A

Report on

PHONE LOCK AND UNLOCK MANAGEMENT SYSTEM

CS205ES-PYTHON PROGRAMMING PROJECT

Submitted in partial fulfillment of the requirements for

The Award of the degree of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

By

K.PRASHANTH – 22R01A0523

Abstract

The Phone Lock and Unlock management system is a software application developed to enhance the security and usability of mobile phones. This system provides a convenient way to lock and unlock using various authentication methods,ensuring that only authorized users can access the device's contents . The system is built using the python programming language ,leveraging its versatility and extensive libraries .It utilizes a combination user interface components,database management ,and authentication mechanisms to deliver a seamless user user experience while maintaining high-security standards . Users can lock and Unlock their phones using the selected authentication method. The system securely stores the user's chosen lock mechanisms and ensures that unauthorized access is prevented . The system employs strong encryption algorithms to store sensitive user data securely . It ensures that personal information , including authentication credentials ,is protected from unauthorized access. Overall ,the phone lock and Unlock management system provides an effective and secure solution to protect personal data and prevent unauthorized access to mobile devices.Its user-friendly interface and robust authentication mechanisms make it an essential tool for mobile phone Security..

CONTENTS

1. INTRODUCTION

In this project, we will be building a phone lock and unlock management system using Python and data structures. This system will allow users to lock and unlock their phones using a secure mechanism. We will utilize data structures such as dictionaries, lists, and sets to implement this functionality.

The main objective of this system is to provide an efficient and secure way for users to protect their phones from unauthorized access. By implementing a lock and unlock mechanism, users can ensure that their personal data and information remain secure.

Key Features:

1. User Registration: Users can register their phone numbers along with a unique PIN (Personal Identification Number) to set up their accounts.
2. Phone Locking: Once registered, users can lock their phones by entering their PIN. The system will validate the entered PIN and lock the phone.
3. Phone Unlocking: To unlock the phone, users need to enter their PIN correctly. If the PIN matches, the system will unlock the phone.
4. PIN Recovery: In case a user forgets their PIN, they can request a PIN recovery option. This will prompt the user to answer security questions or perform a secondary authentication process to regain access to their phone.
5. Data Storage: User information, including phone numbers, PINs, and security questions, will be stored using data structures such as dictionaries and sets to ensure efficient access and retrieval.

2. Literature Survey

2.1 PYTHON AND ITS APPLICATIONS

Python is a versatile and widely used programming language known for its simplicity readability and ease of use it has broad range of applications across various fields and industries due to its powerful libraries, extensive community support and cross platform compatibility some of the key applications of python include:

Web Development, Data Science, Artificial Intelligence and Machine Learning, Scientific Computing, Automation and Scripting, Game Development, Desktop Applications.

2.2 VERSIONS OF PYTHON

Python 1.x
Python 2.x
Python 3.x

Python 1.x:

The initial release of python introducing the core language and its fundamental features added support for functional programming tools like Lambda map philtre and reduce functions introduce list comprehensions and added support for Unicode.

Python 2.x:

Added list comprehensions, garbage collection, and support for Unicode. Introduced a cycle detecting garbage collector and extended Unicode support. Added support for garbage collection of cycles and a unified and faster method resolution order. Added support for conditional expressions (if-else expressions), function annotations, and the with statement for file handling.

Python 3.x:

Introduced many changes to improve language consistency and remove legacy features. Not backward compatible with Python 2. Added improvements to the unittest module and other minor enhancements. Introduced the concurrent. futures module, updates to the pickle module, and improved SSL support. Added enhancements to the yield from expression, exception chaining, and improved library modules.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

These systems often involve a combination of hardware and software solutions to provide security and ease of use.Keep in mind that advancements may have occurred since then.Here are some of the common methods and technologies used in Phone Lock and unlock Management system.

1. **password/PIN:** This is the traditional method where users set a combination of numbers or characters that they need to enter to unlock the phone .It’s widely used but ,might not be the most secure method ,as passwords or PIN can be easily guessed or forgotten.

2. Smart Lock : This feature keep the phone unlocked as long as it’s on your person or moving ,utilizing sensors like the accelerometer to determine its status.

3. **Emergency Unlock:** Some phones have emergency unlock options that allow users to quickly access their device in urgent situations ,even if they can’t provide their regular unlock method.

