

SOFTWARE ENGINEERING LAB

Project Assignment 1

January 15 2018

Professor In-Charge

K. Chandrasekaran

Nihal Haneef 16CO128

Gurupungav Narayanan 16CO114

Assignment 1

Software Engineering Lab

Members

1. Nihal Haneef
 - 16CO128
 - 9008818469
 - nihalh55@gmail.com
2. Gurupungav Narayanan
 - 16CO114
 - 9008791776
 - gurupungavn@gmail.com

Problem Title

The Parking Assignment Problem

Abstract

In a world where time is valued more than ever; where appointments need to happen *on time* and *deadlines need to be met*, finding a parking slot for your car only wastes time and fuel. The search for a parking place is a task which can consume a lot of time and affect the efficiency of economic activities, social interactions, and the health of citizens. Searching for a parking space is unnecessary contributor to pollution, since drivers need to keep their car running to find an open space.

An ideal parking space also needs to be as close as possible to the driver's destination, while conserving fuel and time as much as possible. An absence of sufficient parking lots results in drivers using undesignated areas for parking, resulting in traffic blocks and congestion.

Our proposed solution is a software that would automatically detect open slots, and would assign the most ideal free slot to a driver. The software also needs to be able to detect and adapt to various layouts, and be able to ensure that the driver only parks at the designated spot.

Brief Work Envisaged for Proposed Problem

- Choose an appropriate life cycle model for the proposed problem.
- Learn the software tools to be used to implement the proposed solution.
- Schedule work and set appropriate deadlines.

Part 1

Software Crisis

Case 1 - Failure at Customer Site

- **Problem 1** : Scanner used to detect a car fails and returns wrong outputs.
Solution : Error detection program that detects erratic/constant output.
- **Problem 2** : The driver parks at a spot that was not allocated to him/her.
Solution : Optical Character Recognition to detect the number plate of the car present. Fines should be provided as necessary and the software should automatically adapt to the unexpected change.

Case 2 - Delay in Development Time

Plausible Reasons:

- Unorganized work and bad planning
- Planned timeline is not followed
- Inadequate technical skill to solve problems and resolve bugs

Mitigation:

