RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya

Approved by AICTE, New Delhi

Go, change the world

Major Project

18MCA61

on

Development of quality assurance system for the restructured NPI

Submitted by
AJAY DK
USN: 1RV18MCA03

Under the Guidance of

Dr Jasmine KS
Associate Professor
Department of MCA
RV College of Engineering®
Bengaluru – 560059

Nevadita
QA Lead
21North EuropAssistance
Bengaluru – 560103

Submitted in partial fulfillment of the requirements for the award of degree of

MASTER OF COMPUTER APPLICATIONS

2020-2021

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF
MASTER OF COMPUTER APPLICATIONS

Bengaluru– 560059

CERTIFICATE

Certified that the project work titled **"Development of quality assurance system for the restructured NPI"** carried out by **AJAY DK, USN:1RV18MCA03**, a bonafide student of **RV College of Engineering®, Bengaluru** submitted in partial fulfilment for the award of **Master of Computer Applications** of **RV College of Engineering®, Bengaluru affiliated to Visvesvaraya Technological University, Belagavi** during the year **2020-21.** It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the said degree.

Dr Jasmine KS Associate Professor Department of MCA RVCE, Bengaluru –59	**Dr. Andhe Dharani** Professor and Director Department of MCA RVCE, Bengaluru–59	**Dr. K. N. Subramanya** Principal RVCE, Bengaluru–59

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF
MASTER OF COMPUTER APPLICATIONS

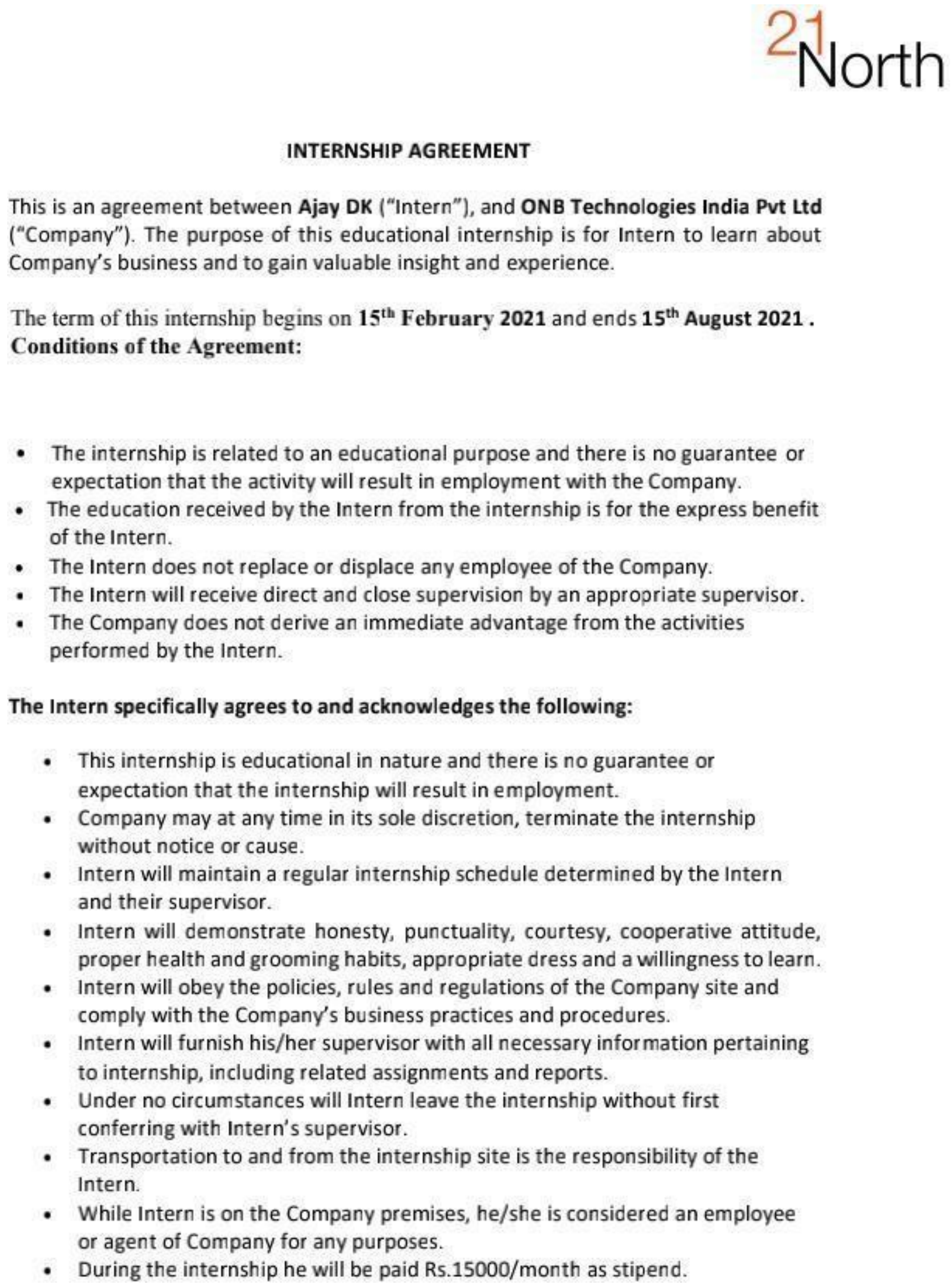
Bengaluru– 560059

DECLARATION

I, **AJAY DK**, student of sixth semester MCA in **Department of Master of Computer Applications,** RV College of Engineering®, Bengaluru declare that the project titled **"DEVELOPMENT OF QUALITY ASSURANCE SYSTEM FOR THE RESTRUCTURED NPI"** has been carried out by me. It has been submitted in partial fulfilment of the course requirements for the award of degree in **Master of Computer Applications** of RV College of Engineering®, Bengaluru affiliated to Visvesvaraya Technological University, Belagavi during the academic year **2020-21**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree or diploma.

Date of Submission :15/06/2021 **Signature of the Student**

Student Name: Ajay DK
USN:1RV18MCA10
Department of Master of Computer Applications
RV College of Engineering®
Bengaluru-560059



21North

INTERNSHIP AGREEMENT

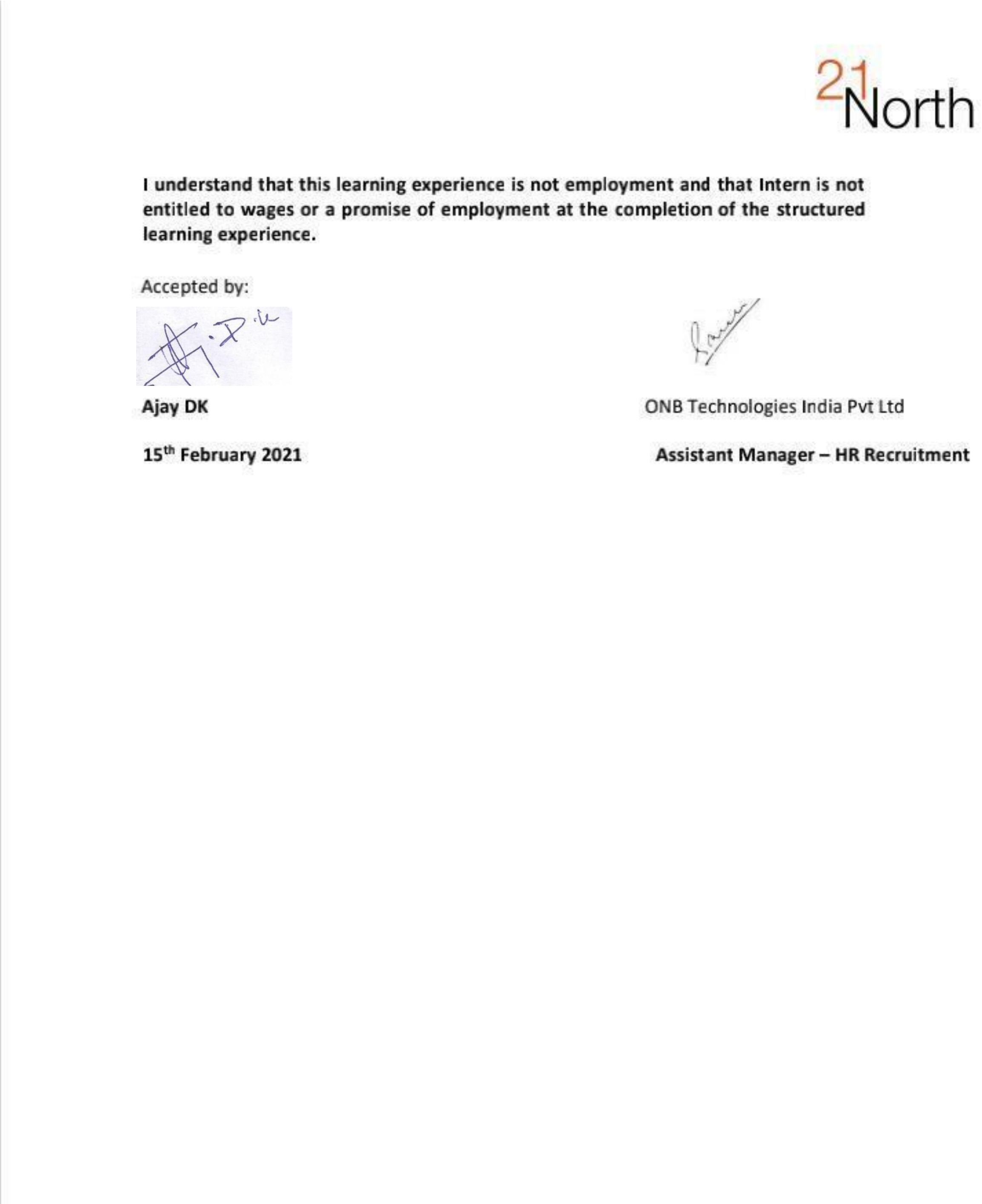
This is an agreement between **Ajay DK** ("Intern"), and **ONB Technologies India Pvt Ltd** ("Company"). The purpose of this educational internship is for Intern to learn about Company's business and to gain valuable insight and experience.

The term of this internship begins on **15th February 2021** and ends **15th August 2021 .**
Conditions of the Agreement:

- The internship is related to an educational purpose and there is no guarantee or expectation that the activity will result in employment with the Company.
- The education received by the Intern from the internship is for the express benefit of the Intern.
- The Intern does not replace or displace any employee of the Company.
- The Intern will receive direct and close supervision by an appropriate supervisor.
- The Company does not derive an immediate advantage from the activities performed by the Intern.

The Intern specifically agrees to and acknowledges the following:

- This internship is educational in nature and there is no guarantee or expectation that the internship will result in employment.
- Company may at any time in its sole discretion, terminate the internship without notice or cause.
- Intern will maintain a regular internship schedule determined by the Intern and their supervisor.
- Intern will demonstrate honesty, punctuality, courtesy, cooperative attitude, proper health and grooming habits, appropriate dress and a willingness to learn.
- Intern will obey the policies, rules and regulations of the Company site and comply with the Company's business practices and procedures.
- Intern will furnish his/her supervisor with all necessary information pertaining to internship, including related assignments and reports.
- Under no circumstances will Intern leave the internship without first conferring with Intern's supervisor.
- Transportation to and from the internship site is the responsibility of the Intern.
- While Intern is on the Company premises, he/she is considered an employee or agent of Company for any purposes.
- During the internship he will be paid Rs.15000/month as stipend.



21North

I understand that this learning experience is not employment and that Intern is not entitled to wages or a promise of employment at the completion of the structured learning experience.

Accepted by:

Ajay DK ONB Technologies India Pvt Ltd

15th February 2021 **Assistant Manager – HR Recruitment**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the success of any work would be incomplete unless I mention the name of the people, who made it possible, whose constant guidance and encouragement served a beacon light and served our effort with success.

I express my wholehearted gratitude to the **Dr. Subramanya K N**, Principal, RV College of Engineering® for providing me an opportunity.

I express my special thanks to **Dr. Andhe Dharani**, Professor and Director, Department of MCA, RV College of Engineering®, Bengaluru for her constant support and guidance.

I express my sincere thanks and wholehearted credit to my Internal guide **Dr. Jasmine K S**, Associate Professor, Department of MCA, RV College of Engineering®, Bengaluru for her constant encouragement, support and guidance during the project work.

I am very thankful to my External guide **Nevadita** for all her support and guidance. I am also thankful to the lab in-charge staff and all faculty of the department for their help and support during the project.
On a more personal note, my deepest appreciation and gratitude to my beloved family, who have been a fountain of inspiration and have provided unrelenting encouragement and support.

AJAY DK

Department of MCA

RV College of Engineering®

Bengaluru-59

ABSTRACT

Network partner interface(NPI) is a web interface used by the organisation where it is used by the service centres for the booking purpose of the vehicles where picked up has to be from the client’s destination.The web interface NPI allows the service station to accurately record all the details of the client’s vehicle, address and the package of service the client requires.The existing system works entirely on stored procedures and PHP which is completely based on monolithic architecture which makes it difficult to have independent components. This makes the existing system less scalable and not conducive for the test driven development. In order to make the existing system more scalable , it has been completely shifted to micro-service architecture which allows individual components to be developed and tested in the test driven environment.

The methodology followed was agile with test driven approach where it allows the testing engineer to constantly test each and every component of the developed module simultaneously with developers. The restructured NPI is developed using Flask micro-framework using python,HTML,Material UI and MySQL.The testing process of the individual components are done using J- meter for load testing, POSTMAN for API Testing, Python-pytest-allure for API integration and Appium for mobile application testing.

The developed system has made it easier to test, understand, and maintain application builds with the combination of independent components. It is easier to carry out the test driven development methodology because of the flexible and isolatable nature of the micro-service architecture. Since the code is written based on the test-cases the quality of the code is high and also reduces the time taken for project development.

Table of Contents

CONTENTS **PAGE NO.**

LIST OFTABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Description

Network partner interface(NPI) is a web interface used by the organisation where it is used by the service centres for the booking purpose of the vehicles where picked up has to be from the client's destination.The web interface NPI allows the service station to accurately record all the details of the client's vehicle, address and the package of service the client requires.The existing system works entirely on stored procedures and PHP which is completely based on monolithic architecture which makes it difficult to have independent components. This makes the existing system less scalable and not conducive for the test driven development. In order to make the existing system more scalable,it has been completely shifted to micro-service architecture which allows individual components to be developed and tested in the test driven environment.

The methodology followed was agile with test driven approach where it allows the testing engineer to constantly test each and every component of the developed module simultaneously with developers. The restructured NPI is developed using Flask micro- framework using python,HTML,Material UI and MySQL.The testing process of the individual components are done using J-meter for load testing, POSTMAN for API Testing, Python-pytest-allure for API integration and Appium for mobile application testing.

