

# Individual Final Project BLJ

## Creating a Notes App using Rust

**Name:** by Matteo Luciano Siro Cipriani

**Abgabedatum:** June 3, 2025

**Lehrfirma:** Soreco AG

# Contents

<b>Contents</b>	<b>2</b>
<b>1 Versioning</b>	<b>3</b>
<b>2 Introduction</b>	<b>4</b>
2.1 Task Definition . . . . .	4
2.2 Project Description . . . . .	4
2.3 Known Risks . . . . .	4
<b>3 Planning</b>	<b>5</b>
3.1 Schedule . . . . .	5
3.2 Entscheidungsmatrix . . . . .	5
<b>4 Hauptteil</b>	<b>6</b>
4.1 Vorgehen und Zwischenschritte . . . . .	6
4.2 Ergebnis der Arbeit . . . . .	6
4.3 Funktionsbeschreibung . . . . .	6
4.4 Arbeitsjournal . . . . .	6
4.5 Testplan . . . . .	6
<b>5 Dailies</b>	<b>7</b>
5.1 Day 1: 02.06.2025 . . . . .	7
5.2 Day 2: 03.06.2025 . . . . .	7
<b>6 Appendix</b>	<b>8</b>
6.1 Sources . . . . .	8
6.2 Begriffserklärungen/Glossar . . . . .	8
6.3 Programm-Code . . . . .	8
6.4 Foto-Dokumentation . . . . .	8
6.5 KI-Chat-Auszüge . . . . .	8

# 1 Versioning

Version	Date	Time	Updates	Author
1.0.0	03.06.2025	13:06	Started Documentation	Matteo Cipriani
1.1.0	03.06.2025	14:27	Started Introduction	Matteo Cipriani
1.2.0	03.06.2025	15:08	Started Listing Sources	Matteo Cipriani

## 2 Introduction

### 2.1 Task Definition

At the end of the year, each apprentice who worked in the ZLI is required to do a project of their own choosing. They have to plan, execute and document an appropriate project over the span of 4 weeks, while working Monday - Wednesday (or Wednesday - Friday, depending on their school days). With this project, the apprentices can demonstrate what they have learned from the coaches during the last year, as all competences required to fulfill the project have been topics during this past year, some have been used very frequently, while others have only been discussed during 1 week.

### 2.2 Project Description

I chose to create a Notes App using Rust. I initially wanted to make a To-Do App, but as I have already done a To-Do App using Dart & Flutter as my Sportferienprojekt, I chose to go with something different. I want to try to write this project purely in Rust, to see how much of the language I have learned during the last year, and I can definitely learn new things from this project too. Because Rust is quite famous for being a really safe programming language, I want to try and implement one or two ways to encrypt and store the data safely.

### 2.3 Known Risks

I know that creating an application purely in Rust might be difficult, especially because Rust isn't really made to design, but to work. To implement a GUI, you have to use crates, which are known to sometimes be even more difficult than the standard Rust syntax itself. And Rust itself has a pretty steep learning curve too. Managing lifetimes, references, and borrowing can be complex, especially with dynamically changing data like note content. On top of that, Rust's error system (e.g., `Result` and `Option`) is safe but verbose, requiring you to explicitly handle many cases.

