Voice Enabled Complaint Solution

##### **A MINI PROJECT REPORT**

###### **Submitted by**

##### **PRITHIKA SRINITHI.M [211422104355]**

##### **PRIYA.K[211422104357]**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

# COMPUTER SCIENCE AND ENGINEERING

****

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

##### OCTOBER 2024

**BONAFIDE CERTIFICATE**

Certified that this project report **“Voice Enabled Complaint Solution”**is the bonafide work of “**Prithika Srinithi.M (211422104355), Priya.K (211422104357)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.L.JABASHEELA,M.E.,Ph.D., MRS.S.SOPHANA JENNIFER,M.E**

**HEAD OF THE DEPARTMENT SUPERVISOR**

**PROFESSOR ASSISTANT PROFESSOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE,

NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on…………………

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. L.Jabasheela ,M.E., Ph.D.,HOD** for the support extended throughout the project.

We would like to thank my project guide **MRS.S.Sophana Jennifer, M.E,,Assistant Professor** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

**PRITHIKA SRINITHI.M**

**PRIYA.K**

**ABSTRACT**

Effective customer service is crucial for enhancing user satisfaction and loyalty. The Voice Enabled Complaint Solution streamlines the complaint process using natural language processing and voice recognition, allowing users to express grievances verbally and intuitively.

The Voice Enabled Complaint Solution employs advanced speech-to-text technology and machine learning algorithms to categorize and prioritize complaints in real time, ensuring quicker responses from the appropriate resolution teams. Additionally, it integrates sentiment analysis to assess the emotional tone of users’ voices, fostering empathetic interactions.

Preliminary tests indicate that the Voice Enabled Complaint Solution significantly increases user engagement and satisfaction compared to traditional text-based systems. Users report that verbal submissions are easier, leading to higher completion rates. The system's analytics dashboard provides organizations with insights into common complaint trends, enabling proactive measures.

Accessibility is a priority, accommodating users with varying technological skills and disabilities. The integration of artificial intelligence allows the system to learn from interactions, continuously improving its responses. This adaptability enhances user experience and equips organizations to meet changing customer expectations.

Beyond resolving complaints, the Voice Enabled Complaint Solution creates a feedback loop that informs product development and service enhancements, promoting a customer-centric approach. Future developments aim to expand capabilities, including multilingual support and integration with customer relationship management platforms.

By leveraging real-time data analytics, the Voice Enabled Complaint Solution empowers organizations to identify recurring issues and implement targeted improvements, transforming the complaint-handling process and reinforcing the value of effective complaint management for sustained success and customer loyalty.

Furthermore, the Voice Enabled Complaint Solution can serve as a valuable tool for data-driven decision-making within organizations. By capturing and analyzing user feedback in real time, businesses can identify patterns and trends that inform not just complaint resolution, but also broader strategic initiatives. This capability allows organizations to proactively address recurring issues, enhance product offerings, and refine service delivery methods. As the system evolves, it could also integrate with other customer engagement platforms, creating a holistic approach to customer relationship management. This interconnectedness will further empower organizations to anticipate customer needs and respond more effectively, ultimately driving continuous improvement across all facets of customer service.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO:** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | **ABSTRACT**  **LIST OF FIGURES** |  |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Overview |  |
|  | 1.2 Problem Definition |  |
| **2.** | **LITERATURE SURVEY** | 3 |
| **3.** | **SYSTEM ANALYSIS** | 5 |
|  | 3.1 Existing System |  |
|  | 3.2 Proposed system |  |
|  | 3.3 Technology Stack |  |
| **4.** | **SYSTEM DESIGN** | 10 |
|  | 4.1. Flow Chart |  |
|  | 4.2 UML diagrams |  |
| **5.** | **SYSTEM ARCHITECTURE** | 13 |
|  | 5.1 Architecture Overview |  |
|  | 5.1 Description of the Modules |  |
| **6.** | **SYSTEM IMPLEMENTATION** | 20 |
|  | 6.1 Source coding |  |
|  | 6.2 Server-side coding  6.3 Appendices |  |
| **7.** | **SYSTEM TESTING** | 36 |
|  | Test Cases & Reports / Performance Analysis |  |
| **8.** | **CONCLUSION** | 39 |
|  | Conclusion and Future Enhancements |  |
|  | **REFERENCES** | 43 |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

1.Flowchart 10

2.Usecase diagram 10

3.Class diagram 11

4.Sequence diagram 11

5.Activity diagram 12

6.Architecture diagram 13

**1.INTRODUCTION**

**1.1 Overview**

In a developing country like India, direct communication between citizens and the government is often lacking. This gap in communication can lead to increased bribery and unethical practices, making it uncertain whether issues will be resolved in a timely manner. Furthermore, the cost of sacrificing a day of work to lodge a complaint in person at municipal offices can be prohibitive for many.

The Voice Enabled Complaint Solution is an innovative solution aimed at transforming this landscape by streamlining the process of lodging complaints. Utilizing advanced natural language processing and voice recognition technologies, the system allows users to express their grievances verbally, creating a seamless and intuitive experience. By employing sophisticated speech-to-text capabilities, it captures user input and leverages machine learning algorithms to categorize and prioritize complaints in real time, ensuring that issues are swiftly directed to the appropriate resolution teams.

Additionally, the system integrates sentiment analysis to assess the emotional tone of users’ voices, fostering more empathetic interactions. Preliminary tests have demonstrated significant increases in user engagement and satisfaction compared to traditional text-based systems, with users appreciating the ease and immediacy of verbal submissions. The analytics dashboard provides organizations with valuable insights into common complaint trends, enabling proactive measures to address underlying issues.

Designed for accessibility, the Voice Enabled Complaint Solution accommodates users with varying technological proficiency, including those with disabilities, thus promoting inclusivity. As the system continues to evolve, it aims to expand its capabilities to include multilingual support and integration with existing customer relationship management platforms. Overall, the Voice Enabled Complaint Solution not only enhances operational efficiency but also promotes a customer-centric service model, redefining how citizens engage with their government.

1.2 **Problem Definition**

In developing countries like India, a significant gap exists between citizens and government agencies, leading to ineffective communication and unresolved grievances. This disconnect fosters an environment where bribery and unethical practices can thrive, as individuals often feel compelled to resort to these methods to expedite their concerns. Additionally, the process of lodging complaints in person is not only time-consuming but also burdensome, requiring citizens to sacrifice valuable work hours or personal time.

Current complaint submission methods, predominantly text-based, can be cumbersome and inaccessible, particularly for individuals with varying levels of technological proficiency or disabilities. As a result, many grievances remain unreported or unresolved, exacerbating public frustration and diminishing trust in governmental processes.

The lack of an efficient, user-friendly system for complaint management hinders the ability of government agencies to address issues promptly and effectively, ultimately impacting citizen satisfaction and engagement. There is a clear need for an innovative solution that simplifies the complaint submission process, enhances communication between citizens and government, and promotes transparency and accountability.

The existing complaint mechanisms often fail to accommodate the diverse needs of the population, resulting in a lack of inclusivity and accessibility. Many citizens, particularly those from marginalized communities, encounter barriers that prevent them from effectively voicing their concerns. This not only perpetuates feelings of disempowerment but also limits the government’s understanding of the issues facing its constituents.

Moreover, the absence of real-time feedback and follow-up creates frustration among citizens who seek timely resolutions to their grievances. Without a streamlined and empathetic approach to complaint handling, the relationship between the government and the public remains strained, further undermining trust and cooperation. Thus, there is an urgent demand for a robust system that addresses these challenges, enabling citizens to engage with their government in a more meaningful and effective manner.

**2. LITERATURE SURVEY**

The intersection of technology and public service has been a growing area of interest in recent years, particularly in enhancing citizen engagement and improving complaint management systems.Various studies highlight the significance of effective communication between government entities and citizens, with a focus on addressing the barriers that hinder this interaction.

