Xi'an Jiaotong-Liverpool University
西交利物浦大學

SCHOOL OF INTERNET OF THINGS

IOT305TC FINAL YEAR PROJECT

Research and Implementation of An Affordable Simple Smart Mirror

Final Thesis

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Engineering

Student Name:	Yuesiyu Wu
Student ID:	2036834
Supervisor:	Prof. Matilda Isaac Dr. Shuangyao Huang
Assessor:	Dr. Wenzhang Zhang

Abstract

This project explores the design and implementation of a cost-effective and user-friendly smart mirror using a Raspberry Pi 4 Model B. The smart mirror aims to retain its reflective properties while providing interactive features such as real-time weather updates, news feeds, calendar events, and personalized compliments. The hardware setup includes an 11.6-inch IPS HDMI LCD panel, a USB industrial camera, a mini microphone, and a loudspeaker system, all selected for their affordability and efficiency. The software configuration leverages the MagicMirror² platform, allowing extensive customization through various modules. Despite challenges with voice control and facial recognition features, the project demonstrates significant progress in integrating advanced technologies into a seamless, multifunctional smart mirror. Future work will focus on resolving these issues and enhancing the smart mirror's capabilities to ensure it meets diverse user needs, thereby offering a versatile and engaging user experience.

Keywords: *Internet of Things, Smart Mirror, Interactive Features, Data Source, Software Interface*

Acknowledgements

With utmost sincerity, I extend my heartfelt gratitude to my esteemed supervisors, Prof. Matilda Isaac, and Dr. Shuangyao Huang, for their unwavering guidance, support, and encouragement throughout the journey of my undergraduate dissertation. Their wealth of knowledge, boundless patience, and insightful feedback have profoundly shaped the essence of this project.

I am immensely grateful to Dr. Bong-hwan Oh and Dr. Yuji Dong for their invaluable guidance and support throughout my academic pursuits. Their advice not only facilitated my personal growth but also contributed to maintaining my mental well-being. Their encouragement and mentorship have been instrumental in shaping my academic journey.

I would like to express my sincere gratitude to Dr. Bintao Hu and Dr. Wenzhang Zhang for their mentorship. They made my college life enjoyable. Their expertise and encouragement have been invaluable assets in my academic journey, contributing significantly to my growth and development.

Additionally, I would like to extend my appreciation to all the instructors who have taught and guided me throughout my academic journey. Their dedication, knowledge, and passion for teaching have played a significant role in my intellectual growth and academic achievements.

I am incredibly grateful for my parents, I owe an immeasurable debt of gratitude for their boundless love, unwavering faith, and unwavering support. Their unwavering encouragement and belief in me have been the cornerstone of my academic pursuit, propelling me forward through every challenge and triumph.

To my cherished friends, I am profoundly thankful for their companionship, infectious laughter, and unwavering support. In times of adversity, their accompany and boundless enthusiasm have been my guiding light, guiding me through my whole university life, even the darkest of moments.

Lastly, I extend my heartfelt appreciation for the invaluable support and resources provided by Xi'an Jiaotong-Liverpool University. The nurturing academic environment and access to cutting-edge research facilities have played an instrumental role in facilitating the completion of this dissertation.

Between the blooming and withering of flowers, wounds heal, sprouting anew. We exist in the fleeting moment, never to be repeated. As I continue my journey towards my goals, I will always remember everyone for their support and encouragement. I extend my sincerest gratitude for your unwavering belief in me.

Contents

1 Introduction **5**

2 Project Description **6**

2.1 Seamless Transition Between States 6
2.2 Customizable Features 7
2.3 Low-Cost Affordability 7
2.4 Hardware Integration 7
2.5 Enhanced User Interaction 8
2.6 Software Configuration and Customization 8
2.7 Seamless Operation and Maintenance 8
2.8 User-Friendly Interface 8
2.9 Problem Statement 8
2.9.1 Challenges 8
2.9.2 Solutions 9

3 Literature Review **10**

4 Methodology **12**

4.1 Tools and Instruments for Smart Mirror 12
4.1.1 Hardware Instruments 12
4.1.2 Software Instruments 16
4.2 Procedures 18
4.2.1 Hardware Splicing 18
4.2.2 Software Installation and Configuration 20

5 Achievement Exhibition **25**

5.1 Implemented Features 26
5.1.1 Calendar Module 26
5.1.2 Clock Module 26
5.1.3 Compliments Module 26
5.1.4 News Fees Module 26
5.1.5 Weather Module 27
5.1.6 Spotify Integration 27
5.1.7 Update Notification Module 28

6 Analysis of Cost-Effective and User-Friendly Smart Mirror **28**
6.1 Cost-Effective Hardware Choices 28
6.2 Simplified Installation and Maintenance 30
6.3 User-Friendly Interface 30
6.4 Enhanced Modular Customization 31

7 Conclusion **31**

8 Future Work **32**

Reference **33**

1 Introduction

Smart mirrors sometimes referred to as magic mirrors, can be considered quite a high-end amalgamation of conventional uses and current technological integration. Smart mirrors provide the user with an ordinary reflection and allow a computer interface placed behind the mirror to display real-time information such as weather updates, current news, and personal schedules. In fact, this is the history-enriched narrative behind the magic mirror, dating back to ancient mythology. Over centuries, the idea underwent important evolution, among which important developments in the 19th century featured hidden projectors and semi-reflective glass to create engaging optical illusions. In the 20th century, with the addition of early digital displays and sensor capabilities, it hit a record turning point in the 21st century with LCDs, embedded sensors, and augmented reality features. Since that time, advancements have redefined smart mirrors from mere flat, reflective surfaces to modern, interactive devices for both personal and public spaces [1].

The problem of the current designs of smart mirrors is, however, that they fail to be a mirror on a basic level when out of digital mode. As demand for more high-definition in display capability is increasing, most of the available smart mirrors lose their reflective capability when turned off, hardly being able to represent the basic idea of a mirror. This is a critical design failure that is a show-stopper element in smart mirrors becoming part of a user's daily life. Also, the fact that these gadgets cannot be used as a normal mirror from the default state affects majorly their usability and disrupts the balance between aesthetics and practical elements in homes and public settings [2]. Apart from that, a new feature addition was unviable without a strong hardware support system, which escalated the production cost sharply. Therefore, it is a problem to find a balance in the development of smart mirrors that is suitable in terms of increased capabilities and economic solutions.

This project makes the smart mirror possess all refinements through Raspberry Pi 4 so that the smart mirror retains the feature of being a mirror during both active and inactive modes. This will also help the normal usage of the mirror as a mirror during the mirror-off mode and as a smart interactive device when it is turned on. The use of advanced-level display technologies available in the market—like Transparent OLED or reflective LCD—makes the mirror stay as a reflective surface even during the time when information is displayed. This makes a perfect mix of form and function, where the smart mirror is able to shift its functionality gracefully between being a mirror when not in use and a smart device when in use, through the seamless integration of the two functionalities. Additionally, it would address the cost-effective hardware solutions, making it an inexpensive and widely spread smart mirror. This project will address all the inherent limitations associated with existing smart mirrors, thereby massively enhancing the user's experience. The result of the project will be a smart mirror that disappears into its environment while being off

but provides robust function once it is activated, thereby resetting the benchmark of smart mirror technology [3].

2 Project Description

In the realm of smart mirrors, the prevailing focus lies on features like information display, facial recognition, and basic interactivity. Therefore, to ensure the efficient implementation of these functions, many smart mirrors on the market directly use the camera to complete the initial imaging work of the mirror, and the mirror is also replaced by a high-definition display. This ensures both clear character imaging and high-quality hardware to support more interesting functions. However, such a smart mirror loses its basic imaging function—reflection as a mirror, and the price is also greatly increased at the same time. Consequently, a notable gap and price non-balance exists in seamlessly integrating these technological advancements into the mirror's core functionality.

This project aims to address these limitations by proposing an innovative solution that ensures the smart mirror functions as a conventional mirror when off-state, seamlessly transitioning to display pertinent information when activated. The goal is to provide users with a subtle yet highly functional mirror, enhancing their overall experience. The transition between states will be intuitive and user-friendly, blurring the line between an ordinary mirror and a smart one.

The proposed smart mirror will boast a seamless design integration, maintaining the appearance of a traditional mirror when not in use. Advanced display technologies such as transparent OLED or reflective LCD will be utilized to preserve the mirror's reflectiveness while displaying information, striking a harmonious balance between form and function. This approach aims to create a low-cost, versatile smart mirror that seamlessly integrates into users' daily lives, elevating their experience beyond conventional smart mirrors' capabilities.

2.1 Seamless Transition Between States

The transition between the mirror's off-state and on-state will be seamless and natural, ensuring uninterrupted functionality and user experience. When not in use, the mirror will maintain its reflective surface, resembling a conventional mirror and blending seamlessly into its surroundings. Upon activation, the mirror will display relevant information and interactive features, such as weather updates, news headlines, and calendar events, without compromising its reflective capabilities. This seamless transition ensures that the mirror remains a versatile and indispensable tool in users' daily routines, effortlessly transitioning between being a mirror and a smart display as needed.

