Product Requirements for Duke Parking App

Shengxin Qian

Functional requirements

1. End users' requirements:

1.1. The Duke Parking app shall offer at least 4 options for end users to look up expected parking lot which include but not limited to:

- Expected Duke parking lot name
- The destination name or address
- The nearest parking lot
- List of all parking lot (include marking parking lot on Duke Maps).

1.2. The users' could find software on Duke Appstore, login with Netid logging system and submit users' permit type.

1.3. The search function shall display the expected parking lot or best parking lot choice depending on users' search option and users' submitted permit type.

1.4. The Duke Parking app shall offer information of each parking lot and itemized as below and some information is available without Internet:

<u>Offline Available:</u>

- GPS location and location on Duke Maps
- Street address
- Permit requirement
- Number of total spots
- Open time (When is this parking lot available)
- Popularity (How quickly the parking lot would be full. This parameter is to help users determine the availability of each parking lot)

- Nearby buildings' name (This parameter help users determine parking lots through their expected destinations)

  Offline Unavailable:
- Number of available spots
- Anonymous or Non-Anonymous Feedback of former users of each specific parking lot
- Special announcement (This part help users to know specific adjustment of former information, such as open time adjustment in summer, …)

1.5. The Duke Parking app shall offer link to the guidance of each platform default Map app (Google Maps on Android, Apple Map on iOS).

1.6. The Duke Parking app shall offer option to set parking lot information refresh rate.

2. System managers' requirements:

2.1. The Duke Parking app shall generate the real time report of vacancy and permit type distribution of each parking lot;

2.2. The Duke Parking app shall generate the analysis of vacancy of parking lots include but not limited to the relationship between different periods of time or location and vacancy rate.

2.3. The Duke Parking app shall generate a report of users' feedbacks of any period of time and offer reply function for maintenance personnel.

2.4. The app shall allow manager modify parameters of each parking lot.

Non-functional Requirements

1. Product requirement

1.1. The information of each parking lot shall set time period related refresh rate to balance the precision of information and power consumption on mobile device (The refresh rate shall be higher during the rush hour and lower on the other period).

1.2. The Duke Parking app shall be intuitive and user-friendly, no training time required.

1.3. The Duke Parking app shall work without Internet, update as much information as possible and display offline available information when Internet connection is not reliable.

