A logo of a university

AI-generated content may be incorrect.

American International University-Bangladesh (AIUB)  
**Department of Computer Science  
Faculty of Science & Technology (FST)**

**Quick Loan Application**

A Software Quality and Testing Project Submitted

|  |  |  |
| --- | --- | --- |
| **EVALUATION CRITERIA** | **Total Marks (50)** | |
|  | |
| Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions | [10 Marks] |  |
| Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements) | [10 Marks] |  |
| Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases | [10 Marks] |  |
| Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk | [10 Marks] |  |
| Approval, Format, Submission, and Defense | [10 Marks] |  |

**Software Test Plan**

**for**

**Quick Loan Application**

Prepared by   
 Rafiah Salsabil Labanya

American International University-Bangladesh (AIUB)

22 September,2025

Table of Contents

[Revision History 3](#_Toc136846938)

[1. TEST PLAN IDENTIFIER: AT-TP01.3 4](#_Toc136846939)

[2. REFERENCE MATERIALS 4](#_Toc136846940)

[3. INTRODUCTION 4](#_Toc136846941)

[3.1 Background to the Problem 4](#_Toc136846942)

[3.2 Solution to the Problem 4](#_Toc136846943)

[4. REQUEIREMNT SPECIFICATION 4](#_Toc136846944)

[4.1 System Features 4](#_Toc136846945)

[4.2 System Quality Attributes 5](#_Toc136846946)

[4.3 System Interface 5](#_Toc136846947)

[4.4 Project Requirements 5](#_Toc136846948)

[5. FEATURES NOT TO BE TESTED 5](#_Toc136846949)

[6. TESTING APPROACH 5](#_Toc136846950)

[6.1 Testing Levels 5](#_Toc136846951)

[6.2 Test Tools 5](#_Toc136846952)

[6.3 Meetings 5](#_Toc136846953)

[7. TEST CASES/TEST ITEMS 6](#_Toc136846954)

[8. ITEM PASS/FAIL CRITERIA 6](#_Toc136846955)

[9. TEST DELIVERABLES 6](#_Toc136846956)

[10. STAFFING AND TRAINING NEEDS 6](#_Toc136846957)

[11. RESPONSIBILITIES 6](#_Toc136846958)

[12. TESTING SCHEDULE 7](#_Toc136846959)

[13. PLANNING RISKS AND CONTINGENCIES 7](#_Toc136846960)

[14. APROVALS 7](#_Toc136846961)

#### **1. Test Plan Identifier: AT-TP01.3**

#### **2. Reference Materials**

* Software Requirement Specification (SRS) Document [DOCUMENT filename="SQT - Test Plan Template [OBE]\_2.1 (1).docx"]

### **3. INTRODUCTION**

#### **3.1 Background to the Problem**

In the rapidly evolving financial landscape of Bangladesh, where mobile penetration exceeds 100 million users and digital financial services have surged post-COVID-19, access to quick and affordable credit remains a critical challenge for many individuals and small businesses. The Quick Loan Mobile App project is situated within the domain of fintech and mobile financial services (MFS), specifically targeting instant personal and micro-loans to address short-term financial gaps. This problem domain is characterized by the need for seamless, on-the-go financial solutions that bypass traditional banking hurdles like lengthy paperwork, branch visits, and high interest rates, which disproportionately affect urban and semi-urban populations with irregular income streams, such as freelancers, small traders, and young professionals.

The root cause of this problem lies in the structural inefficiencies of conventional lending systems in Bangladesh. Traditional banks and non-bank financial institutions (NBFIs) often require extensive documentation, collateral, and physical presence, leading to approval times of days or weeks, which is impractical for urgent needs like medical emergencies, education fees, or business inventory replenishment. This is exacerbated by Bangladesh Bank's regulatory framework, which, while promoting digital finance through MFS operators like bKash and Nagad, still leaves gaps in instant, collateral-free micro-loans for unbanked or underbanked segments—estimated at over 50% of the adult population according to World Bank reports. This issue is particularly important to consider because it perpetuates financial exclusion, stifles economic mobility, and hinders the growth of the burgeoning MSME sector, which contributes nearly 25% to Bangladesh's GDP. Addressing it aligns with national goals under the Digital Bangladesh initiative, enabling faster poverty alleviation and inclusive growth by empowering users to manage cash flow effectively without resorting to informal, high-risk lenders like moneylenders charging exorbitant rates (often 20-50% monthly).

This background provides a common view of the project: a mobile-first platform designed to democratize access to credit, fostering trust and efficiency in everyday financial transactions while ensuring compliance with local regulations like NID verification and OTP security.

