A Project Report

On

Graphical Password by Image Segmentation

Submitted in partial fulfilment of the

requirement for the award of the degree of

MASTER OF COMPUTER APPLICATION

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

DEGREE

Session 2024-25 In

COMPUTER APPLICATION AND TECHNOLOGY

By

SOURABH SINHA [23SCSE2150018]

SAURABH KUMAR [23SCSE2030726]

TANISH TYAGI [23SCSE2150023]

Under the guidance of

Dr. SUDESHNA CHAKROBARTY

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

May, 2025

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"Graphical Password by Image Segmentation"** in partial fulfilment of the requirements for the award of the **MCA (Master of Computer Application)** submitted in the School of Computer Applications and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of August, 2023 to Jan and 2024, under the supervision of

Dr. SUDESHNA CHAKROBARTY, Department of Computer Science and Engineering/School of Computer Applications and Technology, Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

SOURABH SINHA [23SCSE2150018]

SAURABH KUMAR [23SCSE2030726]

TANISH TYAGI [23SCSE2150023]

This is to certify that the above statement made by the candidates is correct to the best of my

knowledge.

Dr. SUDESHNA CHAKROBARTY

CERTIFICATE

This is to certify that Project Report entitled **"Graphical Password by Image Segmentation"** which is submitted by **SOURABH SINHA [23SCSE2150018], SAURABH KUMAR [23SCSE2030726]** and **TANISH TYAGI [23SCSE2150023]** in partial fulfilment of the requirement for the award of degree MCA. In Department of School of Computer Applications and Technology, Galgotias University, Greater Noida, India is a record of the candidate own work carried out by them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Signature of Examiner(s) **Signature of Supervisor**

Date: Jan, 2025

Place: Greater Noida

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We would like to extend our gratitude to our project head Dr. Avneesh Kumar for his help. He helped us with all the technicalities and other details regarding the system.

We also appreciate the effort our project guide Dr. Sudeshna Chakraborty showed in helping me throughout the project as and when required. We were new in the field of development and was unaware of the ambiance of a web development organization. It was she who guided us to work in a fashion of meeting the target time. With her encouragement, we completed this project.

We extend my heartfelt thanks to our parents because without their help it would not be possible for me to complete this project.

Last but not least we would like to thank the almighty because this project would be impossible without his blessing.

SOURABH SINHA [23SCSE2150018]

SAURABH KUMAR [23SCSE2030726]

TANISH TYAGI [23SCSE2150023]

ABSTRACT

Traditional text-based passwords are one of the most widely used methods for securing access to systems and sensitive data. However, they are often vulnerable to attacks due to poor password choices, limited memorability, and susceptibility to brute-force, phishing, and social engineering attacks. To address these issues, **Graphical Password Systems** offer an alternative authentication method, leveraging the human brain's ability to recall visual information more effectively than text.

This project focuses on implementing a **Graphical Password by Image Segmentation system**, where users create passwords by selecting specific segments of an image. During authentication, users must select the same image segments in the correct order to gain access. The system combines the advantages of visual memory with enhanced security by increasing the password space, making it difficult for attackers to guess or crack passwords. This method also improves user experience by providing a more intuitive and memorable approach to password creation and authentication.

This abstract introduces key components of graphical password systems, explores the challenges of traditional authentication methods, and highlights the advantages and considerations of implementing graphical passwords, particularly through image segmentation. The system aims to balance security, usability, and performance, making it suitable for a wide range of applications where user engagement and robust security are paramount.

LIST OF TABLES

1. **Users Table**
    - COLUMS
        - user_id (Primary Key)
        - username (Unique)
        - password_hash (Encrypted password hash for backup authentication)
2. **Images Table**
    - COLUMNS
        - image_id (Primary Key)
        - image_name (Name or description of the image)
        - image_path (File path or URL to the image)
3. **Image Segments Table**
    - COLUMNS
        - segment_id (Primary Key)
        - image_id (Foreign Key to Images.image_id)
        - segment_coordinates (Coordinates or description of the segment region, e.g., x1, y1, x2, y2).\
4. **Document Table**
    - COLUMNS
        - document_id (Primary Key)
        - document_name (Name or description of the image)
        - document_path (File path or URL to the image)
        - uploaded_by (Foreign Key to Users.user_id

LIST OF FIGURES

LIST OF ABBREVATIONS

1. GUI: - Graphical User Interface
2. API: - Application Programming Interface
3. DBMS: - Database Management System
4. IP: - Internet Protocol
5. HTTPS: - HyperText Transfer Protocol Secure
6. SSL/TLS – Secure Socket Layer / Transport Layer Security
7. AES: - Advanced Encryption Standard
8. MFA: - Multi-Factor Authentication
9. UID - User Identification
10. PK - Primary Key
11. FK – Foreign Key
12. UI - User Interface
13. DPI- Dots Per Inch
14. DNS – Domain Name System
15. OCR - Optical Character Recognition

CHAPTER 1

INTRODUCTION

Graphical password systems represent a new approach to authentication, meant to solve the inherent flaws of traditional text-based passwords. While text passwords are the most often used means of safeguarding access to sensitive information, they suffer from major drawbacks. Users typically construct simple, readily guessable passwords, such as "password123" or "123456," which may be cracked with reasonable ease. Additionally, memorizing long and complicated passwords is cognitively difficult, forcing users to either reuse the same password across various platforms or turn to unsafe password storing methods.

To circumvent these issues, Graphical Password Systems offer an alternative authentication technique that exploits the human brain's intrinsic ability to perceive and retain visual information more successfully than text. Instead, then typing a string of characters, users interact with images to generate and authenticate their passwords. This method can require picking specific points, locations, or objects within an image or recalling patterns connected with the image, depending on the type of graphical password system.

Graphical passwords capitalize on many key features of human cognition:

Recognition over Recall: Humans are often better at recognizing visual information (such as faces, places, or objects) than memorizing arbitrary sequences of characters. This makes graphical passwords easier to remember while still delivering a high level of protection.

Visual Memory: Our visual memory is resilient and capable of storing complex patterns and details over long durations. By employing images, graphical passwords generate a more natural memory association, decreasing the cognitive load on users.

Increased Security: The visual complexity of graphics allows for an exponential rise in the number of possible password combinations, making graphical passwords more resistant to brute-force and dictionary attacks. The sheer range of potential selections within an image (such as specific points or areas) adds layers of difficulty to the authentication procedure.

Types of Graphical Password Systems

Graphical password systems can be divided into three basic categories, each giving unique techniques for password creation and authentication:

Recognition-Based Systems:

In this system, users are provided with a series of photographs during the registration process and are forced to select one or more images that constitute their "password." During authentication, users are shown a sequence of photos, and they must properly identify the ones they picked previously. An example of this is the Passfaces system, where users authenticate by recognizing pre-selected faces from a bigger set. This solution depends on users' ability to recognize familiar images, which is a more intuitive procedure than recalling text-based passwords.

Recall-Based Systems:

Recall-based graphical passwords:
In this system it requires users to reproduce a pattern or artwork as their password. The Draw-a-Secret (DAS) method, for example, allows users to draw a pattern on a 2D grid during password formation. During authentication, users must repaint the same pattern. The precision of the drawing, particularly the sequence of strokes, decides whether access is given.
This method combines components of freeform input with recall memory, where users need to accurately replicate the visual pattern.
Cued-Recall Systems:

Cued-recall systems:

mix recognition and recall by employing an image as a cue for users to remember specific locations or patterns. One prominent example is the PassPoints system, where customers click on specified spots within a single image during registration. During authentication, the same image is presented, and users must click the correct points in the correct order to authenticate successfully.
Cued-recall systems are particularly effective in exploiting both visual memory and spatial awareness, enabling a mix of security and usability.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Product Perspective

i. User Experience (UX) and Interface

- Ease of Use: The graphical password system should be intuitive. Users can select pieces of an image rather than inputting alphanumeric passwords. For example, a user may click on certain elements of an image (such corners of a photo or parts of a pattern).
- Image Selection: Allow users to choose from a set of predetermined images or submit their own. The uploaded images are split into clickable zones.

- Interactive Feedback: As users engage with the image (e.g., clicking or touching parts of the screen), provide visual feedback to show selected regions for validation.

- Accessibility: Ensure the product is functional across platforms (touchscreens, desktops) and is inclusive for users with disabilities (e.g., voice commands or expanded clickable zones for vision-impaired users).

ii. Security

- Image Segmentation: Use segmentation techniques to separate images into distinct sections. Each region functions as a password component.

- Pattern and Order: The graphical password could rely not simply on the exact parts clicked but also the order in which they're selected, offering a combination of factors for greater security.

- Dynamic Segmentation: For enhanced protection, consider randomizing the segmentation or utilizing slightly different segmentations each time the user attempts to log in. This decreases the likelihood of shoulder surfing or screen recording assaults. Two-Factor Authentication: Integrate an optional two-factor authentication (2FA) mechanism. After picking bits of the image, consumers can obtain a temporary code on their device.
- **Two-Factor Authentication**: Integrate an optional two-factor authentication (2FA) mechanism. After selecting segments of the image, users can receive a temporary code on their device.

iii. Technical Considerations

- Storage: Instead of keeping the complete image, store a hashed representation of the clicked locations or segments. This strategy boosts security by lowering the danger of password leaks if the system is compromised.

- Segmentation techniques: Leverage advanced techniques such as edge detection, region expanding, or machine learning models to construct separate and non-overlapping segments. High precision in segmentation is necessary to avoid ambiguity.

- Device Compatibility: Ensure interoperability with a wide range of input devices, from mouse clicks to touch gestures.

- Performance: Ensure the segmentation and validation processes are optimized for short response times, especially in mobile situations.

.

iv **Business Considerations**

- Target Audience: Primarily focus on people that are security-conscious, but want an easier alternative to standard passwords. This method could be particularly appealing in fields like finance, healthcare, or mobile app security.

- Monetization: Offer a free basic version with premium capabilities (such custom picture uploads, greater security, or two-factor authentication) for a subscription cost.

- Scalability: Develop the system so it can be incorporated into many platforms (mobile applications, websites, etc.), making it extremely adaptable for diverse enterprises.

v. **Privacy**

- **User Data Protection:** Ensure that no sensitive user information is stored directly on the server. If user-uploaded photos are allowed, they should be encrypted and split in a way that assures their security.

- Anonymization: Ensure that no direct identifying data can be derived from the image segmentation pattern to ensure user anonymity in case of data breaches

2.1.1 System Interfaces

Designing a system interface for a **graphical password using image segmentation** involves creating a user-friendly, secure, and intuitive environment where users can interact with images to create and authenticate their graphical passwords. Below is a breakdown of the key interface components and their functionalities**.**

1. Login and Registration Screens

a. A. *Registration Screen (Create Password):*

- **Image Selection**:
  - Users can either select a pre-defined image or upload their own.
  - UI Components:
    - **Image Gallery**: A grid of pre-defined images for users to select from.
    - **Upload Button**: An option to upload an image from the device. Supported file formats (e.g., JPG, PNG) and size limitations should be visible.
- **Image Segmentation Display**:
  - Once an image is selected or uploaded, it is divided into clickable segments. The segments should be visibly highlighted or have hover effects.
  - UI Components:
    Image Viewer: Displays the segmented image. Each segment should provide visual feedback (highlight, zoom, or color change) when hovered over or clicked.
- **Segment Selection**:
  - Users are prompted to click on a set number of segments (e.g., 4 or 5) to create a password pattern. They may also need to follow a specific order of clicks.
  - UI Components:
    - **Password Instruction**: A prompt or tooltip indicating how many segments need to be selected.
    - **Selection Tracker**: A visual indication of the selected segments, either below or beside the image. It could be represented as small numbered boxes showing the order of clicks.
    - **Reset Button**: Users can reset their selections if they make a m
- **Confirmation Step**:
  - After selecting the required segments, users are asked to re-select the same segments in the correct order to confirm their password.
  - UI Components:
    - **Confirm Selection**: A repeat image segmentation interaction where users confirm their previously selected pattern.
    - **Error Feedback**: If the confirmation doesn't match, provide an error message, highlighting the differences.

