A project report on

SAPPHIRE - SUPPORT CASE ASSIGNMENT Q-BOT

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering spec. Information Security

by

Tushar Singh (17BCI0156)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

June 2021

DECLARATION

I hereby declare that the thesis entitled “SAPPHIRE - SUPPORT CASE ASSIGNMENT Q-BOT” submitted by me, for the award of the degree of Bachelor of Technology VIT is a record of bonafide work carried out by me under the supervision of Ms. Anshica Sharma.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Bengaluru, India **TUSHAR SINGH**

Date: 20/06/2021 Signature of the Candidate

NUTANIX

@nutanix

3 February 2021

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Tushar Singh**, **Systems Reliability Engineer - Intern** is pursuing the internship at Nutanix Technologies India Private Limited from **January 04, 2021** to **June 30, 2021.**

The Company's business days is generally **Monday** to **Friday** from **9:00am** to **5:30pm**, subject to a lunch break of **30 minutes.**

Reference ID : *5295b1849e2d01fa63d70a50400118a1*

For **Nutanix Technologies India Private Limited**

Danish George
Manager, HR Services
danish.george@nutanix.com

ACCEPTED AND AGREED: I confirm I am Danish George and I intend to electronically sign this document. I intend that my electronic signature shall be binding upon me in the same way as my handwritten signature.

ABSTRACT

Sapphire Bot is a management tool that is used in Nutanix worldwide to suggest cases to the System Reliability Engineer's based on their skills and other metrics such as shift timings, case priority, seniority of the SRE handling the case and calculated backlogs. The problem of manual case assignment is solved by this bot, thereby removing all inconsistencies. Before this bot, case assignments were random and manual, which often led to variation and human resource wastage.

The database is integrated with the SFDC (Salesforce) API and 8x8 API. MongoDB is employed within the backend with several collections that contain the small print of the agents. The agents' details are fetched from the 8x8. The Salesforce API is employed to bring incoming case details for the bot code. The bot is constructed on a NodeJS framework which suggests cases to the agents based on their skill sets and few other parameters like case priority, case type and tags, backlogs, etc. The agile software development methodology is used to deploy the bot by using Heroku pipelines. Timber.IO is used for logging information, and a host of GUI is used for the bot and the connected dashboards.

As a result of successful deployment of the bot, the cases are now assigned more efficiently, taking into consideration several important factors like backlogs, shifts and availability in addition to case priority, skills required and seniority of the engineer handling the case. This is not only very effective but also highly profitable to the company in terms of human resource and customer satisfaction. Furthermore, it also provides a centralized platform which is essential for management.

Further work on the bot would include running the bot on weekends, health monitoring for every individual component of the bot on regular intervals, the addition of a kill switch to enable graceful shutdown of the bot and extensive integration of NLP to classify cases into more granular buckets further.

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to Dr G. Viswanathan, Chancellor, VIT University for creating an environment conducive for learning and growth academically as well as in extracurricular activities that enable every student to reach greater heights.

I would like to thank Dr Prabu S., HOD, School of Computer Science and Engineering (SCOPE, IS), Vellore Institute of Technology, for always encouragement and support, without which this project and the last four years would not have been possible.

I would also like to thank all the faculty and laboratory staff of the Department of Computer Science and Engineering.

I would like to express my gratitude to Abhinav Bhargava, Director, WW Support, Nutanix for giving me an internship opportunity at Nutanix Inc. Additionally, I would like to thank my managers Subhash Shankar, and Abhishek Mukherjee, mentor Anish Singh Walia and my project lead Srimoyee Mukhopadhyay for guiding me throughout the internship.

Lastly, I would like to thank the entire SRE team at Nutanix Inc. for welcoming me into the organization and helping me tirelessly during every point in the internship.

It is indeed a pleasure to thank my family and friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Bengaluru, India

Date: 20/06/2021 **Tushar Singh**

CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

1) HCI : Hyper-Converged Infrastructure
2) CLI: Command Line Interface
3) SSH: Secure Shell
4) KPI: Key Performance Indicator
5) PR: Percentile Rank
6) IQR: Inter Quartile Range

Chapter 1

Introduction

1.1 About the Company

Fig 1.1: Company Logo

Nutanix Inc. is an industry leader in Hyperconverged Infrastructure (HCI). It is essentially a cloud computing organisation that provides several cloud-based services like cloud monitoring, Desktop as a service, Disaster Recovery as a service, mainly based on software-defined storage (SDN). Founded by Dheeraj Pandey, Mohit Aron and Ajeet Singh in 2009, Nutanix integrates storage and compute, thus providing a powerful solution for small, medium and large scale operations.

Sophisticated machine learning technology has now enabled IT teams to do advanced tasks in a one-click framework. This now enables such consumers to focus on other initiatives that are critical to their business as the management of datacentres is now simplified into a single software solution.

Application mobility is one of the central areas of focus that strives to unify the standard three-tier architecture to a single unit that enables Nutanix to be a global leader.

Brian Stevens was appointed into the board of directors of Nutanix recently. The investment raised $ 312 million from venture capital firms in successive rounds. By 2013, Nutanix came to be known as a “unicorn start-up” due to its valuation at $1 billion. Series E funding led to the company being valued at $2 billion by 2014. An Initial Public Offering (IPO) filed for, by Nutanix in 2015.

