Commerce Bank Web Application

Architecture/Design Document

**Table of Contents**

[**1**](#_heading=h.gjdgxs) **INTRODUCTION 3**

[**2**](#_heading=h.30j0zll) **DESIGN GOALS 4**

[**3**](#_heading=h.1fob9te) **SYSTEM BEHAVIOR 4**

[**4**](#_heading=h.3znysh7) **LOGICAL VIEW 6**

[**4.1 High-Level Design (Architecture)**](#_heading=h.2et92p0) **6**

[**4.2 Mid-Level Design**](#_heading=h.3dy6vkm) **7**

[**4.3**](#_heading=h.1t3h5sf) **View 10**

**5****PHYSICAL VIEW 11**

**6 USE CASE VIEW 12**

Change History

**Version:** 0.1

**Modifier:** Toni Tan

**Date:** 03/27/2020

**Description of Change:** Create the document, contains introductions

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Version:** 0.2

**Modifier:** Mao Zheng

**Date:** 03/30/2020

**Description of Change:** Added design goals, Physical view.

**Version:** 0.3

**Modifier:** Micheal

**Date:** 04/03/2020

**Description of Change:** Added/modified middle level, details view.

# Introduction

**Architecture and Design**

The purpose of the architecture/design document is to explain how the system works. This document will help future developers know and understand the system on how the system works, what tools and processes were sued to build the software.

The purpose of this document is to describe the architecture and design of the Commerce Bank Web Application in a way that addresses the interests and concerns of all major stakeholders. The software was built with HTML, C#, CSS, MicrosoftSQL and .Net. Users can login to their account through any device with a browser either on a computer or any mobile device. Sort and search to filter out the transaction that users want to check. Set up notifications for bank transaction alerts to keep account in high security.

For this application the major stakeholders are:

* Users and the customer – they want assurances that the architecture will provide for system functionality. They would also want their data being secured.
* Developers – they want an architecture that will minimize complexity and development effort and secure the users sensitive information.
* Project Manager – the project manager is responsible for assigning tasks and coordinating development work. He or she wants an architecture that divides the system into components of roughly equal size and complexity that can be developed simultaneously with minimal dependencies.
* Maintenance Programmers – they want assurance that the system will be easy to keep track and maintain.

The architecture and design for a software system is complex and individual stakeholders often have specialized interests. There is no one diagram or model that can easily express a system’s architecture and design. For this reason, software architecture and design is often presented in terms of multiple views or perspectives [IEEE Std. 1471]. Here the architecture of the Commerce Bank Web Application is described from 4 different perspectives [1995 Krutchen]:

1. Logical View – major components, their attributes and operations. This view also includes relationships between components and their interactions. When doing OO design, class diagrams and sequence diagrams are often used to express the logical view.
2. Process View – the threads of control and processes used to execute the operations identified in the logical view.
3. Development View – how system modules map to development organization.
4. Use Case View – the use case view is used to both motivate and validate design activity. At the start of design the requirements define the functional objectives for the design. Use cases are also used to validate suggested designs. It should be possible to walk through a use case scenario and follow the interaction between high-level components. The components should have all the necessary behavior to conceptually execute a use case.

