

Pebcak

Hayden Anderson, Michael Elliott, Albert Morgan, Alex Ruef

<http://web.engr.oregonstate.edu/~morgaalb/cs340-project/public/>

1. Introduction

Pebcak Enterprises is the Internet service provider that works for you! We have created a website that allows our customers to pick and choose the Internet plan that works best for them. Customers are also empowered with the ability to update their account information to ensure that their excellent Internet service continues.

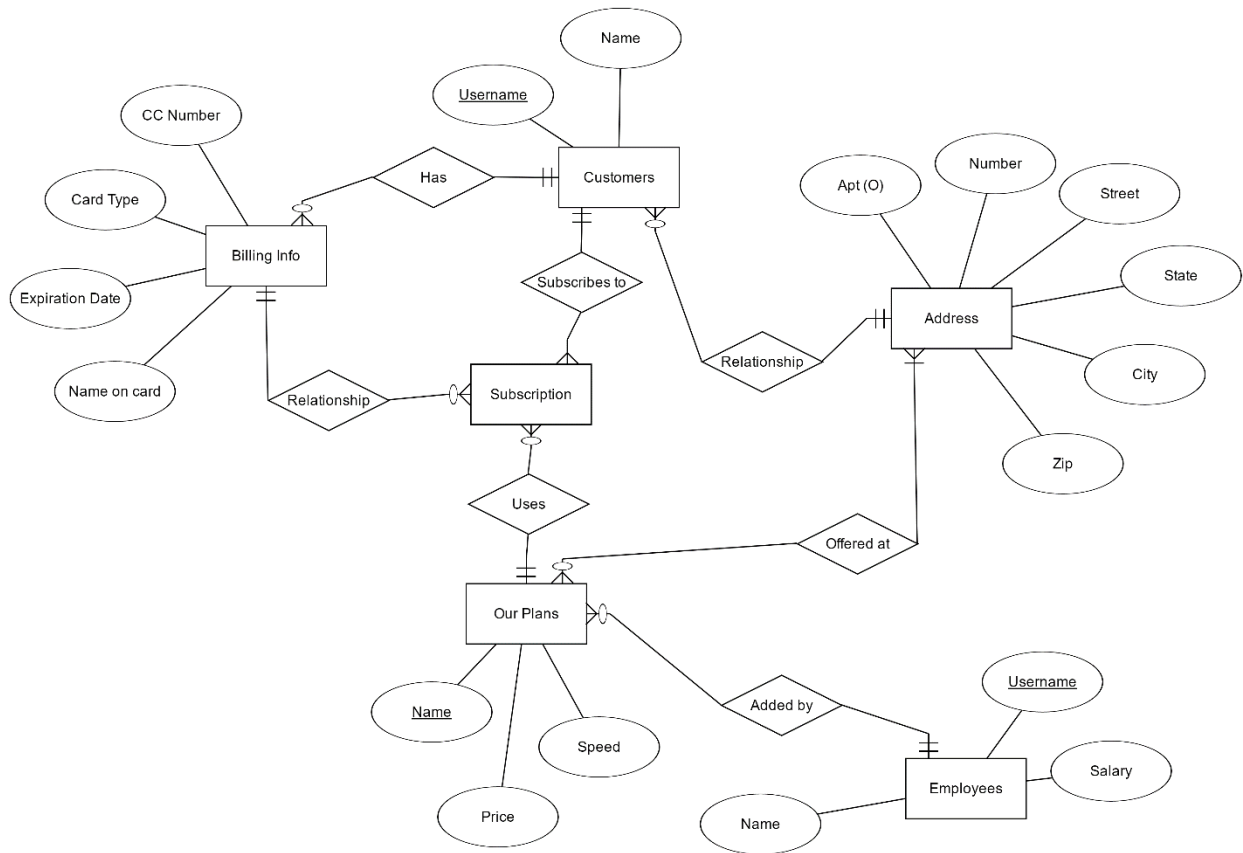
2. Detailed Functionality & Requirements

