TICKET AUTOMATION SYSTEM USING RPA

A PROJECT REPORT

Submitted by

SANJAY.R.B [711716104075]

SAM DANY.R [711716104074]

RAMSHANKARAN.Y [711716104069]

PRATHVIK.S [711716104063]

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KGiSL INSTITUTE OF TECHNOLOGY, SARAVANAMPATTI

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2020

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"TICKET AUTOMATION SYSTEM USING RPA"** is the bonafide work of "**SANJAY.R.B, SAM DANY.R, RAMSHANKARAN.Y, PRATHVIK.S"** who carried out the project work under my supervision.

SIGNATURE
Prof. J. SRIKANTH
HEAD OF THE DEPARTMENT

Computer Science and Engineering
KGiSL Institute of Technology
Saravanampatti
Coimbatore-35.

SIGNATURE
Ms. R. JAYASHREE
SUPERVISOR
Assistant Professor
Computer Science and Engineering
KGiSL Institute of Technology
Saravanampatti
Coimbatore-35.

Submitted for the Anna University Viva-Voce examination held on 22.09.2020

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We express our deepest gratitude to our **CHAIRMAN AND MANAGING DIRECTOR** for providing us with an environment to complete our project successfully.

We are very grateful to our **DIRECTOR Dr.R.Ravichandran, Ph.D and PRINCIPAL Dr.VijayaChamundeswari, ME, Ph.D** for their valuable guidance and blessings.

We would like to thank **Prof.J.Srikanth, HOD, Department of Computer Science and Engineering** for his unwavering support during the entire course of their project first phase work and who modelled us both technically and morally for achieving greater success in this project work.

We express our sincere thanks to our faculty guide **Ms.R.Jayashree, Assistant Professor, Department of Computer Science and Engineering.** We also thank our industry guide **Ms.Sandhya, Project engineer** for their constant encouragement and support throughout our course, especially for the useful suggestions given during the course of the project period and being instrumental in the completion of our project with their complete guidance.

We also thank all the **faculty members** of our department for their help in making this project a successful one.

Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all they meant to us during the crucial times of the completion of our project.

ABSRTACT

Under the acceleration of the wheel of technology around the world, various ways of automating information have occurred to replace old-fashioned ways that don't serve the pace of everyday lifestyle. There are many things that we should change in our work systems such as paper, stand-alone systems, phone calls or even email. Every kind of companies has different sections that help the process of paperwork concerning personnel and maintenance. However, some of this paperwork might be overlooked or delayed, which could lead to slow down the outcome of the organization. So, they need to improve their system work to save time, effort and make an efficient mechanism to follow requests and tickets of any organization or customer. So, we should replace any old system with RPA to facilitate the work of supporting and following tickets. As a solution to that, we have come with the idea of the "Ticket Automation System Using RPA." The application aim is to eliminate manual intervention in ticket creation and it raise a ticket based on the complaint mail. If the details are incomplete (e.g. customer id is missing), send a mail to customer asking for missing details and Link the subsequent responses from the customer to the original ticket. Recognize the bounced mails and initiate appropriate action via sending auto response to template-based mails (complaints/queries) i.e. no free text.. It is a modern way to solve problems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

ABBREVIATION	EXPLANATION
NLTK	The Natural Language Toolkit
NLP	Neuro-Linguistic Programming
SRS	Software Requirements Specification
PyPI	Python Package Index
SAP	System Applications

CHAPTER 1
INTRODUCTION

The ticket automation is playing a huge role in maintaining a successful support operation and is helping to revolutionize the industry. The days of the “gatekeeper” that directed customer support issues to the right people are over. Now, with customer support software and ticket automation, triggers can be created so if a ticket is submitted via email or a customer portal it goes to a specific agent or group. For example, a trigger where the subject line contains the word “how” (i.e. how do I add a user) could automatically route all emails to the training group,
And it also Remind customers to follow up, Close tickets automatically, Get alerts for urgent tickets, Weekends and holidays

In this project we are trying to perform few of the basic ticket automation operations via uipath.

1.1 PROBLEM DEFINITION

- Eliminate manual intervention in ticket creation, so raise a ticket based on the complaint mail. A database used for storing and administering all types of data required for efficient and accurate functioning of the bot.
- If the details are incomplete (e.g. customer id is missing), send a mail to customer asking for missing details and Link the subsequent responses from the customer to the original ticket.
- Recognize the bounced mails and initiate appropriate action via sending auto response to template-based mails

1.2 OBJECTIVE OF THE PROJECT

The objective of this project is to contribute to the solution of the problem of eliminating manual intervention in ticket creation, so raise a ticket based on the complain mail. The main objectives of the project are as follows:

- **Uipath** : To develop a bot in order to fetch the mail from the mail server.
- **Python** : To develop a python module in order to assign a priority based on analyzing the key words
- **Database:** To develop a database were all the relevant information about questions, answers, keywords, logs and feedback will be stored.
- **Algorithm:** To develop a keyword matching algorithm using NLTK in order to identify the keywords and tokenize.

1.3 SIGNIFICANCE OF THE PROJECT

- Effective in both Installation and Working with the Queues.
- Previously solved solutions are stored in a separate Database.
- When same queries arise by the other users, then it matches to the saved results.
- This makes the Ticketing System Faster Comparatively.
- Applies certain algorithms (Sequential / Priority) to process the queries even faster.
- Splits entire queries by word tokenization.
- Missing Details and Fraud are identified by NLP, then Updates the information to existing tickets accordingly.

1.4 OUTLINE OF THE PROJECT

- Get the complaint mails from the users .
- Segregate the mail, understands the user queries and assign the ticket to the particular complain mail
- If the queries already solved, then replay with the existing solution and close the ticket.
- If the problem is not solved then perform the analysis on the mail and raise the ticket to appropriate person.

CHAPTER 2

LITERATURE REVIEW

Management and maintenance of IT infrastructure resources such as hardware, software and network is an integral part of software development and maintenance projects. Service management ensures that the tickets submitted by users, i.e. software developers, are serviced within the agreed resolution times. Failure to meet those times induces penalty on the service provider. To prevent a spurious penalty on the service provider, non- working hours such as waiting for user inputs are not included in the measured resolution time, that is, a service level clock pauses its timing. Nevertheless, the user interactions slow down the resolution process, that is, add to user experienced resolution time and degrade user experience. Therefore, this work is motivated by the need to analyze and reduce user input requests in tickets' life cycle. To address this problem, we analyze user input requests and investigate their impact on user experienced resolution time. We distinguish between input requests of two types: real, seeking information from the user to process the ticket and tactical, when no information is asked but the user input request is raised merely to pause the service level clock. Next, we propose a system that preempts a user at the time of ticket submission to provide additional information that the analyst, a person responsible for servicing the ticket, is likely to ask, thus reducing real user input requests. Further, we propose a detection system to identify tactical user input requests. We observed that around 57% of the tickets have user input requests in the life cycle, causing user experienced resolution time to be almost twice as long as the measured service resolution time. The proposed preemptive system preempts the information needs with an average accuracy of 94–99% across five cross validations while traditional approaches such as logistic regression and naive Bayes have

accuracy in the range of 50–60%. The detection system identifies around 15% of the total user input requests as tactical. Therefore, the proposed solution can efficiently bring down the number of user input requests and, hence, improve the user-experienced resolution time.

CHAPTER 3
SYSTEM ANALYSIS

System analysis is a problem-solving technique that decomposes a system into component pieces of purpose of studying how well those component parts work and interact to accomplish their purpose the following chapter provides the detail description of the existing system. It also provides an overview of the proposed system and feasibility of the ticket automation system.

3.1 EXISTING SYSTEM

In existing system Man power is required to assign a unique ticket. It also saves the solved solutions in a database in order to solve when the same queries arise.

3.1.1 Ticket Automation

A Ticket automation project is built using NLP algorithms that analyses user's queries and understand user's message. The system is designed for support team in all product organizations where users can ask any product related questions. The system recognizes user's query and understands what he wants to convey and simultaneously answers them appropriately. The NLP algorithm analyses users queries then convert into tokens. Then the tokens are assigns to the persons in the particular domain.

3.2 DRAWBACKS

3.2.1 Ticket Automation

- It is used to assign the tokens to the persons in the particular domain.
- It can provide the solutions only for the frequently solved queries.

3.3 PROPOSED SYSTEM

The proposed system is RPA bot which uses NLP algorithm to solve the user's queries. This system should be able to analyze and understand user's queries and react accordingly.

For any product related queries, we have put a mail to customer support team and it will be forwarded to the particular domain members in order to provide the solution. Frequently solved queries will be stored in the database to provide solutions when the similar queries raised by the user. Without this system an organization needs a team for assigning the tokens to the domain related persons.

3.4 FEASIBILITY STUDY

Feasibility studies are conducted where large sums are at stake. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing ticketing systems and threats present in the environment, the resources required to carry through, and ultimately the prospects for success in this system.

3.4.1 Tests of Feasibility

Feasibility study is conducted once the problem clearly understood. Feasibility study is necessary to determine that the proposed system is feasible by considering the technical, operational, and economical

factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system of the automation system. Feasibility study encompasses the following things:

- Technical Feasibility
- Economical Feasibility
- Operational feasibility

3.4.1.1 Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project of Ticket automation. The technical requirements are then compared to the technical capability of the Ticketing system. The systems project is considered technically feasible if the internal technical capability is sufficient to support the Ticket automation system's requirements.

