APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 79-93
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Student Ride-Sharing System with Multi-Factor Authentication

Siti Nurhafiza Junaidih[1], Zubaile Abdullah[1]*

[1] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: zubaile@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.005

Article Info

Received: 19 July 2024
Accepted: 16 October 2024
Available online: 15 December 2024

Keywords

Ride-sharing, Multi-factor authenticatin, Ride-hailing

Abstract

The Student Ride-Sharing System with Multi-Factor Authentication project aims to revolutionize transportation for Universiti Tun Hussein Onn (UTHM) students by addressing the challenges they face in arranging affordable and secure rides. In response to the existing manual ride-hailing system, which lacks proper verification measures and centralized control, this project introduces a secure online platform designed exclusively for UTHM students. This system emphasizes security with multi-factor authentication that consist of strong password and email one-time password. It mitigates risks associated with unauthorized access, data breaches and fraud. This project uses Agile software development life cycle (SDLC) approach and hypertext preprocessor (PHP). By fusing improved security measures including one-time password for vefication and input validation with a user-friendly approach, the project offers an inviting solution that prioritizes the needs and safety of UTHM students.

1. Introduction

In the current student community at Universiti Tun Hussein Onn (UTHM) Parit Raja, a Telegram group chat has been established for students to interact with potential ride-hailing drivers with the goal of providing cost-effective transportation and earning options for students. Through a dedicated Telegram group chat, they initiate the process by posting a message outlining their specific transportation needs.

However, this approach has drawbacks. A major issue is the lack of proper identity verification, enabling non-students to exploit and overcharge peers. This is unfair and raises safety concerns due to the decentralized messaging system.

For starters, the existing system insufficient user verification and authentication, which allows non-students to penetrate the group. This not only compromises the system's integrity but also compromises critical data. The lack of user data security. Telegram offers both public chats and secret chats as distinct features, with users commonly favoring public chats due to their familiarity. The level of privacy and security provided by public chats is significantly lower compared to secret chats[1]. In public chats and regular chats, the messages are not end-to-end encrypted and they are stored on Telegram's servers, making them potentially more vulnerable to exposure or interception[1]. The lack of sufficient measures against harmful actions such as phishing poses a clear concern.

Lastly, inadequate evidence trails within the platform. Currently, chat and message histories may be subject to deletion by passengers or drivers through their Telegram accounts, resulting in the loss of valuable communication records. It is possible to modify or remove sent messages entirely from the chat history at any point by using Telegram[1]. This lacks a structured and easily accessible evidence trail for ride bookings and communication between students and drivers.

To address this challenge, the Student Ride-Sharing System is proposed. It prioritizes safeguarding sensitive user information like passwords and id. This signifies a strong commitment to improving security standards. A key measure is implementing a strict multi-factor authentication (MFA) system. Users must authenticate via their official student email, ensuring only authorized students access the system during login. This tight security protocol not only serves to shield sensitive user data but also produces a secure and trustworthy environment, instilling a profound sense of integrity in the platform[2].

This project holds significance in improving the system, focusing on its database functionality to ensure smooth operations with enhanced security features. For example, users can easily update their passwords for added account protection through password masking, keeping the password hidden from potential threats like shoulder surfing.

Moreover, the project serves as a valuable resource for peers, juniors and other interested readers in the field of web and application development. It provides useful information that can be a foundation for future enhancements, contributing to the academic knowledge in this domain.

Lastly, the project prioritizes user-friendliness. A simple and intuitive interface is crucial for effective user-system interaction, especially for new users. By combining improved security measures with an approachable user interface, the project aims to create a safer and user-friendly environment, ensuring that users can confidently utilize the platform for their transportation needs.

2. Literature Review

This section explains about the literature review of the project. It includes student ride-sharing, authentication, hashing algorithm and study of related systems.

2.1 Authentication

Authentication is a process of verifying the identity of a user, system, or other entity[3]. It verifies that people or organizations are who they say they are before allowing them to use specific resources, networks, or data. Credentials like usernames and passwords are frequently used for authentication. This project implements multi-factor authentication (MFA). Multi-Factor Authentication (MFA) takes the concept of 2FA further by requiring at least two types of identification. It can involve a combination of something you know, something you have and something you are[4]. MFA provides an additional layer of security compared to single factor methods. The goal is to make unauthorized access more challenging by requiring multiple independent means of verification[3]. This approach is particularly effective in protecting sensitive systems and data.

A one-time password (OTP) is a temporary and unique code used for authentication. An OTP's primary characteristic is its ability to change either immediately upon use or over time, providing an additional degree of security to the authentication process[1]. Email One-Time Password (OTP) is a method of authentication that also being implement in this project. This type of authentication allows a user to confirm their identity by sending a one-time, temporary code to themselves via email. The system creates a distinct OTP and sends it to the user's registered email address whenever the user starts a transaction or logs in. To finish the authentication process, the user has to enter this OTP within a given window of time.

2.2 Password

A password is a secret word or code that functions as a security measure to prevent any unauthorized access to the protected data[5]. The strong password describes a password that humans and computer program or bots find difficult to guess a it must consist of at least eight characters, have combination of uppercase and lowercase, a mixture of letters and numbers and have at least one special character[5]. It is important to protect the system from any brute force and password hacking technique.

2.3 Study of Related Systems

2.3.1 Manual Ride Hailing System (Telegram Group Chat)

Telegram is a messaging application [16] launched in 2013 with a focus on rapid communication, security and user confidentiality. Renowned for its innovative features like end-to-end encryption, self-erasing messages and cloud-based storage, Telegram enables users to send text, multimedia and participate in group conversations[6]. In the current manual ride-hailing service employed by students at Universiti Tun Hussein Onn (UTHM) Parit Raja through a Telegram group chat, various security vulnerabilities are apparent due to the absence of essential features. Notably, the service lacks a formal login procedure, registration mechanism and authentication protocols. Consequently, users initiate ride requests without undergoing any user verification, exposing the system to potential misuse by unauthorized individuals

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 79-93

2.3.2 MyCar

The second application studied is the MyCar Application[7]. MyCar provides users with a primary login option via phone number, utilizing SMS verification for initial registration. This simple authentication method ensures user convenience but does not enforce additional multi-factor authentication (MFA) layers beyond the initial phone number verification. The platform likely uses standard encryption protocols to secure transactions and protect user data, although specific details on these protocols are not publicly disclosed. MyCar encourages users to follow strong password practices where applicable. The security of this centralized system relies on basic encryption methods, initial user authentication and overall system security measures to safeguard user information and maintain the integrity of its services.

