

# Test Plan for Grocery Store Automation Framework

## 1. Project Overview

The **Grocery Store Automation Framework** automates the testing of a mobile e-commerce supermarket app. The goal is to ensure seamless user experience, functionality, and stability across various Android devices using **Appium**, **TestNG**, and **BrowserStack**.

---

## 2. Test Objectives

- Validate core functionalities such as login, product search, cart, checkout, and order tracking.
- Ensure app compatibility on multiple Android devices via **BrowserStack**.
- Automate repetitive test scenarios for **faster regression testing**.
- Generate detailed test execution reports using **Extent Reports**.

## 3. Test Scope

Platform	Details
Mobile	Android Native app

### In-Scope:

- Functional testing (Login, Search, Add to Cart, Checkout, Order History).
- UI/UX validation on multiple Android devices.
- Data-driven testing using **ExcelUtility**.
- BrowserStack execution for cross-device testing.
- Gesture-based interactions (swipe, tap, long press).
- Parallel execution for optimized testing speed.
- Reporting and logging via **Extent Reports & TestNG Listeners**.

## 4. Test Approach

### Test Automation Strategy:

- **Framework Type:** Hybrid (Page Object Model + Data-Driven Testing)
- **Test Execution:** Local and Cloud (BrowserStack)
- **Test Data Management:** Excel-based using **ExcelUtility**
- **Logging & Reporting:** Extent Reports + Screenshots on Failure

### Test Types:

Smoke Testing - Ensure critical app features work.

Regression Testing - Validate the new updates don't break existing features.

Functional Testing - Verify expected app behaviour across features.

### Script Development Approach:

Creation of Name mapping (Object Repository)

- Capturing objects following Page Object Model Design Pattern.
- Creation of Aliases for mapped Objects.

Creation of application specific reusable methods

- Identify common functionalities.
- Identify repeating actions and create reusable functions.

Creation of test scripts & Storage

- Creation of scripts by calling methods referring to test case steps.
- Creation of Test Groups to run scripts in batch.
- Scripts are stored in code management tools (GitHub).

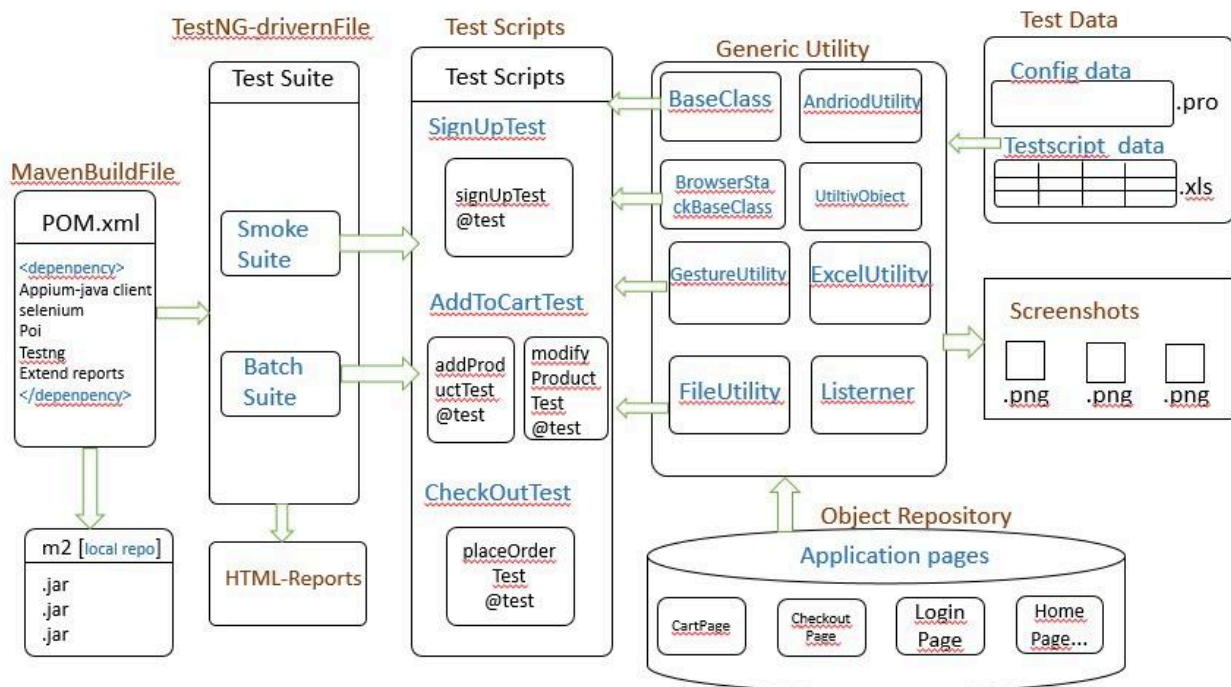
Creation of Automation reports

- Integration of Extent Reports to framework to generate visually informative reports.
- Requisite logging to track automation flows.

## 5.Tools & Technologies

Automation Tool	- Appium, Selenium
Testing Framework	- TestNG
Cloud Execution	- BrowserStack
Build Tool	- Maven
Version Control	- Github
Reporting	- ExtentReports

## 6.FrameWork Structure



## 7.Test Cases & Scenarios

LoginTest	- Validate login with valid and invalid credentials.
Add to cart Test	- Ensure product can be added to cart.
Checkout Test	- Validate the Order place and payment.

## 8.Entry Criteria

- Application apk file should be ready.
- BrowsersStack account should be present.
- Testing Resources should be available.
- Test environment setup is completed (Appium, TestNG, BrowserStack).
- Application build is stable and deployed on the test environment.
- Test data is prepared and available in the required format.
- Automation scripts are developed and ready for execution.
- Required tools and dependencies are configured (Maven, GitHub, Extent Reports).
- All critical defects from previous cycles are fixed and verified.

## 9.Exit Criteria

- All planned test cases have been executed.
- No critical or high-priority defects remain unresolved.
- Test execution reports are reviewed and shared with stakeholders.
- Defect triage is completed, and all identified issues are logged.
- Test summary report is prepared and validated.
- Sign-off received from the testing team and product stakeholders.

## 10.Effort Estimation

Took 2 story points to develop the framework and 2 story points to develop 4 automation testscripts.

Framework Development	2 story point
TestCase Automation	2 story point

## 11.Deliverables

- 1.Testcase document
- 2.TestPlan document
- 3.BrowserStack Account credentials
- 4.Automation Framework design document
- 5.Test Execution Report
- 6.Github link