**1. Citizen Engagement and Complaint Management:** Research indicates that traditional complaint systems often fail to meet the needs of citizens, resulting in underreporting of grievances. A study by D’Auria et al. (2020) found that users prefer streamlined processes that offer real-time feedback, emphasizing the importance of user experience in public service applications. This aligns with findings from the World Bank (2019), which suggest that effective complaint management can significantly improve citizen satisfaction and trust in government.

**2. Voice Technology in Public Services:** The use of voice recognition and natural language processing in public service applications has gained traction. According to Gupta and Sharma (2021), implementing voice-enabled systems can enhance accessibility and usability for diverse populations, particularly for those with disabilities or limited literacy. These technologies can facilitate a more inclusive environment by allowing users to express grievances in their natural language, thus reducing barriers to engagement.

**3. Sentiment Analysis for Improved Interactions:** Sentiment analysis has been increasingly applied in customer service settings to gauge user emotions and improve interactions. Research by Kumar et al. (2020) demonstrates that integrating sentiment analysis into complaint systems can foster empathetic responses from service providers, leading to higher user satisfaction. This approach is particularly relevant for government systems, where emotional tone can significantly impact the perception of responsiveness and care.

**4. Emotional Intelligence in Customer Service:** The integration of emotional intelligence into complaint resolution is gaining recognition as a key factor in customer satisfaction. Research by Pappas (2019) demonstrates that systems capable of detecting and responding to user emotions, through sentiment analysis, can significantly enhance the quality of interactions. This is particularly relevant in public service contexts, where citizens often express frustration and dissatisfaction with bureaucratic processes. A voice-enabled system that responds empathetically can transform the user experience and foster a sense of being heard.

**5. Case Studies of Successful Implementations:** Several case studies illustrate the successful implementation of voice-enabled complaint systems in various contexts. For instance, the “Smart City” initiatives in countries like Singapore and Estonia have showcased how technology can facilitate effective communication between citizens and government. These systems not only improve complaint resolution times but also gather valuable data for policy-making and resource allocation.

**6. Challenges and Future Directions:** Despite the promising potential of voice-enabled systems, challenges remain. Issues such as data privacy, security, and the need for robust infrastructure must be addressed to ensure successful implementation. Future research should focus on developing scalable models that can be adapted to different governmental contexts, as well as exploring the integration of multilingual support to cater to diverse populations.

**7. Future Research Directions:** Despite the promising outcomes associated with voice-enabled complaint systems, challenges such as data security, user privacy, and system integration remain pertinent. Future research should explore frameworks for ensuring data protection while leveraging user feedback effectively. Additionally, studies should focus on the scalability of these systems in different socio-economic contexts to understand how they can be tailored to meet the needs of diverse populations.

Overall, the literature underscores the necessity for innovative solutions like the Voice Enabled Complaint Solution to bridge the communication gap between citizens and government, enhance engagement, and foster a more transparent and responsive public service environment.

**3. SYSTEM ANALYSIS**

|  |
| --- |
| **3.1 Existing System**    **1.Voice Input Interface:** This component enables users to lodge complaints verbally, providing a seamless and intuitive experience. Utilizing advanced speech recognition technology, it captures spoken input and converts it into text, significantly reducing barriers associated with traditional text-based complaint methods. This interface is designed to accommodate users with varying levels of technological proficiency, promoting inclusivity.  **2.Complaint Processing Engine:** Once the complaint is transcribed, it is processed by a natural language processing (NLP) engine. This engine categorizes and prioritizes complaints in real-time, ensuring that issues are routed to the appropriate resolution teams swiftly. Additionally, the system integrates sentiment analysis to assess the emotional tone of the complaints, allowing for more empathetic and tailored responses.    **Limitations:**  The existing system has demonstrated notable improvements in user engagement and satisfaction compared to conventional complaint mechanisms. Initial user tests indicate that verbal submissions result in higher completion rates and a greater sense of immediacy in addressing grievances. Furthermore, the analytics component of the system provides organizations with valuable insights into common complaint trends, enabling proactive measures to improve services. By incorporating these technologies, the Voice Enabled Complaint Solution not only enhances operational efficiency but also fosters a more customer-centric approach, redefining the interaction between citizens and government entities. |
| **3.2 Proposed system**  The proposed Voice Enabled Complaint Solution aims to enhance the complaint submission process by integrating advanced technologies to provide a more user-friendly and efficient experience. It consists of the following components:   1. **Voice Input Interface:**    * **Functionality:** Users can submit complaints verbally using natural language, making the process accessible to individuals with varying levels of technological proficiency.    * **Technology:** Utilizes state-of-the-art speech recognition algorithms to accurately capture and transcribe spoken input. 2. **Natural Language Processing (NLP) Engine:**    * **Functionality:** Analyzes the transcribed complaints to categorize and prioritize them based on urgency and type.    * **Technology:** Employs machine learning models trained on extensive datasets to improve accuracy in understanding user intent. 3. **Sentiment Analysis:**    * **Functionality:** Assesses the emotional tone of the user’s voice to gauge frustration or urgency, enabling empathetic responses from support teams.    * **Technology:** Integrates advanced sentiment analysis algorithms that provide insights into user emotions. 4. **Real-Time Feedback Mechanism:**    * **Functionality:** Keeps users informed about the status of their complaints through notifications and updates.    * **Technology:** Uses push notifications or SMS alerts to enhance user engagement and satisfaction. 5. **Analytics Dashboard:**    * **Functionality:** Provides organizations with insights into complaint trends, common issues, and response effectiveness.    * **Technology:** Utilizes data visualization tools to help stakeholders make informed decisions based on user feedback. 6. **Multilingual Support:**    * **Functionality:** Accommodates users who speak different languages, broadening accessibility and inclusivity.    * **Technology:** Implements language translation capabilities to ensure accurate communication regardless of the user’s language. 7. **Integration with Existing Systems:**    * **Functionality:** Connects with current customer relationship management (CRM) systems to streamline complaint resolution processes.    * **Technology:** Employs APIs to facilitate smooth data exchange and workflow management.   **Advantages:**   * **Enhanced Accessibility: Provides a user-friendly platform that encourages broader participation in the complaint process.** * **Improved User Experience: Streamlined verbal submissions increase ease of use, resulting in higher satisfaction rates.** * **Faster Resolution Times: Real-time categorization and prioritization ensure that complaints are directed to the right teams promptly.** * **Data-Driven Insights: Analytics enable organizations to proactively address recurring issues and improve service quality.** |
| **3.3 Technology Stack** |

This stack leverages modern web technologies and cloud services to create an efficient, user-friendly, and scalable voice-enabled complaint system. Adjust components based on your specific requirements and team expertise.

### Front-End Technologies

1. **HTML**
   * **Structure:** Use HTML to create the basic structure of your web application, including forms for submitting complaints and sections for displaying responses.
2. **CSS**
   * **Styling:** Use CSS to style your application, ensuring it's visually appealing and user-friendly. Consider using frameworks like Bootstrap or Tailwind CSS for responsive design and faster development.
3. **JavaScript**
   * **Interactivity:** JavaScript can be used for client-side scripting, enabling dynamic behavior like form validation, handling voice input, and making API calls to the backend.
   * **Voice Recognition:** You can implement the Web Speech API to capture voice input directly in the browser. This allows users to speak their complaints instead of typing them.

**Back-End Technologies**

1. **Java Servlets**

* **Description:** Servlets are Java classes that handle HTTP requests and responses. They can be used to process incoming data (like complaints) and send responses back to the frontend.
* **Functionality:** You can create servlets to handle different endpoints, such as submitting a complaint, retrieving past complaints, and processing voice input from the frontend.

**2.MySQL**

* **Description:** MySQL is a popular relational database management system. It’s great for storing structured data like user complaints, user information, and any other related data.
* **Data Schema:** Design a database schema that includes tables for users, complaints, and possibly categories or statuses for complaints.

