International Journal of Advanced Engineering Science and Information Technology (IJAESIT)

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

Scalable Mobile Automation Testing Frameworks for Government Digital Service Platforms

Mahendar Ramidi

Independent Researcher, USA

ABSTRACT: It must be ensured that automation testing frameworks can be used to test government mobile applications to ensure that the workflows are correct, accessibility standards, and cross platform behavior are met on a large scale. The paper presents a proposed scalable mobile automation testing framework, which can be used to test public-sector applications, consisting of UI automation, API-validation, accessibility testing, and regression coverage on both the iOS and Android platforms. This framework focuses on the maintainability, isolation of tests in the test environment, functionality of CI/CD pipelines, which give government agencies the ability to maintain a high rate of releases without affecting the compliance to reliability and security as well as accessibility. Besides, it also points to an Appium-based architecture that will provide the flexibility and scalability of testing. The strategy will make sure that the mobile application in the public sector can address the special needs of security, compliance and performance and sustain a sound testing ecosystem. This framework will provide the government agencies with an effective tool to test the automation to provide faster and more reliable deployment cycles and enhance the overall quality of mobile digital services.

KEYWORDS: Mobile Automation Testing, Government Digital Services, Appium Testing, Cross-Platform Testing, Accessibility Compliance, Regression Testing, CI/CD Integration, Public Sector Quality

I. INTRODUCTION

The government services have undergone digital transformation which has seen the spread of mobile applications that are aimed at increasing the engagement of the citizens, improving accessibility and simplifying the delivery of the public services. Government mobile applications include simple informational applications to complex systems that provide support to important services such as healthcare, transport, tax filing, and emergency management. These applications are the key in the quest by the public sector to modernize and enhance services delivery. But the quality, reliability, security, and accessibility of these mobile applications is a challenge that has to be continuously maintained. This becomes especially accurate because the government agencies have been employing faster releases to keep abreast with the changing requirements of the users without compromising on strict regulatory requirements [1] [2].

Automation testing is one of the most important means of keeping the standards of quality high. Automation testing is very important in making sure that the mobile applications can be used as per the requirements in different circumstances. Government applications are a particular concern, as they are likely to have a wide range of users and are likely to demand high accessibility, security, and privacy standards. Conventional manual testing approaches though effective are no longer scaleable with the speed of development and release cycles of the modern digital realm of the public sector. Due to this, automated testing frameworks are becoming more significant towards attaining high quality standards in government mobile applications [3] [4].

The concept of mobile automation testing frameworks is not new, as they have been used in a few years mostly in the industry, but their introduction in digital services of the government is yet to develop. The challenges that mobile applications in government encounter are also unique and require a special degree of complexity when it comes to testing. Such issues are the need to meet legal and regulatory requirements, the requirement to support a large variety of devices and operating systems, and the necessity to provide applications that could be utilized by users with disabilities. With such complexities, requirements of scalable and flexible automation testing frameworks that can fulfill such special needs have never been greater.

The aim of the study is to present a scalable mobile automation testing model that is specifically tailored to government digital service platforms. The framework combines various features of mobile app testing such as UI automation, API validation, accessibility testing, and regression coverage, but was designed in a way that allows it to work both on iOS and Android. The suggested framework will provide the government agencies with the strong solution in the automation of the testing and maintain high standards of the quality and assist in the quick and iterative cycles of development.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

Testing in the context of the mobile application in the government is naturally difficult due to a number of reasons. Government mobile apps hear on many occasions different populations such as individuals with disabilities, the elderly, and other individuals with different digital literacy. This variety necessitates the use of applications that are both extremely accessible and that adhere to international accessibility standards (including WCAG 2.1) and applications that can be used on a variety of devices and sizes of screens. Additionally, government applications should be subject to high security and privacy standards including the requirements of the General Data Protection Regulation (GDPR) in the European Union or the Health Insurance Portability and Accountability Act (HIPAA) in the United States. These rules define the manner in which the user data is meant to be manipulated and kept and any default can be very costly in terms of legal repercussions [5].

The other critical issue of the government mobile application testing is that it should be capable of testing its mobile application with a large number of different devices and operating system compatibility. Government mobile applications have to work with a variety of smartphones and tablets, including older models, which may continue to be used by some groups in the populace (unlike applications in the private sector, which may be designed to work on a narrow set of devices or platforms). This brings further complexity in the form of fragmentation of devices making the process of testing in various operating systems (iOS, Android) and various versions of those operating systems more challenging [6] [7].

Moreover, any government applications should be used in the environment, which values security. The information that is dealt with in public sector organizations is usually sensitive and very vulnerable to malicious interference by interested parties. In that regard, it is important to make sure that mobile applications can withstand outside attacks. Security testing does not only play a crucial role in ensuring that the user data is safeguarded but also serves the purpose of ensuring that the citizens have confidence in the government on its capabilities of protecting their personal data. Thus, automated testing frames should be able to conduct security checks that would identify vulnerability and guarantee the integrity of the application.

