

QEA-Using Selenium & JAVA

Project report submitted in partial fulfillment of the requirement for
the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Lakshya Tyagi (181350)

Under the supervision of

Dr. Amol Vasudeva

To



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat, Solan-
173234, Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**QEA-Using Selenium & JAVA** ” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2022 to June 2022 under the supervision of Dr Amol Vasudeva Assistant Professor (SG) Department of Computer Science And Engineering.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Lakshya Tyagi, 181350

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

Dr Amol Vasudeva

Assistant Professor (SG)

Department of Computer Science

And Engineering.

Nithyakalyani Muthukumar

Programmer

Human Resource - Genc

Acknowledgement

This article is to acknowledge all the people without whom this project would not have been completed. Majorly, I would like to thank my supervisors Dr. Amol Vasudeva and Nithyakalyani Muthukumar who gave their immense support, dedicated their time towards it and made me understand how to make this project. Without their guidance and learning, this work would not have been completed.

The preparation of this project was an immense learning experience and we inculcated many personal qualities during this process like responsibility, punctuality, confidence and others.

We consider, project as a bridge between practical and theoretical learning and with this thinking we worked on the project and made it successful due to timely support and efforts of all who helped me.

Once again, we would like to thank our classmates and our families for their encouragement and help in designing and making our project creative, we are obliged to all of these. Only because of them we are able to create our project and make it a good and enjoyable experience.

Table Of Contents

S.No	Description	Page No.
1.	Abstract	(iv)
2.	Chapter-1	1-10
3.	Chapter-2	11-20
4.	Chapter-3	21-32
5.	Chapter-4	33-38
6.	Chapter-5	38
7.	References	39

Abstract

Quality engineering and assurance (QEA) is a subtype of quality management that incorporates ensuring that quality control measures and checks are applied at every step of product or software development. It's the act of having quality assurance (QA) and quality engineering .

We all are aware that we live in a perfectionist period when every single flaw has an impact on us. Bad applications can cost a lot of money to customers and users. As a result, you have to make amends or run the risk of losing your faith.. With a robust QA testing process, you may identify any potential problems or difficulties that may arise in the future.

This project will deal with testing the various functionality and working of the various websites. We will use JAVA and work with spring core, selenium, and sql to achieve good results and test every functionality.

Spring is a simple framework that is easy to use. Because it supports different frameworks such as Struts, Hibernate, Tapestry, EJB, JSF, and others, it can be thought of as a framework of frameworks. In a larger sense, the framework can be defined as a structure that allows us to solve numerous technical challenges.

Selenium is an free browser automation tool. It provides a single interface for writing test scripts in various computing languages such as Ruby, Java, NodeJS, PHP & Perl, Python and C #.

CHAPTER NO. 1

1 Introduction

Software testing determines whether the actual software product matches the expected requirements and ensures that it is defect-free. It involves putting software/system components through their paces with manual or automated techniques in order to assess one or more qualities of interest.

Software testing's purpose is to detect mistakes, gaps, or missing requirements in relation to the actual requirements.

When working on a software development project, keep in mind that faults might occur at any point during the life cycle.

There are only a few that have yet to be discovered. As a result, Quality Assurance is crucial.

There's a good probability the final code will include design and functionality flaws. It is a requirement for completing software testing to identify flaws before they occur in a critical environment.

Program testing is vital because it allows any defects or errors in the software to be found and corrected before the final product is delivered. A well-tested software solution assures dependability, security, and excellent performance, resulting in time savings, cost savings, and customer satisfaction.

It's an important aspect of the procedure.

Testing Principles:

- (i) All tests should adhere to the customer's specifications.
- (ii) We need to hire a third party to test our program.
- (iii) Exhaustive testing is impossible. Because we require the most testing possible based on the application's risk assessment.
- (iv) Before doing any tests, ensure they are all set out.
- (v) It follows the Pareto rule, which claims that 80% of software mistakes are caused by 20% of program components.
- (vi) Begin with little parts and work your way up.

Types of Testing:

1. Unit Testing It focuses on the smallest part of software development. We examine either a single item or a group of interconnected components. The programmer often performs this by seeing the outputs of sample input.

2. Integration Testing The idea is to employ unit-tested components to build a design-driven program structure. Integration testing entails merging several components to get a result.

(a) Testing in a Black Box: It's a validation tool. We disregard the internal working mechanisms and concentrate on the result.

(b) White Box Testing: It's a verification tool. This section is concerned with internal mechanics, or how the output is generated.

3. Functional-Testing Functional testing is a type of testing that determines whether each feature of the application is working as required by the program. The output of each function is compared with the related requirement to see if it meets the end user's expectations.

1.2 Problem Statement

The following are the general issues that the Test Automation project will address:

- (a) to provide a dependable and simple-to-use software platform for administration.
- (b) to create a database of questions that can be used to build tests automatically based on provided criteria
- (c) to develop a secure test-taking application with a sophisticated automated grading tool as well as non-automated grading capabilities

While there are a variety of available and public domain programs that have addressed these issues, the Test Tool offers a solution with several feature improvements over previous tools.

1.3 Objectives

Quality assurance is the primary function of QEA. A quality assurance engineer works to improve software development processes and prevent production problems. To put it another way, they ensure that the software development team is doing things correctly. A QA engineer's job includes a variety of responsibilities. The next paragraphs will discuss them.

QA tasks

The following is a list of typical QA specialist tasks:

- Verifying if the product meets the specifications
- Risk evaluation
- Idea generation to increase product quality
- Test preparation
- Examining the results of the tests

There are several sub-specializations within the QA specialization, such as quality control, software testing, and others. Let's go over the differences between these QA sub-disciplines.

Quality assurance engineers' primary goal is to remove flaws. Quality Control specialists, on the other hand, examine test findings and look for flaws. They are in charge of detecting and

eliminating flaws in a product (or, in other words, these engineers make sure that developers get the results they expect).

The most important QA jobs

Within specific projects, QA specialists can also play a variety of roles. Test Analyst, Test Designer, Test Executor, and Test Manager are the four main QA responsibilities.

Static testing- of requirements is performed to ensure that they are comprehensive and consistent.

Test Designer- develops a collection of tests based on requirements and plans the appropriate settings for testing.

Test Executor- executes pre-planned tests, documents and describes the faults found, and describes how to recreate (or correct) them.

The Test Manager- organizes and oversees meeting deadlines, keeping to a schedule, regulating test requirements, giving responsibilities to team members, and engaging with stakeholders are all examples of testing duties.

1.4 Methodology

Programming With JAVA

Java was first released in May 1995 and was created at Sun Microsystems by James Gosling as a core component of the Java platform. Java is an object-oriented programming language with minimal implementation requirements.

It's a general-purpose programming language that allows programmers to write once and execute anywhere, This means that compiled Java code can run on any platform that supports Java without requiring a recompilation. Java applications are often compiled to bytecode, which may execute on any Java virtual machine (JVM), regardless of the computer architecture.

Java's syntax is comparable to those of C and C++, but it offers less low-level features. Traditional compiled languages lack dynamic capabilities, which the Java runtime provides.

Javascript

This is a high-level ECMAScript-compliant, usually just-in-time compiled language. It has first-class functions, dynamic typing, and prototype-based object orientation. It supports event-driven, functional, and imperative programming styles and is multi-paradigm. It includes, among other things, APIs for working with text, regular expressions, text, dates, standard data structures, and Document Object Model.

Bootstrap

Bootstrap is a free CSS framework that allows for responsive web development for mobile devices. HTML, CSS, and JavaScript design templates for typography, form, button, navigational, and other UI elements.

SQL

SQL is a domain-specific language used to manage data in a relational database management system (RDBMS) or to process streams in a relational data stream management system (RDSMS). It's especially useful when working with structured data, which contains entity and variable connections. SQL offers two main benefits over earlier read-write APIs such as ISAM or VSAM. To begin, it introduced the idea of using a single command to retrieve several records. Second, it removes the requirement to specify how to retrieve a record, such as whether or not to use an index.

