Design and Development of a Smart Student Rental Platform for Sustainable Campus Communities

Muhammad Danish Mohamad Jafri, Kurk Wei Yi, Nuridawati Mustafa, Raja Rina Raja Ikram, Noorrezam Yusop

University Technical Malaysia Melaka (UTeM), Durian Tunggal, Melaka, 76100, Malaysia

DOI: https://dx.doi.org/10.47772/IJRISS.2025.90900071

Received: 24 August 2025; Accepted: 29 August 2025; Published: 29 September 2025

ABSTRACT

This study explores consumer-to-consumer (C2C) e-commerce systems, with a focus on platforms such as Carousell and Facebook Marketplace, and examines the financial challenges commonly faced by university students. Many students require essential academic or lifestyle items for short-term use but often rely on unofficial channels, such as social media groups, that are unregulated, inefficient, and prone to trust issues. To address this gap, the proposed solution introduces UniLend, a centralized rental-based mobile platform that enables students to lend and borrow items within their academic community. By promoting a "rent-instead-of-owning" model, the system enhances affordability, fosters trust, and encourages sustainability through shared item usage. The system was developed following an agile methodology. Flutter was used to implement the cross-platform mobile client, while Google Firebase supported backend services including authentication, database management, cloud storage, and notifications. Stripe was integrated for secure payment handling. The prototype consists of seven core modules: user authentication, item listing, catalog browsing, search functionality, rental history tracking, integrated payment, and user reviews. Evaluation included black-box testing to validate functional requirements, confirming that all modules operated as expected with both valid and invalid inputs. A usability study is also planned, involving a pilot test with students to gather feedback on navigation, trust in transactions, and clarity of workflows using the System Usability Scale (SUS) and interviews. The results demonstrate the feasibility of a structured, trust-driven rental ecosystem tailored for higher education communities. UniLend contributes to ongoing efforts to develop sustainable, affordable, and student-centered digital platforms.

Keywords—Peer-to-Peer Rental, Mobile Application, Student Affordability, Sharing Economy, Sustainable Campus Communities.

INTRODUCTION

The sharing economy has transformed how individuals access goods and services, with peer-to-peer (P2P) rental platforms emerging as a significant trend. These platforms allow underutilized items to be temporarily rented, offering affordability, convenience, and sustainability compared to outright ownership [1]. The global P2P rental applications market is projected to grow from USD 18.1 billion in 2024 to USD 84.7 billion by 2034, driven by increasing demand for cost-effective and flexible consumption models [2].

This shift is particularly relevant to student communities, where affordability is a major concern and many items are required only for short-term use. Studies show that younger generations, especially students, increasingly prioritize access to goods over ownership, reflecting broader values of sustainability and resource optimization [3]. Nevertheless, students often rely on informal methods such as social media groups or bulletin boards to borrow or lend items. While convenient, these channels lack verification, accountability, and transaction security, exposing students to risks of mistrust, fraud, and inefficiency [4].

To address these challenges, this study introduces UniLend, a student-oriented P2P rental application designed to provide a secure and structured rental ecosystem. UniLend integrates features such as university email verification, catalog browsing, in-app chat, secure payments, and review mechanisms, enabling students to

confidently borrow and lend items within their academic community. By emphasizing affordability, trust, and sustainability, UniLend seeks to alleviate financial burdens, promote responsible consumption, and foster collaboration among students.

Related work

Carousell is a leading consumer-to-consumer (C2C) marketplace in Southeast Asia that enables users to buy and sell both new and secondhand items. Established in 2012, it has expanded into markets such as Singapore, Malaysia, Indonesia, and Hong Kong, offering a platform for affordable goods with simple listing and chat features. Carousell has introduced "Carousell Protection," which secures payments by holding funds until buyers confirm receipt of items (Carousell, n.d.-a). However, its structure primarily supports permanent sales rather than short-term rentals, and its protections remain limited compared to dedicated rental platforms [5].

Fat Llama, on the other hand, was developed as a peer-to-peer rental platform focused specifically on temporary access to items. It provides structured workflows such as identity verification, integrated payments, and damage protection for rented goods [6]. The platform also supports a formal resolution process for disputes, requiring evidence submission and mediation through its dedicated support team (Fat Llama, n.d.-b). These mechanisms provide a stronger sense of security and trust among users. Nevertheless, Fat Llama operates primarily in the United Kingdom, United States, and Canada, which restricts its reach. Additionally, it does not cater to the unique short-term needs of students, focusing instead on a broader market for high-value rentals [7][8].

To bridge these gaps, this study proposes UniLend, a rental platform designed specifically for university students. UniLend combines features of general marketplaces with those of dedicated rental services but adapts them to academic communities. Core functions include university email verification, in-app communication, secure payment processing, rental history tracking, and user reviews to strengthen trust. By integrating accessibility, affordability, and security, UniLend provides a structured ecosystem that enables students to share resources effectively, reduce financial strain, and encourage sustainable consumption on campus. However, adoption of such platforms varies across cultural and academic contexts. In Western universities, where the sharing economy is more established, students are generally more receptive to rental-based platforms and show higher trust in structured digital marketplaces [6][7]. In contrast, many Asian universities rely heavily on informal channels such as social media groups, exposing students to greater risks of fraud and inefficiency [4]. These differences suggest that UniLend's success depends not only on technical functionality but also on its ability to overcome local adoption barriers, including secure payments, digital trust, and cultural norms of student collaboration.

