Robustheitsanlyse:

```
@startuml
actor Car

rectangle ParkhausSystem {
  entity ParkhausManager
  entity UndoRedo
  entity Statistik

  boundary enter
  boundary leave

  Car -up- enter
  Car -right- leave

  control Parkhaus

  Parkhaus .down.> leave
  Parkhaus .down.> enter
  Parkhaus .left.> ParkhausManager
  Parkhaus .right.> UndoRedo
  UndoRedo .up.> Statistik
}
@enduml
```

http://www.plantuml.com/plantuml/uml/PP1HIi0m38RVSufSO3k237mMnE41uXgtiYuTQQeCyUws8gsmj_1_7-N_EoKZjNm4eCMYuYCfWF9Y96jWV2RTDyfflfBnWL-0o6BUhaQUI6XbxU1LN7nX5tio6vbFvdSesLlCuaYl2bd_1u7fWm6mTi0XdqDJVWFrwsP3jvOefZ6q8YLhDuukVihuy6VVGNkwWy3lLi1zLrTgXUhqYUt4CP-5zAdVCB6uyh4_

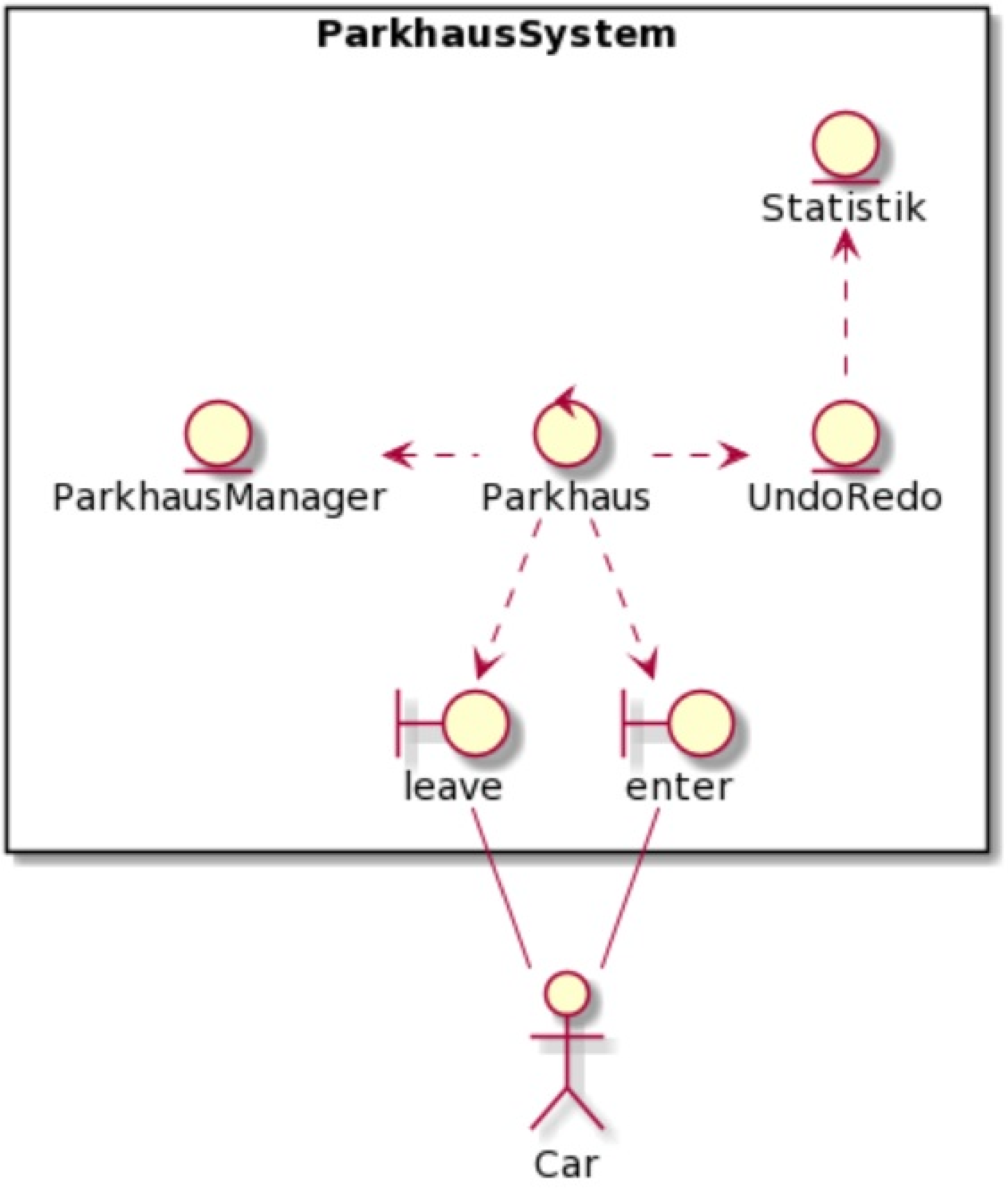

*Abbildung 1*

Sequenz:

```
@startuml
actor Kunde
boundary Parkhaus
control ParkhausManager
control Statistik
database Parkplaetze

Kunde -> Parkhaus : enter
Parkhaus -> ParkhausManager : enter
```

```
ParkhausManager -> Parkplaetze : frei?
Parkplaetze -> ParkhausManager : Platz
ParkhausManager -> Parkhaus : Abbruch wenn voll
Parkhaus -> Kunde : Abbruch wenn voll
ParkhausManager -> Parkhaus : Platz
Parkhaus -> Statistik : neuer Eintrag
Statistik -> Parkhaus : Return
Parkhaus -> Kunde : Platz + Ticket
@enduml
```

http://www.plantuml.com/plantuml/uml/VP31IWKn34Nt-OhinHzmeS-5An4UwW-adVYcJ2ULD5MShtVe8kXGtgfmx-7qjgTgfDRM31IjA3utcHX2sG_I7poJBZEr2h68QSaUl93GXTNpDoDBrT824na5gjpHpqni6mDqBzxUkm3laCLOmGCltNvaFE-estViGpazm9ziR3jdicriyrMF8MYBCtwp27wLdFyjxEyOKrUjVhER_CVsLhYnubCIKxg0DmV7ArjJ6SxfUhp1znGNDZYnJ6tDlm00