Automation has become a must in testing process so as to overcome such challenges. The testing frameworks are automated, which means that it is possible to test mobile applications faster and with greater assurance because they do not rely on testing as much as on human effort. This is specially critical in the case of government applications where it is necessary to regularly update and fix bugs in applications that must be tested on a variety of devices and platforms without adversely affecting the quality or security of the application software [8] [9].

There are a number of major benefits of automation testing as compared to manual testing. To start with, it enhances efficiency. Automated tests are also much fast than the manual tests, hence problems can be detected much faster. This is more so in a government setting where fast turn around time is critical in meeting the deadlines and in making sure that when an application is required, the application is made available to the citizens. There are also consistency results that are given by automated testing. Given that tests are run in a consistent way, the human error is reduced and hence the test results are more consistent.

Furthermore, automated testing will support continuous integration and continuous deployment (CI/CD), which is vital in the high-pressure nature of the current software development environment. Having automated tests included in the CI/CD pipeline will allow government agencies to verify the tests of every new code change properly before it is deployed. The method assists in the detection of problems at the initial stages of development hence minimizing chances of faults in the end product.

The mobile automation testing framework proposed concerning the government digital services is intended to be built on the combination of multiple key testing methodologies in order to meet the peculiarities of the government-sector application. These are UI automation, API validation, accessibility testing and regression testing.

- UI Automation: UI automated testing is a method that makes sure that the visual part of the application functions as intended with different devices and operating systems. This involves validation of alignment of buttons, text, graphics and interactive features to make them workable and easy to use. Since there is a high count of mobile devices among the users of the government application, UI testing is used to minimize the number of hours of testing each release as it will ensure that the application will act similarly in different environments.
- API Validation: API test is used to verify the communication between the backend systems and the mobile application. It is especially needed when using in government applications, where APIs can be connected to other government systems and databases, or third-party services. To prevent potential failures of these systems, automated API validation can be useful in making sure that they do as they are expected to do and in the event that the application is being updated or when new functionality is being introduced.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

- Accessibility Testing: Accessibility testing involves accessibility testing of government mobile applications to make them accessible to the disabled. The compliance with such standards as WCAG 2.1 can be verified with the help of automated accessibility testing, which should ensure that the application can be used by a broad range of individuals, including those with visual, auditory, and motor impairments. This is a critical element of the services offered by the government because the applications of the public sector should be inclusive and affordable to every citizen.
- Regression Testing: Automated regression testing assists in making sure that new code additions do not cause any defect in the old functionality. Since government mobile applications tend to be somewhat complex, involving a number of workflows, integrations, and services, regression testing is essential to achieving the stability and reliability of the application upon each release.

In this paper, a scaled and scalable mobile automation testing framework (fit to the needs of government digital services platform challenges) is presented. The framework incorporates the use of UI automation, API validation, accessibility testing, and regression coverage, which make it a comprehensive solution to the government agencies to ascertain the quality, security, and accessibility of their mobile applications. The fact that it is possible to automate testing on the iOS and Android platforms, as well as to integrate it with CI / CD pipelines, allows government agencies to maintain high release rate without reducing the reliability or compliance of their digital services. This is a scalable model offering decisive basis in the provision of high quality mobile application in the public sector, ensuring to service the needs of everyone in the citizenry, and in the process, taking into consideration the utmost standards of security and accessibility.

III. CURRENT CHALLENGES IN MOBILE AUTOMATION TESTING FOR GOVERNMENT DIGITAL SERVICES

Although the advantages of automated testing structures are manifold, there are still a number of problems in adopting and implementing mobile automation testing in digital services of the government. These issues tend to be specific to the public sector, where regulation factors, security issues, and the inclusivity requirement have to be considered. The main obstacles that government agencies encounter in their implementation of the mobile automation testing in their digital platforms are outlined below.

1. Device and OS Fragmentation

The inter-operability of devices and operating systems is one of the greatest issues in the testing of mobile automation. Government applications typically have to accommodate a large number of devices, both old and new, unlike private-sector applications which usually target a single device or platform. The users can also access the services of the government using different versions of the iOs and Android in their smartphones and tablets. This complicates the efforts to maintain standard behavior on all the devices.

Despite the possibility of performing tests simultaneously on multiple devices, automated testing frameworks might still have difficulties attempting to replicate the variety of user experiences because of the difference in screen size, hardware conditions, and operating system version. The sheer number of possible devices that citizens can use makes it difficult to support all possible combinations in a test suite, so it is possible that not all combinations will be covered by the test.

2. Security and Compliance

Mobile applications created by the government should be of high security and regulatory level, depending on the country and jurisdiction. Such laws usually set a requirement that the sensitive personal information is safeguarded both in transit and storage. The processing of the data in accordance with the legislation on data protection, including the General Data Protection Regulation (GDPR) in the EU or the HIPAA in the US, complicates the testing procedure.

The automated test systems should be specially designed to check that the applications are safe and comply with these legal demands. To be sure that automated checks accurately confirm the encryption of data and safe API communication and other security provisions, it is important to make sure that the framework has been heavily configured and regularly updated as well. It is even more difficult to integrate with external systems and services that can have their level of security procedures.

