

**CSE6224 SOFTWARE REQUIREMENTS ENGINEERING**

**Context Objects Documentation**

By TT2L Group F

Lecturer: Nur Haifa binti Mohd Fathil

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| Chia Kai Jun | 1211111053 |
| Wee Jia Sheen | 1211110222 |
| Yap Wei Jian | 243UC246NA |
| Tang Zhi Qian | 243UC246NP |

**Table of Contents**

[**1.0** **Project-Specific Guidelines** 3](#_Toc199028523)

[**2.0 System Context Objects** 5](#_Toc199028524)

[**2.1 Subject facets** 5](#_Toc199028525)

[**2.2 Usage facets** 6](#_Toc199028526)

[**2.2.1 All User (Common Task)** 6](#_Toc199028527)

[**2.2.2 Student and Staff Usage Facets** 6](#_Toc199028528)

[**2.2.3 Staff Usage facets** 6](#_Toc199028529)

[**2.2.4 Admin Usage facets** 7](#_Toc199028530)

[**2.3 IT System facets** 7](#_Toc199028531)

[**2.3.1 Sub System** 7](#_Toc199028532)

[**2.3.2 APIs** 8](#_Toc199028533)

[**2.3.3 Data Management** 8](#_Toc199028534)

[**2.3.4 Security and Access Control** 8](#_Toc199028535)

[**2.3.5 Monitoring and Logging** 9](#_Toc199028536)

[**2.4 Relationship Types** 9](#_Toc199028537)

[**2.4.1 Entities Relationship Model** 9](#_Toc199028538)

[**2.4.2 Relationship Table** 10](#_Toc199028539)

[**2.5 Roles and Responsibilities** 10](#_Toc199028540)

[**3.0 Development Context** 11](#_Toc199028541)

[**4.0 Requirement Resource** 13](#_Toc199028542)

[**4.1 Potential Relevant Requirement Source Identification** 13](#_Toc199028543)

[**4.1.1 Checklist for Carpool and Parking System** 13](#_Toc199028544)

[**4.2.2 Selection of Checklist for Carpool and Parking System** 15](#_Toc199028545)

[**4.2.3 Selection of Development Plan for Carpool and Parking System** 16](#_Toc199028546)

[**4.2.4**  **100 Dollars Test** 16](#_Toc199028547)

[**4.2.5 Adequate Representation of Stakeholders Selection** 17](#_Toc199028548)

[**4.2.6 Development Team Requirements and Concerns** 18](#_Toc199028549)

[**4.2.7 Selection Rationale** 19](#_Toc199028550)

[**5.0 Data Flow Diagram** 20](#_Toc199028551)

[**5.1 Ride Sharing platform with Parking System** 20](#_Toc199028552)

[**5.2 Reserve Parking** 21](#_Toc199028553)

[**5.3 Request Carpool** 23](#_Toc199028554)

# **Project-Specific Guidelines**

This documentation defines the key context objects, relationships, formats, and roles involved in building a Campus Ride-Sharing Platform with Parking System Integration to ensure consistent understanding and development.

The goal is to ensure that all stakeholders involved and share a common understanding of the various entities interacting within the system and the relationships between them.

Furthermore, the documentation outlines the roles and responsibilities of each party involved in documenting and maintaining this context information. This structured approach helps in reducing ambiguity, streamlining communication, supporting system scalability, and ultimately delivering a reliable and user-friendly platform tailored to the specific needs of a campus environment.

# **2.0 System Context Objects**

## **2.1 Subject facets**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| User | * ID (Primary Key) * Name * Email * Role |
| Admin\_Activity\_Log | * Log\_ID (Primary Key) * Admin\_ID (User ID Foreign Key) * Action * Target\_ID * Timestamp |
| Parking\_Slot | * Slot\_ID (Primary Key) * Faculty * Reserved\_by * Status |
| Carpool | * Carpool\_ID (Primary Key) * Driver\_ID (Foreign Key) * Destination * Departure Time * Available\_seats |
| Vehicle | * ID (Primary Key) * User\_ID (Foreign Key) * Plate\_Number * Color * Capacity |
| Carpool\_Request | * Request\_ID (Primary Key) * Carpool\_ID (Primary Key) * Request\_ID (Foreign Key) * Request\_time (Foreign Key) * Status |
| Booking | * Booking\_ID (Primary Key) * User\_ID (Foreign Key) * Reference\_ID * Booking Time |