#### **3.2 Solution to the Problem**

To address the challenges of delayed and cumbersome loan access, the Quick Loan Mobile App proposes a fully digital, user-centric mobile application that streamlines the entire loan lifecycle from registration and application to status tracking and repayment directly on smartphones. This solution is particularly appropriate because it leverages Bangladesh's high mobile usage (over 170 million subscriptions as of 2023) and existing MFS infrastructure, delivering loans in minutes via integrated bank APIs and e-KYC processes, thereby reducing approval times from days to under an hour for eligible users. It is feasible to meet business objectives, such as achieving 100,000 active users in the first year and maintaining a 95% on-time repayment rate, through scalable cloud-based architecture, low operational costs (no physical branches), and partnerships with banks like City Bank or Dhaka Bank for disbursements—all while adhering to Bangladesh Bank's digital lending guidelines and ensuring data security via encrypted uploads.

The software being specified is a native Android/iOS mobile app named "Quick Loan," designed to provide instant personal loans ranging from BDT 5,000 to 50,000 with tenures of 3-12 months, targeting salaried individuals and small entrepreneurs aged 18+. Its purpose is to offer a secure, intuitive platform for end-to-end loan management, including phone/email-based registration, NID and document uploads, purpose-specific applications (e.g., medical, education), real-time status tracking via progress indicators (under review, approved, rejected), and repayment schedules with total paid/remaining balances. Relevant benefits include 100% paperless processing for faster approvals, transparent interest rates (starting at 9-18% APR), no collateral requirements, and integrated notifications for reminders, reducing default risks. Objectives encompass enhancing financial inclusion by serving 1 million underserved users annually, while goals focus on achieving a Net Promoter Score (NPS) above 70 through user-friendly UI/UX and seamless OTP/NID verification, ultimately driving revenue via processing fees and upselling premium features like top-up loans.

Existing studies presented in the problem area highlight the transformative impact of digital lending in emerging markets like Bangladesh. A 2023 Bangladesh Institute of Bank Management (BIBM) study on "Digital Financial Services and Financial Inclusion" found that MFS adoption increased loan accessibility by 40% for low-income groups but noted persistent barriers like verification delays and low digital literacy. Similarly, a World Bank report (2022) on "Fintech and MSME Financing in South Asia" emphasized that mobile loan apps reduced borrowing costs by 15-20% compared to informal sources, though integration with national ID systems is key to scaling. Other research, such as a 2024 Asian Development Bank (ADB) paper on "Microfinance Digitization in Bangladesh," underscores the need for apps with robust tracking and repayment tools to achieve 80% recovery rates.

Existing software solutions available to solve the problem include several prominent mobile apps in Bangladesh that offer instant loans, though many lack comprehensive end-to-end tracking or user-friendly interfaces. Notable examples are:

* **bKash Loan**: Integrated within the bKash app (over 70 million users), it provides instant City Bank loans up to BDT 20,000 with minimal verification, focusing on quick disbursements via MFS but limited to existing bKash account holders.
* **e-Rin Personal Loan App**: A dedicated app for salaried professionals offering BDT 10,000-50,000 loans in 3-6 months, with e-KYC via NID and OTP, emphasizing paperless applications and direct bank credits, though it requires Dhaka Bank linkage.
* **Nagad and Rocket**: Top-ranked MFS apps (per SimilarWeb 2025 rankings) that include loan features for up to BDT 30,000, integrated with government postal services (Nagad) or Dutch-Bangla Bank (Rocket), providing broad reach but slower approvals for non-MFS users.
* **Phandora Credit**: A free lending platform for quick personal loans (BDT 5,000-100,000) with flexible 3-12 month terms and APRs of 18-26%, featuring easy applications and status checks, targeted at Bangladeshi users but criticized for aggressive debt collection.
* **Drutoloan**: Focuses on MSME business loans with at-location verification and high approval rates, using a streamlined digital system for instant account opening and disbursements under a minute, though more geared toward enterprises than individuals.

These solutions demonstrate the viability of mobile-based lending but often fall short in unified repayment tracking and customizable purposes, gaps that the Quick Loan app addresses through its intuitive dashboard and progress visualization.

### **4. REQUIREMENT SPECIFICATION**

#### **4.1 System Features**

* Here’s the List of the system functional requirements that describe the system’s functionalities.

