Software Design Documentation

By Using UML for OOAD Project

Presented by:

MD ALAMGIR KABIR

Graduate Research Assistant

Big Data and Cloud Computing Laboratory

Date: 7 August 2016

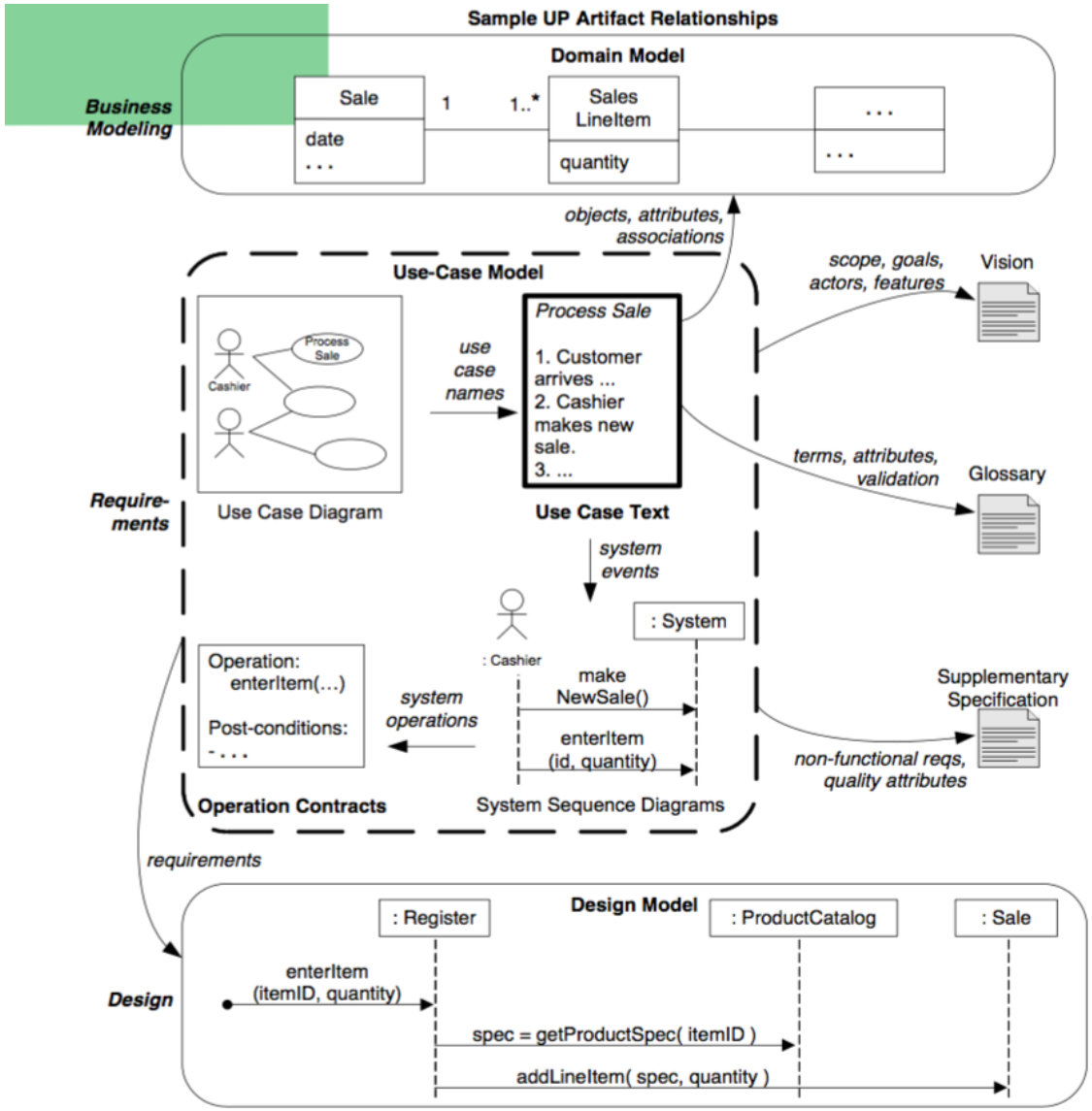
WUHAN UNIVERSITY

International School Of Software

Overview

1. Historical View on Modeling Languages
2. Use Case Diagram
3. Use Case Description
4. System Sequence Diagram
5. Operational Contracts
6. Domain Modeling
7. Class Diagram