## **2.2 Usage facets**

### **2.2.1 All User (Common Task)**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| Register / Login | Use university credentials |
| Manage Profile | Update user info, vehicle details, preferences |
| View carpool/parking history | Access logs of previous carpool and parking reservations |

### **2.2.2 Student and Staff Usage Facets**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| Reserved parking | Pre-book spot tied to a ride |
| Request carpool | Select a matching ride and send a request |
| View parking | use real-time data tracking parking status |
| View carpool details | Check driver name, vehicle plate number |
| View parking details | Check the vehicle plate number, owner name, parking status |

### **2.2.3 Staff Usage facets**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| Reserved parking | Pre-book spot tied to a ride |
| Offer parking | required a reserved parking for release. |
| Change parking | change reserved parking with another available parking. |

### **2.2.4 Admin Usage facets**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| Monitor System Activity | Dashboard of Carpool and Parking Metrics |
| Manage User | Profile Management and Permission |
| System Maintenance | Troubleshooting System Issue |

## **2.3 IT System facets**

### **2.3.1 Sub System**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| Carpool System | Provides carpool matching, booking, carpool history |
| Parking System | Real-time parking availability, reservation system. |
| User Authentication System | University login, role-based access control (student, staff, admin), user profile management |
| Mobile Application | User-friendly interface for carpool service, parking reservation, notifications, and real-time carpool updates |
| Web Portal | Admin dashboard  user, carpool and parking management  system monitoring tools |
| Backend Server (APIs) | Handles business logic, processes requests from mobile or web apps, connects to database and external systems |
| Database System | Stores user data, carpool info, parking data, notifications, supports backup and recovery |
| Monitoring and Logging System | Captures system logs, tracks errors and user activity, supports alerting for system health and security |
| Security Service | Data encryption, data protection policies, secure API tokens (JWT), compliance with campus IT and privacy standards |

### **2.3.2 APIs**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| RESTful APIs | To connect frontend and backend communication |
| Microservice | Separate services for Carpool Management, User Management, Notifications, Parking Reservation |

### **2.3.3 Data Management**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| Relational Database | Table for users, carpool, parking slot, parking reservation. |
| Data Backup and Recovery | Regular automated backups with secure restore capability |

### **2.3.4 Security and Access Control**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| Authentication | University Sign-in (SSO)  Optional two-factor authentication |
| Authorization | Role-based access control (Admin, Student, Staff, Lecturer) |
| Data Encryption | Encryption of sensitive data at RESTAPI |
| Secure API Access | Token-based access control for APIs |

### **2.3.5 Monitoring and Logging**

|  |  |
| --- | --- |
| **System Context Object** | **Properties** |
| System Logs | User activity, errors, carpool or parking events |