1. **System Registration**   
   1.1 The software shall allow users to register by entering their name.   
   1.2 The software shall allow users to register by entering their phone number.   
   1.3 The software shall allow users to set and confirm a password during registration. 1.4 The software shall allow users to upload their NID for verification during registration.   
   1.5 The software shall provide an OTP verification process for phone number confirmation.  
    1.6 The software shall enable users to submit the registration by tapping the "Register" button.   
   **Priority Level:** High   
   **Precondition:** User must have a valid phone number and NID.   
   **Cross reference:** 4.1, 7.2
2. **System Login**    
   2.1 The software shall allow users to log in by entering their phone number.   
   2.2 The software shall allow users to log in by entering their password.  
    2.3 The software shall enable users to submit the login by tapping the "Login" button. **Priority Level:** High   
   **Precondition:** User must have a valid phone number and password.   
   **Cross reference:** 4.1, 7.2
3. **System Homepage** 3.1 The software shall display a welcome message upon successful login.   
   3.2 The software shall provide a button for users to apply for a loan.   
   3.3 The software shall provide a button for users to track loan status.   
   3.4 The software shall provide a button for users to view repayment details.   
   3.5 The software shall provide a button for notifications.  
    3.6 The software shall provide a control panel button for user settings or oversight.   
   **Priority Level:** High   
   **Precondition:** User must be logged in.   
   **Cross reference:** 4.1, 7.2
4. **System Loan Application**   
   4.1 The software shall allow users to enter the loan amount.   
   4.2 The software shall allow users to select the purpose of the loan from a dropdown. 4.3 The software shall allow users to upload images and documents.   
   4.4 The software shall enable users to submit the loan application by tapping the "Submit" button.   
   **Priority Level:** High   
   **Precondition:** User must be logged in.   
   **Cross reference:** 4.1, 7.2
5. **System Loan Status**   
   5.1 The software shall display a list of submitted loans.  
   5.2 The software shall show the loan status with stages: under review, rejected, or approved.   
   5.3 The software shall display the masked loan amount for reference.   
   **Priority Level:** High   
   **Precondition:** User must have submitted a loan application.   
   **Cross reference:** 4.1, 7.2
6. **System Repayment**   
   6.1 The software shall display the purpose of the loan.   
   6.2 The software shall display the repayment schedule.   
   6.3 The software shall display loan details.   
   6.4 The software shall display the payment status.   
   6.5 The software shall show the total paid and remaining amounts.   
   6.6 The software shall provide a help and FAQ section.   
   **Priority Level:** High   
   **Precondition:** User must have an active loan.   
   **Cross reference:** 4.1, 7.2

### **4.2 System Quality Attributes**

* Here’s the list of the quality attributes that describe how well the system should perform.

**QA1 - Usability:** A trained user shall be able to complete registration, including name entry, phone number, password confirmation, NID upload, and OTP verification, in an average of three and a maximum of five minutes.

**Priority Level:** High   
**Precondition:** N/A   
**Cross reference:** QA3, QA4

**QA2 - Performance:** The system shall load each screen (e.g., login, homepage, loan application) and process submissions (e.g., login or loan application) within two seconds on a standard mobile device with stable internet.

**Priority Level:** High   
**Precondition:** User device meets minimum specs (Android/iOS with 4G+ connection).   
**Cross reference:** QA1, QA5

**QA3 - Security:** The system shall ensure all sensitive data (e.g., phone number, password, NID uploads, loan amounts) is encrypted during transmission and storage, with OTP verification mandatory for registration and masked displays for loan amounts.

**Priority Level:** High   
**Precondition:** N/A   
**Cross reference:** QA1, QA4

**QA4 - Reliability:** The system shall maintain 99% uptime, ensuring that features like status tracking and repayment viewing display accurate, real-time data without crashes during navigation or uploads.

**Priority Level:** High   
**Precondition:** Backend server is operational.   
**Cross reference:** QA2, QA5

**QA5 - Maintainability:** The system shall allow for easy updates to UI elements (e.g., buttons, fields, progress indicators) without affecting core functionalities like submission or display.

