PROJECT SEMESTER REPORT

# AUTOMATED TESTING OF VOUCHER SERVER

By

## Karandeep Parashar

**Roll No.** 101512029

Under the Guidance of

## Shamim K Khan

Product Owner

## Jasmeet Singh

## 

Lecturer

Submitted to the

## Computer Science & Engineering Department Thapar Institute Of Engineering And Technology, Patiala

In Partial Fulfillment of the Requirements for the Degree of Bachelor of Engineering in Software Engineering

Thapar Institute Of Engineering And Technology, Patiala

## May 2019

**AUTOMATED TESTING OF VOUCHER SERVER**

by: Karandeep

Place of work: Ericsson India Global Services Pvt Ltd, Gurugram, Haryana

Submitted to the Computer Science & Engineering Department, Thapar Institute Of Engineering And Technology

May 2019

In Partial Fulfillment of the Requirements for the Degree of Bachelor of Engineering in Software Engineering.

# ABSTRACT

The objective of the project is to automate the process of testing of the New Generation Voucher Server, which is one of the important nodes of the Ericsson Charging System. The project aims to reduce human effort and time, maintaining the efficiency. The section of testing which is being targeted is robustness and recoverability. The automatable test cases were identified from all the test cases which were under the category of manual execution.

The project verifies the following scenarios:

* Whether the product(VS) can handle unforeseen or foreseen errors gracefully.
* Is it possible to use the product after the fatal error?
* Will the product recover and perform well after the error?

Author : Karandeep Parashar

Certified by: Mr Ashish Ranjan (R&D Manager)

Certified by: Mr. Jasmeet Singh(Faculty Mentor)

Accepted by: Dr. Vinay Arora(Project Semester Coordiator,CSED)

# CERTIFICATE FROM THE COMPANY



**ACKNOWLEDGEMENT**

I would like to express my thanks to Mr. Ashish Ranjan, R&D Manager, for giving me this opportunity to do such an important project. He has been of great help in my venture, and. He is truly an amazing mentor to have.

I am also thankful to Shamim K Khan, my mentor, Team Head, NGVS, for an indispensable resource of technical knowledge and helping me in all possible ways towards successful completion of this project.

I thank all those who have contributed either directly or indirectly towards this project. Lastly, I would also like to thank my family for their unyielding love and encouragement.

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| **ABSTRACT** | 2 |
| **CERTIFICATE**  **ACKNOWLEDGEMENT** | 3  4 |
| **TABLE OF CONTENTS** | 5 |
| **LIST OF FIGURES** | 7 |
| **LIST OF ABBREVIATIONS** | 8 |
| **COMPANY PROFILE**  **1. INTRODUCTION** | 9 |
| 1.1 Project Overview | 11 |
| 1.2 Need Analysis | 17 |
| 1.3 Problem Definition and Scope of the Project | 18 |
| 1.4 Approved Objectives | 19 |
| 1.5 Methodology Used | 19 |
| 1.6 Summary of Project Outcomes | 19 |
| **2. METHODOLOGY ADOPTED** | 21 |
| 2.1 Investigative Techniques | 21 |
| 2.2 Proposed Solution | 23 |
| 2.3 Tools and Technologies Used | 23 |

|  |  |
| --- | --- |
| **3. FINDINGS AND OBSERVATIONS** | 25 |
| **4.LIMITATIONS** | 26 |
| **5.CONCLUTIONS AND FUTURE WORK** | 27 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** |  | **Page No.** |
| Figure 1 | Rating in CS | 1 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **VS** | Voucher Server |
| **CS** | Charging System |
| **OS** | Operating System |
| **NGVS** | Next Generation Voucher Server |

**COMPANY PROFILE**

Ericsson is a Swedish multinational networking and telecommunications company headquartered in Stockholm. The company offers services, software and infrastructure in information and communications technology for telecommunications operators, traditional telecommunications, and Internet Protocol (IP) networking equipment, mobile and fixed broadband, operations and business support services, cable television, IPTV, video systems, and an extensive services operation.