Table I Comparison Between Existed And Proposed System

Criteria	**Carousell**	**Fat Llama**	**Proposed UniLend**
Platform Type	General C2C marketplace for buying and selling items	Dedicated peer-to-peer rental platform	Student-focused peer-to-peer rental platform
Target Users	General public in Southeast Asia	General public in UK, US, Canada	University students and campus communities
Transaction Type	Permanent ownership (buy/sell)	Temporary use (short-term rentals)	Temporary use (short-term rentals)
Trust & Security	Carousell Protection (payment hold until receipt confirmed); limited safeguards	Identity verification, damage protection, dispute resolution team	University email verification, in-app reviews, rental history, secure payments
Key Features	Listings, chat, photo uploads, Carousell	ID verification, insurance coverage,	Item catalogue, chat, notifications, secure payments, rental history,

	Protection	booking workflow, integrated payments	review system
Limitations	No rental workflows, risk of fraud, not tailored for temporary student use	Limited geographic availability, not designed for student academic needs	Prototype stage; requires adoption and trust-building among student communities

METHODOLOGY

The development of UniLend adopted the Agile methodology, which emphasizes iterative progress, continuous feedback, and adaptability to user needs. Agile is well suited for mobile application development, where user experience and rapid prototyping are essential [9]. The process was structured into four phases: planning, system design, development, and evaluation.

A. Planning

In this phase, project objectives were defined, target users identified, and existing C2C and P2P platforms analyzed. Informal surveys and literature reviews were conducted to better understand student needs, particularly short-term rental requirements and the limitations of current informal practices. A product backlog was created to capture essential features, including catalog browsing, chat functionality, secure payments, notifications, and review systems. Requirements were prioritized based on their importance to the student rental experience, and Agile sprints were used to break the backlog into manageable development cycles [10].

B. System Design

The system design of UniLend adopts a three-tier architecture consisting of the client layer, backend services, and third-party integrations.

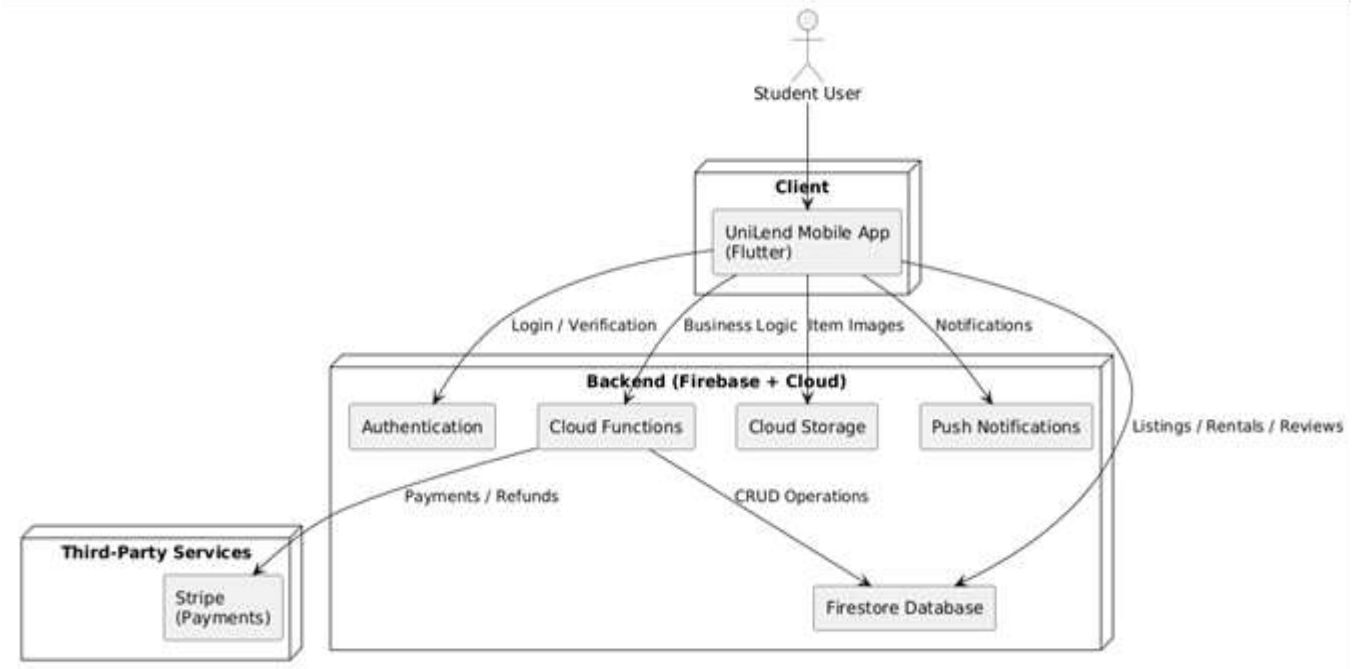

Fig. 1. UniLend architecture diagram

At the client layer, the UniLend mobile application was developed using the Flutter framework to ensure cross-platform compatibility for both Android and iOS devices. This application serves as the primary interface through which students interact with the system, including browsing items, submitting rental requests, sending messages, and managing their rental activities.

The backend layer was implemented on Firebase, leveraging Authentication for secure login and student verification, Firestore for managing structured collections such as users, items, rentals, and reviews, and Cloud Storage for storing item images. Business logic was automated through Firebase Cloud Functions, which handled workflows such as processing rental requests, updating statuses, and coordinating notifications. Real-time updates were supported through Firebase Cloud Messaging (FCM), ensuring that users received timely notifications of rental activities.

Finally, the system integrates with Stripe as a third-party service to manage payments, deposits, and refunds securely. This architecture ensures that UniLend delivers a scalable, secure, and user-friendly rental platform tailored to the needs of university students.

C. Development

The development of UniLend was guided by the Agile methodology, with the system built iteratively through sprints to enable continuous testing and refinement.

