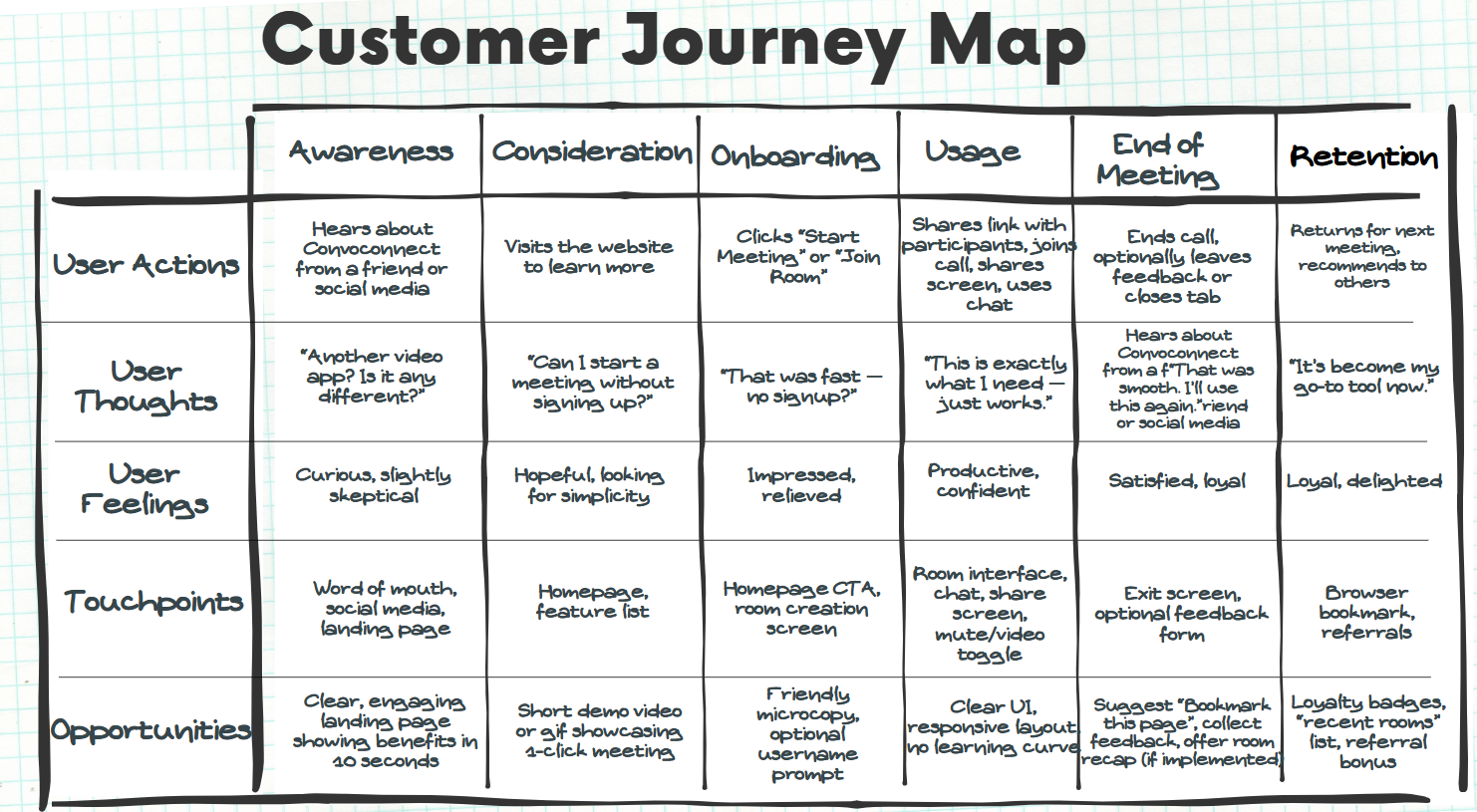
**REQUIREMENT ANALYSIS**

**3.1 Customer Journey map**

|  |  |
| --- | --- |
| Date | 3rd, April 2025 |
| Team ID |  |
| Project Name | Convoconnect |
| Maximum Marks | 2 Marks |



**3.2 Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 3rd, April 2025 |
| Team ID |  |
| Project Name | Convoconnect |
| Maximum Marks | 2 Marks |

**Functional Requirements:**

**Following are the functional requirements of the proposed solution.**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| **FR-1** | **Room Management** | One-click room creation  Join via room link  Auto-generated unique room ID |
| **FR-2** | **Real-Time Communication** | Peer-to-peer video and audio via WebRTC  Screen sharing  Mute/unmute mic and video toggle |
| **FR-3** | **Chat Functionality** | Text chat within the meeting room  Display user names with messages |
| FR-4 | **Room Security** | Optional passcode protection  Room auto-expiry after session ends |

