eShopCart

Parth Sutaria

Daniel Lecheler

# DELIVERABLES I:

# Task Delegation

Parth Sutaria: EER diagram, schema diagram, relational model mapping, database implementation

Daniel Lecheler: Introduction, comparison, research, EER Diagram

# Introduction

Since the rise of the internet and as access to it became widespread, entrepreneurial companies have presented the consumer with the option to order goods and services remotely with relative ease. The ease of use for the user shopping online is limited to the specific website being used, with switching between sites becoming a great annoyance and a security risk, now that company hackings are becoming commonplace. We purpose a universal cart and payment system that allows the user to manage virtual shopping carts across multiple sites and store personal information including payment methods in a singular, secure database. According to the United States Census Bureau’s Quarterly report released last month, the percentage of online (e-commerce) retail sales of total retail sales is at 8.5%, which has steadily increased annually since 2007 (5% in a decade). With increasing amount of shopping being done online, we believe that out database would increase security and user productivity, while also greatly reducing in the amount of login information the average individual has to remember. We selected this concept out of a desire to make our own shopping experiences more pleasant and efficient.

# Comparison

The closest widely implemented and well known comparison to our concept of a universal shopping cart database is that of PayPal. [1] PayPal stores user’s payment information to make payments between individuals and small scale retailers (or even other individuals) more secure and less likely for one party to take advantage (or potentially rob) the other party. Where PayPal falls short in comparison to our purposed database is that PayPal is implemented in unison with the specific company/website’s account often making things even less convenient for the customer. By focusing on the user’s comfort over security (not neglecting security though), the implementation of a cross-website shopping cart can remove the time-consuming portions of shopping online (multiple accounts, orders, and payments to different websites) without also sacrificing the ability to shop for the best deals on the whole internet.

In an article on Wired.com, the Keep mobile app, created by a startup in New York, in principle mirrors our concept in the form of a smart phone application. The article claims “shoppers open the Keep app, click a web icon, browse stores, and add items. Once those items are in the Keep store, they click ‘Buy,’” with “the next release of the product will be even simpler…in that version, when shoppers browse other online stores, a “Buy” button generated by Keep will automatically appear at the bottom of the page, though making that happen turns out to be a major technological and design challenge.” [2] Despite the similarities between our database and the Keep app, this article was published almost three years ago, and has yet to gain more media or public attention. Similarly, Amazon Pay debuted in 2007, and is designed to reduce checkout time among customers of Amazon and Amazon subsidiaries, but in

the decade from the release has yet to become as successful as other Amazon services like Amazon Prime or Amazon Pantry. [3] In summary, the successful implementation of a mainstream universal internet shopping cart has yet to happen and we intend to improve on our predecessors’ mistakes to create a usable database that’s not limited or discriminatory towards any one website.