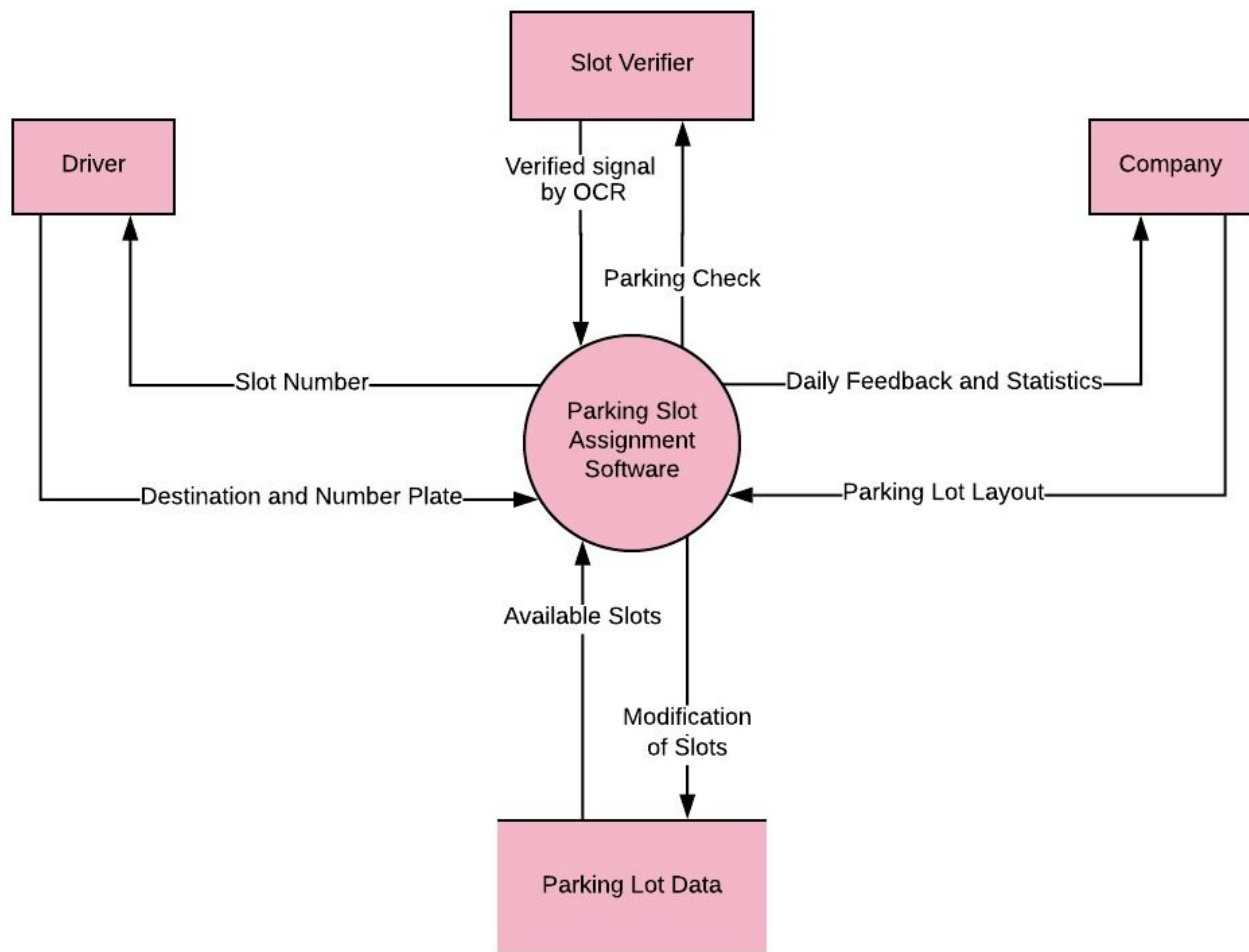
- Adapting and following an apt software development cycle model for the problem would help organize and plan the project
- Setting realistic deadlines for given work and working to meet that deadline
- Preemptively predicting problems and gain adequate knowledge to resolve them

Efforts taken to Convince Customer:

- Explain to the customer that the software is actually beneficial and would result in easier management.
- Softwares generally replace manual work by humans. Explain to them that the software would bring increased efficiency at lower costs.
- Show them functional prototypes of the software, if possible, to help them understand how much work has been done and what the software is trying to bring to the table.

Part 2

Context Diagram, Data Flow Diagram & Entity Relationship Diagram

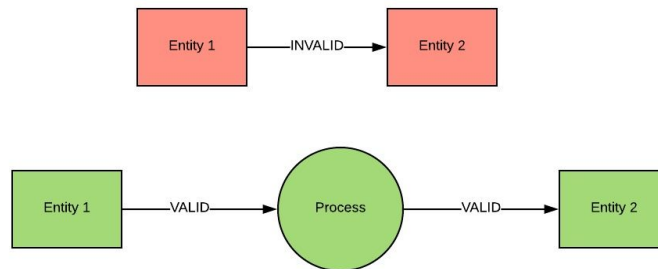


Context Diagram

Common Data Flow Diagram Mistakes

- **Illegal Data flow**

Data flow cannot take place between two entities without processing it.

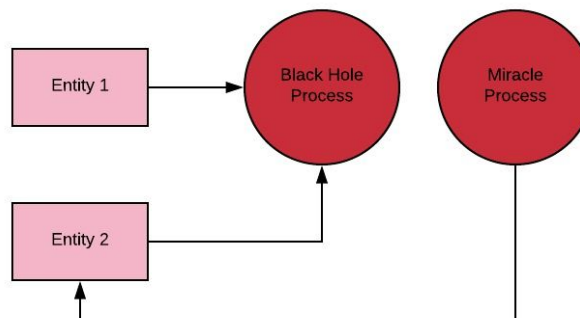


- **Black Hole**

A component takes in inputs but doesn't provide with any outputs.

- **Miracle**

A component provides outputs but doesn't take any inputs.