2.3.3 BlaBlaCar

The third application is BlaBlaCar Application[8]. BlaBlaCar uses email and password for login, focusing on strong passwords and secure authentication. During registration, users provide personal details and a valid email. BlaBlaCar likely confirms identities through email verification. Transaction security and password hashing align with industry standards, though specifics are undisclosed. BlaBlaCar encourages robust passwords, with details kept private. Security relies on encryption, user authentication and overall system measures.

Table 1 *Comparison between the Existing System and the Proposed System*

Feature	Manual Ride Hailing System[6]	MyCar[7]	BlaBlaCar[8]	Student Ride-Sharing System with Multi-Factor Authentication
System Platform	Messaging application (Telegram)	Application	Application	Web-based application
Login/Register for user	No	Yes	Yes	Yes
Multi-Factor Authentication (MFA)	No	No	Yes	Yes
Database Used	No	Not Stated	No	Yes
Request ride option	Yes	Yes	Yes	Yes
Ride feedback	No	Yes	Yes	Yes
Programming Language	No	Not Stated	Javascript	PHP

3. Methodology

This chapter provides a comprehensive overview of the chosen methodology, Agile, for the development of the student ride-sharing system. The methodology selection is a critical decision that significantly influences the system's evolution. The chapter details the fundamental principles of Agile, outlining its iterative and collaborative nature.

3.1 Agile Model

The Agile software development life cycle (SDLC) model, known for its iterative and incremental approach, prioritizes process flexibility and rapid delivery of functional software. Adopting the Agile approach, the project follows an incremental build strategy. As depicted in Figure 3.1, the AGILE model encompasses planning, requirement analysis, design, coding, unit testing and acceptance testing. Agile methodologies are chosen for their effectiveness in delivering high-quality products within budget constraints. This agile model promotes flexibility, adaptability and increased productivity through visualizing data, enhancing the team's capacity for productive work.

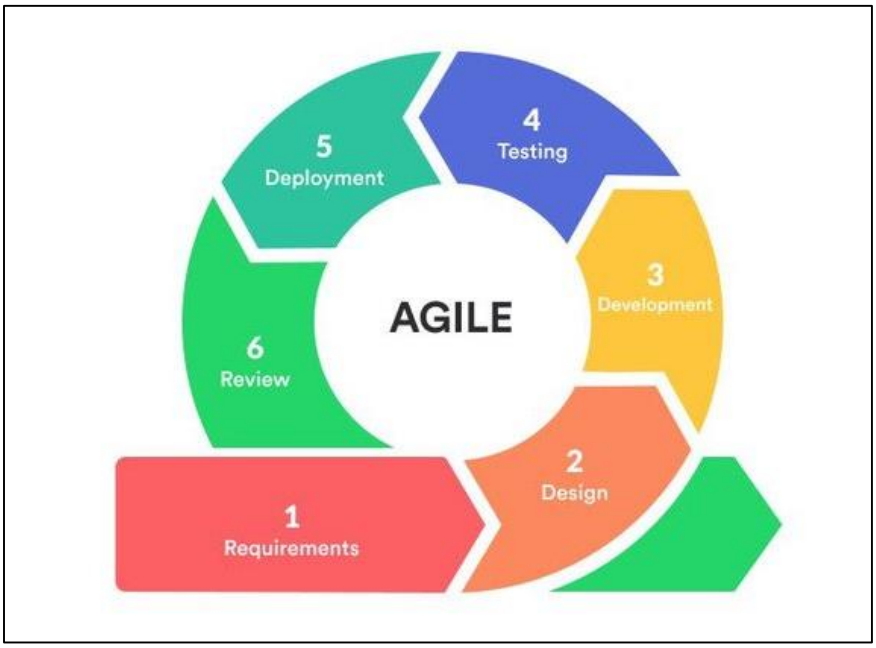

Fig. 1 *Agile Model*

3.1.1 Requirement Analysis Phase

In this phase, the process includes collecting and analyzing specific detail that are required for the proposed system. This system aims to only allow student to use the system by using their verified student email accounts. The goal is to design, develop and test an online complaint management solution with added security features, including multi-factor authentication. Ensuring the security of the proposed system is crucial. Multi-factor authentication is recommended to provide an extra layer of security, protecting student's accounts and sensitive data from unauthorized access. The system should enable students to make reports efficiently and conveniently.

3.1.2 Design Phase

Next, in the system design phase, the focus is on understanding the targeted user of the system which is the students of Universiti Tun Hussein Onn Malaysia. The students will use their student email to registered and log into the system to use the develop system. Here, the security measures are prioritized early in the development of Student Ride-Sharing System. Multi-factor authentication, utilizing one-time password sent via email, is proposed along with other security features includes password hashing. These measures are critical to protecting sensitive data and ensuring the security of user information, aligning with the system's commitment to data security.

3.1.3 Development Phase

The development phase centers on translating the defined requirements and user needs into an actual system. Using the Agile methodology, this phase promotes continuous improvement and adaptation. The development process prioritizes the implementation of planned features and functionalities while adhering to coding standards. Key security measures, including One-Time Password via email for multi-factor authentication, password hashing, input validation and input sanitization, are integrated. guarantee that the system operates smoothly and remains protected against potential cyber threats, external attacks and data breaches. User feedback is continually reviewed to support ongoing improvements and enhancements to the system.

3.1.4 Testing Phase

A comprehensive and thorough strategy is used in the testing phase of the Agile development model for the Student Ride-Sharing System to guarantee the system's usability, security and functionality. The ideals of Agile development are in line with this process. Through acceptance testing, the system's compliance with the requirements is verified. Security testing, especially penetration testing, is essential because of the sensitive nature of data. Usability testing evaluates the overall experience and user interface, taking user comments into account for continuous development.

3.1.5 Deployment Phase

In order to provide students with a safe and dependable platform for student to log into the system using their student emails, the deployment phase is an essential step. Enabling students to access the fully completed system, which has multi-factor authentication and other upgraded security measures, is the main objective of this phase. In addition, a verification process is part of the deployment phase, which makes sure the system meets all applicable requirements and protects user privacy and data in compliance with security and data protection laws. In order to give students a reliable and safe platform on which to turn in their reports, this stage is essential.