The developed system has made it easier to test, understand, and maintain application builds with the combination of independent components. It is easier to carry out the test driven development methodology because of the flexible and isolatable nature of the micro-service architecture. Since the code is written based on the test-cases the quality of the code is high and also reduces the time taken for project development.

1.2 Company Profile

21North Europ Assistance provide services to automobile service centers to outsource non- core activities like pickup and drop-off of the serviceable vehicles so they can focus their time and resources on providing the best service possible to their customers. It is a product- based business to business (B2B) company.21North Europ Assistance is a world-wide vehicle ownership life-cycle assistance platform, this platform continues to revolutionize the auto aftermarket industry for the benefit of global auto and auto-related brands, assistance professionals, and consumers. It combines location-based services, real-time data, artificial intelligence, and end-to- end connected communication, this provides the quickest, safest, and most innovative vehicle assistance service, products, and technology.

The associated partners of the company receive a web interface to automate estimations, invoicing, and communication, reducing errors and increasing transparency. Vehicle pick-up and drop services, chauffer services, as well as online payment options including NEFT, free up space and manpower, allowing the partners to offer a wider range of options to your customers.

The business covers every aspect from customer acquisition to billing and feedback. Customers also can track the concerned vehicles at all times, get 24-hour roadside assistance, detailed billing, and leave instant feedback and ratings.

1.3 Dissertation Organization

The rest of the dissertation is organized as follows: Chapter Two will discuss about the details of the existing systems or projects that are relevant to backend services, as well as which frameworks are best suited to solve the problems. The third chapter focuses on software requirement specifications, particularly functional and non-functional requirements. The system design, or how the APIs are created, is discussed in Chapters four and five, as well as the project's flow. Chapter Six focuses on the project's implementation in terms of the pseudo-codes and procedures used in the project.

The software testing specifications and various types of unit testing that are required for efficient development and debugging are covered in Chapter Seven Chapter Eight focuses on summarizing all of the project's positive and negative aspects in terms of speed, reliability, and development. Finally, the chapters include the project's future scope and closing remarks.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Survey

Individuals' positions, solaces, wellbeing, amusement, choices, and even lives rely upon PC programming in this day and age, so it's smarter to take care of business [1]. Thus, programming testing is critical. Programming testing is similarly just about as significant as some other phase of the product advancement life cycle. Testing should be performed preceding conveying the product for use; this guides in the disclosure of bugs[2]. Programming testing comprises of exercises performed exclusively to recognize mistakes in programming. It approves and checks that the product or item is working appropriately without any blunders or bugs that could cause a deformity. Bugs from past stages ought to be identified during the testing stage to guarantee programming dependability and quality confirmation [3].Product quality is controlled by different boundaries, for example, reaction time, execution, unwavering quality, practicality, rightness, test-capacity, ease of use, and re-ease of use, which can be accomplished through testing [3,4].In spite of the fact that testing can be tedious and about half of the financial plan of programming associations spent on testing [2,4], it is smarter to hit the nail on the head at first than dealing with the expense of a bombed programming thereafter. Thus, the explanation and objective for testing is to lead a stage shrewd and efficient recognition of various classes of blunders inside a base measure of time and less measure of exertion [2,5].

According to Marconi. [8], the micro-service configuration should contain units that each have a single business capability. If the business capability is large, it must be divided into smaller units, and the database in micro-services is defined as a distributed-type database rather than a centralized database. For the modern apps on the cloud , many researchers have been proposed. Here's a list of some of the most recent studies on container service. Microservices, Containers, and their Underestimated Impact on Network Speed, according to Bane Tratzke [18].The libraries are usually accessible through Application Programming Interfaces (APIs) to improve modularity and reduce development time (APIs). The users reinforce the system's reliance on third-party software. At the point when libraries become old or their APIs change, rolling out the fundamental improvements to subordinate frameworks can be tedious, work escalated, and mistake inclined. A procedure is introduced in this paper to decrease the exertion designers should spend to moderate library oldness. The procedure incorporates steps like representation of problem areas, source code investigation, code-based change, and confirmation of the adjusted framework [25].Token-based confirmation permits the certifications of the client's name and secret phrase to get tokens that permit them to recover certain assets. After a token is available, clients can reuse similar token to get to assets without utilizing a username and secret phrase again [28].As businesses expand, so do the demands placed on the IT department. IT departments may begin to lose their ability to effectively provide employees with the technical support they require to do their jobs if they do not have good Help Desk Software.Many businesses rely on IT Ticketing Systems to provide quick, dependable internal customer service, resulting in more efficient IT departments and happier employees [22].Electronic support ticket system that, among other things, seamlessly integrates into inquiries or inquiries sent via e-mail and web forms. The service is easy to use and the electronic support ticketing system has a multi-user web interface that works across multiple networks. Administrators can use the ticket assistance system to manage, archive and organize help requests submitted via a web form on the website [30].

2.2 Existing and Proposed System

The existing NPI system is a Monolithic system which consists of interconnected and interdependent services or applications. It includes a web server, load balancer, and various services such as a network partner interface, a ticketing system, a payment function, and a pickup drop location tracker app, all of which are interconnected and form a large system. So there maybe change of system failure as, if any of the services fails or get interrupted, the entire system will get crashed. Single point of failure is a major issue with monolithic systems.Monolithic architecture is a good idea for quickly getting an application up and running, but it quickly becomes a problem. The monolithic approach becomes a hindrance to growth as the business and number of user grows, customers may expect new user UI, and integration requirements grow. Monolithic applications are notorious for having a long time to market, a steep learning curve for new developers, large dependencies, and a large deployment time.This system is tightly coupled and runs on a server as a self-contained package, that the reason it is no longer meets the needs of scalability and a quick development cycle.

To overcome all the above mentioned flaws, the existing system will be updated to the Microservices architectural pattern. This approach uses loosely coupled services that can be developed, deployed, and maintained independently.Each of these services is in charge of a specific task or say a business logic and can communicate with other services via APIs in order to solve a larger, more complicated business problem.Since the independent services are small, it can be built from the scratch by one or more small teams separated by user story boundaries, making it easier to scale up the development effort if necessary. Once developed, these services can be deployed independent of one another, making it simple to identify and scale hot services without affecting the overall application.They can also be tested in different stages individually there by adapting the test-driven development. Microservices also provide improved fault isolation, so that if one service fails, the entire application does not have to stop working as it was the biggest problem with the existing monolithic architecture.

2.3 Tools and Technologies

PyCharm

PyCharm is an integrated development environment (IDE) for the Python programming language. JetBrains, a Czech company, created it. It includes code analysis, a graphical debugger, an integrated unit tester, VCS integration, and web development with Django and data science with Anaconda.It is possible to get Pycharm for Windows, Mac OS X, and Linux. The Apache License applies to the Community Edition, and there is also a Professional Edition with additional features that is released under a proprietary license.

Python

Python's design philosophy prioritizes the readability of the code, as evidenced by the widespread use of spaces.Structured programming (especially procedural), object-oriented programming, and functional programming are all supported programming paradigms.

Flask

Flask is a WSGI web application framework that is easy to use. It's built to be simple to get started with and scale up to more complex applications. It started as a simple wrapper for Werkzeug and Jinja and has since grown to become one of the most popular Python web application frameworks.Flask makes recommendations but does not impose dependencies or project layout. The developer is free to use whatever tools and libraries he or she wants. The community has a plethora of extensions that make adding new functionality simple.

MySQL

MySQL is a social information base administration framework dependent on the Structured Query Language (SQL), the most broadly utilized language for getting to and overseeing data set records. Under the GNU permit,MySQL is open-source and free programming.

Apache Jmeter

Apache JMeter is a Java-based open source application designed for load testing and performance measurement.

Postman

Postman is a platform for API development. Postman's features help in simplifying each step of building an API and streamline collaboration so you can create better APIs.

Zephyr Squad

Jira's agile test management solutions are ideal for agile teams that focus on test-driven development through test design, execution, and automation.

2.4 Hardware and Software Requirements

2.4.1 Hardware requirements

Processor: Intel Core i3-72000U (from 1.5GHz up to 3.1GHz)

RAM: 4GB RAM DDR4 Memory or higher

Hard Disk Space: 10GB or more

2.4.2 Software requirements

Pycharm 2021.1

Postaman 8.5.1

Zephyr Squad 2.8.6

Jmeter 5.4.1

Flask 1.1.4

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

A software requirements specification (SRS) is a documentation that consists of a detailed description of the functionality of the system. The Software Requirement Specification (SRS) stage of software development is where the requirements of the system under consideration are documented in order to lay the foundation for the software development activities. Consistency, completeness of all essential requirements and their definitions, and correctness are all characteristics of a valid SRS document.

This chapter provides a detailed description of the proposed system as well as its expected results. The system's definitions, general product description and functions, user characteristics, functional and non-functional requirements, and system limitation constraints are all included in the specification.

3.1.1 Definitions,Acronyms and Abbreviations

API:Application program interface.

JSON:Javascript Object Notation.

NPI:Network Partner interface.

3.2 General Description

The work-flow of the system is presented in this section, which includes the product description, product functions, system user characteristics, general constraints, and assumptions.

3.2.1 Product Description

The NPI provides the Service center with the ability to book the services which are requested by the clients. The SVC Admin can book different types of services like Pickup and Dropoff or chauffer services like internal movement,home delivery,stockyard or custody.The SVC admin books the slot with needed services and also books the time slot and enters the address of the client or any other needed addresses based on the service selected by the client.This information is then passed onto the ambassador based on the manhattan location algorithm to the nearest ambassador and the ambassador carries out the services.

The existing NPI system is a Monolithic system which consists of interconnected and interdependent services or applications. It includes a web server, load balancer, and various services such as a network partner interface, a ticketing system, a payment function, and a pickup drop location tracker app, all of which are interconnected and form a large system. So there maybe change of system failure as, if any of the services fails or get interrupted, the entire system will get crashed. Single point of failure is a major issue with monolithic systems. Monolithic architecture is a good idea for quickly getting an application up and running, but it quickly becomes a problem. The monolithic approach becomes a hindrance to growth as the business and number of user grows, customers may expect new user UI, and integration requirements grow. Monolithic applications are notorious for having a long time to market, a steep learning curve for new developers, large dependencies, and a large deployment time. This system is tightly coupled and runs on a server as a self-contained package, that the reason it is no longer meets the needs of scalability and a quick development cycle.

3.2.2 Product Functions

The NPI provides the Service center with the ability to book the services which are requested by the clients. The SVC Admin can book different types of services like Pickup and Dropoff or chauffer services like internal movement,home delivery,stockyard or custody.The SVC admin books the slot with needed services and also books the time slot and enters the address of the client or any other needed addresses based on the service selected by the client.This information is then passed onto the ambassador based on the manhattan location algorithm to the nearest ambassador and the ambassador carries out the services assigned to him. The NPI also provides the SVC with ability to put in credit requests using the interface where in the SVC admin can apply for it via online payment or NEFT approval which inturn will be verified by the finance team on notify and will be added to the SVC credit account.

3.2.3 User Characteristics

The system is used by the people who work in the service center and the ambassador who will provide the service. The user must be familiar with the internet, browsers, and mobile apps.

3.2.4 Assumptions and Dependencies

The user who will use this application should have a good and stable internet connection as well as a smartphone with the latest software update.

3.3 Functional Requirements

3.3.1 Quality Assuarance for Network Partner Interface(NPI)

Input:

Customer requests for service from service center
Service-center Admin will log-in to the NPI

Process:

Show the different services available

Show the booking status

NEFT/Online payment request

Output:

Upcoming Bookings details

NEFT/Online payment Approval/Disapproval Status

3.3.2 Quality Assurance for Notify

Input:

Ambassador attendance status

Tickets are raised at Notify

Ambassador service status

Process:

Resolve the tickets raised by the ambassador

Amabssador is allowed to login

Ambassador live location and current state handling

Output:

Resolve the ticket generated by ambassador

Manage the ambassador status and tickets

3.3.3 Quality Assurance for Ambassador Application

Input:

Login with proper equipments verification

Accept task within 90 seconds task

Process:

Based on the ambassador location, ambassador is

assigned to a particular SVC

Travel to Customer’s location

Raises a ticket incase of issue

Generate payment based on the service

Payment calculation based on the type of service

Output:

Send status to Notify

Invoice Upload Service

completion status Send

payment status to Notify

3.4 External Interfaces Requirements

Zephyr Squad: Jira's agile test management solutions are ideal for agile teams that focus on test-driven development through test design, execution, and automation.

Swagger UI: Swagger UI is a Json based interafce depiction language for portraying restful Apis.Swagger is utilized based on open source services which can be used to create documentations helpful for quality assurance team.

3.5 Non Functional Requirements

Performance

All the Test cases should pass without an exception on the Zephyr squad for the NPI to be working in optimal condition.

Reliability

The developed testcases should be reliable, especially if the internet connection is slow. It takes at least 90KBPS of internet to load the response.

Correctness

The testcases should complete its task while delivering accurate results.

Robustness

When handling exceptions, the testcases should be flexible.The testcases should be able to deal with errors that occur during execution.The testcase fail report should be given out in 15-30 seconds time based on the number of testcases imported onto Zephy squad.

Scalability

The system is scalable because it is built on a microservices architecture.

A scalable,performance microservice is one that is driven by efficiency,that efficiently and is prepared for future tasks or requests.

Security

The API gateway provides API keys, which ensures that the API is well protected.

3.6 Design Constraints

The internet connection should be stable.

Web browser must be JavaScript enabled.

CHAPTER 4

SYSTEM DESIGN

System design provides an overview of the system's architecture, including how the system is connected internally, how work flows within the system, and the concept of complete system components.