**Example Flow**

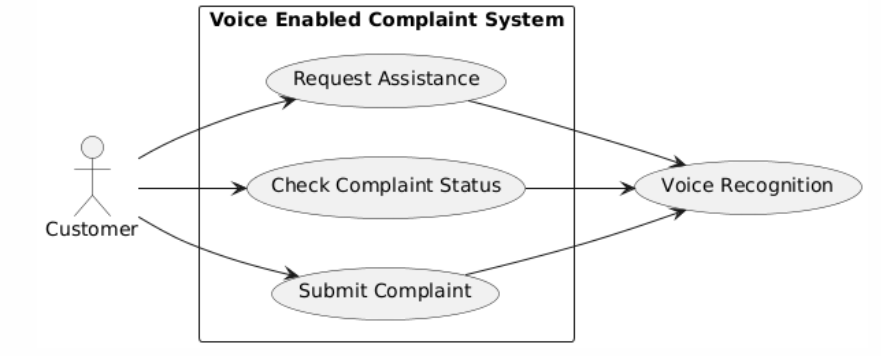
1. **User Submits Complaint:**The frontend captures voice input (converted to text) and sends it to a servlet via an AJAX call.
2. **Servlet Processing:**The servlet processes the request, validates the input, and interacts with the MySQL database to insert the complaint. You can use JDBC (Java Database Connectivity) to connect and execute SQL queries against your MySQL database.

3. **Database Interaction:**Use prepared statements to prevent SQL injection and ensure secure database operations.Store the complaint details, including the user ID, complaint text, timestamp, and status.

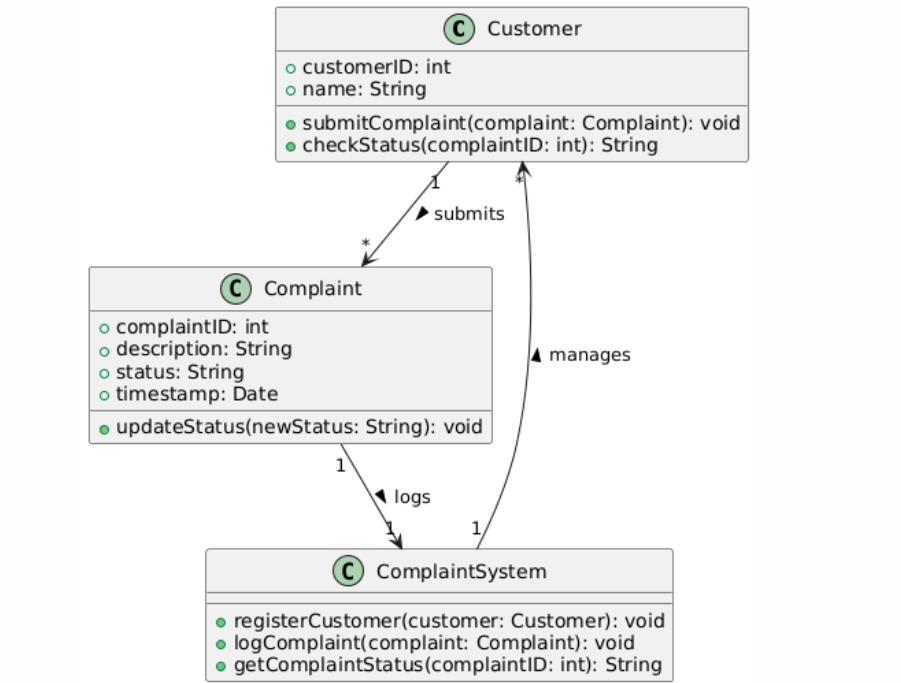
1. **Response Handling:**After processing, the servlet can send a response back to the frontend indicating whether the submission was successful or if there were any errors.

**4. SYSTEM DESIGN**

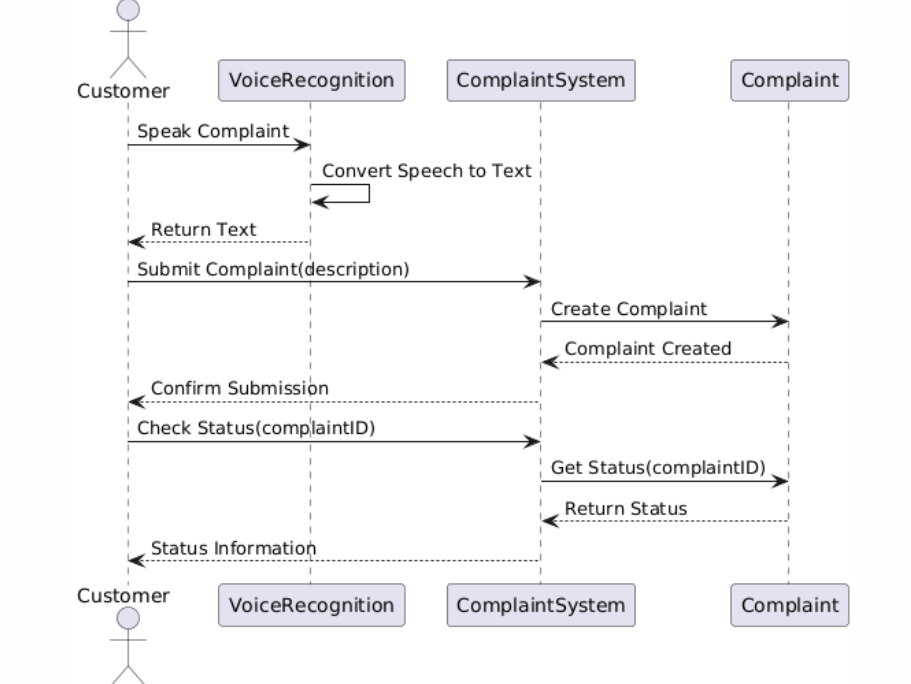
|  |
| --- |
| 4.1. Flow Chart |
| (fig 4.1)System Flowchart |



