Software Testing I

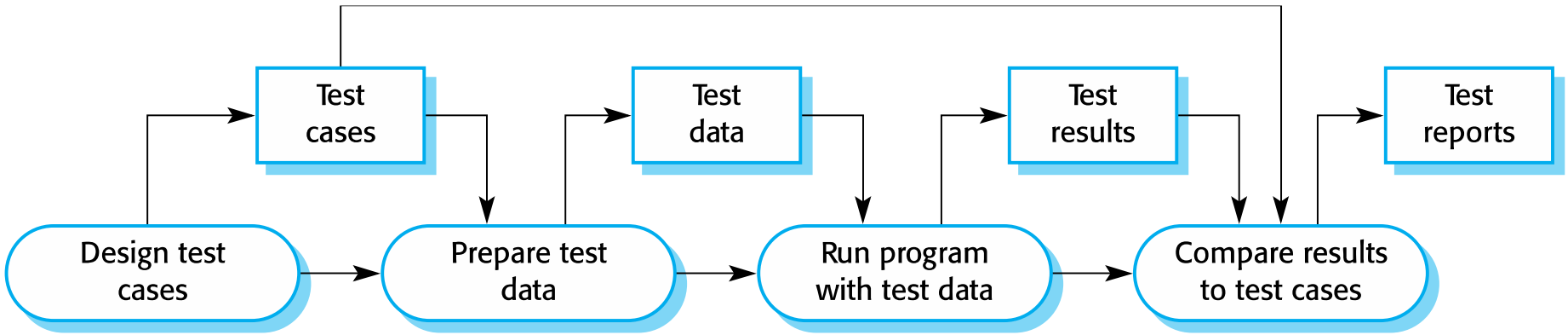
Asangi Jayatilaka

Week #9: Lectorial

Question

Why do we test software systems?

A model of the software testing process



Example of Test Cases

Test Scenario	Test Cases	Test Data	Expected Result	Test Result	Pass or Fail

Example of Test Cases

Requirement: Passwords should include at least one number, one letter, and a special character.

Test Scenario	Test Cases	Test Data	Expected Result	Test Result	Pass or Fail
Check the functionality of the **Password_Checking** function	**Test Case 1.** Verify the function with an INCORRECT password	Password: ABCD1234	Password is Invalid	Password is Invalid	Pass
	Test Case 2. Verify the function with a CORRECT password	Password: ABCD1234@	Password is Valid	Password is Invalid	Fail

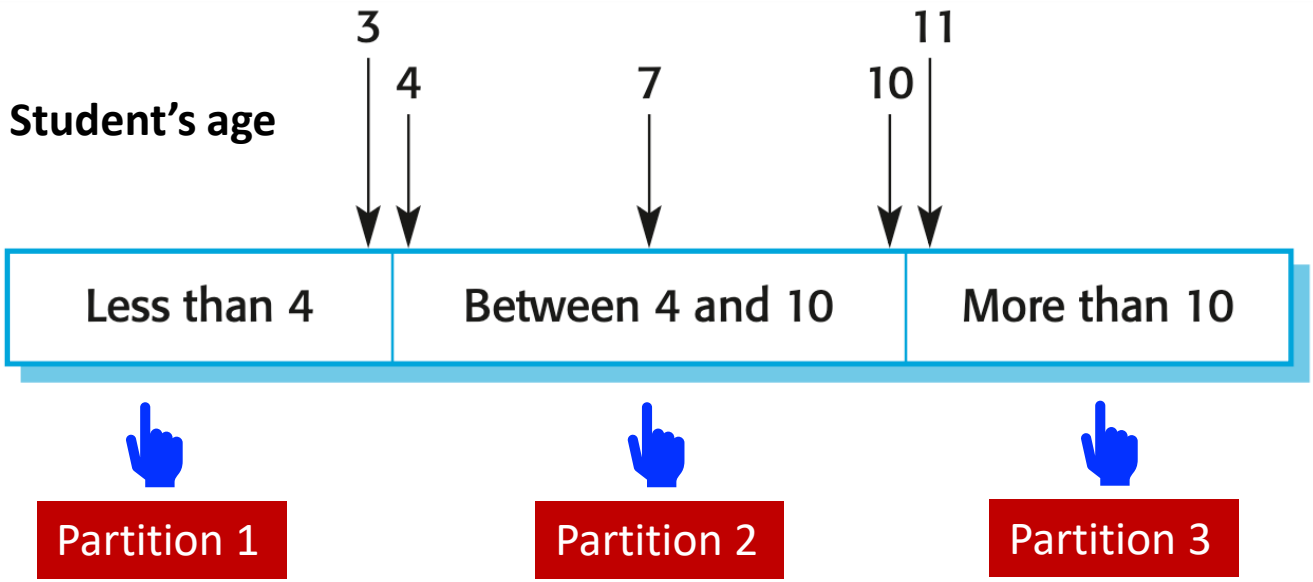
Equivalence Partitioning

- To keep down our testing costs, we don't want to write several test cases that test the same aspect of our program.
- **Equivalence partitioning** is a strategy that can be used to reduce the number of test cases that need to be developed.
- **Equivalence partitioning** divides the input domain of a program into classes.
  - For each of these equivalence classes, the set of data should be treated the same by the module under test and should produce the same answer.
- **Once you have identified these partitions, create test cases for each partition.**

Equivalence partitions

Let's suppose you are asked to write a function that only retrieves the name of students aged between 4 and 10.

