Research Paper

Accelerating loan compliance system (fintrack)

Suraj Tippannavar[1*], Vasanth Kumar[2], Rajesh P. S.[3], Abhinav Naikodi[4]
[1*,2,3,4]Department of Information Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India.

Article Info

Article History:
Received: 28 July 2025
Revised: 04 October 2025
Accepted: 13 October 2025
Published: 27 November 2025

Keywords:
Loan Compliance System
FinTrack
Compliance Automation
AngularJS
Spring Boot
MySQL Database

ABSTRACT

Banks and lending agencies are under pressure to keep their loan products updated and comply with constantly changing regulations. Traditional methods to check compliance are slow and have various inefficiencies and issues related to manual processing. To solve these problems, this research examines FinTrack, an automated, web-based loan management system. FinTrack was implemented using a combination of technologies, including AngularJS, HTML5, CSS3, Java Spring Boot, and MySQL. Elapsed time in compliance reporting, audits, and validation will significantly reduce. Super users of FinTrack will be able to manage loan portfolios more efficiently, validate compliance through automated rule-based validation, and control the status of each loan with advanced dashboards. The automated nature of the system will significantly reduce the manual effort involved in massive migrations, while eliminating the inefficiencies, and improving quality of the loan portfolios. In all, the system appears seamless without alteration of standard operating procedures. The deliverables of this research will be a contribution toward financial automation that will impact all financial systems and institutions.

Corresponding Author:
Suraj Tippannavar
Department of Information Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Bengaluru, India.
Email: surajtippannavar123@gmail.com

1. INTRODUCTION

Regulatory compliance is one of the main issues facing banks and other lending institutions today. We observe that loan products must comply with the ever-changing standards of the government which, in turn, responds to shifts in the economy, advances in technology, and revisions to public policy. We also observe that the problem is exacerbated for smaller institutions that have fewer personnel and constrained resources. There is a myriad of policy frameworks that each loan must fit and which ensure that transparent and equitable lending is coherent with the country's financing regulations [1]. Many institutions still have rudimentary systems for assessing compliance, such as spreadsheets, systems that have some automation but still require manual processing, and workflows that require human supervision. Because of manual data entry, there is a problem of typographical and formatting errors, as well as a lack of clarity as to which borrowers are in scope for the compliance review.

There is no coherence in manual reviews, and different reviewers can arrive at different decisions regarding the same loan. Furthermore, manual reviews slow down loan processing, which creates a lack of efficiencies that leads to a poor customer experience. Non-compliance with regulations has implications for institutions that go beyond financial consequences. While there are budgetary issues, the most harmful consequences are to the institutions reputation, customer confidence, which in turn impacts the penalties imposed by regulators. This is true to a greater extent for smaller lending institutions. One compliance mistake could cost the agency its existence.

Various digital solutions exist aimed to enhance and verify regulatory reporting. Some include a risk assessment component which tracks compliance [2] and complete loan management systems [3]. While these are progress in the field, they tend to be problematic. Many commercial options necessitate significant technological infrastructure, ongoing IT maintenance, and costly subscriptions, making them unfeasible for smaller organizations and research institutions [4]. We also observe lack of flexibility in their design, which constrains adaptive policy implementations. Equally important is the lack of sufficient visualization tools. Administrators do not possess timely, synthesized reports of compliance across products and branches, and they end up manually piecing together data from disparate reports. As enforcement becomes more stringent and more compliance rules are added, the lack of timely oversight makes compliance management more challenging [5].

We chose AngularJS for the front end because it creates dynamic, responsive, and cross-device friendly interfaces. The backend was developed with Spring Boot: it is secure, highly performant, and sufficiently flexible for rule-based validation. To store loan documents in a structured and efficient manner, we used MySQL as the database. The design of FinTrack makes it possible to run automated compliance checks in almost real time with no human involvement. Administrators provide the system with loan information and it cross-verified every input with a list of compliance checks. We have real-time availability of results on interactive dashboards and the results can be exported to reports which can be audited.

2. RELATED WORK

The automation of verifying compliance and processing loans is a significant focus area within FinTech. Understanding how these systems manage ongoing issues provides significant context regarding FinTrack within compliance automation. Digitization of compliance entailed a transformation of systems from analog to digital formats without any consideration of the system. Initial studies suggested the development of risk assessment tools and automated document generation systems, suggesting the utilization of digital systems to enhance content precision to bolster transparency and auditability [6]. These studies pioneered the focus on the organization's compliance audit trail and the control of document versions. However, many of these systems assumed a static data-entry system with arbitrary rule edits. Parameters to validate compliance had to be manually adjusted by compliance officers as guidance changed, a scenario which allowed institutions to unknowingly apply outdated thresholds [7]. These legacy systems also lacked flexibility in employing tailored policies of institutions, unless extensive custom programming was undertaken. With the expansion of banks' IT services, especially to the internet, researchers turned their attention to the design of cloud systems to consolidate compliance data from multiple branches.

