**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.

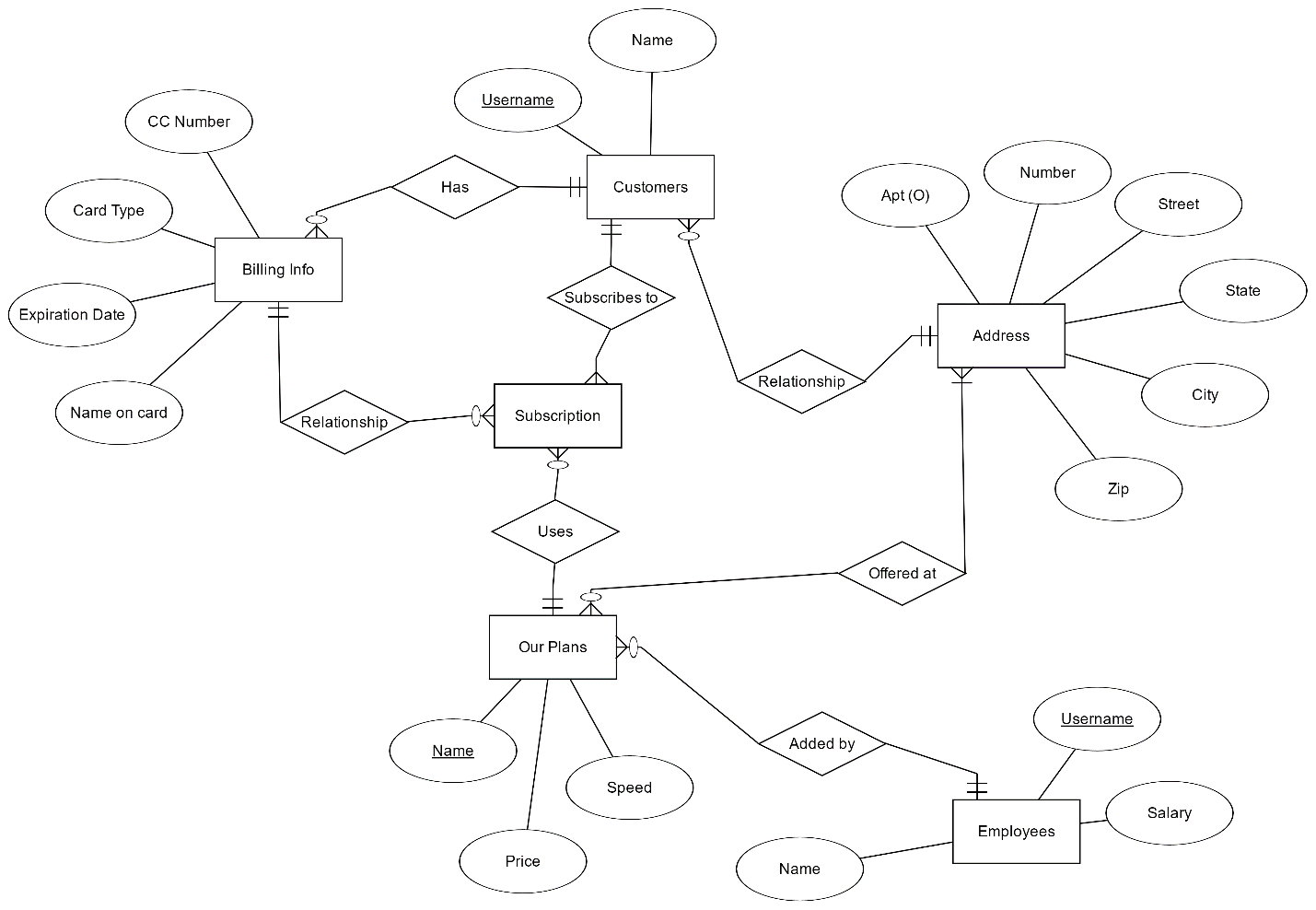
**2. Detailed Functionality & Requirements**

