

RPA Based Automation To Reduce Human Intervention In Data Extraction Process

by

Afia Anjum
18101239

An internship report submitted to the Department of Computer Science and
Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
May 2022

© 2022. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The internship report submitted is my own original work while completing degree at Brac University.
2. The internship report does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The internship report does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

Afia Anjum

Afia Anjum

18101239

Approval

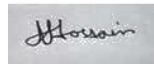
The internship report titled “RPA Based Automation To Reduce Human Intervention In Data Extraction Process” submitted by

1. Afia Anjum (18101239)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 26, 2022.

Examining Committee:

Supervisor:
(Member)



Syed Hasan Ibne Hossain
GM, Platform Management
Digital Service, IoT & VAS , IT
Robi Axiata Limited

Program Coordinator:
(Member)



Ms. Nadia Rubaiyat
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In this era of modern development, everything is controlled by technology. We can not even think our life without smart devices. Robots, AI, bots, and other similar technologies are being used to replace every minor task. To survive in this competitive modern era, people want to make their work more efficient and secure. The work that was done manually is now replaced by automated programs and bots. People do not want to waste their time on repetitive work, rather big industries prefer to handle these works with automation.

I am glad that I worked on some amazing automation projects which helped Robi to replace their repetitive manual work with my automated programs. It saved their time and made the work more efficient. On the other hand, I was able to learn new concepts like RPA, docker, git. It was not easy to learn these new concepts all at once. I faced a lot of challenges. But to overcome those I had to do a lot of researches which helped to increase my problem solving skills.

I only had basic programming knowledge on JAVA and Python. But in this internship journey I got chance to use these knowledge in extensive works. At first I faced many challenges because automation was a new field to me. But with the proper support from my instructors helped me to accept this challenge.

Learnings become more satisfactory when you can solve the real life challenges with your work and critical thinking. I gained this ability during this internship journey. I recommend that every student should apply for internship so that they can get a real taste of professional life and develop their problem solving skills.

Acknowledgement

Firstly, all praise to Almighty Allah for whom I was able to complete this internship without any difficulties.

Secondly, my Line Manager Syed Hasan Ibne Hossain and my instructors Sakibul Islam and S.M. Yeaminul Islam for guiding me throughout this journey.

Thirdly, Nadia Rubaiyat miss for always being there to solve any kind of problems and taking updates of my work.

Finally, my parents who always encouraged and supported me throughout the journey.

Table of Contents

| | |
|--|------------|
| Declaration | i |
| Approval | ii |
| Abstract | iii |
| Acknowledgment | iv |
| Table of Contents | v |
| List of Figures | vii |
| 1 Introduction | 2 |
| 1.1 About Internship | 2 |
| 1.2 About this Report | 3 |
| 1.3 Objective | 3 |
| 1.4 Methodology | 3 |
| 2 Company Profile | 4 |
| 2.1 Background of the Company | 4 |
| 2.2 Objective of the Company | 5 |
| 2.3 Mission | 5 |
| 2.4 Vision | 5 |
| 2.5 Departments of the Company | 5 |
| 3 Training Phase | 6 |
| 3.1 Overview | 6 |
| 3.2 Getting Familiar with the Workplace | 6 |
| 3.3 Communication and Meetings | 6 |
| 3.4 Resources and Study Material | 7 |
| 3.5 VAS and Microservice | 7 |
| 4 My Contributions | 8 |
| 4.1 Report Automation with Data Visualization Project | 8 |
| 4.1.1 Software Installation | 8 |
| 4.1.2 Database Creation | 8 |
| 4.1.3 Implementation | 11 |
| 4.1.4 Service Diagram | 12 |
| 4.2 Traffic Log File Parsing in txt format and CSV Report Generation . . | 12 |
| 4.2.1 Output | 13 |

| | | |
|----------|---|-----------|
| 4.2.2 | Service Diagram | 14 |
| 4.3 | Multiple txt format traffic log files consolidated into single csv report | 14 |
| 4.3.1 | Implementation | 14 |
| 4.3.2 | Output | 16 |
| 4.3.3 | Service Diagram | 16 |
| 4.4 | Generalized CSV Reports Email Broadcaster | 17 |
| 4.4.1 | Implementation | 17 |
| 4.4.2 | Output | 18 |
| 4.4.3 | Service Diagram | 19 |
| 4.5 | RPA: Extracting CSV Reports by web scraping in HTML webpage . | 19 |
| 4.5.1 | Implementation | 19 |
| 4.5.2 | Output | 21 |
| 4.5.3 | Service Diagram | 21 |
| 4.6 | Pending e-Approval Report Parsing in CSV Format and Consolidated Report Generation | 22 |
| 4.6.1 | Implementation and Output | 22 |
| 4.6.2 | Service Diagram | 26 |
| 4.7 | Outlook Email Parsing Automation | 26 |
| 4.7.1 | Implementation | 26 |
| 4.7.2 | Output | 28 |
| 4.7.3 | Service Diagram | 29 |
| 4.8 | Data Extraction from Complaint File | 30 |
| 4.8.1 | Implementation | 30 |
| 5 | Learnings from the Internship | 33 |
| 5.1 | Professional Growth | 33 |
| 5.2 | Team Work | 33 |
| 5.3 | Time management | 33 |
| 5.4 | Technical Growth | 33 |
| 5.4.1 | RPA | 33 |
| 5.4.2 | Git | 34 |
| 5.4.3 | Docker | 34 |
| 5.4.4 | Documentation | 34 |
| 6 | Conclusion | 35 |
| 6.1 | Conclusion | 35 |
| 6.2 | Recommendations | 35 |
| | Bibliography | 36 |

List of Figures

| | | |
|------|---|----|
| 2.1 | Logo | 4 |
| 4.1 | Employee table | 9 |
| 4.2 | Department table | 9 |
| 4.3 | Attendance table | 10 |
| 4.4 | Salary table | 10 |
| 4.5 | Screenshots of automated generated email | 11 |
| 4.6 | Screenshots of automated generated email | 11 |
| 4.7 | Service Diagram | 12 |
| 4.8 | Screenshot of the log file. | 13 |
| 4.9 | Generated .csv file. | 13 |
| 4.10 | Service Diagram. | 14 |
| 4.11 | Code Snippet of python script | 15 |
| 4.12 | Log File in .txt format generated from server 1(Total 3456 data). | 15 |
| 4.13 | Consolidated output from four log files. | 16 |
| 4.14 | Service Diagram | 16 |
| 4.15 | Code Snippet of the python script | 17 |
| 4.16 | Code Snippet of the python script | 18 |
| 4.17 | Output of Email Notification | 18 |
| 4.18 | Service Diagram | 19 |
| 4.19 | Sign In page of e-approval website | 20 |
| 4.20 | Screenshot of the e-Approval Website | 20 |
| 4.21 | Screenshot of the generated csv file | 21 |
| 4.22 | Service Diagram | 21 |
| 4.23 | Screenshot of the codebase | 22 |
| 4.24 | Screenshot of the loaded website | 23 |
| 4.25 | Web bot clicks on this Group Task link | 23 |
| 4.26 | Executor page | 24 |
| 4.27 | Output of the generated pending e-Approval details | 24 |
| 4.28 | Output of generated pending e-Approval Count | 25 |
| 4.29 | Service Diagram | 26 |
| 4.30 | Code Snippet | 27 |
| 4.31 | Code Snippet | 27 |
| 4.32 | Screenshot of the 'Inbox' folder | 28 |
| 4.33 | Screenshot of 'Bot_Emails' folder | 28 |
| 4.34 | Screenshot of the 'Sent Items' folder | 29 |
| 4.35 | Service Diagram | 29 |
| 4.36 | Code Snippet | 30 |

| | | |
|------|--|----|
| 4.37 | Excel file read as Dataframe | 31 |
| 4.38 | Raw Complaint File | 31 |
| 4.39 | Complaint File after populating it with USSD Code and Date | 32 |

Chapter 1

Introduction

1.1 About Internship

Internship is an opportunity to gain experience of the working environment and relate the education life with a practical life. It opens a door for fresh graduates and also undergraduates to gather real life experience. Interns are able to get trained on solving real world problems, gain knowledge about the working style of a company. Though it is a short term learning phase, it helps to develop both hard and soft skills. For example, starting with writing resumes, giving interviews, doing daily meetings, sharing ideas and assisting seniors are some valuable lifelong learnings. Sometimes it is difficult to find out from where we should begin our journey in terms of job searching. Here comes the internship to guide you through the professional life and helps to find out the right career path.

To do an internship from the CSE department is not a common thing to see in most of the universities. But BRAC University is providing the opportunity to do an internship instead of thesis. The university provides proper support and resources to help students to pursue their research work in their interested fields. Students are open to choose their desired company and have to complete at least 72 credits to start the internship program. The duration of this internship program has to be at least 6 months which is quite a long period to gain experience and gather adequate knowledge about corporate life.

As I want to pursue my career in the IT field of a corporate company, I thought an internship will be a great chance to get an idea about it. So I applied at Robi Axiata Limited which is the 2nd largest mobile network operator in Bangladesh. I was called for an interview and got selected for the IT division to work in VAS, DATA, IoT and Digital service projects. My responsibility was to develop scalable applications by analyzing business requirements and to automate manual processes using RPA to reduce repetitive activities. Though it was a virtual internship, but I got adequate support and guidance from my instructors which reflects how professional they are to train their interns.

1.2 About this Report

This internship is a part of my academic curriculum. To complete my graduation I have to do an internship of 6 months and show my working progress in this internship report. This report has these following purposes:

- Detail idea about the company and my job responsibilities.
- My working experience with the team.
- Relating the learning from BRACU with my work.
- Learnings from this internship.

1.3 Objective

The main purpose of this report is to reflect my real life work experience at Robi Axiata Limited as an intern. This will be a documentation of my journey and will highlight all the necessary information about this internship experience.

1.4 Methodology

From topic selection to final report production, the study is carried out in a systematic manner. Finding the essential points required identifying and collecting data, which was then sorted, analyzed, evaluated, and presented in a methodical manner. In general, the methods used in the study are outlined in greater detail. Most of the information was collected from the official website of Robi Axiata Limited and the facts of this report reflects my personal experience.

Chapter 2

Company Profile

2.1 Background of the Company



Figure 2.1: Logo

Robi Axiata Limited is a public limited telecommunication company. The parent company is Axiata Group where Axiata Group Berhad of Malaysia holds the controlling stake of 61.82%, Bharti Airtel of India contains 28.18%, and general public contains the rest 10% stake. The company entered in the country's twin stock markets in Dhaka and Chittagong on the 24th of December 2020 with the largest ever IPO. The company started its journey in 1997 as Telekom Malaysia International (Bangladesh) with the brand name 'Aktel'. In 2010, they decided to change their brand name as 'Robi' and the company changed its name to Robi Axiata Limited. Robi is the second largest mobile network operator in Bangladesh with 53M subscribers, out of which close to 22.5M are 4G subscribers. Besides, the company also holds the highest percentage of data users in the industry which is 74%.

After merging with Airtel Bangladesh, the combined company, Robi Axiata Limited (Robi) started its journey on 16 November 2016. Till now, this is the biggest ever merger of the country and first ever merger in the mobile telecom sector of Bangladesh [1].

2.2 Objective of the Company

Robi believes in honesty, integrity and openness. It focuses on development of an agile work culture that thrives on collaboration. It wants to give the best telecommunication service to the people in Bangladesh. It now believes working in teams helps to develop a great working environment with so many creative minds.

2.3 Mission

- Empowering employees and customers.
- Develop and grow with their services.
- Ensuring quality management in each department.
- Value and share information and analytics to gain perspective.

2.4 Vision

- To be agile.
- Inspire to innovate opportunity.
- Collaborate and engage with teams.
- Build trust and transparency around communication.

2.5 Departments of the Company

Robi is a large company with various departments. The role of each department is unique from each other and works together to build a strong foundation for the company. It has each and every different department that is needed to run a company properly. Starting from the Finance department to Security and Safety Department.

