1(a)

It is important to have a clear definition of the problem to be solved before the analyst starts work. For example, he has to define the problems encountered in the existing system in order to avoid making the same mistakes again in the new system. He has to clearly specify all the requirements of the new system that will help in doing a good outline development plan with cost/benefit analysis to avoid disagreements with the Company, SMRT over development cost issues in future.

Clear definition of the problem would also help to ensure the objectives and all requirements will be met. For example, if the SMRT does not clearly define the extensions of the new lines to be completed in a specific time line, say 9 months, then there might be delayed of completing tasks according to the planned schedule. (other logical examples will also be accepted)

1(b) Diagrams could include: [any three with 2 marks each].

- data flow diagram [1] — shows what is happening to data in the system [1]
- system flow chart [1] — shows overview of new system [1]
- program flowchart [1]— describes algorithm used of the program [1]
- decision tree/table [1] — for showing the possible actions to be carded out under a given set of conditions [1]
- structure chart [1]— used to express a solution tackled using a top down approach [1]
- E-R diagram (Entity Relationship Diagram) — shows relationships between entities/tables [1]

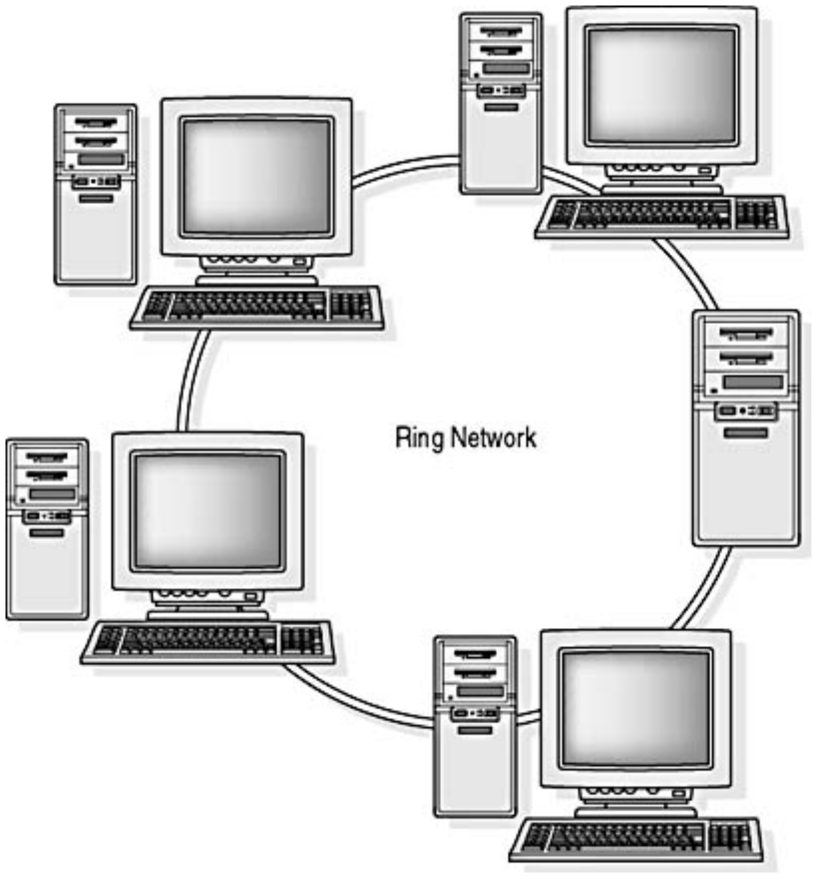
HIPO chart (High-Level Input-Process-Output) [1] — shows overall organisation of a program to present a high level view of the functions performed by a system, as well as the decomposition of functions into subfunctions, and so on.

2(a)

- LAN is restricted to a small geographic area / WAN is as dispersed as necessary
- LAN can be hard wired /WAN requires some other type of communication medium
- LAN can be served by digital information / WAN often needs information type to be altered
- LAN is secure because it can be easily controlled / WAN is more difficult to

2(b)

In a *ring* network, all workstations and servers are connected in a closed loop. There are no terminating ends, signal travel in one direction only around the ring using token passing therefore, if one computer fails, the entire network fails. Each computer in the network acts like a repeater and boosts the signal before sending it to the next station. If a station wants to send data, it will get a free token travel along the ring then attaches the data and indicates the address of the receiving station, then send the data. When the token reaches its destination computer, the data is removed and the token sent on.



Advantages of a Ring Topology in preference to a star network

- It is less expensive than the star network. Because similar equipment is linked together, no central processor is required.
- Expanding a ring network generally involves only the physically contiguous hardware.
- Using a token-passing communications routine prevents major contention problems and has higher transmission rate. As previously mentioned, each node waits for the electronic token before transmitting data.
- Each computer may communicate with any other computer in the ring directly.

