

**A  
Project Report  
On  
"Handset Automation"**

(IT450 – Software Group Project)

**Prepared by**  
DhruvKumar Nagar (19IT070)

**Under the Supervision of**  
Prof. Hemant Yadav

**Submitted to**  
Charotar University of Science & Technology (CHARUSAT)  
for the Partial Fulfillment of the Requirements for the  
Degree of Bachelor of Technology (B.Tech.)  
in Smt. Kundanben Dinsha Patel Department of Information  
Technology (IT)  
for B.Tech Semester 7

**Submitted at**



**Accredited with Grade A by NAAC**



**Smt. Kundanben Dinsha Patel Department of Information Technology  
Chandubhai S. Patel Institute of Technology (CSPIT)  
Faculty of Technology & Engineering (FTE), CHARUSAT  
At: Changa, Dist: Anand, Pin: 388421.  
April, 2022**

## **DECLARATION BY THE CANDIDATE**

I hereby declare that the project report entitled “**Handset Automation**” submitted by me to Chandubhai S. Patel Institute of Technology, Changa in partial fulfilment of the requirements for the award of the degree of **B.Tech Information Technology**, Smt. Kundanben Dinsha Patel Department of Information Technology, CSPIT, FTE, is a record of bonafide IT450 Software Group Project (project work) carried out by me under the guidance of **Prof. Hemant Yadav**. I further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

**(DhruvKumar Nagar -19IT070)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Prof Hemant Yadav**

Associate Professor

Smt. Kundanben Dinsha Patel Department of

Information Technology, Chandubhai S Patel Institute

of Technology (CSPIT) Faculty of Technology (FTE)

Charotar University of Science and Technology (CHARUSAT) - Changa.



**MOTOROLA SOLUTIONS**

November 10, 2022

**INTERNSHIP CERTIFICATE**

This is to certify that **Mr. DhruvKumar Nagar** is working with Kodiak (a Motorola Solutions Company) located in Vadodara as an Intern from August 29, 2022 to May 31, 2023.

DhruvKumar Nagar was extremely eager to learn and take on new challenges. We always appreciated his sincere work, dedication, and quest for professional excellence.

We wish him all the best in future endeavors.

**For Motorola Solutions Kodiak Networks (I) Pvt Ltd.,**

A handwritten signature in black ink, appearing to read 'Disha', followed by a horizontal line.

**Disha Thomas**  
**Human Resources**

**Motorola Solutions Kodiak Networks India Private Limited**

9th Floor | MFAR Manyata Tech Park | Greenheart Phase IV | Nagawara | Bengaluru - 560045 INDIA  
P : +91 80 4000 5555 | [www.motorolasolutions.com](http://www.motorolasolutions.com) | CIN-U72900KA2003PTC031841



# CHARUSAT

CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A by NAAC

## CERTIFICATE

This is to certify that the report entitled “**Handset Automation**” is a bonafied work carried out by **DhruvKumar Nagar (19IT070)** under the guidance and supervision of **Prof. Hemant Yadav & Mr. Hardik Kothari** for the subject **Software Group Project (IT450)** of 7<sup>th</sup> Semester of Bachelor of Technology in **Information Technology** at Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of the candidate himself, has duly been completed and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of the content, presentation, and language for being referred by the examiner(s).

Under the supervision of,

**Prof. Hemant Yadav**

Associate Professor  
Smt. Kundanben Dinsha Patel Department of  
Information Technology, CSPIT, FTE,  
CHARUSAT, Changa, Gujarat

**Mr. Hardik Kothari**

Senior Manager  
Handset Integration  
Motorola Solutions

**Dr. Parth Shah**

Head - Smt. Kundanben Dinsha Patel Department of Information  
Technology, CSPIT, FTE, CHARUSAT, Changa, Gujarat.

---

**Chandubhai S. Patel Institute of Technology (CSPIT)**

**Faculty of Technology & Engineering (FTE), CHARUSAT**

At: Changa, Ta. Petlad, Dist. Anand, Pin: 388421. Gujarat.

## **ABSTRACT**

Software testing is an important process in product development to ensure it fulfills the client's needs as It helps in identifying any bugs or errors in the software. Thoroughly tested software ensures reliable and high-quality performance software operation. Testing can contribute to the success of an overall project in a way. The proposed tool is built to reduce the tedious task of manually testing and it helps to automate the feature testing using Automation scripts. This tool allows users to execute scripts for the feature of their choice which helps to understand the working of particular features in different environments. This tool generates various logs (Client Logs, UI Logs, ADB Logs, TRX Logs) which help users analyze via logs where the functionality is broken and where to provide an appropriate fix. Apart from that, Tool also helps to test the stability of features in different Lab Environments. With the use of this tool, it promises better business optimization (less maintenance cost), reliability, and superior user experience. Substantially, the iterative efforts spent molding a powerful yet error-free software are far and wide.

## ACKNOWLEDGMENT

During this project work, I have experienced developing Automation Scripts, I gathered plenty of technical knowledge, and the whole process was not that easy, but many people helped us achieve this milestone that we fixed at the beginning of the semester.

First, I want to thank our external guide, **Mr. Hardik Kothari**, for having faith in me and providing a very effective environment to work seamlessly. It is because of him that I had been motivated to work. I am also grateful to **Prof. Hemant Yadav** for his exceptional guidance and constant supervision during the project work as an Internal Guide.

I would like to express my gratitude to H.O.D of Information Technology **Dr. Parth Shah** and I am also grateful to others faculty members of Chandubhai S. Patel Institute of Technology (CSPIT) for their kind cooperation and encouragement which helped me in the completion of this project and preparing the report.

Apart from that, I want to express my heartfelt gratitude to the whole MSI Team, who have helped me in various sectors of the development part and provided many valuable components so I can develop a very effective tool.

Last but not the least, I would also like to thank my colleagues, who have cooperated during the preparation of my report and without them, this project has not have been possible. Their ideas helped me a lot to improve my project report.

## Table of Contents

### Table of Contents

ABSTRACT.....	I
ACKNOWLEDGMENT.....	II
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 PROJECT SUMMARY .....	1
1.2 PURPOSE .....	1
1.3 OBJECTIVES .....	2
1.4 SCOPE .....	2
1.5 TOOLS AND TECHNOLOGY .....	2
<b>CHAPTER 2: PROJECT MANAGEMENT</b>	<b>4</b>
2.1 PROJECT PLANNING.....	4
2.1.1 Project Development Approach and Justification.....	4
2.1.2 Project Effort and Time .....	6
2.1.3 Roles And Responsibilities .....	6
2.1.4 Group Dependencies .....	6
2.2 PROJECT SCHEDULING .....	7
<b>CHAPTER 3: SYSTEM REQUIREMENTS STUDY</b>	<b>8</b>
3.1 USER CHARACTERISTICS .....	8
3.2 HARDWARE AND SOFTWARE REQUIREMENTS.....	8
3.2.1 Hardware Specification:.....	8
3.2.2 Software Specification: .....	8
3.3 ASSUMPTIONS AND DEPENDENCIES.....	8
<b>CHAPTER 4: SYSTEM ANALYSIS</b>	<b>9</b>
4.1 STUDY OF CURRENT SYSTEM.....	9
4.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM .....	9
4.3 REQUIREMENTS FOR NEW SYSTEM .....	9
4.3.1 Functional Requirements .....	9
4.3.2 Non-Functional Requirement.....	11
4.4 FEASIBILITY STUDY .....	12
4.5 FEATURES OF NEW SYSTEM.....	13
4.6 ACTIVITY IN NEW SYSTEM.....	14
4.7 USE CASE DIAGRAM .....	15
4.8 CLASS DIAGRAM .....	16
4.9 HIGH-LEVEL DIAGRAM.....	16
<b>CHAPTER 5: SYSTEM DESIGN</b>	<b>17</b>

5.1	SYSTEM APPLICATION DESIGN .....	17
5.2	SCREENSHOTS .....	18
5.2.1	Handset Automation Tool Dashboard .....	18
5.2.2	TestSuite Selection.....	18
5.2.3	Automation Reports .....	20
5.2.4	Google Drive Integration .....	22
5.2.5	Google Sheet Integration .....	22
5.2.6	Seetest Logs .....	23
5.2.7	Mail Integration .....	24
<b>CHAPTER 6: IMPLEMENTATION PLANNING</b>		<b>25</b>
6.1	IMPLEMENTATION ENVIRONMENT.....	25
6.2	SECURITY FEATURES .....	25
6.3	CODING STANDARD.....	26
6.3.1	Purpose of Coding Standards and Best Practices .....	26
6.3.2	Several Reasons Why Coding Specifications Are Important .....	26
<b>CHAPTER 7: SYSTEM TESTING</b>		<b>28</b>
7.1	TESTING PLAN.....	28
7.2	TESTING STRATEGY .....	28
7.3	TEST SUITS DESIGN.....	29
<b>CHAPTER 8: CONCLUSION AND DISCUSSION</b>		<b>32</b>
8.1	SELF ANALYSIS OF PROJECT VIABILITIES .....	32
8.2	PROBLEM ENCOUNTERED AND POSSIBLE SOLUTION .....	32
8.2.1	SUMMARY OF PROJECT WORK.....	33
<b>CHAPTER 9: LIMITATIONS AND FUTURE ENHANCEMENT</b>		<b>34</b>
9.1	Limitation .....	34
9.2	Future Enhancements .....	34
<b>CHAPTER 10: BIBLOGRAPHY</b>		<b>35</b>



