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

AITCS

Vol. 7 No. 1 (2026) 155-175
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Development of a Driving School Simulation: Driving Success

Muhammad Shahrul Ikhwan Abdul Halim[1], Mohd Farhan Md.Fudzee[1]*,

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: farhan@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.009

Article Info

Received: 13 June 2025
Accepted: 14 June 2026
Available online: 30 June 2026

Keywords

Application, Agile, Virtual Reality (VR), Driving, Driving School, Target user, Methodology, Technology.

Abstract

Education is a very important subject and can be delivered well through the right methods. This study examines the difficulties associated with the conventional physical driving school system in Malaysia, especially focusing on the high expenses associated with vehicle maintenance. Driving learning centers suffer from a shortage of circuits which affects students' learning sessions. Therefore, the main objective is to design a virtual driving school simulation application for driving school students that focuses on theory and circuit tests, featuring modules specifically designed to align with Malaysian standards, focusing on individuals preparing to take the driving license examination. The application was developed using Agile methodology, integrating Virtual Reality (VR) to create an immersive experience. Virtual driving school simulation using VR technology and appropriate hardware has the potential to increase accessibility and improve user learning outcomes. Future work can expand the scope of the application to include mobile platforms and involve advanced technologies for personalized training and feedback. The project demonstrated excellent usability and high user satisfaction through calculating an average System Usability Scale (SUS) score of 85.50 percent from respondent user acceptance score.

1. Introduction

The traditional driving school system in Malaysia offers a physical learning process that require individuals to attend a Driving Training Centre to complete both theoretical and practical courses. Before they are eligible to take the driving license exam, prospective drivers must complete a series of lessons, which may include a theoretical course and driving lessons that last between six and 16 hours. During these training sessions, driving instructors provide practical instruction to guarantee that students acquire the fundamental skills required to drive a vehicle safely on the road.

Advanced technology has facilitated and innovated more effective learning methods. Among the learning technologies that have been innovated are online learning, E-books and technologies such as applications with virtual reality (VR) integration that are easier to understand. Driving learning can also be innovated by creating applications that can help provide more effective experiences and understanding, for example the development of driving school simulation application. The integration of Virtual Reality (VR) technology in applications will allow users to experience virtual driving scenarios without the need for a physical vehicle. This immersive virtual environment will help users improve their driving skills and support their confidence before using a physical vehicle, thereby minimizing the possibility of accidents or mistakes during real-world training. The development of Driving School Simulation application offers a transformative solution to the challenges faced by conventional driving schools in Malaysia. The physical driving learning system does not offer satisfactory

practical sessions due to certain factors such as the large number of students that can affect all students to have the opportunity to practice on the circuit. Driving training centers have to pay for the repair of damaged cars, but some cannot be due to lack of capital [1]. In addition, this is also due to the lack of sufficient circuits for driving schools to implement training and assessment, including road damage [2]. Some Driving Training Centres are unable to cover the cost of repairing vehicles damaged by inexperienced drivers due to lack of capital [2].

Therefore, the objective of this project initiative is to develop a virtual driving school simulation application for driving school students focusing on theory and circuit tests. This initiative also offers driving training and experience as an initial preparation for taking the driving test. This application is designed for users between the ages of 16 and 25 who participate in driving school sessions. This application contains modules tailored to driving training standards. The challenges of answering the theory course related to road rules, circuit driving training, and circuit driving test are the third module. The academic material and the test for the theory course will be obtained from the driver education course test (KPP). The driving training module includes the rules that must be successfully completed on the driving circuit, such as parking checkpoints. Guidance and learning activities are also included with this module. For the driving test module, each checkpoint score will be calculated without any guidance. Users can obtain a driving license easily and successfully by engaging in as much training and learning as possible. This application is created according to the Agile methodology. This methodology guarantees the success of the application by adhering to the phases that have been divided. It can guarantee that the application runs without any errors by following the methodological phases. Furthermore, the efficiency of the application is enhanced by its compatibility with virtual reality (VR) devices. Accordingly, this application has the potential to improve the learning process by allowing students to practice a sufficient amount to foster their driving confidence.

The rest of the paper is arranged as follows: Section 2 covers the domain of study, the technology used, and the result of the comparative analysis. Section 3 describes the Agile methodology that has been chosen to be used in this project, as well as the output of the analysis and design phases of this project. Furthermore, Section 4 discusses the result and discussion, while Section 5 states the conclusion of the current progress.

2. Related Work

This section discusses the background of the study, the technology used, and the result of the comparative analysis.

2.1 Driving Education

In Malaysia, driving education is an important component of road safety initiatives. Driving without a valid license is illegal in most countries [3]. However, this is the approach by which driving training centers were established. It usually works through physical training sessions at driving schools. Students are usually required to practice on real vehicles while under the guidance of a certified instructor during these sessions. Furthermore, the scope of physical training is limited as students primarily practice in a controlled environment that may not fully prepare them for real-world driving challenges. Traditional driving schools provide a series of assessments, practical on-road training, and classroom theory lessons to ensure students have the knowledge and skills required for responsible driving. Driving schools in Malaysia have great opportunities to innovate and improve their operations, regardless of the challenges they face. In addition, driving schools contribute to the mitigation of road accidents that are the result of distracted drivers. The learning modules of driving schools serve as the basis of safe driving practices.

2.2 Virtual Reality (VR)

Virtual Reality (VR) is an artificial environment that exists in images but not in real life. It is described as real-time interactive graphics with three-dimensional models, combined with display technology that gives the user a sense of immersion in the model world and allows direct manipulation with objects within it [4]. There are various types of virtual reality, such as fully immersive, semi-immersive and non-immersive. There are games that have started to use this technology such as F1 2025 VR to provide good effects and experiences to users [5]. Virtual Reality (VR) has the advantage of simulating the environment into the form of images in the application. This can provide an interesting experience for users who use the application. Most driving games have used this technology, namely Virtual Reality, either semi-immersive or fully immersive. Semi-immersive VR gives users a partial virtual experience to interact, this is especially suitable for educational and training purposes. Meanwhile, fully immersive is a digital technology that allows users to experience the artificial environment as the real world. For example, fully immersive is the game Beat Saber which uses VR headset technology with a controller to interact in the game which produces an immersive experience. The most used Virtual Reality hardware is Oculus, which is suitable for use on PC platforms.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 155-175

2.3 Comparative analysis

There are numerous sorts of simulations used in industry and academia, including flying simulations, driving simulations, and shipping simulations [1]. A comparative analysis was conducted on three applications related to the proposed application Driving School Simulation: Driving Success. The three applications are Gran Turismo 7 [6], City Car Driving [7], and Driving School Simulator [8], the respective interfaces are as shown in Figure 1(a), (b), and (c). Table 1 shows the results of the comparative analysis.

The comparison between the proposed application and the three reviewed applications will be evaluated in Table 1 below. The three applications that were compared are Gran Turismo 7, City Car Driving, and Driving School Simulator. The proposed application is called Driving School Simulation. Comparisons are made between the applications to determine the technology implemented. The comparison will evaluate a total of seven features, including platform compatibility, driving realism, educational value, technology integration, customisability, target audience, and regional customisation.

