

Service.java

```
1//Gruppe 1: Madeleine, Peter og Torben
2
3package application.service;
4
5import java.time.LocalDate;
6import java.util.ArrayList;
7
8import application.model.Beboelse;
9import application.model.Booking;
10import application.model.Deltager;
11import application.model.Konference;
12import application.model.Ledsager;
13import application.model.Organisation;
14import application.model.Tilmelding;
15import application.model.Udflugt;
16import storage.Storage;
17
18public class Service {
19
20    public static Deltager createDeltager(String navn, String adresse, String
    by, String land, String tlfnr) {
21        Deltager deltager = new Deltager(navn, adresse, by, land, tlfnr);
22        Storage.addDeltager(deltager);
23        return deltager;
24    }
25
26    public static ArrayList<Deltager> getDeltagere() {
27        return Storage.getDeltagers();
28    }
29
30    public static Deltager createDeltager(String navn, String firmanavn,
    String adresse, String by, String land,
31        String tlfnr, String firmatlfnr) {
32        Deltager deltager = new Deltager(navn, firmanavn, adresse, by, land,
    tlfnr, firmatlfnr);
33        Storage.addDeltager(deltager);
34        return deltager;
35    }
36
37    public static void removeDeltager(Deltager deltager) {
38        for (Tilmelding tilmelding : deltager.getTilmeldinger()) {
39            tilmelding.getKonference().removeTilmelding(tilmelding);
40            removeLedsager(tilmelding);
41        }
42        Storage.removeDeltager(deltager);
43    }
44}
```

Service.java

```
45     public static Deltager getDeltager(String tlfnr) {
46         for (Deltager deltager : Storage.getDeltagers()) {
47             if (tlfnr.equals(deltager.getTlfnr())) {
48                 return deltager;
49             }
50         }
51         return null;
52     }
53
54     public static ArrayList<String> getDeltagere(Konference konference) {
55         ArrayList<String> deltagere = new ArrayList<String>();
56
57         for (Tilmelding tilmelding : konference.getTilmeldinger()) {
58             String deltagerStreng = "";
59             if (tilmelding.getLedsager() != null) {
60                 deltagerStreng += tilmelding.getDeltager().getNavn() + " & " +
61                 tilmelding.getLedsager().getNavn();
62             } else {
63                 deltagerStreng += tilmelding.getDeltager().getNavn();
64             }
65
66             if(tilmelding.getDeltager().getFirmanavn() != null) {
67                 deltagerStreng += ", Firma: " +
68                 tilmelding.getDeltager().getFirmanavn() + ".";
69             }
70
71             deltagere.add(deltagerStreng);
72         }
73
74         return deltagere;
75     }
76
77     public static void removeLedsager(Tilmelding tilmelding) {
78         Ledsager ledsager = tilmelding.getLedsager();
79         for (Udflugt udflugt : ledsager.getUdflugter()) {
80             udflugt.removeLedsager(ledsager);
81         }
82     }
83
84     //
85
86     public static Tilmelding createTilmelding(Deltager deltager, Konference
konference, LocalDate ankomstdato,
LocalDate afrejsedato, boolean foredragsholder) {
Tilmelding tilmelding = deltager.createTilmelding(konference,
```

Service.java

```
    ankomstdato, afrejsedato, foredragsholder);
87    deltager.addTilmelding(tilmelding);
88    konference.addTilmelding(tilmelding);
89    return tilmelding;
90 }
91
92 //
-----
-----
93
94    public static Booking createBooking(Tilmelding tilmelding,
    ArrayList<application.model.Service> services,
95        Beboelse beboelse) {
96        Booking booking = tilmelding.createBooking(services, beboelse);
97        return booking;
98    }
99
100    public static Booking createBooking(Tilmelding tilmelding, Beboelse
    beboelse) {
101        Booking booking = tilmelding.createBooking(beboelse);
102        return booking;
103    }
104    //
-----
-----
105
106    public static Beboelse createBeboelse(String name, double
    prisEnkeltvaerelse, double prisDobbeltvaerelse) {
107        Beboelse beboelse = new Beboelse(name, prisEnkeltvaerelse,
    prisDobbeltvaerelse);
108        Storage.addBeboelse(beboelse);
109        return beboelse;
110    }
111
112    //
-----
-----
113
114    public static Ledsager createLedsager(Tilmelding tilmelding, String navn)
    {
115        Ledsager ledsager = tilmelding.createLedsager(navn);
116        return ledsager;
117    }
118    //
-----
-----
119
```

