORIES (productID, Atype, prodName, brand, model, color, price, stockQuantity, size)
 - vi. ORDER_PRODUCT (id, quantity, totalPrice, orderNum, productID)

4.3 Data Dictionary for Logical Database Design

Entity Name	Attributes	Description	Data Type & Length	Nulls	Multi-valued
CUSTOMER	username	Uniquely identifies a customer	20 variable characters	No	No
	password	Password for the username	30 variable characters	No	No
	fname	First name of customer	20 variable characters	No	No
	lname	Last name of customer	20 variable characters	No	No
	street	Street address of customer	30 variable characters	No	No
	city	City address of customer	20 variable characters	No	No
	postcode	Postcode address of customer	Integer	No	No
	email	Email address of customer	50 variable characters	No	No
ORDER	orderNum	Uniquely identifies an order	10 variable characters	No	No
	orderDate	Order date of an order	Date	No	No
	orderStatus	Order status of an order	15 variable characters	No	No
	quantity	Quantity of item in an order	Integer	No	No
	subtotal	The sum of item price	Decimal	No	No
	grandtotal	The total of subtotal plus delivery fee	Decimal	No	No
	deliveryStatus	Delivery Status	15 variable characters	No	No
	username	Foreign key references CUSTOMER (username)	20 variable characters	No	No
PAYMENT	transactionNum	Uniquely identifies a payment	30 variable characters	No	No

	bank paymentDate totalPrice orderNum	Bank that use by customer for payment Date of the payment made The total price of the payment Foreign key references ORDER (orderNum)	30 variable characters Date Decimal 10 variable characters	No No No No	No No No No
BIKE	productID	Uniquely identifies a bike	20 variable characters	No	No
	Btype	Type of bike	20 variable characters	No	No
	prodName	Name of bike	100 variable characters	No	No
	brand	Brand of bike	50 variable characters	No	No
	Bmodel	Model of bike	50 variable characters	No	No
	color	Color of bike	20 variable characters	No	No
	price	Price of bike	Decimal	No	No
	stockQuantity	Number of bike available	Integer	No	No
	height	Height of bike	Float	Yes	No
	weight	Weight of bike	Float	Yes	No
	shifter	Type of shifter and speed of bike	30 variable characters	Yes	No
	frame	Frame type of bike	30 variable characters	Yes	No
	tyre	Size of tyre of bike	30 variable characters	Yes	No
	chainWheel	Length and tyre of chain wheel of bike	30 variable characters	Yes	No
ACCESSORIES	productID Atype prodName	Uniquely identifies a accessories Type of accessories Name of accessories	20 variable characters 20 variable characters 100 variable characters	No No No	No No No

	brand aModel color price stockQuantity aSize	Brand of accessories Model of accessories Color of accessories Price of accessories Number of accessories available Size of accessories	50 variable characters 50 variable characters 20 variable characters Decimal Integer 50 variable characters	No No No No No Yes	No No No No No No
ORDER_PRODUCT	op_id quantity totalPrice orderNum productID	Uniquely identifies an order_product Quantity of item customer purchase Total price of the item Foreign key references ORDER (orderNum) Foreign key references BIKE (productID), Foreign key references ACCESSORIES (productID)	20 variable characters Integer Decimal 10 variable characters 20 variable characters	No No No No No	No No No No No

5.0 Project Implementation

5.1 Structured Query Language

```
--CREATE TABLE FOR OUR DATABASE-----
CREATE TABLE CUSTOMER_(USERNAME VARCHAR(20) PRIMARY KEY,
PWORD VARCHAR2(30) NOT NULL,
FNAME VARCHAR2(30) NOT NULL,
LNAME VARCHAR2(30) NOT NULL,
STREET VARCHAR2(30) NOT NULL,
CITY VARCHAR2(20) NOT NULL,
POSTCODE INT NOT NULL);

CREATE TABLE ORDER_(ORDERNUM VARCHAR2(10) PRIMARY KEY,
ORDERDATE DATE NOT NULL,
ORDERSTATUS VARCHAR2(15) NOT NULL,
QUANTITY INT NOT NULL,
SUBTOTAL DECIMAL (8,2) NOT NULL,
GRANDTOTAL DECIMAL (10,2) NOT NULL,
DELIVERYSTATUS VARCHAR2(15) NOT NULL,
USERNAME VARCHAR2(20) NOT NULL,
FOREIGN KEY(USERNAME) REFERENCES CUSTOMER_(USERNAME));

CREATE TABLE ORDER_(ORDERNUM VARCHAR2(10) PRIMARY KEY,
ORDERDATE DATE NOT NULL,
ORDERSTATUS VARCHAR2(15) NOT NULL,
QUANTITY INT NOT NULL,
SUBTOTAL DECIMAL (8,2) NOT NULL,
GRANDTOTAL DECIMAL (10,2) NOT NULL,
DELIVERYSTATUS VARCHAR2(15) NOT NULL,
USERNAME VARCHAR2(20) NOT NULL,
FOREIGN KEY(USERNAME) REFERENCES CUSTOMER_(USERNAME));

CREATE TABLE PAYMENT_(TRANSACTIONNUM VARCHAR2(30) PRIMARY KEY,
BANK VARCHAR2(30) NOT NULL,
PAYMENTDATE DATE NOT NULL,
TOTALPRICE DECIMAL (10,2) NOT NULL,
ORDERNUM VARCHAR2(10),
FOREIGN KEY (ORDERNUM) REFERENCES ORDER_(ORDERNUM));

CREATE TABLE BIKE_(BPRODUCTID VARCHAR2(20) PRIMARY KEY,
BTYPE VARCHAR2(20) NOT NULL,
PRODNAME VARCHAR2 (100) NOT NULL,
BRAND VARCHAR2(50),
BMODEL VARCHAR2(50),
COLOR VARCHAR2(20) NOT NULL,
PRICE DECIMAL (8,2) NOT NULL,
STOCKQUANTITY INT NOT NULL,
HEIGHT FLOAT,
WEIGHT FLOAT,
SHIFTER VARCHAR2(30),
FRAME VARCHAR2(30),
TYRE VARCHAR2(30),
CHAINWHEEL VARCHAR2(30));
```

```

CREATE TABLE ACCESSORIES_( APRODUCTID VARCHAR2(20) PRIMARY KEY,
ATYPE VARCHAR2(20) NOT NULL,
PRODNAME VARCHAR2(100) NOT NULL,
BRAND VARCHAR2(50),
AMODEL VARCHAR2(50),
COLOR VARCHAR2(20) NOT NULL,
PRICE DECIMAL(8,2) NOT NULL,
STOCKQUANTITY INT NOT NULL,
ASIZE VARCHAR2 (50));

CREATE TABLE ORDER_PRODUCT ( OP_ID VARCHAR2(20) PRIMARY KEY,
QUANTITY INT,
TOTALPRICE DECIMAL (8,2),
ORDERNUM VARCHAR2(10),
FOREIGN KEY (ORDERNUM) REFERENCES ORDER_(ORDERNUM),
BPRODUCTID VARCHAR2(20),
FOREIGN KEY (BPRODUCTID) REFERENCES BIKE_(BPRODUCTID),
APRODUCTID VARCHAR2(20),
FOREIGN KEY (APRODUCTID) REFERENCES ACCESSORIES_(APRODUCTID));

-----MODIFY THE TABLE-----
ALTER TABLE CUSTOMER_
ADD EMAIL VARCHAR2(50);

ALTER TABLE ORDER_
DROP COLUMN DELIVERYSTATUS;

ALTER TABLE ORDER_
MODIFY ORDERSTATUS VARCHAR2(50);

ALTER TABLE ORDER_
ADD DELIVERYDATE DATE;

ALTER TABLE ORDER_
DROP COLUMN QUANTITY;

-----ADD PRODUCT(STAFF)-----
INSERT INTO BIKE_
VALUES ('B001','MOUNTAIN','CITIZEN BIKE FOR HONEST TEA 166-SPEED FOLDING BIKE',
'TOKYO', 'XTC-600 ADVANCE','GREEN',399, 6, 137,13,
'6-SPEED INDEX GEAR SYSTEM', '16" STEEL FOLDABLE FRAME', '16"',
'170MM ALLOY CRANK ARMS');

INSERT INTO BIKE_
VALUES ('B002','BMX','ALLOY RIM BICYCLE WITH MTB HANDLE BAR',
'MARSSTAR', '2001 WARRIOR','BLACK',186, 3, 130,15,
'3X7 SPEED SHIFTER', '15" STEEL', '20"',
'165MM F/R ALLOY');

INSERT INTO BIKE_
VALUES ('B003','ROAD','700C ALLOY ROAD BIKE',
'TOTEM', 'T15B CARINA','BLACK',799, 2, 138,16,
'EF51-3L-7R', '700C*400 ROAD', '20"',
'168MM F/R ALLOY');

```

```

INSERT INTO ACCESSORIES_
VALUES ('A001', 'BIKE BELL', 'ALUMINIUM ALLOY BIKE BELL','SHIMANO',
'TOMTOP-Y0376', 'RED', 5, 10, '5cmx5cmx6.34cm');

INSERT INTO ACCESSORIES_
VALUES ('A002', 'BIKE BOTTLE HOLDER', 'WATER BOTTLE HOLDER RACK CAGE','MTB',
'YMST4534M64VOI', 'BLACK', 17.10, 5, '8.5cmx14cmx15cm');

INSERT INTO ACCESSORIES_
VALUES ('A003', 'BIKE HELMET', 'BICYCLE HELMET WITH ADJUSTABLE VISOR','GIANT',
'GIANTHM-ORANGE', 'BLUE', 44.50, 2, '54cm to 64cm');

```

-----MODIFY PRODUCT(STAFF)-----

```

UPDATE BIKE_
SET PRICE= 399
WHERE BPRODUCTID = 'B001';

```

-----DELETE PRODUCT(STAFF)-----

```

DELETE FROM BIKE_
WHERE BPRODUCTID='B001';

```



-----SIGN UP (CUSTOMER)-----

```

INSERT INTO CUSTOMER_
VALUES ('A16CS0098', 123456, 'KAY','MARY','JALAN RESAK','UTM',81310);

```

```

UPDATE CUSTOMER_
SET CITY='JOHOR BAHRU'
WHERE CITY='UTM';

```

```

INSERT INTO CUSTOMER_
VALUES ('A16CS0234', 567889, 'JOHN','CENA','JALAN MERANTI','JOHOR BAHRU',81310);

```

```

INSERT INTO CUSTOMER_
VALUES ('A16KA0014', 678453, 'BRUNO','MARS','JALAN CENGAL','JOHOR BAHRU',81310);

```

-----UPDATE CUSTOMER INFO(CUSTOMER)-----

```

UPDATE CUSTOMER_
SET EMAIL='kay@gmail.com'
WHERE FNAME='KAY';

```

```

UPDATE CUSTOMER_
SET EMAIL='john234@gmail.com'
WHERE FNAME='JOHN';

```

```

UPDATE CUSTOMER_
SET EMAIL='bm14@gmail.com'
WHERE FNAME='BRUNO';

```

-----QUERY BIKE(CUSTOMER)-----

```

SELECT BTTYPE, PRODNAME, PRICE, BRAND FROM BIKE_;

```

```

SELECT BTTYPE, PRODNAME, PRICE, BRAND FROM BIKE_
WHERE BTTYPE='MOUNTAIN';