**Business rules**

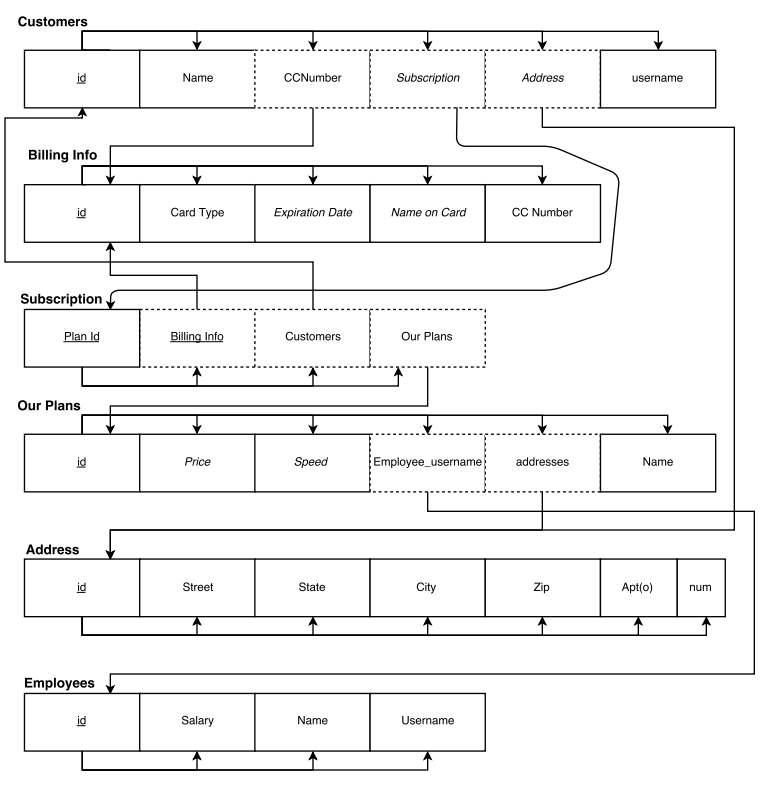
1. Customers must log in before being able to view any details about their plans or billing
2. Customers must be able to add new billing information when logged in
3. Customers must be able to add and remove subscriptions when logged in
4. Customers may have 0 or many plans
5. Customers may have 0 or many billing methods
6. A plan may not be signed up for at the same address twice.
7. Each subscription must have exactly one customer and exactly one billing method
8. Employees must login before viewing customer information or adding new plans
9. You do not need to log in to view all of the plans

**3. Database Design**

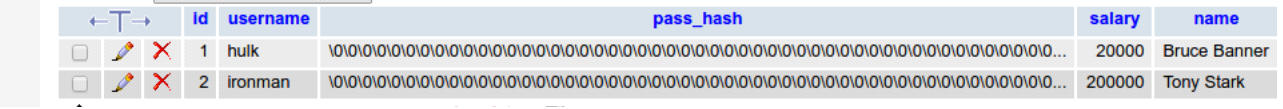
* ***ER Diagram of Database***



* ***Relation Schema***



* ***Database Tables***



DROP TABLE IF EXISTS Employee;

CREATE TABLE Employee (

id INT AUTO\_INCREMENT NOT NULL,

username VARCHAR(255) UNIQUE NOT NULL,

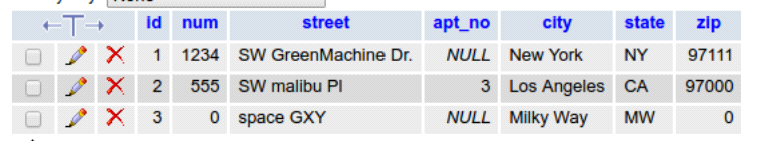
pass\_hash BINARY(60) NOT NULL,

salary INT,

name VARCHAR(255),

PRIMARY KEY (id)

) ENGINE=InnoDB, CHARACTER SET=UTF8;



