Database System for Customer Support

Team III
David Song
Nishchal Nigam
Dominick Caimano

Overview:

We created a marketplace and customer support database system. The system will allow customers to track purchase and cases for issues, employees can track cases, assign them, and search for them and also aggregate data for them.

Types of Users:

Customer: Can login, track purchases, make cases, view and make common resolutions to cases.

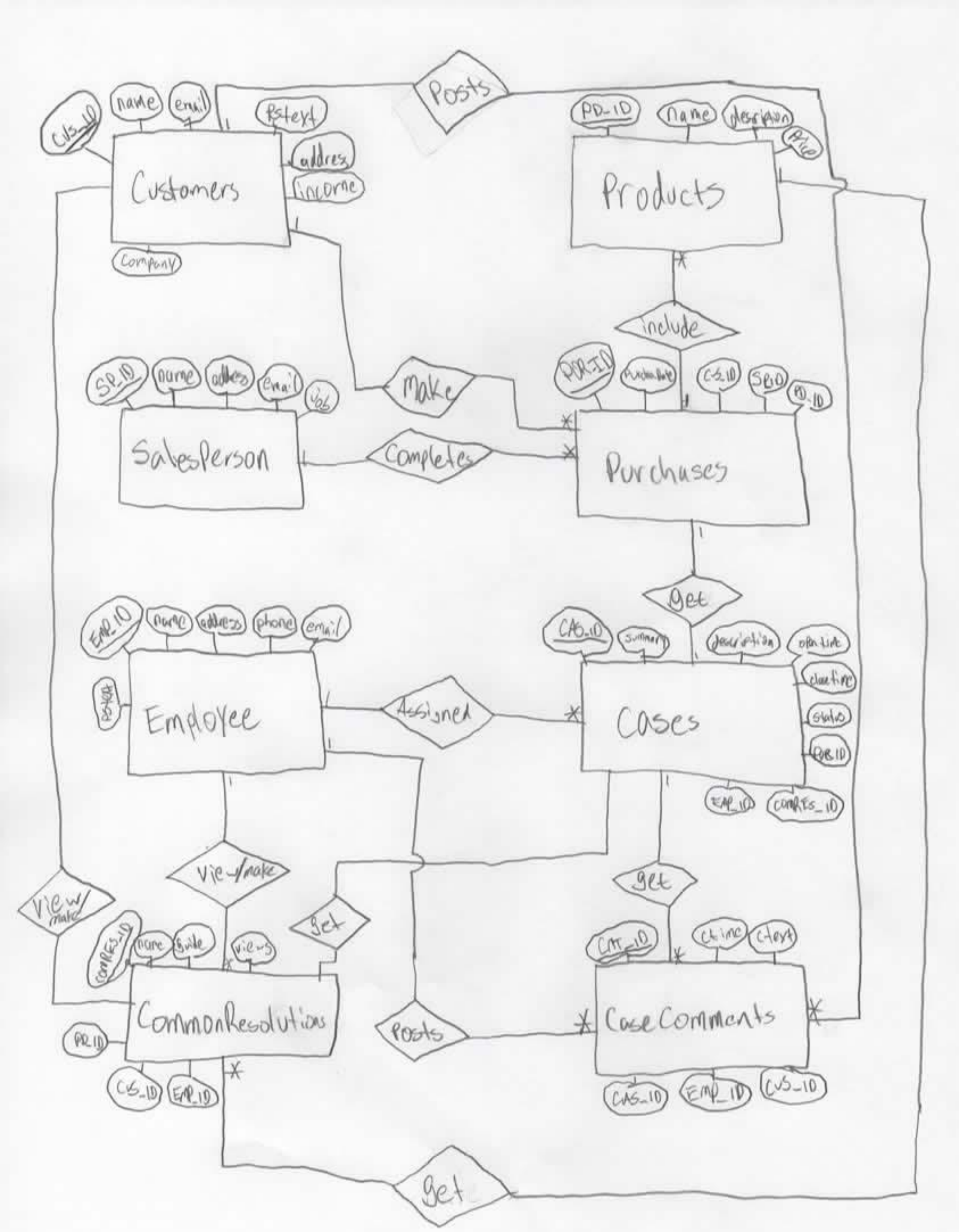
Salesperson: Don't need to login, they are linked to sales they made to customers.

Employee: Can login to the employee side of the system, can search for customers and see their purchases, search for cases by status and timeframe, search for common resolutions. Employees will also assign themselves to cases, check for new cases, and can close cases.

