

# SMARTE PARKPLATZLÖSUNG

Muhamad Wahdan , Maximilian Jaik , Morris Schacht

# AGENDA

- Einführung
- Reservierung
- Kommunikation zwischen Systemkomponenten
- Anforderungen
- Erfordernisse
- Proof of Concept
- Risiken

# AGENDA

- Stakeholder
- Deskriptives Domänenmodell
- Präskriptives Domänenmodell
- Klassendiagramm
- Quellcode
- Verfügbarkeiten
- Zuweisung
- Benachrichtigung
- Zusammenfassung

# EINFÜHRUNG

- **Parkplatzmangel** : Die Suche nach einem Parkplatz stellt eine alltägliche Herausforderung dar , die kostbare Zeit und Nerven raubt .
- **Smarte Parkplatzlösungen** : Das System bietet innovative Ansätze, um das Problem des Parkplatzmangels zu lösen .
- **Reservierung** :Die Schwerpunkte der smarten Parkplatzlösung liegen auf der Reservierung von Parkplätzen und der Anzeigen der Parkplatzverfügbarkeiten

# RESERVIERUNG

- **Stressreduktion** : Die Möglichkeit , einen Parkplatz im Voraus zu reservieren , reduziert den mit der Parkplatzsuche verbundenen Stress.
- **Parkplatzsuche** : Die Reservierung von Parkplätzen beseitigt Wettlauf um freie Parkplätze

# KOMMUNIKATION ZWISCHEN SYSTEMKOMPONENTEN

- **Sensoren** : Sensoren sammeln Echtzeitdaten vor den Parkplätzen .
- **Plattform** : Eine zentrale Plattform analysieren die Daten und liefert aktuelle Verfügbarkeitsinformationen.

# ANFORDERUNGEN

- 1. Das System muss die Fähigkeit haben, die Verfügbarkeit von Parkplätzen in Echtzeit zu überwachen um die Effizienz bei der Parkplatzsuche zu verbessern.
- 2. Das System muss Benutzern Informationen zugänglich machen,
- 3. Das System muss in der Lage sein, automatisch Parkplätze zuzuweisen um eine effiziente Nutzung der Parkplätze zu ermöglichen.
- 4. Das System und dabei Präferenzen der Autofahrer, wie z.B. Nähe zu Eingängen oder Sicherheitsfaktoren, zu berücksichtigen,
- 5. Das System muss die Fähigkeit besitzen, Daten über die Auslastung der Parkplätze zu sammeln um Berichte für den Parkgaragenbetreiber zu generieren.
- 6. Das System muss sicherstellen, dass alle gesammelten Benutzerdaten gemäß den Datenschutzbestimmungen behandelt werden, um die Privatsphäre der Benutzer zu gewährleisten.
- 7. Das System muss in der Lage sein, Benachrichtigungen in Echtzeit an Autofahrer zu senden, um sie über die Parkplatzverfügbarkeit zu informieren.

# ANFORDERUNGEN

- 8. Das System muss die Fähigkeit besitzen, Anpassungen an unterschiedliche Standorte und Umgebungen vorzunehmen, um eine flexible Implementierung in verschiedenen städtischen Kontexten zu ermöglichen.
- 9. Das System muss Berichte über die ökologischen Auswirkungen der intelligenten Parklösung für Umweltschützer bereitstellen, um die positiven Umweltauswirkungen zu dokumentieren.
- 10. Das System muss in der Lage sein, statistische Daten über die Reduzierung von Verkehrsunfällen und Schäden im Straßenverkehr zu generieren, um Versicherungsunternehmen dabei zu unterstützen, prämienmindernde Maßnahmen zu ergreifen.
- 11. Das System muss eine Benutzeroberfläche bieten, die benutzerfreundlich ist und es Autofahrern ermöglicht, mühelos Parkplatzreservierungen vorzunehmen, Zahlungen durchzuführen und auf Kundensupport zuzugreifen.



# ANFORDERUNGEN

- 12. Das System muss eine Skalierbarkeit aufweisen, um bei Bedarf erweitert zu werden
- 13. Das System muss Fernwartungstools bereitstellen, um Wartungspersonal in die Lage zu versetzen, zeitnahe Wartungsarbeiten durchzuführen.
- 14. Das System muss Fehlerprotokolle bereitstellen um Wartungspersonal bei der Fehlerbehebung zu unterstützen
- 15. Das System muss eine Möglichkeit bieten, anonymisierte Daten für Forschungszwecke bereitzustellen, um Forschungsinstituten die Analyse der Auswirkungen des intelligenten Parksystems auf städtische Mobilität zu ermöglichen.

