1. Mission statement
The purpose of the EasyParking database system is to provide users of the UoL (University of Liverpool) car parks an easy to use system to pre-book parking spaces and pay the parking fee. Further to this, the system should collect, store, manage and control access to the data that supports the car parking for students, staff and visitors in the UoL.

2. Mission objectives
2.1 General objectives
I. The system must include a fully functioning database that keeps data on parking customer’s bookings and data on the parking customer themselves (in terms of their car and identity).

II. The system must include a web-interface that allows users to use the system (place bookings, view bookings and any other services offered to users in this system) and allows administrators to manage the system after authentication.

2.2 Specific objectives
I. The system must be able to maintain (enter, update, and delete) data on users (students, staff or visitors and their respective cars)

II. The system must be able to maintain (enter, update, and delete) data on bookings.

III. The system must be able to show the amount of available parking space for a specific time

IV. The system must be able to query data on users.

V. The system must be able to query data on bookings. VI. The system must be able to calculate parking fees.

VII. The user must be able to place a booking for a parking space and enter information about themselves as a customer.

VIII. The system must have authentication and authorisation controls for the administrator

IX. The administrator must be able to enter SQL (data manipulation language) queries

X. The system must be able to generate a unique code that allows users to view their booking and provide proof for their booking

XI. The system must be able to generate a report showing the times in which the car park has been at the highest or lowest capacity

XII. The system must be able to generate a report on the amount of parking fees that have been paid and the total that this has come to for the current month.

XIII. The user must be able to login to the system using their login ID which is recommended to be

Right here, after the mission statement, one needs some explanation, with examples of the type of system one has in mind, with examples of use. This explanation could then be formalized in a more precise Mission Objective section … this could take the form of a list or a section of paragraphs, whichever seems more appropriate.
their student ID number and password

XIV. The user must be able to register an account to be able to use the system.

XV. The system must be able to provide a visitor option instead of making the user create an account

2.3 Extended objectives

I. The user should be able to pay for the parking fees via PayPal

II. The system should be able to maintain data on transactions that have occurred via PayPal

After Section 2, the document is just a dull sequence of diagrams, tables, and lists. More textual material, connecting the various parts would be needed.
3. System boundary diagram

![System boundary diagram]

Is this really THREE different user types??? Or just one?

4. User views and requirements

<table>
<thead>
<tr>
<th>User view Function</th>
<th>Description</th>
<th>Access to data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visitor view</td>
<td>Non-regular user of the system.</td>
<td>Read-only information on available bookings</td>
</tr>
<tr>
<td>Member view</td>
<td>Regular user of the system with an account.</td>
<td>Read-only information relating to bookings they've made and their account details</td>
</tr>
<tr>
<td>Administrator view</td>
<td>Controls and maintains the database</td>
<td>Read and write to all data</td>
</tr>
</tbody>
</table>

VERY SIMPLE USER VIEWS. ALMOST NO PROCESSING AT ALL!
5. Transaction requirements

<table>
<thead>
<tr>
<th>Data entered</th>
<th>User views involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Booking information</td>
<td>Students or professors</td>
</tr>
<tr>
<td>Identity information</td>
<td>All users</td>
</tr>
<tr>
<td>Parking time</td>
<td>All users</td>
</tr>
<tr>
<td>Car information</td>
<td>All users</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data deleted or updated</th>
<th>User views involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fee information</td>
<td>Administrator</td>
</tr>
<tr>
<td>Booking information</td>
<td>Students or professors</td>
</tr>
<tr>
<td>Car information</td>
<td>All users</td>
</tr>
<tr>
<td>Identity information</td>
<td>All users</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data queried</th>
<th>User views involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Booking information</td>
<td>Students or professors</td>
</tr>
<tr>
<td>Available space</td>
<td>Car owners</td>
</tr>
<tr>
<td>Car information</td>
<td>All users</td>
</tr>
<tr>
<td>Fee information</td>
<td>All users</td>
</tr>
</tbody>
</table>

6. System requirements

1. Initial database size: contain at least 1000 tuples
2. High frequency of use: about thousand times per day
3. Connect to network and mobile App
4. Use identity information to log in
5. Have recovery system and quick recover once the main system breaks down
6. Back-up once new data entered
7. Project Gantt chart

The Gantt chart provided seamlessly works for most projects ever carried out in this module. There is NOTHING specific about it. A totally uninformative diagram.

A lot of information missing in this section:
1. The individual tasks are NOT elementary (each of them will probably require weeks to complete)
2. task assignment is not given (who is going to do what)
3. temporal dependences, or any other constraining missing.

---

8. References


Where are these used?? Why are they required?

Generally speaking, a very limited proposal.
Nothing too wrong with a parking scenario, but then why not being more daring?
The system could have managed MANY car-parks (in fact the restriction to cars is in itself NOT needed). A vehicle could have been stored on a short/medium/long term agreement. Spaces available could have been of different sizes. One could have thought of a system whereby stored vehicles could have been moved around while parked (as long as they were returned to the customer in perfect working order) to optimize the number of available parking spaces. The system could have had a mobile app for customers, or operators (with number plate reading, perhaps?) … and I can think of many other requirements that could have been added to this.

THE RULE IS ALWAYS THE SAME: THINK BIG!