## **2.4 Relationship Types**

### **2.4.1 Entities Relationship Model**

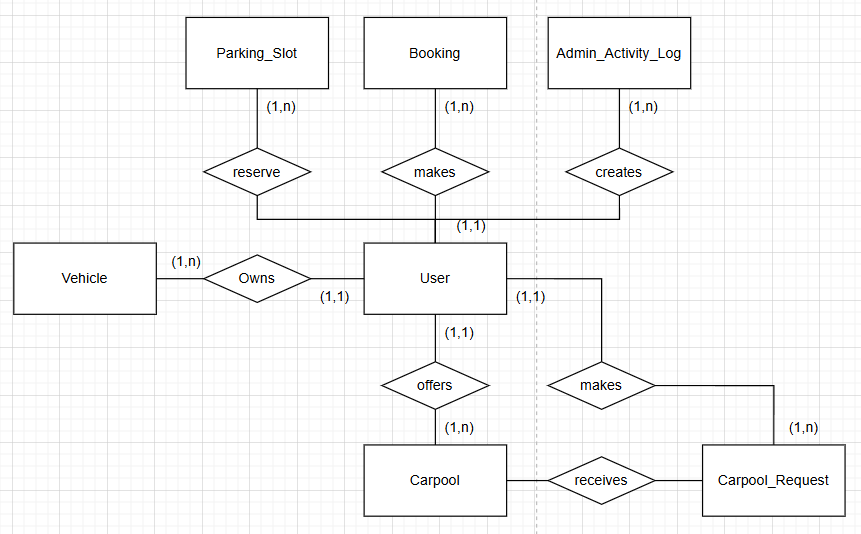


Figure 2.4.1 Entities Relationship Model Diagram

### **2.4.2 Relationship Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Relationship** | **From** | **To** | **Description** |
| makes | User | Booking | User makes a Booking. |
| creates | User | Admin\_Activity\_Log | Admin creates activity log. |
| reserves | User | Parking\_Slot | User reserves a parking slot. |
| Owns | User | Vehicle | User owns a vehicle. |
| offers | User | carpool | User offers a carpool. |
| makes | User | Carpool\_Request | User makes a carpool request. |
| receives | Carpool | Carpool\_Request | Carpool receives a carpool request. |

## **2.5 Roles and Responsibilities**

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| System Analyst | Define structure and context guidelines |
| Backend Developer | Implement database and context logic |
| Frontend Developer | Show real-time context |
| QA Tester | Validate accuracy of context data |
| Admin | Manage parking slots and carpool. |
| Technical support | Maintain the system performance |
| User | Keep their profile and vehicle info up to date |

# **3.0 Development Context**

**Development Plan A**

|  |  |
| --- | --- |
| **Budget** | $120,000 |
| **Max development time** | 2 months |
| **Development method** | Scrum (Agile method) |

|  |  |
| --- | --- |
| **Available personal** | **Quantity** |
| Solution Architect | 1 |
| Quality Assurance Engineer | 1 |
| Requirement Engineer | 1 |
| Software Engineer | 2 |

**Development Plan B**

|  |  |
| --- | --- |
| **Budget** | $150,000 |
| **Max development time** | 2 months |
| **Development method** | Scrum (Agile method) |

|  |  |
| --- | --- |
| **Available personal** | **Quantity** |
| Solution Architect | 1 |
| Quality Assurance Engineer | 1 |
| Requirement Engineer | 1 |
| Software Engineer | 1 |
| DevOps Engineer | 1 |
| Technical Support | 1 |
| UX Designer | 1 |

**Development Plan C**

|  |  |
| --- | --- |
| **Budget** | $130,000 |
| **Max development time** | 2 months |
| **Development method** | Scrum (Agile method) |

|  |  |
| --- | --- |
| **Available personal** | **Quantity** |
| Software Architect | 2 |
| Quality Assurance Engineer | 1 |
| Requirement Engineer | 1 |
| Software Engineer | 1 |
| DevOps Engineer | 1 |
| Technical Support | 1 |

# **4.0 Requirement Resource**

## **4.1 Potential Relevant Requirement Source Identification**

### **4.1.1 Checklist for Carpool and Parking System**

|  |  |
| --- | --- |
| **Usage facets** | * Maintenance staff * Manage profile * System monitoring * Reserve parking * Carpooling System * Map management * Manage payment * Role-based control |
| **Subject facets** | * User * Carpool * Admin * Student * Parking * Carpool * Payment * Department * Faculty * Vehicle * Car * Motor * Bicycle * Admin\_Activity\_Log * Parking\_Slot * Booking * Carpool\_Request |
| **IT System facet** | * Authentication * Data encryption * Non-relational Database * Relational Database * Frontend framework * Backend framework * Cross platform * Microservice * API * Google Map (3rd Party) * Web Portal |
| **Development Context** | * Solution Architect * Quality Assurance Engineer * Requirement Engineer * Software Engineer * Software Developer * Technical Support * Network Engineer * Database Administrator * Data Engineer * Web Designer * Software Architect * Mobile application Engineer * DevOps Engineer * UX Designer |
| **Requirement Engineering Context** | * Campus Accessibility * Transportation Policy Makers * Campus IT Department * Legal and Compliance Officers * Campus Security and Safety Department * University Administration |
| **Stakeholders** | * Algorithm Experts * Maintenance Staff * Lawyer |
| **Documents** | * System manual * User interface description * Documents of carpool and parking system |
| **Systems** | * Campus Database System |