Chapter 3

Training Phase

3.1 Overview

Being able to work as an intern in such a reputed organization has a great impact on both academic and professional life. It really needs guts to start working in a professional section alongside academic studies. In today's world, the competition in the job market is very high. Therefore, this internship opportunity played an important role to gather some work experience. It played as a medium for transition from theoretical to practical.

I joined Robi Axiata Limited as a virtual intern at the IT division. This internship is a graduation requirement for BRAC University and alongside that it is a great opportunity for me to learn various new things and gather valuable experience while working here. I barely had any idea about corporate life and how people work in teams at Robi. My Line Manager and co-supervisors helped me out in this regard. Their guidance and support helped me to gradually gather knowledge about Robi's IT environment.

3.2 Getting Familiar with the Workplace

Before starting the work, they gave me a KPI report which is a Key Responsibilities Report that contained all the necessary instructions about the projects. I was told to learn some fundamental concepts on RPA, Java, Python, Selenium, Testing, Deployment, Linux, Bash Scripting, Data Analysis, SQL, Web Scraping for creating highly scalable applications which will help in automation context.

3.3 Communication and Meetings

As my internship started from September 02,2021, firstly we interns had a meeting with the HR team to know about the policies and all other necessary information about our work and workplace. We got a laptop from the office to access Microsoft teams, outlook and other official sites. The office hour at Robi is basically from 10 am to 6.30 pm. I had to attend two meetings weekly where we discussed the agendas and the deliverables that I had to deliver every week. To ensure health safety in this pandemic, Robi decided to continue this internship virtually. Around 70 to 80% of employees worked from home. Therefore, our primary mode of communication

was Microsoft teams, outlook and whatsapp. Though we did home office, there was no communication gap among employees because we all were connected through meetings, group chats.

There was also a monthly huddle meeting with the HR where we interns had to show our progress and talk about any issues that we were facing. Overall, these online meetings and communications made it easier to share thoughts and connect with each other virtually.

3.4 Resources and Study Material

As I was new to automation projects, my instructors gave me website links and resources to get some ideas about automation. For example, how to generate csv files, broadcast automated emails and many more. They also encouraged me to enroll in Coursera to get a full overview of certain concepts. After that, they gave a brief overview of every project and gave ideas of what outputs I have to show.

3.5 VAS and Microservice

VAS means value added service which is a popular term in telecommunication industry. Like SMS, MMS and data access all belongs to VAS. Microservices deals with each function of an application as an independent service that can be altered, updated or taken down without affecting the rest of the application. RPA based projects can be a great way to start with microservice architecture. RPA is helpful to automate manual processes to reduce repetitive activities.

Chapter 4

My Contributions

I got the chance to work on some amazing automation projects which are described in details below.

4.1 Report Automation with Data Visualization Project

My first automation project was to create a generalized email broadcaster with data visualized email body from csv reports. This service connects to a database and extracts table contents. Then generates csv reports which is broadcasted through email as attachments and also as email body. To create this project I used python as a programming language and Mysql to create a dummy database.

4.1.1 Software Installation

- PyCharm IDE.
- Mysql.

4.1.2 Database Creation

As I do not have permission to access Robi's database, I had to create a dummy database named "employeedb" which contains four tables named Employee, Department, Attendance and Salary. This database contains information of 30 employees. I have attached the screenshots of the following database tables.

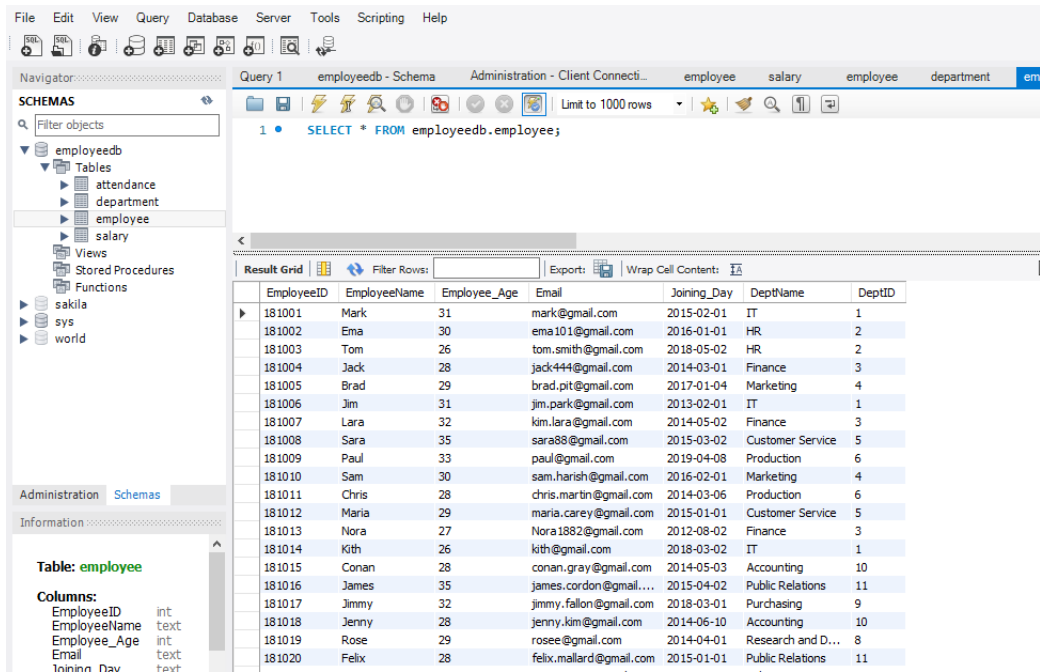


Figure 4.1: Employee table

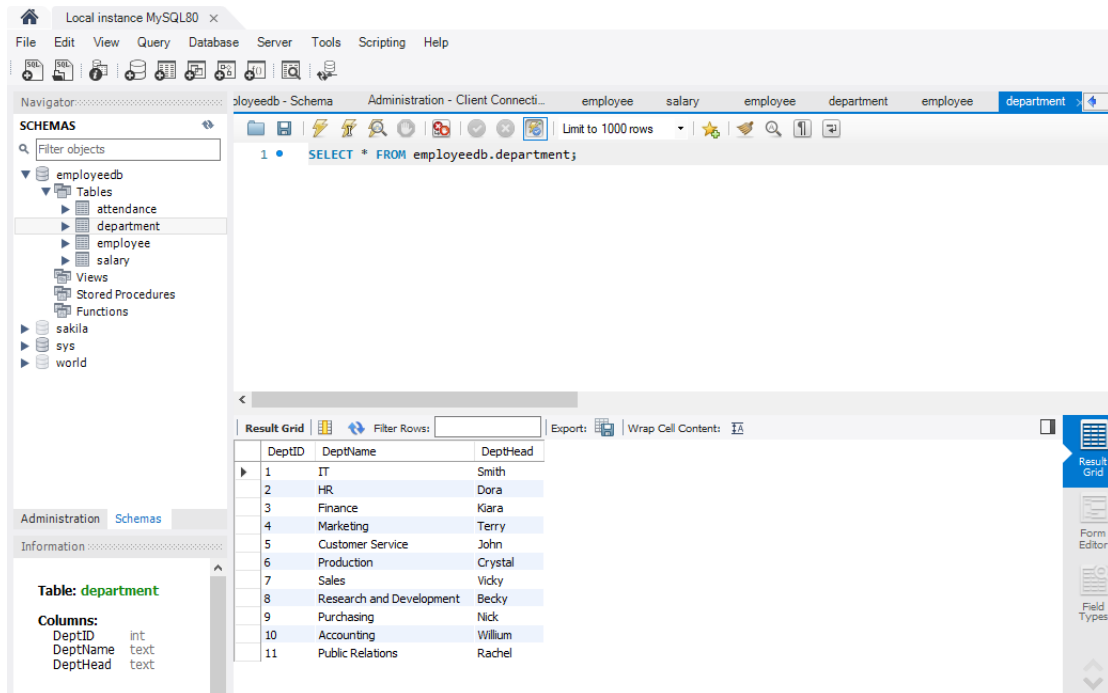


Figure 4.2: Department table

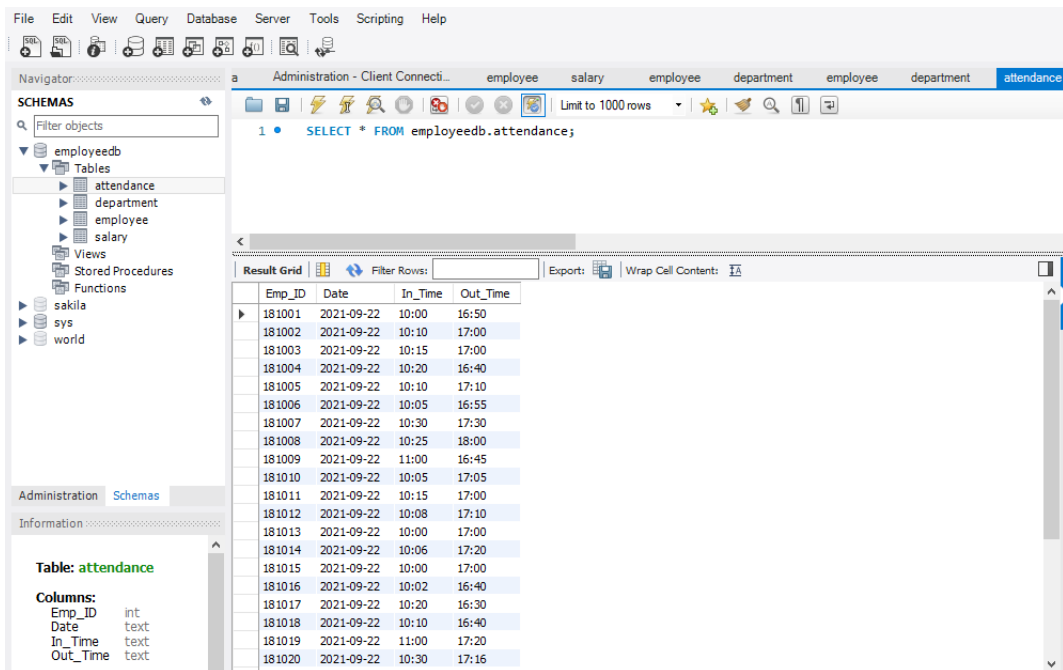


Figure 4.3: Attendance table

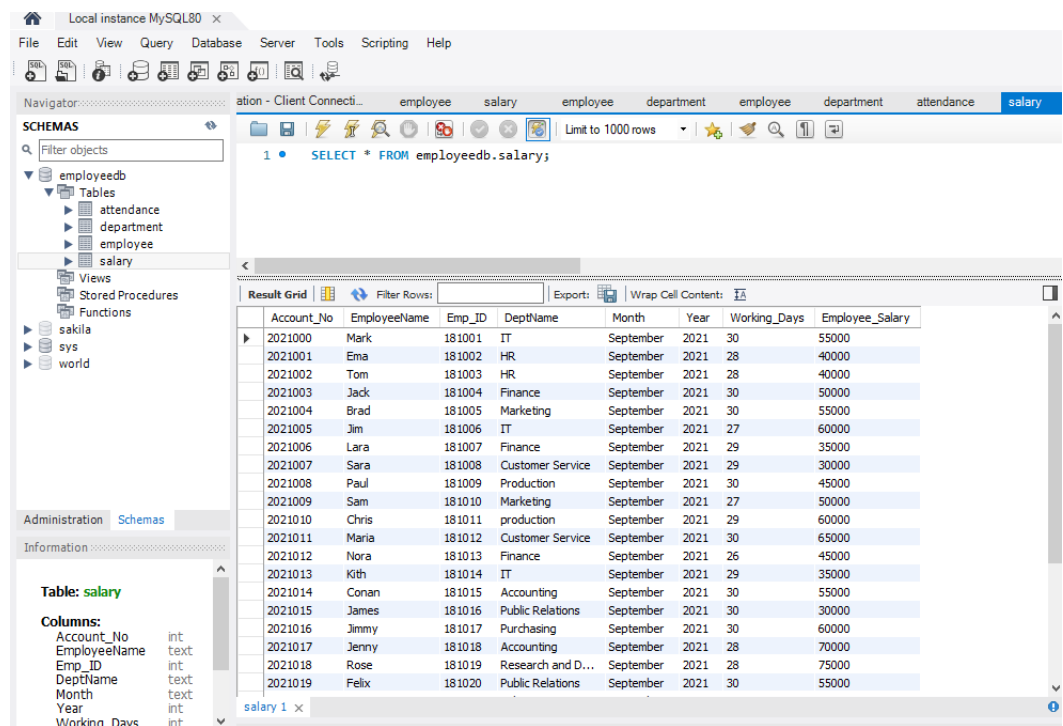


Figure 4.4: Salary table

4.1.3 Implementation

The full project was done using python programming language. Firstly, I wrote a program to connect the database and then extract the table contents as csv files. For this I had to use third party library mysql.connector and csv. If we run this python script, it will automatically generate four csv files for four database tables. Then I wrote another python script to broadcast these csv files as email attachments. As per the project requirement, the Salary Table has to be shown as an email body for which I used html formatting. If we run this script, it will automatically send four attachments to the receiver and the Salary Table as email body. The screenshot of the generated email is shown below

