1. **Requirements analysis**

In this section, the software development process used in this project is defined. The functional and nonfunctional requirements of the system are addressed.

* 1. **Software development process**

The software development process in our project is Scrum. Scrum is an iterative and incremental agile software development process that focuses on adapting to rapid changes. Scrum was chosen for our project because it does not require us to know exactly how and what to do from the beginning of the development process. The Scrum process is responsive to changes and improvements with repeated reviews and work inspection [17]. Figure 3-1 shows the Scrum process overview

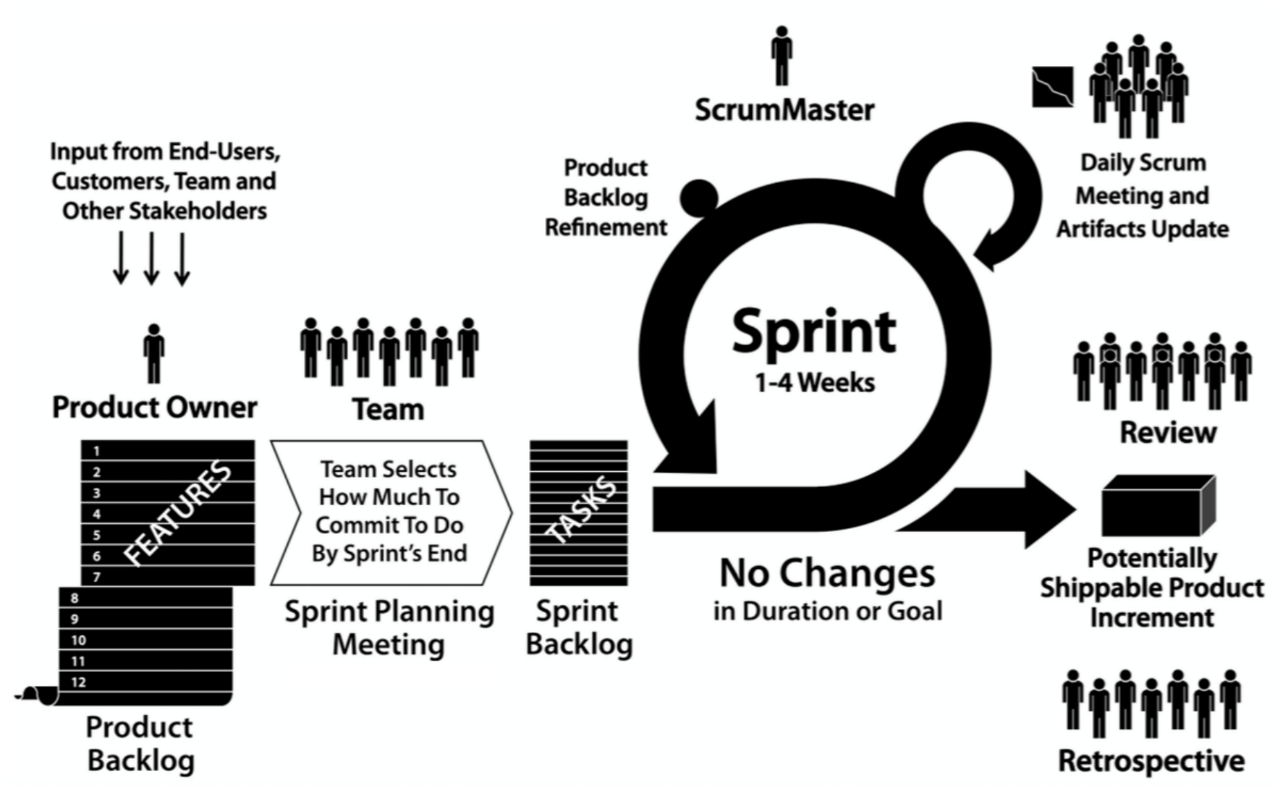
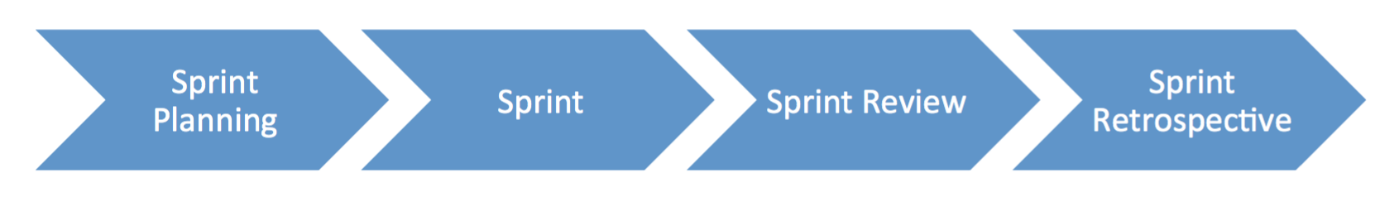


Figure ‎3‑1: Scrum process overview

The main part of the Scrum cycle is the Sprint. Each Sprint is time restricted to one week and has 4 steps as shown in Figure 3-2.



