Parking Assignment Problem

Software Engineering Lab - Implementation Report 1

Build 1

March 16th 2018

Professor In-Charge
K. Chandrasekaran

Nihal Haneef	16CO128
Gurupungav Narayanan	16CO114

Basic Information

The implementation of the Parking Slot Assignment System was coded in ***Java*** version 1.8. The IDEs that were used are ***IntelliJ and Android Studio*** which is also IntelliJ based. They were used to aid us with the proper organisation of our code and other resources and helped us in coding our program by providing a very user friendly environment to do so. The builder used to build and compile our code was ***Gradle*** which was very useful in resolving dependencies during compilation and build. Apart from the general input out libraries, ***Java Swing*** was utilized in the making of the GUI for the java application by providing various functionalities like the ability to incorporate headings, scroll panes, dialog boxes etc. ***Github*** was used for collaboration purposes.

The system consists of three components at present. Firstly we have the ***console application*** which is an android application which should be installed and run on an android device. It is responsible for getting driver(user) input and it will communicate with the second component , which is the ***main Java application*** via networking. The Java application will be run on the computer system either a laptop or a desktop and provides administrator functionalities ,i.e, it helps the client(owner) to overview the working of the software and his layout. The third component is the **SOU (Slot OCR Unit)** which is also an android application which is to be installed on an android device. It is responsible for reading number plate information and communicates with the main java application via networking.

First run the main java application. After setting up the system by inputting the initial layout, servers will be created for communication purposes with the console and SOU apps, so an active network connection is needed between all three components. The two android applications should be installed on two different devices. After these steps the system will be up and running.

Functional Requirements - Implemented

Following are the functional requirements that were implemented. Refer to the SRS document for more details on the functional requirements.

FR-ID	NAME	DESCRIPTION
1.1	Console - Destinations View	The console app on start shows a list of the available destinations to the user in the form of a list view
1.2	Console - Select Destination	The console app provides an interface to select a destination from the given list of destinations on touch
1.3	SOU - Input Number Plate Picture	The SOU app captures an image of the number plate
1.4	SOU - Return number plate value	The SOU app uses OCR and converts the image into a string containing the number plate information
1.5	Java App - Calculate ideal slot	The main java application calculates the optimal slot from the destination ID and input layout provided.
2.1	Java App - Display Layout	The main java application displays the layout with an interactive interface to see details of each slot
3.2	Manage Layout	Provided the add layout feature

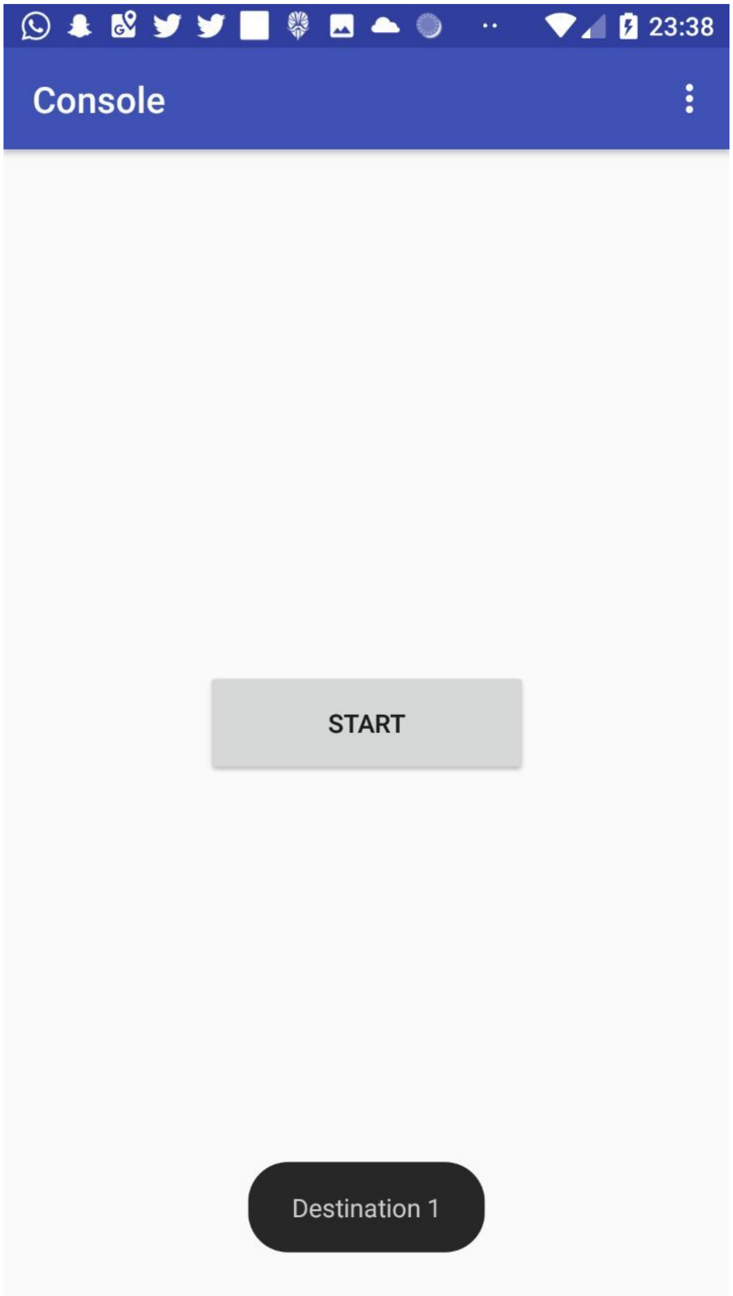
Functional Requirements - Not Implemented

