**KitchenSync**

**Testing Report**

**A green and white logo

Description automatically generatedDeveloped by: Byte Blazers**

**Testing Scope**

* **Functionality Testing:** During this phase, We need to ensure the functionality of all KitchenSync features. Each feature must be tested to make sure it functions as intended in a variety of situations.
* **Usability Testing:** The purpose of usability testing is to assess the usability, intuitiveness, and general user satisfaction of the KitchenSync application by analyzing its interface and user interactions.
* **Unit Testing:** Unit testing for KitchenSync concentrates on codebase modules, classes, or methods. Test cases, which cover various input combinations, edge situations, and error conditions, are written to confirm that each unit functions as intended.
* **Integration Testing:** Testing integration points between features such as the database, user interface, external APIs, and linked kitchen gadgets is part of KitchenSync's integration testing process.
* **Performance Testing:** Performance testing measures the KitchenSync's stability, scalability, and responsiveness by analyzing how it behaves in various scenarios. Performance testing guarantees that the program operates at peak efficiency under anticipated usage scenarios and maintains dependability even in times of high demand.
* **Security Testing:** Finding any application vulnerabilities that can compromise data availability, confidentiality, or integrity is the goal of our KitchenSync security testing.

• **Compatibility Testing**: Ensuring the KitchenSync application operates as expected across different devices, operating systems, browsers, and network environments. This is crucial for KitchenSync's to run in a variety of technical ecosystems.

• **Accessibility Testing**: Verifying that KitchenSync is usable by people with a wide range of disabilities, including visual, auditory, physical, speech, cognitive, language, learning, and neurological disabilities.

**Testing Environment**

* **Devices and OS Compatibility:** To guarantee complete platform compatibility, testing was done on a range of devices. This covers the Android and iOS operating systems. Although testing was done on both Android and iOS devices, Android 14 and later versions were tested, with special emphasis placed on Android API 34.
* **Network Conditions:** Testing was conducted under different network conditions to imitate real-world utilization scenarios. This included testing beneath high-speed Wi-Fi associations as well as lower transmission capacity portable information associations
* **Heat and Resource Utilization under Different Conditions**: Test how the application impacts device heat generation and resource utilization (battery, CPU, memory) across different devices and under varying network conditions.
* **Security Testing Across Devices and Networks:** Ensure security testing covers the spectrum of devices and network conditions.

**Test Methodology**

A combination of automated and manual testing strategies was employed:

* **Automated Testing:** Automated testing was utilized essentially for regression testing and load testing purposes. Automated scripts were created to efficiently execute test cases and confirm the application's usefulness and execution.
* **Manual Testing:** In expansion to automated testing, manual testing was conducted by our testing group to complement the automated tests. Detailed exploratory testing was performed to closely imitate client intelligence and confirm the convenience and usefulness of new features.
* **Security Testing Integration:** Both our automated and manual testing strategies include specific security testing protocols. Automated tools can scan for common vulnerabilities, while manual testing can simulate sophisticated attack scenarios to evaluate the application’s resilience against security threats.
* **Localization and Internationalization Testing**: This involves verifying the application's ability to adapt to various languages, cultural norms, and regional standards without requiring significant changes.

**Testing Tools and Environment**

The Flutter framework and the Dart programming language are the main components of the test environment and tools utilized for the KitchenSync project.

