Assessment 1: Gore oriented requirements

Apply gore techniques to define goals and refine them into requirements

Task 1: use K.A.O.S to decompose the goal to improve an online shopping experience

Task 2: identify conflicts between the goals and suggest resolutions

Deliverables: KAOS Diagrams & conflict resolution plans

Answers:

Task 1

1.Main Goal: improve an online shopping experience.

2.Sub-Goals :

(a). Make the website easier to use.

(b). Keep customers happy.

(c). Ensure transactions are secure.

3.Requirements for each Sub-Goal:

(a) -Organize products into clear categories.

- Add an easy-to-use search and filter system.

- Improve loading speed so pages don't take forever to show up.

(b). -Provide 24/7 customer support.

- Make returns and refunds simple and stress-free.

-Personalize recommendations so customers find what they love faster.

(c).- Add multiple secure payment options .

-Add two-factor authentication for extra account security.

-Encrypt sensitive customer information to keep it safe.

Task 2

Identifying Conflicts:

-Making navigation simpler might mean cutting down on product details, which could make it harder for customers to make informed choices.

- Extra security measures and encryption could slow down the checkout process and frustrate customers.
- Providing 24/7 customer support could be expensive and difficult to maintain.

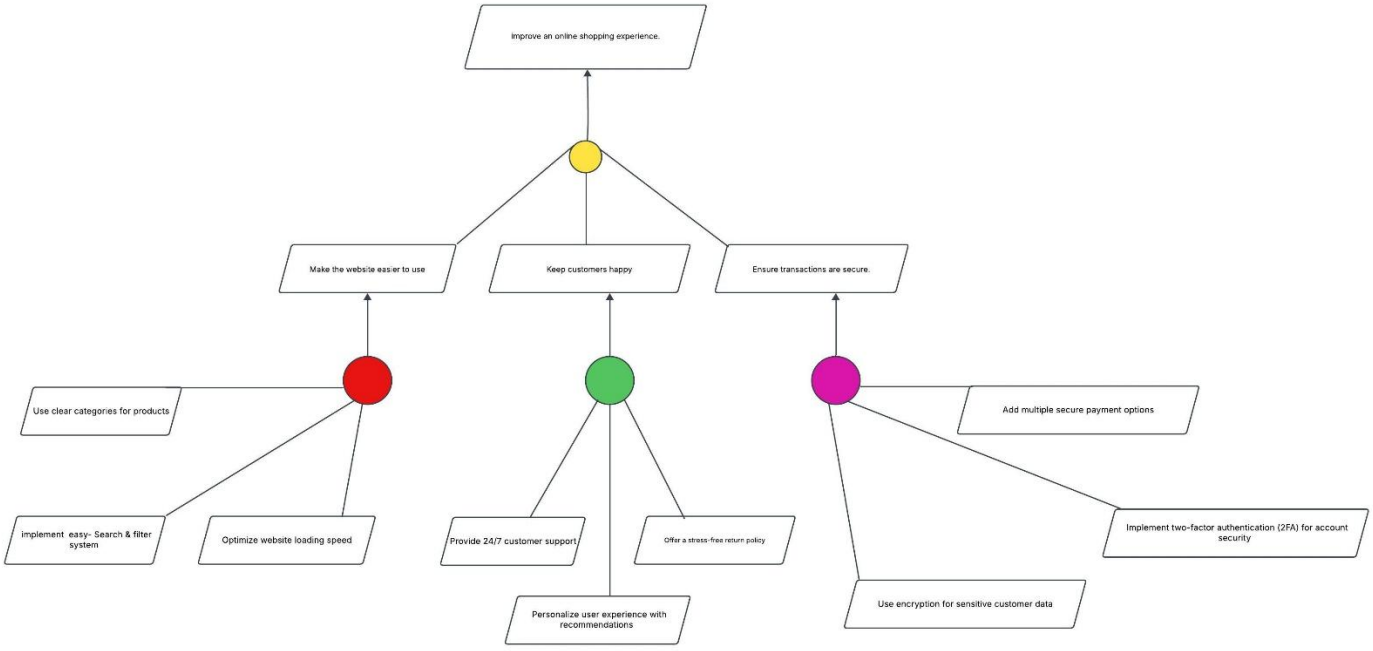
Resolutions:

-Create a separate product details page that's easy to find, so the main pages stay clean and simple.