DROP TABLE IF EXISTS Address;

CREATE TABLE Address (

id INT AUTO\_INCREMENT NOT NULL,

num INT NOT NULL,

street VARCHAR(255) NOT NULL,

apt\_no INT,

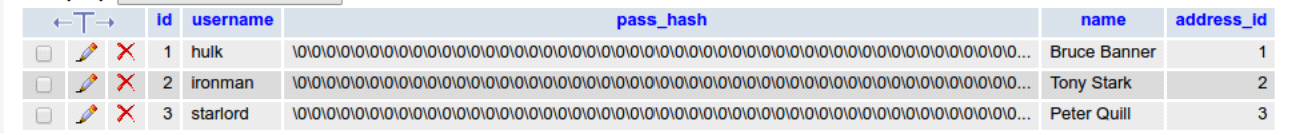
city VARCHAR(255) NOT NULL,

state CHAR(2) NOT NULL,

zip DECIMAL(5) NOT NULL,

PRIMARY KEY (id)

) ENGINE=InnoDB, CHARACTER SET=UTF8;



DROP TABLE IF EXISTS Customer;

CREATE TABLE Customer (

id INT AUTO\_INCREMENT NOT NULL,

username VARCHAR(255) UNIQUE NOT NULL,

pass\_hash BINARY(60) NOT NULL,

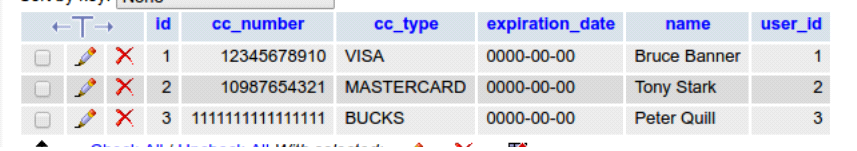
name VARCHAR(255) NOT NULL,

address\_id INT,

PRIMARY KEY (id),

FOREIGN KEY (address\_id) REFERENCES Address(id) ON DELETE SET NULL

) ENGINE=InnoDB, CHARACTER SET=UTF8;



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,

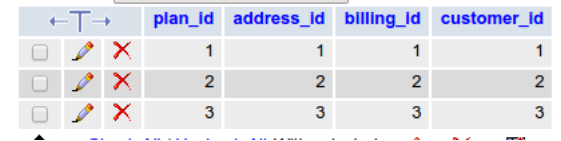
name VARCHAR(255) NOT NULL,

user\_id INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (user\_id) REFERENCES Customer(id) ON DELETE CASCADE

) ENGINE=InnoDB, CHARACTER SET=UTF8;



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;



DROP TABLE IF EXISTS Plan;

CREATE TABLE Plan (

id INT AUTO\_INCREMENT NOT NULL,

name VARCHAR(255),

price DECIMAL(9, 2),

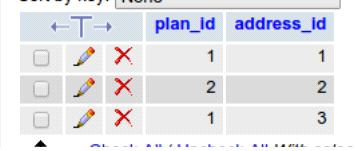
speed INT,

added\_by INT,

PRIMARY KEY (id),

FOREIGN KEY (added\_by) REFERENCES Employee(id) ON DELETE SET NULL

) ENGINE=InnoDB, CHARACTER SET=UTF8;



