5.1 Section-1

Firebase

5.2 Section-2

5.2.1 Phase 1:

The main functionalities of the system were implemented in this phase. (Talk about scrum)

Reservation Free Parking Module (available for VIP users and Normal users)

Check Availability:

* One of ParQU main features is to give users the ability to view the current parking area status. This is done by displaying the parking area map with a status shown for each parking spot.
* From the hardware’s point of view, an Arduino will detect if any user enters or leaves the parking spot using a sensor that is installed at each spot. The Arduino then informs the NodeMCU with the change in status, and the NodeMCU updates the spot status in the Firebase database. On the other hand, from a software’s point of view, the application or website will keep listening to any change in the parking spots’ status in the Firebase database and reflect the change directly onto the map without needing to refresh the page.

Reserved Parking Module (available for VIP users only)

Reserve a parking:

* Another main feature that ParQU provides to VIP users is the ability to reserves a parking spot for several hours. The VIP user is allowed to reserve 24 hours before the reservation time and can reserve up to 6 hours per day, whether it was multiple or one reservation. Additionally, the VIP user is charged with 5 QR per hour.
* In order to reserve, a VIP user must select the date on the application or website. When the date is selected, the status for each hour will be displayed based on the number of reservations in each hour. The application displays the status of each hour through colors. Blue for “available” and red for “not available”. Afterwards, the VIP user selects the reservation start time and its duration. Then, the application or website adds the reservation in the Firebase.
* To check in the reserved parking lot, the VIP user must scan his/her UID tag onto the RFID reader installed at the gates of the parking lot. The Arduino sends the UID tag received from the RFID reader to the NodeMCU. The NodeMCU then communicates with the Firebase to check if the VIP user has a reservation at this time and sends the response back to the Arduino. After, the Arduino opens the gates to the VIP user if it receives the positive response from the NodeMCU.
* Furthermore, to check out from the reserved parking lot, the VIP user must also scan his UID tag onto the RFID reader and the same process as the check in above is applied. Along with that, NodeMCU checks if the VIP user has exceeded his reservation time, if so, a penalty is added to the reservation. Each extra hour price is tripled (15 QR per hour) as a penalty.

Extend a reservation:

* The VIP user can extend his current reservation with one hour at the last hour of his/her reservation given that there is a parking spot empty. Additionally, the VIP user can extend (one hour per extension) as many times as he desires if the above conditions hold. The VIP user will be charged 5 QR per extension and the extended hours will not be part of the allowable reservation time (6 hours per day). This is ParQU way of making use of the recourses available as efficient as possible.

Cancel a reservation:

* A VIP user can cancel a reservation with 50% of the reservation price refunded. The reservation can be cancelled any time from reserving in the application or website till the reservation start time.

Show reservations:

* The VIP user can view a list of current and upcoming reservations. Each one of the reservations has 2 options which are “Extend” and “Cancel”.

Request car care:

* The VIP user is redirected to the Servesni application that provides various car care services.

5.2.2. Phase 2:

As a way to improve the system, some functionalities were improved and other new functionalities were added in this phase. (Talk about scrum/ imporved to meet project objective, purpose, etc)

Reservation Free Parking Module (available for VIP users and Normal users)

Get Directions:

* This feature gives users the ability to get directions for specific parking spot. Each spot is associated with specific coordination (latitude, longitude). By using Google Maps, we are able to give users directions to the wanted parking spot from his/her current location (device location).

Availability percentage:

* This feature shows the availability percentage of spots in each zone to the users. A progress bar is used to illustrate the availability percentage. The progress bar color will be changed based on the percentage. Green if more than 75% of the spots are available, orange if 50% of the spots are available and red if 25% of the spots are available.

Reserved Parking Module (available for VIP users only)

Reserve a parking:

* The availability of each spot in each hour was added to the reservation page as an improvement.
* The availability status (Available or not available) of each hour is improved by instead implementing the availability percentage through colors. Green if >75% is left*,* orange if 50% is left, red if 25% is left and gray if 0% is left.

Cancel a reservation:

* A VIP user can cancel part of the reservation (i.e. after the reservation has started). This feature was added as an improvement to phase 1’s cancel feature. Moreover, 50% of the price of each cancelled hour is refunded to the VIP user

Notification:

* A VIP user is aided with a reminder about his/her current reservation expiry time. The application/website notifies the VIP user 30 minutes before expiry time.

