MAJOR PROJECT - 1 SRS

“SMART MIRROR”

Submitted By:-

Name	SapID	Specialization
Arpit Kansal	500091746	B.tech (IoT and SC)NH
Madhav Maheshwari	500092026	B.tech (IoT and SC)NH
Nikita Singh	500091108	B.tech (IoT and SC)NH

Under the guidance of

Dr. Rohit Tanwar

Associate Professor at UPES, Dehradun

Department of Cybernetics

School of Computer Science

UPES

Dehradun-248007

Oct-Dec, 2024

INDEX

S.no	Topic	Page.no
1	Introduction	1
	1.1 Purpose of the Project	1
	1.2 Target Beneficiary	2
	1.3 Project Scope	2
	1.4 References	
2	Project Description	2
	2.1 Data/ Data Structure	3
	2.2 SWOT Analysis	3
	2.3 Design Diagram	4
	2.4 Project Features	4
	2.5 User Classes and Characteristics	5
	2.6 Design and Implementation Constraints	5
3	System Requirements	5
	3.1 User Interface	5
	3.2 Protocols	5
4	Non-functional Requirements	6
	4.1 Performance requirements	6
	4.2 Security requirements	6
	4.3 Software Quality Attributes	6
5	Glossary	6

1 Introduction

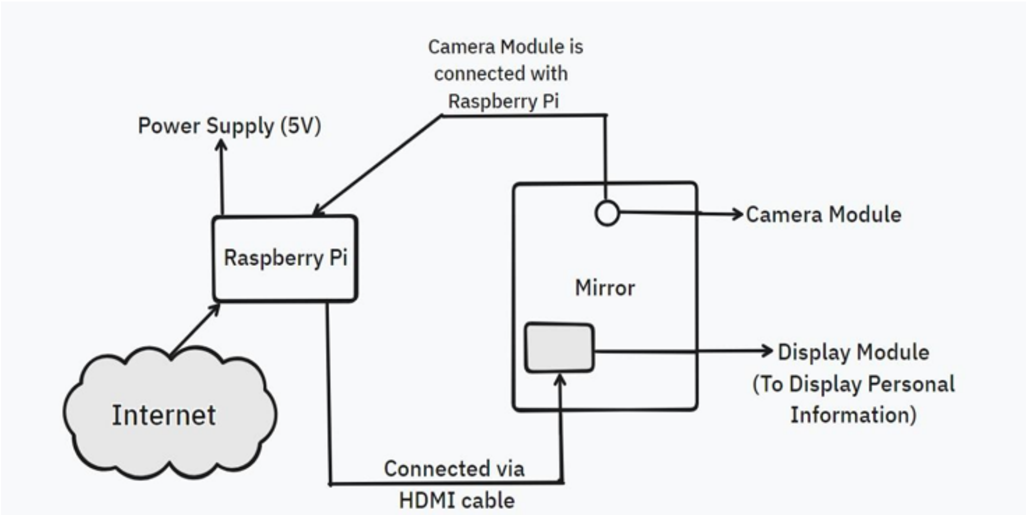


Figure 1: Project layout

The Smart Mirror is an interactive device that integrates facial recognition technology with a customizable user interface, making it a practical tool for smart homes and personal productivity. Unlike traditional mirrors, the smart mirror displays dynamic information such as time, weather updates, calendar reminders, and personalized content for users. By utilizing a Raspberry Pi 4 as its core processing unit, coupled with a Pi Camera and a display, the project leverages facial recognition to offer a tailored experience.
The smart mirror project stands at the intersection of home automation and artificial intelligence (AI), demonstrating the potential for everyday objects to become smarter and more useful. The incorporation of the MagicMirror framework allows for flexibility in adding modules and functionalities, making the mirror more than just a reflective surface—it becomes a personalized digital assistant. The project also explores the potential of expanding this smart mirror's functionalities to integrate with smart city initiatives, focusing on both individual convenience and larger infrastructural improvements.