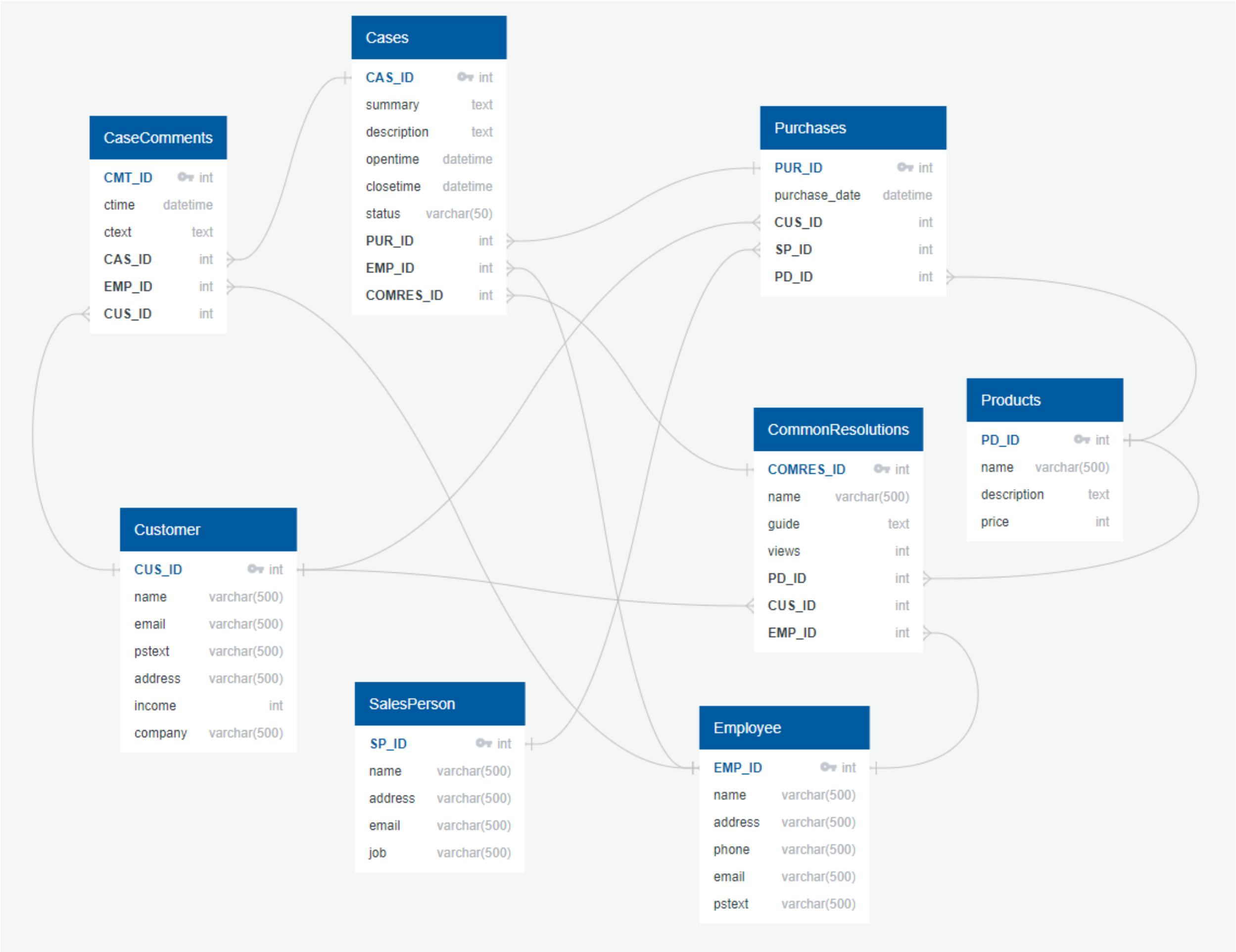
Assumptions of Database:

- Products Table: Products will be searched by name so UNIQUE not required.
- CommonResolutions Table: Can be written by an employee or customer.
- Case Table: Created by customers and only if they purchased the product

ER Diagram


Posts
CUS_ID
name
email
PStext
address
income
Customers
Company
PD_ID
name
description
Price
Products
include
SP_ID
name
address
email
Job
SalesPerson
make
Completes
PUR_ID
PurchaseDate
CS_ID
SP_ID
PD_ID
Purchases
Get
EMP_ID
name
address
phone
email
Position
Employee
Assigned
CAS_ID
summary
description
open time
close time
status
PUR_ID
Cases
EMP_ID
COMRES_ID
view/make
vie-/make
Get
get
COMRES_ID
name
Guide
views
CommonResolutions
CR_ID
CUS_ID
EMP_ID
Posts
CMT_ID
ctime
ctext
CaseComments
CAS_ID
EMP_ID
CUS_ID
Get

Set of relational schema resulting from the E-R diagram with identification of primary and foreign keys.



Cases	
CAS_ID	🔑 int
summary	text
description	text
opentime	datetime
closetime	datetime
status	varchar(50)
PUR_ID	int
EMP_ID	int
COMRES_ID	int

CaseComments	
CMT_ID	🔑 int
ctime	datetime
ctext	text
CAS_ID	int
EMP_ID	int
CUS_ID	int

Purchases	
PUR_ID	🔑 int
purchase_date	datetime
CUS_ID	int
SP_ID	int
PD_ID	int

Products	
PD_ID	🔑 int
name	varchar(500)
description	text
price	int

CommonResolutions	
COMRES_ID	🔑 int
name	varchar(500)
guide	text
views	int
PD_ID	int
CUS_ID	int
EMP_ID	int

Customer	
CUS_ID	🔑 int
name	varchar(500)
email	varchar(500)
pstext	varchar(500)
address	varchar(500)
income	int
company	varchar(500)

