Development of Android-based Rental System for Mountaineering Equipment

Mohammed N. Abdulrahman Elias[1]*, Fikri Ahmad Taufiqulhakim[2], and Umi Najiah[3,]

[1] Computer Center, University of Mosul, Mosul, Nineveh, Iraq
[2] Faculty of Information Sciences and Engineering, Management and Science University, Shah Alam, Selangor, Malaysia
[3] Faculty of Information Sciences and Engineering, Management and Science University, Shah Alam, Selangor, Malaysia
***Corresponding email:** mohammad.nawfal@uomosul.edu.iq, 012020092212@fise.msu.edu.my

Received 27 May 2025; **Revised** 22 June 2025; **Accepted** 7 July 2025; **Published** 28 July 2025

Abstract: *The proliferation of mobile commerce has not uniformly addressed the needs of specialized, high-stakes recreational activities. Mountaineering, for instance, presents unique logistical and safety challenges that generic rental platforms fail to accommodate, creating a distinct research gap in domain-specific usability and system design. This study investigates the efficacy of applying a user-centered design (UCD) and Agile development framework to create a specialized mobile rental system tailored for the mountaineering community. The research aimed to determine the critical success factors for designing such a niche application and to evaluate the suitability of an iterative development model in this context. Adopting a constructive research approach, a mobile application for Android was developed as the primary artifact to explore the research problem. The Agile methodology guided the iterative development, while a mixed-methods evaluation, including observational analysis and user usability surveys, provided empirical data. The findings demonstrate that the synthesis of Agile and UCD principles is highly effective, leading to a measurable 2-4% enhancement in key usability metrics across development cycles. The final artifact was validated by target users, depending on the feedback indicating a positive reception to the app's design and functionality — affirming its intuitive design and effectiveness in addressing their specific needs. This thesis contributes a validated model for developing mobile applications in niche recreational markets, offering insights into the critical interplay between domain-specific requirements and software development methodologies. The results confirm that for high-stakes user groups, a tailored design process is paramount and provides a framework for future research in similar specialized domains.*

Keywords: Mountaineering, Android, Agile, equipment, rental system.

1. Introduction

The rapid expansion of mobile technology has fundamentally reshaped consumer behavior, particularly within the sharing economy and service industries. As internet usage and smartphone and tablets based on Android — Android is a mobile operating system that is a collection of open and free software from Google for mobile devices, such as smartphones. The main applications, middleware, and operating system are included in it [1] –- adoption has become ubiquitous, users increasingly expect convenient, on-demand access to goods and services through mobile applications. This trend extends to the growing field of adventure tourism and recreation, such as mountaineering, where specialized equipment that does not break down and can function properly over a long period is essential but often prohibitively expensive for casual participants, making rental services a critical enabler.

Despite the growth of mobile commerce, the mountaineering equipment rental sector has been underserved by technological innovation. Many rental businesses still rely on traditional, inefficient methods. Potential renters are often required to visit stores in person, leading to significant time consumption. For administrators, the process of documenting rental data, tracking equipment inventory, and managing availability is typically a manual and error-prone task. This situation reveals a distinct research gap: a lack of specialized, user-centered mobile systems designed to address the unique logistical challenges of the mountaineering equipment rental domain. The objectives of the project are: Firstly, to create an Android-Based Mountaineering Equipment Rental System, so customers do not have to come when they want to rent equipment directly. Secondly, to facilitate administrators in documentation and administration by providing a system that helps manage equipment rental data. Thirdly, to provide an option for a renter to search for mountaineering equipment based on the list available in the catalog.

Furthermore, to address these challenges, this study details the design, development, and evaluation of a specialized Android-based Mountaineering Equipment Rental System. The primary aim of this research is to create a seamless and efficient digital bridge between rental providers and mountaineering enthusiasts. The system is designed not only to provide a convenient rental platform for customers but also to offer a robust administrative tool that automates and streamlines equipment management, booking, and data documentation. For instance, **"The future of service will likely involve more proactive and predictive elements, often delivered through mobile interfaces that learn user preferences and behaviors to offer tailored assistance"** [21].

The contributions of this research are multifaceted, impacting both practical application and academic understanding. Primarily, this work presents the design and implementation of a novel, domain-specific mobile application. Unlike generic rental platforms, this system is tailored to the unique safety and logistical requirements of mountaineering, representing a significant step forward for technology in this niche field. Secondly, the research introduces a solution that tangibly enhances user convenience and accessibility. By enabling remote browse of equipment catalogs, real-time availability checks, and secure online ordering, it removes critical friction from the customer journey and aligns the rental process with modern consumer expectations. For service providers, this study offers a centralized system that dramatically simplifies the complex tasks of administration. The automation of equipment tracking and rental data management not only improves operational efficiency but also reduces the potential for human error and provides valuable data for business intelligence. Finally, this project serves as a practical and validated case study on the application of an Agile development methodology to create a user-centric solution for a niche market, offering a repeatable framework for developers tackling similar challenges.

There is a research methodology in this project: the questionnaire and observation used in the project to gather information on the usability of the mobile application developed among users. The process of collecting data needs is done by distributing questionnaires to the community to determine the need for renting mountaineering equipment online and the level of public interest in using the information system. Observation is to understand the development of the era by looking for information from various media and sources and observing various existing applications as reference material to design and develop this information system.

The software development in this study uses the Agile software development methodology. Sustainable planning, learning, enhancement, group cooperation, evolution in development, and early delivery are the approaches used in Agile software development. A core characteristic of the Agile Software Development Model is its firm emphasis on "learning by doing," a stark contrast to traditional models that often demand extensive upfront planning and design. [1] Agile methodologies categorically reject the notion of Big Design Up Front (BDUF), recognizing that for many complex projects, it is impractical or even impossible to define all requirements exhaustively at the outset [21]. Mobile devices, especially smartphones, are often carried around 24/7 and are typically turned on at all times. This constant accessibility allows users to perform a wide range of activities—such as communication, information search, entertainment, and commerce—whenever and wherever they desire. This 'anytime, anywhere' characteristic fundamentally enhances convenience, reduces dead time, and enables individuals to manage their personal and professional lives more efficiently. Mobile applications, tailored for specific tasks, further amplify this convenience by providing streamlined and optimized user experiences for these on-the-go interactions, profoundly impacting how people access services and information in their daily lives [2]. Although the number of meetings involved in Agile may appear wasteful, it saves time during the planning stage by decreasing errors and optimising tasks for development [3]. The Agile software development life cycle is divided into six phases, which will be discussed in the literature review.

This paper begins by reviewing the relevant literature concerning mobile rental systems and software development methodologies. Following this, we detail the research methodology, outlining the system architecture and the adopted Agile framework. The subsequent section presents the results of the system's implementation and the findings from our user testing. The paper concludes with a summary of our key findings and offers recommendations for future work.

2. Literature Review

The current technological landscape is based on research that is defined by a significant and growing number of mobile application users. Globally, smartphone ownership continues to rise, and research indicates that users spend the vast majority of their mobile time—up to 90%—within dedicated applications rather than on mobile websites. This shift has profoundly impacted various sectors, including transportation and financial services, by offering enhanced convenience and creating new channels for consumer engagement. The rise of mobile applications has profoundly impacted various sectors in developing economies, including transportation, financial services and personal safety [4]. A key driver of this trend is a consumer preference for renting over purchasing, particularly when the decision is influenced by convenience and current trends. The results of other studies on reasons for preferring renting show that orientation of convenience and trends positively impact consumer's decision to rent rather than purchasing kinds of stuff [6]. Customers increasingly seek services and products that save them time, and mobile applications have emerged as a transformative force in delivering this value.

