



Use Cases

Contents



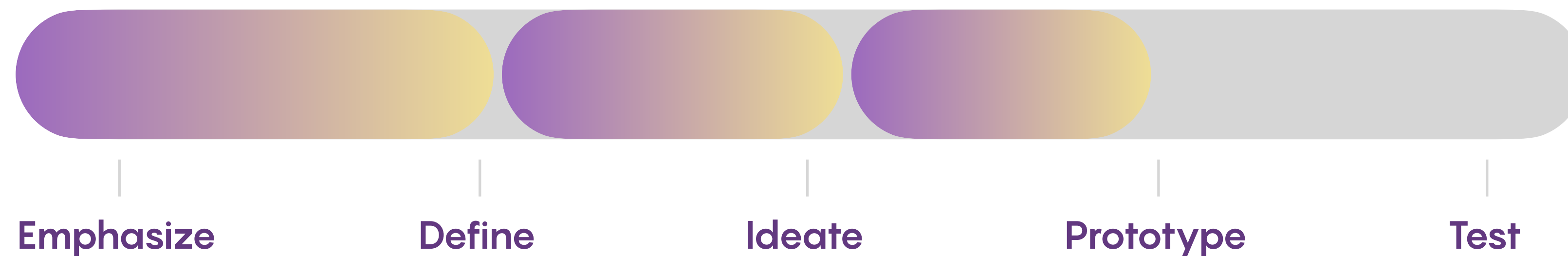
- I. Introduction
- II. Instructions
- III. Strengths & Weaknesses
- IV. Contact

Introduction

A use case is a methodology used in system analysis to identify, clarify and organize system requirements.

- It outlines, from a user's point of view, a system's behavior as it responds to a request
- It is a form of Prototyping.
- It shows all the possible ways to use a system for achieving a specific goal.
- It used in system analysis, software development, user experience design etc.

Design Phases





Introduction – History

- In 1987, Ivar Jacobson presents the first article on use cases at the OOPSLA'87 conference
- In 1992, he co-author a book 'Object-Oriented Software Engineering - A Use Case Driven Approach' where he shows how we can use use case in software engineering
- Later he also publishes book about use cases applied to business models and business process reengineering.

Introduction – Types of Use Cases

- Textual Use Case
 - Shows the use case or flow of the process in textual way (table, flow charts etc)
 - Not used so frequently.
- Use case diagram
 - Shows the use case or flow of the process in diagram.
 - Most commonly used in the industry.