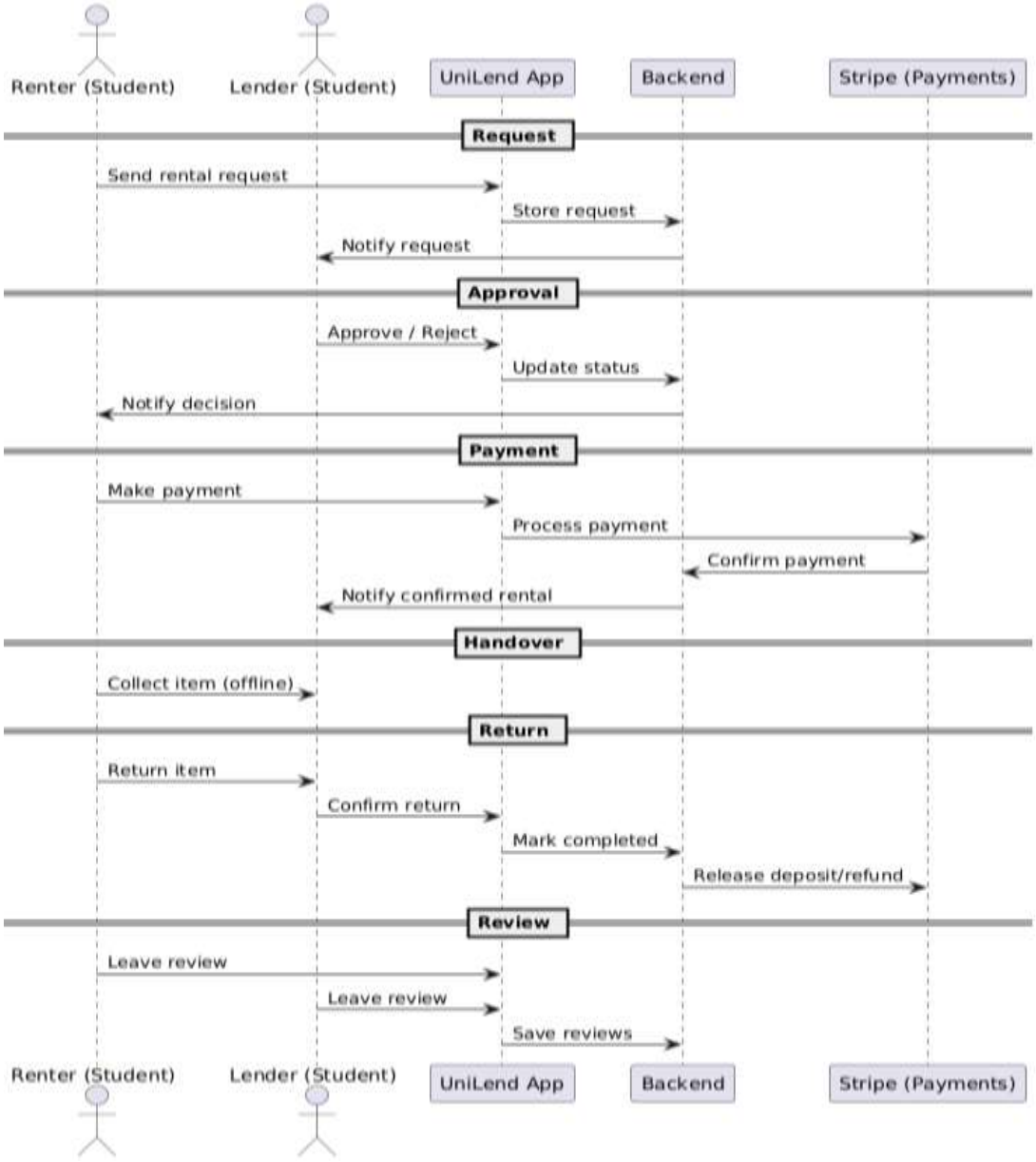

Fig. 2. UniLend rental workflow

The mobile application was implemented using the Flutter framework, selected for its ability to deliver native-like performance across both Android and iOS from a single codebase. This approach reduced development time and ensured consistency in the user experience.

The backend was deployed on Firebase services, with Firestore serving as the main NoSQL database to manage structured data such as user profiles, item listings, rental records, and reviews. Firebase Authentication was integrated to provide secure login and enforce university email verification, ensuring that only verified students could access the platform. Business logic was automated through Firebase Cloud Functions, which handled processes such as rental request validation, status updates, and notification triggers. Cloud Storage was used for the uploading and retrieval of item images, while Stripe was incorporated as a third-party service to process rental payments, deposits, and refunds securely.

A modular approach was adopted, with each sprint delivering specific features such as user authentication, catalogue browsing, chat functionality, and rental management. Each module was tested independently before integration into the overall system, allowing early error detection and maintaining system stability throughout the development cycle

D. Evaluation

1) Functional Testing (Black-Box Testing)

The evaluation of UniLend was conducted using a black-box testing approach, which verifies system functionality based on input–output behavior without requiring knowledge of the internal code structure. This method was appropriate as the objective was to confirm whether the system modules performed according to the specified requirements and provided a seamless experience for student users.

Testing focused on the main functional modules, including user authentication, item listing, rental requests and approvals, payment processing, chat, notifications, and review submission. Each module was tested with both valid and invalid inputs to ensure correct outputs and effective error handling. These tests confirmed that the system successfully managed expected scenarios and provided appropriate feedback to users.

