

**UPTIME - "No Time for Downtime"**

**A PROJECT REPORT**

*Submitted by*

**ARJUN V PANKAJAKSHAN(TKM20MCA-2011)**

**to**

**The APJ Abdul Kalam Technological University**

*In partial fulfillment for the award of the degree of*

**MASTER OF COMPUTER APPLICATIONS**



**Thangal Kunju Musaliar College of Engineering  
Kerala**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**JULY 2022**

## DECLARATION

I undersigned hereby declare that the project report on **UPTIME - "No Time for Downtime"**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Dr. Fousia M Shamsudeen. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously served as the basis for the award of any degree, diploma, or similar title by any other University.

Kollam

18-07-2022



**ARJUN V PANKAJAKSHAN**

**DEPT. OF COMPUTER APPLICATIONS TKM COLLEGE OF ENGINEERING**

**KOLLAM**

**2021 - 22**



**CERTIFICATE**

This is to certify that the report entitled **UPTIME - "No Time for Downtime"** submitted by **ARJUN V PANKAJAKSHAN** (TKM20MCA2011) to the APJ Abdul Kalam Technological University in partial fulfillment of the Masters degree in Computer Applications is a bonafide record of the project work carried out by him under our guidance and supervision. This report, in any form, has not been submitted to any other University or Institute for any reason.

Internal Supervisor

Head of the Department

External Examiner

---

Date:19-07-2022

TO WHOM IT MAY CONCERN

This is to certify that Mr. Arjun V Pankajakshan from TKM College of Engineering, Kollam is pursuing the internship on “**Uptime-No Time for Downtime**” project at **Knowledge Lens Pvt Ltd** starting from 2<sup>nd</sup> May 2022 to till 22<sup>nd</sup> July 2022. During the period of his training with us he is found punctual, hardworking, and inquisitive.

We wish him all the success in future endeavors.

Yours Truly,

For **Knowledge Lens Pvt Ltd.**



Jayashree S

HR Manager

hr@knowledgelens.com

## **Acknowledgement**

First and foremost, I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

First and foremost, I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia M Shamsudeen**, Head of the Department, Department of Computer Applications, for providing me with best facilities.

I would like to thank my coordinator and project guide **Prof. Vaheetha Salam**, Department of Computer Applications, who motivated me throughout the project.

I would like to thank my external coordinator **Mr. Amal G Jose** and **Mr. Chandrakanth Gottam**, Knowledge Lens, who guided me throughout my work.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

**ARJUN V PANKAJAKSHAN**

## ABSTRACT

**UPTIME - "No Time for Downtime"**, is a multi-tenant smart ticketing system that receives anomalies as events from self-diagnosable critical systems of multiple OEM's which then creates tickets and assigns a support engineer based on the availability, technology, and expertise with an SLA of less than a minute. UPTIME - "No Time for Downtime" effectively reducing the time between the client and customer. It gives a single platform for monitoring the performance of teams and team members. The application also helps in having a coordinated approach to manage these anomalies. The end customer was able to get a highly available and robust system able to bring down the SLA of the alarms to minimum one minute from two minutes. With Uptime, able to reduce the massive footprint of infrastructure compared to the old system.

Six microservices connected to the message-brokering platform Kafka make up the uptime architecture. The data from the microservices will remain durable even if something awful happens to any of the microservices because Kafka sits in between them. Uptime transforms internal transactions into a single, standard format, making it simple for visualization tools like Power BI to understand the data the system produces. For notification, the system integrated notification services Twilio, Sendgrid, and Azure Push Notification. Once a ticket is created the service engineers get notified through these channels. Twilio handles SMS notifications. SendGrid for sending an email notification. Azure Push Notification deals with generating notifications on PowerBI.

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Abbreviations</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Objective . . . . .	2
1.2 Company Profile . . . . .	2
1.2.1 Products . . . . .	2
1.2.2 Services . . . . .	4
<b>2 Literature Survey</b>	<b>5</b>
2.1 Purpose of the Literature Review . . . . .	5
2.2 Related Works . . . . .	6
<b>3 Methodology</b>	<b>11</b>
3.1 Module Description . . . . .	12
3.1.1 Harmonization Engine . . . . .	12
3.1.2 Correlation Engine . . . . .	14
3.1.3 Audit Engine . . . . .	15
3.1.4 Broadcast Engine . . . . .	15
3.1.5 Notification Engine . . . . .	15
3.1.6 Elastic Engine . . . . .	15
3.2 Model Design . . . . .	15
3.3 System Specifications . . . . .	16
3.3.1 Software Specification . . . . .	16
3.3.2 Software Description . . . . .	17

<b>4</b>	<b>RESULT AND DISCUSSION</b>	<b>21</b>
4.1	Kafka Tuning . . . . .	21
4.2	High Data Throughput and Query Performance in Elasticsearch . . . . .	21
4.3	Output Screens and Results . . . . .	22
<b>5</b>	<b>CONCLUSION</b>	<b>28</b>
5.1	Future Enhancement . . . . .	28
	<b>REFERENCES</b>	<b>29</b>
	<b>APPENDIX</b>	<b>31</b>

# List of Figures

3.1	UPTIME-”No Time for Downtime” Architecture . . . . .	12
3.1	Login Page . . . . .	22
3.2	Home Page . . . . .	23
3.3	Dashboard Page . . . . .	23
3.4	Admin . . . . .	24
3.5	Harmonization . . . . .	25
3.6	Correlation . . . . .	25
3.7	Auditing . . . . .	26
3.8	User . . . . .	26
3.9	User . . . . .	27
A.1	Login Page . . . . .	31
A.2	Dashboard List View . . . . .	31
A.3	Dashboard Map View . . . . .	32
A.4	Admin . . . . .	32
A.5	Add new User . . . . .	33
A.6	Harmonization . . . . .	33
A.7	Correlation . . . . .	34
A.8	Auditing . . . . .	34
A.9	User . . . . .	35
A.10	Health . . . . .	35

# List of Abbreviations

**HDTS** Help Desk ticketing System

**OEM** Original Equipment Manufacturer

**SLA** Service Level Agreement

# Chapter 1

## Introduction

**UPTIME - "No Time for Downtime"** is a multi-tenant smart ticketing system that receives anomalies as events from self-diagnosable critical systems of multiple OEM's which then creates tickets and assigns a support engineer based on the availability, technology, and expertise with an SLA of less than a minute. It is used to solve problems associated to approach of Remote Monitoring today.

For an organization, for every minute of downtime on a crucial manufacturing line, thousands of dollars will be lost in productivity. Even though downtime is a common problem for all manufacturing facilities, the response of an organisation can have a significant impact on its productivity. It takes significant time to find out the best support expert to match an anomaly condition at a customer site, this reduces the chance to meet SLA's. Typically, a customer care or an engineer will pick up the phone, and they will decide based on some criteria who are the best expert who can help, in many cases 'Rules of Engagement' documents help with that decision, but it can take significant time from the SLA that in some cases is as aggressive as 10 minutes or shorter. For alarm monitoring, this brokering process would rely completely on an engineer. It is hard to 'sync up' what support expert are doing as they don't have visibility with each other work except for phone or chat, so there is no 'coordinated' or 'brokered' effort to take care of the anomalies. It is hard to monitor and manage multiple devices without a single stop platform and a way to monitor the teams performance in resolving the tickets.

The application helps in having a coordinated approach to manage these anomalies. The end customer was able to get a highly available and robust system able to bring down the SLA of the alarms to minimum one minute from two minutes. With Uptime, able to reduce the massive footprint of infrastructure compared to the old system.

## 1.1 Objective

The goal is to accomplish the following:

- Assigns a support engineer based on the availability, technology, and expertise with an Service-level-agreement of less than a minute.
- End customer was able to get a highly available and robust system able to bring down the SLA of the alarms between one minute to two minutes.
- Service Engineers management
- Monitor the teams performance in resolving the tickets
- Create six distinct microservices that are connected by the message broker Kafka. Elastic Engine, Broadcast Engine, Notification Engine, Audit Engine, Harmonization Engine, and Correlation Engine are the microservices.

## 1.2 Company Profile

The Lens product line from Knowledge Lens automates and simplifies the discovery of hidden insights within Big Data. The objective is to convert unstructured data into actionable business insights. We are big data technology nerds with a wide range of big data projects, including big data engineering, data science, and industry expertise.

### 1.2.1 Products

- **iLens(Intelligent Lens)**

iLens offers a centralised platform for the intelligent integration of several devices or sensors in major businesses, the manufacturing sector, the household, commercial properties, etc. In order to record time series data in real time, iLens offers a MQTT interface for smooth integration of various field sensor devices. iLens has the capacity to produce alerts and alarms depending on pre-configured rules.

- **MLens**

MLens is a one-step solution for managing disaster recovery for big data and platforms.  
Features of MLens :

1. Big Data Backup Migration
2. Automated Disaster Recovery
3. Data Encryption, compression Archival
4. High Speed Batch Data Ingestion
5. Monitoring Scheduling
6. Secured Access controls

- **AiLens**

Next-generation AI platform with a collaborative workspace, experiment designer, engineering workbench for modelling features, and integrations for enterprise security and DevOps with the repository of AI/ML assets.

AiLens is an intelligent AI assistant with a single graphical user interface for creating Data Engineering and AI/ML pipelines. AiLens' unified AI orchestrator enables users to launch model executions on any runtime, including Tensorflow, SparkML, H2O, MxNet, Theano, PyTorch, and AWS / Azure, via a centralised console. Given that every user will have the same experience, AiLens is extremely adaptable. regardless of any future technology developments due to the platform's meta model-driven design. Our software stands out from the competition by a wide margin thanks to its intuitive job submission and monitoring structure, secure integration with external entities, built-in encryption, and support for role-based access control. Key features are:

1. Any AI Stack, Any AI Algorithm, Anywhere
2. Unified AI Orchestrator
3. Simplified User Experience
4. Intelligent Assistant for AI
5. Integrated Data Preparation AI Modelling Environment
6. Seamless Enterprise Security Integration

- **GLens** GLens is available for the acquisition, monitoring, and analysis of industrial emissions, effluent discharges, and ambient air monitoring in real time. GLens DAS Software, GLens Server Platform, and GLens Environ Data Logger offer a comprehensive solution for all industrial environmental needs. Any analyzer, sensor,

or device can be connected to the platform using a plug-and-play model, and real-time data can be acquired.