In one figure



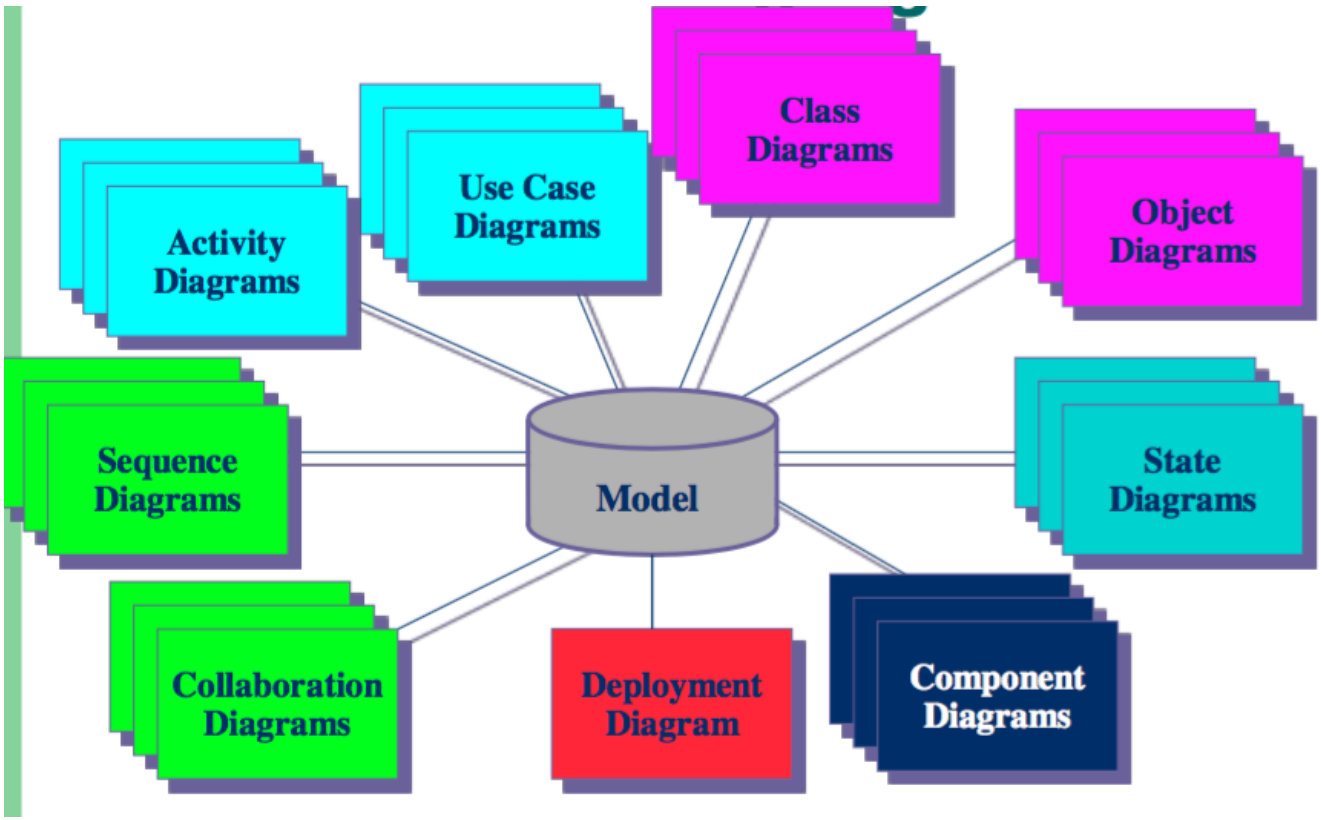
Historical View on Modeling Languages

Historical View on Modeling Languages

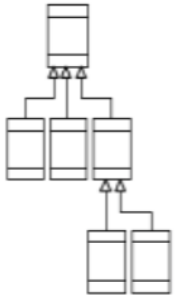
Modeling Languages

- ***1970's:*** *– Process-oriented methods (Structured System Analysis and Design – SSAD):*
  - Use DFDs – Data Flow Diagrams
- ***1980's:*** *– Data-oriented methods:*
  - Use ERDs – Entity-Relationship Diagrams
- ***1990's:*** *– Object-oriented methods (OMT, OOD, OOSE, UP):*
  - Standard: UML – Unified Modeling Language