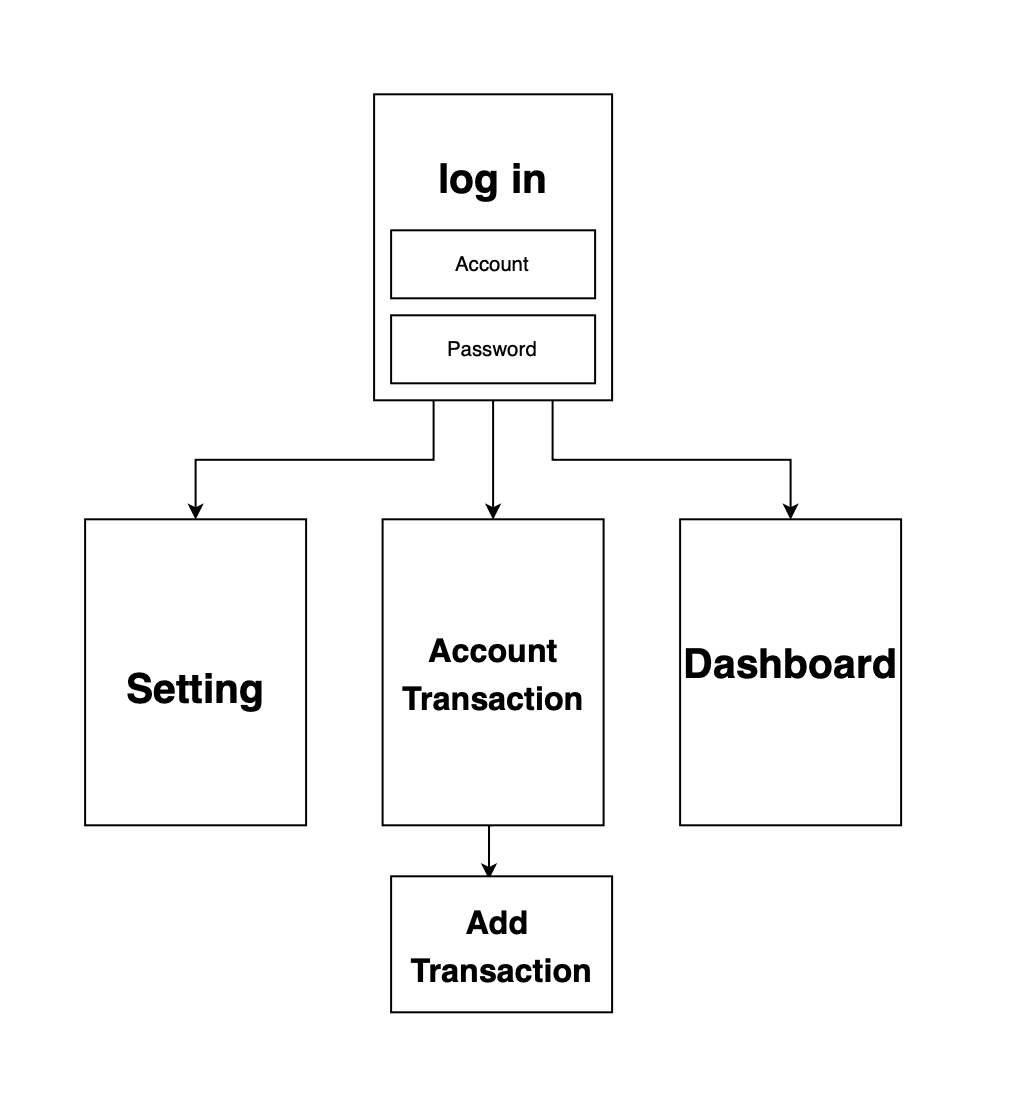
# Design Goals

The design priorities for the Commerce Bank Web Application are:

* The design should minimize complexity which offers clients user friendly pages to use
* The design should minimize development effort which helps the design process to be more efficient bringing down the overall cost. For example, using development tools we are familiar with, having a simplified design plan which will reduce the complexity of the project. Have clear divided tasks for every team member and each task is broken down into small pieces so that each individual can work on the task independently. Also make sure each session doesn’t interfaces with another to increase the working efficiency.
* The design should have a minimalist design feel to it.
* The design should be organized and intuitive.
* The design should be user-friendly, most users should not need a complexed User Manual.
* The design should not be disorganized or cluttered.