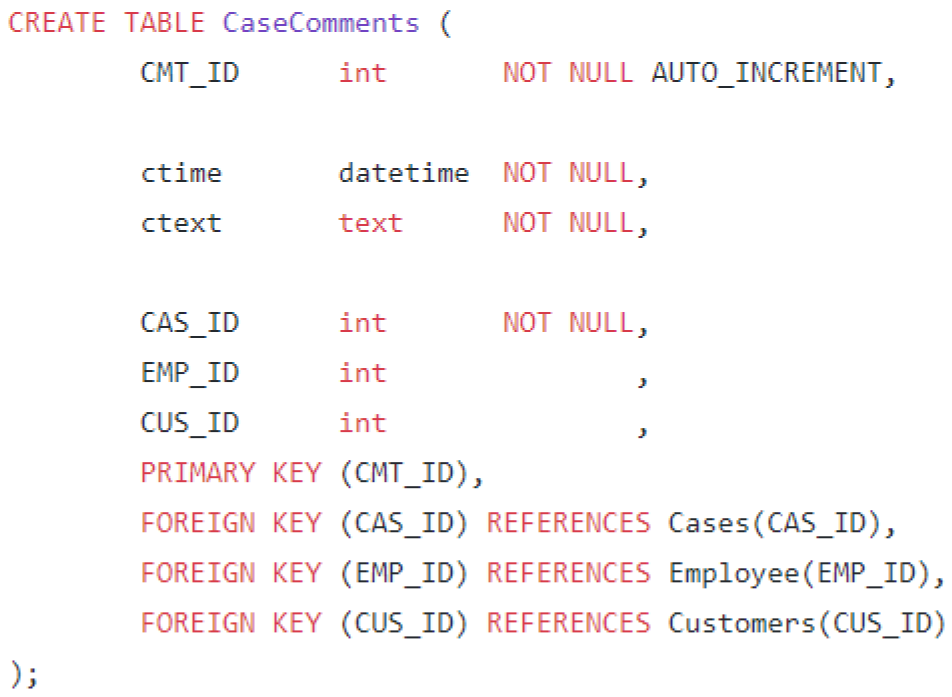
SalesPerson	
SP_ID	🔑 int
name	varchar(500)
address	varchar(500)
email	varchar(500)
job	varchar(500)

Employee	
EMP_ID	🔑 int
name	varchar(500)
address	varchar(500)
phone	varchar(500)
email	varchar(500)
pstext	varchar(500)

DDL statements to create the relational schema

```
CREATE TABLE Customers (
        CUS_ID  int          NOT NULL AUTO_INCREMENT,
        name    varchar(500) NOT NULL,
        email   varchar(500) NOT NULL UNIQUE,
        pstext  varchar(500) NOT NULL,
        address varchar(500) NOT NULL,
        income  int          NOT NULL,
        company varchar(500),
        PRIMARY KEY (CUS_ID)
);

CREATE TABLE Products (
        PD_ID       int          NOT NULL AUTO_INCREMENT,
        name        varchar(500) NOT NULL,
        description text         NOT NULL,
        price       int          NOT NULL,
        PRIMARY KEY (PD_ID)
);

 CREATE TABLE Purchases (
        PUR_ID  int NOT NULL AUTO_INCREMENT,

        purchase_date datetime NOT NULL,

        CUS_ID  int NOT NULL,
        SP_ID   int NOT NULL,
        PD_ID   int NOT NULL,
        PRIMARY KEY (PUR_ID),
        FOREIGN KEY (CUS_ID) REFERENCES Customers(CUS_ID),
        FOREIGN KEY (SP_ID) REFERENCES SalesPersons(SP_ID),
        FOREIGN KEY (PD_ID) REFERENCES Products(PD_ID)
 );
```

```
CREATE TABLE CommonResolutions (
        COMRES_ID   int          NOT NULL AUTO_INCREMENT,

        name        varchar(500) NOT NULL,
        guide       text         NOT NULL,
        views       int          NOT NULL,

        PD_ID       int          NOT NULL,
        CUS_ID      int,
        EMP_ID      int,
        PRIMARY KEY (COMRES_ID),
        FOREIGN KEY (PD_ID) REFERENCES Products(PD_ID),
        FOREIGN KEY (CUS_ID) REFERENCES Customers(CUS_ID),
        FOREIGN KEY (EMP_ID) REFERENCES Employee(EMP_ID)
);

CREATE TABLE Cases (
        CAS_ID      int      NOT NULL AUTO_INCREMENT,

        summary     text     NOT NULL,
        description text     NOT NULL,
        opentime    datetime NOT NULL,
        closetime   datetime,
        status      varchar(50) NOT NULL,

        PUR_ID      int      NOT NULL,
        EMP_ID      int,
        COMRES_ID   int,
        PRIMARY KEY (CAS_ID),
        FOREIGN KEY (PUR_ID) REFERENCES Purchases(PUR_ID),
        FOREIGN KEY (EMP_ID) REFERENCES Employee(EMP_ID),
        FOREIGN KEY (COMRES_ID) REFERENCES CommonResolutions(COMRES_ID)
);
```



```
CREATE TABLE CaseComments (
        CMT_ID      int       NOT NULL AUTO_INCREMENT,

        ctime       datetime  NOT NULL,
        ctext       text      NOT NULL,

        CAS_ID      int       NOT NULL,
        EMP_ID      int               ,
        CUS_ID      int               ,
        PRIMARY KEY (CMT_ID),
        FOREIGN KEY (CAS_ID) REFERENCES Cases(CAS_ID),
        FOREIGN KEY (EMP_ID) REFERENCES Employee(EMP_ID),
        FOREIGN KEY (CUS_ID) REFERENCES Customers(CUS_ID)
);
```

Indexes added to tables:

Alterations to tables.

```
ALTER TABLE Customers ADD INDEX CUS_IDX (name);

ALTER TABLE Products ADD INDEX PD_IDX (name);

ALTER TABLE SalesPersons ADD INDEX SP_IDX (name);

ALTER TABLE Employee ADD INDEX EMP_IDX (name);

ALTER TABLE Purchases ADD INDEX PUR_IDX (purchase_date);

ALTER TABLE CommonResolutions ADD INDEX COMRES_IDX (name);

ALTER TABLE Cases ADD INDEX CAS_IDX (status);

ALTER TABLE CaseComments ADD INDEX CASCMT_IDX (ctime);
```

Tables are in BCNF Normal Form. Full DDL statements and justifications are also included in the attached database.sql file.

