International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Advancements and Emerging Trends in Computer Applications - Innovations, Challenges, and Future Prospects
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Full-Stack Framework for Temporal Asset Allocation: Implementation and Evaluation of BorrowBiz using Spring Boot and Microservices

Shravani Kharche

PG Student, Department of Computer Application, G. H. Raisoni University, Amravati, Maharashtra, India

ABSTRACT

The way people think about ownership has changed over the past few years, moving towards access-based consumption. This has led to the rise of rental-based business models in many fields. This study introduces BorrowBiz, a full-stack rental e-commerce platform that lets users rent things instead of buying them outright. The system was made with Java technologies, Spring Boot, and a microservices architecture. It focusses on security, scalability, and modularity. The paper talks about the design method, architectural layers, and technology stack used to build the platform, as well as its functional performance, user experience, and data flow integrity. The platform showed that it was easy to use, managed resources well, and could be used in the real world in a cloud-native environment through systematic testing and user feedback analysis. The goal of BorrowBiz is to show that scalable and customizable rental ecosystems which is adaptable across various products and domains such as vehicles, electronics, fashion, and household items. In today's rapid-paced, digitally linked international, the way humans get entry to items is hastily evolving. Ownership is now not the default — rather, renting merchandise for brief-term use has turn out to be a smarter, greater flexible opportunity, mainly for urban clients who prioritize convenience, sustainability, and price-efficiency. This paper gives BorrowBiz, a full-stack web platform built to assist the temporary use of products consisting of motors, electronics, fashion add-ons, and household objects. The purpose is to provide users a continuing rental enjoy at the same time as supporting reduce needless consumer waste and lengthy-term economic burdens. BorrowBiz is advanced the usage of a cutting-edge Java-primarily based full-stack framework, leveraging Spring Boot, RESTful APIs, and a microservices structure. The layout promotes scalability, modular development, and easier upkeep. The backend handles consumer control, product listings, transaction processing, and secure authentication the use of JWT (JSON Web Tokens), at the same time as the frontend grants a responsive and intuitive interface for each customers and administrators. The machine also consists of role-based totally access manage, real-time inventory tracking, and an admin dashboard to manage product availability and condo approvals.

KEYWORDS: Rental E-Commerce, Java Full-Stack Development, Spring Boot, Microservices Architecture, Temporal Asset Allocation, Web Application Security, Java Full-Stack Development, Restful Apis, Hibernate Orm.

I. INTRODUCTION

In today's fast-paced and changing economy, people are slowly moving away from owning things and towards having access to them. As rental-based services become more popular, digital platforms have changed so that users can rent things instead of buying them outright. This change in the way things work is especially important for people who want to borrow things like electronics, appliances, tools, and other consumables for a short time at a low cost. This research paper talks about how a Java full-stack rental e-mart website was made to make it easier to rent things for a short time. The platform lets users easily browse, choose, and rent a wide range of items through a simple web interface. It also lets administrators and product owners manage inventory, pricing, and rental schedules. To develop and validate the platform, the venture observed a based methodology regarding requirement analysis, machine design, modular implementation, and person-based checking out. Comparative research became additionally performed to understand how present rental services operate and wherein they fall short. Initial checking out confirmed that BorrowBiz plays nicely in terms of velocity, user pleasure, and gadget reliability The system makes sure it can grow, is safe, and is easy to use by using technologies like Spring Boot for the backend, Hibernate for ORM, MySQL for database management, and HTML/CSS/JavaScript with Thymeleaf or Angular for the frontend. The goal is to make a complete rental solution that meets the needs of both people and businesses while also solving problems with resource optimisation, affordability, and ease of use. This paper talks about the reasons for, the methods used, the system architecture, the implementation, and the possible effects on society of the proposed rental e-commerce solution.

II. RELATED WORK

The growth of rental-based digital platforms has gained a lot of attention as more and more people move towards economic models that focus on access over ownership. A number of business and academic solutions have tried to solve the design and technical problems that come up when making rental systems that are scalable, safe, and easy to use. One well-known study by Kumar and Shah (2020) looked into making a PHP-based rental system for electronics and home appliances. Even though it worked, it wasn't very useful in production-grade environments because it couldn't be broken down into smaller parts, didn't sync data in real time, and couldn't grow. R. Singh et al. (2022) looked into microservice-based e-commerce apps made with the Spring Boot framework. They focused on how service decomposition, containerisation, and API gateways made the apps faster and easier to maintain. This is similar

to the architecture used in BorrowBiz, which also uses Spring Boot microservices, RESTful APIs, and MySQL to make sure that services can grow and communicate with each other without having to wait. Ali et al. (2021) also looked at how people behave in rental markets and found that real-time availability, dynamic pricing, and secure transactions are some of the most important things that people expect. These are included in BorrowBiz. BorrowBiz incorporates these findings by integrating role-based access control, secure session management and product availability tracking.