**Priority Level:** Medium   
**Precondition:** N/A   
**Cross reference:** QA1, QA3

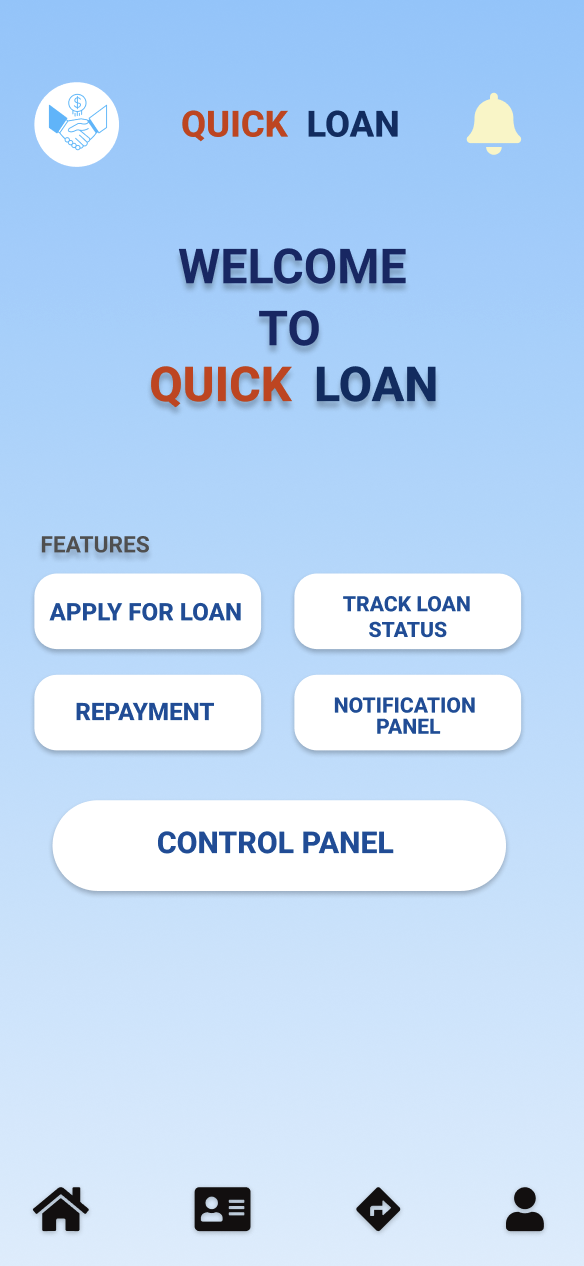
### 

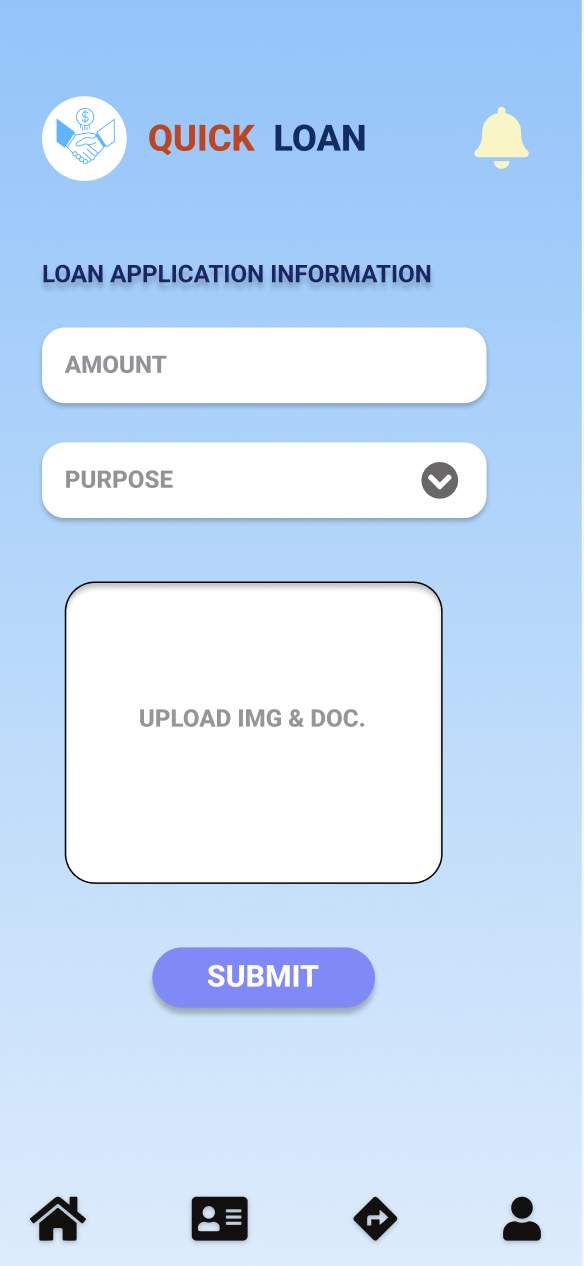
### 

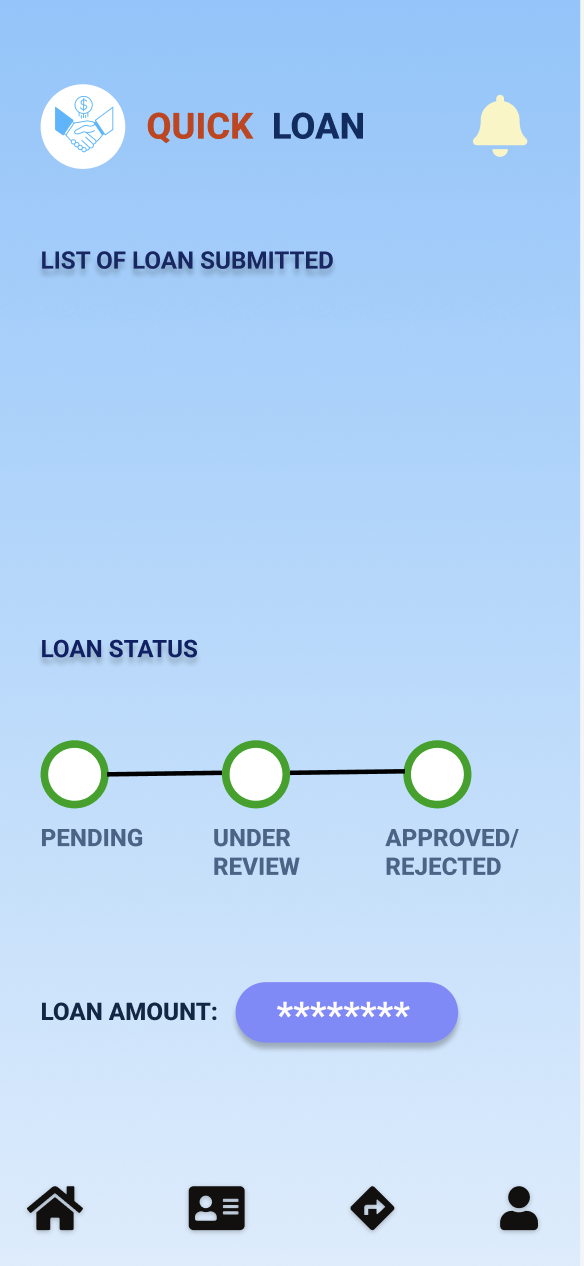
**4.3 System Interface**

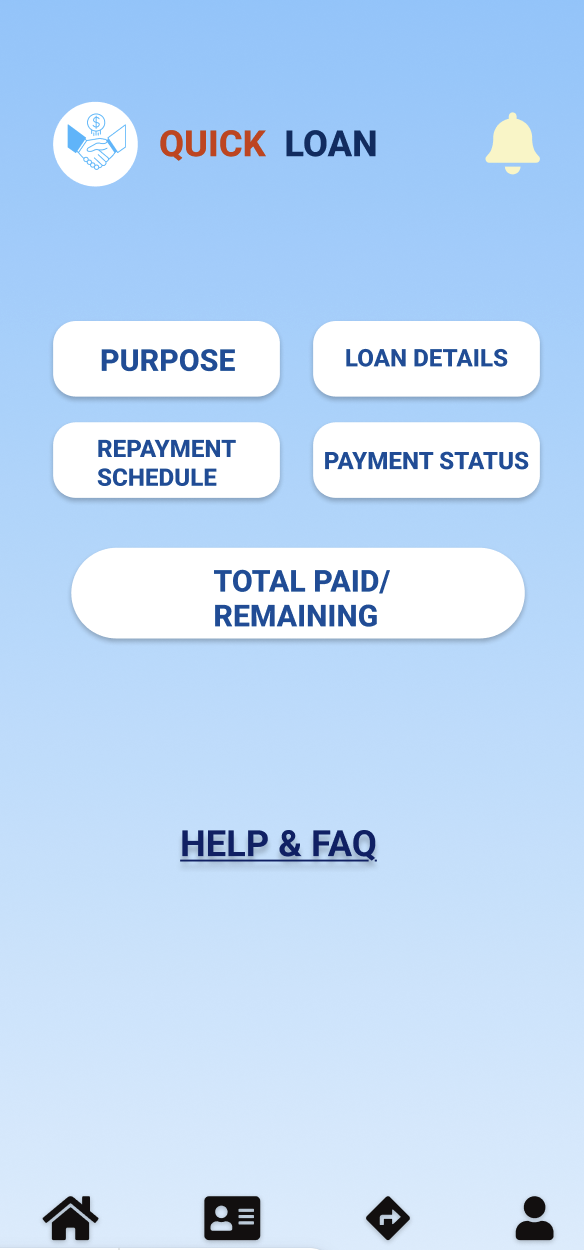
### 

### 









### **4.4 Project Requirements**

Using the COCOMO (Constructive Cost Model) for effort estimation, assuming an organic project mode with an estimated size of 5 KLOC (thousands of lines of code) for the Quick Loan Mobile App's core functionalities (registration, login, homepage, loan application, status tracking, and repayment):

* **Effort Estimation**: Approximately 13 person-months.
* **Development Time**: Approximately 7 months.
* **Required Personnel**: Average of 2 developers.