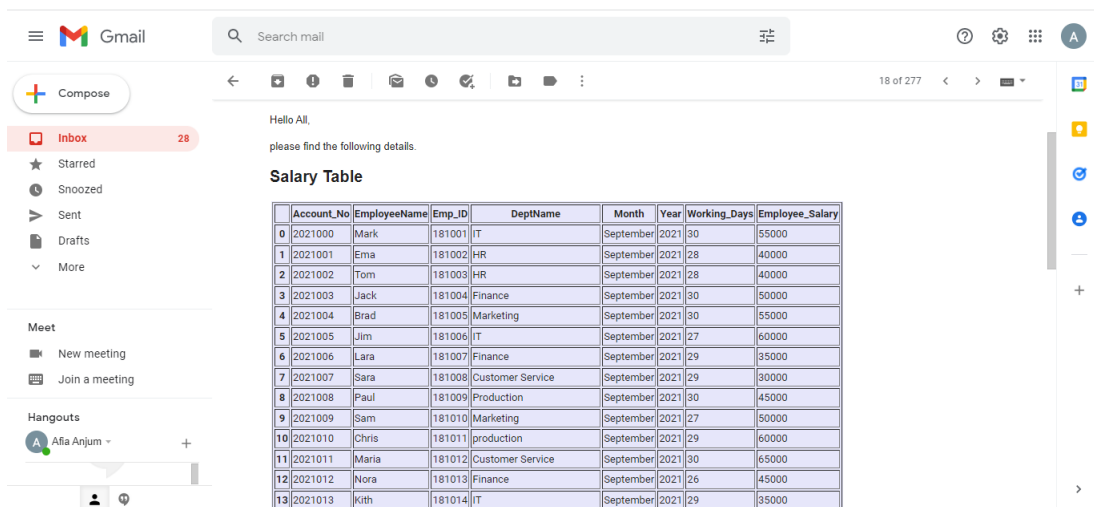


Figure 4.5: Screenshots of automated generated email

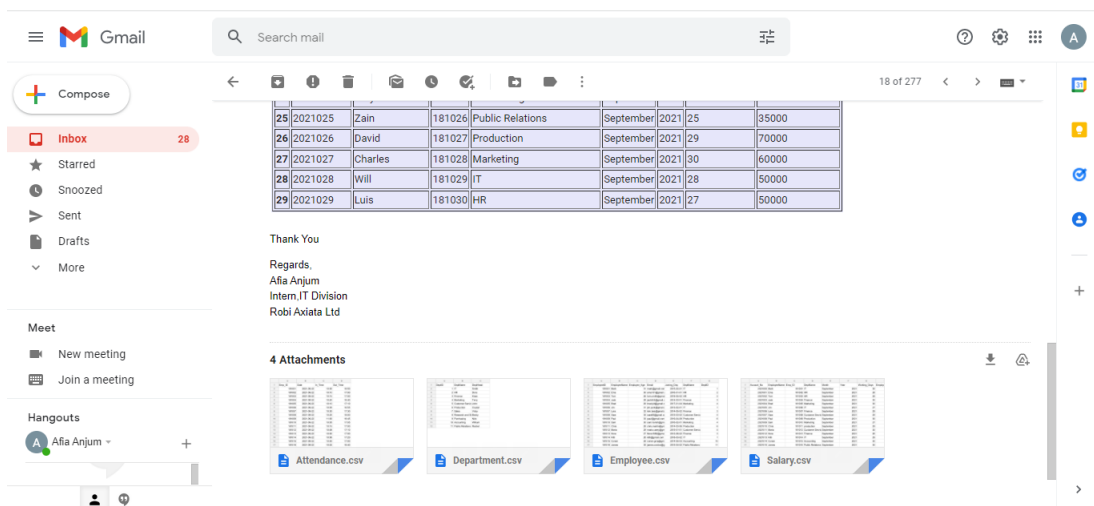


Figure 4.6: Screenshots of automated generated email

4.1.4 Service Diagram

**Generalized Email Broadcaster with Data
Visualized Email Body from CSV Reports**

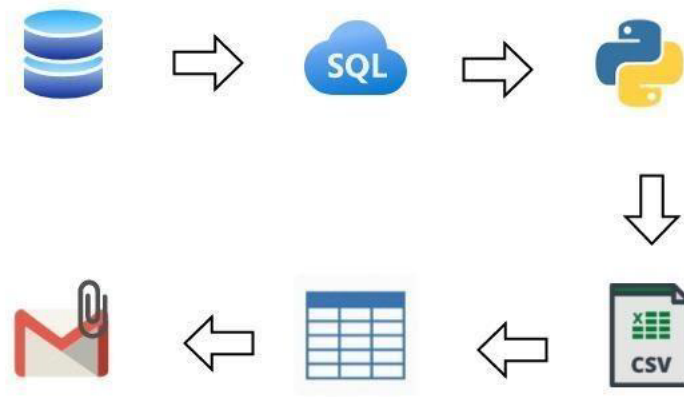


Figure 4.7: Service Diagram

4.2 Traffic Log File Parsing in txt format and CSV Report Generation

This is my second microservice project on automation. This service will read a .txt log file and generate a .csv file which will contain hit counts of the log file extracted using regex pattern. The log files are generated from one of the Robi servers. They used to manually check the hit counts from the log file. The main purpose of this automation is to extract those information without any human intervention which saves time and is also very fast. For this project, I also used python programming language. I wrote a python script which will read the log file and will extract Date, total success count, total failed count and total count from that. If we run the script, it will generate a .csv file containing these information.

```

localhost_access_log.2021-11-01 - Notepad
File Edit Format View Help
172.18.10.30 - - [01/Nov/2021:00:00:00 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801842104479 HTTP/1.1" 200 15
172.18.10.24 - - [01/Nov/2021:00:00:00 +0600] "GET /somapi/customerinfo/getallinfo/msisdn/8801849010097 HTTP/1.1" 200 25692
172.18.10.31 - - [01/Nov/2021:00:00:00 +0600] "POST /somapi/subscriberplan/add//8801618882495 HTTP/1.1" 200 153
172.18.10.29 - - [01/Nov/2021:00:00:00 +0600] "GET /somapi/eir/getmsisdn/msisdn/470022542747581 HTTP/1.1" 200 13
172.18.10.28 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801815856089 HTTP/1.1" 200 206
172.18.10.30 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801873935006 HTTP/1.1" 200 203
172.18.10.27 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801647464805 HTTP/1.1" 200 488
172.18.10.29 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801833760459 HTTP/1.1" 200 130
172.18.10.30 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801828451127 HTTP/1.1" 200 15
172.18.10.28 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801690158098 HTTP/1.1" 200 127
172.18.10.31 - - [01/Nov/2021:00:00:01 +0600] "POST /somapi/cdrquery/getinfo HTTP/1.1" 200 736261
172.18.10.31 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801815444067 HTTP/1.1" 200 16
172.18.10.26 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/customerinfo/getallinfo/msisdn/8801849010097 HTTP/1.1" 200 25692
172.18.10.29 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801608250777 HTTP/1.1" 200 346
172.18.10.30 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801831026720 HTTP/1.1" 200 15
172.18.10.25 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801832200021 HTTP/1.1" 200 206
172.18.10.31 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801629328264 HTTP/1.1" 200 198
172.18.10.29 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801642188133 HTTP/1.1" 200 486
172.18.10.29 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801660017970 HTTP/1.1" 200 125
172.18.10.31 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801641492529 HTTP/1.1" 200 16
172.18.10.27 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801875370832 HTTP/1.1" 200 15
172.18.10.31 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/customerinfo/getallinfo/msisdn/8801849010097 HTTP/1.1" 200 25692
172.18.10.29 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801613505555 HTTP/1.1" 200 198
172.18.10.27 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801864296468 HTTP/1.1" 200 15
172.18.10.31 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/point/msisdn/8801608674616 HTTP/1.1" 200 272
172.18.10.24 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/point/msisdn/8801627506776 HTTP/1.1" 200 273
172.18.10.24 - - [01/Nov/2021:00:00:03 +0600] "POST /somapi/subscriberplan/add//8801614258148 HTTP/1.1" 200 154
172.18.10.11 - - [01/Nov/2021:00:00:03 +0600] "-" 400 -
172.18.10.25 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/point/msisdn/8801871331163 HTTP/1.1" 200 273
172.18.10.27 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801843104215 HTTP/1.1" 200 16
172.18.10.31 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/optin/msisdn/8801648097754 HTTP/1.1" 200 136
172.18.10.30 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/lms/point/msisdn/8801849013120 HTTP/1.1" 200 935
172.18.10.31 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/balanceValidity/getinfo/msisdn/8801832795930 HTTP/1.1" 200 15
172.18.10.31 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/subscriberappender/subscriber/8801819841938/productid/1930518150 HTTP/1.1" 200 78
172.18.10.25 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/lms/point/msisdn/8801853084002 HTTP/1.1" 200 130
172.18.10.26 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/lms/point/msisdn/8801860451895 HTTP/1.1" 200 207

```

Figure 4.8: Screenshot of the log file.

4.2.1 Output

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|----------|---------------|--------------|-------------|---|---|---|---|---|---|---|---|
| 1 | Date | Total Success | Total Failed | Total Count | | | | | | | | |
| 2 | 1-Nov-21 | 3342 | 114 | 3456 | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | |

Figure 4.9: Generated .csv file.

4.2.2 Service Diagram

Traffic Log File Parsing in TXT Format and CSV Report Generation Service Diagram

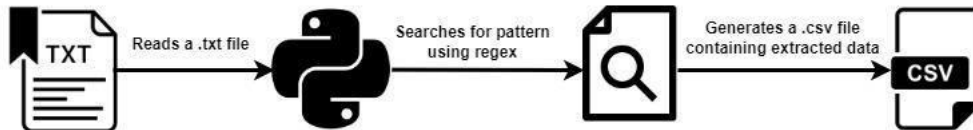


Figure 4.10: Service Diagram.

4.3 Multiple txt format traffic log files consolidated into single csv report

It is the extended service of the previous one. Here, this service will read multiple .txt files and generate a single csv report which will contain the Date, total success count, total failed count and total count which will be extracted from the log file. These log files are generated from the four servers of Robi. In the previous project, I worked with a single log file, in this task I had to extract data from four log files. At first place, this task was totally manual. One employee had to enter the four servers to take all the history and calculate the hit count manually. My task was to automate this system which will save their time.

