



Specification Document: Web Application For Microfinance Loan Appraisal

MUBAKU IT

Specification Document: Web Application for Microfinance Loan Appraisal

Contents

Table of Tables	2
Abbreviations	3
1. Introduction	4
2. Project Objectives	4
3. System Actors	4
4. Loan Types and Associated Forms	4
4.1 Loans within Savings	4
4.2 Loans Above Savings	4
4.3 Loans Covered by Salary	4
4.4 Loans Covered by Standing Order	4
4.5 Mortgage Loans	4
5. Data Modeling	5
6. Loan Rating System	5
7. Use Cases	5
7.1 Loan Submission (Loan Officer)	5
7.2 Loan Approval or Rejection (Branch Manager)	5
7.3 Viewing and Printing Loan Details (Branch Manager)	5
7.4 Customer Notification (Automated via Email/WhatsApp)	5
8. System Activities	6
9. Project Timeline (10/02/2025 - 10/04/2025)	7
10. Scalability, Performance, and Security	8
10.1 Scalability	8
10.2 Performance Optimization	8
10.3 Security Measures	8
11. Loan Calculation Formulas	8
GLOSSARY	9
12. Conclusion	12

Table of Tables

Table 1: System Activities.....	6
Table 2: Project Timeline (10/02/2025 - 10/04/2025).....	7

Abbreviations

1. **API:** Application Programming Interface
2. **AES:** Advanced Encryption Standard
3. **AWS:** Amazon Web Services
4. **RBAC:** Role-Based Access Control
5. **JWT:** JSON Web Token
6. **XSS:** Cross-Site Scripting
7. **CFA:** Central African Franc (currency)

1. Introduction

This document outlines the complete specifications for the development of a web application based on Django, designed for microfinance institutions within a city.

2. Project Objectives

- Automate loan management for microfinance institutions.
- Provide a centralized system for loan submission, validation, and tracking.
- Notify customers via email or WhatsApp.
- Ensure high scalability, performance, and security.

3. System Actors

- **Institutions:** Registered microfinance institutions on the platform.
- **Loan Officer:** Submits loan applications.
- **Branch Manager:** Approves or rejects loans and manages printouts.
- **Customer:** Has no direct access but receives notifications via email/WhatsApp.

4. Loan Types and Associated Forms

4.1 Loans within Savings

- Customers can request a loan within the amount they have saved.
- The loan is instantly approved if it does not exceed the savings balance.

4.2 Loans Above Savings

- Customers can request loans beyond their savings balance.
- Requires additional guarantees and is subject to the branch manager's approval.

4.3 Loans Covered by Salary

- Granted to customers with a stable salary.
- The amount is calculated based on a percentage of the applicant's monthly salary and repayment capability.

4.4 Loans Covered by Standing Order

- Requires a fixed-income source that can be used as collateral.
- Loan approval is subject to the evaluation of the customer's financial obligations.

4.5 Mortgage Loans

- Requires property as collateral.

Specification Document: Web Application for Microfinance Loan Appraisal

- The loan amount is determined by the property's market value and risk assessment.

NB: Each loan type has a dedicated application form with required fields tailored to its specific conditions.

5. Data Modeling

- **Institution:** Name, address, contacts.
- **LoanOfficer:** Name, affiliated institution, credentials.
- **Loan:** Amount, type, rating (0-100%), status (submitted, approved, rejected), associated Loan Officer.
- **Customer:** Name, contact, loan history, associated Loan Officer.

6. Loan Rating System

- **0 - 49.999%:** Instantly rejected.
- **50% - 75%:** Averagely approved, saved in the system, and notified to the Branch Manager.
- **76% - 100%:** Eligible, saved in the system, and notified to the Branch Manager.

7. Use Cases

7.1 Loan Submission (Loan Officer)

Includes: Loan data entry, validation check.

7.2 Loan Approval or Rejection (Branch Manager)

Includes: Loan review, decision making, loan detail checking. **Extends:** Notification system (email/WhatsApp).

7.3 Viewing and Printing Loan Details (Branch Manager)

Includes: Loan search, data retrieval. **Extends:** Print functionality.

7.4 Customer Notification (Automated via Email/WhatsApp)

Generalization: Loan status notification extends from approval/rejection decisions.

