Test Design Techniques

Astek Mauritius

www.groupeastek.com

ASTEK

Content

Overview of Software

- WHAT DOES SOFTWARE DO?
  - Accepts **Input**
  - Send **Outputs**
  - Stores and manipulates **Data**
  - Performs **Computation**

"This is what we must test"

- HOW DO WE TEST IT?
  - Study Input
  - Study Output
  - Study Data Manipulation
  - Study Computation
  - Design how to attack all of these

Input/Output and Data Attacks

- **Input/Output Attacks**
  - Create impossible input combinations
  - Force invalid output
  - Input sequence - Repeat inputs
  - Input combinations and permutations
  - Single input – All error messages, default values, screen refresh problems, overflow display areas…

- **Data Attacks**
  - Variable values – Incorrect data types, exceed permissible ranges
  - Data item size – Overflow input buffers, create too many values, force too few values
  - Access – Change the same data from different routes
  - Wrong data types
  - Wrong data formats
  - Leading zeros
  - Leading spaces
  - Nulls
  - Rounding
  - Overflows
  - Negative values
  - Broken links

Different Types of Design Techniques

- **Prototype Tests**
  - Might also be know as: Visible State Transition Tests, Installation Verification Tests, Health/Sanity Checks or Smoke Tests
  - Is it all there? Do all parts exist?
  - Test drive the system
  - Concentrate on the system usability
  - If the smoke test fails, the software is not complete and therefore not testable.
  - Follow the specification

    Has each function been catered for?
  - Navigate through the whole system (and back)
  - Look for unexpected behaviour (crashes)

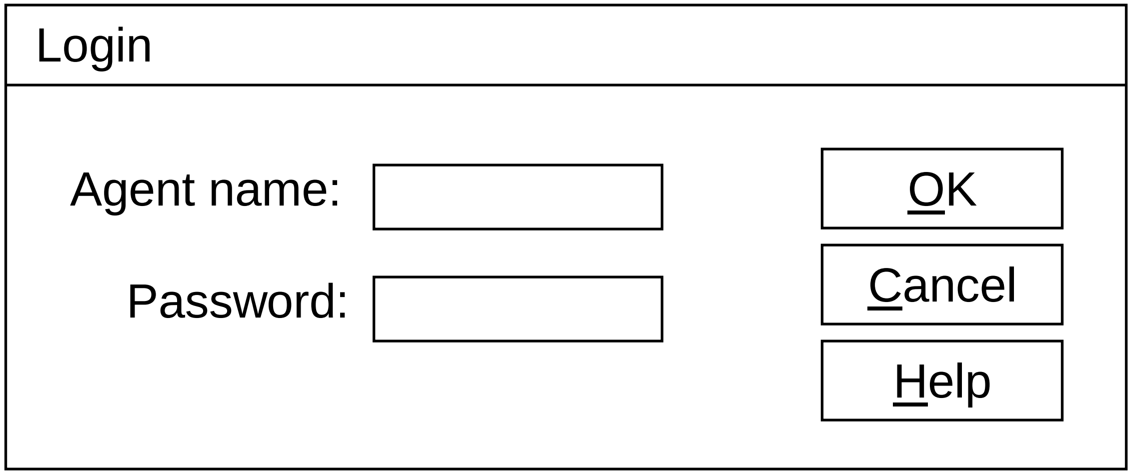
Prototype Test Example

Login	Main window appears
Select File -> New	Screen is cleared
Select New Customer	Customer Details window is displayed
Select Back	Returns to the main window
Select New Account	Account Details window is displayed
Select Back	Returns to main window
Select File -> Exit	Main window closes

