PROJECT REPORT

Automated Helpdesk Ticketing System

Sarah Martin
C00257967

Contents

1. Introduction.. 3
1.1 Challenges of Manual Ticket Assignment.. 3
1.2 Automation Solution Overview.. 3
2. Project Inception...4
2.1 Research Phase... 4
2.2 Automation & UiPath..5
2.2.1 Automation Anywhere Vs UiPath..5
2.3 Coding Languages... 5
3. Development Planning Phase.. 6
3.1 Key Functionality.. 6
3.2 Design Considerations.. 6
4. Development Phase... 8
4.1 Keyword Creation...10
4.2 Categorisation UiPath Bot.. 11
4.3 Ticket Assignment UiPath Bot... 14
5. Reflection..17
5.1 Problems encountered... 17
5.2 Lessons Learned.. 18
5.3 Skill development.. 19
6. Conclusion..20
7. References.. 20

Fig 1 Field names CategoryID, DepartmentID and Keyword from the CategoryKeywords table. 8
Fig 2 . CategoryID 1 (Account Expired) category/queue keywords list. 9
Fig 3. The categories table that contains the fields CategoryID, CategoryName and DepartmentID (foreign key) 9
Fig 4. The department table that contains the fields DepartmentID and DepartmentName 9
Fig 5. Retrieving TicketID, Description and DateCreated for tickets in the tickets table. 12
Fig 6. Retrieving CategoryID, DepartmentID and Keyword for categories in the CategoryKeywords table. 12
Fig 7. Keyword comparison and match found code. 13
Fig 8. TicketID 174 has been successfully categorised and given the ID 1 (Account Expired). 13
Fig 9. RoleID, RoleName and DepartmentID (Foreign key) from Roles table. 14
Fig 10. CategoryID (foreign key) and RoleID (foreign key) from CategoryRoleMapping table.14
Fig 11. The script begins by querying the database to retrieve tickets that have been assigned categories. The query selects the TicketID and CategoryID from the tickets table where the CategoryID is not null. 15
Fig 12. CategoryID is being extracted from the current row and uses it to query the CategoryRoleMapping table to retrieve the associated role IDs for that category 16
Fig 13. AssignedRoleID field gets updated with the users role ID and the CategoryID gets updated to the category ID e.g if the matched category 1 was one then it belongs in the Account management department under the Account Expired category. 16

1. Introduction

This project aims to develop an automated ticketing system that is designed to streamline and enhance the management of helpdesk ticketing issues within an organisation. The current landscape of many organizational settings often involves manual ticket assignment processes, leading to various challenges for users.

1.1 Challenges of Manual Ticket Assignment

One significant challenge arises from human errors in ticket allocation. Within a manual ticketing system, users are allowed to allocate their tickets to any category (queue) they want to, this can lead to mistakes such as assigning tickets to incorrect categories(queues). These errors can result in delays in ticket resolution as tickets placed in the wrong category have to be reassigned to the correct category to be resolved.

1.2 Automation Solution Overview

Another significant challenge arises from the time consuming process of determining the appropriate category for incorrectly categorised tickets. When a ticket is assigned to the wrong category, it falls upon the personnel assigned to that ticket to assess the ticket's nature and decide on the appropriate course of action, including reassigning it to the correct category if necessary. Based on my research, this can significantly prolong the resolution process, as communication among teams is often required to determine the correct assignment for the ticket.

By implementing an automated ticketing system, we can effectively address these challenges. With this system in place, users will not be allowed to select a category for their ticket instead the system will automate ticket assignment using Workflow automation with UiPath Bots (1) (see Section 2.2). By defining predefined keywords for each category and utilizing these keywords to determine the appropriate category for each ticket, the system will ensure accurate and efficient ticket allocation. Additionally, a bot will be in place to help the assignment of tickets to the appropriate personnel. This will help with reducing the likelihood of human errors occurring and help with the resolution process.

2. Project Inception

2.1 Research Phase

During the initial stages of brainstorming ideas for my 4th-year project, I decided to explore automation solutions aimed at enhancing user efficiency with their daily tasks. Conversations with friends working in IT departments highlighted their frustrations with inefficient helpdesk ticketing systems. During these conversations, I discovered that most help desk ticketing systems they encountered often led to slow resolution. Often this was due to human error as sometimes users are uncertain about what to classify their ticket as and will pick a category that they believe their ticket belongs in.

Most of my research for my project stems from a technical officer who is employed by a government body and who has experience on a helpdesk. Within their helpdesk ticketing system, a user submits a ticket but often the technical officer finds he has to end up reassigning it as it doesn't belong in the categories/queues he is in charge of. The reassignment process is frustrating and time-consuming as he needs to locate the correct personnel who oversees the ticket.

This realisation prompted the idea of automating a helpdesk ticketing system, perhaps if we could limit human intervention and allow an automated system to do most of the heavy lifting then the assignment and resolution of tickets wouldn't be as slow for all users of the system.

After conversing back and forward, I brought forward the idea of automating the system he uses. I asked him various questions such as: what would you change about your system? How would you handle the assignment of tickets? Would you change the reassignment process?

After questioning him, he provided me with information to help build an efficient system. He agreed that automating a ticketing system would be beneficial and agreed that having something in place to do the categorisation and then do the assignment of a ticket to a queue to help user efficiency. Even with an automated system in place, there is still a chance that tickets can potentially be placed into the wrong queue so he proposed leaving in manual reassignment of a ticket in case human intervention is required at some point. He suggested creating a separate category for tickets that could not be assigned automatically, this way all users of the system can see which tickets could not be assigned correctly.

With this information from my research, I decided to plan out how I was going to implement a system that is beneficial for all users. After the initial research phase of determining the project scope, I then had to decide how exactly I was going to implement my project. I then decided to use the tools and languages I used during my internship to carry out the creation of the project.

2.2 Automation & UiPath

While doing an internship in my third year of university, I worked as a software engineer intern in the Automation Exchange team under the Automation department at Dell Technologies. The team I worked with was tasked with designing and implementing a website to host automated bots used by departments within the organization. The two main automated tools used were Automated Anywhere (2)and UiPath (3).

2.2.1 Automation Anywhere Vs UiPath

Automation Anywhere and UiPath are both robotic process automation (RPA). They help organisations automate repetitive tasks that are normally performed by humans. These tasks can range from data entry and document processing to customer service interactions and IT support. Both RPA platforms provide the tools and capabilities to develop, manage, and deploy software robots to help organisations improve efficiency, reduce human error and free up employee's time to focus on other tasks at hand.

With the knowledge of what RPA is I decided to choose RPA as the tool to use for my automated ticketing system. Since I had to pick one platform for my project, I compared the differences. Although Automation Anywhere had many pros such as built-in artificial intelligence technologies, I decided to use UiPath due to a couple of factors. UiPath is often considered more user friendly and has a visual workflow designer that allows users to create automation workflows through a drag and drop interface. From researching, users of UiPath found bots offer faster execution times and better resource management, which is beneficial for creating a ticketing system that may deal with high volume automation tasks. I noticed that UiPath was the more popular RPA tool used in Dell. When observing the departments that used RPA the most, the finance department came out on top when it came to utilising UiPath's capabilities in comparison to any other department within the organisation.

