

My GitHub Project

C++ & wxWidgets

Marvus in C++

Contents

Glossary	1
Acronyms	2
1 Introduction	3
1.1 Technologies Used	3
1.1.1 wxWidgets	3
1.1.2 SQLite	3
1.1.3 ConsoleLib	3
1.2 Committing a git commit	4
2 Core	5
2.1 UML	5
2.2 Controller	5
3 GUI	6
3.1 Foreword	6
3.2 Window	6
4 Database	7
4.1 Foreword	7
4.2 ERD Diagram	7
Lists	9
Listings	9
Figures	9
List of tables	9

Glossary

wxWidgets A cross-platform GUI toolkit for C++. 1, 3

Acronyms

GUI Graphical User Interface.

1 Introduction

This application originally began as a purely CLI-based project (which is also the reason for its name, *ConsoleArt*). As the project evolved, a Graphical User Interface (GUI) was later added. Development started at a time when C++17 was the newest standard, so the use of modern features such as smart pointers was adopted gradually throughout the project.

1.1 Technologies Used

1.1.1 wxWidgets

This project is written mainly in C++ and utilize wxWidgets for the GUI. wxWidgets is cross-platform and utilize naive GUI libraries for each supported platform.

1.1.2 SQLite

1.1.3 ConsoleLib

ConsoleLib is my custom library developed to simplify the creation of CLI applications. It provides an abstract `IConsole` interface and several platform-specific implementations:

- **DefaultConsole** – A basic implementation using standard output (`std::cout`) without any additional formatting features.
- **UnixConsole** – Adds support for text coloring and formatting through ANSI escape codes commonly available on Unix-like systems.
- **WindowsConsole** – Inherits from `UnixConsole` and enables UTF-8 output on Windows while ensuring compatibility with ANSI escape codes.

The library also includes additional utility modules such as an argument parser and other helpers commonly required in CLI tools, making it reusable across multiple projects.

1.2 Committing a git commit

Commit type cheat table	
Type	Repository change
Feat:	A new feature is added
Fixed:	A bug is fixed
Docs:	A documentation is updated
Refactor:	A code change that is not affecting functionality
Test:	A code change in unit tests
Chore:	A repository maintenance action
Version x.y	When new version is released for clearer commit history