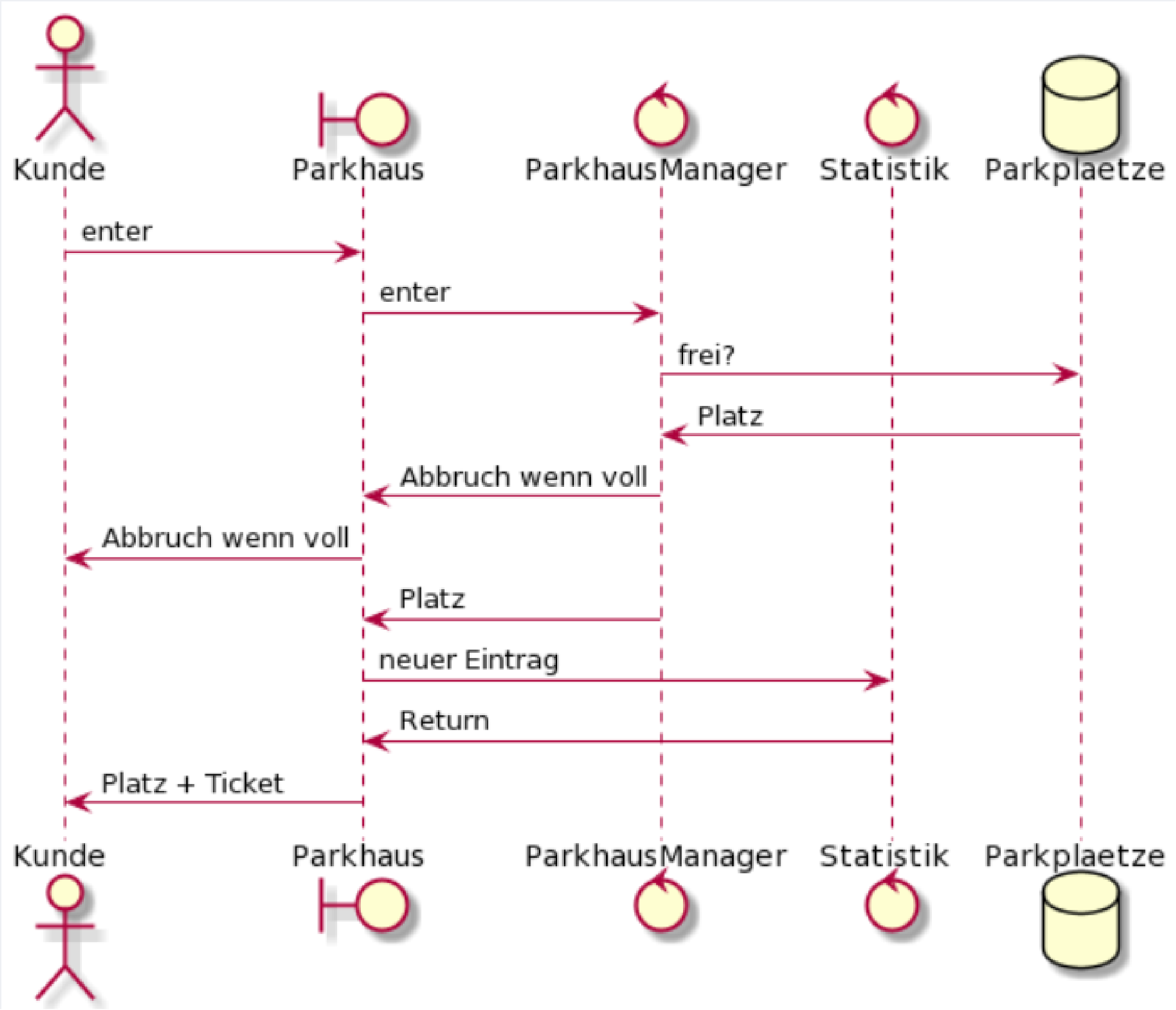

*Abbildung 2*

Use Case UML:

```
@startuml
skinparam packageStyle rectangle
actor Kunde

rectangle Parkhaus {
  usecase "parkhaus" as UC1
  usecase "ticket" as UC2
  usecase "payment" as UC3
  usecase "parking spot" as UC4
  usecase "ParkhausManager" as UC5
  usecase "Statistik" as UC6
}

UC5 .left.> UC2 : include
UC6 .up.> UC3 : include
UC4 <.down. UC5 : include
UC1 .down.> UC5 : include
UC6 <.up. UC1 : include


Kunde --> UC1 : enter/leave
Kunde --> UC2 : get by enter
Kunde --> UC3 : pay by leave
Kunde --> UC4 : parking on
@enduml
```

http://www.plantuml.com/plantuml/uml/PT31IiGm40RW-pp5iFUjh5izo58MF8eWB3x0c8urD4r3Cb6A-EvkjgaqxlN__ao6EGOXBt4m47fj7NaQq97ggUEJJ8RHinAodM4W9QF7fsXR1lYByOLy_q4nu3SWni2A0kF6vNI35F3rSRSsqQfdoN9Rlfe6jWlj_m_KjiFWniNhjIzhF9EbZdsktAqh9o7HGNIVyHv-0CuTh0o_IzMaRV01jLKcjWpd0bRHNV9zaTTugDhnorRfXq9sE4DJo3pha8QbMwnoWCizSRjjih0LzZU6wPCBIvjrBFWspPK2qtgEfeJNB-k4-NgZXIFRDWxc5m00