XML

The Extensible Markup Language (XML) is just a type of language and file format for storing, transporting, and recreating arbitrary data. It defines a set of guidelines for encoding materials that are both human- and machine-readable.

The design goals of XML emphasise on simplicity, universality, and cross-platform usability. It is a textual data format that supports numerous human languages with strong Unicode support. XML was created to represent documents, but it is now widely used to represent arbitrary data structures such as those used in web services.

JSON

JSON (JavaScript Object Notation) is an open platform file format and data exchange format that stores and transports data items made up of attribute-value combinations and arrays using human-readable language. It's a popular data format for electronic data transfers, such as between web applications and servers.

JSON is a language-independent data format. Although functionality to produce and interpret JSON-format data is included in many current computer languages, it was inspired by JavaScript. The extension.json is used for JSON filenames.

Spring Core

Spring is the most widely used enterprise Java application development framework. Millions of developers all over the world use Spring Framework to create high-performing, tested, and reusable code.

Spring is a free and open-source Java platform. It was written by Rod Johnson and released under the Apache 2.0 license in June 2003.

1.5 Organisation

“ABOUT-COGNIZANT”

The Domain on which cognizant works is “IT Consulting”, we could learn more about the company on its website is “www.cognizant.com”, currently company works under the guidance of “ Brian Humphries”.Revenue generated by the company could be figured as 1680 cr. USD which was calculated in 2019.

COMPANY'S MARKET PRESENCE

- | | |
|---------------------------------------|--------------------------------|
| 1. Auto | 2. Insurance |
| 3. Banks | 4. Life Sciences |
| 5. Capital Markets | 6. Manufacturing |
| 7. Communications, Media & Technology | 8. Oil & Gas |
| 9. Consumer Goods | 10. Retail |
| 11. Education | 12. Transportation & Logistics |
| 13. Healthcare | 14. Travel & Hospitality |
| 15. Information Services | 16. Utilities |

SERVICES PROVIDED BY THE COMPANY

Cognizant is a vast company that provides a variety of services like

- Security
- Digital experience and engineering
- Business & Enterprise services
- Intelligent Platforms
- QEA
- Artificial Intelligence & Internet of Things

There are more vast services which are provided by the company.

The Business Aligned Project provides real-time problem-solving experience using Agile methodology.

Cognizant has teamed up with UdeMy to provide world-class instructional videos for tomorrow's dynamic workforce. These UdeMy courses are organized into a learning path, allowing you to plan and learn according to your preferences.

The app also connects you with SMEs so you can get expert help on your questions about your learning path.

COGNIZANT IN INDIA

On January 20/1994, "Cognizant" opened a branch in Chennai, Tamil Nadu, India, under the formal title "Cognizant Technology Solutions India Private Limited." Among the cities served are Bengaluru, Chennai, Coimbatore, Gurugram, Noida, Hyderabad, Bengal, Mumbai, and Pune.

Cognizant Ind is one of Cognizant Technology Solutions' most well-known global distribution centres, with a focus on market outsourcing, consulting, and IT-related services.

ABOUT COGNIZANT ACADEMY

By allowing associates who will make it happen, Cognizant Academy contributes to Cognizant's mission of becoming "fit for development."

The organization's self-developed learning and development arm, Cognizant Academy, is committed to preparing employees for the future by providing role-based and business-aligned learning solutions.

During guided, self-study hours, the Generation Cognizant curriculum promotes more self-directed learners to attain their learning goals and prepare for hands-on hours at their own pace and style. Millennials can network with subject matter experts (SMEs), comprehend the corporate environment, and take care of themselves thanks to Gen C training programmes, which provide a holistic learning route for young talent.

Learner Autonomy is a concept promoted by Cognizant in which students take charge of their own education by utilising the tools and resources available to them.

The emphasis is on "learning" rather than "teaching." Prepare to start on your own learning trip.

PROGRAM -HIGHLIGHTS

- "The entire learning path is structured using best learning principles, with problem solving and applying newly acquired abilities taking precedence over conceptual learning".
- "Learners autonomy is emphasized through Flipped Classroom, where the learning platform provides world-class learning resources and students are not bound by an instructor's supervision".

- “Get mentored by a subject matter expert, whose motivation and assistance will help you expedite your learning journey”.

During the guided self-learning hours, you will be encouraged to be more autonomous learners, completing the learning objectives at your own pace and preparing for the hands-on practise time.

- **Time allotted for Self-Training**
 - You have to go through the learning courses yourself.
 - Try to complete learning tasks like quizzes and different assignments.
- **Time allotted for Practise**
 - In this time you practice under SMEs.
 - Queries are being taken out.

LEARNING JOURNEY

The learning journey is divided into two parts, each followed by a Business Aligned Project.

Stage 1 – QEA Basics

- Funtional-Testing
- Automation Concepts Basics
- Java
- WebUI
- Data Source(SQL, XML, JSON)

Stage 2 – Selenium with Digital Technologies

- Spring Core, Spring Boot and MVC
- Spring Restful Webservices
- Microservices
- Basics of WebDriver
- Project Deliverables
- Automation Techniques and Dynamic XPath
- Project Deliverables

The Business Unit provides real-time problem-solving experience using Agile methodology.

Cognizant & Udemy has partnered to create world-class instructional videos for the growing future of work. These Udemy modules are intertwined into a learning path, allowing you to plan and learn at your own pace.

The application also connects you with Subject Matter Experts to seek professional advice on your learning journey questions.

The program constantly assesses your ability to apply self-learned skills to solve business problems. The three major learning components, which are scattered throughout the learning journey for continual evaluation, are depicted here.

SCHEDULING OF LEARNING COMPONENTS

This curriculum will last 12-16 weeks in total. Every day of the program, beginning on day one will have some learning components that the GEN C must accomplish.

The day-by-day timetable will be included in the complete student handbook. All learning components must be completed in the order stated in the daily timetable. According to the schedule, these components will be automatically arranged in your unique learning paths.

INNOVATIVE LEARNING MODELS IN 2022

The Generation Cognizant programme includes campus workers' complete learning path, from the moment a firm accepts a proposal until their first year with the organization. At each level, a dependable training process is utilized.

- Modern Upside Down Classroom
- Campus Events
- Interactions with Branding Leaders
- Warmth Webinars and Workshops Active Volunteering

CHAPTER NO. 2

Literature Survey

Survey on Software-Testing Practices

Jihyun Lee, Sungwon Kang

Department of Computer Science, Korea Advanced Institute of Science and Technology

Abstract

Software testing techniques and tools (STMTs) must be used successfully and efficiently to generate high-quality software. We surveyed a wide range of software companies and industry experts.

STMTs are being assessed in order to identify present practices and areas where they can be improved. The findings of the poll there were 5 major results involving STMTs' existing methods and areas for growth.

“STMT” utilisation rate, issues caused by a lack of STMTs, and limited use of testing tools Interoperability support between software research and deployment procedures and tools are in great demand, as is guidance on evaluating and explaining "STMTs". capabilities. These findings, along with other survey-related data, will aid in the improvement of STMT procedures and the development of software testing tools.

1. Introduction

Software testing techniques comprise methodologies, procedures/patterns/templates, and other tools used to carry out software testing operations in an effective and efficient manner. Examples include metric-based test estimation procedures and black-box or white-box approaches.

Approaches for designing test boxes, static testing techniques, and so on. Software testing tools are software components that are used to test software.

Automated or semi-automatic assistance with software testing procedures and processes such as dynamic testing tool analysis, test design, and coverage analysis Software testing tools are

intended to be used with certain applications and methods for testing and automating them entirely or partially.

According to several software testing research, existing software testing techniques are far from acceptable.

The state of testing of software research and its problems, and reviews of testing methodologies and tools, as well as instruments required for testing. They claim that sophisticated tools are still required.

There exist gaps between testing research and industrial processes, and construct and test are seamlessly integrated.

They offer the potential for improving test procedures through the process, testing approaches and tools.