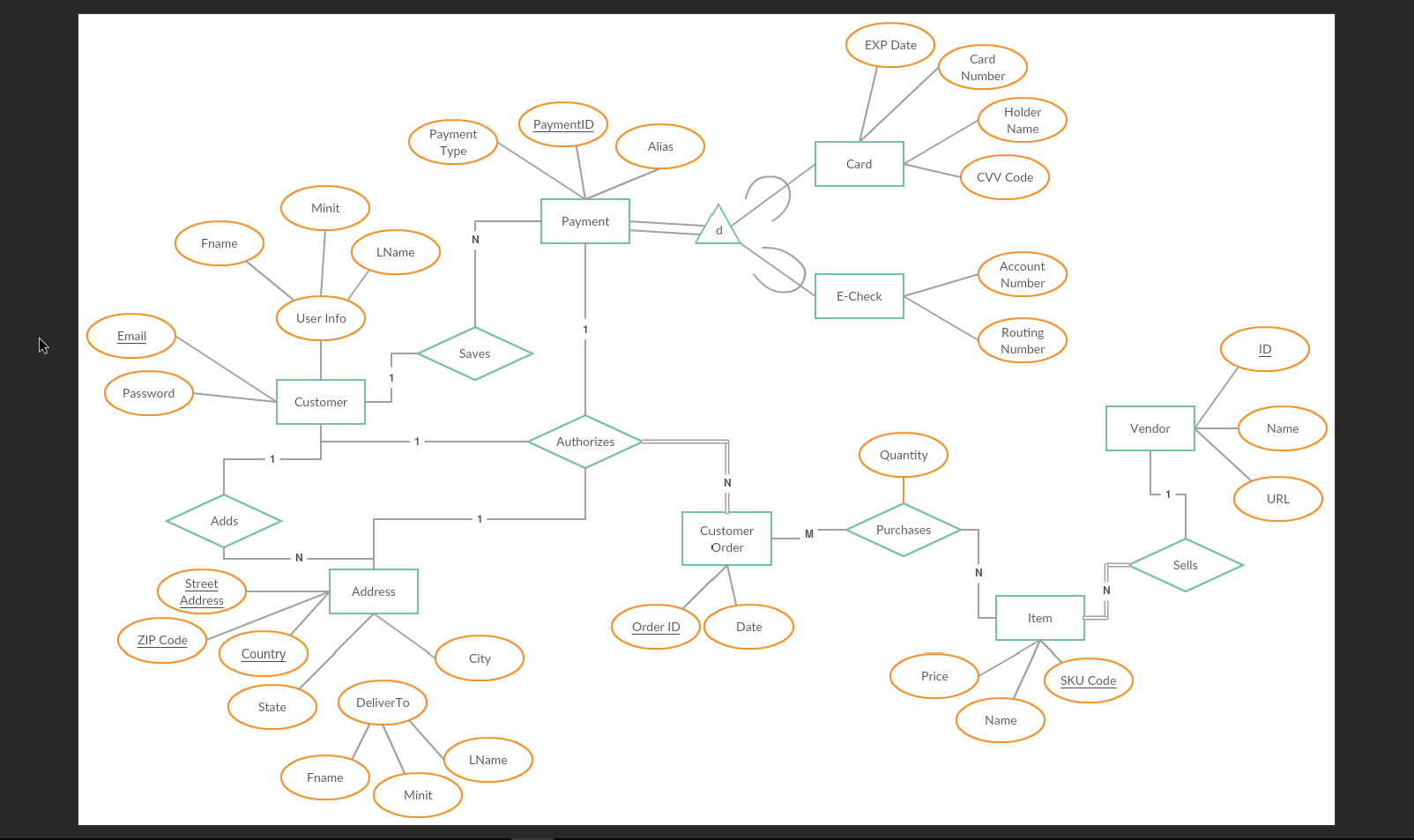


Figure 1 EER Diagram.

# DELIVERABLES II:

# Task Delegation

Parth Sutaria: Java FX Code, Debugging, Consultation and Corrections on Dependency Diagrams & Normalization

Daniel Lecheler: Dependency Diagrams, Creation of Views, Normalization, Report, Assistance at Debugging

# Final Report:

The eShopCart is a combination of Java FX code that uses the SQL code for the database written in SQL Management Studio. The normalized EER diagram, shown in Figure 2, has changed very little from the previous iteration of the diagram as shown in Figure 1. The change refers to the Recipient entity, previously known as the address entity. The entity now uses a generated value like the Customer Order Entity. This is because although the address could be uniquely identified by some combination of other attributes of the address, that property of the entity may not hold across various countries. The Database in its 3NF normalized state has 10 tables: item, vendor, itemOrder, authorizedOrders, payment, eCheck, custOrder, customer, recipient and card. Item represents a given product that the user might desire to place in there cart to order at the conclusion of their shopping. Item is related to vendor that are identified by their name, URL, and unique ID; and related to itemOrder that is an order placed by the user on completion of shopping. ItemOrder is related to CustOrder that serves as a table of orders and the corresponding dates they were placed. AuthorizedOrders are CustOrders, except with date being substituted for email. Customer, address, and payment are all related to the AuthorizedOrders entity; customer includes personal details such as name, email, and password; address is the location the order will go to; payment is the details need to make an order. Payment is related to e-checks and cards with their respective details. Recipients is the name and email of the person receiving the items ordered.

When the eShopFX.jar (located at eShopFX/dist/eShopFX.jar) is executed prompts the user with a login screen that requires inputs for user email and password, or to register as a new customer/user of the eShopCart. If the user chooses to register, the display changes to that of prompts for a valid email, password, and name (first, middle initial and last) to be entered before continuing to the User Info page. The User Info page is displayed and for the current logged in user (whether they choose to login or register), the options to alter the user’s (represented by his or her email) name or password are available, along with the delete account function. The User Info page includes a logout button, and the ability to change to the Addresses screen. The Addresses tab lists the possible destinations the user has saved to their account, and offers the user the ability to add, remove, or edit the stored shipping locations.