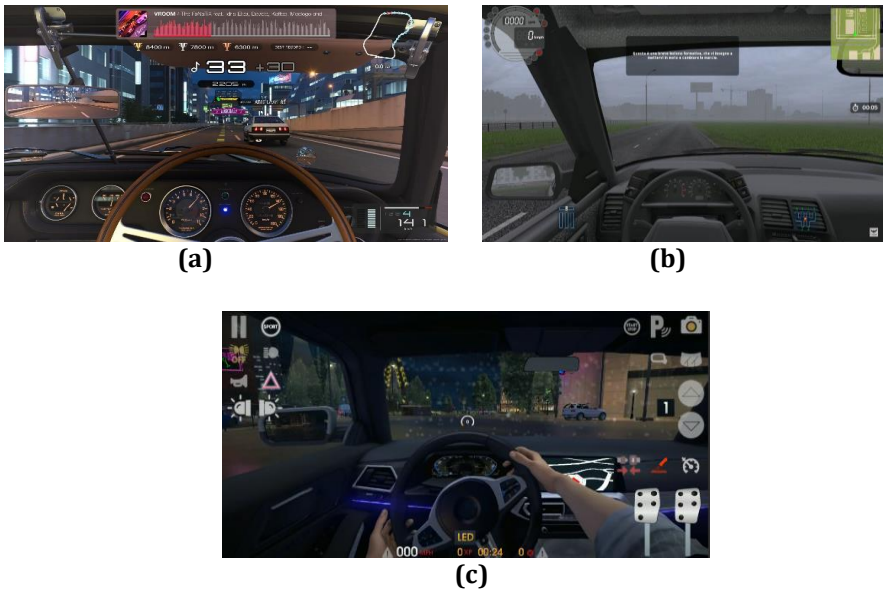


Fig. 1 *(a) Gran Turismo 7 [6]; (b) City Car Driving [7]; (c) Driving School Simulator [8]*

Table 1 *Comparison between existing application and proposed application*

Features/ Applications	Gran Turismo 7	City Car Driving	Driving School Simulator	Driving School Simulation: Driving Success
Platform Compatibility	PlayStation	Windows PC	Mobile (Android/IOS)	Windows PC
Driving Realism	High (Racing-focused, detailed physics)	High (Real-world scenarios, advanced physics)	Moderate (Mobile limitations, basic physics)	Moderate (VR integration)
Education Value	Low (Racing mechanics, no driving tutorials)	Moderate (Focus on driving skills)	Moderate (Traffic rules, driving scenarios)	High (Theoretical + Practical Driving Tutorials)

Table 1: *Comparison between existing application and proposed application (continue)*

Technology Integration	Advanced Graphics, Steering Wheel Support	VR Support, Steering Controls	Limited to Touch/Basic Control	Semi- Immersive VR, High-Quality Graphics
Customizability	Vehicle Tuning Only	Weather, Traffic, Difficulty Levels	Basic Scenario Selection	Module, Challenge
Target Audience	Racing Enthusiasts	Aspiring Drivers, Driving Enthusiasts	Beginner Drivers	Beginner Drivers
Regional Customization	None	Minimal	Minimal	Fully Localized (Malaysian Road Rules)

3. Methodology

The methodology used for the Driving School Simulation: Driving Success application project is the Agile methodology. This methodology serves as a guideline for the development of good quality applications, with each phase organized. The emergence of agile methodologies allows for an increase in the success rate of application development projects and a reduction in the budget of projects that are ultimately abandoned [9]. There are many opportunities to incorporate new findings into project versions, such as agile iterations combining testing, analysis, and review [10]. In addition, agile project management facilitates the integration of user feedback throughout the development process. Among the stages are requirements, design, development, testing, deployment, and finally the review stage. Figure 2 shows the cycle of these phases.

Fig. 2 *Agile methodology [7]*

3.1 Requirement Stage

The initial process of developing an initial idea of an application is carried out during the requirements stage. This process begins with the application emphasizing the needs and objectives of the end users by documenting the needs as user requirements [11]. The requirements stage produces a simple concept and description of the application. In addition, this stage involves concept analysis, which includes idea generation, concept definition, feasibility study, market research, and key feature definition. All information from Subject Matter Experts (SMEs) was collected. An interview session was conducted with SME En Ahmad Nata bin Salim, a driving school instructor at the Penggaram Driving Training Academy and has 24 years of experience in the field of driving training. As a result, Table 2 depicts an overview of the concept analysis and table 3 the user analysis during the requirements phase.

Table 2 *Concepts analysis at the initiation stage*

Item	Description
Idea generation and concept definition	Develop a virtual reality driving school simulation.
Market research	Analyze comparable games and applications to identify trends, target audiences and features.
Feasibility study	Determine the feasibility of VR platform, development tools and languages.
Target Audience	Driving school students
Key features definition	Core mechanics, immersive features delivered learning and accessibility.

Table 3 *User analysis*

Stakeholder category	Role in product	Design implication	Action needed
Subject Matter Expert (SME) Mr Ahmad Nata bin Salim	Content expert in driving school subject	Content, based on driving school student syllabus	The first module quiz questions must be followed from the driver education course (KPP) syllabus
		Multimedia content	put audio (voice over), suitable images to guide the user
		User friendly interface	Use the common icon and navigation based on user experience.
VR Users	Experience practical driving scenarios in VR	Semi-immersive Interface	Comfort settings for VR, test across VR headsets
Learner Drivers	End-users	Language that is understandable, interesting	Develop easy understanding UI/UX, including practice and test mode

3.2 Design Stage

The next stage is the design phase, rapid prototyping techniques, such as low-fidelity design or wireframes, assist in the initial validation of ideas with stakeholders and end users before development [12]. Application references were identified to guide planning, ensuring project scope, resources and workflow were clearly defined. Application functionality was finalized, covering learning modules on road rules, circuit driving and road scenarios. Hardware and VR platform requirements were also reviewed, along with a level progression system. Basic prototypes were developed for core mechanics such as vehicle control, with driving physics and a feedback system featuring instructions and performance scoring. Technical specifications were established, with Unity selected for VR development and Blender to create 3D assets. Initial concept art was created for vehicles, obstacle courses such as parking lots, hills and road environments. This phase concluded with presenting these designs and prototypes to stakeholders for feedback and approval before moving on to development. Functional and non-functional requirements are shown in Table 4 and Table 5 respectively. The flowchart, content structure, and navigation structure design can be seen in the attached appendix A, B, and C. Table 6 will schedule the button design while appendix D will showcase the interface design. The 3D models of application are shown in Table 7.

Table 4 *Functional Requirements*

Functional Requirement	Module	Description
User interaction	Theory module quiz	The system shall provide user selects and initiate a quiz based on the KPP syllabus.
	All module	The system shall provide user uses buttons or menu to switch between modules Theory, Practice, and Test.
	Practical module	The system shall provide user begins practical driving using VR setup.
Autonomous System Activities	Level unlocking	The system shall unlock levels based on user performance module.
	Performance scoring	The system shall evaluate theory and driving performance to determine pass and fail result.
	Temporary data storage	The system shall temporarily store session data and user input during session.