## List of Figures

<b>Figure 1.1</b> Product Applications .....	1
<b>Figure 2.1</b> Automation Life Cycle .....	4
<b>Figure 4.1</b> Activity Diagram .....	14
<b>Figure 4.2</b> Use Case Diagram .....	15
<b>Figure 4.3</b> Class Diagram.....	16
<b>Figure 4.4</b> System Architecture .....	16
<b>Figure 5.1</b> Flow Diagram .....	17
<b>Figure 5.2</b> Tool Dashboard .....	18
<b>Figure 5.3</b> Batch Selection .....	18
<b>Figure 5.4</b> Feature Selection .....	19
<b>Figure 5.5</b> Sanity Selection .....	19
<b>Figure 5.6</b> Stability Selection.....	20
<b>Figure 5.7</b> Execution Details.....	20
<b>Figure 5.8</b> Execution Summary .....	21
<b>Figure 5.9</b> Execution Results(passed).....	21
<b>Figure 5.10</b> Execution Results (Fail) .....	21
<b>Figure 5.11</b> Google Drive Logs Uploads .....	22
<b>Figure 5.12</b> Execution Results updates in Sheet.....	22
<b>Figure 5.13</b> Execution Details updates in Sheet.....	23
<b>Figure 5.14</b> Seetest Logs .....	23
<b>Figure 5.15</b> Seetest Logs containing Steps .....	24
<b>Figure 5.16</b> Mail Integration .....	24

## List of Tables

<b>Table 1.1</b>	Tools and Technology with Their Usages .....	3
<b>Table 2.1</b>	Individual Responsibilities.....	6
<b>Table 2.2</b>	Group Dependencies .....	7
<b>Table 6.1</b>	Standard naming convention.....	27
<b>Table 7.1</b>	Test Cases.....	29

## CHAPTER 1: INTRODUCTION

### 1.1 PROJECT SUMMARY

The project involves the development of an automation tool to validate the product (mobile application) Functionality in various Lab environments and on different device such as mobile (Android & IOS both), tab, ipad etc. It includes execution of automation scripts, development of testscripts for new features, identifying bugs/failure on daily basis and thus fixing issues to improve automation stability and product efficiency in various different kind of environment and production setup.

This project also helps in delivering performance report of the app to the external clients and thus certify the product.



**Figure 1.1** Product Applications

### 1.2 PURPOSE

The main goal of this project for the organization is to make transition from manual testing to automation testing to improve efficiency, reduce testing time, increase productivity, more test coverage, higher precision and accuracy.

### 1.3 OBJECTIVES

- To build an automated scripts that depict the Application working and its accuracy which provides the highend view of how reliable our application in real world usage.
- To provide stability to existent scripts for all new carrier

### 1.4 SCOPE

The scope of Handset Automation is vast in terms of the capabilities it has. It has the capabilities to execute testscripts on different OS and servers, as well as generated different type of logs. The Handset Automation Tool can be used to validate different feature functionalities that are critical to the proper functioning of the PTT handset application.

Handset Automation Tool can also dynamically generate the logs and uploads it to drive. However, the scope of this tool doesn't end here, it can be integrated into other automation tools.

### 1.5 TOOLS AND TECHNOLOGY

Reviewing log data provides critical insights into a wide range of situations, from resolving security risks to debugging application performance irregularities or ensuring compliance with laws. One of the most visible applications for handset automation is undoubtedly the generation of logs for use in troubleshooting servers and the handset application, which can range from application crashes to configuration issues and hardware failure.

To develop this software, we have used the following tools and technology.

- C#
- Seetest
- Experitest and MSTest Framework
- Jira

Table 1.1 Tools and Technology with Their Usages

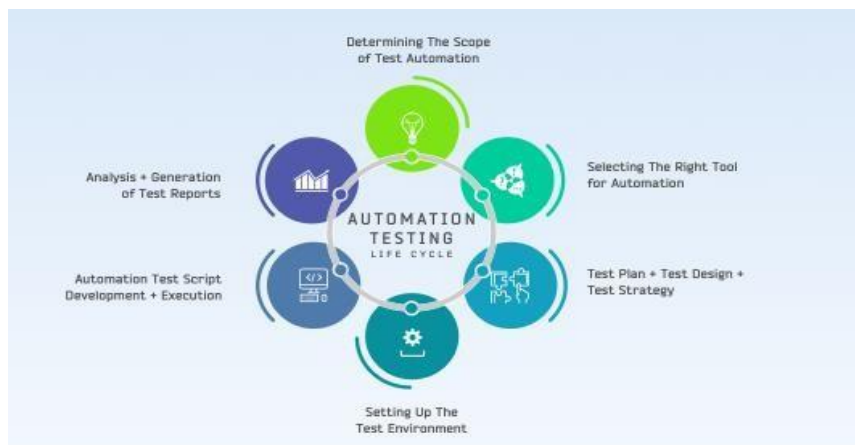
Tools/Technology	Usage
	<p>C# is a simple &amp; powerful object-oriented programming language developed by Microsoft. C# can be used to create various types of applications, such as web, windows, console applications, or other types of applications using Visual studio.</p>
	<p>SeeTest Automation is one of the leading Mobile Automation tools in the industry that supports both Image-Based and Object-Based recognition. It also supports automation of iOS, Android, Windows Phone and BlackBerry applications.</p>
	<p>MSTest framework is a test framework which is included, by default, with Microsoft Visual Studio. It is also referred to as Visual Studio Unit Testing Framework and is popular with developers using the Visual Studio IDE.</p>
	<p>Jira is a software application used for issue tracking and project management. It has become widely used by agile development teams to track bugs, stories, epics, and other tasks.</p>

## CHAPTER 2: PROJECT MANAGEMENT

### 2.1 PROJECT PLANNING

Our Tool is already developed so my major task is to add new feature or to stabilize the existing tool . In the development of new feature, we checked in the first Sprint where we understand the testsuite and try to reproduce testcase manually to see the working of Feature. then, we gathered all the requirements which we need to develop our system. Then, after thoroughly understanding the requirements, we started development. Our development process divides basically into three parts:

- 1) Design and Development of test suite
- 2) Perform Unit testing
- 3) Perform Development testing



**Figure 2.1** Automation Life Cycle

#### 2.1.1 Project Development Approach and Justification

The Agile Model is used for project development. We have selected Agile Model because of its beneficial speed without affecting the quality of product and agile makes the team so much more productive.

- **Agile Model**

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software products. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders.

- **What is Agile?**

Agile models believe that each project must be handled differently, and that existing methods must be tailored to best meet the project's requirements. To deliver specific features for a release, tasks in agile are divided into time boxes (small time frames). The approach is iterative, with each iteration delivering a working software build. In terms of features, each build is incremental; the final build contains all of the features required by the customer.

- **Advantages of using Agile Model**

Customer satisfaction by rapid, continuous delivery of useful software. People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other. Working software is delivered frequently (weeks rather than months). Face-to- face conversation is the best form of communication. Close, daily cooperation between businesspeople and developers. Continuous attention to technical excellence and good design.

- **Disadvantages of using Agile Model**

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is a lack of emphasis on necessary designing and documentation. The project can easily get taken off track if the customer representative is not clear what final outcome that they want.

- Only senior programmers can take the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

### 2.1.2 Project Effort and Time

- We plan on taking a time-based approach to deliver new feature on time. To begin with, we created a set of deadlines to complete certain tasks. This gave us a general idea on how to meet the final deadline on time. Furthermore, these tasks contain various independent divisions of the software, which can be tested individually. Thus, this will help us report our project in parts and test it as we move along.
- Firstly, We Understand application feature in order to have it design in MSTest framework. Then we write scripts according to the testsuite design.
- Then, Testing ensures that our system works efficiently using all valid values and does not give errors. To test the system, we must perform unit testing, development testing and finally the system testing.

### 2.1.3 Roles And Responsibilities

Table 2.1 Individual Responsibilities

Name	Role	Responsibility
Aagam Shah	Automation Intern	Development of new feature and automate the process of logs collection.

### 2.1.4 Group Dependencies

The members of the project should be dedicated to the project and should in turn help each other in whatever problems concerning the project. We are expected not to have any internal or external communication gaps. We are also expected to share the challenges faced by them during design or development so that the team and mentors can brainstorm over every possible dimension. We should report periodically to the concerned faculty and keep them updated regarding the Project.



Table 2.2: Group Dependencies

<b>Name</b>	<b>Responsibilities</b>
Hardik Kothari	<ul style="list-style-type: none"> <li>• Senior Manager, Project Planning</li> </ul>
Syamasundar Jana	<ul style="list-style-type: none"> <li>• Manager, Project Planning, Requirement Gathering, Development</li> </ul>
Dheeraj Sharma, Naveen Kumar	<ul style="list-style-type: none"> <li>• Development, Testing, Feature Execution(IOS)</li> </ul>
Ravinder Singh, Ramesh Bala	<ul style="list-style-type: none"> <li>• Development, Testing, Feature Execution(Android)</li> </ul>
Aagam Shah(Intern)	<ul style="list-style-type: none"> <li>• Development, Testing, Feature Execution (Android)</li> <li>• Automation of Logs Collection and results preparation and updation</li> </ul>
Aaditya Khetan (Intern)	<ul style="list-style-type: none"> <li>• Development ,Testing ,Feature Execution (IOS)</li> <li>• Automation of Logs Collection and results preparation and updation</li> </ul>

## 2.2 PROJECT SCHEDULING

This Tool is already developed and we need to add new feature or to stabilize existing scripts. Development works in iterative nature so we have daily work plan assign by the manager. We have daily standup meeting where task is assign and at the end of the Day we have review meeting where we discuss task done on that day.

## CHAPTER 3: SYSTEM REQUIREMENTS STUDY

### 3.1 USER CHARACTERISTICS

- Ops Team use this Tool to Validate Server upgrade
- Customer representatives use this to validate functionalities on production server.
- It can be used by developers of the company to provide certification of application.

### 3.2 HARDWARE AND SOFTWARE REQUIREMENTS

#### 3.2.1 Hardware Specification:

Following is the list of hardware required for handset Automation Tool:

- Android Devices
- iOS Devices
- Windows Machine ported in MSI Domain
- Non-touch Devices

#### 3.2.2 Software Specification:

Following is the list of software required for the development of the Handset Automation Tool:

- Visual Studio 2019
- Visual Studio Code
- Seetest
- Jira
- PTT Handset Application (Proprietary)
- Postman
- Google API Libraries
- SoapUI
- Experitest and MSTest Framework

### 3.3 ASSUMPTIONS AND DEPENDENCIES

The Working of the Handset Automation Tool requires the following software dependencies to be installed on the users' computers.

