

Systems for Design and Implementation

2015-2016

Course 1

Contents

- ▶ Distributed applications, client-server applications, web applications
- ▶ Client-server applications based on TCP communication, threads (C# , Java)
- ▶ Remote procedure call based client-server applications (C#, Java)
- ▶ Reflection (C#, Java)
- ▶ Database access components, ORM, Hibernate (C#, Java)
- ▶ XML language, XML documents, XML processing (C#, Java)
- ▶ WebSockets
- ▶ Spring Framework, Remoting, Web
- ▶ SOAP, Web Services
- ▶ Service Oriented Architecture, OSGi, iPOJO
- ▶ Web Patterns
- ▶ etc.

Bibliography

- ▶ Albahari, J. Albahari, B., C# 4.0 in a Nutshell, 4th Ed, O'Reilly, 2010.
- ▶ Larman, C., Applying UML and Design Patterns: An Introduction to OO Analysis and Design and Unified Process, 3rd Ed, Prentice Hall, 2005.
- ▶ Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
- ▶ Hohpe, G., Woolf, B., Enterprise integration patterns, Addison-Wesley, 2003.
- ▶ ***, Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>
- ▶ ***, The Java Tutorials, 2010. <http://download.oracle.com/javase/tutorial/>

Lab requirements

- ▶ Choose a client-server application with distributed observer (max. 5 use-cases: authentication, search, modify, ...)
- ▶ Design the chosen application using a CASE tool (at your choice)
- ▶ Implement the design using C# or Java.
- ▶ Implement the design using Java or C#.
- ▶ Implement de design using C# and Java (Java client – C# server or C# client – Java server)
- ▶ Add a web part to your solution (one use-case)
- ▶ You have to use Reflection, TCP/Remoting/RMI, XML, databases, text files, etc.

Final Mark

- ▶ Lab mark:
 - ▶ design 15 %
 - ▶ each implementation 20 %
 - ▶ web part 15%
 - ▶ lab activity 10 %
- ▶ Exam: exam (written and practical) during the session
- ▶ Final mark:
 - ▶ 40 % lab, 60 % exam

Software Development

► Phases:

- Requirements
- Analysis
- Design
- Implementation
- Testing
- Maintenance

Requirements

Requirements are descriptions of how a software product should perform.

A requirement typically refers to some aspect of a new or enhanced product or service.

Types of requirements:

▸ **Functional:**

- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- features, capabilities, security.

Requirements

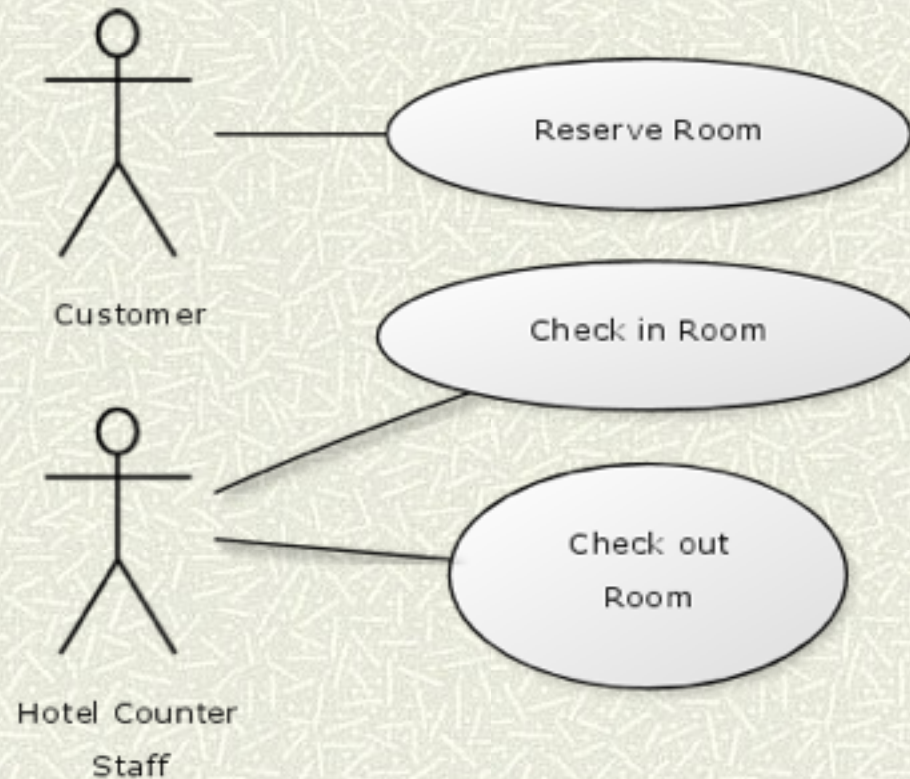
► Non-functional:

- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Usability - human factors, help, documentation.
- Reliability - frequency of failure, recoverability.
- Performance - response times, accuracy, usage.
- Supportability - adaptability, internationalization.

Requirements - Use cases

- ▶ Use cases are functional requirements that indicate what the system should do.
- ▶ Use cases define a promise or contract of how a system will behave.
- ▶ Use cases diagram.
- ▶ Use case specification

Use cases - example



Use Case Specification -outline

- ▶ Brief Description
- ▶ Actors
- ▶ Basic Flow of Events
- ▶ Alternative Flows
- ▶ Preconditions
- ▶ Post-conditions
- ▶ Special Requirements

Use Case Specification: Reserve Room

Brief Description

This use case describes how the customer reserves a room.

Basic Flows

B1. Reserve Room

The use case begins when a customer wants to reserve a room.

1. The customer selects to reserve a room.
2. The system displays the types of rooms the hotel has and their rates.
3. The customer Check Room Cost.
4. The customer makes the reservation for the chosen room.
5. The system deducts from the database the number of rooms of the specified type available for reservation.
6. The system creates a new reservation with the given details.
7. The system displays the reservation confirmation number and the check-in instructions.
8. The use case terminates.

Use Case Specification: Reserve Room

Alternate Flows

A1. Duplicate Submission

In step 5 of the basic flow there is an identical reservation in the system (same name, email and start and end dates), the system displays the existing reservation and asks the customer if he wants to proceed with the new reservation.

- 1.If the customer wants to continue, the system proceeds with the reservation, and the use case resumes.
- 2.If the customer indicates that the new reservation is a duplicate, the use case terminates.

A2. ...

Subflows

S1. Check Room Cost

- 1.The customer selects his desired room type and indicates his period of stay.
- 2.The system computes the cost for the specified period.

Use Case Specification: Reserve Room

Preconditions

The customer has logged in to the system.

Postconditions

Upon successful reservation, a new reservation record is created, and the number of rooms available for the specified dates is decreased. If reservation is unsuccessful, there is no change in the database.

Special Requirements

The system must handle five concurrent reservations. Each reservation should not take more than 20 seconds.

Analysis

- ▶ Requirements analysis
- ▶ An investigation of a domain that results in models describing its static and dynamic characteristics.
- ▶ Domain model:
 - ▶ concepts (conceptual classes),
 - ▶ attributes,
 - ▶ associations.

Design

- ▶ Object-oriented design
- ▶ It emphasizes a conceptual solution that fulfills the requirements, rather than its implementation.
- ▶ Interaction diagrams:
 - ▶ Sequence diagrams
 - ▶ Communication diagrams
- ▶ Design class diagram

UML CASE Tools

- ▶ MagicDraw
- ▶ Visual Paradigm
- ▶ StarUML
- ▶ ArgoUML
- ▶ ...

Case study

- ▶ Develop a simple chat program.
- ▶ In order to use the program one must authenticate.
- ▶ After authentication, the program shows all the online friends of the user.
- ▶ After authentication, the program allows the user to send messages to one of his/her online friends.
- ▶ When a user signs-in / signs-out, all his/her online friends are notified.