Waterfall Model Project

Unified Customer Support Platform

Document 1

**Question 1: Why is this project initiated?**

This project is initiated to build a cost effective, multichannel customer support platform that supports chat, email and calling features all in one.

**Question 2: What are the current problems?**

The current problems are:

* Different 3rd party applications for chat, email and calls. Making it expensive to subscribe and maintain.
* As there are different platforms, KPI, reporting and analysis of agent performance and other metrics becomes a lengthy process.
* The current system is not AI integrated.

**Question 3: With this project how many problems can be solved?**

This project has the capacity to resolve all the three major issues mentioned above. This would be an indigenous product and can be customized later as per requirement.

**Question 4: What are the resources required?**

The resources required are: Lead BA, BA, Front-end developer, backend developer, AI engineer, DB Admin, Devops Engineer, QA, BI developer, Stakeholders.

**Question 5: How much organizational change is required to adopt this technology?**

A moderate to high level organizational changes is required to adopt this technology.

It involves:

* Restructuring workflows to accommodate AI-based ticket routing and multi-channel support.
* Training staff on new tools and dashboards.
* Migrating data from legacy systems to the new platform.
* Aligning teams (support, IT, dev, QA) for a unified response strategy.
* Updating compliance protocols to meet regulatory requirements when using integrated AI and cloud solutions.

**Question 6: Time frame to recover ROI?**

The time frame to recover ROI is 9-12 months after the software has been implemented.

**Question 7: How to identify stakeholders?**

To identify stakeholder’s, we will be using RACI matrix and analyze the roles in current system.

Document 2

* **Elicitation Technique:** 
  + Document Analysis: Since we already have a 3rd party application is use, we can do document analysis to get functional requirements for our indigenous application.
  + Interview: This is used to get information of what additional functionalities stakeholders wants to have that is not in the current system.
  + JAD session: JAD sessions will help us to guide on how should we proceed with the development and what is required by the development team before proceeding for development of the project.
* **Stakeholder Analysis:** 
  + RACI matrix technique: Since we also have an external client as a stakeholder, I have decided to proceed with RACI technique to do stakeholder analysis.
* **Documents to write:** 
  + Functional requirement documents: Majority of functional requirements are collected from document analysis. Requirements that are not available in current system, but needs to be implemented in upcoming application, will be collected by organizing stakeholder interviews.
  + RACI chart: This chart will display the stakeholder’s analysis (Responsible, Accountable, Consulted, Informed).
  + Feasibility study: This is done to check if the current system and environment is feasible for developing the upcoming application or not.
  + Risk analysis: Risk analysis is done to avoid delay or failure of the project.
  + Sign-off documents: Sign-off documents is for stakeholders’ approval to move into next step and will be taken through email.
  + Change tracker document: The requested changes will be documented in change tracker document and this will help us in keeping the track of change requests.
  + MoM: Minutes of Meetings will be sent out to all stakeholders after each meeting.
* **Process to follow to Sign-off documents:** After the updates had been shared with stakeholders through email, sign-offs will be requested in same email chain.
* **How to take approval from client:** Through email.
* **Communication channels to establish and implement:** Email and group chat (Slack/ MS teams).
* **Handle change requests:** Since we are working in a traditional waterfall model, making a large number of changes won’t be possible, for limited number of change requests, these will be documented in change tracker document.
* **Update progress to stakeholders:** Through email, weekly meetings.
* **Signoff on UAT:** Through email.
* **Client Project Acceptance Form:** Through email.

