

Heju Chung
Sean Kruse



*Morse Code Password System:
A Touch-Based Approach to
Security and Accessibility*

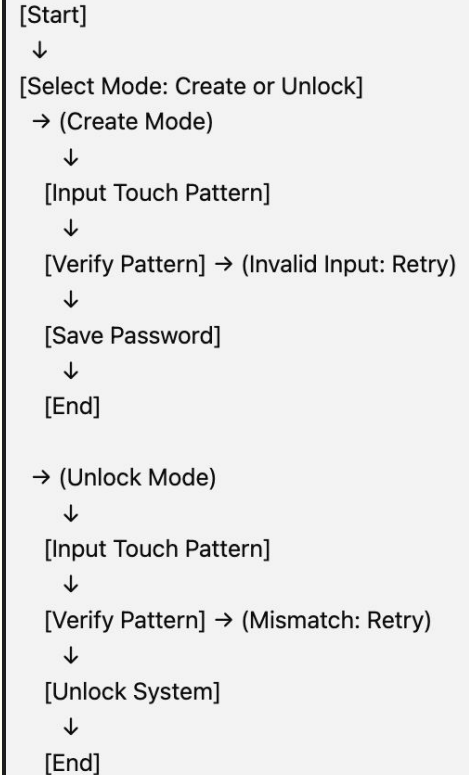
summary of the flow

Create Mode

1. **Start** → The process begins.
2. **Select Mode** → User selects "Create Mode."
3. **Input Touch Pattern** → User enters a new touch pattern.
4. **Verify Pattern** → System checks the pattern:
 - If invalid: prompt to retry.
5. **Save Password** → Successful verification saves the pattern.
6. **End** → The process concludes.

Unlock Mode

1. **Start** → The process begins.
2. **Select Mode** → User selects "Unlock Mode."
3. **Input Touch Pattern** → User enters their touch pattern.
4. **Verify Pattern** → System checks the pattern:
 - If mismatch: prompt to retry.
5. **Unlock System** → Successful verification unlocks the system.
6. **End** → The process concludes.



Main Objectives



1. Designing a user-friendly interface for touch-based password entry.

2. Collecting and analyzing user interaction data to evaluate usability.

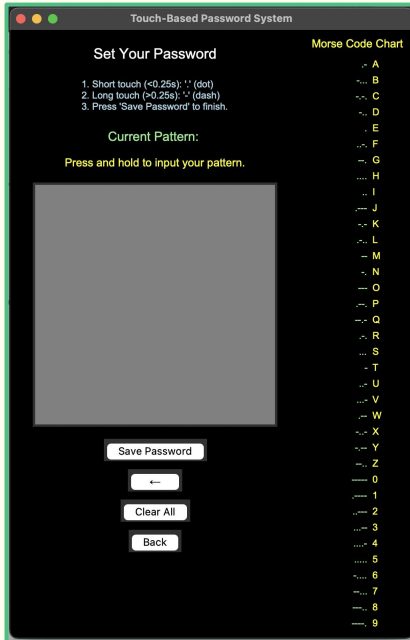
3. Ensuring security and adaptability for users with diverse needs.

```
1 import tkinter as tk
2 import time
```

```
def show_main_menu(self):
    self.clear_screen()
    self.current_screen = "main_menu"

    tk.Label(self.root, text="Morse Code Password System", font=("Arial", 18), fg="white", bg="black").pack(pady=10)

    tk.Button(self.root, text="Set Password", command=self.show_set_password_screen, bg="darkgray", fg="black").pack(pady=5)
    tk.Button(self.root, text="Unlock Password", command=self.show_unlock_screen, bg="darkgray", fg="black").pack(pady=5)
```



```
def stop_timer(self, event):
    elapsed_time = time.time() - self.start_time
    input_type = "short" if elapsed_time < 0.25 else "long"

    self.log_touch_data(elapsed_time, input_type)

    if self.morse_code and (time.time() - self.last_input_time) > 0.5:
        if self.morse_code[-1] != " ":
            self.morse_code.append(" ")

    if elapsed_time < 0.25:
        self.morse_code.append('.')
        self.action_label.config(text="Short touch recorded (.)", fg="lightblue")
    else:
        self.morse_code.append('-')
        self.action_label.config(text="Long touch recorded (-)", fg="lightblue")

    self.last_input_time = time.time()
    self.update_display()
```

```
def stop_timer(self, event):
    elapsed_time = time.time() - self.start_time
    input_type = "short" if elapsed_time < 0.25 else "long"

    self.log_touch_data(elapsed_time, input_type)

    if self.morse_code and (time.time() - self.last_input_time) > 0.5:
        if self.morse_code[-1] != " ":
            self.morse_code.append(" ")

    if elapsed_time < 0.25:
        self.morse_code.append('.')
        self.action_label.config(text="Short touch recorded (.)", fg="lightblue")
    else:
        self.morse_code.append('-')
        self.action_label.config(text="Long touch recorded (-)", fg="lightblue")
```