Table 5 *Non-Functional Requirements*

Non-Functional Requirement	Requirement	Description
Performance	The application simulation runs smoothly on mid-range hardware	Ensure a seamless user experience.
Usability	The application can integrate with VR. The interface clears navigation.	Enhances user engagement and ease of use.
Localization	Ensure that all media and visuals are culturally appropriate and specific to terms of Malaysia road environment	Maintains relevance and authenticity in the simulation.

Table 6 *Button Design*

Button	Description	Button	Description
	• This is a button design • This button is designed for navigation menu to menu or confirmation button		• This is a button design • This is a button design for the next scene.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 155-175

Table 6: *Button Design (continue)*

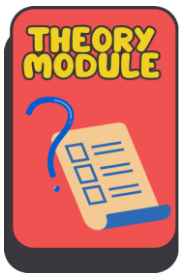
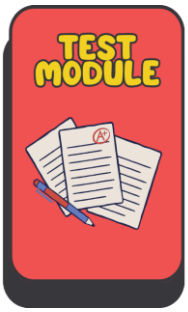
Button	Description	Button	Description
	• This is a button design • This is a button design for the previous scene.		• This is a button design. • This is a button design for back to the main menu.
	• This is a button design. • This is a button design for exiting the application.		• This is a menu button design • This menu button design for going into theory module scene.
	• This is a menu button design • This menu button design for going into practical module		• This is a menu button design • This menu button is designed for going into test module scene.
	• This is a button design • This is a button design for close the popup UI.		

Table 7 *3D models of application*

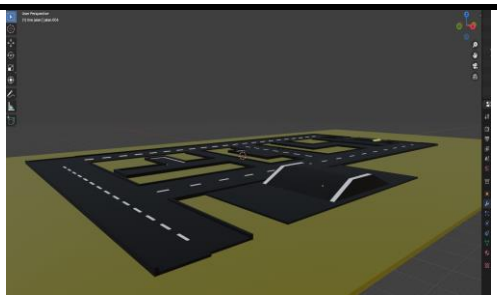
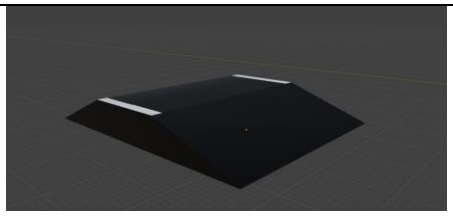
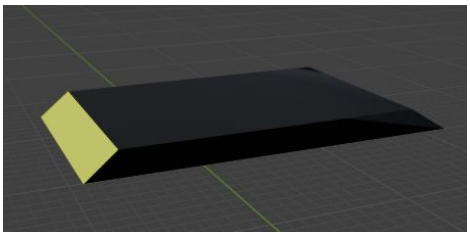
3D Models	Description
	• This is a model of circuit practical module • This circuit is specially built like a circuit at a driving training center.

Table 7: *3D models of application (continue)*

3D Models	Description
	• This is a model of hill road • This hill was built for hill driving training and testing.
	• This a model of ramp driving • This ramp was built for ramp driving training and testing.

3.3 Development Stage

During the development stage, the application is built to its full scope, making it playable with all integrated assets, features and systems. This includes finalizing and adding models such as 3D vehicles, environments and objects, while optimizing them for VR integration. Sound design is also developed, with effects for vehicles, environments and background sounds. Core driving mechanics such as acceleration, braking and steering are implemented, along with realistic driving scenarios on circuits. An adaptive feedback system provides user guidance, and a scoring system rewards progress through lessons and scenarios. Interactive elements are added to increase engagement. Level design introduces challenges in stages, from learning traffic rules to circuit training and testing. The stage ends with VR integration, including camera adjustments to reduce motion sickness and ensure compatibility with VR headsets. At this stage, to achieve the diversity of functions, it has been integrated within Unity using C# Scripting as described in Table 8.

Table 8 *Integration within Unity using C# Scripting*

Function	C# Script	Description
Vehicle Control (CARCONTROLLER.cs)	`public class CARCONTROLLER1 : MonoBehaviour { public float acceleration = 500f; public float steering = 15f; public float maxSpeed = 20f; public Transform steeringWheel; // Assign in inspector public float steeringWheelRotationSpeed = 360f; // Speed in degrees per second public float steeringWheelReturnSpeed = 180f; // Return-to-center speed (degrees per second) public float maxSteeringWheelAngle = 1080f; // 3 full turns left/right private Rigidbody rb; private float currentSteerInput = 0f; private float wheelAngle = 0f; // Accumulated wheel angle void Start() { rb = GetComponent<Rigidbody>(); }` `void Update() { float moveInput = Input.GetAxis("Vertical"); // W/S or Up/Down float turnInput = Input.GetAxis("Horizontal"); // A/D or Left/Right currentSteerInput = turnInput; // Forward/backward movement if (Mathf.Abs(moveInput) > 0.01f) { float moveAmount = moveInput * acceleration * Time.deltaTime; transform.Translate(Vector3.forward * moveAmount); } // Turning (only when moving) if (Mathf.Abs(moveInput) > 0.01f && Mathf.Abs(turnInput) > 0.01f) { float steerAmount = turnInput * steering * Time.deltaTime * Mathf.Sign(moveInput); transform.Rotate(Vector3.up, steerAmount); } RotateSteeringWheel(); }`	A snippet of the CARCONTROLLER1.cs script code which is responsible for managing player input using the Unity Input System. This script functions to control the vehicle's acceleration, braking and steering controls.

Table 8: *Integration within Unity using C# Scripting (continue)*

Function	C# Script	Description
Save temporary username (UIManager.cs)	`private void OnStartQuizClicked() { string username = usernameInputField.text.Trim(); if (!string.IsNullOrEmpty(username)) { PlayerPrefs.SetString("Username", username); PlayerPrefs.Save(); SceneManager.LoadScene(quizSceneName); } else { Debug.LogWarning("Username is empty!"); } }`	A snippet of the UIManager.cs script code that is responsible for managing player input using the Unity Input System. This script functions to receive and temporarily store the user's username entered throughout the session.

Calculation score (QuizManager.cs)

```
// Display the final score and percentage
private void DisplayResults()
{
    float percentage = (float)quizScore / totalQuestions * 100;
    scoreText.text = "Score: " + quizScore + "/" + totalQuestions;
    percentageText.text = "Percentage: " + percentage + "%";

    // Save the quiz score (you can save it using PlayerPrefs or a more complex system)
    PlayerPrefs.SetInt("QuizScore", quizScore);
    PlayerPrefs.SetFloat("QuizPercentage", percentage);

    // Update the theory progress in MenuManager
    if (menuManager != null)
    {
        // Pass the percentage to MenuManager
        menuManager.UpdateTheoryProgress(percentage);
    }
    PlayerPrefs.Save();

    // Load the result scene after displaying results
    LoadResultScene();
}
```

A snippet of the QuizManager.cs script code, which is responsible for managing scoring by calculating percentages, in addition this code also has the function of storing user scores

Condition unlock the module (MenuManager.cs)

```
public void UpdateMenu()
{
    // Unlock Practical Module if Theory Progress >= 70%
    if (theoryProgress >= 84 && !practicalUnlocked)
    {
        practicalUnlocked = true;
        practicalButton.interactable = true;     // Enable the Practical button
        practicalLockIcon.SetActive(false);     // Hide the lock icon for Practical
    }

    // Unlock Test Module if Practical Score == 100%
    if (practicalScore == 100 && !testUnlocked)
    {
        testUnlocked = true;
        testButton.interactable = true;         // Enable the Test button
        testLockIcon.SetActive(false);         // Hide the lock icon for Test
    }

    // Ensure the Theory Button is always interactable
    theoryButton.interactable = true;
}
```

A snippet of the MenuManager.cs script code which is responsible for managing conditions to ensure that the next module is unlocked after meeting the conditions. For example, the theory module requires 84% overall marks to unlock the practical module.