A good rule of thumb for test-case selection is to choose test cases on the boundaries of the partitions, plus cases close to the midpoint of the partition.



Question

- **In what order should the testing activities be carried out?**

a) System Testing, Unit Testing, Component Testing, Acceptance Testing
b) Unit Testing, System Testing, Acceptance Testing, Component Testing
c) Unit Testing, Component Testing, System Testing, Acceptance Testing
d) Component Testing, Unit Testing, System Testing, Acceptance Testing

Question

- **Acceptance Testing should be derived based on**

a) system design document
b) requirements document
c) code modules

Question*

- You are testing an automatic auction system. Suppose there is an auction in which the bids can only be placed between 1/1/2018 and 1/7/2018. The starting bid price of this auction must be at least $20.00 and a minimum incremental bid of $5.00 is required. **Using the equivalence partitioning to derive a set of test cases for the bid placement.**

Taken from: (Partial) Introduction to Software Engineering Practices and Methods by Dr. Laurie Williams NCSU CSC326 Course Pack 2010-2011 (Seventh) Edition
https://sdc.csc.ncsu.edu/files/resources/williams-software-engineering-2011.pdf

Answer - Partitions

Date

Earlier than 1/1/2018	Between 1/1/2018 and 1/7/2018	Later than1/7/2018

Bid Price

Less than $20.00	$20.00 or more

Bid Increment

Less than $5 incremental bid	$5 incremental bid or more

Answer - Test Cases

We fill these parts when we run the test

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 1. Bid Number: Bid1 Bid1 with correct information	(Bid1, 4/5/2018, 30)	This bid can be placed.		
Test Case 2. Bid Number: Bid2 Bid2 with incorrect increment	(Bid2, 30/6/2018, 34)	This bid should not be placed.		
Test Case 3. Bid Number: Bid3 Bid3 with correct information	(Bid3, 30/6/2018, 35)	This bid can be placed.		

The minimum incremental bid should be $5.00

Question*

- You are asked to test a method called **ReplaceWhiteSpace** in a "Paragraph" object. Within a paragraph, this method replaces a sequences of blank characters with a single blank character. Identify testing partitions for this example and derive a set of tests for the **ReplaceWhiteSpace** method.

Example

A Sequence of Blank Spaces

"Java is a high-level, class-based, object-oriented programming."

"Java is a high-level, class-based, object-oriented programming."

Taken from: Ian Sommerville, Software Engineering, 10th Edition, 2015.

Answer - Partitions

Blank Character

Strings with only a single blank character	Strings with sequences of blank characters

String with sequences of blank characters at the at the beginning	String with sequences of blank characters in the middle	String with sequences of blank characters at the end

Answer - Test Cases

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 1. Single Blank Character	*"Java is a high-level, class-based, object-oriented programming."*	*"Java is a high-level, class-based, object-oriented programming."*		

Answer - Test Cases

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 2. Sequences of blank characters at the at the beginning	" *Java is a high-level, class-based, object-oriented programming.*"	" *Java is a high-level, class-based, object-oriented programming.*"		

Answer - Test Cases

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 3. Sequences of blank characters in the middle	*"Java is a high-level, class-based, object-oriented programming."*	*"Java is a high-level, class-based, object-oriented programming."*		

Answer - Test Cases

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 4. Sequences of blank characters at the end	*"Java is a high-level, class-based, object-oriented programming [redacted]."*	*"Java is a high-level, class-based, object-oriented programming ."*		

Question

- In the Uber Eat app, there is a class called **Food** with a function named "**AddFood**" to add new food. The type of food can be "Kid Food", "Adult Food", "Healthy Food", or "Elderly Food"). You are asked to write test cases for AddFood() with the following conditions. The Food class has the following attributes.

```
public class Food {
  private String foodName
  private String foodDescription;
  private String foodType
  private int foodCalorie;
  private double foodPrice;
}
```

If conditions are met, AddFood adds food and returns **True**, else it does not add food and returns false **False**

Conditions

1. Food name should be between 5 and 30 characters
2. Food description should be between 5 and 50 words
3. It should not be possible to add foods with more than 1500 calorie
   3.1 If the type of food is "Kid Food", their calorie should be less than 800.
4. The price of each food should be between $5 and $150
   4.1 The price of foods with more than 1000 calories should be less than $100.

Answer - Partitions

Condition 1. The food's name should be between 5 and 30 characters

Food Name

Less than 5 characters	Between 5 and 30 characters	More than 30 characters

At least one test case

At least one test case

At least one test case

Answer - Partitions

Condition 2. Food description should be between 5 and 50 words

Food Description

Less than 5 words	Between 5 and 50 words	More than 50 words

At least one test case

At least one test case

At least one test case

Answer - Partitions

Condition 3. It should not be possible to add foods with more than 1500 calorie

Condition 3.1 If the type of food is “Kid Food”, their calorie should be less than 800.

All Food Calorie except for Kid Food

<= 1500	> 1500
At least one test case	At least one test case

Kid Food Calorie

<= 800	> 800
At least one test case	At least one test case

Answer - Partitions

Condition 4. The price of each food should be between $5 and $150

Condition 4.1 The price of foods with more than 1000 calories should be less than $100.

