Gameday Parking Payment System

# Team Members

1. Sergio Carrera
2. Judith Kwassieu

# Introduction

This project is a deluxe parking payment system.

# System Requirements

The system is to be allowed to be programmed by someone when it arrives to the parking area. A technician sets up the parking rates for sedans, SUVs and/or trucks. A manager then sets up the capacity for each type of vehicle the technician is allowing. The system then can be used by people to select their vehicle type and allow them to pay the parking fee to allow them to park as long as there is capacity left for their vehicle type.

# System Design

The system uses polling and is aided by functions. Our design starts off with the initialization mode which is where the prices of certain types of vehicles are set. After the vehicles have had their rates set, the board displays which type/s of vehicles are set and the system is then allowed to go into preparation mode. In this mode, the capacities of certain types of vehicles are set. After setting the capacities for the vehicles, the board displays the capacities remaining for each type of vehicle. The system is now in payment mode. The customer chooses their vehicle type then the system briefly displays the parking rate to them for their vehicle. It goes back to the capacities remaining but is now blinking, indicating they can start inserting bills. A function we created for this part determines what type of bill is being inserted depending on the combination of keys that are pressed. The system displays change after they are done if they are owed any or goes back to vehicle type selection if they insert less than the rate or if it’s been thirty seconds after they inserted the last bill. If capacity for a customer’s vehicle type is zero and they try to select it, the system displays a zero. The system will display “- - -“ if garage has no remaining capacity. At any point, the system can be taken back to initialization mode to change the rates or to preparation mode to change capacities. For the project, we considered using interrupts, but we felt more comfortable taking the polling approach. The difficulty of the project was probably a 3. At first it seemed overwhelming with the amount of information about the deluxe system, but once we really got down what the project required, programming it wasn’t too difficult. Difficulty in programming came from small bugs that would pop up as we were writing the code and testing it.

# Technical Challenges

The five most difficult challenges encountered during this project were:

1. Getting the alarm to work correctly, it would continually go off every thirty seconds after being triggered once.
2. Displaying the capacities left when there was no more space for trucks and/or SUVs was difficult. For example, if there was no space left for trucks, and 2 spaces left for SUVs and 2 spaces left for sedans, our system would just display “2 2” when it should display “- 2 2”.
3. After leasing out all parking spaces, our system displays “- - -“ indicating there is no more space left. After doing a second initialization, the display would still display “- - -“ instead of clearing until the new capacity was set which was only a minor bug.
4. Having the system figure out which bill was being inserted was quite difficult. Certain bills require buttons to be pressed simultaneously, but due to human error, this is pretty much impossible, so we had to figure out how to deal with this.
5. The system has very specific steps to setting it up. This made us make a lot of flag variables to keep track of what had been done when setting up the system to make sure it works as intended. It was difficult to keep track of so many flag variables in the conditions when we were writing the code.

# Metalevel Discussion

Total hours your team spent on the project (hours, not person-hours): We estimate we spent a total of around 10 hours over the past couple weeks, coming up with the design, writing the code, and dealing with bugs as we tested.

## Five suggestions on how to improve the project:

1. This would’ve been a lot easier if we could insert bills with the switches instead of using only 3 keys, maybe change that in the future but we understand it was done the way it is for learning purposes.
2. This is minor but if the project could’ve been opened earlier that would’ve been great because we had finished lab 6 the day it came out and if the project was available to start the day lab 6 became available, that would’ve given us even more time to work on it.
3. We think dividing it into a base system and a deluxe system is unnecessary if we don’t get the full points for doing the base system and some people may do their base system in such a way that then makes it more difficult to expand to the deluxe system. The assignment would maybe be more straight forward if all it was about was the deluxe system and just the instructions on the requirements for it.
4. Possibly give more options for extra credit on the project that involve just extending the system a bit further instead of the scrolling display or the sound extra credit.
5. Maybe give a demonstration in class of what a finished system is supposed to look like and act like to allow us to visually see what we have to make because parts of the instructions didn’t mention certain things or weren’t clear about certain things which the TAs would then tell us how those things were meant to be after we had programmed things a different incorrect way.

## Comments on the suitability of this project for CS/SE majors (why is it suitable or not suitable?):

We think this project is very suitable for both CS/SE majors. It involved a little bit of a lot of things that go into making a program and it seems like the system we made for this project could almost really be used in the real world with a few more additions done to it so it gives us a taste of what programming is really like for the real world.