4. SYSTEM REQUIREMENTS

Some of the System requirements are listed below:

1. Authentication /Authorization
2. Locking and Unlocking Mechanisms
3. User management
4. Security
5. Mobile App

4.1 SOFTWARE REQUIREMENTS

Python
Tkinter Library
Text Editor

4.2 HARDWARE REQUIREMENTS

Laptop/Pc….

5. ALGORITHM

Step 1: Start

Step 2: The code starts by importing the necessary modules from the Tkinter library. It also imports the `message box` and `simple dialouge` modules for displaying messages and dialouges.

Step 3: The main functionality of the app is encapsulated within this class. The class constructor (`__init__`) initializes the GUI elements and sets up the initial state of the phone (locked) and the default password.

Step 4: The GUI elements include:

- `lock_button`: A button that triggers the `lock_phone` method.
- `unlock_button`: A button that triggers the `unlock_phone` method.
- `change_password_button`: A button that triggers the `change_password` method.
- `status_label`: A label to display the current status of the phone (locked or unlocked).

Step 5: This method updates the state (enabled or disabled) of the buttons and the text of the status label based on whether the phone is locked or unlocked.

Step 6: This method is called when the "LockPhone" button is clicked. If the phone is already locked, a message is displayed. Otherwise, a password prompt appears using a simple dialog. If the entered password matches the stored password, the phone is locked, and the button states are updated accordingly.

Step 7: Similar to the `lock_phone` method, this method handles the unlocking process. It checks if the phone is already unlocked, prompts for the password, and unlocks the phone if the correct password is entered.

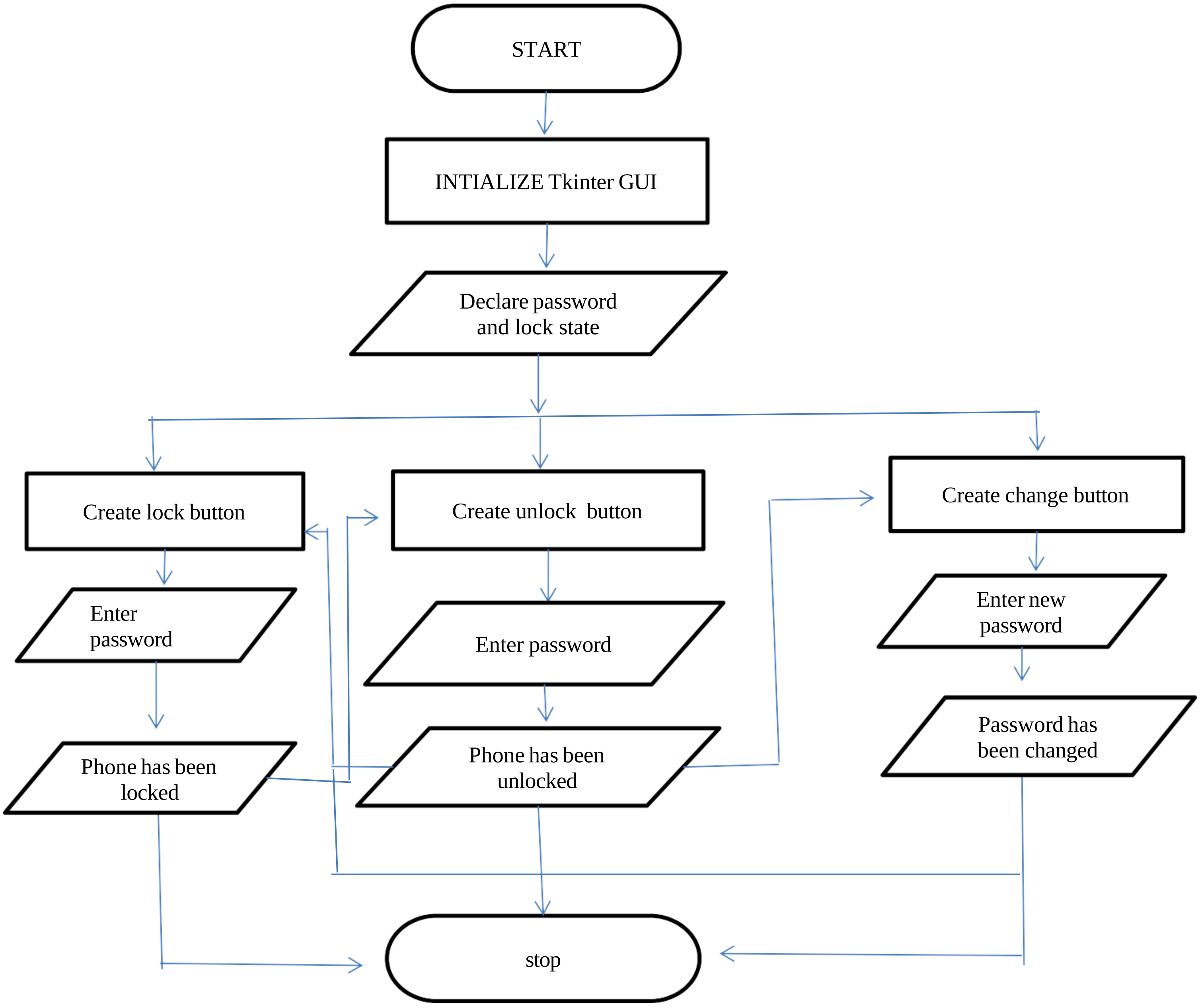
Step 8: This method allows changing the password. If the phone is unlocked, a dialog prompts for a new password. If a valid new password is provided, it is updated, and a message is displayed. If the phone is locked, a warning message is displayed.

