|  |  |
| --- | --- |
| **­** | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

|  |  |
| --- | --- |
| **Report 4** | |
| **Parking Guidance System Solution** | |
| **Group 1** | |
| **Group members** | Trần Nguyễn Minh Trung – Team Leader – SE61496  Bùi Phú Hiệp – Team Member – SE61438  Nguyễn Đỗ Phương Huy – Team Member – SE61358 |
| **Supervisor** | Nguyễn Đức Lợi |
| **Ext. Supervisor** | N/A |
| **Capstone Project Code** | PGSS |

- Ho Chi Minh City, Jan, 2017

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 1](#_Toc479711604)

[List of Tables 7](#_Toc479711605)

[List of Figures 14](#_Toc479711606)

[Definitions, Acronyms and Abbreviations 17](#_Toc479711607)

[D. Software – Hardware Design Description 18](#_Toc479711608)

[1. Design Overview 18](#_Toc479711609)

[2. System Architectural Design 19](#_Toc479711610)

[2.1. Hardware Program Architecture Description 20](#_Toc479711611)

[2.2. Mobile Application Architecture Description 22](#_Toc479711612)

[2.3. Web API Architecture Description 23](#_Toc479711613)

[3. Component Diagram 23](#_Toc479711614)

[3.1. General Component Diagram 23](#_Toc479711615)

[3.2. Lot Device Diagram 24](#_Toc479711616)

[3.3. Information Device Diagram 24](#_Toc479711617)

[3.4. Server Diagram 25](#_Toc479711618)

[3.5. Component Dictionary 25](#_Toc479711619)

[4. Detailed Description 27](#_Toc479711620)

[4.1. Web API Detailed Description 27](#_Toc479711621)

[4.1.1. Class Diagram 27](#_Toc479711622)

[4.1.2. Class Diagram Explanation 29](#_Toc479711623)

[4.1.2.1. Model Class Diagram 29](#_Toc479711624)

[4.1.2.2. View Model Class Diagram 32](#_Toc479711625)

[4.1.2.3. Repository Class Diagram 35](#_Toc479711626)

[4.1.2.4. Service Class Diagram 36](#_Toc479711627)

[4.1.3. Interaction Diagram 39](#_Toc479711628)

[4.2. Mobile Detailed Description 43](#_Toc479711629)

[4.2.1. Class Diagram 43](#_Toc479711630)

[4.2.2. Class Diagram Explanation 53](#_Toc479711631)

[4.2.2.1. [Activities] MainActivity 53](#_Toc479711632)

[4.2.2.2. [Activities] CarParkDetailActivity 54](#_Toc479711633)

[4.2.2.3. [Activities] ManagerActivity 55](#_Toc479711634)

[4.2.2.4. [Activities] AreaListActivity 56](#_Toc479711635)

[4.2.2.5. [Activities] ParkingLotListActivity 57](#_Toc479711636)

[4.2.2.6. [Activities] LoginActivity 57](#_Toc479711637)

[4.2.2.7. [Activities] RegisterActivity 58](#_Toc479711638)

[4.2.2.8. [Activities] UserActivity 59](#_Toc479711639)

[4.2.2.9. [Activities] CarParkListActivity 60](#_Toc479711640)

[4.2.2.10. [Activities] TransactionActivity 61](#_Toc479711641)

[4.2.2.11. [Adapters] AreaAdapter 61](#_Toc479711642)

[4.2.2.12. [Adapters] AreaAdapter.OnItemClickListener 62](#_Toc479711643)

[4.2.2.13. [Adapters] AreaAdapter.ViewHolder 62](#_Toc479711644)

[4.2.2.14. [Adapters] CarParkAdapter 63](#_Toc479711645)

[4.2.2.15. [Adapters] CarParkAdapter.OnItemClickListener 63](#_Toc479711646)

[4.2.2.16. [Adapters] CarParkAdapter.ViewHolder 63](#_Toc479711647)

[4.2.2.17. [Adapters] ParkingLotAdapter 64](#_Toc479711648)

[4.2.2.18. [Adapters] ParkingLotAdapter.OnItemClickListener 64](#_Toc479711649)

[4.2.2.19. [Adapters] ParkingLotAdapter.ViewHolder 64](#_Toc479711650)

[4.2.2.20. [Adapters] CarParkAdvanceAdapter 65](#_Toc479711651)

[4.2.2.21. [Adapters] CarParkAdvanceAdapter.OnItemClickListener 65](#_Toc479711652)

[4.2.2.22. [Adapters] CarParkAdvanceAdapter.ViewHolder 65](#_Toc479711653)

[4.2.2.23. [Adapters] TransactionAdapter 66](#_Toc479711654)

[4.2.2.24. [Adapters] TransactionAdapter.OnItemClickListener 66](#_Toc479711655)

[4.2.2.25. [Adapters] TransactionAdapter.ViewHolder 67](#_Toc479711656)

[4.2.2.26. [Adapters] UserInfoWindowAdapter 67](#_Toc479711657)

[4.2.2.27. [Helpers] AppDatabaseHelper 67](#_Toc479711658)

[4.2.2.28. [Helpers] PubnubHelper 68](#_Toc479711659)

[4.2.2.29. [Helpers] AccountHelper 69](#_Toc479711660)

[4.2.2.30. [Helpers] MapMarkerHelper 69](#_Toc479711661)

[4.2.2.31. [Helpers] InternetHelper 69](#_Toc479711662)

[4.2.2.32. [Interfaces] AreaClient 70](#_Toc479711663)

[4.2.2.33. [Interfaces] TransactionClient 70](#_Toc479711664)

[4.2.2.34. [Interfaces] CarParkClient 70](#_Toc479711665)

[4.2.2.35. [Interfaces] AccountClient 71](#_Toc479711666)

[4.2.2.36. [Interfaces] ParkingLotClient 71](#_Toc479711667)

[4.2.2.37. [Models] Account 71](#_Toc479711668)

[4.2.2.38. [Models] Area 72](#_Toc479711669)

[4.2.2.39. [Models] CheckCode 72](#_Toc479711670)

[4.2.2.40. [Models] Geo 72](#_Toc479711671)

[4.2.2.41. [Models] CarPark 73](#_Toc479711672)

[4.2.2.42. [Models] Transaction 73](#_Toc479711673)

[4.2.2.43. [Models] ParkingLot 74](#_Toc479711674)

[4.2.2.44. [Models] TransactionStatus 74](#_Toc479711675)

[4.2.2.45. [Models] AreaStatus 75](#_Toc479711676)

[4.2.2.46. [Models] ParkingLotStatus 75](#_Toc479711677)

[4.2.2.47. [Network] AccountPackage 76](#_Toc479711678)

[4.2.2.48. [Network] AreaPackage 76](#_Toc479711679)

[4.2.2.49. [Network] CarParkPackage 76](#_Toc479711680)

[4.2.2.50. [Network] CheckCodePackage 77](#_Toc479711681)

[4.2.2.51. [Network] CommandPackage 77](#_Toc479711682)

[4.2.2.52. [Network] RealtimeMapPackage 78](#_Toc479711683)

[4.2.2.53. [Network] ControlPubnubPackage 78](#_Toc479711684)

[4.2.2.54. [Network] MobilePubnubPackage 78](#_Toc479711685)

[4.2.2.55. [Network] GetCoordinationPackage 79](#_Toc479711686)

[4.2.2.56. [Network] CarParkAdvancePackage 79](#_Toc479711687)

[4.2.2.57. [Network] NotificationPackage 79](#_Toc479711688)

[4.2.2.58. [Network] TransactionPackage 80](#_Toc479711689)

[4.2.2.59. [Network] ParkingLotPackage 80](#_Toc479711690)

[4.2.2.60. [Network] ServiceGenerator 80](#_Toc479711691)

[4.2.2.61. [Network] DateTypeDeserializer 81](#_Toc479711692)

[4.2.3. Interaction Diagram 81](#_Toc479711693)

[4.2.3.1. Load car parks on map 81](#_Toc479711694)

[4.2.3.2. Load car parks in list 82](#_Toc479711695)

[4.2.3.3. Car park detail information 83](#_Toc479711696)

[4.2.3.4. Reserve parking lot process 84](#_Toc479711697)

[4.2.3.5. Transaction controller 85](#_Toc479711698)

[4.2.3.6. Manage car parks 87](#_Toc479711699)

[4.2.3.7. Manage areas 88](#_Toc479711700)

[4.2.3.8. Manage parking lots 89](#_Toc479711701)

[4.2.3.9. Check reservation PIN code 90](#_Toc479711702)

[4.3. Embedded Program Detailed Description 91](#_Toc479711703)

[4.3.1. Class Diagram 91](#_Toc479711704)

[4.3.2. Class Diagram Explanation 94](#_Toc479711705)

[4.3.2.1. [Lot] Main 94](#_Toc479711706)

[4.3.2.2. [Lot] HMC5883L 95](#_Toc479711707)

[4.3.2.3. [Lot] RGBLED 95](#_Toc479711708)

[4.3.2.4. [Lot] common\_type 96](#_Toc479711709)

[4.3.2.5. [Sign] Main 96](#_Toc479711710)

[4.3.2.6. [Sign] SEVEN\_SEGMENT 97](#_Toc479711711)

[4.3.2.7. [Lot/Sign] RFUtil 97](#_Toc479711712)

[4.3.2.8. [Lot/Sign] CRC24 98](#_Toc479711713)

[4.3.2.9. [Hub] hub 99](#_Toc479711714)

[4.3.2.10. [Hub] rf\_meta 100](#_Toc479711715)

[4.3.2.11. [Hub] crc24 101](#_Toc479711716)

[4.3.2.12. [Hub] api\_service 101](#_Toc479711717)

[4.3.2.13. [Hub] pubnub\_meta 101](#_Toc479711718)

[4.3.2.14. [Hub] Area 102](#_Toc479711719)

[4.3.2.15. [Hub] ParkingLot 102](#_Toc479711720)

[4.3.2.16. [Hub] UserPackage 103](#_Toc479711721)

[4.3.3. Interaction Diagram 103](#_Toc479711722)

[4.3.3.1. Lot node process 103](#_Toc479711723)

[4.3.3.2. Sign node process 105](#_Toc479711724)

[4.3.3.3. Hub polling process 107](#_Toc479711725)

[4.3.3.1. Hub execute request process 109](#_Toc479711726)

[4.4. Hardware Detailed Description 111](#_Toc479711727)

[4.4.1. Raspberry Pi 3 model B 111](#_Toc479711728)

[4.4.2. Arduino Nano 113](#_Toc479711729)

[4.4.3. Compass module 3-Axis HMC5883L 115](#_Toc479711730)

[4.4.4. RF module nRF24L01+ 117](#_Toc479711731)

[4.4.5. Lot node 122](#_Toc479711732)

[4.4.5.1. RGB LED common anode 122](#_Toc479711733)

[4.4.5.2. Transistor TIP122 124](#_Toc479711734)

[4.4.5.3. Servo motor SG90 126](#_Toc479711735)

[4.4.6. Sign node 127](#_Toc479711736)

[4.4.6.1. 7-Segment LED display 127](#_Toc479711737)

[4.4.6.2. Power logic 8-bit shift register TPIC6B595 127](#_Toc479711738)

[4.4.7. Components connection 130](#_Toc479711739)

[5. Interface 130](#_Toc479711740)

[5.1. Component Interface 130](#_Toc479711741)

[5.2. User Interface Design 130](#_Toc479711742)

[5.2.1. Common Interface 130](#_Toc479711743)

[5.2.1.1. Login screen 130](#_Toc479711744)

[5.2.1.2. Register screen 132](#_Toc479711745)

[5.2.2. User/Guest Interface 133](#_Toc479711746)

[5.2.2.1. Map view screen 133](#_Toc479711747)

[5.2.2.2. Navigation view 135](#_Toc479711748)

[5.2.2.3. Car park list view screen 137](#_Toc479711749)

[5.2.2.4. Car park detail view screen 138](#_Toc479711750)

[5.2.2.5. History transaction list view screen 140](#_Toc479711751)

[5.2.2.6. History transaction detail view screen 141](#_Toc479711752)

[5.2.2.7. Reservation process view screen 143](#_Toc479711753)

[5.2.3. Manager Interface 146](#_Toc479711754)

[5.2.3.1. Car park list view screen 146](#_Toc479711755)

[5.2.3.2. Car park detail view screen 147](#_Toc479711756)

[5.2.3.3. Area list view screen 149](#_Toc479711757)

[5.2.3.4. Area detail view screen 150](#_Toc479711758)

[5.2.3.5. Parking lot list view screen 151](#_Toc479711759)

[5.2.3.6. Parking lot detail view screen 152](#_Toc479711760)

[5.2.3.7. Check code view screen 153](#_Toc479711761)

[6. Database Design 155](#_Toc479711762)

[6.1. Logical Diagram 155](#_Toc479711763)

[6.2. Data Dictionary 155](#_Toc479711764)

[7. Algorithms 157](#_Toc479711765)

[7.1. GetDistance formula 157](#_Toc479711766)

[7.2. CRC Error Detection 157](#_Toc479711767)

[7.2.1. Definition 157](#_Toc479711768)

[7.2.2. Define Problem 157](#_Toc479711769)

[7.2.3. Solution theory 157](#_Toc479711770)

[7.2.4. Implementation 160](#_Toc479711771)

[7.2.5. Table-driven implementation 161](#_Toc479711772)

# List of Tables

[Table 1: Definitions, Acronyms and Abbreviations 17](#_Toc479711773)

[Table 2: Component dictionary 26](#_Toc479711774)

[Table 3: CarPark class attributes 30](#_Toc479711775)

[Table 4: CarParkWithAmountEntities class attributes 30](#_Toc479711776)

[Table 5: Area class attributes 30](#_Toc479711777)

[Table 6: AreaCustom class attributes 30](#_Toc479711778)

[Table 7: ParkingLot class attributes 31](#_Toc479711779)

[Table 8: Transaction class attributes 31](#_Toc479711780)

[Table 9: AspNetUser class attributes 32](#_Toc479711781)

[Table 10: CarParkViewModel class attributes 32](#_Toc479711782)

[Table 11: CarParkWithAmount class attributes 32](#_Toc479711783)

[Table 12: CarParkUpdateViewModel class attributes 33](#_Toc479711784)

[Table 13: AreaViewModel class attributes 33](#_Toc479711785)

[Table 14: AreaCustomViewModel class attributes 33](#_Toc479711786)

[Table 15: ParkingLotViewModel class attributes 33](#_Toc479711787)

[Table 16: ParkingLotUpdateViewModel class attributes 34](#_Toc479711788)

[Table 17: Transaction class attributes 34](#_Toc479711789)

[Table 18: TransactionCustomViewModel class attributes 34](#_Toc479711790)

[Table 19: TransactionCreateViewModel class attributes 34](#_Toc479711791)

[Table 20: TransactionCreateReturnViewModel class attributes 35](#_Toc479711792)

[Table 21: TransactionUpdateViewModel class attributes 35](#_Toc479711793)

[Table 22: AspNetUser class attributes 35](#_Toc479711794)

[Table 23: BaseRepository class methods 36](#_Toc479711795)

[Table 24: BaseService class methods 37](#_Toc479711796)

[Table 25: AreaService class methods 38](#_Toc479711797)

[Table 26: CarParkService class methods 38](#_Toc479711798)

[Table 27: ParkingLotService class methods 38](#_Toc479711799)

[Table 28: TransactionService class methods 39](#_Toc479711800)

[Table 29: Packages dictionary 44](#_Toc479711801)

[Table 30: Activities package class dictionary 46](#_Toc479711802)

[Table 31: Adapters package class dictionary 48](#_Toc479711803)

[Table 32: Helpers package class dictionary 49](#_Toc479711804)

[Table 33: Interfaces package class dictionary 50](#_Toc479711805)

[Table 34: Models package class dictionary 52](#_Toc479711806)

[Table 35: Networks package class dictionary 53](#_Toc479711807)

[Table 36: [Activities] MainActivity methods 53](#_Toc479711808)

[Table 37: [Activities] CarParkDetailActivity attributes 54](#_Toc479711809)

[Table 38: [Activities] CarParkDetailActivity methods 54](#_Toc479711810)

[Table 39: [Activities] ManagerActivity attributes 55](#_Toc479711811)

[Table 40: [Activities] ManagerActivity methods 56](#_Toc479711812)

[Table 41: [Activities] AreaListActivity attributes 56](#_Toc479711813)

[Table 42: [Activities] AreaListActivity methods 57](#_Toc479711814)

[Table 43: [Activities] ParkingLotListActivity attributes 57](#_Toc479711815)

[Table 44: [Activities] ParkingLotListActivity methods 57](#_Toc479711816)

[Table 45: [Activities] LoginActivity attributes 58](#_Toc479711817)

[Table 46: [Activities] LoginActivity methods 58](#_Toc479711818)

[Table 47: [Activities] RegisterActivity attributes 58](#_Toc479711819)

[Table 48: [Activities] RegisterActivity methods 58](#_Toc479711820)

[Table 49: [Activities] UserActivity attributes 59](#_Toc479711821)

[Table 50: [Activities] UserActivity methods 60](#_Toc479711822)

[Table 51: [Activities] CarParkListActivity attributes 61](#_Toc479711823)

[Table 52: [Activities] CarParkListActivity methods 61](#_Toc479711824)

[Table 53: [Activities] TransactionActivity attributes 61](#_Toc479711825)

[Table 54: [Activities] TransactionActivity methods 61](#_Toc479711826)

[Table 55: [Adapters] AreaAdapter attributes 62](#_Toc479711827)

[Table 56: [Adapters] AreaAdapter methods 62](#_Toc479711828)

[Table 57: [Adapters] AreaAdapter.OnItemClickListener methods 62](#_Toc479711829)

[Table 58: [Adapters] AreaAdapter.ViewHolder attributes 62](#_Toc479711830)

[Table 59: [Adapters] AreaAdapter.ViewHolder methods 62](#_Toc479711831)

[Table 60: [Adapters] CarParkAdapter attributes 63](#_Toc479711832)

[Table 61: [Adapters] CarParkAdapter methods 63](#_Toc479711833)

[Table 62: [Adapter] CarParkAdapter.OnItemClickListener methods 63](#_Toc479711834)

[Table 63: [Adapters] CarParkAdapter.ViewHolder attributes 63](#_Toc479711835)

[Table 64: [Adapters] CarParkAdapter.ViewHolder methods 64](#_Toc479711836)

[Table 65: [Adapters] ParkingLotAdapter attributes 64](#_Toc479711837)

[Table 66: [Adapters] ParkingLotAdapter methods 64](#_Toc479711838)

[Table 67: [Adapters] ParkingLotAdapter.OnItemClickListener methods 64](#_Toc479711839)

[Table 68: [Adapters] ParkingLotAdapter.ViewHolder attributes 64](#_Toc479711840)

[Table 69: [Adapters] ParkingLotAdapter.ViewHolder methods 65](#_Toc479711841)

[Table 70: [Adapters] CarParkAdvanceAdapter attributes 65](#_Toc479711842)

[Table 71: [Adapters] CarParkAdvanceAdapter methods 65](#_Toc479711843)

[Table 72: [Adapters] CarParkAdvanceAdapter.OnItemClickListener methods 65](#_Toc479711844)

[Table 73: [Adapters] CarParkAdvanceAdapter.ViewHolder attributes 66](#_Toc479711845)

[Table 74: [Adapters] CarParkAdvanceAdapter.ViewHolder methods 66](#_Toc479711846)

[Table 75: [Adapters] TransactionAdapter attributes 66](#_Toc479711847)

[Table 76: [Adapters] TransactionAdapter methods 66](#_Toc479711848)

[Table 77: [Adapters] TransactionAdapter.OnItemClickListener methods 67](#_Toc479711849)

[Table 78: [Adapters] TransactionAdapter.ViewHolder attributes 67](#_Toc479711850)

[Table 79: [Adapters] TransactionAdapter.ViewHolder methods 67](#_Toc479711851)

[Table 80: [Adapters] UserInfoWindowAdapter attributes 67](#_Toc479711852)

[Table 81: [Adapters] UserInfoWindowAdapter methods 67](#_Toc479711853)

[Table 82: [Helpers] AppDatabaseHelper attributes 68](#_Toc479711854)

[Table 83: [Helpers] AppDatabaseHelper methods 68](#_Toc479711855)

[Table 84: [Helpers] PubnubHelper attributes 69](#_Toc479711856)

[Table 85: [Helpers] PubnubHelper methods 69](#_Toc479711857)

[Table 86: [Helpers] AccountHelper attributes 69](#_Toc479711858)

[Table 87: [Helpers] AccountHelper methods 69](#_Toc479711859)

[Table 88: [Helpers] MapMarkerHelper methods 69](#_Toc479711860)

[Table 89: [Helpers] InternetHelper methods 70](#_Toc479711861)

[Table 90: [Interfaces] AreaClient methods 70](#_Toc479711862)

[Table 91: [Interfaces] TransactionClient methods 70](#_Toc479711863)

[Table 92: [Interfaces] CarParkClient methods 71](#_Toc479711864)

[Table 93: [Interfaces] AccountClient methods 71](#_Toc479711865)

[Table 94: [Interfaces] ParkingLotClient methods 71](#_Toc479711866)

[Table 95: [Models] Account attributes 72](#_Toc479711867)

[Table 96: [Models] Account methods 72](#_Toc479711868)

[Table 97: [Models] Area attributes 72](#_Toc479711869)

[Table 98: [Models] Area methods 72](#_Toc479711870)

[Table 99: [Models] CheckCode attributes 72](#_Toc479711871)

[Table 100: [Models] CheckCode methods 72](#_Toc479711872)

[Table 101: [Models] Geo attributes 73](#_Toc479711873)

[Table 102: [Models] Geo methods 73](#_Toc479711874)

[Table 103: [Models] CarPark attributes 73](#_Toc479711875)

[Table 104: [Models] CarPark methods 73](#_Toc479711876)

[Table 105: [Models] Transaction attributes 74](#_Toc479711877)

[Table 106: [Models] Transaction methods 74](#_Toc479711878)

[Table 107: [Models] ParkingLot attributes 74](#_Toc479711879)

[Table 108: [Models] ParkingLot methods 74](#_Toc479711880)

[Table 109: [Models] TransactionStatus attributes 75](#_Toc479711881)

[Table 110: [Models] TransactionStatus methods 75](#_Toc479711882)

[Table 111: [Models] AreaStatus attributes 75](#_Toc479711883)

[Table 112: [Models] AreaStatus methods 75](#_Toc479711884)

[Table 113: [Models] ParkingLotStatus attributes 76](#_Toc479711885)

[Table 114: [Models] ParkingLotStatus methods 76](#_Toc479711886)

[Table 115: [Network] AccountPackage attributes 76](#_Toc479711887)

[Table 116: [Network] AccountPackage methods 76](#_Toc479711888)

[Table 117: [Network] AreaPackage attributes 76](#_Toc479711889)

[Table 118: [Network] AreaPackage methods 76](#_Toc479711890)

[Table 119: [Network] CarParkPackage attributes 77](#_Toc479711891)

[Table 120: [Network] CarParkPackage methods 77](#_Toc479711892)

[Table 121: [Network] CheckCodePackage attributes 77](#_Toc479711893)

[Table 122: [Network] CheckCodePackage methods 77](#_Toc479711894)

[Table 123: [Network] CommandPackage attributes 77](#_Toc479711895)

[Table 124: [Network] CommandPackage methods 78](#_Toc479711896)

[Table 125: [Network] RealtimeMapPackage attributes 78](#_Toc479711897)

[Table 126: [Network] RealtimeMapPackage methods 78](#_Toc479711898)

[Table 127: [Network] ControlPubnubPackage attributes 78](#_Toc479711899)

[Table 128: [Network] ControlPubnubPackage methods 78](#_Toc479711900)

[Table 129: [Network] MobilePubnubPackage attributes 78](#_Toc479711901)

[Table 130: [Network] MobilePubnubPackage methods 79](#_Toc479711902)

[Table 131: [Network] GetCoordinationPackage attributes 79](#_Toc479711903)

[Table 132: [Network] GetCoordinationPackage methods 79](#_Toc479711904)

[Table 133: [Network] CarParkAdvancePackage attributes 79](#_Toc479711905)

[Table 134: [Network] CarParkAdvancePackage methods 79](#_Toc479711906)

[Table 135: [Network] NotificationPackage attributes 80](#_Toc479711907)

[Table 136: [Network] NotificationPackage methods 80](#_Toc479711908)

[Table 137: [Network] TransactionPackage attributes 80](#_Toc479711909)

[Table 138: [Network] TransactionPackage methods 80](#_Toc479711910)

[Table 139: [Network] ParkingLotPackage attributes 80](#_Toc479711911)

[Table 140: [Network] ParkingLotPackage methods 80](#_Toc479711912)

[Table 141: [Network] ServiceGenerator attributes 81](#_Toc479711913)

[Table 142: [Network] ServiceGenerator methods 81](#_Toc479711914)

[Table 143: [Network] DateTypeDeserializer attributes 81](#_Toc479711915)

[Table 144: [Network] DateTypeDeserializer methods 81](#_Toc479711916)

[Table 145: Lot node class dictionary 92](#_Toc479711917)

[Table 146: Sign node class dictionary 92](#_Toc479711918)

[Table 147: Hub class dictionary 94](#_Toc479711919)

[Table 148: [Lot] Main attributes 94](#_Toc479711920)

[Table 149: [Lot] Main methods 94](#_Toc479711921)

[Table 150: [Lot] HMC5883L attributes 95](#_Toc479711922)

[Table 151: [Lot] HMC5883L methods 95](#_Toc479711923)

[Table 152: [Lot] RGBLED attributes 95](#_Toc479711924)

[Table 153: [Lot] RGBLED methods 96](#_Toc479711925)

[Table 154: [Lot] common\_type attributes 96](#_Toc479711926)

[Table 155: [Sign] Main attributes 96](#_Toc479711927)

[Table 156: [Sign] Main methods 97](#_Toc479711928)

[Table 157: [Sign] SEVEN\_SEGMENT attributes 97](#_Toc479711929)

[Table 158: [Sign] SEVEN\_SEGMENT methods 97](#_Toc479711930)

[Table 159: [Lot/Sign] RFUtil attributes 98](#_Toc479711931)

[Table 160: [Lot/Sign] RFUtil methods 98](#_Toc479711932)

[Table 161:: [Lot/Sign] CRC24 attributes 98](#_Toc479711933)

[Table 162: [Lot/Sign] CRC24 methods 98](#_Toc479711934)

[Table 163: [Hub] hub attributes 99](#_Toc479711935)

[Table 164: [Hub] hub methods 100](#_Toc479711936)

[Table 165: [Hub] rf\_meta attributes 100](#_Toc479711937)

[Table 166: [Hub] rf\_meta methods 101](#_Toc479711938)

[Table 167: [Hub] crc24 attributes 101](#_Toc479711939)

[Table 168: [Hub] crc24 methods 101](#_Toc479711940)

[Table 169: [Hub] api\_service methods 101](#_Toc479711941)

[Table 170: [Hub] pubnub\_meta attributes 102](#_Toc479711942)

[Table 171: [Hub] pubnub\_meta methods 102](#_Toc479711943)

[Table 172: [Hub] Area attributes 102](#_Toc479711944)

[Table 173: [Hub] ParkingLot attributes 103](#_Toc479711945)

[Table 174: [Hub] UserPackage attributes 103](#_Toc479711946)

[Table 175: HMC55883L Register list 117](#_Toc479711947)

[Table 176: nRF24L01+ pinout 119](#_Toc479711948)

[Table 177: nRF24L01+ SPI commands 121](#_Toc479711949)

[Table 178: Servo motor SG90 specifications 126](#_Toc479711950)

[Table 179: Servo motor SG90 pinout 127](#_Toc479711951)

[Table 180: Power logic 8-bit shift register TPIC6B595 pinout 129](#_Toc479711952)

[Table 181: Login screen fields 131](#_Toc479711953)

[Table 182: Login screen button/hyperlinks 131](#_Toc479711954)

[Table 183: Register screen fields 133](#_Toc479711955)

[Table 184: Register screen button/hyperlinks 133](#_Toc479711956)

[Table 185: Map view screen fields 134](#_Toc479711957)

[Table 186: Map view screen button/hyperlinks 134](#_Toc479711958)

[Table 187: Navigation view fields 135](#_Toc479711959)

[Table 188: Navigation view button/hyperlinks 136](#_Toc479711960)

[Table 189: Car park list view screen fields 138](#_Toc479711961)

[Table 190: Car park list view screen button/hyperlinks 138](#_Toc479711962)

[Table 191: Car park detail view screen fields 139](#_Toc479711963)

[Table 192: Car park detail view screen button/hyperlinks 139](#_Toc479711964)

[Table 193: History transaction list view screen fields 141](#_Toc479711965)

[Table 194: History transaction list view screen button/hyperlinks 141](#_Toc479711966)

[Table 195: History transaction detail view screen fields 142](#_Toc479711967)

[Table 196: History transaction detail view screen button/hyperlinks 142](#_Toc479711968)

[Table 197: Reservation process view screen 1 fields 143](#_Toc479711969)

[Table 198: Reservation process view screen 1 button/hyperlinks 144](#_Toc479711970)

[Table 199: Reservation process view screen 2 fields 144](#_Toc479711971)

[Table 200: Reservation process view screen 2 button/hyperlinks 145](#_Toc479711972)

[Table 201: Reservation process notification fields 145](#_Toc479711973)

[Table 202: Car park list view screen fields 147](#_Toc479711974)

[Table 203: Login screen button/hyperlinks 147](#_Toc479711975)

[Table 204: Car park detail view screen fields 148](#_Toc479711976)

[Table 205: Car park detail view screen button/hyperlinks 148](#_Toc479711977)

[Table 206: Area list view screen fields 149](#_Toc479711978)

[Table 207: Area list view screen button/hyperlinks 150](#_Toc479711979)

[Table 208: Area detail view screen fields 150](#_Toc479711980)

[Table 209: Area detail view screen button/hyperlinks 151](#_Toc479711981)

[Table 210: Parking lot list view screen fields 152](#_Toc479711982)

[Table 211: Parking lot list view screen button/hyperlinks 152](#_Toc479711983)

[Table 212: Parking lot detail view screen fields 153](#_Toc479711984)

