

Team Mitglieder

Kai Friese

und Jasmin Knott





Produkt - Vision

Vision Statement

Zielgruppe:

Parkhausbetreiber

Bedarf:

Software, die ein Parkhaus verwaltet

Produkt:

- -leicht zu bedienen -liefert notwendige Informationen über alle Prozesse
- Kunden entsprechende Parkplätze zuweisen

Mehrwert:

- -testen des Parkhaussystems
- -Geld sparen durch einfache, automatische Parkhausverwaltung
- -bessere Orientierung für Kunden

Userstorys

User 01:

Als Kunde benötige ich Informationen über verfügbare Parkplätze, um für mein Auto den perfekten Parkplatz zu finden

User 04

Als Betreiber des Parkhauses will ich mit einem Knopfdruck alle Statistiken anzeigen lassen, um genaue Informationen über mein Parkhaus zu erhalten

User 07:

Als Kunde möchte ich Informationen über die Öffnungszeiten des Parkhauses haben, damit ich weiß wann ich rein/ raus fahren kann

User 02:

Als Firmenbesitzer hätte ich gerne feste Parkplätze, um meinen Angestellten und Kunden Platz zu bieten

User 05:

Als Abonnent des Parkhauses ziehe ich am Automaten ein Monatsticket, um einen ganzen Monat mit nur einem Parkschein in das Parkhaus fahren zu können

User 08:

Als Kunde will ich auf dem Parkschein eine Parkplatznummer haben, um nicht an der falschen Stelle zu parken

User 03:

Als Pförtner stelle ich Parkscheine aus, um die Parkdauer und den Preis ausrechnen zu können

User 06:

Als Kunde möchte ich eine Anzeige haben, ob es noch freie Parkplätze gibt, um nicht unnötig ins Parkhaus zu fahren

User 09:

Als Kunde möchte ich wenn es keine freien Parkplätze meiner Gruppe gibt, eine andere Parkalternative haben, um nicht außerhalb des Parkhauses parken zu müsssen

User 10:

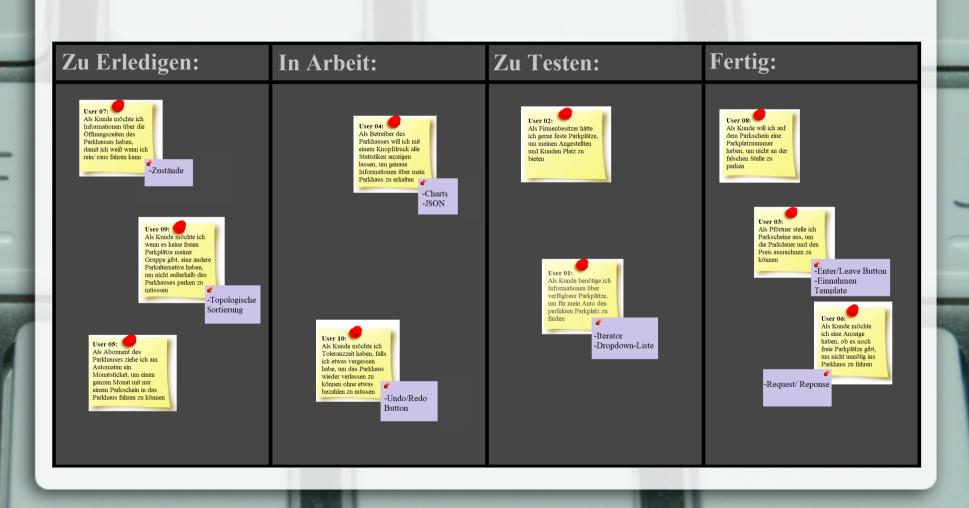
Als Kunde möchte ich Toleranzzeit haben, falls ich etwas vergessen habe, um das Parkhaus wieder verlassen zu können ohne etwas bezahlen zu müssen

Priorisierung der Userstorys und MVP

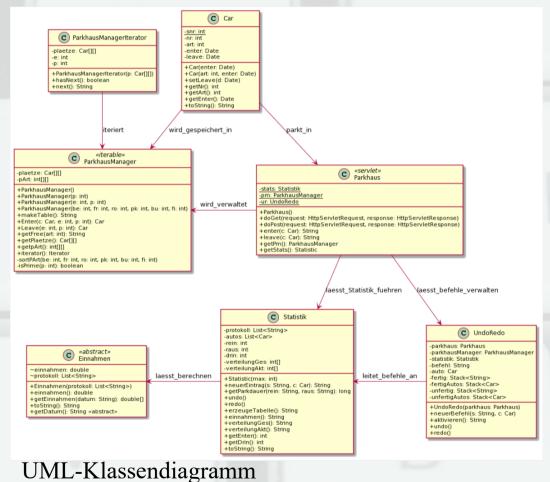
	Value Points	Story Points	Quotient
User 08	233	1	233
User 03	144	5	28,8
User 06	55	2	27,5
User 01	21	1	21
User 02	34	3	11,3
User 04	89	55	1,62
User 10	13	34	0,382
User 07	3	8	0,375
User 09	2 2	21	0,096
User 05	1	13	0,077

