1. There is a page where users can add a credit card as a payment method in Shopee (see picture below). Write a test case if you are responsible for testing the page

   Test Case 1: User successfully adds a credit/debit card payment method by manually type in card number
   Test Steps :
   1. Given user is at add credit/debit card page
   2. When user fills in **‘Nama Tertera di Kartu’** field as displayed name their in card
   3. And user fills in correct card number in ‘**No. Kartu’** field
   4. And user selects correct card provider in **‘Tipe Kartu’** field options
   5. And user fills in correct expiration date in **‘Tanggal Akhir Masa Berlaku’** field
   6. And user fills in correct CVV code in **‘CVV’** field
   7. And user fills in correct address in **‘Alamat Tagihan’** filed
   8. And user clicks **‘Kirimkan’** button
   9. Then verify that success toast is shown with message : **‘kartu berhasil ditambahkan’**
   10. And verify that user is able to see the additional card in my cards lists

   Test Case 2: User successfully adds a credit/debit card payment method by scanning card using camera
   Test Steps :
   1. Given user is at add credit/debit card page
   2. When user fills in **‘Nama Tertera di Kartu’** field as displayed name their in card
   3. And user clicks camera icon button in **‘No. Kartu’** field
   4. Then verify that user is shown a camera scanning page
   5. When user scan card number using camera
   6. Then verify that **‘No. Kartu’** field is auto filled with correct card number as displayed in card
   7. And user selects correct card provider in **‘Tipe Kartu’** field options
   8. And user fills in correct expiration date in **‘Tanggal Akhir Masa Berlaku’** field
   9. And user fills in correct CVV code in **‘CVV’** field
   10. And user fills in correct address in **‘Alamat Tagihan’** filed
   11. And user clicks **‘Kirimkan’** button
   12. Then verify that success toast is shown with message : **‘kartu berhasil ditambahkan’**
   13. And verify that user is able to see the additional card in my cards lists

2. In production, you found a bug where users can not add products to wishlist/favorite if they had 1000 product wishlisted before. From that case :
   a. Write a bug ticket reporting to your team (can use any format)
      Answer :
      Environment : Shoppe ID Production
      App Version : Android Mobile App, v4.1.5.78
      Issue Description : Users are not able to add products to wishlist if number of current products in wishlist equal to or exceed 1000 products.

Steps to Reproduce :
1. Given a User has 1000 Products in their wishlist
2. When User selects 1 more product to the wishlist
3. And User clicks ‘Add to Wishlist’ button
4. Then verify that added product is successfully added to wishlist

Expected Result :
Product is successfully added to wishlist and user is able to see newly added product on their wishlist

Actual Result :
An error message is shown after user click ‘Add to Wishlist’ button
<Insert Screenshot of the error message here>
<Insert request & response returned by BE API with the error message (result from inspect network request if any/applicable)> -> this to provide dev additional information about the issue, and for them to easily replicate the issue.

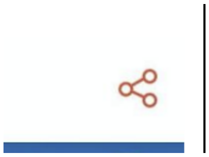
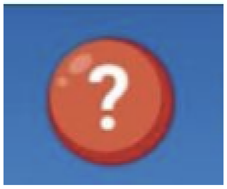
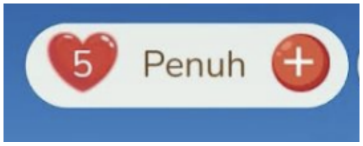
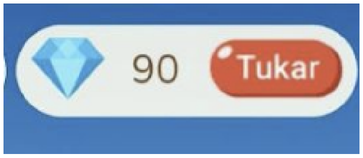
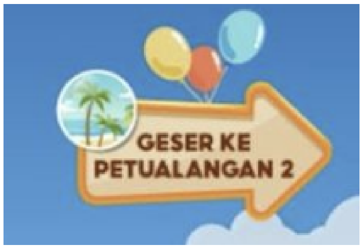
b. How do you validate (test) dev fixes in a non-production environment to make sure the ticket was fixed before deployment to production?
Answer :
First thing, I will test the case where a normal flow is done by the user to make sure normal flow/happy-path still working properly and no affected by the changes/fixes. Example as follow :
Test Case 1 : User successfully adds less than 10 products to the wishlist

