High-Level Requirements

* Creation of a REST API in Python to support standard bank account operations including:
  + Open a new account
  + Retrieve information on all accounts
  + Retrieve information for a specific account
  + Execute a withdrawal from an existing account
  + Execute a deposit to an existing account
  + Close an existing account
* API will integrate with a back-end PostgreSQL database for storage of account, customer, and address details
* Database schema will include the following:
  + Address
    - ID (autoincrementing integer) – primary key
    - Address (text)
    - City (text)
    - State (text)
    - Zip code (text)
  + Customer
    - ID (autoincrementing integer) – primary key
    - First name (text)
    - Last name (text)
    - Address ID (foreign key)
    - Email address (text)
  + Account
    - ID (autoincrementing integer) – primary key
    - Account number (text)
    - Customer ID (foreign key)
    - Current balance (decimal value)
* Conceptual architecture:

Diagram

Description automatically generated

* Error handing:
  + Prevent a withdrawal that would result in an overdraw
  + Require that a minimum of $25.00 be included to open a new account
  + Prevent specification of invalid values on a withdrawal or deposit (positive decimal numbers only)
* You are to use TDD (Test-Driven Development) as the methodology for implementation with a target unit test coverage % of 85% (minimum)
* This application is to be deployed in the Cloud (AWS) using an EC2 instance for hosting the API and RDS for hosting the database
* This application is to leverage AWS CodeBuild, CodeDeploy, and CodePipeline for CI/CD deployments of the infrastructure components supporting the defined application architecture

Technical Requirements

* Python 3.8+
* FastAPI and Pydantic
* PostgreSQL (target Amazon Aurora)
* AWS for IaaS, PaaS, and CI/CD