Set Your Password

1. Short touch (<0.25s): '.' (dot)
2. Long touch (>0.25s): '-' (dash)
3. Press 'Save Password' to finish.

```
def set_password(self):
    """Save the user's Morse code password."""
    if self.morse_code:
        morse_string = ''.join(self.morse_code).strip().replace(" ", " ")
        translated_password = self.convert_to_morse_sequence(morse_string)
```



Hypothesis



The introduction of a touch-based Morse code password system enhances user interaction by providing an intuitive and accessible security mechanism. Furthermore, it can serve as an alternative password input method, especially for users with disabilities, while maintaining a high level of usability and adaptability.

Key Elements of the Hypothesis:

**Intuitive and
User-Friendly**

**Inclusivity
and
Accessibility**

**Security and
Flexibility**

**Learning
Curve
Effectiveness**

**Verification of
the
Hypothesis**



'touch_durations.csv'

```
1 import tkinter as tk
2 import time
3 import csv
```



'touch_duration_report.csv'

Heju's code

```
import pandas as pd
import matplotlib.pyplot as plt

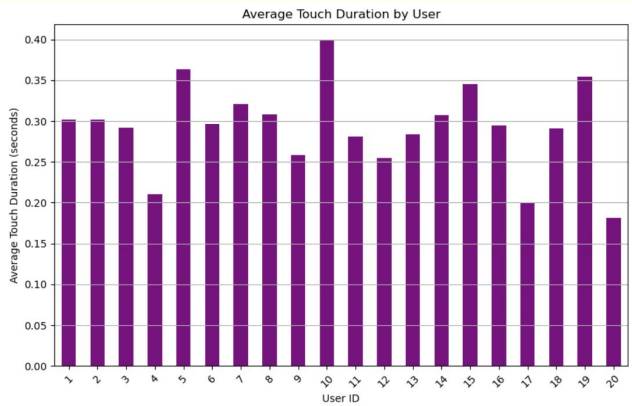
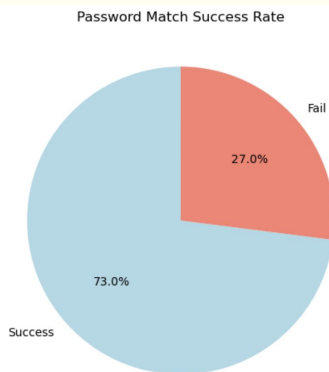
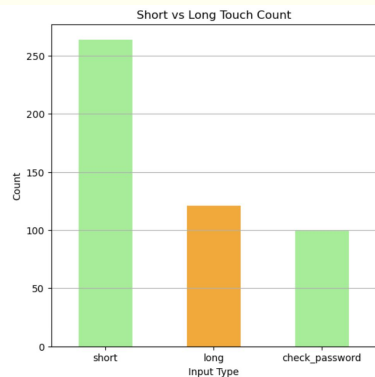
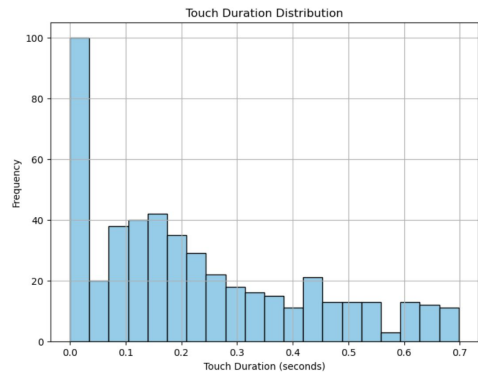
data = pd.read_csv("touch_duration_report.csv")

print(data.head())
```

	User ID	Password	Touch Duration	Input Type	Success/Fail
0	1	Password_1	0.078980	short	NaN
1	1	Password_1	0.658972	short	NaN
2	1	Password_1	0.238199	long	NaN
3	1	Password_1	0.000000	check_password	Success
4	1	Password_2	0.144071	long	NaN