The key features of GLens are::

1. Rest based open protocol for multi-client deployment.
2. Real time alerts and alarms with SMS and Email integration.
3. Remote calibration and configuration of analyzers.
4. Plug and play complete protocol integration with any analyzer make and model. – Integrated and data quality codes as per ISO 7168.
5. Integrated analytics and predictive models for effective pollution control.
6. Live consolidated industry dashboards.

### 1.2.2 Services

- **Big Data Engineering Services**

Offers complete Big Data Protects Architecture, Design, Development, Testing, and Deployment services.

- **Big Data Security Services**

We are a niche consulting firm that offers specialised Big Data Services.

- **Big Data Analytics Services**

Using our pre-built analytical Lens, we deliver hidden insights from a wide range of data sources.

- **Big Data Competency Development**

We provide one of the best Big Data Competency Development programmes for the enterprise despite the lack of unique Big Data expertise.

# Chapter 2

## Literature Survey

A literature review is a comprehensive analysis and interpretation of the relevant literature on a particular subject. Using a literature review, research questions are formulated, and then answers are sought by searching for and analysing pertinent literature. The re-analysis of the study's results frequently yields new insights, which is an advantage of literature reviews. A literature review is both a summary and an explanation of the complete and current state of knowledge on a subject, as presented in academic books and journal articles. There are two types of literature reviews that you may be asked to write in college: one is written as an independent assignment during a course. The second is an introduction or preparatory work for a longer piece, such as a thesis or research report. The type of review you are writing will determine the review's focus, perspective, and the formulation of a clear hypothesis or thesis argument. By reading published literature reviews or the introductory chapters of relevant theses and dissertations, you can learn the distinctions between these two types. Consider the structure of their arguments and how they approach the issues.

### 2.1 Purpose of the Literature Review

1. It makes research on a specific topic accessible to readers by selecting and summarising high-quality articles or studies that are relevant, meaningful, significant, and valid.
2. It offers beginning researchers in a new field an excellent starting point by requiring them to summarise, evaluate, and compare original research in that field.
3. It ensures that researchers do not repeat work already completed.

4. It may provide hints as to the direction of future research or suggest areas of emphasis.
5. It emphasises the principal findings.
6. It identifies contradictions, gaps, contradictions, and inconsistencies in the literature.
7. It provides a constructive analysis of other researchers' methodologies and approaches.

## 2.2 Related Works

Help Desk ticketing System (HDTS) comes with the whole complain management solution for any styles of business organizations. HDTS can overcome the barriers of traditional customer complain management system by easy accessing the customer information, complain and backbone. To be a successful Organization and to take care of the position customer satisfaction is key. Many organization now has branch concept and maintaining global offices, so it's really a badly have to monitor and resolve the customer complain together with internal support activities to all or any over the planet. It is essential for all businesses and services that their data be accurate and current. In contrast, manual systems are notoriously difficult due to their numerous drawbacks. This documentation will outline the HDTS software development procedure. Enterprise Architect is used as a modelling tool for the design and development of this project. EA is utilised because it will result in a better understanding of requirements, a clearer design, and systems that are easier to maintain. PHP and HTML are used as artificial languages or front-end tools, and MySQL is used as the database or back-end. HDTS is a subset of customer relationship management consisting of the practises, strategies, and technologies used by businesses to manage and analyse customer interactions and data throughout the customer lifecycle in order to improve customer service relationships, aid in customer retention, and drive sales growth. HDTS collects customer data from multiple customer touchpoints, including its website, phone, live chat, unsolicited mail, marketing materials, and social media platforms. HDTS systems can also provide customer-facing personnel with extensive personal customer information. At its most fundamental level, HDTS software consolidates customer data and documents into a single database, making access and management easier for business users. Over time, numerous functions are added to HDTS systems to increase their utility. Among these functions are logging customer interactions via email, phone, social media, and other channels; automating various workflow automation processes, such as tasks, calendars,

and alerts based on system capabilities; and enabling managers to trace performance and productivity-supported information logged within the system.[1]

To obtain product information, one must visit the company's website or contact the company via phone or email. Typically, the customer will receive the required information sent by a person. Typically, there are numerous disadvantages associated with this method. Initially, if a particular customer calls the company, he or she will be required to wait for an extended period of time. There is also a problem with customers waiting to be connected to the corporate line due to a shortage of human resources and phone lines. In contrast, if a customer sends an email, the response will be delayed because the email must be read by another employee in order to be responded to. There are few disadvantages besides the inevitable occurrence of human error. Eventually, a loyal customer may become dissatisfied and lose trust in a particular company as a result of the above negative information. Using artificial intelligence, a customer can receive any type of information in real time in the form of text/voice or e-mails. The advantages over the alternative method are that customers won't have to wait for a response, there is no room for human error, and the company can use its human resources for other tasks while the system handles at least a portion of the Customer care unit. In addition, this technique is useful for those who require immediate customer service assistance and are able to help themselves without involving a human agent from a third party.[2]

Python is a flexible and currently widely known programming language. The ability to comprehend lines of code more quickly is an essential component. The language constructs allow the user to create programs of any size, regardless of their magnitude. Python supports a variety of programming paradigms, including imperative, object-oriented, functional, and procedural programming. This is its most notable characteristic. Python utilises dynamic typing. By default, automatic memory management is enabled. There are Python interpreters for a variety of operating systems.[7]

Angular is the most popular framework for developing reactive single-page web applications and artistic website elements. Because it's the most widely used JavaScript Framework, It was definitely designed as a full-featured framework to emphasise simplicity and potency. The significance of Angular before finishing web development is discussed further below. Angular is extremely effective in terms of creating dynamic, single-page applications and supporting the MVC (Model-View-Controller) programming structure, according to developers. At a present span of some time, within the competitive business surroundings, Angular has emerged thanks

to the superhero thanks to its intuitiveness and wealthy options. Angular provides some superb capabilities out of the box and will convert your non-trivial jQuery based totally apps into one thing lots of powerful and nevertheless a lot easy to stay up with. Some frameworks merely bundle together with the current tools that build app development terribly tough. However, Angular was fastidiously designed to create sure that every tool works optimally and delivers a pleasant result.[5]

The robust, free PostgreSQL object-relational database system combines the SQL language with a number of capabilities to reliably store and scale even the most challenging data demands. PostgreSQL has been actively developed on the core platform for more than 30 years and has its roots in the University of California, Berkeley's POSTGRES project from 1986. Because of its well-proven architecture, dependability, data integrity, extensive feature set, extensibility, and the commitment of the open source community behind the software to continually delivering performant and cutting-edge solutions, PostgreSQL has established a solid reputation. Since 2001, PostgreSQL has been ACID-compliant and is compatible with all popular operating systems. It also features robust add-ons like the well-liked PostGIS geographical database extender. It's no surprise that many individuals and organisations now prefer PostgreSQL as their preferred open source relational database. It's no surprise that many individuals and organisations now prefer PostgreSQL as their preferred open source relational database.[8]

Today's general architectures for real-time data and processing suffer from excessive complexity. There are numerous technologies that must be darned and operated in tandem, and each individual technology is frequently complex in and of itself. Apache Kafka is a messaging system that is similar to a message queue. To store fault-tolerant streams of records: Kafka processes record streams in real time. Kafka is superior for two types of applications: building real-time streaming applications that make use of data streams and structuring real-time streaming data pipelines. These operations are possible because Kafka operates as a cluster on one or more servers. Each record in the Kafka cluster consists of a key, a value, and a timestamp, and is organised into topics. There are four major APIs available for Kafka: Consumer API, Producer API, Connector and API Streams API. Utilizing the Producer API to publish record streams. Consumer API is utilised for subscribing to topics and logging streams. API for streams that transforms input streams into output streams and returns the result. The Connector API executes reusable producer and consumer APIs to connect existing applications

to topics.[3]

Power BI is the market leader in business intelligence, data visualisation, and analytics. Power BI is a web-based service that allows users to find data, transform it, visualise it, and share the resulting reports and dashboards with other users in the same or different departments/organizations, as well as the general public. Power BI is used by a large number of organisations across the globe. Power BI has emerged as a serious contender for use as a business intelligence platform in small and medium-sized businesses, with a free option that includes many features and functionality. Power BI's Quick Insights feature is one of its most innovative, as it is based on an ever-expanding set of advanced analytical algorithms. After uploading a dataset to Power BI, a button click can be used to activate this feature, which automatically generates many reports based on its data analysis, with no human intervention required. This also helps to reduce human errors in calculations and statistical techniques, which result in untrustworthy research. Power BI, which accepts Excel spreadsheets as input, is easy to use and ready for adoption as a Research Data Analysis platform. This paper attempts to demonstrate how easily Power BI can transform a dataset of research data into a set of analytical reports and dashboards that can be easily shared.[4]

Large data analysis information technologies are rapidly evolving. Users in a variety of data fields require simple, adaptable, and efficient tools for processing large datasets. Multiple teams of researchers typically generate and analyse large amounts of knowledge in complex experiments. Various software packages are developed by distinct teams with varying computing infrastructure requirements. Providing a unified framework for integrating heterogeneous software into uniform application pipelines is a challenging task. These pipelines should facilitate the simple replacement and modification of software components, as well as the scalability of large-scale computing infrastructures. One of the most recent developments in this field is the combination of microservice pipeline architecture and container technologies. When combined with container orchestration, this method permits the management and scalability of containerized applications in large computing infrastructures. Container orchestration technology was first introduced by technology behemoths such as Google and Yandex for the deployment of scalable highload services. It was later adapted for a much broader range of applications. Recent container orchestration applications include scaling Machine Learning software, distributed algorithms, bioinformatics research, high energy physics, and many others. The basic concept of the HKube and Kubeflow projects

is extremely appealing: a researcher can develop software on his local system, test it with a limited dataset, and then deploy it on a large scale system with minimal effort and no changes to the software code. Another noteworthy application of this technology is the deployment of complex application pipelines and frameworks for genomic research. Installation and tuning of a knowledge processing pipeline can be difficult for a researcher who lacks sophisticated IT experience. Container orchestration frameworks such as Kubernetes are a simple and effective solution because they enable the deployment of complex software using a single configuration file and a predefined set of container images. Due to the use of docker images, it is not necessary to install a long list of software dependencies or to prepare the required application environment. Container technology cannot, however, be viewed as a panacea for all big data analysis tasks. There exists an ideal application payload, interaction patterns between software components, and data storage and transfer strategies within pipelines. Newly developed software pipelines can capitalise on effective software usage patterns in a containerized environment during the look phase. In contrast, the more frequent task is to prepare existing software for a new environment in an efficient and scalable manner, without requiring significant code modifications.[6]

