Republic of Yemen
Minister of Education
and Scientific Research
University of Sana'a
Faculty of Computer and Information
Computer Science Department

الجمهورية اليمنية
وزارة التربية والتعليم والبحث العلمي
جامعة صنعاء
كلية الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسوب

AI Exam Proctoring System

نظام مراقبة الامتحانات باستخدام الذكاء الاصطناعي

فريق المشروع

عبد الرحمن علي حسين رعدان	22160029
مصعب فهد قاسم الحبيشي	22164060
هادي أمين علي سيف الجعمي	22164024
مالك عبد الله أحمد حمود المصبحي	22164034
عبدالعزيز عبدالغني سعيد محمد	22164096

المشرف

د/ أحمد الشلبي

وثيقة مشروع تخرج مقدمة إلى قسم علوم الحاسوب
استكمالاً لمتطلبات الحصول على درجة البكالوريوس في قسم علوم الحاسوب

للعام 2024 /2025

۞ يَا أَيُّهَا الَّذِينَ آمَنُوا كُونُوا قَوَّامِينَ بِالْقِسْطِ شُهَدَاءَ لِلَّهِ وَلَوْ عَلَىٰ أَنفُسِكُمْ أَوِ الْوَالِدَيْنِ وَالْأَقْرَبِينَ ۚ (135)

ملخص المشروع

تُمثل نزاهة الامتحانات تحديًا مستمرًا للمؤسسات التعليمية، خاصة مع تزايد الاعتماد على الاختبارات الإلكترونية. تعتمد طرق المراقبة التقليدية بشكل كبير على المراقبين البشريين، مما قد يؤدي إلى عدم الاتساق، وصعوبة مراقبة جميع الطلاب بفعالية في نفس الوقت، ونقص الأدلة الملموسة في حالات الغش، بالإضافة إلى الازدحام والتأخير أثناء التحقق من هوية الطلاب عند الدخول. ومع التقدم في الذكاء الاصطناعي والتحول الرقمي، تبرز الحاجة إلى أنظمة آلية وموثوقة لتعزيز أمن ونزاهة العملية الامتحانية.

يقدم هذا المشروع "نظام مراقبة الامتحانات باستخدام الذكاء الاصطناعي" (AI Exam Proctoring System)، الذي يعتمد على تقنيات الذكاء الاصطناعي، الرؤية الحاسوبية (خاصة التعرف على الوجه)، وإدارة قواعد البيانات لتحسين عملية مراقبة الامتحانات والتحقق من هوية الطلاب. يهدف النظام إلى التحقق من هوية الطالب آليًا عند دخول مركز الامتحان وفي قاعة الاختبار، ومراقبة سلوك الطلاب أثناء أداء الامتحان عبر تحليل الصور الملتقطة من الكاميرات للكشف عن الأنماط المشبوهة (مثل النظر بعيدًا عن الشاشة). يقوم النظام بتسجيل هذه السلوكيات وإرسال تنبيهات فورية للمراقبين لاتخاذ الإجراءات اللازمة.

تضمن تنفيذ النظام تطوير خوارزميات للتعرف على الوجوه وتحليل السلوك، بالإضافة إلى تصميم واجهات لإدارة المستخدمين (الطلاب، المراقبين، المديرين) والأجهزة. يتطلب النظام توفر كاميرات بجودة مناسبة، واتصال إنترنت مستقر، وسيرفرات ذات قدرة معالجة كافية. تشمل التحديات والافتراضات ضرورة تدريب المراقبين، وتعاون الطلاب بعدم إعاقة الكاميرات، وتوفر قواعد بيانات طلابية وجداول امتحانات مسبقة، بالإضافة إلى إمكانية ظهور طرق غش متطورة قد لا يكتشفها النظام الحالي.

من خلال أتمتة عمليات التحقق من الهوية والمراقبة السلوكية، يساهم هذا المشروع في تعزيز نزاهة الامتحانات، وتقليل العبء على المراقبين البشريين، وتوفير أدلة موضوعية عند الحاجة، وتحسين كفاءة إدارة وتنظيم الامتحانات. تسعى التحسينات المستقبلية المحتملة إلى زيادة دقة الكشف، وتوسيع نطاق السلوكيات المكتشفة، وتحسين التكامل مع الأنظمة التعليمية القائمة.

تفويض

نحن نأذن لجامعة صنعاء كلية الحاسوب وتكنولوجيا المعلومات بتوفير نسخ من تقرير مشروع التخرج لدينا إلى المكتبات، أو المنظمات او الافراد عند الطلب، والجامعة مخولة باستخدامها في المسابقات المحلية او الدولية.

اسم الطالب	الرقم الأكاديمي	التوقيع
عبد الرحمن علي حسين رعدان	22160029	
هادي أمين علي سيف الجعمي	22164024	
مالك عبدالله أحمد حمود المصبحي	22164034	
عبدالعزيز عبدالغني سعيد محمد	22164096	
مصعب فهد قاسم الحبيشي	22164060	

Date: …/.../…….

الإهداء

إلى من كانوا النور الذي أضاء دربنا، والسند الذي اشتد به عزمنا، إلى من غرسوا فينا الأمل، وبذلوا من أجلنا العمر والجهد دون انتظار مقابل ...**آباؤنا وأمهاتنا**، أنتم سر قوتنا، ودافعنا الأول نحو هذا الإنجاز.

إلى زملائنا الذين شاركونا التحديات، وكانت عزيمتهم وقودًا لمسيرتنا، معاً تجاوزنا الصعوبات، ورسمنا خطوات النجاح ...**إخوتنا في العلم والعمل**، أنتم شركاء هذه الرحلة.

إلى وطننا الذي نحمله في قلوبنا، والذي نطمح أن نرد له بعضًا مما أعطانا... نهدي هذا العمل ليكون جزءًا من مسيرة التقدم، خطوة نحو مستقبل نؤمن أنه سيكون أفضل بجهودنا جميعًا.

فريق المشروع

شكر وتقدير

شُكرنا وخالص حَمدنا والثناء لله سبحانه وتعالى المُنعم العظيم الكريم، من لمَسنا مَعيتّه وتوفيقه لنا في كل خطوة منذ بداية المشروع وإلى نهايته فـ لله الحمد من قبل ومن بعد وفي كل وقت وحين

نتوجه أيضا بخالص الشكر والتقدير إلى الدكتور احمد الشلبي، الذي لم يدخر جهدًا في توجيهاته ودعمه خلال مراحل إعداد هذا المشروع. لقد كان لخبرته العميقة ونصائحه السديدة دور كبير في تخطي العديد من الصعوبات التي وجهناها، مما ساهم في إنجاز هذا العمل على الوجه الأمثل.

نتوجه أيضا بجزيل الشكر والتقدير إلى الأستاذ الوليد الدعيس على دعمه المستمر وإرشاداته القيمة خلال مرحلة إعداد هذا المشروع. لقد كان لتوجيهاته السديدة وملاحظاته البنّاءة أثر كبير في تحسين جودة العمل وتجاوز التحديات التي واجهتنا، مما ساهم في إتمام هذا المشروع بأفضل صورة ممكنة.

كما أود أن أعرب عن امتناني لكل من ساندنا وقدم لنا يد العون خلال فترة الدراسة، سواء من أفراد الأسرة أو الأصدقاء، على دعمهم وتشجيعهم المستمر الذي كان له الأثر الكبير في تحفيزنا على مواصلة العمل وتحقيق هذا الإنجاز.

ولا يفوتني أن أتقدم بالشكر الجزيل إلى جميع أعضاء هيئة التدريس في قسم علوم الحاسوب، على ما قدموه من علم ومعرفة، وما بذلوه من جهد في سبيل تعليمنا وتوجيهنا طوال الفترة الدراسية.

ختاماً، أسأل الله العلي القدير أن يجعل هذا العمل خالصاً لوجهه الكريم، وأن ينفع به، وأن يكون خطوة في سبيل تطوير العلم والمعرفة.

شهادة المشرف

أشهد ان إعداد هذا المشروع بعنوان AI Exam Proctoring System المقدم من الطلاب

عبد الرحمن علي حسين رعدان	22160029
هادي أمين علي سيف الجعمي	22164024
مالك عبد الله أحمد حمود المصبحي	22164034
عبدالعزيز عبدالغني سعيد محمد	22164096
مصعب فهد قاسم الحبيشي	22164060

تم تنفيذه تحت إشرافي في كلية الحاسوب جامعة صنعاء لتحقيق متطلبات نيل درجة البكالوريوس في قسم علوم الحاسوب

اسم الدكتور المشرف / أحمد الشلبي

التوقيع

التاريخ

لجنة التدقيق

عنوان المشروع : AI Exam Proctoring System

المشرف

الرقم	الاسم	الدور	التوقيع
1	د/ احمد الشلبي	مشرف	
2	أ/ الوليد الدعيس	مشرف مساعد	

لجنة المناقشة

الرقم	الاسم	الدور	التوقيع
1	أ.د/ عبدالرحمن الصبري	مناقش	
2	د/ أنور الشميري	مناقش	
3	أ/ شكري عبدالواسع	مناقش	

رئيس القسم

.....

الفهرس

آية قرآنية..... I.
الملخص..... II.
تفويض..... III.
الإهداء..... IV.
الشكر والتقدير..... V.
شهادة المشرف..... VI.
لجنة التدقيق..... VII.
الفهرس..... VIII.

الفصل الأول: مقدمة 1

1.1 نظرة عامة 2

1.2 المشاكل..... 3

1.3 الاهداف 4

1.4 النطاق..... 5

1.5الفوائد 6

1.6 المتطلبات 7

1.7 القيود 7

1.8 الافتراضات 8

1.9منهجية المشروع 8

1.11. ترتيب الفصول 8

الفصل الثاني: الدراسات السابقة 9

2.1 الخلفية..... 10

2.2 ProctorU..... 10

2.3Examity..... 11

2.4Proctorio..... 12

الفصل الثالث: توصيف متطلبات النظام(SRS).....13
مقدمة 14
المتطلبات..... 14
3.2.1 المتطلبات الوظيفية 14
3.2.2 المتطلبات غير الوظيفية 16
3.3 سيناريوهات الاستخدام 17
3.4 التصورات والافتراضات 19
3.5 المخططات.....20
3.5.1 Use Case Diagram.....20
3.5.2 Use Case Descriptions.....21
3.5.3 Sequence Diagram..... 32
3.5.4 Activity Diagram 33
3.5.4 Class Diagram 35
3.5.4 ERD 38

الفصل الرابع: تصميم المشروع 40
4.1 المقدمة 41
4.2 واجهات النظام (أمثلة) 41
4.2.1 تسجيل الدخول ..... (Sign in)
4.2.2 لوحة التحكم .... (Dashboard)
4.2.3 إدارة المستخدمين .... (Users)
4.2.4 إدارة الموجهات ..... (Vectors)
4.2.5 لوحة إحصائيات الغش ....(Cheating Statistics Dashbrd)

4.2.6 البحث المباشر ..... (Search Real Time)
4.2.7 إدارة الطلاب ..... (Student)
4.2.8 إدارة الامتحانات ..... (Exams)
4.2.9 إدارة الكليات ..... (Colleges)
4.2.10 قائمة المراكز ..... (Centers List)
4.2.11 نموذج المقررات والمواد .....(Courses And Subject Form)
4.2.12 قائمة توزيع الامتحانات ..... (Exam Distribution List)
4.2.13 نموذج توزيع الامتحانات ..... (Exam Distribution Form)
4.2.14 نموذج أنواع التنبيهات ..... (Alert Type Form)
4.2.15 اختبار نموذج وكيل التنبيه .....(Alert Agent Form Test)
4.2.16 قائمة أنواع التنبيهات ..... (Alert Type List)
4.2.17 قائمة وكلاء التنبيه ..... (Alerts Agent List)
4.2.18 قائمة الأجهزة ..... (Device List)
4.2.19 نتائج كشف الطالب (بوابة المركز) 51
4.2.20 نتائج كشف الطالب (الأجهزة) 51

الفصل الخامس: التنفيذ والاختبار 52
5.1 مقدمة 53
5.2 بيئة التنفيذ 53
5.2.1 المكونات المادية 53
5.2.2 المكونات البرمجية 53
5.3 عملية التنفيذ 54
5.3.1 تطوير الواجهات 55

5.3.2 تنفيذ الخوارزميات 55
5.3.3 ربط المكونات 55
5.3.4 تكامل النظام 57
5.4 استراتيجية الاختبار 57
5.4.1 أنواع الاختبارات 51

الفصل السادس: النتائج والمقترحات 62
6.1 مقدمة 63
6.2 النتائج الرئيسية 63
6.2.1 دقة التعرف على الوجه 63
6.2.2 كفاءة تحليل السلوك 63
6.2.3 أداء النظام 63
6.2.4 قابلية الاستخدام 64
6.3 المقترحات 64
6.3.1 تحسين دقة التعرف على الوجه 64
6.3.2 تحسين كفاءة تحليل السلوك 64
6.3.3 تحسين قابلية الاستخدام 64
6.3.4 توسيع نطاق النظام 65
6.4 خاتمة 65

الفصل السابع: التوصيات 66
7.1 مقدمة 67
7.2 التوصيات الخاصة بتطوير النظام 67

7.2.1 الأولوية العالية 67
7.2.2 الأولوية المتوسطة 67
7.2.3 الأولوية المنخفضة 67
7.3 التوصيات الخاصة بتطبيق النظام 68
7.3.1 التدريب 68
7.3.2 السياسات والإجراءات 68
7.3.3 البنية التحتية 68
7.4 التوصيات الخاصة بالبحث المستقبلي 68
7.5 خاتمة 68

الفهرس المخططات

1.9منهجية المشروع 8
Use Case Diagram 3.5.1 20
Use Case Descriptions 3.5.2 21
Sequence Diagram 3.5.3 32
Activity Diagram 3.5.4 33
Class Diagram 3.5.4 35
ERD3.5.4 38

الفصل الأول

(مقدمة)

1.1 نظرة عامة:

شهد العالم في السنوات الأخيرة تطورات كبيرة في مجالات التقنية والتكنولوجيا، مما أدى إلى تغييرات جذرية في العديد من المجالات مثل التعليم، والرعاية الصحية، والصناعة، وغيرها. أصبحت الابتكارات التكنولوجية مثل الذكاء الاصطناعي، وإنترنت الأشياء، والحوسبة السحابية أدوات أساسية تساعدنا في حياتنا اليومية وتجعل العمليات في مختلف المجالات أكثر فعالية وسهولة. بفضل هذه التطورات، أصبح بإمكاننا التعامل مع التحديات المعقدة بطرق جديدة وفعّالة.

في مجال التعليم، أصبحت المؤسسات التعليمية بحاجة ماسة لاستخدام تقنيات الذكاء الاصطناعي لتحسين نزاهة الامتحانات والحد من حالات الغش، خاصةً مع تزايد الاعتماد على الاختبارات الإلكترونية. لهذا الغرض، يهدف هذا المشروع إلى تطوير نظام رقمي لمراقبة الطلاب داخل قاعات الامتحانات باستخدام كاميرات المراقبة وتقنيات الذكاء الاصطناعي لتحليل السلوكيات.

وصف النظام:

هو نظام يتكون من خوارزميتين لإدارة والتعرف على الطلاب والوقت المتاح لهم والسماح لهم للدخول الى مركز الاختبار ولإدارة ومراقبة الامتحانات باستخدام تقنيات الذكاء الاصطناعي والتعرف على الوجه.

تتكون الخوارزمية الأولى (التعرف على هوية الطالب) من قسمين منفصلين القسم الأول وتطبق في بوابه المراكز الامتحانية وتعمل على الكشف عن هويه الطالب ومعرفة الوقت المتاح له اثناء دخول الطالب الى مركز الاختبار.

وتتكون القسم الثاني من الخوارزمية الأولى (التعرف على هوية الطالب) في معرفة المكان المخصص للطالب في قاعة الامتحان وتطبق في قبل بدأ الاختبار في قاعات الاختبار من خلال التقاط صورة للطالب من الجاهز(الحاسوب) الذي سيبدأ الاختبار فيه والتعرف على الطالب ومقارنتها برقم الجهاز المخصص للطالب مسبقا من توزيع جداول الامتحان.

في حيث ان القسم الثاني من النظام يتكون من خوارزمية تحليل سلوك الطالب اثناء الاختبار من خلال كاميرات تثبّت على أجهزة الكمبيوتر في قاعات الامتحانات بحيث ان هذه الخوارزمية تساعد على الكشف عن أي محاولات غش، وتخزين السلوك للطالب اثناء الامتحان في قاعدة بيانات للمراجعة والتحليل اللاحق، وفورياً.
يقوم البرنامج بمراقبه حركات الطالب وفي حال اكتشاف أي سلوك مشبوه يقوم البرنامج بأرسال تنبيهات فورية للمراقبين من خلال الاتي:

يقوم بأرسال تقارير السلوك الى نظام الاختبارات الذي سيقوم بدورة في إدارة هذا التنبيهات بالطرق المناسبة واتخاذ الإجراءات المناسبة للطلاب.

1.2 المشاكل:

1. **الازدحام في بوابه مراكز الامتحانات:**

- وذلك نتيجة تأخر المراقبين في فلتره الطلاب القادمين الى مركز الاختبار وفحص بياناتهم بشكل يدوي، للسماح فقط للطلاب المخصص لهم بالدخول في الوقت واليوم المحدد.
- يؤدي ذلك الى تأخير الطلاب على الامتحان، نتيجة ازدحام الممرات عند الدخول الى مركز الامتحان.

2. **دخول الطالب في الوقت الغير مخصص له:**

- في حال كان لا يوجد تدقيق في بيانات الطلاب اثناء دخول المركز.
- قد ينتج ذلك دخول طالب ما الى مركز الامتحان في وقت غير مخصص له. او دخول طالب مكان طالب اخر. او دخول طلاب ليس مصرح لهم بالدخول الى مركز الاختبار، او اختبار طالب في غير المكان المخصص له.

3. **الاعتماد الكلي على المراقبين البشريين**

- المراقبون قد يتعبون أو يتشتتون أثناء مراقبة الطلاب، مما يجعل من الصعب عليهم ملاحظة كل محاولات الغش.

- يمكن أن يؤدي ذلك إلى تفويت بعض الحالات وزيادة فرص الغش دون اكتشاف.

4. **عدم القدرة على مراقبة جميع الطلاب في نفس الوقت**

- في القاعات الاختبارية الكبيرة، من الصعب جدًا على المراقب أن يراقب كل طالب بنفس الوقت.

- هذا يمكن أن يؤدي إلى تفاوت في المراقبة ويجعل من السهل على بعض الطلاب الغش دون أن يُلاحظوا

5. **نقص الأدلة الملموسة على الغش**

- إذا لم يُمسك بالطالب متلبسًا بالغش، قد يكون من الصعب توفير دليل قوي على الغش.

- هذا يضعف من القدرة على اتخاذ إجراءات حازمة ويؤثر على مصداقية العملية التعليمية.

6. **صعوبة توثيق محاولات الغش بشكل منظم**

- تسجيل كل حالة غش يدويًا يمكن أن يكون صعبًا وفوضويًا بالنسبة للمراقبين.

- هذا يجعل من الصعب مراجعة الحالات لاحقًا أو فهم الأنماط لتحسين أساليب المراقبة.

7. تفاوت معايير المراقبة بين المراقبين

- كل مراقب قد يكون له طريقته الخاصة في مراقبة الطلاب بناءً على خبرته أو اجتهاده.

- يؤدي ذلك إلى عدم تناسق في مستوى المراقبة بين قاعة وأخرى، مما يؤثر على العدالة بين الطلاب.

8. الجهد والوقت الكبير المطلوب من المراقبين

- المراقبة اليدوية تستنزف الكثير من وقت وجهد المراقبين، مما يجعلهم يشعرون بالتعب مع مرور الوقت.

- يؤدي ذلك إلى تقليل تركيزهم وكفاءتهم في اكتشاف حالات الغش بشكل مستمر.

1.3 الاهداف:

1. تسريع عملية التحقق من هوية الطلاب

- تطوير نظام ذكي للتحقق من هوية الطلاب بشكل آلي وسريع باستخدام تقنيات التعرف على الوجه والباركود الموجود في البطاقة، مما يضمن دخولهم إلى قاعات الامتحان دون تأخير.
- تقليل التزاحم والضغط على المراقبين، وتحسين كفاءة تنظيم الامتحانات وبدءها في الوقت المحدد.

2. تقليل الضغط على المراقبين

- تطوير نظام ذكي يساعد في اكتشاف الغش، حتى لا يضطر المراقبون لبذل جهد كبير لمتابعة كل طالب طوال الوقت.

3. مراقبة كل الطلاب بنفس اللحظة

- إنشاء نظام يستخدم تقنيات الذكاء الاصطناعي لمراقبة جميع الطلاب في القاعة في نفس الوقت، للتأكد من أن الجميع تحت المراقبة ولا أحد يتمكن من الغش دون ملاحظة.

4. **توفير دليل واضح على الغش**

- تسجيل أي محاولة غش تلقائيًا، لتوفير دليل قوي يساعد في اتخاذ الإجراءات المناسبة ويجعل العملية التعليمية أكثر مصداقية.

5. **تسهيل توثيق حالات الغش**

- تصميم نظام يسجل محاولات الغش بشكل منظم، مما يسهل على المراقبين المراقبة اثناء الامتحان أو الرجوع إليها لاحقًا وفهم الأنماط لتحسين المراقبة في المستقبل.

6. **توحيد طرق المراقبة بين المراقبين**

- استخدام نظام يعتمد على تقنيات الذكاء الاصطناعي لتحديد الغش، مما يضمن أن جميع المراقبين يتبعون نفس المعايير ويوفر العدالة بين جميع الطلاب.