2.3 Coding Languages

During my internship, I also learned new coding languages. Most of my time was spent working with C# in the backend of an application. I was also introduced to Angular briefly, I didn't get to spend a lot of time working with Angular, but I took training courses to help with learning the language and did pair programming sessions with team members.

I decided to create my project utilizing C#, Angular and UiPath Studio.

3. Development Planning Phase

The next step was to determine the key functionality and design considerations for the automated ticketing system. This involved thinking about various aspects such as ticket categorisation, role assignment, API requirements, and front-end design.

I decided to outline the key functionality and design aspects required for implementing the system.

3.1 Key Functionality

Ticket Categorisation

- Define the criteria for categorising tickets based on keywords associated with each category/queue.
- Implement a system that automatically assigns categories to tickets to streamline the process of ticket resolution.

Category/Queue Assignment

- Create a system that allows routing tickets to the appropriate category/queue based on their categorisation.

User-Controlled Ticket Management

- Design features that allow users to create, view and manage their tickets effectively.

3.2 Design Considerations

User-Friendly Interface

- Create UI elements for ticket creation, viewing, and management

Ticket Attributes

- Define fields for subject, priority, and description to capture information.

Ticket Viewing

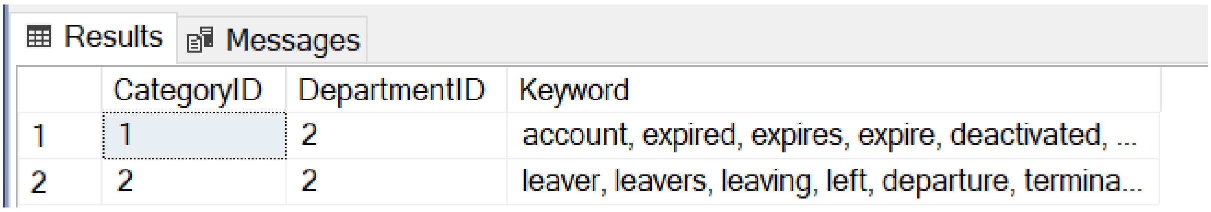
- Implement filters and sorting options for users to access their own tickets and those assigned to them.

4. Development Phase

After gathering the project requirements, I started the development process by setting up my development environments. For the backend, I opted for Visual Studio with C#, while for the front end, I chose Visual Studio Code paired with Angular. Given my recent familiarity with Microsoft SQL Server during my internship, I selected it to use for my database. I also configured UiPath to meet the project's requirements, ensuring that the bots operated as unattended bots, capable of running autonomously in the background without human intervention.

The most significant aspect of the development process revolved around implementing the automation component of the system. Initially, I faced challenges in conceptualizing how the automation would integrate seamlessly. To address this, I devised a solution involving the utilization of keyword recognition. I created table in the database called **CategoryKeywords** to store category keywords, opting for a comma-separated format to accommodate the potential for many keywords per category.

CategoryKeywords Table



	CategoryID	DepartmentID	Keyword
1	1	2	account, expired, expires, expire, deactivated, ...
2	2	2	leaver, leavers, leaving, left, departure, termina...

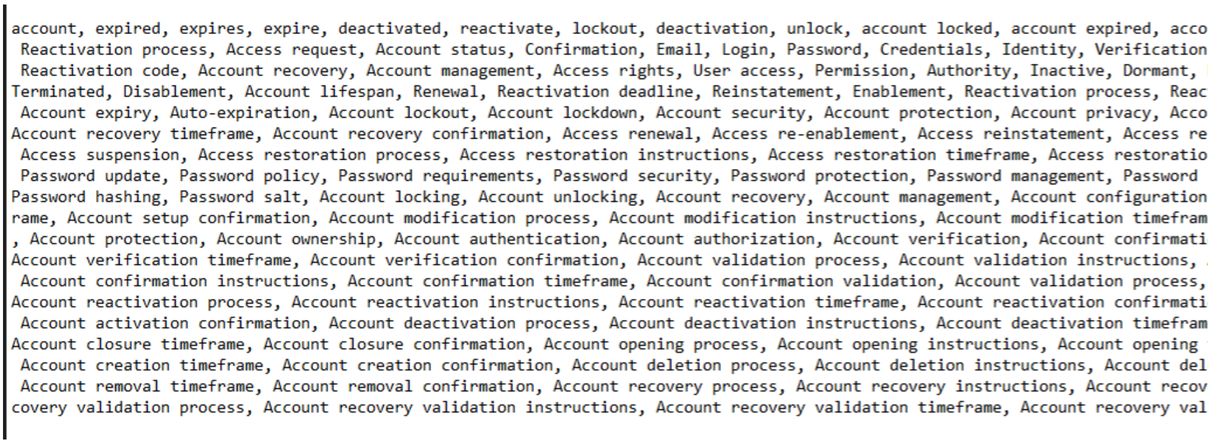
Fig 1 Field names CategoryID, DepartmentID and Keyword from the CategoryKeywords table.

In Fig. 1, we observe the layout comprising two distinct categories. CategoryID 1 represents the 'Accounts Expired' category, while CategoryID 2 represents the 'Leavers' category. Each category is associated with specific keywords. For instance, in my prototype, for the Business Support department, I have a total of six categories, and for the Account Management Department, I have two categories. Both departments and their corresponding category IDs are seen in Fig. 3 and the list of departments is seen in Fig. 4.

It's worth noting that each organization can create its own categories to reflect the range of help desks needed. This flexibility allows organisations to tailor the system according to their specific needs.

CategoryID and DepartmentID are foreign keys, sourced from the **Categories** and **Department** tables.

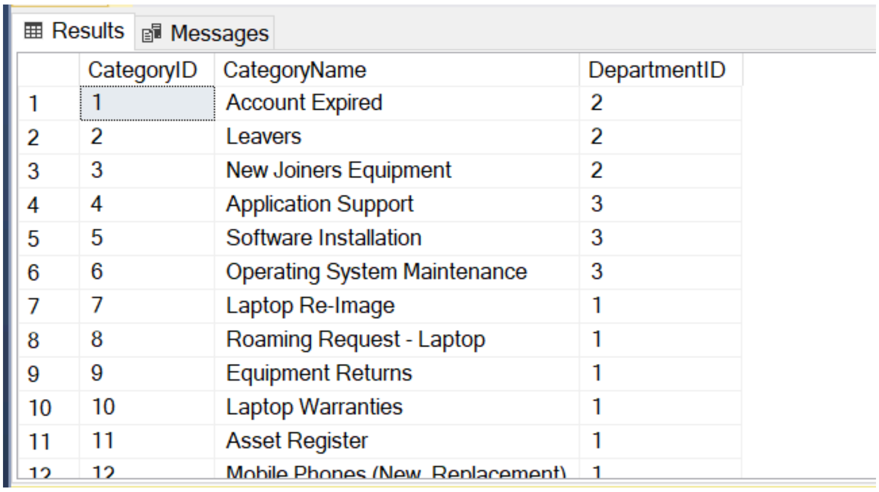
The keywords column in Fig. 1 contains all the keywords associated with each category. For example, some of the keywords associated with the CategoryID 2 (Leavers) keywords list:



```
account, expired, expires, expire, deactivated, reactivate, lockout, deactivation, unlock, account locked, account expired, acco
 Reactivation process, Access request, Account status, Confirmation, Email, Login, Password, Credentials, Identity, Verification
 Reactivation code, Account recovery, Account management, Access rights, User access, Permission, Authority, Inactive, Dormant,
Terminated, Disablement, Account lifespan, Renewal, Reactivation deadline, Reinstatement, Enablement, Reactivation process, Reac
 Account expiry, Auto-expiration, Account lockout, Account lockdown, Account security, Account protection, Account privacy, Acco
Account recovery timeframe, Account recovery confirmation, Access renewal, Access re-enablement, Access reinstatement, Access re
 Access suspension, Access restoration process, Access restoration instructions, Access restoration timeframe, Access restoratio
 Password update, Password policy, Password requirements, Password security, Password protection, Password management, Password
Password hashing, Password salt, Account locking, Account unlocking, Account recovery, Account management, Account configuration
rame, Account setup confirmation, Account modification process, Account modification instructions, Account modification timefram
, Account protection, Account ownership, Account authentication, Account authorization, Account verification, Account confirmati
Account verification timeframe, Account verification confirmation, Account validation process, Account validation instructions,
 Account confirmation instructions, Account confirmation timeframe, Account confirmation validation, Account validation process,
Account reactivation process, Account reactivation instructions, Account reactivation timeframe, Account reactivation confirmati
 Account activation confirmation, Account deactivation process, Account deactivation instructions, Account deactivation timefram
Account closure timeframe, Account closure confirmation, Account opening process, Account opening instructions, Account opening
 Account creation timeframe, Account creation confirmation, Account deletion process, Account deletion instructions, Account del
 Account removal timeframe, Account removal confirmation, Account recovery process, Account recovery instructions, Account recov
covery validation process, Account recovery validation instructions, Account recovery validation timeframe, Account recovery val
```

