Software Requirements Specification

for

Parking System

Version 0.0.3

Prepared by Muhammad Habib

Future Batch 3 Phase 3 - Medan

01 Feb. 20

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Blasius	1 Oct. 2019	Change application flow of online booking	v0.0.1
Blasius	8 Oct 2019	Specification every part	v0.0.2

1. Introduction

1.1 Purpose

The purpose of this document is to build an online system to manage parking slot, to ease the Parklux. In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2 Document Conventions

This template is fetched at https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc. Defined terms are highlighted with bolding. Requirements will come with priority to indicate in which order they will be implemented. Versions might be released with only some of the total list of requirements implemented.

1.3 Intended Audience and Reading Suggestions

While the software requirement specification (SRS) document is written for a more general audience, this document is intended for individuals directly involved in the development of SplitPay. This includes software developers, project consultants, and team managers. This document need not be read sequentially; users are encouraged to jump to any section they find relevant.

1.4 Product Scope

Primarily, the scope of Parklux pertains to manage the parking slot, this application only for registered users, and for android users, this application focuses on managing the parking, generating and scanning the QR, managing slot, level and sections of the parking zone. This application didn’t provide payment system, customer paid on checkout gate, this application can’t provide the customers who park on wrong zone, customer can’t select parking zone system generate automatically

1.5 References

https://www.utdallas.edu/~chung/RE/Presentations07S/Team_1_Doc/Documents/SRS4.0.doc

2. Overall Description

2.1 Product Perspective

In this part, the system allows customers to scan the QR and show the parking slot that selected system for customers. In order to do this, the system randomizes the available slot for the customer that has scanned the generated QR. In Parklux the customer can keep personal by seeing the history of parking and manage the profile. Parklux is a mobile application. The mobile application will work on mobile Android devices. When customers run the application, they can use the functionality of the device. All information will be kept on a database which can be accessed by customers with login.

2.2 Product Functions

The Parklux has main functions for :

Function ID	System Function	Description
F1	Login/Register	Customer must login/register first to access Parklux functions
F2	Home	Customer able to check ongoing and history booking
F3	History Details	Customer able to check the details of history after selecting the history from home
F4	Profile	Customer able to edit the profile like name, email, phone number and password
F5	Scan QR	Customer able to scan the QR for book the parking
56	Parking Details	Customer after scan the QR successfully, be able to see the parking details and parking directions
F7	Check-out	Customer able to checkout parking
F8	Receipt	Customer able to see receipt after checkout or by select history from homepage
F9	Logout	Customer able to logout from the system

- This application for book and system select the parking area after scan the QR code
  - o Customer can : scan the QR (FR-1)
  - o Admin can : Generate QR for user(FR-2)

- This application send notification whenever certain condition met (e.g. booking success after scan QR) (FR-3)

- Can see the receipt and share it
  - o Customer can : see the receipt after parking on history page (FR-4)

- Manage parking slot
  - Admin can : manage parking slot (e.g. edit Levels, sections, slots) and allocation the slot, manage price/hour, add details of the site (FR-5)

- See history
  - Customer can : See the parking history and see the details (FR-6)
  - Admin can : See the parking history on their site (FR-7)
  - Super admin can : See all history and sort by completed or still ongoing (FR-11)

- Receipt will be shows up after checkout from the parking area
  - Customer can : see the receipt by selecting the booking on history page (FR-8)

- Parking slot status will change after booked or checkout (FR-9)
- This application only for registered customer and admin of the parking lot for manage the parking area(FR-10)

2.3 User Classes and Characteristics.

- Admin

Can manage the parking zone layout, update parking sections, parking slot allocation, manage parking level, generate QR code, update details of the parking zone

- Customer

Scan the QR code to create booking parking, update profile, see parking history.

- Super admin

Booking parking zone, update profile, see parking history.

2.4 Operating Environment

Android Based Minimum API Level 21 Android 5.0 (Lollipop), Java Spring Boot v2.1.3, Mongodb Database v4.0.9

2.5 Design and Implementation Constraints

The system has dependency on the database
Internet connection is a constraint for this system because the system is available from cloud therefore customers need to have good network connection to connect to do business process.

2.6 Assumptions and Dependencies

Users of this application are any Android device user that loads this application to their devices. All of the users are in the same class, only one type of user exists. Operating environment is, as just mentioned above, is an Android OS mobile device. An android device that can support basic dependencies of the application is expected for proper user experience. On the other hand, our database server and services can operate on any OS Windows that can supply the database server's fundamental dependencies and needs. One important constraint is privacy and security. Users should be accessing only the authenticated data.

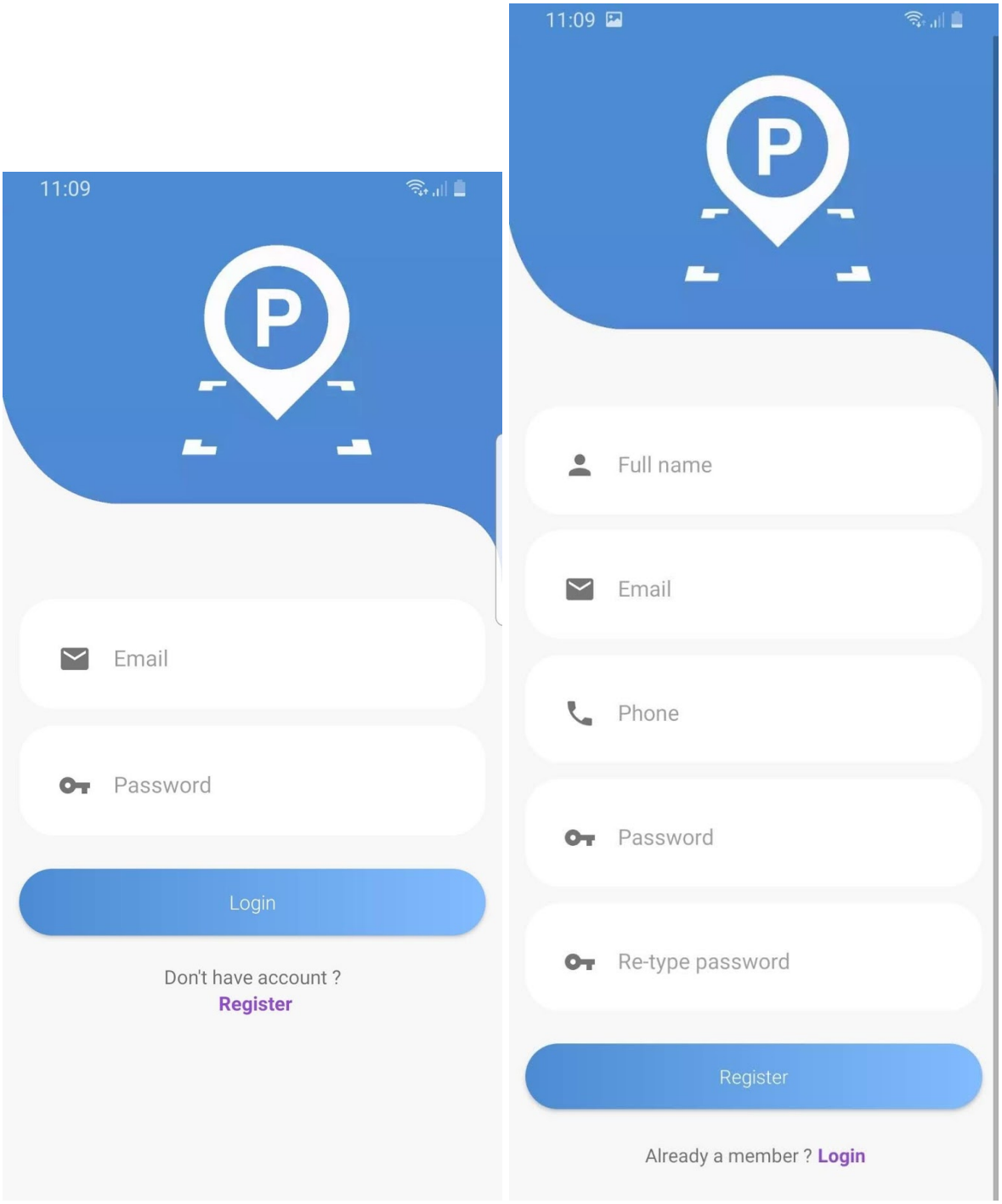
3. External Interface Requirements

3.1 User Interfaces

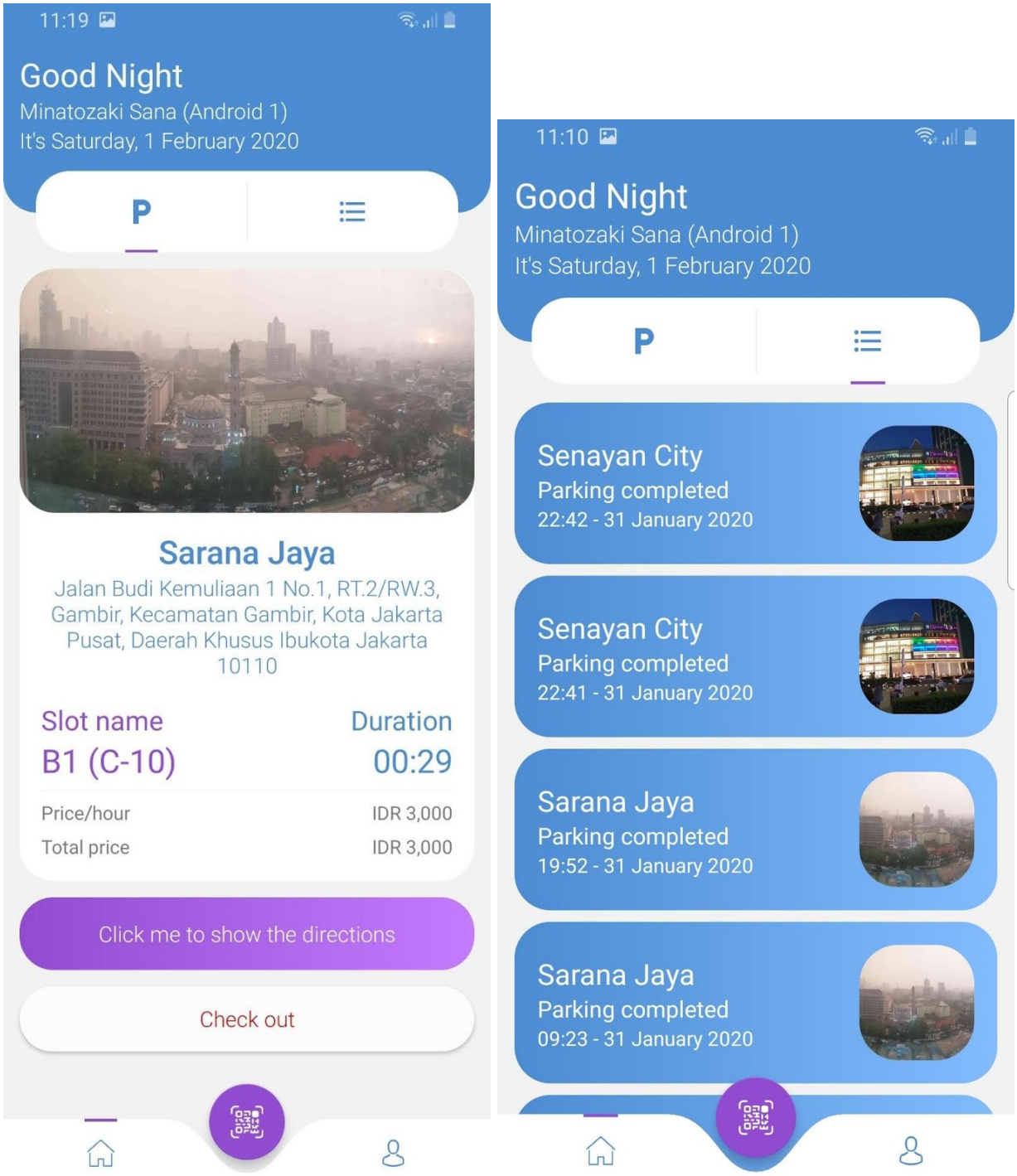
Once the application has been launched, the user will interact primarily with the Android device running the login page if the user have not login yet, after login user directly go to home page or can register

