TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY
MADAN BHANDARI MEMORIAL COLLEGE

PROJECT PROPOSAL

Customer Support Chat-bot

Submitted by:

Firoj Paudel (79011003)

SUBMITTED TO:

LAXMI PRASAD YADAV

***Lecturer* — System Analysis And Design**

March 15, 2025

Abstract

This project proposes the development of a fine-tuned customer support chatbot leveraging the BART model to enhance customer service efficiency through advanced NLP techniques. The chatbot, hosted on Streamlit with SQLite as the database, aims to automate 60% of customer inquiries, reduce response times, and improve satisfaction by providing accurate, context-aware responses. Key features include user authentication, intent-based query processing, and speech-to-text input, with the system achieving a response accuracy above 85%. Building on prior research into LLM generalization challenges, this project integrates the GEM architecture to ensure robust performance on niche datasets, offering a scalable solution for business adoption.

Contents

List of Figures

List of Symbols and Acronyms

API Application Programming Interface. 5, 10

BART Bidirectional Auto-Regressive Transformer. i, 1, 3–9, 11, 14

GDPR General Data Protection Regulation. 10

GEM Generalized Edge Model. i, 1, 3, 4, 6, 7, 14

LLM Large Language Model. i, 1, 6, 14

NLP Natural Language Processing. i, 1, 5–7, 9, 14

SDLC Software Development Life Cycle. 7, 13

1. Introduction

This project focuses on developing a fine-tuned customer support chatbot to address inefficiencies in traditional customer service systems. By leveraging the BART model and advanced NLP techniques, the chatbot aims to automate a significant portion of customer inquiries, delivering fast, accurate, and context-aware responses. Hosted on Streamlit with SQLite as the database, the system includes features like user authentication, intent-based query processing, and speech-to-text input, ensuring a seamless user experience. This work builds on prior research conducted by our team, where we explored the generalization challenges of LLM on niche topics as part of a research paper. During that study, we developed the GEM architecture, which we have integrated into this project to enhance the chatbot's performance on a small, specialized dataset, mitigating issues like hallucination and ensuring robust generalization.

2. Problem Statement

Traditional customer support systems heavily rely on human agents, resulting in several inefficiencies: delayed response times, inconsistent answers, and limited scalability to handle high query volumes. While basic rule-based chatbots exist, they often struggle with complex or nuanced inquiries, leading to frequent escalations to human agents, which further slows down the process and increases operational costs. Additionally, these systems lack the ability to understand context or adapt to user intent, causing frustration for customers seeking quick and accurate resolutions. This project aims to address these challenges by developing an intelligent, fine-tuned chatbot capable of handling a wide range of inquiries autonomously, reducing the burden on human agents and improving overall customer satisfaction.

3. Objectives

The primary objective of this project is to design, implement, and deploy a customer support chatbot fine-tuned on the BART model to enhance customer service efficiency. Specifically, the project seeks to achieve the following goals:

- Automate at least 60% of customer inquiries, significantly reducing the workload on human agents and enabling businesses to scale their support operations without proportional cost increases.

- Minimize response times to under 30 seconds per query, ensuring customers receive prompt assistance and improving their overall experience.

- Enhance customer satisfaction by delivering accurate, context-aware responses with a target accuracy rate above 85%, leveraging intent-based processing and prior conversation history stored in an SQLite database.

- Integrate advanced features such as user authentication, speech-to-text input, and intent specification to provide a seamless and user-friendly interface via a Streamlit-hosted application.

- Incorporate the GEM architecture, developed from prior research, to improve the chatbot's generalization on niche datasets, ensuring robust performance despite limited training data.

4. Scope

The scope of this project encompasses the development, testing, and deployment of a customer support chatbot using the BART model, fine-tuned on an open-source dataset from BiText [1]. The chatbot will be integrated into an existing system via a Streamlit application, utilizing SQLite as the database for storing user data and conversation history. Key deliverables include achieving a response accuracy above 85%, automating 60% of customer inquiries, and supporting features like user authentication, intent-based query processing, and speech-to-text input. The project also incorporates the GEM architecture [2] to enhance generalization on niche topics. However, the initial phase will not support multiple languages or involve building a new information system from scratch, focusing instead on English-language support and integration with existing infrastructure. Future work may explore multi-language support and broader system enhancements.

