DOCUMENT MANAGEMENT SYSTEM

Surbhi Gandhi*[1], Rutika Gavali*[2], Pranjali Ghume*[3], Rutuja Hinge*[4] Prof. Amol Jadhav*[5]

***[12345] Zeal college of Engineering and Research, India.**

ABSTRACT

In an era of rapid digitization, managing personal and sensitive documents effectively and securely is essential. This paper presents a Smart Document Management System (DMS) that offers functionalities like upload, scan, OCR integration, categorization, and search within a secure mobile-based application. Built using Flutter for the frontend and Firebase for the backend, the system ensures user-friendly interaction and reliable cloud-based storage. The goal is to bridge the gap between conventional document handling and modern digital solutions with enhanced security, ease of access, and scalability.

Keywords: Document Management System, Flutter, Firebase, OCR, Cloud Storage, Secure Access

I. INTRODUCTION

In the digital age, managing documents effectively has become a critical challenge for individuals and organizations alike. Traditional paper-based systems are not only prone to loss, damage, and disorganization, but they also consume valuable time and physical space. As digital transformation accelerates across sectors, there is a growing demand for systems that can provide secure, scalable, and efficient management of personal and official documents.

A Document Management System (DMS) is a software application that enables users to store, organize, retrieve, and manage documents electronically. Such systems help users transition to a paperless workflow, reduce the risk of losing important records, and provide instant access to files when needed. With the growing reliance on digital documents—ranging from academic certificates and identification proofs to bills, contracts, and personal records—having a centralized platform that ensures both accessibility and security is essential.

This project presents a mobile-based Document Management System designed using Flutter for the frontend and Firebase for the backend. It is equipped with features such as user authentication, document upload, Optical Character Recognition (OCR) for scanned text extraction, categorization, and cloud-based storage. The proposed system is not only responsive and user-friendly but also scalable to meet the increasing demands of digital document handling.

By enabling real-time access, categorization, secure storage, and intelligent search features, this DMS offers a smart and reliable solution for managing personal documents. The application aims to simplify the document handling experience, ensure data privacy, and empower users to take full control over their digital information in a fast-paced and increasingly paperless world.

II. LITERATURE REVIEW

1. Traditional vs. Digital DMS (Sharma et al., 2022)

Traditional document management relies on file-based storage systems that lack intelligent organization, making retrieval cumbersome and time-consuming. Sharma et al. discuss how these systems fail to support metadata tagging or indexing, leading to inefficiencies in search and categorization. Documents are stored in unstructured formats without version control, making it difficult to track changes. Digital DMS solutions address these limitations by enabling structured data storage and fast retrieval. The study emphasizes the need for transitioning to smarter, metadata-driven systems for better document accessibility.

2. AI in Document Classification (Kumar, 2023)

Kumar highlights how Natural Language Processing (NLP) enhances the classification of documents based on their content. By using pre-trained AI models, documents can be automatically tagged and sorted into appropriate categories. This minimizes the need for manual intervention, saving time and reducing errors. The study demonstrates how AI improves the accuracy of document retrieval through semantic understanding. It concludes that AI is a valuable tool in building intelligent, self-organizing document systems.

3. Blockchain for Document Integrity (Chandra et al., 2024)

Chandra et al. explore how blockchain technology can ensure document authenticity and integrity through cryptographic hashing. By storing hash values of documents on a blockchain ledger, any unauthorized modification becomes detectable. This tamper-evidence is especially useful in legal, financial, and academic records where trust is critical. The decentralized nature of blockchain further enhances security and eliminates single points of failure. The research proves blockchain to be a reliable method for secure and transparent document verification.

4. Decentralized Storage with IPFS (Patel, 2024)

Patel discusses the application of the InterPlanetary File System (IPFS) for distributed document storage. Unlike traditional cloud systems, IPFS stores data across a decentralized network, making it resistant to failure and censorship. It assigns unique content hashes to files, allowing precise version control and retrieval. The study finds IPFS particularly beneficial in compliance- heavy industries where data integrity and accessibility are crucial. Patel concludes that IPFS can complement existing DMS setups by enhancing redundancy and control.

III. METHODOLOGY

3. Methodology

The development of the Document Management System (DMS) was guided by an iterative, user- centric approach, combining Agile principles with structured software engineering practices. The project aimed to deliver a mobile-friendly, secure, and scalable platform for personal document storage and retrieval using **Flutter** and **Firebase**. The following methodology outlines the stages undertaken during the project lifecycle:

3.1 Requirement Analysis

The process began with a detailed analysis of user needs and document-related challenges. Through surveys, existing system evaluations, and comparative studies, the team identified key pain points—such as unstructured storage, lack of accessibility, and low document security—as critical issues. Requirements were then documented into functional (e.g., upload, scan, categorize, OCR) and non-functional (e.g., performance, security) specifications.

3.2 System Design

After defining the requirements, a modular system design was created. The system follows a **client-server architecture**, where the client is a cross-platform mobile application developed using Flutter, and the server-side logic is handled via Firebase services. The architecture supports seamless document upload, cloud storage, user authentication, and OCR processing using Google ML Kit. The design was represented through **DFD**, **UML**, and **system architecture diagrams** for clarity and maintainability.