Price for all foods except for foods with more than 1000 calories

Less than 5	Between 5 and 150	More than 150

Price for foods with more than 1000 calories

Less than 5	Between 5 and 100	More than 100

Answer – Test Cases

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 1. Check the function with <u>valid inputs</u>	Food Name, Food Description ("pesto pasta", "delicious pesto served on a bed of penne", "Adult Food", 1499, 149) Type Calorie, Price ("bolognese", "delicious and tomatoey spaghetti bolognese", "Kid Food", 799, 6)	True		

Answer – Test Cases

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 2 Check the function with <u>an invalid Name</u>	("**egg**", "2 hard boiled eggs served with toast", "Elderly Food", 156, 5) ("**cake**", "spongy and creamy strawberry cake", "Kid Food", 138, 8.50)	False		

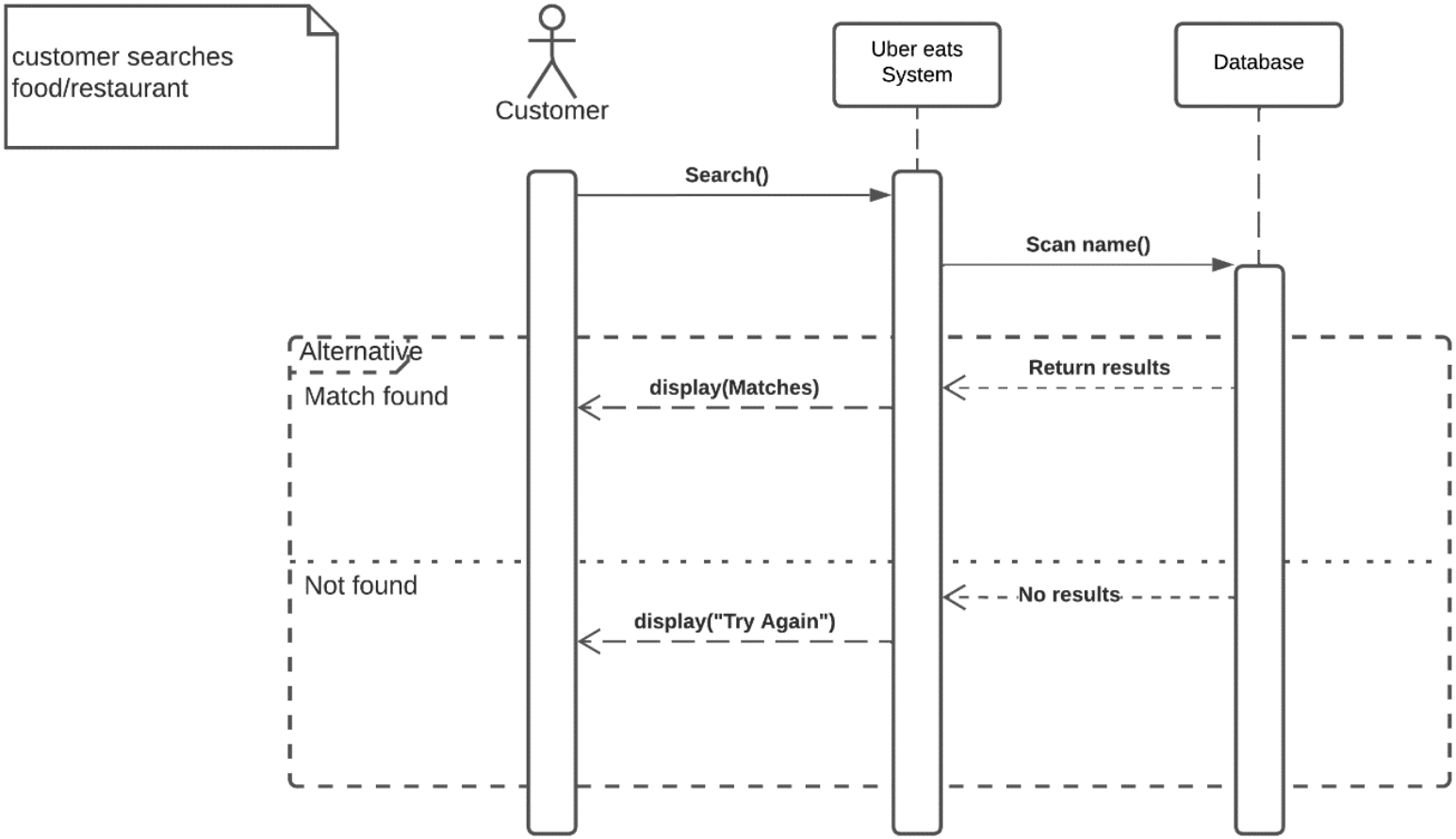
Answer – Test Cases

Test Cases	Test Data	Expected Result	Test Result	Pass/ Fail
Test Case 3 Check the function <u>with an invalid Price</u>	("chips", "fresh fried chips sprinkled with paprika seasoning", "Kid Food", 312, **4.99**) ("water", "sustainably bottled mineral spring water", "Kid Food", 0, **151**) ("50g caviar", "a smooth and luxurious burst of flavour", "Adult Food", 1001, **101**)	False		