4.1 System perspective and Architectural design

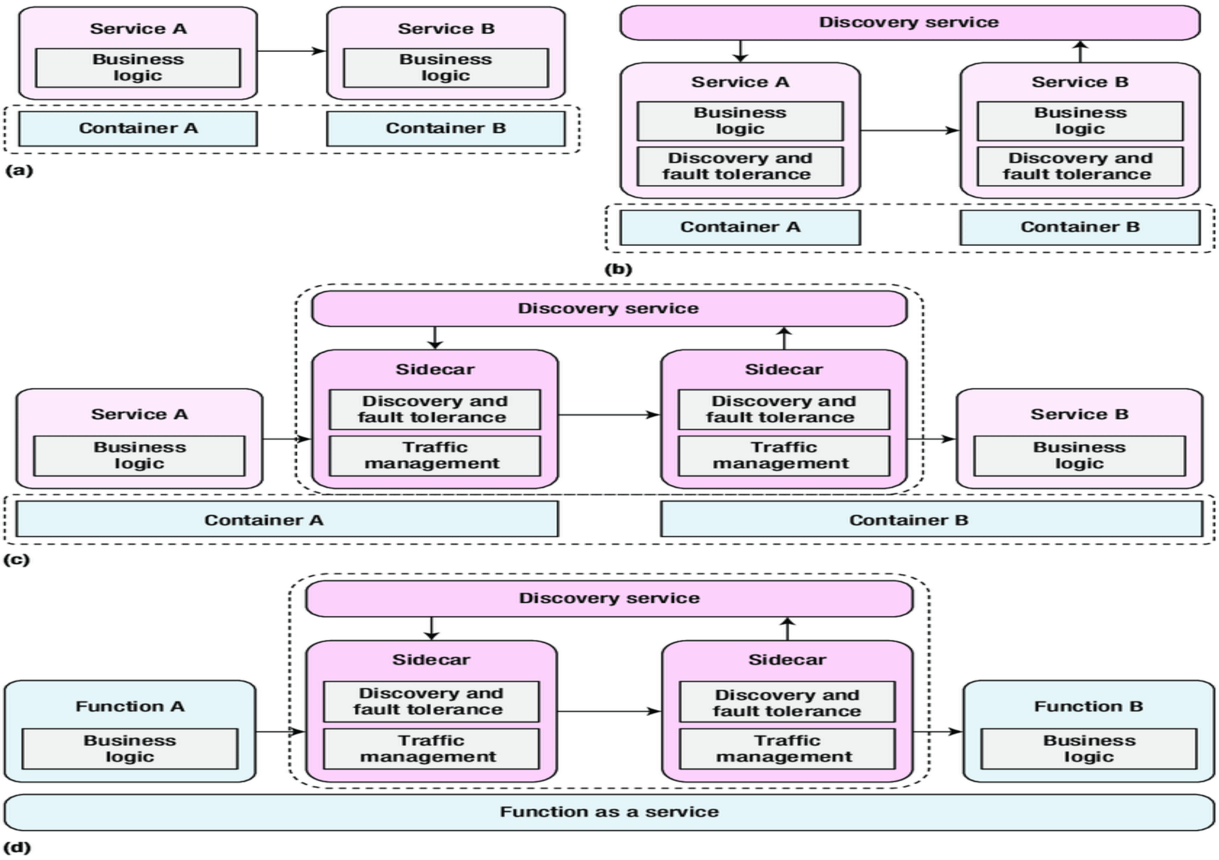


Fig 4.1 Architecture Diagram

Micro service approach uses loosely coupled services that can be developed, deployed, and maintained independently. Each of these services is in charge of a specific task or say a business logic and can communicate with other services via APIs in order to solve a larger, more complicated business problem.Since the independent services are small, it can be built from the scratch by one or more small teams separated by user story boundaries, making it easier to scale up the development effort if necessary. Once developed, these services can be deployed independent of one another, making it simple to identify and scale hot services without affecting the overall Application.

4.1.1 Problem Statement

The existing NPI system is a Monolithic system which consists of interconnected and interdependent services or applications. It includes a web server, load balancer, and various services such as a network partner interface, a ticketing system, a payment function, and a pickup drop location tracker app, all of which are interconnected and form a large system. So there maybe change of system failure grows, customers may expect new user UI, and integration requirements grow. Monolithic applications are notorious for having a long time to market, a steep learning curve for new developers, large dependencies, and a large deployment time. This system is tightly coupled and runs on a server as a self-contained package, that the reason it is no longer meets the needs of scalability and a quick development cycle.

To overcome all the above mentioned flaws, the existing system will be updated to the Microservices architectural pattern. This approach uses loosely coupled services that can be developed, deployed, and maintained independently. Each of these services is in charge of a specific task or say a business logic and can communicate with other services via APIs in order to solve a larger, more complicated business problem.Since the independent services are small, it can be built from the scratch by one or more small teams separated by user story boundaries, making it easier to

scale up the development effort if necessary. Once developed, these services can be deployed independent of one another, making it simple to identify and scale hot services without affecting the overall application. They can also be tested in different stages individually there by adapting the test-driven development. Microservices also provide improved fault isolation, so that if one service fails, the entire application does not have to stop working as it was the biggest problem with the existing monolithic architecture.

4.1.2 Block Diagram

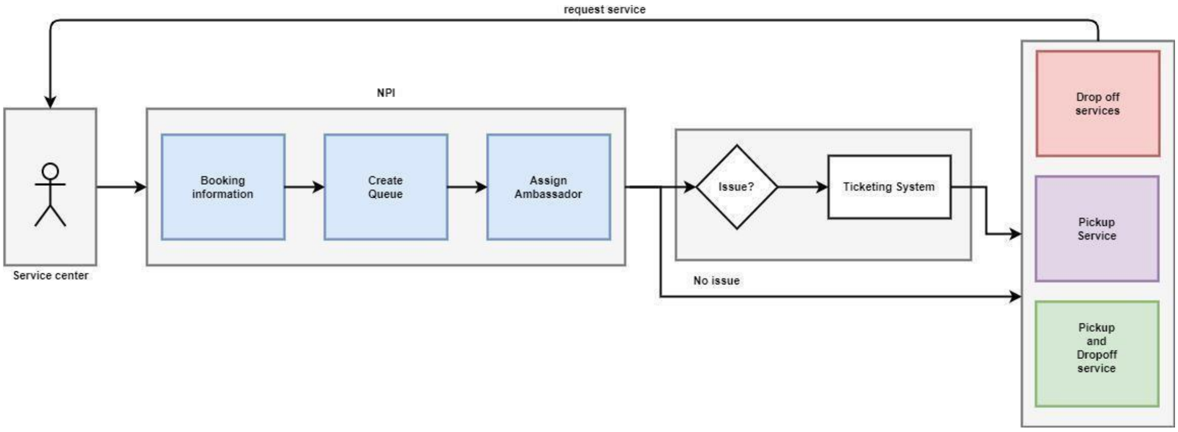


Fig 4.2 Block Diagram

Fig 4.2 shows the flow of the request from customer to the providing of service. When the service is requested by the customer, it is accepted through the NPI and the task is added to ambassador who is nearest to the location.

4.1.3 Module Specification

Quality Assurance for NPI

Services Booking

NEFT/Online Payment request

Chauffer Services

Booking Status

Quality Assurance for Notify

Ambassador attendance management

Ambassador live status and tracking

Ticket handling

Quality Assurance for Invoice Uploader

Challan Upload management

Identifying Duplicate challan uploads

Avaoiding multiple challan upload

Quality Assurance for Ambassador App

Ambassador task assignment

Amabassador task flow

Ambassador payment

4.2 Context Diagram

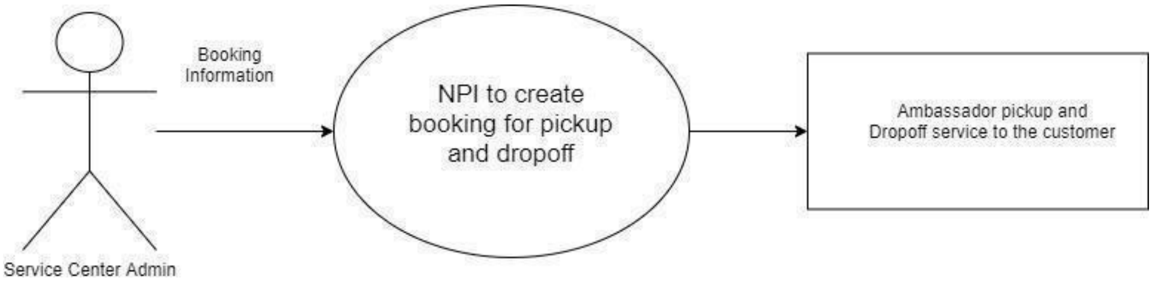

Booking Information
NPI to create booking for pickup and dropoff
Ambassador pickup and Dropoff service to the customer
Service Center Admin

Fig 4.3 Context Diagram

Fig 4.3 shows the flow between the components of the system. This is a high-level diagram which shows the internal process that takes place when a customer requests a service.The SVC admin through NPI approves to provide service based on the request to the type of service by the client. Once the availability of the ambassador is confirmed, the information of the service that is to be provided to the customer is delivered to the ambassador.

CHAPTER 5

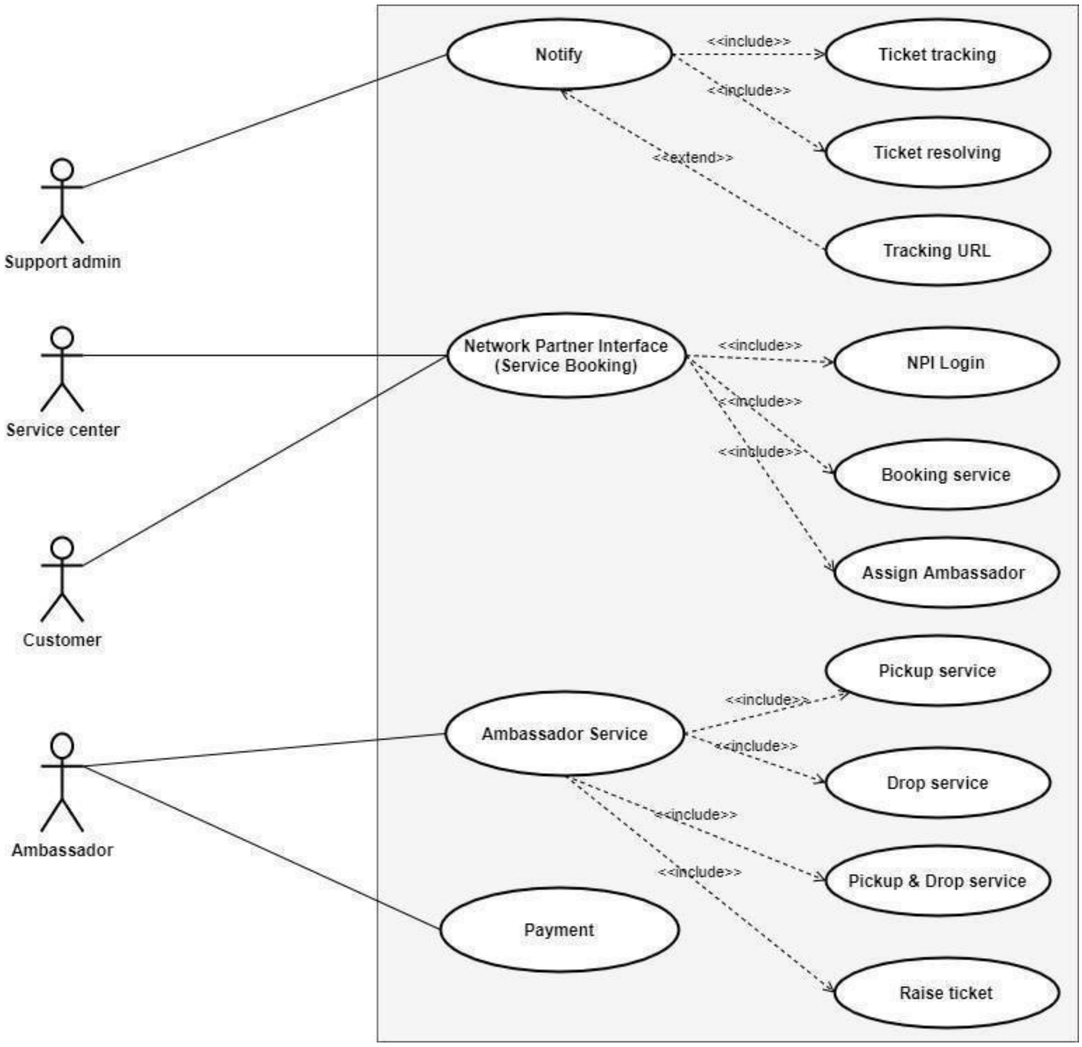
DETAILED DESIGN

5.1 System Design

System design provides an overview of the system's architecture, including how the system is connected internally, how work flows within the system, and the concept of complete system components.

5.1.2 Dynamic Modelling

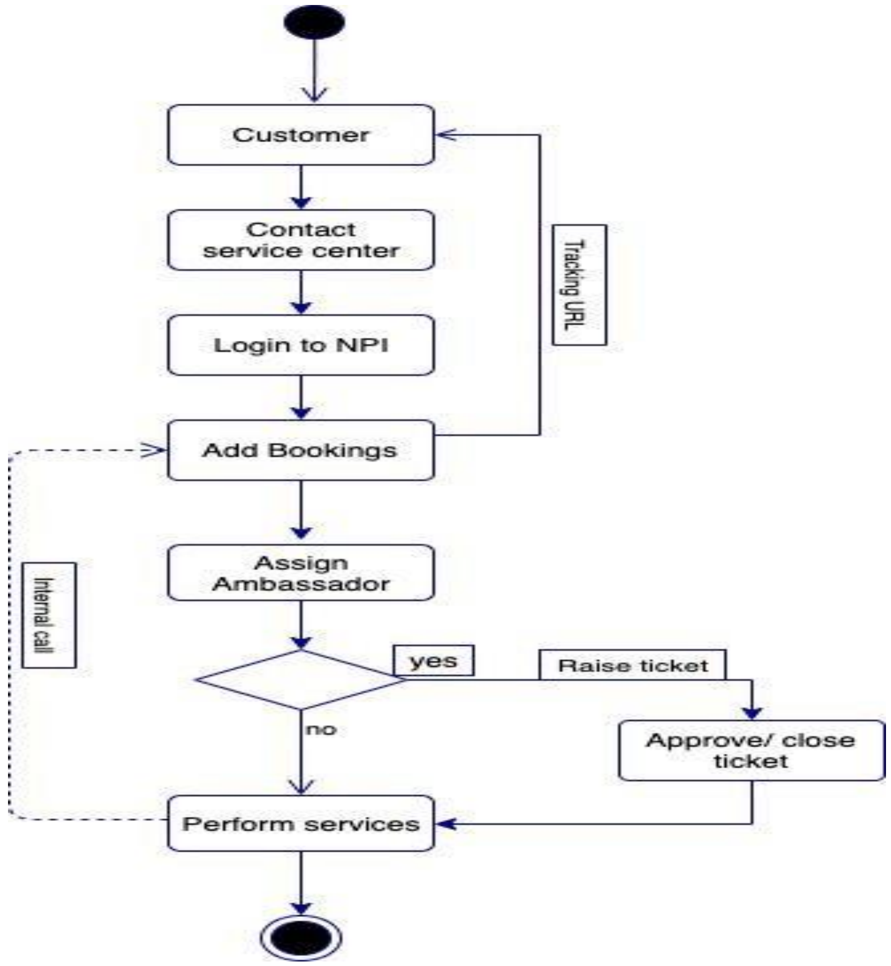
5.1.2.1 Use-case diagram



5.1 Use-case Diagram

Fig 5.1 shows the different actors, Support admin, service center, customer and ambassador present in the system. Each actor has specific functions to perform. The support admin notifies whenever a request is raised. He has to perform ticket resolving, track the URL generated and track the ticket once it is generated. Similarly other actors are also having their respective tasks specified in the diagram.