Unified Modeling Language (UML)

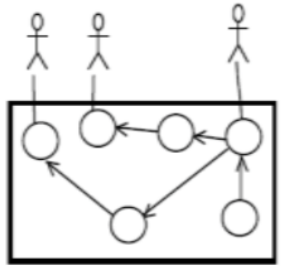


UML Visual Languages



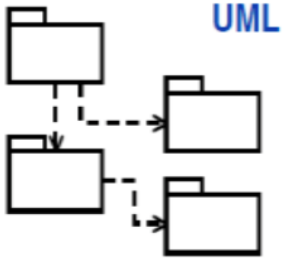
UML Class Diagrams

- information structure
- relationships between data items
- modular structure for the system



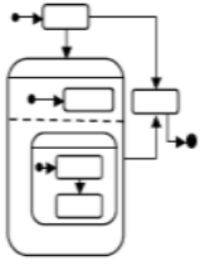
Use Cases

- user's view
- Lists functions
- visual overview of the main requirements



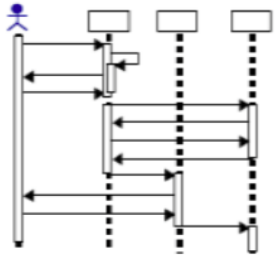
UML Package Diagrams

- Overall architecture
- Dependencies between components



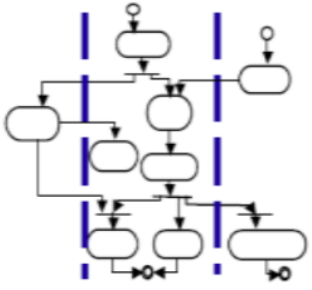
(UML) Statecharts

- responses to events
- dynamic behavior
- event ordering, reachability, deadlock, etc



UML Sequence Diagrams

- individual scenario
- interactions between users and system
- Sequence of messages