In particular, report on the components of testing methodologies employed, testing activities conducted, and automated testing activities with the hurdles to adoption in Australia.

There have been analyses to discover which testing strategies and processes are employed, as well as which approaches and instruments are popular. Analyze software testing methods to find characteristics that impact the testing process.

Previous research looked at the current state of software testing methods and tools (STMTs) and the barriers to STMT adoptions; Little is known, however, regarding the present status of "STMT" practices in terms of capabilities, restrictions, advancements, and requirements.

It is necessary to identify which two "STMT" skills should be strengthened in order to ensure testing efficiency and effectiveness. Therefore, we performed a survey of software testing businesses and experts to determine current methods.

In addition, there are opportunities for promotion in "STMTs." The survey findings must be used to determine if STMT capabilities meet international standards. The survey's focus is not on a single issue.

In the industry, methodologies or tools are utilized, but in the software testing process and activities that are supported by the procedures and instruments. The survey asked respondents which software testing processes and activities are effectively supported and which are not, with the goal of investigating as-is practices, perceived limitations of STMTs, and needs for STMT capabilities.

This paper is organized as follows: Section 2 describes the structure of the survey and the demographic groups participating. Section 3 shows the results of the survey analysis. One section of the survey describes the threat to effectiveness and Section 5 presents the results and conclusions.

Survey's design and its implementation.

Questionnaire 1

Respondents' key business areas are software we built, according to background inquiries for firms. covers approximately 50 questions and it is separated into three sections: "basic corporate" "information:" and "testing environment", present practices and perceived deficiencies of STMTs, and demands for increased STMT capabilities. The first section of the questionnaire asks for general information about the firm or the survey respondents. The second section inquires about the software testing techniques used by the organization. The queries in this part address the percentage of the testing budget, software testing issues, and the amount of test automation. The final section, based on "some thesis," offers questions about STMTs' current practices in terms of test procedures and activities.

The activities chosen for testing are more refined than those used in previous studies. As an illustration,

In order to determine which activities in the test-design process is assisted by systematic "STMTs", we conducted a survey.

To investigate whether STMT properties that aid in test design should be enhanced in order to boost their use we also created a survey for software testing expertise. It has the same basic structure as the Questionnaire for businesses The questionnaire uses a five-level rating system ranging from "strongly agree" to "strongly disagree," including "don't know," or from "not relevant" to "extremely relevant," with the exception of STMTs issues related to users usage. Companies whose CIO or CTO names were on the Fortune 1000 website were eligible for the survey. There were 73 people in total, with 14 legitimate responses. Furthermore, members of the ISO/IEC Study Group contacted companies with a considerable market share in their countries, with a target population of 19 and a response rate-of '19'. Through the combined amount, 33 businesses took part in the business survey, resulting in a response rate of 36% as measured by "AAPOR.". We removed survey responses from companies that target the domestic market to lessen the risk. As a consequence, 09 statements were eliminated, leaving only Twenty-four for examination.

3- "(33%) finance/ (15%) electronics /(15%) telecommunications /(7%) transportation (4%) and other (26 percent)".

Other commercial areas retail, packaging, insurance, biotechnology, the military, and automobiles are just a few examples. All res-ponders develop software for use in corporate operations or as a component of finished goods. The majority of survey respondents were

CTOs, CIOs, project managers, or quality control engineers. Twenty of the 50 invited ISO/IEC JTC1/SC7 members participated in the expert survey. 59 percent of those polled work in education, 24 percent are consultants, and 17 percent are managers or project managers. Total Eighty-three percent of respondents have at least ten years of software development experience, and forty-seven percent have at least ten years of software-testing and experience.

.3. Analysis of the survey responses

This Part Summarizes the study respondents' replies. A part of survey discusses the variations in software testing between businesses that use “CMM/CMMI Level 3” and higher and those that use “Level 2” and lower. Section 3.5 discusses the variations in software testing between organizations with key items and businesses examines discrepancies in expert and industry perspectives on STMT.

3.1 Testing-environments

Respondents to the poll were asked to describe their testing situations. First,, in terms of budget and respondents reported that software testing consumes 24 percent of their total budget on average. This conclusion is consistent with Ng et al. [10] and “Kasurinen et al”.

But not as much as Juristo et al. (More than half of the time). Despite this, nearly half of those asked answered.

They have no knowledge how much money is spent on each software testing assignment because they have no understanding how much money is spent on each software testing task. In order to quantify the complexity of software testing, respondents were also asked to describe the level of application for each of the four barriers. The majority of respondents believe that software testing issues are caused by a lack of support from STMTs and testers' incapacity. Other impediments, on the other hand, received a lower rating than test processes. This backs up Ng et al. [10] and Grindal et al. [11] findings.

Second, respondents were asked about test automation's current state. Approximately 74 percent of practitioners indicated that software test automation is less than 50%, while only 16 percent claimed that it is greater than 50%.

The number is lower than the 68.2 percent reported by Ng et al. The majority of practitioners, among other things, did not automate their test estimating, test planning, or test design processes. It is depicted that the average amount of test automation for each of the software testing techniques.

3.2 Usage of STMTs

The usage of tools was far less popular than the use of testing methods. STMTs were implemented in defect management and reporting procedures more frequently than in other activities. Respondents responded that they are satisfied with the current STMT support, but that it is restricted and that additional analysis on related STMT capabilities is needed. Survey participants specifically suggested that capabilities such as interoperability between software development tools and software testing tools, inter-operability of management tools and testing tools (for illustration, "coordinating between quality-assurance and testing-results"), and interoperability of management solutions and testing tools; Improved automated test case generation and seamless show trial management must be implemented during execution. The goal of this study is to identify the capabilities of widely used STMTs that need to be developed or improved in order to learn how STMTs may help with software testing. Four times, participants were asked which software testing activities they utilize STMTs for. They had to pick one or more activities from the list for each testing phase and explain how they employed STMTs in situations where they weren't stated. The responses are summarized in Table. The first column shows processes, with a row of activities and a utilization rate for each activity for each process. Unit tests, integration tests, system tests, and acceptance tests are used to demonstrate test design and execution procedures' activities and utilization rates. The goal of this study is to find out what characteristics of STMTs are regularly used and also what features need to be created or improved so that STMTs can aid with software testing.

Respondents said they are satisfied with the current support for STMTs, but that it is restricted and that additional study on related STMT capabilities is needed. Respondents also specifically suggested that capabilities such as interoperability between software development tools and software testing tools, inter-operability between management solutions and testing tools, better automated test case generation, and better seamless management of test cases during execution be put in place. The following subsections outline how each test process applies test scenarios.

3.2.1 Test estimation

According to 14% of those questioned, they do not use any software testing methods. Firms utilize test estimating methods to estimate test efforts, test durations, and the quantity of test cases, although STMTs are rarely used to estimate prices. When it came to tool usage, test effort and cost estimation were rather low. They said they usually estimate tests manually, relying on their own knowledge or skills rather than standardized recommendations.

According to the test estimate row in the below table, 57% of participants replied that they do not utilize any test estimating tools. Nearly half of method users use tools. Below Table shows how STMTs are used.

