

Problem Statement 1 (Automation for DSD) – Solution

“Intelligent email autoresponders”

Introduction/ Understanding the Problem statement:

SkyCraper Solution is an audit firm. The employees of SkyCraper Solution are extremely friendly with emails, which is their main mode of communication. The email volume is about 85% of the total contact to Service Desk.

As the service matured, the IT Manager identified that 70% of the issues or request reported to the Service Desk have very simple task and can be automated. He also noticed from the point of reporting the issue to the same getting resolved, it takes more than 5 hrs. He realized that the issue which should have been resolved immediately was **getting delayed resolution as there is human intervention**. This has decreased the satisfaction level of the user base. Also this is waste of human resource at Service Desk who can be skilled to do more meaningful work. **Another challenge he was faced with is that his company was expanding in Non-English speaking countries, while his IT team can understand only English.**

He decided to resolve both the challenges with automation. He created an automation team and tasked them in creating tool called “Intelligent email autoresponders”. He provided them with the following requirements for tool.

“Intelligent email autoresponders” – Requirements

1. Reads the email sent by the user
2. Create the appropriate ticket.
 - a. If an issue is reported it should create an Incident ticket
 - b. If a request is reported it should create a Service Request
3. Resolve the Incident or Service Request which is available in the knowledge base
4. If the issue is unavailable in KB (Knowledge Base), then the ticket needs to be assigned to the correct resolving group
5. Response to be sent to the customer with details of resolution (if issue resolved)
6. If ticket is assigned to a support team response to be sent to customer giving a generic name of the resolving team and approx. time period for resolution
7. The tool should be language independent, meaning it understands any language but creates the ticket in English language only
8. While responding to the user the tool should respond back in the language, in which the user had written the mail.

Details of technology Used:

Language used: Python3

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets us work quickly and integrate systems more efficiently.

Package(s) used:

Selenium

Selenium is a web automation framework that can be used to automate website testing. Because Selenium starts a web browser, it can do any task we would normally do on the web. Supported browsers are:

- Chrome
- Firefox
- Internet Explorer
- Safari
- Opera
- PhantomJS (invisible)

Scrapy

Scrapy is a Python framework for large scale web scraping. It gives all the tools needed to efficiently extract data from websites, process them as you want, and store them in preferred structure and format.

NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in Python. It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. NLTK includes graphical demonstrations and sample data.

Required Software /Hardware:

- Python3
- 2 GB RAM
- 1 GB of Disk Space
- Windows/macOS/Linux
- Web Browser
- Text Editor

Achieved cost saving:

Since 70% of the issues or requests are being automated by the tool “Intelligent mail autoresponders”, a huge cost reduction can be achieved. As most of the issues are being classified and solved by the tool itself, no payment needs to be done to the service desk or the resolving group for those tasks.

Delaying reduces the efficiency of the firm, thus decreasing the profit earned. The tool easily solves this problem as it addresses the issue or request as soon as the e-mail arrives and takes the necessary actions. It solves the issue or forwards it to the correct resolving group without delay thus eliminating the 5-hour delay which was a major concern before. Also, this decreases the wastage of human resource in the firm as the employees at the service desk are assigned their task without any delay or they can be assigned some more meaning work according to their skills while minor tasks are resolved by the tool.

The tool will also increase the user’s satisfaction level as they will receive a response from the firm in a few minutes after submitting their requests. The tool will mail them back the details of resolution as soon as it resolves the issue or the expected time it will take to resolve the issue if the tool is not able to resolve the issue by itself. This will, in turn, increase the productivity of the employees and thus decreasing the losses incurred due to the employees not being able to continue their work because of their issues.

At last, since the tool is language independent, the money that was spent hiring employees that know multiple languages or training the employees to understand multiple languages is saved. This achieves great cost saving for the firm.

Architecture:

Upon reading the problem statement and based on the requirements the system functionality can be divided into use cases and actors as seen below.