5. Methodologies

5.1 Planning

5.1.1 Requirement Identification

The planning phase initiates with a comprehensive identification of requirements essential for developing a fine-tuned customer support chatbot. This project aims to enhance the efficiency and effectiveness of customer service operations within an information system framework, addressing the growing demand for automated, scalable, and intelligent support solutions in modern businesses. Key requirements include:

- **Natural Language Understanding**: The chatbot must process and comprehend a wide range of customer queries, from simple FAQs to complex, context-dependent requests, leveraging NLP capabilities.
- **Accuracy and Relevance**: A target response accuracy of over 85% is required to ensure reliable interactions, reducing the need for human intervention.
- **System Integration**: Seamless integration with existing customer support systems via APIs, ensuring compatibility with current workflows.
- **Scalability**: The solution must handle varying query volumes, initially on local infrastructure, with potential for future cloud-based scaling.

The decision to pursue this project stems from the limitations of current systems, which rely heavily on human agents, leading to delays and inconsistent responses. By fine-tuning a pre-trained BART model [3], the chatbot will deliver a robust, context-aware experience tailored to real-world customer interactions.

5.1.2 Studying Existing Systems

A detailed study of existing customer support systems provides critical insights into their strengths and shortcomings, shaping the development strategy for this project. Current systems predominantly depend on human agents, supplemented by basic rule-based chatbots. These systems exhibit the following characteristics:

- **Human Dependency**: Most interactions are managed manually, resulting in longer response times (averaging 5–10 minutes per query) and limited scalability during peak demand periods, such as sales seasons or product launches.

- **Rule-Based Limitations**: Existing chatbots operate on predefined scripts, lacking the ability to interpret nuanced or ambiguous queries. This leads to frequent escalations—estimated at 70% of interactions—burdening human agents further.

- **Infrastructure Constraints**: Many systems use legacy software with minimal automation, lacking integration with modern NLP tools or datasets like BiText [1].

This analysis highlights the need for an advanced solution. The proposed chatbot, built on the BART-base model, will overcome these limitations by autonomously handling a significant portion of inquiries, reducing response times to under 30 seconds, and adapting to diverse customer needs through fine-tuning on real-world data.

5.1.3 Requirement Collection

Requirement collection involves gathering detailed inputs to ensure the chatbot meets its objectives within the one-month timeline. This process combines data-driven analysis with insights from market surveys:

- **Dataset Analysis**: The BiText dataset [1], an open-source repository of customer support interactions, serves as the primary training resource. I analyzed this dataset, which includes thousands of query-response pairs covering topics like troubleshooting, billing, and product inquiries. Initial preprocessing (Week 1) will clean and structure this data for BART fine-tuning, ensuring the model can handle diverse customer queries effectively.

- **Market Surveys**: Prior to this project, I conducted surveys with business owners from hotels and consulting companies to understand their customer support challenges. The feedback highlighted a pressing need for automation in handling tedious, repetitive customer interactions, such as answering FAQs or resolving common billing issues. These business runners expressed that a well-generalizing chatbot could significantly reduce operational costs and improve response times, especially during high-demand periods. This insight aligns with the broader market need for scalable, intelligent support solutions that can adapt to niche domains without requiring extensive training data.

- **Research Context**: This chatbot project emerged as a practical application of my research paper on LLM generalization challenges, where my team developed the GEM architecture[2]. Originally, the goal was to explore how LLM could generalize on niche topics with small datasets, but the potential for real-world impact led to the development of this chatbot. By integrating the GEM architecture, the chatbot achieves robust generalization, mitigating issues like hallucination and ensuring accurate responses despite the limited dataset size.

