Informatics Institute of Technology

In Collaboration With

University of Westminster, UK



*University of Westminster, Coat of Arms*

GATOT

Design

Mr. Nazhim Kalam

w1761265 / 2019281

Supervised by

Mr. Torin Wirasingha

September 2022

This Project Proposal is submitted in partial fulfilment of the requirements for

the BSc (Hons) Computer Science degree at

the University of Westminster.

Table of Contents

[List of Figures ii](#_Toc120522959)

[List of Tables ii](#_Toc120522960)

[1. CHAPTER OVERVIEW 1](#_Toc120522961)

[2. DESIGN GOAL 1](#_Toc120522962)

[3. HIGH-LEVEL DESIGN 1](#_Toc120522963)

[4. SYSTEM DESIGN 1](#_Toc120522964)

[4.1 Choice of the Design Paradigm. 1](#_Toc120522965)

[4.2 Data Flow Diagram 1](#_Toc120522966)

[4.3 Algorithm Design 1](#_Toc120522967)

[4.4 UI Design 1](#_Toc120522968)

[5. CHAPTER SUMMARY 2](#_Toc120522969)

[REFERENCES I](#_Toc120522970)

# List of Figures

[Figure 12.1 - Prototype Feature Diagram (Self-composed) 13](#_Toc117550682)

[Figure 13.1 - Gantt Chart 16](#_Toc117550683)

[Figure 13.2 - Model development flow (Self-composed) 21](#_Toc117550684)

# List of Tables

[Table 5.1 - Related work in abstractive text summarization 3](#_Toc117584436)

[Table 11.1 - Research Objectives 9](#_Toc117584437)

[Table 13.1 - Research Methodology 13](#_Toc117584438)

[Table 13.2 - Deliverables and Dates 17](#_Toc117584439)

[Table 13.3 - Risk Mitigation Plan 19](#_Toc117584440)

**Acronyms**

|  |  |
| --- | --- |
| AI | Artificial Intelligence. |
| DL | Deep Learning |
| GUI | Graphical User Interface |
| ML | Machine Learning |
| NLP | Natural Language Processing |
| ROUGE | Recall-Oriented Understudy for Gisting Evaluation. |
| BLEU | BiLingual Evaluation Understudy. |
| T5 | Text to Transfer Transformer. |
| BART | Bidirectional Auto-Regressive Transformers. |
| BERT | Bidirectional Encoder Representations from Transformers. |
| PEGASUS | Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence |
| ILP | Inductive logic programming. |
| LSTM | Long Short-Term Memory. |
| RNN | Recurrent Neural Network. |
| CNN  SEQ2SEQ | Convolutional Neural Network.  Sequence to Sequence |
| RoBERTa | Robustly Optimized BERT Pre-training Approach |
| GPT-3  REST  GPU | Third Generation Generative Pre-Trained Transformer  Representational State Transfer  Graphical Processing Unit |

# Chapter Overview

The design choices taken to create a suitable architecture for implementation, depending on the requirements received, are discussed in this chapter. To explain how the design goals are intended to be accomplished while outlining the justification for selected design decisions, high-level design, low-level design, design diagrams, and UI wireframes have been utilized.