Table II Black-Box Test Cases For Unilend

ID	Test Scenario	Expected Output	Actual Result	Status
TC-01	Valid login with university email	Login successful, user redirected to home screen	As expected	Pass
TC-02	Invalid email format	Error message: “Invalid email format. Please use university email.”	As expected	Pass
TC-03	Create new item listing	Item saved to database and visible in catalogue	As expected	Pass
TC-04	Request item with valid details	Rental request sent to lender; lender receives notification	As expected	Pass
TC-05	Request item with missing duration	Error message: “Please specify rental duration.”	As expected	Pass
TC-06	Successful payment	Payment processed, rental status updated to Confirmed, notifications sent	As expected	Pass
TC-07	Failed payment (insufficient funds)	Error message: “Payment failed. Please try again.”	As expected	Pass

TC-08	Send rental status update	Renter receives push notification: "Your request has been approved."	As expected	Pass
TC-09	Confirm item return	Rental marked as Completed, deposit refunded via Stripe	As expected	Pass
TC-10	Submit review after rental completion	Review saved in database, linked to lender's profile	As expected	Pass

2) Usability Study (Planned Pilot Implementation)

In addition to functional testing, a usability study is planned to evaluate the effectiveness and user experience of UniLend among university students. A pilot test will involve a small group of 10–15 students from different academic backgrounds, who will be asked to perform core rental activities such as account creation, item browsing, rental request submission, and payment processing. Their feedback will be collected using the System Usability Scale (SUS) questionnaire, supplemented by short interviews to capture qualitative insights regarding ease of navigation, trust in payment handling, and clarity of the rental workflow.

This mixed-method approach will provide both quantitative usability scores and qualitative feedback on potential barriers, such as concerns over data security, trust in peer transactions, and willingness to share personal items. Findings from the pilot will guide iterative improvements of the prototype before scaling to a larger deployment across campus communities, following usability testing frameworks outlined in prior research [15].

RESULT

The implementation of UniLend produced a functional mobile application prototype designed to facilitate peer-to-peer item rentals within university communities. The following subsections present the core modules of UniLend, illustrated with screenshots of the application interface.

A. Login and Register

Figure 3 presents the login and registration interface. Users are required to enter their email and password to access the system, while new users must provide their name, email address, and password to create an account.

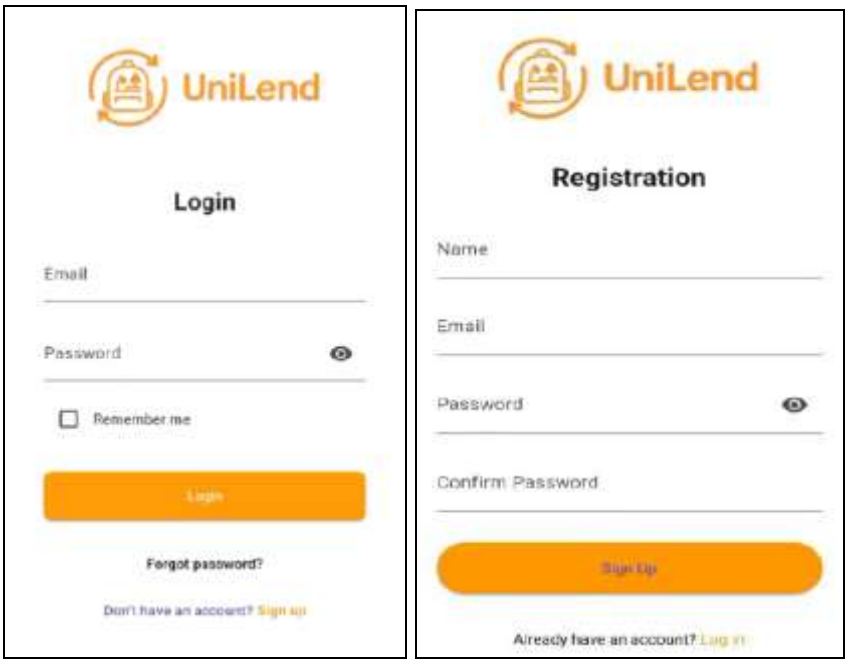

Fig. 3. Login and Register interface

B. Main page and Search

As shown in Figure 4, the main page displays item cards that users can select to view item details. A category icon allows users to filter items by category, while the bottom navigation bar provides direct access to the search page.