This transformation is clearly visible in the transportation and tourism sectors. In the car rental industry, for instance, mobile apps have reshaped user expectations by providing real-time availability checks, direct booking, price comparisons, and mobile check-in. Consequently, the strategic development and deployment of effective mobile applications are critical for competitiveness and customer satisfaction in the modern travel and transportation landscape [8]. These platforms often integrate features like GPS, digital payments, and customer reviews, creating a more transparent and user-centric experience. The shift towards mobile-first solutions underscores a broader trend of digitalization in services, where accessibility and ease of use are paramount for customer satisfaction and market competitiveness [7]. This model of enhanced convenience and accessibility is not limited to cars. The rapid development of dockless bike-sharing systems, largely facilitated by mobile applications, has significantly impacted urban mobility. The success of these apps hinges on their usability, the reliability of the information provided, and the seamlessness of the user experience from locating to unlocking and paying for a bicycle. Furthermore, factors such as perceived convenience, cost-effectiveness, and environmental benefits contribute to the attractiveness of app-based bike-sharing [9]. Similarly, in recreational tourism, platforms catering to campervan and motorcycle rentals have emerged. Wicked Ride, for example, supports guided motorcycle tours with backup vehicles and included insurance, while Wicked Campers offers budget-friendly campervan hires, appealing to a "slow travel" demographic seeking autonomy and a closer connection to the environment. The appeal lies in the blend of transport and accommodation, offering a self-contained travel experience that can be adapted to individual preferences and a slower rhythm of exploration [10].

While these existing applications demonstrate a mature model for mobile rentals in their respective domains, a critical analysis reveals their limitations when applied to the high-stakes context of mountaineering. The features that make a car or bike rental app successful—primarily speed and inventory volume—do not address the paramount importance of equipment reliability, detailed technical specifications, and safety history required for climbing gear. The rental of a campervan is a lifestyle choice; the rental of a carabiner is a matter of personal safety. Furthermore, the motivations for why individuals engage in mountaineering are unique, often involving deep personal goals like socialization, escape, or self-development [5]. This suggests the need for a service that goes beyond a simple transactional interface to potentially include community features, expert advice, and curated gear packages—elements entirely absent in the aforementioned rental models.

Therefore, this review of related systems and literature identifies a clear and significant gap. The existing models for mobile rental applications are insufficient and ill-equipped to handle the specific logistical, safety, and motivational needs of the mountaineering community. The proposed system in this study is designed to fill this void by providing a specialized, trustworthy, and user-centric platform. To build a system that can effectively address these nuanced requirements, an Agile development methodology was selected for its iterative nature, which allows for continuous feedback and refinement based on the specific needs of this expert user group.

3. Proposed Method

The software development in this study uses the Agile software development methodology. Sustainable planning, learning, enhancement, group cooperation, evolution in development, and early delivery are the approaches used in Agile software development. There are phases of the Agile software development life cycle.

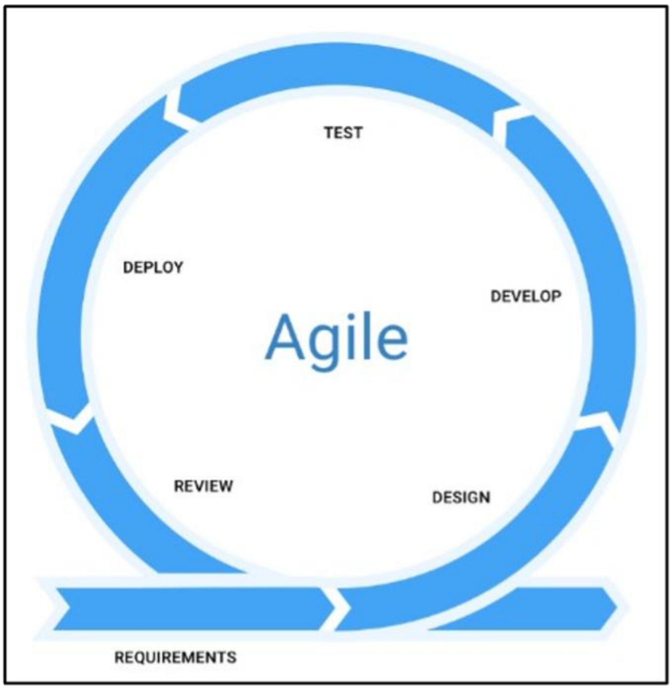


Figure 1: The agile sprint cycle

The project was implemented using an iterative approach built around a series of two-week development cycles, or **sprints**. Before development began, an initial **product backlog** of all necessary features, framed as user stories, was created. The work then proceeded through multiple iterations, with each two-week sprint following the six core activities shown in Figure 4. The following list describes the process that was **repeated in every sprint cycle**:

1. **Requirements:** At the beginning of each two-week cycle, a selection of high-priority requirements for that specific sprint was pulled from the main product backlog. This focused the team's effort on a small, manageable set of features [3].
2. **Design:** The team designed the necessary user interface and system architecture to fulfill the requirements chosen for that sprint [11].
3. **Develop:** The coding and implementation of the designed features were carried out by the development team [12].
4. **Test:** The newly written code was immediately tested within the same sprint to identify and resolve bugs, ensuring the quality of the work [3].
5. **Deploy:** At the end of the two-week cycle, the functional, tested increment of the software was deployed to a staging server, making it ready for demonstration and review [3].
6. **Review:** The work completed during the sprint was reviewed by the team and stakeholders. Feedback gathered during this review was used to guide the selection of requirements for the next sprint, thus completing the iterative loop and ensuring the project adapted to feedback continuously [3].

The Agile movement is anchored by a set of core values and supporting principles, most famously articulated in the "Manifesto for Agile Software Development." These tenets represent a significant departure from traditional software development philosophies, emphasizing human-centric collaboration, adaptability, and the consistent delivery of value, formulated by Beck et al. in 2001, outlines four foundational values that prioritize certain aspects of software development over others [22]. These values are:

1. **Individuals and interactions** over processes and tools.
2. **Working software** over comprehensive documentation.
3. **Customer collaboration** over contract negotiation.
4. **Responding to change** over following a plan.

The type of research methodology that is used is survey and observation. The instrument used in the survey is an online questionnaire which is Google Forms, that is distributed to the community. The questionnaire consists of two sections which were Section A (Demographic Data) and Section B (Application Usability Testing). Observation is to understand the development of the era by looking for information from various media and sources and observing various existing applications as reference material to design and develop this information system. The hardware used in developing this application is personal property. The specifications of the hardware used are presented in Table 1.

Table 1. Hardware Specifications

Hardware	Specifications
G40-70 Laptop (Lenovo)	Intel(R) Core(TM) i7-4510U CPU 2.6GHz 4096MB RAM 500GB SSD
Oppo A1601 (Phone)	Octa-core 1.5 GHz Cortex-A53 Android 5.1 (Lollipop), ColorOS 3 Mediatek MT6755 (28 nm) or MTK7650 (28 nm) 32GB 3GB RAM

There is some software used in developing this application. The specifications of the software used are presented in Table 2.

Table 2. Software Specifications

Software	Specifications
Windows 10	Operating systems developed by Microsoft were used to develop and test the application.
Android Studio	Code editor for Android application development.
MySQL Database Service	The database system used on the web and runs on a server.
Visual Studio Code	Code editor for building web.
MEmu Play	The software serves as a special Android emulator.