B. Login Screen (Password Authentication):

- **Image Segmentation**:
  - When logging in, users are shown the segmented image they used during registration. The image is divided into segments the same way.

- UI Components:
  - **Segmented Image Display**: Users see the segmented image and click on the previously selected segments in the correct order.

- **Authentication Process**:
  - As users click on segments, feedback is given, such as highlighting or numbering the segments as they're selected.
  - UI Components:
    - **Selection Tracker**: Numbered boxes or visual indicators that track the selected segments and their order.
    - **Reset Button**: Users can reset their selection if they make a mistake.
- **Login Feedback**:
  - If the selected segments match the stored password, the user is logged in. If not, an error message is displayed with options to retry or reset the password.
  - UI Components:
    - **Error Message**: Clear feedback if authentication fails, providing hints (e.g., "Try again" or "Incorrect selection").

2. System Interface Components

A. **Image Segmentation Algorithm Interface**:

- The backend system needs to dynamically segment images either through predefined zones or using an algorithm to create consistent and distinct zones.

- UI Component:
  - **Backend Logic**: Image processing algorithms generate the segments and provide coordinates for click detection.

B. User Management:

- **Profile Section**: A section where users can manage their graphical password, view their account details, and change their password by selecting a new image or pattern.

  - UI Components:
    - **Change Password Option**: Users can go through the password creation process again by selecting a new image or new segments on the current image.

    - **Password Recovery**: If users forget their graphical password, the system can offer recovery options like email verification, 2FA, or a fallback to traditional password input.

C. Admin/Developer Interface (Optional):

- **Analytics**: Track how often users reset or forget their graphical passwords. Provide statistics on login attempts and segment selection trends.

- **Customization Options**: Enable admins to configure the difficulty of image segmentation (e.g., number of segments, randomizing segment positions).

3. Security Features

a. Dynamic Image Segmentation:

- For security, the interface should support **dynamic segmentation** where segment locations or shapes may change slightly on each login attempt to prevent shoulder surfing attacks.

b. Click Timing Analysis:

- **Additional Security Layer**: The system may track the timing between clicks on different segments and use this as an additional authentication factor, making it harder for attackers to replicate the exact sequence.

c. Multi-factor Authentication (MFA):

- After selecting the correct image segments, the system could request a second layer of authentication, like a one-time password (OTP) or biometric verification (e.g., fingerprint or face recognition).
- UI Components:
    - **OTP Input Field**: A simple input for entering the code sent to a registered device.
    - **Biometric Prompt**: A biometric login prompt on compatible devices.

4. Usability Considerations

a. Accessibility:

- Ensure the interface is accessible by:
    - Providing **keyboard navigation** for non-mouse users.
    - Implementing **screen reader** support for users with visual impairments.
    - Using **large clickable segments** to accommodate users with motor impairments.

b. Device Adaptability:

- The interface should responsive and work smoothly across different devices:
    - **Mobile**: On touch devices, segments should be large enough for users to easily tap with fingers.
    - **Desktop**: On desktops, segments can be smaller, with hover or mouseover effects to assist in segment identification.

- **Tablet**: Provide a middle-ground interface that's equally intuitive for both touch and mouse input.

5. Feedback and Error Handling

a. Success Feedback:

- Upon successful authentication, show a brief confirmation message (e.g., "Login Successful") before redirecting to the desired application or dashboard.

b. Error Feedback:

- For failed login attempts, display clear error messages (e.g., "Incorrect Pattern, Please Try Again").
- Implement rate limiting or CAPTCHA mechanisms after multiple failed attempts to prevent brute force attack

Example Workflow:

1. **Registration**:
   - Select or upload an image.
   - Segment the image automatically (visibly shown).
   - User selects 4-5 segments in a specific order.
   - Confirm the segment selections by repeating the pattern.
   - Success message confirms the password is set.
2. **Login**:
   - Present the same segmented image.
   - User selects the correct segments in the correct order.
   - System authenticates the selections and provides success or failure feedback.

2.1.2 Hardware Interfaces

For a **graphical password system using image segmentation**, the **hardware interface** plays a significant role in ensuring smooth interaction between the user and the system across various devices. The hardware interface will involve input, output, and processing components that allow users to effectively interact with the segmented images to create and authenticate their passwords. Here's how the hardware interface can be designed.

1. Input Devices:

- **Touchscreens** (smartphones, tablets, touchscreen monitors
    - **Primary Interaction**: Users will interact with the image through touch gestures.
    - **Touch Precision**: Ensure that touch sensitivity is accurate and that each segment is easily selectable, even on smaller screens.
    - **Multi-Touch Support**: In cases where users need to select multiple segments at once or perform pinch/zoom actions, multi-touch support should be available.
    - **Haptic Feedback**: For enhanced user experience, enable haptic feedback (vibration) when users select or click on a segment. This provides tactile confirmation of their actions.

- **Mouse & Keyboard** (desktops, laptops): Mouse clicks or keyboard navigation for segment selection.
    - **Mouse Clicks**: Users interact with the segmented image by clicking on the regions with a mouse or trackpad.
    - **Hover Effects**: Enable hover effects for mouse devices so that users can see which segment they are about to click. This increases accuracy.
    - **Keyboard Navigation**: For accessibility, users should be able to navigate between image segments using the keyboard (e.g., arrow keys) and select a segment using the Enter or Space key.
- **Stylus/Pen**: Precise selection on supported devices.
    - **Stylus Interaction**: For devices like tablets or specialized touchscreen monitors, users may use a stylus for more precise segment selection.
    - **Pressure Sensitivity**: Although pressure sensitivity may not be essential for graphical passwords, supporting stylus pressure can add an additional layer of interaction (e.g., drawing a line between segments).

- **Biometrics** (fingerprint, face recognition): For multi-factor authentication (MFA).
    - As part of two-factor authentication (2FA), hardware like fingerprint scanners or facial recognition could be used after the graphical password input to add a second layer of security.

2. Output Devices:

- **Displays**: High-resolution touch or non-touch displays to clearly show image segments.
    - **High-Resolution Display**: Graphical password systems rely on visual clarity, so the display must

have a resolution high enough to distinguish between segments.

- **Touchscreen Displays**: Devices with integrated touchscreens need to support responsive and accurate touch interaction, as well as visual feedback when segments are selected.
- **Large Displays**: If used in environments like kiosks or ATMs, large displays should be used to allow users to see and interact with image segments comfortably.

- **Feedback**: Visual, auditory, and haptic feedback (e.g., vibration, sound) to confirm selections.

3. Processing Hardware:

a. Image Processing and Segmentation

- **CPU/GPU**: For fast image segmentation and password validation.
  - These components handle the computational load of image segmentation, where the image is divided into regions. The system must efficiently handle this process, especially if dynamic segmentation or image randomization is employed.

- **Embedded Image Processing Hardware**: For devices with built-in image recognition capabilities, such as smartphones or IoT devices, dedicated image processing units (IPUs) can be utilized to speed up segmentation tasks.

b. Memory and Storage

- **Secure Password Storage:** The system should store encrypted representations of the graphical password (e.g., hashed coordinates or segments).

- **Secure Enclave:** On devices like iPhones or modern Android devices, a secure enclave can be used to store and encrypt graphical password data, increasing security against attacks.

c. Network Connectivity

- **Real-Time Processing**: For systems that store and validate passwords on a server, high-speed network connections are necessary to ensure that image and segment data is transmitted securely and quickly**.**

- **Offline Mode:** In certain use cases (e.g., mobile apps or remote areas), the system should be able to function in offline mode, where the graphical password is validated locally and synced later.

4. Specialized Hardware for Security

a. **Hardware Security Modules (HSM)**: Securely manage cryptographic keys.
   - HSMs can be used to securely generate, store, and manage cryptographic keys for the graphical password system. These keys could protect the graphical password data, such as the image coordinates or segment patterns.

b. Trusted Platform Module (TPM)
   - TPM chips are used in many computers to enhance security by providing hardware-level encryption for passwords, and could be employed to store graphical password patterns securely.

5. **Multi-Factor Authentication (MFA) Hardware Integration:**

- **Fingerprint Sensors**: Integrated fingerprint sensors can provide an additional layer of security when verifying the graphical password.
- **Face Recognition Cameras:** Devices like USB security keys (e.g., YubiKey) can be used as part of two-factor authentication, complementing the graphical password system.

Summary of Hardware Interface

The **hardware interface** for a graphical password system using image segmentation involves various components that ensure secure, fast, and seamless interaction between the user and the system. The main input devices include touchscreens, mouse and keyboard, and stylus, while output devices like high-resolution displays and haptic feedback systems provide visual, auditory, or tactile feedback. The processing hardware, including CPU, GPU, and storage units, is responsible for image segmentation and secure password handling. Additionally, security devices like fingerprint scanners, face recognition systems, and HSMs can be integrated for added security.

2.1.3 Software Interfaces

A **software interface** for a **graphical password system using image segmentation** includes various components to facilitate user interaction, secure password management, and seamless authentication. Here's a brief overview:

1. User Interface (UI)

a. Registration (Create Graphical Password)

- **Image Selection**: Users can choose or upload an image.
- **Image Segmentation Display**: The selected image is divided into segments, clearly highlighted for easy selection.

- **Segment Selection**: Users click/tap on segments in a specific order to create their password.
- **Confirmation**: Users are asked to repeat the selection process to confirm their password.
- **Feedback**: Visual feedback like color changes or borders when segments are selected.

b. Login (Password Authentication)

- **Image Segmentation**: The same segmented image is shown for users to input their graphical password by selecting the correct segments in the right order.
- **Visual Feedback**: The selected segments are highlighted as the user clicks them.
- **Error Handling**: If the pattern is incorrect, display an error message with options to retry.

2. Backend Software

a. Image Segmentation Algorithm

- **Dynamic Image Processing**: The backend uses algorithms (like edge detection or machine learning) to divide the image into distinct, non-overlapping regions.
- **Randomized Segmentation**: Option to slightly adjust segmentation for added security against replay attacks.

b. Secure Password Storage

- **Hashing and Encryption**: The selected segments are hashed and stored securely, rather than storing the raw image or coordinates.
- **Database**: A secure database to store encrypted password data.

3. Security Features

- **Multi-Factor Authentication (MFA)**: Option to integrate additional security layers like OTP, fingerprint, or facial recognition.

- **Brute Force Protection**: Rate-limiting, CAPTCHA, or lockout mechanisms after multiple failed attempts.

4. Accessibility

- **Keyboard Navigation**: Support for non-mouse users to navigate between segments.
- **Screen Reader Compatibility**: Interface design ensures compatibility with screen readers for visually impaired users.

5. Device Compatibility

- **Responsive Design**: The software adapts to various devices (desktops, tablets, smartphones) ensuring smooth interactions across all platforms.

2.1.4 Communications Interfaces

A **communication interface** for a **graphical password system using image segmentation** defines how data is transmitted between the user interface, backend system, and external services. It ensures secure, efficient, and reliable communication for tasks like password creation, authentication, and security checks.

1. Client-Server Communication

a. User Requests

- **Registration Request**: When users create a graphical password, the client sends the selected image segments (hashed or encrypted) to the server.