Since the program will be designed with simple mechanics and ease of use in mind, the GUI need not be overly complex. Whatever interface is used should in fact be kept readable and minimalist in order to accommodate the smaller screen size of the Android device.

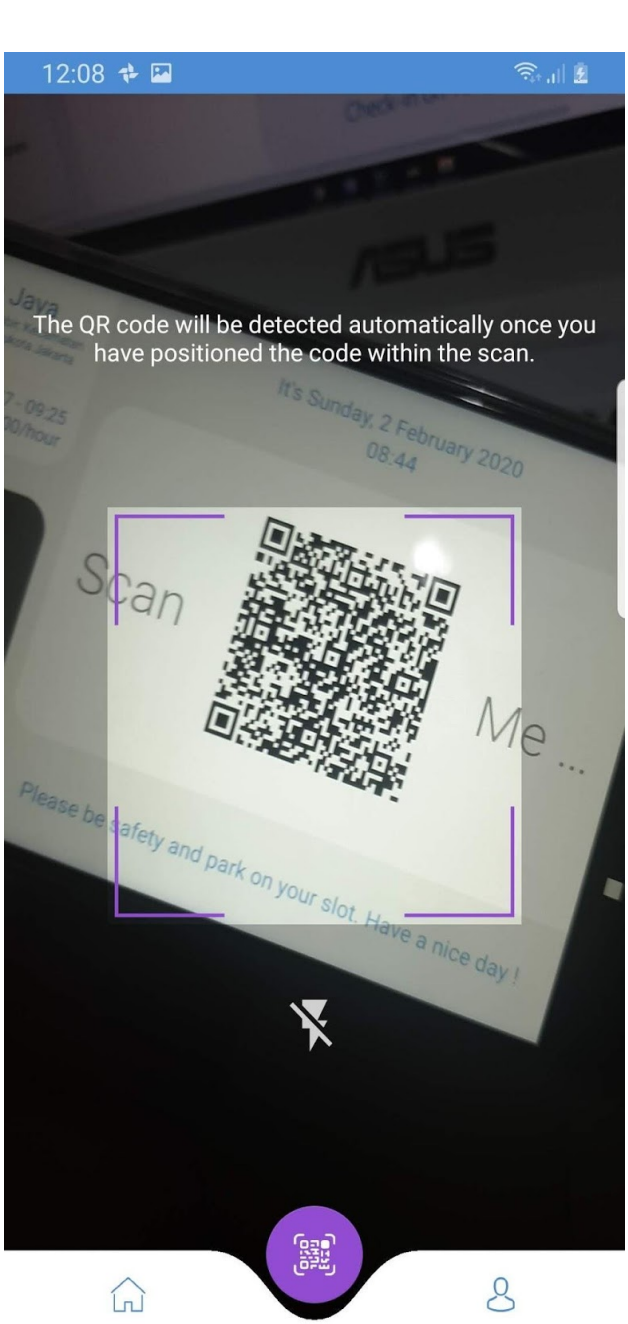
3.1.1 For Customers

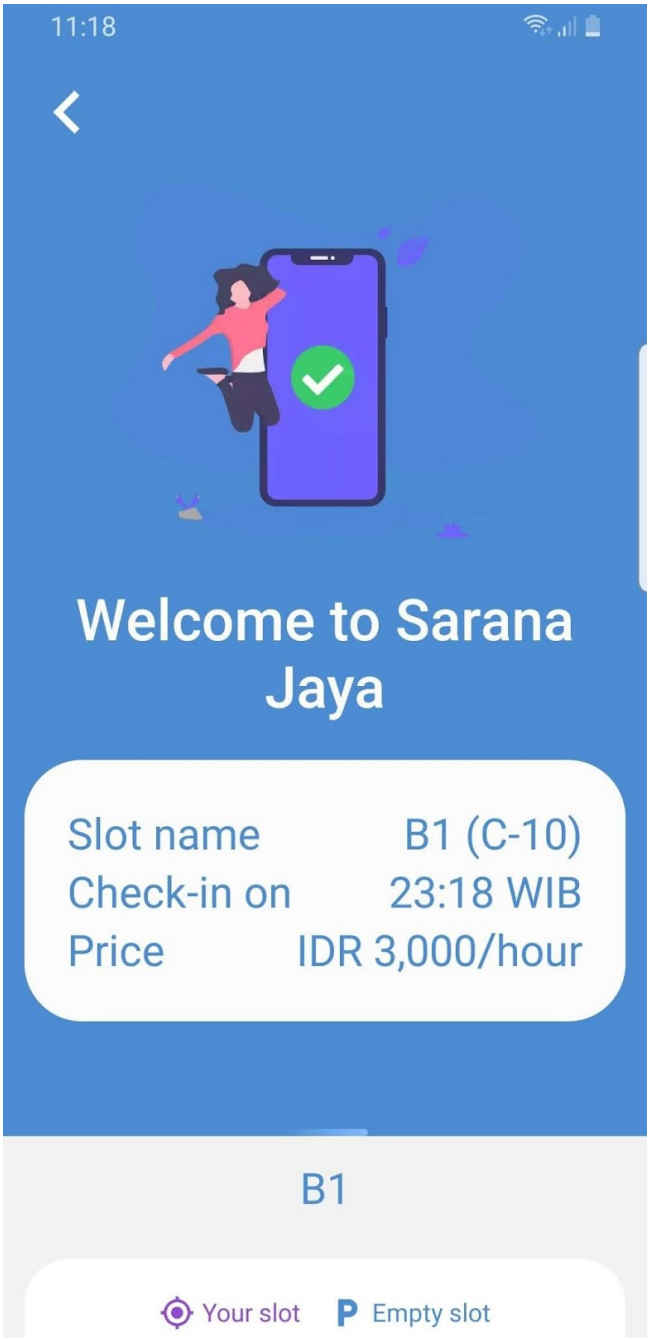


- Login Page (left)
- Register Page (right)

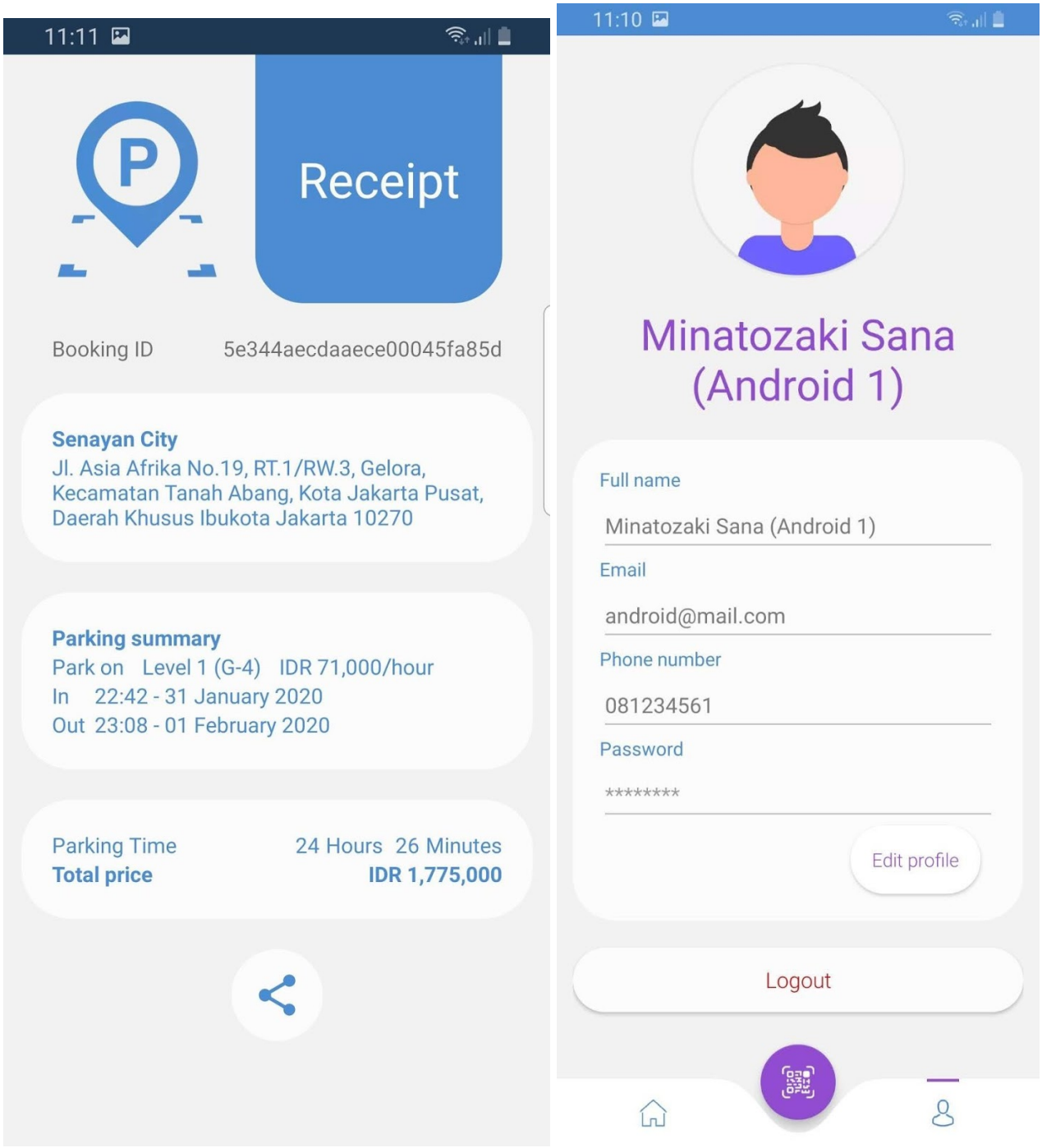


- Home Page, Have announced the customer name. And ongoing parking details(left), list of history booking (right). This app bar have 3 menu : Home, Scan QR, and Profile

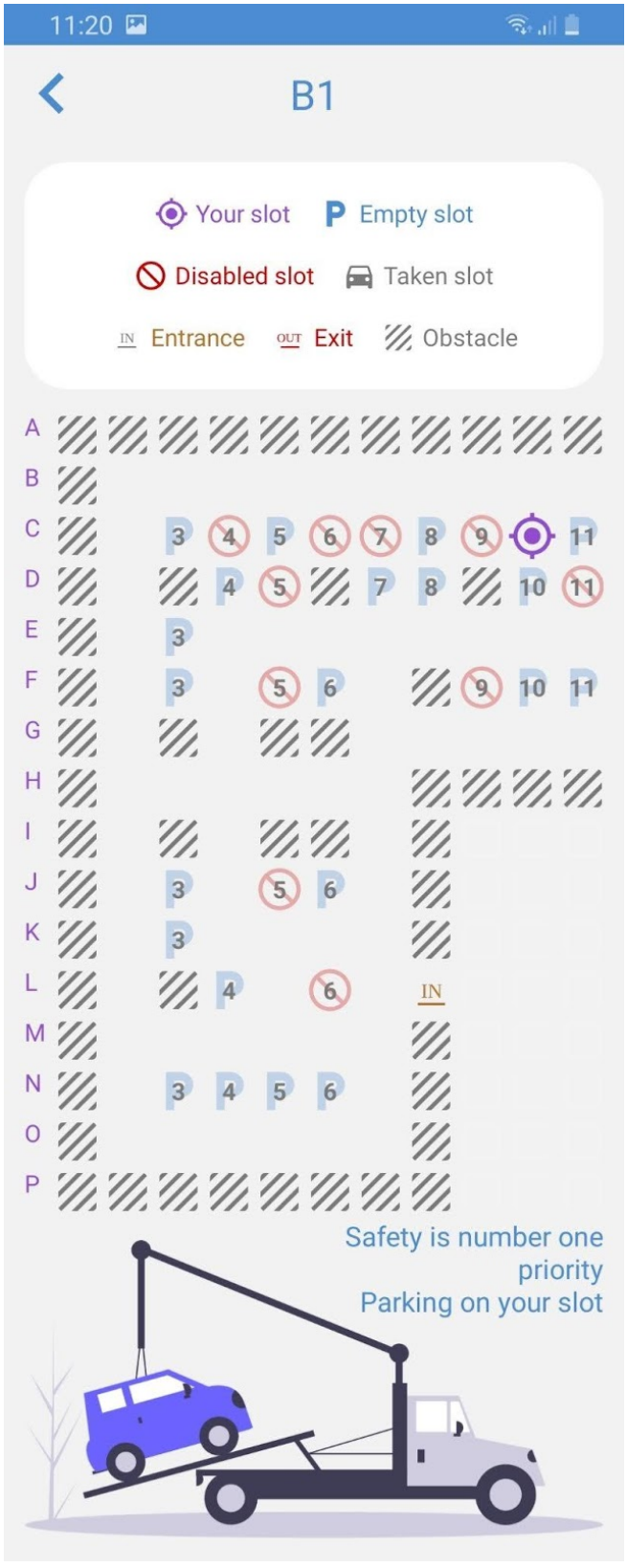




- Scan QR Page, this page for scan QR that generated from admin page(left)
- After scan the QR successfully directly go to the parking details page, this page contains parking details like slot price, and leve(right)

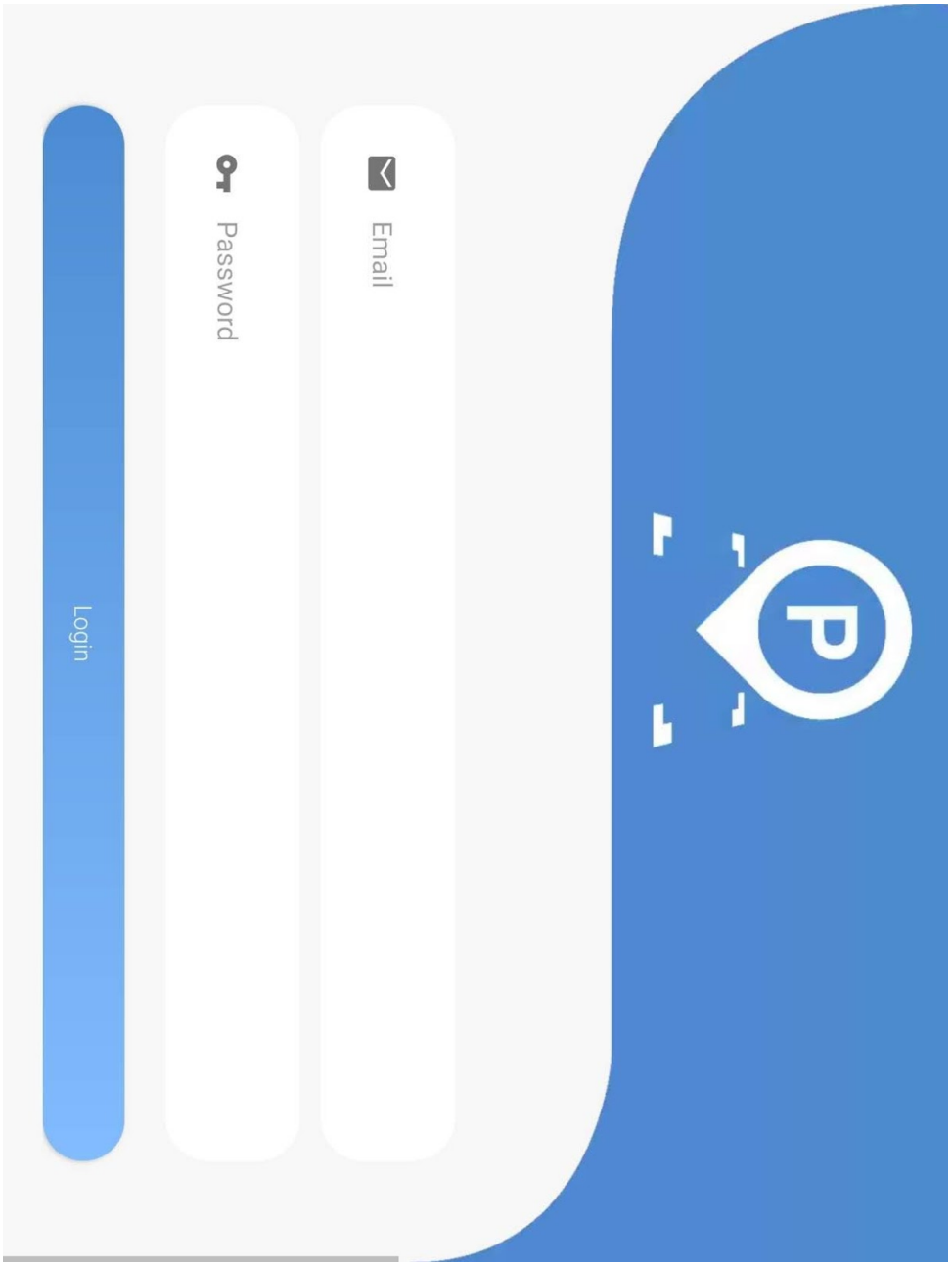


- And after clicking on checkout page or click on history directly go to the receipt details page that has the details (left). profile page(right)