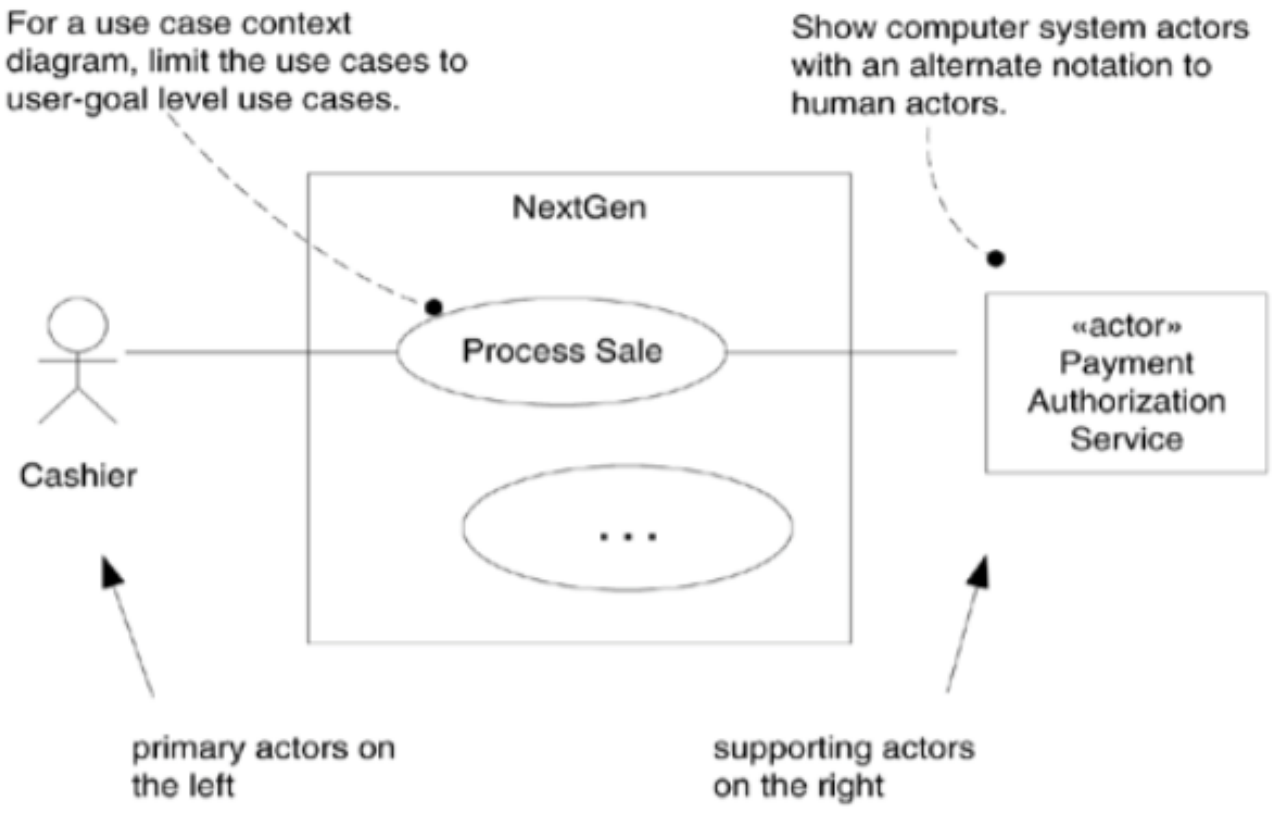


Activity diagrams

- business processes;
- concurrency and synchronization;
- dependencies between tasks;

Use Case Diagram

Use Case Diagram



Use Case Description

Use Case Description

Use Case Description

- Description for each use case
- Success scenario and alternative scenario of each factor