3.4 Testing Stage

This testing stage refers to the internal and external evaluations carried out to evaluate the functionality of a developed application. The application functionality, performance, and user experience are subject to testing. conducting alpha and beta tests according to test categories, which include functionality, performance, VR specific, usability, and compatibility.

3.5 Deployment Stage

The Deployment Stage focuses on releasing a near-final version of the application to identify and fix any performance, functionality, or usability issues before the final release. Testers conduct extensive testing to detect defects based on functional and quality standards, with bugs prioritized by severity as critical, major, and minor. For example, errors in logging. If issues are found, the application is sent back to the relevant stage for correction. Once no major issues remain, the project moves to the review stage and is uploaded for public access.

3.6 Review Stage

The Review stage marks the final review to ensure that the Driving School Simulation: Driving Success application is error-free and aligned with the project goals. The focus is on gathering feedback, refining the user experience, and preparing for release. Feedback is used to make final improvements, ensuring the application is fully polished and ready to launch.

3.7 Application Development Workflow

This subsection describes the application development workflow for agile methodology in the form of table 9, covering the requirements analysis phase, design phase, development phase, testing phase, deployment phase and review phase. Each phase will include specific tasks and the expected outputs for those tasks.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 155-175

Table 9 *The Application Development Workflow*

Stage	Activity	Output
Requirement	• Prepare time planning for requirements analysis and development.	• Determine user analysis requirement • Gathering research and information • Planning timeline by creating Gantt Chart
Design	• Prototype development • Asset and architecture planning	• Create a low fidelity storyboard • Create a navigation structure and flow chart • Define a mechanic • Build a basic prototype to test core gameplay mechanics.
Development	• Application development • Asset development and integration System development	• Implement core application mechanics, interactions and driving scenarios Integrate VR, 3D models and audio assets
Testing	• Bug fixing • Test the functional hardware VR	• Resolve issues identified during testing • Ensuring hardware works well.
Deployment	• Display to the user	• Feedback on application and VR experience from user • Identify edge cases and gather diverse feedback by expanding testing.
Review	• Perform final functional application • Perform finalization product	• Deploy application to user • Finalize final product

4. Result and Discussion

This section will discuss the results obtained from the implementation and testing phases of the project. The implementation phase involves the elaboration of the three modules of the Driving School Simulation: Driving Success application developed in Unity, and various functional integrations involving scripting, as well as VR integration. After that, the testing phase will involve the display of results for functional testing and user acceptance testing that illustrate the improvements made to the application.

4.1 Implementation Stage

This stage outlines the development of assets and integration of core functionality in Unity, along with VR support and the PC version of the Driving School Simulation: Driving Success app. Three main types of assets have been integrated: audio, graphics and 3D animation. Audio elements include background sounds and various sound effects triggered during gameplay such as sound feedback. Graphic assets such as user interface components, buttons and icons were designed using platforms such as Canva and refined in Photoshop to

achieve a consistent and user-friendly visual style. Transform animations, such as vehicle movement, button interactions and UI transitions, were created directly in Unity.

Several key functionalities were implemented using custom scripts to drive the app's interactivity. These include vehicle control functions, acceleration management, braking and steering mechanics. Scoring functions track user performance based on criteria such as passing the theory quiz and successful parking. Interaction functions allow users to engage with the environment by triggering responses when clicking on UI elements or activating tutorials. Additionally, the temporary data storage function manages user progress based on sessions, storing information such as username and will be displayed on theory quiz scoring results along with username.

The development process concludes with building applications in Unity. After confirming the correct configuration in the Player settings including the application name, version and icon, all scenes are added to the Build Settings. The application is then compiled into an executable file, ready for use and further testing on the PC platform.

4.2 Testing Stage

The testing stage for the Driving School Simulation application: Driving Success involves functional testing and user acceptance testing. Functional testing is conducted by the developer to identify and fix bugs or errors that may occur during the development process. This includes thorough testing of interactive elements such as buttons, launchers, scoring systems and other key features to ensure that they function as expected. Meanwhile, user acceptance testing (UAT) is conducted by distributing a questionnaire based on the System Usability Scale (SUS) to the target users of the driving simulation. In 1986, John Brooke developed the System Usability Scale to evaluate the efficacy of new computing systems that were being installed in DEC offices at the time. It was promptly adapted to a diverse array of applications [13]. This allows users to evaluate the overall usability, effectiveness, and experience of the application. The results from the functional tests are recorded in a results table as in Table 10, listing the expected output of each function, the actual output, and the corrective actions taken to resolve any discrepancies.

Table 10 *Result of functional testing*

Function	Test Case	Expected Output	Actual Output	Corrective Action
Buttons	Start application	Press any key to continue the main menu	Pressing any key to continue is successful, there is a delay trigger	adjusted by trigger timing
	Back	display the previous scene	Works well as expected.	Not needed
	Main menu	Display the main menu scene	Works well as expected.	Not needed
	Navigation button	Correct screen or action loads on click	Some text buttons use inappropriate terms	adjust the text button according to the appropriate terms

Table 10: *Result of functional testing (continue)*

Function	Test Case	Expected Output	Actual Output	Corrective Action
Audio	Background sound	Background sound plays continuously	Audio loop cut off after a few seconds	Fix audio loop setting in Unity
	Sound effects	sound feedback of interaction	sound feedback there is a delay after the trigger	adjust the sound output timing

Graphic	Display main menu	displays a working main menu	Works well as expected.	Not needed
	Scene transition	Transitions from one interface to another smoothly.	Transition to another interface delayed and crash screen	Adjusted to adding the loading screen as a transition interface
	3D environment	3D models render properly with no missing textures	Works well as expected.	Not needed
Scripting	Scoring sytem	Update scores based on performance and errors	Works well as expected.	Not needed
	Module level unlocking	Complete the theory module with a minimum score 84% pass to unlock next module	Works well as expected.	Not needed
	Temporary data storage	Session data retained temporarily during the session	Works well as expected.	Not needed
Mechanics	Vehicle control	The vehicle responds accurately to user input with smooth control	Works well as expected.	Not needed
VR integration	Simulation using VR headset	Simulation runs smoothly in VR with head tracking	Head tracking worked, but UI misaligned in headset view	Adjust VR canvas placement and calibrate camera settings

Table 11 *Respondents' score (User Acceptance Test)*

Respondent	Item score										Odd items	Even items	SUS Score
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10			
R01	5	5	5	2	5	1	5	1	5	1	25	10	87.5
R02	5	5	3	5	5	3	5	3	5	5	23	21	110
R03	4	3	4	5	5	2	5	1	4	3	22	14	90
R04	5	5	5	1	4	1	4	1	4	1	22	9	77.5