4.3.1 Implementation

The project is done using Python programming language. I wrote one python script which read the four .txt files and finds out the total line count. Then using a regex pattern, it will find out the date, total success count and failed count. The code snippet is given below

```

import csv
import re
import glob
import datetime

today = datetime.date.today()
yesterday = today - datetime.timedelta(days=1)
path = "*" + str(yesterday) + ".txt"

count = 0
for files in glob.glob(path):
    with open(files, "r") as f:
        for lines in f:
            count += 1

Total_success_list = []
for files in glob.glob(path):
    with open(files) as f:
        file_data = f.read().replace('\n', '')
        date = re.search(r'(\d[0-9]*\/[a-zA-Z]*\/\d[0-9]*)', file_data).group(1)
        pattern = re.compile(r'(\s?200\s?)')
        success_list = pattern.findall(file_data)
        Total_success_list = Total_success_list + success_list

Total_Success = len(Total_success_list)
Total_Failed = count - Total_Success

```

Figure 4.11: Code Snippet of python script

```

172.18.10.30 - - [01/Nov/2021:00:00:00 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801842104479 HTTP/1.1" 200 15
172.18.10.24 - - [01/Nov/2021:00:00:00 +0600] "GET /somapi/customerInfo/getallInfo/msisdn/8801849010097 HTTP/1.1" 200 25692
172.18.10.31 - - [01/Nov/2021:00:00:00 +0600] "POST /somapi/subscriberplan/add/8801618882495 HTTP/1.1" 200 153
172.18.10.29 - - [01/Nov/2021:00:00:00 +0600] "GET /somapi/elr/getmsisdn/msisdn/470022542747581 HTTP/1.1" 200 13
172.18.10.28 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801815856089 HTTP/1.1" 200 206
172.18.10.30 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801873935006 HTTP/1.1" 200 203
172.18.10.27 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801647464805 HTTP/1.1" 200 488
172.18.10.29 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801833760459 HTTP/1.1" 200 130
172.18.10.30 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801828451127 HTTP/1.1" 200 15
172.18.10.28 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801690158098 HTTP/1.1" 200 127
172.18.10.31 - - [01/Nov/2021:00:00:01 +0600] "POST /somapi/cdrquery/getInfo HTTP/1.1" 200 736261
172.18.10.31 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801815444067 HTTP/1.1" 200 16
172.18.10.26 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/customerInfo/getallInfo/msisdn/8801849010097 HTTP/1.1" 200 25692
172.18.10.29 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801608250777 HTTP/1.1" 200 346
172.18.10.30 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801831026720 HTTP/1.1" 200 15
172.18.10.25 - - [01/Nov/2021:00:00:01 +0600] "GET /somapi/lms/point/msisdn/8801832200021 HTTP/1.1" 200 206
172.18.10.31 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801629328264 HTTP/1.1" 200 198
172.18.10.27 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801642188133 HTTP/1.1" 200 486
172.18.10.29 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801660017970 HTTP/1.1" 200 125
172.18.10.31 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801641492529 HTTP/1.1" 200 16
172.18.10.27 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801875370832 HTTP/1.1" 200 15
172.18.10.31 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/customerInfo/getallInfo/msisdn/8801849010097 HTTP/1.1" 200 25692
172.18.10.29 - - [01/Nov/2021:00:00:02 +0600] "GET /somapi/lms/point/msisdn/8801613505555 HTTP/1.1" 200 198
172.18.10.27 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801864296468 HTTP/1.1" 200 15
172.18.10.31 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/point/msisdn/8801608674616 HTTP/1.1" 200 272
172.18.10.24 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/point/msisdn/8801627506776 HTTP/1.1" 200 273
172.18.10.24 - - [01/Nov/2021:00:00:03 +0600] "POST /somapi/subscriberplan/add/8801614258148 HTTP/1.1" 200 154
172.18.10.11 - - [01/Nov/2021:00:00:03 +0600] "-" 400 -
172.18.10.25 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/point/msisdn/8801871331163 HTTP/1.1" 200 273
172.18.10.27 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801843104215 HTTP/1.1" 200 16
172.18.10.31 - - [01/Nov/2021:00:00:03 +0600] "GET /somapi/lms/optin/msisdn/8801648097754 HTTP/1.1" 200 136
172.18.10.30 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/lms/point/msisdn/8801849013120 HTTP/1.1" 200 935
172.18.10.31 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/balanceValidity/getInfo/msisdn/8801832795930 HTTP/1.1" 200 15
172.18.10.31 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/subscriberappender/subscriber/8801819841938/productid/1930518150 HTTP/1.1" 200 78
172.18.10.25 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/lms/point/msisdn/8801853084082 HTTP/1.1" 200 130
172.18.10.26 - - [01/Nov/2021:00:00:04 +0600] "GET /somapi/lms/point/msisdn/8801860451895 HTTP/1.1" 200 207

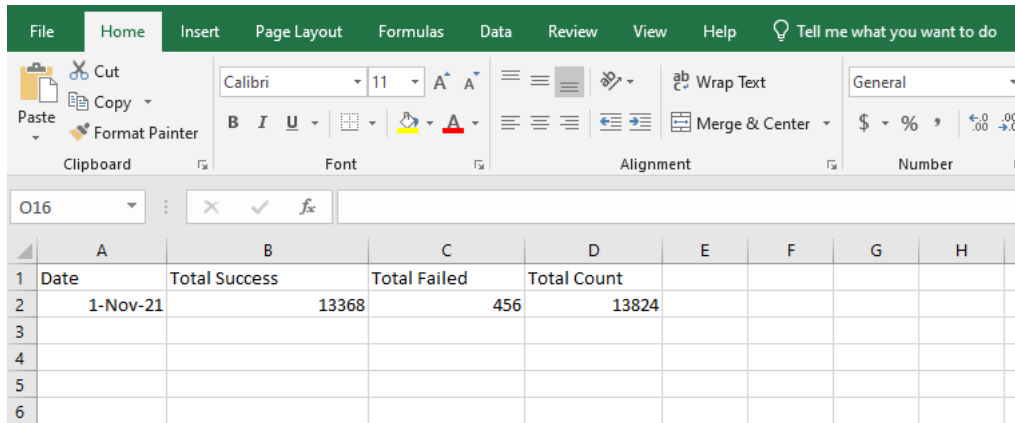
```

Figure 4.12: Log File in .txt format generated from server 1(Total 3456 data).

In Fig 4.12, we can see the log file generated from server 1. The other three servers log files also look like this. Each of these log file contains 3456 datas.

4.3.2 Output

This is the consolidated output after collecting datas from the four servers.



| | A | B | C | D | E | F | G | H |
|---|----------|---------------|--------------|-------------|---|---|---|---|
| 1 | Date | Total Success | Total Failed | Total Count | | | | |
| 2 | 1-Nov-21 | 13368 | 456 | 13824 | | | | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| 6 | | | | | | | | |

Figure 4.13: Consolidated output from four log files.

4.3.3 Service Diagram

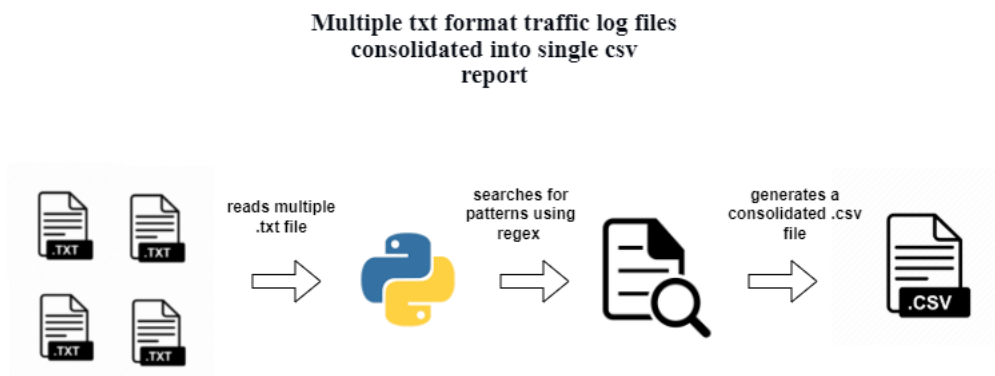


Figure 4.14: Service Diagram

4.4 Generalized CSV Reports Email Broadcaster

With this microservice we can send notifications to someone automatically. This service will help to visualize the contents of a csv file in the mail body and the csv file as attachment. We can execute this program using the windows scheduler which will notify the concern at a certain time and give him an overview of the content. As a result, the employee does not have to write an email daily and send it to someone.

4.4.1 Implementation

The project is done using the Python programming language. For this I used smtplib library for sending email, pandas for converting the csv file into a dataframe dynamically and then html formatting to visualize the csv file content in the email body. We can format any kind of csv file with this code because the csv file is converted to tabular format dynamically. We have to just mention the csv file name into the `pd.read_csv()` method.

```
import smtplib
from email.message import EmailMessage
import pandas as pd
import glob

msg = EmailMessage()
msg['Subject'] = "Traffic Automation with Data Visualization"
msg['From'] = "*****"
msg['To'] = "*****@gmail.com"

df = pd.read_csv("Consolidated_Output.csv")
msg.add_alternative('''\
    <head></head>
    <body>
        <p>Dear Concern,<br>
        <p>Please find the following details.</br>
        <h1 style="font-size:15px;" >Hit Count</h1>
        <tr style="background-color:#E6E6FA">
        <body>''' + df.to_html(classes='table ', justify="center") +
        '''</br></body>

        <tr style="text-align: center;"></tr>
        </tr>
        </table>
        <font color = "black">
        <br></br>
```

Figure 4.15: Code Snippet of the python script

```
for files in glob.glob("*.csv"):
    with open(files, "rb") as f:
        file_data = f.read()
        file_name = f.name
        msg.add_attachment(file_data, maintype="application", subtype="csv", filename=file_name)
        print("Attachment added ", file_name)

with smtplib.SMTP_SSL("smtp.gmail.com", ***) as server:
    server.login("*****@gmail.com", "*****")
    server.send_message(msg)

print("Email Sent!")
```

Figure 4.16: Code Snippet of the python script

4.4.2 Output

Here the output is shown in a tabular format. So, when the receiver opens the email he/she can get an overview of the attachment in a glance.

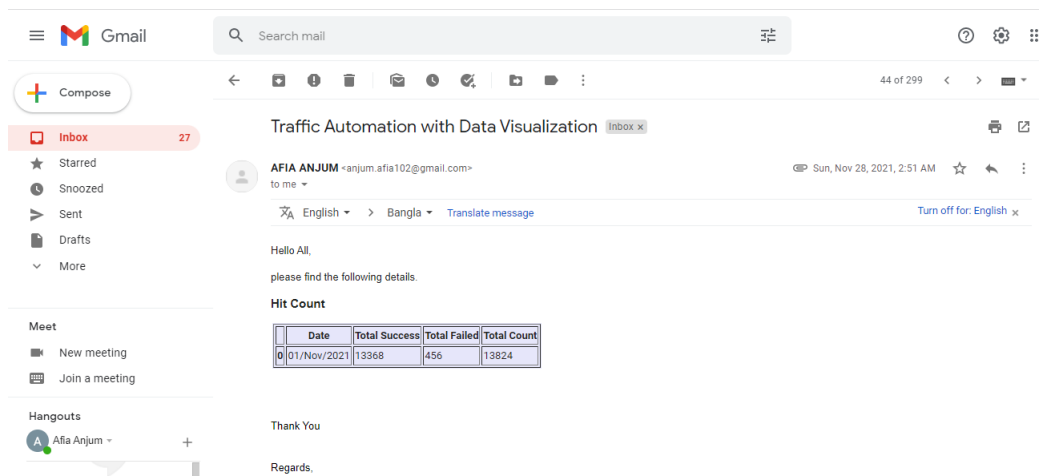


Figure 4.17: Output of Email Notification

4.4.3 Service Diagram

Generalized CSV Reports Email Broadcaster

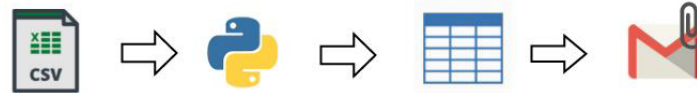


Figure 4.18: Service Diagram

4.5 RPA: Extracting CSV Reports by web scraping in HTML webpage

Web scraping is a common process to extract data and information from a website. With this service we can extract information from any website by using selectors and get the content of that website or webpage. It was a practice for the next web scraping project. My instructors guided me to scrape Robi's e-Approval website. The task of this microservice is to load the "e-Approval" webpage in chrome browser and extract information which will be stored in a csv file.

