|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Met/Not Met** | **How the design was met?** |
| Availability | The system should always be available and real time. |  | The android application and website can be available at all times given that the devices accessing the application or website is connected to the internet. Also, for the system to work, NodeMCU must be connected to the internet.  The system is real time by using Firebase Realtime Database that allows any change in the sensors to be immediately updated in the database, which in turn automatically displays the changes in the android application and website without the need to refresh the page |
| Performance | The system should complete the required functionalities within a short response time, providing accurate readings from sensors and efficient information from Android application and website. |  | Refer to section |
| Reliability | No data loss is allowed. |  | NodeMCU, Arduino, Android and the website do not save any data locally and automatically save data in the Firebase, hence if any malfunction occurs, the data will not be lost. |
| Connectivity | NodeMCU needs to be connected to the internet to collect updated data from the sensors through the Arduino board then save it in the Firebase. In addition, the mobile application and website needs to be connected to the internet to get the data from the Firebase |  | Refer to section |
| Scalability | The system can support the addition of sensors and components as well as having more users and the cloud can be upgraded accordingly. |  | ParQU can handle more VIP users with more RFID tags. Additionally, zones can be added with more components (In our prototype, a switch button is used to illustrate the scalability of our system, as explained in section (-)). All of the previous additions require more database storage, the Firebase Realtime Database allows us to store up to 1GB for free (Free Plan). Any more storage would require us to upgrade to higher plans with a specific pricing. (reference) |
| ~~Portability~~  Mobility | The system can be accessed from many different platforms. |  | Website can be accessible from any type of devices and platforms such as iOS, Android and Windows, etc. Android applications can be accessible from Android devices only whether it was a mobile or tablet. |
| Power | Power source needed for:  Motor: 3-7V  Sensor: 5V  Arduino: 7-12V |  | Arduino is supplied by a 9V battery. Sensors and Servo Motor are both supplied by Arduino |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quality | Name | Description | Met/Not Met | Verification |
| Economic | Design Cost | The prototype uses high quality component with affordable prices |  | The whole prototype costed us on average 550 QR (Prototype consists of two zones, each zone has four parking spots) |
| Social | Usability | A normal user with minimal software knowledge should be able to use the mobile application and the website with ease |  | Refer to section 6.3 Acceptance testing |
| Sustainability | Maintenance | The system components should be easy to replace, remove and implement. |  | The components could easily be replaced by uploading our code to a new component. |
| Quality | Performance | The system should provide efficient information and accurate readings from the parking area. |  |  |