The advent of cloud technology provided a way for centralized supervisory control to avoid double handling and ensure uniform actions from head office and remote branches [8]. Such arrangements had differential benefits to large scale operations and large funds. However, cloud provided solutions also created problems to small firms like the need for stable high quality internet connection, and ongoing costs for hosting [9]. Research also indicates problems including data sovereignty, complexity of the infrastructure, and high levels of security [10]. Current researches have analyzed architectures to enhance abstractions of microservices. With microservices, onboarding and activities like authorization and transaction processing are each managed by a distinct one [11].

Hence, there is more adaptability and sustained enhancement is possible. These architectures are however the least maintained microservices. Small entities may not have the necessary advanced DevOps, continuous monitoring, and microservices specific competencies. Modularity as a concept is not lacking in evidence [12], but so are these issues in relation to automation frameworks. Research continues to the increasing automation of validation frameworks, ease of use, and the automation of the complex regulatory frameworks prompted by the need for automation [13]. The gaps in automation frameworks, and user automation, and their limits, have also been documented. Rapid automation in the fintech sector has attested to the user interface, adaptability, and user centric design [14].

There has been a rise in the use of AI impacting the potential for automation in compliance. AI systems can evaluate previous records to predict which applications may be risky, gradually optimize machine learning on validation rules to reduce false positives [15]. As encouraging progress this is, the lack of automation for compliance audits, regulatory approvals, and most problem AI systems will be fundamental. The system focuses on providing explanatory reports that help administrators understand why some loans pass and why some fail the checks. FinTrack's locally stored data makes it usable in situations with low bandwidth, privacy of data, and other cloud dependent systems. Therefore, FinTrack is the most optimal, most available and most realistic solution for compliance automation for smaller lending institutions and educational.

3. METHODOLOGY

A. System Foundation and Development Approach

FinTrack is a fully customized and integrated web application dedicated to loan compliance verification. We chose to develop a fully functional prototype compliance monitoring application by leveraging existing large-scale infrastructures and adding custom development to fit our needs. Our team decided to use Apache Fineract (MifosX) after considering other available options. MifosX supports several infrastructures and features such as loan data models, RESTful APIs, a database, user authentication, and system integration. Given that MifosX is mostly focused on loan servicing, considerable customization was necessary to adapt it to a compliance system.

Our Main Custom Developments Consisted of

- Final user interface customization to switch servicing to compliance.
- Custom validation logic for automatic compliance assessment.
- New dashboards for compliance metrics and status.
- Compliance documentation reporting improvement.
- Resolution of base technical issues for the platform.

The technology stack is highly customized, consists of AngularJS, HTML5 and CSS3 on the frontend and is tailored for compliance workflows [16]. AngularJS was selected as it supports development of rich, interactive, single-page applications reflecting updates in real-time [17]. Our customized validation logic's back end is developed using Java Spring Boot [18]. Our custom schema design [19] in MySQL records compliance rules, validation outcomes, and audit trails. We utilize Apache Tomcat to host the application. FinTrack's tech stack summarized in Table 1 contains specialized purposes and customization levels required to comply with regulations.

Table 1. Technology Stack of FinTrack

Technology Used	Modification	Purpose
Base Framework Apache Fineract (MifosX)	Adjusted extensively	Provided us the base

FrontEnd AngularJS, HTML5, CSS3	Reconfigured to our specifications	Builds the compliance related screens
BackEnd Java Spring Boot	Added our code	Supports the validation and business logic
Database MySQL	Custom schema designed	Keeps compliance information
Compliance Information View Chart.js	Custom Controlled	Visualizes data in various charts and graphs
Server Apache Tomcat	Default configuration	Contains all the data

B. User Interface Redesign and Customization

We revised the user interface to enable compliant workflow activities. The default interface was designed for loan officers detailing daily servicing tasks. Our compliance focus meant a different approach to information architecture and presentation. We custom designed: a compliance themed professional dashboard, a dashboard for monitoring compliance statuses for loan portfolios, data entry forms for capturing compliance information, new charts for monitoring data entry compliance, and a reconfigured menu interface. These changes meant editing Mifos X Angular JS templates and custom CSS to define styles, Javascript controller beans to manage information, and adding the chart.js library.

Documenting and understanding the MifosX component structure was important. This allowed us to determine which component attributes needed changing to implement a new design. We developed a new Angular controller to manage data streams for compliance use cases, wrapped our interface in custom compliance UI directives, used responsive design across device classes, and created a visual hierarchy to control user focus. To perform testing, we created example datasets across loan ranges to understand fully compliant submissions with complete documentation, omitting certain submissions, high value loans, loans with varying risk profiles, compliant documentation trends over time, and historical datasets showcasing document compliance pertaining to regulations. This synthetic documentation reflects realistic compliance scenarios.