# ERFORDERNISSE

- 1. Als Autofahrer muss man die Verfügbarkeit von Parkplätzen in Echtzeit wissen, um einen geeigneten Parkplatz finden und reservieren zu können.
- 2. Als Parkhausbestreiber muss man eine übersichtliche Aufzeichnung über die Belegung aller Parkplätze haben, um ein effizientes Ressourcenmanagement der Parkplatzauslastung durchführen zu können.
- 3. Als Parkhausbestreiber muss man eine übersichtliche Aufzeichnung über die Belegung aller Parkplätze haben, um eine Optimierung der Parkplatzauslastung durchführen zu können.
- 4. Als Umweltschützer muss man Informationen zur Reduzierung von CO<sub>2</sub>-Emissionen durch das System wissen, um die Umweltauswirkungen des Systems bewerten zu können.
- 5. Als Umweltschützer muss man Informationen zur Reduzierung von Verkehrsaufkommen durch das System wissen, um die Umweltauswirkungen des Systems bewerten zu können.
- 6. Als Versicherungsunternehmen muss man Zugang zu Daten über Unfall- und Schadensraten im Straßenverkehr haben, um die Auswirkungen des intelligenten Parksystems auf die Verkehrssicherheit beurteilen und prämienmindernde Maßnahmen ergreifen zu können.

# ERFORDERNISSE

- 7. Als App-Nutzer muss man eine benutzerfreundliche mobile Anwendung haben, um einfach Parkplatzreservierungen vornehmen zu können.
- 8. Als App-Nutzer muss man eine benutzerfreundliche mobile Anwendung haben, um einfach Zahlungen durchführen zu können.
- 9. Als App-Nutzer muss man eine benutzerfreundliche mobile Anwendung haben, um einfach Benachrichtigungen erhalten zu können.
- 10. Als Entwickler des intelligenten Parksystems muss man Datenverschlüsselung und Datenschutzmaßnahmen implementieren, um die Sicherheit und Privatsphäre der Benutzerinformationen zu gewährleisten.
- 11. Als Behörde für Verkehrsplanung muss man Zugang zu Berichten über die Verkehrseffizienz erhalten um Stadtverkehrsstrategien zu optimieren.

# ERFORDERNISSE

- 12. Als Fahrzeughersteller muss man kompatible Schnittstellen für das Parksystem bereitstellen, um eine nahtlose Interaktion zwischen Fahrzeugen und dem intelligenten Parksystem zu ermöglichen.
- 13. Als Wartungspersonal des Parksystems muss man Fernwartungstools und -protokolle haben, um zeitnahe Wartungsarbeiten und Fehlerbehebungen durchführen zu können.
- 14. Als Stadtplaner muss man Zugang zu Daten über Parkraumnutzung und Verkehrsfluss haben, um Städtebau und Infrastrukturprojekte besser planen zu können.
- 15. Als Stadtplaner muss man Zugang zu Daten über den Verkehrsfluss erhalten, um Infrastrukturprojekte besser planen zu können.
- 16. Als Forschungsinstitut muss man anonymisierte Daten für Forschungszwecke erhalten können, um die Auswirkungen intelligenter Parksysteime auf Umweltauswirkungen zu analysieren.

# ERFORDERNISSE

- 16. Als Forschungsinstitut muss man anonymisierte Daten für Forschungszwecke erhalten können, um die Auswirkungen intelligenter Parksysteme auf städtische Mobilität zu analysieren.
- 17. Als Kundensupportmitarbeiter des Parksystems muss man ein effizientes System für Problemlösungen haben, um einen qualitativ hochwertigen Kundenservice sicherstellen zu können.

# PROOF OF CONCEPT

- 1. Parkplatzbelegungserfassung (3 Tage ) - Implementierung eines Prototyps für die Parkplatzbelegungserfassung unter Verwendung von Sensoren.
- Simulation von Sensordaten, um die Verfügbarkeit von Parkplätzen zu visualisieren.
- 2. Benutzeroberfläche für Autofahrer (2 Tage):
  - Entwurf und Implementierung einer einfachen Benutzeroberfläche für Autofahrer zur Anzeige von Echtzeit-Parkplatzverfügbarkeiten und zur Reservierung von Parkplätzen.
  - Integration der Benutzeroberfläche mit dem Sensordaten-Prototyp.
- 3. Automatische Parkplatzzuweisung (2 Tage):
  - Implementierung von Algorithmen zur automatischen Parkplatzzuweisung basierend auf Verfügbarkeit und Benutzerpräferenzen.
  - Simulation von Reservierungen und automatischen Zuweisungen.