3.1.6 Review Phase

Review phase is the last phase of Agile methodology. This phase ensures that each component aligns with project goals and expectations. There may be problems or errors that need to be addressed, replicated and fixed by the maintenance teams after the system has been put into use and students have actively been using it to book available driver. The maintenance phase provides opportunity to add and update features as user needs and college requirements change. This involves expanding the system's capability to meet shifting demands through constant improvement. Maintaining data integrity, controlling data growth and performing inquiries are all critical aspects of regular database management.

4. Analysis and Design

This section explains the design of the system which consist of requirement analysis, entity diagram relationship and interface design.

4.1 Requirement Analysis

Requirement analysis involves identifying the user's expectations for the application, whether it is being developed or modified. The analysis results will be illustrated in the Unified Modelling Language (UML), including the use of use case diagrams, sequence diagrams and activity diagrams. Table 2 and Table 3 shows the functional and non-functional requirement.

Table 2 *Functional Requirement*

Module	Description	Entity
Registration	Allow user to create an account within the system.	passenger
Login	Allow user to access into the system	admin, driver, passenger
User Profile	Allow user to manage their account and setting preference.	driver, passenger, admin
Manage User	Allow user to manage and approve the registered user.	admin
Book Driver	Allow user to book a ride of available driver.	passenger
Confirm Booking	Allow user to confirm booking.	driver, passenger
Booking List	Allow user to view their ride activity such as pending, confirm, pick up, drop off or cancelled.	admin, driver, passenger
Driver Rating	Allow user to give rating of the driver.	passenger
Logout	Allow user to sign out of the system.	admin, driver, passenger

Table 3 *Non-functional Requirement*

Module	Description
Performance	The system should have a response time of no more than 3 seconds for user interactions and feedback submissions.
Security	The system should undergo regular security audits and comply with industry best practices to ensure a high level of security.
Reliability	The system must be available anytime to ensure continuous access for users
Usability	The system interface must be user-friendly to build a good interaction between user and the system

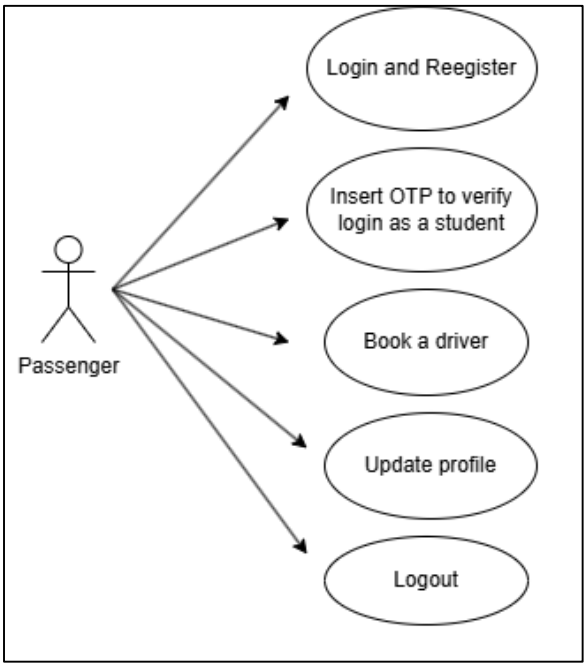

Fig. 2 *Passenger use case*

The use case diagram in Figure 2 shows how a passenger interacts with a system to perform various actions. The passenger can "Login and Register," verify their login with an OTP as a student, "Book a driver," "Update profile," and "Logout."

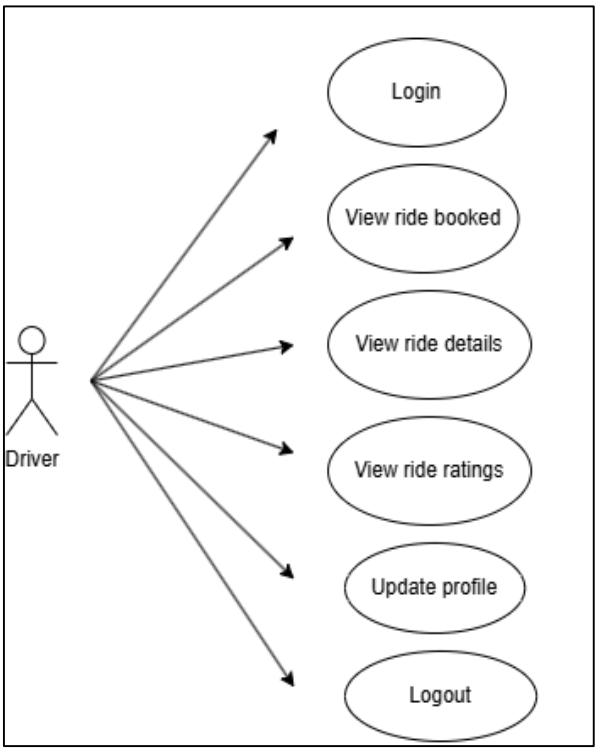

Fig. 3 *Driver use case*

Driver use case diagram as shown in Figure 3, depicts the different actions. Driver can perform within a system. The driver can "Login," "View ride booked," "View ride details," "View ride ratings," "Update profile," and "Logout."

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 79-93

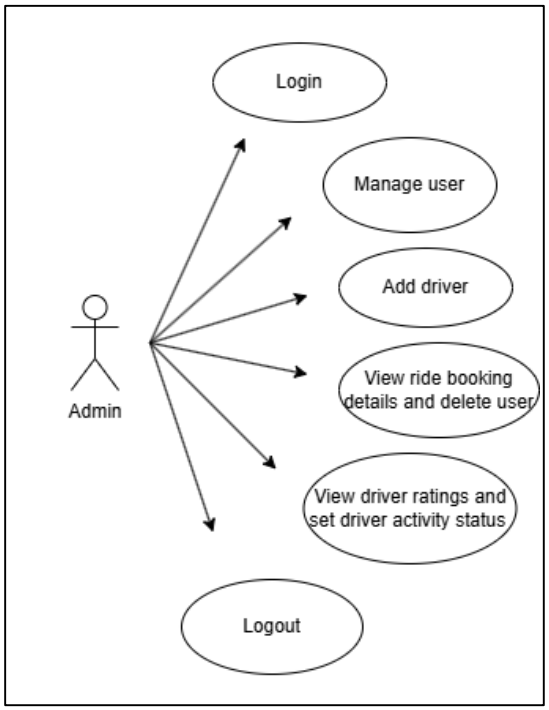

Fig. 4 *Admin use case*

Figure 4 shows admin use case diagram. Admin are able to perform several key actions. These actions include "Login," "Manage user," "Add driver," "View ride booking details and delete user," "View driver ratings and set driver activity status," and "Logout.

