**University of British Columbia, Vancouver**
Department of Computer Science

# CPSC 304 Project Cover Page

Milestone #: 1

Date: July 16, 2022

Group Number: 22

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---|---|---|---|
| Luis Victoria | 78827979 | x1p2b | luisv@student.ubc.ca |
| Richard Chen | 45564895 | n4z2o | richard.chen@ualberta.ca |
| Yan Zhang | 33283136 | l8b3b | yyzhang@student.ubc.ca |

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

### Project Description

What is the domain of the application?

The application, Bitlink, is a Bitcoin payment processor inspired by our group's interest in the fintech area, similar to the likes of PayPal & Stripe. It aims to lower the barrier of entry for merchants to accept cryptocurrency by acting as an intermediary at the point of purchase to allow customers to pay in Bitcoin, while ensuring merchants are paid in fiat currency ($USD). It achieves this by allowing the customer to fund a Bitlink account with Bitcoin and using that to pay for purchases with supported merchants.

What aspects of the domain are modeled by the database?

The database is modeled around the concept of customer and merchant entities, their respective cryptocurrency wallet, or bank accounts, along with the payment processing relationship associated with each order. This includes the type of purchase order, the order's associated items, along with payouts to the merchants and the according fee taken out of the payment as profit for Bitlink.

### Database Specification

- What benefit does the database provide to the application?
  The database tracks the available funds customers have to make purchases, along with the purchases themselves and the amounts to be remitted to merchants. An appropriate database design is essential for a high performing fintech company like Bitlink to keep up with high volume transactions.

- What functionality will the database provide?
  The database will need to be updated with the current balance of the customer's wallet with Bitlink on deposit or withdrawal to the customer's personal wallet. It will also need to be updated on every payment with the details of the purchase while attributing the owed amount to merchants and Bitlink's fee taken as profit.

### Application Platform

- What platform will the final project be on? (PHP/JDBC/etc.)
  Bitlink will be built with a Python backend running Django and a PostgreSQL database.

- What is your application technology stack? (i.e., any other things that
- you're using other than whether you're using PHP or JDBC)
  - Note that for DBMSs, we will only provide support for using the department's installation of Oracle. You are on your own for anything else.
  - You can change/adjust your tech stack later as you learn more about how to get started for the project via latter tutorials.

  The front-end of the stack will be built with ReactJS.

**University of British Columbia, Vancouver**

Department of Computer Science

**ER Diagram**

- An ER diagram for the database that your application will use. It is OK to hand-draw it but if it is illegible or messy or confusing, marks will be taken off. You can use software to draw your diagram (e.g., draw.io, GoogleDraw, Microsoft Visio, Powerpoint, Gliffy, etc.)
    - The result should be a legible PDF or PNG document. Note that your ER diagram must use the conventions from the textbook and the lectures (Do not use crow's feet notation or notation from other textbooks).
- Your E/R diagram should adhere to the expectations listed above.

  (ER Diagram is on the next page)

**Other comments, as appropriate, to explain your project.**

When a customer registers for an account in Bitlink, they will be automatically assigned an account where they will be able to deposit Bitcoin to, this is the ***Internal e-Wallet*** entity. These bitcoins would be held internally by Bitlink, and will give customers the ability to withdraw these bitcoins into their own wallets.

