

Software Engineering I (02161)

Week 1: Requirements Analysis, Use Cases, and Glossary

Hubert Baumeister

Informatics and Mathematical Modelling
Technical University of Denmark

Spring 2010

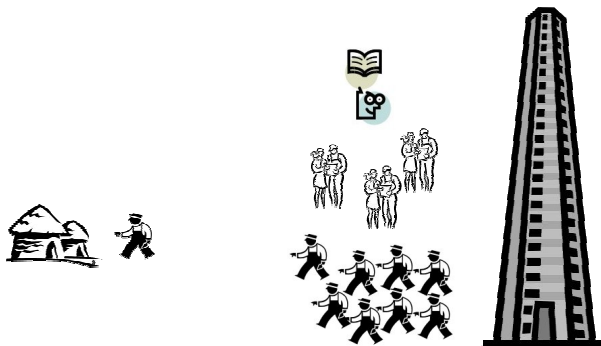


Software Development Problem: Building large software

If you can build a hut, can you also build a skyscraper?



Software Development Problem: Building large software



- You learn **techniques** for building **skyscrapers**
- But the **projects** you are doing are still only **little huts**

Activities in Software Development

- **Understand** and **document** what kind of the software the **customer** wants
 - **Requirements Analysis**
- Determine **how** the software is to be built
 - **Design**
- **Build** the software
 - **Implementation**
- **Validate** that the software solves the customers problem
 - **Testing**
 - **Verification**
 - **Evaluation:** e.g. User friendliness

Requirements Analysis

Requirements Analysis

Understand and **document** the kind of software the **customer** wants

- Techniques for understanding and documentation
 - Use Cases
 - Detailed Use Cases
 - Use Case Diagram
 - Glossary

Travel Agency Example

The travel agency TravelGood comes to you as software developers with the following proposal for a software project:

Problem Description

TravelGood wants to offer a trip-planning and booking application to its customers. The application should allow the customer to plan trips consisting of flights and hotels. First the customer should be able to assemble the trip, before he then books all the flights and hotels in one step. The user should be able to plan several trips. Furthermore it should be possible to cancel already booked trips.

- What do you do?
- What are the requirements?

Types of Requirements

Functional Requirements

Describe the users expectation **which functionalities** the system should have: E.g.

- the user should be able to plan and book a trip

Non-functional Requirements

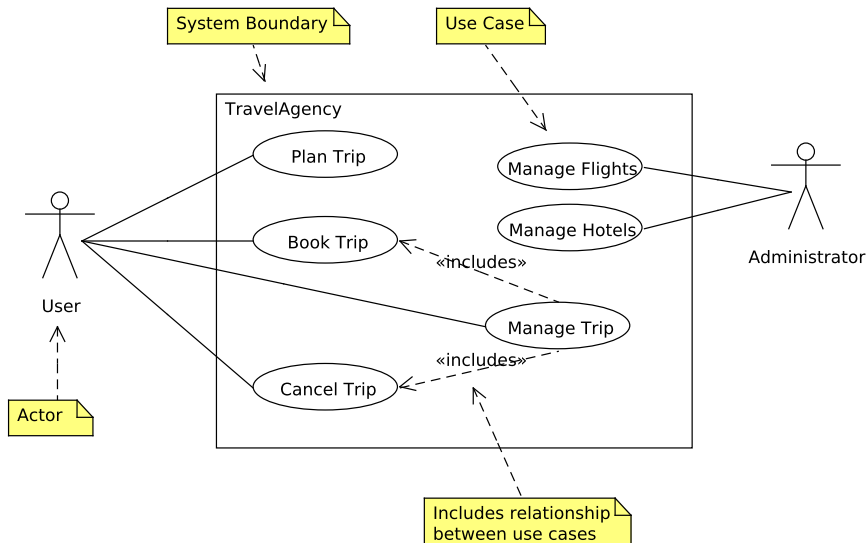
Everything which the user requires from the system, but which is not functionality: E.g.