The design of the system can use the system model, which is the Unified Modeling Language (UML). UML in this system consists of several diagrams such as use case diagrams, activity diagrams, class diagrams, and sequence diagrams. In addition, this development uses a flowchart that consists of a user system flowchart and an admin system flowchart.

The interface design is designed to show the function and the feature that consists of each system page. The interface component is the object used inside the interface to deliver information [13]. Adapting the interface to the needs of the user can keep them from forsaking the application [14]. In the development of this project, interface design consists of interface design for user pages and interface design for admin pages.

The system is implemented using a three-tier client-server architecture to ensure scalability and separation of concerns, as depicted in Figure [add your new figure number here]. This architecture is composed of a presentation tier (the clients), a logic tier (the server), and a data tier (the database).

- **Presentation Tier (Clients):** To accommodate the distinct roles of users and administrators, the system provides two separate client applications. For renters, the primary interface is a native **Android mobile application**, providing a portable and user-friendly experience. For system administration, a comprehensive **web-based dashboard** is provided, allowing the administrator to manage equipment inventory, user accounts, and rental transactions from a web browser.

- **Logic Tier (Backend Server):** All business logic, data processing, and user authentication are handled by a central backend server. Both the Android app and the admin web dashboard communicate with this server through a **REST API**. All data is transmitted securely over **HTTPS** to protect data in transit.
- **Data Tier (Database):** The system's data, including user profiles, equipment details, bookings, and rental history, is persistently stored in a **MySQL database**. The backend server is solely responsible for interacting with the database via a secure connection, executing SQL queries to retrieve and store information.

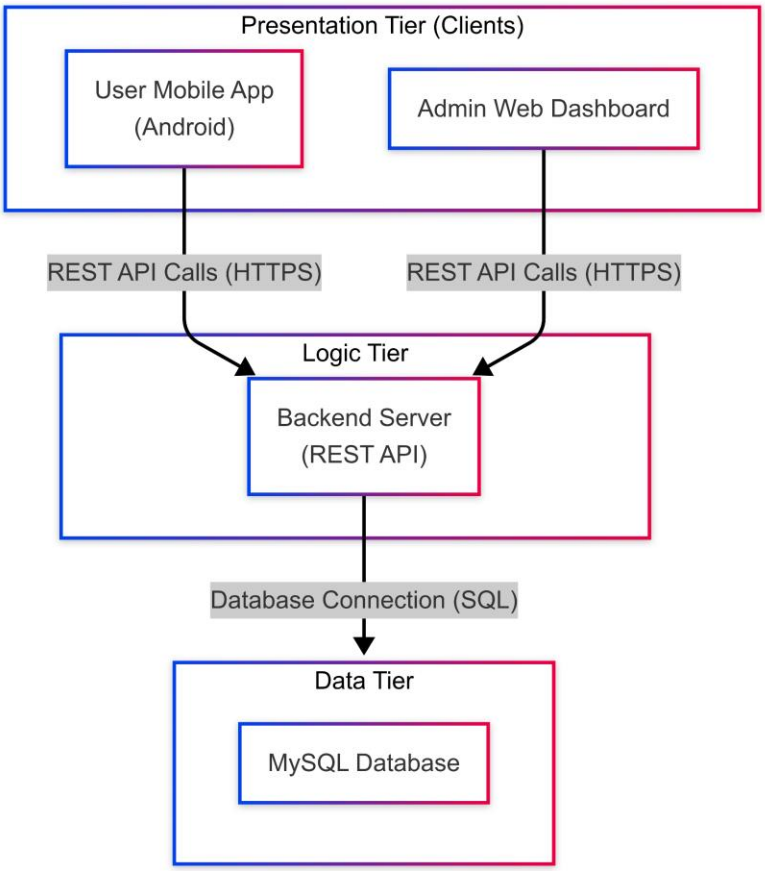
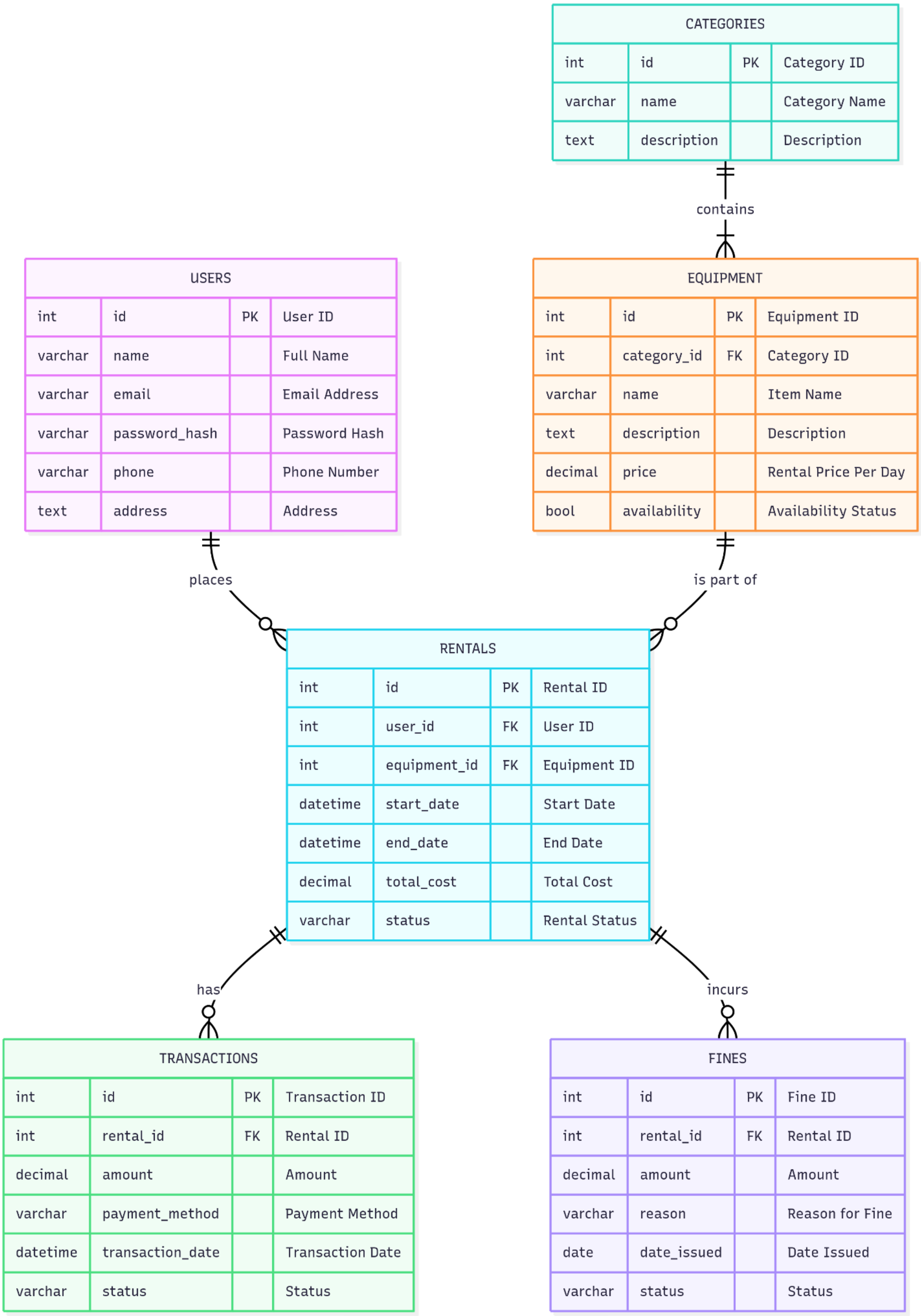

Presentation Tier (Clients)
User Mobile App (Android)
Admin Web Dashboard
REST API Calls (HTTPS)
REST API Calls (HTTPS)
Logic Tier
Backend Server (REST API)
Database Connection (SQL)
Data Tier
MySQL Database

Figure 2: Flow chart illustrating system tiers

The data persistence layer is managed by a MySQL relational database. The database schema, illustrated in the Entity-Relationship Diagram (ERD) in Figure [add your new figure number here], is designed around six core tables: **Users**, **Categories**, **Equipment**, **Rentals**, **Transactions**, and **Fines**.

