Date:

March 13th, 2017

Team Members:

Ashish Shrestha

Kenneth Balogh

Application URL:

http://resin.cci.drexel.edu:8080/~as3828

Login Details:

Email: **joedrexel@drexel.net**   Password: **default**

Code:

See README.md for code implementation details

Project Description:

For our project, we have chosen to implement a Food Ordering System. Our application will provide a database of restaurants that allows a user to order food from one of them. Similar to services such as Grubhub or UberEats, we modeled entities that allow to user to get as much information about the restaurants as possible. One unique feature is that we allow a user to select foods based on taste such as sweet, sour, hot ,etc. The following entities were modeled: Restaurants, User, Menu, Invoice, Ratings, Cuisine, Taste and Payments.

Entity Sets, Relationship Sets and Business Rules:

Restaurants:

- The Restaurants has a primary key: restaurant id (rid)
- The other attributes are: name, address, phone number
- A Restaurant has at least one Menu
- A restaurant has at least one Cuisine

Users:

- The User has a primary key: user id (uid)
- The other attributes are: name, address, phone number, email
- A user has at least one payment
- User rates zero, one or more Restaurants with stars, dates and comments

Menus:

- The Menu has a primary key: menu id (mid)
- The other attributes are: name, description, price
- A menu has at least one taste

Invoices:

- The Invoices has a primary key: invoice id (tid)
- The other attributes are: date, description, amount
- An invoice has only one payment
- An invoice has only one user
- An invoice has only one restaurant

Cuisines:

- The Cuisines has one attribute that is its primary key: cuisine name (name)

Tastes:

- The Tastes has one attribute that is its primary key: flavor (flavor)

Payments:

- The payments has a primary key: card number (card#)
- The other attributes are: expiration date, cvc number and zipcode

- A payment only has one user

ER Translation

We utilized the ER diagram to translate into the relational schema. Based on the diagram we knew that Users, Cuisines, Tastes and Restaurants did not require any foreign keys so we built those tables first. Then we built the remaining tables ensuring that we had the proper foreign keys identified.

Data Acquisition Approach

When building the data tables we looked at real restaurants around the university. We used the restaurant' name, address and phone number when populating the restaurants table. We looked at the restaurants menus and created items so that we could demonstrate different searches such as cuisine and tastes. Generic invoices were created in the Invoices so that we could demonstrate the "Past Orders" query. We also created generic payment options in system so a user could select a payment at checkout options. Finally, all this data was loaded into a script (schema.sql) so that the data could be loaded in the database.

Overview of the System

The user of the system is prompted with a login page as shown in Figure 1. This requires the user to enter an email and password. We populated the database with ten different users. The user with the most entries is Joe Drexel. The email is **joedrexel@drexel.net** and the password is: **default**. Additional logins are listed in the Readme.md or can be seen in schema.sql. Based on the user's information this is the first search done on the database looking for both an email and password that must match.
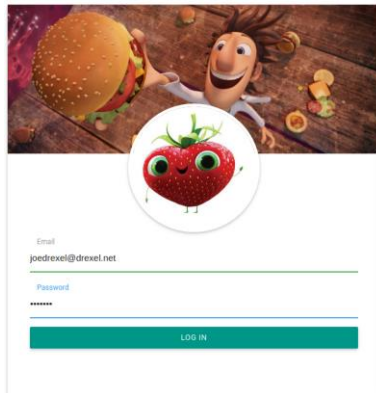
Once logged in the user is presented with a list of restaurants they can order from, along with the average rating of each restaurant. The user can search for different cuisines or flavors to narrow the query results as seen in Figure 2. Depending on the selected options it will run several different queries on the database to show the results. The user can also see their previous orders by clicking the *Past Orders* button on the left hand side of the screen. This runs the a query to display all the orders from that user and totals up all the orders at the bottom of the page. These are just a few of the queries that are run to display information to user.

The user can click the *Order* button for a selected restaurant. This returns a list of all the menu options at that restaurant. Restaurants such as Chipotle and BBQ Ribs have the most items loaded. Figure 3 shows an example of a menu item that is selected. Once *Place Order* is clicked a user is displayed the Checkout Option(Figure 4) which lists any payments they have stored. After selecting *Use This* an invoice is generated and inserted into the database.