Step 9: The `if __name___== "__main__":` block is the starting point of the script. It creates a Tkinter root window, initializes the `PhoneLockApp` class, and enters the main event loop (`root.mainloop()`) to display the GUI and respond to user interactions.

Step 10: Stop

6. SYSTEM ARCHITECTURE

6.1 FLOW CHART



7. SYSTEM IMPLEMENTATION

7.1 SOURCE CODE

```
 import tkinter as tk
from tkinter import messagebox, simpledialog

class PhoneLockApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Phone Lock & Unlock Management System")

        self.password = "1234"
        self.is_locked = True

       self.lock_button=tk.Button(root,text="LockPhone",command=self.lock_phone)

        self.unlock_button=tk.Button(root,text="UnlockPhone",command=self.unloc
        k_phone)
        self.change_password_button = tk.Button(root, text="Change Password",
command=self.change_password)
        self.status_label = tk.Label(root, text="Phone is locked")

        self.update_button_states()

        self.lock_button.pack(pady=10)
        self.unlock_button.pack(pady=10)
        self.change_password_button.pack(pady=10)
        self.status_label.pack(pady=20)

    def update_button_states(self):
        if self.is_locked:
            self.lock_button.config(state=tk.DISABLED)
            self.unlock_button.config(state=tk.NORMAL)
            self.change_password_button.config(state=tk.DISABLED)
            self.status_label.config(text="Phone is locked")
        else:
            self.lock_button.config(state=tk.NORMAL)
            self.unlock_button.config(state=tk.DISABLED)
            self.change_password_button.config(state=tk.NORMAL)
            self.status_label.config(text="Phone is unlocked")

    def lock_phone(self):
        if self.is_locked:
            messagebox.showinfo("Already Locked", "Phone is already locked.")
        else:
            password_input = simpledialog.askstring("Lock Phone", "Enter password:")
            if password_input == self.password:
                self.is_locked = True
                self.update_button_states()
                messagebox.showinfo("Phone Locked", "Phone has been locked.")
            else:
                messagebox.showerror("Incorrect Password", "Incorrect password entered.")

    def unlock_phone(self):
        if not self.is_locked:
            messagebox.showinfo("Already Unlocked", "Phone is already unlocked.")
        else:
            password_input = simpledialog.askstring("Unlock Phone", "Enter password:")
            if password_input == self.password:
                self.is_locked = False
                self.update_button_states()
                messagebox.showinfo("Phone Unlocked", "Phone has been unlocked.")
            else:
                messagebox.showerror("Incorrect Password", "Incorrect password entered.")

    def change_password(self):
        if not self.is_locked:
            new_password = simpledialog.askstring("Change Password", "Enter new password:")
            if new_password:
                self.password = new_password
                messagebox.showinfo("Password Changed", "Password has been changed successfully.")
        else:
            messagebox.showwarning("Phone Locked", "Please unlock the phone to
change the password.")

if __name__== "__main__":
    root = tk.Tk()
    app = PhoneLockApp(root)
    root.mainloop()
```

