**Software Requirements Specification**

For

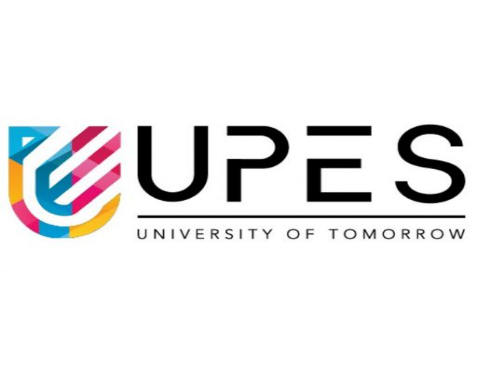
Air Touch

Prepared by

|  |  |  |
| --- | --- | --- |
| **Specialization** | **SAP ID** | **Name** |
| AIML | 500091953 | Nitish Gupta |
| AIML | 500091937 | Devansh Dutta |
| AIML | 500091012 | Ahhan Chhetri |

Under the guidance of

**Dr. Archana Kumari**



School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

Table of Contents

|  |  |  |
| --- | --- | --- |
| **Topic** | | **Page No** |
| Table of Content | | 2 |
| Revision History | | 3 |
| 1 | Introduction | 4 |
|  | 1.1 Purpose of the Project | 4 |
|  | 1.2 Target Beneficiary | 4 |
|  | 1.3 Project Scope | 5 |
|  | 1.4 References | 5 |
| 2 | Project Description | 6 |
|  | 2.1 Proposed Mathematical Model | 6 |
|  | 2.2 Data/ Data structure | 7 |
|  | 2.3 SWOT Analysis | 8 |
|  | 2.4 Project Features | 9 |
|  | 2.5 User Classes and Characteristics | 9 |
|  | 2.6 Design and Implementation Constraints | 10 |
|  | 2.7 Design diagrams | 11 |
|  | 2.8 Assumption and Dependencies | 13 |
| 3 | System Requirements | 14 |
|  | 3.1 User Interface | 14 |
|  | 3.2 Software Interface | 15 |
|  | 3.3 Database Interface | 15 |
|  | 3.4 Protocols | 16 |
| 4 | Non-functional Requirements | 17 |
|  | 4.1 Performance requirements | 17 |
|  | 4.2 Security requirements | 18 |
|  | 4.3 Software Quality Attributes | 18 |
| Appendix A: Glossary | | 19 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Change** | **Reason for Changes** | **Mentor Signature** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

INTRODUCTION

# PURPOSE OF THE PROJECT

* The Virtual Mouse project is all about creating a new way to control your computer without using a traditional mouse. Instead of clicking and moving a physical device, you’ll use simple hand movements to move the cursor and select things on your screen. This new system aims to make using a computer easier, especially for people who find it hard to use a regular mouse. It will work with your existing software and make interacting with technology more natural and comfortable.

TARGET BENEFICIARY

The key beneficiaries for a "Air Touch" project include:

1. People with disabilities (motor impairments, ALS, etc.) needing alternative input methods.

2. Elderly individuals with limited dexterity or mobility.

3. Students in special education requiring accessible technology.

4. Gamers with physical disabilities seeking adaptive solutions.

PROJECT SCOPE

The Air Touch project aims to develop an accessible system that allows users to control a computer cursor using alternative input methods such as gestures, and voice commands. It targets individuals with physical disabilities, the elderly, and gamers needing adaptive solutions. Key features include customizable settings, integration with various operating systems, and compatibility with sensors and voice recognition technology. The project will focus on ensuring privacy, security, and ease of use, with continuous testing, user feedback, and support for ongoing improvements.

**Benefits**

* **Accessibility:** Enables hands-free interaction, making computing more accessible for individuals with disabilities or mobility limitations.
* **Hygiene and Convenience:** Eliminates the need for physical contact, ideal for sterile environments and shared spaces.
* **Innovation and Portability:** Uses built-in cameras or sensors, reducing dependency on physical hardware and showcasing advanced technologies like gesture recognition and computer vision.

**Functional Requirements**

* Gesture Detection: Recognize hand gestures for mouse actions like click and scroll.
* Cursor Control: Move the on-screen cursor using hand or finger movements.
* Custom Settings: Allow users to adjust gestures and sensitivity.