2.2 Customizable Features

The value of the smart mirror can be greatly enhanced by offering customizable features tailored to individual customer preferences. By allowing customers to specify their desired functionalities during the purchasing process, developers can customize the software accordingly. For instance, customers may express interest in beauty and lifestyle recommendations, educational content for children, live sports updates for sports enthusiasts, or health-related information. This level of customization ensures that the smart mirror remains relevant and useful to a wide demographic of users with diverse needs and interests.

This adaptability not only leads to increased customer satisfaction and engagement but also creates opportunities for product diversification to meet evolving demands. For example, developers can offer different software packages or modules catering to specific interests or demographics, thereby fostering growth and competitiveness in the smart mirror market segment. Ultimately, such personalized content and adaptability serve to enhance the daily life experience, making the smart mirror a valuable addition to users' routines.

2.3 Low-Cost Affordability

In this smart mirror design, I pick all hardware components to be cost-effective. Affordability of all materials and components is considered, without compromising quality and functionality. This is how I can ensure that the cost of the hardware elements is minimized for manufacturing. This ensures that I can come up with an affordable smart mirror in the market without compromising on the performance and durability of the smart mirrors. Concentrating on cost-effective hardware solutions will also make this smart mirror technology available to a wide audience.

2.4 Hardware Integration

This project aspires to build a sturdy and stable smart mirror by using essential hardware components. These include a powerful microcomputer, a high-resolution display panel, and a camera module for visual input. Other components include a microphone for receiving voice input, a loudspeaker system for audio output, and a wireless keyboard for the development process. All these components will be enclosed in a personalized frame, which will be both structurally stable and aesthetically pleasing. Integration will involve proper placement and secure attachment to ensure the smart mirror's durability and full functionality.

2.5 Enhanced User Interaction

The developed system focuses on enhancing user interaction with the smart mirror, prioritizing a seamless and intuitive experience. Users can effortlessly engage with the mirror to access real-time information and perform tasks such as checking the weather, managing calendar events, and controlling music playback. This improved interaction emphasizes the human aspect of the smart mirror, ensuring that users can easily navigate its features and functionalities.

2.6 Software Configuration and Customization

This will involve configuring and personalizing software elements to enhance user experience. Personalization includes default modules like the calendar, weather, clock, and news feed, tailored to meet user needs and localized preferences. Additional third-party modules will provide personalized services, including music playback. The software will be designed for efficient dependency management, ensuring smooth updates and module management.

2.7 Seamless Operation and Maintenance

Ensuring the smart mirror remains operational at all times and maintaining high performance are key project goals. The setup will include features such as automatic reboot, load balancing, and monitoring to ensure uptime and performance. The smart mirror will function as a responsive background service, providing reliable operation. Additionally, the system will allow for easy updates to ensure the latest features and security patches are applied.

2.8 User-Friendly Interface

The project aims to design a highly user-friendly and intuitive interface, utilizing modern GUI development principles. The interface will be easy to navigate, based on responsive design principles and user-driven functionality. Users will have the ability to customize the layout, themes, and information display according to their preferences, providing an engaging and fluid experience.

2.9 Problem Statement

2.9.1 Challenges

Seamless Design Integration It is quite a task to be able to address the top-edge capabilities—say, transparent OLED or reflective LCD—without adding a significant cost. In general, such features will be of relatively higher costs and can actually increase the total production cost

of the smart mirror. In this context, therefore, the capability factor should be weighed against cost-effectiveness so that the smart mirror remains accessible to many consumers.

Seamless Transition Between States However, switching between the off-state of the mirror, as a normal mirror, and its on-state, as a digital information display, is definitely a highly technical process. The engineering innovations that are necessary guarantee that the mirror remains essentially reflective and digitalized simultaneously. It also includes optimization of display technologies, power management systems, and user interface designs that interfere as little as possible and make sure the experience is totally seamless.

Hardware Optimization It is very challenging to balance optimal performance and the cost of hardware components for a smart mirror. Only hardware component selection and optimization, such as processors, memory, and modules for connectivity, allow advanced functionalities to be implemented in a smart mirror with good affordability. Furthermore, it should be assured that every hardware device is connected correctly and can be put to use with proper functionality to set a base for later software development.

Software Module Installation and Customization This might be a quite serious challenge: to install and configure various software modules in such a way that personalized features and functionalities can be made available. Each of these modules might require fine-tuning according to the specific necessities and desires, and that would require a clear understanding of requirements and an effective way to implement those strategies. This, besides, happens to be a great technological challenge—to ensure seamless integration and compatibility of the different modules while still maintaining the stability and performance of the system.

2.9.2 Solutions

Seamless Design Integration I would adopt a multipronged approach to address the challenge of integrating advanced display technologies while remaining affordable. This would include extensive market research aimed at identifying viable low-cost alternatives for premium display technologies without compromising performance or visual quality. In addition, I emphasize design simplicity and efficiency, reducing material use and assembly complexity, which further helps with maintaining low production expenses.

Seamless Transition Between States Optimizing the hardware and software components so that they work in harmony for a seamless transition from one state to another could also be a way to reduce technical challenges. Energy-efficient display solutions will be developed that should not

draw much power and should be power-efficient, at the same time offering a clear, responsive view to the user. Further improvement in the design of the user interface will ensure easy navigation and interaction, which will not disturb the user in the course of state transitions. Specifically, the brightness of the display screen will not be high, to ensure the normal imaging of the mirror, and the user can also see the displayed information.

Hardware Optimization After a rigorous test and benchmarking, I will sieve the most cost-effective yet efficient hardware components. From the point of ensuring function, another part of the selection of hardware components that are low-cost and suitable will reduce the cost of the whole project. Specifications would be refined continuously to be sure the smart mirror hardware is market trend-friendly and cost-competitive in time, with technology moving forward.

Software Development Complexity The solution will take a stepwise approach to resolve the issues of software development complexity by embodying the characteristics of modularity, scalability, and flexibility. This means that the application will be developed using modular software architectures and application programming interfaces in a way that allows the easy integration of new functionalities and software upgrades. In addition, the API keys increase security by acting as an authentication credential, offer the ability to track and monitor the usage of the API, provide access control and control quota management, and enable the provision of personalized services and customization as per user needs and preferences. Besides, usability testing will be performed to ensure that the software interface developed for the smart mirror is user-friendly and intuitive in such a way that the usability of the smart mirror will be optimized based on design principles that are user-centric through in-depth usability testing.

3 Literature Review

Smart mirrors, which merge everyday convenience with advanced technology, represent the evolution of the traditional mirror into what is now referred to as ”magic mirrors.” These innovative devices integrate the classic functionality of a mirror with the dynamic capabilities of digital displays. They provide real-time updates such as weather, news, and personal schedules, seamlessly enhancing daily routines. Commonly used in both personal and public spaces, smart mirrors are a prime example of sophisticated Internet of Things (IoT) technology applications. They not only add practicality to daily life but also introduce a modern flair that aligns with contemporary lifestyles and the demands for multitasking and efficiency [4]. These mirrors are designed to address the challenges of modern life by allowing users to access a variety of information and control options, such as news and entertainment, while they prepare for the day, making daily routines more efficient

and enjoyable.

One of the standout applications of smart mirrors is in personal enhancement. For example, using a combination of Raspberry Pi 3 and Alexa voice service, researchers have developed Maleficent Mirrors that not only aid in personal grooming but also provide essential information like weather updates, thus simplifying daily routines [5]. In educational and entertainment environments, the magic mirror paradigm—an augmented reality (AR) system—enables unique interactive experiences. By combining a camera and display device, this system functions like a mirror, blending users' reflections with virtual objects. Fiducial markers on handheld and wearable items allow computer vision to recognize these objects and render related virtual elements according to the chosen theme. This setup facilitates an immersive experience that can be shared with multiple onlookers without the need for special equipment, such as head-mounted displays (HMDs) or tablet PCs, making it particularly suited for public spaces like museums and science centers [6].

Despite these advancements, many smart mirrors lose their core functionality as a reflective surface when turned off, presenting a significant gap in their design [2]. My project addresses this limitation by ensuring that my smart mirror retains its reflective properties in both its active and inactive states, allowing for a flawless blend of traditional aesthetics and modern technology. Additionally, my project includes upgrades in both hardware and software, ensuring that the smart mirror is not only cost-effective but also fully functional, meeting a wide range of user needs without compromising on quality or performance.

Security is a pivotal aspect of smart mirror technology. The innovative system, designed with Raspberry Pi 3 and a touch-enabled screen, incorporates advanced biometric authentication techniques, including voice and facial recognition, to restrict device access to registered users, thus enhancing personal data protection [7]. Additionally, the smart mirror acts as a multifaceted home security device. It offers intrusion detection through sophisticated background subtraction and frame difference approaches, sending alerts and photos of intruders to the administrator. Human monitoring is further implemented using machine learning techniques with YOLO and OpenCV, which send alerts if a person moves out of the camera's vision range [7]. In the broader context of smart homes and cities, these mirrors integrate seamlessly into IoT ecosystems, providing a cost-effective solution that displays essential information like weather, time, and news while maintaining high-end security across the entire system [8].

