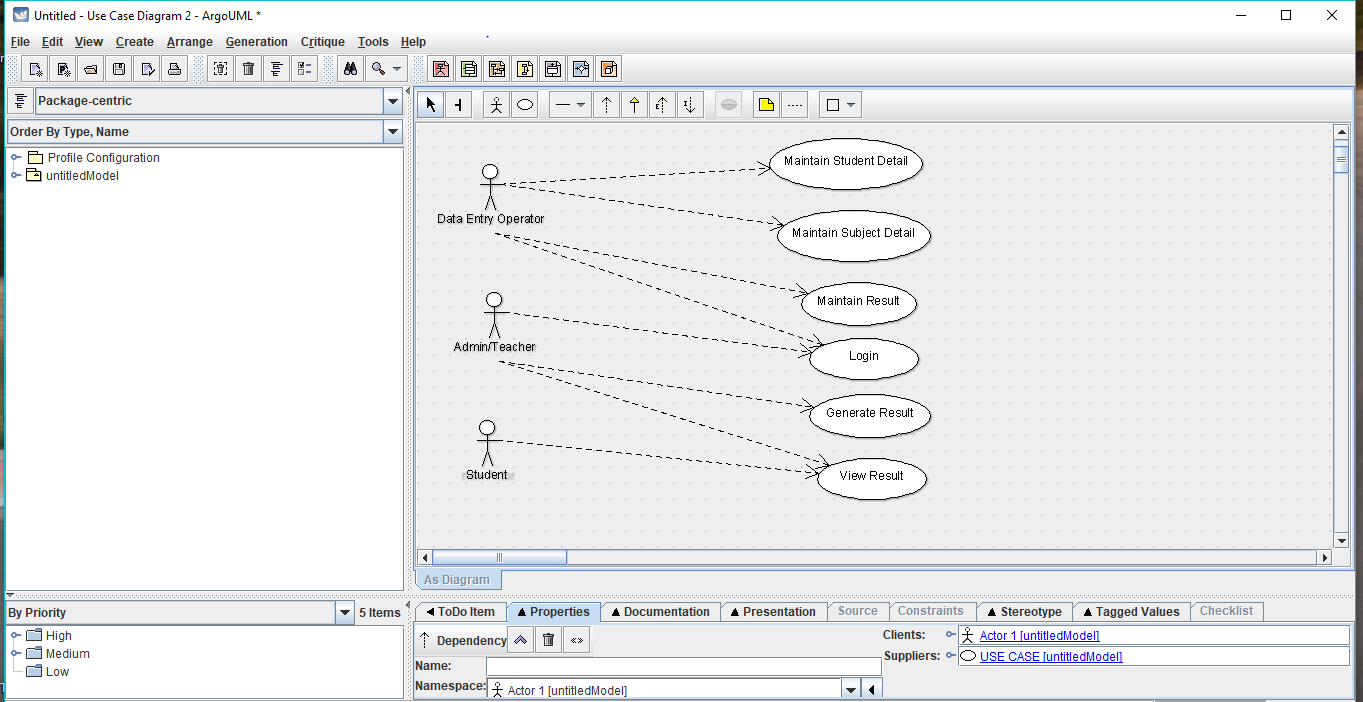
USE CASE DIAGRAM

Use case diagram is a behavioral UML diagram type and frequently used to analyze various systems. They enable you to visualize the different types of roles in a system and how those roles interact with the system.

Although use case diagrams can be used for various purposes there are some common guidelines you need to follow when drawing use cases.These include naming standards, directions of arrows, the placing of use cases, usage of system boxes and also proper usage of relationships.

