IT2010 – Mobile Application Development
BSc (Hons) in Information Technology
2nd Year
Faculty of Computing
SLIIT

2023 – Assessment 01
Phase 2 Report

Description	Student ID	Name
Group Leader	IT21228094	Mendis A.R.P.
Member 2	IT21239366	Niwantha W.P.I.
Member 3	IT21225574	Jayasinghe J.A.J.M.
Member 4	IT21231100	Sandaruwan W.M.I.M.

Application Topic	Agarwood plant schedule inquiries and purchase and sales management- Agarwood OudCraft
Group Name	Agarwood OudCraft_ IT_WD_ 03_01

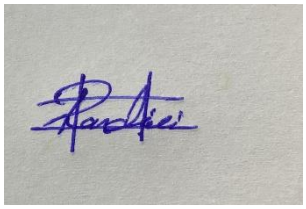
Project Declaration

We, the members of Agarwood OudCraft_ IT_WD_ 03_01, hereby declare that our group project is entirely authentic and original. We have conducted thorough research and analysis to ensure that our work is not plagiarized or copied from any other sources.

We have followed all guidelines provided by our LIC and have complied with all ethical and academic standards.

We take full responsibility for the authenticity of our work and understand the implications of academic dishonesty. We have worked collaboratively to produce this project, and each member has contributed to the best of their abilities.

We hereby affirm that our project represents our honest effort and commitment to academic integrity, and we take pride in presenting it as our own.



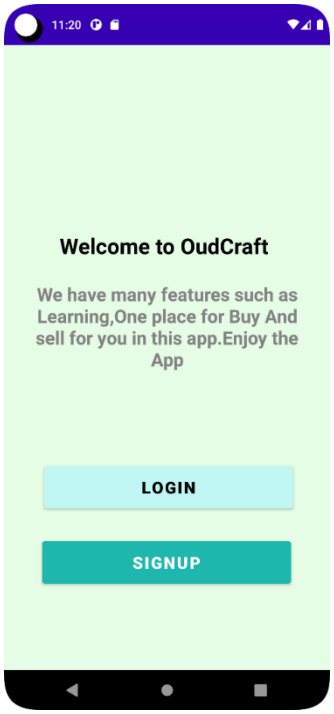
Group Leader (Signature)
Mendis A.R.P.

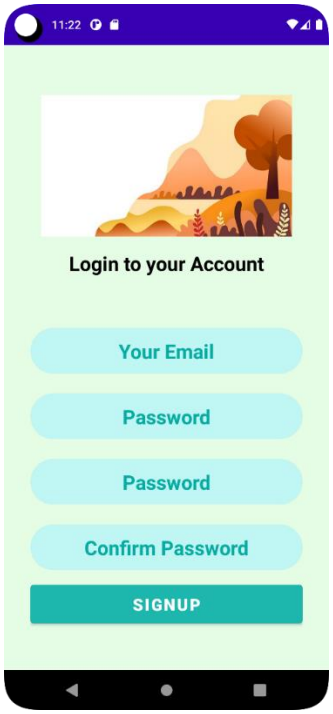
Git Repository: [randinimendis/Agarwood-OudCraft_-IT_WD_-03_01 (github.com)]

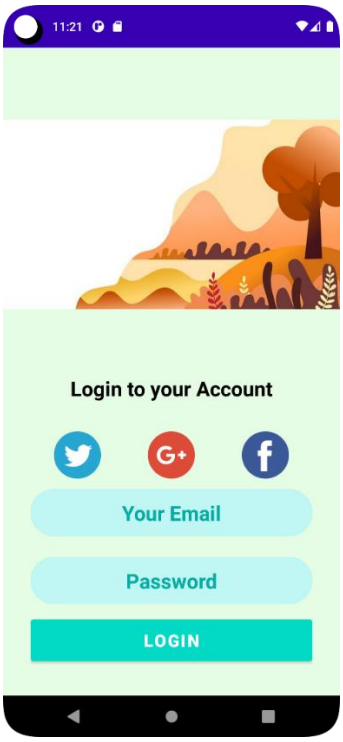
Screen shots of the application (Member wise)