With the escalating rate of thefts and crimes, security has become a major concern in the present day. Studies indicate that burglars are more likely to target homes without security systems than those with professionally monitored systems. Typically, security systems are the first line of defence against intrusions. Due to technological and Internet of Things (IoT) advancements, it is now possible to envision pervasive connectivity, storage, and automation in systems. This paper focuses on the design and development of a model for an IoT-based security system for security-required homes, banks, and offices. The proposed system includes provisions for fire and theft detection and alerts the user via text message. It also captures multiple images of the intruder and sends them to the user via email. The advantage of preferring this system over other existing systems is that the user receives alerts regardless of his location and internet connectivity. This system also uses less energy because the camera is activated and takes pictures only when an intruder is present, as opposed to the CCTV system, which records live feed continuously. This system notifies both the user and the neighbour so that prompt action may be taken.[10]

# Chapter 3

## Methodology

**UPTIME - "No Time for Downtime"** is a automated remote monitoring software to overcome the pitfalls of the current Remote Monitoring approach. It is a multi-tenant smart ticketing system that receives anomalies as events from self-diagnosable critical systems of multiple OEM's which then creates tickets and assigns a support engineer based on the availability, technology, and expertise with an SLA of less than a minute. It is used to solve problems associated approach of Remote Monitoring today.

Provided a brokering system that matches an expert based on availability, technology, geography, time of the anomaly, etc. When an anomaly is detected the right expert will be selected and it will acknowledge the selection through a mobile or web application.

By leveraging collaborative analytics, capture the expertise through a set of rules that the engineers will contribute over time to make the support more autonomous.

- The project Uptime receives alarms and TSSD data from many customers around the world.
- Alarms/Events are 'harmonized before ingested to the brokering engine.
- TSSD data includes data from PLC's, Drives, etc.
- The brokering engine leverages data from SAP's DKD (Domain Knowledge Database), Knowledge Base, etc. and determines the best resource available to resolve the issue that generated the alarm.
- Project uptime sends notification to selected engineer via mobile app and keeps track of the issue till resolution

- Support engineers will create their own rules to monitor the multi-tenant stream of data without relying on analyst to write code for them.
- There is a rules governance layer to prevent conflicting rules to jeopardize the SLA's.

### 3.1 Module Description

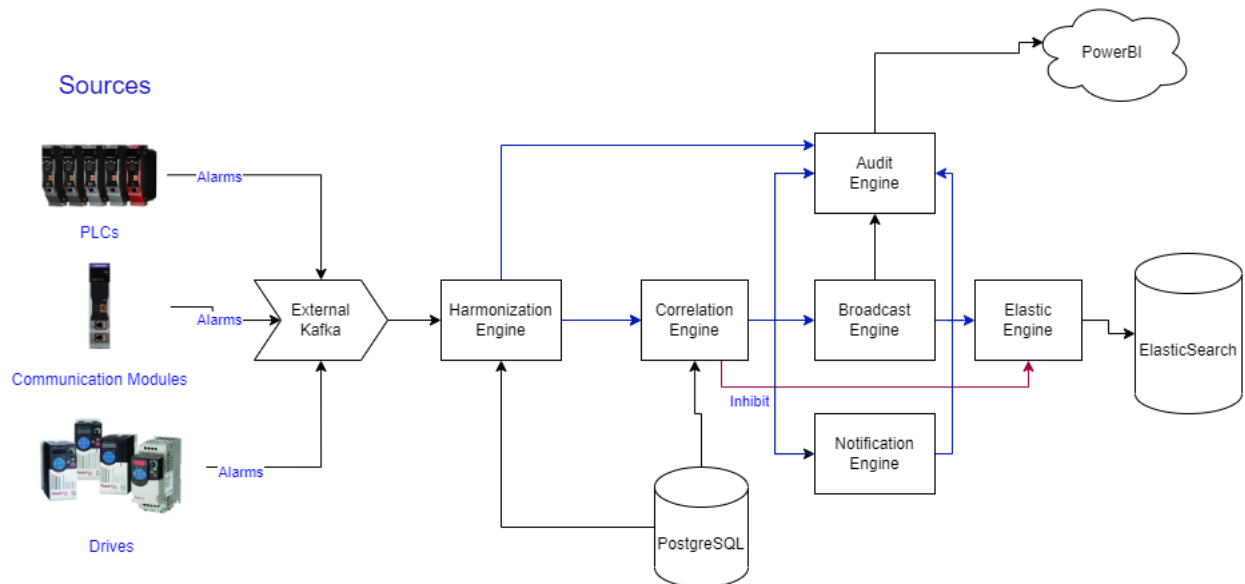


Figure 3.1: UPTIME-"No Time for Downtime" Architecture

The message broker Kafka connects the six separate microservices that make up Uptime. The microservices are the Elastic Engine, Broadcast Engine, Notification Engine, Audit Engine, Harmonization Engine, and Correlation Engine. Harmonization produces structured data from unstructured data. The correlation engine has rules that correlate the alarms OEMs produce. Audit Engine To keep track of tickets issued in several states, utilize this. Broadcast Engine where actual brokering of tickets and service engineers done based on the different conditions. Notification Engine is responsible for notifying the service engineer, Once the ticket is created. Elastic Engine is responsible for saving the status of the tickets created to the database

#### 3.1.1 Harmonization Engine

Project Uptime Brokering functionality (harmonization assembly), The primary function of this new assembly will be to collect alarms from various sources in batches. The assembly will perform alarm harmonisation on the batch before transferring it to the brokering system

via a subscription channel into a Brokering-level entry queue. The entry queue will provide confirmations to the harmonisation assembly. The acknowledgement may indicate success, failure, unavailability, etc. The harmonisation assembly will utilise the Project Uptime Brokering acknowledgments to manage the alarm batch status. The alarm batches are indexed in a manner that enables the establishment of a structured batch table structure.

Alarm harmonisation aims to establish a common schema for alarms. It is necessary to add a harmonisation envelope to the original alarm structure while maintaining the alarms' original format. This harmonisation enables the use of a common schema to organise Business Intelligence (BI) rules around it for efficient and consistent alarm processing. Then, alarms transferred from on-premises to the Data Lake correspond to alarms at the raw level.

Each raw-level alarms contains necessary information for identifying the locale of origin, alarms description and time stamp. In this process, the harmonization assembly adds time zone, technology, status, mode, and filter key data fields to the original alarms. In this, 'status' can be specified as: INPROCESS=1, WAITING=2, SERVED=3, SCALATE=4, and COMPLETED=5. And 'mode' can be specified as: FILTERED=1, CORRELATED=2, BASE=3, MANAGED=4, and MASTER=5. In this, the content of 'filter key' field can be 'Application Id' type. The harmonization assembly prepares the alarm information for the brokering function in the brokering system using this common schema. Every time an alarm or alarm set gets reworked, the rework attribute gets incremented to follow the number of tries the system has attempted in processing the alarms.

Before being transmitted to the Brokering system, each harmonized alarm is accumulated in a batch for transfer. The size of the batch and its transmission rate are configuration settings that are managed by the system administrator using the broker management application during the system configuration phase. An alarm batch contains a mix of alarms from that emerge from the various applications under monitoring per tenant basis. The customer data sovereignty is a primary condition to be kept all the way throughout the alarm processing. The harmonization assembly must be aware of data sovereignty.

The harmonization assembly must keep track of the alarm indexing as these alarms are being copied and harmonized. To keep the batch indexing information, the harmonization assembly will store the batch information in a batch indexing database. In this manner, the harmonization assembly can keep track of what data has been sent to the Brokering system. In situations where the transmission channels is not be available for transmission due to some

issue, the harmonization assembly must be able to recognize this condition to temporarily ceased transmission of the batches. In such a situation, the raw-data continues is normal accumulation flow into the RA Data Lake but no further action is taken on harmonization until the harmonization assembly is able to re-establish communication with the Brokering system. The harmonization assembly transmits the batch to the Brokering system synchronously. For each batch transmission, the harmonization assembly expects a success acknowledgement from the Brokering system. In case of interruptions or unsuccessful transmissions, the harmonization assembly will proceed to prepare the pending batch for retransmission and simultaneously it will log the event as an anomaly, logging the time and the Brokering system response. In the event of persistent unsuccessful transmission tries, the harmonization assembly will then trigger a notification to the system manager via mobile device and/or email. The harmonization assembly logs all the transactions in the batch indexing database.

### 3.1.2 Correlation Engine

This defines a set of rules to define whether alarms to be inhibited or not. Due to maintenance or production shutdown, there is the need to inhibit alarms both at the individual level and a device or process level. Based on the technology/application rules, find alarms that represent a sequence of alarms (alarms that correlate to other alarms). These types of alarms are classified as symbiotic alarms. Select a representative alarm of the set as the header and change its status to WAITING and mode to MASTER. Select all subordinate alarms belonging to the cascaded set and change their status to WAITING and their mode to CORRELATED. Assign the Application Id of header alarm to 'filter key' attribute of each correlated alarm. This technology filtered alarm set is then sent to the search engine to initiate a search for experts that understand the alarm correcting procedures. The experts are contacted throughout a handshaking process that emits messages to the experts and that is able to keep track of the communication lifecycle to continue to modify the alarm status and mode. The search engine is another worker role with multithreading capabilities to manage the various application and technology sets that will emerge from the brokering process.

### **3.1.3 Audit Engine**

This is used to keep logs of tickets in different states. The log will create when the alarm is got acknowledged, escalated, resolved. The Audit Engine saving logs in similar format, so the visualization with Power BI can easily work with the data processed by Uptime. By the data from audit engine, it is easy to process the performance of teams and the engineers in solving ticket with in Power BI.

