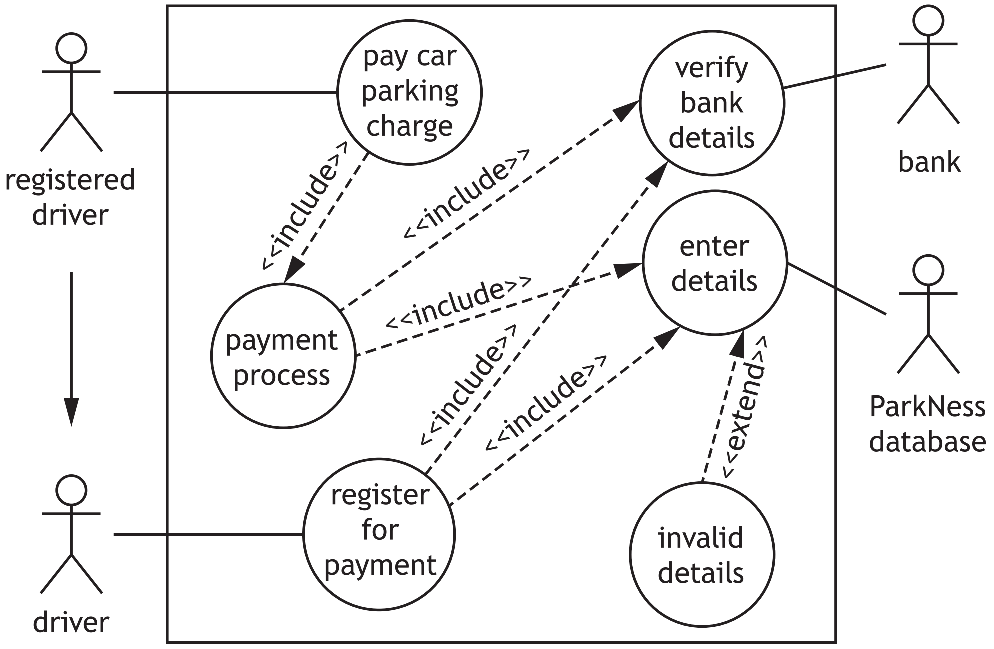
7. ParkNess manages six car parks in Inverness city centre using a database-driven website.

Drivers can use the ParkNess app to pay for their parking by providing the licence plate of their car and details of their bank payment card.

The UML use case diagram below illustrates the functionality of the ParkNess app.



(a) (i) Describe the relationship between the actors ‘driver’ and ‘registered driver’. **1**

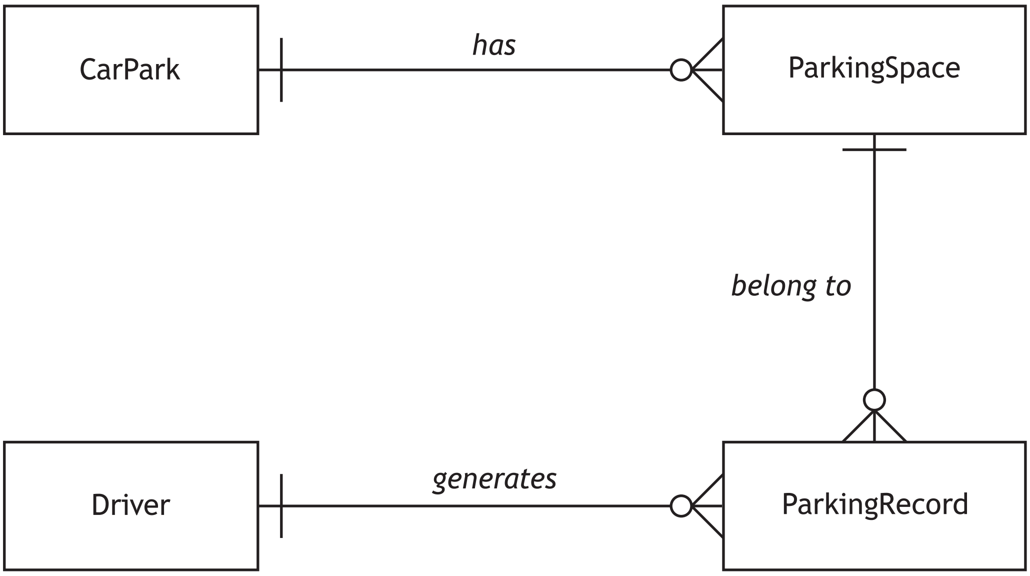
(ii) Describe the use of the ‘extend’ relationship between the ‘enter details’ and ‘invalid details’ use cases. **1**

7. **(continued)**

ParkNess makes use of Automatic Number Plate Recognition to record the licence plates of parked cars. Each parking space has a sensor that records whether it is occupied. The records for every parking space in the car parks are stored in a relational database. The structure of the database is shown below.

CarPark	**ParkingSpace**	**ParkingRecord**	**Driver**
carParkID	spaceID	licencePlate	driverID
carParkName	carParkID *	entryDate	email
capacity	occupied	entryTime	paymentCardNumber
		departureTime	expiry
		spaceID *	
		cost	
		driverID *	

(b) The entity-relationship diagram for this database is shown below.



