Chapter 1

Introduction

Legal driving license acquisition takes a long time. Getting a provisional licence is the first step, and then passing the theory and practical tests comes next. The theory test syllabus is specified in appendix A. In the UK, an increasing number of young adults are pursuing full driving licenses each year [1], reflecting a growing interest in driving.

As of 2024, there are a variety of resources available for new learners designed to help with driving theory. The Driver and Vehicle Standards Agency (DVSA) published the Official Highway Code Book [2], which consists of a comprehensive collection of road rules and regulations. Additionally, the DVSA also released an official driving theory learner app, allowing individuals to study at their own pace on their devices. Beyond this, the Internet, YouTube videos and digital libraries are accessible with the relevant material. With such a range of study tools available, it is expected that learners can pass their Driving Theory tests easily.

Despite this, research conducted by Scott Le Vine and John Polak [3] indicates that the learning experience for Driving Theory is significantly unsatisfactory. Their study explores why many "young adults delay or forgo acquiring a driving license". Several key factors have been identified based on the motivations as to why they are not fully licensed. The common reasons drawn are that driving is often considered a "low priority", many feel "too busy to learn," or they lack sufficient interest. Additionally, their analysis shows that young adults spend an average of 1.7 years learning to drive after reaching the eligible age of 17. Given that people are preoccupied with other responsibilities, the considerable time needed to learn can be seen as a deterrent to fully commit for license acquisition.

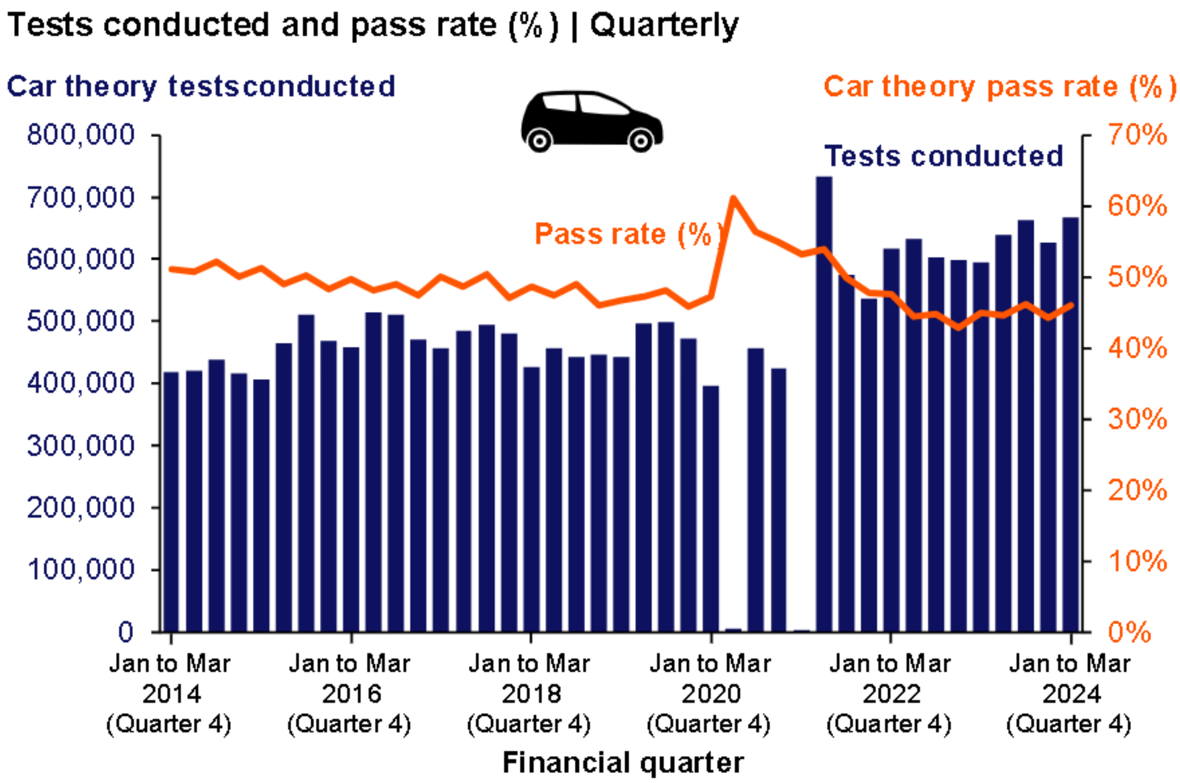

Figure 1.1: Graph of pass rate of the Driving Theory exam [4]

According to Figure 1.1, the pass rate for the UK driving theory test (Category B) peaked during the global pandemic in 2020 because of the lenient exam conditions at the time with COVID-19 social distancing policies in place. However, it has been steadily declining as the lockdown has slowly been lifted, at a rate of 0.6%. Even with the resources provided, the pass rate in 2024 is currently under 50%. This trend strongly suggests that there may be issues with the Driving Theory learning process. Therefore, it is critical to study the current methodologies in the learning process, especially from the computer science perspective.

1.1 Aims

The research aim for this project is to be able to improve upon the learning experience for the Driving Theory test for **category B licenses only**. To achieve this, the following objectives need to be met beforehand:

- 1. Identify factors why the current learning experience is poor.
- 2. Create an application as a tool for driving theory revision.
- 3. Evaluate how effective this is with a questionnaire.

Once the factors are identified and written into requirements for the development of the application, they can be tested for several non-drivers willing to learn. Therefore, the learning experience is now improved.

1.2 Time Management

In figure 1.2 below, this is the Gantt chart of the project that is the plan in a timeline from October 2024 to May 2025. It also includes key milestones that need to be achieved in steps and subtasks for this project.