[Table 213: Parking lot detail view screen button/hyperlinks 153](#_Toc479711985)

[Table 214: Check code screen fields 154](#_Toc479711986)

[Table 215: Check code screen button/hyperlinks 154](#_Toc479711987)

[Table 216: Database data dictionary 156](#_Toc479711988)

# List of Figures

[Figure 1: System architecture design 19](#_Toc479711989)

[Figure 2: Hardware Program architectural 20](#_Toc479711990)

[Figure 3: Mobile Application architecture 22](#_Toc479711991)

[Figure 4: Hardware - Software Connection Architecture Description 23](file:///C:\Users\TrungTNM\Documents\-FPT-CAPSTONE-PGSS\Common\Reports\Report%204.docx#_Toc479711992)

[Figure 5: General Component Diagram 23](#_Toc479711993)

[Figure 6: Lot Device Diagram 24](#_Toc479711994)

[Figure 7: Information Device Diagram 24](#_Toc479711995)

[Figure 8: Server Diagram 25](#_Toc479711996)

[Figure 9: Model Class Diagram 27](#_Toc479711997)

[Figure 10: Vỉew Model Class Diagram 28](#_Toc479711998)

[Figure 11: Repository Class Diagram 29](#_Toc479711999)

[Figure 12: Service Class Diagram 29](#_Toc479712000)

[Figure 13: Register Diagram 39](#_Toc479712001)

[Figure 14: Login Diagram 40](#_Toc479712002)

[Figure 15: GetCarPark Diagram 40](#_Toc479712003)

[Figure 16: GetArea Diagram 41](#_Toc479712004)

[Figure 17: GetParkingLot Diagram 41](#_Toc479712005)

[Figure 18: UpdateArea Diagram 42](#_Toc479712006)

[Figure 19: UpdateParkingLot Diagram 42](#_Toc479712007)

[Figure 20: UpdateListStatusParkingLot Diagram 43](#_Toc479712008)

[Figure 21: Main class diagram and package view 43](#_Toc479712009)

[Figure 22: Activities package class diagram 45](#_Toc479712010)

[Figure 23: Adapters package class diagram 47](#_Toc479712011)

[Figure 24: Helpers package class diagram 49](#_Toc479712012)

[Figure 25: Interfaces package class diagram 50](#_Toc479712013)

[Figure 26: Models package class diagram 51](#_Toc479712014)

[Figure 27: Networks package class diagram 52](#_Toc479712015)

[Figure 28: Interaction diagram - Load car parks on map 82](#_Toc479712016)

[Figure 29: Interaction diagram - Load car parks in list 83](#_Toc479712017)

[Figure 30: Interaction diagram - Car park detail information 84](#_Toc479712018)

[Figure 31: Interaction diagram - Reserve parking lot process 85](#_Toc479712019)

[Figure 32: Interaction diagram - Transaction controller 86](#_Toc479712020)

[Figure 33: Interaction diagram - Manage car parks 87](#_Toc479712021)

[Figure 34: Interaction diagram - Manage areas 88](#_Toc479712022)

[Figure 35: Interaction diagram - Manage parking lots 89](#_Toc479712023)

[Figure 36: Interaction diagram - Check reservation PIN code 90](#_Toc479712024)

[Figure 37: Lot node class diagram 91](#_Toc479712025)

[Figure 38: Sign node class diagram 92](#_Toc479712026)

[Figure 39: Hub class diagram 93](#_Toc479712027)

[Figure 40: Interaction diagram - Lot node process 104](#_Toc479712028)

[Figure 41: Interaction diagram - Sign node 106](#_Toc479712029)

[Figure 42: Interaction diagram - Hub polling process 108](#_Toc479712030)

[Figure 43: Interaction diagram - Hub request process 110](#_Toc479712031)

[Figure 44: Raspberry Pi 3 model B 111](#_Toc479712032)

[Figure 45: Raspberry Pi 3 Model B pinount 112](#_Toc479712033)

[Figure 46: Arduino Nano 113](#_Toc479712034)

[Figure 47: Arduino Nano specifications 114](#_Toc479712035)

[Figure 48: Arduino Nano pinout 114](#_Toc479712036)

[Figure 49: 3-Axis HMC5883L 115](#_Toc479712037)

[Figure 50: RF module nRF24L01+ 117](#_Toc479712038)

[Figure 51: nRF24L01+ pinout 118](#_Toc479712039)

[Figure 52: nRF24L01+ packet format 119](#_Toc479712040)

[Figure 53: RGB LED common anode 122](#_Toc479712041)

[Figure 54: RGB LED common anode pinout 123](#_Toc479712042)

[Figure 55: Transistor TIP122 124](#_Toc479712043)

[Figure 56: Transistor TIP122 pinout 125](#_Toc479712044)

[Figure 57: Servo motor SG90 126](#_Toc479712045)

[Figure 58: Power logic 8-bit shift register TPIC6B595 127](#_Toc479712046)

[Figure 59: Power logic 8-bit shift register TPIC6B595 pinout 128](#_Toc479712047)

[Figure 60: Login screen 130](#_Toc479712048)

[Figure 61: Register screen 132](#_Toc479712049)

[Figure 62: Map view screen 133](#_Toc479712050)

[Figure 63: Navigation view 135](#_Toc479712051)

[Figure 64: Car park list view screen 137](#_Toc479712052)

[Figure 65: Car park detail view screen 138](#_Toc479712053)

[Figure 66: History transaction list view screen 140](#_Toc479712054)

[Figure 67: History transaction detail view screen 141](#_Toc479712055)

[Figure 68: Reservation process view screen 1 143](#_Toc479712056)

[Figure 69: Reservation process view screen 2 144](#_Toc479712057)

[Figure 70: Reservation process notification 145](#_Toc479712058)

[Figure 71: Car park list view screen 146](#_Toc479712059)

[Figure 72: Car park detail view screen 147](#_Toc479712060)

[Figure 73: Area list view screen 149](#_Toc479712061)

[Figure 74: Area detail view screen 150](#_Toc479712062)

[Figure 75: Parking lot list view screen 151](#_Toc479712063)

[Figure 76: Parking lot detail view screen 152](#_Toc479712064)

[Figure 77: Check code view screen 153](#_Toc479712065)

[Figure 78: Database logical diagram 155](#_Toc479712066)

[Figure 79: CRC arithmetic addition 158](#_Toc479712067)

[Figure 80: CRC arithmetic subtraction 158](#_Toc479712068)

[Figure 81: CRC arithmetic multiplication 158](#_Toc479712069)

[Figure 82: CRC arithmetic division 159](#_Toc479712070)

[Figure 83: CRC implementation register 160](#_Toc479712071)

[Figure 84: CRC implementation simple flow 161](#_Toc479712072)

[Figure 85: CRC 32-bits register 162](#_Toc479712073)

[Figure 86: CRC implementation table-driven flow 163](#_Toc479712074)

[Figure 87: CRC24 precomputed table 164](#_Toc479712075)

# Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| PGS | Parking Guidance System |
| Parking area | An area set aside for parking vehicles, aircraft, etc. |
| Parking lot | A place inside parking area that provide space for one vehicle |
| IoT | Internet of Things |
| CCU | Central Control Unit |
| ASIC | Application-specific integrated circuit |
| LCC | leadless chip carrier |

Table : Definitions, Acronyms and Abbreviations

# D. Software – Hardware Design Description

## Design Overview

This document describes the technical and user interface design of **PGSS.** It includes the architectural design, the detailed design of common functions and business functions and the design of database model.

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use case.

The database design describes the relationships between entities and details of each entity.

Document overview:

* Section 2: gives an overall description of the system architecture design.
* Section 3: gives component diagrams that describe the connection and integration of the system.
* Section 4: gives the detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: describe screen design.
* Section 6: describe a fully attributed ERD.
* Section 7: describe algorithms.

## System Architectural Design

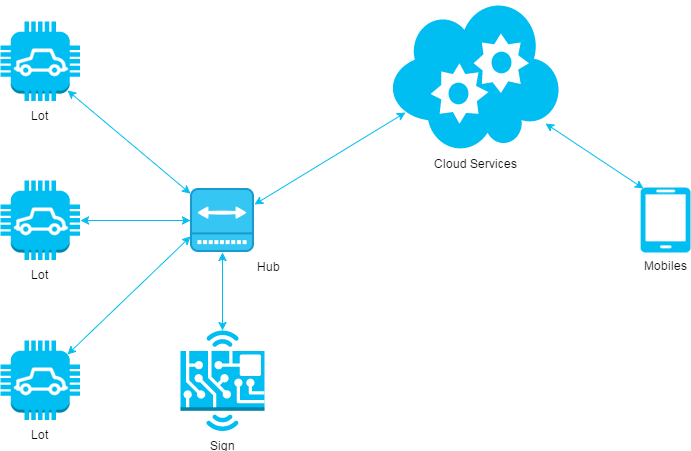


Figure : System architecture design

This diagram is referenced and modified from an original concept from: The Internet of Things: An Overview by The Internet Society (ISOC). In this device-to-gateway model, or more typically, the device-to-application-layer gateway (ALG) model, the IoT device connects through an ALG service as a conduit to reach a cloud service. The hub serves as a local gateway between individual IoT devices and a cloud service, but they can also bridge the interoperability gap between devices themselves.

### Hardware Program Architecture Description



Figure : Hardware Program architectural

**Parking Lot Node** is a device placed at each lot in the car park. It plays the role as a passive IoT node and controls other components, which connects to it through several kinds of port to perform various works:

* **Arduino Nano**: plays the roles as central processing unit for IoT node.
* **Magnetometer**: an instrument that measures magnetism, which is used as metal detectors to detect ferrous metals in a car.
* **Servo**: a linear actuator that allows for precise control of angular or linear position, velocity and acceleration. This will be used to control the barrier in each lot.
* **LED**: a RGB LED used as an Indicator LED in each lot.
* **RF Module**: a small device used to transmit/receive radio signals between two devices. This is used to create a network bridge for the whole system.

**Sign Node** is a device placed at each area and at the entrance of the car park. It also a passive node and used to display the available lots in each area:

* **Arduino Nano**: plays the roles as central processing unit for IoT node.
* **RF Module**: a small device used to transmit/receive radio signals between two devices. This is used to create a network bridge for the whole system.
* **LED:** a large LED display screen

**Hub** is a stand-alone gateway device that has RF Module installed to communicate with all other IoT nodes. It also connects to cloud service, allowing the user to gain access to the devices using a smartphone app and an Internet connection:

* **Raspberry Pi**: plays the roles as central processing unit for Hub node.
* **RF Module**: a small device used to transmit/receive radio signals between two devices. This is used to create a network bridge for the whole system.
* **Network Interfaces:** an interfaces to connect to the Internet.
* **API Service:** is a cloud service that provides all necessary APIs and database.
* **Real time Service:** a cloud service that provides real time communication between smartphone app and Hub.

### Mobile Application Architecture Description

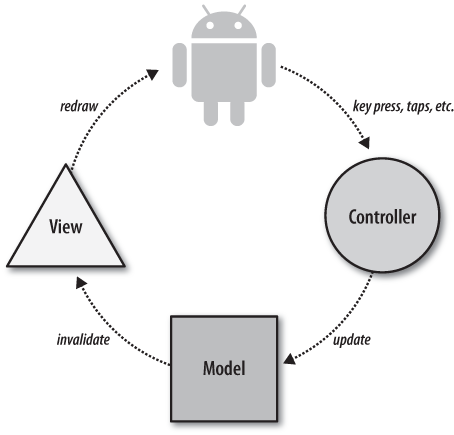


Figure : Mobile Application architecture

In Mobile Application, the system is developed under Standard Android Architecture. This is the default approach with layout files, Activities/Fragments acting as the controller and Models used for data and persistence. With this approach, Activities are in charge of processing the data and updating the views. Activities act like a controller in MVC but with some extra responsibilities that should be part of the view.

### Web API Architecture Description

Figure : Hardware - Software Connection Architecture Description

## Component Diagram

### General Component Diagram

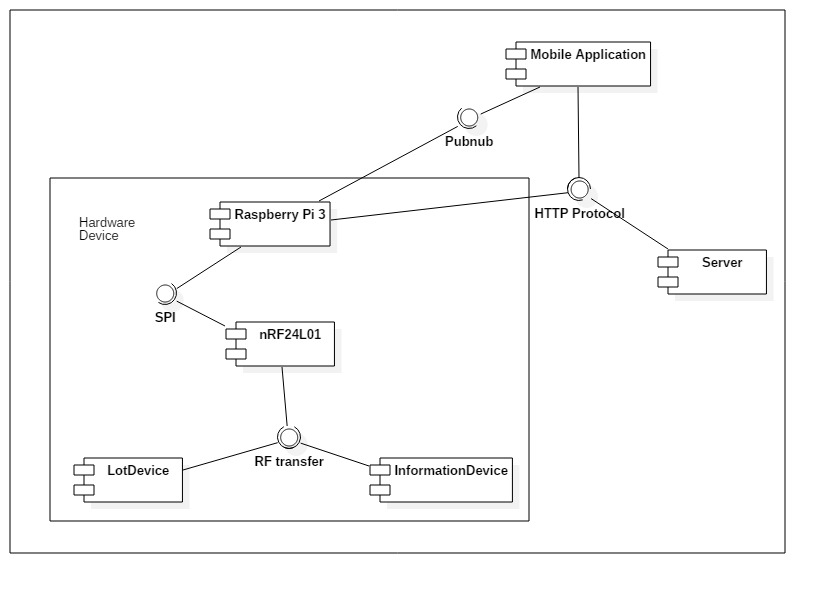


Figure : General Component Diagram

### Lot Device Diagram

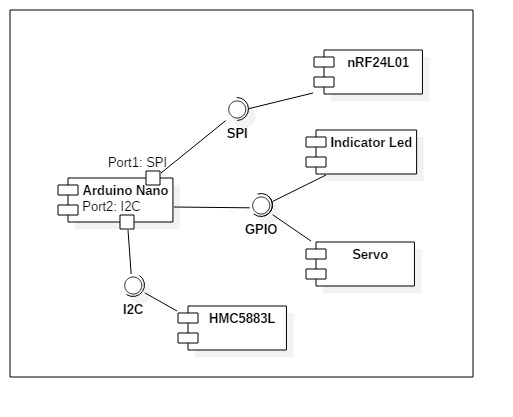


Figure : Lot Device Diagram

### Information Device Diagram

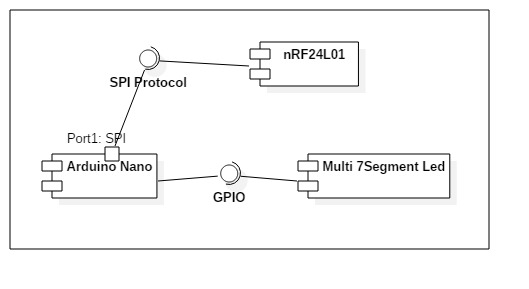


Figure : Information Device Diagram

### Server Diagram

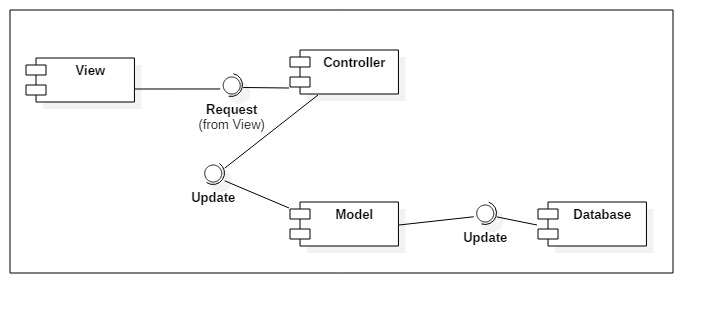


Figure : Server Diagram

### Component Dictionary

|  |  |
| --- | --- |
| Component Dictionary: Describes components | |
| **Mobile Application** | The application for user on android devices |
| **Server** | The Web API server, which make the interaction between the Raspberry Pi and the Mobile Application. It also save information to the database. |
| **Raspberry Pi 3** | The main controller in the hardware system that control all Lot Device and Information Device in each car park. |
| **Lot Device** | The device in each parking lot that control the indicator led, servo and use the magnetometer sensor to detect car. |
| **Information Device** | The number led board that show the number of empty lot in each area and number of empty lot in car park. |
| **nRF24L01** | The RF module, which help main controller, lot device and information device interact with each other. |
| **Arduino Nano** | The controller in lot device and information device. |
| **Indicator Led** | The RGB led to show the status of parking lot (empty, in used, reserved, disabled). |
| **Servo** | The barrier to prevent other car get in the parking lot when it is reserved. |
| **HMC5883L** | The magnetometer sensor to realize if the metal is near. |
| **Pubnub** | The third party API that help real time interaction. |
| **HTTP Protocol** | A transfer protocol to interact between server, mobile application and Raspberry Pi 3. |
| **SPI** | Synchronous serial communication interface used for short distance communication. |
| **I2C** | The inter-intergrated circuit to communicate between arduino and magnetometer sensor. |
| **GPIO** | The communication between Arduino and led, servo, … |
| **Multi 7Segment Led** | The 7 segment led to indicate number. |
| **View** | The JSON result of the API controller. |
| **Controller** | The API controller in the server. |
| **Model** | The entity model. |
| **Database** | The collection of data. |

Table : Component dictionary

## Detailed Description

### Web API Detailed Description

#### Class Diagram

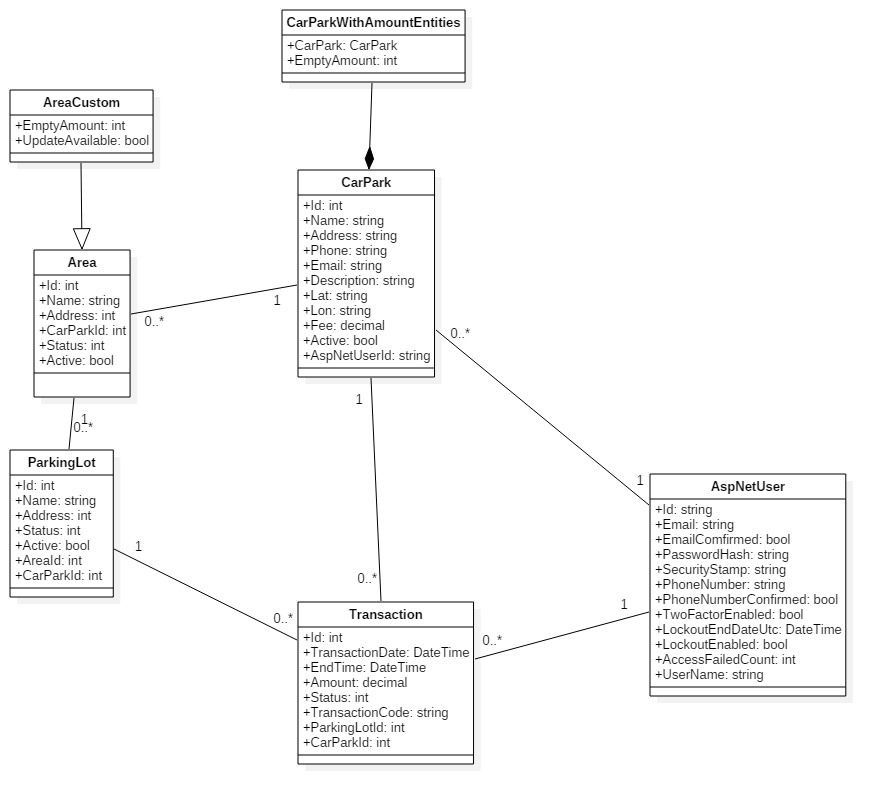


Figure : Model Class Diagram

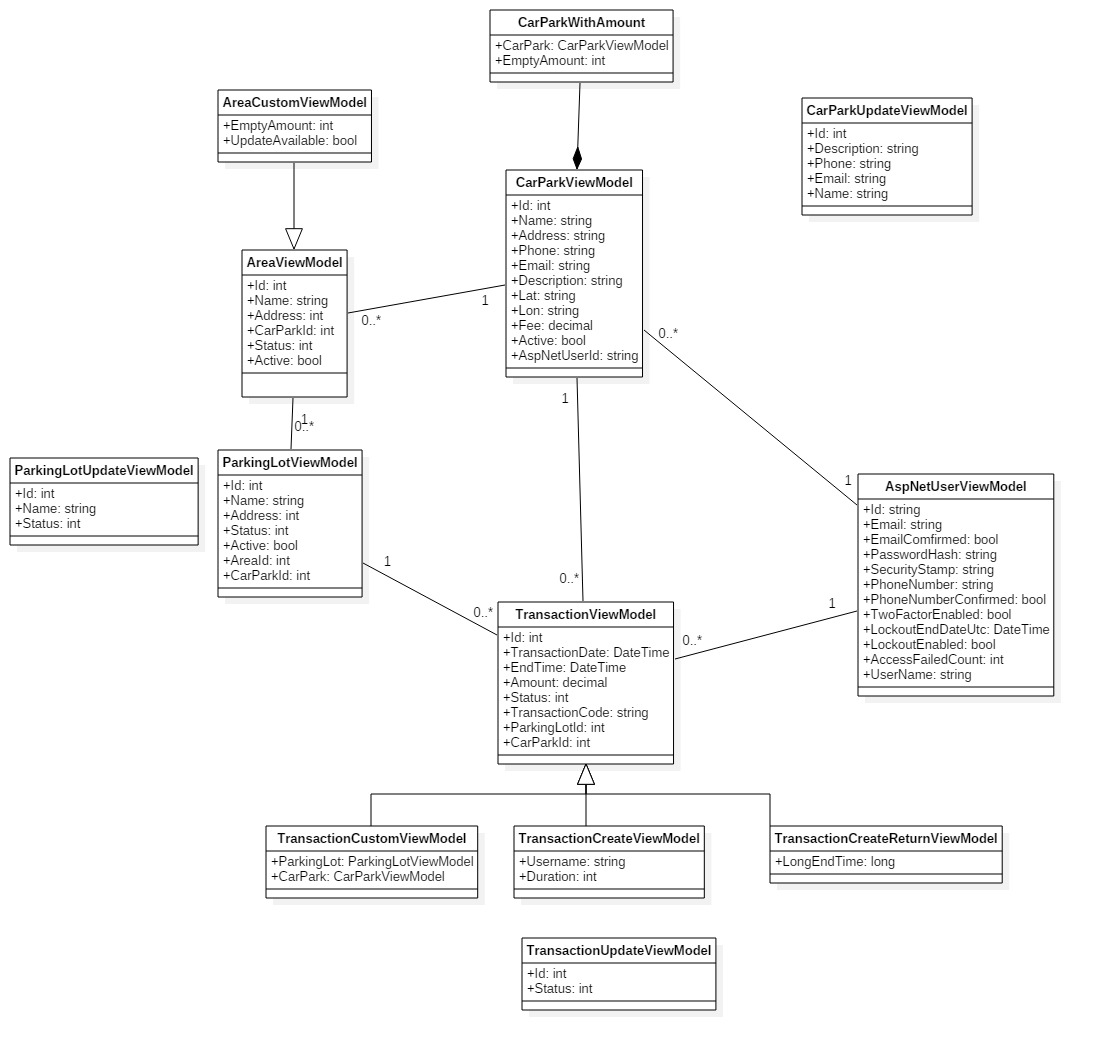


Figure : Vỉew Model Class Diagram

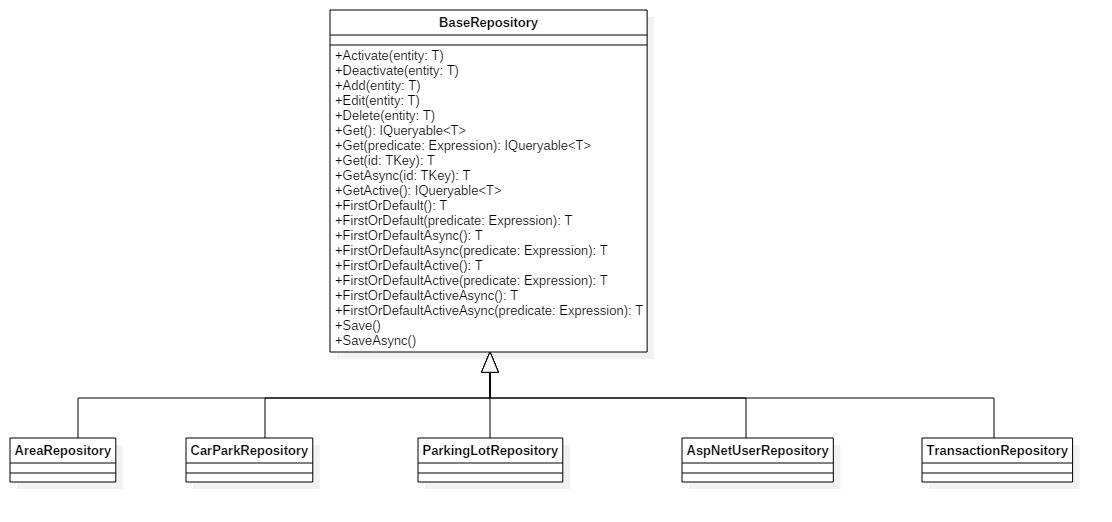


Figure : Repository Class Diagram

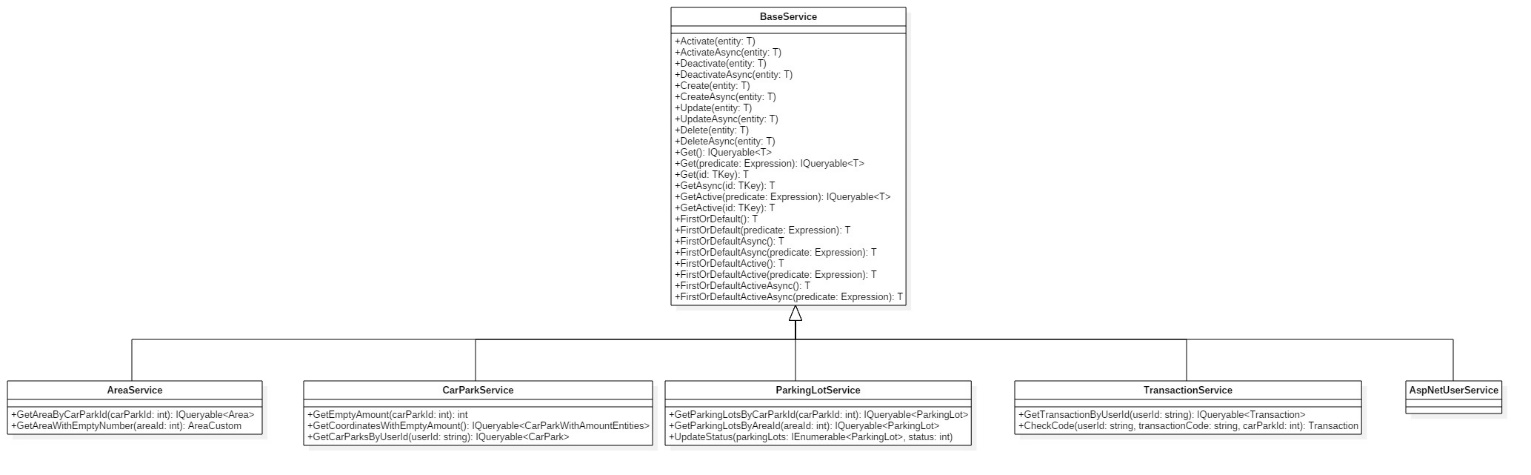


Figure : Service Class Diagram

#### Class Diagram Explanation

##### Model Class Diagram

**CarPark**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each car park |
| Name | string | public | Name of car park |
| Address | string | public | Address of carpark |
| Phone | string | public | Phone number of car park |
| Email | string | public | Email of car park’s owner |
| Description | string | public | Information of car park |
| Lat | string | public | Latitude of a car park |
| Lon | string | public | Longitude of a car park |
| Fee | decimal | public | Amount of money each hour |
| Active | bool | public | Value show that the car park is deleted or not |
| AspNetUserId | string | public | Id of the owner |

Table : CarPark class attributes

**CarParkWithAmountEntities**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| CarPark | CarPark | public | The car park entity |
| EmptyAmount | int | public | The number of empty lot in car park |

Table : CarParkWithAmountEntities class attributes

**Area**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each area |
| Name | string | public | Name of area |
| Address | int | public | Bit address of a area |
| CarParkId | int | public | Id of car park which hold area |
| Status | int | public | The number indicate the status of area |
| Active | bit | public | Value show that the area is deleted or not |

Table : Area class attributes

**AreaCustom**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| EmptyAmount | int | public | The number of empty lot in area |
| UpdateAvailable | bool | public | The boolean value check that area is available for update |

Table : AreaCustom class attributes

**ParkingLot**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each parking lot |
| Name | string | public | Name of parking lot |
| Address | int | public | Bit address of a parking lot |
| Status | int | public | The number indicate the status of parking lot |
| Active | bit | public | Value show that the parking lot is deleted or not |
| AreaId | int | public | Id of area which hold parking lot |
| CarParkId | int | public | Id of car park which hold parking lot |

Table : ParkingLot class attributes

**Transaction**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each transaction |
| TransactionDate | string | public | The time customer book the parking lot |
| EndTime | int | public | End of the booking time |
| Amount | int | public | The total fee of transaction |
| Status | bit | public | The number indicate status of the transaction |
| TransactionCode | string | public | The code for the user to enter in the car park to unlock the booked parking lot |
| ParkingLotId | int | public | Id of area which hold parking lot |
| CarParkId | int | public | Id of car park which hold parking lot |

Table : Transaction class attributes

**AspNetUser**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | string | public | Unique Id for each user |
| Email | string | public | Email of the user |
| EmailConfirm | bool | public | Check if the user confirm the email |
| PasswordHash | string | public | Hash of the user password, don’t save exact password |
| SecurityStamp | string | public |  |
| PhoneNumber | string | public | The number of the user |
| PhoneNumberConfirmed | bool | public | Check if the number is confirmed |
| TwoFactorEnabled | bool | public |  |
| LockoutEndDateUtc | DateTime | public |  |
| LockoutEnabled | bool | public |  |
| AccessFailedCount | int | public | The number of time access failed |
| UserName | string | public | Username of the user |

