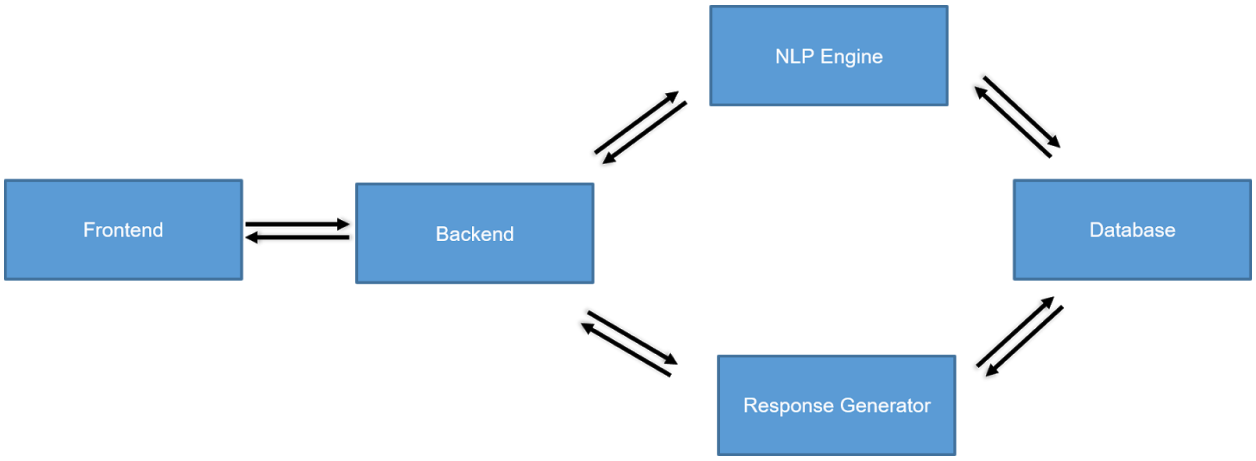
SYSTEM DOCUMENTATION

Purpose of the system:

Wipro offers Work integrated learning program(WILP) and Wipro Academy of Software Excellence(WASE) courses to ensure better education culture in India. So the students who pursue these courses have a variety of common queries like the queries related to attendance, Transfer policies, Examinations, etc. So the main purpose of the chatbot that we have developed is to answer such queries of the students. The chatbot is developed in such a way that it can not only handle the queries that are already provided but it can also handle future queries easily just by importing the excel sheet for bulk upload or by adding one query at a time using the admin panel. The purpose was to make a flexible chatbot such that wipro can extend and modify its functionalities for future use easily.

Structure:



Technologies used: Machine learning, Deep learning, Natural language processing
Languages used: Frontend- Android studio, HTML, CSS, JQuery
Backend- Python
NLP Engine- Python
Response Generator- Python, SQL
Database- SQL (MySQL)
Technologies used: DJANGO and MySQL
Tools used: Python IDE, Scipy, Pytorch, Sentence Transformer

Sequence of activities:

The frontend is responsible for dealing with the end user. The input (i.e. query) can be typed in by the user in either the Web app or Android App. The frontend will then pass on this query to the backend component. The backend component is responsible for connecting the various components of the chatbot. The query will now be passed from the backend to the NLP Engine. The NLP engine will pre-process the data and convert it into sentence embeddings. The NLP engine will return the matched question (if it is above a certain score) to the response generator team. The response generator will then extract the appropriate answer from the database and return the answer to the backend. If the query is not matched, the response generator team will store this query in a separate table in the database for further analysis. The backend will then send this appropriate response to the frontend by which the user can see the answer.