1.4. The Duke Parking app shall offer Android, Apple and Web versions.

2. Organizational requirement

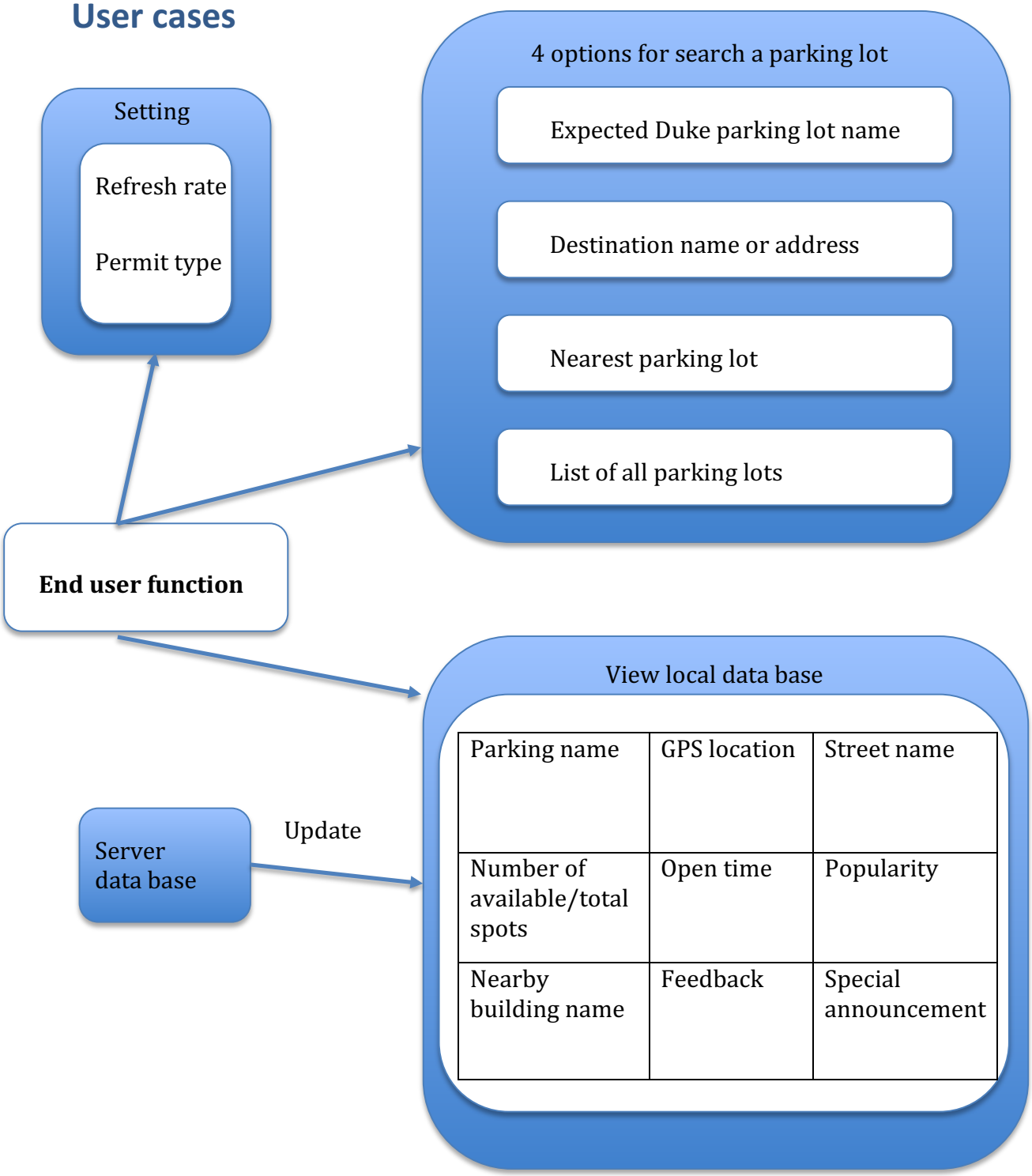
2.1. The Duke Parking app unlocks corresponding functions depending on users' type and access level.