**Project Deliverables**

1. Functional Virtual Mouse: A working system with gesture-based cursor control.

2. User Guide: Instructions for setup and use.

3. Source Code: Well-documented project code.

REFERENCES

1. Hinckley, K., Morris, M. R., Baudel, A. A., & Lindh, R. W. (2006). Sensing Techniques for Human-Computer Interaction. ACM Transactions on Computer-Human Interaction, 13(4), 340-372.

2. Liu, X., Hsieh, M. H., & Huang, T. J. (2013). Touchless Interaction Techniques for People with Disabilities. Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems, 2315-2324.

3. Zhai, S., Kong, S. W., & Li, M. P. K. (2004). The Role of Virtual Mouse in Ergonomics. International Journal of Human-Computer Studies, 61(5), 621-635.

4. Igarashi, T., Saito, A., & Kiyokawa, K. (2005). Adapting Virtual Mouse to Existing Software Environments. Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, 1-8.

5. Elmqvist, N., Schmidt, T. M. K., & Williams, J. M. (2011). Emerging Trends in Virtual Input Devices. Journal of Computer Graphics, Multimedia & Animation, 15(2), 45-59.

PROJECT DESCRIPTION

The Air Touch project aims to replace a traditional hardware mouse with a gesture-based system, enabling users to control a computer using hand movements captured by a camera. It uses technologies like computer vision and gesture recognition to interpret user inputs such as cursor movement, clicks, and scrolling. This project enhances accessibility, hygiene, and convenience by eliminating the need for physical contact, making it ideal for environments like healthcare, shared spaces, or personal use. The system is designed to be user-friendly, portable, and compatible with various platforms.

CHARACTERISTIC OF DATASET

1. Relevance

- The dataset should include images or videos that represent the gestures and actions required for the virtual mouse, such as finger movements, hand positions, and click gestures.

2. Diversity

- Capture data from different angles, lighting conditions, and backgrounds to improve model generalization.

- Include hands of different sizes, shapes, skin tones, and orientations.

3. Annotation

- Each image or frame must be labeled with corresponding actions (e.g., "click," "scroll up," "cursor move").

- Use consistent and clear annotation formats for easy training.

4. Volume

- Ensure the dataset has a sufficient number of samples for each gesture to avoid class imbalance during training.

5. Quality

- High-resolution images or videos for precise feature detection.

- Minimize noise and blurriness in data samples.

6. Scalability

- The dataset should allow for the addition of new gestures or features as the project evolves.

7. Privacy and Ethics

- Ensure that the data is collected ethically, with consent if using other participants, and avoid capturing unrelated personal information.