Furthermore, while many smart mirrors focus on delivering special functionalities, my project pushes the envelope by integrating cutting-edge technologies such as transparent OLED or reflective LCD. These additions aim to preserve the mirror's aesthetic value while enhancing its utility with interactive and personalized features, making the user experience more engaging and intuitive [9]. Further enhancing the versatility of smart mirrors, [10] introduces virtual fitting rooms that

allow users to design and try fashionable t-shirts. This innovative application leverages Augmented Reality (AR) and body pose tracking, integrating seamlessly with projectors and printers for tangible outputs. Progressing into the convergence of technology and beauty, [11] details an automatic personal makeup system embedded within a smart mirror. This system employs advanced Machine Learning (ML) and AI techniques to meticulously analyze facial features and tailor personalized makeup suggestions. Expanding on smart beauty solutions,[12] and [13] explore diverse applications for makeup. While [12] utilizes 3D technology for immersive virtual makeup experiences, [13] enhances user engagement by introducing a web page that allows friends to vote on makeup results, promoting interactive beauty choices. To ensure these features are efficiently implemented, the functionalities typically replace traditional mirrors by displaying mirror capabilities directly on screens.

In essence, the landscape of smart mirror research is filled with innovative solutions that enhance functionality across various aspects of daily life. Our project distinguishes itself by focusing on the seamless integration of technology with traditional aesthetics. Utilizing the Raspberry Pi 4, we have developed a smart mirror that retains the essential reflective capabilities of a conventional mirror. By prioritizing the preservation of this core functionality and ensuring the affordability of the hardware, our smart mirror is designed to be more accessible and appealing, making it an attractive new device for both personal and public domains.

4 Methodology

4.1 Tools and Instruments for Smart Mirror

4.1.1 Hardware Instruments

The Smart Mirror project integrates a Raspberry Pi 4 Model B microcomputer, an 11.6-inch Waveshare IPS HDMI LCD panel, and Raspberry Pi Camera Module V2 for visual input. It also includes a Raspberry Pi Mini Microphone for speech recognition, a loudspeaker system, and a wireless keyboard for development purposes. To ensure stability, the mirror will be encased in a wooden frame, and installation tools like adhesives and sealants will be used.

Raspberry Pi 4 Model B Microcomputer The study conducted by Dokrimare [14] employs Raspberry Pi to develop a Smart Mirror, leveraging its versatility as a single-board computer [15]. This project utilizes the Raspberry Pi 4 Model B microcomputer with a microSD card, which boasts a robust quad-core 1.5 GHz ARM Cortex-A72 processor, memory options ranging from 2 GB to 8 GB, enhanced graphics supporting dual 4K displays, Gigabit Ethernet, and USB-C power

supply. With significant improvements in processing power, memory capacity, and connectivity, the Raspberry Pi 4 serves as an ideal platform for various applications [16]. Figure 1 is the Raspberry Pi 4 Model B.

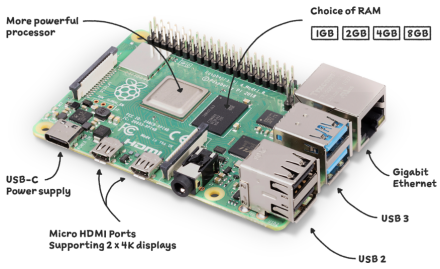

Figure 1: Raspberry Pi 4 Model B. Source: Online, [17]

Display Panel The 11.6-inch IPS HD screen, renowned for its high-quality IPS (In-Plane Switching) display technology, will be seamlessly incorporated behind the two-way mirror.

Figure 2: 11.6-inch IPS HD screen.

Bidirectional Lenses Translucent lenses of optical-grade quality will be employed to conceal electronic components behind the display. As detailed in the research conducted by Uddin and colleagues [18], they utilized a two-way mirror featuring transparency on one side and a reflective surface on the other. Specifically, the Spyglass Two-Way Acrylic Mirror will function as the reflective surface of the Smart Mirror, preserving its reflective characteristics even when the display is not in use.

Camera Module - USB Industrial Camera In [20], Mohamed et al. utilized a 5MP Raspberry Pi Camera for visual input. Nonetheless, in my project, I intend to employ the USB industrial camera. The primary distinctions between the USB Industrial Camera and the 5MP Raspberry Pi Camera lie in their resolution and lens design. Notably, the industrial camera module meets the

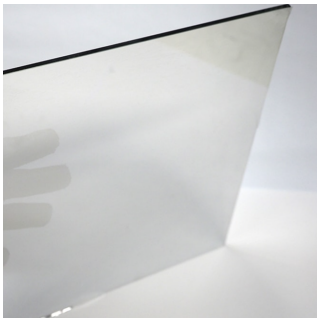

Figure 3: Two-way Mirror. Source: Online [19]

characteristics of being small, cheap, and powerful at the same time [21]. The industrial camera module offers a higher resolution and improved lens design, enhancing the quality and clarity of visual input. This upgrade is crucial for capturing finer details and achieving better overall performance in the project's objectives, whether it be in image recognition, surveillance, or any other application requiring precise visual data. Additionally, the industrial camera module may provide enhanced compatibility with other hardware components or software tools, further optimizing the project's functionality and efficiency.

Figure 4: USB Industrial Camera. Source: Online [22]

Raspberry Pi Mini Microphone The project uses a Raspberry Pi 4B driver-free speech recognition recording microphone. The characteristics of this microphone are: 1. 360° omnidirectional, anti-noise, and can be used as any device with a USB interface. 2. Portable and easy to use, no driver required.

USB Loudspeaker The Smart Mirror will incorporate a loudspeaker system to provide audio feedback and interactive responses to the user. This feature will be particularly useful for applications that involve multimedia content, such as voice control the system's interaction (system standby or information acquisition), and plays the user's favorite music list, making the smart mirror more engaging and user-friendly.

Figure 5: Raspberry Pi Mini Microphone. Source: Online [23]

Figure 6: USB Loudspeaker. Source: Online [24]

Wireless Keyboard Integrated into the development and configuration of the Smart Mirror, a wireless keyboard serves as an essential tool for developers to efficiently set up and program the device. This keyboard is not intended for end-user interaction but is crucial during the development phase for entering commands, coding, and navigating the system's software. It connects seamlessly via Bluetooth, providing a clutter-free environment that facilitates the rapid development and testing of the mirror's software components. This setup enhances the developer's ability to make adjustments and updates without the need for direct hardware connections.

Frame and Bracket To provide structural support and stability, the Smart Mirror will be encased in a customized wooden frame, ensuring durability, aesthetics, and longevity.

Additional Tools To ensure the secure attachment and perfection of hardware components, the Smart Mirror project will utilize a variety of installation and maintenance tools. This includes nail-free adhesives and silicone sealants, which are used for fixing and sealing components to ensure the stability and durability of the device. Additionally, tools such as nail guns will be employed for quick and precise assembly or repair of the mirror frame and bracket structures when necessary. These tools not only enhance the overall structural strength of the device but also help maintain its aesthetic and functional integrity during the development phase.

4.1.2 Software Instruments

The Smart Mirror project will leverage an extensive software stack to seamlessly integrate and oversee the diverse hardware components, ensuring a cohesive and intelligent user interface. Utilizing a variety of software tools, frameworks, and programming languages, the project aims to achieve this integration:

Operating System and Programming Language The Raspberry Pi 4B, runs the Raspbian OS, which functions effectively for various tasks, including operating computer systems and connecting to networks. The Raspbian OS is a Debian-based distribution optimized for Raspberry Pi. It provides a user-friendly interface and strong functionality.

MagicMirror² is an open-source modular smart mirror platform, which provides extensive customization and versatility for creating personalized smart mirror systems. During the configuration process of the smart mirror system, the 'apt' command, derived from the Advanced Packaging Tool (APT) system in Linux, is utilized for the efficient management of software packages. This command enables users to search for, install, upgrade, and remove packages seamlessly, ensuring the smooth operation of the smart mirror system.

In the architecture of the smart mirror system, all modules and configurations are developed using JavaScript, a versatile and widely used programming language. These modules are compiled through the 'config.js' file, which serves as a centralized configuration file for defining the behavior and appearance of various components within the smart mirror interface. JavaScript's flexibility and extensive ecosystem of libraries and frameworks facilitate the development of interactive and dynamic features, enhancing the functionality and user experience of the Smart Mirror system.

Node.js and npm Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a web browser. Built on Chrome's V8 JavaScript engine, Node.js enables developers to use JavaScript for server-side scripting, allowing the development of scalable and high-performance network applications[25]. One of the significant advantages of Node.js is its event-driven, non-blocking I/O model, which makes it efficient and suitable for real-time applications with high data throughput requirements.

npm (Node Package Manager) is the default package manager for Node.js and serves as a robust ecosystem for managing dependencies. It facilitates the installation, sharing, and updating of code modules, streamlining the development process. The npm registry hosts thousands of packages, which can be easily integrated into projects to extend functionality.

In the context of the Smart Mirror ecosystem, Node.js and npm play a crucial role in facilitating

the development and management of the Smart Mirror software. Node.js enables developers to leverage their JavaScript skills for server-side development, ensuring consistency in language usage across different layers of the smart mirror application. The event-driven nature of Node.js aligns well with the real-time nature of smart mirror interactions, allowing for responsive and interactive user experiences.

Additionally, npm provides a vast repository of reusable code modules that can be seamlessly integrated into the Smart Mirror project. This allows developers to incorporate advanced functionalities, such as weather updates, calendar events, and news feeds, with minimal effort. By utilizing Node.js and npm, the Smart Mirror project can efficiently install and manage dependencies, ensuring that the system remains modular, maintainable, and extensible over time.