1.1 Purpose of the Project

The purpose of the smart mirror project is to explore how an ordinary object can be enhanced with technology to provide real-time data and personalized services. This project is a proof of concept for how IoT devices can integrate into everyday life, contributing to the broader idea of a smart home ecosystem.
By utilizing facial recognition technology, the mirror not only identifies users but also provides tailored information based on their preferences. This could range from displaying a daily schedule for a professional to suggesting health metrics or news updates for other users. It merges both utility and convenience, offering a futuristic solution to how individuals interact with technology in their home environments.

1.2 Target Beneficiary

The primary beneficiaries of the smart mirror project include homeowners and individuals interested in home automation. This demographic is looking for technology solutions that simplify daily routines while providing more personalized, real-time information. Additionally, the project targets tech enthusiasts who enjoy exploring new innovations in AI and IoT, offering them a platform that can be customized and expanded upon.
In a broader sense, the project could also be of interest to the smart city and smart home industry, where it represents an emerging trend toward making everyday objects "smart." Professionals working in fields related to urban development, interior design, and technology integration could see this device as a product or system that bridges the gap between personal convenience and larger IoT infrastructures.

1.3 Project Scope

- Research and Development: Research will be conducted to understand user needs, preferences, and the technological requirements for facial recognition and security features. The development phase will focus on optimizing the recognition algorithms and creating a more dynamic display interface.
- Design and Engineering: The physical structure of the Smart Mirror will be designed, alongside the development of enhanced facial recognition software and new security protocols. A custom display interface will be created to present personalized content beyond the limitations of the basic Raspberry Pi screen.
- Testing and Refinement: Extensive testing will be performed to ensure the accuracy and reliability of the facial recognition system, focusing on the improvements made for higher performance and scalability. The custom interface will also undergo user experience testing.
- Integration: The Smart Mirror will be integrated into various environments, such as homes, offices, hotels, and retail stores. In the major project, it will also be tested for larger commercial settings where high-accuracy recognition and intuitive display interfaces are critical.

1.4 References

- Yusri, M. M. I., Kasim, S., Hassan, R., Abdullah, Z., Ruslai, H., Jahidin, K., Arshad, M. S. (2017, May). Smart mirror for smart life. In 2017 6th ICT International Student Project Conference (ICT-ISPC) (pp. 1-5). IEEE.
- Johri, A., Jafri, S., Wahi, R. N., Pandey, D. (2018, December). Smart mirror: A time-saving and affordable assistant. In 2018 4th International Conference on Computing Communication and Automation (ICCCA) (pp. 1-4). IEEE.
- Sun, Y., Geng, L., Dan, K. (2018, January). Design of smart mirror based on Raspberry Pi. In 2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS) (pp. 77-80). IEEE.
- Patel, M. N., Shabandri, K. F., Balse, G., Naik, S. (2019). Smart Mirror. International Journal of Engineering Research and Technology (IJERT), 8(05).
- Sun, Y., Geng, L., Dan, K. (2018, January). Design of smart mirror based on Raspberry Pi. In 2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS) (pp. 77-80). IEEE.
- Kumar, A., Singh, J. (2018, August). Smart Mirror with Advanced Features and IoT Integration. In 2018 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-6). IEEE.
- Yusof, Y., Kamarudin, K. (2020, July). Implementation of Smart Mirror System using Raspberry Pi and Facial Recognition.

2 Project Description

1. Data Structure and Reference Algorithm: For facial recognition, a Raspberry Pi 4 and Pi Camera module will be used alongside the Eigenfaces algorithm. This algorithm, based on Principal Component Analysis (PCA), is suitable for resource-constrained devices like the Raspberry Pi. In the major project, further optimization of this algorithm will be explored, as well as experimenting with deep learning techniques for improved recognition accuracy.
2. Eigenfaces Algorithm
   - Data Collection: Images will be captured using the Pi Camera to serve as the training dataset.
   - Preprocessing: The captured images will be processed to ensure consistency in lighting, alignment, and scale, then converted to grayscale and resized if necessary.
   - Feature Extraction: Principal components will be extracted from the preprocessed images using PCA, with each component representing a unique facial feature.
   - Dimensionality Reduction: Only the most significant principal components will be retained, reducing the dataset's dimensionality.
   - Face Representation: Each face will be represented by its projection onto the reduced-dimensional space formed by the principal components.