SWOT ANALYSIS

Strengths

1. Accessibility: Offers a contact-free solution, aiding individuals with physical disabilities.

2. Innovative Technology: Leverages modern tools like computer vision and AI.

3. Cost-Effective: Requires minimal hardware, using existing cameras and sensors.

Weaknesses

1. Environment Dependency: Performance may vary under poor lighting or complex backgrounds.

2. Learning Curve: Users may take time to adapt to gesture-based controls.

3. Accuracy Issues: Misinterpretation of gestures could lead to errors.

Opportunities

1. Healthcare and Hygiene: Increased demand for touchless systems in medical and shared spaces.

2. Expansion Potential: Integration with AR/VR or smart devices.

3. Customization: Opportunities for user-specific adaptations and commercial scalability.

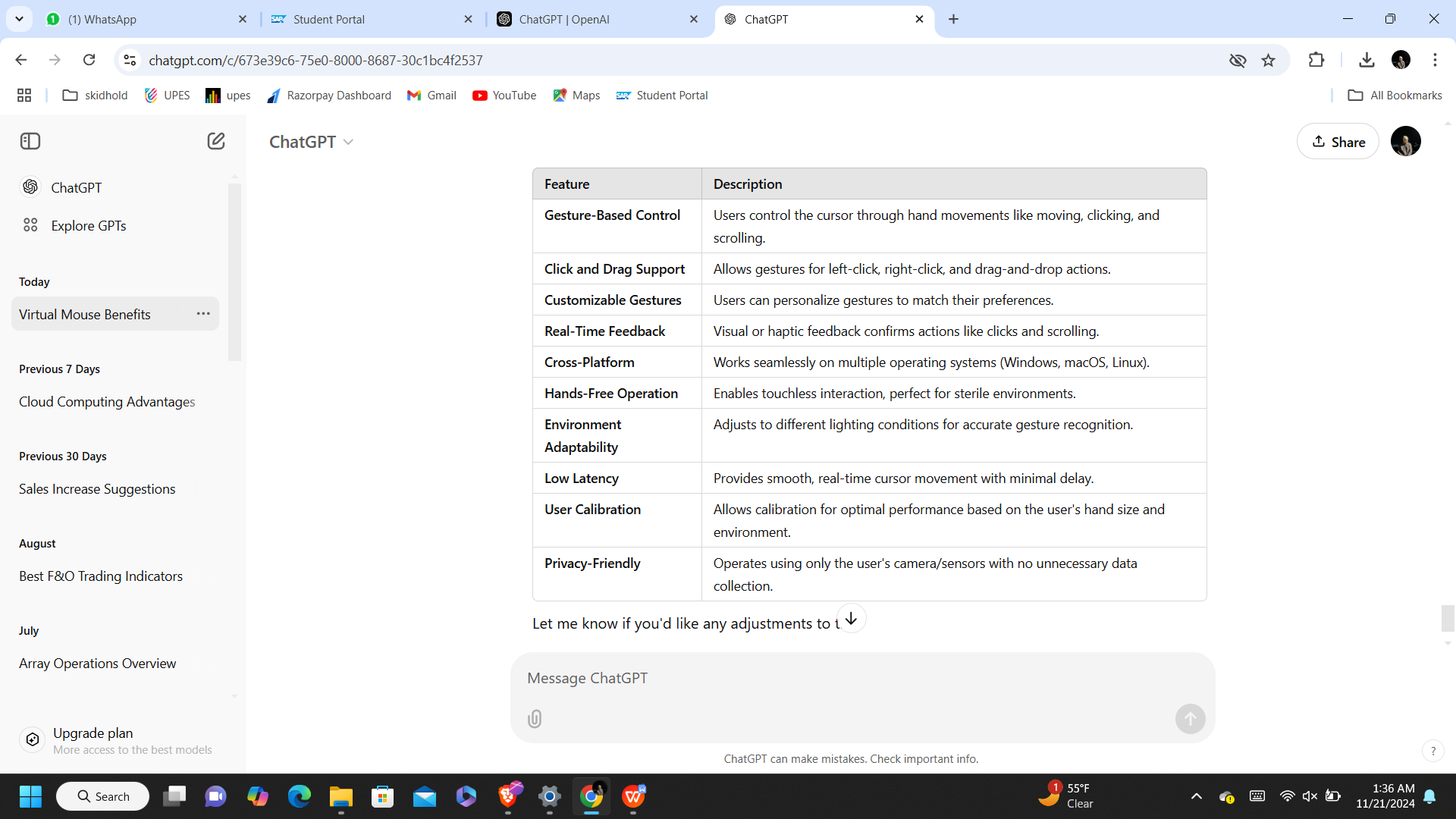
Threats

1. Competition: Similar technologies and devices in the market.

2. Technology Limitations: Dependence on camera quality and processing power.

3. User Resistance: Preference for traditional mice over adapting to new systems.

PROJECT FEATURES



USER CLASSES AND CHARACTERICSTICS

|  |  |  |
| --- | --- | --- |
| **User class** | **Characteristics** | **Needs** |
| **General Users** | Basic tech-savvy users, comfortable with gesture-based technology. | Simple, intuitive controls for everyday tasks; customizable settings. |
| **People with Disabilities** | Users with physical disabilities (e.g., limited hand mobility. | Alternative control methods (gestures/voice), high accessibility, error correction. |
| **Gamers and Creative Professionals** | Advanced users needing precision and responsiveness. | Precision control, fast response, advanced functionality for specialized tasks (e.g., gaming, 3D modeling). |
| **Business Users and Professionals** | Professionals who use computers in work environments (e.g., office workers, presenters). | Hands-free, ergonomic interaction, easy cursor navigation for presentations or meetings. |

DESIGN AND IMPLEMENTATION CONSTRAINTS

1. Hardware Requirements:

The system depends on the camera or sensor available on the user's device, which may affect performance due to variations in quality.

2. Environmental Factors:

Lighting and background conditions can impact gesture recognition accuracy, requiring controlled environments for optimal performance.

3. Real-Time Processing:

Gesture recognition and cursor control must be processed in real-time to avoid delays, which demands efficient algorithms and sufficient system resources.

4. User Calibration:

Users may need to calibrate the system for better accuracy, which could involve time-consuming setup steps.

5. Operating System Compatibility:

The system needs to work across multiple operating systems (Windows, macOS, Linux), which may require adapting to different system APIs and hardware management features.

