

# Software Requirement Specification (SRS)

## Smart Student Scheduler

### المقدمة

#### 1. الهدف من الوثيقة

تصف هذه الوثيقة جميع المتطلبات الخاصة بنظام Smart Student Scheduler ، وهو تطبيق يساعد الطلاب على تنظيم وقتهم وجدولة المهام الدراسية تلقائيًا.

#### 2. نطاق النظام

يهدف النظام إلى مساعدة الطلاب على إنشاء جداول دراسية ذكية بناءً على المحاضرات الثابتة، أوقات الفراغ، والمهام الأكاديمية المختلفة.

#### 3. الجمهور المستهدف

الطلاب الجامعيون، محللو النظم، المطورون، والمشرفون على المشروع.

### 2. الوصف العام

#### 1. نظرة عامة على النظام

Smart Student Scheduler هو نظام ذكي يهدف إلى تنظيم وقت الطالب من خلال توزيع المهام على أوقات الفراغ بطريقة متوازنة.

#### 2. وظائف النظام الرئيسية

- تسجيل الدخول
- إدخال مواعيد المحاضرات
- إضافة المهام الدراسية
- توليد الجدول الذكي
- التنبيهات

#### 3. خصائص المستخدمين

الطالب هو المستخدم الأساسي للنظام. يجب أن يكون لديه حساب للدخول.

#### 4. القيود

النظام يحتاج إلى اتصال بالإنترنت في حال استخدام النسخة السحابية.

#### 5. الافتراضات والتبعيات

يفترض أن الطالب سيدخل بيانات دقيقة ليتمكن النظام من العمل بكفاءة.

### 3. المتطلبات الوظيفية

1. يجب أن يستطيع المستخدم تسجيل الدخول والخروج.
2. يمكن للمستخدم إدخال مواعيد المحاضرات.
3. يمكن للمستخدم تحديد أوقات الفراغ.
4. يمكن للمستخدم إدخال المهام الدراسية.
5. يقوم النظام بتوليد الجدول تلقائيًا.
6. يتيح النظام التعديل اليدوي للجدول.
7. يرسل النظام تنبيهات للمهام القادمة.
8. يدعم النظام تعدد المستخدمين.

### 4. المتطلبات غير الوظيفية

1. سهولة الاستخدام: واجهة بسيطة وسهلة.
2. الأداء: يولد الجدول خلال ثوانٍ.
3. الاستقرار: لا يتعطل النظام أثناء الاستخدام العادي.
4. الأمان: البيانات محمية ومشفرة.
5. قابلية التوسع: يمكن تطوير النظام لاحقًا.
6. دعم الأجهزة: متاح كتطبيق ويب وأندرويد.
7. دعم اللغة: يدعم العربية والإنجليزية.

### 5 حالات الاستخدام:

#### حالة الاستخدام 1: تسجيل طالب جديد

الاسم: تسجيل طالب جديد

وصف موجز: تمكّن هذه الحالة الطالب من إنشاء حساب جديد في النظام.

الشرط المسبق: لا يجب أن يكون الطالب مسجلًا مسبقًا في النظام.

الشرط اللاحق: يتم إنشاء حساب جديد للطالب ويتم تخزين بياناته في قاعدة البيانات.

حالات الخطأ:

البريد الإلكتروني مستخدم مسبقًا.

تنسيق البريد الإلكتروني غير صالح.

أحد الحقول المطلوبة مفقود.

حالة النظام عند حدوث خطأ: يتم عرض رسالة خطأ للمستخدم ولا يتم إنشاء الحساب.

الجهات الفاعلة: الطالب و مدير النظام

المحفّز: الطالب يضغط على زر "تسجيل".

**العملية القياسية:** يدخل الطالب اسمه، بريده الإلكتروني، وكلمة المرور.

يضغط زر التسجيل.

يتحقق النظام من صحة البيانات.

يتم إنشاء الحساب وتخزين البيانات.

تظهر رسالة "تم التسجيل بنجاح".

#### **العمليات البديلة:**

إذا كان البريد الإلكتروني موجودًا، يتم إعلام الطالب ولا يتم إنشاء الحساب.