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the project feasible within the limits of current technology?
- Does the technology exist at all?
- Is it available within given resource constraints?
- Manpower- programmers, testers & debuggers
- Are the current technical resources sufficient for the new system?

3.4.1.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if

it is developed and implemented. Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis

Phase of ticketing system development.

3.4.1.3 Economical Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of the Ticketing system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs.

If benefits outweigh costs, then the decision is made to design and implement the ticket automation system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs and system study?

CHAPTER 4

SYSTEM SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS:

4.1.1 Mailing:

a. The queries should be mailed to the customer support mail.

b. The RPA bot should read a mail from the customer support mail inbox.

c. The system shall inform the user about the adequate information needed to solve solution.

4.1.2 Logs:

a. The system should maintain a log of the current question and answer if the user is not satisfied.

4.1.3 Feedback:

a. The user should be able to leave feedback, which is comprised of a text message.

b. The user will get a ticket id in order to have a reference for the user.

c. Feedback management: The administrator should be able to view and delete feedbacks.

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 Portability:

a. The system should run on a Uipath software.

b. The system should run with the required python scripts.

4.2.2 Maintainability:

a. The system should be easy to maintain.

b. There should be a clear separation between the interface and the business logic code.

c. There should be a clear separation between the data access objects that map the database and the business logic code.

4.2.3 Exception handling:

a. Exceptions should be reported effectively to the user if they occur.

4.2.4 Ethics:

a. The system shall not store any information about its users.

4.3 HARDWARE REQUIREMENTS

Processor	:	Dual core processor
RAM	:	2 GB
Hard Disk	:	250 GB
Monitor	:	16'' Color Monitor
Keyboard	:	Standard 110 keys
Pointing Device	:	Mouse

4.4 SOFTWARE REQUIREMENTS

Programming Language	:	Python
Operating System	:	Windows/Ubuntu/Linux/Mac
Front End	:	RPA Uipath
Back End	:	Excel
Web Browser	:	Mozilla Firefox, Google Chrome

CHAPTER 5

SOFTWARE DESCRIPTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between users and RPA bot on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

5.1 FRONT END

The front end is designed using Uipath which includes a collaborative platform for bot creating RPA bot. Here all the bot tools are integrated and it allows automatic detection of entities. It uses NLP and Gensim.

5.1.1 PYTHON

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate.

there are two major Python versions: **Python 2 and Python 3**. Both are quite different.

Features

- **Easy to code**: Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.
- **Python is Portable language**: Python language is also a portable language. for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform. Faster: It is faster than other scripting language e.g. asp and jsp.
- **Large Standard Library** : Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

Advantages

- **Presence of Third Party Modules**: The Python Package Index (PyPI) contains numerous third-party modules that make Python capable of interacting with most of the other languages and platforms.
- **Extensive Support Libraries**: Python provides a large standard library which includes areas like internet protocols, string operations, web services tools and operating system interfaces. Many high use programming tasks have already been scripted into the standard library which reduces length of code to be written significantly.

- **Productivity and Speed**: Python has clean object-oriented design, provides enhanced process control capabilities, and possesses strong integration and text processing capabilities and its own unit testing framework, all of which contribute to the increase in its speed and productivity. Python is considered a viable option for building complex multi-protocol network applications.

5.1.1 UIPATH

UiPath Studio – a tool that enables you to design automation processes in a visual manner, through diagrams. UiPath Robot - executes the processes built in Studio, as a human would. UiPath Orchestrator - a web application that enables you to deploy, schedule, monitor and manage Robots and processes.

Features

- Uipath helps to perform the activity by using the drag and drop features.
- It can be hosted in cloud environments or virtual terminals.
- Scraping solution that works with any application like,.Net Java, Flash, PDF, Legacy, SAP, with absolute accuracy.

Advantages

- Increased compliance
- Best customer experience
- Productivity improvement
- Good management capabilities

5.1.2 NLP

Natural-language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data.

Advantages

- No training.
- Relives burden of learning syntax.

5.1.3 GENSIM

Gensim is a free Python library designed to automatically extract semantic topics from documents, as efficiently (computer-wise) and painlessly (human-wise) as possible.

Gensim is designed to process raw, unstructured digital texts ("plain text").

Features

- Memory independence – there is no need for the whole training corpus to reside fully in RAM at any one time (can process large, web-scale corpora)
- Memory sharing – trained models can be persisted to disk and loaded back via mmap. Multiple processes can share the same data, cutting down RAM footprint.
- Efficient implementations for several popular vector space algorithms, including

5.2 BACK END

The back end is designed using excel, whose primary function is to store data and retrieve it later, as requested by other software applications.

5.2.1 Excel

Microsoft Excel is a spreadsheet program that is used to record and analyze numerical data. Think of a spreadsheet as a collection of columns and rows that form a table. Alphabetical letters are usually assigned to columns and numbers are usually assigned to rows. The point where a column and a row meet is called a cell. The address of a cell is given by the letter representing the column and the

Excel offers at least three ways to set up data so your reports and analyses can use it easily as a reliable data source. Excel offers three general ways to arrange data in your spreadsheet so you can use it as a database with your worksheet formulas: Simple (or "Gray Cell") Tables, which I've used since Excel 2.0.

Features

- PivotTables summarise large amounts of Excel data from a database that is formatted where the first row contains headings and the other rows contain categories or values.
- Conditional formatting, as its name suggests, changes the format of a cell dependent on the content of the cell, or a range of cells, or another cell or cells in the workbook.
- Sorting and filtering your data will save you time and make your spreadsheet more effective.

Advantages

- Easy data entry and operations
- Accurate comparisons and analysis options
- Allows graphical representation of data
- Compatible with other business applications
- Ready to use formulas

CHAPTER 6

PROJECT DESCRIPTION

In this project we are trying to perform few of the basic ticket automation operations via uipath. The objective of this project is to contribute to the solution of the problem of eliminating manual intervention in ticket creation, so raise a ticket based on the complaint mail.

6.1 OVERVIEW OF THE PROJECT

The RPA bot provides a complete set of methods in order to perform the ticketing operations. Get the complaint mails from the users.

- Segregate the mail, understands the user queries and assign the ticket to the particular complaint mail
- If the queries already solved, then reply with the existing solution and close the ticket.
- If the problem is not solved then perform the analysis on the mail and raise the ticket to appropriate person.

6.2 MODULE DESCRIPTION

6.2.1 Segregating the mails

The Segregating the mail module makes the way for understanding the user queries and assigns the ticket to the particular complaint mail. This module gets the mails from the company mail inbox and analyses the mail for further processing. This is done by the RPA bot which collects the mail by connecting it to the IMAP mail server.

6.2.2 Auto Reply

The Auto Reply module is used to solve the problems which was raised previously and solved by the technical domain person. And the previously solved solutions will be stored in the database

6.2.3 Ticket Generation

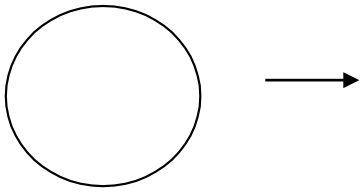
When the solution generated by the chatbot is not completed then the ticket will be generated and the ticket details copy will be given to the customer through mail. This ticket will be forwarded to the service person. After solving the problem the result will be provided to the costumer.

6.3 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identifies how the information flows through the processes. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs. The data flow diagram also depicts the flow of the process and it has various levels. The initial level is context level which describes the entire system functionality and the next level describes each and every sub module in the main system as a separate process or describes all the process involved in the system separately.

Data flow diagram are made up of number of symbols,

Square representing external entities, which are sources or destinations of data.



Circle representing processes, which take data input, do something to it and output it.

Arrows representing the data flows, which can either, be electronic data or physical items.

Parallel lines representing data stores, including electronic stores such as databases or XML files and physical stores