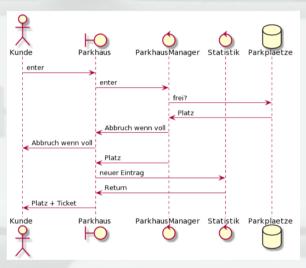
MVP

Scrum Board



UML - Diagramme





UML-Sequenzdiagramm

Parkhausdesign

Parkhaus

Statistik

Log

	Platz: 0	Platz: 1	Platz: 2	Platz: 3	Platz: 4	Platz: 5	Platz: 6	Platz: 7	Platz: 8	Platz: 9
Etage: 0	Be11	Be12	Be15	Be42	Be25	Be53	Be59	Be61	Fr8	Fr23
Etage: 1	Fr30	Fr32	Fr37	Fr50	Fr56	null	Ro5	Ro18	Ro43	Ro49
Etage: 2	Ro51	Ro55	null	null	PkW20	PkW1	PkW4	PkW6	PkW9	PkW36
Etage: 3	PkW44	PkW21	PkW22	PkW26	PkW46	PkW47	PkW52	null	null	null
Etage: 4	null									
Etage: 5	null	null	null	null	null	null	Bus41	Bus7	Bus45	Bus13
Etage: 6	Bus48	Bus54	null							
Etage: 7	null	null	Fi2	Fi14	Fi57	Fi58	Fi60	null	null	null

Statistik: Einnahme



Gesamte Verteilung:







Prei
PKW
PKW
Frauen
Dis
Roller
Firmen

Event	Ticket	Timestamp	Duration	Price	Slot	Parking Type	Car
enter	0	2020-10-08 14:34:12.052	-	-	1:1	3	PkW0
enter	1	2020-10-08 14:34:12.802	-	-	2:4	4	Bus1
enter	2	2020-10-08 14:34:14.086	-	-	1:2	3	PkW2
undo	2	2020-10-08 14:34:14.956	-	-	-	-	PkW2
leave	0	2020-10-08 14:34:15.81	3	0.015	1:1	3	PkW0
redo	2	2020-10-08 14:34:16.922	-	-	-	-	PkW2
enter	3	2020-10-08 14:34:17.807	-	-	1:2	3	PkW3
enter	4	2020-10-08 14:34:18.253	-	-	1:3	3	PkW4
undo	4	2020-10-08 14:34:18.792	-	-	-	-	PkW4
leave	2	2020-10-08 14:34:19.904	5	0.025	1:1	3	PkW2
enter	5	2020-10-08 14:34:28.739	-	-	0:0	0	Be5
enter	6	2020-10-08 14:34:30.308	-	-	0:1	0	Be6
enter: error	-	2020-10-08 14:34:32.078	-	-	-	-	-

Eigenleistung

Kai Friese

- Parkhaus-Servlet
- ParkhausManager
- ParkhausManagerIterator
- tabs.css
- Car
- StatistikTest
- EinnahmenTest

Jasmin Knott

- Statistik
- Index.jsp
- UndoRedo
- Einnahmen
- ParkhausTest
- ParkhausManagerTest