**4.2 Requirement Source Selection**

### **4.2.2 Selection of Checklist for Carpool and Parking System**

|  |  |
| --- | --- |
| **Usage facets** | * Maintenance staff * Manage profile * System monitoring * Reserve parking * Carpooling System * Map management * Manage payment * Role-based control |
| **Subject facets** | * User * Carpool * Vehicle * Admin\_Activity\_Log * Parking\_Slot * Booking * Carpool\_Request |
| **IT System facet** | * Authentication * Data encryption * Relational Database * Frontend framework * Backend framework * Microservice * API * Web Portal |
| **Development Context** | * Solution Architect * Quality Assurance Engineer * Requirement Engineer * Software Engineer * Software Developer * Technical Support * UX Designer |
| **Requirement Engineering Context** | * Campus Accessibility * Transportation Policy Makers * Campus IT Department * Legal and Compliance Officers * Campus Security and Safety Department * University Administration |
| **Stakeholders** | * Maintenance Staff * Lawyer |
| **Documents** | * System manual * User interface description * Documents of carpool and parking system |
| **Systems** | * Campus Database System |

### **4.2.3 Selection of Development Plan for Carpool and Parking System**

**Development Plan B**

|  |  |
| --- | --- |
| **Budget** | $150,000 |
| **Max development time** | 2 months |
| **Development method** | Scrum (Agile method) |

|  |  |
| --- | --- |
| **Available personal** | **Quantity** |
| Solution Architect | 1 |
| Quality Assurance Engineer | 1 |
| Requirement Engineer | 1 |
| Software Engineer | 1 |
| DevOps Engineer | 1 |
| Technical Support | 1 |
| UX Designer | 1 |

### **4.2.4** **100 Dollars Test**

|  |  |
| --- | --- |
| **Carpool Context ($40)** | Campus users’ surveys |
| Analysis of carpool platforms |
| Campus transportation policies |
| **Parking Integration Context ($35)** | Parking management system documentation |
| Campus parking regulations and policies |
| Integration API specifications |
| **Security & Compliance Context ($25)** | University IT security requirements |
| Data privacy regulations |
| Financial transaction compliance standards |

### **4.2.5 Adequate Representation of Stakeholders Selection**

**End Users**

|  |  |
| --- | --- |
| **Students / Staff** | Main platform users for both carpooling and reserve parking |
| **Admin** | Monitoring user activity, errors, carpool or parking events |

**Development Team**

|  |  |
| --- | --- |
| **Solution Architect** | Design system and making decision of system architecture |
| **Requirements Engineer** | Gather, analyse, and document requirements of system |
| **Software Engineer** | Develop system features and functionality |
| **Quality Assurance Engineer** | Test system quality and functionality  Find the hidden bug of system |
| **DevOps Engineer** | Deployment, infrastructure, and CI/CD |
| **Technical Support** | Post-launch user support and maintenance |
| **UX Designer** | Design simple and good user experience interface |

**Tertiary Users**

|  |  |
| --- | --- |
| **Lawyer** | Ensure the system is against to the IT laws |

**Business Stakeholders**

|  |  |
| --- | --- |
| **Campus Security** | Safety policy and compliance oversight |
| **University Administration** | Policy approval and governance |
| **Campus IT Department** | Infrastructure and security requirements |

### **4.2.6 Development Team Requirements and Concerns**

**Solution Architecture**

|  |  |
| --- | --- |
| **Scalability Requirements** | Handle peak usage |
| **System Integration Constraints** | Integrate with existing campus systems |
| **Technology Stack Decisions** | Choose the technologies fit campus IT standards |
| **Security Architecture** | Provide a workflow to protect users and sensitive data |
| **Performance Requirements** | Handle response times and concurrent users |

**Requirements Engineer**

|  |  |
| --- | --- |
| **Stakeholder Access** | Availability of students, staff for surveys |
| **Documentation Standards** | Determine the preferred requirement formats |
| **Change Management** | The way of requirements will evolve during development |
| **Traceability Needs** | Linking requirements to features and tests |
| **Validation Processes** | Verify requirements with stakeholders |