3(a)

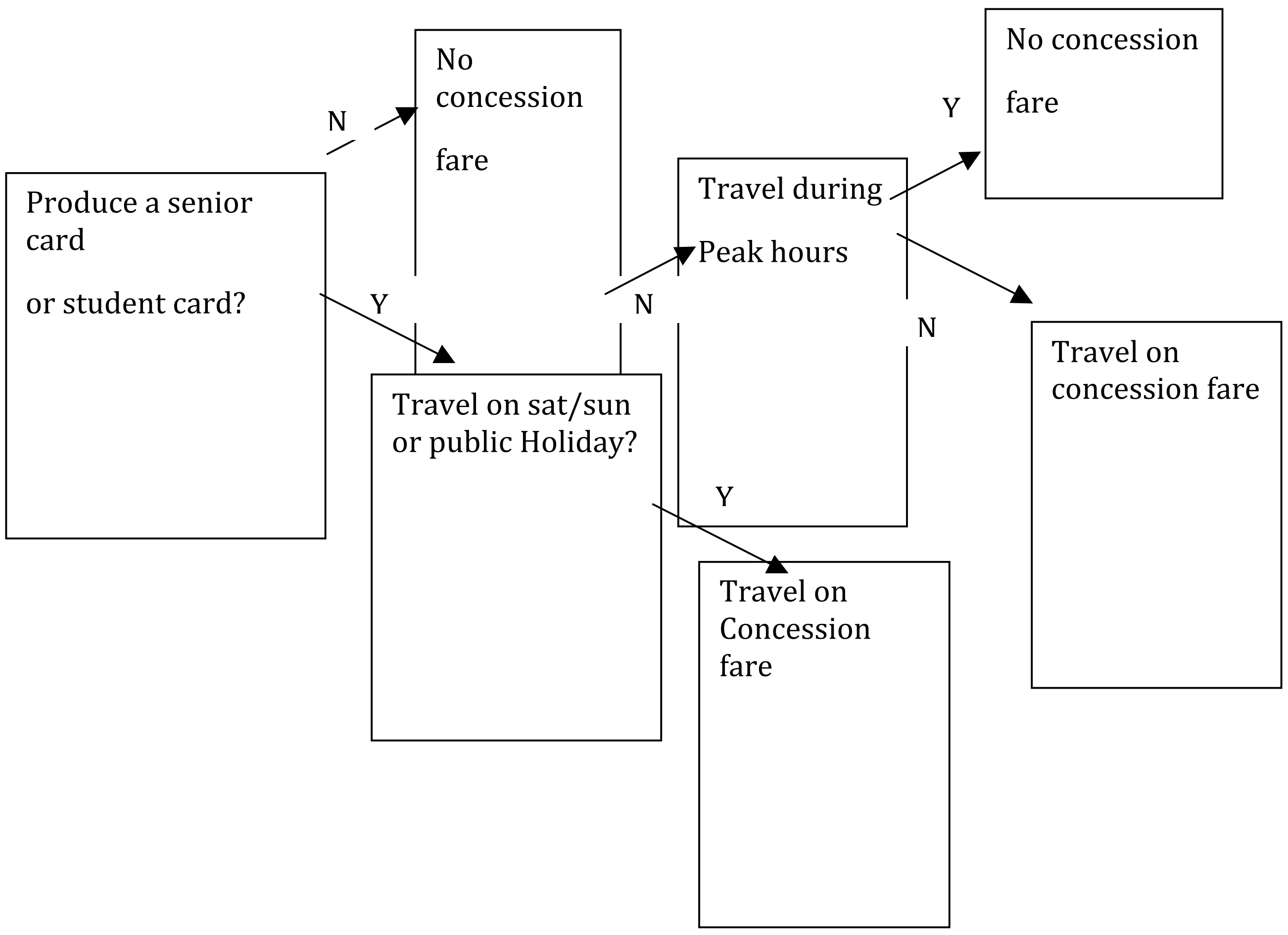
Short for **Network operating system, NOS** is the software that allows multiple computers to communicate, share files and hardware devices with one another. Some examples of network operating systems include Novell NetWare, Microsoft Windows NT, Microsoft Windows 2000, Microsoft Windows XP, Sun Solaris, Linux, etc...

3(b)

The interface should be user friendly with LED screen that display the information of train route. For example, the active route maps indicate the line of travel, direction of travel, side of doors opening, current station, next station, terminal station, interchange station and the corresponding line for interchange.

Interactive enquiries system with dialogue boxes for passengers to enquiry information of arriving time, departure time, travel during for one station to another, first train service and last train service time etc. over its web page.

