

Master Thesis

Development of a Handheld Speed Camera System

3D Model Prototyping, GUI Design, and Programming

Muhammad Haziq Bin Mohd Sabtu



Technische Hochschule Brandenburg

Fachbereich Technik

Studiengang Maschinenbau

Supervisor 1: Prof. Dr.-Ing. Christian Oertel

Supervisor 2: ...

Brandenburg, den 25. Oktober 2027

Ich, VORNAME NAME, Student im Studiengang ??? der Technischen Hochschule Brandenburg, versichere an Eides statt, dass die vorliegende Abschlussarbeit selbstständig verfasst und nicht mit anderen als den angegebenen Hilfsmitteln erstellt wurde.

Sie wurde in dieser oder hnlicher Form noch keiner Prfungskommission vorgelegt.

VORNAME NAME

Contents

List of Abbreviations	I
1 Introduction	1
I Prototype Development	2
1 Methodology	3
1.1 Design Methodology	3
2 Task Clarification and Specification	4
2.1 Requirement of the Prototype	4
2.2 Requirement List	5
3 Concept Generation	6
3.1 Abstraction	6
3.2 Function Structure	6
3.3 Idea Generation	6
3.4 Combination of Ideas with Morphological Chart	7
4 Design	8
4.1 Concept Selection and Evaluation	8
4.1.1 Design 1	8
4.1.2 Design 2	8
4.1.3 Design 3	8
4.2 Final Design	8
4.2.1 CAD Drawing	9
4.2.2 Bill of Materials	9
5 Conclusion	10

II	GUI Development	11
1	Methodology	12
1.1	MVC Pattern	12
1.2	Design Patterns - Thread Pool	12
2	Requirements and Design	14
2.1	Requirements	14
2.2	Wireframe	14
2.3	GUI Design	15
3	Solutions and Implementations	16
3.1	Model	16
3.2	View	16
3.3	Controller	17
4	Testing	18
4.1	Unit Testing	18
5	Conclusion	19
III	Indexes and Appendix	20
	List of Figures	21
	List of Tables	21
	Bibliography	22
A	Appendix	24
A.1	CAD Drawings	24
A.2	Bill of Materials	24
A.3	Code snippets	24
A.4	Additional information, pictures, handout, etc.	24

Abstract

Abstract

Keywords: Keywords1, Keywords2, Keywords3

Abstract

Kurzfassung

Schlüsselwörter: Schlüsselwörter1, Schlüsselwörter2, Schlüsselwörter3

1 Introduction

Project Introduction

Part I

Prototype Development

1 Methodology

1.1 Design Methodology

Explanation of the design methodology from VDI 2221 [Con]

- What is VDI 2221 and what are its key principles?
- What are the main objectives and goals of VDI 2221?
- What are the key stages or phases outlined in VDI 2221?

2 Task Clarification and Specification

2.1 Requirement of the Prototype

List of requirements for the prototype

Must have:

- Ergonomic - Comfortable to hold, Easy to use
- Portable - Lightweight, Small
- Size (MAX)
 - Length: 25 cm
 - Width: 25 cm
 - Height: 25 cm
- Weight (MAX): 3 kg
- Compliance and Regulation - Comply with the regulations of the country of use
- Cost - Affordable, < 300 Euro (including Pi, Camera and Screen)
- Appointments - Completed within 3 months
- Design - Components are packed in a chasis
- Camera - Camera must be presented in the prototype
- Power - Battery powered, Rechargeable battery, Duration min. 1 hour

- Control - Control via touch screen

Optional Requirements:

- Durability - Water resistance, Dust resistance
- Modular - Easy to assemble and disassemble, Swappable parts
- Features - Mountable on a tripod
- Fertigung - 3D printed parts

2.2 Requirement List

List of requirements will be generated from the must have and optional requirements (Section [2.1](#))

3 Concept Generation

3.1 Abstraction

- What is Abstraction?
- How does it defined and utilized in the design process?
- What are the benefits of using abstraction?

3.2 Function Structure

- What is a function structure?

3.3 Idea Generation

This section will discuss the methods used for idea generation.

Methods used:

- Market Research
- Competitive Analysis
- Brainstorming

Method is suitable, due to the fact that handheld devices are common in the market

3.4 Combination of Ideas with Morphological Chart

List of ideas from brainstorming will be combined with the function structure to generate a morphological chart

Atleast 3 Design Concepts will be generated from the morphological chart

4 Design

4.1 Concept Selection and Evaluation

- Explanation of the design and discussion of advantages and disadvantages
- What are the performance characteristics and limitations of each design option, and how do they align with the desired outcomes?
- What are the cost implications associated with each design option?
- What are the potential risks and uncertainties associated with each design option, and how can they be mitigated or managed effectively?