- Face Recognition: To recognize a face, a new image will be captured and projected onto the same reduced-dimensional space. The new projection will be compared with the known faces using a similarity measure like Euclidean distance to identify the closest match.

3. In the major project, additional algorithms like Local Binary Patterns Histograms (LBPH) or a deep learning-based Convolutional Neural Network (CNN) may be incorporated to improve recognition accuracy, especially in diverse lighting conditions and angles.

2.1 Data Structures

```
# Dictionary to store facial data and PCA components
facial_data = {
    "faces": [],          # List to store preprocessed facial images
    "labels": [],         # List to store labels (identifiers) associated with each face
    "mean_face": None,    # Mean face vector
    "eigenfaces": [],     # List to store top-k eigenfaces
    "projection": []      # List to store projected faces onto the eigenfaces
}
```

A dictionary data structure in Python will be employed to store facial data and PCA components This dictionary structure simplifies the implementation of facial recognition on the Raspberry Pi using Python. The major project may explore more efficient storage solutions and data handling techniques as the system scales for larger datasets.

The Eigenfaces algorithm involves data preparation, mean calculation, data centering, eigenvalue decomposition, and projection onto the reduced eigenfaces subspace. For the major project, this mathematical foundation will be built upon by integrating alternative methods, such as using CNNs to reduce the error rate and improve recognition under various conditions.

1. Image Preprocessing:
   - Grayscale Conversion: The RGB image is converted to grayscale without any specific formula.
   - Resizing: The image is resized to a predefined width and height without any specific formula.
   - Normalization: The formula for normalization is: $Normalized\_Pixel\_Value = \frac{Pixel\_Value}{255}$, assuming pixel values range from 0 to 255.
2. Feature Extraction:
   - Haar Cascades: Various mathematical calculations based on Haar wavelets are utilized for feature detection.
3. Classification:
   - Template Matching: Pixel values of the detected facial features are compared with templates of known faces without any specific formula.
   - Nearest Neighbor Classification (Euclidean Distance): The formula for Euclidean distance is: $d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$.
4. Thresholding:
   - Distance Thresholding: The match function is defined as: $Match(d) = \{\ True, if d < Threshold$ $False, otherwise.$

