**Unit Enrollment System Specification**

**Process Description**

The unit enrollment system automates the decision-making process for determining whether a student can enroll in a university course unit. This process evaluates multiple criteria to ensure that enrollments comply with academic policies, resource constraints, and student progress requirements. The system aims to streamline the enrollment workflow by implementing consistent rule application while accommodating special cases such as priority enrollments and late registrations.

The enrollment decision process consists of eight distinct validation checks:

1. **Prerequisite validation**: Ensures students have completed required prerequisite units with sufficient grades
2. **Capacity check**: Prevents enrollment when class capacity is exceeded
3. **Schedule conflict detection**: Identifies time clashes with already enrolled units
4. **Credit limit enforcement**: Ensures students don't exceed their maximum allowed credit points
5. **Academic standing verification**: Applies special rules for students on academic probation
6. **Late enrollment handling**: Manages enrollments after the official deadline
7. **Priority enrollment processing**: Gives enrollment priority to graduating students
8. **Final decision aggregation**: Combines all check results into a final enrollment decision

This model is based on common university enrollment policies observed at various institutions, including requirements for prerequisite completion, class capacity limitations, and special considerations for students nearing graduation.

**Data Structures**

**Student**

* id: String - Unique student identifier
* name: String - Student's full name
* hasCompleted: Boolean - Indicates if student has completed prerequisite unit
* currentCredits: Number - Current credits enrolled for the term
* maxAllowedCredits: Number - Maximum credits student can take per term
* isacademicStanding: String - Current academic standing (e.g., "Good Standing", "Probation")
* enrolledUnits: Array - List of units student is currently enrolled in
  + Each unit contains:
    - schedule: Object - Schedule information with days and times
* expectedGraduationTerm: String - Term when student is expected to graduate
* requiresUnitForGraduation: Boolean - Whether unit is required for graduation

**Unit**

* id: String - Unique unit identifier
* name: String - Unit name
* grade: Number - Student's grade in prerequisite unit
* currentEnrollment: Number - Current number of enrolled students
* maximumCapacity: Number - Maximum number of students allowed
* availableSpots: Number - Number of remaining available spots
* creditPoints: Number - Credit points awarded for completing the unit
* isRequiredCore: Boolean - Whether unit is a core requirement
* isHighDemand: Boolean - Whether unit is in high demand
* enrollmentDeadline: String - Deadline date for normal enrollment
* schedule: Object - Schedule information
  + days: Array - Days when unit is offered
  + time: String - Time slot in format "HH:MM-HH:MM"

**Response**

* status: String - Enrollment decision (e.g., "Approved", "Rejected", "Waiting List")
* summary: String - Explanation of the enrollment decision

**Business Rule Statements**

1. **Prerequisite Rules**:
   * If a student has completed the prerequisite unit with a grade of 50 or higher, they are allowed to proceed in the enrollment process.
   * If a student has completed the prerequisite unit with a grade below 50, their enrollment is denied.
   * If a student has not completed the prerequisite unit, their enrollment is denied.
2. **Class Capacity Rules**:
   * If the current enrollment is less than or equal to the maximum capacity and the prerequisite check is passed, the capacity check is successful.
   * If the current enrollment exceeds the maximum capacity and the prerequisite check is passed, the capacity check fails with message "Class is full".
   * If the prerequisite check fails, the capacity check returns the prerequisite check message.
3. **Schedule Conflict Rules**:
   * If a student attempts to enroll in a unit whose schedule overlaps with any of their already enrolled units (same days and overlapping times), a schedule conflict is detected.
   * A schedule overlap exists when: the units share at least one day AND the time ranges overlap.
4. **Credit Limit Rules**:
   * If enrolling in a new unit would cause a student to exceed their maximum allowed credits, the credit limit check fails.
   * If a student's total credits after enrollment would remain within their allowed limit, the credit limit check passes.
5. **Academic Standing Rules**:
   * Students on academic probation can only enroll in core required units without special approval.
   * Students on academic probation attempting to enroll in non-core units require advisor approval.
   * Students in good academic standing can enroll in any units without standing-based restrictions.
6. **Late Enrollment Rules**:
   * If the current date is after the unit's enrollment deadline, the enrollment is considered late.
   * Late enrollments require department head approval and incur a late fee.
   * Enrollments before the deadline proceed normally without additional approvals or fees.
7. **Priority Enrollment Rules**:
   * If a high-demand unit has 10 or fewer available spots, priority rules are activated.
   * Students who require the unit for graduation in the next term receive priority enrollment.
   * Students who don't meet priority criteria for high-demand units with limited spots are placed on a waiting list.
8. **Final Decision Rules**:
   * Enrollment is approved when all conditions are satisfied: prerequisites met, class not full, no schedule conflicts, within credit limit, good academic standing (or core unit for probation students), and before deadline.
   * Enrollment is set for review when prerequisites are met and credit limit is within bounds, but other conditions fail.
   * Students are placed on a waiting list for high-demand units with low availability if they don't qualify for priority enrollment.
   * All other cases result in rejected enrollment.