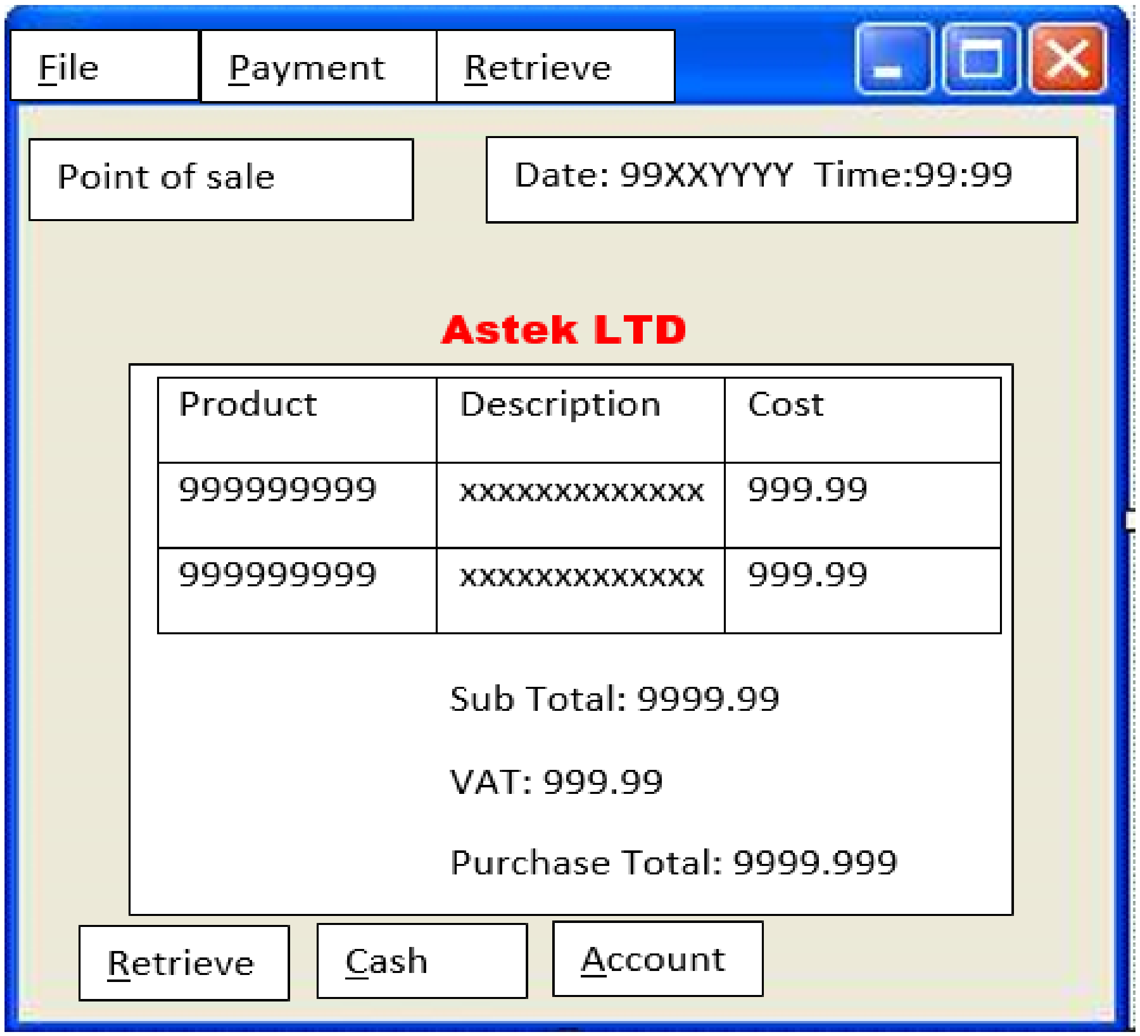
Different Types of Design Techniques

- **User Interface Testing**
  - We do user interface testing to check that each component of the user interface performs as expected:
    - Every window
    - Every page
    - Every field
    - Default state (static, visible, enabled, disabled, focused,..)
    - Default value
    - Behaviour
  - Where does the information comes from?
    - The specifications
    - Logic (understanding how the screen is to be used)

User Interface Testing Example



User Interface Testing Example



Date: 99XXYYYY Time:99:99

Astek LTD

Product	Description	Cost
999999999	xxxxxxxxxxxxx	999.99
999999999	xxxxxxxxxxxxx	999.99

Sub Total: 9999.99

VAT: 999.99

Purchase Total: 9999.999

Different Types of Design Techniques

- **Functional Testing**
  - Black Box Testing
  - Inputs and outputs are as per specifications
  - Gives up to 80% coverage
  - Reduce whole system to black boxes
    - As small as possible, not too complex
  - Define inputs
  - Define expected results
  - Do most critical, highest risk functions first

Different Types of Design Techniques

- **Domain Testing**
  - Used for data validation
  - Applied to set of related variables
  - Requires complete specification of fields
- **Equivalence Classes**
  - Set of values that have something in common
    - E.g. {months with 31 days}
  - One member = all (Equivalence classes have same expected results)
  - First task: Equivalence partitioning
    - i.e. identify all the classes
- **Equivalence partitioning:**
  - Divide input into classes
    - Identify valid and invalid data for each class
  - Affect the same output
- **3 main types**
  - Numeric
    - Range of numbers e.g. 1-99
  - Value set
    - List of options e.g. months
  - Free Form
    - None of the above e.g. addresses

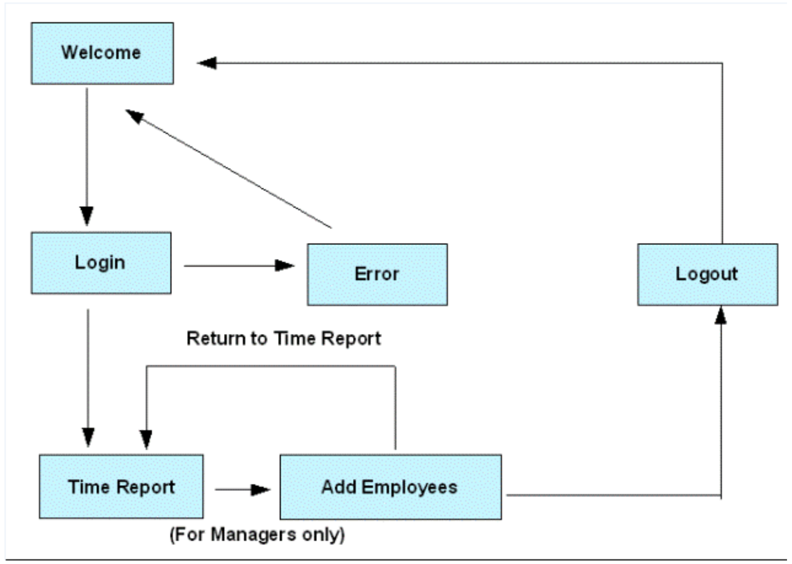
Decision Tables & Pairwise Testing

- **Decision Tables**
  - List inter-dependencies
  - List combinations and permutations
  - Identify expected results for each

- **Pair-wise testing**
  - Identify variables
  - Work out combinations and permutations
  - Manipulate
  - Identify expected results for each

Data Flow Testing

- **Flow of data through the system**
  - Where does the data enter the system?
  - Where is it used to change other data?
  - Where does it get changed?
  - Where is it output from the system?
  - Select data items to be monitored
    - i.e. not all



Regression Testing

- Regression testing is testing done to determine whether a product has regressed to a less perfect state than in the previous build
- Regression testing is repetitive
- Verify changes
- Re-use
  - Test Cases
  - Test Data