- **Login Request**: During login, the client sends the user's selected segments to the server for validation against stored password data.

b. Server Responses

- **Password Confirmation**: The server verifies the graphical password and sends a success or failure response to the client.
- **Error Handling**: If the password is incorrect or there's an issue (e.g., multiple failed attempts), the server sends back appropriate error messages or lockout responses.

2. Protocols

- **HTTPS (Secure HTTP)**: All communications between the client and server are encrypted using HTTPS to prevent eavesdropping or man-in-the-middle attacks.
- **WebSocket** (optional for real-time feedback): If real-time interaction is required (e.g., feedback while selecting segments), WebSocket can be used for continuous, low-latency communication.

3. Data Encryption

- **End-to-End Encryption**: All sensitive data (image segments, hashed passwords) is encrypted both in transit (TLS/SSL) and at rest.
- **Encryption Algorithms**: Common encryption algorithms (e.g., AES for symmetric encryption, RSA for public-key encryption) are used for securing graphical password data.

4. API Layer

- **RESTful API**: The backend communicates with the client via a RESTful API, offering endpoints like `/register`, `/login`, `/validate`, etc.
- **JSON Data Format**: Typically, JSON (JavaScript Object Notation) is used for data exchange between client and server, ensuring lightweight and easy-to-parse data.

5. **Authentication and Security.**

a. Token-Based Authentication

- **Session Tokens**: Upon successful login, the server issues a session token (JWT or OAuth token) to maintain user sessions securely across multiple requests.
- **Multi-Factor Authentication (MFA)**: If MFA is enabled, the server might also request an additional authentication factor (e.g., OTP, biometric data) as part of the communication process.

b. Brute Force Prevention

- **Rate Limiting**: The communication interface monitors the number of authentication attempts from a single IP to prevent brute force attacks.
- **CAPTCHA**: After several failed login attempts, the communication interface can request CAPTCHA verification to ensure the user is legitimate.

2.1.5 Memory Constraints

When implementing a **graphical password system using image segmentation**, memory constraints are an important consideration, particularly regarding the storage and processing of images, password data, and security measures. Here's a brief breakdown of memory constraints and how they can be managed.

1. Image Storage and Segmentation Data

a. Image Storage:

- **Large Image Files**: Storing large, high-resolution images may consume significant memory, especially if the system needs to store multiple images.
- **Memory Optimization**:
  - Use image compression techniques (e.g., JPEG, PNG) to reduce file sizes without sacrificing too much quality.
  - Store images in external cloud storage and retrieve them when needed, instead of saving all images locally.
- **Consideration**: If images are stored locally, low-memory devices (e.g., mobile phones) may struggle with large images.

b. Image Segmentation Data:

- **Segment Data**: Each image is divided into several segments, and the system needs to store information about the coordinates of the segments or their patterns.

- **Memory Usage**:
  - Store only the essential data (segment coordinates or hashed segment selections) rather than the entire segmented image.

  - Use efficient data structures to store segment information, such as arrays or dictionaries, to minimize memory footprint.

2. Hashed Password Data

- **Encrypted/Hashed Password Storage**:
  - The system stores the hashed representation of the graphical password rather than the raw image segments. This reduces memory usage as the hash (e.g., SHA-256) is much smaller compared to storing images or coordinates.

  - **Consideration**: Hashes and encrypted data take up significantly less space but must be stored securely (in a database or secure enclave).

- **Optimization**: Only store essential password metadata (e.g., image identifier, hash, salt) rather than the entire image or segment details.

3. Memory Usage on Client Side

a. Mobile Devices:

- **Limited Memory**: Mobile devices typically have more limited memory compared to desktop systems.

- **Optimization Strategies**:
  - Limit image resolution or use compressed versions of images to reduce memory usage.
  - Ensure the graphical password system is lightweight and does not consume excessive RAM when displaying or processing segmented images.

b. Desktop Devices:

- **Higher Capacity**: Desktops and laptops generally have more memory, but the system should still optimize image processing and storage to avoid unnecessary memory .

4. Caching Mechanisms

- **Client-Side Caching**: Caching images and segment data on the client-side (temporarily storing in memory) can improve performance, but this must be done carefully to avoid excess memory usage.

- **Server-Side Caching**: Efficient caching mechanisms on the server can reduce the need to repeatedly process images, minimizing memory load.

5. Session Data

- **Temporary Data**: While a user is interacting with the system (e.g., selecting segments), the system holds temporary session data in memory.

- **Memory Constraint**: Session data should be cleared or minimized after each interaction to free up memory.

2.1.7 Site Adaptation Requirements

Site Adaptation Requirements involve the specific modifications and considerations needed to deploy a system, such as a graphical password system, in various physical, technical, and environmental contexts. These requirements ensure that the system functions optimally across different platforms, locations, and user scenarios.

Here are the key site adaptation requirements**:**

1. Device and platform Compatibility

a. Screen Size and Resolution

- Adapt the user interface to different screen sizes (smartphones, tablets, desktops).
- Implement **responsive design** for optimal display on small and large screens.
- Ensure high-resolution images for devices with high DPI (e.g., Retina displays).

b. Input Methods

- Support various input methods, including:
    - **Touchscreens** for mobile and tablet devices.
    - **Mouse and keyboard** for desktop computers.

- **Stylus/Pen** for touch-enabled devices with precise inputs.

c. Operating System and Browser Support

- Ensure compatibility with major operating systems (e.g., **Windows**, **macOS**, **iOS**, **Android**).
- Support for common web browsers (Chrome, Firefox, Safari, Edge) with cross-browser

2. Environmental Conditions

a. Physical Location

- **Public environments**: Adapt the system for kiosks or ATMs by ensuring easy-to-use touch interfaces with clear instructions.
- **Private environments**: Simplify the user interface for home or personal use.

b. Light Conditions

- Adapt the system for use in varying lighting environments, such as low-light (night mode) or bright-light scenarios, by implementing **adaptive brightness** or a **dark mode**.

c. Network Conditions

- Design the system to function well in both high-speed and low-speed internet conditions.
- Implement an **offline mode** where users can still interact with the system and sync data later when a connection is available.

3. User Demographics and Experience

a. Age and Technical Expertise

- For users with different technical skill levels, provide a simplified or advanced mode.
- Ensure the interface is user-friendly, especially for older adults or children, by offering larger icons or easy-to-understand instructions.

b. Cultural Adaptation

- Support **localization** by translating the interface into multiple languages.
- Adapt images, color schemes, and content based on regional cultural norms and preferences.

c. Accessibility

- Ensure the system meets **accessibility standards** (e.g., WCAG) by providing features like screen reader support, keyboard navigation, and high-contrast display options.
- Design touch areas to be large enough for users with limited motor control and make all visual elements readable for colorblind users.

4. Security and Regulatory Compilance

a. Local Data Security Regulations

- Adapt the system to comply with local and international regulations (e.g., **GDPR**, **CCPA**) regarding data privacy and user consent.
- Use **data encryption** (in transit and at rest) to protect user data and graphical password information.

b. Authentication Requirements

- Support **multi-factor authentication** (MFA) in environments requiring higher security (e.g., banks or healthcare systems).
- Ensure secure communication protocols, such as **HTTPS**, are in place to prevent man-in-the-middle attacks.

c. User Activity Logging

- Implement secure logging and auditing features for environments that require user activity tracking (e.g., enterprise systems).

5. **Performance and Scalability**

a. Low Resource Devices

- Optimize the system for devices with limited resources, such as older smartphones or tablets, by reducing memory usage, processing requirements, and data load.

b. High Traffic Environments

- Ensure the system can scale to support many users simultaneously, especially in enterprise or public deployments (e.g., at airports, hospitals).
- Use **cloud infrastructure** to dynamically scale and handle large volumes of data and user activity.

2.2 Product Function

The **product functions** of a **graphical password system using image segmentation** refer to the core capabilities and operations the system must perform to fulfill its intended purpose. Below are the primary functions of such a system:

1. Graphical Password Creation

- **Image Selection**: The system allows users to choose or upload an image that will be used for the password creation process.
- **Image Segmentation**: The system automatically segments the selected image into predefined regions (e.g., grids) or allows the user to define custom segments.
- **Password Selection**: Users create a password by selecting a sequence of segments from the image. This sequence serves as the graphical password.
- **Save and Store Password**: The system securely stores the selected image segments as the user's password, typically by hashing or encrypting the segment data.

2. Graphical Password Authentication

- **Login Interface**: The system provides a user interface for selecting image segments as part of the login process.
- **Image and Segment Display**: The user is presented with the same image (or a set of images) used during password creation. The system highlights or allows the user to select the same segments for authentication.

- **Password Verification**: The system compares the selected segments with the stored password. If the segments match, access is granted; otherwise, access is denied.
- **Feedback Mechanism**: The system provides real-time feedback (e.g., "incorrect password" or "login successful") during the login process.

3. Password Security Management

- **Hashing and Encryption**: The system hashes or encrypts the graphical password (image segments) for secure storage.
- **Salt Generation**: Adds random salt to password hashes to make them unique and protect against rainbow table attacks.
- **Multi-Factor Authentication (MFA)**: Optionally, the system supports additional layers of security, such as using a graphical password along with a one-time password (OTP), biometric authentication, or SMS verification.
- **Brute Force Prevention**: Implements measures to limit repeated login attempts (e.g., account lockout or CAPTCHA after a number of failed attempts).

4. Image and Segment Management

- **Image Customization**: Users can upload their own images or select from a predefined library. • **Segmentation Flexibility**: The system can either automatically segment images into grids or

allow users to define custom segment patterns based on user preferences.

Dynamic Scaling: The system adapts the size and resolution of the image and segments based on the user's device (e.g., scaling for mobile or desktop devices).

5. Password Recovery and Reset

- **Forgot Password Workflow**: The system provides a mechanism for users to recover or reset their password if they forget it, typically by verifying identity through alternate means (e.g., email verification, OTP, security questions).
- **Password Reset**: Users can select a new image and create a new password if needed, with the previous password securely deleted from the system.
- **Notification System**: The system notifies the user (via email, SMS) when a password reset has occurred for security purposes.

2.3 User Characteristics

User characteristics refer to the attributes, needs, behaviors, and demographics of the users interacting with a **graphical password system using image segmentation**. Understanding these characteristics helps in designing a system that caters to a diverse user base while ensuring security and usability. Here are the key user characteristics to consider:

1. Demographic Characteristics

a. Age Group

- **Young Adults (18-35)**: Generally tech-savvy, familiar with various digital authentication methods, and open to trying new password mechanisms.
- **Middle-Aged Adults (35-55)**: Likely to prioritize both ease of use and security, often with experience in traditional password systems.
- **Elderly Users (55+)**: May prefer simpler, more intuitive systems with clear guidance and larger visual elements for easy interaction.

b. Education and Technical Proficiency

- **Tech-Savvy Users**: Users who are familiar with advanced technologies, likely to prefer more complex graphical password options for higher security.
- **Non-Technical Users**: Users with limited technology exposure may need a simplified process and clear instructions for creating and using graphical passwords

2. Behavioural Characteristics

a. Security Consciousness

- **Security-Focused Users**: Prefer highly secure password systems, willing to invest time in creating complex graphical passwords and utilizing multi-factor authentication.
- **Convenience-Oriented Users**: Prefer simple and fast password creation processes, valuing ease of use over high-level security, especially in low-risk environments.

b. Familiarity with Graphical Passwords

- **Experienced Users**: Users familiar with graphical passwords from previous systems (e.g., Android's pattern lock) may adapt quickly to image segmentation.
- **First-Time Users**: Require additional guidance and tutorials on how to set up and use graphical passwords effectively.

c. Interaction Preferences

- **Visual Learners**: Likely to engage well with graphical systems, preferring to remember images and patterns over traditional text-based passwords.
- **Auditory/Kinaesthetic Learners**: May require additional cues or instructions, such as audio feedback or clear touch interactions, to improve their experience with graphical passwords.