My code

Data Analyze

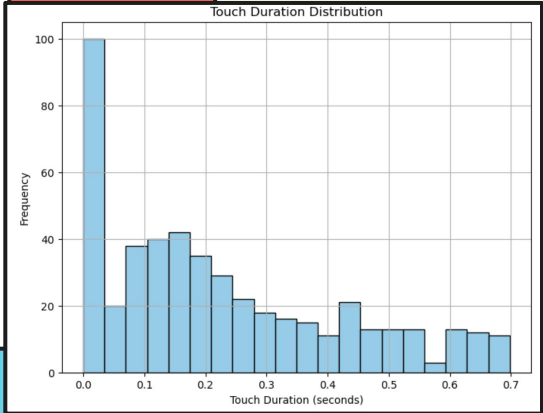
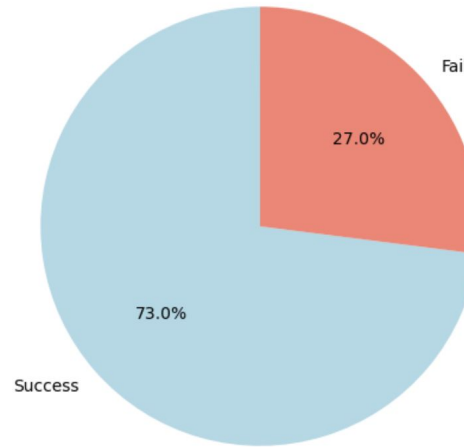


Key Findings from the Analysis:

73% success rate for password attempts and a 27% failure rate

Frequency decreasing as the duration increased

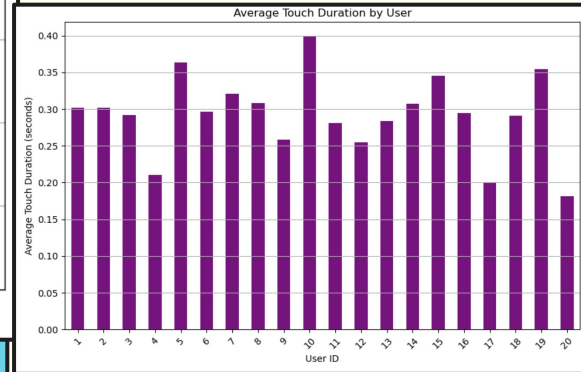
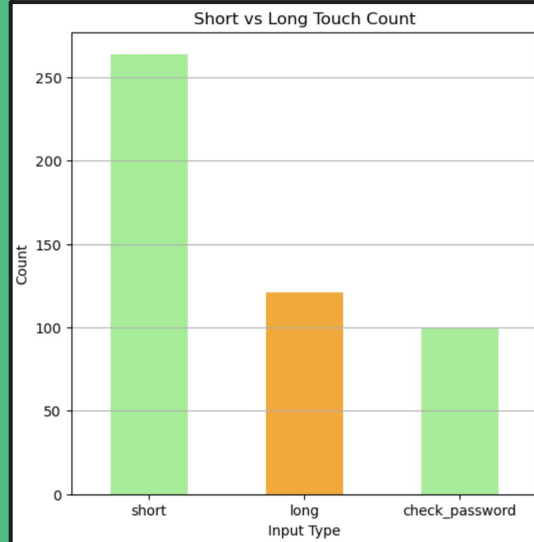
Password Match Success Rate



Key Findings from the Analysis:

Short touches were significantly more frequent than long touches or password-check inputs.

Average touch durations varied across users, but most remained within a consistent range



Implications:

Touch-based password system is both user-friendly and effective, with the majority of users quickly adapting to the short-touch mechanism.

However, the 27% failure rate highlights opportunities for improvement in reducing errors and enhancing the learning experience for new users.

Survey Results Summary

- **Usability Ratings (Scale: 1–5):**
 - An average score of **3.7** indicates that most participants found the system easy to use and intuitive.
- **Accessibility Ratings for Special Needs Users:**
 - An average score of **4.25** underscores the system's strong potential as an inclusive tool for users with disabilities.





THANK YOU

A large light blue rectangle with a black border, containing the text "THANK YOU". A yellow square is attached to its left side, and a red square is attached to its bottom right corner.