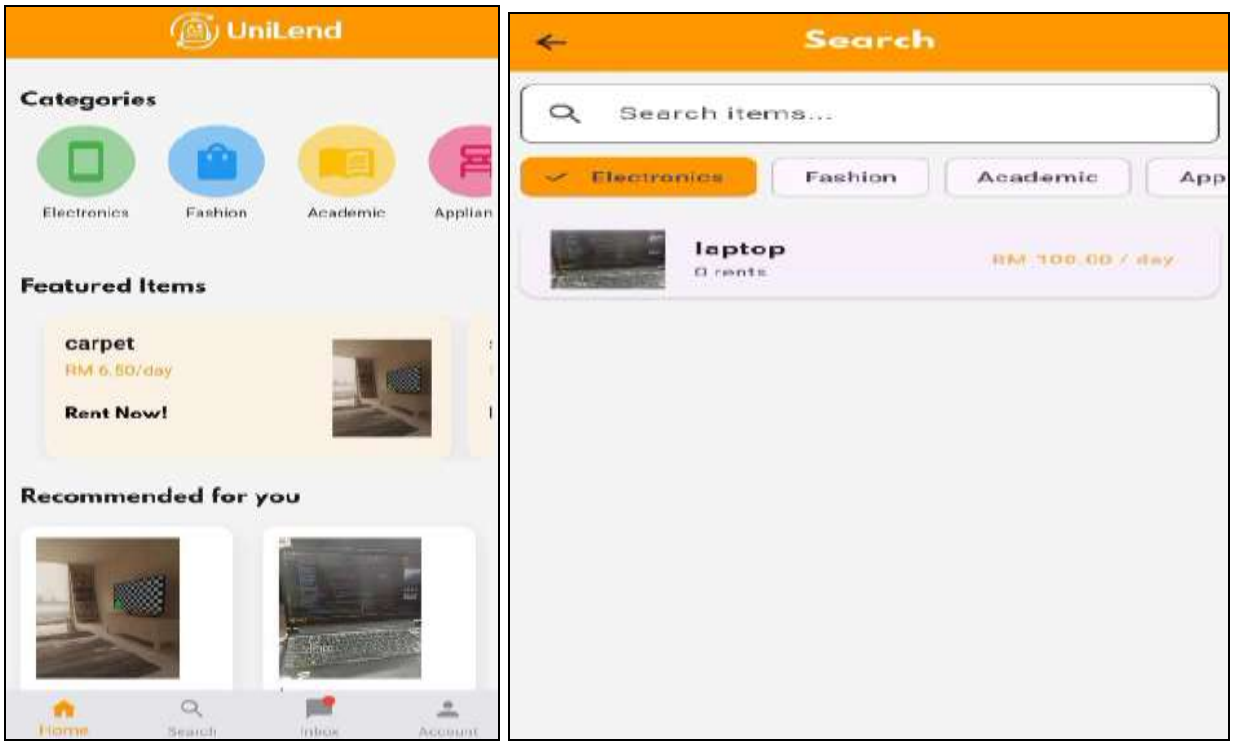

Fig. 4. Main and Search interface

C. Item Detail and Chat Box

Figure 5 illustrates the item detail page, which provides information on availability and item owner details. A "Chat" button enables direct communication between the renter and the owner prior to submitting a rental request through the "Request Rent" option.

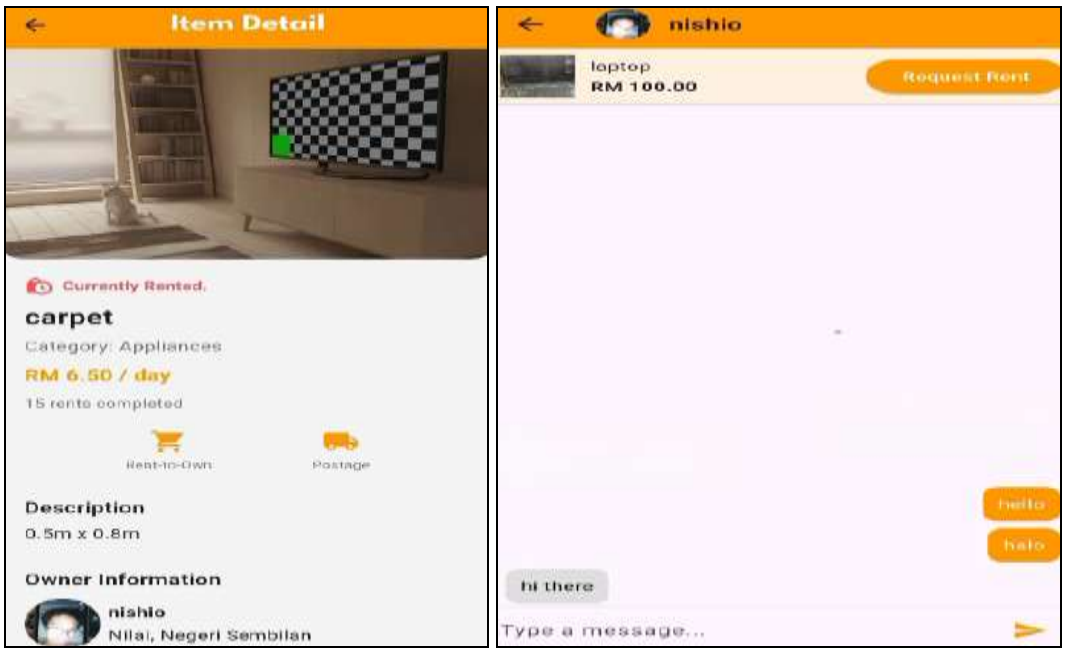

Fig. 5. Item Detail and Chat Box interface

D. Rent Request and Finalizing Rental

Once an owner approves a rental request, a “Finalize Rental” button appears, as shown in Figure 6. This allows the renter to confirm the transaction and proceed with finalizing the rental.

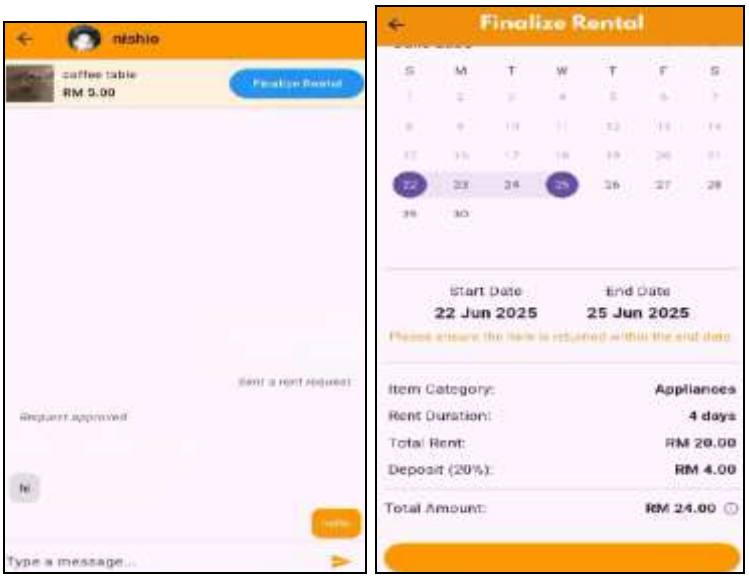

Fig. 6. Rent Request and Finalizing Rental interface

E. Payment

Figure 7 displays the Stripe payment interface, which processes transactions after rental details are finalized.