Table 1.1: Types of commit headers

2 Core

2.1 UML

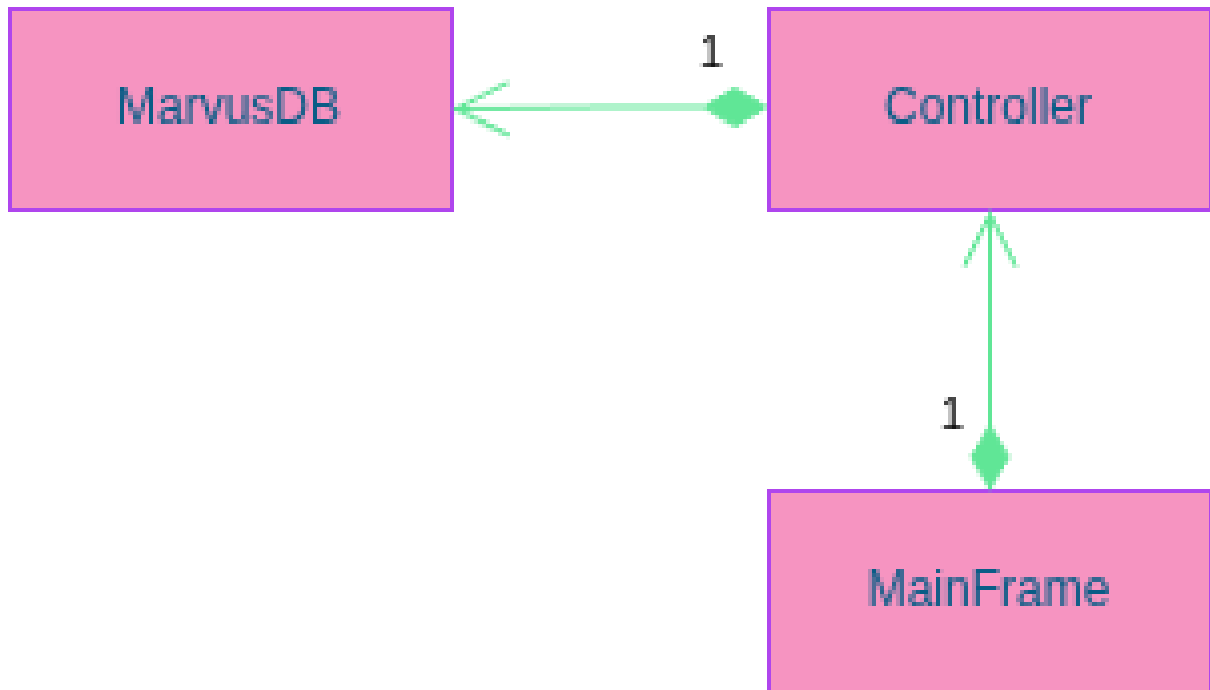


Figure 2.1: Controller relations UML

2.2 Controller

The Controller class is core of this application.

3 GUI

3.1 Foreword

3.2 Window

4 Database

4.1 Foreword

Even though SQLite has a wxWidgets wrapper I didn't use it as I want my application logic to be independent from the GUI framework.

4.2 ERD Diagram

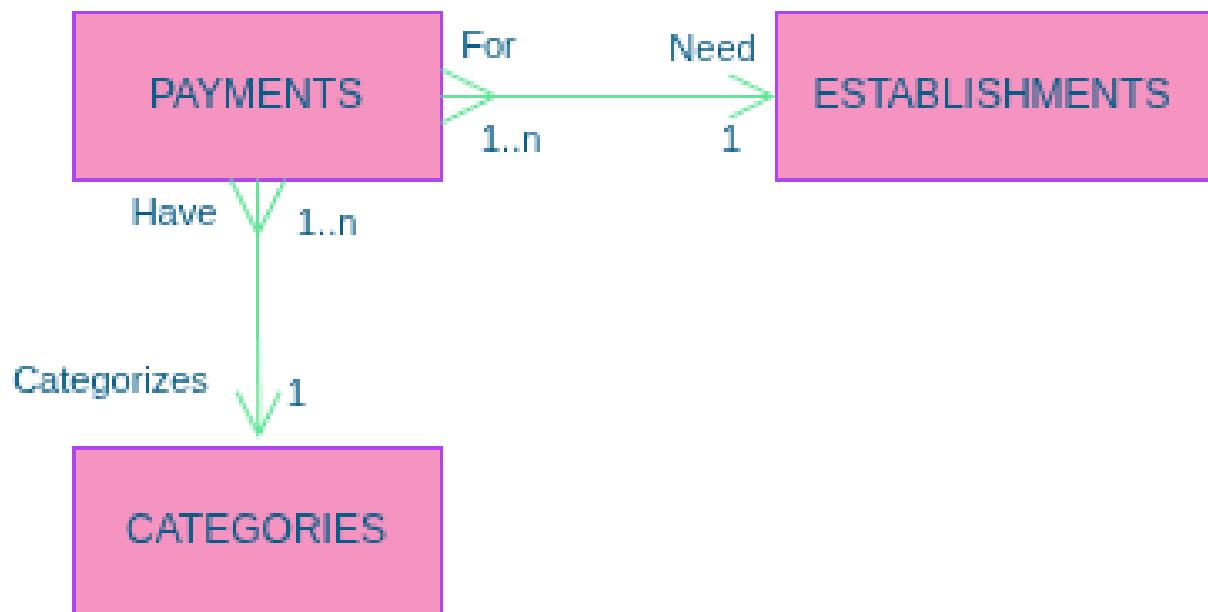


Figure 4.1: Database ERD

Each **PAYMENT** has one **CATEGORY** for simplification and this will give us rough statistics per **CATEGORY** as the purpose of this database/application is to track income and spending's and not fully accurate accountant statistics.

Summary

Listings

List of Figures

2.1	Controller relations UML	5
4.1	Database ERD	7

List of Tables

1.1	Types of commit headers	4
-----	-----------------------------------	---