3. Physical and Cognitive Abilities

a. Motor Skills

- **Fine Motor Control**: Users with precise motor skills can interact more easily with systems that require selecting small image segments.
- **Impaired Motor Control**: For users with limited motor abilities, the system should offer larger, more easily selectable segments or assistive technology support.

b. Cognitive Abilities

- **Memory Retention**: Some users may find it easier to remember graphical passwords compared to alphanumeric ones, while others may need simple images or patterns that are easy to recall.
- **Users with Cognitive Impairments**: The system should provide options for simpler image segmentation, larger grids, or alternative password methods to accommodate users with memory or learning challenges.

c. Visual Impairments

- **Color Blindness**: Ensure that image segmentation does not rely solely on color differences and offers contrast or texture-based options.

2.4 Constraints

The **constraints** for a graphical password system using image segmentation involve technical, security, usability, and operational factors that limit or influence the system's design and functionality. These constraints must be carefully considered to balance security, performance, and user experience.

1. Technical Constraints

a. Device Compatibility

- **Screen Size**: The system must work on a wide range of devices, from small mobile screens to large desktop monitors. Smaller screens may limit the complexity of image segmentation, making it harder to select precise segments.

- **Touch vs. Mouse Input**: Different input methods (touch on mobile devices, mouse on desktop) may impact how easily users can interact with image segments. This may require optimizing interfaces for each device type.
- **Performance Limitations**: Lower-end devices with limited CPU and memory may struggle with real-time image segmentation or high-resolution images, requiring optimizations in image processing and system performance.

b. Storage and Memory Constraints

- **Password Storage**: Storing graphical passwords (image segments) securely can require more space than text-based passwords, especially if the images and segmentation data need to be encrypted.
- **Memory Usage**: The system must be efficient in terms of memory consumption, particularly on mobile devices or environments with low memory availability. Large images or complex segmentation can increase memory usage.
- **Offline Usage**: In some cases, the system may need to function offline or in areas with intermittent connectivity. Handling image storage and verification without constant server access poses a challenge.

c. Network and Bandwidth

- **Data Transmission**: High-resolution images and complex data related to graphical passwords can increase the amount of data that needs to be transferred over the network. This may be a constraint in low-bandwidth or high-latency environments.
- **Cloud Dependency**: If the system is reliant on cloud infrastructure for storing images or processing passwords, it may be constrained by network connectivity, affecting users in regions with unreliable or slow internet access.

2. Security Constraints

a. Brute Force and Dictionary Attacks

- **Password Complexity**: Graphical passwords must be sufficiently complex to avoid brute force attacks, but increasing complexity can also reduce usability. The system must balance between security (e.g., many image segments) and ease of remembering the password.
- **Segmentation Limits**: If the number of image segments is too low, the password space becomes smaller, making it easier to guess. However, increasing the number of segments may complicate the user experience.

b. Encryption and Storage

- **Password Hashing**: Graphical passwords are often stored in hashed form, but hashing graphical passwords (e.g., coordinates of image segments) can be more complex than traditional text-based hashing.
- **Data Protection**: The system must securely store the image and segmentation data without exposing sensitive information. Compliance with encryption standards and regulations like GDPR can impose constraints on storage and handling.

c. Phishing and Shoulder Surfing

- **Visual Exposure**: Unlike text-based passwords, graphical passwords are susceptible to **shoulder surfing** (observation by nearby individuals). The system must include anti-shoulder surfing measures, such as randomized grids or obfuscation techniques, which may complicate the design.
- **Phishing Resistance**: Users may be tricked into entering their graphical password on malicious interfaces (phishing attacks). Preventing these attacks requires secure and recognizable user interfaces, adding complexity to the system.

d. Session Management

- **Timeouts**: The system must automatically log out users after a period of inactivity to prevent unauthorized access. However, if timeouts are too aggressive, it can frustrate users.
- **Multi-Factor Authentication (MFA)**: Users may need additional security layers, such as MFA, but integrating these securely without affecting the user experience can be challenging.

3. Usability Constraints

a. Memory Load

- **Password Recall**: Users may struggle to remember image segments, especially if the password creation process involves selecting multiple, complex segments. A balance between memorable and secure segmentation must be found.
- **Complexity vs. Usability**: The more segments or grids an image contains, the more secure the system becomes, but this increases the difficulty of remembering and selecting the correct segments during login. Simpler segmentations may reduce security.

b. Learning Curve

- **User Familiarity**: Users unfamiliar with graphical passwords may require more time to learn the system. Providing tutorials or onboarding instructions can mitigate this but adds complexity to the user experience.

- **Error Tolerance**: Users may accidentally select the wrong segments during login, especially on small screens. Designing the system to account for minor errors (e.g., allowing close but not exact matches) may impact the level of security.

c. Accessibility

- **Visual Impairments**: Users with visual impairments (e.g., color blindness, low vision) may find it difficult to interact with graphical passwords, especially if segment differentiation relies on color. Accessibility requirements limit design choices, such as segment size, contrast, and color schemes.
- **Motor Impairments**: Users with limited motor control may struggle to select small or precise segments, necessitating larger targets or alternative input methods, which could reduce password complexity and security.

4. SystemConstraints

a. Scalability

- **User Volume**: For systems with large user bases (e.g., enterprise environments), the system must be capable of handling many users simultaneously. This requires scalability in both the storage of graphical passwords and the performance of authentication processes.
- **Real-Time Processing**: Image segmentation and password verification must occur in real-time, which can strain system resources, especially in high-traffic environments.

b. Backup and Recovery

- **Password Recovery**: Unlike text-based passwords, recovering a graphical password can be more challenging since it involves image segments. The system must provide secure and user-friendly recovery mechanisms that don't compromise security (e.g., using recovery questions, alternative login methods).
- **Data Backup**: The system must regularly back up user data, including graphical password data, without increasing the risk of exposing sensitive information.

c. Cross-Platform Functionality

- **Platform-Specific Constraints**: The system must work across multiple platforms (mobile, desktop, kiosks) without compromising user experience or security. Some features may work well on certain platforms (e.g., touch on mobile) but not on others (e.g., mouse input on desktops), requiring different design approaches

5. Regulatory and Compliance Constraints

a. Data Privacy Laws

- **GDPR and CCPA Compliance**: The system must comply with data protection regulations, such as GDPR (Europe) and CCPA (California), which impose strict rules on how user data (including graphical passwords) is stored, processed, and deleted.
- **User Consent**: Collecting and processing user data, including images used in graphical passwords, may require explicit user consent, which complicates the user registration and login process.

b. Security Standards

- **ISO/IEC 27001 Compliance**: The system may need to adhere to industry security standards, such as ISO/IEC 27001, which governs information security management systems. Compliance may impose constraints on the design of the authentication system, particularly in terms of data encryption, access control, and auditing

2.5 Assumptions and Dependencies

When designing a **graphical password system using image segmentation**, several **assumptions** and **dependencies** can shape the system's architecture, functionality, and user experience. Understanding these aspects is crucial for effective implementation and deployment. Below are the key assumptions and dependencies to consider:

Assumptions

1. **User Familiarity with Technology**
    - It is assumed that users have a basic level of comfort with technology and can navigate digital interfaces. The system may not account for users with very limited technological experience without additional guidance.
2. **Visual Recognition Capabilities**

- the system assumes users can recognize and differentiate between images and their segments, which is fundamental to the graphical password concept.

3. **Device Availability**
    - It is assumed that users will access the system using devices capable of displaying images and allowing for interaction (e.g., smartphones, tablets, computers).
4. **Security Awareness**
    - Users are expected to have a general understanding of security practices and the importance of strong passwords, thus influencing their willingness to adopt a graphical password system.
5. **Network Connectivity**
    - It is assumed that users will have access to stable internet connectivity.

Dependencies

1. **Image Processing Libraries**
    - The system depends on robust image processing libraries for efficient segmentation and rendering of images. This includes libraries capable of handling various image formats and resolutions.
2. **User Interface Design**
    - A well-designed user interface is crucial for facilitating user interaction with the graphical password system. The system relies on UI frameworks that support dynamic rendering and user-friendly navigation.
3. **Data Storage Solutions**
    - The system depends on secure data storage solutions (e.g., databases) for storing user data, including graphical passwords, encrypted images, and user profiles. These solutions must adhere to security standards.

4. **Security Protocols**
    - The implementation of security protocols (e.g., encryption, hashing algorithms) is essential for protecting user data and graphical passwords, ensuring that sensitive information is not exposed.
5. **User Authentication Frameworks**
    - The system may depend on existing user authentication frameworks or services (e.g., OAuth, SAML) for handling user registration, login, and password recovery processes securely.

2.6 Apportioning of Requirements.

Apportioning of requirements refers to the process of breaking down the overall requirements of a system into more manageable components or categories. This helps in prioritizing, organizing, and assigning responsibilities for development and implementation. In the context of a **graphical password system using image segmentation**, the requirements can be apportioned into several key categories:

1. Functional Requirements

Functional requirements specify what the system should do, outlining the expected behavior and features.

a. User Authentication

- Users should be able to create a graphical password by selecting segments from a chosen image.
- The system must verify user credentials upon login by comparing the selected segments to the stored graphical password.
- Users should have the ability to reset their passwords through a secure recovery process

b. Image Segmentation

- The system should provide a mechanism for segmenting images into selectable regions.
- Users must be able to select their preferred images from a predefined library or upload their own.
- The segmentation process should be dynamic, allowing for different grid sizes or patterns based on user preference.

c. User Interface

- The interface must provide a user-friendly experience for selecting images and segments. • The system should include clear instructions and visual cues to guide users through the
  password creation and login process.
- Feedback mechanisms (e.g., error messages, success notifications) should be implemented for better user interaction.

2. Non-Functional Requirements

Non-functional requirements define the quality attributes and constraints of the system.

a. Performance

- The system should process image segmentation and password verification in real-time, with a target response time of less than 2 seconds for user actions.
- The application must handle multiple concurrent users efficiently, ensuring no significant degradation in performance.

b. Security

- Passwords must be securely stored using strong encryption and hashing algorithms.
- The system should implement measures to prevent brute-force attacks and unauthorized access.
- User data must comply with applicable privacy regulations (e.g., GDPR, CCPA) regarding storage and handling.

c. Usability

- The graphical password system should be intuitive and easy to use, with a low learning curve for first-time users.
- The design must accommodate various user abilities, including accessibility features for those with disabilities.

3. Technical Requirements

Technical requirements outline the technology stack and system architecture needed to support the functional and non-functional requirements.

a. Platform Compatibility

- The system must be compatible with major web browsers and mobile operating systems (iOS, Android).
- The application should be responsive, adapting to various screen sizes and input methods (touch, mouse).

b. Image Processing

- The system should utilize efficient image processing libraries for segmentation and rendering, supporting various image formats (JPEG, PNG, etc.).
- The image processing operations should be optimized for performance to minimize resource consumption.

c. Data Storage

- The application must use a secure database to store user profiles, images, and hashed passwords.
- Data should be backed up regularly to prevent loss and ensure data integrity.

4. Regulatory and Compliance Requirements

These requirements ensure that the system adheres to legal and ethical standards.

a. Data Protection Regulations

- The system must comply with data protection regulations regarding the collection, storage, and processing of user data.
- User consent must be obtained before collecting personal information or images.

b. Accessibility Standards

- The system should conform to accessibility standards such as WCAG to ensure usability for individuals with disabilities.