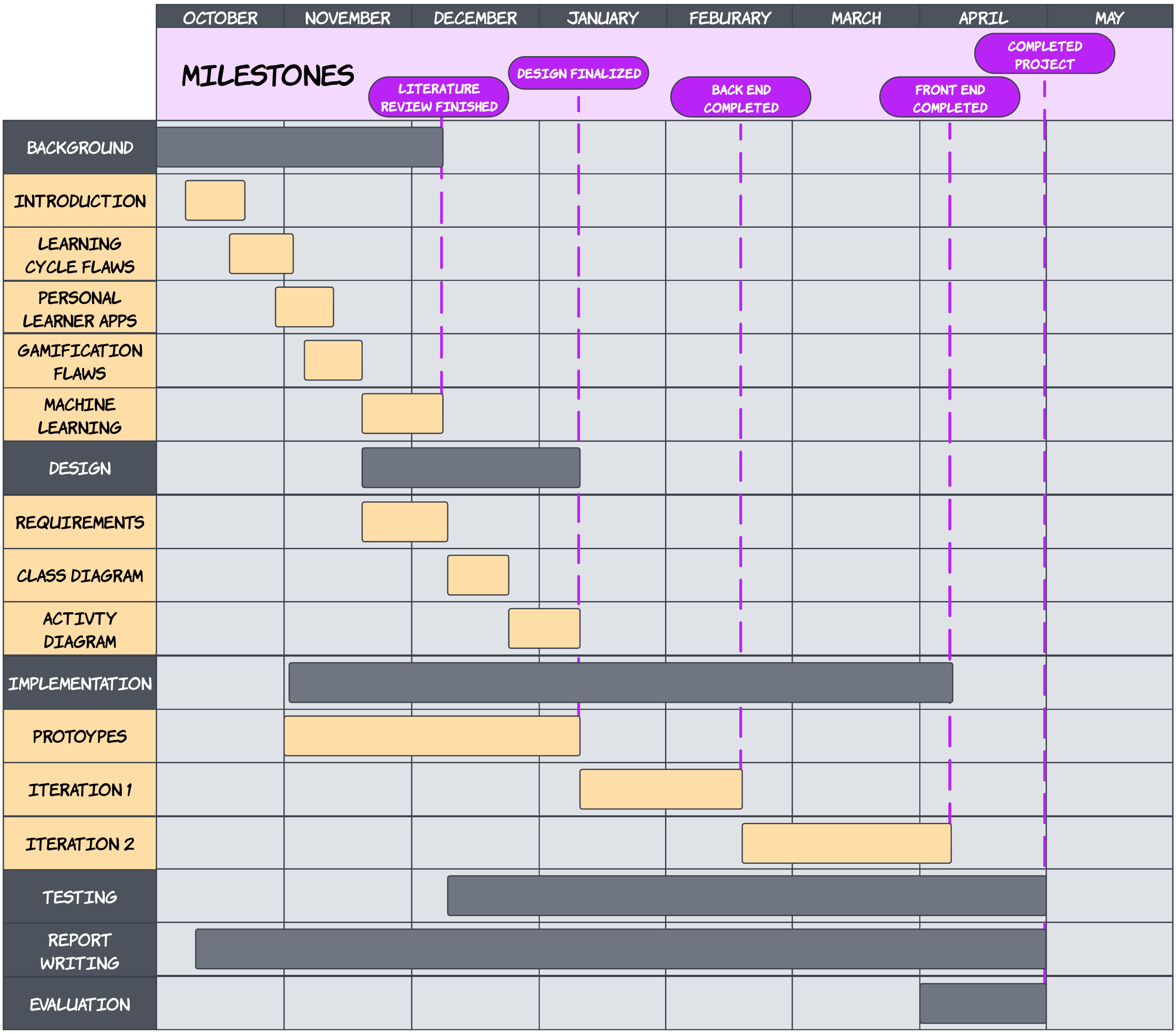

Figure 1.2: The Gantt Chart plan for the whole project

Chapter 2

Literature Review

It is important to identify the factors that affect the learning experience, which is the first research objective of the project. With these factors in mind, it is also worth investigating the existing methods and identifying any gaps in the literature, which is covered in this section. This will contribute to a solution that will help those learning Driving Theory.

2.1 Kolb's Learning Cycle

The foundation to address the problem is to explore how people learn in general. Thus Learning Cycles, which are the frameworks that abstract the phases of the learning process, are a key concept to begin with. This allows people to absorb and remember new information and skills about anything.

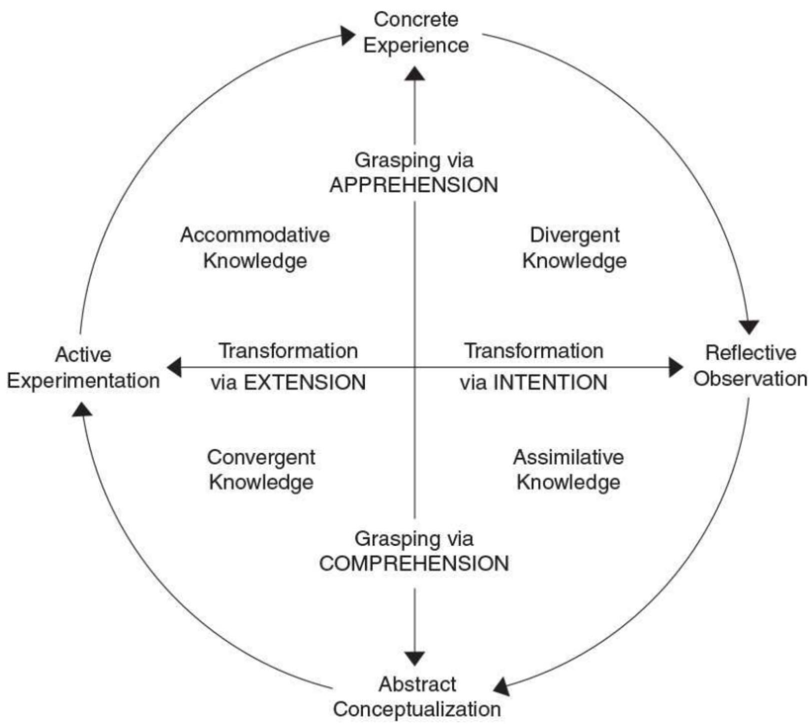

Figure 2.1: Diagram of Kolb's learning cycle [5]

Kolb's Learning Cycle, shown in figure 2.1 above, is developed by educational theorist David Kolb [5]. It is an experiential-based cycle illustrating how individuals gain a deeper understanding of concepts through means of *repetition*. Kolb depicts the cycle as a spiral as when learners revisit a topic or concept, they grasp more depth and detail about it each time. The stages of the cycle are:

1. **Concrete Experience:** Have a new experience or reinterpret an existing one.
2. **Reflective Observation:** Reflect on the experience.
3. **Abstract Conceptualisation:** Form or change theories based on reflections.
4. **Active Experimentation:** Apply the theory to see any changes.

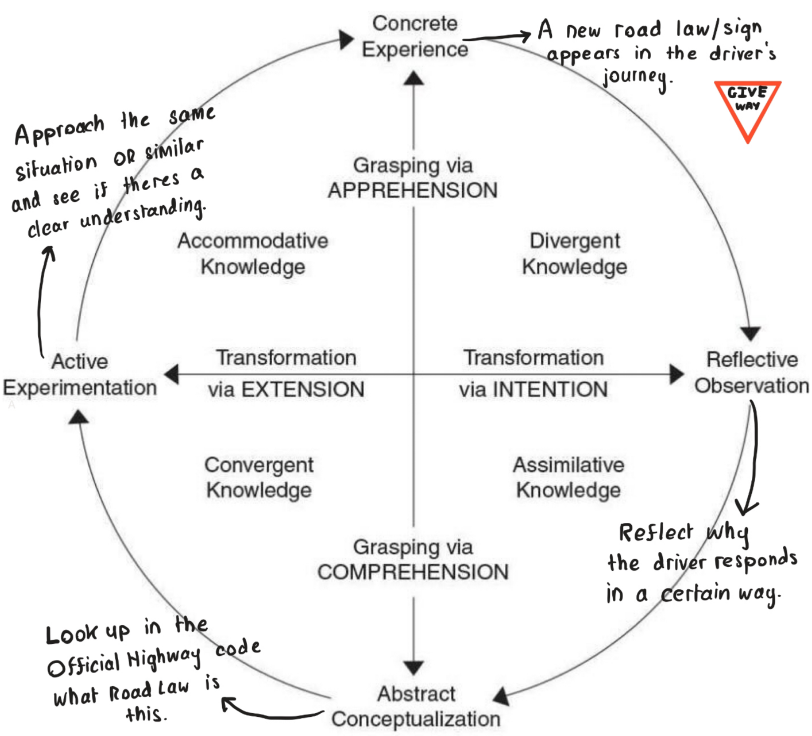

Figure 2.2: Learning how to drive example

Kolb's Learning Cycle can be applied to learning how to drive. As shown in Figure 2.2 above, a learner faces a "yield" road sign, which instructs the driver to give way to incoming vehicles or pedestrians. Firstly, the learner observes a licensed driver when they stop in response to the sign. Next, the learner then reflects on the experience and refers to the sign's definition in the Driving Theory material for confirmation. Finally, the learner applies this knowledge in future driving experiences, progressively improving their understanding.

Nonetheless, Kolb's learning cycle has faced criticism in recent papers, such as in John Boyd's work [6]. According to Boyd, the model is overly "linear" and "rigid", which could slow one's learning. Kolb's framework places a strong emphasis on moving through each step in order. It can be discouraging for the learner to have to repeat the entire cycle if they make a mistake during the

”active experimentation” phase. For instance, a novice driver can come across another ”Yield” sign on a different road and not instantly recall its meaning. This scenario highlights the limitations of Kolb’s rigid structure, as the learner could find it difficult to adjust.

Furthermore, time between cycle stages is not taken into consideration by Kolb’s learning cycle assumption of a continuous learning flow. This presumes that all learners hold perfect motivation. In reality, individuals often need time to mentally recover before they can continue to learn. This makes handling mistakes within the cycle require even more time with mental breaks, which can be discouraging due to the noteworthy time and effort. Reiterating Scott Le Vine et al. [3], it is important to recall many individuals lack either the time or the motivation to dedicate themselves fully to learning how to drive.

2.2 The Ebbinghaus Forgetting Curve

The ”complex task” of learning to drive, which requires a ”range of lower and higher-order abilities” [7], is another important problem. On top of learning how to use the car’s gears, indicators and steering, one needs to also immediately recall what road rules and laws apply to certain situations. Large volumes of this content can make learning cycles quite overwhelming and, as a result, demotivating. Managing the learnt information will be extremely challenging, soon after the learning quality declines. Since this is memory-based, this could potentially lead to a rote understanding rather than a true understanding.

For beginners, this is a significant challenge because they must review Driving Theory on a regular basis. Learners need to interact with the material actively to internalise the information. Passive study methods like underlining and reading text are less efficient than active study methods like ”active recall” and ”spaced repetition” [8], which involve going over the material again at regular intervals.