R05	5	5	5	1	5	1	5	1	5	1	25	9	85
R06	1	1	1	5	1	5	1	5	1	5	5	21	65
R07	5	3	5	4	2	2	5	2	5	3	22	14	90
R08	5	1	5	1	5	1	5	1	5	1	25	5	75
R09	5	5	5	5	5	3	3	1	5	3	23	17	100
R10	5	1	5	1	5	1	5	1	5	1	25	5	75
								Average score					85.50

As part of the User Acceptance Testing process, participants completed the System Usability Scale (SUS) questionnaire to assess the usability of the application. The efficacy of the system from the perspective of each user is quantitatively represented by this score [14]. The questionnaire included 10 standard SUS items and was created using Google Forms. Each question used a 5-point Likert scale, ranging from "Strongly Disagree" to "Strongly Agree." A total of 10 respondents completed the form. Their responses were analyzed by calculating the total scores for the questions, followed by normalization to obtain an overall SUS score. The results, shown in Table 11, revealed an average SUS score of 85.50 percent, indicating an excellent level of usability and high user satisfaction with the application. Detailed user feedback and calculations are included in Appendix E.

5. Conclusion

In conclusion, this project has been successfully completed with the development of the Driving School Simulation application. The results of this project can help provide training and overcome the challenges of driving training centers, where driving schools usually do not have the right platform for training and testing [15]. Furthermore, the agile methodology has contributed to the project being completed on time. Positive feedback has been received from early users, indicating a good level of usability and effectiveness of the content. In addition, the project objectives set at the beginning have all been successfully achieved. The first objective, which is to design a virtual driving school simulation application for driving school students, was achieved by integrating the educational content of driving learning content in Malaysia, as well as ensuring that it is user-friendly enough for driving school students. The second objective, which is to develop a basic driving school simulation with virtual reality (VR) technology focusing on module theory, training and circuit testing that provides virtual experience and learning on desktop platform (PC), was achieved by developing vehicle control mechanics such as braking and acceleration, along with ensuring that the application can run smoothly on the desktop platform (PC) without significant bugs or performance issues. The third objective, which is to evaluate the function and usability function of the application through user feedback and testing, was achieved by undergoing functional testing and user acceptance testing to fix major bugs and evaluate its usability.

Suggestions will also be used for application improvements. To overcome the limitation that the application is currently only available on desktop platforms with VR integration, it can be developed with more advanced integration with other hardware such as steering wheels for desktops in the future. For the second limitation that the application is currently available with theory modules and circuits, it can be further developed to extend training and road test modules. Overall, the application is on track and shows strong potential to develop into a comprehensive virtual driving school solution, but further improvements can be made to enable the application to be improved in terms of its content soon.

Acknowledgement

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

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design**: Muhammad Shahrul Ikhwan Abdul Halim, Mohd Farhan Md. Fudzee; **data collection**: Muhammad Shahrul Ikhwan Abdul Halim, Mohd Farhan Md. Fudzee; **analysis and interpretation of results**: Muhammad Shahrul Ikhwan Abdul Halim, Mohd Farhan Md. Fudzee; **draft manuscript preparation**: Muhammad Shahrul Ikhwan Abdul Halim, Mohd Farhan Md. Fudzee. All authors reviewed the results and approved the final version of the manuscript.

References

[1] Muhammad Sulaiman, M. N. H., & Shapi'i, D. A. (2018). *Simulasi pemanduan kereta: Driving car simulator* [Technical report]. Universiti Kebangsaan Malaysia. https://www.ftsm.ukm.my/v5/file/research/technicalreport/PTA-FTSM-2018-148.pdf.

[2] Mahpar, N. (2024, Aug 24). *Sepatutnya risau lesen murah bukan mahal, kata persatuan institut memandu*. Free Malaysia Today

[3] lesen2u, "Kesan Tidak Perbaharui Lesen Memandu," LESEN2U, https://lesen2u.com/kesan-tidak-perbaharui-lesen-memandu/ (accessed Oct. 11, 2024).

[4] Xie, B., et al. (2021). A review on virtual reality skill training applications. *Frontiers in Virtual Reality, 2*(1). https://doi.org/10.3389/frvir.2021.645153

[5] Electronic Arts. (2025). *F1® 25 system requirements for PC*. https://www.ea.com/en/games/f1/f1-25/system-requirements (accessed Jun. 11, 2025).

[6] Polyphony Digital. (2025). *Gran Turismo 7*. https://www.gran-turismo.com/us/products/gt7/

[7] Forward Development. (2019). *Forward*. http://forward-development.ru/en (accessed Jan. 02, 2025).

[8] OviLex Software. (2025). *Driving School Simulator*. https://www.ovilex.com/projects/driving-school-simulator/ (accessed Jan. 02, 2025).

[9] 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

[10] "What is Agile Project Management? A Comprehensive 101 Guide,"slite.com. https://slite.com/learn/agile

[11] Highsmith, J. (2009). *Agile project management: Creating innovative products* (2nd ed.). Addison-Wesley

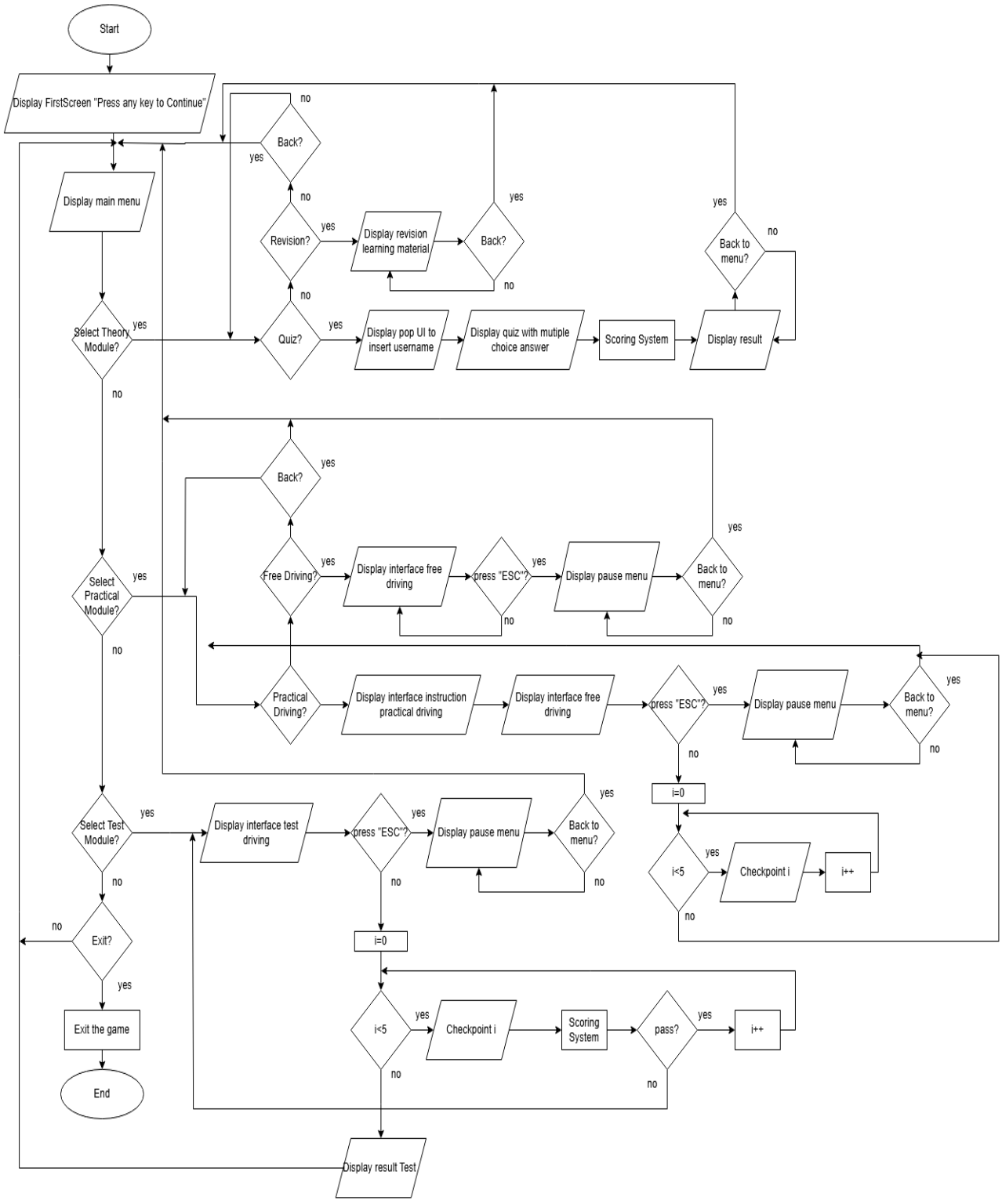
[12] Beck, K., et al. (2001). *Manifesto for agile software development*. Agile Alliance. https://agilemanifesto.org/