7. **تقليل الجهد والوقت للمراقبين**

- توفير نظام سهل الاستخدام يعرض التنبيهات بشكل واضح، مما يساعد المراقبين على اتخاذ القرارات بسرعة دون تعب أو إضاعة للوقت.

8. **تسريع عملية التحقق من هوية الطلاب**

- تطوير نظام ذكي للتحقق من هوية الطلاب بشكل آلي وسريع باستخدام تقنيات التعرف على الوجه أو بصمات الأصابع، مما يضمن دخولهم إلى قاعات الامتحان دون تأخير.
- تقليل التزاحم والضغط على المراقبين، وتحسين كفاءة تنظيم الامتحانات وبدءها في الوقت المحدد.

1.4 النطاق:

1.4.1 النطاق الوظيفي:

يساهم النظام في تحسين نزاهة الامتحانات الإلكترونية من خلال:

- تحويل صور الطلاب إلى فكرز لاستخدامها في التحقق من الهويات.
- إدارة الطلاب الذي يتم تحويل صورهم إلى فكترز.
- التحقق من هوية الطلاب باستخدام تقنيات التعرف على الوجه.
- التحقق من هوية الطالب مع بينات اختباره (المركز، والتاريخ والوقت) قبل دخول مركز الاختبارات.
- التحقق من ان كل طالب يجلس في المكان المخصص له.
- مراقبة سلوك الطلاب من خلال الكاميرات وتقنيات الذكاء الاصطناعي.
- إرسال تنبيهات فورية للمراقبين عند رصد أي سلوك مريب.

- إرسال تنبيه عند عدم تطابق رقم جلوس الطالب مع رقم الجهاز.
- إرسال تنبيه عند عدم تطابق صورة الطالب بعد دخوله في الاختبار.
- إصدار تقارير شاملة عن سلوك الطلاب بعد انتهاء الامتحانات.
- إدارة الأجهزة (تسجيل أو إضافة، تعديل، حذف، عرض قائمة الأجهزة)
- إدارة المراكز (إضافة، تعديل، حذف، عرض)
- عرض تنبيهات وفلترتها حسب (المركز، القاعة، تاريخ الاختبار، فترة الاختبار)
- عرض عدد التنبيهات وعدد الطلاب الذين غشوا في كل الكليات بناء على السنة الدراسية.
- عرض عدد التنبيهات وعدد الطلاب الذين غشوا في كل تخصص بناء الكلية والسنة الدراسية.
- عرض عدد التنبيهات وعدد الطلاب الذين غشوا في كل مادة وفلترة ذلك عبر الكلية التخصص المستوى السنة الدراسية.
- حذف التنبيهات
- عرض تفاصيل التنبيهات لكل حالة مع جعل التنبيهات الجديدة مقروءه.
- إدارة أنواع الغش(إضافة، حذف، تعديل)
- إدارة الكليات
- إدارة الأقسام
- إدارة المستويات
- إدارة السنوات الدراسية
- إدارة الفصول الدراسية
- إدارة الاختبارات (تضاف جداول الاختبارات لأنها ترتبط بالتنبيهات لذلك تعتبر ضرورية)
- إدارة توزيع الاختبارات (ليتم التأكد من رقم جهاز الطالب وقاعته ومركزه وبينات توزيع الطالب تستبدل كل مرة)
- إدارة المواد الدراسية
- إدارة المستخدمين
- ضبط موديل مراقبة الغش

1.4.2 النطاق الجغرافي:

اليمن صنعاء – جامعة صنعاء.

1.4.3 الفئة المستهدفة:

طلاب جامعة صنعاء والمراقبون المسؤولون عن مراقبة الامتحانات.

1.5 الفوائد:

تحسين فعالية المراقبة في الامتحانات، تقليل محاولات الغش، وتوفير بيئة تعليمية أكثر عدالة وشفافية.

1.6 المتطلبات:

سوف نناقشها في الفصل الثالث بتفصيل.

1.7 القيود:

يعتمد النظام على توفر كاميرات عالية من متوسطه الى الجودة وتكون في الجهة المقابلة للطالب واتصال إنترنت قوي لضمان الدقة في التحليل.

يعتمد على معالجات قوية في السيرفر بحيث يكون مناسب لعدد الأجهزة المتصلة به.

1.7.1 الزمنية:

- التسليم في الموعد الزمني المحدد من الكلية.

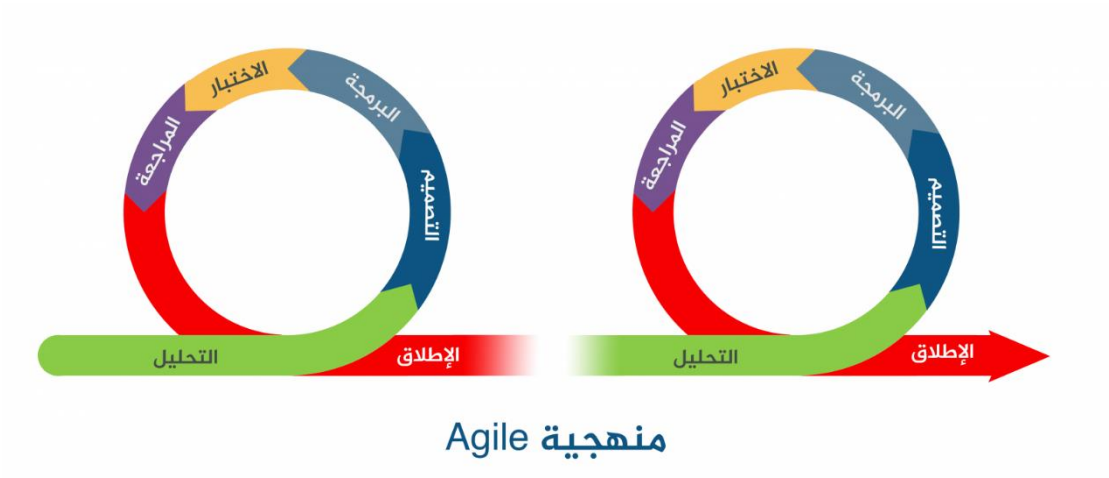
1.8 الافتراضات:

- **الطلاب** :يجب عليهم تأكيد هويتهم عند بداية الامتحان والامتثال لقوانين الامتحان.
- **المراقبون** :يستقبلون تنبيهات فورية ويستعرضون تقارير الامتحان.
- **المشرفون** :إدارة إعدادات النظام بما في ذلك البروتوكولات الأمنية.

1.8.2 التقنية:

1. يتكامل التطبيق مع نظام الامتحانات لإرسال بيانات التحقق إلى خوارزمية التعرف على الهوية، مما يتيح للبرنامج التحقق من هوية الطلاب بشكل دقيق قبل السماح لهم بدخول الامتحان. يعمل النظام على **web** ويكون متصلاً بالخادم المركزي المسؤول عن نظام مراقبة الطلاب. من خلال هذا الاتصال، يتم إرسال التنبيهات الفورية إلى المراقبين عند اكتشاف أي نشاط مشبوه أثناء الامتحان.

1.9 منهجية المشروع:



(شكل خطوات المنهجية)

1.10 ترتيب الفصول:

ويتم تنظيم بقية هذا الفصول على النحو التالي:

1. الفصل الثاني (الدراسات السابقة):
2. الفصل الثالث (SRS):
3. الفصل الرابع (تصميم النظام):
4. الفصل الخامس (تطبيق واختيار النظام):
5. الفصل السادس (النتائج والمقترحات):
6. الفصل السابع (التوصيات):

الفصل الثاني

(الدراسات السابقة)

2.1 الخلفية:

في عصرنا الحالي، ومع التقدم السريع في التكنولوجيا وظهور تقنيات جديدة، أصبح التحول الرقمي جزءاً أساسياً من حياتنا اليومية في مختلف المجالات مثل التعليم، والصحة، والتجارة. مع تزايد استخدام التكنولوجيا في العمليات اليومية، تسعى المؤسسات إلى تبني حلول ذكية تساعد في تحسين الأداء وتوفير الوقت والجهد. في مجال التعليم، مثلاً، أصبحنا بحاجة ملحة لاستخدام الذكاء الاصطناعي لمراقبة الامتحانات والحد من حالات الغش، خاصةً مع الانتشار الواسع للاختبارات الإلكترونية. هذه التكنولوجيا لا تساعد فقط في تسهيل عمليات المراقبة، بل أيضًا في ضمان نزاهة العملية التعليمية عبر حلول ذكية تراقب سلوك الطلاب بشكل مستمر وتصدر تنبيهات فورية عند ملاحظة أي نشاط غير طبيعي.

وفي هذا الفصل سوف نناقش عدة مشاريع مشابهة لمشروعنا مع ذكر مميزاتها وعيوبها.

2.2. ProctorU

2.2.1 نظرة عامة:

ProctorU هو نظام مراقبة للاختبارات عبر الإنترنت، يجمع بين التكنولوجيا والمراقبة البشرية. يسمح للطلاب بأداء امتحاناتهم عن بُعد، مع وجود مراقبين يتابعونهم عبر كاميرات الويب. كما يعتمد على الذكاء الاصطناعي لكشف أي سلوكيات مريبة.

2.2.2المميزات:

- يوفر مراقبة مباشرة بالفيديو والصوت.
- يستخدم الذكاء الاصطناعي لتحليل سلوك الطلاب والتنبيه في حال اكتشاف أي شيء غير طبيعي.
- يقدم تقارير مفصلة للمؤسسات التعليمية تشمل التسجيلات والملاحظات.

2.2.3 المشاكل:

- يواجه المستخدمون مشاكل تقنية، مثل بطء الاتصال أو تعطل البرنامج.
- التكلفة قد تكون مرتفعة للمؤسسات الصغيرة.

- الموقع: (https://www.proctoru.com)

2.3. Examity

2.3.1نظرة عامة:

Examity هو نظام مراقبة اختبارات عبر الإنترنت يوفر خيارات مرنة، حيث يمكن للمؤسسات التعليمية الاختيار بين مراقبة آلية أو بشرية. يستخدم تقنيات الذكاء الاصطناعي لكشف الغش ويتيح إعداد تقارير مفصلة.

2.3.2 المميزات:

- يقدم مستويات مراقبة متنوعة تناسب مختلف الاحتياجات.
- يدعم العديد من اللغات لتسهيل الاستخدام العالمي.
- يتيح تقارير مفصلة عن كل اختبار.

2.3.3 المشاكل:

- خيارات المراقبة صعبة بعض الشيء للمستخدمين الجدد.
- يتطلب اتصال إنترنت قوي لضمان فعالية المراقبة.

- الموقع: (https://www.examity.com)

Proctorio 2.4

2.4.1 نظرة عامة:

Proctorio هو نظام مراقبة عبر الإنترنت يعتمد بالكامل على الذكاء الاصطناعي لمراقبة الطلاب أثناء الامتحانات. يمكنه تحليل الأنماط السلوكية بشكل متقدم للكشف عن أي محاولات للغش.

2.4.2 المميزات:

- يستخدم الذكاء الاصطناعي بشكل كامل دون الحاجة لمراقبين بشريين.
- يدعم أجهزة وأنظمة تشغيل متعددة.
- يقدم تقارير دقيقة وتحليلات شاملة للامتحانات.

2.4.3 المشاكل:

- التكلفة قد تكون مرتفعة لبعض المؤسسات.
- من الممكن أن يعطي نتائج سلبية خاطئة بناءً على التحليل الآلي.

- الموقع: (https://proctorio.com)

2.4.4 المشاكل التي يعالجها النظام :

- تحسين الاستقرار وسهولة الاستخدام :سنتجنب المشاكل التقنية الشائعة عبر تصميم نظام مستقر وذو واجهة مستخدم واضحة وبديهية، مما يضمن تجربة سلسة للمراقبين والطلاب ويقلل من الحاجة إلى دعم فني مكثف .

- تقديم حلول مرنة واقتصادية :إدراكًا لأهمية التكلفة، سيوفر مشروعنا النظام بشكل يناسب مع جميع مواصفات الأجهزة المتوسطة، مع الحفاظ على جودة الميزات الأساسية .

- تعزيز دقة المراقبة وتقليل الأخطاء :سيعتمد نظامنا على تقنيات الذكاء الاصطناعي مع إمكانية المراجعة البشرية عند الضرورة. هذا النهج يهدف إلى تحقيق توازن بين الكفاءة الآلية وتقليل احتمالية إصدار نتائج سلبية خاطئة، وبالتالي ضمان عدالة أكبر في عملية اتخاذ القرار.

الفصل الثالث

SRS

Software Requirements)
(Specification

3.1 مقدمة

سيتم توصيف ونمذجة العمليات في النظام المراد بناؤه باستخدام (Unified Modeling Language UML)هي مرحلة مهمة في عملية تطوير البرمجيات، حيث أن UMLهي أداة تحليلية توضح آلية عمل النظام من خلال استخدام الأشكال الهندسية.

هذه الأشكال والرسوم البيانية تعكس صورة كاملة للبرنامج المراد إنشاؤه
هذه الطريقة مرنة وتسهل عملية التصور الكامل للبرنامج مما يسهل صيانته وإزالة العيوب بالإضافة إلى إمكانية توزيع هذه المخططات إلى مجموعة من المبرمجين حتى يتمكنوا من تطوير البرنامج بالتوازي مما يسرع عملية إنشاء البرامج

يمكن أيضًا مراجعة هذه المخططات مرة أخرى بعد فترة طويلة لفهم النظام بسهولة.
تم تصميم هذه الطريقة من قبل مجموعة من الخبراء والمتخصصين، بما في ذلك "بوش" وآخرين، وتم دمج أساليبهم المختلفة في UML
التوجه نحو الكائن باستخدام UML هو لغة النمذجة الموحدة المعتمدة للترميز في البيئة الصناعية
يتم تطبيق هذه اللغة في التحليل والنهج المستخدم حاليًا، والذي يتضمن وسائل رمزية مبسطة للتعبير عن البرامج المختلفة
توفر UML نماذج عمل تسهل التواصل وتبادل المعلومات بين المحللين، المصممين، المبرمجين وحتى المستخدمين UML .
تشبه لغة التخطيط التي يستخدمها المهندسون والمعماريون والمساحون للتعبير عن مخططات البناء أو الكهربائية والإلكترونية وغيرها

3.3 المتطلبات

3.3.1 المتطلبات الوظيفية

متطلبات المستخدم (الطالب):

- **التحقق من الهوية عند دخول مركز الامتحان :**
  - السماح للنظام بالوصول إلى كاميرا الجهاز لالتقاط صورة الوجه.
  - إرسال الصورة الملتقطة إلى النظام للتحقق من الهوية.

- استلام نتيجة التحقق (قبول أو رفض الدخول).

- **التحقق من الهوية عند بدء الامتحان :**
  - السماح للنظام بالوصول إلى كاميرا الجهاز لالتقاط صورة الوجه.
  - إرسال الصورة وبيانات الجهاز الحالي إلى النظام للتحقق من الهوية والموقع.
  - التأكد من أن الجهاز المستخدم هو الجهاز المخصص للامتحان (إذا لزم الأمر).
- **المراقبة أثناء الامتحان :**
  - السماح للنظام بالوصول إلى كاميرا الجهاز لمراقبة السلوك.
  - تشغيل عملية تحليل السلوك داخل المتصفح لتقليل الحمل على الخادم.
- **استقبال التنبيهات (بشكل غير مباشر) :**
  - يجب أن يكون النظام واضحًا في حال تم اكتشاف سلوك مشبوه وقد يؤدي إلى تدخل المراقب. (لا يوجد تفاعل مباشر مع التنبيهات، ولكن سلوك الطالب يؤدي إلى إرسالها).

ب. متطلبات المستخدم (المراقب):

- **استقبال الإشعارات الفورية :**
  - تلقي تنبيهات فورية عند اكتشاف سلوك مشبوه أو محاولة غش من قبل الطلاب أثناء الامتحان.
- **التفاعل مع التنبيهات :**
  - عرض تفاصيل التنبيه (مثل نوع السلوك المشبوه، وقت الحدوث، الطالب المعني).
  - إمكانية مراجعة التسجيلات (إذا توفرت) للسلوك المشبوه.
  - اتخاذ الإجراء المناسب بناءً على التنبيه (مثل توجيه إنذار للطالب عبر النظام أو اتخاذ قرار بإنهاء الامتحان).
- **مراقبة حالة الكاميرات :**
  - تلقي تنبيهات فورية في حال وجود مشكلة في كاميرات الطلاب (توقف، عدم اكتشاف الوجه).
- **إدارة الجلسات الامتحانية :**قد تعتمد على مستوى صلاحيات المراقب

- عرض قائمة بالجلسات الامتحانية الحالية والطلاب المشاركين.
- مراقبة حالة الطلاب في كل جلسة.

ج. متطلبات المستخدم (المدير/المسؤول):

- **إدارة حسابات المستخدمين والمراقبين :**
  - إنشاء حسابات جديدة للمستخدمين والمراقبين.
  - تعديل صلاحيات المستخدمين والمراقبين.
  - حذف حسابات المستخدمين والمراقبين.
  - إنشاء حسابات لأجهزة الكمبيوتر المستخدمة في القاعات.
  - تسجيل وتتبع أرقام الأجهزة وأرقام القاعات.
- **استقبال تنبيهات النظام :**
  - تلقي تنبيهات فورية في حال وجود مشاكل فنية عامة في النظام.
  - تلقي تنبيهات حول مشاكل الكاميرات التي قد تؤثر على عملية المراقبة.
- **إدارة نموذج الذكاء الاصطناعي للمراقبة :**
  - تحديد معايير المراقبة (مثل أنواع الحركات أو السلوكيات التي تعتبر مشبوهة).
  - التحكم في حساسية ودقة نموذج الذكاء الاصطناعي.
  - تحديد عدد مرات التنبيهات المسموح بها.
  - تحديد زوايا الرؤية التي يجب على النظام التركيز عليها.
  - تعديل هذه الإعدادات في أي وقت دون الحاجة إلى تعديل في الكود نفسه.
- **التكامل مع قواعد البيانات الخارجية :**
  - ضمان الربط السلس مع قواعد بيانات الطلاب، الاختبارات، الأجهزة، والمشرفين.
  - إدارة عملية التحقق من جداول الامتحانات وتخصيص الأجهزة والقاعات بشكل ديناميكي.

3.3.4 المتطلبات غير الوظيفية

- **الأداء** :يجب أن يكون الجهاز قادرًا على معالجة الفيديو وتحليله في الوقت الفعلي مع تأخير لا يتجاوز 5 ثوان لضمان أمان البيانات.
- **قابلية التوسع** :النظام يجب أن يكون قادرًا على مراقبة ما يصل إلى 500 طالب في جلسة واحدة دون التأثير على الأداء.
- **سهولة الاستخدام** :يجب أن يتمكن المراقبون من استخدام النظام بسهولة دون الحاجة إلى تدريب مكثف.
- **سرعة الانترنت:** يتطلب اتصالًا ثابتًا بالإنترنت وسرعة لا تقل عن 5 ميغابت/ثانية لضمان ارسال واستقبال البيانات والاشعارات.

3.5 سيناريوهات الاستخدام

1. تسجيل الدخول

- بعد ان يتم تشغيل النظام في السيرفر يستطيع المستخدمين (agent) الدخول عبر المتصفح من خلال الأجهزة وعندها يخزن توكن لكل جهاز ليكون كل جهاز تكون خاص به.

- بعد ان يتم تسجيل الأجهزة يصبح للمراقب او المدير لديه صلاحية الوصول إلى النظام عبر واجهة تسجيل دخول مخصصة. عند الدخول، يقوم المدير بإدخال اسم المستخدم وكلمة المرور الخاصة به. بمجرد تسجيل الدخول بنجاح، يمكنه إدارة حسابات المراقبين، بما في ذلك أداة الأجهزة وإضافة مراقبين جدد، تعديل الصلاحيات، أو حذف الحسابات غير النشطة. بالإضافة إلى ذلك، يستطيع المدير الاطلاع على البيانات الواردة من الامتحانات الجارية وإجراء التعديلات الفورية عند الضرورة.

2. التحقق من الهوية الطالب

- **التحقق من الهوية الطالب عند الدخول من البوابة المركز الامتحان**

عند وصول الطالب إلى مركز الامتحان، يتوجه مباشرة إلى نقطة التفتيش حيث انه يوجد فيها أجهزة قد تم تسجيل الدخول للجهاز الى النظام منها (قد تم تسجيلها وحفظ Token).
حيث يتم اخذ صورة الطالب وجلب بيانات الطالب من الصورة ورقم القيد ووقت الاختبار وغيرها ليتم من خلال هذه البيانات التحقق أولاً من وقت اختبار الطالب وإذا كان

الوقت هو الوقت الصحيح لهذا الطالب من هذا الكلية مطابق للوقت المخزن في جدول الاختبارات لهذه الكلية وهذه السنة وهذا القسم والترم الصحيحة.
بعد ذلك يقوم النظام باستعلام عن بيانات الطالب الذي تم ارسالها سابقا. يتم الاستعلام باستخدام الباركود والصورة المرسلة في الوقت الفعلي وإرسالها مباشرة إلى نظام التعرف على الهوية المدمج مع نظام الاختبارات في السيرفر.
ثم يقوم النظام بتحليل الصورة عن طريق خوارزمية تعتمد على الذكاء الاصطناعي، والتي تقارن الصورة الحالية مع البيانات المسجلة مسبقًا للطالب التي تم جلب البيانات من خلال الباركود.
إذا تطابقت البيانات، بمقدار أكثر من قيمه يتم تحديدها من قبل المراقب او الدير مثلا 70%.
ثم في حالة تم التعرف على هوية الطالب يتم التحقق من بيانات موعد ومكان اختبار الطالب عبر البيانات المستوردة من النظام الرئيسي ومقارنتها بالبيانات في الوقت والمكان الحالي، وفي حال كانت البيانات صحيحة يسمح للطالب بالدخول إلى مركز الامتحانات، ويرفض النظام إذا كان وقت او مكان الطالب غير متطابق، مع ارسال سبب رفض الدخول الى مركز الاختبار.
في حال وجود أي اختلاف أو عدم تطابق في الصورة، يُمنع الطالب من الدخول ويتم إرسال إشعار فوري إلى المراقبين المتواجدين في المركز لاتخاذ الإجراءات المناسبة.

