Software Testing Mentor

www.softwaretestingmentor.com

ISTQB Foundation Level and Software Testing Training

Module 4
Test Design Techniques

Session 3 – Part 3 – Specification-based or Black-box Techniques

Part 3 – Specification-based or Black-box Techniques

In part 3 of session 3 we will learn about following 2 black box techniques

- State Transition Testing
- Use Case Testing

State Transition Testing

What is a System State?

- A state is the position or situation that a particular object or system is in at any given time.

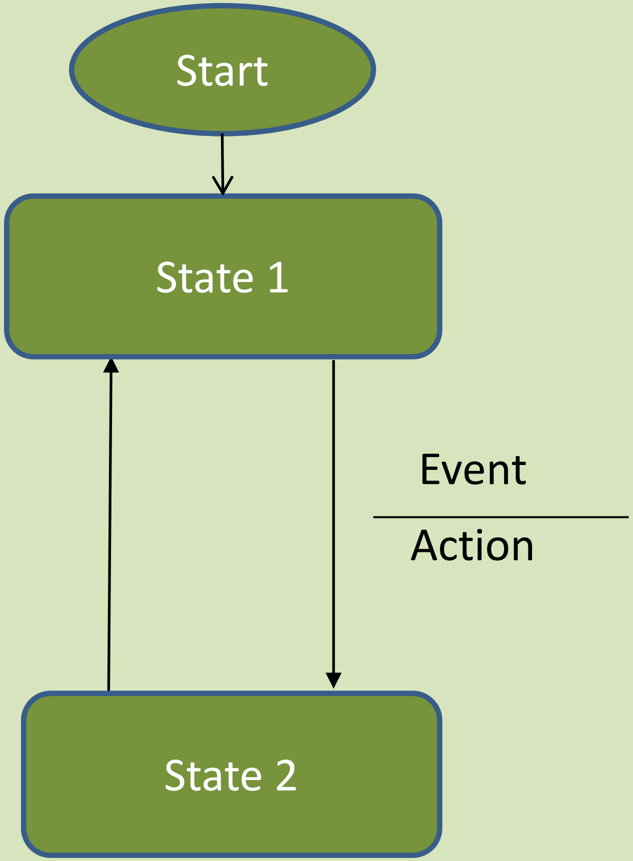
What is a transition?

- A transition describes the way in which the system moves from one state to another.

Systems move from a state to another set based on the inputs applied on the system

Mapping these changes in states and their transitions are the basic parts of state transition testing

How to Create a State Transition Diagram?



