

Hotel-project

Week 4: Design 3

This weeks topics are: *Component diagrams*, *Class diagrams* and *Sequence diagrams*.

Assignment

This is the last assignment where you create new models. From next week on, you will implement your models using Java and will only update the existing models. Before that, you will do three more types of diagrams: component diagrams, class diagrams, and sequence diagrams. Besides these, you shall also review the work of another group. The instructions for the halftime review can be found in another PDF in the project part of the course homepage.

Class Diagram

Draw a class diagram that supports your complete use cases from last week's assignment. Use the domain model as a starting point for the class diagram and add classes with any operations and attributes you need in order to realize your use cases. Remember that the domain model reflects a conceptual perspective of the domain, while class diagrams represent a software development perspective. Draw the class diagram using Papyrus and hand it in as an image export from Papyrus.

Iterate between the class diagram, the component diagram and the sequence diagrams.

Component Diagram

Draw a component diagram of your system and its surroundings.

Your booking system shall be placed in at least one component. You are free to divide it into further components if you think this is reasonable! Use your actors (assignment 2) and your system operations (assignment 3) in order to define interfaces for you component(s). Visualise your actors in the component diagram by adding one component per actor and connecting them to your system's interfaces.

Sequence diagrams

Construct a sequence diagram for at least 3 of the complete use case that you wrote last week. Use the system operations which you have defined in assignment 3. The sequence diagrams shall depict a interaction between the use case actor and your system component(s). This means that they have to follow the component interface!

The implementation of the search for available rooms does not need to be sophisticated. For passing the course it is enough to return a room with the right amount of beds or deny the booking if there are no available rooms. An implementation returning a sufficient amount of rooms to accomodate the number of people in the booking is of course more satisfying.

Checklist

- A Component Diagram, modelled in Papyrus, containing your system's component(s), and components visualising your actors.
- A Class Diagram, modelled in Papyrus.
- A Sequence Diagram, modelled in Papyrus, for at least 3 of the your complete Use Case from assignment 3.
- Halftime review, see separate document.

Literature

- Craig Larman, "*Applying UML and Patterns*", chapters 15 and 16.