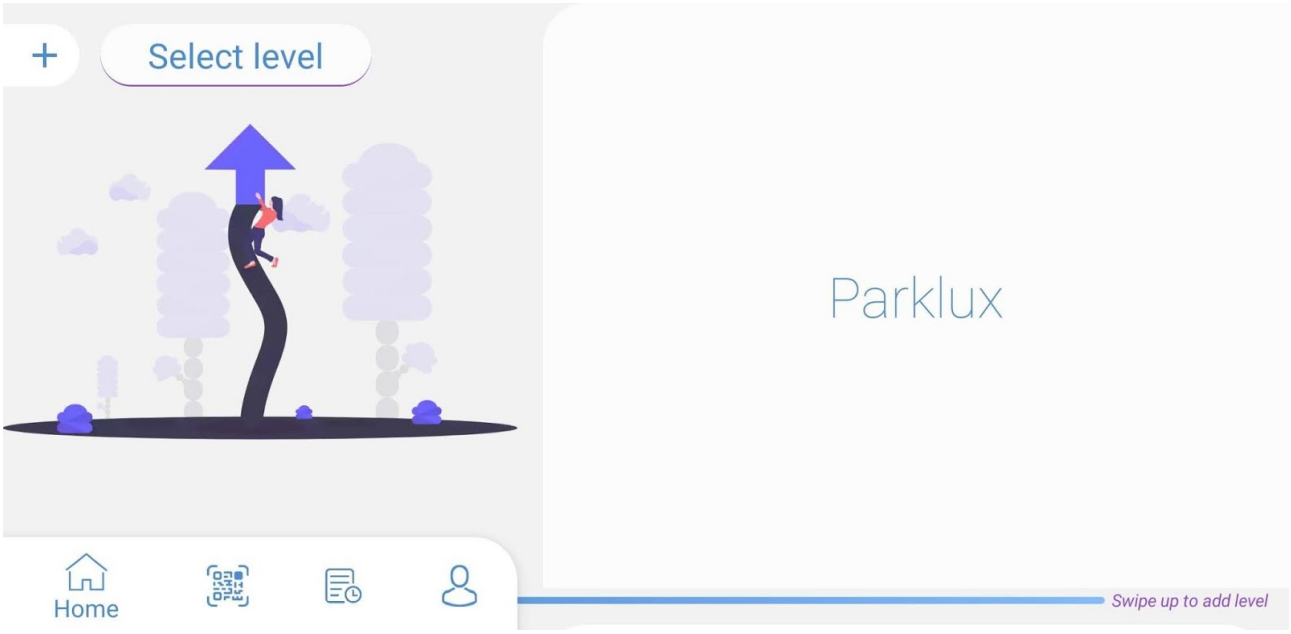


parking direction page

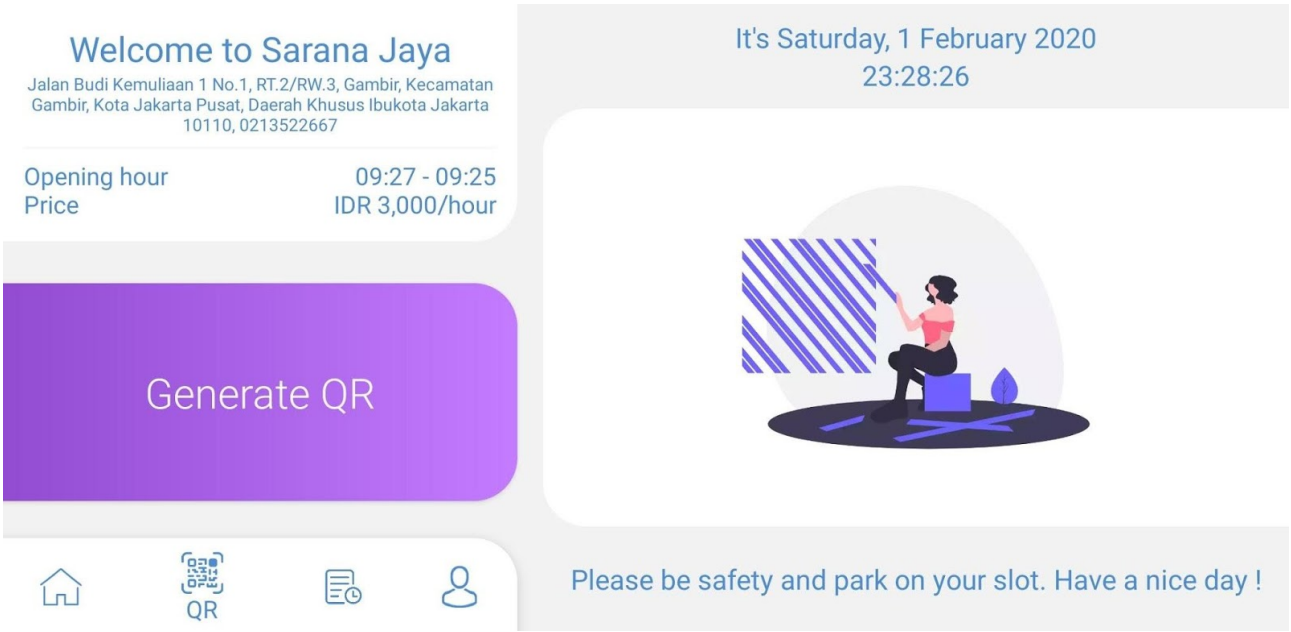
3.1.2 For Admin



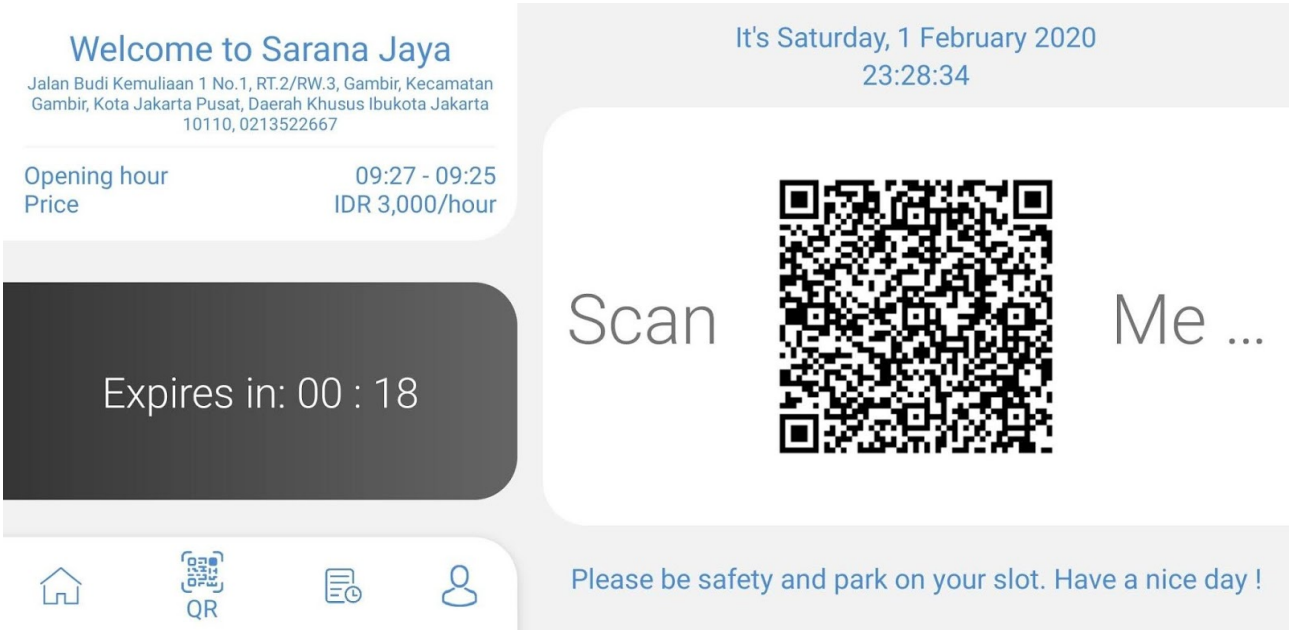
- Login Page for admin



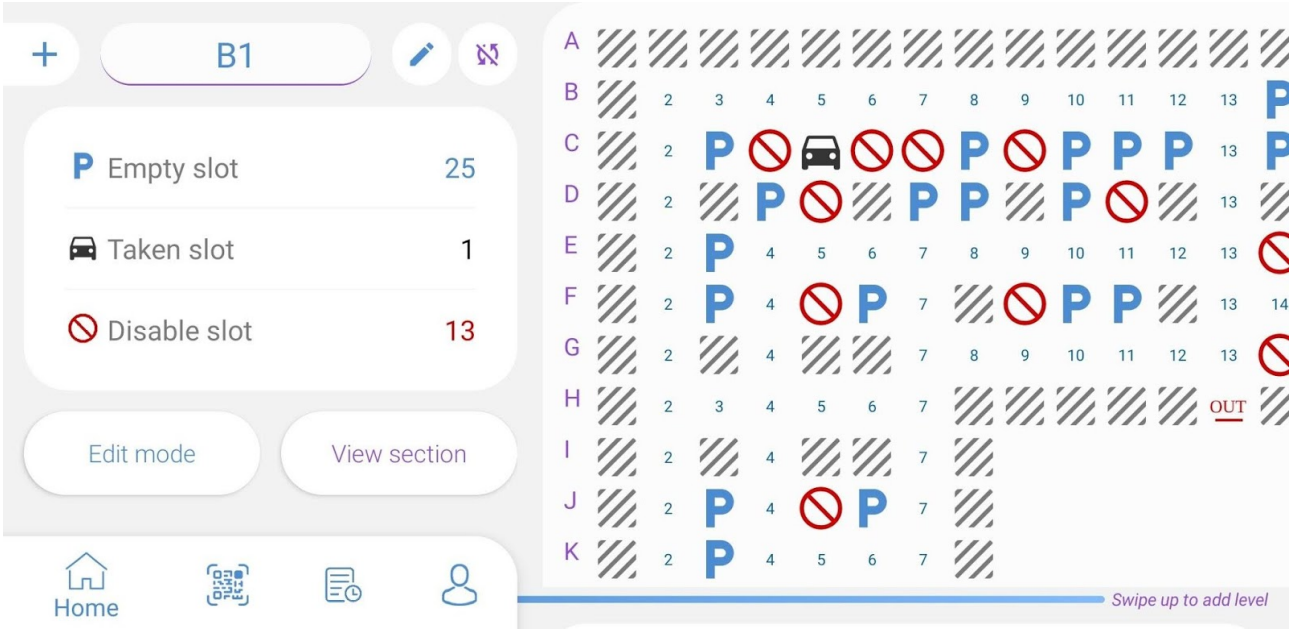
- After login, admin can select the level for managing the parking area