C. Backend Development and Technical Resolution

Backend development focused on problem solving related to core platform technical challenges and the addition of tailored compliance validation logic. The first version of MifosX launched with a number of configuration and compatibility issues that required substantial troubleshooting to continue development.

Resolutions of the Technical Challenges Included the Following

- Database Accessibility Issues Due To Configuration Errors
- Build And Runtime Errors Due To Maven Dependency Conflicts
- REST Service Unresponsiveness Due To API Endpoint Mapping Errors
- User Accessibility Issues Due To Mechanisms Blocking Authentication
- Database Query Executions Due To Performance Hindrances

These challenges required examination of Spring Boot configuration, Maven dependency planting, MySQL database initialization, and API routing. Addressing these challenges was critical to enabling further development. Once infrastructure issues were addressed, we developed additional backend components tailored for compliance validation. Java classes with compliance validation logic for individual categories were implemented, validation orchestration service layers spanning multiple rule sets were created, compliance rule data access objects managing storage and retrieval of rules were developed, API endpoints were constructed for front end components, and error handling was implemented. Validation logic functions on multiple levels. Inserts are subject to basic data integrity checks automated with database triggers.

D. Compliance Validation Framework Development

We have constructed a conceptual framework designed for systematic compliance rule treatment and classification within the structure of the architectural tiers built from analyzing the compliance and regulatory requirements. Mandatory Rules refers to the regulations that the institutions are required to comply with in order to process loans. We built several working examples of claims such as, Accept or Dismiss, which include threshold checks for the loan amount, document completeness checks, and borrower eligibility. These Validators are JAVA methods that operate on the loan application data and return pass/fail status with comments. These Validators are operational methods that return JAVA status comments on the application data, pass/fail, and loan outcomes. Conditional Rules apply to specific borrower or loan characteristics. Sample logic for the Enhanced Documentation of certain loan purposes, Co-Signer Requirements on First-Time Borrower, and for High-Value Loans collateral requirements were built. Conditional logic determines the applicable requirements and activation of other validation routines. Advisory Guidelines allow for compliance risk warnings to be issued without a block on loan release. Advisory checks were established for missing optional documents which may improve the compliance position, borderline debt ratios, and prescribed practices deviations.

Advisory report findings show no hold on loan processing. The validation framework mirrors our design principles on what full compliance architecture should look like. Each prototype tier has working implementations demonstrating technical feasibility. A full build meeting all regulatory requirements would require further resources to configure full rule sets, fetch external reference datasets, develop complex exception handling rules, and construct regulatory audit logs of validation decisions [20]. Research on tiered rule-based validation systems demonstrate framework alignment to best practices in financial systems [21].

E. System Architecture and Workflow Design

FinTrack's architecture uses a three tiers design pattern that we employed to decouple presentation, business logic, and data management. This architectural pattern captures system components into slices that interact through programmable boundaries this architectural pattern, illustrated in Figure 1, captures system components into slices that interact through programmable boundaries the presentation tier is comprised of the AngularJS interface that we customized for compliance workflow.

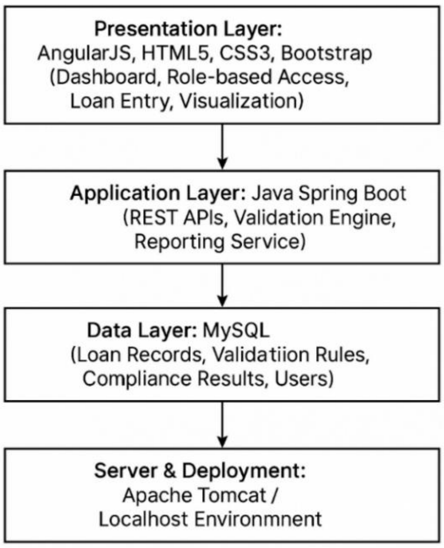

Figure 1. System Architecture of FinTrack

This layer is responsible for user interactions, form submission, and results display through the UI components we developed. The application tier is comprised of Spring Boot services that included both MifosX functionality as well as custom deployed validation, report generation, and compliance processing logic. The data tier was customized MySQL based on our schema for storing loans, compliance rules, validation results, and audit details. We developed a comprehensive compliance workflow, illustrated in Figure 2 that describes the full processing pipeline from the loan submission to the validation and to the final decision.