```

```

-----ADD ITEM INTO ORDER(CUSTOMER)-----
INSERT INTO ORDER_PRODUCT
VALUES ('0001', 1, NULL, '112233', NULL, NULL);

UPDATE ORDER_PRODUCT
SET (TOTALPRICE, BPRODUCTID) = (SELECT PRICE, BPRODUCTID FROM BIKE_
WHERE BPRODUCTID='B001')
WHERE OP_ID='0001';

INSERT INTO ORDER_PRODUCT
VALUES ('0002', 1, NULL, '112234', NULL, NULL);

UPDATE ORDER_PRODUCT
SET (TOTALPRICE, APRODUCTID) = (SELECT PRICE, APRODUCTID FROM ACCESSORIES_
WHERE APRODUCTID='A001')
WHERE OP_ID='0002';

INSERT INTO ORDER_PRODUCT
VALUES ('0003', 1, NULL, '112235', NULL, NULL);

UPDATE ORDER_PRODUCT
SET (TOTALPRICE, BPRODUCTID) = (SELECT PRICE, BPRODUCTID FROM BIKE_
WHERE BPRODUCTID='B002')
WHERE OP_ID='0003';

INSERT INTO ORDER_PRODUCT
VALUES ('0004', 1, NULL, '112235', NULL, NULL);

UPDATE ORDER_PRODUCT
SET (TOTALPRICE, APRODUCTID) = (SELECT PRICE, APRODUCTID FROM ACCESSORIES_
WHERE APRODUCTID='A002')
WHERE OP_ID='0004';

INSERT INTO ORDER_PRODUCT
VALUES ('0005', 1, NULL, '112236', NULL, NULL);

UPDATE ORDER_PRODUCT
SET (TOTALPRICE, APRODUCTID) = (SELECT PRICE, APRODUCTID FROM ACCESSORIES_
WHERE APRODUCTID='A002')
WHERE OP_ID='0005';

-----SHOW CURRENT ORDER-----
SELECT * FROM ORDER_PRODUCT
WHERE ORDERNUM='112235';

-----CHOOSE QUANTITY (CUSTOMER)-----
UPDATE ORDER_PRODUCT
SET QUANTITY=2
WHERE APRODUCTID='A001';

```

```

-----DELETE ITEM FROM ORDER(CUSTOMER)-----
DELETE FROM ORDER_PRODUCT
WHERE OP_ID='0003';

-----ORDER->SUBTOTAL-----
INSERT INTO ORDER_
VALUES ('112233', SYSDATE, 'ORDER RECEIVED', NULL, 0, 'A16CS0098');

UPDATE ORDER_
SET SUBTOTAL = (SELECT QUANTITY * TOTALPRICE FROM order_product WHERE OP_ID='0001')
WHERE ordernum = '112233';

INSERT INTO ORDER_
VALUES ('112234', SYSDATE, 'READY TO DELIVER', 0, 0, 'NO', 'A16CS0234');

UPDATE ORDER_
SET SUBTOTAL = (SELECT QUANTITY * TOTALPRICE FROM order_product WHERE OP_ID='0002')
WHERE ordernum = '112234';

INSERT INTO ORDER_
VALUES ('112235', SYSDATE, 'CONFIRM DELIVERED', 203.10, 208.10, 'A16KA0014');

UPDATE ORDER_
SET SUBTOTAL = (SELECT QUANTITY * TOTALPRICE FROM order_product WHERE OP_ID='0003')
WHERE ordernum = '112235';

UPDATE ORDER_
SET SUBTOTAL = SUBTOTAL + (SELECT QUANTITY * TOTALPRICE FROM order_product WHERE OP_ID='0004')
WHERE ordernum = '112235';

INSERT INTO ORDER_
VALUES ('112236', SYSDATE, 'ORDER RECEIVED', 0, 0, 'A16CS0098', NULL);

UPDATE ORDER_
SET SUBTOTAL = (SELECT QUANTITY * TOTALPRICE FROM order_product WHERE OP_ID='0005')
WHERE ordernum = '112236';

-----ORDER->GRANDTOTAL-----
UPDATE ORDER_
SET GRANDTOTAL = 5 + SUBTOTAL;

-----PAYMENT-----
INSERT INTO PAYMENT_
VALUES ('CIMB00001','CIMB',SYSDATE, (SELECT GRANDTOTAL FROM ORDER_
WHERE ORDERNUM='112233'), '112233');

INSERT INTO PAYMENT_
VALUES ('CIMB00002','CIMB',SYSDATE, (SELECT GRANDTOTAL FROM ORDER_
WHERE ORDERNUM='112234'), '112234');

INSERT INTO PAYMENT_
VALUES ('CIMB00003','CIMB',SYSDATE, (SELECT GRANDTOTAL FROM ORDER_
WHERE ORDERNUM='112235'), '112235');

```

```
INSERT INTO PAYMENT_
VALUES ('MAYBANK00001','MAYBANK',SYSDATE, (SELECT GRANDTOTAL FROM ORDER_
WHERE ORDERNUM='112236'), '112236');
```

-----3. CONFIRMATION (RECEIPT)-----

```
SELECT ORDER_.ORDERNUM, ORDER_.ORDERDATE, ORDER_.SUBTOTAL, ORDER_.GRANDTOTAL,
ORDER_PRODUCT.BPRODUCTID, BIKE_.PRODNAME
FROM ORDER_, ORDER_PRODUCT, BIKE_
WHERE ORDER_.ORDERNUM='112233' AND ORDER_PRODUCT.ORDERNUM='112233'
AND BIKE_.BPRODUCTID='B001';
```

```
SELECT ORDER_.ORDERNUM, ORDER_.ORDERDATE, ORDER_.SUBTOTAL, ORDER_.GRANDTOTAL,
ORDER_PRODUCT.APRODUCTID, ACCESSORIES_.PRODNAME
FROM ORDER_, ORDER_PRODUCT, ACCESSORIES_
WHERE ORDER_.ORDERNUM='112234' AND ORDER_PRODUCT.ORDERNUM='112234'
AND ACCESSORIES_.APRODUCTID='A001';
```

-----MY ORDER-----

```
SELECT ORDER_.ORDERNUM, ORDER_.ORDERSTATUS, BIKE_.PRODNAME, ACCESSORIES_.PRODNAME
FROM ORDER_
FULL JOIN ORDER_PRODUCT
ON ORDER_.ORDERNUM = ORDER_PRODUCT.ORDERNUM
FULL JOIN BIKE_
ON ORDER_PRODUCT.bproductid = BIKE_.bproductid
FULL JOIN ACCESSORIES_
ON ORDER_PRODUCT.APRODUCTID = ACCESSORIES_.APRODUCTID
WHERE ORDER_.USERNAME = 'A16CS0098';
```

-----UPDATE ORDER STATUS(STAFF)-----

```
UPDATE ORDER_
SET ORDERSTATUS = 'READY TO DELIVER'
WHERE ORDERNUM = '112233';
```

-----SET DELIVERY DATE(STAFF)-----

```
UPDATE ORDER_
SET DELIVERYDATE='18-DEC-17'
WHERE ORDERNUM='112234';
```

-----PRINT DELIVERY STATEMENT(STAFF)-----

```
SELECT CUSTOMER_.FNAME, CUSTOMER_.EMAIL, CUSTOMER_.STREET, CUSTOMER_.CITY, CUSTOMER_.POSTCODE,
ORDER_.DELIVERYDATE, BIKE_.PRODNAME, ACCESSORIES_.PRODNAME,
ORDER_PRODUCT.QUANTITY, ORDER_PRODUCT.TOTALPRICE, PAYMENT_.TOTALPRICE
FROM CUSTOMER_
FULL JOIN ORDER_
ON CUSTOMER_.USERNAME = ORDER_.USERNAME
FULL JOIN PAYMENT_
ON ORDER_.ORDERNUM = PAYMENT_.ORDERNUM
FULL JOIN ORDER_PRODUCT
ON ORDER_.ORDERNUM = ORDER_PRODUCT.ORDERNUM
FULL JOIN BIKE_
ON ORDER_PRODUCT.BPRODUCTID = BIKE_.BPRODUCTID
FULL JOIN ACCESSORIES_
ON ORDER_PRODUCT.APRODUCTID = ACCESSORIES_.APRODUCTID
WHERE CUSTOMER_.FNAME = 'JOHN';
```

```
-----VIEW PAYMENT DETAIL(STAFF)-----
SELECT CUSTOMER_.EMAIL, ORDER_.ORDERNUM, PAYMENT_.TRANSACTIONNUM, PAYMENT_.BANK,
PAYMENT_.PAYMENTDATE, PAYMENT_.TOTALPRICE, ORDER_.ORDERSTATUS
FROM CUSTOMER_
FULL JOIN ORDER_
ON CUSTOMER_.USERNAME = ORDER_.USERNAME
FULL JOIN PAYMENT_
ON PAYMENT_.ORDERNUM = ORDER_.ORDERNUM
WHERE CUSTOMER_.FNAME = 'BRUNO';

-----UPDATE STOCK QUANTITY-----
UPDATE BIKE_
SET STOCKQUANTITY = STOCKQUANTITY - 1
WHERE BPRODUCTID = 'B001';

UPDATE ACCESSORIES_
SET STOCKQUANTITY = STOCKQUANTITY - 1
WHERE aproductid = 'A001';
```

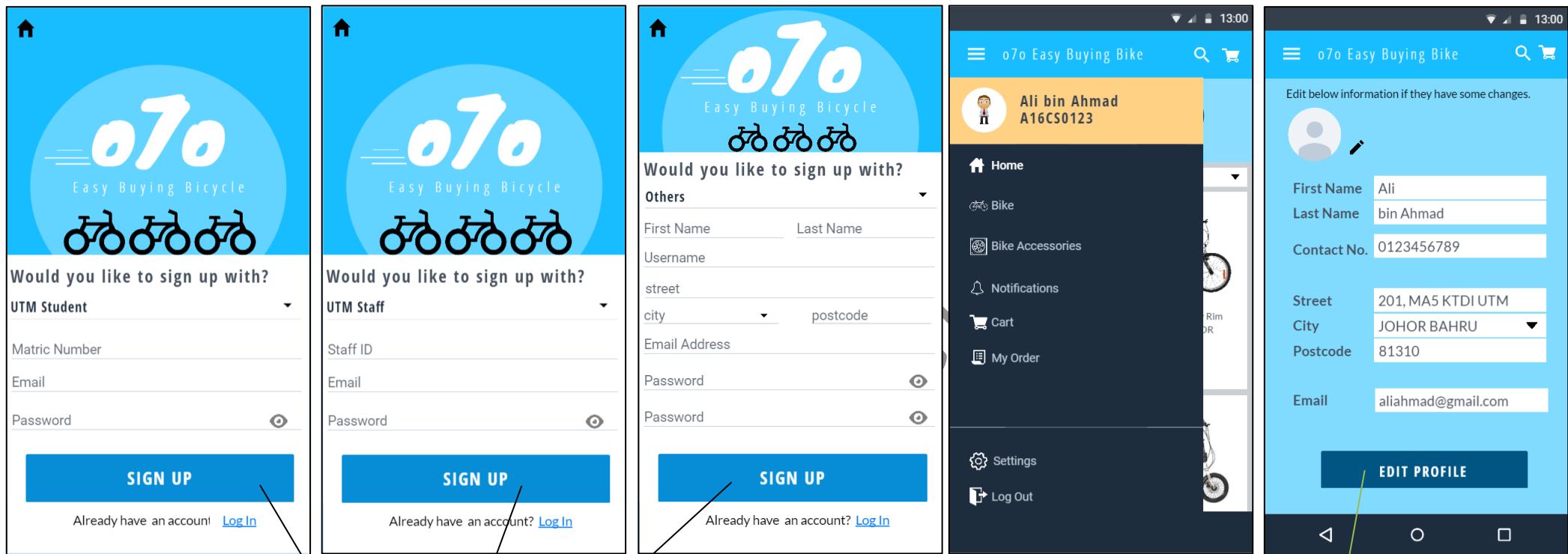
DATABASE PROJECT SAMPLE

5.2 System Prototype Interface

The image displays four sequential screenshots of a mobile application's login screen, illustrating the user authentication process. The background is blue with the 'oTo' logo and the text 'Easy Buying Bicycle'. A circular icon shows two people on bicycles.