The system also allows users to see ratings for restaurants by clicking the stars rating as shown in Figure 7. Finally, the user can click *Account* to display all the users information as shown in Figure 6. Each of these runs a different query on the database to display the corresponding information.
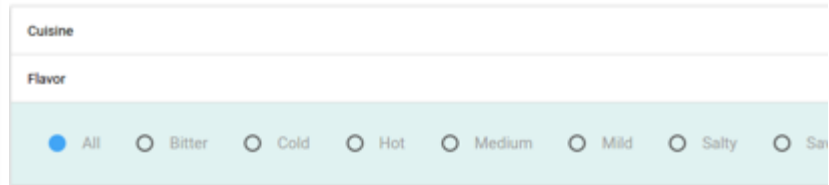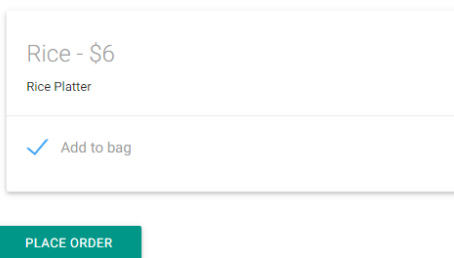
Figure 1 Login Page
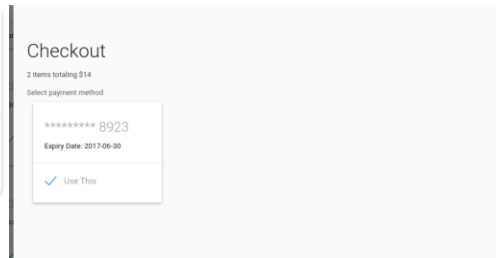
Figure 2 Example of Selecting Flavor or Cuisine
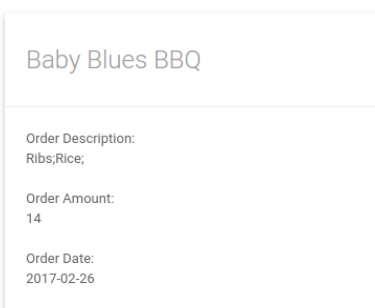
Figure 3 Menu Selection
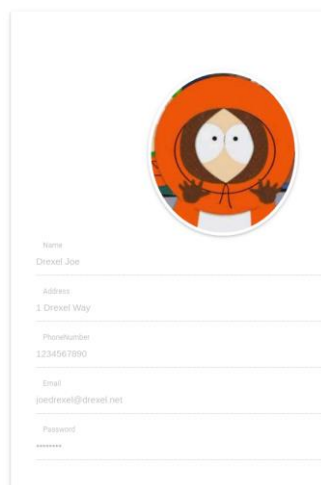
Figure 4 Payment/Checkout

Figure 5 Invoice

Figure 6 Summary Page

Figure 7 Clickable Ratings
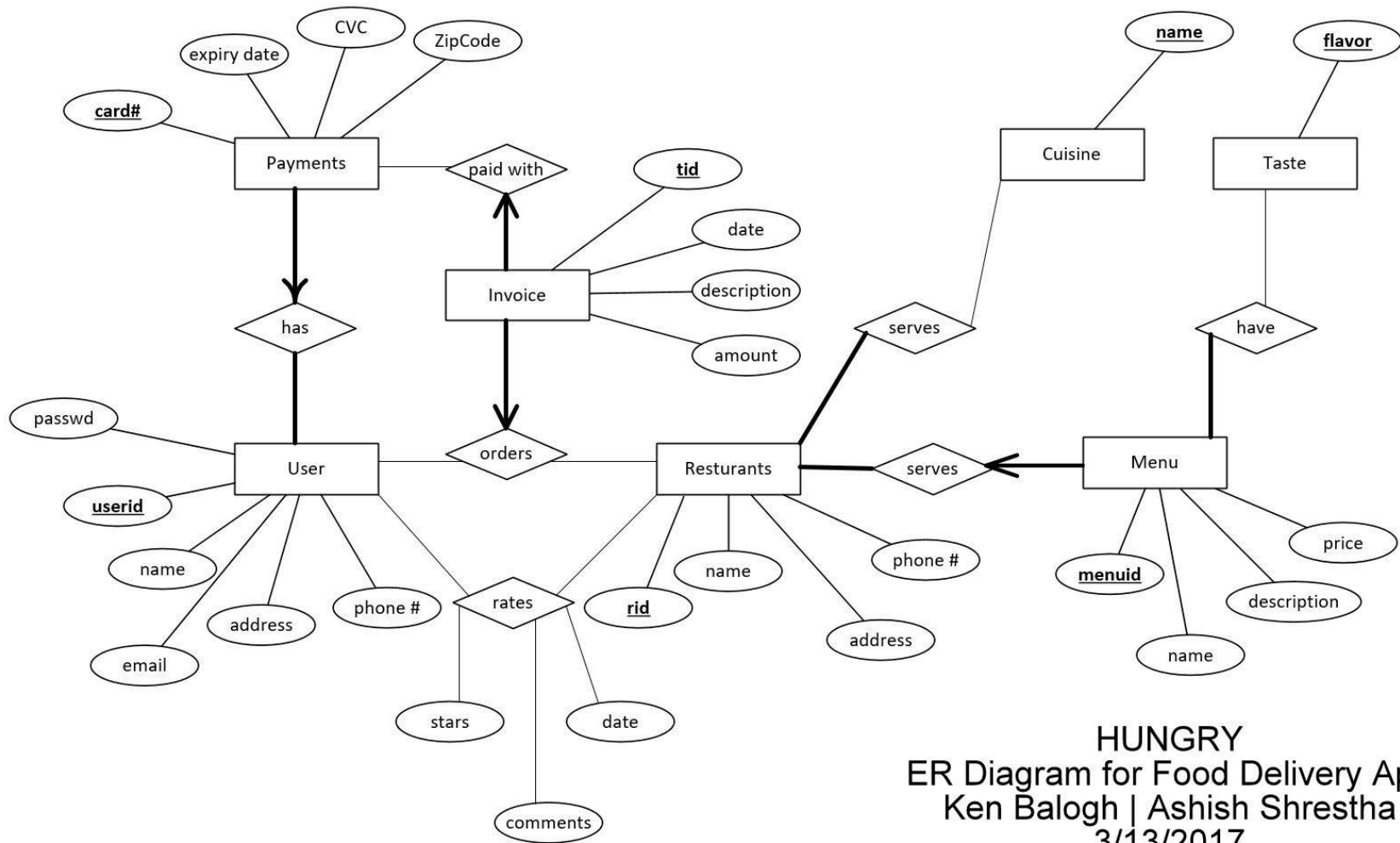
HUNGRY
ER Diagram for Food Delivery App
Ken Balogh | Ashish Shrestha
3/13/2017