المدير :يستطيع المدير متابعة عملية التحقق من هوية الطلاب عبر لوحة تحكم مركزية. هذه اللوحة تتيح له الاطلاع على جميع محاولات التحقق التي تمت عند جميع المداخل مراكز الاختبارات. إذا فشل أي طالب في عملية التحقق، يتلقى المدير تنبيهًا يتضمن صورة الطالب الملتقطة، وسبب الفشل (مثل عدم تطابق الصورة مع البيانات المخزنة). بناءً على هذه المعلومات، يمكن للمدير اتخاذ قرارات فورية، مثل التواصل مع المراقبين لحل المشكلة، أو تقديم استثناء خاص في حال وجود خلل تقني.

- **التحقق من هوية الطالب داخل القاعة**

  - بمجرد جلوس الطالب على الجهاز المخصص له في قاعة الامتحانات، يتم التقاط صورة جديدة للتحقق من أن الشخص الجالس هو نفس الطالب الذي خصص له المكان هذا في جدول الاختبارات.

  - الخوارزمية تتحقق مرة أخرى من هوية الطالب بمطابقة الصورة الجديدة مع الصور الأصلية للطالب المخصص له هذا المكان. هذه العملية تضمن عدم تبديل الطالب مكانه مع أي شخص آخر. في حال تم التحقق بنجاح، يستمر الطالب في أداء الامتحان. أما في حال اكتشاف أي تباين بين الصور، فيتم إرسال إشعار فوري إلى المراقبين.

- **المدير :**من خلال لوحة التحكم، يستطيع المدير مراقبة جميع عمليات التحقق داخل القاعة بشكل مستمر. إذا فشل الطالب في عملية التحقق داخل القاعة، يتم تبليغ المدير مباشرةً عبر إشعار يتضمن تفاصيل الطالب والجهاز الذي يجلس عليه. يمكن للمدير إما إعادة التحقق يدويًا أو إرسال تنبيه إلى المراقبين لتوجيه الطالب إلى مكانه الصحيح، أو لاتخاذ إجراء تأديبي إذا لزم الأمر.

3. مراقبة الطلاب

- **الطالب** :أثناء الامتحان، يتم مراقبة الطالب بشكل مستمر من خلال الكاميرات المثبتة امام الطلاب. هذه الكاميرات متصلة بالنظام الذي يكون مثبت على جهاز الحاسوب، الذي يعتمد على الذكاء الاصطناعي لتحليل سلوكيات الطلاب. على سبيل المثال، إذا قام الطالب بالنظر بعيدًا عن الشاشة لفترة طويلة، يقوم النظام بتحديد هذا السلوك ك "مشبوه" ويصدر تنبيه للطالب بانه ينظر بعيدا وإذا تجاوز سيناريو معين حسب تحديد مدير النظام وإذا تم تجاوز السيناريو يتم ارسال تحذير للمراقبين ومدير النظام مع تسجيل زمن هذا السلوك، وترسل الاشعارات الى النظام المركزي في السيرفر لتخزينها وارسالها للمراقبين.

- **المدير** :من لوحة التحكم، يمكن للمدير الاطلاع على سلوكيات الطلاب أثناء الامتحان بشكل مباشر. عبر البث الحي للتنبيهات الواردة من نظام المراقبة في أجهزة الاختبارات. وتتم المراقبة لجميع الأجهزة في شاشة عرض جميع الأجهزة، في حال اكتشاف أي سلوك مريب، يتلقى المدير إشعارًا فوريًا يتضمن تفاصيل حول السلوك المكتشف، مثل التوقيت والتاريخ، وموقع الطالب في القاعة (رقم الجهاز). وبيانات الطالب (اسمه ورقم القيد) بناءً على هذه التحذيرات التي ترسل للمراقبين للتعامل مع الطالب.

4. الخاتمة

- من خلال هذه السيناريوهات التفصيلية، يظهر النظام كأداة قوية لدعم الإدارة الفعالة للامتحانات من خلال التعرف على الهوية والمراقبة المستمرة للسلوكيات. يلعب كل من الطالب والمدير دورًا محوريًا في هذه العملية لضمان نزاهة وشفافية الامتحانات.

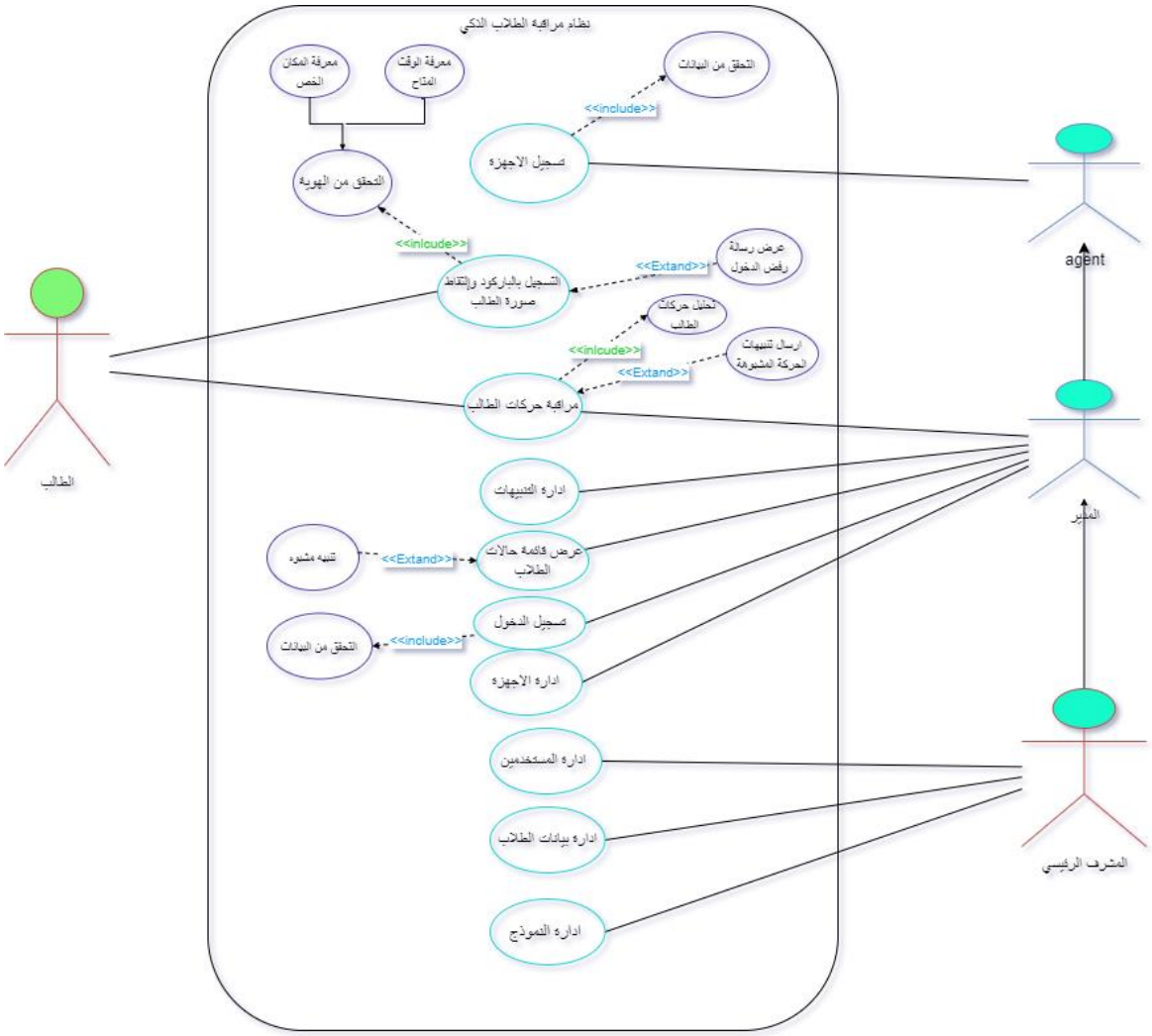
3.4 التصورات والافتراضات

- النظام يفترض أن جميع الطلاب سيتبعون القواعد ولن يحاولوا تعطيل النظام.
- يجب أن يكون المراقبون مدربين جيداً على استخدام النظام والاستجابة للتنبيهات.

- يُفترض أن أنظمة الامتحانات موجود مسبقا، بما في ذلك قاعدة بيانات الطلاب وجدول الامتحانات وبيانات الطلاب مثل موعد اختباره وتحصصه.
- يفترض تواجد كاميرات لجميع أجهزة الاختبارات. تكون ذات جودة متوسطة.
- النظام يفترض أن الطلاب لن يستخدموا تقنيات جديدة أو متطورة في محاولات الغش لم يتم برمجة النظام للتعامل معها.
- يفترض ان يظهر وجوة جميع الطلاب الذين يتم مراقبتهم، وعدم وجود أشياء ستعيق عمل النظام مثل الشال او الكوفية او مثلا البرقع في حال كان يتم مراقبه الطالبات.

3.5 المخططات

Use Case Diagram 3.5.1



(figure 1)

Use Case Descriptions 3.5.2()

Use Case Number	**01**
Use Case Name	التسجيل برقم القيد
Overview	يقوم الطالب بإدخال الرقم القيد الذي في البطاقة الجامعية، في واجهة النظام
Actor(s)	الطالب
Pre-condition	يحتاج بطاقة جامعية ويحتاج الى ان يكون في واجهات واجهات(جهاز)
Scenario Flow	1. يفتح الواجهة. 2. يدخل رقم القيد لكي يقرأ النظام هل الموجود فيه 3. يقوم النظام بجلب بيانات الطالب الى الكاش 4. عرض رسالة تأكيد
Alternate Flown	4a. عرض رسالة لم يتم إيجاد رقم القيد
Post Conditions	تظهر واجهة التحقق من الصورة

(figure 2)

Use Case Number	**02-A**
Use Case Name	التقاط صورة الطالب
Overview	التحقق من هويه الطالب بالتقاط صورة له وارسالها لكي يتم التحقق من هويته والوقت المتاح له
Actor(s)	الطالب
Pre-condition	يجب ان يكون قد سجل دخول برقم القيد الذي في البطاقة
Scenario Flow	1. يلتقط الطالب صورة له 2. يرسل الطالب الصورة 3. يتم التحقق من صورة الطالب اذا كانت مطابقة لنفس الشخص صاحب رقم القيد السابق 4. يتم التحقق من الوقت المتاح له 5. عرض سالة تأكيد العبور
Alternate Flown	1. يعيد الطالب التقاط الصورة 2. يرفض النظام اذا لم يتطابق الصورة مع صورة الطالب المخزنة مسبقا 3. يرفض النظام اذا لم يكون الوقت الحالي هو وقت اختبار النظام 4. يتم عرض رسالة رفض الدخول وارسال تنبيه للمدير
Post Conditions	يسمح للطلاب بالدخول الى مركز الامتحانات

(figure 3)

Use Case Number	**02-B**
Use Case Name	التقاط صورة الطالب
Overview	التحقق من هويه الطالب بالتقاط صورة له وارسالها لكي يتم التحقق من هويته والمكان المخصص له
Actor(s)	الطالب
Pre-condition	يجب ان يكون قد سجل دخول برقم القيد الذي في البطاقة
Scenario Flow	1. يلتقط الطالب صورة له 2. يرسل الطالب الصورة 3. يتم التحقق من صورة الطالب اذا كانت مطابقة لنفس الشخص 4. يتم التحقق من المكان المخصص له 5. ارسال رسالة التأكد للسيرفر لتخزينها
Alternate Flown	1. يعيد الطالب التقاط الصورة 2. يرسل النظام تنبيه للأمن اذا لم يتطابق الصورة مع صورة الطالب الحقيقية 3. يرسل النظام تنبيه للأمن اذا لم يكون المكان الحالي هو المكان المخصص له في النظام
Post Conditions	يكتشف عمليه استبدال الأماكن بين الطلاب

(figure 4)

Use Case Number	**03**
Use Case Name	مراقبة حركات الطالب
Overview	يراقب حركات الطالب اثناء الامتحان ليتم اكتشاف أي سلوك مشبوه يقوم به الطالب
Actor(s)	الطالب والمدير
Pre-condition	يكون النظام مثبت في الجهاز، ومتصل بالسيرفر. ينبغي ان يشتغل النظام مع بدأ الاختبار. يجب ان يكون بيانات الجهاز مسجله في النظام مع رقم الطالب الجامعي
Scenario Flow	1. تبدأ عملية المراقبة عند بدأ الامتحان 2. عند أي سلوك مشبوه يرسل النظام تنبيه بهذا السلوك للسيرفر 3. يقوم السيرفر بتخزين هذا السلوك للطالب المحدد 4. يعرض النظام السلوك المشبوه للأدمن والمراقبين
Alternate Flown	2a.عند ما لا يوجد تستمر المراقبة 2b.تستمر علمية المراقبة الى نهاية الاختبار
Post Conditions	يكتشف عمليات الغش اثناء الامتحان

(figure 5)

Use Case Number	**04**
Use Case Name	التسجيل الدخول
Overview	يقوم المدير بتسجيل الدخول للنظام من السيرفر ليتم إدارة النظام
Actor(s)	المدير
Pre-condition	يجب ان يكون لديه حساب في النظام
Scenario Flow	1. يدخل المدير اسم المستخدم 2. يدخل المدير رقم السر 3. ينقر على تسجيل الدخول 4. تعرض له سالة التأكيد
Alternate Flown	4a. يعرض له رسالة الفشل في كانت البيانات خاطئة
Post Conditions	يدخل الى النظام

(figure 6)

Use Case Number	**05**
Use Case Name	إدارة الأجهزة
Overview	يعمل المشرف الرئيسي ب CRUD للأجهزة التي يتم فيها الاختبارات
Actor(s)	المشرف الرئيسي
Pre-condition	يجب ان يكون مسجل دخول. يجب ان يكون مشرف رئيسي
Scenario Flow	1. يدخل الى صفحة الأجهزة 2. يعمل أي عمليه للأجهزة انشاء تعديل حذف عرض 3. يحفظ البيانات في قاعدة البيانات
Alternate Flown	3a.يرفض عملية الحفظ في حال كانت البيانات غير صحيحة او مكررة
Post Conditions	إدارة الأجهزة

(figure 7)

Use Case Number	**06**
Use Case Name	إدارة المستخدمين
Overview	يعمل المشرف الرئيسي ب CRUD للمستخدمين (المراقبين)
Actor(s)	المشرف الرئيسي
Pre-condition	يجب ان يكون مسجل دخول. يجب ان يكون مشرف رئيسي
Scenario Flow	1. يدخل الى صفحة عرض المستخدمين 2. يعمل أي عمليه للمستخدمين انشاء تعديل حذف عرض 3. يحفظ البيانات في قاعدة البيانات
Alternate Flown	3a.يرفض عملية الحفظ في حال كانت البيانات غير صحيحة او مكررة
Post Conditions	إدارة المستخدمين

(figure 8)

Use Case Number	**07**
Use Case Name	إدارة بيانات الطلاب
Overview	يعمل المشرف الرئيسي ب CRUD للطلاب
Actor(s)	المشرف الرئيسي
Pre-condition	يجب ان يكون مسجل دخول. يجب ان يكون مشرف رئيسي يجب ان يمتلك صور للطلاب الجدد
Scenario Flow	1. يدخل الى صفحة عرض الطلاب 2. يعمل أي عمليه للطلاب انشاء تعديل حذف عرض 3. يحفظ البيانات في قاعدة البيانات
Alternate Flown	3a.يرفض عملية الحفظ في حال كانت البيانات غير صحيحة او مكررة
Post Conditions	إدارة بيانات الطلاب

(figur9)

Use Case Number	**08**
Use Case Name	تسجيل بيانات وجه الطالب
Overview	يقوم المشرف بإضافة الصور المرجعية لوجه طالب معين إلى النظام. يقوم النظام بمعالجة هذه الصور لإنشاء تمثيل رقمي (مثل المتجه أو البصمة الوجهية) يُستخدم لاحقًا للتحقق من هوية الطالب عند دخوله مركز الاختبارات أو جلوسه في قاعة الامتحان.
Actor(s)	المشرف الرئيسي
Pre-condition	ينبغي أن تكون بيانات الطالب الأساسية (مثل رقم القيد، الاسم) مسجلة مسبقًا في النظام. يجب توفر صور واضحة لوجه الطالب المراد تسجيله.
Scenario Flow	1. يفتح المشرف الرئيسي واجهة إدارة بيانات الطلاب أو الواجهة المخصصة لتسجيل الوجوه. 2. يقوم المشرف بالبحث عن الطالب أو إدخال رقم قيده لتحديد سجله. 3. يقوم المشرف برفع مجموعة الصور المطلوبة لوجه الطالب المحدد. 4. ينقر المشرف على زر "حفظ بيانات الوجه" أو "تسجيل الوجه". 5. يقوم النظام بمعالجة الصور المدخلة، استخلاص السمات الوجهية، وإنشاء التمثيل الرقمي الخاص بوجه الطالب. 6. يتم حفظ هذا التمثيل الرقمي في قاعدة البيانات وربطه بمعرف الطالب. 7. يعرض النظام رسالة تأكيد نجاح عملية تسجيل بيانات وجه الطالب.
Alternate Flown	• فشل معالجة الصور: إذا كانت الصور غير واضحة أو لا تفي بالمتطلبات، يعرض النظام رسالة خطأ ويطلب إعادة رفع صور مناسبة. • الطالب مسجل مسبقًا: إذا كان للطالب بيانات وجه مسجلة بالفعل، قد يعرض النظام تنبيهًا ويسأل عن تأكيد التحديث أو الإلغاء.
Post Conditions	• تم إنشاء وحفظ التمثيل الرقمي لوجه الطالب في قاعدة البيانات وربطه بمعرفه. • أصبح النظام قادرًا على استخدام هذه البيانات للتحقق من هوية الطالب لاحقًا عبر مقارنتها بصورة مباشرة يتم التقاطها للطالب.