-Use smart authentication, where 2FA is only required for high-risk transactions, keeping smaller purchases quick and hassle-free.

-Use AI-powered chatbots to handle basic customer questions, so human agents are only needed for more complex issues.

KAOS DIAGRAM



Assessment 2: Dependency analysis

Analyze dependencies in a systems environment

Task 1: Draw a dependency graph for a university course registration system.

Task 2: Suggest modularization strategies to manage dependencies.

Deliverables: Dependency Graphs & a modularization strategy

Answers:

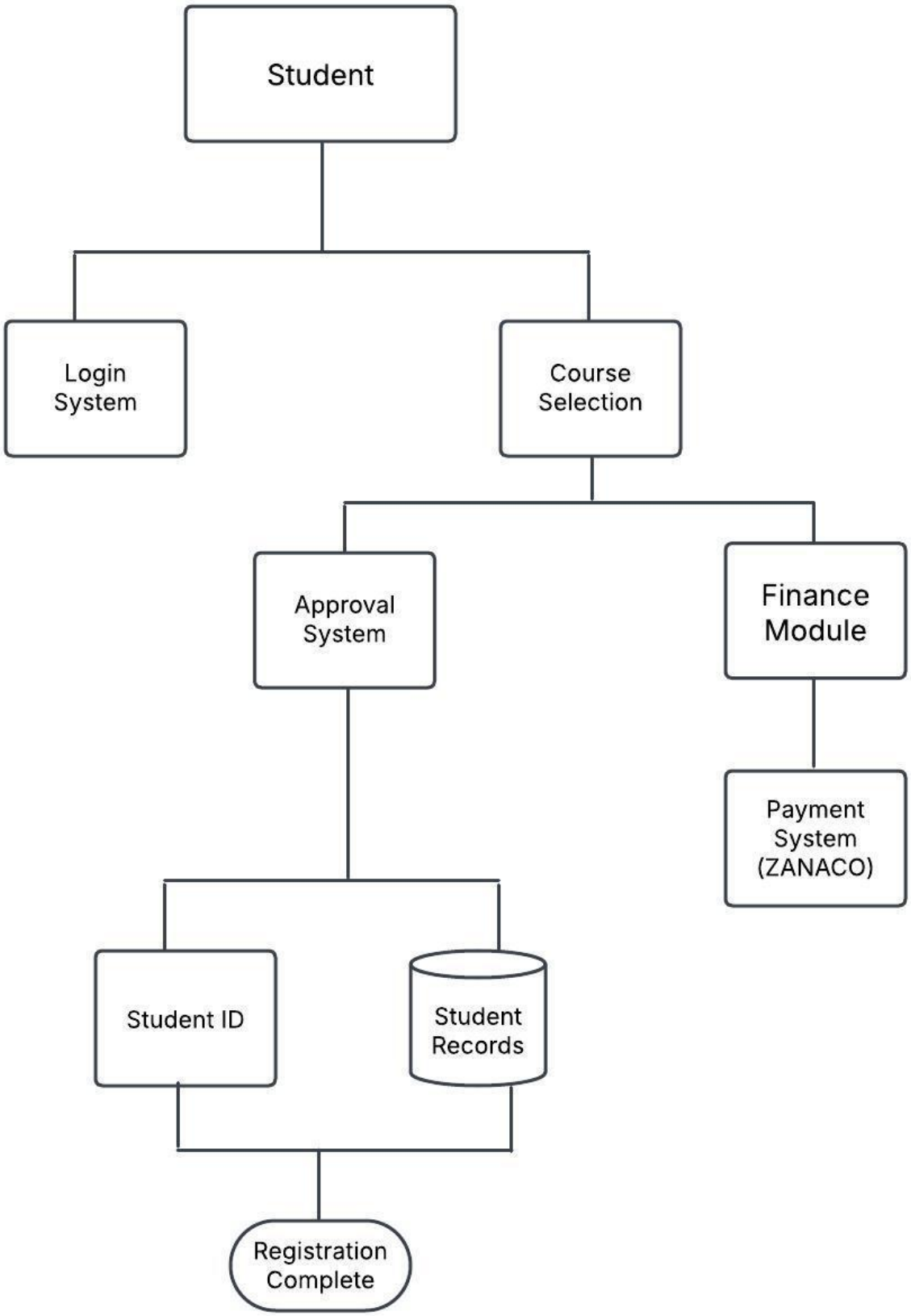
Task 1:

How It Works

Breakdown of Dependencies

1. Student logs in → via Login System (using Student ID and Computer Number).
2. Selects courses → from Course Selection Module.
3. Approval required → via HOD/Assistant Dean in Approval System.
4. Fees Checked → in Finance Module.
5. Payment Processed → through ZANACO Payment System.
6. Student ID Issued → after payment confirmation.
7. Record Kept → in Student Affairs Unit.
8. Registration Complete!

DEPENDENCY GRAPH FOR UNZA COURSE REGISTRATION SYSTEM


Student
Login System
Course Selection
Approval System
Finance Module
Payment System (ZANACO)
Student ID
Student Records
Registration Complete

Task 2:

Modularization Strategy

We divide the system into four core modules:

1. **User Authentication Module** → Manages login and security.
2. **Course Registration Module** → Handles course selection & approvals.
3. **Finance & Payment Module** → Manages tuition calculation & payments**.**
4. **Student Services Module** → Issues student ID & keeps student records.

-This reduces interdependencies, making the system scalable and easy to maintain.

Assessment 3: Requirement prioritization

Prioritize requirements for a project using different techniques

Task 1: Categorize requirements for a smart home system using the **MOSCOW** method

Task 2: Perform a Value vs Effort Analysis for the same system

Deliverables: Priority Requirements list & an analysis result

ANSWERS:

Task 1: Categorizing Requirements Using the MoSCoW Method

The MoSCoW method prioritizes requirements into four categories:

- **M (Must-Have)** – Essential for system functionality.
- **S (Should-Have)** – Important but not critical.
- **C (Could-Have)** – Nice to include if resources allow.
- **W (Won't-Have)** – Not a priority for now.

Smart Home System Requirements (MoSCoW Categorization)

Requirement	MoSCoW Category	Justification
Secure user authentication (biometric or PIN)	**Must-Have**	Ensures only authorized access
Remote control of lights & appliances	**Must-Have**	Core functionality of a smart home
Automated security alerts (intrusion detection)	**Must-Have**	Critical for safety
Mobile app integration	**Should-Have**	Improves usability but not essential
Voice command support (Alexa, Google Assistant)	**Should-Have**	Enhances convenience but can be added later
AI-based energy optimization	**Could-Have**	Reduces power consumption but requires complex implementation
Customizable automation routines	**Could-Have**	User-friendly customization, but not urgent
VR-based home simulation	**Won't-Have**	Unnecessary for now, costly to implement

---

Task 2: Value vs. Effort Analysis

The **Value vs. Effort Matrix** helps prioritize tasks by comparing:

- **Value (Impact on Users & System Goals)**
- **Effort (Complexity, Cost, and Resources Required)**

Value vs. Effort Grid for Smart Home System

Requirement	Value (High/Medium/Low)	Effort (High/Medium/Low)	Priority
Secure user authentication	High	Medium	✅ **High Priority**
Remote control of lights & appliances	High	Medium	✅ **High Priority**
Automated security alerts	High	High	✅ **High Priority**
Mobile app integration	Medium	Medium	⚡ **Medium Priority**

Requirement	Value (High/Medium/Low)	Effort (High/Medium/Low)	Priority
Voice command support	Medium	High	⚡ **Medium Priority**
AI-based energy optimization	High	High	🚀 **Low Priority (Future Implementation)**
Customizable automation routines	Medium	Medium	⚡ **Medium Priority**
VR-based home simulation	Low	High	❌ **Not a Priority**