- The **Users** table stores all personal and authentication information for registered customers.
- The **Categories** table defines the different types of equipment available for rent (e.g., 'Tents', 'Climbing Shoes'), allowing for easy management.
- The **Equipment** table contains detailed attributes for each rental item and is linked to the **Categories** table via a foreign key.
- The **Rentals** table serves as a central hub, linking users and equipment for each booking event. It also acts as the parent record for financial events, connecting to both the **Transactions** table, which logs all payments, and the **Fines** table, which records any penalties. This normalized structure ensures data integrity and allows for comprehensive tracking of all system activities.



CATEGORIES

int	id	PK	Category ID
varchar	name		Category Name
text	description		Description

USERS

int	id	PK	User ID
varchar	name		Full Name
varchar	email		Email Address
varchar	password_hash		Password Hash
varchar	phone		Phone Number
text	address		Address

EQUIPMENT

int	id	PK	Equipment ID
int	category_id	FK	Category ID
varchar	name		Item Name
text	description		Description
decimal	price		Rental Price Per Day
bool	availability		Availability Status

RENTALS

int	id	PK	Rental ID
int	user_id	FK	User ID
int	equipment_id	FK	Equipment ID
datetime	start_date		Start Date
datetime	end_date		End Date
decimal	total_cost		Total Cost
varchar	status		Rental Status

TRANSACTIONS

int	id	PK	Transaction ID
int	rental_id	FK	Rental ID
decimal	amount		Amount
varchar	payment_method		Payment Method
datetime	transaction_date		Transaction Date
varchar	status		Status

FINES

int	id	PK	Fine ID
int	rental_id	FK	Rental ID
decimal	amount		Amount
varchar	reason		Reason for Fine
date	date_issued		Date Issued
varchar	status		Status

Figure 3: Database schema

The system incorporates several industry-standard measures to ensure data confidentiality, integrity, and user security. The security framework is built on the following principles:

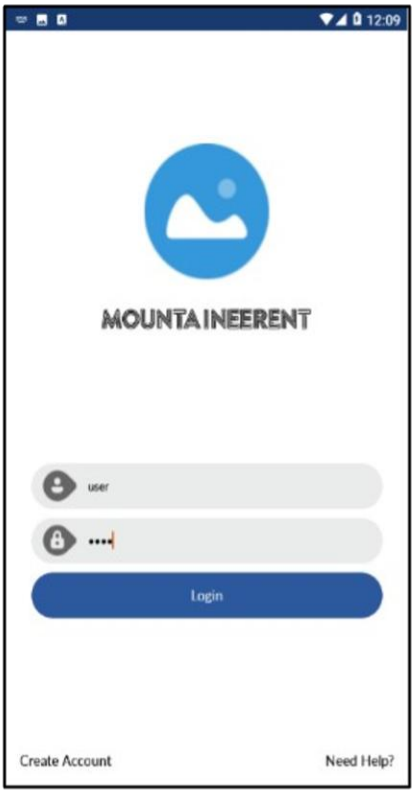
- **User Authentication and Authorization:** User authentication is managed using **JSON Web Tokens (JWT)**. When a user registers, their password is not stored in plain text. Instead, it is converted into a secure, one-way hash using the **SHA-256** algorithm before being saved to the database. Upon a successful login, the server generates and sends a signed JWT to the client application. To access protected routes and data, this token must be included in the header of all subsequent API requests, ensuring that only authenticated and authorized users can perform actions.
- **Data Transmission and Server Security:** All communication between the client applications (Android and web) and the backend server is encrypted using **HTTPS**, protecting data from eavesdropping during transit. On the server side, the risk of database attacks is mitigated by using an **Object-Relational Mapper (ORM)**, which handles database queries programmatically and inherently prevents SQL injection vulnerabilities.
- **Infrastructure Protection:** The entire system is deployed behind **Cloudflare**, which provides an additional layer of security. This service offers protection against common web threats and Distributed Denial-of-Service (DDoS) attacks, enhancing the overall availability and resilience of the application.

4. Result and Discussion

There are several types of tests included in the development of this project which are project testing, usability testing, user acceptance testing, unit testing, and system testing. Software testing is broadly recognized as a crucial phase within the software development lifecycle, dedicated to evaluating a software item to detect differences between given input and expected output. This phase systematically examines the behavior of a program or system under specified conditions to assess its quality, verify that it meets the specified requirements, and identify any defects. The ultimate goal of the testing phase is to provide stakeholders with comprehensive information about the quality of the product under test, thereby supporting informed decisions about its release and deployment, and ensuring it functions as intended in the hands of the end-users [15]. The objective of the testing phase is to examine and evaluate stated requirements, features, and expectations of the project before deployment to ensure that the project complies with the foremost requirements be avowed in documents of specification. Usability testing is a technique used to assess the usability of a system [16].

Next User acceptance testing (UAT) focuses on demonstrating that the system is fit for its intended purpose and meets the business needs of its users. This testing is typically carried out by the users themselves, or representatives of the users, often in their own environment or a simulated operational setting that closely mirrors it. UAT serves as a final verification of the required business functionality and usability from the end-user’s perspective, aiming to provide confidence that the system can support day-to-day business operations effectively and efficiently before it is formally accepted and goes live. It's less about finding every bug and more about confirming overall readiness for the business environment [17]. Acceptance testing enables customers to validate all requirements for a software system. It is conducted by the end user and can range from an informal 'test drive' to a planned and systematically executed series of tests. In fact, acceptance testing can be conducted over a period of weeks or even months, thereby uncovering cumulative errors that might otherwise degrade the system over time. The primary goal is to ensure the software is fit for its intended purpose from the end-user's perspective before deployment, confirming that it meets their business needs and operational workflows in a real-world or simulated real-world environment [18]. That questionnaire is distributed to the community. The questionnaire consists of two sections which were Section A (Demographic Data) and Section B (Application Usability Testing).