**Project Constraints:**

1. **Time**: The project must be completed within 7 months to align with the academic semester schedule (Fall 2023-2024), including phases for design, development, testing, and deployment.
2. **Budget**: Estimated at 780,000 BDT, based on COCOMO effort (13 person-months) and an average developer salary of 60,000 BDT per month in Bangladesh; this covers personnel costs with minimal allocation for tools (assuming free/open-source software).
3. **Resources**: 2-3 team members (e.g., developers and a tester), utilizing free development tools such as Android Studio for app building, Figma for UI/UX design, and cloud services like Firebase for backend prototyping.
4. **Environment**: Development on standard laptops (Windows or macOS) with mobile emulators; testing on physical Android/iOS devices; deployment targeting Android 10+ and iOS 14+; compliance with local regulations for data handling (e.g., NID uploads via secure APIs).

### **5. FEATURES NOT TO BE TESTED**

The List of the areas/features that will not be specifically addressed in testing is given below.

1. **Backend Loan Approval Processing**: The actual algorithmic evaluation of loan applications (e.g., credit scoring or risk assessment) will not be directly tested, as it relies on external backend services managed by financial partners and is outside the app's frontend scope; any testing would be indirect through simulated status updates in the loan status screen.
2. **Third-Party Integrations for Disbursement**: Integration with external banking APIs or MFS providers (e.g., bKash or bank transfers) for actual fund disbursement upon approval will not be tested, as this is under client (financial institution) control and beyond the project's mobile app boundaries.
3. **Push Notification Delivery**: The external delivery of notifications (e.g., via Firebase Cloud Messaging or similar services) outside the app's notification button will not be specifically addressed, with testing limited to in-app display only, as real-time push relies on device OS and network services not in scope.
4. **Actual Payment Processing for Repayments**: Handling of real monetary transactions for repayments (e.g., integration with payment gateways) will not be tested, as it involves secure financial systems under external control; indirect verification occurs via the repayment screen's display of totals and status.
5. **Device-Specific Hardware Interactions**: Features like camera access for NID or document uploads, or GPS for any implied location-based verification (not visible in UI), will not be directly tested beyond basic upload functionality, as they depend on device hardware variations outside the project's core app testing.
6. **Scalability and Load Testing**: Performance under high user loads (e.g., thousands of simultaneous applications) will not be addressed, as this requires production-level infrastructure not within the academic project's scope.

## **6. TESTING APPROACH**

### **6.1 Testing Levels**

The Quick Loan App will be tested across four main levels:

1. **Unit Testing**:  
   Each module (e.g., registration, login, loan application, repayment) will be tested individually to verify correctness of inputs, validations, and outputs. Example: OTP verification logic in registration.
2. **Integration Testing**:  
   Focus on interactions between modules, such as ensuring successful transition from login → homepage → loan application, and correct data flow (e.g., loan amount entered appearing in status tracking).
3. **System Testing**:  
   End-to-end testing of the entire app against functional and non-functional requirements (performance, security, usability). Example: testing whether loan applications can be submitted with documents uploaded under real device conditions.
4. **Acceptance Testing**:  
   Conducted from the user’s perspective to validate that the app meets business needs. Example: students acting as borrowers will test whether they can register, apply for a loan, and view repayment information smoothly.

### **6.2 Test Tools**

The following tools will be used in different stages of the project:

* **Development & Build Tools**: Android Studio, Xcode
* **UI/UX Design Reference**: Figma
* **Testing Tools**:
  + JUnit / Espresso (for Android unit & UI tests)
  + XCTest (for iOS testing)
  + Postman (for API testing where applicable)
* **Project Management & Collaboration**: Trello / Jira, Google Meet, GitHub
* **Device Testing**: Physical Android & iOS devices, plus emulators for different OS versions

### **6.3 Meetings**

* **Frequency**: Monthly team meetings, with additional ad-hoc meetings during major testing phases.
* **Format**: Online via Google Meet / in-person depending on schedule.
* **Agenda**:
  + Review of previous week’s test progress
  + Identification of bugs/issues and assignment of fixes
  + Planning test activities for upcoming week
  + Verification of completed test cases
  + Discussion of risks, blockers, and solutions

### **7. TEST CASES / TEST ITEMS**

### **Test Case 1 – Registration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Quick Loan Application | | | Test Designed by: Team Members | | |
| Test Case ID: TC\_REG\_01 | | | Test Designed date: | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Q/A Testers | | |
| Module Name: Registration | | | Test Execution date: | | |
| Test Title: Verify registration with valid data | | |  | | |
| Description: Test registration page functionality | | |  | | |
| Precondition (If any): User must have valid phone number and NID | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open app → Sign Up page  2. Enter name, phone, password, confirm password  3. Upload NID  4. Enter OTP  5. Click Register | Name: Rafiah Phone: 017XXXXXXXX Password: Test@123 OTP: 123456 | User account created, redirected to Login screen | | As expected, | Pass |
| Post Condition: User registered successfully, credentials stored in DB. | | | | | |