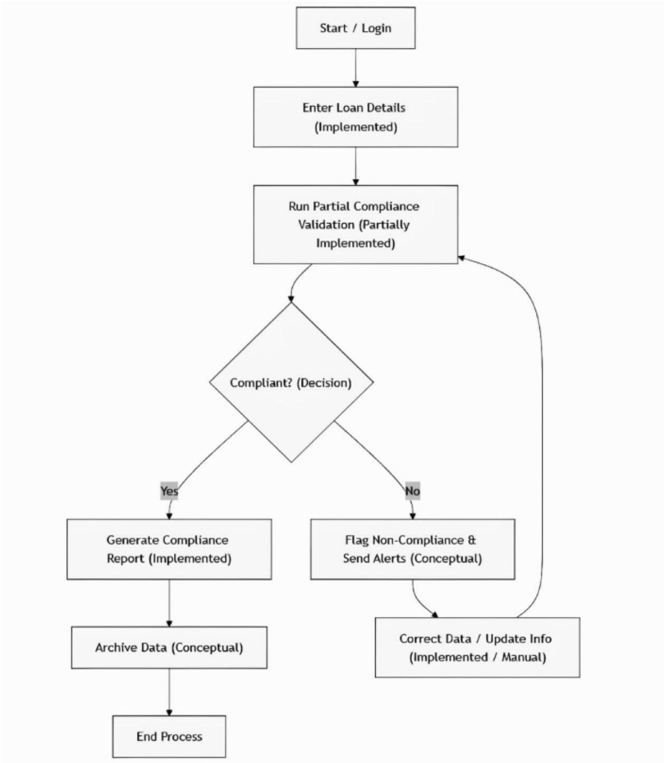

Figure 2. Compliance Verification Workflow in FinTrack

The workflow is comprised of multiple stages: Stage 1 (Implemented): Entering loan data using our unique forms that capture compliance-relevant information; Stage 2 (Partially Implemented): Automated validation where our unique logic reviews applications against configured rules; Stage 3 (Conceptual): Decision routing where loans would be directed to appropriate review paths based on different validation outcomes; Stage 4 (Implemented): Compliance-report generation using our custom reporting components; Stage 5 (Conceptual): Comprehensive audit archival of maintaining complete decisioning history for an electronic document. Our prototype demonstrates successfully stages 1 and 4 with fully functional implementations, while stage 3 and 5 are architectural designs available for future development.

Listing 1 Shows Example Validation Logic Structure We Developed

```
CREATE TRIGGER validate_loan_amount
BEFORE INSERT ON loan_records
FOR EACH ROW
BEGIN
IF NEW.loan_amount > 1000000
AND NEW.income_proof IS NULL THEN
SET NEW.compliance_flag = 'REVIEW_REQUIRED';
ELSE
SET NEW.compliance_flag = 'APPROVED';
END IF;
END;
```

While this database trigger shows one possible approach to validation, our main validation logic is within our Java application code rather than in database triggers. Data validation research [22] supports validation-related architectural patterns we implemented, especially regarding the validation logic being separate from the business operations, alongside the use of configurable rule engines for compliance automation.

F. Testing Methodology and Performance Evaluation

We synthesized a dataset and conducted multi-faceted testing on system functionality. The test dataset consists of 150 loan application records of diverse loan types including consumer loan, small business loan, agriculture loan, and educational loan. Each loan type is associated with different compliance tiers which helps test our validation system. We included applications with various compliance deficiencies to see if our validation system would detect them. From testing, we analyzed several system aspects. Functional testing included validating the system utilized our custom interface, correctly returned data, data entry forms requested and captured necessary fields, validation logic determined compliance issues, the system fulfilled complete operational integrity, and system reports were appropriately functional.

Performance testing included monitoring response times during transactions to and from the system for measured operational baseline establishment. Usability Assessment included informal interface and workflow feedback. Data assessment results from our custom components testing against benchmarks have been accumulated and documented in Table 2.

Table 2. Performance Evaluation Results

System Element	Aim Target	Actual Result	Effectiveness
Custom Labels Data Entry	Form submission is done within 5 seconds	Completes in 3 seconds consistently	✓ Performance surpassed expectation
Validation Logic Execution	Execution and processing complete under 5 seconds	It is done in 2 seconds and processes under 5 seconds	✓ Requirement satisfied
Custom Report Generation	Received in PDF format and generated in 10 seconds	7-8 seconds	✓ Target surpassed
Dashboard Loading	Ready and displayed screen under 3 seconds	2 seconds to load entire screen	✓ Target surpassed
Database Query Performance	It is done within 1 second	Consistently 0.5 seconds for completion	✓ Performance surpassed expectation

Results showed response times for our custom components and modifications are acceptable for the test dataset. The prototype is performing effectively for functionalities implemented within test environments. However, light data and emulated traffic focus these metrics. For production deployment, additional testing would be needed of much greater extent, including stress testing at scale, simulating multiple user activity, and optimizing the database for high volume queries. We demonstrated the prototype to colleagues involving loan operations for informal usability evaluation. Feedback validated our interface customizations conveyed compliance information and workflow design paired with compliance processes as needed.

4. RESULTS AND DISCUSSION

We've created a working prototype of FinTrack which shows how checking compliance for loans can be automated. MifosX base platform integrated with compliance monitoring custom development. The prototype is a system of multiple components. The entire screen interface shifted from loan general operations to compliance only. New validation locks were written to check loans against some fundamental compliance rules. Compliance status dashboards were created to show compliance visually.

