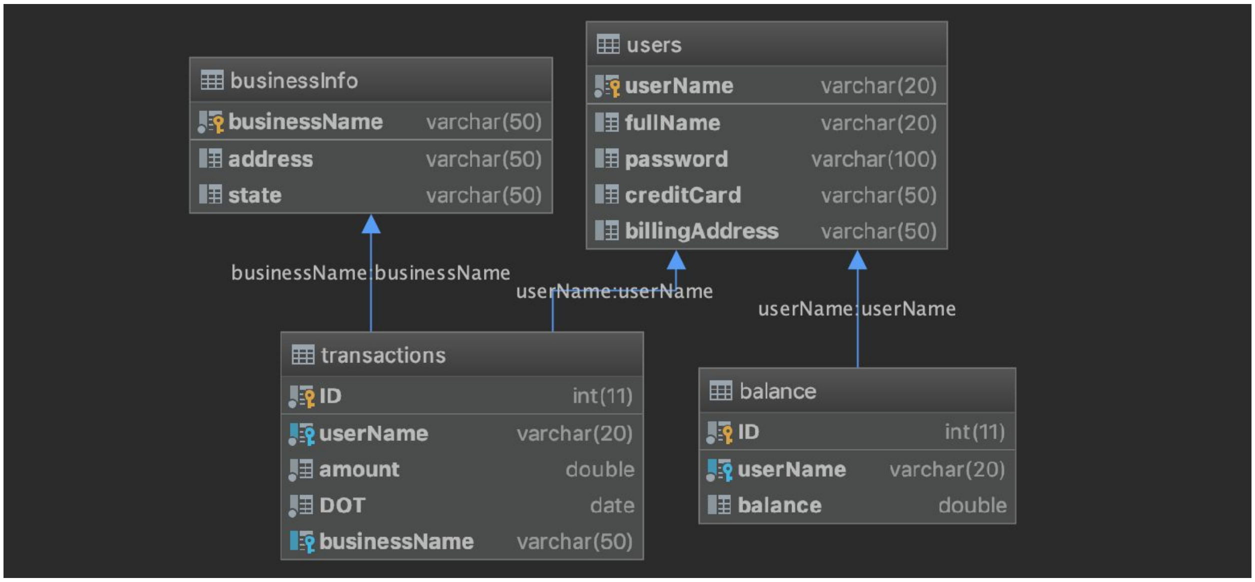
Karl Hickel
Josh Anderson
Database Management
CPSC 408
German,Rene
May 15, 2019

Database Final Project

In our very busy world we often find ourselves spending more money than we realize, in doing so we often forget to keep track of our expenditures. Even if we have access to our past expenditures, it can be difficult to analyze and follow certain trends of our spending habits. So we wanted to create an application that solved this 21 century problem of ours. We brainstormed extensively to figure out what would be the best approach to solving this issue. After a long thought out process we decided to create a web application that cohesively stores your transactions, income and provides basic analytical graphs to give a better understanding of your account. To achieve our goal we utilized a Python framework called Django, HTML, CSS, MySQl, and a little bit of JavaScript. (Google graphing API) We have not personally used any personal banking applications that perform the same functionalities as our project. However we have seen some website that manage personal stock and provide an analytical overview at the end of each month.

We imported a Django framework to interface primarily with our HTML code. To create our Databases we utilized DataGrip and stored our tables on Josh's personal server. To create our data we used a faker library that we found on github to randomly generate transaction amounts and to produce fake credit card numbers, street addressed, dates, business names, states and a few other attributes. An initial complication that we had encountered was that the randomizes did not organize the dates in chronological order so we had to rearrange it utilizing excel. After creating our data we stored it into our tables using an apache library in Intellij. It would have been easier to simply import the CSV into DataGrip but we were unaware of this functionality until after we stored our data.



businessInfo	
businessName	varchar(50)
address	varchar(50)
state	varchar(50)

users	
userName	varchar(20)
fullName	varchar(20)
password	varchar(100)
creditCard	varchar(50)
billingAddress	varchar(50)