Table : AspNetUser class attributes

##### View Model Class Diagram

**CarParkViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each car park |
| Name | string | public | Name of car park |
| Address | string | public | Address of carpark |
| Phone | string | public | Phone number of car park |
| Email | string | public | Email of car park’s owner |
| Description | string | public | Information of car park |
| Lat | string | public | Latitude of a car park |
| Lon | string | public | Longitude of a car park |
| Fee | decimal | public | Amount of money each hour |
| Active | bool | public | Value show that the car park is deleted or not |
| AspNetUserId | string | public | Id of the owner |

Table : CarParkViewModel class attributes

**CarParkWithAmount**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| CarPark | CarParkViewModel | public | The car park entity |
| EmptyAmount | int | public | The number of empty lot in car park |

Table : CarParkWithAmount class attributes

**CarParkUpdateViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Id of update car park |
| Description | string | public | New description |
| Phone | string | public | New phone number |
| Email | string | public | New email |
| Name | string | public | New name |

Table : CarParkUpdateViewModel class attributes

**AreaViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each area |
| Name | string | public | Name of area |
| Address | int | public | Bit address of a area |
| CarParkId | int | public | Id of car park which hold area |
| Status | int | public | The number indicate the status of area |
| Active | bit | public | Value show that the area is deleted or not |

Table : AreaViewModel class attributes

**AreaCustomViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| EmptyAmount | int | public | The number of empty lot in area |
| UpdateAvailable | bool | public | The boolean value check that area is available for update |

Table : AreaCustomViewModel class attributes

**ParkingLotViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each parking lot |
| Name | string | public | Name of parking lot |
| Address | int | public | Bit address of a parking lot |
| Status | int | public | The number indicate the status of parking lot |
| Active | bit | public | Value show that the parking lot is deleted or not |
| AreaId | int | public | Id of area which hold parking lot |
| CarParkId | int | public | Id of car park which hold parking lot |

Table : ParkingLotViewModel class attributes

**ParkingLotUpdateViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each parking lot |
| Name | string | public | New name if change |
| Status | int | public | New status if change |

Table : ParkingLotUpdateViewModel class attributes

**Transaction**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Unique Id for each transaction |
| TransactionDate | string | public | The time customer book the parking lot |
| EndTime | int | public | End of the booking time |
| Amount | int | public | The total fee of transaction |
| Status | bit | public | The number indicate status of the transaction |
| TransactionCode | string | public | The code for the user to enter in the car park to unlock the booked parking lot |
| ParkingLotId | int | public | Id of area which hold parking lot |
| CarParkId | int | public | Id of car park which hold parking lot |

Table : Transaction class attributes

**TransactionCustomViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| ParkingLot | ParkingLotViewModel | public | ParkingLot’s model |
| CarPark | CarParkViewModel | public | CarPark, where transaction is booked |

Table : TransactionCustomViewModel class attributes

**TransactionCreateViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Username | string | public | Username of the customer |
| Duration | int | public | Time of booking |

Table : TransactionCreateViewModel class attributes

**TransactionCreateReturnViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| LongEndTime | long | public | DateTime in number format |

Table : TransactionCreateReturnViewModel class attributes

**TransactionUpdateViewModel**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | int | public | Id of the transaction |
| Status | int | public | The integer indicate the status of transaction |

Table : TransactionUpdateViewModel class attributes

**AspNetUser**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Visibility | Description |
| Id | string | public | Unique Id for each user |
| Email | string | public | Email of the user |
| EmailConfirm | bool | public | Check if the user confirm the email |
| PasswordHash | string | public | Hash of the user password, don’t save exact password |
| SecurityStamp | string | public |  |
| PhoneNumber | string | public | The number of the user |
| PhoneNumberConfirmed | bool | public | Check if the number is confirmed |
| TwoFactorEnabled | bool | public |  |
| LockoutEndDateUtc | DateTime | public |  |
| LockoutEnabled | bool | public |  |
| AccessFailedCount | int | public | The number of time access failed |
| UserName | string | public | Username of the user |

Table : AspNetUser class attributes

##### Repository Class Diagram

**BaseRepository**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Type** | **Visibility** | **Description** |
| Activate | void | public | Set the active attribute of current model to True |
| Deactivate | void | public | Set the active attribute of current model to False |
| Add | void | public | Add a model to database |
| Edit | void | public | Edit a input model |
| Delete | void | public | Delete a input model in database |
| Get | IQueryable<T> | public | Get a list of model with condition in database |
| Get | T | public | Get a model by Id in database |
| GetAsync | T | public | Get a model by Id in database asynchronous |
| GetActive | IQueryable<T> | public | Get a list of model with condition in database asynchronous |
| FirstOrDefault | T | public | Get first model in a list with condition in database |
| FirstOrDefaultAsync | T | public | Get first model in a list with condition in database asynchronous |
| FirstOrDefaultActive | T | public | Get first model in a list of active item with condition in database |
| FirstOrDefaultActiveAsync | T | public | Get first model in a list of active item with condition in database asynchronous |
| Save | void | public | Save a current state of entity to database |
| SaveAsync | void | public | Save a current state of entity to database asynchronous |

Table : BaseRepository class methods

##### Service Class Diagram

**Base Service**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Type** | **Visibility** | **Description** |
| Activate | void | public | Set the active attribute of current model to True |
| ActivateAsync | void | public | Set the active attribute of current model to True asynchronous |
| Deactivate | void | public | Set the active attribute of current model to False |
| DeactivateAsync |  |  | Set the active attribute of current model to False asynchronous |
| Create | void | public | Create a new item in database |
| CreateAsync | void | public | Create a new item in database asynchronous |
| Update | void | public | Update a item in database |
| UpdateAsync | void | public | Update an item in database asynchronous |
| Delete | void | public | Delete an item in database |
| DeleteAsync | void | public | Delete an item in database asynchronous |
| Get | IQueryable<T> | public | Get a list of model with condition in database |
| Get | T | public | Get a model by Id in database |
| GetAsync | T | public | Get a model by Id in database asynchronous |
| GetActive | IQueryable<T> | public | Get a list of model with condition in database asynchronous |
| GetActive | T | public | Get active model by Id in database if exist |
| FirstOrDefault | T | public | Get first model in a list with condition in database |
| FirstOrDefaultAsync | T | public | Get first model in a list with condition in database asynchronous |
| FirstOrDefaultActive | T | public | Get first model in a list of active item with condition in database |
| FirstOrDefaultActiveAsync | T | public | Get first model in a list of active item with condition in database asynchronous |

Table : BaseService class methods

**AreaService**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Type** | **Visibility** | **Description** |
| GetAreaByCarParkId | IQueryable<Area> | public | Get the area with input car park id |
| GetAreaWithEmptyNumber | AreaCustom | public | Get the area with empty number of lot by area id |

Table : AreaService class methods

**CarParkService**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Type** | **Visibility** | **Description** |
| GetEmptyAmount | int | public | Get the number of empty lot by car park id |
| GetCoordinates  WithEmptyAmount | IQueryable  <CarParkWith  AmountEntities> | public | Get the list of car park with coordinate and number of empty lot |
| GetCarParksByUserId | IQueryable  <CarPark> | public | Get the list of car park by AspNetUserId (owner of car park) |

Table : CarParkService class methods

**ParkingLotService**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Type** | **Visibility** | **Description** |
| GetParkingLotsByCarParkId | IQueryable<ParkingLot> | public | Get the list of parking lot in the car park |
| GetParkingLotsByAreaId | IQueryable<ParkingLot> | public | Get the list of parking lot in the area |
| UpdateStatus | void | public | Change the status of a list parking lot |

Table : ParkingLotService class methods

**Transaction Service**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Type** | **Visibility** | **Description** |
| GetTransactionByUserId | IQueryable<Transaction> | public | Get the list of transaction create by a user |
| CheckCode | Transaction | public | Get the transaction base on user, transaction code and car park |

Table : TransactionService class methods

#### Interaction Diagram

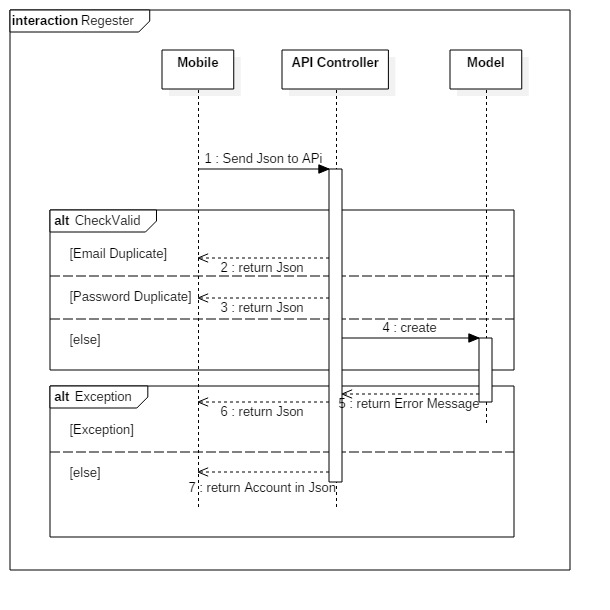


Figure : Register Diagram

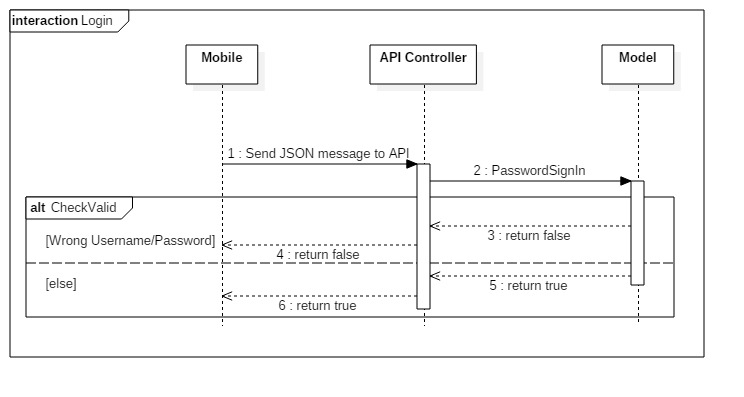


Figure : Login Diagram

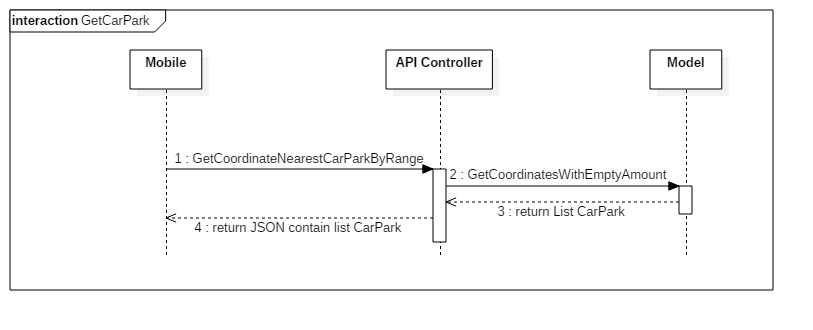


Figure : GetCarPark Diagram

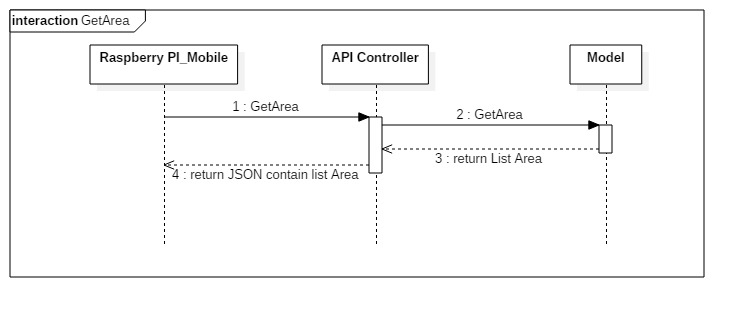


Figure : GetArea Diagram



Figure : GetParkingLot Diagram

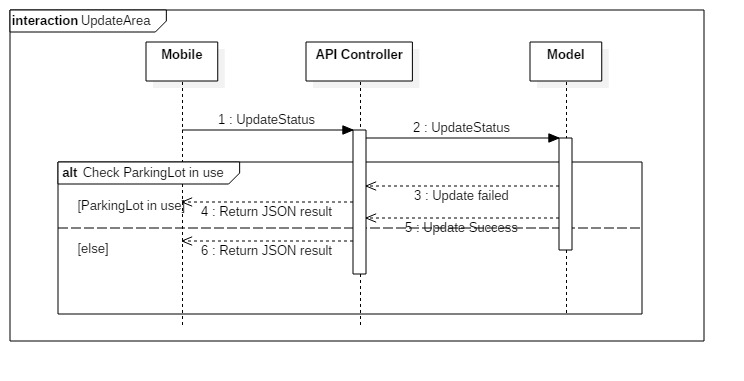


Figure : UpdateArea Diagram

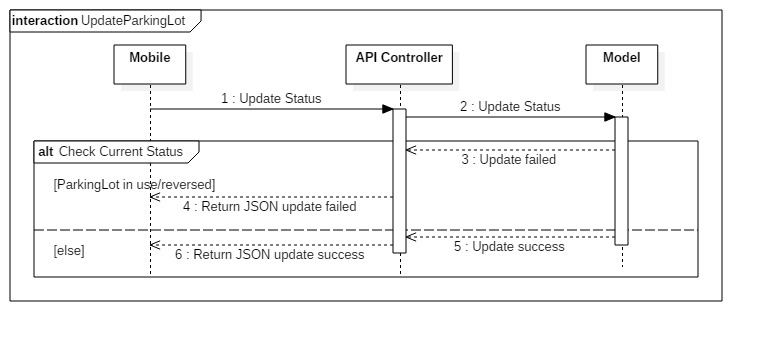


Figure : UpdateParkingLot Diagram

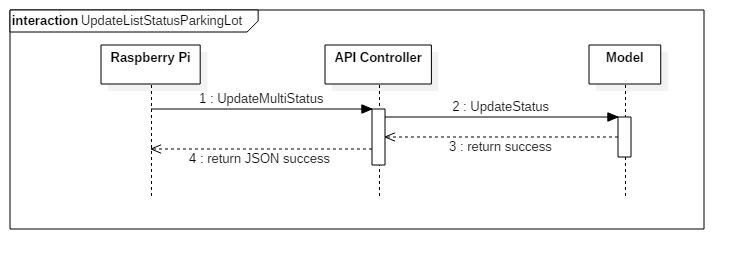


Figure : UpdateListStatusParkingLot Diagram

### Mobile Detailed Description

#### Class Diagram

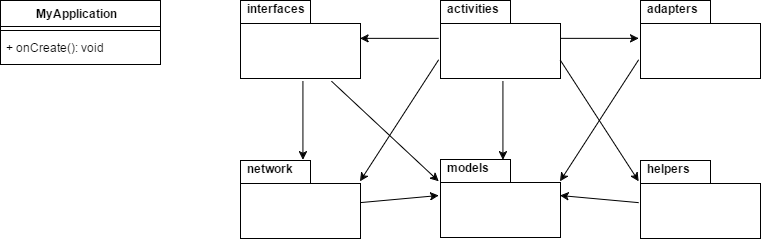


Figure : Main class diagram and package view

| **Packages Dictionary: describe packages** | |
| --- | --- |
| **Package Name** | **Description** |
| Interfaces | Package that provides interfaces for API communication |
| Activities | Package that provides all activities for Mobile Application |
| Adapters | Package that provides adapters for Recycler View / List View / Info Window |
| Network | Package that provides network models and network classes |
| Models | Package that provides all models |
| Helpers | Package that provides all helper class for database, network, pubnub… |

Table : Packages dictionary



Figure : Activities package class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| MainActivity | Activity to handle direction of activity flow when app start |
| CarParkDetailActivity | Activity to provide detail information of car park and to handle request from user |
| ManagerActivity | Activity to provide manager with car parks information and to handle commands |
| AreaListActivity | Activity to provide list of areas in car park and handle commands |
| ParkingLotListActivity | Activity to provide list of parking lots in area and handle commands |
| LoginActivity | Activity to handle login request of user |
| RegisterActivity | Activity to handle register request of user |
| UserActivity | Activity to provide user with car parks information in map view and handle commands and directions to other activities |
| CarParkListActivity | Activity to provide user with car parks information in list view and handle commands |
| TransactionActivity | Activity to provide user with list of history transactions and handle commands |

Table : Activities package class dictionary



Figure : Adapters package class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| AreaAdapter | Provides a binding from Area data set to views that are displayed within a Recycler View |
| AreaAdapter  .OnItemClickListener | Interface definition for a callback to be invoked when an item in AreaAdapter has been click |
| AreaAdapter  .ViewHolder | Describes an item view and metadata about its place within the Recycler View |
| CarParkAdapter | Provides a binding from Car Park data set to views that are displayed within a Recycler View |
| CarParkAdapter  .OnItemClickListener | Interface definition for a callback to be invoked when an item in CarParkAdapter has been click |
| CarParkAdapter  .ViewHolder | Describes an item view and metadata about its place within the Recycler View |
| ParkingLotAdapter | Provides a binding from Parking Lot data set to views that are displayed within a Recycler View |
| ParkingLotAdapter  .OnItemClickListener | Interface definition for a callback to be invoked when an item in ParkingLotdapter has been click |
| ParkingLotAdapter  .ViewHolder | Describes an item view and metadata about its place within the Recycler View |
| CarParkAdvanceAdapter | Provides a binding from Car Park Advance data set to views that are displayed within a Recycler View |
| CarParkAdvanceAdapter  .OnItemClickListener | Interface definition for a callback to be invoked when an item in CarParkAdvanceAdapter has been click |
| CarParkAdvanceAdapter  .ViewHolder | Describes an item view and metadata about its place within the Recycler View |
| TransactionAdapter | Provides a binding from Transaction data set to views that are displayed within a Recycler View |
| TransactionAdapter  .OnItemClickListener | Interface definition for a callback to be invoked when an item in TransactionAdapter has been click |
| TransactionAdapter  .ViewHolder | Describes an item view and metadata about its place within the Recycler View |
| UserInfoWindowAdapter | Provides views for customized rendering of info windows |

Table : Adapters package class dictionary

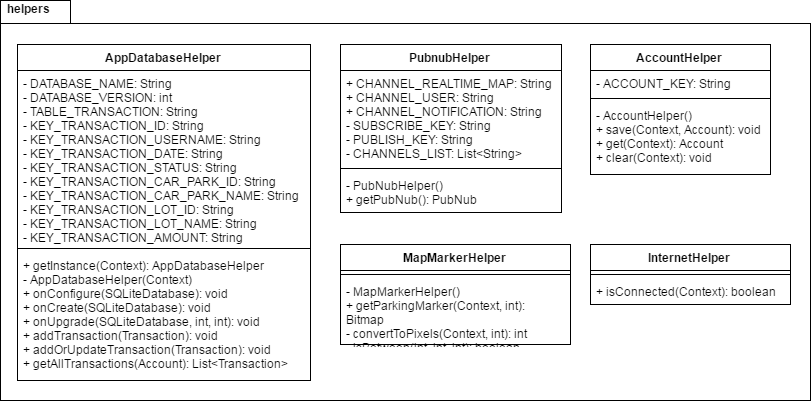


Figure : Helpers package class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| AppDatabaseHelper | A helper class provides API to work with SQLite |
| PubnubHelper | A helper class provides API to work with PubNub |
| AccountHelper | A helper class provides API to work with account data in shared preferences |
| MapMarkerHelper | A helper class provides utility for Google Map Marker |
| InternetHelper | A helper class provides utility for network |

Table : Helpers package class dictionary

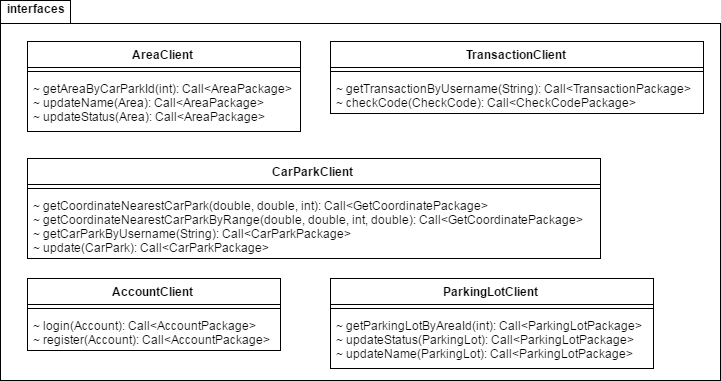


Figure : Interfaces package class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| AreaClient | An interface to consume API call for Area |
| TransactionClient | An interface to consume API call for Transaction |
| CarParkClient | An interface to consume API call for Car Park |
| AccountClient | An interface to consume API call for Account |
| ParkingLotClient | An interface to consume API call for Parking Lot |

Table : Interfaces package class dictionary



Figure : Models package class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| Account | Contain the account information |
| Area | Contain the area information |
| CheckCode | Contain the check code information |
| Geo | Contain the geo information |
| CarPark | Contain the car park information |
| Transaction | Contain the transaction information |
| ParkingLot | Contain the parking lot information |
| TransactionStatus | An enumeration contains the status of transaction |
| AreaStatus | An enumeration contains the status of area |
| ParkingLotStatus | An enumeration contains the status of parking lot |

Table : Models package class dictionary

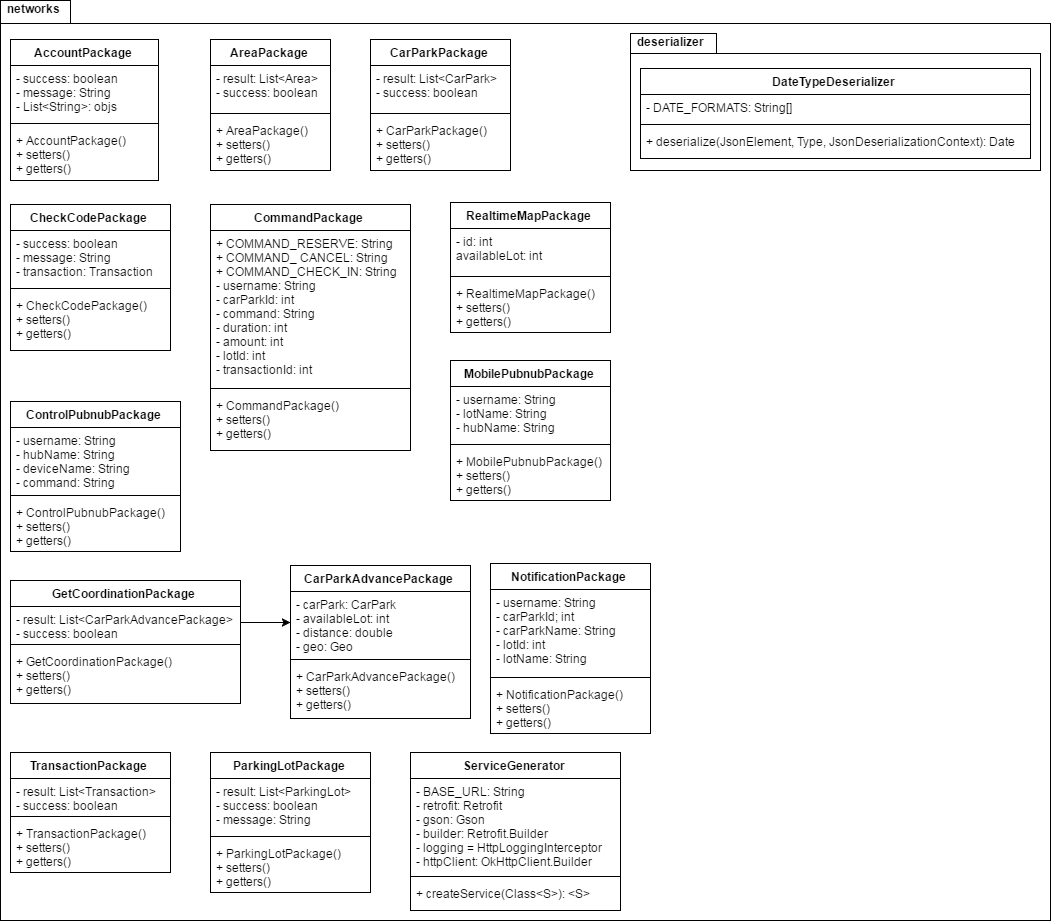


Figure : Networks package class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| AccountPackage | Contain the account information used in network |
| AreaPackage | Contain the area information used in network |
| CarParkPackage | Contain the car park information used in network |
| CheckCodePackage | Contain the check code information used in network |
| CommandPackage | Contain the command information used in network |
| RealtimeMapPackage | Contain the realtime map information used in network |
| ControlPubnubPackage | Contain the control pubnub information used in network |
| MobilePubnubPackage | Contain the mobile pubnub information used in network |
| GetCoordinationPackage | Contain the get coordination information used in network |
| CarParkAdvancePackage | Contain the car park advance information used in network |
| NotificationPackage | Contain the notification information used in network |
| TransactionPackage | Contain the transaction information used in network |
| ParkingLotPackage | Contain the parking lot information used in network |
| ServiceGenerator | A generator to generate retrofit service with interfaces |
| DateTypeDeserializer | A deserializer that can deserialize multiple type of Date format in json |

Table : Networks package class dictionary

#### Class Diagram Explanation

##### [Activities] MainActivity

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | Void | Protected | Initialize activity |
| directActivity | Void | Private | Direct flow of activity depend on shared preferences status |

Table : [Activities] MainActivity methods

##### [Activities] CarParkDetailActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| EXTRA\_CAR\_PARK | String | private | Extra name used in Intent |
| account | Account | private | Login user account |
| pubNub | PubNub | private | Instance of PubNub |
| toolbar | Toolbar | package | Instance of Toolbar |
| tvName | TextView | package | View of car park name |
| tvDistanceAway | TextView | package | View of distance between car park and target |
| tvAvailableLot | TextView | package | View of car park available lot |
| tvAddress | TextView | package | View of car park address |
| tvPhone | TextView | package | View of car park phone |
| tvEmail | TextView | package | View of car park email |
| tvDescription | TextView | package | View of car park description |
| btnCall | Button | package | View of call button |
| btnRoute | Button | package | View of request routing button |
| btnReserve | Button | package | View of request reservation button |
| carpark | CarPark | package | Object of CarPark |

Table : [Activities] CarParkDetailActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| createIntent | Intent | public | Return Intent of activity |
| onCreate | void | protected | Initialize activity |
| onOptionsItemSelected | boolean | public | Callback when options item is selected |
| initiateFields | void | private | Initialize all member variables |
| initiateViews | void | private | Initialize all views |
| setToolbarBackground | void | private | Set background for toolbar |
| onCallButtonClick | void | protected | Callback when call button is clicked |
| onRouteButtonClick | void | protected | Callback when route button is clicked |
| onReserveButtonClick | void | protected | Callback when reserve butotn is clicked |
| finalConfirm | void | private | Show final confirmation of request |
| reserveParkingLot | void | private | Process to reserve a parking lot |
| initatePubnub | void | private | Initalize PubNub instance |
| isBetween | boolean | private | Check if the number is between 2 other numbers |
| getDistanceString | String | private | Return distance in string format |

Table : [Activities] CarParkDetailActivity methods

##### [Activities] ManagerActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| toolbar | Toolbar | protected | Instance of Toolbar |
| toolbarProgress | ProgressBar | protected | Instance of ProgressBar |
| drawerLayout | DrawerLayout | protected | Instance of DrawerLayout |
| navigationView | NavigationView | protected | Instance of NavigationView |
| rvCarParkList | RecyclerView | protected | Recycler View of car park list |
| detailDialog | MaterialDialog | private | Detail dialog of car park |
| etName | EditText | private | View of car park name in detail dialog |
| etPhone | EditText | private | View of car park phone in detail dialog |
| etEmail | EditText | private | View of car park email in detail dialog |
| etAddress | EditText | private | View of car park address in detail dialog |
| etDescription | EditText | private | View of car park description in detail dialog |
| focusedCarPark | CarPark | private | Instance of focused car park |
| btnPositive | MDButton | private | View of positive button of detail dialog |
| checkCodeDialog | MaterialDialog | private | Check code dialog |
| spnCarParks | Spinner | private | Drop down list of car parks in check code dialog |
| etUsername | EditText | private | View of username in check code dialog |
| etPin | EditText | private | View of pin code in check code dialog |
| drawerToggle | ActionBarDrawerToggle | private | Instance of ActionBarDrawerToggle |
| carParks | List<CarPark> | private | List of CarPark |
| adapter | CarParkAdapter | private | Adapter of CarPark |

Table : [Activities] ManagerActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Initialize activity |
| onOptionsItemSelected | boolean | public | Callback when options item is selected |
| onPostCreate | void | protected | Callback after activity is created |
| onConfigurationChanged | void | public | Callback when activity change configuration |
| initiateFields | void | private | Initialize member variables |
| initiateViews | void | private | Initialize views |
| setupDetailDialog | void | private | Initialize detail dialog |
| setupCheckCodeDialog | void | private | Initialize check code dialog |
| onDrawerItemClick | void | private | Callback when drawer item is clicked |
| getCarParkData | void | private | Get car park data using API |
| updateCarPark | void | private | Update car park data using API |
| checkCode | void | private | Check username and pin code using API |

Table : [Activities] ManagerActivity methods

##### [Activities] AreaListActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| EXTRA\_CAR\_PARK\_ID | String | private | Extra name of intent |
| EXTRA\_CAR\_PARK\_NAME | String | private | Extra name of intent |
| toolbar | Toolbar | protected | Instance of Toolbar |
| toolbarProgress | ProgressBar | protected | Instance of ProgressBar |
| rvArea | RecyclerView | protected | Recycler View of area list |
| detailDialog | MaterialDialog | private | Detail dialog of area |
| etName | EditText | private | View of area name in detail dialog |
| scActive | SwitchCompat | private | View of area active switch in detail dialog |
| focusedArea | Area | private | Instance of focused Area |
| btnPositive | MDButton | private | View of positive button in detail dialog |
| carParkId | int | private | Value of car park id |
| carParkName | String | private | Value of car park name |
| areas | List<Area> | private | List of Area |
| adapter | AreaAdapter | private | Adapter of Area |

