

جامعة اللاذقية

كلية الهندسة المعلوماتية

قسم البرمجيات ونظم المعلومات

تحليل النظم المالية والادارية

# (SRS) وثيقة المتطلبات البرمجية

اسم النظام:

نظام إدارة شؤون الطلاب والامتحانات - كلية الهندسة المعلوماتية

التاريخ:

حزيران 2025 27

إعداد الطلاب:

أحمد نواف مخبير

نغم طلال إسماعيل

مرام هيثم عباس

جوليا وائل الخطيب

## 1. المقدمة

### 1.1 الهدف من الوثيقة

تهدف هذه الوثيقة إلى تحديد المتطلبات البرمجية لنظام إدارة شؤون الطلاب والامتحانات، الذي يخدم كلية الهندسة المعلوماتية، ويُستخدم من قبل الطلاب، الإداريين، وأعضاء الهيئة التدريسية. توفر هذه الوثيقة وصفاً شاملاً لوظائف النظام، خصائصه، وقيوده، بالإضافة إلى المخططات التصميمية الأساسية التي توضح هيكلته وسلوكه.

### 1.2 نطاق النظام

يتضمن النظام الوظائف الرئيسية التالية:

- إدارة بيانات الطلاب: تسجيل، تعديل، وتتبع حالة الطلاب.
- تسجيل المقررات: السماح للطلاب بتسجيل المقررات الدراسية والتحقق من المتطلبات المسبقة.
- تنظيم الجداول الدراسية والامتحانات: إنشاء وتوزيع جداول المحاضرات والامتحانات.

- إدخال وإدارة العلامات :تمكين المدرّسين من إدخال العلامات وتدقيقها من قبل الشؤون الامتحانية.
- إصدار الوثائق الأكاديمية :طباعة كشوف العلامات، وثائق القيد، ومصدقات التخرج.
- استخراج التقارير الإحصائية :توليد تقارير عن أداء الطلاب وإحصائيات النجاح والرسوب.

### 1.3 تعريفات

- **SRS** (Software Requirements Specification) وثيقة متطلبات البرمجية.
  - **الطالب** :المستخدم المستفيد من الخدمات الأكاديمية المقدمة من النظام.
  - **الإداري** :موظف مسؤول عن تحديث البيانات، معالجة الطلبات، وإعداد الجداول.
  - **المدرّس** :عضو هيئة تدريس مسؤول عن التعليم وإدخال العلامات.
  - **مدير النظام** :مسؤول عن إدارة المستخدمين، الصلاحيات، والنسخ الاحتياطية للنظام.
- 

## 2. وصف عام للنظام

### 2.1 منظور النظام

النظام عبارة عن تطبيق ويب/موبايل يربط بين قاعدة بيانات مركزية وواجهة مستخدم تفاعلية. يتميز النظام بصلاحيات مختلفة لكل نوع من المستخدمين (طالب، مدرّس، إداري، مدير نظام) لضمان أمان البيانات والوصول المحدود للوظائف حسب الدور.

### 2.2 وظائف النظام الرئيسية

- **تسجيل الطلاب والمقررات** :إدارة عملية تسجيل الطلاب الجدد والمقررات الدراسية.
- **حفظ ومتابعة العلامات والنتائج** :تتبع أداء الطلاب الأكاديمي.
- **تنظيم جداول التدريس والامتحانات** :جدولة المحاضرات والامتحانات بكفاءة.
- **طباعة الوثائق الرسمية** :توفير إمكانية إصدار الوثائق الأكاديمية الهامة.
- **توليد تقارير إدارة وتحليل أداء الطلاب** :تقديم رؤى إحصائية لدعم اتخاذ القرار.

### 2.3 خصائص المستخدمين

- **طالب** :يمكنه تسجيل المقررات، عرض العلامات، وتحميل الوثائق الأكاديمية.
- **مدرّس** :يمكنه إدخال العلامات وعرض قوائم الطلاب المسجلين في مقرراته.

- إداري: يمكنه تعديل بيانات الطلاب، إصدار الوثائق، وإعداد الجداول الدراسية والامتحانية.
- مدير النظام: يمتلك صلاحيات كاملة لإدارة المستخدمين، تعيين الصلاحيات، وإدارة النسخ الاحتياطية للنظام.

## 2.4 قيود النظام

- يعمل النظام ضمن بيئة شبكة الإنترنت.
- يعتمد على قاعدة بيانات مركزية لتخزين جميع البيانات.
- واجهة المستخدم الرئيسية للنظام ستكون باللغة العربية.