5. User Requirements

User requirements capture the needs and expectations of the end users

a. User Preferences

- Users should have the ability to customize their graphical passwords by selecting images and segment sizes.
- The system must allow users to change their passwords easily, including options for password recovery.

b. Support and Documentation

- The system should provide comprehensive documentation, tutorials, and help resources to assist users in understanding the password creation process.
- A support mechanism (e.g.contact support) should be available for user assistance

.

2.7. Use case

This use case outlines the process of user authentication via a graphical password system that uses image segmentation. It includes the primary flow of actions, alternate flows for various scenarios, and conditions for successful authentication. This structured approach helps in understanding user interactions with the system and guides the design and development of the authentication mechanism.

2.7.1. Use case Model

Use Case: User Authentication via Graphical Password

Use Case ID: UC-001

Title: User Authentication Using Graphical Password Actors:

- **Primary Actor**: User
- **Secondary Actor**: System (Graphical Password System)

Preconditions:

- The user must have an existing account in the graphical password system.
- The user has previously created a graphical password by selecting segments from an image. • The system must be operational and accessible via the internet or the local network.

Postconditions:

- The user is successfully authenticated and granted access to their account, or they receive an error message if authentication fails.

constrain Flow:

1. **User Initiates Login**:
    - The user navigates to the login page of the graphical password system.
    - The system displays the login interface with options to enter their graphical password.
2. **User Selects Image**:
    - The system presents the user with a set of images from which they can choose (either from a predefined library or an uploaded image).
    - The user selects their designated image used for the graphical password. 3. **User Selects Segments**:
    - The system displays the chosen image divided into segments or a grid layout.
    - The user interacts with the image by selecting the specific segments they previously selected during the password creation process.
4. **System Processes Input**:
    - The system captures the segments selected by the user and processes the input.

- The system checks the selected segments against the stored graphical password for that user.

5. **Authentication Result**:
   - If the selected segments match the stored graphical password:
     - The system grants access to the user and redirects them to their account dashboard.
   - If the selected segments do not match:
     - The system displays an error message indicating failed authentication and provides options to retry or recover the password.

Alternate Flows:

- **AF-1: Forgot Password**
  1. If the user clicks on the "Forgot Password" link:
     - The system prompts the user to verify their identity (e.g., via email).
     - The user follows the recovery process to reset their graphical
- **AF-2: Session Timeout**
  1. If the user does not interact with the login screen for a predefined period:
     - The system automatically logs them out and displays a session timeout message.
     - The user must restart the login
- **AF-3: Invalid Image Selection**
  1. If the user selects an image that is not valid (e.g., an unsupported format):
     - The system displays an error message and prompts the user to select a valid image.

Extensions:

- **E-1: System Maintenance**
  - If the system is undergoing maintenance or is temporarily unavailable

2.7.2 Use Case Diagram

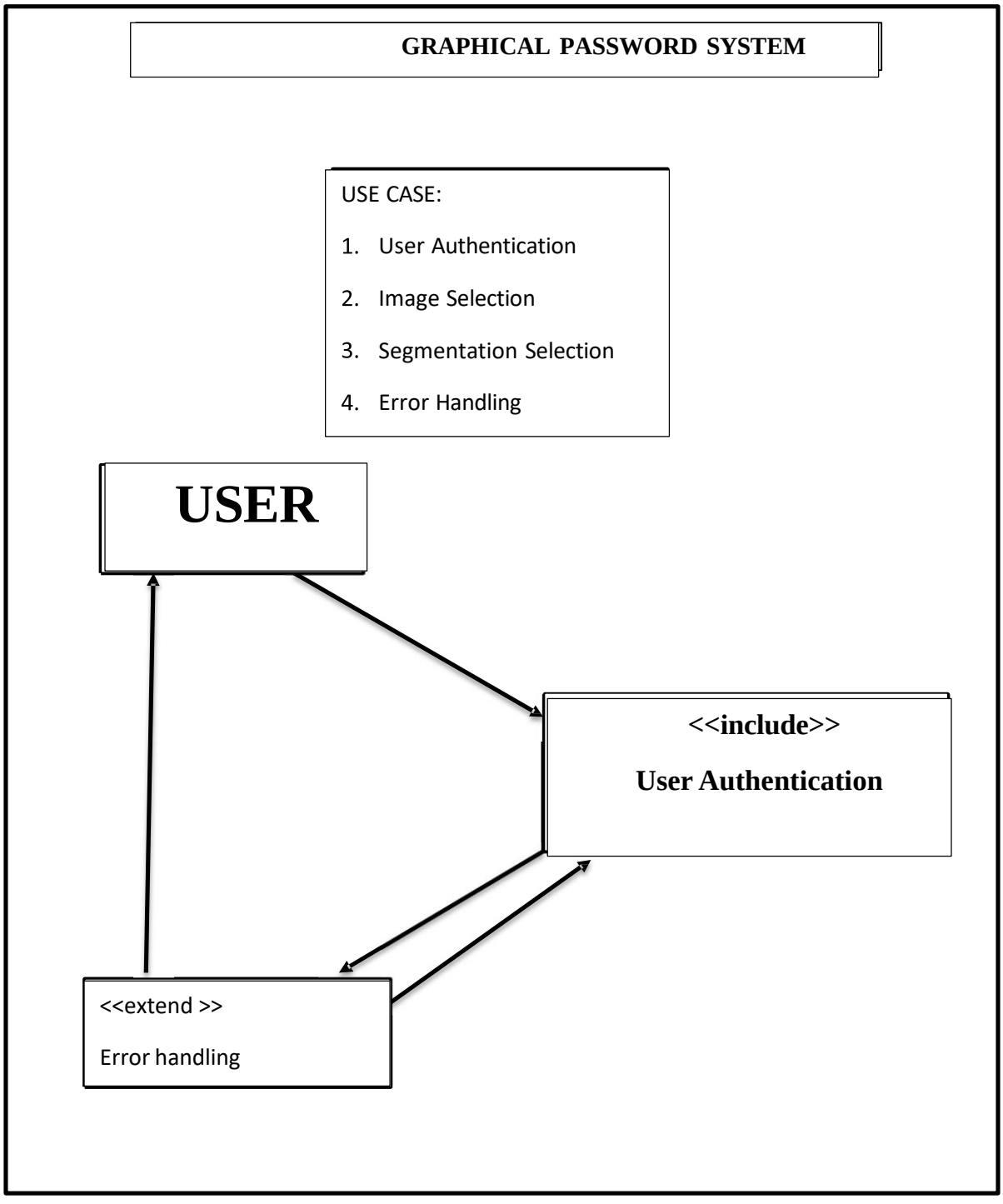


Fig – i(Use Case Diagram)

7.3 Use Case Scenario (Following details can be provided for a use case scenario)

Use Case Scenario: User Authentication with Graphical Password

Scenario Title: User Logs into the System

Actors:

- **Primary Actor**: User
- **Secondary Actor**: System (Graphical Password System)

Preconditions:

- The user has already created an account and set up a graphical password.
- The user has access to a device with internet connectivity to reach the graphical password system.
- The user remembers the image and the segments they selected during the password creation process.

Trigger:

- The user wants to log into their account to access their personalized dashboard and features.

Main Flow of Events:

1. **User Navigates to the Login Page**:
    - The user opens their web browser or app and enters the URL for the graphical password system.
    - The system displays the login interface, prompting the user to log in. 2. **User Selects Their Image**:
    - The system presents a set of images that the user can choose from, which they used when creating their graphical password.
    - The user reviews the images and selects their designated image. 3. **System Displays Segments**:
    - Once the user selects the image, the system segments the image into identifiable regions (e.g., grid layout).
    - The user observes the segmented image displayed on their screen. 4. **User Selects Segments**:
    - The user interacts with the segmented image by clicking on the specific segments they had chosen during their password setup.
    - The system captures the selected segments. 5. **System Verifies Input**:
    - After the user makes their selections, the system processes the input to compare the selected segments with the stored graphical password.
    - The system performs the verification process.
6. **Authentication Outcome**:

- **Successful Authentication**:
  - If the selected segments match the stored graphical password, the system grants access and redirects the user to their account dashboard.
  - The user sees a welcome message and their personalized
- **Failed Authentication**:
  - If the selected segments do not match, the system displays an error message stating, "Authentication failed. Please try again."
  - The user is given the option to retry selecting the segments or click a "Forgot Password?" link to initiate the recovery process.

Alternate Flows:

- **AF-1: User Forgets Their Password**:
  1. If the user clicks on "Forgot Password?":
     - The system prompts the user to enter their email address associated with the account.
     - The user receives an email with instructions on how to reset their graphical password.
- **AF-2: Invalid Image Selected**:
  1. If the user tries to select an image that is not available:
     - The system displays an error message indicating that the image is invalid. ▪ The user must select a valid image from the available options.
- **AF-3: Session Timeout**:
  1. If the user takes too long to make their selections:
     - The system automatically logs them out and displays a message, "Session timed out. Please log in again."

Postconditions:

- The user is either logged into their account with access to their dashboard or remains on the login page with an error message, depending on the success or failure of the authentication attempt.

CHAPTER 3
(For Web-Based Projects)
SYSTEM DESIGN

3.1. Architecture diagrams

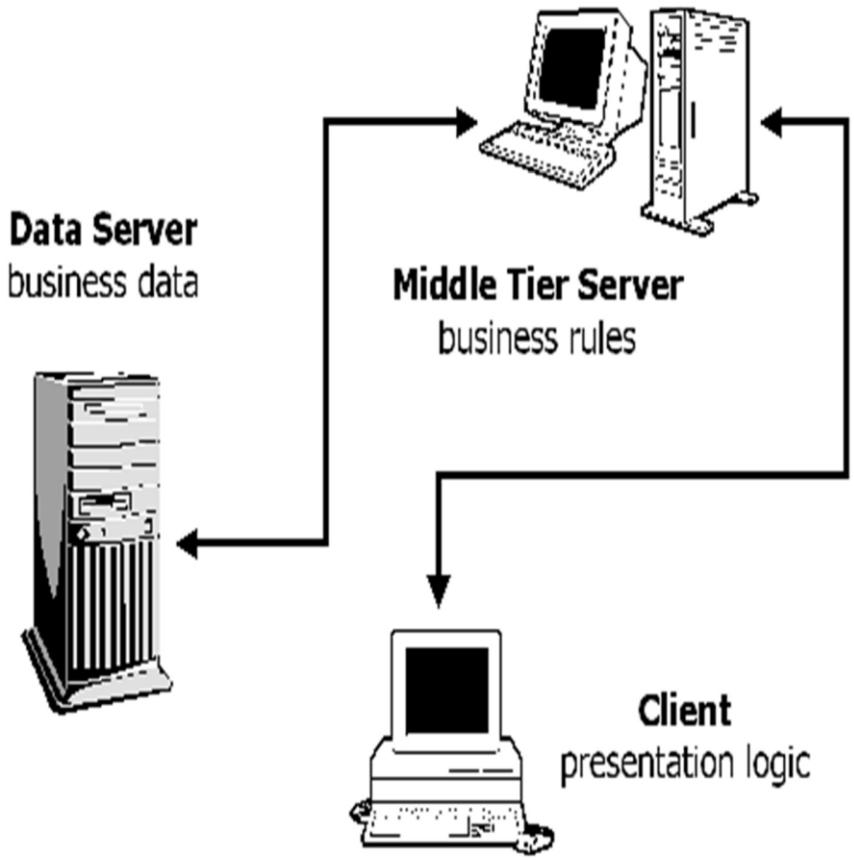


Fig ii(architecture diagram)

