

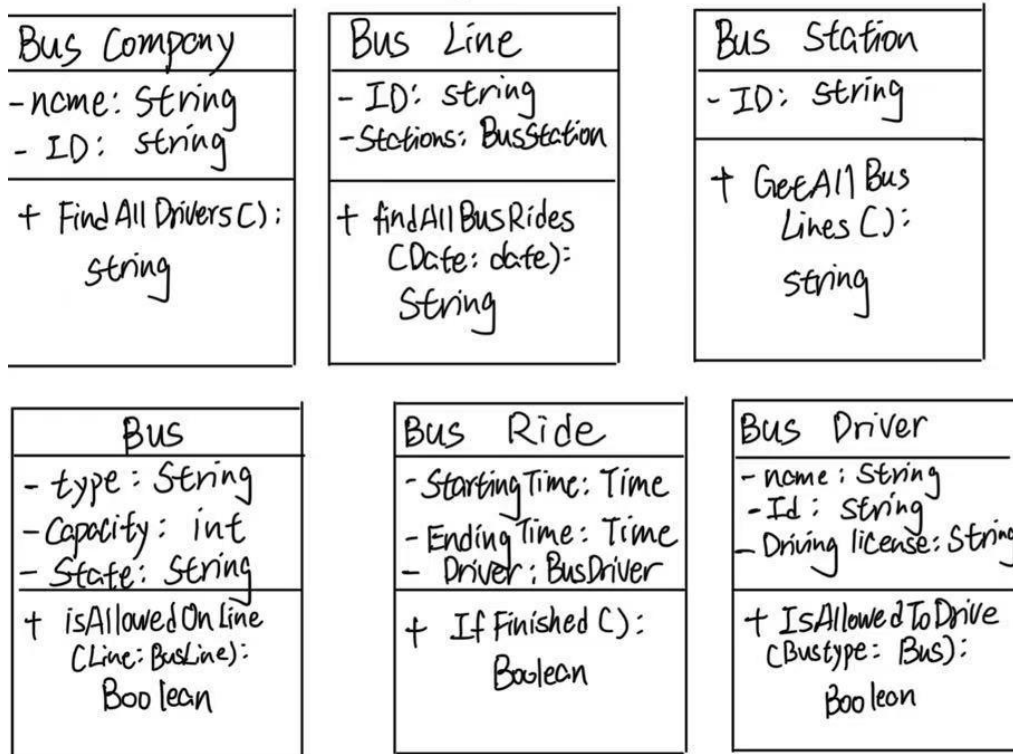
COMP 3211
Software Engineering

Zhu Jin Shun
22101071d

Assignment 2
Answers

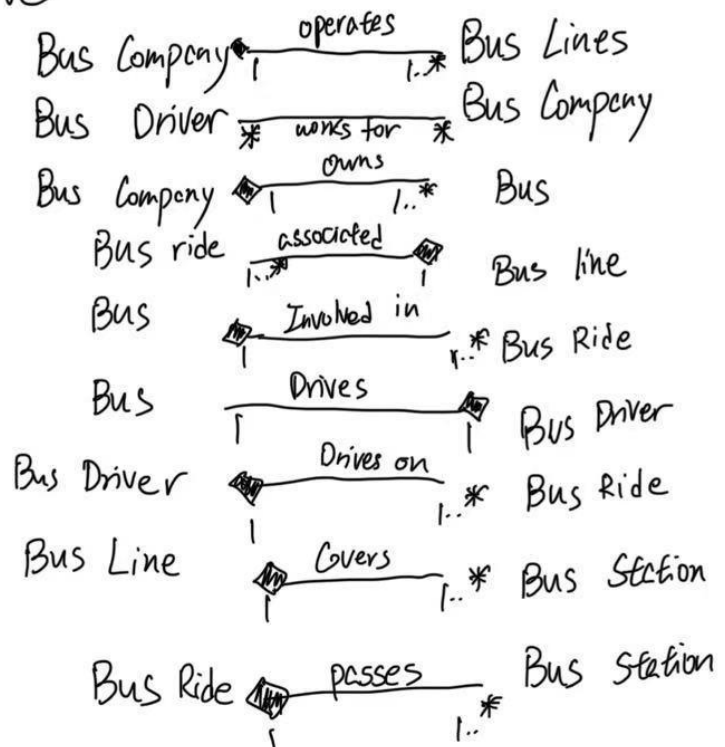
Question 1: Class Diagram

Question 1) Class Diagram

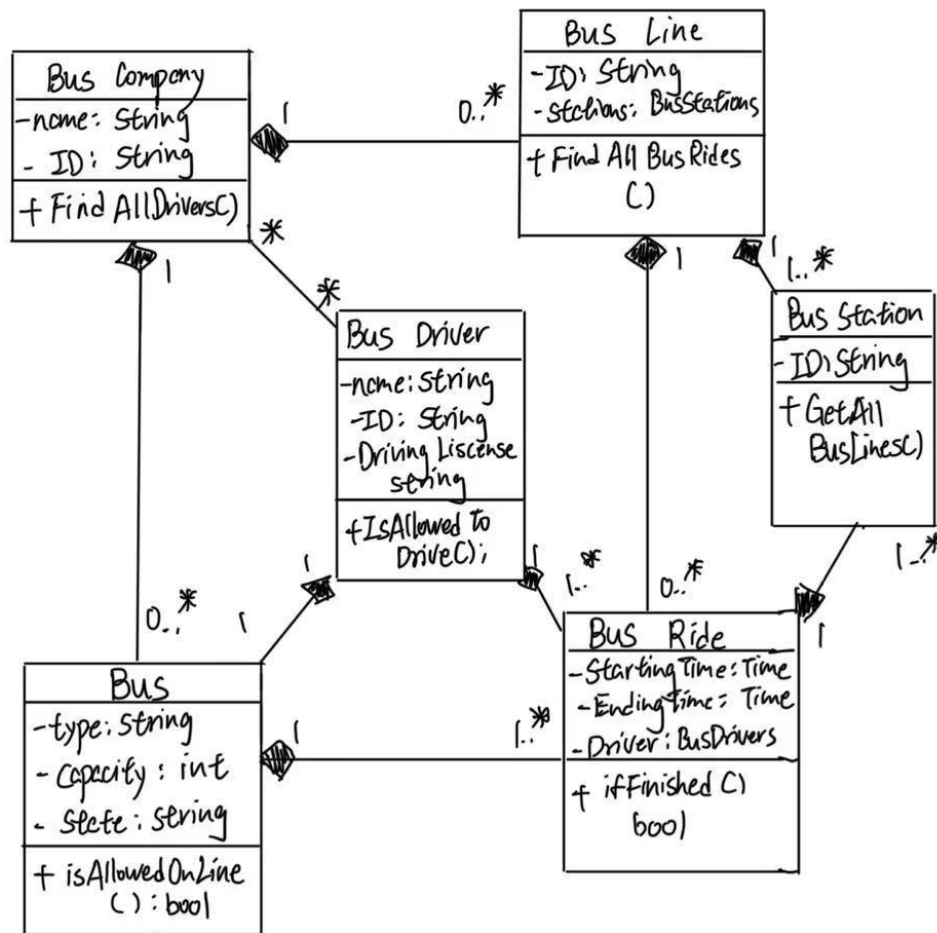


Designed classes above

Designed Relationships:



Question 1) UML Class Diagram



Question 2: Architectural Design

For Design 1:

In this design, both the presentation and application processing layers operate on the client side and only the database layer operates on the server side. The application processing functions are executed directly on the client, allowing reduced server load for the server. This setup enables clients to customize and modify application logic as needed, leading to improved satisfaction and

responsive user experience. By minimizing the amount of data sent between client and server, the application will have faster response times for users and will be essential in situations that require high interactivity and quick communication.

For Design 2:

In this design, where the presentation layer runs on the client side while the application processing and database layers are hosted on the server side, the server gains streamlined access to the entire system, encompassing both data storage and application logic. This centralization simplifies system maintenance and makes it easier to update or modify application logic, as any changes can be implemented directly on the server without the need to adjust client applications. This architecture enhances the overall manageability and efficiency of the system while ensuring all client applications are running under latest version of the system.

Question 3: Software Testing

Conditions:

Condition 1 on line 2 is c2 (if(x <= 0 || y <= 0 || z <= 0))

Condition 2 on line 5 is c5 (if(x + y <= z))

Condition 3 on line 7 is c7 (else if(x + z <= y))

Condition 4 on line 9 is c9 (else if(y + z <= x))

Tests	x	y	z	ExpectedResult	BranchesCovered
-------	---	---	---	----------------	-----------------