4.5.1 Implementation

The whole project is made using JAVA-jdk1.8. To do web scraping I used selenium jar as a third party library. Then to load the website in chrome browser, chrome web driver was installed. When the script is run, it loads the webpage in chrome browser, but a "Sign in" popup comes to enter to the website.

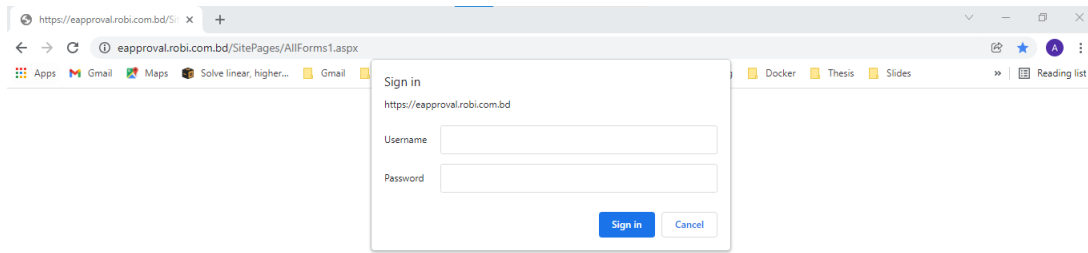


Figure 4.19: Sign In page of e-approval website

To handle this popup and sign in automatically, I had to install another software which is AutoIT. In this software I wrote a script with my username and password which helps to sign in to the website. It automatically fills out the username and password block with the user's information and clicks on the Sign in button. After signing in the website looks like this.

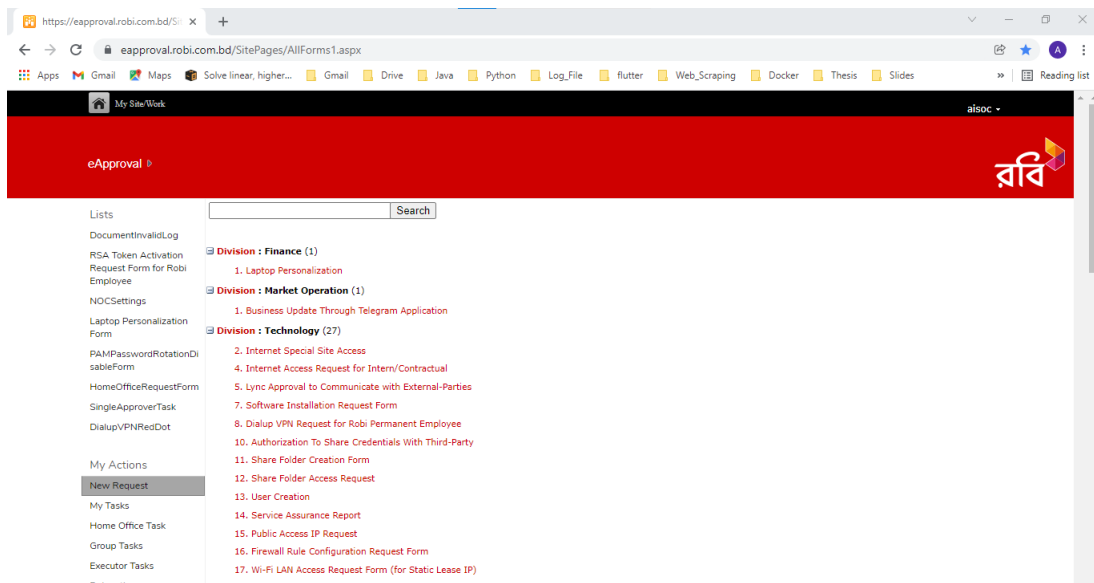
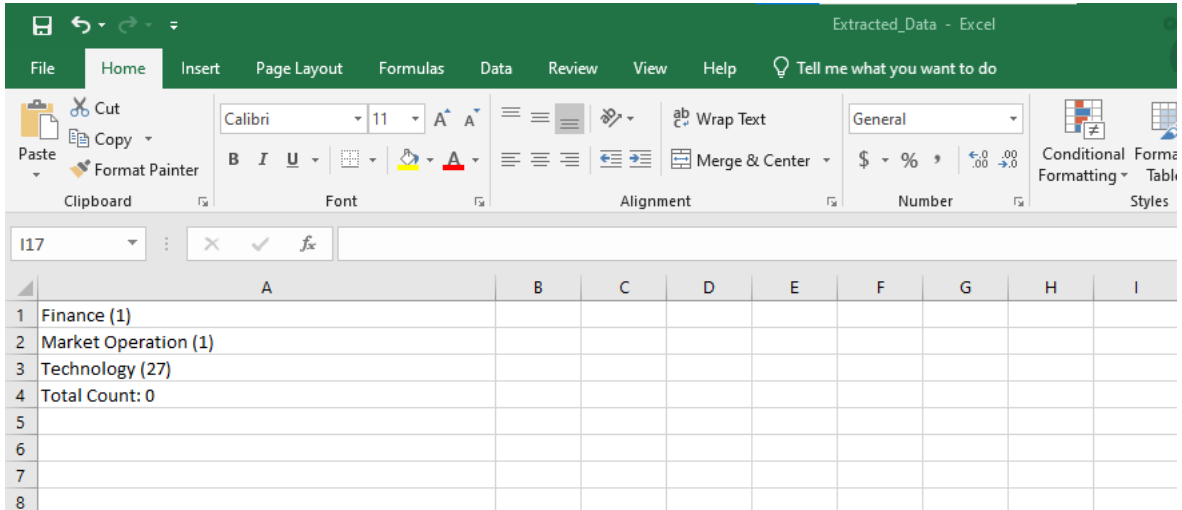


Figure 4.20: Screenshot of the e-Approval Website

4.5.2 Output

As this was for practice, so I had to collect the information from the first loaded page which is the division name and the count by scraping this and generate a csv file containing the extracted information. The output is given below:



The screenshot shows the Microsoft Excel interface with a CSV file named 'Extracted_Data'. The data is as follows:

| | A | B | C | D | E | F | G | H | I |
|---|----------------------|---|---|---|---|---|---|---|---|
| 1 | Finance (1) | | | | | | | | |
| 2 | Market Operation (1) | | | | | | | | |
| 3 | Technology (27) | | | | | | | | |
| 4 | Total Count: 0 | | | | | | | | |
| 5 | | | | | | | | | |
| 6 | | | | | | | | | |
| 7 | | | | | | | | | |
| 8 | | | | | | | | | |

Figure 4.21: Screenshot of the generated csv file

4.5.3 Service Diagram

RPA: Extracting CSV Reports by Web Scraping in HTML Webpage

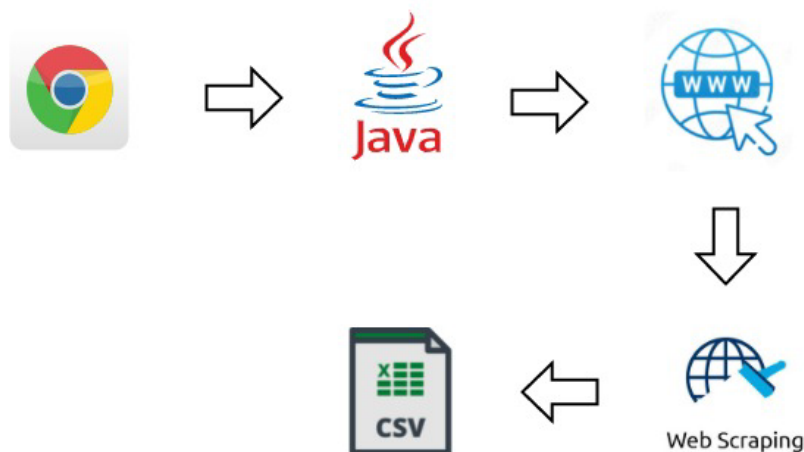


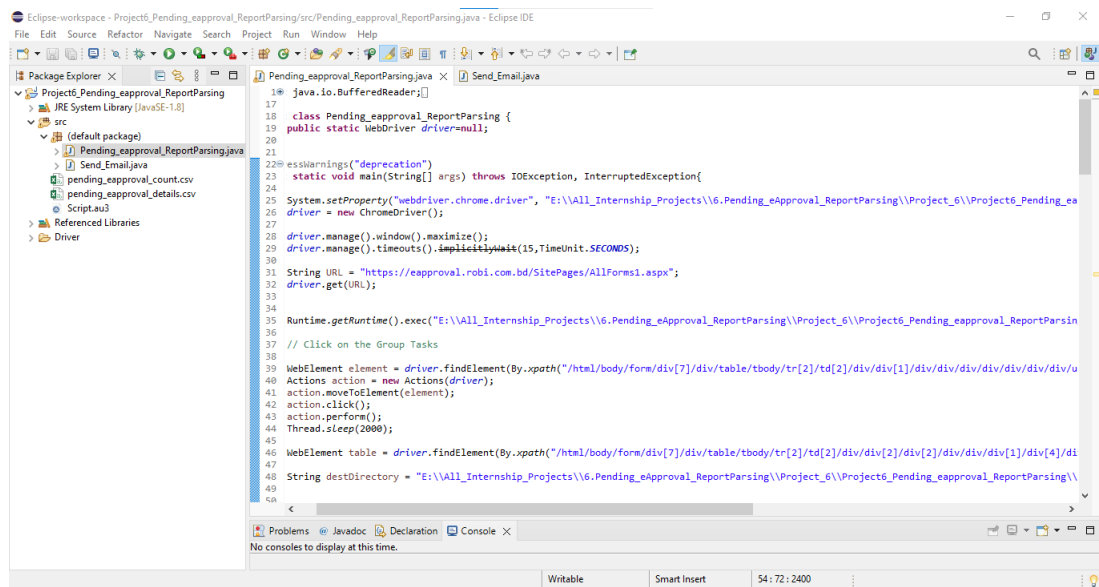
Figure 4.22: Service Diagram

4.6 Pending e-Approval Report Parsing in CSV Format and Consolidated Report Generation

The main purpose of this microservice is to load the e-Approval website and generate two csv reports which contain pending e-Approval details and pending e-Approval counts extracted by web scraping. Then sends these as attachments through email. It is a web bot which is given with some instructions and it performs those accordingly. Previously, these details were handled manually. For example, employees had to check all the pending details and counts manually. By this web bot they can have a daily overview of these details through email notification which has increased the efficiency of their daily work and saves a lot of time.

4.6.1 Implementation and Output

The whole project is done using JAVA-jdk1.8. There are two java files. One is for launching the chrome browser and generate two csv files by web scraping. The second java file is for sending these two csv files as email attachments to an employee so that he/she can get an overview of the pending details without logging into the e-approval website. The code snippet is given below



```
17 java.io.BufferedReader;[]
18 class Pending_approval_ReportParsing {
19     public static WebDriver driver=null;
20
21
22     @SuppressWarnings("deprecation")
23     static void main(String[] args) throws IOException, InterruptedException{
24
25         System.setProperty("webdriver.chrome.driver", "E:\\All_Internship_Projects\\6.Pending_eApproval_ReportParsing\\Project_6\\Project6_Pending_ea
26         driver = new ChromeDriver();
27
28         driver.manage().window().maximize();
29         driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
30
31         String URL = "https://eapproval.robi.com.bd/SitePages/AllForms1.aspx";
32         driver.get(URL);
33
34
35         Runtime.getRuntime().exec("E:\\All_Internship_Projects\\6.Pending_eApproval_ReportParsing\\Project_6\\Project6_Pending_eapproval_ReportParsin
36
37         // Click on the Group Tasks
38
39         WebElement element = driver.findElement(By.xpath("/html/body/form/div[7]/div/table/tbody/tr[2]/td[2]/div/div[1]/div/div/div/div/div/div/u
40         Actions action = new Actions(driver);
41         action.moveToElement(element);
42         action.click();
43         action.perform();
44         Thread.sleep(2000);
45
46         WebElement table = driver.findElement(By.xpath("/html/body/form/div[7]/div/table/tbody/tr[2]/td[2]/div/div[2]/div[2]/div/div/div[1]/div[4]/di
47
48         String destDirectory = "E:\\All_Internship_Projects\\6.Pending_eApproval_ReportParsing\\Project_6\\Project6_Pending_eapproval_ReportParsing\\"
49
50
```

Figure 4.23: Screenshot of the codebase

At first, it automatically signs in to the website and clicks on the “Group Task” link which loads the Executer page.