3.3 Technology Stack Selection

The chosen technologies include:

- **Flutter and Dart** for building responsive cross-platform mobile interfaces.
- **Firebase Authentication** for user sign-up and login with secure token-based access.
- **Firebase Cloud Firestore and Cloud Storage** for storing document metadata and files respectively.
- **Google ML Kit OCR** for real-time text extraction from scanned images.

These tools were selected for their scalability, real-time capabilities, and developer support.

3.4 Implementation Phases

The implementation was divided into logical modules:

1. **Authentication Module** – handles user registration and secure login.
2. **Upload & Scan Module** – enables file uploads or camera-based scanning.
3. **OCR & Metadata Module** – extracts text using OCR and auto-generates searchable tags.
4. **Categorization & Search Module** – stores documents with user-defined categories and enables intelligent retrieval.
5. **UI/UX Layer** – provides a user-friendly experience with responsive design across devices.

Each module was tested individually and then integrated through Firebase services and APIs to ensure smooth functionality.

3.5 Security and Data Integrity Measures

Security was addressed using:

- Firebase Authentication with encrypted credentials
- Role-based access control
- Secure HTTPS communication
- Metadata indexing to prevent unauthorized access

The app handles document integrity by linking metadata with document hashes and enforcing upload validations.

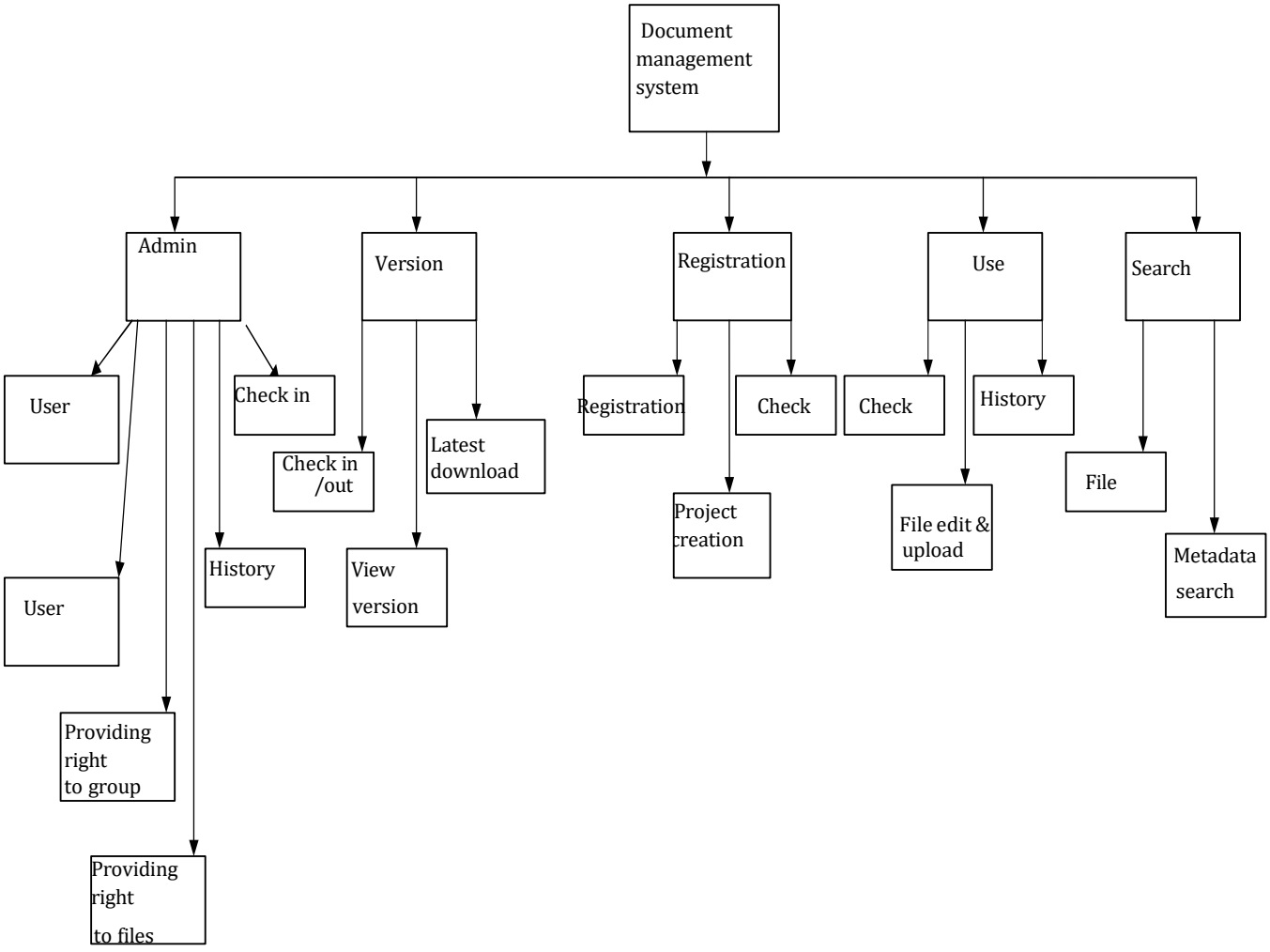
3.6 Testing and Validation

Multiple testing strategies were used:

- **Unit Testing** for isolated functions like login, OCR output
- **Integration Testing** for Firebase interactions
- **Usability Testing** through user trials for UI/UX feedback
- **Performance Testing** under simulated multi-user load conditions

The system demonstrated high reliability and fast response times across Android and iOS environments.

