**1. Introduction**

Our ABC HQ offices are facing a challenge with available parking spaces. ABC Currently has 4 parking areas: 

* 2 outside parking areas who are available for everybody
* 2 inside areas, spots here are assigned to people who :
* Are with the company more than 15 years
* Are ranked vice president or higher in the company hierarchy.

There are currently not enough spaces outside

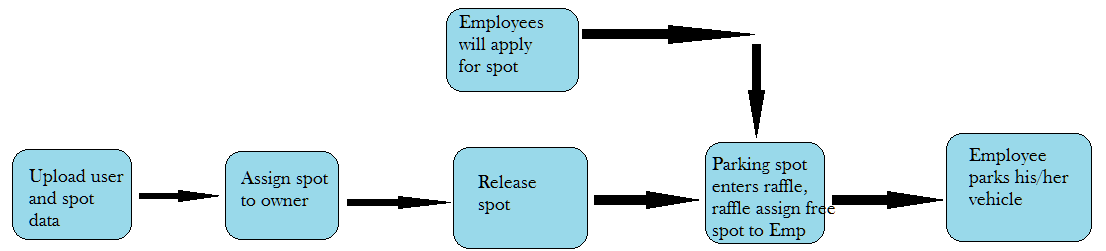
We want to solve this problem by creating a mobile application. Main idea is that parking spots can be marked available by a spot owner and that a lottery runs every day assigning free spots to people who have signed up for one. 

**2. Application overview**

Description

The parking app should maximize parking usage at ABC HQ. The goal of the application is minimizing the number of empty spots in the garage whereby more spots are available to people without reserved spots.

High level process:



Application overview

* Employee data like Emp id, name, desg, years of experience… etc can be uploaded using CSV.
* Spot details like id, block, wing, building, location… etc can be hard coded in DB.
* Spot can be assigned to an employee (generally its admin functionality)
* Spot owner should be able to free up their parking spot (single or multiple days at once)
* The parking spots that were freed up are being put into the raffle
* The user should be able to indicate the days he wishes to receive a parking spot
* Raffle should assign free spots to employees who applied for it.
* The user should be able to see the spot they won for today.
* The user should be able to see the spot they won for tomorrow.

**Technologies:**

* Use spring batch for uploading the data from csv to DB.
* Follow the microservices architecture.
* Use the Eureka for registering all services.
* Use Feign client for service to service communication.
* Use the Ribbon for load balance.
* Use gateway for single entry point.
* Follow the all java standards. (Naming conventions, package structure … etc)
* Use proper exception handling.
* Always send the proper data with status code and status message.
* Implement the swagger for all services.
* Use proper DB schema for each service. Don’t use single schema for all services.
* Follow the relationships between the tables.
* Follow the REST API standards.