

# **Ticket Enrichment using Machine Learning & Automation Catalog Development**

*A project report submitted*

*to*

**MANIPAL ACADEMY OF HIGHER EDUCATION**

*For Partial Fulfillment of the Requirement for the*

*Award of the Degree*

*of*

**Bachelor of Technology**

*in*

**Information Technology**

*by*

**B N S VISHAL Reg. No. 160911122**

*Under the guidance of*

Mrs. Anuradha Rao

Assistant Professor Senior Scale  
Department of I&CT

Manipal Institute of Technology  
Manipal, India

Mr. Govindarajan V.S.

Director  
TOC - Strategy and Transformation  
UBS Business Solutions (India) Pvt. Ltd.  
Building-8, CommerZone,  
Yerwada, Pune - 411006



**MANIPAL INSTITUTE OF TECHNOLOGY**  
**MANIPAL**  
*(A constituent unit of MAHE, Manipal)*

July 2020

I dedicate my thesis to my friends and family.

## **DECLARATION**

I hereby declare that the projects entitled **Ticket Enrichment using Machine Learning and Automation Catalog Development** are original and has been carried out by me as an Intern in the Strategy and Transformation Automation Center of Excellence team at Technology Operations Center, UBS India under the guidance of Mr. Govindarajan VS, Head of Automation CoE, UBS, India and Mrs. Anuradha Rao, Assistant Professor Senior Scale , Department of Information and Communication Technology, M.I.T., Manipal. No part of this work has been submitted for the award of a degree or diploma either to this University or to any other Universities.

Place: Manipal

Date :15-07-2020

*Vishal*

B N S Vishal

160911122

IT- ‘A’, MIT

Manipal



# MANIPAL INSTITUTE OF TECHNOLOGY

## MANIPAL

(A constituent unit of MAHE, Manipal)

## CERTIFICATE

This is to certify that the projects entitled **Ticket Enrichment using Machine Learning and Automation Catalog Development** are bonafide project work done by **Mr. B N S VISHAL (Reg.No.:160911122)** at Manipal Institute of Technology, Manipal, independently under my guidance and supervision for the award of the Degree of Bachelor of Technology in Information Technology.

Mrs. Anuradha Rao  
Assistant Professor- Senior Scale  
Department of I&CT  
Manipal Institute of Technology  
Manipal, India

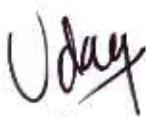
Dr.Balachandra  
Professor & Head  
Department of I&CT  
Manipal Institute of Technology  
Manipal, India

# Certificate of Completion

This certificate is awarded to

**Vishal B N S**

For successfully completing the 2020 Technology Internship Program at UBS India from 5<sup>th</sup> February 2020 to 30<sup>th</sup> June 2020.



Uday Odedra  
Managing Director,  
Head of APAC Technology



Ashish Mishra  
Executive Director,  
Tech India Graduate Program Lead

June 2020

To whom it may concern

Vishal BNS joined the Strategy & Transformation Automation Center of Excellence team as an intern in Feb' 2020 & served us 5 months. During his internship program, he demonstrated full commitment & delivered the assigned projects on time.

He worked on below 2 projects primarily along with delivering few sessions for the whole team based on his new learnings.

1. Automation Catalog – this is a highly visible project across the organization. This project is focused on developing a market place within the organization to showcase already developed automations to encourage reuse and accelerate delivery.

Vishal had picked up Angular & JS skills within a short period and became an expert in that technology

He was expected to lead the development of Automation Catalog V3 & he delivered the product over and beyond our expectations before the deadline

His commitment and passion towards learning new skills & applying them in work environment helped a lot to deliver the new version on time

Apart from the defined features, he was also tasked to perform a market research on similar products (Amazon Market Place, AA Bot store, etc). Based on his research outcome, couple of new features added to the catalog which were very much appreciated by stakeholders.

2. Supervised Machine Learning solution for batch failures – this project is aimed at resolving batch failures in production environment using some of the ML techniques

Vishal was tasked to learn the already built ML component. He learnt the architecture of the component within a month and started involving in expanding the component to other divisions.

He collaborated very closely with different teams & helped their developers to incorporate this component.

Vishal had also taken the opportunities to conduct few knowledge sharing sessions internally within the team based on his new learnings and experience with the projects. He had also initiated few employee engagement activities within the team which is very much needed during this tough situation to ensure team is connected.

Vishal had already proven his value to UBS within his short stint & we are looking forward to welcome him back soon.

Govindarajan VS  
Head of Automation Center of Excellence  
Strategy & Transformation, Technology Operations Center  
UBS India Pvt. Ltd.  
Tower 8, Commerzone IT Park  
Pune 411006  
Tel: (+91) 9673776789

## **ACKNOWLEDGEMENTS**

I would like to express my special thanks to Mr. Govindarajan VS, Director - TOC Strategy and Transformation - Automation COE, for giving me the opportunity to be a part of not one but two crucial projects of the team. He has always been motivational and supportive and provided the best inputs while developing the projects.

I would like to thank Mr. Abrar Girkar (project lead) and Mr. Vedavyas Burli (mentor) who have always been there to help me through out the internship - from getting me onboard to pushing the project to production.

I would like to thank Dr. D Srikanth Rao (Director), Dr. Balachandra (HOD of ICT Department), Mrs. Poornalatha G (Project Coordinator), Mrs. Anuradha Rao (Internal Guide) and all the teaching and non-teaching staff at MIT, Manipal for giving me the opportunity to do my 8th semester project at UBS and helping me throughout the journey. I would also like to thank them for providing the best in class curriculum that kept me updated with the latest tools and technologies.

I would also like to thank my parents for their constant support throughout this journey.

## ABSTRACT

Technology Operations Center (TOC) is the IT support end of UBS. It is basically the maintenance sector of Software Development Life Cycle (SDLC). The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to enhance a system's performance, correct problems, enhance security, or address user requirements.

Automation is the creation of software and systems to replace repeatable processes and reduce manual intervention. Regular health checks of remote servers, resolution of errors in applications that were repeatable and mundane tasks, are all automated. To achieve this, many tools are available in the market. UBS majorly uses two such tools - IPCenter and Automation Anywhere.

Whenever applications generate errors, IT tickets are raised and resolved through the automations developed in these software tools or by manual intervention. The first project aims at developing an automation in IPCenter that extracts the latest errors for particular jobs of an application and if they cannot be resolved automatically, it predicts the type of error using Machine Learning and attaches the label to the ticket. The ticket is then escalated for manual resolution of the error where the support analyst is provided with the information about the error that would help them to resolve the errors quick.

The second project aims to provide the automation developers with a catalog of all the automations developed in TOC. The purpose of this website is to promote reusability for standardized development and time saving. Developers can use the automations as-it-is or reuse them in their own automations. The Automation Catalog is a user friendly website with great methods for navigating as all automations are sorted based on their service type and the development platform. It also provides a medium for developers to enroll their automations to the catalog and also to subscribe to the catalog to get notified when new automations get added to it.

### **ACM Taxonomy -**

**[Software and its engineering]:** Software creation and management - Software post-development issues-Maintaining software; Software creation and management - Design Software - Requirements analysis, Software design engineering.

# Contents

Acknowledgements	vi
Abstract	vii
List of Tables	x
List of Figures	xi
Abbreviations	xii
1 Introduction	1
1.1 Health Check of a Remote Server.....	1
1.2 Ticket Enrichment using Machine Learning.....	1
1.3 Automation Catalog Development.....	2
2 Assignment - Health Check of a Remote Server.....	3
2.1 Introduction to IPCenter.....	3
2.2 Methodology.....	3
2.3 Implementation & Results.....	5
2.4 Conclusion.....	7
3 Project 1 - Ticket Enrichment using Machine Learning	7
3.1 Methodology.....	7
3.2 Implementation & Results.....	8
3.2.1 APIs used.....	11
3.3 Conclusion & Future Scope.....	15

4 Project 2 - Automation Catalog Development	16
4.1 Literature Survey.....	17
4.2 Methodology.....	17
4.3 Implementation & Results.....	18
4.3.1 Routing Module.....	18
4.3.2 Shared Service.....	19
4.3.3 Header Component.....	20
4.3.4 Automations Component.....	21
4.3.4.1 Homepage and Widgets.....	21
4.3.4.2 onSelect function.....	23
4.3.4.3 doSearch function.....	26
4.3.4.4 searchbyID function.....	27
4.3.4.5 openTab function.....	27
4.3.4.6 checkGitHubLink function.....	28
4.3.4.7 Access automation via ID.....	29
4.3.5 Registration Page.....	29
4.3.6 Subscription Page.....	31
4.3.7 Contact us Page.....	32
4.3.8 User Guide.....	33
4.4 Conclusion and Future Scope.....	34
References	35
Project Details	40

## **List of Tables**

B.1 Project Detail.....	40
-------------------------	----

## List of Figures

2.1 - Health check IPC automata (1/2)	5
2.2 - Health check IPC automata (2/2)	6
2.3 - Health Check Job JIL	7
2.4 - Automation Run Status	7
3.1 - Sample Dataset	8
3.2 - MLP model details	8
3.3 - Project Workflow	9
3.4 - APIs description	11
3.5 - ML Component (1/3)	12
3.6 - ML Component (2/3)	13
3.7 - ML Component (3/3)	14
4.1 - Home Page	16
4.2 - Application Structure	18
4.3 - Header	20
4.4 - Widgets	23
4.5 - Incident type automations	23
4.6 - Automation Description tab	24
4.7 - Automation Details tab	24
4.8 - Reusability Considerations tab	25
4.9 - Using existing data sources tab	25
4.10 - Execution Stats tab	25
4.11 - Search Results	26
4.12 - Automation by ID	27
4.13 - Link to Source Code	29
4.14 - Registration Form	30
4.15 - Subscription Form	31
4.16 - Contact us Form	33

## **ABBREVIATIONS**

TOC: Technology Operations Center

SDLC: Software Development Life Cycle

IPC: IPCenter

ML: Machine Learning

UI: User Interface

API: Application Package Interface

URL: Uniform Resource Locator

HTTP: Hyper Text Transfer Protocol

NPM: Node Package Manager

TS: TypeScript

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

# **1 Introduction**

Being a part of the Strategy and Transformation - Automation - Center of Excellence, the following two projects described in section 1.1 and section 1.2 are crucial to the team as they change the way automation developers in TOC develop.

Automation in IT is the creation of software and systems to replace repeatable processes and reduce manual intervention. Some of the processes that can be automated are health checks of servers, software maintenance etc. Automation is one of the main work areas in Technology Operation Center. IPCenter and Automation Anywhere are the two major platforms utilized in TOC. IPcenter is an autonomic IT management platform that is introduced in section 2.1. Section 1.1 and 1.2 introduce the two projects assigned and describe their objectives.

## **1.1 Health Check of a Remote Server**

This aim of this assignment is to automate health check of a remote server. The objective of this assignment is to get a hands-on for automation development in IPCenter.

Section 2 describes the methodology, implementation and results of the assignment.