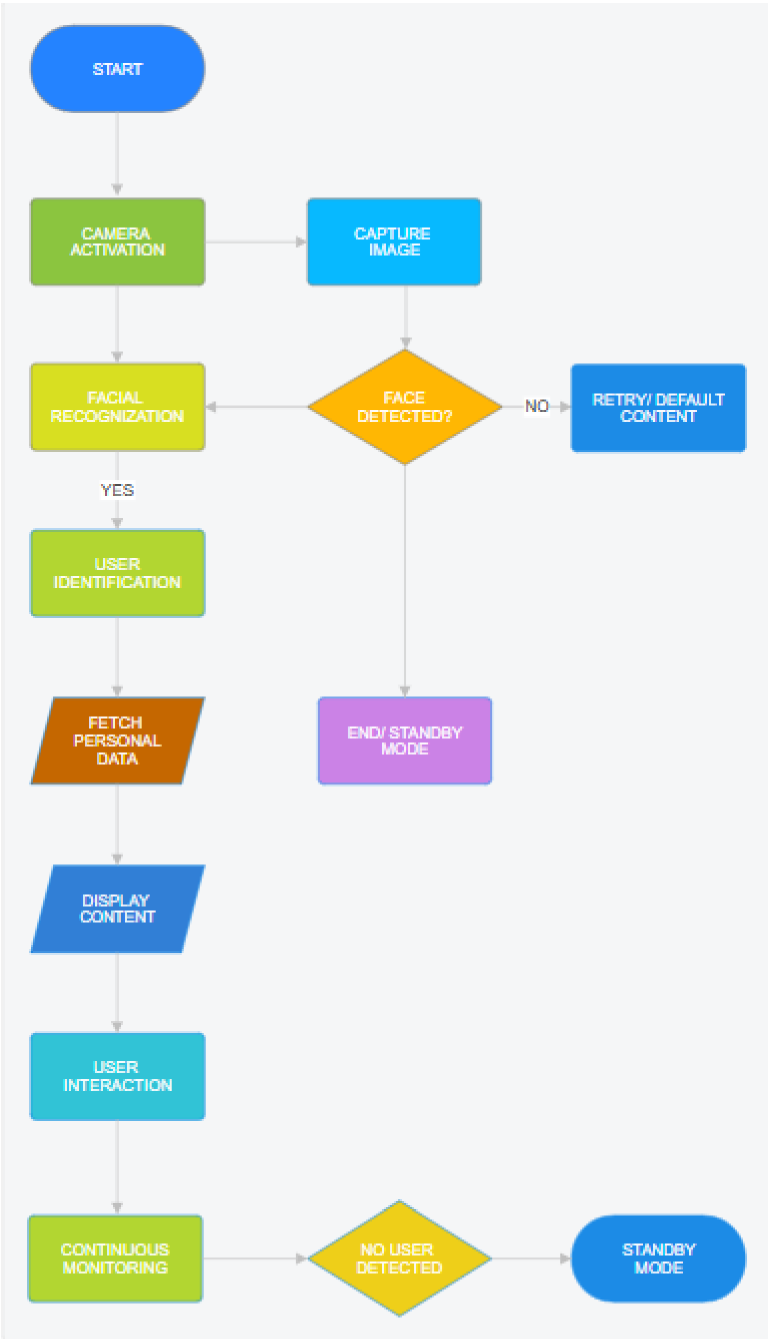
2.2 SWOT Analysis

1. Strengths: Innovative technology, convenience, enhanced security, and significant market potential. The new custom interface and improved recognition accuracy enhance the system's competitiveness.
2. Weaknesses: Dependence on technology, cost, privacy concerns, and limited functionality. Increasing the system's complexity may also add to development and operational costs.

3. Opportunities: Expansion into new markets, partnerships, technological advancements, and customization options. Enhanced features may open new opportunities for commercial applications.

4. Threats: Regulatory challenges, competition, cybersecurity risks, and economic factors. Competition in the smart security market remains a threat, as does evolving privacy legislation.

2.3 Design Diagrams



2.4 Project Features

1. Facial Recognition: Identifies and authenticates users.

2. Enhanced Facial Recognition: The major project will implement more advanced algorithms for higher accuracy.

3. Integrated Display: Provides real-time information such as weather, calendar events, and notifications.

4. Customizable Interface: Layouts and widgets can be personalized, with the major project introducing a new display interface with advanced customization.
5. Data Privacy Protection: Encryption protocols safeguard sensitive data.
6. User Recognition: Automatically adjusts settings based on user profiles.
7. Remote Access: Controlled via mobile app or web interface.
8. Custom Display Interface: In the major project, a new custom-built interface will allow for more detailed, user-specific content beyond the capabilities of the basic Raspberry Pi screen.

2.5 User Classes and Characteristics

1. Administrators: Manage system configuration and permissions.
2. End Users: Regular users for authentication.
3. Security Personnel: Monitor system security and require real-time alerts.
4. Developers/Integrators: Implement the system into larger solutions.
5. Regulatory Authorities: Oversee compliance with privacy and security regulations.