Test Estimation		Activities	Estimating test effort	Estimating costs involved	Estimating test schedule	Estimating number of test cases			Do not use	
	Usage Rates	Methods	57%	38%	57%	48%			14%	
		Tools	24%	19%	33%	33%			57%	
Test Planning		Activities	Determining test items	Identifying testing-related risks	Determining a test schedule	Determining testing approaches	Defining test deliverables	Defining testing environment	Do not use	
	Usage Rates	Methods	57%	38%	62%	52%	57%	43%	14%	
		Tools	19%	0%	5%	19%	10%	5%	52%	
Test Design		Activities	Specifying test design	Specifying test case	Specifying test procedure	Generating test cases	Generating test scripts	Documenting test environment requirements	Do not use	
Test Levels	Unit Test	Usage Rates	Methods	33%	38%	24%	48%	29%	10%	29%
		Tools	10%	24%	14%	33%	19%	5%	48%	
	Integration Test	Methods	52%	71%	48%	57%	43%	29%	10%	
		Tools	19%	24%	19%	19%	29%	10%	43%	
	System Test	Methods	48%	71%	57%	71%	57%	43%	10%	
		Tools	33%	33%	38%	43%	48%	14%	19%	
	Acceptance Test	Methods	24%	38%	38%	38%	33%	29%	33%	
		Tools	10%	14%	10%	19%	14%	10%	48%	
Test Execution		Activities	Executing tests	Recording test log	Recording test incident report log				Do not use	
Test Levels	Unit Test	Usage Rates	Methods	43%	29%	33%				38%
		Tools	33%	33%	38%					29%
	Integration Test	Methods	57%	52%	67%					14%
		Tools	48%	43%	52%					19%
	System Test	Methods	62%	62%	71%					24%
		Tools	57%	57%	62%					14%
	Acceptance Test	Methods	43%	33%	48%					33%
		Tools	38%	33%	43%					38%

(Survey table part 1)

Defect management		Activities	Recording defects	Tracking defect status	Preventing defects	Resolving defect	Reporting defect resolution results	Analyzing and reporting defect information	Do not use
	Usage Rates	Methods	67%	67%	33%	52%	62%	52%	14%
		Tools	67%	76%	24%	43%	57%	52%	10%
Test reporting		Activities	Writing test summary report	Documenting anomaly report	Controlling test	Tracking test progress	Managing testware		Do not use
	Usage Rates	Methods	57%	52%	48%	57%	62%		24%
		Tools	52%	38%	48%	62%	57%		24%

(Survey table part 2)

3;2.2- Test planning

Nearly 50 % of the respondents stated that they do not use any exam preparation materials. while the other half (50%) said they use methods for the majority of their test-development and planning. Table 1's test planning row shows that tool usage is relatively low when compared to method usage. In software testing, risk management, scheduling, and environment definition tools are rarely employed. Because of the nature of planning, automating activities is difficult. Test plans, on the other hand, govern the entire software testing process; thus, test planning skills should be employed.

3.2.3- Test design

When asked to choose activities from each test level 6 that are carried out using STMTs, as shown in the test design row of Table 1, 30% chose unit test methods, the lowest of the test levels, and 58 percent chose system test methods, the highest. More than half of those polled

said they utilize procedures to generate and create test cases, whereas only half said they use tools. The tool utilization rate for creating test cases/scripts is similar to that is reported by

3.2.4- Test execution

The questions on test execution were asked in the same manner as the questions about test design. The adoption rate is somewhat greater than in the test design. The results are lower than Ng et al [10] .s stated 79.5 percent. On the other hand, the test execution rows in Table 1 show that the gap between STMT and test design utilisation rates is closing. We discovered that, while test execution methodologies were commonly used in integration and system test activities, tool adoption was low. STMTs are not used in the design or execution of unit tests, according to the vast majority of respondents.

3.2.5 Defect management

According to the Table, the majority of practitioners use STMTs to handle defects. Only about ten percent of those polled said they don't use any STMTs for defect management. A large number of practitioners, according to Table 1, use "STMTs" for failure recoding, fault tracking, and defect reporting. STMTs were used substantially less for defect prevention than in the other defect-related areas.

3.2.6 Test reporting

According to the test reporting rows in Table 1, nearly 50% of practitioners utilise STMTs for test reporting. STMTs are utilised for fault recording, tracking, and reporting 65 percent of the time. It is comparable to the 61 percent reported by Ng et al. in [10], despite the fact that the survey did not deconstruct activities. Descriptive technologies are used less typically than other types of tools. When compared to other software testing techniques, however, the diversity in STMTs utilisation rates is minor. The majority of respondents reported being extremely happy or satisfied with the STMTs they now use.

Survey finding & conclusions-

The survey's findings and conclusions are: The goal of the research was to look into the current state of software testing practices in terms of "STMTs." Furthermore, we aimed to discover "STMTs'" "long-term plan for aiding software development organizations in efficiently and effectively improving and integrating existing software testing practices." The results of the survey are described below: "STMTs" in particular are rarely handled. In order to maximise the prospects of improvement, our survey questions were developed in greater depth than prior surveys..Tool usage was found to be substantially lower (less than 30%) than method usage, with the exception of defect management (approximately 50 percent). This shows that the STMTs now in use are not sufficiently developed, restricting their applications. As a result, specifying "STMT" capabilities is essential to integrating processes, methodologies, and tools so that they might complement each other's responsibilities, thereby improving "STMT" adoption and helping to successful and efficient software-testing. Furthermore, leading to a shortage of "STMTs", responders are experiencing troubles with software testing. This agrees with the findings of "Ng et al." Who has discovered that a lack of support tools was a major barrier to method acceptance, whereas tool complexity was a barrier to tool acceptance. This indicates that there is a significant gap between the capabilities of currently used STMTs and industry expectations, or that STMT capabilities are not widely recognised. STMTs' basic characteristics can help users evaluate and choose appropriate STMTs, as well as provide the bare minimum of tool capabilities for supporting operations.

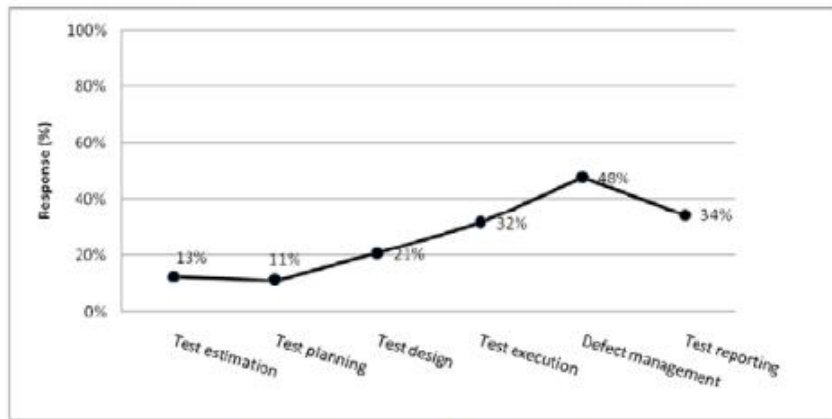
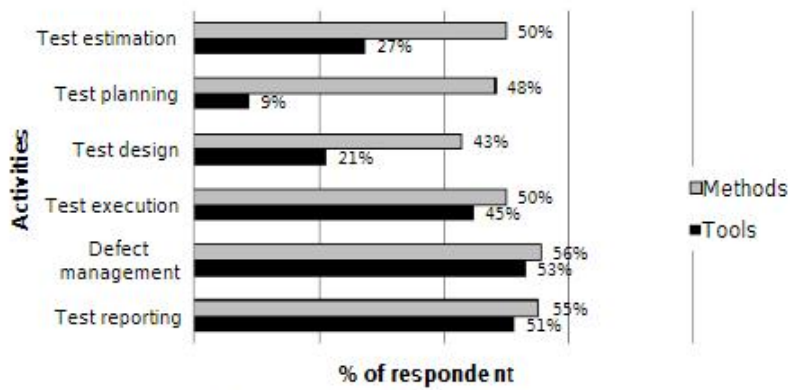
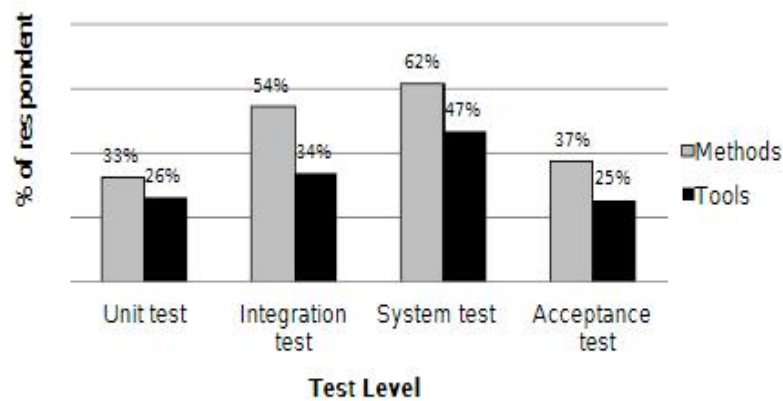


Fig. 1. Level of test automation



(a) Usage of STMTs in test processes



(b) Average usage rate of STMTs at test levels

Fig. 2. Usage of STMTs

CHAPTER NO. 3

System Development

Understanding Java

Java is a very valuable programming language made up of class-based objects and is designed to minimize reasonably foreseeable conditions of execution. Application engineers must assume that structured Java code can be run at any stage and need to run once, run anywhere (WORA) with Java support without the need for recompilation.