3. Accessibility Compliance

Besides security, government applications also need to meet the accessibility requirements so that it can be used by all citizens including the disabled. The concept of accessibility compliance is not a mere best practice but in many cases, a regulation such as the Americans with Disabilities Act (ADA) and WCAG may mandate it.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

Accessibility testing is relatively hard to automate since it presupposes special tools to test the conformity to a number of guidelines, and it is not always sufficient to consider whether the accessible features are present or not. The issue here is making sure that the automated tests do not miss any part of the user experience of users with different disabilities like those who use a screen reader or other input devices. The implementation of such accessibility tests in the current automated testing system complicates things, as it will need continuously updating with the new accessibility requirements as they are invented.

4. Complicatedness of Multi-Layered Government Applications.
Government mobile apps tend to have several services that can be ranging two aspects to payment processing and data retrieval, real-time notifications, and geolocation services. Such applications can communicate with multi-faceted back-end systems, databases, and third-party services, which need to be well tested to make sure that everything works correctly. Test automation systems should not only test the mobile app but the API and backend systems and even external-service systems as well.

It is not an easy feat to ensure that all layers of the application (UI, API, security, accessibility, and performance) are covered with the test, particularly because these systems change over time. Modifications to one of the layers may affect the performance of the whole application, and changes to the testing scripts are required on a regular basis, as well as the framework should be responsive to the frequent changes in the system.

5. Cesura Resources on Infrastructure testing.
Lastly, the problem with a lot of the government agencies is that they lack resources to develop and sustain a thorough mobile automation testing framework. Construction and maintenance of infrastructure required to perform automated testing: the test devices, emulators and cloud services can be expensive and time-intensive. The smaller government agencies especially might not be able to invest in scalable testing infrastructure to make this practice fully adopted in automated testing.

Further, there is a possibility that most government agencies have budget constraints that will not allow them to adopt the most recent testing tools or employ specialized staff to manage the testing infrastructure, resulting in possible testing coverage gaps.

Although the suggested mobile automation testing architecture has considerable merits, device fragmentation, security and compliance, accessibility issues, complexity of government applications, and resource limitations are the major obstacles. To overcome them, special attention should be paid to the planning, the constant revision of testing procedures, and the substantial investment in testing facilities and skills. These notwithstanding, the incorporation of scalable, automated testing systems is essential in enhancing the quality and reliability of the government mobile applications in order to be able to meet the needs of the citizens without going against the regulations.

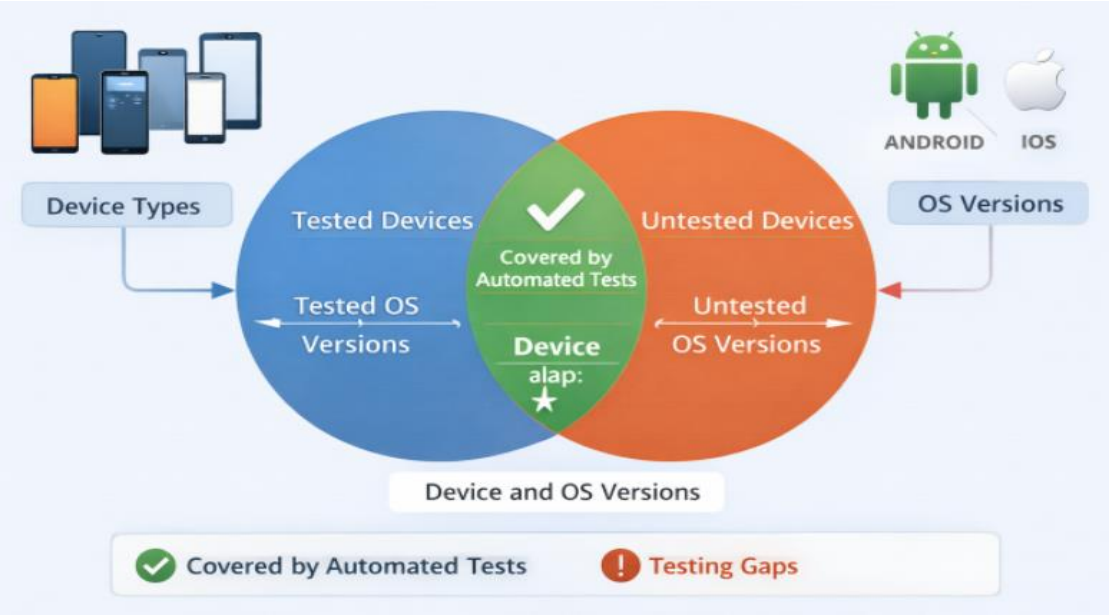

Figure 1: Device and OS Fragmentation in Mobile Testing

International Journal of Advanced Engineering Science and Information Technology (IJAESIT)
|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0704002

IV. FRAMEWORK FOR SCALABLE MOBILE AUTOMATION TESTING IN GOVERNMENT DIGITAL SERVICE PLATFORMS

Here we are introducing a scalable mobile automation testing system suited to the case of government digital service platforms. The framework unites different testing methodologies, which offer an overall solution to the reliability, availability, safety, and functionality of government mobile applications. This framework will guide in solving the special problems specific to applications in the public sector, including the problem of device fragmentation, compliance with security, and compliance with regulations. The testing framework is founded on the concepts of flexibility, scalability, and maintainability, and the government agencies are able to provide automated tests and do it in a large scale with a variety of devices and platforms but ensure continuous integration and delivery (CI/CD) of high-quality mobile applications.