- License Seetest Tool
- Microsoft Visual studio
- Server Lab access to clients in use
- Atleast five devices connected to machine

## CHAPTER 4: SYSTEM ANALYSIS

### 4.1 STUDY OF CURRENT SYSTEM

- The Handset automation tool was build in the year 2015 and we are maintaining it till now. The Tool was able to run few of the features of the Push-To-Talk handset application then.
- With time, some of the features became obsolete as the new versions of the mobile operating system as well as the new PTT application versions were released.
- As per the new version release, latest feature need to added in tool to completely automate the feature testing.

### 4.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM

- To complete whole cycle of all feature testing it require more than 3-4 week of time. Which is generally more than actual manual testing time. furthermore clients doesn't have such long time to validate new releases.
- Another weakness of the current system is like user need to collect logs manually from remote machine which is again are subject to availability. some time due to unavailability of machine it delay the analysis of failure. Thus it delay the delivery of products.
- Further, user need to manually generate the report of its execution which generally takes time and might be having some human errors.

### 4.3 REQUIREMENTS FOR NEW SYSTEM

#### 4.3.1 Functional Requirements

##### 4.3.1.1 Automate New Features

**Description:** After proper understanding of new features that are introduced in new versions of the PTT application Scripts need be added to validate the new feature. This is the High priority feature.

**Input:** Testsuite selection, specific client build and server used

**Output:** Automation report and failure summary

#### 4.3.1.2 Generate Various report

**Description:** After Completion of testcase several logs need to be collected thus it helps user to identify failure if any. After analyzing several logs one can differentiate the issue type server issue, clients issue, scripts issue

**Input:** Testcase name and Environment detail

**Output:** generate various report

#### 4.3.1.3 Incorporate new carrier

**Description:** Every new clients have different requirement, scripts should be written in such a way that it should support all new clients with minimal changes from scripts side .

**Input:** Carrier specific build should be chosen

**Output:** Automation reports for that carrier

#### 4.3.1.4 Client Stability

**Description:** Stability code are written in such a way it will do stress test of feature in different environment by running same feature for multiple times

**Input:** Feature name and number times feature executed

**Output:** Stability report

#### 4.3.1.5 Google Sheet Integration

**Description:** After Execution is completed the we need to update the results in sheet for future requirement so after testcase completed we directly update it into Google sheet and send mail to specific user.

**Input:** user mail and Google path, key.

**Output:** Send Drive link to User in Mail

#### 4.3.1.6 Google Drive Integration

**Description:** When Execution is started we need to create drive folder structure and all of the folder Id's are store in config file. When generating reports/ Logs Dynamically and uploading Logs to it and send drive link in mail to specific user

**Input:** User mail ID and Google path, key.

**Output:** Send Drive link to User in Mail

#### 4.3.1.7 Batch Creation

**Description:** Instead of Running single feature clubbing various feature into batch to optimize the execution time. Single precondition for multiple feature ,thus it reduce the execution time as it eliminate Deprovision and Provision clients

**Input:** Batch Name and Server details

**Output:** Batch Reports

### 4.3.2 Non-Functional Requirement

#### 4.3.2.1 Usability

- Usability defines how difficult it will be for a user to learn and operate the system. Usability can be assessed from different points of view.

#### 4.3.2.2 The Efficiency of Use

- Users can easily interact with the system. Most of the tasks a user can complete without any help. It has no complex design, so any user can easily interact.

**4.3.2.3 Intuitiveness:**

- How simple it is to understand the interface, buttons, headings, etc.

**4.3.2.4 Security:**

- Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different users roles. Security also includes protection against viruses and malware attacks.

**4.3.2.5 Reliability**

- Reliability defines how likely it is for the software to work without failure for a given period. Reliability decreases because of bugs in the code, hardware failures, or problems with other system components. To measure software reliability, you can count the percentage of operations that are completed correctly or track the average period of time the system runs before failing.

**4.3.2.6 Performance**

- Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it. Poor performance leads to negative user experience. It also jeopardizes system safety when it's overloaded

**4.4 FEASIBILITY STUDY**

A feasibility study is carried out to see whether the proposed new feature can be developed. A feasibility study is carried out to judge if the new feature is automatable or not and also need to take care for all type of OS if new feature is support for all OS(iOS,Android).

The Handset Automation tool is well-aligned with the objective of Motorola Solutions Inc. and helps in the integration and stability of the Push-To-Talk handset

application. The Handset Automation Tool can be implemented using the currently available technology stack and can be developed under all the given costs and constraints.

#### **4.5 FEATURES OF NEW SYSTEM**

The Handset Automation Tool now supports many new features. All the new and improved features of Tool are listed below:

- In Handset Automation tool now allows to select network type
- New feature added(Geofence/UIRefresh/Sanity/Native Tone etc)
- New Sanity Suite are added.
- New Stability Suite are added.
- Handset Automation Tool is now able to send report mail to user after execution.
- It support to uploads failure logs directly to drive.
- Results are update directly in the sheet.
- Tool has a new and improved User Interface.
- Tool now support Batches executions.
- Support for new Clients added.