REAL-WORLD APPLICATIONS

Java has various practical applications, ranging from online trading sites to Android applications, from logic applications to currency-related applications such as electronic exchange systems, from games like Minecraft to workspace programs like Eclipse, Netbeans, and IntelliJ.

JAVA-VIRTUAL-MACHINE

The Java-Virtual-Machine is an execution engine that allows the deployment of Java code & applications. To machine language, convert Java code. The "Java-Virtual-Machine" (JVM) is a piece of artwork (JRE). Machine code is generated by compilers in various computer languages. Virtual machine code is generated by the Java compiler in any scenario.

OBJECT-ORIENTED-PROGRAMMING SYSTEM

The Object-Oriented Programming System (OOPS) is a method of creating "objects" that contain data and functions. Creating flexibility and a common sense of conduct is an important component of PC software development. It integrates data and its behavior (techniques) into a single region (object) and simplifies the understanding of how a programme operates.

CONCEPTS OF OBJECT-ORIENTED PROGRAMMING SYSTEM

Class: This is a union of closely related elements. I have a class called Fancy Cars, for example. The cost or speed of these vehicles might be a characteristic (data). The technology can function with these vehicles, but they move or slow down.

Objects: An object can be stated as an evolution of a class, and the class can have various adaptations in a program. Seats, bicycles, markers, pencils, tablets, automobiles, and so forth.

Inheritance: Classes can share, acquire, or "acquire" the attributes and methods of other classes. This allows you to reuse existing code and save on coding time. A derived class or lad class is one that you inherit from someone else. Base classes or superclasses are classes that share attributes and techniques.

Polymorphism: Indicates the possibility of using the strategy in a different way. That is, it can take on different structures in different cases (poly + morph). There are two kinds of polymorphism: compile-time polymorphism and runtime polymorphism. When the technique is overloaded, compile-time (static) polymorphism occurs. That is, the point at which the clashes used in the strategy change. This is done to get different results. In this case, a different name is proposed for the president of the country, giving different results each time, but will be known as the president anyway. Runtime (dynamic) polymorphism occurs when the method itself changes. We need to change our current strategy when we don't need to touch the prime minister and not the president.

Indicates the possibility of using the strategy in a different way. That is, it can take on different structures in different cases (poly + morph). There are two kinds of polymorphism: compile-time polymorphism and runtime polymorphism. When the technique is overloaded, compile-time (static) polymorphism occurs. That is, the point at which the clashes used in the strategy change. This is done to get different results. In this case, a different name is proposed for the president of the country, giving different results each time, but will be known as the president anyway. Runtime (dynamic) polymorphism occurs when the method itself changes. We need to change our current strategy when we don't need to touch the prime minister and not the president.

Abstraction: Discussion of information gives hints on procedures that simply display the properties and methods applicable to working with items while hiding everything else. Informational thinking allows you to (obviously) reduce the complexity of your programs, and there is a tremendous amount of room offered by classes of OOP dialects.

Encapsulation: Incarnations give you hints on how to store objects with strategy in one place. We also ensure the validity of the information. Ideally, you restrict access to the information by hiding it from external components, preventing it from being changed unnecessarily.

INTERFACES IN JAVA

Like classes, interfaces can have methods and elements, but the strategies spoken by interfaces are inherently unique (only strategy signatures, no bodies).

1. If a class implements an interface and does not provide a strategy body for all the privileges assigned to the interface, the class should theoretically be declared.

2. Java Library Model - Comparator Interface. This interface is often used to sort categories if a class implements it.

EXCEPTION HANDLING

Java's exception handling is one of the most powerful tools for dealing with runtime failures, and it may help with the overall development of your programme. The throwable “java-lang” class is the parent class of the Java-exception sequence, which is descended from its two subclasses Exception and Error.

Exception type:

1. Built-in exception
2. User-defined exception

JAVA EXCEPTION KEYWORDS

There are various exception keywords in Java that are used to perform various operations like

- Try-Catch Block: This is a pair of keywords that helps reduce run-time exceptions. Try helps you execute a code that gives an error and throws exceptions catch catches those exceptions and prevents code failure.
- Throw: This lets you throw a run-time error.
- Throws: This is a java class that helps in handling run-time exceptions and is implemented to gain the ability to throw exceptions.

STRINGS

A string is defined as a collection of characters. String instances in Java are forever, which means they cannot be changed once generated.

1. “char charAt(int list)”- A character from a predetermined str is restored.
2. “int compareTo(Object obj)”- This is used string is compared to another item.
3. “int compareTo(String AnStr)”-It compares two alphabetical strings.

4. “int compareToIgnoreCase(String s)”- Checks the two strings in alphabetical order, with case differences ignored.
5. “String concat(String str)”:- it adds up two strings.
6. “boolean equals (Object anObj)”-this compares string and return boolean
7. “int indexOf(int ind)”- This function returns a list of the significant events of a preset symbol inside this line.
8. “int indexOf(String str)” :-Helps restore a file from a preset substring inside this main event string.
9. “int lastIndexOf(int var, int fromInd)”- Look backwards at the beginning of the predefined item and restore the file within this line of the last event of the predefined symbol.
10. “int lastIndexOf(String str)”:- Look for files containing the rightmost event of the provided substring within this string.

StringBuffer

The StringBuffer class in Java represents a mutable string. In Java, the StringBuffer class is identical to the String class, however, it is not persistent. This, for example, can vary dramatically.

- StringBuffer():- This has an unfilled limit of 16.
- StringBuffer(String str):- It helps in creating support for pre-determined string.
- StringBuffer(int L):- This has an unfilled limit of “L”.

WORKING WITH ARRAYS IN JAVA

An array could be referred as a container object that houses a set number of the same kind of classes. The size of the array is determined during the exhibit's development. The length is fixed after construction. All exhibitions are prominently displayed in Java.

Exhibits can contain distinct types of information, such as class objects, based on cluster values. When the raw type of information appears, the actual quality is stored in a contiguous memory area. When a class object is created, the genuine item is placed in the Downloads section.

COLLECTIONS IN JAVA

Collection framework in Java provides an architecture to handle and manipulate objects. It has implementations for various data structures, which have very different use cases some are faster for storage or retrieval others store data in a certain way and so on. Java collection is a bundle of single objects. It provides many interfaces like Set, List, Queue, HashSet, etc.

ARRAYLIST

ArrayList class in java is a resizable array which very similar to vectors in C++. It is implemented in java.util package. ArrayList provides many different functions for manipulating ArrayList from adding elements, removing elements, sorting elements.

JDBC With MySQL

MYSQL DATABASE CONNECTION IN JAVA

To connect Java applications to the MySQL database, we must make the following five steps.

MySql is used as the database in this model. As a result, we must know the following things about the MySql database:

1. Driver class: com.mysql.jdbc.Driver is the driver class for the MySql database.
2. Connection URL: The URL for the MySQL database association is "jdbc:mysql://localhost:3306/sonoo," where JDBC is the API running, but we may alternatively use the IP address. Any database may be used; simply replace "sonoo" with the name of our database.
3. Username: The MySQL-database standard user's name is "root".
4. Password: This is the secret phrase provided by the client when introducing the MySQL database.

