

# Chat Application Design Document

This report defines the planned user interface (UI) and fundamental features of a chat solution capable of managing visitor conversations effectively. The goal is to offer a streamlined, easy-to-understand experience for agents serving multiple customer conversations.

## 1. Overall Layout: Three-Column Structure

The chat window will be split logically into three separate vertical parts to minimize information display and user workflow:

- **Column 1 (Left): Visitor List**
  - Shows a list of all currently logged-in visitors.
  - Acts as the main navigation for alternating between conversations.
- **Column 2 (Middle): Current Selected Chat**
  - Committed to the active conversation with the currently chosen visitor.
  - Where live messages are presented and agents type responses.
- **Column 3 (Right): Visitor Profile & History**
  - Offers contextual information regarding the chosen visitor.
  - Displays a summary of their recent chat history.

## 2. Column 1: Visitor List

This column will serve as the central point for managing incoming and live chats.

- **Visitor Information:** Each list item will display significant visitor information (e.g., visitor name/ID, last message snippet, Image).
- **Single Selection:** There can be only one visitor at any given time. The entry of the selected visitor will be highlighted visually to indicate the current conversation.
- **New Message Indicator (Unselected Visitor):**
  - If a visitor has a new message who is not currently selected, a counter will be displayed against his/her chat entry in this column.
  - This counter will be followed by a highlighter that is readily noticeable (e.g., a contrasting background color, a glowing appearance) to immediately bring attention to the unread message.
  - The highlight will continue until the visitor is selected, and the counter will be reset thereafter.

## 3. Column 2: Current Selected Chat

It is the central point of interaction for the agent.

- **Visibility:** Only the chat messages of the currently selected visitor will be visible here.
- **New Message Handling (Selected Visitor):**

- Whenever a fresh message from one of the already selected guests is sent, it will be added to the chat messages that are shown in this column immediately.
- **Message Highlighting & Fading (Customer Messages):**
  - New incoming messages from the customer side will be momentarily highlighted (e.g., brighter color background, a small animation) to indicate their newness.
  - This highlight will subsequently decline, after a short, user-controllable interval, merging the message in with earlier sent, regular messages.
  - The message will also lose its highlight and shade out if the agent responds back to that specific message or the conversation progresses.
- **Message Alignment:**
  - **Customer Messages:** Will be aligned to one side (e.g., left-aligned) with a clear background color.
  - **Agent Messages:** Will be on the opposite side (e.g., right-aligned) with a contrasting background color so that agent and customer messages can be easily visually differentiated.

## 4. Column 3: Visitor Profile & Last 5 Chat History

The column provides background and immediate access to historical facts.

- **Visitor Profile:** Displays vital information of the selected visitor (e.g., name, account status, last activity).
- **Last 5 Chat History:** provides a short list of the last five messages with the selected visitor, with immediate context without having to scroll throughout the entire conversation.
- **Hover Interactions on History:**
  - If the user positions the cursor over any message within the "Last 5 Chat History" section, there will be two distinctive icons displayed:
    - **Chat Details Icon:** (e.g., an "i" for information, or a magnifying glass)
    - **Chat Summary Icon:** (e.g., a document icon, or a bulleted list icon)
- **Modal Window on Icon Click:**
  - Clicking on either the "Chat Details" or "Chat Summary" button will trigger a modal window to pop up.
  - **Chat Details Modal:** Will display complete information of the specific hovered message (complete timestamp, message ID, any associated metadata).
  - **Chat Summary Modal:** Will give a short AI-produced summary of the chat to this point, or of a given portion of the chat, depending on whether it is used.

## 5. Summary of Key Interactions & Features

- Live output of current conversations.
- Clear notification of unread messages from unselected visitors.
- Dynamic highlighting and fading for new incoming messages.
- Unambiguous visual differentiation between customer and agent messages.
- Instant access to visitor profile and last chat history.
- In-depth overview on request and AI-generated summary of conversation history via modal windows.