**Software Engineer**

|  |  |
| --- | --- |
| **Development Environment** | System development infrastructure |
| **Coding Standards** | Software development guidelines |
| **API Specifications** | Integration with parking and payment systems |
| **Database Access** | Users’ data access permissions and limitations |
| **Testing Environment** | Separate environments for development and testing |

**DevOps Engineer**

|  |  |
| --- | --- |
| **Deployment Infrastructure** | System development infrastructure |
| **Monitoring Tools** | System health and performance monitoring |
| **Backup Procedures** | Data protection and disaster recovery |
| **Release Processes** | The ways of updates will be deployed |
| **Security Compliance** | Campus network security requirements |

**Technical Support**

|  |  |
| --- | --- |
| **Support Processes** | The ways of users will report issues |
| **Documentation Needs** | User manuals and troubleshooting guides |
| **Escalation Procedures** | The timing of involve campus IT or security |
| **Training Materials** | The guidelines of educate users on platform features |
| **Issue Tracking** | Integration with campus carpool and parking systems |

**UX Designer**

|  |  |
| --- | --- |
| **Component Reusability** | Use reusable UI components where possible. |
| **Error Handling UX** | Include user-friendly error states and recovery options |
| **Navigation Structure** | Design a clear and scalable navigation model for users with different roles |
| **Performance Awareness** | Minimize use of heavy visuals or complex animations. |
| **User Roles Clarity** | Clearly differentiate experiences and UI for different user roles |

### **4.2.7 Selection Rationale**

**Most Critical Sources**

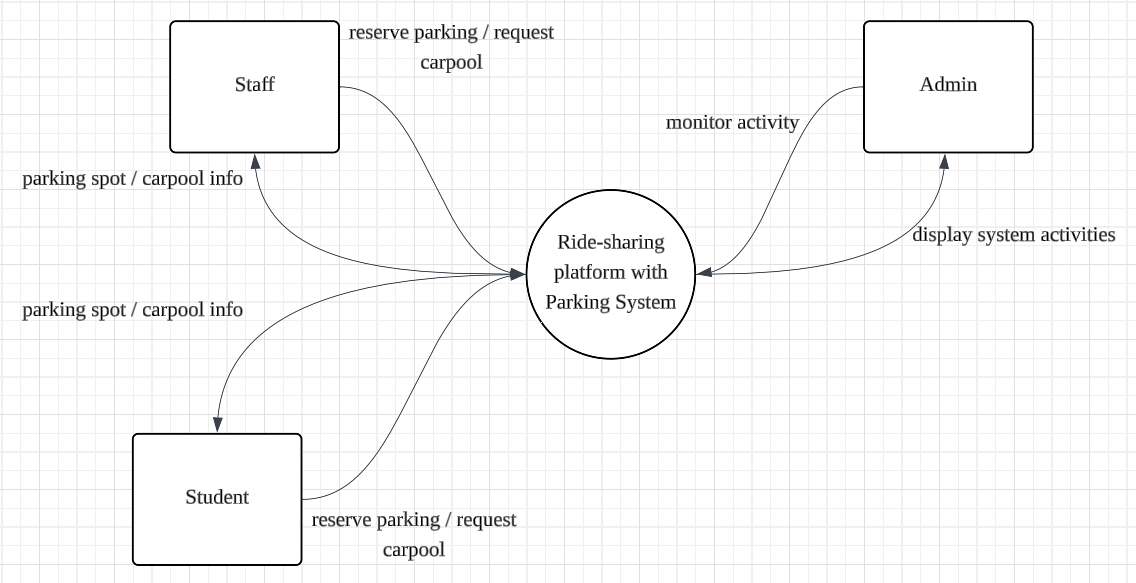
|  |  |
| --- | --- |
| **Student focus groups** | Direct users with highest frequency of use |
| **Carpool and Parking system documentation** | Critical integration component |
| **Campus security policies** | Non-negotiable safety requirements |

# **5.0 Data Flow Diagram**

This diagram illustrates the data flow between external actors and the Campus Ride-Sharing Platform with Parking System focusing on ride booking and parking reservation functionality. It provides a high-level view of the system's interactions and core data processes.