Table : [Activities] AreaListActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| createIntent | Intent | public | Return intent of activity |
| onCreate | void | protected | Initialize activity |
| onOptionsItemSelected | boolean | public | Callback when options item is selected |
| initiateFields | void | private | Initialize member variables |
| initiateViews | void | private | Initialize views |
| getAreaData | void | private | Get area data using API |
| updateArea | void | private | Check change of area and call correct update method |
| updateAreaStatus | void | private | Update area status using API |
| updateAreaName | void | private | Update area name using API |

Table : [Activities] AreaListActivity methods

##### [Activities] ParkingLotListActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| EXTRA\_AREA\_ID | String | private | Extra of intent |
| EXTRA\_AREA\_NAME | String | private | Extra of intent |
| toolbar | Toolbar | protected | Instance of Toolbar |
| toolbarProgress | ProgressBar | protected | Instance of ProgressBar |
| rvParkingLot | RecyclerView | protected | Recycler View of parking lot list |
| dialog | MaterialDialog | private | Detail dialog of parking lot |
| etName | EditText | private | View of parking lot name in detail dialog |
| tvStatus | TextView | private | View of lot status in detail dialog |
| focusedLot | ParkingLot | private | Instance of focused Parking Lot |
| btnPositive | MDButton | private | View of positive button in detail dialog |
| areaId | int | private | Value of area id |
| areaName | String | private | Value of area name |
| lots | List<ParkingLot> | private | List of Parking Lot |
| adapter | ParkingLotAdapter | private | Adapter of Parking Lot |

Table : [Activities] ParkingLotListActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| createIntent | Intent | public | Return intent of activity |
| onCreate | void | protected | Initialize activity |
| onOptionsItemSelected | boolean | public | Callback when options item is selected |
| initiateFields | void | private | Initialize member variables |
| initiateViews | void | private | Initialize views |
| getParkingLotData | void | private | Get parking lot data using API |
| updateLot | void | private | Update lot information using API |
| updateLotStatus | void | private | Update lot status using API |

Table : [Activities] ParkingLotListActivity methods

##### [Activities] LoginActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| etUsername | EditText | package | View of username |
| etPassword | EditText | package | View of password |
| btnSignIn | EditText | package | View of sign in button |
| tvRegister | TextView | package | View of register text |
| tvSignInGuest | TextView | package | View of sign in as guest text |
| dialog | MaterialDialog | private | Progress dialog |
| account | Account | private | Instance of Account |

Table : [Activities] LoginActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | public | Initialize activity |
| onLoginButtonClick | void | protected | Callback when login button is clicked |
| onRegisterTextClick | void | protected | Callback when register text is clicked |
| onSignInGuestTextClick | void | protected | Callback when sign in as guest is clicked |

Table : [Activities] LoginActivity methods

##### [Activities] RegisterActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| etEmail | EditText | package | View of email |
| etPassword | EditText | package | View of password |
| etConfirmPassword | EditText | package | View of confirm password |
| etUsername | EditText | package | View of username |
| etFullname | EditText | package | View of fullname |
| btnRegister | Button | package | View of register button |
| tvLogin | TextView | package | View of login text |

Table : [Activities] RegisterActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Initialize activity |
| validateEmail | boolean | private | Check email format |
| validatePassword | boolean | private | Check password format |
| onRegisterButtonClick | void | protected | Callback when register button is clicked |
| onLoginTextClick | void | protected | Callback when login text is clicked |

Table : [Activities] RegisterActivity methods

##### [Activities] UserActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| MAP\_ZOOM\_TO | int | private | Value of map zoom |
| REQUEST\_CHECK\_SETTING | int | private | Value of request check setting |
| map | GoogleMap | private | Instance of GoogleMap |
| centerLocation | LatLng | private | Location of map center |
| currentLocation | LatLng | private | Location of current postion |
| placeLocation | LatLng | private | Location of searched place |
| fromYou | boolean | private | Flag to check if the distance show on marker is from current location or from place |
| placeName | String | private | Name of searched place |
| autocompleFragment | Place  Autocomplete  Fragment | private | Instance of PlaceAutocompleteFragment |
| refreshCarParkData | boolean | private | Flag to check if need to refresh car park data on map |
| googleApiClient | GoogleApi  Client | private | Instance of GoogleApiClient |
| markerMap | HashMap<  Integer, Marker> | private | Map of marker and car park id |
| pubNub | PubNub | private | Instance of PubNub |
| drawerLayout | DrawerLayout | private | Instance of DrawerLayout |
| toolbar | Toolbar | private | Instance of Toolbar |
| toobarProgress | ProgressBar | private | Instance of ProgressBar |
| navigationMenu | NavigationView | private | Instance of NavigationView |
| drawerToggle | ActionBar  DrawerToggle | private | Instance of ActionBarDrawerToggle |
| searchRange | double | private | Value of search range |
| numberOfCar | int | private | Value of number of car to request data from API |

Table : [Activities] UserActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Initialize activity |
| onPostCreate | void | protected | Callback after activity is created |
| onConfigurationChanged | void | public | Callback after activity change configuration |
| onResume | void | protected | Callback when activity resume |
| onStart | void | protected | Callback when activity start |
| onStop | void | protected | Callback when activity stop |
| onDestroy | void | protected | Callback when activity destroy |
| initiateInstance | void | private | Initialize member variables and views |
| setupDrawerContent | void | private | Setup drawer |
| onDrawerItemClick | void | private | Callback when drawer item is clicked |
| onOptionsItemSelected | boolean | public | Callback when options item is selected |
| initiatePubnub | void | private | Initialize PubNub |
| onMapReady | void | public | Callback when google map is ready |
| onConnected | void | public | Callback when google client is connected |
| getCurrentLocation | void | private | Get current location |
| getCarParkData | void | private | Get car park data using API |
| onConnectionSuspended | void | public | Callback when connection is suspended |
| onConnectionFailed | void | public | Callback when connection is failed |
| onActivityResult | vodi | public | Callback when intent return result |
| locationSettingRequest | void | private | Request turn on location for app |

Table : [Activities] UserActivity methods

##### [Activities] CarParkListActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| EXTRA\_CURRENT  \_LOCATION\_LAT | String | private | Extra for intent |
| EXTRA\_CURRENT  \_LOCATION\_LON | String | private | Extra for intent |
| EXTRA\_FROM\_TARGET | String | private | Extra for intent |
| rvCarParkList | RecyclerView | package | Recycler View for car park list |
| toolbar | Toolbar | package | Instance of Toolbar |
| toolbarProgress | ProgressBar | package | Instance of ProgressBar |
| currentPosition | LatLng | private | Current location |
| adapter | CarParkAdvance  Adapter | private | Advance adapter of Car Park |
| carParks | List<CarPark> | private | List of CarPark |
| target | String | private | Name of target |
| numberOfCar | int | private | Value of number of car to request search |

Table : [Activities] CarParkListActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| createIntent | Intent | public | Return intent of activity |
| onCreate | void | protected | Initialize activity |
| onOptionsItemSelected | boolean | public | Callback when options item is selected |
| initiateViews | void | private | Initialize views |
| initiateFields | void | private | Initialize member variables |
| getCarParkData | void | private | Get car park data using API |

Table : [Activities] CarParkListActivity methods

##### [Activities] TransactionActivity

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| transactions | List<Transaction> | private | List of Transaction |
| adapter | TransactionAdapter | private | Adapter of Transaction |
| rvTransactionList | RecyclerView | package | Recycler View of transaction list |
| toolbar | Toolbar | package | Instance of Toolbar |
| toolbarProgress | ProgressBar | package | Instance of ProgressBar |

Table : [Activities] TransactionActivity attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreate | void | protected | Initialize activity |
| onOptionsItemSelected | boolean | public | Callback when options item is selected |
| initiateViews | void | private | Initialize views |
| initiateFields | void | private | Initialize fields |
| cancelTransaction | void | private | Callback when cancel transaction button is clicked |
| checkInTransaction | void | private | Callback when check in transaction button is clicked |
| getTransactionData | void | private | Get transaction data using API |

Table : [Activities] TransactionActivity methods

##### [Adapters] AreaAdapter

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| listener | AreaAdapter.  OnItemClickListener | private | Interface of item click listener |
| context | Context | private | Context of adapter |
| areas | List<Area> | private | List of Area |

Table : [Adapters] AreaAdapter attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreateViewHolder | AreaAdapter.  ViewHolder | public | Callback when creating view holder |
| onBindViewHolder | AreaAdapter.  ViewHolder | public | Callback when binding data to view holder |
| getItemCount | int | public | Return number of items in list |
| setOnItemClickListener | void | public | Set listener callback for item |
| getContext | Context | private | Return adapter context |

Table : [Adapters] AreaAdapter methods

##### [Adapters] AreaAdapter.OnItemClickListener

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onItemClick | void | package | Callback when item in list is clicked |
| onItemLongClick | void | package | Callback when item in list is being hold |

Table : [Adapters] AreaAdapter.OnItemClickListener methods

##### [Adapters] AreaAdapter.ViewHolder

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| tvName | TextView | package | View of area name |
| tvAvailableLot | TextView | package | View of area available lot |

Table : [Adapters] AreaAdapter.ViewHolder attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| bind | void | package | Bind data to view holder |
| getAvailableColor | int | private | Return color for available view |
| isBetween | boolean | private | Check if number is between 2 other numbers |

Table : [Adapters] AreaAdapter.ViewHolder methods

##### [Adapters] CarParkAdapter

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| listener | CarParkAdapter.  OnItemClickListener | private | Interface of item click listener |
| context | Context | private | Context of adapter |
| carParks | List<CarPark> | private | List of CarPark |

Table : [Adapters] CarParkAdapter attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreateViewHolder | CarParkAdapter.  ViewHolder | public | Callback when creating view holder |
| onBindViewHolder | CarParkAdapter.  ViewHolder | public | Callback when binding data to view holder |
| getItemCount | int | public | Return number of items in list |
| setOnItemClickListener | void | public | Set listener callback for item |
| getContext | Context | private | Return adapter context |

Table : [Adapters] CarParkAdapter methods

##### [Adapters] CarParkAdapter.OnItemClickListener

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onItemClick | void | package | Callback when item in list is clicked |
| onItemLongClick | void | package | Callback when item in list is being hold |

Table : [Adapter] CarParkAdapter.OnItemClickListener methods

##### [Adapters] CarParkAdapter.ViewHolder

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| tvName | TextView | package | View of car park name |
| tvAddress | TextView | package | View of car park address |
| tvAvailableLot | TextView | package | View of car park available lot |

Table : [Adapters] CarParkAdapter.ViewHolder attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| bind | void | package | Bind data to view holder |
| getAvailableColor | int | private | Return color for available view |
| isBetween | boolean | private | Check if number is between 2 other numbers |

Table : [Adapters] CarParkAdapter.ViewHolder methods

##### [Adapters] ParkingLotAdapter

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| listener | ParkingLotAdapter.  OnItemClickListener | private | Interface of item click listener |
| context | Context | private | Context of adapter |
| lots | List<ParkingLot> | private | List of ParkingLot |

Table : [Adapters] ParkingLotAdapter attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreateViewHolder | ParkingLotAdapter.  ViewHolder | public | Callback when creating view holder |
| onBindViewHolder | ParkingLotAdapter.  ViewHolder | public | Callback when binding data to view holder |
| getItemCount | int | public | Return number of items in list |
| setOnItemClickListener | void | public | Set listener callback for item |
| getContext | Context | private | Return adapter context |

Table : [Adapters] ParkingLotAdapter methods

##### [Adapters] ParkingLotAdapter.OnItemClickListener

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onItemClick | void | package | Callback when item in list is clicked |
| onItemLongClick | void | package | Callback when item in list is being hold |

Table : [Adapters] ParkingLotAdapter.OnItemClickListener methods

##### [Adapters] ParkingLotAdapter.ViewHolder

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| tvName | TextView | package | View of parking lot name |

Table : [Adapters] ParkingLotAdapter.ViewHolder attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| bind | void | package | Bind data to view holder |

Table : [Adapters] ParkingLotAdapter.ViewHolder methods

##### [Adapters] CarParkAdvanceAdapter

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| listener | CarParkAdvanceAdapter.  OnItemClickListener | private | Interface of item click listener |
| context | Context | private | Context of adapter |
| carParks | List<CarPark> | private | List of CarPark |

Table : [Adapters] CarParkAdvanceAdapter attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreateViewHolder | CarParkAdvanceAdapter.  ViewHolder | public | Callback when creating view holder |
| onBindViewHolder | CarParkAdvanceAdapter.  ViewHolder | public | Callback when binding data to view holder |
| getItemCount | int | public | Return number of items in list |
| setOnItemClickListener | void | public | Set listener callback for item |
| getContext | Context | private | Return adapter context |

Table : [Adapters] CarParkAdvanceAdapter methods

##### [Adapters] CarParkAdvanceAdapter.OnItemClickListener

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onItemClick | void | package | Callback when item in list is clicked |
| onItemLongClick | void | package | Callback when item in list is being hold |

Table : [Adapters] CarParkAdvanceAdapter.OnItemClickListener methods

##### [Adapters] CarParkAdvanceAdapter.ViewHolder

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| tvName | TextView | package | View of car park name |
| tvAvailableLot | TextView | package | View of car park available lot |
| tvDistance | TextView | package | View of car park distance |
| tvAddress | TextView | package | View of car park address |

Table : [Adapters] CarParkAdvanceAdapter.ViewHolder attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| bind | void | package | Bind data to view holder |
| getAvailableColor | int | private | Return color for available view |
| isBetween | boolean | private | Check if number is between 2 other numbers |
| getDistanceString | String | private | Get distance in string format |

Table : [Adapters] CarParkAdvanceAdapter.ViewHolder methods

##### [Adapters] TransactionAdapter

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| listener | TransactionAdapter.  OnItemClickListener | private | Interface of item click listener |
| context | Context | private | Context of adapter |
| transactions | List<Transaction> | private | List of Transaction |

Table : [Adapters] TransactionAdapter attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onCreateViewHolder | TransactionAdapter.  ViewHolder | public | Callback when creating view holder |
| onBindViewHolder | TransactionAdapter.  ViewHolder | public | Callback when binding data to view holder |
| getItemCount | int | public | Return number of items in list |
| setOnItemClickListener | void | public | Set listener callback for item |
| getContext | Context | private | Return adapter context |

Table : [Adapters] TransactionAdapter methods

##### [Adapters] TransactionAdapter.OnItemClickListener

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| onItemClick | void | package | Callback when item in list is clicked |
| onItemLongClick | void | package | Callback when item in list is being hold |

Table : [Adapters] TransactionAdapter.OnItemClickListener methods

##### [Adapters] TransactionAdapter.ViewHolder

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| tvAddress | TextView | package | View of transaction address |
| tvDate | TextView | package | View of transaction date |
| tvStatus | TextView | package | View of transaction status |
| tvAmount | TextView | package | View of transaction amount |
| tvLotName | TextView | package | View of transaction lot name |
| tvPin | TextView | package | View of transaction pin code |

Table : [Adapters] TransactionAdapter.ViewHolder attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| bind | void | package | Bind data to view holder |
| getLotNameText | String | private | Get lot name in string format |
| getAmountText | String | private | Get amount in string format |
| getFormatedDate | String | private | Get date in string format |

Table : [Adapters] TransactionAdapter.ViewHolder methods

##### [Adapters] UserInfoWindowAdapter

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| contentView | View | private | Content view |
| tvCarParkName | TextView | package | View of car park name |
| tvCarParkAddress | TextView | package | View of car park address |
| tvAwayDistance | TextView | package | View of car park away distance |

Table : [Adapters] UserInfoWindowAdapter attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getInfoWindow | View | public | Return Info Window view |
| getInfoContents | View | public | Return Info Contents view |
| getDistanceString | String | private | Get distance in string format |

Table : [Adapters] UserInfoWindowAdapter methods

##### [Helpers] AppDatabaseHelper

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| DATABASE\_NAME | String | private | Name of database |
| DATABASE\_VERSION | int | private | Version number of database |
| TABLE\_TRANSACTION | String | private | Name of transaction table |
| KEY\_TRANSACTION\_ID | String | private | Key of transaction id |
| KEY\_TRANSACTION  \_USERNAME | String | private | Key of transaction username |
| KEY\_TRANSACTION  \_DATE | String | private | Key of transaction date |
| KEY\_TRANSACTION  \_STATUS | String | private | Key of transaction status |
| KEY\_TRANSACTION  \_CAR\_PARK\_ID | String | private | Key of transaction car park id |
| KEY\_TRANSACTION  \_CAR\_PARK\_NAME | String | private | Key of transaction car park name |
| KEY\_TRANSACTION  \_LOT\_ID | String | private | Key of transaction lot id |
| KEY\_TRANSACTION  \_LOT\_NAME | String | private | Key of transaction lot name |
| KEY\_TRANSACTION  \_AMOUNT | String | private | Key of transaction amount |

Table : [Helpers] AppDatabaseHelper attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getInstance | AppDatabaseHelper | public | Return singleton instance of helper |
| onConfigure | void | public | Callback when database is configured |
| onCreate | void | public | Callback when database is created |
| onUpgrade | void | public | Callback when database is upgraded |
| addTransaction | void | public | Add transaction to database |
| addOrUpdateTransaction | void | public | Add or Update transaction to database |
| getAllTransactions | List<Transaction> | public | Return all transactions of user in database |

Table : [Helpers] AppDatabaseHelper methods

##### [Helpers] PubnubHelper

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| CHANNEL\_REALTIME\_MAP | String | public | Name of realtime map channel |
| CHANNEL\_USER | String | public | Name of user channel |
| CHANNEL\_NOTIFICATION | String | public | Name of notification channel |
| SUBSCRIBE\_KEY | String | private | Pubnub subscribe key |
| PUBLISH\_KEY | String | private | Pubnub publish key |
| CHANNELS\_LIST | List<String> | private | List of pubnub channels |

Table : [Helpers] PubnubHelper attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getPubNub | PubNub | public | Return a singleton instance of PubNub |

Table : [Helpers] PubnubHelper methods

##### [Helpers] AccountHelper

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ACCOUNT\_KEY | String | private | Key name of account in shared preferences |

Table : [Helpers] AccountHelper attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| save | void | public | Save account information in shared preferences |
| get | Account | public | Get account information from shared preferences |
| clear | void | public | Delete all account informations in shared preferences |

Table : [Helpers] AccountHelper methods

##### [Helpers] MapMarkerHelper

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getParkingMarker | Bitmap | public | Return a custom bitmap marker |
| convertToPixels | int | private | Convert dp to pixels |
| isBetween | boolean | private | Check if the number is between 2 other numbers |

Table : [Helpers] MapMarkerHelper methods

##### [Helpers] InternetHelper

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| isConnected | boolean | public | Check if internet is available |

Table : [Helpers] InternetHelper methods

##### [Interfaces] AreaClient

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getAreaByCarParkId | Call<AreaPackage> | package | Interface for get area by car park id API |
| updateName | Call<AreaPackage> | package | Interface for update area name API |
| updateStatus | Call<AreaPackage> | package | Interface for update area status API |

Table : [Interfaces] AreaClient methods

##### [Interfaces] TransactionClient

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getTransactionByUsername | Call<Transaction  Package> | package | Interface for get transactions by username API |
| checkCode | Call<Transaction  Package> | package | Interface for check code API |

Table : [Interfaces] TransactionClient methods

##### [Interfaces] CarParkClient

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getCoordinate  NearesCarPark | Call<Get  CoordinatePackage> | package | Interface for get nearest car park API |
| getCoordinate  NearestCarPark  ByRange | Call<Get  CoordinatePackage> | package | Interface for get nearest car park in range API |
| getCarPark  ByUsername | Call<CarPark  Package> | package | Interface for get car park by username API |
| update | Call<CarPark  Package> | package | Interface for update car park API |

Table : [Interfaces] CarParkClient methods

##### [Interfaces] AccountClient

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| login | Call<Account  Package> | package | Interface for login API |
| register | Call<Account  Package> | package | Interface for register API |

Table : [Interfaces] AccountClient methods

##### [Interfaces] ParkingLotClient

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getParkingLot  ByAreaId | Call<Parking  LotPackage> | package | Interface for get parking lot by area id API |
| updateStatus | Call<Parking  LotPackage> | package | Interface for update parking lot status API |
| updateName | Call<Parking  LotPackage> | package | Interface for update parking lot name API |

Table : [Interfaces] ParkingLotClient methods

##### [Models] Account

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| ROLE\_USER | String | public | Value of role user |
| ROLE\_MANAGER | String | public | Value of role manager |
| email | String | private | Email of account |
| username | String | private | Username of account |
| fullname | String | private | Fullname of account |
| password | String | private | Password of account |
| confirmPassword | String | private | Confirm password |
| role | String | private | Role of account |

Table : [Models] Account attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Models] Account methods

##### [Models] Area

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique id of area |
| name | String | private | Name of area |
| emptyAmount | int | private | Current available lot of area |
| updateAvailable | boolean | private | Flag to check if area is updatable |
| status | int | private | Current status of area |

Table : [Models] Area attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Models] Area methods

##### [Models] CheckCode

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| carParkId | int | private | Unique id of car park |
| username | String | private | Username of user want to check in with code |
| transactionCode | String | private | Pin code come with transaction |

Table : [Models] CheckCode attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Models] CheckCode methods

##### [Models] Geo

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| latitude | double | private | Geo location latitude |
| longitude | double | private | Geo location longitude |
| altitude | String | private | Geo location altitude |
| horizontalAccuracy | String | private | Geo location horizontal accuracy |
| verticalAccuracy | String | private | Geo location vertical accuracy |
| speed | String | private | Geo location speed |
| course | String | private | Geo location course |
| unknown | boolean | private | Flag to check Geo location |

Table : [Models] Geo attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Models] Geo methods

##### [Models] CarPark

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique id of car park |
| name | String | private | Name of car park |
| address | String | private | Address of car park |
| phone | String | private | Phone of car park |
| email | String | private | Email of car park |
| description | String | private | Description of car park |
| lat | String | private | Latitude of car park |
| lon | String | private | Longitude of car park |
| fee | int | private | Fee per hour of car park |
| active | boolean | private | Flag to check if car park is active |
| availableLot | int | private | Current available parking lot in car park |
| awayDistance | double | private | Away distance of car park from target |
| fromTarget | String | private | Target to check away distance |

Table : [Models] CarPark attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Models] CarPark methods

##### [Models] Transaction

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique id of transaction |
| username | String | private | Username of transaction |
| carParkId | int | private | Car park id of transaction |
| carpark | CarPark | private | Instance of CarPark object in transaction |
| lotId | int | private | Parking Lot id of transaction |
| lot | ParkingLot | private | Instance of ParkingLot object in transaction |
| date | Date | private | The moment user request reservation |
| endTime | Date | private | The end of reservation |
| status | int | private | Status of reservation |
| amount | double | private | Cost of the reservation |
| transactionCode | String | private | Pin code of transaction |

Table : [Models] Transaction attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Models] Transaction methods

##### [Models] ParkingLot

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Unique if of parking lot |
| name | String | private | Name of parking lot |
| status | int | private | Status of parking lot |

Table : [Models] ParkingLot attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Models] ParkingLot methods

##### [Models] TransactionStatus

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| PENDING | enum | package | Enumeration of transaction pending status |
| RESERVED | enum | package | Enumeration of transaction reserved status |
| FINISHED | enum | package | Enumeration of transaction finished status |
| CANCELED | enum | package | Enumeration of transaction canceled status |
| UNKNOWN | enum | package | Enumeration of transaction unknown status |
| name | String | private | Name of enum |
| id | int | private | Id of enum |

Table : [Models] TransactionStatus attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| getById | TransactionStatus | public | Get enum by id |

Table : [Models] TransactionStatus methods

##### [Models] AreaStatus

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| DEACTIVE | enum | package | Enumeration of area deactive status |
| ACTIVE | enum | package | Enumeration of area active status |
| UNKNOWN | enum | package | Enumeration of area unknown status |
| name | String | private | Name of enum |
| id | int | private | Id of enum |

Table : [Models] AreaStatus attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| getById | AreaStatus | public | Get enum by id |

Table : [Models] AreaStatus methods

##### [Models] ParkingLotStatus

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| DEACTIVE | enum | package | Enumeration of parking lot deactive status |
| ACTIVE | enum | package | Enumeration of parking lot active status |
| RESERVED | enum | package | Enumeration of parking lot reserved status |
| NONAVAILABLE | enum | package | Enumeration of parking lot non-available status |
| name | String | package | Name of enum |
| id | int | package | Id of enum |

Table : [Models] ParkingLotStatus attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| getById | ParkingLotStatus | public | Get enum by id |

Table : [Models] ParkingLotStatus methods

##### [Network] AccountPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| success | boolean | private | Flag to check status of package |
| message | String | private | Message from server |
| objs | List<String> | private | List of account role |

Table : [Network] AccountPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] AccountPackage methods

##### [Network] AreaPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| result | List<Area> | private | List of area returned from server |
| success | boolean | private | Flag to check status of package |

Table : [Network] AreaPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] AreaPackage methods

##### [Network] CarParkPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| result | List<CarPark> | private | List of car park returned from server |
| success | boolean | private | Flag to check the success of package |

Table : [Network] CarParkPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] CarParkPackage methods

##### [Network] CheckCodePackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| success | boolean | private | Flag to check the success of package |
| message | String | private | Message of package |
| transaction | Transaction | private | Transaction object returned from server |

Table : [Network] CheckCodePackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] CheckCodePackage methods

##### [Network] CommandPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| COMMAND\_RESERVE | String | public | Value of reserve command |
| COMMAND\_CANCEL | String | public | Value of cancel command |
| COMAND\_CHECK\_IN | String | public | Value of check in command |
| username | String | private | Username request command |
| carParkId | int | private | Id of target car park |
| command | String | private | Command of package |
| duration | int | private | Duration of reservation if command is reserve |
| amount | int | private | Total cost of reservation if command is reserve |
| lotId | int | private | Id of parking lot if command is cancel or check in |
| transactionId | int | private | Id of transaction if command is cancel or check in |

Table : [Network] CommandPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] CommandPackage methods

##### [Network] RealtimeMapPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | int | private | Id of car park |
| availableLot | int | private | Current available lot of car park |

Table : [Network] RealtimeMapPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] RealtimeMapPackage methods

##### [Network] ControlPubnubPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | private | Username request command |
| hubName | String | private | Name of target hub |
| deviceName | String | private | Name of target device |
| command | String | private | Command of package |

Table : [Network] ControlPubnubPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] ControlPubnubPackage methods

##### [Network] MobilePubnubPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | private | Username of package |
| lotName | String | private | Lot name of package |
| hubName | String | private | Hub name of package |

Table : [Network] MobilePubnubPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] MobilePubnubPackage methods

##### [Network] GetCoordinationPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| result | List<CarPark  AdvancePackage> | private | List of car park advance package |
| success | boolean | private | Flag to check status of package |

Table : [Network] GetCoordinationPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] GetCoordinationPackage methods

##### [Network] CarParkAdvancePackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| carpark | CarPark | private | Instance of car park object |
| availableLot | int | private | Current available lot in car park |
| distance | double | private | Calculated distance away from car park |
| geo | Geo | private | Instanc of geo object |

Table : [Network] CarParkAdvancePackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] CarParkAdvancePackage methods

##### [Network] NotificationPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| username | String | private | Username of notification |
| carParkId | int | private | Id of car park in notification |
| carParkName | String | private | Name of car park |
| lotId | int | private | Id of reserved lot in notification |
| lotName | String | private | Name of reserved lot |

Table : [Network] NotificationPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] NotificationPackage methods

##### [Network] TransactionPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| result | List<Transaction> | private | List of transaction returned from server |
| success | boolean | private | Flag to check status of package |

Table : [Network] TransactionPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] TransactionPackage methods

##### [Network] ParkingLotPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| result | List<ParkingLot> | private | List of parking lot returned from server |
| success | boolean | private | Flag to check status of package |
| message | String | private | Message of package |

Table : [Network] ParkingLotPackage attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Getter | Attribute type | public | Get attribute value |
| Setter | Void | public | Set value of attribute |

Table : [Network] ParkingLotPackage methods

##### [Network] ServiceGenerator

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| BASE\_URL | String | private | Base url of API server |
| retrofit | Retrofit | private | Instance of Retrofit 2 |
| gson | Gson | private | Instance of Gson |
| builder | Retrofi.Builder | private | Builder to create retrofit instance |
| logging | HttpLoggin  Interceptor | private | Logging object |
| httpClient | OkHttpClient.  Builder | private | Builder to create http client instance |

Table : [Network] ServiceGenerator attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| createService | Service type | public | Generate new instance of service interface |

Table : [Network] ServiceGenerator methods

##### [Network] DateTypeDeserializer

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| DATE\_FORMAT | String[] | private | List of common date time format |

Table : [Network] DateTypeDeserializer attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| deserialize | Date | public | Return date object from json |

Table : [Network] DateTypeDeserializer methods

#### Interaction Diagram

##### Load car parks on map

**Summary:** This diagram shows the process of app loads car parks data from API server and display them on map as marker, and update their available lots in real time.