## **1.2 Ticket Enrichment using Machine Learning**

The aim of the first project is to enrich the ticket data with error label for the latest error generated for particular jobs of an application and incorporate this automation component with the automation components of other teams. The resolution team can then immediately know what steps to take next. Prediction of the error label is done using the Multi-layer Perceptron algorithm of Machine Learning.

### **Objective -**

To develop an automation in IPCenter that extracts the latest errors for particular jobs of an application and enriches the IT ticket data with the error labels predicted using Machine Learning.

Section 3 describes the methodology, implementation and results of the first project.

## **1.3 Automation Catalog Development**

The Automation Catalog is a website that lists all the automations developed in TOC. Automation developers who have a certain use case, can find the specific automation in the catalog and reuse it. Reusable automations listed in the catalog are standardized. Hence, the developers using it, can be sure of its credibility. Reusability saves time of development and test. The use of the catalog can reduce risk, time of deployment and other costs.

The catalog enlists automations - their descriptions and other details for better understanding of the reusability of the automation. Developers can enroll their automations to the catalog by providing the details in the registration form. Users can subscribe to the catalog to get notified when new automations are added to the catalog.

### **Objectives-**

- I) To develop a catalog of all the automations developed in IPCenter and Automation Anywhere platforms in TOC.
- II) To provide a medium for developers to enroll their automations to the catalog and also to subscribe to the catalog to get notified when new automations get added to it.

Section 4 describes the methodology, implementation and results of the second project.

## **2 Assignment - Health Check of a Remote Server**

### **2.1 Introduction to IPCenter**

IT tickets is the generalized term used to refer to a record of work performed (or needing to be performed) by the IT support organization to operate the company's technology environment, fix issues and resolve user requests. Tickets may represent many different types of tasks or activities depending on the nature of the IT environment and the focus of the support team. Tickets have other names such as "service requests", "trouble tickets" or "support cases".

Managing IT tickets effectively is important to ensure that business receives full value of the company's IT investments. It helps IT function manage costs, provide better systems and services to users and mitigate the impact of business disrupting events.

IPcenter creates and controls "Virtual Engineers" that perform end-to-end operational processes across all IT stacks: servers, mainframes, storage, networks, virtualized environments, help desks, and applications operations. It is done by creating and solving IT Tickets.

### **2.2 Methodology**

Health checks of servers are to ensure the servers are working fine and this check is done on a regular basis. The server being checked is a unix box. Following are the steps/ flow of the health check performed-

- 1) Initially all the variables are declared-  
current date, hostname, componentOutput, escalationRequired (bool value- initially set as false.), lastUpdateLogTime, timeDifference, User.
- 2) A host command is run to get the current date (time is seconds).
- 3) Login into the server. If successful, go to step 4, else componentOutput variable is updated with the message 'Automation Failed when switching user in \$hostname. Please check manually', then 'escalationRequired' variable is set 'True' and a ticket is raised for resolution.

- 4) Open bash by running the command ‘bash;’.If successful, go to step 5, else componentOutput variable is updated with the message ‘Automation Failed when executing bash command. Please check manually’, then ‘escalationRequired’ variable is set ‘True’ and a ticket is raised for resolution.
- 5) Run adapter command. If successful, go to step 6, else componentOutput variable is updated with the message ‘Automation Failed when executing adapter command. Please check manually’, then ‘escalationRequired’ variable is set ‘True’ and a ticket is raised for resolution.
- 6) Run status command. If successful, go to step 7, else componentOutput variable is updated with the message ‘Automation Failed when executing status command. Please check manually’, then ‘escalationRequired’ variable is set ‘True’ and a ticket is raised for resolution.
- 7) Change directory to logs and get the latest log details (latest updated time).If successful, go to step8, else componentOutput variable is updated with the message ‘Automation Failed when changing directory to logs. Please check manually’, then ‘escalationRequired’ variable is set ‘True’ and a ticket is raised for resolution.
- 8) Compare latest updated time with current time. If the time difference is less than 3 hours, the server is in good condition and is running well and the ‘escalationRequired’ variable is false hence no ticket will be raised. Else the componentOutput variable is updated with the message ‘Time difference is less than 3 hours. Please stop and restart’ and the escalationRequired variable is set to ‘True’ and the ticket is raised for resolution.

## 2.3 Implementation & Results

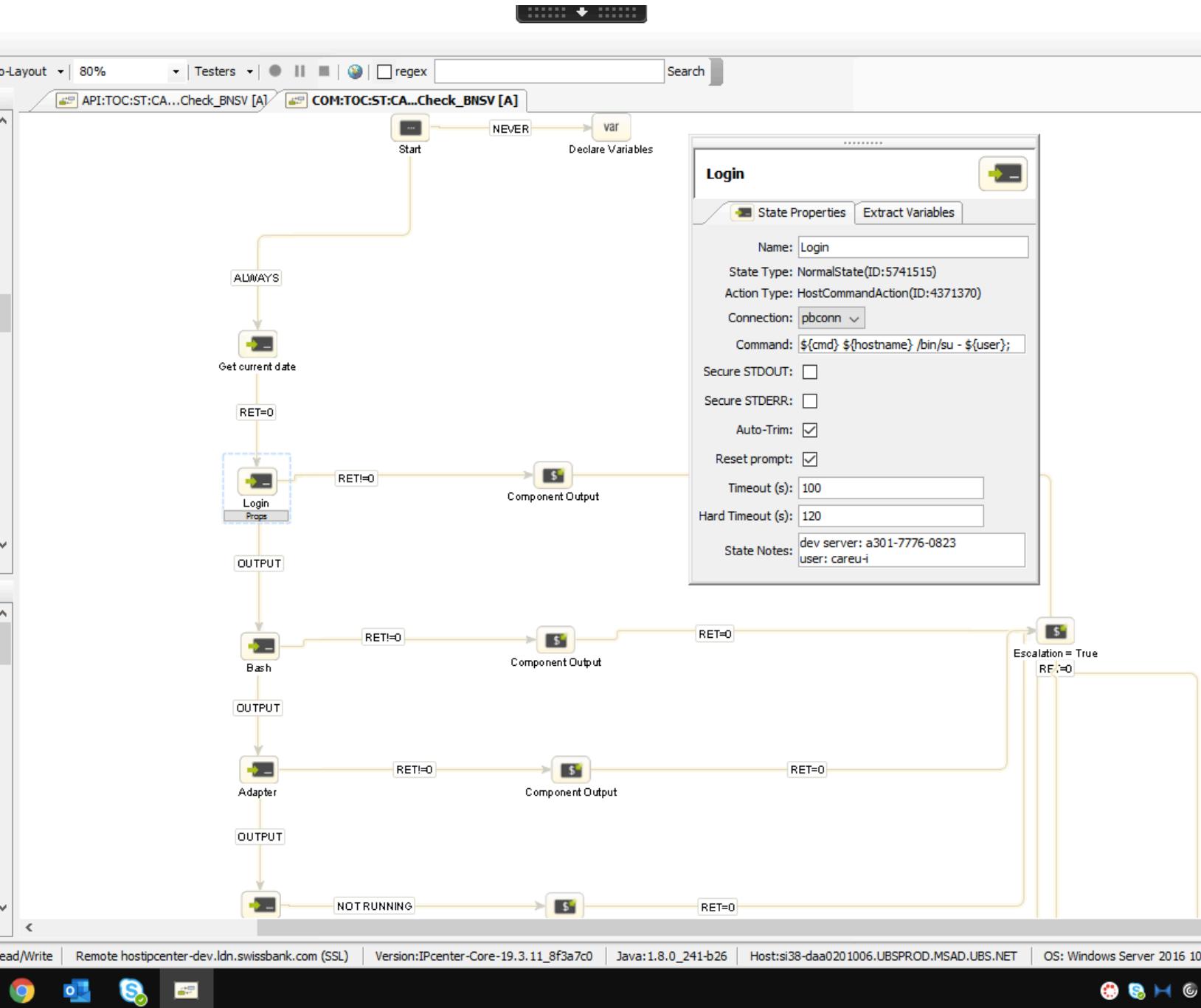


Figure 2.1 - Health check IPC automata (1/2)

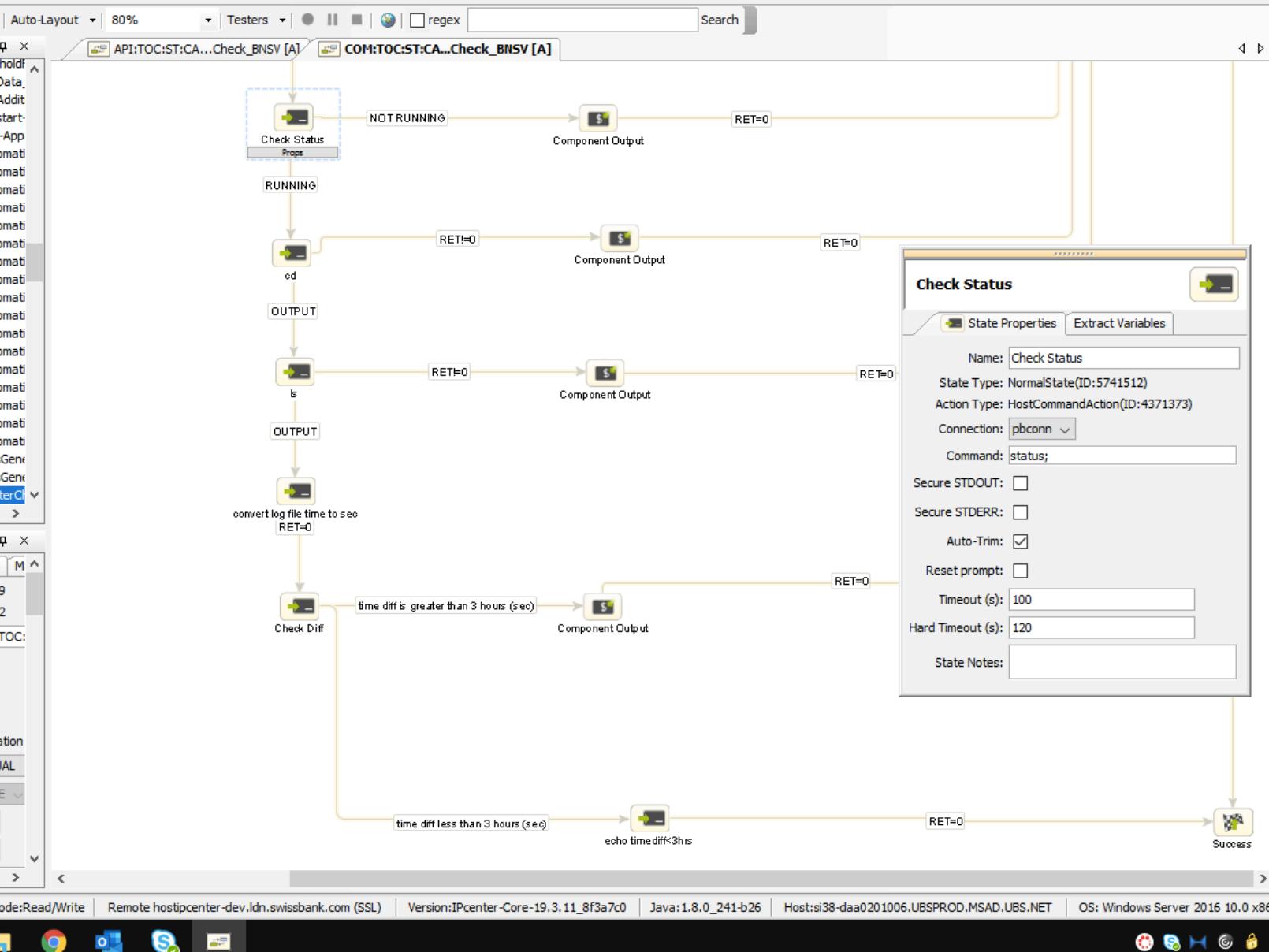


Figure 2.2 - Health check IPC automata (2/2)

To run the automation, a job is created with name -  
**IB\_ISIT\_IPSOFT\_1542\_API\_CAIP\_Adapter\_health\_check\_BNSV**

To automate the health check, a JOB JIL (job information language) is to be created with the schedule when the automation is to be run and other details.