Fig 2 . CategoryID 1 (Account Expired) category/queue keywords list.

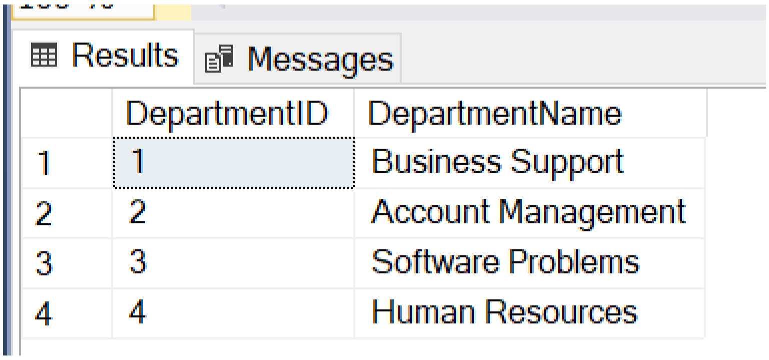
Categories Table



	CategoryID	CategoryName	DepartmentID
1	1	Account Expired	2
2	2	Leavers	2
3	3	New Joiners Equipment	2
4	4	Application Support	3
5	5	Software Installation	3
6	6	Operating System Maintenance	3
7	7	Laptop Re-Image	1
8	8	Roaming Request - Laptop	1
9	9	Equipment Returns	1
10	10	Laptop Warranties	1
11	11	Asset Register	1
12	12	Mobile Phones (New, Replacement)	1

Fig 3. The categories table that contains the fields CategoryID, CategoryName and DepartmentID (foreign key)

Department Table



	DepartmentID	DepartmentName
1	1	Business Support
2	2	Account Management
3	3	Software Problems
4	4	Human Resources

Fig 4. The department table that contains the fields DepartmentID and DepartmentName

4.1 Keyword Creation

The creation of the keyword list was influenced by two primary factors: end-user testing and the usage of ChatGPT for keyword generation.

To create the keyword list, I used end-user testing by requesting individuals to generate sample tickets. On purpose, I provided a vague scenario involving a user's account expiration or lockout to observe how users would describe their issues. Despite the minimal information provided, testers tended to provide detailed ticket descriptions. Some testers included additional information such as fake names and departments.

I leveraged ChatGPT to generate additional keywords based on the sample ticket descriptions. This approach aimed to increase the keyword list with relevant terms and phrases taken from the ticket descriptions.

I presented an example scenario to end-user testers involved in the task of creating a ticket to address a user account expiration or lockout. Testers were instructed to focus solely on creating the ticket description, which resulted in varied descriptions.

Example scenario:

For example, you need to create a ticket in regard to gaining access to your account again. You try to log in one morning and your account is locked as it won't let you log in and you are not sure why other than a message popping up that your account has expired. It's urgent that you get it fixed or you can't do your job.

End User Tickets:

Ticket No.	Ticket Description
1	Hello, My name is Jim Kerry, and I am the manager for the Quality Assurance section and my employee John Duffy. Cannot get into his account as he keeps getting a message saying his “account has been locked due to inactivity”. Is there any way to solve this issue? His details are: Email: john.duffy@qas.ie Username: john.duffy@qas.ie Kind Regards, Jim Kerry
2	Account Locked Section: ICT Support User: John Frank Email: John.Frank@aud.ie Regards, Pat
3	User cannot log into account Username : John Frank Section: Helpdesk Support Error Message said account expired
4	Hello, Staff member Caoimhe Byrne has returned from Maternity Leave but her account has been deactivated due to inactivity. Would you be able to reactivate her old account or will I need to make a request for a new one? Thank you for anything you can do to help, Cynthia
5	To whom it may concern, My colleague Brenda Cahill has asked me to contact you on her behalf as her account is locked and is giving a message saying your account has expired due to inactivity. Would you be able to call her at 08777777777 and let her know how to fix this. Thank you so much, Jonathan
6	Hello, My employee’s account seems to have expired despite being in active use. I would like the account to be enabled immediately as work is required to continue. Regards, Kieth

After analysing the testers tickets, I observed a recurring pattern whereusers tended to create tickets with different phrasing but containing common keywords. This then allowed me to leverage ChatGPT to generate additional keywords associated with the 'Accounts Expired' category (I followed a similar process for other ticket categories when it came to their creation). I provided ChatGPT with a specific prompt, instructing it to generate keywords based on example tickets relevant to a helpdesk ticketing system, specifically targeting the 'Accounts Expired' category.

ChatGPT Prompt:

"Here is some tickets associated with a helpdesk ticketing system. These tickets belong in a queue/category called the Accounts Expired category. I want tickets to be automatically categorised based on certain keywords in each ticket e.g if a ticket said anything to do with accounts expired, locked, issue with account then it belongs in the Accounts Expired category". Help me generate one hundred or more keywords that you think a user would use when creating a ticket that should belong in this queue".

I then began the creation of my ticket categorisation bot.

4.2 Categorisation UiPath Bot

With the keywords in hand, I began with developing the categorisation bot. I initiated the coding process by configuring UiPath Studio to connect to Microsoft SQL Server.