## 4.6 ACTIVITY IN NEW SYSTEM

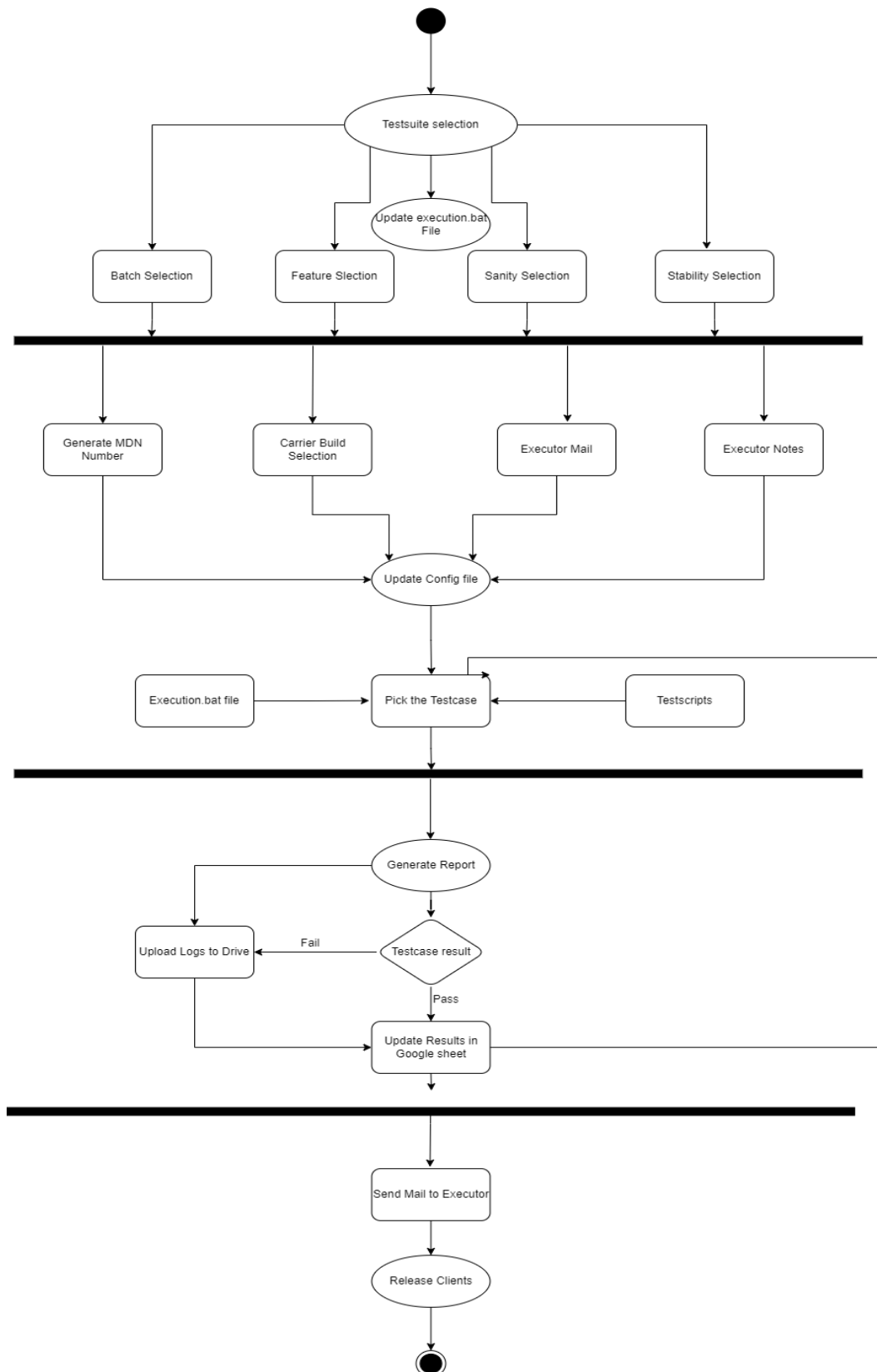


Figure 4.1 Activity Diagram



## 4.7 USE CASE DIAGRAM



Figure 4.2 Use Case Diagram

## 4.8 CLASS DIAGRAM

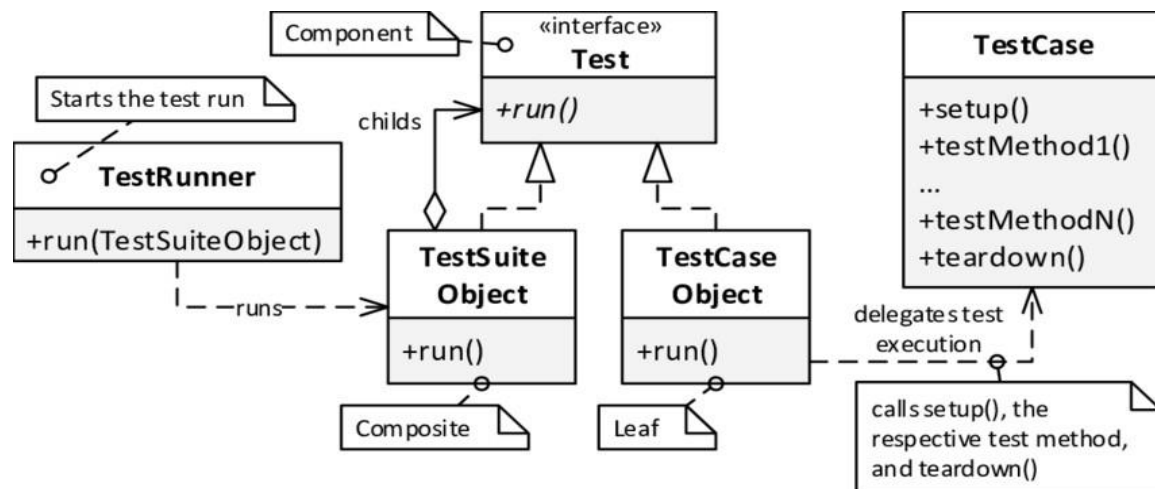


Figure 4.3 Class Diagram

## 4.9 HIGH-LEVEL DIAGRAM

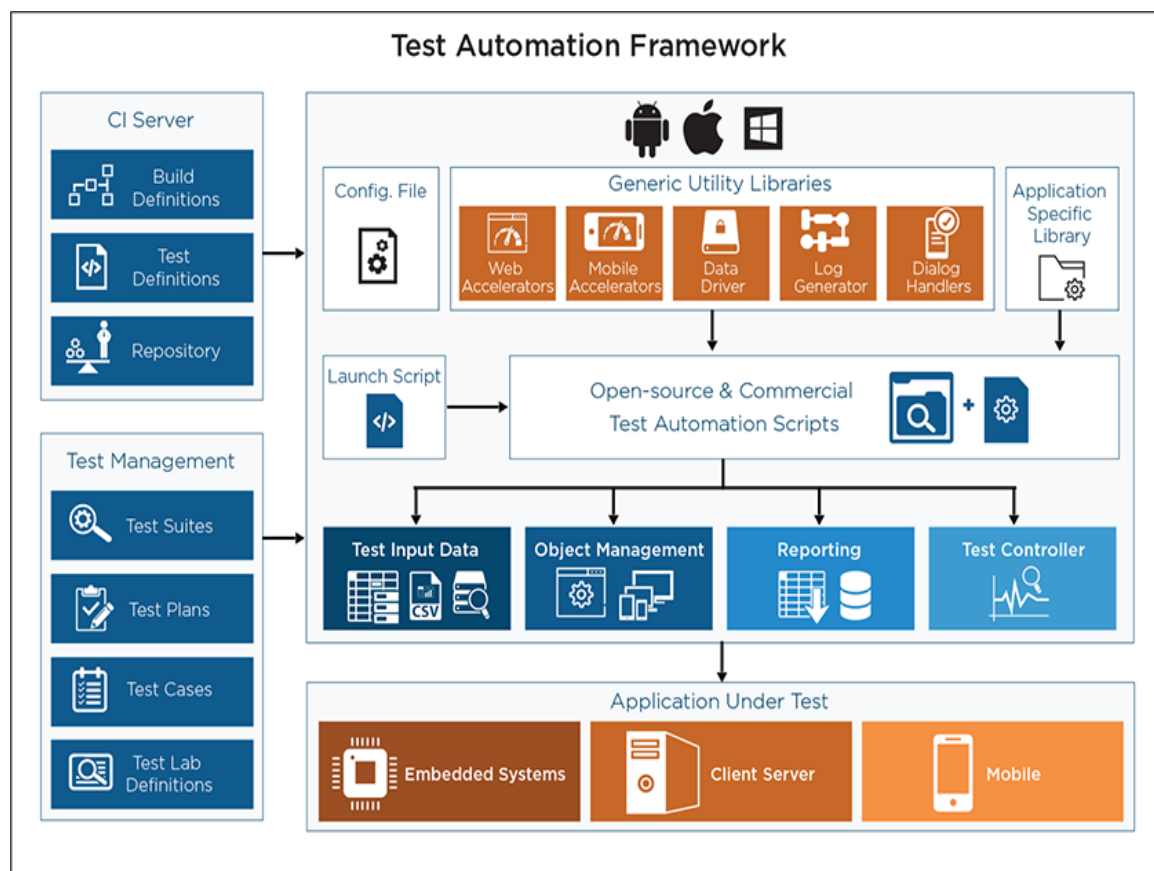
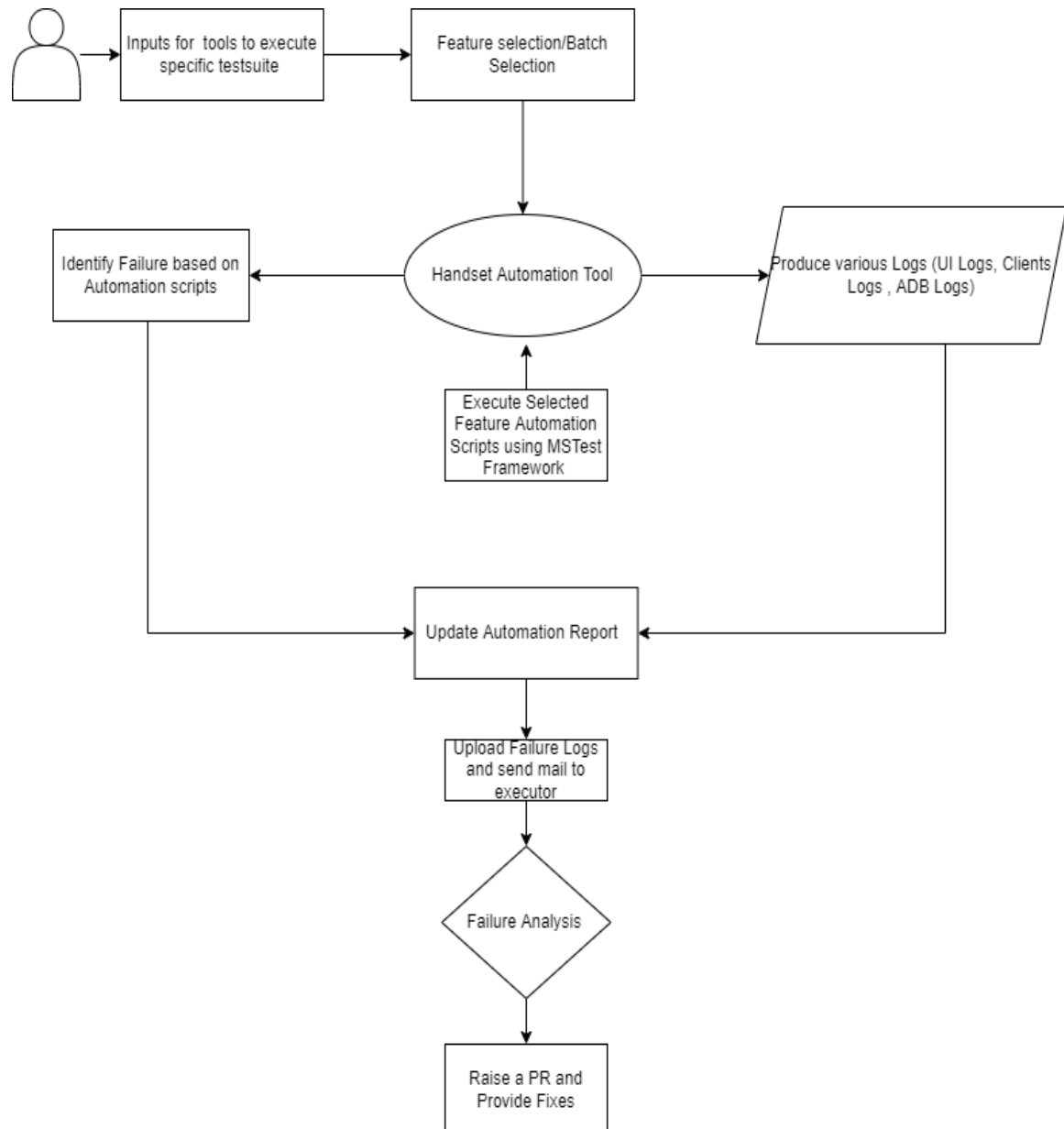


Figure 4.4 System Architecture

## CHAPTER 5: SYSTEM DESIGN

### 5.1 SYSTEM APPLICATION DESIGN

Following is the flow of the software.



**Figure 5.1** Flow Diagram

## 5.2 SCREENSHOTS

### 5.2.1 Handset Automation Tool Dashboard

**Client Automation(SeeTest)**

Help ? About

Test Suite Selection: [Dropdown] Executor Core id: [Text] @motorolasolutions.com Execution Notes: [Text]

Build\_Path: [Text] Browse Update Server Details

Select N/W: [Dropdown: None] ☐ Client Logs collection

CAT Selection: [Dropdown: SOAP] ☐ CrossCarrier

**Selected TestSuites:**

**Server Details**

Release: -  
Corp Name: -  
Server: -  
Account\_ID: -  
Test\_Subsystem: -  
ClientName: -  
Build Version: -

**Main Targets**

Target 1: [Text]  
Target 2: [Text]  
Target 3: [Text]

**Common Targets**

Activate Common Targets From The Tool: ☐ Yes ☒ No

Target 4: [Text]  
Target 5: [Text]

**Execution Logs**

Adding Devices...