Ericsson has been associated with the Indian telecom industry for over 110 years. The company’s partnership with India began in 1903, when we supplied manual switchboards to the Government of India. Since then, it has powered virtually every facet of telecommunications in India, across Mobile broadband, Managed Services and more recently in new areas like Media, IT, and Industries. It has successfully collaborated the growth of the country's cellular revolution since 1994, when cellular services were first launched in India.

Ericsson introduced the world's first fully automatic mobile telephone system, MTA, in 1956. It released one of the world's first hands-free speaker telephones in the 1960s. In 1954, it released the Ericofon. Ericsson crossbar switching equipment was used in telephone administrations in many countries.

Ericsson Global Services, established in India in 2010, has been set up to manage operator networks across geographies. This strengthens the company’s existing worldwide capabilities in Consulting & Systems Integration Services, Managed Services, IP & Broadband R&D and Revenue Management R&D supporting local, regional, and global customers.

The offices of Ericsson in India are present in Pune, Gurgaon, Bangalore, and Chennai. Ericsson's R&D facilities in Gurgaon, Chennai, and Bengaluru focus on value-added applications for the pre-paid, multimedia segments. The Bengaluru R&D Center also works on Ericsson's SmartEdge (SE) and Smart Services Router (SSR) and Software Defined Networks (SDN) product lines.

Ericsson Global Private Limited Gurgaon, works on improving the scale and level of communication and management of the finances related to the cost than occur when we make

calls as well use internet on our phones. It is a large scale company with its offices in many countries with the main head office in Sweden.

The project -NGVS which I was part of comes under the Ericsson Charging System, which is explained in the next section.

**INTRODUCTION**

## Project Overview

The project, Automated Testing Of Voucher Server comes under the Ericsson Charging System, which is one of the most important projects of Ericsson.VS being one of the important

node of CS.

### **ERICSSON CHARGING SYSTEM**

Ericsson Charging System is a modular, scalable, open, single convergent Online Charging System(OCS) using industry standards and protocols.

Comprising all the data charging and yield management capabilities previously available only in separate offerings, CS with mobile broadband charging gives the operator an invaluable competitive advantage in the new Networked Society environment.

It is a real-time heart of BSS: a scalable, flexible solution with a strong user-experience focus and future-proof product roadmap, enabling real-time convergent charging, policy control, decoupling and fast service creation. It lets the operator control credit while letting users control their costs through flexible packaging, bonuses, and discounts. Whatever the service, whatever the network, whoever the customer, CS lets the operator realize and create new opportunities.

Capitalizing on today’s tremendous opportunities can be challenging. Customers are increasingly demanding and, at the same time, many new players are entering the marketplace. To compete in this environment, operators must not only maintain but also grow their part of the new value chain to gain loyal customers. This requires true service agility – the ability to provide a constant stream of new, innovative services, quickly developed and executed to perfection.

Comprising all the data charging and yield management capabilities previously available only in separate offerings, Ericsson Charging System with mobile broadband charging gives an invaluable competitive advantage in this environment. It is the real-time heart of all

BSS: a scalable, flexible solution with a strong user-experience focus and future-proof product roadmap, enabling real-time convergent charging, policy control, decoupling and fast service creation. It lets you control your credit while letting users control their costs through flexible packaging, bonuses, and discounts. Whatever the service, whatever the network, whoever the customer, Ericsson CS with mobile broadband charging let’s one realize and create new opportunities.

### Real Time Charging

To understand the real-time charging system, the process behind it must be known.

* + 1. The account is initiated, the rates and rules for the account are set up and the SIM card is distributed to a reseller.
    2. The subscriber buys the account at the reseller and activates the account using a voice response service.
    3. The subscriber makes a phone call and the telecom system makes a request to the charging system.
    4. The charging system checks if there is money in the account and the applicable rates at this particular time.
    5. Our subscriber has enough money in the account. The Charging system reserves money for normally four minutes, this is configurable, and sends an OK to the telecom system to set up the connection. This means that the phone call will not be set up until the money for it is reserved in the account. If there is not enough money, the Charging System will send a message to the telecom system not to set up the call.
    6. The Charging system checks the money in the account and the rates again. Note that the rates may change during a call, for example if it continues into a low-fee time period. The charging system reserves money for another four minutes and sends an OK message back to the telecom system.
    7. The phone call ends.
    8. When the call has been disconnected, the telecom system informs the Charging system of the actual time used.
    9. The Charging system settles the account and returns the reserved money that was not used.
    10. The Charging system then sends a message about the cost of the call and the new account balance via the telecom system to the subscriber’s phone.

