**TEST PLAN**

Author: Kamal Gurung and Zhiyu Zhang

We the Team 2, worked on building an INVENTORY APPLICATION that allows customers to do shopping as well as the employees to restock the materials. Since Test plan has several testing strategies, it involves a combination of testing processes. Since team 2 successfully performed the testing activities in different phases of the development process. As we executed the test plan, the application is able to run and outputs the desired output. We performed updates, fixed the bug towards the development of the application.

1. **TESTING STRATEGY:**

In the testing strategy, We Team 2 have worked on four different levels of testing. They are as follows.

- Unit testing

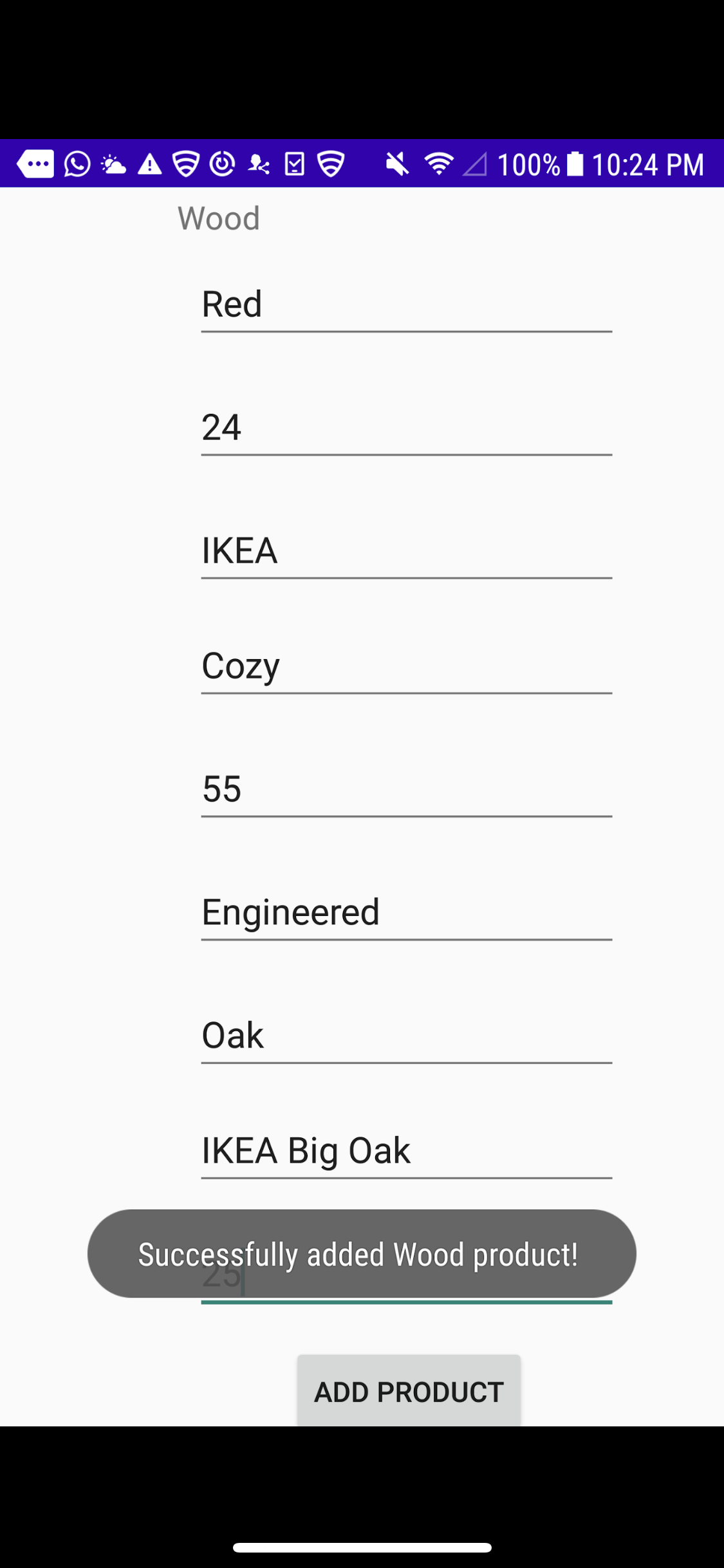
-Integration testing

-System testing

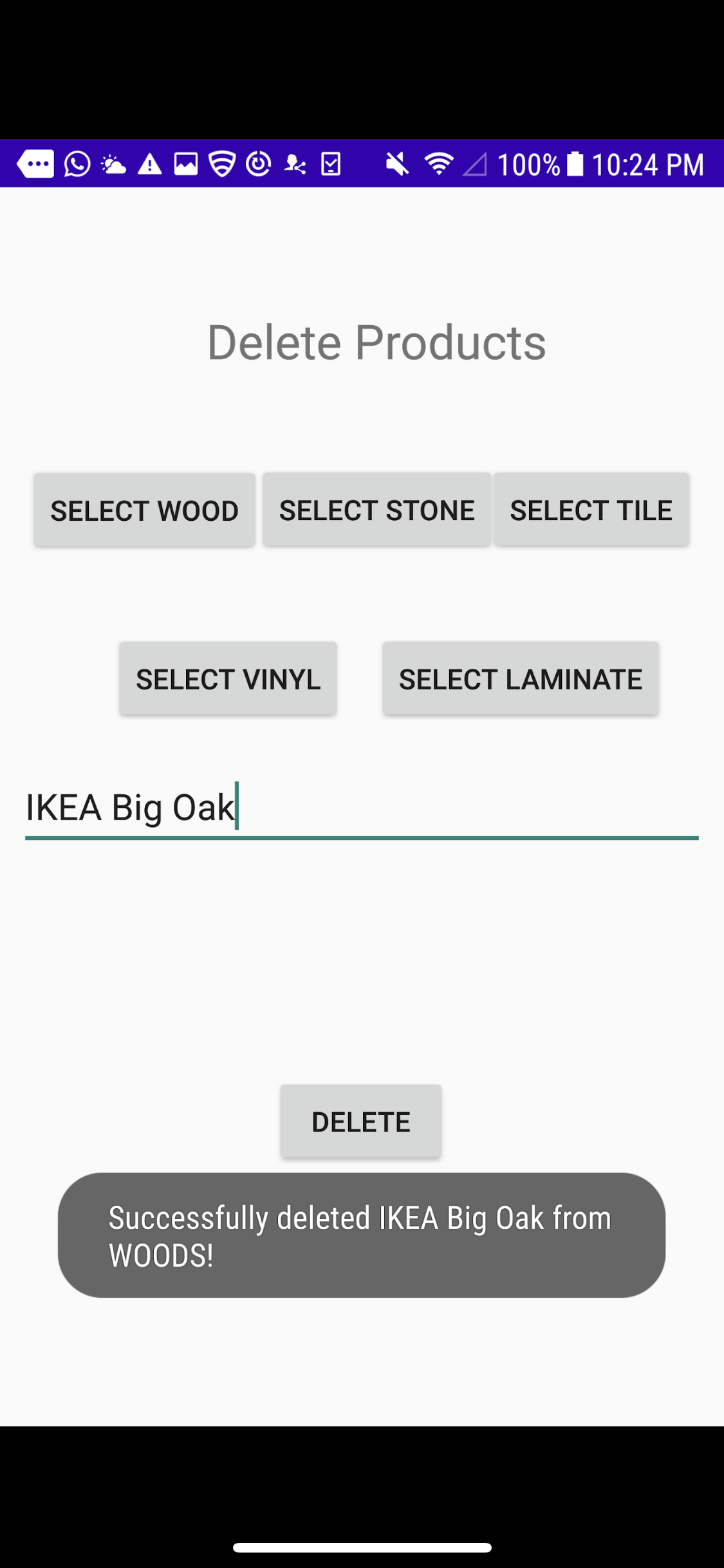
- Regression testing:

-Acceptance testing

**Unit testing:** We input test cases code manually to get expected results at the end of testing. Such as adding the items manually. we successfully completed the full code coverage in statement, branch, and condition coverage). For integration and system testing, we will focus on functional coverage, ensuring that all major user scenarios and application functionalities are tested.