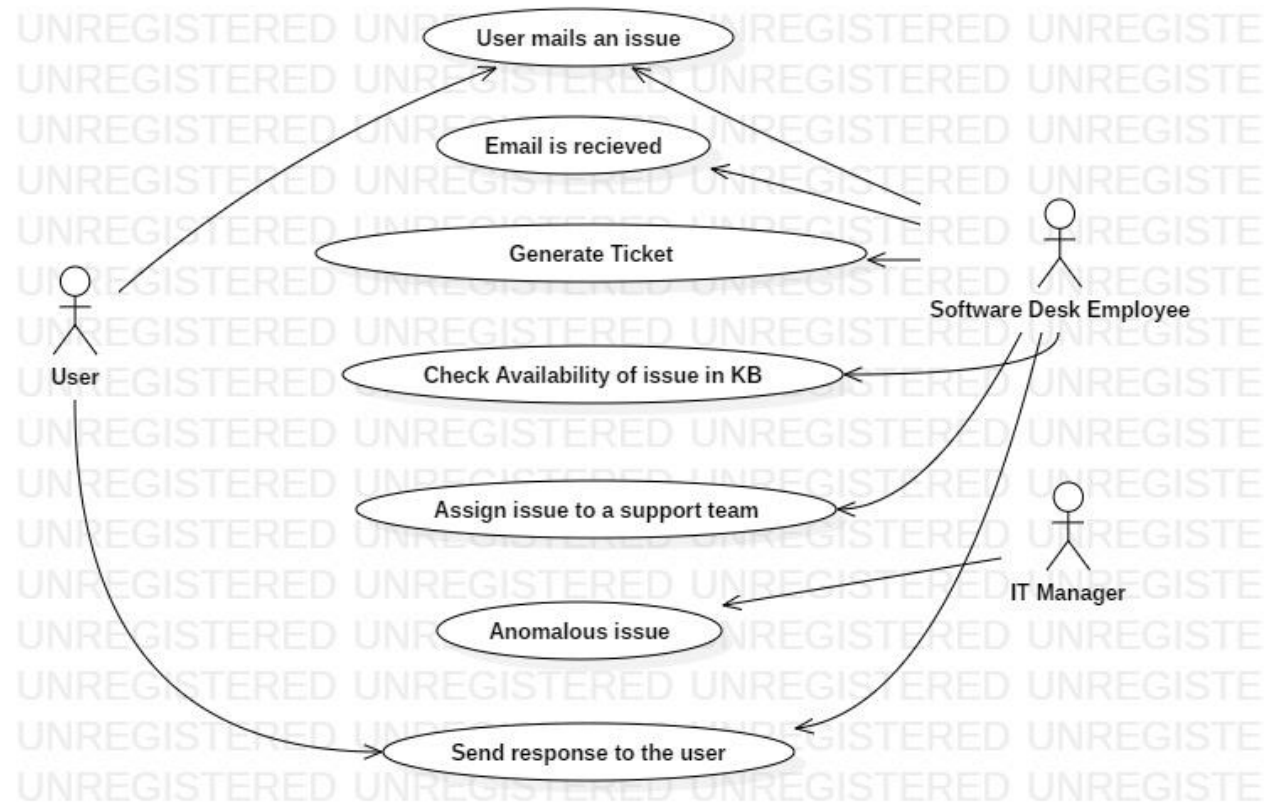


Fig-1. Use Case Diagram - Representing general use cases and actors involved in the system

After the email is received by the Service Desk, the email needs to be translated into English language (if it is not). Then an appropriate ticket is to be generated as follows:

- a. If an issue is reported it should create an Incident ticket
- b. If a request is reported it should create a Service Request

After the generation of the ticket the Service Desk will check the availability of the issue in the knowledge base. If the issue is unavailable in KB (Knowledge Base), then the ticket needs to be assigned to the correct resolving group. Next response is sent to the customer with details of resolution (if issue resolved). If ticket is assigned to a support team response to be sent to customer giving a generic name of the resolving team and approximate time period for resolution. All responses sent to users must be in the same language in which the user has written the email.

The following activities and architecture of the automation tool is illustrated by the following activity diagram:

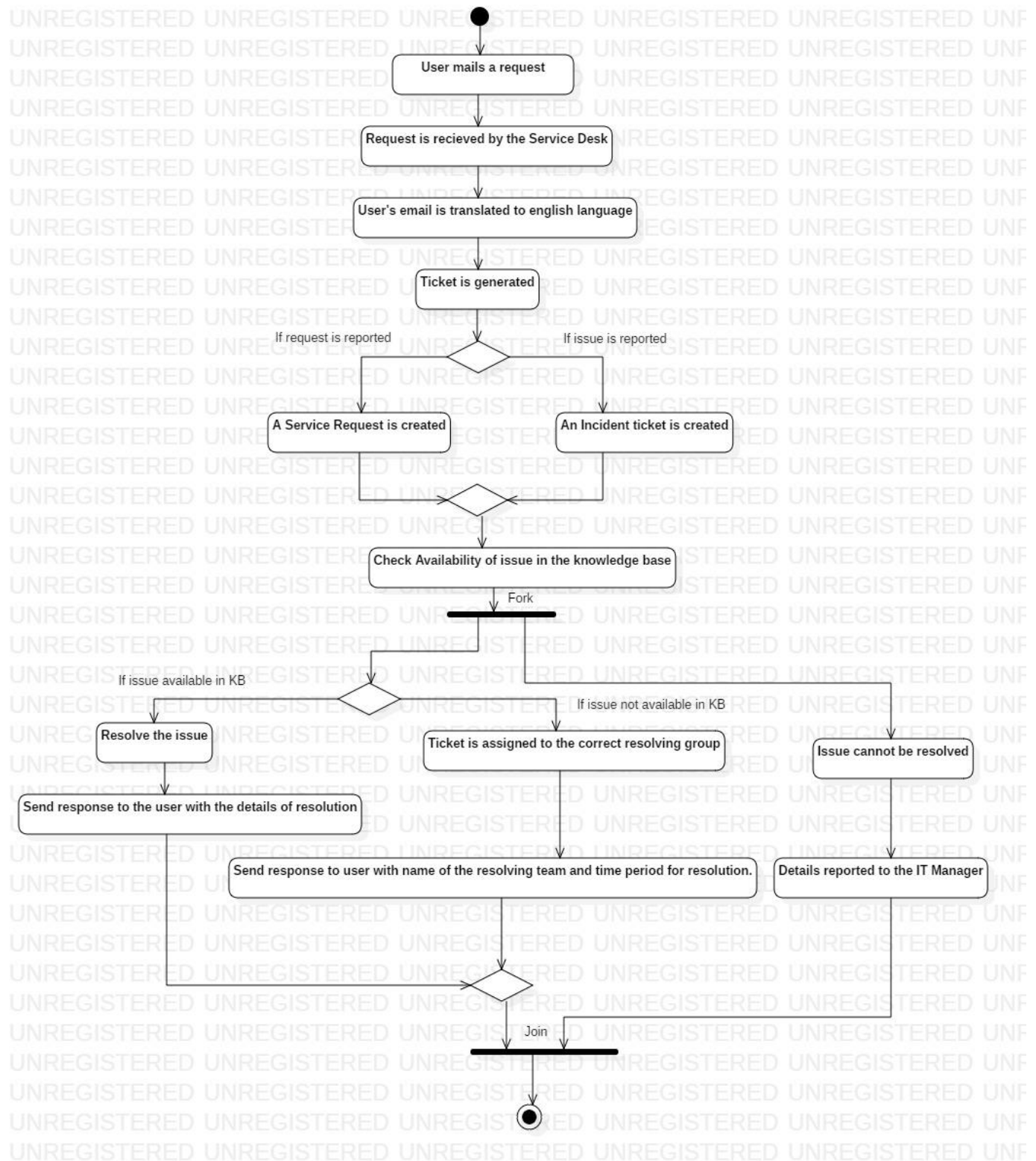


Fig-2. Architecture of the solution

Solution Brief Description:

Using the python packages selenium and scrapy the email of the user is extracted and converted to the English language for the service desk to understand using Google APIs. Then based on the specific keywords (using natural language processing and decision trees) it is decided whether the email is an issue or a request and accordingly an Incident ticket or a Service Request is generated.

The user's email is processed using natural language processing using nltk package in python. Then a query is executed to the knowledge base with the specific keywords from the email of the user. If there is a match then the issue is available in the knowledge base and is resolved. If there is no match then there is a search for the correct resolving group based on the number of resolving group available.

The response is again translated to the user's language and emailed to him/her once the issue is resolved. If the ticket is assigned to a support team response to be sent to customer giving a generic name of the resolving team and approximate time period for resolution.

Scope of Automation:

The automation tool that will be created is a python script and its scope is only limited to web browsers, that is, the script will perform all its operations through web browser since selenium which is a web driver.

This automation can be used where the tasks are monotonous and repetitive. A task that involves a large amount of labor can also be easily tackled using automation.

Conclusion:

The tool that will be developed will surely enhance the efficiency of the employees and improve productivity. More profit and loss reduction can be surely achieved. The tool will also help employees to work without much obstruction and will increase their satisfaction level. The amount of time saved by using the above-mentioned tool is a great advantage too. The utility of this tool can be enhanced by making small changes and adding additional features if needed. The tool can be used in other areas too like banking, telecommunications, etc.