Commerce Bank Web Application:

Architecture/Design Document

Group 6

Table of Contents

Change History

Version: 0.1
Modifier: Lauren Magee
Date: 03/31/2021
Description of Change: Initial rough draft and template, added introduction.

Version: 0.2
Modifier: Cory Free
Date: 4/3/2021
Description of Change: Added design goals, logical view, and process view.

Version: 0.3
Modifier: Txomin Chivite
Date: 4/3/2021
Description of Change: Added logical view parts (high-level design, mid-level design, and detailed class design)

Version: 0.4
Modifier: Mohamud Abdi
Date: 4/3/2021
Description of Change: Added system behavior

Version: 0.5
Modifier: Saurav Pawar
Date: 4/4/2021
Description of Change: Added development view and use case view

---

Version: 1.0
Modifier: Lauren Magee
Date: 4/4/2021
Description of Change: Final edits for first draft

---

1 Introduction

This document describes the architecture and design for the web application being developed for Commerce Bank. This web application is intended to be used as an online transactional ledger. Only pre-registered customers of Commerce Bank will be authorized access to the software's features. Upon initial login, all users will be directed to a home page where a dashboard will showcase recent transactions and other attributable account characteristics. This dashboard will also provide access to the customer's actual ledger, which will spawn in another window. None of the windows require any special training of the users. The intuitive nature of the entire application should enable all users a seamless experience.

The purpose of this document is to introduce the methods and ideas behind the setup and architecture of this web application. The intended readers of this document are the project's stakeholders, all of whom are listed below.

- Commerce Bank – requesting a design that will appeal to their users, is actually usable and adds value to their brand, and is simple to upkeep for future updates and maintenance.
- Project Sponsor – wants the development team to provide quality work for Commerce Bank utilizing methodologies and strategies taught throughout his capstone class.
- Project Manager – responsible for overall completion of the project, focuses primarily on deadlines for submission, work quality, revises developmental details, acts a testing user, provides insight and ideas to meet project requirements, and encourages teamwork flow.
- Developers – goal is to create a system that can be managed and edited by any technically-able individual, while also focusing on delivering efficient and concise work that abides by development standards.

The information featured in this document highlights the system's behavior, logical view (high-level, mid-level, and detailed class diagram), process view, development view, and use-case view. Any stakeholder in need of specific system architecture information can refer to the sections referenced for further guidance.

1. System Behavior – demonstrates the window flow of the system and the intent behind the design to fulfill stakeholder requirements.
2. Logical View - details the general overview of how the system is augmented together, contains three subsections that elaborate the level of design with more and more detail through various diagrams.
3. Process View – shows the function flow of the Logical View's diagrams in a more detailed illustration.
4. Development View – describes the basic setup of the window within the application, what platforms and languages are used to code them, and shares how they're connected on the backend.

5. Use Case View – provides a list of more detailed functions from the application in an algorithmic form, for example, this user action will trigger this backend response from the system.

2 Design Goals

The value of our design will be determined based on whether or not it meets certain restraints set by the project stakeholders or developers. After reviewing the requirements given by these individuals, we've determined the following to take priority in our implementation of their desired system. Our definition of a high quality design will be accomplished if the following design goals are met (1-highest design priority, 2 - second highest design priority, 3 - etc.):