# Design Goals

Table 6.1 – Design Goals of the proposed system

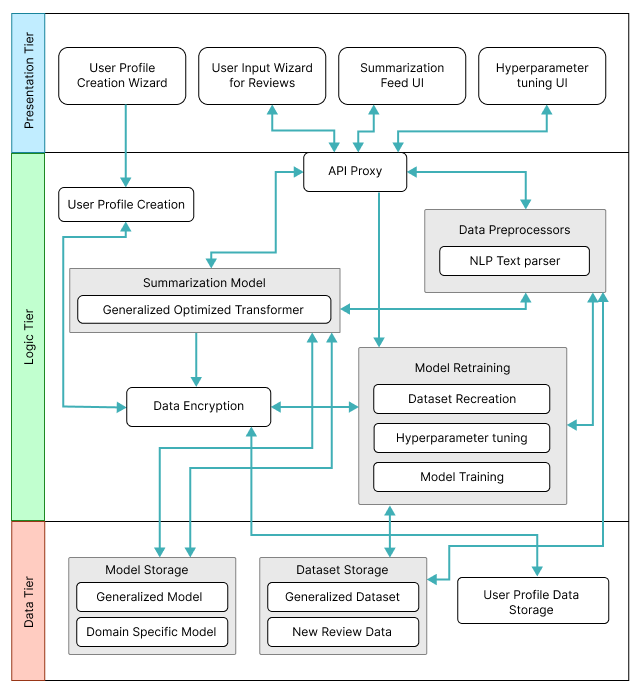
|  |  |
| --- | --- |
| **Design Goal** | **Description** |
| Performance | To find the new set of hyperparameters with the new data, model retraining requires a significant amount of time. As a result, the newly created dataset (with unseen data) should be accurately made, and it is best if it takes the least amount of time to query the data from various businesses within the same domain to create the dataset. Moreover, other core functionalities should be designed effectively to increase overall performance. |
| Correctness | The correctness & quality of the output should be of the highest possible level utilizing the optimized transformer architecture. Since several approaches are considered in order to get the optimized solution the expected output should of the best possible form. |
| Usability | The system's usability must be straightforward for users of all levels of knowledge because its primary function is to summarize review text for any domain, including movies and general users. |
| Scalability | In a production environment, the system may need to accommodate a large number of concurrent user requests. This should be manageable by the backend. The system should be easily expandable to accommodate new data. |
| Adaptability | Adopting new features or components need to be a simple procedure. The system shouldn't be broken if a component is added or removed, and it shouldn't be affected overall. |

# High-level Design

## **3.1 Tiered Architecture**

The image below depicts the architecture of the system. Three tiers of architecture separate the data, logic, and presentation levels. The system's generalization and domain specific adaptive hyperparameter tuning and data preprocessing represent the research contribution.

Figure 3.1: Three-Tiered Architecture (*self-composed*)



A microservice architecture is followed during the implementation of the project using several backend services, the design representation given above is just to give an idea of how all the components are expected to work together. To enable system scalability while assuring that failure areas may be quickly identified and handled individually, a microservices design is used.

The final prototype is expected to be run in a single machine even though the designed system is proposed to be run in a distributed server system. A thorough explanation of each module seen on the architectural diagram above is provided below.

**Data Tier**

1. Model Storage - The text summarization models which will be used for both generalized text summarization and domain specific text summarization will be stored here.
2. Generalized Model – The model which will be used by general users to generated review summarized, this model will be hyperparameter tuned for genialized purpose.
3. Domain Specific Model – The model will be used by domain specific users for review summarization, this model will be replaced whenever the model retraining is triggered from the domain user.
4. Dataset Storage – The data which is required for model training will be available.
5. Generalized Dataset – The data which is used for creating the generalized model will be stored for retraining when it comes to domain specific model retraining.
6. New Review Data – The data stored here are from the domain users when they use the application, the data will be storage and used for retraining along with the generalized dataset.
7. User Profile Data Storage – The metadata data related to the domain specific user when creating the user profile will be stored, for updating and profile deletion.

**Logic Tier**

1. User Profile Creation – Allowing to create unique user profiles for each domain user, main purpose comes when working with model retraining to figure out the data to be used.
2. API Proxy – Interface which allows the frontend to communicate with the backend services via HTTP calls/ request.
3. Data Preprocessors –
4. NLP Text parser –
5. Summarization Model –
6. Generalized Optimized Transformer –
7. Data Encryption –
8. Model Retraining –
9. Dataset Recreation –
10. Hyperparameter tuning –
11. Model Training –

