Parking System application

Mahmoud

Contents

About developer

Developed by Mahmoud Sayed

FCAI-CU student

Mail: mahmoudsayed1332002@gmail.come

Document Purpose and Audience

This document explains the working details of the Parking Garage application.

Through the system, it is possible to configure it, enter customer information, and choose a place to park the vehicle, know the extent of its parking, calculate the cost and know the total income.

It is expected to be read by the owner of the garage

Introduction

Software Purpose

Facilitating the process of parking vehicles in the garage by make it automated and satisfying the customer for the speed of work of the system and reducing paperwork

Software scope

Providing information about the garage to the garage owner, such as income from it, and accelerating the process of parking the vehicle in the garage

Process model

I have applied scrum process model for development

The system development is spread into 3 sprints each sprint take maximum 3 days

Sprint1

In the first sprint we well develop part of the system –parking in process –

Sprint goal: customer can park in

Day1:

Understanding the system work

Understanding the purpose and scope

Writing functional requirements

Writing use case diagram for the system

Writing use case tables for parking in process

Day2:

Rewrite use case table and use case diagram

Design temporary class diagram for the this part of the system

Apply some of OOP and SOLID principles on the class diagram

Implement some classes –classes has a clear job-

Day3:

Rewrite use case table

Quick view on OOP and SOLID principles on the class diagram

Update implemented classes

Writing all classes

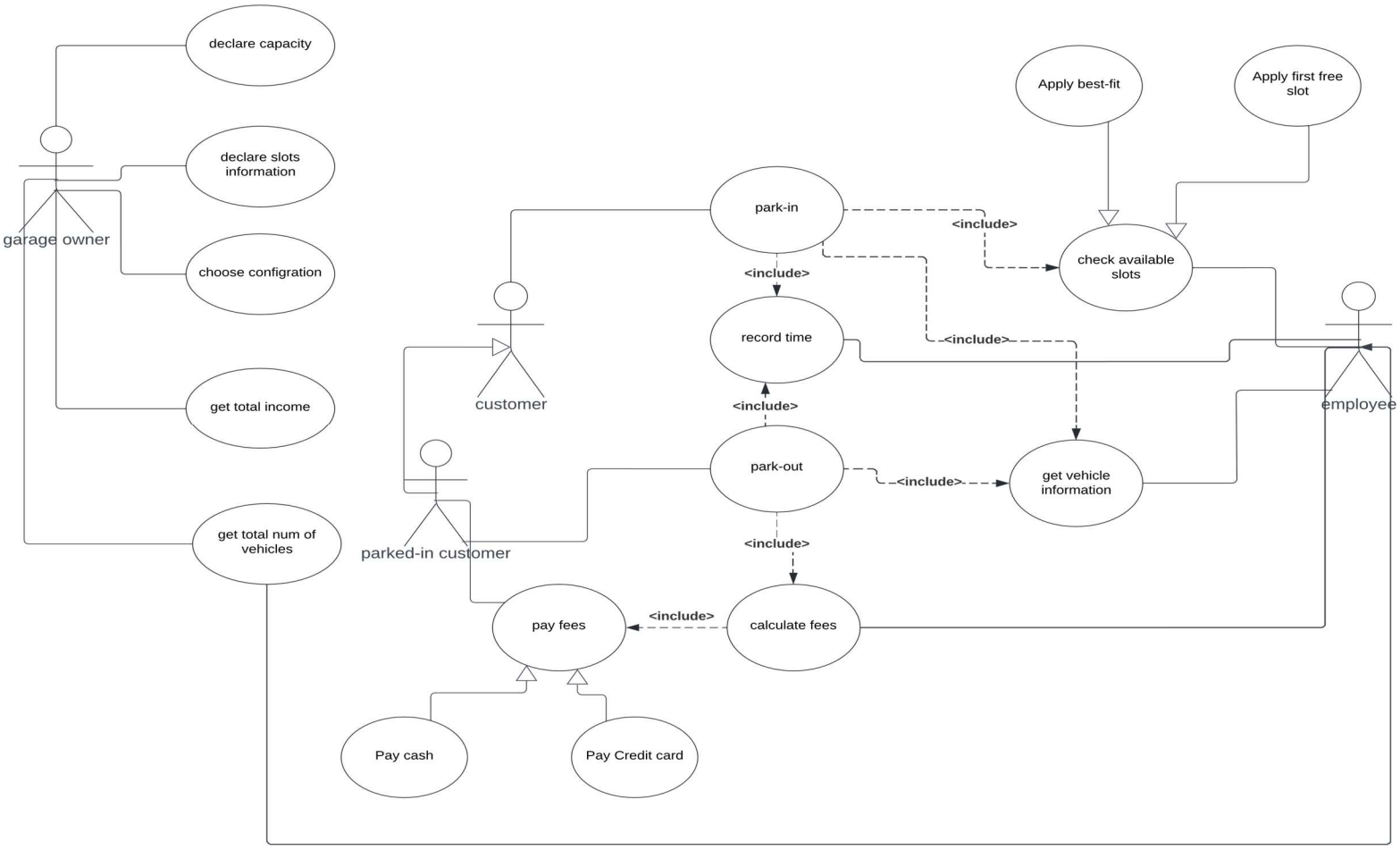
Writing the MianTest class

Black box testing -for the park-in part-

Unit testing for the implemented classes

Use case diagram

This diagram is a use case diagram for the whole system


declare capacity
declare slots information
choose configration
get total income
get total num of vehicles
garage owner
Apply best-fit
Apply first free slot
park-in
<include>
check available slots
record time
customer
employee
park-out
get vehicle information
parked-in customer
pay fees
calculate fees
Pay cash
Pay Credit card

