# SC2006 TransitEase

SDAD Group 7
Ashwin, Dave, Jonathan, Jun Heng

# Table of contents



# 01

# Introduction

# About TransitEase

## Aim

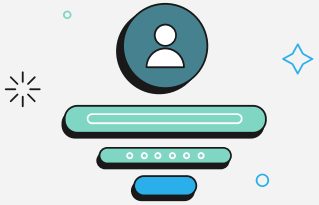
- Reduce urban congestion, fuel waste and emissions
- Smoother mobility and reducing environmental impacts

## Vision

TransitEase simplifies finding available parking spaces in real time, helping users save time and reduce city congestion.

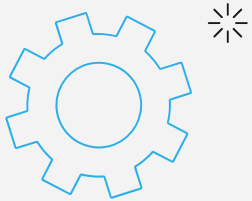
# Solution Overview

TransitEase empowers users with instant access to parking availability, supporting better parking decisions in real-time



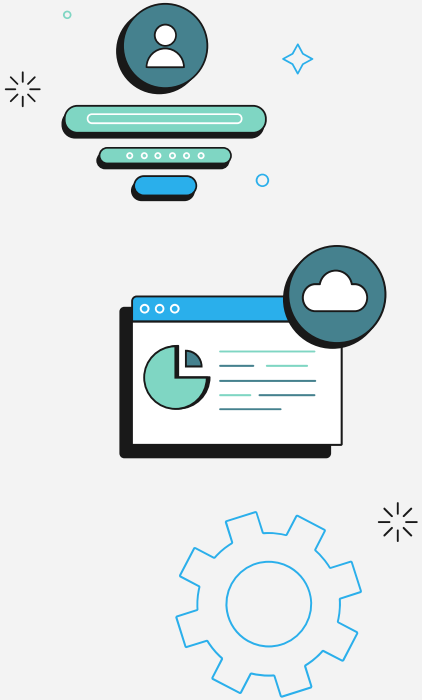
## Unique Selling Points

- Real-time parking availability.
- Car and motorcycle lot indicators.
- User-friendly map and list views for easy navigation.



# Key Features

- Live Data on Available Lots: Displays updated car and motorcycle lot availability.
- Intuitive UI: Clean, easy-to-navigate map and list views.
- Personalization: Ability to filter based on user vehicle preferences.



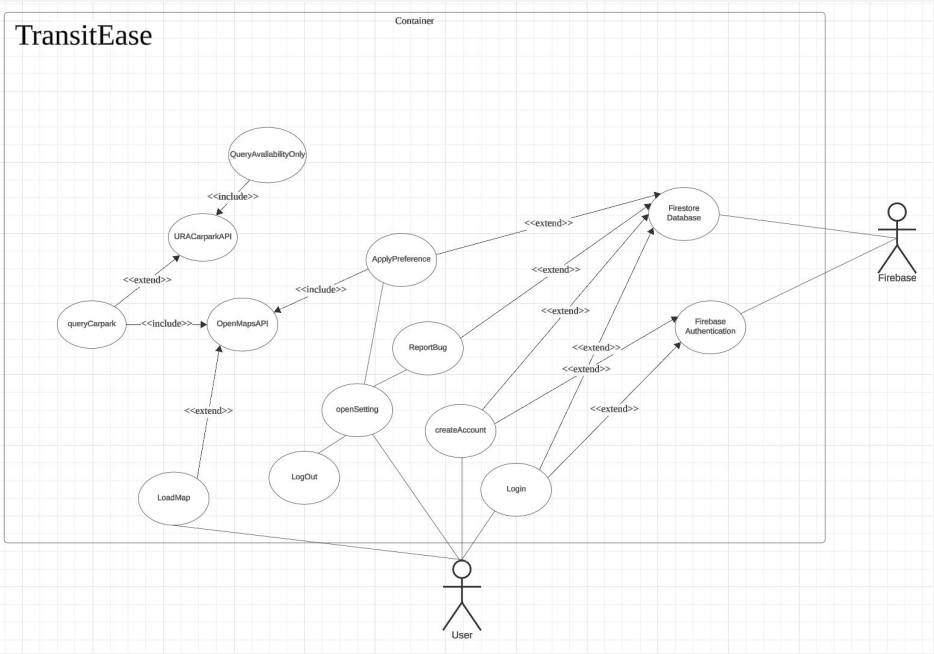
# Technical Overview

- - Firebase Integration: For real-time data storage and authentication.
- - Geolocation Services: For displaying car parks relative to user's location.
- - Flutter Framework: Cross-platform compatibility.
- - Firestore Database: Robust backend to handle data scaling.
-