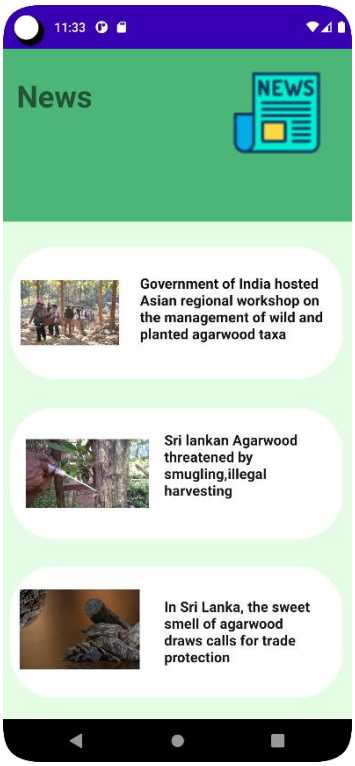
- Please note that the images should be captured after running the app on the Android Emulator or on the physical device.

(IT21239366| Login page ,signup, dashboard page)

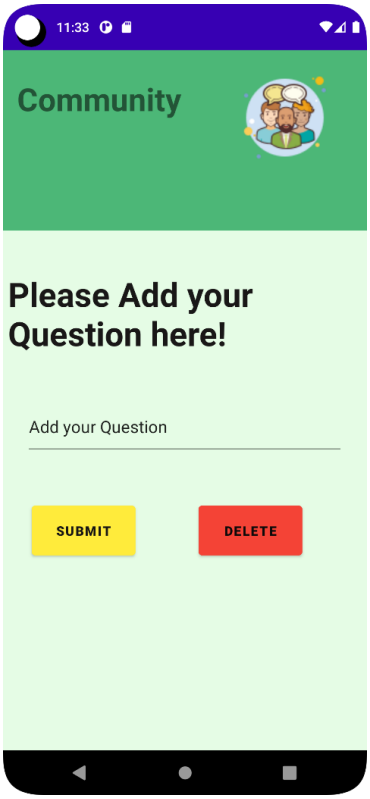


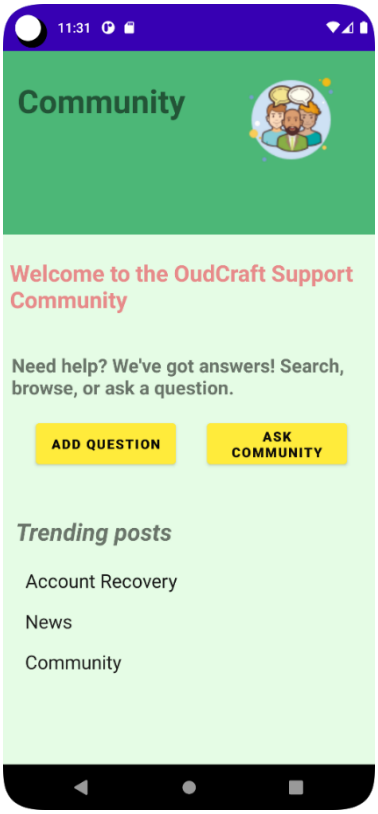




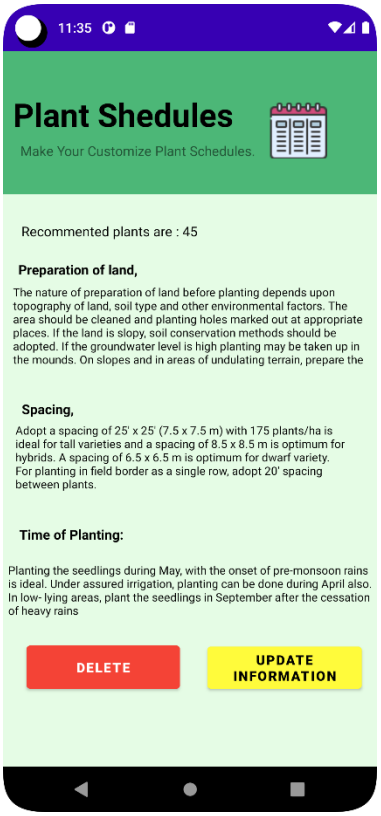


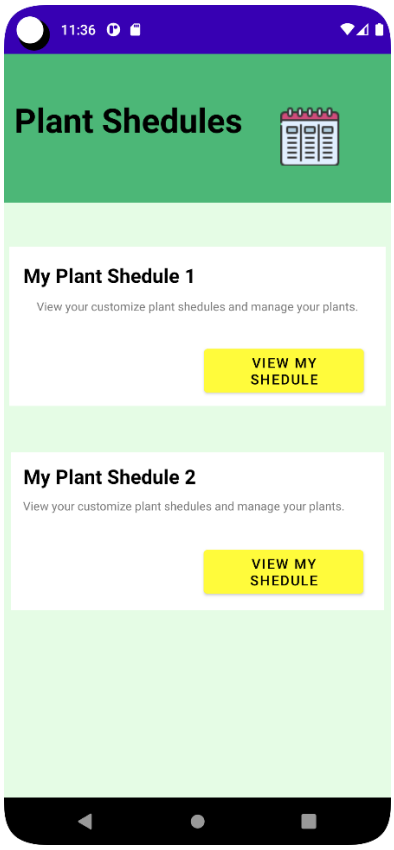
IT21239366 – Community

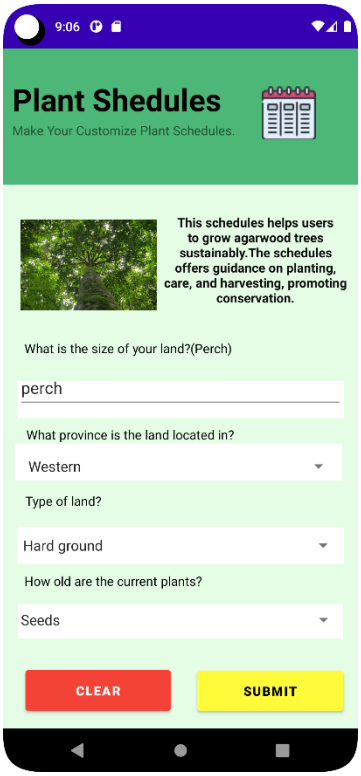




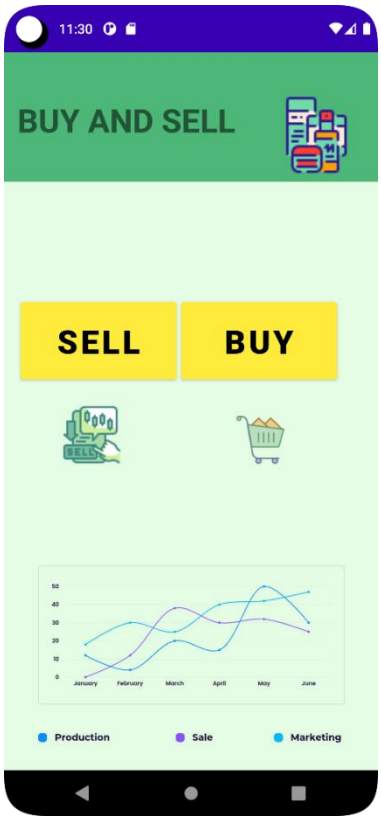
IT21225574 – Customized plant schedules

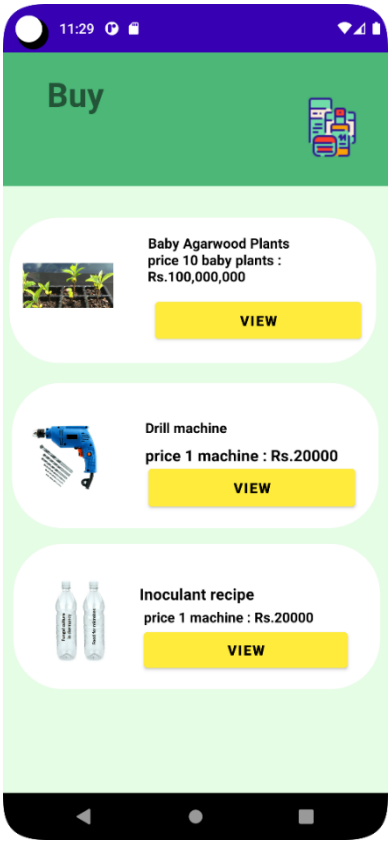


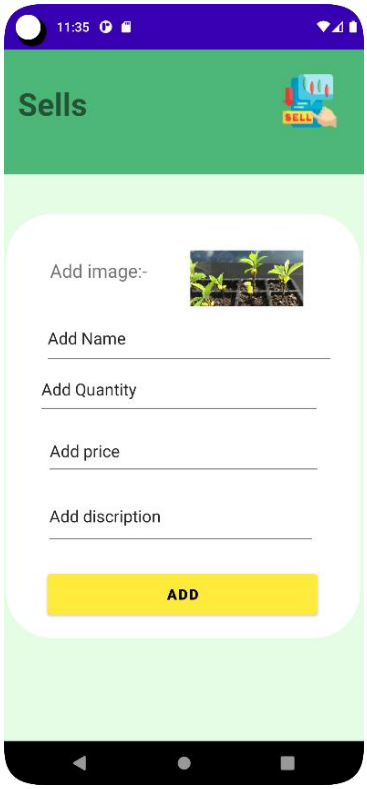


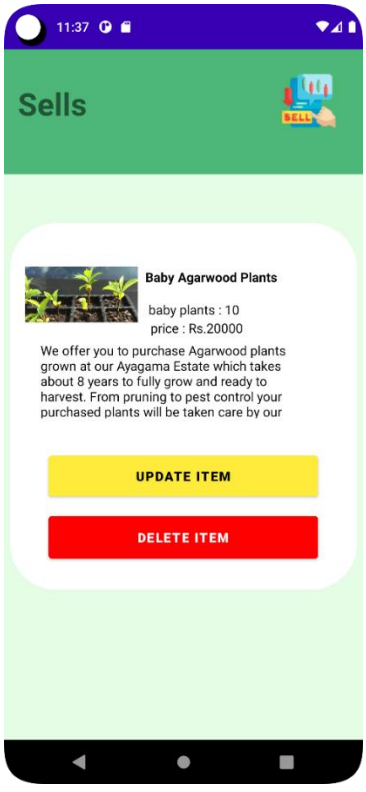


IT21228094-BUY & SELL