This aligns with the concept of the ”Ebbinghaus forgetting curve”, which describes how memory declines over time. The theory discloses that a large amount of newly absorbed information is forgotten rapidly, especially within the first few hours [9]. This means that if a learner cannot stay consistent with spaced repetition, they’ll inevitably forget which can result in reduced motivation due to a lack of progress in learning driving theory.

In Figure 2.3 below, the curve shape demonstrates how quickly learners forget information initially, but then the forgetting rate flattens over time. The rapid decline shows that memory retention drops exponentially after learning new content. Because of this problem, Ebbinghaus has also endorsed the study technique of spaced repetition that mitigates the forgetting process. Every review slows down the rate which in turn strengthens memory retention. This means the time in between reviews increases due to the information being more encoded in the long-term memory.

Figure 2.3: Diagram of the curve [10]

In support of the learning cycle concept, active recall has shown to have helped learners spot any misconceptions early on [11] to recall for future study reviews. With this in mind, John Boyd has endorsed a more agile mindset and philosophy to learn as efficiently as possible. It's "the need to be flexible within a given system, and to keep one's orientation matched to the real world" [6]. He believes in instant feedback so the previously learned knowledge can be reinforced and contextualised further, rather than statically learning the road rules individually.

2.3 Learner applications

There are various approaches to deal with memory retention and motivation with active recall, such as flashcards, mind-maps and anything that involves visual learning [12]. But, one mainstream solution for routine-based learning is the use of learner applications. These are digital tools, educational apps, hands-on exercises, or interactive methods that are generally more accessible. They are made to help learners acquire knowledge and develop skills more effectively [13].

Currently, numerous driving theory learner applications are available, including the Official DVSA Theory Test Kit [14], Driving Theory Test 2024 by Driving Test Success [15], and Driving Theory Test 4 in 1 [16]. This section looks at the features of these applications, highlighting key characteristics and identifying common elements that contribute to their effectiveness for various users. These Driver-learner apps provide the following features to prepare users with the knowledge and skills to pass their driving theory and practical tests [13], which include:

- Multiple choice quiz games.
- Interactive videos for hazard perception.

- Road sign quizzes with a list of signs and their meanings.
- Mock exams designed like the actual exam.

Additionally, these apps also have progress-tracking features that track user performance, identify areas of weakness, and recommend categories to suit each user's needs. Concurrently, educators underscore the necessity of robust feedback systems and the availability of structured curricula to support effective learning outcomes [17]. This addresses Boyd's criticisms of Kolb's learning cycle. By implementing dynamic adaptation to where a learner needs improvement, this can give users guidance on where to focus in their following learning sessions.

A key characteristic outside of the traditional learning styles all the applications have in common is their reliance on gamification. By incorporating achievements, leaderboards, daily challenges, and progress-based rewards, they transform the learning process for Driving Theory into an interactive, game-like experience. This characteristic is widely used to uplift learner's motivation consistently. Since this is also implementing spaced repetition, this also helps to retain information prevent the learner to forget what they have learned on a regular basis.

2.4 Gamification

To address the motivation and memory retention issues as explained in sections 2.1 and 2.2, Gamification in learning integrates game-like elements into the educational experience. Learners can be motivated and engaged to practice daily. This section looks into real-world examples of how well gamification works in learner applications.

Incorporating gamification elements into the Driving Theory learning experience is essential because it "increases the learners' interest" [18]. Kačerauskas et al. [19] highlight that gamification boosts motivation, and supports self-organization and iterative learning in education. Still, its "temporality" remains its biggest inadequacy. The user experience diminishes the effectiveness of gamification, even reducing long-term interest.

Similarly, a systematic review by M. Shortt et al. [20] on Duolingo, a learner app for languages, showed a generally positive correlation between the usage of gamification in education but with some flaws. Notably again, repetitive content caused some learners to feel "diminished motivation", which makes spaced repetition obsolete as a study technique to rely on when using learning apps. This ultimately results in learners to forget and hence tarnish the entire learning experience.

Temporality and lack of replayability are two significant problems with the design of many of the learner apps available today. This observation highlights the importance of balancing reinforcement with the introduction and internalisation of new material. The aim is to sustain learner engagement and optimize long-term retention.

2.5 Replayability

As previously mentioned, many learner applications have temporal issues, leading to a decline in engagement from learners. This section examines the idea of replayability; and how it can be integrated into learner apps to ensure it will be used more regularly.

The research study by Rose Adetunji [21], explores how replayability impacts educational game design. Factors are found to enable students to stay consistent doing active recall. With the recent rise of articles on gamification, there is a gap in the literature on the impact of replayability. The study suggests the importance of "supporting repeated play and diverse learning experiences." The key factors that drive a player to replay a game are:

1. **Variety/Randomness:** Adding variation to scenarios to add depth.
2. **Goals/Completion:** Providing feedback gives insight to users.
3. **Difficulty:** Increase in difficulty can allow test a user's understanding.
4. **Social aspects:** Playing in friendly competition with other users.

Through repetition in progressively different circumstances, replayability enables students to engage with the content differently each time. As a result, this strengthens their retention of the theory. An experiment [22] tested a replayable RPG, adding elements similar to Adetunji's study [21]. Player motivations varied. For instance, some wanted to explore game mechanics (Variety/Randomness), others aimed to master the game (Goals/Completion), and some were driven by the desire to overcome challenges (Difficulty).

By augmenting more replayability with the factors discussed, the game design in the current Driving Theory Learner applications can mitigate the loss of motivation. The purpose of the application being as replayable as possible is to help the learner carry out spaced repetition consistently before their Driving Theory exams.

2.6 Framing the problem in Machine Learning

Now that the factors have been identified, there's room for improvement in learner applications. Machine learning can be applied, as it can use one's learning statistics. Performance can be measured with quiz scores and time taken on questions for instance.

It is required to use a learner's performance over a study session as input data to a statistical model to evaluate and generate the appropriate feedback. Predictions also need to be made to recommend topics/questions based on which are their general strengths and weaknesses. Therefore this is a **Supervised learning** task that utilises **Multivariate Regression.**

Supervised learning is a type of machine learning that uses labelled input output pair datasets to train models to learn to recognise patterns and predict unknown data points [23]. For this case, the inputs/features would be attempted questions as well as other performance metrics already mentioned. In contrast, a typical target variable would be the question's score. One research paper that has reviewed the usage of machine learning in education, 88% of found studies that utilise them, have specifically applied supervised learning [24], which endorses the use of the model.

Regression, on the other hand, is a Supervised learning technique that predicts continuous values [23], in this instance that would be the performance scores for future attempts at mock Driving Theory exams. Multivariate analysis has been applied in the educational setting, to statistically represent a distribution of students' marks across different exam papers as an example [25]. Therefore this problem can also be considered **Multivariate** as the evaluation is based on multiple Driving Theory topics, time spent on questions and scores on quizzes. It is important to make the model flexible for a range of learners. This is essential to evaluate what topics and questions need more practice.

Several regression algorithms, such as Linear Regression, Decision trees and Random Forest, offer robust methods for Multivariate analysis [23]. These models were commonly mentioned to be used for education [24]. The table below shows comparison of the models:

Model	**Details**
Linear Regression	• Models the relationship between one or more input variables in a straight line [26]. • Multiple linear regression models were used to find any correlation for academic and non-academic variables to predict a student's likelihood in academic success [27].
Decision trees	• Splitting data into different branches based on feature values, leading to a prediction (decision) to a leaf node. • Used to evaluate an education course performance using student demographic data, online engagement and course factors. By following the branches based on their attributes, it predicts their potential performance outcome [28].
Random Forest	• Collection of decision trees where each is trained differently, therefore the randomness factor. • Has been used in various statistical contexts such as Prediction of default Credit Card activity to detect any suspicious actions [29], medicine and economics [30].

Given the large input features and the unpredictability in learners' progression, Random Forest proves highly effective. It excels at capturing complex, non-linear relationships of learner performance [23]. In contrast, linear regression always assumes a linear relationship, which doesn't fit well with features with too much noise between features like time spent and accuracy, as each learner is different. Moreover, it may capture noise rather than any underlying patterns due to its inefficiency with outliers [31].

Decision trees, though interpretable and capable of modelling non-linear data, are prone to overfitting, especially with large datasets, and very sensitive to minor data or parameter changes, leading to unreliable predictions. Random Forest, an ensemble method that introduces randomness during tree building, mitigates these issues. As Becker's paper on decision trees and Random forests [32] suggests, this stochastic approach effectively handles the variability inherent in user learning by considering numerous diverse trees simultaneously, thus improving robustness.