**Figure 3‑2: Scrum cycle**

**1. Sprint Planning**

At the beginning of the project, we collected the requirements into a Product Backlog from the users by researching the parking problem in Qatar University and the users’ needs. Before starting a Sprint, we hold up a meeting to decide on the top priority requirements from the Product Backlog and create a Sprint Backlog. The Sprint Backlog contains tasks that we think will be done during the Sprint which is in our case one week.

**2. Sprint**

In this stage, we start working on the tasks as specified in the Sprint Backlog. In addition, a Daily Scrum Meeting is held to discuss the tasks that was done by each member so far, and what will be done before the next meeting. Also, if any member has faced any difficulties during the task, we try to find a solution for it.

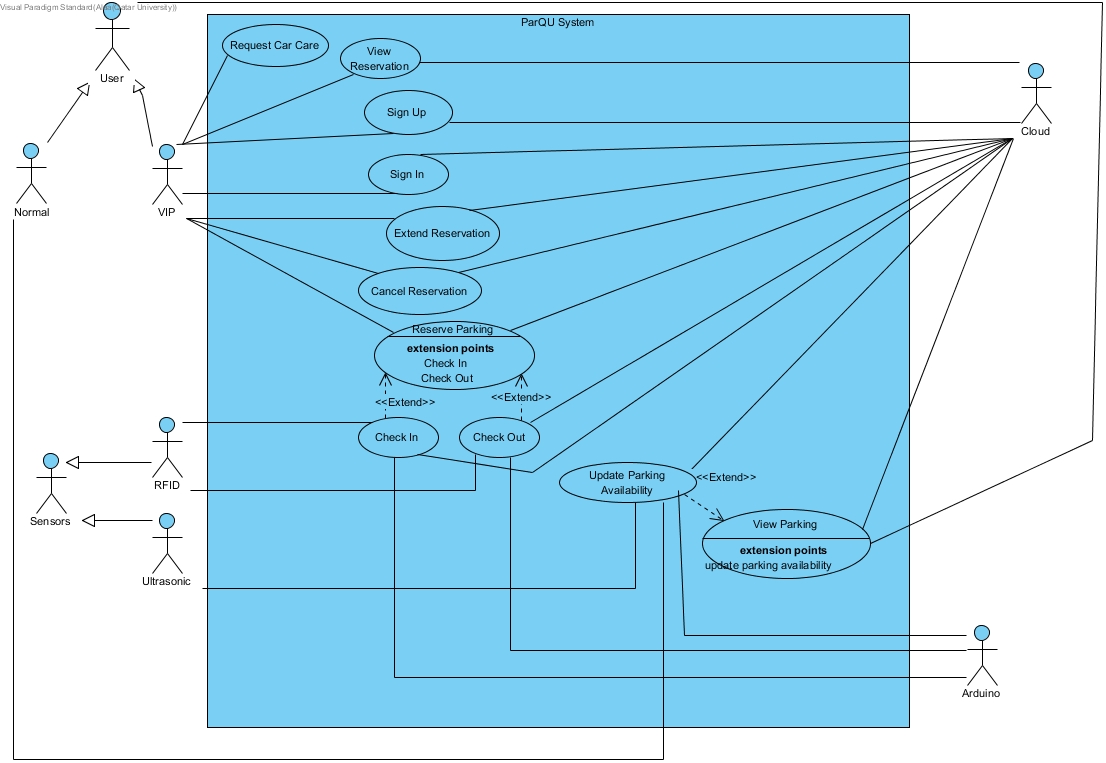
**3. Sprint Review**

At the end of a Sprint, a meeting is held with the ScrumMaster (supervisor) to demonstrate what was done. The ScrumMaster suggest solutions to the issues we faced during the Sprint.

**4. Sprint Retrospective**

After the Sprint Review, we refine our tasks to meet the comments and suggestions given by the ScrumMaster and decide on the improvements to be done to make the next Sprint smoother.