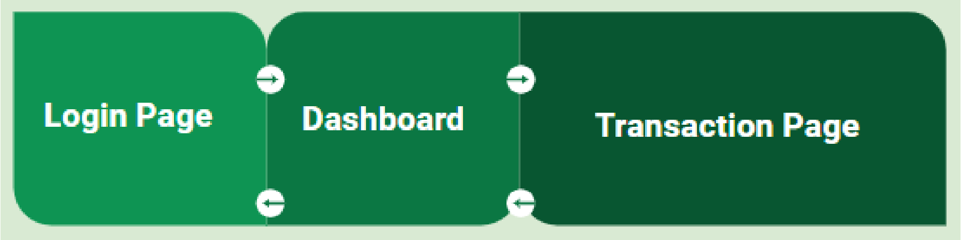
1. **Simplistic and Intuitive -** the design will avoid any false/ hidden affordances to allow for a user-friendly experience.
2. **Brand Recognition and Trustworthiness** - the design will accurately represent Commerce Bank through their colors and logos to establish recognition with their consumers.
3. **Reliability** - the design will be created using a methodology that will constantly be accessible to users unless otherwise stated by site administrators.
4. **Consistency** -the design will be cohesive from window to window in theme and through usability methods.
5. **User-Friendly** - the design should not require any special training or preparation from the user to utilize its features.

3 System Behavior

The web application consists of three main pages that enable users to carry out tasks relating to their account at Commerce Bank. The pages are the login page, the dashboard and the transaction page.

- The login page is the first page that users will see when they launch the web application. Users must enter their account information before they get access to their account.
- The dashboard is the first page that users see once they sign into their accounts. There are a couple tasks that users can carry out in the dashboard. First, users have the option to access their transaction history. Next, users can get information about their spending habits. Finally, users can access notifications related to their Commerce Bank account.
- The transaction page allows users to see their transaction history. Transaction history is data collected from consumers when they make monetary purchases.

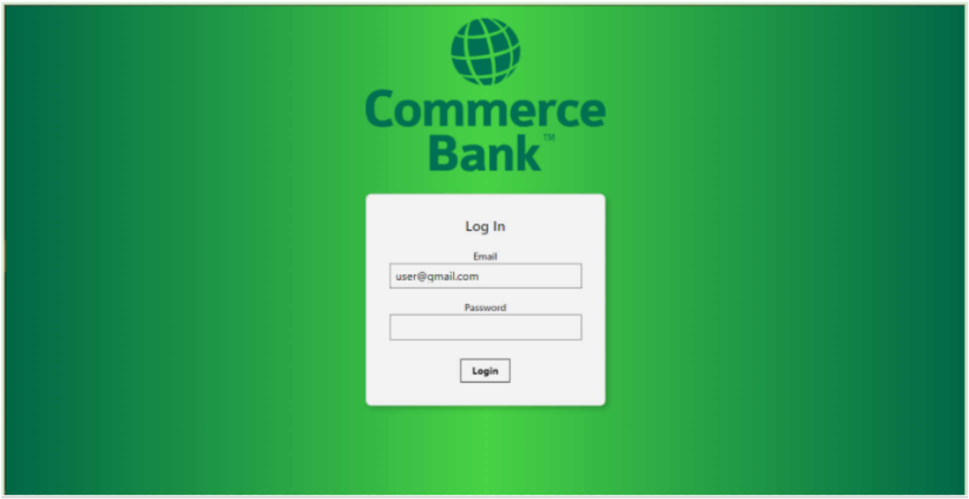
The below pictures illustrate the flow of the windows mentioned.