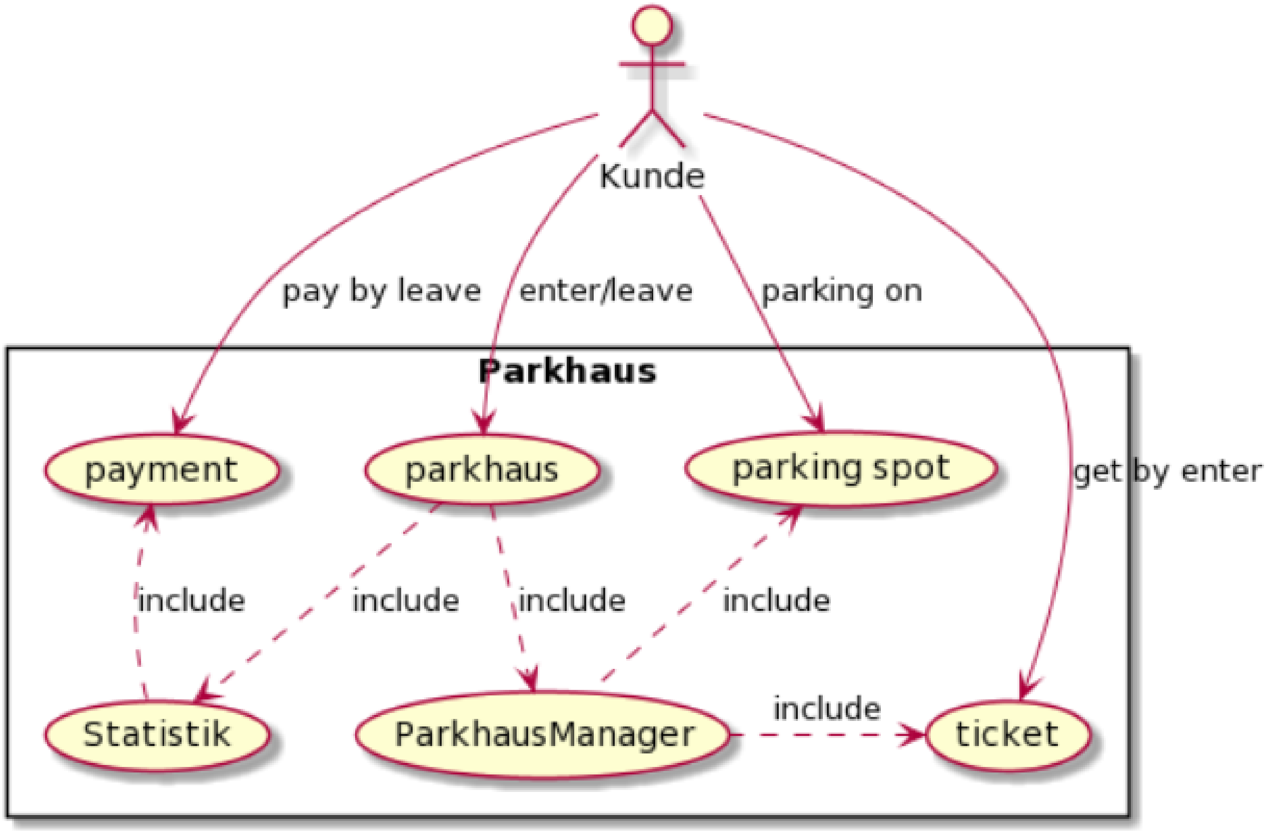

*Abbildung 3*

UML Klassendiagramm:

```
@startuml
skinparam classAttributeIconSize 0
class Parkhaus <<servlet>> {
  -stast: Statistic {static}
  -pm: ParkhausManager {static}
  -ur: UndoRedo {static}
  +Parkhaus()
  +doGet(request: HttpServletRequest, response: HttpServletResponse)
  +doPost(request: HttpServletRequest, response: HttpServletResponse)
  +enter(c: Car): String
  +leave(c: Car): String
  +getPm(): ParkhausManager
  +getStats(): Statistic
}
@startuml
skinparam classAttributeIconSize 0
class Parkhaus <<servlet>> {
  -stats: Statistic {static}
  -pm: ParkhausManager {static}
  -ur: UndoRedo {static}
  +Parkhaus()
  +doGet(request: HttpServletRequest, response: HttpServletResponse)
  +doPost(request: HttpServletRequest, response: HttpServletResponse)
  +enter(c: Car): String
  +leave(c: Car): String
  +getPm(): ParkhausManager
  +getStats(): Statistic
}

class Car {
  -snr: Integer {static}
  -nr: int
  -art: int
  -enter: Date
  -leave: Date
  +Car(enter: Date)
  +Car(art: int, enter: Date)
  +setLeave(d: Date)
  +getNr(): int
  +getArt(): int
  +getEnter(): Date
  +toString(): String

}

@startuml
skinparam classAttributeIconSize 0
class Parkhaus <<servlet>> {
  -stats: Statistik {static}
  -pm: ParkhausManager {static}
  -ur: UndoRedo {static}
  +Parkhaus()
  +doGet(request: HttpServletRequest, response: HttpServletResponse)
  +doPost(request: HttpServletRequest, response: HttpServletResponse)
  +enter(c: Car): String
  +leave(c: Car): String
  +getPm(): ParkhausManager
  +getStats(): Statistic
}

class Car {
  -snr: int {static}
  -nr: int
  -art: int
  -enter: Date
  -leave: Date
  +Car(enter: Date)
  +Car(art: int, enter: Date)
  +setLeave(d: Date)
  +getNr(): int
  +getArt(): int
  +getEnter(): Date
  +toString(): String

}

class Statistik {
  -protokoll: List<String>
  -autos: List<Car>
  -rein: int
  -raus: int
  -drin: int
  -verteilungGes: int[]
  -verteilungAkt: int[]
  +Statistic(max: int)
  +neuerEintrag(s: String, c: Car): String
  +getParkdauer(rein: String, raus: String): long
  +undo()
