Parking Management

Analysis and Design Document

Student: Aionitoaie Mihai  
 Bercean Andrei

**Group: 30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24/April/19 | 1.0 | Iteration 1.1 | Bercean Andrei |
| 6/May/19 | 1.1 | Iteration 1.2 | Bercean Andrei |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

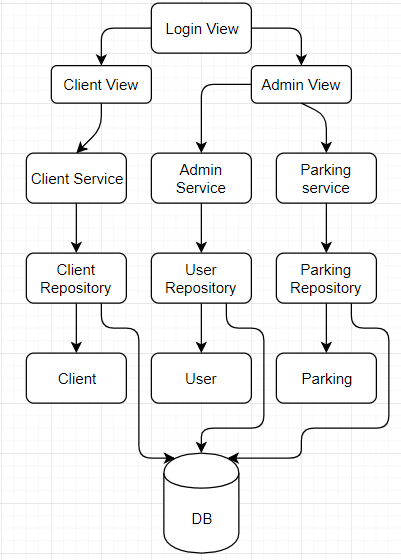
# Project Specification

The Parking Management application manages a parking facility with multiple levels. Parking spots can be reserved as well as take as walk-in. A user will need an account and all the accounts will be managed by an admin account.

# Elaboration – Iteration 1.1

# Domain Model

[Define the domain model and create the conceptual class diagrams]

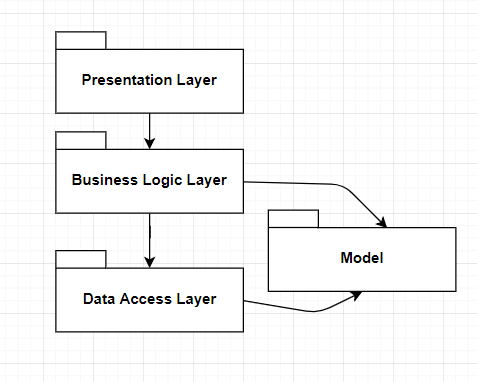


# Architectural Design

## Conceptual Architecture

The application will be designed in accordance with the Layered architectural pattern as it can be seen. The pattern was chosen for its intuitive way of separating the code as well as making changes to logic layers or data access be more self-contained and not propagate into other layers.

## Package Design



## Component and Deployment Diagrams

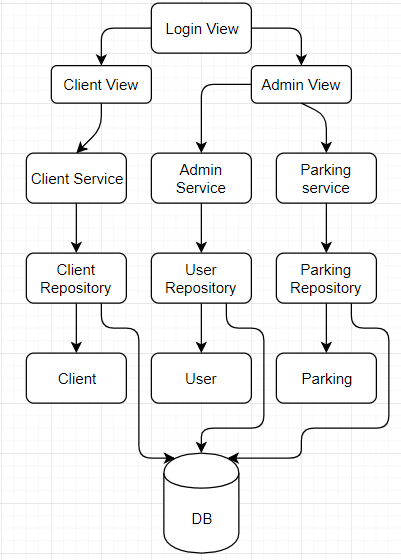
Deployed locally on a single device, with no web component.

# Elaboration – Iteration 1.2

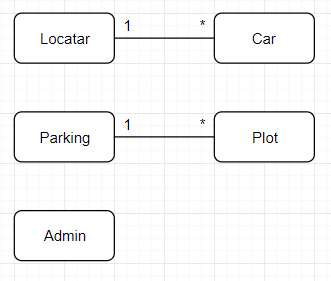
# Design Model

## Dynamic Behavior

## Class Design



# Data Model

**

# Unit Testing

The tests will be done with JUnit on the CRUD operations for each

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography