FACTFILE:
GCE SOFTWARE SYSTEMS DEVELOPMENT

A2 1 UNIFIED MODELLING LANGUAGE

Unified Modelling Language (UML)

UML is a pictorial language developed to specify, visualise, construct and document a software system.

Conceptual Model of UML is made up of concepts and relationships and it helps in understanding real life entities and their interaction.

- UML building blocks representing things, relationships and diagrams
- Rules to connect the building blocks
- Common mechanisms of UML

UML can represent Objects and the basic concepts of encapsulation, abstraction, inheritance and polymorphism. To develop the UML diagrams the system requirements must be defined and each object and its functions identified.

The following are UML Building Blocks that also give an indication of the stages of System Development (Design, Implementation, Process and Deployment).

Class Diagram – static view denotes minimum properties for understanding, shows collaboration and describes functionality required. It supports the design stage of

STOCK
- stockNo: String # qty: Integer -description: String
+ display() - stockValue()
Responsibilities

Name of Class

Attributes

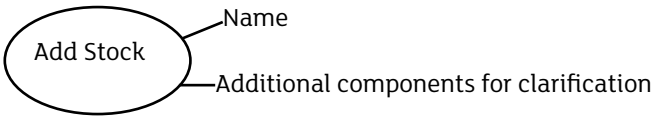
Visibility
+ public
# protected
-private

Operations

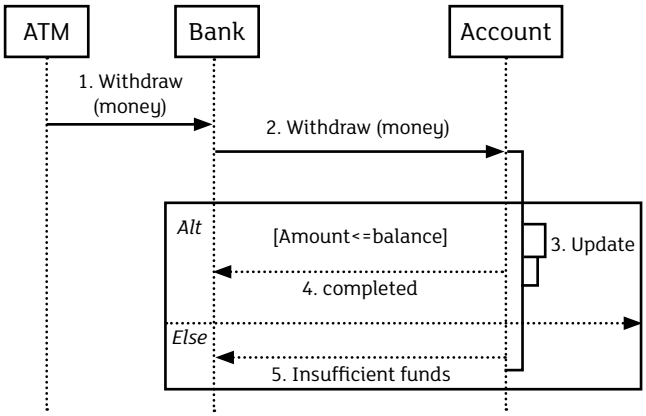
Extra component, not mandatory

Object Diagram- same notation as Class diagram but the name is underlined. This depicts an instance of time and is only used when impacting on the system.

Use Case- represents the functionality of the system and is used to, gather the requirements, get an inside and outside view, determine internal factors and show interaction with actors. It is a high level of design and supports all four levels of development.



Sequence Diagram- A type of Interaction diagram denoting dynamic behaviour, structure and interaction.



Component Diagram supports the Implementation phase.

Deployment Diagram supports the deployment phase and is used by the engineers.

Package Diagram depicts the architecture.

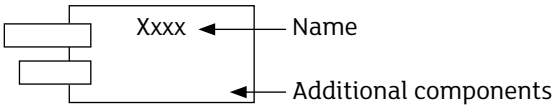
Basic notations for things and Relationships

Actor -  internal or external entity that interacts with the system.

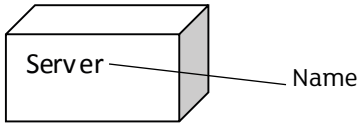
Initial State - indicates start of a process.

Final State - indicates end of a process.

Component



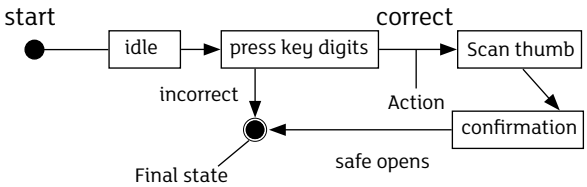
Node - represents a physical component



Behavioural

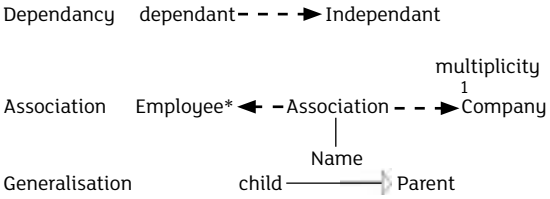
StateChart (also call state machine) describes different states of a component in its life cycle.

For example: A keypad security with thumb scan.



A **Package** groups UML models. As the diagrams build up in the system they can be organised using the package model. See examples on the internet.

Relationships

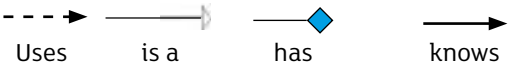


UML Diagrams for use with:

Structural - Class, Object, Component and Deployment;

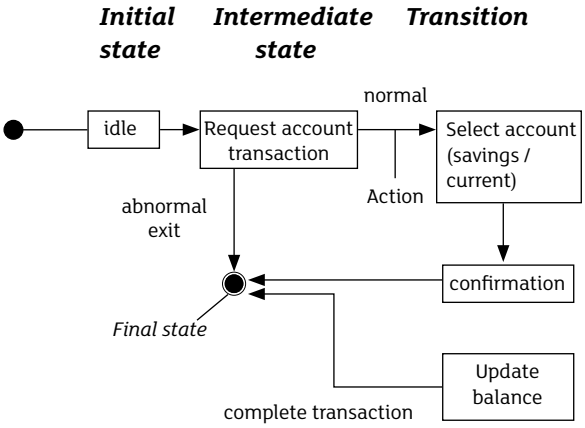
Behavioural- Use case, Sequence, Collaborative, Statechart and Activity.