Business rules

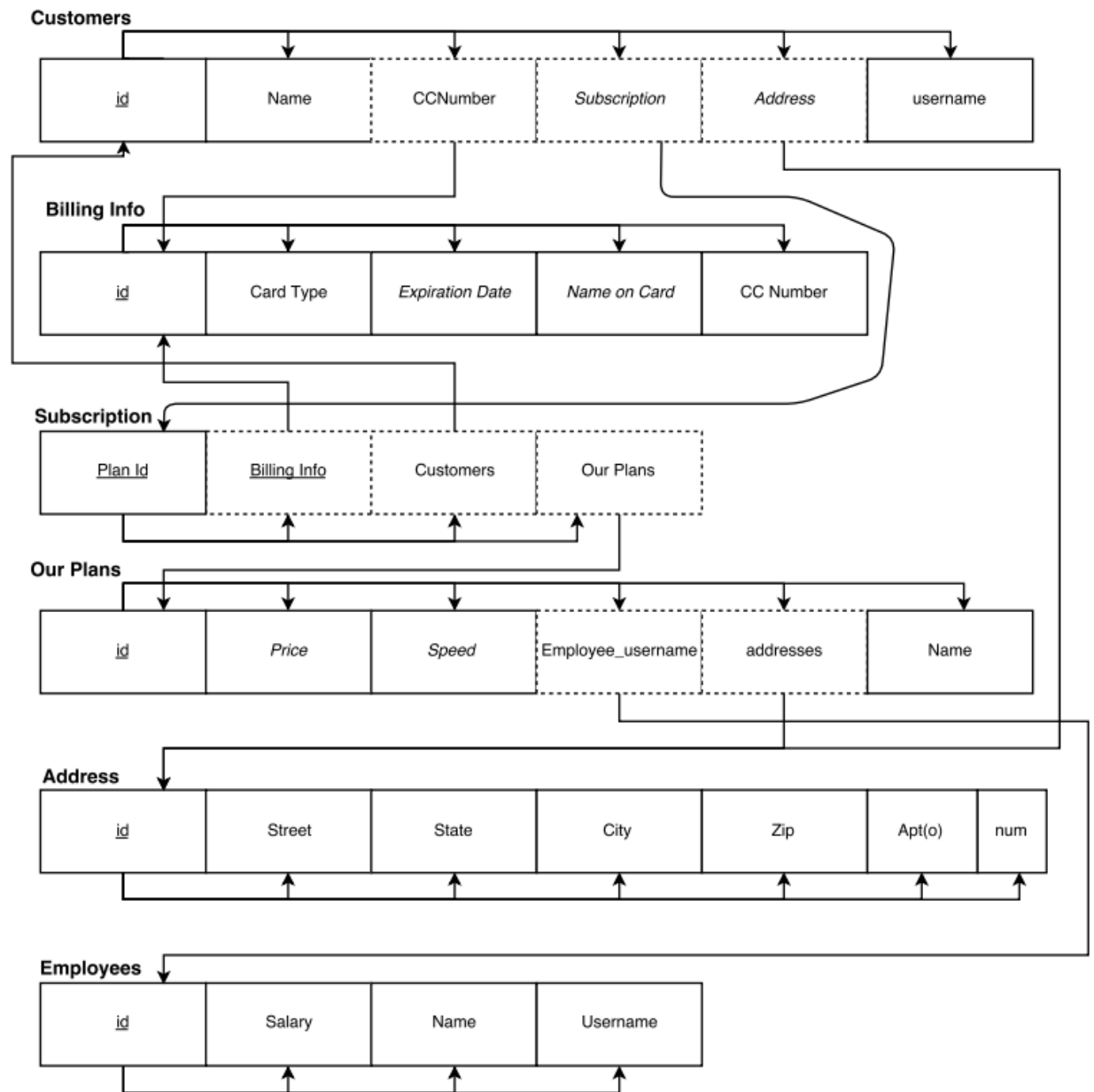
- a. Customers must log in before being able to view any details about their plans or billing
- b. Customers must be able to add new billing information when logged in
- c. Customers must be able to add and remove subscriptions when logged in
- d. Customers may have 0 or many plans
- e. Customers may have 0 or many billing methods
- f. A plan may not be signed up for at the same address twice.
- g. Each subscription must have exactly one customer and exactly one billing method
- h. Employees must login before viewing customer information or adding new plans
- i. You do not need to log in to view all of the plans