8. SYSTEM TESTING

8.1 TEST CASE 1 : Unlock the password

Input : 1234

Output : phone has been unlocked

8.2 TEST CASE 2 : If Password is wrong

Input :6343

Output: Incorrect password entered

8.3 TESTCASE 3: Change password

Input :1234567890

Output : Your pass is changed

8.4 TEST CASE 4: Phone Lock

Input : 1234567890

Output : Your Phone is Locked

8.5 TEST CASE 5: Unlock Password

Input : 1234567890

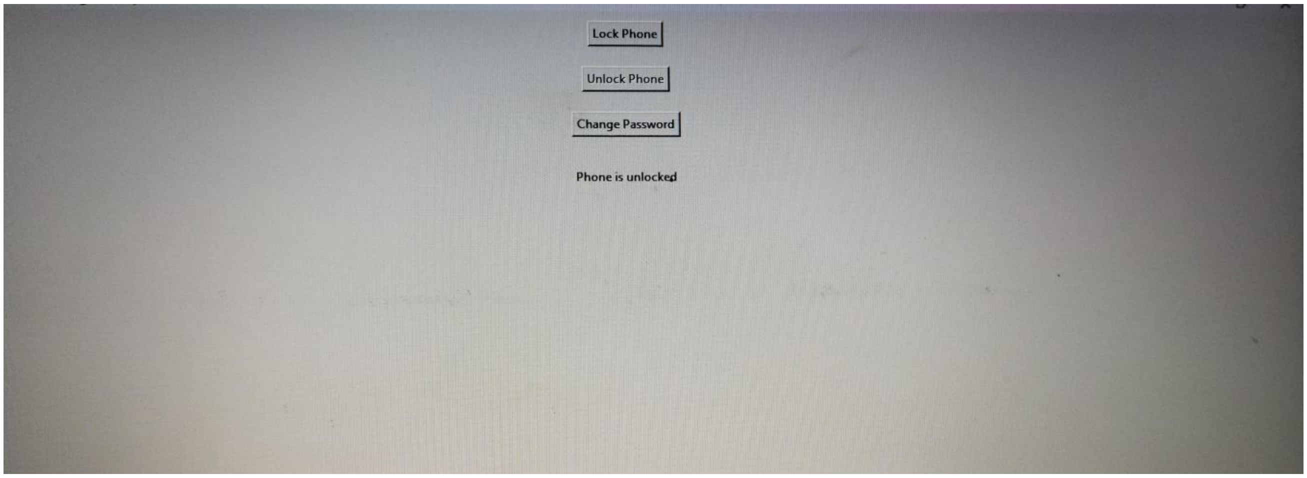
Output : Phone has been Unlocked

8.6 TEST CASE 6: Lock phone

Input : 1234567890

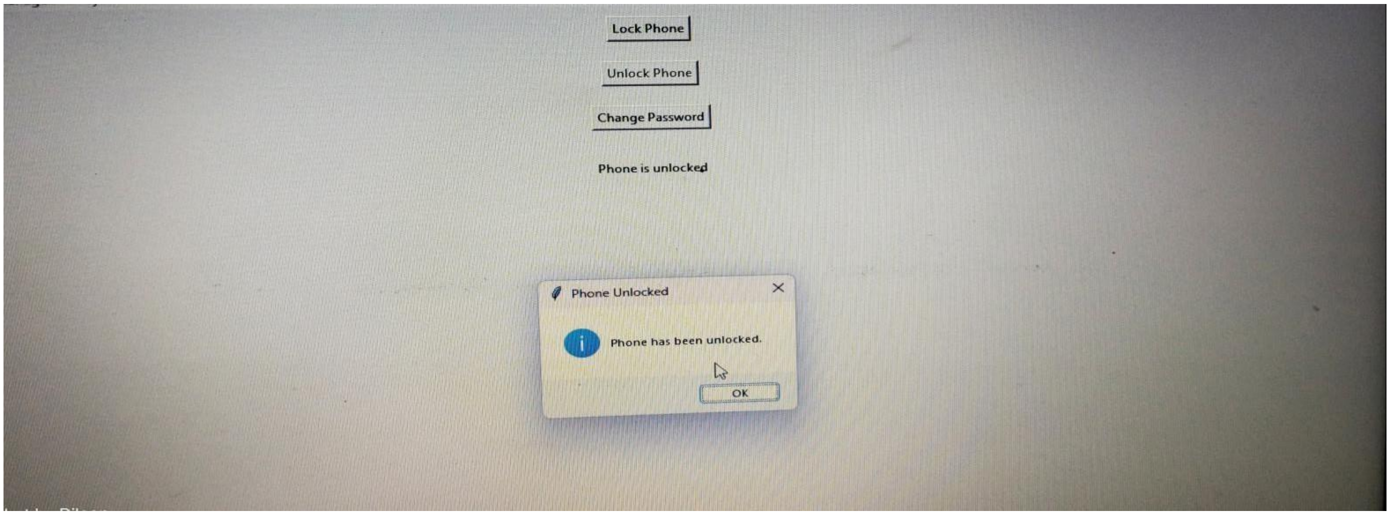
Output : Phone is Locked.