- Where should the software run (e.g. operating system, software environment, ...)
- What kind of UI the user prefers (e.g. stand alone application, Web application, command line interface, graphical interface, ...)
- Travel Agency Example of non-functional requirements
 - System should be a Web application accessible from all operating systems and most of the Web browsers
 - It must be possible to deploy the Web application in a standard Java application servers like GlassFish or Tomcat
 - The system should be easy to handle (it has to pass a usability test)

Requirements: Issues

- Problem descriptions can be very vague
 - it is important to **discuss** with the **customer** what his/her requirements are
 - Process called: **Requirements elicitation**
- Requirements can change
 - e.g. after the **customer** has seen a first version of the software
 - the **business situation** has changed (cf. finance crises)
- Requirements need to be **documented**
 - **Customer** and **software developer agree** on what the system should be doing)
 - Several ways: **user stories** and **acceptance tests** or
 - **use cases** (**use case diagram**, **detailed use case**)

Travel Agency functional requirements: Overview use case diagram



Use Case

Use Case (Wikipedia)

- A use case in software engineering and systems engineering is a description of a system's behavior as it **responds** to a **request** that originates from **outside** of that system.
- In other words, a use case describes **who** can do **what** with the system in question.

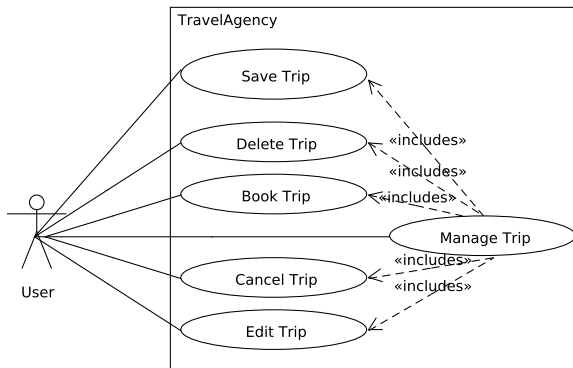
Use Case Diagram

A use case diagram provides an overview over the use cases of a **system** and **who** is using the functionality.

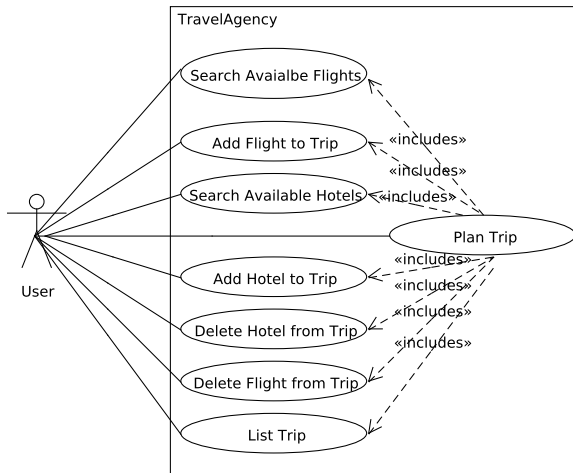
Detailed Use Case description

A detailed use case description describes the interaction between the user and the system as a set of **scenarios**

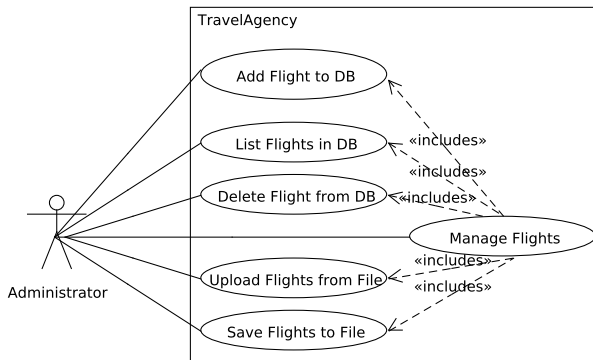
Travel Agency functional requirements: Detailed use case diagram **manage trip**



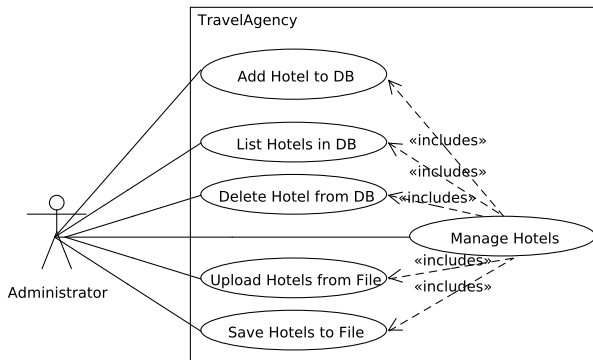
Travel Agency functional requirements: Detailed use case diagram **plan trip**