This was a brief overview of how a real-time charged phone call is set up. The Charging system can handle real-time charging, as in our example, and offline charging. The telecom system and the charging system are tightly integrated, since real-time charging is part of the telecom service delivery.

### Vouchers

A voucher is a credit note that is used for refilling an account. Vouchers in the Charging System are of two types, standard vouchers, and value vouchers.

A standard voucher refill will add money to the account and extend the service fee and supervision periods. The monetary value of the voucher is stored in the activation code on the voucher.

A value voucher refill will initiate a service class change that allows the subscriber to take advantage of, for example, lower tariffs or GPRS. The change can be time-limited or permanent.

Via the premium refill feature it is possible to do a refill combining a standard voucher refill and a value voucher. The activation of a voucher is done through IVR (Interactive Voice Response), USSD message, or the Customer care service.

### Rating

Rating is the process of computing a price for a session or event. Call and account data are used as input in tariff tree structures when computing the rate. A tariff consists of rates – a rate per unit (time, volume, or units), and fees – a fixed price for certain events.

The call and account related data considered in the rating process are data like volume, duration, account balance, etc., and the context in which the event takes place – for example day of week and time of day.

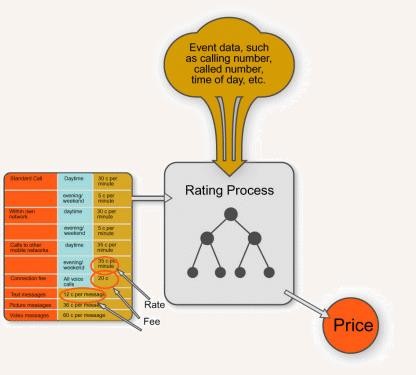


Figure 1: Rating in CS

### Network Elements of Charging System

Here, is the explanation of all active network elements and explanations on how they interact and create the different parts of the Charging System.

The network elements can be grouped into the following:

* + - * Charging Access, Account and Rating part
      * User Interaction and Refill part
      * Provisioning and Administration part
      * Reporting part
      * Upgrade part

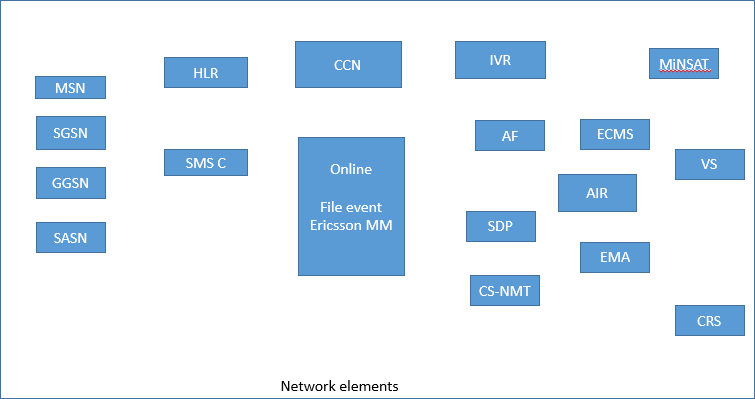


Figure 2: Network Elements of CS

* **MSC**(Mobile services Switching Centre)

The MSC handles all Circuit Switched accesses in the telecom system.

* **SGSN** (Serving GPRS Support Node)

Handles all packet switched data in the telecom system.

* **GGSN**(Gateway GPRS Support Node)

Handles all packet switched data in the telecom system.

* **SASN** (Service Aware Support Node)

The SASN is a network element that performs deep packet inspection. It interacts in real- time with Charging System and provides wireless operators with service awareness and service differentiated charging to their data networks.

* **HLR** (Home Location Register)

Handles account information in the telecom network.

The HLR is used by the charging system for triggering IN/CAMEL services and barring of services in the telecom system. It is also used for the Unstructured Supplementary Service Data (USSD) communication.

