Software Requirements Specification

for

Oceanus ERP

Version 0.1.0 approved

Prepared by CMIV (Chuck)

Oceanus Digital Consulting

March 24, 2023

The Oceanus ERP maintainer(s).

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Oceanus ERP	3-24-23	Project initialization	0.1.0

1. Introduction

1.1 Purpose

This document goes over the software requirement specifications (SRS) for the Oceanus enterprise resource planner (ERP) that allows users to accept Monero cryptocurrency payments and manage their day-to-day activities as small business owners.

1.2 Document Conventions

The following terms/conventions are used:

- *ERP: A type of software that organizations and businesses use to manage their day-to-day activities such as payments, project management, services offered, etc.*
- *API: An intermediary layer or middleman that processes data between systems – like a user interface and a database.*
- *Database: An organized collection of data stored and accessed electronically, written in SQL (Structured Query Language).*
- *Rust: A relatively new programming language that began as a side project in 2006 by Graydon Hoare, a Mozilla employee. Rust is a systems programming language that adopts the approach of "zero cost abstraction". The main selling point is that the language maintains memory safety while not sacrificing speed or control.*
- *Monero (XMR): Private, decentralized cryptocurrency that specializes in the use-case of money. Monero is an electronic currency to be exchanged for goods and services that avoids the pitfalls of other conventional money – such as inflation, usury, etc.*
- *CRUD: The functions necessary for a persistent storage application: Create, Read, Update, and Delete*

1.3 Intended Audience and Reading Suggestions

This document is written for small business owners and Monero users who want to automate their daily business practices whilst dipping their toe into the world of cryptocurrency. Business owners who normally make invoices and send them to clients would benefit from reading this document in order to see how to integrate Monero payments into that workflow. The following section will discuss system features and requirements. Section 2 will go over the steps that will be taken to implement this ERP system. Section 3 will show off some of the terminal interface features that the user would be interacting with.

1.4 Product Scope

My product is a terminal application that allows its users to keep track of and manage day-to-day business activities – along with accepting Monero payments by sending the receiving address to their clients/customers. The program will watch for incoming Monero payments and process them as they arrive – only removing pending status after a certain number of confirmations.

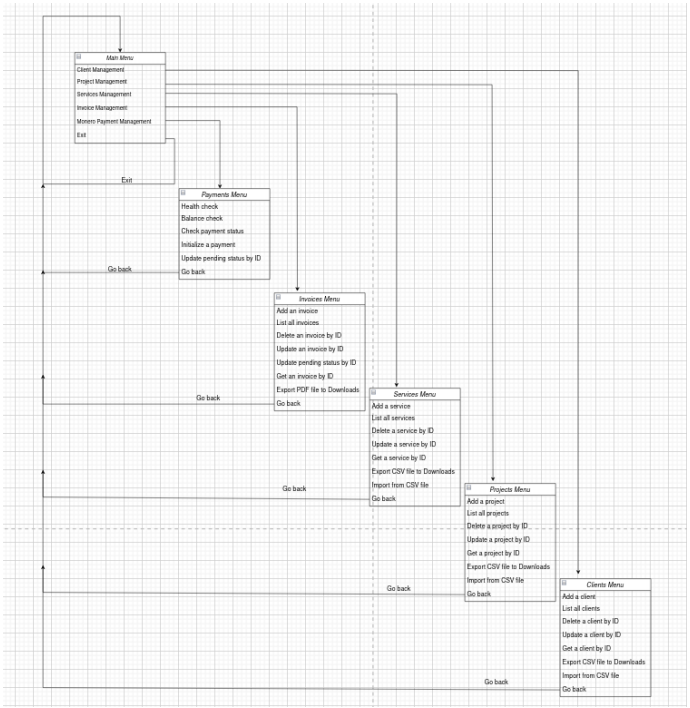
1.5 References

- https://www.getmonero.org/
- https://www.researchgate.net/publication/361274399_Rust_The_Programming_Language_for_Safety_and_Performance
- https://github.com/busyboredom/acceptxmr
- https://github.com/moneropay/moneropay

2. Overall Description

2.1 Product Perspective

Oceanus ERP is a new product that will aid its users in business management and accepting Monero payments. The software is terminal-based and can be installed on any platform that a Rust program can compile for.



2.2 Product Functions

- *Monero payments: Use either AcceptXMR or Moneropay API in order to automate the acceptance of Monero payments for small businesses.*
- *Client management: Full CRUD functionality for handling business clients.*
- *Project management: This includes full CRUD functionality for handling projects.*
- *Service management: CRUD functionality for keeping track of services offered to clients.*
- *Invoice management: CRUD functionality for the handling of invoices to be given to clients in PDF form after a payment is initialized.*

2.3 User Classes and Characteristics

There is only one user within this system. The user executes the program and from there take advantage of full functionality. The user arrives at the Main Menu with an ASCII splash screen with a predetermined number of options to choose. This means that panics/crashes resulting from user input cannot occur.