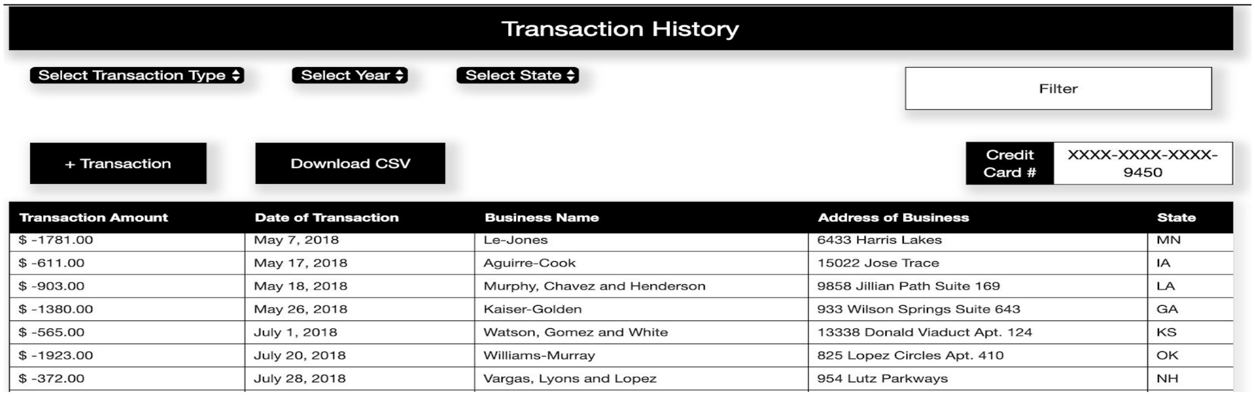
transactions	
ID	int(11)
userName	varchar(20)
amount	double
DOT	date
businessName	varchar(50)

balance	
ID	int(11)
userName	varchar(20)
balance	double

In our schema diagram we to ensure we maintained referential integrity utilizing 3NF. Our users table is our main table that extends to the users current balance in table balance. It also extends into our transactions table with username as a foreign key. The purpose of separating these tables is that we wanted to make sure we could query from our users table without having the lengthy records that we would find in our transactions table.

After creating our tables we shifted our attention to creating the basic interface of our website. We primarily utilized HTML and CSS to make our website look presentable. Initial complications arose when Karl's Pycharm was only compatible with Python 2.7 and Django utilizes Python 3.X. There was a relatively easy workaround so we managed to fix that complication. We also utilized a little bit of JavaScript for our analytics page. This was relatively complex in implementation, but we managed to make it presentable.

Our layout for the website is relatively straightforward, after creating an account or signing, the user has the capabilities to view his/her past transactions. We laid our our query and merged 3 different tables to make it more informative. We included their transaction amount, date of transaction, business name, address of business, state where business is located, and their credit card number. We wanted to present as much information as possible to the user without making it appear unpractical. On the same page the user has the ability to filter through different categories in order to see a desired timeframe, income/expense, or state where transaction was made. On top of that the user can also download a csv file of their transactions even if they have applied filters. This was Karl's first time using CSS and HTML so there was a slight learning although because Josh has prior experience with web development this allows the experience to go by a lot smoother than expected. We built the entire interface from scratch without using an existing template.



Transaction History

Credit Card # XXXX-XXXX-XXXX-9450

Transaction Amount	Date of Transaction	Business Name	Address of Business	State
$ -1781.00	May 7, 2018	Le-Jones	6433 Harris Lakes	MN
$ -611.00	May 17, 2018	Aguirre-Cook	15022 Jose Trace	IA
$ -903.00	May 18, 2018	Murphy, Chavez and Henderson	9858 Jillian Path Suite 169	LA
$ -1380.00	May 26, 2018	Kaiser-Golden	933 Wilson Springs Suite 643	GA
$ -565.00	July 1, 2018	Watson, Gomez and White	13338 Donald Viaduct Apt. 124	KS
$ -1923.00	July 20, 2018	Williams-Murray	825 Lopez Circles Apt. 410	OK
$ -372.00	July 28, 2018	Vargas, Lyons and Lopez	954 Lutz Parkways	NH

Another key functionality in our project is the ability to change details about the user account. The user has the ability to change their full name, credit card number, delete their account and change their password. In order to accomplish this we created a number of SQL procedures that allowed us to create these changes with ease. We also implemented a couple of analytics graphs in our project to inform the user about their transactions. One is a bar graph that counts highest number of states where transactions were made. The second graph tells the user their average income and their average expenses.

Going forward into the future we would like to create a function that filters the analytics so we could provide those graphs based on year, state, and or a full bar graph of their transactional history. Other functionalities we would like to add would be in regards to the profile of the individual. Possibly allow the user to upload a profile photo or have multiple credit cards stored on the account. Another potential area for improvement would be our home page. Currently our home page is kept at a bare minimum and there isn’t a lot to look at, but going forward possibly including some kind of interactive graphic would be beneficial going forward.

In conclusion, this we challenged ourselves to create the best transactional history web application that we possibly could within our time constraint. There were a number of compatibility issues that temporarily halted progress but we overcame. Our desired goals were met and now we have a fully functional web application that effectively records expenditures and incomes.