State 1 and State 2 boxes specify the current state of the system

The transition from one state to another is shown by arrows

The event is the input which must be applied to achieve the required state

The action is the outcome of the applied input

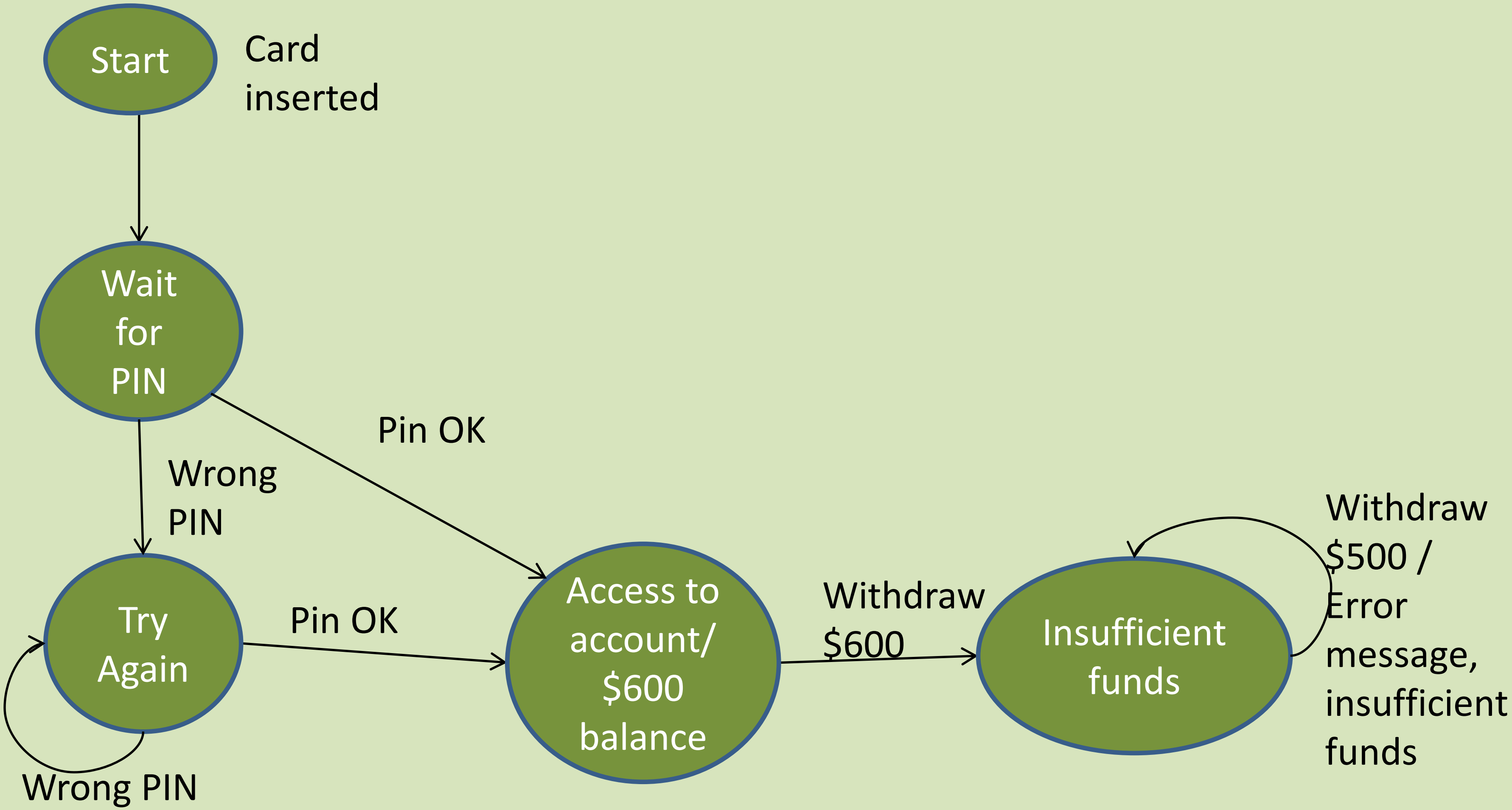
State Transition Testing example

Bank ATM example Scenario:

- Suppose you want to withdraw $500 from a bank ATM, you may be given cash.
- After some time you again try to withdraw $500 but you may be refused the money (because your account balance is insufficient).
- This refusal is because your bank account state has changed from having sufficient funds to cover withdrawal to having insufficient funds.
- The transaction that caused your account to change its state was probably the earlier withdrawal.

A state diagram can represent a model from the point of view of the system or the customer.

Bank ATM state transition diagram



Start

Card inserted

Wait for PIN

Pin OK

Wrong PIN

Try Again

Pin OK

Wrong PIN

Access to account/ $600 balance

Withdraw $600

Insufficient funds

Withdraw $500 / Error message, insufficient funds

Basic parts of state transition model

A state transition model has four basic parts:

- The states that the software may occupy (funded/insufficient funds)
- The transitions from one state to another (all transitions are not allowed)
- The events that cause a transition (like withdrawing money)
- The actions that result from a transition (an error message or being given your cash)

Please note that in any given state, one event can cause only one action, but that the same event from a different state may cause a different action and a different end state.

Deriving test cases from State transition diagram

State table is very useful for deriving test cases from state transition graph.

State table lists all the states down one side of the table and all the events that cause transitions on the top, each cell then represents state-event pair.

	Insert card	Invalid PIN	Valid PIN	Withdraw $600
S1(Start state)	S2		-	-
S2(Wait for PIN)	-	S3	S4	-
S3(Try Again)	-	S3	S4	-
S4(Access Account - $600)	-		-	S5
S5(Insufficient funds)	-		-	S5

Use Case Testing

What is a use case?

- A Usecase is a description of a particular use of the system by the end user of the system.
- Usecases are a sequence of steps that describe the interactions between the user and the software system.
- Each usecase describes the interactions the end user has with the software system in order to achieve a specific task.

What is Usecase Testing?

Usecase testing is a technique that helps us identify test cases that exercise the whole system on a transaction by transaction basis from start to finish.

Usecases are defined in terms of the end user and not the system, use case describe what the user does and what the user sees rather than what inputs the software system expects and what the system outputs.

Usecases use the business language rather than technical terms.

Each usecase must specify any preconditions that need to be met for the use case to work. Use cases must also specify post conditions that are observable results and a description of the final state of the system after the use case has been executed successfully.

A set of use cases make up Functional requirements of a system

Benefits of Use Case testing

Use cases capture system’s functional requirements from an end user’s perspective

Actively involve end users and stakeholders in the requirement gathering process

Serve as the foundation for developing system test cases

Conclusion

We learned about following black box test design techniques

- State Transition Testing
- Use Case Testing

THANK YOU!!!