3. Database Design

- **ER Diagram of Database**



- **Relation Schema**









- **Database Tables**

[illegible]

DROP TABLE IF EXISTS Employee;

```
CREATE TABLE Employee (
```

			id	num	street	apt_no	city	state	zip
<input type="checkbox"/>			1	1234	SW GreenMachine Dr.	NULL	New York	NY	97111
<input type="checkbox"/>			2	555	SW malibu PI	3	Los Angeles	CA	97000
<input type="checkbox"/>			3	0	space GXY	NULL	Milky Way	MW	0

[illegible]

) ENGINE=InnoDB, CHARACTER SET=UTF8;



			id	cc_number	cc_type	expiration_date	name	user_id
<input type="checkbox"/>			1	12345678910	VISA	0000-00-00	Bruce Banner	1
<input type="checkbox"/>			2	10987654321	MASTERCARD	0000-00-00	Tony Stark	2
<input type="checkbox"/>			3	1111111111111111	BUCKS	0000-00-00	Peter Quill	3

DROP TABLE IF EXISTS Billing_Info;

CREATE TABLE Billing_Info (

id INT AUTO_INCREMENT NOT NULL,

cc_number DECIMAL(16) NOT NULL,

cc_type VARCHAR(255) NOT NULL,

expiration_date DATE NOT NULL,

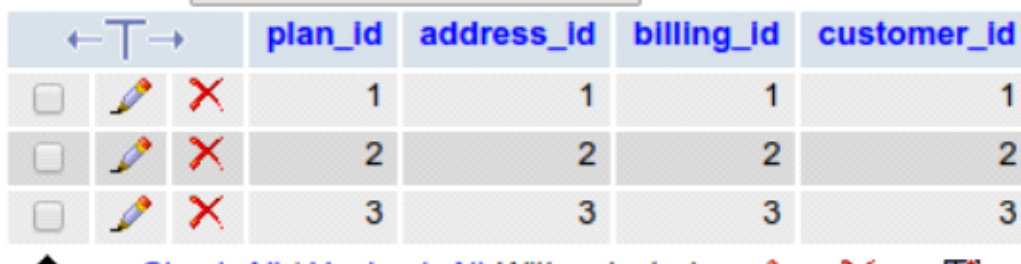
name VARCHAR(255) NOT NULL,

user_id INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (user_id) REFERENCES Customer(id) ON DELETE CASCADE

) ENGINE=InnoDB, CHARACTER SET=UTF8;



			plan_id	address_id	billing_id	customer_id
<input type="checkbox"/>			1	1	1	1
<input type="checkbox"/>			2	2	2	2
<input type="checkbox"/>			3	3	3	3

DROP TABLE IF EXISTS Subscription;

CREATE TABLE Subscription (

plan_id INT,

address_id INT,

billing_id INT,

customer_id INT,







PRIMARY KEY (plan_id, address_id),

FOREIGN KEY (billing_id) REFERENCES Billing_Info(id) ON DELETE CASCADE,

FOREIGN KEY (customer_id) REFERENCES Customer(id) ON DELETE CASCADE,

FOREIGN KEY (address_id) REFERENCES Address(id) ON DELETE CASCADE

) ENGINE=InnoDB, CHARACTER SET=UTF8;

			id	name	price	speed	added_by
<input type="checkbox"/>			1	nada	5.00	10	1
<input type="checkbox"/>			2	The kitchen Sink	10000.00	9999	2
<input type="checkbox"/>			3	nada	1.00	1000	1