1. Overview of the Framework

The suggested framework integrates multiple layers of testing and testing processes to automate the mobile application testing most important parts, such as UI automation, API validation, and accessibility testing and regression testing. It can be connected to the contemporary CI/CD pipelines to facilitate continuous testing and smooth deployment of the government applications. The modular architecture of the framework ensures easy maintenance and updates, and is also flexible to the ever-changing demands of the government digital services.

It uses industry standard tools and libraries to construct its framework architecture to be compatible with both iOS and Android platforms. The major tools deployed are Appium to automate their UI, Postman to test their API, Axe to test their accessibility and several other best practices in the industry to integrate continuously and automation testing. All these elements are coordinated in a clear process to make sure that tests are carried out in a consistent, reliable and efficient manner.

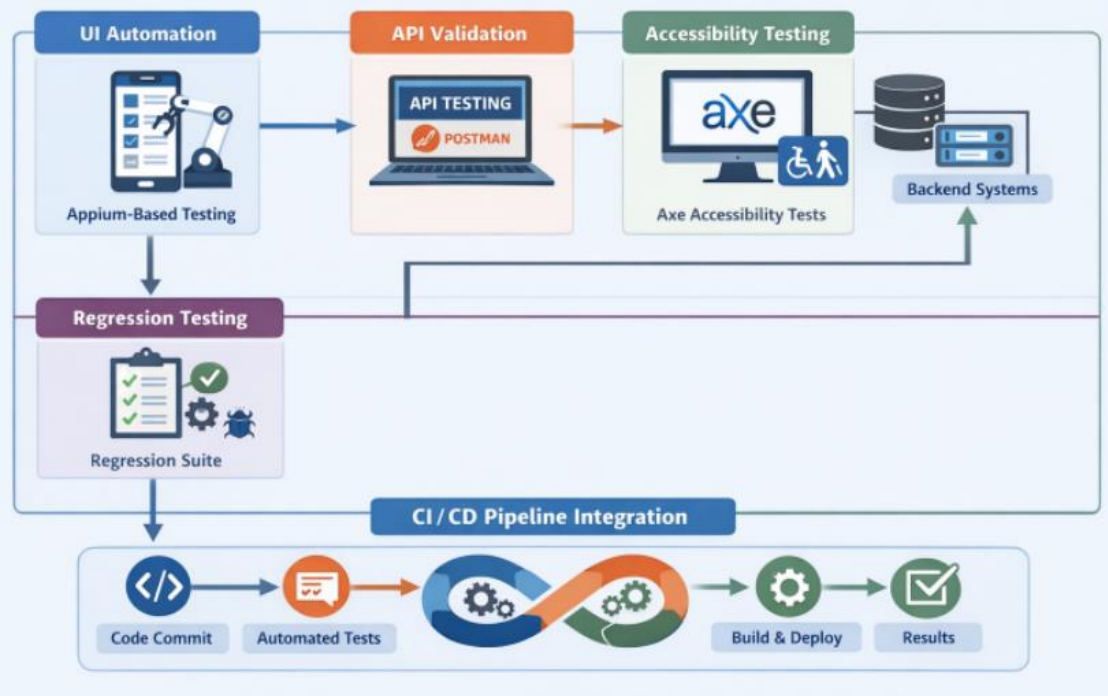

Figure 2: Mobile Automation Testing Framework Overview

2. Key Components of the Framework

The framework comprises of a number of important elements which collaborate to have a full-fledged and effective testing of the mobile apps within the public sector. These components include:

2.1 UI Automation

In the framework, user Interface (UI) automation lays the basis of mobile testing. The process of AI automation is concerned with the visual aspects of the mobile application, whether it works or not, whether usable or performable.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

Mobile applications created by the government, usually, should address certain standards of usability to support a great number of diverse users, including people with disabilities, various types of devices, and different screen sizes.

In iOS and Android applications, the framework will be based on the use of Appium, the open-source tool that helps to carry out a cross-platform test. Appium allows the use of automated UI testing on a variety of devices that capitalize on the native native automation of iOS and Android. The following Appium functionalities are used in the framework:

- Element Interaction: Automated scripts are used to imitate user interaction by tapping, swiping and typing into elements to familiarize themselves as to how the UI elements like buttons, dropdowns, forms, and checkboxes will act.
- Cross-Platform Testing: Appium supports both iOS and Android and allows the framework to test applications on various devices with little extra configuration.
- Parallel Testing: This structure is intended to execute tests in parallel on several devices or emulators to accelerate the testing process enabling the government agencies to verify cross-device consistency.

