

TEST PLAN

Individual Track

Bamboo Restaurant

Eindhoven, October 20th, 2021

Table of Contents

Revision History:.....	2
User Acceptance Tests	3
Tests Strategies.....	4
Reference.....	7

Revision History:

Date Changed	Document Version	What was changed
20/10/2021	1	
30/11/2021	2	Add more test strategies
15/12/2021	3	Add bullets for test strategies

User Acceptance Tests

Test Case Number	Test Case Name	Requirement	Steps Defined	Expected Outcome	Pass / Fail	Comment
TC-01	Login		Type username and password to the textbox Username: test1234 Password: Bamboo123	It will direct to the dashboard, means login successfully.		
TC-02	Login unsuccessful		Same as TC-01, but type incorrect password. Username: test1234 Password: Bamboo12	The message show up says "Wrong username/password, please try again"		
TC-03	Open the make new reservation dialog	TC-01 successful	Click on make new reservation button	"Make new reservation" dialog appear		
TC-03	Make a reservation	TC-01 successful	Type the date, time, description, seat choice in the future.	The dialog disappear and the new reservation appear on screen		

	Make a reservation with missing info		Same as TC, but put blank on date or time input box	A message show the missing input that needed to type in		
TC-04	List of reservation appears	TC-03 successful	After make reservation successful, refresh the page	There is a reservation appear on screen		
TC-05	A list of coupons appears	TC-01 successful	Click on the "Promotion" tab on header	Coupons will appear on the screen		
TC-06	Show the Self Message	TC-01 Successful	Click on the "Chat" blue button at the right-bottom of the page. Type your message on input and click submit	Your message will show in white container.		

Tests Strategies

Currently, we have made the customers' web application and server application to handle the request from/ response to customers' web application.

From the research documents [1][2], we will have variety of test strategies that we can apply to project:

- Unit Testing (Mockito for backend, Jest for front-end)
- SonarqubeTest (checking the vulnerable coding style in backend)
- Postman Testing (checking API request, respond)
- Cypress (For end-to-end testing)

First, we will test the customer application and server application separately, before testing complete suite applications.

For customers' web application, we will test how React rendering UI by given sample data and check the correctness of rendering processes, apply to different pages.

For server application, we will test the API functions to check the correctness by given sample data and the api should return expected output.

However, testing is a long and frustrated process, will need high quality dataset and also the implementation for the actual testing mechanism, so I will suggest an efficient test strategy and then from that we can continue to improve on

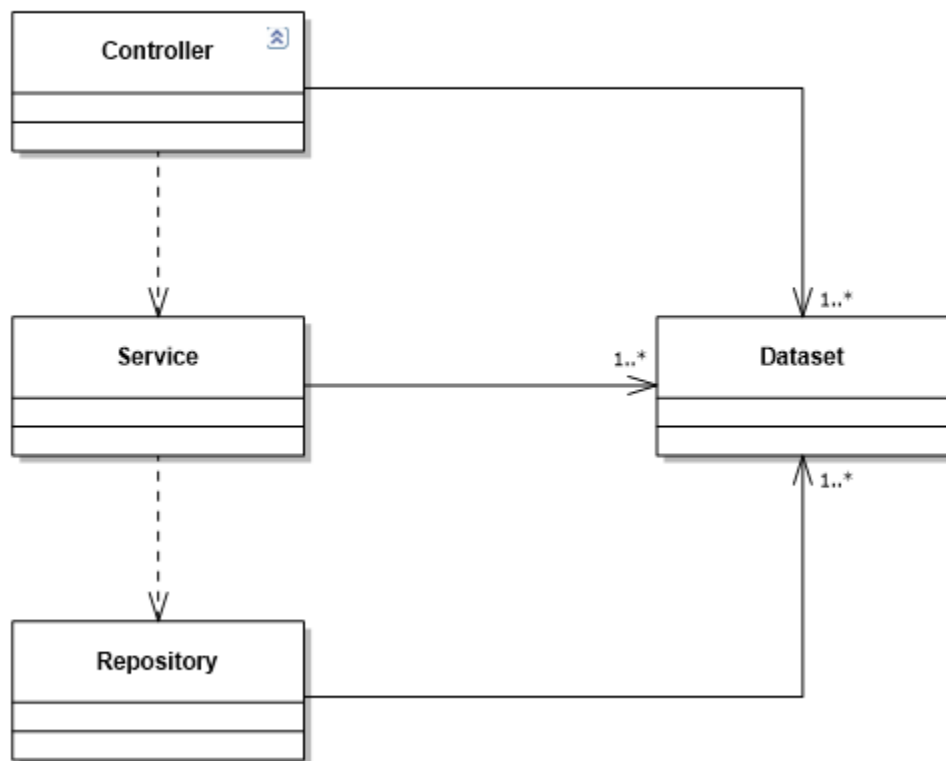


Figure 1 The diagram for testing strategies on backend platform.