5.1.2.2 Activity Diagram



5.2 Activity Diagram

Fig 5.2 describes the activities that take place in the system. The activity starts when the Service center admin logs into the system. The customer contacts the service center, logs into the NPI. once the customer books the service, the URL is tracked by the system admin.The ambassador is assigned to the task, ambassador can raise tickets if there is some issue which will be resolved by the tech team in notify.

5.1.3 Functional Modelling

5.1.3.1 Data Flow Diagram

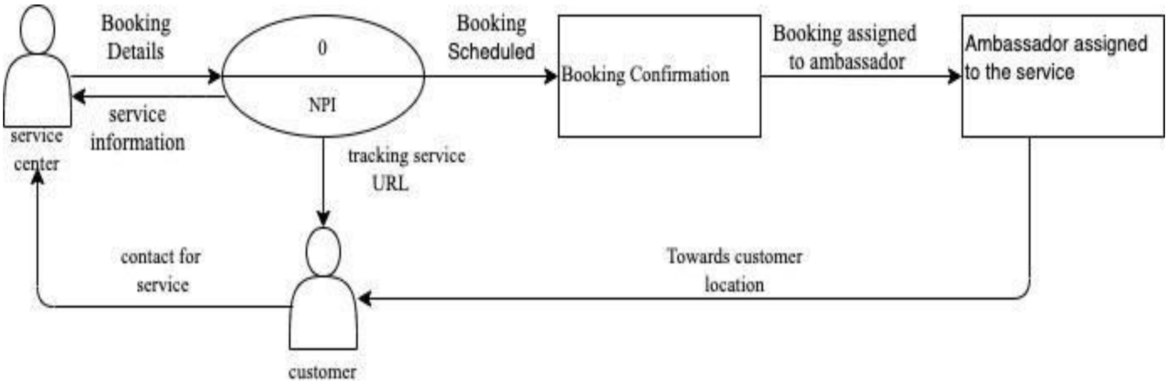


Fig 5.3 DFD Level-0

The fig. 5.3 shows the highest level of the DFD. The process involved when a customer requests a service is shown. The SVC admin via NPI provides service. Once the availability of the ambassador is confirmed, the information of the service that is to be provided to the customer is delivered to the ambassador.

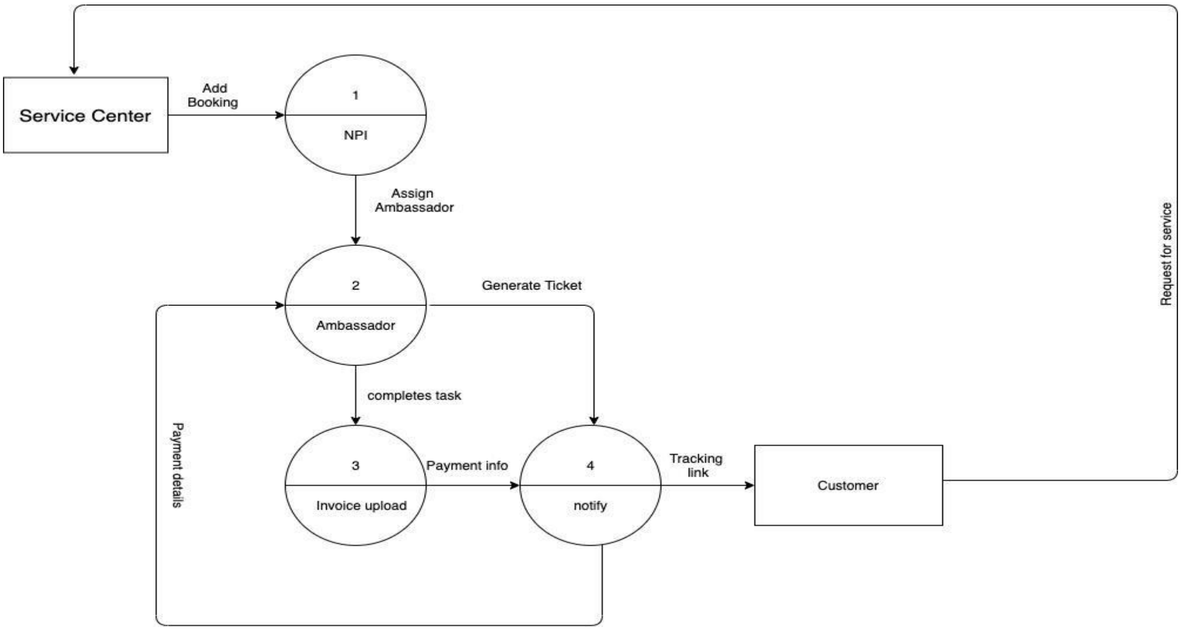


Fig 5.4 DFD Level-1

The fig. 5.4 shows the data flow between customer, service center, ambassador and invoice uploader. When the request is accepted by the SVC admin, the details of the customer is taken and the URL is tracked. The customer information is transferred to the assigned ambassador. Once the service is confirmed without no other issue, the service details with the ambassador detail is send to the customer.

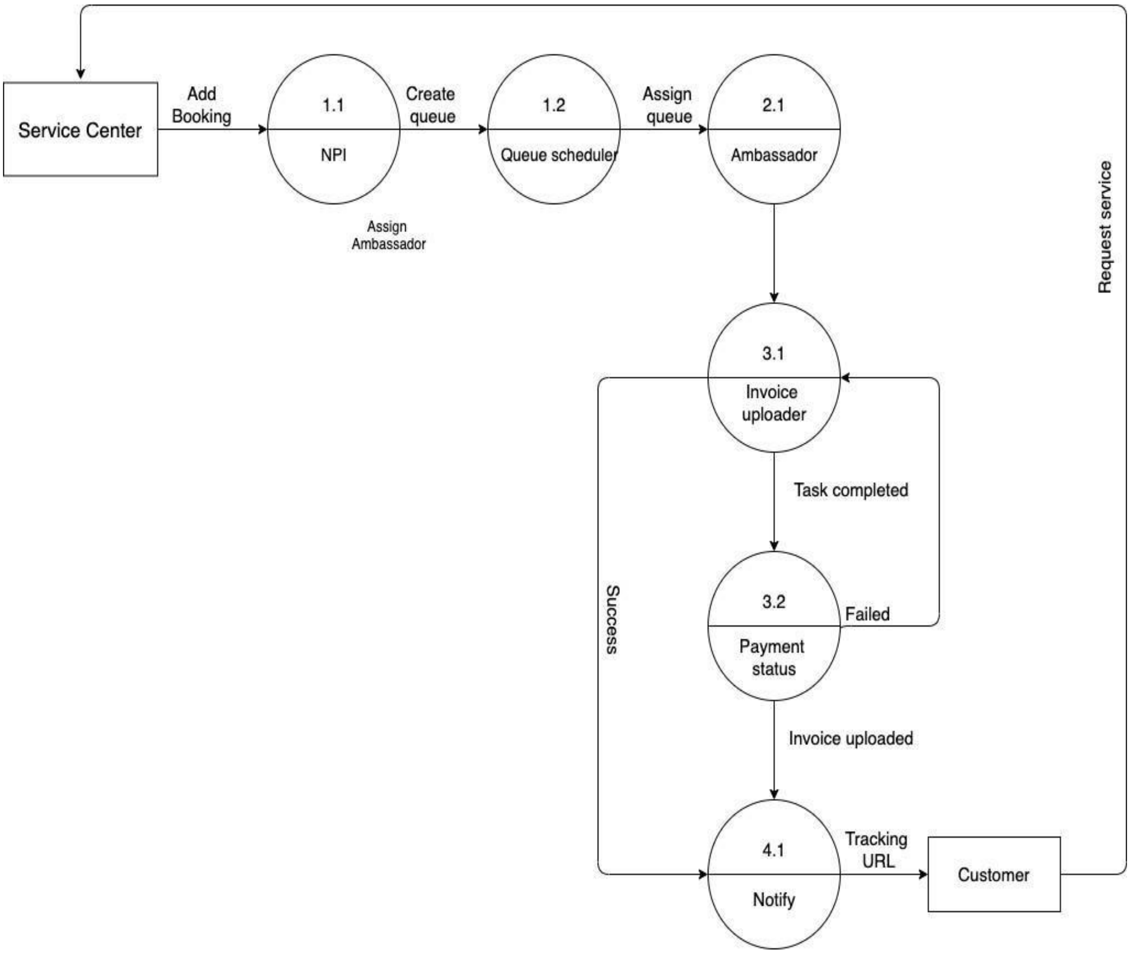


Fig 5.5 DFD Level-2

Fig 5.5 shows the next level of DFD that is the in-depth flow of data that takes place. The data that is flown from the admin to the customer, from the customer to the admin, from admin to the ambassador.

5.1.3.2 Sequence Diagram

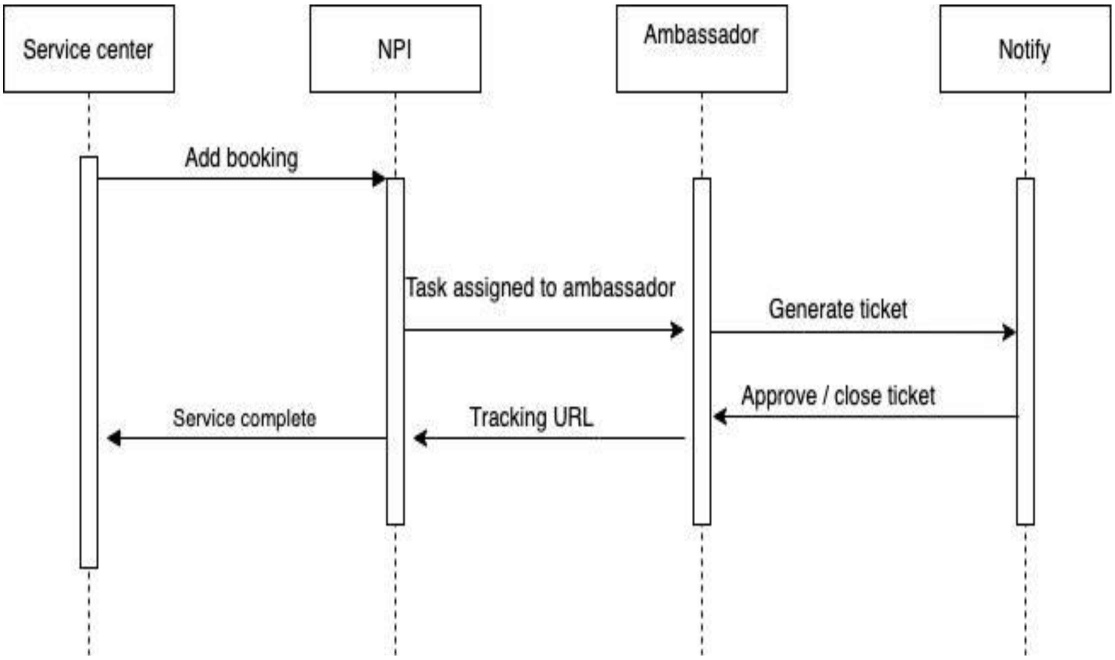


Fig 5.6 Sequence Diagram

Fig 5.6 gives a clear description of the data flow inside the Service center from NPI To the Notify until the task is completed.

CHAPTER 6

IMPLEMENTATION

6.1 Code Snippets

Dashboard

```
    PROCEDURE DASHBOARD
    Manage the user, service request
    DO FOR each new customer
        Track the requested service
        Track the payment details
        print (Service is being tracked)
    ENDDO
```

Booking

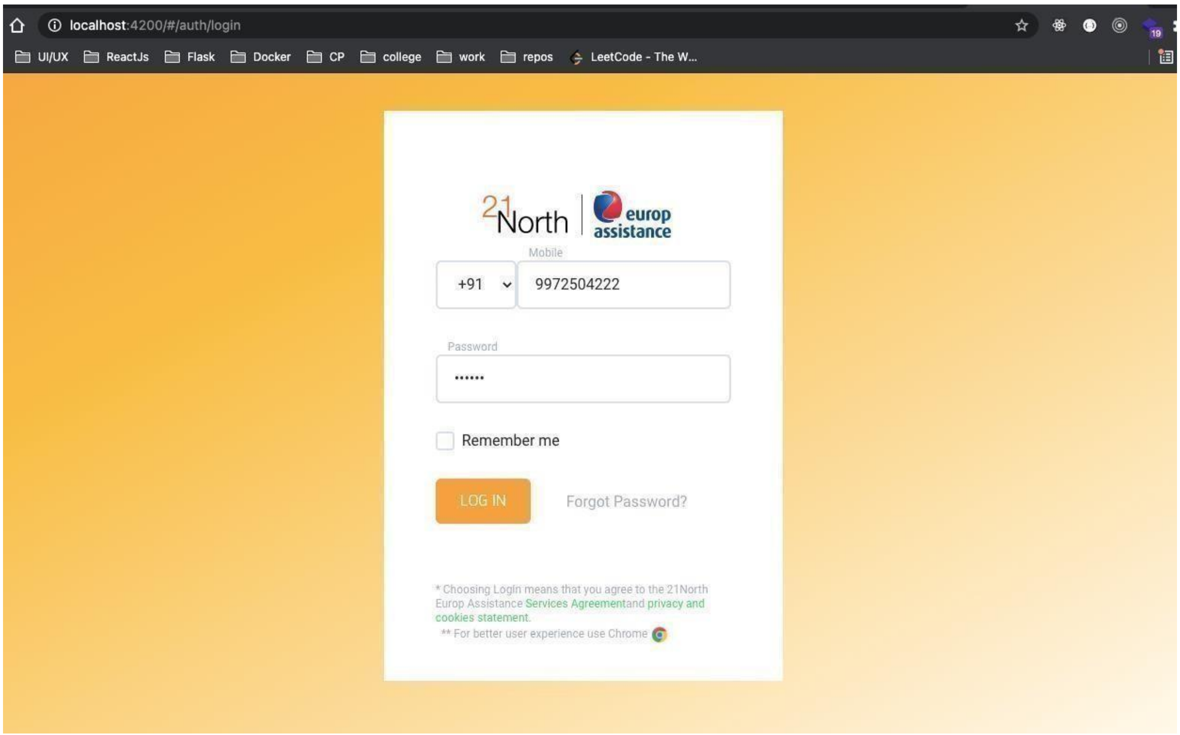
```
    PROCEDURE BOOKING
    Book service for a particular user.
    DO FOR each new customer
        Book service based on the request
        print (Service is booked)
ENDDO
```

Payment

```
    PROCEDURE PAYMENT
    Provide service for a particular user.
    DO FOR each new customer
    If service done THEN
          Generate bill
        print (payment details)
```

```
ENDDO
Notify
    PROCEDURE NOTIFY
    Verify raised ticket for the issue
    DO FOR each new ticket
        IF issue is found THEN
        resolve ticket
        Add details for the ticket print (Ticket is resolved)
    ENDDO
```

6.2 Implementation



Mobile

+91 9972504222

Password

••••••

□ Remember me

* Choosing Login means that you agree to the 21North Europ Assistance Services Agreementand privacy and cookies statement.