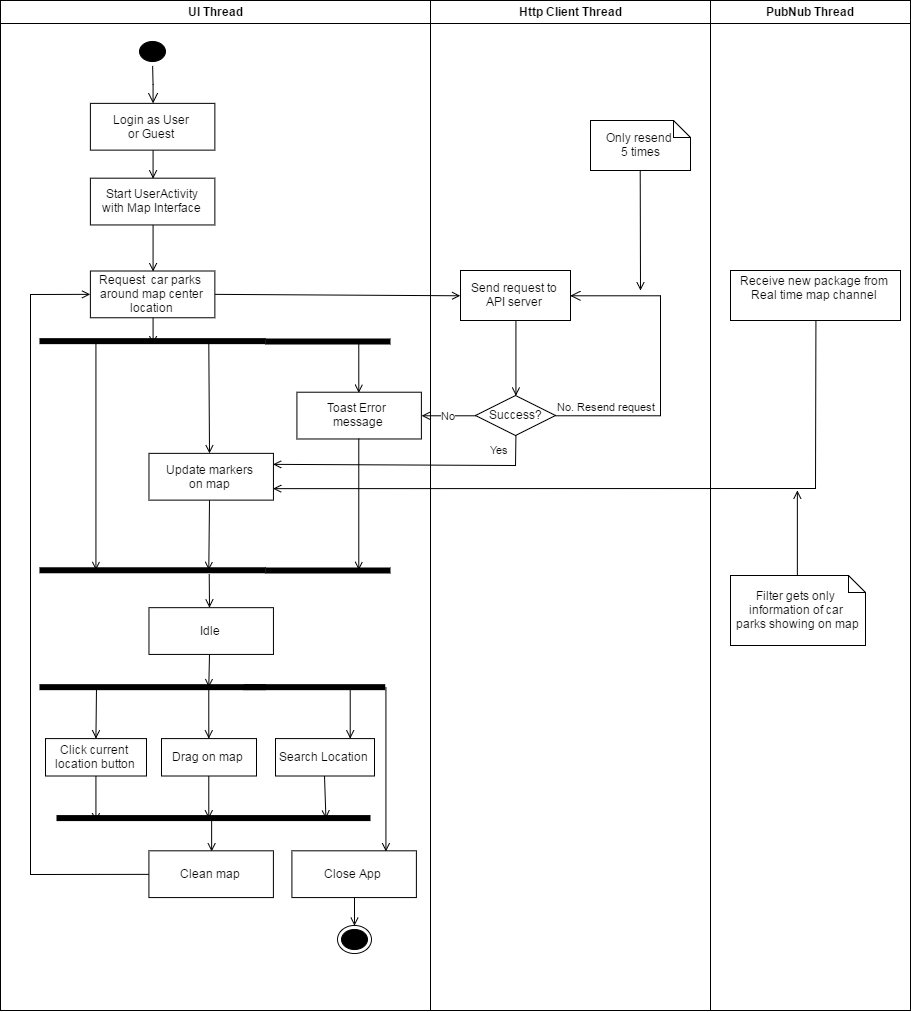


Figure : Interaction diagram - Load car parks on map

##### Load car parks in list

**Summary:** This diagram shows the process of app loads car parks data from API server and display them in list format, and update their available lots in real time.

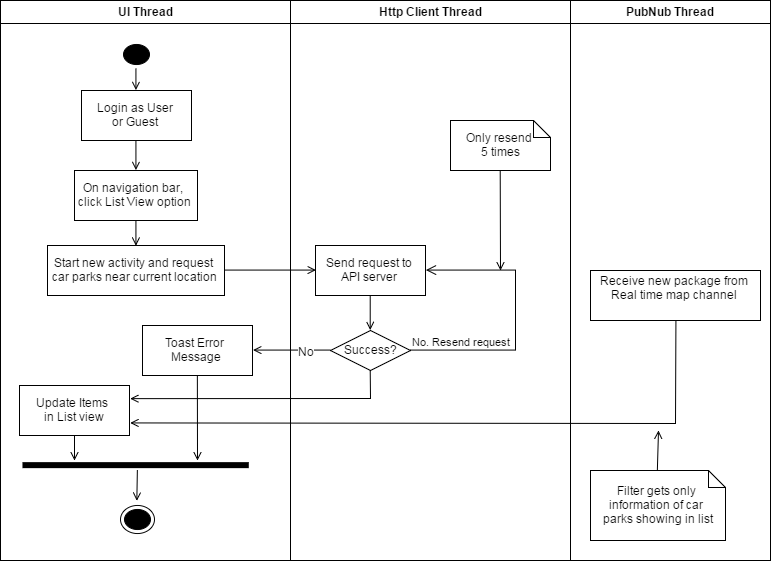


Figure : Interaction diagram - Load car parks in list

##### Car park detail information

**Summary:** This diagram shows the process of app loads car park detail information from API server and display it, update its available lots in real time and other action user can perform on the detail page.

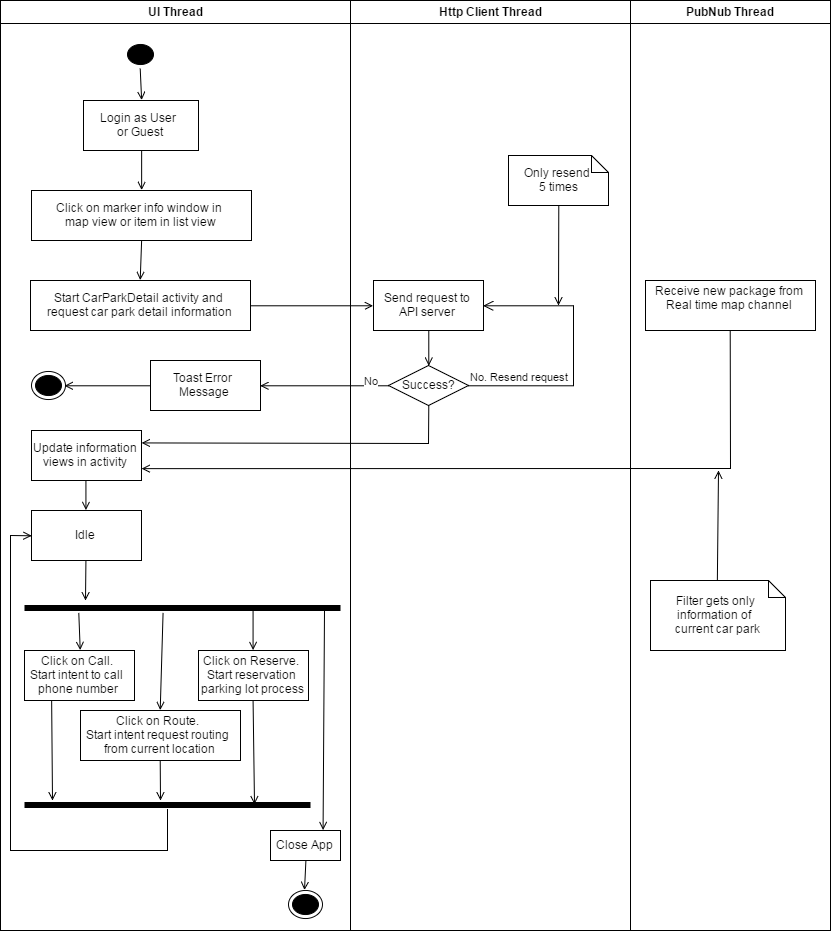


Figure : Interaction diagram - Car park detail information

##### Reserve parking lot process

**Summary:** This diagram shows the process of reserving parking lot, and receive notification about the reservation.

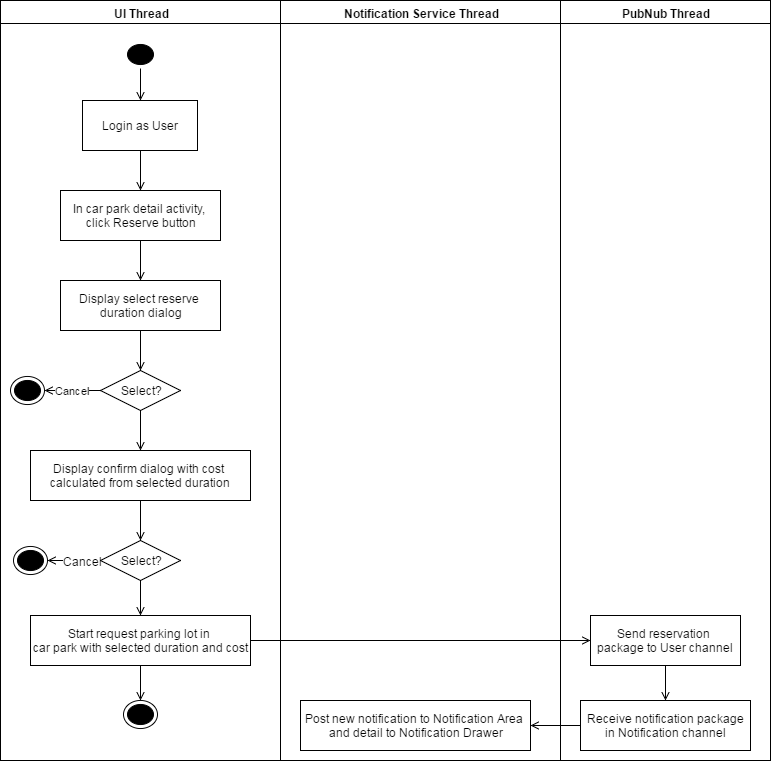


Figure : Interaction diagram - Reserve parking lot process

##### Transaction controller

**Summary:** This diagram shows the process of getting all transactions of user from API server and perform check-in, cancel process.

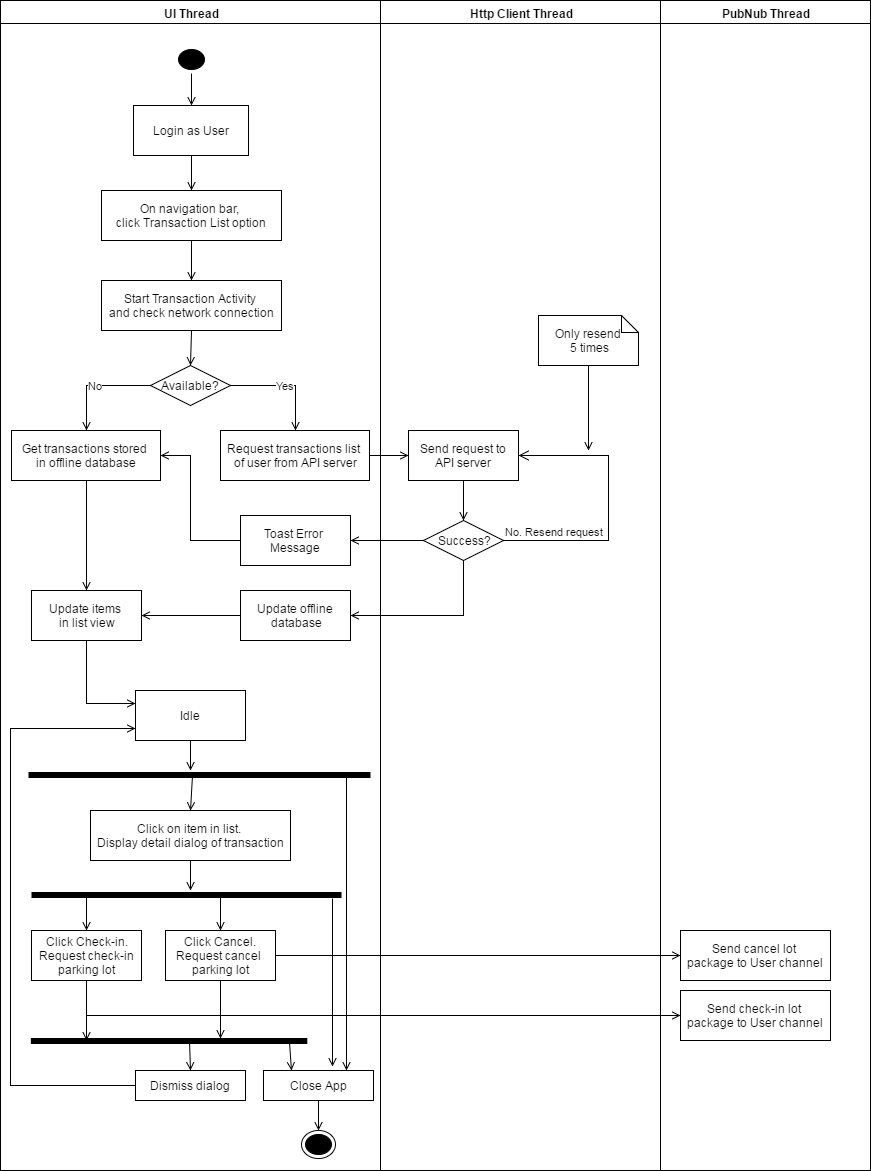


Figure : Interaction diagram - Transaction controller

##### Manage car parks

**Summary:** This diagram shows the process of app loads car parks from API server and display them in list from, update their available lots in real time, manage information and update to server and hub.

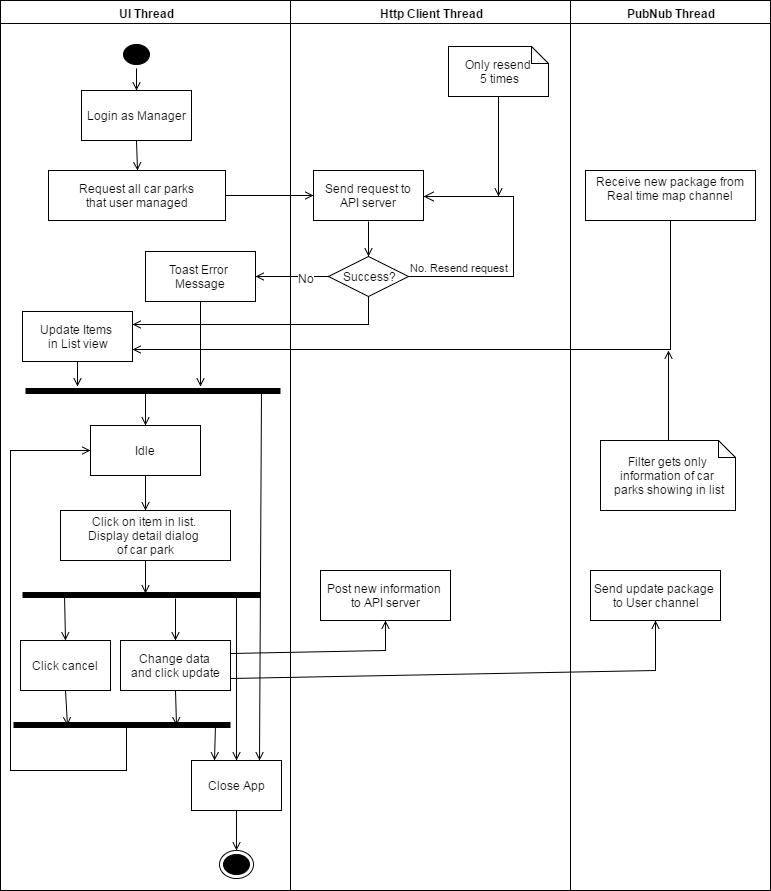


Figure : Interaction diagram - Manage car parks

##### Manage areas

**Summary:** This diagram shows the process of app loads areas from API server and display them in list from, manage information and update to server and hub.

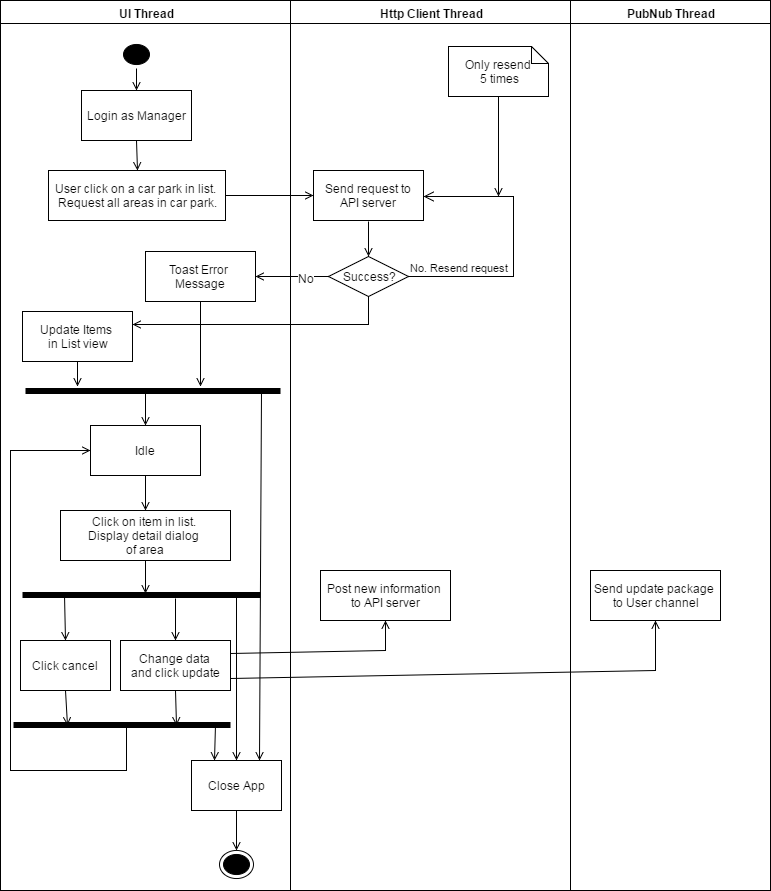


Figure : Interaction diagram - Manage areas

##### Manage parking lots

**Summary:** This diagram shows the process of app loads parking lot from API server and display them in list from, manage information and update to server and hub.

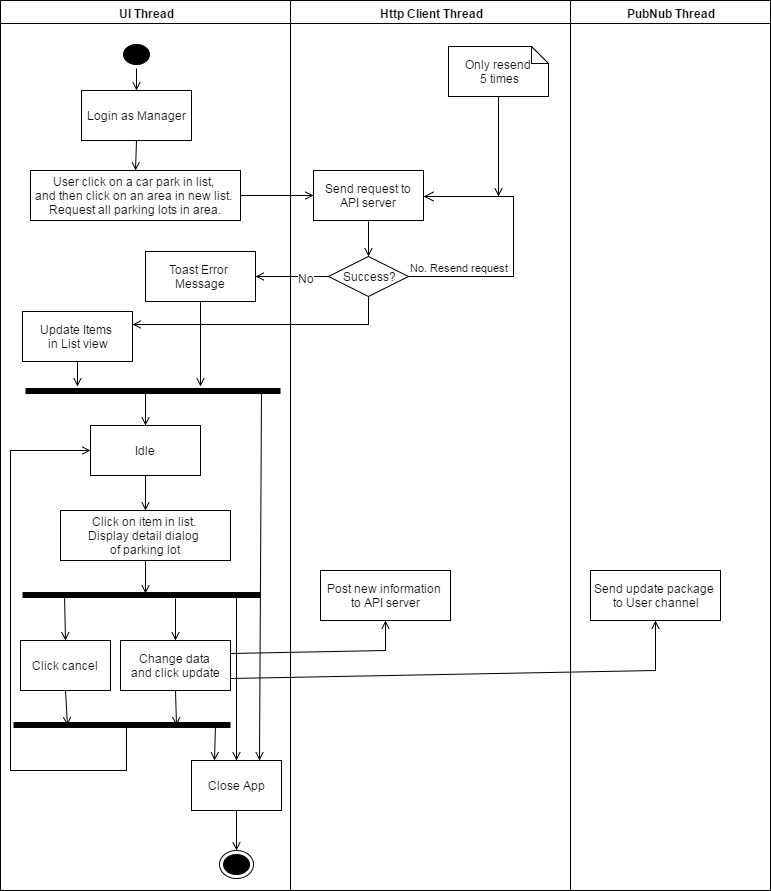


Figure : Interaction diagram - Manage parking lots

##### Check reservation PIN code

**Summary:** This diagram shows the process of how manager of car park can help user check-in their reservation.

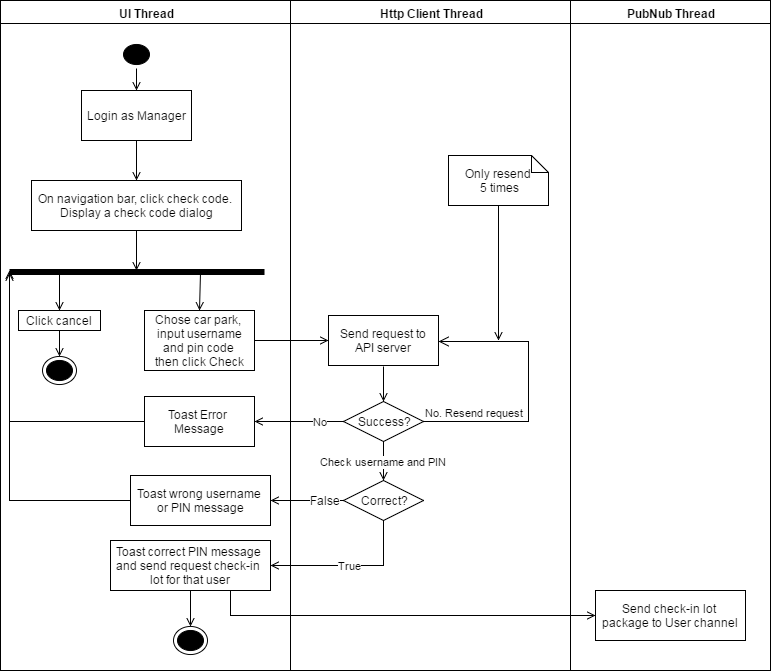


Figure : Interaction diagram - Check reservation PIN code

### Embedded Program Detailed Description

#### Class Diagram

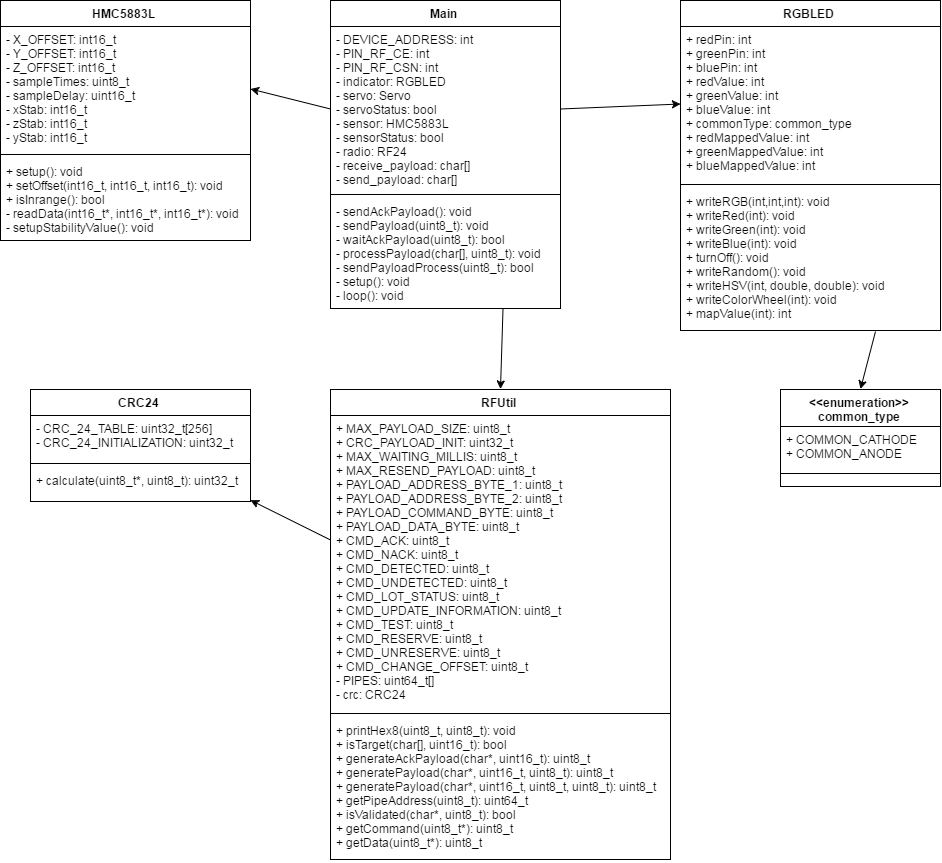


Figure : Lot node class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| Main | Core program of node |
| RFUtil | Contain necessary information and methods support RF module |
| CRC24 | Contain the checksum method for RFUtil |
| HMC5883L | Provides methods to use the magnetometer |
| RGBLED | Provides methods to control LED RGB |
| common\_type | Enumeration contains type of LED RGB |

Table : Lot node class dictionary

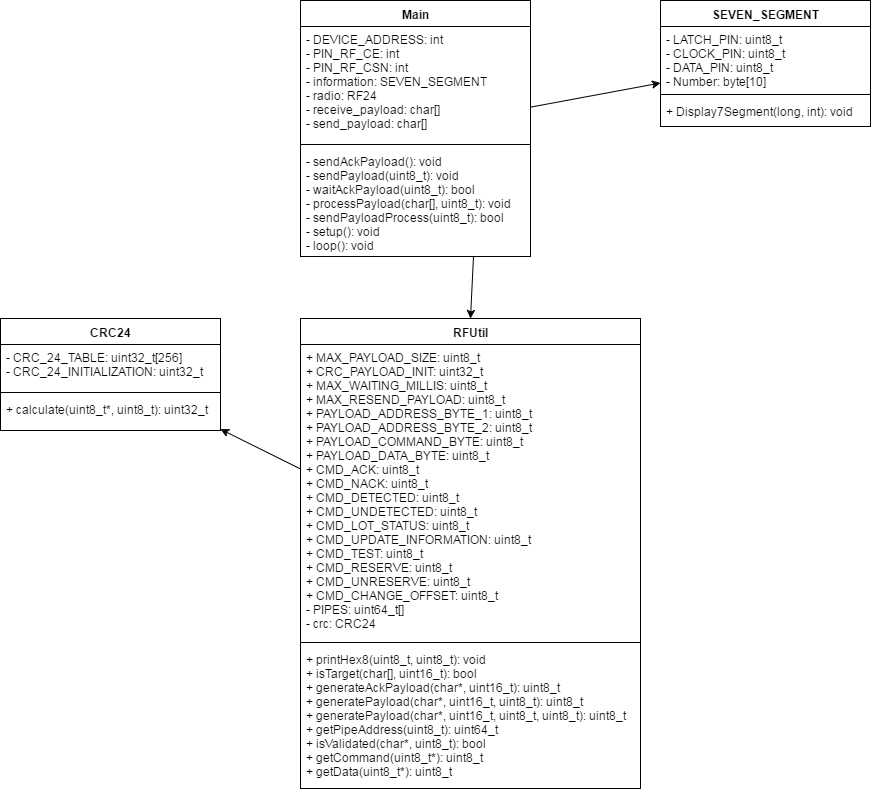


Figure : Sign node class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| Main | Core program of node |
| RFUtil | Contain necessary information and methods support RF module |
| CRC24 | Contain the checksum method for RFUtil |
| SEVEN\_SEGMENT | Provide methods to use 7-segments sign |

Table : Sign node class dictionary

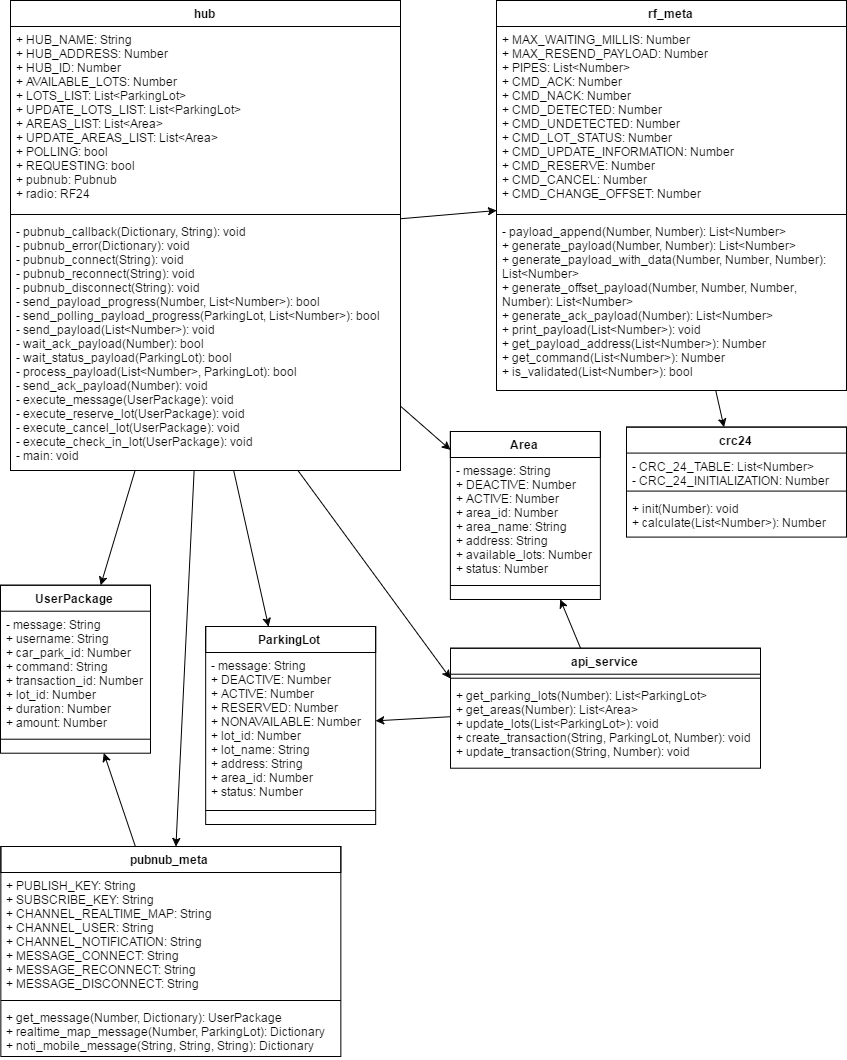


Figure : Hub class diagram

| **Class Dictionary: describes class** | |
| --- | --- |
| **Class Name** | **Description** |
| hub | Main program of Hub |
| rf\_meta | Contain information and necessary methods for RF |
| crc24 | Contain the checksum method for rf\_meta |
| ParkingLot | Information of parking lot |
| Area | Information of area |
| UserPackage | Information of user message in PubNub |
| api\_service | Contain methods for API service |
| pubnub\_meta | Contain methods for PubNub service |

Table : Hub class dictionary

#### Class Diagram Explanation

##### [Lot] Main

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| DEVICE\_ADDRESS | int | private | Address of node |
| PIN\_RF\_CE | int | private | Pin used for CE pin of RF |
| PIN\_RF\_CSN | int | private | Pin used for CSN of RF |
| indicator | RGBLED | private | Instance of RGBLED |
| servo | Servo | private | Instance of Servo |
| servoStatus | bool | private | Flag to check if servo is running |
| sensor | HMC5883L | private | Instance of HMC5883L |
| sensorStatus | bool | private | Flag to check if the sensor detect car nearby |
| receive\_payload | char[] | private | Received payload from RF |
| send\_payload | char[] | private | Payload that will be send using RF |

Table : [Lot] Main attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| sendAckPayload | void | private | Send ACK payload to Hub |
| sendPayload | void | private | Send payload to Hub |
| waiAckPayload | void | private | Wait for ACK payload from Hub |
| processPayload | void | private | Check received payload and perform work depend on the command |
| sendPayloadProcess | bool | private | Perform the process to send payload |
| setup | void | private | Run the setup whenever device start or reset |
| loop | void | private | Repeat until device stop |