Document 3

**Functional Specifications:**

|  |  |
| --- | --- |
| Project Name | Unified Customer Relation Management Platform |
| Customer Name | IN-Energize |
| Project Version | 1.0 |
| Project Sponsor | IN-Energize |
| Project Manager | Pradeep Nair |
| Project Initiation date | 4/15/2025 |

**Functional Requirements Specifications:**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Name | Requirement Description | Priority |
| FR0001 | Multi-Channel Communication | The system must support real-time chat, email ticketing, and VoIP-based customer calling. | 10 |
| FR0002 | Ticket Management | The platform should allow creation, assignment, status tracking, and closure of tickets. | 10 |
| FR0003 | Smart Routing | AI should automatically route tickets to appropriate agents based on category or sentiment. | 9 |
| FR0004 | Integration Framework | Must integrate with tools like Jira, Azure DevOps, Slack, and CRM systems via API. | 9 |
| FR0005 | Knowledge Base | Users and agents should have access to a searchable knowledge base for self-service. | 6 |
| FR0006 | AI Chatbot | An NLP-based chatbot should answer FAQs and generate tickets for complex queries. | 8 |
| FR0007 | Analytics Dashboard | Provide real-time and historical insights on KPIs such as SLA adherence and agent activity. | 9 |
| FR0008 | Role-Based Access Control | Restrict access and permissions based on user roles (Admin, Agent, Viewer, etc.). | 8 |
| FR0009 | SLA & Escalation Handling | Support definition of SLA policies and automatic escalation of overdue tickets. | 7 |
| FR0010 | Notification System | Trigger email/SMS alerts for ticket status changes, escalations, and system errors. | 6 |
| FR0011 | Customer Feedback Capture | Allow users to rate support experience and leave comments after ticket resolution. | 4 |

Document 4

**Requirement Traceability Matrix:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req. ID | Req. Name | Req. Description | Design | Development | Testing | UAT |
| FR0001 | Multi-Channel Communication | The system must support real-time chat, email ticketing, and VoIP-based customer calling. | Pending | Pending | Pending | Pending |
| FR0002 | Ticket Management | The platform should allow creation, assignment, status tracking, and closure of tickets. | Pending | Pending | Pending | Pending |
| FR0003 | Smart Routing | AI should automatically route tickets to appropriate agents based on category or sentiment. | Pending | Pending | Pending | Pending |
| FR0004 | Integration Framework | Must integrate with tools like Jira, Azure DevOps, Slack, and CRM systems via API. | Pending | Pending | Pending | Pending |
| FR0005 | Knowledge Base | Users and agents should have access to a searchable knowledge base for self-service. | Pending | Pending | Pending | Pending |
| FR0006 | AI Chatbot | An NLP-based chatbot should answer FAQs and generate tickets for complex queries. | Pending | Pending | Pending | Pending |
| FR0007 | Analytics Dashboard | Provide real-time and historical insights on KPIs such as SLA adherence and agent activity. | Pending | Pending | Pending | Pending |
| FR0008 | Role-Based Access Control | Restrict access and permissions based on user roles (Admin, Agent, Viewer, etc.). | Pending | Pending | Pending | Pending |
| FR0009 | SLA & Escalation Handling | Support definition of SLA policies and automatic escalation of overdue tickets. | Pending | Pending | Pending | Pending |
| FR0010 | Notification System | Trigger email/SMS alerts for ticket status changes, escalations, and system errors. | Pending | Pending | Pending | Pending |
| FR0011 | Customer Feedback Capture | Allow users to rate support experience and leave comments. | Pending | Pending | Pending | Pending |

Document 5

1. Document Revision:

|  |  |  |
| --- | --- | --- |
| Date | Version | Document Changes |
| 4/14/2025 | 1.0 | Initial Draft |

1. Approvals:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Name | Title | Signature | Date |
| Project Sponsor | Bala S | CEO |  | 4/15/2025 |
| Business Owner | Prakash | Vice President |  | 4/15/2025 |
| Project Manager | Naresh | Project Manager |  | 4/15/2025 |
| System Architect | Manoj | Staff System Administrator |  | 4/15/2025 |
| Development Lead | Gunjan | Principal Software Developer |  | 4/15/2025 |
| User Experience Lead | Paresh | Senior Designer |  | 4/15/2025 |
| Quality Lead | Sanyam G | Lead QA |  | 4/15/2025 |
| Content Lead | Vihana | Lead Content Manager |  | 4/15/2025 |