# PROOF OF CONCEPT

- 4. Datenaustauschsimulation (2 Tage):
  - Simulation des Datenaustauschs zwischen dem Parksystem und mobilen Geräten der Benutzer.
  - Überprüfung der Zuverlässigkeit und Geschwindigkeit des Datenaustauschs
- 5. Berichtsmodul für Betreiber (2 Tage):
  - Entwicklung eines einfachen Berichtsmoduls für Betreiber zur Analyse der Parkplatznutzung und Effizienz des Systems.
  - Erstellung von Beispielsberichten zur Auswertung.

# PROOF OF CONCEPT

- Messbare Erfolgskriterien:
- Erfolgreiche Erfassung und Anzeige von Echtzeit-Parkplatzverfügbarkeiten.
- Erfolgreiche Reservierung und Zuweisung von Parkplätzen über die Benutzeroberfläche.
- Effiziente Simulation des Datenaustauschs zwischen dem System und mobilen Geräten.
- Zuverlässige Umsetzung der automatischen Parkplatzzuweisungsalgorithmen.
- Erstellung von aussagekräftigen Berichten über die Parkplatznutzung und -effizienz.



# RISIKEN

- Sicherheitslücken: Technische Schwächen in der Systemarchitektur könnten zu Datenschutz- und Sicherheitsproblemen führen.
- Fehlende Fachkenntnisse: Unzureichende Kompetenzen des Entwicklerteams im Bereich intelligenter Parkplatzlösungen könnten zu Entwicklungsverzögerungen oder Qualitätsproblemen führen.
- .Unzureichende Schulung: Mangelnde Schulung der Benutzer und Betreiber könnte zu ineffizienter Nutzung und Betriebsproblemen führen.

# RISIKEN

- Technische Herausforderungen:
- Komplexität bei der Integration von Sensoren und der Gewinnung genauer Sensordaten könnte zu Verzögerungen führen.
- Schwierigkeiten bei der Entwicklung zuverlässiger Algorithmen für die automatische Parkplatzzuweisung könnten die Umsetzung beeinträchtigen.
- Benutzeroberfläche und Benutzererfahrung:
- Akzeptanzprobleme könnten auftreten, wenn die Benutzeroberfläche nicht benutzerfreundlich ist oder wenn die Echtzeitinformationen nicht korrekt angezeigt werden.
- Verzögerungen oder Fehler in der Reservierungsfunktion könnten die Nutzererfahrung negativ beeinflussen.

# RISIKEN

- Datenaustausch und Sicherheit:
- Schwierigkeiten beim Simulieren zuverlässiger und schneller Datenaustauschprozesse zwischen dem Parksystem und mobilen Geräten könnten die Funktionalität beeinträchtigen.
- Sicherheitsrisiken im Zusammenhang mit dem Datenaustausch müssen sorgfältig berücksichtigt werden, um Datenschutz und Integrität zu gewährleisten.
- Simulation und Realitätsnähe:
- Die Simulation könnte von der tatsächlichen Umgebung abweichen, was zu inkorrekten Annahmen und Ergebnissen führen könnte.
- Schwierigkeiten bei der Erstellung realistischer Beispielsberichte könnten die Aussagekraft der Berichtsmodule beeinträchtigen.