**Presentation Tier (Client Tier)**

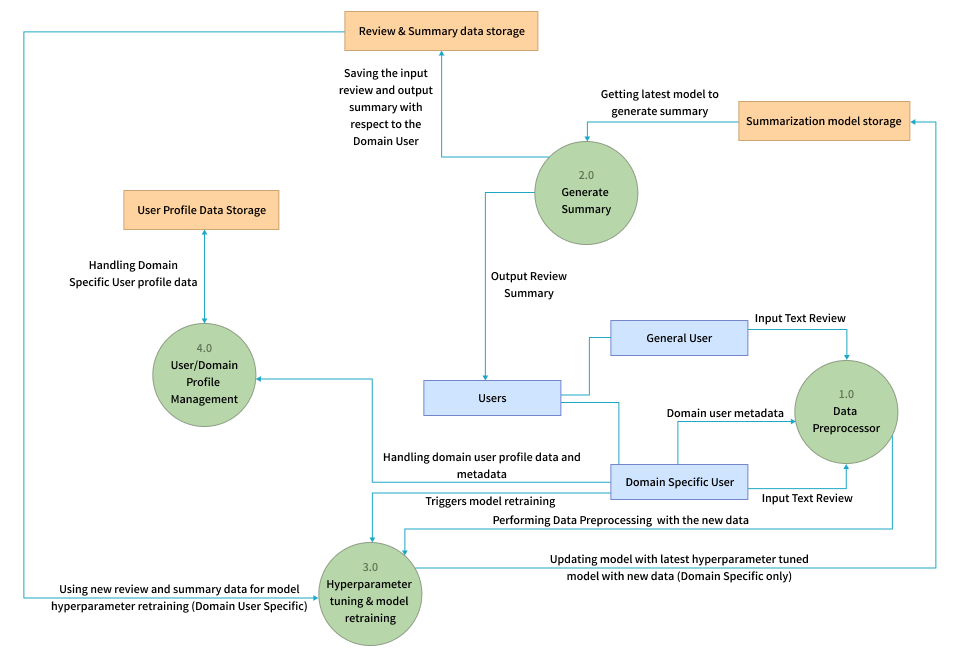
1. User Profile Creation Wizard –
2. User Input Wizard for Reviews –
3. Summarization Feed UI –
4. Hyperparameter tuning UI –

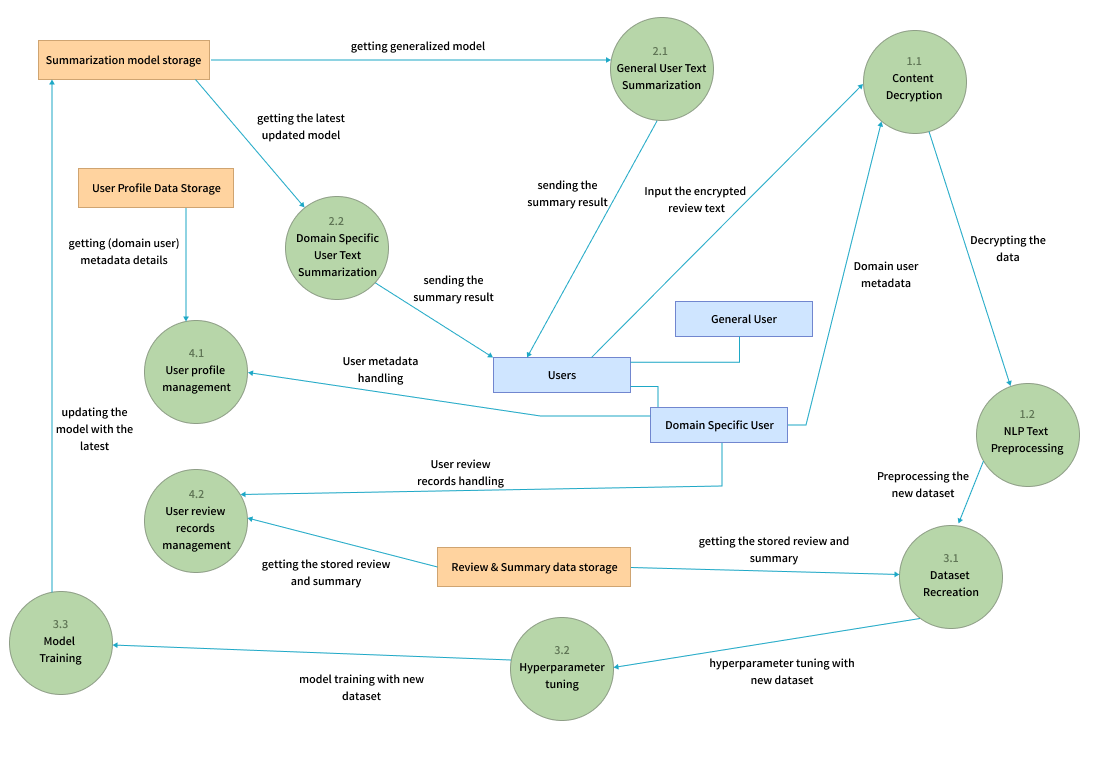
# System Design

## **Choice of the Design Paradigm.**

## **Data Flow Diagram**

In order to show the relationships between components and provide a clearer understanding of how data flows across the whole system, the context diagram's components have been extensively broken down in the diagram below, which was detailed in the SRS previously.

****

****

## **Algorithm Design**

## **4.4 UI Design**

# Chapter Summary

# References