# Normalized (3NF) EER:

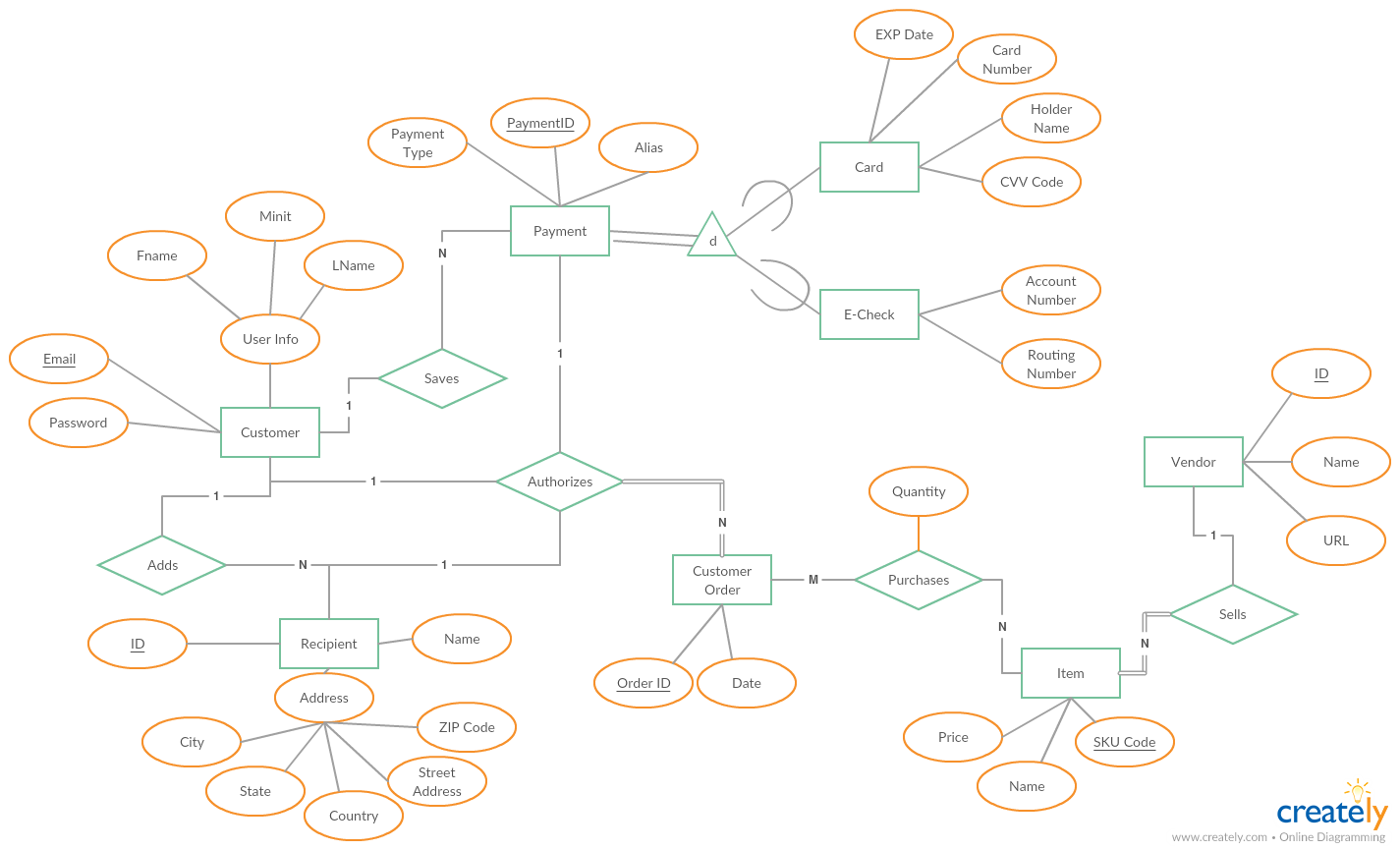


Figure 2: Normalized EER Diagram

# Screenshots:

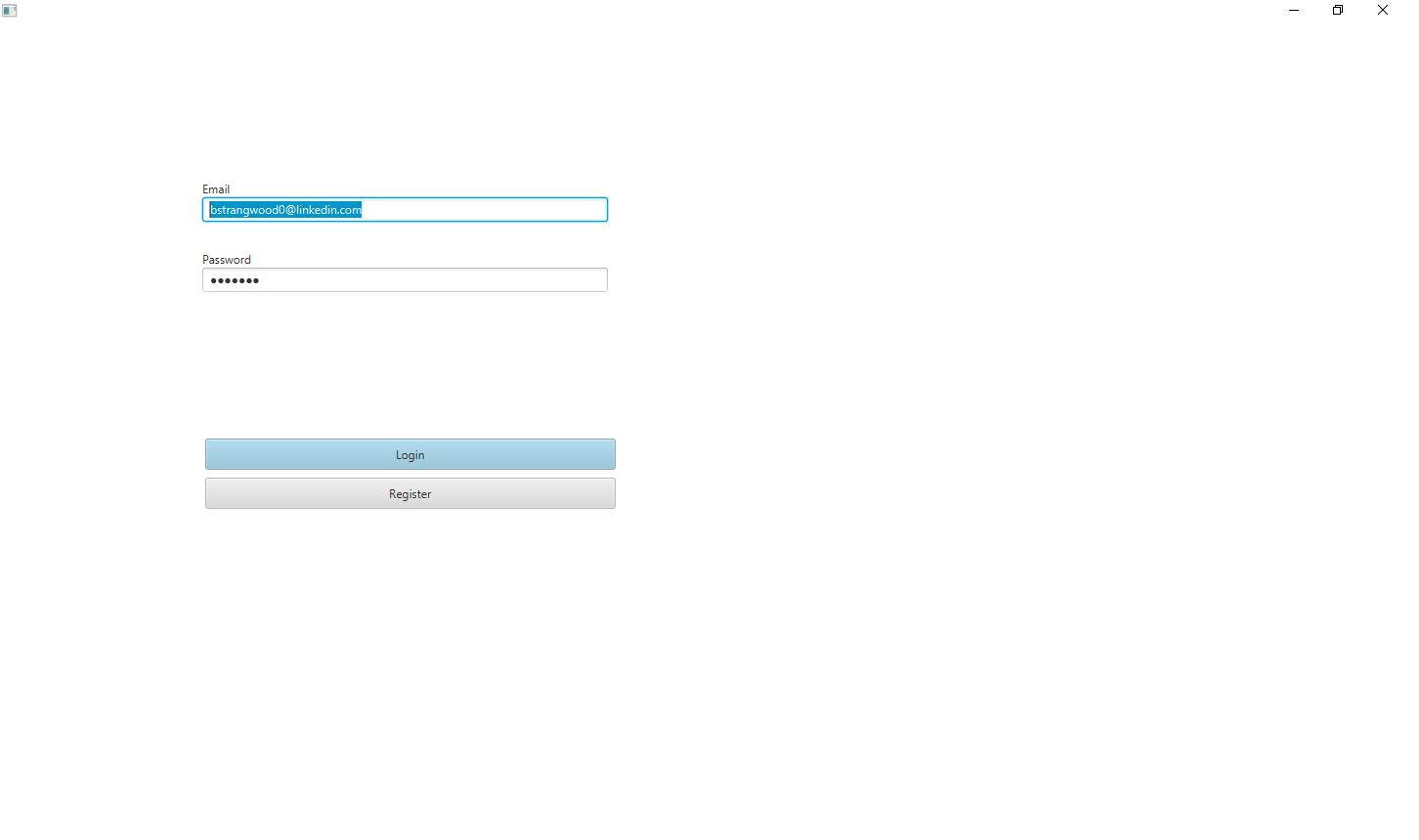


Figure 3: Login Screen

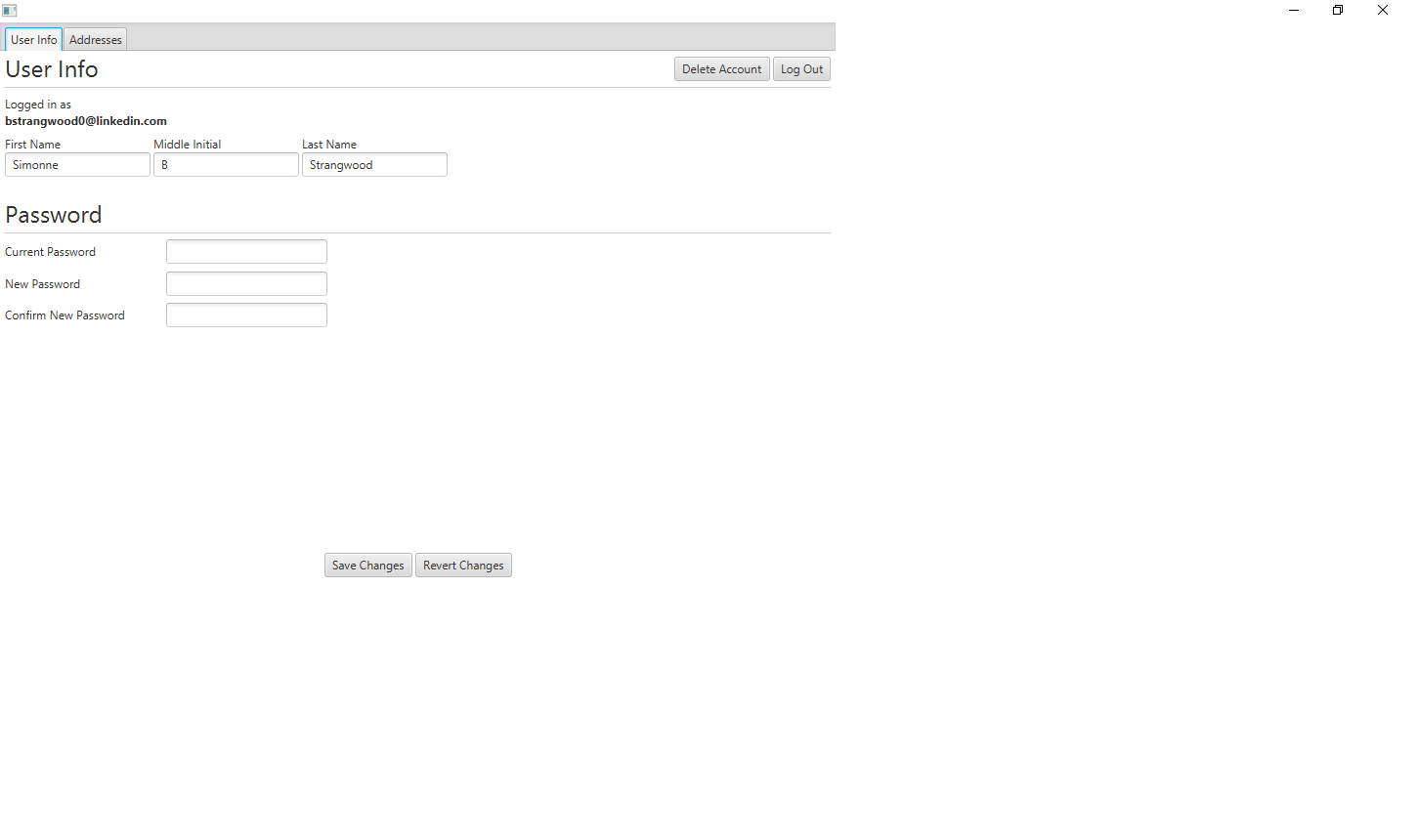


Figure 4: Logged in