SUNWAY
INT'L BUSINESS SCHOOL

Programmed Name: **BCS**

Course Code: **CSC 1013**

Course Name: **System Analysis and Design**

Open book Examination

Date**: 10/2/2020**

Submitted By: **Submitted To:**

Student Name: **Dipesh Tha Shrestha** Name: **YOGENDRA BAHADUR MAHATA**

IUKL ID: **041902900028**

Semester: **second**

Intake:**2019 September**

CSC 1013_Dipesh tha Shrestha
System Analysis and Design

1. Name THREE (3) feasibility aspects used in systems planning, and for each provide an example of associated risk.

a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that a feasibility study examines, described below.

- Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

- Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

- Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't

zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

- Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

2. Compare generalization/specialization and part-whole relationships in terms of three differences.

Specialization and **generalization** are fundamental concepts in database modeling that are useful for establishing superclass-subclass relationships.
Specialization is a top-down approach in which a higher-level entity is divided into multiple *specialized* lower-level entities. In addition to sharing the attributes of the higher-level entity, these lower-level entities have *specific* attributes of their own. Specialization is usually used to find subsets of an entity that has a few different or additional attributes.

Generalization is a bottom-up approach in which multiple lower-level entities are combined to form a single higher-level entity. Generalization is usually used to find common attributes among entities to form a generalized entity. It can also be thought of as the opposite of specialization.

BASIS FOR COMPARISON	GENERALIZATION	SPECIALIZATION
Basic	It proceeds in a bottom-up manner.	It proceeds in a top-down manner.
Function	Generalization extracts the common features of multiple entities to form a new entity.	Specialization splits an entity to form multiple new entities that inherit some feature of the splitting entity.
Result	Generalization results in forming a single entity from multiple entities.	Specialization results in forming the multiple entity from a single entity.

3. Name and compare TWO (2) methods of collecting systems requirements in terms of two similarities and two differences.

Modern Methods for Collecting System Requirements

- Joint Application Design (JAD)
- Group Support Systems
- CASE Tools
- Prototypes

Traditional techniques for collecting requirements include

1) interviewing and listening,
2) administering questionnaires,
3) observing users, and
4) analyzing procedures and other documents.

Interviewing and Listening:

- Interviewing and listening involves talking with users individually or as a group to discover their views about the current and target systems; it also involves careful preparing an interview outline and guide before conducting the interview.

. Questionnaires:

- Administering questionnaires involves designing a questionnaire and determining who should respond to it; this method is typically used when there are too many key users to interview individually.
- Questionnaires are best when many people are involved, each person is to answer roughly the same questions, and people are remote or do not need personal care.

two similarities and two differences.

- +) Interviews provide large amounts of rich, detailed information, but -) they are expensive to conduct in terms of the time they demand.
- +) Questionnaires, on the other hand, can reach many people at once, making them relatively less costly than interviews, but -) the data collected in this way will not be as rich or as plentiful as is the case with interviews.
- Both techniques involve careful planning and execution to be successful.
- Deciding which technique to use will be dependent on such factors as - the size and complexity of the information system under study, - the size and complexity of the organization in which the system resides, - the funding available, and the expertise and - preferences of the analysts.

4. List FOUR (4) characteristics of the object-oriented methodology of systems development you study in this course.

Answer

The FOUR (4) characteristics of the object-oriented methodology of systems development is given below:

Encapsulation

Encapsulation is a process of information hiding. It is simply the combination of process and data into a single entity. Data of an object is hidden from the rest of the system and available only through the services of the class. It allows improvement or modification of methods used by objects without affecting other parts of a system.

Abstraction

It is a process of taking or selecting necessary method and attributes to specify the object. It focuses on essential characteristics of an object relative to perspective of user.

Relationships

All the classes in the system are related with each other. The objects do not exist in isolation, they exist in relationship with other objects.

There are three types of object relationships −

- **Aggregation** − It indicates relationship between a whole and its parts.
- **Association** − In this, two classes are related or connected in some way such as one class works with another to perform a task or one class acts upon other class.
- **Generalization** − The child class is based on parent class. It indicates that two classes are similar but have some differences.

Inheritance

Inheritance is a great feature that allows to create sub-classes from an existing class by inheriting the attributes and/or operations of existing classes.