4.1.1 Design 1

4.1.2 Design 2

4.1.3 Design 3

4.2 Final Design

- How is the final design selected?
- What methods are used to evaluate the final design?

- Which evaluation criteria are being used?
- How well does each design option fulfill the functional requirements specified in VDI 2221?

4.2.1 CAD Drawing

Final CAD Design will be presented here. Including with the features

4.2.2 Bill of Materials

List of parts used in the final design

5 Conclusion

Conclusion of the project

Part II

GUI Development

1 Methodology

1.1 MVC Pattern

- What is MVC?
- What are the distinct responsibilities and roles of the Model, View, and Controller components in the MVC pattern?
- What are the benefits of using MVC?

The Model-View-Controller (MVC) pattern is a software architectural pattern that separates an application into three interconnected components: the model, the view, and the controller. The model represents the data and logic of the application, the view displays the data to the user, and the controller handles user input and updates the model and view accordingly. This pattern promotes separation of concerns, modularity, and code reusability in software development. [Ver19]

1.2 Design Patterns - Thread Pool

- What is a thread pool?
- What are the benefits of using a thread pool?
- What are the drawbacks of using a thread pool?

A thread pool is a software design pattern that manages a pool of worker threads to efficiently execute tasks. Instead of creating a new thread for each task,

a thread pool reuses existing threads, minimizing the overhead of thread creation. It improves performance and resource utilization by limiting the number of concurrent threads and providing a queue to handle incoming tasks.[Bro22]

2 Requirements and Design

2.1 Requirements

Must have:

- Usability - Easy to use
- Performance - Fast processing by utilising multiple threads
- Responsiveness - Responsive GUI, avoid methods that block the GUI thread
- Error Handling - Handle errors gracefully, avoid crashing the application
- Scalability - For future development
- Documentation - user guides, Tooltips, comments
- Design - Clean and simple design

2.2 Wireframe

Program flow and GUI design will be presented in a wireframe.

* Flow of the program is not finalized, will be updated in the future

- All panels involved in the program will be presented here
- Flow of the program will be presented here.
- Which panel comes before and after another panel will be presented here

- What happens when the user clicks on a button will be presented here

2.3 GUI Design

Design of the GUI will be presented here. Panels, Buttons, Textfields, etc.

- Layout of the GUI will be defined here
- What panels will be used will be defined here

3 Solutions and Implementations

In this chapter, the solutions and implementations of the project will be presented.

3.1 Model

Implementation of the Model

- What is the Model?
- What are the key responsibilities of the Model?
- What is the primary purpose and responsibility of the Model component in the application's architecture?

3.2 View

Implementation of the View

- What is the View?
- What are the key responsibilities of the View?
- How does the View handle the presentation and visualization of data to the user?
- How does the View respond to user input and events, and how are these interactions managed?

- What are the mechanisms for updating the View based on changes in the Model or instructions from the Controller?

3.3 Controller

Implementation of the Controller

- What is the Controller?
- What are the key responsibilities of the Controller?
- How does the Controller handle user input and events?
- How does the Controller update the Model and View?

4 Testing

4.1 Unit Testing

Unit testing is a software testing approach that involves testing individual components or units of code in isolation to ensure they function correctly. It verifies the behavior of small, independent units of code, such as functions or methods, to validate their expected functionality and catch any defects early in the development process. [Ham23]

5 Conclusion

Conclusion of the project

Part III

Indexes and Appendix

List of Figures

List of Tables

Bibliography

- [Bro22] BROWNLEE, JASON: *Threadpoolexecutor in Python: The complete guide*, Dec 2022.
- [Con] CONRAD, KLAUS-JÖRG: *Konstruktionsphase Konzipieren*, pages 169–249.
- [Ham23] HAMILTON, THOMAS: *Unit testing tutorial - what is, Types; Test example*, May 2023.
- [Ver19] VERSTEHEN, DATENBANKEN: *Model View Controller Pattern Definition; Erklärung: Datenbank, DWH; Bi Lexikon*, Oct 2019.

A Appendix

- [Docs](#)
- [Repository](#)

A.1 CAD Drawings

A.2 Bill of Materials

A.3 Code snippets

A.4 Additional information, pictures, handout, etc.