1. ***RACI Chart:***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Position | \* | R | A | S | C | I |
| Bala S | CEO | Checkmark with solid fill |  | Checkmark with solid fill |  |  |  |
| Prakash | VP |  |  | Checkmark with solid fill |  |  |  |
| Naresh | Project Manager |  | Checkmark with solid fill |  |  |  |  |
| Manoj | System Administrator |  |  |  | Checkmark with solid fill |  |  |
| Gunjan | Principal software Dev |  | Checkmark with solid fill |  |  |  |  |
| Paresh | Senior UI/UX designer |  | Checkmark with solid fill |  |  |  |  |
| Sanyam G | Lead QA |  |  |  |  | Checkmark with solid fill |  |
| Viahana | Lead Content Manager |  |  |  |  | Checkmark with solid fill |  |
| Vipul | Lead BA |  | Checkmark with solid fill |  |  |  |  |
| Jay | BA |  | Checkmark with solid fill |  |  |  |  |
| Vikash | Front-End Developer |  | Checkmark with solid fill |  |  |  |  |
| Abhsihek | Backend Developer |  | Checkmark with solid fill |  |  |  |  |
| Arindam | AI Engineer |  | Checkmark with solid fill |  |  |  |  |
| Deepak | Devops Engineer |  | Checkmark with solid fill |  |  |  |  |
| Akash | QA |  |  |  |  | Checkmark with solid fill |  |
| Sonu | BI Developer |  |  |  |  | Checkmark with solid fill |  |
| Pradeep | Product Owner |  |  | Checkmark with solid fill |  |  |  |
| Ronak | Support Lead/ Delivery Lead |  |  | Checkmark with solid fill |  |  |  |
| Deepshikha | Sales |  |  |  |  |  | Checkmark with solid fill |
| Sushmita | Legal Associate |  |  |  |  | Checkmark with solid fill | Checkmark with solid fill |

1. Introduction:
   1. **Business Goal:**

To enhance the efficiency, responsiveness, and scalability of customer support services by developing a unified, AI-powered platform that consolidates communication channels, streamlines workflows, and delivers actionable insights.

* 1. **Business Objective:**
     + - To reduce customer support response times by 40%.
       - To centralize all support communications into one integrated platform.
       - To enable data-driven decisions through built-in analytics and reporting tools.
       - To increase customer satisfaction (CSAT) by at least 20% within the first year of deployment.
  2. **Business Rules:**
     + - All customer data must comply with GDPR and local data privacy laws.
       - Every support interaction must be logged and traceable.
       - AI decisions (e.g., routing, summaries) must be auditable.
       - Ticket resolutions must adhere to SLA timelines as defined by client contracts.
       - Any escalation must follow the predefined chain of command.
       - All integrations must be approved by the IT security team before implementation.
  3. **Background:**

This project originated from recurring complaints regarding fragmented support experiences and inefficiencies in existing systems. Current tools lack real-time analytics, AI support, and seamless cross-channel capabilities, leading to delays, lost tickets, and poor customer satisfaction. Competitive benchmarking highlighted the opportunity to create a proprietary, fully integrated support tool akin to Zendesk but tailored to the organization's unique workflows and integrations. This platform is expected to increase operational efficiency, improve customer satisfaction, and reduce operational costs.

* 1. **Project Objective:**

To develop a secure, cloud-based, unified customer support platform that:

* + - * Integrates chat, email, and voice call capabilities.
      * Supports AI-driven ticket routing, summarization, and customer interaction.
      * Provides customizable dashboards and analytics reports.
      * Offers out-of-the-box integrations with third-party tools like Jira, Azure DevOps, and Slack.
      * Aligns with business goals by improving support KPIs and enabling better decision-making.
  1. **Project Scope:**

The scope of this project includes the design, development, and deployment of a Minimum Viable Product (MVP) for the Unified Support Platform that integrates communication, ticketing, AI, and analytics features.