9. OUTPUT SCREENS



Phone is unlocked

9.1 Home Window

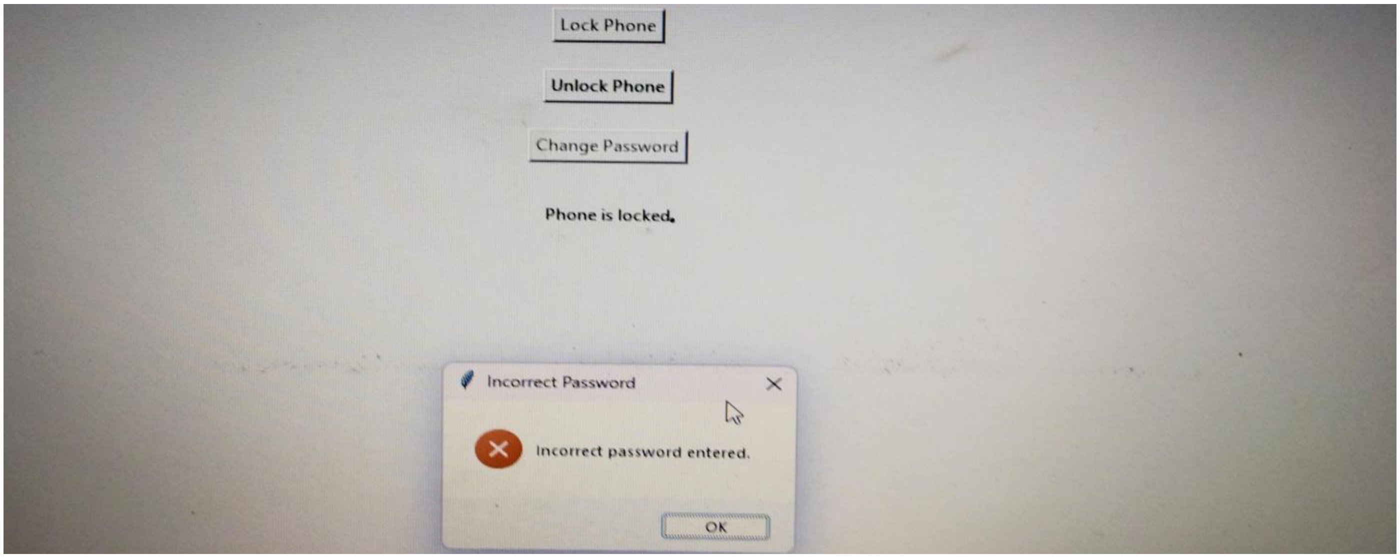


Phone is unlocked

Phone Unlocked

Phone has been unlocked.