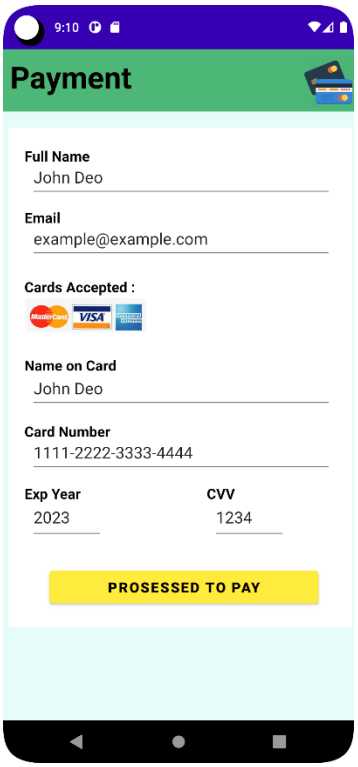


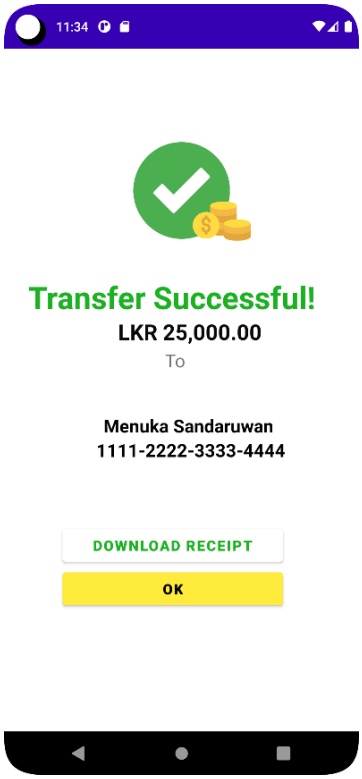




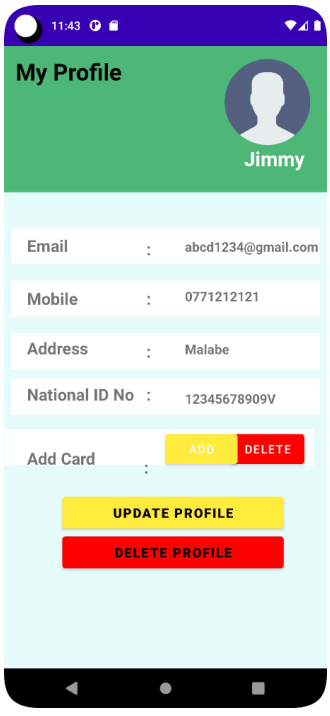


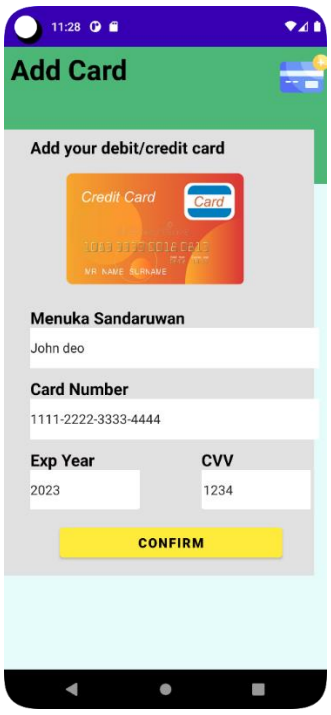
IT21231100- Buy (sub category payment gateway)





MY profile & (sub category add card)





Member Wise Descriptions About What Is Their CRUD Operations

IT21239366 – Community

The community feature in an Android app is designed to create a space where users can engage with each other and share their experiences, opinions, and ideas related to the app's content or services. This feature can include a range of tools and functionalities, such as discussion forums, messaging systems, user profiles, and social sharing options. The community feature serves several purposes. Firstly, it can help foster a sense of belonging among app users, creating a community around a shared interest or goal. This can lead to increased engagement and loyalty towards the app, as users feel that they are part of a larger community that shares their values or interests.

Secondly, the community feature can help app developers gain insights into their users' needs and preferences. By monitoring user activity and engagement within the community feature, developers can identify popular topics, frequently asked questions, and areas of confusion or dissatisfaction. This information can then be used to make improvements to the app's content, features, or user interface, resulting in a better user experience overall.

Finally, the community feature can be a valuable tool for user support and feedback. Users can use the community to ask questions, report issues or bugs, and offer suggestions for improvement. This can help developers identify and address problems quickly, as well as prioritize feature requests based on user demand.

Overall, the community feature is an essential part of many successful Android apps, providing users with a platform to connect with one another, share their thoughts and ideas, and contribute to the ongoing development and improvement of the app.