Automatic cancellation:

* A VIP user can cancel his/her reservation without needing to cancel through an application or website. This is done when the VIP user checks out of his reservation by at least one hour before the reservation ends. The check-out process is the same as the normal process explained above (phase 1: reserve a parking), but NodeMCU also checks how many hours are left in the checked out reservation. If the reservation still has an hour or more left, NodeMCU automatically cancels the remaining hours as partial cancellation (explained above), refunds the VIP user and updates the Firebase accordingly.

**Features that are available in both modules**

Currently looking:

* It gives the users the ability to know how many people are viewing certain zone in the current moment.
* It was quite challenging to implement this feature as the implementation logic in the application was different from the website. In the website, it was a bit easy to know how many one visit specific page by using IP address. However, this logic is impossible to be implemented in the application.
* To implement this feature, we were thinking about adding an attribute called currently looking in the zone node in the database. This attribute acts as a counter whenever people visit the page this counter will be incremented and vis versa.
* For application implementation, we benefit from a concept called "activity life cycle" where each activity goes through several stages. we benefited from onStart () and onStop (). whenever the activity gets started, onStart () method will be invoked so we should place the logic needed for increasing the number of people looking inside the onStart () method. In contrast, the logic needed for decrementing the number of people looking was placed in onStop () as this method will be called whenever the user leaves the page (activity).
* After implementing this logic, we figure out that it will not work properly in the case that the user lost internet connection. Logically if the user lost the connection the number of people looking should be decremented by 1, however in reality the number stay as it is. The reason behind that was the database has not been updated as there is no internet connection.
* To solve this issue, we thought about adding new relation called “currently looking” in the database to store zone name, login time and temporary id for each access. Whenever the user closes the page, the stored record for this user will be removed. Whenever other users visiting page, 2 things will happen:

1. All the records that are older than 10 minutes will be removed.
2. Show users how many people are currently looking.

By applying this logic, we can give an approximate number of how many currently looking, as in some rare cases the user might stay looking in the page more than 10 minutes.

Current Occupancy Trend:

Switching zones:

* As a way to show the scalability of ParQU. A switch button is installed to switch between two zones for each module.
* With each update from any of the two main modules, the Arduino informs the NodeMCU with the update along with which zone. After that, the NodeMCU updates the Firebase with the zone taken into consideration.

5.2.3. Phase-3

In this phase, the initial design of the application/website is improved, and the final prototype is shown along with the hardware implementation.

* Realistic photos were added to the zone list
* Minimize the number of clicks (app case => where to put histogram, website case=> ??)
* Consistency between website and application design
* Prototype

Section-3

Challenges

* We started the website design with bootstrap template in order to design the website faster and easier as it contains HTML and CSS-based design templates for forms, buttons, navigation and other interface components. After building the required pages, we re-restructured the website again as we figured out there were a lot of elements that can distracted the user. So, we removed any unnecessary components and pages. Again, we re-revised the website design and reduced more pages and components in order to minimize the number of user’s click as much as possible to make the user experience more user-friendly and streamlined. This required Moreover, colors and images have been changed and unified between the website and the application to ensure the consistency. As a result of all above,
* It gives the ability for both users to view the current parking area status by displaying the parking area map with a status shown for each parking spot. It required a hardware and software implementation.
* then the value will be sent to the NodeMCU which is in term updating the Firebase Database. On the other hand, the application or website will keep listen to any change in the spot status in the Firebase Database and reflect any changes directly to the map without needing to refresh the page.
* can start cancelling the reservation from the reserving time to the start time of reservation.
* was added to the check availability service to give the user the ability to get directions for specific spot. It required a software implementation only.
* Each spot is associated with specific (latitude, longitude) coordinate. By using google map we were able to get the location and the directions from current location of the user (device location) to the spot location.
* Also, his feature was added to check availability service to tell the user the availability percentage of spots in each zone.
* This service is already implemented in the phase-1 however in the phase-1 the user can cancel the whole reservation only, so we improved this service by letting the user to cancel part of the reservation. 50% of the price of each cancelled hour will be refunded to the VIP user.
* Providing this feature will help VIP user as many of them might forget about the expiring time of their reservation. It is mainly about notifying the VIP user 30 minutes before expiring time.
* the design and to implement the prototype and connect it with hardware
* For knowing the left percentage, for each hour the code will loop through all the reservation that are not cancelled and count how many reservations in this hour. Now the left percentage is computed by (number of reservation in specified hour/ total allowable reservation per hour) \* 100.