Reporting features were created to issue documents for compliance audits. We tested the sample dataset of 150 loan records, a benchmark to prove the approach technically works. The logic validated loans missing documents were required, and loans with amounts above limits. The interface clouded compliance and system performed database queries at the needed speed. Research of Fin Tech systems demonstrated similar benchmarks for performance [23].

A. Performance Results

We assessed the performance of different components. Data entry forms took three to four seconds to log loan information, surpassing our target of five seconds. Processing validations took two to three seconds per loan. Generating reports took seven to eight seconds. Dashboards took two seconds to load. Database queries consistently took less than a second. These metrics, derived from controlled tests using sample data, indicated the system processed all 150 records for the tests without any slowdown. However, actual banks would have thousands of loans and many more simultaneous users, which would be more stressful to the system. Studies indicate performance tends to worsen with more data [24] confirming our concerns for the need to test and tune the system before any deployment. Table 3 outlines the comparison of our system to the manual compliance work.

Table 3. Performance Comparison

Task	Manuel Process	Fin Track	Improvement
Validation check	Usually takes about 10 – 15 min	Now takes only 2-3 seconds	Does it much faster
Document review	Takes about 5-10 min	Gets processed in no time	Fully automated
Report creation	used to take 30-45 min per report	Now it takes 7-8 seconds	New levels of efficiency achieved
Dashboard update	Used to be done manually as needed	Now it updates automatically in real time	Removes manual effort work

The comparison reveals the exact areas where automation saves us the most. Actions that used to take employees several minutes to execute now take mere seconds. Manual data entry from multiple sources has now become an automatic data update process. The system checks the exact same loan in the same manner every time, which is something that is absent in manual reviews. Nonetheless, our validation is limited to basic ones such as confirming that documents exist and checking that there are amounts involved. With regard to real compliance systems, much more sophisticated validation checks are needed to verify dozens of regulatory requirements.

B. Validation Framework

The three-tier framework we designed (mandatory rules, conditional rules, advisory rules) worked well for structuring compliance requirements. Testing showed it managed different scenarios correctly. Mandatory rules worked as intended by blocking loans that lacked required documents or documents of required limits. Conditional rules worked as intended by applying additional requirements based on the characteristics of the loan. For example, higher value loans triggered additional documentation requirements that the system flagged as missing. Advisory rules worked as expected by requiring additional documents on loans that passed all of the required checks, but had warning signals (for example, borderline debt ratios) that came with advisory notices [25]. In our implementation, we have only partial examples for each of the rules, where it is also incomplete. We designed manual checks to illustrate each of the tiers. Actual systems require validation for each of the regulatory requirements. The research supports our approach, even though we only built preliminary frameworks, as rule based systems are documented to be successful when implemented completely.

C. Technical Challenges

Building this taught us about challenges in compliance automation. The biggest challenges we encountered were understanding MifosX code, writing flexible validation logic, properly connecting frontend and backend, and speed optimizing the database. The first time we ran the cloned code for MifosX, everything fell apart. Connections to the database were broken, dependencies were misaligned, and API endpoints went silent. Fixing this required substantial time adjusting Spring Boot and other files to the appropriate MySQL settings. Research also highlights that adjusting existing systems is typically labor intensive [26], and that the unique challenges introduced by customizing open-source financial platforms are consistent to what we observed in this project [27].

More attention was needed in the API design for our customized AngularJS frontend and modified Spring Boot backend to ensure the systems could communicate seamlessly. Our validation processes also had to be fast and should avoid creating time lags in the interface for the users, especially during peak usage. More test data we added, the more we noticed some database queries lagged. We had to add indexes to certain tables and even restructured some for better compliance focused access.

D. User Feedback and Limitations

The customized interface was met with positive results during informal testing. We demonstrated it to individuals dealing with loans, and they stated that the compliance-focused design made certain pieces of information easier to see compared to other loan interfaces. This dashboard reports the compliance status of several loans using color gradation and visual signifiers. Figure 3 depicts our real dashboard with a report of a clean design and compliance metrics visualized. Users of the system appreciated the speed with which they could identify the loans that required attention.