6. Gesture Recognition Accuracy:

The system must be able to detect a wide range of hand sizes, gestures, and actions accurately, which can be challenging.

7. Privacy and Security:

Camera usage should be transparent, and personal data must not be collected, ensuring compliance with privacy regulations.

8. Hardware Limitations:

Lower-quality cameras or sensors on some devices may limit the system's effectiveness.

9. System Resource Usage:

The application must be optimized to use minimal CPU and memory resources, particularly for lower-end devices.

10. User Adaptability:

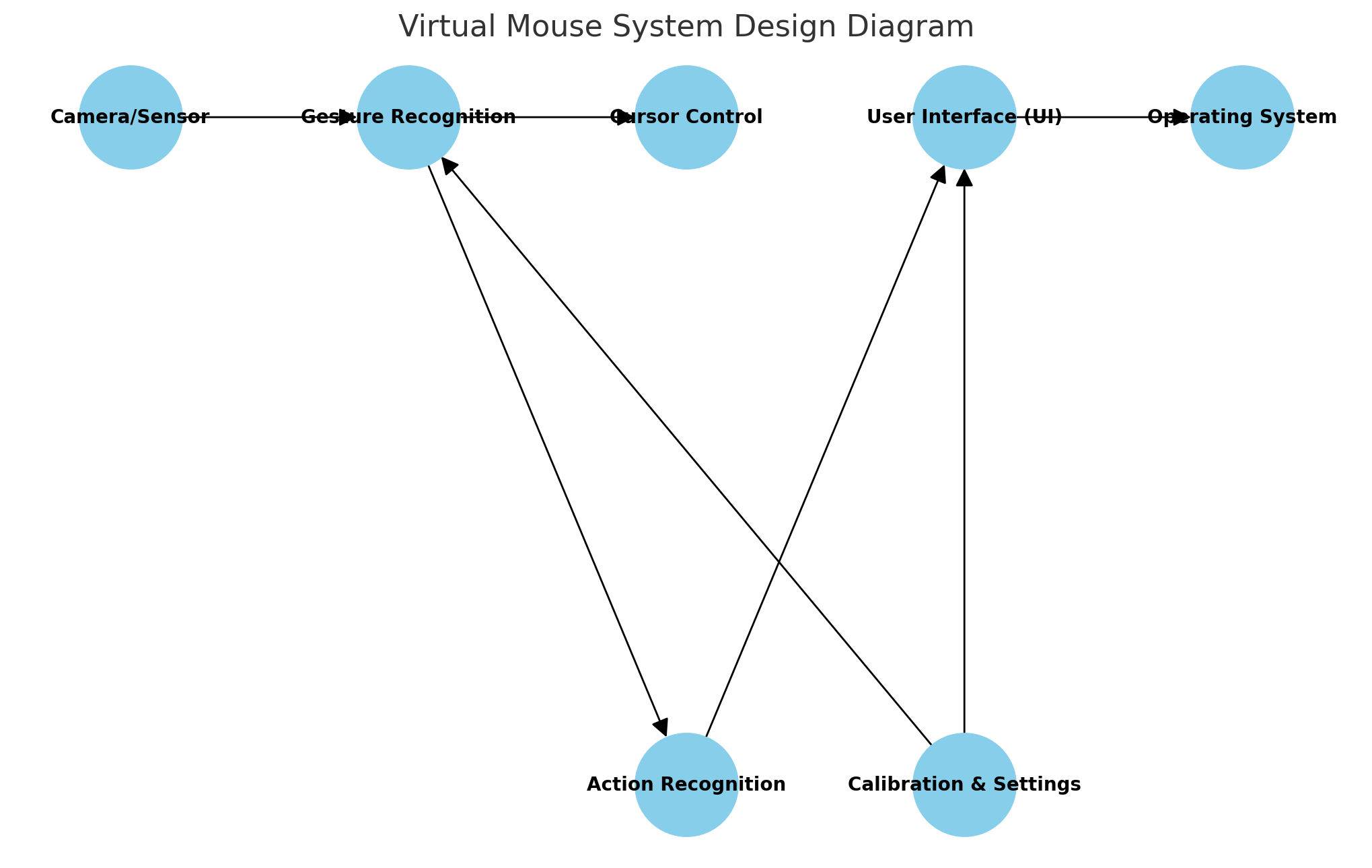
Users must find the system intuitive, and it should not require significant learning time to operate effectively.