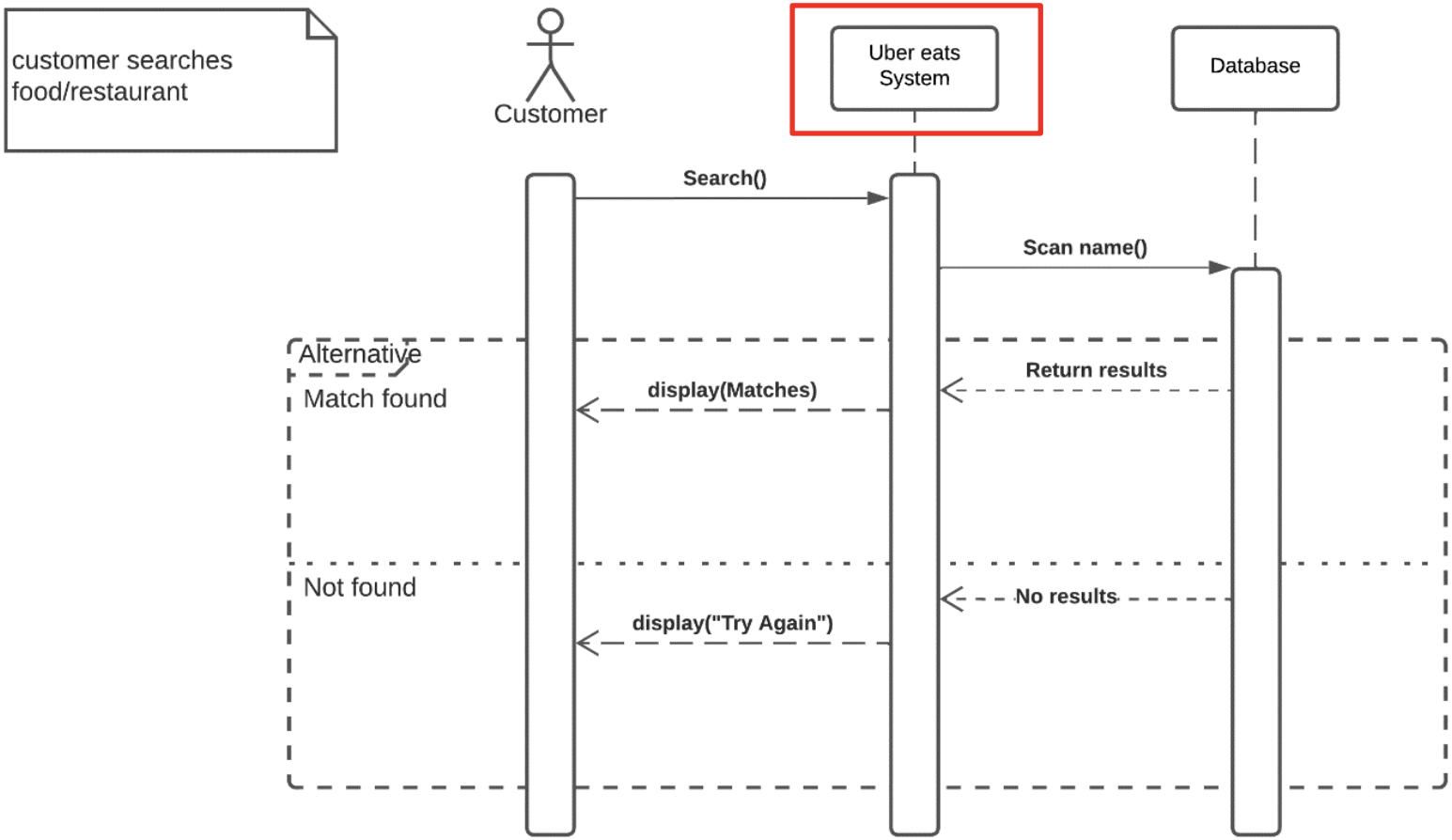
Students' Common Mistakes in Sequence and Activity Diagrams

Context: The Uber Eat app

Sequence Diagram for "Search Food" – Mistakes?