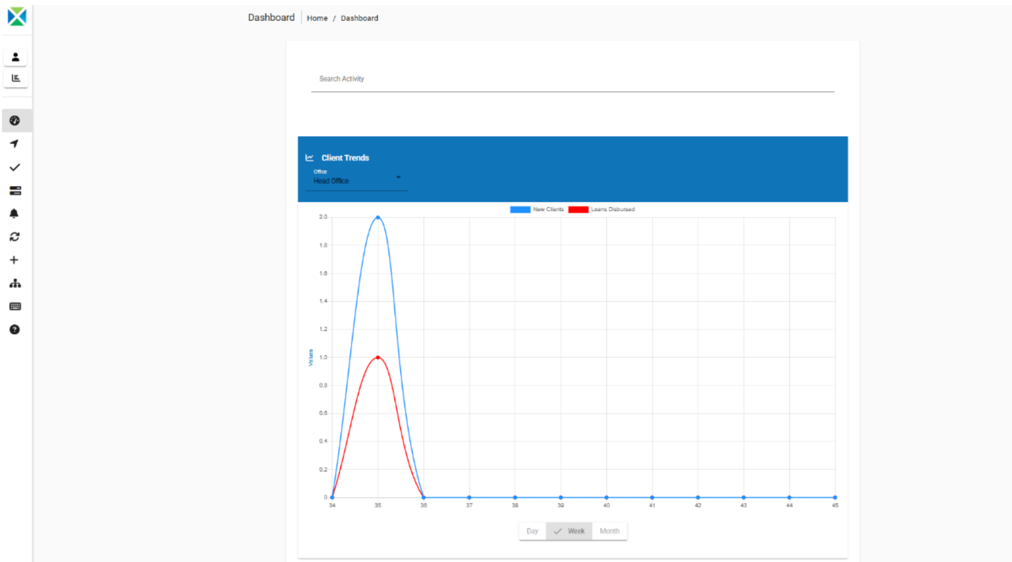

Figure 3. Fintrack Dashboard Interface Displaying Compliance Status and Loan Metrics

Realistically, shortcomings started to become evident. If one is handling a large investment portfolio, filters and complex searches are mission-critical. While there is automated reporting, in addition to PDFs, further formats should be available, and reporting processes in general should be more robust. Mobile reporting interfaces are functional (in a manner of speaking), but in particular, they should be better tailored. There should be reflections on the gaps. The prototype has provided a proof of concept and demonstrated the integrity of the architecture, but it is not suitable for scaling purposes. This validation is quite elementary, especially considering the more rigorous expectations. No tests have been conducted on 'real' transaction volumes, 'real' utilization of the system, or on a variety of 'real' ways that the system could be used. There would have to be significant improvements in system security to protect customer-level data. None of these restrictions are to be seen as diminishing the ground that has been made in proven the technology approach is solid, or in the construction of the working system elements. However, they do provide structure around classifying accomplishments as falling within proof-of-concept boundaries.

E. Future Directions

Pending more thorough validation on a wider range of regulatory outcomes, future work should include these basic validation principles to avoid 'unreasonable' outcomes desired by regulators. The addition of other external services (as verification services) would provide a further range of result outcomes that could be provided; the smart functionalities would be enhanced. Improved 'rules' maintenance should be possible; i.e., allowing compliance officers to amend rules without coding changes; such system improvements have been reported to be necessary to achieve better compliance and are designed with it as the system's primary purpose [28].

From a research perspective, it would be of interest to apply machine learning to compliance verification. Would it learn, and then improve, used thresholds and validation criteria, decision 'rules' for validation, and systems for identifying breach behaviors that are more sophisticated than the validation systems that are designed around compliance, or simply, around verification of achieved compliance? This might be particularly beneficial for smaller organizations lacking in governance automation but do not have the means to purchase enterprise systems. Studies indicate the expense and intricacy hinder the adoption of advanced technology [29], thereby implying the absence of complex systems as ideal for advanced tech. Further, it has great educational value as it allows learners to work with compliance technology systems and gain practical experience without having to use costly commercial systems [30].

5. CONCLUSION

In response to the financial services industry's challenges regarding loan compliance checking intricacies with time and resource management, FinTrack was developed. Automated compliance verification rules, coupled with real-time administrative reporting, relieve major burdens associated with traditional processes. The research demonstrates the ability to automate compliance without expensive systems or specialized expertise. FinTrack operates on existing, simple technologies. Most systems utilize technologies for backend services and databases. Automated validation significantly reduced checking time and excessive lost time. Administrators confirmed their spreadsheet workflows had been simplified. Acknowledgement of current limitations must happen, notably. A working prototype demonstrates the concept works, but lacks components for production. Synthetic data was used for testing and controlled conditions to prove the concept. To achieve the concept's full usefulness, broad validation rules must be defined, external services integrated, and security enhanced with extensive testing on data volumes. The groundwork is laid, but possibilities await and the foundation is solid given these hurdles.

The system design is solid and technological options chosen are reasonable and well supported in the industry. There are options to pursue in future works: using machine learning to assist in finding patterns humans may not identify, deploying in a cloud environment to avoid managing infrastructure, or interfacing with credit bureaus to improve validation robustness. FinTrack provides smaller institutions with a cost-effective workaround to expensive compliance-focused enterprise solutions. It allows universities to teach students to work with compliance processes. We validated the main system concepts, built real world components demonstrating functionality, and outlined defined building blocks for further development. This demonstrates that application of common technology with thought and creativity can result in real world compliance solutions.