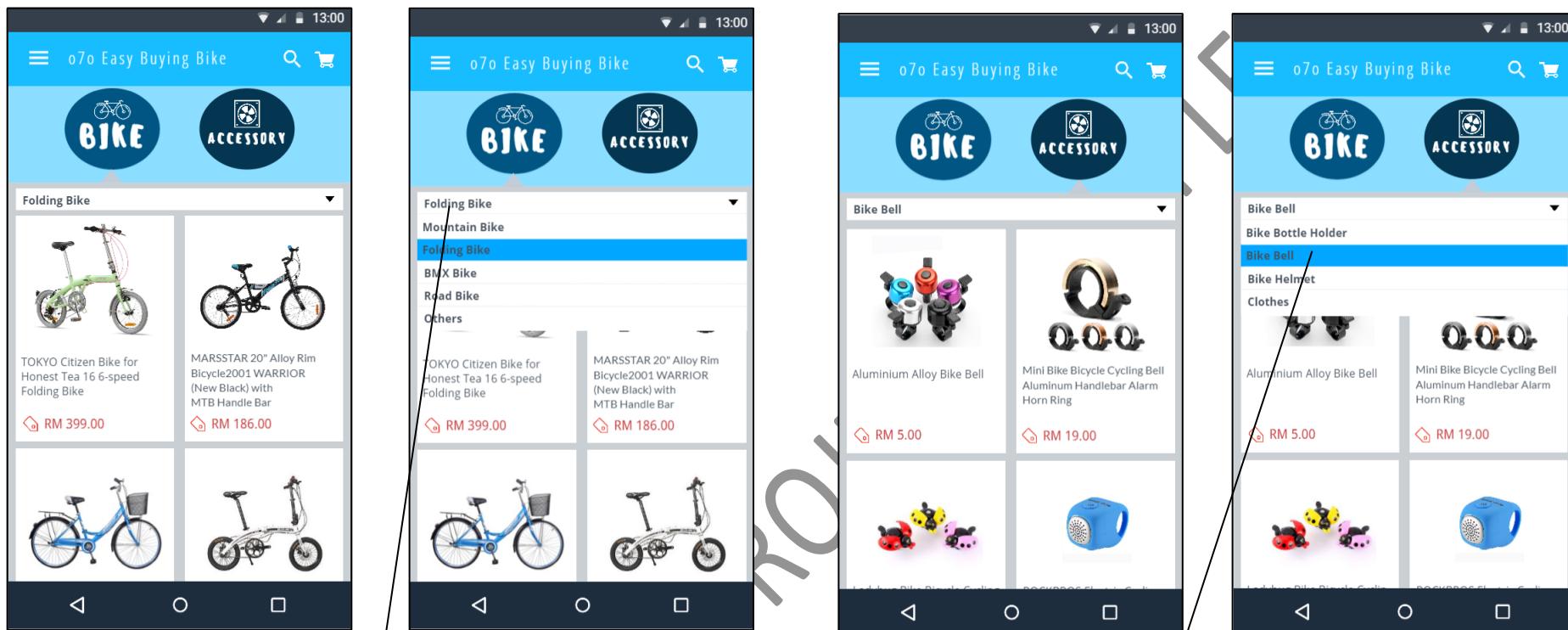
- Screenshot 1:** Shows a large 'Home' button at the top left and 'Log in' and 'Sign up' buttons at the bottom.
- Screenshot 2:** Shows a dropdown menu for 'Would you like to login with?' containing 'UTM Student'. Below it are fields for 'Matric Number' and 'Password', and a 'LOG IN' button at the bottom.
- Screenshot 3:** Shows a dropdown menu for 'Would you like to login with?' containing 'UTM Staff'. Below it are fields for 'Staff ID' and 'Password', and a 'LOG IN' button at the bottom.
- Screenshot 4:** Shows a dropdown menu for 'Would you like to login with?' containing 'Others'. Below it are fields for 'username' and 'Password', and a 'LOG IN' button at the bottom.

Each screenshot includes a 'No a member? [Sign up now](#)' link at the bottom of the form area.



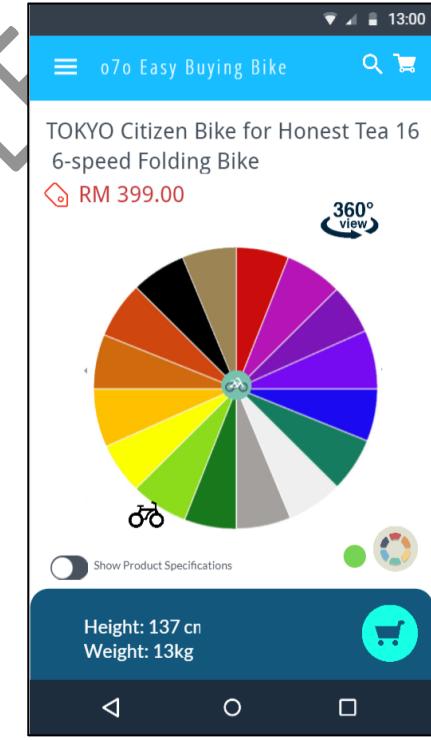
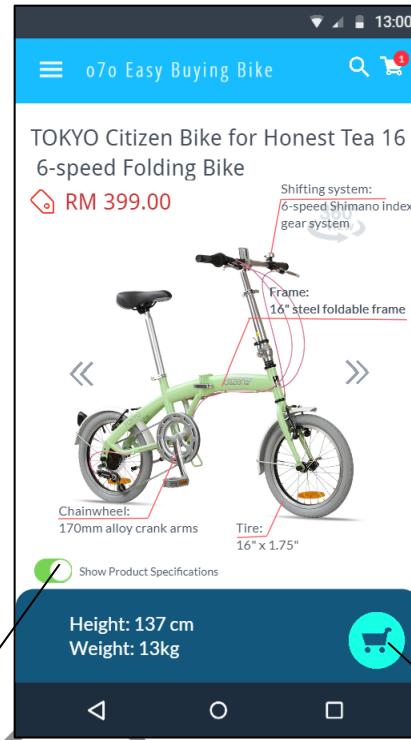
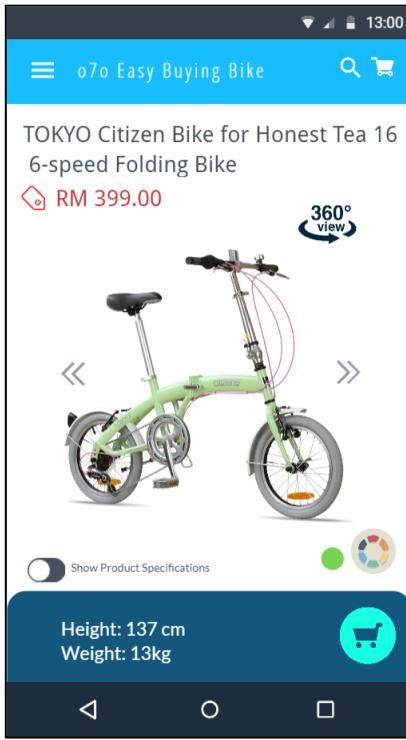
**UPDATE CUSTOMER_
SET CITY='JOHOR BAHRU'
WHERE CITY='UTM';**

INSERT INTO CUSTOMER_
VALUES ('A16CS0123', 123456, 'Ali','bin Ahmad','201, MA5 KTDI UTM','JOHOR BAHRU',81310, 'aliahamd@gmail.com');
** Note: For Students and staffs, once they key in matric number or staff id, their information will be taken from UTM system.



**SELECT BTYPE, PRODNAME, PRICE, BRAND FROM BIKE
WHERE BTYPE='FOLDING';**

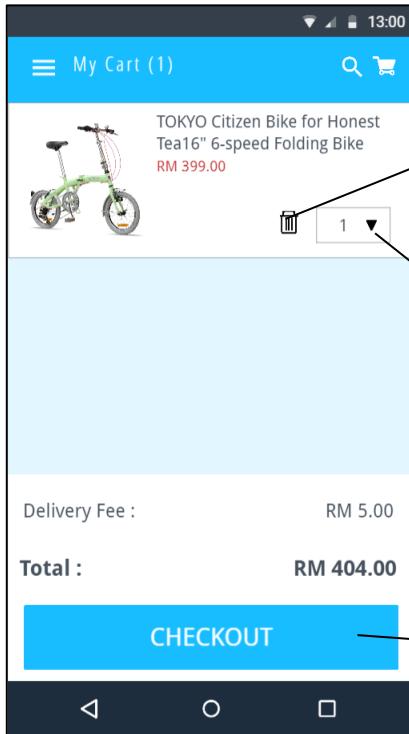
**SELECT ATYPE, PRODNAME, PRICE, BRAND FROM ACCESSORIES_
WHERE ATYPE='BIKE BELL';**



**SELECT BTYPE, PRODNAME, PRICE, BRAND, COLOR, HEIGHT, WEIGHT,
SHIFTER, FRAME, TYRE, CHAINWHEEL FROM BIKE_
WHERE PRODNAME='CITIZEN BIKE FOR HONEST TEA 166-SPEED FOLDING
BIKE';**

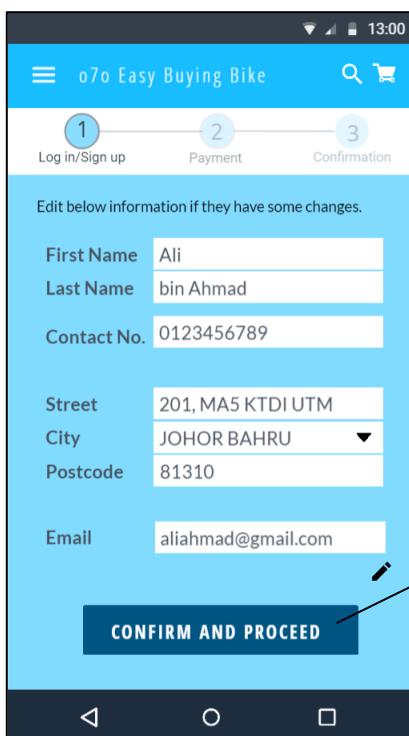
**INSERT INTO ORDER_PRODUCT
VALUES ('0001', 1, NULL, '112233', NULL, NULL);**

**UPDATE ORDER_PRODUCT
SET (TOTALPRICE, BPRODUCTID) = (SELECT PRICE, BPRODUCTID FROM BIKE_
WHERE BPRODUCTID='B001')
WHERE OP_ID='0001';**



```
DELETE FROM ORDER_PRODUCT  
WHERE OP_ID='0001';
```

```
UPDATE ORDER_PRODUCT  
SET QUANTITY=1  
WHERE BPRODUCTID='B001';
```