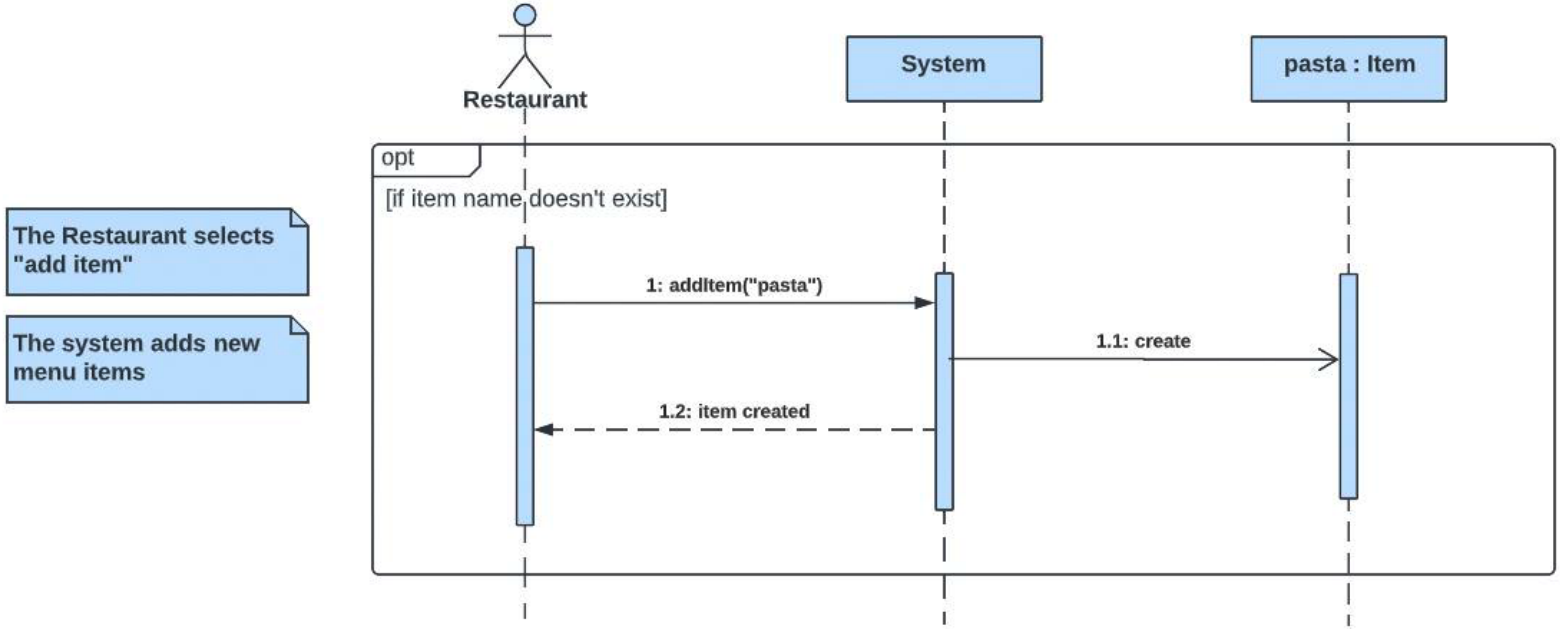


Sequence Diagram for "Search Food" – Mistakes?

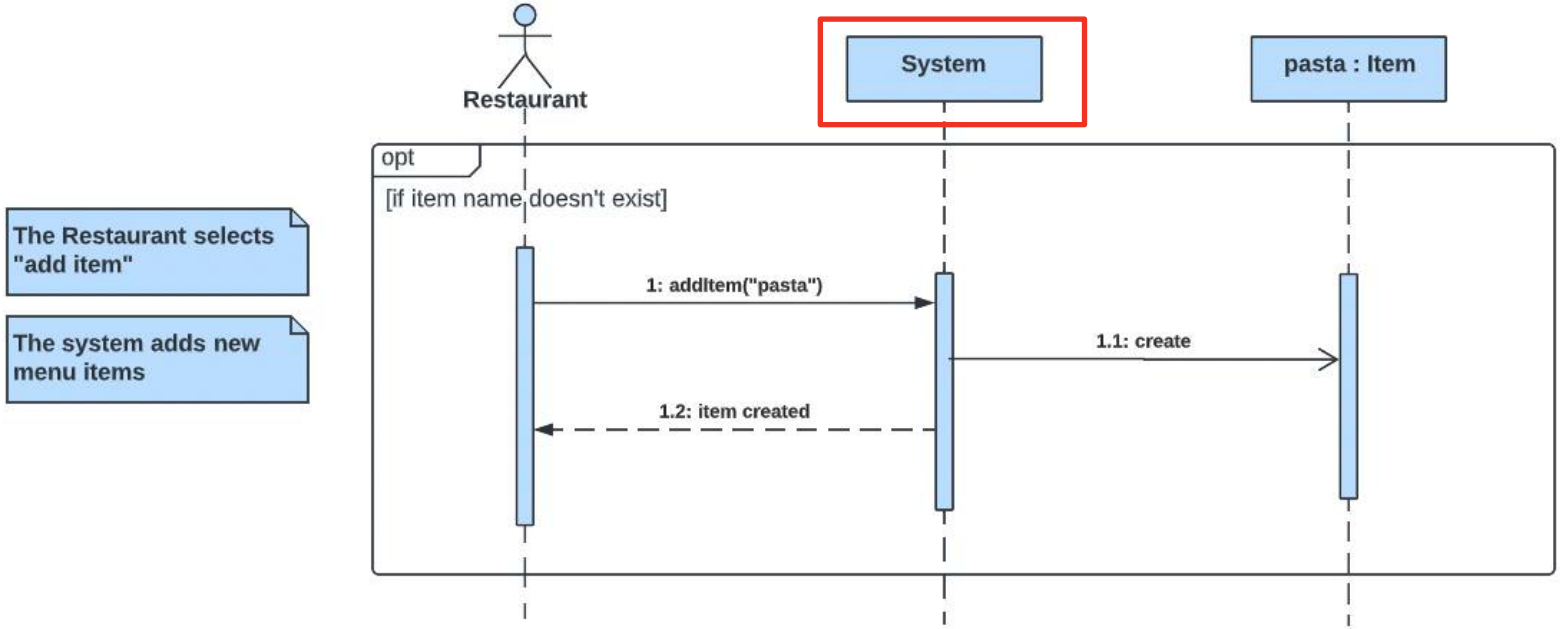


Mistakes:
a) You are modelling the User Eat system. How can Uber Eat System can be a lifeline here?

Sequence Diagram for “Add Item by Restaurant” - Mistakes?



Sequence Diagram for “Add Item by Restaurant” - Mistakes?