**Test Case 2 – Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Quick Loan Application | | | Test Designed by: Team Members | | |
| Test Case ID: TC\_LOG\_01 | | | Test Designed date: | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Q/A Testers | | |
| Module Name: Login | | | Test Execution date: | | |
| Test Title: Verify login with valid credentials | | |  | | |
| Description: Test login page | | |  | | |
| Precondition (If any): User must be registered with valid phone and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open app → Login page  2. Enter phone number  3. Enter password  4. Tap Login | Phone: 017XXXXXXXX Password: Test@123 | User successfully logged in and redirected to Homepage | | As expected, | Pass |
| Post Condition: User is logged in, session stored. | | | | | |

**Test Case 3 – Homepage Navigation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Quick Loan Application | | | Test Designed by: Team Members | | |
| Test Case ID: TC\_HOM\_01 | | | Test Designed date: | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Q/A Testers | | |
| Module Name: Homepage | | | Test Execution date: | | |
| Test Title: Verify homepage buttons | | |  | | |
| Description: Test homepage navigation | | |  | | |
| Precondition (If any): User must be logged in | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. From homepage, tap each button (Apply for Loan, Track Status, Repayment, Notification) | N/A | Each button should open correct screen | | As expected, | Pass |
| Post Condition: User navigates to desired module. | | | | | |

**Test Case 4 – Loan Application**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Quick Loan Application | | | Test Designed by: Team Members | | |
| Test Case ID: TC\_APP\_01 | | | Test Designed date: | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Q/A Testers | | |
| Module Name: Loan Application | | | Test Execution date: | | |
| Test Title: Verify loan application submission | | |  | | |
| Description: Test loan application page | | |  | | |
| Precondition (If any): User must be logged in | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to Apply for Loan  2. Enter loan amount 3. Select purpose  4. Upload docs  5. Submit | Amount: 20,000 Purpose: Education Upload: valid PDF | Application submitted successfully, confirmation shown | | As expected, | Pass |
| Post Condition: Loan request stored in system. | | | | | |

**Test Case 5 – Loan Status**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Quick Loan Application | | | Test Designed by: Team Members | | |
| Test Case ID: TC\_STA\_01 | | | Test Designed date: | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: Q/A Testers | | |
| Module Name: Loan Status | | | Test Execution date: | | |
| Test Title: Verify loan status progress | | |  | | |
| Description: Test loan status screen | | |  | | |
| Precondition (If any): User must have submitted a loan | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to Track Status 2. Select submitted loan | Loan ID: 1001 | Status shown as Under Review / Approved / Rejected | | As expected, | Pass |
| Post Condition: User views current loan status. | | | | | |

**Test Case 6 – Repayment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Quick Loan Application | | | Test Designed by: Team Members | | |
| Test Case ID: TC\_REP\_01 | | | Test Designed date: | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Q/A Testers | | |
| Module Name: Repayment | | | Test Execution date: | | |
| Test Title: Verify repayment details display | | |  | | |
| Description: Test repayment page | | |  | | |
| Precondition (If any): User must have an active loan | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to Repayment 2. View purpose, schedule, loan details, payment status, totals | Loan ID: 1001 | Repayment details correctly displayed | | As expected, | Pass |
| Post Condition: User reviews loan repayment status. | | | | | |

### **8. ITEM PASS/FAIL CRITERIA**

* **Pass Criteria**:
  + A test case will be marked as **Pass** if the actual results match the expected results without deviation.
  + All functional requirements are met as described in the SRS (e.g., successful registration, correct loan submission, repayment display).
  + Non-functional requirements such as usability, performance (≤2 seconds response time), and security (data encryption, OTP validation) are satisfied.
  + No **critical or high-severity defects** remain unresolved for that test case.
* **Fail Criteria**:
  + A test case will be marked as **Fail** if the actual results deviate from the expected results.
  + Missing or incorrect functionality (e.g., loan not being submitted despite valid inputs).
  + System crashes, freezes, or error messages preventing the completion of a test step.
  + Any **critical defect** that blocks the main user flow (e.g., user unable to register, log in, or apply for loan).
* **Conditional Pass (Needs Review)**:
  + If a minor or cosmetic defect is found (e.g., UI alignment issue, spelling mistakes) but the core functionality still works as expected, the test case may be marked as “Pass with Notes” for review in the next cycle.
* **Exit Criteria for Test Cycle**:
  + At least 95% of high-priority test cases must pass.
  + All critical defects must be resolved and retested.
  + Medium/low priority defects may be deferred to future releases if they don’t affect the core loan process.