Login Page

Dashboard

Transaction Page

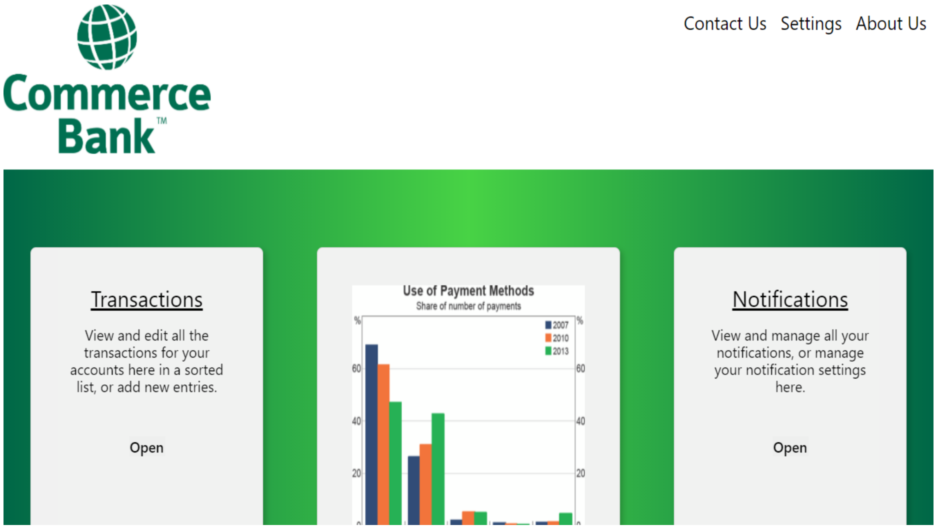


Log In

Email

user@qmail.com

Password

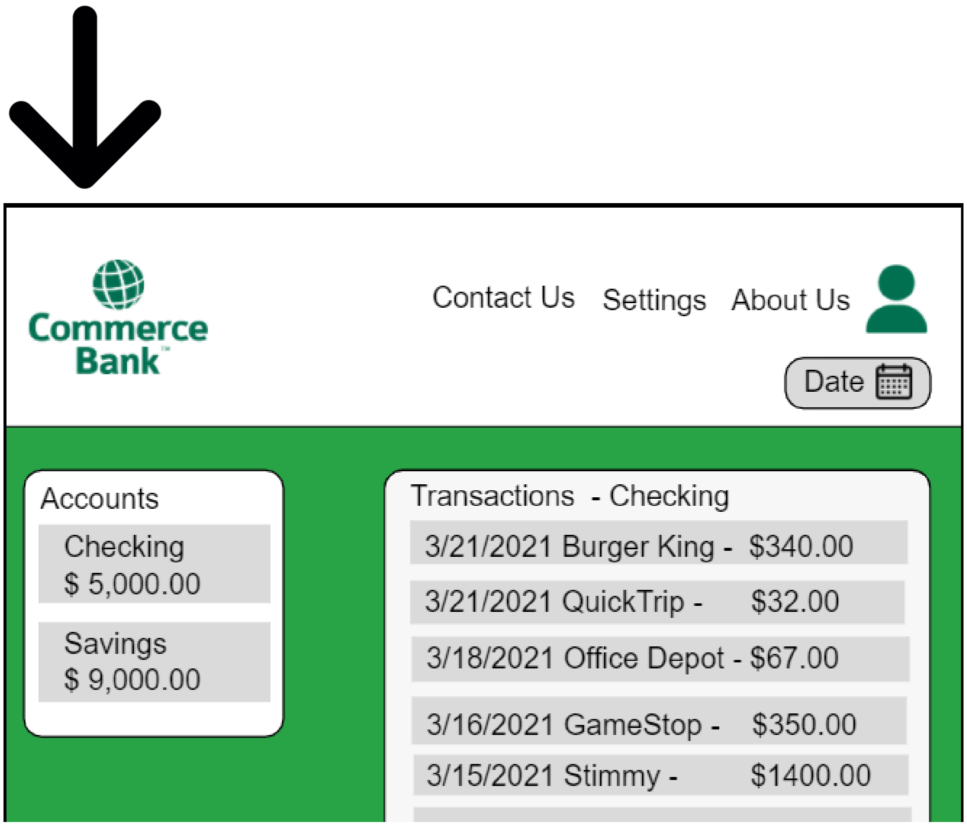


Transactions

View and edit all the transactions for your accounts here in a sorted list, or add new entries.

Notifications

View and manage all your notifications, or manage your notification settings here.