(i) State whether each entity in this entity-relationship diagram is strong or weak. 2

(ii) Describe the relationship participation between the `ParkingSpace` and `ParkingRecord` entities. 2

[Turn over

7. (continued)

Sample data to be stored in each table of the database is shown below.

CarPark		
carParkID	**carParkName**	**capacity**
cp1	Rose Street	95
cp2	Telford Street	100
cp3	Bank Street	150
cp4	Clachnaharry Road	80
cp5	Ness Walk	64
cp6	Stadium Road	144

ParkingSpace		
spaceID	**carParkID**	**occupied**
33687	cp4	0
41722	cp2	1
…	…	…

Note that a value of 0 in the `occupied` field indicates that the space is unoccupied.

Driver			
driverID	**email**	**paymentCardNumber**	**expiry**
145	stevenm@inverness.co.uk	1111222233334444	08/21
158	daveb@ness.com	2222111133334444	09/21
193	melanieph@north.com	3333444411112222	10/21
…	…	…	…

ParkingRecord						
licencePlate	**entryDate**	**entryTime**	**departureTime**	**spaceID**	**cost**	**driverID**
SM68 JIX	2021-04-08	08:55	15:18	33687	10.50	145
SS16 PVM	2021-04-09	11:12	19:50	54256	13.50	158
DB19 SJS	2021-04-07	18:08	20:19	12346	4.50	193
SM68 JIX	2021-04-09	09:12	16:22	36027	12.00	145
SJ19 DBM	2021-04-07	09:43	17:02	12346	12.00	127
MI17 BSS	2021-04-09	07:22	18:17	65431	13.50	149
GC15 SNE	2021-04-08	11:10	13:18	12346	4.50	137
…	…	…	…	…	…	…

(c) Describe the output produced from the following query. **1**

```
SELECT spaceID
FROM ParkingSpace
WHERE NOT carParkID = "cp4"
AND occupied = 0;
```

7. (continued)

(d) (i) The SQL query below is used to display details of those drivers who parked in a ParkNess car park at least 10 times during April 2021.

```
SELECT email, licencePlate, COUNT(*) AS [Number of
Occasions]
FROM Driver, ParkingRecord
WHERE Driver.driverID = ParkingRecord.driverID
AND ____________________________________________________
GROUP BY email, licencePlate
_______________________________________________________;
```

Using appropriate Advanced Higher operators, write the missing conditions needed to complete this query. **2**

(ii) Andrew works near the Rose Street and Bank Street car parks.

The SQL query below is used to show the name of these two car parks along with the number of available spaces in each car park.

```
SELECT carParkName, capacity - COUNT(*) AS [Spaces
Available]
FROM CarPark, ParkingSpace
WHERE CarPark.carParkID = ParkingSpace.carParkID
AND ____________________________________________________
GROUP BY carParkName;
```

Using the `IN` operator, write a condition needed to complete the `AND` clause of this query. **1**

[Turn over

7. (continued)

(e) ParkNess uses a SQL query to display details of individual parking spaces from all car parks other than the Ness Walk car park.

Each parking space displayed by the query has a total income which is more than twice the income generated from all the parking spaces in the Ness Walk car park that were used between 14 and 26 April 2021 inclusive.

Complete the design of this query by writing the missing parts labelled A, B and C of the query design below. **3**

<table>
<tr><td colspan="2">Field(s) / Calculation(s)</td><td colspan="3">spaceID, carParkName, totalIncome = sum(cost)</td></tr>
<tr><td colspan="2">Table(s)</td><td colspan="3">CarPark, ParkingSpace, ParkingRecord</td></tr>
<tr><td colspan="2">Search Criteria</td><td colspan="3"><b>A</b></td></tr>
<tr><td colspan="2">Grouping</td><td colspan="3">spaceID, carParkName</td></tr>
<tr><td rowspan="4">Having</td><td rowspan="4"><b>B</b></td><td rowspan="4">Inner Query</td><td>Field(s) / Calculation(s)</td><td><b>C</b></td></tr>
<tr><td>Table(s)</td><td>ParkingSpace,
ParkingRecord,
CarPark</td></tr>
<tr><td>Search criteria</td><td>entryDate between 2021-04-14 and 2021-04-26</td></tr>
<tr><td>AND</td><td>carParkName = "Ness Walk"</td></tr>
<tr><td colspan="2">Sort Order</td><td colspan="3">spaceID ASC</td></tr>
</table>

(f) Before using the car park for the first time, drivers have to register using an online form.

An extract of the HTML form code is shown below.

```
<input type="text" id="driverID" name="driverID" required>
<input type="email" id="email" name="email" required>
<input type="text" id="paymentCardNumber"
name="paymentCardNumber" required>
<input type="text" id="expiry" name="expiry" required>
<input type="submit" value="Submit"> </form>
```

(i) Explain the use of `required` in the code above. **1**

(ii) Describe how code would be used to ensure the `paymentCardNumber` is exactly 16 characters long. **1**