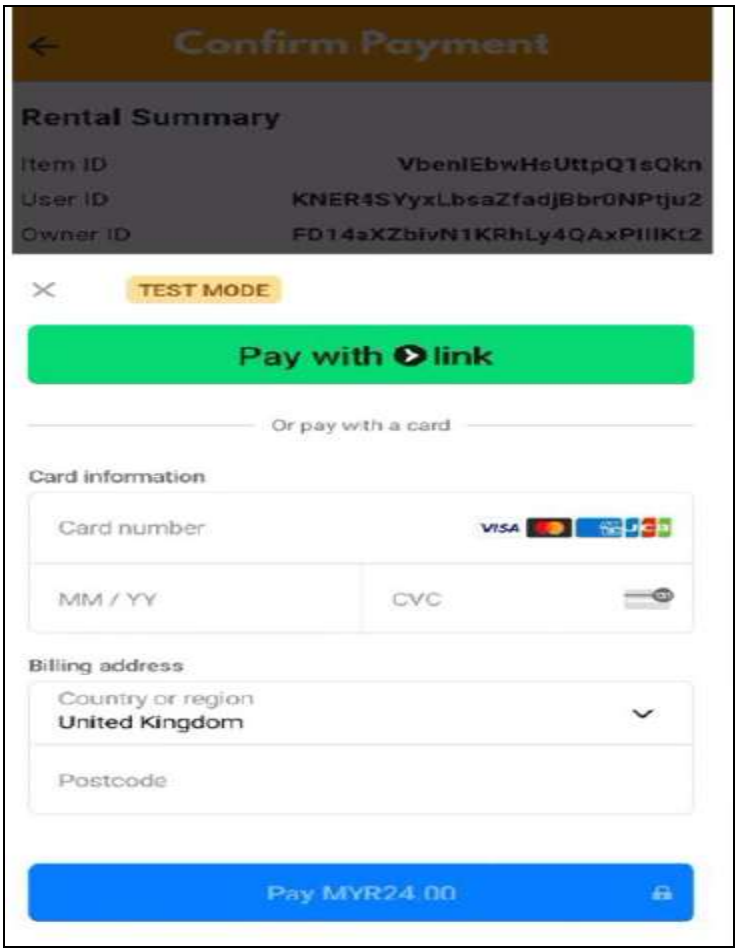

Fig. 7. Payment interface

F. Inbox

The inbox interface is shown in Figure 8. Unread notifications are highlighted in orange and flagged with an indicator. Notifications are categorized into “Renting” (items borrowed by the user) and “Rented Out” (items lent by the user). A separate “Chat” tab lists all conversations, organized into “Incoming Requests” (chats about the user’s items) and “My Requests” (chats about items borrowed by the user).

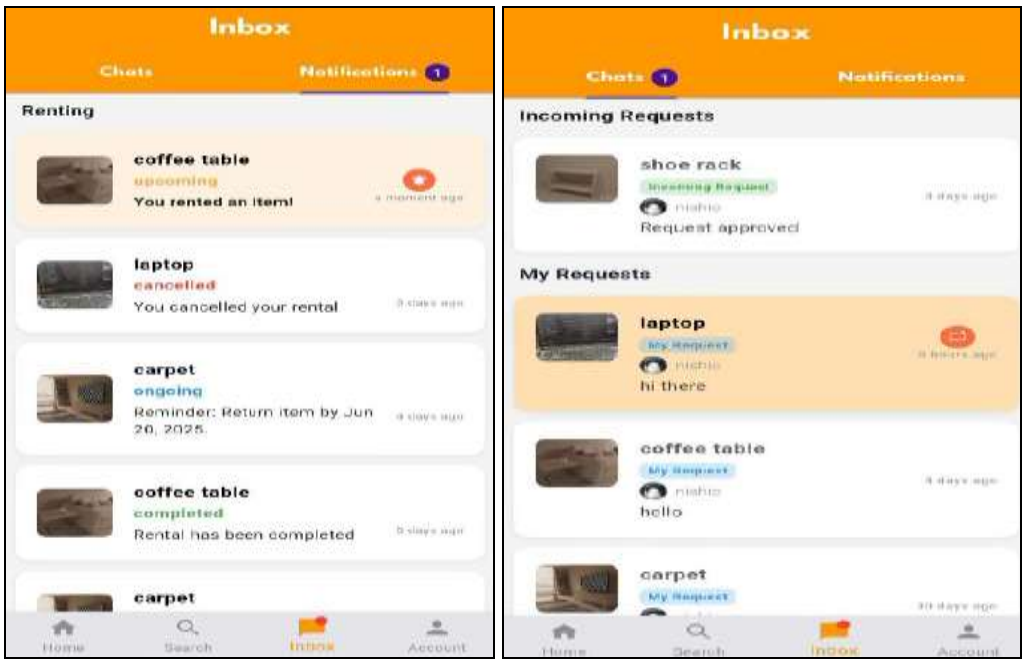

Fig. 8. Inbox interface

G. Upcoming and Ongoing Rental

Figure 9 shows the rental details page. Items marked as “Upcoming” indicate rentals awaiting pickup. Both renter and owner must confirm pickup before the status changes to “Ongoing.” Renters may cancel a booking before pickup. Once the rental is active, the interface displays status updates and return reminders.