Front End to Back End Design and System Implementation:

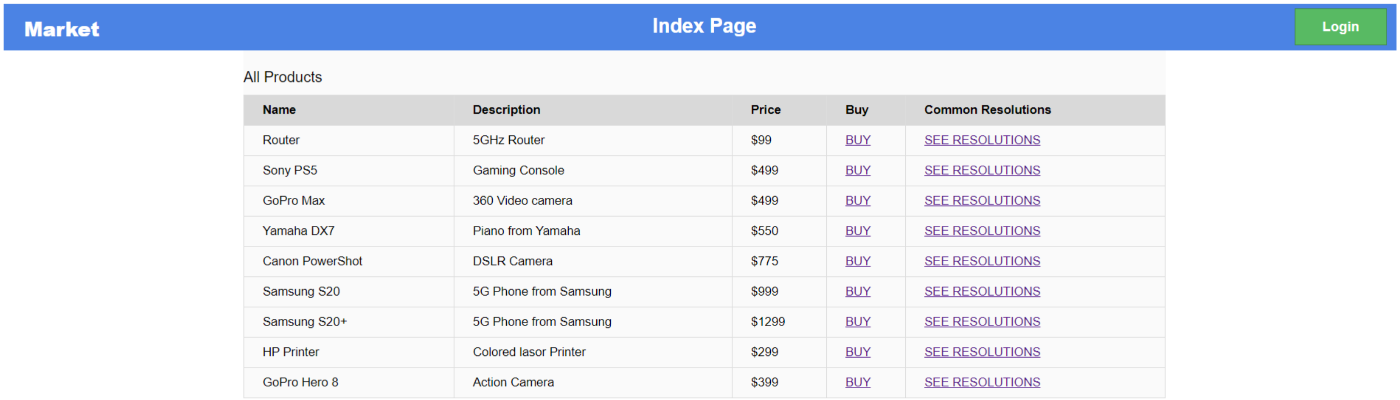
We used Amazon Web Services cloud based IDE for real time collaboration. This was one of the best options when doing this project, as the only other alternative was to do work separately in GIT branches, and merge work together later on. Not only were we able to share the same development environment, but also allowed us to see what each other was doing on screen, which helped up divide our work even more efficiently.

Its run on an AWS EC2 Instance with Ubuntu 18 preinstalled and later we installed MySQL and NodeJS afterwards. The backend server was developed using NodeJS with the ExpressJS framework, which allowed us to not only handle basic GET and POST request routing asynchronously and non-blocking, but also allowed us to add in our own middle-wares for extra functionality. One such feature is the file-system based session store, which handles session cookies and internal session data for users that are logged in. Another security middleware we attached was a simple DOS protection, to prevent outside attackers overwhelming our server, which is already running on a very underpowered, free virtual machine (single-core, 1GB RAM, and 10GB of storage).

We also allocated a free static IP address and installed and configured Nginx to forward all public traffic into our instance, which is hidden behind a private IP under AWS's subnet, and that allowed the server to be accessible from the internet from this address http://3.23.28.11. AJAX and JQUERY were also used to communicate to the Database. Complete API documentation can be found here: http://3.23.28.11/api/.

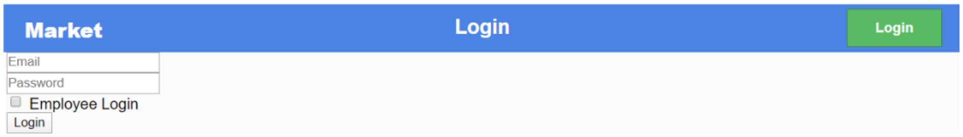
One downside of using Cloud9 is that since AWS manages our EC2 instances, after some time once nobody is accessing, the instance will automatically shutdown. We do however plan to host this project on a separate server so that we can keep the server alive indefinitely.

The home screen of the system starts with a list of products with a option to buy and login