[13] Betteridge, K. (n.d.). *What every client should know about SUS scores*. Bentley University. https://www.bentley.edu/centers/user-experience-center/what-every-client-should-know-about-sus-scores

[14] Suria, O. (2024). A statistical analysis of System Usability Scale (SUS) evaluations in online learning platform. *Journal of Information Systems and Informatics, 6*(2), 992–1007. https://doi.org/10.51519/journalisi.v6i2.750.

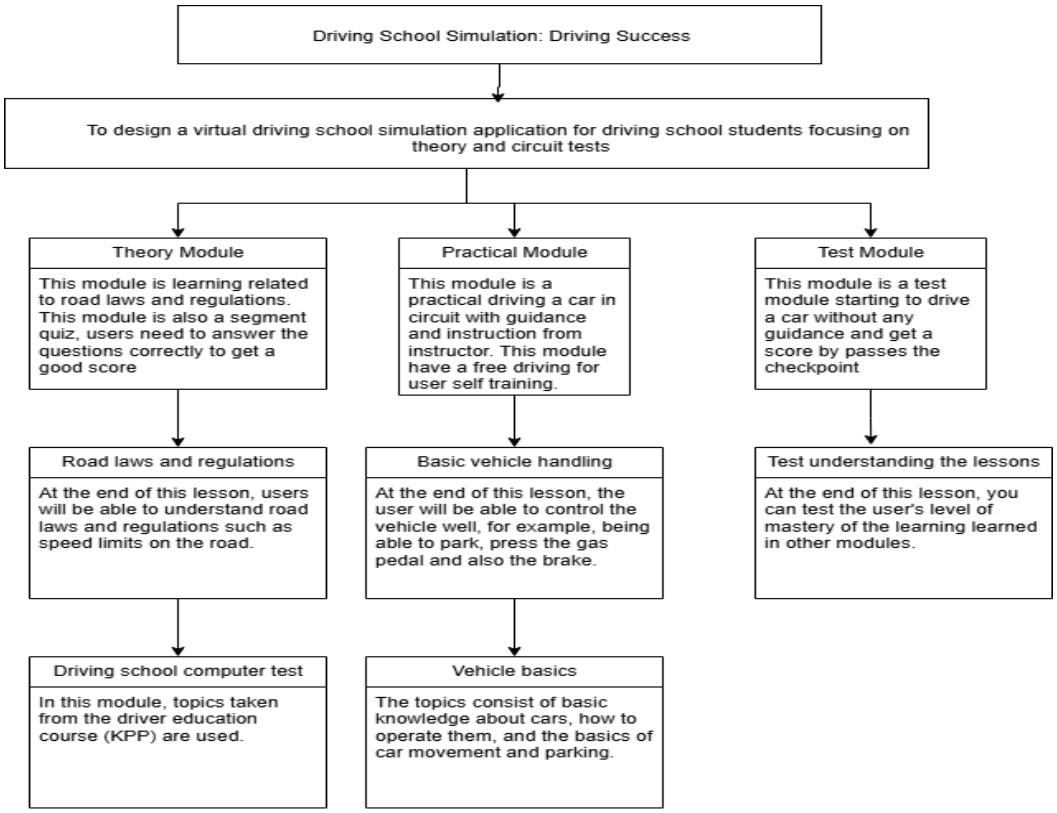
[15] Jawi, Z. M., Deros, B. M., Osman, M. R., & Awang, A. (2015, November). A systemic overview on driver training and driver licensing system in Malaysia. In *Proceedings Conference ASEAN Road Safety* (pp. 185-91).