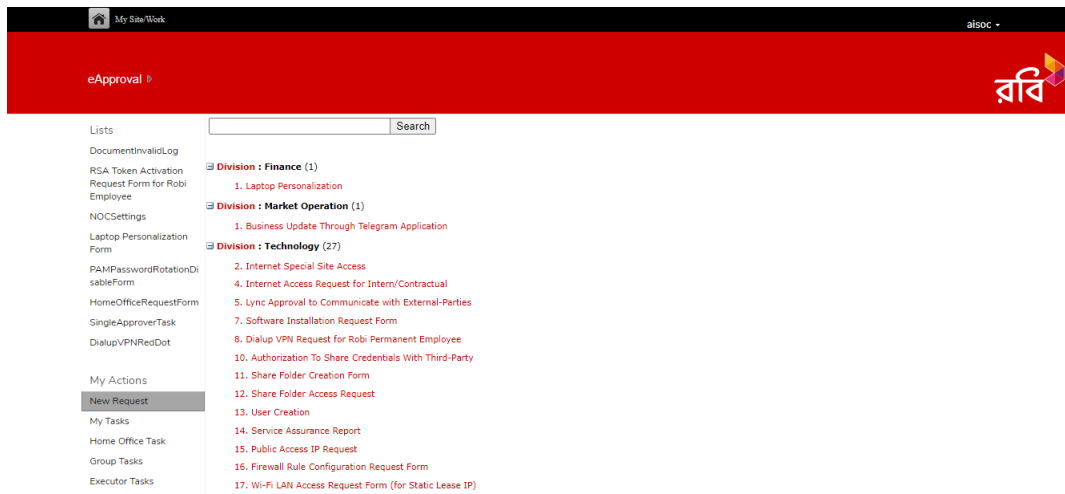


Figure 4.24: Screenshot of the loaded website

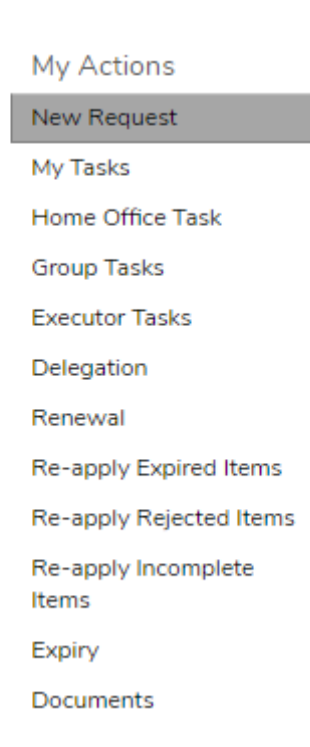


Figure 4.25: Web bot clicks on this Group Task link

After clicking on the “Group Task” link, following Executor page opens.

The screenshot shows the 'eApproval Executor' interface. On the left is a navigation menu with options like 'Lists', 'My Actions', and 'Group Tasks'. The main area displays a table of tasks. At the top, it shows 'Total Seconds: 25,235207 Total Count: 80' and an 'Assign To Me' button. The table has the following data:

| RequestID | Title | Executor's Group | Created | Status | Last Approved At |
|-----------|--|-------------------------------|---------------------|--------------|---------------------|
| AUC108851 | Application user creation by Md. Rokibul Islam for IoT (Renewal 1) | All Digital Service IoT & VAS | 11/01/2022 12:26:40 | All Approved | 11/01/2022 17:57:41 |
| AUC105336 | Application user creation by Md. Rokibul Islam for IoT (Renewal 1) | All Digital Service IoT & VAS | 11/01/2022 12:26:31 | All Approved | 11/01/2022 17:57:43 |
| AUC113475 | Application user creation by Md. Ershad Ferdous for Ringing tones(XRBT) | DSIV Platform Management | 06/01/2022 12:53:23 | All Approved | 06/01/2022 15:13:10 |
| AUC99932 | Application user creation by Md Mizanur Rahman for VAS Cloud - SMSC (Renewal 1) | DSIV Platform Management | 05/01/2022 17:18:59 | All Approved | 06/01/2022 15:13:37 |
| AUC75633 | Application user creation by Md Mizanur Rahman for Missed Call Alert (MCA) (Renewal 2) (Renewal 3) | DSIV Platform Management | 06/01/2022 17:12:24 | All Approved | 06/01/2022 15:13:35 |
| AUC106166 | Application user creation by Md Mizanur Rahman for Digital Reach (Renewal 1) | DSIV Platform Management | 05/01/2022 17:11:46 | All Approved | 06/01/2022 15:13:30 |
| AUC102900 | Application user creation by Md Mizanur Rahman for VAS Cloud - SMSC (Renewal 1) | DSIV Platform Management | 05/01/2022 17:10:22 | All Approved | 06/01/2022 15:13:28 |
| AUC99932 | Application user creation by Md Mizanur Rahman for VAS Cloud - SMSC (Renewal 1) | DSIV Platform Management | 05/01/2022 17:08:28 | All Approved | 06/01/2022 15:13:21 |
| AUC98725 | Application user creation by Md. Rokibul Islam for IoT (Renewal 1) | All Digital Service IoT & VAS | 05/01/2022 10:48:16 | All Approved | 11/01/2022 17:57:45 |
| AUC103157 | Application user creation by Rahnuma Sharmin for VAS Cloud - SMSC (Renewal 1) | DSIV Platform Management | 02/01/2022 12:01:45 | All Approved | 05/01/2022 16:07:05 |
| AUC98691 | Application user creation by Md. Rokibul Islam for dCRM (Renewal 1) | DSIV Platform Management | 09/49:21 | All Approved | 11/01/2022 17:57:48 |

Figure 4.26: Executor page

The web bot reads this whole table and extracts the details of all the columns of it. Then writes the all information in csv file which is the “pending e-Approval detail.csv” that looks as follows.

The screenshot shows an Excel spreadsheet titled 'pending_eapproval_details - Excel'. The data is organized as follows:

| RequestID | Title | Executor's Group | Created | Status | Last Approved At |
|-----------|--|-------------------------------|---------------------|--------------|---------------------|
| AUC112911 | Application user creation by Ismat Ara Begum for dCRM | DSIV Platform Management | 20/12/2021 17:16:12 | All Approved | 21/12/2021 10:42:55 |
| DBUC6763 | Database user creation by Ohida Binte Amin for USSD GW | DSIV Platform Management | 20/12/2021 16:39:27 | All Approved | 21/12/2021 14:59:48 |
| DBUC6762 | Database user creation by Ohida Binte Amin for USSD GW | DSIV Platform Management | 20/12/2021 16:38:24 | All Approved | 21/12/2021 14:59:46 |
| DBUC6761 | Database user creation by Ohida Binte Amin for USSD GW | DSIV Platform Management | 20/12/2021 16:37:12 | All Approved | 21/12/2021 14:59:44 |
| AUC97656 | Application user creation by Mahfuz Ahmed Chowdhury for dCRM (Renewal 1) | DSIV Platform Management | 20/12/2021 10:21:36 | All Approved | 20/12/2021 13:57:57 |
| WUC2710 | Windows User Creation by Md. Khabbab Hossain for FnF (Renewal 1) | DSIV Platform Management | 19/12/2021 17:45:29 | All Approved | 20/12/2021 13:57:53 |
| AUC112883 | Application user creation by Maisha Hoque for VAS Cloud - SMSC | DSIV Platform Management | 19/12/2021 17:45:29 | All Approved | 21/12/2021 10:42:53 |
| AUC112882 | Application user creation by Md. Arifur Rahman for dCRM | DSIV Platform Management | 19/12/2021 16:55:54 | All Approved | 19/12/2021 19:21:31 |
| AUC103628 | Application user creation by S.M. Imteaj for dCRM (Renewal 1) | DSIV Platform Management | 19/12/2021 12:47:18 | All Approved | 19/12/2021 16:55:06 |
| AUC101549 | Application user creation by S.M. Imteaj for dCRM (Renewal 1) | DSIV Platform Management | 19/12/2021 12:42:00 | All Approved | 19/12/2021 16:55:08 |
| AUC101544 | Application user creation by S.M. Imteaj for dCRM (Renewal 1) | DSIV Platform Management | 19/12/2021 12:38:34 | All Approved | 19/12/2021 16:55:11 |
| AUC112849 | Application user creation by Md. Mezbah Uddin for dCRM | DSIV Platform Management | 16/12/2021 17:17:11 | All Approved | 19/12/2021 11:08:47 |
| WUC3124 | Windows User Creation by Md. Khabbab Hossain for FnF | DSIV Platform Management | 15/12/2021 16:05:11 | All Approved | 17/12/2021 09:35:19 |
| AUC96597 | Application user creation by Hasan MD Nafis Rahman for dCRM (Renewal 1) | DSIV Platform Management | 15/12/2021 14:50:21 | All Approved | 17/12/2021 09:35:13 |
| AUC96881 | Application user creation by Md. Abu Bakar Shibly for dCRM (Renewal 1) | DSIV Platform Management | 15/12/2021 12:19:02 | All Approved | 17/12/2021 09:35:16 |
| AUC112832 | Application user creation by Chhanda Saha Pinki for dCRM | DSIV Platform Management | 14/12/2021 14:28:37 | All Approved | 15/12/2021 11:04:12 |
| AUC104009 | Application user creation by Md. Saydul Hoque for dCRM (Renewal 1) | DSIV Platform Management | 14/12/2021 09:43:59 | All Approved | 15/12/2021 11:04:17 |
| AUC112811 | Application user creation by Mohammad Zahurul Islam for dCRM | DSIV Platform Management | 13/12/2021 17:34:46 | All Approved | 20/12/2021 10:38:05 |
| WUC2790 | Windows User Creation by Md. Khabbab Hossain for FnF (Renewal 1) | DSIV Platform Management | 13/12/2021 16:37:46 | All Approved | 15/12/2021 11:04:19 |
| AUC112795 | Application user creation by Ayman Ahmed Tanmoy for IoT | All Digital Service IoT & VAS | 13/12/2021 10:32:55 | All Approved | 13/12/2021 21:25:27 |
| AUC112794 | Application user creation by Ayman Ahmed Tanmoy for IoT | All Digital Service IoT & VAS | 13/12/2021 10:18:47 | All Approved | 13/12/2021 21:25:25 |
| AUC112783 | Application user creation by Ismat Ara Begum for dCRM | DSIV Platform Management | 12/12/2021 16:57:11 | All Approved | 20/12/2021 10:38:03 |

Figure 4.27: Output of the generated pending e-Approval details

Then the bot reads this csv file and uses regex pattern to extract the information of the type of requests done by employees and keeps a count of it.

| | A | B | C |
|----|--|-------------------------|----|
| 1 | Service Name/ User Name | Pending Execution Count | |
| 2 | dCRM (Renewal 2) | | 1 |
| 3 | dCRM (Renewal 1) | | 12 |
| 4 | FnF (Renewal 1) | | 2 |
| 5 | USSD GW | | 3 |
| 6 | VAS Cloud - SMSC | | 1 |
| 7 | Network Element User Creation Request Form : Requested By : Ohida Binte Amin | | 1 |
| 8 | Mushfiq Solaiman Khan | | 2 |
| 9 | IoT | | 6 |
| 10 | Mohammad Nur Alam (Renewal 1) (Renewal 1) (Renewal 2) | | 2 |
| 11 | dCRM | | 15 |
| 12 | IoT (Renewal 3) | | 2 |
| 13 | IoT (Renewal 1) | | 2 |
| 14 | Campaign Management | | 1 |
| 15 | Abdullah- Al- Mamun | | 3 |
| 16 | FnF | | 1 |
| 17 | Total | | 54 |
| 18 | | | |
| 19 | | | |
| 20 | | | |
| 21 | | | |
| 22 | | | |
| 23 | | | |