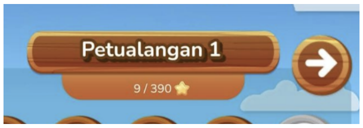
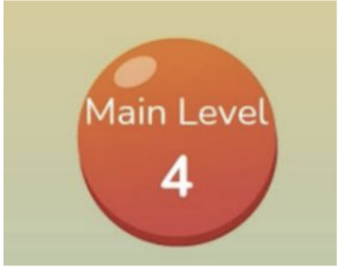
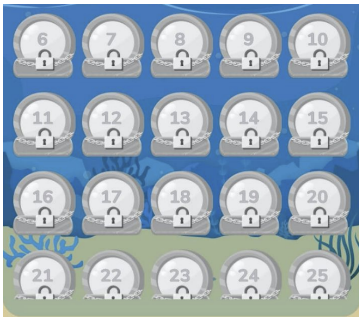
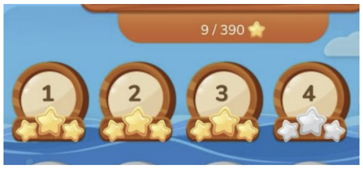
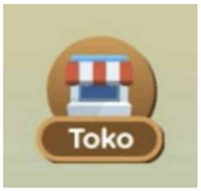
Then, I will test that maximum number of products allowed in the previous bug found, that 1000 products is maximum number of products to be allowed to add to wishlist according to the bug, which should be fixed in this case.
Test Case 2: User successfully adds 1000 products to wishlist
NOTE : assuming that the test is done in the front end/mobile app, to add 1000 products, I will use the API request using **Postman** to add 999 products to the current wishlist, and then add 1 more product using mobile app directly (actual user flow)

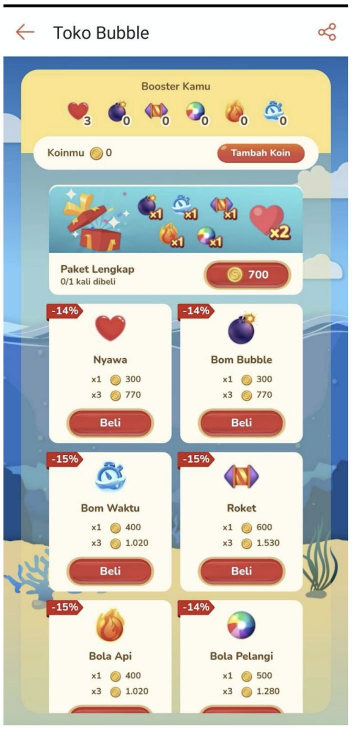
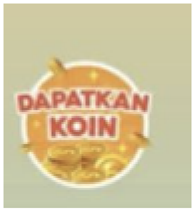
And finally, I will try to test adding 1 or more products to the wishlist after user has already have 1000 products in their current wishlist, to see if the issue was really fixed/still replicable.
Test Case 3 : User successfully adds more products to wishlist when user already has 1000 products in current wishlist

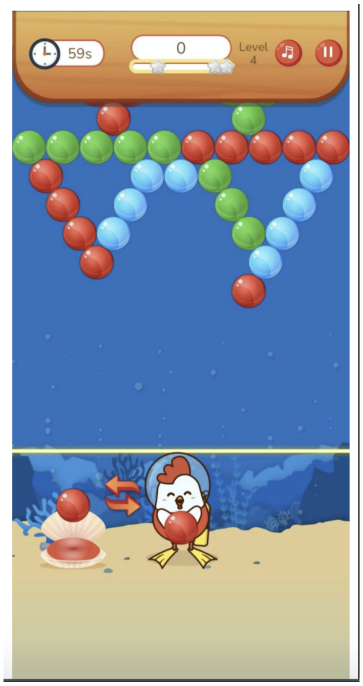
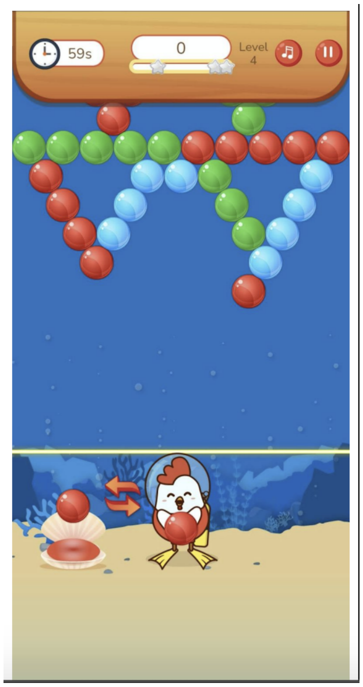
3. Please write high level test cases from the 3 designs above. Your high level test cases should at least have the following information:

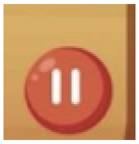
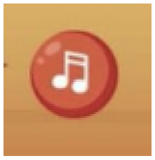
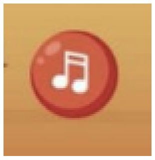
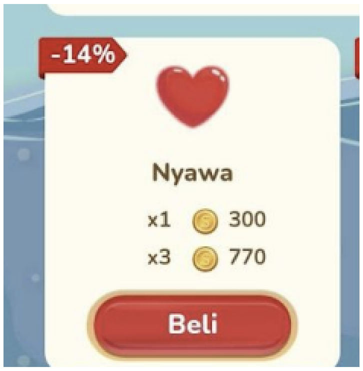
Test Case Name	Test Case Objective	Note/Assumption
Share Shopee Bubble	Verify that user is able to	user will be redirected to a

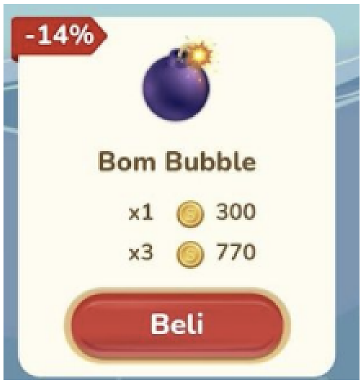
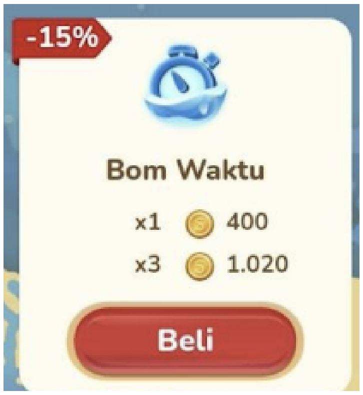
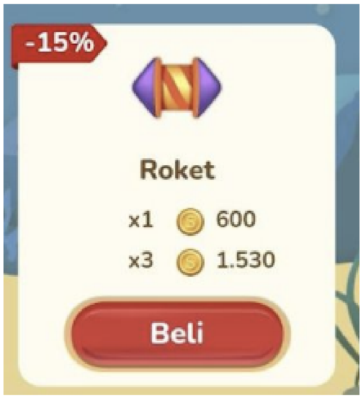
Game	share Shopee Bubble using the Share Button	list of external social media/apps to share like whatsapp, facebook, etc 
Read Game Instruction on Shopee Bubble	Verify that user is able to get instruction on how to play the game	User will get prompted with pop up modal that contains instruction on how to play the game step by step 
Add Weapon ‘Nyawa’	Verify that user is able to add weapon ‘Nyawa’ to their list	User will be redirected to Toko Bubble screen and can choose weapon to buy 
Exchange Diamond with/to buy Weapon	Verify that user is able to exchange their diamonds with weapons in Toko Bubble if number of diamonds is eligible to exchange with	User will be redirected to Toko Bubble screen and can choose weapon to exchange diamond with 
Disable ‘Geser ke Petualangan 2’ button if Current Level is not completed	Verify that user is not able to go to next stage if current Level has not been completed	
Start game by Arrow button	Verify that user is able to start to play the game by Arrow button	User will be redirected to Gameplay screen at the latest level they are in (in this case level 4)

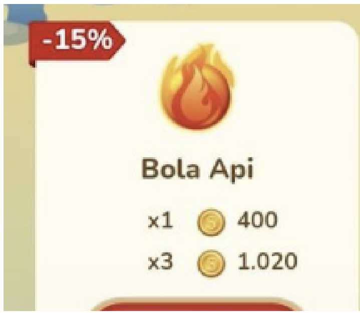
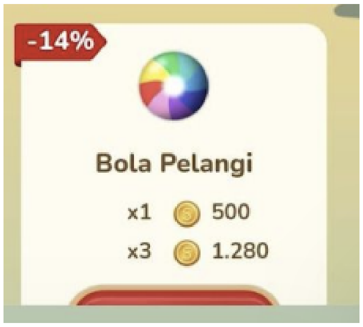
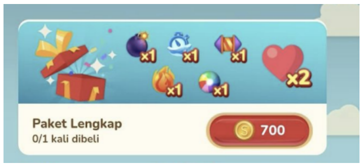
		