DESIGN DIAGRAMS

A design diagram for the **Air Touch project** typically involves components like **gesture recognition**, **cursor movement**, and **user interaction**. Here’s a simplified conceptual diagram outlining the key modules:

1. **Camera/Sensor**: Captures the user’s hand gestures.
2. **Gesture Recognition Module**: Processes the input from the camera to detect gestures.
3. **Cursor Control Module**: Moves the cursor based on recognized gestures.
4. **Action Recognition Module**: Detects specific actions like clicks or scrolls.
5. **User Interface (UI)**: Displays feedback and interacts with the operating system.
6. **Calibration & Settings**: Allows the user to adjust gestures and system settings.
7. **Operating System**: The environment where the virtual mouse operates, interacting with applications.

I’ll now generate a diagram to visually represent this flow.



**Server Listener**

1. Client (User Input): The user provides hand gesture data.

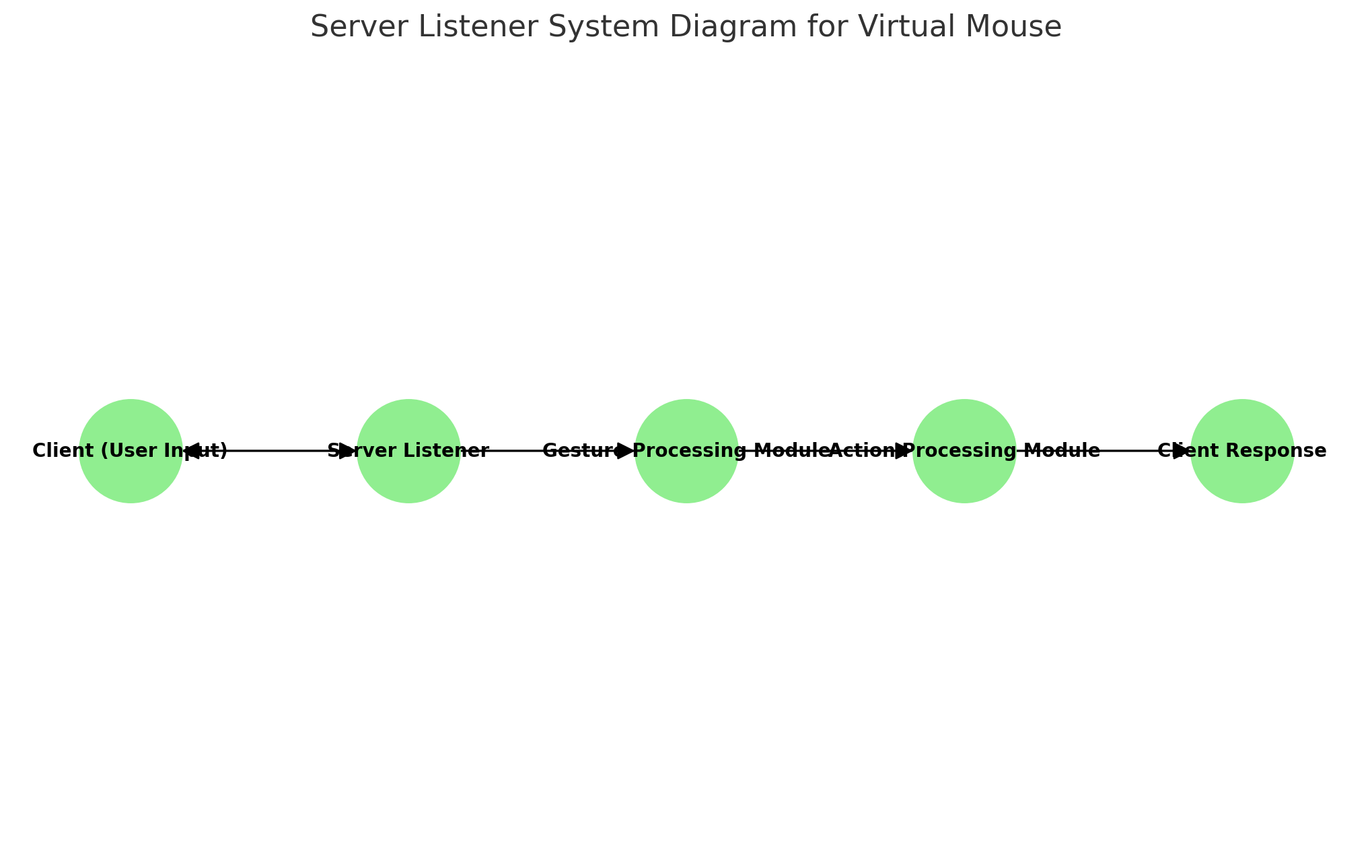
2. Server Listener: Receives and listens for gesture data from the client.