1.2 Products

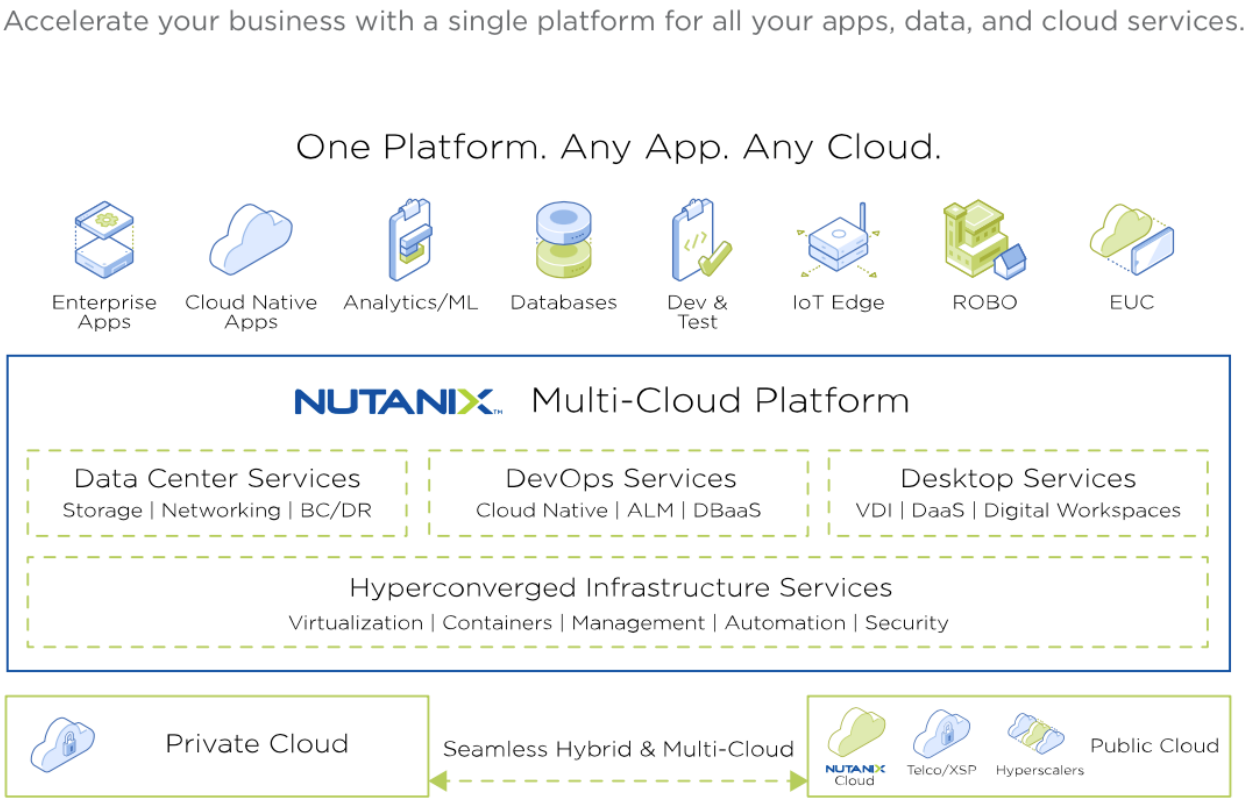


Fig 1.2.1 Nutanix Multi cloud Platform

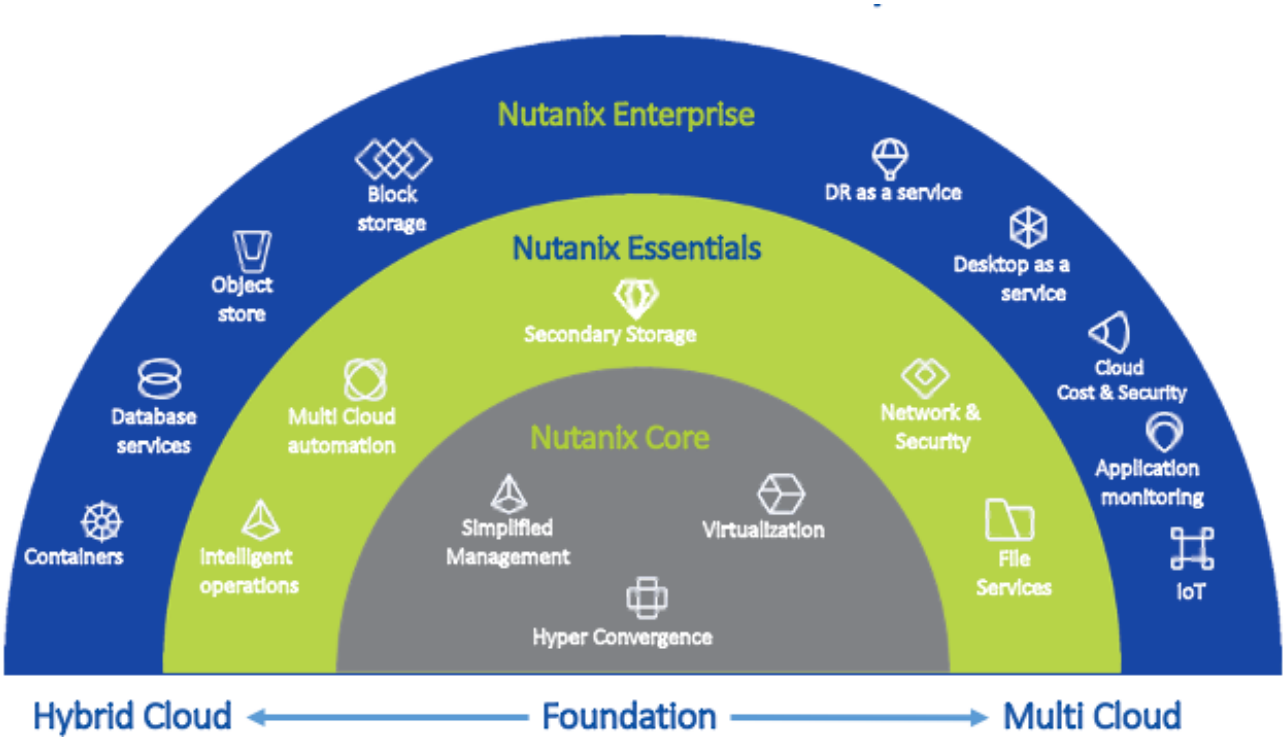


Fig 1.2.2 Nutanix Services

Fig 1.2.3 Nutanix Hardware Specs

Chapter 2

Background

This chapter comprises the overall overview of the project. The existing system for the reconciliation process is discussed, and the requirements for an automated system are discussed. It also talks in detail about the Cause, Motive, Problem Statement, Objectives and end goals of the project.

2.1 Business Problem

With the rapid expansion of the company in recent times, there has been an increase in the number of employees and the cases to be tackled, and hence a need for a more efficient and automated management tool for scheduling and tracking work progress of the employees. Sapphire is an immense help to the System Reliability Engineers at Nutanix. The cases are taken from Salesforce and then scheduled and displayed by the bot on Slack with essential details such as priority and tags simply and lucidly along with buttons to accept and reject the case. With an easy to use interface and simplified data flow, the proposed solution will reduce the hassle and scope for errors.

With the rapid expansion of the company in recent times, there has been an increase in the number of employees and the cases to be tackled, and hence a need for a more efficient and automated management tool for scheduling and tracking work progress of the employees. Sapphire is an immense help to the System Reliability Engineers at Nutanix. The cases are taken from Salesforce and then scheduled and displayed by the bot on Slack with essential details such as priority and tags simply and lucidly along with buttons to accept and reject the case. With an easy to use interface and simplified data flow, the proposed solution will reduce the hassle and scope for errors.

Sapphire, the Auto Case Assignment Bot, is a management tool used by the Bangalore- based Nutanix Worldwide Support Team to recommend cases to System Reliability Engineers based on a variety of metrics such as their expertise, availability on a specific day and importance of the matter posed by the customer. Until Sapphire was introduced, cases were given manually to

System Reliability Engineers, which would result in a great deal of variation in the number of cases chosen by each person. This was one of the significant problems that this tool resolved once it was brought into production. It also saved a lot of workforces and made the managers work much easier as it provided a centralised platform to keep track of the number of cases each System Reliability Engineer (SRE) handles in a day.

Nutanix Inc. is an industry leader in Hyperconverged Infrastructure (HCI). It is essentially a cloud computing organisation that provides several cloud-based services like cloud monitoring, Desktop as a service, Disaster Recovery as a service, mainly based on software-defined storage (SDN). Founded by Dheeraj Pandey, Mohit Aron and Ajeet Singh in 2009, Nutanix integrates storage and compute, thus providing a powerful solution for small, medium and large scale operations.

Sophisticated machine learning technology has now enabled IT teams to do advanced tasks in a one-click framework. This now enables such consumers to focus on other initiatives that are critical to their business as the management of datacentres is now simplified into a single software solution.

Application mobility is one of the central areas of focus that strives to unify the standard three-tier architecture to a single unit that enables Nutanix to be a global leader.

Brian Stevens was appointed into the board of directors of Nutanix recently. The investment raised $ 312 million from venture capital firms in successive rounds. By 2013, Nutanix came to be known as a “unicorn start-up” due to its valuation at $1 billion. Series E funding led to the company being valued at $2 billion by 2014. An Initial Public Offering (IPO) filed for, by Nutanix in 2015.

2.2 Objective

The following are the objectives of the project:

- To develop a bot in order to automate case assignment and reduce the workload of duty managers.
- To determine which of the cases, the System Reliability Engineer has been assigned to work on and whether the SRE presented the underlying reaction to the client before the administration level understanding could be abused. The SRE who has been allocated to that particular case must be explicitly sent a message on Slack telling him/her that the case's SLA will be violated.
- To assess which of the scheduled cases could exceed their planned SLA time. In this case, too, the duty manager must be notified explicitly that the scheduled SLA period is approaching.
- To calculate total cases that a System Reliability Engineer has accepted or refused, along with the case ID. When this information has been gathered, distribute it in a dashboard which consequently refreshes and gives the capacity to see the information for any past dates and further to channel the information. Include an option for easy review to export the data to CSV format.
- To determine the severity of the case based on customer input and assign SRE of higher seniority.
- To determine whether if a case is due to an issue in the core infrastructure and tag it accordingly, to be assigned to SREs with suitable expertise.
- To provide a quarterly summary on the dashboard and daily review to the DM.

Chapter 3

Methodology

3.1 Requirements Gathering

3.1.1 Specific Requirements

This area covers all product necessities with enough subtleties, which can be utilized by developers to construct a framework to fulfil prerequisites. The item point of view and client qualities don't express the genuine necessities required that the framework but instead state how the item should function as for client comfort. The essential particulars are genuine subtleties which can be chosen by the client and programming supplier. The last execution is required to follow the entirety of the particulars laid out here.

3.1.2 Functional Requirements

The following lists the functional requirements of the system:

- The system must automatically suggest cases to the SREs but should be limited to 4 per day.
- The system should provide equal distribution of cases amongst all the employees through a randomisation function, and the bot should consider the case count taken from SFDC as well.
- The bot should consider the priority of the cases and the status of the employees before assigning the cases.
- The bot case assignment code should run every 15 minutes and should run only during the queue time for that day.
- The system runs on a Heroku server at the backend. The server must be up and running during the queue time. The dashboard must be compatible with all the browsers.
- MongoDB must be running, and there should be a connection to the database with appropriate authentication.

- The NodeJS GET and POST requests should be running and listening for requests on the right port.
- The messaging service Slack must be running.
- All Environment settings must be configured correctly.

3.1.2 Non-Functional Requirements

- Efficiency: The system must be performing at the highest possible efficiency at any given time.
- Availability: The system must be highly available and running at any given time.
- Security: There must be no threats to the safety of the system.
- Uniformity: The system should be such that it is not incompatible with standard execution systems.
- Speed: Responsiveness should be high at all times and minimum or no delay at any given time.
- Accuracy: The system must perform all operations without error and ambiguity.

