

**2ND SEMESTER 2021/2022 ACADEMIC YEAR**

**MID SEMESTER EXAMINATION ANSWER BOOKLET**

THE FOLLOWING DETAILS MUST BE COMPLETED BY THE STUDENT

400

ADS19A00110Y

STUDENT’S ID NUMBER­­­­­­­­­­: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ LEVEL:\_\_\_\_\_\_\_\_\_\_

CS456

Advanced System Analysis and design

COURSE COD**E: \_\_\_\_\_\_\_\_\_** COURSE TITLE:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FERDINAND HIAGBE

LECTURER’S NAME: (Refer to the Question Paper) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**QUESTION NUMBER: (e.g. Q1) \_\_\_1\_\_\_SUB-QUESTION (e.g. 1(a))\_\_\_1A,B,C,D\_**

1. The primary roles I will take as the main systems analyst will be;

The consultant: The systems analyst may be employed particularly to handle information systems problems within a business because they regularly serve as systems consultants to people and their enterprises. The engagement of outside consultants might be advantageous since they can bring with them a new viewpoint that other employees within a firm may not have. Additionally, it implies that external analysts are at a disadvantage because they can never fully understand the organizational culture. As an outside consultant, you will significantly rely on the methodical approaches covered in this chapter to analyze and create information systems that are suitable for users operating in a certain industry. You will also depend on information system users to assist you in understanding organizational structure from other people's perspectives.

The supporting Expert: You can also be required to serve as a supporting expert for a company that routinely hires you in a systems-related function. The analyst employs their specialized knowledge of computer technology, software, and their applications in the workplace in this capacity. This task frequently involves making a tiny change or decision that just affects one department rather than a whole systems effort. You are not managing the project as the supporting expert; rather, you are acting as a resource for those who are. Many of your everyday tasks may be covered by this function if you work as a systems analyst for a manufacturing or service company.

Agent of change: The most thorough and accountable function a systems analyst can assume is that of a change agent, either for the business internally or externally. Every time you carry out a task from the systems development life cycle (described in the following section) and spend a significant amount of time dealing with users and the business, you are acting as an agent of change (from two weeks to more than a year). A person who acts as a catalyst for change, creates a strategy for change, and collaborates with others to facilitate that change is known as an agent of change.

1. The three qualities that I must possess are;

Communicator: The analyst needs to be proficient in both speaking and presenting. He needs to be a master of the language that the user can grasp. The systems analyst and users should be able to communicate with each other without any problems.

Problem solver: The analyst should be exceptionally creative in order to translate user ideas into workable proposals. He or she should be able to create systems and make plans by developing diagrams, charts, and other illustrations. By examining the current system, the analyst must be able to identify the users' actual difficulties. He is anticipated to offer the greatest answers to the issues. In order for the customers to choose the best solution, he should be able to offer more than one to a single issue. Any user issue must be able to be solved by the systems analyst. He needs to be a problem-solver rather than a problem-maker.

Self-disciplined and self-motivated: The systems analyst must be a self-disciplined, self-driven person with the ability to manage and organize several project resources in addition to other people.

1. Important activities I will perform as an analyst in the phase/stage are;

Activity:

* Interviewing user management
* Summarizing the knowledge obtained
* Estimating the scope of the project
* Documenting the results

And its Output will be:

* Feasibility report containing problem definition and objective summaries from which management can make a decision on whether to proceed with the proposed project.

1. The implantation Activity will be:

* Train users
* Plans the conversion from old system to new system
* Review and evaluate system

And my role/Output will be:

* Trained personnel
* Installed system

**QUESTION NUMBER: (e.g. Q2) \_\_2\_\_\_SUB-QUESTION (e.g. 2(a))\_\_2A,B,C,D\_\_**

1. Agile Approach in developing the system:

Agile lets teams offer value to their clients more quickly and with fewer difficulties through an iterative approach to project management and software development. An agile team produces work in manageable, small-scale increments rather than staking all on a "big bang" launch. Teams have a built-in mechanism for fast adjusting to change since requirements, plans, and results are regularly evaluated. Agile development's greatest benefit is that it helps teams produce results more quickly, with higher quality and predictability, and with a better ability to adapt to change. Agile methodologies typically encourage a structured project management approach that encourages frequent inspection and adaptation, a leadership philosophy that promotes teamwork, self-organization, and accountability, a set of engineering best practices meant to enable the quick delivery of high-quality software, and a business strategy that ties development to customer needs and company objectives.