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),

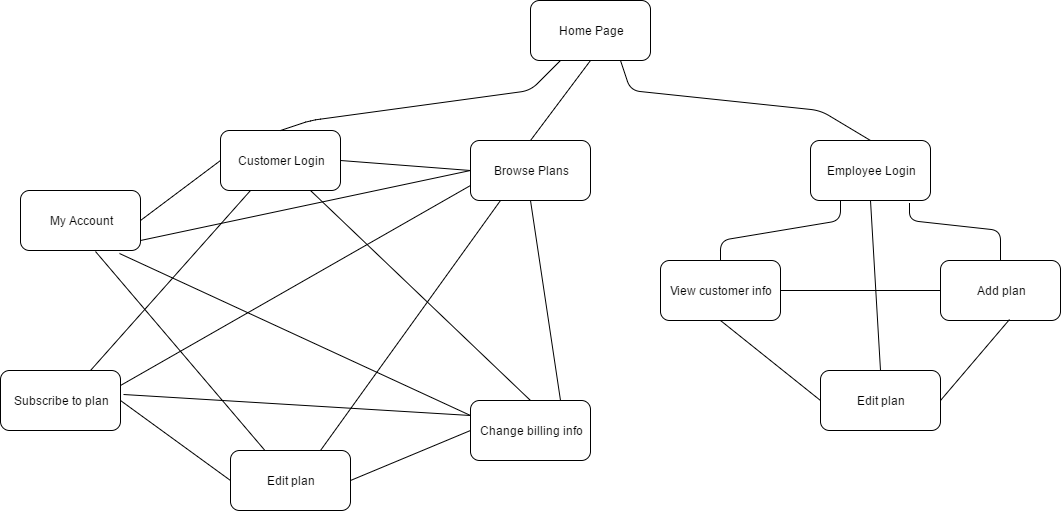
FOREIGN KEY (plan\_id) REFERENCES Plan(id) ON DELETE CASCADE,

FOREIGN KEY (address\_id) REFERENCES Address(id) ON DELETE CASCADE

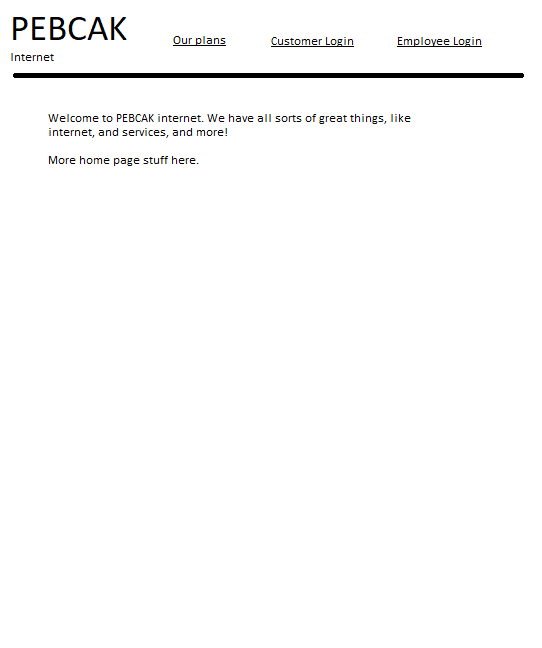
) ENGINE=InnoDB, CHARACTER SET=UTF8;

**4. Website Design**

Discuss the design of your website.

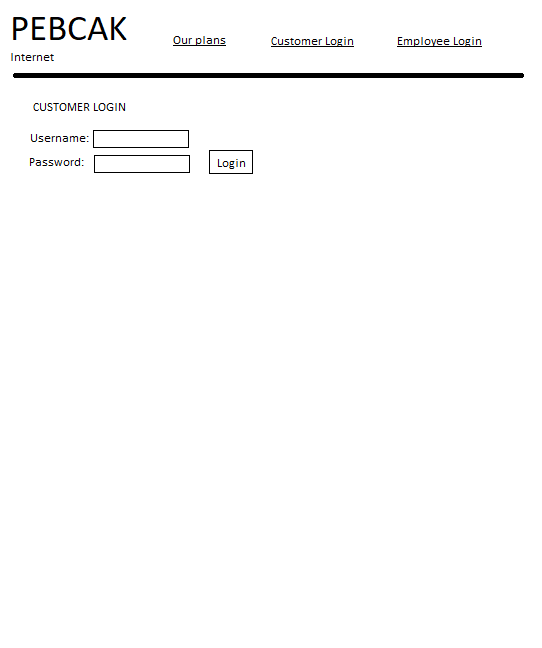
* ***Website Layout***
* ***User Interface***