3.2. Class diagrams

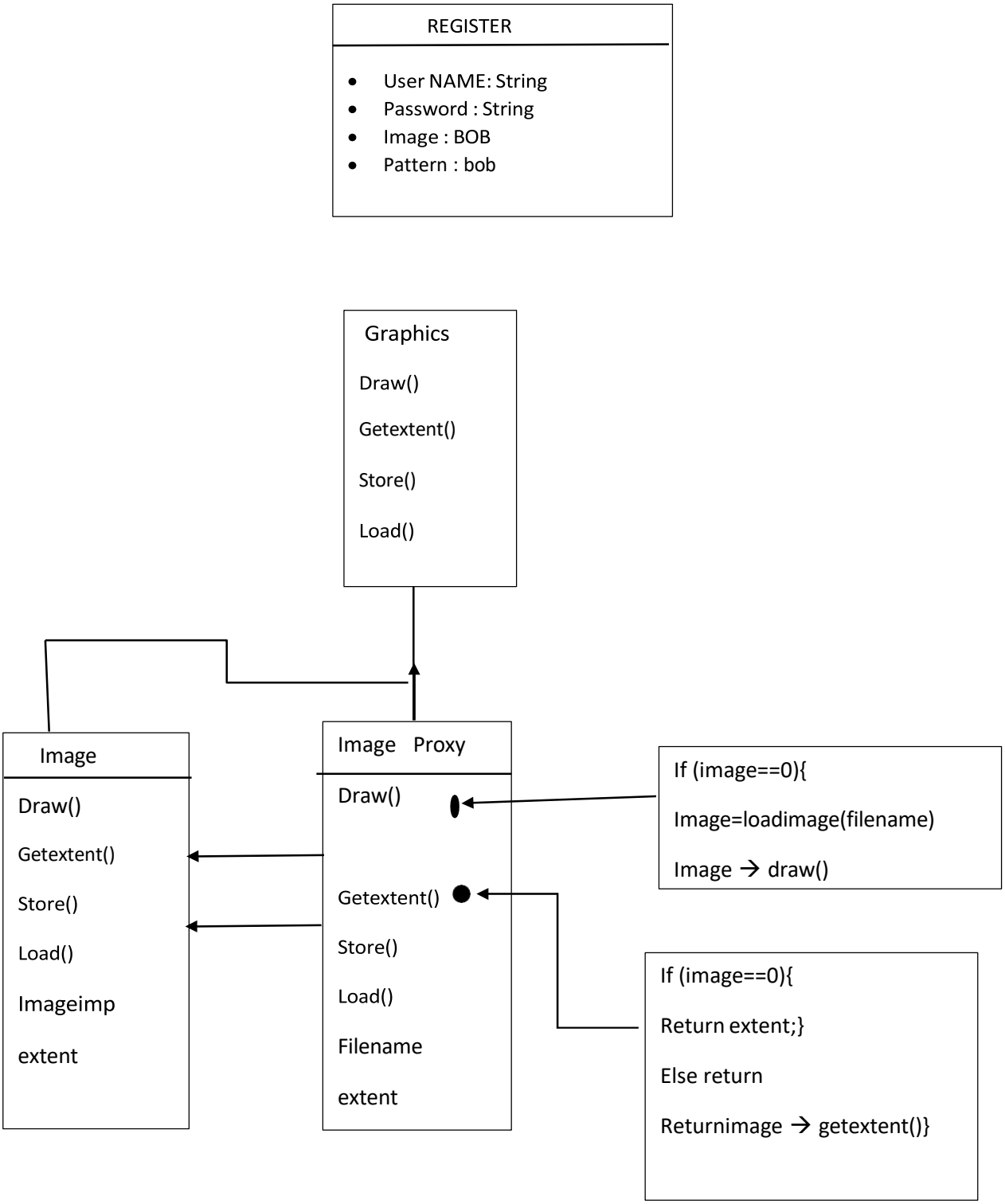
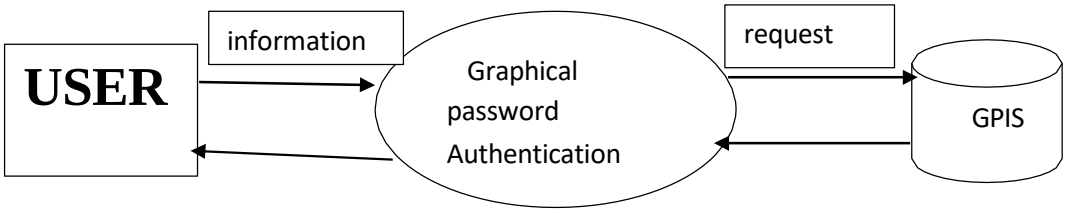


Fig iii(Activity Diagram)

3.3. Data Flow Diagram

DFD Level-0



DFD Level-1

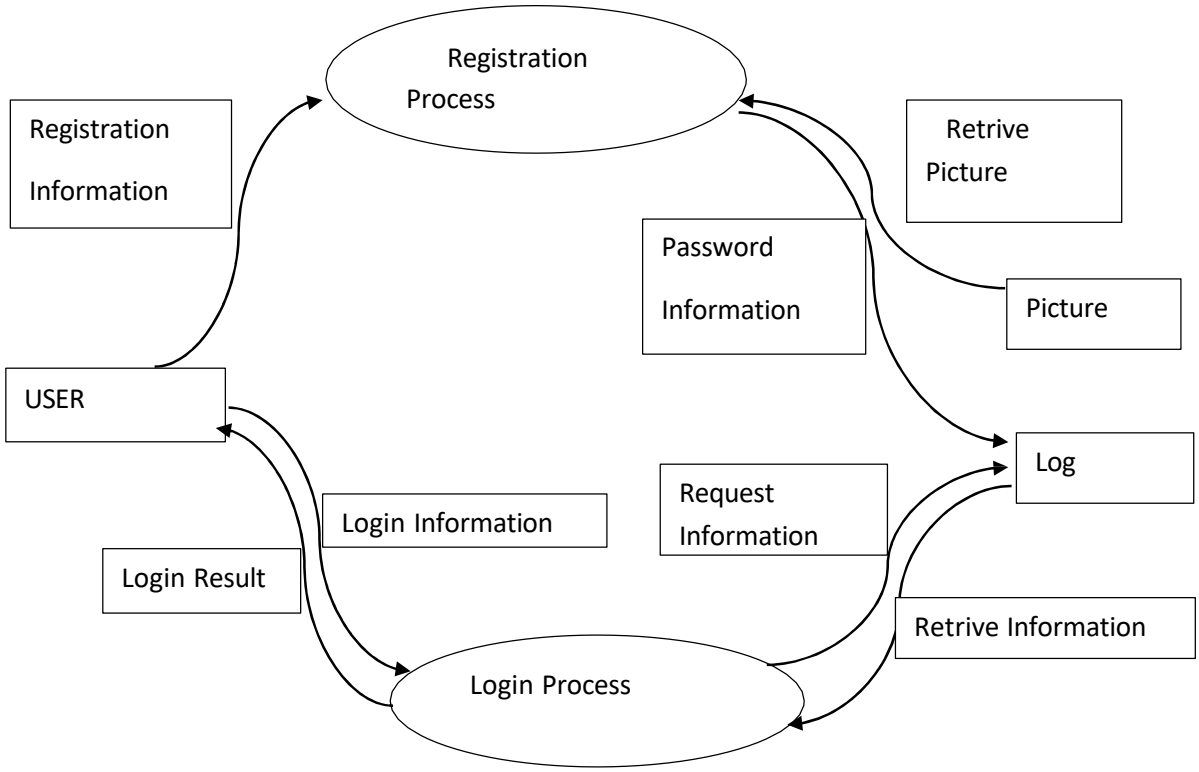
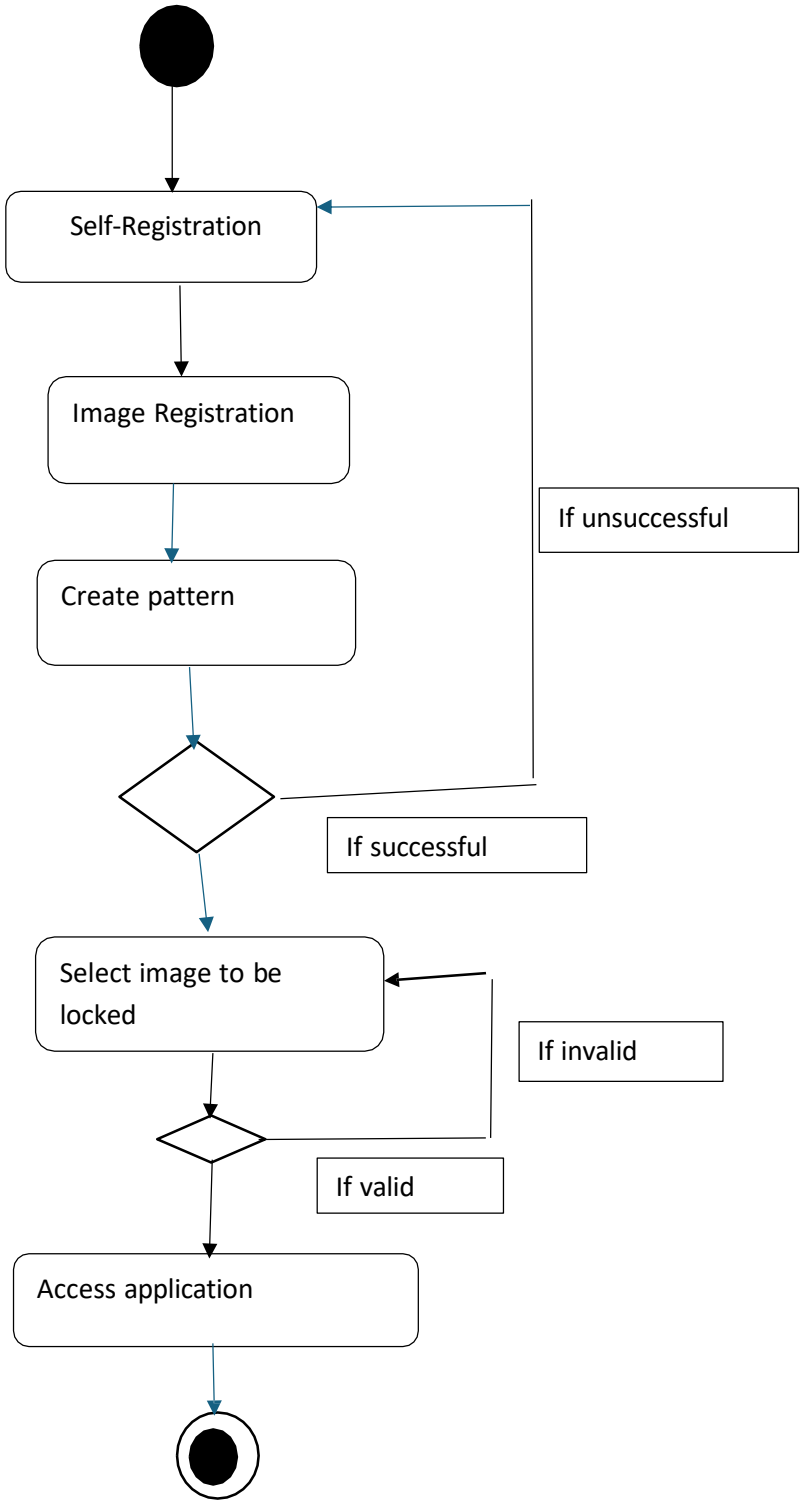


Fig – iv(Data Flow Diagram)