** For better user experience use Chrome

Fig 6.1 Login Page

In the figure 6.1 Service center Login Page, the SVC Admin can login to the application to book service for the requested customer. The credentials will be validated against the one which was given during the registration and will be allowed to login if the credentials match, if the credentials do not match then the user will be redirected to the same login page.

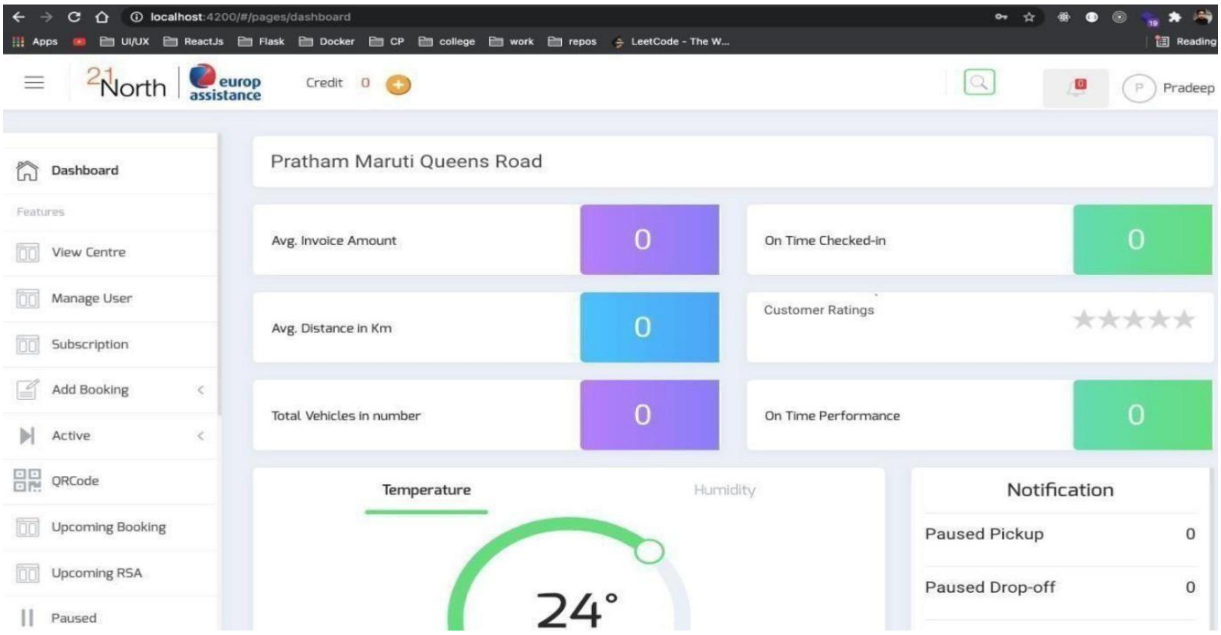


Fig 6.2 NPI Dashboard

The figure 6.2 is the Dashboard Page or the landing page of the product where the service center agent will be landing once they can successfully log in. In this the center member can track each of the details related to service and can manager user. Here they can track the services booked by the customer and the payment related details.

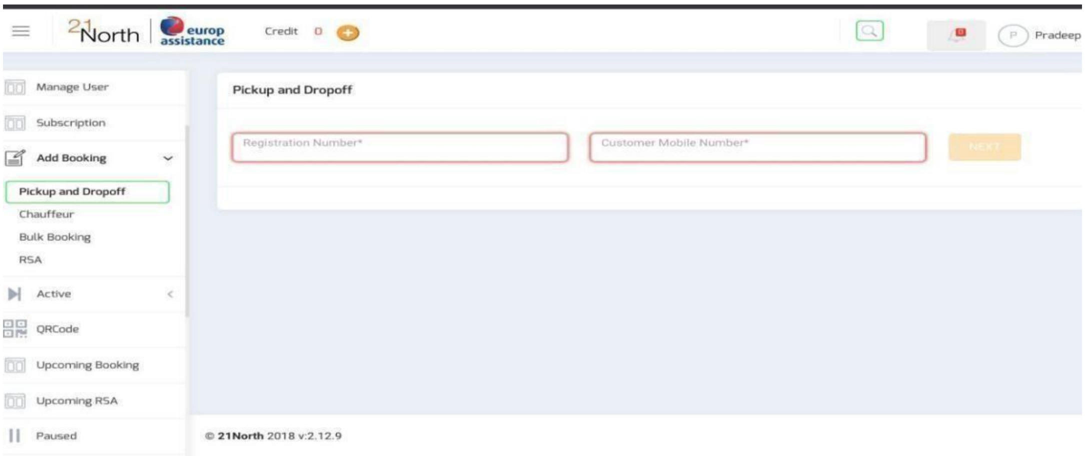
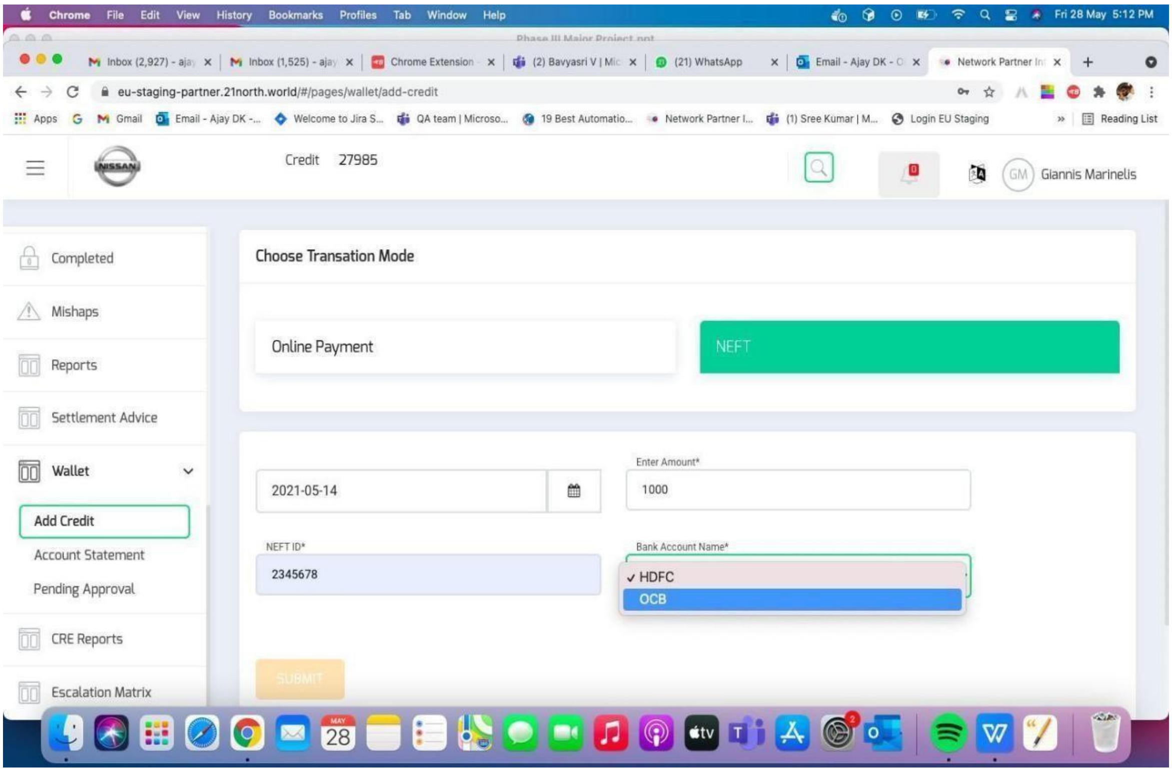


Fig 6.3 Add Booking Page

The figure 6.3 is booking panel where a service center agent can book service for requested user.



Credit 27985

Choose Transation Mode

Online Payment | NEFT

2021-05-14

Enter Amount*: 1000

NEFT ID*: 2345678

Bank Account Name*: ✓ HDFC / OCB

SUBMIT

Fig. 6.4 Add credit option in Wallet

SVC Admin can put in requests for credits for the wallet using this page where in two modes of application is provided like online payment or NEFT.

CHAPTER 7

SOFTWARE TESTING

Every business enterprise need to provide number one cognizance to checking out whilst constructing any project.Testing is an crucial part of improvement , specially to make certain quality. Many smaller and mid- sized corporations don’t provide checking out lots interest albeit it’s important for handing over a strong product. Just like it’s crucial to own checks present, it additionally topics how the company run those checks. Every business enterprise growing a product need to have checks present. Testing is an crucial part of improvement , specially to make certain quality. Many smaller and mid- sized corporations don’t provide checking out lots interest albeit it’s important for handing over a strong product. Just like it’s crucial to own checks withinside the specific situation, it additionally topics how the company run those checks. Some corporations prefer to do guide checking out, even though that’s now no longer the handiest approach. The subsequent logical step is automating your checking out technique the business enterprise have checks present. Manual checking out need to be minimized in any respect costs. And take a look at automation will increase average software program improvement performance and lets in for extra strong equipment to be built.The following are examples of the types of testing carried out as part of this project.

Unit Testing

Testcases are prepared for each and every feature and the entire flow is tested in such a manner that each and every testcase is passed in each of the cycle so that there is no bugs remaining due to code changes after each cycle,which enables to have a very bug free product after the end of the unit testing.

Load Testing

Load testing emphasizes testing of the behaviour of system during the course of load being put on it or computing device can handle high loads given a high demand of end-users.

7.1 Test Cases

Test cases define the set of conditions that a tester will use to determine whether or not a system under test meets the requirement and functions correctly. The Test Case Template contains the following information: Test Case ID, Test Case Summary,Sample Input, Expected Output, and System Performance Remark.

Table. 7.1 NPI login Module

TC_ID	Feature Tested	Sample Input	Actual Output	Expected Output	Result
TC_01	SVC Admin login to NPI	email: ajay@21north.in Pass: 12345	{ “status”: “success”, “res”: “Login successful”, “code”:200 , “JWT”: { “token”:”sdfrtuipo-kjhh12-nbnvc”, }	{ “status”: “fail”,“res”: “Login successful”, “code”:400 }	PASS
TC_02	Wrong login	email: ajj@21north.in Pass: 12345	{ “status”: “fail”, “res”: “Login successful”, “code”:200 , “JWT”: { “token”:”qwertyuipo-kjhh12-nbnvc”,	{ “status”: “success”, “res”: “Login successful”, “code”:200 , “JWT”: { “token”:”qwertyuipo - kjhh12-nbnvc”,	FAIL

Table 7.2 NPI Services Booking in hamburger menu

TC_ID	Feature Tested	Sample Input	Actual Output	Expected Output	Result
TC_01	Pickup booking	Register_no: KA-41-201 Mobile: 1231231231	Select Pickup Address	Select Pickup Address	PASS
TC_02	Drop off Booking	Register_no: KA-05-AH-5524 Mobile: 1231231231	Select Dropoff Address	Invalid data	FAIL
TC_03	Chaufer Booking	Register_no: KA-05-AH-5524 Mobile: 1231231231	Select Address	Select Address	Pass

Table 7.3 Ambassador module

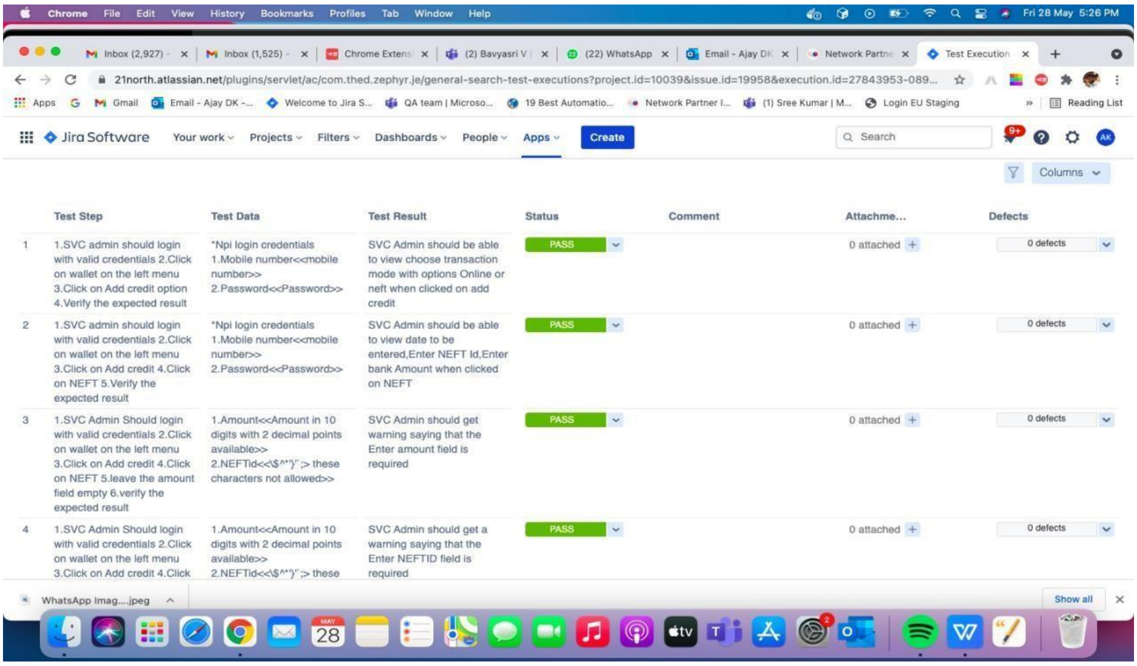
TC_ID	Feature Tested	Sample Input	Actual Output	Expected Output	Result
TC_01	Ambassador available free	Queue id: 123123	Ambassadormarked busy	Ambassadormarked busy	Pass
TC_02	Ambassador marked busy	Queue id: 123123	Ambassador marked free	Unable to Sync API Error	Fail
TC_03	Ambassador tracking	Queue id: 123123	Lat: 25.25 Lon: 33.45	Lat: 25.25 Lon: 33.45	Pass

TC_04	Ambassador completed task	Queue ID: 123123	You have completed the task	You have completed the task	Pass
TC_05	Payment Report	Queue ID: 123123	Details of payment	Details of payment	Pass