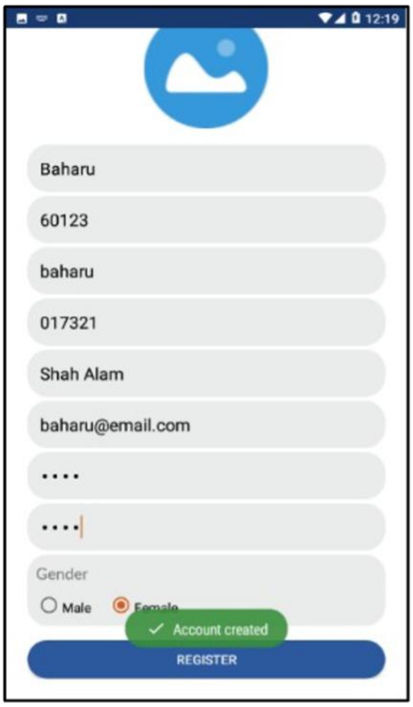
Furthermore, Unit testing is a fundamental software testing practice where individual components or 'units' of software—such as functions, methods, procedures, or modules—are tested in isolation to verify their correctness. The primary objective is to ensure that each unit performs as designed before it is integrated with other parts of the system. This early detection of defects at the smallest testable software level significantly reduces the cost and complexity of bug fixing later in the development cycle. Effective unit testing often involves writing automated test cases that can be run repeatedly and consistently, contributing to improved code quality and providing living documentation [19]. Unit testing conceptually divides a programme into discrete pieces and tests each piece separately, whereas integration tests ensure that the pieces work together properly as a whole. Unit testing is divided into two categories: user application and admin application. In this project, the user application uses an android-based system, while the admin application uses a web-based system.



user

••••

Figure 4: shows a feature in which users can access the main menu after entering their username and password on the login page. After successfully logging in, a pop-up appears, and the user is taken to the main menu.



Baharu

60123

baharu

017321

Shah Alam

baharu@email.com

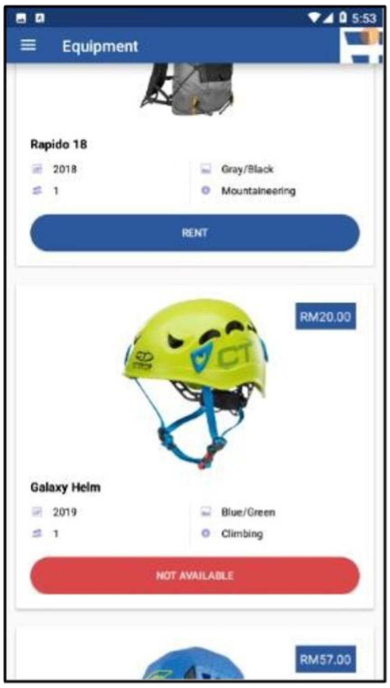
••••

••••

Gender ○ Male ● Female

✓ Account created

Figure 5: shows a feature when the user account is successfully registered, the data will be saved in the database, and the user will be able to log in with that account.


5:53

Equipment

Rapido 18

2018 Gray/Black

1 Mountaineering

RENT

RM20.00

Galaxy Helm

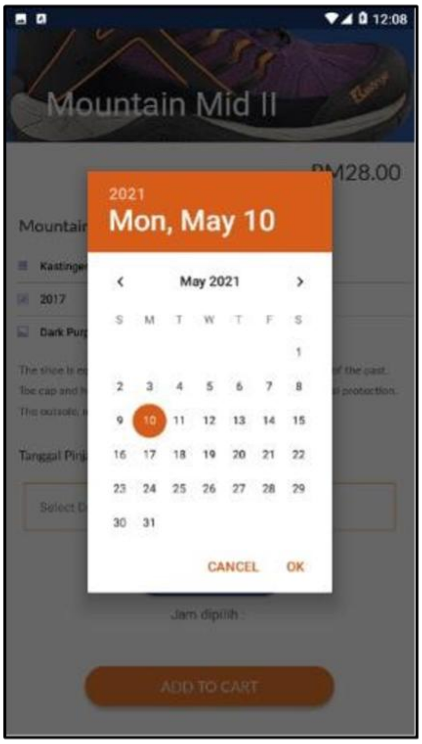
2019 Blue/Green

1 Climbing

NOT AVAILABLE

RM57.00

Figure 6: shows a feature that the users can view the equipment catalog and select equipment available on the main page. Users can view product details by selecting one of the equipment in the catalogue.


12:08

Mountain Mid II

RM28.00

Mountair

Kastinge

2017

Dark Pur

Tanggal Pinj

Select D

2021

Mon, May 10

May 2021

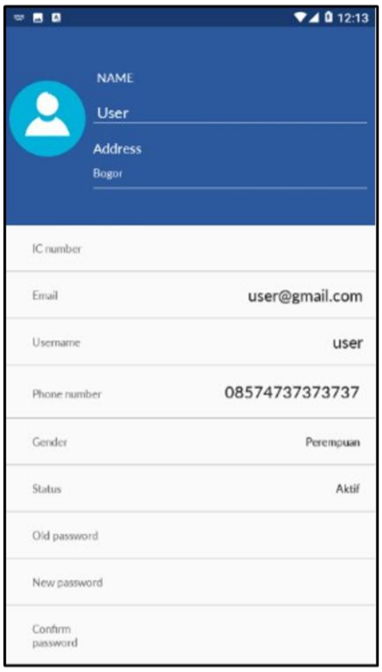
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

CANCEL OK

Jam dipilih :

ADD TO CART

Figure 7: shows a feature that the users can order by selecting the available equipment and then setting the rental time. After the order is complete, the order details will appear on the cart page.



NAME

User

Address

Bogor

IC number

Email user@gmail.com

Username user

Phone number 08574737373737

Gender Perempuan

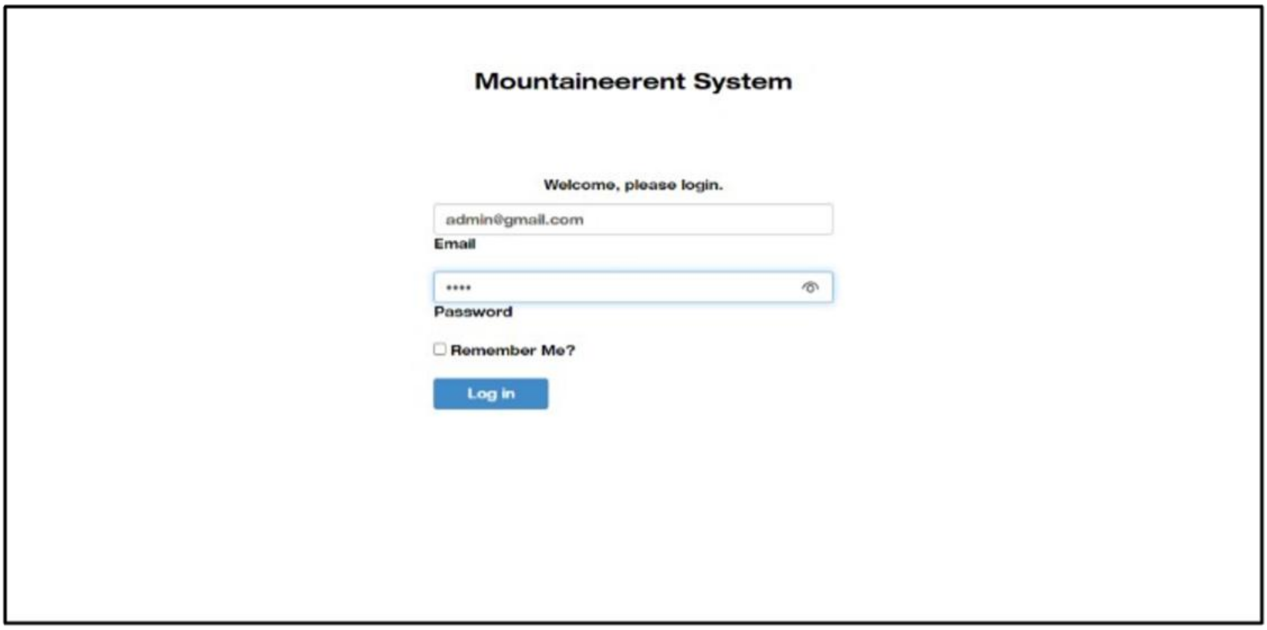
Status Aktif

Old password

New password

Confirm password

Figure 8: shows a feature that the users can update their profile data, and then it is displayed in both the database and the application.