Parkhaus-Servlet

```
if(request.getParameter("Raus") != null) {
   if(!request.getParameter("rausEtage") .equals("") || !request.getParameter("rausPlatz").equals("")) {
        Leave(Integer.parseInt(request.getParameter("rausEtage")),
         protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
43
             if(request.getSession().getAttribute("stats") != null) stats = (Statistik) request.getSession().getAttribute("stats");
                                                                                                                                                        if(request.getParameter("Erstellen") != null)
                                                                                                                                                                                         Integer.parseInt(request.getParameter("rausPlatz")));
                  if(request.getParameter("etagen") == null || request.getParameter("plaetze") == null) {
                      pm = new ParkhausManager(1, 1);
                                                                                                                                                                               int in = stats.getDrin();
int ran = (int)(Math.random()*in-1)+1;
                      stats = new Statistik(pm.getPlaetze().length * pm.getPlaetze()[0].length);
                       ur = new UndoRedo(this);
                                                                                                                                                                                Iterator it = pm.iterator();
                      while(it.hasNext()) {
                                                                                                                                                                                    String next = (String) it.next();
String[] plz = next.split(",");
                       ur = new UndoRedo(this);
                                                                                                                                                                                     int e = Integer.parseInt(plz[0]);
                                                                                                                                                                                     int p = Integer.parseInt(plz[1]);
                                                                                                                                                                                    if(ran == 0) {
59
                                                                                                                                                                                         break;
             if(request.getParameter("parkplatz") != null && request.getParameter("Rein") != null) {
                                                                                                                                                                                     if(pm.getPlaetze()[e][p]!=null) {
                  String param = request.getParameter("parkplatz");
                                                                                                                                                                                         ran--;
if(ran == 0) {
                 Date denter = new Date();
if(param.equals("zufall")) {
                                                                                                                                                                                             x = e;
                      Enter(new Car(denter));
                                                                                                                                                                                             y = p;
                                                                                                                                                         112
113
114
115
116
                  else if(param.equals("behindert")) {
                      Enter(new Car(0,denter));
                  else if(param.equals("frau")) {
                                                                                                                                                                                Leave(x,y);
                      Enter(new Car(1,denter));
                                                                                                                                                        118
119
120
                  else if(param.equals("roller")) {
                                                                                                                                                                       if(request.getParameter("Undo") != null) {
                      Enter(new Car(2,denter));
                  else if(param.equals("pkw")) {
                                                                                                                                                         121
                      Enter(new Car(3,denter));
                                                                                                                                                         122
                                                                                                                                                         123
124
125
                                                                                                                                                                       if(request.getParameter("Redo") != null) {
                  else if(param.equals("bus")) {
                                                                                                                                                                            ur.redo();
                      Enter(new Car(4, denter));
                  else if(param.equals("firmen")) {
                                                                                                                                                                       request.setAttribute("tabelle", pm.makeTable());
                      Enter(new Car(5,denter));
                                                                                                                                                         128
                                                                                                                                                                       request.setAttribute("statistikTabelle", stats.erzeugeTabelle());
                                                                                                                                                                       request.setAttribute("verteilungGes", stats.verteilungGes());
request.setAttribute("verteilungAkt", stats.verteilungAkt());
request.setAttribute("einnahmen", stats.einnahmen());
                                                                                                                                                         129
130
                                                                                                                                                                       request.getSession().setAttribute("stats", stats);
                                                                                                                                                                       RequestDispatcher view = request.getRequestDispatcher("/index.jsp");
                                                                                                                                                         134
135
                                                                                                                                                                       view.forward(request, response);
```