DROP TABLE IF EXISTS Plan;

CREATE TABLE Plan (

id INT AUTO_INCREMENT NOT NULL,

name VARCHAR(255),

price DECIMAL(9, 2),







speed INT,

added_by INT,

PRIMARY KEY (id),

FOREIGN KEY (added_by) REFERENCES Employee(id) ON DELETE SET NULL

) ENGINE=InnoDB, CHARACTER SET=UTF8;

			plan_id	address_id
<input type="checkbox"/>			1	1
<input type="checkbox"/>			2	2
<input type="checkbox"/>			1	3

DROP TABLE IF EXISTS Address_Plans;

CREATE TABLE Address_Plans (

plan_id INT NOT NULL,

address_id INT NOT NULL,

PRIMARY KEY (plan_id, address_id),

FOREIGN KEY (plan_id) REFERENCES Plan(id) ON DELETE CASCADE,

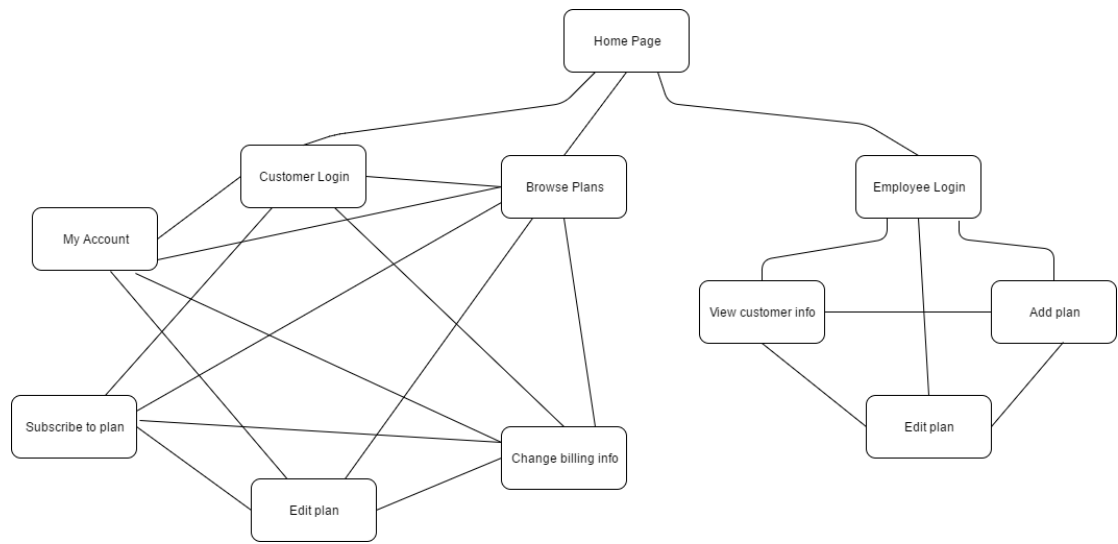
FOREIGN KEY (address_id) REFERENCES Address(id) ON DELETE CASCADE

) ENGINE=InnoDB, CHARACTER SET=UTF8;

4. Website Design

Discuss the design of your website.

- **Website Layout**



- **User Interface**

Home page



Our homepage allows users to login and access the rest of the site.

Customer login page

PEBCAK

Internet

[Our plans](#)

[Customer Login](#)

[Employee Login](#)

CUSTOMER LOGIN

Username:

Password:

Login

Customers can use this page to login and gain access to their account and plan pages.

Employee login

PEBCAK

Internet

[Our plans](#)

[Customer Login](#)

[Employee Login](#)

EMPLOYEE LOGIN

Employee ID:

Password:

Login

Allows employees to login and access plans.

Plans page

PEBCAK

Internet

[Our plans](#)

[Customer Login](#)

[Employee Login](#)

OUR PLANS

Plan name #1	Speed	Price	Available at address (only when logged in)
Plan name #2	Speed	Price	Available at address (only when logged in)

Shows plans to users and employees who are logged in.