  +redo()
  +erzeugeTabelle(): String
  +einnahmen(): String
  +verteilungGes(): String
  +verteilungAkt(): String
  +getEnter(): int
  +getDrin(): int
  +toString(): String
}

class ParkhausManager <<iterable>>{
  -plaetze: Car[][]
  -pArt: int[][]
  +ParkhausManager()
  +ParkhausManager(p: int)
  +ParkhausManager(e: int, p: int)
  +ParkhausManager(be: int, fr: int, ro: int, pk: int, bu: int, fi: int)
  +makeTable(): String
  +Enter(c: Car, e: int, p: int): Car
  +Leave(e: int, p: int): Car
  +getFree(art: int): String
  +getPlaetze(): Car[][]
  +getpArt(): int[][]
  +iterator(): Iterator
  -sortPArt(be: int, fr: int, ro: int, pk: int, bu: int, fi: int)
  -isPrime(p: int): boolean
}

class ParkhausManagerIterator {
  -plaetze: Car[][]
  -e: int
  -p: int
  +ParkhausManagerIterator(p: Car[][])
  +hasNext(): boolean
  +next(): String
}

class UndoRedo {
  -parkhaus: Parkhaus
  -parkhausManager: ParkhausManager
  -statistik: Statistik
  -befehl: String
  -auto: Car
  -fertig: Stack<String>
  -fertigAutos: Stack<Car>
  -unfertig: Stack<String>
  -unfertigAutos: Stack<Car>
  +UndoRedo(parkhaus: Parkhaus)
  +neuerBefehl(s: String, c: Car)
  +aktivieren(): String
  +undo()
  +redo()
}

class Einnahmen <<abstract>> {
  ~einnahmen: double
  ~protokoll: List<String>
  +Einnahmen(protokoll: List<String>)
  +einnahmen(): double
  +getEinnahmen(datum: String): double[]
  +toString(): String
  +getDatum(): String <<abstract>>
}

Car -down-> Parkhaus: parkt_in
Car --> ParkhausManager: wird_gespeichert_in
Parkhaus -down-> Statistik: laesst_Statistik_fuehren
ParkhausManager <- Parkhaus: wird_verwaltet
Parkhaus -down-> UndoRedo: laesst_befehle_verwalten
Statistik <-left- UndoRedo: leitet_befehle_an
ParkhausManagerIterator -down-> ParkhausManager: iteriert
Einnahmen <-left- Statistik: laesst_berechnen
@enduml
```

http://www.plantuml.com/plantuml/uml/fLP1hzis3BxdL-Ze82_0paOGD5lVkW9T4VHjfw9ua6Cc4InB6KMbnIkwtxv8icZxETvXkrdVHrBaHuh96qSIoRTwuHfbBX9bAmvQEhSZGbLvWlS7QvxK2uYV5f4GUudDMNedDXi7UDL0swtulX1YxKYIAyKJILAEL2E-1qGTVWJoqfRi-hiqyWGuuZsMuazJsqzGso6noax5Cfng-mweGFZBWw