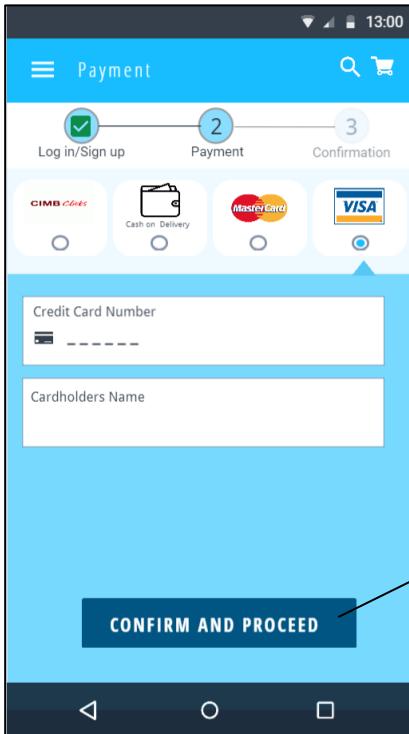


```
INSERT INTO ORDER_  
VALUES ('112233', SYSDATE, 'ORDER RECEIVED',  
NULL, 0, 'A16CS0098');
```

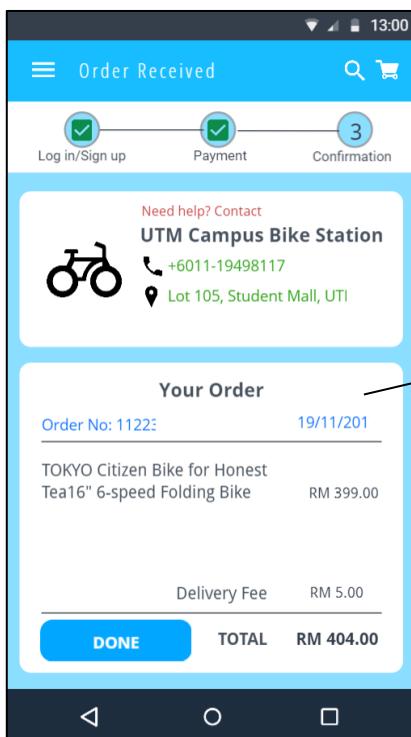
```
UPDATE ORDER_  
SET SUBTOTAL = (SELECT QUANTITY *  
TOTALPRICE FROM order_product WHERE  
OP_ID='0001')  
WHERE ordernum = '112233';
```

```
UPDATE ORDER_  
SET GRANDTOTAL = 5 + SUBTOTAL;
```

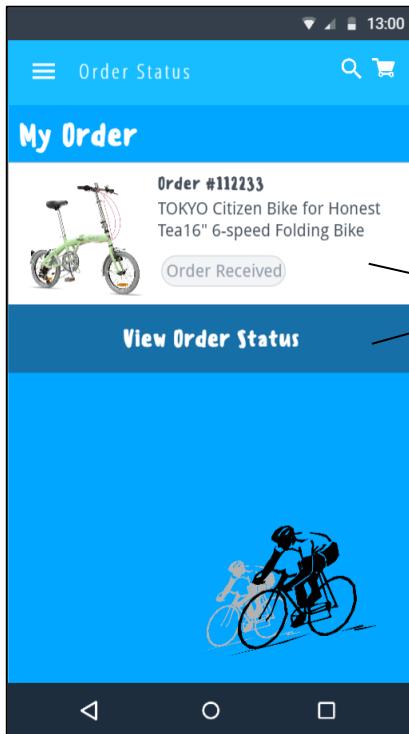
```
UPDATE CUSTOMER_  
SET CITY='JOHOR BAHRU'  
WHERE CITY='UTM';
```



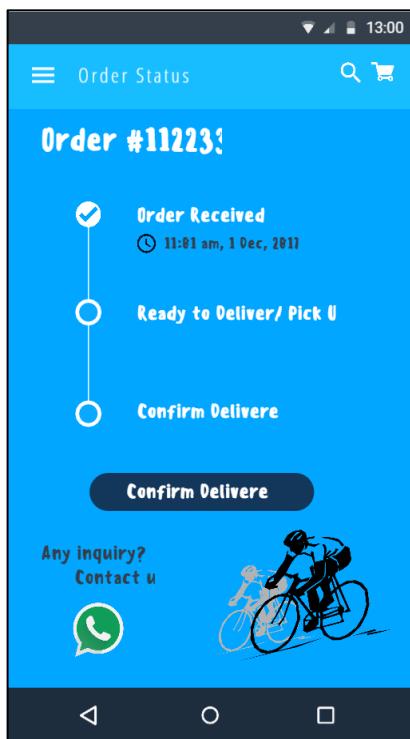
```
INSERT INTO PAYMENT_
VALUES ('CIMB00001','CIMB', SYSDATE, (SELECT
GRANDTOTAL FROM ORDER_
WHERE ORDERNUM='112233'), '112233');
```



```
SELECT ORDER_.ORDERNUM,
ORDER_.ORDERDATE, ORDER_.SUBTOTAL,
ORDER_.GRANDTOTAL,
ORDER_PRODUCT.BPRODUCTID,
BIKE_.PRODNAME
FROM ORDER_, ORDER_PRODUCT, BIKE_
WHERE ORDER_.ORDERNUM='112233' AND
ORDER_PRODUCT.ORDERNUM='112233'
AND BIKE_.BPRODUCTID='B001';
```



```
SELECT ORDER_.ORDERNUM,  
ORDER_.ORDERSTATUS, BIKE_.PRODNAME,  
ACCESSORIES_.PRODNAME  
FROM ORDER  
FULL JOIN ORDER_PRODUCT  
ON ORDER_.ORDERNUM =  
ORDER_PRODUCT.ORDERNUM  
FULL JOIN BIKE  
ON ORDER_PRODUCT.bproductid =  
BIKE_.bproductid  
FULL JOIN ACCESSORIES  
ON ORDER_PRODUCT.APRODUCTID =  
ACCESSORIES_.APRODUCTID  
WHERE ORDER_.USERNAME = 'A16CS0123';
```



The image displays two screenshots of a web-based product management application. The left screenshot shows a list of products with columns for Product ID, Name, Brand, Model, Color, Weight (kg), Quantity in Stock, Unit Price, and Action. The right screenshot shows a modal dialog for confirming the deletion of a specific product.

Product List (Left Screenshot):

Product ID	Name	Brand	Model	Color	Weight (kg)	Quantity in Stock	Unit Price	Action
20-03-RD	GoMax 7 Speed 20" Red	GoMax	GM-20-03	Red	11	1	299.00	
20-03-GR	GoMax 7 Speed 20" Green	GoMax	GM-20-03	Green	11	1	299.00	
20-03-BL	GoMax 7 Speed 20" Blue	GoMax	GM-20-03	Blue	11	1	299.00	
20-08-BK	TRS Folding Bike Black	TRS	TRS-20-08	Black	13	1	359.00	
20-08-BL	TRS Folding Bike Blue	TRS	TRS-20-08	Blue	13	0	359.00	

Showing 1 to 5 of 126 entries

Confirm Delete Dialog (Right Screenshot):

Confirm Delete: GoMax 7 Speed 20" Red

Product ID	Name	Unit Price	Action
20-03-RD	GoMax 7 Speed 20" Red	299.00	
20-03-GR	GoMax 7 Speed 20" Green	299.00	
20-03-BL	GoMax 7 Speed 20" Blue	299.00	
20-08-BK	TRS Folding Bike Black	359.00	
20-08-BL	TRS Folding Bike Blue	359.00	

Showing 1 to 5 of 126 entries

**DELETE FROM BIKE
WHERE BPRODUCTID='B001';**

ojo ADD Product Information

Back Search Database

Product ID
Product Name
Brand
Model
Color
Choose Category of Product BIKE Choose Bike Type
Height (cm) Weight (kg) Tyre Size ()
Shifter
Frame
Chain wheel

Add Product

The product has been added.

DONE

ACCESSORY Choose AccessoryType
Size

ojo ADD Product Information

Back Search Database

Product ID
Product Name
Brand
Model
Color
Choose Category of Product BIKE Choose Bike Type
Height (cm) Weight (kg) Tyre Size ()
Shifter
Frame
Chain wheel

Add Product

The product has been added.

DONE

ACCESSORY Choose AccessoryType
Size

INSERT INTO BIKE

```
VALUES ('B004','MOUNTAIN', 'ALLOY MTB MOUNTAIN BIKE BICYCLE', 'GARION', 'G2623-BC','GREEN', 865,  
2, 137, 13, '624 SPEED GEAR ', '16" STEEL FOLDABLE FRAME', '16"', '170MM ALLOY CRANK ARMS');
```

INSERT INTO ACCESSORIES

```
VALUES ('A002', 'BIKE BOTTLE HOLDER', 'WATER BOTTLE HOLDER RACK  
CAGE','MTB', 'YMST4534M64VOI', 'BLACK', 17.10, 5, '8.5cmx14cmx15cm');
```

oto Modify Product Information

Back Search Database

Product ID	20-03-RD
Product Name	GoMax 7 Speed 20" Red
Brand	GoM
Model	GM-20-03
Color	Red

The product has been modified.

DONE

Choose Category of Product

BIKE Road Bike

Height (cm) 137 Weight (kg) 11 Tyre Size () 16

Shifter 6-speed Shimano index gear system

Frame 16" steel foldable frame

Chain wheel 170mm alloy crank arms

ACCESSORY Choose AccessoryType

Size

Modify Product

oto Verify Customer Purchase

Back Search Database

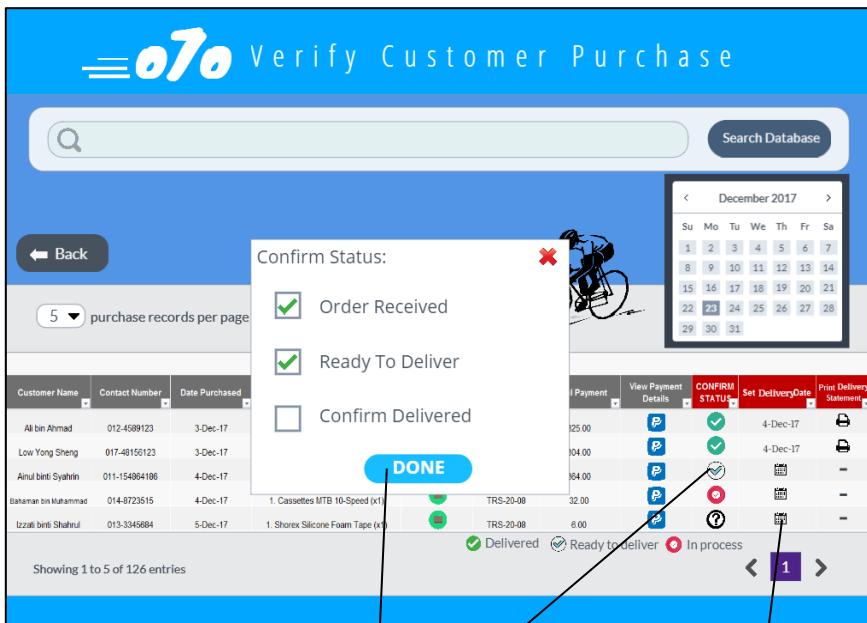
5 purchase records per page

December 2017

Customer Name	Contact Number	Date Purchased	Products Purchased	View Purchase Details	Model	Total Payment	View Payment Details	CONFIRM STATUS	Set Deliver Date	Print Deliver Statement
Ali bin Ahmad	012-4589123	3-Dec-17	1. GoMax 7 Speed 20" Red (x1) 2. N-Lite Water Bottle Holder (x1)		GM-20-03	325.00			4-Dec-17	
Lew Yong Sheng	017-48156123	3-Dec-17	1. GoMax 7 Speed 20" Green (x1)		GM-20-03	304.00			4-Dec-17	
Ainul Binti Syabirah	011-15484196	4-Dec-17	1. GoMax 7 Speed 20" Blue (x1)		GM-20-03	364.00			4-Dec-17	
Banarwan bin Matamad	014-8723515	4-Dec-17	1. Cassettes MTB 10-Speed (x1)		TRS-20-08	32.00			4-Dec-17	
Izzati Binti Shahrul	013-3345684	5-Dec-17	1. Shorex Silicone Foam Tape (x1)		TRS-20-08	6.00			4-Dec-17	