Given the layers of design, we will test each component separately with the same dataset testing. Firstly, we will apply Unit Testing for each component for checking the correctness, so that we can reduce the effort to find the bugs if we going to test on the bigger scale like Integration and End-to-End testings.

For the front end design, we will have different components that we can reuse for other parts of the projects.

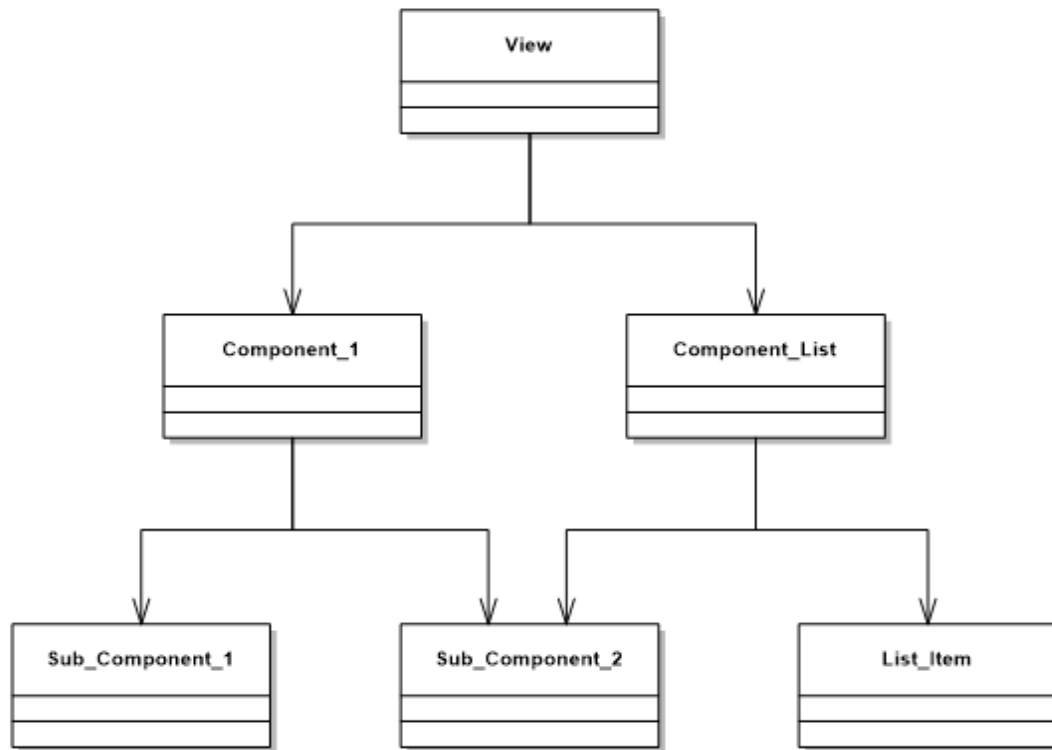


Figure 2 The diagram for example frontend design

Firstly, we will analyze and separate the components into small reasonable size for testing, for example, a component should not longer 50 lines of code, separate the states in Javascript and the hard-code html/css UI.

Secondly, we will apply Unit tests with the same dataset to all components, the UI hard-code is test for the correctness of rendering, the states handling is test for the correctness of logic.

Both tests for the separate the front end and backend will handle on CI automatic so that we can focus on fix bugs on the real implementation of components.

After the automatic tests have passed for at least 80%, we will test the whole project with the User Acceptance Test above – that we are continue to build with additional criteria, before we can deliver to clients in an amount of time.

Reference

1. Elazar, E., Elazar, E., Binny, D., Binny, D., Weissman, S., & Weissman, S. (2021, February 10). *What is user acceptance testing (UAT) - the full process explained*. Panaya. Retrieved November 30, 2021, from <https://www.panaya.com/blog/testing/what-is-uat-testing/>
2. Hamilton, T. (2021, October 7). *Test plan: What is, how to create (with example)*. Guru99. Retrieved November 30, 2021, from <https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html>