Start game by 'Main Level X' button	Verify that user is able to start to play the game by 'Main Level X' button	User will be redirected to Gameplay screen at the latest level they are in (in this case level 4) 
Next level button is locked	Verify that user is not able to click on all next locked level buttons	 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Previous level button is unlocked	Verify that user is able to click on all previous unlocked level buttons	User will be redirected to Gameplay screen at the respective level that they click on 
Go to Toko Bubble from Main Screen	Verify that user is redirected to Toko Bubble screen	User will be redirected to Toko Bubble Screen 

		 Toko Bubble Booster Kamu 3 0 0 0 0 0 Koinmu 0 Tambah Koin x1 x1 x1 x2 x1 x1 Paket Lengkap 0/1 kali dibeli 700 -14% Nyawa x1 300 x3 770 Beli -14% Bom Bubble x1 300 x3 770 Beli -15% Bom Waktu x1 400 x3 1.020 Beli -15% Roket x1 600 x3 1.530 Beli -15% Bola Api x1 400 x3 1.020 -14% Bola Pelangi x1 500 x3 1.280
Get free coins	Verify that user is able to get shopee coins for free	User will be prompted pop up to get eligible coins 
User win the game before time is up	Verify that user is able to play and win the game before time is up	User will be able to unlock to next level and redirected to next unlocked level

		
User lost current level if time is up	Verify that user is not able to win current level if time is up and game has not finished yet	User will not able to unlock next level, and next level is still locked 
User is able to pause the current game time	Verify that user is able to pause the time for awhile during playing the game	Game timing will stop from forwarding

		
Enable Music during game play	Verify that user is able to enable music during game play	User will hear the music sound during the game play time 
Disable Music during game play	Verify that music is disabled during game play	User will not hear the music sound during the game play time 
User buy ‘Nyawa’ from Toko Bubble	Verify that user is able to add ‘Nyawa’ to their list of weapons	User will be prompted pop up to proceed buying the weapon and successfully added to current list of ‘Booster Kamu’ 
User buy ‘Bom Bubble’ from Toko Bubble	Verify that user is able to add ‘Bom Bubble’ to their list of weapons	User will be prompted pop up to proceed buying the weapon and successfully added to current list of ‘Booster Kamu’

		
User buy ‘Bom Waktu’ from Toko Bubble	Verify that user is able to add ‘Bom Waktu’ to their list of weapons	User will be prompted pop up to proceed buying the weapon and successfully added to current list of ‘Booster Kamu’ 
User buy ‘Roket’ from Toko Bubble	Verify that user is able to add ‘Roket’ to their list of weapons	User will be prompted pop up to proceed buying the weapon and successfully added to current list of ‘Booster Kamu’ 
User buy ‘Bola Api’ from Toko Bubble	Verify that user is able to add ‘Bola Api’ to their list of	User will be prompted pop up to proceed buying the

	weapons	weapon and successfully added to current list of 'Booster Kamu' 
User buy 'Bola Pelangi' from Toko Bubble	Verify that user is able to add 'Bola Pelangi' to their list of weapons	User will be prompted pop up to proceed buying the weapon and successfully added to current list of 'Booster Kamu' 
User buy 'Paket Lengkap' from Toko Bubble	Verify that user is able to add 'Paket Lengkap' to their list of weapons	User will be prompted pop up to proceed buying the weapon and successfully added to current list of 'Booster Kamu'  x1 x1 x1 x1 x1 x2 Paket Lengkap 0/1 kali dibeli 700
User add coins from Toko Bubble	Verify that user is able to add coins to their coins	User will be prompted pop up to proceed buying the weapon and successfully added coins

4. You are responsible for testing a ticket "As a user, I can see a maintenance error message in the homepage when the server returns a timeout (504) response". How do

you validate the page it’s shown when you do testing?
Answer :
First, i will ask developers if the timeout threshold is configurable on their code implementation, if it is then i will ask them to set it to lower timeout threshold, so that its easier for tester to reach timeout status by sending huge amount of requests to access the page and eventually server return 504 gateway timeouts and i can verify the message is shown correctly