6.3.1 DFD Level 0:

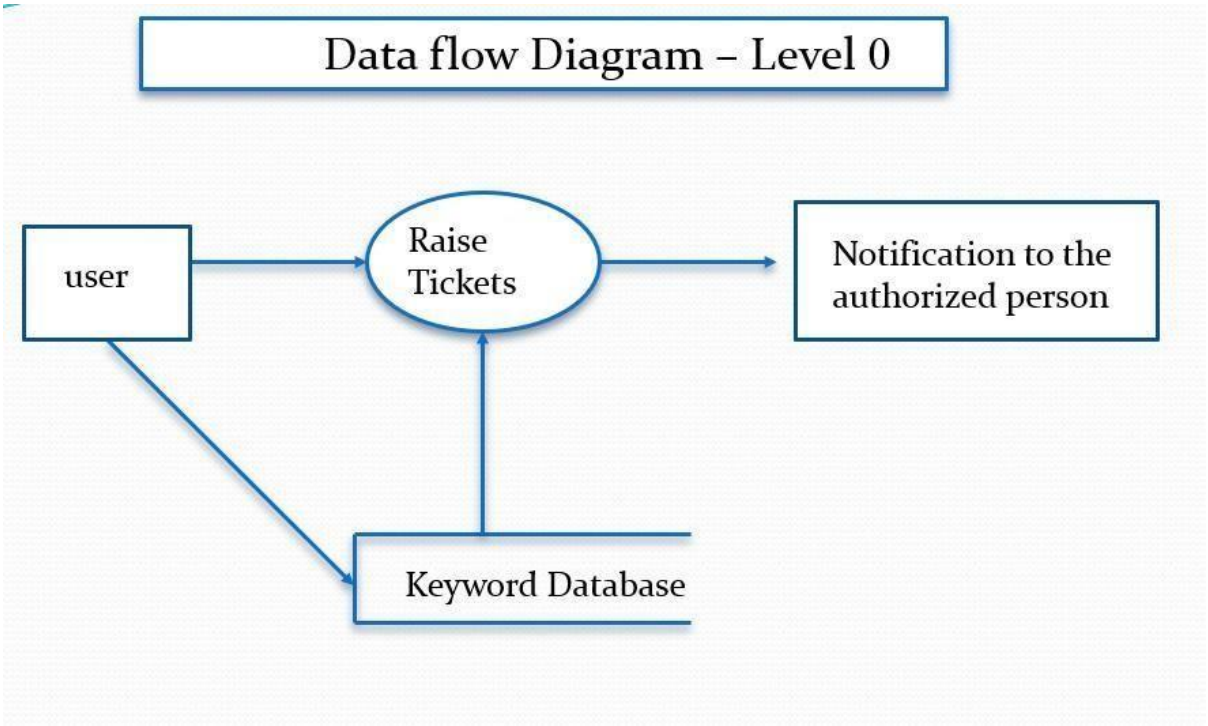


Fig 6.3.1 DFD Level 0

6.3.2. DFD Level 1:

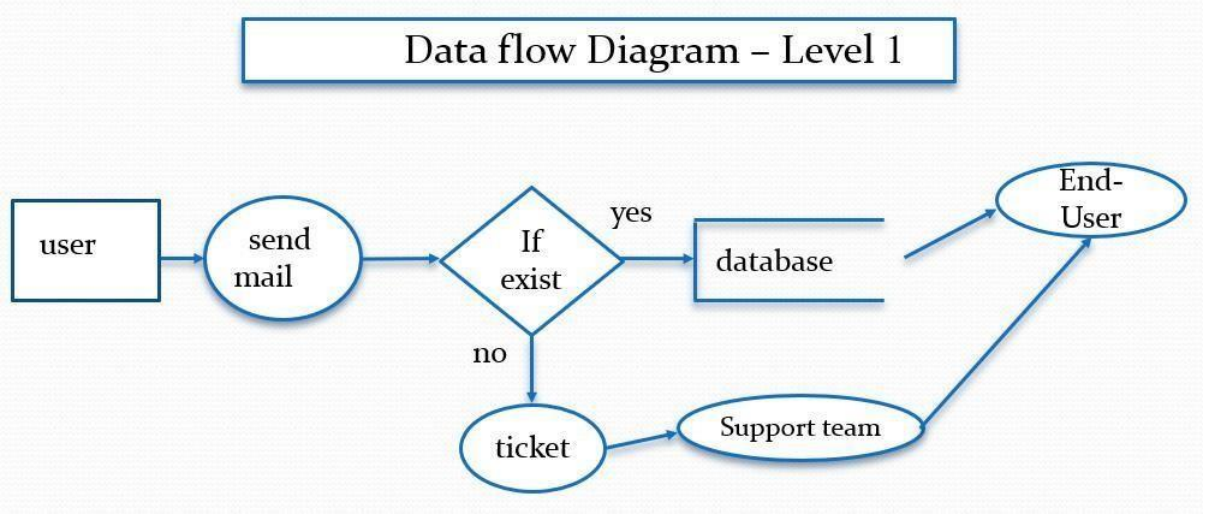


Fig 6.3.2 DFD Level 1

6.4 ER DIAGRAM

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes.

Entity–relationship modelling was developed for database design by Peter Chen and published in a 1976 paper. Some ER modelers show super and subtype entities connected by generalization-specialization relationships, and an ER model can be used also in the specification of domain-specific ontology.

6.4.1 Extracting the information

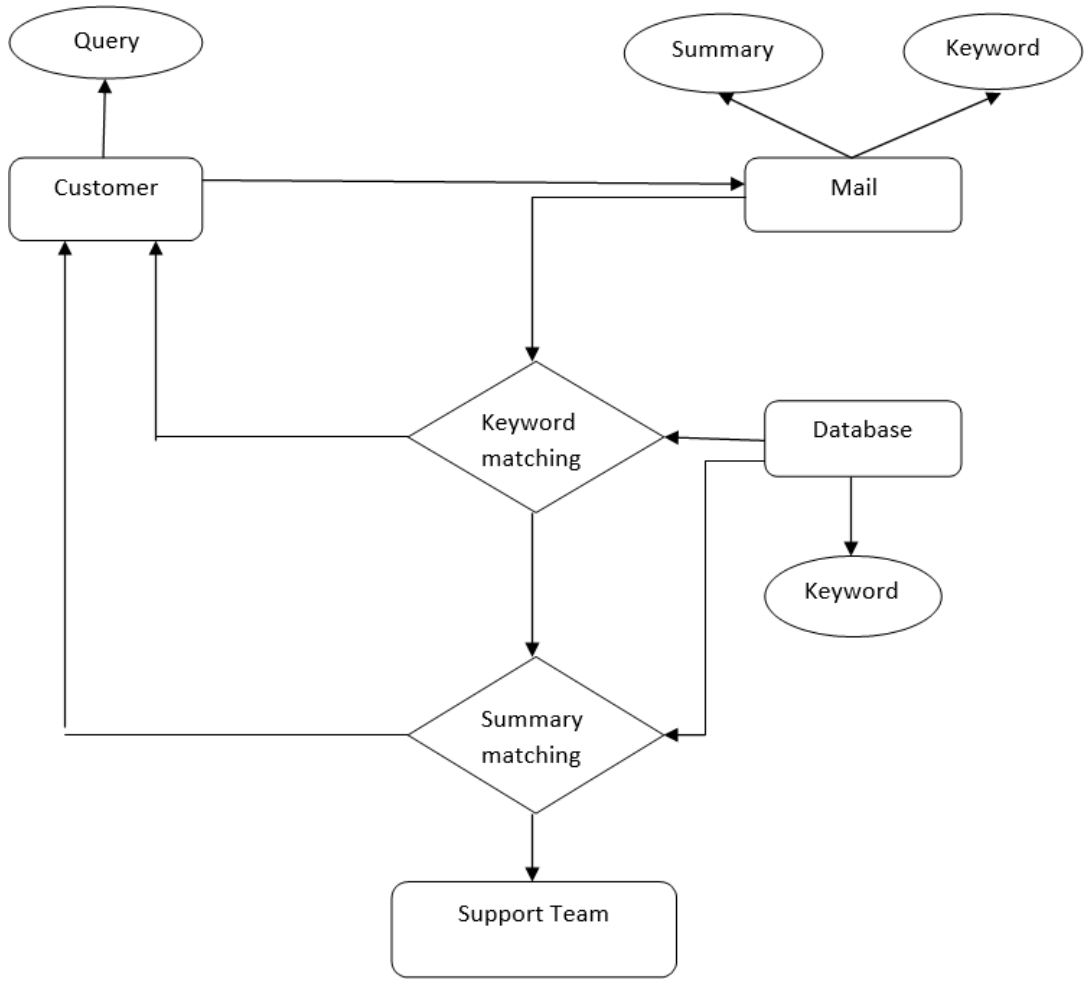


Fig 6.4.1 ER Diagram

6.5 Excel Database Design

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

TABLE 6.5.1 predefined keyword in DB

S.NO	KEYWORDS	SOLUTION
1	Not receiving bills	Provide a valid email id
2	Incorrect billing	Refund will be credited with 2 days
3	Payment not updated	Restart your router
4	Can’t connect to network	Restart your router

TABLE 6.5.2 predefined domain persons in DB

S.NO	DOMAIN	SERVICE PERSON
1	PAYMENT	XXXXX
2	LOS	YYYYY

6.6 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer

to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

. What data should be given as input?

. How the data should be arranged or coded?

. Methods for preparing input validations and steps to follow when error occur.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of mail. Thus the objective of input design is to create an input layout that is easy to follow.

6.7 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help the user for making decisions.

CHAPTER 7

SYSTEM TESTING

System Testing is a level of the software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. By definition of ISTQB system testing is the process of testing an integrated system to verify that it meets specified.

7.1 TESTING METHODS

Software Testing Type is a classification of different testing activities into categories, each having, a defined test objective, test strategy, and test deliverables. The goal of having a testing type is to validate the Application under Test for the defined Test Objective.

For instance, the goal of Accessibility testing is to validate the AUT to be accessible by disabled people. So, if your Software solution must be disabled friendly, you check it against Accessibility Test Cases.

7.2 TYPES OF TESTING

7.2.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this intrusion detection system using IP cameras, every units of code is been tested and the correctness of every module is been ensured.

7.2.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

In this intrusion detection system using IP cameras, the units are been tested as a whole and the testing was successful.

7.2.3 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing usually describes what the system does. Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system.

Functional testing has many types:

- Smoke testing
- Sanity testing
- Regression testing
- Usability testing

7.2.4 Acceptance Testing

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user or other authorised entity to determine whether or not to accept the system.

In this intrusion detection system using IP cameras , the customer's acceptance is been monitored and it is been put into usage.

7.2.5 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box are testing. It is one of two parts of the "box testing" approach of software testing. Its counter-part, black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing.

The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

In this intrusion detection system using IP cameras, all the inner

functionality is been tested and it is been correctly implemented.

7.2.6 Black Box Testing

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In this intrusion detection system using IP cameras, the implementation part is been checked for its correctness.

7.2.6.1 Methods of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

- Functional testing - This black box testing type is related to functional requirements of a system; it is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- Regression testing - Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected
  the existing code.