- **Technical Specifications**: The chatbot will use local computing resources (e.g., a mid-range GPU or high-performance CPU) and open-source NLP libraries like Hugging Face Transformers. The SDLC phases—planning, analysis, design, implementation, and deployment—are scheduled as follows:
  - **Week 1**: Dataset preparation and requirement finalization.
  - **Weeks 2–3**: Model fine-tuning and iterative testing to achieve $>85\%$ accuracy.
  - **Week 4**: Deployment within the existing system.
- **Performance Metrics**: Beyond accuracy, metrics like response time (< 30 seconds), query resolution rate (60% automation), and user satisfaction (via post-deployment feedback) will guide development.
- **Future Potential**: The surveys also revealed interest in advanced features like voice-based chatbots, where the system could accept voice inputs and respond in voice. While this is beyond the current scope, it highlights a future direction for the project, building on the chatbot's ability to generalize effectively across diverse interaction modes.

This comprehensive collection ensures the project aligns with both technical feasibility and market needs, leveraging the BART model's bidirectional capabilities and the GEM architecture to process queries effectively. The planning phase, currently underway, sets the stage for subsequent analysis and design by defining clear objectives and resources.

5.2 Feasibility Analysis

This section evaluates the feasibility of developing a fine-tuned customer support chatbot across multiple dimensions: economic, technical, operational, and legal. Each aspect is analyzed to ensure the project's viability within the one-month timeline and initial zero-cost constraints, with considerations for future scaling.

5.2.1 Economic Feasibility

Economic feasibility assesses whether the project's benefits justify its costs, both in the initial phase and as it scales. The starter project leverages personal effort and free tools, while future revenue models ensure sustainability.

Cost-Benefit Analysis In the initial phase, development costs are effectively zero, as the project is undertaken solo using existing skills and resources:

- **Development Effort**: Performed entirely by the developer (myself), requiring no monetary investment—just time and expertise over the one-month timeline.
- **Hosting**: Hosted on Streamlit's free tier, which supports rapid deployment and testing of the BART-based chatbot without upfront costs.
- **Dataset**: The open-source BiText dataset [1] is used for fine-tuning, incurring no expense.

Tangible benefits in this phase are limited to proof-of-concept validation, but intangible benefits include skill enhancement and a functional prototype for demonstration. No immediate financial return is expected initially, as this is a self-funded pilot.

For scaling, costs emerge as the project grows into a business-oriented solution:

- **Dataset**: As the open-source BiText dataset becomes insufficient for larger-scale customization, businesses will provide their own datasets, eliminating purchase costs. This shifts the burden to clients while ensuring relevance to their needs.
- **Production Costs**: Scaling introduces expenses like cloud hosting (e.g., 20,000 NPR annually for a basic cloud server), security measures (e.g., 10,000 NPR for firewalls, encryption), and potential part-time developer support (e.g., 20,000 NPR annually). Total estimated scaling cost is capped at 50,000 NPR for a medium-scale rollout.

Revenue generation hinges on a premium subscription model targeting business customers:

- **Customer Base**: Businesses needing customer support automation are the target, as the chatbot reduces their workforce requirements (e.g., cutting labor costs by 60%, or 10,000 NPR annually per business for a small team).

- **Premium Models**: Subscription tiers will offer customized fine-tuning on client datasets, with pricing such as:
  - Basic Tier: 5,000 NPR/year for standard features.
  - Premium Tier: 15,000 NPR/year for advanced customization and priority support.
- **Profit Mechanism**: By reducing client expenses (e.g., 10,000 NPR saved vs. 5,000–15,000 NPR subscription), businesses see a net gain, incentivizing adoption. With 10 clients at the basic tier, revenue reaches 50,000 NPR/year, covering scaling costs and yielding profit with more subscribers.

Data security is prioritized to build trust—client datasets are encrypted, and investments in firewalls and compliance (part of the 10,000 NPR security cost) ensure safety. The initial zero-cost phase breaks even instantly, while scaling achieves profitability within 6–12 months with 10–20 subscribers.

