

1. System Development Life Cycle:

(1) Identifying Problems, opportunities, and objects

This phase primarily focuses on identifying and analyzing the relevant stakeholders, understanding their concerns, exploring opportunities for system improvement, and clarifying the objectives for updating the system.

(2) Determining human information requirements

Key activities include conducting surveys and interviews to gather feedback from teachers and students, designing prototypes for system updates, writing pseudocode, and evaluating whether the existing system's issues have been addressed.

(3) Analysis system needs

Key activities include creating data flow diagrams to analyze the requirements that the updated registration system must address, as well as identifying improvements to the system's architecture.

(4) Designing the recommender system

Key activities include designing the database schema for the new system, such as how to manage payment information and course details for teachers and students.

(5) Developing and documenting Software

Key activities include writing program and user manuals to explain the new system's features to teachers and students, such as the integration of a payment system and improvements to the course registration process; developing the program code for the updated system and creating demos.

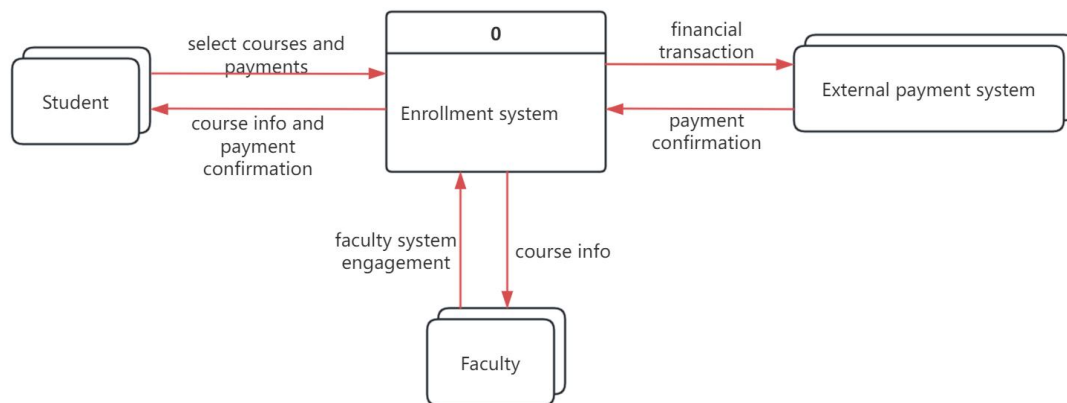
(6) Testing and managing the system

Key activities include documenting the system and code throughout the development cycle, as well as testing the performance of the updated system to minimize maintenance costs before release.

(7) Implementing and evaluating the system

Key activities include conducting comprehensive analysis and overall testing of the updated system, inviting teachers and students to participate in testing, and deciding how to replace the old system—whether to fully replace it or run the new system alongside the existing one.

2. context-level data flow diagram



3. Problem definition

Problem Definition

The current university enrollment system has the following main problems:

- Confusion and delays occur during course selection and registration for students and staff due to poor communication of information;
- Some students are temporary workers, making scheduling and resource management complex and affecting course arrangements;
- It is difficult to notify changes in courses and event adjustments in a timely manner, affecting user experience;
- Suppliers have poor management of event supplies, resulting in frequent delays and misdelivery of materials;
- Lack of efficient communication between departments leads to chaotic facility management;

- The system lacks historical data accumulation, making it difficult to carry out effective statistics and analysis.

Issues

- Customers (students) have a large number of random questions, affecting normal processes and management. (Weight: 10)
- Temporary student working hours lead to scheduling chaos and errors. (Weight: 9)
- The system is difficult to flexibly adapt to temporary course changes or event adjustments. (Weight: 7)
- Incomplete material supply management, with multiple shipments and untimely delivery of goods. (Weight: 6)
- Frequent communication barriers between different departments' staff, affecting event facility management. (Weight: 5)
- Lack of historical information accumulation, unable to effectively collect and analyze customer and resource information. (Weight: 3)

Objectives

- Provide a web-based system for students to conveniently query prices and place orders;
- Create or purchase a human resources system with a scheduling component to facilitate time management;
- Allow clients to sign event contracts online and provide web access accounts, supporting client-initiated updates for event changes;
- Provide tools to accurately determine the supply of materials during events;
- Establish an efficient communication system to facilitate timely communication with staff from various departments;
- Store all event data for easy aggregation and information sharing.