* **SMS-C** (Short Message Service Centre)

The SMS-C is responsible for the relaying and storing and forwarding of short messages.

* **CCN** (Charging Control Node)

Responsible for charging control of circuit switched calls and data, GPRS, SMS and other service application requiring session or event based charging.

* **Ericsson MM** (Ericsson Multi Mediation)

Handles mediation of Data Records and provides them to other network elements for further processing. Also acts as an integration point in Charging System for service and content charging.

* **IVR** (Interactive Voice Response)

The IVR enables self-care administration and refill services by guiding the subscriber with voice prompts and receiving Dual Tone Multi frequency tones (DTMF) in response. The IVR interacts with AIR to be able to provide this service.

* **MINSAT** (Mobile IN Service Administration Tool)

Responsible for subscriber provisioning and customer care interaction

* **ECMS** (Ericsson Customer Management System)

Responsible for subscriber provisioning and customer care interaction. ECMS is an alternative to MINSAT.

* **AIR** (Account Information and Refill server)

The AIR is responsible for the refill function, including processing of different refill rules depending on used voucher and requesting subscriber, and handling of refill promotions and division towards the subscriber accounts. It is also responsible for balance enquiries. These operations are offered over UCIP (which the IVR can use) or USSD communication. AIR also supports a number of file based batch-jobs for making bulk adjustments, promotions and refills.

* **AF** (Account Finder)

Provides the system with routing information in the form of an SDP identifier. The routing is performed by translating MSISDN into SDP identifier.

* **SDP** (Service Data Point)

The SDP contains the database of subscriber and accounts information, including balances and life-cycle information. It is responsible for rating of sessions and events, post-processing of CDRs and delivery of USSD- and SMS-notifications. As a part of this

the SDP provides the evaluation logic and the reserve/deduct mechanism required for the call-control performed by the CCN.

* **VS**(Voucher Server)

The VS stores vouchers, including handling of voucher state, and provides the AIR with voucher information during refill. The VS also supports voucher generation, import/export, enquiries, purging, reporting and individual or batch based voucher state changes.

* **EMA** (Ericsson Multi Activation)

The EMA provides central provisioning and activation for Charging System. EMA also handles barring/unbarring towards HLR when redundant HLRs are used.

* **CS-NMT** (Charging System - Network Management Toolkit)

Hosts the System Upgrade Framework (SUF) and can act as an archive server for CCN.

* **CRS** (Charging data Reporting System)

Collects data records from CCN, SDP and AIR and provides usage and account history and financial reports.

## Need Analysis

Software Development Life Cycle (SDLC) aims to produce a high-quality system that meets or exceeds customer expectations, works effectively and efficiently in the current and planned information technology infrastructure, and is inexpensive to maintain and cost-effective to enhance.

A process followed in software projects is SDLC. Each phase of SDLC produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design. Testing should be done on a developed product based on requirement. Testing being one of the most important phase of SDLC, this phase should be completed accurately and efficiently.

[Software tests](https://smartbear.com/products/testing/) should be repeated often during development cycles to ensure quality. Every time source code is modified software tests should be repeated. For each release of the software it may be tested on all supported operating systems and hardware configurations. Manually repeating these tests is costly and time consuming. Once created, automated tests

can be run repeatedly at no additional cost and they are much faster than manual tests. Automated software testing can reduce the time to run repetitive tests from days to hours. A time savings that translates directly into cost savings.

Automated software testing can increase the depth and scope of tests to help improve software quality. Lengthy tests that are often avoided during manual testing can be run unattended. They can even be [run on multiple computers](https://smartbear.com/product/testcomplete/features/distributed-testing/) with different configurations. Automated software testing can look inside an application and see memory contents, data tables, file contents, and internal program states to determine if the product is behaving as expected. Test automation can easily execute thousands of different complex test cases during every test run providing coverage that is impossible with manual tests.

Even the most conscientious tester will make mistakes during monotonous manual testing. Automated tests perform the same steps precisely every time they are executed and never forget to record detailed results. Testers freed from repetitive manual tests have more time to create new automated software tests and deal with complex features.