Mountaineerent System

Welcome, please login.

admin@gmail.com

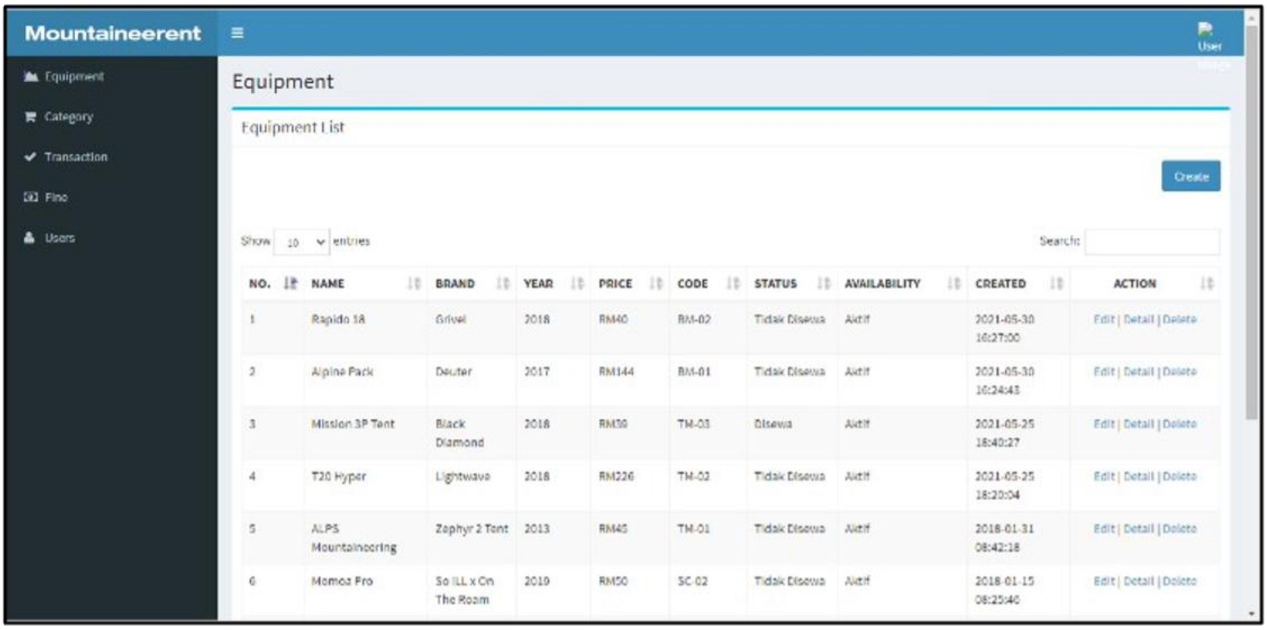
Email

••••

Password

□ Remember Me?

Figure 9: shows a feature that the admin can access the main menu after first entering their username and password on the login page.

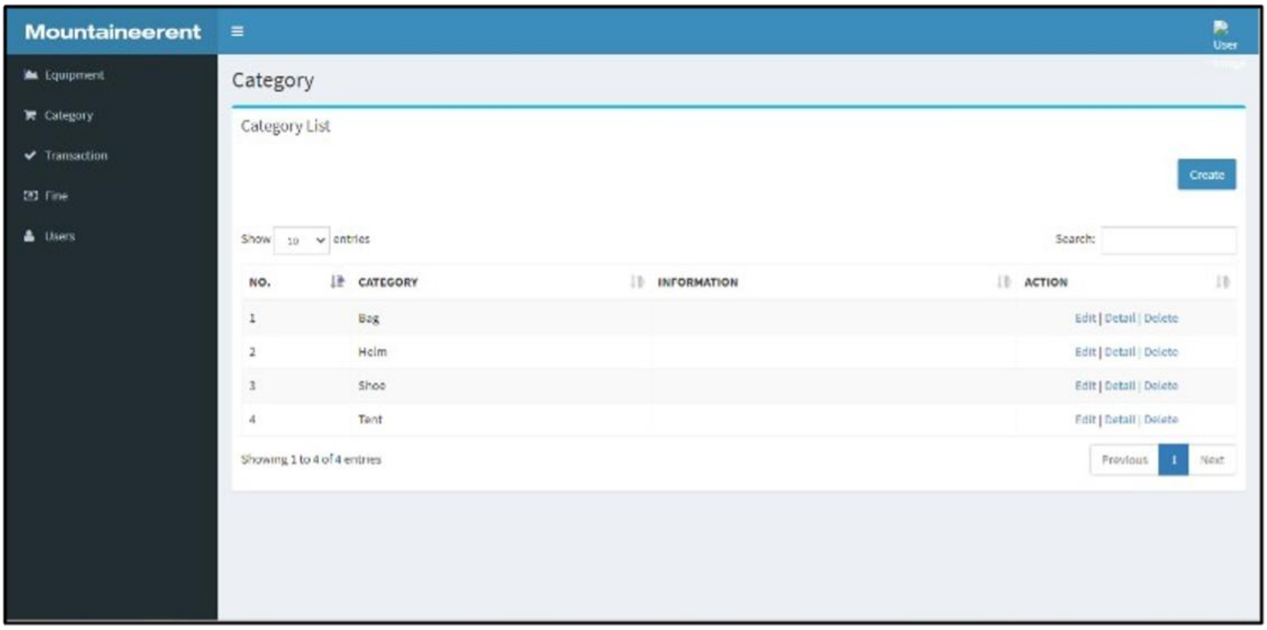


Equipment

Equipment List

NO.	NAME	BRAND	YEAR	PRICE	CODE	STATUS	AVAILABILITY	CREATED	ACTION
1	Rapido 18	Grivel	2018	RM40	BM-02	Tidak Disewa	Aktif	2021-05-30 16:27:00	Edit | Detail | Delete
2	Alpine Pack	Deuter	2017	RM144	BM-01	Tidak Disewa	Aktif	2021-05-30 16:24:43	Edit | Detail | Delete
3	Mission 3P Tent	Black Diamond	2018	RM30	TM-03	Disewa	Aktif	2021-05-25 18:40:27	Edit | Detail | Delete
4	T20 Hyper	Lightwave	2018	RM226	TM-02	Tidak Disewa	Aktif	2021-05-25 18:20:04	Edit | Detail | Delete
5	ALPS Mountaineering	Zephyr 2 Tent	2013	RM45	TM-01	Tidak Disewa	Aktif	2018-01-31 08:42:18	Edit | Detail | Delete
6	Momoz Pro	So ILL x On The Roam	2010	RM50	SC-02	Tidak Disewa	Aktif	2018-01-15 08:25:40	Edit | Detail | Delete

Figure 10: shows a feature that the admin can create, search, edit, view details, and delete equipment data.