Following are the functional requirements that were not implemented. Refer to the SRS document for more details on the functional requirements.

FR-ID	NAME	REASONS FOR NOT IMPLEMENTING IT
2.2	Display List of offenses	Time constraints lead to the deprioritization of this functional requirement.
2.3	Resolve Offenses	Time constraints lead to the deprioritization of this functional requirement. Didn't really show proof of concept.
3.1	Log in	Didn't really show proof of concept leading to it's deprioritization.
3.3	List Broken SOU units	Time constraints lead to the deprioritization of this functional requirement. Didn't really show proof of concept.
3.4	Display statistics	Need a lot of data for proper utilization of this functionality hence leading to its deprioritization.
1.6	Return Ideal Slot	Inexperience from developers side

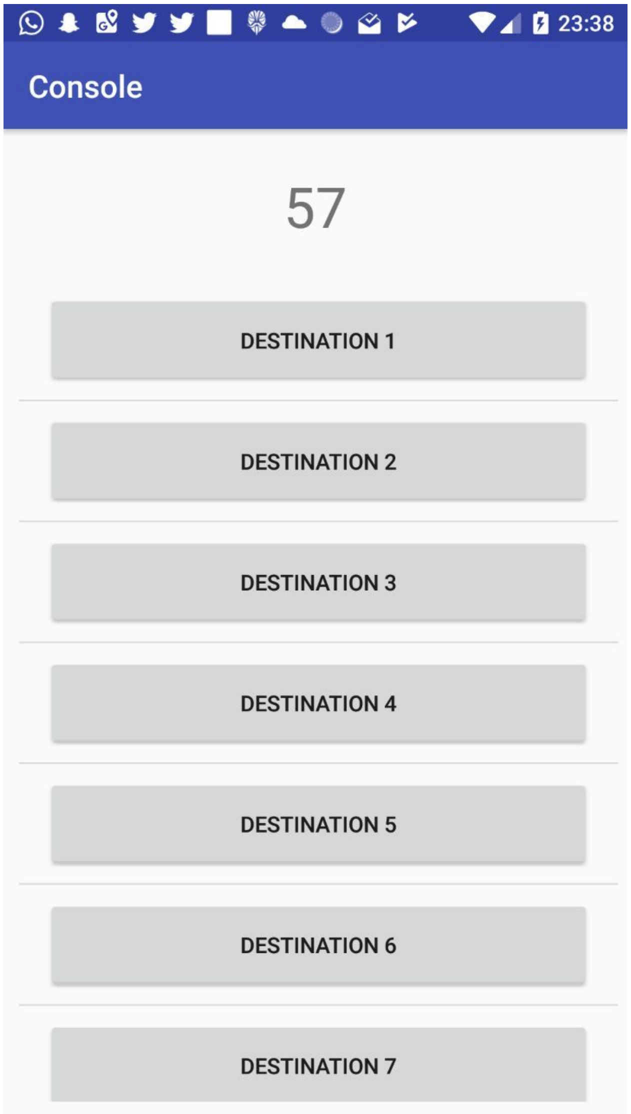
Screen Shots

1. FR 1.1



Destination 1

2. FR 1.2


23:38
Console
57
DESTINATION 1
DESTINATION 2
DESTINATION 3
DESTINATION 4
DESTINATION 5
DESTINATION 6
DESTINATION 7

3. FR 1.3 and 1.4

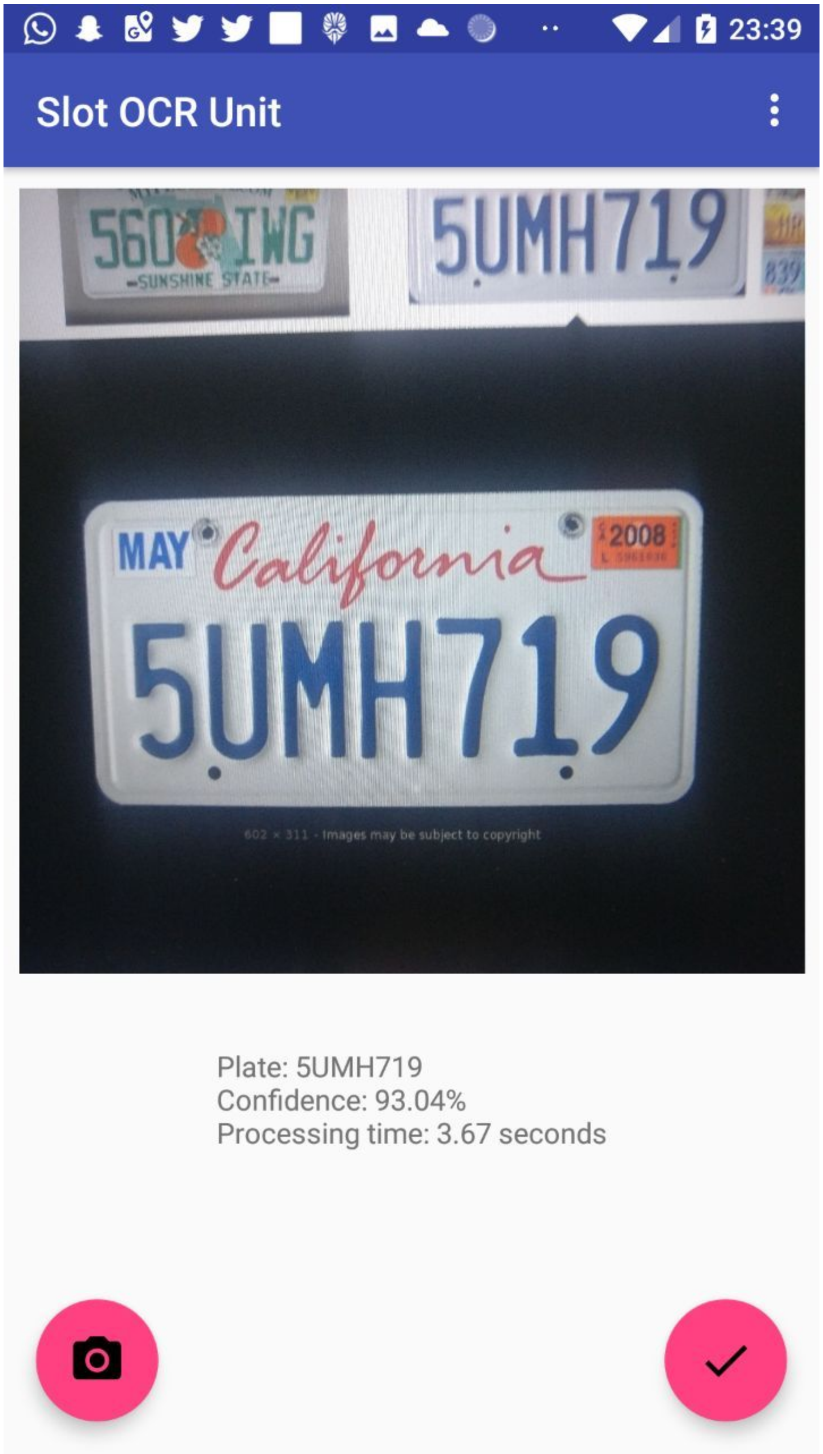
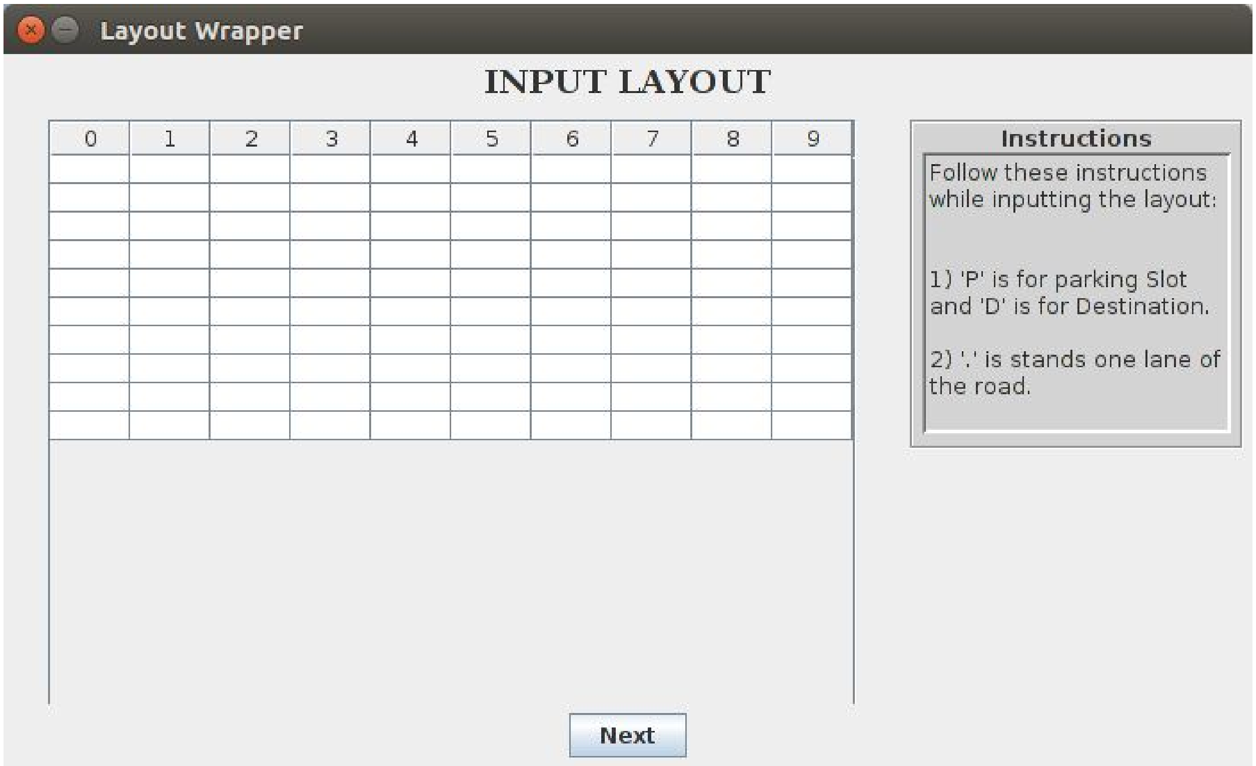