Service.java

```
120     public static Organisation createOrganisation(String navn) {
121         Organisation organisation = new Organisation(navn);
122         Storage.addOrganisation(organisation);
123         return organisation;
124     }
125
126     public static ArrayList<Organisation> getOrganisations() {
127         return new ArrayList<>(Storage.getOrganisations());
128     }
129     //
-----
130
131     public static Konference createKonference(Organisation organisation,
132         LocalDate startDato, LocalDate slutDato,
133         String lokation, String navn, String tema, double pris) {
134         Konference konference = organisation.createKonference(startDato,
135             slutDato, lokation, navn, tema, pris);
136         Storage.addKonference(konference);
137         return konference;
138     }
139
140     public static ArrayList<Konference> getSpecifikkeKonferencer(Organisation
141         organisation) {
142         return Storage.getSpecifikkeKonferencer(organisation);
143     }
144
145     public static ArrayList<Konference> getKonferencer() {
146         return Storage.getKonferencer();
147     }
148
149     public static LocalDate getKonferenceStartdato(Konference konference) {
150         return konference.getStartDato();
151     }
152
153     public static LocalDate getKonferenceSlutdato(Konference konference) {
154         return konference.getSlutDato();
155     }
156
157     public static Beboelse addBeboelseToKonference(Konference konference,
158         String navn, double prisEnkelvaerelse,
159         double prisDobbeltvaerelse) {
160         Beboelse beboelse = new Beboelse(navn, prisEnkelvaerelse,
161             prisDobbeltvaerelse);
162         konference.addBeboelse(beboelse);
163         Storage.addBeboelse(beboelse);
164         return beboelse;
165     }
```

Service.java

```
160     }
161     //
-----
162
163     public static Udflugt createUdflugt(Konference konference, String navn,
    LocalDate dato, double pris,
164         boolean frokost) {
165         Udflugt udflugt = konference.createUdflugt(navn, dato, pris, frokost);
166         return udflugt;
167     }
168
169     public static ArrayList<Udflugt> getUdflugter(Konference konference) {
170         return new ArrayList<>(konference.getUdflugter());
171     }
172
173     public static ArrayList<Ledsager> getUdflugtTilmeldte(Udflugt udflugt){
174         return new ArrayList<>(udflugt.getLedsagere());
175     }
176 }
177
178 //
-----
179
180     public static double beregnSamletPris(Tilmelding tilmelding) {
181         return tilmelding.beregnSamletPris();
182     }
183
184     public static String getKonferenceNavn(Tilmelding tilmelding) {
185         return tilmelding.getKonference().getNavn();
186     }
187
188     public static void removeTilmelding(Tilmelding tilmelding) {
189         // TODO Auto-generated method stub
190         tilmelding.getDeltager().fjernTilmelding(tilmelding);
191     }
192
193     public static ArrayList<Beboelse> getBeboelser(Konference konference) {
194         return new ArrayList<>(konference.getBeboelser());
195     }
196
197     public static ArrayList<application.model.Service> getServices(Beboelse
    beboelse) {
198         return new ArrayList<>(beboelse.getServices());
199     }
200
```

Service.java

```
201     public static application.model.Service addService(Beboelse beboelse,
202     String navn, String beskrivelse,
203     double pris) {
204         application.model.Service service = beboelse.createService(navn,
205         beskrivelse, pris);
206         return service;
207     }
208     public static Boolean hasLedsager(Tilmelding tilmelding) {
209         return tilmelding.getLedsager() != null;
210     }
211     public static Boolean hasOvernatning(Tilmelding tilmelding) {
212         return tilmelding.getBooking() != null;
213     }
214 }
215 }
216 }
```