1. Three things that would be required if the agile method is to work;

Exploration: You will investigate your surroundings while asserting your belief that agile development can and should be used to solve the problem. You will also assemble the team and evaluate team members' abilities. If you already know your team members and technologies, this step should just take a few weeks to a few months (if everything is new). Additionally, you will be actively researching potential technological requirements for the new system. You should get comfortable calculating how long certain chores will take at this point. Customers are experimenting with writing user stories as part of exploration. The goal is to convince the customer to refine a story sufficiently so that you can accurately predict how long it will take to integrate the solution into the system you're designing. At this stage, it's crucial to develop a fun and enquiring mindset toward the workplace, its issues, technologies, and employees.

Planning: Planning is the next phase of the agile development process. Planning, as opposed to the first stage, might just take a few days to complete. At this point, you and your clients decide on a timeframe—anywhere between two months and six months from the present—by which you will present them with solutions to their most pressing business issues (you will be addressing the smallest, most valuable set of stories). Depending on how well your exploring efforts went, this level should just take a few minutes.

Iterations: These typically last about three weeks and are known as iterations (cycles of testing, feedback, and modification). Even if it is only in outline or skeletal form, you will be pushing yourself to draw out the complete system's design. One objective is to execute functional tests created by customers at the conclusion of each iteration. You should consider whether the schedule has to be adjusted or if you are taking on too many stories during the revisions phase. Every successful iteration should become a ritual that both customers and engineers participate in. Celebrate every step forward you take, no matter how tiny, as doing so encourages everyone to work incredibly hard on the project.

1. The best information gathering methods i would adopt will be;

Interviewing in Information Gathering:

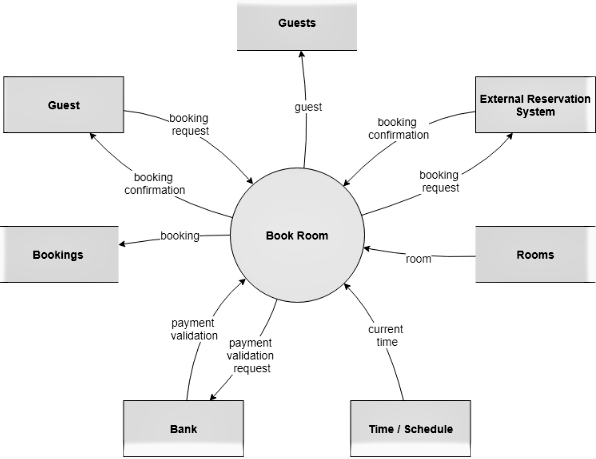
You must in fact interview yourself before you interview anyone else. You must be aware of your biases and how they may skew your judgment. For the information you will hear in your interviews, your education, intellect, upbringing, emotions, and ethical framework all operate as effective filters.

Prior to the interview, you must give it careful thought. Think about your motivation for attending, the questions you'll ask, and what, in your opinion, will make the interview successful. You must plan ahead on how to make the interview enjoyable for the interviewee as well. A question-and-answer formatted dialogue with a specified goal is known as an information-gathering interview. You want to learn the interviewee's ideas and feelings about the system's current status, his or her personal and professional aspirations, and informal protocols for engaging with information technologies.

Above all, ask the individual you are interviewing for their opinions. Facts may not be as significant or illuminating as opinions. For instance, consider questioning the proprietor of a brick-and-mortar company who has recently added an online store about the number of refunds she regularly provides to customers for online transactions each week. "About 20 to 25 a week," she responds. When you keep track of the transactions and see that the typical amount is only 10.5, You can draw the conclusion that the owner is exaggerating the situation. In-depth interviews are a great way to learn about goals. Hard data facts may be able to explain previous success, but goals look to the future of the firm. Interviewing should be used to learn as many of the organization's objectives as you can. Goals might not be discernible with any other data collection techniques. The interview is a great opportunity to discuss important HCI (human-computer interaction) issues, such as ergonomics, system usability, how appealing and fun the system is, and how effective it is at assisting with specific tasks. You are establishing a relationship with a potential stranger during the interview. While establishing trust and understanding rapidly, you also need to keep sway over the interview. Additionally, you must persuade the interviewee to buy the system by giving them the essential details. Do this by preparing for the interview in advance so that you can conduct it naturally.

1. CONTEXT  
   It depicts the complete software system as a whole because it represents all of the external elements that might interact with the system. This kind of diagram places the system in the center and shows its surroundings, related entities, and external components around it. The system's internal structure is not fully described in detail.