Table : [Lot] Main methods

##### [Lot] HMC5883L

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| X\_OFFSET | int16\_t | private | Offset of X axis |
| Y\_OFFSET | int16\_t | private | Offset of Y axis |
| Z\_OFFSET | int16\_t | private | Offset of Z axis |
| sampleTimes | uint8\_t | private | The number of times to sample to get stabilized axis |
| sampleDelay | uint16\_t | private | The delay between sample |
| xStab | int16\_t | private | Stabilized value of X axis |
| yStab | int16\_t | private | Stabilized value of Y axis |
| zStab | int16\_t | private | Stabilized value of Z axis |

Table : [Lot] HMC5883L attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| setup | void | public | Run necessary methods to make the module work |
| setOffset | void | public | Set offset for the module to check if car is in range |
| isInRange | bool | public | Check if car is in range |
| readData | void | private | Get x,y,z axis from the module |
| setupStabilityValue | void | private | Perform sample multiple times to calculate stabilized axis |

Table : [Lot] HMC5883L methods

##### [Lot] RGBLED

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| redPin | int | public | LED pin |
| greenPin | int | public | LED pin |
| bluePin | int | public | LED pin |
| redValue | int | public | RGB value |
| greenValue | int | public | RGB value |
| blueValue | int | public | RGB value |
| commonType | common\_type | public | LED type |
| redMappedValue | int | public | RGB value mapped based on commonType |
| greenMappedValue | int | public | RGB value mapped based on commonType |
| blueMappedValue | int | public | RGB value mapped based on commonType |

Table : [Lot] RGBLED attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| writeRGB | void | public | Set all pins values |
| writeRed | void | public | Set just the red pin |
| writeGreen | void | public | Set just the green pin |
| writeBlue | void | public | Set just the blue pin |
| turnOff | void | public | Turn all pins off |
| writeRandom | void | public | Show a random color |
| writeHSV | void | public | Writes the LED based on HSV values |
| writeColorWheel | void | public | Show the HSV Color Wheel |
| mapValue | void | public | Map a value based on the common\_type |

Table : [Lot] RGBLED methods

##### [Lot] common\_type

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| COMMON\_CATHODE | enum | public | Value of common cathode LED type |
| COMMON\_ANODE | enum | public | Value of common anode LED type |

Table : [Lot] common\_type attributes

**Methods**

N/A

##### [Sign] Main

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| DEVICE\_ADDRESS | int | private | Address of node |
| PIN\_RF\_CE | int | private | Pin used for CE pin of RF |
| PIN\_RF\_CSN | int | private | Pin used for CSN of RF |
| information | SERVEN\_SEGMENT | private | Instance of SEVEN\_SEGMENT |
| receive\_payload | char[] | private | Received payload from RF |
| send\_payload | char[] | private | Payload that will be send using RF |

Table : [Sign] Main attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| sendAckPayload | void | private | Send ACK payload to Hub |
| sendPayload | void | private | Send payload to Hub |
| waiAckPayload | void | private | Wait for ACK payload from Hub |
| processPayload | void | private | Check received payload and perform work depend on the command |
| sendPayloadProcess | bool | private | Perform the process to send payload |
| setup | void | private | Run the setup whenever device start or reset |
| loop | void | private | Repeat until device stop |

Table : [Sign] Main methods

##### [Sign] SEVEN\_SEGMENT

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| LATCH\_PIN | uint8\_t | private | Latch pin of 595 |
| CLOCK\_PIN | uint8\_t | private | Clock pin of 595 |
| DATA\_PIN | uint8\_t | private | Data pin of 595 |
| Number | byte[10] | private | Binary array for display 7-segments led using 595 |

Table : [Sign] SEVEN\_SEGMENT attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| Display7Segment | void | public | Display available lot number to 7-segmens sign |

Table : [Sign] SEVEN\_SEGMENT methods

##### [Lot/Sign] RFUtil

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| MAX\_PAYLOAD\_SIZE | uint8\_t | public | Max size of payload |
| CRC\_PAYLOAD\_INIT | uint32\_t | public | Initialize value to use with CRC24 |
| MAX\_WAITING\_MILLIS | uint8\_t | public | Max waiting ACK time in millis |
| MAX\_RESEND\_PAYLOAD | uint8\_t | public | Max resend times |
| PAYLOAD\_ADDRESS\_  BYTE\_1 | uint8\_t | public | Position of 1st address byte |
| PAYLOAD\_ADDRESS\_  BYTE\_2 | uint8\_t | public | Position of 2nd address byte |
| PAYLOAD\_COMMAND\_  BYTE | uint8\_t | public | Position of command byte |
| PAYLOAD\_DATA\_BYTE | uint8\_t | public | Position of data byte |
| CMD\_ACK | uint8\_t | public | ACK command byte |
| CMD\_NACK | uint8\_t | public | NACK command byte |
| CMD\_DETECTED | uint8\_t | public | Detected command byte |
| CMD\_UNDETECTED | uint8\_t | public | Undetected command byte |
| CMD\_LOT\_STATUS | uint8\_t | public | Lot status command byte |
| CMD\_UPDATE  \_INFORMATION | uint8\_t | public | Update information command byte |
| CMD\_TEST | uint8\_t | public | Test command byte |
| CMD\_RESERVE | uint8\_t | public | Reserve command byte |
| CMD\_UNRESERVE | uint8\_t | public | Unreserve command byte |
| CMD\_CHANGE\_OFFSET | uint8\_t | public | Change offset command byte |
| PIPES | uint64\_t[] | private | List of pipes address |
| crc | CRC24 | private | Instance of CRC24 |

Table : [Lot/Sign] RFUtil attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| printHex8 | void | public | Print payload as hex format |
| isTarget | bool | public | Check if the target of payload is current device |
| generateAckPayload | uint8\_t | public | Create ACK payload. Return payload size |
| generatePayload | uint8\_t | public | Create payload. Return payload size |
| getPipeAddress | uint64\_t | public | Return pipe address from position |
| isValidated | bool | public | Check if payload satisfy CRC |
| getCommand | uint8\_t | public | Return command byte from payload |
| getData | uint8\_t | public | Return data byte from payload |

Table : [Lot/Sign] RFUtil methods

##### [Lot/Sign] CRC24

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| CRC\_24\_TABLE | uint32\_t[256] | private | Table of CRC24 defined value |
| CRC\_24  \_INITIALIZATION | uint32\_t | private | Initialization value for crc calculate |

Table :: [Lot/Sign] CRC24 attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| calculate | uint32\_t | public | Return the calculated checksum |

Table : [Lot/Sign] CRC24 methods

##### [Hub] hub

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| HUB\_NAME | String | public | Name of car park |
| HUB\_ADDRESS | Number | public | 2 bytes address of hub |
| HUB\_ID | Number | public | Id of car park |
| AVAILABLE\_LOTS | Number | public | Current available lots in car park |
| LOTS\_LIST | List<ParkingLot> | public | List of parking lots |
| UPDATE\_  LOTS\_LIST | List<ParkingLot> | public | List of lots need to update |
| AREAS\_LIST | List<Area> | public | List of areas |
| UPDATE\_  AREAS\_LISST | List<Area> | public | List of areas need to update |
| POLLING | bool | public | Flag to check if the main progress is polling |
| REQUESTING | bool | public | Flag to check if there is a need to change main progress to execute message |
| pubnub | Pubnub | public | Instance of Pubnub |
| radio | RF24 | public | Insantce of RF24 |

Table : [Hub] hub attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| pubnub\_callback | void | private | Callback when a pubnub message is received |
| pubnub\_error | void | private | Callback when there is a error in sending pubnub message |
| pubnub\_connect | void | private | Callback when successfully connect to pubnub channel |
| pubnub\_reconnect | void | private | Callback when need to reconnect channel |
| pubnub\_disconnect | void | private | Callback when the connection with pubnub channel is cut |
| send\_payload\_progress | bool | private | Progress to send payload |
| send\_polling\_  payload\_progress | bool | private | Progress to send polling payload |
| send\_payload | void | private | Send payload using RF module |
| wait\_ack\_payload | bool | private | Wait for ack payload from target |
| wait\_status\_payload | bool | private | Wait for update status from lot |
| process\_payload | bool | private | Check and process received payload |
| send\_ack\_payload | void | private | Send ACK payload to target |
| execute\_message | void | private | Check and execute received pubnub message |
| execute\_reserve\_lot | void | private | Process to perform reserve lot action |
| execute\_cancel\_lot | void | private | Process to perform cancel lot action |
| execute\_check\_in\_lot | void | private | Process to perform check in lot action |
| main | void | private | Perform program of main thread |

Table : [Hub] hub methods

##### [Hub] rf\_meta

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| MAX\_WAITING  \_MILLIS | Number | public | Max wait times for ACK |
| MAX\_RESEND\_  PAYLOAD | Number | public | Max times to resend payload |
| PIPES | List<Number> | public | List of pipes address for RF |
| CMD\_ACK | Number | public | Byte of ACK command |
| CMD\_NACK | Number | public | Byte of NACK command |
| CMD\_DETECTED | Number | public | Byte of detected command |
| CMD\_UNDETECTED | Number | public | Byte of undetected command |
| CMD\_LOT\_STATUS | Number | public | Byte of lot status command |
| CMD\_UPDATE\_  INFORMATION | Number | public | Byte of update information command |
| CMD\_RESERVE | Number | public | Byte of reserve command |
| CMD\_CANCEL | Number | public | Byte of cancel command |
| CMD\_CHANGE\_  OFFSET | Number | public | Byte of change offset command |

Table : [Hub] rf\_meta attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| payload\_append | List<Number> | private | Add byte to payload |
| generate\_payload | List<Number> | public | Create payload with target address and command |
| generate\_payload\_  with\_data | List<Number> | public | Create payload with target address, command and data byte |
| generate\_offset\_  payload | List<Number> | public | Create special payload to update offset |
| generate\_ack\_  payload | List<Number> | public | Create ACK payload |
| print\_payload | void | public | Print payload in hex form |
| get\_payload\_address | Number | public | Get address from payload |
| get\_command | Number | public | Get command from payload |
| is\_validated | bool | public | Check checksum of payload |

Table : [Hub] rf\_meta methods

##### [Hub] crc24

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| CRC\_24\_TABLE | List<Number> | private | Table of CRC24 defined value |
| CRC\_24  \_INITIALIZATION | Number | private | Initialization value for crc calculate |

Table : [Hub] crc24 attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| init | void | public | Set crc24 initialization to another value |
| calculate | Number | public | Return the calculated checksum |

Table : [Hub] crc24 methods

##### [Hub] api\_service

**Attributes**

N/A

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| get\_parking\_lots | List<ParkingLot> | public | Get all active lots of car park from API server |
| get\_areas | List<Area> | public | Get all active areas of car park from API server |
| update\_lots | void | public | Update lot information using API |
| create\_transaction | void | public | Create new transaction using API |
| update\_transaction | void | public | Update transaction status using API |

Table : [Hub] api\_service methods

##### [Hub] pubnub\_meta

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| PUBLISH\_KEY | String | public | Publish key for pubnub |
| SUBSCRIBE\_KEY | String | public | Subscribe key for pubnub |
| CHANNEL  \_REALTIME\_MAP | String | public | Key name of realtime map channel |
| CHANNEL\_USER | String | public | Key name of user channel |
| CHANNEL  \_NOTIFICATION | String | public | Key name of notification channel |
| MESSAGE \_CONNECT | String | public | Value of connect message |
| MESSAGE  \_RECONNECT | String | public | Value of reconnect message |
| MESSAGE  \_DISCONNECT | String | public | Value of disconnect message |

Table : [Hub] pubnub\_meta attributes

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| get\_message | UserPackage | public | Return user package from json |
| realtime\_map  \_message | Dictionary | public | Create a json for realtime map channel |
| noti\_mobile  \_message | Dictionary | public | Create a json for notification channel |

Table : [Hub] pubnub\_meta methods

##### [Hub] Area

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| message | String | private | Format to string of object |
| DEACTIVE | Number | public | Value of deactive status |
| ACTIVE | Number | public | Value of active status |
| area\_id | Number | public | Unique Id of area |
| area\_name | String | public | Name of area |
| address | Number | public | Byte address of sign node |
| availables\_lots | Number | public | Current available lots in area |
| status | Number | public | Status of area |

Table : [Hub] Area attributes

**Methods**

N/A

##### [Hub] ParkingLot

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| message | String | private | Format to string of object |
| DEACTIVE | Number | public | Value of deactive status |
| ACTIVE | Number | public | Value of active status |
| RESERVED | Number | public | Value of reserved status |
| NONAVAILABLE | Number | public | Value of non-available status |
| lot\_id | Number | public | Unique id of parking lot |
| lot\_name | String | public | Name of parking lot |
| address | Number | public | Byte address of lot node |
| area\_id | Number | public | Unique Id of area contains lot |
| status | Number | public | Status of lot |

Table : [Hub] ParkingLot attributes

**Methods**

N/A

##### [Hub] UserPackage

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| message | String | private | Format to string of object |
| username | String | public | Username send the request |
| car\_park\_id | Number | public | Id of car park |
| command | String | public | Command of package |
| transaction\_id | Number | public | Transaction Id if command is cancel or check in transaction |
| lot\_id | Number | public | Lot node id if command is cancel or check in |
| duration | Number | public | Duration of the reservation if command is reserve |
| amount | Number | public | Cost of the reservation if command is reserve |

Table : [Hub] UserPackage attributes

**Methods**

N/A

#### Interaction Diagram

##### Lot node process

**Summary:** This diagram shows the process of the MCU in lot node detects car, transmits and receives data from RF network and controls other actuator.

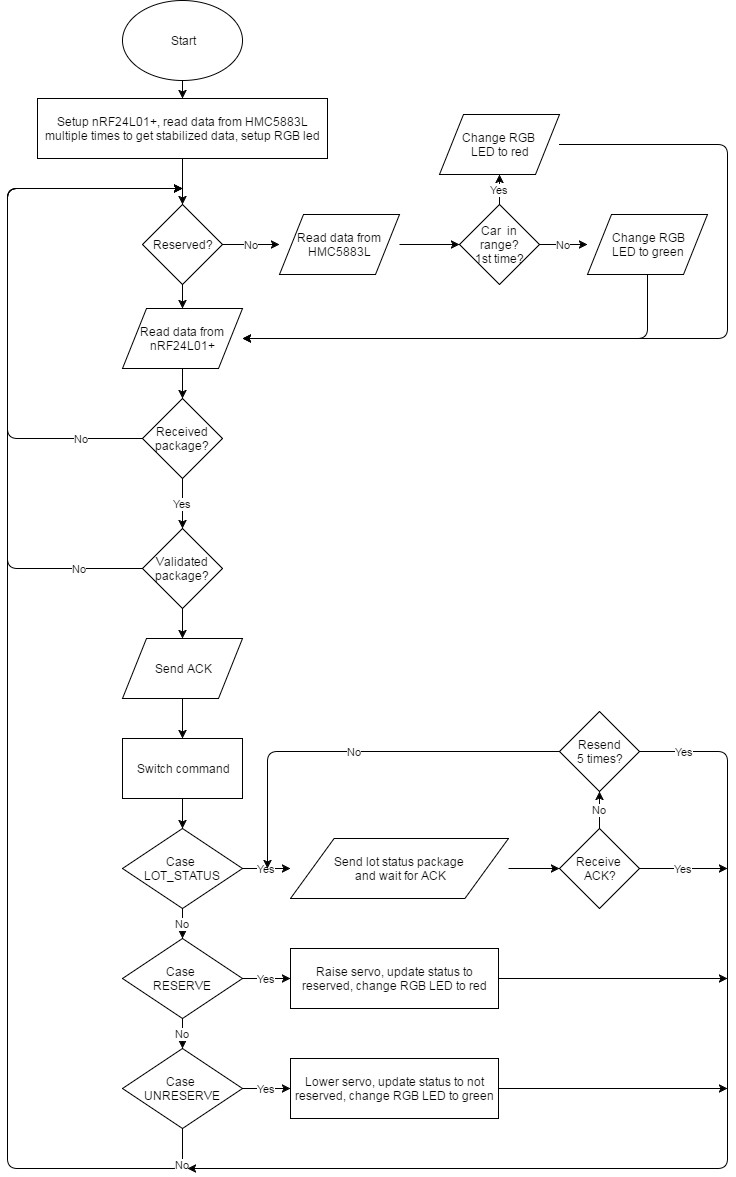


Figure : Interaction diagram - Lot node process

##### Sign node process

**Summary:** This diagram shows the process of the MCU in sign node transmits and receives data from RF network and controls led board.

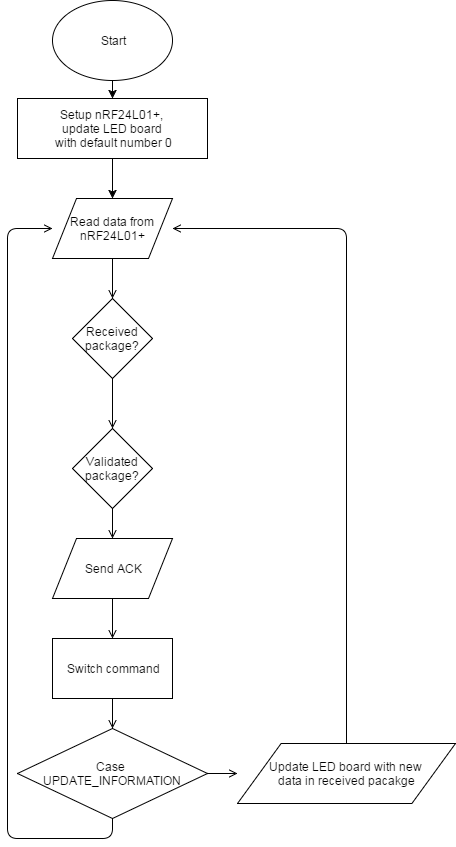


Figure : Interaction diagram - Sign node

##### Hub polling process

**Summary:** This diagram shows the process of the hub performs polling to get new data from lot nodes, update sign nodes if necessary, send updated information to API server and PubNub server

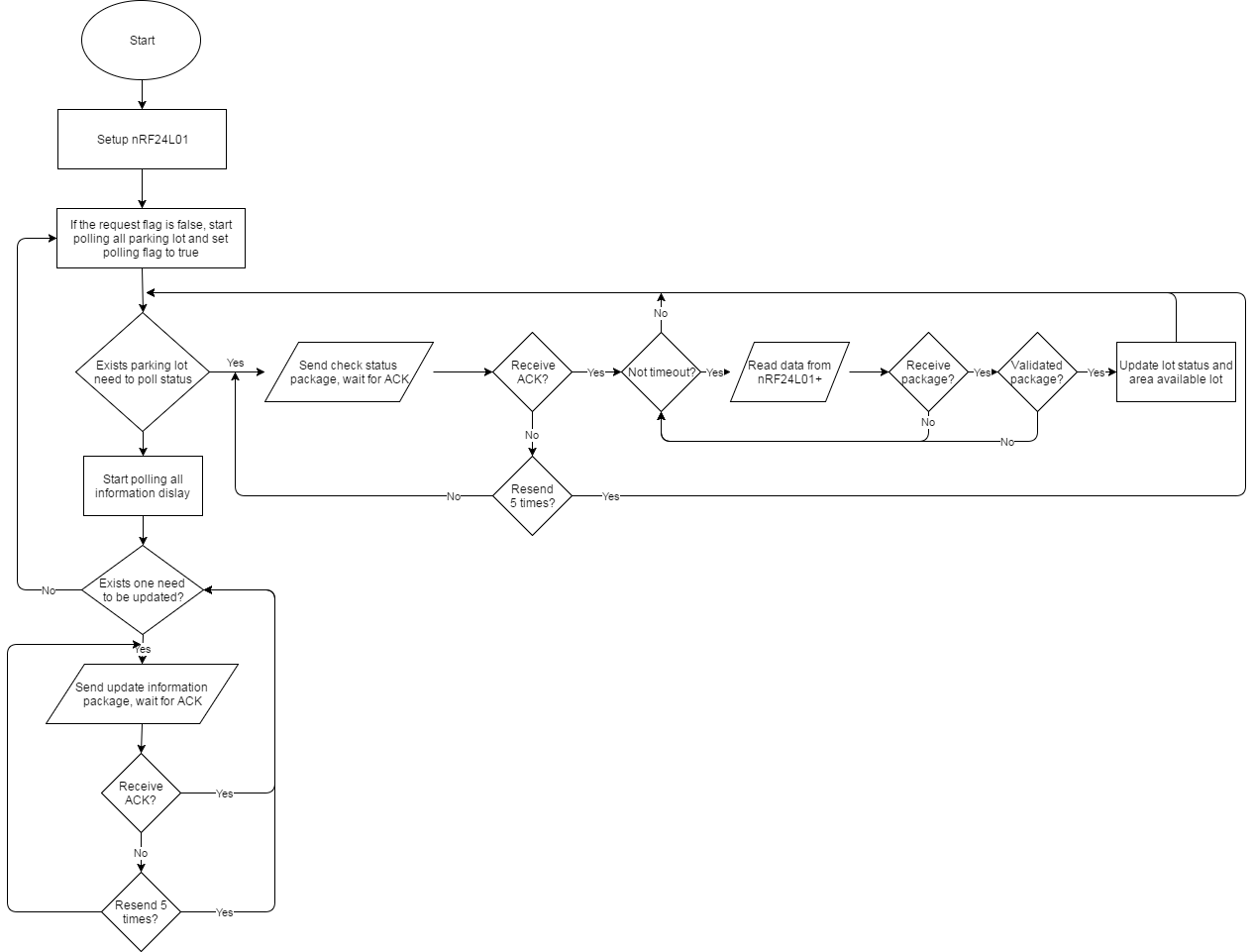


Figure : Interaction diagram - Hub polling process

##### Hub execute request process

**Summary:** This diagram shows the process of the hub receives requests from PubNub server and execute them.

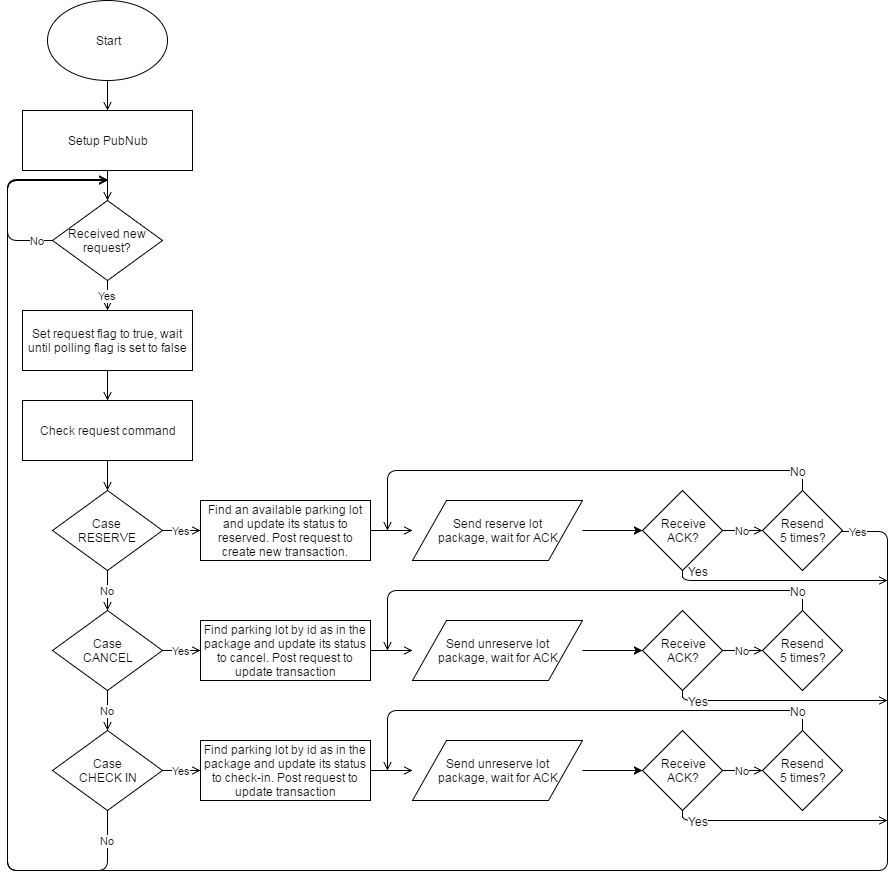


Figure : Interaction diagram - Hub request process

### Hardware Detailed Description

#### Raspberry Pi 3 model B

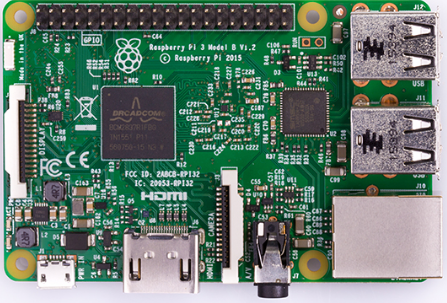


Figure : Raspberry Pi 3 model B

The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016.

The Raspberry Pi 3 is a low cost, credit-card size computer that includes 802.11n Wi-Fi and Bluetooth 4.1. Even though it requires lots of additional hardware to function as a full PC with limited operating system selection, with the introduction of wireless connectivity and a boost in performance over its previous iteration, make the Raspberry Pi 3 Model B appropriate for a wider variety of projects – and it still costs just $35.

**PROS:**

* Great value
* Significant performance increase
* Great free online learning resources
* Built-in wireless connections

**CONS:**

* No Android support yet

**SPECS:**

* Quad-core 1.2GHz Broadcom BCM2837RIFBG SoC
* 400MHz VideoCore IV graphics
* 1GB RAM
* Single-band 802.11n Wi-Fi
* Bluetooth 4.1
* 4 x USB 2
* 40 x GPIO pins
* 10/100 Ethernet
* MicroSD slot
* 86 x 56 x 21mm (WDH)
* 45g

**PINOUT:**

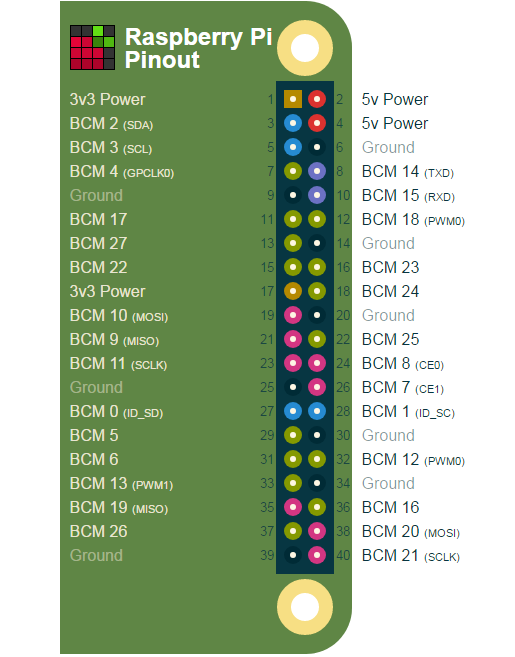


Figure : Raspberry Pi 3 Model B pinount

#### Arduino Nano

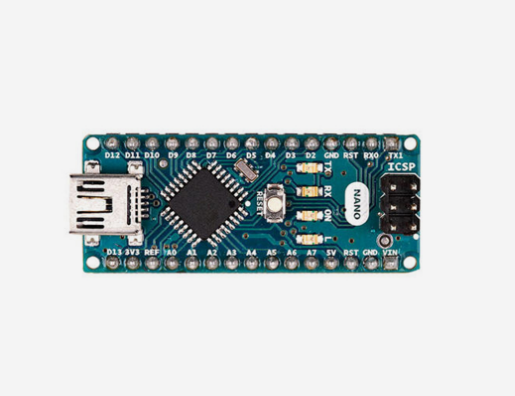


Figure : Arduino Nano

Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a small, complete, and breadboard-friendly board based on ATmega328 (Arduino Nano 3.x).

**SPECS:**

|  |  |
| --- | --- |
| Microcontroller | Atmel ATmega328 |
| Operating Voltage (logic level) | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 8 |
| DC Current per I/O Pin | 40mA |
| Flash Memory | 32KB (of which 2 KB used by bootloader) |
| SRAM | 2KB |
| EEPROM | 1KB |
| Clock Speed | 16MHz |
| Dimensions | 0.70” x 1.70” |

Figure : Arduino Nano specifications

**FEATURES:**

* Automatic reset during program download
* Power OK blue LED
* Green (TX), red (RX) and orange (L) LED
* Auto sensing/switching power input
* Small mini-B USB for programming and serial monitor
* ICSP header for direct programming and serial monitor
* Standard 0.1” spacing DIP (breadboard friendly)
* Manual reset switch

**POWER**:

The Arduino Nano can be powered via the mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

**PINOUT:**

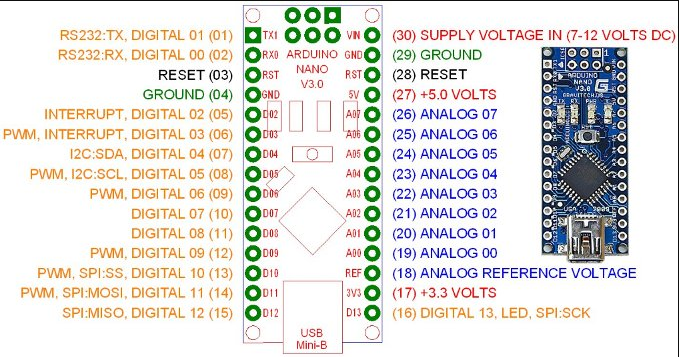


Figure : Arduino Nano pinout

#### Compass module 3-Axis HMC5883L

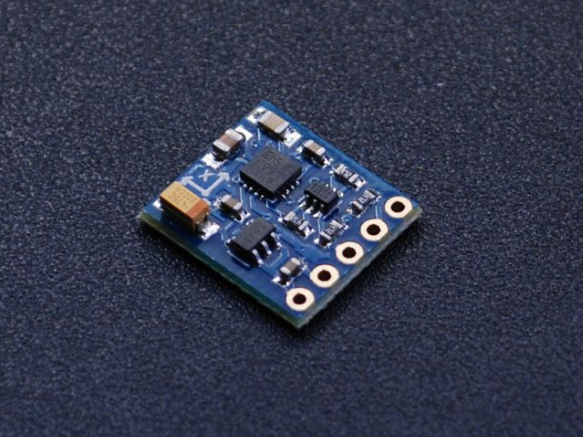


Figure : 3-Axis HMC5883L

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low cost compassing and magnetometry.

