A University is proposing to build a web based system where students and lecturers can create their profiles and the student accounts are verified by their lecturers.

[Modules]
LoginModule -> LoginController, StudentController, LecturerController
UserRoleModule -> UserRoleController

[Database table]
Need login systems with multiple users and roles. [students, lecturers]
Need approval from lecturer users. [user_roles]

The students can pay fees using this web system.

[Modules]
PaymentModule -> PaymentController

[Database table]
Support multiple payment gateways. Each gateways may has multiple methods to accept payment.
Need to cater multiple payment gateways. [paymentGateways, paymentMethods]

The students can buy and sell merchandise within the campus by posting their products on their profiles.

Each student will get their own customized profiles where they can post their products.

The transactions will be directly debited to the accounts of sellers.

The sellers should easily be able to manage the inventory and shipping of the products.

[Modules]
StoreModule -> StoreController (handle all personalized store setups)
CartModule -> CartController (handle customer checkout)
CheckoutModule -> CheckoutController
ShipmentModule -> ShipmentController

[Database table]
Student should have access to a dashboard to setup their store (for selling)

- *Setup/Update profile*
- *Add product to sell*
- *Update product availability*

- *View orders, Fulfill/Decline an order.*

Site visitors/Customers can see each studentStore.
Customer can add item to their cart.
Customer can checkout the item, choose shipment method, fill in billing & shipping details, and choose payment method to checkout.
Support for multiple shipment vendors, each should have different shipment methods with different charges.

[students, stores, products, customers, cart, orders, order_billing, order_shipping, shipment_vendors, shipment_methods]

Each student is linked with different lecturer

[Modules]
UserRelationModule -> UserRelationController

[Database table]
[students, lecturers, user_relations]

Students can help other students in their tasks for a customized fee.

[Modules]
OrderAssistModule -> OrderAssistController
File an request to assist in task and each task should comes with an estimated charges.
Accept the task and when marked as completed, deduct the charges from corresponding students/sellers account.

[Database table]
[students, orders, orderAssist]

Lecturers need to sign up for their accounts also, but the accounts will be verified physically by the system owners by sending them a meeting email after the signup.

[Modules]
BackendModule -> BackendController
An operational backend users to manage lecturers and students at higher level.

[Database table]
[administrators, user_approvals]

Students can make an appointment with their lecturer. The system will automatically show them the available times.

The lecturers will have their own profiles including their classes, expertise and their qualifications. The students should be able to interact will lecturers by reacting to the activities of the lecturers.

At the end of the day the system will automatically send email to the students and lecturers about the activities.

[Modules]
EventModule -> EventsController
NotificationModule -> NotificationController (manage email notifications)
Schedule an event with users in the system (student -> student, student -> lecturers, lecturers -> student, student -> customers)

Notifications module to handle reminders, alerts and communications between users regarding to certain event.

[Database table]
[events, notifications]