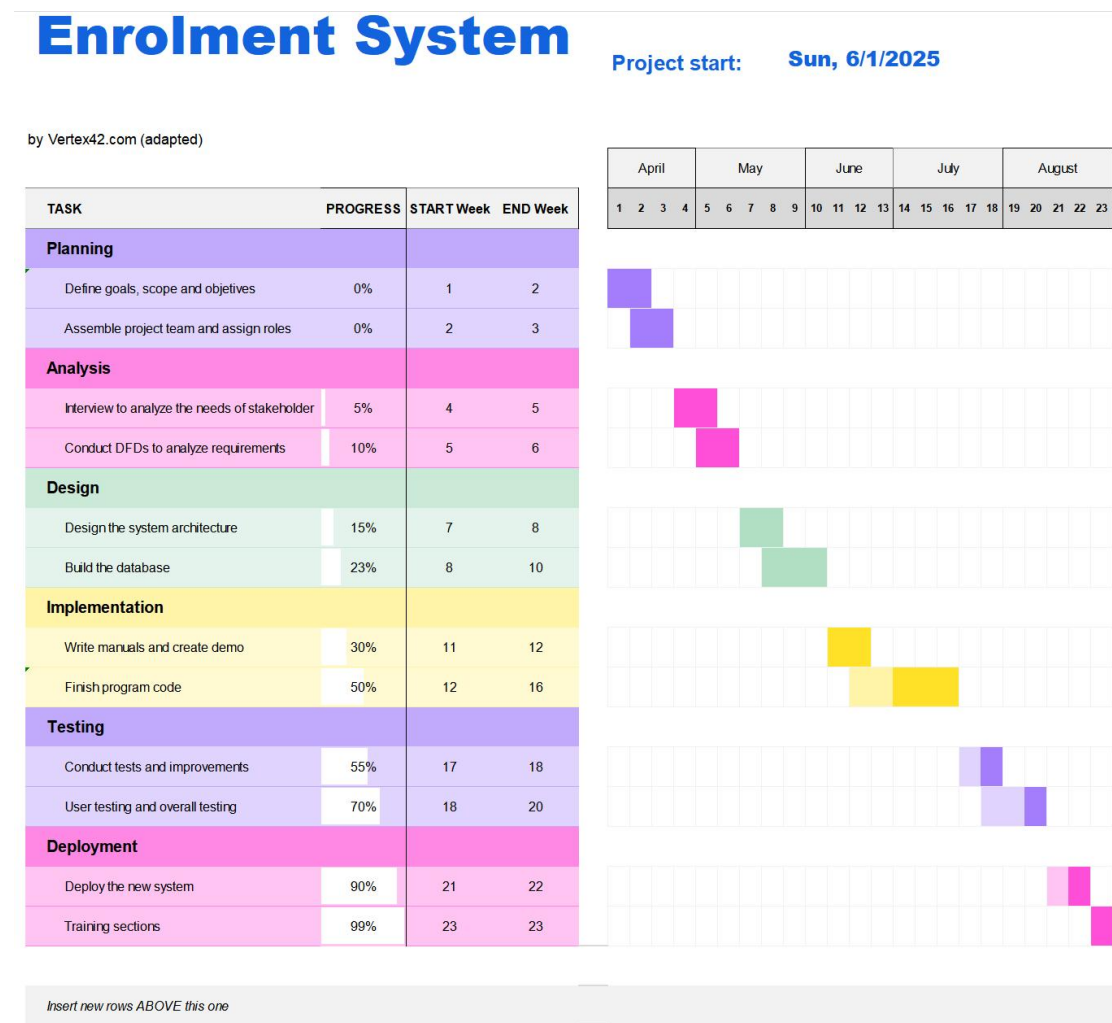
Requirements

- The system must be secure and reliable to prevent data leakage;
- Event managers must be able to input and manage event data;
- Event coordinators must be able to modify their contact information;
- The system must be easy to use and accessible to non-technical users.

Constraints

- The development budget must not exceed 50,000;
- The customer-facing website must go live by March 1 to meet the booking needs for graduation parties and weddings.

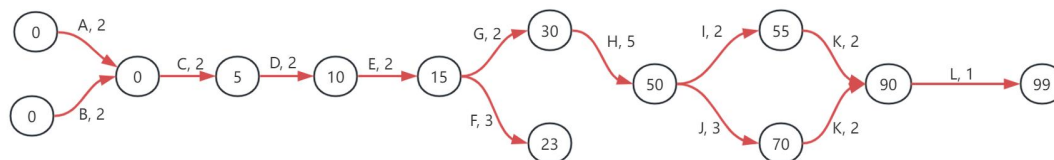
4. Gantt chart



Insert new rows ABOVE this one

5. PERT diagram

Activity	Description	Predecessor	Duration(weeks)
A	Define goals, scope and objectives	None	2
B	Assemble project team and assign roles	None	2
C	Interview to analyze the needs of stakeholder	A, B	2
D	Conduct DFDs to analyze requirements	A, B, C	2
E	Design the system architecture	C, D	2
F	Build the database	D, E	3
G	Write manuals and create demo	E	2
H	Finish program code	G	5
I	Conduct tests and improvements	H, G	2
J	User testing and overall testing	H, G	3
K	Deploy the new system	H, I, J	2
L	Training sections	K	1



Critical Path:

(A, 2 weeks) → (C, 2 weeks) → (D, 2 weeks) → (E, 2 weeks) → (G, 2 weeks) → (H, 5 weeks) → (J, 3 weeks) → (K, 2 weeks) → (L, 1 week)

Total duration: 21 weeks

Impact of Delays

Delays in tasks on the critical path will directly delay the overall project completion time. For example, if task H (5 weeks) is delayed by 1 week, the entire project will be delayed by 1 week.

Delays in tasks off the critical path will not affect the overall timeline as long as they do not exceed their slack time. For instance, delays in tasks F or I will not impact the total project duration if managed within their available float.

Any delay on the critical path must be managed carefully to avoid postponing the project delivery date.