as an existing user

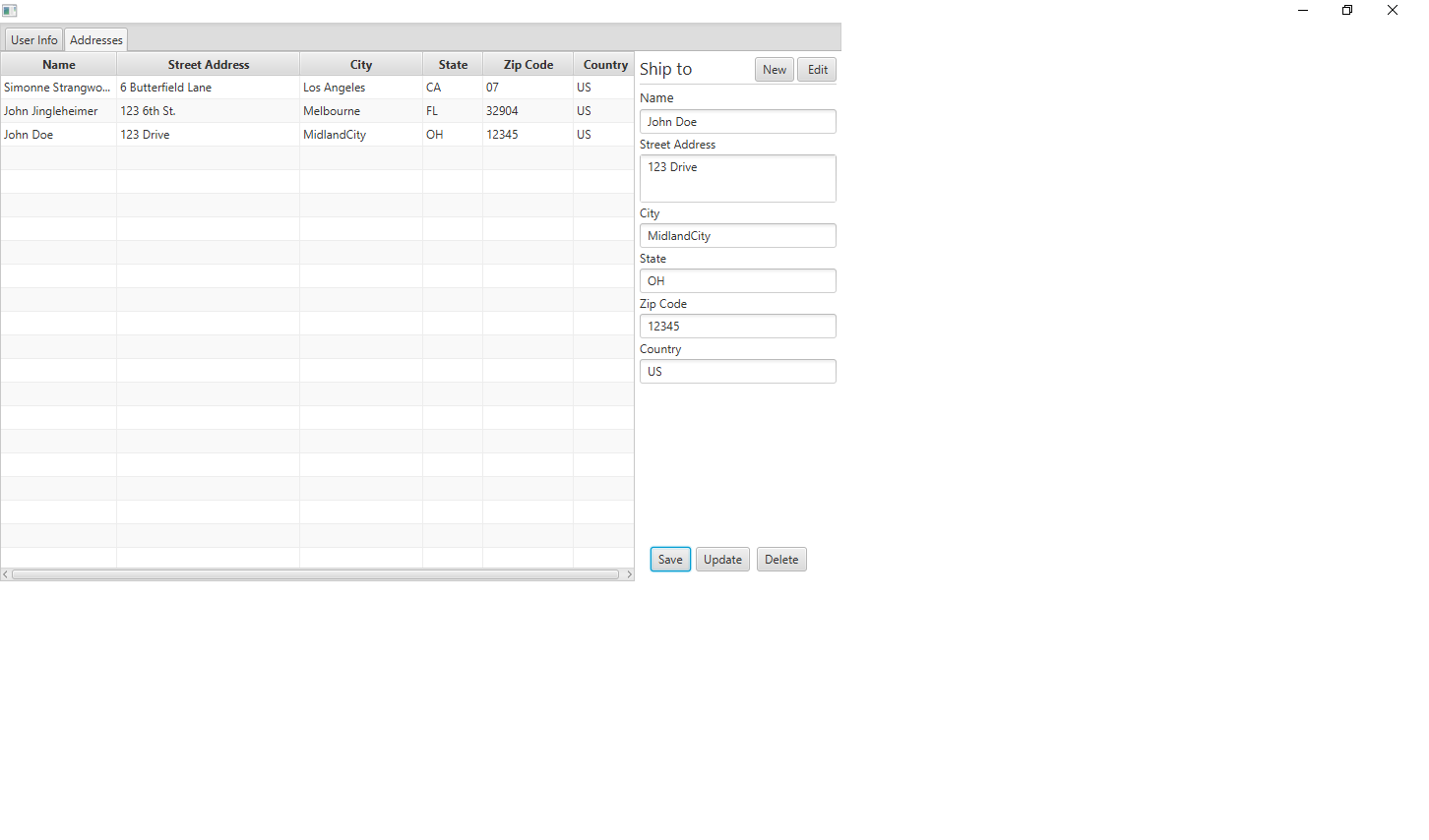


Figure 5: Existing user’s addresses

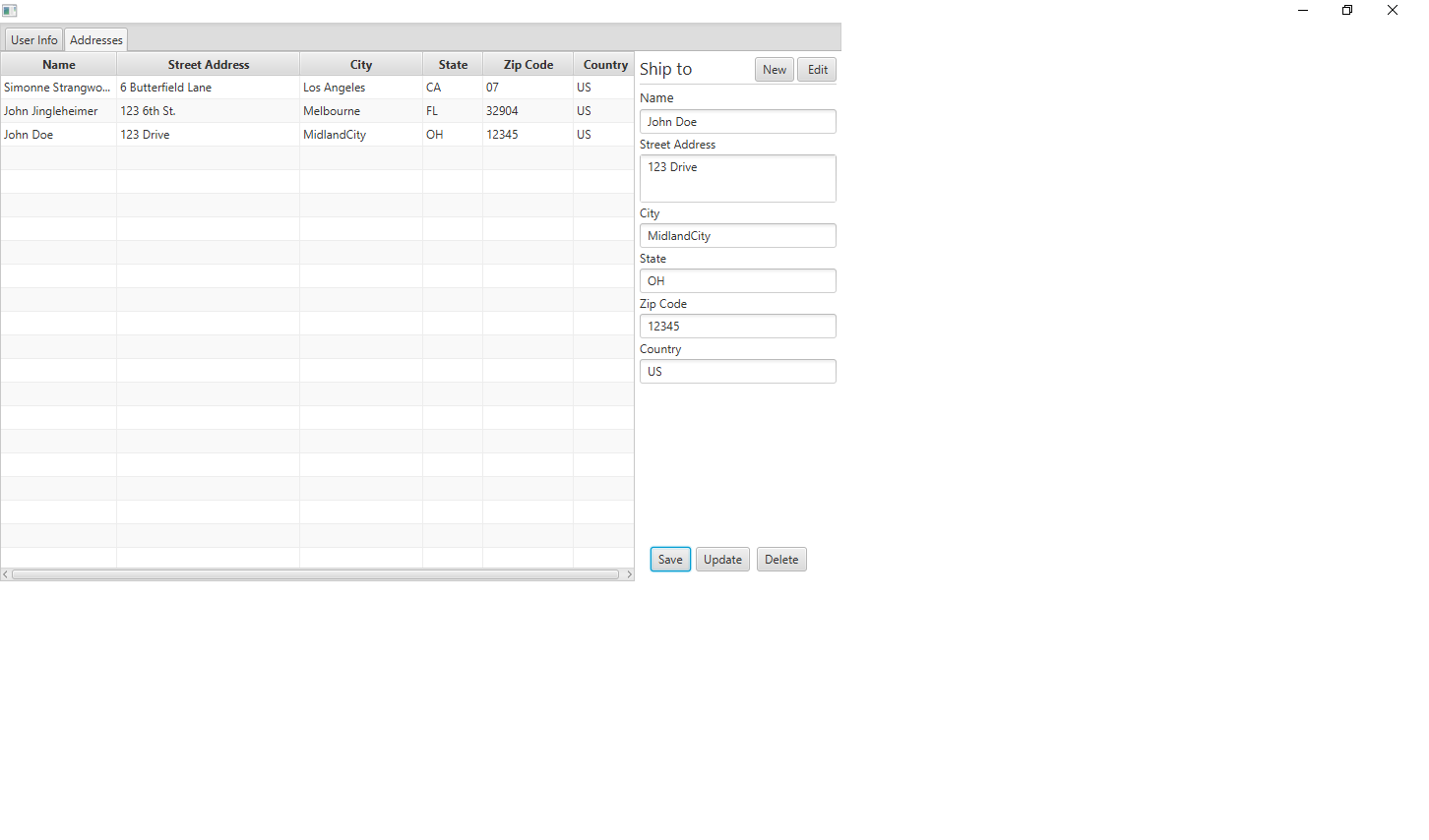