Statistik

```
public class Statistik {
                                                                                                                                                                                                                               public String erzeugeTabelle() {
             private List<String> protokoll;
                                                                                                                                                                                                                                    //Freignis, Ticketnummer, timestamp, Dauer, Preis, Parkplatz, Parkplatzart, Auto
String tabelle = "ttable border = 1>" + '\n';
             private List<Car> autos;
             private int rein, raus, drin;
                                                                                                                                                                                                                                     if(protokoll.size() > 0) {
   tabelle += "Event" + '\n';
             private int[] verteilungGes, verteilungAkt;
                                                                                                                                                                                                                                           tabelle += ">th>Ticket' + '\n';
tabelle += ">Ticket' + '\n';
tabelle += ">Ticket' + '\n';
tabelle += ">Duration' + '\n';
            public String neuerEintrag(String s, Car c) {
    //Freignis, likethnummer, timestamp, Bauer, Preis, Parkplatz, Parkplatzart, Auto
    String[] parameter = s.split(","); //Eingabe auticilen
    if("enter".equals(parameter[0])) { //Statistik für rein annassen
    verteilungese[Integer.preseInt(parameter[6])] + = 1;
                                                                                                                                                                                                                103
28
29
30
                                                                                                                                                                                                                                           tabelle += "cth>Price(nt/n)" + \n;
tabelle += "cth>Frice(/th)" + '\n';
tabelle += "cth>Slots/th)" + '\n';
tabelle += "cth>Parking Type" + '\n';
tabelle += "cth>Carc/th)" + '\n';
31
32
33
                          verteilungAkt[Integer.parseInt(parameter[6])] += 1;
                           verteilungAkt[6] -= 1;
                                                                                                                                                                                                                                           for(int z = 0; z < protokoll.size(); z++) { //pro Eintrag
    tabelle += "<tr>)" + '\n':
                                                                                                                                                                                                                109
                          rein++:
                                                                                                                                                                                                                                                  factor = { (f) + \n ;

String[] l = protokoll.get(z).split(","); //Eintrag aufteilen
for(int s = 0; s < 1.length; s++) { //pro Spalte
    tabelle += "<td>" + l[s] + "</tb>" + '\n'; //Inhalt
                                                                                                                                                                                                                110
34
35
36
37
38
39
40
41
42
43
                    if("leave".equals(parameter[0])){ //Statistik für raus anpassen
                          String rein = "":
                           for(int i = 0; i < protokoll.size(); i++) { //passendes Reinprotokoll finden
                                String[] tempParameter = protokoll.get(i).split(',");
if(tempParameter[].equals(parameter[1]) && tempParameter[0].equals("enter")) { //gleiches Ticket
rein = protokoll.get(i);
                                                                                                                                                                                                                                                  tabelle += "" + '\n';
                                                                                                                                                                                                                115
                                                                                                                                                                                                                                      tabelle += "";
                                                                                                                                                                                                                118
                                                                                                                                                                                                                                     return tabelle:
                          if(!rein.equals("")) { //wenn ein passendes Protokoll vorhanden ist
                                                                                                                                                                                                                119
                                irein.equals( )) { //wenn ein passenges rrotokoil worhangen lat.
long d = getParkdauer(rein, s); //Rackdauer berechnen
String format = new SimpleDateFormat("SS:mm:ss").format(d); //Rackdauer umformen
String[] formatParameter = format.split(":");
int zeit = Integer.parseInt(formatParameter[2]) + Integer.parseInt(formatParameter[1])*60
                                                                                                                                                                                                                120
44
45
46
47
48
49
                                                                                                                                                                                                                              public void undo(String s) {
                                                                                                                                                                                                                                    for(int i = protokoll.size()-1; i >= 0; i--) { //passendes Protokoll finden
if(protokoll.get(i).equals(s)) {
    String[] parameter = s.split(",");
                                                                                                                                                                                                                150
                               int let = Integer.parseInt(formatrainmeter[2]) + Integer.

double preis = zeit * 0.005; /P.re.is berechnen

parameter[3] = "" + zeit;

parameter[4] = "" + preis;

s = parameter[0]; //string mit neuen Parametern erstellen

for(int i = 1; i < parameter.length; i++) {
                                                                                                                                                                                                                                                  if(parameter[0].equals("enter")) { //undo yon rein
  verteilungGes[Integer.parseInt(parameter[6])] -= 1;
  verteilungAkt[Integer.parseInt(parameter[6])] -= 1;
                                                                                                                                                                                                                153
                                                                                                                                                                                                                154
                                                                                                                                                                                                                155
                                                                                                                                                                                                               156
157
                                                                                                                                                                                                                                                          verteilungAkt[6] += 1;
                                                                                                                                                                                                                                                         rein--;
                                        s += "," + parameter[i];
54
55
56
57
58
59
                                                                                                                                                                                                                158
                                                                                                                                                                                                                159
                                                                                                                                                                                                                                                   if(parameter[0].equals("leave")) { //undo yon raus
                                 verteilungAkt[Integer.parseInt(parameter[6])] -= 1;
                                                                                                                                                                                                                                                         verteilungAkt[Integer.parseInt(parameter[6])] += 1;
verteilungAkt[6] -= 1;
                                                                                                                                                                                                                160
                                 verteilungAkt[6] += 1;
                                                                                                                                                                                                                161
                                raus++;
                                                                                                                                                                                                                163
60
61
62
                                                                                                                                                                                                                164
                    drin = rein - raus;
                                                                                                                                                                                                                165
                    protokoll.add(s);
                                                                                                                                                                                                               166
167
                    System.out.println(s); //Konsolenausgabe
                                                                                                                                                                                                                              public void redo(String s) {
                   return s:
                                                                                                                                                                                                                                    169
170
                                                                                                                                                                                                               172
173
                                                                                                                                                                                                                175
176
177
                                                                                                                                                                                                                                                          verteilungAkt[6] -= 1;
                                                                                                                                                                                                                                                         rein++;
                                                                                                                                                                                                                                                   if(parameter[0].equals("leave")) { //redo yon raus
                                                                                                                                                                                                                                                         verteilungAkt[Integer.parseInt(parameter[6])] -= 1;
verteilungAkt[6] += 1;
                                                                                                                                                                                                                182
                                                                                                                                                                                                                183
```