Graphical User Interface (GUI) Development Graphical User Interface (GUI) Development refers to the process of developing a graphical user interface. A graphical user interface is an interface that allows users to interact with computer programs through graphical elements such as buttons, text boxes, menus, and so on. The goal of GUI Development is to create user-friendly interfaces that enable users to intuitively manipulate an application or system without having to understand the underlying technical details. Figure 1 shows the steps of developing the graphical user interface[26].

Smart Mirror's GUI Development includes the following aspects:

Module development: The core of Smart Mirror is a series of modules, each module is responsible for displaying specific information or providing specific functions. Developers can extend the functions of the mirror by writing their modules, including displaying weather, calendar, news, clock, and other information, or adding custom interactive functions.

Interface layout: Users can adjust the position, size, and arrangement of modules according to their own needs. Developers can define the layout of the interface through configuration files to achieve personalized display effects.

Interaction design: Smart Mirror supports multiple interaction methods, including touch screen, voice control, gesture recognition, etc. Developers can implement these interactive functions by adding corresponding modules to improve user experience.

APIs and Data Sources Smart Mirror systems extensively leverage Application Programming Interfaces (APIs) to fetch and display real-time data, thereby enhancing their functionality and relevance. APIs serve as intermediaries that allow different software applications to communicate with each other, facilitating the integration of external data sources into the smart mirror platform.

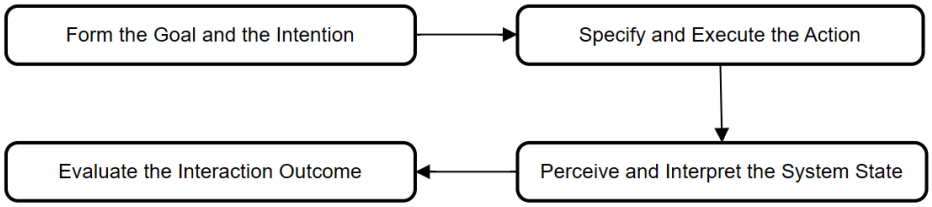

Figure 7: Steps for Graphical User Interface

For instance, weather APIs such as OpenWeatherMap provide current weather conditions, forecasts, and historical weather data. By integrating these APIs, the smart mirror can display accurate and up-to-date weather information to users. Similarly, calendar APIs like CalendarLabs offer access to various calendar data, including public holidays, events, and personal schedules, enabling the smart mirror to serve as an interactive calendar interface.

4.2 Procedures

4.2.1 Hardware Splicing

Prepare the Mirror and Display Panel Begin by preparing a transparent two-way mirror, ensuring one side is transparent while the other is reflective. Then mount the 11.6-inch Waveshare IPS HDMI LCD panel behind the mirror. Ensure the display panel fits perfectly with the mirror and is secured in the appropriate position.

Install the Raspberry Pi 4 Model B Microcomputer Place the Raspberry Pi 4 Model B in a suitable position, typically behind the display panel. Then Connect the power supply and other necessary external devices, such as a mouse, and keyboard, to the Raspberry Pi 4 Model B.

Mount the Raspberry Pi Camera Module V2 Install the Raspberry Pi Camera Module V2 in the appropriate position on the display panel to ensure it can capture the desired angle and images. Connect the Raspberry Pi Camera Module V2 to the Raspberry Pi 4 Model B, and install 'cheese' to ensure the camera can work properly with the system.

Integrate the Raspberry Pi Mini Microphone and Loudspeaker Install the Raspberry Pi Mini Microphone near the display panel to capture the user's voice input. Mount the Loudspeaker near the display panel to provide audio feedback and interactive responses.

Add the Wireless Keyboard Pair the wireless keyboard with the Raspberry Pi 4 Model B and ensure it can connect and function properly. Place the wireless keyboard in an easily accessible po-

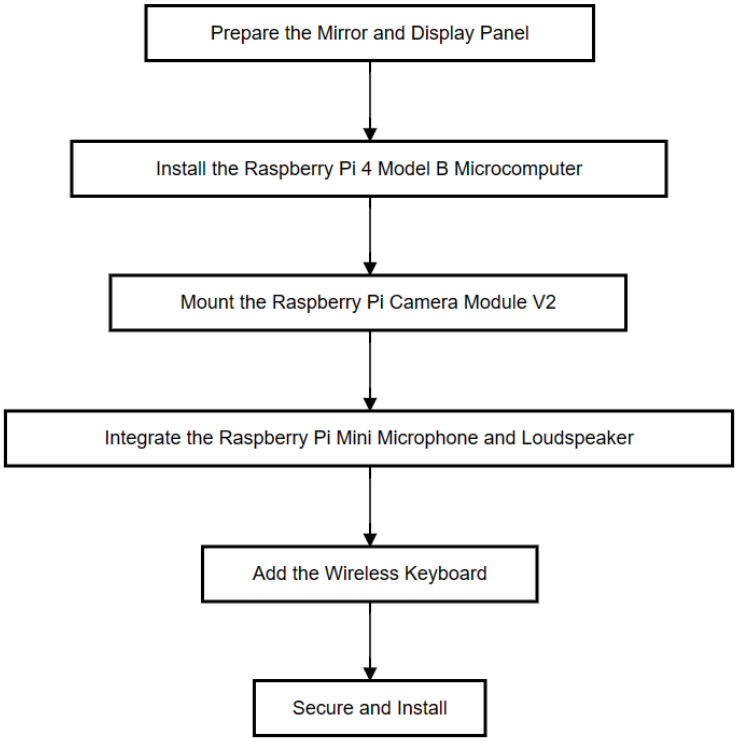

Figure 8: Steps for Hardware Splicing

sition during the development and configuration of the smart mirror, facilitating input of commands and system settings by developers.

Secure and Install Use a customized wooden frame to secure all hardware components together, ensuring the entire smart mirror structure is stable and robust. Utilize appropriate installation tools such as nail-free adhesives and silicone sealants to secure each hardware component, ensuring stability and durability.

Figure 9: Front and back of Prototype

4.2.2 Software Installation and Configuration

Install Raspberry Pi OS using Raspberry Pi Imager Download and install Raspberry Pi Imager from the official website. Choose the latest Raspberry Pi OS version and write it to the SD card using the Imager software. Insert the SD card into the Raspberry Pi, connect the peripherals, and power it on. Follow on-screen instructions to finish setup, selecting language, time zone, etc.

Install the Smart Mirror System Smart Mirror is an open-source system that can run on any successfully installed device and has basic modules. Figure 10 shows the installation steps of the Smart Mirror system.

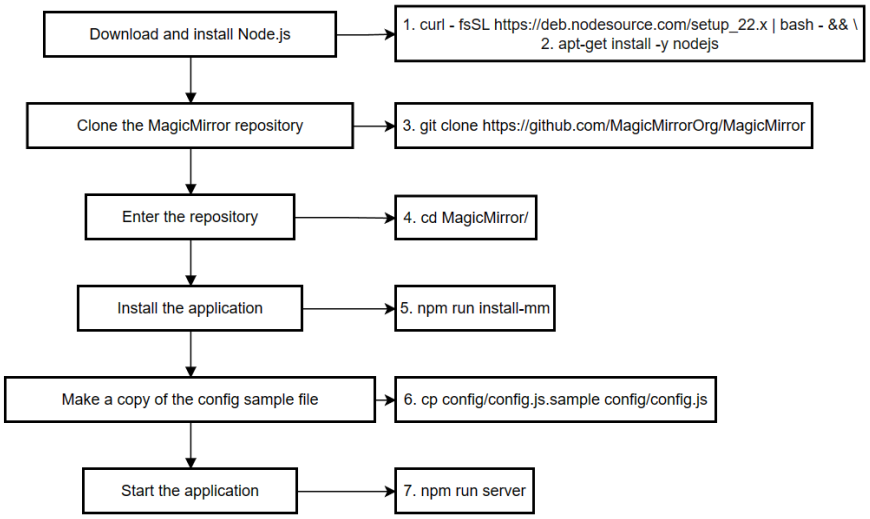

Figure 10: How to install the Smart Mirror System

Autostart the Smart Mirror - PM2 PM2 is a Node.js application manager designed for production environments, featuring a built-in load balancer. It ensures continuous uptime for applications, facilitates seamless reloading without disruptions, and simplifies various system administration tasks. In this scenario, PM2 will be utilized to maintain the execution of a shell script. Figure 11 shows the specific installation methods.