2.4 Operating Environment

This application will work in any environment where a Rust application can be compiled, no other special software is necessary here.

2.5 Design and Implementation Constraints

The implementation and design constraints would mostly reflect the timeline necessary to get Oceanus ERP's functions up and running. The implementation of the Rust API that communicates with the Monero payment processor(s) would take up most of that timeline.

2.6 User Documentation

Basic computer skills and reading comprehension are required to run/operate this program. Know how to use a terminal and know ahead of time what kind of business you would like to operate.

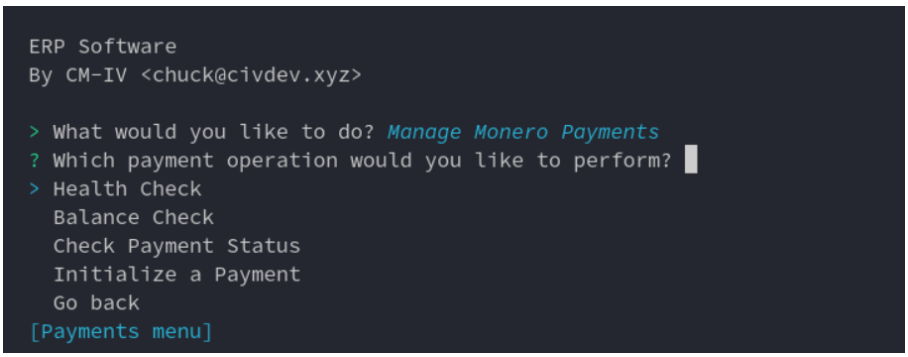
2.7 Assumptions and Dependencies

- *If the user does not download the pre-compiled binary for their computer system, then they would need a Rust compiler (ideally v1.60.0 or later) present in order to compile the program themselves.*
- *The user is assumed to have a self-hosted instance of the Moneropay API running locally.*
- *A Sqlite file database is used in this program with SQL queries via the SQLX crate.*

3. External Interface Requirements

3.1 User Interfaces

All interaction with the Oceanus ERP application is done through a terminal on a computer. Pictured below is what the Main Menu screen and options would look like.



3.2 Hardware Interfaces

There is currently no hardware interface with this terminal software.

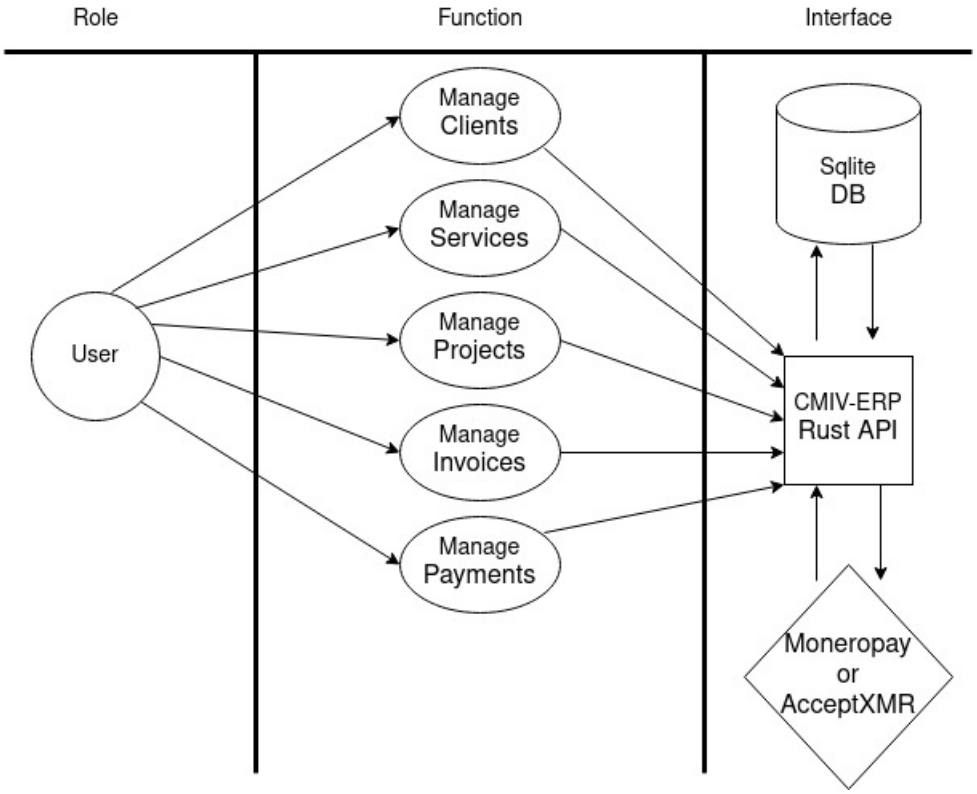
3.3 Software Interfaces

This software is meant to run in the terminal and it will make HTTP requests to the Moneropay API. An internet connection is required to use Monero payment functionality (when not using a local Monero node via the wallet RPC).