9.2 Phone has been Unlocked Successfully

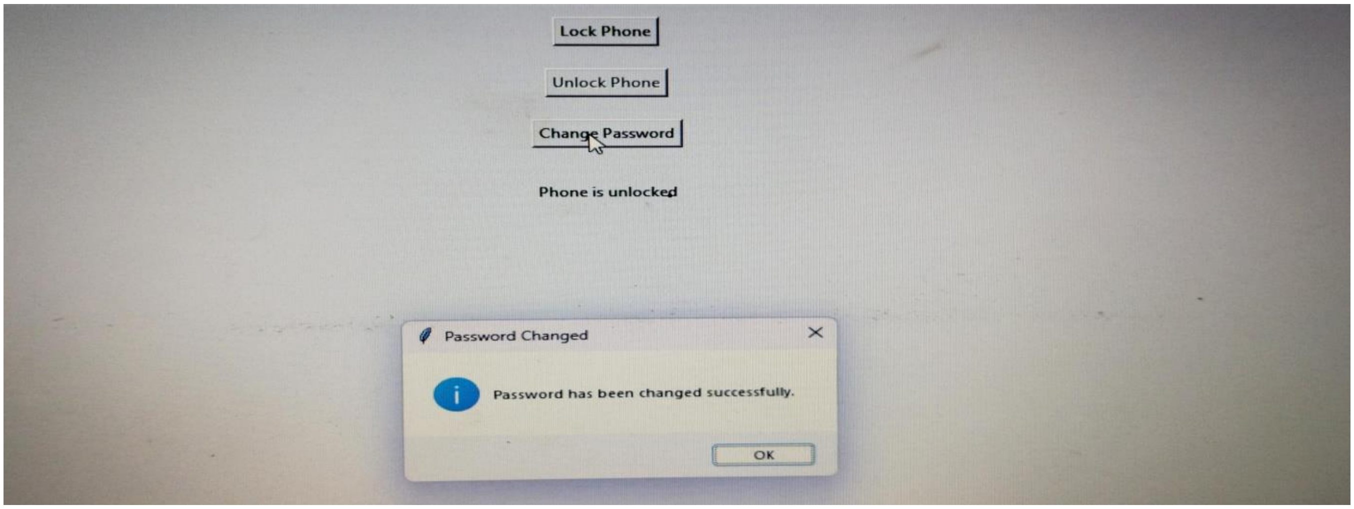


Phone is locked.

Incorrect Password

Incorrect password entered.

9.3 When the Incorrect password entered

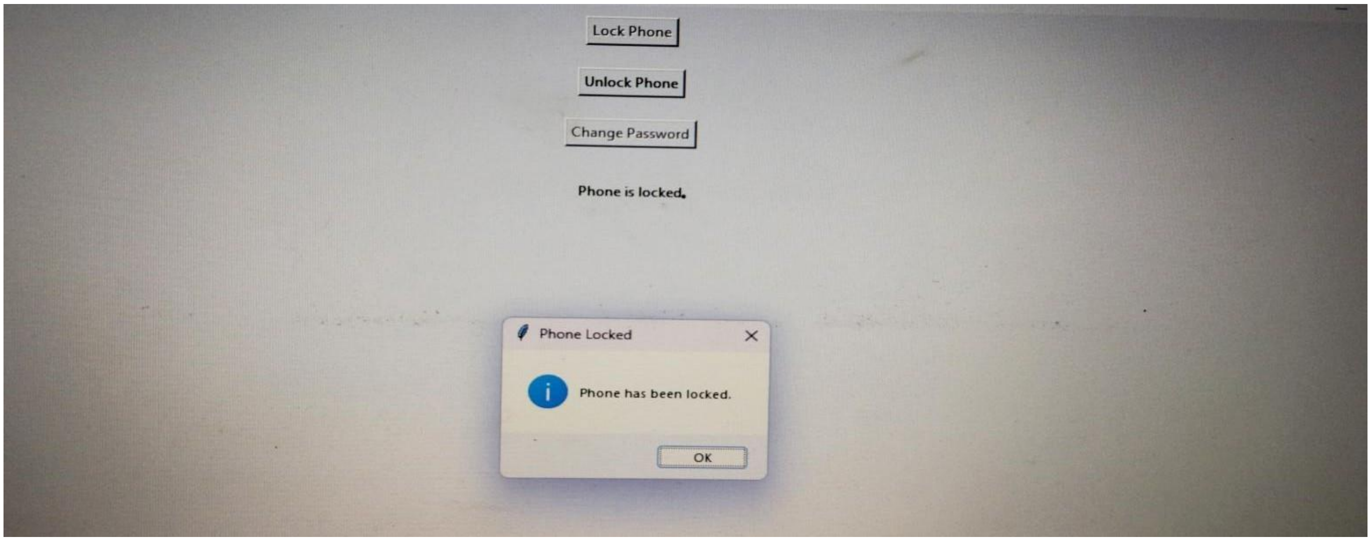


Phone is unlocked

Password Changed

Password has been changed successfully.