## **5.1 Ride Sharing platform with Parking System**

**Level 0**



**Level Description**

|  |  |
| --- | --- |
| **Level** | **Description** |
| Level 0 | Entire system as one process, with external entities such as student, staff, admin |

**Elements Description**

**External Entities**

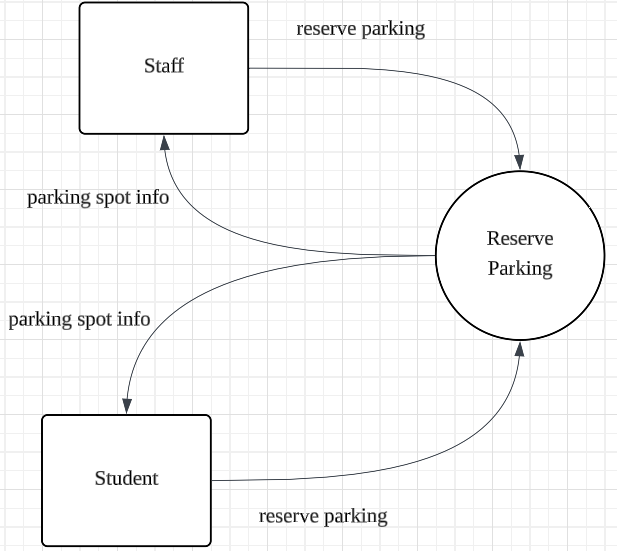
|  |  |
| --- | --- |
| **Entities** | **Description** |
| Staff | A staff using the system to request carpool or reserve parking. |
| Student | A student using the system to request carpool or reserve parking. |
| Admin | An admin using the system to review system activities. |

**Process**

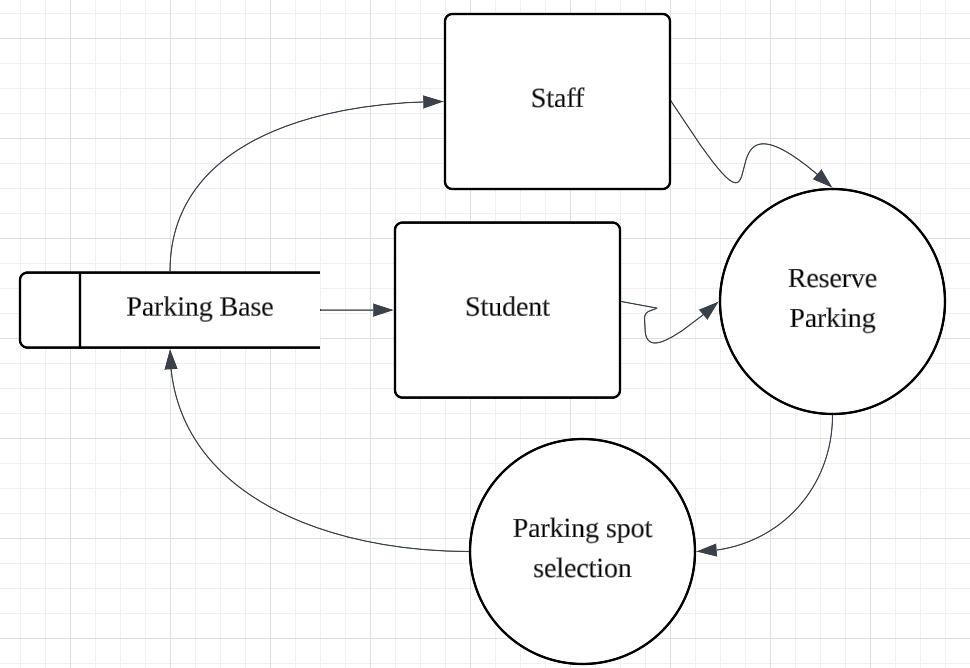
|  |  |
| --- | --- |
| **Processes** | **Description** |
| Ride Sharing platform with Parking System | Handles real-time parking reservations and carpool system. |