(figure 10)

(figure 1)

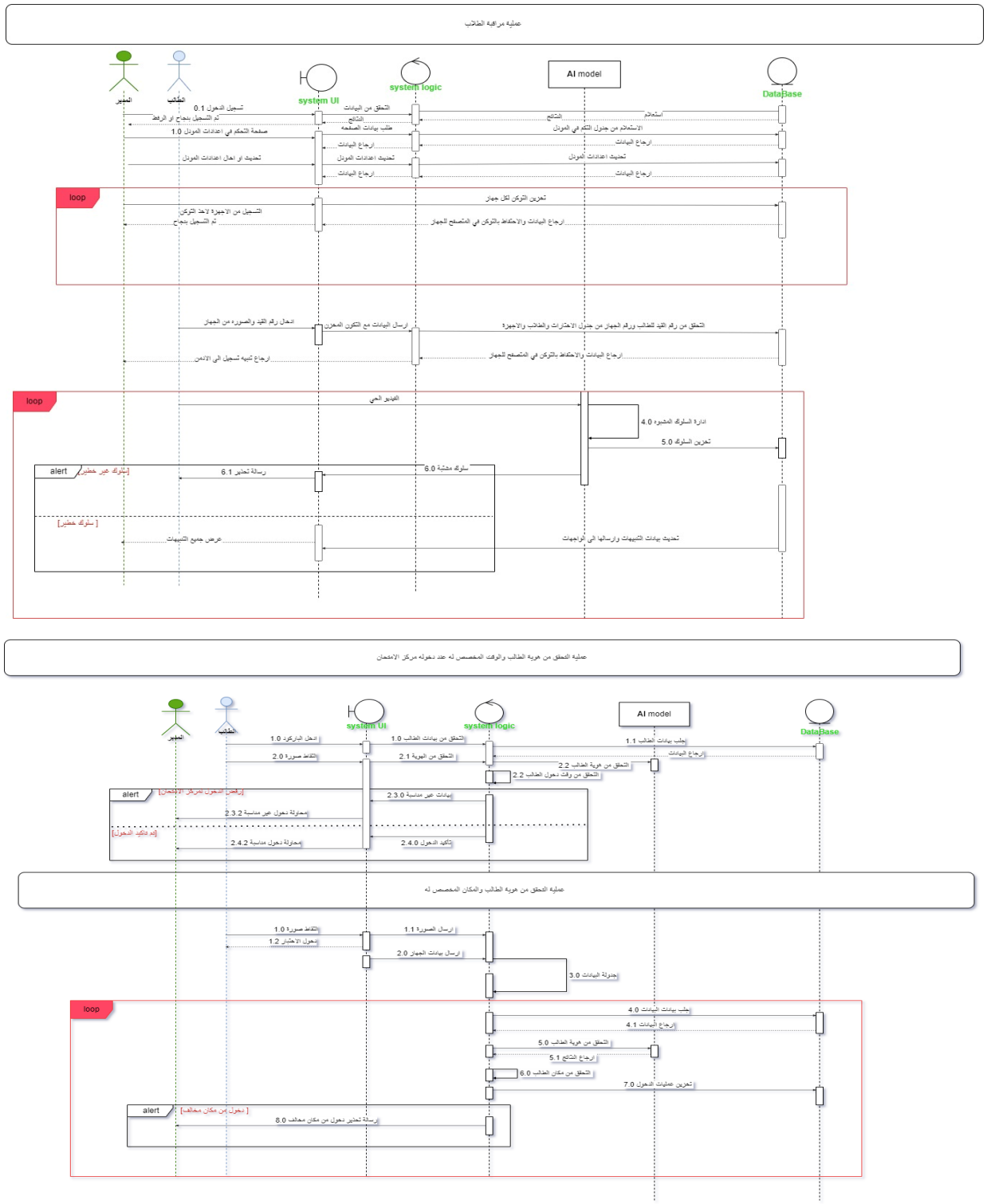
Use Case Number	**09**
Use Case Name	عرض قائمة الطلاب
Overview	عرض في السيرفر قائمة الطلاب الذي تم التقاط سلوك مشبوه اثناء الامتحان
Actor(s)	المستخدمين (المراقب – المشرف الرئيسي)
Pre-condition	يجب ان يكون مسجل دخول يجب ان يكون في وقت الفعلي للامتحان
Scenario Flow	1. يدخل المراقب الى صفحة إدارة الامتحان 2. يستقل المراقب التنبيهات المشبوه فيها صورة ورقم الجهاز ومقدار الحركة وعدد المرات التي تم التقاط التنبيهات 3. يطلع المراقب على التنبيه
Alternate Flown	3a.ينقر المراقب على أي تنبيه يريده ويطلع على بيانات الطالب مع تفاصيل السلوك المشبوه (السلوك- الوقت- والمدة)
Post Conditions	يدير مراقبة الامتحان

(figure 11)

Use Case Number	**10**
Use Case Name	إدارة التنبيهات
Overview	يتم إدارة التنبيهات الخاصة بالطلاب في أي وقت بعد الامتحان عرض او مراجعة
Actor(s)	المدير، والمشرف الرئيسي
Pre-condition	يجب ان يكون مسجل دخول
Scenario Flow	1. يدخل المستخدم الى صفحة إدارة التنبيهات 2. يبحث عن التنبيه الذي يريده باستخدام رقم قيد الطالب 3. يكون في فلترة للتنبيهات حسب اليوم والفترة والتخصص
Alternate Flown	
Post Conditions	إدارة التنبيهات ومراجعتها عند الحاجة

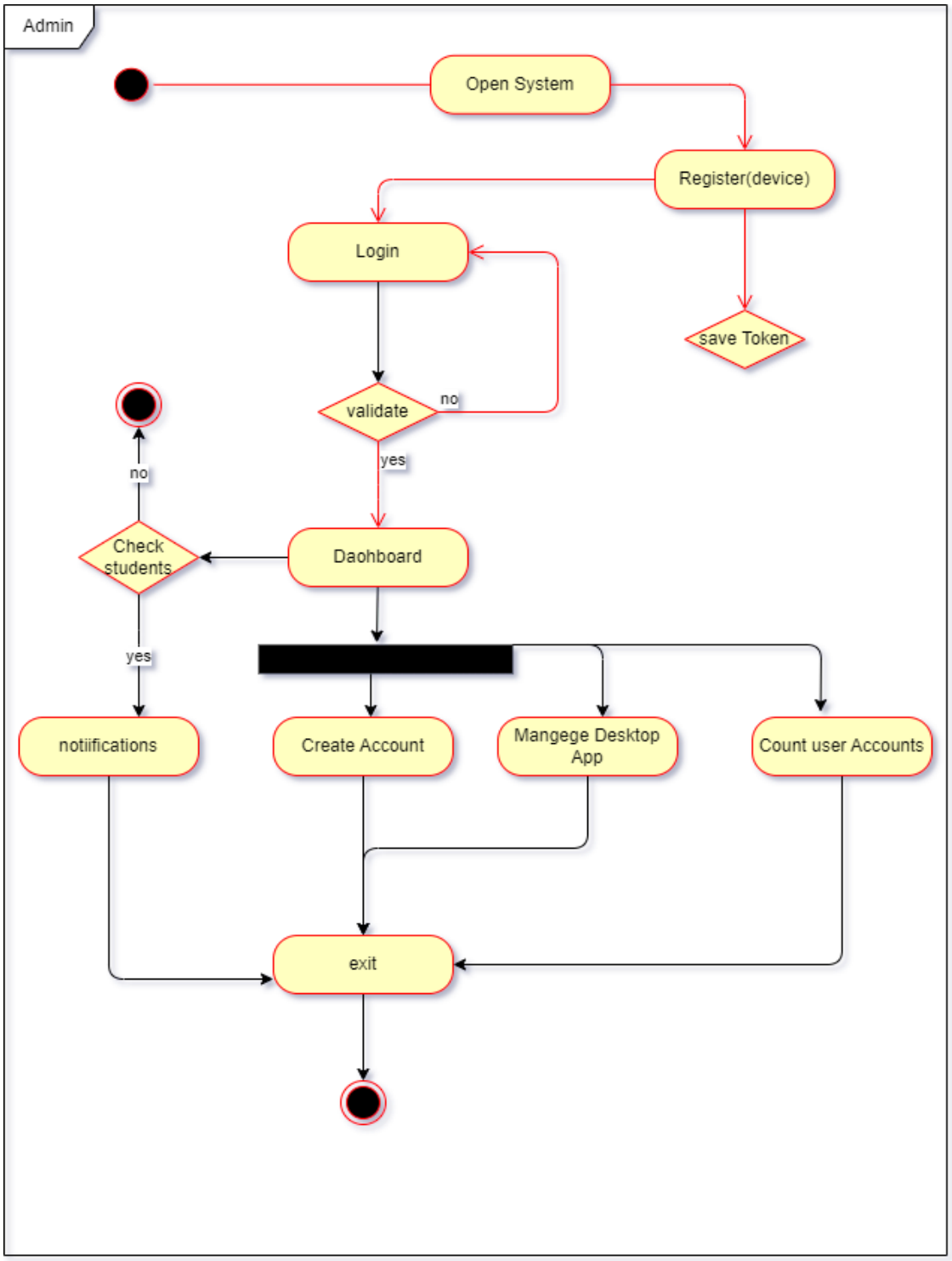
(figure 12)

Sequence Diagram 3.5.3

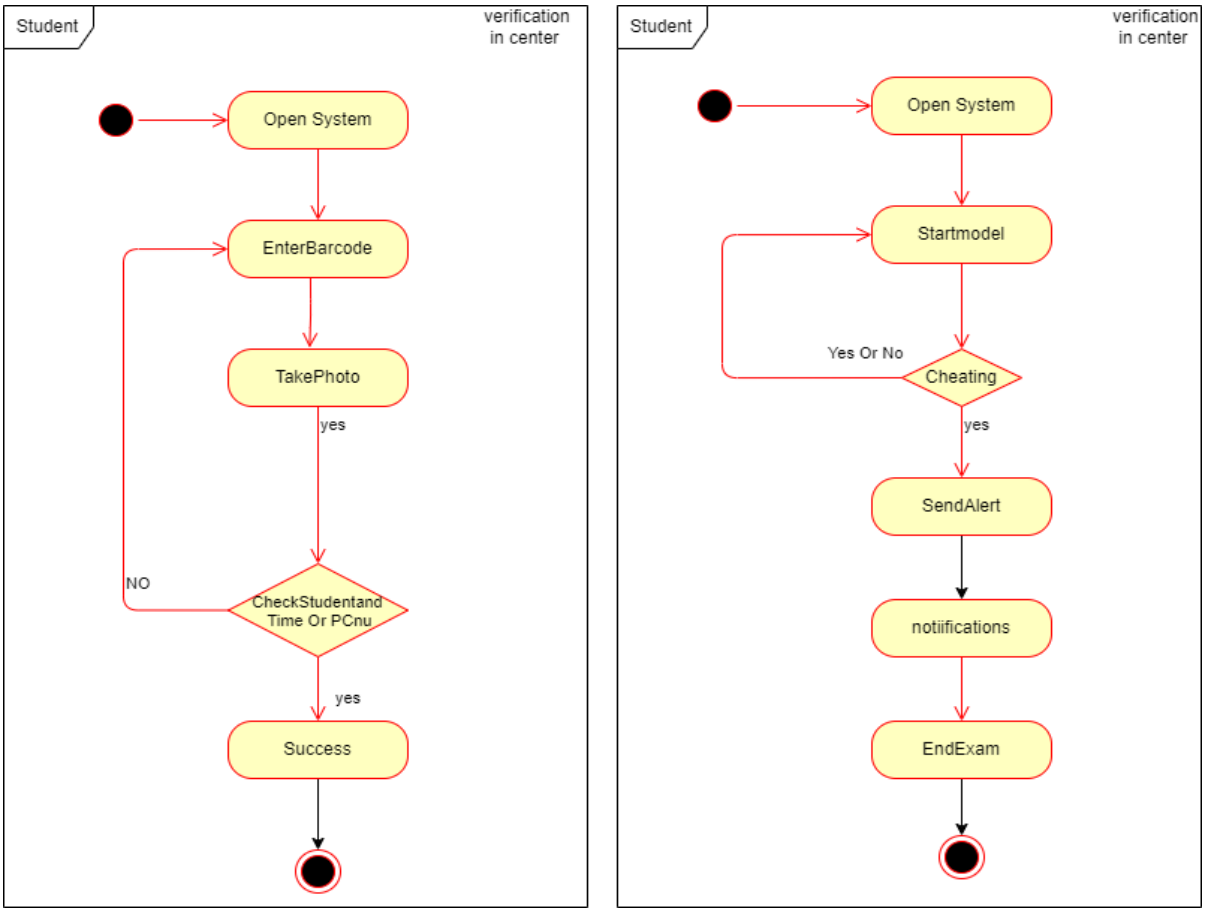


(figure 14)

Activity Diagram 3.5.4

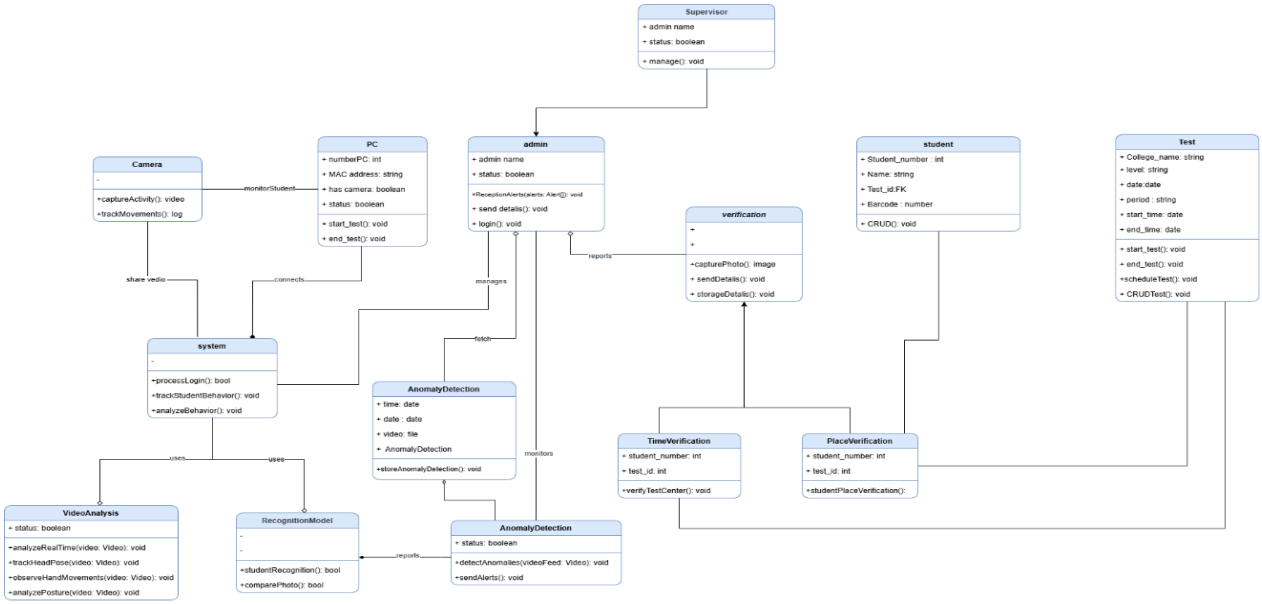


(figure 15)



(figure 16)

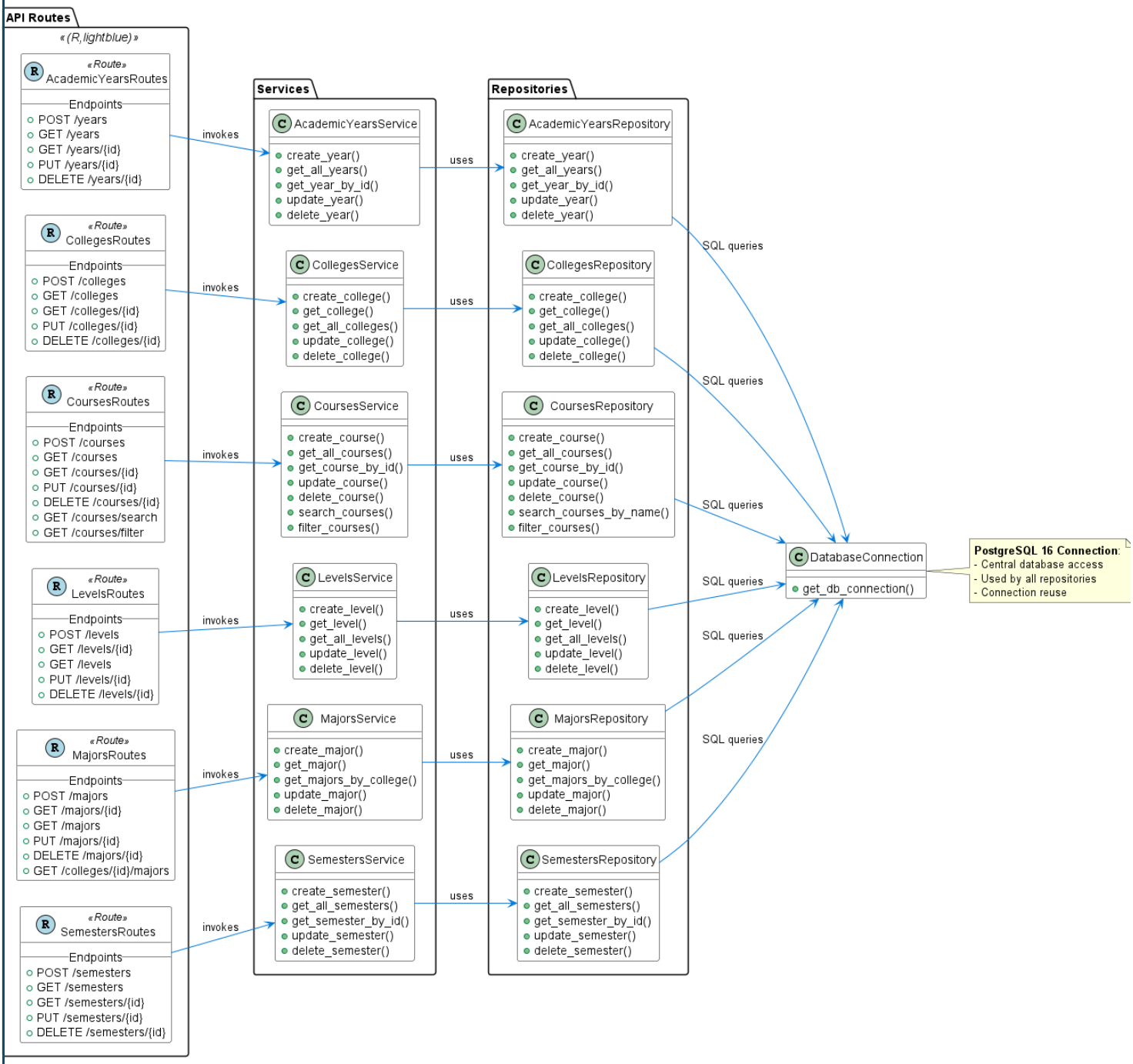
Class Diagram 3.5.5



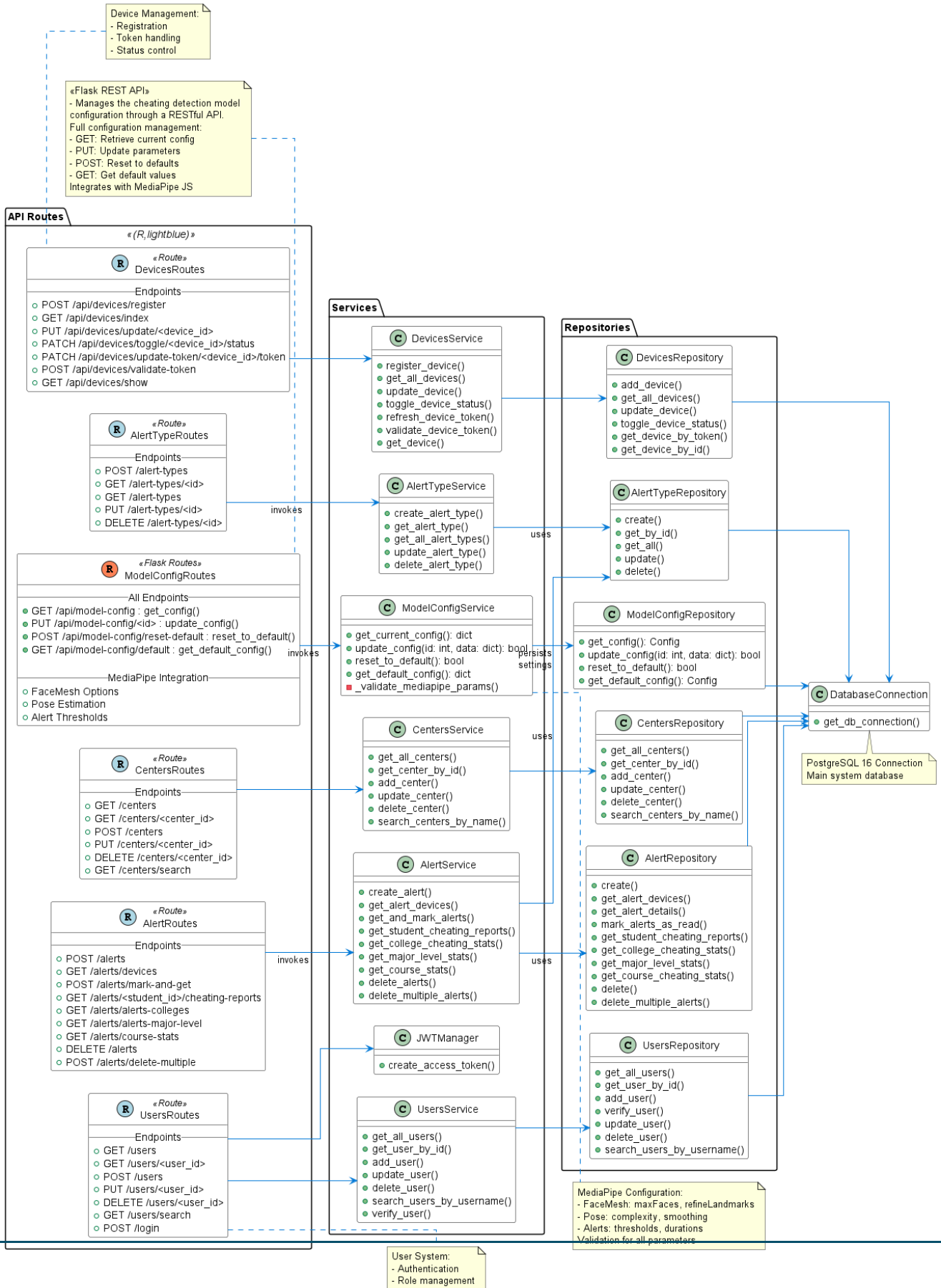
(figure 17)