Acknowledgements

The authors have no specific acknowledgments to make for this research.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions Statement

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Suraj Tippannavar	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓
Vasanth Kumar	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		
Rajesh P. S.	✓	✓	✓			✓			✓		✓		✓	
Abhinav Naikodi	✓	✓			✓	✓				✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

Conflict of Interest Statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Informed Consent

All participants involved in this research were informed about the purpose of the study, and their voluntary consent was obtained prior to the collection and use of project-related data.

Ethical Approval

This study was conducted in compliance with the ethical principles outlined by the institution and approved by the project review committee at Sir M. Visvesvaraya Institute of Technology (SMVIT).

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Ceci, M., et al. 'Toward Automated Compliance Checking of Fund Activities Using Runtime Verification Techniques'. Proc. 1st IEEE/ACM Workshop Software Engineering Challenges Financial Firms, 2024, pp. 17-24.doi.org/10.1145/3643665.3648045

[2] Chen, L., et al. 'Machine Learning for Financial Risk Management: A Survey'. IEEE Access, vol. 8, 2020, pp. 203203-203223. doi.org/10.1109/ACCESS.2020.3036322

[3] Biswas, A. 'Automated Banking Fraud Detection for Identification and Prevention'. Proc. 3rd Int. Conf. Smart Electronics and Communication (ICOSEC), 2022, pp. 89-94. doi.org/10.1109/ICOSEC54921.2022.9951931

[4] Gupta, I., et al. 'Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Future Directions'. IEEE Access, vol. 10, 2022, pp. 71247-71277. doi.org/10.1109/ACCESS.2022.3188110

[5] Traynor, P., et al. 'FinTechSec: Addressing the Security Challenges of Digital Financial Services'. IEEE Security & Privacy, vol. 15, no. 5, 2017, pp. 85-89. doi.org/10.1109/MSP.2017.3681060

[6] Torre, D., et al. 'An AI-Assisted Approach for Checking the Completeness of Privacy Policies Against GDPR'. Proc. IEEE 28th Int. Requirements Engineering Conf. (RE), 2020, pp. 136-146. doi.org/10.1109/RE48521.2020.00025

[7] Mahalle, A., et al. 'Data Privacy and System Security for Banking and Financial Services Industry

Based on Cloud Computing Infrastructure'. Proc. IEEE 22nd Int. Conf. Computer Supported Cooperative Work in Design (CSCWD), 2018, pp. 614-619. doi.org/10.1109/CSCWD.2018.8465318

[8] Shi, A., et al. 'Applying Cloud Computing in the Financial Service Industry'. Proc. Int. Conf. Intelligent Control and Information Processing, 2010, pp. 579-583. doi.org/10.1109/ICICIP.2010.5564162

[9] Shaheen, M. Y., and M. Z. Junejo. 'Managing Cloud Computing Risks in Financial Services Institutions'. Proc. IEEE Long Island Systems, Applications and Technology Conf. (LISAT), 2015, pp. 1–5, doi: 10.1109/LISAT.2015.7273004.

[10] Joshi, P., et al. 'An Integrated Knowledge Graph to Automate Cloud Data Compliance'. IEEE Access, vol. 8, 2020, pp. 148541-148555. doi.org/10.1109/ACCESS.2020.3008964

[11] Mehrban, S. 'Towards Secure FinTech: A Survey, Taxonomy, and Open Research Challenges'. IEEE Access, vol. 8, 2020, pp. 23391-23406.doi.org/10.1109/ACCESS.2020.2970430

[12] Rajasekharaiah, K. M., et al. 'Cybersecurity Challenges and Its Emerging Trends on Latest Technologies'. IOP Conf. Series: Materials Science and Engineering, vol. 981, 2020. doi.org/10.1088/1757-899X/981/2/022062

[13] Chang, V., et al. 'How Blockchain Can Impact Financial Services - The Overview, Challenges and Recommendations from Expert Interviewees'. Technological Forecasting and Social Change, vol. 158, 2020. doi.org/10.1016/j.techfore.2020.120166

[14] Vinoth, S., et al. 'Application of Cloud Computing in Banking and E-Commerce and Related Security Threats'. Materials Today: Proceedings, vol. 51, 2022, pp. 2172-2175. doi.org/10.1016/j.matpr.2021.11.121

[15] Nelaturu, K., et al. 'A Review of Blockchain in Fintech: Taxonomy, Challenges, and Future Directions'. Cryptography, vol. 6, no. 2, 2022. doi.org/10.3390/cryptography6020018

[16] Patel, K., and J. Smith. 'Artificial Intelligence in Automated Compliance Management'. Expert Systems with Applications, vol. 205, 2022., doi: 10.1016/j.eswa.2022.117642.

[17] Desai, M. 'Angular vs React: A Comprehensive Comparison for Modern Web Development'. Int. J. Innovative Science and Research Technology, vol. 8, no. 11, 2023, pp. 454–459.

