(Minimum of 5 systems)

Sharvy-

<https://www.sharvy.com/en/>

One currently available turnkey system that provides some similar functionality to the candidate system is Sharvy, which allows you to create user accounts that can access a calendar on a website or mobile app to view the availability of parking spaces and make bookings. Sharvy provides a dynamic map which helps direct users to their reserved parking space. You can also create admin accounts that can view graphs produced by the system which visualise the occupancy of the car park and statistics that show trends. The system is also capable of managing bookings at multiple sites within the organisation so could be used for booking spaces in car parks that are closer to the user’s destination. However, Sharvy is a PMS primarily designed for managing company car parks and thus, does not have payment methods built into the system which are required for the University’s parking management system. Overall, the system is a convenient way for drivers to find spaces and for admins to monitor the sites but as it lacks a payment system, it is not a viable solution.

Swarco-

<https://www.swarco.com/solutions/parking/parking-management-solutions>

Similarly to the candidate system, Swarco allows users to register accounts, view parking availability, book reservations and make payments through a web browser or their “SwappAccess” mobile app. They can also receive directions to an available space and whilst the drivers cannot communicate with the admin through text messages, they can get assistance via an intercom. Like Sharvy, the system can manage parking at multiple sites which could be used for creating bookings in the location closest to the user’s destination. Swarco also provides a parking occupancy management system called “Count Controller” that produces information on the number of cars in the car parks and the occupancy of individual parking spaces. Overall, the system is good for helping drivers find spaces and make payments and produces useful information for the admin but lacks the functionality to allow drivers and admins to communicate via text messages.

Smart parking – features: using cameras to detect number plates, useful for monitoring customer’s cars

<https://www.smartparking.com/uk/our-system>

The Smart Parking system records a large amount of data and statistics on their “SmartCloud” hub which can be accessed by the admin to monitor the car park and its occupancy. It can record a driver’s time of arrival, the time they’re allowed to park and their time of departure. Like the candidate system, if a driver overstays their allotted time, the system can automatically send a report to the admin containing the driver’s information. However, unlike Sharvy or Swaco, Smart Parking does not allow drivers to register with the system and make bookings, it also only lets the users make payments on site at pay and walk machines. The system also does not provide a way for drivers and admins to communicate. Overall, the system produces a lot of information in the SmartCloud which is useful for assisting the admin in managing the car parks, but as the drivers cannot create acounts and make bookings and payments it lacks the functionality to be a viable solution.

Parkable –

<https://parkable.com/parking-management-system>

Another available solution is Parkable, which also fits many of the criteria for the proposed system and is like the Sharvy and Swarco systems. This allows users to reserve spaces in advance through a simple calendar interface that shows dates in one-week blocks. Users are also able to share their space with other users, using the same calendar interface to denote the days when they want to do this (a useful feature but given that most of our customers will be members of the public, it is not needed for our PMS). When looking at available spaces, users can see EV charging points (which can also be charged for via the app). If the system is used on multiple parks, users can see how many spaces are available per park and how many are taken, displayed using a simple green bar that gradually fills up (which is classed as an admin feature for our PMS system). The app has facilities for charging users, allowing for easy payments. Admin accounts can be used to sync with access control technologies and generate usage reports. Their website claims that this app can be used for universities. Overall, this system is a good match for our required PMS, with many matching features and some additional ones that would be useful to have. However, it has a few extra features that would not be needed such as access control syncing for admins and shared parking spaces.

Workero:

<https://www.workero.com/parking-management-solution/>

Workero’s parking solution allows for the definition of both fixed and pool parking. Spaces with EV charging stations can be highlighted (and their use can be controlled) along with spaces for people with disabilities. The service can also connect with other booking services to prevent double bookings. The admin account can handle data management and reports and can connect to access control devices. Regular users can use the service’s payment features.

Wayleadr:

<https://wayleadr.com/>

* Occupancy shown on front page, shown as gradually filling circle
* Exact number of spaces available shown to user with total spaces
* Gives option to enable notifications when spaces are below a certain amount
* Make bookings of spaces for the future and send notifications to customers about their bookings
* Can integrate the app with existing systems like gates and cameras and other access control systems
* “Forecast” of car park availability for certain days is calculated to avoid congestion
* Admin account can track metrics such as total bookings overtime and rejected bookings as well as measuring how these have increased and decreased over time
* Admins can also see who is currently in the car park and when they’re parking
* Users can release their spaces for other drivers to use
* Can be used to manage car park access with select people being allowed in