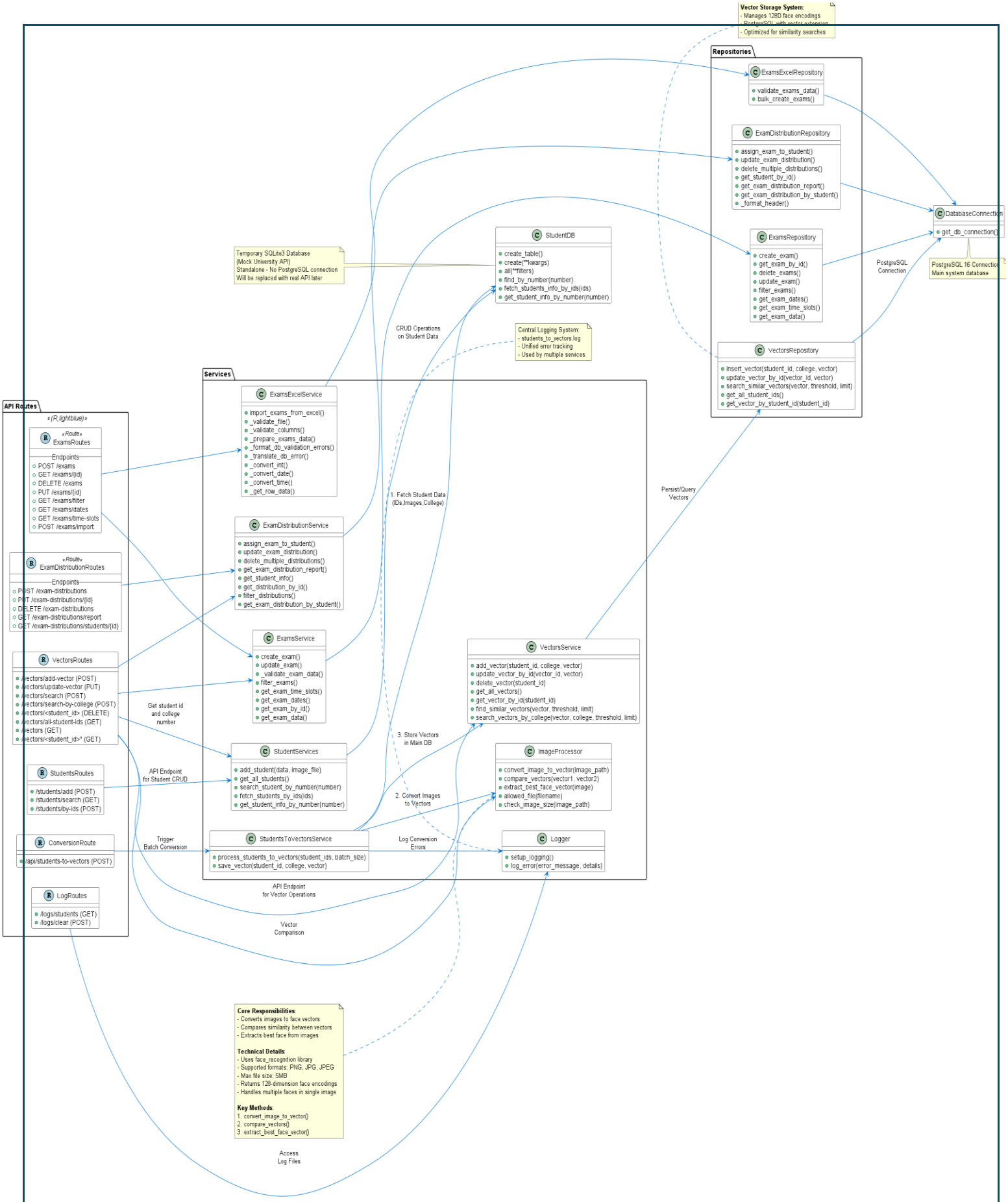
Class Diagram

(figure 18)



(figure 19)

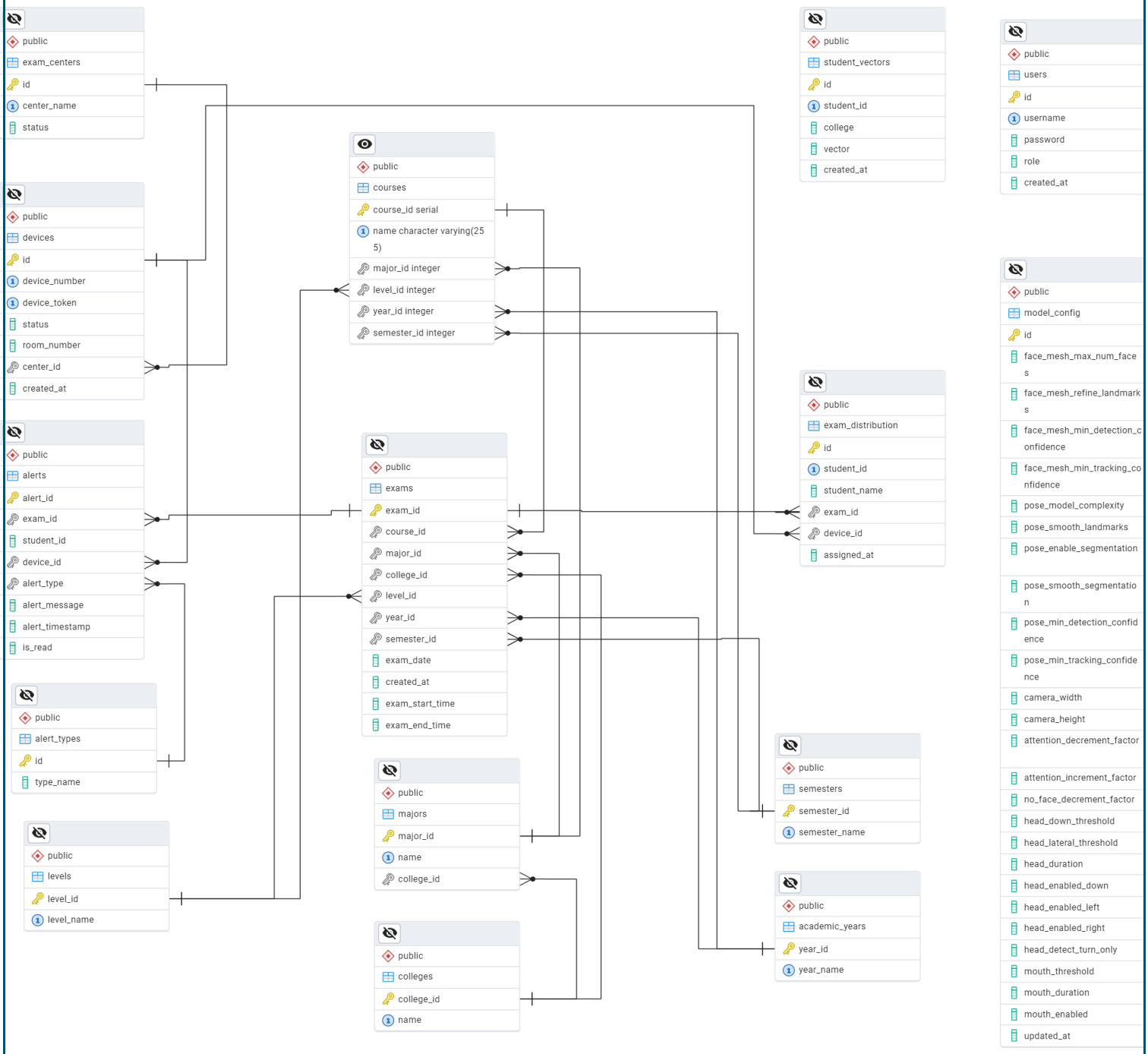

Device Management:
- Registration
- Token handling
- Status control
«Flask REST API»
- Manages the cheating detection model configuration through a RESTful API.
Full configuration management:
- GET: Retrieve current config
- PUT: Update parameters
- POST: Reset to defaults
- GET: Get default values
Integrates with MediaPipe JS
API Routes
«(R,lightblue)»
«Route»
DevicesRoutes
Endpoints
POST /api/devices/register
GET /api/devices/index
PUT /api/devices/update/<device_id>
PATCH /api/devices/toggle/<device_id>/status
PATCH /api/devices/update-token/<device_id>/token
POST /api/devices/validate-token
GET /api/devices/show
«Route»
AlertTypeRoutes
Endpoints
POST /alert-types
GET /alert-types/<id>
GET /alert-types
PUT /alert-types/<id>
DELETE /alert-types/<id>
invokes
«Flask Routes»
ModelConfigRoutes
All Endpoints
GET /api/model-config : get_config()
PUT /api/model-config/<id> : update_config()
POST /api/model-config/reset-default : reset_to_default()
GET /api/model-config/default : get_default_config()
MediaPipe Integration
FaceMesh Options
Pose Estimation
Alert Thresholds
invokes
«Route»
CentersRoutes
Endpoints
GET /centers
GET /centers/<center_id>
POST /centers
PUT /centers/<center_id>
DELETE /centers/<center_id>
GET /centers/search
«Route»
AlertRoutes
Endpoints
POST /alerts
GET /alerts/devices
POST /alerts/mark-and-get
GET /alerts/<student_id>/cheating-reports
GET /alerts/alerts-colleges
GET /alerts/alerts-major-level
GET /alerts/course-stats
DELETE /alerts
POST /alerts/delete-multiple
invokes
«Route»
UsersRoutes
Endpoints
GET /users
GET /users/<user_id>
POST /users
PUT /users/<user_id>
DELETE /users/<user_id>
GET /users/search
POST /login
Services
DevicesService
register_device()
get_all_devices()
update_device()
toggle_device_status()
refresh_device_token()
validate_device_token()
get_device()
AlertTypeService
create_alert_type()
get_alert_type()
get_all_alert_types()
update_alert_type()
delete_alert_type()
uses
ModelConfigService
get_current_config(): dict
update_config(id: int, data: dict): bool
reset_to_default(): bool
get_default_config(): dict
_validate_mediapipe_params()
persists settings
CentersService
get_all_centers()
get_center_by_id()
add_center()
update_center()
delete_center()
search_centers_by_name()
uses
AlertService
create_alert()
get_alert_devices()
get_and_mark_alerts()
get_student_cheating_reports()
get_college_cheating_stats()
get_major_level_stats()
get_course_stats()
delete_alerts()
delete_multiple_alerts()
uses
JWTManager
create_access_token()
UsersService
get_all_users()
get_user_by_id()
add_user()
update_user()
delete_user()
search_users_by_username()
verify_user()
Repositories
DevicesRepository
add_device()
get_all_devices()
update_device()
toggle_device_status()
get_device_by_token()
get_device_by_id()
AlertTypeRepository
create()
get_by_id()
get_all()
update()
delete()
ModelConfigRepository
get_config(): Config
update_config(id: int, data: dict): bool
reset_to_default(): bool
get_default_config(): Config
CentersRepository
get_all_centers()
get_center_by_id()
add_center()
update_center()
delete_center()
search_centers_by_name()
AlertRepository
create()
get_alert_devices()
get_alert_details()
mark_alerts_as_read()
get_student_cheating_reports()
get_college_cheating_stats()
get_major_level_stats()
get_course_cheating_stats()
delete()
delete_multiple_alerts()
UsersRepository
get_all_users()
get_user_by_id()
add_user()
verify_user()
update_user()
delete_user()
search_users_by_username()
DatabaseConnection
get_db_connection()
PostgreSQL 16 Connection
Main system database
MediaPipe Configuration:
- FaceMesh: maxFaces, refineLandmarks
- Pose: complexity, smoothing
- Alerts: thresholds, durations
Validation for all parameters
User System:
- Authentication
- Role management


Vector Storage System:
- Manages 128D face encodings
- PostgreSQL with vector extension
- Optimized for similarity searches
Repositories
ExamsExcelRepository
validate_exams_data()
bulk_create_exams()
ExamDistributionRepository
assign_exam_to_student()
update_exam_distribution()
delete_multiple_distributions()
get_student_by_id()
get_exam_distribution_report()
get_exam_distribution_by_student()
_format_header()
DatabaseConnection
get_db_connection()
ExamsRepository
create_exam()
get_exam_by_id()
delete_exams()
update_exam()
filter_exams()
get_exam_dates()
get_exam_time_slots()
get_exam_data()
PostgreSQL Connection
PostgreSQL 16 Connection
Main system database
Temporary SQLite3 Database
(Mock University API)
Standalone - No PostgreSQL connection
Will be replaced with real API later
StudentDB
create_table()
create(**kwargs)
all(**filters)
find_by_number(number)
fetch_students_info_by_ids(ids)
get_student_info_by_number(number)
CRUD Operations
on Student Data
Central Logging System:
- students_to_vectors.log
- Unified error tracking
- Used by multiple services
VectorsRepository
insert_vector(student_id, college, vector)
update_vector_by_id(vector_id, vector)
search_similar_vectors(vector, threshold, limit)
get_all_student_ids()
get_vector_by_student_id(student_id)
Persist/Query
Vectors
Services
ExamsExcelService
import_exams_from_excel()
_validate_file()
_validate_columns()
_prepare_exams_data()
_format_db_validation_errors()
_translate_db_error()
_convert_int()
_convert_date()
_convert_time()
_get_row_data()
1. Fetch Student Data
(IDs,Images,College)
ExamDistributionService
assign_exam_to_student()
update_exam_distribution()
delete_multiple_distributions()
get_exam_distribution_report()
get_student_info()
get_distribution_by_id()
filter_distributions()
get_exam_distribution_by_student()
ExamsService
create_exam()
update_exam()
_validate_exam_data()
filter_exams()
get_exam_time_slots()
get_exam_dates()
get_exam_by_id()
get_exam_data()
VectorsService
add_vector(student_id, college, vector)
update_vector_by_id(vector_id, vector)
delete_vector(student_id)
get_all_vectors()
get_vector_by_id(student_id)
find_similar_vectors(vector, threshold, limit)
search_vectors_by_college(vector, college, threshold, limit)
3. Store Vectors
in Main DB
StudentServices
add_student(data, image_file)
get_all_students()
search_student_by_number(number)
fetch_students_by_ids(ids)
get_student_info_by_number(number)
ImageProcessor
convert_image_to_vector(image_path)
compare_vectors(vector1, vector2)
extract_best_face_vector(image)
allowed_file(filename)
check_image_size(image_path)
2. Convert Images
to Vectors
StudentsToVectorsService
process_students_to_vectors(student_ids, batch_size)
save_vector(student_id, college, vector)
Log Conversion
Errors
Logger
setup_logging()
log_error(error_message, details)
API Endpoint
for Vector Operations
Vector
Comparison
API Routes
«(R,lightblue)»
«Route»
ExamsRoutes
Endpoints
POST /exams
GET /exams/{id}
DELETE /exams
PUT /exams/{id}
GET /exams/filter
GET /exams/dates
GET /exams/time-slots
POST /exams/import
«Route»
ExamDistributionRoutes
Endpoints
POST /exam-distributions
PUT /exam-distributions/{id}
DELETE /exam-distributions
GET /exam-distributions/report
GET /exam-distributions/students/{id}
VectorsRoutes
/vectors/add-vector (POST)
/vectors/update-vector (PUT)
/vectors/search (POST)
/vectors/search-by-college (POST)
/vectors/<student_id> (DELETE)
/vectors/all-student-ids (GET)
/vectors (GET)
/vectors/<student_id>* (GET)
Get student id
and college
number
StudentsRoutes
/students/add (POST)
/students/search (GET)
/students/by-ids (POST)
API Endpoint
for Student CRUD
ConversionRoute
/api/students-to-vectors (POST)
Trigger
Batch Conversion
LogRoutes
/logs/students (GET)
/logs/clear (POST)
Core Responsibilities:
- Converts images to face vectors
- Compares similarity between vectors
- Extracts best face from images
Technical Details:
- Uses face_recognition library
- Supported formats: PNG, JPG, JPEG
- Max file size: 5MB
- Returns 128-dimension face encodings
- Handles multiple faces in single image
Key Methods:
1. convert_image_to_vector()
2. compare_vectors()
3. extract_best_face_vector()
Access
Log Files

(figure 20)

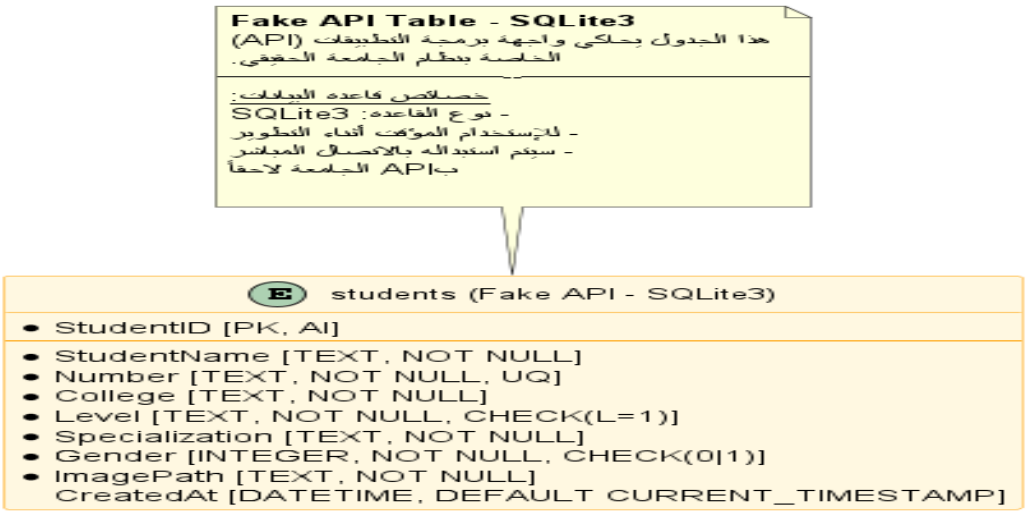
Relational Schema for the PostgreSQL 16

(figure 21)



Relational Schema for Temporary SQLite3 Database

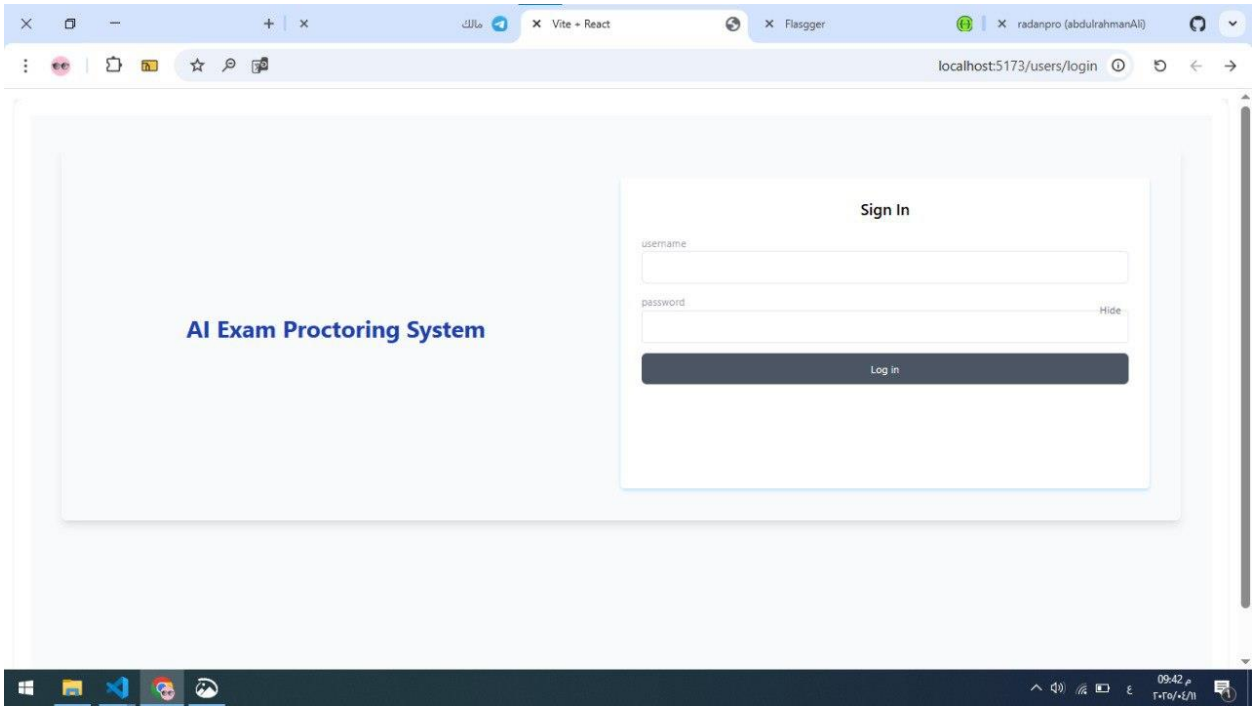
(figure 22)