8. System Activities

Activity	Step Description	Actions	Actors
1. Loan Submission	Loan Officer enters all necessary details for the loan application.	- Enter loan amount, type, and customer details.- Provide guarantees or collateral if applicable.- Submit loan application for validation.	Loan Officer
2. Rating Calculation	System processes the loan application and assigns a rating based on input details.	- Calculate the loan rating using predefined formulas.- Assign a rating (0-100%) based on loan type and customer information.	System
3. Branch Manager Review	Branch Manager evaluates the loan application based on the rating and loan details.	- Review loan rating and associated data.- Assess customer's loan history, financial situation, and collateral (if applicable).	Branch Manager
4. Approval/Rejection Decision	Branch Manager decides whether the loan is approved or rejected.	- If rating < 50%: Reject loan.- If rating between 50%-75%: Mark as pending, notify the Branch Manager.- If rating > 75%: Approve loan.	Branch Manager
5. Loan Storage	Store the loan application (approved or pending) in the system for record-keeping.	- Store approved or pending loans in the database.- Include all relevant details (loan amount, customer, approval status).	System
6. Loan Notification	Customer receives a notification about the status of their loan application.	- Send automated notifications via email or WhatsApp.- Notify customers on approval, rejection, or pending status.	System, Customer
7. Print Loan Details	Branch Manager can print the loan details for documentation or communication purposes.	- Retrieve loan data from the database.- Print loan details (e.g., loan amount, terms, repayment schedule).	Branch Manager

Table 1: System Activities

Actors:

- **Loan Officer:** Responsible for submitting the loan application with all necessary details.
- **Branch Manager:** Reviews the loan application, approves or rejects loans, and has the ability to print loan details.
- **System:** Handles rating calculations, storing of loan data, and automated notifications.
- **Customer:** Receives notifications about the status of their loan application but has no access to the system.

9. Project Timeline (10/02/2025 - 10/04/2025)

Period	Activity	Description	Deliverables
10/02/2025 - 20/02/2025	Requirement analysis, architecture design	Define system requirements, identify core functionalities, create initial wireframes, and database design.	- System requirements document - Initial wireframes and system design - Database schema and entity-relationship diagrams - Core functionality and system architecture plan
21/02/2025 - 15/03/2025	Backend development (Django)	Implement data models, loan rating system, user authentication, API endpoints, and backend logic.	- Django models for Institutions, Loan Officers, Loans, and Customers - Loan rating system implementation - User authentication and RBAC - API endpoints for loan submission, approval/rejection, and notifications - Backend logic for loan calculations and approval workflows
16/03/2025 - 31/03/2025	Frontend development (HTML, CSS, JS)	Develop user interfaces, forms for loan applications, dashboards for loan officers and branch managers.	- User interfaces for loan application forms and dashboards - Forms for each loan type (Loans within savings, Loans above savings, etc.) - Admin and Branch Manager dashboards for loan management - Responsive frontend design (mobile and desktop-friendly)
01/04/2025 - 07/04/2025	Testing and debugging	Conduct unit tests, integration tests, and fix bugs. Validate loan calculation accuracy and notification system.	- Unit tests for backend logic (loan rating, submission, etc.) - Integration tests for complete loan processing workflows - Bug fixes and optimization reports - Loan calculation validation and notification system testing
08/04/2025 - 09/04/2025	Deployment and training	Deploy system to production server, train loan officers and branch managers on system usage.	- Deployed web application on production server (e.g., AWS, Azure) - Training materials for loan officers and branch managers - Training sessions for end-users on how to use the system - User documentation and technical support materials
10/04/2025	Official launch	System goes live and is fully operational.	- Fully operational system - Go-live status report - Post-launch support plan and monitoring tools

Table 2: Project Timeline (10/02/2025 - 10/04/2025)

10. Scalability, Performance, and Security

10.1 Scalability

- **Modular design:** The system will be developed with a microservice-ready architecture to allow for future expansion.
- **Database optimization:** Use indexing and caching for improved query performance.
- **Cloud deployment:** The application will be deployed on a scalable cloud environment (e.g., AWS, Azure).
- **Load balancing:** Distribute traffic efficiently to prevent overload.

10.2 Performance Optimization

- **Efficient database queries:** Optimize queries to prevent performance bottlenecks.
- **Caching:** Implement Redis or Memcached to store frequently accessed data.
- **Asynchronous processing:** Use background tasks for notifications and loan processing.
- **Optimized frontend:** Minimize HTTP requests and use CDNs for static files.

10.3 Security Measures

- **Data encryption:** Use AES-256 encryption for sensitive data storage.
- **Authentication & authorization:** Implement role-based access control (RBAC) using Django's authentication framework.
- **Secure API endpoints:** Use HTTPS and JWT authentication for secure communication.
- **Input validation:** Prevent SQL injection and XSS attacks.
- **Logging & monitoring:** Enable security logs and real-time monitoring to detect anomalies.

