

GROUP 8 PRESENTATION

NAME	REG NO	SIGNATURE
ANTHONY MUSEMBI	SC150/0077/2022	
HENRY NDUNGU	SC150/0362/2022	
JERUSHA KURIA	SC150/0508/2021	
JUVENILE NAPUU	SC150/0356/2023	
GRACE KIARIE	SC150/4040/2022	

ADVANCED SYSTEM DESIGN

- Advanced system design is the process of creating a system that meets the complex needs of a business or organization.
- It involves considering a wide range of factors, including performance, scalability, security, and reliability.

Types of Advanced System Design

1. Distributed Systems

- Distributed systems are a collection of independent components located on different machines that share messages with each other in order to achieve common goals.

Advantages of Distributed Systems

- ✓ Scalability: Distributed systems can be scaled up or down by adding or removing nodes, which makes them well-suited for handling large workloads.
- ✓ Reliability: Distributed systems are more reliable than centralized systems because they are not dependent on a single point of failure.
- ✓ Performance: Distributed systems can often achieve better performance than centralized systems by distributing the workload across multiple machines.

Disadvantages of Distributed Systems

- ✓ Complexity: Distributed systems are typically more complex to design and implement than centralized systems.
- ✓ Coordination: The components of a distributed system must be carefully coordinated in order to achieve common goals.
- ✓ Consistency: It can be difficult to maintain data consistency in a distributed system, especially if the components are located in different time zones or are subject to network delays.

2. Real-Time Systems

- A real-time system is a system that must meet specific time constraints. This means that the system must produce the expected result within a defined deadline, or the system fails.

Real-time systems are used in a wide range of applications, including:

- ✓ Industrial automation systems
- ✓ Medical devices
- ✓ Transportation systems
- ✓ Aerospace systems
- ✓ Military systems
- ✓ Financial trading systems

Real-time systems are typically classified into two categories: hard real-time systems and soft real-time systems.

Hard real-time systems have strict deadlines that must be met. If a hard deadline is missed, the system fails. Examples of hard real-time systems include:

- ✓ Air traffic control systems
- ✓ Nuclear power plant control systems
- ✓ Anti-lock braking systems in cars

Soft real-time systems have less strict deadlines. If a deadline is missed, the system may still be able to function, but the performance may be degraded. Examples of soft real-time systems include:

- ✓ Multimedia streaming systems
- ✓ Online gaming systems
- ✓ Voice over IP (VoIP) systems

3. Embedded Systems

- An embedded system is a combination of computer hardware and software designed for a specific function within a larger system

Additional advantages of embedded systems:

- ✓ Embedded systems can be tailored to specific applications, which can lead to improved performance and efficiency.
- ✓ Embedded systems can be embedded in other systems, which can lead to reduced complexity and size.
- ✓ Embedded systems can be designed to be very reliable and fault-tolerant, which is critical for safety-critical applications.

Additional disadvantages of embedded systems:

- ✓ Embedded systems can be more difficult to develop and program than general-purpose computers.
- ✓ Embedded systems can be more expensive than general-purpose computers, especially for low-volume production.
- ✓ Embedded systems can be more difficult to maintain and update than general-purpose computers.

4. Client-Server Systems

- A client-server system is a type of computer network architecture in which computers are divided into two categories: clients and servers. Clients are computers that request services from servers. Servers are computers that provide services to clients.

Advantages of client-server systems:

- ✓ Scalability: Client-server systems are highly scalable, meaning that they can be easily expanded to support more users and more data.
- ✓ Reliability: Client-server systems are more reliable than other network architectures because they are not dependent on a single point of failure. If a server fails, other servers can continue to provide service to clients.
- ✓ Security: Client-server systems are more secure than other network architectures because they can centralize security and access control.
- ✓ Performance: Client-server systems can achieve better performance than other network architectures by distributing the workload across multiple servers.

Disadvantages of client-server systems:

- ✓ Complexity: Client-server systems are more complex to design and implement than other network architectures.
- ✓ Cost: Client-server systems can be more expensive to implement than other network architectures because they require additional hardware and software.
- ✓ Single point of failure: If the server that is providing a critical service fails, all clients that are depending on that service will be affected.