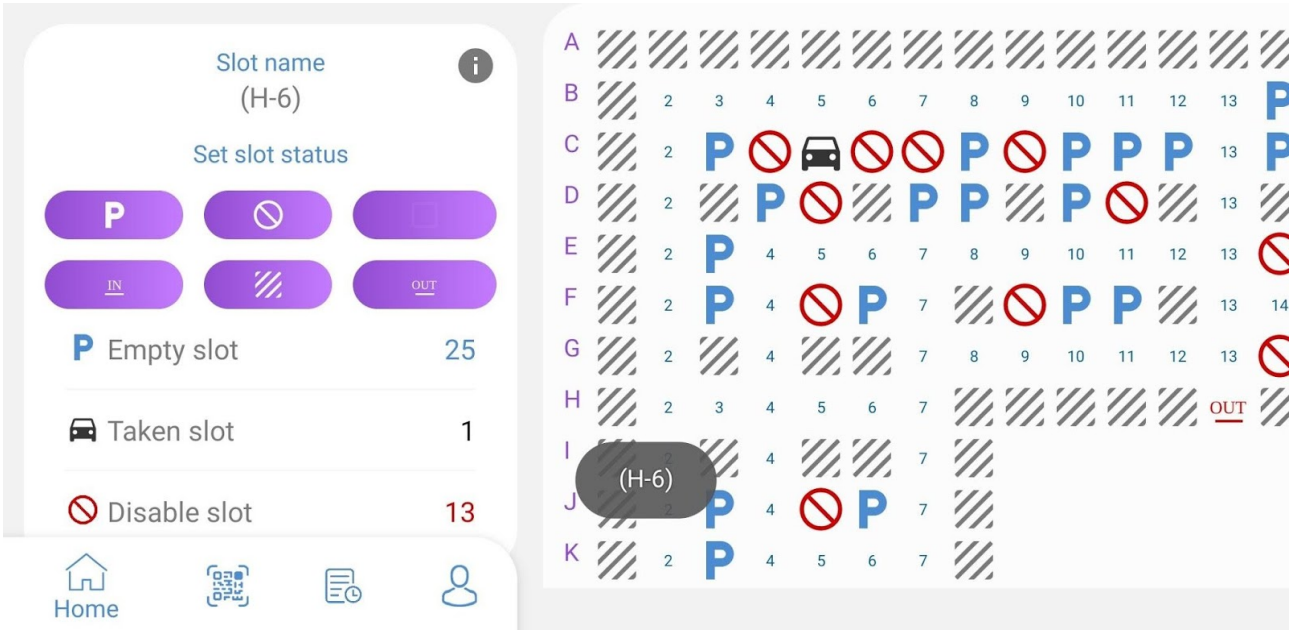
- Page that contains generated QR



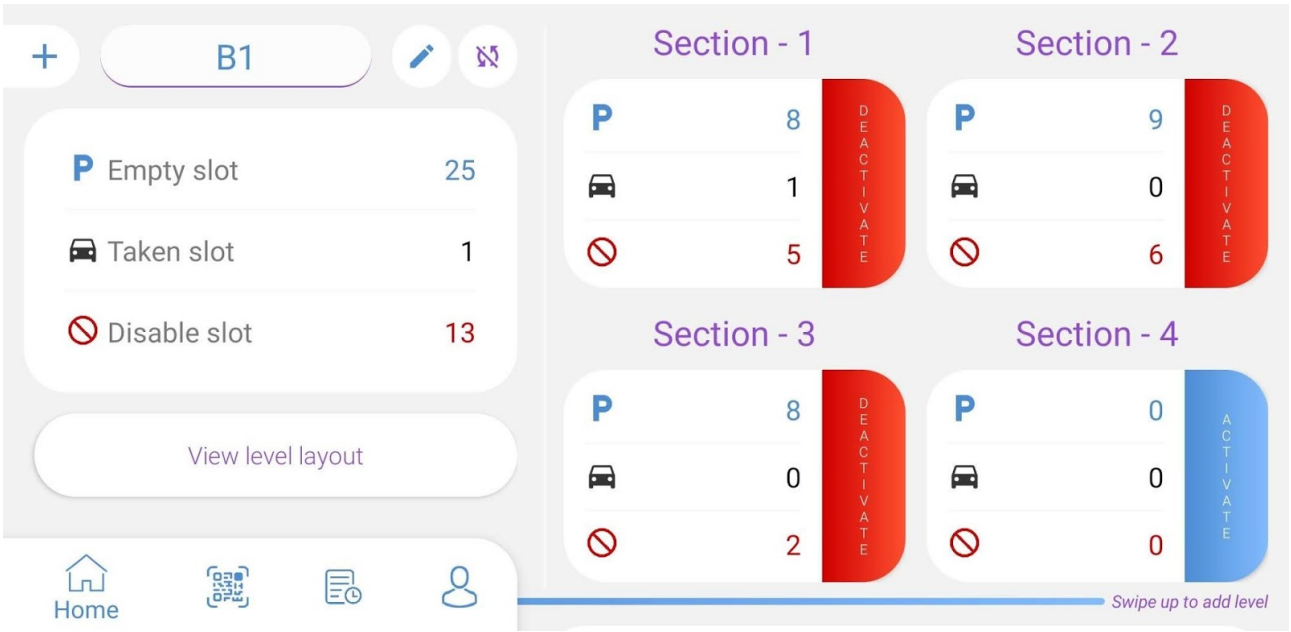
- When generate QR pressed



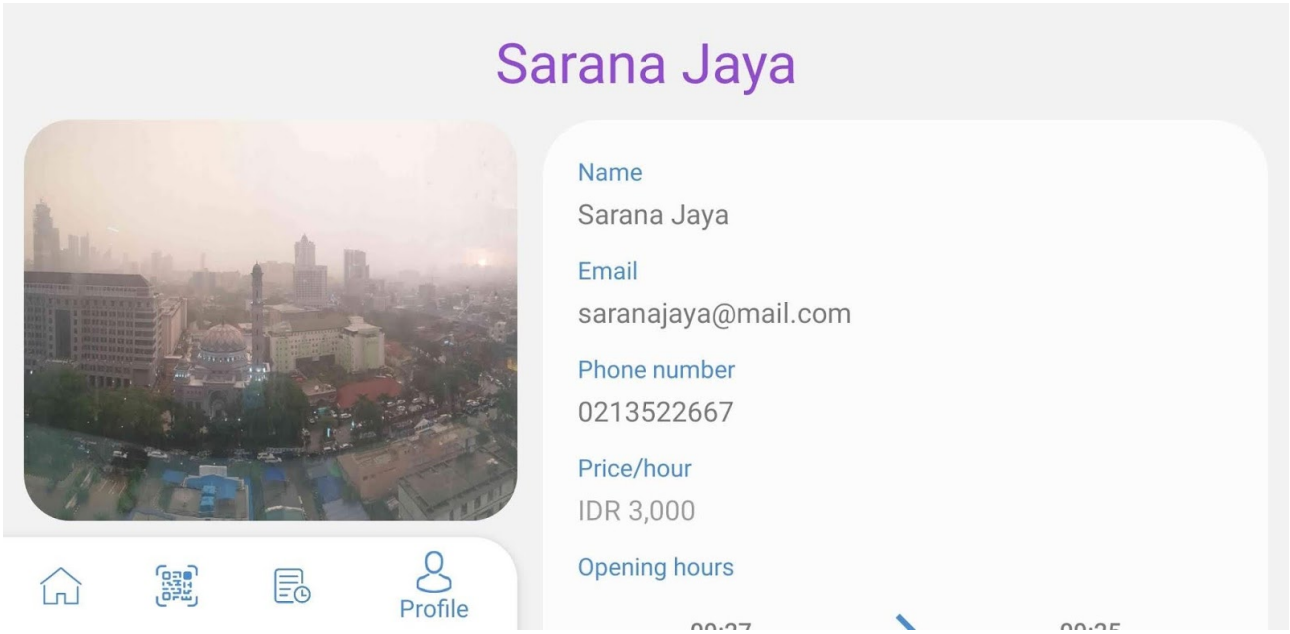
- Manage page, can change the section on view section, select the parking slot and disable that on edit mode, and change the parking level