Figure 4.28: Output of generated pending e-Approval Count

4.6.2 Service Diagram

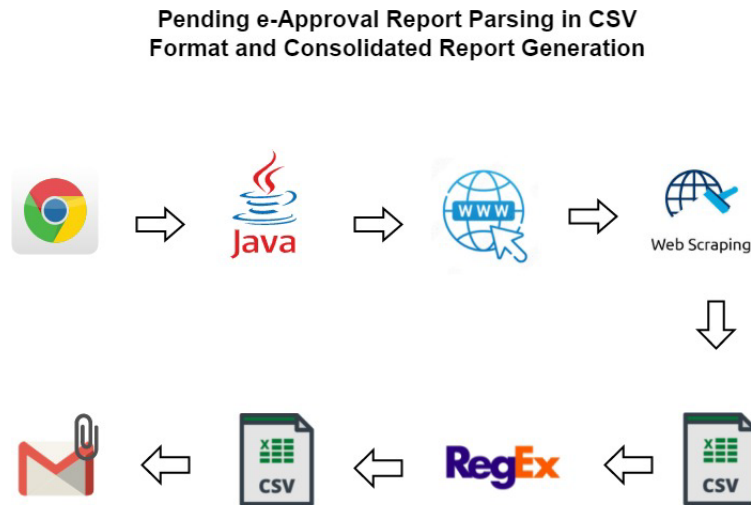


Figure 4.29: Service Diagram

4.7 Outlook Email Parsing Automation

Outlook is a popular software used in many organizations. Robi uses outlook for official communication among the employees. As a result, each employee gets tons of emails every day. So my instructor thought of automating this outlook emails so that they can send auto replies to some specific emails and save their time. In this service, I am instructed to design an automation system which will read some specific unread emails from the inbox and give auto replies to that. After that, it will move those replied emails to another folder by keeping them unread so that when the user opens the outlook he/she can see the auto replied emails later on and keep track of it.

4.7.1 Implementation

The whole project is done using python. I used win32com.client library. Firstly, it connects to the outlook application using win32com.client.Dispatch() function. Then it accesses the inbox folder using GetDefaultFolder(6) function. Here number 6 defines the inbox folder in outlook. After that, the program creates a folder under the inbox folder named 'Bot_Emails'. Then the program checks every email in the inbox using a for loop and searches for certain keywords in the email subject or in the email body as per the requirement. I set some auto replies for each of the keyword according to our criteria. If it finds the matching keyword then the program auto replies to those unread emails and moves the replied emails to the 'Bot_Emails' folder as unread emails using Move() function.

Some code snippets are given below.

```
import win32com.client

outlook = win32com.client.Dispatch("Outlook.Application")
namespace = outlook.GetNameSpace('MAPI')
inbox = namespace.GetDefaultFolder(6)
Bot_Emails = inbox.Folders['Bot_Emails']
messages = inbox.Items
Bot_Emails_list = []
count = 0

print("The Total Number of Messages in Inbox:", len(messages))
print("-----")

for msg in messages:
    mail = outlook.CreateItem(0)
    if msg.UnRead == True:
        count = count + 1
        if "approval" in msg.subject.lower():
            print("The Status of the Message: Unread =>", msg.UnRead)
            print("Subject:", msg.subject)

            mail.Subject = 'Regarding approval'
            mail.Body = 'Dear Concern,\nI have received your mail.For further information please follow this link.' \
                '\nhttps://eapproval.robi.com.bd/SitePages/AllForms1.aspx'
            mail.To = msg.SenderEmailAddress
```

Figure 4.30: Code Snippet

```
52
53     elif 'congratulations' in msg.Body.lower():
54         print("The Status of the Message: Unread =>", msg.UnRead)
55         print("Body:", msg.Body)
56         mail.Subject = 'Thank You'
57         mail.Body = "Thank you for reaching out to me! It really means a lot."
58         mail.To = msg.SenderEmailAddress
59         mail.Send()
60         print("Reply sent for the congratulations mail")
61         print("-----")
62         Bot_Emails_list.append(msg)
63
64     else:
65         print("No New Email Found")
66         print("-----")
67
68 print("The Total Number of Unread Messages:", count)
69 print("-----")
70
71 for emails in Bot_Emails_list:
72     emails.Move(Bot_Emails)
73
```

Figure 4.31: Code Snippet

4.7.2 Output

From the figure 4.32 we can see that there are two unread emails. So if we run the program, it will send auto replies to the sender and move these two emails to 'Bot_Emails' folder.

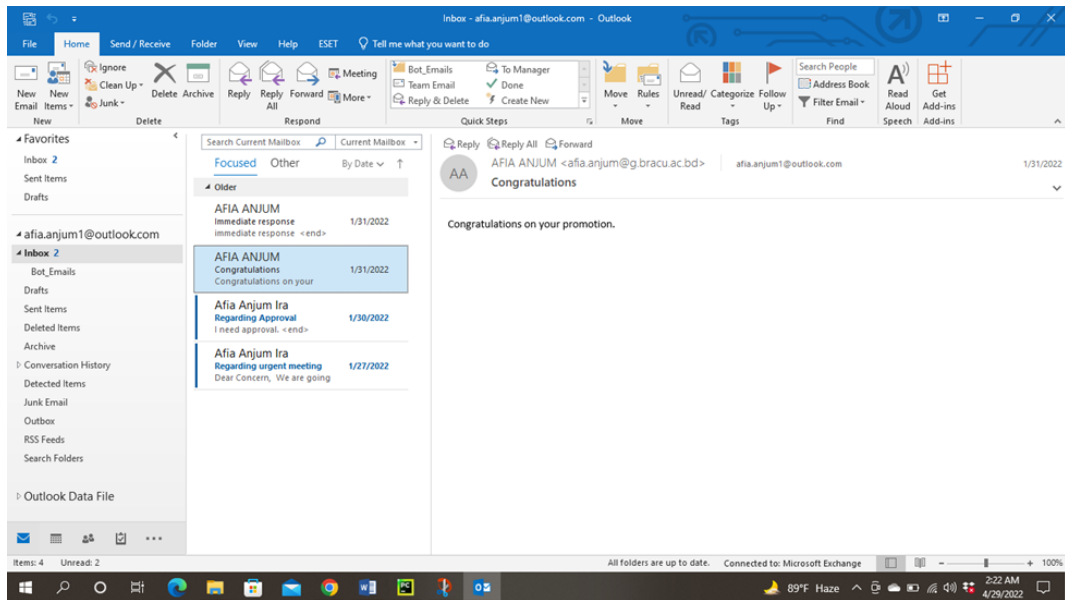


Figure 4.32: Screenshot of the 'Inbox' folder

The replied emails are moved to Bot_Emails folder.

