

Team 2 (Parking Garage)

1. **Adriane San Gaspar**
  - Student ID: 028648163
  - CSULB Email: adriane.sangaspar012@student.csulb.edu
2. **Daphne Rios**
  - Student ID: 031013643
  - CSULB Email: daphne.rios01@student.csulb.edu
3. **Flavien Jean Guy Aurelien Maameri**
  - Student ID:
  - CSULB Email:
4. **Mia Alvarez**
  - Student ID: 030261697
  - CSULB Email: mia.alvarez03@student.csulb.edu
5. **Nicolas Piker**
  - Student ID: 029966545
  - CSULB Email: nicolas.piker01@student.csulb.edu
6. **Porter Clevidence**
  - Student ID: 029238207
  - CSULB Email: porter.clevidence01@student.csulb.edu
7. **Abram Ibarra**
  - Student ID: 031652736
  - CSULB Email: abram.ibarra01@student.cuslb.edu

## Preface

Version	Date	Changes
1.0	3/15/2025	Initial Version

## Purpose

This document serves as a comprehensive guide for the development and understanding of the software project titled "Pick&Park".

## Audience

The intended audience of this document includes project stakeholders, developers, testers, project manager, parking lot owners/operators, and end users(drivers/parkers).

## Introduction

Pick&Park is a software system that helps users conveniently find and reserve parking spots in advance, reducing the time spent searching for parking. The system provides real-time availability updates, secure reservations, and user-friendly payment options.

## Project Overview

"Pick&Park" is a mobile based application that simplifies the process of finding and reserving parking spots. It secures and streamlines parking space tracking, reservation, management, and reporting.

## Project Goals

- Enhance User Convenience.
  - Provide a seamless user experience for finding and reserving parking spots with minimal effort.
- Improve Parking Efficiency.
  - Reduce traffic congestion caused by drivers searching for parking.
- Ensure Secure Transactions.
  - Implement robust security measures for payment processing.
  - Offer trusted third-party payment options to enhance user confidence.

## Glossary

- **Web-based:** Refers to applications that are accessed and used via a web browser, rather than being installed locally on a computer or device.
- **Inventory Management System:** A system used by businesses to track inventory levels, orders, sales, and deliveries. It helps manage the supply of goods, products, or materials.
- **Streamlines:** To make a process more efficient by eliminating unnecessary steps or tasks, often through automation or simplification.
- **Order Management:** The process of receiving, tracking, and fulfilling customer orders. It involves inventory updates, order status tracking, and shipping arrangements.

## Team 2 (Parking Garage)

- **Parking Spot:** The physical location where one vehicle is allowed to park.
- **User:** The person who uses the app to find and reserve parking.
- **Reservation:** A parking spot that is currently in use, with a vehicle parked there.
- **Parking Garage:** A building (or section of building) containing one or more floors dedicated to parking.
- **Parking Level:** a floor in a parking garage (can be the only floor available).
- **Spot Inventory:** The stock of parking spots available for reservation.
- **API:** Application Programming Interface.

## User Requirements and Use Cases

(Outline what the system must do from the user's perspective. User stories need to use the format discussed in class and on our slides. Use cases provide detailed scenarios of system interactions.)

## User Stories

(A collection of user stories that apply to the project.)

1. As a registered parker, I want to log in securely so that I can easily and unquestionably reserve a parking spot.
2. As a user, I want to search for available parking spots so that I can easily find a place to park.
3. As a user, I want to reserve a parking spot so that I am guaranteed a space until arrival.
4. As a parking lot owner, I want to automatically update the availability of parking spots so that users see up to date information.
5. As a user, I want to extend my parking time remotely so that I don't get fined.
6. As a user, I want to view my reservation history so that I can keep track of my parking expenses.
7. As a parking lot owner, I want user payment methods to be valid so that I receive proper compensation in a timely fashion.
8. As a parking lot owner, I want to have a traffic log with timestamps so I can analyze the times of the day my lot is in higher demand.
9. As a user, I want to see parking lot rates while browsing locations so I can make an informed decision on where to park.
10. As a user, I want to navigate to a parking lot I've reserved so I can conveniently get to my desired parking spot.
11. As a parking garage owner, I need to maintain spot inventory so that parking spots shown on the app match their availability in real life.
12. As a sales manager, I want to generate sales reports by date range so that the upper management can be kept up to date on the parking garage's performance & profitability.