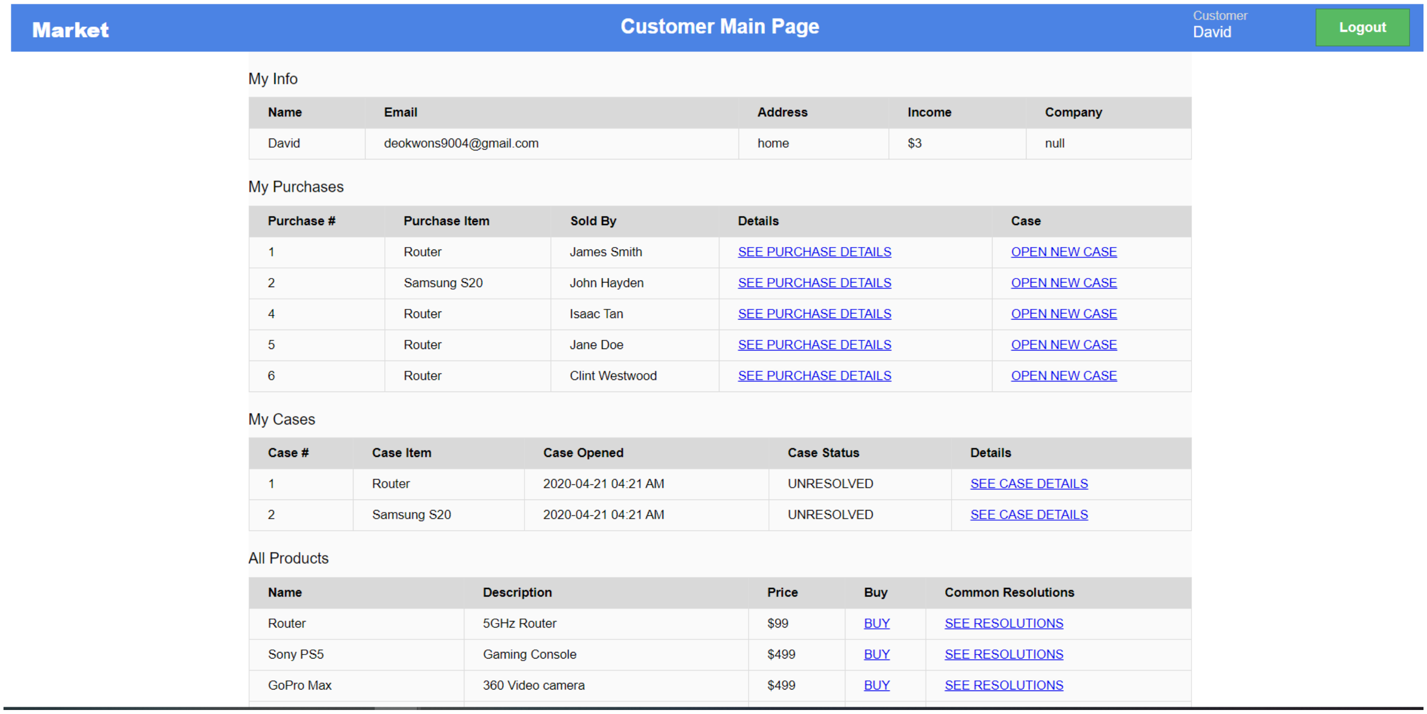
All Products

Name	Description	Price	Buy	Common Resolutions
Router	5GHz Router	$99	BUY	SEE RESOLUTIONS
Sony PS5	Gaming Console	$499	BUY	SEE RESOLUTIONS
GoPro Max	360 Video camera	$499	BUY	SEE RESOLUTIONS
Yamaha DX7	Piano from Yamaha	$550	BUY	SEE RESOLUTIONS
Canon PowerShot	DSLR Camera	$775	BUY	SEE RESOLUTIONS
Samsung S20	5G Phone from Samsung	$999	BUY	SEE RESOLUTIONS
Samsung S20+	5G Phone from Samsung	$1299	BUY	SEE RESOLUTIONS
HP Printer	Colored lasor Printer	$299	BUY	SEE RESOLUTIONS
GoPro Hero 8	Action Camera	$399	BUY	SEE RESOLUTIONS



For employees to login they must select Employee login.

The Customer Main Page:



My Info

Name	Email	Address	Income	Company
David	deokwons9004@gmail.com	home	$3	null

My Purchases

Purchase #	Purchase Item	Sold By	Details	Case
1	Router	James Smith	SEE PURCHASE DETAILS	OPEN NEW CASE
2	Samsung S20	John Hayden	SEE PURCHASE DETAILS	OPEN NEW CASE
4	Router	Isaac Tan	SEE PURCHASE DETAILS	OPEN NEW CASE
5	Router	Jane Doe	SEE PURCHASE DETAILS	OPEN NEW CASE
6	Router	Clint Westwood	SEE PURCHASE DETAILS	OPEN NEW CASE

My Cases

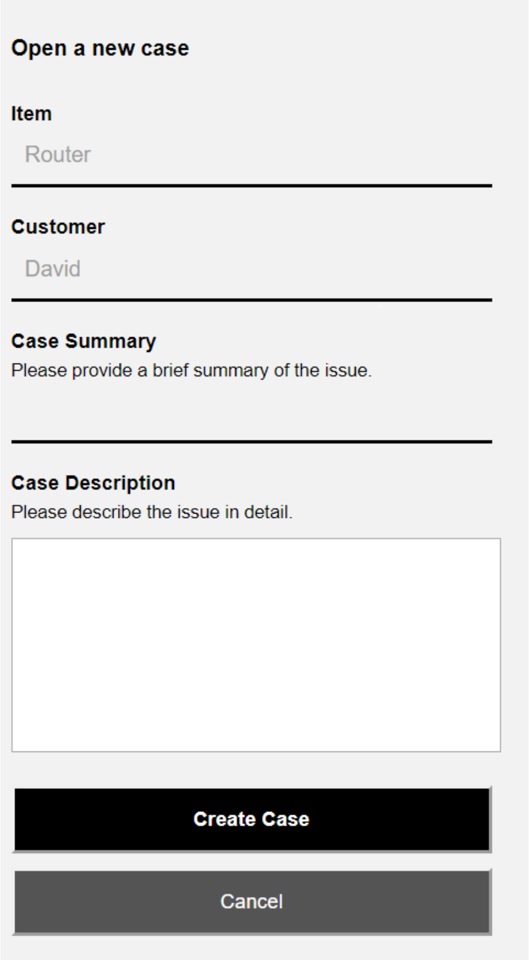
Case #	Case Item	Case Opened	Case Status	Details
1	Router	2020-04-21 04:21 AM	UNRESOLVED	SEE CASE DETAILS
2	Samsung S20	2020-04-21 04:21 AM	UNRESOLVED	SEE CASE DETAILS

