IEEE 829 STANDARD TEST PLAN TEMPLATE

|  |  |
| --- | --- |
| Test plan identifier | Cookbook Application CBA-001 |
| Introduction | 1.1 -1.3 |
| Test items | Components/Screens: Daily Specials, Recipes, Favourites, Shopping List  Stack Navigation? Bottom tab navigation? |
| Features to be tested | Daily Specials screen/stack:   * Make sure it generates 6 random recipes from the API * Make sure they are “clickable”/touchable with the buttons * Make sure the recipe selected is displayed with all required features   Recipes screen/stack:   * Display of all cuisines * Display of all recipes within the cuisine selected   Favourites screen:   * Async storage (buggy) * Check that favourite recipes are being added inside the screen when selecting icon   Shopping list:   * Async storage (buggy) * Items being deleted * Check that ingredients from a recipe are being added inside the screen when selecting icon |
| Features not to be tested | TBD |
| Approach | Testing Types:   * Exploratory testing * Unit Testing: To test individual components and functions. * Integration Testing: To check how various components work together, such as the stack/bottom tab navigation and the pressed components like pressing a recipe in the daily specials * System Testing: To evaluate the application as a whole.   Testing methods:  Testing Tools: For integration/unit testing, Jest and Testing library  Exploratory testing could be using different screens for the Android studio emulator/ screenshots of ios  <https://dev.to/stiv_ml/writing-e2e-tests-for-react-native-using-expo-2ia8> |
| Item pass/fail criteria |  |
| Suspension and resumption criteria |  |
| Test deliverables |  |
| Test tasks |  |
| Environmental needs |  |
| Responsibilities |  |
| Staffing and training needs |  |
| Schedule | E2E testing?  Exploratory testing  Unit  Integration |
| Risks and contingencies |  |
| Approvals |  |