3.1.3 Hardware Requirements

- Server to run the auto-case assignment code
- 64-bit architecture
- Four or more CPUs/cores
- 4 GB RAM
- Minimum 25 GB of disk space

3.1.4 Software Requirements

- Timber.io
- NodeJS
- Heroku
- 8x8 API
- MongoDB
- Google Sheets API

3.2 Work Schedule

1. January 2021
    - Company orientation and onboarding activities
    - Project assignment
2. February 2021
    - Understanding existing code and learning relevant technologies
    - Deciding on enhancements to be made
    - P0 task assignment
3. March 2021
    - P0 task submission
    - Unit testing of P0 task and debugging errors
    - P1 task assignment
4. April 2021
    - P1 task submission
    - Modifications made according to requirements
    - Unit testing P1 task
    - P2 task assignment
5. May 2021
    - P2 task submission
    - Unit testing P2 task 4

6. June 2021

    6. Integration testing of all components
    7. Fixing issues and other maintenance activities
    8. Submission of final project report for six months project

3.3 Detailed Methodology

3.3.1 System Design :

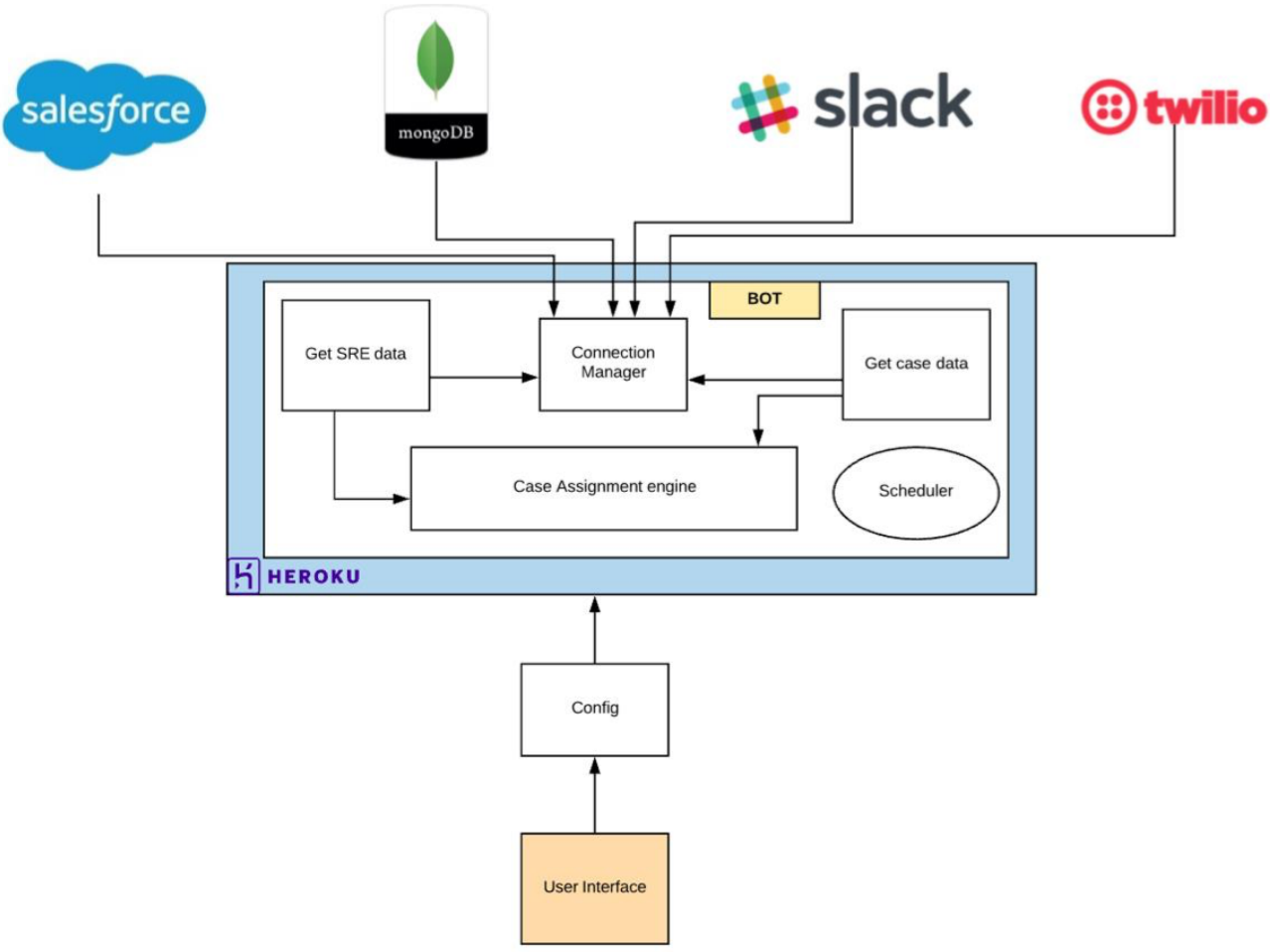


Fig 3.1 System Design

Sapphire Q Bot, as it is commonly known as, is implemented using NodeJS. The bot depends on many other functionalities and features that allow it to operate smoothly.

The Google Sheets are refreshed consistently with the accessibility of each SRE, alongside move data and their individual aptitudes. The Google Sheets API is utilised to recover and store this information in MongoDB.

8X8 is an integration software that organisations use to evaluate the status of the worker and the call status at that particular second. 8X8 is utilised to catch the SRE's call status data during the day on the grounds that a high need case won't be doled out to the SRE on the off chance that he is as of now on a call. Thus, 8X8 APIs are utilised to get the above subtleties just as the points of interest of what number of SRE's are available in the line when Sapphire is going to propose the cases.

All these data are stored in multiple collections in MongoDB along with other information such as who is the day's Duty Manager(DM), the slack IDs of each and every SRE, the number of cases an SRE picked in a day etc.

The database that is being utilised is coordinated with the SFDC API and 8x8 API. The bot is based on a NodeJS system which proposes cases to the workers on their ranges of abilities and not many different boundaries. The product improvement philosophy being embraced is here is Agile. Heroku will be utilised to have the creation bot just as the organising bot while Timber.IO would be utilised for logging purposes for the duration of the time the bot spends in the line. Different graphical UI entrances would be made at various cases for straightforward entry and to acquire and show information in a start to finish way.

Sapphire's working logic takes this knowledge into account through REST API before recommending the cases lying unassigned to an SRE in the queue. A case recommendation is made via a group chat in SLACK that involves all of the SRE's. The slack API's are incorporated into the code for this.

The bot is hosted and staged on a server via Heroku, and it can be accessed via Timber.IO to all logs created while it operates.

i. **Implementation Requirements**

The execution stage is noteworthy stages in the improvement of the task, as it gives the last answer for the issues. During this procedure, low-level plans are changed over into language-explicit projects to meet the details set out in the determination of programming particulars. This stage incorporates actual usage of thoughts which have been characterised in the period of

examination and plan. The approach and strategies utilised for programming usage must advance reusability, simplicity of support and ought to be notable.

ii. **Programming Language Selection**

The programming language picked to actualise the undertaking are NodeJS. NodeJS is one of the most helpful dialects in the current age and time with broad inclusion in help devices and structures for a quick turn of events. A portion of the advantages that NodeJS gives, which were critical to picking the equivalent is:

- Simple, easy and highly readable program syntax
- Extensive support for Web Applications
- Platform independent and portability
- Myriad set of libraries for tools like stream handling, exception handling
- Multi-threaded approach and annotation processing

iii. **Platform Selection**

The bot is designed to be independent of the hosting platform. It works equivalently on Mac OS, Linux and Windows OS though the development phase was covered in Mac OS. It was further tested on Linux and Windows to ensure that it is not incompatible with such systems. The dashboard, on the other hand, is engineered to be portable and hence does not run into any compatibility issues.

iv. **Code Conventions**

This segment tends to the coding rules that were followed all through the whole venture. It contains the product applications required to finish the venture. Appropriate coding norms ought to be followed since coding of massive undertakings in a predictable style is needed. It makes each part of the code more obvious absent a lot of difficulties. Code shows are significant in light of the fact that they improve coherence in programming, empowering the software engineers to obviously understand code.

v. **Naming Conventions**

Naming conventions help programs in a comprehensible way which makes reading easier. The

names given to packages, texts, graphs, and classes must be concise and straightforward in order to be readily understood as their content. The project uses both NodeJS and Bootstrap, and the naming convention followed in the two are slightly divergent from each other.

The standards followed throughout this project are as follows:

- Classes: Names of classes are typically nouns. The first letter of every word is capital according to upper camel casing method.
  Example: CaseAssignment.
- Methods: Methods musty be doing words or verbs for better quality. Here too, the method name starts with a non- capital letter and the rest of the name adheres to upper camel casing.