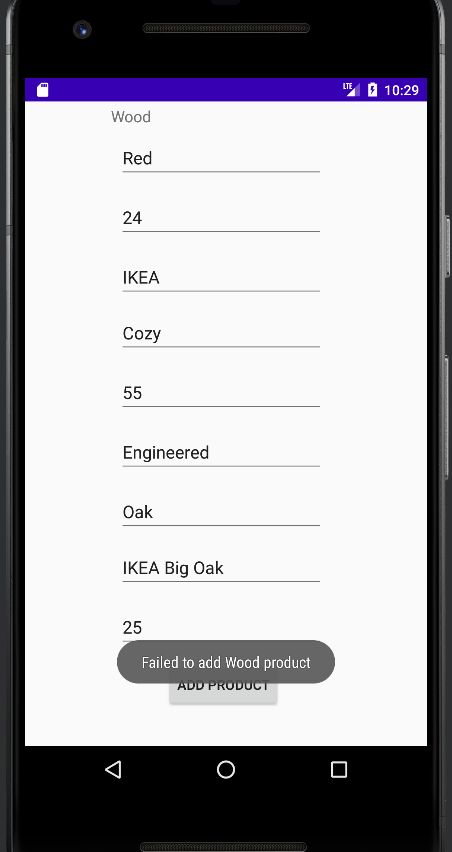


**Integration testing:** Each design pages is tested, and it works perfectly runnable and responsive. We are able to deliver smooth access to the user. E.g. create, reminder, search, delete, add and so on.

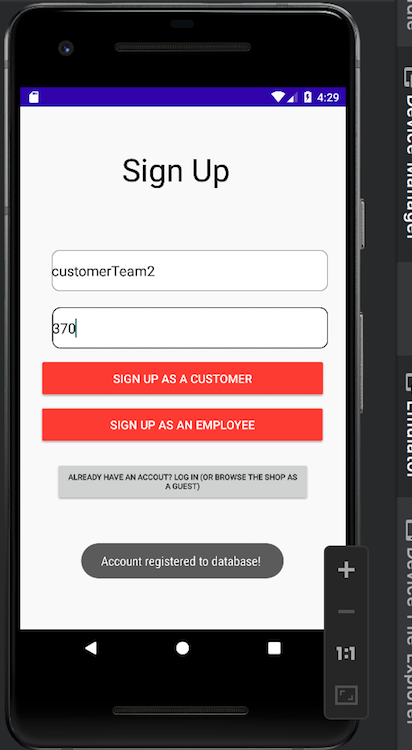
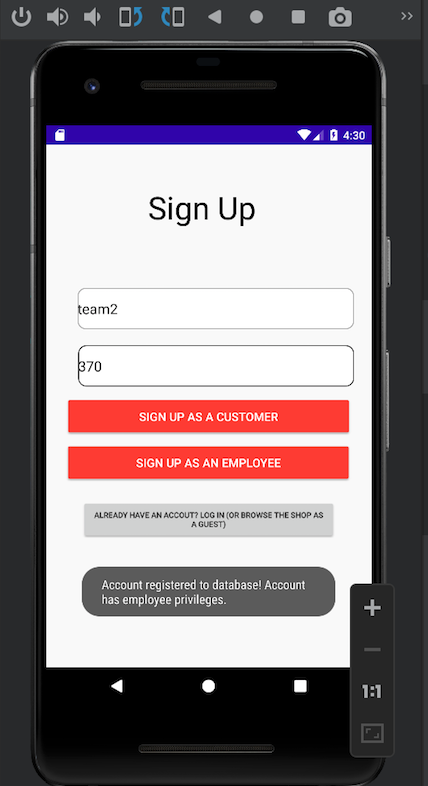


**System testing:** We tested all the components of our developed application in android device. It works perfectly. Also, we ran in Emulator which the application has no defects while running. We are able to meet the goal to make an inventory app.

**Regression testing:** Since our developer team have worked on the updates, bug fix and develop codes. We have come up with the ideas and implementing the bug fix, fixing app crash.



**Acceptance testing:** we have tested the app with providing the correct input acceptance and we succussed in delivering the required output as accepted in the acceptance testing.



1. **Test Selection (Black box /White box) Testing**

We have implemented both **black-box** and **white-box** testing techniques to select test cases. For unit testing, we have focused on white box testing that covers individual components and their interactions. In contrast, for integration and system testing, we have used black box testing techniques that help us to focus on the application's functionality from the user's perspective. The main functionality of the application is responsiveness, bug fix, updates and valid inputs and errors will be checked while performing Blackbox technique.

1. **(Structural / Functional) Coverage**

We focused on Functional Coverage as equally as Structural Coverage, we have a well-developed application that can function and execute properly. In Unit testing, we performed many tests in each function of the application to see if it runs and executes properly. In System testing, we performed the test as a group to check the developed program.

1. **Technology Use and Bug Fix**

We used GitHub to develop the code, see changes, commits, updates, Bug tracking and helping out each team member from the development to the deployment. Similarly, we used JUNIT platform for testing technology, Espresso for Android UI testing, and Selenium for end-to-end testing. We carried out manual test cases after the design and code development is finished. Our program uses Java Programming Language in Android studio application for the whole group project.