**Non-functional Requirements:**

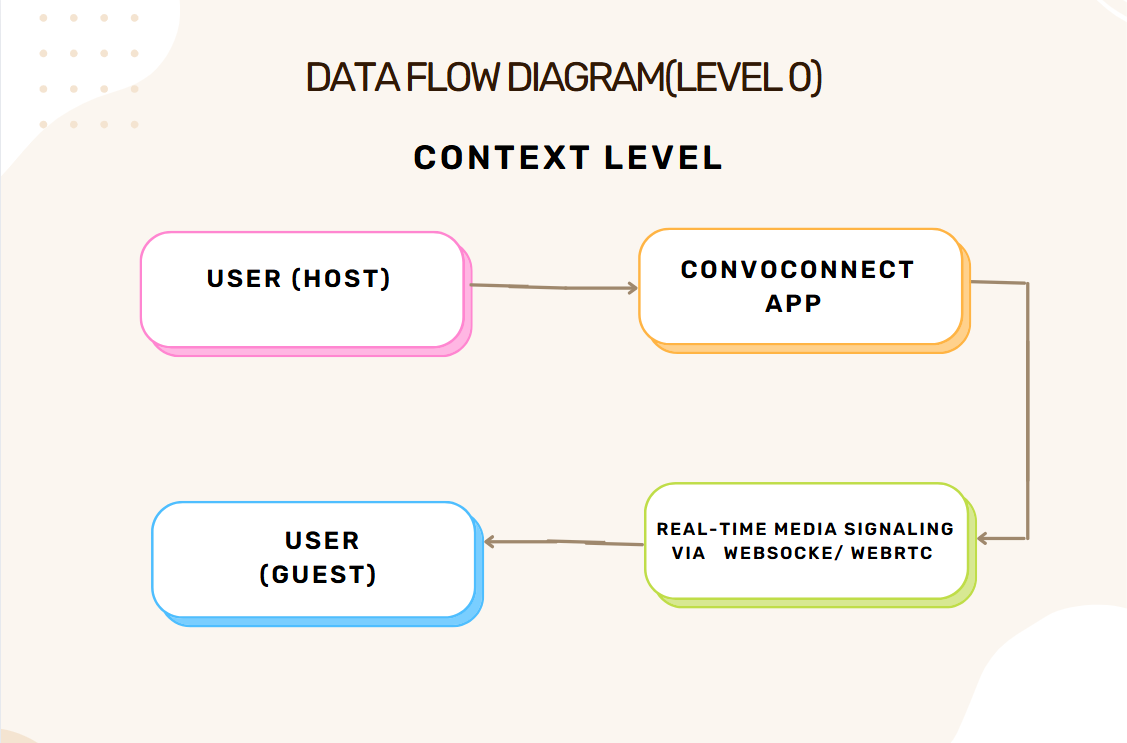
**Following are the non-functional requirements of the proposed solution.**