2.7 Applying the Random Forest Algorithm

The concluding section of this review looks into the Random Forest algorithm in depth and how it's potentially applied as a component to evaluate user performance in a study session to this new learner application for Driving Theory.

As established earlier, this algorithm utilises Decision Trees. Random Forest trains multiple decision trees on different combinations of subsets of the data and subsequently aggregate their predictions to produce an overall prediction. It's a versatile method as it can carry out regression (By averaging the decision tree predictions) or classification (By selecting via majority vote) [23]. Specifically for learner's performance data, regression is needed as stated previously. Here's how the algorithm works:

1. **Bootstrap Sampling:** Create subsets of training data by sampling with replacement.
2. **Random Feature Selection:** In each split in a decision tree, use random features.
3. **Ensemble Prediction:** Prediction based on the decision trees' final decision.

In order to increase the precision, one model can be utilised per topic. This means there will be predictions based on the performance on all the questions under the same category, so that the model will avoid too much generalisation. An example usage in personalised application to predict a learner's score in a category can use the following features like time spent, accuracy and difficulty level of a question. This is due to the differing nature of the Driving Theory Topics. For example, Hazard Perception heavily depends on the accurate timing spotting the developing hazard whereas a multiple choice question needs to be answered quickly and precisely.

2.8 Summary

To summarize, this section showcased a problem in the learning experience for Driving Theory. The main factors are that people need motivation and are prone to forgetting material they've studied, especially for how immense Driving Theory is as a subject. There are flaws in general learning models like Kolb's learning cycle. Boyd's critique of the cycle's inherent inflexibility and overly rigid structure underscores the need for balance between periods of mental rest and focused study [6]. These shortcomings contribute to challenges like the forgetting curve, where learners struggle with retention over time, leading to diminished motivation.

Current solutions, such as Driving Theory learner apps, heavily utilize gamification to improve memory retention and engagement. While effective initially, limited replayability often results in boredom, reduced motivation, and decreased learning outcomes. To combat these issues, Supervised Learning with Multivariate Regression through **Random Forest** can be applied, enabling deeper monitoring of learner performance, personalized feedback, and sustained engagement.

With the background research addressing all the current problems of the learning experience of Driving Theory, this can help establish the design choices for the newly refined Driving Theory learner application. It will utilise Gamification and attempt to improve the replayabilty and give continuous, precise and constructive feedback. These are especially important requirements to include for this application in order to keep learners motivated and avoid forgetting. Therefore, the learning experience can be improved.

Chapter 3

Application

Now with the literature review, the requirements and user stories can be generated in order to improve the learning experience. This section goes in-depth on the design choices of this newly refined application.

3.1 Requirements

The following key functional requirements can be generated from what the literature review has placed importance on. This is the baseline of what the newly refined application has to exhibit.

ID	Requirement	Reference
#1	The application generally follows a routine-based learning procedure on a regular basis for users.	• Sections 2.1, 2.2 • Ref [5], [6]
#2	The learner must get instant and constructive feedback after a study/training session.	• Sections 2.2 • Ref [6]
#3	The application shows users a comprehensive view of their progress of their mastery over the variety of distinct topics.	• Sections 2.6, 2.5 • Ref [23], [22]
#4	The application must provide a social & community aspect.	• Sections 2.5, 2.4 • Ref [21], [20]
#5	The quizzing has to be similar to the format of an actual Driving Theory test.	• Section 2.3 • Ref [14], [15], [16]
#6	The application must reward the user for their study progress through Gamification elements such as Levels, XP and Achievements.	• Section 2.4 • Ref [18], [19]

ID	Requirement	Reference
#7	The application must allow the user to study on specific topics.	• Section 2.5 • Ref [23]
#8	Application must support Multimedia such as Text, images, and videos.	• Section 2.3 • Ref [14], [15], [16]

3.2 User Understanding

Development will follow the Agile development approach [33]. To start off, analysing user personas and stakeholders is essential to the plan of this application in order to simulate the driving theory learning experience. These are important to identify to ensure the app is user-friendly, effective in knowledge retention, and aligned with stakeholder expectations.

Primary Stakeholder:

- **Category B license learner**: an individual preparing to obtain a standard passenger vehicle (car) driving license. They are first-time drivers who have secured their provisional license. However they need to revise for and pass the Driving Theory exam, before they can take on the practical test.

User Persona:

Novice Learner
Tristan is an 18-year-old student who has recently begun his journey toward becoming a licensed driver. He is eager to gain his Category B license, as driving represents a major step towards his adulthood. However, as a novice learner, Tristan finds certain aspects of the driving theory exam particularly challenging. Topics such as road signs, hazard perception, right-of-way rules, and defensive driving strategies can sometimes feel overwhelming.

Tristan is highly motivated by the independence that comes with obtaining a driver's license, but he prefers an interactive and engaging learning approach rather than dry textbook study. He is looking for a learning experience that is structured, easy to follow, and effective in helping him retain key driving knowledge.

3.3 User Stories

User stories encapsulate the functional requirements and expectations of our stakeholders, aligning with the INVEST criteria [34]. Story number codes will be referred to in subsequent sections. Each user story is been given a number ID for easier referencing, and a nickname for quick reminder of the story's content.

ID	Nickname	User Story
#1	Daily Quiz	**As a** user, **I want** to receive daily quizzes on Driving Theory overall, **so that** I can maintain my retention of the material.
#2	Streaks	**As a** user, **I want** to keep track of my studies in check with a streak counter, **so that** I don't forget my learning routine.
#3	Quiz Category Breakdown	**As a** user, **I want** get a breakdown of my performance by question category after daily quizzes, **so that** I know what went wrong in that quiz and remember for next time.
#4	Quiz Scores	**As a** user, **I want** to receive instant results after completing a quiz, **so that** I can immediately know how well I performed.
#5	In-depth Reasons	**As a** user, **I want** to see explanations for each correct and incorrect answer, **so that** I can understand the reasoning behind the answer.
#6	Motivate	**As a** user, **I want** to receive positive reinforcement or encouragement, **so that** I stay motivated even if I make mistakes.
#7	Visual Progress	**As a** user, **I want** to view my learning progress in the form of graphs or charts, **so that** I can quickly understand my strengths and weaknesses.
#8	Mastery Level	**As a** user, **I want** to see a mastery level (e.g., Beginner, Intermediate, Expert) for each topic, **so that** I can track my improvement over time.
#9	Ready for Exam	**As a** user, **I want** to see an estimated score that predicts my readiness for the real driving test, **so that** I know when I am prepared to take the exam.
#10	Friends	**As a** user, **I want** to add friends and connect with other learners, **so that** I can share my progress and learn together.
#11	Leaderboard	**As a** user, **I want** to see my ranking compared to my friends and other learners, **so that** I stay motivated to improve my scores and engagement.
#12	Challenge	**As a** user, **I want** to challenge my friends to driving theory quizzes, **so that** I can make learning more fun and competitive.
#13	Mock Exam	**As a** user, **I want** to have mock exams that follow the actual exam format, **so that** I can get used to the actual exam structure.
#14	Multiple Choice	**As a** user, **I want** to practice with the multiple choice question format, **so that** I can get used to the first section of the exam structure.
#15	Hazard Perception	**As a** user, **I want** to practice hazard perception tests with video-based scenarios, **so that** I can get used to the second section of the exam structure.
#16	EXP System	**As a** user, **I want** earn EXP for completing lessons, quizzes, and challenges, **so that** I feel rewarded for my learning progress.

ID	Nickname	User Story
#17	Levels	**As a** user, **I want** level up based on the EXP I earn, **so that** I can track my progress and feel motivated to reach higher levels.
#18	Achievements	**As a** user, **I want** unlock badges for milestones like streaks, high scores, or completing topics, **so that** I get a sense of accomplishment.
#19	Topic Quizzes	**As a** user, **I want** take quizzes focused only on the topic I'm currently studying, **so that** I can reinforce my knowledge in specific areas.
#20	Recommend Topic	**As a** user, **I want** receive recommendations for topics to study based on my performance, **so that** I can prioritize areas where I need more practice.

3.4 Product Backlog and Increment Plan

This Product Backlog shows the priorities of the features and user stories in Agile Development. Each user story has been referenced using it's **Number ID** and **Nickname**.