3.4. Activity Diagram (Example for Registration and Login)



3.5. ER Diagrams

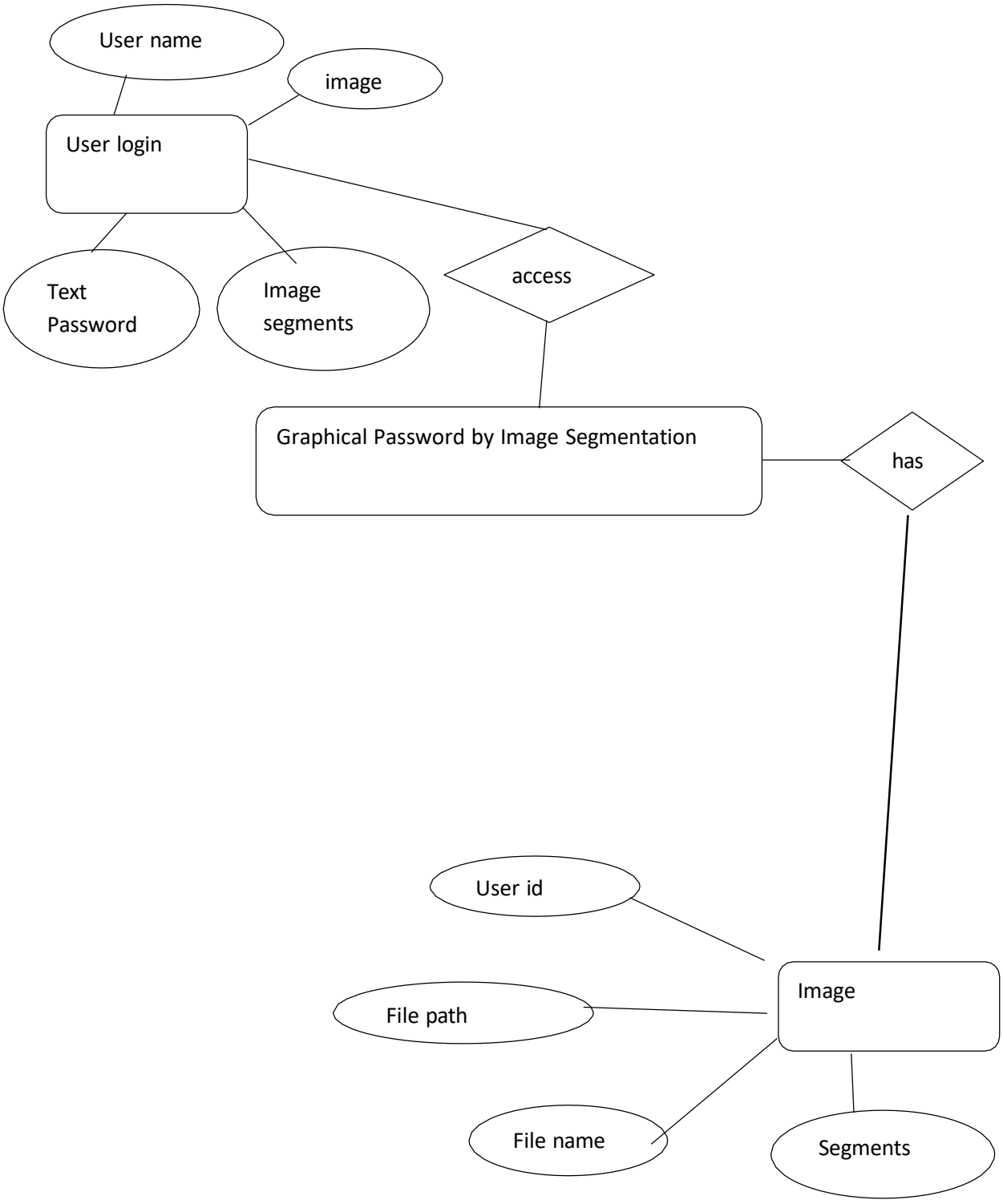


Fig vi (ER Diagram)

CHAPTER 4
IMPLEMENTATION AND RESULT

4.1 Software and Hardware Requirements

a. Software Requirements

1. **Operating System**:
    - **Server-side**:
        - Linux (Ubuntu, CentOS, etc.) or Windows Server for hosting the system.
    - **Client-side**:
        - Cross-platform support (Windows, macOS, Linux) for users accessing the system.
        - Mobile OS (iOS, Android) if the system includes a mobile application.
2. **Web Server**:
    - **Apache**, **Nginx**, or **IIS** to serve the web application to users.
    - **WebSocket support** for real-time authentication responses (optional).
3. **Database**:
    - **Relational databases**: MySQL, PostgreSQL, or SQLite for storing user information, image data, and graphical passwords.
    - **NoSQL databases**: MongoDB, Firebase for flexibility if dealing with unstructured image data and segments.
    - Data storage should ensure encryption for user credentials and password segments.
4. **Programming Languages**:
    - **Back-end**:
        - Python (Django, Flask), Node.js, PHP, or Java (Spring) for server-side programming.
        - Python's Pillow or OpenCV library for handling image segmentation.
    - **Front-end**:
        - HTML5, CSS3, JavaScript (with frameworks like React, Angular, or Vue) for building the user interface.
        - SVG or Canvas API for rendering images and handling user interactions with segmented images.
    - **Mobile Development** (optional):
        - Swift for iOS, Kotlin/Java for Android.
5. **Encryption Libraries**:
    - **AES-256 encryption** for securely storing and encrypting user passwords and images.

- Libraries such as **OpenSSL**, **PyCryptodome**, or **BCrypt** for secure password hashing and encryption.

6. **Image Processing**:
   - **Image Segmentation Tools**:
     - OpenCV or custom-built algorithms to handle image segmentation, allowing the system to break an image into segments that users will select.
   - **Image Storage**:
     - AWS S3, Google Cloud Storage, or a local server for storing uploaded images securely.
7. **Authentication**:
   - **OAuth 2.0** or **JWT** for handling secure user sessions and token-based authentication.
   - **Two-factor authentication (optional)**: Implement third-party services like Google Authenticator or SMS-based OTP as an added layer of security.
8. **API for Image Segmentation (Optional)**:
   - If the system needs to segment images dynamically on a server, an API might be used to process images in real-time.
9. **Development Tools**:
   - **IDE**: Visual Studio Code, PyCharm, or IntelliJ IDEA for efficient code writing and testing.
   - **Version Control**: Git/GitHub or GitLab for managing code versions and collaboration.

b. Hardware Requirements

1. **For Server (Backend)**:
   - **Processor**:
     - At least **quad-core** Intel Xeon or AMD Ryzen processors, depending on the load and number of users.
   - **RAM**:
     - Minimum **8 GB** for small to medium-sized deployments.
     - **16 GB** or more for larger-scale systems handling multiple concurrent users.
   - **Storage**:
     - **SSD storage** (at least **500 GB**) to store user information, images, and password data.
     - Scalable storage solutions like AWS S3 or Google Cloud Storage for high-volume image data.
   - **Network Bandwidth**:
     - Minimum **100 Mbps** internet speed for real-time image processing and user interactions.
     -

2. **For User (Client-Side)**:
    - **Device**:
        - Desktop/laptop with a modern web browser (Google Chrome, Firefox, Safari).
        - Mobile devices (smartphones and tablets) should be equipped with basic processing power (minimum **2 GB RAM**) to handle the image rendering and segment selection.
    - **Display**:
        - A **high-resolution display** (minimum 720p) for clear visualization of image segments.
        - Touch support for mobile devices to enable users to select image segments with ease.

3. **Image Processing Hardware (Optional)**:
    - If the system processes a large number of images, consider **GPU acceleration** (NVIDIA CUDA-enabled GPUs) for faster image segmentation processing.

4. **Backup and Recovery Hardware**:
    - **RAID-configured hard drives** or cloud-based backups for data redundancy and recovery in case of hardware failures.
    - Cloud services like **AWS Backup**, Google Cloud Storage, or local physical storage for routine backups of sensitive user data and system files.

CHAPTER 5

CONCLUSION

The graphical password system utilizing image segmentation is an innovative approach to user authentication that aims to overcome the limitations of traditional text-based passwords by introducing more intuitive, secure, and user-friendly interactions. Below is a detailed conclusion of the system's advantages, challenges, and the overall impact it can have:

1. Enhanced Security

Graphical passwords inherently provide stronger security than text-based passwords because:

- Memory Advantage: Users find it easier to remember visual patterns or images compared to long, complex text passwords, reducing reliance on insecure practices like writing passwords down or reusing them across platforms.
- Increased Password Space: With images segmented into multiple parts, the possible combinations of segments a user can choose vastly increase, thus making brute-force attacks more challenging.
- Resilience to Keyloggers: Graphical passwords reduce vulnerability to common threats like keylogging and phishing, as attackers need to know the exact image and the corresponding segments to compromise the system.

2. Improved Usability

- User-Friendly Interface: The system provides an engaging and intuitive interface where users select images and specific segments, enhancing the user experience, especially for non-technical users.
- Cross-Platform Functionality: With proper design, the graphical password system can work seamlessly on desktop, mobile devices, and tablets, offering a touch-friendly interface that makes password selection easy, especially on touch-enabled devices.

3. Software and Hardware Efficiency

The software and hardware requirements ensure that the system can scale effectively:

- Scalability: Leveraging modern web servers (e.g., Apache, Nginx) and databases (MySQL, MongoDB), the system can handle a large number of users and complex image segmentation tasks. Cloud-based storage solutions like AWS or Google Cloud further enhance scalability and ensure data availability and redundancy.
- Image Processing: Using tools like OpenCV and Pillow for segmenting images ensures that image processing is efficient and quick. For systems with high traffic,

GPU acceleration may be added to enhance performance further, ensuring real-time segmentation and password validation.

- Encryption and Security Standards: Utilizing encryption standards like AES-256 ensures that user passwords and images are securely stored and protected from potential breaches.

4. Challenges and Constraints

While the graphical password system offers many advantages, it also comes with challenges that need to be addressed:

- Usability vs. Security Trade-off: Although the system improves usability, there is still a balance that needs to be struck between ease of use and security. Users may find it difficult to remember complex segmentation patterns over time, which may lead to frequent password resets.
- Device Compatibility: Ensuring that the system works equally well on all platforms (desktop, mobile, tablet) can be technically challenging. Additionally, segmenting images on devices with smaller screens might lead to usability issues if not designed with responsiveness in mind.
- Storage and Processing: Storing high-resolution images and segment data in the database can lead to storage bloat if the user base grows rapidly. Efficient database management and cloud storage solutions are critical in handling large volumes of image data.

5. Broader Implications

- Application in High-Security Domains: This system is well-suited for high-security domains such as financial services, healthcare, and government sectors where password-based authentication is essential, but existing methods may fall short. The added layer of complexity introduced by image segmentation can deter attackers and improve the overall security landscape of such applications.
- Customizability: The flexibility of choosing different images and segmenting them as per user preferences makes this system highly customizable for different industries and user bases. It could be adapted for gamification, corporate branding, or educational tools to create a more personalized experience.

6. Future Developments

- Integration with Biometrics: The graphical password system can be further enhanced by integrating with biometric authentication methods like fingerprint recognition or facial recognition. This combination can offer multi-factor authentication (MFA),

where users would need both their graphical password and biometric data to authenticate.

- AI-based Image Segmentation: In the future, artificial intelligence (AI) could be employed to dynamically segment images based on object recognition, making the image segmentation process smarter and more adaptable to a user's behavior.
- Augmented Reality (AR): With the rise of AR technologies, graphical passwords can evolve to use immersive 3D images or objects, taking user interaction to a new level.

Conclusion Summary

In conclusion, the graphical password system using image segmentation offers a unique and secure alternative to traditional passwords. By leveraging the human brain's innate ability to remember visual patterns more easily, this system improves both the security and usability of password-based authentication methods. With the right balance of software and hardware resources, the system can scale efficiently and provide an enhanced user experience across different platforms.