7.3 TESTING STRATEGY

Test Strategy is also known as test approach defines how testing would be carried out. Test approach has two techniques:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

- Reactive - An approach in which the testing is not started until after design and coding are completed.

Test strategy calls for implementing two entirely different methodologies for testing this project. The intrusion detection system using IP cameras includes a fair amount of manual UI-based testing.

In this intrusion detection system using IP cameras, proactive approach is been used for testing. Since proactive approach is efficient it is been used in this intrusion detection system using IP cameras.

CHAPTER 8

SYSTEM IMPLEMENTATION

8.1 Mail trigger

In the Mail trigger module the real work starts this mail trigger will be running at the background synchronously to get the mail from the organisation mail inbox if there is no email it will wait for 1 minute else if a new mail comes it will redirect it to the check mail module.

8.2 Process mail

This module initially loads the obtained text information's from the previous module the python script which uses NLP and Gensim in order to make the unstructured data to structured by removing space, special characters. Then call getsummary() method gets the summary and getkeywords() method gets the keywords and after generating the summary it will be redirected to the check mail module.

8.3 Check Mail

This module checks with Datastore module whether this problem is solved already or not if it exists it replies with that solution else it raises a ticket to the technical support team.

CHAPTER 9

CONCLUSION & FUTURE ENCHANCEMENT

9.1 CONCLUSION

In the proposed Ticket automation System, the manual intervention of the ticket creation is eliminated it raise a ticket based on the complaint mail. If the details are incomplete (e.g. customer id is missing), send a mail to customer asking for missing details and Link the subsequent responses from the customer to the original ticket. Recognize the bounced mails and initiate appropriate action via sending auto response to template-based mails (complaints/queries) i.e. no free text. This system makes short time response to the customer.

9.2 FUTURE ENHANCEMENT

A login account can be created which deals with customer support services instead of giving queries in the mail. So, this serves as a private login which helps the user to define their queries using the predefined keywords and the customer id can be easily notified to the company with the help of their login id. This eliminate the process of sending the mails regarding the incomplete data (e.g. customer id is missing) to the customer. The system which has this RPA workflow can be considered as a hub system and this hub plays as a role of routing the tickets to the particular persons system(node) according to their domain. This overcomes the process of installation of this workflow in all the system.

CHAPTER 10

APPENDIX

10.1 SOURCE CODE

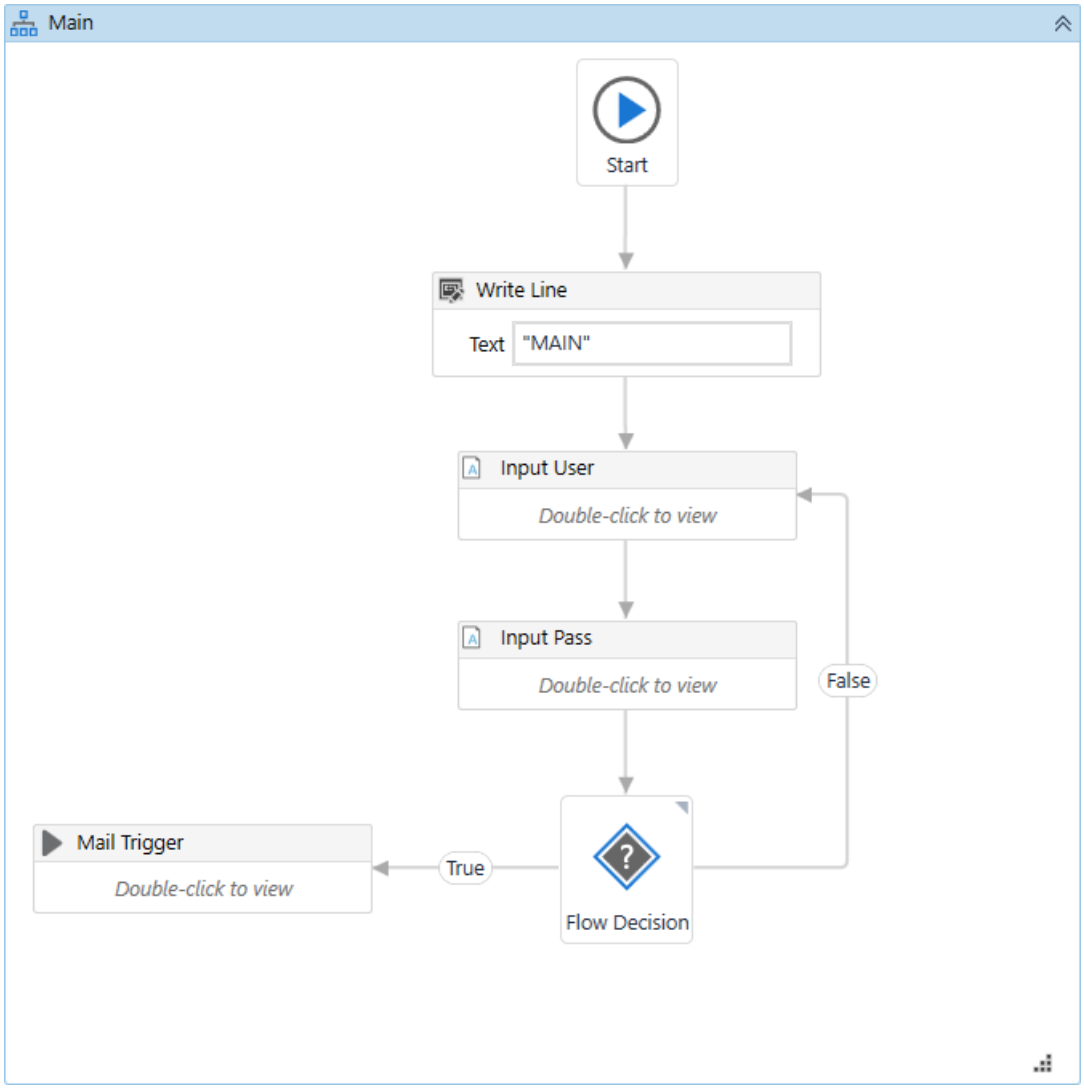
```
import nltk, re, heapq, string, gensim

from gensim.summarization import summarize
from gensim.summarization import keywords
```

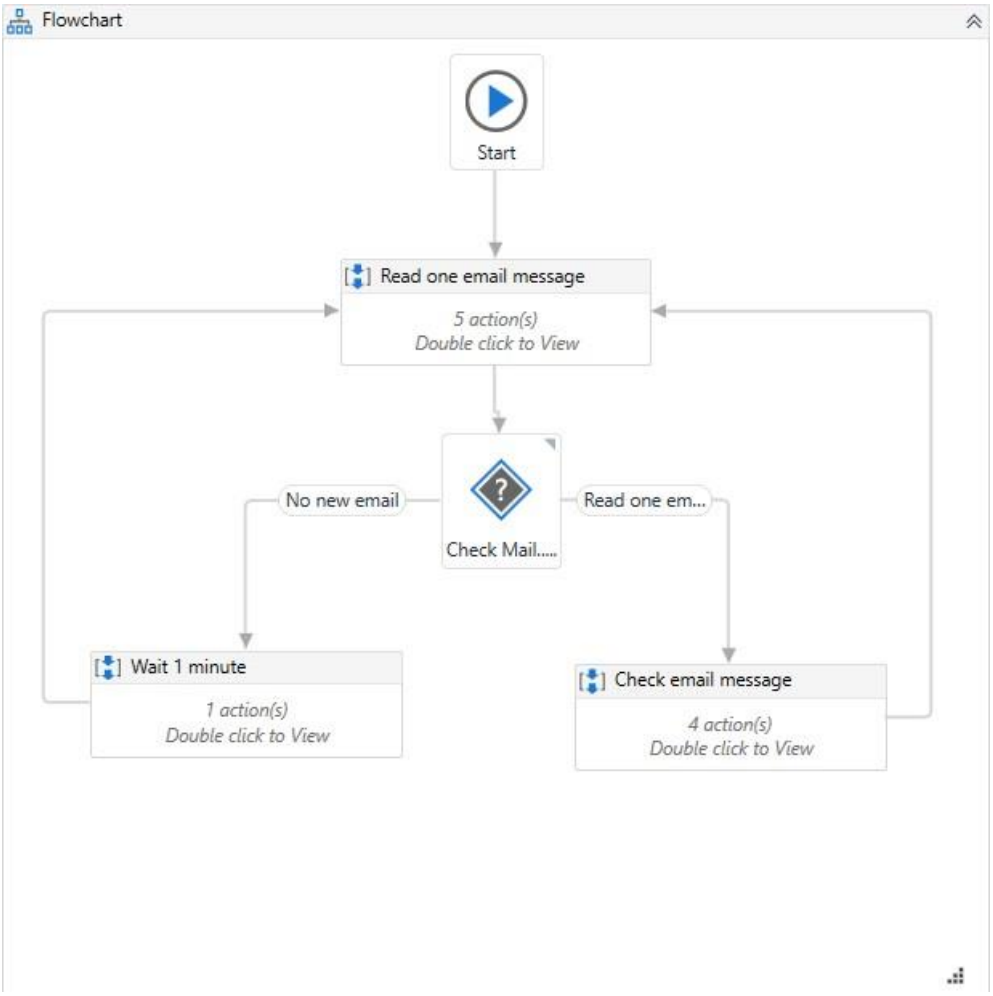
```
import nltk, re, heapq
text = '''"Hi I am Sam, Recently I bought a new Air Quality sensor module from you! Suddenly GPRS of the module is not working properly!



article_text = re.sub(r'\s+', ' ', text) # remove unwanted space....

# print("After remove brakets and space : \n\n", article_text, "\n")

formatted_article_text = re.sub(r'\s+', ' ', article_text)

# print("After remove special char : \n\n", article_text, "\n")

# token the sentence....
sentence_list = nltk.sent_tokenize(article_text)

print("Sentence in the paragraph : \n\n", sentence_list, "\n")

# Word frequency....

stopwords = nltk.corpus.stopwords.words('english')

word_frequencies = {}
for word in nltk.word_tokenize(formatted_article_text):
        if word not in stopwords:
                if word not in word_frequencies.keys():
                        word_frequencies[word] = 1
                else:
                        word_frequencies[word] += 1
```