  Example: getValue( ).

vi. **Comments**

Comments are a vital part of any programming conventions as it enhances the comprehensibility of the established code. In the project files, owing to the IDE and GUI based environments, grey is the default colour for showing commented texts, so they are easy to identify. This project contains a variety of comments throughout the entire code to enable easy understanding and handover of the project.
Comments help in better understanding of code. In NodeJS, if the comment spans for only one line then its starts with '//' and if the comment spans more than one line then its starts and ends with '/*' and '*/' respectively. These comments increase the readability of the code and make it more lucid.

3.3.2 Assumptions Made

1. The bot is set up on Heroku with the required environment variables filled in.
2. The bot is connected to a functional MongoDB instance with proper permissions

   for reading/writing data.

3. All required data regarding cases is available on Salesforce via the Salesforce API.

4. Employee status is available via the 8x8 API.
5. All notifications are to be delivered to the employee via Slack.

3.2.3 Design and Modelling

The high-Level design represents a broad overview of the system being developed. It shows the different modules at a glance, giving an overall understanding of the functioning. System architecture, Data flow diagrams, Architecture diagrams and Entity Relationship diagrams all come under high-level design and are described in the following subsections.

i. **Architecture Diagram**

This is a pictorial model used for depicting the interaction of the user with the system. Fig 3.2 helps in portraying the functionalities of the system, which are also called as use cases and how the user could interact with the system using those functionalities provided by the system. The following architecture diagram highlights the main components of the bot and how they interact with one another in order to ensure the functioning of the bot.

When the bot application is run from Heroku interface or Heroku CLI, it first invokes the bot code written in NodeJS. Once this code starts running, it first invokes Salesforce in order to get a list of incoming cases, also containing case details such as SLA, priority, keywords, etc. After this, the code invokes 8x8 API to get a list of available agents and Google Sheets API to verify the same. Finally, the bot matches agents with cases and the respective agents get messages on their Slack messaging applications where they can choose to accept or reject the case. In case of an error in running the bot, the application log messages collected by Timber.io can be viewed.

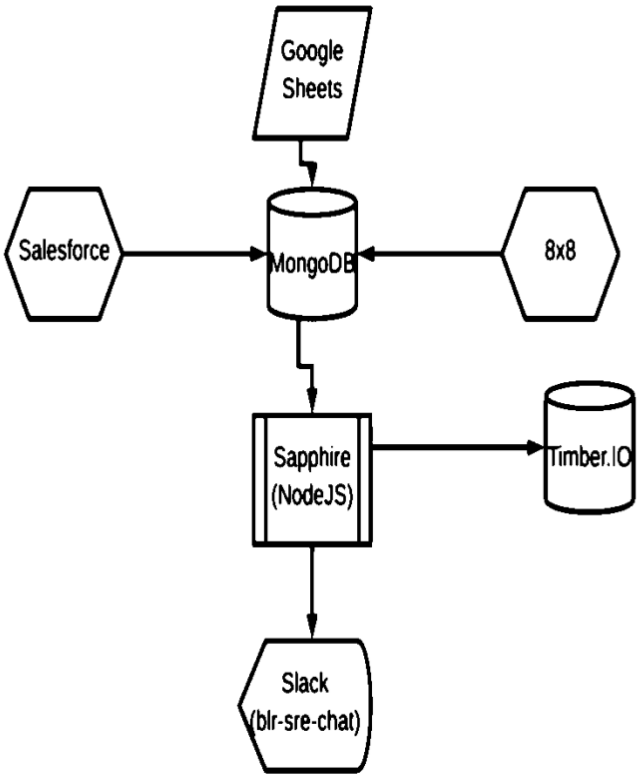


Fig 3.2 Architecture Diagram of all working components of the bot ii. **Data Flow Diagram**

An information stream outline (DFD) portrays the information "stream" graphically through a data framework. Information Flow models clarify how the information moves through a handling stage arrangement. DFD comprises of four components including strategy, information stream, outside element, and information store. Utilising information stream graph, clients can without much of a stretch envision the activities inside the program, what should be possible utilising the program and framework usage. DFDs give end clients a theoretical comprehension of the information provided to the framework as information, the effect the data would, in the long run, have in the general framework.

Level 0

Level 0 DFD gives an overview of the operation of the bot. It mainly highlights how SREs interact with the bot.

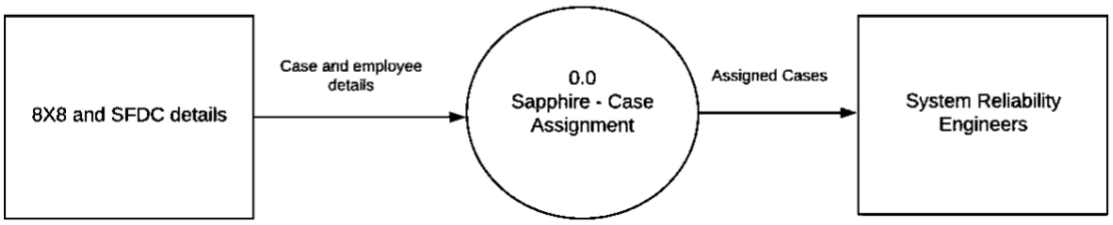


Fig 3.3 Level 0 Data Flow Diagram

Level 1

The Level 1 DFD goes into more granularity than the Level 0 DFD. It specifies the four main modules involved in the system.

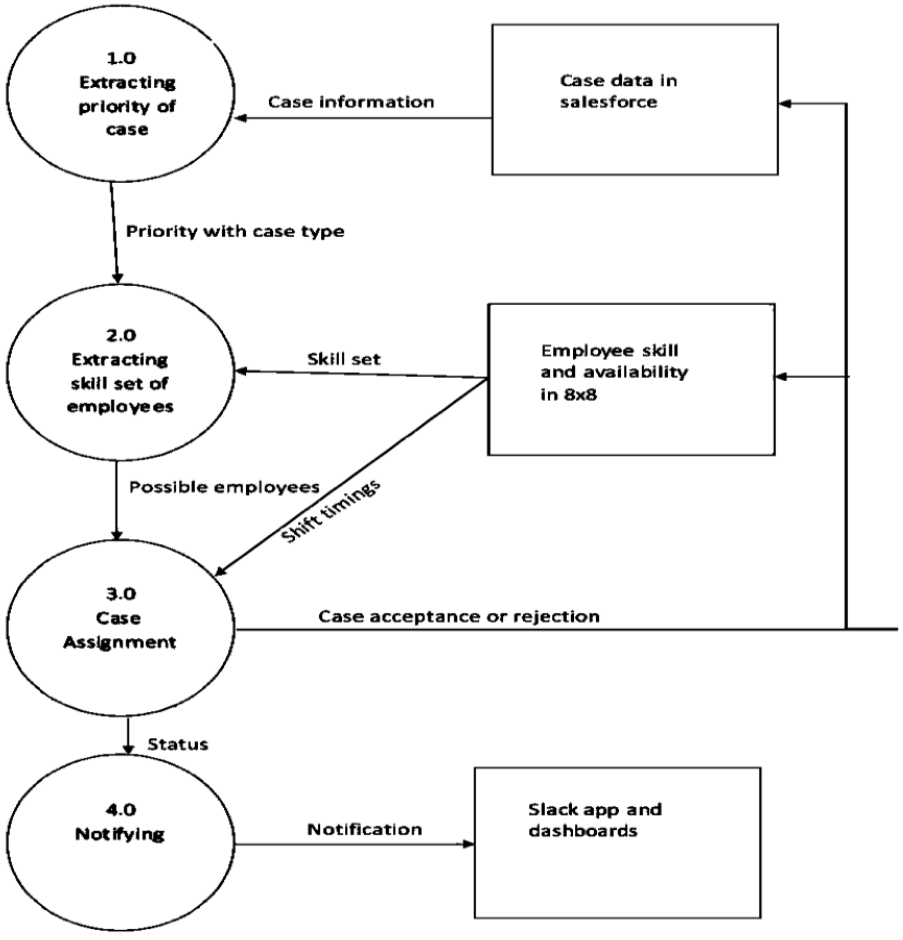


Fig 3.4 Level 1 Data Flow Diagram

Level 2

The processes in level 1 are expanded here. The SFDC API is queried using GET requests which give us the details of the case such as case numbers currently present in the queue. The crucial details of the case, such as priority and owner, are then considered by the case assignment function. The employee details, such as skill set and status on 8x8, are obtained by querying the MongoDB and 8x8 API. These details, along with the case details, are used by the bot to assign the cases to the employees on Slack by obtaining the Slack ID and sending a notification on Slack.