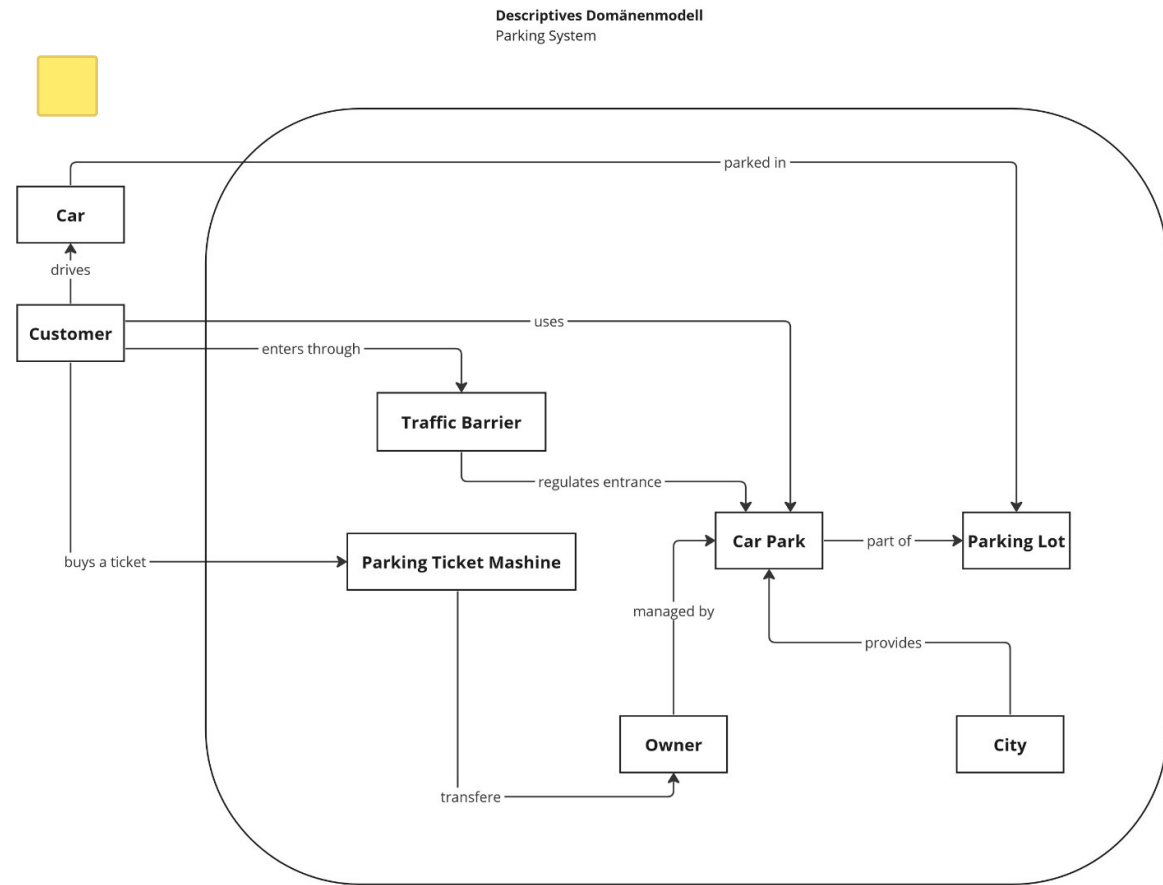
# RISIKEN

- Zeitliche Einschränkungen:
- Die festgelegte Frist von 15 Arbeitstagen könnte zu Zeitdruck führen und die Qualität der Implementierung sowie die gründliche Überprüfung der Funktionalitäten beeinträchtigen.
- Ressourcenmangel:
- Ein Mangel an qualifizierten Entwicklern, Testern oder Datenanalysten könnte zu Engpässen führen und die Effizienz des Projekts beeinträchtigen.
- Akzeptanz durch Stakeholder:
- Schwierigkeiten bei der Akzeptanz des Systems durch Autofahrer und Betreiber könnten auftreten, insbesondere wenn die Vorteile des Systems nicht klar kommuniziert werden.

