Software Requirement Specifications

Amazon Bus Pass Maintenance

1. General Instructions

- This project is aimed at testing the hands-on skills and programming attitude to a practical problem.

- The program can be evaluated on the following parameters.
    1. **Data structure design**
    2. **Modularity:** The code shall be properly modularized and divided into different files as required.
    3. **Implementation**
        - Functionality implemented
        - Quality of code (Good Programming Practices)
        - The delivered code should compile without any build errors right-away. The Evaluator will use your project to build the code and if the build fails, the submission will be rejected outright.
    4. **Readability:** The code shall be easily readable and follow proper alignments and structure
    5. **Portability:** The code shall be completely portable and independent of the platform you are developing on.
    6. **Optimization:** The code shall be optimized to the extent possible. Take care of obvious cases like reducing the number of file writes, avoiding loss of data on unexpected program crashes.

- Consider non-functional requirements within the scope of the Problem Statement. However, the primary focus should be on Engineered Code.

- Provide Design document.

- Provide Test Data that you have used to test your program.

- Please report bugs in your program by yourselves in an Excel sheet.

- Packaging and delivery of the implementation is as important as correct implementation.

2. Problem Statement

Implement a system which will maintain complete information about the bus-pass details of every employee who has ever used the Amazon transport service.

Types of accesses: Administrator, User, Visitor

The developers need to implement administrator, user and visitor as different profiles. In different profiles, they have to implement the following features:

Administrator can:

1. Approve/reject bus pass applications, and allocate a bus pass-ID.
2. Appoint/change bus coordinators.
3. Generate reports that can be sent to the finance team/admin team for billing.
4. Add/remove routes.
5. Change the type of bus, number of buses on each route.
6. Change the route of the bus.
7. Number of vehicles of each type being used.
8. **To-do:** Vehicle maintenance.
9. Timing information about each bus.

Users can:

1. View all the routes, their stops, type of vehicles, no of vehicles on each route.
2. Request to cancel a bus pass / suspend a bus pass for a few months/ request to change to another route.
3. Request for new route/alteration of existing route
4. Give feedback to the administrator about the bus-facilities/improvements; or complaints on the driver.
5. Update their contact details.
6. Get a snapshot of their details (bus route, bus timing, bus type, stops, etc)

Visitors (Persons thinking of applying for bus pass) can:

1. View all the routes.
2. View percentage of seats occupied on each route.
3. Write to the administrator, requesting addition of a new route (if not already present).
4. Apply for a bus pass.