Test Plan

DIY Max Application

Date	Version	Description	Author	Project Manager
24/07/2024	0.1	Test plans and test strategy of DIY App	Ali Shozab	

1. Objective:

The purpose of this test plan is to define the testing strategy and approach for the DIY Max Application, ensuring the application meets its functional requirements, performs reliably, and provides a satisfactory user experience.

Key Feature:

- Customize with Filters: Budget, Style, Dimensions input, Technology
- AR Tool Available: Visualize projects with Augmented Reality
- Tools, Materials, Supply List: Detailed lists, Links to tutorials
- Cost Comparison: Compare costs for up to 3 retailers
- Checkout: Integrated purchasing process
- Budget Tools: Manage and track project budgets
- Step-by-Step Instructions: Detailed instructions, Live progress tracking
- Chatbot for support

2. Scope:

This test plan covers functionality, usability, performance, security, and compatibility testing of the DIY Max Application. Features such as customizable filters, augmented reality tools, detailed material lists, cost comparison, integrated checkout, budget management, step-by-step instructions, and chatbot support will be tested.

3. Objectives:

- Verify that the app accurately estimates and provides material and budget details.
- Ensure AR tools function correctly for project visualization.

- Validate cost comparison features for up to 3 retailers.
- Confirm the integrated checkout process is smooth and secure.
- Test the effectiveness of budget management tools.
- Check the functionality of step-by-step instructions and live progress tracking.
- Ensure the chatbot provides appropriate support.

4. Test Items:

- App Name: DIY Max App
- Version: Specify the version being tested, Pending…
- Features to be Tested:
    - Customize with Filters (Budget, Style, Dimensions, Technology)
    - AR Tool for Project Visualization
    - Tools, Materials, and Supply List with Links to Tutorials
    - Cost Comparison for up to 3 Retailers
    - Integrated Checkout Process
    - Budget Management Tools
    - Step-by-Step Instructions with Live Progress Tracking
    - Chatbot Support

5. Language & Platform:

Mobile Application

- Dart
- Flutter (Frontend)
- Python/Django (Backend)

6. Features Not to be Tested:

Specify any features out of scope, if applicable we mention here later.

7. Test Strategy

- **Test Types:**
    - **Functional Testing:** Validate that all features work as expected, including customizable filters, AR tools, cost comparison, and checkout.
    - **Negative Testing:** Assess the user experience, put invalid data to check if all features are working fine or not.
    - **Performance Testing:** Evaluate app performance, including response times, load times, and the app’s behavior under various conditions.

  - **Regression Testing:** Use regression method for bug free app. Check all bugs and features again and again after each build.
  - **UI Testing:** Ensure the app functions correctly across different devices, operating systems, and screen sizes.
  - **Test all flow:** Test all flow and calculate estimates are working well.

- **Test Levels:**
  - **Integration Testing:** Ensure that integrated components work together seamlessly.
  - **System Testing:** Validate the entire system's functionality and performance.
  - **Acceptance Testing:** Confirm that the app meets user requirements and expectations.

8. Test Environment:

- **Hardware:** Tablet, window PC, Macbook, different size of screens to be tested.
- **Software:** Android(Google pixel, samsung)
- **Network:** Network conditions (e.g., Wi-Fi, 4G, 5G) for testing.

9. Test Criteria

- **Entry Criteria:**
  - The app is fully developed and stable.
  - All major features are implemented.
  - Test environments are set up.
- **Exit Criteria:**
  - All critical and major defects are resolved.
  - The app meets the defined acceptance criteria.
  - Test coverage and quality metrics are satisfactory.

10. Test Schedule

- **Test Timeline:** Outline the timeline, including key milestones for preparation, execution, and review.
- **Test Phases:**
  - **Test Preparation:** 1.5 week
  - **Test Execution:** 1 week
  - **Defect Fixing and Retesting:** Depend on fixing by developers
  - **Final Review and Approval:**

11. Test Deliverables:

- **Test Cases:** Document test cases for each feature, including detailed steps and expected results.
- **UAT Report:** Generate reports on test execution, defect tracking, and test coverage.

12. Roles and Responsibilities

- **Project Manager:** Oversee the developing & testing process, manage the all team, and ensure timely delivery of Application.

Name	Role	Duties
Muhammad Nouman	Project Manager	communicating with stakeholders,managing the developing & testing process, managing the all team, and ensuring timely delivery of Mobile Application.

- **Testers:** Execute test cases, report defects, and provide feedback.

Name	Role	Duties
Iqra Khan	SQA Engineer	Manages the QA process, Making test plans, strategy of testing, making test cases, leads and mentors the testing team, ensures test coverage and quality standards, coordinates with stakeholders, and drives continuous process improvement.
Ali Shozab	SQA	Execute test cases, report defects, and provide feedback. coordinates with stakeholders, and drives continuous process improvement. Making UAT Report

- **Developers:** Develop the Application according to requirements.

Name	Role	Duties
Faisal	Flutter Developer	Responsible for designing, developing, and maintaining mobile applications using the Flutter framework, ensuring high performance, and collaborating with cross-functional teams to deliver a seamless user experience.
Hamza Khan	Flutter Developer	Responsible for designing, developing, and maintaining high-performance mobile applications using the Flutter framework, optimizing user interfaces, integrating with backend services, collaborating with cross-functional teams, ensuring code quality.
Mudassir Habib	Python/Django Developer	Responsible for developing API's, and maintaining Backend of mobile applications using the Django framework, implementing server-side logic, ensuring database integration, optimizing application performance, collaborating with front-end developers, and adhering to best practices for code quality and security.

13. Risk Management

- **Potential Risks:**
  - Device compatibility issues.
  - Network-related performance issues.
  - Security vulnerabilities in the checkout process.
- **Mitigation Strategies:**
  - Use a range of devices for compatibility testing.
  - Conduct thorough performance and security testing.
  - Regularly update and patch the app to address vulnerabilities.

14. Approval

- **Sign-off:** Obtain approval from stakeholders, including the project manager, development team, and any other relevant parties.