

Softwaretechnik 2 (ST2)

Informatik Bachelor, SoSe 2026

Prof. Dr.-Ing. Stefan Bente

Workshop M3:

Änderungen in den Use Case Interfaces

Technology
Arts Sciences
TH Köln

```
public interface OrderUseCases {
    int placeOrder( UUID guestId, String mealName, LocalDateTime currentTime );
    boolean isOrderedButNotYetReady( int orderNumber );
    boolean isOrderBeingPrepared( int orderNumber );
    boolean isOrderReady( int orderNumber );
    boolean hasOrderBeenPickedUp( int orderNumber );
    void prepareOrder( int orderNumber );
    void pickupOrder( int orderNumber );
    List<Integer> getOrdersForTable( int tableNumber );
    int getTableNumberForOrder( int orderNumber );
    UUID getGuestIdForOrder( int orderNumber );
    String getMealNameForOrder( int orderNumber );
}
```

M2

```
public interface OrderUseCases {
    UUID placeOrder( UUID guestIdAsUUID, String mealName, LocalDateTime currentTime );
    int getOrderNumber( UUID orderIdAsUUID );
    boolean isOrderedButNotYetReady( UUID orderIdAsUUID );
    boolean isOrderBeingPrepared( UUID orderIdAsUUID );
    boolean isOrderReady( UUID orderIdAsUUID );
    boolean hasOrderBeenPickedUp( UUID orderIdAsUUID );
    void prepareOrder( UUID orderIdAsUUID );
    void pickupOrder( UUID orderIdAsUUID );
    List<UUID> getOrdersForTable( int tableNumber );
    int getTableNumberForOrder( UUID orderIdAsUUID );
    UUID getGuestIdForOrder( UUID orderIdAsUUID );
    String getMealNameForOrder( UUID orderIdAsUUID );
}
```

M3

Problem bei OrderUseCases

```
/**  
 * Resolves an order number to a single order. Because numbers are released on  
 * pickup and re-assigned to fresh orders, the table can hold several rows for  
 * the same number — at most one of them not yet picked up. Active orders are  
 * preferred (so the lifecycle/status methods act on the current holder), and a  
 * picked-up row is returned only when it is the sole remaining one (e.g. a  
 * just-completed order whose number has not been recycled).  
 */
```

```
@Query( "SELECT o FROM Order o WHERE o.orderNumber = :orderNumber "  
 + "ORDER BY CASE WHEN o.status = thkoeln.archilab.restaurant.solution.order.domain.OrderStatus.PICKED_UP "  
 + "THEN 1 ELSE 0 END, o.orderedAt DESC LIMIT 1" )
```

```
Optional<Order> findByOrderNumber( @Param( "orderNumber" ) OrderNumber orderNumber );
```



<https://www.archi-lab.io/infopages/architecture/business-and-technical-id.html>

ReservationUseCases

```
public interface ReservationUseCases {  
    int createReservation( String guestName, LocalDateTime reservationTime, int partySize, int tableNumber );  
    void cancelReservation( int reservationNumber );  
    boolean isReservationActive( int reservationNumber );  
    boolean isReservationConfirmed( int reservationNumber );  
    boolean isReservationCancelled( int reservationNumber );  
    boolean isReservationNoShow( int reservationNumber );  
    List<Integer> getReservationsForDate( LocalDate date );  
    List<UUID> getGuestsForReservation( int reservationNumber );  
}
```

M2

```
public interface ReservationUseCases {  
    UUID createReservation( String guestName, LocalDateTime reservationTime, int partySize, int tableNumber );  
    int getReservationNumber( UUID reservationIdAsUUID );  
    void cancelReservation( UUID reservationIdAsUUID );  
    boolean isReservationActive( UUID reservationIdAsUUID );  
    boolean isReservationConfirmed( UUID reservationIdAsUUID );  
    boolean isReservationCancelled( UUID reservationIdAsUUID );  
    boolean isReservationNoShow( UUID reservationIdAsUUID );  
    List<UUID> getReservationsForDate( LocalDate date );  
    List<UUID> getGuestsForReservation( UUID reservationIdAsUUID );  
}
```

M3

ReservationCheckInUseCases

```
public interface ReservationCheckInUseCases {  
    UUID reservationCheckIn( int tableNumber, int reservationNumber, LocalDateTime currentTime );  
}
```

M2

```
public interface ReservationCheckInUseCases {  
    UUID reservationCheckIn( int tableNumber, UUID reservationId, LocalDateTime currentTime );  
}
```

M3