The automation of UI testing by the framework will guarantee that government mobile applications are as per the performance expectation, they respond properly to the input of the user, and they are aesthetically similar across devices. Moreover, the framework is made to be customizable, so that custom UI validation rules may be added to it that are related to the government domain, like, requiring national accessibility standards.

2.2 API Validation

The other important layer of the framework is the validation of API, which makes the backend systems functional with the mobile app. Mobile applications in the public sector might require connectivity with a number of government databases, third-party services, and other back-ends. These systems are integrated to make sure that the mobile application works perfectly, and automated API validation is to check it.

Postman is employed in order to undertake API validation in the proposed framework. Postman is another popular software in testing the RESTful APIs, and its support to the framework makes API testing reliable and automated. The steps involved in the API validation process are:

- Request Testing: This is an automated test that guarantees the requests that are made to the back systems by the mobile application are properly formatted and have the right data.
- Response Validation: Postman verifies the responses that were obtained by the APIs and makes sure they follow the expected data structures and values.
- Authentication: The mobile applications used by the government should be highly secure. The framework conducts security tests, including checking the secure transmission of data (e.g. using HTTPS) and checking that the authentication tokens are properly processed.
- Performance Validation: API response times are checked to match the performance requirements since slow response may interfere with the user experience and the efficiency of the government services.

It is important to note that through the automation of the API validation, the framework enables the government applications to communicate reliably with their back-end systems to protect the integrity of data and the security of information without the high probability of errors.

2.3 Accessibility Testing

Government mobile applications have to be accessible as the services need to be inclusive and accessible to all citizens including those with disabilities. Automated accessibility testing is important in ensuring that government mobile applications are both legally accessible (i.e. the Web Content Accessibility Guidelines (WCAG) 2.1) and accessible to those with visual, auditory, or motor disabilities.

In order to automatize accessibility testing, the framework uses Axe, which is a free accessibility testing tool. The testing pipeline will also be integrated with Axe to automatically report and identify accessibility problems. The major aspects of accessibility testing are:

- Axe can also verify the application, which is in UI, against the general accessibility problems, including the lack of alt text, low color contrast, missing form labels, and screen reader accessibility.
- Compliance Checks: The framework assures that the mobile application will comply with the national and international standards of accessibility so that the laws such as the Americans with Disabilities Act (ADA) and European Accessibility Act are met.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

- User Simulation: The framework emulates the user experience of a user who has many disabilities, including visual impairment, and validates the behaviour of the application when used with assistive devices like the screen reader or voice recognition.

By integrating the accessibility testing into the automation system, government agencies are able to detect and address the accessibility concerns at an early development stage. The framework allows organizations of the public sector to make their mobile applications more inclusive to everyone and prevent expensive fixes at the post-deployment stage by automating accessibility checks.

2.4 Regression Testing

Regression testing is an extremely important aspect of the testing framework in such a way that there are no unwanted errors or functionality in the new changes or features that may occur. Since the government applications are often updated through new features, bugs or adjustments to changing regulatory needs, regression testing assists to maintain the stability of the application over time.

The framework uses a set of a full package of automated regression tests which are run every time the code is changed. The tests discuss the following:

- Core Functionalities: Regression tests will be used to ensure that the core functions, e.g. form submission, data retrieval and payment processing, remain operational.
- Cross-Platform Compatibility: Tests will make sure the application is able to perform in different devices, screen sizes, and operating systems in a manner that will avoid platform issues after changing the code.
- Performance and Load Testing: Regression testing involves some performance tests that the application will not degrade because it cannot sustain the user load that is anticipated.

The framework allows government agencies to test their applications at high quality without the overhead of manual testing by performing regression testing automatically, as well as delivering the benefit that application stability will not be impaired by updates.

2.5 CI/CD Integration

One of the most important elements of the framework is that it is integrated with Continuous Integration (CI) and Continuous Deployment (CD) pipelines. Government agencies are growing to embrace CI/CD practices in an attempt to accelerate their development and implementation of their digital services. CI/CD requires minimal human participation as it automatically tests and deploys code changes, integrates and deploys them, and thus, delivers updates and new features much faster.

The framework is compatible with CI/CD tools, such as Jenkins, CircleCI, or GitLab CI. This integration makes possible the following:

- Automated Testing on Every Code Commit: Each time new code is added to the version control system (e.g. Git), automated testing is run to detect any defects being introduced in the new code.
- Continuous Monitoring: The framework constantly checks the health of the application, and it carries out automated tests and produces reports which point out any problems.
- Seamless Deployment: The successful passing of automated tests causes the deployment of new versions of the mobile application into the production environment, so that the government services are always up to date.