Job JIL for this automation is shown in Figure 2.3

```
* ----- IB_ISIT_IPSOFT_1542_API_CAIP_Adapter_health_check_BNSV ----- */
```

```
insert_job: IB_ISIT_IPSOFT_1542_API_CAIP_Adapter_health_check_BNSV job_type: CMD
command: /app/auto-ipc/bin/auto-ipc2.py -c ${AUTOSERV} -s API.TOC.ST.CAIP_Adapter_health_check_BNSV -e {} -i ${AUTO_JOB_NAME}-${AUTOSERV}${AUTORUN}
achine: 1542_VM_IPcenter_Dev_Util
wner: auto_ipc
ermission:
ate_conditions: 1
ays_of_week: all
tart_times: "05:00,05:30,06:00,08:00,08:30,09:00,10:00,11:00,13:00"
escription: "Training API.TOC.ST.CAIP_Adapter_health_check_BNSV"
td_out_file: "/app/auto-ipc/logs/${AUTO_JOB_NAME}-${AUTOSERV}${AUTORUN}_$.out"
td_err_file: "/app/auto-ipc/logs/${AUTO_JOB_NAME}-${AUTOSERV}${AUTORUN}_$.err"
larm_if_fail: 1
larm_if_terminated: 1
imezone: GMT
how job run
```

[Back](#) For any questions/issues please mail: [artur.heise@ubs.com](mailto:artur.heise@ubs.com)

Figure 2.3 - Health Check Job JIL

The Health check is performed 9 times everyday and the schedule is written in the start\_times field.

Job Name							Last Start	Last End	ST Run/Ntry	Pri/Xit	TimeZone
IB_ISIT_IPSOFT_1542_API_CAIP_Adapter_health_check_BNSV							04/25/2020 13:00:02	04/25/2020 13:00:12	SU	112700955/1 0	(GMT)
<ul style="list-style-type: none"> <li><a href="#">Extended job report</a></li> <li><a href="#">Show job definition (JIL)</a></li> </ul>											
<a href="#">Back</a> For any questions/issues please mail: <a href="mailto:artur.heise@ubs.com">artur.heise@ubs.com</a>											

Figure 2.4 - Automation run status

## 2.4 Conclusion

Automation run status is ‘successful’ indicating the server is in good health as on 25/04/2020 13:00:12 hours.

## 3 Ticket Enrichment using Machine Learning

### 3.1 Methodology

Initially, historical data (~90 days) is collected containing the error logs of the jobs names and location details are collected. They are labelled manually (data preparation) and then multi-layer perceptron model of machine learning is applied.

The model is developed in Python3.7 using the standard libraries. The model is then pickled.

Error	Label
Return code of script 'cp -f /sbclocal/localstorage/AMDHUB/FeedFiles/Source/Internal/IPMA/Outbound/NEW_AMDH_* /app/localstorage/AMDHUB/FeedFiles/Archive/' is 1 Connection to a301-7064-2996.ldn.swissbank.com closed by remote host. File importation is running. Stream: /sbclocal/localstorage/AMDHUB/FeedFiles/Source/Bloomberg/BankOffice/Inbound/UBSAM_fixedincome_bo_namr_sp.out.20200320. 227312 parsed lines, 0 skipped lines and 26692 invalid lines. 0 inserted objects, 57964 updated objects, 211969 unchanged objects, 60727 rejected objects, 0 skipped objects.403 retries. Provider return 0 objects errors. No reject has been generated.	FeedFile-Archive-ReturnCodeOne Connection-Lost
Bad args : Error while truncating 1 end line(s) of the file : /sbclocal/localstorage/AMDHUB/FeedFiles/Source/ICE/Inbound/gpm_APUBSGLO_GPMF1AP_1930au.20200322.csv .. IO exception : /sbclocal/localstorage/AMDHUB/FeedFiles/Source/ICE/Inbound/gpm_APUBSGLO_GPMF1AP_1930au.20200322.csv (No such file or directory) [java.io.FileNotFoundException]	FileNotFoundException

Figure 3.1 - Sample Dataset

Request
URL ' <a href="http://a301-5564-1050.sng.swissbank.com:5000/jf/create_model">http://a301-5564-1050.sng.swissbank.com:5000/jf/create_model</a> ' JSON {     "model_algo_used": "MLP",     "model_id": 101,     "model_labels": "[ 'Feed issue-Threshold error' 'Feed issue-Validation error'\n 'Feed Issue - File Unavailable']",     "model_pkl_file_name": "pkl_model_101.h5",     "model_score": 0.9893617021276596,     "model_test_data_size": 94,     "model_training_data_size": 280,     "model_vect_pkl_file_name": "pkl_vectr_101.pkl" }
Response
{     "status": "SUCCESS" }

Figure 3.2 - MLP model details

# WORKFLOW:

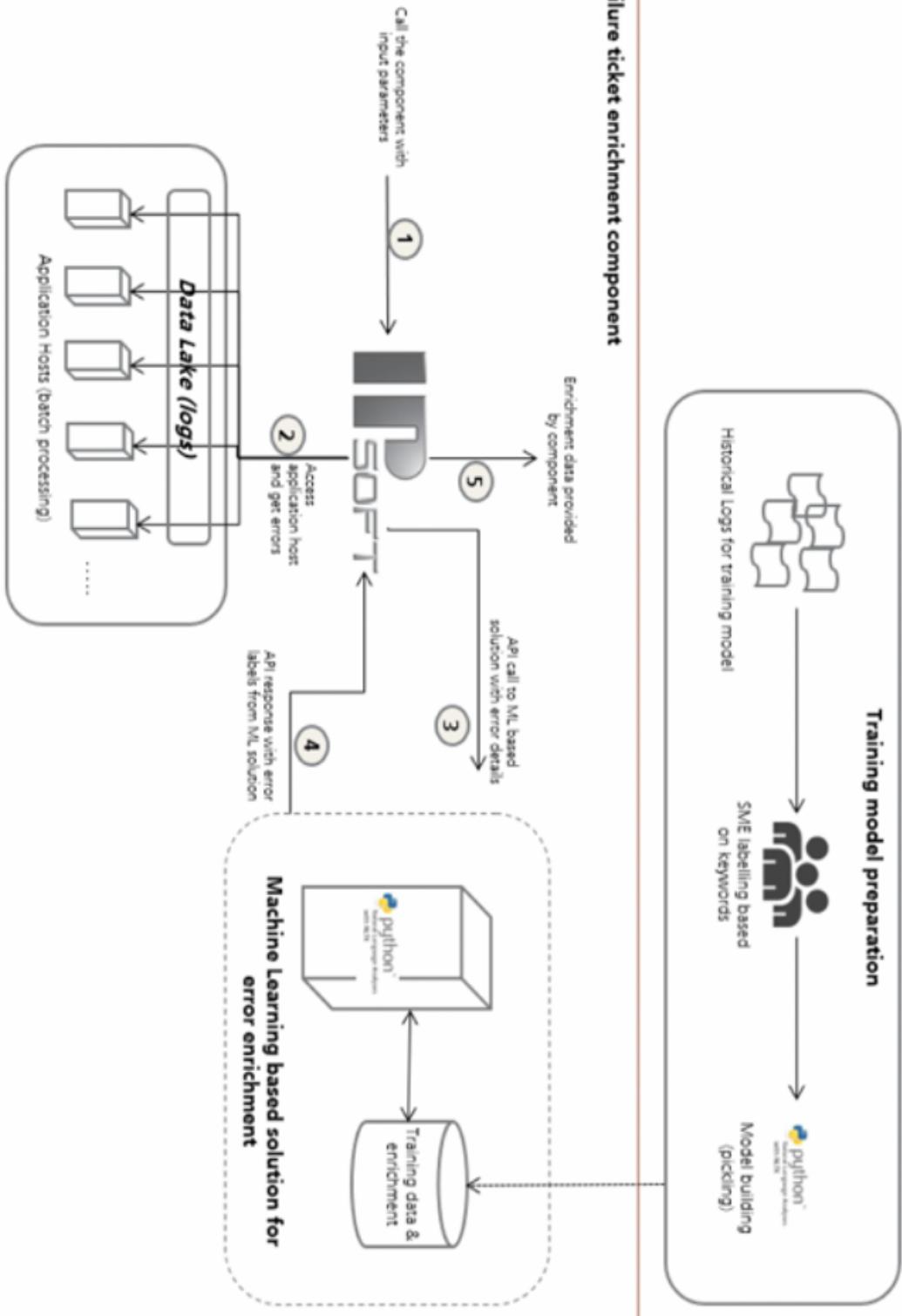


Figure 3.3 - Project Workflow

Flow of the automation component-

- 1) The automation component is called whenever events occur for particular jobs. Job name is matched with the ‘Matcher’ detail provided when developing the automation. If matched, the automation is run.
- 2) Next step is to login into the application server, get the latest error log and format is so as to feed it into the ML model.
- 3) Then, the ML API is called from the IPcenter’s component and the error label is predicted and is returned to IPcenter.
- 4) Ticket data is enriched with the error label and then escalated.

After this automation component is developed, it is integrated with the automation wrapper of other teams. We, as the central automation team, contact other teams those who require this component to be integrated with their wrapper.

After linking the component, the wrapper is tested in the IPCenter’s development environment. A job is selected for this purpose that gives error when run. The error is then labelled, ticket is enriched and escalated for resolution.

Figure 3.3 shows the workflow of the project.

## **3.2 Implementation & Results**

The automation component is developed in the following steps-

- 1) The component initially gets the following details from the wrapper from which it is being called (wrapper automations are those having different requirements developed by different teams but call the same component) - Job name, radar id, hostname, alertSummary; which is then wrapped up as one variable - event data. The jobs for which the component has to run are embedded by calling the job\_onboard API from the wrapper automation.
- 2) Event data is then stored in the batch failure database. This is done by calling the add\_event API.
- 3) db\_get\_model API is then called that returns the job path, ML model file name, error pattern and other details for the specific job.