While it faces some challenges related to user memory retention and compatibility, its potential for high-security applications and future integrations with cutting-edge technologies like biometrics and AI make it a highly promising innovation. This system can reduce reliance on text-based passwords, strengthen defences against common cyber threats, and significantly improve the overall security posture of any application or service.

SCREENSHOTS

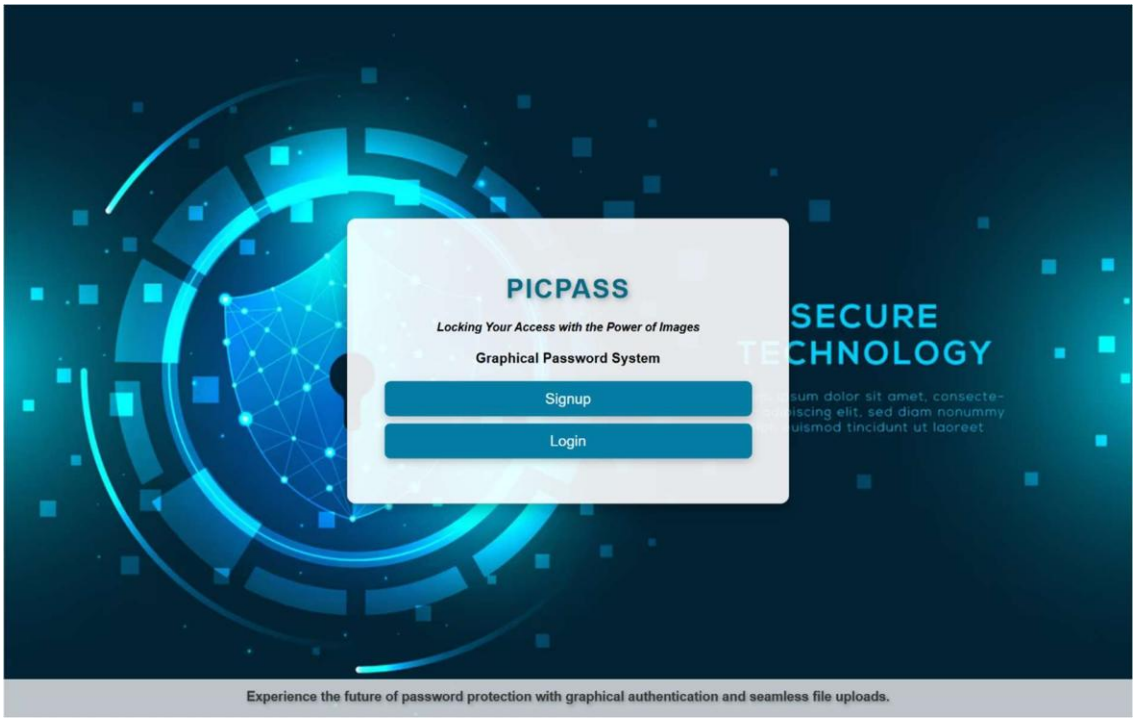


Fig vii :- SCREEN 1 – HOME PAGE

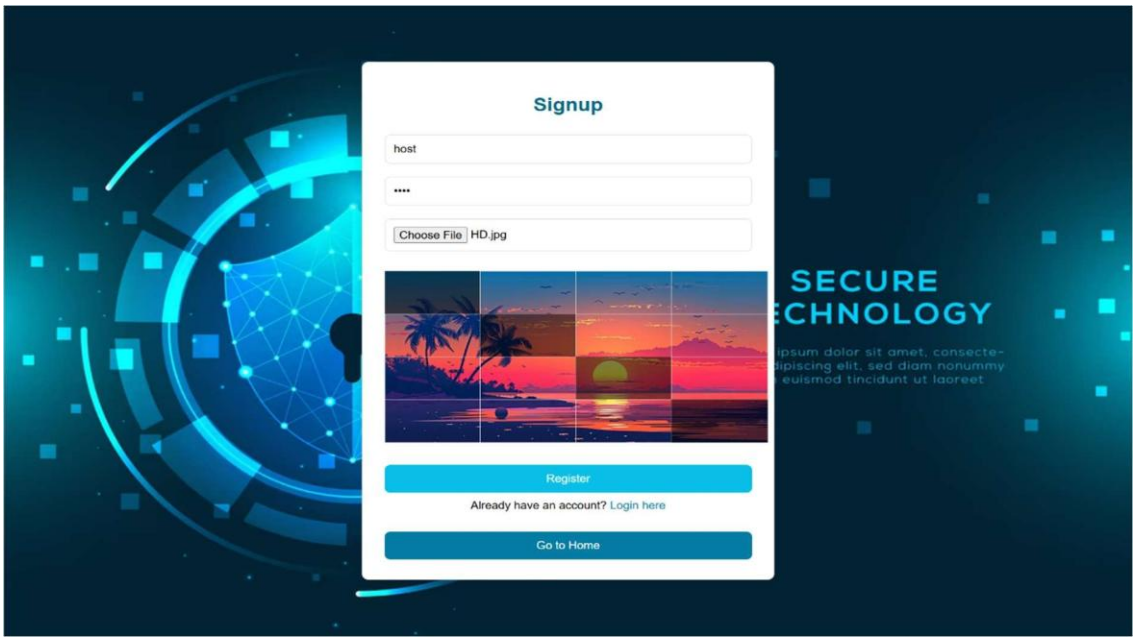


Fig viii:- SCREEN 2 – Signup Page

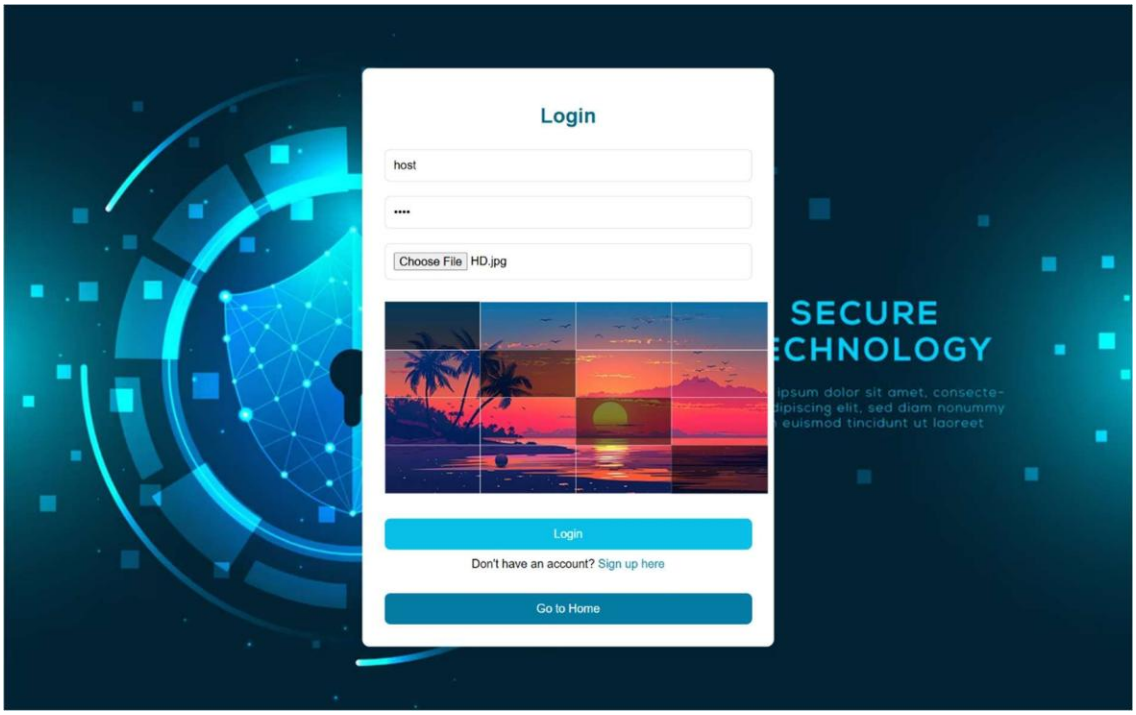


Fig ix :- SCREEN 3 – Signin Page

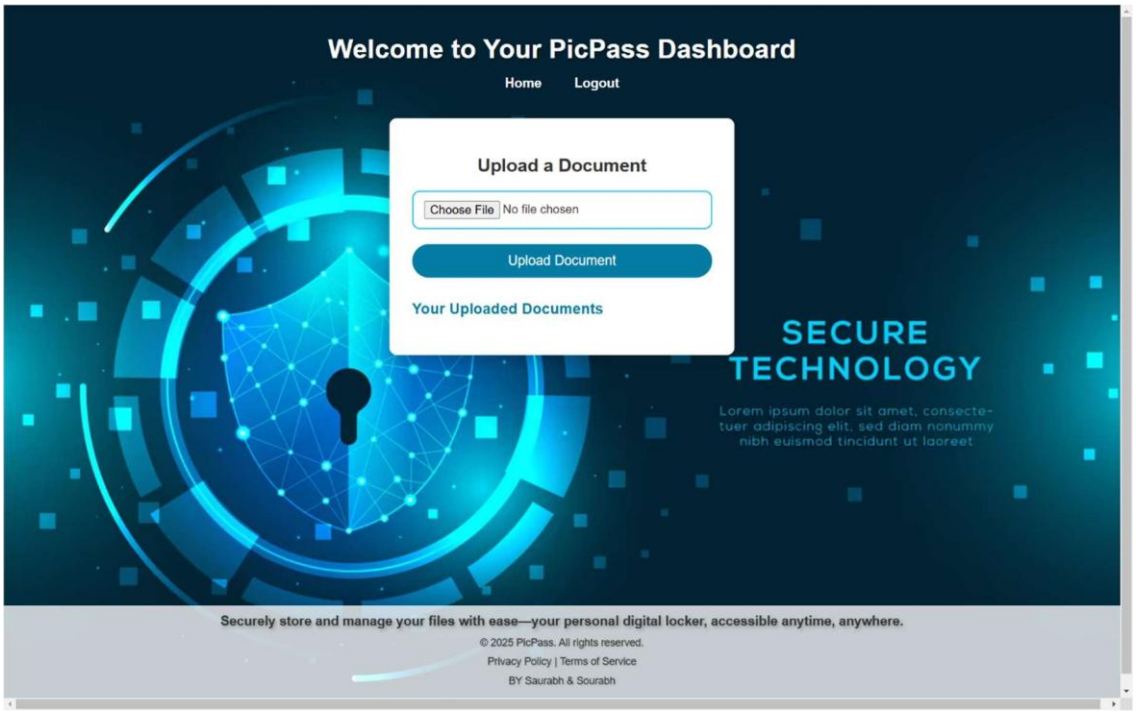


Fig ix :- SCREEN 4 – Dashboard

REFRENCES

1 Books:

- Schneier, B. (2015). *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley.
    - A comprehensive guide to encryption and security mechanisms relevant to password systems.

2 Research Papers:

- Wiedenbeck, S., Waters, J., Birget, J. C., Brodskiy, A., & Memon, N. (2005). *PassPoints: Design and longitudinal evaluation of a graphical password system*. *International Journal of Human-Computer Studies*, 63(1-2), 102-127.
    - Discusses graphical password systems and usability testing.

3 Web Resources:

- OWASP (2025). *Authentication Security Guidelines*. https://owasp.org/
    - Best practices for designing secure authentication systems.

4 Articles:

- Kumar, P., & Zhang, L. (2023). *Graphical Password Authentication: A Comparative Study*. *Journal of Information Security*, 12(3), 45-60.
    - A comparative analysis of various graphical password methods.

5 Official Documentation:

- OpenCV. (2025). *Image Processing Techniques*. https://opencv.org/
    - Detailed documentation on image segmentation and related techniques