* + 1. **In Scope Functionality**:
       - Real-time live chat support
       - Email ticket creation and tracking
       - VoIP-based calling functionality
       - AI-powered chatbot and ticket summarization
       - Smart ticket routing and tagging
       - Role-based access control (RBAC)
       - SLA and escalation rules management
       - Integrations with Jira, Azure, Slack, and CRM systems
       - Real-time analytics dashboards and reporting
       - Ticket lifecycle tracking and management
       - Notification and alert system (email/SMS)
    2. **Out Scope Functionality**
* Multilingual NLP chatbot (only English in MVP)
* Mobile application development (MVP is web-only)
* Integration with payment systems or billing modules
* Internal HR or payroll system integrations

1. Assumptions:

These are the conditions believed to be true for successful project planning and execution:

* User Readiness: End users (agents, admins) will be trained before the system goes live.
* Tool Access: All necessary third-party platforms (Jira, Azure, Slack) will grant API access for integration.
* Data Availability: Historical ticket data will be made available for AI training and analytics setup.
* Compliance Support: Legal and compliance teams will guide the handling of sensitive data in accordance with GDPR/industry standards.
* Vendor Support: External services like Twilio (for calls) and Open AI (for NLP) will remain stable and accessible during project duration.
* Stakeholder Commitment: Key stakeholders will be available for timely reviews and approvals at each project milestone.
* Browser Compatibility: Users will access the platform using modern browsers like Chrome, Firefox, or Edge.
* UI/UX Standards: Design assumptions are based on existing usability standards and user feedback patterns.

1. ***Constraints:***

These are limitations that must be considered during the project’s planning and execution:

* Budget Limitation: The project must be delivered within the approved budget envelope.
* Fixed Timeline: The MVP must be launched within a 12–15-month development window as per Waterfall methodology.
* Scope Freeze: Once the project moves past the requirement phase, no new features can be added until the next phase or release cycle.
* Security Regulations: The platform must comply with GDPR, ISO 27001, and any applicable local data protection laws.
* Platform Scope: Only the web version is included in the MVP. Mobile or desktop apps are out of scope.
* Third-Party Dependency: Integration success depends on the reliability and documentation of third-party APIs (e.g., Jira, Azure).
* Language Limitation: MVP will only support English language content and responses.
* Hosting Restrictions: Only pre-approved cloud providers (e.g., Azure, AWS) can be used for deployment.
* Licensing Boundaries: Use of third-party AI tools is subject to licensing terms and monthly usage limits.

1. ***Risks:***
   1. **Technological Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact Cost | Strategy | Details |
| Unstable third-party APIs (e.g., Jira, Twilio) | Medium | High | Mitigate | Add API fallback mechanisms and sandbox testing before full integration |
| AI features (e.g., NLP, routing) underperform | High | High | Mitigate | Begin AI model prototyping early and iterate with real ticket data |
| Platform scalability issues under heavy load | Medium | Medium | Mitigate | Conduct performance testing and plan horizontal scaling with load balancers |

* 1. **Skills Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact Cost | Strategy | Details |
| Lack of skilled AI/NLP engineers | Medium | High | Mitigate | Hire early or outsource to experienced consultants |
| Inadequate DevOps or cloud infrastructure expertise | Medium | Medium | Transfer | Contract deployment and CI/CD tasks to DevOps vendors if in-house resources are limited |
| Limited frontend/backend bandwidth due to parallel projects | Medium | Medium | Mitigate | Allocate dedicated developers or adjust timeline |

* 1. **Political Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact Cost | Strategy | Details |
| Change in leadership or internal priorities | Medium | High | Accept | Keep project documentation and KPIs transparent to gain long-term buy-in |
| Departmental conflicts over platform ownership | Medium | Medium | Mitigate | Involve cross-functional teams early and clearly define ownership zones |

* 1. **Business Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact Cost | Strategy | Details |
| Project cancellation before full deployment | Low | High | Accept | Have a clear MVP that delivers tangible value early |
| Budget overruns due to AI/analytics feature costs | Medium | High | Mitigate | Prioritize features and include cost buffer in budget |

* 1. **Requirements Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact Cost | Strategy | Details |
| Misunderstood AI automation expectations | Medium | Medium | Mitigate | Regular demos and walkthroughs with stakeholders |
| Underdefined integration scenarios | High | Medium | Mitigate | Schedule early workshops with integration stakeholders |
| Scope creep due to unclear ticketing rules | Medium | Medium | Avoid | Freeze scope after BRD approval and handle additions via CR (Change Request) process |