Accounts

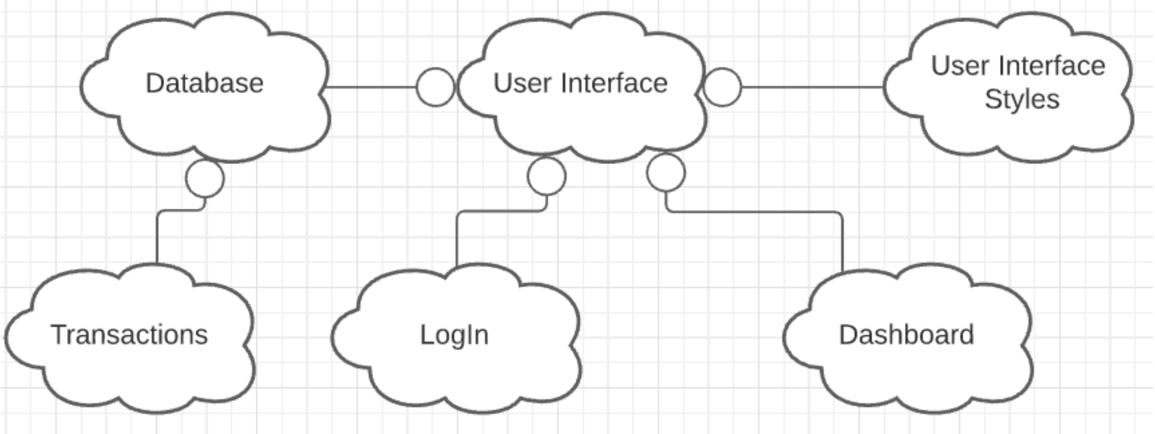
Checking
$ 5,000.00

Savings
$ 9,000.00

Transactions - Checking

3/21/2021 Burger King - $340.00
3/21/2021 QuickTrip - $32.00
3/18/2021 Office Depot - $67.00
3/16/2021 GameStop - $350.00
3/15/2021 Stimmy - $1400.00

4 Logical View



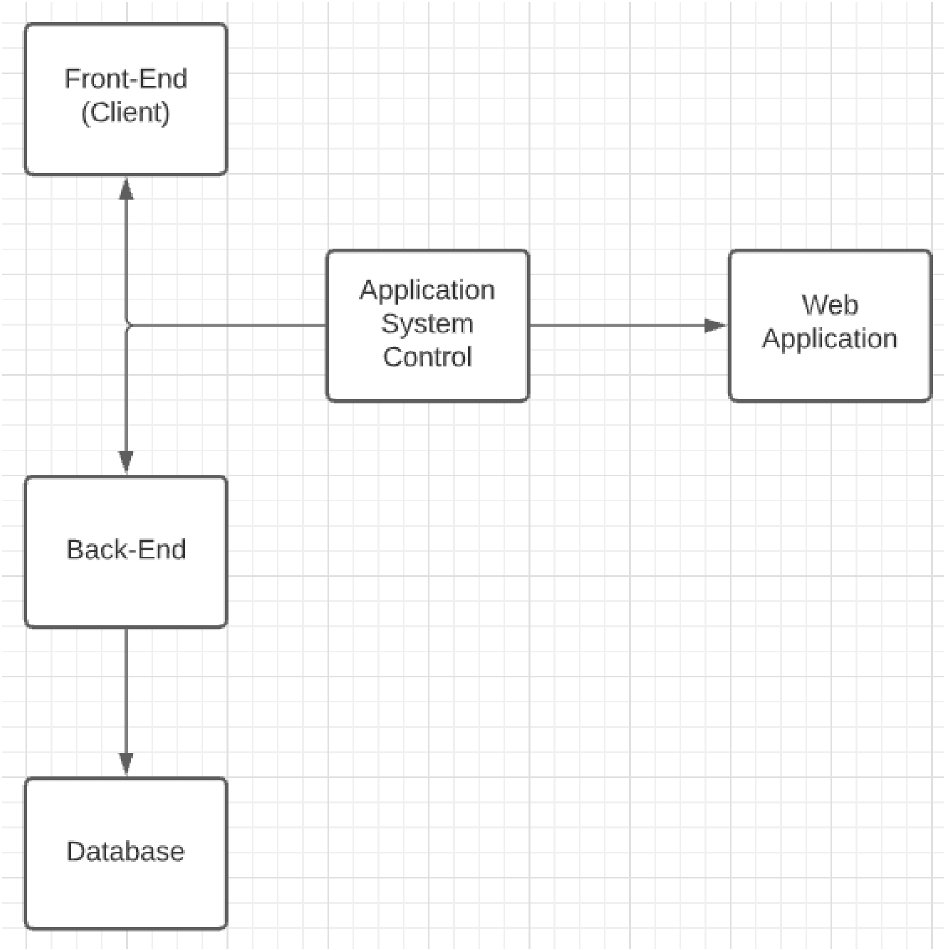
The logical view of our application showcases the main parts of our program, this includes design styles, interfaces, windows, and the backend. In general, it provides a high level overview of what our application provides for its users.