الفصل الرابع
تصميم المشروع

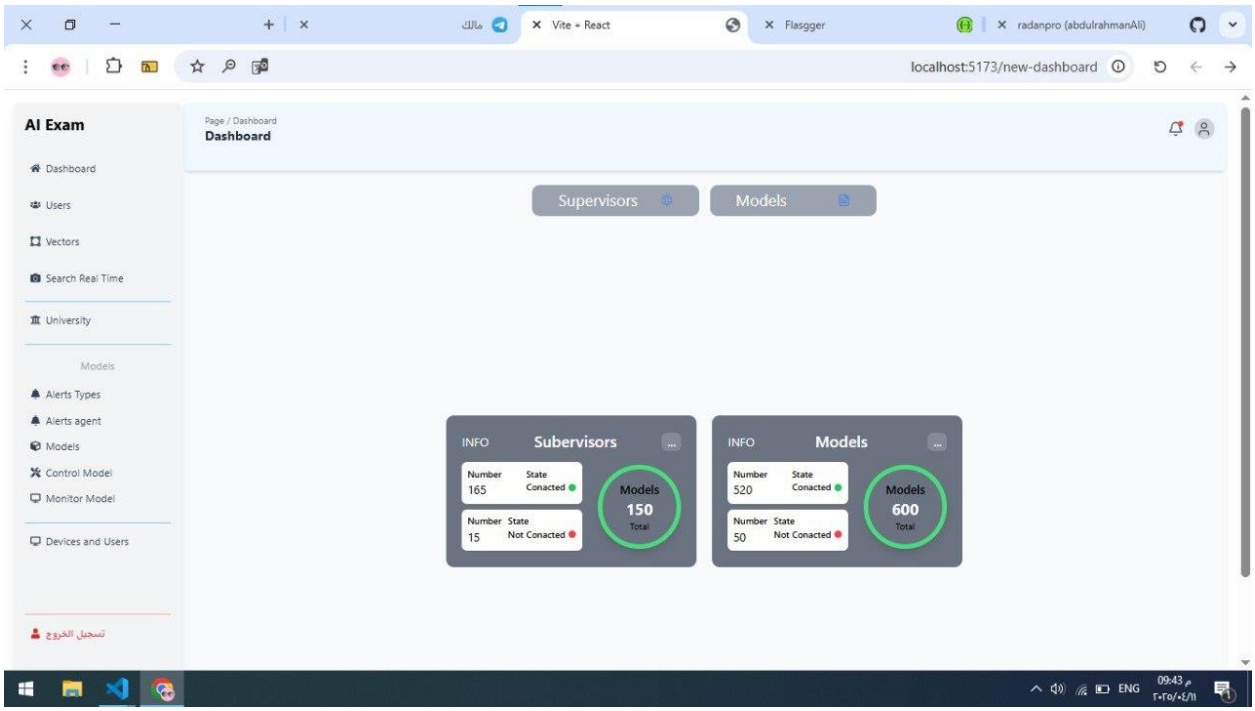
4.1 المقدمة

سنعرض هنا واجهات النظام وعمل كل واجهة

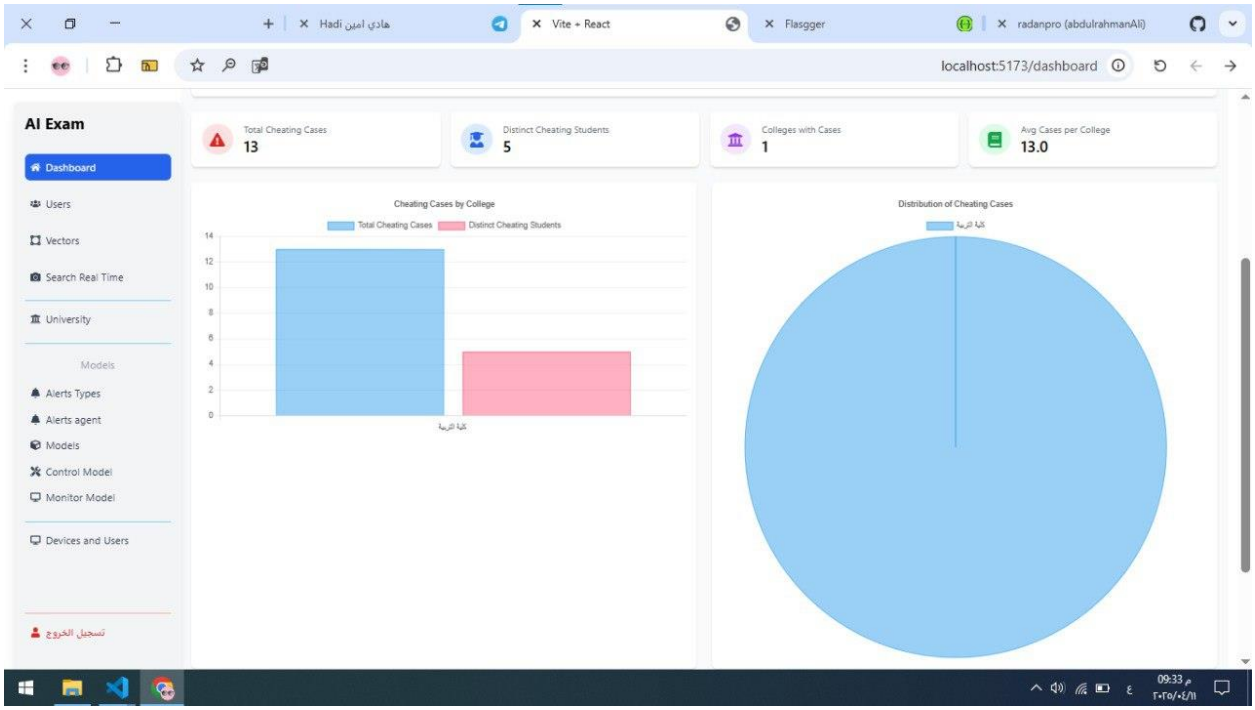


4-1 .شاشة تسجيل الدخول

- في هذه الواجهة يتم تسجيل الدخول للنظام سواء عبر المشرف او المراقب.

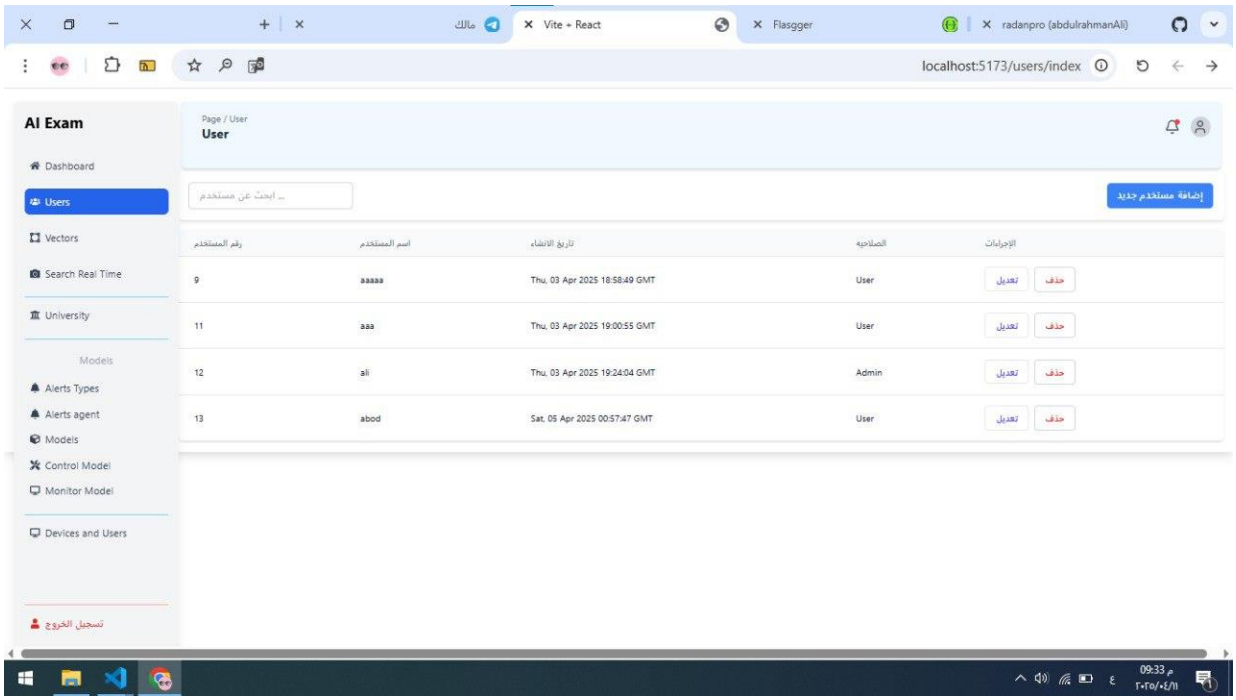


- في هذه الواجهة يتم عرض معلومات الأجهزة والمراقبين من حالة الاتصال وعدم الاتصال وكذألك العدد



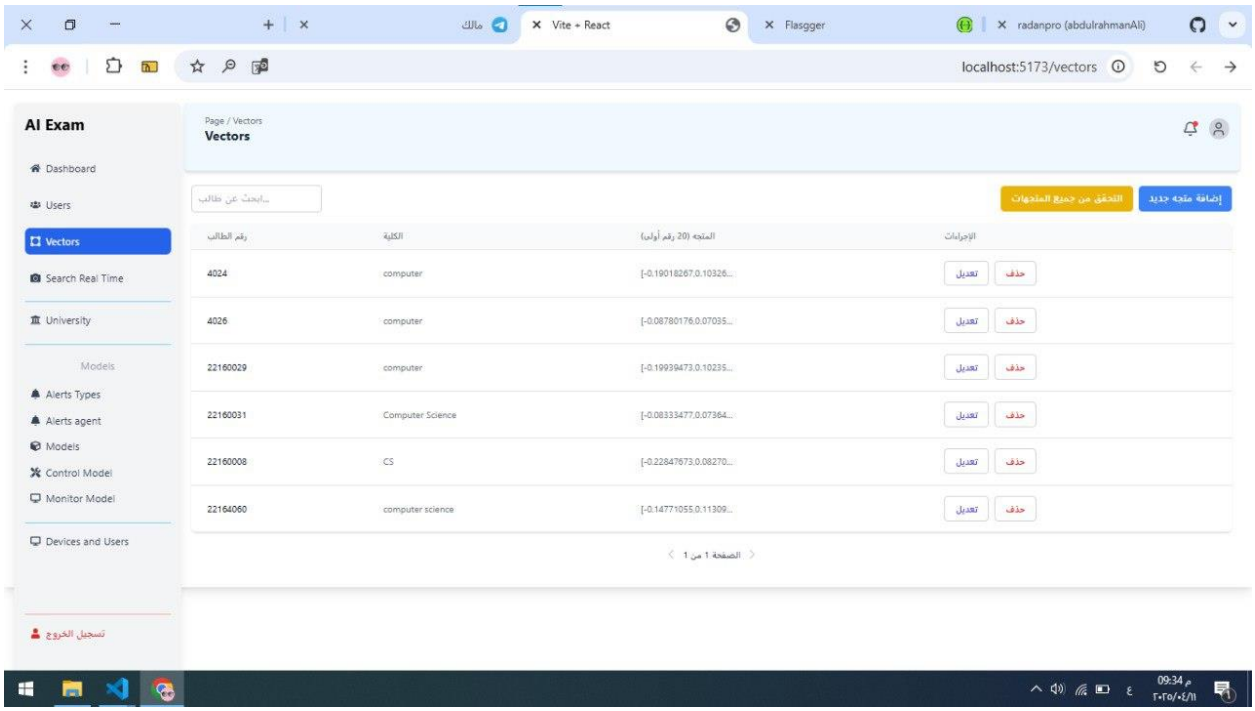
4.2 شاشة التحكم الرئيسية

- في هذه الواجهة يتم عرض معلومات عن المراكز الامتحانية والطلاب وحالات الغش



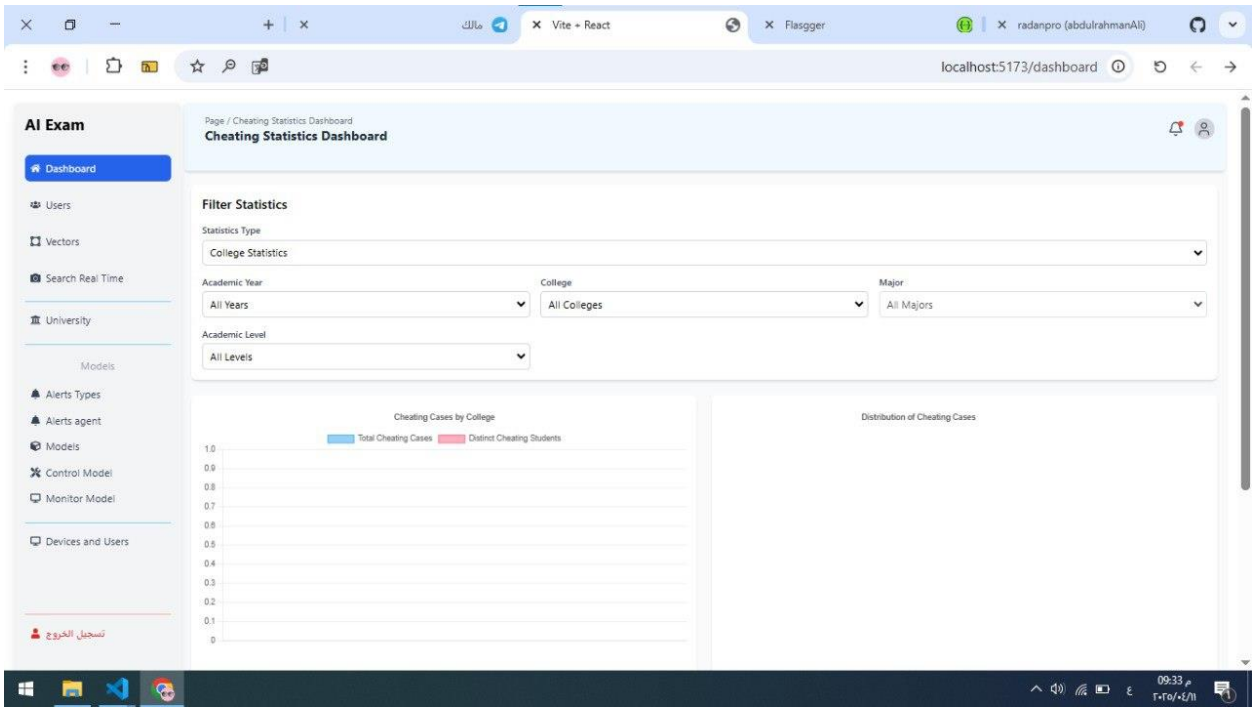
4.3 شاشة المستخدمين في النظام

- في هذه الواجهة يتم عرض واضافة المراقبين او المشرفين في النظام والتحكم في صلاحياتهم.

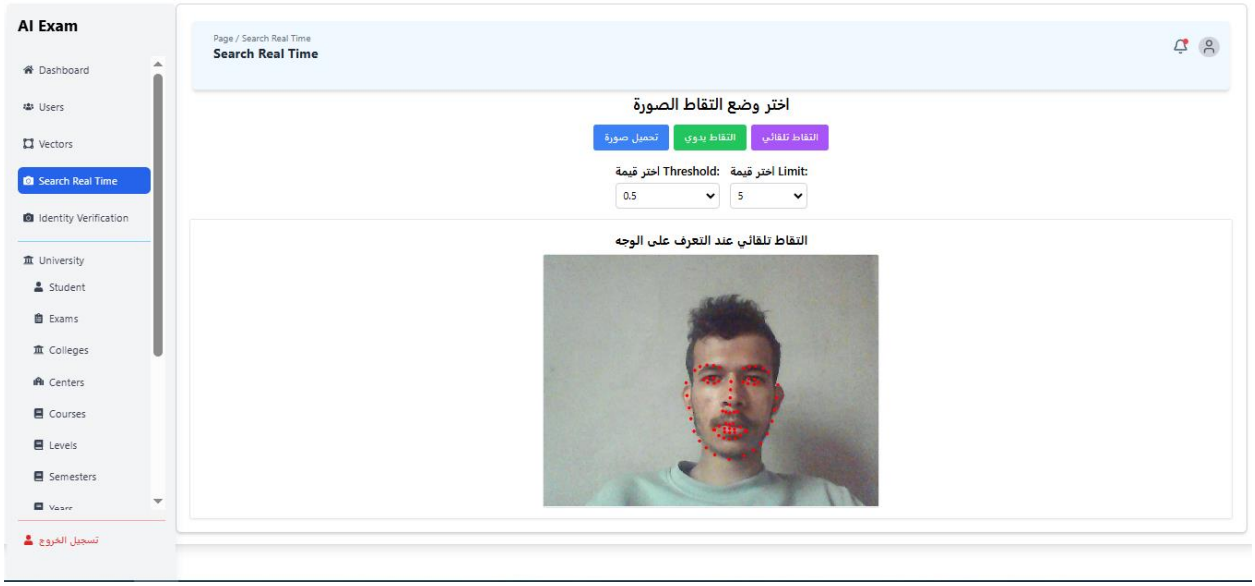


4.4 شاشة صور الطلاب ك متجهات

في هذه الواجهة يتم عرض واضافة صور الطلاب بشكل فيكتور لكي تتم المطابقة اثناء التحقق بشكل عملي وسريع

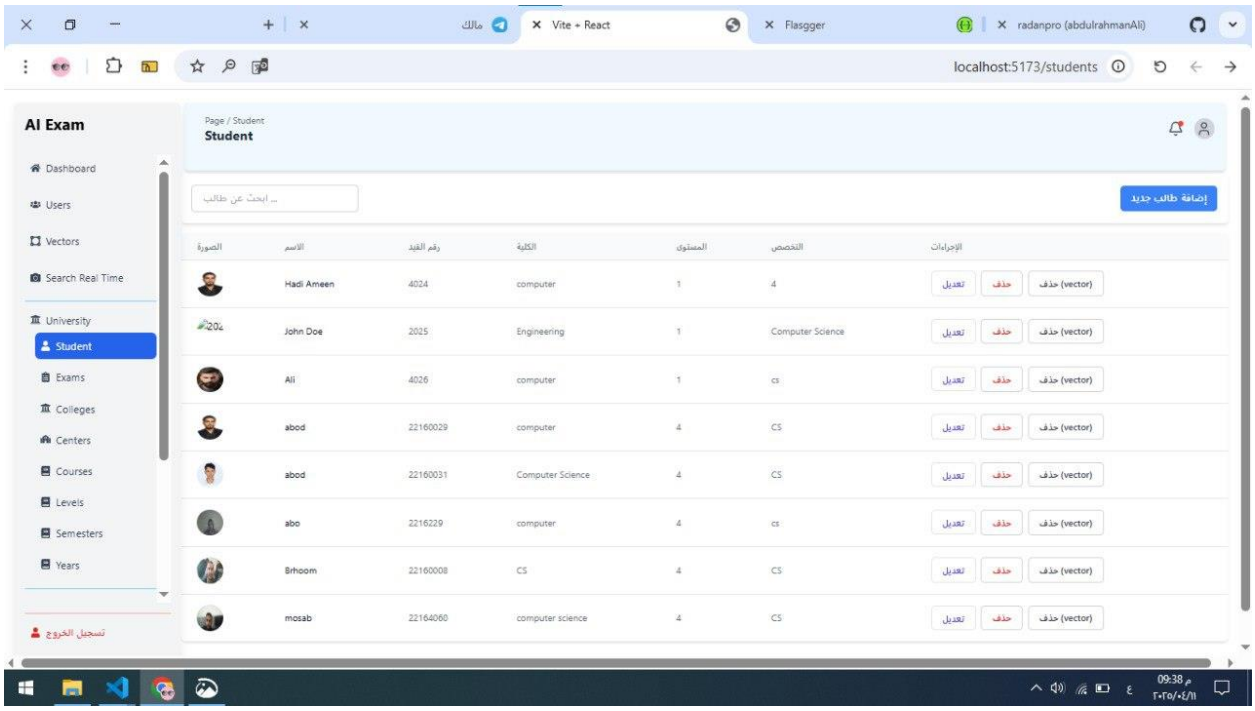


- Filter Statistics Dashboard
- في هذه الواجهة يتم عمل فلترة في الية مطابقة الصور من خلال تحديد سنة او كلية او تخصص معين



