**Advising System (AS)**

**Team Plan**

**Team 2**

**System Requirements**

**1. Actors/End-Users:**

Advisor

Advisee (MBA student)

Registrar

Admin

Prospective Students

**2. High-Level Features:**

2.1 Create Long-Term Schedule (18 mo, 2 yr):

Advisors can generate a long-term schedule for advisees, considering both mandatory and elective courses over 18 months or 2 years.

2.2 Update Class Information:

Advisors and admin can update details of existing courses, including schedules, prerequisites, and available seats.

2.3 Add Class:

Advisors and admin can add new courses to the system, specifying details such as course code, name, description, prerequisites, and available seats.

2.4 Delete Class:

Advisors and admin can remove a course from the system, updating all related records and notifying affected advisees.

2.5 Show Class Info (All, By Semester):

Advisees can view information on all available courses or filter by semester, displaying details like course code, name, schedule, and available seats.

2.6 Degree Evaluation:

Advisees can request a degree evaluation to track their progress, ensuring they meet all MBA requirements.

2.7 List Remaining Requirements:

Advisees can view a list of remaining courses and other requirements needed to complete their MBA degree.

2.8 List/Evaluate Prerequisites:

Advisees can check and evaluate prerequisites for a specific course, helping them plan their schedule efficiently.

**3. Additional Considerations:**

3.1 User Authentication:

Secure login for all users (advisor, advisee, registrar, admin) with appropriate role-based access controls.

3.2 Notifications:

The system sends notifications to advisees for important events such as registration deadlines, class updates, and degree evaluation results.

3.3 Reporting:

Admin and advisors can generate reports on various aspects, including student progress, course popularity, and system usage statistics.

**4. Technical Requirements:**

4.1 Platform:

Web-based applications are accessible through standard browsers.

4.2 Database:

Robust database to store and manage course information, student records, and system logs.

4.3 Security:

Implement secure protocols for data transmission and storage. Regular security audits and updates.

**Dependencies:**

**User Authentication Setup:**

Dependency: User authentication must be implemented before any other feature can be accessed, ensuring secure and role-based access for advisors, advisees, registrar, and admin.

**Database Design and Implementation:**

Dependency: The database needs to be designed and implemented before features like adding, updating, and deleting classes can be developed. This ensures that the system has a reliable data storage mechanism.

**Create Long-Term Schedule:**

Dependency: Availability of course information in the system. Therefore, updating and adding classes should be completed before generating a long-term schedule.

**Update Class Information and Add Class:**

Dependency: Database functionality must be in place to support updating and adding classes. Additionally, user authentication needs to be set up for access control.

**Delete Class:**

Dependency: Similar to updating and adding classes, the ability to delete a class depends on having a functional database and appropriate user authentication.

**Show Class Info (All, By Semester):**

Dependency: Availability of course information in the database.

**Degree Evaluation:**

Dependency: Complete and up-to-date student records and course information.

**List Remaining Requirements and List/Evaluate Prerequisites:**

Dependency: Accurate and complete information about the student's completed courses, as well as the prerequisites for each course.

**Notifications:**

Dependency: User authentication, as notifications need to be sent to specific users. Additionally, the completion of relevant features like registration and degree evaluation is necessary for triggering notifications.

**Reporting:**

Dependency: Availability of data and completed functionality for generating reports.

**Platform Development:**

Dependency: Development of the web-based platform, including the user interface and necessary functionalities.

**Security Implementation:**

Dependency: Implementation of security measures should be completed before any features are deployed to ensure the protection of sensitive data.