Appendix A: The Application Flowchart

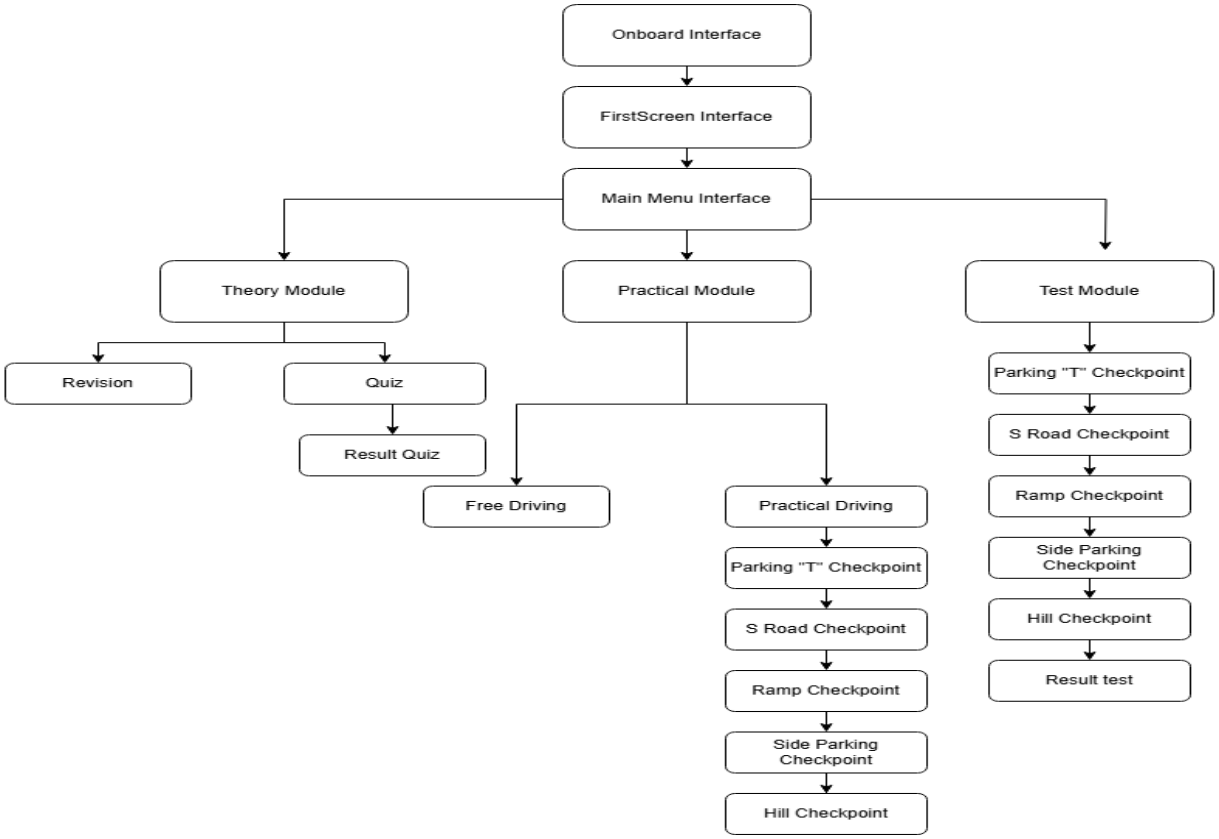


Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 155-175

Appendix B: The Application Content Structure

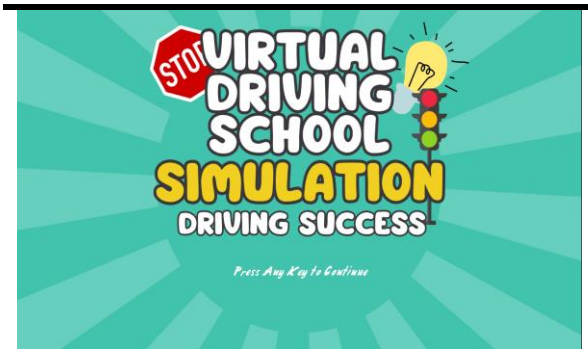


Appendix C: The Application Navigation Structure



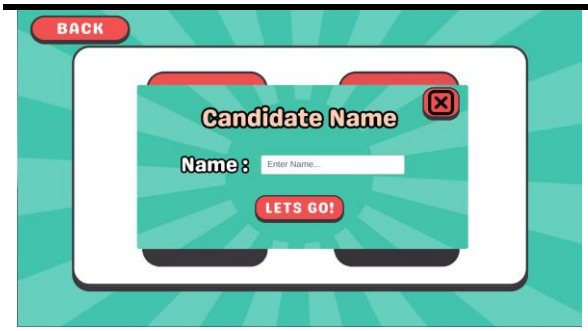
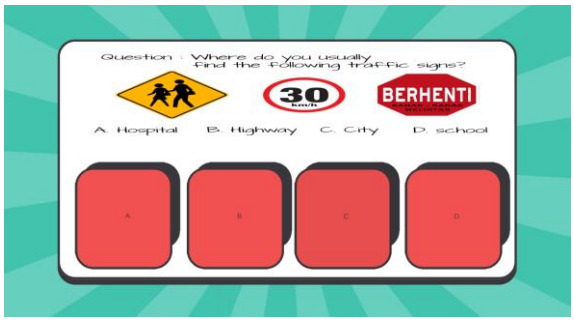
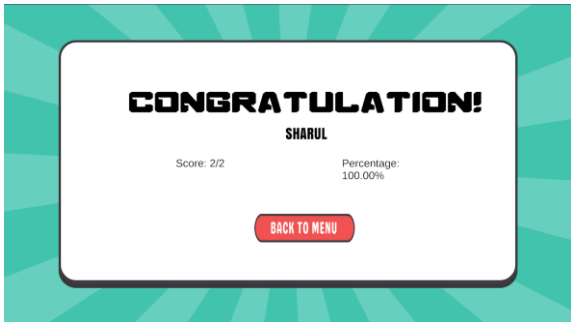
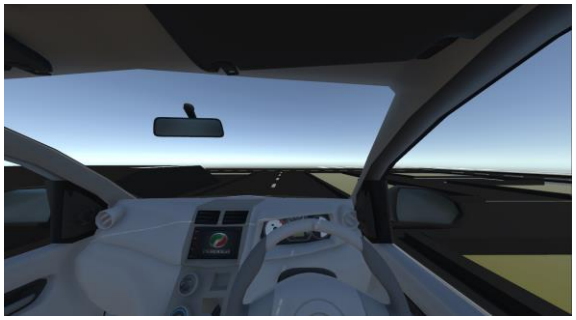
Appendix D: Interfaces Application

Table 2 *Storyboard and interface of game application*

Items	Description
	• This is the starting page of the application. • Users need to press any key will bring the users to the main menu.
	• This is a main menu to choose a module • The exit button will cause the users to quit the game • Three buttons of theory, practical and test module. • Users need to pass a minimum of 84% of the theory module to unlock the practical module.
	• This is a menu inside of the theory module • Two buttons of revision and quiz in theory module. • Revision is to study the learning material about road laws. • Quiz to test the user's level of understanding
	• This is a revision scene • The material is obtained from the KPP syllabus • Users can press the button next and back to change the slide. • Back button to go back the theory module menu.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 155-175

Table 2: *Storyboard and interface of game application (continue)*

Items	Description
	• This is a popup UI to obtain username by user input. "Let's go" button is to continue the quiz scene
	• This is a scene quiz of a multiple choice • 4 buttons of answer A, B, C, and D • Question at the top of the answer
	• This is a scene result to display the score • The username will display on this scene with the score and percentage. • Back to the menu button will go back to the main menu.
	• This is a menu of practical module. • Two buttons of choice for free driving and practical driving module. • Back button to main menu.
	• This is a scene of practical driving and free driving • This is an interface that will display VR integration.