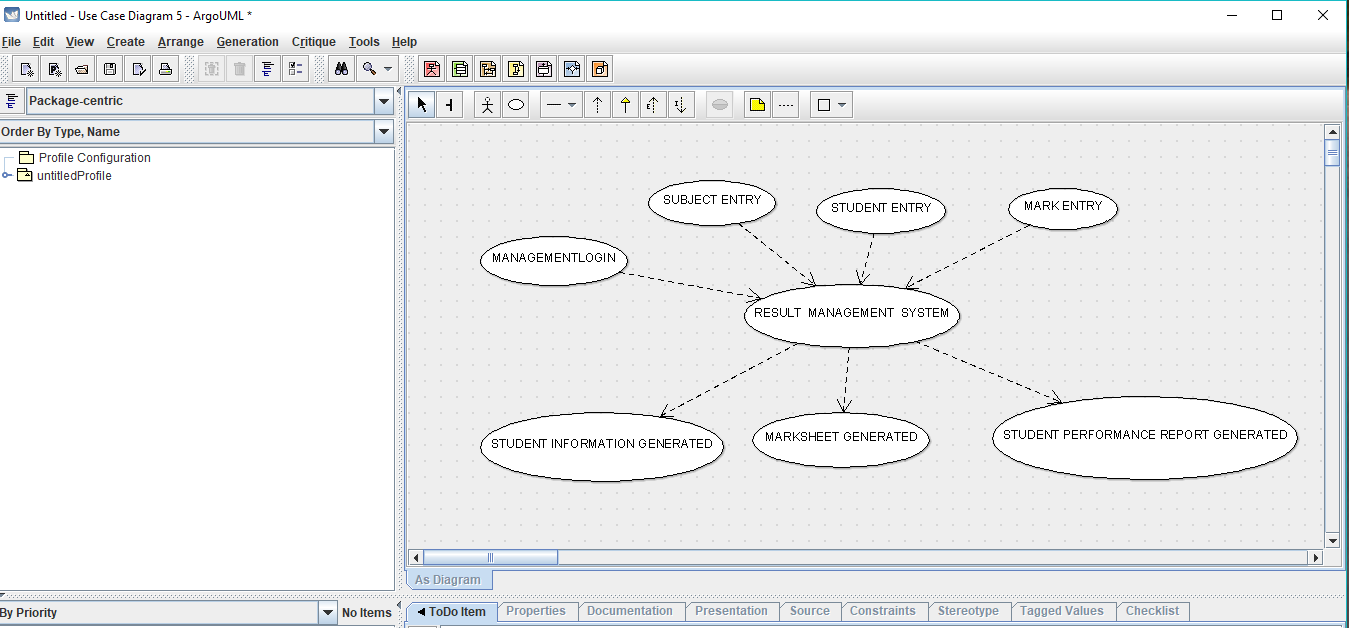
Result Management System Using UseCase Diagram

LEVEL O DFD DIAGRAM

Level 0 Dfd is known as context diagram.

A context diagram is a data flow diagram that only shows the top level, otherwise known as Level 0. At this level, there is only one visible process node that represents the functions of a complete system in regards to how it interacts with external entities. Some of the benefits of a Context Diagram are:

1. Shows the overview of the boundaries of a system
2. No technical knowledge is required to understand with the simple notation
3. Simple to draw, amend and elaborate as its limited notation

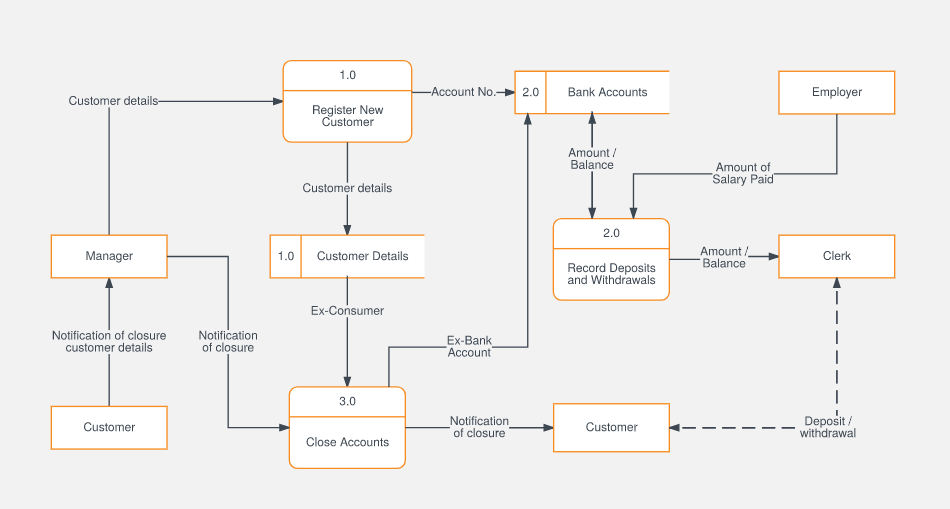


Result Management System Using 0 Level DFD

LEVEL 1 DFD DIAGRAM

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an “exploded view” of the context diagram.

* It provides a more detailed view of the Context Level Diagram.
* Here, the main functions carried out by the system are highlighted as we break into its sub-processes

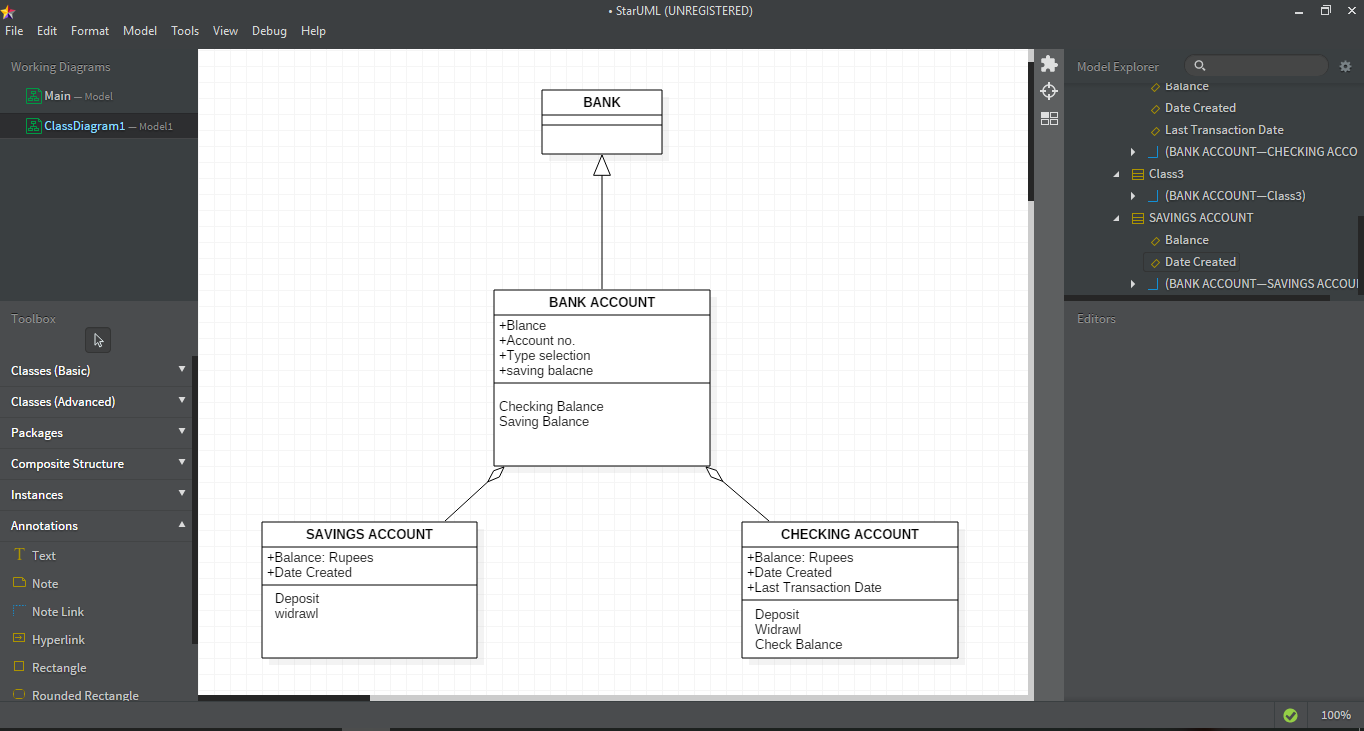


LEVEL 1 DFD FOR BANK MANAGEMENT SYSTEM

CLASS DIAGRAM

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

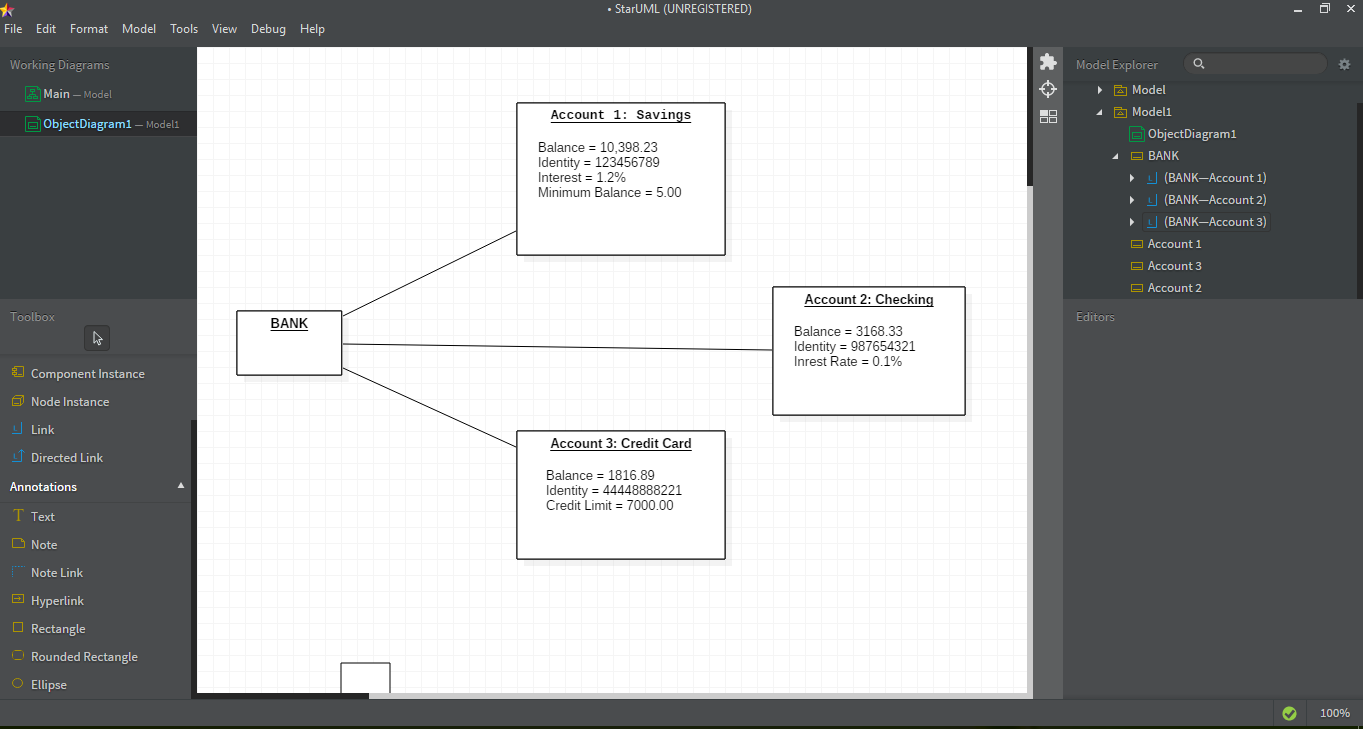


CLASS DIAGRAM OF BANK SYSTEM

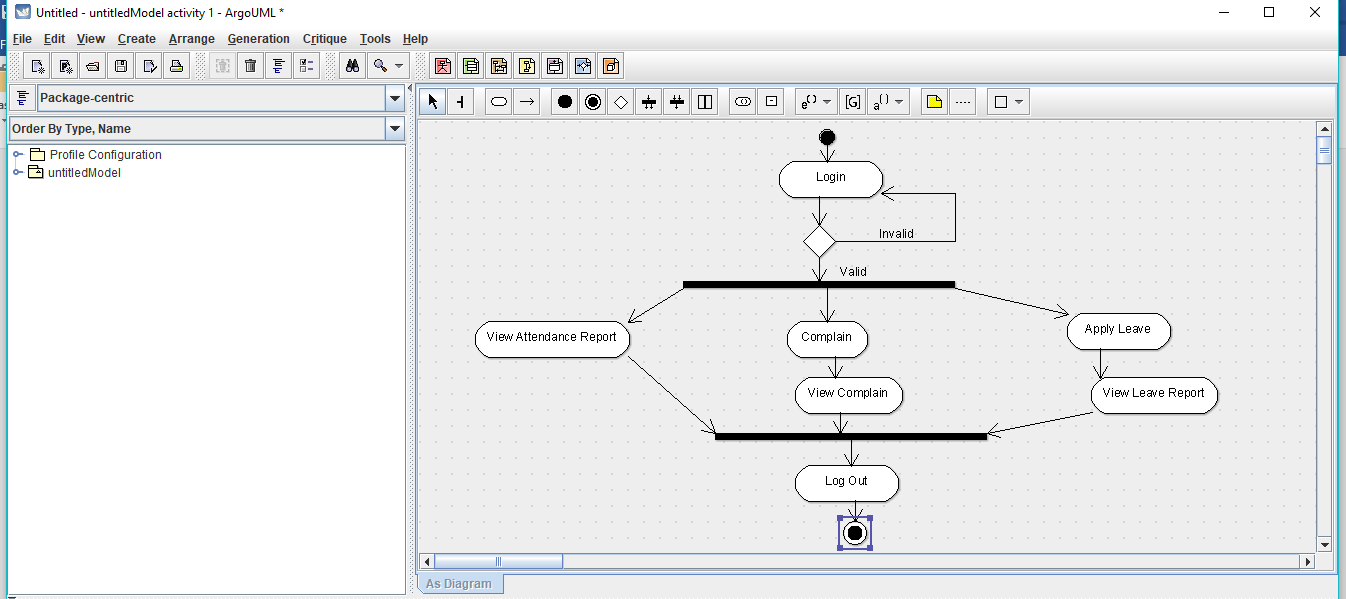
OBJECT DIAGRAM

"An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time. The use of object diagrams is fairly limited, namely to show examples of data structure.

Object diagrams and class diagrams are closely related and use almost identical notation. Both diagrams are meant to visualize static structure of a system. While class diagrams show classes, object diagrams display instances of classes (objects).Object diagrams are more concrete than class diagrams.