### **3.1.4 Broadcast Engine**

It is called the heart of Uptime because here the actual brokering is takes place. Once the alarm is passed through the Correlation Engine to Broadcast Engine, It starts to find the best service engineer based on the availability, technology, and expertise with an SLA. Then it enables Notification Engine.

### **3.1.5 Notification Engine**

Once an SLA is created, the service engineer should be notified, as that Notification Engine comes in to the picture. It is responsible for creating notification to service engineer through different channels such as SMS, Email, Azure Push Notification. For SMS notification Engine activates Twilio and for email SendGrid gets activated.

### **3.1.6 Elastic Engine**

This is responsible for saving the status of the tickets created to the database. Elasticsearch. The status can be broadcasted, acknowledged, escalated or resolved. Everything saved in time series format. By data from elastic engine, the audit engine is working. With this data, health of the different device can be identified.

## **3.2 Model Design**

A large system that is dispersed across several nations need to maintained by Uptime. It is a tedious job to manage the interactions between the devices connected to the system and the service engineers. As a result, a model is needed to organize and manage the OEMs and service

engineers connected to the system. The system has a super admin that manages all admins associated with system who are doing the actual tasks of Uptime.

- Super Admin
  - Has ability to add or remove admin users.
  
- Admin
  - Access to all dashboards
  - Can see status of Tickets
  - Add Service Engineers
  - Create Teams
  - Converting raw data to structured
  - Adding Rules for correlation
  - Maintain Customers and Service Engineers
  - Can see the performance of teams in solving tickets.
  - Can see Available Service Engineers
  - Can see the health of different Equipments by location.

### **3.3 System Specifications**

The application development architecture recognized for this project is specified in this section on the basis of requirements.

#### **3.3.1 Software Specification**

- Python - for building microservices
- Angular Js - Front end
- PostgreSQL - Database
- Kafka - for brokering platform

- Azure Event Hub - for integration with Power BI
- Spark Streaming - Core Engines (Harmonize and Correlate data)
- Web Browser : Any web browser
- Kubernetes - Platform as a service
- Sendgrid, Twilio, Azure Push Notification - Notify Engineers

### 3.3.2 Software Description

- **Python**

Python is a widely used general purpose, high level programming interactive, interpreted, object-oriented programming language. By python language developer can easily express concepts in few lines of codes. Python has has large amount of library and a huge community support. Python supports in most of the operating system such as Windows, Linux, macOS.

- **Angular**

Angular is a TypeScript-based, open-source framework for developing web-based applications. It is currently one of the most popular UI frameworks. With Angular, it is simple to create faster web applications, and Angular is a platform that makes it simple to create web applications. Angular provides a simple and powerful template syntax for rapidly generating UI views. The main features of Angular are ,It is component based, Using TypeScript, Using concepts of Services that could be used across multiple applications

- **PostgreSQL**

PostgreSQL is a powerful object relational database. It has a powerful community support, because it is developing since last 30 years with a powerful open source community. Nowadays most of the companies like Uber, Netflix, Spotify, Instagram are using PostgreSQL in their infrastructure. PostgreSQL can be accessed by command line tool psql and or postgres client PgAdmin.

- **Kafka**

Apache Kafka is a platform for message brokering. This is used to connect various microservices within a single application. Data persistence can be guaranteed with Kafka. Producer, Consumer, Streams, and Connect are the four APIs provided by Kafka. Using Kafka as a message buffer to protect a legacy database that cannot handle modern workloads, or the Connect API to keep said database in sync with an accompanying search indexing engine, or the Streams API to process data as it arrives and return aggregated data to your application. Kafka and its APIs simplify the development of data-driven applications and the management of complex back-end systems. Kafka guarantees that your data is always fault-tolerant, replayable, and in real time. By providing a single event streaming platform to process, store, and connect your apps and systems to real-time data, we can assist you in building more quickly.

- **Azure Event Hubs**

Azure Event Hubs is a platform for the streaming of large amounts of data and an event ingestion service. It is capable of receiving and processing a large number of events per second. Any real-time analytics provider or batching/storage adapter is capable of transforming and storing event hub data. Only when it is simple to process and obtain timely insights from data sources is data valuable. Event Hubs provides a distributed stream processing platform with low latency and seamless integration with data and analytics services inside and outside Azure in order to construct a comprehensive big data pipeline. Event Hubs serve as the "front door" to an event pipeline, which in solution architectures is also known as an event ingestor. An event ingestor is a component or service that decouples event production from event consumption by sitting between event publishers and event consumers. Event Hubs is a unified streaming platform with a buffer of time retention that separates event producers and event consumers.

- **Microsoft Power BI**

Microsoft Power BI is a business intelligence platform that provides tools for nontechnical business users to aggregate, analyse, visualise, and share data. The user interface of Power BI is fairly intuitive for Excel users, and its deep integration with other Microsoft products makes it a very versatile self-service tool that requires little upfront training. Microsoft Power BI is used to discover patterns in an organization's data. Power BI can assist in connecting disparate data sets, transforming and cleaning the data into a data model, and creating charts or graphs to visualise the data. All of this information can be shared with other Power BI users in your organisation. Power BI data models can be used in a variety of ways for organisations, including telling stories with charts and data visualisations and examining "what if" scenarios within the data. Power BI reports can also provide real-time answers to questions and assist with forecasting to ensure departments meet business metrics. Power BI can also provide executive dashboards for administrators or managers, allowing management to gain a better understanding of how departments are performing.

- **Kubernetes**

Kubernetes, also known as "K8s," orchestrates the execution of containerized applications across a cluster of hosts. The K8s system automates the deployment and management of cloud native applications on-premises or in the public cloud. It distributes application loads across a Kubernetes cluster and its nodes and handles dynamic container networking requirements automatically. Kubernetes also assigns storage and persistent volumes to running containers, performs automatic scaling, and works continuously to keep applications in the desired state, thereby providing resiliency. Kubernetes has many features that help orchestrate containers across multiple hosts, automate K8s cluster management, and maximise resource usage through better infrastructure utilisation. Auto-scaling, Lifecycle management, Declarative model, Resilience and self-healing, Persistent storage, Load balancing, and DevSecOps support are all important features.

- **SendGrid**

SendGrid is a cloud-based Simple Mail Transfer Protocol provider that is used for send email notifications without the need of email servers. It handles all technical aspects, in-

cluding infrastructure scaling, ISP outreach, reputation monitoring, whitelisting services, and real-time analytics.

- **Twilio**

Messaging Services is a free Twilio feature that allows to send messages at scale and across multiple countries. A Messaging Service, in short, is a container for multiple Twilio message senders (e.g. phone numbers, WhatsApp senders). When the Messaging Service information is passed to the API requests instead of a specific From number, it provides with additional message sending intelligence and content features, such as automatic number selection. Setting up one or more Messaging Services for use cases ensures that it can easily expand the messaging use case with Twilio.

# Chapter 4

## RESULT AND DISCUSSION

### 4.1 Kafka Tuning

Implemented some tuning to enhance the performance of the cluster. There are two primary metrics to consider when doing so: Throughput refers to the number of messages that arrive within a given time period. As the number of published messages increases, it is necessary to scale up brokers, storage, and consumers to avoid bottlenecks. Latency is the time required to process each message. High latency in the data pipeline will result in a significant delay in the message consumption. If there are insufficient brokers to meet the required throughput, producers' requests to brokers will timeout, resulting in message loss. This could become a significant pain point for your business, so balancing throughput and latency within a Kafka cluster is crucial.

### 4.2 High Data Throughput and Query Performance in Elasticsearch

When massive amounts of data are continuously ingested into an index, the overhead increases and query efficiency decreases. Given the rate at which technology is expanding, we can only anticipate an increase in data, which would result in a decline in performance over time. Utilized Elasticsearch's ILM function to solve this specific issue. ILM guarantees high query performance and reduced costs by periodically reindexing the data. As indexes age and are queried less frequently, they can be migrated to a less costly disc, reducing the number of

primary and replica shards. ILM can also roll over an index automatically if the policy condition is met. Roll an index over based on the number of days, months, the year, documents, and storage space.

## 4.3 Output Screens and Results

### 1. Login Page

User can log in by giving username and password.

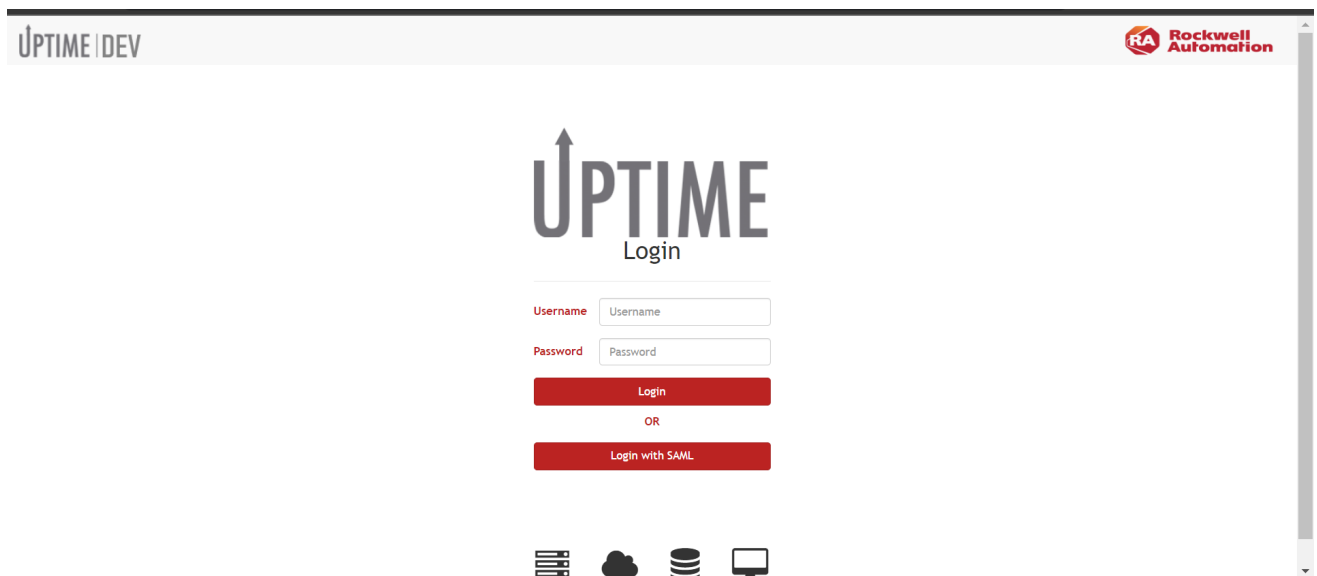


Figure 3.1: Login Page

### 2. Home Page

Landing page after user login. Shows Status like Broadcasted, Acknowledged, Escalated, Resolved of the tickets created.

