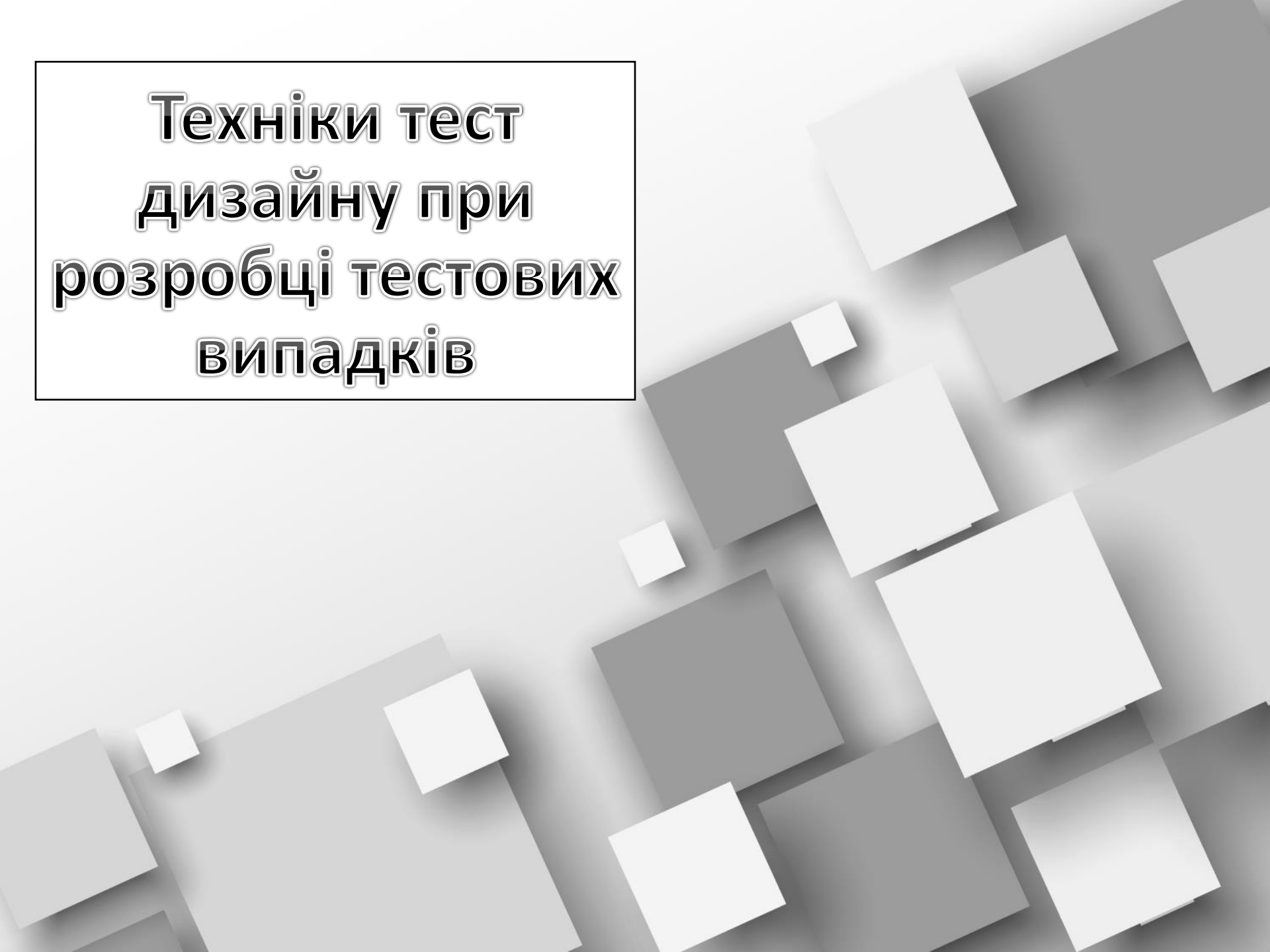


Техніки тест дизайну при розробці тестових випадків



Завдання

Variant 6

Equivalence partitioning and Boundary value analysis

A system is designed to calculate the electricity cost. User can enter only an old and a new values of electricity counter and press Calc button. If data entered correctly and user consumed less than 100 kw the price of 1 kw will be 25 cents. The calculator will count 1\$ per 1 kw for more than 600 kw. In another case it will count 65 cents per 1 kw.

1. Build equivalence classes (partitions) based on given information
2. Stand Out boundary values

Завдання

Decision tables

E-store works with three types of customers according to the total sum of their orders: Regular, Silver and Gold. The list of products to be shown to the client is built according to the rules: products with label Silver or Gold are placed in the top of the list for corresponding types of clients. All the rest products are alphabetically sorted, except the promo products, which have the highest priority and are placed the topmost.

1. Build decision table based on given information.

State transition

User wants to replenish the account using self-service kiosk. He enters the account number and in case it is active the user is asked to enter the sum and select the way of replenishment: from the credit card or by cash. After the cash/credit card is inserted the system checks whether enough money is available (available on the credit card or inserted into the kiosk). If there is enough money the account is replenished. In other case the user is asked to correct sum or cancel the operation.

