

This QA review covers:

- Screens / UI:
 - Login Screen
 - Dashboard Screen (including Daily Recipe area)
 - Recipes Screen (list of recipes, open details, etc.)
 - Add Recipe Screen (form to add a new recipe)
 - Pantry Screen
 - Settings Screen (including notification toggle)
- User flows:
 - Start app → Login → Dashboard
 - Dashboard → Recipes → Recipe Detail → back navigation
 - Recipes → Add Recipe → Save recipe → see it in the list
 - Dashboard / Navigation bar → Pantry
 - Dashboard / Navigation bar → Settings → toggle options

Out of scope:

- External integrations (e.g., real backend server or cloud database) beyond what is implemented.
- Performance/load testing with large datasets.
- Cross-platform UI differences outside the main test environment (e.g., only tested on laptop)

Test Environment

- Platform: Local development environment (laptop/desktop).
- Framework: Kivy / KivyMD.
- Language: Python.
- Data Storage: JSONStore / Recipes.json for recipe storage.
- Window Size: 400 x 700 (as configured in MealMaster.py).

QA Checklist

| Item | Yes/No | Comments |
|---------------------------------------|--------|---|
| All mockups/screens are implemented | Yes | Login, Dashboard, Recipes, Add, Pantry, Settings present. |
| Each page has a clear purpose | Yes | No “mystery” pages; labels describe usage. |
| Navigation flow is correct | Yes | Login → Dashboard; nav to Recipes/Pantry/Settings works. |
| Wireframe / original plan is followed | Yes | Screen order and links match proposal. |
| Page titles and styles are consistent | Yes | Same fonts/colors used across screens. |
| Recipes feature works (view/add) | Yes | Save shows new recipe in list. |
| Pantry screens open without errors | Yes | Layout readable; no crash. |
| Settings toggle works | Yes | Notification switch logs on/off status. |
| No missing or duplicate screens | Yes | No unused or orphan screens. |
| Document follows required format | Yes | QA plan + checklist + summary included. |
| Grammar and spelling checked | Yes | Text reviewed for basic errors. |

Test Cases

TC-01 – Login with non-empty fields

- Precondition: App launched, on Login screen.
- Steps:
 1. Enter a non-empty username.
 2. Enter a non-empty password.
 3. Press the Login button.

- Expected result:
 - User is taken to Dashboard screen.

TC-02 – Navigate to Recipes from Dashboard

- Steps:
 1. From Dashboard, press the “Recipes” button.
- Expected result:
 - Recipes screen opens and displays recipe list or placeholder text.

TC-03 – Add a new Recipe

- Steps:
 1. From Recipes screen, press “Add New Recipe”.
 2. Fill in recipe name, ingredients, and steps.
 3. Press “Save Recipe”.
- Expected result:
 - No crash: recipe is stored.
 - New recipe appears in Recipes list.

TC-04 – Open Recipe Details

- Steps:
 1. On Recipes list, tap an existing recipe entry.
- Expected result:
 - Dialog or detail view shows recipe title and steps.

TC-05 – Open Pantry

- Steps:
 1. From Dashboard nav bar, press “Pantry”.
- Expected result:
 - Pantry screens open without errors and shows any existing items or an empty-state message.

TC-06 – Toggle Notifications in Settings

- Steps:
 1. From Dashboard nav bar, press “Settings”.
 2. Toggle the Notifications switch on and off.
- Expected result:
 - Switch visually toggles between states.
 - Console/log prints “Notifications Enabled” or “Notifications Disabled”.

QA Summary – MealMaster Recipe Management App

As the QA Tester, I reviewed the final version of the MealMaster app, focusing on completeness, usability, and accuracy of the implemented features. All main screens (Login, Dashboard, Recipes, Add Recipe, Pantry, Settings) are present and follow the intended navigation flow. Core user actions, logging in, viewing recipes, adding recipes, opening the pantry, and changing settings, work without crashes.

I executed the planned test cases for navigation, recipe management, pantry access, and the settings toggle. No **critical** defects remain that would block basic usage of the app. Minor issues, if any, are limited to visual polish and do not affect core functionality.

Based on this testing, the application meets the acceptance criteria for our project, and I consider QA complete for final submission.