5.2.2 Technical Feasibility

The project's technical feasibility is grounded in available tools and expertise, ensuring success within the constrained timeline and local infrastructure. Key factors include:

- **Pre-trained Models**: The BART model [3] is accessed via Hugging Face, with its base version fine-tuned for context-aware responses.
- **NLP Frameworks**: Hugging Face Transformers supports fine-tuning on local hardware (e.g., 16GB RAM, 4GB VRAM), with Streamlit enabling free hosting and a user-friendly interface.
- **Local Infrastructure**: Existing hardware handles preprocessing (Week 1) and fine-tuning (Weeks 2–3), with deployment on Streamlit (Week 4). Risks like overfitting are mitigated via hyperparameter tuning.
- **Scaling Considerations**: Future cloud hosting (e.g., AWS, Google Cloud) will support higher query volumes, while client-provided datasets ensure scalability without additional dataset costs.

The solo developer's expertise in NLP and software development ensures technical viability, with Streamlit simplifying deployment.

5.2.3 Operational Feasibility

Operational feasibility evaluates the chatbot's fit within customer support workflows and its acceptance by future business clients:

- **Workflow Integration**: The chatbot integrates via APIs into existing systems, processing queries and escalating complex cases, aligning with business operations.
- **Staff Impact**: Businesses require minimal retraining (1–2 hours) to monitor escalations and feedback, leveraging the chatbot's 60% automation to reduce workload.
- **User Acceptance**: Clients will adopt it due to cost savings (e.g., 10,000 NPR/year) and improved service (response time ¡30 seconds), with customization enhancing relevance.

The initial Streamlit-hosted prototype proves operational fit, scalable to business needs with client-specific fine-tuning.

5.2.4 Legal Feasibility

Legal feasibility ensures compliance and risk mitigation:

- **Data Privacy**: The BiText dataset [1] is anonymized initially, while client datasets are encrypted and protected (part of scaling security costs), complying with GDPR (if applicable) or local laws.
- **Industry Standards**: The chatbot identifies itself as automated and uses secure APIs, meeting guidelines for business tools.
- **Legal Barriers**: No issues arise from open-source tools or client-provided data, with security investments ensuring compliance as it scales.

The project remains legally sound, with zero initial cost and future safeguards for client trust.

6. Diagrams

6.1 Context Diagram

The context diagram illustrates the high-level interaction between the user, the Streamlit-hosted chatbot application, and the SQLite database. It serves as the equivalent of a Level 0 DFD, showing the system's external interactions.

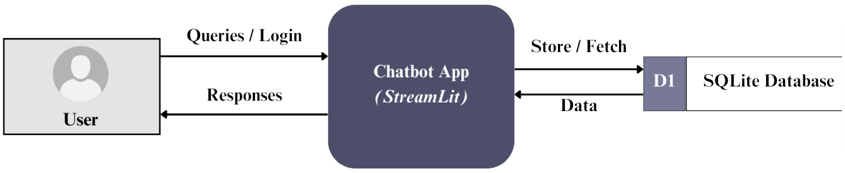


Figure 6.1: Context Diagram for Customer Support Chatbot

6.2 Data Flow Diagrams

6.2.1 Level 1 DFD

The Level 1 DFD details the chatbot application's core workflow after successful login. Users specify an intent, provide input (text or speech-to-text via a voice recognition feature), and set the desired token length for the response. The app processes this through the fine-tuned BART model, fetching prior responses from the SQLite database to enhance context, and displays the generated output to the user.

6.2.2 Level 2 DFD: Login System

The Level 2 DFD details the login system, with the main process labeled as 1.1 (Login Process) and its subprocesses as 1.2.1 (Create Account) and 1.2.2 (Forget Password). It shows interactions between the user and the SQLite database for authentication, account creation, and password reset.

