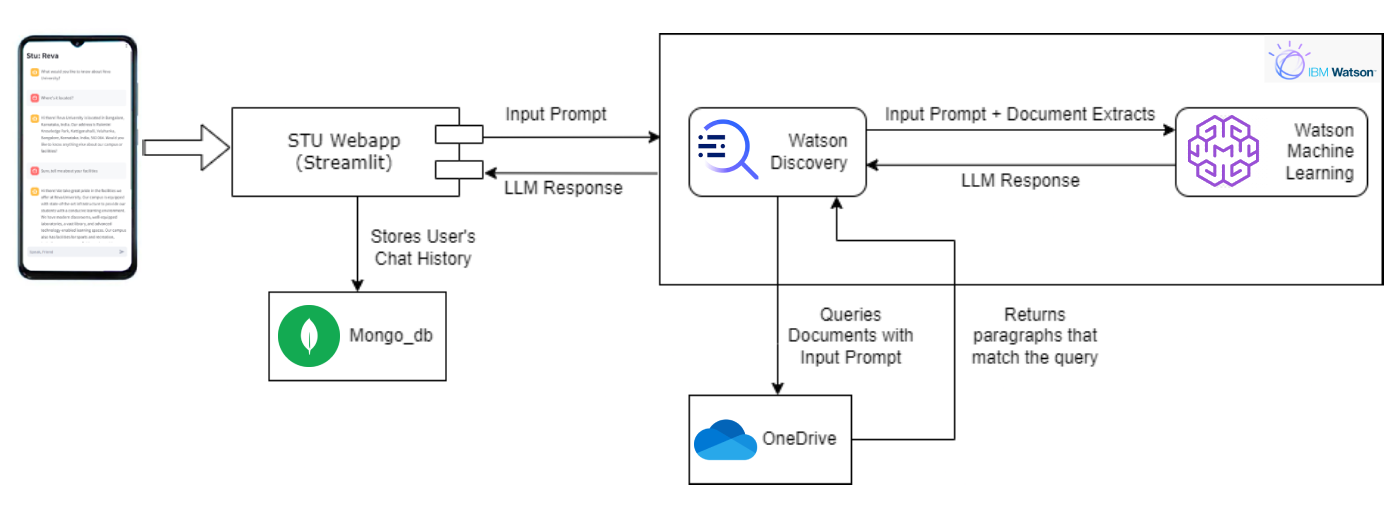
STU - Current Plan

# Overview

Stu is a chatbot that uses a Large Language Model (LLM) to create responses for the user, and can be modified to have specific domain based knowledge for any client through a Retrieval Augmented Generation (RAG) approach, where a dataset of documents directly relevant to the client is provided to the chatbot, and for every response, the relevant information is extracted from those documents.

To implement this, our current plan is to use IBM’s Watson platform, specifically the services Watson Machine Learning, Watson Discovery, and Watson Assistant. We have two possible approaches that can be immediately implemented.

# Approach 1

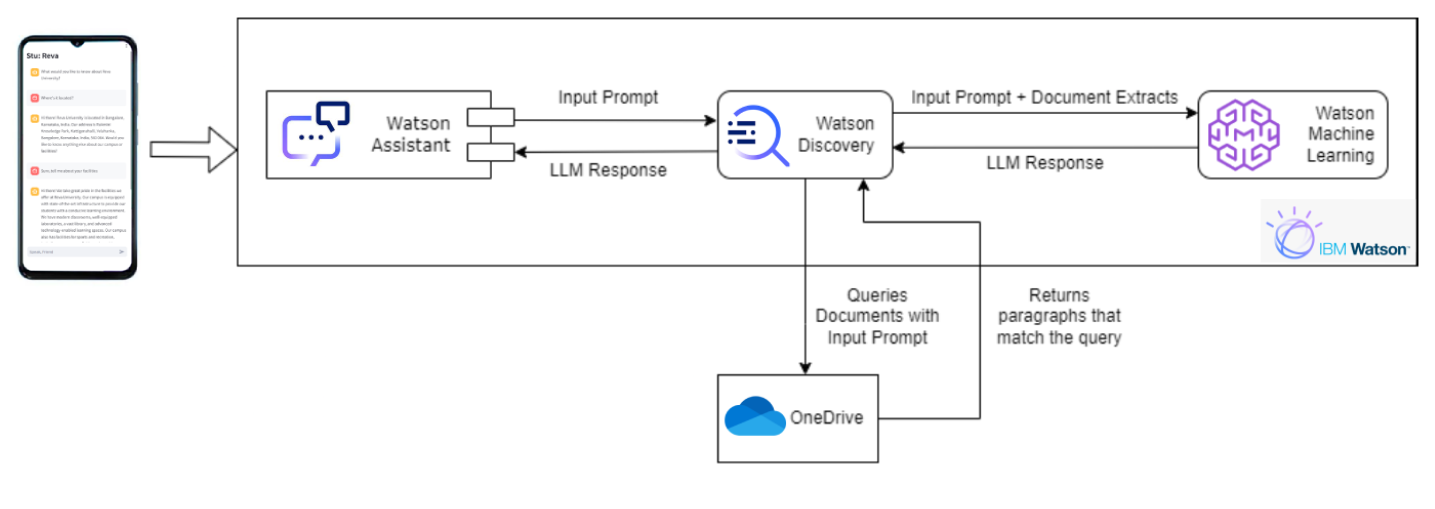


In Approach 1, we will use Streamlit as the interface for Stu, and use APIs to connect it to Discovery and Watson Machine Learning (WML). The user’s prompt is used as a query in Discovery’s database, which will be a set of documents stored on Onedrive. This will return a set of extracts from the document that match the query, and this will be sorted from most relevant to least. This approach also allows to maintain the context from between 5 to 35 previous messages, depending on the length of those messages.

The extracts are added to the user’s prompt and set as input to WML, where an LLM will create a response for it. Currently, we use LLAMA 70b as the LLM, which is the largest open source model available in WML. The response generated is passed as output on the chat interface.

When then happens, both the input and output are stored as an entry in MongoDB, with a few other parameters like the User’s ID, conversation ID, and Timestamp. This allows both the user and the admins to retrieve any user’s conversations for analysis.

# Approach 2



Approach 2 is almost completely within the Watson suite. We will use IBM’s Watson Assistant as the chat interface, and use backend functions provided by IBM called Actions. The user’s prompt will first be analysed by Assistant to determine it’s intent, and then it’s sent to Discovery as a dataset query. The results of this query are attached to the prompt and then sent as input prompt to WML, where an LLM will create a response. This response will be displayed in the chat interface.

There are certain limitations, unlike Approach 1 where the backend is done in Python and is only limited to what can be done with a programming language, Assistant requires any configuration to be done through actions, which are more comparatively arcane. The bot will only remember the last three messages in the conversation.

Due to a natural language analysis being done before the prompt is sent to Discovery and WML, there is an additional delay, which can last anywhere between 3-6 extra seconds. Since the entire flow, other than the dataset, is within the Watson suite, there isn’t a way to extract and store User conversations.