Table 7.4 Notify module

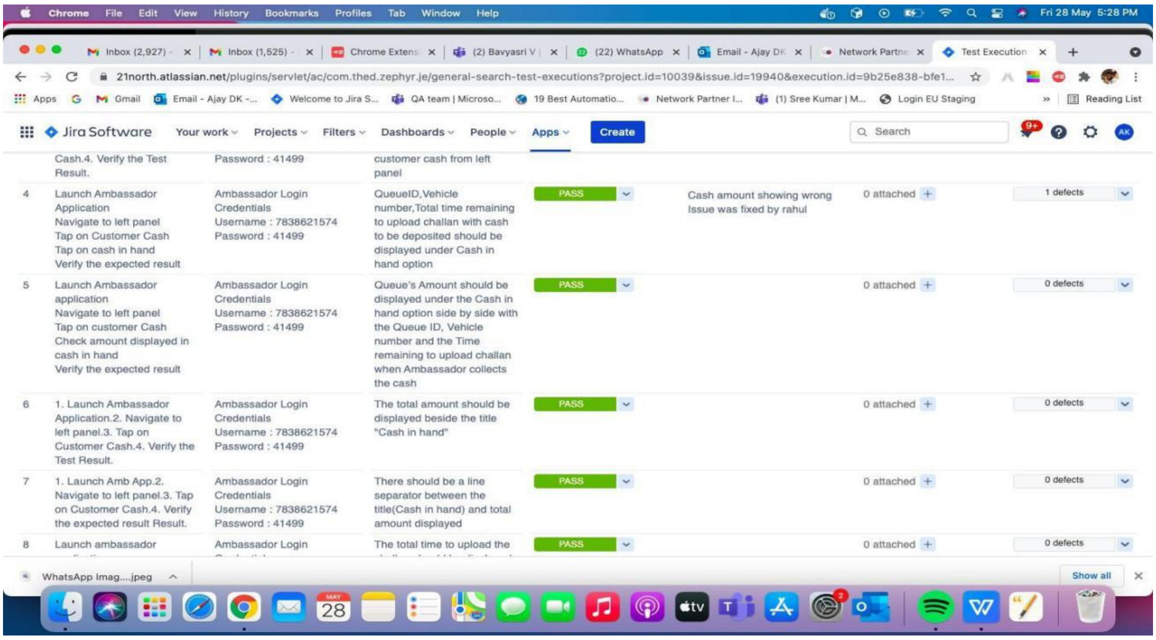
TC_ID	Feature Tested	Sample Input	Actual Output	Expected Output	Result
TC_01	Active pickup Ticket	Amb_id:123 Queue id: 123123	Resolveaddress issue	Resolveaddress issue	Pass
TC_02	Active dropoff ticket	Amb_id:123 Queue id: 123123	Ticket resolved	Ticket resolved	Pass

7.2 Testing and validations


21north.atlassian.net/plugins/servlet/ac/com.thed.zephyr.je/general-search-test-executions?project.id=10039&issue.id=19958&execution.id=27843953-089...
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Columns
WhatsApp Imag....jpeg
Show all

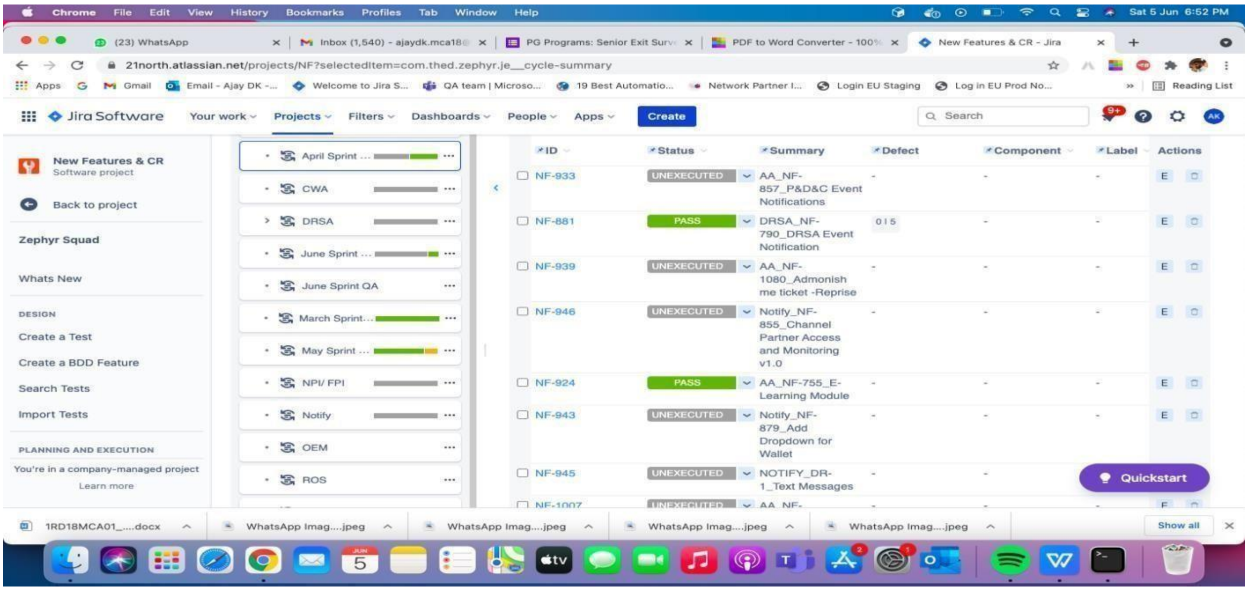
	Test Step	Test Data	Test Result	Status	Comment	Attachme...	Defects
1	1.SVC admin should login with valid credentials 2.Click on wallet on the left menu 3.Click on Add credit option 4.Verify the expected result	*Npi login credentials 1.Mobile number<<mobile number>> 2.Password<<Password>>	SVC Admin should be able to view choose transaction mode with options Online or neft when clicked on add credit	PASS		0 attached	0 defects
2	1.SVC admin should login with valid credentials 2.Click on wallet on the left menu 3.Click on Add credit 4.Click on NEFT 5.Verify the expected result	*Npi login credentials 1.Mobile number<<mobile number>> 2.Password<<Password>>	SVC Admin should be able to view date to be entered,Enter NEFT Id,Enter bank Amount when clicked on NEFT	PASS		0 attached	0 defects
3	1.SVC Admin Should login with valid credentials 2.Click on wallet on the left menu 3.Click on Add credit 4.Click on NEFT 5.leave the amount field empty 6.verify the expected result	1.Amount<<Amount in 10 digits with 2 decimal points available>> 2.NEFTid<<\$^*}";> these characters not allowed>>	SVC Admin should get warning saying that the Enter amount field is required	PASS		0 attached	0 defects
4	1.SVC Admin Should login with valid credentials 2.Click on wallet on the left menu 3.Click on Add credit 4.Click	1.Amount<<Amount in 10 digits with 2 decimal points available>> 2.NEFTid<<\$^*}";> these	SVC Admin should get a warning saying that the Enter NEFTID field is required	PASS		0 attached	0 defects

Fig. 7.1 Test case execution in Zephyr of NPI login module


21north.atlassian.net/plugins/servlet/ac/com.thed.zephyr.je/general-search-test-executions?project.id=10039&issue.id=19940&execution.id=9b25e838-bfe1...
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
WhatsApp Imag....jpeg
Show all

	Cash.4. Verify the Test Result.	Password : 41499	customer cash from left panel				
4	Launch Ambassador Application Navigate to left panel Tap on Customer Cash Tap on cash in hand Verify the expected result	Ambassador Login Credentials Username : 7838621574 Password : 41499	QueueID,Vehicle number,Total time remaining to upload challan with cash to be deposited should be displayed under Cash in hand option	PASS	Cash amount showing wrong Issue was fixed by rahul	0 attached	1 defects
5	Launch Ambassador application Navigate to left panel Tap on customer Cash Check amount displayed in cash in hand Verify the expected result	Ambassador Login Credentials Username : 7838621574 Password : 41499	Queue's Amount should be displayed under the Cash in hand option side by side with the Queue ID, Vehicle number and the Time remaining to upload challan when Ambassador collects the cash	PASS		0 attached	0 defects
6	1. Launch Ambassador Application.2. Navigate to left panel.3. Tap on Customer Cash.4. Verify the Test Result.	Ambassador Login Credentials Username : 7838621574 Password : 41499	The total amount should be displayed beside the title "Cash in hand"	PASS		0 attached	0 defects
7	1. Launch Amb App.2. Navigate to left panel.3. Tap on Customer Cash.4. Verify the expected result Result.	Ambassador Login Credentials Username : 7838621574 Password : 41499	There should be a line separator between the title(Cash in hand) and total amount displayed	PASS		0 attached	0 defects
8	Launch ambassador	Ambassador Login	The total time to upload the	PASS		0 attached	0 defects

Fig. 7.2 Test case execution in Zephyr of NPI add booking module



ID	Status	Summary	Defect	Component	Label	Actions
□ NF-933	UNEXECUTED	AA_NF-857_P&D&C Event Notifications	-	-	-	E
□ NF-881	PASS	DRSA_NF-790_DRSA Event Notification	015	-	-	E
□ NF-939	UNEXECUTED	AA_NF-1080_Admonishme ticket -Reprise	-	-	-	E
□ NF-946	UNEXECUTED	Notify_NF-855_Channel Partner Access and Monitoring v1.0	-	-	-	E
□ NF-924	PASS	AA_NF-755_E-Learning Module	-	-	-	E
□ NF-943	UNEXECUTED	Notify_NF-879_Add Dropdown for Wallet	-	-	-	E
□ NF-945	UNEXECUTED	NOTIFY_DR-1_Text Messages	-	-		
□ NF-1007	UNEXECUTED	AA_NF-	-	-	-	E

Fig. 7.3 Zephyr execution status based on each sprint cycle

CHAPTER 8

CONCLUSION

The restructuring of the NPI has been done in the backend where in the existing feature which runs on stored procedures and PHP has been converted into micro services architecture using python and flask micro framework.The restructured NPI makes it easier to test, understand, and maintain application builds with the combination of independent components. It is easier to carry out the test driven development methodology because of the flexible and isolatable nature of the micro-service architecture.Since the code is written based on the test-cases the quality of the code will be high and also reduces the time taken for project development.The modules developed inside the NPI are loosely coupled so that the modules are independent of each other so the bug fixing, addition of new features, testing and integrating them to the NPI is done in an easier manner when compared to the existing monolithic architecture of NPI.

The previously written APIs which were not built with a proper test-driven development environment often failed to load and the entire web-interface had to be refreshed for them to work properly. But the new APIs are tested rigorously under simulated loads using JMeter such that they can handle huge loads and traffic and should not crash when a huge load is applied to it.

The restructured NPI is much more flexible since it has been converted into micro-service. This architecture enables to add new features to the NPI seamlessly and these new features can be added to the existing environment on the go and the other modules are not effected when a certain module is being upgraded or being removed according to the business needs.The testing is done in multiple cycles using Zephyr. So the system has undergone multiple cycles of trails and errors thereby rectifying the bugs multiple times until the entire system is bug/error free. So the restructured

NPI is working in the optimal condition.

Technical features implemented in the current system are:

Conversion from monolithic architecture to microservice architecture.

Conversion of the backend from stored procedures and PHP to python in the Flask micro-framework environment.

Testing is done parallel in the test driven environment.

Code changes.

CHAPTER 9

FUTURE ENHANCEMENTS

The NPI has been restructured in the backend from stored procedure and PHP to micro- services architecture using python and flask micro framework . The system can be upgraded to give the users a better UI by implementing the GUI using Angular JS and also the entire architecture can be further shrunk down to smaller architecture with a single database using the migrations of tables inbuilt feature provided in the flask micro framework.

In the future,for the database migrations instead of using flask microframework, Django framework can be used as it has much more defined and stricter rules to be followed in terms of file placements so that the migration files can be much more organised for the developer when the size of the database increases.

The testing quality can also be increased by assigning multiple QAs for the job such that the other QA comes up with new perspective for the testing so that the quality of the product can further be increased. Instead of a single person doing multiple cycles of test if a new broader perspective was brought into the testing in the form of a second QA, the quality of the testing can be increased and as a result the code quality would also increase by leaps and bounds.

The technical features which can be implemented in the future are:

Switching from a micro-framework to a normal framework.

Multiple cycles of testing on Zephyr by different QA.

GUI implementation using Angular JS.

BIBLIOGRAPHY

[1] Pressman, R. S. (n.d.). Software engineePressman, R. S. (n.d.). Software engineering (2nd ed.). New York: McGraw-Hill Book Company.ring.

[2] Chauhan, R. K. & Singh, I. (2014). Latest Research and Development on Software Testing Techniques and Tools, 4(4), 2368–2372.

Wala, T. & Sharma, A. K. (2014). Improvised Software Testing Tool, 3(9), 573–581.

[3] Merina, C. (2019). Tool Usability Parameter in Determining the Performance of Software Testing Tool, IJTB (International J. Technol. Business), 3(1), 8–18.

[4] Testing in Global Software Development Environment, Data Science and Analytics, 309- 317, DOI: 10.1007/978-981-10-8527-7_25.

[5] Tassey, G. (2002). "The Economic Impacts of Inadequate infrastrure for software testing". Natl. Inst. Stand. Technol.

[6] Raulamo-Jurvanen, P., Mäntylä, M. & Garousi, V. (2017). Choosing the Right Test Automation Tool, in 21st International Conference on Evaluation and Assessment in Software Engineering (EASE 2017), 21–30.

[7] Jain, V. & Rajnish, K. (2018). Comparative Study of Software Automation
Testing Tools: OpenScript and Selenium, Int. J. Eng. Res. Appl., 8(2), 29–33.
https://doi.org/10.9790/9622-0802032933

[8] Odun-Ayo, I., Falade, A., & Samuel, V. (2018). Cloud Computing and Open Source Software: Issues and Developments, Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2018, 14-16 March, 2018, Hong Kong,140-145.

[9] Uppal, N. & Chopra, V., (2012). Design and Implementation in Selenium IDE with Web, Int. J. Comput. Appl., 46(12), 8–11.

[10] A.Martin-Lopez, S.Segura and A.Ruiz-Cortés, "A Catalogue of InterParameter Dependencies in RESTful Web APIs", Intern. Conference on Service- Oriented Computing, pp. 399-414, 2019.

[11] T. Fertig and P. Braun, "Model-Driven Testing of RESTful APIs", Intern.

Conferenceon World Wide Web, pp. 1497-1502, 2015.

[12] S. Yatskiv, I. Voytyuk, N. Yatskiv, O. Kushnir, Y. Trufanova and V. Panasyuk, "Improved Method of Software Automation Testing Based on the Robotic Process Automation Technology", 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), pp. 293-296, June, 2019

[13] "The Difference between Robotic Process Automation and Artificial Intelligence Available", pp. 293-296, June, 2018

[14] J.G.Enríquez,A.Jiménez-Ramírez, F. J. Domínguez-Mayo, J. A. García-García, "Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study",Access IEEE, vol. 8, pp. 39113-39129, 2020

[15] S. Aguirre and A. Rodriguez, "Automation of a business process using robotic process automation (rpa): A case study", Workshop on Engineering Applications, pp. 65-71, 2017.

[16] C.Lamberton, D.Brigo and D.Hoy, "Impact of robotics rpa and ai on the insurance industry: challenges and opportunities", Journal of Financial Perspectives: Insurance edition, 2016.