**Home page**



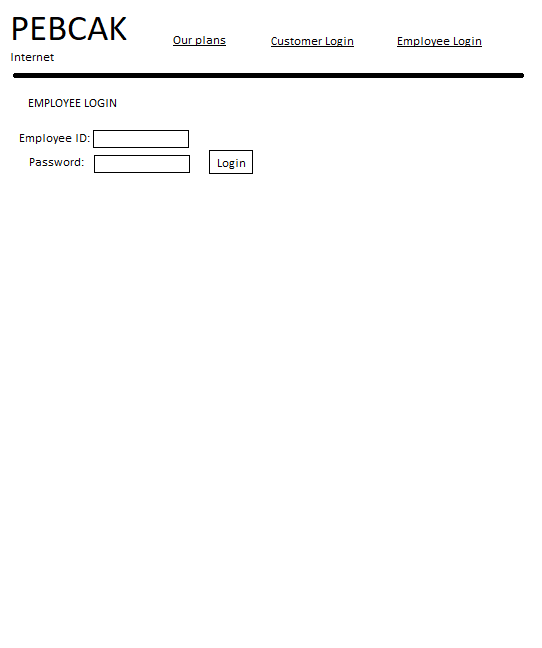
Our homepage allows users to login and access the rest of the site.

**Customer login page**



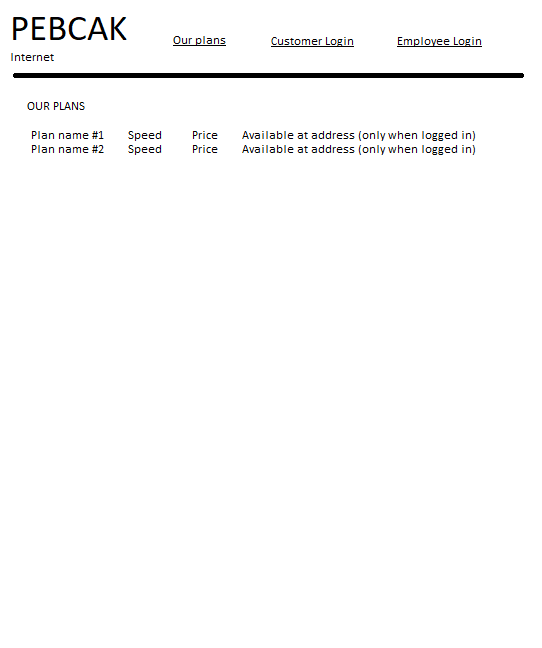
Customers can use this page to login and gain access to their account and plan pages.

**Employee login**



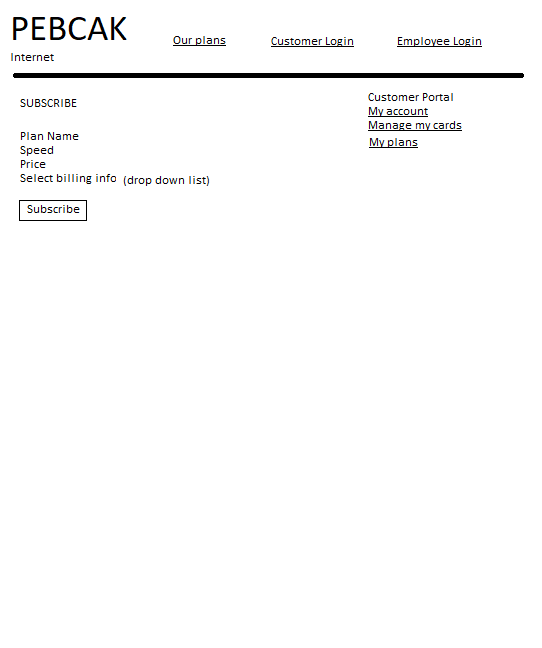
Allows employees to login and access plans.

**Plans page**



Shows plans to users and employees who are logged in.

**Subscribe page**



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 Adddress 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