3. Gesture Processing Module: Analyzes the gestures and determines the corresponding action.

4. Action Processing Module: Executes the appropriate action (e.g., move cursor or click).

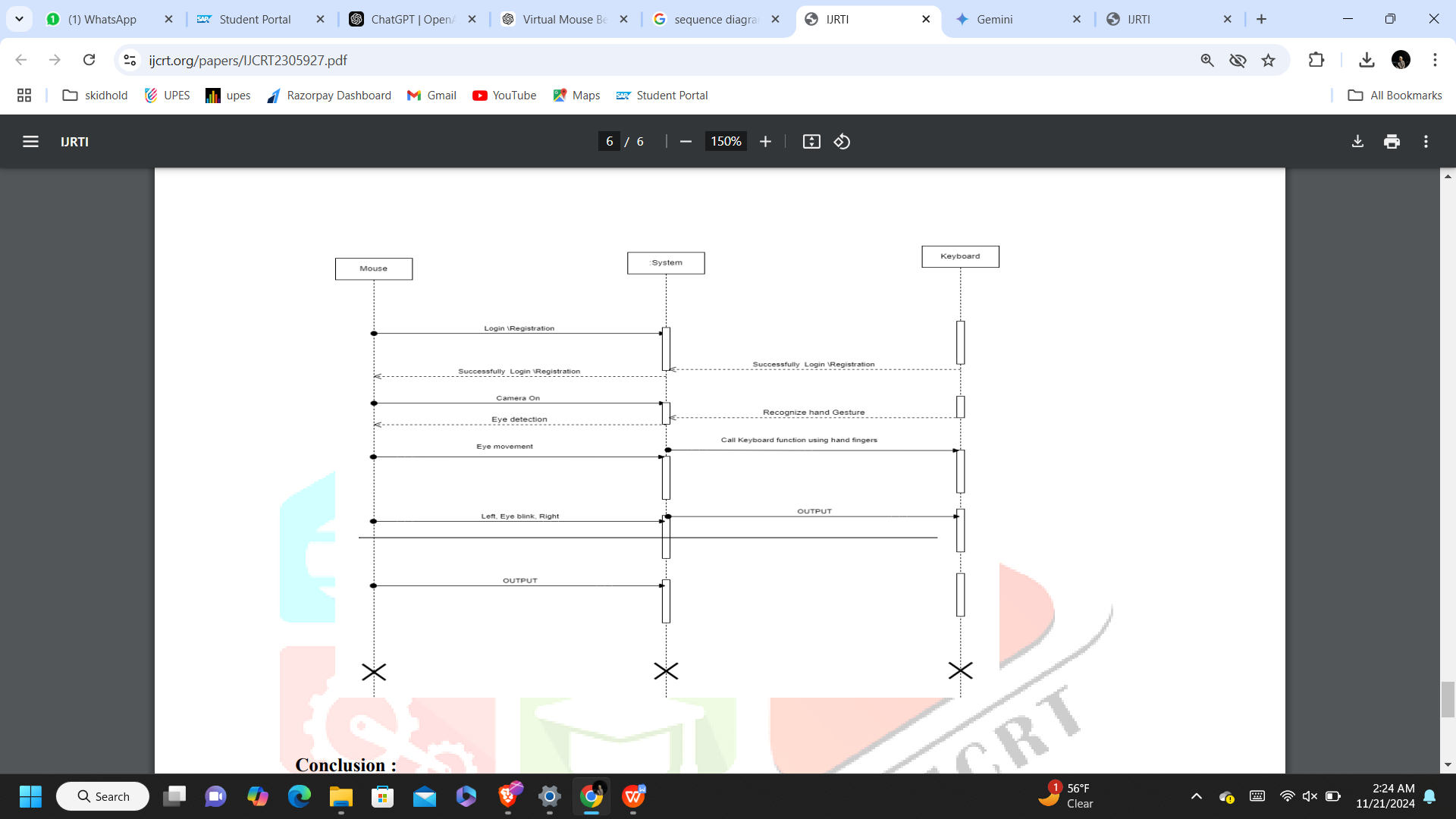
5. Client Response: Sends feedback (e.g., updated cursor position) back to the client.

