

System Design for Flick Picker

Team 7, 7eam

Talha Asif - asift

Jarrold Colwell - colwellj

Madhi Nagarajan - nagarajm

Andrew Carvalino - carvalia

Ali Tabar - sahraeia

January 18, 2023

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
Flick Picker	Explanation of program name
[... —SS]	[... —SS]

Contents

1	Revision History	i
2	Reference Material	ii
2.1	Abbreviations and Acronyms	ii
3	Introduction	1
4	Purpose	1
5	Scope	1
6	Project Overview	1
6.1	Normal Behaviour	1
6.2	Undesired Event Handling	1
6.3	Component Diagram	1
6.4	Connection Between Requirements and Design	1
7	System Variables	1
7.1	Monitored Variables	2
7.2	Controlled Variables	2
7.3	Constants Variables	2
8	User Interfaces	2
9	Design of Hardware	2
10	Design of Electrical Components	2
11	Design of Communication Protocols	2
12	Timeline	2
A	Interface	3
B	Mechanical Hardware	3
C	Electrical Components	3
D	Communication Protocols	3
E	Reflection	3
E.1	Talha Reflection	3

List of Tables

List of Figures

3 Introduction

[Include references to your other documentation —SS]

4 Purpose

[Purpose of your design documentation —SS]
[Point to your other design documents —SS]

5 Scope

[Include a figure that show the System Context (showing the boundary between your system and the environment around it.) —SS]

6 Project Overview

6.1 Normal Behaviour

6.2 Undesired Event Handling

[How you will approach undesired events —SS]

6.3 Component Diagram

6.4 Connection Between Requirements and Design

[The intention of this section is to document decisions that are made “between” the requirements and the design. To satisfy some requirements, design decisions need to be made. Rather than make these decisions implicit, they are explicitly recorded here. For instance, if a program has security requirements, a specific design decision may be made to satisfy those requirements with a password. —SS]

7 System Variables

[Include this section for Mechatronics projects —SS]

7.1 Monitored Variables

7.2 Controlled Variables

7.3 Constants Variables

8 User Interfaces

[Design of user interface for software and hardware. Attach an appendix if needed. Drawings, Sketches, Figma —SS]

9 Design of Hardware

[Most relevant for mechatronics projects —SS] [Show what will be acquired —SS] [Show what will be built, with detail on fabrication and materials —SS] [Include appendices as appropriate, possibly with sketches, drawings, CAD, etc —SS]

10 Design of Electrical Components

[Most relevant for mechatronics projects —SS] [Show what will be acquired —SS] [Show what will be built, with detail on fabrication and materials —SS] [Include appendices as appropriate, possibly with sketches, drawings, circuit diagrams, etc —SS]

11 Design of Communication Protocols

[If appropriate —SS]

12 Timeline

[Schedule of tasks and who is responsible —SS]

A Interface

[Include additional information related to the appearance of, and interaction with, the user interface —SS]

B Mechanical Hardware

C Electrical Components

D Communication Protocols

E Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)
2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)

E.1 Talha Reflection

1. Our solution's two most significant limitations are the number of developers and time constraints. As 7team only has five developers with solely software engineering experience, we need to learn how to market a product. We also need multiple years of development experience to avoid pitfalls that individuals who have worked in the industry would know to avoid. If we were given an unlimited number of individuals to make an entire development team with dedicated teams per dev-op, frontend, backend, and business teams, the application would see a much more significant influx of users and have an incredibly refined design. Furthermore, we are constrained by time, not just by the due dates but by how much content we five individual developers can put in. We could refine the application incredibly well with infinite time and no other resources.
2. Another design solution was a minor decision about pairing and preferences together. The benefit is that they go hand-in-hand, where users are directly linked to their preferences. The downside being it adds an overhead to updating the preferences list, and since we are going to take into account user feedback for the preferences, it could go through a handful of updates. Therefore splitting it off into a module in isolation and linking the user is a safer implementation.