The screenshot shows the 'My Status' page in the Uptime IDEV application. At the top, there's a navigation bar with the application name 'UPTIME IDEV' and the Rockwell Automation logo. Below that, a status bar shows various locations and their current times: Melbourne/Brisbane: 07/07 15:00:12, Singapore: 07/07 13:00:12, Katowice: 07/07 07:00:12, Cleveland: 07/07 01:00:12, Sao Paulo: 07/07 02:00:12, Bogota: 07/07 00:00:12. The main navigation menu includes Home, Dashboard, Admin, Harmonization, Correlation, Reminders, Auditing, User, and Health. The user is logged in as Amal Jose.

The 'My Status' section shows a 'Dark' theme toggle, 'My Teams' dropdown, and 'Last Update At: Jul 07 2022, 10:30:12'. There are buttons for 'Configure Columns', 'Delete Alarms', 'Team Alarms', 'All Alarms', and 'Refresh'. Below this, there are tabs for 'Broadcasted', 'Acknowledged', 'Escalated', and 'Resolved'. The main content is a table of alarms with columns: Customer, Site, Alarm Description, Device, Severity, Technology, Detected Time, Uptime Time, and Action. The table shows several critical alarms for FedEx customers, all related to 'Connectivity [Packet Loss/RTT/DNS Resolution]'. The severity is 'Critical' and the technology is 'ISM - IDC'. The detected time is 'May 17 2021, 18:34:39'. The uptime time is 'May 17 2021...'. The action column has 'Preview' and 'Group' buttons. At the bottom, there's a pagination control showing 'Showing Items: 6' and '100 items per page'.

Figure 3.2: Home Page

### 3. Dashboard Page

Shows all details of different IOT devices. Can search IOT devices.

The screenshot shows the 'Dashboard' page in the Uptime IDEV application. The navigation bar is similar to the Home page, but the 'Dashboard' menu item is highlighted. The user is still logged in as Amal Jose. The main content area has a search bar and a filter dropdown set to 'All'. Below the search bar, there's a table of IOT devices with columns: Customer Name, Site Name, Technology, Availability, Modified On, and Action. The table lists various customers and their sites, such as 'Agro Industrial San Carlos', 'Alnylam', 'Anv Support Monitoring', 'APODI', 'Asian Paint PPG', 'Baypines Enterprises', 'Bimbo Bakeries', and 'MONTEBELLO'. The availability column shows '1 - Up' or '2 - Down'. The modified on column shows dates and times. The action column has 'Inhibit' and 'History' buttons. At the bottom, there's a pagination control showing 'Showing Items: 100' and '100 items per page'.

Figure 3.3: Dashboard Page

### 4. Admin Page

Management of Customers and Teams.

The screenshot displays the 'Admin' section of the Uptime IDEV application. At the top, there is a navigation bar with the 'UPTIME IDEV' logo on the left and the 'Rockwell Automation' logo on the right. Below the logo, a status bar shows server locations and their last update times: Melbourne/Brisbane: 07/06 23:30:17, Singapore: 07/06 21:30:17, Katowice: 07/06 15:30:17, Cleveland: 07/06 09:30:17, Sao Paulo: 07/06 10:30:17, and Bogota: 07/06 08:30:17. A secondary navigation bar includes links for Home, Dashboard, Admin (highlighted), Harmonization, Correlation, Reminders, Auditing, User, and Health. A user greeting 'Welcome Amal Jose' and a 'Logout' link are also present.

Below the navigation, a breadcrumb trail shows 'Customers' as the active page, with other options like Sites, Contracts, Severities, Technologies, Business Units, Teams, Unharmonized Alarms, and Inhibited Alarms. The main content area features a '+ New Customer' button and a dropdown menu set to 'Active'. A table lists customer names with checkboxes for selection and an 'Action' column containing 'Edit' and 'De-Activate' icons. The table includes entries for Green Bay Packaging, OMMSA, Gelita, Grain Processing Corp, Yara, Incyte Biosciences, Bristol Myers Squibb, FedEx, and Atlas Tiha. At the bottom of the table, it indicates 'Showing items: 148' and '1 - 148 of 148 items'.

Figure 3.4: Admin

### 5. Harmonization Page

Converting raw data to structured format.

The screenshot shows the 'Harmonization' page in the Uptime Dev application. At the top, there are status indicators for various locations: Melbourne/Brisbane: 07/06 23:30:57, Singapore: 07/06 21:30:57, Katowice: 07/06 15:30:57, Cleveland: 07/06 09:30:57, Sao Paulo: 07/06 10:30:57, and Bogota: 07/06 08:30:57. The navigation menu includes Home, Dashboard, Admin, Harmonization (active), Correlation, Reminders, Auditing, User, and Health. A sub-menu for 'Harmonization' includes Data Source, Site, Technology, and Severity. A '+ New Site' button is visible. The main table has columns for Data Source, Raw Site, Customer, Harmonized Site, TimeZone, and Action. The table lists several entries, including Thingworx, SolarWinds, and Bimbo Bakeries. Each row has 'Edit' and 'Delete' icons. At the bottom, it shows 'Showing Items: 500' and '1 - 500 of 560 Items'.

Data Source	Raw Site	Customer	Harmonized Site	TimeZone	Action
Thingworx	Green Bay	Green Bay Packaging	Green Bay	America/Menominee	Edit Delete
SolarWinds	372	PepsiCo	PEP_WS1_PBNA_USAM951_GAT_DALLAS, TX	America/Indiana/Knox	Edit Delete
SolarWinds	375	PepsiCo	PEP_WS1_PBNA_USAM958_GAT_KISSIMMEE, FL	America/New_York	Edit Delete
SolarWinds	444	PepsiCo	PEP_WS2_ESSA_TIMASHEVSK_HS_5166	Europe/Moscow	Edit Delete
SolarWinds	689	PepsiCo	York	America/New_York	Edit Delete
SolarWinds	837	PepsiCo	PEP_WS2_LATAM_MEX1009_CELAYA, MX	America/Bogota	Edit Delete
SolarWinds	802	PepsiCo	PEP_WS1_ESSA_POLA011_GRODZISK, POLAND	Europe/Minsk	Edit Delete
SolarWinds	828	Bimbo Bakeries	Northumberland	America/New_York	Edit Delete
SolarWinds	861	KraftHelm	IRVN	America/Phoenix	Edit Delete

Figure 3.5: Harmonization

### 6. Correlation Page

The rules for brokering are adding from here. All rules are listed here.

The screenshot shows the 'Correlation' page in the Uptime Dev application. At the top, there are status indicators for various locations: Melbourne/Brisbane: 07/06 23:31:43, Singapore: 07/06 21:31:43, Katowice: 07/06 15:31:43, Cleveland: 07/06 09:31:43, Sao Paulo: 07/06 10:31:43, and Bogota: 07/06 08:31:43. The navigation menu includes Home, Dashboard, Admin, Harmonization, Correlation (active), Reminders, Auditing, User, and Health. A sub-menu for 'Correlation' includes Add Profile. The main table has columns for Name, End Date, Technology, Description, and Action. The table lists three entries: Metso, ISM-OEE, and ISM Default. Each row has 'Edit' and 'De-Activate' icons. At the bottom, it shows 'Showing Items: 3' and '1 - 3 of 3 Items'.

Name	End Date	Technology	Description	Action
Metso	2120-05-01 23:59:59+05:30	Metso Infrastructure		Edit De-Activate
ISM-OEE	2036-12-29 23:59:59+05:30	ISM - OEE	Rules for Advanced Technology team OEE Technology.	Edit De-Activate
ISM Default	2030-07-01 23:59:59+05:30	ISM - Default		Edit De-Activate

Figure 3.6: Correlation

### 7. Auditing Page

The status logs of all tickets are visible here.

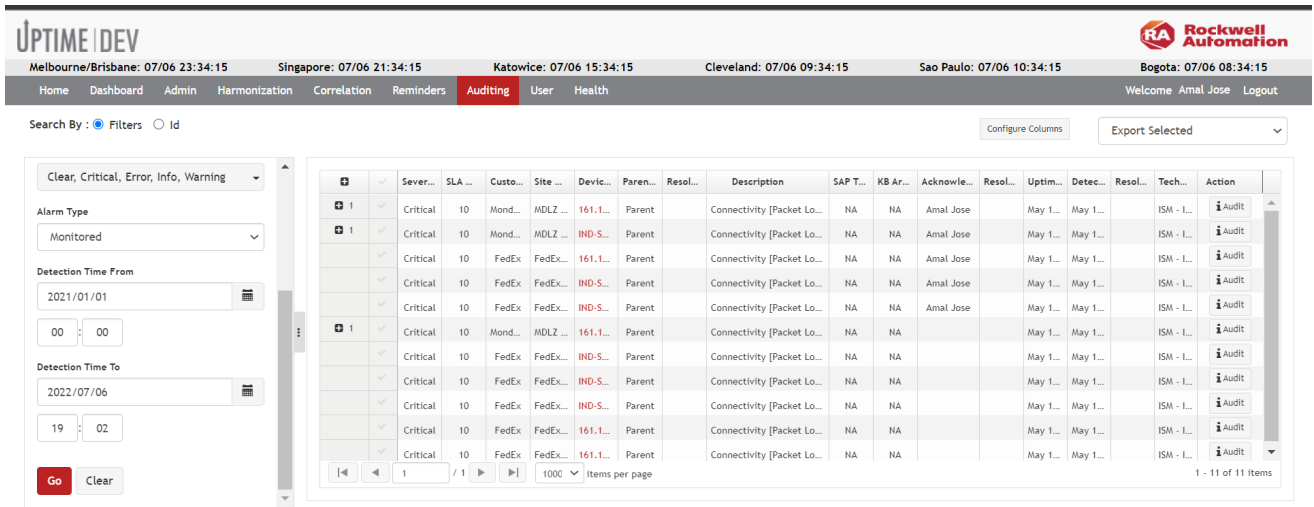


Figure 3.7: Auditing

### 8. User Page

The Service Engineers and teams are listed here. The availability of Service engineers is also visible from here.