Module Configuration As mentioned above, the newly installed Smart Mirror system already has installed basic modules. They are Alert, Calendar, Clock, Compliments, News Feed, Update Notification, and Weather Module. In my project, I need to improve these existing modules to make them more suitable for users in China. All the modules can customize the location of the module in which the module will be loaded, the animated features, and more detailed information. One of

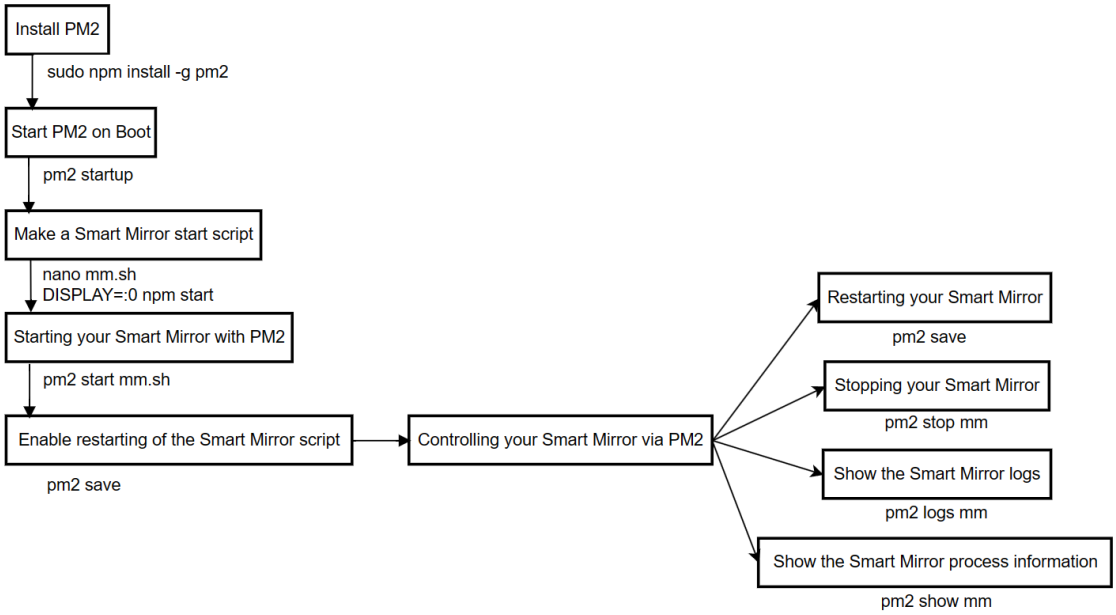

Figure 11: Installation Procedure

```
yuesiyu@raspberrypi:~ $ pm2 start mm.sh
(node:126732) [DEP0044] DeprecationWarning: The `util.isArray` API is deprecated. Please use `Array.isArray()` instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
[PM2] Applying action restartProcessId on app [mm](ids: [ 0 ])
[PM2] [mm](0) ✓
[PM2] Process successfully started
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
0	mm	default	N/A	fork	126739	0s	0	online	0%	3.0mb	yuesiyu	disabled

```
yuesiyu@raspberrypi:~ $ pm2 stop mm
(node:127327) [DEP0044] DeprecationWarning: The `util.isArray` API is deprecated. Please use `Array.isArray()` instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
[PM2] Applying action stopProcessId on app [mm](ids: [ 0 ])
[PM2] [mm](0) ✓
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
0	mm	default	N/A	fork	0	0	0	stopped	0%	0b	yuesiyu	disabled

Figure 12: The actual operation of Autostart

the smart mirror's default modules is (1) the alert module. Notifications from other modules are shown in this module. We do not need to change any configuration.

(2) The calendar module needs to be changed to the Chinese holiday arrangement. I use the specific URL from CalendarLabs which shows Chinese holidays. In this way, the user can always check the information of the upcoming vacation, and it is convenient to arrange the trip later.

(3) The clock module can add many features such as "showSunTimes", "secondsColor", "timeFormat", and so on. Users can modify them according to their preferences.

(4) The compliments module displays a random compliment. The module can change the words, even though show specific compliments in different periods.

(5) The news feed module is located in the bottom bar of the screen. The setting is to change the news points displayed at regular intervals, and the news source is replaced by the XML file provided by China People's Daily Online. XML (Extensible Markup Language) is a markup language that structures data to be easily readable and structured. This forms the basis for many data interchange formats, such as those used in RSS and Atom. The use of XML files as a source of news maintains concurrence and standardization of the data format, and thus, it is easy to parse and process. The XML files are clear in their hierarchy and labeled so that the programming languages can parse them and extract whatever information they want with ease. For instance, the programming languages Python, JavaScript, and many others have quite advanced libraries and tools for the parsing of XML data into news headlines, descriptions, links, and so forth. It is very flexible and extensible and can describe complex data structures. This then comes in handy in the news feed part of the Smart Mirror, where the addition or modification of data fields like publication date, author, category, and so forth is easy without breaking the overall data structure of the XML files [27]. In addition, the system is also equipped the update notifications. This will display a message whenever a new version of the MagicMirror application is available.

(6) The next part is the weather module, which is about the weather information. I have divided it into two parts, one is the real-time temperature and sunset information, and the other is the weather forecast information for the next 5 days. The source of the information is OpenWeatherMap, and after applying the API based on the location, you can see the local weather information. It is much more accurate and easier to get information from more specialized information providers than from sensors, such as temperature sensors, or other devices.

The above is the adjustment and modification of the default module of the smart mirror system, and the following is the third-party module that I added and configured according to the needs of my project.

(7) Although this smart mirror is not used with a touchable screen, I hope that the use of the mirror can be controlled by an intelligent voice, the function may not be complicated, but it is a good foundation for the development of more complex functions in the future. As I mentioned above, the system uses the MMM-GoogleAssistent module to realize the voice control function. This module can make conversations with Google Assistant, control your CPU, open links discovered by an assistant, etc. After installing the module easily, we need to do the credentials setup, which is the procedure that can connect the voice assistant to the online platform. The detailed steps are shown in Figure 13. Then there's the part where we configure the voice assistant module. In addition to the basic functionality of the configuration, the developers have provided us with several extended plugins to improve the functionality of the voice assistant, such as EXT-Alert, EXT-Browser, and EXT-Motion, etc.

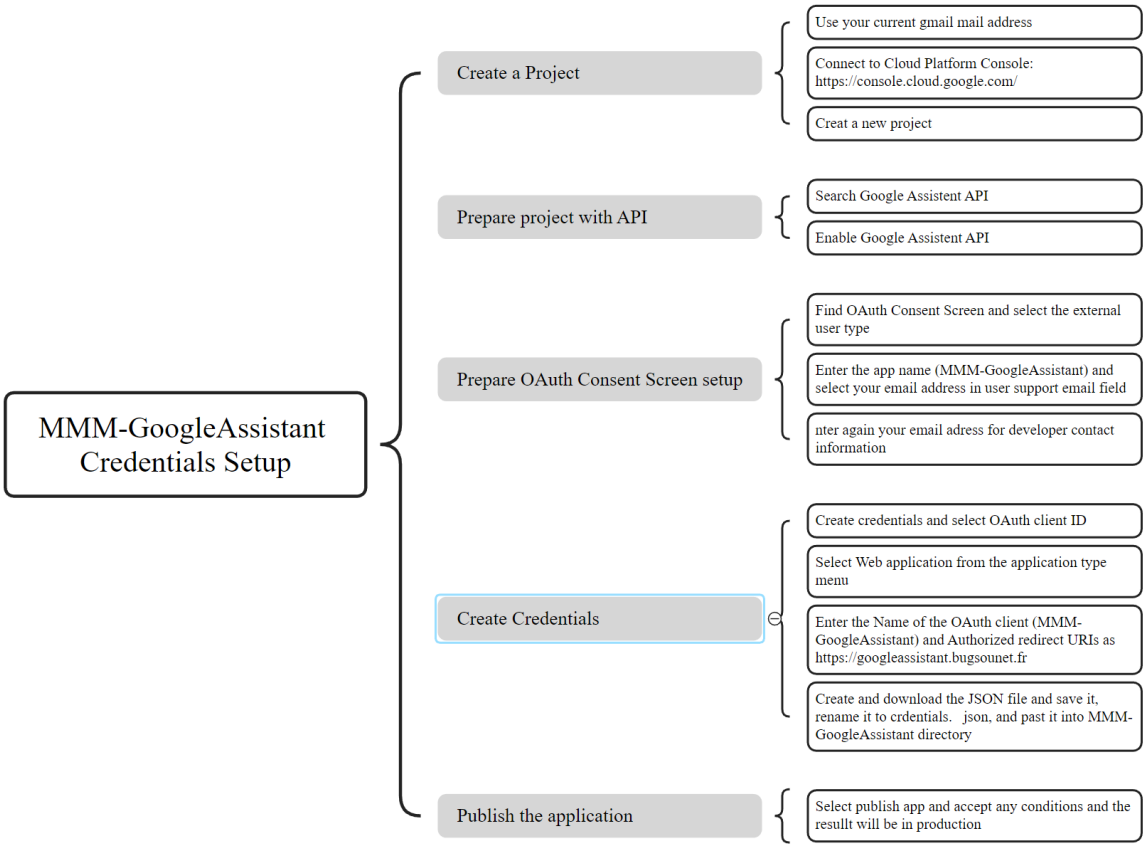

Figure 13: Credentials Setup

So far all the internal configurations have been completed, and now all we need to finish is the operation before the first launch of MMM-GoogleAssistant - generate the Google Assistant token: The process of getting a Google Assistant token with the help of the MMM-GoogleAssistant module is colorfully described in the provided documentation. It is important because this peculiar procedure lets the Google Assistant service work with the module, which ensures the ability of users to use Google Assistant functionality accurately in the smart mirror environment. Users are prompted to go to the MMM-GoogleAssistant directory. The latter is in the smart mirror modules folder of the Raspberry Pi and can be accessed in the terminal. The second command will result in invoking a further process to achieve the creation of the token by the usage of the ‘npm run token‘. This command produces several prompts that the user interacts with, which eventually sets up this token. The first of these prompts inquires whether the user would like to install or reinstall the Google Assistant token. The expectation for the user is to respond positively, with ’Y’, which is an abbreviation for yes. Then the process will open an OAuth URL toward which the user has to authenticate with the Google account. If the browser is not opened automatically with the URL, the user should manually copy the link and paste it into their browser. Google sends back a code that should be ideally entered in the command prompt. Following the entry of the code,

the system reports to the user that their Google Assistant token has been created by outputting the message "[GA] Token created!" The procedure confirms that the module MMM-GoogleAssistant is perfectly connected with the services of Google. This produces proper security and functional integration, providing users with the full power of Google Assistant within the interface of the smart mirror. Finally, it completes with the message "Done." This means that the setup process is over, and the module is ready to function. This process gives proper security and functional integration, providing users with the full power of Google Assistant within a smart mirror interface.