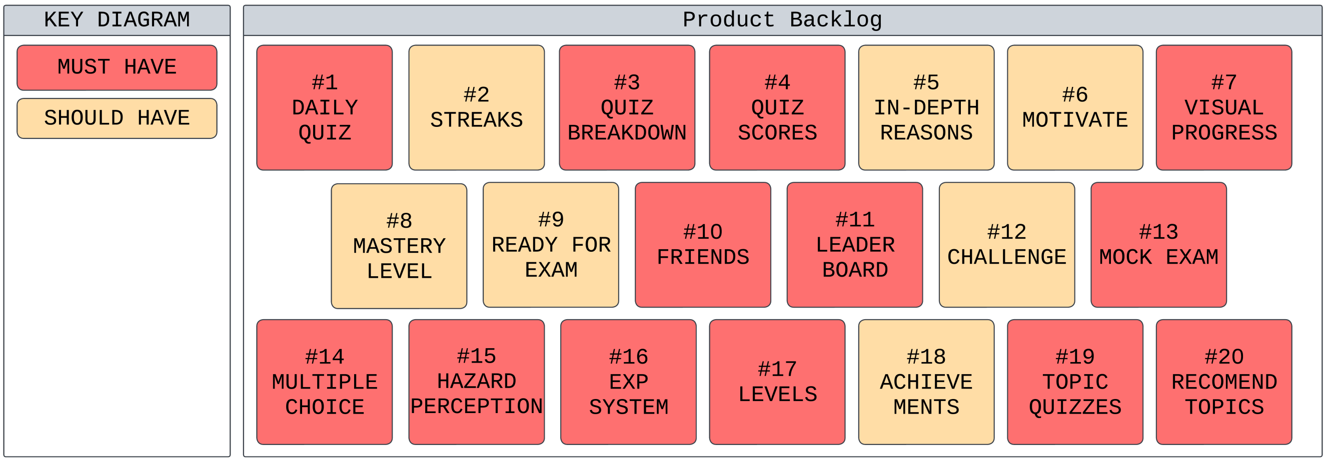

Figure 3.1: Product of the application

The iteration plan serves as a structured roadmap for iterative development, focusing on incremental delivery of key objectives with deadlines. The workloads are indicated by **T-Shirt sizes (S-XL)**, with **XL** being user stories with the largest workload and **S** being the smallest.

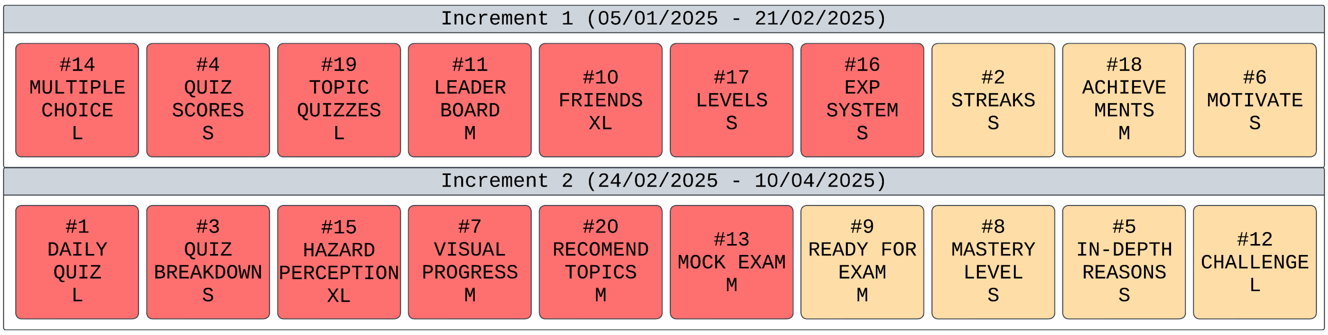

Figure 3.2: Product of the application

3.5 Design

During the early stages of development, a rough model was created to lay the groundwork for the overall structure of the application. The application consists of client-server interaction. The client (user interface) sends requests to the server and updates the UI according to any changes.

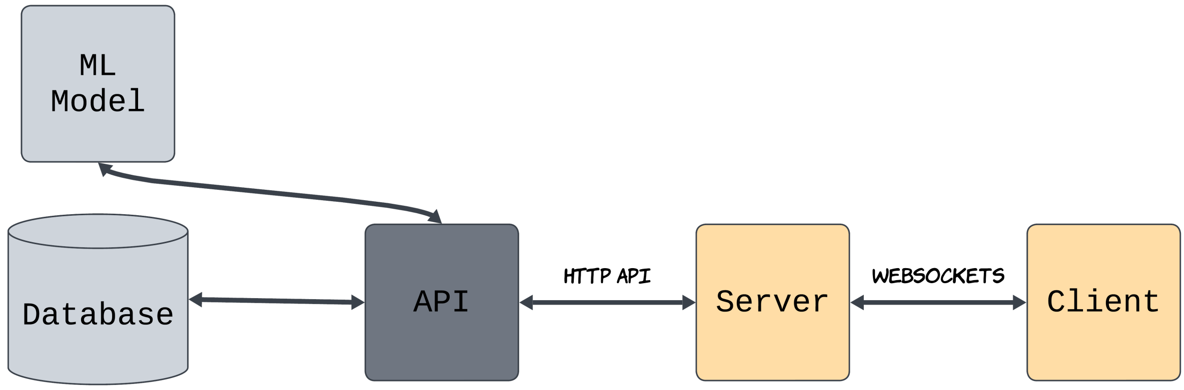

Figure 3.3: Rough model of the application

In addition, the server acts as a bridge between user requests and back-end services, including the API that integrates both the machine learning model and the database. The machine learning model is designed to evaluate user performance through scores and recommend what topics need to be worked on. Meanwhile, the database efficiently stores user data, the questions, and learning history to enhance the learning experience.

3.5.1 Tools

Tool	Details
Visual Studio Code	• The development environment where the whole application is developed in as it's extensions support the following tools.
Microsoft Azure Services	• Used for the back-end and database API services. • **Function App** is used for the API *python* functions. • **CosmosDB** databases are used for fetching user and question data.
VueJS	• Static front-end views are coded in VueJS. • The server logic is handled and coded in Javascript. • API calls are also implemented here.
Numpy Python Library	• Used to code the Machine Learning Model and has the Random Forest Regressor method to be used. • Makes the calculations needed to predict and evaluate a user's performance.

3.5.2 Class Diagram

In the next phase of development, a Class Diagram was devised to structure the driving theory learning application using the Model-View-Controller (MVC) design pattern [35] as shown in figure 3.4. The Model handles the back-end functionality, including data management, user information, and machine learning model integration. The View presents the front-end functionality, displaying the application as a whole, while the Controller mediates between the Model and View, processing user input such as the api function calls and updating the interface accordingly.