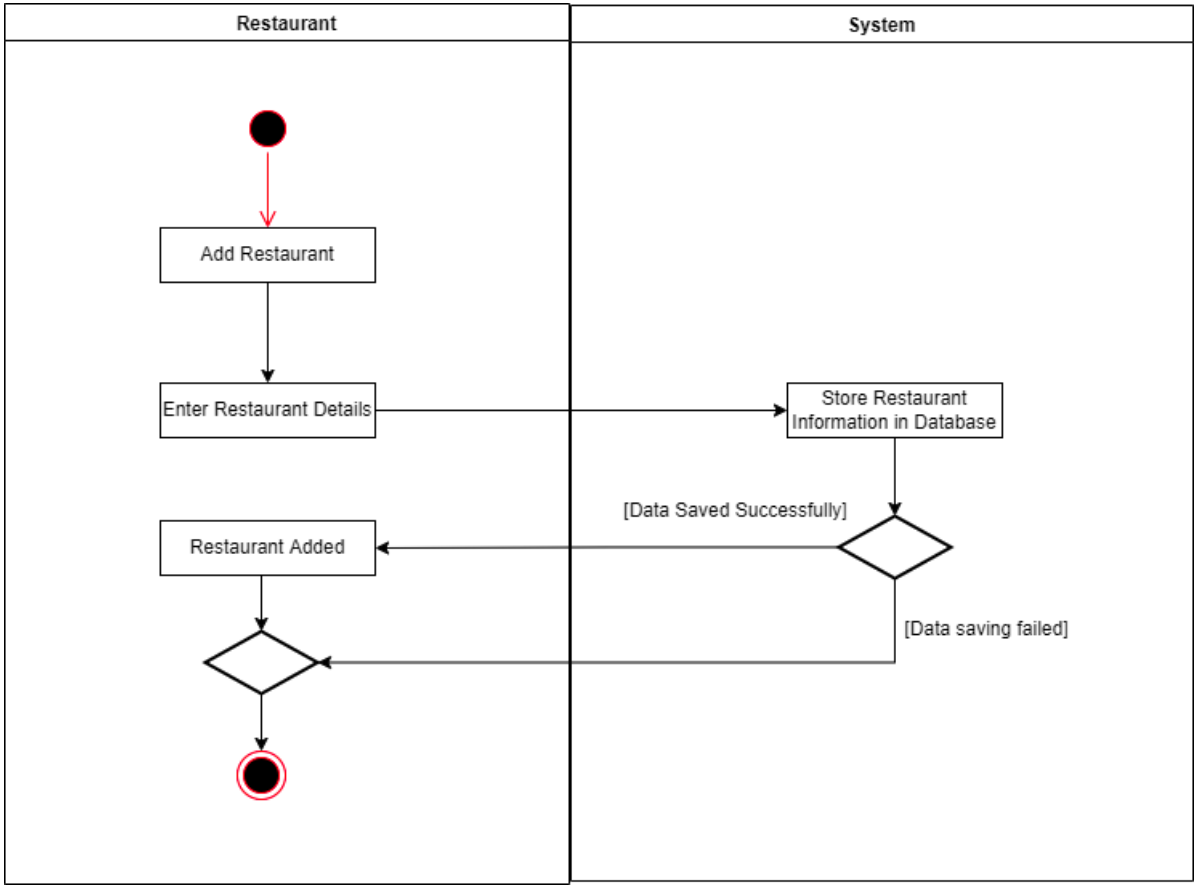
Mistakes:

a) What do you mean by "system"? “System” is a very general term.

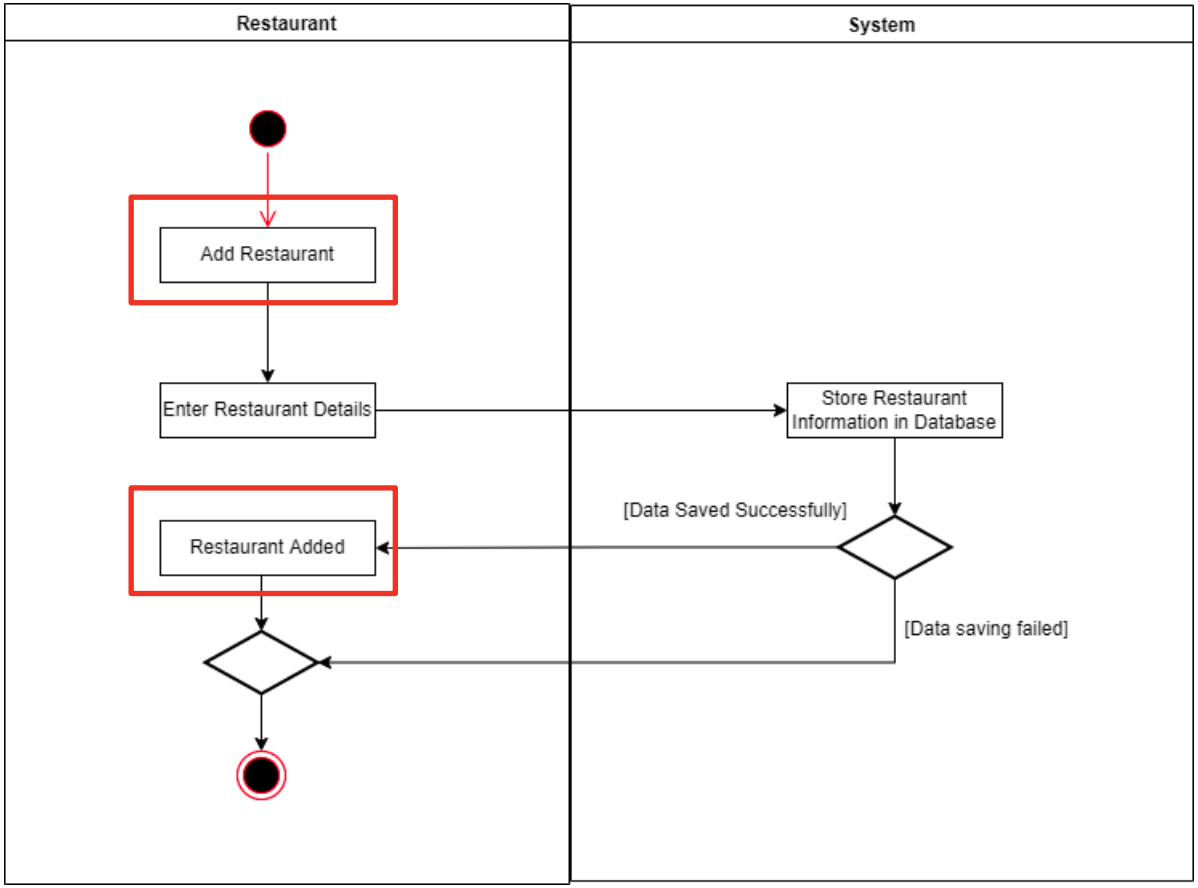
b) Do you mean the Uber Eat system? Note that you are modelling this system!! You cannot use it as a lifeline here