This flow ensures real-time interaction between the user and the system.server side.



**Sequence Diagram**

A virtual mouse is a software application that allows users to control a computer cursor using gestures or movements captured by a camera. It eliminates the need for a physical mouse, providing a hands-free and potentially more intuitive interaction method.



**Data Flow Diagram**

The Level 2 Data Flow Diagram (DFD) is an extension of the Level 1 DFD and provides a more detailed view of the functional processes within the system. While the Level 1 DFD provided an overview of the entire system, the Level 2 DFD delves into specific modules and processes.

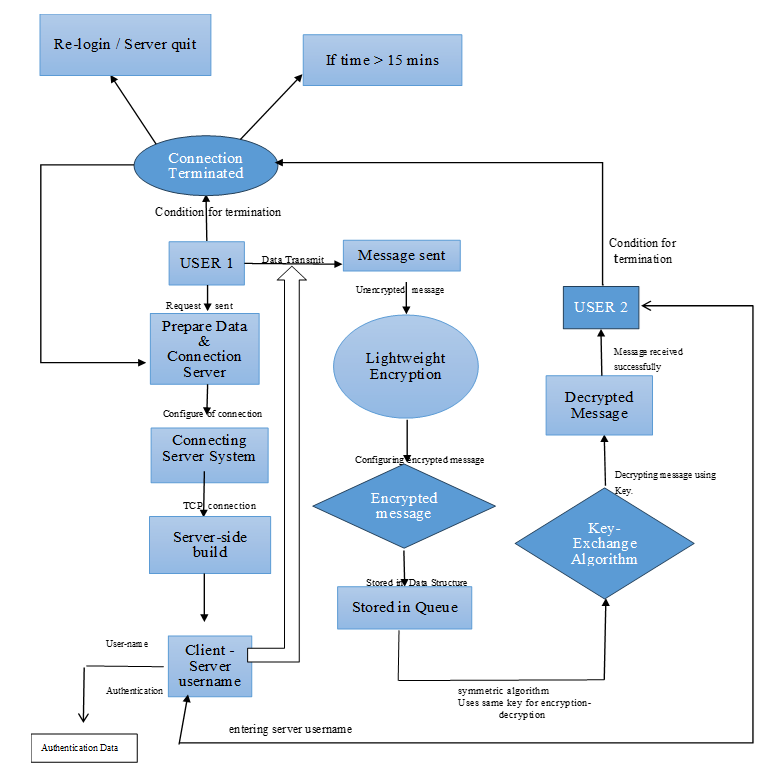
The Level 2 DFD encompasses finer details regarding the system's data flow, processes, data storage, and external entities. It provides a closer look at the internal workings of the system, breaking down complex processes into more manageable and understandable components.

The primary elements included in the Level 2 DFD are:

1. **Processes:** Detailed sub-functions representing specific activities and actions within the system.
2. **Data Flows:** Detailed connections showing the movement of data between processes and data stores.
3. **External Entities:** Various external sources or destinations interacting with the system.

The Level 2 DFD will act as a blueprint for the development and implementation of the software, serving as a crucial reference for software engineers, developers, and system architects, providing a clearer understanding of system processes and relationships.

ASSUMED FACTORS AND DEPENDENCIESAssumed factors and dependencies refer to conditions or resources essential for a system's functionality, performance, or success.**Dependencies on External Factors**

1. **Hardware Quality:** The performance of the system depends on camera resolution and processing power.
2. **Environmental Conditions:** Adequate lighting and minimal background noise are crucial for accurate gesture recognition.
3. **User Adaptation:** System success relies on users adapting to and adopting the gesture-based interface effectively.
4. **Technological Compatibility:** The system must be compatible with various operating systems and existing software applications.

**SYSTEM REQUIREMENTS**

USER INTERFACE

The following software components necessitate a user interface to ensure seamless interaction and utilization:

Each of these components requires a user interface to support user interactions, facilitate system navigation, and provide a user-friendly environment for the functionality offered by the application.