All Products

Name	Description	Price	Buy	Common Resolutions
Router	5GHz Router	$99	BUY	SEE RESOLUTIONS
Sony PS5	Gaming Console	$499	BUY	SEE RESOLUTIONS
GoPro Max	360 Video camera	$499	BUY	SEE RESOLUTIONS

Here we can see the customers info, their purchases, cases, and at the bottom is a list of products they can purchase.

The Case Creation Page:



Open a new case

Item

Router

Customer

David

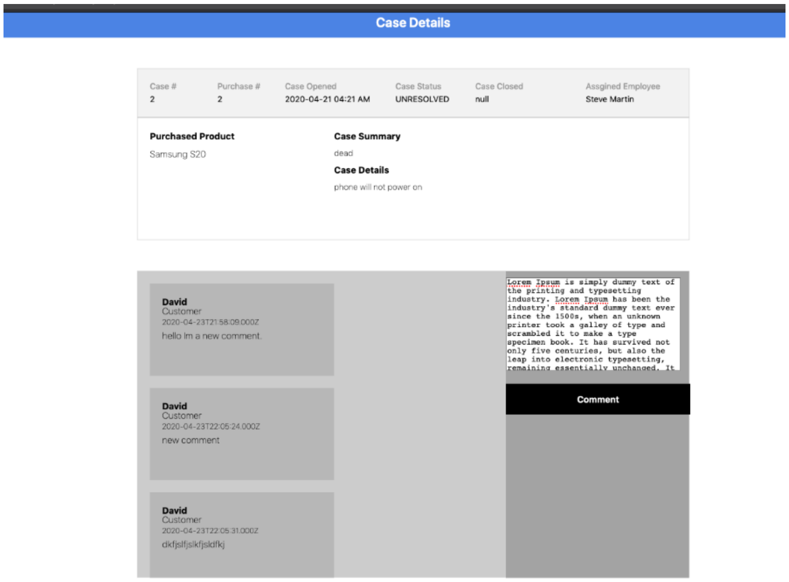
Case Summary

Please provide a brief summary of the issue.

Case Description

Please describe the issue in detail.

Case Details page with comments, also can add notes:



Case #	Purchase #	Case Opened	Case Status	Case Closed	Assgined Employee
2	2	2020-04-21 04:21 AM	UNRESOLVED	null	Steve Martin

Purchased Product

Samsung S20

Case Summary

dead

Case Details

phone will not power on

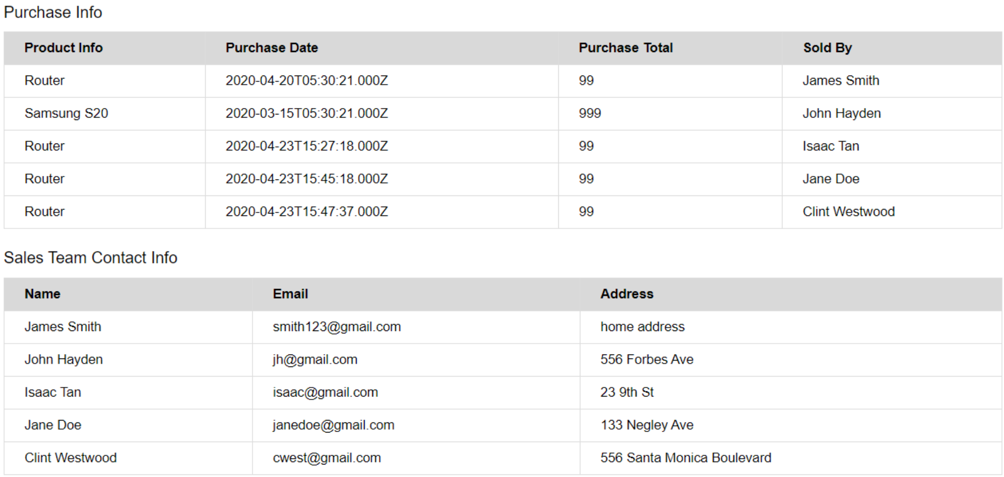
David
Customer
2020-04-23T21:58:09.000Z
hello Im a new comment.

David
Customer
2020-04-23T22:05:24.000Z
new comment

David
Customer
2020-04-23T22:05:31.000Z
dkfjslfjslkfjsldfkj

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It

The purchase details page shows all products purchased and salesperson contact info.