**Tests**

The following JSON test cases will be used to validate the decision model:

**Test Case 1: Successful Enrollment**

{

"student": {

"id": "S12345",

"name": "John Smith",

"hasCompleted": true,

"currentCredits": 12,

"maxAllowedCredits": 24,

"isacademicStanding": "Good Standing",

"enrolledUnits": [

{

"schedule": {

"days": ["Monday", "Wednesday"],

"time": "10:00-12:00"

}

}

],

"expectedGraduationTerm": "2026-1",

"requiresUnitForGraduation": false

},

"unit": {

"id": "COMP101",

"name": "Introduction to Programming",

"grade": 75,

"currentEnrollment": 40,

"maximumCapacity": 60,

"availableSpots": 20,

"creditPoints": 6,

"isRequiredCore": true,

"isHighDemand": false,

"enrollmentDeadline": "2025-05-15",

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "14:00-16:00"

}

},

"currentTerm": "2025-1"

}

**Test Case 2: Failed Prerequisite**

{

"student": {

"id": "S67890",

"name": "Emma Brown",

"hasCompleted": true,

"currentCredits": 18,

"maxAllowedCredits": 24,

"isacademicStanding": "Good Standing",

"enrolledUnits": [

{

"schedule": {

"days": ["Monday", "Wednesday"],

"time": "13:00-15:00"

}

}

],

"expectedGraduationTerm": "2026-1",

"requiresUnitForGraduation": false

},

"unit": {

"id": "COMP201",

"name": "Data Structures",

"grade": 45,

"currentEnrollment": 35,

"maximumCapacity": 50,

"availableSpots": 15,

"creditPoints": 6,

"isRequiredCore": false,

"isHighDemand": false,

"enrollmentDeadline": "2025-05-15",

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "09:00-11:00"

}

},

"currentTerm": "2025-1"

}

**Test Case 3: Class Full**

{

"student": {

"id": "S13579",

"name": "Mike Wilson",

"hasCompleted": true,

"currentCredits": 12,

"maxAllowedCredits": 24,

"isacademicStanding": "Good Standing",

"enrolledUnits": [

{

"schedule": {

"days": ["Monday", "Wednesday"],

"time": "14:00-16:00"

}

}

],

"expectedGraduationTerm": "2025-2",

"requiresUnitForGraduation": false

},

"unit": {

"id": "COMP301",

"name": "Algorithm Design",

"grade": 65,

"currentEnrollment": 50,

"maximumCapacity": 50,

"availableSpots": 0,

"creditPoints": 6,

"isRequiredCore": false,

"isHighDemand": true,

"enrollmentDeadline": "2025-05-15",

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "10:00-12:00"

}

},

"currentTerm": "2025-1"

}

**Test Case 4: Schedule Conflict**

{

"student": {

"id": "S24680",

"name": "Sarah Johnson",

"hasCompleted": true,

"currentCredits": 12,

"maxAllowedCredits": 24,

"isacademicStanding": "Good Standing",

"enrolledUnits": [

{

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "11:00-13:00"