Links

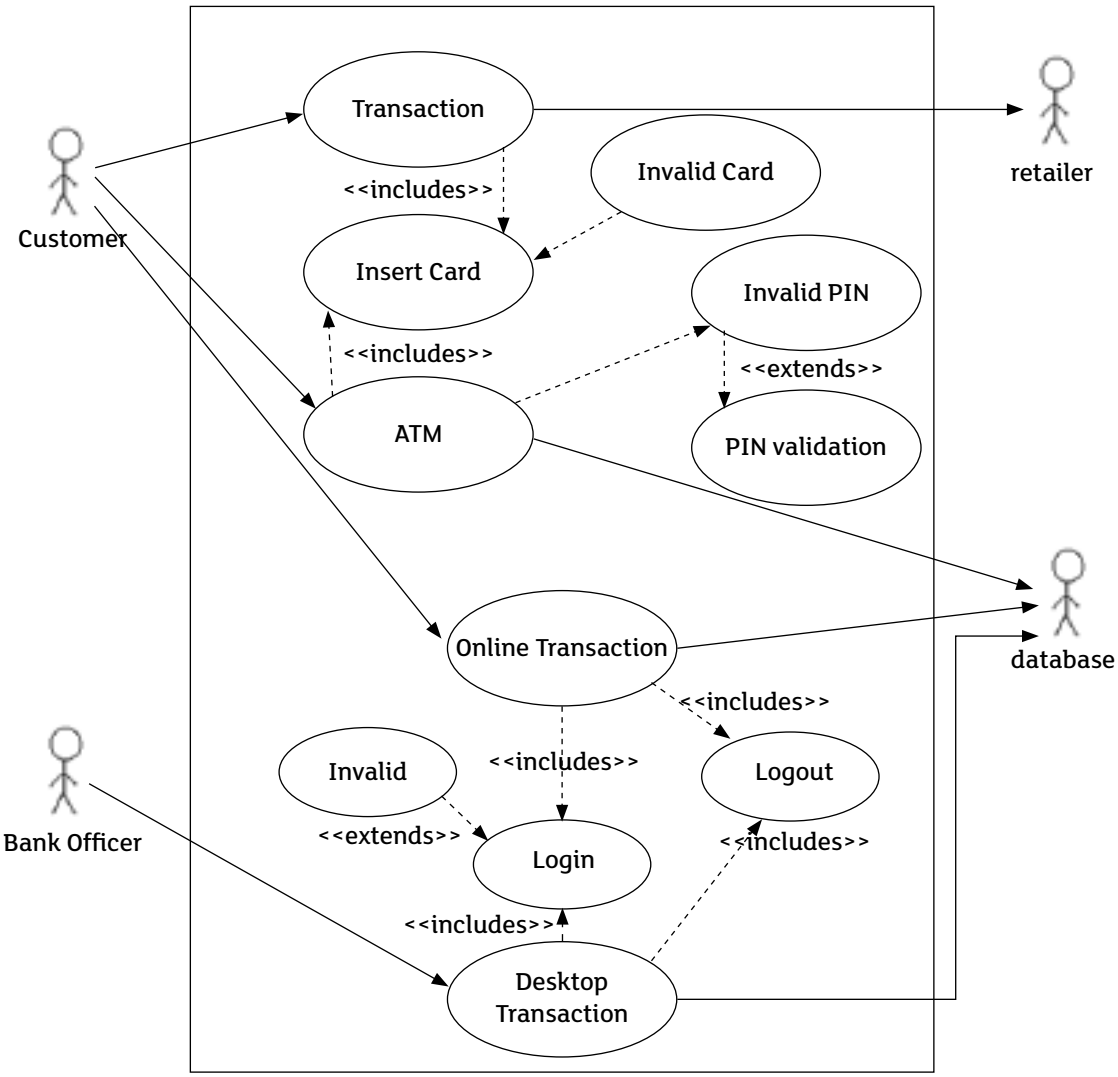


Examples

Statechart diagram for Account Management system

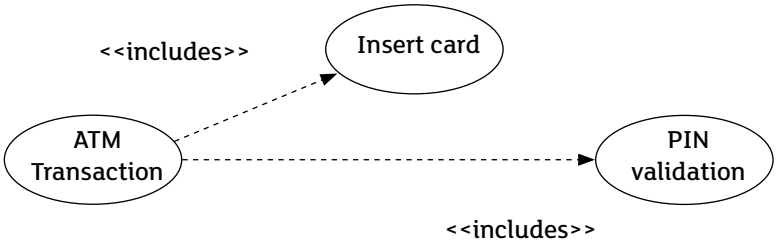


Use Case Diagram –



Customer may initiate transaction – Transfer, deposit, withdrawal, balance transfer. Customer may use ATM transaction. Customer may do an On-line transaction. Retailer may initiate transfer for customer sale / refund.
Bank employee may initiate transaction –open, stop, close account.
Bank database supports actions.

ATM Transaction has a dependency on the Insert Card and Pin Validation



Extending Use case –example, checking card may result in Invalid card process.