Subscribe page

PEBCAK

Internet

[Our plans](#)

[Customer Login](#)

[Employee Login](#)

SUBSCRIBE

Plan Name

Speed

Price

Select billing info (drop down list)

Customer Portal

[My account](#)

[Manage my cards](#)

[My plans](#)

Allows users to subscribe to an Internet plan of their choosing.

- ***User Manual or Help page***

We have included a help page which can be found here:

<http://web.engr.oregonstate.edu/~morgaalb/cs340-project/public/help.php>

5. Application Implementation

We used a lot of HTML, PHP, and CSS to build our web site. We didn't use any JavaScript at all. There are a lot of tasks such as validation that JavaScript could do, but we used the built in HTML input validation regular expressions for this task instead.

We wrote a lot of queries that we could discuss as part of this project. My favorite query is a stored procedure that updates users' addresses. There is an important relation between addresses, customers, and plans in our database, and if an address is already in the database it is important to find that address in the Address table instead of inserting a new value. This stored procedure

does exactly that. It accepts a customerId and some address information (number, street, etc.). If looks in the database for this address, and if it finds it, it sets the customers address_id to that addresss. If it is not found, it inserts the new address into the database.

```
DELIMITER $$
CREATE PROCEDURE UpdateAddressCustomer(customerId INT, addNum
INT, addStreet
    VARCHAR(255), addAptNo INT, addCity VARCHAR(255), addSt
CHAR(2), addZip DECIMAL(5))
    BEGIN
        DECLARE addressId INT;
        IF NOT EXISTS(SELECT * FROM Address WHERE num = addNum
AND street = addStreet AND (addAptNo IS NULL OR apt_no =
addAptNo) AND city = addCity AND state = addSt AND zip = addZip)
THEN
            BEGIN
                INSERT INTO Address (num, street, apt_no, city,
state, zip)
                    VALUES (addNum, addStreet, addAptNo, addCity,
addSt, addZip);
            END;
        END IF;

        SET addressId = (SELECT id FROM Address WHERE num = addNum
AND
    street = addStreet AND (addAptNo IS NULL OR apt_no =
addAptNo) AND
        city = addCity AND state = addSt AND zip = addZip);

        UPDATE Customer SET address_id = addressId WHERE id =
customerId;
    END
    $$
```

Another interesting piece of SQL is the query that populates the plans page. In this query, the \$_SESSION['id'] is the id (auto-incrementing int primary key) of the customer. This query find all plans in the database, then cross references the Address_Plans table (which tells us which plans are available at which addresses), and if the plan is available at the customer's address, then the c.id column will be set to the customer's id. If it is not, then it will be set to NULL.

```
SELECT p.name, p.price, p.speed, c.id, p.id FROM Plan p LEFT
JOIN Customer c ON c.address_id IN (SELECT address_id FROM
Address_Plans WHERE plan_id = p.id) AND c.id = " .
$_SESSION['id'];
```

6. Evaluation

We had some friends who are also OSU students look at our website and give us some feedback. They were confused by some of the tasks such as signing up for a subscription at a location you are not at. We decided to remove this function because of the trouble that it caused the end-users.

7. Future Work & Lessons Learned

This is the first opportunity that the entire team had to write a trigger in a database. The biggest challenge that we faced was finding time in all of our busy schedules to get together to get the project done. We overcame this challenge by using Slack to communicate and coordinate our efforts. We also scheduled a couple of in-person work days to all get together and get the work done.

If we had this project to do over again, we would create a schedule of work earlier. This would allow us to figure out which things need to get done first, which would make everything easier later on. We had a little bit of trouble integrating everyone's different code, and better organization earlier on would help this problem

Appendix – Team Report *If you worked in a team summarize the division of labor.*

R. Hayden Anderson - Did the presentation and a lot of the original website architecture and some css.

Alex Ruef - The reports, some employee pages

Albert Morgan - Everything

Michael Elliott - Most of the css