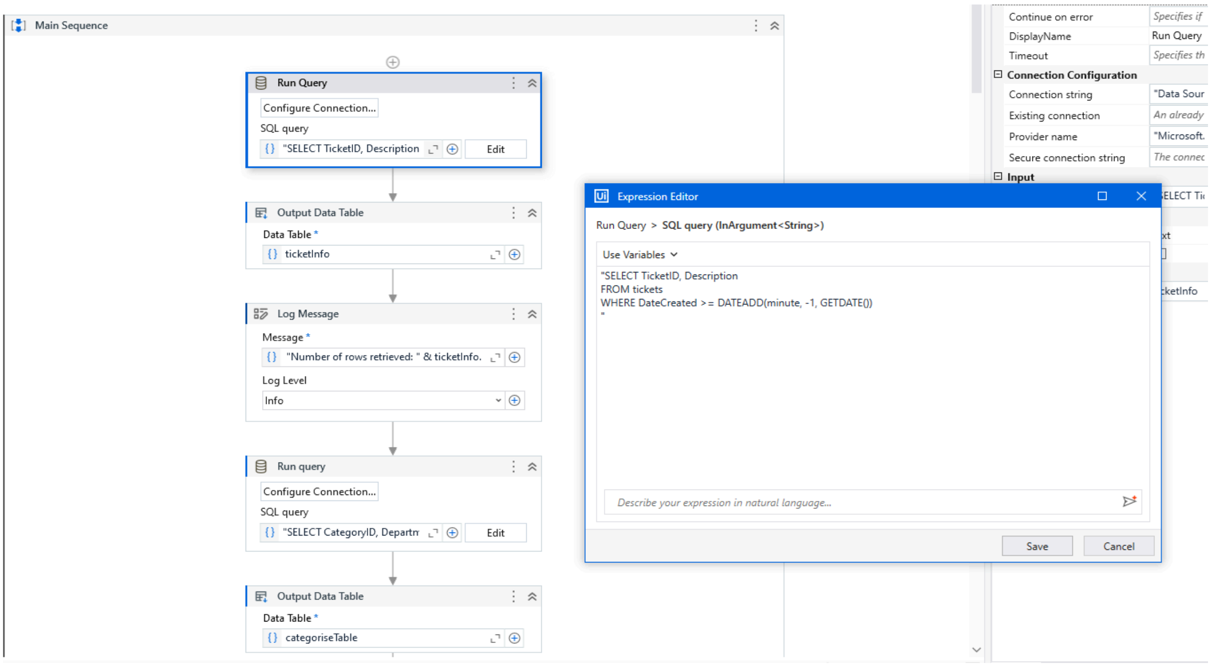
To do the categorisation process, I incorporated data from the **categoriseKeywords** table, which included categoryID, departmentID, and Keyword fields.

I then implemented code to compare each keyword from the ticket description, based in the tickets table, with the keywords stored in the **categoriseKeywords** table. To ensure accurate matches, the ticket description was required to contain three or more keywords. This approach was refined based on feedback from users who provided example tickets so that in case of category overlap, we can ensure that a ticket has a higher chance of going to the right category.

Additionally, I set up the bot to run every minute, considering the project's context and the manageable ticket volume. I deemed that this was sufficient for processing newly created tickets as new tickets are not being created every second.

After a ticket has been categorised, the CategoryID column in the tickets table updates with the matched category ID.

I will provide snippets of code for the categorisation bot below to just get an idea of how it operates.



Main Sequence

Run Query
Configure Connection...
SQL query: "SELECT TicketID, Description

Output Data Table
Data Table *: ticketInfo

Log Message
Message *: "Number of rows retrieved: " & ticketInfo.
Log Level: Info

Run query
Configure Connection...
SQL query: "SELECT CategoryID, Departm

Output Data Table
Data Table *: categoriseTable

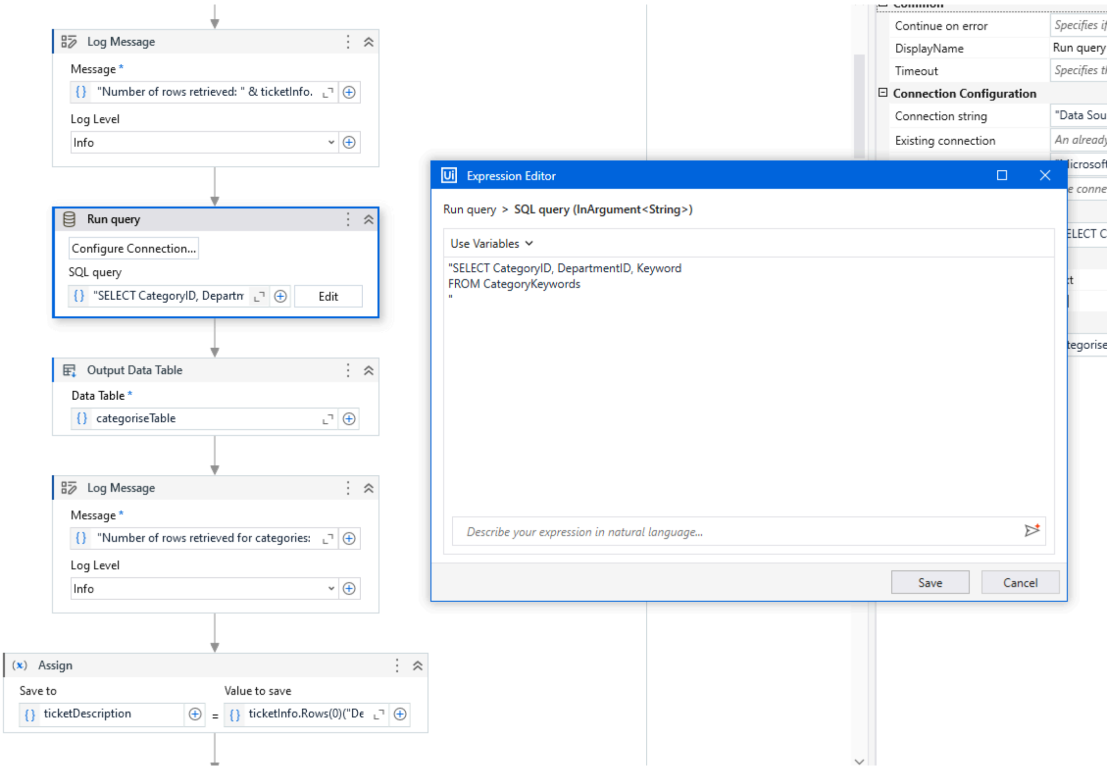
Continue on error	Specifies if
DisplayName	Run Query
Timeout	Specifies th
Connection Configuration	
Connection string	"Data Sour
Existing connection	An already
Provider name	"Microsoft.
Secure connection string	The connec
Input	

Expression Editor

Run Query > SQL query (InArgument<String>)

```
"SELECT TicketID, Description
FROM tickets
WHERE DateCreated >= DATEADD(minute, -1, GETDATE())
"
```

Fig 5. Retrieving TicketID, Description and DateCreated for tickets in the tickets table.