4.1.1 System Architecture Design

System architecture is a conceptual model that outlines the system's behavior to achieve its objectives. Figure 5 illustrates the system architecture design for the proposed system. This system employs a web-based user interface enabling passengers, drivers and admins to manage and access various modules. These modules include login, registration, driver booking, user management, user approval and deletion, viewing booking details and viewing driver ratings.

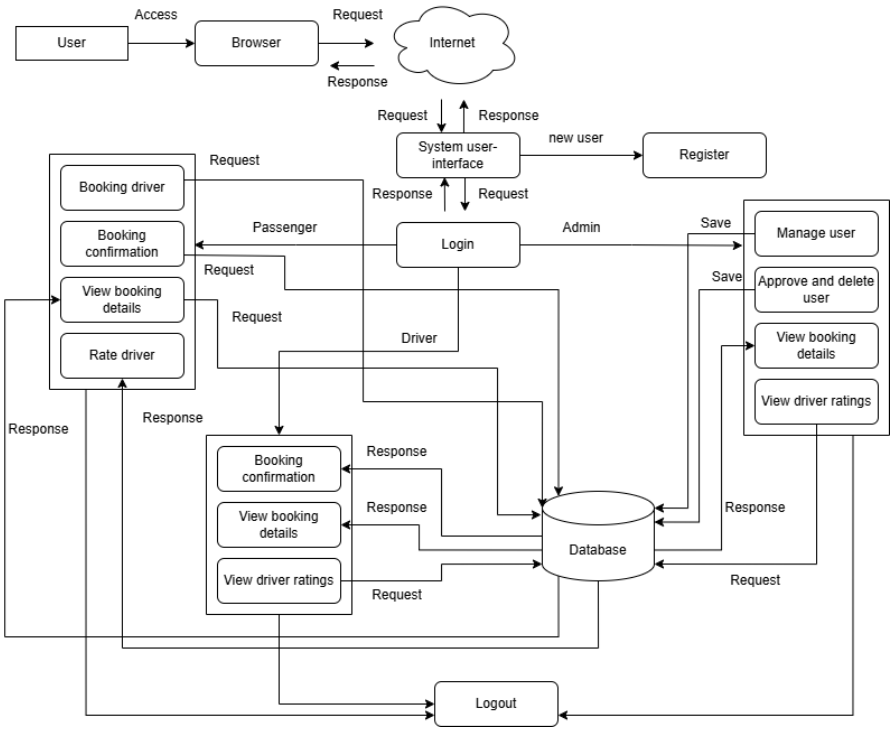

Fig. 5 *System architecture design*

4.1.2 Flowchart

Flowchart is a simple algorithm that represents different with different function that connect by arrow. The combination of the shapes shows the flow and process of the system. Figure 6 shows the flowchart for the purposed system that will explains all processes involving passenger, driver and admin.

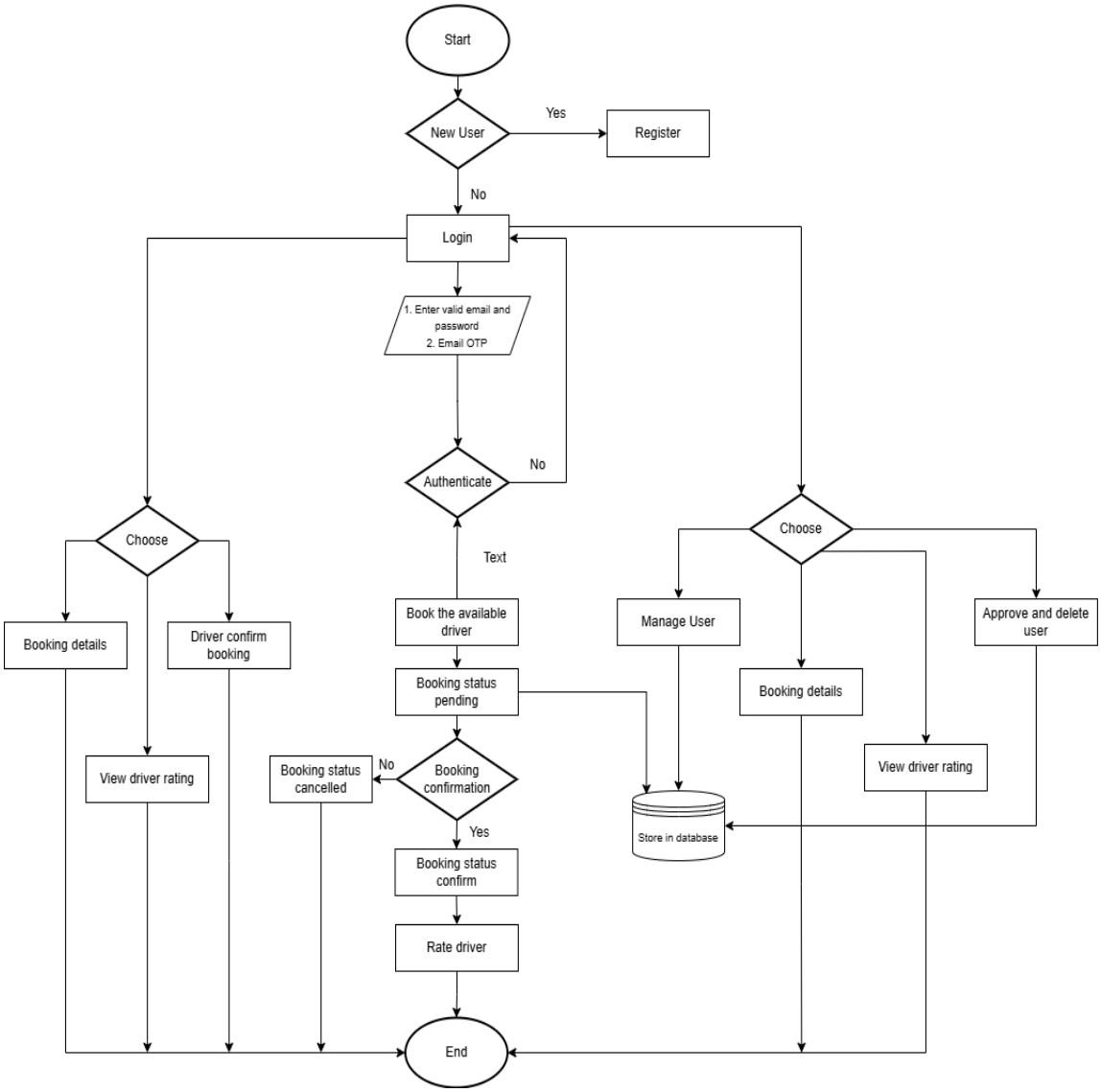

Fig. 6 *Flowchart of the purposed system*

4.2 Entity Relationship Diagram

In the design of databases, the Entity Relationship Diagram (ERD) is employed. The ERD, alternatively referred to as the ER Diagram or ER model, is a type of structural diagram utilized in database design. It encompasses entities, attributes and illustrates the inter-relationships among entity sets stored in a database. The system entities include Passenger, Driver, Booking and Category.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 79-93

