Aufgabe 12.1

Adapting the sequence diagram for the lend movie functionality will involve the following notable changes:

i) Lend Movie

1) *movieDetailEditActivity* activity binds the *lentUntilDate* attribute and to the view via *buildLentDateAttr()*. A new movie begins with a *null* attribute, as well as a movie which is currently free to be lent (alternatively, one can work with the flag *isLent* and the *lendingHistory* attributes, as proposed in sheet 10).

2) The *lentUntilDate* attribute can be changed via user interaction, therefore, the event for triggering the *Lend Movie* function are:

- The user selects a “Lend Movie” button
- The user selects a date until the movie will be lent and confirms

Note, that the above assumes that the movie is not currently lent.

3) We would probably need a *lentUntilDate* DateAttribute class (similarly to the watchDate implementation)

4) We can reuse the *DateSelectionView* which will create a dialog for lending the movie with minDate until tomorrow and maxDate as a pre-defined maxLendingDate limit. The view appears after clicking the “Lend Movie” button.

5) For the dialog implementation, we can reuse the *DateSelectionDialog* which, again, will onlz interact with the *DateSelectionView.*

6) In case the user has confirmed lending the movie, the same changes will occur as in the case of setting a watch date (since the *MovieDetailActivity*) will similarly register changes. The DateAttribute *lentUntilDate* will also be a part of it.

The following addition is applied to *MovieTransformations*:

```
public static ReversibleTransformation<Movie> setLendDate(Date lendDate) {
    return reversibleTransformation(
            Movie::getLendDate,
            Movie::setLendDate,
            lendDate
    );
}
```

and the following to *MovieFromJsonObject*:

```
registerConversion("lendDate", MovieFromJsonObject::convertStringToDate);
```

ii) Receive Returned Movie

1) The events triggering this system function will be user interaction in the *movieDetailEditActivity.* Assuming the *lentUntilDate* is not null, the user will have an option to click a button “Return Movie”.

2) Upon interaction, the *lentUntilDate* DateAttribute ist changed to null (we can also reuse the setLentUntilDate with a null argument).

3) The *movieDetailEditActivity* is informed about the change and the view updated accordingly. There is no interaction with the *DateSelectionDialog* and *DateSelectionView* classes.

iii) Change Due Date

1) Changing the *lentUntilDate* is possible only of the movie has a current lentUntilDate on the first place. The system function is therefore triggered when the user selects a “Change Due Date” button which is only visible while the movie is lent.

2) Upon selection, the same steps are taken as for lending the movie:

3) Again, we can reuse the *DateSelectionView* which will create a dialog for lending the movie with minDate until tomorrow and maxDate as a pre-defined maxLendingDate limit.

4) Again, we can reuse the *DateSelectionDialog* which enables the user to pick the new lent date.

5) Two possible branches go back to the *movieDetailEditActivity.* Either the user selects a new date and confirms, which triggers the *setLentUntilDate* and applies the change in the *Movie* object, OR the user cancels the selection and is back to *movieDetailEditActivity.*