## Use Case: Log in

Identifier	UC-1 Login
Goal	Allow the user to securely log in to the system to access the features of the application
Requirements	The user must have a registered account; the system must validate user credentials securely; the system should provide appropriate error messages for failed login attempts
Initiating Actor	User
Participating Actor(s)	Authentication Service Provider, Database System
Pre-conditions	The user must have valid credentials(e.g. username and password)
Post-conditions	The user is successfully authenticated and redirected to the dashboard or homescreen; if authentication fails the user receives an appropriate error message and remains on the login screen
Included Use Case(s)	Password recovery
Extension(s)	Incorrect Credentials: User receives a message prompting them to retry or reset their password. System Downtime: User is notified that login services are currently unavailable.

Table 1: Typical Course of Action

Seq#	Actor's Action	System's Response
1	User opens the app	
2		System display the login screen
3	User enters their username and password	
		System validates the entered credentials
		System grants access and redirects the user to the home screen

Table 2: Alternate Course of Action

Seq#	Actor's Action	System's Response
1	User opens the app	
2		System display the login screen
3	User press forgot their password button	
4		System redirects the user to the password recovery page
5	User completes the password reset process	
6		System sends a password reset confirmation and prompts the user to log in again
7	User successfully logs in after resetting password	
8		System grants access and redirects the user to the home screen

Table 3: Exceptional Course of Action

Seq#	Actor's Action	System's Response
1	User opens the app	
2		System display the login screen
3	User enters their username and password	
4		System experiences a timeout during the login process
5		System display a timeout error to the user

## Team 2 (Parking Garage)

### Use Case: Reserve a Parking Spot

Identifier	UC-2 Reserve
Goal	Reserve a physical parking space
Requirements	Know the layout of the parking garage
Initiating Actor	User
Participating Actor(s)	Parking Lot Owners, Third party payment vendors
Pre-conditions	Locations for the app turned on, User is logged in, Another User not parked in the spot
Post-conditions	User has a physical parking space reserved in the app
Included Use Case(s)	UC-1 Login, Report, Payment method
Extension(s)	N/A

**Table 1: Typical Course of Action**

Seq#	Actor's Action	System's Response
1	User opens the app	
2		Display parking garages on a map
3	User selects a parking garage	
4		System displays the bottom level of the parking garage
5	User cycles through parking garage levels	
6	User selects a parking space	
7		System asks the user to input their payment information to pay
8	User submits their payment information	
9		System asks user if System can save their payment information for later
10		System confirms the reservation
11	User review their reservation in the reservation page	
12		System confirms that the transaction was a success
13		System displayed the reservation of the User

**Table 2: Alternate Course of Action**

Seq#	Actor's Action	System's Response
1	User opens app, reserves parking spot, and submits their payment information	
2		System checks information
3		System determines submitted payment information is unacceptable
4		System informs User that the payment information was incorrect
5	User submits correct payment information	
6		System checks the submitted payment information
7		System determines submitted information was correct
8		System proceeds to display reservation to User

**Table 3: Exceptional Course of Action**

Seq#	Actor's Action	System's Response
1	User opens the app	
2		System attempts to display garages to map
3		System fails to display garages to map
		System displays an error message

Use Case: Template”copy this when adding a new use case”

Identifier	[E.g., “UC-1 Book Trip”]
Goal	[The initiating actor’s goal]
Requirements	[Requirements that are addressed by this use case]
Initiating Actor	[Actor who initiates interaction with the system]
Participating Actor(s)	[Other actors who participate, if any]
Pre-conditions	[The state of the system before the interaction can start]
Post-conditions	[What must be true about the system after the goal is achieved or abandoned]
Included Use Case(s)	[Use cases that are included by this use case, if any]
Extension(s)	[Extensions of this use case, if any]

*Table 1: Typical Course of Action*

Seq#	Actor’s Action	System’s Response
1		
2		
3		

*Table 2: Alternate Course of Action*

Seq#	Actor’s Action	System’s Response
1		
2		
3		

*Table 3: Exceptional Course of Action*

Seq#	Actor’s Action	System’s Response
1		
2		
3		

## System Architecture

(Describe the high-level design of the software.)

Components

Deployment Diagram