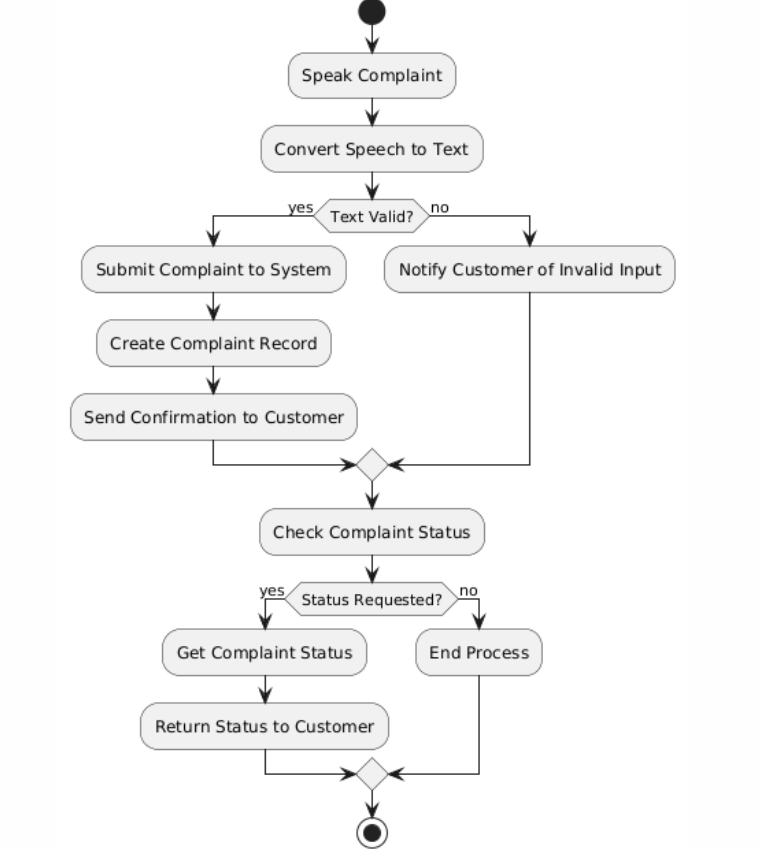
(fig 4.2)Usecase diagram



(fig 4.3)Class diagram



(fig 4.4)Sequence diagram



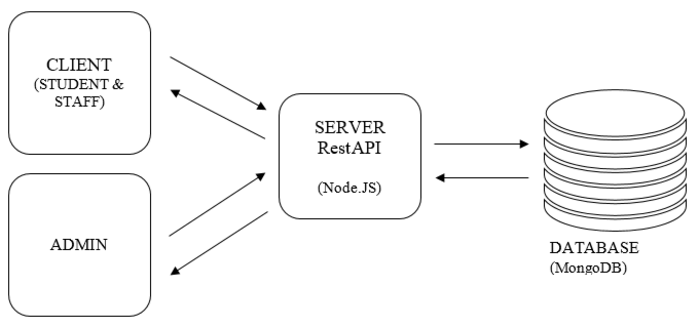
(fig 4.5)Activity diagram

**5.SYSTEM ARCHITECTURE**

**5.1 ARCHITECTURE OVERVIEW**

The architecture of the voice-enabled complaint system features a client-side frontend and a server-side backend. The frontend, built with HTML, CSS, and JavaScript, allows users to record complaints via voice input, which is converted to text using the Web Speech API. Upon submission, JavaScript sends an request to a Java Servlet, which processes the complaint and interacts with a MySQL database using JDBC to store the data. The database maintains tables for user complaints and related information. The servlet then sends a response back to the frontend, confirming the submission's success or any errors. This architecture ensures a seamless data flow, while considerations for scalability, security, and error handling enhance the overall user experience.

#### 1.Architecture Diagram



**(fig:5.1)System architecture**

|  |  |
| --- | --- |
| **5.2 Description of the Modules** |  |

The architecture consists of several key modules, each contributing to the functionality and user experience of the application. This section provides a detailed description of these modules.

#### 1.login page

* **Purpose**:The purpose of this login page is to provide users with a straightforward and secure interface for accessing an application or service. It allows users to input their credentials—username and password—to authenticate themselves and gain access to their accounts. By requiring users to fill out both fields before submission, the page ensures that all necessary information is provided, which enhances security. Additionally, the inclusion of links for password recovery and account registration improves user experience by offering support and pathways for new users. Overall, this login page is designed to facilitate secure user authentication while being accessible and easy to navigate.
* **Functionality**

**User Input:** Allows users to enter their username and password in designated fields, with placeholders providing guidance on the required format.

**Form Submission:** When the user clicks the "LOG IN" button, the form data is sent to a specified backend endpoint (e.g., a servlet) using the POST method for processing.

**Required Fields:** Both input fields are marked as required, preventing the form from being submitted unless they are filled out, thus ensuring that necessary information is provided.

**Forgot Password Link:** Provides a link that users can click if they need to recover or reset their password, facilitating user support.

**Registration Prompt:** Encourages users without accounts to register, directing them to a sign-up page, which enhances user engagement.

**Responsive Design:** The layout is designed to be user-friendly and accessible on various devices, ensuring a positive user experience across different screen sizes.

**Accessibility:** Labels associated with input fields improve accessibility for users utilizing assistive technologies, making it easier for all users to interact with the form.

#### 2. Registration page

* **Purpose**: The purpose of this registration page is to provide a user-friendly interface for individuals to create an account for a service or application. By filling out the form with their name, contact number, email, and password, users can securely register and gain access to personalized features. The page ensures that all necessary information is collected by marking fields as required, enhancing data integrity. Additionally, by including a note about agreeing to terms and privacy policies, it informs users of their rights and responsibilities when signing up. The design is intended to be clear and accessible, making it easy for users to navigate the registration process and encouraging them to engage with the platform. Overall, this page serves as a crucial entry point for new users, facilitating account creation and onboarding.

**Functionality**:

* **User Input Collection:** The registration form collects essential information from users, including their name, contact number, email, and password, enabling account creation.
* **Validation Requirements:** Each field is marked as required, ensuring users must fill out all necessary information before submission, which helps maintain data integrity.
* **Password Confirmation:** Users are prompted to enter their password twice to confirm accuracy, minimizing the chance of errors during registration.
* **Form Submission:** When the user clicks the "Register" button, the form data is submitted to a specified backend endpoint (to be defined in the action attribute). This is where the server-side logic would handle the registration process.
* **Terms and Privacy Notice:** The page informs users that by creating an account, they agree to the site's terms and privacy policies, promoting transparency.
* **Navigation Links:** A link is provided for users who already have an account, directing them to the login page, encouraging engagement with the platform.
* **Responsive Design:** The layout is designed to be accessible and user-friendly across various devices, ensuring a positive user experience.

#### 3.Complaint page

* **Purpose**: The purpose of this complaint page is to provide a structured and user-friendly platform for individuals to report incidents or issues they have encountered. By allowing users to submit detailed information—including personal details, incident descriptions, severity assessments, and supporting files—the form aims to gather comprehensive data for analysis by the Complaint Center. This helps ensure that reported issues are addressed effectively and that appropriate measures are taken to prevent future occurrences. The page also emphasizes the importance of truthful reporting, fostering transparency and accountability. Overall, the complaint page serves as a vital tool for facilitating communication between users and the organization, enabling timely responses to concerns and improving overall service quality
* **Functionality:**

U**ser Input Collection:** The form collects essential details from users, including their name, email, district, pin code, incident date, and incident location, enabling thorough reporting.

**Dropdown for District Selection:** Users can select their district from a predefined list, ensuring accurate data entry and making it easier for users to report their location.

**Required Fields:** Key input fields are marked as required, ensuring that users provide all necessary information before submitting the form.

**File Uploads:** Users can upload relevant images and audio files to support their complaints, enhancing the clarity and context of their submissions.

**Text Areas for Detailed Descriptions:** Multiple text areas allow users to provide detailed information about the complaint, including the title, description, severity, and desired outcomes.

**Digital Signature Upload:** Users can upload a file for their digital signature, adding authenticity to their complaint.

**Error Display Area:** Each input group includes an error div that can be used to display validation messages if the user does not meet the required criteria.

**Submission Button:** The "SEND COMPLAINT" button triggers the submission of the form, which sends the collected data to the specified backend for processing.

**Responsive Layout:** The form is designed to be user-friendly and accessible across various devices, ensuring a smooth experience for all users.

#### 4.loginServlet

* **Purpose**: The purpose of the LoginServlet is to authenticate users attempting to access the application by verifying their login credentials. Specifically, it:

1. User Authentication:Ensures that only registered users can log in by checking the provided username and password against stored records in the MySQL database.

2.Secure Access: Protects sensitive areas of the application, such as the complaint submission page, by allowing access only to authenticated users.

3.Error Handling: Provides feedback to users in case of invalid login attempts, enhancing the user experience by clearly indicating issues with their credentials.

4.Database Interaction:Demonstrates safe database practices by using prepared statements to prevent SQL injection attacks.

Overall, the LoginServlet serves as a critical component of the application's security framework, facilitating secure user authentication and access control..

* **Functionality**:

**Set Response Content Type:** It sets the response type to HTML, preparing the server to send back an HTML response to the client.

**Retrieve User Credentials:** The servlet retrieves the user's name and password from the incoming HTTP POST request.

**Establish Database Connection:**Loads the MySQL JDBC driver to enable communication with the MySQL database.Connects to the database using the specified URL, username, and password.

**Prepare SQL Query:**Constructs a SQL query to check for a user record in the register table that matches the provided username and password.Uses a PreparedStatement to securely include user input in the query, protecting against SQL injection.

**Execute Query:**Executes the prepared SQL query and retrieves the results using a ResultSet.

**Check User Existence:**Determines if a matching user record exists in the database.If a match is found, it redirects the user to the complaint page (complaint.html).If no match is found, it sends an error message indicating invalid username or password.

**Error Handling:**Catches any SQL or class loading exceptions that may occur during database operations and prints error messages for debugging.

**Resource Cleanup:** Ensures that database resources (ResultSet, PreparedStatement, and Connection) are properly closed to prevent memory leaks and maintain application performance.

#### 5.registerServlet

* **Purpose**: The purpose of the register servlet is to facilitate user registration by securely storing new user information in a MySQL database. Specifically, it aims to:

**Enable Account Creation:** Allow users to create accounts by submitting their name and password, which are essential for accessing the application.