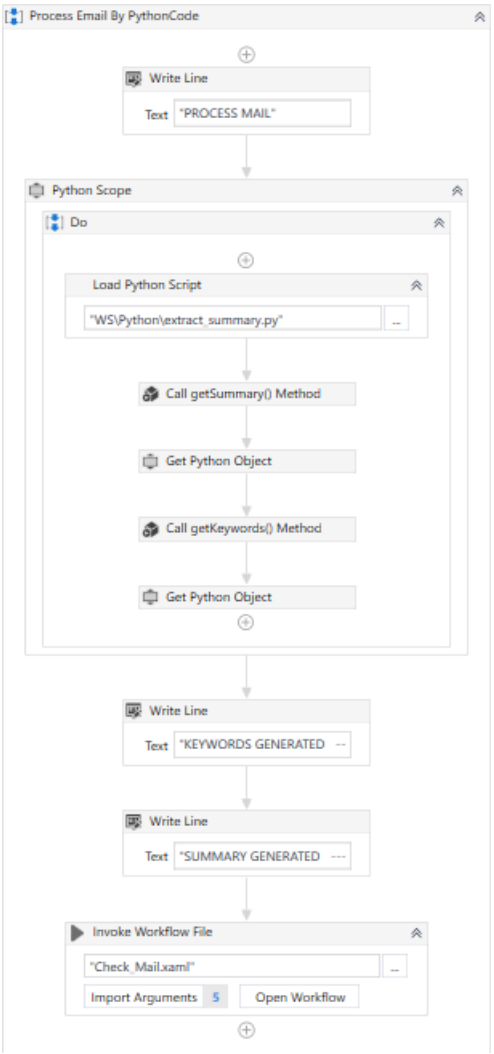
```
    summary_sentences = heapq.nlargest(sentence_limit+1, sentence_scores, key=sentence_scores.get)

    summary = ' '.join(summary_sentences)
    # print("Input Text : ",  text, "\n\n")

    print("Final Summary : ", summary)

    dsc = sorted(sentence_scores.items(),reverse=False)
    print(dsc)
```

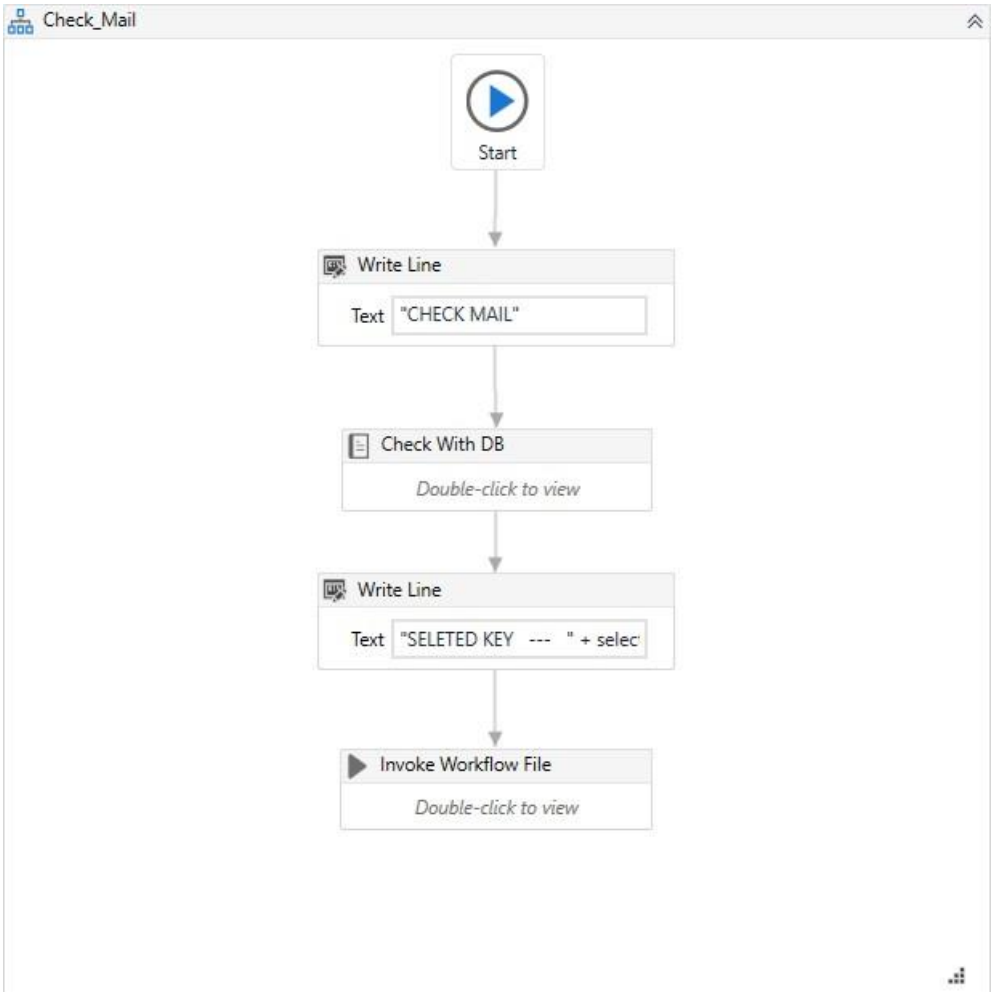
```
    # max of frequency
    maximum_frequency = max(word_frequencies.values())
    # print(maximum_frequency)
    for word in word_frequencies.keys():
        word_frequencies[word] = (word_frequencies[word]/maximum_frequency)
```