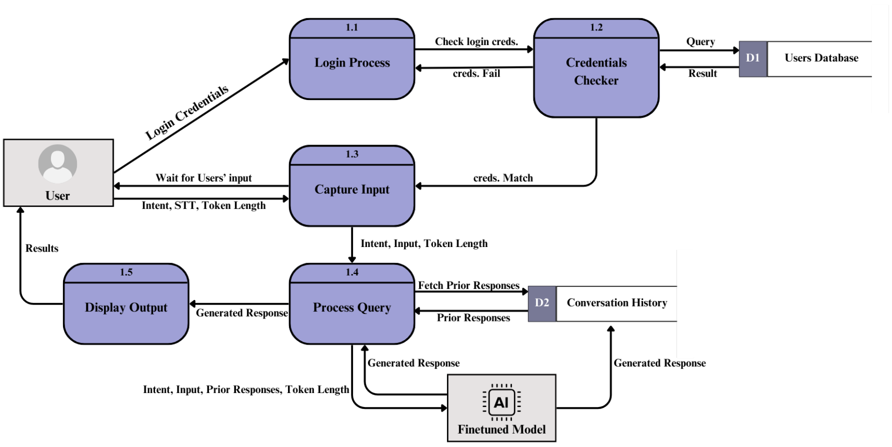


Figure 6.2: Level 1 Data Flow Diagram

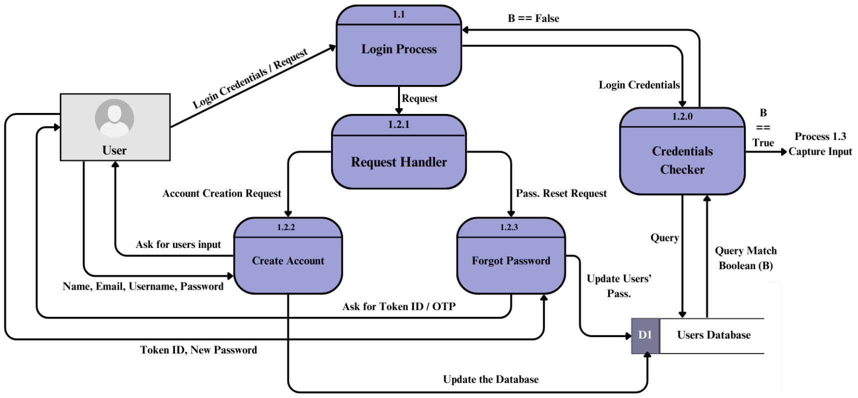
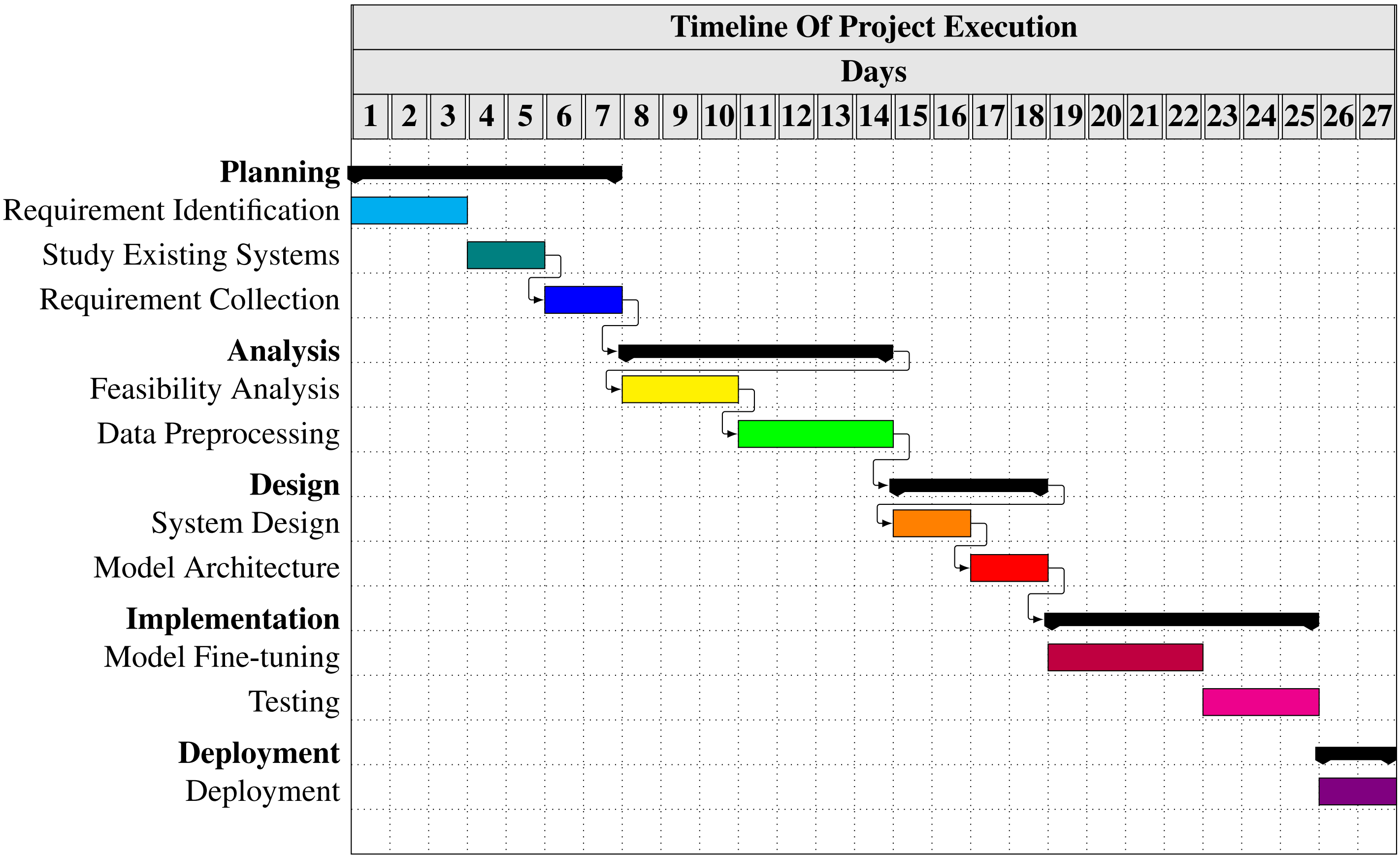