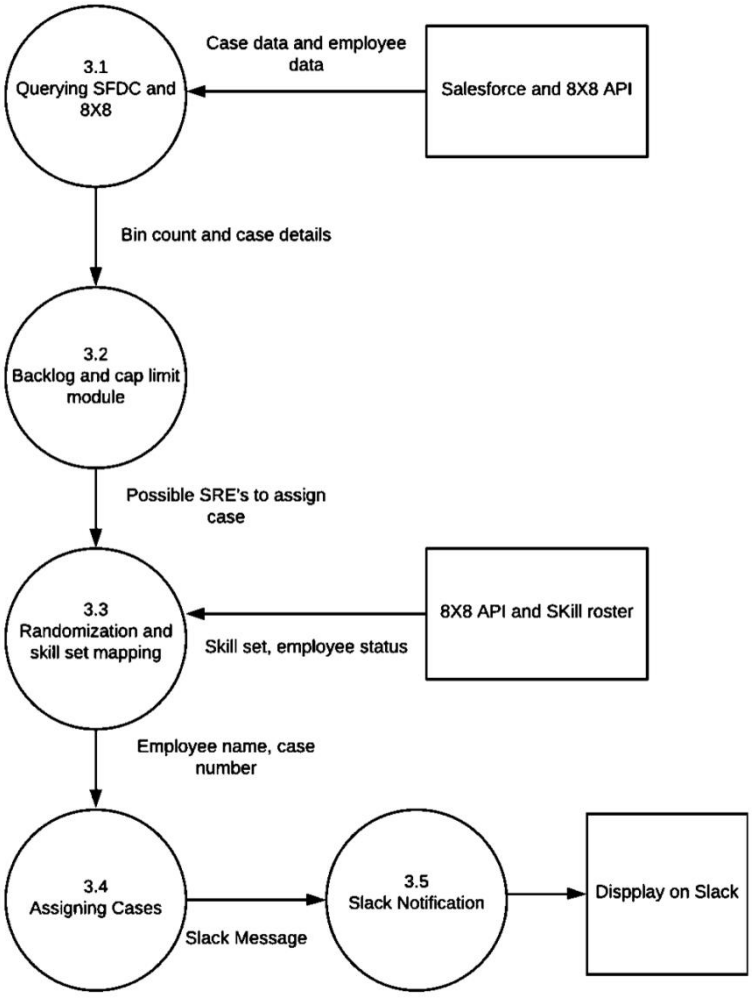
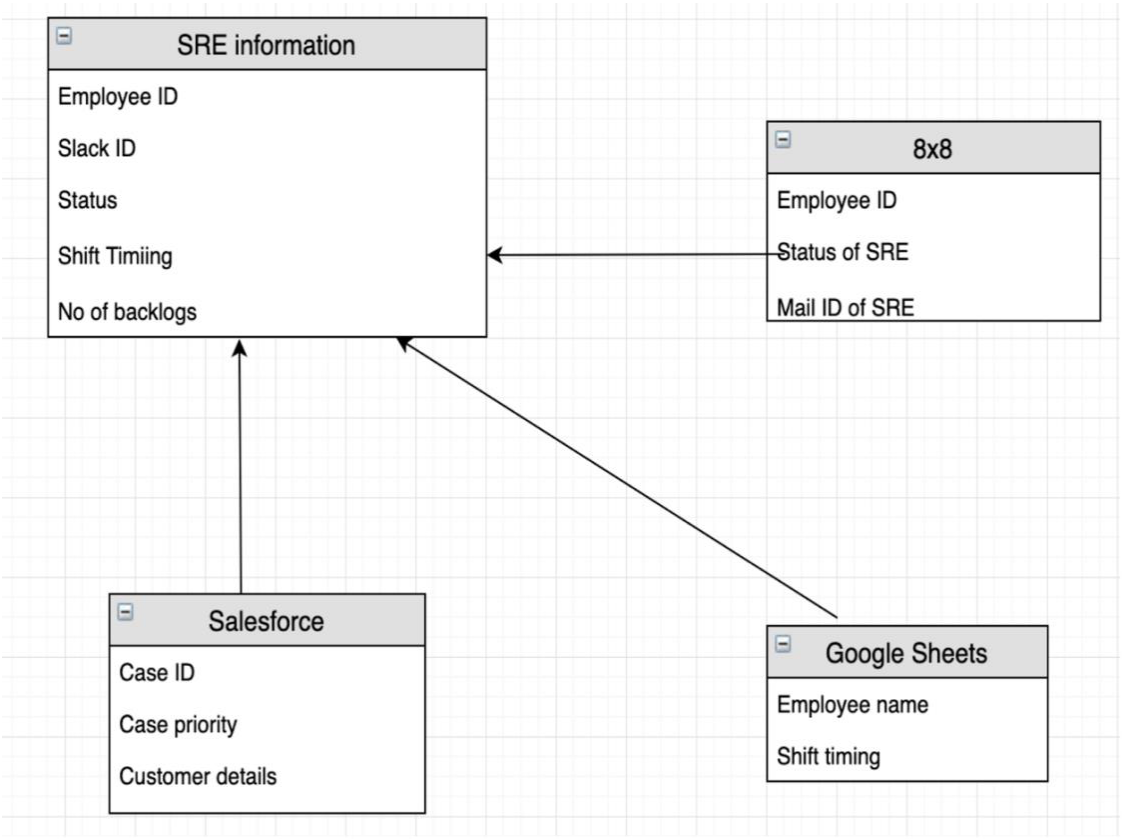


Fig 3.5 Level 2 Data Flow Diagram

Entity Relationship Diagram

An entity-relationship model likewise called an element relationship (ER) outline, is a graphical portrayal of elements and their connections to one another, typically utilised in figuring as to the association of information inside databases or data frameworks.



3.2.4 Module Specifications

Fig 3.6 Entity Relationship Diagram

This sub-section describes some of the modules implemented to enhance the functionality of Sapphire Q bot. These modules resolve to incorporate several features such as including case summaries and quarterly summaries to ensure monitoring of the bot's accuracy at regular intervals and add enhancements or corrective features in the event of malfunctioning. This sub-section also includes two other modules which focus on assigning certain types of cases on priority to a certain level of seniority of engineers to ensure that these cases are handled by the

highest level of expertise available. This is beneficial for the company and the customers as certain cases and customers, based on the Service Level Agreement (SLA) to get a certain level of prioritisation so as to enable immediate attention of engineers to any failure of components of the Nutanix Hyperconverged Infrastructure (Nutanix HCI). These modules form the basis of testing the bot and any failure alerts the Sapphire Q Bot development team not only to the fact that an error exists, causing the bot to malfunction, but also which specific module or component is causing the error, allowing for a quicker resolution of the error with minimum downtime and inconvenience.

i. **Reporting Case summaries in Slack**

Daily summaries must be sent to the duty manager so they can evaluate the performance of the bot, and check if it has made any errors in its case scheduling algorithm, as well as if an SRE is dropping too many cases. The duty manager can then manually complete any failed case assignments and ask the Sapphire bot development team to fix the issue.

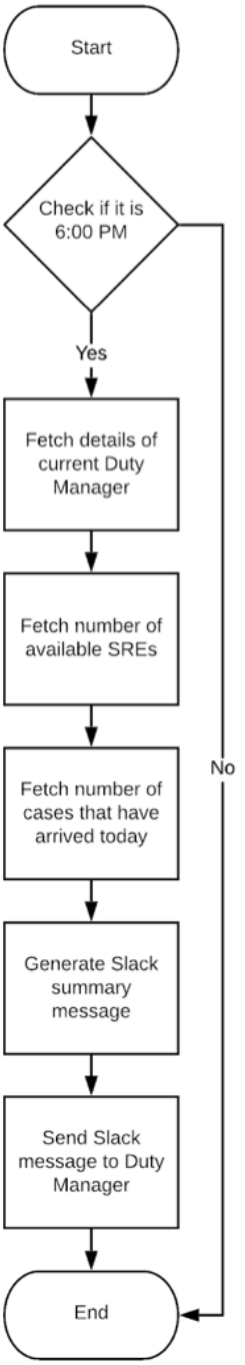


Fig 3.7 Case Summary Flowchart

ii. **Assignment of Mission Critical Cases**

This module was created to assign mission critical cases to SRE-3. Certain classes of errors can cause the entire production to go down. These errors need to be resolved at the earliest to prevent more loss and damage to the client. Such cases are known as mission critical cases and have to be assigned to SREs who have higher support level. In case of SRE-3 are not available, the bot assigns these cases to any of the available SREs using the best-effort algorithm.