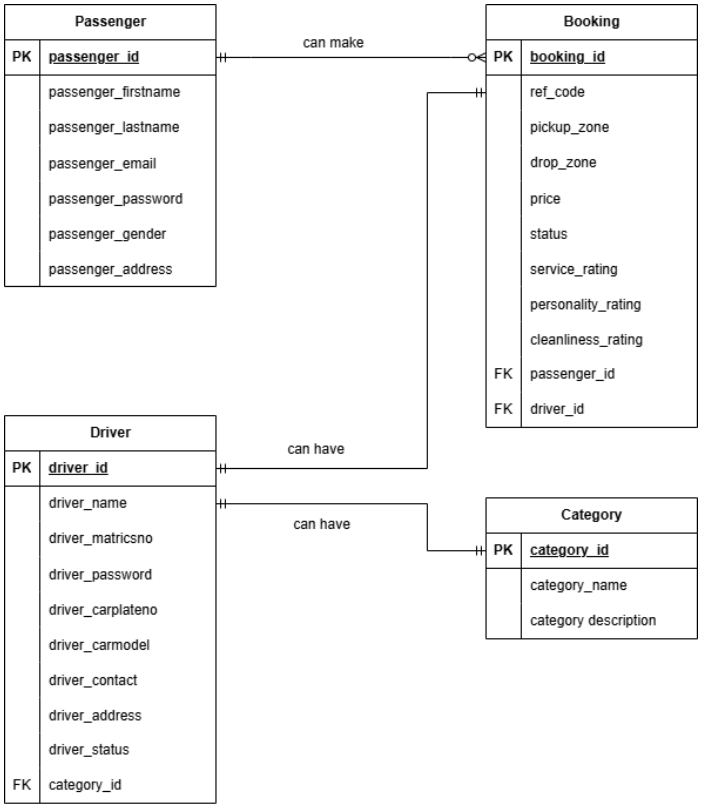

Fig. 7 *Entity Relationship Diagram*

4.3 Interface Design

The interface design of student ride-sharing system application is illustrated by using storyboard. There will be three part of the interface design which is passenger interface, driver interface and admin interface.

4.3.1 Registration and Login Page

Figure 8 shows the login page where the user decided to login as passenger to login to the system. They will need to fill the login form with their email and password. A one-time password will be sent to their student email address to authenticate the passenger as a valid student.

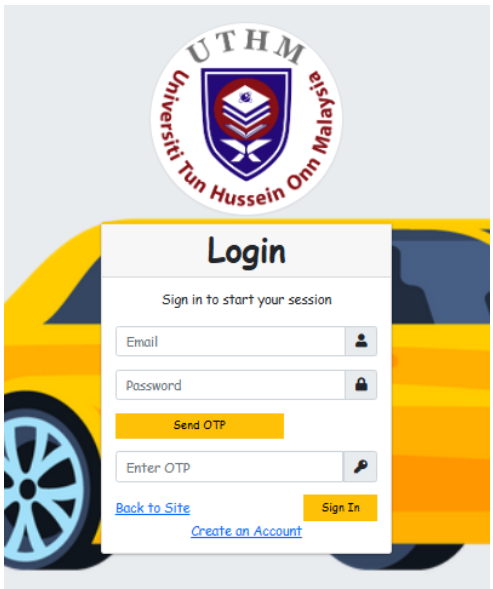

Fig. 8 *Passenger Login Page*

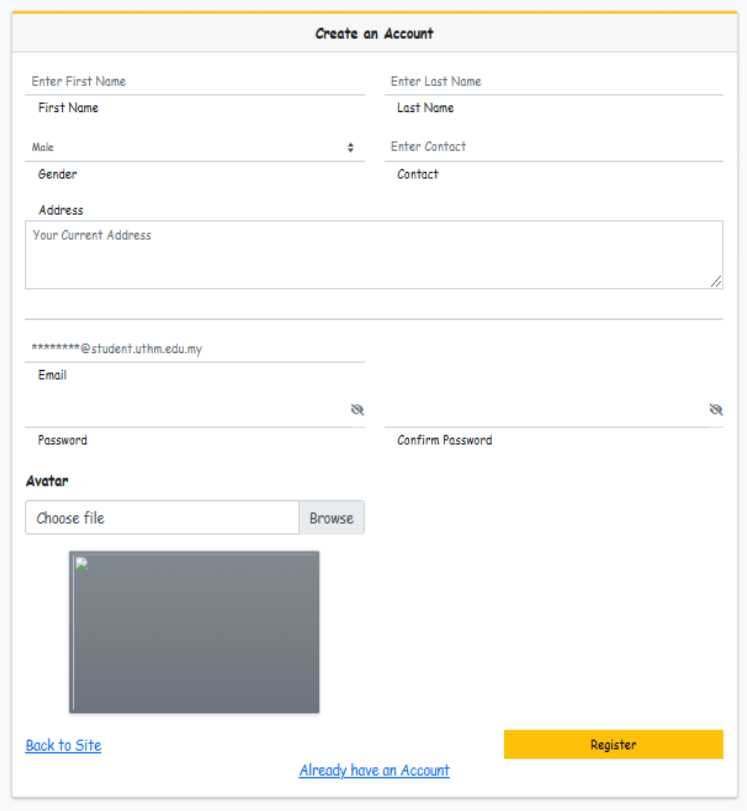

Fig. 9 *Register Page*

If there is any new user who does not have an account yet, they can click the create new account hyperlink under the login form. They will need to fill the registration form with their first name, last name, email (student email), gender, contact number and alphanumeric password. If the input is correct, then passenger can proceed to login to the system as shown in the Figure9 below.