إذا كانت البيانات ناقصة أو غير صحيحة، يُطلب من الطالب تصحيحها.

#### **حالة الاستخدام 2: تسجيل الدخول**

**الاسم:** تسجيل الدخول

**وصف موجز:** تسمح هذه الحالة للطالب بالدخول إلى النظام باستخدام بياناته المُسجلة.

**الشرط المسبق:** يجب أن يكون الطالب قد سجّل مسبقًا.

**الشرط اللاحق:** يتم التحقق من بيانات الدخول ويتم توجيه الطالب إلى لوحة التحكم الخاصة به.

#### **حالات الخطأ:**

البريد الإلكتروني أو كلمة المرور غير صحيحة.

أحد الحقول مفقود.

**حالة النظام عند حدوث خطأ:** لا يتم تسجيل الدخول وتُعرض رسالة خطأ.

**الجهات الفاعلة:** الطالب. و مدير النظام

**المحفّز:** الطالب يضغط على زر "تسجيل الدخول".

#### **العملية القياسية:**

يدخل الطالب بريده الإلكتروني وكلمة المرور.

يضغط زر تسجيل الدخول.

يتحقق النظام من صحة البيانات.

إذا كانت صحيحة، يتم الدخول وتحميل الصفحة الرئيسية.

**العمليات البديلة:** في حال خطأ في كلمة المرور، يظهر إشعار بالمشكلة.

إذا لم يكن الحساب موجودًا، يظهر اقتراح بالتسجيل أولاً.

### حالة الاستخدام 3: إدخال مواعيد المحاضرات

الاسم: إدخال مواعيد المحاضرات

وصف موجز: تتيح هذه الحالة للمستخدم إدخال تفاصيل مواعيد محاضراته ضمن الجدول.

الشرط المسبق: يجب أن يكون المستخدم مسجلاً ومُسجلاً الدخول إلى النظام.

الشرط اللاحق: يتم حفظ مواعيد المحاضرات بنجاح في قاعدة البيانات.

حالات الخطأ: إدخال بيانات غير مكتملة (مثل عدم تحديد اليوم أو الوقت).

تعارض الموعد مع موعد آخر مُدخل مسبقاً.

حالة النظام عند حدوث خطأ: يتم عرض رسالة توضح الخطأ، ولا يتم حفظ الموعد.

الجهات الفاعلة: المستخدم (الطالب) ومدير النظام

المحفز: المستخدم يضغط على زر "إضافة محاضرة".

العملية القياسية:

يدخل المستخدم اسم المادة، يوم المحاضرة، ووقتها.

يضغط على "حفظ".

يتحقق النظام من صحة البيانات وعدم وجود تعارض.

يتم حفظ البيانات في الجدول.

تظهر رسالة تأكيد.

العمليات البديلة:

في حال وجود تعارض في الوقت، يُطلب من المستخدم تعديل الموعد.

إذا كانت البيانات ناقصة، يتم تنبيهه لإكمالها.

### حالة الاستخدام 4: إضافة مهمة أكاديمية

الاسم: إضافة مهمة أكاديمية

وصف موجز: تتيح هذه الحالة للمستخدم إدخال مهمة (مثل واجب، امتحان، مشروع) مرتبطة بمادة معينة.

الشرط المسبق: يجب أن يكون المستخدم مسجلاً في النظام، ويكون قد أضاف مواد دراسية مسبقاً.

الشرط اللاحق: يتم حفظ المهمة الأكاديمية في قائمة المهام مع تاريخ استحقاقها.

حالات الخطأ: عدم اختيار المادة المرتبطة بالمهمة.

تحديد تاريخ غير صحيح (مثل تاريخ ماضي).

حالة النظام عند حدوث خطأ: يتم عرض رسالة خطأ ولا تُحفظ المهمة.

الجهات الفاعلة: المستخدم (الطالب) و مدير النظام

المحفز: المستخدم يضغط على زر "إضافة مهمة".

العملية القياسية:

يختار المستخدم اسم المادة.

يدخل عنوان المهمة وتاريخ الاستحقاق.

يضغط على "حفظ المهمة".

يتم التحقق من صحة البيانات.