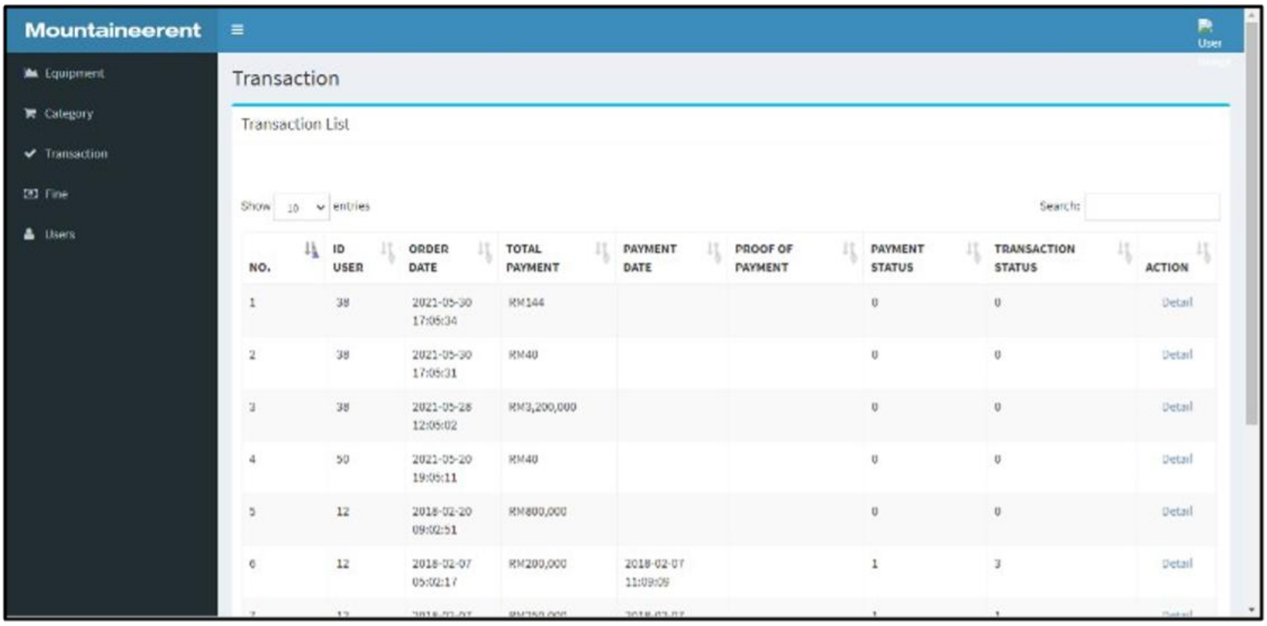
Category

Category List

NO.	CATEGORY	INFORMATION	ACTION
1	Bag		Edit | Detail | Delete
2	Helm		Edit | Detail | Delete
3	Shoe		Edit | Detail | Delete
4	Tent		Edit | Detail | Delete

Showing 1 to 4 of 4 entries

Figure 11: shows a feature that the admin can create, search, edit, view details, and delete category data.

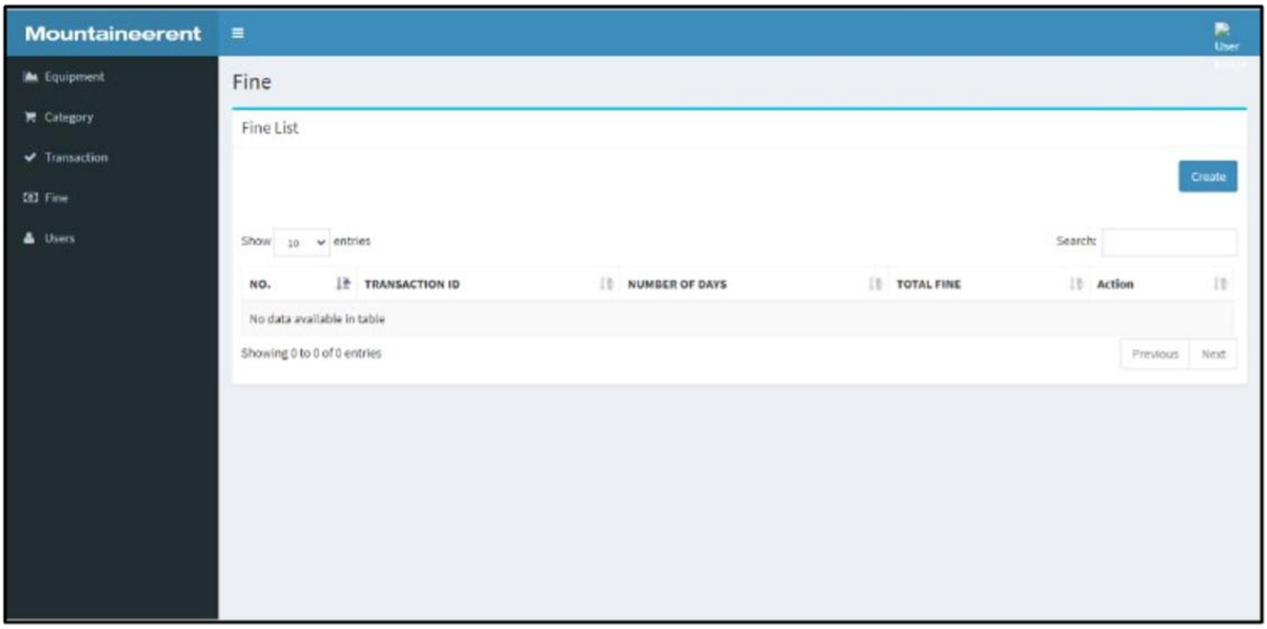


Transaction

Transaction List

NO.	ID USER	ORDER DATE	TOTAL PAYMENT	PAYMENT DATE	PROOF OF PAYMENT	PAYMENT STATUS	TRANSACTION STATUS	ACTION
1	38	2021-05-30 17:06:34	RM144			0	0	Detail
2	38	2021-05-30 17:06:31	RM40			0	0	Detail
3	38	2021-05-28 12:06:02	RM3,200,000			0	0	Detail
4	50	2021-05-20 19:06:11	RM40			0	0	Detail
5	12	2018-02-20 09:02:51	RM800,000			0	0	Detail
6	12	2018-02-07 05:02:17	RM200,000	2018-02-07 11:09:09		1	3	Detail
7	12	2018-02-07	RM250,000	2018-02-07		1	1	Detail

Figure 12: shows a feature that the admin can view details, confirm, and cancel orders from users.