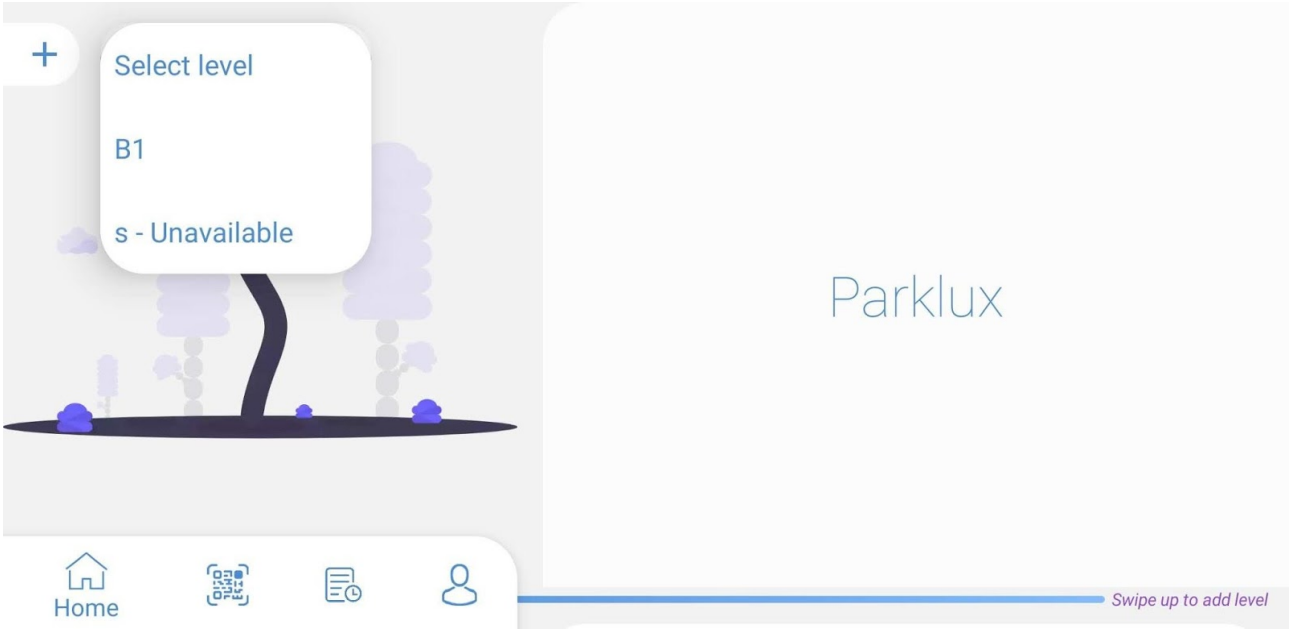
- edit mode



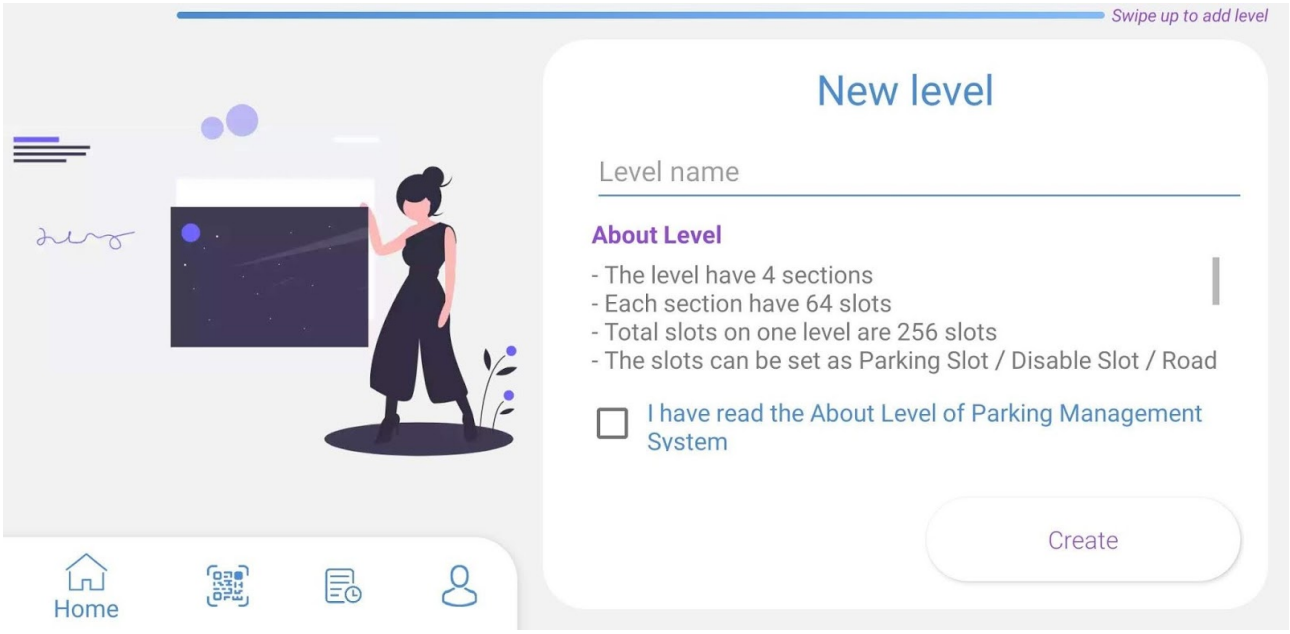
- View sections page



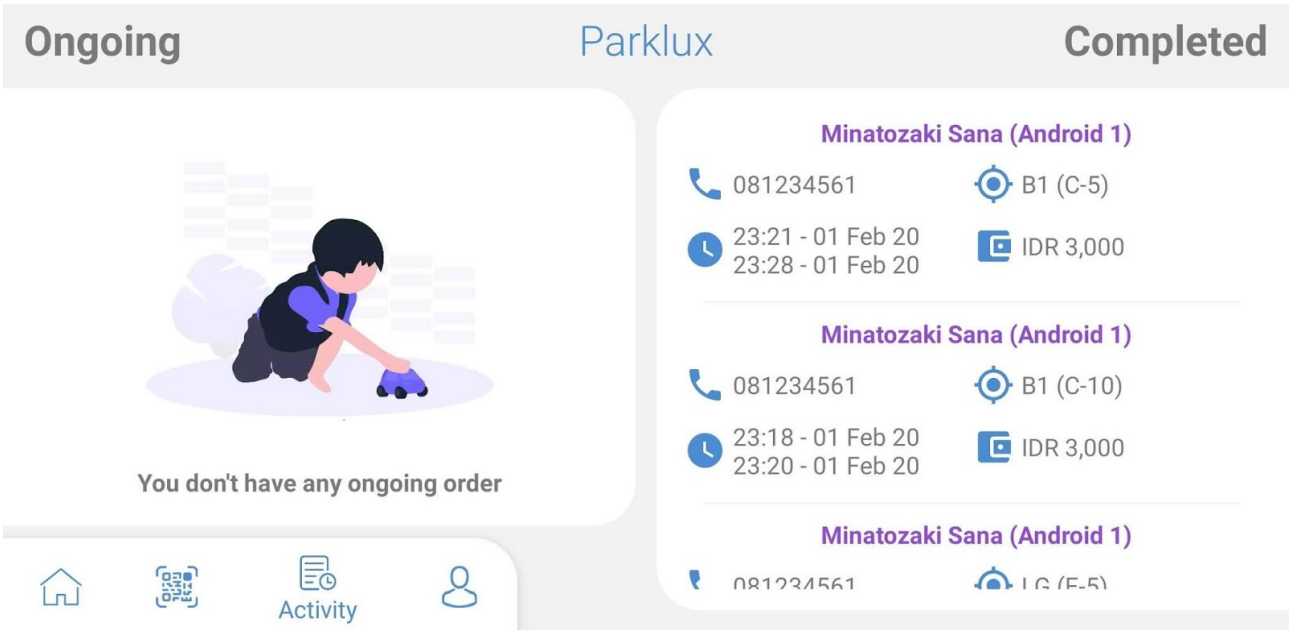
- Profile page, can change the details of parking zone profile



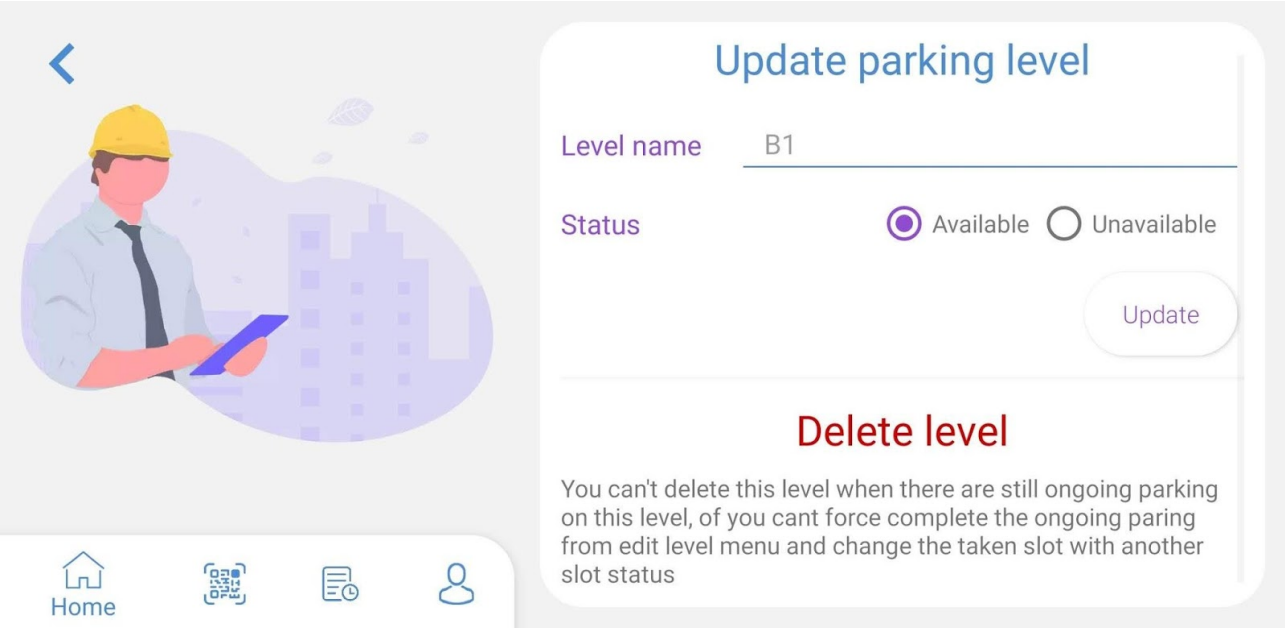
- manage parking page and must select the level first