**Data Integrity:** Ensure that the user data entered is safely handled and stored, maintaining the integrity of the database.

**Redirect Users After Registration:** Provide a seamless user experience by redirecting users to the login page after successful registration, encouraging them to log in and use the application.

**Error Feedback:** Offer clear feedback in case of registration failures, helping users understand issues that may have occurred during the process.

Overall, the register servlet is a key component in establishing user accounts within the application, supporting the onboarding process while emphasizing security and user experience.

* **Functionality**:

**Set Response Content Type:** The servlet sets the response content type to HTML, preparing to send a response back to the client.

**Retrieve User Input:** It collects the user's name and password from the request parameters submitted through the registration form.

**Establish Database Connection:**Loads the MySQL JDBC driver to enable database communication.Connects to the MySQL database using the specified URL, username, and password.

**Prepare SQL Insert Statement:**Constructs an SQL INSERT query to add the new user's details into the register table.Uses a PreparedStatement to safely include user input, which helps prevent SQL injection attacks.

**Execute the Insert Operation:**Executes the prepared statement to insert the new user record into the database.Checks the number of affected rows to determine if the insertion was successful.

**Redirect or Display Message:**If the insertion is successful (one or more rows affected), the user is redirected to the login page (login.html).If the insertion fails, an error message is displayed, informing the user of the failure.

**Error Handling:**Catches and logs any SQL or class loading exceptions that may occur during database interactions, displaying an error message if necessary.

**Resource Cleanup:** Ensures that the PreparedStatement and database connection are closed properly in a finally block to prevent memory leaks and maintain application performance.

Overall, the register servlet provides a secure and efficient mechanism for user registration, handling data collection, database interaction, and user feedback seamlessly.

**6. SYSTEM IMPLEMENTATION**

**6.1 Source Coding**

# Login.html:

<!DOCTYPE html>

<html>

<head>

<link href="style.css" rel="stylesheet">

<title>Log In</title>

</head>

<body>

<div class="login">

<form class="form">

<h1>Log In</h1>

<label>Username</label>

<input type="text" placeholder="Enter Your Username"><br>

<label>Password</label>

<input type="password" placeholder="Enter Your Password"><br>

<p><a>Forgot Password</a></p>

<button>LOG IN</button>

<p>Don't have an account?<a>Register</a></p>

</form>

</div>

</body>

</html>

## Registration.html:

<!DOCTYPE html>

<html>

<head>

<style>

    body

        {

        font-family: Arial, Helvetica, sans-serif;

        margin-left:25%;

        margin-right:25%;

        border: 1px solid #000000;

        margin-bottom: 5px;

        padding: 0px 15px 0 15px;

        }

    input[type=text], input[type=password]

        {

        width: 97%;

        padding: 10px;

        margin: 5px 0 22px 0;

        display: inline-block;

        border: none;

        background: #F5F5F5;

        }

    hr

        {

        border: 1px solid #e6e6e6;

        margin-bottom: 5px;

        }

    .registerbutton

        {

background-color: #29a329;

        color: white;

        padding: 15px 20px;

        margin: 10px 0px;

        border: none;

        cursor: pointer;

        width: 100%;

        text:bold;

        }

</style>

</head>

<body>

    <form action="action.php">

        <h1>Register</h1>

        <p>Please fill in this form to create an account.</p>

        <hr>

        <label for="name"><b>Name</b></label>

        <input type="text" placeholder="Enter Your Name" name="name" required><br>

        <label for="number"><b>Contact No.</b></label>

        <input type="text" placeholder="Enter Your Contact No." name="number" required><br>

        <label for="email"><b>Email</b></label>

        <input type="text" placeholder="Enter Your E-mail" name="email" required><br>

        <label for="pwd"><b>Password</b></label>

        <input type="password" placeholder="Enter Your Password" name="psw" required><br>

        <label for="psw-repeat"><b>Repeat Password</b></label>

        <input type="password" placeholder="Repeat Your Password" name="psw-repeat" required><br>

        <hr>

        <p>By creating an account you agree to our <a href="#">Terms & Privacy</a>.</p>

        <button type="submit" class="registerbutton">Register</button>

        <p>Already have an account? <a href="#">Sign in</a>.</p>

    </form>

</body>

</html>

## Complaint.html:

<!DOCTYPE html>

<html>

<head>

 <title>complaint page</title>

 <link rel="stylesheet" href="complaint.css">

 <script src="complaint.js" defer></script>

<head>

<body>

 <centre><h2>Complaint Form</h2></centre>

 <p>Please send us details about the incident you would like to report. Our Complaint Center will analyze your

  complaint and take the appropriate measures in order that the reported situation will not occur at any other

  time in the future.</p>

 <hr />

 <centre>

 <form id="form">

  <div class="input-group">

   <label>Name:</label><br>

   <input type="text" placeholder="Enter first name" required id="fir"><br/><br/>

   <input type="text" placeholder="Enter last name" id="las"><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Email:</label><br>

   <input type="email" required id="email" placeholder="Enter email address"><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label> District:</label><br>

   <select id="dis">

    <option value="" disabled selected>District</option>

    <option>Ariyalur</option>

    <option>Chengalpattu</option>

    <option>Chennai</option>

    <option>Coimabatore</option>

    <option>Cuddalore</option>

    <option>Dharmapuri</option>

    <option>Dindugul</option>

    <option>Erode</option>

    <option>kallakurichi</option>

    <option>Kancheepuram</option>

    <option>Karur</option>

    <option>Krishnagiri</option>

    <option>Madurai</option>

    <option>Mayiladuthurai</option>

    <option>Nagapattinam</option>

    <option>Kanniyakumari</option>

    <option>Namakkal</option>

    <option>Perambalur</option>

    <option>Pudukottai</option>

    <option>Ramanadhapuram</option>

    <option>Ranipet</option>

    <option>Salem</option>

    <option>Sivagangai</option>

    <option>Tenkasi</option>

    <option>Thanjavur</option>

    <option>Theni</option>

    <option>Thiruvallur</option>

    <option>Thiruvarur</option>

    <option>Thoothukudi</option>

    <option>Trichirapalli</option>

    <option>Thirunelveli</option>

    <option>Tirupathur</option>

    <option>Tiruppur</option>

    <option>Tiruvannamalai</option>

    <option>Nilgiris</option>

    <option>Vellore</option>

    <option>Villupuram</option>

    <option>Virudhunagar</option>

   </select><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Pin code:</label><br>

   <input type="text" placeholder="Pin code" required id="zip"><br><br>

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Date of the reported incident:</label><br>

   <input type="date" required id="dor"><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Incident location:</label><br>

   <textarea id="incloc" name="incloc" rows="4" cols="10"></textarea><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Complaint title:</label><br>

   <input type="text" required id="comtit" placeholder="complaint title"><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Complaint description:</label><br>

   <textarea id="comdet" name="Comdet" rows="4" cols="10"></textarea><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Severity:</label><br>

   <textarea id="sev" name="sev" rows="4" cols="10"></textarea><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Upload image:</label><br>

   <input type="file" accept=".jpg,.png" required id="img"><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Upload audio:</label><br>

   <input type="file" accept=".mp3" class="file"><br /><br />

   <div class="error"></div>

  </div>

  <div class="input-group">

   <label>Desired outcome:</label><br>

   <textarea id="desout" name="desout" rows="4" cols="10"></textarea><br /><br />

   <div class="error"></div>

  </div>

  <p class="sug">Suggest the steps that should be taken to avoid a repeat of the problem.</p><br />

  <div class="input-group">

   <label>Signature:</label><br>

   <input type="file" accept=".jpg,.png" class="fil">

   <div class="error"></div>

  </div>

  <p>Before signing declare that all information you entered is truthful and accurate.</p>

  <button type="submit">SEND COMPLAINT</button>

 </form>

 </centre>

</body>

</html>

**6.2 Server-side Scripting**

|  |  |
| --- | --- |
|  |  |

## LoginServlet.java:

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class LoginServlet extends HttpServlet {

    private static final String url = "jdbc:mysql://localhost:3306/users";

    private static final String user = "root";

    private static final String pass = "klnp@2005";

    protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

        // Set response content type

        res.setContentType("text/html;charset=UTF-8");