Log Message
Message *: "Number of rows retrieved: " & ticketInfo.
Log Level: Info

Run query
Configure Connection...
SQL query: "SELECT CategoryID, Departm

Output Data Table
Data Table *: categoriseTable

Log Message
Message *: "Number of rows retrieved for categories:
Log Level: Info

Assign
Save to: ticketDescription = Value to save: ticketInfo.Rows(0)("De

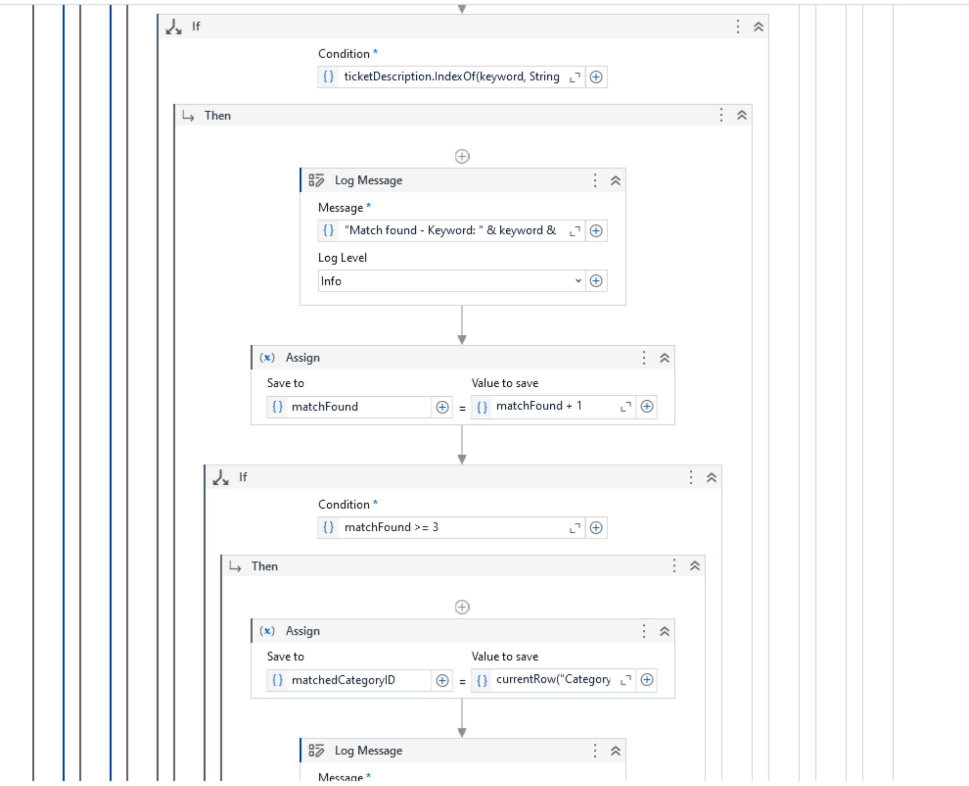
Continue on error	Specifies if
DisplayName	Run query
Timeout	Specifies ti
Connection Configuration	
Connection string	"Data Sou
Existing connection	An already

Expression Editor

Run query > SQL query (InArgument<String>)

```
"SELECT CategoryID, DepartmentID, Keyword
FROM CategoryKeywords
"
```

Fig 6. Retrieving CategoryID, DepartmentID and Keyword for categories in the CategoryKeywords table.



If

Condition *

ticketDescription.IndexOf(keyword, String

Then

Log Message

Message *

"Match found - Keyword: " & keyword &

Log Level

Info

Assign

Save to: matchFound

Value to save: matchFound + 1

If

Condition *

matchFound >= 3

Then

Assign

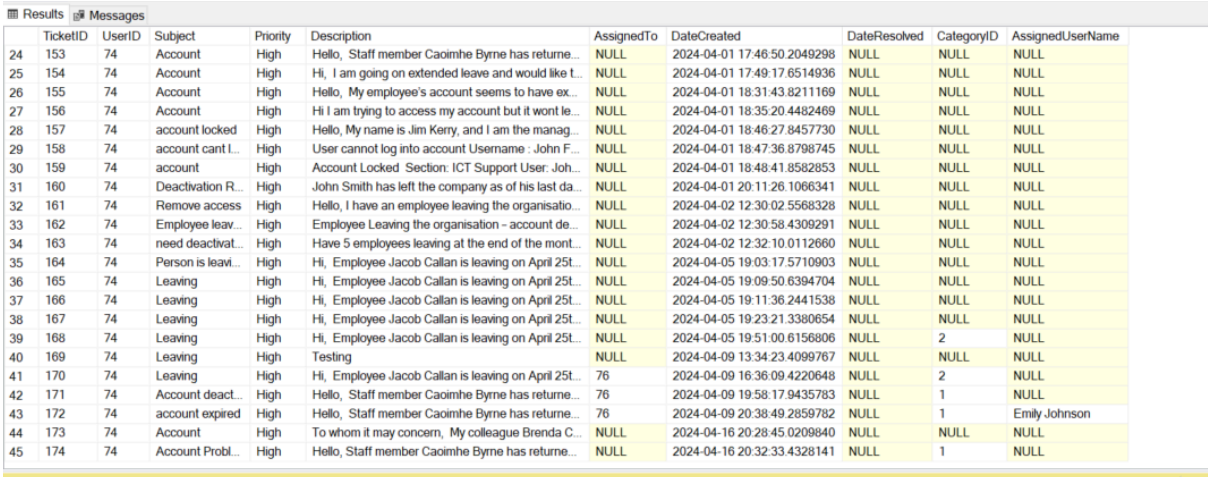
Save to: matchedCategoryID

Value to save: currentRow("Category

Log Message

Message *

Fig 7. Keyword comparison and match found code.



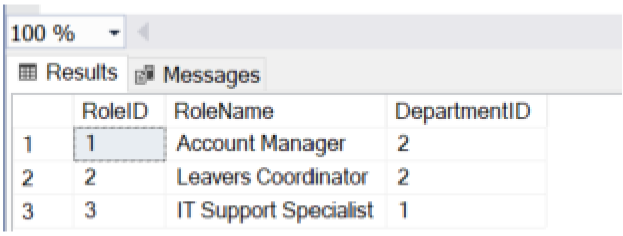
	TicketID	UserID	Subject	Priority	Description	AssignedTo	DateCreated	DateResolved	CategoryID	AssignedUserName
24	153	74	Account	High	Hello, Staff member Caoimhe Byrne has returne...	NULL	2024-04-01 17:46:50.2049298	NULL	NULL	NULL
25	154	74	Account	High	Hi, I am going on extended leave and would like t...	NULL	2024-04-01 17:49:17.6514936	NULL	NULL	NULL
26	155	74	Account	High	Hello, My employee's account seems to have ex...	NULL	2024-04-01 18:31:43.8211169	NULL	NULL	NULL
27	156	74	Account	High	Hi I am trying to access my account but it wont le...	NULL	2024-04-01 18:35:20.4482469	NULL	NULL	NULL
28	157	74	account locked	High	Hello, My name is Jim Kerry, and I am the manag...	NULL	2024-04-01 18:46:27.8457730	NULL	NULL	NULL
29	158	74	account cant l...	High	User cannot log into account Username : John F...	NULL	2024-04-01 18:47:36.8798745	NULL	NULL	NULL
30	159	74	account	High	Account Locked Section: ICT Support User: Joh...	NULL	2024-04-01 18:48:41.8582853	NULL	NULL	NULL
31	160	74	Deactivation R...	High	John Smith has left the company as of his last da...	NULL	2024-04-01 20:11:26.1066341	NULL	NULL	NULL
32	161	74	Remove access	High	Hello, I have an employee leaving the organisatio...	NULL	2024-04-02 12:30:02.5568328	NULL	NULL	NULL
33	162	74	Employee leav...	High	Employee Leaving the organisation - account de...	NULL	2024-04-02 12:30:58.4309291	NULL	NULL	NULL
34	163	74	need deactivat...	High	Have 5 employees leaving at the end of the mont...	NULL	2024-04-02 12:32:10.0112660	NULL	NULL	NULL
35	164	74	Person is leavi...	High	Hi, Employee Jacob Callan is leaving on April 25t...	NULL	2024-04-05 19:03:17.5710903	NULL	NULL	NULL
36	165	74	Leaving	High	Hi, Employee Jacob Callan is leaving on April 25t...	NULL	2024-04-05 19:09:50.6394704	NULL	NULL	NULL
37	166	74	Leaving	High	Hi, Employee Jacob Callan is leaving on April 25t...	NULL	2024-04-05 19:11:36.2441538	NULL	NULL	NULL
38	167	74	Leaving	High	Hi, Employee Jacob Callan is leaving on April 25t...	NULL	2024-04-05 19:23:21.3380654	NULL	NULL	NULL
39	168	74	Leaving	High	Hi, Employee Jacob Callan is leaving on April 25t...	NULL	2024-04-05 19:51:00.6156806	NULL	2	NULL
40	169	74	Leaving	High	Testing	NULL	2024-04-09 13:34:23.4099767	NULL	NULL	NULL
41	170	74	Leaving	High	Hi, Employee Jacob Callan is leaving on April 25t...	76	2024-04-09 16:36:09.4220648	NULL	2	NULL
42	171	74	Account deact...	High	Hello, Staff member Caoimhe Byrne has returne...	76	2024-04-09 19:58:17.9435783	NULL	1	NULL
43	172	74	account expired	High	Hello, Staff member Caoimhe Byrne has returne...	76	2024-04-09 20:38:49.2859782	NULL	1	Emily Johnson
44	173	74	Account	High	To whom it may concern, My colleague Brenda C...	NULL	2024-04-16 20:28:45.0209840	NULL	NULL	NULL
45	174	74	Account Probl...	High	Hello, Staff member Caoimhe Byrne has returne...	NULL	2024-04-16 20:32:33.4328141	NULL	1	NULL