4.3.2 Booking Driver Page

Figure 10 shows the Driver available page where passenger can view the available driver to book a ride. It also shows the rating of each driver to help with the good selection of driver. Once the passenger clicks on the selected driver, they need to input the booking details the booking form that includes pickup zone and drop off zone for driver to view the booking details as shown in Figure 11.

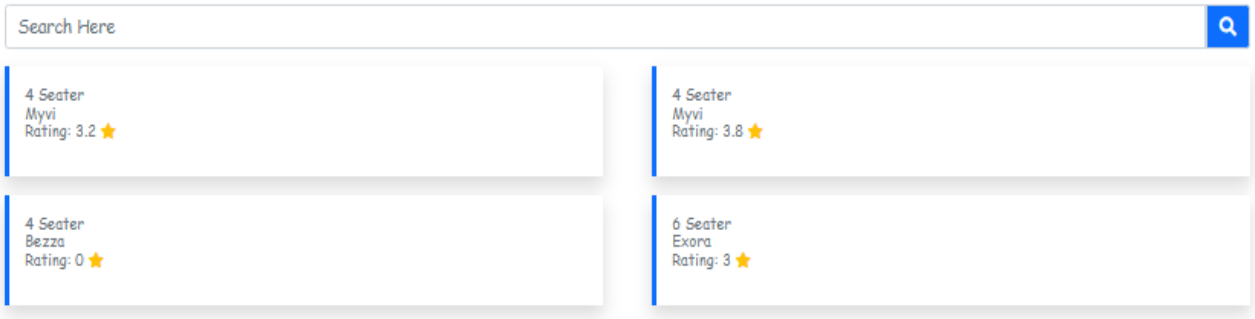

Fig. 10 *Driver Available Page*

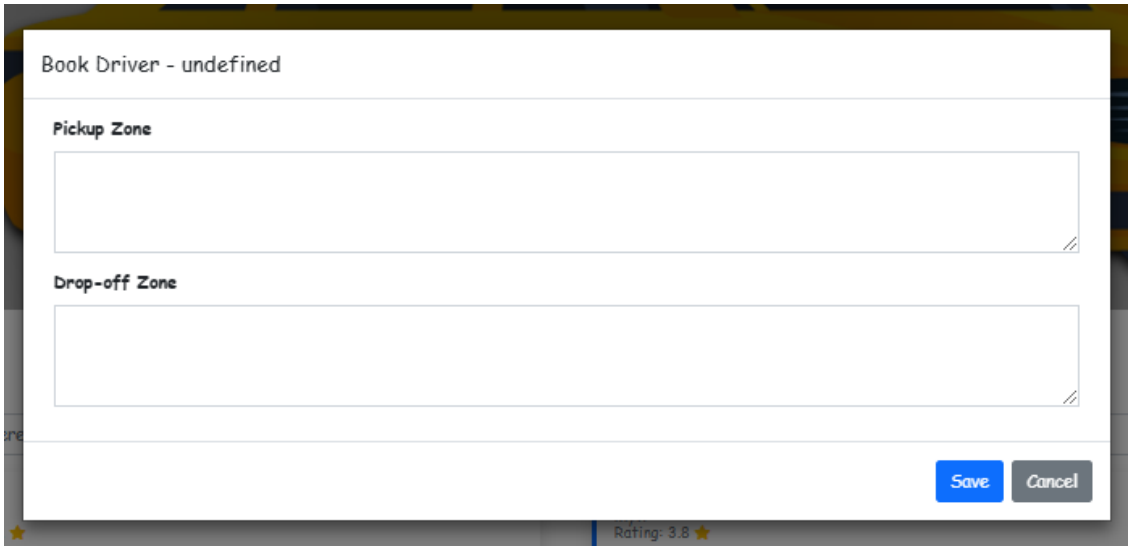

Fig. 11 *Book Driver Form*

4.3.3 Booking List Page

Figure 12 shows the booking list where the status for each booking and booking history are stated. This can be view by the passenger, driver and admin.

My Booking List

Show 10 entries Search:

#	Date Booked	Ref Code	Details	Status	Actions
1	2024-06-07 06:48	202406-00001	Pickup: library Dropoff: fsktm	Dropped off	View
2	2024-06-07 06:45	202406-00001	Pickup: mulot Dropoff: library	Cancelled	View
3	2024-06-07 03:46	202406-00001	Pickup: library Dropoff: fsktm	Dropped off	View
4	2024-06-06 11:37	202406-00001	Pickup: mulot Dropoff: library	Dropped off	View
5	2024-06-01 23:31	202406-00001	Pickup: mulot Dropoff: tdi	Dropped off	View
6	2024-06-01 23:30	202406-00001	Pickup: mulot Dropoff: library	Dropped off	View
7	2024-06-01 22:11	202406-00001	Pickup: library uhtm Dropoff: dewan f2	Cancelled	View
8	2024-06-01 22:09	202406-00001	Pickup: mulot Dropoff: tdi	Dropped off	View
9	2024-05-30 12:39	202405-00001	Pickup: Mulot Dropoff: Library	Dropped off	View
10	2024-05-28 16:06	202405-00001	Pickup: mulot Dropoff: library	Dropped off	View

Showing 1 to 10 of 10 entries Previous 1 Next

Fig. 12 *Booking List Page*

4.3.4 Update Profile Page

The figure 13 shows update profile page where user can view and manage their account. They are requiring to fill the current password to update their details.