CI/CD integration will guarantee the government agencies are able to deliver updates faster and of high quality standards, security, and compliance.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0704002

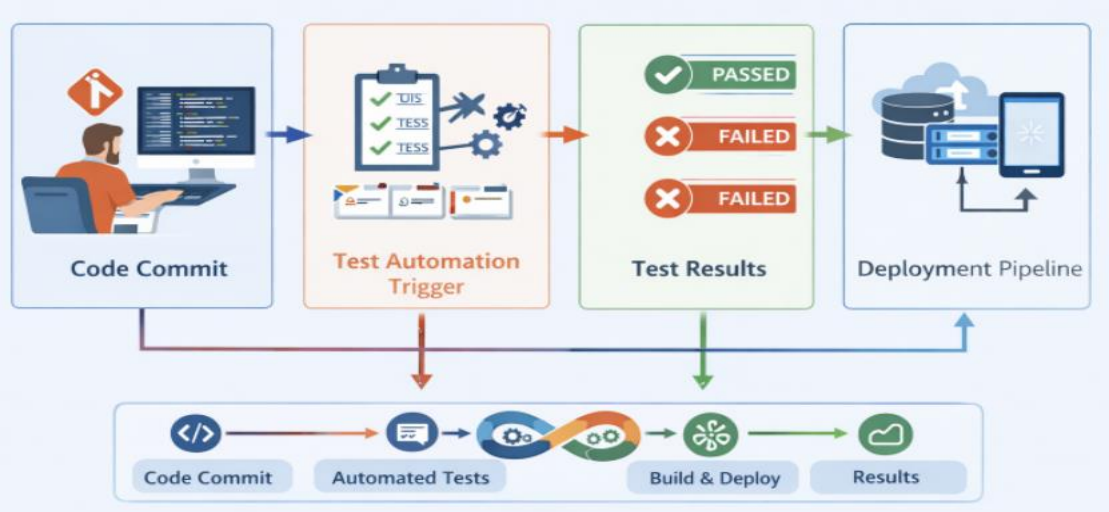

Figure 3: CI/CD Integration Process in Mobile Testing

3. Scalability and Maintainability

Scalability and maintainability is one of the purposes of the proposed framework. They are frequently government applications which need to serve millions of users and need a testing framework that is able to scale up to these needs. The framework has been structured to be scalable where parallel execution of the test is possible enabling testing on a large number of devices within a short time. Along with that, the framework is modular, which allows making updates easily and adding new testing features as the requirements of the government agencies change.

The framework maintainability also features first. The continuous updates in the operational system of the mobile devices, the release of the new devices and the alterations in the regulatory norms demand the regular revision of the testing structure. The framework is composed of modular design, which is easy to update individual components without affecting the overall testing pipeline and this means that the framework can be easily modified to suit changing requirements.

The suggested scalable mobile automation testing framework will provide an all-encompassing solution to the specifics of the challenges that government digital service platforms have. The integration of UI automation, API validation, accessibility testing, and regression testing along with the combination of these elements with CI/CD pipelines will guarantee that the government mobile applications will be of the finest quality, security, and accessibility. The framework is also scalable and maintainable, which enables it to accommodate the changing requirements and fast release cycles of government agencies, and it offers an efficient and reliable testing solution to applications in the public sector.

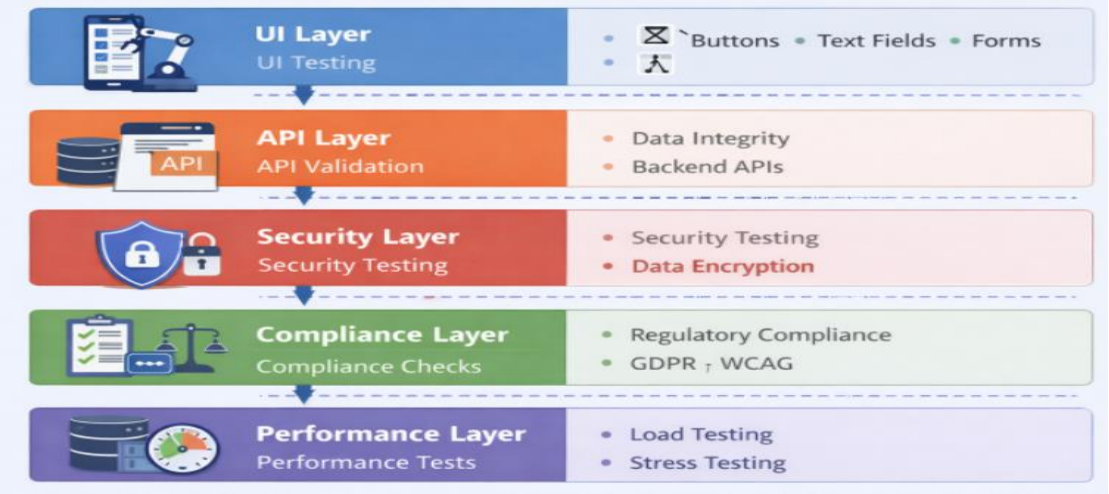

Figure 4: Mobile Application Testing Layers

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

V. FRAMEWORK EVALUATION

The scaffold of mobile automation testing of government digital service platforms proposed has been created to handle the special issues of applications in the public sector. This section will assess the framework in regards to some major criteria: effectiveness, scalability, maintainability, ease of integration, security, and compliance. Another aspect that the evaluation takes into consideration is the strengths and possible constraints of the framework and the applicability in actual government settings.