Shared automated tests can be used by developers to catch problems quickly before sending to QA. Tests can run automatically whenever source code changes are checked in and notify the team or the developer if they fail. Features like these save developers time and increase their confidence.

## Problem Definition and Scope of the Project

**Goal:** Develop an automated system for testing of VS. The main aim being reducing the human effort and time while maintaining the efficiency and accuracy.

**Problem Description**: There was a need to develop a system which could automatically generate results of the test cases which a were under the category of manual execution. Firstly, the test cases which had a scope of being automated had to be identified. The main area which was considered was robustness and recoverability. To carry forward this there was need to invoke every possible system and services failure, so that VS could be tested on every foreseen or unforeseen circumstances.

The system must be able to verify the following

* Whether the product(VS) can handle unforeseen or foreseen errors gracefully.
* Is it possible to use the product after the fatal error?
* Will the product recover and perform well after the error?
* Even if once the problem occurs it does not occur ever again in any upgraded version.

## Approved Objectives

The objectives of the project include:

1. Identify the automatable test cases i.e Jives.
2. Automate the execution of the test cases identified.
3. Analyze the effect of new test cases on older ones.
4. Find an optimized way to execute of all the test cases in the minimal time.
5. To integrate the test cases with the existing system.

## Methodology Used

The methodology used for the project is:

1. **Identification:** This phase involved the identification of the test case which were feasible enough to be automated.
2. **Requirement analysis:** In this phase the requirement of the project was gathered and studied.
3. **Planning:** The planning was necessary as to find the most optimal way to reduce the time of execution of the test cases.
4. **Execution:** This phase included the execution of the test cases. The areas of testing which were to be automated included robustness, recoverability, sanity testing.

So for this, every possible situation had to be replicated to test how the system behaves in any foreseen or unforeseen circumstances.

## Summary of Project Outcomes

The project should be able to perform the following operations:

1. Identify the test cases which are feasible enough to be automated.
2. Identify the scenarios where there has been a TR raised in the passed but could have been handled before if a Jive was already written.
3. Automate the process of testing of all the test cases identified in the above phase.
4. Analyze that the new test cases do not effect the older ones.
5. Optimized way of testing the system in minimal time.
6. Output the results of the test cases automated in the user friendly format.

# METHODOLOGY ADOPTED

## Investigative Techniques

For the completion of the project in the most efficient way a proper planning and research was needed. Firstly, the study of Jives framework was necessary so that it would be clear how already existing integration testing works. Jives(Integration test cases) were only possible of the methods that were exposed. The NGVS used three ways to expose its services to customers:

* + 1. Graphical User Interface(GUI)
    2. Command Line(CLI)
    3. VSIP requests

Even though methods were exposed using different means still the implementation was same. In order to automize the test cases different scenarios had to be identified. Different approaches were taken to identify these. The whole unit of NGVS was working to identify and assign the test cases to be written.

Following approaches were used to identify the new scenarios:

## TRs

Along with the development of new features the NGVS unit also works towards the maintenance of the existing product. If a customer faces a problem while using the product and it is verified as a legit problem it is assigned to a member of the NGVS unit member to be resolved. These TRs often occurs when the NGVS product is being pushed to its limit and a new scenario comes to light. These scenarios if are already identified then Jives which run on a daily basis would fail on that particular scenario and force the unit to update the test case or code accordingly to handle it. These TRs that are being raised constantly were a source of cases where Jives have not been written. Every new TR was leading to a new Jive. In order to write that jive a team member was also assigned to alter the code accordingly.

## Exposed Methods

The only scenarios that needed to be tested are the one that could occur while the customer uses the product. Customer is only able to access the methods that being exposed to them. These exposed methods could be identified by understanding the code of the product. The services that NGVS provides is exposed to customers using the following three ways:

1. GUI
2. CLI
3. VSIP

The GUI is the graphical user interface that is available to customers and administrators to use. The functionalities that are exposed here are limited but are easy to access.

The CLI works according a issued document that is distributed as the register to use the product of Ericsson. The methods that are exposed using CLI have more properties that can be set.

VSIP requests can be understood as similar to HTML requests. These requests hit server to do the action.

Any of the above ways can be used to perform a task. These exposed methods should have a test case that verifies that they behave as expected.

