PROJECT PLAN

Current state:

We have a completed working prototype and Capital One is satisfied with our minimum viable product. Next semester we plan to integrate more components into our event driven system to improve user experience and more align the functionality of our prototype to real life (improved) financial transactions (such as, having different credit card types available, updating what the different credit cards can do, types of accounts that can be created, a UI etc.). Capital One mentors and we have also been discussing using an actual credit card reader and test card containing simulated user information to run our CLI. Furthermore, we plan to implement Stripe API in order to emulate real consumer data in our pipeline.

The command line interface has been built in order to initiate events.

Finalized architecture (for this semester) complete - CapitalOne mentors and we have discussed making even more improvements and minor tweaks.

Pipelines using AWS to book account and process transactions. Producers and consumers defined, event bus defined, event rules defined.

Define Problem to be solved:

Current modern financial transaction fulfillment, and banking requires fulfillment in bulk at the end of each business day. This is a constraint that limits the flow of transactions, and results in bottlenecks and untimely business transaction fulfillments due to the monolithic and highly coupled nature of their architecture.

In order to solve this issue, many banking businesses set out to build an event driven, cloud-based system for processing financial transactions. This paradigm processes significant changes in states as events in order to facilitate efficiency and modularity.

We are tasked with creating a small-scale prototype for the Fall semester utilizing Amazon Web Services with the requirements: 1) Create new user accounts 2) Initiate and fulfill transactions. In addition to this, the storing of financial and user information in secure databases must be considered as well as the connection between event producers and consumers in order to improve the overall processing efficiency for financial transactions.

Business requirements (sem1):

Build an event driven system that allows mainly 2 things:

1) Booking account

Users should be able to choose different cards they want and make an account. Different cards can have different benefits.

2) Process transaction

We want to be able to process transactions initiated by customers using their accounts. Authorization/authentication needs to occur

Constraints:

Follow EDA pattern, using AWS. Capital One is a major consumer of AWS Lambda and we should roughly follow their pattern.

Keep in mind that one event bus may result in a bottleneck, consider having multiple buses in order to prevent system failure.

Time constraints, as undergraduate students we must make sure that we are efficient in order to provide business value during our weekly sprint meetings with Capital One.

Communication is essential with our mentors so they understand where our work is at and how we can allocate our time efficiently so that their requirements are met to the fullest extent.

List of Stake Holders:

1. Capital One – They put a lot of time and effort into mentoring us. EDA is a topic that interests Capital One for their own business prospects, if we are able to create a working prototype that proves to be efficient, they may choose to implement a similar model into their own business structure.
2. VCU – Our group project and efforts reflect the university as a whole. To ensure the relationship between Capital One and VCU, we must put our best foot forwards and fulfill our project requirements to the best of our abilities.