# System Behavior

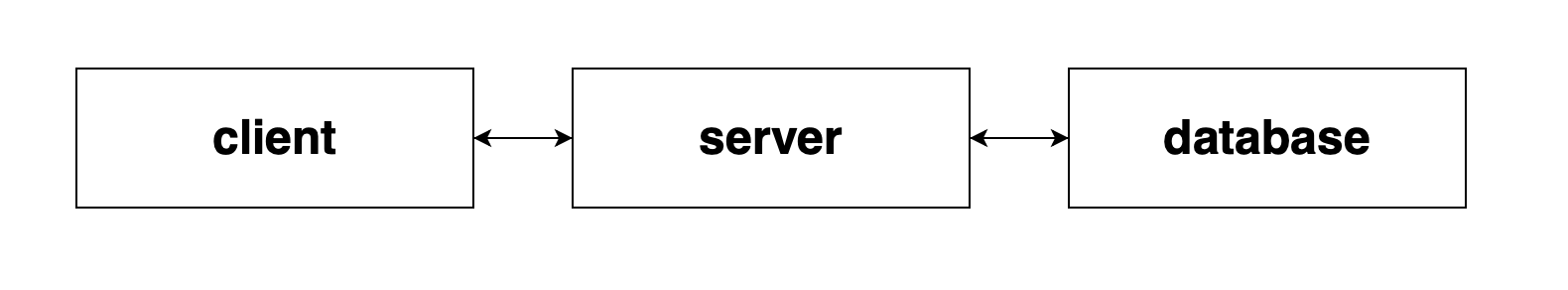
The use case view is used to both drive the design phase and validate the output of the design phase. The architecture description presented here starts with a review of the expected system behavior in order to set the stage for the architecture description that follows. For a more detailed account of software requirements, see the requirements document.



# Logical View

## High-Level Design (Architecture)

The high-level view or architecture consists of 3 major components:



The **client** provides a website interface to users.

* The **server** contains business processes and business transactions.
* The **database** is a central repository for stored user information and transaction data.

## Mid-Level Design

4.2.1 Client

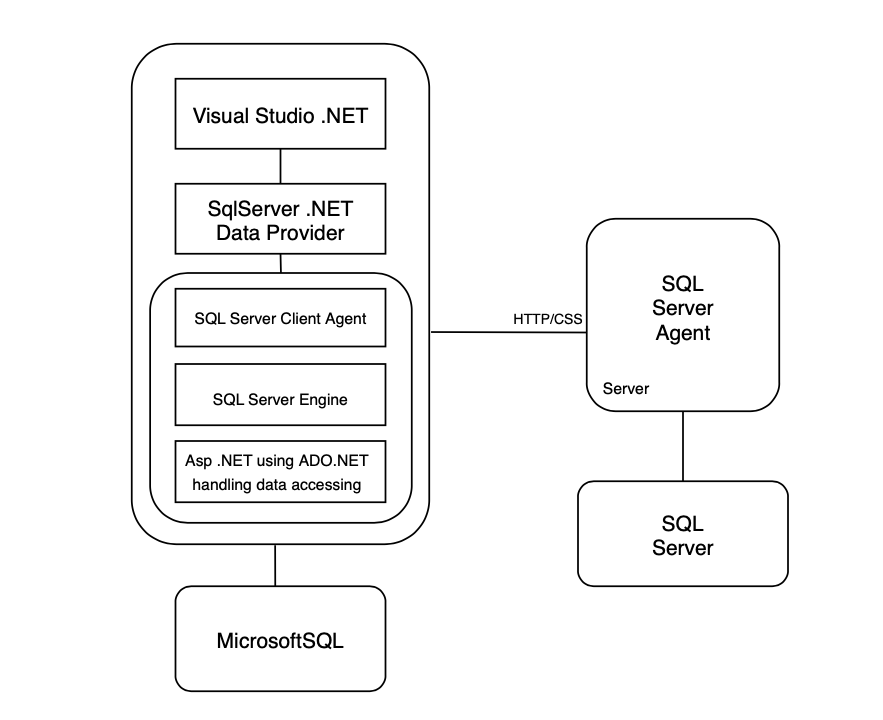
Purpose: Client uses HTML and CSS to display information for the user.

Specification nature: Clients can see the information displayed on HTML and CSS web pages for any information they want to see. They can login their accounts and go to homepage dashboard, will be able to export spread sheet, add/edit/delete/hide/pull/compare notifications, add transactions as well as set up Notification rules.

4.2.2 Server

Purpose: Server uses .NET framework to communicate between the database and client.

Specification nature: ASP.NET connects the data that are stored in our database and connects the information allowing us to display information on our HTML/CSS webpage.ADO.NET is used to handle data access in the database. ASP.NET uses ADO.NET to connect and work with databases.