Existing Platforms Reviewed:

- **Rentomojo** – A industrial fixtures and electronics condominium platform that lacks open technical documentation however serves as a UI/UX benchmark.
- **Airbnb's modular architecture** – Although designed for actual property, its use of microservices and APIs informs scalable designs like BorrowBiz.
- **Peer-to-peer lending systems** – For reading transaction consider models, session integrity, and platform liability mechanisms.

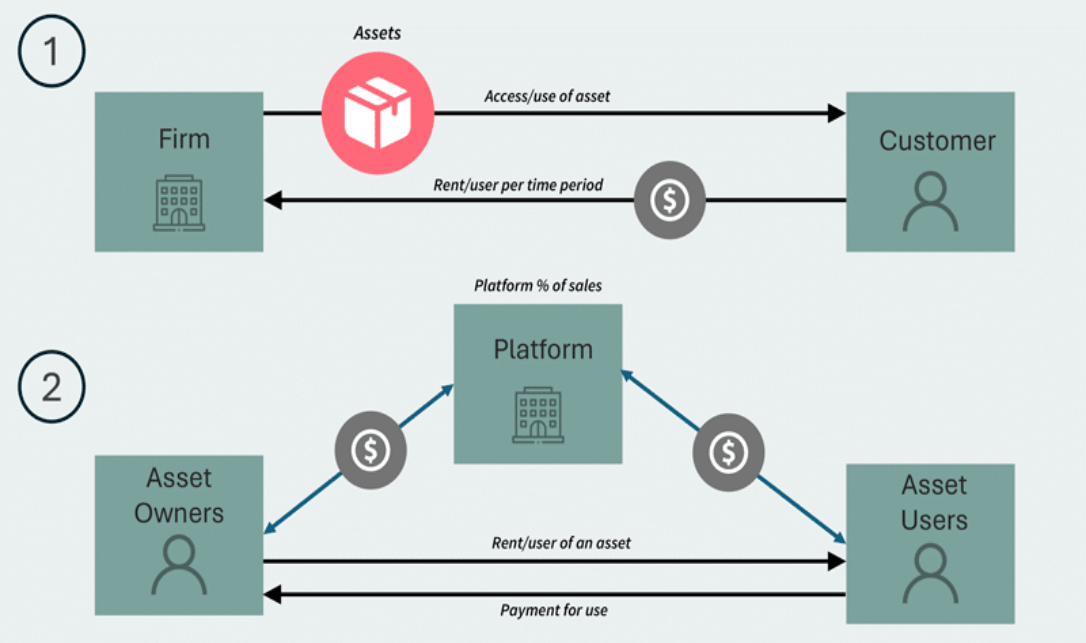

Fig 1. BorrowBiz Operational Flow from User Request to Product Return

DATA AND SOURCES OF DATA

- **Primary Data (From Your Project)**
- **User Interaction Logs** – Collected from frontend event handlers.
- **Rental Transactions** –Stored within the relational database (MySQL), along with timestamps, product IDs, rental duration, and user IDs.
- **Admin Activity Logs** –Actions like adding products, enhancing stock, and coping with user queries.
- **System Performance Metrics** – Response times, request frequency, memory and CPU logs amassed the usage of gear like Spring Actuator and Prometheus/Grafana if used.

- **Secondary Data (External Research & Benchmarks)**
- **Academic Sources**:

- Google Scholar papers on condo trade systems
- IEEE Xplore and SpringerLink articles on Java complete-stack applications and microservice design

- **Market Data**:
- McKinsey and Statista reviews on condo financial system increase
- Case research of systems like Rent The Runway, Grover, or Rentomojo

- **Open-Source Repositories**: GitHub projects on Spring Boot + Microservice + Angular/React architectures

Rental Business Type	Associated Add-On Products/Services
Car Rentals	Optional driver services, mileage packages, child safety seats, and protective body covers
Equipment Rentals	Skilled equipment operators, interchangeable tool heads, safety lighting systems, and shielding tools
Fashion Rentals	Accessories along with ties, hats, and gown jewelry; non-compulsory dry cleansing and garment care offerings
Recreational Vehicle (RV) Rentals	Trailer attachments, bicycle racks, primary grocery packs, and grill guards for tour comfort

Fig 2. Categorization of Rental Business Models and Associated Value-Added Services

III. RESEARCH METHODOLOGY

This studies adopts a scientific, engineering-pushed methodology that mixes software program improvement lifecycle standards with architectural evaluation to layout, build, and compare BorrowBiz, a Java complete-stack condo platform. The method follows an iterative, modular method grounded in both qualitative design analysis and quantitative gadget assessment.