# 02

# Use Case Diagram

# Use Case Diagram



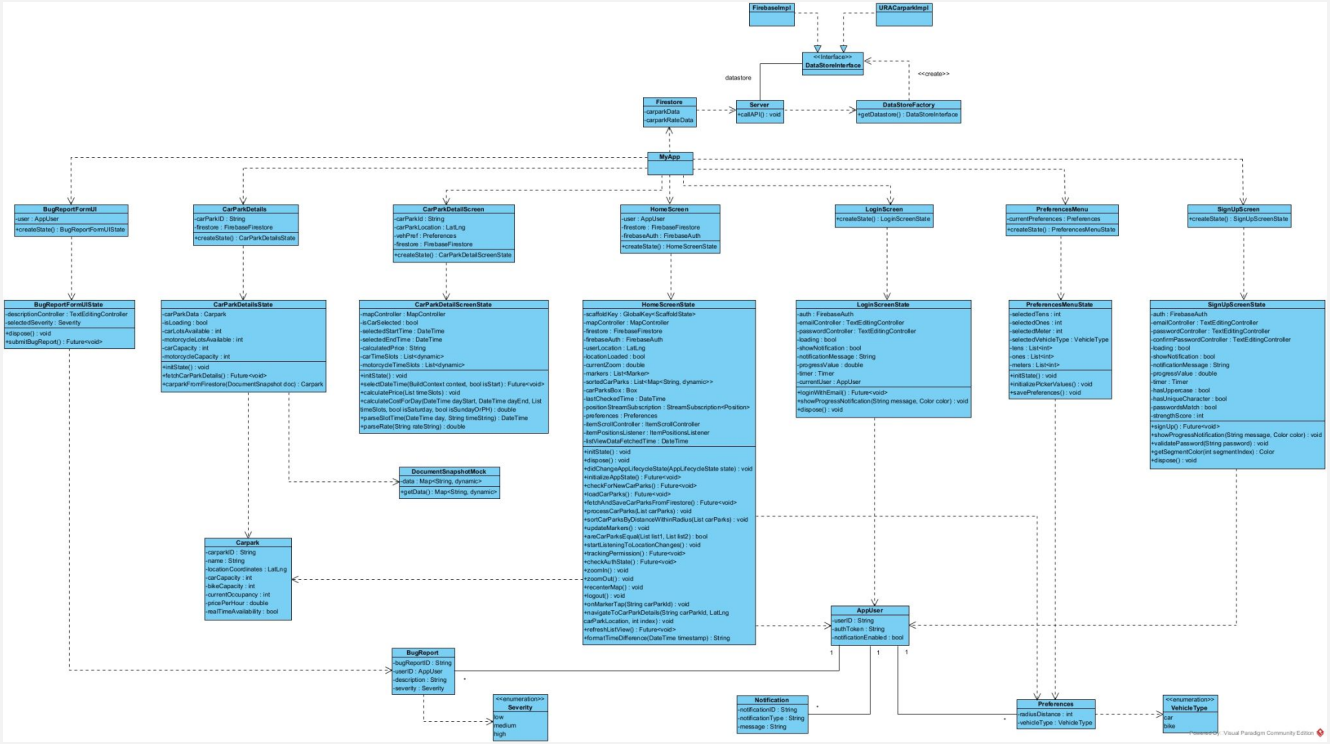

## Overview

- Interaction between users, API and Firebase for managing parking solutions
- Query carparks through APIs
- Firebase provides backend support to handle account creation, login, bug reporting and settings
- Dependencies and extensions indicated by <<include>> and <<exclude>> relationship

# 03

# Class Diagram

# Class Diagram



# Class Diagram

### Core

The app integrates Firebase and URA Carpark API for data storage and real-time carpark data retrieval, respectively.

### Entity Classes

Essential entity classes include Carpark, BugReport, AppUser and Preferences.

### Enumerations

The VehicleType and Severity enumerations allow users to customise their Preferences and BugReports, respectively.

### Main Classes

There are main screens for Login, SignUp, Home, CarParkDetails, BugReport and Preferences, each with a corresponding state class for managing user interactions.

### Controllers

The app uses controllers to handle user inputs and data updates.

### Setup

The app is designed to have streamlined functionality for user authentication, parking data management and personalized preferences.

# 04

# Core Technologies

# Core Technology

- Docker For Back-End Computing and Data Updates
- FireStore & Firebase
  - FireBaseAuth: Login
  - FireStore: Data Collection Containing Carpark Data
- Flutter: Mobile Application

# Challenges

- Government URA API
    a. Requires Constant Renewal of token every **24 hours**
    b. Output of request of car parks and available lots are usually **ALL** car parks given as a response.
    c. Coordinates of all car parks returned are in SVY21 rather than the common WGS84
    d. Requires heavy computation of data
- Firebase
    a. Limited Read and Write to database with free tier.
    b. Application cannot keep Reading and Writing from database.

# Solutions

- Implemented a Docker with Python and cronjobs as external API controller.
  - Advantages
    - Allows for changes should there be any updates on API policy
    - Allows for any other implementation in the future without disrupting front end.
- Implemented Geocaching
  - This is a technique to allow conversion of Longitude and Latitude to a hash
  - Allowing for creation of a hashmap for lookup
- Flutter
  - Stores the hashmap containing all the carparksIDs and their respective hashes.
  - This allows only specific carparks data to be pulled greatly reducing number of reads and writes.

# 05

# Application Demo

# 06

# Conclusion

# Conclusion

- Users will be able to find car parks more easily -> Reduce congestion on commonly used carparks.
- Users will be able to find car parks that suit their needs with ease
- Facilitate smoother mobility and reducing environmental impacts by cutting down on fuel wasted on finding car parks.

# Thank You!