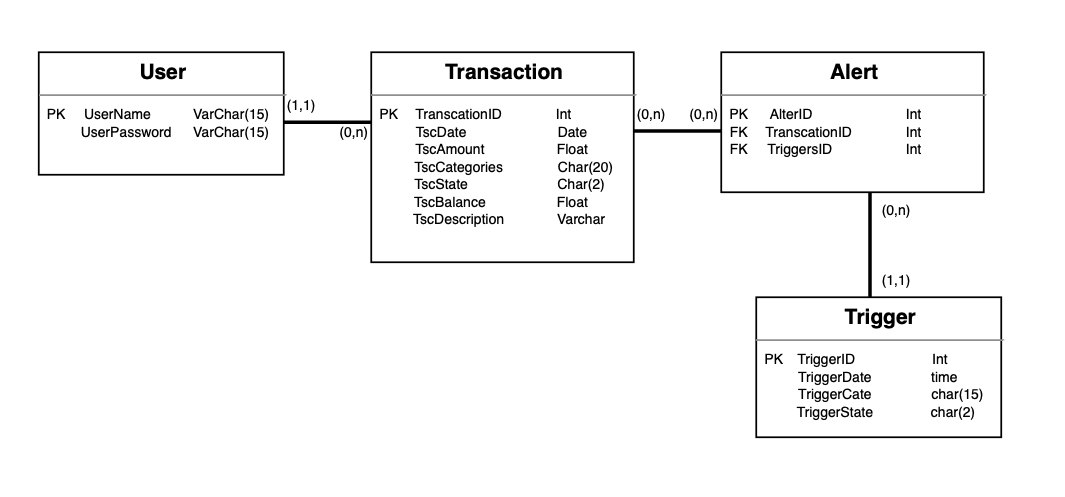


4.2.3 Database

Purpose: Database uses SQL to manage stored data.

Specification nature: The database will be a SQL relational database using ASP.NET framework. The database will store the data in the diagram below as tables and related objects that are created in those diagrams . Also it will handle the data in the database, allowing the system to modify any information.

4.2.4 Database Diagram



## View

# 4.3.1 Client

Purpose: Client uses simple HTML and CSS for formatting information. A view function, or view for short, is simply a C# function that takes a Web request and returns a Web response. Views can also implement specific business logic and context to pass into templates.

Specific Nature:

View Functions

* Index
* Loads all transactions from DB passes to Index template.
* Dashboard
* Loads all notifications
* Settings
* Loads all settings and categories and passes to Settings template.
* Load Alert Transactions Table
* Loads all alerts related to specific passes in rule. Passes that information to the Alert Transactions Table template.
* Add Rule
* Creates rules by state, categories and transaction time range and saves to database. Reloads Active Rules template.
* Remove Rule
* Deletes passed in rule model object from database. Reloads Active Rules template.
* Load Sorted Transactions Table
* Gets dates from request data and sort by user requested. Reload sorted Transactions Table template.
* Mark as Read
* Marks all alerts of passed in dashboard as read. Renders Rule List template.

# 

# Physical View

* .NET
  + We are using .NET for our production environment.
  + ASP.NET connects the data that are stored in our database and connects the information allowing us to display information on our HTML/CSS webpage.
  + ADO.NET is used to handle data access in the database. ASP.NET uses ADO.NET to connect and work with databases. ASP is partially object oriented. ASP.NET is fully object oriented.
* Microsoft SQL Database
  + We are using Microsoft SQL Database.
  + The web page will be connected to an existing SQL Server Compact database. We set the connection to the database via configuration variables in the .NET settings.
  + Microsoft SQL Database is run on our local machine.
  + It manages data for our database where we can pull all of our clients’ information and allows the system to modify any information displayed on the web page as well.

# 

# Use Case View

* View transactions - Show all transaction in database to the clients
* Add/Delete/Modified alerts - clients changing or setting alerts
* View notification - A dashboard shows all alerts
* Update transaction data - Amin manual update transaction in database