3.4 Communications Interfaces

The Oceanus ERP Rust API will take advantage of POST/GET HTTP requests in order to communicate with Moneropay API endpoints. Otherwise, the Oceanus ERP Rust API will communicate with a local Sqlite database file that is created upon execution of the program and placed in the user's "config" directory.

4. System Features



4.1 Monero Payment Management

4.1.1 Description and Priority

This functionality is the crux of the issue, where Monero payments are adopted within software that helps small businesses automate and deal with their daily issues.

4.1.2 Stimulus/Response Sequences

Once the user has chosen to Manage Monero Payments via the Main Menu, they will have the ability to perform various operations - such as health checks, balance checks, payment status, etc. This functionality takes a high priority within The Oceanus ERP application.

4.1.3 Functional Requirements

REQ-1: Payment address and description info is stored within the database.
REQ-2: Database stores and uses correct information.

4.2 Client Management

4.2.1 Description and Priority

This functionality is designed to help the small business owner keep track of their clients, it has a medium priority.

4.2.2 Stimulus/Response Sequences

Once the user has chosen to Manage Clients via the Main Menu, they will have the ability to perform various CRUD operations - such as add a client, list clients, export clients to CSV file, etc. This functionality takes a medium priority within The Oceanus ERP application.

4.2.3 Functional Requirements

REQ-1: Client information is required
REQ-2: CRUD HTTP requests are made by the Rust API

4.3 Project Management

4.3.1 Description and Priority

This functionality is designed to help the small business owner keep track of their projects that they assign to clients, it has a medium priority.

4.3.2 Stimulus/Response Sequences

Once the user has chosen to Manage Projects via the Main Menu, they will have the ability to perform various CRUD operations - such as add a project, list projects, export projects to CSV file, etc. This functionality takes a medium priority within The Oceanus ERP application.

4.3.3 Functional Requirements

REQ-1: Client and Project information is required
REQ-2: CRUD HTTP requests are made by the Rust API

4.4 Service Management

4.4.1 Description and Priority

This functionality is designed to help the small business owner keep track of their services that they offer to clients - including the dollar value that is charged per service, it has a medium priority.

4.4.2 Stimulus/Response Sequences

Once the user has chosen to Manage Services via the Main Menu, they will have the ability to perform various CRUD operations - such as add a service, list services, export services to CSV file, etc. This functionality takes a medium priority within The Oceanus ERP application.

4.4.3 Functional Requirements

REQ-1: Service information is required
REQ-2: CRUD HTTP requests are made by the Rust API

4.5 Invoice Management

4.5.1 Description and Priority

This functionality is designed to help the small business owner bill their clients for services rendered. It is to operate seamlessly with Monero payments such that a Monero subaddress QR code is included after a payment is started, it has a high priority.

4.5.2 Stimulus/Response Sequences

Once the user has chosen to Manage Invoices via the Main Menu, they will have the ability to perform various CRUD operations - such as add an invoice, list all invoices, export invoice to PDF, etc. This functionality takes a high priority within The Oceanus ERP application.

4.5.3 Functional Requirements

REQ-1: Client, Project, and Service information is required
REQ-2: CRUD HTTP requests are made by the Rust API

5. Other Nonfunctional Requirements

5.1 Performance Requirements

No known performance requirements.

5.2 Safety Requirements

No known safety requirements.

5.3 Security Requirements

This application contains all of its information on a local Sqlite database file. Communication does occur with an external Monero RPC interface if need be for the Moneropay API to operate correctly.

5.4 Software Quality Attributes

This application is adaptable for future scenarios that may require more complexity. A terminal is required to access and use the application.

5.5 Business Rules

No special operating principles.