SOFTWARE INTERFACE

**1. Virtual Mouse and Authentication Module**

* Service Requirements:

The virtual mouse enables secure and gesture-based authentication to ensure that only authorized users can access the system. It supports integration with biometric recognition, like hand or face detection, to enhance security measures

* Communication Nature:
  + Data related to gestures or biometrics is securely transmitted between the virtual mouse interface and the authentication module using encrypted protocols to prevent data breaches or unauthorized access

**2. Virtual Mouse and Encryption Module:**

* Service Requirements:
  + The virtual mouse employs robust encryption mechanisms to protect user data and gestures during transmission and storage. Algorithms like AES ensure secure handling of sensitive information..
* Communication Nature:
  + Encrypted communication ensures real-time data security between the virtual mouse interface and connected modules, maintaining system integrity and preventing unauthorized interception..

DATABASE INTERFACE

The application doesn’t incorporate a traditional Database Management System (DBMS) for data storage and retrieval. Instead, it utilizes in-memory data structures to handle real-time data flow.

PROTOCOLS

**NON-FUNCTIONAL REQUIREMENTS**

PERFORMANCE REQUIREMENTS

**General Performance Considerations**

* Responsiveness: The virtual mouse should register and respond to user gestures with minimal delay, ensuring seamless interaction.
* Efficient Data Transfer: The system must efficiently transmit gesture data between the camera and the processing unit, ensuring minimal latency.

**Real-time System Considerations**

* Gesture recognition and cursor movement must occur within milliseconds to maintain a real-time experience.
* The system should utilize optimized algorithms to ensure low CPU and memory usage.

SECURITY REQUIREMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Protection** | **User Authentication** | **System Verification and Validation** | **Compliance Requirements** |
| All user gestures, biometric data, and interaction details must be encrypted to prevent unauthorized access. | The system must integrate secure authentication methods, such as biometric recognition (fingerprint, hand gesture recognition) and PIN-based systems. | The virtual mouse must undergo thorough testing to ensure that all gestures and hand movements are accurately recognized. | User acceptance testing (UAT) should be conducted to ensure that the virtual mouse meets the needs and expectations of the end-users. |
| The system should use secure protocols (e.g., HTTPS) for data transmission between the user interface and backend. | Multi-factor authentication (MFA) should be employed, adding an additional layer of security for user access. | Functional testing should verify that the system operates as expected across different environments, devices, and operating systems. | The system should be validated under real-world conditions, ensuring it performs well in various lighting conditions, user behaviors, and usage scenarios. |

SOFTWARE QUALITY ATTRIBUTES

1. Adaptability: The system should easily adapt to various environments, including different lighting conditions and screen resolutions

2. Availability: The virtual mouse should be highly available, with minimal downtime, ensuring that users can rely on it at all times.

3. Correctness: The system must accurately detect gestures and translate them into corresponding actions without errors.

4. Flexibility: The virtual mouse should allow customization of gestures, enabling users to personalize the system for their needs..

5. Interoperability: The system should function seamlessly with a wide range of devices, such as computers, tablets, and touch screens.

6. Maintainability: The codebase should be modular and well-documented, making it easier for developers to update and maintain.

7. PortabilityThe virtual mouse should be easily portable across different platforms, ensuring it works on various devices without requiring significant modifications.

1. Reliability: It must handle occasional system glitches or failures gracefully and recover without affecting user experience.

Each of these qualities is essential to ensure a well-rounded, high-quality software system. They contribute to the performance, user satisfaction, and success of the software in different ways.

**APPENDIX A: GLOSSARY**

1. SRS - Software Requirements Specification: A document outlining the functional and non-functional requirements of the software system to be developed.
2. Glossary of Terms: Key definitions such as Gesture Recognition, Server Listener, and UI (User Interface).
3. References: Notable books like "Computer Vision: Algorithms and Applications" and websites like OpenCV and GitHub for resources.
4. Acronyms: Common abbreviations, including API (Application Programming Interface), AI (Artificial Intelligence), and UI (User Interface).
5. Code Snippets: Example code for gesture detection using OpenCV in Python.
6. Troubleshooting: Solutions for common issues such as gesture recognition in low light or lagging cursor movements.
7. System Requirements: Hardware and software requirements, including compatible operating systems and minimum camera quality.
8. Testing Protocols: Procedures for testing system accuracy, responsiveness, and cross-platform compatibility.
9. User Manual: A simple guide on how to set up and use the virtual mouse system, including calibration steps.
10. Performance Optimizations: Techniques to improve system performance, reduce lag, and handle resource-intensive tasks.
11. Future Enhancements: Planned features like voice command integration, multi-user support, and haptic feedback.

This list covers some of the most common terms and acronyms used in software development and networking, which could be present in an SRS.