Figure 6: Adding a new address

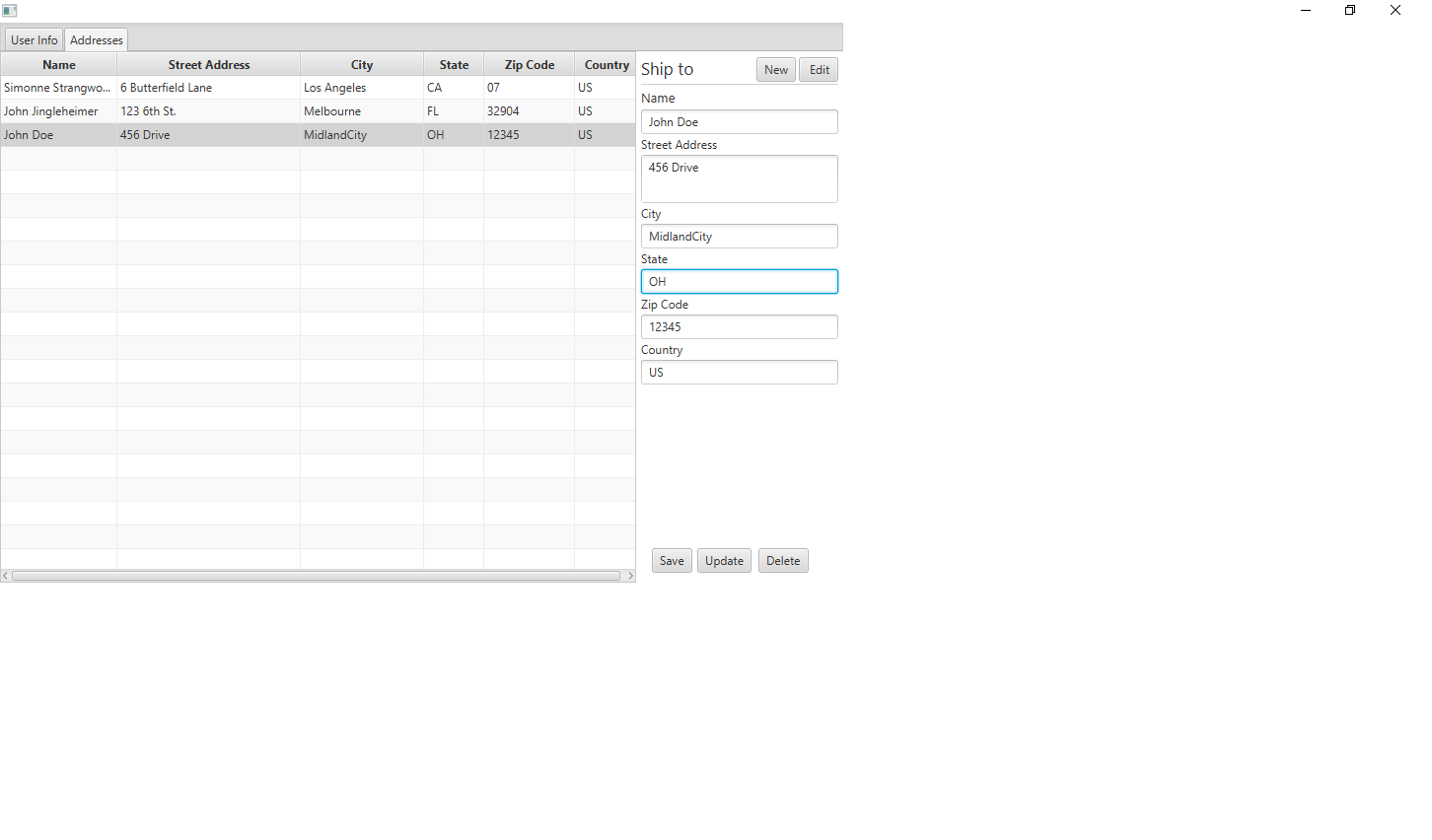


Figure 7: Updating an Address:

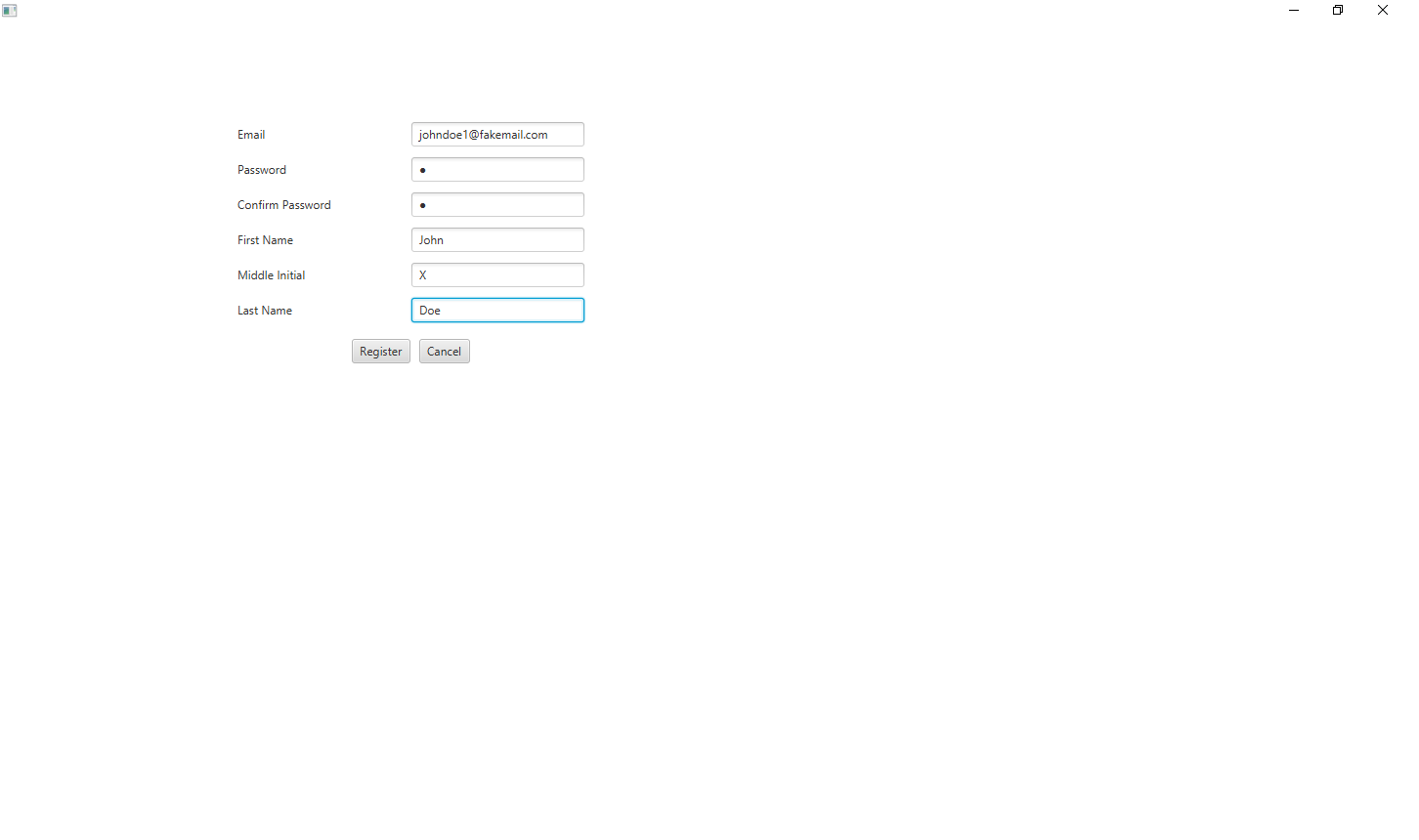


Figure 8: Registering

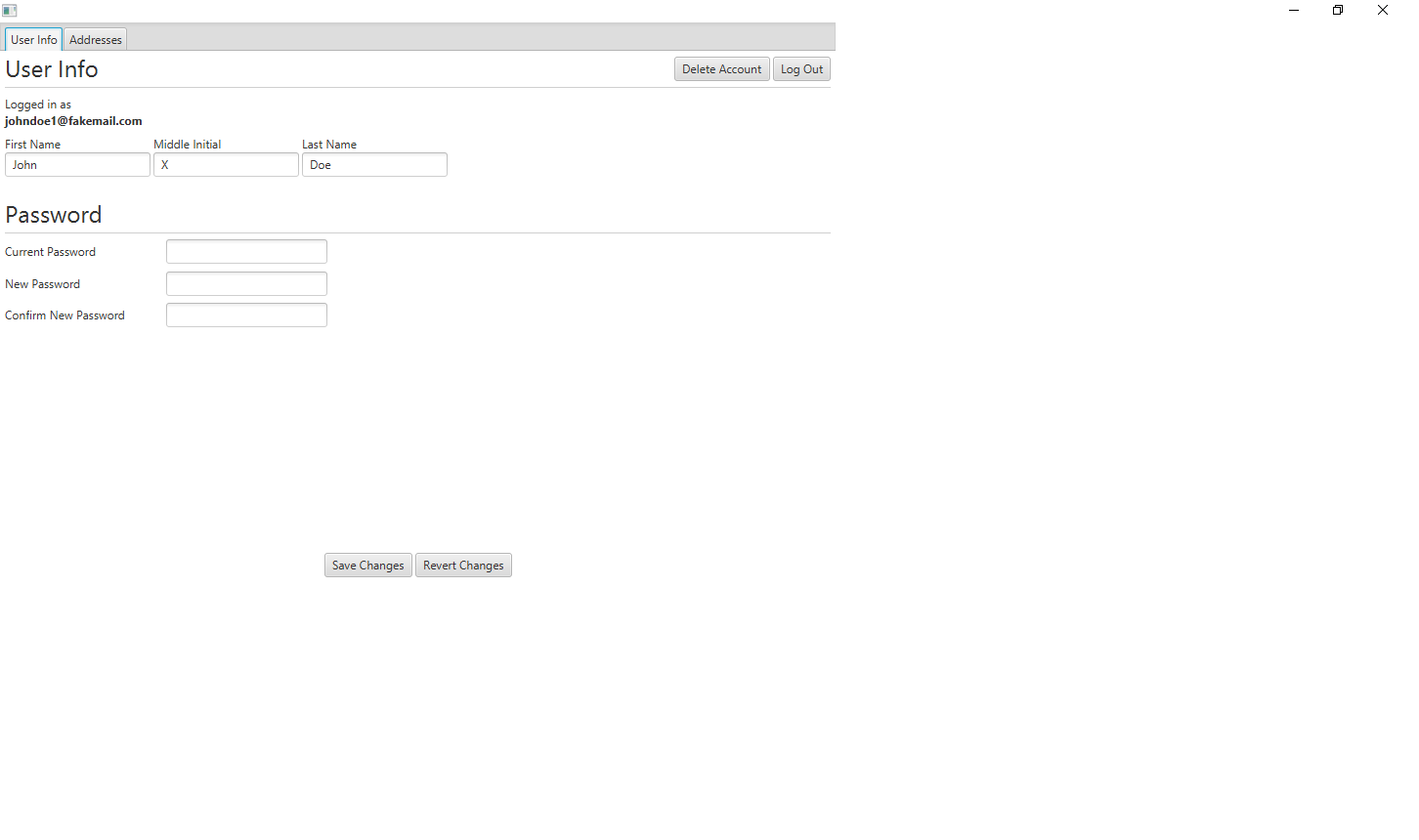


Figure 9: Logged in as new user:

# Dependency Diagrams:

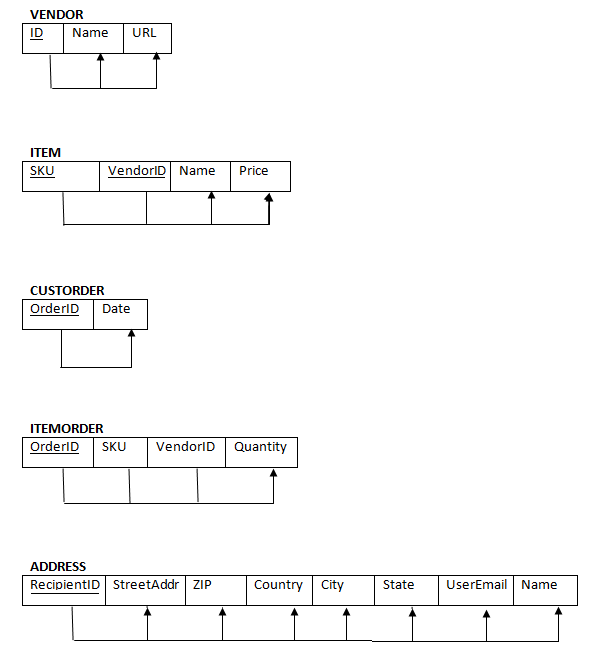


Figure 10: Dependency tables

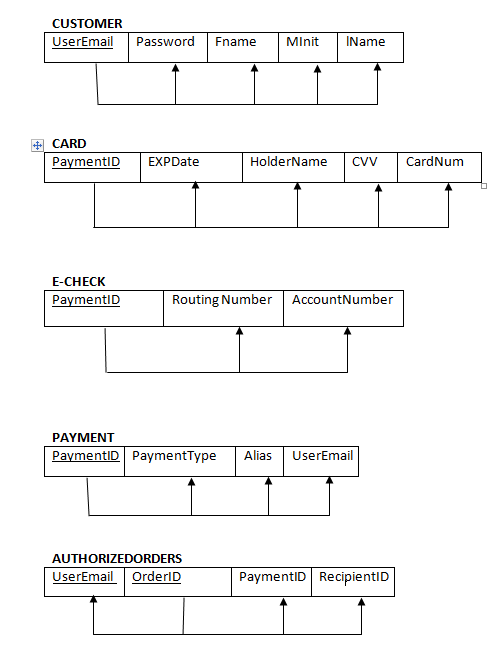


Figure 11: Dependency Diagrams continued

Works Cited:

[1] “PayPal About - Home", Paypal.com, 2017. [Online]. Available: https://www.paypal.com/us/webapps/mpp/about. [Accessed: 09- Jul- 2017].

[2] Lapowsky, “At Long Last, a Universal Shopping Cart for the Web,” Wired, 03-Jun-2017. [Online]. Available: https://www.wired.com/2014/07/keep-onecart-mobile-app/.

[3] "Accept Payments Online And On Mobile | Amazon Pay", Pay.amazon.com, 2017. [Online]. Available: https://pay.amazon.com/us/.