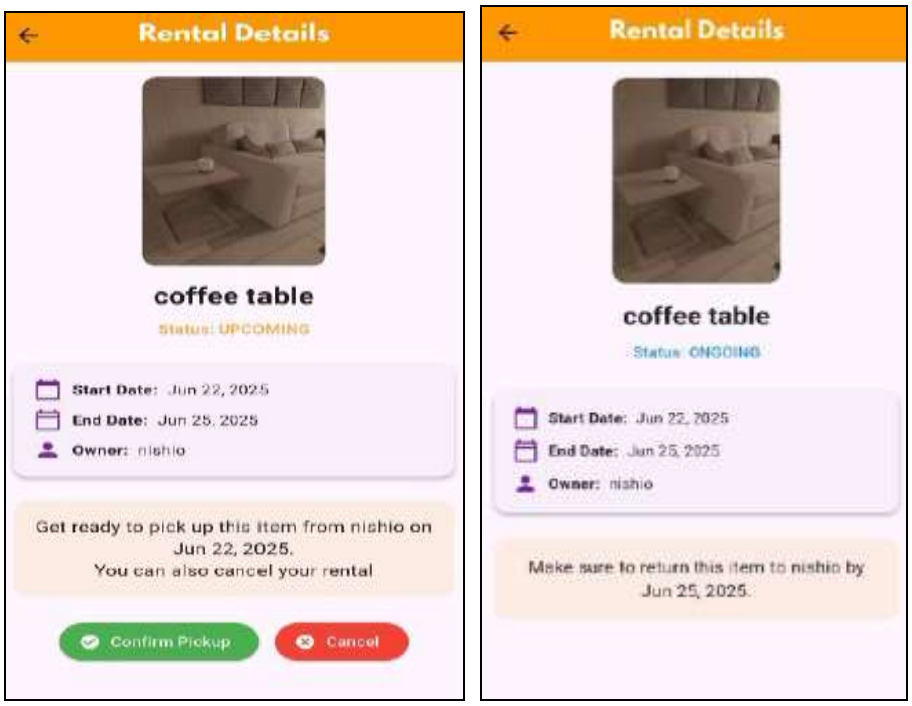

Fig. 9. Upcoming and Ongoing Rental interface

H. Return and Completed Rental

As shown in Figure 10, the "Confirm Return" option is available at the end of the rental period. Both renter and owner must confirm the return to complete the transaction and mark it as successfully closed.

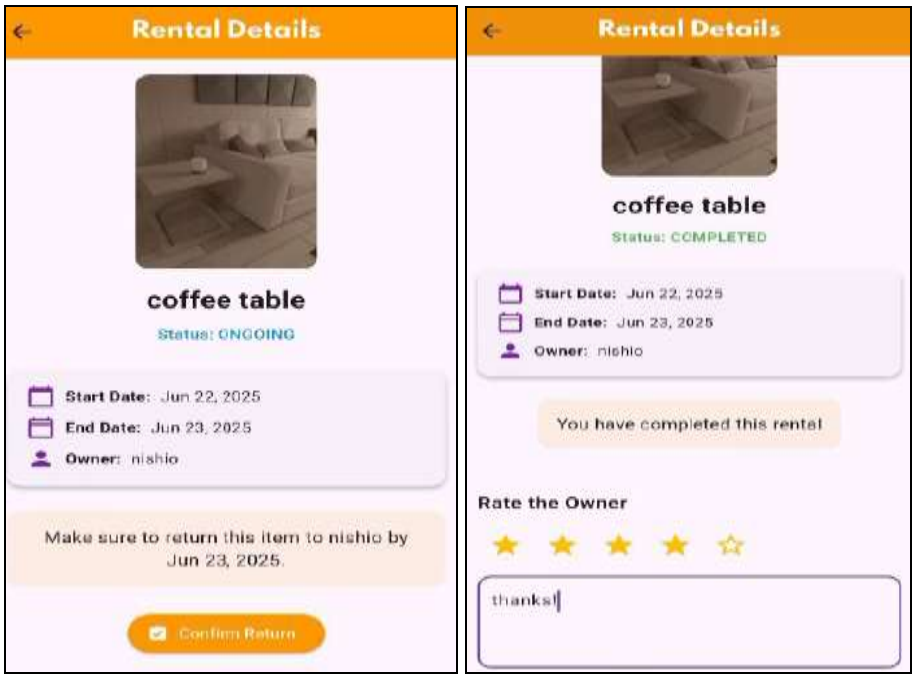

Fig. 10. Return and Completed Rental interface

I. Review and Rental History

Figure 11 presents the review feature, where feedback is displayed on the item detail page under the owner's profile. The "My Rents" tab provides users with a history of ongoing and completed rentals, categorized into items rented and items rented out.

Fig. 11. Review and Rental History interface

J. Item Listing and Item Management

Figure 12 illustrates the item management interface. Users can list new items by uploading images, specifying a name, category, rent rate per day, and handover method. Existing listings can also be updated or removed through this module.