This diagram previews the five following main segments:

User Interface: what the customers/users will use to navigate between the various windows/functionalities of the application.
User Interface Styles: responsible for making the UI more viewer friendly while remaining consistent with the Commerce Bank theme.
Transactions: shows the information obtained through the database and allows the user to scroll through all the transactions in a sorted list.
Login: allows users to access the application assuming they already have a registered Commerce Bank account.
Dashboard: allows the user to navigate to any part of the application and acts as the homepage. It will also provide a graphic to show a quick synopsis of recent transactions.

4.1 High-Level Design (Architecture)

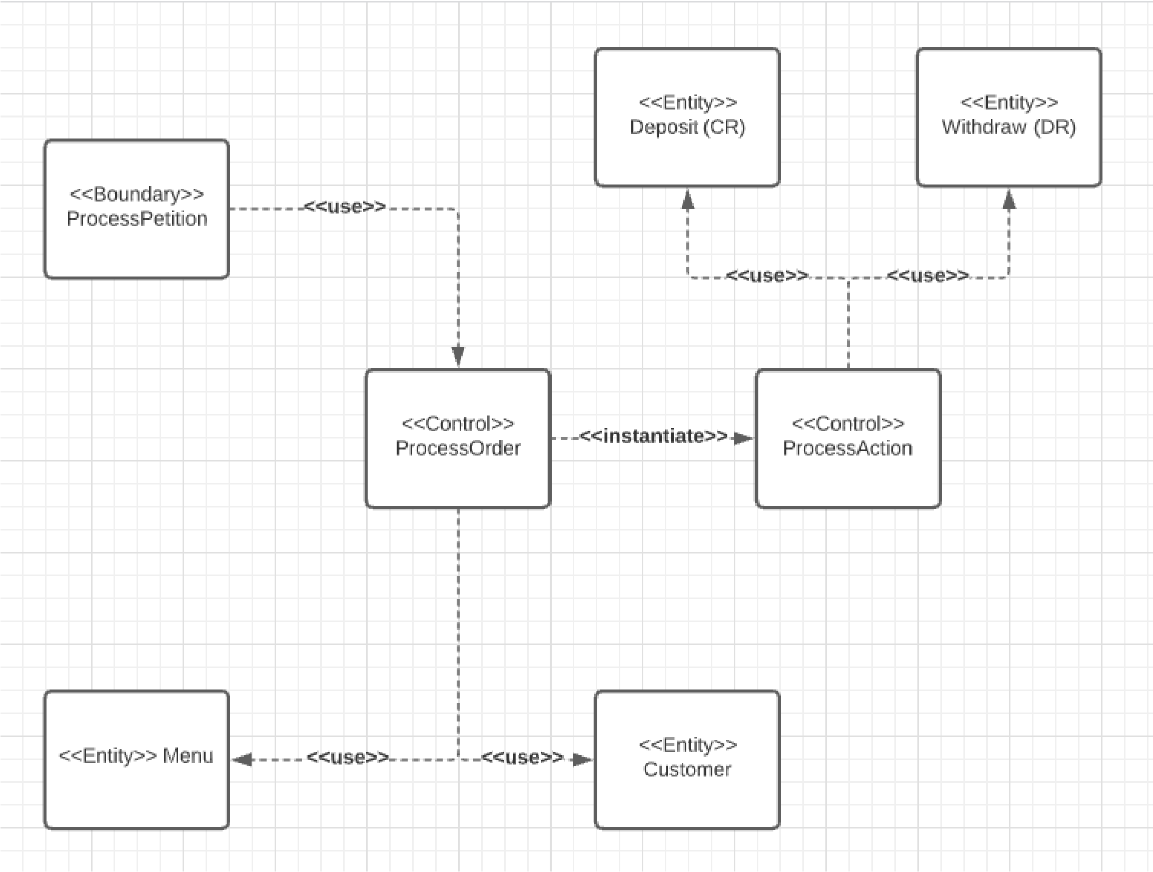
The high-level view or architecture consists of 4 major components:



The high-level view or architecture is composed by three main factors:

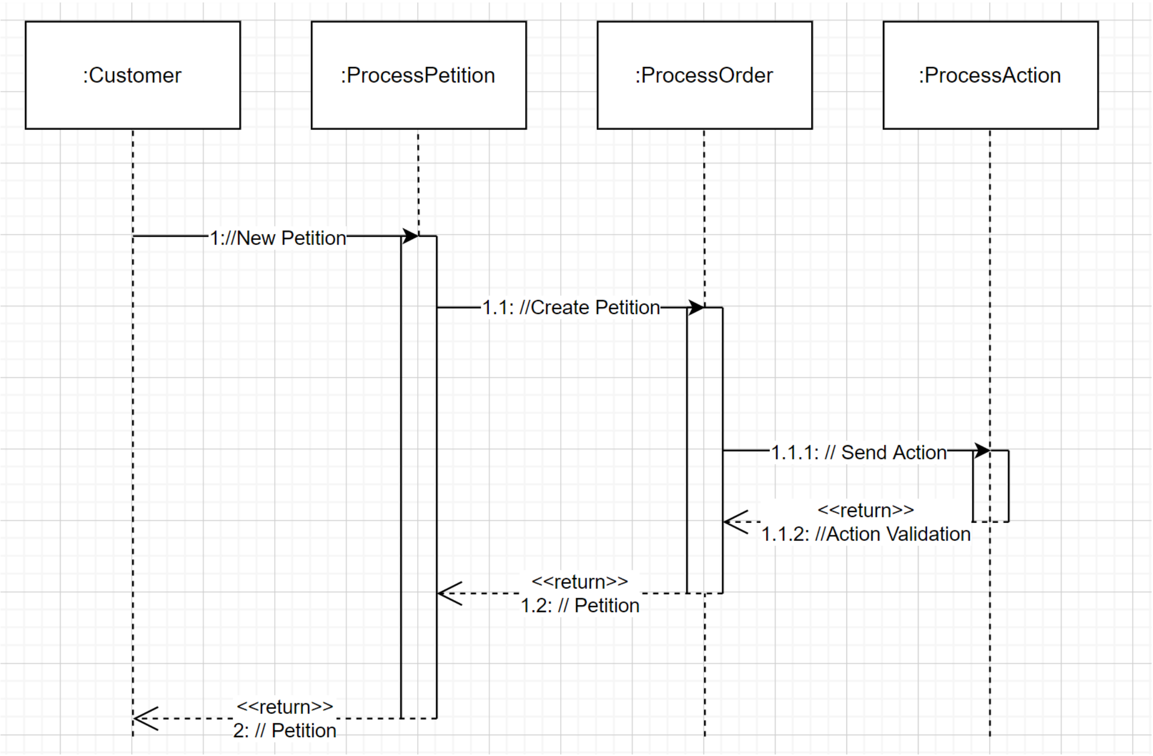
- **Web Application:** Obtains a customer's petition coming from the front-end user, processes their request, and sends their petition to the back-end (Database) which will give them back their desired requested result back to the users.
- **Front-end:** The user is able to navigate through the banking application. The customer will be able to access data based on their search criteria from the database.
- **Back-end**
    - **Database Server:** Incorporates the data which is stored for users interaction. It accommodates data access requests coming from the web application. Additionally, it processes customers requests via queries and returns the results back to the web application.