- Add level page

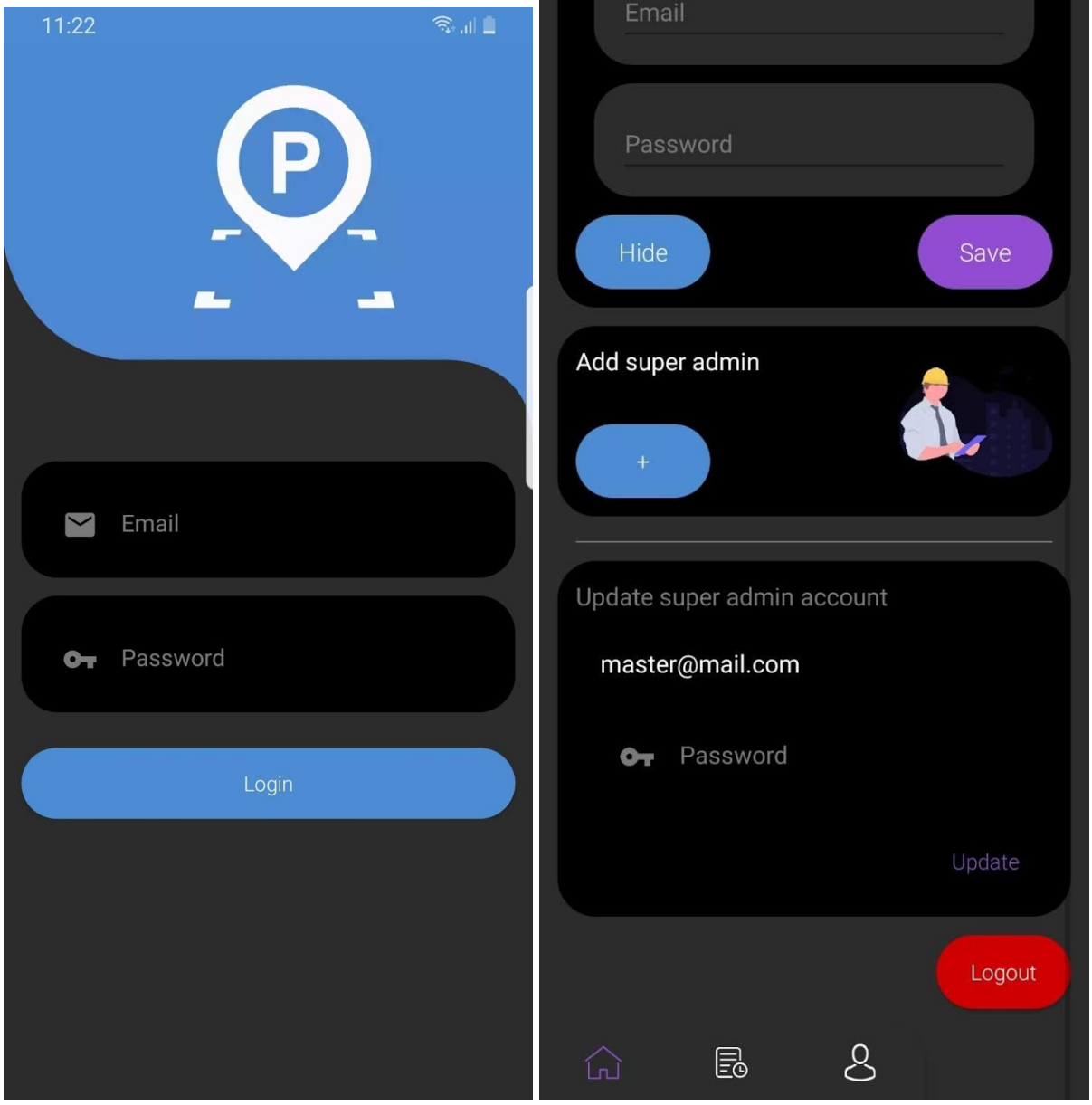


- Activity page

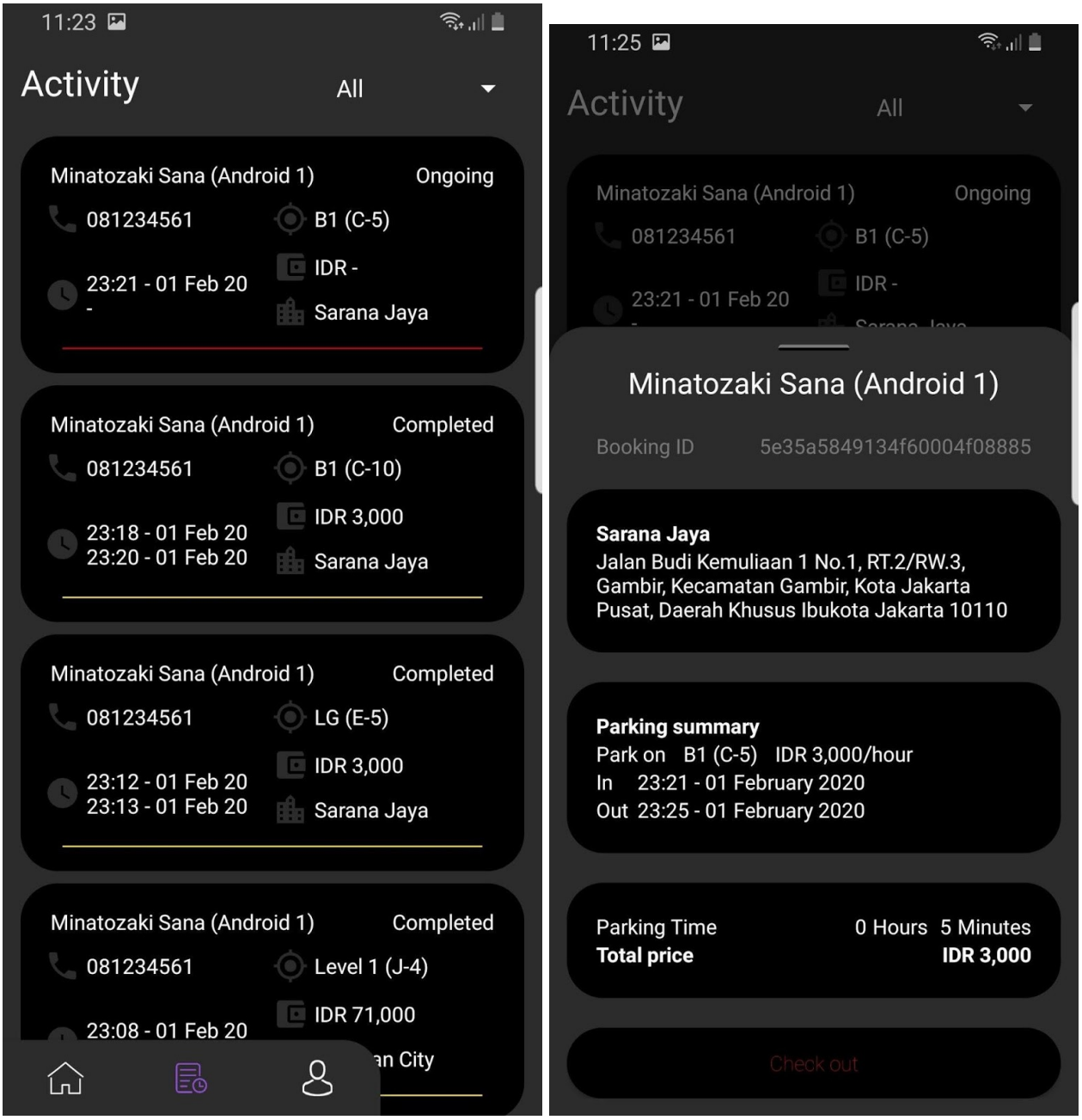


- Edit level page

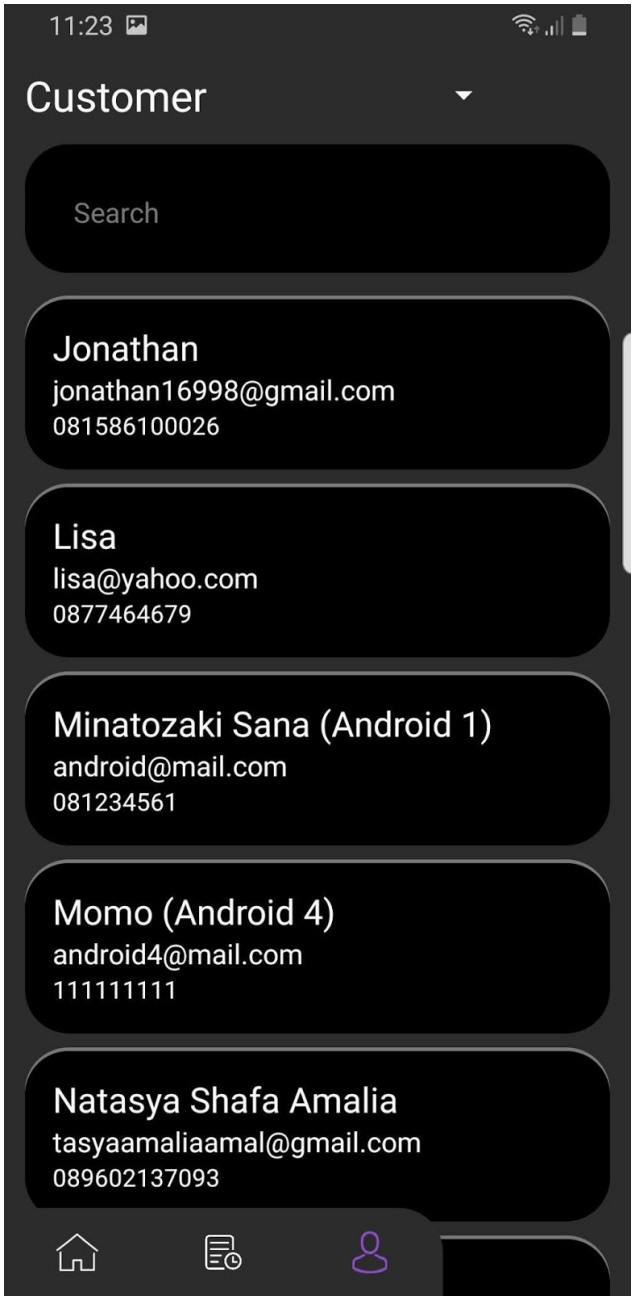
3.1.3 For Super Admin

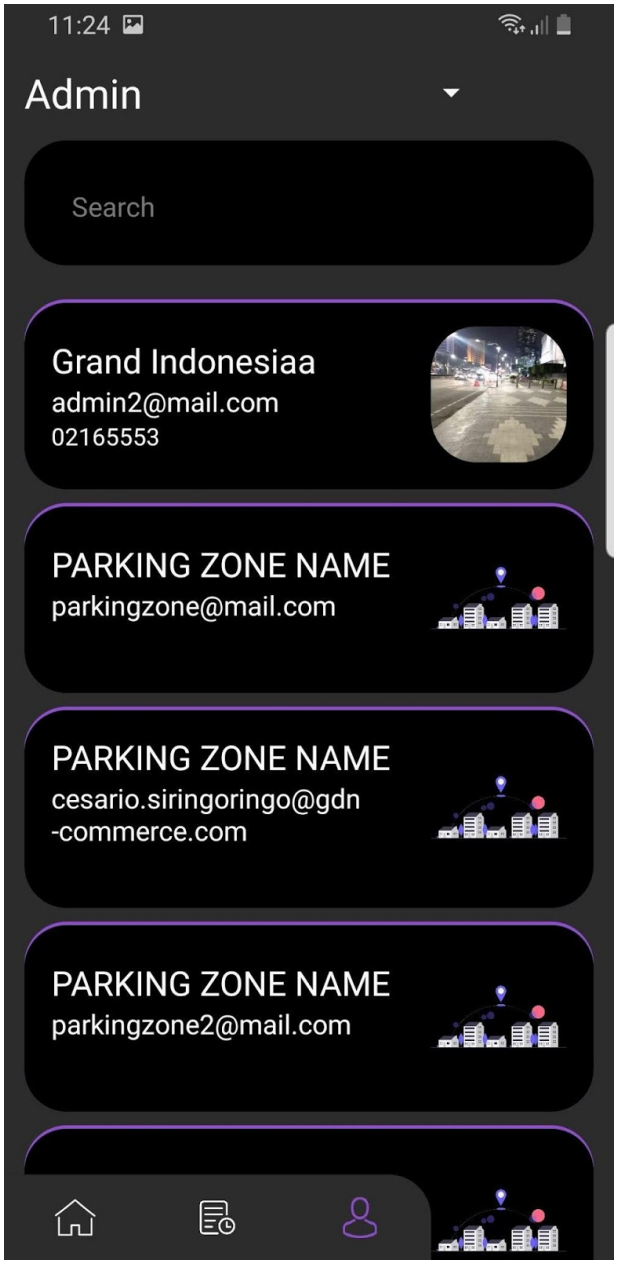


- Login page super admin (left). Home page super admin (right)

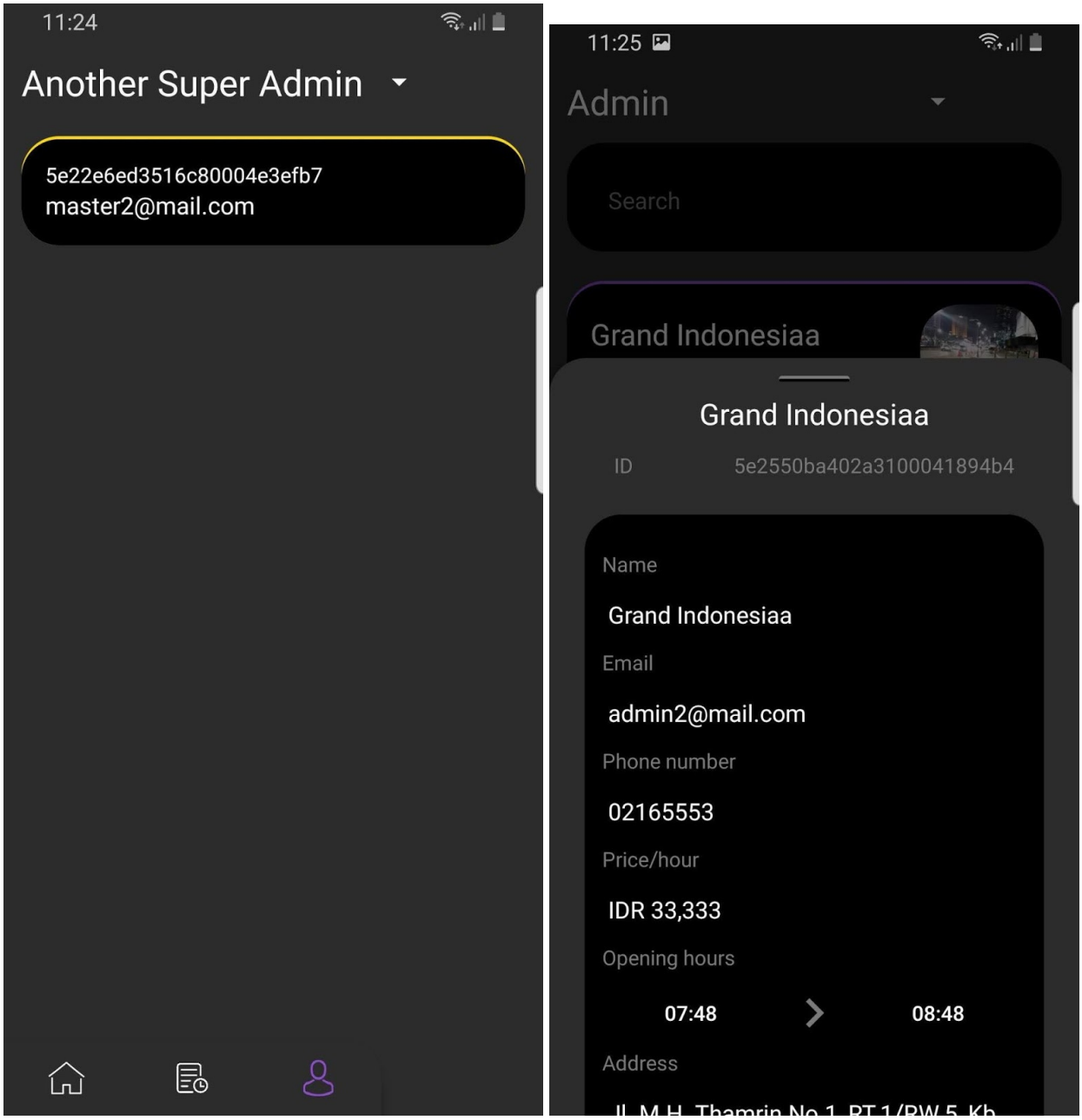


- Activity list (left). Ongoing activity details (right)





