SENG 350
- Software Architecture & Design

Shuja Mughal

Architecture Description Languages – UML

Fall 2024

University of Victoria

Modeling with UML

University of Victoria

Jens H. Weber

Overview: modeling with UML

- What is modeling?
- What is UML?
- Use case diagrams
- Class diagrams
- Sequence diagrams
- Activity diagrams

University of Victoria

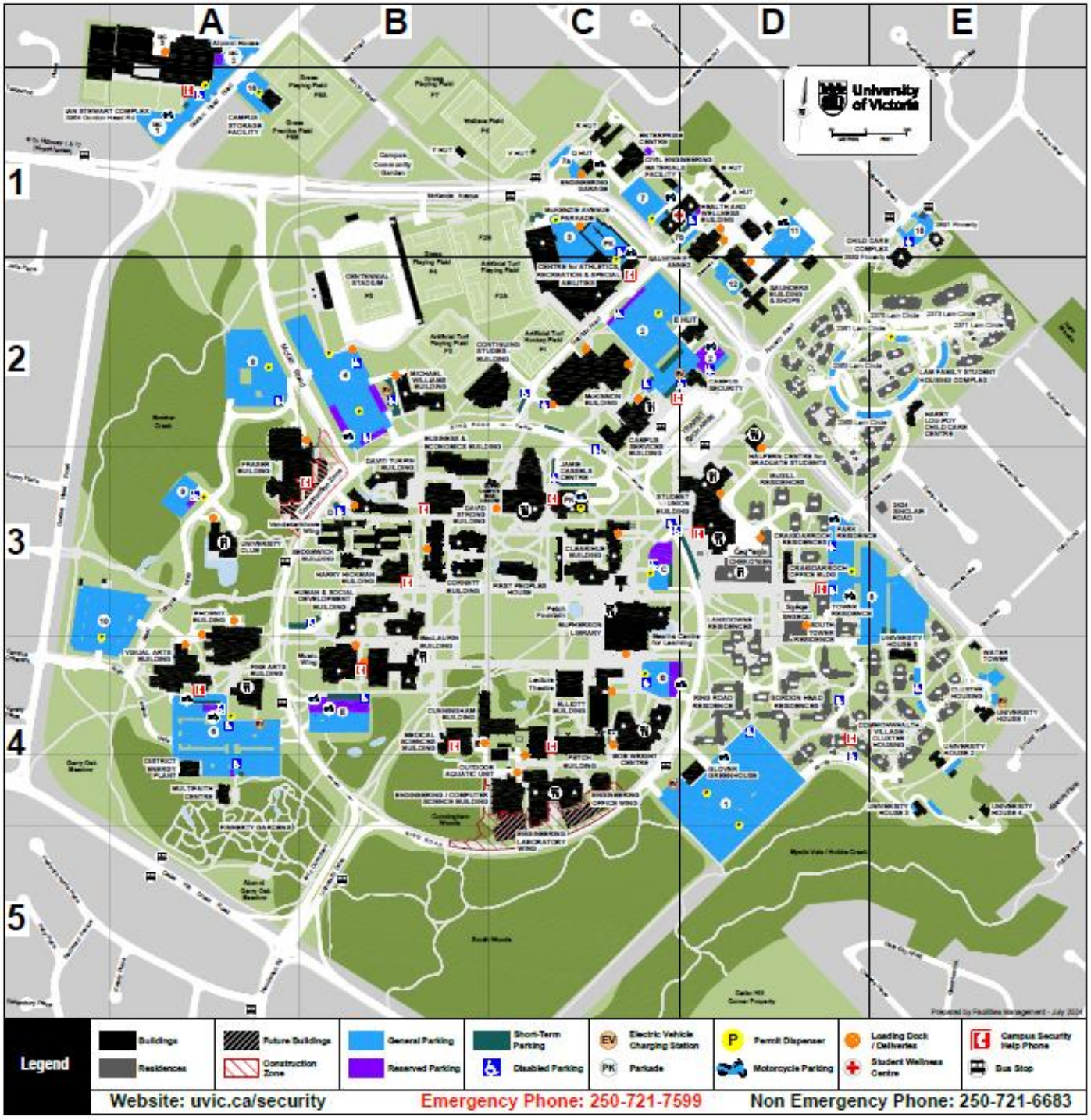
What is modeling?

- Modeling consists of building an abstraction of reality.
- Abstractions are simplifications because:
  - They ignore irrelevant details and
  - They only represent the relevant details.
- What is *relevant* or *irrelevant* depends on the purpose of the model.

University of Victoria

University of Victoria | Campus Security Services

University of Victoria Parking Guide Map



Example: A street map

University of Victoria

Why model software?

Why model software?