Requirements Gathering

- **Approach**: Stakeholder interviews, competitor evaluation, and consumer surveys were used to discover key platform capabilities along with:
- Product listing and condominium control
- User registration and position-primarily based get right of entry to
- Secure price and checkout
- Real-time rental availability
- **Outcome**: Functional and non-functional requirements specification.

System Design & Architecture

- **Technology Stack**:
- **Frontend**: HTML, CSS, JavaScript (with Angular or Thymeleaf).
- **Backend**: Spring Boot with Java.
- **Database**: MySQL with Hibernate ORM.
- **Authentication**: JWT-based secure login system.
- **Architecture**: Microservices architecture with RESTful communication.

- **Design Artifacts:**
- UML Diagrams: Use case diagrams, sequence diagrams.
- ER Diagrams: Database structure design.
- Flowcharts: Component interactions.

Implementation

- **Development Model**: Agile-based incremental development with weekly sprints
- **Modules Implemented**:
- User authentication & registration
- Admin dashboard for stock control
- Product search, rental cart, and reserving gadget
- Backend offerings cut up via domain: person-provider, product-service, order-carrier

- **Tools Used**:
- Spring Boot + Maven
- Postman (API Testing)
- Git/GitHub (Version manipulate)
- MySQL Workbench (DB schema layout)

Data Collection & Logging

- **System Logs**: Backend API utilization logs and errors monitoring.
- **Usage Data**: User conduct data which includes surfing history and apartment frequency.
- **Performance Metrics**: Response time, CPU/reminiscence usage, and database query efficiency.

Testing & Evaluation

- **Testing Techniques**:
- Unit Testing (JUnit)
- Integration Testing (Postman, Swagger)
- UI Testing (Selenium or guide)

- **Security Evaluation:**
- Session validation, token expiration, and SQL injection exams

Feedback and Refinement

- **User Testing**: Feedback accumulated from a small pattern of check customers.
- **Iteration**: Features were delicate primarily based on usability checking out and worm reviews.

Documentation & Analysis

- Development logs and dash retrospectives have been maintained
- Results were analyzed based totally on system performance and person delight metrics
- Comparative evaluation changed into done with related structures