4.2 Mid-Level Design

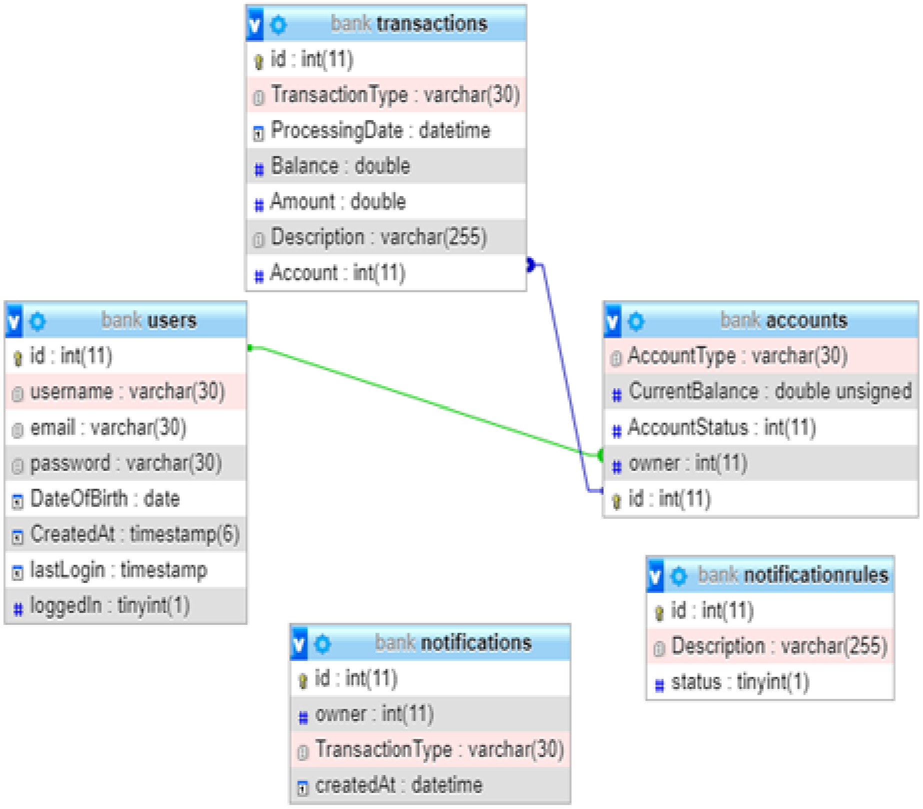


Mid-Level design parts and their relationships

The below figure illustrates the elementary view of the system’s Mid-Level Design segments.