- 4) Next, it logs into the server and changes the directory to the logs file. The latest error is then extracted and changed into appropriate format for ML prediction.
- 5) predict\_label API is called which and the latest error is passed. It returns the predicted label.
- 6) IP Ticket is then enriched with this label by calling the POST method for db\_add\_enriched\_event\_data and the component output is sent back to the wrapper.

In all the above steps, if any error occurs, the error message is wrapped up into the componentOutput variable and forwarded / escalated for manual intervention and it does not go ahead with further steps.

Figures 3.5, 3.6 and 3.7 are the screenshots of the automation component.

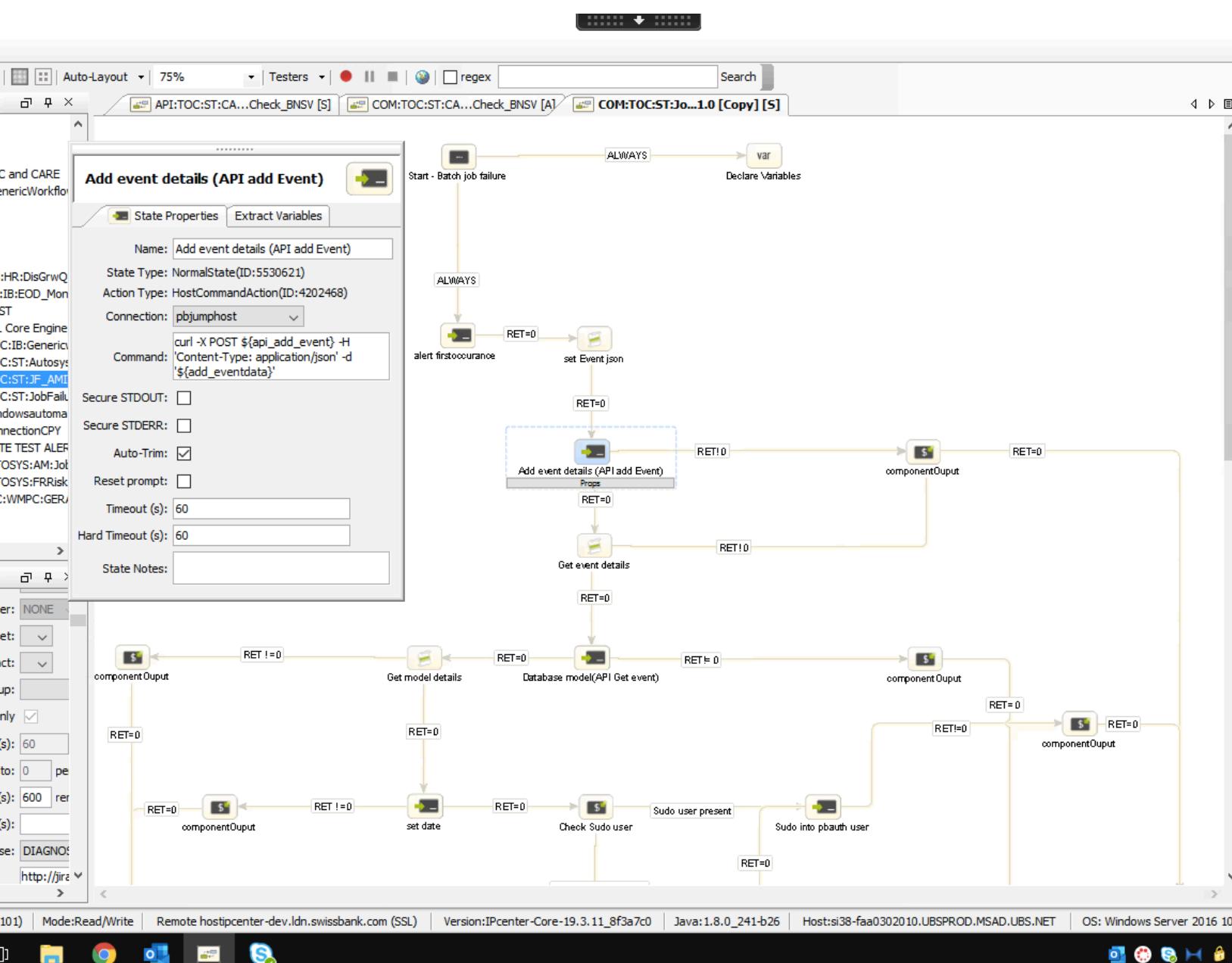
### 3.2.1 APIs used -

Base API - <http://a301-5564-1050.sng.swissbank.com:5000/>

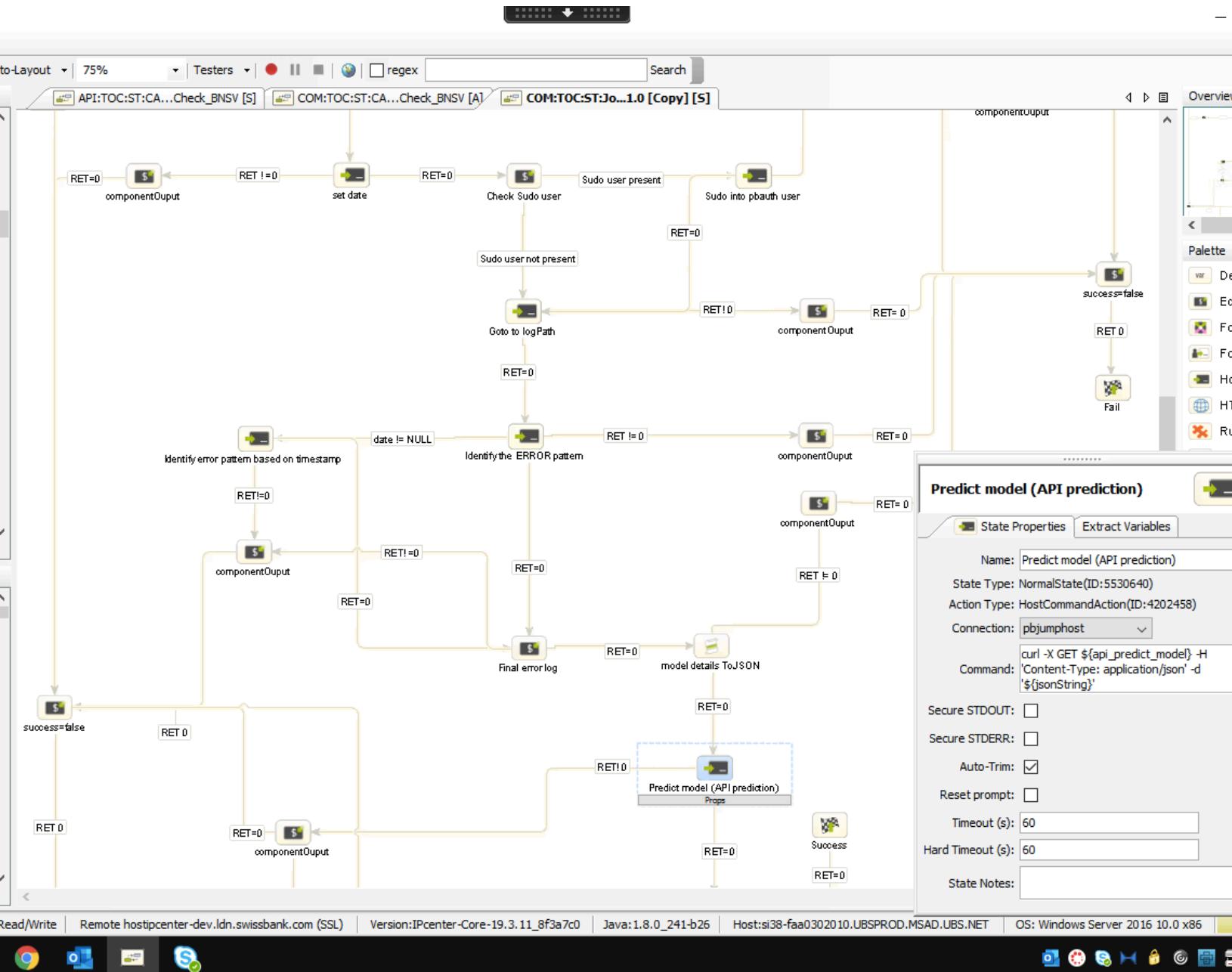
The first API - job\_onboard is called from the Wrapper component and rest APIs are called from this component.

Method	URL	Description
POST	/jf/job_onboard	Onboard a new job metadata to backend
POST	/jf/create_model	Create the Machine Learning model for a job in backend (please note the ML model for each job type needs to be created outside of this API by application SME)
POST	/jf/db_add_event	Insert each event data to batch failure database
GET	/jf/db_get_model	Retrieve the Machine Learning model detail from backend for the requested batch job
GET	/jf/predict_label	Trigger machine learning model and returns the predicted label for the error message
POST	/jf/db_add_enriched_event_data	Enriches the already inserted event data with the predicted label (this will be used for feedback mechanism with application SMEs in future)