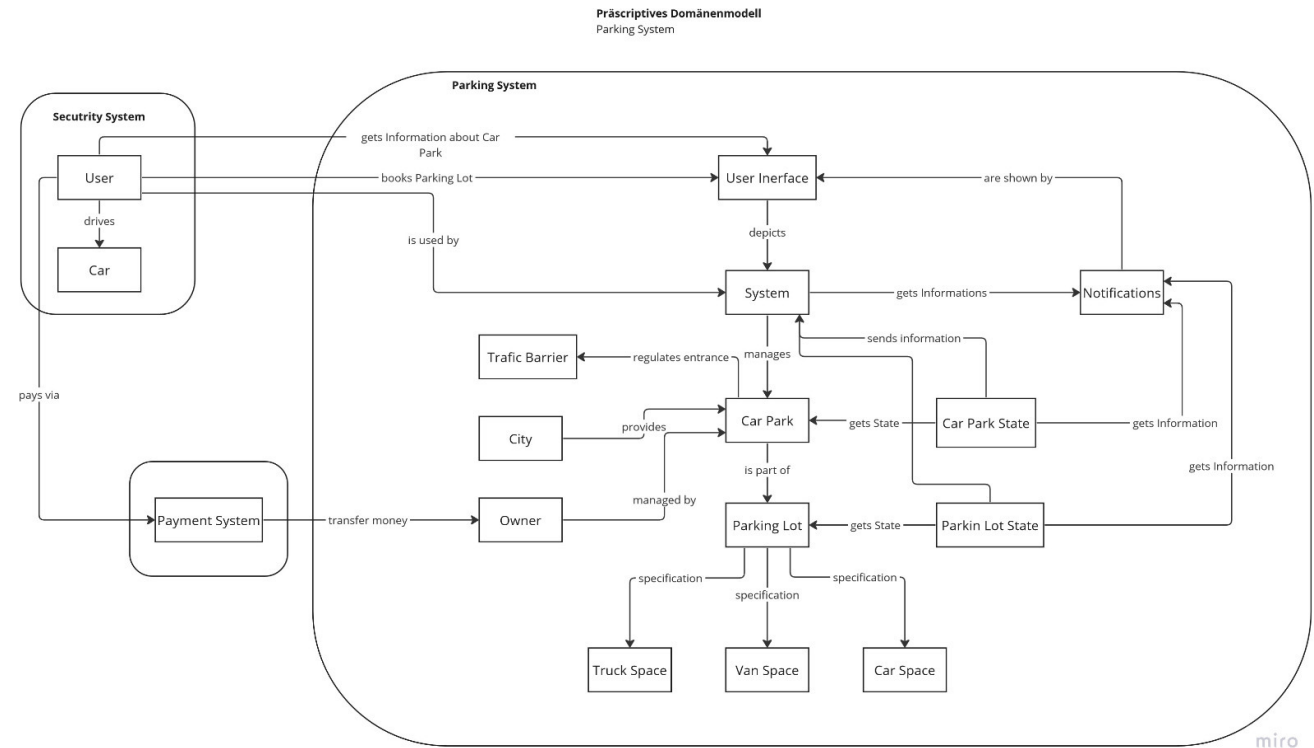
Bezeichnung	Einz	Orga	Bezug System	Objektbereich	Erwartung
Autofahrer	Nein	Ja	Interesse	Kunde	Stressfreies Parken
Parkhausbetreiber	Nein	Ja	Anteile	Anbieter	Einfaches managen des Parkhauses
Parkhausbetreiber	Nein	Ja	Interesse	Anbieter	Kostenoptimierung
Umweltschützer	Nein	Nein	Interesse	Regulierung	Niedriger CO2 Bilanz in der Stadt
Umweltschützer	Nein	Nein	Anspruch	Regulierung	Einfache Überwachung der Umweltauswirkungen
Versicherungsunternehmen	Nein	Ja	Interesse	Recht	Einfache nachvollziehbarkeit bei Unfällen