But if timeout threshold is not configurable, assuming that i have previous result on load testing on how many requests at the same time the system can handle, i will send huge amount of request that exceed this limit, that eventually will result in server returns 504 gateway timeout and i can verify the message is shown correctly

5. What is the difference between load testing and stress testing? When you want to do performance testing on search product module, what kind of performance testing will you do?
Answer :
Load Testing :
The type of performance testing to define the limit/upper threshold of a system's performance. The purpose of load testing is to know what is the maximum load a system can handle to perform while still maintaining the system's reliability. Preformance is tested by under huge number of users.

Stress Testing :
The type of performance testing to define how robust the system is if it were to be put under extreme load above its threshold/upper limit. The purpose of stress testing is to know how is the behaviour of the system while handling extreme load. Performance is tested by various amount of data (e.g too much users, too much data being processed etc).

For **search product module**, I will do **Stress Testing** to determine it's performance because search product module is usually the main module user will use a lot and we cannot control how many users are accessing this module and the way they use this module, so we need to maintain stability & reliability of this module based on stress testing result.

6. Give example a bug with these condition :
    a. **Severity level is high, but priority level is low**
    1. Shopee mobile app is crashing on certain old mobile device when User is adding more than 1000 products into their Cart lists.
    2. When user tries to input buyer name before checking out product to buy, whose has emoji characters on it, Shopee mobile app does not allow the name to have emoji characters and user cannot proceed to check out.

b. **Severity level is low, but priority level is high**
Example :
1. The front page logo of Shopee website is spelled wrongly, for example instead of Shopee, it is Shoppe.
2. Any action button on Shopee mobile app appears to be not visually appealing, example button label is not translated into human readable language instead still in programing language component although it's **still clickable and functioning correctly**, e.g **component.add_to_cart** button instead of **Add to Cart** button

7. What tools do you usually use during testing and development daily? Also mention the purpose of each tool.
Answer :
1. **Cucumber studio :** for test cases management where as a QA we write/create, store and organize our test cases in, using **gherkin syntax** :
https://studio.cucumber.io/users/sign_in
2. **JIRA :** as Product Development Tasks Management Tool used by Agile development team, to track & store stories, bugs, epics and other tasks. Specific for QA, JIRA is used for bug reporting platform.
3. **Confluence :** to document some important informations about the products being developed, can include but not limited to, high level user flows, logic of the flows, pre-conditions of the user flows, etc.
4. **Postman :** to help mimic back-end API request being called by front-end, mostly for API testing OR to create some pre-condition data easily without the need to use front-end directly (which will take less time to do so). We can also use postman to do simple load testing using postman runner
5. **SQL pro :** to verify result in Database by using SQL query to retrieve data.
6. **Android Mobile phone :** to specifically test an android-based mobile app
7. **Android emulator :** to complement/substitute the use of mobile phone for testing
8. **Script writing and versioning tools** to be used when writing automation test script :
Intellij, source tree, bitbucket etc
9. **Other online tools like :** google sheet, google docs, Miro board etc to help with productivity

8. Explain in detail about game testing, what needs to mainly be verified by a game tester for any kind of mobile games?
Answer :
1. Gameplay & Logic
Tested by playing the game as a regular user from stage to stage/level to level. check every single part of the game properties such as wall & floor location, character movement like jump & walk, usage of weapons etc.
2. User Interface
There are several aspects to test from this such as user interface layout, screen orientation, menu structure and functions, and screen resolution.

3. Game Functionality

Need to verify that all playing modes, character statuses, and features are available.

4. Graphics Performance

Smoothness, responsiveness of the graphics within the game should be tested under real life gadget/device. Make sure no crashes happen and visual glitches. This can be achieved by doing some stress testing.

5. Sound and Captions/Texts

Make sure no mismatch between sound game music and voice commands.

No misplaced captions or texts as directions to play. No localization bugs such as displaying texts in a language a user doesn’t understand/different from game language settings.

6. Multiplayer/User Features

All players must be able to perform the same set of app interactions at the same time.

7. Social Media Integrations

To test embedded connection and sharing tools integration to social media