c) Very incomplete diagram

Activity Diagram for “Add Restaurant” - Mistakes?

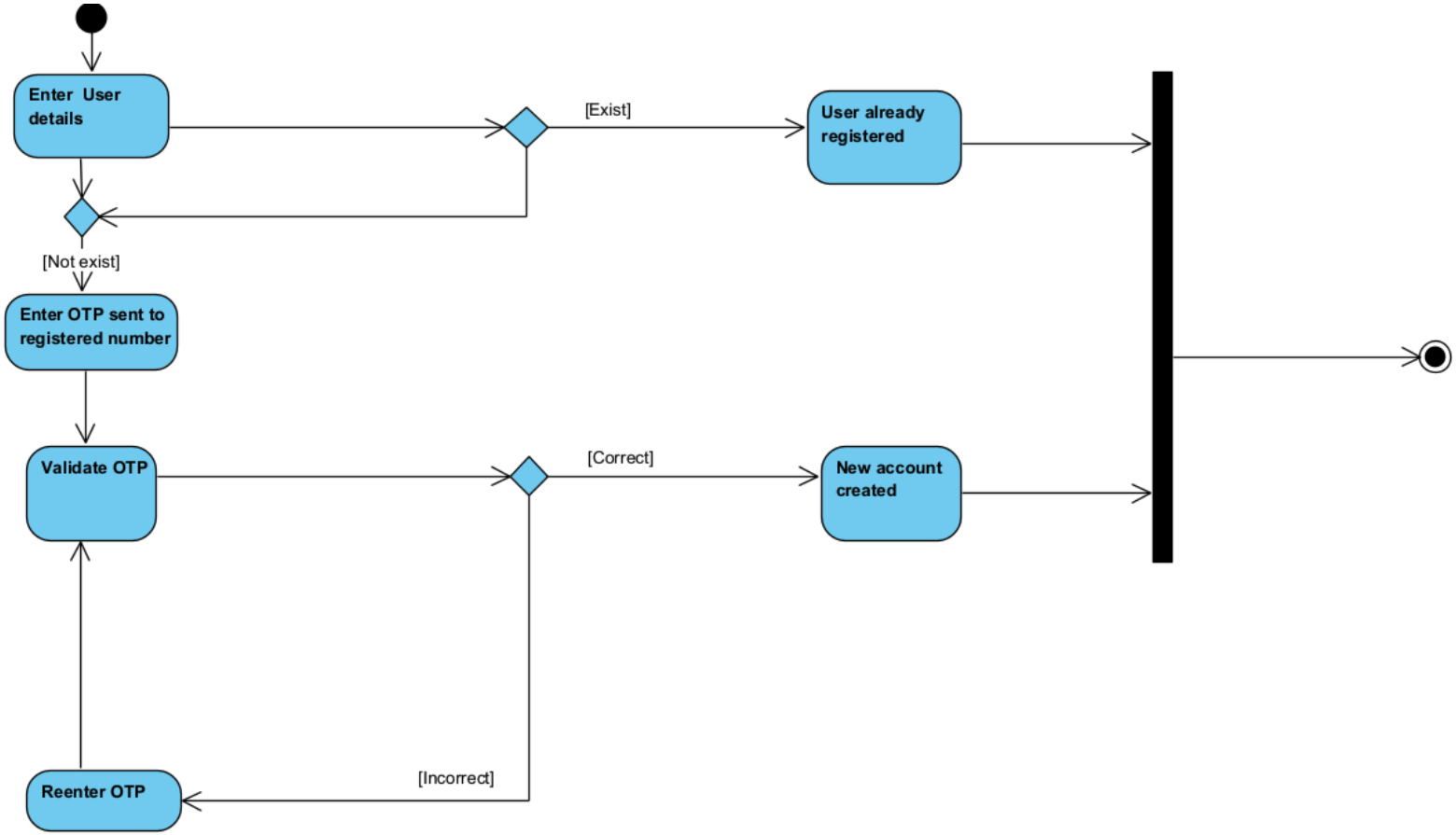


Activity Diagram for “Add Restaurant” - Mistakes?