# STAKEHOLDER

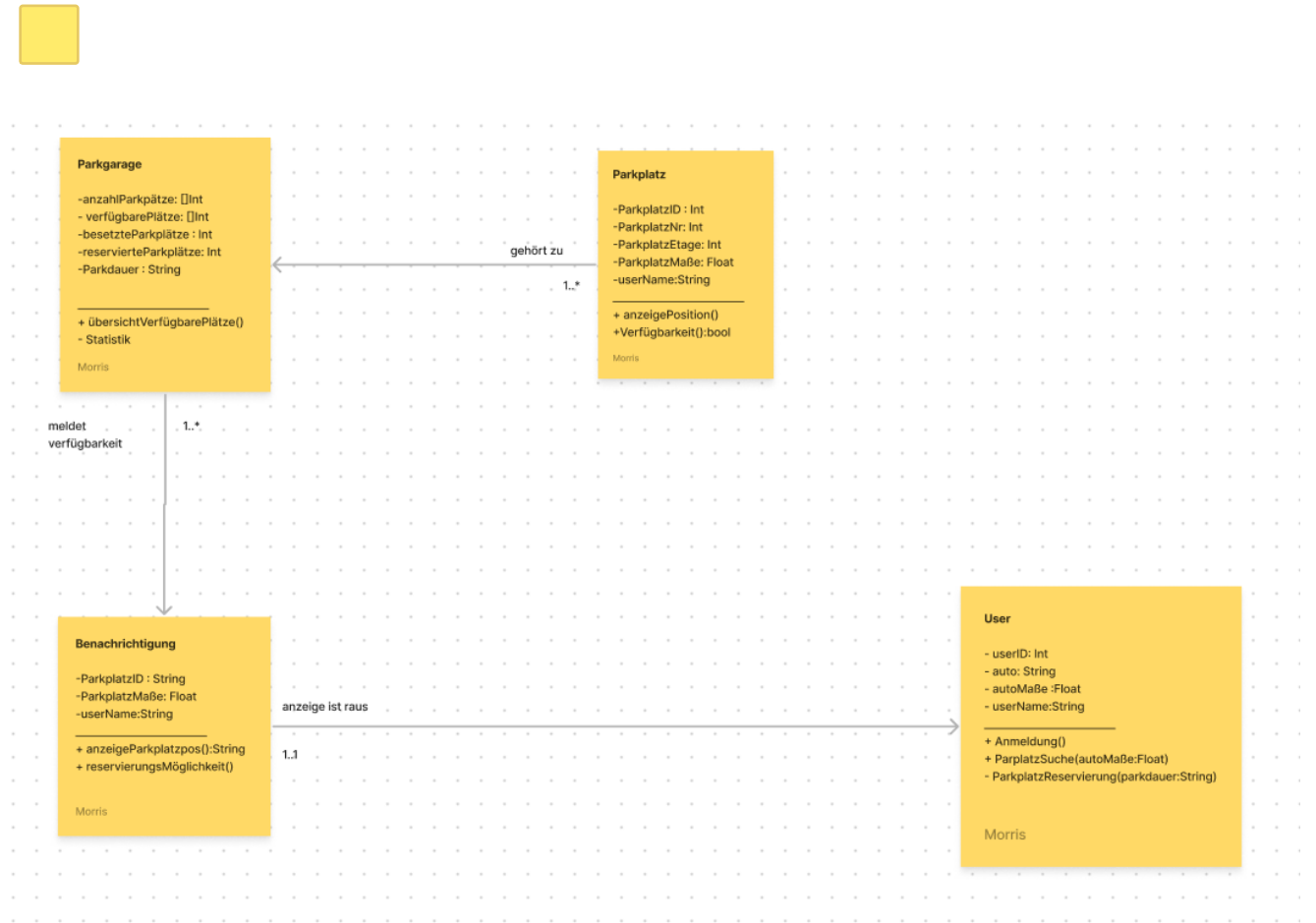
# DESKRIPTIVES DOMÄNENMODELL



# PRÄSKRIPTIVES DOMÄNENMODELL



# KLASSENDIAGRAMM





# QUELLCODE

```
3 class DBManagement {
4   constructor(id) {
5     this.id = id;
6   }
7
8   jsonDateiAuslesen(dateiName) {
9     return fs.readFileSync(dateiName, 'utf-8');
10  }
11
12  dbDateiUpdaten(platzIndex, status) {
13    const dateiName = './DB/DB.json';
14
15    // Datei auslesen
16    const jsonData = JSON.parse(fs.readFileSync(dateiName, 'utf-8'));
17
18    // Statusänderung des Platzes mit dem übergeben Index
19    const zuAktualisierenderParkplatz = jsonData.parkhaus.parkplaetze.find(parkplatz => parkplatz.platzNummer === platzIndex);
20
21    if (zuAktualisierenderParkplatz) {
22      // Status Aktualisieren
23      zuAktualisierenderParkplatz.status = status;
24
25      // aktualisierte Daten speichern
26      fs.writeFileSync(dateiName, JSON.stringify(jsonData, null, 2));
27
28      console.log('Parkplatz wurde erfolgreich aktualisiert.');
```

# QUELLCODE

```
1  const Parkhaus = require('./framework/Parkhaus.js');
2  const DBManagement = require('./framework/DBManagement.js');
3
4  const express = require('express');
5  const path = require('path');
6  const app = express();
7  const port = 3000;
8
9  app.use(express.static('public'));
10
11  |
12  |
13  //Alle Daten des Parkhaus
14  app.get('/', (req, res) => {
15
16      const DBManagementInstanz1 = new DBManagement();
17      const DBJsonData = DBManagementInstanz1.jsonDateiAuslesen('DB.json');
18      res.send(DBJsonData);
19  })
20
21
22  app.get('/update', (req, res) => {
23      //hier wird das updaten der DB.json implementiert
24      const DBManagementInstanz2 = new DBManagement();
25      DBManagementInstanz2.dbDateiUpdaten(1, 'belegt');
26      res.send(DBManagementInstanz2.jsonDateiAuslesen('./DB/DB.json'));
27  })
28
29
30  app.post('/reservierung', (req, res) => {
31      console.log('JSON:', req.body);
32      res.send(req.body)
33  })
34
35  app.get('/getReservierung', (req, res) => {
36      const indexPath = path.join(__dirname, 'public', 'reservierung.html');
37      res.redirect(indexPath);
38  })
39
40
41  app.listen(port, () => console.log
42  (`App listening on http://localhost:${port}`))
43
```