* 1. **Functional requirements**
* An RFID reader should be installed and kept operational at the gate of the parking facility.
* VIP users must have an RFID tag in their possession before entering the parking facility.
* Sensors should be hardwired to Arduino.
* Arduino must be connected to a power source.
* Arduino must be connected to a Wi-Fi module.
* Firebase must have a connection with the Arduinos, the mobile application and the website via Wi-Fi.
* The mobile must have internet connection.
* The user should be a university member or visitor.



**Figure** 3.**‑3: Use case diagram**

**Table 3‑1: Actor type and descriptions**

|  |  |
| --- | --- |
| Actor types | Description |
| VIP User | Anyone who subscribed to the VIP services such as reservation service and request car care service. |
| Normal User | Anyone who is not subscribed to the VIP services. |

**Table 3‑2: Use cases and descriptions**

|  |  |
| --- | --- |
| Use case | Brief description |
| Sign Up | VIP user registers in the system. |
| Sign In | VIP user logs into the system. |
| Reserve Parking | VIP user reserves a parking spot for several hours. |
| View Reservation | VIP user views reservation details. |
| Extend Reservation | The VIP user extends a reservation. |
| Cancel Reservation | The VIP user cancels a reservation. |
| Check In | The system reads the RFID tag on the car, checks its validity, then allows reserved VIP users only to enter the parking lot. |
| Check Out | The system reads the RFID tag on the car, checks its validity, then update available parking spots. |
| View Parking | The user views a map with the current status of parking spots. |
| Update Parking Availability | The system receives updated data from sensors through the Arduino board. |
| Request Car Care | The VIP user request car care service which will display contact numbers of car care company. |

* 1. **Non-functional requirements**

**Design constraints**

**Table 3‑3: Technical design constraints**

|  |  |
| --- | --- |
| Name | Description |
| Reliability | Data is saved at all times on the Firebase database, so no data will be lost. |
| Scalability | The system can support the addition of sensors and components as well as having more users and the cloud can be upgraded accordingly. |
| Connectivity | Firebase database needs to be connected to the internet to collect updated data from the sensors through the Arduino board. Also, the mobile application needs to be connected to the internet to get the data |
| Availability | The system should be always available. During using the application, if any data changed in firebase database, the application should automatically display the updates without needing to refresh the page. |
| Mobility | The system can be accessed from two different platforms: an Android application and a website |
| Power | Power source needed for:  Motor: 3-7V  Sensor: 5V  Arduino: 7-12V |

**Table 3‑4: Practical design constraints**

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| Economics | Cost | The prototype for availability module which consists of 4 parking spots should cost on average 270 QR.  The prototype for reservation module should cost on average 260 QR. |
| Social | Usability | A normal user with minimal software knowledge should be able to use the mobile application and the website with ease |
| Sustainability | Maintenance | The system’s component should be easy to replace, remove and implement. |
| Quality | Performance | The system should provide efficient information and accurate readings from the parking area. |

**Design standards**

**Table 3‑5: Standard types and descriptions**

|  |  |
| --- | --- |
| Standard Types | Description |
| Networking Standard | HTTP (Hypertext transfer protocol) |
| Security Standard | OpenID Connect (used for delegated authentication) |
| Wireless Standard | IEEE 802.11 (Wi-Fi) |
| Serial Communication Standards | SPI (Serial Peripheral Interface) |

* 1. **Assumptions**

Below are some assumptions that are out of the scope of the work done in this project but necessary for the project to work properly:

* Arduino is always connected to the sensors.
* A power source is connected to the Arduino.
* Arduino always has a Wi-Fi connection.
* Firebase can always store data without limits.
* Application and website are always available.
* Mobile and PC that access our application and website is always connected to the internet.
* VIP user always has an RFID tag.
  1. **Ethics**

**Table 3‑6: IEEE and ACM code of ethics**

|  |  |
| --- | --- |
| Code | Project Perspective |
| IEEE  To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others.  ACM  - Honor property rights including copyrights and patent.  - Give proper credit for intellectual property. | Any project or work used is referenced in this paper. Feedback will be accepted by the supervisor and the examiners. |
| IEEE  To treat fairly all persons and to engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identify, or gender expression.  ACM  - Be fair and take action not to discriminate. | Project members have all worked together to produce this work which can be used by all types of people. |
| IEEE  To assist colleagues and co-workers in their professional development and to support them in following this code of ethics. | The project is collaborative; every team member will work on it and ensure the code of ethics is strictly followed. |
| ACM  - Contribute to society and human well-being.  - Manage personnel and resources to design and build information systems that enhance the quality of working life. | The design aims to achieve better standards and benefits. |