**CMPE287 – Software Quality Assurance and Testing**

**Deliverable #3 – Test Automation Documentation**

For

**AI Mobile Application Testing**

**A blue and yellow text on a white background

Description automatically generated**

**Group-3 Team Members**

Tirumala Saiteja Goruganthu – 016707210

Harish Marepalli – 016707314

Sowjanya Bheemineni – 016672214

Sohan Leburu – 017408768

**Guided By**



Prof. Jerry Gao

1. **Introduction**

Software testing has principally evolved into two overarching classifications: manual testing and automated testing. As the software industry has progressed, testing has transitioned naturally from a manual process to one that leverages automation in an evolutionary way.

Automated testing techniques utilize software tools to conduct evaluations on software systems. Implementing automated testing improves software quality by systematizing test data, capitalizing on test outcomes, and substituting human effort with specialized systems and appliances as part of quality control procedures.

As stated by Dr. Jerry Z. Gao in his published work, software test automation encompasses the focused efforts and initiatives that aim to replace manual engineering activities and processes in software testing with automated solutions that are executed through predefined systematic methodologies and structured techniques.

1. **Test Automation Focuses**

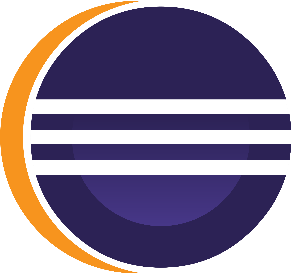
The primary purpose of automated testing is to ensure continuous evaluation of software as an integral component of the software delivery workflow, thereby providing instant feedback on a mobile application for food detection in this case. Over a short timeframe, automation facilitates attaining predefined benchmarks for software testing. Executing manual testing is fatiguing and consumes employee time and exertion. Additionally, human errors may transpire over the course of manual testing. Certain error categories evade detection via manual testing. Test automation enables executing this evaluation more effectively and efficiently. Upon initial creation, automated tests can be quickly and easily generated. This can constitute an optimal approach for software with an extensive maintenance timeline.

1. **Objectives**

In automated testing, automation technologies are leveraged to execute tests and validate outputs against anticipated results rather than relying on manual human oversight. Automating the testing process can curtail evaluation timeframes and heighten productivity. By automating test operations, repetitive tasks that cannot viably be performed manually are streamlined. Thus, test automation constitutes an integral component for enabling continuous testing practices.

1. **Selected Tools**

In order to test the functioning of our selected application, Socratic, we used the following tools.



**AI Testing Tool:**

Utilizing this AI testing tool, classification trees were constructed to structure input variables, contextual parameters, and output metrics. By permuting and pairing the nodes within these classification trees, test cases were systematically generated to enable thorough evaluation of application performance across a wide range of conditions. This facilitated comprehensive and methodical assessment across the solution space.

**Appium:**

A number of platforms, including Android, IOS, and Windows, are tested using Appium, an open-source framework that allows engineers to automate testing.

**Android Studio:**

We used Android Studio to run the Android emulator on which we tested the test scripts.

1. **AI Test Automation**
2. **AI Test Automation Test Strategy**

Rather than relying on manual human execution, test sequences are automated utilizing various automation technologies. Performance assessments are subsequently carried out to gauge the outcomes of these automated test runs. Implementing test automation serves to enhance software development productivity and compact the development timeline by minimizing manual efforts.

Test scripts are implemented by employing Appium web drivers within the Android studio environment to evaluate applications operating on emulators. In this framework, Appium serves as an intermediary layer connecting the application being executed on the emulator and the test script being run on Eclipse.