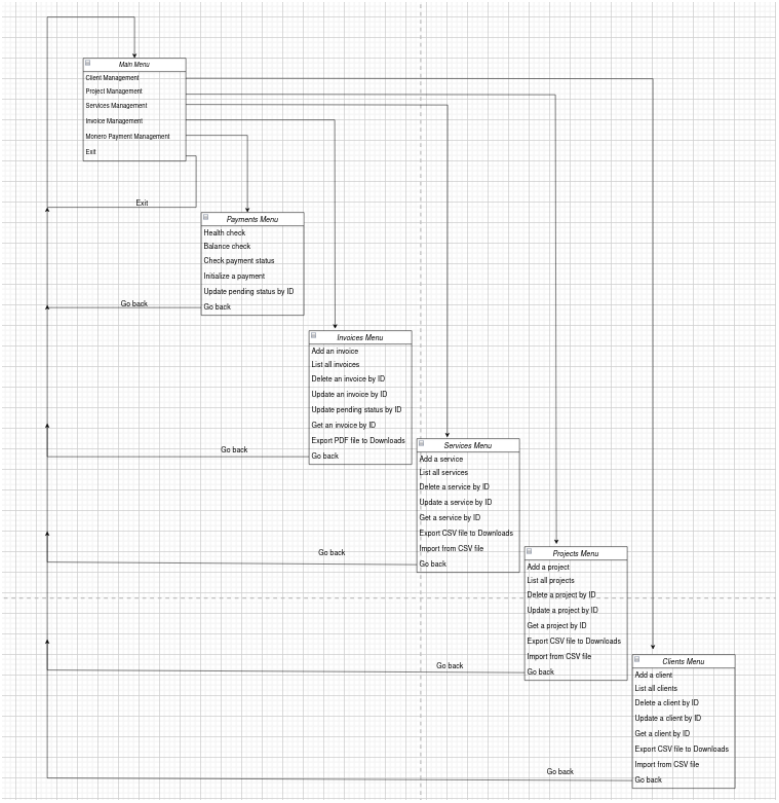
6. Other Requirements

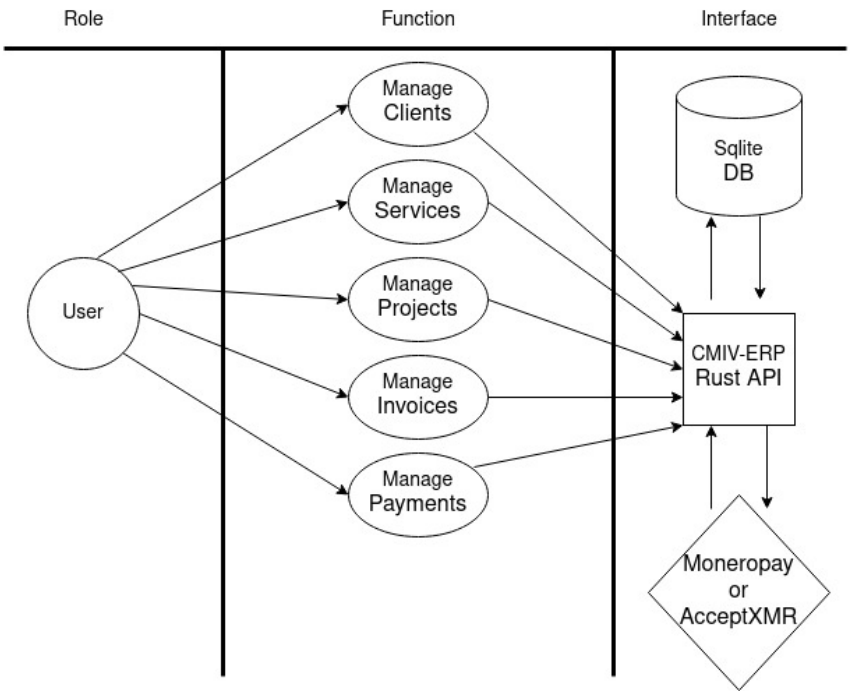
The code for The Oceanus ERP will be made open source and available for all to run themselves.

Appendix A: Glossary

- *ERP: A type of software that organizations and businesses use to manage their day-to-day activities such as payments, project management, services offered, etc.*
- *API: An intermediary layer or middleman that processes data between systems – like a user interface and a database.*
- *Database: An organized collection of data stored and accessed electronically, written in SQL (Structured Query Language).*
- *Rust: A relatively new programming language that began as a side project in 2006 by Graydon Hoare, a Mozilla employee. Rust is a systems programming language that adopts the approach of "zero cost abstraction". The main selling point is that the language maintains memory safety while not sacrificing speed or control.*
- *Monero (XMR): Private, decentralized cryptocurrency that specializes in the use-case of money. Monero is an electronic currency to be exchanged for goods and services that avoids the pitfalls of other conventional money – such as inflation, usury, etc.*
- *CRUD: The functions necessary for a persistent storage application: Create, Read, Update, and Delete*

Appendix B: Analysis Models





Appendix C: To Be Determined List

No TBD references with this version.