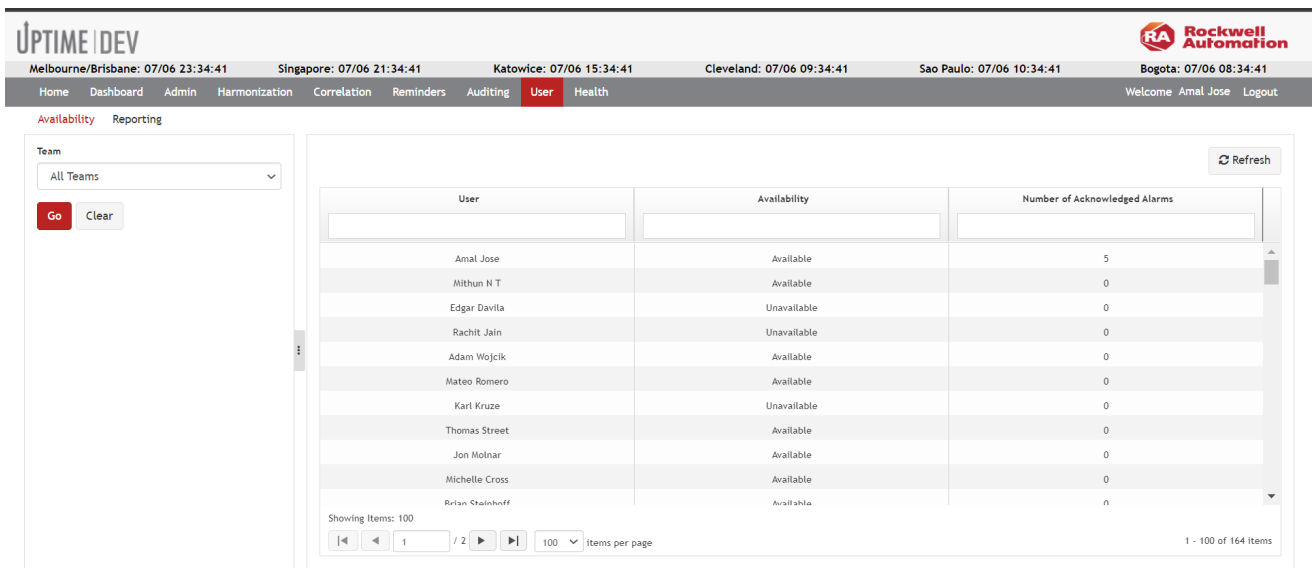


Figure 3.8: User

### 9. Health Page

The health of all IOT devices are showing here.

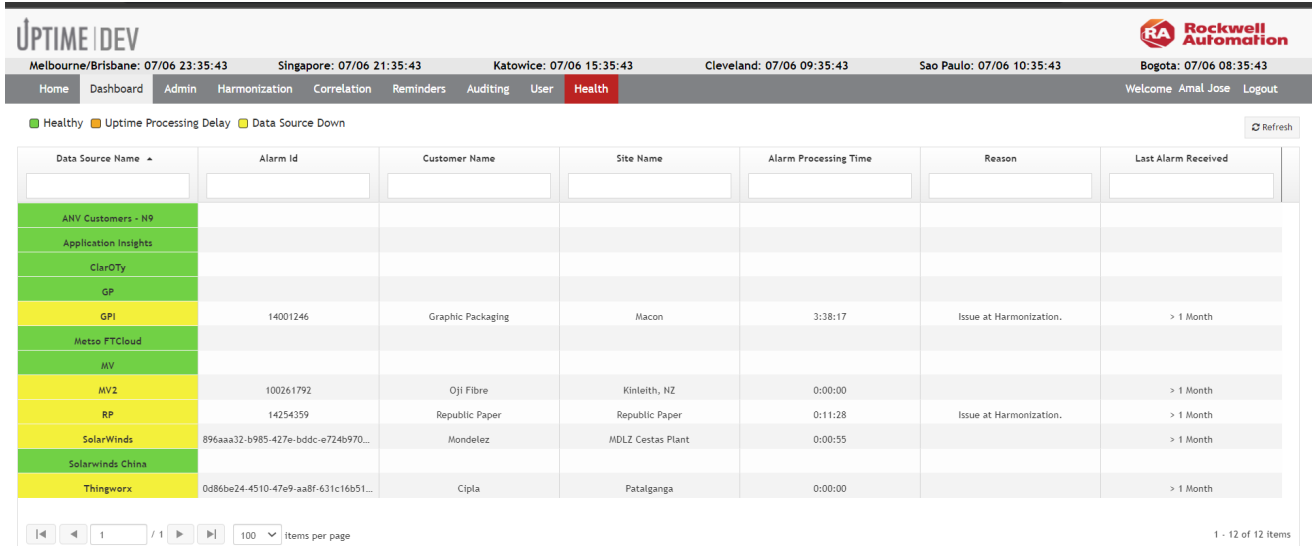


Figure 3.9: User

# Chapter 5

## CONCLUSION

The project is a smart ticketing system that receives anomalies as events from self-diagnosable critical systems of multiple OEM's which then creates tickets and assigns a support engineer based on the availability, technology, and expertise with an SLA of less than a minute .This overcome the pitfalls of the current customer care ticket booking system. The first attempt was made to determine the need for the system. To meet the needs of the users, a detailed study was created in such a way that it is user friendly and simple to use. This particular system has been designed in such a way that even a user with limited knowledge can easily operate it.

### 5.1 Future Enhancement

New modules may be easily added to the system because of the way it is constructed. The system will be more flexible as a result of the reconstruction. With the latest capabilities of this technology in mind, the system has been created to be as adaptable and user-friendly as possible. All SRS requirements have been implemented in the system as of this point.

# REFERENCES

- [1] Monzur Alam,"Project Report On Help Desk Ticketing System (HDTs)", 2018,*Institute Of Information Technology (IIT)Jahangirnagar University,Savar*
- [2] H.P.Ranawaka,A.K Warnapura, D.S Rajapaksha, P.S.S.J.Fernando, K.T.S.Kasthuriarachchi, D.Wijendra "Automated Customer Care Service System for Finance Companies",2017,*Department of Information Technology, Sri Lanka Institute of Information Technology.*
- [3] javed Ahmed Shaheen, "Apache Kafka: Real Time Implementation with Kafka Architecture Review",2017,*International Journal of Advanced Science and Technology*
- [4] G Krishnamoorthy, Vijay rishnan S, Bharanidharan, " Research Data Analysis with Power BI ",2017,*11th International CALIBER-2017 Anna University,Chennai, Tamil Nadu*
- [5] Miss.Vaibhavi A Dhopate, Dr.Rahul J Jadhav "Role Of Angular In Web Development",2021,*Journal of Emerging Technologies and Innovative Research (JETIR)*
- [6] Alexey Poyda and Vasily Velikhov,Sergey Bobkov,Anton Tesliuk, Viacheslav Ilyin, Alexander Novikov, " Kubernetes container orchestration as a framework for flexible and effective scientific data analysis",2019,*Ivannikov Ispras Open Conference (ISPRAS)*
- [7] Deepak Sharma,Akshansh Sharma,Firoj Khan,Dr.Sunil Gupta "Python: The Programming Language of Future",2020,*International Journal of Innovative Research in Technology*
- [8] Lawrence A.Rowe,Michael Stonebraker, and Michael Hirohama "The Implementation of Postgres",2017,*IEEE Transactions on Knowledge and Data Engineering*
- [9] O.Vinati Reddy,Nikita Kathare,Dr.Vishalakshi Prabhu "A Comprehensive Study of Elasticsearch ",2020,*International Journal of Science and Research (IJSR)*

- [10] S Varsha, A Jawahar, Supreeta Venkateshan, Roshne N "*Design and Implementation of an Automated Security System using Twilio Messaging Service, 2017 International Conference on Smart Cities, Automation Intelligent Computing Systems*"

