GROUP SIX PRESENTATION

	NAME	REG NO	SIGNATURE	DATE
1.	GILBERT NGARUIYA	SC150/0118/2022	--------------------------	---------------------
2.	SHYLEEN NKATHA	SC150/4043/2022	--------------------------	---------------------
3.	NYAOMA BRIAN	SC150/0462/2021	--------------------------	---------------------
4.	ROBIN CHEGE	SC150/0101/2022	--------------------------	---------------------
5.	MORRIS WAWERU	SC150/0108/2022	--------------------------	---------------------
6.	JOHN KIHARA	SC150/0524/2021	--------------------------	---------------------

System design

- Systems design is the process of defining system elements like modules, architecture, components, and their interfaces and data for a system based on the specified requirements.

Components of system design

The following are major components of the System Design.

1. **Load balancers:** The most crucial component for scalability, availability, and performance measures for systems. It involves delegating tasks over a set width of resources.
2. **Key-Value Stores:** It is a storage system where key-value stores are distributed hash tables. Examples of contemporary, large-scale key-value stores are **Amazon's DynamoDB** and **Microsoft Cassandra**.
3. **Blob Storage:** Blob stands for binary large objects, as the name suggests is storage for unstructured data such as YouTube, and Netflix.
4. **Databases:** It is an organized collection of data so that they can be easily accessed and modified.
5. **Rate Limiters:** These set the maximum number of requests a service can fulfill.
6. **Monitoring System:** This is software where system administrators monitor infrastructures such as bandwidth, CPU, routers, switches, etc.
7. **Distributed System Messaging Queue:** Transaction medium between producers and consumers. A **producer** creates messages and **consumers** receive and process them.
8. **Distributed Unique ID generator:** In the case of large distributed systems, every moment multiple tasks occur so to distinguish it assign a tag corresponding to every event.). These are 128-bit numbers that look like this: 123e4567e89b12d3a456426614174000 in hexadecimal.
9. **Distributed Search:** Over every website, crucial information that visitors will seek is put into the search bar. Search bars can be crucial for browsing large websites with hundreds or even thousands of pages. Most modern websites have search bars to help users find precisely what they're looking for. Behind every search bar, there is a **search system**.
   Search systems are composed of three main entities:

   - *Crawler*: finds/fetches content and creates documents
   - *Indexer*: builds a searchable index
   - *Searcher*: runs the search query against the index

10. **Distributed Logging Services:** Tracing sequences of events from end to end.
    - **Logging** is the process of recording data, in particular the events that occur in a software system.
    - A **log file** is the recording of these details.
11. **Distributed Task Scheduler:** Computational resources such as CPU, memory, storage, etc

- In computing, a **task** is a unit of work that requires computational resources for some specified amount of time.

Stages of system design

➢ The design stage in the software development process is crucial for creating a blueprint for the system based on the requirements gathered in the previous stage.

1. Requirement Analysis:

✓ Review and analyze the requirements gathered from stakeholders. Identify any ambiguities or gaps in the requirements. Work closely with stakeholders to resolve uncertainties and document a comprehensive set of detailed requirements. These detailed requirements will serve as the foundation for the design process.

2. System Architecture Design:

✓ Develop a high-level system architecture that outlines major components/modules, their relationships, and how they interact. Choose appropriate architectural patterns (such as client-server, microservices, etc.) and technologies. Consider scalability, performance, security, and other non-functional requirements.

3. User Interface (UI) and User Experience (UX) Design:

✓ Design the layout, navigation, and visual elements of the user interface. Focus on user experience principles, ensuring ease of use and efficiency. Develop wireframes, and prototypes, to visualize the user interface. Gather feedback from stakeholders and end-users to refine the design iteratively.

4. Database Design:

✓ Design the database schema based on the data requirements. Define tables, columns, keys, relationships, and constraints. Optimize the database schema for efficient data retrieval and storage. Consider normalization, indexing, and query optimization techniques. Plan for data integrity, security, and backup/recovery mechanisms.

5. Detailed Component and module Design:

✓ Define detailed functionalities, algorithms, data structures, and interfaces for each component/module identified in the system architecture. Specify how components will communicate and interact with each other. Create low-level design documents that provide a detailed blueprint for the developers, including class diagrams, sequence diagrams, and API specifications.

Inputs and outputs of system design

Inputs to a system design include:

- Requirements: The requirements of the system, as gathered from users and stakeholders.
- Constraints: Any constraints on the system design, such as budget, schedule, and technology.
- Existing systems: Any existing systems that the new system will need to interact with.
- User data: Any data that the system will need to process or store.
- Design standards: Any design standards that the system must comply with.

Outputs of a system design include:

- System architecture: A high-level overview of the system, including its components and how they will interact with each other.
- Detailed design: A more detailed design of each system component, including its interfaces and how it will be implemented.
- Test plan: A plan for testing the system to ensure that it meets the requirements and is free of bugs.
- Deployment plan: A plan for installing and configuring the system in its production environment.
- Maintenance plan: A plan for fixing bugs and adding new features to the system over time.

Example

Designing a new e-commerce website.

Inputs:

- Requirements from users and stakeholders, such as the ability to browse and purchase products, create and manage an account, and track orders.
- Constraints, such as budget, schedule, and technology.
- Existing systems, such as a customer relationship management (CRM) system and an inventory management system.
- User data, such as product catalogs, customer data, and order data.
- Design standards, such as web accessibility standards and security standards.

Outputs:

- System architecture, such as a three-tier architecture with a web server, application server, and database server.
- Detailed design, such as the design of the product catalog database, the shopping cart functionality, and the checkout process.
- Test plan, such as a plan to test the functionality, performance, and security of the website.
- Deployment plan, such as a plan to deploy the website to a cloud hosting provider.
- Maintenance plan, such as a plan to monitor the website for errors and to add new features as needed.