**Key Findings**

* **Functionality:** The core functionalities of the app, such as kitchen inventory management, device synchronization, and item expiration tracking, were altogether tested and found to function consistently.
* **Usability:** The app's interface was evaluated to be instinctive and user-friendly. Testers detailed that unused clients were able to explore through the app effortlessly, showing a positive client involvement.
* **Performance:** Performance testing revealed that the app illustrates excellent responsiveness and stability over different devices and network conditions. Under stress testing, it scaled well with no critical execution corruption observed.
* **Security:** Security testing did not reveal any major vulnerabilities inside the app. The implementation of strong data protection measures guarantees that client information is secure and adjusted with industry standards. This instils confidence in clients concerning the privacy and security of their information.
* **Notifications**: Initially, there were a few issues experienced with the delivery of notifications on different platforms. However, these issues were instantly recognized and resolved.
* **Item Expiration**: The automatic check for item expiration was tested and found to be working as planned. Clients get opportune notices, which essentially help in overseeing the freshness of their kitchen inventory.
* **Integration Compatibility:** Testing demonstrated the app's smooth integration compatibility with a variety of kitchen appliances, increasing its adaptability and user-friendliness. Smart cabinets, refrigerators, and other kitchen appliances can be easily synchronized by users, which helps create a seamless kitchen management system.
* **Data Accuracy and Reliability**: The app's data management features were shown to be accurate and reliable during testing. KitchenSync gives consumers dependable information to help them choose their kitchen supplies wisely by keeping track of inventory items, expiration dates, and device statuses.

**Conclusion**

The KitchenSync app has established itself as a leading solution for kitchen inventory management through extensive testing and continuous improvement. Our commitment to excellence drives us to persistently enhance the app, ensuring it meets the evolving needs of our clients. We promise to provide unparalleled user experience, making kitchen tasks easy and efficient. Our testing efforts have made KitchenSync the go-to tool for seamless kitchen management, and we're dedicated to maintaining this standard for all our users.

Test Log (Unit Testing)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | System | Module | Coordinator | Results |
| 13-March | Home Page | Switch Users | Maha | Loads user list correctly from database, needs further error checking |
| 13-March | Home Page | Switch Themes | Maha | Dark and Light themes switch according to specification |
| 13-March | Home Page | UI Interactivity | Amna | UI buttons and device list work as intended, can be tweaked further |
| 13-March | Home Page | Functionality | Amna | List of devices refreshes and displays information correctly |
| 15-March | Inventory Page | Devices Display | Zyead | All device data retrieved and rendered properly |
| 14-March | Inventory Page | Load Data | Zyead | Category and item data numbers display the correct information |
| 15-March | Inventory Page | Device Switching | Zeyad | Functions properly |
| 16-March | Inventory Page | Device Contents | Zeyad | Displays all device contents with numbers accurately |
| 18-March | Inventory Page | Add / Remove Items | Ahmad | User can add and remove items accordingly |
| 21-March | Recipe Page | AI Chat Assistant | Ahmad | Responds to selected input, load time can be improved |
| 20-March | Recipe Page | UI | Mustafa | Displays needed categories and text box as required |
| 24-March | App Interface | Animations | Mustafa | Page transition and loading animations load properly |
| 23-March | App Interface | Buttons | Saif | All buttons functional, and redirect approprietly |
| 24-March | Donation Page | Nearest Bank Locator | Amna | User can find the nearest food banks to donate to |
| 21-March | Donation Page | Food Banks List | Saif | Loads and functions as intended |
| 21-March | Donation Page | Schedule Donation | Zyead | Reflects dates accurately and prompts user action |
| 25-March | Donation Page | Donation Rewards | Zyead | User can use earned donation points to claim rewards |
| 17-March | Scanning Page | Scan New RFID Item | Maha | Scans item thought RFID successfully and adds it to database |
| 17-March | Scanning Page | Scan New NFC Item | zeyad | Scans items through NFC and add data accordingly to database |
| 19-March | Settings | Notifications | Amna | Asks permission and sets notifications according to spec |
| 19-March | Settings | Manage Device | Mustafa | User can switch, add, or remove devices |
| 19-March | Settings | Manage Account | Mustafa | User can access and edit multiple accounts successfully |
| 20-March | Settings | Manage Privacy | Ahmad | User can modify and disable privacy settings successfully |
| 20-March | Settings | About Page | Ahmad | Displays and reports correct information |
| 23-March | Settings | Customer Support | Saif | User can easily talk and resolve issues with customer support |
| 24-March | Settings | Social Accounts | Saif | User can easily access all pages of social media |
| 25-March | Settings | Expiry Notifications | Saif | User can recieve item expiry notifications |