---

## 3. المتطلبات الوظيفية (Functional Requirements)

### FR1. إدارة حسابات المستخدمين

- تسجيل الدخول/الخروج الآمن.
- صلاحيات مختلفة لكل من (طالب، مدرس، إداري، مدير نظام).

### FR2. إدارة بيانات الطلاب

- إدخال وتعديل بيانات الطلاب الشخصية والأكاديمية.
- تتبع حالة القيد للطلاب (فعال، متوقف، متخرج).
- أرشفة بيانات الخريجين.

### FR3. التسجيل الأكاديمي

- تسجيل المقررات بناءً على الخطة الدراسية المعتمدة.
- التحقق التلقائي من المتطلبات المسبقة للمقررات.

### FR4. إدارة المقررات

- إضافة وتعديل بيانات المقررات الدراسية.
- ربط المقررات بالمدرسين وعدد الساعات المعتمدة.

### FR5. إدارة الجداول

- إنشاء جداول المحاضرات الدراسية.
- توزيع القاعات الدراسية والمدرسين على الجداول.

### FR6. تنظيم الامتحانات

- إعداد جداول الامتحانات النهائية والفصلية.

- توزيع القاعات الامتحانية والمراقبين.
- تسجيل حالات الغياب والمخالفات أثناء الامتحانات.

#### FR7. إدخال وإدارة العلامات

- إدخال علامات أعمال الفصل والامتحان النهائي.
- تدقيق النتائج من قبل الشؤون الامتحانية قبل الاعتماد.

#### FR8. إصدار الوثائق الأكاديمية

- إصدار كشف علامات رسمي.
- إصدار وثيقة قيد طالب.
- إصدار مصدقة تخرج.

#### FR9. التقارير الإدارية

- توليد إحصائيات عن نسب النجاح والرسوب.
- تقارير عن عدد الطلاب، التحصيل الأكاديمي، والمعدلات التراكمية.

### 4. المتطلبات غير الوظيفية (Non-Functional Requirements)

| الوصف  | المتطلب          | الرمز |
|--|------------------|-------|
| التحقق من الهوية، صلاحيات متعددة، تشفير البيانات الحساسة.  | الأمان           | NFR1  |
| سرعة الاستجابة يجب أن تكون أقل من 3 ثوانٍ لمعظم العمليات.  | الأداء           | NFR2  |
| 99% متاح 24/7 بنسبة توافر تزيد عن 99.                      | التوافر          | NFR3  |
| واجهة مستخدم عربية سهلة وبديهية للطالب والمدرس والإداري.   | سهولة الاستخدام  | NFR4  |
| هيكلية واضحة للكود، سهولة تحديث النظام وإضافة ميزات جديدة. | القابلية للصيانة | NFR5  |
| نسخ يومي للبيانات مع إمكانية الاسترجاع السريع عند الطوارئ. | النسخ الاحتياطي  | NFR6  |
| دعم جميع المتصفحات الحديثة وأنظمة التشغيل الشائعة.         | التوافق          | NFR7  |

## 5. المتطلبات المستقبلية (Future Enhancements)

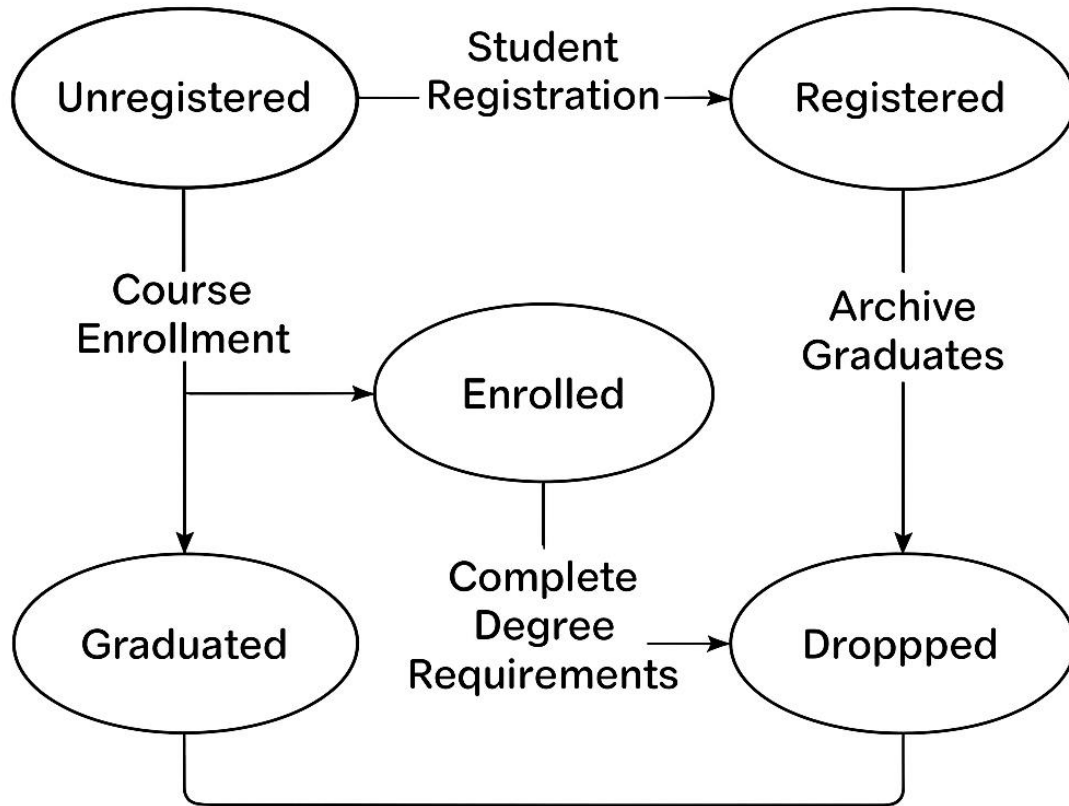
- للطلاب والمدرّسين (SMS/Email) نظام تنبيهات ذكي.
  - تكامل مع بوابة الجامعة المركزية.
  - (iOS أندرويد و) دعم تطبيقات الموبايل الأصلية.
  - لتحليل البيانات (Dashboard) تقارير رسومية تفاعلية.
- 