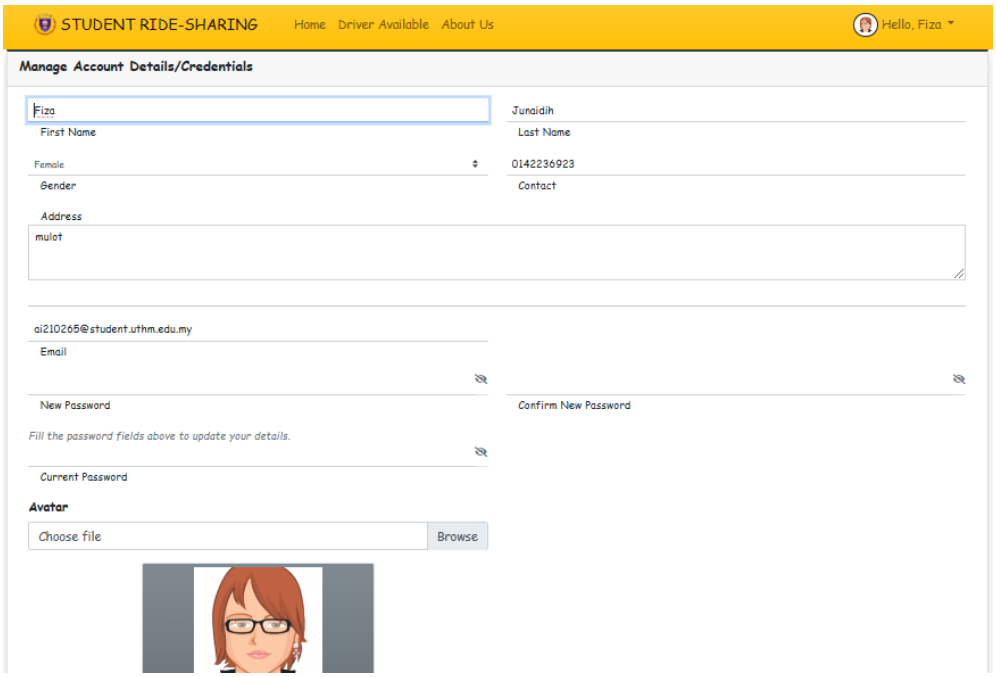

Fig. 13 *Update Profile Page*

4.3.5 Create New Driver Page

This module can only access by admin. Admin is able to create the new driver and able to manage active status driver. The related details will appear in the passenger's interface during booking available driver.

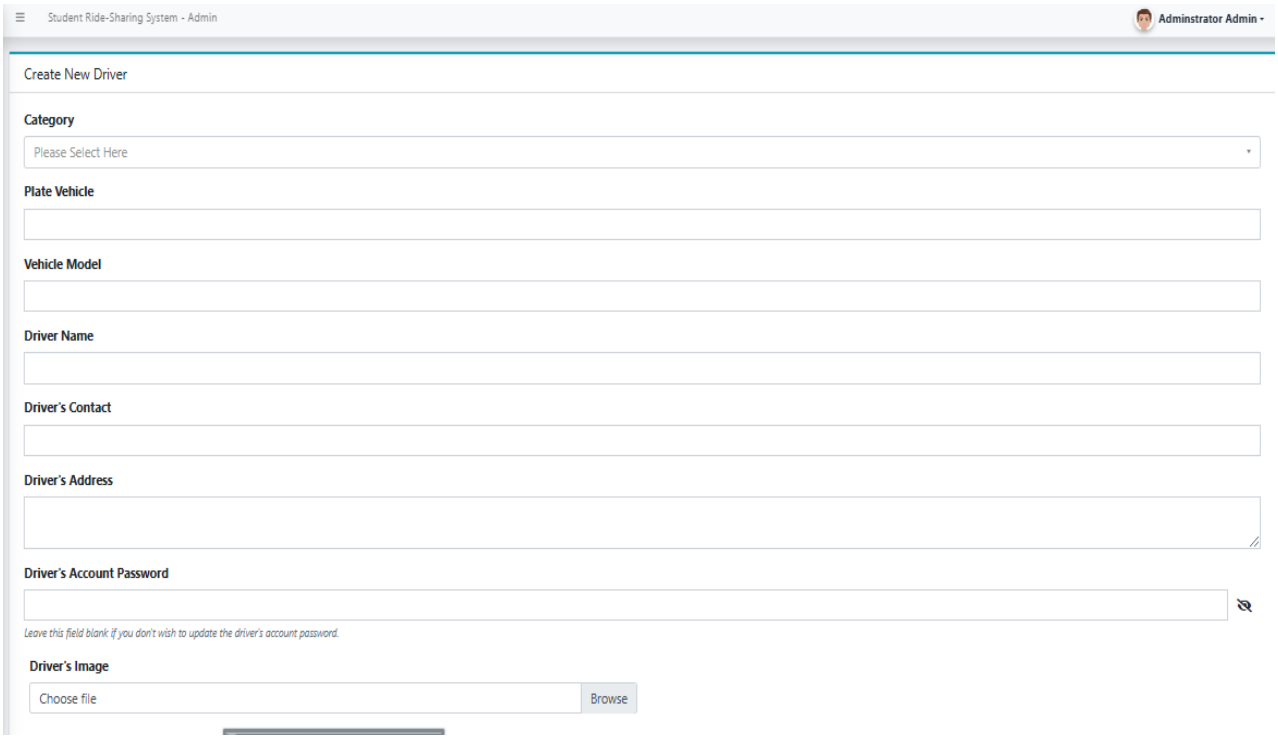

Fig. 14 *Create New Drive Page*

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 79-93

4.4 System Testing

After the system has been completely developed, the system undergoes the testing phase to test the functionality of the system. It is used to identify if there are any bugs and errors that occurred during the testing phase. In the meantime, the scope and objectives of this purpose system can be achieved by improving the errors. Table 4 shows the summary of the functional testing results.

Table 4 *Functional Test*

No	Function Testing	Expected Result	Result
	Register Function		
i.	Users leave fields empty	Error message appear	Success
ii.	Users enter the wrong input	Error message shown	Success
iii.	Users enter the same email	Error message 'Registered email' will show	Success
iv.	Users enter all correct input	User can proceed to login	Success
2.	Login Function		
i.	Users leave field empty	Error message shown	Success
ii.	Users enter wrong input	Error message shown	Success
iii.	User not input OTP code	Error message shown	Success
3.	Booking driver		
i.	Passenger choose available driver	User will direct to booking form	Success
ii.	Passenger cancel booking	Booking status change to cancelled	Success
4.	Add driver function		
i.	Only admin can add new driver	The data will store in database	Success
5.	Manage user		
i.	Only admin can manage passenger and driver active status	The data will store in database	Success
6.	Update profile		
i.	Passenger and driver can update their profile	The updated profile stored in database	Success

4.5 User Testing

User testing was conducted to ensure that the system meets the requirements and criteria set by the target users. A total of 14 users, consisting of 5 women and 9 men, were selected as respondents to answer a comprehensive questionnaire. The questions included the ease of registration, logging in, booking rides and overall satisfaction with the system's security and usability. The results indicated a 92.9% satisfaction rate among users, with all participants appreciating the multi-factor authentication feature. Additional questions assessed the interface design, perceived security of personal data and any suggestions for improvement. This thorough user testing highlighted the system's effectiveness in providing a secure and user-friendly platform for UTHM students.