The parts of a computerized system that delivers and saves hotel information are shown in the image below. By enabling them to adjust their room rates and available rooms and making them visible on their sales channels, it helps hotel management manage their sales and online marketing operations. These include both traditional and internet travel agencies.



**QUESTION NUMBER: (e.g. Q2) \_\_4\_\_\_SUB-QUESTION (e.g. 2(a))\_4A,B,C,D\_\_\_\_**

1. A context diagram of MTN mobile money;

Functions:

ECG,

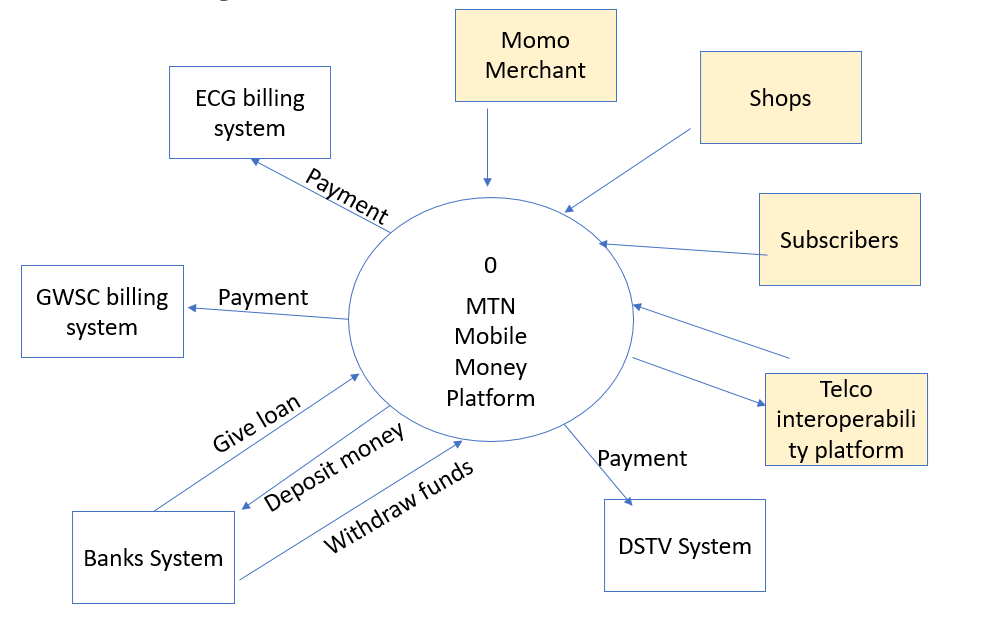
GWSC,

SHOP,

DSTV,

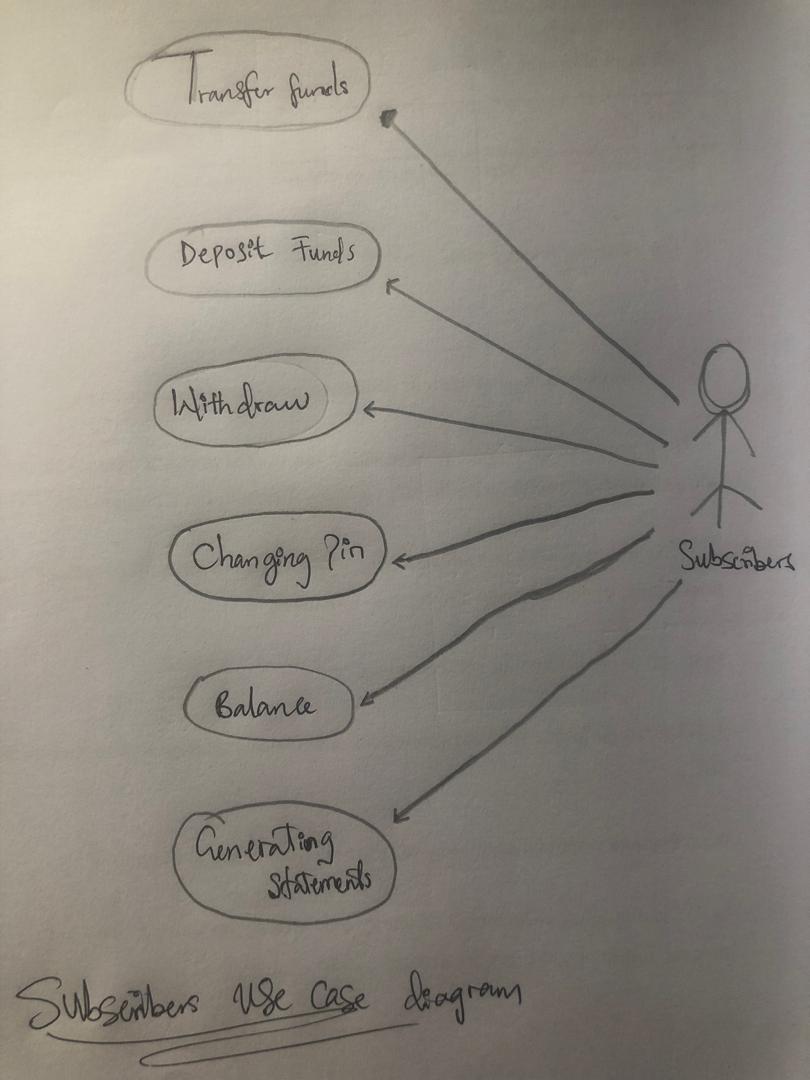
BANKS

Diagram below;



MTN MOMO CONTEXT DIAGRAM

1. Use case diagram involving subscribers only and the relevant transactions they perform:



1. A simple use case scenario for the Transfer Money function:

**Name of Use Case**: Transfer Money

**Actor**(s): Merchant

**Stakeholder**(s): Merchant, receiver

**Description**: This use case is meant to explain how a merchant will make transfer to a customer.

**Steps**:

|  |  |  |
| --- | --- | --- |
| **#** | **Actions** | **Remarks** |
| 1 | Dial \*170# |  |
| 2 | Input the necessary numbers required |  |
| 3 | Input phone number / reference (optional) | The number must be MTN |
| 4 | Input amount and verify name |  |
| 5 | Input password and click proceed | Pass word must be correct for transfer to be successful |

**Precondition**:

* The receiver must have an active account
* The receiver will be expected to receive a message for the transfer

**Post Condition**:

* The receiver balance must be updated
* System must be ready to records another transaction

**Risk**: ---

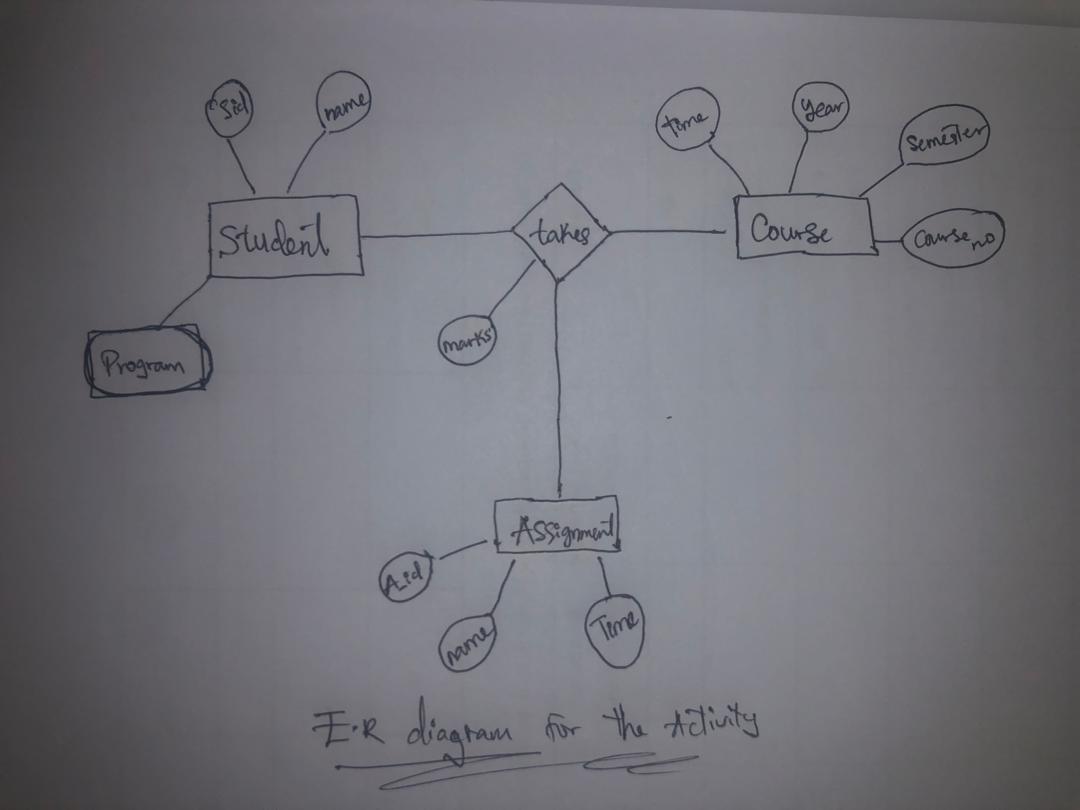
**Priority**: the receiver must receive the money