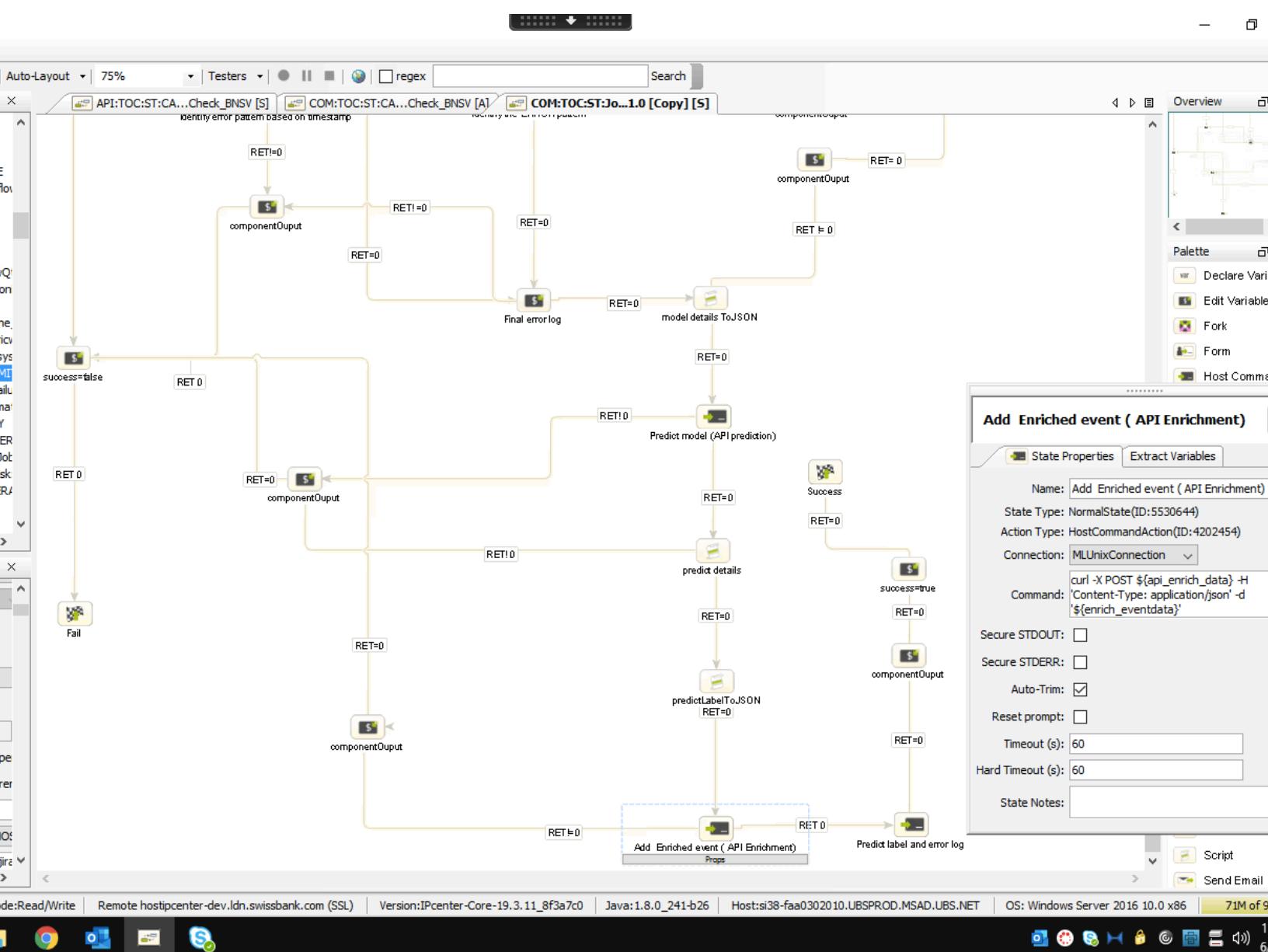
Figure 3.4 - APIs description



### 3.5 - ML Component (1/3)



### 3.6 - ML Component (2/3)



### 3.7 - ML Component (3/3)

The two teams - Asset Management and Investment Banking were assisted with incorporating this ML component into their automation wrappers. Following are the steps followed for both the teams-

- 1) Job names and error logs were collected from the teams.
- 2) Preprocessing of the errors data - removing headers and footers and manual labeling of the errors was performed.
- 3) MLP algorithm was run on those error data.
- 4) The model was pickled and stored by calling the create\_model API.
- 5) The ML automation component was added as a link in the automation wrappers developed by the respective teams.

### **3.3 Conclusion and Future Scope**

The ML automation component linked to the automation wrappers, enriched the tickets with the error labels.

The future scope is to automate some of the resolutions of the errors based on the error label predicted.

For example, most of the errors require a restart of the application. Hence, whenever such errors get predicted with the error label, the error label can be mapped with the process it must follow for resolution. Here, the process is restart of the application.

## 4 Project 2 - Automation Catalog Development

The screenshot shows the homepage of the Technology Operations Center - Automation Catalog. At the top, there's a navigation bar with links for Home, Register Your Automation, User Guide, Subscribe, and Contact Us. A search bar is also present. The main content area has a title 'Browse Catalog' and several sections: 'By Source' (ServiceNow, Netcool), 'By Type' (Incident, Run, Query, Request, Governance and Op Design, Package or Deploy, Production Assurance, Management Support, Risk and Controls, Continuous Improvement), 'By Tools' (IPCenter, Automation Anywhere), and 'What is the Catalog?' which describes the catalog's purpose of accelerating digital transformation through a library of automation solutions. Below these are sections for 'How does it Help Me?', 'How do I use it? In three easy steps!', and 'Can I share my own Automations?'. The bottom section, 'What's New?', features five cards: 'Execution Stats' (14211 visitors), 'Link to source code' (551 automations onboarded), 'Extended Search' (416 IPC Automata), 'Subscription' (117 AA Components), and 'Script Automations' (18). The footer contains a copyright notice: '© 2020 Technology Operations Center, All Rights Reserved'.

Figure 4.1 - Home Page

The automation catalog is a website conceived as a data store to record all live automations in TOC deployed through strategic platforms - IPCenter and Automation Anywhere and non-strategic platforms - Python, Shell, Perl etc. in order to:

- i) Drive standardization and commoditization (off the shelf readily deployable automations).

- ii) Save cost through minimal effort to identify & reconfigure existing automation
- iii) Reduce risk and timeline of deployment.
- iv) Report on existing automations' performance and running volumes.

In layman's terms, automation developers can reuse the automations listed in the catalog to reconfigure their own automations by adding these standardized automation components and save time, reduce risk and other costs that may incur.

## **4.1 Literature Survey**

Before developing the website, a market research was performed to gather ideas for addition of new features to the catalog.

Google cloud services website [3] was referred for the subscription feature (Section 4.3.6)

AWS Service Catalog [2] was referred for the design of the header (Section 4.3.3), the widgets section on the home page (Section 4.8.1), routing service for individual automations (Section 4.8.7) and listing automations based on platform.

## **4.2 Methodology**

The website is built on Angular 8.

Angular is a Typescript based web application framework. Typescript is an open source programming language which is a strict syntactical superset of JavaScript.

IDE - IntelliJ Idea Ultimate 2019

Node version - 10.16.0

NPM version - 6.9.0

Angular CLI version - 9.1.4

Bootstrap version - 3.3.7

The application is built on one module - the 'App' module. A module is mechanism to group components, directives and services that are related, in order to create an application.

The module contains 6 components each having a TS file, HTML file and CSS file, a routing module (TS file, described in section 4.3.1) and a Service (TS file, described in section 4.3.2). Figure 4.2 is a pictorial representation of application structure.

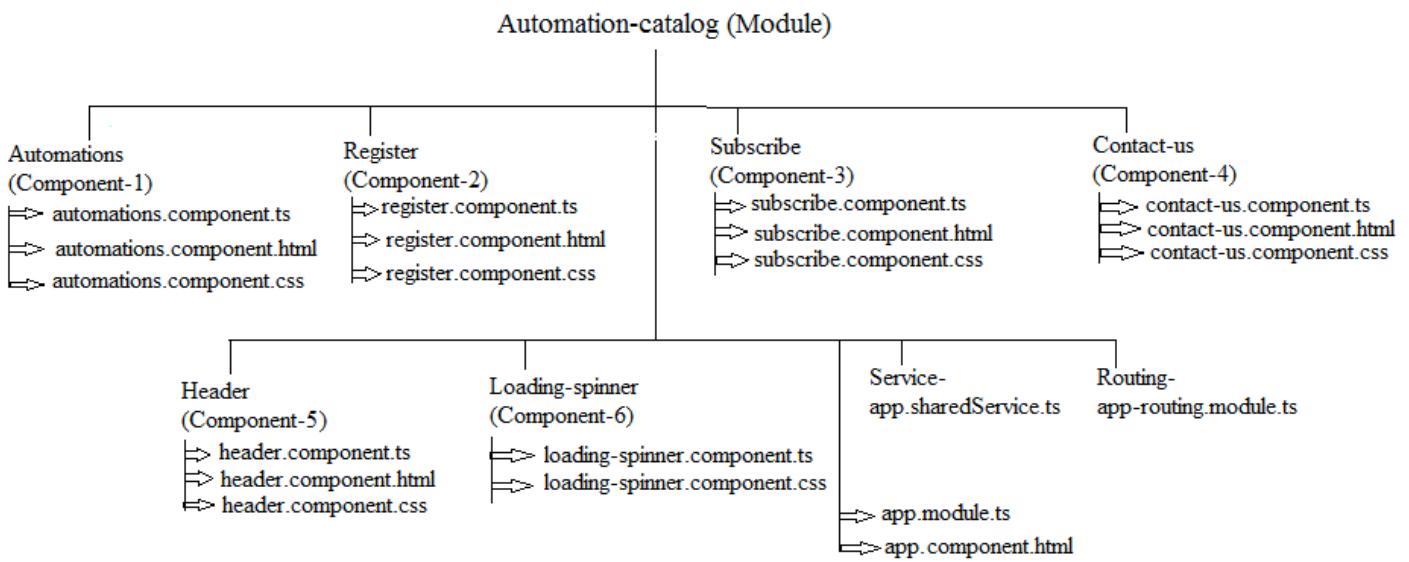


Figure 4.2 - Application Structure

The app.module.ts is the module's typescript file of which class, describes how the application parts fit together. All the components are registered in the declaration array and their paths are imported.

### 4.3 Implementation & Results

Following Sections from **4.3.1** to **4.3.8** describe each and every feature and its implementation in detail.

#### 4.3.1 Routing Module

All the routes are configured in the 'app-routing.module.ts' file. Following are the routes-

- i) path=‘’ (empty path), component=AutomationsComponent. This takes the application to the homepage. URL=‘http://goto/toc-automation-catalog/’.
- ii) path=‘/register-automation’, component=RegisterComponent. This loads the register component. URL=‘http://goto/toc-automation-component/register-automation’.
- iii) path=‘/subscribe’, component=SubscribeComponent. This loads the subscribe component. URL=‘http://goto/toc-automation-catalog/subscribe’.

iv) path=''/contact-us', component=ContactusComponent. This loads the contact-us component. URL='http://goto/toc-automation-catalog/contact-us'.

v) path=':automationstype', component=AutomationsComponent.

'automationstype' is a variable that is matched with in the OnInit function of the Automations component-

```
if(automationstype === 'Incident') {
```

onSelect(domainname) function is called and 'Incident' is passed as parameter. (onSelect function is described in Section 4.3.4.2)

```
} else if (automationstype === 'Run') {
```

onSelect(domainname) function is called and 'Run' is passed as parameter.}

```
.
```

```
.
```

Similarly other domains (Request, Query etc.) are called.

URLs= 'http://goto/toc-automation-catalog/Incident' or 'http://goto/toc-automation-catalog/Run'....etc.

If ':automationstype' is a number, this is mapped with the automation's catalog ID and the 'searchbyid(id)' function is called to display that automation. (searchbyid(id) function is described in section 4.3.4.3).

### 4.3.2 Shared Service

Services in Angular are used for sharing of code between different components. It is for data connection that needs to be shared across components.

The service used in this application is 'app.sharedService.ts' and it is used for communication between the Header component, that reads in the input text for automation search from the user, and the Automations component that displays the results.

Angular's 'Subject' is a special type of Observable that allows data to be multicasted to many observers. A subject is defined in the sharedService.

When the user enters search text in the input field and submits the form, the service's sendMessage function is called that calls the ".next()" method of the subject to feed in a new value.

SendMessage function in the service-

```
sendMessage(message: string) {  
    this.subject.next(message);  
}
```

Automations component is the observer and it subscribes to the service's getMessage method. The getMessage method is the observer to the subject and when '.next()' method is called, it is casted to the observer in getMessage and this function returns the value to the automations component that calls the doSearch() method, passing the search text to display the results.

GetMessage function in the service-

```
getMessage(): Observable<string>{  
    return this.subject.asObservable();  
}
```

Subscription the the service in Automation component's constructor-

```
constructor(private appSharedService: AppSharedService) {  
    this.appSharedService.getMessage().subscribe (message =>  
        this.doSearch(message);  
}
```

### 4.3.3 Header Component

The header component is the navigation bar on the top of the page. It comprises of two bootstrap-navbars, both fixed on top of every page.