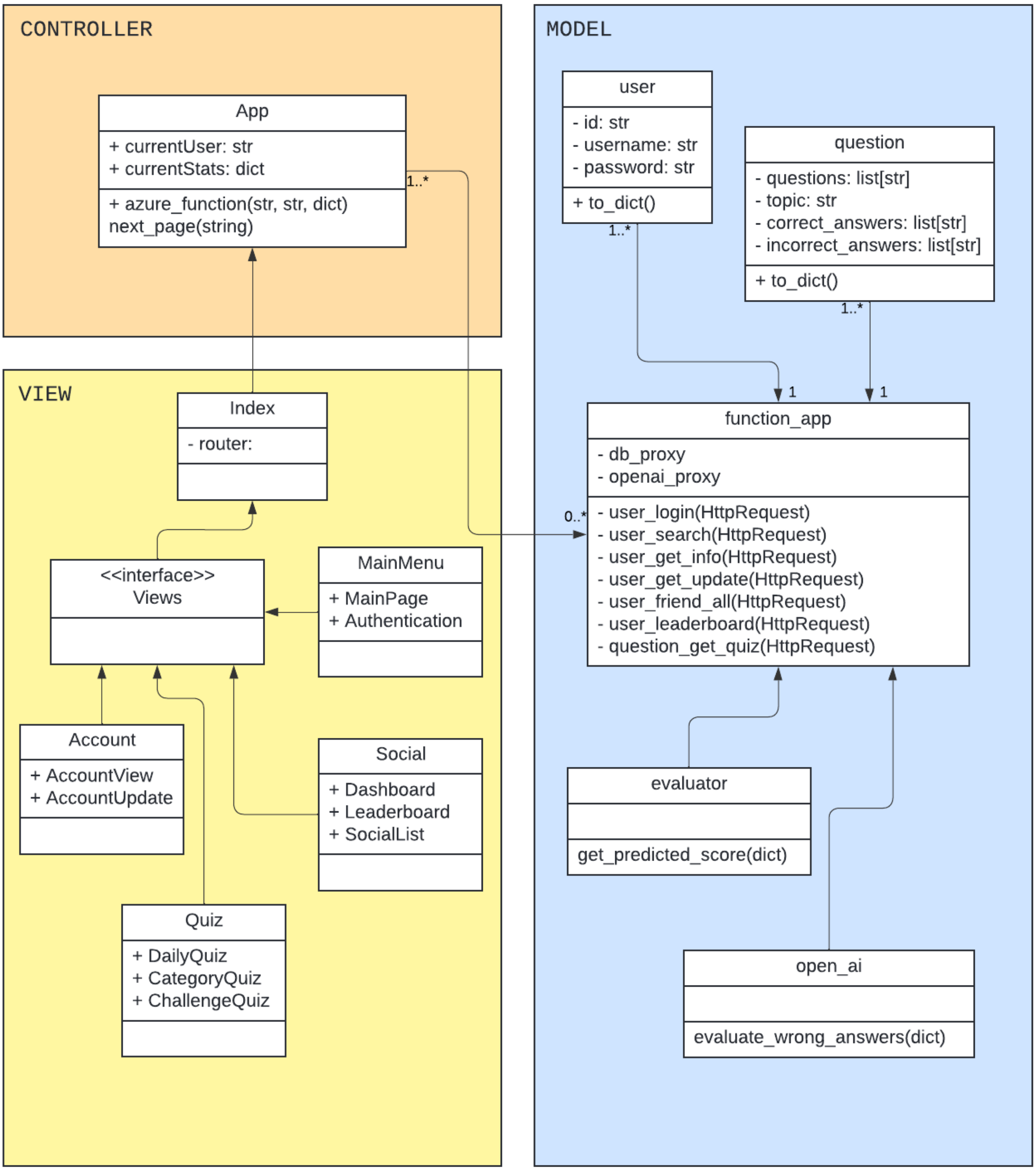

Figure 3.4: Class Diagram for whole application

3.5.3 Activity Diagrams

These diagrams show how the application and user interact in various scenarios.

Category Quiz

The user selects a category and answers a series of questions. Their score is based on accuracy and response time, with a maximum of one minute and a half allowed per question in the real exam.

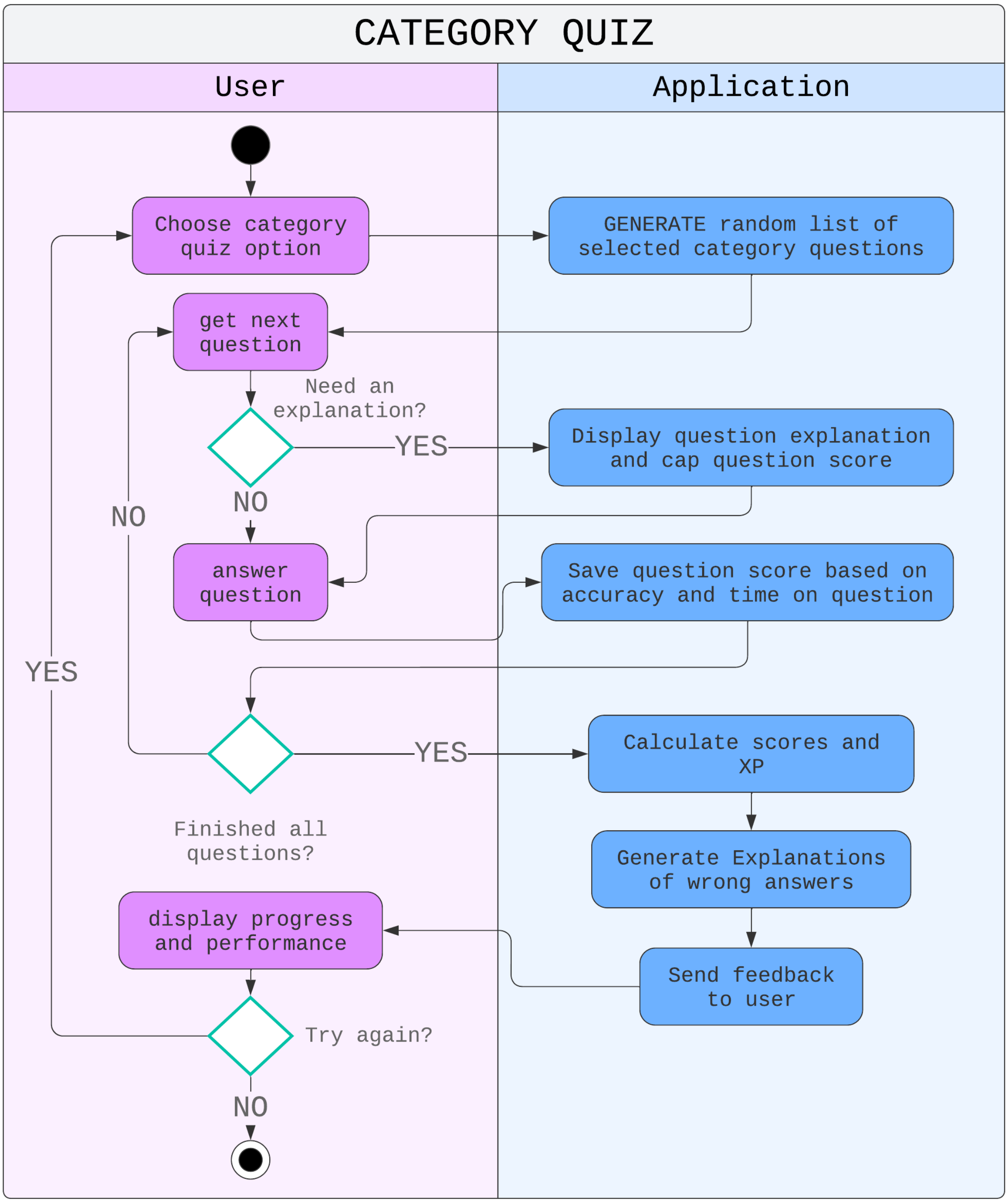

Figure 3.5: Sequence Diagram for Category Quiz

Hazard Perception Practice

The user selects a category and a video to practice on. The app plays a hazard perception video, allowing a maximum of 12 clicks. The score is determined by the accuracy and timing of their clicks.