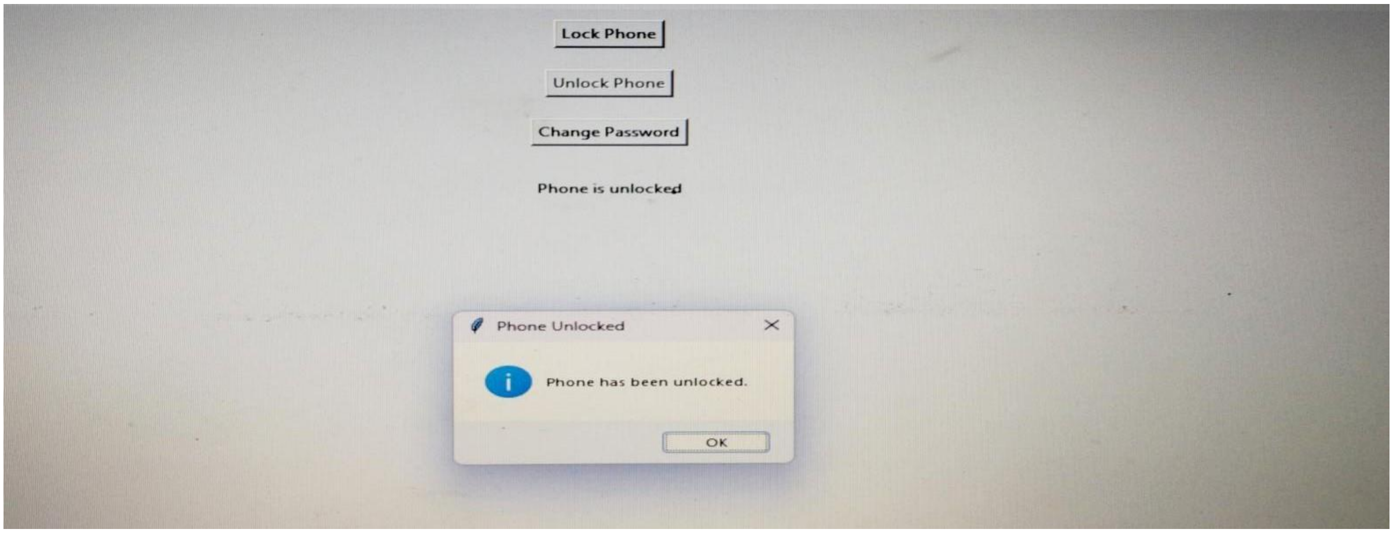
9.4 Password has been changed successfully



Phone is locked.

Phone has been locked.

9.5 Phone has been Locked



Phone is unlocked

Phone has been unlocked.

9.6 Phone has been Unlocked

Conclusion

In conclusion, the phone lock and unlock management system presents an essential solution in ensuring the security ,privacy,and convenience of mobile device user. Through a combination of robust authentication methods,intelligent algorithms,and user-friendly interfaces ,the system effectively safeguards sensitive data and mitigates unauthorized access.By implementing a multi-layered authentication process ,including PIN and device recognition ,the system ensures that only authorized users can access the device. This significantly reduces the risk of unauthorized data breaches and enhances user confidence in the security of their personal information. In the era where digital privacy and security are paramount concerns, the phone lock and unlock management system emerges as a crucial tool for individual and organizations alike.Its integration of advanced security measures and user- Centric features offers a balanced solution that upholds privacy while embracing technological convenience . As the digital landscape evolves ,this system stands as a become of trust,safeguarding sensitive information in an increasingly interconnected world…

FUTURE ENHANCEMENT

1. Behavioral Authentication:

- **Gesture Recognition:** Utilize machine learning to recognize unique user gestures or patterns ,adding an extra layer of personalization to authentication.

- **Typing patterns:** Analyze a use's typing speed ,rhythm,and patterns to detect unauthorized access attempts.

2. Security Recovery:

- **Trusted Contacts :** Integrate a mechanism for users to designate trusted contacts who can help them regain access in case of emergency.

- **Backup passcodes:** Offer an option for users to set backup passcodes or patterns to recover their device.

3. Blockchain Technology

- **Utilize blockchain for authentication:** Leverage blockchain's decentralized and tamper-resistant nature to enhance authentication processes

REFERENCES

1. https://youtu.be/CliXVvPixTc
2. https://youtu.be/v-N86l2a-CM
3. https://youtu.be/BG3mpdzk0Rw
4. https://youtu.be/s4Iw_n0jdBc
5. https://youtu.be/IXElBJqJAy4