**FEATURES:**

* 3-Axis magnetoresistive sensors and ASIC in a 3.0 x 3.0 x 0.9mm LCC surface mount package.
* 12-Bit ADC coupled with low noise AMR sensors achieves 2 milli-gauss field resolution in ±8 Gauss Fields.
* Built-in self-test.
* Low voltage operations (2.16 to 3.6V) and low power consumption (100μA).
* Built-in strap drive circuits.
* I2C digital interface.
* Lead free package construction.
* Fast 160Hz maximum output rate.

**PINOUT:**

* **VIN:** this is the power in pin, give it 3-5VDC. It is reverse-polarity protected. Use the same voltage as you do for the controlling microcontroller's logic.
* **GND:** this is the common power and data ground pin.
* **SDA** and **SCL:** these are the I2C data and clock pins used to send and receive data from the module to your microcontroller. There are 10K pull-ups on these pins to the VIN pin. You can connect these pins to 5V I2C lines, there are level shifters on board to safely bring the pins down to 3V.
* **RDY:** this is the 'data ready' pin output. If you want to stream data at high speed (higher than 100 times a second) you may want to listen to this pin for when data is ready to be read.

**ANISOTROPIC MAGNETO-RESISTIVE SENSORS**

The Honeywell HMC5883L magneto-resistive sensor circuit is a trio of sensors and application specific support circuits to measure magnetic fields. With power supply applied, the sensor converts any incident magnetic field in the sensitive axis directions to a differential voltage output. The magneto-resistive sensors are made of a nickel-iron (permalloy) thin-film and patterned as a resistive strip element. In the presence of a magnetic field, a change in the bridge resistive elements causes a corresponding change in voltage across the bridge outputs.

These resistive elements are aligned together to have a common sensitive axis (indicated by arrows in the pinout diagram) that will provide positive voltage change with magnetic fields increasing in the sensitive direction. Because the output is only proportional to the magnetic field component along its axis, additional sensor bridges are placed at orthogonal directions to permit accurate measurement of magnetic field in any orientation.

**MODES OF OPERATION**

* **Continuous-Measurement Mode:** the device continuously makes measurements, at user selectable rate, and places measured data in data output registers.
* **Single-Measurement Mode:** this is the default power-up mode. During single-measurement mode, the device makes a single measurement and places the measured data in data output registers. After the measurement is complete and output data registers are updated, the device is placed in idle mode.
* **Idle Mode:** during this mode the device is accessible through the I2C bus, but major sources of power consumption are disabled, such as, but not limited to, the ADC, the amplifier, and the sensor bias current. All registers maintain values while in idle mode. The I2C bus is enabled for use by other devices on the network while in idle mode.

**REGISTER LIST**

|  |  |  |
| --- | --- | --- |
| **Address Location** | **Name** | **Access** |
| 00 | Configuration Register A | Read/Write |
| 01 | Configuration Register B | Read/Write |
| 02 | Mode Register | Read/Write |
| 03 | Data Output X MSB Register | Read |
| 04 | Data Output X LSB Register | Read |
| 05 | Data Output Z MSB Register | Read |
| 06 | Data Output Z LSB Register | Read |
| 07 | Data Output Y MSB Register | Read |
| 08 | Data Output Y LSB Register | Read |
| 09 | Status Register | Read |
| 10 | Identification Register A | Read |
| 11 | Identification Register B | Read |
| 12 | Identification Register C | Read |

Table : HMC55883L Register list

#### RF module nRF24L01+



Figure : RF module nRF24L01+

The nRF24L01+ is a single chip 2.4GHz transceiver with an embedded baseband protocol engine (Enhanced ShockBurst™), suitable for ultra-low power wireless applications. The nRF24L01+ is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz.

**FEATURES:**

* Worldwide 2.4GHz ISM band operation
* 250kbps, 1Mbps and 2Mbps on air data rates
* Ultra low power operation
* 11.3mA TX at 0dBm output power
* 13.5mA RX at 2Mbps air data rate
* 900nA in power down
* 26µA in standby-I
* On chip voltage regulator
* 1.9 to 3.6V supply range
* Enhanced ShockBurst™
* Automatic packet handling
* Auto packet transaction handling
* 6 data pipe MultiCeiver™
* Drop-in compatibility with nRF24L01
* On-air compatible in 250kbps and 1Mbps with nRF2401A, nRF2402, nRF24E1 and nRF24E2
* Low cost BOM
* ±60ppm 16MHz crystal
* 5V tolerant inputs
* Compact 20-pin 4x4mm QFN package

**PINOUT:**

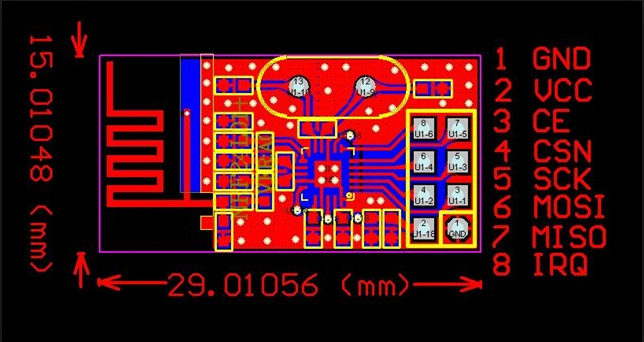


Figure : nRF24L01+ pinout

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | **Name** | **Pin function** | **Description** |
| 1 | GND | Power | Ground (0V) |
| 2 | VCC | Power | Power Supply (+1.9V - +3.6V DC) |
| 3 | CE | Digital Input | Chip Enable Actives RX or TX mode |
| 4 | CSN | Digital Input | SPI Chip Select |
| 5 | SCK | Digital Input | SPI Clock |
| 6 | MOSI | Digital Input | SPI Slave Data Input |
| 7 | MISO | Digital Output | SPI Slave Data Output, with tri-state option |
| 8 | IRQ | Digital Output | Maskable interrupt pin. Active low. |

Table : nRF24L01+ pinout

**MODES OF OPERATION**

* **Power down mode:** In power down mode nRF24L01+ is disabled using minimal current consumption. All register values available are maintained and the SPI is kept active, enabling change of configuration and the uploading/downloading of data registers.
* **Standby-I mode:** By setting the PWR\_UP bit in the CONFIG register to 1, the device enters standby-I mode. Standby-I mode is used to minimize average current consumption while maintaining short start up times. In this mode only part of the crystal oscillator is active. Change to active modes only happens if CE is set high and when CE is set low, the nRF24L01 returns to standby-I mode from both the TX and RX modes.
* **Standby-II mode:** In standby-II mode extra clock buffers are active and more current is used compared to standby-I mode. nRF24L01+ enters standby-II mode if CE is held high on a PTX device with an empty TX FIFO. If a new packet is uploaded to the TX FIFO, the PLL immediately starts and the packet is transmitted after the normal PLL settling delay (130µs).
* **RX mode:** The RX mode is an active mode where the nRF24L01+ radio is used as a receiver. To enter this mode, the nRF24L01+ must have the PWR\_UP bit, PRIM\_RX bit and the CE pin set high. In RX mode the receiver demodulates the signals from the RF channel, constantly presenting the demodulated data to the baseband protocol engine. The baseband protocol engine constantly searches for a valid packet. If a valid packet is found (by a matching address and a valid CRC) the payload of the packet is presented in a vacant slot in the RX FIFOs. If the RX FIFOs are full, the received packet is discarded.
* **TX mode:** The TX mode is an active mode for transmitting packets. To enter this mode, the nRF24L01+ must have the PWR\_UP bit set high, PRIM\_RX bit set low, a payload in the TX FIFO and a high pulse on the CE for more than 10µs. The nRF24L01+ stays in TX mode until it finishes transmitting a packet.

**PACKET FORMAT:**



Figure : nRF24L01+ packet format

* **Preamble:** The preamble is a bit sequence used to synchronize the receiver’s demodulator to the incoming bit stream. The preamble is one byte long and is either 01010101 or 10101010. If the first bit in the address is 1 the preamble is automatically set to 10101010 and if the first bit is 0 the preamble is automatically set to 01010101. This is done to ensure there are enough transitions in the preamble to stabilize the receiver.
* **Address:** This is the address for the receiver. An address ensures that the packet is detected and received by the correct receiver, preventing accidental cross talk between multiple nRF24L01+ systems.
* **Packet Control Field:** The packet control field contains a 6 bit **payload length** field, a 2 bit **PID** (Packet Identity) field and a 1 bit **NO\_ACK flag**. **Payload length** specifies the length of the payload in bytes. **PID** is used to detect if the received packet is new or retransmitted. **NO\_ACK flag** is only used when the auto acknowledgement feature is used.
* **Payload:** The payload is the user defined content of the packet. It can be 0 to 32 bytes wide and is transmitted on-air when it is uploaded to nRF24L01+.
* **CRC:** The CRC is the mandatory error detection mechanism in the packet. It is either 1 or 2 bytes and is calculated over the address, Packet Control Field and Payload.

**SPI COMMANDS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Command name** | **Command** | **# Data bytes** | **Operation** |
| R\_REGISTER | 000A AAAA | 1 to 5 LSByte first | Read command and status registers. AAAAA = 5 bit Register Map Address |
| W\_REGISTER | 001A AAAA | 1 to 5 LSByte first | Write command and status registers. AAAAA = 5 bit Register Map Address Executable in power down or standby modes only. |
| R\_RX\_PAYLOAD | 0110 0001 | 1 to 32 LSByte first | Read RX-payload: 1 – 32 bytes. A read operation always starts at byte 0. Payload is deleted from FIFO after it is read. Used in RX mode. |
| W\_TX\_PAYLOAD | 1010 0000 | 1 to 32 LSByte first | Write TX-payload: 1 – 32 bytes. A write operation always starts at byte 0 used in TX payload. |
| FLUSH\_TX | 1110 0001 | 0 | Flush TX FIFO, used in TX mode |
| FLUSH\_RX | 1110 0010 | 0 | Flush RX FIFO, used in RX mode Should not be executed during transmission of acknowledge, that is, acknowledge package will not be completed. |
| REUSE\_TX\_PL | 1110 0011 | 0 | Used for a PTX device  Reuse last transmitted payload. TX payload reuse is active until W\_TX\_PAYLOAD or FLUSH TX is executed. TX payload reuse must not be activated or deactivated during package transmission. |
| R\_RX\_PL\_WIDa | 0110 0000 | 1 | Read RX payload width for the top R\_RX\_PAYLOAD in the RX FIFO. |
| W\_ACK\_PAYLOADa | 1010 1PPP | 1 to 32 LSByte first | Used in RX mode. Write Payload to be transmitted together with ACK packet on PIPE PPP. (PPP valid in the range from 000 to 101). Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled using first in - first out principle. Write payload: 1– 32 bytes. A write operation always starts at byte 0. |
| W\_TX\_PAYLOAD\_NO ACKa | 1011 0000 | 1 to 32 LSByte first | Used in TX mode. Disables AUTOACK on this specific packet. |
| NOP | 1111 1111 | 0 | No Operation. Might be used to read the STATUS register |

Table : nRF24L01+ SPI commands

#### Lot node

##### RGB LED common anode



Figure : RGB LED common anode

RGB LED allows you to change the lights to any color to show state of parking slot.

**SPECS:**

* Forward Voltage (RGB): (2.0, 3.2, 3.2)V
* Max Forward Current (RGB): (20, 20, 20)mA
* Max Luminosity (RGB): (2800, 6500, 1200)mcd

**PINOUT:**

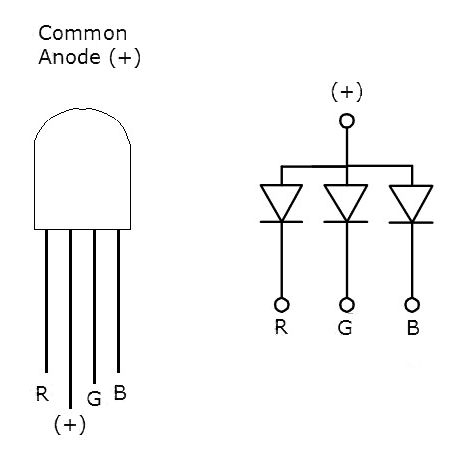


Figure : RGB LED common anode pinout

##### Transistor TIP122

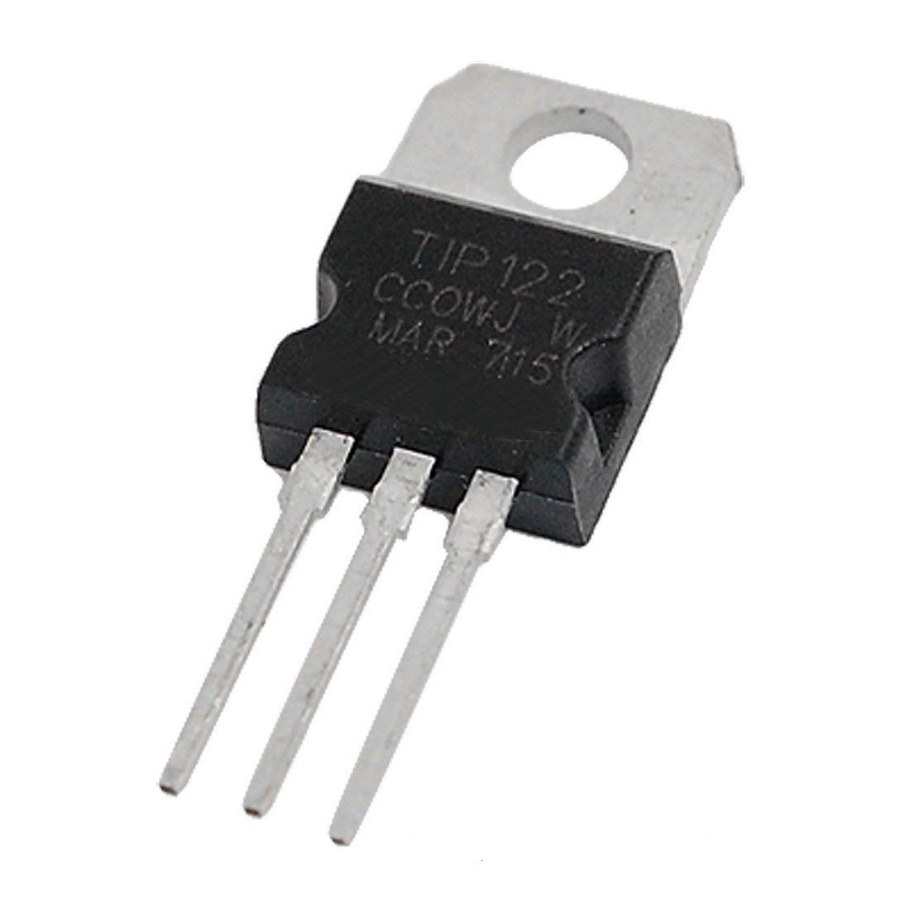


Figure : Transistor TIP122

A single digital pin on Arduino Nano do not provide enough current to power RGB LED, A solution for this situation is to use an NPN Darlington Transistor designed for medium power linear switching applications, so we use TIP122 Transistor to provide RGB LED with power from an external source. It can power devices up to 100VDC at 5 Amps.

**SPECS:**

* TIP122 is power transistors
* Collector Current: 5 ampere
* Collector-Emitter Volt: 100 volts
* Power Dissipation: 65 watts

**PINOUT:**

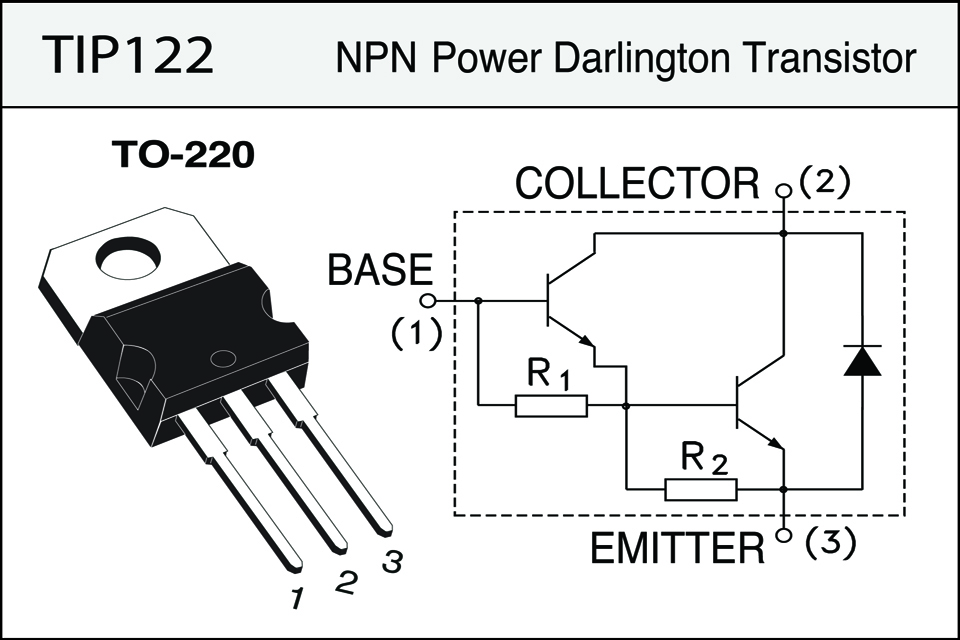


Figure : Transistor TIP122 pinout

##### Servo motor SG90

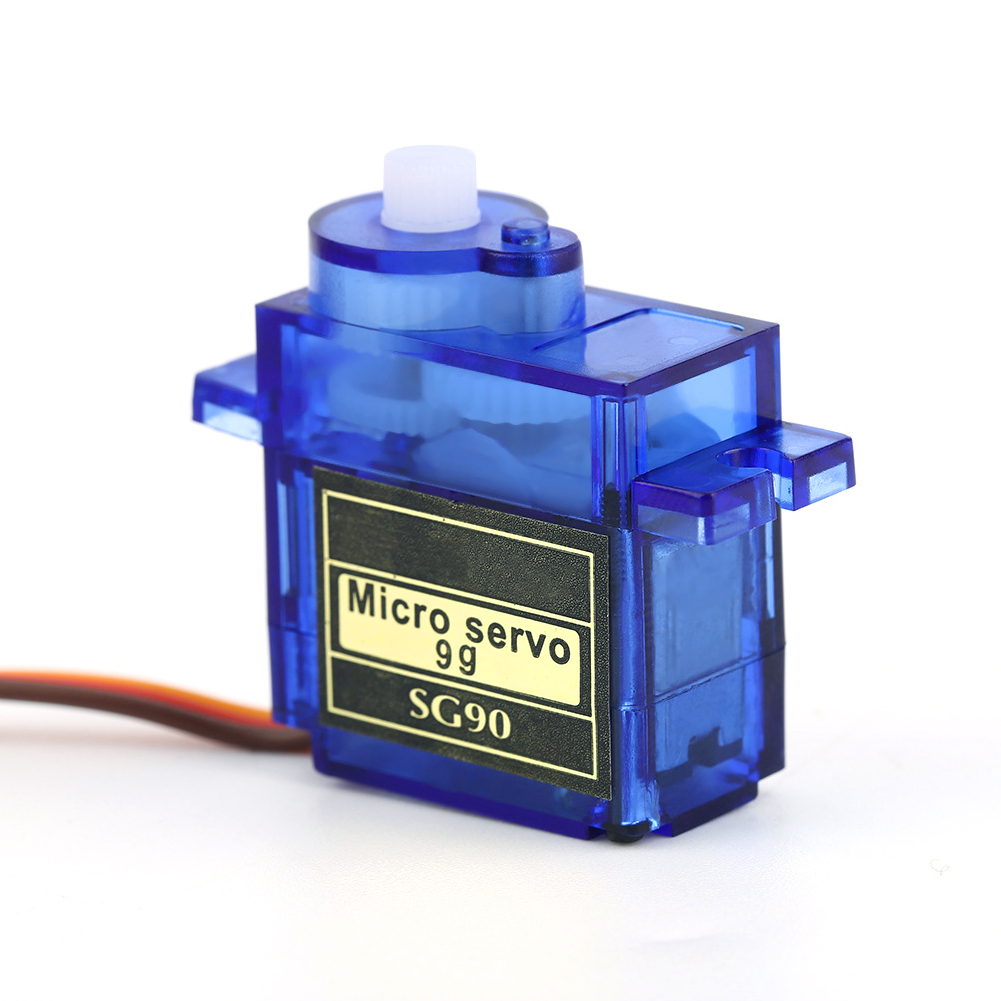


Figure : Servo motor SG90

PGSS use Servo Motor SG90 to control barrier at each parking slot.

**SPECS:**

|  |  |
| --- | --- |
| **Torque** | 1.80 kg-cm at 4.8V |
| **Speed** | 0.1sec/60° at 4.8V |
| **Voltage** | 4.0V to 7.2V, 4.6V - 5.2V nominal |
| **Dimensions** | 23mm x 12.2mm x 29mm |
| **Rotation range** | 180o |
| **Weight** | 9g |
| **Pulse width** | 500-2400uS |
| **Operating Temperature range** | 30°C to 60°C |

Table : Servo motor SG90 specifications

**PINOUT:**

|  |  |  |
| --- | --- | --- |
| **Pin of Servo SG90** | **Name** | **Description** |
| Red | VCC | Power supply 5V |
| Black | GND | Power supply ground |
| Yellow | Signal | The servo will move based on the signal sent to signal wire. |

Table : Servo motor SG90 pinout

#### Sign node

##### 7-Segment LED display

##### Power logic 8-bit shift register TPIC6B595

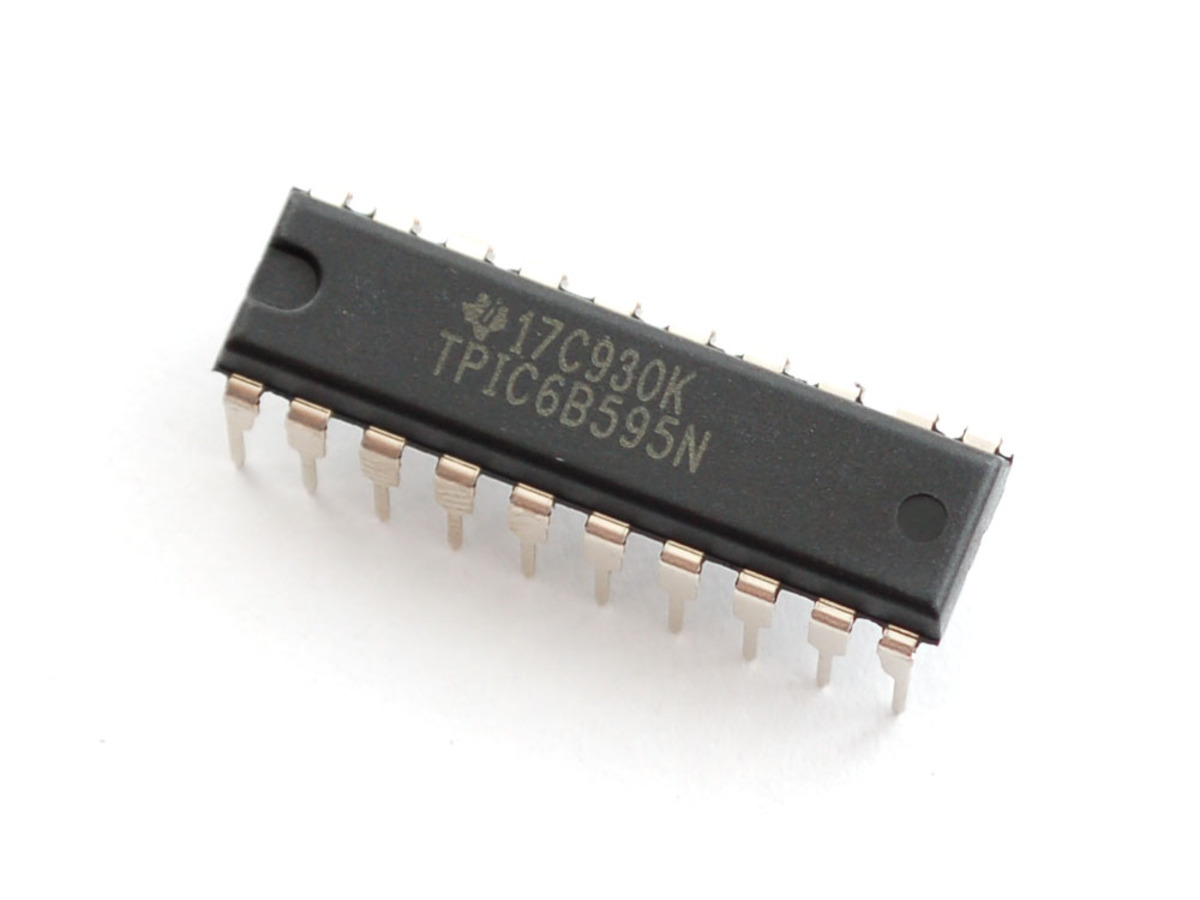


Figure : Power logic 8-bit shift register TPIC6B595

To display high power 7-segment display, we choose IC TPIC6B595 instead of IC 74HC595 because TPIC6B595 is asimple shift register IC that can control high-voltage/high-current devices directly**.** Each output pin can sink 150mA and then supports the maximum of Load Voltage at 50V.

**PINOUT:**

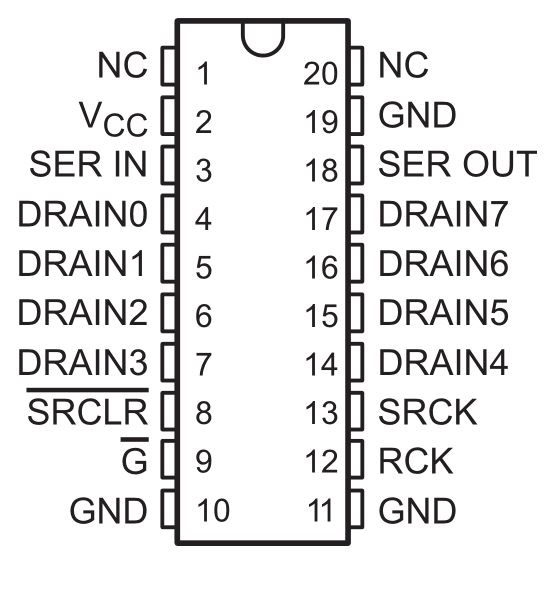


Figure : Power logic 8-bit shift register TPIC6B595 pinout

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin** | | **I/O** | **Description** |
| **Name** | **No.** | O | Open-drain output |
| DRAIN0 | 4 |
| DRAIN1 | 5 |
| DRAIN2 | 6 |
| DRAIN3 | 7 |
| DRAIN4 | 14 |
| DRAIN5 | 15 |
| DRAIN6 | 16 |
| DRAIN7 | 17 |
| G | 9 | I | Output enable, active-low |
| GND | 10,11,19 | - | Power ground |
| NC | 1, 20 | - | No internal connection |
| RCK | 12 | I | Register clock |
| SERIN | 3 | I | Serial data input |
| SEROUT | 18 | O | Serial data output |
| SRCK | 15 | I | Shift register clock |
| SRCLR | 8 | I | Shift register clear, active-low |
| VCC | 2 | I | Power supply |

Table : Power logic 8-bit shift register TPIC6B595 pinout

#### Components connection

## Interface

### Component Interface

### User Interface Design

#### Common Interface

##### Login screen

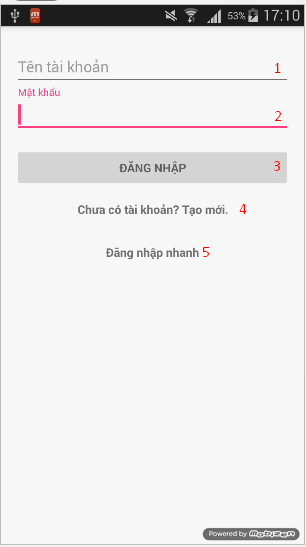


Figure : Login screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Username | Fill in username | N/A | Yes | Edittext | String | N/A |
| 2 | Password | Fill in password | N/A | Yes | Edittext | String | N/A |

Table : Login screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Sign in | Log-in into the system | N/A | Transfer to User Activity |
| 4 | Register | Register new account | N/A | Transfer to Register Activity |
| 5 | Sign In as Guest | Log-in into the system as Guest | N/A | Transfer to User Activity |

Table : Login screen button/hyperlinks

##### Register screen

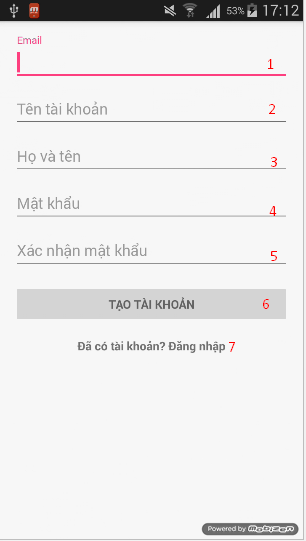


Figure : Register screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Email | Fill in email | N/A | Yes | Edittext | String | N/A |
| 2 | Username | Fill in username | N/A | Yes | Edittext | String | N/A |
| 3 | Fullname | Fill in fullname | N/A | Yes | Edittext | String | N/A |
| 4 | Password | Fill in password | N/A | Yes | Edittext | String | N/A |
| 5 | Confirm Password | Fill in confirm password | N/A | Yes | Edittext | String | N/A |

Table : Register screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 6 | Register | Register new account | N/A | Transfer to Login Activity |
| 7 | Sign up | Sign up to app | N/A | Transfer to Login Activity |