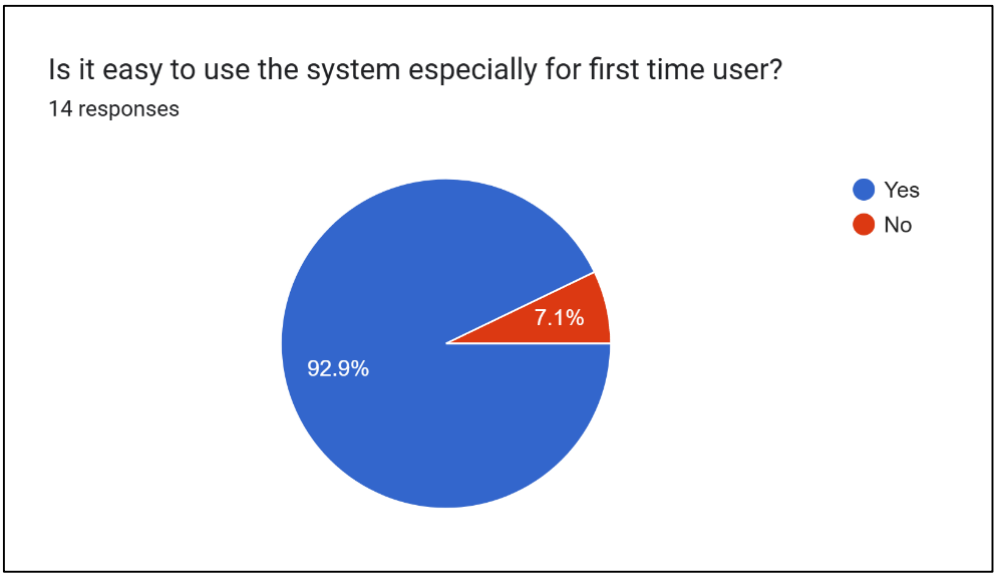

Fig. 15 *Percentage for user acceptance*

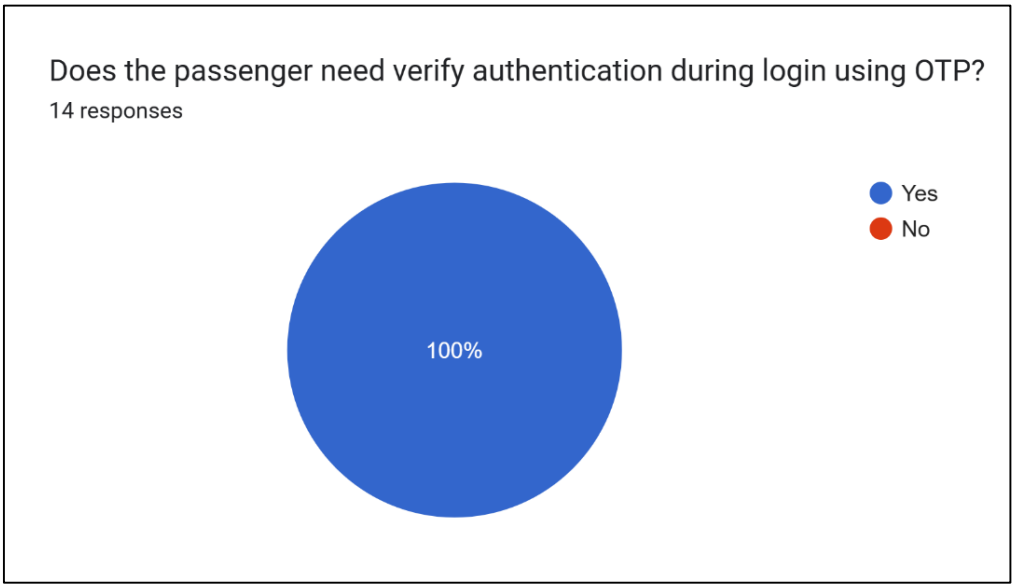

Fig. 16 *Percentage for user acceptance*

Figure 15 shows the percentage of user satisfaction with the purposed system. Based on the survey that has been done, about 92.9% of the respondent choose yes and 7.1% of the respondent choose no that the system is easy to use for first time user. Figure 16 shows the percentage of user satisfaction for the system. 100% respondents choose yes that the passenger need to verify authentication during login.

5. Conclusion

The Student Ride-Sharing System with Multi-Factor Authentication in UTHM successfully addresses the security and usability issues present in existing ride-sharing platforms. The system's robust security measures, including multi-factor authentication and password hashing, ensure that only authorized students can access the platform. Future work could explore integrating biometric authentication and real-time tracking to further enhance security and user experience. Additionally, the system can be adapted for use in other educational institutions or community settings, providing a scalable solution for secure ride-sharing. Limitations of the current study include a small sample size for user testing and the lack of real-world deployment data, which will be addressed in future research.

Acknowledgement

The author would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for the support.

References

[1] L. Thomas and S. Bhat, "A Comprehensive Overview of Telegram Services - A Case Study," *International Journal of Case Studies in Business, IT, and Education*, vol. 6, no. 1, pp. 288–301, 2022, doi: 10.47992/ijcsbe.2581.6942.0165.

[2] F. A. Aloul and W. El-hajj, "Multi Factor Authentication Using Mobile Phones Introduction," no. May, 2014.

[3] V. Papaspirou *et al.*, "Security Revisited: Honeytokens meet Google Authenticator," *7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2022*, pp. 1–8, 2022, doi: 10.1109/SEEDA-CECNSM57760.2022.9932907.

[4] *Hawaii International Conference on System Sciences 2020.*

[5] T. Tanni, T. Taharat, M. Parvez, S. Rumee, and M. Zaber, "Is My Password Strong Enough?: A Study on User Perception in The Developing World," *EAI Endorsed Transactions on Creative Technologies*, vol. 9, no. 30, p. 173452, Mar. 2022, doi: 10.4108/eai.11-2-2022.173452.

[6] "Telegram Apps." Accessed: Nov. 22, 2023. [Online]. Available: https://web.telegram.org/a/

[7] "MyCar Apps." Accessed: Jun. 10, 2024. [Online]. Available: https://www.mycarasia.com/site/

[8] "BlaBla Car." Accessed: Nov. 22, 2023. [Online]. Available: https://www.blablacar.co.uk/