Test 1	1	1	1	true	c2==false, c5==false, c7==false, c9==false
Test 2	1	1	2	false	c2==false, c5==true, c7==false, c9==false
Test 3	1	2	1	false	c2==false, c5==false, c7==true, c9==false
Test 4	2	1	1	false	c2==false, c5==false, c7==false, c9==true
Test 5	0	1	1	false	c2==true, c5==false, c7==false, c9==false
Test 6	1	0	1	false	c2==true, c5==false, c7==false, c9==false
Test 7	1	1	0	false	c2==true, c5==false, c7==false, c9==false
Test 8	-1	1	1	false	c2==true, c5==false, c7==false, c9==false

Question 4: Software Maintenance

4 main types of software maintenance :

Preventive Software Maintenance: This maintenance aims to enhance the reliability and maintainability of application software while preparing for future

changes in both software and hardware environments. It involves proactively adding preventive functions to ensure the application system can adapt to various changes and remain competitive against new software solutions.

Perfective Software Maintenance: This type seeks to modify the software to expand functionality and improve performance. It primarily involves adding features and enhancements that were not specified during the system analysis and design phase.

Corrective Software Maintenance: This type of maintenance focuses on identifying and fixing errors or bugs that emerged during the system development phase but were not detected during testing.

Adaptive Software Maintenance: This maintenance focuses on modifying software to accommodate changes in information technology and operating environments.

Distinguishing between these types can be challenging due to overlapping activities, a single change might serve multiple purposes. For example, fixing a bug can also enhance performance, thereby relating to both corrective and perfective maintenance. Which will be difficult to distinguish each type of maintenance exactly during the development process. Preventive and perfective

maintenance is proactive while corrective and adaptive maintenance is reactive.