Figure 6.3: Level 2 Data Flow Diagram: Login System

6.3 Gantt Chart

The Gantt chart below outlines the project timeline over a one-month period (27 days), aligned with the SDLC phases: planning, analysis, design, implementation, and deployment.



6.4 Expected Output

The expected output of this project includes:

- A fully functional customer support chatbot fine-tuned on the BART model, achieving a response accuracy above 85%.
- Automation of 60% of customer inquiries, reducing response times to under 30 seconds.
- A Streamlit-hosted application with user authentication, intent-based query processing, and speech-to-text input capabilities.
- A scalable prototype ready for business adoption, with a premium subscription model generating revenue (e.g., 50,000 NPR/year with 10 clients).
- Comprehensive documentation, including user guides and technical reports, to support future scaling and maintenance.

6.5 Future Work

This project successfully integrated the GEM architecture, developed as part of our prior research on LLM generalization challenges. Despite the small dataset, the chatbot demonstrates robust generalization, avoiding hallucination and delivering accurate responses on niche topics. The research paper detailing this work, including the GEM architecture, has been prepared and is published on arXiv [2]. Future enhancements may include expanding the chatbot to support multiple languages, integrating more advanced NLP models, and scaling the system for broader enterprise adoption.

Bibliography

[1] Bitext. Bitext customer support llm chatbot training dataset. https://huggingface.co/datasets/bitext/Bitext-customer-support-llm-chatbot-training-dataset, 2024.

[2] Basab Jha and Firoj Paudel. Fragile mastery: Are domain-specific trade-offs undermining on-device language models? *arXiv preprint arXiv:2503.22698*, 2025.

[3] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.