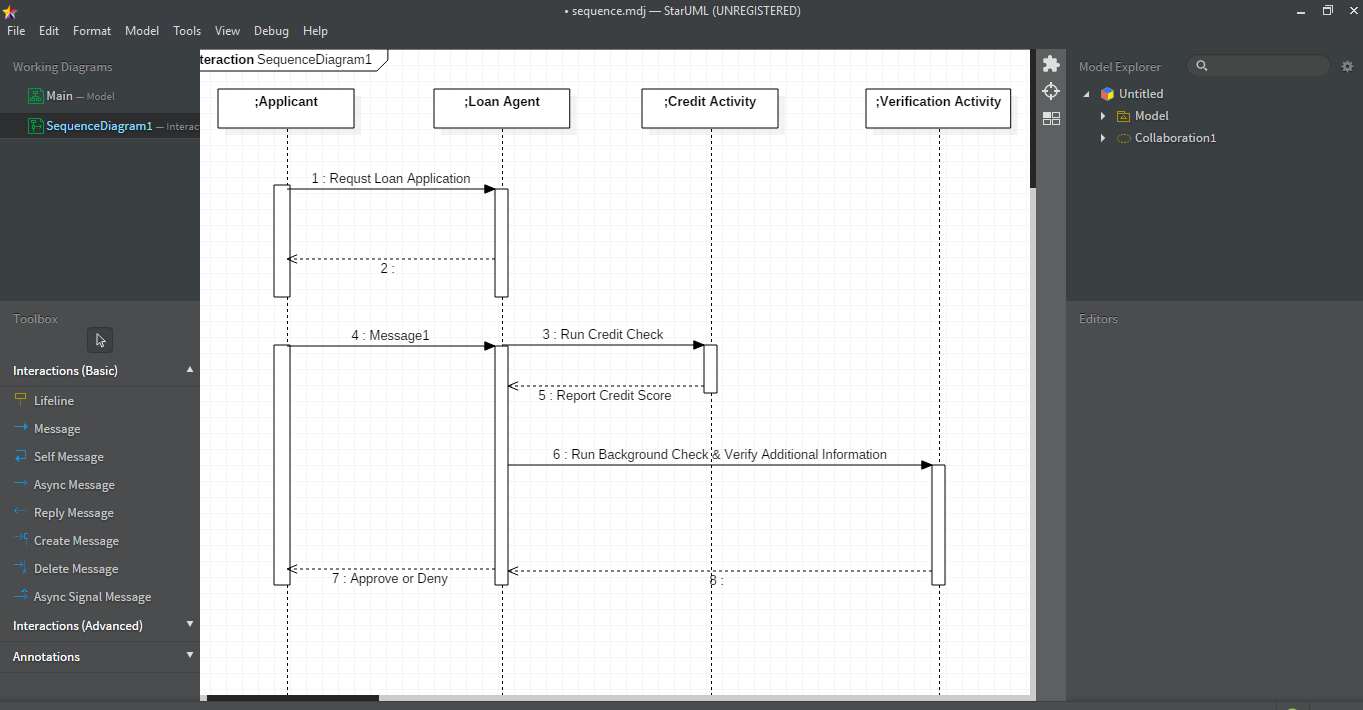
OBJECT DIAGRAM OF BANKING SYSTEM



SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

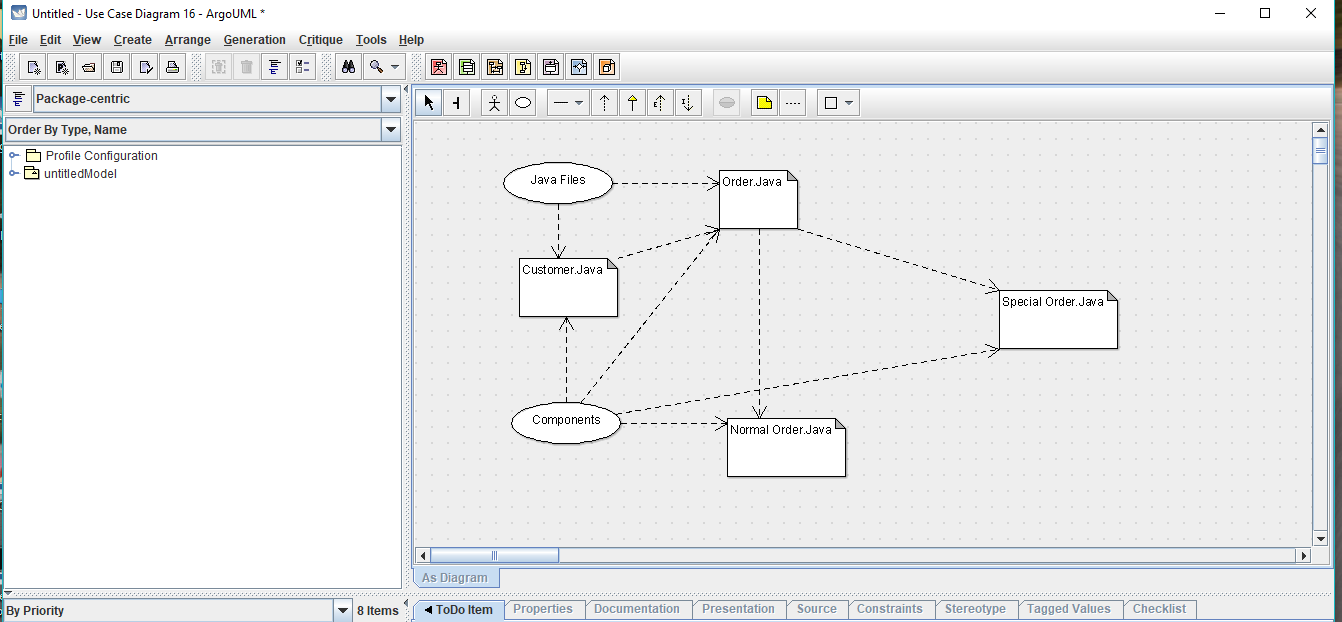
.

Sequence Diagram for Loan Application System

DEPLOYMENT DIAGRAM

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node, and how the different pieces are connected.

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have subnodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



Student mark analysis Deployment Diagram