Modeling UML Use Case Diagrams and Capturing Use Case Scenarios

Introduction

Use case diagram is a platform that can provide a common understanding for the end-users, developers and the domain experts. It is used to capture the basic functionality i.e. use cases, and the users of those available functionality, i.e. actors, from a given problem statement.

In this experiment, we will learn how use cases and actors can be captured and how different use cases are related in a system.

Continue

Identifying the Requirements from Problem Statements

Introduction

Requirements identification is the first step of any software development project. Until the requirements of a client have been clearly identified, and verified, no other task (design, coding, testing) could begin. Usually business analysts having domain knowledge on the subject matter discuss with clients and decide what features are to be implemented.

In this experiment we will learn how to identify functional and non-functional requirements from a given problem statement. Functional and non-functional requirements are the primary components of a Software Requirements Specification.

Continue

Modeling UML Class Diagrams and Sequence diagrams

Introduction

Classes are the structural units in object oriented system design approach, so it is essential to know all the relationships that exist between the classes, in a system. All objects in a system are also interacting to each other by means of passing messages from one object to another. Sequence diagram shows these interactions with time ordering of the messages.

In this Experiment, we will learn about the representation of class diagram and sequence diagram. We also learn about different relationships that exist among the classes, in a system.

From the experiment of sequence diagram, we will learn about different types of messages passing in between the objects and time ordering of those messages, in a system.

Continue

Statechart and Activity Modeling

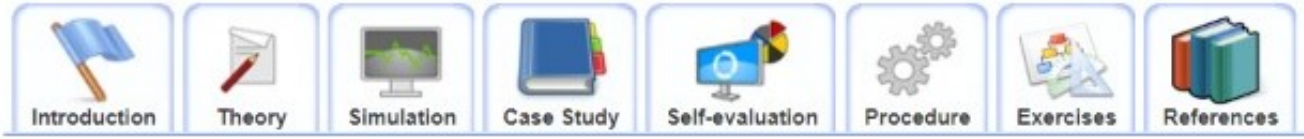
Introduction

Capturing the dynamic view of a system is very important for a developer to develop the logic for a system. State chart diagrams and activity diagrams are two popular UML diagram to visualize the dynamic behavior of an information system.

In this experiment, we will learn about the different components of activity diagram and state chart diagram and how these can be used to represent the dynamic nature of an information system.

Continue

Identifying Domain Classes from the Problem Statements



Introduction

Same types of objects are typically implemented by class in object oriented programming. As the structural unit of the system can be represented through the classes, so, it is very important to identify the classes before start implementing all the logical flows of the system.

In this experiment we will learn how to identify the classes from a given problem statement.

Continue