}

}

],

"expectedGraduationTerm": "2026-1",

"requiresUnitForGraduation": false

},

"unit": {

"id": "COMP401",

"name": "Software Engineering",

"grade": 80,

"currentEnrollment": 40,

"maximumCapacity": 60,

"availableSpots": 20,

"creditPoints": 6,

"isRequiredCore": true,

"isHighDemand": false,

"enrollmentDeadline": "2025-05-15",

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "12:00-14:00"

}

},

"currentTerm": "2025-1"

}

**Test Case 5: Credit Limit Exceeded**

{

"student": {

"id": "S97531",

"name": "David Lee",

"hasCompleted": true,

"currentCredits": 21,

"maxAllowedCredits": 24,

"isacademicStanding": "Good Standing",

"enrolledUnits": [

{

"schedule": {

"days": ["Monday", "Wednesday"],

"time": "09:00-11:00"

}

}

],

"expectedGraduationTerm": "2026-1",

"requiresUnitForGraduation": false

},

"unit": {

"id": "COMP501",

"name": "Artificial Intelligence",

"grade": 85,

"currentEnrollment": 30,

"maximumCapacity": 40,

"availableSpots": 10,

"creditPoints": 6,

"isRequiredCore": false,

"isHighDemand": true,

"enrollmentDeadline": "2025-05-15",

"schedule": {

"days": ["Friday"],

"time": "10:00-13:00"

}

},

"currentTerm": "2025-1"

}

**Test Case 6: Academic Probation for Non-Core Unit**

{

"student": {

"id": "S86420",

"name": "Lisa Chen",

"hasCompleted": true,

"currentCredits": 12,

"maxAllowedCredits": 18,

"isacademicStanding": "Probation",

"enrolledUnits": [

{

"schedule": {

"days": ["Monday", "Wednesday"],

"time": "14:00-16:00"

}

}

],

"expectedGraduationTerm": "2026-1",

"requiresUnitForGraduation": false

},

"unit": {

"id": "COMP601",

"name": "Machine Learning",

"grade": 70,

"currentEnrollment": 35,

"maximumCapacity": 40,

"availableSpots": 5,

"creditPoints": 6,

"isRequiredCore": false,

"isHighDemand": true,

"enrollmentDeadline": "2025-05-15",

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "15:00-17:00"

}

},

"currentTerm": "2025-1"

}

**Test Case 7: Late Enrollment**

{

"student": {

"id": "S11223",

"name": "Alex Taylor",

"hasCompleted": true,

"currentCredits": 12,

"maxAllowedCredits": 24,

"isacademicStanding": "Good Standing",

"enrolledUnits": [

{

"schedule": {

"days": ["Monday", "Wednesday"],

"time": "13:00-15:00"

}

}

],

"expectedGraduationTerm": "2026-1",

"requiresUnitForGraduation": false

},

"unit": {

"id": "COMP701",

"name": "Computer Networks",

"grade": 75,

"currentEnrollment": 30,

"maximumCapacity": 40,

"availableSpots": 10,

"creditPoints": 6,

"isRequiredCore": true,

"isHighDemand": false,

"enrollmentDeadline": "2025-02-28",

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "10:00-12:00"

}

},

"currentTerm": "2025-1"

}

**Test Case 8: Priority Enrollment for Graduating Student**

{

"student": {

"id": "S99887",

"name": "Ryan Martinez",

"hasCompleted": true,

"currentCredits": 18,

"maxAllowedCredits": 24,

"isacademicStanding": "Good Standing",

"enrolledUnits": [

{

"schedule": {

"days": ["Monday", "Wednesday"],

"time": "09:00-11:00"

}

}

],

"expectedGraduationTerm": "2025-2",

"requiresUnitForGraduation": true

},

"unit": {

"id": "COMP801",

"name": "Cybersecurity",

"grade": 85,

"currentEnrollment": 41,

"maximumCapacity": 50,

"availableSpots": 9,

"creditPoints": 6,

"isRequiredCore": true,

"isHighDemand": true,

"enrollmentDeadline": "2025-05-15",

"schedule": {

"days": ["Tuesday", "Thursday"],

"time": "16:00-18:00"

}

},

"currentTerm": "2025-1"

}