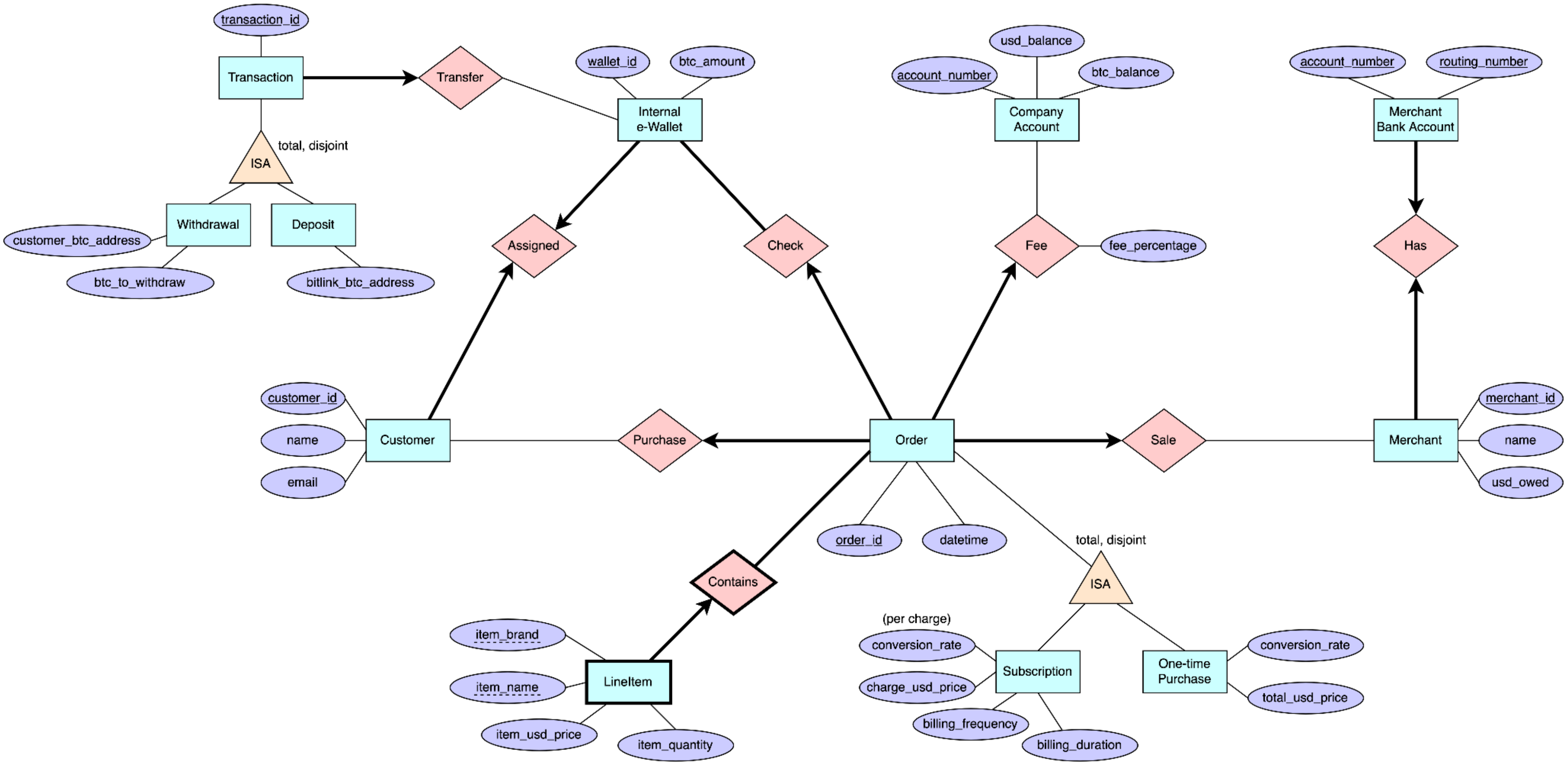
A customer can make an order with a supported merchant (who is also registered with Bitlink) and the price would be listed in USD. An order can be one of two things (as represented by the two subentities), ***Subscription*** or ***One-time purchase***. After placing an order, the total price is converted into BTC using the exchange rate at the moment of the purchase. For instance, a customer places an order for an item at a price of $100; if the exchange rate (represented as *conversion_rate*) was $20,000 = 1 BTC, then the amount to be subtracted from the customer's account is 0.005 BTC. This amount is first checked to see whether or not the customer has enough funds to proceed with the transaction and if so, the transaction is marked as approved, sending a message to the merchant that the transaction has gone through and they can proceed to deliver the product.

In the ***Order*** entity, *conversion_rate* appears to be duplicated across the ***Subscription*** and ***One-time Purchase*** subentities. However, a subscription requires multiple charges throughout a period of time. Due to the volatility of Bitcoin, a charge one month can deduct a different amount of Bitcoin than a charge in a different month. These different conversion rates are stored.

For the purpose of this project, we will assume that we are able to convert BTC to USD instantaneously (i.e. there is no profit/loss being made upon exchanging from one currency to the other). While in a practical sense this would not be the case, we use this assumption in order to focus on the main aspects of the project, the processing of payments, and not get carried away with handling the purchasing and selling of Bitcoin which is not the main focus of this project.

**University of British Columbia, Vancouver**

Department of Computer Science


transaction_id
Transaction
Transfer
ISA
total, disjoint
Withdrawal
Deposit
customer_btc_address
btc_to_withdraw
bitlink_btc_address
wallet_id
btc_amount
Internal e-Wallet
Assigned
Check
usd_balance
account_number
btc_balance
Company Account
Fee
fee_percentage
account_number
routing_number
Merchant Bank Account
Has
customer_id
name
email
Customer
Purchase
Order
Sale
Merchant
merchant_id
name
usd_owed
order_id
datetime
total, disjoint
ISA
Contains
item_brand
item_name
item_usd_price
item_quantity
LineItem
(per charge)
conversion_rate
charge_usd_price
billing_frequency
billing_duration
Subscription
One-time Purchase
conversion_rate
total_usd_price