IV. SYSTEM ARCHITECTURE



1. Document Management System (DMS) – Central System

The core platform that handles the full lifecycle of documents: creation, editing, versioning, access control, and archival.

2. Admin Module

The admin plays a supervisory and control role in the DMS. Their responsibilities include:

a. User Management

The admin manages access for multiple users. Assigns roles or permissions.
Adds or removes users from the system.

b. Access Control Providing rights to group:
Admin assigns document access permissions to user groups (e.g., team, department). Helps in scalable permission management.
Providing rights to files:

Admin gives individual users or groups permission to access/edit specific files. Ensures file security and controlled distribution.

c. Document Monitoring Check-in / Check-out:
Ensures that only one user can edit a file at a time. Prevents overwriting by locking the file during edits. **History:**
Tracks user activities and modifications. Useful for auditing and accountability.

3. User Module

a. User Access

Individual users access the system post-registration.

Users may have varying roles: viewer, editor, contributor, etc.

b. Interaction with Admin

Users interact with the admin for permissions and file requests. Users can be added to groups or granted direct access to files.

4. Version Control Module

Maintains consistency and traceability of documents.

a. Versioning

Tracks multiple versions of a document. Helps in rollback and comparison.

b. Check-in / Check-out

As explained under Admin, used to manage document editing sessions. Ensures version safety and file integrity.

c. Latest Download

Users can download the most recent version of a file. Ensures that they are working with the updated content.

d. View Version

Allows users/admins to view or restore previous document versions. Useful for tracking changes and undoing errors.

5. Registration Module

Allows users to formally enter the system.

a. Registration

User submits credentials and details to gain access. Might involve admin approval.

b. Project Creation

Post-registration, users can initiate new document-based projects. A project can include multiple documents and collaborators.

6. Use Module

Covers all active document interactions.

a. Check

Performs system-level or user-level validation before document actions. For example, checks for proper access rights before upload/edit.

b. File Edit & Upload

Users can upload new documents or edit existing ones. Triggers version updates and check-in/check-out functions.

7. Search Module

Handles document retrieval and discoverability.

a. File

Simple file-based search.

Could include browsing by folder/project.

b. Metadata Search

Advanced search using metadata such as:

- Document title
- Author
- Creation/modification date
- Tags, categories, file type, etc.

8. History Tracking

Tracks all significant user and document activities.

a. Via Admin

Admin can view comprehensive history logs.

b. Via Use Module

Users may also view activity logs on documents they have permission to see. Useful for tracking who edited what and when.

V. CONCLUSION

The Document Management System (DMS) developed in this project effectively addresses the growing need for secure, accessible, and intelligent digital document handling. By integrating modern technologies such as Flutter for a cross-platform user interface and Firebase for real-time backend services, the system offers users a reliable and seamless experience in uploading, scanning, organizing, and retrieving their personal documents. The addition of Optical Character Recognition (OCR) enhances the usability of scanned documents by enabling searchable text extraction, improving both accessibility and convenience.

The implementation of robust security features, including authentication and cloud storage, ensures that sensitive documents remain protected. This system empowers users to transition from unorganized paper-based systems to a structured, digital-first approach, contributing significantly to efficiency, data integrity, and peace of mind.

Overall, the project has demonstrated the feasibility of developing a scalable, user-friendly, and secure DMS using contemporary mobile and cloud-based technologies. Future improvements can include advanced AI-based document classification, real-time collaboration features, and support for decentralized data storage.

VI. REFERNCES

1. Malekani, A. W., & Alphonce, A. B. A. (2022). *Assessing Electronic Document Management Systems in Records Management at Sokoine University of Agriculture*. Library Philosophy and Practice.
2. Andriansyah, R. (2020). *Analysis of the Effectiveness of Electronic Document Management Systems*. International Journal of Innovative Science and Research Technology (IJISRT).
3. Kumar, D., Arun, R. S., & Andrews, A. (2021). *Flutter and Firebase-based Application Development*. IEEE Conference on Smart Computing.
4. Gamido, M. V., & Gamido, H. V. (2022). *Analysis of Network-Based Electronic DMS for Organizational Use*. International Journal of Engineering Sciences (IJES).
5. Patel, R. (2024). *Use of IPFS in Decentralized Document Storage for Regulatory Compliance*. Web3 Research Review.
6. Chandra, M., & Gupta, A. (2024). *Blockchain-Based Document Integrity Verification Using Hashing*. Journal of Cryptographic Systems.