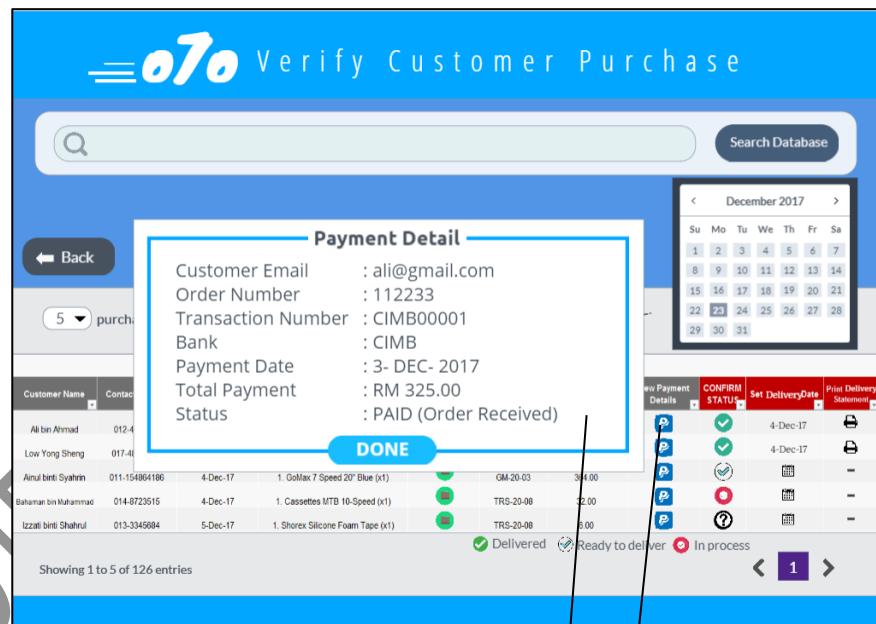
Showing 1 to 5 of 126 entries

**UPDATE BIKE
SET PRICE= 700
WHERE BPRODUCTID = '20-03-RD';**

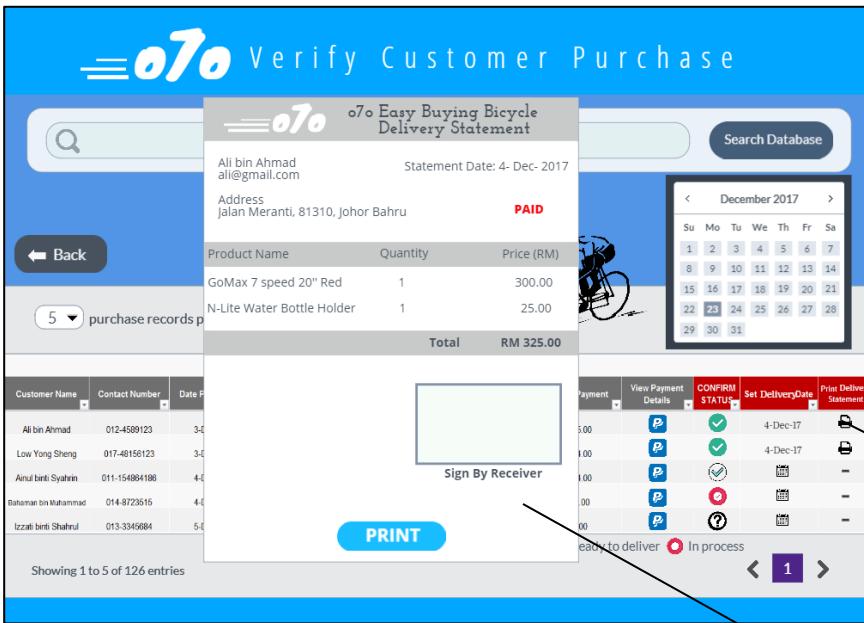


**UPDATE ORDER
SET ORDERSTATUS = 'READY TO DELIVER'
WHERE ORDERNUM = '112233';**

**UPDATE ORDER
SET DELIVERYDATE='18-DEC-17'
WHERE ORDERNUM='112237';**



**SELECT CUSTOMER_.EMAIL, ORDER_.ORDERNUM,
PAYMENT_.TRANSACTIONNUM, PAYMENT_.BANK,
PAYMENT_.PAYMENTDATE, PAYMENT_.TOTALPRICE,
ORDER_.ORDERSTATUS
FROM CUSTOMER_
FULL JOIN ORDER_
ON CUSTOMER_.USERNAME = ORDER_.USERNAME
FULL JOIN PAYMENT_
ON PAYMENT_.ORDERNUM = ORDER_.ORDERNUM
WHERE CUSTOMER_.FNAME = 'ALI';**



```

SELECT CUSTOMER_.FNAME, CUSTOMER_.EMAIL, CUSTOMER_.STREET,
CUSTOMER_.CITY, CUSTOMER_.POSTCODE,
ORDER_.DELIVERYDATE, BIKE_.PRODNAME, ACCESSORIES_.PRODNAME,
ORDER_PRODUCT.QUANTITY, ORDER_PRODUCT.TOTALPRICE,
PAYMENT_.TOTALPRICE
FROM CUSTOMER_
FULL JOIN ORDER_
ON CUSTOMER_.USERNAME = ORDER_.USERNAME
FULL JOIN PAYMENT_
ON ORDER_.ORDERNUM = PAYMENT_.ORDERNUM
FULL JOIN ORDER_PRODUCT
ON ORDER_.ORDERNUM = ORDER_PRODUCT.ORDERNUM
FULL JOIN BIKE_
ON ORDER_PRODUCT.BPRODUCTID = BIKE_.BPRODUCTID
FULL JOIN ACCESSORIES_
ON ORDER_PRODUCT.APRODUCTID = ACCESSORIES_.APRODUCTID
WHERE CUSTOMER_.FNAME = 'ALI';

```

5.3 Set of Queries for each transaction

-----QUERIES-----

```
SELECT * FROM CUSTOMER_;
SELECT * FROM BIKE_;
SELECT * FROM ACCESSORIES_;
SELECT * FROM ORDER_PRODUCT;
SELECT * FROM ORDER_;
SELECT * FROM PAYMENT_;
```

CUSTOMER_

USERNAME	PWORD	FNAME	LNAME	STREET	CITY	POSTCODE	EMAIL
A16CS0098	123456	KAY	MARY	JALAN RESAK	JOHOR BAHRU	81310	kay@gmail.com
A16CS0234	567889	JOHN	CENA	JALAN MERA...	JOHOR BAHRU	81310	john234@gmail.com
A16KA0014	678453	BRUNO	MARS	JALAN CENGAL	JOHOR BAHRU	81310	bm14@gmail.com

BIKE_

BPRODUCTID	BTYPE	PRODNAME	BRAND	BMODEL	COLOR	PRICE
B001	MOUNTAIN	CITIZEN BIKE FOR HONE...	TOKYO	XTC-600 ...	GREEN	399
B002	BMX	ALLOY RIM BICYCLE WIT...	MARSSTAR	2001 WAR...	BLACK	186
B003	ROAD	700C ALLOY ROAD BIKE	TOTEM	T15B CARI...	BLACK	799

STOCKQUANTITY	HEIGHT	WEIGHT	SHIFTER	FRAME	TYRE	CHA
6	137	13	6-SPEED INDEX GEAR S...	16" STEEL FOLDABLE FRAME	16"	170MM
3	130	15	3X7 SPEED SHIFTER	15" STEEL	20"	165MM
2	138	16	EF51-3L-7R	700C*400 ROAD	20"	168MM

ACCESSORIES_

APRODUCTID	ATYPE	PRODNAME	BRAND	AMODEL	COLOR	PRICE	STOCKQUANTITY
A001	BIKE BELL	ALUMINIUM A...	SHIMANO	TOMTOP-Y...	RED	5	8
A002	BIKE BOTTLE...	WATER BOTTL...	MTB	YMST4534...	BLACK	17.1	5
A003	BIKE HELMET	BICYCLE HELM...	GIANT	GIANTHM-...	BLUE	44.5	2

ASIZE
5cmx5cmx6....
8.5cmx14cm...
54cm to 64cm

ORDER_PRODUCT

OP_ID	QUANTITY	TOTALPRICE	ORDERNUM	BPRODUCTID	APRODUCTID
0001	1	399	112233	B001	(null)
0002	1	5	112234	(null)	A001
0003	1	186	112235	B002	(null)
0004	1	17.1	112235	(null)	A002
0005	1	17.1	112236	(null)	A002

ORDER_

ORDERNUM	ORDERDATE	ORDERSTATUS	SUBTOTAL	GRANDT...	USERNAME	DELIVERYDA
112236	20-DEC-17	ORDER RECEIVED	17.1	22.1	A16CS0098	(null)
112233	17-DEC-17	READY TO DELIVER	399	404	A16CS0098	(null)
112234	17-DEC-17	YES	5	10	A16CS0234	18-DEC-17
112235	17-DEC-17	YES	203.1	208.1	A16KA0014	(null)

PAYMENT_

TRANSACTIONNUM	BANK	PAYOUTDATE	TOTALPRICE	ORDERNUM
MAYBANK00001	MAYBANK	20-DEC-17	22.1	112236
CIMB00002	CIMB	20-DEC-17	10	112234
CIMB00001	CIMB	17-DEC-17	404	112233
CIMB00003	CIMB	17-DEC-17	208.1	112235

1. [2 TABLES] Display the order details of customer name JOHN

```
SELECT ORDERNUM, ORDERDATE, ORDERSTATUS, SUBTOTAL, GRANDTOTAL, DELIVERYDATE
FROM ORDER_ INNER JOIN CUSTOMER_
ON CUSTOMER_.USERNAME = ORDER_.USERNAME
WHERE CUSTOMER_.FNAME = 'JOHN';
```

Results:

ORDERNUM	ORDERDATE	ORDERSTATUS	SUBTOTAL	GRANDTOTAL	DELIVERYDATE
1 112234	17-DEC-17	YES	5	10	18-DEC-17

2. [2 TABLES] Display the payment details of order number = 112235

```
SELECT PAYMENT_.TRANSACTIONNUM, PAYMENT_.BANK, PAYMENT_.PAYOUTDATE, PAYMENT_.TOTALPRICE
FROM PAYMENT_ INNER JOIN ORDER_
ON PAYMENT_.ORDERNUM = ORDER_.ORDERNUM
WHERE ORDER_.ORDERNUM = '112235';
```

Results:

TRANSACTIONNUM	BANK	PAYOUTDATE	TOTALPRICE
1 CIMB00003	CIMB	17-DEC-17	208.1

3. [2 TABLES] Display the ordered product details in order number = 112233

```
SELECT ORDER_PRODUCT.OP_ID, ORDER_PRODUCT.QUANTITY, ORDER_PRODUCT.TOTALPRICE
FROM ORDER_PRODUCT INNER JOIN ORDER_
ON ORDER_PRODUCT.ORDERNUM = ORDER_.ORDERNUM
WHERE ORDER_.ORDERNUM = '112233';
```

Results:

OP_ID	QUANTITY	TOTALPRICE
1 0001	1	399

4. [3 TABLES] Display the product details of ordered product id = 0003

```
SELECT *
FROM ORDER_PRODUCT
FULL JOIN BIKE_
ON ORDER_PRODUCT.bproductid = BIKE_.bproductid
FULL JOIN ACCESSORIES_
ON ORDER_PRODUCT.aproductid = ACCESSORIES_.aproductid
WHERE op_id = '0003';
```

Results:

OP_ID	QUANTITY	TOTALPRICE	ORDERNUM	BPRODUCTID	APRODUCTID	BPRODUCTID_1	BTYPE	PRODNAME		
1 0003	1	186 112235	8002	(null)	8002	BMX	ALLOY RIM BICYCLE WITH			

APRODUCTID_1	ATYPE	PRODNAME_1	BRAND_1	AMODEL	COLOR_1	PRICE_1	STOCKQUANTITY_1	ASIZE
(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)

5. [2 TABLES] Display the order date of customer who lives in street

```
SELECT ORDER_.ORDERDATE
FROM ORDER_
INNER JOIN CUSTOMER_
ON ORDER_.USERNAME = CUSTOMER_.USERNAME
WHERE CUSTOMER_.STREET = 'JALAN MERANTI';
```

Results:

ORDERDATE
1 17-DEC-17

6. [3 TABLES] Display the payment date and price paid for customer with first name 'KAY'

```
SELECT PAYMENT_.PAYMENTDATE, PAYMENT_.TOTALPRICE
FROM PAYMENT_
INNER JOIN ORDER_
ON PAYMENT_.ORDERNUM = ORDER_.ORDERNUM
INNER JOIN CUSTOMER_
ON CUSTOMER_.USERNAME = ORDER_.USERNAME
WHERE CUSTOMER_.FNAME = 'KAY';
```

Results:

PAYOUTDATE	TOTALPRICE
1 20-DEC-17	22.1
2 17-DEC-17	404

7. [2 TABLES] Display the first name and order date of username A16CS0098

```
SELECT FNAME, ORDERDATE
FROM CUSTOMER_, order_
WHERE CUSTOMER_.USERNAME='A16CS0098' AND ORDER_.USERNAME='A16CS0098';
```

Results:

	FNAME	ORDERDATE
1	KAY	20-DEC-17
2	KAY	17-DEC-17

8. [4 TABLES] Display the product name of the product in order number = 112234

```
SELECT BIKE_.PRODNAME, ACCESSORIES_.PRODNAME
FROM ORDER_
FULL JOIN ORDER_PRODUCT
ON ORDER_.ORDERNUM = ORDER_PRODUCT.ORDERNUM
FULL JOIN BIKE_
ON ORDER_PRODUCT.BPRODUCTID = BIKE_.BPRODUCTID
FULL JOIN ACCESSORIES_
ON ORDER_PRODUCT.APRODUCTID = ACCESSORIES_.APRODUCTID
WHERE ORDER_.ORDERNUM = '112234';
```

Results:

PRODNAME	PRODNAME_1
1 (null)	ALUMINIUM ALLOY BIKE BELL

9. [1 TABLE] Display the payment details of price more than 100

```
SELECT *
FROM PAYMENT_
WHERE totalprice > 100;
```

Results:

TRANSACTIONNUM	BANK	PAYMENTDATE	TOTALPRICE	ORDERNUM
1 CIMB00001	CIMB	17-DEC-17	404	112233
2 CIMB00003	CIMB	17-DEC-17	208.1	112235

10. [1 TABLE] Display the bike product name ascendingly based on its stock quantity

```
SELECT BIKE_.PRODNAME, BIKE_.STOCKQUANTITY
FROM BIKE_
ORDER BY STOCKQUANTITY;
```

Results:

PRODNAME	STOCKQUANTITY
1 700C ALLOY ROAD BIKE	2
2 ALLOY RIM BICYCLE WITH MTB HANDLE BAR	3
3 CITIZEN BIKE FOR HONEST TEA 166-SPEED FOLDING BIKE	6

11. [2 TABLES] Display the number of bike that have been ordered

```
SELECT COUNT(ORDER_PRODUCT.BPRODUCTID)
FROM BIKE_ INNER JOIN ORDER_PRODUCT
ON ORDER_PRODUCT.Bproductid = BIKE_.Bproductid;
```

12. [1 TABLE] Display the number of the order based on customers username

```
SELECT USERNAME, COUNT(USERNAME)
FROM ORDER_
GROUP BY username;
```

Results:

USERNAME	COUNT(USERNAME)
1 A16CS0098	2
2 A16CS0234	1
3 A16KA0014	1

13. [2 TABLES] Calculate the total payment made by customer username A16CS0098

```
SELECT SUM(PAYMENT_.TOTALPRICE)
FROM PAYMENT_
INNER JOIN ORDER_
ON ORDER_.ORDERNUM = PAYMENT_.ORDERNUM
WHERE ORDER_.USERNAME = 'A16CS0098';
```

Results:

SUM(PAYMENT_.TOTALPRICE)
426.1

14. [1 TABLE] Display the maximum price in bike

```
SELECT MAX(PRICE)
FROM BIKE_;
```

Results:

MAX(PRICE)
799

15. [6 TABLES] Display the payment price and product name bought by customer last name ‘CENA’

```
SELECT CUSTOMER_.LNAME, CUSTOMER_.EMAIL, CUSTOMER_.STREET, CUSTOMER_.CITY, CUSTOMER_.POSTCODE,
ORDER_.DELIVERYDATE, BIKE_.PRODNAME, ACCESSORIES_.PRODNAME,
ORDER_PRODUCT.QUANTITY, ORDER_PRODUCT.TOTALPRICE, PAYMENT_.TOTALPRICE
FROM CUSTOMER_
FULL JOIN ORDER_
    ON CUSTOMER_.USERNAME = ORDER_.USERNAME
FULL JOIN PAYMENT_
    ON ORDER_.ORDERNUM = PAYMENT_.ORDERNUM
FULL JOIN ORDER_PRODUCT
    ON ORDER_.ORDERNUM = ORDER_PRODUCT.ORDERNUM
FULL JOIN BIKE_
    ON ORDER_PRODUCT.BPRODUCTID = BIKE_.BPRODUCTID
FULL JOIN ACCESSORIES_
    ON ORDER_PRODUCT.APRODUCTID = ACCESSORIES_.APRODUCTID
WHERE CUSTOMER_.LNAME = 'CENA';
```

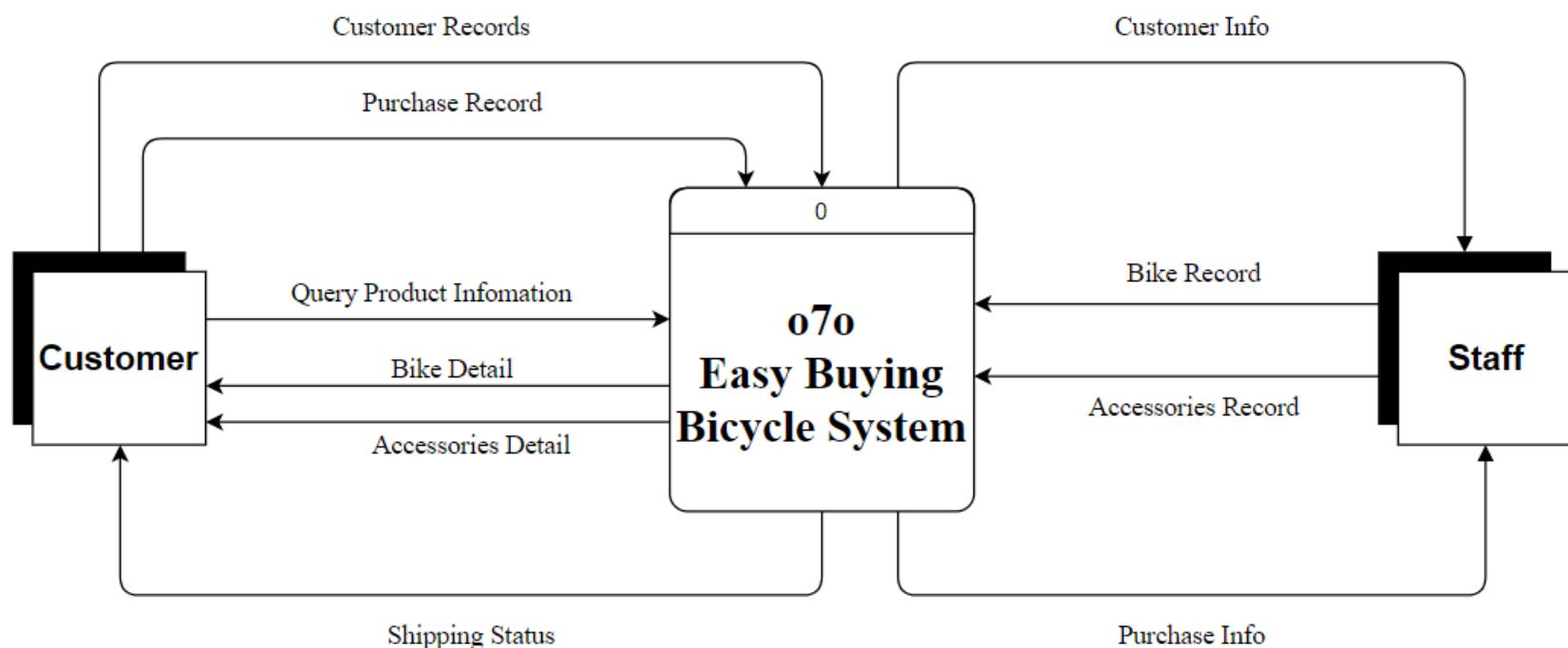
Results:

LNAME	EMAIL	STREET	CITY	POSTCODE	DELIVERYDATE	PRODNAME	PRODNAME_1
1 CENA	john234@gmail.com	JALAN MERANTI JOHOR BAHRU		81310	18-DEC-17	(null)	ALUMINIUM ALLOY BIKE BELL