* 1. **Other Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact Cost | Strategy | Details |
| Security breach due to poor access control | Low | High | Avoid | Conduct security audits and implement RBAC from day one |
| Resistance to change from customer support teams | Medium | Medium | Mitigate | Invest in strong training, communication, and pilot testing |

1. ***Business Process Overview:***
   1. **AS-IS:**

The current support system relies on multiple disjointed tools:

* Email clients for handling support tickets
* Manual phone support via legacy PBX or desk phones
* Chat handled via a separate widget with no integration
* Ticketing handled manually in spreadsheets or basic CRMs.

Challenges:

* No unified dashboard to track customer issues across channels
* Lack of SLA enforcement or escalation triggers
* No real-time analytics or performance reporting
* No AI assistance or auto-routing capabilities
* Duplication of effort and communication gaps between channels
* No direct integration with Jira, Azure, or CRM tools
* [Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]

Ticket Generated In CRM

Customer Email

Status Tracking in CRM

Assign to particular team manually

Agent Handles ticket

Customer receives update through email

Ticket resolved and status updated manually.

**Process Flow Diagram:**

* 1. **To-BE:**

The proposed Unified Customer Support Management Platform introduces:

* A centralized platform for handling chat, email, and VoIP-based calls
* A robust ticketing system with lifecycle management
* AI automation for chatbots, smart routing, and ticket tagging
* Integrated analytics dashboards
* Built-in SLAs and escalation logic
* Seamless integration with Jira, Azure, Slack, CRM systems

Benefits Over Legacy System:

* Eliminates tool fragmentation
* Automates repetitive tasks (routing, summarizing, tagging)
* Real-time visibility into support performance
* Reduces human error
* Improves customer response and resolution time
* Enables proactive support via analytics

1. ***Business Requirements:***

**Communication & Ticketing**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority (1–10) | Linked Use Case |
| COMM-01 | Platform must support real-time live chat between agents and customers | 10 | UC-Chat-01 |
| COMM-02 | System should convert customer emails into support tickets | 9 | UC-Email-01 |
| COMM-03 | Enable VoIP calling and call logging within platform | 9 | UC-Call-01 |
| COMM-04 | Support ticket lifecycle: Create → Assign → Resolve → Close | 10 | UC-Ticket-01 |

**AI Automation**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority (1–10) | Linked Use Case |
| AI-01 | AI should suggest ticket categorization and tagging | 8 | UC-AI-01 |
| AI-02 | AI chatbot should resolve FAQs and escalate complex queries | 9 | UC-Bot-01 |
| AI-03 | System should summarize email threads and customer conversations | 8 | UC-AI-02 |
| AI-04 | Sentiment detection for prioritization | 7 | UC-AI-03 |

**Integrations**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority (1–10) | Linked Use Case |
| INTEG-01 | Integrate with Jira for ticket escalation and syncing | 9 | UC-Integrate-01 |
| INTEG-02 | Sync with Azure DevOps for internal bug tracking | 8 | UC-Integrate-02 |
| INTEG-03 | Slack notification integration for support alerts | 7 | UC-Integrate-03 |

**Reporting & Analytics**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority (1–10) | Linked Use Case |
| REPORT-01 | Dashboard showing open, resolved, and escalated tickets by category | 10 | UC-Report-01 |
| REPORT-02 | SLA performance and agent activity reporting | 9 | UC-Report-02 |
| REPORT-03 | Downloadable reports in CSV and PDF formats | 7 | UC-Report-03 |

**Security & User Management**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Priority (1–10) | Linked Use Case |
| SEC-01 | Support Role-Based Access Control (Admin, Agent, Viewer) | 10 | UC-Security-01 |
| SEC-02 | System must log all user actions and changes | 8 | UC-Security-02 |
| SEC-03 | Compliance with GDPR and data retention policies | 10 | UC-Security-03 |