System Sequence Diagram

System Sequence Diagram

- System Sequence Diagram for each use case of success and alternative scenario

Operational Contracts

Operational Contracts

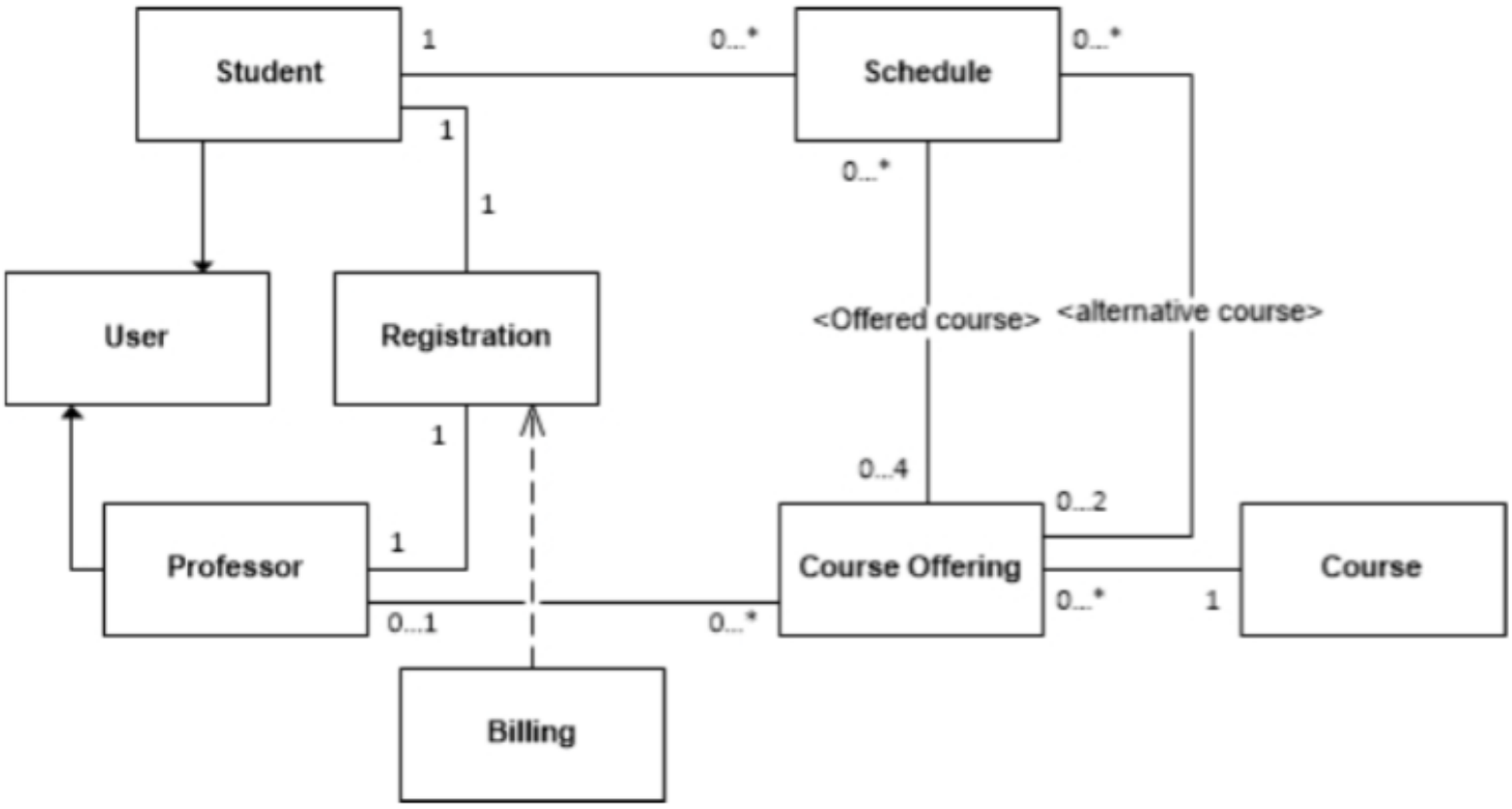
2.4.1 Operation Contracts. Use Case 1

Operation Contracts: enterAccountInfo
Operation: enterAccountInfo(StdLoginName,StdPassword)
Cross References: Use Case 1: Process selecting courses (Success Scenario)
Preconditions: Student has stable internet access to the system
Post conditions:
- A new enterAccountInfo instance eAI was created
- eAI was associated with current request to log into the system
- eAI. StdLoginName becomes variable StdLoginName
- eAI. StdPassword becomes variable StdPassword
- eAI was associated with a StdLoginName,StdPassword

Operation Contracts: selectCourses
Operation: selectCourses(MainCourses,AltCourses)
Cross References: Use Case 1: Process selecting courses (Alternative Flow (a))
Preconditions: Student successfully received course catalog
Post conditions:
- A new selectCourses instance sC was created