        // Retrieve the form data

        String name = req.getParameter("name");

        String password = req.getParameter("psw");

        Connection conn = null;

        PreparedStatement pstmt = null;

        ResultSet rs = null;

        try {

            // Load MySQL JDBC driver

            Class.forName("com.mysql.cj.jdbc.Driver");

            // Connect to the database

            conn = DriverManager.getConnection(url, user, pass);

            // Prepare SQL query to check if user exists with matching name and password

            String sql = "SELECT \* FROM register WHERE name = ? AND password = ?";

            pstmt = conn.prepareStatement(sql);

            pstmt.setString(1, name);

            pstmt.setString(2, password)

            // Execute the query

            rs = pstmt.executeQuery();

            // Check if user exists

            if (rs.next()) {

                res.sendRedirect("complaint.html");

            } else {

                res.getWriter().println("Invalid username or password.");

            }

        } catch (SQLException | ClassNotFoundException e) {

            e.printStackTrace();

            res.getWriter().println("Database error: " + e.getMessage());

        } finally {

            // Close the database resources

            try {

                if (rs != null) rs.close();

                if (pstmt != null) pstmt.close();

                if (conn != null) conn.close();

            } catch (SQLException e) {

                e.printStackTrace();

            }

        }

    }

}

**RegisterServlet.java:**

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class register extends HttpServlet{

    private static final String url = "jdbc:mysql://localhost:3306/users";

    private static final String user = "root";

    private static final String pass ="klnp@2005";

    protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

        res.setContentType("text/html;charset=UTF-8");

        String name = req.getParameter("name");

        String password = req.getParameter("psw");

        Connection conn = null;

        PreparedStatement pstmt = null;

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            conn = DriverManager.getConnection(url,user,pass);

            String sql = "INSERT INTO register(name,password) values(?,?)";

            System.out.println(sql);

            pstmt = conn.prepareStatement(sql);

            pstmt.setString(1, name);

            pstmt.setString(2, password);

            int rows = pstmt.executeUpdate();

            if(rows>0) {

                res.sendRedirect("login.html");

            }

            else {

                res.getWriter().println("Failed to add user");

            }

        }catch(SQLException | ClassNotFoundException e) {

            e.printStackTrace();

            res.getWriter().println("Database error:" +e.getMessage());

        }finally {

            try {

                if(pstmt != null)pstmt.close();

                if(conn != null)conn.close();

            }catch(SQLException e) {

                e.printStackTrace();