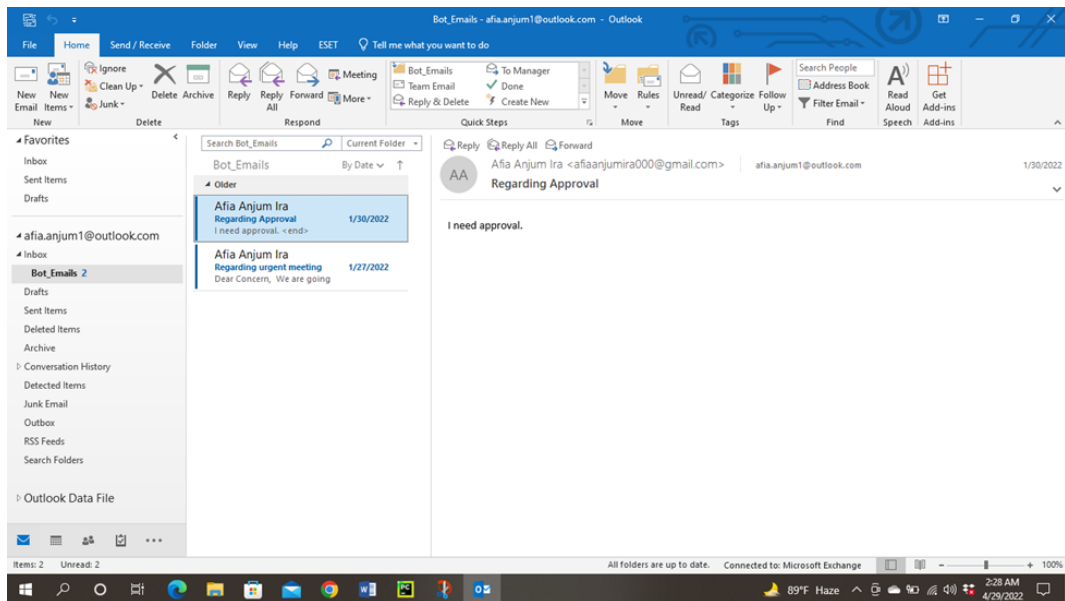


Figure 4.33: Screenshot of 'Bot_Emails' folder

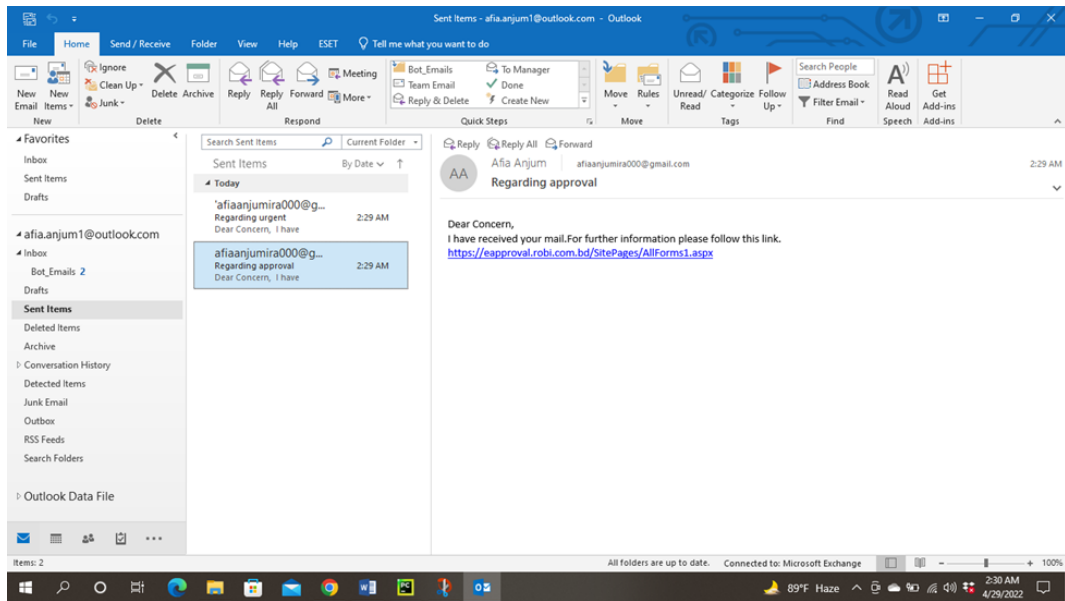


Figure 4.34: Screenshot of the 'Sent Items' folder

4.7.3 Service Diagram

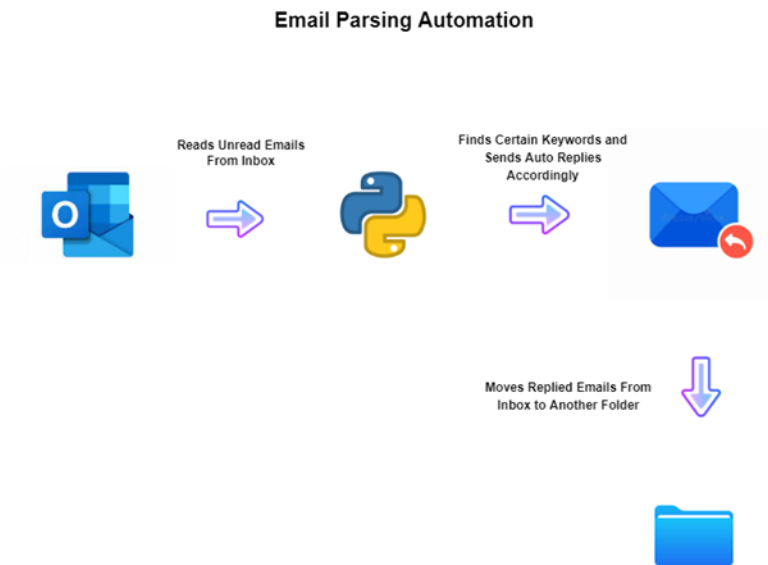


Figure 4.35: Service Diagram

| | Summary | Description | Root Cause Detail | DateTime | USSD Code | DateTime.1 | USSD Code.1 |
|---|--|---|---|----------|-----------|---------------------------|-------------------------|
| 0 | | | USSD not working related no problem found from... | 0 | 0 | .. | .. |
| 1 | | As per customer claim that he facing ussd dial... | Please ask cx to try again after restarting ha... | 0 | 0 | .. | .. |
| 2 | | As per customer claim that he want to activate... | As per HLR, cx already activate call waiting s... | 1 | 1 | , 10/10/202111 , | , *43# , |
| 3 | 100taka 10GB data for 30days *212*901# | 100taka 10GB data for 30days *212*901# | | 0 | 2 | .. | *212*901# , *212*901# , |
| 4 | | | Customer query was about *2# ussd code dialing... | 0 | 1 | .. | .. *2# |
| 5 | | as per customer voice he claim that he can't p... | Unable to purchase data pack by dial USSD *4# | 0 | 2 | .. | , *4# , *4# |
| 6 | | cx dial *2# but its shows requesting .vneror ... | Please ask Cx to check again, no issue found f... | 2 | 1 | , 06.10.2100 12.3009.10 , | , *2# , |
| | | his previous complaint, he facing | USSD not working no discrepancy | | | | |

Figure 4.37: Excel file read as Dataframe

The complaint file before populating it with the extracted values.

| | Summary | Description | Root Cause Detail | DateTime | USSD Code | DateTime | USSD Code |
|----|--|--|---|----------|-----------|--------------------|-----------|
| 1 | | | USSD not working related no problem found from our end. | 0 | 0 | | |
| 2 | | As per customer claim that he facing | Please ask cx to try again after restarting handset or cross check his handset. If pr | 0 | 0 | | |
| 3 | | As per customer claim that he want ! | As per HLR, cx already activate call waiting service. Please ask cx to dial *#43# an | 1 | 1 | 10/10/2021 11:35am | *43# |
| 4 | 100taka 10GB data for 30days *212*901# | 100taka 10GB data for 30days *212*901# | | 0 | 1 | | *212*901# |
| 5 | | | Customer query was about *2# ussd code dialing related.as per ussd code team f | 0 | 0 | | *2# |
| 6 | | as per customer voice he claim that | Unable to purchase data pack by dial USSD *4# | 0 | 1 | | *4# |
| 7 | | cx dial *2# but its shows requesting . | Please ask Cx to check again, no issue found from our end. if issue persists ask hi | 1 | 1 | 06.10.21 00.00.00 | *2# |
| 8 | | his previous complaint, he facing sai | USSD not working no discrepancy found from our end. | 0 | 1 | | |
| 9 | | As per customer claim that he facing | No issue found. Ask cx to try now. | 0 | 0 | | |
| 10 | | What is the error message subscribe | Ask CX to try now. No issue found. | 1 | 0 | | |
| 11 | | | | 0 | 0 | | |
| 12 | new sim offer 19 taka 2gb 7days. *121*887# | | | 1 | 0 | | |
| 13 | | As per customer voice he try to purc | As per system info, No dial attempt found on that mentioned timestamp. Cx's AI | 1 | 0 | | |
| 14 | | as per customer voice when customi | USSD manually checked *6# , *123# 12:06PM, USSD working fine, please ask cx to | 1 | 1 | | |
| 15 | | he got sms 1 GB 15 taka and dialed * | Customer did not get 15k 1GB due to insufficient balance. | 0 | 1 | | |
| 16 | | customer cannot buy data pack with | Customer Query was about *4# USSD code dial Related.No Problem found from c | 0 | 1 | | |
| 17 | | 65k 5GB *121*5138# BUT WHEN HE C | Customer did not get 65k 5GB due to not eligibility. | 0 | 1 | | |
| 18 | | *1600# code number not workingEn | There is nothing to check from our end. *1600# USSD information from EC server | 0 | 1 | | |
| 19 | | as per customer voice after dialing " | customer complaint ussd not working, we have checked no ussd code related pri | 0 | 1 | | |
| 20 | | He can not check his own number to | USSD is working fine. Please ask cx to crosscheck his handset and try again. If issu | 1 | 1 | | |
| 21 | | | As checked by concern, This USSD Code does not land to USSDGW or any other V. | 0 | 0 | | |
| 22 | | | No problem found about USSD Code related from our end, | 0 | 0 | | |
| 23 | | call waiting service ussd is not worki | No problem found from our end about call waiting service related. Customer ne | 0 | 0 | | |
| 24 | | He can not check his account balance | *222# and *1# USSD working fine, please ask cx to check again, | 0 | 1 | | |
| 25 | | As per customer voice his ussd coad | As per ICMS feedback, This MSISDN is not eligible for that offer, as CX is not inclu | 0 | 1 | | |
| 26 | | | Checked multiple times. USSD *2# working properly. Response time is 7 seconds | 0 | 0 | | |

Figure 4.38: Raw Complaint File

The complaint file after populating it with the extracted values.

| A | B | C | D | E | F | G | H | I | J | K | L |
|----|---------|------------------|---|----------|-----------|------------------------|---|---|---|---|---|
| | Summary | Description | Root Cause Detail | DateTime | USSD Code | DateTime.1 | USSD Code.1 | | | | |
| 1 | | | | | | | | | | | |
| 2 | 0 | | USSD not working relate | 0 | 0 | | | | | | |
| 3 | 1 | | As per customer c Please ask cx to try again | 0 | 0 | | | | | | |
| 4 | 2 | | As per customer c As per HLR, cx already ac | 1 | 1 | 10/10/202111, | *43# | | | | |
| 5 | 3 | 100taka 10GB da | 100taka 10GB data for 30days *212*901# | 0 | 2 | | *212*901# , *212*901# | | | | |
| 6 | 4 | | Customer query was abo | 0 | 1 | | *2# | | | | |
| 7 | 5 | | as per customer v Unable to purchase data | 0 | 2 | | *4# , *4# | | | | |
| 8 | 6 | | cx dial *2# but its Please ask Cx to check ag | 2 | 1 | 06.10.2100 12.3009.10, | *2# | | | | |
| 9 | 7 | | his previous comg USSD not working no dis | 0 | 1 | | *778*00# | | | | |
| 10 | 8 | | As per customer c No Issue found. Ask cx tc | 0 | 0 | | | | | | |
| 11 | 9 | | What is the error i Ask CX to try now. No iss | 0 | 6 | | *2# *222# *3# *2# *222# *3# | | | | |
| 12 | 10 | new sim offer 19 | taka 2gb 7days. *121*887# | 0 | 1 | | *121*887# | | | | |
| 13 | 11 | | As per customer v As per system info, No d | 0 | 1 | | *2*7*3# | | | | |
| 14 | 12 | | as per customer v USSD manually checked | 1 | 7 | 05/10/217, | *6# , *6# , *121# , *6# , *121# , *6# , *121# , *6# , *6# , *12 | | | | |
| 15 | 13 | | he got sms 1 GB ! Customer did not get 15t | 0 | 1 | | *212*707# | | | | |
| 16 | 14 | | customer cannot l Customer Query was abc | 0 | 2 | | *4# , *4# | | | | |
| 17 | 15 | | 65tk 5GB *121*51: Customer did not get 65t | 0 | 1 | | *121*5138# | | | | |
| 18 | 16 | | *1600# code numl There is nothing to check | 0 | 2 | | *1600# , *1600# | | | | |
| 19 | 17 | | as per customer v customer complaint ussc | 0 | 0 | | | | | | |
| 20 | 18 | | He can not check l USSD is working fine. Pei | 0 | 1 | | *2# | | | | |
| 21 | 19 | | As checked by concern, T | 0 | 0 | | | | | | |
| 22 | 20 | | No problem found about | 0 | 0 | | | | | | |
| 23 | 21 | | call waitine servic No problem found from | 0 | 1 | | *43# | | | | |

Figure 4.39: Complaint File after populating it with USSD Code and Date

Chapter 5

Learnings from the Internship

5.1 Professional Growth

Internship is a great opportunity for students to get an idea of the professional work. I learned a lot from this journey. Starting from creating resumes, going through interviews, attending meetings, understating what my instructors are expecting as outputs etc. I barely had any idea about automation. Now I am able to make projects on these concepts. Each of these learnings made me confident about myself and my works.

5.2 Team Work

Now in every industries, people are encouraging working in teams. I worked with three amazing instructors who guided me throughout this internship period and always encouraged me to learn new concepts. After working with them, I overcame my fear of talking with new people and adjusting in new environment.

5.3 Time management

I did this internship as an undergrad student. So, I had to manage both my studies and office work. Honestly, it was not easy to maintain these two task parallelly. But I was able to manage time for both of these tasks and complete everything perfectly. I followed the deadlines for each deliverables and showcased my works on time.

5.4 Technical Growth

As I said earlier, I learned a lot throughout this whole internship journey. I want to highlight some points regarding my technical learnings:

5.4.1 RPA

Report Automation was a new term for me. I did not have any idea about Web Bot and web scraping. I really enjoyed a lot doing these RPA based automation projects. I was able to automate Robi's e-approval website and help them to generate automated reports.

5.4.2 Git

As we worked in team, so it was important to showcase my updates of the codes to my instructors. For this we used Git and I learned how to properly use Git and sync the codes.

5.4.3 Docker

I had to use many third party libraries for my projects which creates dependency while deploying those projects. I was introduced to docker, so that I can solve this dependency issues. I learnt how to create image, container, docker file and etc.

5.4.4 Documentation

I had to document my every projects. I learnt how to create a README file which contains the setup and configuration guide. I also learnt how to portray the whole project in service diagram. It is very important to document the projects so that other people can understand what I have done in the project and how to use it in their system.

Chapter 6

Conclusion

6.1 Conclusion

Learnings become more enjoyable when we can relate our theoretical concepts to practical life. Internship gives that opportunity to make our learnings more realistic and gives us chance to experience the feeling of working as an employee in a renowned company. I feel really lucky that I got a chance to share the same platform with some amazing instructors who supported me throughout this whole journey. In these four months of my internship, I learnt many basic concepts about automation, microservices and many more. In large companies, it is really time consuming to manually handle any data. Automation is the best way to reduce manual work and make the whole process faster. I learned about microservices which was a whole new concept for me. It really needs patience and hard work to handle both workload and studies. Sometimes, we need to search through the resources a lot to get an idea about an unknown topic. By doing this, we can come across many other new concepts which really helps to develop our thinking. We can try to solve a problem in various different ways.

I really want to conclude by thanking my Line manager Syed Hasan Ibne Hossain, my instructors Sakibul Islam and S.M. Yeaminul Islam for guiding me and made me able to do such amazing projects. I also like to thank BRAC University for introducing the concept of doing internship instead of thesis for the students from CSE background.

6.2 Recommendations

Robi Axiata Limited is providing a great working opportunity for young talented people to begin their experience in a professional field. Everything about the company is fantastic, from the office environment to the amenities to the work ethics. I would highly recommend Robi to anyone interested in doing an internship in such an amazing environment while learning new skills from some incredible executives.

Bibliography

- [1] “Company profile.” (2022), [Online]. Available: <https://www.robi.com.bd/en/corporate/company-profile>. (accessed: 07- Jan- 2022).