Current Execution ID:

GO TO RESULTS WEB PORTAL

Start Execution

Figure 5.2 Tool Dashboard

Above Screenshot shows the main dashboard for Handset Automation Tool

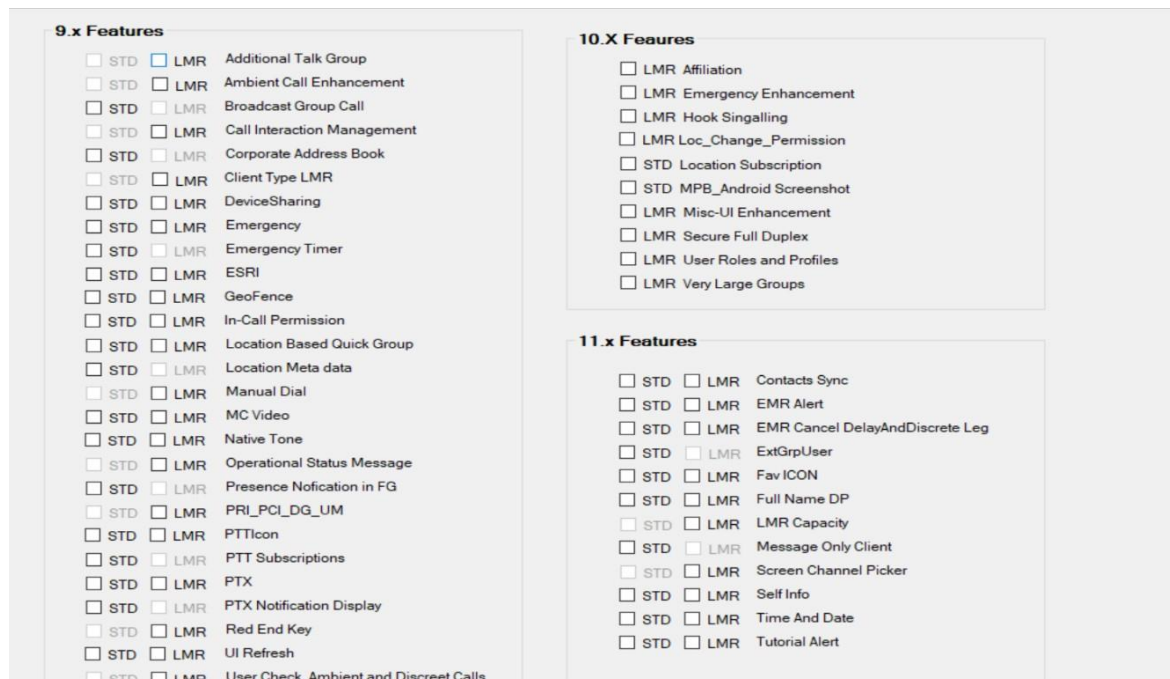
### 5.2.2 TestSuite Selection

Batches	Type	Features Contents
<input type="checkbox"/> Batch 1	STD	Activation, Contact Management, Group Management, Presence, Serach, Settings, Call Processing
<input type="checkbox"/> Batch 2	STD	Contact Management, Decouple Presence, IPA, Settings, Call Processing, Call Alert, Call Rejoin, Misc
<input type="checkbox"/> Batch 3	STD	Call Processing, Presence, Call Rejoin, Supervisor Override, CAT, TalkGroup Scanning, Call Alerts
<input type="checkbox"/> Batch 4	STD	Location Meta Data, Emr Timer, Presence Notify (Android ONLY, iOS-NA)
<input type="checkbox"/> Batch 5	STD	PTX Notification Display, PTX
<input type="checkbox"/> Batch 6	STD	Full Name Display, Tutorial Alert, Contacts Sync, Emr Alert, Self Info, Fav Icon, Time And Date, EMR Cancel Delay
<input type="checkbox"/> Batch 7	LMR	MC Video, Manual Dial, Red EndKey, Manual Dial, Red EndKey
<input type="checkbox"/> Batch 8	LMR	PTT Radio Integration, Private GroupCall Interaction, Duplicate Group, Affiliation
<input type="checkbox"/> Batch 9	LMR	Emergency Enhancement, Emergency, LBQG
<input type="checkbox"/> Batch 10	STD	CAB, PTT Subscription, BGC, ExtGroupUser
<input type="checkbox"/> Batch 11	LMR	ATG, PTX, InCallPermission
<input type="checkbox"/> Batch 12	LMR	HookSignalling, SFDC
<input type="checkbox"/> Batch 13	LMR	URP, VeryLarge Group, UAD
<input type="checkbox"/> Batch 14	STD	LBQG, Location Subscription, GeoFence
<input type="checkbox"/> Batch 15	STD	Emergency, InCallPermission, MCV
<input type="checkbox"/> Batch 16	STD	UI_Refresh, NativeTone, PTT Icon
<input type="checkbox"/> Batch 17	LMR	GeoFence, UIrefresh
<input type="checkbox"/> Batch 18	LMR	Full Name Display, Tutorial Alert, LMR Capacity, Contacts Sync, Emr Alert, Self Info, Fav Icon, Time And Date, EMR Cancel Delay, Emr Discrete Leg

DONE

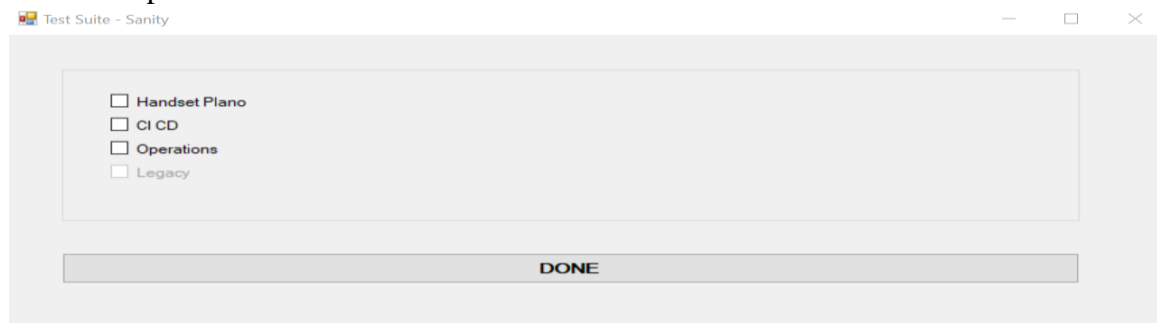
Figure 5.3 Batch Selection

Above Screen appear when user want to select Batches for executions from testsuite selection options.



**Figure 5.4** Feature Selection

Above Screen appear when user want to select Feature for executions from testsuite selection options



**Figure 5.5** Sanity Selection

Above Screen appear when user want to select Sanity for executions from testsuite selection options

**Please select features and no of iterations**

STD	LMR	Features	Iterations	Each TC
<input type="checkbox"/>	<input type="checkbox"/>	PTX	<input type="text"/>	
<input type="checkbox"/>	<input type="checkbox"/>	HKS	<input type="text"/>	
<input type="checkbox"/>	<input type="checkbox"/>	SFD	<input type="text"/>	
<input type="checkbox"/>	<input type="checkbox"/>	EMR	<input type="text"/>	
<input type="checkbox"/>	<input type="checkbox"/>	MCV	<input type="text"/>	
<input type="checkbox"/>	<input type="checkbox"/>	QGC	<input type="text"/>	

**DONE**

**Figure 5.6** Stability Selection

Above Screen appear when user want to select Stability for executions from testsuite selection options

### 5.2.3 Automation Reports

<b>Execution Details</b>	
ID	3299
Date	16-04-2022
Client Version	Android:11.1.0.27B
Server Setup	CI Setup
Automation Version	11_3_0_030
Seetest Version	21.6.8379
CAT Request Type	SOAP
Cross Carrier	Disabled
Network Bearer Type	WiFi
Log Collection	Enabled
Executor Name/email	mcb374@motorolasolutions.com
Target 1 MDN	3795766583
Target 2 MDN	7852839528
Target 3 MDN	5680667237
Device1 Model/OS	Galaxy A21s(11)
Device2 Model/OS	Galaxy A21s(11)
Device3 Model/OS	Galaxy A21s(11)
Device4 Model/OS	Galaxy M31(11)
Device5 Model/OS	Galaxy M21(11)
Failure log path	<a href="#">Drive link</a>

**Figure 5.7** Execution Details

Above figure shows execution details part of Automation results.



Batch 14						
Feature	Pass	Fail	NA	Total Executed	Pass %	Duration (H:M:S)
LBQG_STD	20	0	0	20	100	01:53:00
LocationSubscription_STD	3	0	0	3	100	00:08:16
GeoFence_STD	22	2	0	24	91.67	03:56:17
Total	45	2	0	47	95.74	05:57:33

Figure 5.8 Execution Summary

Above figure shows Execution Summary part of Automation results.

Testcases summary						
Test Case	Result	Start time	End time	Duration	Reason For Failures	
HS_PoC_LBQG_STD_PreCondition	Passed	11.26.46 16 Apr	11.58.08 16 Apr	00:31:22		
HS_PoC_LBQG_01	Passed	11.58.29 16 Apr	12.01.18 16 Apr	00:02:49		
HS_PoC_LBQG_02	Passed	12.01.20 16 Apr	12.03.56 16 Apr	00:02:35		
HS_PoC_LBQG_03	Passed	12.03.57 16 Apr	12.07.10 16 Apr	00:03:12		
HS_PoC_LBQG_04	Passed	12.07.11 16 Apr	12.29.47 16 Apr	00:22:36		
HS_PoC_LBQG_05	Passed	12.29.50 16 Apr	12.31.18 16 Apr	00:01:27		
HS_PoC_LBQG_06	Passed	12.31.19 16 Apr	12.33.21 16 Apr	00:02:01		
HS_PoC_LBQG_07	Passed	12.33.22 16 Apr	12.35.48 16 Apr	00:02:26		
HS_PoC_LBQG_08	Passed	12.35.49 16 Apr	12.42.09 16 Apr	00:06:19		
HS_PoC_LBQG_09	Passed	12.42.11 16 Apr	12.45.43 16 Apr	00:03:32		
HS_PoC_LBQG_10	Passed	12.45.45 16 Apr	12.47.20 16 Apr	00:01:34		
HS_PoC_LBQG_11	Passed	12.47.21 16 Apr	12.49.26 16 Apr	00:02:04		
HS_PoC_LBQG_13	Passed	12.49.27 16 Apr	12.53.12 16 Apr	00:03:45		
HS_PoC_LBQG_14	Passed	12.53.14 16 Apr	12.54.54 16 Apr	00:01:40		
HS_PoC_LBQG_15	Passed	12.54.55 16 Apr	12.58.49 16 Apr	00:03:54		
HS_PoC_LBQG_16	Passed	12.58.51 16 Apr	13.01.59 16 Apr	00:03:08		
HS_PoC_LBQG_17	Passed	13.02.00 16 Apr	13.04.44 16 Apr	00:02:43		
HS_PoC_LBQG_18	Passed	13.04.47 16 Apr	13.08.25 16 Apr	00:03:37		