## **5.2 Reserve Parking**

**Level 0**



**Level 1**



**Level Description**

|  |  |
| --- | --- |
| **Level** | **Description** |
| Level 0 | Entire reserve parking as one process, with external entities |
| Level 1 | Reserve parking major internal processes and data stores |

**Elements Description**

**External Entities**

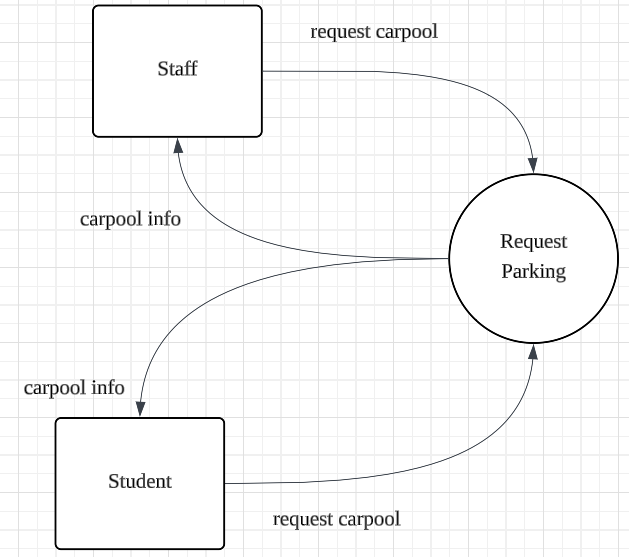
|  |  |
| --- | --- |
| **Entities** | **Description** |
| Staff | A staff using the system to request carpool or reserve parking. |
| Student | A student using the system to request carpool or reserve parking. |

**Process**

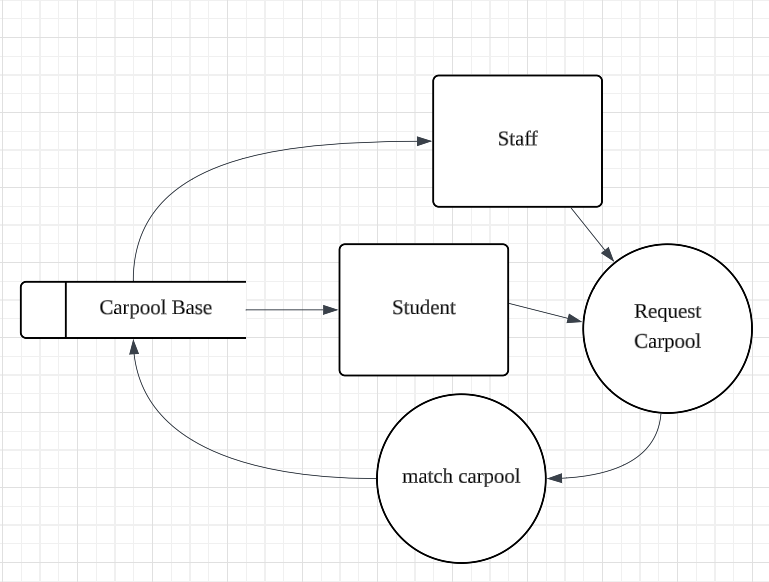
|  |  |
| --- | --- |
| **Processes** | **Description** |
| Reserve parking | Output the parking data with status available to parking selection process |
| Parking selection | Allows the user to view and select from available parking options received from the Reserve Parking process. |

## **5.3 Request Carpool**

**Level 0**



**Level 1**



**Level Description**

|  |  |
| --- | --- |
| **Level** | **Description** |
| Level 0 | Entire request carpool as one process, with external entities |
| Level 1 | Reserve parking major internal processes and data stores |

**Elements Description**

**External Entities**

|  |  |
| --- | --- |
| **Entities** | **Description** |
| Staff | A staff using the system to request carpool or reserve parking. |
| Student | A student using the system to request carpool or reserve parking. |

**Process**

|  |  |
| --- | --- |
| **Processes** | **Description** |
| Request carpool | Captures user input including pickup location, destination, time. |
| Match carpool | Find suitable matches based on criteria such as route similarity, timing, seat availability, and user preferences. |