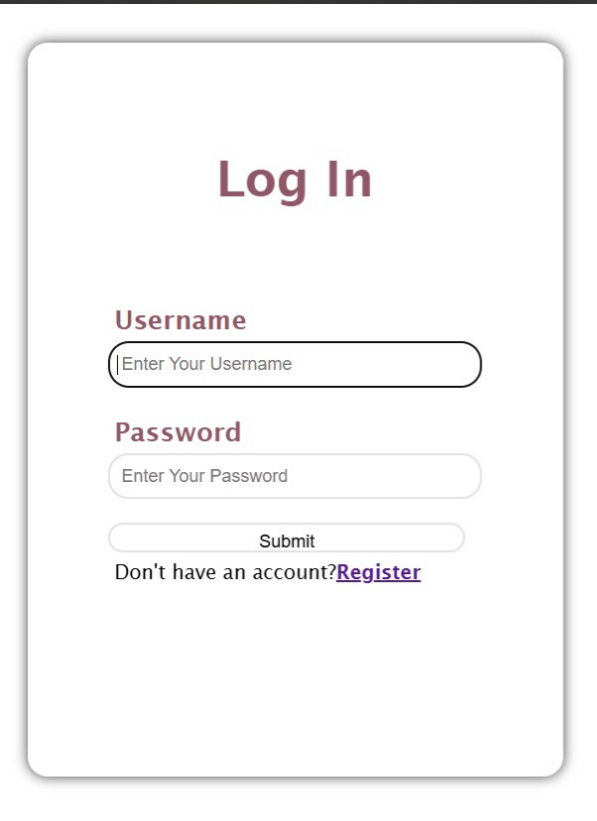
            }

        }

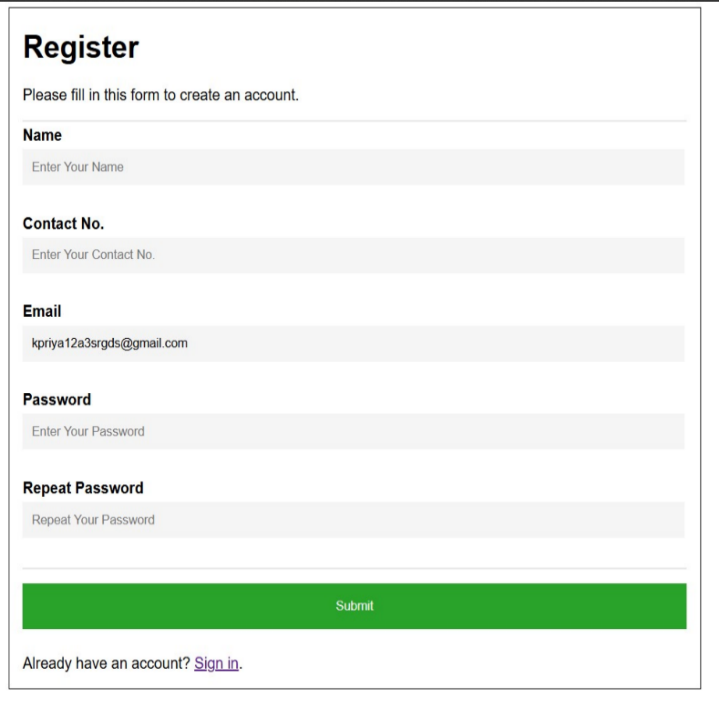
    }

}

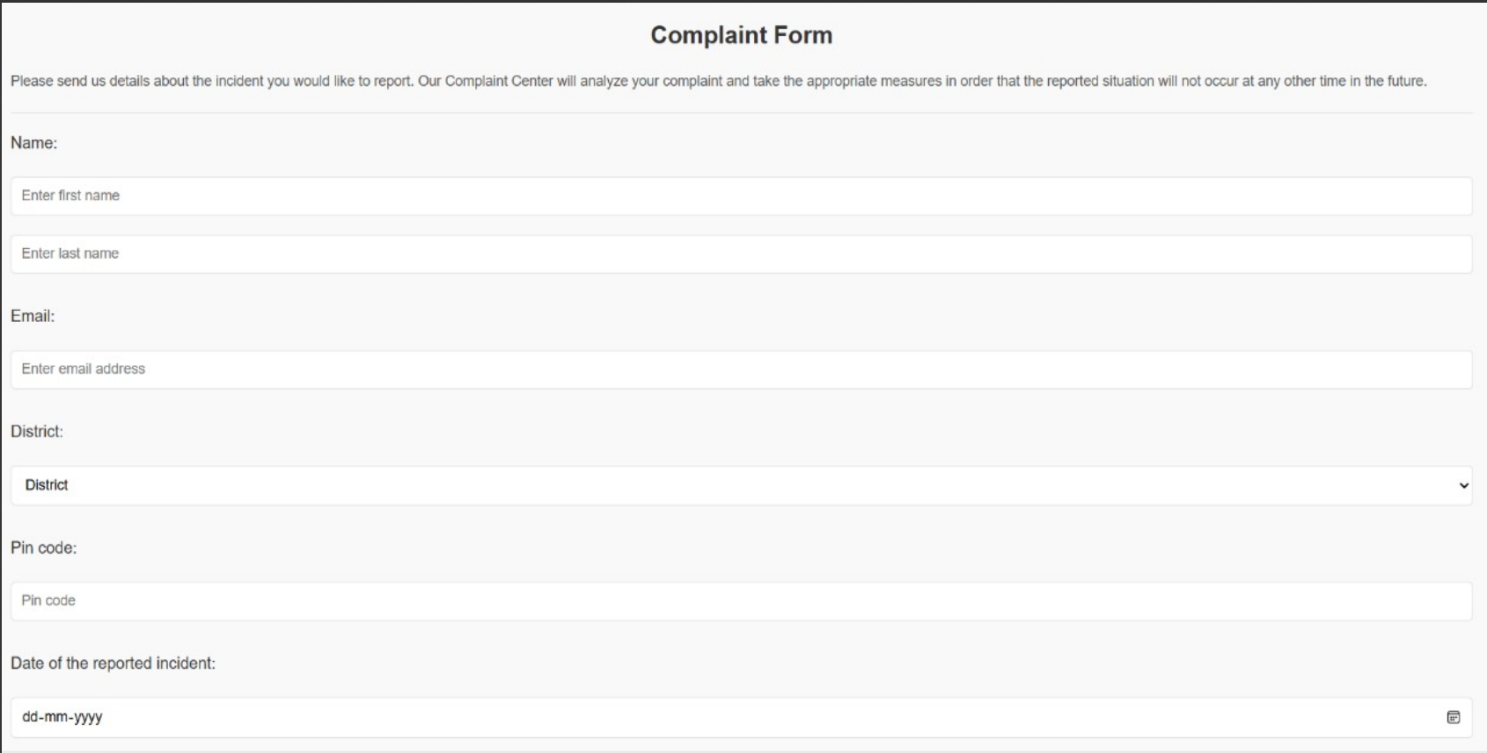
**6.3 APPENDICES**



**(fig:6.1)Login page**

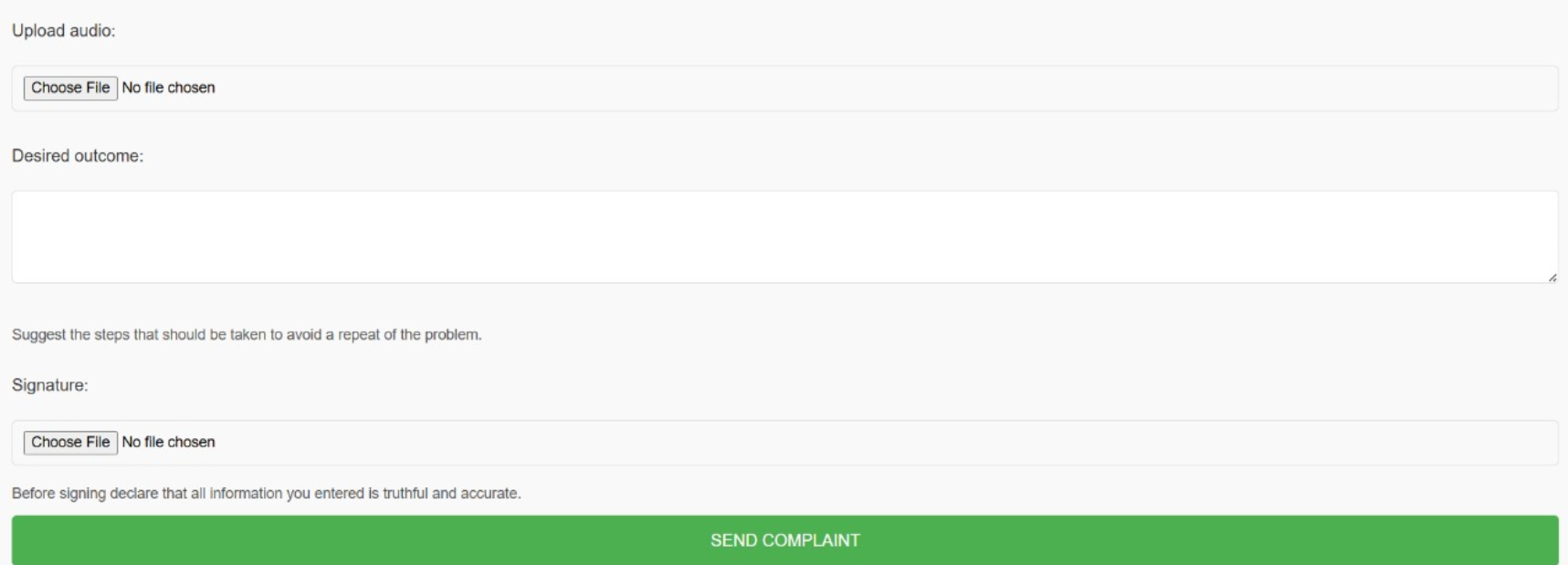


**(fig:6.2)Registration page**

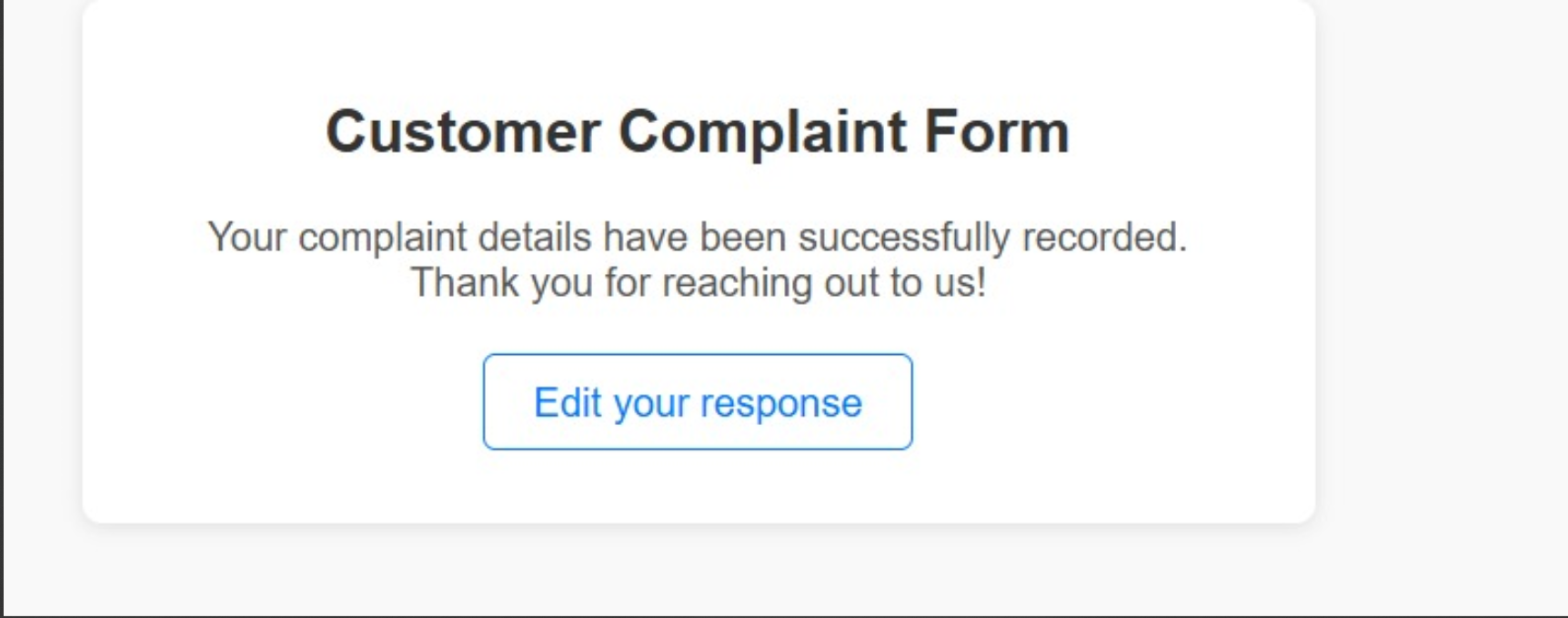
 **(fig:6.3a)Complaint page**



**(fig:6.3b)Complaint page**



**(fig:6.3c)Complaint page**



**(fig:6.4)Ticket page**

**7.SYSTEM TESTING**

**1. Functional Testing**

**Objective**: Ensure all features work as intended.

* **Voice Recognition Accuracy**:
  + Test with different accents, speech speeds, and background noise levels.
  + Verify the system correctly recognizes and processes complaints.
* **Input Validation**:
  + Ensure the system prompts for necessary information (e.g., contact info).
  + Test invalid inputs (e.g., nonsense phrases) and ensure the system handles them gracefully.
* **Confirmation Prompts**:
  + Check that the system accurately repeats back user inputs for confirmation.
  + Test handling of user corrections (e.g., “No, I meant…”).
* **Submission Process**:
  + Verify that complaints are successfully submitted and stored in the database.
  + Check for proper error messages if submission fails.