Fig 8. TicketID 174 has been successfully categorised and given the ID 1 (Account Expired).

4.3 Ticket Assignment UiPath Bot

After creating the ticket categorisation bot, I began working on the assignment bot that routes tickets to the correct personnel. For this to work I have set up the following tables in my database:

Roles table

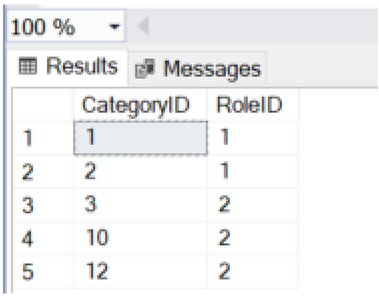


	RoleID	RoleName	DepartmentID
1	1	Account Manager	2
2	2	Leavers Coordinator	2
3	3	IT Support Specialist	1

Fig 9. RoleID, RoleName and DepartmentID (Foreign key) from Roles table.

This table contains role names, with each role being associated with a role ID and department ID. Every user is assigned a role ID for identification purposes. This structure helps in accurately assigning tickets to the relevant users, particularly as each role may oversee multiple categories. For instance, an Account Manager assigned the Role ID 1 may be responsible for overseeing categories/queues 1 and 2, such as the Account Expired and Leavers categories. To manage this association effectively, I created an additional table named **CategoryRoleMapping**.

CategoryRoleMapping Table

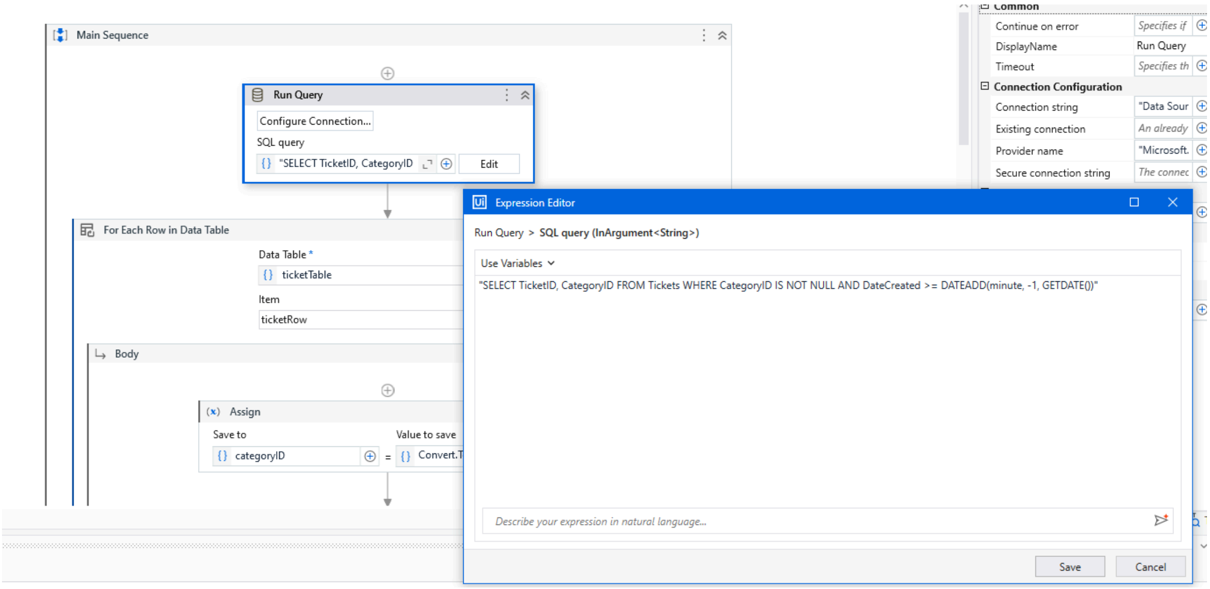


	CategoryID	RoleID
1	1	1
2	2	1
3	3	2
4	10	2
5	12	2

Fig 10. CategoryID (foreign key) and RoleID (foreign key) from CategoryRoleMapping table.

For the UiPath code, I decided to make it so that every time the CategoryID in the tickets table was updated less than a minute ago and it wasn't empty, the assignment bot will start its process.

The code fetches tickets with assigned categories, determines the role IDs associated with each category, retrieves users associated with those role IDs, and assigns the ticket to each user. This process ensures that tickets are assigned to appropriate users based on their roles and the categories associated with the tickets. I will provide code snippets below of this process.