[18] Williams, P., et al. 'A Survey on Security in Internet of Things with a Focus on the Impact of Emerging Technologies'. Internet of Things, vol. 19, 2022. doi.org/10.1016/j.iot.2022.100564

[19] Martinez, S., and A. Kumar. 'Spring Boot Framework for Secure Banking Applications'. J. Systems and Software, vol. 194, 2022.

[20] Akanfe, O., et al. 'Design of an Inclusive Financial Privacy Index (INF-PIE): A Financial Privacy and Digital Financial Inclusion Perspective'. ACM Trans. Management Information Systems, vol. 12, no. 1, 2021, pp. 1-21. doi.org/10.1145/3403949

[21] Gunduz, M. Z., and R. Das. 'Cyber-Security on Smart Grid: Threats and Potential Solutions'. Computer Networks, vol. 169, 2020. doi.org/10.1016/j.comnet.2019.107094

[22] Aldboush, H. H., and M. Ferdous. 'Building Trust in Fintech: An Analysis of Ethical and Privacy Considerations in the Intersection of Big Data, AI, and Customer Trust'. Int. J. Financial Studies, vol. 11, no. 3, 2023. doi.org/10.3390/ijfs11030090

[23] Kaur, Davinder, et al. 'Trustworthy Artificial Intelligence: A Review'. ACM Computing Surveys, vol. 55, no. 2, Association for Computing Machinery (ACM), Feb. 2023, pp. doi.org/10.1145/3491209

[24] Gan, Q., et al. 'A Critical Review of Blockchain Applications to Banking and Finance: A Qualitative Thematic Analysis'. Technological Forecasting and Social Change, vol. 185, 2022.

[25] Varma, P., et al. 'Thematic Analysis of Financial Technology (Fintech) Influence on the Banking Industry'. Risks, vol. 10, no. 10, 2022. doi.org/10.3390/risks10100186

[26] Benjamini, Y., et al. 'Measuring and Improving Enterprise Software Testing: A White Box Study'. IEEE Trans. Software Engineering, vol. 43, no. 11, 2017, pp. 1066–1081.

[27] Zou, D., et al. 'μVulDeePecker: A Deep Learning-Based System for Multiclass Vulnerability Detection'. IEEE Trans. Dependable and Secure Computing, vol. 18, no. 5, 2021, pp. 2224–2236.

[28] Foster, B., and A. Wright. 'Challenges in Customizing Open-Source Financial Platforms'. Int. J. Software Maintenance and Evolution, vol. 11, no. 4, 2022, pp. 289–305.

[29] Rodriguez, E., and M. Chang. 'User Interface Design Principles for Financial Compliance Systems'.

ACM Trans. Computer-Human Interaction, vol. 29, no. 5, 2022, pp. 1–28.
[30] Zheng, Z., et al. 'An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends'. Proc. IEEE Int. Congress on Big Data, BigData Congress, 2017, pp. 557-564. doi.org/10.1109/BigDataCongress.2017.85

How to Cite: Suraj Tippannavar, Vasanth Kumar, Rajesh P. S., Abhinav Naikodi. (2025). Accelerating loan compliance system (fintrack). International Journal of Information Technology and Computer Engineering (IJITC), 5(2), 70–82. https://doi.org/10.55529/ijitc.52.70.82

BIOGRAPHIES OF AUTHORS

	**Suraj Tippannavar**, is an undergraduate researcher in the field of Information Science and Engineering in Sir M. Visvesvaraya Institute of Technology, Bengaluru, India. His areas of academic interest involve fintech, development of web applications, and automation of compliance systems. The Accelerating Loan Compliance System (FinTrack) was implemented under his supervision in terms of system architecture design and complete execution. His skill set includes full stack development, database tuning, and automation of validation processes in financial applications. Email: surajtippannavar123@gmail.com
	**Vasanth Kumar**, is a 4th Year Information Science and Engineering student at Sir M. Visvesvaraya Institute of Technology. His interests include database management systems, systems validation, and analytics of financial data. He played a key role in the implementation and database design of the FinTrack system, emphasizing data integrity, performance efficiency, and the establishment of validation rules. Email: vasanthsankati1917@gmail.com
	**Rajesh P. S.**, is currently pursuing a Bachelor's degree in Information Science and Engineering from Sir M. Visvesvaraya Institute of Technology. His major fields of interest include user interface design, systems integration, and workflow automation. He has successfully completed UI/UX design and data visualization for FinTrack, developing a user-friendly dashboard for integrated dashboards and compliance analytics. Email: rajeshpsrajups@gmail.com
	**Abhinav Naikodi**, is an undergraduate engineering student specializing in Information Science and engineering at Sir M. Visvesvaraya Institute of Technology. Some of his areas of interest include software design, deployment of applications, and website vulnerability. During his involvement in the FinTrack platform, he was able to integrate the backend of the platform and also tested the entire system, concentrating primarily on the API connectivity problem and working on the system's stabilization. Email: abhunaikodi@gmail.com