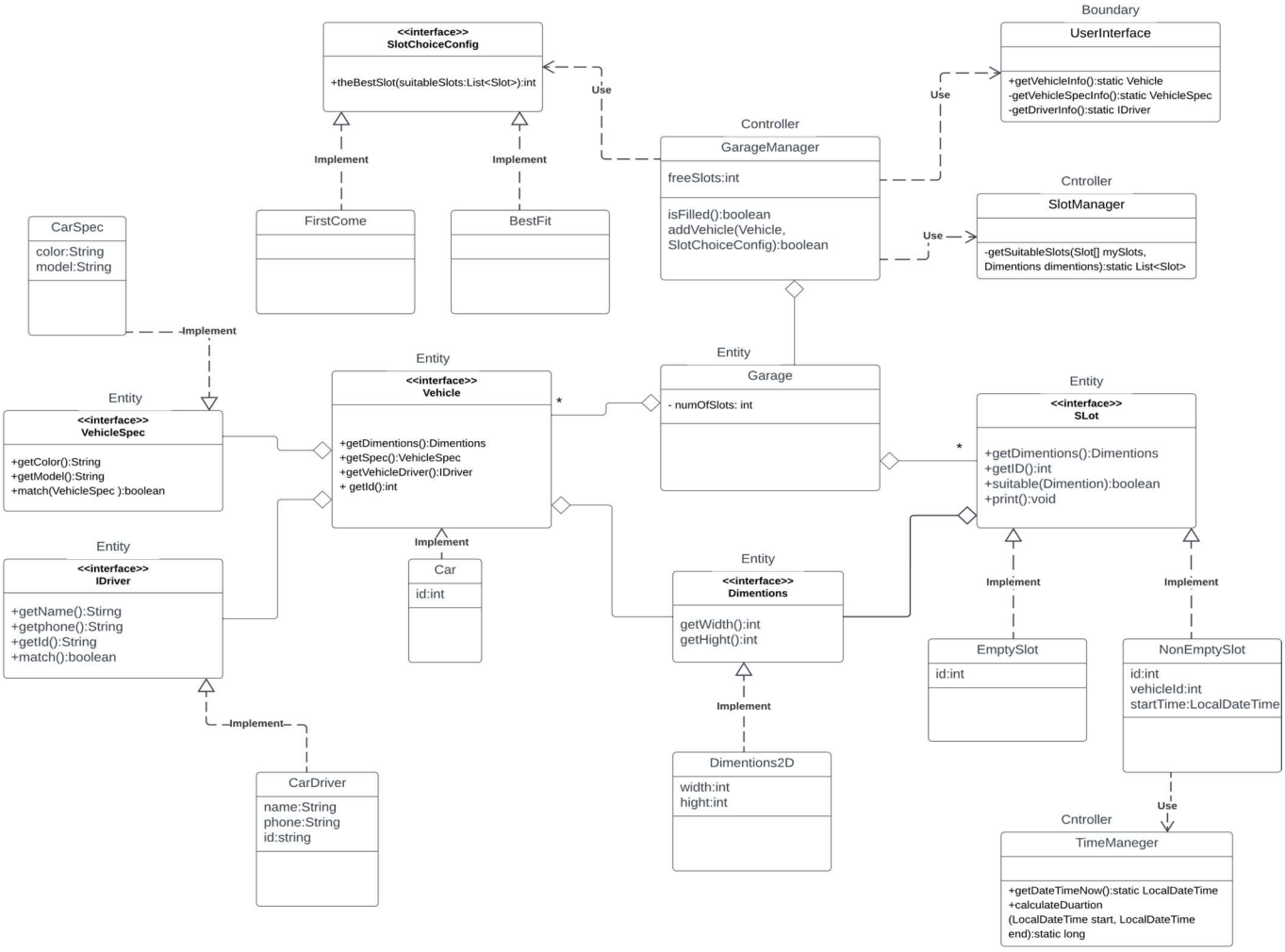
Use case tables

Use case table for park-in use case that we want it for this sprint

Use Case ID:		
Use Case Name:	Park-in	
Actors:	Customer, Employee	
Pre-conditions:	Garage owner declared the garage capacity Garage owner declared the slots info Garage owner chose a configuration for slot picking	
Post-conditions:	The customer parked	
Flow of events:	**User Action**	**System Action**
	1- customer enter the garage to park the vehicle 2-employee takes and enter the vehicle info	
		3- system get empty list of slots in the garage 4-system check valid slots 5-system display a suitable slot
	6-employee tell customer for his slot 7-customer go to park in	
		8- System capture the arrival time automatically 9-increment total number of vehicles
Exceptions:	1- employee checks available slots	
		2- no available slots
Includes:	check available slots get vehicle information record time	

Class diagram

This diagram is the class diagram for the first sprint



Implementation

The whole implementation was uploaded in

Sprint2

In the second sprint we well develop part of the system –parking out and payment processes –

Sprint goal: customer can park out and pay the fees

Day1:

Quick look for system requirements

Quick look for the use case diagram

Writing use case tables for parking out and payment processes

Design a temporary class diagram for this part and merge it with the first part

Apply OOP and SOLID principles on the class diagram

Implement some classes –classes has a clear job-

Day2:

Quick look for use case table and class diagram

Implement the remain classes

Update the old classes that should to be changed to deal with the update

Update MainTest class for to deal with park out process

Black box testing – for the park out part –

Unit testing for the implemented classes

Use case table

Class diagram

Sprint3

Working on