# QUELLCODE

```
1  {
2    "parkhaus": {
3      "name": "Parkhaus 1",
4      "standort": "Musterstraße 1",
5      "kapazitaet": 20,
6      "verfuegbarePlaetze": 20,
7      "parkplaetze": [
8        {
9          "platzNummer": 1,
10         "status": "belegt"
11       },
12       {
13         "platzNummer": 2,
14         "status": "frei"
15       },
16       {
17         "platzNummer": 3,
18         "status": "belegt"
19       },
```

# QUELLCODE

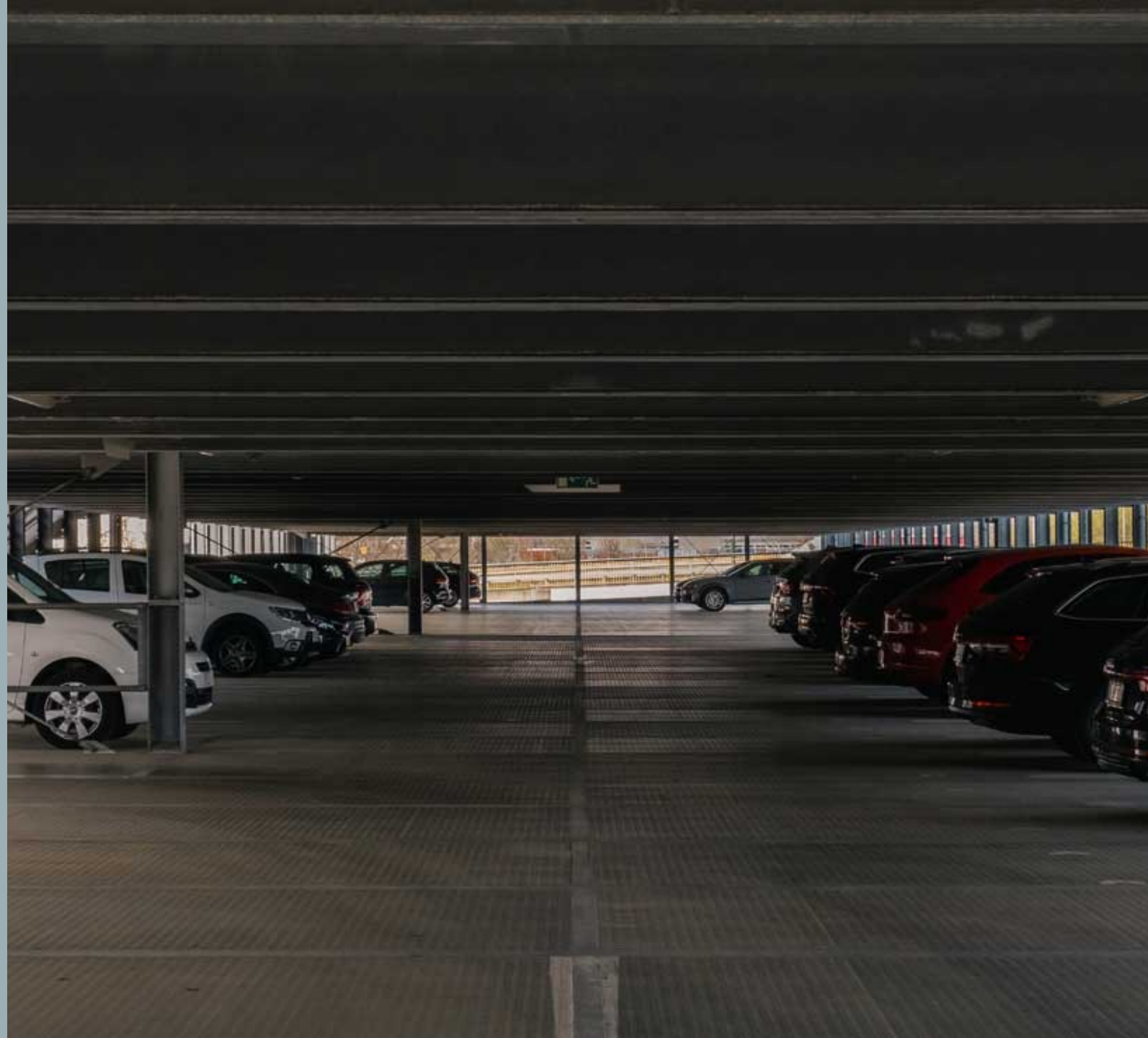
```
1  const Parkplatz = require('./Parkplatz.js'); // Passe den Pfad entsprechend an
2
3  class Parkhaus {
4    constructor(id, name, adresse) {
5      this.id = id;
6      this.name = name;
7      this.adresse = adresse;
8      this.parkplaetze = Array.from({ length: 20 }, (_, index) => new Parkplatz(index + 1, false, this));
9      this.anzahlBelegterParkplaetze = 0;
10   }
11
12   einfahren() {
13     // Implementierung des Einfahrens
14     const aktuelleZeit = new Date();
15     console.log(`Ein Fahrzeug ist ${aktuelleZeit} eingefahren.`);
16   }
17
18   ausfahren() {
19     // Implementierung des Ausfahrens
20     const aktuelleZeit = new Date();
21     console.log(`Ein Fahrzeug ist am ${aktuelleZeit} ausgefahren.`);
22   }
23
24   parkhausOverview() {
25     return {
26       parkhaus: {
27         name: this.name,
28         standort: this.adresse,
29         kapazitaet: this.parkplaetze.length,
30         verfuegbarePlaetze: this.parkplaetze.filter(p => !p.belegt).length,
31         parkplaetze: this.parkplaetze.map(p => ({ platzNummer: p.id, status: p.belegt ? "belegt" : "frei" })),
32       },
33     };
34   }
35
36   verfügbar() {
37     // Hier könntest du die Logik implementieren, um zu überprüfen, ob Parkplätze verfügbar sind.
38   }
39 }
40
41 module.exports = Parkhaus;
42
```