## 6. Action Points

Based on this framework, the following action points are proposed:

1. **Detailed UI/UX Mockups:** Create design wireframes and high-fidelity mockups of the three columns and the modal windows, for example, new message, selected visitors, and hover states.
2. **Front-end Development:**
  - Implement the three-column responsive layout.
  - Design visitor list sections, chat message display, and visitor profile.
  - Implement real-time append and highlight logic for messages.
  - Include the hover effect and the modal window function in chat history.
3. **Back-end Integration:**
  - Set up real-time communication (e.g., WebSockets) for message exchange.
  - Develop APIs to get visitor lists, chat history, and visitor profiles.
  - Integrate with an AI chat summary generation service.
4. **Database Design:** Design the database schema to hold visitor information, chat messages, and user profiles.
5. **Testing:** Perform rigorous testing for functionality, responsiveness, and user experience.

## Data Structures (Object Models)

We will determine the highest-level entities and attributes. These will map one-to-one to database schema and API response structure.

### 1. Visitor Object

Represents a single user who is conversing with the application.

Property Name	Data Type	Description
visitorId	String (UUID)	Unique identifier for the visitor.
name	String	Display name or ID of the visitor (e.g., "Guest-123", "Abhay").
email	String	Visitor's email address (optional, for profile).
accountStatus	String	Current status of the visitor's account (e.g., "Active", "Premium", "Guest").
lastActivity	DateTime	Timestamp of the visitor's last interaction.
isActive	Boolean	True if currently logged in/active in a chat session.
unreadMessageCount	Integer	Number of unread messages from this visitor (for agents).
lastMessageSnippet	String	A short snippet of the last message sent by this visitor.
profileImageUrl	String (URL)	URL to the visitor's profile picture/avatar.
customAttributes	JSON/Map	Flexible field for additional visitor-specific data (e.g., "MembershipTier").

currentChatSessionId	String (UUID)	ID of the currently active chat session with this visitor.
----------------------	---------------	--

## 2. Agent Object

Represents a customer service agent using the application.

Property Name	Data Type	Description
agentId	String (UUID)	Unique identifier for the agent.
name	String	Agent's name.
email	String	Agent's email address.
status	String	Agent's availability status (e.g., "Online", "Offline").
assignedVisitorIds	Array of String	List of visitorIds currently assigned to this agent.

## 3. ChatSession Object

Represents a single conversation thread between an agent and a visitor.

Property Name	Data Type	Description
sessionId	String (UUID)	Unique identifier for the chat session.
visitorId	String (UUID)	ID of the visitor participating in this session.
agentId	String (UUID)	ID of the agent participating in this session (can be null if unassigned).
startTime	DateTime	Timestamp when the session started.
endTime	DateTime	Timestamp when the session ended (can be null if active).
status	String	Current status of the session (e.g., "Open", "Closed", "Pending").
lastMessageTimestamp	DateTime	Timestamp of the last message in this session.

#### 4. ChatMessage Object

Represents an individual message within a chat session.

Property Name	Data Type	Description
messageId	String (UUID)	Unique identifier for the message.
sessionId	String (UUID)	ID of the chat session this message belongs to.
senderId	String (UUID)	ID of the sender (either visitorId or agentId).
senderType	String	Type of sender ("Visitor" or "Agent").
timestamp	DateTime	Timestamp when the message was sent.
content	String	The actual message text.
isReadByReceiver	Boolean	True if the message has been read by the recipient.
isHighlighted	Boolean	Transient property for UI highlighting (not typically stored in DB).
metadata	JSON/Map	Additional message-specific data (e.g., attachmentUrls, messageType).

**5. ChatHistorySummary Object (for AI-generated summaries)**

This object would typically be generated on-demand or pre-processed and stored, linking to a specific chat session or portion.

Property Name	Data Type	Description
summaryId	String (UUID)	Unique identifier for the summary.
sessionId	String (UUID)	ID of the chat session this summary pertains to.
summaryText	String	The AI-generated summary of the chat.
generationTimestamp	DateTime	Timestamp when the summary was generated.
startMessageId	String (UUID)	Optional: ID of the first message included in the summary.
endMessageId	String (UUID)	Optional: ID of the last message included in the summary.

## Data Flow and Object Changes based on Actions

This section describes how the objects that were defined earlier are changed or manipulated when some operations are executed in the application.