1. Effectiveness

The success of the framework lies in the fact that it covers all areas of important testing such as UI automation, API validation, accessibility testing, and regression testing. All these elements are critical in determining the overall quality and functionality of mobile applications of the government.

- UI Automation: Appium is used to perform cross-platform UI testing to guarantee the framework can successfully test the graphic component of applications in numerous devices, operating systems, and various screen sizes. The testability of user interactions, element visibility, and responsiveness promotes the dependability of government application which has to be operational to a diverse demographic.
- API Validation: As the Postman is integrated to perform API testing, the framework is outstanding to guarantee that the backend systems can communicate with the mobile application as desired. Government applications with frequent interaction with third party services or legacy systems need to go through API validation. The automated API tests of the framework will ensure integrity and security of the data is achieved.
- Accessibility Testing: Axe provides accessibility testing which is also a major strength since it will ensure government mobile applications can meet the standard of accessibility like WCAG 2.1. Since it is a requirement that the government services should be made available to all citizens, including individuals with disabilities, the accessibility tests in the framework are important in making them inclusive.
- Regression Testing: The entire regression testing package allows the framework to guarantee that the new code changes do not interfere with the functionality of the existing features. This is especially significant in the government sector, where the applications can regularly be updated and require to be stable and reliable in their performance.

On the whole, the framework proves to be very effective in solving major areas of mobile application testing. It offers government agencies with an effective, automated way of ensuring that applications are operational, safe, and available.

2. Scalability

Government mobile applications have to be scalable and in this respect, it has to be able to serve a large and diverse user base using various devices and platforms. The architecture is scalable and allows it to test easily when the number of devices, users and features in the application increases.

- Parallel Testing: It has a parallel testing capability, which allows execution of tests simultaneously on several devices and emulators, to guarantee the capability to support the high number of tests needed to conduct cross device, and cross platform validation. This parallel processing saves the time of testing a crucial part in the deployment of updates and fixes as fast as possible.
- CI/CD Integration: The CI/CD pipelines with the testing framework enable the government agencies to have a rapid release cycle and comprehensive testing. Constant tests in CI/CD keeps any modifications applicable to the codebase automated and this is critical in those environments where speedy updates are valued.

Scalability of the framework enables its use on an application which requires a high number of users since it can be efficiently tested on various devices, operating systems and versions.

3. Maintainability

The aspect of maintainability is a critical factor in any testing system particularly where government applications are involved, which need to be easily adjusted to the fluctuating changes in the regulatory requirement, technology, and the needs of the users.

- Modular Design: The modular structure of the framework enables the separate components (UI automation, API validation, and accessibility testing) to be updated or changed separately. It simplifies the process of updating and maintaining the framework as mobile operating systems are released as new versions or alterations in accessibility standards or security regulations take place.
- CI/CD Pipelines Integration: As the framework is coupled with CI/CD pipelines, it means that testing will be automated in the ongoing process of integration, so there will be no manual efforts, and errors during updates are minimized. This enables automatic updates on testing scripts, and makes sure that the framework remains relevant and current.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

Its modularity and alignment with the latest practices of CI/CD make the framework maintainable and enables it to respond to the changing demands of governmental digital services.

4. Ease of Integration

The ease of integration is an important consideration that can be used to identify how fast and efficient the framework is when adopted by government agencies. The framework will be in line with popular tools and systems deployed in the public sector.

Author states that the framework is compatible with commonly used tools in the industry, such as Appium (UI testing), Postman (API testing), and Axe (accessibility testing). These tools are likely familiar with the government agencies and thus, the learning curve is minimized and implementation will be faster.

- CI/CD Pipeline Integration: The smooth interoperability of the framework with CI/CD tools, such as Jenkins, CircleCI, and GitLab CI, implies that the software can be implemented in any software development pipeline without major modifications to the existing processes. This is a significant strength to government agencies who are already practicing CI/CD.

This is because the framework is based on a set of principles that are very easy to integrate and can hence be easily adopted by the government agencies, since it does not need them to make significant modification to the established development and testing processes.

5. Security and Compliance

Security and compliance are the most important considerations during the development of government applications particularly those that involve sensitive information or those that must comply with heavy regulatory structures. The framework also makes certain that the security and compliance is also carried out during the testing process.

- Security Testing: The framework will contain automated verifications of safe API requests (including HTTPS) and make sure that the data requiring attention is treated accordingly. This is essential in government applications, which are known to deal with personally identifiable information (PII) and other sensitive data.
- Regulatory Compliance: The automated accessibility tests in the framework allow the application of national and international regulations of accessibility, including the ADA and WCAG 2.1, which are especially critical in the context of the application utilized by the government. Also, security validation controls in the framework are used to verify that applications are operating within the data protection laws, including GDPR and HIPAA.