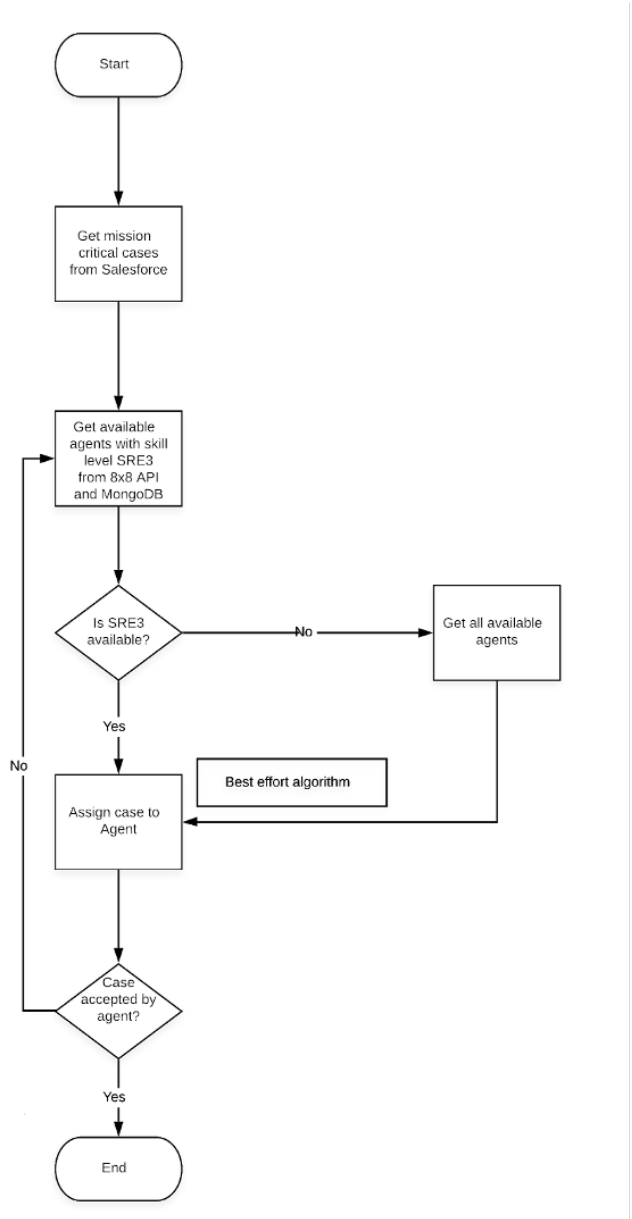


Fig 3.8 Mission Critical Cases Flowchart

iii. **Assignment of DSE Cases**

Nutanix prides itself in its top-class customer support. Often enough, there are extremely important servers running on Nutanix technology. If there is any issue in those servers, it can cost the company money for every second of delay in fixing the issue. For that reason, a lot of high-priority companies are made DSE clients, which means that there is a fixed SRE whose duty it is to ensure that they have no issues in their servers. Thus, there is a need to implement custom logic in the bot to assign DSE cases to their specific SREs and handle the cases where that SRE is unavailable.

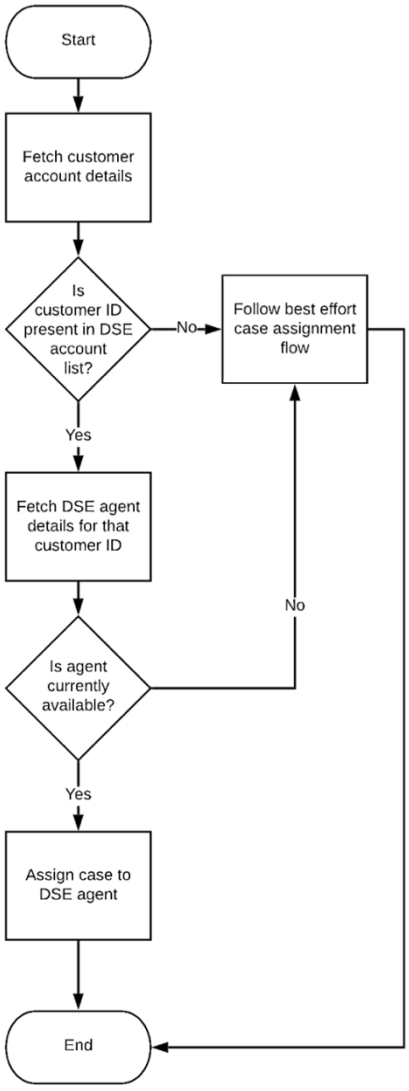


Fig 3.9 DSE Cases Flowchart

iv. **Quarterly Summary of CDP and Infra**

Core Data Path (CDP) and Infrastructure Services (Infra) are two broad categories to classify certain types of cases based on the case tag fetched from Salesforce API. This module is used to display the summary of such case assignments per quarter and the calculated percentage efficiency of the bot in assigning these accurately.

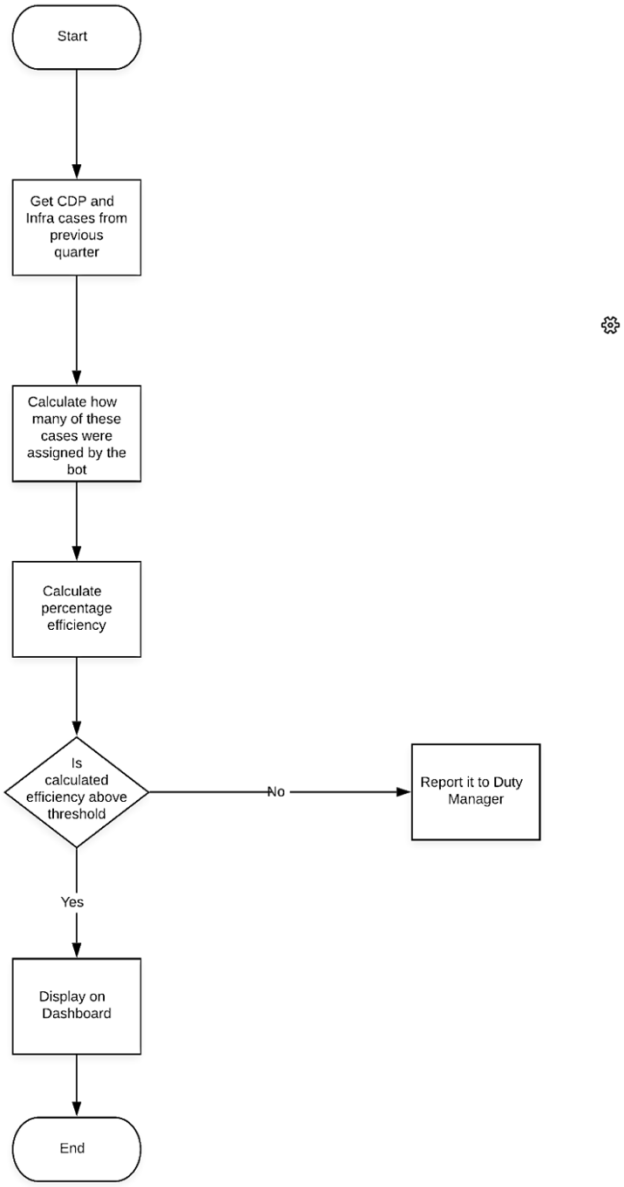


Fig 3.10 Quarterly Summary Flowchart

3.2.5 Justification for Modules

i. **Reporting Case Summaries in Slack**

The main goal of the Sapphire bot was to conserve valuable human capital and save time. Be that as it may, it plays out a critical capacity, and any blunder in the bot's working could undoubtedly cost the organisation a great deal of money. Hence, there is a need to continually screen and assess that the bot is functioning as intended, and particularly that no cases are as a rule left unassigned by the bots booking calculation. Thus, we have assembled a summary module that sends every day rundowns to the DM. In the event that there is any warning in the outline, the DM will promptly realise that something isn't right and assume the role of manually assigning the cases in place of the bot.

ii. **Assignment of Mission Critical Cases**

Certain types of failures and errors can cause the entire production to stop. Such cases are mission critical cases and need to be handled with a higher level of expertise as even a slight error or prolonged time is taken to resolve the issue can be very costly for the client and also for Nutanix. Hence, it is imperative that such cases are handled by senior level SREs or SRE-3. Therefore, we have created this module to assign such cases by order of seniority so as to enable the bot to assign such cases to SREs with level 3 support.

iii. **Assignment of DSE Cases**

The DSE cases module enables Nutanix to tend to the issues faced by specific clients who are ranked as a top priority. These clients are called DSE clients, and their issues need to be resolved as quickly as possible. Hence, a single dedicated engineer is assigned such a case. The bot logic thus needs a way to classify incoming cases as DSE or any other, and this particular module is designed for the same. This module basically picks the DSE cases from the incoming cases and then assigns a single SRE to such a case.

iv. **Quarterly Summary of CDP and Infra**

Nutanix provides support for a large majority of cases that come under the broad categories Core Data Path (CDP) and Infrastructure Services (Infra). The bot assigns these cases to SREs by matching their skillset with the skillset specified for the cases. The quarterly summary module basically gets the total number of CDP and Infra cases from the Salesforce API and subset of these cases assigned by the bot accurately. It calculates a percentage efficiency which is displayed in the dashboard along with the summary of

cases assigned per quarter. This module is necessary from the Duty Manager's perspective to determine the efficiency and accuracy of the bot.