IT21225574 – Customized plant schedules

The create operation on the Customize Plant Schedule page allows users to create customized planting schedules for their agarwood plants. The user will fill out a form that requests information such as the size of the land in perch, the province in which the land is located, the type of land, and the age of the current plants. Once the user submits this information, the application will create a new schedule based on the data entered.

The read operation on the Customize Plant Schedule page allows users to view their customized agarwood plant schedules. Once a user has entered their information and created a schedule, the application will generate a customized plant schedule by doing calculations based on the user's input. The user can then read the generated schedule and plan their planting accordingly.

The update operation on the Customize Plant Schedule page allows users to update their previously created schedules as needed. If the user encounters a situation where they need to change the information they entered, they can do so by clicking the "Update Schedule" button. The user will then be able to modify the entered information and save the updated schedule.

The delete operation on the Customize Plant Schedule page allows users to remove any previously created schedules that are no longer needed. If a user wishes to delete a schedule, they can do so by clicking the "Delete Schedule" button. The application will then prompt the user to confirm the deletion before removing the schedule from the system.

IT21228094-Buy/Sell

The app features several CRUD operations that enable sellers to manage their selling products and customers to view and purchase products.

Firstly, to create a new selling product, sellers can use the "Add Item" button on the selling page. This button prompts sellers to enter the item name, quantity, and price of the product. Once the information is entered, the system adds the new product to the list of selling products. This feature ensures that sellers can easily add their products to the platform and make them available for purchase by customers. (Create)

Secondly, the "Buy" page of the app allows customers to view each selling product and its details, such as product description, price, and quantity. Customers can easily browse through the list of selling products and choose the product they want to buy. This feature allows customers to make an informed decision about which product they want to purchase and ensures that the buying process is easy and straightforward. (View)

In the "Sell" page of the app, sellers can update the product details of their selling products. The "Update Selling Product Details" button enables sellers to change the product description, price, and quantity of their products. This feature ensures that the selling products remain up-to-date and relevant to customers. Sellers can make changes to their products whenever necessary, ensuring that their products remain competitive in the market. (update)

Finally, the "Delete Product" button allows sellers to remove any selling product that is no longer available or relevant. This feature ensures that the list of selling products remains accurate and up-to-date for customers. Sellers can easily remove products that are no longer available or that are no longer relevant to the market. This feature ensures that customers only see available products and that the list of selling products is always accurate. (delete)

In conclusion, the Buy and Sell CRUD Operations in the Agarwood Development Mobile App provide an efficient and effective way for sellers to manage their selling products and for customers to view and purchase Agarwood products.

IT21231100- Buy (Subcategory Payment), MY Profile & (Subcategory Add Card)

In our project Android app, “Agarwood oudcraft”, My part is to build up user profile, Add card and payment. Here I have implemented the profile page to view the users profile details which are given to the system when the user is registering. Here I have given a button to the user if he/she wants to delete their profile also another button to update the details. In the profile page I have implemented an extra advantage to the customer to add his banking card. Then the user doesn’t need to enter the card details when he/she going to purchase something through the app. When the user clicks the buy button on the item page the card details will fill automatically. The other part which is implemented by me is the payment part. Here when the user clicks the buy button on each item page. The user redirects to the payment page. Here if the user has already saved the card details and add it to the system the card details will be filled automatically. If not, the user must fill the ask details about the banking card. Such as Card number, name on card Exp Year and CVV. After the user clicks the processed to pay button the payment will be done. If not, the system will pop up an error message. When the payment is done, I have implemented and successful message to view to the user to be confirmed that the payment is successfully done. Here I have given an advantage to the user to download this successful message as a pdf. Because if there is any issue with the user the user can sent to us by confirming his/her transaction.

Member Wise Descriptions About What Is Their Testing Methods,

IT21239366 – Community

Here's an example of how I might use Unit Tests/ Instrumented testing to ensure that the community/feedback feature in our Android app is functioning properly:

Unit Tests:

•Test that the user profile is properly created and includes the necessary information (name, profile picture, etc.).