Plate: 5UMH719
Confidence: 93.04%
Processing time: 3.67 seconds

4. FR 3.2



INPUT LAYOUT

0	1	2	3	4	5	6	7	8	9

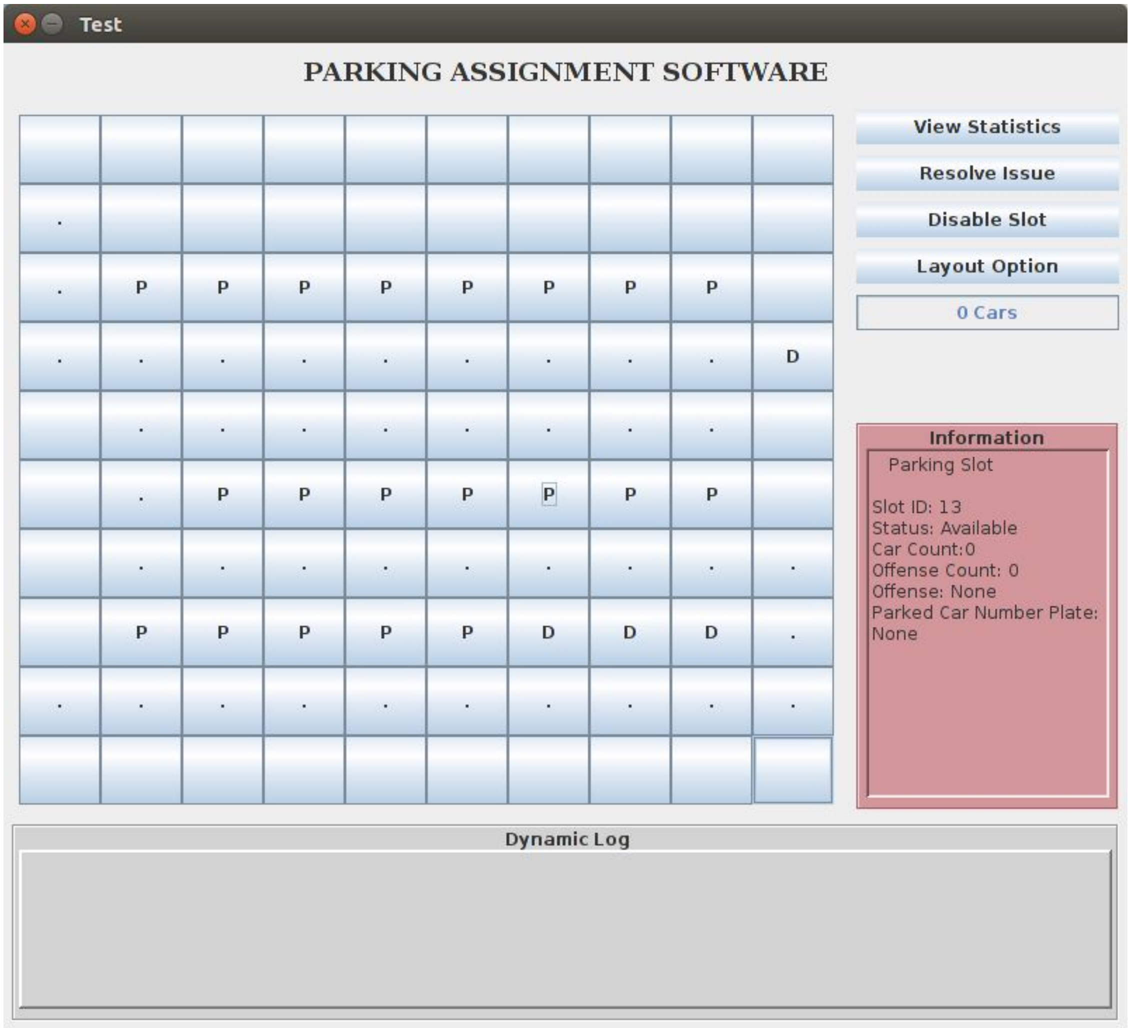
Instructions

Follow these instructions while inputting the layout:

1) 'P' is for parking Slot and 'D' is for Destination.

2) '.' is stands one lane of the road.

5. FR 2.1



PARKING ASSIGNMENT SOFTWARE

.									
.	P	P	P	P	P	P	P	P	
.	.	.	.	.	.	.	.	.	D
	.	.	.	.	.	.	.	.	
	.	P	P	P	P	P	P	P	
	.	.	.	.	.	.	.	.	.
	P	P	P	P	P	D	D	D	.
.	.	.	.	.	.	.	.	.	.

Information

Parking Slot

Slot ID: 13
Status: Available
Car Count:0
Offense Count: 0
Offense: None
Parked Car Number Plate:
None

Plan for Next Build

The main area of focus for the next build will be to make the main java application be more user friendly and functional for the client in terms of administration of the layout. Furthermore attempt will be made to start developing the guard application which will help in resolving offenses that take place at the parking layout.

More time will be spent on making feasible deadlines and scheduling work amongst team-mates in an orderly and formal fashion.

The current build just showed a prototype of the software which would seem to be very ill made. This is mainly because of time constraints and inexperience with the tools and languages used from the developer's side. These problems will not be faced nor tolerated for the next build.

Summary

The build 1 for the software has been successfully created. It includes the console app, SOU app and the main java application all communicating with each other over a common network. The SOU and console app are properly functioning, meeting all requirements. The main java application is made in such that it shows proof of concept. The main java application will be completed in the next build.