Appendix E: System Usability Test

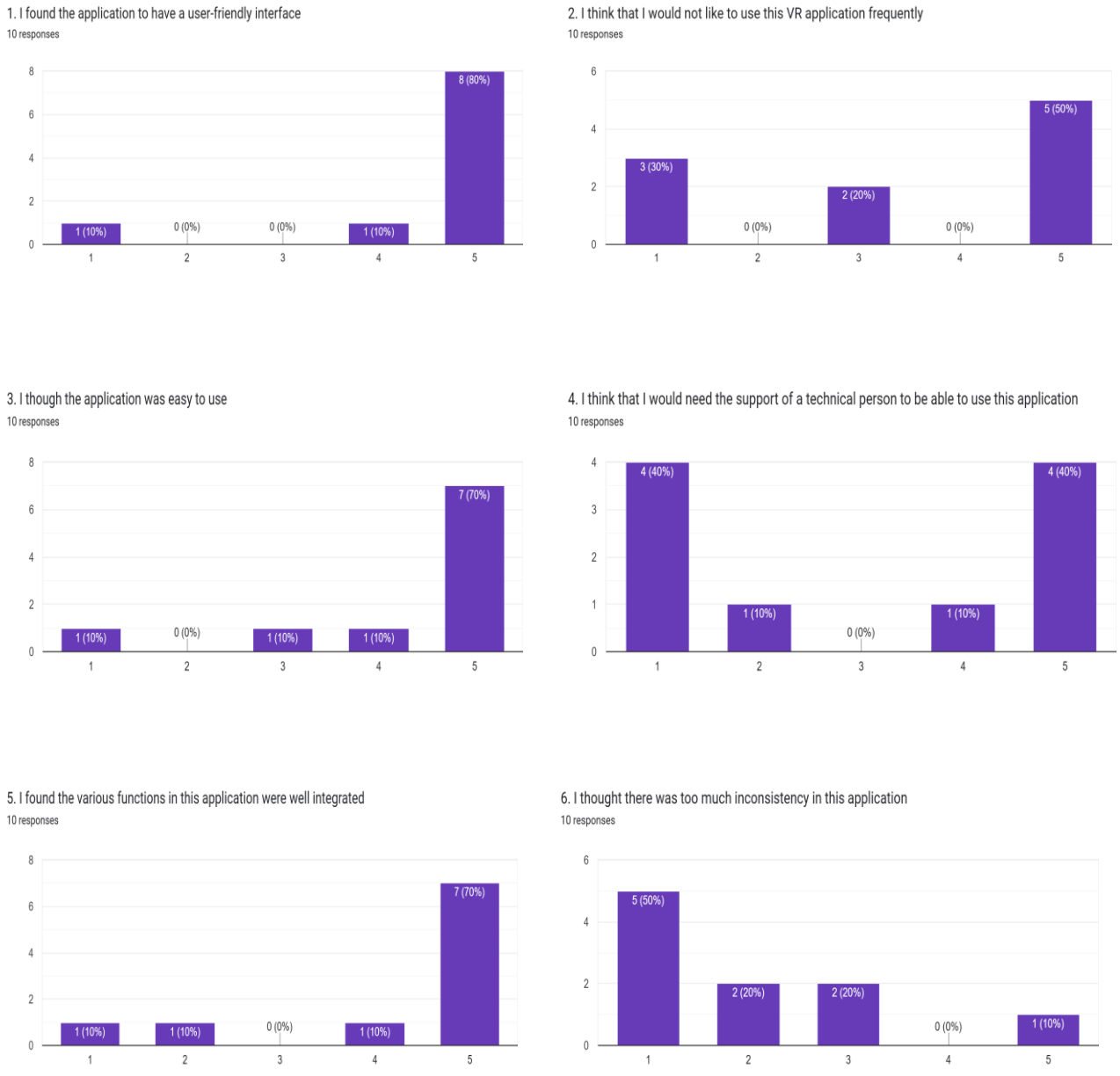


Fig. 3 *Responses to the ten System Usability Scale (SUS) questions*

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 155-175

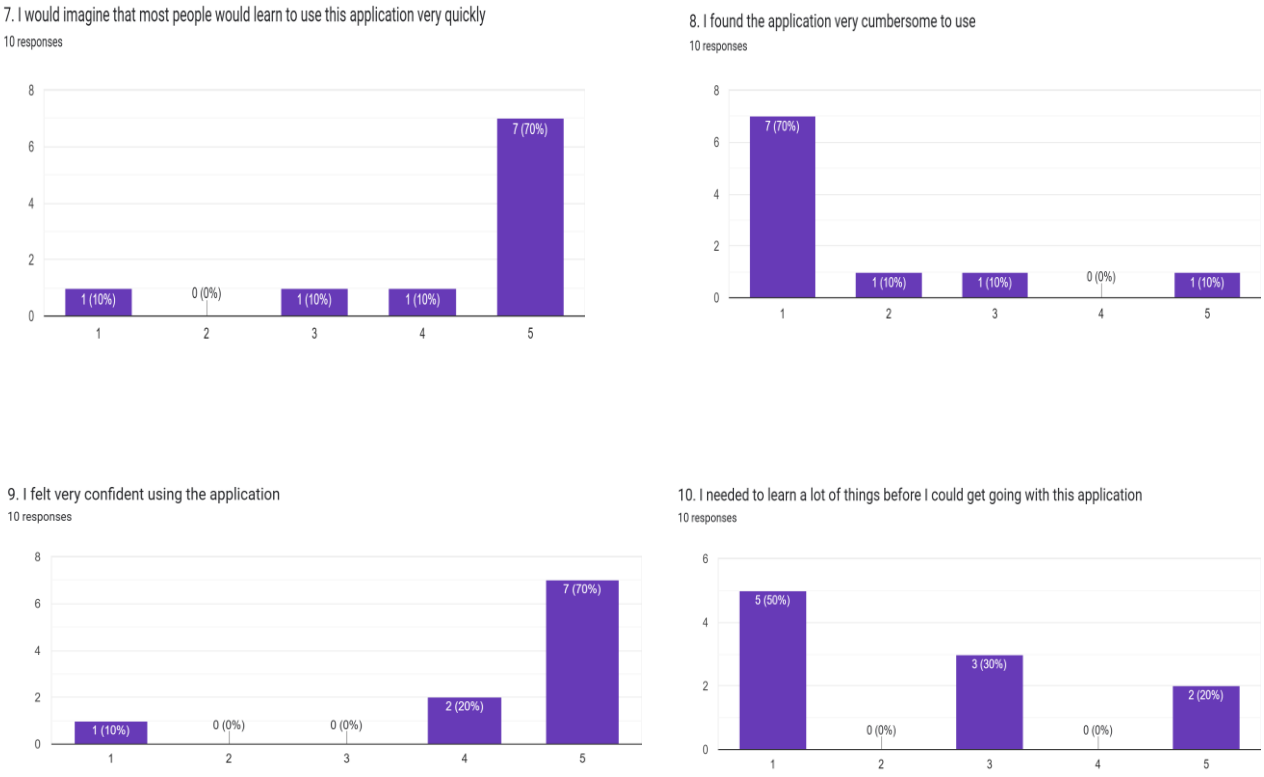


Fig. 3 *Responses to the ten System Usability Scale (SUS) questions (continue)*

Step 1: Convert the user ratings from the 10 questions into points

For odd numbered questions 1, 3, 5, 7, 9:

- [User Rating] – 1 = ___ points

For even-numbered questions 2, 4, 6, 8, 10:

- 5 – [User Rating] = ___ points

Step 2: Add up the points from the 10 questions into a user's total points

- [Question 1: __ points] + [Question 2: ___ points] + … [Question 10: ___ points] = ____ total points from user 1

Step 3: Multiply the user's total points X 2.5* to get an individual user's score.
*This converts the scale to be out of 100.

- [Total points from user 1] x 2.5 = User #1 score

Step 4: Repeat steps 1-3 for all users, and then average all user's Scores together to get a SUS score:

- User #1 score + user #2 score + user #3 + [etc.] / # of users = SUS Score

Fig. 4 *Calculation SUS score [14]*