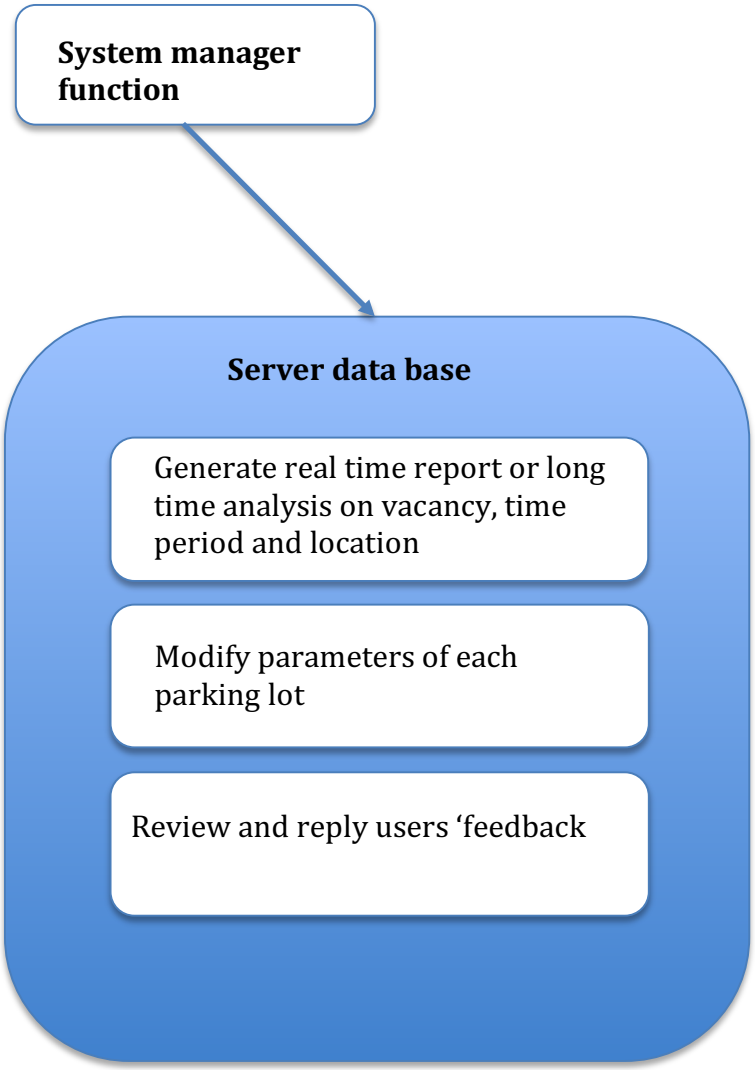
3. Information Security

3.1. The maintenance personnel shall not access to users' operation record on server and see Netid logging history.

3.2. The Duke Parking app shall only save permit type on local storage. The Netid and other Duke account correspond information shall not save on local storage or server.

User cases


Setting
Refresh rate
Permit type
4 options for search a parking lot
Expected Duke parking lot name
Destination name or address
Nearest parking lot
List of all parking lots
End user function
View local data base
Parking name
GPS location
Street name
Number of available/total spots
Open time
Popularity
Nearby building name
Feedback
Special announcement
Server data base
Update


System manager function
Server data base
Generate real time report or long time analysis on vacancy, time period and location
Modify parameters of each parking lot
Review and reply users 'feedback

System Requirements

1. Users' part

1.1. When Parking app is opened, app shall try to connect to remote server. If connection failed, Parking app shall remind user of Internet connection is broken, stay offline mode and retry building connection with default refresh rate. Otherwise, Parking app shall update each parking lot information in local storage and stay online mode.

1.2. At the start interface, when users choose:

- Enter expected parking lot name: App shall sort all parking lot in local storage with ascending sequence of distance from expected parking lot to the rest and display the sorted list.
- Enter the destination name or address: App shall sort all parking lot in local storage with ascending sequence of distance from destination to each parking lot and display the sorted list.
- The nearest lot: App shall request current GPS location. If request failed, sort all parking lot in local storage with descending sequence of the number of users' parking times and display.
- List all Parking lot: App shall display default list of all parking lot in local storage and offer map view choice. Map

view choice offer distribution of all parking lot on saved Duke map.

1.3. After app returned list of sorted parking lot, app shall reorder the list with users' submitted permit type and the required permit type of each parking lot. The Parking app shall put the user accessible lots at front, inaccessible at last and mark the accessibility of each parking lot.

1.4. After user choose one parking lot, Parking app shall display all offline available information listed above in local storage. The most recent updated offline unavailable information shall be displayed with "Not the last" marked.

1.5. Parking app shall have link to guidance of system default Map app when each parking lot detailed information was displayer.

1.6. Parking app shall allow user to change local storage refresh rate range from 1 min per time to no automatic refresh. (Too frequent refresh would cause large Internet data and mobile device power consumption. Users have option to manually refresh local storage at any time.

1.7. Parking app shall display users' Netid on their non-anonymous feedback for maintenance personnel to reply directly through Duke email. App shall display "Anonymous" on anonymous feedback.

2. System managers' part

2.1. The Parking app shall generate the real time report when maintenance personnel need it and click generate button. The

Parking app shall have option for system managers to generate analysis report in one week, one month or one year.

2.2. The Parking app shall generate real time feedback every 10 minutes and offer option direction send the non-anonymous feedback to users' Duke email.

2.3. The Parking app shall generate a settled form for managers to modify the corresponding information of each parking lot in database.

3. Product requirement

3.1. The default refresh rate shall be 1 min per time in rush hour and 10 min per time in the rest of time in a day.

3.2. If the app stays offline mode, it shall show the last updated information and mark the last update time on offline unavailable information. If the app stays online mode it shall update the local storage depending on refresh rate. If the Internet connection is not reliable, the app shall stay offline mode.

3.3. App server shall get real time available spot data of each parking lot from Duke RFID parking system.

4. Information Security requirement

4.1. The app shall option for users to choose users' type (end users' or system manager) and logging to Netid account to check the access level.

4.2. The users' operation on server shall not available to system managers. Only Duke OIT managers who have right to access Netid account to check operation on server.