- Software is getting increasingly more complex
  - Windows = Millions of lines of code
  - A single programmer cannot entirely manage this amount of code.
- Code is not easily understandable by developers who did not write it
- We need more straightforward representations for complex systems
  - Modeling is a means of dealing with complexity

What is UML?

- UML (Unified Modeling Language)
- An emerging standard for modeling software systems.
- Supported by several CASE tools
  - Visio
  - Rational ROSE
  - TogetherJ
  - Many More

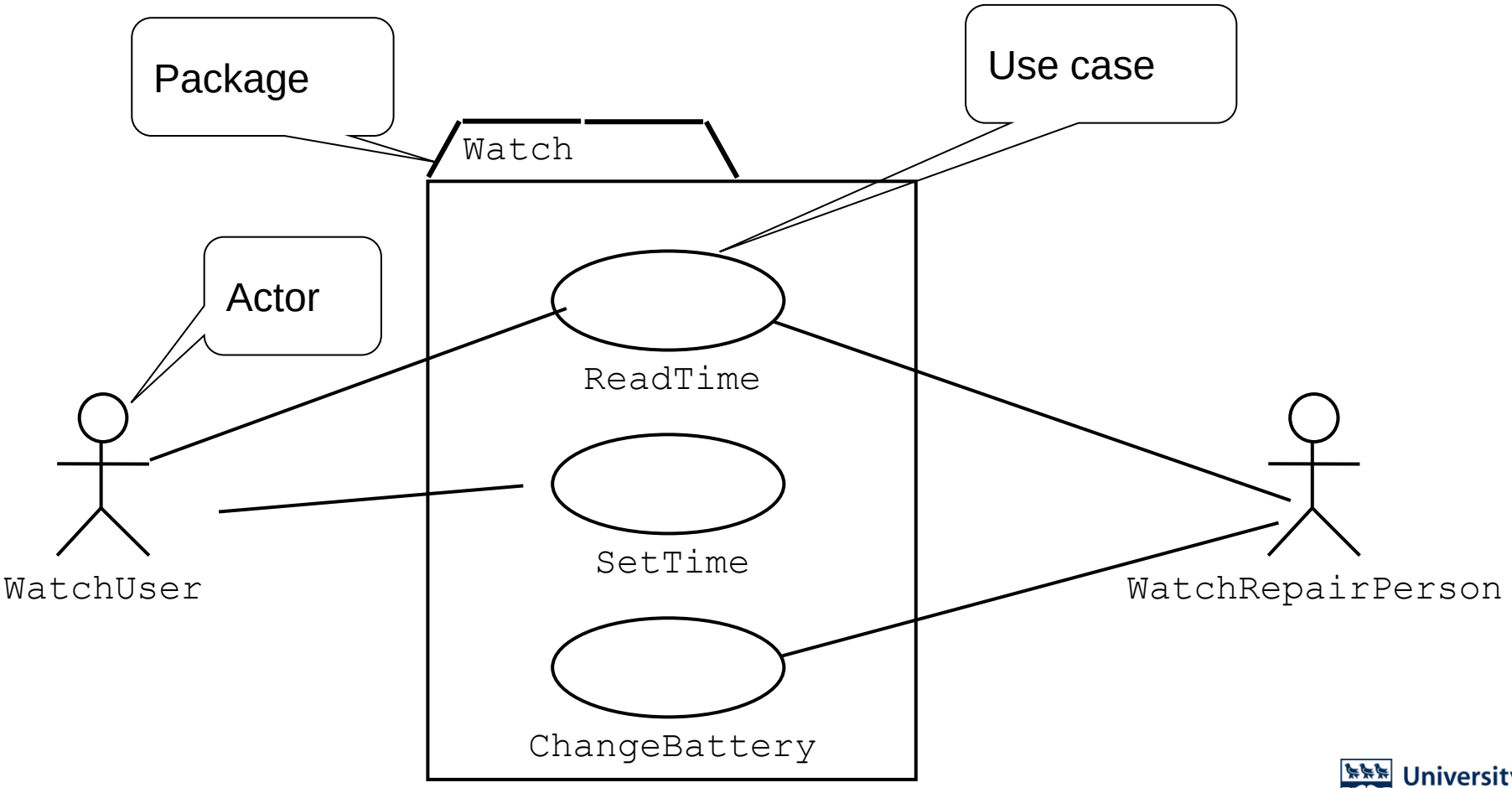
UML Diagrams

- Use case Diagrams
- Class diagrams
- Sequence diagrams
- Statechart diagrams
- Activity Diagrams