3.3 Tools Used

1. Salesforce API

Salesforce is the essential endeavour offering inside the Salesforce platform. It furnishes associations with an interface for the board of case and task, and a framework is sufficiently wise to naturally directing and raising significant occasions without manual intercession. The Salesforce client entryway gives clients the capacity to follow their own cases, incorporates an interpersonal interaction module that empowers the client to join the discussion about their organisation on a person to person communication sites, gives explanatory devices and different administrations including email alert, Google search, and access to clients' privilege and agreements.

2. 8x8 API

8x8 is an organisation which has practical experience in offering types of assistance organisations which are chiefly engaged in the territories of voice, video, versatile and bound together correspondences. They are various sorts of APIs to couple needs with the Virtual Contact Center, which are given by 8X8. These API uses HTTP/HTTPS with help for GET, MODIFY, ADD, DELETE and LIST activities in XML designing. There are various kinds of API gave by 8X8, and each is utilised for inquiry distinctive database.

3. MongoDB

MongoDB is an open-source database management system where it supports various types of data. MongoDB works in NoSQL format and is used in the project to store case as well as the SRE status. MongoDB is made up of collections and documents. As it is NoSQL, it can store any type of data and doesn't require any predefined schemas.

4. NodeJS

NodeJS is a server-side platform engineered on Google Chrome's JavaScript Engine (V8 Engine). Developed by Ryan Dahl, its latest version is 10.36 NodeJS is an open-source, cross-stage runtime environment for creating server-side and systems administration applications. NodeJS applications region unit written in JavaScript, and maybe run at spans the NodeJS runtime on OS X, Microsoft Windows, and Linux.

NodeJS also gives a rich library of various JavaScript modules which unravels the headway of web applications using NodeJS overall.

5. Heroku

Heroku is a holder based cloud Platform as a Service (PaaS). Designers use Heroku to convey, oversee, and scale present-day applications. Its foundation is rich, adaptable, and simple to utilise, offering designers the most straightforward way of getting their apps to deploy on several platforms.

6. Timber.IO

Timber.io is the main tool used for logging information. It is a cloud based platform mainly designed to work with Node, Elixir and Ruby applications. It creates easily decipherable logs and adds context to them. It can automatically sync with open-source packages and structure the data contained in records.

7. Postman

Postman is a well known API customer that makes it simple for designers to develop, offer, test and archive APIs. This is finished by permitting clients to establish and spare straightforward and complex HTTP/s demands, just as read their reactions—the outcome - increasingly proficient and less dreary work.

8. Robo3T

Robo 3T (some time ago Robomongo) is a well known work area graphical UI (GUI) for your MongoDB facilitating arrangements that permits you to collaborate with your information through visual markers rather than a book based interface

3.4 Preliminary Analysis

The requirement for this project stems from the fact that duty managers have a lot of repetitive work with regards to case assignment that could easily be automated.

Usually, the duty manager looks at incoming cases, assigns them to SREs, and sends the SRE a message on Slack. Thus, we need a system that is able to identify incoming cases, find a suitable SRE for it, and send them a message notifying them of their assignment.

A bot capable of performing these actions would require access to Salesforce and 8x8, just like the duty manager would have. It would require hosting, both for the bot and the database. However, the cost of building, maintaining, and hosting such a system to automate case assignment would be far less than the cost of the duty manager's valuable time that gets wasted.

We have elected to go with Heroku for hosting. Heroku is a relatively cheap hosting solution, and it makes hosting a trivial affair since it handles most of the process and hides it behind a user-friendly UI. For database hosting, we are going with mLabs' MongoDB hosting, since it is available as a "Heroku Element" (which is a third-party service that is well-integrated with Heroku). This makes database deployment and connecting to it extremely easy.

3.5 Conclusion

This chapter highlights the central methodology of the project, along with the design and implementation details. It further contains module specifications, tools and preliminary analysis.

Chapter 4

Result Analysis

4.1 Introduction to the Chapter

This chapter discusses and analyses the results obtained by the Sapphire bot. The bot has already been deployed into a production environment and is currently powering case assignments. It has already saved hours of valuable time and works well without any errors.

4.2 Result Analysis

1. Report Case Summary in Slack

The following shows the test case for Case Summary module on which this testing is performed.

Table 4.1 Result Analysis for Case Summary Module

Sl.no. of test case	1
Name	Checking dashboard for cases accepted and rejected
Feature being tested	Testing the case count of accepted and rejected cases
Description	Testing if the dashboard has the live count of cases being accepted and rejected
Sample input	Case counts in MongoDB and the names of the employees
Expected output	Display of cases accepted and rejected on the dashboard
Actual output	Number of cases accepted and rejected per STE on that day
Remarks	Successful function flow

2. Assignment of Mission Critical Cases

The following shows the test case for Mission Critical Cases module on which this testing is performed.

Table 4.2 Result Analysis of Mission Critical Cases Module

Sl.no. of test case	2
Name	Checking the case assignment of mission critical cases to SRE-3.
Feature being tested	Testing the case count of mission critical cases assigned to. SRE-3
Description	Testing if a case tagged as mission critical is being assigned to an SRE-3
Sample input	Case description from Salesforce and the description of the employees
Expected output	Mission critical cases being assigned to SRE-3
Actual output	Mission critical cases being assigned to SRE-3
Remarks	Successful function flow

3. Assignment of DSE Cases

The following shows the test case for DSE Cases module on which this testing is performed.

Table 4.3 Result Analysis of DSE Cases Module

Sl.no. of test case	3
Name	Checking DSE case assignments
Feature being tested	DSE cases are being given to only the specific SREs assigned for them
Description	Testing if DSE cases are assigned to the associated SREs
Sample input	A new case in Salesforce with a dummy customer ID who has been set as a DSE customer
Expected output	The DSE case is assigned and sent to the associated SRE
Actual output	The case was assigned to the correct SRE
Remarks	Successful function flow

4. Quarterly Summary of CDP and Infra

The following shows the test case for the Quarterly Summary of CDP and Infra module on which this testing is performed.

Table 4.4 Result Analysis of Quarterly Summary Module

Sl.no. of test case	4
Name	Checking dashboard for cases tagged as CDP and INFRA
Feature being tested	Testing the case count of CDP and INFRA cases
Description	Testing if the dashboard has the quarterly summary for tagged cases
Sample input	Tagged Case counts in MongoDB and the closing tag assigned by the SREs
Expected output	Display of accuracy of the cases correctly tagged
Actual output	Number of cases correctly tagged as CDP and INFRA and the accuracy
Remarks	Successful function flow

4.3 Significance of the Results Obtained

The significance of the results obtained are as follows:

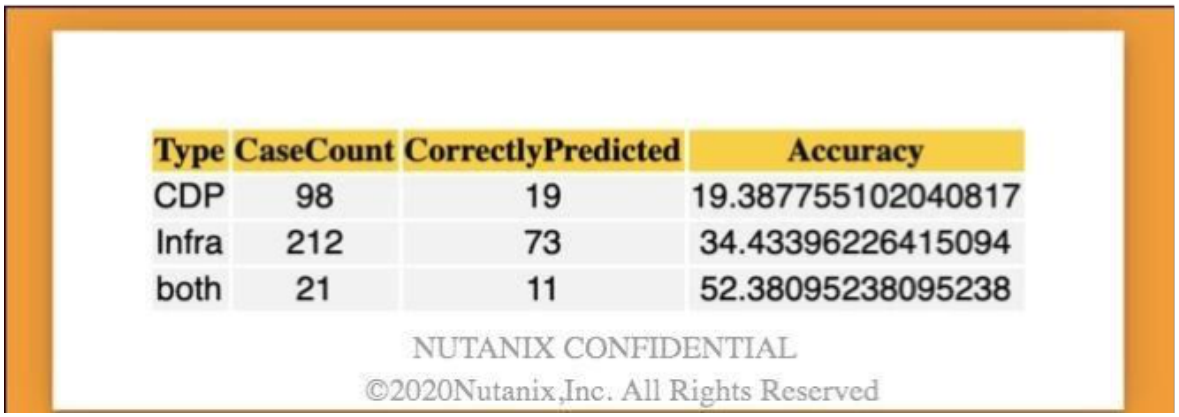
1. Assigning mission critical cases by order of seniority ensures that these cases are handled with the attention and level of expertise that they require.
2. Case summary and report to the duty manager verifies the proper functioning of the bot.
3. The efficiency of the bot is ensured by the quarterly summary of assigned cases.
4. Assignment of DSE cases to fixed personnel as opposed to handing off ensures that high priority customers are satisfied.