# APPENDIX

## Screenshots

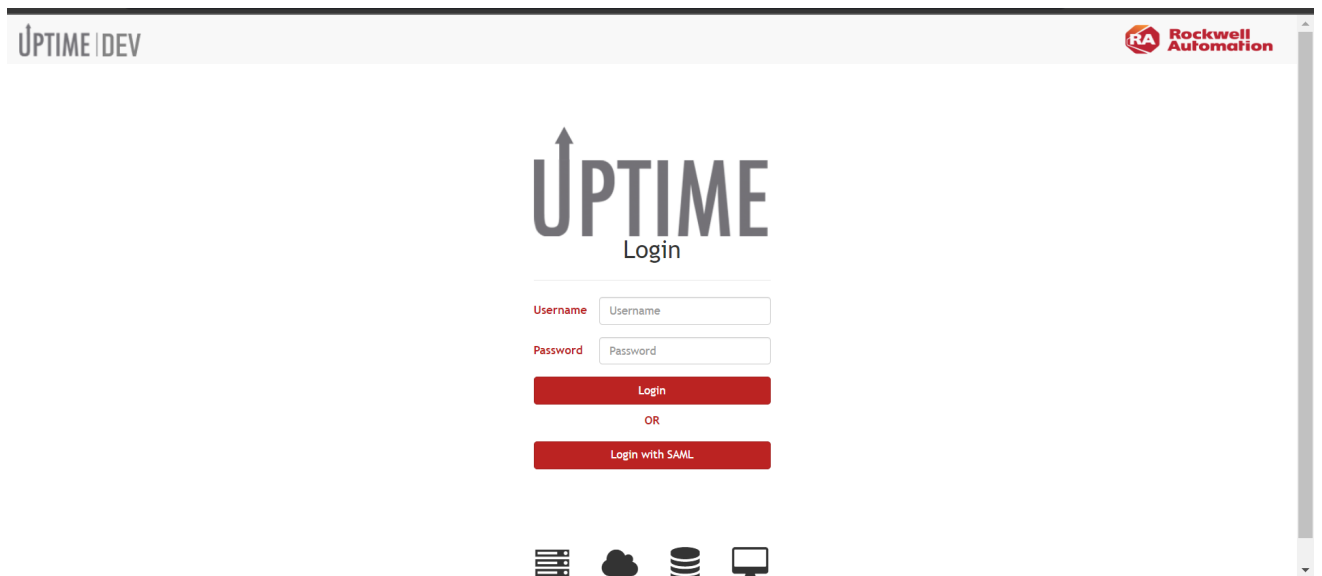


Figure A.1: Login Page

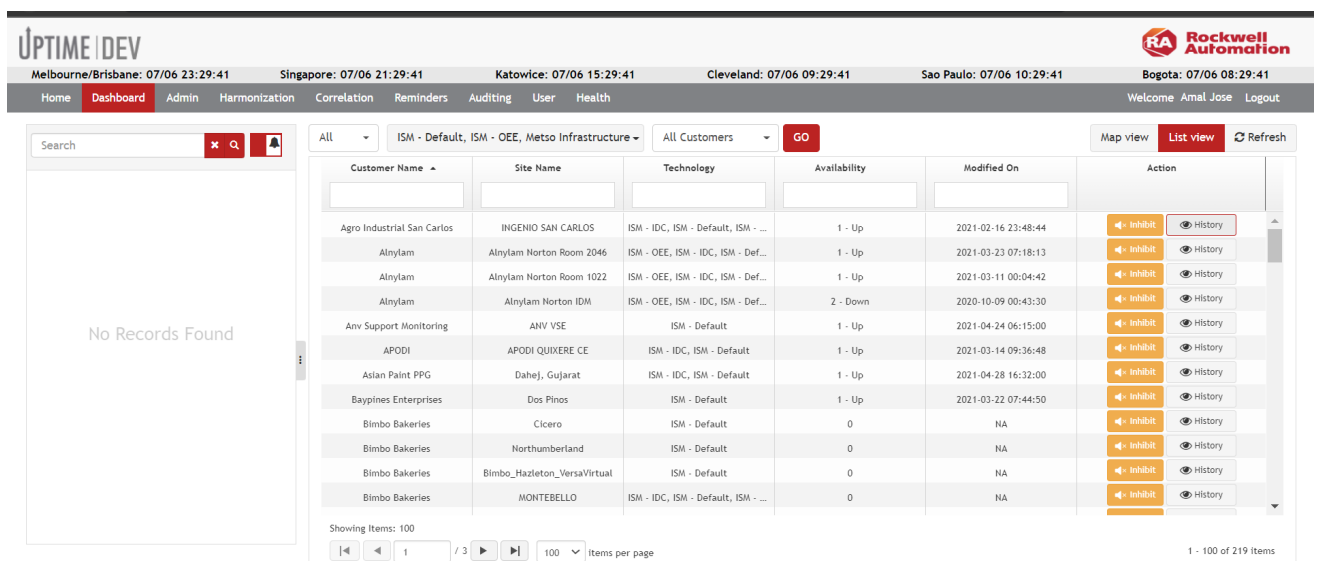


Figure A.2: Dashboard List View

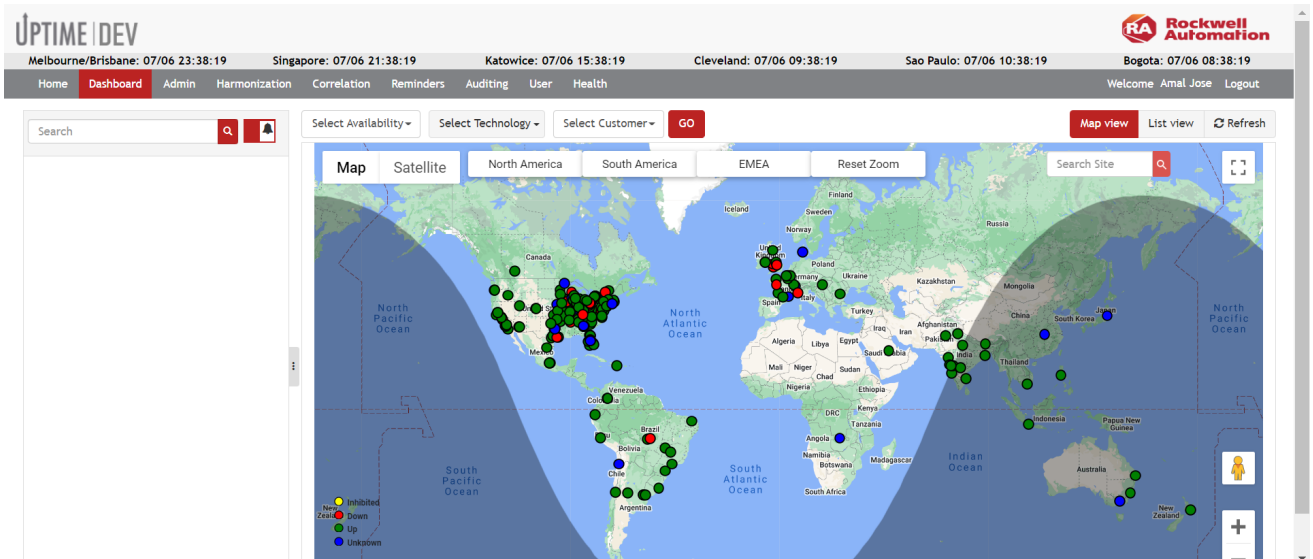


Figure A.3: Dashboard Map View

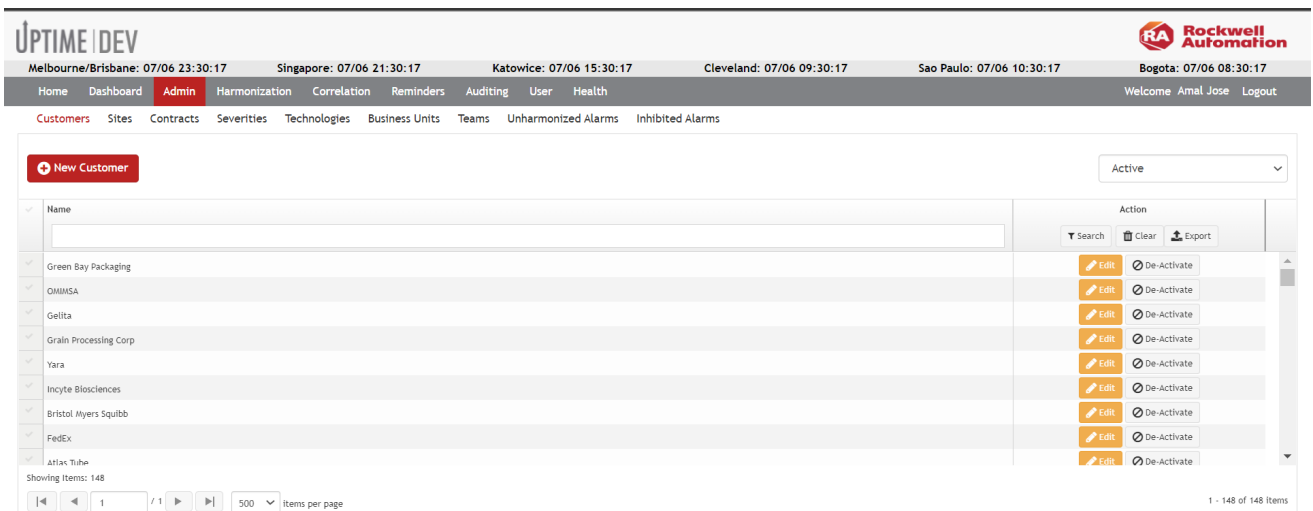


Figure A.4: Admin

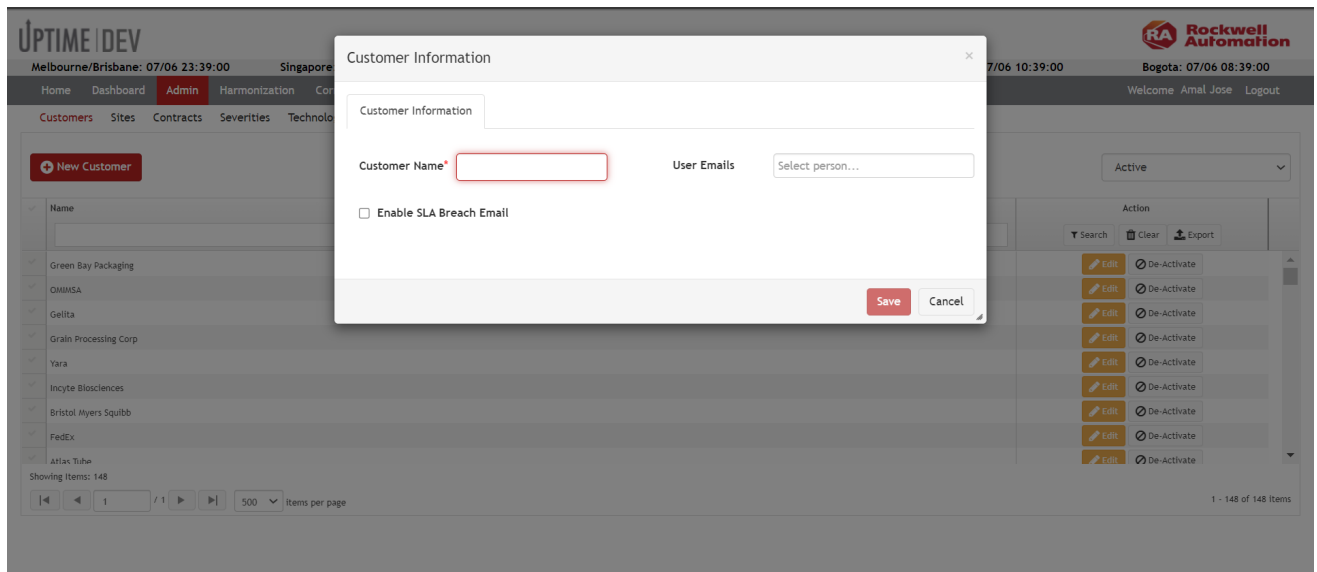


Figure A.5: Add new User

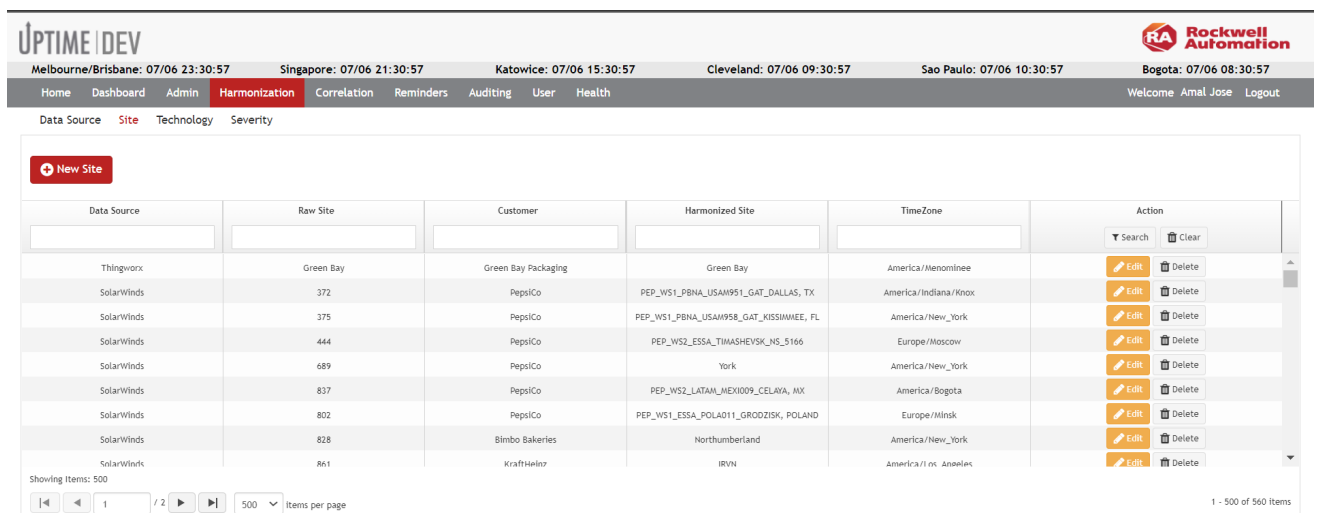


Figure A.6: Harmonization

**UPTIME IDEV** Rockwell Automation

Melbourne/Brisbane: 07/06 23:31:43   Singapore: 07/06 21:31:43   Katowice: 07/06 15:31:43   Cleveland: 07/06 09:31:43   Sao Paulo: 07/06 10:31:43   Bogota: 07/06 08:31:43

Home   Dashboard   Admin   Harmonization   **Correlation**   Reminders   Auditing   User   Health   Welcome Amal Jose   Logout

**Correlation**

Name	End Date	Technology	Description	Action
Metso	2120-05-01 23:59:59+05:30	Metso Infrastructure		<a href="#">Edit</a> <a href="#">De-Activate</a>
ISM-OEE	2036-12-29 23:59:59+05:30	ISM - OEE	Rules for Advanced Technology team OEE Technology.	<a href="#">Edit</a> <a href="#">De-Activate</a>
ISM Default	2030-07-01 23:59:59+05:30	ISM - Default		<a href="#">Edit</a> <a href="#">De-Activate</a>

Showing Items: 3   1 / 1   500 items per page   1 - 3 of 3 items

Figure A.7: Correlation

**UPTIME IDEV** Rockwell Automation

Melbourne/Brisbane: 07/06 23:34:15   Singapore: 07/06 21:34:15   Katowice: 07/06 15:34:15   Cleveland: 07/06 09:34:15   Sao Paulo: 07/06 10:34:15   Bogota: 07/06 08:34:15

Home   Dashboard   Admin   Harmonization   Correlation   Reminders   **Auditing**   User   Health   Welcome Amal Jose   Logout

Search By:  Filters    Id   [Configure Columns](#)   [Export Selected](#)