Figure 4.3 - Header

Top-most navigation bar-

The brand-image when clicked navigates the app to the home page by changing the router link to “/Home” and calling the Automations component to display the home page.

The search field reads in input string and on clicking the submit button, the sendMessage function of the app.sharedService.ts is called and the text is passed.

```
this.appSharedService.sendMessage(searchtext);
```

Second navigation bar-

This bootstrap navbar is placed to switch between different pages-

- i) Home - This changes the router link to “/Home” and calls the Automations component and displays the home page.
- ii) Register Your Automation - This changes the router link to “/register-automation” and calls the Register component and displays the registration form page.
- iii) User Guide - On clicking this button, the User Guide pdf file is downloaded and opened.
- iv) Subscribe - This changes the router link to “/subscribe” and calls the Subscribe component and displays the subscription form page.
- v) Contact Us - This changes the router link to “/contact-us” and calls the Contactus component and displays the contact us form page.

#### **4.3.4 Automations Component**

This component is the main component of the application. This is where the homepage, the automations lists and the main functions such as onSelect(domain), doSearch(searchQuery) etc. are present. It is the component that is loaded first in the application. URL=“<http://goto/toc-automation-catalog/>” or “<http://goto/toc-automation-component/Home>”.

##### **4.3.4.1 Homepage and Widgets**

The homepage (Figure 4.1) is divided into 3 HTML tables.

- 1) The first table has one row and two columns. The first cell is the side panel. The side panel comprises links to browse the catalog based on

- 2) i) By Source ( Source from which the tickets are generated and automations are built to resolve them.)
- a) ServiceNow
  - b) Netcool
- ii) By Type ( Automations are segregated here based on its service type)
- a) Incident
  - b) Run
  - c) Query
  - d) Request
  - e) Governance and Op Design
  - f) Package or Deploy
  - g) Production Assurance
  - h) Management Support
  - i) Risk and Controls
  - j) Continuous Improvement
- iii) By Tools ( Automation development platforms)
- a) IPCenter
  - b) Automation Anywhere

The next cell is the textual content describing what the catalog is, its benefits and usage guidelines.

- 2) Second table describes few of the latest features added to the catalog - Execution Stats, Link to Source Code, Extended Search and Subscription.
- 3) Third table contain the widgets that shows the count of -
- i) Visitors - This is counted in the backend and is read from Flask API.
  - ii) Total Automations Onboard - This is again read from API.
  - iii) IPCenter Automata
  - iv) Automation Anywhere bots/ Components
  - v) Script Automations - This is calculated as- total automations onboard - IPC automata count - AA bot count.



Figure 4.4- Widgets

#### 4.3.4.2 onSelect function

When user clicks on either of the links on the side panel, the onSelect(domain) function is called and the domain is passed as parameter. This function hides all the content of the homepage except the side panel and displays the automation-list division.

Then, data is requested using the HTTP GET method passing the domain selected as parameter in the request message. For example, when user clicks on ‘Incident’, the GET method is called and ‘Incident’ is passed as parameter. Data is received and displayed on the automation-list division. Figure 4.5 shows the Incident type automations. Route is dynamically changed. New URL- “<http://goto/toc-automation-catalog/Incident>”.

The screenshot shows the Technology Operations Center - Automation Catalog page. The top navigation bar includes links for Home, Register Your Automation, User Guide, Subscribe, and Contact Us. A search bar is also present. The main content area is titled "Browse Catalog" and features a sidebar with filters for "By Source" (ServiceNow, Netcool), "By Type" (Incident, Run, Query, Request, Governance and Op Design, Package or Deploy, Production Assurance, Management Support, Risk and Controls, Continuous Improvement), and "By Tools" (IPCenter, Automation Anywhere). The main content area displays a grid of automation cards:

- IIS\_Application\_Alerts-v1.15** (IP SOFT): Enhancement : To set escalation level to L1 for alerts originating from servers operated for GLAM AA24951 AT7730 MIRO and Matcher Updated for JIRA - AUTOIS- Read more
- Dynamic Correlation** (IP SOFT): IPCenter automatically correlates similar patterns raised in netcool within a specific timeframe, raising one SNOW ticket only and adding any future occurrences within the Read more ★★★★★
- serviceCrashAlerts-v1.3.02** (IP SOFT): AA24951 AT7730 Read more
- High Memory Utilization** (IP SOFT): Resolution of these tickets involves checking & analyzing memory utilization and reporting to the server owner in case of frequent high memory utilization of any Read more
- Memory Utilization** (IP SOFT): Enhancement of AUTOIS-10 High memory utilization Read more
- Dbaas Alerts - Memory Utilization** (IP SOFT): Takes in DBaaS memory alerts and suppress them. Agent/Domain=DBaaS - GBL, AlertGroup/ParameterName=DBaaS Infra: Host, Read more

Figure 4.5 Incident type automations

When clicked on ‘Read more’ button a popup appears to show the automation details. For the ‘Dynamic Correlation’ automation as shown in Figure 4.5, the following popup appears to display its complete details.

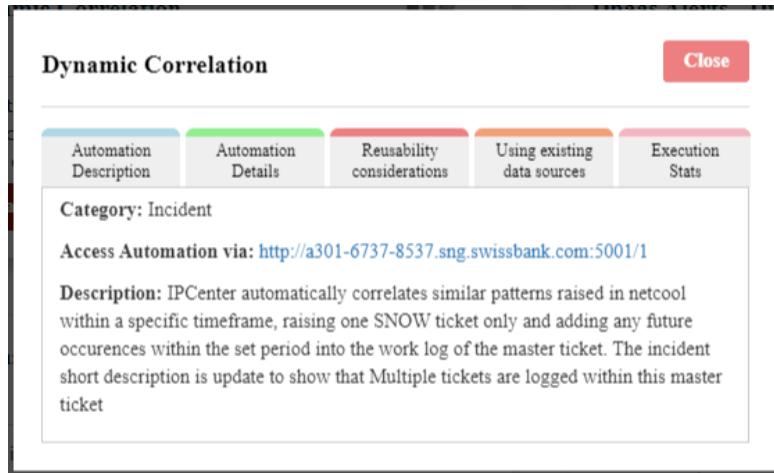


Figure 4.6 Automation Description tab

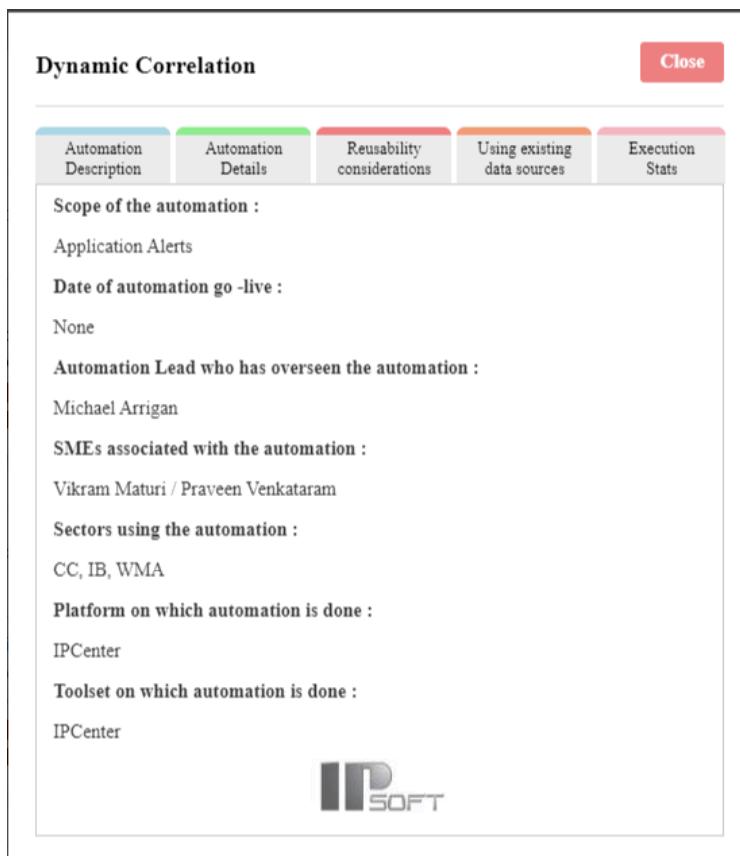


Figure 4.7 Automation Details tab

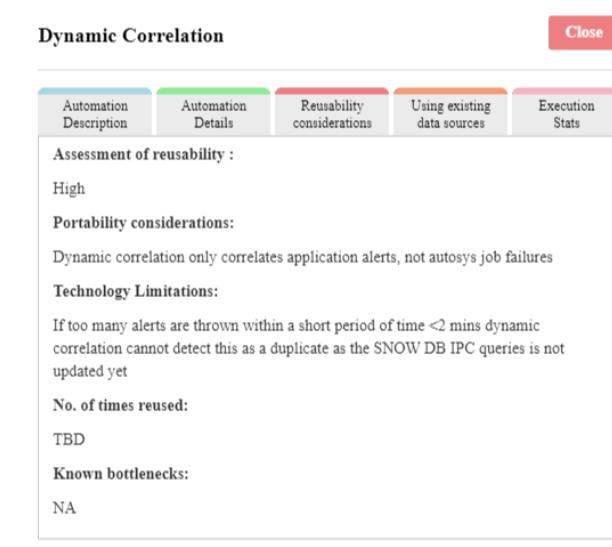


Figure 4.8 Reusability Considerations tab

Figure 4.9 Using existing data sources tab

Figure 4.10 Execution Stats tab

These tabs can be toggled upon clicking that calls the ‘openTab(event, tabName)’ function explained briefly in section 4.3.4.5.

On clicking the close button, the pop-up closes and we are back to the automation listing page (Figure 4.5).

#### 4.3.4.3 doSearch function

When the user inputs a search query in the header, the automation component that is subscribed to the getMessage() function of the Angular Service (Section 4.3.2), calls the doSearch(searchQuery) function.

This function reads in the search query as parameter and calls the API using ‘GET’ method passing the ‘searchQuery’. The response from the API is the list of automations as results for the search query. The search query is mapped to the automation name and the automation description to generate results.

It then calls the display function and the homepage items - ‘main-slide’, ‘what’s new table’ and the ‘widgets table’ are hid and the ‘automation-list’ section is displayed with that list of automations. Figure 4.11 shows the search result for the searchQuery=“health check”.