### 3 Planning

#### 3.1 Schedule

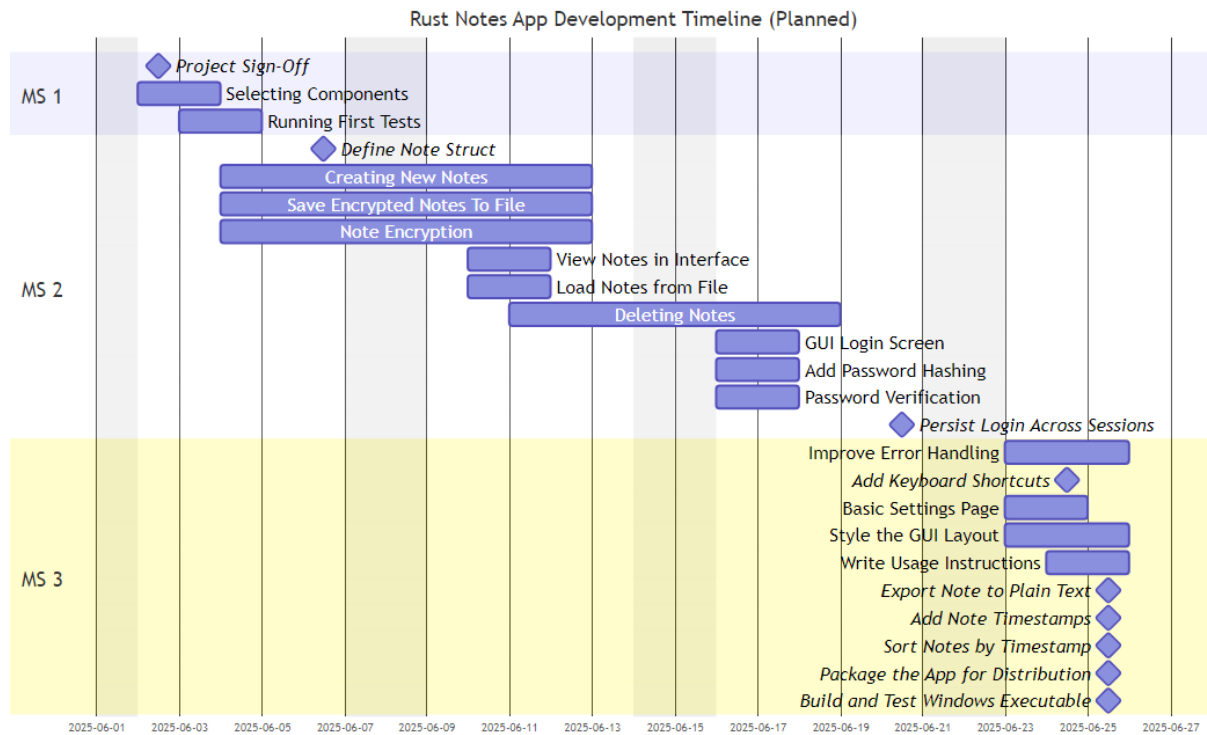


Figure 1: Schedule (Planned)

#### 3.2 Entscheidungsmatrix

Übersicht über mögliche Wege und Entscheidungskriterien

## **4 Hauptteil**

### **4.1 Vorgehen und Zwischenschritte**

Beschreibung des Vorgehens

### **4.2 Ergebnis der Arbeit**

Konstruktionszeichnungen, Zusammenstellungen etc.

### **4.3 Funktionsbeschreibung**

Mit Querverweis zum Anhang

### **4.4 Arbeitsjournal**

Dokumentation des Arbeitsprozesses

### **4.5 Testplan**

Beschreibung der durchgeführten Tests

## **5 Dailies**

**5.1 Day 1: 02.06.2025**

**5.2 Day 2: 03.06.2025**

## 6 Appendix

### 6.1 Sources

- [docs.rs](#)
  - Basic Documentation for all of the crates used
  - Further linking to official websites / GitHub repositories with official examples / code snippets
  - Structs, Enums, Functions, Models & Type Aliases for each crate (if available)
- [GitHub](#)
  - Extensive Documentation about crates
  - Function snippets
  - Implementation Methods
- [Rust's official website](#)
  - Basic Questions about Rust's functionality
  - Further linking to community boards
- [THE Rust Book](#)
  - Basic Introduction to Rust
  - Easy explanations for some more complicated topics of Rust
- [Rust By Example](#)
  - Examples for some crucial features
    - \* Examples are editable & executable
  - Good "playground"
- [The Cargo Book](#)
  - Guide through Rust's package manager
  - Easy point to access Features, Commands and general infos about cargo

### 6.2 Begriffserklärungen/Glossar

Definitionen wichtiger Begriffe

### 6.3 Programm-Code

Code-Beispiele und Scripts

### 6.4 Foto-Dokumentation

Relevante Bilder und Screenshots

### 6.5 KI-Chat-Auszüge

Dokumentation von KI-Unterstützung