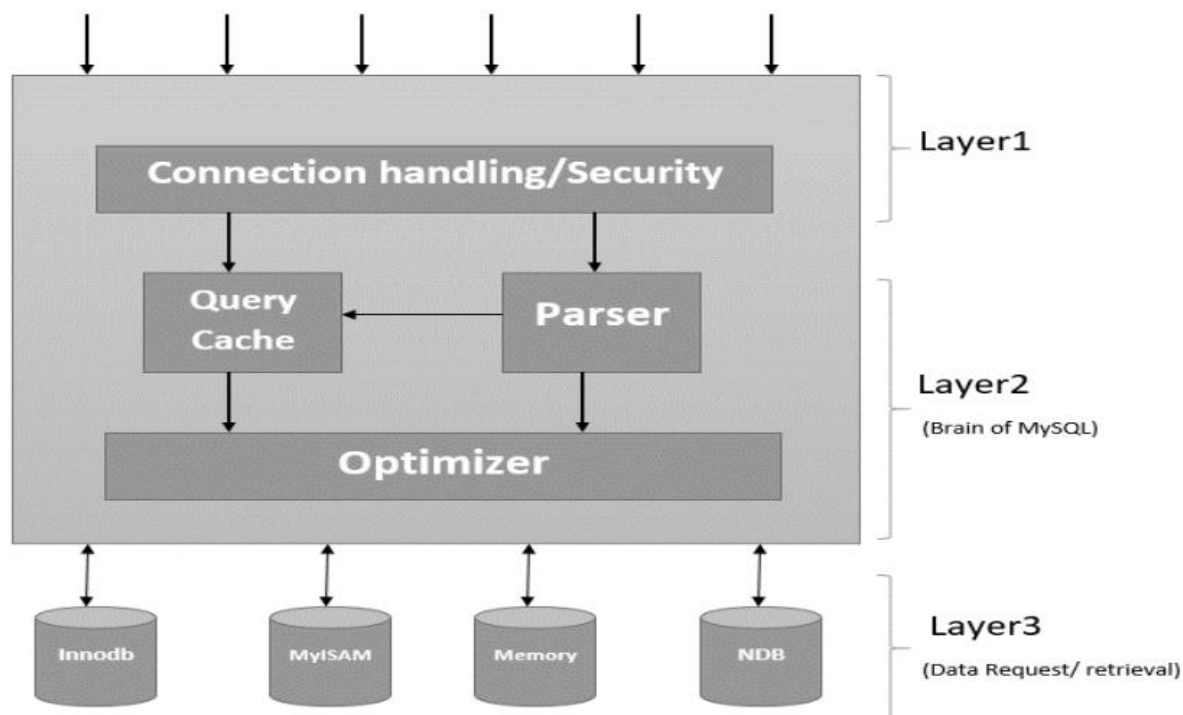
MYSQL INTRODUCTION

SQL stands for "Structured Query Language". A common language for interacting with databases. SQL is a fantastic programming language used to interact with databases. It works by capturing and analyzing an information base with information fields in tables.

MySQL is a community database that also serves as the board shell. It provides us with a UI for connecting to the database.

The programme is used for a variety of tasks, including data warehousing, online trading, and logging. Despite this, MySQL's most prevalent application is as an online information

base.



STEPS FOR INSTALLATION OF MYSQL

Step 1: The first step is to go to www.dev.sql.com and enter MySQL for the required working environment.

Step 2: After connecting MySQL to the system, start the workbench where you can create queries.

Step 3: As soon as you open the workbench, a password summary will appear and you will need to enter the root password you created during the setup procedure.

Step 4: You can then create your first database in your production environment.

RELATIONAL DATABASE MANAGEMENT SYSTEM

A relational dataset is a dataset that stores statistics in well-organized groups with lines and segments. As a result, clear qualities can be easily found and incorporated into the parts database. The social design allows you to quickly spread doubts to unrelated tables.

XPATH

Xpath is a query language or syntax that may be used to locate a node or collection of nodes in an XML or HTML document. The most important aspect of automation script development has always been the localization of web elements. Finding the right, efficient and accurate tracking device has always been a source of frustration in the automated test development process. This caused QA engineers to consider locators other

than ID, Name, Class, Link, or TagName. XPath has long been the QA's preferred locator, especially for finding dynamic elements. XPath Selenium provides many xpath and axis (relationship) features that allow you to write efficient XPaths for web elements and create your own layouts for web parts.

We can create XPath using

1. Using Attributes like name, tag, etc.
2. You can use Dynamic elements.
3. You can combine text & attributes.
4. Using various relationships like parent child or ancestor descendant.

ABSOLUTE XPATH VS RELATIVE XPATH

RELATIVE XPATH

This entails searching the entire document using a specified attribute, text, or relationship to provide a search when unique text, attributes, or other elements are not available. As mentioned in the last section, there are several ways to find an element using a relative XPath.

ABSOLUTE XPATH

This method requires that you provide a starting point for the root and provide all children and subtags accurately, without omissions. This could be dangerous in the future. This is because webpages are dynamic and not co-located, so elements cannot be detected.

IN REAL TIME COMPARISON OF ABSOLUTE XPATH AND RELATIVE XPATH FOR USE ON A WEB PAGE

It is not recommended to use Absolute XPath in real-time applications or web pages with dynamic elements that change, as you must provide accurate information about elements when using Absolute XPath, otherwise you may have big problems.

As a result, relative XPaths are more commonly used to find or test components of dynamic web pages that exist in real-time environments.

FUNCTIONALITY OF XPATH

Working in a changing online environment makes it tough to locate a specific web part with common features such as name, class, and so on. Multiple components with the same characteristic, such as the same name type or class name, can exist. Existing XPath routines may be inefficient since the underlying XPath may yield several elements. To address these problems, XPath Selenium includes an XPath feature for

developing efficient XPath expressions that uniquely identify things.

CSS SELECTORS

Cascading style sheets are a way to determine how elements are displayed on a web page. CSS selectors are essentially a combination of element selectors and selector values that can be used to identify specific elements on the page. You can use CSS selectors like XPath to search for web parts without an ID, class, or name. HTML tags, attributes, identifiers, and class strings.

Finding an element on a web page can be difficult. Especially in today's environment where all other online pages are designed dynamically and consequently the web parts that are developed/rendered must also be done dynamically. Defining static properties on dynamic elements is difficult. In most cases, these elements do not have consistent attribute values. Therefore, you cannot directly use locators for identifiers, names, references, sub-references, etc. You must first locate the required web elements before writing UI-based automation test scripts, particularly Selenium automation scripts. The CSS selector comes in handy here. This is due to CSS's ability to recognise the majority of web components on a page. It can also recognise objects with non-constant attribute values, making it ideal for any automation designer. It's regarded one of the quickest methods to discover an element on a web page since it's closer to code. Let's look at how to build CSS selectors for web components syntactically.

So you can apply styles, perform actions, and gather data for testing purposes, essentially as a way to find elements.

Spring Framework Overview

Spring simplifies the development of enterprise Java applications. It provides everything which we need to use Java in a corporate setting, including Kotlin support as Java Virtual Machine fallback languages and the ability to develop different architectures based on your application's demands. Spring Framework needs Java development kit 8+ (Java SE 8+) and includes Java development kit 11 LTS support. Although Java SE 8 Update 60 is the minimum fix release for Java 8, it is usually recommended to use the most recent fix release. Applications can pick and choose the modules they need. The primary container module, which includes the configuration model and the dependency injection mechanism, serves as the system's basis. The Spring Framework also includes native support for messaging, transactional data and persistence, and web application architectures. It also contains the Spring MVC servlet-based web framework and the Spring WebFlux reactive web framework.

“Unit-Testing” using Spring

As we compare to traditional “Java EE” development, dependency injection should have reduced your software's reliance on the container. To test the POJOs in your application, use JUnit or TestNG. with objects produced with the new operator rather than Spring or another container. Mock objects may be used to test your code by isolating it (along with other valuable testing techniques). For example, when running unit tests, you may test service layer objects by stubbing or emulating DAO or repo interactions.