(8) Taking into account people's different preferences, I added a third-party module called Spotify to this smart mirror system, which is a popular music app nowadays. After installing the basic Spotify files on the system, we need to set up Spotify in the following steps: To use the Spotify API in your application, visit the Spotify Developer website at https://developer.spotify.com and log in to the dashboard. From the 'DASHBOARD', navigate and click on 'Create an app'. Fill the dialog form with the application's details—name and description—then click 'EDIT SETTINGS'. One of the key settings in this setup is setting the Redirect URIs to 'http://localhost:8888/callback' to catch callback responses when testing. Make the configuration and save those changes. Finally, note down the details as 'Client ID' and 'Client Secret'. Use these credentials to authenticate requests with Spotify; they should remain safely stored in the development environment only. At this point, the system can now interact with the Spotify API. The music program can be used normally, and other more detailed configurations can be customized according to personal preferences.

(9) To install the MMM-Face-Recognition-SMAI module for the smart mirror, first, navigate to the smart mirror modules directory on the Raspberry Pi and clone the MMM-Face-Recognition-SMAI repository. After cloning, change to the newly created module directory with 'cd MMM-Face-Recognition-SMAI' and install the required dependencies. Following the installation of dependencies, download and configure the necessary face recognition models as per the instructions provided in the module's README file. Additionally, configure the 'config.js' file in the smart mirror directory to include the MMM-Face-Recognition-SMAI module with the appropriate settings. Finally, restart the smart mirror to apply the changes and activate the face recognition functionality. This installation process equips the smart mirror with advanced facial recognition capabilities, enhancing its interactive features.

The smart mirror project encompasses a comprehensive process of hardware integration and software configuration to develop a multifunctional device with diverse capabilities. Beginning with hardware splicing, components such as the Raspberry Pi, display panel, camera module, microphone, and speaker are meticulously assembled to form a stable and robust structure. Subsequent software installation and configuration involve tasks like setting up the Raspberry Pi OS, installing the Smart Mirror system, and customizing modules for specific user preferences, in-

cluding localization for Chinese users and the integration of third-party modules like MMM-GoogleAssistant and Spotify for extended functionalities. Overall, through a combination of hardware assembly, software setup, and module customization, the project culminates in the creation of a sophisticated smart mirror system that not only serves practical functions like displaying weather updates and calendar events but also offers enhanced user interaction and entertainment options, poised to become a versatile addition to smart home environments.

5 Achievement Exhibition

Hardware and software components for the Smart Mirror have been brought together, implemented, and individually implemented in an innovative and interactive device. This chapter presents the progress of the project by introducing the functions that are implemented accordingly and discusses the areas needed for further work. The detailed implementation of each feature serves as an illustration showing the practical application brought about through the seamless integration of technology with the system of the Smart Mirror. Figure 14 shows the application interface of the whole smart mirror.

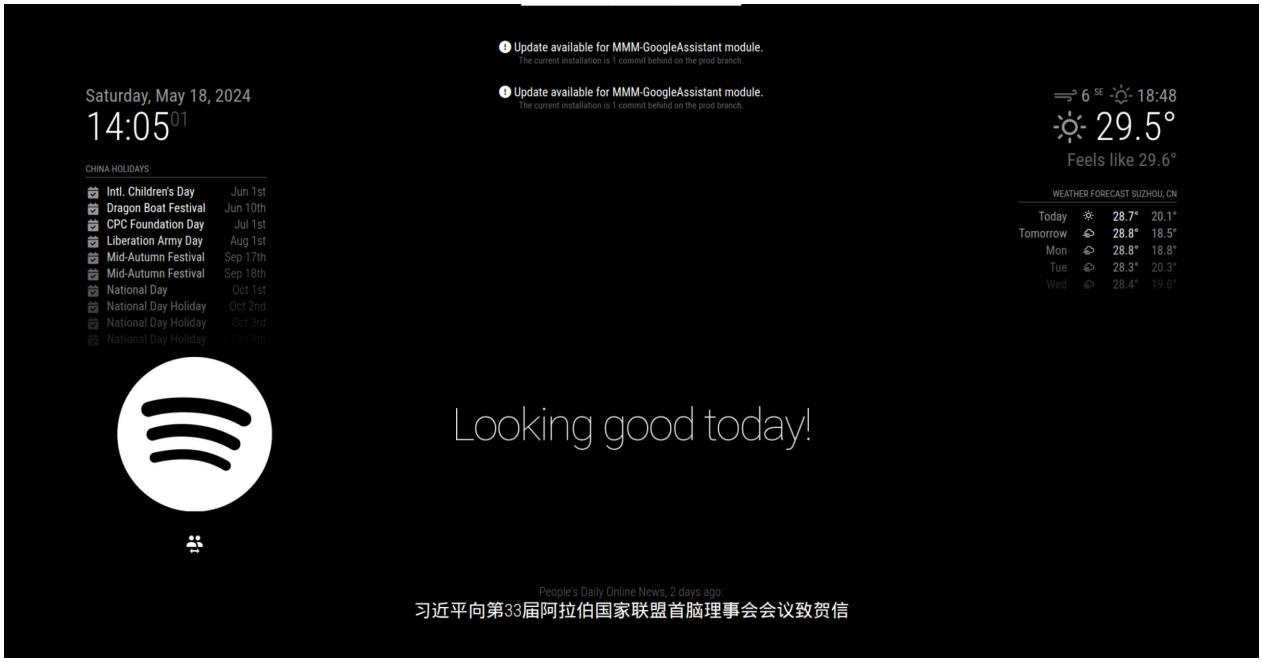

Figure 14: The Smart mirror display interface

5.1 Implemented Features

5.1.1 Calendar Module

The module is for viewing the date at present and the events in the future, according to the holidays in China. This is to ensure that the user receives the right information regarding the holidays, in order to help them plan and schedule. It gets its information from CalendarLabs and is updated from time to time, thereby making it up to date. Figure 15 is the configure code of this module.

```
module: "calendar",
header: "China Holidays",
position: "top_left",
config: {
        calendars: [
                {
                        fetchInterval: 7 * 24 * 60 * 60 * 1000,
                        symbol: "calendar-check",
                        url: "https://ics.calendarlabs.com/40/6e3db494/China_Holidays.ics"
                }
        ]
}
```

Figure 15: Code for the calendar module

5.1.2 Clock Module

The clock module provides real-time updates on the current time, including additional features such as showing sunrise and sunset times. Users can customize the display format and color settings according to their preferences. The module seamlessly blends with the mirror's aesthetics, ensuring it looks like a conventional mirror when not actively displaying information. The code of the clock module is simple, with no special configuration.

5.1.3 Compliments Module

The compliments module adds a personal touch to the application by showing random compliments at different points in the day. If the user wishes, they can set these compliments to be shown based on time, which should make the messages appropriate and encouraging. This module personalizes the user experience by adding a positive, motivational aspect to daily life. Figure 16 shows the detailed configuration of compliments.

5.1.4 News Fees Module

The news feed module displays current news headlines from China People's Daily Online. The use of XML files allows for structured data presentation, ensuring that the latest news is readily available to users. The module updates frequently to provide up-to-date news, keeping users informed.

```
Module.register("compliments", {
	// Module config defaults.
	defaults: {
		compliments: {
			anytime: ["Have a magical day!"],
			morning: ["Good morning!", "Enjoy your day!", "How was your sleep?"],
			afternoon: ["Hello, beauty!", "Looking good today!"],
			evening: ["You look nice!", "Hard work today!"],
			"....-01-01": ["Happy new year!"]
		},
		updateInterval: 6000,
		remoteFile: null,
		fadeSpeed: 4000,
		morningStartTime: 3,
		morningEndTime: 12,
		afternoonStartTime: 12,
		afternoonEndTime: 17,
		random: true
	},
	lastIndexUsed: -1,
	// Set currentweather from module
	currentWeatherType: "",

	// Define required scripts.
	getScripts () {
		return ["moment.js"];
	},

	// Define start sequence.
	async start () {
		Log.info(`Starting module: ${this.name}`);

		this.lastComplimentIndex = -1;

		if (this.config.remoteFile !== null) {
			const response = await this.loadComplimentFile();
			this.config.compliments = JSON.parse(response);
			this.updateDom();
		}

		// Schedule update timer.
		setInterval(() => {
			this.updateDom(this.config.fadeSpeed);
		}, this.config.updateInterval);
	},
```

Figure 16: Configuration of compliments module

5.1.5 Weather Module

The weather module provides comprehensive weather updates, including current temperature, sunset times, and a 5-day weather forecast. Integration with OpenWeatherMap ensures accurate and real-time weather information. This module is essential for users to plan their day effectively.