## VERFÜGBARKEITEN

Die Verfügbarkeitsanzeige informiert die Nutzer darüber, wann der nächste freie Parkplatz voraussichtlich verfügbar ist.

Dies ermöglicht es den Fahrern, ihre Ankunft zu planen und nicht unnötig im Verkehr herumzufahren.

Die Transparenz in Bezug auf die Parkplatzverfügbarkeit fördert eine effizientere Nutzung der vorhandenen Parkflächen



## ZUWEISUNG

Unsere Lösung basiert auf intelligenten Algorithmen und Sensoren, die die Parkplätze in Echtzeit überwachen . Dies ermöglicht eine effiziente Zuweisung von Parkplätzen , wodurch Leerstände minimiert und die Auslastung optimiert wird.

Die Zeit , in denen Fahrer endlose Runden um Parkhäuser drehen , werden dadurch vermieden .





## BENACHRICHTIGUNG

Die Benachrichtigungsfunktion informiert die Nutzer über den Ablauf ihrer Parkplatzreservierung und liefert während der Fahrt relevante InFormationen wie Verkehrslage oder alternative Parkmöglichkeiten.

Dies erhöht die Flexibillität und Effizienz im Parkplatzmanagement , indem Fahrer rechtzeitig reagieren können.





## ZUSAMMENFASSUNG

Die smarte Parkplatzlösung mit dem Schwerpunkten Reservierung , Navigation und Verfügbarkeit bietet eine Antwort auf das wachsende Problem des Parkplatzmangels in Städten und urbanen Gebieten.

Mit dieser innovativen Technologie wird nicht nur der Alltag der Fahrer erleichtert , sondern auch der Verkehrsfluss in den Städten verbessert.





## ZUSAMMENFASSUNG

Durch die Intelligenten Funktionen der smarten Parkplatzlösung wird der Verkehrsfluss optimiert, da Fahrer schnell und stressfrei zu verfügbaren Parkplätzen geführt werden.

Das System optimiert die Parkplatznutzung, spart Raum und schützt somit Grünflächen und verhilft zu einem niedrigeren CO<sub>2</sub> ausstoß