UML Diagrams

- **Use case Diagrams**
  - Describe the functional behaviour of the system as seen by the user.
- **Class diagrams**
  - Describe the static structure of the system: Objects, Attributes, Associations
- **Sequence diagrams**
  - Describe the dynamic behavior between actors and the system and between objects of the system
- **Statechart diagrams**
  - Describe the dynamic behavior of an individual object (essentially a finite state automaton)
- **Activity Diagrams**
  - Model the dynamic behavior of a system, in particular, the workflow (essentially a flowchart)

Use case diagrams

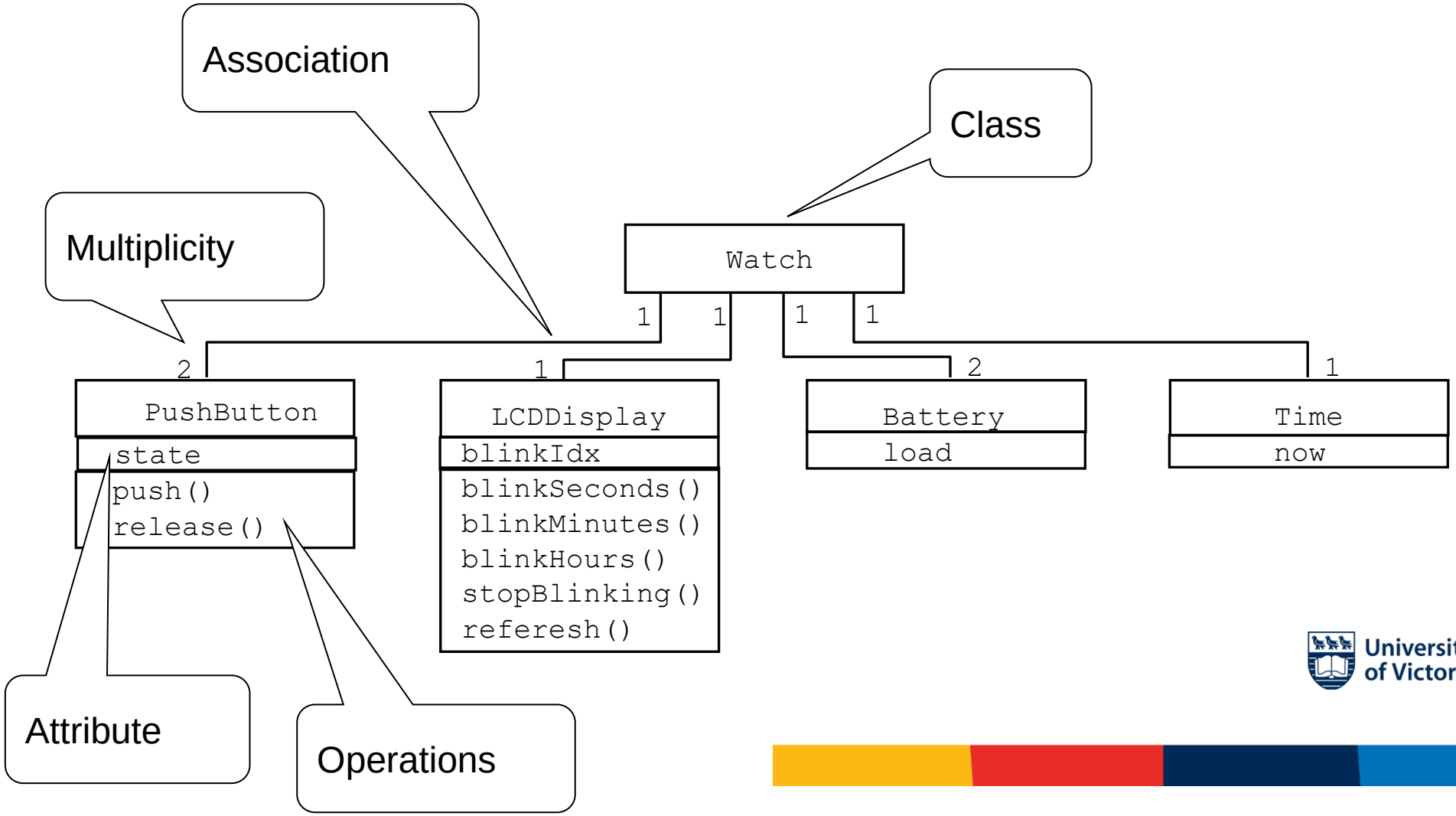


University of Victoria

Use case diagrams represent the functionality of the system from the user's point of view.