Figure 5.9 Execution Results(passed)

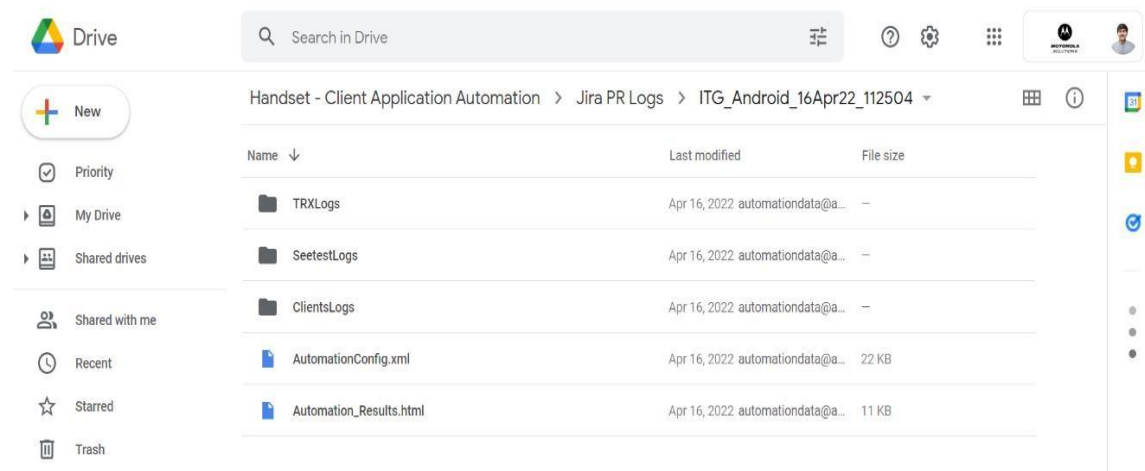
Above figure shows Execution results for passed testcase of Automation results.

HS_PoC_PubCorp_GeoFence_16	Passed	14.20.33 16 Apr	14.34.15 16 Apr	00:13:42	
HS_PoC_PubCorp_GeoFence_19	Passed	14.34.19 16 Apr	14.42.23 16 Apr	00:08:04	
HS_PoC_PubCorp_GeoFence_20	Failed	14.42.27 16 Apr	15.26.32 16 Apr	00:44:04	
HS_PoC_PubCorp_GeoFence_21	Passed	15.30.30 16 Apr	15.42.34 16 Apr	00:12:04	
HS_PoC_PubCorp_GeoFence_22	Passed	15.42.38 16 Apr	15.47.41 16 Apr	00:05:02	
HS_PoC_PubCorp_GeoFence_23	Passed	15.47.45 16 Apr	15.51.00 16 Apr	00:03:15	
HS_PoC_PubCorp_GeoFence_24	Passed	15.51.05 16 Apr	15.56.36 16 Apr	00:05:31	
HS_PoC_PubCorp_GeoFence_25	Passed	15.56.42 16 Apr	16.02.13 16 Apr	00:05:31	
HS_PoC_PubCorp_GeoFence_26	Passed	16.02.18 16 Apr	16.14.46 16 Apr	00:12:28	
ZHS_PoC_GFNH_01	Passed	16.14.51 16 Apr	16.25.33 16 Apr	00:10:42	
ZHS_PoC_GFNH_02	Passed	16.25.38 16 Apr	16.38.06 16 Apr	00:12:27	
ZHS_PoC_PubCorp_Location_01	Failed	16.38.11 16 Apr	17.09.07 16 Apr	00:30:55	
ZHS_PoC_PubCorp_MultLocWtch_01	Passed	17.12.57 16 Apr	17.35.41 16 Apr	00:22:43	

Figure 5.10 Execution Results (Fail)

Above figure shows Execution results with failure testcase.

### 5.2.4 Google Drive Integration



**Figure 5.11** Google Drive Logs Uploads

Above Figure shows how the Folder structure of Google Drive was created and logs uploaded once execution start.

### 5.2.5 Google Sheet Integration

EXECUTED BY	FEATURE	PLATFORM	SERVER	PASS	FAIL	NA	TOTAL	PASS %	TOTAL TIME	RELEASE VERSION	CLIENT TAG	AUTO TAG	EXECUTION ID	SEETEST VERSION	DEVICE 1	DEVICE 2
Naveen Kumar Kumar	EmgCncDiy_LMR	IOS	WUCLAB	4	0	0	4	100.00	00:33:12	11.3.0	11.3.0.05H	11_3_0_01C	1	21.6.8379	iPhone XS(14.6)ne 11 Pro Max(11.3.0.05H)	
Naveen Kumar Kumar	DuplicateGroups_LMR	IOS	WUCLAB	4	0	0	4	100.00	00:51:38	11.3.0	11.3.0.05H	11_3_0_01C	548	21.6.8379	iPhone XS(14.4) iPhone X(14.3) c	
Naveen Kumar Kumar	_DiscreteEmgCal_LMR	IOS	WUCLAB	2	0	0	2	100.00	00:45:41	11.3.0	11.3.0.05H	11_3_0_01C	1	21.6.8379	iPhone XS(14.6)ne 11 Pro Max(11.3.0.05H)	
Naveen Kumar Kumar	DateTime_LMR	IOS	WUCLAB	4	1	0	5	80.00	00:57:38	11.3.0	11.3.0.05H	11_3_0_01C	1	21.6.8379	Device 1 Device 2	
Naveen Kumar Kumar	ULM_LMR	IOS	WUCLAB	6	1	0	7	85.71	02:12:22	11.3.0	11.3.0.05H	11_3_0_01C	548	21.6.8379	Device 1 Device 2	
Agam Shah	FullNameDisplay_LMR	Android	ITG	4	0	0	4	100.00	00:22:02	11.4.0	11.4.0.02C	11_3_0_01C	1144	21.6.8379	Pixel 4(11) LEX L11a(9) s	
Ravinder Singh	Emg_LMR	Android	WUCLAB	0	0	0	0	NaN	0:0:0	11.3.0	11.3.0.08C	11_2_0_11F	1	21.6.8379	Device 1 Device 2	
Dheeraj Sharma	Batch-1	IOS	ITG	191	14	0	205	93.17	18:00:10	11.4.0	11.4.0.01Q	11_3_0_01C	2901	21.6.8379	Device 1 Device 2	
Naveen Kumar Kumar	UIRefreshTest_LMR	IOS	WUCLAB	34	2	0	36	94.44	05:13:22	11.3.0	11.3.0.05H	11_3_0_01C	1	21.6.8379	Device 1 Device 2	
Naveen Kumar Kumar	UIRefreshTest_LMR	IOS	VERIZONLAB	15	21	0	36	41.67	07:41:23	11.3.0	11.3.0.05H	11_3_0_01C	1	21.6.8379	iPhone 11(14.7) iPhone 11(14.5.11)	
Dheeraj Sharma	ManualDial_LMR	IOS	ITG	4	0	0	4	100.00	00:45:26	11.4.0	11.4.0.01Q	11_3_0_01D	2903	21.6.8379	iPhone X(14.7) iPhone XS(14.8.11)	
Naveen Kumar Kumar	InCallPermission_LMR	IOS	VERIZONLAB	2	3	0	5	40.00	01:07:32	11.3.0	11.3.0.05H	11_3_0_01C	1	21.6.8379	Device 1 Device 2	
Dheeraj Sharma	LBOG_LMR	IOS	ITG	11	0	0	11	100.00	01:21:45	11.4.0	11.4.0.01Q	11_3_0_01D	2903	21.6.8379	iPhone X(14.7) iPhone XS(14.8.11)	
Naveen Kumar Kumar	OSM_LMR	IOS	WUCLAB	27	1	0	28	96.43	03:18:14	11.3.0	11.3.0.05H	11_3_0_01D	550	21.6.8379	iPhone XS(14.4) iPhone X(14.3) c	

**Figure 5.12** Execution Results updates in Sheet

Above Figure shows how the Execution Results with extra details are updated in Google sheet