### Action: Agent Logs In

- **Agent Object:** status is updated from "Offline" to "Online".
- **System Action:** Backend retrieves all "Open" or "Pending" ChatSessions and their respective Visitor and ChatMessage data for the agent.

### Action: New Visitor Opens Chat

- **Visitor Object:** A new instance of the Visitor object is created with visitorId, name (such as "Guest-123"), isActive set to true, and unreadMessageCount set to 0. lastActivity is started with the current time.
- **ChatSession Object:** We create a new ChatSession object with sessionId, visitorId to the new visitor, status as "Pending", and startTime to the current time. agentId would be null at first.
- **ChatMessage Object:** The first message from the visitor creates a ChatMessage object with sessionId, senderId (visitor's ID), senderType "Visitor", timestamp, and content.



isReadByReceiver will be false.

- **Column 1 (Visitor List):** The Visitor is displayed in the list. If an agent is not selected, this visitor can be auto-selected (rules-dependent). If an agent is selected, the unreadMessageCount in the visitor's record in Column 1 increases, and the record gets highlighted.

### Action: Agent Selects a Visitor from Column 1

- **Agent Object:** The assignedVisitorIds of the current agent is also updated to include the visitorId of the newly selected visitor. currentChatSessionId for the selected visitor is initialized.
- **Visitor Object:** The unreadMessageCount of the current Visitor is reset to 0. The visual highlight in Column 1 is cleared.
- **ChatSession Object:** If the ChatSession was previously in "Pending" status and now contains an agentId, its status is changed to "Open".
- **Column 2 (Current Selected Chat):** Contains all ChatMessage objects of the current Selected ChatSession, ordered by timestamp.
- **Column 3 (Visitor Profile & History):** Populated with the data of the selected Visitor object and the last 5 ChatMessage objects in their ChatSession.

### Action: Visitor Sends a New Message (to currently selected chat)

- **ChatMessage Object:** Create a new ChatMessage object and store it in the database.
- **ChatSession Object:** lastMessageTimestamp is set.
- **Column 2 (Current Selected Chat):** It is placed on the screen right away. Its isHighlighted flag is set temporarily to true (for UI purposes, not DB) to trigger visual highlight. Within some small time or agent response, the highlight is cleared (the UI element's isHighlighted flag is reset to false).
- **UI State:** Scroll position in Column 2 is updated to show the latest messages.

### Action: Visitor Sends a New Message (to unselected chat)

- **ChatMessage Object:** A new ChatMessage object instance is created and persisted.
- **ChatSession Object:** lastMessageTimestamp is assigned.
- **Visitor Object:** the unreadMessageCount of the above-mentioned visitor is increased. lastMessageSnippet is updated.
- **Column 1 (Visitor List):** The unreadMessageCount indicated with the visitor's entry is refreshed, and an on-screen highlight is placed on the entry.

### Action: Agent Sends a Message

- **ChatMessage Object:** A new ChatMessage object is stored, with senderType "Agent".
- **ChatSession Object:** lastMessageTimestamp is updated.
- **Column 2 (Current Selected Chat):** The new message is included immediately. Highlighted customer messages that were previously selected are unhighlighted (UI isHighlighted for such messages is set to false) as the conversation proceeds.
- **UI State:** Column 2's scrollbar position is set to the position of the last message..

## Action: Agent Hovers over a Message in "Last 5 Chat History" (Column 3)

- **UI State:** Both "Chat Details Icon" and "Chat Summary Icon" are displayed for that particular message.
- No data object is altered, only UI interaction.

## Action: Agent chooses "Chat Details Icon"

- **ChatMessage Object:** The specific ChatMessage object is retrieved using its messageId.
- **UI State:** A modal window is shown with the timestamp, messageId, and metadata of the ChatMessage that was loaded.

