|  |  |
| --- | --- |
|  | Design Specifications  Document – Call Center (Monster UI) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Control** | | | |
| **Reference No.** |  | | |
| **Document Name** | **Design Document** | | |
| **Version No.** | 1.0 | | |
| **Document Status** | Draft | | |
| **Issue Date** | 06 June 2023 | | |
| **Compliance Status** | Mandatory | | |
| **Review Period** | As and when required | | |
| **Security Classification** | Restricted | | |
| **Distribution** | Aleem/Anurag | | |
|  | **Name** | **Role** | **Signature** |
| **Authored by** | Meenakshi Kaul | SDM |  |
| **Reviewed by** | Anurag Saproo | CTO |  |
| **Approved by** |  |  |  |
| **Released by** |  |  |  |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **Document Revision History** | | |
| **Version** | **Release Date** | **Change Description** |
| 1.0 | 06/06/2023 | Draft |
|  |  |  |

## 1.**Agent Call Report Feature Design**

**Overview**

The Agent Call Report feature provides a comprehensive report on agents' call activities based on specific queues. The report displays data for each agent's call performance, including their first name, total calls, answered calls and missed calls, filtered by the selected queues. The report is presented in a tabular format with customizable styling options.

**User Interface**

The user interface consists of a table displaying the agent call data. The table has a responsive design and adjusts its width to fit the container. The header row contains column names: "First Name", "Total Calls", and "Missed Calls". If no data is available for the selected queues, a message is displayed in the table body.

**Functionality**

### **Data Retrieval:**

1. The feature retrieves agent call data from an API endpoint based on the selected queues using an authenticated **HTTP GET request**.
2. The selected queues are passed as parameters in the API request to filter the data accordingly.
3. The data is expected to be an array of objects, with each object containing the agent's first name, total calls, and missed calls for the specified queues.
4. If the data is empty or not in the expected format, a "No data available" message is displayed in the table.

### **Table Creation:**

1. The table is dynamically created using JavaScript DOM manipulation.
2. The table structure includes a **<table>** element, **<thead>**, **<tbody>**, and **<th>** and **<td>** cells.
3. The table is styled with a border and adjusts its width to fit the container.
4. Table Header:
5. The header row is created dynamically.
6. The column names ("First Name", "Total Calls", and "Missed Calls") are displayed in **<th>** elements.
7. The header cells are left-aligned with a left padding of 10 pixels.
8. The header row has a single **<th>** element with **colspan** set to 3, displaying the header name "Agent Call Report".
9. Table Body:
10. The table body is populated with data retrieved from the API for the selected queues.
11. Each agent's information is displayed in a separate row.
12. The agent's first name is displayed in the first column, and the total calls and missed calls for the selected queues are displayed in the respective columns.
13. The table cells have a left padding of 10 pixels and a border.

**Error Handling:**

If there is an error in retrieving the data from the API or the data format is incorrect, an appropriate error message is displayed instead of the table.

**Technical Implementation Details**

* The feature utilizes JavaScript for dynamic table creation and DOM manipulation.
* API requests are made using the Fetch API with appropriate headers for authentication.
* The selected queues are passed as parameters in the API request to filter the data.
* The styling of the table is achieved using CSS properties such as **border**, **width**, and **text-align**.

**Future Enhancements**

* Sorting and filtering options for the table columns to allow users to customize the view.
* Pagination or infinite scrolling for large datasets to improve performance and user experience.
* Export functionality to download the report in different formats such as CSV or Excel.
* This design document provides an overview of the Agent Call Report feature, including its user interface, functionality, technical implementation details, and potential future enhancements. It serves as a guide for development and can be referenced during the implementation process.

# 2.Call Handling Continuity for Logged-in Agents

### **Overview**

This feature enhancement aims to ensure that logged-in agents remain active and available even when another agent receives a call that was initially assigned to them but went unanswered. Instead of automatically logging out the agent who missed the call, the system will keep them logged in and ready to receive subsequent calls.

### **Current Behaviour**

When a call is assigned to an agent and they do not answer or reject the call, the system considers it an unanswered call.

The call is then routed to another available agent, and they handle the call successfully.

Meanwhile, the initially assigned agent is automatically logged out by the system, requiring them to log in again to receive further calls.

### **Proposed Behaviour**

When a call is assigned to an agent and they do not answer or reject the call, the system identifies another available agent to handle the call.

The call is seamlessly transferred to the available agent without logging out the initially assigned agent.

The initially assigned agent remains logged in and ready to receive subsequent calls.

**Technical Implementation Details**

* Call Monitoring: The system continuously monitors the status of ongoing calls and tracks the availability of logged-in agents.
* Call Routing and Transfer: The system automatically transfers the call to another available agent when the initially assigned agent does not respond within a specified time or rejects the call.
* Agent Status Management: The system keeps track of the availability and activity of agents, distinguishing between agents who are actively handling a call and those who are available but currently idle.
* Call Queue Management: If multiple calls are waiting in the queue, the system should prioritize and route the calls efficiently, ensuring a fair distribution among available agents.
* Agent Notifications: The system can provide notifications to agents to indicate that a call assigned to them has been redirected to another agent.
* Agent Re-Assignment: If an agent becomes available again after missing a call, they are automatically assigned new incoming calls without the need to log in again.
* User Interface Updates: The agent interface should reflect the call transfer and clearly indicate when a call is redirected to another agent, ensuring transparency, and minimizing confusion.
* Error Handling and Logging: The system should handle any errors or exceptions that may occur during the call transfer process, log relevant information for debugging and analysis, and provide appropriate feedback to agents.

**Benefits**

* Seamless Call Handling: Agents remain logged in even if they miss a call, ensuring continuity in their availability for subsequent calls.
* Improved Efficiency: Agents can handle calls without interruptions, eliminating the need for repeated logins and minimizing downtime.
* Enhanced Agent Experience: Agents can focus on handling calls rather than managing their login status, leading to a smoother and more productive workflow.
* Optimal Resource Utilization: The system efficiently distributes calls among available agents, maximizing the utilization of resources and reducing wait times for callers.

Please note that the implementation of this feature enhancement may require adjustments to your existing call routing and session management system.