SQL query: "SELECT TicketID, CategoryID

Data Table *: ticketTable

Item: ticketRow

Save to: categoryID = Value to save: Convert.T

```
"SELECT TicketID, CategoryID FROM Tickets WHERE CategoryID IS NOT NULL AND DateCreated >= DATEADD(minute, -1, GETDATE())"
```

Continue on error: Specifies if

DisplayName: Run Query

Timeout: Specifies th

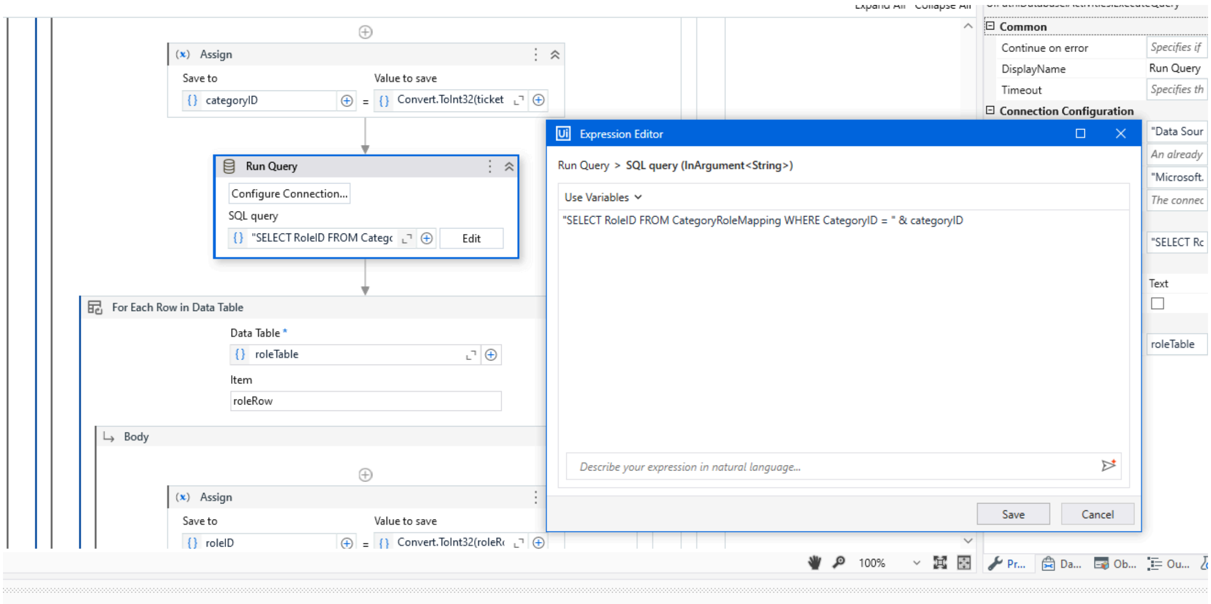
Connection string: "Data Sour

Existing connection: An already

Provider name: "Microsoft.

Secure connection string: The connec

Fig 11. The script begins by querying the database to retrieve tickets that have been assigned categories. The query selects the TicketID and CategoryID from the tickets table where the CategoryID is not null.



Assign: Save to: categoryID = Value to save: Convert.ToInt32(ticket

Run Query: SQL query: "SELECT RoleID FROM Catego

Run Query > SQL query (InArgument<String>)

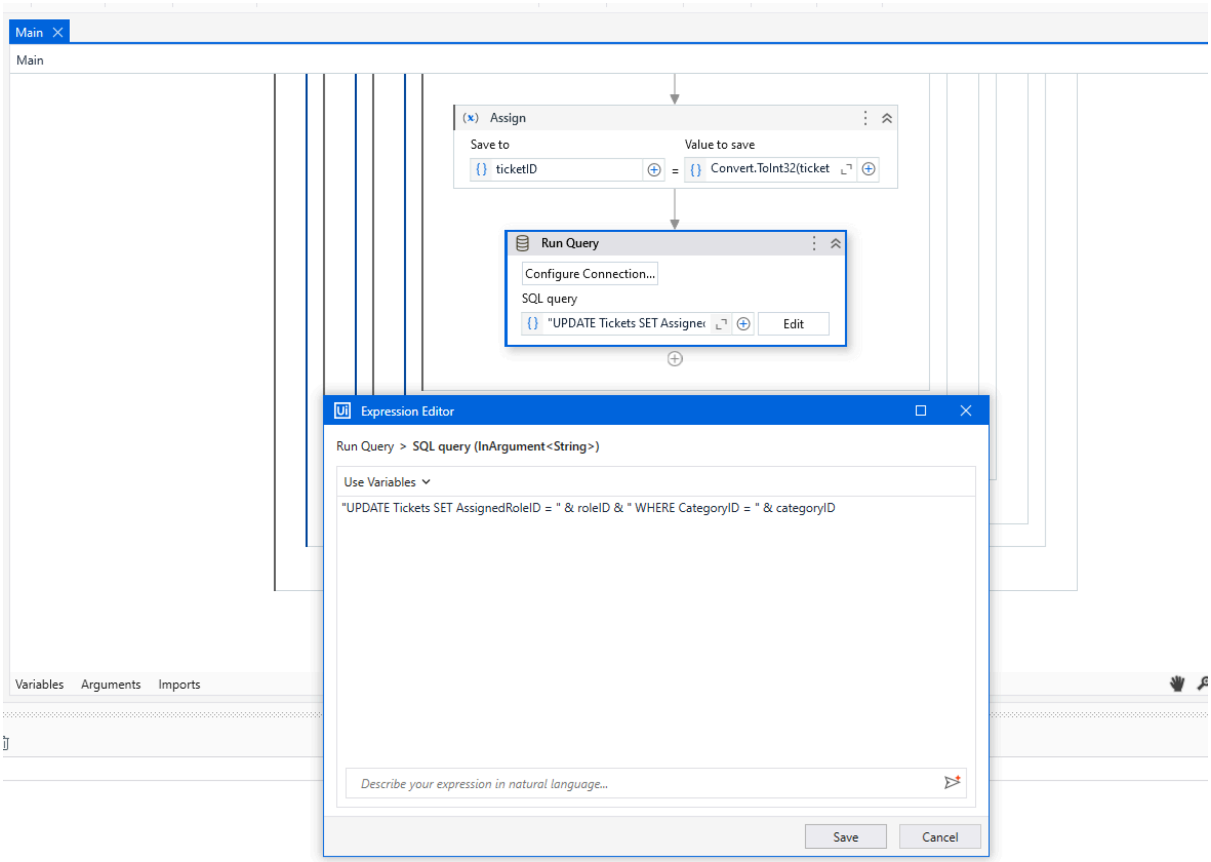
"SELECT RoleID FROM CategoryRoleMapping WHERE CategoryID = " & categoryID

For Each Row in Data Table: Data Table * roleTable; Item roleRow

Assign: Save to: roleID = Value to save: Convert.ToInt32(roleR

Continue on error: Specifies if
DisplayName: Run Query
Timeout: Specifies th
"Data Sour
An already
"Microsoft.
The connec
"SELECT Rc
Text ☐
roleTable

Fig 12. CategoryID is being extracted from the current row and uses it to query the CategoryRoleMapping table to retrieve the associated role IDs for that category



Assign: Save to: ticketID = Value to save: Convert.ToInt32(ticket

Run Query: SQL query: "UPDATE Tickets SET Assigne

Run Query > SQL query (InArgument<String>)

"UPDATE Tickets SET AssignedRoleID = " & roleID & " WHERE CategoryID = " & categoryID

Fig 13. AssignedRoleID field gets updated with the users role ID and the CategoryID gets updated to the category ID e.g if the matched category 1 was one then it belongs in the Account management department under the Account Expired category.