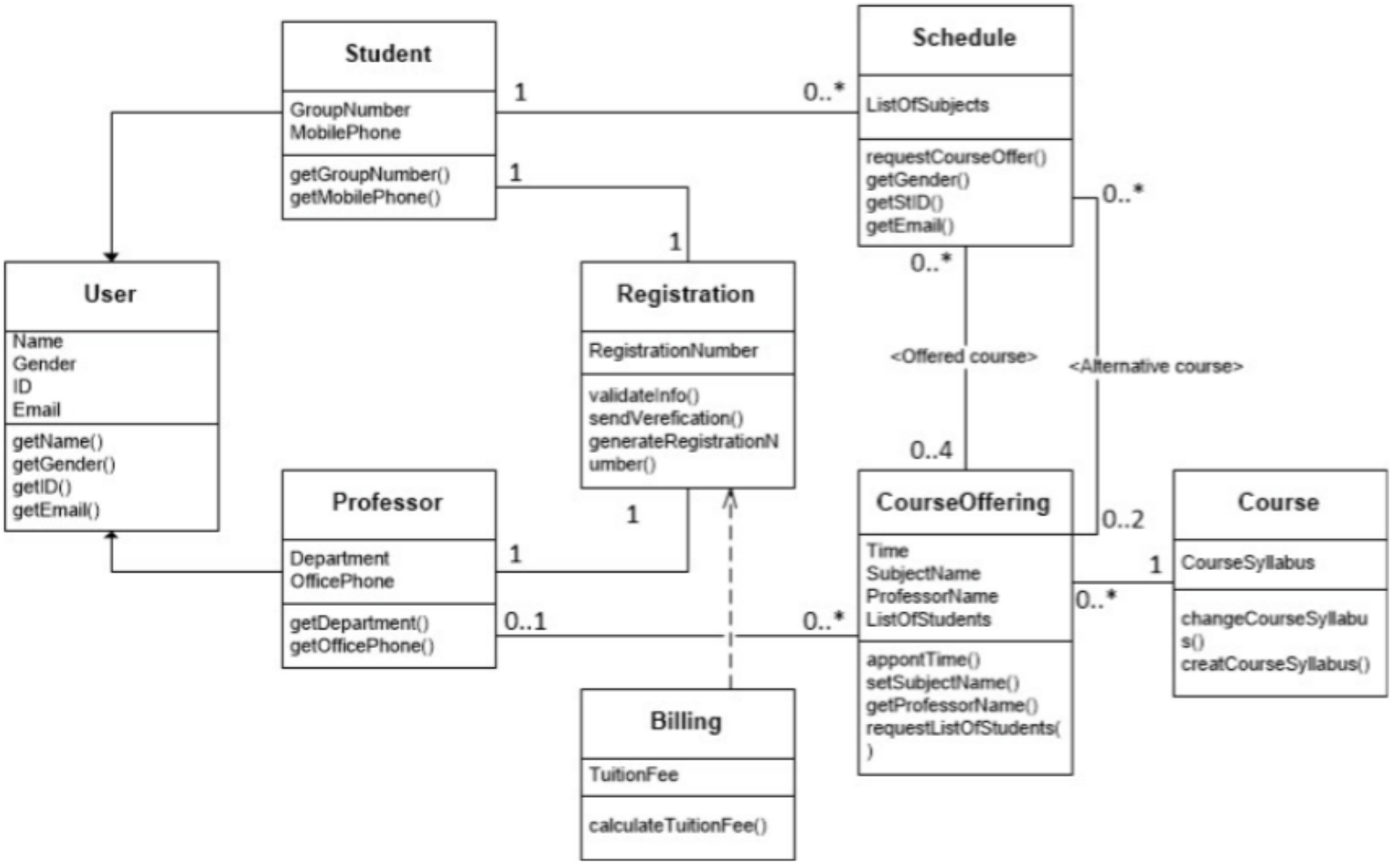
Domain Modeling

Domain Modeling



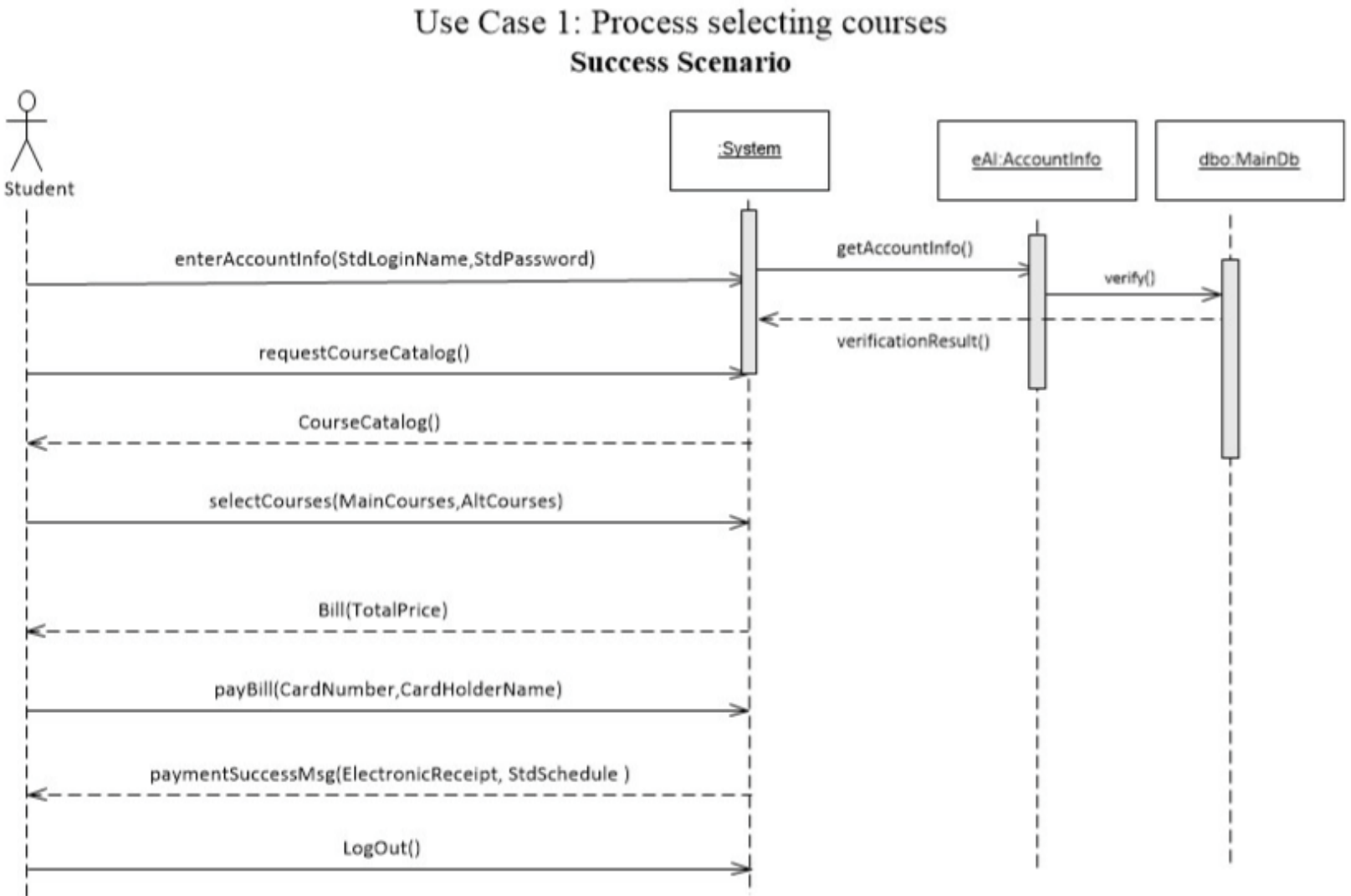
Class Diagram

Class Diagram



Sequence Diagram

Sequence Diagram



Implementation

Implementation

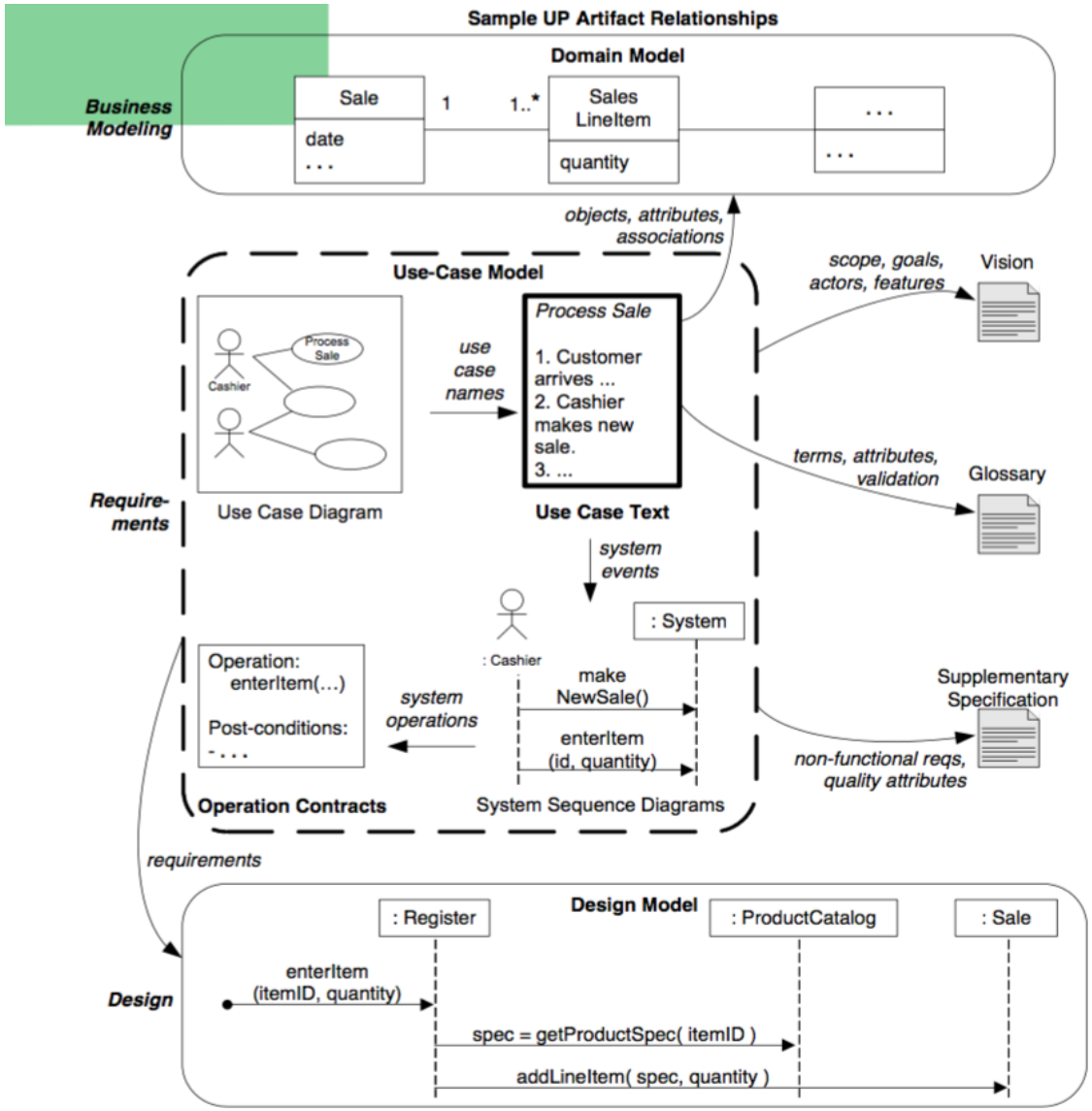
Implementation part

Sum Up

Sum Up /1

- Vision
- Use Case Modeling
  - Use case diagrams
  - Use case description - for success scenario and alternative scenario
  - System Sequence Diagrams - for success scenario and alternative scenario
  - Operation Contracts - for success scenario and alternative scenario
- Domain Modeling
- Class Modeling and Dynamic Modeling
  - Class diagram
  - Sequence diagrams - for success scenario and alternative scenario
- Implementation

Sum Up /2


Sample UP Artifact Relationships
Domain Model
Business Modeling
Sale
date
...
1
1..*
Sales LineItem
quantity
...
...
objects, attributes, associations
Use-Case Model
scope, goals, actors, features
Vision
Process Sale
Cashier
use case names
Process Sale
1. Customer arrives ...
2. Cashier makes new sale.
3. ...
terms, attributes, validation
Glossary
Require-ments
Use Case Diagram
Use Case Text
system events
: System
: Cashier
Operation:
enterItem(...)
Post-conditions:
- ...
system operations
make NewSale()
enterItem (id, quantity)
Supplementary Specification
non-functional reqs, quality attributes
Operation Contracts
System Sequence Diagrams
requirements
Design Model
: Register
: ProductCatalog
: Sale
Design
enterItem (itemID, quantity)
spec = getProductSpec(itemID)
addLineItem(spec, quantity)

References

- Book: Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development, 3rd Edition by Craig Larman

End of Presentation