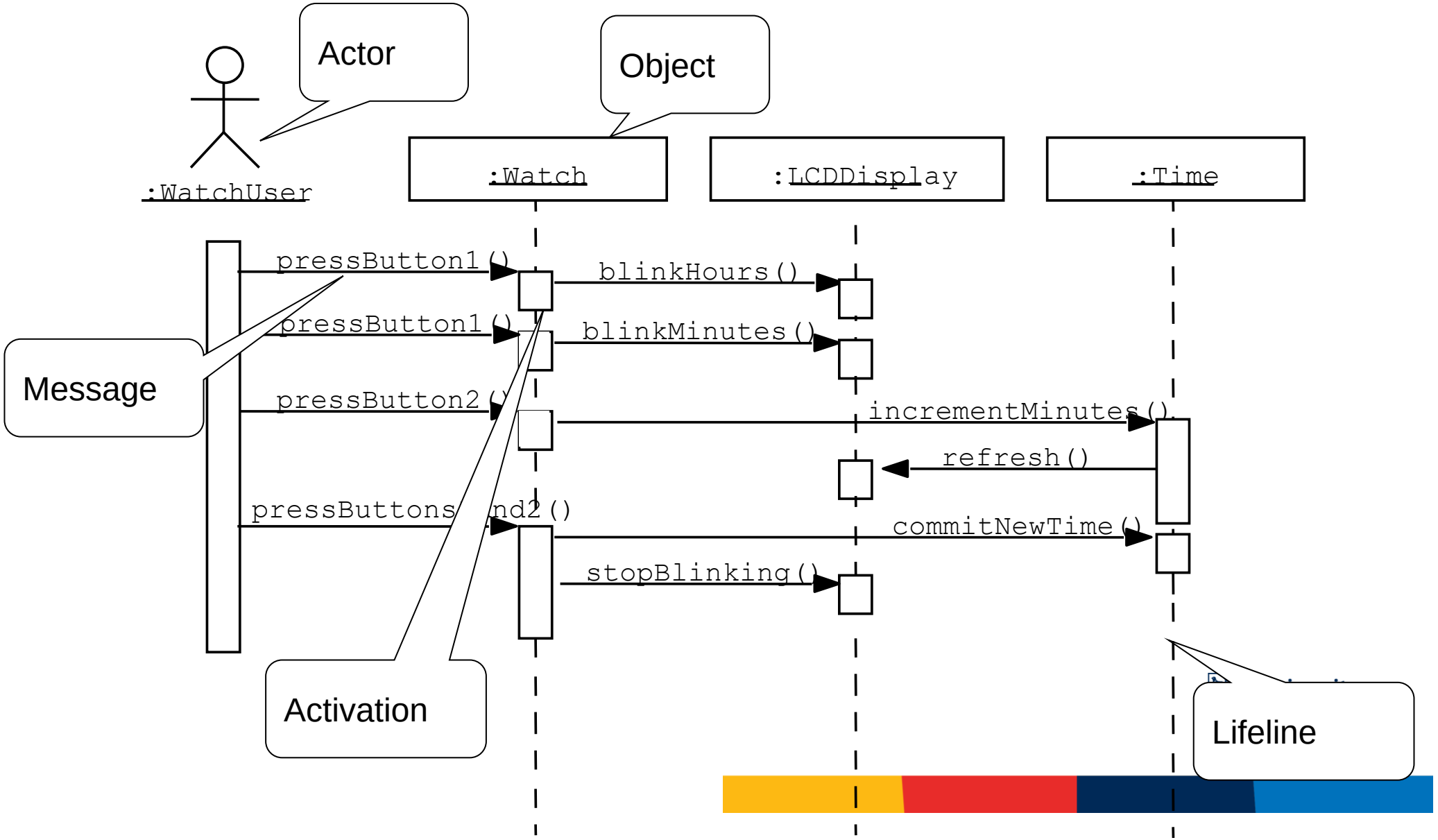
Class diagrams

Class diagrams represent the structure of the system



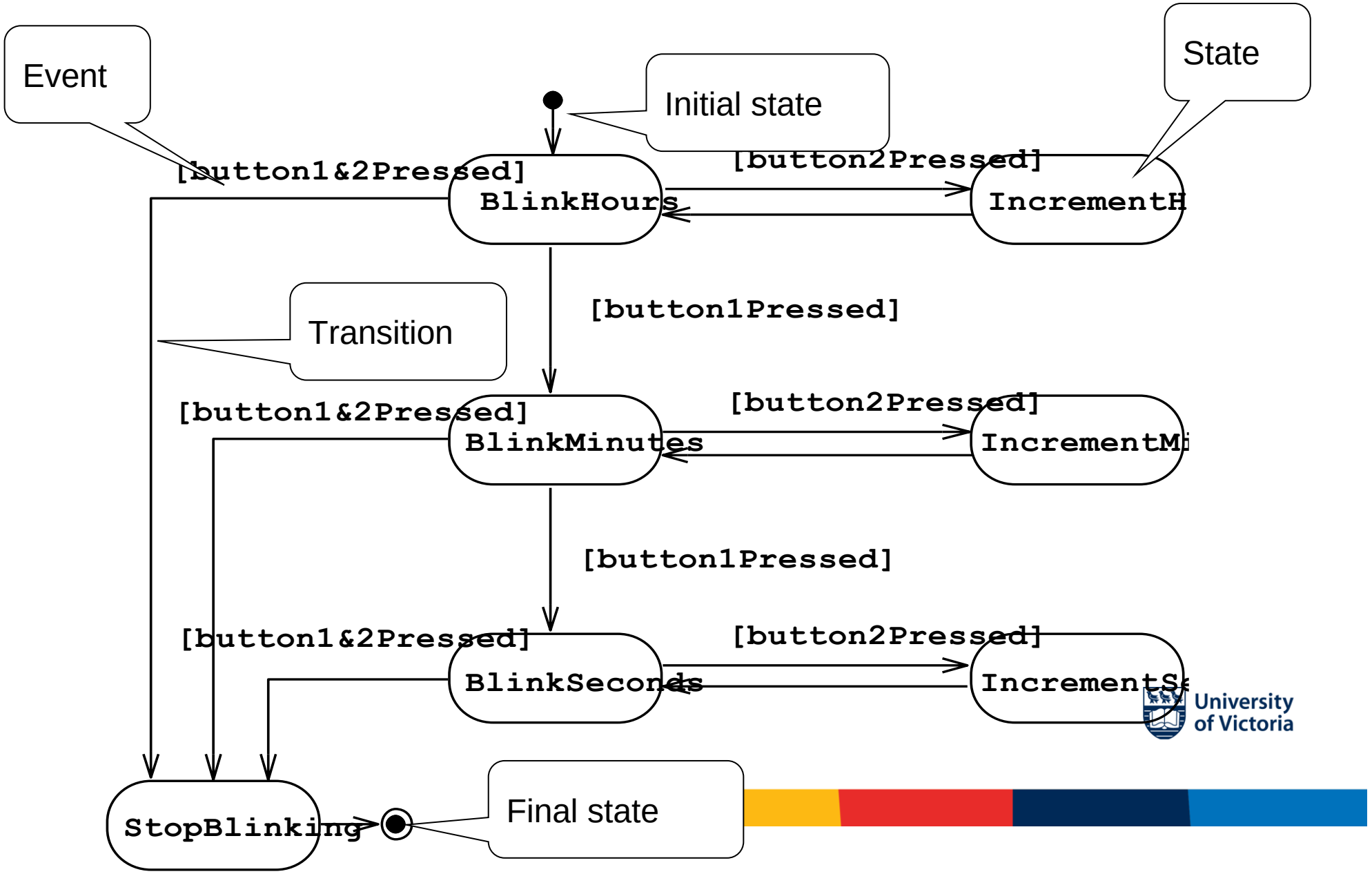