5.1.6 Spotify Integration

The smart mirror includes a Spotify module, allowing users to play their favorite music directly from the mirror. This integration enhances the user experience by providing entertainment options seamlessly. Users can log in to their Spotify account from the beginning setting and control playback through the mirror interface.

```
{
        module: "newsfeed",
        position: "bottom_bar",
        config: {
                feeds: [
                        {
                                title: "People's Daily Online News",
                                url: "http://www.people.com.cn/rss/politics.xml"
                        }
                ],
                showSourceTitle: true,
                showPublishDate: true,
                broadcastNewsFeeds: true,
                broadcastNewsUpdates: true
        }
},
```

Figure 17: Configuration of news feed module

```
{
        module: "weather",
        position: "top_right",
        config: {
                weatherProvider: "openweathermap",
                type: "current",
                location: "Suzhou",
                locationID: "1886760",
                apiKey: "ba7276c5b5a754c98590919c7eb438df"
        }
},
{
        module: "weather",
        position: "top_right",
        header: "Weather Forecast",
        config: {
                weatherProvider: "openweathermap",
                type: "forecast",
                location: "Suzhou",
                locationID: "1886760",
                apiKey: "ba7276c5b5a754c98590919c7eb438df"
        }
},
```

Figure 18: Configuration of news feed module

5.1.7 Update Notification Module

The update notification module tells the user exactly every time a new version of the MagicMirror application has been released. It checks for them automatically; if one is available, a message will pop on the screen of the mirror, allowing the users to bring their system up to date with the latest features and security patches.

6 Analysis of Cost-Effective and User-Friendly Smart Mirror

6.1 Cost-Effective Hardware Choices

The Smart Mirror project considers cost-effective hardware without compromising the performance and functionality of the mirror. The project deals with the utilization of Raspberry Pi 4 Model B hardware powerful and, at the same time, relatively cheap microcomputer hardware to see how performance can be squeezed from it at a low cost. It is a very good choice to deal with processing

```
{
        module: "MMM-Spotify",
        position: "bottom_left",
        config: {
                debug: false, // debug mode
                style: "default", // "default" or "mini" available (inactive for miniBar)
                moduleWidth: 360, // width of the module in px
                control: "default", // "default" or "hidden"
                showAlbumLabel: true, // if you want to show the label for the current song album
                showVolumeLabel: true, // if you want to show the label for the current volume
                showAccountName: false, // also show the current account name in the device label; usefull for multi account setup
                showAccountButton: true, // if you want to show the "switch account" control button
                showDeviceButton: true, // if you want to show the "switch device" control button
                useExternalModal: false, // if you want to use MMM-Modal for account and device popup selection instead of the build-in one (which is restricted to the album image size)
                updateInterval: 1000, // update interval when playing
                idleInterval: 30000, // update interval on idle
                defaultAccount: 0, // default account number, attention : 0 is the first account
                onStart: {
                                deviceName: "yuesiyu",
                                spotifyUri: "spotify:track:2UE9XGbAzicJIyo4bB6sqM",
                                },
                deviceDisplay: "Listen on",
                volumeStep: 5,
                miniBarConfig:{
                                album: true,
                                scroll: true,
                                logo: true,
                        },

                },
},
```

Figure 19: Configuration of Spotify module

```
    {
            module: "updatenotification",
            position: "top_center", // This can be any of the regions.
            config: {
                    updates: [
                    // array of module update commands
                            {
                                    // update of MMM-Test with embed npm script
                                    "MMM-Test": "npm run update",
                            },
                            {
                                    // update of MMM-OtherSample with "complex" process command
                                    "MMM-OtherSample":
                                    "rm -rf package-lock.json && git reset --hard && git pull && npm install",
                            },
                            {
                                    // update of MMM-OtherSample2 with git pull && npm install command
                                    "MMM-OtherSample2": "git pull && npm install",
                            },
                            {
                                    // update of MMM-OtherSample3 with a simple git pull
                                    "MMM-OtherSample3": "git pull",
                            },
                    ],
            },
    },
```

Figure 20: Configuration of update notification module

needs for Smart Mirror. Compared to the Raspberry Pi 3 used in [5], the Raspberry Pi 4 has a quad-core ARM Cortex-A72 processor, 8 GB of RAM, and robust connectivity.

Besides, they have been chosen for the cost-efficient qualities they bear, for instance, the 11.6-inch IPS HDMI LCD Panel, the USB industrial camera, or the mini microphone. For instance, the inclusion of an IPS panel will ensure that what is displayed is of quality and comes with the benefit of wide-viewing angles, while the inclusion of the USB industrial camera will cater to quality and the assurance of quality during the acquisition of visual data. The last one, the mini microphone, was selected for quality and to create a clear audio input for the voice command system and compatible with the Raspberry Pi to provide voice interaction properties. Their selection thus makes the overall cost of the smart mirror well within a reasonable amount, thus reaching a bigger audience while keeping.

The implementation of the smart mirror primarily hinges on essential components such as a

mirror, Raspberry Pi, and an appropriately sized display, barring extensive configuration demands. Emphasis was placed on selecting cost-effective hardware components while maintaining optimal performance to ensure affordability for a broader user base. Each component was meticulously chosen based on the criteria of cost-effectiveness and operational efficiency. The modular design significantly simplified both installation and maintenance processes, catering to users with limited technical expertise. Moreover, the user interface was designed to provide a seamless and intuitive interaction, featuring customizable modules tailored to individual preferences.

6.2 Simplified Installation and Maintenance

It has been designed to highlight a clean process of installation and maintenance. The modular components and step-by-step assembly instructions make it very easy to assemble and maintain a smart mirror, needing only some simple technical skills. This includes mounting the display panel behind the two-way mirror, along with the Raspberry Pi and other hardware, and adjusting these components properly within a customized wooden frame. All designed components are meant to be accessible so that troubleshooting or upgradation is not tiresome.

Moreover, the use of tools like Raspberry Pi Imager and PM2, which manage the MagicMirror application, eases the installation process of software. The installation of the Raspberry Pi OS is made incredibly easy with the Raspberry Pi Imager. PM2 maintains that the MagicMirror software is running and can easily be restarted in case of a failure. It lowers the barrier of users who do not have so much technical skill, making the smart mirror interesting and useful.

6.3 User-Friendly Interface

The interface design of the smart mirror is intuitive and user-friendly. The integration of various modules such as calendar, clock, weather, and news feed provides users with important information in a clear and organized way. The calendar module displays important dates and events, which can be customized to display Chinese holidays. The clock module provides real-time updates and additional information such as sunrise and sunset times. Praise modules deliver uplifting messages throughout the day, enhancing the user experience with positive reinforcement.

The ability to customize the layout and appearance of these modules ensures that users can customize the smart mirror to their liking. Users can choose the position, size, and style of each module to create a personalized interface that suits their needs.

6.4 Enhanced Modular Customization

The system design is a modular smart mirror full of flexibility in customization and scalability to a large base of usage. The user is left to add or remove modules as per their requirements so as to ensure the mirror scales up according to the changing likings and preferences of the user. MagicMirror² is an open-source platform that encourages the development and sharing of new modules within the community to provide users with an ever-expansive array of new features and functionalities.

7 Conclusion

The Smart Mirror Project has been successful in integrating sophisticated hardware and software elements into a versatile and interactive tool. Moreover, the project gives due emphasis to low-cost hardware options like the Raspberry Pi 4 Model B, which ensures excellent performance at a low price, towards making the smart mirror more accessible. Easy installation and minimal maintenance, coupled with a user-friendly interface, would further enhance interest and utility in the smart mirror.

The developed mirror contains a calendar, clock, compliment, news feed, weather, and Spotify integration, which gives the user the necessary information and options for entertainment. For instance, the update notification module will inform the user about new versions of the smart mirror application that have been recently released so that the system is relevant and secure.

However, some features, like voice control and facial recognition, would require more work. The voice control feature, while properly set up with the right generation of the tokens, does not operate inside the smart mirror interface. Also, the facial recognition system has not been fully implemented due to time constraints.

These will now be ironed out with future work, which will include debugging and thorough testing of the voice control feature, integrating the feature for facial recognition, continuously improving the user interface, and adding more modules to increase the versatility of the smart mirror. In this way, the development will come to ensure the level of efficiency in meeting all needs brought by the user, bringing a good fulfilling customer experience to the user using the smart mirror. Consequently, the project, on the whole, has been a revelation showing the possibility of uniting traditional aesthetics with the application of modern technology in the creation of a versatile, user-friendly smart mirror serving the purpose of weakly ameliorating routine life and promising information and entertainment, both personalized and in real-time.

8 Future Work

The smart mirror project has made significant progress in integrating many advanced hardware and software feature sets to arrive at one multifunctional, interactive device. Even though most features run in full from the best interface, functions such as voice and face recognition are among those that need further work. Of further work, correction of these flaws, advancing smart mirror capabilities and ensuring that it serves the purpose of the user.