**2. Usability Testing**

**Objective**: Assess user experience and interface.

* **User Experience**:
  + Conduct user testing sessions to observe how intuitive the system is.
  + Gather feedback on voice prompts and clarity of instructions.
* **Accessibility**:
  + Ensure the system is usable by individuals with varying abilities (e.g., hearing impairments).
  + Test compatibility with assistive technologies.

**3. Performance Testing**

**Objective**: Assess system responsiveness and stability under load.

* **Load Testing**:
  + Simulate multiple users submitting complaints simultaneously to gauge system performance.
  + Monitor response times and system behavior under heavy usage.
* **Stress Testing**:
  + Push the system beyond normal operational capacity to identify breaking points.
  + Test recovery time after overload.

**4. Security Testing**

**Objective**: Ensure data protection and compliance.

* **Data Security**:
  + Test for vulnerabilities in data storage and transmission (e.g., encryption).
  + Verify access controls to ensure only authorized personnel can view or manage complaints.
* **Privacy Compliance**:
  + Ensure compliance with regulations (e.g., GDPR) regarding user data handling.
  + Check for proper user consent processes for data collection.

**5. Integration Testing**

**Objective**: Ensure all components work together seamlessly.

* **System Integration**:
  + Test integration with existing systems (e.g., CRM) to ensure data flows correctly.
  + Verify that notifications and alerts function as expected.

**6. Regression Testing**

**Objective**: Ensure new changes don’t affect existing functionality.

* After any updates or bug fixes, retest existing features to confirm they still work as intended.

**7. User Acceptance Testing (UAT)**

**Objective**: Validate the system meets user needs and requirements.

* Involve a group of end-users to test the system in a real-world scenario.
* Gather feedback and make necessary adjustments before full deployment.

**8. Documentation and Reporting**

* Maintain thorough documentation of all test cases, results, and any identified issues.
* Provide a final report summarizing the testing phases, outcomes, and any recommendations for improvements.

|  |
| --- |
| **8.CONCLUSION** |

The development of a voice-enabled complaint system represents a significant advancement in how organizations can receive and manage user feedback. By leveraging cutting-edge voice recognition technology and integrating it with robust backend systems, this project aims to streamline the complaint submission process, making it more accessible and user-friendly.

Throughout the project, we have identified key components that contribute to the system's functionality, including natural language processing, multilingual support, and enhanced user interface features. We have also outlined a comprehensive testing strategy to ensure reliability, usability, and security, as well as potential future enhancements that can further improve the system's effectiveness.

The successful implementation of this voice-enabled complaint system not only enhances user satisfaction but also equips organizations with valuable insights into customer concerns. By fostering a more responsive and transparent complaint management process, organizations can strengthen their relationships with users, drive continuous improvement, and ultimately enhance their overall service quality.

In summary, this project not only addresses current user needs but also sets the foundation for ongoing innovation and responsiveness in complaint management, positioning organizations to better adapt to the evolving landscape of customer service. With a focus on continuous feedback and enhancement, this system can evolve to meet future challenges and expectations, ensuring long-term success and user engagement.

**FUTURE ENHANCEMENTS**

Future enhancements for a voice-enabled complaint system can significantly improve user experience, functionality, and efficiency. Here are some ideas:

**1. Natural Language Processing (NLP) Enhancements**

* **Sentiment Analysis**: Implement NLP to assess the emotional tone of complaints, allowing for prioritized responses to urgent or sensitive issues.
* **Contextual Understanding**: Enhance the system’s ability to understand context, enabling more relevant responses and follow-up questions.

**2. Multilingual Support**

* **Language Options**: Expand the system to support multiple languages, catering to a more diverse user base.
* **Accent Recognition**: Improve recognition capabilities for various accents and dialects within supported languages.

**3. Advanced User Interface Features**

* **Visual Feedback**: Incorporate a visual component (e.g., a dashboard) that summarizes user complaints and their statuses.
* **Voice Biometrics**: Use voice recognition for user authentication, enhancing security and personalizing the experience.

**4. Integration with AI and Chatbots**

* **AI-Driven Suggestions**: Implement AI that can suggest resolutions based on common complaints or previous interactions.
* **Chatbot Integration**: Allow users to transition seamlessly from voice to text chat for more complex issues.

**5. Enhanced Reporting and Analytics**

* **Data Insights**: Provide advanced analytics tools for organizations to track complaint trends, response times, and resolutions.
* **Real-Time Dashboards**: Offer real-time monitoring of complaints and system performance metrics.

**6. Personalization Features**

* **User Profiles**: Create user profiles that remember previous complaints and preferences, enabling tailored interactions.
* **Follow-Up Notifications**: Implement automated follow-ups or reminders for users regarding their complaints.

**7. Feedback Loop Mechanism**

* **User Feedback Collection**: After a complaint is resolved, prompt users for feedback on their experience to continually improve the system.
* **Survey Integration**: Integrate surveys to gauge overall satisfaction and identify areas for improvement.

**8. Mobile Application Development**

* **Dedicated App**: Develop a mobile application to provide users with easier access to the voice complaint system on the go.
* **Push Notifications**: Use mobile notifications to update users on the status of their complaints.

**9. Integration with Other Communication Channels**

* **Cross-Channel Support**: Enable users to submit complaints via email, social media, or SMS, and allow voice input through these channels.
* **Unified Communication Platform**: Create a centralized platform where all complaints and communications are aggregated.

**10. AI-Powered Predictive Analytics**

* **Trend Prediction**: Utilize AI to predict future complaint trends based on historical data, allowing organizations to proactively address potential issues.

**11. Enhanced Security Measures**

* **Data Encryption**: Continuously update security protocols to protect user data.
* **Anomaly Detection**: Implement machine learning models to detect unusual patterns in complaints that may indicate systemic issues.

By focusing on these enhancements, you can create a more robust and user-friendly voice-enabled complaint system that adapts to the needs of users and organizations.

**REFERENCES:**

[1] Fornell C., & Westbrook A. Robert, “The Vicious Circle of Consumer Complaints”, journal of Marketing, (summer)(1984), 68-78.

[2] Razali R., Abd Halim K.N., & Jusoff K., “Quality Improvement of services in Unversiti Teknology Mara Pahang from a Management Perspective.” Management Science & Engineering Vol.5, No.1, (2011), pp.71-80.

[3] Najar, A.S., Al-Sukhni, H.A., & Aghakhani, N., “The Application of Service-oriented Architecture in Ecomplaint System.” Paper presented at (ICCSN ’10) the Second International Conference on Communication software and Networks, (2010, 26-28 [Feb.@2010](mailto:Feb.@2010)).

[4] Yi, Yooujae, “A Critical Review of Customer Satisfaction”. In Review of Markiting, Valerie A, Zeithaml, ed. Chicago, American Marketing Association, (1990).

[5] ESRAA ABD EL-AZIZ ABD EL-SADEK AFIFY, “A model for Customer Complaint Management system using SOA”, Management Information System Department, Modern Academy for Computer Science & Management Technology, CAIRO, EGYPT.

[6] http://www.ibimapublishing.com/journals/JMRCS/ 2012/624789/624789.pdf

[7] http://www.indiaurbanportal.in/b1estpractices/best practices81/bestpractices810.pdf

[8] http://www.freeproject.co.in/source/ComplaintManagement -System.aspx?pf=Java&t=web

[9] <http://grietinfo.in/projects/main/mca2012/cd-1-3doc.pdf>

[10] http://www.techzoo.org/projects/online-complainmanagement-system-project.html