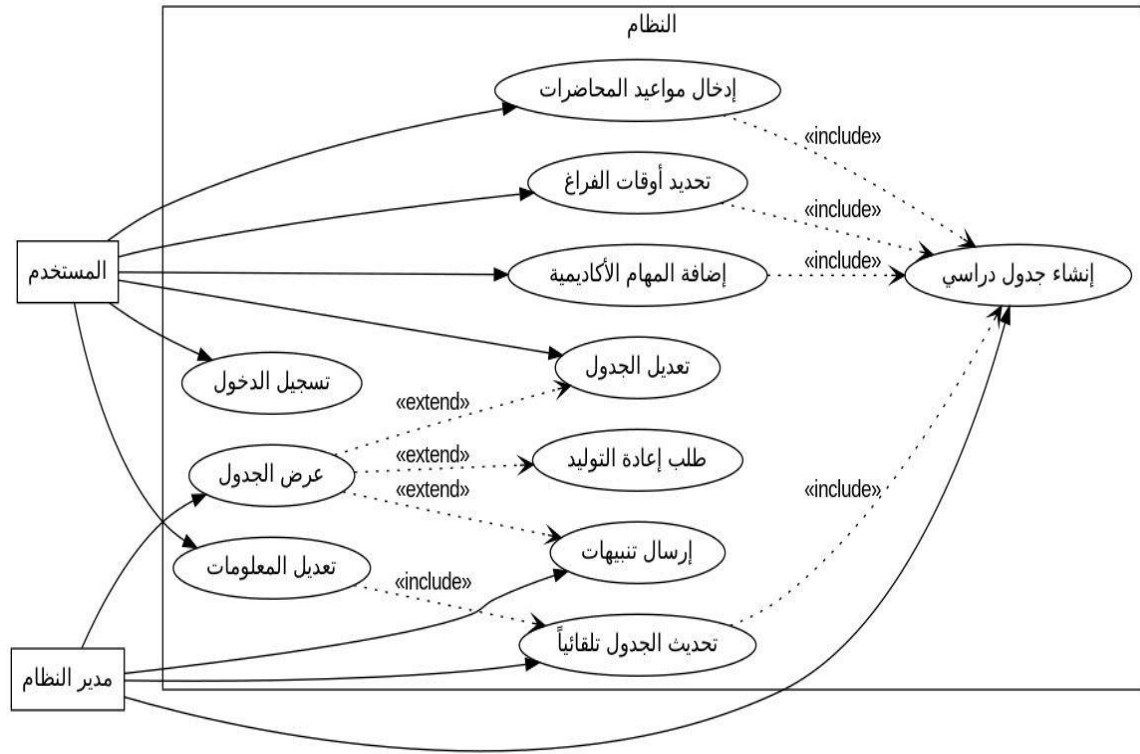
تُحفظ المهمة في قائمة المهام وتُعرض للمستخدم.

العمليات البديلة:

إذا كان التاريخ غير صالح، يُطلب من المستخدم تعديله.

