## System Highlights

- Efficient Customer Relationship Management Powered by Artificial Intelligence.
- Real-time feedback to Customer Complaints.
- System deployed into production on a Client-Server architecture system.
- Server implements sophisticated Machine Learning and Natural Language Processing Algorithms.
- Real-time data visualization using Kibana for allowing the client to keep an eye on the performance of the system.

Description

1. Fetching problem categories from the customer complaints and classify them using NLP and Text Analytics.
2. Check for the issue category and then also check for customer Lifetime value using various factors like Credit Score, Age, Available Balance, Salary, Is Customer Active or not?
3. Generate a ticket if the text is a complaint and not a feedback, or a text to check the status.
4. Evaluate the customer based on customer_score, using different values from the predictors.
5. Store the complaint in the database and update the index in Elasticsearch for real time data visualization.
6. Send the response to the customer based on the type of category predicted by the NLP module.
7. Refresh the dashboard in Kibana in every 5 seconds to check if there is any new data added.

System Architecture
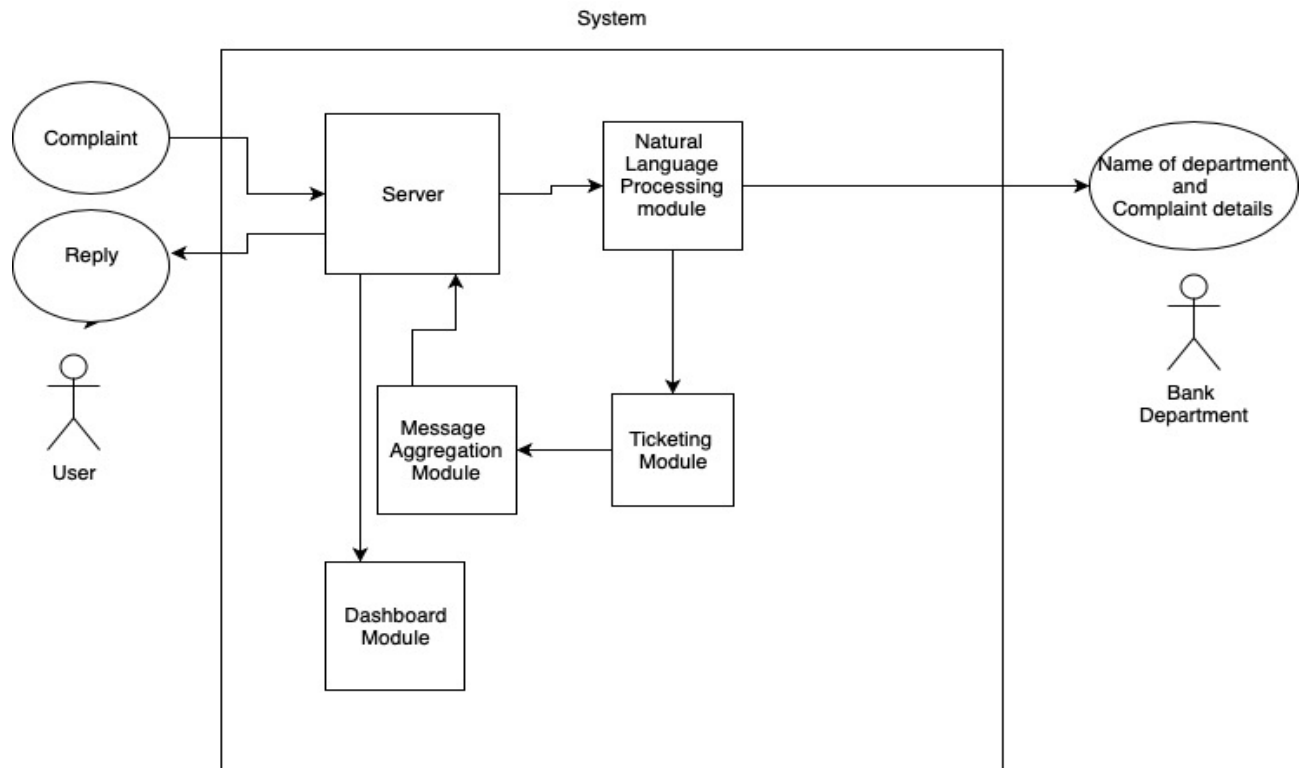
Files

1. **Predict_function.py**
   This file is responsible for reading the trained classification model; it takes the complaint from the user in the form of a text string, extracts features and classifies the text into 3 categories that the complaint might be related to: 'Mortgage', 'Credit reporting', 'Debt collection'.

2. **App11.py**
   'App11.py' is responsible to scrap the data sent by the customer over SMS on the registered phone number and extract the message body to apply NLP and Text Analytics to classify the category of the complaint and pushing coupons to pacify the customer based on customer score and category of complaint.
   It also automatically generates a ticket number which the customer can use to track the status of their complaint.

   This file shows the prediction of the Customer Lifetime value depending on the various factors like credit score, Age, balance, Salary, id customer active or not. This is done by predicting the classifying variable"Exit"[0 if the customer is still with the bank and 1 if the customer will leave bank] using Supervised ML techniques with an accuracy of 80%.

3. **appupdate.py**

   Appupdate.py enables the customer to receive the status of their complaint as a text message on their number whenever the status of the complaint is changed on the server side.

Process Flowchart

```
                        ┌─────────────┐
                        │    Start    │
                        └──────┬──────┘
                               │
                               ▼
                    ┌────────────────────┐
                    │ Send Text Message  │
                    │ from Client Phone  │
                    └─────────┬──────────┘
                              │
                              ▼
                    ┌────────────────────┐
                    │ Receive text from  │
                    │      server        │
                    └─────────┬──────────┘
                              │
                              ▼
                    ┌────────────────────┐
                    │ Extract properties │
                    │   of the text      │
                    │  (number, zipcode, │
                    │        text)       │
                    └─────────┬──────────┘
                              │
                              ▼
                    Identify type of text:
                    Feedback, Complaint,
                          Status
```

Start

Send Text Message from Client Phone

Receive text from server

Extract properties of the text (number, zipcode, text)

Identify type of text: Feedback, Complaint, Status

Status

Complaint

Feedback

Extract the ticket number

Complaint

Generate default message

Fetch complaint Status from Database

Call predit_function.py to predict category of complaint

Send the default message to customer

Send status of the ticket number to the customer

Identify category of complaint

use textrank algorithm to summarize complaint

Create ticket for predicted category

Notify the predicted department representative

Send the ticket number to customer for reference

Store the ticket along with all category, timestamp, summarization info in the database

Store the data in Elasticsearch index, fintech_hackathon_consumer13

Refresh the Data in the index in every 5 seconds

Refresh the dashboard based on the real-time data received

Stop