University of Victoria

Sequence diagram



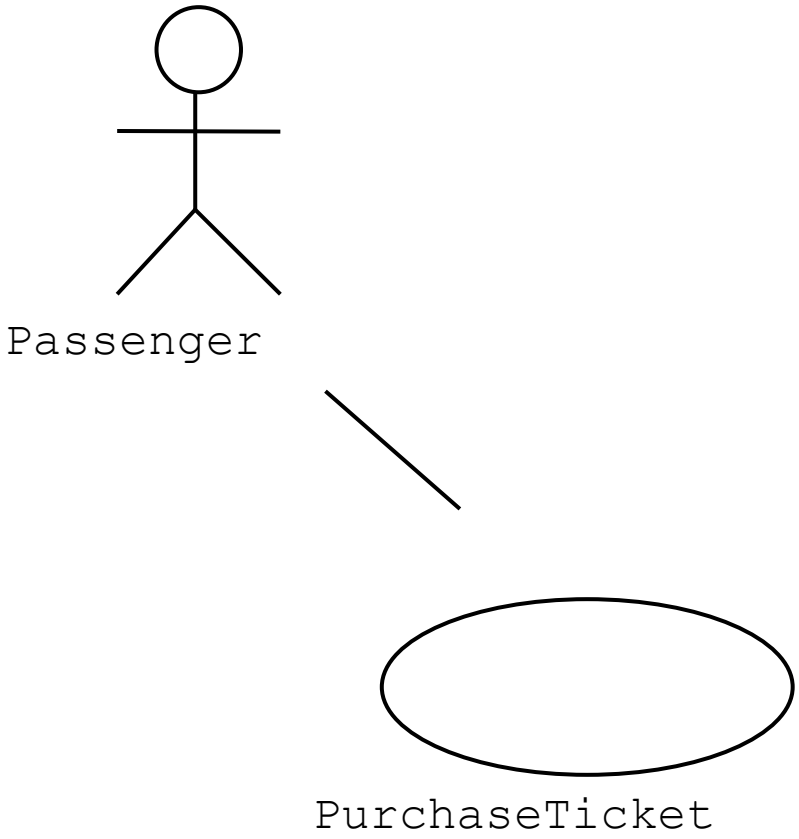
Sequence diagrams represent the behavior as interactions