2.6 Design and Implementation Constraints

1. Hardware: The Smart Mirror's design must consider physical component constraints such as display size, camera quality, and processing power. In the major project, these constraints will be re-evaluated as a custom interface and additional processing power may be required for advanced features.
2. Cost: Managing expenses is essential to maintaining affordability, but the increased complexity of the major project might result in higher development costs.
3. Technical Compatibility: Seamless integration with smart home devices will continue to be a priority. The new interface will be designed to improve compatibility with various smart home ecosystems.
4. Regulatory Compliance: Must meet privacy and security standards.
5. User Experience: The major project will focus on improving user experience through a more intuitive, customizable interface.

3 System Requirements

3.1 User Interface

The smart mirror's user interface is designed to be clean, modern, and highly customizable. It displays various widgets, such as weather, news, time, and personalized greetings based on the user recognized by the facial recognition system. The MagicMirror framework allows for easy modification, enabling users to add or remove modules, change themes, and even develop custom widgets using HTML, CSS, and JavaScript.
The UI is designed with ease of use in mind, allowing users with little technical knowledge to interact with the mirror effortlessly. The layout is minimalistic, ensuring that only relevant information is displayed without overwhelming the user. The modular nature of the MagicMirror^2 platform ensures that developers can add or adjust widgets according to user preferences.

3.2 Protocols

The smart mirror utilizes several communication protocols for data retrieval and processing:

- HTTP (Hypertext Transfer Protocol): Used for retrieving real-time data such as weather, news, and calendar events from web-based APIs.
- WebSocket: Employed for real-time communication between different components, ensuring that updates are pushed to the mirror's display without requiring a manual refresh.
- SSL/TLS: Ensures that all communications, especially those involving user data, are encrypted for security and privacy purposes.

The system uses REST APIs to fetch external data like weather and news. Communication between the hardware components, such as the Raspberry Pi and Pi Camera, is handled through GPIO pins, allowing for smooth interaction between the hardware and software layers.

4 Non-functional Requirements

4.1 Performance Requirements

The performance of the smart mirror system is crucial for ensuring a smooth and responsive user experience. The facial recognition system must respond within 2-3 seconds after the user steps in front of the mirror, and the display should update without noticeable lag. The Raspberry Pi 4, being the core processor, handles data processing and real-time updates, but it is necessary to ensure that the software is optimized to avoid overloading the CPU or causing delays.
The system should also efficiently manage its resources when multiple users are detected or when it processes multiple requests simultaneously. The smart mirror must be able to operate 24/7 with minimal downtime or the need for frequent reboots.

4.2 Security Requirements

Security is a major concern in this project, especially since it involves facial recognition and personal data. The smart mirror must ensure the privacy and security of user data by:

- Encryption: All user data, especially facial recognition models and personal preferences, must be encrypted at rest and in transit.
- Authentication: Implement secure authentication methods to prevent unauthorized users from accessing personal information or making changes to the system.
- Data Privacy: Strict data privacy protocols must be followed to ensure that facial recognition data is not stored indefinitely and is only used for the purposes intended.

4.3 Software Quality Attributes

- Reliability: The system should be highly reliable, with minimal errors in facial recognition and display updates. The software and hardware components must be able to handle continuous operation without crashing or lagging.
- Maintainability: The system is built using open-source components like Raspberry Pi and MagicMirror$^{\mathbf{2}}$, making it easy to maintain and update. New features or modules can be easily added or adjusted as per user requirements.
- Usability: The smart mirror is designed to be user-friendly, with a simple interface that anyone can navigate. The customization options also make it suitable for a wide range of users, from tech-savvy individuals to those with minimal technical knowledge.

5 Glossary

- AI – Artificial Intelligence
- IoT – Internet of Things
- PCA – Principal Component Analysis
- CNN – Convolutional Neural Network
- LBPH – Local Binary Patterns Histograms
- SSL/TLS – Secure Sockets Layer / Transport Layer Security
- HTTP – Hypertext Transfer Protocol
- REST API – Representational State Transfer Application Programming Interface
- GPIO – General-Purpose Input/Output