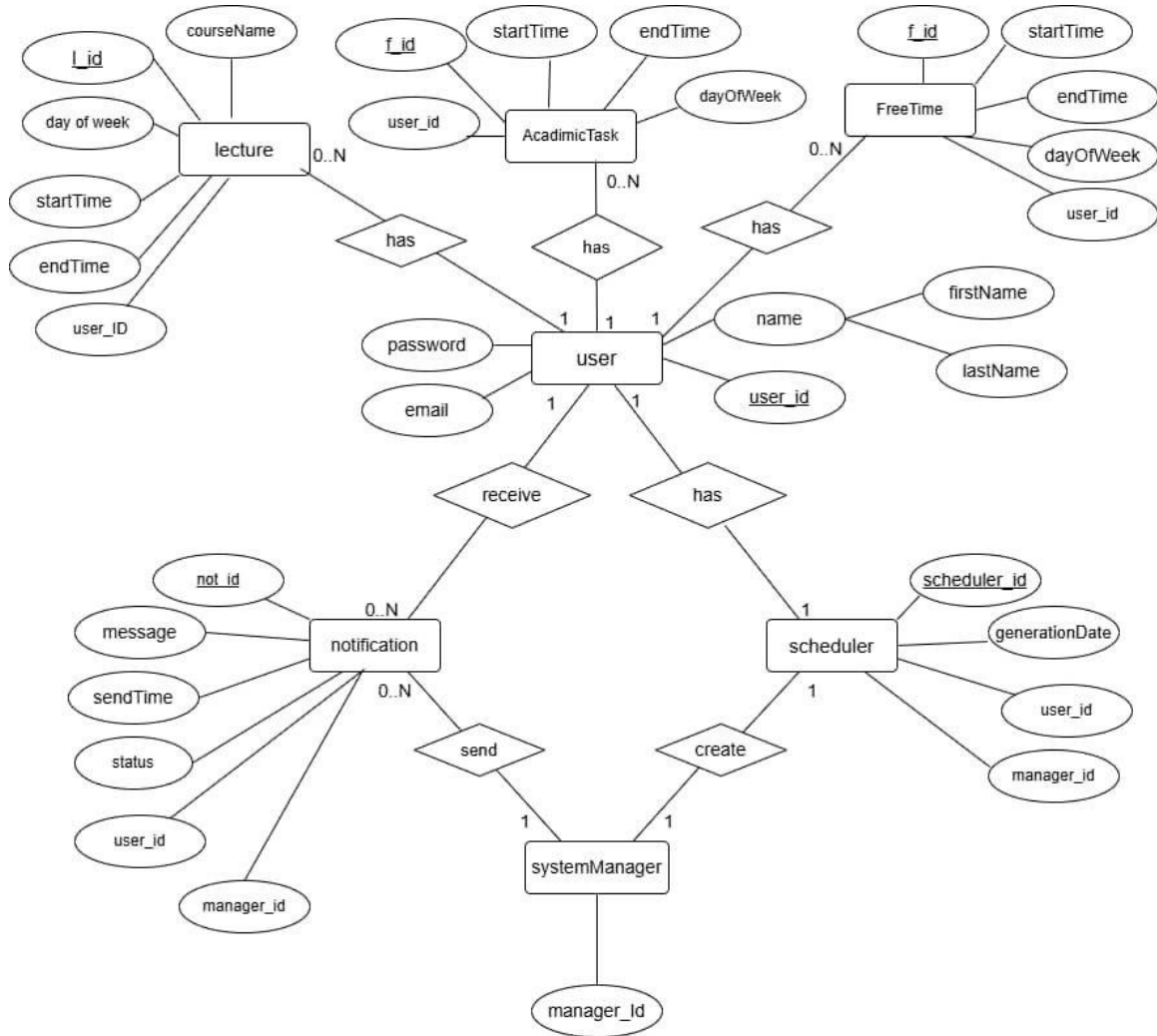
إذا نُسيت الحقول الأساسية، يُطلب إكمالها قبل الحفظ.

## مخطط حالات الاستخدام USE CASE DIAGRAM:



## CLASS DIAGRAM مخطط

### Entity Relation Diagram مخطط



### واجهة برمجة التطبيقات APIs:

مثال: API لتسجيل طالب جديد

اسم الواجهة: تسجيل طالب جديد

نوع الطلب: POST

المسار: api/students/

الوصف: تُستخدم هذه الواجهة لإنشاء حساب جديد لطالب عن طريق إرسال الاسم والبريد الإلكتروني وكلمة المرور.

حالات الخطأ المحتملة:

Bad Request 400 → البيانات ناقصة أو غير صالحة.

Conflict 409 → البريد الإلكتروني مستخدم مسبقاً.

ملاحظات:

يتم تشفير كلمة المرور قبل حفظها.

يُشترط عدم تكرار البريد الإلكتروني.

الاستخدام:

يُستخدم هذا النوع من الواجهات في صفحة التسجيل الخاصة بالطلاب الجدد، سواء على موقع إلكتروني أو تطبيق موبايل، لربط المستخدم الجديد بقاعدة البيانات وإنشاء حساب له.

أعضاء الفريق:

نعم موسى صيداوي

آيه أيمن كوبش

صفاء محمد حكمت

بتول محمد عهد شعله

علي شادي جبيلي