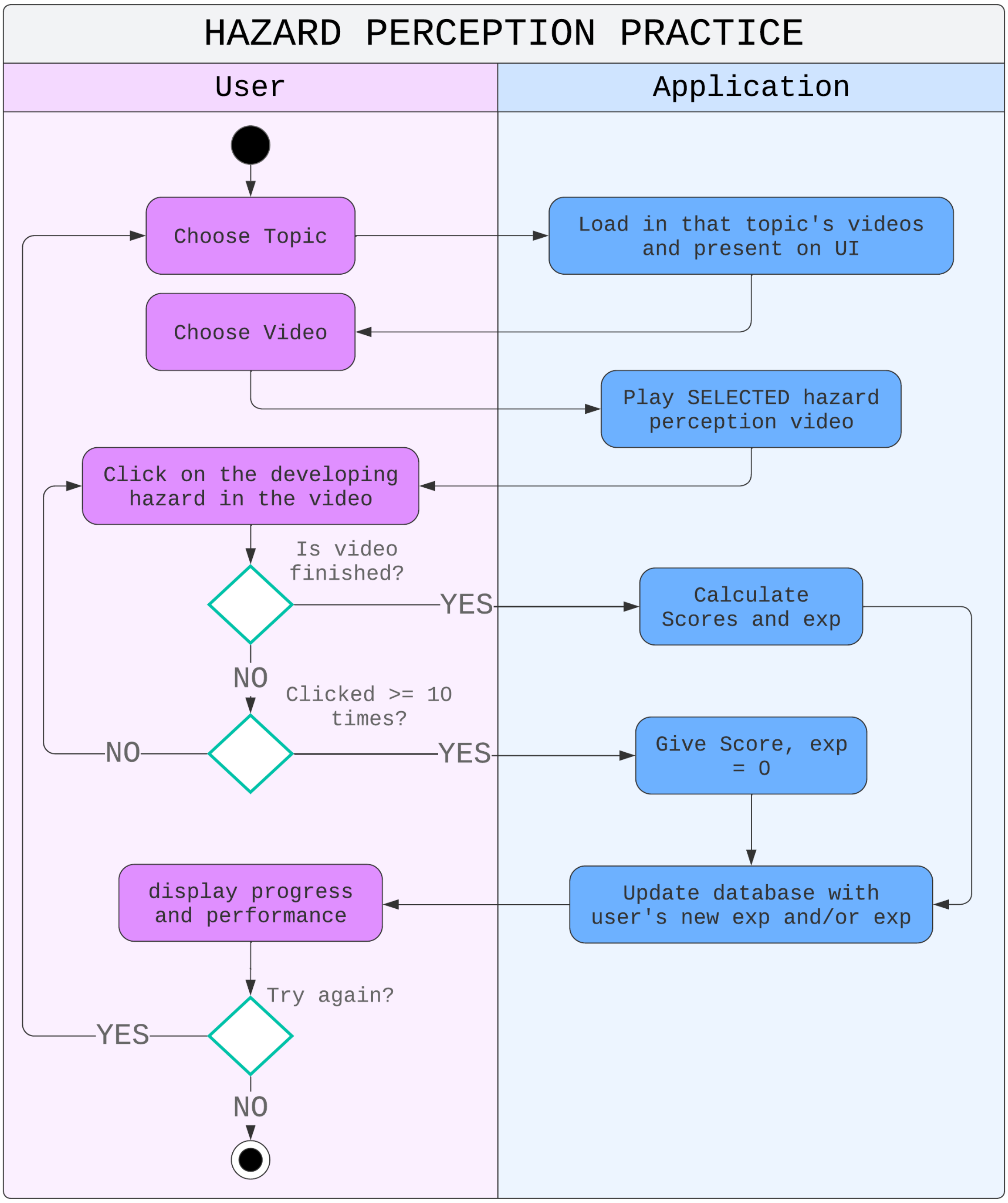

Figure 3.6: Sequence Diagram for Hazard Perception

3.6 Implementation and Testing

Aligning with the agile approach, various components must be developed in a specific order, as some are interdependent while others influence each other. For a detailed breakdown, please refer to the product backlog in Section 3.4. The entire development process will be test-driven, with both the back-end and front-end being systematically developed.

3.6.1 Acceptance Criteria

Firstly, these acceptance criteria required to confirm that the user story requirements have been successfully implemented. Each criterion provides clear, step-by-step validation to ensure that all functionalities meet user needs and objectives. Below are the main user stories that must be fulfilled as examples, and refer to the supporting documents for all acceptance criteria.

ID	Nickname	Acceptance Criteria
#1	Daily Quiz	• Learner can select the DAILY QUIZ option upon login. • On selection, the learner is given instructions of the quiz. • Learner is given a question set such that it covers **general multiple choice questions**, **road signs** and **hazard perception.** • After partaking the quiz, the learner is given feedback on incorrect answers. • The learner gains exp and increments the streak counter, all updated in the database.
#10	Friends	• Learner can search for users by username. • Upon finding usernames, the user can view their profile with their current stats. • The user can send a friend request to a selected user. • That user who has received the friend request will be notified. • The recipient user can either reject or accept the friend request. • Now the new friends can see each other in the friend leaderboards.
#12	Challenge	• Learner can select the VERSUS option upon login. • Upon selection, the learner can either join a lobby or host one. • Users can connect through the same password. • The host can start a game once another player has joined. • Both users are put against a series of questions from the backend API. • Users that answer the quickest and most correct will gain one chance whilst the other user loses one. • The user that loses all their chances will lose and hence the game will end. • Each user gains exp from completing a game.

3.6.2 Back-end

Ensuring a stable and functional back-end was a key priority in the first development increment. Since the application relies on core processes like login and registration, thorough Unit testing was conducted early to identify and resolve potential issues. The testing is more white-box focussed as it's the inner calculations of the application.

Feature	**Testing**
Function App Azure Functions	• Each API function has a python unit test with their own cases, which are done with JUnit.
Azure CosmosDB	• The API functions each manipulate the database, so also tested in JUnit. • Proxy Objects are used to give the database temporary states to test specific situations for each API function.
Machine Learning Model	• Also tested in JUnit and given Mock User data to test with.

3.6.3 Front-end

Once the back-end was successfully deployed and fully operational, the development process could shift focus to the front-end. With a stable foundation in place, this allowed for a more structured and efficient workflow, enabling an intuitive and responsive user interface while leveraging the robust capabilities of the back-end. The testing for the UI is more black-box focussed as it's the user interface side of the project, whilst the testing for the network communication is white-box focussed.

Feature	**Testing**
Server-Client interactions	• Logging and the UI was used to see the state of the server and check that's changing accordingly.
Server-API interactions	• There are messages and logs that pop up based on API responses to verify the deployment is working correctly.
Client UI	• VueJS DevTools was used to check certain states of the front-end model and debugging.

With each component now added from within both parts of the application, integration testing was used especially during the development of the front-end. This is to ensure that the application doesn't introduce any strange behaviours for the testers during the evaluation.

Chapter 4

Evaluation

The third research objective was to evaluate user feedback on the driving theory learner app game, particularly among individuals holding provisional licenses. The focus is to gather both Quantitative and Qualitative data to assess the impact, therefore a combination of Quantitative and Qualitative Research Methods [36] is required.

Several techniques can be utilised for evaluation, including interviews, questionnaires and group work [36]. However for this research, an anonymous questionnaire was chosen as the preferred method. The questionnaire was distributed after user testers had interacted with the prototype for several days, as detailed in Appendix B.

Following approval from ERGO **(ID: ERGO78677.A2)**, the questionnaire was officially released on 17/04/25 on google forums. Between 17/04/25 and 24/04/25, a total of 12 responses were collected, of which 12 were fully completed.

The following graphs represent feedback on key questions addressing the factors contributing to a poor learning experience and stakeholders. The responses were collected from anonymous users who tested the app and provided their input based on how the app performed.

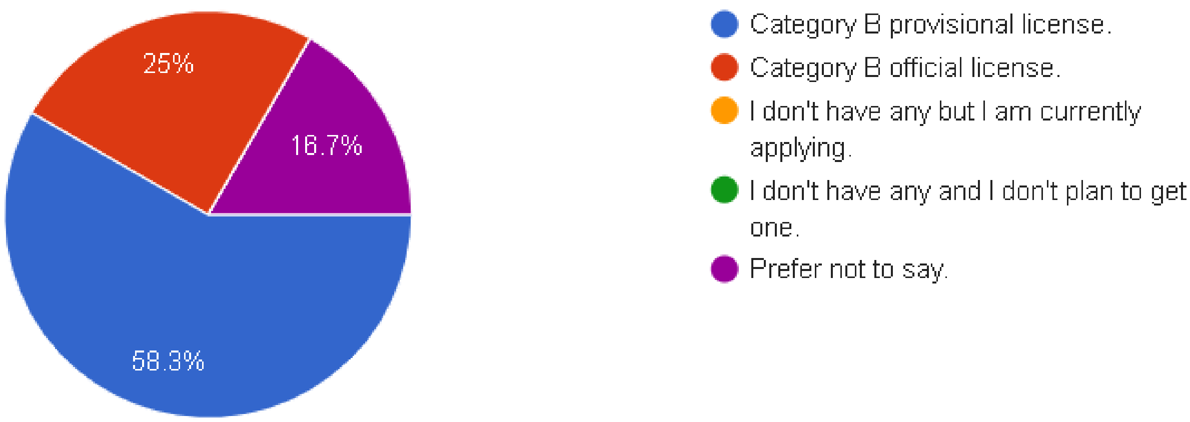

Figure 4.1: Demographic Distribution of Users by License Acquisition Status

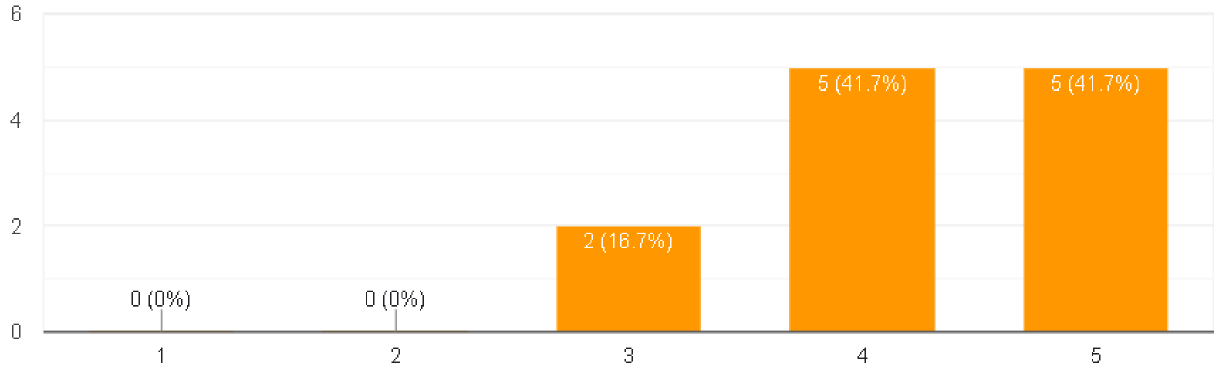

Figure 4.2: "The app helped me identify areas where I needed improvement."

