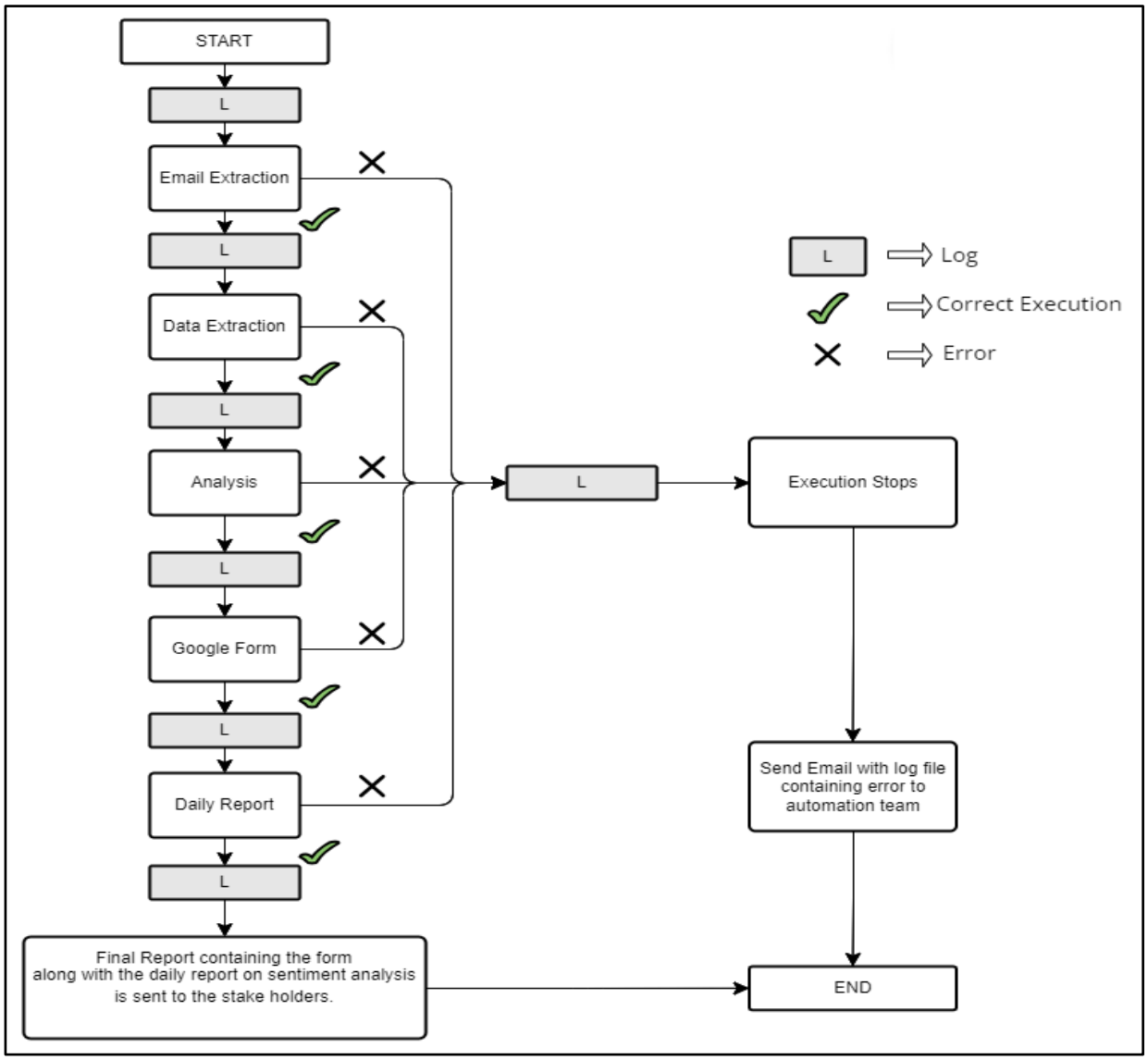
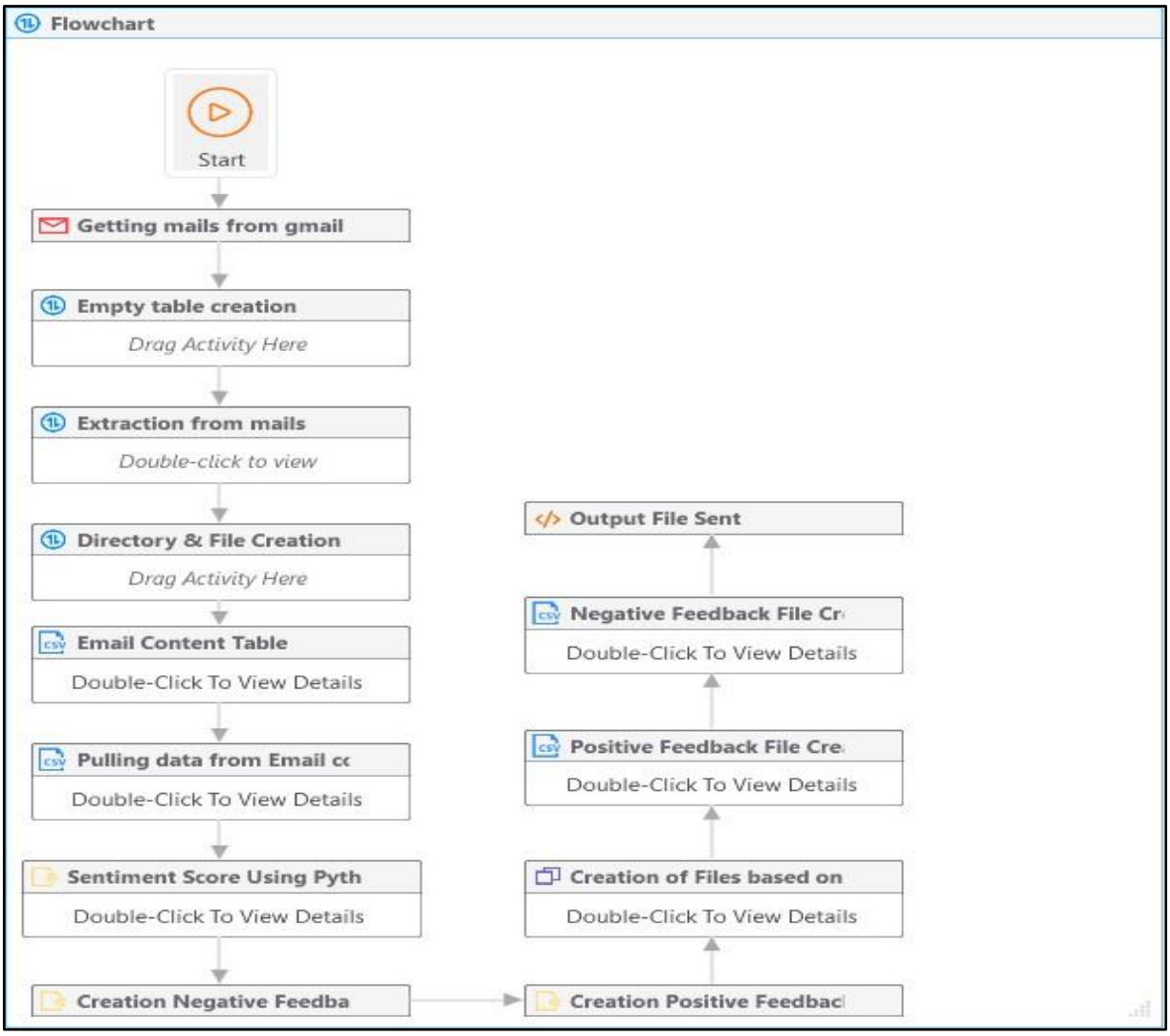
Introduction: In this abstract, we are going to discuss the automation procedure that we follow for handling customer feedback from the email channel. The overall outline is as follows:

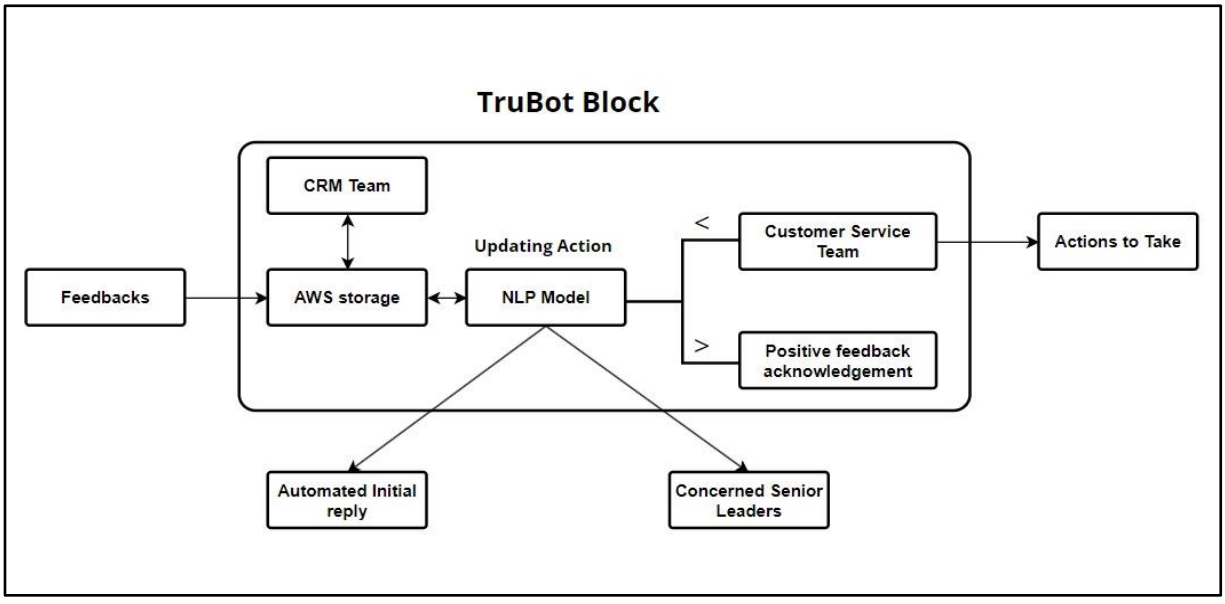
Key Important stages:

1. Collecting feedbacks.
2. Integrating Trubot.
3. Logging: We are logging information of every step into a file to keep a track of execution which will be useful for debugging incase of any error.
4. AWS Storage Service: Data collected through Trubot will be stored in S3 buckets in specific format (Here we will be storing in CSV format but measures can be taken to further compress and store).
5. Applying NLP model: Data from storage will be pulled using Python code, pre-processed, and feed to the NLP model for sentiment extraction. The results will be stored for further use.
6. Sending reviews/insights/output files to Customer Service Team: The detailed output file will be shared with the concerned team for actions, and a detailed analysis of sentiments will be shared with the senior leaders in the organization at regular intervals.
7. Follow up by Customer Service team: Feedback will be handled based on their category. The customer service team will try to resolve the issue related to neutral and negative sentiment emails.



***Proposed Flowchart (Under development stage) Built in TruBot Designer.**





Data Extraction:

Connecting the Gmail using the IMAP and pulling email data. Parts of email extracted

- From (senders email id)
- Subject
- Body
- IP address (Location Purpose)
- Timestamp (time of email received)
- Attachments.

AWS storage S3: After extracting the data, it will be stored in the following tables:

We will have these tables (customers, orders, attachments, and feedback). All these tables are necessary for the NLP model (except for feedback). The customer service team will require this information for addressing the customer issues. After taking actions/closing customer tickets, that action needs to be updated in the table.

NLP Block: This block can be further broken down into Data preparation and manipulation, sentiment score Generation (for each email/feedback), Topic modelling for every email, automated reply and Analysis block.

Note: **This can be done by using pretrained model and web API also.**

1. **Data Preparation:** This is a set of python code which will clean the email and bring it into the required format for storage. e.g. taking certain parts of email, removing unnecessary words, creating word cloud, word Dictionary etc.
2. **Automated reply:** It will be sent for acknowledging the feedback at early stages.
3. **Sentiment Score:** After the data is cleaned it will be fed to the sentiment scoring code. Here a set of python code (using the libraries) will generate a sentiment score for the 3 categories i.e. Positive, Neutral and Negative or the Score between (Most Negative: -1 and Most Positive: 1). There are 2 major libraries we have use
    a. NLTK:
        i. Method -SentimentIntensityAnalyzer()
        ii. Probabilities for Positive, Neutral and Negative and also the overall score i.e. compound

```
# NLTK
sia.polarity_scores('I loved and Enjoyed the DataMatics Hackathon.I am very happy')
```

```
{'neg': 0.0, 'neu': 0.349, 'pos': 0.651, 'compound': 0.9041}
```

    b. TEXTBLOB
        i. Method -TextBlob()
        ii. Score Range: -1 to 1 (Most Negative: -1 and Most Positive: 1)
        iii. Support Probabilities: Confidence of the score generated

```
# TEXTBLOB
TextBlob('I loved and Enjoyed the DataMatics Hackathon.I am very happy').sentiment
```

```
Sentiment(polarity=0.7333333333333334, subjectivity=0.8333333333333334)
```

4. **Topic Modelling:** Based on the subject and body of the email, the model will give every email one of the three existing labels (**Orders, Service or Delivery**) & this is done by LDA
   LDA: Probabilistic way to provide Label/Topic to the content.
   E.g. if there are words like (late, Delivery agent, missing, location, etc.) then it will be labeled as Delivery.

   After all of this is done, we update the feedbacks table with the sentiment and the labels of the mail in a way that we have no table inconsistencies.

5. **Logging:** We log the details of every step into a file for safety issues in the context of execution. Whenever we face any error while execution, the process stops, and a mail containing the details of the error is sent to the automation team for their reference.
6. **Output:** After labeling the mails, these emails are categorized based on the labels and sent to different customer service teams for further actions.

   Positive Feedback: Acknowledge and **thank the customer via email**. Store relevant data in AWS for future customization of search results.

   Neutral Feedback: Neutral emails indicate **mixed feelings**. The customer service team will reach out to understand the customer's feedback better. Details will be updated in AWS upon resolution.

   Negative Feedback: The customer service team will receive details such as **sentiment score, attachments, and order history etc.** They will work on resolving the issue and track progress to improve response effectiveness. Updates on resolutions will be logged in AWS for the CRM Success Team and stakeholders to monitor performance and statistics.

7. **Final Report for the Senior Leaders:** Based on the feedbacks generated, we will provide the stake holders with an analysis of sentiments across the categories (Product, Service, Delivery) along with few statistics related to resolution of issues, positive and negative feedback rates etc.