1. The importance in using a use case modeling in software analysis, design and development is that;

* Use cases help analysts define boundaries.
* Use cases can be traceable, allowing analysts to identify links between use cases and other design and documentation tools.
* Use cases effectively communicate systems requirements because the diagrams are kept simple.
* Use cases allow people to tell stories.
* Use case stories make sense to nontechnical people.
* Use cases do not depend on a special language.
* Use cases can describe most functional requirements (such as interactions between actors and applications).
* Use cases can describe nonfunctional requirements (such as performance and maintainability) through the use of stereotypes.

**QUESTION NUMBER: (e.g. Q2) \_\_5\_\_\_SUB-QUESTION (e.g. 2(a)) \_5A, B, C,\_\_\_\_**

1. The ER diagram of the activity;



1. The types of relationships used:

* One to many: An instance of X may be related to several instances of Y, but an instance of Y is linked to just one instance of X in a one-to-many relationship between the two entities. From the image above, we can say that many students can take one course.
* Many to many: A many-to-many relationship is one in which numerous instances of both X and Y are connected to one another through X, and vice versa. An example of a many-to-many relationship is shown in the image below.

When there is a relationship between two things, it means that they are somehow connected. As an illustration, a student might sign up for a course. As a result, the entities Student and Course are associated, and a connection between them is depicted as a connector.

1. The two-top level interactive information gathering methods i would use are;

Interviewing in Information Gathering: You must in fact interview yourself before you interview anyone else. You must be aware of your biases and how they may skew your judgment. For the information you will hear in your interviews, your education, intellect, upbringing, emotions, and ethical framework all operate as effective filters. Prior to the interview, you must give it careful thought. Think about your motivation for attending, the questions you'll ask, and what, in your opinion, will make the interview successful. You must plan ahead on how to make the interview enjoyable for the interviewee as well. A question-and-answer formatted dialogue with a specified goal is known as an information-gathering interview. You want to learn the interviewee's ideas and feelings about the system's current status, his or her personal and professional aspirations, and informal protocols for engaging with information technologies. Above all, ask the individual you are interviewing for their opinions. Facts may not be as significant or illuminating as opinions. For instance, consider questioning the proprietor of a brick-and-mortar company who has recently added an online store about the number of refunds she regularly provides to customers for online transactions each week. "About 20 to 25 a week," she responds. When you keep track of the transactions and see that the typical amount is only 10.5, You can draw the conclusion that the owner is exaggerating the situation. In-depth interviews are a great way to learn about goals. Hard data facts may be able to explain previous success, but goals look to the future of the firm. Interviewing should be used to learn as many of the organization's objectives as you can. Goals might not be discernible with any other data collection techniques. While establishing trust and understanding rapidly, you also need to keep sway over the interview. Additionally, you must persuade the interviewee to buy the system by giving them the essential details. Do this by preparing for the interview in advance so that you can conduct it naturally.

Joint Application Design (JAD) in Information Gathering: Regardless of how skilled you grow at conducting interviews, you will unavoidably come across circumstances in which one-on-one interviews do not seem to be as beneficial as you would like. Personal interviews are labor-intensive, prone to inaccuracy, and their data is open to misunderstanding. Joint application design (JAD), an alternative to doing individual user interviews, was created by IBM. Utilizing JAD is driven by the need to reduce the time (and associated costs) associated with conducting in-person interviews, enhance the accuracy of the outcomes of information requirements assessments, and increase user identification with new information systems through participatory processes. This method involves IT experts, business users, and project managers working together in discussion, learning, and project management groups to discuss the new information system. The objectives of the new software system can be developed and solved more easily with the involvement of both parties. Depending on how intense the seminars are, the meetings could go for hours, days, or even weeks.

To start a new project, the procedure is called joint application development. The JAD technique is applied to technology projects with Application Management which can be used for either internal organizational processes or for external commercialization. For stakeholders to work together and improve the system in the optimal way, workshops or meetings including both IT experts and business users are required. A key to success is having both viewpoints that can inspire creative ideation for the technology.