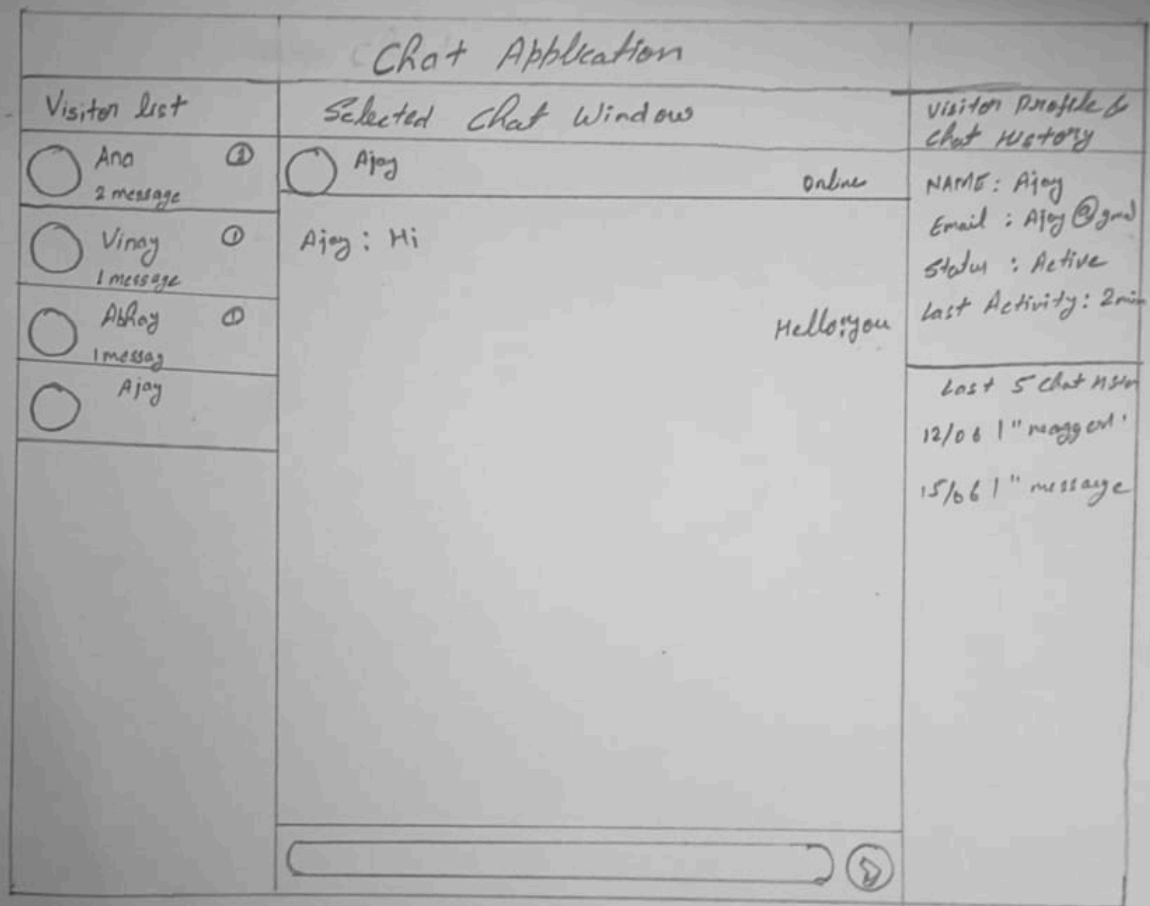
## Action: Agent clicks on "Chat Summary Icon"

- **ChatSession Object / ChatMessage Objects:** The system calculates the relevant part of the chat (e.g., the last minutes, or the entire session up to the hovered message).
- **Backend / AI Integration:** The AI chat summary generation service is called, with the content of the relevant ChatMessage objects being passed in.
- **ChatHistorySummary Object (Optional):** IChatHistorySummary Object (optional): If the summaries are cached or pre-computed, a cached ChatHistorySummary can be retrieved. Otherwise, a new one may be computed and temporarily or persistently stored.
- **UI State:** A modal window comes up, displaying the summaryText of the AI-generated summary.