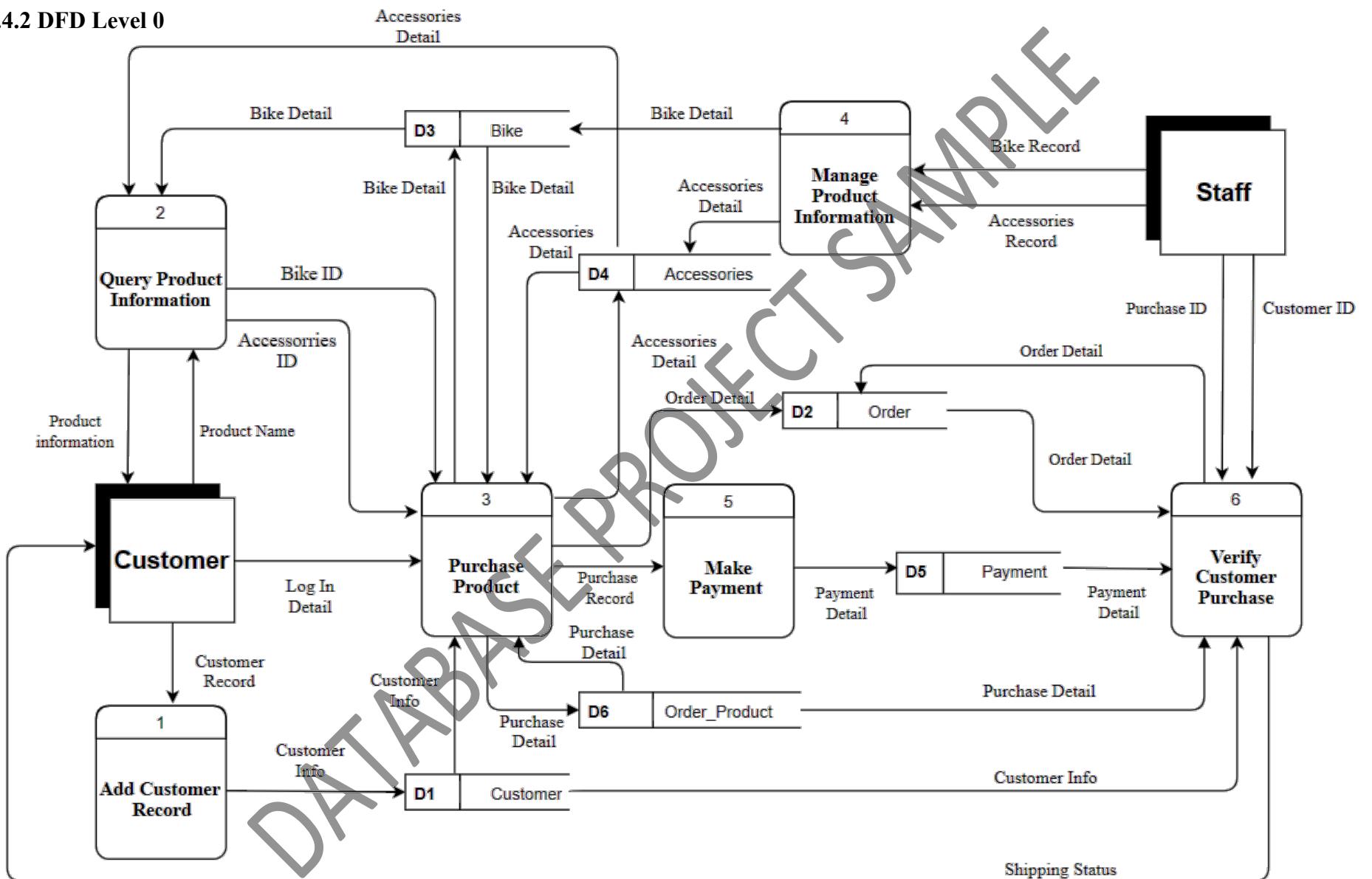
PRODNAME_1	QUANTITY	TOTALPRICE	TOTALPRICE_1
ALUMINIUM ALLOY BIKE BELL	1	5	10

5.4 Data Flow Diagram (DFD)

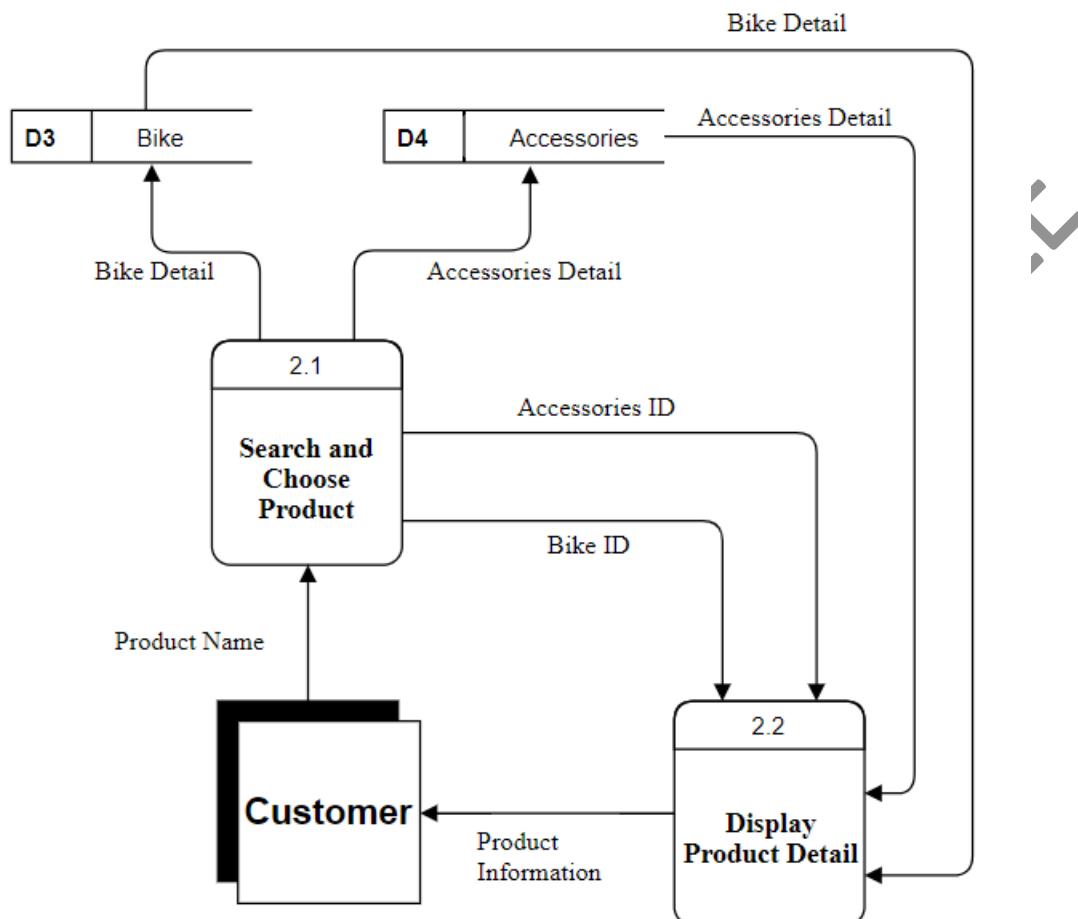
5.4.1 Context Diagram



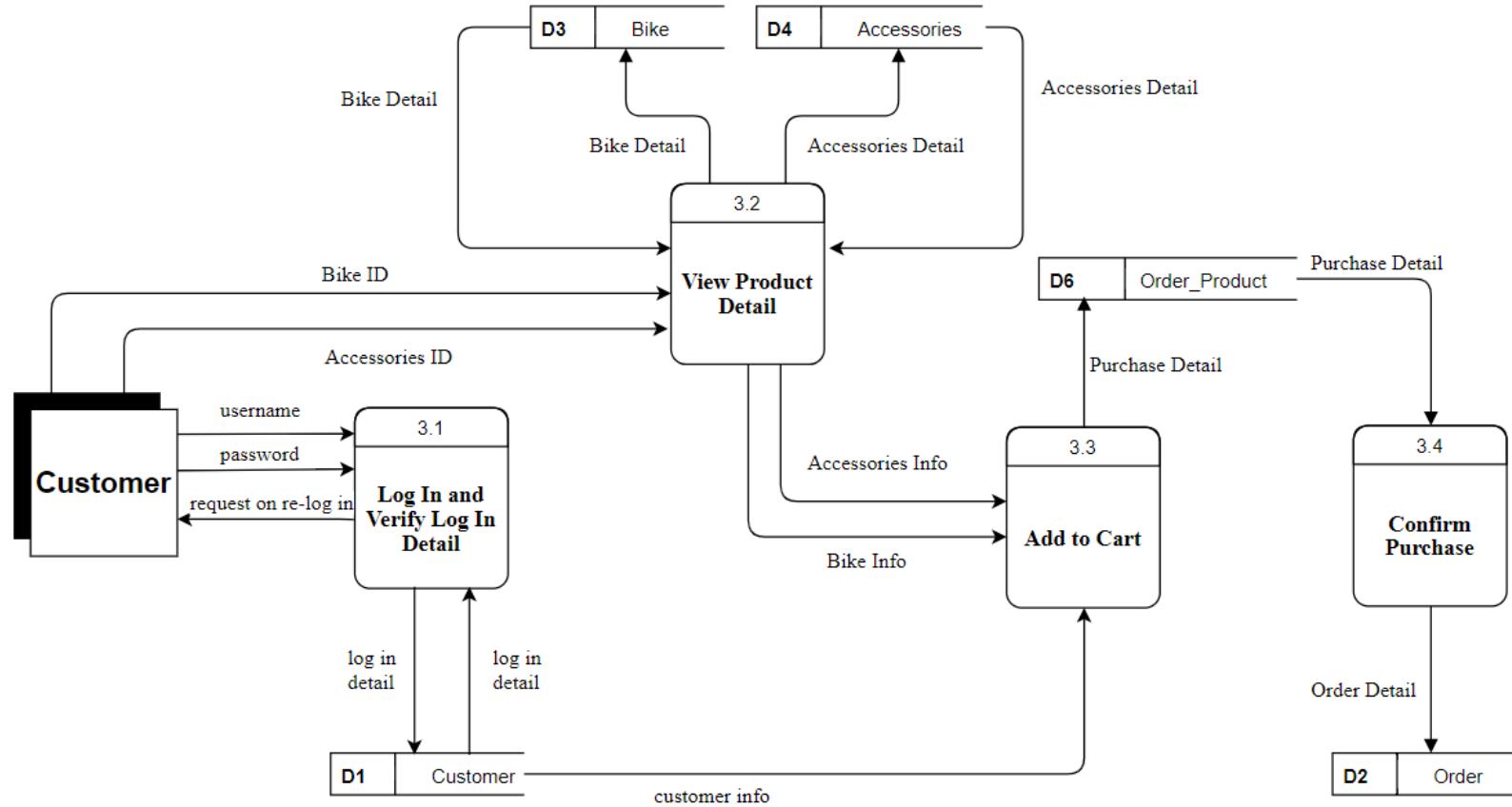
5.4.2 DFD Level 0



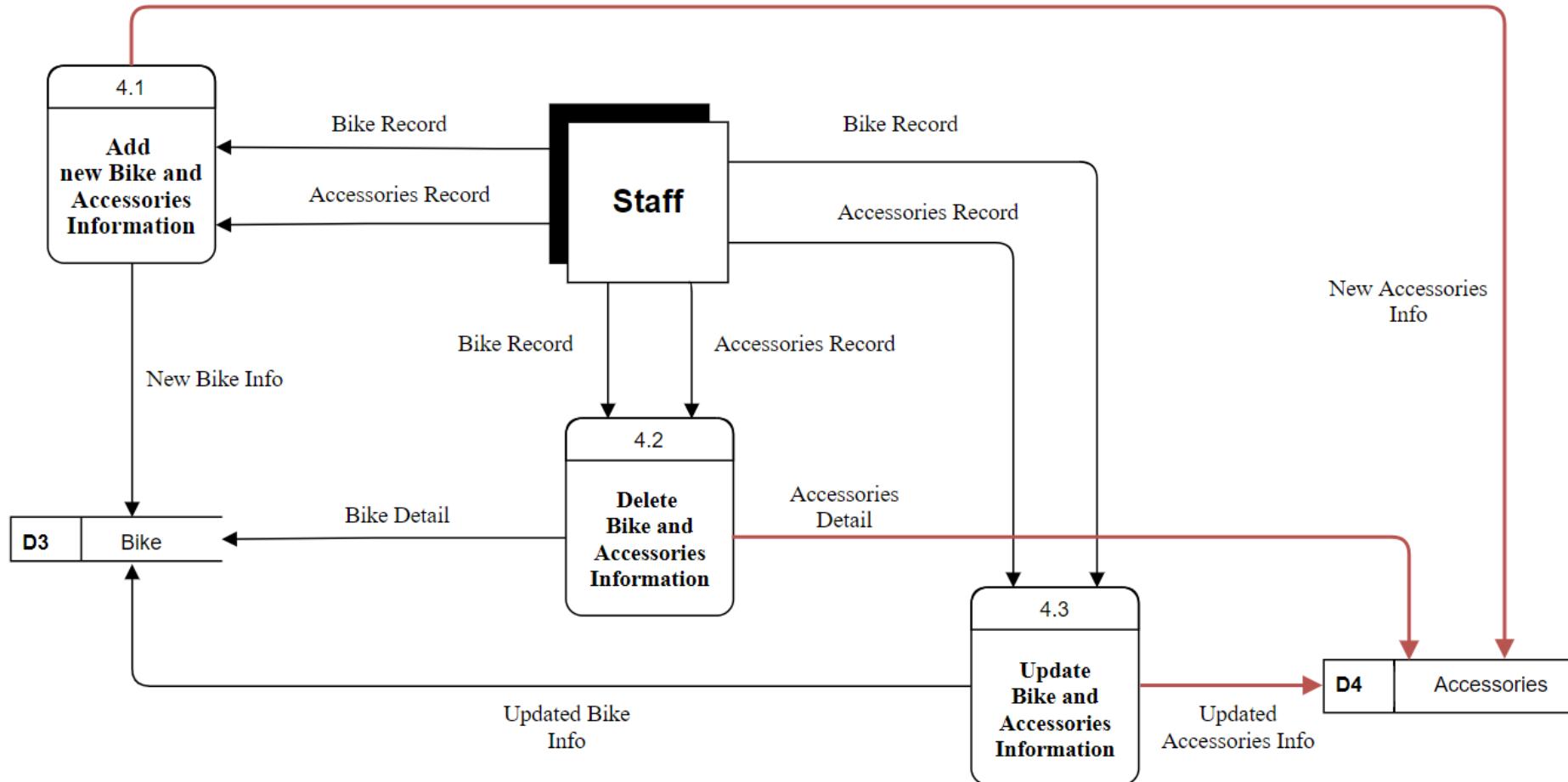
5.4.3 Child Diagram for Process 2



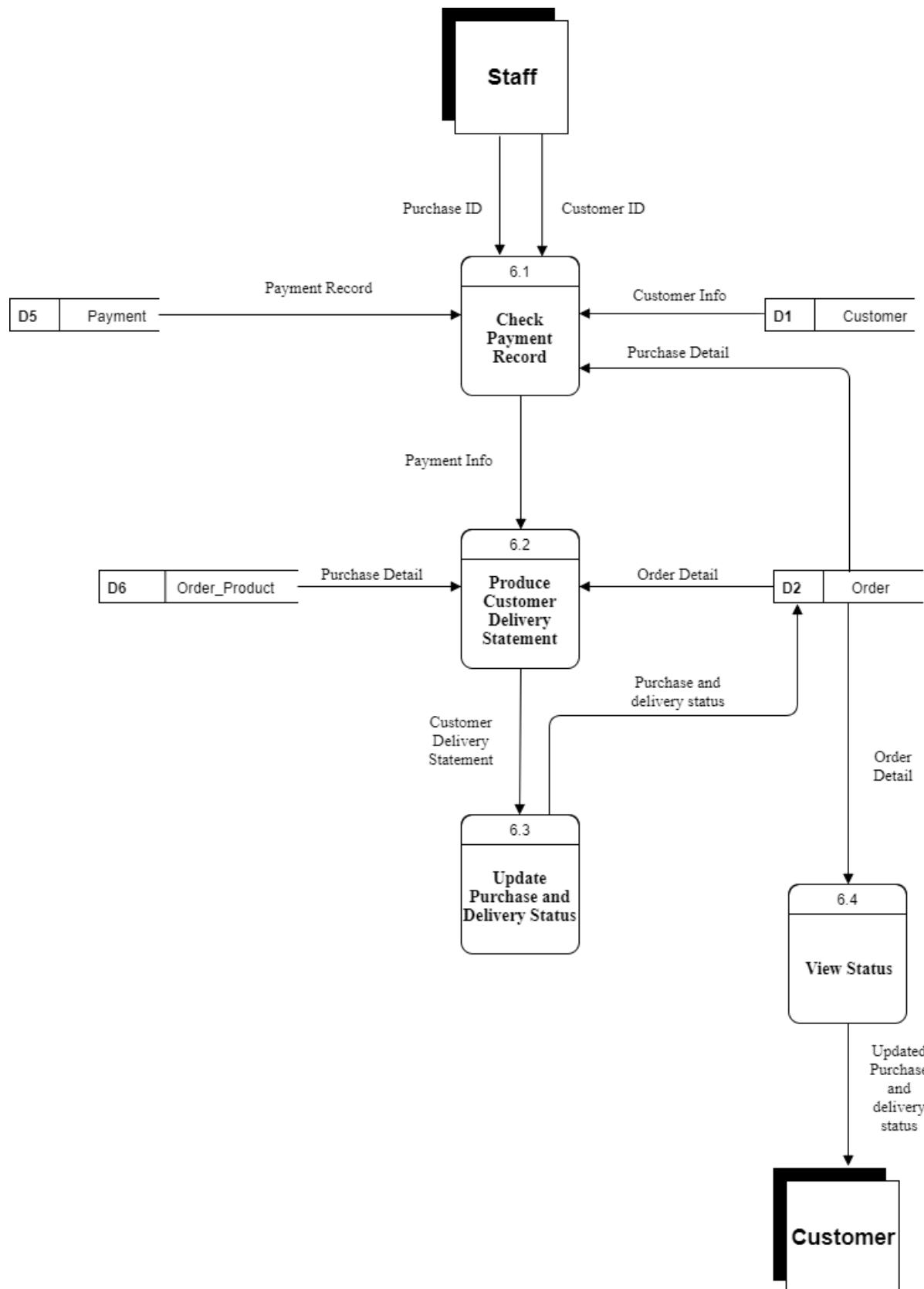
5.4.4 Child Diagram for Process 3



5.4.5 Child Diagram for Process 4



5.4.6 Child Diagram for Process 6



Appendix

Interview Output

4	A very good evening to Mr Anisham, the owner of Bicycle Shop in UTM. I am Vivekaanan s/o Sirgunan,a second year student from Faculty Of Computing, and these are my group members Yap Sin Yee, Lim Hui Ting and Nuraina Syaza. First of all, we would like to thank you for giving us such an opportunity to have an interview with in a very hectic and busy day. We have been assigned a project which has the objectives of analyzing problems and proposing solutions to develop new or improve current system. Thus, we have chosen your Bicycle Shop as our case study.
INTERVIEWEE:	Thank you everyone for choosing my shop as your case study. I will provide all the information you need based on your questions.
INTERVIEWER:	Alright Mr Anisham,without further argue we begin our interviewing session with our first question. When does the bicycle shop was established in UTM?
INTERVIEWEE:	This bicycle shop was established in September, 2014.
INTERVIEWER:	It has been 3 years and your shop still have that new look. Next, what kind of tasks or operations do you perform on a daily basis?
INTERVIEWEE:	For every typical day, we sell, repair and also rent bicycle.
INTERVIEWER:	Noted Mr Anisham. By the way, what types of reports and records you used to take note of your daily data?
INTERVIEWEE:	We used to record our daily data with the aid of manually written books and also computer. We record our customer's data in the book and check the progress of our business once a week. Furthermore, I use my personnel computer to save all of my customer's data by copying and pasting them from the book at the end of each month.
INTERVIEWER:	Well, what types of records do you have to keep track of, Mr Anisham?
INTERVIEWEE:	There are 3 main records that we prioritise. First of all, bicycle rental records. We will keep track of our customers who rent bicycle from my shop. To be specific, I record each of their name, matric

	<p>number ,contact number and also the id of the bicycle they rent. Moreover, we also record the amount of sales we obtained on a particular month. In addition, we will keep track of amount of stock or accessories left in our shop. This gives me awareness to always add on accessories when needed.</p>
INTERVIEWER:	Taken note Mr Anisham. What do you think about recording data in written way which is the traditional method, at this modern era?
INTERVIEWEE:	Based on my opinion, recording data in a written way is not suitable and relevant. As the world is getting modernised day by day, it will be much more easier and comfortable, if we use digital recording system or well known as a database system.
INTERVIEWER:	Thank you for your opinion Mr Anisham. Next question, how do you normally backup your written records?
INTERVIEWEE:	At the moment, we are using receipt as our backup data. Once a customer buys a bicycle from our shop, we will provide a receipt to the customer and keep a copy of that receipt for future reference.
INTERVIEWER:	Mr Anisham, do you think that file based system is reliable and relevant to this bicycle shop?