4.4 Outputs

1. Dashboard Results

Fig 4.1 Case Summary and Quarterly Summary Results

2. Tagged Cases Results

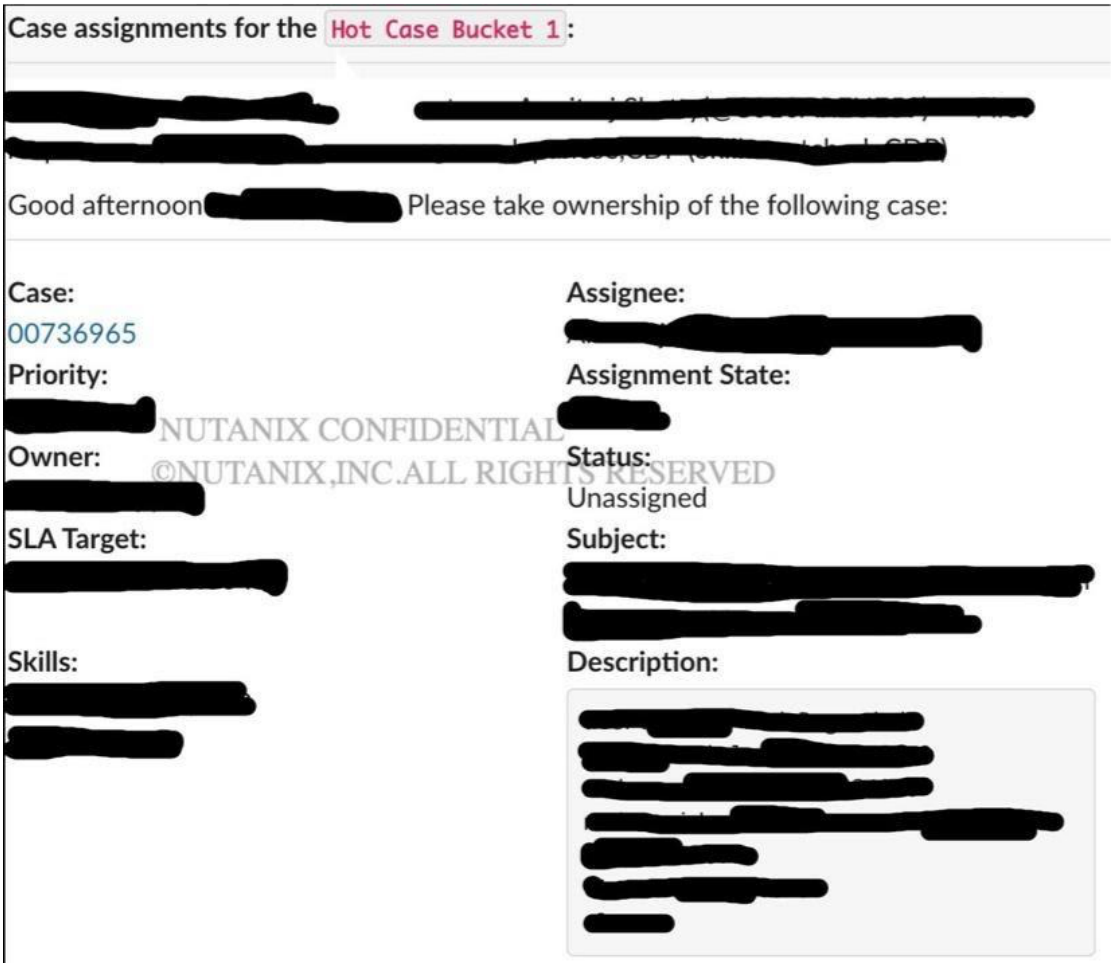


Type	CaseCount	CorrectlyPredicted	Accuracy
CDP	98	19	19.387755102040817
Infra	212	73	34.43396226415094
both	21	11	52.38095238095238

Fig 4.2 Accuracy of cases tagged as CDP and Infra by the bot

3. Case Assignment Results

Fig 4.3 Case Assignment for DSE/ Mission Critical Cases



4.5 Conclusion

From the above result analysis, it is concluded that all the target specifications have been met and the following features have been deployed successfully:

1. All the requirements for the Sapphire bot have been met
2. Case assignments in Slack work reliably
3. The duty manager gets a daily report of the cases that have been assigned, so they know that the bot is functioning as expected
4. DSE cases are assigned to their specific SREs
5. Mission critical cases are prioritised and assigned to more senior SREs
6. Quarterly summaries of cases allow us to verify that the bot is working efficiently

Chapter 5

Conclusion and Learnings

5.1 Introduction to the Chapter

This chapter discusses the relevant conclusions drawn from the development, testing and deployment of the Sapphire Qbot for Nutanix Inc. It also further discusses some of the essential points pertaining to the future scope of the bot.

5.2 Conclusion

With the quick development of the organisation, there has been an expansion in the quantity of representatives and the cases to be handled and subsequently a requirement for a progressively proficient and robotised the board instrument for planning and following work progress of the representatives. With a simple to utilise interface and rearranged information stream, the proposed arrangement will diminish the problem and extension for blunders.

The at present planned framework mechanises the way toward allocating cases to the workers, along these lines, diminishing human reliance. The notification generating framework - Slack reacts in an ideal and time consistent way. The case proposal model empowers the SRE to dismiss a case on the off chance that the individual needs to do as such. The bot has more functionalities, for example, a continuous dashboard which contains data with respect to the cases acknowledged or dismissed. The presently executed framework helps over a hundred representatives for every day.

5.3 Future Scope

i. A kill switch will be introduced to ensure graceful shutdown of the bot.
ii. Feature health monitoring has to be introduced to ensure that all the modules of the bot are working properly.
iii. Enhance the security of the dashboard by providing Okta login.
iv. Hot accounts cases should be assigned only to SRE-3 / preferred SREs.

v. When a case is generated, use natural language processing techniques to figure out the skills required to solve it, and accordingly assign an SRE. Currently, the bot obtains a list of necessary skills by looking for specific keywords in the case text.

vi. In the present scenario where work from home is the new normal, integration of video conferencing apps like Zoom and Cisco Webex would help with better communication and interaction with the customers.

REFERENCES

[1] Cornelia Győrödi, Robert Győrödi , George Pecherle , Andrada Olah," comparative study: MongoDB vs. MySQL", 2015 13th International Conference on Engineering of Modern Electric Systems (EMES)

[2] Li Liang, Ligu Zhu, Wenqian Shang , Dongyu Feng , Zida Xiao, "Express supervision system based on NodeJS and MongoDB", 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)

[3] Takeshi Ogasawara, "Workload characterization of server-side JavaScript", 2014 IEEE International Symposium on Workload Characterization (IISWC)

[4] Stefan Tilkov ; Steve Vinoski, "NodeJS: Using JavaScript to Build High-Performance Network Programs" , 2010 IEEE Internet Computing

[5] Andres Ojamaa ; Karl Düüna, "Assessing the security of NodeJS platform" 2012 International Conference for Internet Technology and Secured Transactions

[6] Zdenek Brabec ; Tomas Cerny ; Jiri Sebek, "On Metadata Extension to Derive Data Presentations with Angular 2", 2016 6th International Conference on IT Convergence and Security (ICITCS)

[7] Yunhua Gu ; Shu Shen ; Jin Wang ; Jeong-Uk Kim, "Application of NoSQL database MongoDB" 2015 IEEE International Conference on Consumer Electronics - Taiwan

[8] Boyu Hou , Lei Li , Yong Shi , Lixin Tao , Jigang Liu, "MongoDB NoSQL Injection Analysis and Detection" 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)

[9] Benymol Jose , Sajimon Abraham , "Exploring the merits of nosql: A study based on mongodb" 2017 International Conference on Networks & Advances in Computational Technologies (NetACT)

[10] Liberies Vokorokos , "MongoDB scheme analysis" 2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)

[11] Gerd Wagner, "Introduction to Simulation using JavaScript", 17th International Conference 2016

[12]Sharath Gude, Munawar Hafiz, Allen Wirfs-Brock, "JavaScript: The Used Parts", IEEE 38th Annual Computer Software and Applications Conference 2014