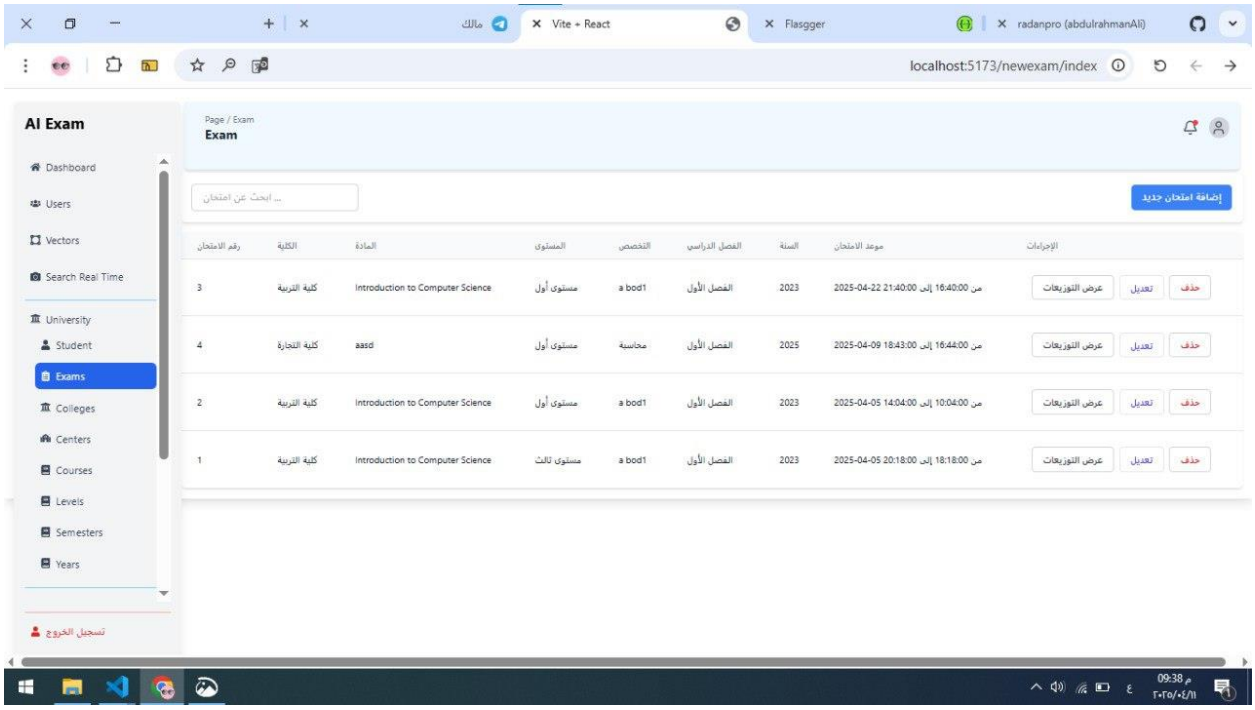
4.5شاشة البحث والمطابقة

- هنا يتم مطابقة الصورة واكتشاف هوية الطالب



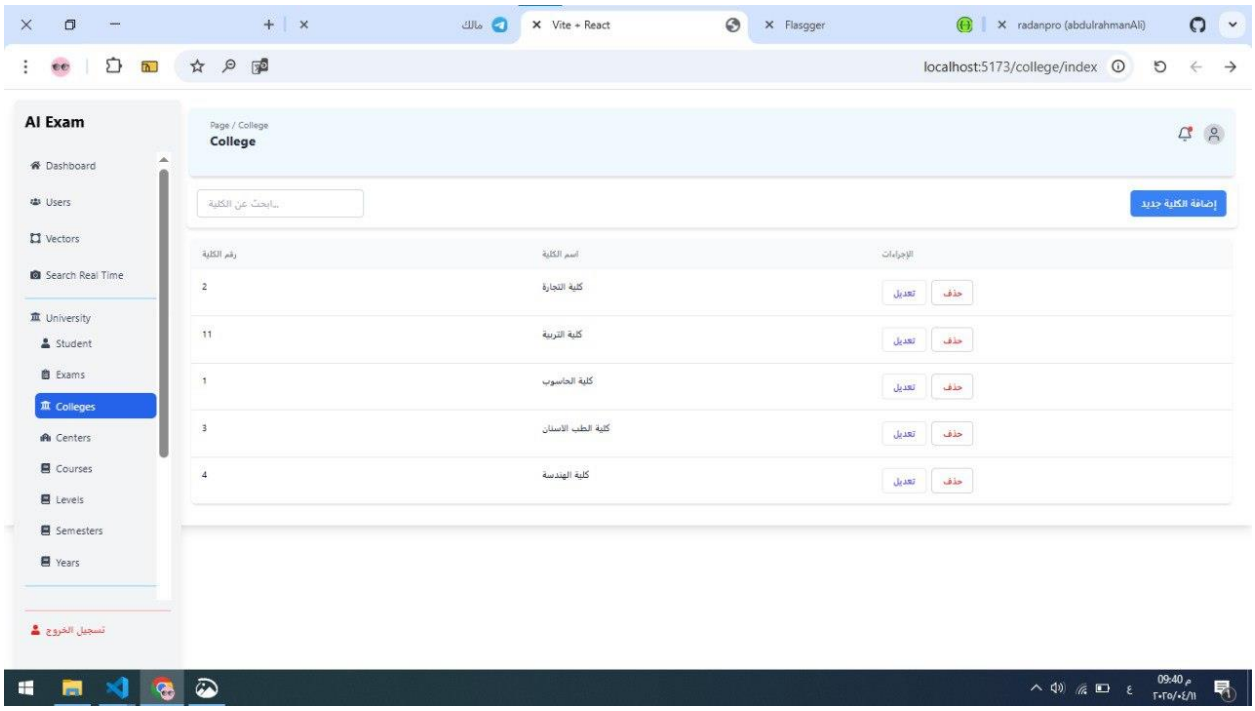
4.6 شاشة الطلاب الموجودين في النظام

- في هذه الواجهة قائمة الطلاب المخزنة ويمكن إضافة وحذف الطلاب منها



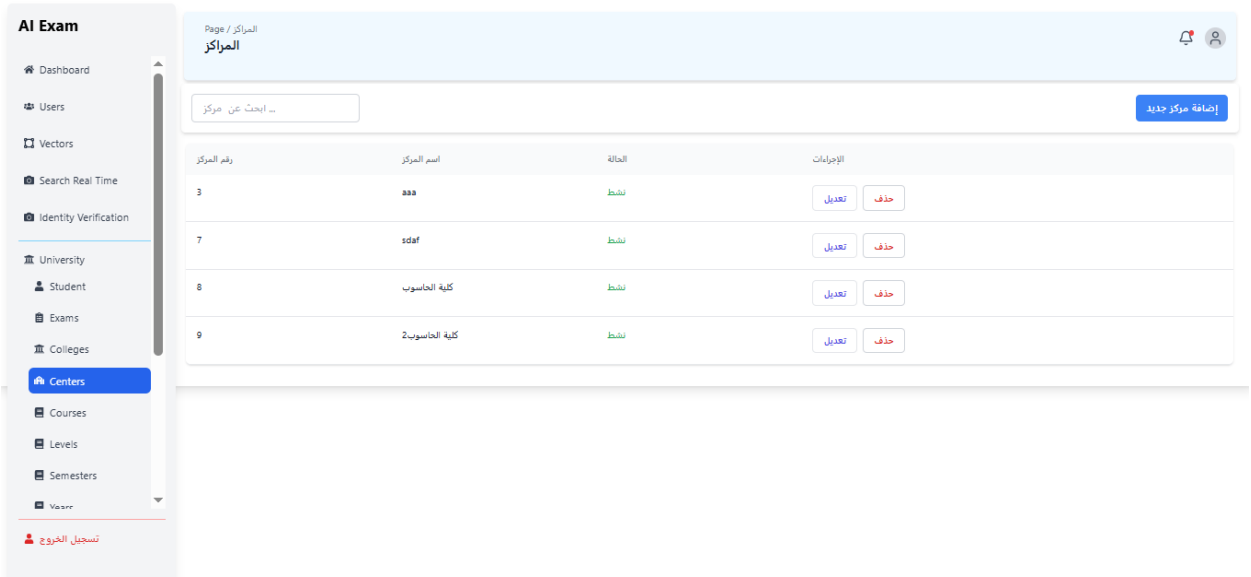
4.7 شاشة الاختبارات الرئيسية في النظام

- في هذه الواجهة قائمة توزيعات الاختبارات مع المراكز التي يتم اصدارها في كل فترة اختبارات



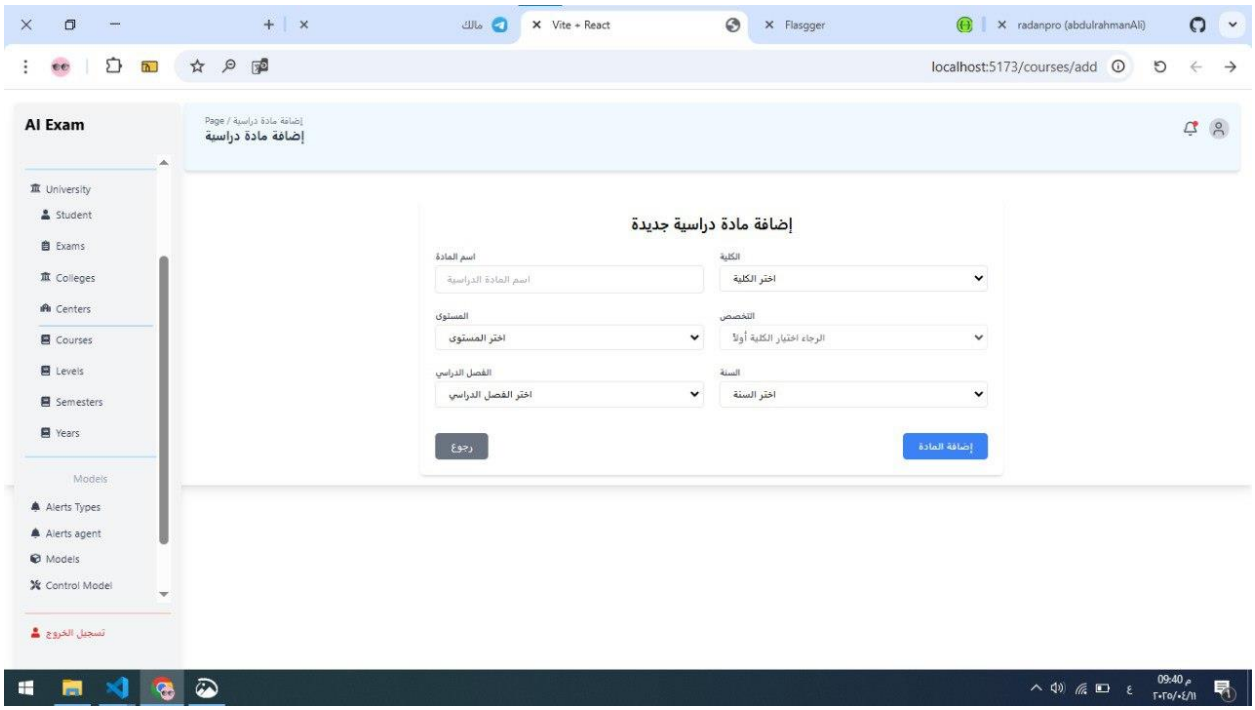
4.8 شاشة الكليات

- في هذه الواجهة يتم إضافة الكليات لقاعدة البيانات



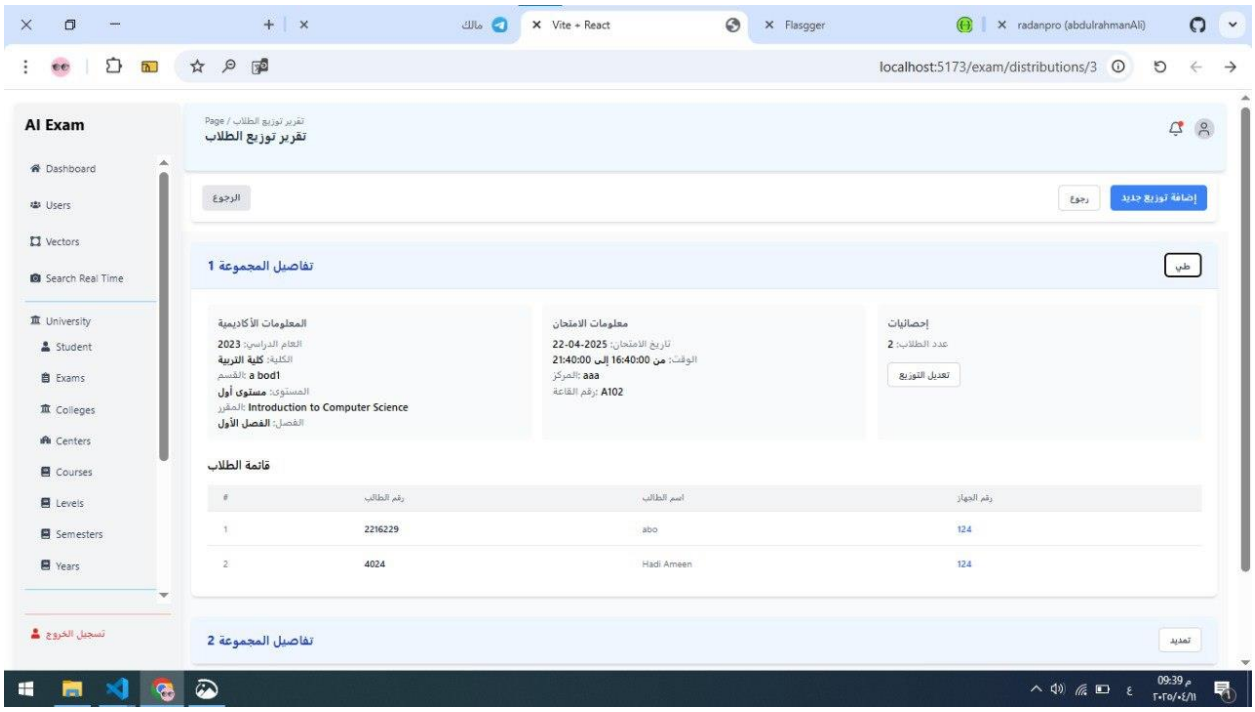
4.9 شاشة المراكز الامتحانية

- وفي هذه الواجهة إضافة مراكز الاختبارات

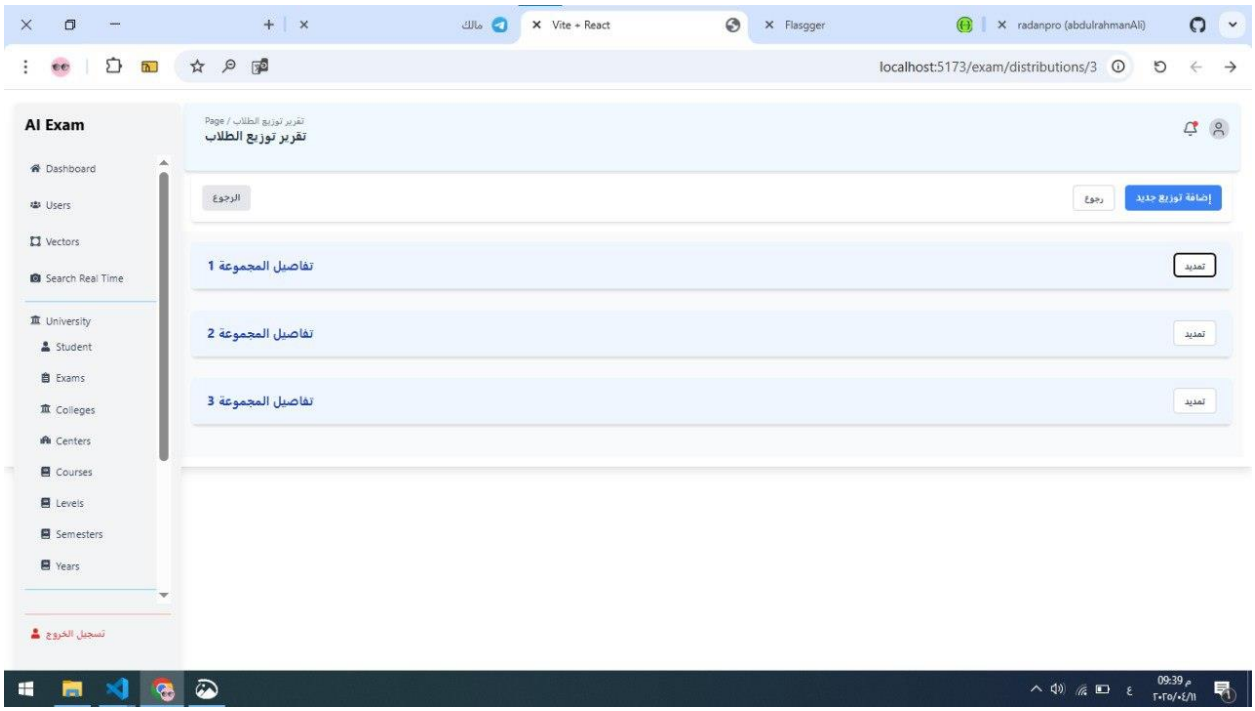


4.10 شاشة المقررات الدراسية والمواد الدراسية

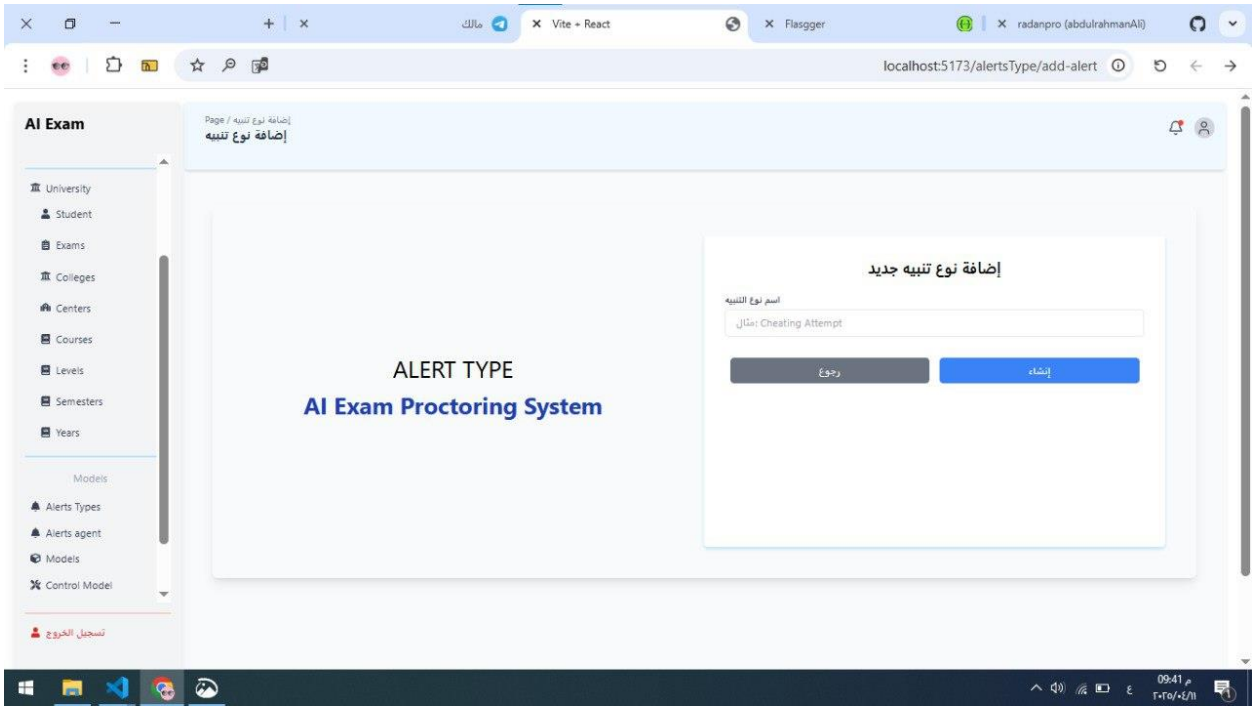
- وفي هذه الواجة إضافة مادة دراسية لقاعدة البيانات



- **4.11 شاشة توزيع الامتحانات**
- هنا في هذه الواجهة يتم إضافة توزيع الطلاب على مستوى المركز ورقم الجهاز