- List customer (right). List Admin (right)



- List another super admin (left). admin details (right)

3.2 Hardware Interfaces

Parklux is android-based project therefore any android (API Minimum 21) based platform which has an internet connection is enough to use this system

3.3 Software Interfaces

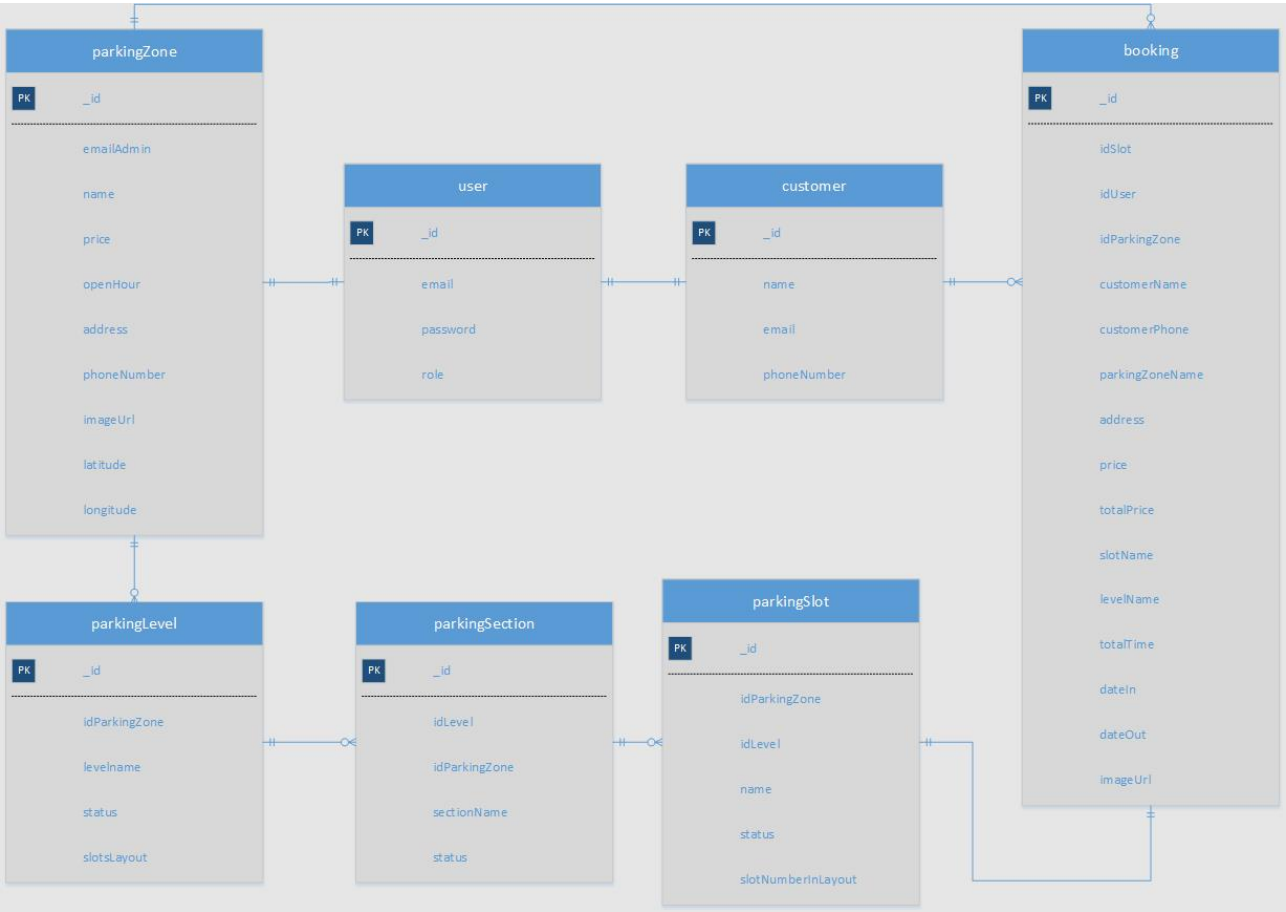
Parklux is android-based project therefore any android (API Minimum 21) based platform which has an internet connection is enough to use this system

3.4 Communications Interfaces

Parklux is android-based project therefore any android (API Minimum 21) based platform which has an internet connection is enough to use this system

4. System Features

4.1 Entity Relationship Diagram



is the entity relationship diagram

4.2 Register

4.2.1 This provide security for system to access login and register the user to the system
Priority : High

4.2.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.

Input full name, email, phone number, and password on register page	

4.2.3 Functional Requirements

REQ-1: The system shall display the register page.
REQ-2: The system shall validate input from the user.
REQ-3: The system shall redirect users to login if validate succeeds.
REQ-4: The system shall display error messages when validate failed.

4.3 Login

4.3.1 This provide security for system to access the feature
Priority : High

4.3.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
Input the email and password if the user not login.	

4.3.3 Functional Requirements

REQ-1: The system shall display the login page.
REQ-2: The system shall validate input from the user.
REQ-3: The system shall redirect users to dashboard if validate succeeds.
REQ-4: The system shall display error messages when validate failed.

4.4 Home

4.4.1 This allow user to see the announce, weather details, and history or ongoing book

4.4.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
The user can see the ongoing / history	Directly go to home page

4.4.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature.
REQ-5: The system shall validate the booking has finished or still ongoing
REQ-6: The system shall show the weather details

4.5 History Details

4.5.1 This allow user to see where user's parking zone, when the user checkin and checkout, name of the parking zone, price per hour and total summary price

4.5.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
The user select history from home page	The System check, validate and direct user to the booking details/receipt
The user can check booking details/receipt	The System displays the receipt of the book

4.5.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature.

REQ-7: The system shall count the total price per hour times total hour the user parking

REQ-8: The system shall show the receipt page

4.6 Profile

4.6.1 This allow allow user to change profile details

4.6.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
The user selects the profile page	The System shall show the profile page with full name, email, phone number, password, and profile picture.
The user can edit the details of their profile and add a picture or change it.	The System checks and validates the changes that users make, if user press save, if any changes and nothing conflict system save the data or if it is not valid, system tells the user that inputed data is not valid and the data not saved yet.

4.6.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature

REQ-9: The system shall validate the changes that user make

REQ-10: The system shall show the profile page

REQ-11: The system shall tell user when the data not saved

4.7 Scan QR

4.7.1 This allow user to scan the QR after press the generate button from the admin system

4.7.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.

The user selects the QR Scanner page from the navigation bar	The System shall show the QR scanner page and system access the camera to scan the QR and system can read the data from the QR

4.7.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature
REQ-12: The system shall open the QR scanner page
REQ-13: The system shall access the camera
REQ-14: The system shall read the data from QR

4.8 Parking Details

4.8.1 This allow user to see where the user park from generated QR

4.8.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
The user can see the details of booked parking spot by selecting from home or directly after scan the QR successful	The System shall show the location like level,section and slot where the user park, and the system save the data when the user check in and system show animation directions to the parking slot

4.8.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature
REQ-15: The system shall open the parking details page
REQ-16: The system shall show the correct location like level, section, and slot where the user park
REQ-17: The system shall blueprint when the user check in
REQ-18: The system shall show animation directions to the parking slot

4.9 Check-out

4.9.1 This allow user to checkout from parking

4.9.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
The user presses the checkout button from ongoing booking	The System shall show the checkout button on ongoing booking, if user press that button the system directly go to the receipt page and the parking slot set to available

4.9.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature

REQ-19: The system shall show the checkout button
REQ-20: The system shall directly go to receipt page after checkout
REQ-21: The system shall set the parking slot to available

4.10 Receipt

4.10.1 This allow user see the receipt of finished booking
4.10.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
The user can see the receipt after checkout or from select the finished past booked history	The System shall count the total price and show the receipt page

4.10.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature
REQ-22: The system shall open the receipt page
REQ-23: The system shall count the total price correctly

4.11 Logout

4.11.1 This provides the user for logout
4.11.2 Stimulus/Response Sequences

User Action	System Application
	The System checks to see if the user is logged in, if not then the systems require the user to login.
The user go to the profile page	The System shall open the profile page
The user tap the logout button	The System shows popup confirmation for users, if the user does not cancel the process the user logout from the system. if the user cancel the system back to the profile page

4.11.3 Functional Requirements

REQ-0: The user must be logged-in to access the system feature
REQ-24: The system shall show the popup confirmation
REQ-25: The system shall log the user out from the system

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Parklux requires a system for server with at least about 1000 megahertz CPU and 350 megabytes of RAM. Performance depends on the system processing and as a result, the system requirements for bigger CPU and RAM are more demanding.

5.2 Safety Requirements

To ensure that no one of Parklux's users loses any data while using Parklux(due to a crash or a bug of some kind) the system checks Parklux system regularly.

5.3 Security Requirements

- Using Spring Security on server
- User's password shall be encrypted

5.4 Software Quality Attributes

Availability of a software system is the percentage of time where the system is fully operational.
The usability describes the ease of use of a software and is defined by adequacy, learnability and robustness.
Maintainability software is considered maintainable if defects and their cause can be isolated easily, defects can be fixed easily, changes can be made easily, extensions can be made easily.
Efficient software emphasizes a careful use of resources like e.g. CPU, RAM or external (secondary) storage.
The portability describes how easy it is to port the program to another platform or device.
The security of a software is partly related to the requirements as security concerns should be part of the functional requirements for the system in question.

5.5 Business Rules

The Parklux will perform under three users which are Customer, Admin and Super Admin. The super admin will perform as the system administrator and has the highest privilege in the system. The admin will manage the parking zone. The customer can only access the business function like book parking functions.