## 6. المخططات التصميمية

### 6.1 مخطط الحالة (State Diagram)

يوضح مخطط الحالة دورة حياة الطالب داخل النظام، من التسجيل وحتى التخرج أو الانسحاب.

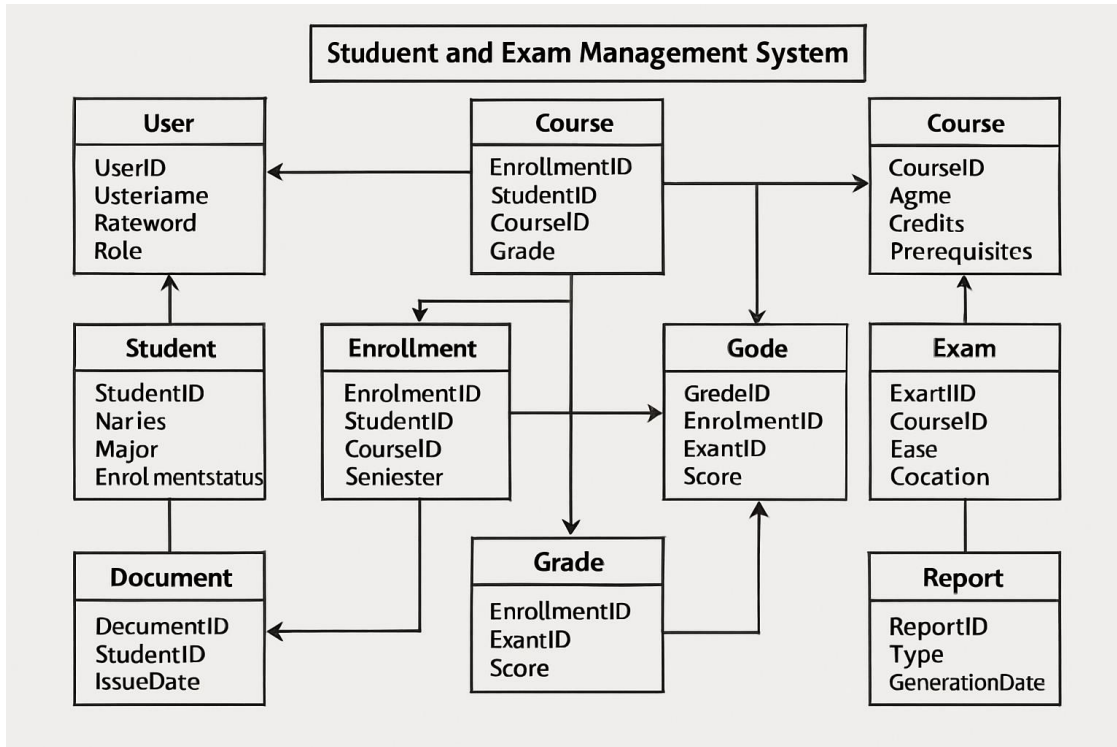
# Student and Exam Management System



*State Diagram*

## 6.2 مخطط الصفوف (Class Diagram)

يمثل مخطط الصفوف الهيكل المنطقي للنظام، موضحًا الكيانات الرئيسية وعلاقاتها.