Introduction – Elements of Use Cases

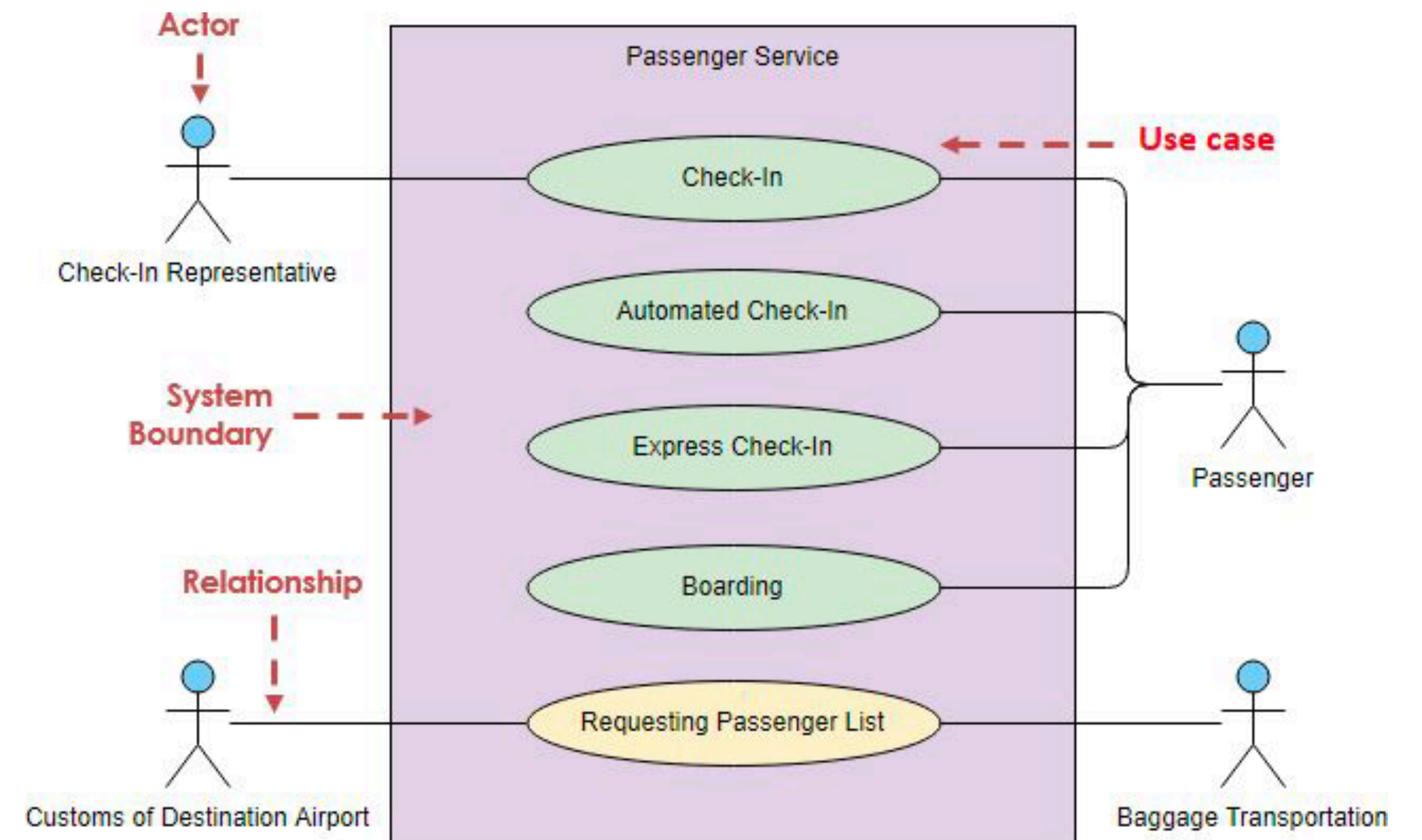
- The Actor
 - The user who will use the system by completing the process.
 - This can be a single person or a group of people interacting with the process.
- The Goal
 - The final successful outcome that completes the process
- The System Process
 - The process that needs to be done to achieve the goal.

Introduction – Components of a Use Case Diagram

- The Actor: An actor portrays any entity (or entities) that perform certain roles in a given system.
- Use case: A use case in a use case diagram is a visual representation of a distinct business functionality in a system.
- System boundary: A system boundary defines the scope of what a system will be.
- Relationships in Use Cases
 - Include: When a use case is depicted as using the functionality of another use case in a diagram, this relationship between the use cases is named as an include relationship.
 - Extend: In an extend relationship between two use cases, the child use case adds to the existing functionality and characteristics of the parent use case.
 - Generalizations: A generalization relationship is also a parent-child relationship between use cases.

Introduction – Components of a Use Case Diagram

- The Actor
- Use case
- System boundary
- Relationships between Use Cases
 - Include
 - Extend
 - Generalizations



Instructions

Process



1. Identifying all system users and creating a profile for each one. This includes every role played by a user who interacts with the system.
2. Selecting one user and defining their goal -- or what the user hopes to accomplish by interacting with the system. Each of these goals becomes a use case.
3. Describing the action taken for each use case through the system to reach that goal.
4. Considering every alternate action of events and extending use cases — or the different actions that can be taken to reach the goal.
5. Identifying commonalities in journeys to create common action use cases and write descriptions of each.
6. Repeating steps two through five for all other system users.

Instructions

Tools

1. Pen
2. Paper
3. Pencil
4. Use Case Template
5. Various kind shape scales





Strengths:

Provides a comprehensive summary of the whole system in a single illustration. Starts from a simple viewpoint in the system. Helps to envision the outcome and develop the system more proactively.

Weaknesses:

Doesn't capture the non-functional requirements. Does not address usefulness and usability. Can be complex to build when system in general is complex.



Contact

usability-siegen.de

fablab-siegen.de