The screenshot shows the 'Technology Operations Center - Automation Catalog' interface. At the top, there's a search bar with the text 'health check'. Below the search bar, the main content area is titled 'Search results displayed for keyword: health check'. The results are presented in a grid format with six items:

- Automated Feed Arrival Checks**: Automation to handle feed delays related batch failures. IPCenter will handle the batch failure alerts by checking the required feed files. Incenter will raise a ticket with a star rating of 5. [Read more](#)
- Automated Weekend HealthChecks**: Automation Anywhere bot to do application health checks (UI) and ensure functionalities of the applications work as expected. A star rating of 5. [Read more](#)
- Automated batch maxrun monitoring and ticket closure**: IPC Automation verifies the job status and determines the average runtime by checking the last 3 runs (+additional). A star rating of 5. [Read more](#)
- Home Directory Deletion Check and Ticket Closure**: The robot checks if the home drive has been deleted. If yes, then closes the respective ticket. A star rating of 5. [Read more](#)
- OBS Sunday Checks**: OBS Sunday Checks. A star rating of 5. [Read more](#)
- Application Heath Check UI automation**: This application health check is to verify critical applications of HR Stream to ensure smooth. A star rating of 5. [Read more](#)

On the left side of the main content area, there's a sidebar with sections for 'Browse Catalog' (By Source: ServiceNow, Netcool; By Type: Incident, Run, Query, Request, Governance and Op Design, Package or Deploy, Production Assurance, Management Support, Risk and Controls, Continuous Improvement; By Tools: IPCenter, Automation Anywhere), 'User Guide', 'Subscribe', and 'Contact Us'.

Figure 4.11 - Search Results

#### 4.3.4.4 searchbyID function

This function is called from the `onInit()` function. When user enters an id in the URL, example: '<http://goto/toc-automation-catalog/1>', the `onInit` function reads the id from the route (explained in section 4.3.1) and calls the `searchbyid(id)` function.

The `searchbyid(id)` function calls the API using the HTTP GET method passing the 'id' as parameter in the request message. The response from the API is the automation details of the automation for the specific Catalog\_id.

It then calls the `display` function and the homepage items - 'main-slide', 'what's new table' and the 'widgets table' are hid and the 'automation-list' section is displayed with that specific automation. Figure 4.12 shows the automation with Catalog\_id=1.

The screenshot shows the homepage of the Technology Operations Center - Automation Catalog. At the top, there is a navigation bar with links for Home, Register Your Automation, User Guide, Subscribe, and Contact Us. A search bar is also present. The main content area features a sidebar titled 'Browse Catalog' with sections for 'By Source' (ServiceNow, Netcool), 'By Type' (Incident, Run, Query, Request, Governance and Op Design, Package or Deploy, Production Assurance, Management Support, Risk and Controls, Continuous Improvement), and 'By Tools' (IPCenter, Automation Anywhere). To the right of the sidebar, a card titled 'Dynamic Correlation' displays a brief description: 'IPCenter automatically correlates similar patterns raised in netcool within a specific timeframe, raising one SNOW ticket only and adding any future occurrences within the set'. It includes a 'Read more' button and a five-star rating icon. The URL in the browser address bar is <http://a301-6737-8537.sng.swissbank.com:5001/>.

Figure 4.12- Automation by ID

#### 4.3.4.5 openTab function

This function is called whenever an event occurs i.e. when either of the tabs are clicked. The tabs are listed as list items of unordered list and for each list item, a click event function is called - '`openTab(event, tabName)`'.

```

<ul class='tab-nav'>
<li class='tablinks' (click)='openTab(event, 'first-tab')><a>Automation Description</a></li>
<li class='tablinks' (click)='openTab(event, 'second-tab')><a>Automation Details</a></li>
<li class='tablinks' (click)='openTab(event, 'first-tab')><a>Reusability considerations</a></li>
<li class='tablinks' (click)='openTab(event, 'first-tab')><a>Using existing data sources</a></li>
<li class='tablinks' (click)='openTab(event, 'fifth-tab')><a>Execution Stats</a></li>
</ul>

```

For example, when ‘Automations Description’ tab is clicked it shows the automation description details - Category, Access Automation Via link and Description as shown in figure 4.6.

Similarly, it displays results for the tabs- Automation Details, Reusability considerations, Using Existing data sources and Execution Stats.

```

openTab (event, tabName) {
  //event is the mouse click event
  // tabName can be one of the following - 'first-tab', 'second-tab',
  //third-tab', 'fourth-tab', 'fifth-tab'.
  //Based on the tabName, specific details are only displayed in the content
  //box.
}

```

#### **4.3.4.6 checkGitHubLink function**

This function is called from the ‘automations.component.html’ file for each automation. It checks if the link in the ‘Links and Attachments’ section of ‘Using existing data sources’ tab is a GitHub link or not. Regex used = ‘/github/’. The functions return a boolean value.

If true, the link is a GitHub link and it is provided as a clickable link below the automation name in the popup as ‘Link to Source Code’. (Figure 4.13).

If false, nothing is displayed.

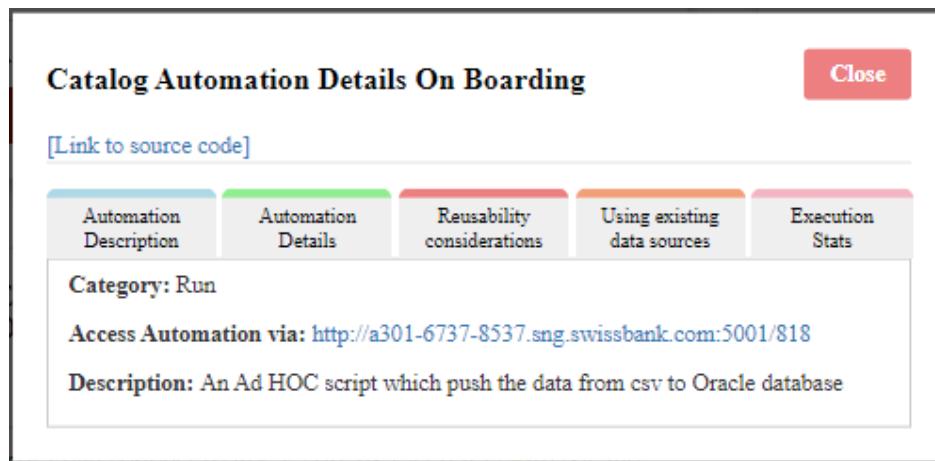


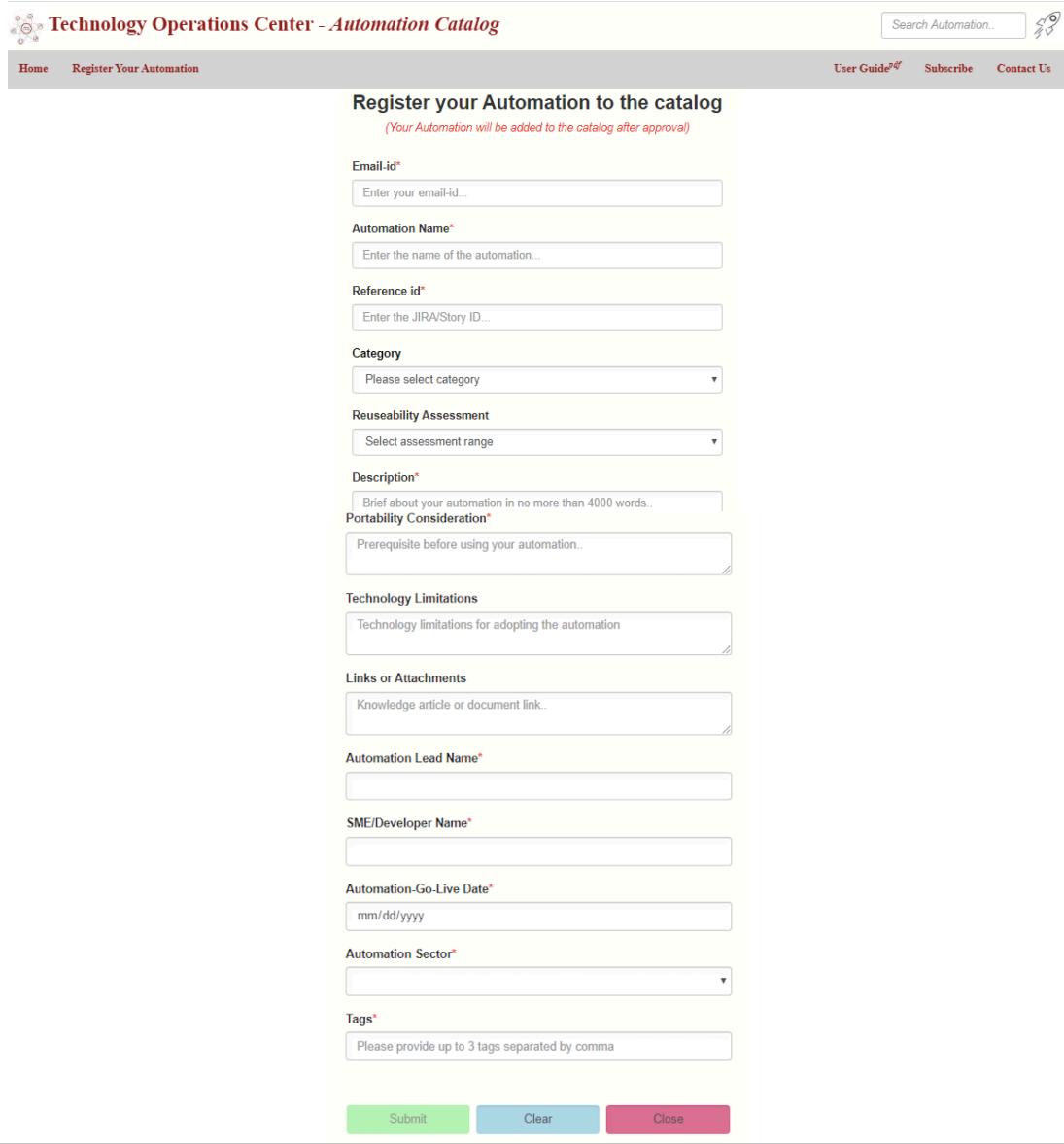
Figure 4.13 - Link to Source Code

#### 4.3.4.7 Access automation via ID

The ‘Access Automation via’ field in the Automation Description tab as shown in Figure 4.13, provides a sharable link for that specific automation. Each automation has a unique ID in the catalog and that is accessed as a parameter in the URL. When this URL is clicked, the ID is read as ‘automationstype’ variable in routing module which calls the `onInit` function of Automations Component that checks if the parameter is a number or not. If the parameter is a number, ‘`searchbyid(id)`’ function is called and the next process is described in section 4.3.4.4.

#### 4.3.5 Registration Page

Users/ Automation developers can add their automations to the catalog by filling in the registration form which can be accessed using the URL - <http://goto/toc-automation-catalog/register-automation>. Figure 4.14 is a screenshot of the page.



The screenshot shows a registration form for 'Automation Catalog'. The top navigation bar includes links for Home, Register Your Automation, User Guide, Subscribe, and Contact Us. A search bar is also present. The main title is 'Register your Automation to the catalog' with a note '(Your Automation will be added to the catalog after approval)'. The form fields include:

- Email-id\* (Input field: Enter your email-id...)
- Automation Name\* (Input field: Enter the name of the automation...)
- Reference id\* (Input field: Enter the JIRA/Story ID...)
- Category (Input field: Please select category)
- Reuseability Assessment (Input field: Select assessment range)
- Description\* (Input field: Brief about your automation in no more than 4000 words..)
- Portability Consideration\* (Input field: Prerequisite before using your automation..)
- Technology Limitations (Input field: Technology limitations for adopting the automation)
- Links or Attachments (Input field: Knowledge article or document link..)
- Automation Lead Name\* (Input field)
- SME/Developer Name\* (Input field)
- Automation-Go-Live Date\* (Input field: mm/dd/yyyy)
- Automation Sector\* (Input field)
- Tags\* (Input field: Please provide up to 3 tags separated by comma)

At the bottom are three buttons: Submit (green), Clear (blue), and Close (pink).