Date	IODT Tag	Email	Build Version	Server	Features	Mac ID	Executor Name
Apr-19	11_3_0_031	kxm48@motorolasolutions.com	11.3.0.11A	ATTLAB	BGC_STD,	C4-34-6B-5A-11-4B	
Feb-8	11_2_0_10Y	Aaditya Khetan	10.0.3.62G	ATTMIRROR	ATG_LMR,UIREFRESH_LMR,	2C-44-FD-24-9E-B7	ATT M138
Feb-8	11_2_0_10Y	Ravinder Singh	11.3.0.07P	VERIZONLAB	GEOFENCE_LMR,UIREFRESH_LMR,NATIVETONE_LMR,InCallPermission_L	64-51-06-61-1F-86	M138
Feb-8	11_2_0_10Y	Naga Ramesh Balla	11.2.0.23P	WOCLAB	RedEndKey_LMR,MCV_LMR,Affiliation_LMR,	60-7B-9D-5D-25-8C	BALLA08FEB
Feb-8	11_2_0_10Y	Aaditya Khetan	11.3.0.08F	WOCLAB	Location_Subst_STD,PTTSub_STD,MD_LMR,NATIVETONE_STD,	E8-8A-64-C7-BF-12	GSsheetResult_UT
Feb-8	11_2_0_10Y	Dheeraj Sharma	10.0.3.46D	ATTMIRROR	UIREFRESH_LMR,GEOFENCE_LMR,	F8-84-6A-AA-0D-6A	
Feb-8	11_2_0_10Y	Dheeraj Sharma	10.0.3.62G	ATTMIRROR	DS_LMR,	98-FA-9B-14-0D-D3	
Feb-8	11_2_0_10Y	Hitendra Patel	11.3.0.08F	QC11	HKS_LMR,	BC-E9-2F-FC-5C-D5	
Feb-8	11_2_0_10Y	Aagam Shah	10.0.3.62J	ATTMIRROR	Affiliation_LMR,BGC_STD,OPS_Sanity,	E8-8A-64-28-D8-65	Att
Feb-8	11_2_0_10Y	Aaditya Khetan	11.3.0.08F	WOCLAB	Location_Subst_STD,PTTSub_STD,MD_LMR,NATIVETONE_STD,	E8-8A-64-C7-BF-12	GSsheetResult_UT
Feb-8	11_2_0_10Y	Hitendra Patel	11.3.0.08F	QC11	HKS_LMR,	BC-E9-2F-FC-5C-D5	
Feb-8	11_2_0_07S	Gobinath Lingappan	10.0.3.62J	ATTMIRROR	IODT_B1,		mobile-Rajeev
Feb-9	11_2_0_10Y	Aaditya Khetan	11.3.0.08C	WOCLAB	Location_Subst_STD,PTTSub_STD,MD_LMR,	E8-8A-64-C7-BF-12	GSsheet_UT
Feb-9	11_2_0_10Y	Ravinder Singh	11.4.0.02B	ITG	IODT_B1,	54-05-DB-87-E8-BE	CI SETUP
Feb-9	11_2_0_10Y	Ravinder Singh	11.4.0.02B	ITG	IODT_B1,	54-05-DB-87-E8-BE	
Feb-9	11_2_0_10Y	Aaditya Khetan	11.3.0.08C	WOCLAB	FULL_NAME_STD,DATE_TIME_STD,SYNC_CONTACTS_STD,	E8-8A-64-C7-BF-12	GSsheet_UT
Feb-9	11_2_0_10Y	Aaditya Khetan	11.3.0.08C	WOCLAB	FULL_NAME_STD,DATE_TIME_STD,SYNC_CONTACTS_STD,	E8-8A-64-C7-BF-12	GSsheet_UT
Feb-9	11_2_0_10Y	Ravinder Singh	11.4.0.02B	ITG	IODT_B1,	54-05-DB-87-E8-BE	
Feb-9	11_2_0_10Y	Ravinder Singh	11.3.0.08F	WOCLAB	IODT_B1,	54-05-DB-87-E8-BE	
Feb-9	11_2_0_10Y	Ravinder Singh	11.3.0.08F	WOCLAB	GEOFENCE_LMR,	54-05-DB-87-E8-BE	
Feb-9	11_2_0_10Y	Aagam Shah	10.0.3.62J	ATTMIRROR	Affiliation_LMR,BGC_STD,OPS_Sanity,	E8-8A-64-28-D8-65	Att
Feb-9	11_2_0_10Y	Aagam Shah	10.0.3.62J	ATTMIRROR	GEOFENCE_STD,	E8-8A-64-28-D8-65	Att
Feb-9	11_2_0_10Y	Aagam Shah	10.0.3.62J	ATTMIRROR	GEOFENCE_STD,	E8-8A-64-28-D8-65	Att
Feb-9	11_2_0_10Y	Aaditya Khetan	11.3.0.08C	WOCLAB	FULL_NAME_STD,DATE_TIME_STD,SYNC_CONTACTS_STD,	E8-8A-64-C7-BF-12	GSsheet_UT
Feb-9	11_2_0_10Z	Ravinder Singh	11.3.0.07P	VERIZONLAB	OPS_Sanity,	64-51-06-61-1F-86	M138
Feb-9	11_2_0_10Z	Aagam Shah	10.0.3.62J	ATTMIRROR	ETimer_STD,Affiliation_LMR,BGC_STD,CIM_LMR,	E8-8A-64-28-D8-65	Anydesk ATT
Feb-9	11_2_0_10Z	Aaditya Khetan	10.0.3.62G	ATTMIRROR	IODT_B1,	2C-44-FD-24-9E-B7	ATT M138
Feb-9	11_2_0_10Z	Aaditya Khetan	11.3.0.08C	WOCLAB	FULL_NAME_STD,SYNC_CONTACTS_STD,FAV_ICON_STD,TUT_ALERT_ST	E8-8A-64-C7-BF-12	WOCLAB

Figure 5.13 Execution Details updates in Sheet

Above Figure shows how the Execution Details with machine deatil are updated in Google sheet

## 5.2.6 Seetest Logs

The screenshot displays the Seetest Framework interface. On the left, a 'Test Execution Summary' lists 22 steps, all marked with green checkmarks, indicating successful execution. The steps include actions like 'Set device to adb:Target3', 'Launch native activity', and 'Click on button'. On the right, the 'Result Details' section shows 'Set device to adb:Target3' as 'Passed' with a 'Total Time: 0:04 Seconds'. Below this, a 'Result Screenshot' shows a mobile app interface with a status bar at the top displaying '16:38' and 'Available'. The app interface includes a navigation bar with icons for home, star, person, and others, and a list of contacts labeled '00.Target1' through '00.Target5'.

Figure 5.14 Seetest Logs

Above Figure shows UI Logs generated by the tool using Seetest Framework

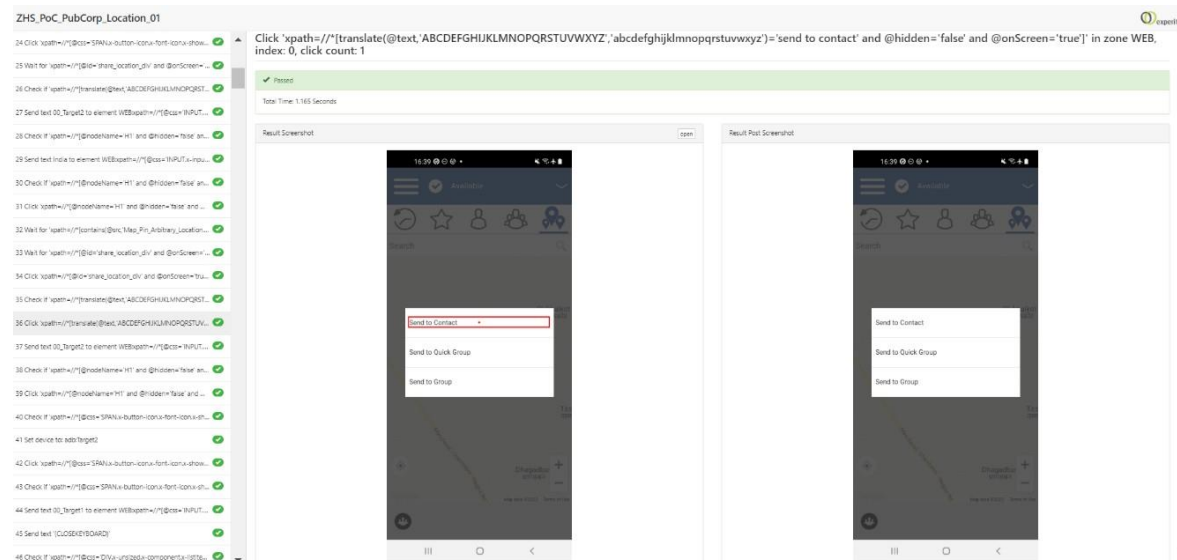


Figure 5.15 Seetest Logs containing Steps

Above Figure shows UI Logs generated and how steps perform by the tool using Seetest Framework

### 5.2.7 Mail Integration

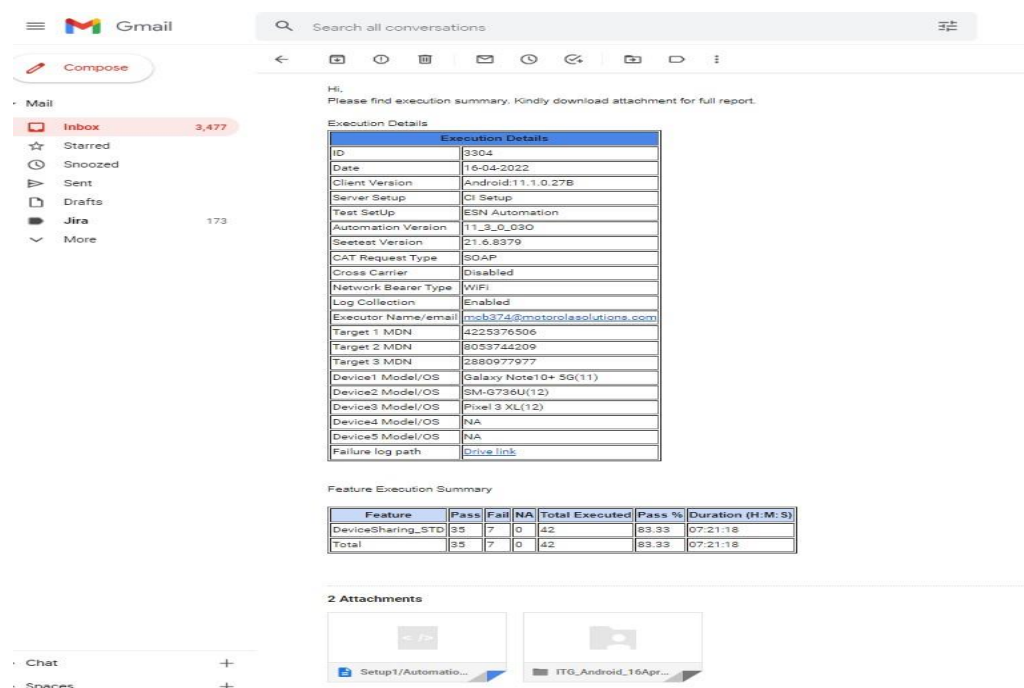


Figure 5.16 Mail Integration

Above Figure shows Mail delivery of automation report by the tool.