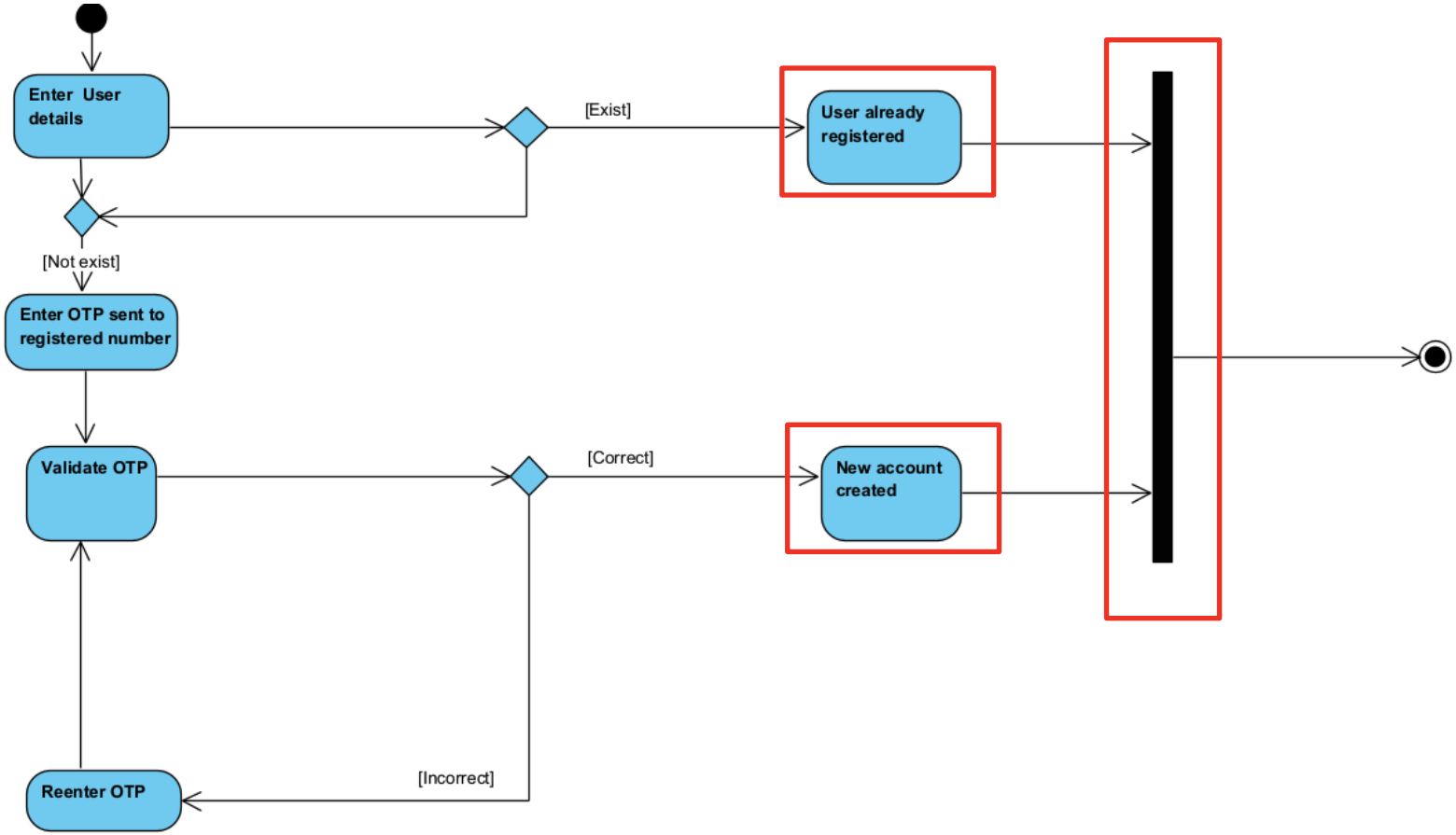


Mistakes:
a) how "Add Restaurant" can be an action in the "Add Restaurant" activity diagram?
b) "Restaurant Added" is the wrong action. Actions should start with a verb phrase

Activity Diagram for “Sign up” - Mistakes?



Activity Diagram for "Sign up" - Mistakes?



Mistakes:
a) "User already register" cannot be an action. Actions should start with a verb or verb phrase. Same for "New account created".
b) Before each "Join", there should be a "Fork". The same for Decision and Merge.

Next Week

Course	Topic
Lecture/Lectorial	Software Testing II
Practical	Software Testing I

References

- Ian Sommerville, Software Engineering, 10th Edition, 2015.
  - https://iansommerville.com/software-engineering-book/slides/
- (Partial) Introduction to Software Engineering Practices and Methods by Dr. Laurie Williams NCSU CSC326 Course Pack 2010-2011 (Seventh) Edition
  - https://sdc.csc.ncsu.edu/files/resources/williams-software-engineering-2011.pdf