Figure 4.14 - Registration Form

On selecting a category for the automation amongst the options - 'Incident', 'Run', 'Request' etc.. a small division section appears below it describing what the specific category is. Users can hence better categorize their automation and see their automation in that specific section when added to the catalog.

The ‘Submit’ button is disabled until all the required fields marked with asterisks are entered with some value. On clicking the submit button, an alert message pops-up with the message - ‘Thank you for registering your automation. It will be added to the catalog after approval by the central team.’

On submitting, the form action method is called and an HTTP- POST method is called and the form object is passed as parameter. On clicking the ‘Clear’ button, the form gets cleared and on clicking the ‘Close’ button, users are redirected to the Home Page.

Users and the central team get an email with the form details entered by the user and after verification by the central team, it is added to the catalog and the user is notified. Emails are generated and sent using Flask Python.

#### 4.3.6 Subscription Page

Users/ Automation developers can subscribe to the catalog and get notified every time new automations get added to the catalog. As routing is enabled, this page can be directly accessed using the URL - <http://goto/toc-automation-catalog/subscribe>. This opens the page as shown in the figure 4.15.

The screenshot shows a web page titled "Technology Operations Center - Automation Catalog". At the top right is a search bar with placeholder text "Search Automation..." and a magnifying glass icon. Below the header are navigation links: "Home", "Register Your Automation", "User Guide", "Subscribe", and "Contact Us". The main content area has a heading "Subscribe to the Catalog" and a sub-instruction "Subscribe to get informed everytime new automations are added to the catalog!". A text input field is labeled "Please provide your email-id\*" and contains a placeholder "Email ID". At the bottom are three buttons: a green "Subscribe" button, a blue "Clear" button, and a pink "Close" button.

Figure 4.15 - Subscription Form

Users can provide their '@ubs.com' email address only. Entering any other email will pop up an alert message saying 'Please enter your UBS email id only.' This is done using a regex check to match the regex - '/@ubs.com\$/'

After submitting their email-id, it is checked if the user has already subscribed by verifying with the subscriber database. If already subscribed, an alert message pops-up saying 'You have already subscribed to the catalog.' If not, an HTTP Post method is called, and the email address is passed as the parameter - `this.http.post(this.api + '/subscription/' + subscriberemail)`.

On subscribing, users are sent a welcome email directly to their email address. This is done using Flask Python.

The Form is built on Angular as an NgForm. The 'Subscribe' button is initially disabled and only after entering some text, the button gets enabled. On clicking the 'Clear' button, the form is cleared and on clicking 'Close' button, users are redirected to the home page.

**Benefit of subscribing** - Users get notified on a weekly basis if any automations are added to the catalog and in the email, all details of the automations are listed along with the links to access them. For example, to access a new automation added to the catalog with catalog\_id- '1' users are provided with the link - <http://goto/toc-automation-catalog/1>, which on clicking will directly open the page as shown in the Figure 4.12. This feature enables easy sharing and navigation amongst developers and teams.

#### 4.3.7 Contact us Page

Users can access the page using the URL - <http://goto/toc-automation-catalog/contact-us>. Users can use this form to ask queries or to provide feedback.

This Angular form requests users to provide their UBS email address only and their message.



Figure 4.16 - Contact us Form

Again, the form is submitted using the HTTP- POST method and the central team (we) get the user query in the team email and the user can be helped post that.

User's can only enter their UBS email address and this is done using a simple regex check. The regex used for matching - '/@ubs.com\$/'.

The 'Send' button is initially disabled and only on entering both the required fields, it is enabled. 'Clear' button clears the form and 'Close' button redirects the page to the Home Page.

### 4.3.8 User Guide

The user guide is a pdf document that is downloaded when user clicks on the 'User Guide' button on the navbar. The document provides guided instructions how to use the catalog. It gives a brief detail of the types of automations listed in the catalog i.e. how the automations are segregated by type- incident, run, request etc.

It helps users with the registration form by describing the necessary fields. It also lists all the new features added to the catalog - subscription, sharable links of automations, link to GitHub repository etc.

## **4.4 Conclusion and Future Scope**

The catalog has seen a really quick spike in visitor count since its release (3000 hits in 3 days) and a good number of users have subscribed to the catalog. This website will revolutionize how automation will be done in UBS. It will make the development process really quick and all automations will be standardized.

Future Scope of the project-

- I) Automate addition of new automations into the database.
- II) Filter automations by execution stats, reusability considerations (high, medium and low).
- III) Further enhancement of search by introducing search by name, description, tags or all.

## **References**

- [1] [https://www.ipsoft.com/wp-content/uploads/2014/03/About\\_IPcenter.pdf](https://www.ipsoft.com/wp-content/uploads/2014/03/About_IPcenter.pdf)
- [2] <https://www.udemy.com/course/the-complete-guide-to-angular-2/>
- [3] <https://aws.amazon.com/servicecatalog/>
- [4] <https://cloud.google.com/solutions>



PRIMARY SOURCES

1	<b>Submitted to Manipal University</b> Student Paper	2%
2	<b>Submitted to Amrita Vishwa Vidyapeetham</b> Student Paper	2%
3	<b>freshservice.com</b> Internet Source	1%
4	<b>ivythesis.typepad.com</b> Internet Source	<1%
5	<b>Submitted to University of Sunderland</b> Student Paper	<1%
6	<b>www.autointelli.com</b> Internet Source	<1%
7	<b>Submitted to Universiteit van Amsterdam</b> Student Paper	<1%
8	<b>Submitted to University of Salford</b> Student Paper	<1%
9	<b>cetd.tmu.edu.tw</b> Internet Source	<1%

10	Submitted to Universiti Teknologi MARA Student Paper	<1 %
11	Theodoros Amanatidis, Alexander Chatzigeorgiou, Apostolos Ampatzoglou, Ioannis Stamelos. "Who is producing more technical debt?", Proceedings of the XP2017 Scientific Workshops on - XP '17, 2017 Publication	<1 %
12	Submitted to Coventry University Student Paper	<1 %
13	alligator.io Internet Source	<1 %
14	Submitted to University of Auckland Student Paper	<1 %
15	Submitted to Manchester Metropolitan University Student Paper	<1 %
16	Submitted to Midlands State University Student Paper	<1 %
17	www.cs.rice.edu Internet Source	<1 %
18	"Learning the Fundamentals of Designing Forms", Foundation Form Creation with Adobe LiveCycle Designer ES, 2008 Publication	<1 %

19	maxpro-tools.com Internet Source	<1 %
20	wittyanswer.com Internet Source	<1 %
21	Submitted to Middlesex University Student Paper	<1 %
22	scholarworks.boisestate.edu Internet Source	<1 %
23	Submitted to The Robert Gordon University Student Paper	<1 %
24	Submitted to East Berkshire College Student Paper	<1 %
25	Submitted to Harrisburg University of Science and Technology Student Paper	<1 %
26	Submitted to Chester College of Higher Education Student Paper	<1 %
27	Thelma Elita Colanzi, Wesley K.G. Assunção, Silvia R. Vergilio, Paulo Roberto Farah, Giovani Guizzo. "The Symposium on Search-Based Software Engineering: Past, Present and Future", Information and Software Technology, 2020 Publication	<1 %

---

Exclude quotes      On                  Exclude matches      < 3 words

Exclude bibliography      On

**Table B.1: Project Detail**

*Student Details*

Student Name	B N S VISHAL		
Registration Number	160911122	Section/Roll No.	A/21
Email Address	bnsvishal@gmail.com	Phone No.(M)	+(91) 9553551069

*Project Details*

Project Title	Ticket Enrichment using Machine learning and Automation Catalog Development		
Project Duration	5 Months	Date of Reporting	05-02-2020

*Organization Details*

Organization Name	UBS Business Solutions (India) Pvt. Ltd.		
Full Postal Ad-dress	Building 8, CommerZone, Yerwada, Samrat Ashok Path, Pune 411006, Maharashtra, India		
Website Address	<a href="http://www.ubs.com/in/en.html">www.ubs.com/in/en.html</a>		

*Supervisor Details*

Supervisor Full Name	Mr. Govindarajan VS		
Designation	Line Manager		
Full Contact Ad-dress with PIN Code	S&T- Automation COE Director, 8th Floor, Building 8, CommerZone, Yerwada, Samrat Ashok Path, Pune 411006, Maharashtra, India		
Email Address	vsgovindarajan@g mail.com	Phone No.(M)	+(91) 9673776789

*Internal Guide Details*

Faculty Name	Mrs. Anuradha Rao		
Full Contact Ad-dress with PIN Code	Department of Information and Communication Technology, Manipal Institute of Technology, Manipal-576104		
Email Address	anuradha.rao@manipal.edu		



UBS Business Solutions (India) Pvt. Ltd.

(CIN: U74999PN2015FTC157258)

Unit 501, 601, 701, 801, 901,

Bldg No.8, Commerzone,

Yerwada, Samrat Ashok Path,

Pune 411006, Maharashtra, India

Tel: +91-20-67421500

[www.ubs.com](http://www.ubs.com)

21 January 2020

**PRIVATE & CONFIDENTIAL**

Mr. Vishal Basutkar Narayana Sudheer

Dear Vishal,

Following our recent discussions, we are pleased to offer you a summer intern position in our Technology Department with UBS Business Solutions (India) Private Limited ("the Firm") under the following terms and conditions. In this letter, the term "UBS Group" means UBS Group AG and its subsidiaries and any branches, business divisions and affiliates thereof wherever incorporated or carrying on business, including the Firm, and "Group Company" means any company within the UBS Group.

**1 Assignment Period and Reporting**

Your assignment will be for the period from 3 February 2020 to 30 June 2020, or an alternate period/date mutually agreed by you and the Firm. You will report to Govindarajan Villuppanoor Srinivasan, Associate Director, Technology, or such other executive as may be nominated from time to time.

**2 Duties and Hours of Work**

Your responsibilities and duties would be intimated to you upon the commencement of your employment. In general, you will be required to work, from Monday to Friday and your actual working hours shall not be less than 40 hours per week including lunchtime, and if necessary on Public holidays or for additional hours as required for performing your duties competently and also depending upon the exigencies of work.

You shall, while undertaking your employment duties, devote the whole of your time and attention and abilities to the Firm and any other Group Company and shall use your best endeavours to promote and protect the general interests and welfare of the Firm and any other Group Company to which you may from time to time render your services.

**3 Internship Allowance**

Your internship allowance has been set to the monthly rate of INR 25,000/- per month.

**4 Compliance Induction**

To provide you with a better understanding of the Firm's compliance policy, you will be invited to attend a Compliance Induction Training Session shortly after the commencement of your employment. Please note that the Compliance Induction Training Session is a compulsory training for all new employees. You should complete such training within your employment.



IND03