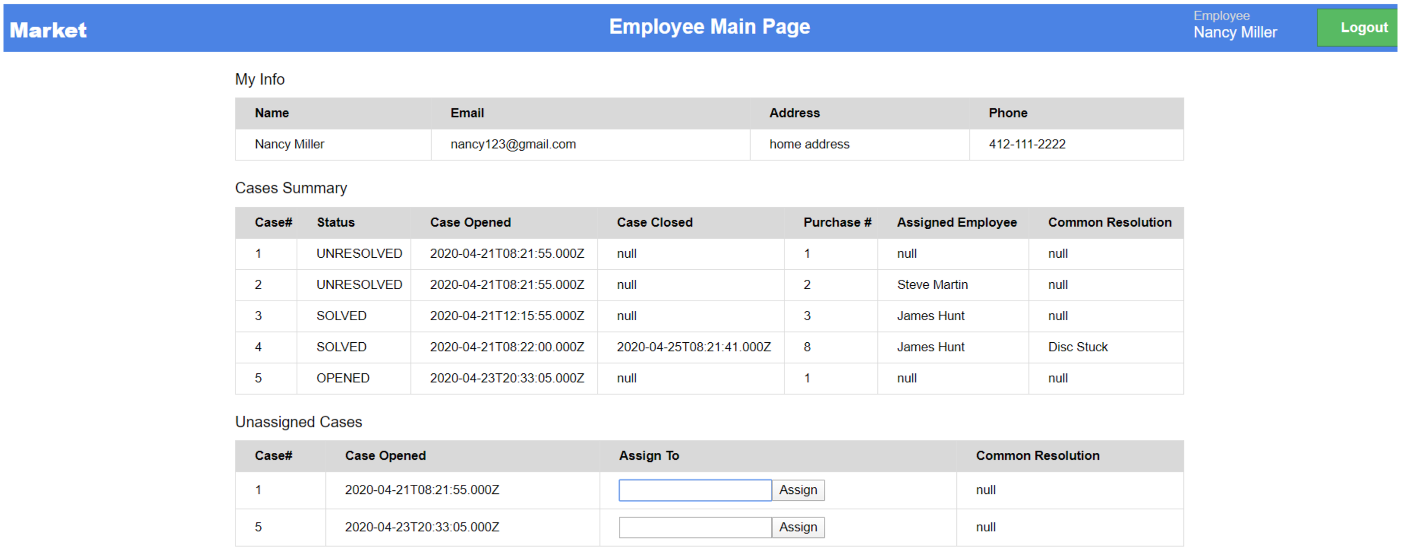
Purchase Info

Product Info	Purchase Date	Purchase Total	Sold By
Router	2020-04-20T05:30:21.000Z	99	James Smith
Samsung S20	2020-03-15T05:30:21.000Z	999	John Hayden
Router	2020-04-23T15:27:18.000Z	99	Isaac Tan
Router	2020-04-23T15:45:18.000Z	99	Jane Doe
Router	2020-04-23T15:47:37.000Z	99	Clint Westwood

Sales Team Contact Info

Name	Email	Address
James Smith	smith123@gmail.com	home address
John Hayden	jh@gmail.com	556 Forbes Ave
Isaac Tan	isaac@gmail.com	23 9th St
Jane Doe	janedoe@gmail.com	133 Negley Ave
Clint Westwood	cwest@gmail.com	556 Santa Monica Boulevard

The employee main page:



My Info

Name	Email	Address	Phone
Nancy Miller	nancy123@gmail.com	home address	412-111-2222

Cases Summary

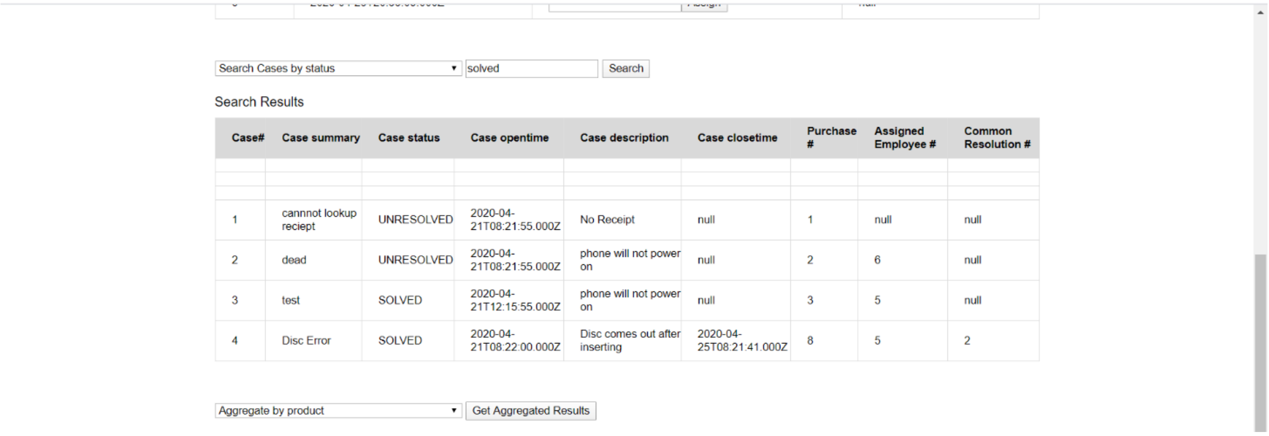
Case#	Status	Case Opened	Case Closed	Purchase #	Assigned Employee	Common Resolution
1	UNRESOLVED	2020-04-21T08:21:55.000Z	null	1	null	null
2	UNRESOLVED	2020-04-21T08:21:55.000Z	null	2	Steve Martin	null
3	SOLVED	2020-04-21T12:15:55.000Z	null	3	James Hunt	null
4	SOLVED	2020-04-21T08:22:00.000Z	2020-04-25T08:21:41.000Z	8	James Hunt	Disc Stuck
5	OPENED	2020-04-23T20:33:05.000Z	null	1	null	null

Unassigned Cases

Case#	Case Opened	Assign To	Common Resolution
1	2020-04-21T08:21:55.000Z	Assign	null
5	2020-04-23T20:33:05.000Z	Assign	null

The Employee main page that shows cases and cases can be assigned to employees by ID.