Table : Register screen button/hyperlinks

#### User/Guest Interface

##### Map view screen

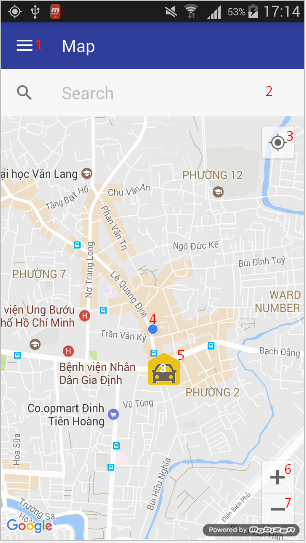


Figure : Map view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 2 | Location Search | Fill in location name to search | N/A | No | Editttext | String | N/A |
| 4 | Current Location | The dot show current location of user | Yes | Yes | Image | Drawable | N/A |
| 5 | Car park marker | Map marker of car park | Yes | Yes | Image | Drawable | N/A |

Table : Map view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Navigation drawer | Open navigation drawer | N/A | Open navigation drawer |
| 3 | Current Location | Button to center map to current location | N/A | Center map to current location |
| 6 | Zoom in | Map zoom in button | N/A | Map zoom in |
| 7 | Zoom out | Map zoom out button | N/A | Map zoom out |

Table : Map view screen button/hyperlinks

##### Navigation view

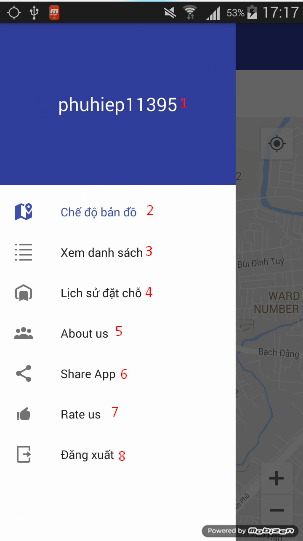


Figure : Navigation view

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Username | Fill in username | Yes | Yes | Textview | String | N/A |

Table : Navigation view fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Map view | Change to map view button | N/A | Transfer to User Activity |
| 3 | List view | Change to car park list view button | N/A | Transfer to Car Park List Activity |
| 4 | Transaction view | Change to history transaction list view button | N/A | Transfer to Transaction Activity |
| 5 | About us | About us button | N/A | Show about us dialog |
| 6 | Share app | Share app button | N/A | Share app on social network |
| 7 | Rate us | Rate us button | N/A | Transfer to app download page on app store |
| 8 | Logout | Logout button | N/A | Transfer to Login Activity |

Table : Navigation view button/hyperlinks

##### Car park list view screen

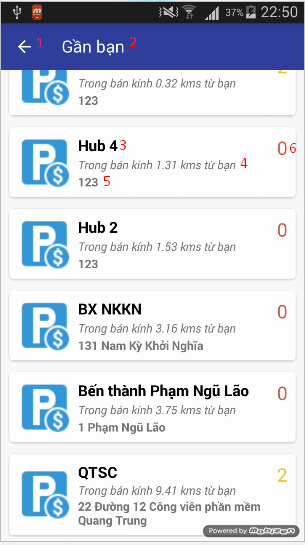


Figure : Car park list view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 2 | Title | Display searched location or user location | Yes | Yes | Textview | String | N/A |
| 3 | Car park name | Name of car park | Yes | Yes | Textview | String | N/A |
| 4 | Distance away | Distance away from car park | Yes | Yes | Textview | String | N/A |
| 5 | Address | Address of car park | Yes | Yes | Textview | String | N/A |
| 6 | Available lot | Current available lot of car park | Yes | Yes | Textview | String | N/A |

Table : Car park list view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Back | Back to previous activity | N/A | Transfer to previous activity |

Table : Car park list view screen button/hyperlinks

##### Car park detail view screen

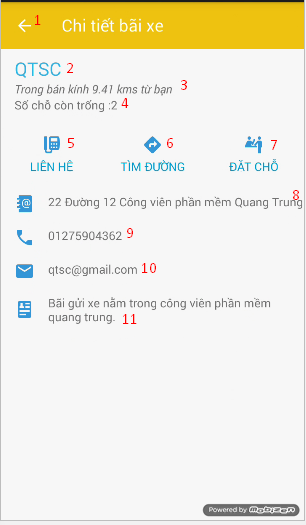


Figure : Car park detail view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 2 | Name | Name of car park | Yes | Yes | Textview | String | N/A |
| 3 | Distance away | Distance away from car park | Yes | Yes | Textview | String | N/A |
| 4 | Available lot | Current available lot of car park | Yes | Yes | Textview | String | N/A |
| 8 | Address | Address of car park | Yes | Yes | Textview | String | N/A |
| 9 | Phone | Phone of car park | Yes | Yes | Textview | String | N/A |
| 10 | Email | Email of car park | Yes | Yes | Textview | String | N/A |
| 11 | Description | Description of car park | Yes | Yes | Textview | String | N/A |

Table : Car park detail view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Back | Back to previous activity | N/A | Transfer to previous activity |
| 5 | Call | Call button | N/A | Call action using car park phone |
| 6 | Routing | Request routing from map app | N/A | Transfer to Map App with routing request |
| 7 | Reserve | Request reserve parking lot | N/A | Show reservation dialog |

Table : Car park detail view screen button/hyperlinks

##### History transaction list view screen

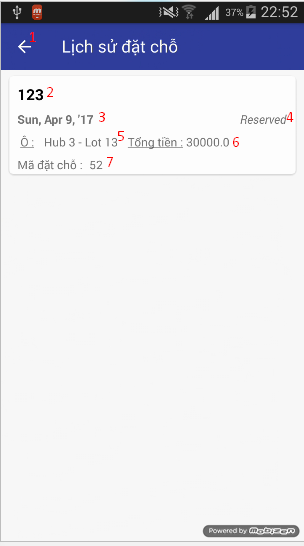


Figure : History transaction list view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 2 | Address | Address of car park | Yes | Yes | Textview | String | N/A |
| 3 | Date | Date of reservation | Yes | Yes | Textview | String | N/A |
| 4 | Status | Status of transaction | Yes | Yes | Textview | String | N/A |
| 5 | Lot name | Name of reserved parking lot | Yes | Yes | Textview | String | N/A |
| 6 | Cost | Total cost of reservation | Yes | Yes | Textview | String | N/A |
| 7 | PIN | PIN code of reservation | Yes | Yes | Textview | String | N/A |

Table : History transaction list view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Back | Back to previous activity | N/A | Transfer to previous activity |

Table : History transaction list view screen button/hyperlinks

##### History transaction detail view screen



Figure : History transaction detail view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Detail | Detail information of transaction | Yes | Yes | Textview | String | N/A |

Table : History transaction detail view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Cancel | Cancel transaction | N/A | Perform cancel transaction process |
| 3 | Check –in | Check-in car park | N/A | Perform check-in transaction process |

Table : History transaction detail view screen button/hyperlinks

##### Reservation process view screen

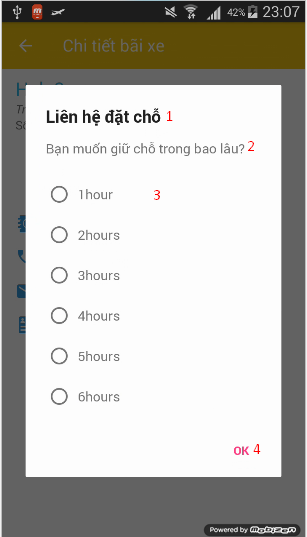


Figure : Reservation process view screen 1

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Title | Dialog title | Yes | Yes | Textview | String | N/A |
| 2 | Content | Content message of dialog | Yes | Yes | Textview | String | N/A |
| 3 | Choice list | Choice list of dialog | N/A | Yes | List | String | N/A |

Table : Reservation process view screen 1 fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 4 | OK | Finish step 1 of reservation | N/A | Show reservation process dialog 2 |

Table : Reservation process view screen 1 button/hyperlinks

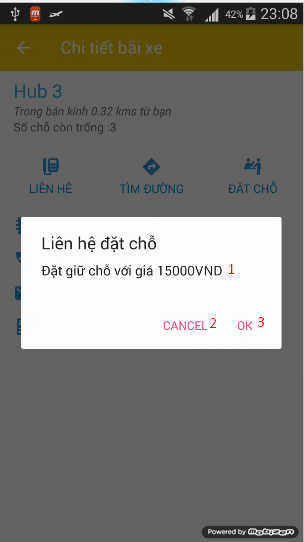


Figure : Reservation process view screen 2

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Cost | Message of the total cost of reservation | Yes | Yes | Textview | String | N/A |

Table : Reservation process view screen 2 fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Cancel | Cancel the reservation | N/A | Close the dialog |
| 3 | OK | Confirm the reservation | N/A | Send the request to servers. |

Table : Reservation process view screen 2 button/hyperlinks

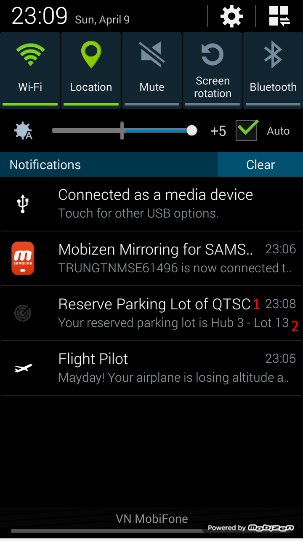


Figure : Reservation process notification

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Title | Title of notification | Yes | Yes | Texview | String | N/A |
| 2 | Content | Content of notification | Yes | Yes | Textview | String | N/A |

Table : Reservation process notification fields

**Button/Hyperlinks**

N/A

#### Manager Interface

##### Car park list view screen

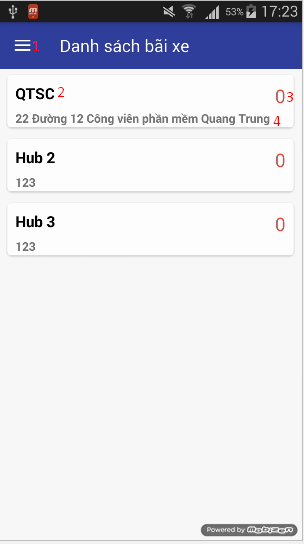


Figure : Car park list view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 2 | Name | Name of car park | Yes | Yes | Textview | String | N/A |
| 3 | Available lot | Current available lot of car park | Yes | Yes | Textview | String | N/A |
| 4 | Address | Address of car park | Yes | Yes | Textview | String | N/A |

Table : Car park list view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Drawer | Show navigation view | N/A | Show navigation |

Table : Login screen button/hyperlinks

##### Car park detail view screen

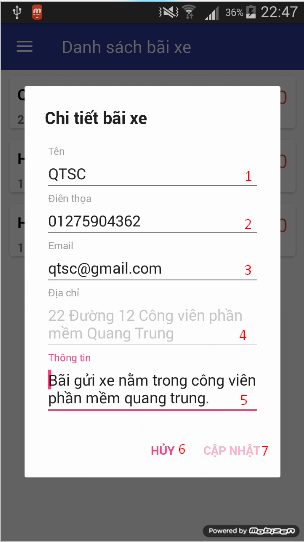


Figure : Car park detail view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Name | Name of car park | N/A | Yes | Edittext | String | N/A |
| 2 | Phone | Phone of car park | N/A | Yes | Edittext | String | N/A |
| 3 | Email | Email of car park | N/A | Yes | Edittext | String | N/A |
| 4 | Address | Address of car park | Yes | Yes | Edittext | String | N/A |
| 5 | Description | Description of car park | N/A | Yes | Edittext | String | N/A |

Table : Car park detail view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 6 | Cancel | Cancel edit car park | N/A | Close dialog |
| 7 | Update | Update car park | N/A | Send request update car park information |

Table : Car park detail view screen button/hyperlinks

##### Area list view screen

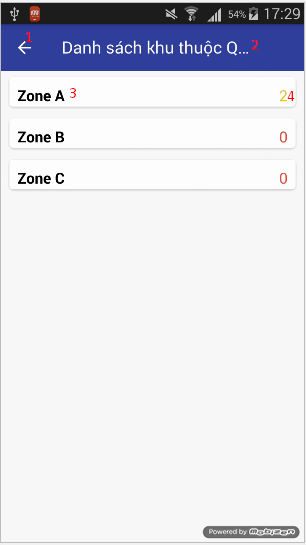


Figure : Area list view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 2 | Title | Car park name | Yes | Yes | Textview | String | N/A |
| 3 | Name | Name of area | Yes | Yes | Textview | String | N/A |
| 4 | Available lot | Current available lot of area | Yes | Yes | Textview | String | N/A |

Table : Area list view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Back | Back to previous activity | N/A | Transfer back to previous activity |

Table : Area list view screen button/hyperlinks

##### Area detail view screen

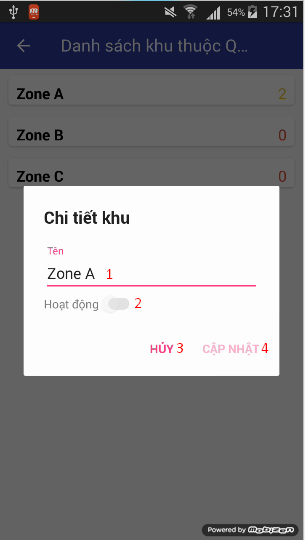


Figure : Area detail view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Name | Name of area | N/A | Yes | Edittext | String | N/A |
| 2 | Active | Active switch of area | N/A | Yes | Switch | Bool | N/A |

Table : Area detail view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Cancel | Cancel of edit area | N/A | Close dialog |
| 4 | Update | Update area information | N/A | Send request to update area |

Table : Area detail view screen button/hyperlinks

##### Parking lot list view screen

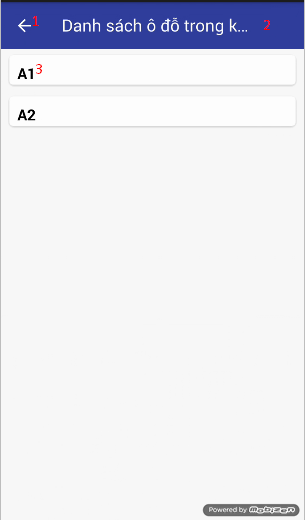


Figure : Parking lot list view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 2 | Title | Area name | Yes | Yes | Textview | String | N/A |
| 3 | Name | Name of parking lot | Yes | Yes | Textview | String | N/A |

Table : Parking lot list view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Back | Back to previous activity | N/A | Transfer back to previous activity |

Table : Parking lot list view screen button/hyperlinks

##### Parking lot detail view screen

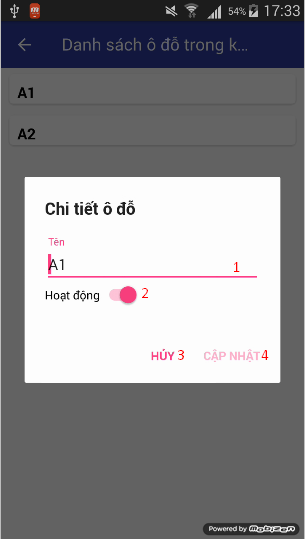


Figure : Parking lot detail view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Name | Name of parking lot | N/A | Yes | Edittext | String | N/A |
| 2 | Active | Active switch of parking lot | N/A | Yes | Switch | Bool | N/A |

Table : Parking lot detail view screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Cancel | Cancel of edit parking lot | N/A | Close dialog |
| 4 | Update | Update parking lot information | N/A | Send request to update parking lot |

Table : Parking lot detail view screen button/hyperlinks

##### Check code view screen

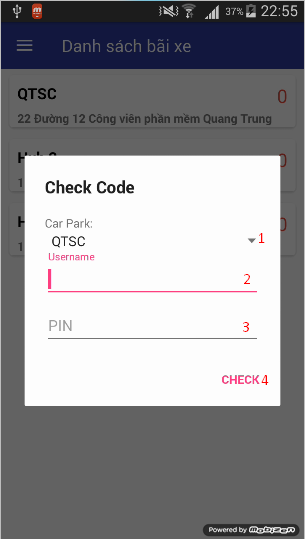


Figure : Check code view screen

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field**  **Name** | **Description** | **Read**  **Only** | **Mandatory** | **Control**  **Type** | **Data**  **Type** | **Length** |
| 1 | Car park list | List of car park | N/A | Yes | Spinner | String | N/A |
| 2 | Username | Fill in username | N/A | Yes | Edittext | String | N/A |
| 3 | PIN | Fill in PIN code | N/A | Yes | Edittext | String | N/A |

Table : Check code screen fields

**Button/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 4 | Check | Check PIN code with user name and car park | N/A | Send request to check and perform check-in if correct username and pin code for car park |

Table : Check code screen button/hyperlinks

## Database Design

### Logical Diagram

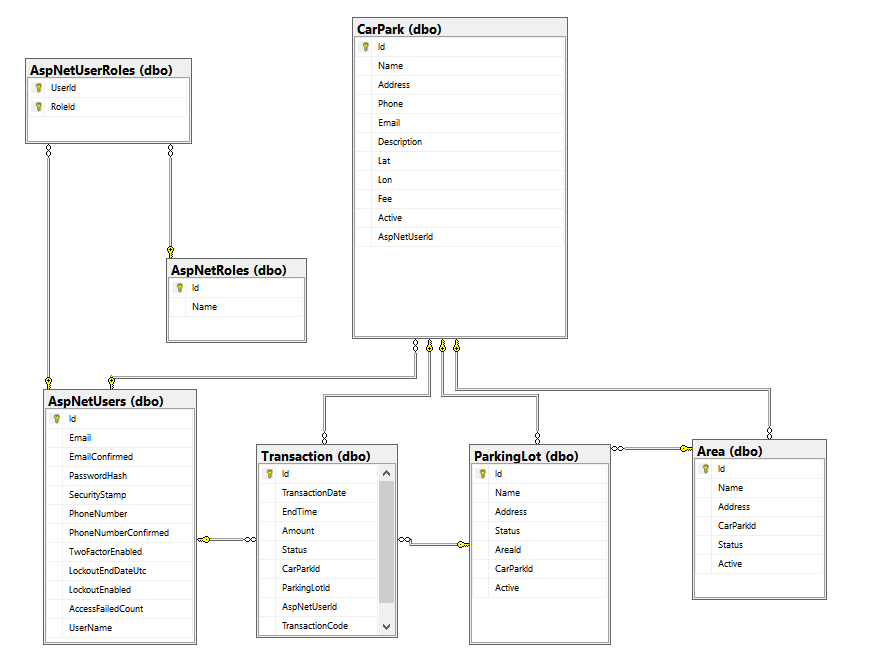


Figure : Database logical diagram

### Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Column Name** | **Is Nullable** | **Data Type** |
| **Area** | Id | NO | int |
| Name | NO | nvarchar |
| Address | YES | int |
| CarParkId | NO | int |
| Status | NO | int |
| Active | NO | bit |
| **ParkingLot** | Id | NO | int |
| Name | NO | nvarchar |
| Address | NO | int |
| Status | NO | int |
| AreaId | YES | int |
| CarParkId | NO | int |
| Active | NO | bit |
| **Transaction** | Id | NO | int |
| TransactionDate | NO | datetime |
| EndTime | YES | datetime |
| Amount | NO | money |
| Status | NO | int |
| CarParkId | NO | int |
| ParkingLotId | NO | int |
| AspNetUserId | NO | nvarchar |
| TransactionCode | YES | nvarchar |
| **CarPark** | Id | NO | int |
| Name | NO | nvarchar |
| Address | NO | nvarchar |
| Phone | YES | nvarchar |
| Email | YES | nvarchar |
| Description | YES | nvarchar |
| Lat | NO | nvarchar |
| Lon | NO | nvarchar |
| Fee | YES | money |
| Active | NO | bit |
| AspNetUserId | YES | nvarchar |
| **AspNetRoles** | Id | NO | nvarchar |
| Name | NO | nvarchar |
| **AspNetUserRoles** | UserId | NO | nvarchar |
| RoleId | NO | nvarchar |
| **AspNetUsers** | Id | NO | nvarchar |
| Email | YES | nvarchar |
| EmailConfirmed | NO | bit |
| PasswordHash | YES | nvarchar |
| SecurityStamp | YES | nvarchar |
| PhoneNumber | YES | nvarchar |
| PhoneNumberConfirmed | NO | bit |
| TwoFactorEnabled | NO | bit |
| LockoutEndDateUtc | YES | datetime |
| LockoutEnabled | NO | bit |
| AccessFailedCount | NO | int |
| UserName | NO | nvarchar |

Table : Database data dictionary

## Algorithms

### GetDistance formula

To calculate the long distance with more accuracy, we must calculate the distance in the circle by using Haversine formula

is the haversine function

*d* is the distance between the two points

*r* is the radius of the sphere,

*φ*1, *φ*2: latitude of point 1 and latitude of point 2, in radians

*λ*1, *λ*2: longitude of point 1 and longitude of point 2, in radians

By apply the inverse haversine or by using the arcsine, we can calculate d:

### CRC Error Detection

#### Definition

The aim of an error detection technique is to enable the receiver of a message transmitted through a noisy (error-introducing) channel to determine whether the message has been corrupted. To do this, the transmitter constructs a value (called a checksum) that is a function of the message, and appends it to the message. The receiver can then use the same function to calculate the checksum of the received message and compare it with the appended checksum to see if the message was correctly received.

#### Define Problem

Because we do not require a license to operate radio equipment in the 2.4GHz band, as a result, other transmitter can broadcast on the same frequency that our devices use.

#### Solution theory

Two important aspects required to form a strong checksum function:

* **WIDTH:** a register width wide enough to provide a low a-priori probability of failure (e.g. 32-bits gives a 1/2^32 chance of failure).
* **CHAOS:** a formula that gives each input byte the potential to change any number of bits in the register.

For that reason, the CRC schemes use division with some changes so that if more bytes were added to the message, the checksum value could change radically again very quickly. All the arithmetic performed during CRC calculation is performed in binary with no carries, we will call it CRC arithmetic.

Adding two numbers in CRC arithmetic is the same as adding numbers in ordinary binary arithmetic except there is no carry. This means that each pair of corresponding bits determine the corresponding output bit without reference to any other bit positions. For example:



Figure : CRC arithmetic addition

Subtraction is identical:



Figure : CRC arithmetic subtraction

In fact, both addition and subtraction in CRC arithmetic is equivalent to the XOR operation, and the XOR operation is its own inverse. This effectively reduces the operations of the first level of power (addition, subtraction) to a single operation that is its own inverse. This is a very convenient property of the arithmetic.

Having defined addition, we can move to multiplication and division. Multiplication is absolutely straightforward, being the sum of the first number, shifted in accordance with the second number. (note: the sum uses CRC addition)

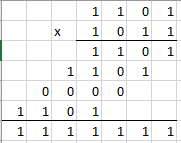


Figure : CRC arithmetic multiplication

Division is a little messier as we need to know when "a number goes into another number". To do this, we invoke the weak definition of magnitude defined earlier: that X is greater than or equal to Y if the position of the highest 1 bit of X is the same or greater than the position of the highest 1 bit of Y.

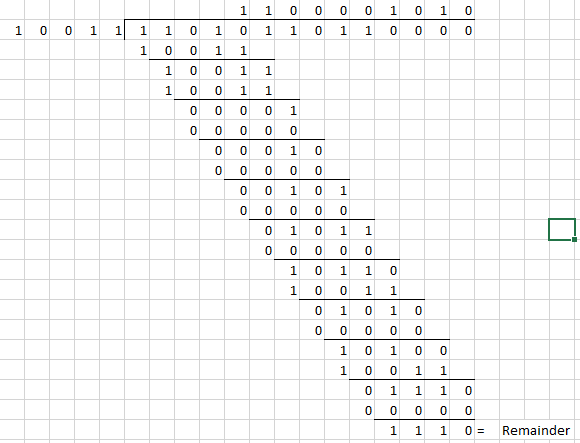


Figure : CRC arithmetic division

Having defined CRC arithmetic, we can now frame a CRC calculation as simply a division, because that's all it is. To perform a CRC calculation, we need to choose a divisor. In math marketing speak the divisor is called the "generator polynomial" or simply the "polynomial", and is a key parameter of any CRC algorithm. As a compromise, we will refer to the CRC polynomial as the "poly".

The width (position of the highest 1 bit) of the poly is very important as it dominates the whole calculation. Typically, widths of 16 or 32 are chosen so as to simplify implementation on modern computers. The width of a poly is the actual bit position of the highest bit. For example, the width of 10011 is 4, not 5. For the purposes of example, we will choose a poly of 10011 (of width W of 4).

Having chosen a poly, we can proceed with the calculation. This is simply a division (in CRC arithmetic) of the message by the poly. The only trick is that W zero bits are appended to the message before the CRC is calculated. Thus we have:

* Original message: 1101011011
* Poly: 10011
* Message after appending W zeros: 11010110110000

This is the same division as above figure, so we got the remainder, which is the calculated checksum is 1110. Usually, the checksum is then appended to the message and the result transmitted. In this case the transmission would be: 11010110111110. This ends the calculation.

A summary of the operation of the class of CRC algorithms:

* Choose a width W, and a poly G (of width W).
* Append W zero bits to the message. Call this M'.
* Divide M' by G using CRC arithmetic. The remainder is the checksum.

#### Implementation

To implement a CRC algorithm all we have to do is implement CRC division. There are two reasons why we cannot simply use the divide instruction of whatever machine we are on. The first is that we have to do the divide in CRC arithmetic. The second is that the dividend might be ten megabytes long, and todays processors do not have registers that big.

So to implement CRC division, we have to feed the message through a division register. At this point, we have to be absolutely precise about the message data. In all the following examples the message will be considered to be a stream of bytes (each of 8 bits) with bit 7 of each byte being considered to be the most significant bit (MSB). The bit stream formed from these bytes will be the bit stream with the MSB (bit 7) of the first byte first, going down to bit 0 of the first byte, and then the MSB of the second byte and so on.

For the purposes of example, consider a poly with W=4 and the poly=10111. Then, the perform the division, we need to use a 4-bit register:

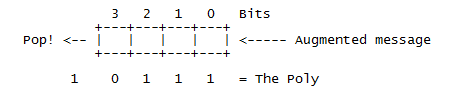


Figure : CRC implementation register

(Augmented message is the message followed by W zero bits)

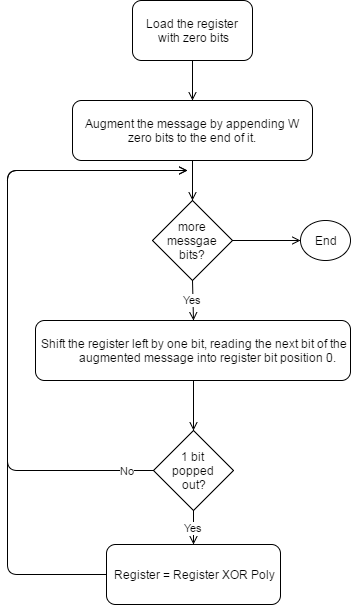


Figure : CRC implementation simple flow

#### Table-driven implementation

The implementation above is a good starting point because it corresponds directly to the theory presented so far, and because it is so simple. However, because it operates at the bit level, it is rather awkward to code (even in C), and inefficient to execute (it has to loop once for each bit). To speed it up, we need to find a way to enable the algorithm to process the message in units larger than one bit.

For the purposes of discussion, let us switch from a 4-bit poly to a 32-bit one. Our register looks much the same, except the boxes represent bytes instead of bits, and the Poly is 33 bits (one implicit 1 bit at the top and 32 "active" bits) (W=32).

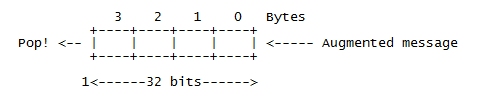
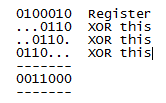


Figure : CRC 32-bits register

Consider for a moment that we use the top 8 bits of the register to calculate the value of the top bit of the register during the next 8 iterations. Suppose that we drive the next 8 iterations using the calculated values (which we could perhaps store in a single byte register and shift out to pick off each bit). Then we note three things:

* The top byte of the register now doesn't matter. No matter how many times and at what offset the poly is XORed to the top 8 bits, they will all be shifted out the right hand side during the next 8 iterations anyway.
* The remaining bits will be shifted left one position and the rightmost byte of the register will be shifted in the next byte.
* While all this is going on, the register will be subjected to a series of XOR's in accordance with the bits of the pre-calculated control byte.

Now consider the effect of XORing in a constant value at various offsets to a register. For example:



The point of this is that you can XOR constant values into a register to your heart's delight, and in the end, there will exist a value which when XORed in with the original register will have the same effect as all the other XORs.

Putting all the pieces together we have an algorithm that goes like this:

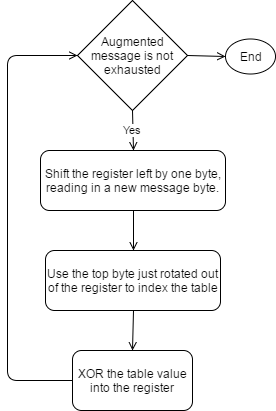


Figure : CRC implementation table-driven flow

For the actual implementation in project, we use 3 bytes in payload package as checksum, so we need to implement CRC24. Here is the table we use for CRC24 table-driven implementation:

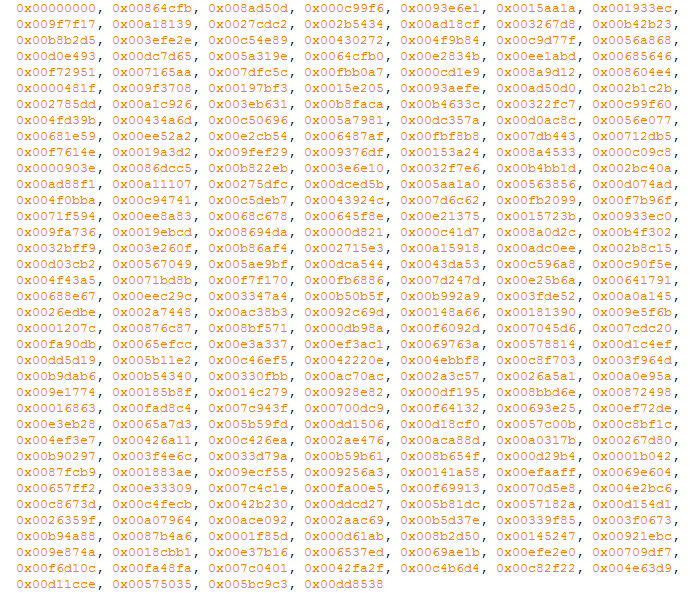


Figure : CRC24 precomputed table