## Experience

People working at Ericsson for a long time have faced a problem at some point. Every one was contributing in this . According to the understanding of every member they were proposing a scenario that could be written and tested that would help the NGVS from future problems.

## Proposed Solution

For carrying out the process of automation of test cases for robustness and recoverability the steps which had to be performed are as follows:

* There are different stages in Voucher Server, and to perform the testing in these stages firstly we must schedule tasks to reach that stage as explained below:

Let us consider that we have 5 stages namely A, B, C, D in the Voucher Server (The names have been assumed to keep the confidentiality). Now to reach any stage we need to firstly complete all the steps involved in all the stages that are present before that stage.

* And after reaching the required stage, we need to initiate the errors which could be faced by the product VS.

The errors are could be of any form for example it could be any service failure which is being used in VS or it could involve restarting of the service.

* After performing the previous step, we need to verify the result of how did the voucher server reacted to that error.

This step involves validating the log files and checking whether the task which was scheduled was completed successfully or not

## Tools and Technologies Used

### **JAVA**

Java is a general-purpose [computer-programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-](https://en.wikipedia.org/wiki/Class-based_programming) [based,](https://en.wikipedia.org/wiki/Class-based_programming) [object-oriented,](https://en.wikipedia.org/wiki/Object-oriented_programming) and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run](https://en.wikipedia.org/wiki/Write_once%2C_run_anywhere) [anywhere](https://en.wikipedia.org/wiki/Write_once%2C_run_anywhere)" (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture.](https://en.wikipedia.org/wiki/Computer_architecture) As of 2016, Java is one of the most [popular programming](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) [languages in use,](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) particularly for client-server web applications, with a reported 9 million developers.

### **BT** **Tools**

It is software that is used to generate a request and hit the server. The response would help in understanding the current behavior of the product. Different requests were being written and used using BT tools.

### **Apache Cassandra**

Apache Cassandra is a database that is being used in distributive computing. It solves various problems that occurred in traditional databases like availability and single point of failure. Cassandra can be customized according to the ones need and be used for a large scale data too.

### **J UNIT**

JUnit is a unit testing framework for Java programming language. It plays a crucial role test-driven development, and is a family of unit testing frameworks collectively known as x-Unit.

JUnit promotes the idea of first testing then coding, which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.

The process of third party dependency extraction involves coding the architecture of the program, its execution and testing in various scenarios that are possible during the time of deployment. This main part of architectural development is the process of extraction of the dependencies from the code which is done by a process.

# OBSERVATIONS AND FINDINGS

There have been a lot of observations while working in this project. This project gave an opportunity to learn and explore in different areas.

Highlights of the observations made through this project is as follows

* Working in the Ericsson CS , I observed how all the phases in the CS work and how different nodes interact with each other.
* Working for NGVS , I observed how the voucher server works and makes it possible.
* Working on Cassandra, helped me understand how the big scale software’s are different from our small projects.
* Working on Apache Zookeeper helped me understand how nodes are configured to be set in a big scale project.
* I explored a lot of new things about JAVA and Maven central. I learnt about some of the new concepts about JAVA .

# LIMITATIONS

The limitations in the project I have been working on were the areas where complete automation was not possible and there was a need of human intervention for the completion of that test case

The main task during the project implementation was to firstly being able to identify the test cases which could be automated and were currently under manual execution.

Now, the test cases which were identified , had to executed in such a way that it should be able to generate the results without any human intervention once the test suit is started.

There were cases in which the human intervention was required in between of the execution of the test cases and hence these set of test cases didn’t have a scope of being automated and thus these were not automated.

# CONCLUTIONS AND FUTURE WORK

Through this project I have learnt quite a new languages and have enhanced my skills.

* The project gave me a chance to enhance my analytical skills.
* I got a chance to get hands on experience in Apache Cassandra, Zookeeper, Spark.
* I got to enhance my JAVA skills.
* I learnt to work with a team and as a team.

In the Future, the project can be taken further by

* Automation goal that was given to me is completed but it could be expanded and be covered to complete the automation goal of the whole unit.
* After the automation of integration testing, there are various scenarios were human effort can be saved that could also be automated.