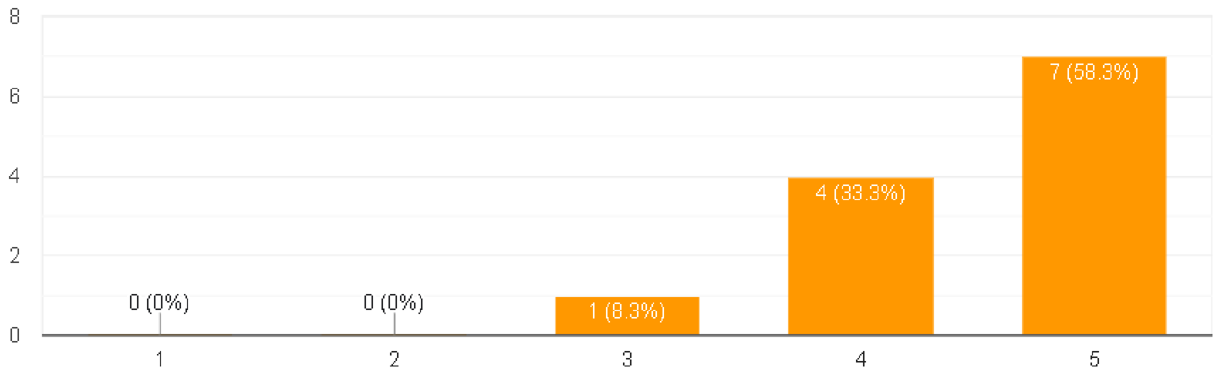

Figure 4.3: "The app helped me retain information about driving theory."

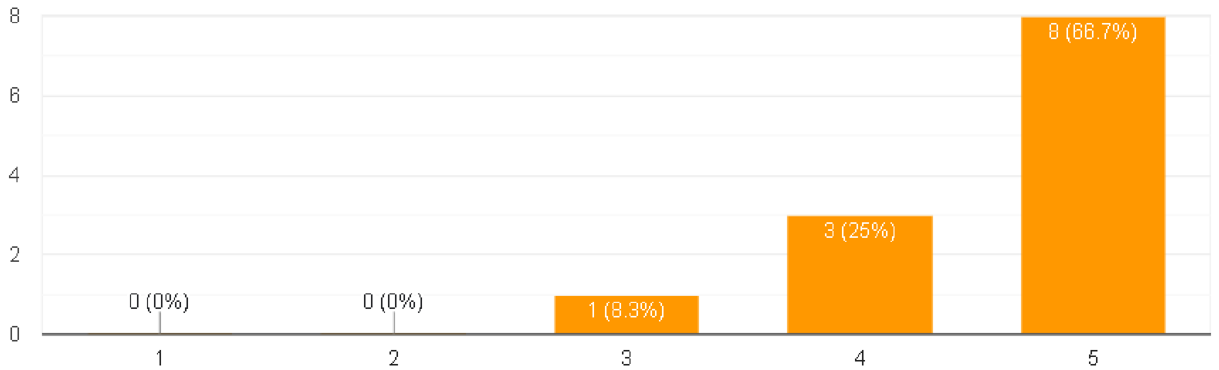

Figure 4.4: "I would recommend this app to my friends."

The main feedback presented in Figures 4.2, 4.3 and 4.4, demonstrates a positive reception towards the app's educational effectiveness. A large proportion of users agreed with the statement "The app was effective in helping me understand driving theory concepts," indicating that the app provides clear explanations of key material. Additionally, users commented that the app helped to identify weaker topics and in retaining information but some of them wished for a tutorial for the mechanics, particularly hazard perception. The feedback demonstrates that the app effectively addresses the identified issues related to user motivation and information retention, by offering an engaging learning environment. Overall, the positive responses show an improvement in the driving theory learning experience.

4.1 Reflection

The development and evaluation of the driving theory learner application demonstrated promising results, though areas for further improvement was identified. The general reception of the application was positive overall, with users commenting on its engaging, informative, and effective contribution to the learning process. It tackled the key learning challenges, notably motivation, retention, and the temporal issues for a learning app. Ultimately, the final product delivered its core objectives.

However, the project also ran into a number of difficulties. A significant issue was the time-consuming process of writing the theory-based questions, which demanded considerable effort and prolonged the development period. In addition, there were technical setbacks during the deployment phase. Server restrictions on Google App Engine limited the availability of the website, resulting in temporary accessibility issues that affected testing and demonstration phases.

Another notable challenge involved the integration of OpenAI models to generate personalised feedback for incorrect answers. While this initially added a valuable layer of interactivity, the models became unreliable due to service interruptions during the final stages of the project. Furthermore, high usage during the testing period rapidly depleted allocated API credits, and budget constraints made it unsustainable to continue using this feature. As a result, this component had to be removed and replaced with manually written feedback. Although effective, this alternative approach further increased the time spent on question writing.

4.2 Risk Analysis

The risk analysis below identifies risks related to the application, including the Probability (P), Severity (S), and Risk Exposure (E), calculated as P x S. Probability ranges from 1 (lowest) to 5 (highest), while Severity ranges from 1 (least severe) to 5 (most severe). The analysis also includes mitigation for each risk.

Risk	**P**	**S**	**E**	**Mitigation**
OpenAI model service retires during development	3	5	15	Ensure other models are still available and test them and have them on standby as a substitute model.
High usage of subscription credits	4	5	20	Minimise usage as much as possible to avoid using up all the credits.
Deployment fails	2	5	10	Test the deployment on a simple page before development on the front-end code, to ensure it works.

Table 4.1: Risk Analysis Table for driving theory learner application

Chapter 5

Conclusions and Future work

In conclusion, this driving theory learner app was developed with the goal of enhancing the learning experience by improving upon existing methods, addressing the problems identified in the literature review. Overall, user feedback was relatively positive, indicating that the app met its objectives of providing an improved learning experience. However, some setbacks on technical and user experience arose. Despite these setbacks, the app was well-received and has shown potential for further development and refinement to better meet the needs of learners.

Research objective 1 aimed to identify factors why the current learning experience is poor. This was achieved through the literature review, where existing research was explored on driving theory learning. This review emphasized that the core issues hindering effective learning were the decay of memory retention and a subsequent drop in motivation. Learners lose motivation to study when they lose previously learnt material, which results in disengagement and a drop in performance. The aforementioned observations highlighted the need for an alternative teaching approach that can address these core issues.

Research Objective 2 required to create an application as a tool for driving theory revision. Based on the insights from the literature review, the second objective was to develop an application that could serve as a tool for driving theory revision. The application used gamification features like existing learner apps, including quizzes, progress monitoring, and achievements. However, to tackle the temporality issues, intentional game design choices have been made. Examples include using randomised quizzes and varied question production to increase replay value. This enabled users to actively repeat and recall driving theory, creating a more engaging learning cycle.

Objective 3 was to evaluate how effective this is with a questionnaire. Feedback gathered from users who tested the prototype, especially those holding provisional licenses, was positive overall, as illustrated in the accompanying graphs. Participants indicated that the application aided their comprehension of crucial driving theory concepts, facilitated the identification of areas needing further attention. The generally favourable reception of the application demonstrated its success in enriching the learning experience, thereby confirming its effectiveness.

Despite these challenges, the project's main objectives were effectively met, and the app provides a strong foundation for future development. In order to improve the learning experience further, user feedback asked for more detailed explanations of specific app mechanics, such as hazard perception. However, due to time constraints and the app's already substantial size, these additional demonstrations could not be implemented within the current scope.

Moving forward, addressing limitations such as content scalability, ensuring stable deployment infrastructure, and implementing cost-effective feedback generation will be critical for enhancing the app. By improving these areas, future versions could take user feedback into account while preserving the functionality and performance of the app, giving users a more thorough and interesting learning experience.