3(c)


Produce a senior card or student card?
N
Y
No concession fare
Travel on sat/sun or public Holiday?
N
Y
Travel during Peak hours
Y
N
No concession fare
Travel on concession fare
Travel on Concession fare

4(a)(i)

```
Insert(Data)
If Queue is not full
  Create newnode and initialize with Data
  Back->next = newnode // link Back pointer to newnode
  Back = newnode  // update Back pointer of Queue
End-If
```

4(a)(ii)

```
Delete()
If Queue is NOT Empty
    ToBeDeleted = Front // link temp pointer to Front pointer of Queue
    Front = Front->next  // logically remove front most item from queue
    Return ToBeDeleted to computer // physically return memory to computer
End-If
```

4(b)(i)

- Add single parity bits in each byte during transmission to an agreed form, either even parity or odd parity. If bytes received are not the agreed form of either even or odd parity, then an error would be detected. (example required)
- Add a checksum byte to a group of bytes transmitted. Recalculated at receiver end to see if the group of bytes transmitted generate the same checksum byte that was received. (example required)

4(b)(ii)

- Request for a resend from sender when error detected
- Include a self-correcting code in message transmitted to self correct when error detected

4(c)(i)

- Information response must be within stipulated critical time frame

4(c)(ii)

- trip planner service where recommended connecting buses or trains could be suggested by the application for customer to reach the destination, and updated if customer misses a bus/train.

5(a)

Check digit provides a mechanism for detecting possible errors that is caused by transposition or possibly transcription errors.

e.g. The first 4 digits of a train code is 2378, Module 11 check digit system can be used to calculate its fifth digit, i.e. Check Digit, with the weightage for each digit, say 1, 2, 3, 4.

(2 x 1) + (3 x 2) + (7 x 3) + (8 x 4) = 61

61 MOD 11 = 6

Check Digit = 11 – 6 = 5

The 5-digit train code for the particular train is 23785

5(b)

(i) Select the first prime number which is bigger than 2000, say 2003
The hashing address of the train record with code number 23788 is

(23785 MOD 2003) = 1752

(ii) Two train codes which would cause a collision using your hashing algorithm are
23582 and 21579

Reason:

23582 MOD 2003 = 1549

21579 MOD 2003 = 1549

(iii)

First method – store in next available free block when collision occurs.
e.g. A train record with code = 23582 hash into block 1549
The record of this train will be stored in block 1549 if it is available. If block 1549 is not available as it has been occupied by record with train code 21579, it will be stored in block 1550 if it is available and so on. In other words, a search of an available block will be made from block 1550 onwards till a free block is found, and then it will be used to store the train record with code 23582.

Second method – store in an overflow area
Using the above example, the record with barcode 23582 will be stored in an overflow area which is linked from the block 1549 where the record with code 21579 is stored.

Block no		**Overflow block1**	**Overflow block2**	
1548				
1549	21579 *	* 23582 **	**19576	
1550				
1551				

The train record with code, 19576, will also hashed into 1549

Because MOD (19576, 2003) = 1549. It will then be stored in the overflow block2 as shown in the diagram.

[any correct illustrations with example each score full mark]

(iv)

Re-create a new random file using a different hashing algorithm that provides more blocks for storing more train records.

Or

Apply the same hashing algorithm to re- create a new Random File.

1. Read first record in the file,
2. If the record is not marked as deleted, hash to the new file.
3. If it is marked as deleted, copy the deleted train record into history file.
4. Read the next record.
5. Repeat the above step for the overflow area.
6. After processing the file, delete the old file and block overflow area.

6(a)
Backups are done more frequently (e.g. daily) while archives usually on weekly/monthly/yearly basis
Backups are copies of current working data, while archives are copies of past data
Backups are usually onsite for quick recovery, archives could be offsite for purposes of analysis or study

6(b)(i)

A module is a self-contained block of code that should be ideally lowly coupled and highly cohesive.

6(b)(ii)

Pass by value: copy of the data is passed in and stored in a local variable of the function. Any changes to this data will not affect the original value of the parameter that was passed in.

Pass by reference: the address of the original parameter is passed in, hence any changes within the function will affect the value of the original parameter

6(b)(iii)

```
timeDifference(h1, m1, s1. h2, m2, s2)
  swapIfNeeded(h1, m1, s1, h2, m2, s2) // function that make h1 time before h2 time if necessary
  total = 0
  hour = h2 – h1
  if (hour > 0)
     m2 = m2 + hour*60
  min = m2 – m1
  if (min > 0)
     total = total + min*60
  sec = s2 – s1
  total = total + sec
  return total
```

6(b)(iv)

Any two that are different in nature, e.g.

- across the hour
- within the hour

6(b)(v)

Most probably a logic error where there is a design flaw in the code such that the results produced was as expected. White-box testing can be carried out to identify the logic errors, like checking the code's loop conditions and selection control structure conditions, formula etc.

6(b)(vi)

```
count = 0
for x = 1 to N step 1
   if timeDifference(plan[x].h, plan[x].m, plan[x].s, actual[x].h,actual[x].m,actual[x].s)>60
       count = count + 1
   end-if
end-for
Print count
```

6(b)(Vii)

```
if N > 0
  Average = count/N
  If (Average > 0.05)
     Print warning indicator
End-if
```

Technical Documentation for maintenance, troubleshooting and further development purposes
User Manual – for reference
Training Guide – for training of end-users