## **9. TEST DELIVERABLES**

During the testing process, several documents and supporting materials will be prepared and shared among the team to ensure a clear record of activities, results, and decisions. These deliverables are important because they make the entire testing workflow transparent and traceable for all stakeholders. The key deliverables include:

* **Test Plan Document** – Outlines the objectives, scope, resources, approach, and schedule of testing activities. It acts as the guiding reference throughout the project.
* **Test Design Artifacts (Test Cases & Test Scripts)** – A detailed set of test cases and automated/manual scripts that describe how each functionality will be verified. These ensure coverage of both functional and non-functional requirements.
* **Test Data Sets** – Sample valid and invalid data, edge cases, and boundary values that will be used to simulate real user behavior.
* **Defect/Issue Reports** – A log of identified defects with proper classification (severity, priority, and status) to help developers in the debugging process.
* **Test Execution Records** – Results from each testing cycle, showing which tests passed, failed, or need further review.
* **Requirements Traceability Matrix (RTM)** – A mapping between requirements and test cases, ensuring that no requirement has been left untested.
* **Test Summary Report** – A consolidated document at the end of testing, highlighting overall test coverage, defect density, major risks, and system readiness for release.
* **Final Approval and Sign-off Report** – A formal document that confirms testing objectives have been met and the application is ready for deployment.

These deliverables not only support quality assurance but also provide accountability and evidence of the testing process for academic and professional evaluation.

## **10. STAFFING AND TRAINING NEEDS**

Successful testing requires the right balance of people, skills, and knowledge. For this project, the testing team will be small but well-coordinated, consisting of a **Test Lead**, **two Test Engineers**, and developer support when needed.

* **Recruitment Strategy:**  
  A mixed approach will be followed. On one hand, a **horizontal recruitment strategy** will be used to select testers with general software engineering backgrounds who can be trained to understand the Quick Loan application. On the other hand, for specific modules such as security and performance, a **vertical approach** will be applied, where individuals with prior expertise will be involved. This ensures both breadth and depth in the team’s capabilities.
* **Training Needs:**  
  Since the project is based on a financial application, proper domain knowledge is essential. Training sessions will be arranged to familiarize testers with fintech processes, such as e-KYC, OTP verification, and repayment tracking.  
  Additionally, technical training will cover tools like **JUnit, Espresso, XCTest, and Postman**, as well as project management platforms such as **Jira/Trello**. Testers will also receive instructions on secure handling of sensitive data (e.g., NID information).
* **Continuous Learning:**  
  Short weekly knowledge-sharing sessions will be held where team members can discuss challenges, clarify doubts, and review test cases together. This peer learning approach will keep everyone aligned and motivated.

By combining structured recruitment with proper training and ongoing support, the project ensures that the testing team will be competent, confident, and fully prepared to deliver reliable results.

## **11. RESPONSIBILITIES**

To avoid confusion and ensure accountability, roles and responsibilities of different stakeholders in the testing process are clearly defined. The following table summarizes each role’s contribution:

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| **Project Manager** | Ensures testing stays aligned with project goals, manages timelines and resources, and provides support in resolving major risks or blockers. |
| **Test Manager** | Defines the overall testing strategy, approves the test plan, monitors progress, and ensures quality objectives are met. |
| **Test Lead** | Breaks down the test plan into detailed tasks, assigns responsibilities to test engineers, monitors execution, and reports test status to management. |
| **Test Engineers** | Design detailed test cases, prepare test data, execute test scripts, log defects, and retest fixes. They are the main executors of the testing effort. |
| **Developers** | Support the testing team by fixing reported bugs, conducting unit tests, and ensuring smooth integration across modules. |
| **End Users (UAT Testers)** | Participate in acceptance testing, validate that the system meets business needs, and provide usability feedback. |
| **Stakeholders/Clients** | Review the test deliverables, evaluate progress reports, and provide final approval or sign-off before release. |

# **12. TESTING SCHEDULE**

# **13. PLANNING RISKS AND CONTINGENCIES**

Table 2: Risk Mitigation Plan for testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Unrealistic time estimate | 60% | Missed test coverage | Prioritize high-risk modules; keep buffer time |
| 2 | Device compatibility issues | 30% | Bugs on untested devices | Test on at least one Android & one iOS device; use emulators |
| 3 | Team member unavailability | 25% | Delays in testing | Crosstrain team members; assign backups |
| 4 | Incomplete documentation | 20% | Missing test cases | Daily progress reviews and peer-checking |
| 5 | Defect leakage due to short cycles | 35% | UAT rejection | Perform regression testing before UAT |
| 6 | Last-minute requirement changes | 15% | Rework & time overrun | Freeze requirements after Day 3 |

# 