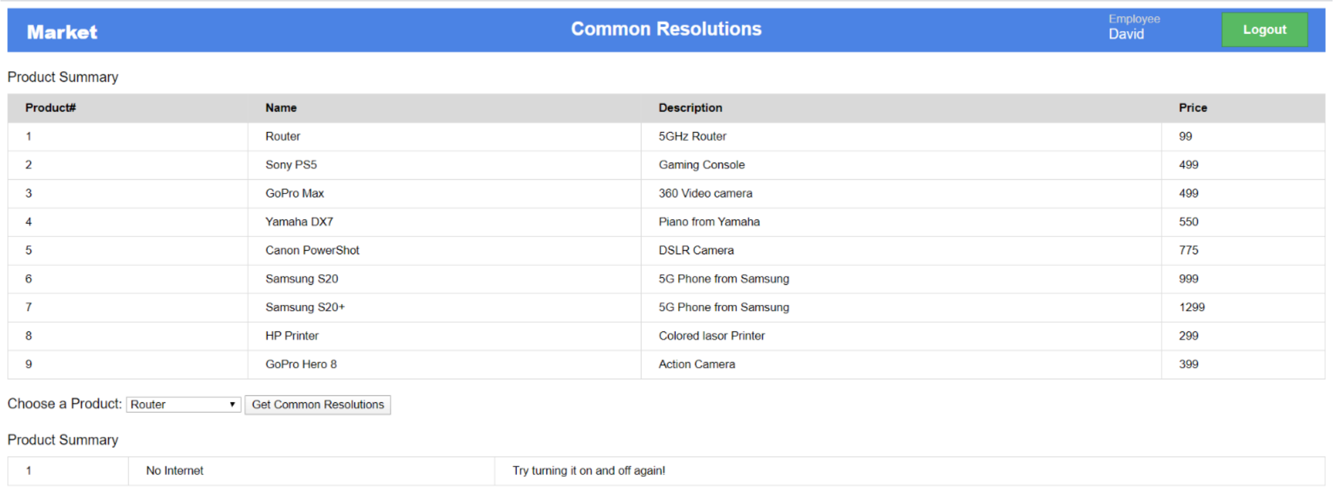
Employees can search for cases, and see aggregated results of case totals:



Search Results

Case#	Case summary	Case status	Case opentime	Case description	Case closetime	Purchase #	Assigned Employee #	Common Resolution #
1	cannnot lookup reciept	UNRESOLVED	2020-04-21T08:21:55.000Z	No Receipt	null	1	null	null
2	dead	UNRESOLVED	2020-04-21T08:21:55.000Z	phone will not power on	null	2	6	null
3	test	SOLVED	2020-04-21T12:15:55.000Z	phone will not power on	null	3	5	null
4	Disc Error	SOLVED	2020-04-21T08:22:00.000Z	Disc comes out after inserting	2020-04-25T08:21:41.000Z	8	5	2

The Common Resolutions Page:



Product Summary

Product#	Name	Description	Price
1	Router	5GHz Router	99
2	Sony PS5	Gaming Console	499
3	GoPro Max	360 Video camera	499
4	Yamaha DX7	Piano from Yamaha	550
5	Canon PowerShot	DSLR Camera	775
6	Samsung S20	5G Phone from Samsung	999
7	Samsung S20+	5G Phone from Samsung	1299
8	HP Printer	Colored lasor Printer	299
9	GoPro Hero 8	Action Camera	399

Choose a Product: Router ▾ Get Common Resolutions

Product Summary

1	No Internet	Try turning it on and off again!

Can select common resolutions by product.

The Server and database does error control and prevents entries from being entered or retrieved without a proper key. For other improvements our web app could use improving especially for buy products, and checkout and cart feature could improve that. The UI could be improved in places like Common resolutions, links instead of drop down boxes. For employee page assigning cases could be improved to not using ID numbers. Remove null from entries on the website.

To go to live website please visit: http://3.23.28.11

Customer Login: deokwons9004@gmail.com Password: pass1234

Employee Login: nancy123@gmail.com Password: pass1234

AWS Login: https://us-east-2.console.aws.amazon.com/cloud9/ide/e969e5e2bf9b4ad08311fe11e0e4bba4

Account ID (12 digits): 599833182250
IAM user name: dbuser
Password: russ

MySQL Login Credentials:

Username: dbuser

Password: russ

For more instructions on testing, check out our github repo:

https://github.com/deokwons9004dev/IS2710-DB-Project

Or checkout the README.md and WORKPAD.md inside the zipped project file for all the bits and pieces we recorded.

And if we do get the project hosted on a differenet server, make sure to check it out at:

http://dbm.lonelyduck.xyz