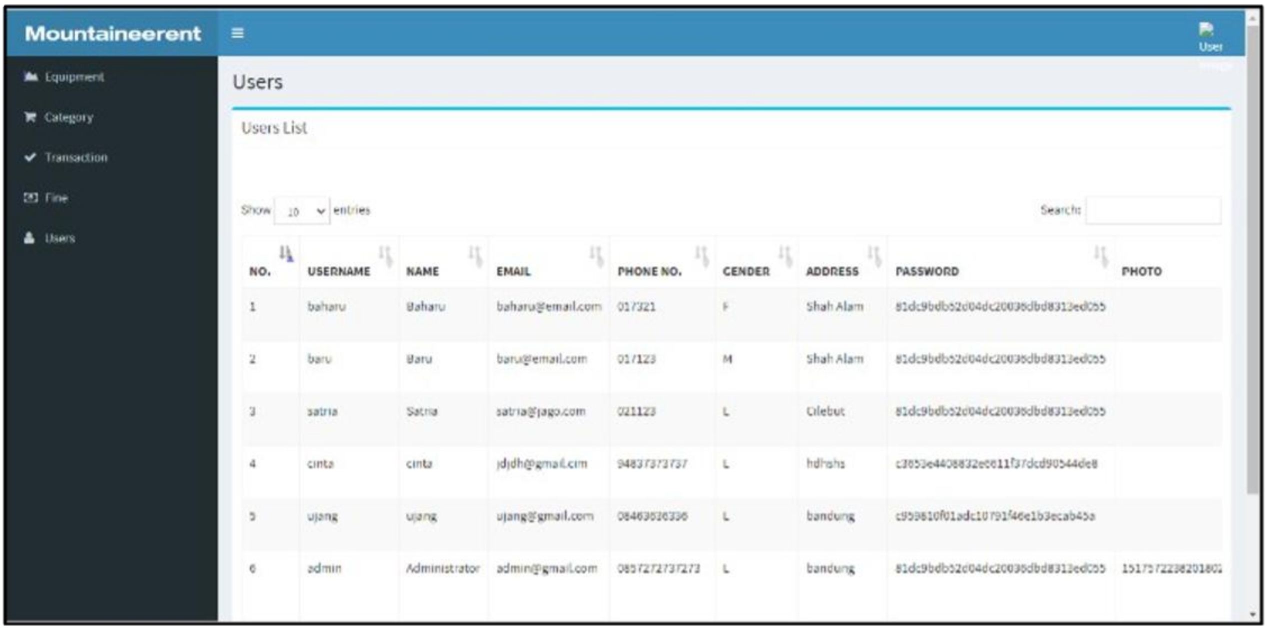
Fine

Fine List

NO.	TRANSACTION ID	NUMBER OF DAYS	TOTAL FINE	Action
No data available in table				

Showing 0 to 0 of 0 entries

Figure 13: shows a feature that the admin can create, search, edit, view details, and delete fine data.



Users

Users List

Show 10 entries

Search:

NO.	USERNAME	NAME	EMAIL	PHONE NO.	GENDER	ADDRESS	PASSWORD	PHOTO
1	baharu	Baharu	baharu@email.com	017321	F	Shah Alam	81dc9bdb52d04dc20036dbd8313ed055	
2	baru	Baru	baru@email.com	017123	M	Shah Alam	81dc9bdb52d04dc20036dbd8313ed055	
3	satria	Satria	satria@jago.com	021123	L	Cilebut	81dc9bdb52d04dc20036dbd8313ed055	
4	cinta	cinta	jdjdh@gmail.cim	94837373737	L	hdhshs	c3653e4408832ec611f37dcd90544de8	
5	ujang	ujang	ujang@gmail.com	08463636336	L	bandung	c959810f01adc10791f46e1b3ecab45a	
6	admin	Administrator	admin@gmail.com	0857272737273	L	bandung	81dc9bdb52d04dc20036dbd8313ed055	151757223820180

Figure 14: shows a feature that the admin can search, view details, and delete users' data.

For system testing — System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of system testing is to detect defects within the system as a whole, in contrast to unit testing or integration testing that focus on smaller pieces. System testing is typically considered a form of black-box testing, meaning that knowledge of the inner design of the code or logic is not required [20]. It is used to evaluate the end-to-end system specifications. Based on the tests carried out, the results showed that the three objectives of this study could be achieved with this application.

5. Conclusion

According to the results of the testing, users can create an account to access this application by entering the personal information requested during registration, users can access the main menu after first entering their username and password on the login page, users can view the equipment catalog and select equipment available on the main page, users can order equipment by selecting the available equipment and then set the rental time, and users can update their profile data, then it is displayed in both the database and the application. Then in the results, the three objectives in this study can be achieved by using this application. This system can make it easier for people who want to climb the mountain to find the equipment needed and help reduce community expenses because the cost of renting is cheaper than buying the equipment. Based on the results, several things can be suggested to make this application more effective. Firstly, the application requires financial technology features to make it easier for users to rent the required equipment. Secondly, this application was designed to be accessible via iOS as well so that application users are not limited to Android users, as there is an increasing number of iOS users. Lastly, the application is developed to be able to use multiple languages; this development is so that the application can be used by various groups who understand specific languages.

ACKNOWLEDGMENT

The author is praised and grateful to Allah SWT because of His grace and mercy. Then, the author would like to express his heartfelt appreciation to the author’s supervisor. The author anticipates constructive criticism and suggestions for improving this final project.

REFERENCES

[1] Beal, V. A. N. G. I. E. (2018). Mobile operating systems (Mobile OS) explained. *Hardware & Software*.

[2] Kaplan, A. M. (2012). If you love something, let it go mobile: Mobile marketing and mobile social media 4x4. *Business horizons*, *55*(2), 129-139.

[3] Feoktistov, I. (2020). Agile Software Development Lifecycle Phases Explained. *Relevant Software*, *5*.

[4] Balogun, A. L., Salimon, M. G., Soetan, A. F., Okeleye, K. A., Siyanbola, A. A., & Ibidunni, A. S. (2024). Bridging the digital divide: Consumer engagement with transportation payment apps in emerging economies. *Journal of Consumer Behaviour, 23*(6), 3011–3029.

[5] Ewert, A. (1985). Why people climb: The relationship of participant motives and experience level to mountaineering. *Journal of Leisure Research*, *17*(3), 241-250.

[6] Moeller, S., & Wittkowski, K. (2010). The burdens of ownership: reasons for preferring renting. *Managing Service Quality: An International Journal*, *20*(2), 176-191.

[7] Luqmani, M., Yavas, U., & Quraeshi, Z. A. (1994). A Convenience- oriented Approach to Country Segmentation: Implications forGlobal Marketing Strategies. *Journal of Consumer Marketing*, *11*(4), 29-40.

[8] Wang, D., Xiang, Z., & Fesenmaier, D. R. (2016). Smartphone use in everyday life and travel. *Journal of travel research*, *55*(1), 52-63.

[9] Shen, Y., Zhang, X., & Zhao, J. (2018). Understanding the usage of dockless bike sharing in Singapore. *International Journal of Sustainable Transportation*, *12*(9), 686-700.

[10] Dickinson, J., & Lumsdon, L. (2010). Slow travel and tourism. Routledge.

[11] Ramli, R., Rozzani, N., & Krishnasamy, R. (2019, October). Examining the effectiveness and user experiences using FabulousFitness: a personalized interactive fitness hologram system. In *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)* (pp. 74-78). IEEE.

[12] Mustapha, S., Hao, C. C., & Ramli, R. (2019, December). Analyzing and enriching English language proficiency in children using interactive storybook. In *2019 IEEE 7th Conference on Systems, Process and Control (ICSPC)* (pp. 96-101). IEEE.

[13] Alexander, O., & Ishak, Z. (2018). The influence of user interface component on university website towards student enrollment. *International journal of engineering & technology*, *7*(4.11), 55-63.

[14] Ishak, Z., Jaafar, A., & Ahmad, A. (2012). Interface design for cultural differences. *Procedia-Social and Behavioral Sciences*, *65*, 793-801.

[15] Bertolino, A. (2007, May). Software testing research: Achievements, challenges, dreams. In *Future of Software Engineering (FOSE'07)* (pp. 85-103). IEEE.

[16] Charlton, S. G., & O'Brien, T. G. (Eds.). (2019). *Handbook of human factors testing and evaluation*. CRC Press.

[17] Black, R. (2016). *Pragmatic software testing: Becoming an effective and efficient test professional*. John Wiley & Sons.

[18] Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.

[19] Runeson, P. (2006). A survey of unit testing practices. *IEEE software*, *23*(4), 22-29.

[20] Ammann, P., & Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.

[21] Rannikko, P. (2011). User-centered design in agile software development. *University of Tampere*.

[22] Schramm, V. B., Damasceno, A. C., & Schramm, F. (2023). Supporting the choice of the best-fit Agile model using Fitradeoff. *Pesquisa Operacional*, *43*(spe1), e264750.