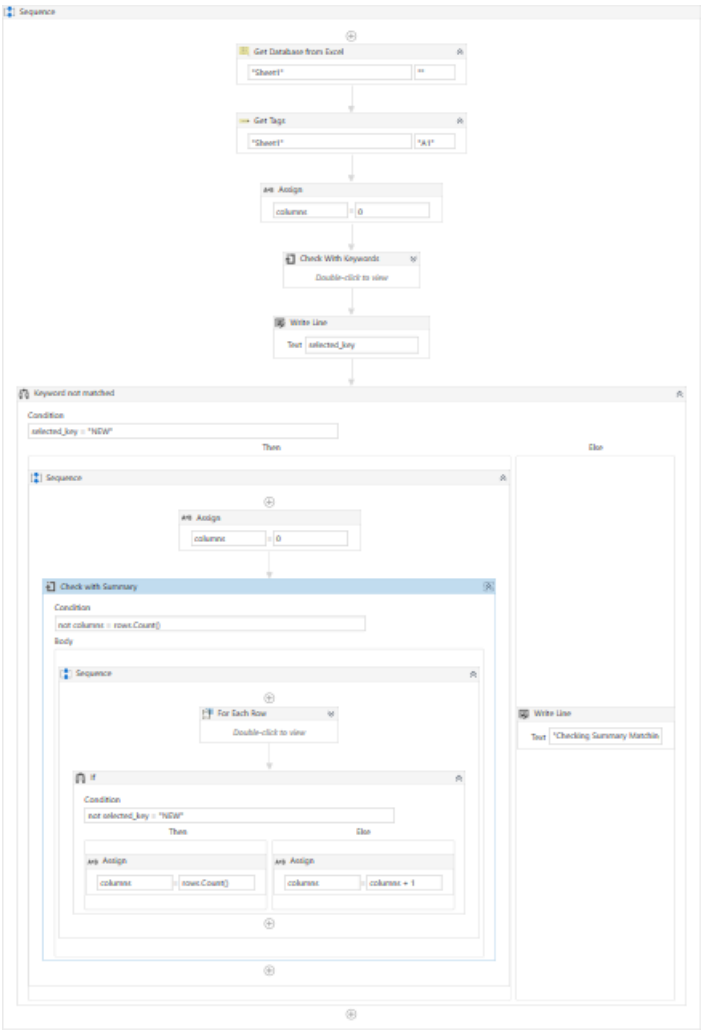
10.2 SCREENSHOT

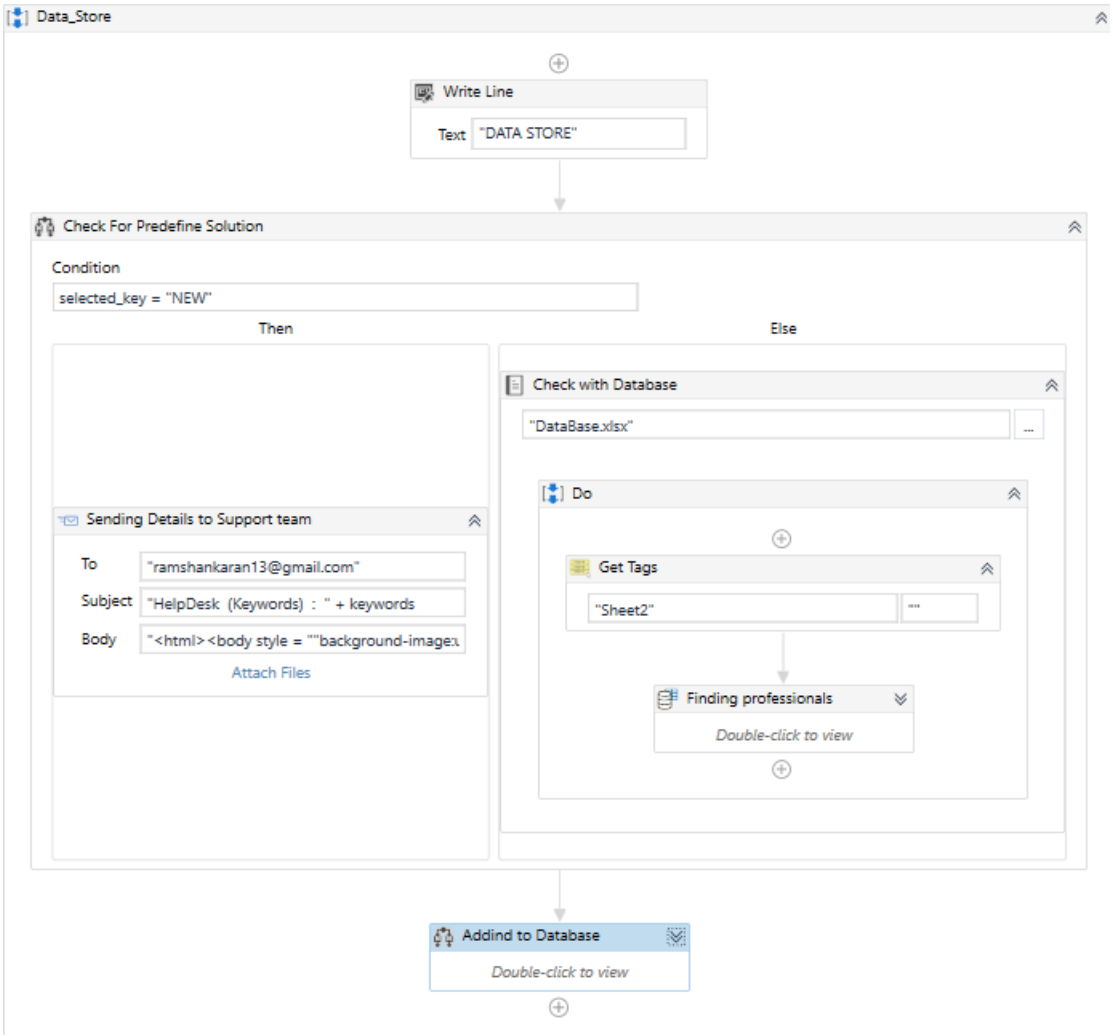


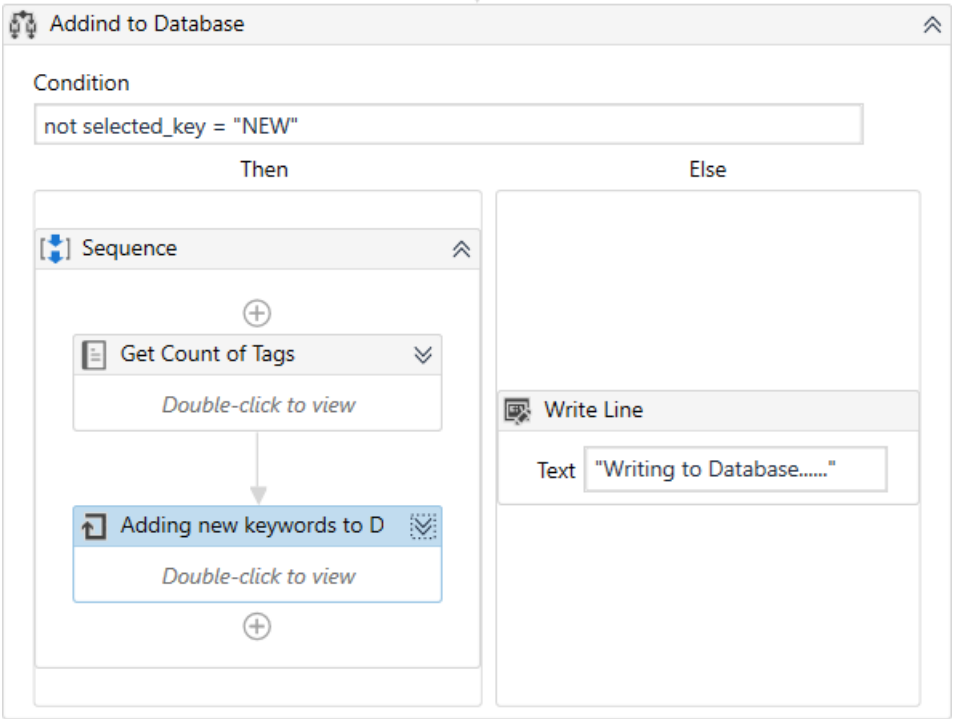

Flowchart
Start
Read one email message
5 action(s)
Double click to View
No new email
Check Mail.....
Read one em...
Wait 1 minute
1 action(s)
Double click to View
Check email message
4 action(s)
Double click to View


Process Email By PythonCode
Write Line
Text "PROCESS MAIL"
Python Scope
Do
Load Python Script
"WS\Python\extract_summary.py"
Call getSummary() Method
Get Python Object
Call getKeywords() Method
Get Python Object
Write Line
Text "KEYWORDS GENERATED
Write Line
Text "SUMMARY GENERATED
Invoke Workflow File
"Check_Mail.xaml"
Import Arguments 5
Open Workflow


Check_Mail
Start
Write Line
Text "CHECK MAIL"
Check With DB
Double-click to view
Write Line
Text "SELETED KEY --- " + selec
Invoke Workflow File
Double-click to view


Sequence
Get Database from Excel
"Sheet1"
""
Get Tags
"Sheet1"
"A1"
Assign
columnx
0
Check With Keywords
Double-click to view
Write Line
Text
selected_key
Keyword not matched
Condition
selected_key = "NEW"
Then
Else
Sequence
Assign
columnx
0
Check with Summary
Condition
not columnx = rows.Count()
Body
Sequence
For Each Row
Double-click to view
If
Condition
not selected_key = "NEW"
Then
Else
Assign
columnx
rows.Count()
Assign
columnx
columnx + 1
Write Line
Text
"Checking Summary Matchin


Data_Store
Write Line
Text "DATA STORE"
Check For Predefine Solution
Condition
selected_key = "NEW"
Then
Else
Check with Database
"DataBase.xlsx"
Do
Get Tags
"Sheet2"
Finding professionals
Double-click to view
Sending Details to Support team
To "ramshankaran13@gmail.com"
Subject "HelpDesk (Keywords) : " + keywords
Body "<html><body style = ""background-image:
Attach Files
Addind to Database
Double-click to view


Addind to Database
Condition
not selected_key = "NEW"
Then
Else
Sequence
Get Count of Tags
Double-click to view
Adding new keywords to D
Double-click to view
Write Line
Text "Writing to Database....."

10.3 LIST OF PUBLICATION

1. Sanjay R B, Sam Dany R, Ramshankaran Y, Prathvik S, Jayashree R "TICKET AUTOMATION SYSTEM USING RPA", in International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), Volume 2, Issue 9,September 2020

TICKET AUTOMATION SYSTEM USING RPA

1. SANJAY.R. B, KGiSL Institute of Technology,Coimbatore.
2. SAM DANY.R, KGiSL Institute of Technology,Coimbatore.
3. RAMSHANKARAN.Y, KGiSL Institute of Technology, Coimbatore.
4. PRATHVIK.S, KGiSL Institute of Technology, Coimbatore.

ABSRTACT

Under the acceleration of the wheel of technology around the world, various ways of automating information have occurred to replace old-fashioned ways that don't serve the pace of everyday lifestyle. There are many things that we should change in our work systems such as paper, stand-alone systems, phone calls or even email. Every kind of companies has different sections that help the process of paperwork concerning personnel and maintenance. However, some of this paperwork might be overlooked or delayed, which could lead to slow down the outcome of the organization. So, they need to improve their system work to save time, effort and make an efficient mechanism to follow requests and tickets of any organization or customer. So, we should replace any old system with RPA to facilitate the work of supporting and following tickets. As a solution to that, we have come with the idea of the " Ticket Automation System Using RPA." The application aim is to eliminate manual intervention in ticket creation and it raise a ticket based on the complaint mail. If the details are incomplete (e.g. customer id is missing), send a mail to customer asking for missing details and Link the subsequent responses from the customer to the original ticket. Recognize the bounced mails and initiate appropriate action via sending auto response to template-based mails (complaints/queries) i.e. no free text. It is a modern way to solve problems.

KEYWORDS

NLP, Gensim, Heapq, Uipath.

1 INTRODUCTION

The ticket automation is playing a huge role in maintaining a successful support operation and is helping to revolutionize the industry. The days of the “gatekeeper” that directed customer support issues to the right people are over. Now, with customer support software and ticket automation, triggers can be created so if a ticket is submitted via email or a customer portal it goes to a specific agent or group. For example, a trigger where the subject line contains the word “how” (i.e. how do I add a user) could automatically route all emails to the training group. And it also Remind customers to follow up, Close tickets automatically, Get alerts for urgent tickets, Weekends and holidays

2 LITERATURE REVIEW

Management and maintenance of IT infrastructure resources such as hardware, software and network is an integral part of software development and maintenance projects. Service management ensures that the tickets submitted by users, i.e. software developers, are serviced within the agreed resolution times. Failure to meet those times induces penalty on the service provider. To prevent a spurious penalty on the service provider, nonworking hours such as waiting for user inputs are not included in the measured resolution time, that is, a service level clock pauses its timing. Nevertheless, the user interactions slow down the resolution process, that is, add to user experienced resolution time and degrade user experience. Therefore, this work is motivated by the need to analyse and reduce user input requests in tickets’ life cycle. To address this problem, we analyse user input requests and investigate their impact on user experienced resolution time. We

distinguish between input requests of two types: real, seeking information from the user to process the ticket and tactical, when no information is asked but the user input request is raised merely to pause the service level clock. Next, we propose a system that pre-empts a user at the time of ticket submission to provide additional information that the analyst, a person responsible for servicing the ticket, is likely to ask, thus reducing real user input requests. Further, we propose a detection system to identify tactical user input requests. To evaluate the approach, we conducted a case study in a large global IT company. We observed that around 57% of the tickets have user input requests in the life cycle, causing user experienced resolution time to be almost twice as long as the measured service resolution time. The proposed pre-emptive system pre-empts the information needs with an average accuracy of 94–99% across five cross validations while traditional approaches such as logistic regression and naive Bayes have accuracy in the range of 50–60%. The detection system identifies around 15% of the total user input requests as tactical. Therefore, the proposed solution can efficiently bring down the number of user input requests and, hence, improve the user-experienced resolution time.

3 EXISTING SYSTEM

In existing system Man power is required to assign a unique ticket. It also saves the solved solutions in a database in order to solve when the same queries arise. So typically, there will be two teams

1) Support team
2) Technical support team

Tickets will be analysed and redirected to the technical support team by the support team.

Customers can call only to the support team and further procedures will be done by the support team

3.1 DISADVANTAGES

Since the system needs manpower it requires funding for those people. And if they seem to do OT then an extra funding should be arranged to fulfil their needs. It also includes the Computer usage and energy consumption according to the manpower. In general humans cannot work continuously so they need to provide the working timings to the customers and this may lead a customer to avoid our products. In Business, customer satisfaction plays a important role in order to promote the product so their interest is more important than our hardships.

In some organizations they have also installed a third-party Ticket automation software. So that system analyses the queries raised by the customers and it assigns the ticket and redirects to the technical support team. But this software is best suited for large scale industries and it also grabs a fund from the organisation.

3.2 PROPOSED SYSTEM

The proposed system is RPA bot which uses NLP algorithm to solve the user's queries. This system should be able to analyse and understand user's queries and react accordingly. For any product related queries, we have put a mail to customer support team and it will be forwarded to the particular domain members in order to provide the solution. Frequently solved queries will be stored in the database to provide solutions when the similar queries raised by the user. Without this system an organisation needs a team for assigning the tokens to the domain related persons.

Our system is built using NLP algorithms that analyses user's queries and understand user's message. The system is designed for support team in all

product organisations where users can ask any product related questions. The system recognizes user's query and understands what he wants to convey and simultaneously answers them appropriately. The NLP algorithm analyses users queries then convert into tokens. Then the tokens are assigns to the persons in the particular domain. The main workflow of our system

- Get the complaint mails from the users.
- Segregate the mail, understands the user queries and assign the ticket to the particular complain mail.
- If the queries already solved, then replay with the existing solutionand close the ticket.
- If the problem is not solved then perform the analysis on the mail and raise the ticket to appropriate person.

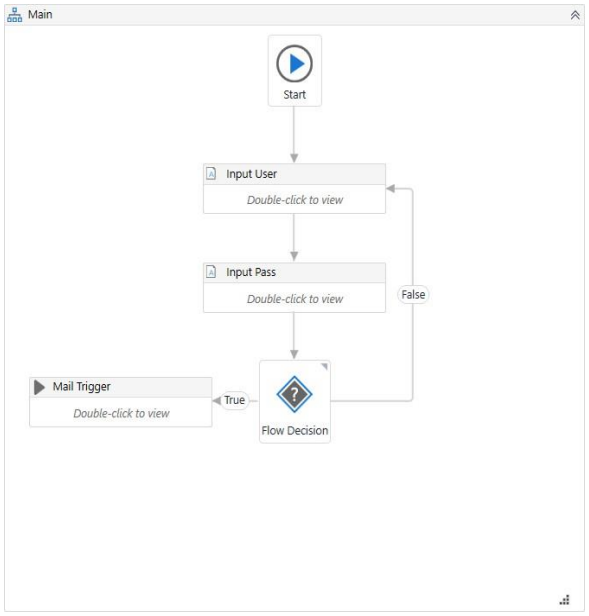
3.3 ADVANTAGES

Since our system is built using Uipath it can be bought by any organization that can be a small scale or large scale and they can be utilized as long as they need. For updates, it requires only one technical person and this system is capable to run in as many systems in parallel. Comparing to other software, it is one of the cheapest software in the market and it won't require any services at a higher level. It uses excel as the backend, it's also a user-friendly application.

4 RESULTS

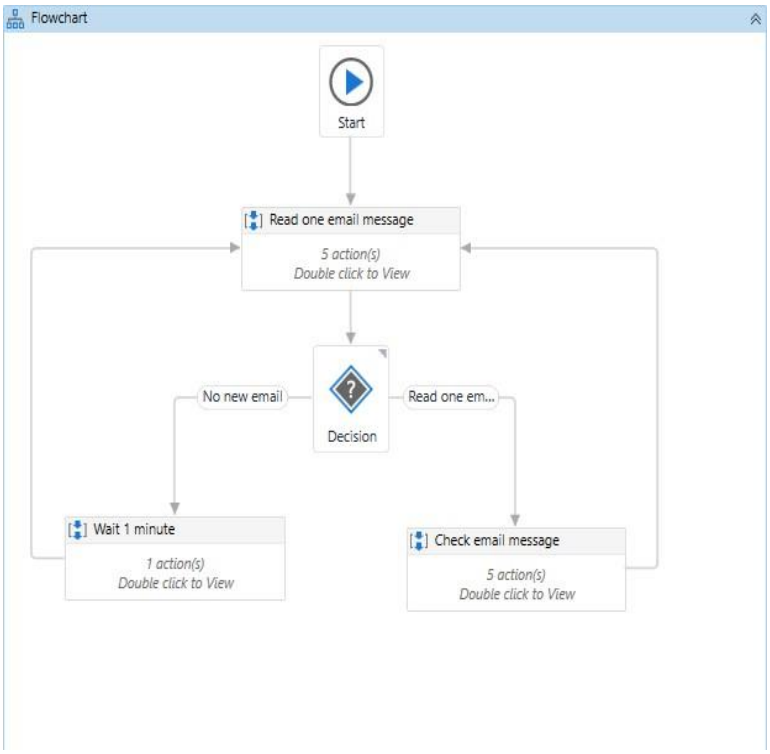
This system is been implemented as follows:

Initially the bot starts its work by getting credentials from the user and if he/she is a valid user the he/she will be redirected to the mail trigger module



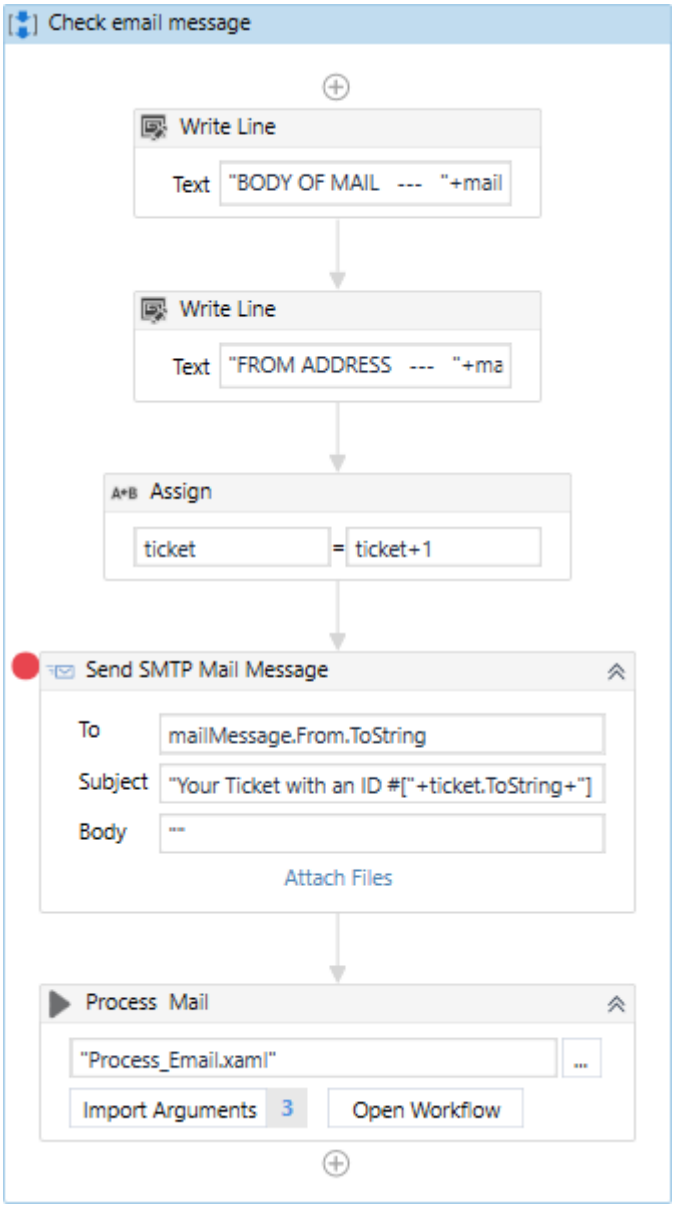
4.1 Mail trigger

In the Mail trigger module the real work starts this mail trigger will be running at the background synchronously to get the mail from the organisation mail inbox if there is no email it will wait for 1 minute else if a new mail comes it will redirect it to the check mail module.