Examples of client-server systems:

- ✓ The internet is a client-server system. When you visit a website, your web browser (client) sends a request to the web server that hosts the website. The web server then sends the website's files to your web browser.
- ✓ Email systems are client-server systems. When you send an email, your email client (client) sends the email to the email server (server) that hosts your email account. The email server then sends the email to the email server of the recipient.
- ✓ File sharing systems are client-server systems. When you upload a file to a file sharing service, your computer (client) sends the file to the file sharing server (server). The file sharing server then stores the file and makes it available to other users.

5. Web-Based Systems

- ✓ Web-based systems are software applications that are hosted on a web server and accessed by users through a web browser. Web-based systems are becoming increasingly popular because they offer a number of advantages over traditional desktop applications.

Advantages of web-based systems:

- ✓ **Accessibility:** Web-based systems can be accessed from any device with a web browser and an internet connection.
- ✓ **Scalability:** Web-based systems can be easily scaled up or down to meet the needs of a growing or shrinking business.
- ✓ **Maintenance:** Web-based systems are easier to maintain and update than traditional desktop applications.
- ✓ **Cost:** Web-based systems are often more cost-effective than traditional desktop applications.

Disadvantages of web-based systems:

- ✓ **Performance:** Web-based systems can be slower than traditional desktop applications, especially if the internet connection is slow.
- ✓ **Security:** Web-based systems can be more vulnerable to security threats than traditional desktop applications. This is because they are exposed to the internet.
- ✓ **Offline access:** Web-based systems typically require an internet connection to use. This can be a disadvantage for businesses that need to access their systems offline.

Examples of web-based systems:

- ✓ Webmail
- ✓ Online banking
- ✓ E-commerce systems
- ✓ Content management systems (CMSs)
- ✓ Customer relationship management (CRM) systems

FILE ORGANIZATION AND ACCESS METHODS

- File organization refers to the way in which data is stored in a file, while access methods refer to the way in which data is retrieved from a file.

File Organization

There are three main types of file organization:

- ✓ **Sequential file organization:** In sequential file organization, data is stored in a file in a linear order. This means that each record in the file is stored after the previous record.
- ✓ **Direct file organization:** In direct file organization, each record in the file is stored in a specific location in the file. This location is determined by a key field in the record.
- ✓ **Indexed file organization:** Indexed file organization is a hybrid of sequential and direct file organization. In indexed file organization, an index is created to store the key fields of all the records in the file and the location of each record in the file.

Access Methods

There are three main types of access methods:

- ✓ **Sequential access:** Sequential access is used to read or write data in a file in a linear order. This means that the file must be read from beginning to end to find the desired data.
- ✓ **Direct/Random access:** Direct access is used to read or write data in a file at a specific location. This is done by using the key field of the desired record to determine its location in the file.
- ✓ **Indexed access:** Indexed access is used to retrieve data from a file using an index. The index is used to find the location of the desired record in the file.

DESIGN STRATEGIES

- ✓ Design strategies are a set of guidelines or principles that can be used to create effective and efficient systems.

Design strategies are important for a variety of reasons, including:

- ✓ To ensure that systems are easy to use and maintain.
- ✓ To improve the performance and reliability of systems.
- ✓ To reduce the cost of developing and implementing systems.
- ✓ To make systems more adaptable to change.

There are a number of different design strategies that can be used, but some of the most common include:

- ✓ **Modularity:** Modularity is the process of dividing a system into smaller, self-contained components. This makes the system easier to design, implement, and maintain.
- ✓ **Abstraction:** Abstraction is the process of hiding the details of a system and exposing only the essential functionality. This makes the system easier to use and understand.
- ✓ **Encapsulation:** Encapsulation is the process of combining the data and code that operate on that data into a single unit. This makes the system more modular and reusable.

- ✓ **Information hiding:** Information hiding is the process of restricting access to the internal data and code of a system. This makes the system more secure and less likely to be affected by errors.
- ✓ **Separation of concerns:** Separation of concerns is the process of dividing a system into different components, each of which is responsible for a specific task. This makes the system easier to design, implement, and maintain.