INTERVIEWEE:	Generally, file based system is neither reliable nor relevant for bigger companies or even shops. At the moment, I am still having a small shop in UTM. But, I am having a plan of upgrading my shop in the future, which I think, file based system is not a anymore compared to database system.
INTERVIEWER:	Alright Mr Anisham, without further ado, we begin our interviewing session with our first question. What are the problems you have faced with your current system?
INTERVIEWEE:	There are a few problems that I have been facing with the existing system. Sometimes, I will miss out to store information about my customers whom bought bicycle from my shop. This causes major problems at the end of the month when I want to calculate total sales and so on. Secondly, I might forget to store customer's record of buying from my shop. For your information, bicycles that has been bought from my shop will be given free UTM sticker, accessories,

	repairing service and more. Thus, customers who have already bought bicycle from our shop will ask for those promotions and the situation gets complicated for me.
INTERVIEWER:	These problems can cause complicated problems to you and your existing system Mr Anisham. If we propose a new system for your shop, how do you expect the system to be?
INTERVIEWEE:	That is the happiest thing I will ever hear for today! Well if that is the case, I prefer having online payment system embedded in the new system. This will surely provide ease to the customers in doing their payment. They don't have to bring a large amount of cash to buy their bike. Elsewhere, I prefer a system with the capability of inserting, modifying, storing and deleting customer's data. This might reduce my problems that arise due to forgetfulness and also keeps me, and my shop updated.
INTERVIEWER:	Thank you for your opinion on the new system Mr Anisham. Can I know, what is the average rental rate for your current shop and also your average sales per month?
INTERVIEWEE:	I will pay an average of RM150-RM200 per month for renting this shop. Moreover, I will obtain from about RM500-RM600 through bicycle sales, repairing services and more.
INTERVIEWER:	Who are your majority customers?
INTERVIEWEE:	Since this shop is located inside UTM campus, majority of my customers are UTM students.
INTERVIEWER:	Alright Mr Anisham. Do you think your sales level is progressing? Explain why.
INTERVIEWEE:	I will consider this as a hypothetical question. Since this shop is placed inside UTM campus, my sales level will progress as semester moves on and students stay in campus. But, my sales level will eventually drop during semester breaks where, students will move to out of the campus to their hometown.
INTERVIEWER:	Well that is considerable Mr Anisham. My last question, do you think your current system is satisfying and totally reliable?

INTERVIEWEE:	As I have informed at the beginning, I am still using a file based system which is hand written data to be exact. It is not totally reliable because keeping data in written records provide very less option for me to backup them. Data could easily be lost since I am only using a receipt system. Furthermore, it is also not so satisfying for me because, I must refer my record book from the beginning till the end of month to come up with the total amount of sales. As we know, using a database system is much more easier and the files can be well organized by itself.
INTERVIEWER:	That opinion of yours, marks the end of this interview session. Once again, we would like to thank you for sparing your priceless time with us, involving in this interview session.
INTERVIEWEE:	Thank you everyone for giving me a chance to express about my bicycle shop.

Other Supporting



Bicycle Catalogue



Accessories Catalogue



Bike Shop Membership Sticker

Bike Shop Poster