1. Differentiates between Class Diagram and Sequence Diagram.

Class Diagram	Sequence Diagram
Class Diagram defines the types of objects in the system and the different types of relationships that exist among them	Sequence diagram is the diagram in which main representation is of the sequence of messages flowing from one object to another also main emphasis is on representing that how the messages/events are

	exchanged between objects and in what time-order.
Purpose of Class Diagrams 1. Shows static structure of classifiers in a system 2. Diagram provides a basic notation for other structure diagrams prescribed by UML 3. Helpful for developers and other team members too 4. Business Analysts can use class diagrams to model systems from a business perspective	Purpose of Sequence Diagram • Model high-level interaction between active objects in a system • Model the interaction between object instances within a collaboration that realizes a use case • Model the interaction between objects within a collaboration that realizes an operation • Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)
It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.	it depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
Class diagram describes the attributes and operations of a class and also the constraints imposed	Sequence diagram mainly focuses to represent interaction between different objects by pictorial

on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.	representation of the message flow from one object to another object. It is time ordered means exact interactions between objects is represented step by step.

Section B

1. Prepare a context diagram for an order system. (The context diagram for an order system must include-

   - Order system process is at the centre of the diagram

   . - There are five entities SALES RES, BANK, ACCOUNTING, single incoming data flows for commission, Bank Deposit and Cash Receipt entry, respectively.

   - The WAREHOUSE entity has one incoming data flow- picking list. (That is, a report that shows the items ordered and their quantity, location, and sequence to pick from the warehouse.

   - The WAREHOUSE entity has one outgoing data flow, COMPLETED ORDER..

   - Finally, the CUSTOMER entity has two outgoing data flows, ORDER and PAYMENT, and two incoming data flows, ORDER REJECT NOTICE and INVOICE.

   Answer

Context diagram is the highest level in a Data Flow Diagram. It is a tool popular among Business Analysts who use it to understand the details and boundaries of

the system to be designed in a project. It points out the flow of information between the system and external components.

It is made up of a **context bubble**, first drawn in the middle of the chart. It is usually a circle shape that represents a conceptual boundary that encloses a group of interconnected processes and activities of a project. The nitty-gritty details of the internal structure of a system are masked in a context diagram since it is strictly a high-level view of the system. This process is called information hiding.

context diagram for an order system is given below:

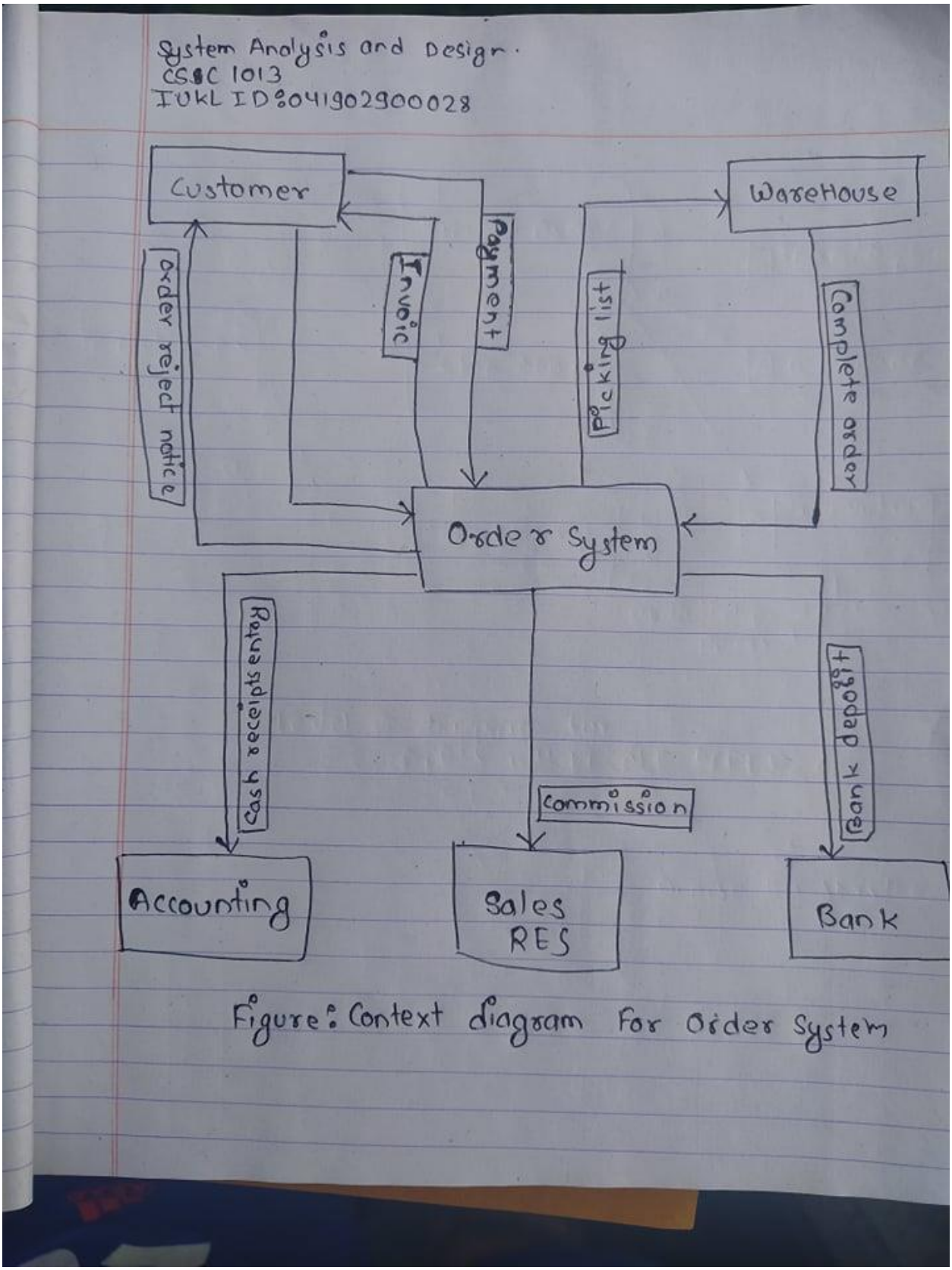


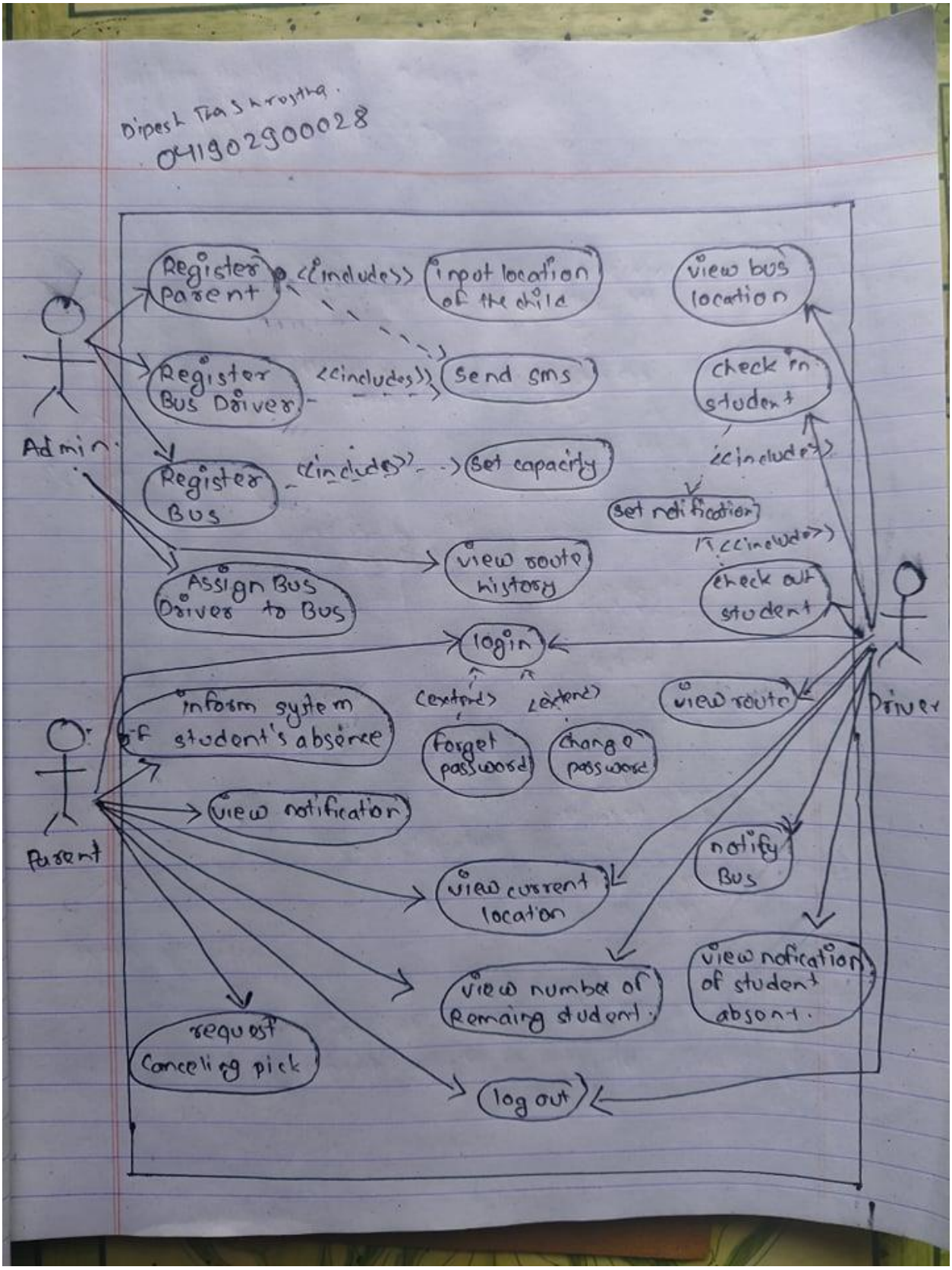
Figure: Context diagram for Order System