11. Loan Calculation Formulas

- **Loans within savings:** $\text{Amount} = \min(\text{savings}, \text{limit})$
- **Loans above savings:** $\text{Amount} = \text{savings} * 1.5$
- **Loans covered by salary:** $\text{Amount} = \text{salary} * 12 \text{ months} * \text{rate}$
- **Loans covered by standing order:** $\text{Amount} = (\text{fixed income} - \text{expenses}) * \text{number of months}$
- **Mortgage loans:** $\text{Amount} = \text{property value} * \text{coverage rate}$

GLOSSARY

1. Web Application:

A software application that runs on a web server rather than on a user's device, which can be accessed through a web browser.

2. Django:

A high-level Python web framework that allows rapid development of secure and maintainable websites. It follows the model-template-views (MTV) architectural pattern.

3. Microfinance Institutions:

Organizations that provide small loans (micro-loans) to individuals or small businesses that do not have access to traditional banking services.

4. Loan Officer:

An employee of a financial institution who is responsible for managing loan applications, including reviewing, approving, and processing loan requests.

5. Branch Manager:

An employee who manages a branch of a financial institution, including overseeing loan approvals, customer service, and daily operations.

6. Customer:

An individual who applies for loans or services from the financial institution but does not have direct access to the internal management system. They are notified of their loan status via email or WhatsApp.

7. Loan Types:

Different categories of loans offered by the institution based on specific criteria:

- **Loans within Savings:** Loans that customers can take based on their available savings.
- **Loans Above Savings:** Loans where the amount exceeds the customer's savings, requiring additional guarantees.
- **Loans Covered by Salary:** Loans granted based on the applicant's salary and repayment capacity.
- **Loans Covered by Standing Order:** Loans secured by a fixed source of income, such as a direct deposit or standing order.
- **Mortgage Loans:** Loans secured by real property, with the loan amount determined by the property's market value.

8. Rating System:

A system that evaluates loan applications based on specific criteria and assigns a rating that determines whether the loan is approved, rejected, or pending.

Specification Document: Web Application for Microfinance Loan Appraisal

9. Data Modeling:

The process of creating data models to define how data will be stored, organized, and interacted with in the system. It includes defining entities (like Institutions, Loan Officers, and Customers) and their relationships.

10. Loan Submission:

The process by which a Loan Officer enters a customer's loan request into the system, including details like the loan amount and type.

11. Approval/Rejection Decision:

The decision-making process, where the Branch Manager reviews the loan application and determines whether it should be approved or rejected based on predefined rules or ratings.

12. Scalability:

The ability of the system to handle increased demand or load by scaling its resources, such as through modular design, cloud deployment, and load balancing.

13. Performance Optimization:

Techniques used to improve the system's speed and responsiveness, such as optimizing database queries, caching frequently accessed data, and reducing frontend load.

14. Security Measures:

Actions and practices put in place to protect the system from threats such as data breaches, unauthorized access, and malicious attacks. This includes encryption, secure authentication, and input validation.

15. Authentication & Authorization:

- **Authentication:** Verifying the identity of a user (e.g., by username and password).
- **Authorization:** Determining the level of access or permissions granted to an authenticated user.

16. Input Validation:

The process of ensuring that the data entered by a user into the system is correct, properly formatted, and free of malicious content (such as SQL injections).

17. Unit Tests:

Tests that check individual components of the system to ensure they function as expected.

18. Integration Tests:

Tests that verify that different parts of the system work together as intended.

19. API Endpoints:

Specific points of interaction in an application that allow external systems to communicate with it, such as retrieving or submitting data.

20. Cloud Deployment:

Hosting the web application on cloud platforms (like AWS or Azure) to take advantage of their scalability, flexibility, and remote access.

21. Load Balancing:

Distributing incoming network traffic across multiple servers to prevent any one server from becoming overwhelmed and ensure better performance and availability.

22. Database Optimization:

Improving the speed and efficiency of the database by indexing, caching, and optimizing queries to handle large amounts of data.

23. Caching:

Storing data in a temporary storage location (cache) to speed up repeated access to frequently requested data.

24. Asynchronous Processing:

The ability to handle tasks in the background without blocking the main application flow, such as sending notifications or processing loans without delaying the user interface.

25. Role-Based Access Control (RBAC):

A security model that restricts system access based on the roles assigned to users. Each role has specific permissions that define what actions the user can perform.

26. AES-256 Encryption:

A strong encryption algorithm used to protect sensitive data by converting it into a secure format that can only be decrypted with a specific key.

27. HTTPS:

A secure version of HTTP, used for secure communication over the internet, ensuring the data sent between the user's browser and the server is encrypted.

28. JWT (JSON Web Token):

A compact, URL-safe token used for secure transmission of information between parties, especially in authentication systems.

12. Conclusion

This specification document serves as the foundation for the project and will guide the development of the loan management system for microfinance institutions. Regular updates will be made to ensure alignment with client needs.