[17] Kaur, H. & Gupta, G. (2013). Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete, Int. Journal of Engineering Research and Applications, 3(5).

[18] H. Ed-douibi, J.L.C. Izquierdo and J. Cabot, "Automatic Generation of Test Cases for REST APIs: A Specification-Based Approach", IEEE 22nd Intern. Enterprise Distributed Object Computing Conference, pp. 181-190, 2018.

[19] T. Fertig and P. Braun, "Model-Driven Testing of RESTful APIs", Intern. Conferenceon World Wide Web, pp. 1497-1502, 2015.

[20] S. Yatskiv, I. Voytyuk, N. Yatskiv, O. Kushnir, Y. Trufanova and V. Panasyuk, "Improved Method of Software Automation Testing Based on the Robotic Process Automation Technology", 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), pp. 293-296, June, 2019

[21] "The Difference between Robotic Process Automation and Artificial Intelligence

Available", pp. 293-296, June, 2018

[22] J.G.Enríquez,A.Jiménez-Ramírez, F. J. Domínguez-Mayo, J. A. García-García, "Robotic Process Automation: A Scientific and Industrial Systematic Mapping Study",Access IEEE, vol. 8, pp. 39113-39129, 2020

[21] S. Aguirre and A. Rodriguez, "Automation of a business process using robotic process automation (rpa): A case study", Workshop on Engineering Applications, pp. 65-71, 2017.

[22] C.Lamberton, D.Brigo and D.Hoy, "Impact of robotics rpa and ai on the insurance industry: challenges and opportunities", Journal of Financial Perspectives: Insurance edition, 2016.

[23] Kaur, H. & Gupta, G. (2013). Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete, Int. Journal of Engineering Research and Applications, 3(5).

[24] Mustafa, K. M., Al-Qutaish, R. E., & Muhairat, M. I. (2009). Classification of software testing tools based on the software testing methods, in 2009 International

[25] Conference on Computer and Electrical Engineering, ICCEE 2009, 1, 229–233. Kolli, R. (2016). An Empirical Study on Software Test Estimation.

[26] Raulamo, P., Hosio, S. & Mäntylä, M. V (2019). Practitioner Evaluations on Software Testing Tools. In Proceedings of the Evaluation and Assessment on Software Engineering, 57–66.

[27] Vos, T., Marín, B., Panach, I., Baars, A., Ayala, C., & Franch, X. (2011). Evaluating software testing techniques and tools. Actas de Las 16th Jornadas de Ingenieria Del Software y Bases de Datos, JISBD 2011, pp. 531–536.

[28] Kannan, S. & Pushparaj, T. (2014). A Study on Variations of Bottlenecks in Software Testing, International Journal of Computer Sciences and Engineering, 2(5), 8 - 14.

[29] Malik, V. & Gahlan, M. (2016). Comparative Study of Automated Web Testing Tools, Int. J. Latest Trends Eng. Technol., 6(3), 1- 8.

Top 10 Testing Automation Tools for Software Testing, pp.

[30]. Giuseppina Cretella and Beniamino Di Martino, "An Overview of Approaches for the Migration of Applications to the Cloud," in Lecture Notes in Information Systems and Organisation, Leonardo Caporarello,Beniamino Di Martino, and Marcello Martinez, Eds.: Springer, 2014, pp.

[31]. Gos, Konrad & Zabierowski, Wojciech. (2020). The Comparison of Microservice and Monolithic Architecture. 150-153. 10.1109/MEMSTECH49584.2020.9109514.

[32]. C. M. Aderaldo, N. C. Mendonça, C. Pahl, and P. Jamshidi, "Benchmark requirements for microservices architecture research," in Proceedings of the 2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering, pp. 8–13, Buenos Aires, Argentina, May 2017

[33]. C. Esposito, A. Castiglione, and K.-K. R. Choo, "Challenges in delivering software in the cloud as microservices," IEEE Cloud Computing, vol. 3, no. 5, pp. 10–14, 2016.

[34]. G. Mazlami, J. Cito, and P. Leitner, "Extraction of microservices from monolithic software architecture," IEEE International Conference on Web Services, vol. 40, no. 11, pp. 524–531, 2017

[35]. Sachchidanand Sing, Nirmala Singh, "Containers & Docker: Emerging Roles

[36]. N. Alshuqayran, N. Ali and R. Evans, "A Systematic Mapping Study in Microservice Architecture," 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), 2016, pp. 44-51, doi: 10.1109/SOCA.2016.15.

[37]. Chen Yang, "Checkpoint and Restoration of Micro-service in Docker Containers," School of Information Security Engineering, Shanghai Jiao Tong University, China 200240

[38]. D. N. Jha, S. Garg, P. P. Jayaraman, R. Buyya, Z. Li and R. Ranjan, "A Holistic Evaluation of Docker Containers for Interfering Microservices," 2018 IEEE International Conference on Services Computing (SCC), 2018, pp. 33-40, doi:

10.1109/SCC.2018.00012

[39]. Fawaz Paraiso, St´ephanie Challita, Yahya Al-Dhuraibi, Philippe Merle, "Model- DrivenManagement of Docker Containers"., University of Lille & Inria Lille - Nord Europe 2016

[40]. Alin Calinciuc, Cristian Constantin Spoiala, Corneliu Octavian Turcu, Constantin Filote, "OpenStack and Docker: building a high-performance IaaS platform for interactive social media applications", May 19-21, 2016

[41]. R. R. Yadav, E. T. G. Sousa, and G. R. A. Callou, Docker Containers Versus Virtual Machine-Based Virtualization: Proceedings of IEMIS 2018

[42]. D. Liu, H. Zhu, C. Xu, I. Bayley, D. Lightfoot, M. Green, and P. Marshall. Cide: An integrated development environment for microservices. In 2016 IEEE International Conference on Services Computing (SCC), pages 808–812, June 2016

[43]. Bhawiyuga A, Data M and Warda A 2017 Architectural Design of Token based Authentication of MQTT Protocol in Constrained IoT Device, 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 1–4

[44]. Zheng K and Jiang W 2014 A Token Authentication Solution for Hadoop Based on Kerberos PreAuthentication, 2014 International Conference on Data Science and Advanced Analytics (DSAA) 354–60

[45]. Ethelbert O, Moghaddam F F , Wieder P and Yahyapour R A 2017 JSON Token- Based Authentication and Access ManagementSchema for Cloud SaaS Applications, in 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (Cornel University Library) p 6

[46]. Mestre P, Madureira R, Melo-Pinto P and Serodio C 2017 Securing RESTful Web Services using Multiple JSON Web Tokens, in Proc. World Congress on Engineering 2017 418–23

[47]. Memeti A, Imeri F and Cico B 2017 REST vs. SOAP: Choosing the best web service while developing in-house web applications, Journal of Natural Sciences and Mathematics of UT 2(3) 63–8

[48]. Shofa R N, Aradea and Kurnia B B 2013 Penerapan Service Oriented

Architecture (SOA) Dalam Pembangunan Web Based Learning, Jurnal Penelitian SITROTIKA 9(2) 221–19

[49]. Pavlovski C, Warwar C, Paskin B and Chan G 2015 Unified Framework for Multifactor Authentication, 22nd International Conference on Telecommunications (ICT 2015) 209–13

[50]. Balaj Y 2017 Token-Based vs Session-Based Authentication: A survey 1–5

AJAY D K-ksj

ORIGINALITY REPORT

11 %	7%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	medium.com Internet Source	2%
2	Submitted to Victorian Institute of Technology Student Paper	1%
3	Submitted to International University - VNUHCM Student Paper	1%
4	stackify.com Internet Source	1%
5	www.21north.world Internet Source	1%
6	www.hindawi.com Internet Source	<1%
7	Submitted to Anatolia College Student Paper	<1%
8	Submitted to Middle East Technical University Student Paper	<1%
9	www.researchgate.net Internet Source	<1%

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

TWsistocetifythotthepoperentiled cERTIFICATE ID:IJAEMA/5713

"Operational Analysis and Comparison of Software Automation Testing Tools"

Authored by :

Ajay .D .K

From

RV College of Engineering, Bengaluru

Has been published in

IJAEMA JOURNAL, Y0LUME XIII, ISSUE V, MAY- 202T

ACADEMIC EXCELLENCE

ज्ञान-विज्ञान विमुक्तये
UGC
APPROVED

Michal A. Olszewski Editor-In-Chief
IJAEMA JOURNAL

IMPACT FACTOR

ISO International Organization for Standardization
7021-2008

http://ijaema.com/

Operational Analysis and Comparison of Software Automation Testing Tools

Ajay D K
VI Sem, MCA
Department of MCA
RV College of Engineering,
Bengaluru

Dr Jasmine KS
Associate Professor
Department of MCA
RV College of Engineering,
Bengaluru

Abstract- Every company should give primary focus to testing while building any project.Testing is an important a part of development , especially to ensure quality. Many smaller and mid- sized companies don’t give testing much attention albeit it’s essential for delivering a robust product. Just like it’s important to possess tests present, it also matters how the corporate run these tests. Every company developing a product should have tests present. Testing is an important a part of development , especially to ensure quality. Many smaller and mid- sized companies don’t give testing much attention albeit it’s essential for delivering a robust product. Just like it’s important to possess tests in the particular situation, it also matters how the corporate run these tests. Some companies favor to do manual testing, although that’s not the most effective approach. The next logical step is automating your testing process the company have tests present. Manual testing should be minimized at all costs. And test automation increases overall software development efficiency and allows for more robust tools to be built.

The study has concentrated on analyzing and comparing the software automation testing tools based on the criteria like UI, Understand-ability, Programming knowledge required, Features and functionality, Value for money and the type of testing. The ultimate goal is to create a document which gives clear indication regarding the performance of some of the popular software automation testing tools.This study is going to be helpful for software testing engineer while choosing which automation tool to use for the project.The tester will get a clear idea by comparing the software testing tools and then choosing the most appropriate tool suited for the testing of the project.

Keywords-Software Testing, Software Testing Tools, Software Automation testing

I. INTRODUCTION

Software testing is a crucial part of software development, it ensures that the software being developed performs all functional requirements and is free from any form of defect and errors. This ensures that the software is of excellent quality and standard. While testing a software, it's important to be time and price consciousness. This very cause is the reason why testers are more inclined towards automation testing than manual testing. It also helps to reduce time and cost. But then selecting a software testing tool for automated testing that best fit a project is vital yet challenging task, the target of this paper is to gauge some of the foremost used software testing tool, identify their strength and weakness and also the sector where they will be used , either for mobile testing, web service testing or both.

The method utilized in the paper involved the analysis of recommended literature to pinpoint necessary testing tools selected based on inclusion and exclusion criteria, that were evaluated. The output of the research showed that based on the certain criteria, testing tools that supports web platform testing made up 17%, while tools that supported desktop and mobile platforms was 12% and 17% respectively. also, 6% of the tools examined were found to be open source tools while 15% were licensed tools. 9% of the testing tools examined supported the test result and report generation criterion while tools that need the knowledge of programming language was least as only 4% of examined tools supported this criterion. It was observed that there is no one perfect tool for testing, but for a particular testing purpose, trade-offs can be made to select the best tool depending on the size of the project, the budgeted cost for testing, the platform of the appliance and also the language that's needed develop the project.

The primary advantages of using the software automation test tools include saving the testers time, Checks for optimal code quality, early bug detection, re-usability,easy and robust reporting,man power utilization.

II. LITERATURE SURVEY

In the world today, peoples' jobs, comforts, safety, entertainment, decisions, and the very lives depends on computer software, so it's better to be done correctly [1]. This is the reason software testing has great importance. Software testing is as essential as the other stage of the software development life cycle. Testing must be done before deploying the software for use, thishelps to discover errors on time and ensure the functionality of the software works as necessary, it also minimize defect discovered after the merchandise has gone into use. Software testing may be a significant a part of software quality assurance (SQA), it's an activity used for evaluating and improving software quality [2]. Software testing comprises of activities carried out with the single aim of finding errors in software. It validates and verifies if the software or product is functioning correctly with none errors or bugs capable of incurring defect. In the phase of testing, bugs from previous phases should be detected, this ensures software reliability and quality assurance [3]. The product quality relies on different parameters such as response time, performance,reliability, maintainability, correctness, test-ability, usability and re- usability are also measured to know how well the quality of a software has been attained, this is achievable via testing [3,4].

Although testing can be time consuming and about 50% of the budget of software organizations spent on testing [2,4], it is better to get it right initially than handling the cost of a failed software afterwards. Hence, the reason and objective for testing is to conduct a step-wise and systematic detection of different classes of errors within a minimum amount of time and less amount of effort [2,5].

III. ADVANTAGES OF SOFTWARE AUTOMATION TESTING

Faster Feedback Cycle: Without automation testing, testing for newly built features can take a while. Test automation helps to scale back the feedback cycle and convey faster validation for phases within the development of your product.Test automation is particularly useful because it helps to seek out issues and bugs initially on during the development phase, which increases the team’s efficiency. Team Saves Times by automating testing procedure, the team has to spend less time validating newly developed features. It increases the level of communication with other departments like marketing, design, or product owners who rely on the results of these tests. These departments can easily check the logs of the automated tests and see what’s happening.Reduced Business Expenses When using an automated test environment, the company will save money as fewer resources are spent on testing the product. The idea is that the company should not be doing any manual testing.