Travel Agency functional requirements: Detailed use case diagram **manage flights**



Travel Agency functional requirements: Detailed use case diagram **manage hotels**



Detailed use cases: Template

Template to be used for detailed use case descriptions

name: The name of the use case

description: A short description of the use case

actor: One or more actors who interact with the system

precondition: Possible assumptions on the system state to enable the use case

main scenario: A description of the main interaction between user and system

→ Note: should **only** explain what the system does from the **user's** perspective

alternative scenarios:

- note: Used for everything that does not fit in the above categories
- Note: One can find many different types of templates in the literature
 - However, all have in common to state the **main scenario** and the **alternative scenarios**
 - "A use case is a set of scenarios tied together by a common user goal" (From Martin Fowler, UML Distilled)



Travel Agency: detailed use case **list available flights**

name: list available flights

description: the user checks for available flights

actor: user

main scenario:

1. The user provides information about the city to travel to and the arrival and departure dates
2. The system provides a list of available flights with prices and booking number

alternative scenario:

- 1a. The input data is not correct (see below)
 2. The system notifies the user of that fact and terminates and starts the use case from the beginning
- 2a. There are no flights matching the users data
 3. The use case starts from the beginning

note: The input data is correct, if the city exists (e.g. is correctly spelled), the arrival date and the departure date are both dates, the arrival date is before the departure date, arrival date is 2 days in the future, and the departure date is not more than one year in the future



Travel Agency: detailed use case **cancel trip**

name: cancel trip

description: cancels a trip that was booked

actor: user

precondition:

- the trip must have been booked
- the first date for a hotel or flight booking must be one day in the future

main scenario:

1. user selects trip for cancellation
2. the system shows how much it will cost to cancel the trip
3. selected trip will be cancelled after a confirmation

Travel Agency: detailed use case **plan trip**

name: plan trip

description: The user plans a trip consisting of hotels and flights

actor: user

main scenario:

repeat any of the following operations in any order until finished

1. list available flights (use case)
2. add flight to trip (use case)
3. list available hotels (use case)
4. add hotel to trip (use case)
5. list trip (use case)
6. delete hotel from trip (use case)
7. delete flight from trip (use case)

Note: the trip being planned is referred to as the current trip

Travel Agency: detailed use case **save trip**

name: save trip

description: provides the current trip with a name and saves it for later retrieval

actor: user

precondition: the current trip is not empty

main scenario:

1. user provides a name for the trip

alternative scenarios:

- 1: the name is not valid
 - 2: notify the user of the fact and end the use case
- 1: a trip with the name already exists
 - 2: ask the user if the trip should overwrite the stored trip
 - 3a: If yes, overwrite the stored trip
 - 3b: If no, end the use case

Glossary

- Purpose: capture the **customer's knowledge** of the domain so that the **system builders** have the **same knowledge**
 - Helps customer and system builders to speak the **same language**
- Necessary to define the **terminology** used
- **Glossary**

glossary (plural glossaries)

"1. (lexicography) A list of **terms** in a particular **domain of knowledge** with the **definitions** for those terms." (Wikitionary)

- List of terms with explanations
- Terms can be nouns (e.g. those mentioned in a problem description) but also verbs or adjectives e.t.c.

Example

Part of a glossary for the travel agency

User: The person who is using the travel agency

Trip: A trip is a collection of hotel and flight informations. A trip can be booked and, if booked, canceled.

Booking a trip: A trip is booked by making a hotel reservation for the hotels on the trip and a flight booking for the flights of the trip

Flight booking: The flight reservation is booked with the flight agency and is payed.

Reserving a hotel: A hotel is reserved if the hotel informed that a guest will be arriving for a certain amount of time. It is possible that the hotel reservation requires a credit card guarantee.

...

● Warning

- Capture only knowledge relevant for the application
- Don't try to capture all possible knowledge

Summary

- Requirements analysis is about finding out **what** the software should be able to do
- Two types of requirements **functional** and **non-functional** requirements
- Use cases: What are the **functions** the user can perform with the software?
 - Detailed use cases: **Detailed** (textual) description of what the software should do:
 - actors, main scenarios, alternative scenarios (including error handling)
 - Use case diagram: **Graphical overview** over the functionality of the software
- Glossary: Document **domain knowledge** and define a **common language** between customer and software developer