Through the incorporation of security and compliance testing into the testing mechanism, the government agencies will be assured that their mobile applications are fit to the required legal and regulatory standards.

6. Limitations and Potential Improvements.

Although the framework has a number of strengths, it is necessary to note that there are a number of limitations and possibilities which can be improved in the future.

- Fragmentation of Devices: Although the framework can simultaneously execute a series of tests on a variety of devices, the sheer amount of the variety of devices and operating systems in use can be trouble. Though the framework can work with both iOS and Android, the government agencies might have to invest in extra infrastructures to test a wider variety of older devices.
- Performance testing: Although the framework will involve performance validation as an aspect of regressions testing, additional comprehensive load testing might be necessary in the case of government applications that are likely to receive massive users during peak hours. More integration with load testing tools like JMeter or Gatling could be used to improve the performance testing of the framework

VI. CONCLUSION AND FUTURE WORK

To summarize, the suggested scalable mobile automation testing system is a holistic approach to the special issues of the government digital service platforms. The framework provides a reliable, secure, and regulatory mobile application to the government through the combination of UI automation, API testing, accessibility testing, and regression testing. The framework can provide customers with speedy release cycles with high-quality product standards through its scalability, maintainability, and integration with CI/CD pipelines. The implementation of the UI automation tool called Appium, API validation tool called Postman and accessibility testing tool called Axe make the framework responsive to a vast variety of mobile devices and platforms, thus making sure that the government apps will satisfy the needs of various user groups.

|ISSN 2349-3216| Volume 7, Issue 4, July-August 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0704002

In spite of its advantages, the framework has issues on device fragmentation, security and compliance, accessibility standards and complexity of multi-layered government applications. These difficulties emphasize the significance of continuous enhancements in order to make the framework effective and topical to meet the rising demands of government digital services. Secondly, the lack of resources and infrastructures in certain government departments might not allow the use of automated testing practices in large-scale. Nevertheless, these obstacles can be overcome, and the framework offers an important resource in improving the quality of mobile applications of the public sector.

Further research on the topic will involve an improvement of the framework in the future to tackle the above challenges. The possible area that could be improved is by extending the capacity of the framework to support more number of devices, particularly older ones, through implementing more sophisticated device farm solutions or cloud-based testing services. This would contribute to the reduction of the problem of device fragmentation and provide more extensive coverage of tests.

The other area to be developed in future is to improve the security testing of the framework. This involves integration further with security tools to detect vulnerabilities like SQL injection, cross scripting and other threats. Also, it will be necessary to make more accessibility testing tools more inclusive of a wider range of disabilities as well as continually revise the framework to meet the current accessibility standards to ensure that government applications are as accommodating as possible.

Moreover, the future labor should take into account the creation of AI-based testing opportunities. The use of AI and machine learning can be used to predict the behavior of the applications, detect the possible problems in advance, and enhance the effectiveness of the test. The use of AI would also make the process of testing of more complicated situations more automated and more efficient.

In the same way that it keeps on changing the framework to suit the needs of government agencies and its users, future developments will see automated testing as an invaluable instrument in providing quality, reliable, and readily available government mobile applications.

REFERENCES

1. A. Li and C. Li, “Research on the Automated Testing Framework for Android Applications,” *Proc. 12th Int. Conf. Computer Engineering and Networks (CENet 2022)*, Lecture Notes in Electrical Engineering, vol. 961, pp. 1056–1064, Oct. 2022.
2. N. G. Berihun, C. Dongmo, and J. A. Van der Poll, “The Applicability of Automated Testing Frameworks for Mobile Application Testing: A Systematic Literature Review,” *Computers*, vol. 12, no. 97, May 2023.
3. *ISO/IEC/IEEE 29119-1:2022 – Software and systems engineering — Software testing — Part 1: Concepts and definitions*, International Organization for Standardization / IEEE, Jan. 2022.
4. S. Chen, C. Chen, L. Fan, M. Fan, X. Zhan, and Y. Liu, “Accessible or Not? An Empirical Investigation of Android App Accessibility,” *arXiv*, Mar. 2022.
5. Y. Zhao, B. Harrison, and T. Yu, “DinoDroid: Testing Android Apps Using Deep Q-Networks,” *arXiv*, Oct. 12, 2022.
6. P. Bose *et al.*, “Columbus: Android App Testing Through Systematic Callback Exploration,” *arXiv*, Feb. 17, 2023.
7. S. Yu, C. Fang, Y. Ling, C. Wu, and Z. Chen, “LLM for Test Script Generation and Migration: Challenges, Capabilities, and Opportunities,” *arXiv*, Sep. 24, 2023.
8. Appium, “Appium: Mobile App Automation Made Awesome,” Open Source Project, 2023. [Online]. Available: https://appium.github.io/appium.io
9. Paricherla M et al, A. Machine learning techniques for accurate classification and detection of intrusions in computer network. Bulletin of Electrical Engineering and Informatics. 2023;12(4):2340-2347. doi:10.11591/eei.v12i4.4708