Higher Coverage of Tests: Manual testing limits the number of tests that can be verified.This increases the test coverage for your product, so more features are properly tested resulting in a higher quality application.Also, automated testing allows developers to write more in-depth tests that test complex cases.Lengthy tests that are often avoided during manual testing can be run unattended

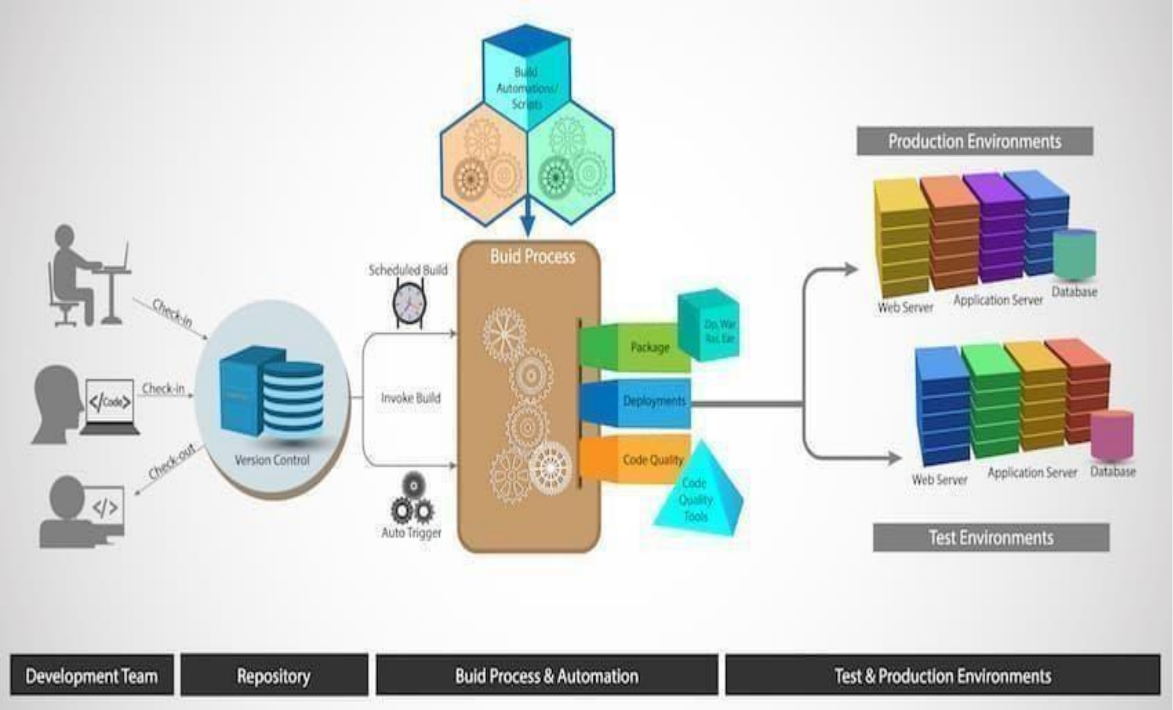


Figure 1.1 Architecture Diagram of Software automation testing system

IV. TOOLS AND TECHNOLOGIES

Open Source Automation Testing Tools

Watir is a light weighted automated software automation testing tool, Watir can be used for cross-browser testing and data-driven testing. Watir is often integrated with Cucumber, Test/Unit, and RSpec, and is free and open source.Watir is an ideal product for organizations that are looking to automate their web testing also as for a business that works during a Ruby environment.

Galen is unique on this list, the Galen is designed for those who want to automate the user experience testing. Galen may be a niche, specific tool which will be wont to verify that your product goes to seem because it should on most platforms. Once testing has been completed, Galen can create detailed reports, including screen-shots, and may help developers and designers analyze the looks of their application over a mess of environments. Galen can also carry out automated tasks using JavaScript, Java, or the Galen Syntax.

Selenium is undoubtedly the foremost popular automated security testing option for web applications. However, it's been extended very often to feature functionality to its core. Selenium is employed in everything from Katalon Studio to Robot, but alone, it's primarily a browser automation product. Those who believe they're going to be actively customizing their automated test environments might want to start out with Selenium and customize it from there. In contrast, those that wish to start during a more structured test environment could also be more happy with one among the systems that are built on top of Selenium. Selenium are often scripted during a multitude of languages, including Java, Python, PHP, C#, and Perl. Selenium isn't as user-friendly as many of the opposite tools on this list; it's designed for advanced programmers and developers. Selenium are often described as a framework for a framework. Many of the foremost modern and specialized frameworks draw design elements from Selenium. They are also often made to figure together with Selenium. Its application was testing web application originally, but over some it has grown considerably. Selenium is compatible with C#, Python, Java, PHP, Ruby, and virtually the other language and protocol needed for web applications. Selenium comprises one among the most important communities and support networks in automation testing. Even tests that aren’t designed initially on Selenium will often draw upon this framework for a minimum of some elements.

Citrus is an automatic testing tool with messaging protocols and data formats. Various protocols like HTTP, REST,JMS, and SOAP can be tested within Citrus, outside of broader scope, functional automated testing tools like Selenium. Citrus will identify whether the program is appropriately dispatching communications and whether the results are needless to say . It also can be integrated with Selenium if another front-end functionality testing has got to be automated. Thus, this is often a selected tool that's designed to automate and repeat tests which will validate exchanged messages.Citrus appeals to those that prefer tried and true. Citrus is designed to test messaging protocol. It contains support for HTTP, REST, SOAP, and JMS.When applications got to communicate across platforms or protocols, there isn’t a more robust choice. It works with other frameworks (like Selenium) and streamlines tests that compare user interfaces with back-end processes (such as verifying that the send button works when clicked). This enables a rise in the number of checks during a single test and an increase in test confidence.

Gauge is produced by an equivalent company that developed Selenium. With Gauge,developers can use C#, Ruby, or Java to make automated tests Gauge itself is an extensible program that has plug-in support, but it's still in beta; use this as long as you would like to adopt cutting-edge technology now. Gauge may be a promising product, and when it's complete will likely become a typical , both for developers and testers, because it has quite lot of technology behind it. Gauge aims to be a universal testing framework.

Commercial Automation Tools

IBM Rational Functional Tester A data-driven performance testing tool, IBM may be a commercial solution that operates in Java, .Net, AJAX, and more. The IBM Rational Functional Tester provides unique functionality within the sort of its “Storyboard” feature, whereby user actions are often captured then visualized through application screen-shots. IBM RFT will give a corporation information about how users are using their product, additionally to how users are potentially breaking their product. RFT is integrated with life cycle management systems, including the Rational Quality Manager and the Rational Team Concert. Consequently, it’s best utilized in a strong IBM environment.

Katalon Studio may be a unique tool that's designed to be run both by automation testers and programmers and developers. There are different levels of testing skill set available, and therefore the testing processes include the power to automate tests across mobile applications, web services, and web applications. Katalon Studio is made on top of Appium and Selenium, and consequently offers much of the functionality of those solutions.Katalon Studio is a superb choice for larger development teams that may require multiple levels of testing. It are often integrated into other QA testing processes like JIRA, Jenkins, qTest, and Git, and its internal analytic system tracks Dev-Ops metrics, graphs, and charts.

Ranorex may be a commercial automation tool designed for desktop and mobile testing. It also works well for web-based software testing. Ranorex has the benefits of a relatively low pricing scale and Selenium integration. When it involves tools, it's reusable test scripts, recording and playback, and GUI recognition. It’s a sufficient all-around tool, especially for developers who are wanting to test on both web and mobile apps. It boasts that it's an “all in one” solution, and there's a free trial available for teams that want to check it.

Sahi Pro is out there in both open source and commercial versions, Sahi is centered around web-based application testing. Sahi is employed inside a browser and may record testing processes that are done against web-based applications. The browser creates an easy-to-use environment during which elements of the appliance are often selected and tested, and tests are often recorded and repeated for automation. Playback functionality further makes it easy to pare to errors. Sahi may be a well-constructed testing tool for smaller parts of an application. Still, it's going to not be feasible to use for more wide-scale automated test production, because it relies totally on recording and playback. Recording and playback is usually an inconsistent method of product testing.

TestComplete is both keyword-driven and data-driven, TestComplete may be a well-designed and highly functional commercial automated testing tool. TestComplete are often used for mobile, desktop, and web software testing, and offers some advanced features like the power to acknowledge objects, detect and update UI objects, and record and playback tasks. TestComplete can be integrated with Jenkins.

TestPlant eggPlant may be a niche tool that's designed to model the user's POV and activity instead of simply scripting their actions. Testers can interact with the testing product because the end users would, making it easier for testers who might not have a development or programming background. TestPlant eggPlant are often wont to create test cases and scenarios with none programming and may be integrated into lab management and CI solutions.

Tricentis Tosca may be a model-based test automation solution, Tricentis Tosca offers analytics, dashboards, and multiple integrations that are intended to support agile test automation. Tricentis Tosca are often used for distributed execution, risk analysis, integrated project management, and may support applications, including mobile, web, and API.

Cypress is meant for developers, Cypress is an end-to-end solution "for anything that runs inside the browser." By running inside the browser itself, Cypress can provide for more consistent results in comparison to other products like Selenium. As Cypress runs, it can alert developers of the actions that are being taken within the browser, giving them more information regarding the behaviors of their applications. Debuggers are often quickly introduced on to applications to streamline the event process.

Appium has one purpose: testing mobile apps. That doesn't mean to imply that it's a limited range of testing options. It works natively with iOS, Android, and other mobile operating systems. It supports simulators and emulators, and it's a darling for test designers who alsoare app developers. Perhaps the foremost notable perk of Appium is that it enables testing environments that don't require any changes to the first app code. That means apps are tested in their ready-to-ship state and produces test results that are as reliable as possible.

Apache JMeter is made for load testing. It works with static and dynamic resources, and these tests are critical to all or any web applications.It can simulate loads on servers, server groups, objects, and networks to make sure integrity on every level of the network. Like Citrus, it works across communication protocols and platforms for a universal check out communication. Unlike Citrus, it's emphasis isn't on basic functionality but in assessing high-stress activity. A popular function among testers is JMeter's ability to perform offline tests and replay test results. It enables much more scrutiny without keeping servers and networks busy during heavy traffic hours.

RedwoodHQ is an Open Source test automation framework. It uses a web-based interface that's designed to run tests on an application with multiple testers. Tests are often scripted in C#, Python, or Java/Groovy, and web- based applications are often tested through APIs, Selenium, and their web IDE. Creating test scripts are often completed on a drag-and-drop basis, and keyword- friendly searches make it easier for testers to develop their test cases and actions.Though it's going to not be suitable for more in-depth testing,

Serenity BDD (also referred to as Thucydides) may be a Java-based framework that's designed to require advantage of behavior-driven development tools. Compatible with JBehave and Cucumber, Serenity makes it easier to make acceptance and regression testing. Serenity works on top of behavior-driven development tools and therefore the Selenium WebDriver, essentially creating a secure access framework which will be wont to create robust and sophisticated products.

V. TABULAR COMPARISON OF THE DEPENDENCIES OF SOFTWARE AUTOMATION TESTING TOOLS

Tool/ Criteria	Open Source	License	Supporting Platform	Supporting Platform	Supporting Platform	Learning Ease/	Coding/	Code Reusability	Test Results Report	Record & Playback
Selenium	G			G		G	G	G	Plugin	G
TestComplete		G	G	G	G	G	x		G	G
Ranorex		G	G	G	G			G		G
Appium	G		G					G		G
Quick Test Professional		G		G	G		x	G	G	
OpenScript		G		G	G	x		G	G	G
Janova		G		Cloud based			x		G	G
Rational Functional Tester	G			G		G				

Table 1 Tabular representation of the dependencies of the software automation testing tools

Table 1 gives a tabular representation of the analysis of the software automation testing tools considering various criteria like open source availability, Licensing,platform supported, Coding skills needed And various other criterias.

Tool	Testing Type
Selenium	Functional testing
TestComplete	Functional testing, Graphical User Interface testing, Unit testing
Ranorex	Graphical User Interface testing, Compatibility testing
Appium	Graphical User Interface testing, Functional testing

QuickTest Professional	Functional testing, Regression testing
OpenScript	Functional testing, Load testing, Database testing
Janova	Functional testing
Rational Functional Tester	Functional testing, Regression testing, Graphical User Interface testing

Table 2 Tabular representation of the different testing types supported by the tools

Table 2 gives a tabular representation of the different types of testing that are supported by the different software testing tools. The different types of testing can be functional testing, regression testing, graphical user interface testing, load testing and so on.

VI. CONCLUSION

Software testing is a significant part of the life cycle of software development. It guarantees that the software been deployed to the market is free from error and effects. To meet up with market demands and time factor, software testers has employed the use of automated testing tool to carry out testing, this is used over the manual approach to testing, as automated testing saves time and also minimize cost incurred in the organization during the testing phase.

This paper evaluates some automated testing tools which includes selenium, testcomplete, Ranorex, OpenScript, Janova, etc, highlighting their basic features and characteristics. All testing tools are efficient to be used for testing but depending on the scenario, some may tend to be more efficient than the others, this paper highlights the details of some selected testing tool and their testing types in tabular format, their licensing, the platform they best operate, the tools that support code reusability among others.

This study would be useful to the industry expertise to know which tool to best use for a particular project and for researchers, more tools can be compared and more criteria for comparison highlighted, this would avail software testers the ability to choose perfect tools for testing applications with ease, this would save more time and minimize cost when the right testing tool is selected. It can therefore be concluded that there is no one perfect tool for testing, but for a particular testing purpose, tradeoffs can be made to select the best tool depending on the size of the project, the budgeted cost for testing and the platform of the application.

VII. REFERENCES

[1] Pressman, R. S. (n.d.). Software engineerPressman, R. S. (n.d.). Software engineering (2nded.). New York: McGraw-Hill Book Company.ring.

[2] Chauhan, R. K. & Singh, I. (2014). Latest Research and Development on Software Testing Techniques and Tools, 4(4), 2368–2372.

[3] Wala, T. & Sharma, A. K. (2014). Improvised Software Testing Tool, 3(9), 573–581.

[4] Merina, C. (2019). Tool Usability Parameter in Determining the Performance of Software Testing Tool, IJTB (International J. Technol. Business), 3(1), 8–18.

[5] Testing in Global Software Development Environment, Data Science and Analytics, 309-317, DOI: 10.1007/978-981-10-8527-7_25.

[6] Tassey, G. (2020). "The Economic Impacts of Inadequate infrastrure for software testing". Natl. Inst. Stand. Technol.

[7] Raulamo-Jurvanen, P., Mäntylä, M. & Garousi, V. (2017). Choosing the Right Test Automation Tool, in 21st International Conference on Evaluation and Assessment in Software Engineering (EASE 2017), 21–30.

[8] Jain, V. & Rajnish, K. (2018). Comparative Study of Software Automation Testing Tools: OpenScript and Selenium, Int. J. Eng. Res. Appl., 8(2), 29–33.

[9] Odun-Ayo, I., Falade, A., & Samuel, V. (2018). Cloud Computing and Open Source Software: Issues and Developments, Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2018, 14-16 March, 2018, Hong Kong,140-145.

[10] Uppal, N. & Chopra, V., (2012). Design and Implementation in Selenium IDE with Web, Int. J. Comput. Appl., 46(12), 8–11.

[11] A.Martin-Lopez, S.Segura and A. Ruiz-Cortés, "Inter-Parameter Dependencies in RESTful APIs [Dataset]", 2019.

[12] M. Bozkurt, M. Harman and Y. Hassoun, "Testing and Verification in Service- Oriented Architecture: A Survey", Software Testing Verification and Reliability, vol. 23, no. 4, pp. 261-313, 2017.

[13] H. Ed-douibi, J.L.C. Izquierdo and J. Cabot, "Automatic Generation of Test Cases for REST APIs: A Specification-Based Approach", IEEE 22nd Intern. Enterprise Distributed Object Computing Conference, pp. 181-190, 2018.

[14] T. Fertig and P. Braun, "Model-Driven Testing of RESTful APIs", Intern. Conferenceon World Wide Web, pp. 1497-1502, 2015.

[15] S. Yatskiv, I. Voytyuk, N. Yatskiv, O. Kushnir, Y. Trufanova and V. Panasyuk, "Improved Method of Software Automation Testing Based on the Robotic Process Automation