Author: Zachary Thomas

[Email address]

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

finance application

Table of Contents

# Introduction

The Church Finance App is a windows application designed to store financial data, generate reports, reconcile data, and provide traceability between financial transactions. The application serves as an interface between the user and a SQLite database that holds the financial data. The application also gives flexibility to the user to create and manage budgets, set reminders for reoccurring bills.

# Architecture

Church Finance App is a windows application written in Java using Swing for the GUI development. It encapsulates a model, view, and controller (MVC) architecture. The database is a SQL. All information is read from and written to these tables of the database. This is the model. The viewers are the dialogs that the users interface with. These are discussed in detail in the design section. The controller is a software component of the application that acts as a broker. The broker knows how to process users requests input through the viewers to retrieve or send that information to the database. The architecture can be seen in figure 1 below.

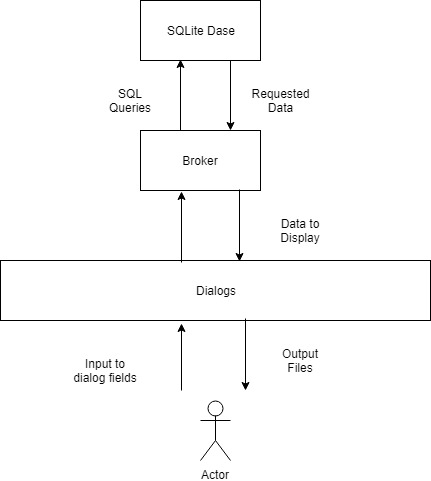


Figure - Software Architecture

## Database

An SQL database is used maintain all of the data. Using an SQL database allows for the application to have scalability to be used locally or accessed remotely depending on the location of the database. The database will contain 5 tables with a fixed number of columns. The imbedded sql tables excel sheet shows the name, description, columns, and data types for each table of the database.



## Broker

The Broker is a software component that is the interface between the GUI and the database. It knows the architecture of the database and how to send SQL queries. This approach keeps the design easily maintainable should another database be used in the future. Figure shows a diagram of the broker interactions.

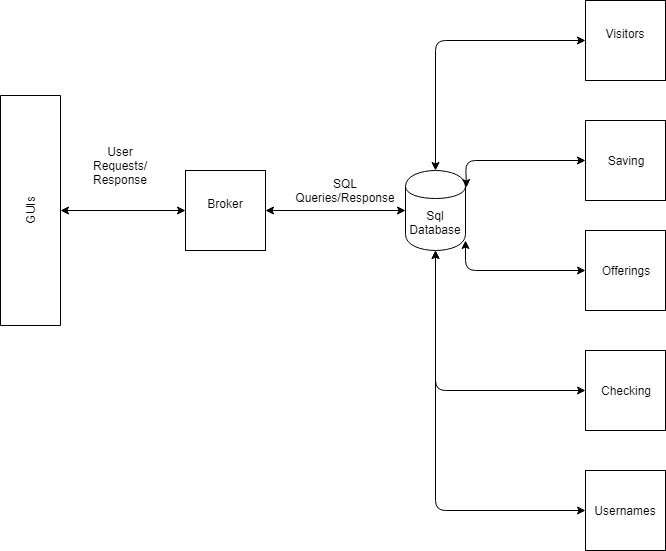


Figure - Broker Interaction

## User Accounts

The application will have user accounts. These are not to be confused with the user accounts associated with the database. (The database only has one user, the broker.) There are three types of users: counters, trustees, and administrators. The difference between the users are the roles assigned to them. Table 1 shows the roles of each user.

Table - User Roles

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Counter** | **Trustee** | **Administrator** |
| **Log offerings** | X | X | X |
| **View ledgers** | X | X | X |
| **Log purchases** |  | X | X |
| **Perform Transfers** |  | X | X |
| **Reconcile Account** |  | X | X |
| **Upload Documents** | X | X | X |
| **Generate Reports** |  | X | X |
| **Generate Contribution Statements** |  | X | X |
| **Update visitor Info** | X | X | X |
| **Manage Budget** |  | X | X |
| **Configure Options** | X | X | X |
| **Manage Users** |  |  | X |

The users will have access to the dialogs according to the roles that they have. The users th.en will enter data into the GUIs that result in a quiery to the database sent by the broker.

# Design

## Login Dialog

## Menu Dialog

## Offerings Dialog

## Purchase Dialog

## Transfers Dialog

## Ledger View Dialog

## Report Creation Dialog

## Upload Document Dialog

## Contribution Statement Dialog

## Visitor and Members Dialog

## Reconcile Dialog

## Options Dialog

## Users Dialog

## Budget Dialog

## Dialogs