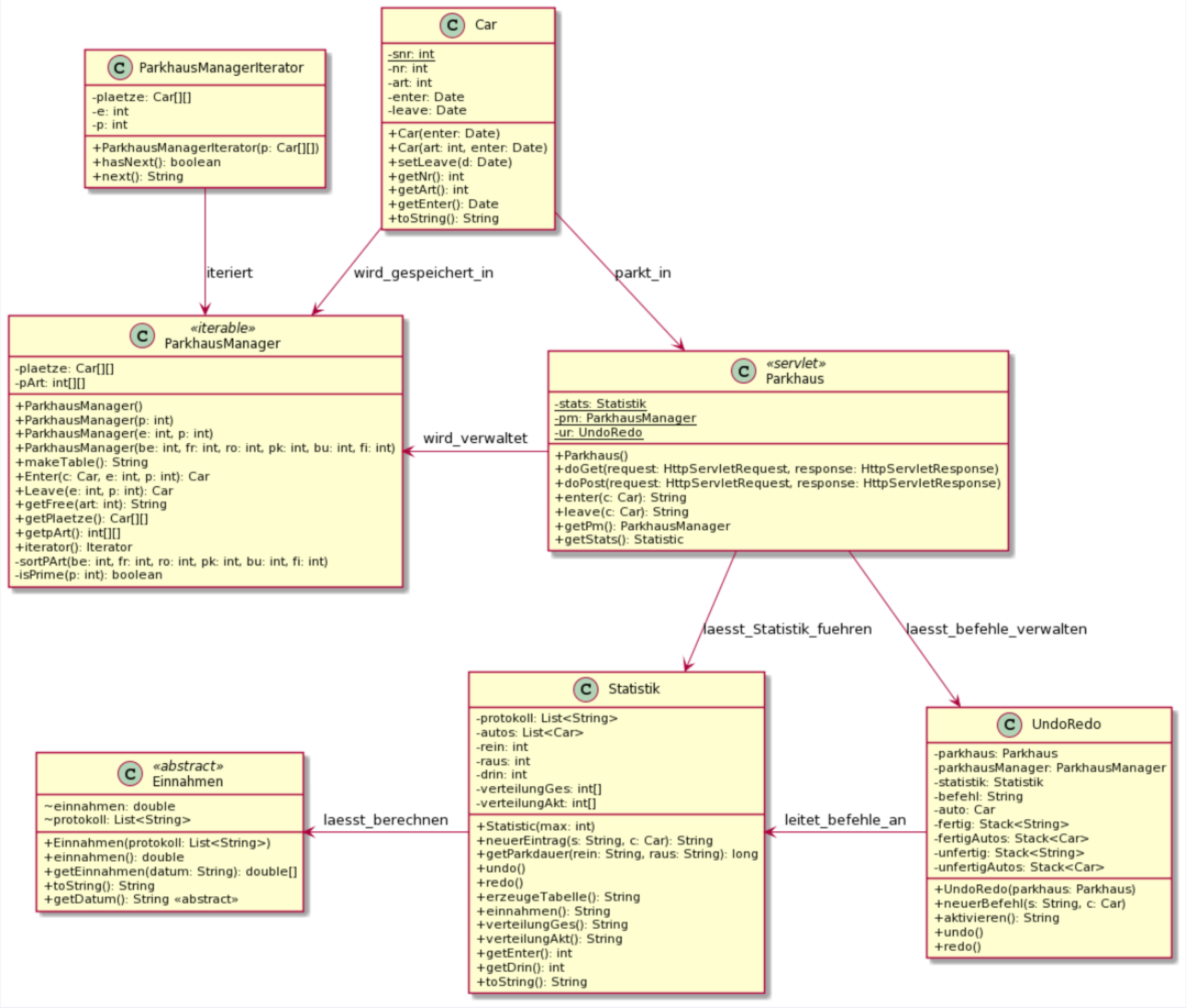

*Abbildung 4*