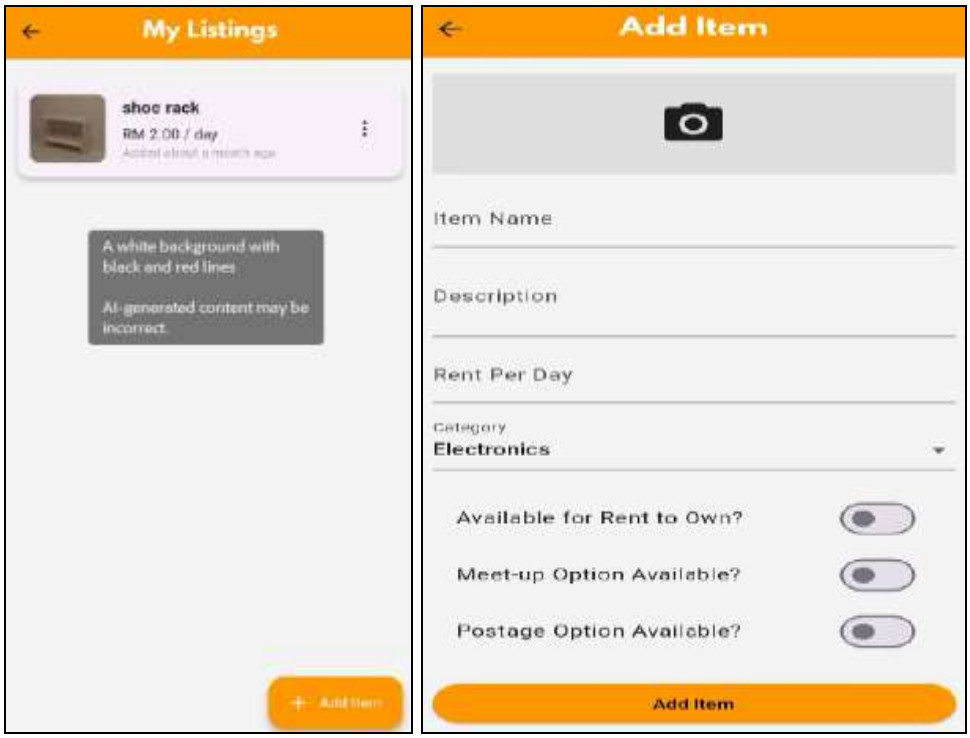

Fig. 12. Item Listing and Item Management interface

CONCLUSION AND FUTURE WORKS

This study presented the design and development of UniLend, a prototype mobile application that enables peer-to-peer rentals among university students. By incorporating features such as student email verification, secure payments, and review mechanisms, UniLend addresses key challenges related to affordability, trust, and sustainability within campus communities. The prototype demonstrates that a structured rental ecosystem can complement traditional buying and selling platforms by facilitating temporary access to academic and lifestyle items.

Despite these contributions, several limitations remain. The prototype is still at an early stage, with evaluation limited to black-box testing and pending large-scale usability validation. Adoption is also influenced by factors such as students' willingness to share personal items, trust in digital transactions, and acceptance of security protocols. Potential barriers include concerns over data privacy, dispute resolution, and the need for incentives to encourage active participation.

Future work will expand evaluation through usability studies and pilot deployments with larger student populations. Planned directions include integrating the platform with existing campus IT systems, introducing incentive mechanisms such as reward points or community recognition, and conducting comparative studies across cultural and academic contexts to examine differences in trust dynamics and sharing practices. Addressing data security and payment dispute mechanisms will also be critical to scaling UniLend into a fully deployable student rental ecosystem.

By acknowledging current limitations and outlining future directions, this study not only demonstrates the technical feasibility of UniLend but also contributes to broader discussions on sustainable, trust-driven digital platforms for higher education communities.

ACKNOWLEDGEMENT

The authors would like to express gratitude to Centre of Technology for Disaster Risk Reduction (CDR), Faculty technology Maklumat dan Komunikasi (FTMK), University Technical Malaysia Melaka (UTeM) for their invaluable support and resources provided throughout this research.

REFERENCES

1. Campbell, J. (2024). Peer-to-peer (P2P) rental. EBSCO Research Starters. Retrieved from https://www.ebsco.com/research-starters/economics/peer-peer-p2p-rental
2. Market.us. (2025, January 14). P2P rental apps market to hit USD 84.7 billion by 2034, growing at 11.2% CAGR. Scoop Market. Retrieved from https://scoop.market.us/p2p-rental-apps-market
3. Wired. (2025, April 16). Young people are making up to $36K a year renting their T-shirts and speakers. Wired. Retrieved from https://www.wired.com/story/young-people-are-making-up-to-dollar36k-a-year-renting-their-t-shirts-and-speakers/
4. Hatamleh, I. H. M., Safori, A. O., Habes, M., Tahat, O., Ahmad, A. K., Abdallah, R. A.-Q., & Aissani, R. (2023). Trust in social media: Enhancing social relationships. Social Sciences, 12(7), 416. https://doi.org/10.3390/socsci12070416
5. Carousell. (n.d.-a). About Carousell Group. Carousell Press. Retrieved August 22, 2025, from https://press.carousell.com/carousell-group/
6. Fat Llama. (n.d.-c). About Fat Llama. Fat Llama. Retrieved August 22, 2025, from https://fatllama.com/about-us
7. Fat Llama. (n.d.-a). How does Fat Llama work? Fat Llama FAQ. Retrieved August 22, 2025, from https://faq.fatllama.com/en/articles/10398626-how-does-fat-llama-work
8. Fat Llama. (n.d.-b). The resolutions process. Fat Llama FAQ. Retrieved August 22, 2025, from https://faq.fatllama.com/en/articles/10401663-the-resolutions-process
9. Laoyan, S. (2025, February 20). What is Agile methodology? (A beginner's guide). Asana. Retrieved August 22, 2025, from https://asana.com/resources/agile-methodology
10. Guerrero-Ulloa, G., Rodríguez-Domínguez, C., & Hornos, M. J. (2023). Agile methodologies applied to the development of Internet of Things (IoT)-based systems: A review. Sensors, 23(2), 790. https://doi.org/10.3390/s23020790
11. Google. (2024). Firebase documentation. Firebase. Retrieved August 22, 2025, from https://firebase.google.com/docs
12. Ali, B. (2024). Seamless payment integration in Flutter using Stripe and Firebase Cloud Functions: A step-by-step guide. Medium. Retrieved from https://medium.com/@basitalyshah51214/seamless-payment-integration-in-flutter-using-stripe-and-firebase-cloud-functions-a-step-by-step-84513bea2d05
13. Firebase. (2025). Process payments with Firebase. Firebase Documentation. Retrieved from https://firebase.google.com/docs/tutorials/payments-stripe
14. Wired. (2025, April 16). Young people are making up to $36 K a year renting their T-shirts and speakers. Wired. Retrieved from https://www.wired.com/story/young-people-are-making-up-to-dollar36k-a-year-renting-their-t-shirts-and-speakers/
15.] Weichbroth, P. (2024). Usability testing of mobile applications: A methodological framework. Applied Sciences, 14(5), 1792. https://doi.org/10.3390/app14051792