Future work will include: (1) Debugging and Testing Voice Control: MMM-GoogleAssistant is properly installed in the system and works, but there is no way to interact. Further investigation is needed to identify and resolve the issues preventing the MMM-GoogleAssistant API from functioning correctly. This will involve checking for potential conflicts, ensuring proper configuration, and possibly updating the software to the latest versions.

(2) Completing Facial Recognition Integration: Implementing and fine-tuning the MMM-Face-Recognition-SMAI module will be a priority. MMM-Face-Recognition-SMAI works on the back of OpenCV face recognition module. This will involve working with several users' data to train the facial recognition system to ensure it identifies users properly and provides personalized interaction.

(3) Enhancing User Interface: Infuse user interface improvements in the system continuously so as to make it more intuitive and friendly. In this regard, user feedback can be obtained to iteratively update the design and functionality.

(4) Adding More Modules: After the smart mirror can interact with the camera normally, expand the range of modules available to users, including fitness tracking, health monitoring, and more advanced home automation features. This will make the smart mirror even more versatile and useful in daily life.

References

[1] B. Cvetkoska, N. Marina, D. C. Bogatinoska, and Z. Mitreski, "Smart mirror E-health assistant — Posture analyze algorithm proposed model for upright posture," in *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, Ohrid, Macedonia: IEEE, Jul. 2017, pp. 507–512. DOI: 10.1109/EUROCON.2017.8011163.

[2] A. S. M. M. Rahman, T. T. Tran, S. A. Hossain, and A. El Saddik, "Augmented Rendering of Makeup Features in a Smart Interactive Mirror System for Decision Support in Cosmetic Products Selection," in *2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications*, Fairfax, VA, USA: IEEE, Oct. 2010, pp. 203–206. DOI: 10.1109/DS-RT.2010.30.

[3] C. Cae-Li, M. E. Rana, and K. Shanmugam, "Recommendations for Developing an IoT based Smart Mirror with Gesture Control," in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, Chiangrai, Thailand: IEEE, Mar. 2022, pp. 650–659. DOI: 10.1109/DASA54658.2022.9764996.

[4] S. Bianco, L. Celona, and P. Napoletano, "Visual-based sentiment logging in magic smart mirrors," in *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin: IEEE, Sep. 2018, pp. 1–4. DOI: 10.1109/ICCE-Berlin.2018.8576217.

[5] S. S. Sarnin, A. Akbar, W. N. W. Mohamad, A. Idris, N. F. Naim, and N. Yaracob, "Maleficent Mirror with ALEXA Voice Services as an Internet of Things Implement Using Raspberry Pi 3 Model B," in *TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea (South): IEEE, Oct. 2018, pp. 1202–1207. DOI: 10.1109/TENCON.2018.8650106.

[6] W. R. Sherman, M. C. Lin, A. Steed, and IEEE Computer Society. Technical Committee on Visualization and Graphics, Eds., *IEEE virtual reality 2007: proceedings: Charlotte, North Carolina, March 10-14, 2007*. Piscataway, N.J: IEEE, 2007.

[7] R. A. Nadaf, R. M, S. P, and V. M. Bonal, "Smart Mirror Using Raspberry Pi for Human Monitoring and Intrusion Detection," in *2019 1st International Conference on Advances in Information Technology (ICAIT)*, Chikmagalur, India: IEEE, Jul. 2019, pp. 116–121. DOI: 10.1109/ICAIT47043.2019.8987294.

[8] A. C. Njaka, N. Li, and L. Li, "Voice Controlled Smart Mirror with Multifactor Authentication," in *2018 IEEE International Smart Cities Conference (ISC2)*, Kansas City, MO, USA: IEEE, Sep. 2018, pp. 1–8. DOI: 10.1109/ISC2.2018.8656932.

[9] K. Jin, X. Deng, Z. Huang, and S. Chen, "Design of the Smart Mirror Based on Raspberry PI," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, Xi'an: IEEE, May 2018, pp. 1919–1923. DOI: 10.1109/IMCEC.2018.8469570.

[10] D. Saakes, H.-S. Yeo, S.-T. Noh, G. Han, and W. Woo, "Mirror Mirror: An On-Body T-shirt Design System," en, in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA: ACM, May 2016, pp. 6058–6063. DOI: 10.1145/2858036.2858282.

[11] T. V. Nguyen and L. Liu, "Smart Mirror: Intelligent Makeup Recommendation and Synthesis," en, in *Proceedings of the 25th ACM international conference on Multimedia*, Mountain View California USA: ACM, Oct. 2017, pp. 1253–1254. DOI: 10.1145/3123266.3127926.

[12] A. S. M. M. Rahman, T. T. Tran, S. A. Hossain, and A. El Saddik, "Augmented Rendering of Makeup Features in a Smart Interactive Mirror System for Decision Support in Cosmetic Products Selection," in *2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications*, Fairfax, VA, USA: IEEE, Oct. 2010, pp. 203–206. DOI: 10.1109/DS-RT.2010.30.

[13] E. Iwabuchi, M. Nakagawa, and I. Siio, "Smart Makeup Mirror: Computer-Augmented Mirror to Aid Makeup Application," en, in *Human-Computer Interaction. Interacting in Various Application Domains*, J. A. Jacko, Ed., vol. 5613, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 495–503. DOI: 10.1007/978-3-642-02583-9_54.

[14] S. Dokrimare, "A Review Paper Design of a Smart Mirror using Raspberry Pi," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, no. 4, pp. 1407–1410, Apr. 2019. DOI: 10.22214/ijraset.2019.4253.

[15] Y. Sun, L. Geng, and K. Dan, "Design of Smart Mirror Based on Raspberry Pi," in *2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, Xiamen, China: IEEE, Jan. 2018, pp. 77–80. DOI: 10.1109/ICITBS.2018.00028.

[16] H. Han, K. Kim, S. Jung, and Y. Chang, "Performance Evaluation on the Parallel Processing System with the Raspberry Pi 4," in *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA: IEEE, Dec. 2022, pp. 1372–1374. DOI: 10.1109/CSCI58124.2022.00246.

[17] Raspberrypi. "Raspberry pi 4 model b." Accessed on: June 2, 2024. (), [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[18] K. M. M. Uddin, S. K. Dey, G. U. Parvez, A. S. Mukta, and U. K. Acharjee, "MirrorME: Implementation of an IoT based smart mirror through facial recognition and personalized information recommendation algorithm," en, *International Journal of Information Technology*, vol. 13, no. 6, pp. 2313–2322, Dec. 2021. DOI: 10.1007/s41870-021-00801-z.

[19] "11.6-inch ips hd screen." Accessed on: June 2, 2024. (), [Online]. Available: https://www.twowaymirrors.com/wp-content/uploads/2018/07/two-way-mirror-film-applied-to-clear-glass-finished.jpg.

[20] A. S. A. Mohamed, M. N. A. Wahab, S. S. Suhaily, and D. B. L. Arasu, "Smart Mirror Design Powered by Raspberry PI," en, in *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference*, Tokyo Japan: ACM, Dec. 2018, pp. 166–173. DOI: 10.1145/3299819.3299840.

[21] R. W. Bowman, B. Vodenicharski, J. T. Collins, and J. Stirling, "Flat-Field and Colour Correction for the Raspberry Pi Camera Module," en, *Journal of Open Hardware*, vol. 4, no. 1, p. 1, Apr. 2020. DOI: 10.5334/joh.20.

[22] "Usb industrial camera." Accessed on: June 2, 2024. (), [Online]. Available: https://ae01.alicdn.com/kf/HTB1KX.GkQomBKNjSZFqq6xtqVXaU/MJPEG-260fps-640X360-USB-Industrial-camera-MJPEG-120fps-720P-60fps-1080P-CMOS-OV4689-Camera-Module-170.jpg.

[23] "Raspberry pi mini microphone." Accessed on: June 2, 2024. (), [Online]. Available: https://www.walmart.com/ip/TureClos-USB-Mini-Microphone-Driver-Free-Mic-for-Laptop-Computer-Recording-Equipment-Device-White/537113808.

[24] "Usb loudspeaker." Accessed on: June 2, 2024. (), [Online]. Available: https://ae01.alicdn.com/kf/HTB1fhe7SXXXXXaPXVXXq6xXFXXXw/ANENG-1-pair-3W-4-Ohm-3070-Advertising-LCD-TV-Speakers-Loudspeaker-Rectangle-Speaker.jpg.

[25] M. Wandschneider, *Learning Node.js: a hands-on guide to building Web applications in JavaScript* (Addison-Wesley learning series). Upper Saddle River, NJ: Addison-Wesley, 2013, OCLC: ocn824734082.

[26] M. Singh and L. Shrivastava, "Performance Analysis of IOT based Wireless Sensor Network and its Graphical User Interface for Air Quality Monitoring," in *2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN)*, Bangkok, Thailand: IEEE, Dec. 2023, pp. 35–39. DOI: 10.1109/CICN59264.2023.10402236.

[27] H. Erlenkötter, *XML: Extensible Markup Language von Anfang an* (rororo rororo Computer 61209), ger, Orig.-Ausg., 2. Aufl. Reinbek bei Hamburg: Rowohlt-Taschenbuch-Verl, 2008, ISBN: 9783499612091.