Statechart diagrams



Represent behavior as states and transitions

Use Case Diagrams

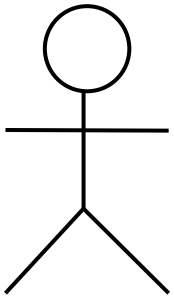


Used during requirements elicitation to represent external behavior

Actors represent roles, that is, a type of user of the system
Use cases represent a sequence of interaction for a type of functionality
The use case model is the set of all use cases. It is a complete description of the functionality of the system and its environment

University of Victoria

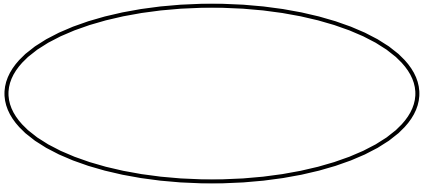
Actors



- An actor models an external entity which communicates with the system:
  - User
  - External system
  - Physical environment
- An actor has a unique name and an optional description.
- Examples:
  - Passenger: A person in the train
  - GPS satellite: Provides the system with GPS coordinates

Use Case

- A use case represents a class of functionality the system provides as an event flow.



PurchaseTicket

- A use case consists of:
- Unique name
- Participating actors
- Entry conditions
- Flow of events
- Exit conditions
- Special requirements

University of Victoria

Use Case Diagram: Example

Name: Purchase ticket

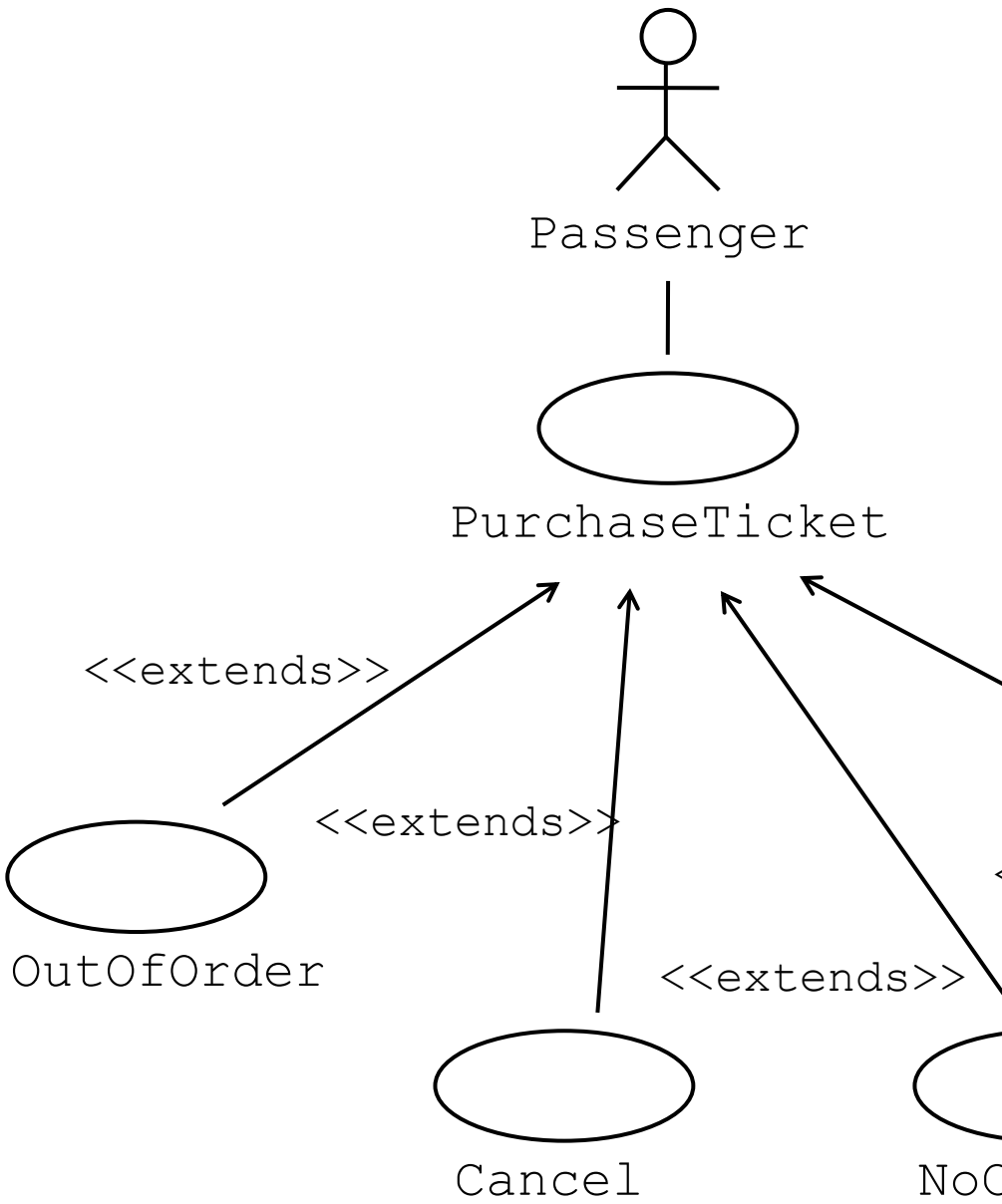
Participating actor: Passenger

Entry condition:
Passenger standing in front of ticket distributor.
Passenger has sufficient money to purchase ticket.

Exit condition:
The passenger has a ticket.

Event flow:
1. Passenger selects the number of zones to be traveled.
2. Distributor displays the amount due.
3. Passenger inserts money, of at least the amount due.
4. Distributor returns change.
5. Distributor issues ticket.

The <<extends>> Relationship

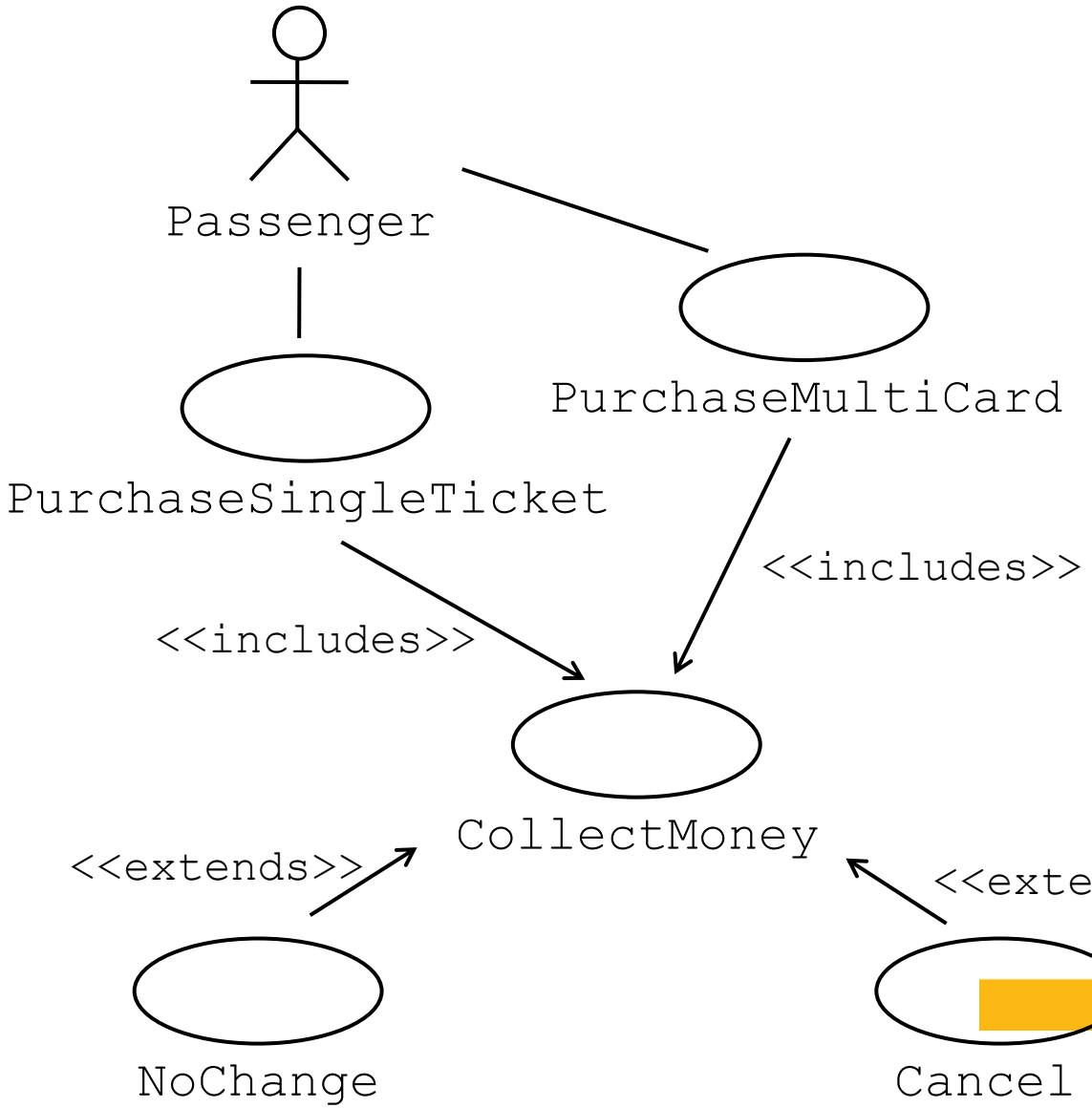


This use case is optional and comes after the base use case.

