**Parking Genie LOGO**

**Technical Design Document**

**Application/ Project: Parking Genie**

**Document Version : 0.4**

Table of Contents

[1 REVISION HISTORY 5](#_Toc423253338)

[2 REQUIRED APPROVERS 5](#_Toc423253339)

[3 TECHNICAL DESIGN OVERVIEW 5](#_Toc423253340)

[4 TECHNICAL ASSUMPTIONS, DEPENDENCIES, CONSTRAINTS AND RISKS 6](#_Toc423253341)

[5 SYSTEM FRAMEWORK 6](#_Toc423253342)

[5.1 CONTEXT DIAGRAM 6](#_Toc423253343)

[5.2 Business process flow diagram 6](#_Toc423253345)

[5.3 COMPONENT Design 6](#_Toc423253346)

[5.3.1 DESIGN (for each component) 6](#_Toc423253347)

[5.4 RESOURCES 10](#_Toc423253348)

[6 OBJECT ORIENTED DESIGN APPROACH (Distributed systems: Java &.NET platforms) 11](#_Toc423253349)

[6.1 PACKAGE & SUBSYSTEM DESIGN 11](#_Toc423253350)

[6.2 CONFIGURATION FILES 11](#_Toc423253351)

[6.3 Design Patterns 11](#_Toc423253352)

[7 MONITORING 11](#_Toc423253353)

[7.1 LOGGING MECHANISM 11](#_Toc423253354)

[7.2 Packaging and DeLivery 11](#_Toc423253355)

[8 APPENDIX 12](#_Toc423253356)

[8.1 ACRONYMS AND DEFINITIONS 12](#_Toc423253357)

[8.2 REFERENCES 12](#_Toc423253358)

**LEGEND:**

|  |
| --- |
| Required |
| Optional |
| Applicable |
| Not Applicable |
| Both are applicable |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Section No** | **Section Description** | **Object Oriented Design** | | **Modular Design** | | **Conversion** |
|  |  | **New /Development** | **Enhancement /Defect fix** | **New /Development** | **Enhancement /Defect fix** |  |
| 1 | Revision History |  |  |  |  |  |
| 2 | Required Approvers |  |  |  |  |  |
| 3 | Technical Design Overview |  |  |  |  |  |
| 4 | Technical Assumptions, Dependencies, Constraints and Risks |  |  |  |  |  |
| 5 | System Framework |  |  |  |  |  |
| 6 | Object Oriented Design Approach |  | Update the object changes in Section 6.8 |  |  | Choose any one design approach |

# 

# REVISION HISTORY

| Revision History | | | | |
| --- | --- | --- | --- | --- |
| Version  No. | Details of Change | Changed Sections | Prepared by | Date |
| *0.1* | *Initial Version* |  | *Rahul* | *1/6/18 3:31 PM* |
|  |  |  |  |  |
|  |  |  |  |  |

# 

# REQUIRED APPROVERS

|  |  |  |
| --- | --- | --- |
| **Name** | **Signature** | **Date** |
| Rahul Sood |  |  |
| Raudel |  |  |
| Chandan |  |  |

# TECHNICAL DESIGN OVERVIEW

Parking Genie is an IOS, Android and Desktop based Webapp that promises to take the pain away from parking and make parking rewarding. The app serves both businesses and end consumers. Thus it will have two interfaces. Business person will log in to establish a business relation with Parking Genie while end consumer utilize the app to reserve parking spots.

Our base package name will be ***com.rrc.parkingGenie***

Parking Genie has been designed with following philosophy

* **Flexibility**
* **Open to extension** (The ‘O’ from the SOLID)
* **Ease of refactoring**
* **User convenience**

# TECHNICAL ASSUMPTIONS, DEPENDENCIES, CONSTRAINTS AND RISKS

|  |  |
| --- | --- |
| Assumptions | *Design will be interface driven with stress on composition over inheritance. Will try to follow the test-driven design paradigm.* |
| Dependencies | *Maven, JUnit 4, log4j2, Spring Boot Framework, Spring WebSockets, MongoDB, JavaFX (front end)* |
| Constraints | *Developed at off hours* |
| Risks | *Time,New technology* |

# SYSTEM FRAMEWORK

## CONTEXT DIAGRAM

## 

QUARTZ PROCESS

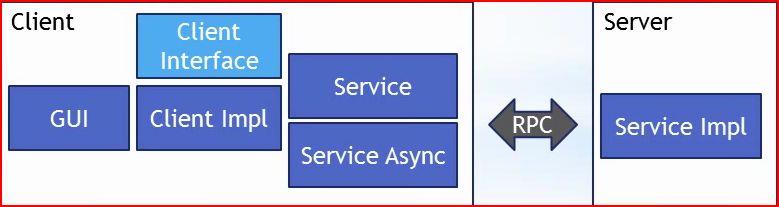
To Email

Supplies data to independent Quartz process

## 

## Business process flow diagram

Object db

 Q

Quartz Jobs

(run independently)

## COMPONENT Design

### DESIGN (for each component)

# OBJECT ORIENTED DESIGN APPROACH (Distributed systems: Java &.NET platforms)

## PACKAGE & SUBSYSTEM DESIGN

* UI – See above
* Model – See above
* Services – See above
* Jobs – See above
* Constants – constants for Tipping Point App
* Util – all Utility classes will be stored here, these provide implementation of common utility functions

## CONFIGURATION FILES

* Applicationcontext.xml
* Log4j.xml

## Design Patterns

* Composition
* Interface
* Strategy
* Template

# MONITORING

## LOGGING MECHANISM

* Log4j.
* All logs will be at error level
* debug logging will be added at appropriate sections of code and for all exception handlings.
* User will have the ability to change logging level to provide more information
* Log4e will be use to create logging statements.
* Log file *tippingPoint.log* will be stored at *<user.home>/tippingPoint\_<version>/*

## Packaging and DeLivery

Windows

Apple

# APPENDIX

## ACRONYMS AND DEFINITIONS

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| UI | User Interface |
| <install.dir> | Location where tipping point is installed. This may differ based on operating system. In windows this will usually be the program files folder. |
| <user.home> | Users home folder. On windows this is usually at c:\users\<username> |
|  |  |
|  |  |

## REFERENCES

|  |  |  |
| --- | --- | --- |
| **Reference Document / Supporting Document** | **Document Name** | **Document Location** |
|  |  |  |
|  |  |  |