4.2 Check Email Message

In this module the mail will be analysed and extracts the text and sends a response email to the respective customer after sending the response mail it will be redirected to the process mail module.



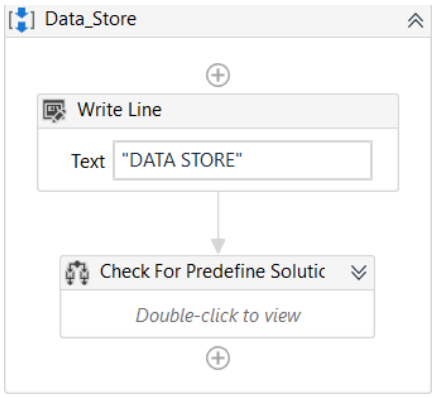
4.3 Process mail

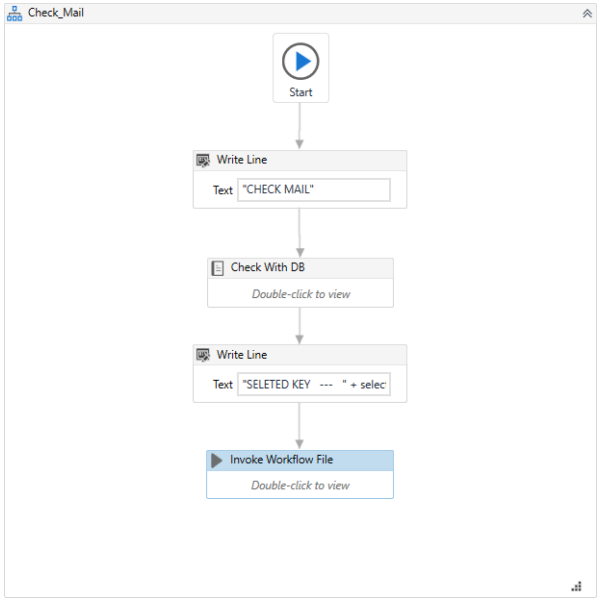
This module initially loads the obtained text information's from the previous module the python script which uses NLP and Gensim in order to make the unstructured data to structured by removing space, special characters.

Then call getsummary() method gets the summary and getkeywords() method gets the keywords and after generating the summary it will be redirected to the check mail module.

4.4 Check Mail

This module checks with Datastore module whether this problem is solved already or not if it exists it replies with that solution else it raises a ticket to the technical support team.

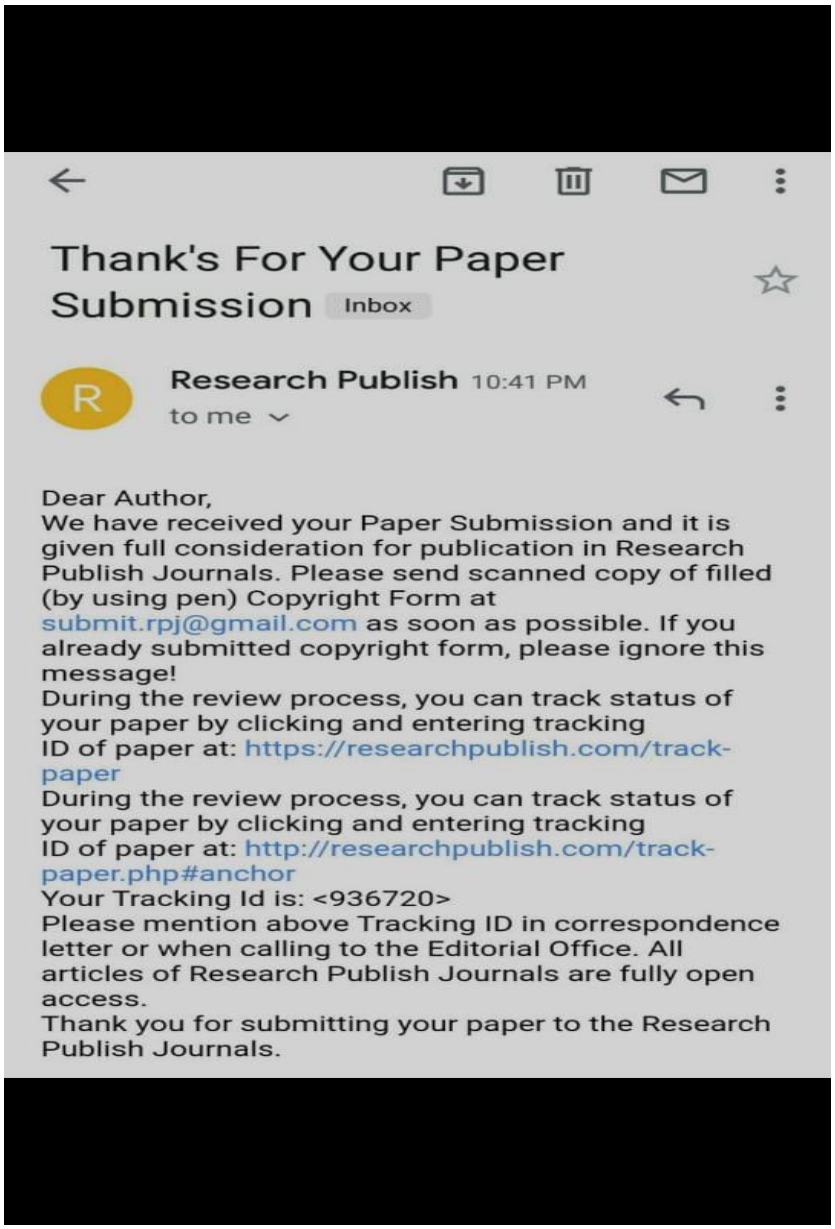




4.5 Database Design:

Billing	Payment	Services	Wireless
not receiving bills	unable to pay online	not able to use the internet	slow speed
temporary disconnection	payment done but service not restored	slow speed	connection issue
incorrect billing	payment done but not updated	router configuration	device not reachable
charged daily	payment	reg blinking	
separate connection	online pay	los light	
	pay	connection issue	
	pay online	connection	
	money	pon blinking	
	credited	inside shifting	
		outside shifting	
		interruption	
		los light gets blinking	
		service person	
		disconnection	
		router	

TAGS	Emails
Billing	sanjaybabu618@gmail.com
Payment	samdany454@gmail.com
Services	prathviksambath@gmail.com
Wireless	ramshankaran13@gmail.com



Thank's For Your Paper Submission Inbox

Research Publish 10:41 PM
to me

Dear Author,
We have received your Paper Submission and it is given full consideration for publication in Research Publish Journals. Please send scanned copy of filled (by using pen) Copyright Form at submit.rpj@gmail.com as soon as possible. If you already submitted copyright form, please ignore this message!
During the review process, you can track status of your paper by clicking and entering tracking ID of paper at: https://researchpublish.com/track-paper
During the review process, you can track status of your paper by clicking and entering tracking ID of paper at: http://researchpublish.com/track-paper.php#anchor
Your Tracking Id is: <936720>
Please mention above Tracking ID in correspondence letter or when calling to the Editorial Office. All articles of Research Publish Journals are fully open access.
Thank you for submitting your paper to the Research Publish Journals.

CHAPTER 11

REFERENCES

[1] S. Agarwal, R. Sindhgatta, B. Sengupta **SmartDispatch: enabling efficient ticket dispatch in an IT service environment** Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery anddata mining (2012), pp. 1393-1401

[2] C. Aggarwal, C. Zhai **A survey of text classification algorithms** Mining Text Data, Springer, Boston, MA (2012), pp. 163-222

[3] A. Cater-Steel, R. Valverde, A. Shrestha, M. Toleman **Decision support systems for IT service management** Int. J. Inf. Decision Sci. (2016)

[4] S. Foo, S.C. Hui, P.C. Leong, S. Liu **An integrated help desk support for customer services over the World Wide Web – a case study** Comput. Ind., 41 (2000), pp. 129-145

[5] M. Jäntti **Examining challenges in IT service desk system and processes: a case study** In the Seventh International Conference on Systems (ICONS) (2012), pp. 105108

[6] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov **Bag of tricks for efficient text classification** Short Papers Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (2017), pp. 427-431

[7] X. Tang, Y. Todo **A study of service desk setup in implementing IT service management in enterprises** Technol. Investment, 4 (3) (2013), pp. 190-196

[8] A. Tanovic, N.E. Mastorakis **Advantage of using service desk management systems in real organizations** Int. J. Econ. Manage. Syst., 1 (2016), pp. 81-86

[9] X. Zhu, W. Zhou, T. Li **A visual tool for ticket monitoring** 2017 IEEE 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE) (2017), p. 107