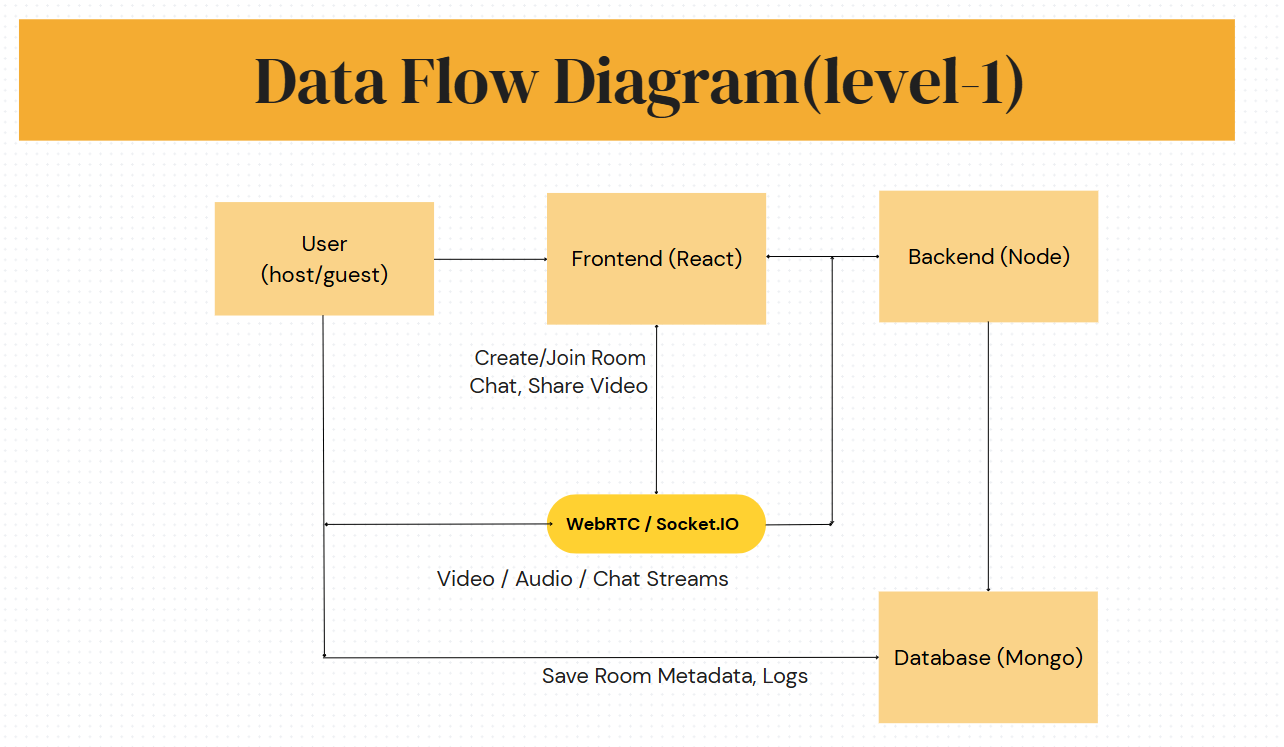
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR**-1** | Usability | Minimal, intuitive UI requiring no learning curve |
| NFR**-2** | Security | End-to-end encrypted media streams, no unnecessary data collection |
| NFR**-3** | Reliability | Stable performance in varying network conditions |
| NFR-4 | Performance | Fast room joining and low-latency communication |
| NFR-5 | Availability | Accessible across modern browsers and mobile devices |
| NFR-6 | Scalability | Support multiple rooms and up to 10–15 users per session |

**3.3 Data Flow Diagram & User Stories**

|  |  |
| --- | --- |
| Date | 3rd, April 2025 |
| Team ID |  |
| Project Name | Convoconnect |
| Maximum Marks | 2 Marks |

**Data Flow Diagrams:**





**User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
| Priya, College Student | Room Creation | USN-1 | As a user, I can create a video call room with a single click and get a shareable link. | Room is created and link is generated instantly | High | Sprint-1 |
| Rakesh, Freelance Consultant | Join Room | USN-2 | As a user, I can join a video room via link without needing to log in. | User enters room with working video/audio | High | Sprint-1 |
| Meera, Online Educator | Screen Sharing | USN-3 | As a user, I can share my screen during a call to present my content. | Participants can view the shared screen in real time | Medium | Sprint-2 |
| Aman, Small Business Owner | Chat Functionality | USN-4 | As a user, I can send and receive messages in the meeting room chat. | Messages appear instantly and are visible to all | Medium | Sprint-2 |

**3.4 Technology Stack**

|  |  |
| --- | --- |
| Date | 3rd, April 2025 |
| Team ID |  |
| Project Name | Convoconnect |
| Maximum Marks | 2 Marks |

**🖥️ Frontend (Client-Side)**

|  |  |
| --- | --- |
| **Tech** | **Purpose** |
| **React.js** | Building dynamic and responsive UI components |
| **React Router** | For page routing (Home, Room, etc.) |
| **Tailwind CSS** | Utility-first styling framework for UI |
| **Axios** | To make HTTP requests to backend (e.g., create/join room) |
| **Socket.IO Client** | Real-time communication for signaling/chat |
| **WebRTC APIs** | Handling video/audio/screen stream directly in the browser |

**⚙️ Backend (Server-Side)**

|  |  |
| --- | --- |
| **Tech** | **Purpose** |
| **Node.js** | Runtime for server-side JavaScript |
| **Express.js** | Web framework to handle routing and APIs |
| **Socket.IO Server** | To manage real-time connections and signaling |
| **WebRTC Signaling Logic** | Facilitates peer-to-peer media exchange setup |
| **JWT (optional)** | For secure token-based authentication (if added) |

**🗃️ Database**

|  |  |
| --- | --- |
| **Tech** | **Purpose** |
| **MongoDB** | NoSQL database for storing room data, chat history, and metadata |
| **Mongoose** | ODM library for schema-based modeling with MongoDB |

**📦 DevOps & Deployment**

|  |  |
| --- | --- |
| **Tech** | **Purpose** |
| **Vercel / Netlify** | Hosting the frontend (React app) |
| **Render / Railway / Heroku** | Hosting the backend server and WebSocket server |
| **MongoDB Atlas** | Cloud-based MongoDB database |
| **GitHub** | Version control and source code management |