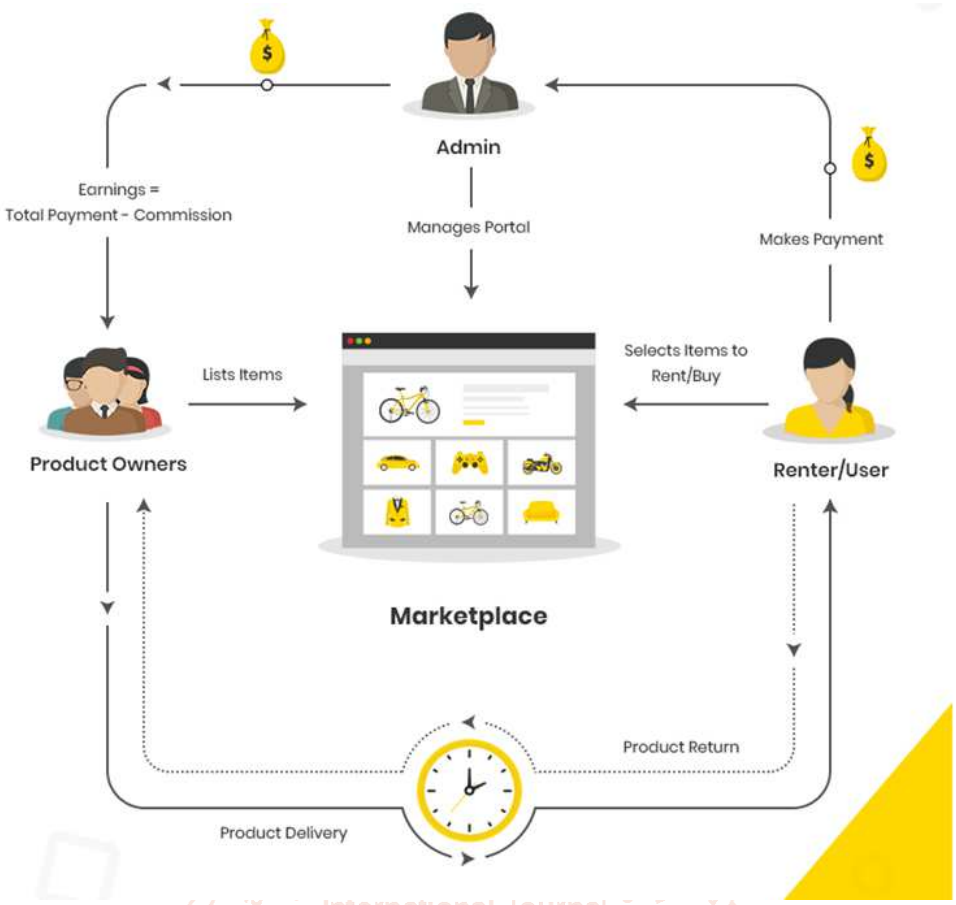

Fig 3. Workflow Diagram of BorrowBiz Rental Lifecycle

IV. RESULT AND CONCLUSION

The development and deployment of the BorrowBiz platform established the effectiveness of the use of a Java complete-stack architecture, incorporated with Spring Boot and microservices, for constructing a scalable and modular apartment e-trade device. The platform accomplished reliably under each everyday and simulated height hundreds, with common API reaction times final within proper limits. The separation of offerings the use of RESTful APIs enabled green communication among person, product, and order modules, even as using Hibernate ORM progressed database interplay efficiency and reduced reaction latency. User checking out discovered a positive revel in average, with contributors locating the interface intuitive and the rental manner straightforward. Administrative features such as inventory control, apartment tracking, and consumer manipulate were carried out easily thru a covered admin dashboard. In evaluation to existing rental platforms, BorrowBiz stood out for its clean architectural design, easier scalability, and superior backend modularity. These strengths function it as a feasible basis for similarly enlargement and integration into large e-trade ecosystems. However, comments from test users indicated room for improvement in person engagement functions such as live chat, product wishlists, and superior filtering—features that could be taken into consideration in future iterations.

V. REFERENCES

[1] Ali, R., Singh, M., & Kumar, A. (2021). Consumer behavior trends in digital condo markets: A information-pushed perspective. Journal of Retail Technology, 45(2), pp. 108–122.

[2] Kumar, A., & Shah, P. (2020). Design of an internet-primarily based digital rental device the usage of PHP and MySQL. International Journal of Computer Applications, 177(14), pp. 12–18.

[3] Sharma, V., & Gupta, P. (2021). Spring Boot and microservices: A scalable solution for e-commerce structures. International Journal of Advanced Research in Computer Science, 12(4), pp. Ninety eight–104.

[4] Goyal, R., & Mishra, K. (2019). RESTful services architecture for modular internet improvement. Procedia Computer Science, 152, pp. 374–380.

[5] Martin, R. C. (2009). Clean code: A guide of agile software program craftsmanship. Prentice Hall.

[6] Newman, S. (2015). Building microservices: Designing exceptional-grained structures. O'Reilly Media.

[7] Deka, G. C. (2018). Building web apps with Spring 5 and Angular. Apress.

[8] Patel, D., & Sinha, R. (2022). Security implementation in e-trade the usage of JWT authentication and function-based totally get entry to manage. International Journal of Network Security & Its Applications, 14(three), pp. 49–60.

[9] Joshi, M., & Verma, S. (2020). User experience enhancement in condo structures via modular UI/UX practices. Journal of Digital Design and Innovation, five(three), pp. 71–79.

[10] Statista. (2023). Rental e-commerce revenue international from 2018 to 2024. Retrieved from

https://www.Statista.Com/data/1244065/global-condominium-e-trade-market-length/

[11] Spring Team. (2023). Spring Boot documentation. Retrieved from https://spring.Io/initiatives/spring-boot

[12] Oracle. (2023). Java Platform, Standard Edition documentation. Retrieved from https://docs.Oracle.Com/javase/

[13] Rentomojo. (2024). About us & era stack. Retrieved from https://www.Rentomojo.Com

[14] Airbnb Engineering. (2021). Microservice structure at scale. Retrieved from https://medium.Com/airbnb-engineering

[15] Fernandez, A. (2021). A evaluate on full-stack Java improvement for scalable platforms. Software Engineering Review, 18(2), pp. One hundred ten–a hundred twenty five.

[16] Singh, A., & Patel, K. (2020). Database optimization for transactional e-commerce structures the usage of Hibernate ORM. International Journal of Data Engineering, 7(1), pp. 43–55.

[17] IBM Cloud Docs. (2023). Microservices reference structure. Retrieved from https://cloud.Ibm.Com/medical doctors

[18] Dehghani, R. (2021). Cloud-local styles: Design scalable systems with microservices, cloud-native structure, and Kubernetes. Manning Publications.

[19] Raj, S., & Bansal, T. (2022). Enhancing e-trade security via Spring Security and OAuth2. Journal of Web Engineering and Security, nine(1), pp. 22–35.

[20] Mehta, Y., & Trivedi, S. (2021). Agile methodology in web software improvement: Case have a look at of a condominium platform. Journal of Agile Systems and Innovation, 6(2), pp. Fifty nine–67.