5. Reflection

Throughout the development of my project, I encountered several challenges, particularly in building my automated ticketing system prototype. I found this year to be overall challenging, I knew 4th year would be no easy task to tackle but I was certaintly tested when it came to the creation of this project and course modules. I also realised that coding was not my strongest skill and particularly the development part of this project brought on so many difficulties. Towards the beginning of year I had suffered some minor health problems and as the year went on, particularly towards the end of semester one I was deeply effected by these problems as I was chronically fatigued and struggled mentally to continue on the course at times. Unfortunately my project did suffer in aspects due to this and if I could have a redo of everything knowing what I know now, I would have taken a break from university. Although I did mostly enjoy creating a project from scratch, again as someone who does not have programming as their strongest skill I did enjoy the challenge from the initial phase of requirement gathering, interviewing technical employees to putting together what I consider something that could be potentially beneficial to users. My target from the beginning was to help make a system that anyone could use whether you are a tech-savvy employee working at Google to someone who may dabble in technology from time to time. I have no regrets choosing this project but I have encountered many technical problems. In the following subsections I will outline some of the more technical problems I had but also the skills and lessons learned.

5.1 Problems encountered

I will discuss the three main areas I worked on in detail.

Frontend & Backend

During the setup of the frontend, I grappled with integrating API calls to my backend. This aspect was relatively new to me, as my previous experience revolved around creating APIs solely in the backend during my work placement. While I had worked on basic CRUD apps in college in 2nd and 3rd year, the complexity of this task presented a significant learning curve. There were numerous instances where I found myself frustrated while attempting to debug errors appearing in my console. To diagnose these issues, I relied on Swagger to identify the root cause. It became evident that the backend was not receiving the necessary information, indicating a flaw in my API call implementation. Upon closer inspection, I discovered the error lay in the how the API was set up in the backend and after many frustrated attempts I did manage to fix the problem.

UiPath

Implementing the categorisation and assignment bots posed several challenges. Initially, I had to figure out the scripting for categorisation and devise a plan for integration with my system, followed by the creation of another bot to allocate tickets to the appropriate personnel.

The initial setup of UiPath proceeded smoothly, with no complications encountered in connecting it to my database to retrieve ticket information. However, challenges arose when attempting to read keywords from the categorisationKeywords table and compare them with the ticket description. I needed to devise a method to remove any whitespace and ensure accurate comparison of keywords separated by commas in the categorizationKeywords table. Storing each keyword from the table in a UiPath variable designed to handle table data caused significant trial and error, as discrepancies often arose in keyword comparisons with the ticket description. This led to misclassification of tickets, where they were routed to queues other than their intended target.

With fixing the issue of keyword separation and comparison accuracy, I encountered another obstacle related to overlapping keywords in different categories. For instance, the "Leavers" category belonged to employees leaving the company, while the "Accounts Expired" category belonged to expired employee accounts. Occasionally, tickets meant for one category were routed to another due to shared keywords. To fix this, I conducted end-user testing, wherein multiple individuals generated tickets based on given scenarios. Following this, I implemented a requirement that a ticket must contain three or more keywords for proper categorisation. This refinement significantly enhanced the accuracy of ticket categorisation and reduced the number of tickets going to the wrong personnel.

The assignment bot didn’t pose too much issues other than spending time trying to figure out how the logic of it should be implemented. Once I had the categorisation bot in place, I was able to use my knowledge from creating the categorisation bot to create my assignment bot. I did initially have some issues with the assignment bot not correctly reading my table data.

5.2 Lessons Learned

I have learned quite a lot from the development of this project. This was the first big solo project I have ever tackled on my own. In previous projects, I would have worked on a medium/small sized team. While on work placement, if you ran into difficulties you had your team members to bounce ideas off whereas in this regard, you only have yourself as well as the support (and moral support) of your project supervisor. I have learned that theres a lot more that goes into the development of a project than what I assumed from the beginning. I was quite naïve in thinking at the start that a project can be produced “overnight” but I learned quickly that it takes time and a lot of mental strength to figure out what is needed to firstly begin researching and then figuring out what coding languages are required, tools needed to develop a project. I have learned that research is good and its useful to get advice from people who work with helpdesk ticketing systems in organisations. I found it to be beneficial as it helped narrow my target goal for my project. In the future if Im ever tackling another project like this I will definitely be seeking the advice of proffesionals.

Ive also learned its better to start projects right from the very beginning. Sometimes I feel intimidated starting a project but I’ve learned its better to tackle it head on than wait as time is not on your side when it comes to meeting deadlines.

5.3 Skill development

When initially developing the project I had some prior knowledge of C# due to my internship as well as some minor knowledge of Angular. Thanks to the development of this project I have enhanced my understanding of both programming languages. I have mentioned previously that coding is not my strongest skill but I did find that I came out of this project with the ability of being able to put together a Angular front end that is integrated with a C# backend.

I also learned to set a Microsoft MySql database up. I also used this database on workplacement but I never interacted with setting up tables and logging info. I learned how to do mitigations in the backend to update the database tables which I never knew you could do. Back to Angular, on work placement this language did terrify me as it was initimating looking at the source code of the application my team was working on, but I found with now developing my own Angular frontend I can now comfortably set up an Angular project.

With UiPath, I never used it prior to this project. UiPath was something I heard was used significantly in my work placement. I didn’t realise a software existed that could mimic human interactions and save users time so they can focus on other tasks at hand. To me this software seemed useful to learn. Automation is quite a vast area and UiPath is just a small part of it but it being an RPA its very powerful. I also liked the drag and drop interface as I learned overtime that code was much easier to read when it was broken up into sections. At first learning how UiPath studio works and its many components I did find it difficult and struggled with knowing what component to use with which piece of code but overtime I developed an understanding of its interface and really enjoyed working with it. Learning UiPath and its abilities has been a valuable lesson and its something I will certainly be using again in the future.

6. Conclusion

To conclude this project and chapter, I set out with the task of developing an application designed to automate helpdesk ticketing systems. I had conversations with helpdesk employees to get a better understanding of their frustrations within the system they use in their work environment. With these frustrations known, I set out to build my automated ticketing system using the languages and tools I learned while in college and work placement. Over the course of the development process, I realised a few things, coding wasn't my strongest suit, creating a project from scratch was difficult and with being chronically fatigued I realised I was having a difficult time with development of the project overall due to my mental health. My main difficulties came from creating the categorisation bot and figuring out bugs that were causing a roadblock with the backend. However, I did create a system in the end which has a lot of areas for improvement but carries out the required task of categorising user tickets. I also learned a lot of valuable lessons and skills along the way and that is the main concept I am taking away from completing this project.

7. References

1. Anon, n.d. *Automation Anywhere - Soais.* [Online]
   Available at:
   https://soais.com/automation-anywhere/#:~:text=Automation%20anywhere%20tool%20employs%20software,offer%20even%20greater%20business%20value.

2. Inc, U., 2019. *Robotic Process Automation | UiPath.* [Online]
   Available at: https://www.uipath.com/

3. Inc, U., n.d. *Intelligent Software Robots - Unattanded and Attended Bot | UiPath.* [Online]
   Available at: https://www.uipath.com/product/robots

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Work submitted for assessment which does not include this declaration will NOT be assessed

DECLARATION

1. I declare that all material in this submission e.g., thesis/essay/project/assignment is entirely my own work except where fully acknowledged.
2. I have cited the sources of all quotations, paraphrases, summaries of information, Tables, diagrams, or other material; including software and other electronic media in which intellectual property rights may reside.
3. I have provided a complete bibliography of all works and sources used in the preparation of this submission.
4. I understand that failure to comply with SETU's regulations governing Plagiarism constitutes a serious offence.

Student Name (Printed): ____Sarah Martin_______

Student Number(s):
________C00257967__________________________

Student Signature(s):
_______________________Sarah martin________

Date: 19/4/2024