Testklassen

```
package tests;
 3⊕ import static org.junit.Assert.assertEquals; [...
   public class ParkhausTest {
13
15⊝
16
       public void reinRausTest() {
           Parkhaus parkhaus = new Parkhaus();
17
18
           Date date0 = new Date();
19
20
           String string0 = "enter,0,"+new Timestamp(date0.getTime())+",-,-,4:3,4,Bus0";
21
           Car auto0 = new Car(4, date0);
22
           assertEquals(parkhaus.Enter(auto0), string0);
23
           Date date1 = new Date();
24
25
           String string1 = "enter: error,-,"+new Timestamp(date1.getTime())+",-,-,-,-";
26
           Car auto1 = new Car(6, date1);
27
           assertEquals(parkhaus.Enter(auto1), string1);
28
29
           //Zeitpunkt, Parkdauer und Preis hängen von dem in der Methode erstellten
30
           //Zeitpunkt ab und kann so nicht getestet werden
31
           String string2 = "leave,0,";
32
           String string3 = ",4:3,4,Bus0";
33
           String string4 = parkhaus.Leave(4, 3);
34
           assertEquals(string4.startsWith(string2) && string4.endsWith(string3), true);
35
36
37
```

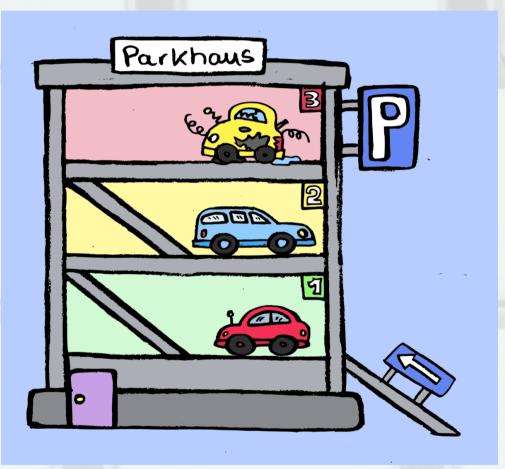
```
3⊕ import static org.junit.Assert.assertEquals;
14 public class StatisticTest {
16⊖
        public void addTest() {
18
            Statistik stats = new Statistik(10);
             Date d0 = new Date();
20
             String a = "enter,0,"+new Timestamp(d0.getTime())+",-,-,4:3,4,Bus0";
             Car bu0 = new Car(4, d0);
22
             Date d1 = new Date();
             String b = "enter,1,"+new Timestamp(d1.getTime())+",-,-,0:3,3,PkW1";
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
             Car pk0 = new Car(3, d1);
             Date d2 = new Date();
             String c = "enter,2,"+new Timestamp(d2.getTime())+",-,-,0:2,2,Ro2";
             Car ro0 = new Car(2, d2);
            assertEquals(stats.neuerEintrag(a, bu0),a);
            assertEquals(stats.neuerEintrag(b, pk0),b);
            assertEquals(stats.neuerEintrag(c, ro0),c);
             assertEquals(stats.getDrin(), 3);
            assertEquals(stats.verteilungGes(),"0,0,1,1,1,0");
assertEquals(stats.verteilungAkt(),"0,0,1,1,1,0,7");
            Date d4 = new Date();
            String dv = "leave,0,"+new Timestamp(d4.getTime())+",-,-,4:3,4,Bus0";
             long du = stats.getParkdauer(a, dv);
             String sdf = new SimpleDateFormat("SS:mm:ss").format(du);
             String[] sdfa = sdf.split(":");
             int time = Integer.parseInt(sdfa[2]) + Integer.parseInt(sdfa[1])*60 + Integer.parseInt(sdfa[0])*3600;
             double price = time * 0.005;
             String dn = "leave,0,"+new Timestamp(d4.getTime())+","+time+","+price+",4:3,4,Bus0";
45
             assertEquals(stats.neuerEintrag(dv, bu0),dn);
             assertEquals(stats.getDrin(), 2);
             assertEquals(stats.verteilungGes(),"0,0,1,1,1,0");
49
             assertEquals(stats.verteilungAkt(), "0,0,1,1,0,0,8");
50
51
52
```



- Was lief gut?
 - → Was waren Erfolgsrezepte?

- Was lief schlecht?
 - → Was könnte man beim nächsten mal anders machen?

Danke fürs Ansehen



Quelle: https://teddylingua.de/bildkarten-transport/