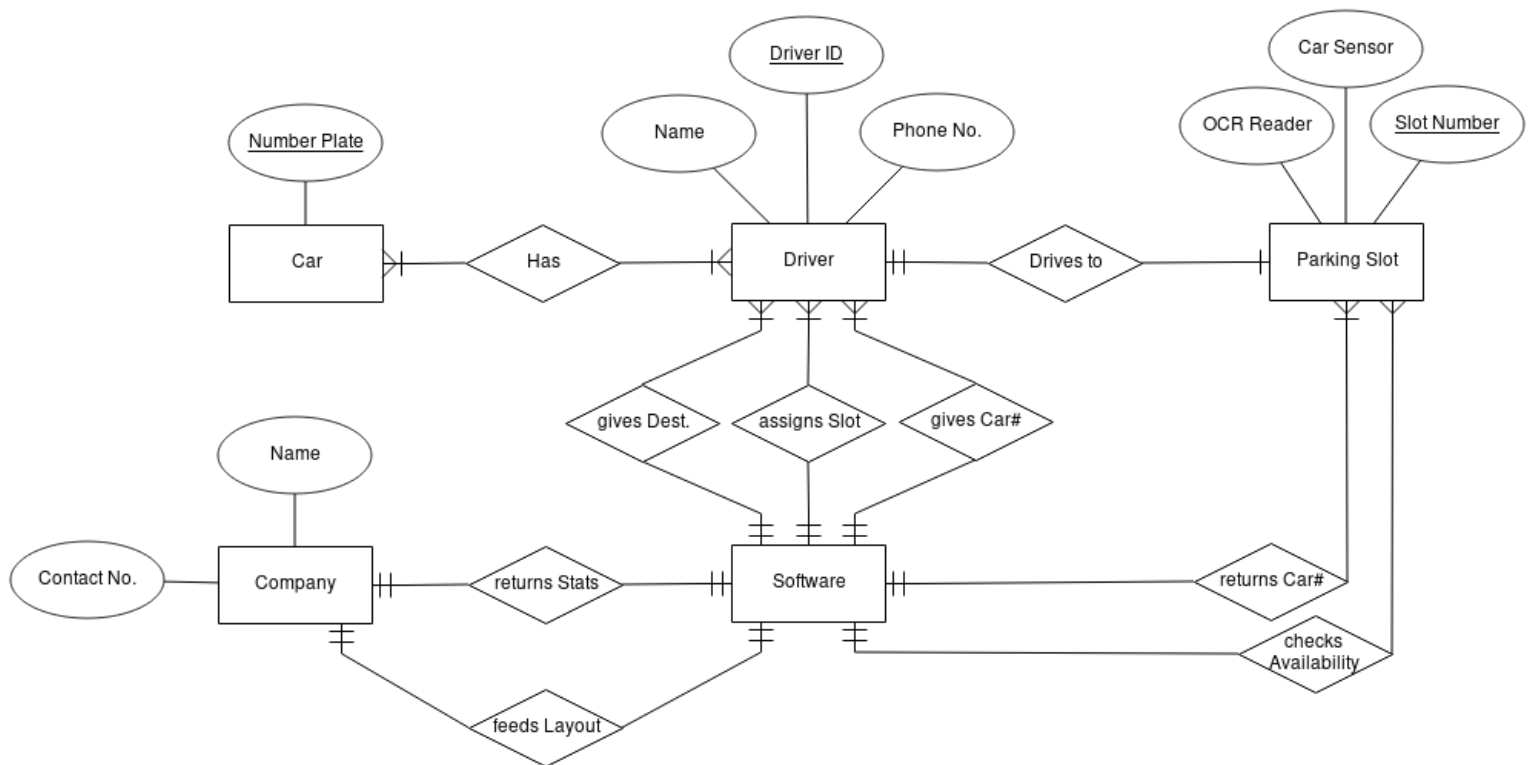
- **Lack of Understanding of DFD**

DFD is not the same as flowchart.

- **Wrong diagramming schemes**

No labelling of data flow lines, wrong usage of shapes.