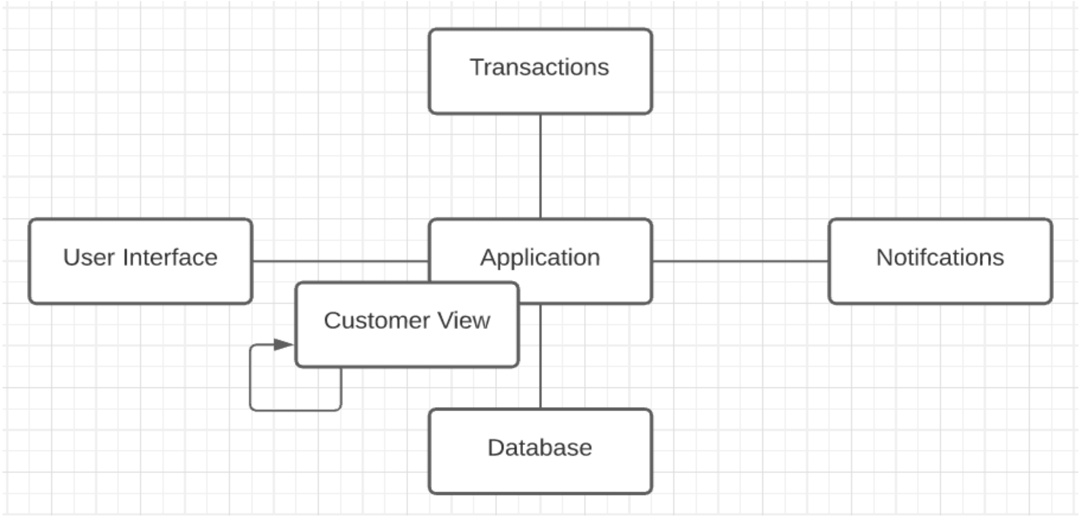

:Customer
:ProcessPetition
:ProcessOrder
:ProcessAction
1://New Petition
1.1: //Create Petition
1.1.1: // Send Action
<<return>>
1.1.2: //Action Validation
<<return>>
1.2: // Petition
<<return>>
2: // Petition

4.3 Detailed Class Design



5 Process View

Each edge shows the connections of the application and its components.



6 Development View

A web-application's characteristics consist of the ability to read, update, and delete data with the program. The frontend of this web-application will utilize HTML5, CSS3 (tachyons), and JavaScript. HTML5 will instruct the browser how to display on-screen content while CSS keeps the user displayed information in correct format. JavaScript will be implemented to aid with functionality on the backend and will assist with login, transaction, and notification operations.

More specifically, the frontend features a bold color gradient which aligns with the stakeholders' company branding. It also has traditional login page features including a login box and a centered logo. If given the correct credentials, the backend will navigate the user to the next window, the dashboard/ home page. The dashboard has three main components showcased to the user: recents transactions, notification settings, and a visual representation of their recent activity.Here the user will be able to manage their online ledger and reset different notification rules to customize their banking experience.

The server side will use MySQL and .NET framework for implementation. The back-end will run on MySQL, which will store the user's transactions and output the data when called from the web application. The deployment tests will include quality assurance, usability, and accessibility. Other tests may include HTML/CSS validation, cross-browser and cross-device tests.

7 Physical View

[TBD]

8 Use Case View - Saurav

The web application should be dynamic across all devices and browsers. Every component in the application should render quickly for the users. The use cases detailed below will be implemented and executed from the program.

(1) Users may login to their account with the right credentials. Unrecognized username or password will give the user a prompt or error.
(2) Given the correct credentials, the login form should take the user to the dashboard.
(3) The branding (Logo) on the pages should be unresponsive with user selection.
(4) The 'About Us' and the 'Settings' on the dashboard should be functional and should lead to the footer of the homepage or another page that its affordance is set to.
(5) Users will be able to preview a summarized snippet of their account in one of the three components on the dashboard.
(6) Users can manage their Transaction and Notifications settings from the other two components of the dashboard. When selected it should lead them to their respective destinations.
(7) The user has the option to turn off/ on different notification rules from their dashboard.
(8) In the transaction window, the user will be able to view their transaction history and add additional entries.
(9) After adding entries to their ledger, the user's edits will be saved to the system and updated in the SQL backend.
(10) The user can then return to their dashboard, without logging out of their account, or proceed to log out of their account in that window.
(11) Once a user is logged out of their account they should not be able to go back into the application without logging in again with correct credentials.
(12) If the user has turned on notifications for the application, their updates should give them a prompt notification with the associated change.