Clear, Critical, Error, Info, Warning

Alarm Type: Monitored

Detection Time From: 2021/01/01

Detection Time To: 2022/07/06

Sever...	SLA...	Custo...	Site ...	Devic...	Paren...	Resol...	Description	SAP T...	KB Ar...	Acknowle...	Resol...	Uptim...	Detec...	Resol...	Tech...	Action
Critical	10	Mond...	MDLZ ...	161.1...	Parent		Connectivity [Packet Lo...	NA	NA	Amal Jose		May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	Mond...	MDLZ ...	IND.S...	Parent		Connectivity [Packet Lo...	NA	NA	Amal Jose		May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	FedEx	FedEx...	161.1...	Parent		Connectivity [Packet Lo...	NA	NA	Amal Jose		May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	FedEx	FedEx...	IND.S...	Parent		Connectivity [Packet Lo...	NA	NA	Amal Jose		May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	Mond...	MDLZ ...	161.1...	Parent		Connectivity [Packet Lo...	NA	NA			May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	FedEx	FedEx...	IND.S...	Parent		Connectivity [Packet Lo...	NA	NA			May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	FedEx	FedEx...	IND.S...	Parent		Connectivity [Packet Lo...	NA	NA			May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	FedEx	FedEx...	161.1...	Parent		Connectivity [Packet Lo...	NA	NA			May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	
Critical	10	FedEx	FedEx...	161.1...	Parent		Connectivity [Packet Lo...	NA	NA			May 1...	May 1...	ISM - I...	<a href="#">i Audit</a>	

1 - 11 of 11 items

Figure A.8: Auditing

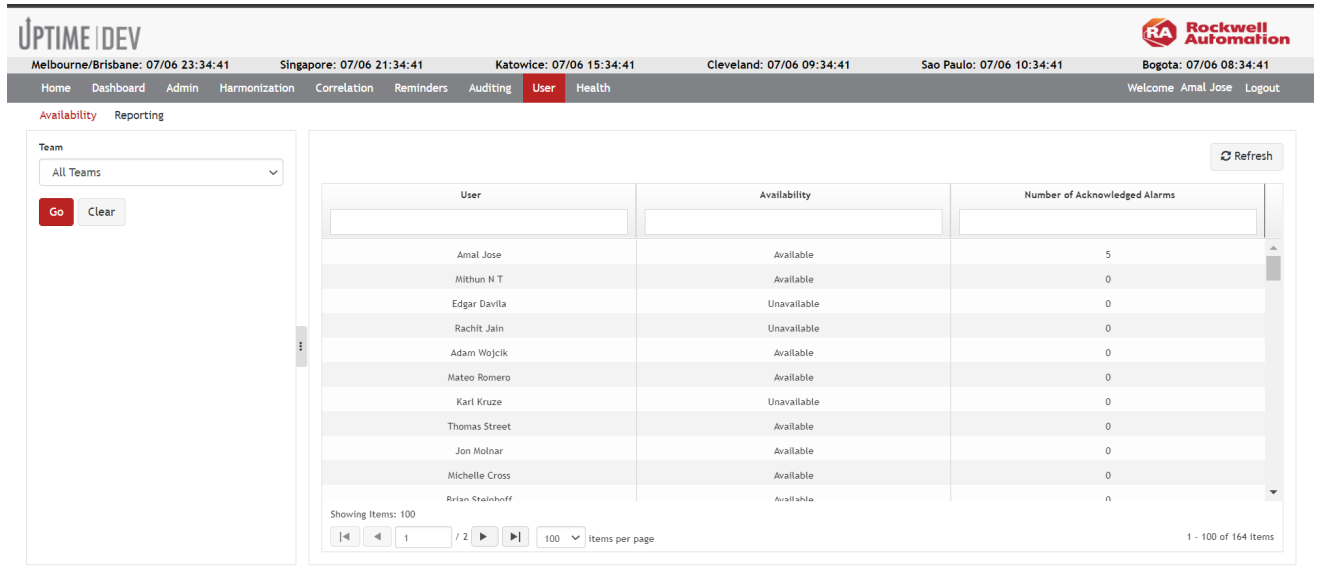


Figure A.9: User

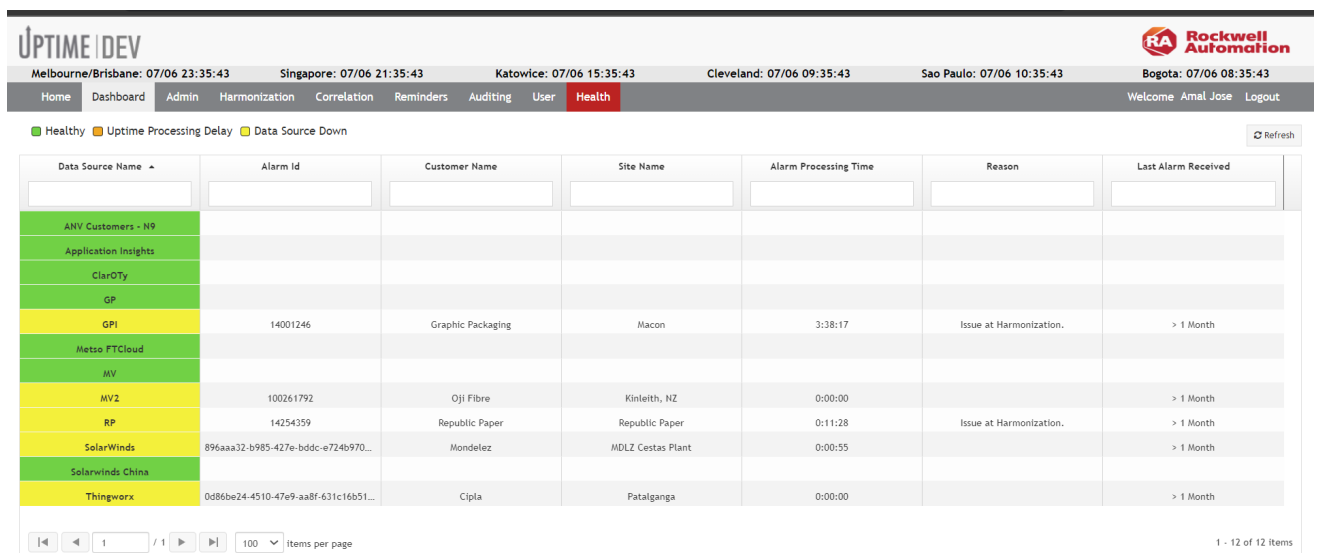


Figure A.10: Health