## CHAPTER 6: IMPLEMENTATION PLANNING

### 6.1 IMPLEMENTATION ENVIRONMENT

The application is a single server multiple client application. Multiple users can use the system at a time. Each machine need to connect with appropriate VPN to access Lab server. Apart from this Each machine should have 5 devices connected to it. Machine should have active connection during the time of execution.

Handset Automation Tool is implemented for Windows 10 and onwards operating systems which are ported to the MSI domain

The basic hardware requirement for implementation are as follows:

- 8 GB DDR4 RAM
- Intel Core i7 8250 CPU @1.60 GHz
- 512 GB SSD + 2TB HDD
- Internet Speed should be More than 2 MBPS

### 6.2 SECURITY FEATURES

The security of the system is one of the main aspects of any software building. Miss handled data could lead to a huge forfeiture and a complete breakdown of the business for the client. Our Tool does not store any user data so it is not vulnerable to any data breaches or other cyber security incidents. Tool encrypts all the authentication tokens using SHA-256 encryption to avoid unauthorized access. Tool is developed in such a way that it can run on only Motorola Solutions Inc. ported devices.

## **6.3 CODING STANDARD**

### **6.3.1 Purpose of Coding Standards and Best Practices**

To develop reliable and maintainable applications, one must follow coding standards and best practices. The naming conventions, coding standards, and best practices described in this document are referred to by us. Several standards exist in the programming industry. None of them are wrong or bad, and one may follow any of them. More important is, selecting one standard approach and ensuring that the set of standards defined is well adopted.

### **6.3.2 Several Reasons Why Coding Specifications Are Important:**

Your peer programmers have to understand the code you have written. A coding standard acts as the blueprint for all the team to decipher the code. Simplicity and clarity achieved by consistent coding save you from common mistakes. If you revise your code after some time, it becomes easy to understand that code. Its industry-standard to follow a particular standard to be more quality in software. The coding standards contribute to an improved comprehension of source code. Perhaps one of the most influential aides to understanding the logical flow of an application is how the various elements of the application are named. A name should tell "what" rather than "how". By avoiding names that expose the underlying implementation, which can change, you preserve a layer of abstraction that simplifies the complexity. Naming Conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier – for example, whether it's a constant, class, etc. which can help understand the code.

Some of the specifications are as follows:

- Indentations: 4 spaces
- Maximum line length: 79 characters
- Class separation: 2 Blank line
- Function separation: 1 Blank line

- Method definition separation: 1 Blank line
- Encoding: UTF-8
- Standard naming convention:

Table 6.1 Standard naming convention

Identifier	Convention
Module	lowercase
Class	CapWords
Functions	lowercase
Methods	lowercase
Type variables	CapWords
Constants	UPPERCASE
Package	lowercase

## CHAPTER 7: SYSTEM TESTING

### 7.1 TESTING PLAN

Planning the test procedure plays an important role in the success of the application. This phase includes important discussions around understanding what features to automate, the use case and the requirements of the client in the field, organizing resources, preparing the team on task distribution, setting up the budget, deciding the deadline to deliver test cycle results and reporting issues in the product and allocating the issues to respective teams to review and fix the bugs. Once the plan is prepared, it is important to review the plan with the team.

### 7.2 TESTING STRATEGY

The testing strategy equipped by the team mainly consists of the following stages –

- Define the scope of the tests

In this phase, it's important to segregate the requirements as automatable or non-automatable. The goal is to automate the most of the possible features that are feasible with the developed automation tool and can be verified with the SeeTest library.

- Preparing the testbed

This step involves setting up the server required to be used for the clients under test and prepare the environment in order to make sure that the tests are executed properly.

- Implementing the test cases

After carefully reviewing the feature requirements, test cases are written to validate the input and output expectations keeping in mind other factors like duration of test, dependency of test on other factors, compatibility with different OS – like android and IOS, etc.

- Review the test cases developed

Once the test suite is developed it is thoroughly reviewed with the entire team to discuss about the possible portholes and obstacles, if any, can be blocked before

executing the entire test cycle and delivering test reports to the external teams and clients.

- Maintenance of the test suite.

Test maintenance is a non-going effort, where all the tests are updated regularly to accommodate the functional, UI changes, network changes, introduction of new releases and client. It includes addition of any new test cases, enhancing existing test or the automation framework components for better performance.

### 7.3 TEST SUITS DESIGN

The main task in writing test cases is designing the test suites to make sure that all the features of the product are covered and well tested. It is at this stage where all the corner cases and use cases in the field are studied deeply in order to come up with an accurate test suite.

Table 7.1 Test Cases

Test Case	Expected Outcome	Actual Outcome	Status
Launching of automation tool	The connected devices must be added.	The connected devices are added.	Passed
Launching the tool without VPN connected	Tool should launch but execution should fail with network error	Tool launched but execution failed to start with network error	Passed
Drive folder creation on start execution	Once the execution is started a folder should be created on google drive.	On the start of execution, drive folder is created successfully	Passed
Log collection for passed test cases.	Seetest logs and trx files must be created in the current folder.	Seetest logs and trx are created in the current folder	Passed

Log collection for failed test cases.	Seetest logs, client logs and trx files must be created.	Seetest logs, client logs and trx files are created	Passed
Update results table for each test case	Each test case result must be updated with result, time taken and test case name in the result table	Results for each test case are updated as expected	Passed
Feature wise result table updating	Result for each feature must include total pass, fail, executed, total time parameters	Result for each feature contains total pass, fail, executed, and total time parameter	Passed
Updating result to google sheet after each feature	Once a feature completes, the result should be updated in the google sheet	On feature completion, results are updated in google sheet.	Passed
Uploading failed logs to drive	The logs for all the failed cases must be uploaded to google drive	The logs for failed cases are uploaded on the google drive	Passed
Random MDN generation for feature execution	Once the execution starts, 5 randomly generated MDNs must be provisioned in the server	On execution start, 5 MDNs are provisioned in the server	Passed
Execution failure if device is offline	Execution must fail if any of the connected device goes offline	Execution fails if any of the connected device is not available	Passed
Execution behavior when	If an offline device is made available	Execution proceeds as expected when	Passed



offline device is available again	again, the test execution should proceed without any problem.	devices are available again	
Features selection from automation tool	All the features selected must be executed one after the other	All the features selected from the tool are executed after one another	Passed
Picking up test cases in an execution	Test cases should be picked up according to the priority number assigned	Test cases are executed according to the allotted priority	Passed
Test case category behavior	Test cases must be executed for the categories allocated only	Test cases are executed on the basis of categories allocated.	Passed
Send mail on execution completion	The executor must receive a mail on completion of execution	The executor receives an email on execution completion	Passed

## **CHAPTER 8: CONCLUSION AND DISCUSSION**

### **8.1 SELF ANALYSIS OF PROJECT VIABILITIES**

The automation tool is very effective and helpful in delivering test report cycles for any client and is used to test the product working before making it available to the client. The results are extremely crucial in releasing new features and client builds. It also helps in identifying failures and troubleshooting when the client network goes down in the field.

### **8.2 PROBLEM ENCOUNTERED AND POSSIBLE SOLUTION**

As the number of features and test cases are large, it takes around 3 to 4 weeks to deliver test cycle reports which is considered to be considerably more. To tackle this problem, we optimized the structure of many features to re-use the already activated and provisioned clients instead of provisioning new ones every time.

Another problem encountered by the team was the manual effort required to update results in google sheet and copying the failures logs to the google drive every day. Every team member spent around 30 mins daily in this manual task. We automated the task to update the results and upload logs during the execution itself to completely remove the manual effort required.

This also solved the problem of machine dependency as most of our lab machines and servers are located outside of India and would most of the times be not accessible due to various unavoidable reasons, which would not allow the test engineer to collect logs and provide fixes.

### 8.2.1 SUMMARY OF PROJECT WORK

Since, Motorola solutions is spread across 100+ countries, it has a huge customer base of which most of them are security organizations, emergency workers, and frontline workers. Since, the product directly effects the lives of thousands and hundreds of people it is utmost important that the product works under all scenarios and provides a success rate of above 99%. For a product with such importance, automation testing can work wonders as it allows us to test the product rigorously with all corner cases, simulate all type of network environments and provide accurate, precise and error free results to the client. Developing and working on a tool that automates the testing of mobile devices provided me an opportunity to learn automation testing with selenium, experitest, MSTest framework and seetest tool. I automated the product on different OS types (IOS and android), different network types (3G, 4G, and Wi-Fi), and different lab environments and setup. I also worked on optimization of execution time and reduced the execution time from 13-14 hours to around 8-9 hours per run. I also worked on google APIs to insert test results directly into the google sheets and upload logs directly to google drive to save 30-45 mins of each team member daily.

## **CHAPTER 9: LIMITATIONS AND FUTURE ENHANCEMENT**

### **9.1 Limitation**

- One of the major limitations of the automation tool is its inability to analyze the failure logs and provide a reason on why a particular test case had failed in the first place.
- It still takes around 2-3 weeks to deliver complete test cycle report which according to me can be further reduced to 10-14 days.
- Inability of the tool to collect clients logs for IOS devices due to permissions and policies devised by IOS.

### **9.2 Future Enhancements**

- Develop a tool to analyze failure logs and provide initial analysis to the tester with the reason why a particular test case failed.
- Develop an app to collect client logs for IOS and upload it to google drive directly from the device itself.

## CHAPTER 10: BIBLIOGRAPHY

- **C# Documentation:**  
<https://docs.microsoft.com/en-us/dotnet/csharp/>
- **Google Drive .NET API Reference document:**  
<https://developers.google.com/drive/api/quickstart/dotnet>
- **Windows Forms Application Documentation:**  
<https://docs.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>
- **SeeTest Tool:**  
<https://digital.ai/continuous-testing>