Environment

Mock implementations of the Environment and PropertySource abstractions are included in the spring package. Out-of-container tests for code that relies on environment-specific attributes may be written using MockEnvironment and MockPropertySource.

Servlet API

Several Servlet API mock objects are included in the "org.springframework.mock.web" package for testing web contexts, controllers, and filters. These fictitious objects are intended for use with Spring's Web MVC framework and are more user-friendly than dynamic mock objects (such as EasyMock) or alternative Servlet API mock objects .

Spring Web MVC

Spring Web MVC, the first web framework built on the Servlet API, has always been included in the Spring Framework. Spring Web MVC's official name is derived from the name of its core module, although it is more typically referred to as "Spring MVC."

DispatcherServlet

"Spring MVC", like many other web frameworks, is built from the front "controller architecture", with a centralised-api, The “DispatcherServlet” is an api that offers a standard mechanism for request processing while the actual work is performed by customizable delegate components. This method is adaptable to a wide range of operations.

The following Java setup example registers and initialises the DispatcherServlet.

```

public class MyWebApplicationInitializer implements WebApplicationInitializer {

    @Override
    public void onStartup(ServletContext servletContext) {

        // Load Spring web application configuration
        AnnotationConfigWebApplicationContext context = new AnnotationConfigWebApplicationContext();
        context.register(AppConfig.class);

        // Create and register the DispatcherServlet
        DispatcherServlet servlet = new DispatcherServlet(context);
        ServletRegistration.Dynamic registration = servletContext.addServlet("app", servlet);
        registration.setLoadOnStartup(1);
        registration.addMapping("/app/*");
    }
}

```

The following "web.xml" setup example registers and initializes the DispatcherServlet:

```

<web-app>

    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/app-context.xml</param-value>
    </context-param>

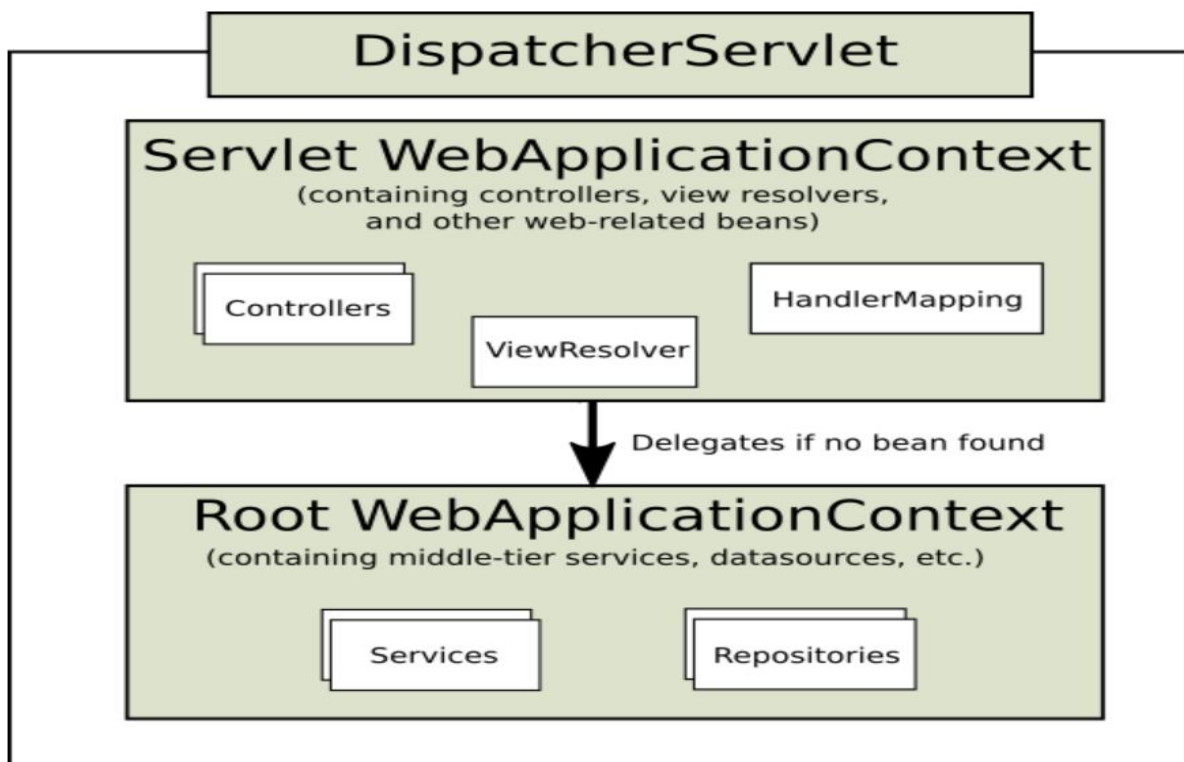
    <servlet>
        <servlet-name>app</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value></param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>app</servlet-name>
        <url-pattern>/app/*</url-pattern>
    </servlet-mapping>

</web-app>

```

Context Hierarchy



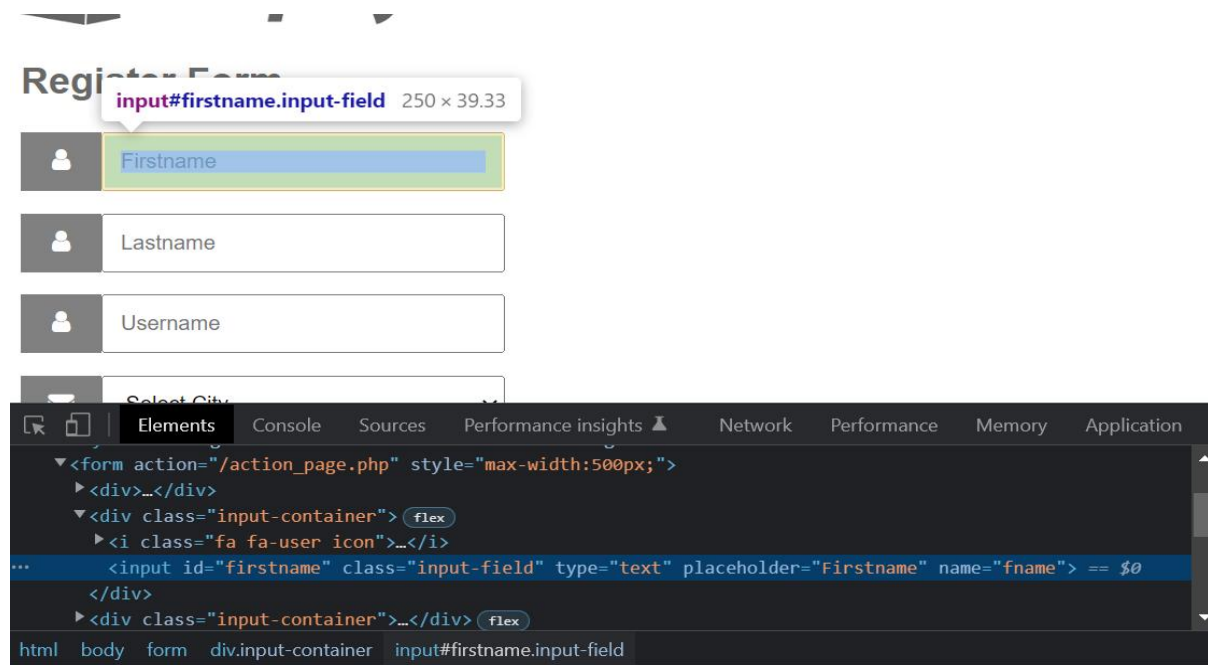
CHAPTER NO. 4

Performance Analysis

Exp-1

Objective:

To understand the basic selenium operations like invoking the driver, identifying the web element by name and getting the specific attribute of the web element.



```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.By;

public class NameLocator
{
    String fName;
    static WebDriver driver;

    public WebDriver setupDriver()
    {
        driver=DriverSetup.getWebDriver(); /*Invoke the getWebDriver method from
    }
    return driver;
}
public String getNameLocator()
{
    /*Identify the Firstname
    Get the placeholder value
    Store the placeholder value in the static variable fName.*/
    driver.findElement(By.name("fname"));
    fName=driver.findElement(By.name("fname")).getAttribute("placeholder");
    return fName;
}
```

```

public static void main(String[] args)
{
    NameLocator namLocator=new NameLocator();
    String name=namLocator.getNameLocator();
    System.out.println("The name is "+name);
}
}

```

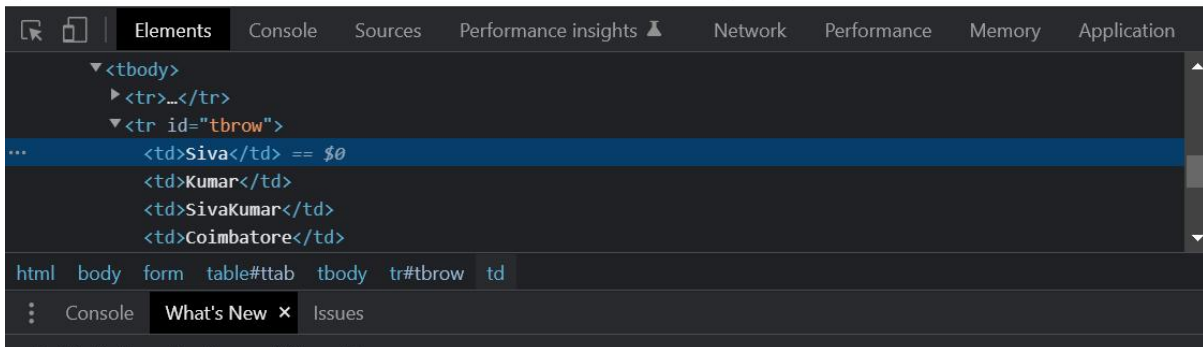
Exp 2

Objective:

To understand the basic selenium operations like invoking the driver, identifying the web element by id and getting the specific attribute of the web element.

Male Female Other

FirstName	LastName	UserName	City
Siva	Kumar	SivaKumar	Coimbatore



```

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;

public class PageLocator //DO NOT Change the class Name
{
    String lname;
    static WebDriver driver;
    static WebElement element;

    public WebDriver createDriver() //DO NOT change the method signature
    {
        //Invoke getWebDriver method from DriverSetup and return it
        driver=DriverSetup.getWebDriver();
        return driver;
    }

    public WebElement getPageLocator(WebDriver driver) //DO NOT change the method signature
    {
        WebElement lastName=driver.findElement(By.id("lastname"));
        lname= lastName.getAttribute("placeholder");
        return lastName;
        /*Find the element by id */
    }

    public String getName(WebElement element) //DO NOT change the metl
    {
        return lname;
        //Get the attribute value from the element and return it
    }

    public static void main(String[] args){
        PageLocator pl=new PageLocator();
        //Add required code
        String name = pl.getName(element);
        System.out.println("The last name is"+name);
    }
}

```

Exp-3

Objective:

To understand the basic selenium operations like invoking the driver, identifying the web elements and setting values for the basic text elements and Drop down list.



Register Form

	Firstname
	Lastname
	Username
	Select City
	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
	Password
Register	Cancel

```
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.By;  
import org.openqa.selenium.support.ui.Select;
```

```
public class Registration  
{  
    static WebDriver driver;  
    static String baseUrl;  
  
    public WebDriver setupDriver()  
    {  
        //Assign the value for baseUrl  
        /* Get the driver, and launch the app using get() with baseUrl */  
        baseUrl="http://webapps.tekstac.com/Shopify/";  
        driver=DriverSetup.getWebDriver();  
        driver.get(baseUrl);  
        return driver;  
    }  
}
```

```

public void setElements()
{
    /*Using the driver, Find the elements by id and send the values to the elements*/
    driver.findElement(By.id("firstname")).sendKeys("Mithali");
    driver.findElement(By.id("lastname")).sendKeys("Raj");
    driver.findElement(By.id("username")).sendKeys("Mithali Raj");
    Select city = new Select(driver.findElement(By.id("selectcity")));
    city.selectByVisibleText("Chennai");
    driver.findElement(By.xpath("//input[@value='female']")).click();
    driver.findElement(By.id("pass")).sendKeys("MR@123");
    driver.findElement(By.id("reg")).click();
}

public static void main(String[] args)
{
    Registration reg=new Registration();
    //Implement Code Here
    reg.setupDriver();
    reg.setElements();
}
}

```

CHAPTER NO. 5

Conclusions

Around the last few years, developers all over the world have been continually working on and developing applications in order to deliver better software and quality software to users. With the application of many approaches and ideologies that aid in development. Agile technique has emerged throughout the years to improve development and customer satisfaction.

Selenium is often regarded as the best solution for web-based application testing automation. Selenium has been the primary choice for automation and testing since its inception because of the popular languages it supports. Using Selenium is simple and should be considered if testing of web applications is to be done because the tests can be replicated on various machines and the test coverages provide extensive coverage that allows the developers to make robust and quality software within budget while also providing customer satisfaction and well-built products.

This internship involved software testing and testing automation using Selenium. The learning process was extensive, and the learning and assistance supplied were adequate for learning development for automation and testing. During this internship, selenium was not the only technology employed. This internship also made extensive use of a variety of technologies such as JavaScript, SQL, HTML, and CSS. Overall, the learning experience was incredibly fruitful, and the knowledge gained from it was extremely valuable.

FUTURE WORK

As software is developed, and the majority of them are web-based applications, testing becomes more important for the software to provide a reliable way to develop software and that makes a quality product, and because of the large number of applications with varying levels of complexity, automation is a perfect candidate for providing vital test coverage with speed and efficiency that makes a quality product. Software testing, like software development, is a vital component of the whole process of getting a product or service to market; therefore, automation of testing is the way to go in order to minimise time to market, boost software quality, and lower development costs.

REFERENCES

- [1] Devi et al., International Journal of Advanced Research in Computer Science and Software Engineering 7(5), May- 2017, pp. 855-862 © 2017, IJARCSSE All Rights Reserved
- [2] Itkonen, Mäntylä & Lassenius (2009)
- [3] Ashmore, S. & Runyan, K. 2015. Introduction to Agile Methods. Crawfordsville, Indiana, United States: Pearson Education, Inc.
- [4] Avasarala, S. 2014. Selenium WebDriver Practical Guide. Birmingham B3 2PB, UK: Packt Publishing.
- [5] Barab, S. & Squire, K. 2004. Design-based Research: Putting a Stake in the Ground. The Journal of the Learning Sciences, 13(1), 1 –14.
- [6] Bath, G. & McKay, J. 2008. The Software Test Engineer’s Handbook. Santa Barbara, CA: Rocky Nook Inc. Bentley, J.E. 2004.
- [7] Software Testing Fundamentals - Concepts, Roles, and Terminology, Wachovia Bank, Charlotte NC.
- [8] Crispin, L. & Gregory, J. 2008. Agile Testing: A Practical Guide for Testers and Agile Teams. Crawfordsville, Indiana, United States: Pearson Education Inc. 44 of 44
- [9] Saunders, M., Lewis, P. & Thornhill, A. 2012. Research Methods for Business Students. 6th ed. Harlow, England: Pearson Education Limited.
- [10] Schwaber, K. & Beedle, M. 2002. Agile Software Development with Scrum. Upper Saddle River, NJ: Prentice-Hall.
- [11] <https://cognizant.udemy.com/course/selenium-webdriverweb-based-automation/testing>
- [12] <https://cognizant.udemy.com/course/testerbootcamp>
- [13] <https://cognizant.udemy.com/course/javascript-basics-forbeginners>
- [14] <https://www.javatpoint.com/java-tutorial>
- [15] <https://www.geeksforgeeks.org/>