The extending use case depends on the base use case; it extends the behavior described by the base use case.

University of Victoria

The <<includes>> Relationship



This use case is mandatory and part of the base use case.

A base use case depends on the included use case(s); without it/them, the base use case is incomplete, as the included use case(s) represent sub-sequences of the interaction that may happen always or sometimes.

University of Victoria

Use Case Diagrams: Summary

- Use case diagrams represents external behavior
- Use case diagrams are useful as an index into the use cases
- Use case descriptions provide meat of model, not the use case diagrams.
- All use cases need to be described for the model to be useful.

Case Study

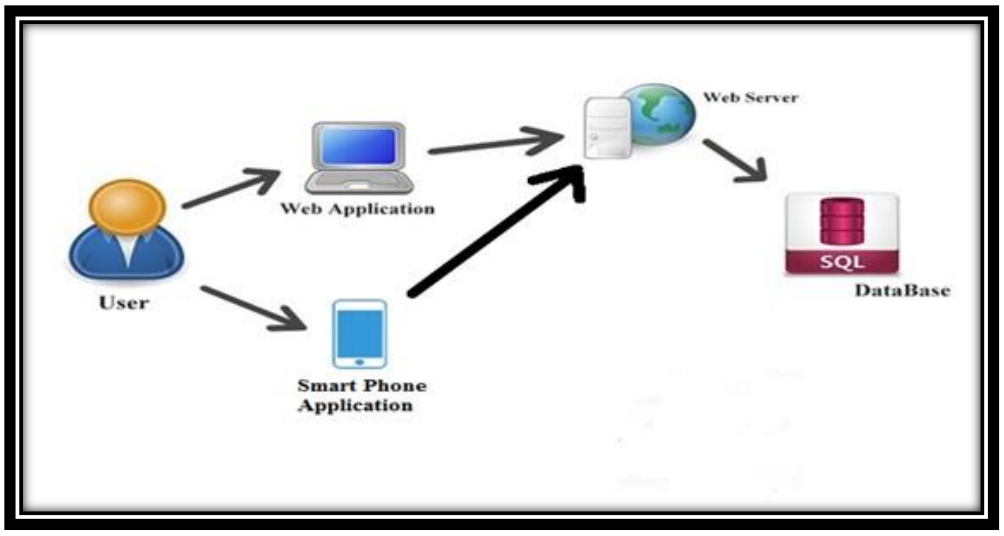
University of Victoria

Jens H. Weber

Case Study

Online School Portal (OSP) is a smartphone and web-based application that targets schools. It facilitates students, teachers, administration and parents. This system provides a platform that automates all the work, which is much more efficient than the legacy systems. In this software, each stakeholder can perform certain tasks to get their job done. Moreover, the application provides ease of use, implementing features of human-computer interaction. OSP includes a website and a smartphone application. The smartphone application and website are combined with a mutual server and a database.

Students using OSP should access various features to enhance their academic journey. They should log in, log out, view their marks, attendance, and faculty details, view homework/assignments, and update their profiles.



University of Victoria

Use Case Diagram

Students using OSP can access various features to enhance their academic journey. They should log in, log out, view their marks, attendance, and faculty details, view homework/assignments, and update their profiles.

University of Victoria

Fully Dress Use Case/ Textual Specification

Scope: Online Student Portal

Level: User Goal

Primary Actor: Student

Secondary Actor: N/A

Stakeholders and interests:

- Student: Wants to view Marks.

Precondition(s):

- Student is logged in to the system.
- The faculty has uploaded and accepted the marks.

Success Guarantee: View the marks successfully.

Candidate	**System**
This use case starts when the student wants to view marks.	
1) Student selects the login option	2) System prompts the student to enter account credentials
3) Student provides the account credentials.	4) The system authenticates the Student
5) Student selects "View Marks "option	6) The system opens the view marks form.

University of Victoria

Upcoming Lecture

- Class diagrams
- Sequence diagrams
- Activity diagrams
- State transition diagrams
- Collaboration diagrams

University of Victoria