2. Define an actor, a use case, and draw a use case diagram for a school bus system that creates a new bus route?

Actors can be people, other systems, time triggers, or event triggers.
An actor specifies a role played by a user or any other system that interacts with the subject. It may represent roles played by human users, external hardware, or other subjects. Actors are always outside the system and interact directly with it by initiating a use case, provide input to it, and/or receive outputs from it. While a single physical instance may play the role of several different actors, an actor does not necessarily represent specific physical entities, i.e. a timer that triggers sending of an e-mail reminder.
A **use case** is a description of how a person who actually uses that process or system will accomplish a goal. It's typically associated with software systems, but can be used in reference to any process. For example, imagine you're a cook who has a goal of preparing a grilled cheese sandwich. The use case would describe through a series of written steps how the cook would go about preparing that sandwich. A use case helps you understand where errors could occur in the process and design features to resolve those errors.
Three elements that a use case must contain:
Actor, which is the user, which can be a single person or a group of people, interacting with a process
System, which is the process that's required to reach the final outcome
Goal, which is the successful user outcome

A use case diagram for a school bus system that creates a new bus route is given below:

CSC 1013_Dipesh tha Shrestha
System Analysis and Design



3. Assume you are a System Analyst for a consulting company and have been asked to assist the Chief Executive Officer (CEO) of a regional bank. Because of Covid-19 the bank recently implemented a plan to reduce the number of staff, including loan officers, as a strategy to maintain profitability, subsequently, the bank experienced chronic problems with backlogged loan officers who are able to review and approve or disapprove loans. The CEO of the bank is interested in solutions that would allow the approval process to move faster without increasing the number of loan officer and has engaged your company to come up with suggestions. What is one type of system that you might recommended to the bank?

Answer

A **banking system** is a group or network of institutions that provide financial services for us. These institutions are responsible for operating a payment system, providing loans, taking deposits, and helping with investments.

Transaction Processing System

A small business processes transactions that result from day-to-day business operations, such as the creation of paychecks and purchase orders, using a transaction processing system, or TPS. The TPS, unlike a batch system, requires that users interact with the system in real time to direct the system to collect, store, retrieve and modify data. A user enters transaction data by means of a terminal, and the system immediately stores the data in a database and produces any required output.

For example, a small-business owner may direct a bank system to debit a savings account for $500 and credit the company's checking account for $500. Because of constant system updates, a user can access current TPS data, such as an account balance, at any point.

I will recommend Transaction processing system In the case of a bank account, a TPS keeps track of all the events associated with a single account: deposits, withdrawals, transfers, fees, interest paid, etc. This provides a good description of the account activity.

Now let's say the customer comes into the bank and requests a car loan. The account activity is useful information but not enough for the bank to make a decision on the car loan. This requires combining information from different sources and analyzing the financial profile of the customer.