1. Build state transition diagram based on given information

План (Agenda)

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition

Equivalence partitioning 1 (contain only digits)

Equivalence partitioning 1 (contain only digits)

Invalid			Invalid			Valid		
a	<u>abc</u>	z	*	<u>#</u> \$	/	0	123	9
Partition 1			Partition 2			Partition 3		

Equivalence partitioning 2 (new value is not less than an old one)

- Let old value be 49999.
- The table with the values of new entered value:

Invalid			Invalid			Valid		
1	20000	30000	30001	35555	49998	49999	76890	99999
Partition 1			Partition 2			Partition 3		

Boundary value analysis 1 (contain only digits)

Boundary value analysis 1 (contain only digits)

Invalid		Invalid		Valid	
a	z	*	/	1	9
Partition 1		Partition 2		Partition 3	

Boundary value analysis 2 (new value is not less than an old one)

- Let old value be 49999.
- The table with the values of new entered value:

Invalid		Invalid		Valid	
1	30000	30001	49998	49999	99999
Partition 1		Partition 2		Partition 3	

Decision tables (1)

Causes (inputs)	R1	R2	R3	R4	R5	R6	R7	R8
Regular User	+	+	+	+	-	-	-	-
Silver user	+	+	-	-	+	+	-	-
Gold user	+	-	+	-	+	-	+	-

Decision tables (2)

<u>Causes (inputs)</u>	<u>R4</u>	<u>R6</u>	<u>R7</u>
<u>Regular User</u>	+	-	-
<u>Silver user</u>	-	+	-
<u>Gold user</u>	-	-	+

Decision tables (2)

Causes (inputs)

- Regular User
- Silver user
- Gold user

R4

+

-

-

R6

-

+

-

R7

-

-

+

Effects (outputs)

- Silver products on top
- Promo products (topmost)
- Rest products alphabetically sorted
- Gold products on the top
- Rest products alphabetically sorted on the top

-

+

-

-

+

+

+

+

-

-

-

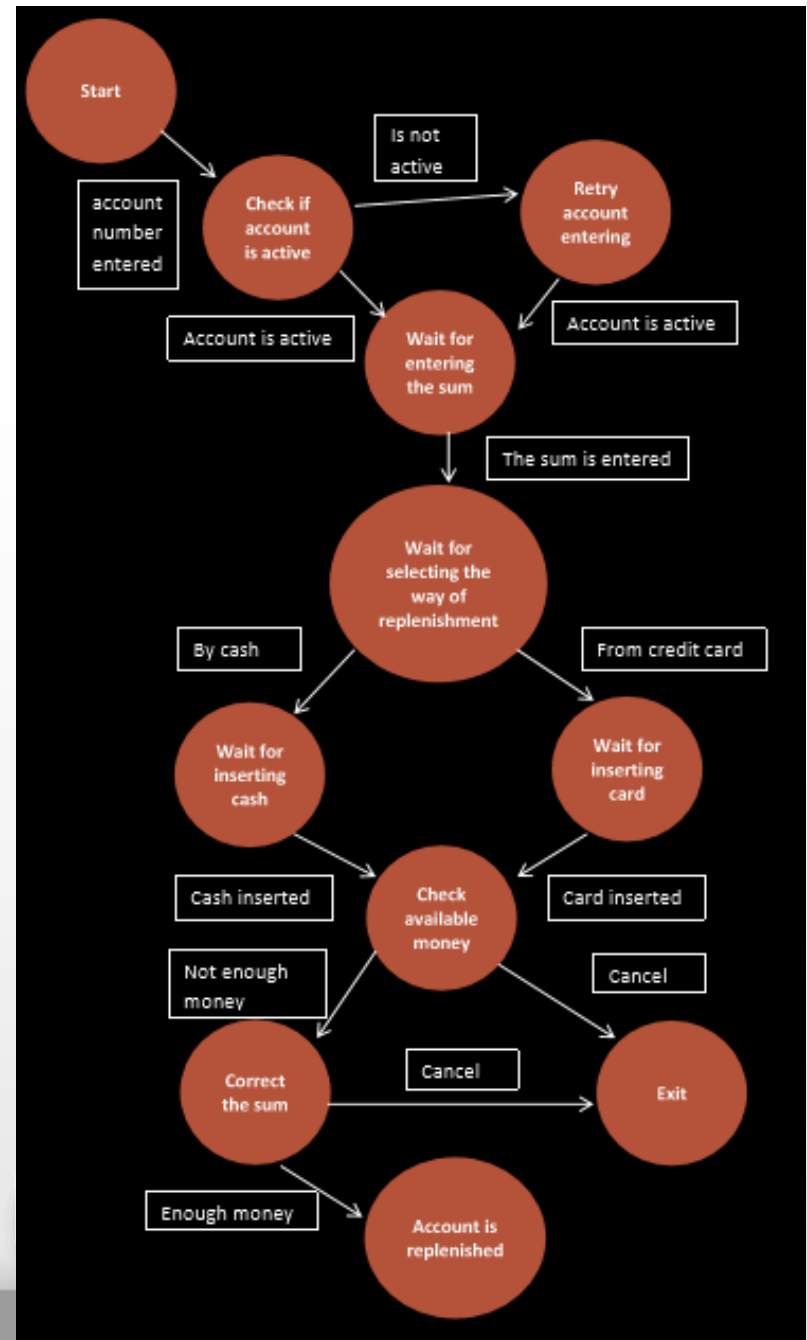
+

+

+

-

State transition



Висновок

Виконуючи дану лабораторну роботу, ми навчилися використовувати техніки тест дизайну для визначення набору тестів.