•Test that users can create new posts on the discussion board and that these posts are properly saved in the app's database.

•Test that users can vote on content and that the voting system properly updates the app's database.

•Test that users receive notifications when new content is posted or when someone responds to their posts or comments.

Instrumented testing:

•Test that the discussion board displays properly on different screen sizes and device types.

•Test that the notification system works properly on different Android versions.

•Test that the app's database properly saves and retrieves user information, posts, and voting data.

By using these testing methods, we can ensure that the community/feedback feature in our Android app is functioning properly and providing a positive user experience.

IT21225574 – Customized plant schedules

Unit Testing:

Unit testing can be used to test each individual function or method of the CRUD operations. For example, a unit test could be created to ensure that the form data is being saved correctly when the user submits it.

Integration Testing:

Integration testing can be used to test how different parts of the application work together.an integration test could be created to ensure that the form data is being passed correctly to the backend when the user submits it.

IT21228094 -Buy/Sell

CRUD operations in an Agarwood Development Mobile App. Here are some of the test methods that could be suitably applied to test the CRUD operations:

Unit Testing:

Unit testing is a type of testing that verifies the functionality of individual code units. This test method is particularly useful for testing the Create, Read, Update, and Delete operations in the app. Unit tests can be used to ensure that each CRUD operation is performing as intended and that each operation is interacting correctly with the database.

Performance Testing:

Performance testing is a type of testing that verifies how the app performs under different loads and conditions. This test method is particularly useful for testing the Create and Read operations in the app. Performance tests can be used to ensure that the app can handle a large number of requests and that the response times are acceptable for customers.

IT21231100- Buy (Subcategory Payment), MY Profile & (Sub Category Add Card)

The profile update feature should allow users to edit their personal information, and changes should be reflected immediately throughout the app. Test for typical and edge cases to ensure the app handles unexpected input gracefully. The profile delete feature should permanently remove a user's account and all associated data. Display a confirmation dialog to prevent accidental deletions and test for edge cases such as offline deletion and errors during the deletion process. By testing these features thoroughly, you can ensure your users can effectively manage their personal information and that their data is handled securely. The "save bank card to the app" feature allows users to securely store their bank card information within the app. This information can then be used for future purchases within the app, saving the user time and effort. To implement this feature, a form can be created within the app where users can input their card information, such as the card number, expiration date, and security code. This information should be securely stored within the app, potentially using encryption or other security measures to protect the user's sensitive information. To test this feature, two types of tests can be used: unit tests and instrumented tests. Unit tests can be used to test the logic of the code responsible for saving the user's card information, ensuring that it correctly handles different scenarios (e.g. invalid card numbers, expiration dates in the past, etc.) and that it properly stores the information. Instrumented tests, on the other hand, can be used to test the user interface and user experience of the feature. This might involve simulating user interactions with the form for entering card information, verifying that the information is correctly displayed and saved, and testing edge cases Payment in-app feature allows users to make purchases within the app using their preferred payment method. This feature is commonly used in e-commerce, gaming, and other types of mobile applications that require users to pay for goods or services. To implement payment in-app, the developer needs to integrate a payment gateway or payment processor into the app. The payment gateway acts as an intermediary between the user, the app, and the payment processor. It allows users to make payments using different payment methods, such as credit cards, debit cards, digital wallets, or bank transfers. The payment gateway also ensures the security of the payment transaction by encrypting the payment data and using secure protocols to transmit it to the payment processor. Once the payment gateway is integrated into the app, developers can write unit tests and instrumented tests to ensure that the payment in-app feature is working as intended. Unit tests can verify the functionality of individual components of the payment feature, such as payment validation, payment processing, and error handling. Instrumented tests can test the integration between the payment feature and the app, including the user interface, payment flow, and payment confirmation.