Entity Relationship Diagram



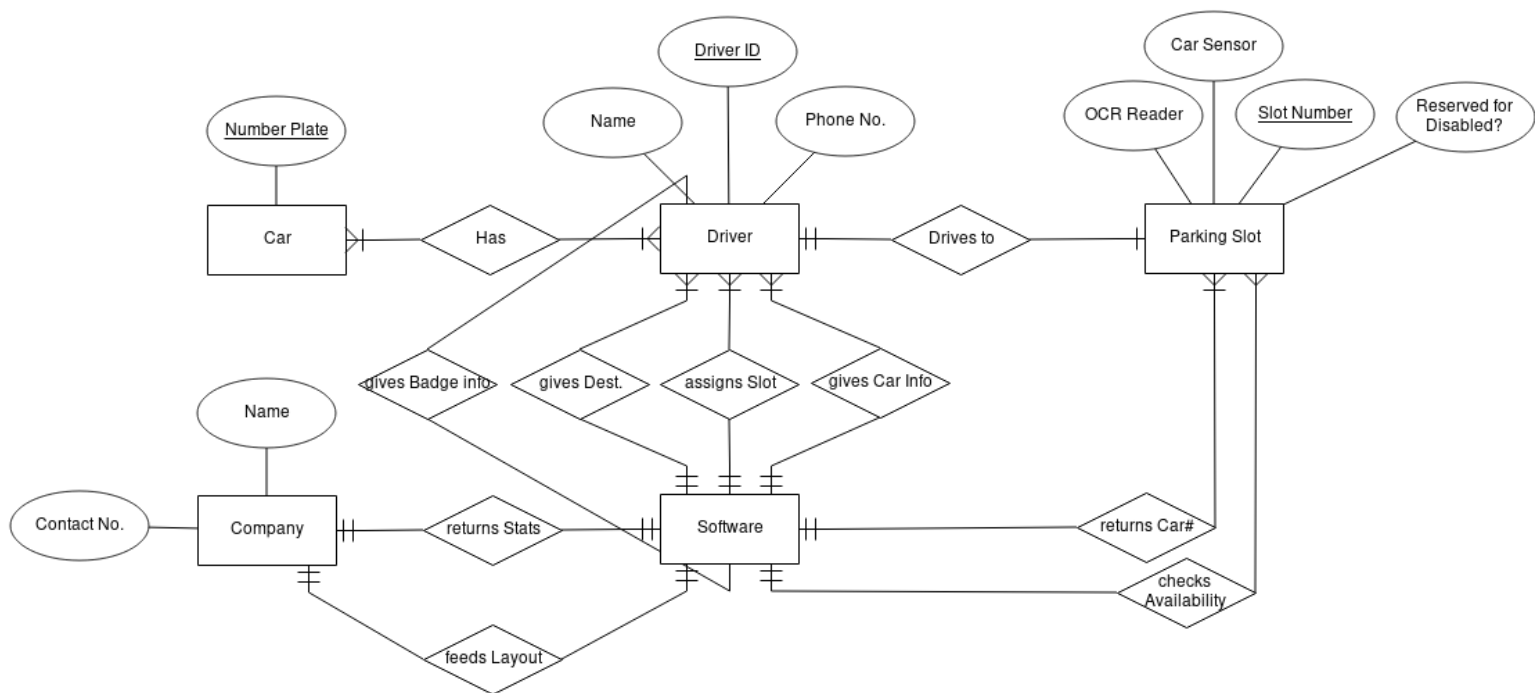
Possible Changes in Requirements

- 1) Provision for reserved parking lots for the disabled
- 2) The cars need to be parked according to some special characteristic of the car.
Example: Color, Type etc

The changes in the ERD for these cases is shown in the following pages.

Reserved Slots for the Disabled

In this scenario, the driver would need to tell whether he is a disabled parking badge-holder. That would be another relationship between the driver and the software. The Parking Slot would now have another attribute that would characterize it as a spot reserved for the disabled.



Allotting slots based on characteristic

We are taking the example of color here. Say, Blue cars must only be parked in a certain section of the layout. In such a scenario, the driver would either need to provide the color of his car or we need some kind of color detection mechanism. Either way, in the eyes of the ERD, this information is something provided by the driver. The Parking Slot must have another attribute that would say what colored car must be parked there.