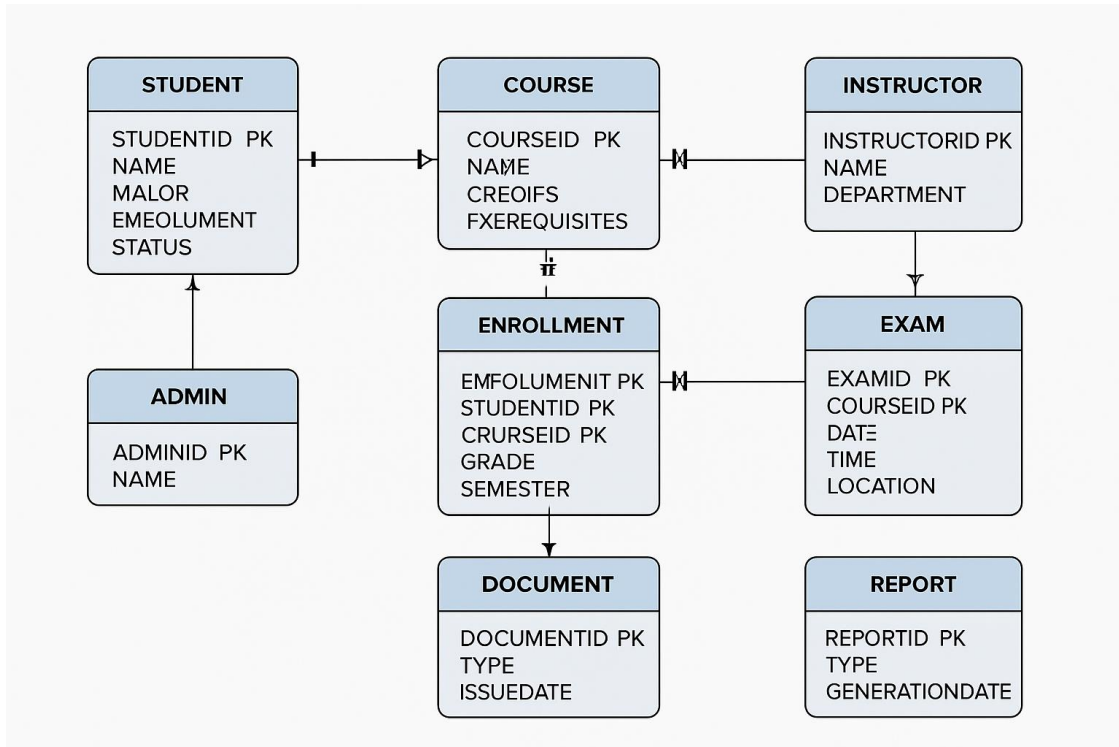


*Class Diagram*

### 6.3 مخطط ERD (Entity Relationship Diagram)

هيكل قاعدة البيانات، موضحًا الكيانات والعلاقات بينها ERD يصف مخطط





ERD Diagram

## 7. الكود الرئيسي لواجهة برمجة التطبيقات (API)

Django REST تم تطوير واجهة برمجة تطبيقات بسيطة باستخدام إطار عمل المحولات (Models)، لإدارة بيانات الطلاب يتضمن الكود النماذج (Serializers)، طرق العرض (Views)، ومسارات URL.

### 7.1 students/models.py (نموذج الطالب)

```
from django.db import models
```

```
class Student(models.Model):
    name = models.CharField(max_length=100)
    major = models.CharField(max_length=100)
    enrollment_status = models.CharField(max_length=50)

    def __str__(self):
        return self.name
```

## 7.2 students/serializers.py (محول الطالب)

```
from rest_framework import serializers
from .models import Student

class StudentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Student
        fields = ["id", "name", "major",
"enrollment_status"]
```

## 7.3 students/views.py (طرق عرض الطالب)

```
from rest_framework import viewsets
from .models import Student
from .serializers import StudentSerializer

class StudentViewSet(viewsets.ModelViewSet):
    queryset = Student.objects.all()
    serializer_class = StudentSerializer
```

## 7.4 students/urls.py (لتطبيق الطلاب URL مسارات)

```
from rest_framework.routers import DefaultRouter
from .views import StudentViewSet
```

```
router = DefaultRouter()
router.register(r'students', StudentViewSet)
```

```
urlpatterns = router.urls
```

## 7.5 student\_management\_system/urls.py (URL مسارات للمشروع الرئيسي)

```
from django.contrib import admin
from django.urls import path, include
from students.urls import router
```

```
urlpatterns = [
```

```
    path("admin/", admin.site.urls),  
    path("api/", include(router.urls)),  
]
```