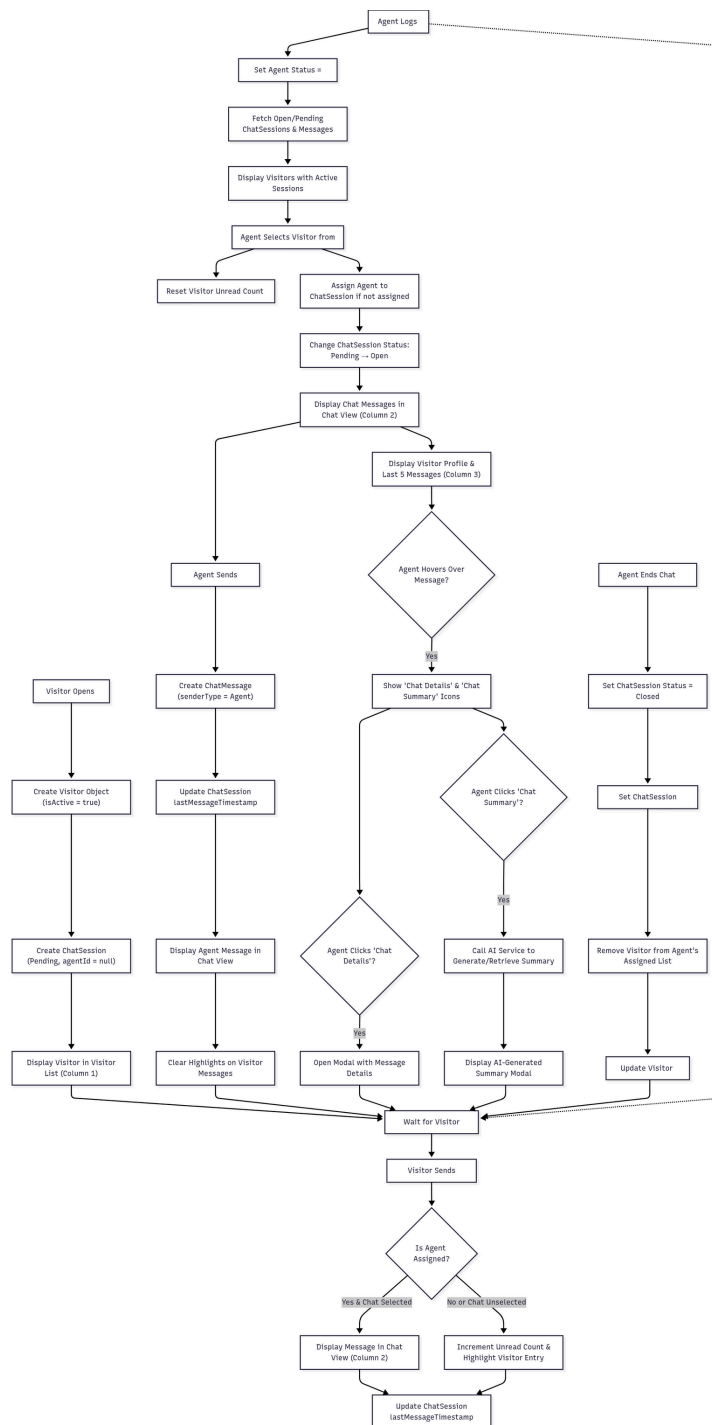
## Action: Agent Finished Chat Session

- **ChatSession Object:** status is updated from "Open" to "Closed". endTime is set to the current time.
- **Agent Object:** The visitorId of the ending session is removed from the assignedVisitorIds of the agent.
- **Visitor Object:** isActive for the current visitor can be made false if no other active sessions exist.
- **Column 1 (Visitor List):** The closed chat session (and possibly the visitor, per UI rules for closed chats) is deleted or indicated as closed visually.

## Wireframe



Traditional flow diagram



Link - [🌐 Mermaid Chart](#)

## Reference Websites for Research

### 1. Zendesk Chat

- URL: <https://www.zendesk.com/chat/>
- Features: Visitor list, chat window, visitor profile, unread message indicators.

### 2. LiveChat

- URL: <https://www.livechat.com/>
- Features: Multi-visitor chat interface, message highlighting, visitor info panel.

### 3. Intercom

- URL: <https://www.intercom.com/live-chat>
- Features: Visitor list, conversation view, profile/history sidebar.

### 4. Freshdesk Messaging (formerly Freshchat)

- URL: <https://freshdesk.com/messaging>
- Features: Visitor list, chat window, message notifications.

### 5. WhatsApp Messenger

- URL: <https://freshdesk.com/messaging>
- Features: Visitor list, chat window, message notifications.