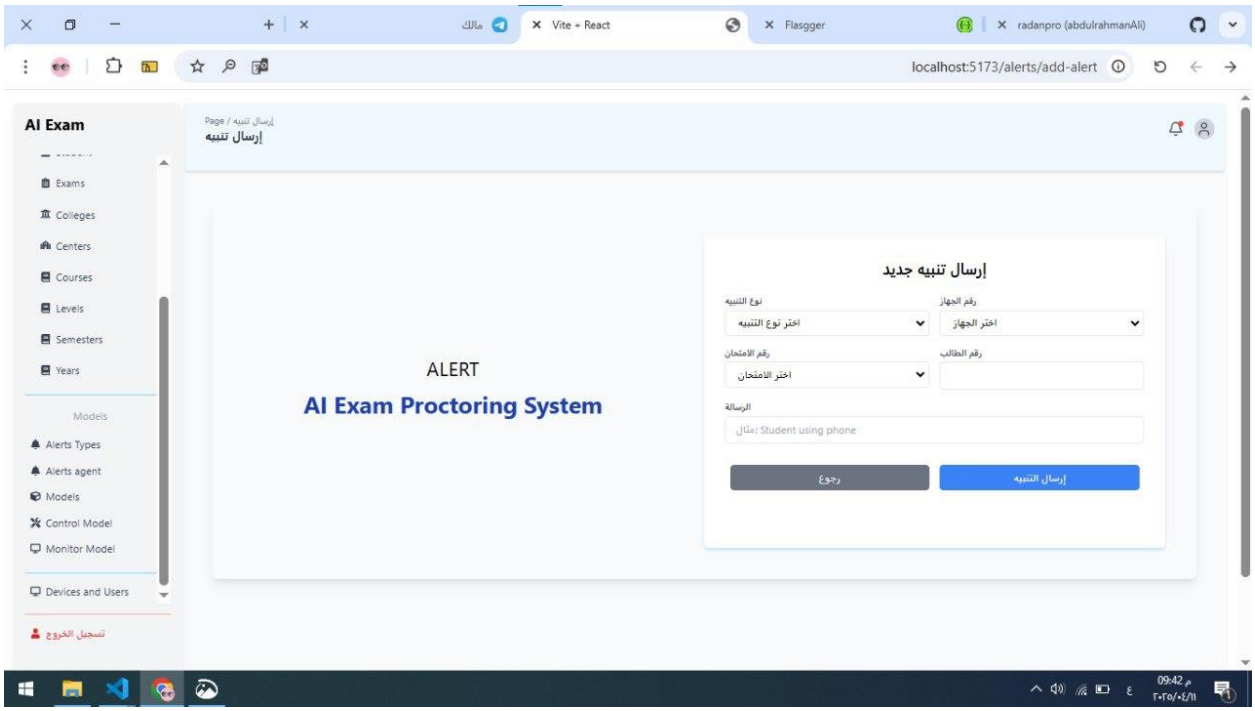


4.12 **تابع شاشة توزيع الامتحانات**

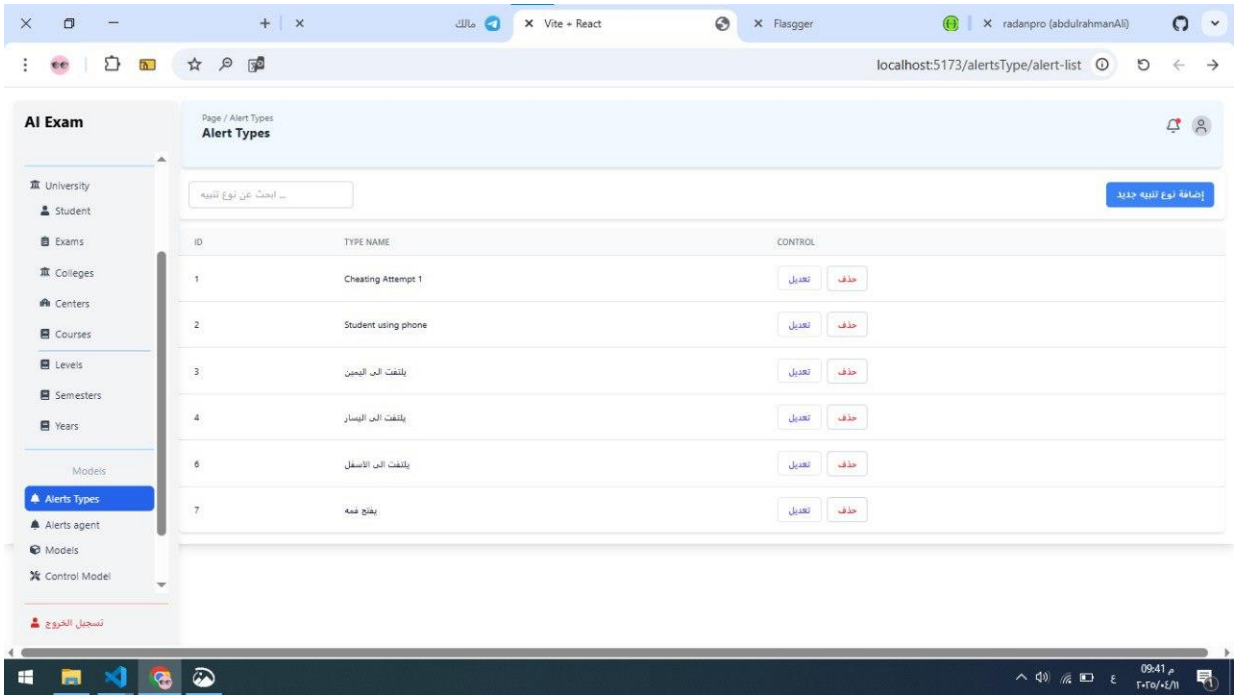


4.13 شاشة إضافة تنبيهات

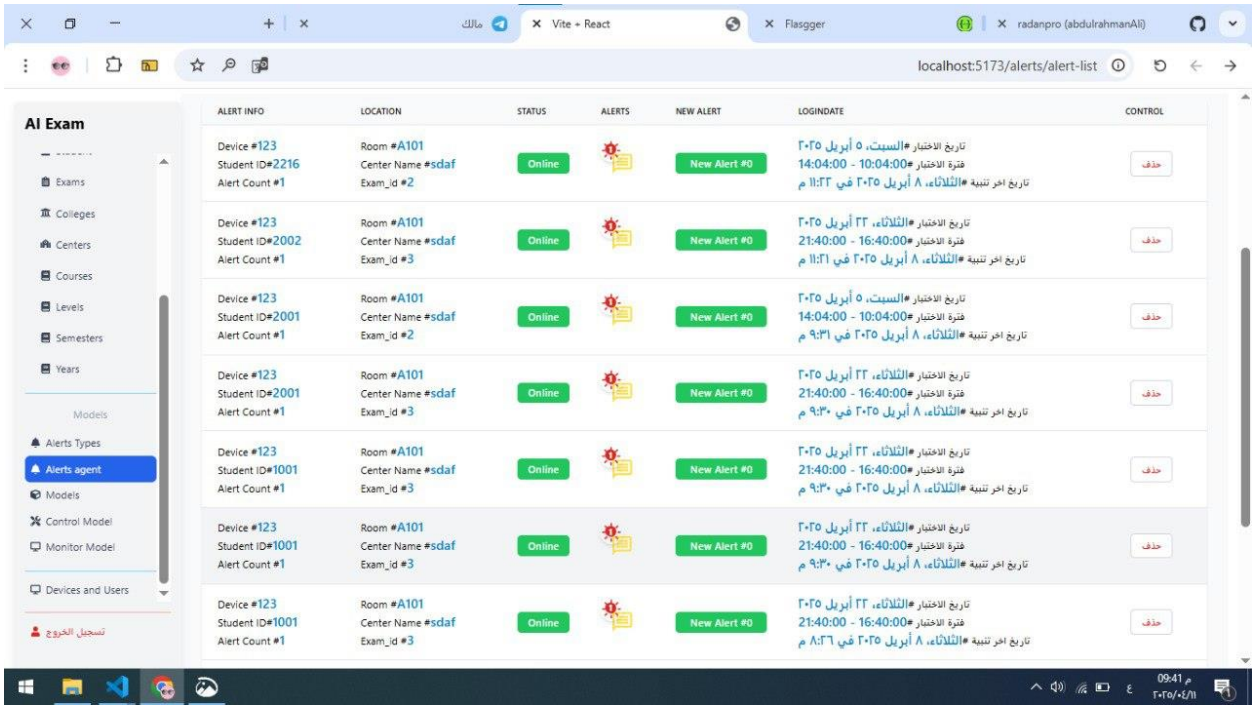
- في هذه الواجهة إمكانية انشاء تنبية حسب الاحتياج والسيناريو المقرر من مشرف النظام



4.14 تابع شاشة إضافة تنبيهات

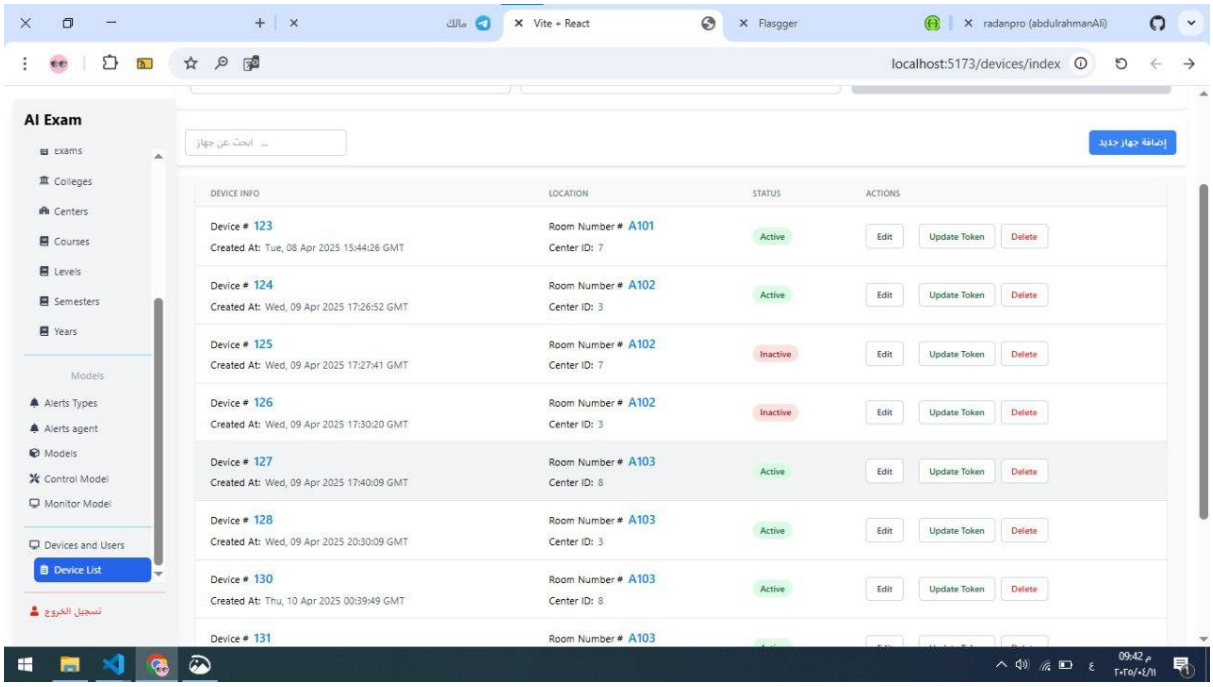


4.15.1 شاشة انواع تنبيهات النظام



4.16 شاشة تنبيهات النظام

- في هذه الواجهة قائمة التنبيهات القادمة من الأجهزة سواء من مطابقة الصور او اكتشاف سلوكيات الغش

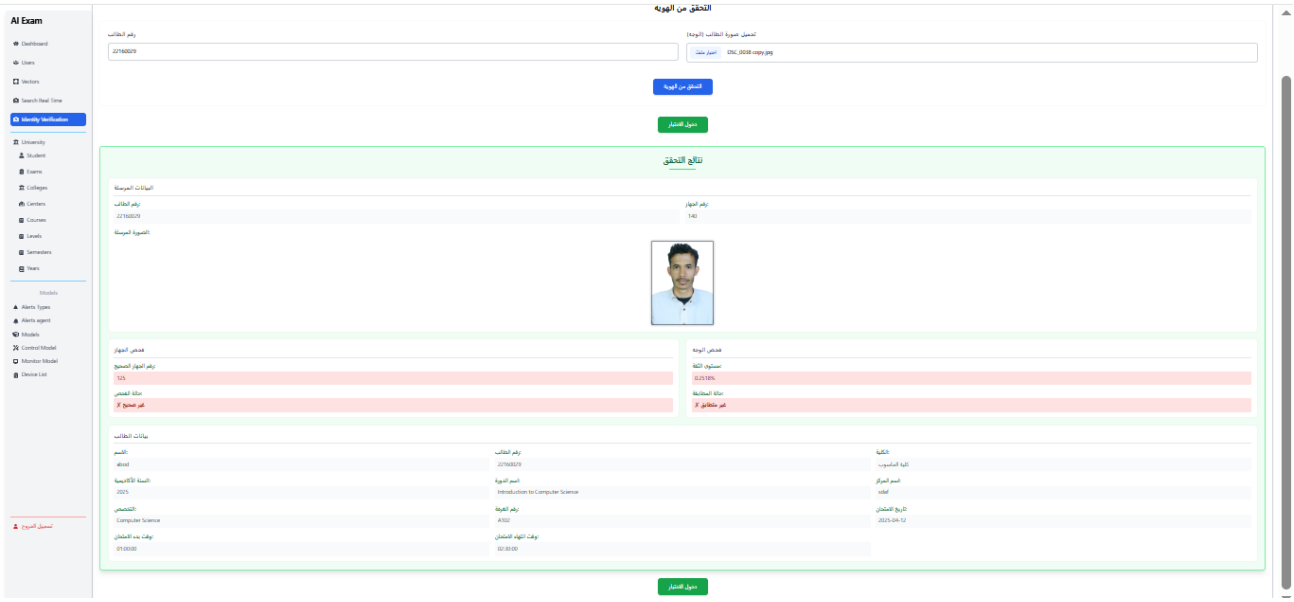


4-17 شاشة الأجهزة في النظام

- هنا يتم إضافة الأجهزة التي في مراكز الامتحان وربطها بنظام

4-18 شاشة اكتشاف الطلاب والتحقق من هوية

هنا يتم اكتشاف الصورة ومطابقتها وارجاع بيانات المطابقة



4-19 شاشة اكتشاف الطالب في جهاز الامتحان

- في هذه الواجهة اكتشاف هوية الطالب هل مصرح له برقم الجهاز المسموح له باجراء اختبار المادة علية ام لا

الفصل الخامس
(التنفيذ والاختبار)

5.1 مقدمة

يهدف هذا الفصل إلى توثيق عملية تنفيذ نظام مراقبة الامتحانات باستخدام الذكاء الاصطناعي، بدءًا من بيئة التنفيذ وصولًا إلى استراتيجيات الاختبار والنتائج التي تم الحصول عليها. يعتبر هذا الفصل حاسمًا لتقييم جودة النظام وفعاليته في تحقيق الأهداف المرجوة، كما يوضح كيفية ترجمة متطلبات النظام وتصميمه إلى نظام عملي قابل للاستخدام.

5.2 بيئة التنفيذ

يصف هذا القسم البيئة التي تم فيها تطوير وتشغيل النظام، بما في ذلك المكونات المادية والبرمجية اللازمة لضمان الأداء الأمثل.

- **5.2.1 المكونات المادية**
  - **الخوادم :**
    - المعالج Intel Core i9-9900K :أو ما يعادله (لضمان قوة المعالجة وسرعة الاستجابة)
    - الذاكرة: 64 GB DDR4 RAM لضمان تعدد المهام وسرعة الوصول إلى البيانات
    - التخزين: 2TB NVMe SSD لضمان سرعة القراءة والكتابة والوصول إلى البيانات
  - **أجهزة المستخدمين (أجهزة الطلاب) :**
    - المعالج Intel Core i5 :الجيل ال أو AMD Ryzen 5 2600 أو أفضل
    - الذاكرة: 8 GB DDR4 RAM أو أفضل
    - الكاميرا: كاميرا ويب بدقة 1080p ومعدل إطارات 30 fps لضمان جودة الفيديو اللازمة للتعرف على الوجه وتحليل السلوك
  - **الشبكة :**
    - نوع الشبكة LAN :لتوفير المرونة في الاتصال
    - سرعة الاتصال: 20 Mbps على الأقل لكل جهاز (لضمان تدفق الفيديو بسلاسة وتجنب التأخير)
- **5.2.2المكونات البرمجية**

- نظام التشغيل للخوادم Ubuntu Server 20.04 LTS :لضمان الاستقرار والأمان
- نظام التشغيل لأجهزة المستخدمين Windows 10 :أو macOS Big Sur أو أحدث (لتوفير التوافق)
- قاعدة البيانات PostgreSQL 16 :أو أحدث (لضمان الأمان والموثوقية وقابلية التوسع)
- إضافة pgvector إلى قاعدة البينات للبحث عن الفكترز المتشابهة
- لغة البرمجة للخادم Python 3.8 :أو أحدث (لقوة المكتبات وسهولة التطوير)
- إطار عمل الخادم flask :أو أحدث (لسرعة التطوير والأمان)
- لغة البرمجة للواجهةJavaScript :للتفاعل الديناميكي
- إطار عمل الواجهة React 17 :أو أحدث (لكفاءة الأداء وقابلية الصيانة)
- مكتبات الذكاء الاصطناعي **Media Pipe js** لتحليل الفيديو ومعالجة الصور والتعرف على الأنماط
- مكتبة face_recognition لتحويل الصور إلى فكترز ومقارنة الفكترز

5.3عملية التنفيذ

يصف هذا القسم الخطوات المتخذة لتطوير النظام، بدءًا من تطوير الواجهات وصولًا إلى تكامل المكونات المختلفة.

- **5.3.1تطوير الواجهات**

- تم تصميم واجهات المستخدم باستخدام Figma لضمان سهولة الاستخدام والتجربة البديهية. تم التركيز على تصميم بسيط وواضح يقلل من الارتباك ويسهل على المستخدمين (الطلاب والمراقبين) التفاعل مع النظام.
- تم استخدامHTML5 ، CSS3، و JavaScript لإنشاء الواجهات. تم استخدام React.jsكإطار عمل لتطوير الواجهات الديناميكية والتفاعلية.

- ### 5.3.2 الخوارزميات المستخدمة

  - تم تطوير خوارزمية التعرف على الوجه باستخدام مكتبة OpenCV و **MediaPipe** لتحقيق دقة عالية في التعرف على الطلاب من خلال مقارنة وجوههم بالصور المخزنة في قاعدة البيانات.

  - تم استخدام **MediaPipe** لتحليل السلوكيات المشبوهة باستخدام لتحليل تسلسل الإطارات في الفيديو واكتشاف الأنماط السلوكية التي قد تشير إلى الغش، مثل النظر المتكرر خارج الشاشة، أو التحدث، أو استخدام الهاتف. تم تدريب النموذج على مجموعة بيانات تتضمن أمثلة على السلوكيات المشبوهة والسلوكيات الطبيعية .

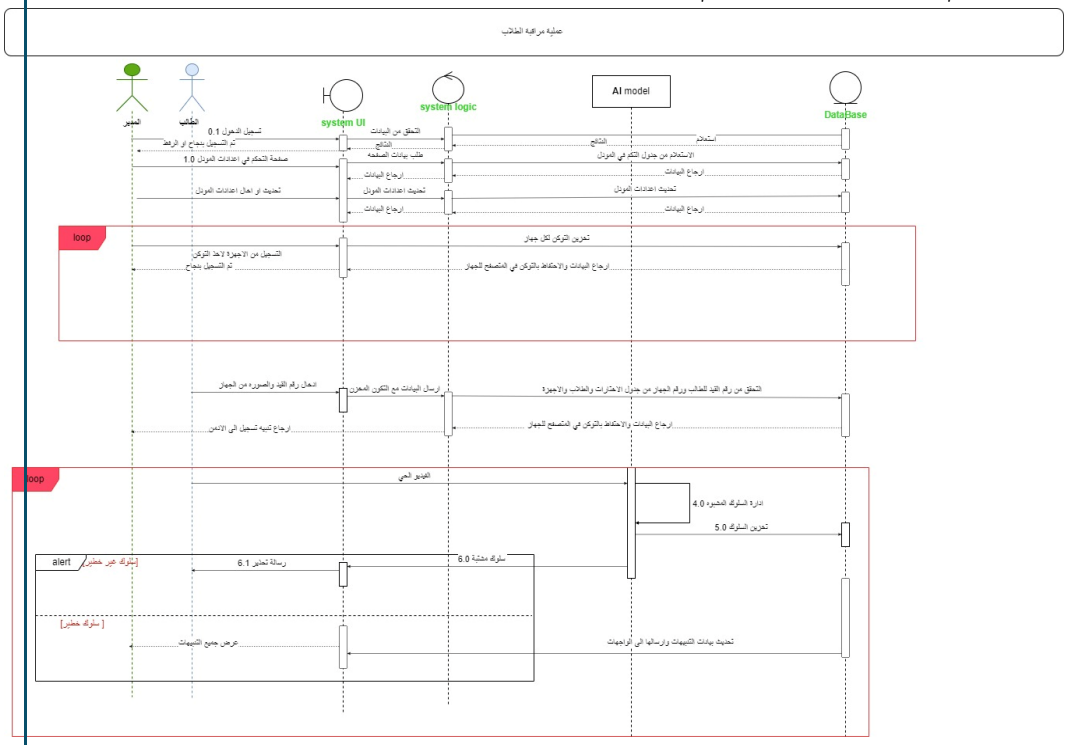
  - تم استخدام خوارزمية مكتبة face_recognition لتحويل الصور إلى فكترز ومقارنة الفكترز بجودة وسرعة عالية وموارد قليلة

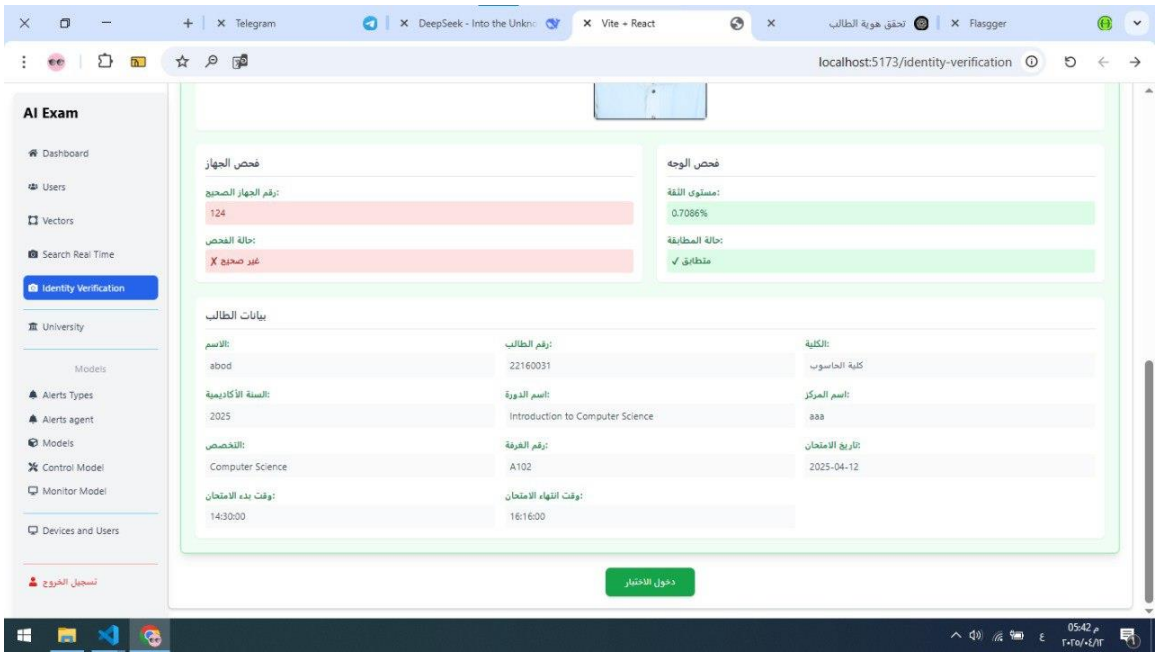
- **5.3.3ربط المكونات**

  - تم ربط واجهات المستخدم بقاعدة البيانات PostgreSQL باستخدام واجهات برمجة التطبيقات .RESTful (APIs) تم استخدام flask Framework لإنشاء هذه الواجهات، مما يسمح بتدفق البيانات بسلاسة بين الواجهة والخادم.

  - تم دمج خوارزميات الذكاء الاصطناعي مع النظام باستخدام خدمات الويب. يتم إرسال إطارات الفيديو من الواجهة إلى الخادم، حيث تتم معالجتها بواسطة خوارزميات الذكاء الاصطناعي، ويتم إرجاع النتائج (مثل التنبيهات حول السلوكيات المشبوهة) إلى الواجهة لعرضها على المراقبين .

    - مخطط يوضح كيف يتم إرسال الفيديو من كاميرا الطالب إلى الخادم، وكيف تتم معالجته، وكيف يتم إرسال التنبيهات إلى واجهة المراقب





- **5.3.4 تكامل النظام**

  - تم دمج جميع المكونات (الواجهات، الخادم، قاعدة البيانات، خوارزميات الذكاء الاصطناعي) لتكوين النظام الكامل. تم التركيز على ضمان التوافق والأداء السلس من خلال اختبار شامل للتكامل.
  - تم حل التحديات المتعلقة بتأخير معالجة الفيديو عن طريق تحسين الخوارزميات واستخدام المعالجة المتوازية. تم تحسين دقة التعرف على الوجه عن طريق تحسين الخوارزمية والقواعد
  - فيها.

5.4 استراتيجية الاختبار

يصف هذا القسم أنواع الاختبارات التي تم إجراؤها لضمان جودة النظام وفعاليته، والتحقق من أنه يلبي المتطلبات المحددة في الفصل الثالث.(SRS)

- **5.4.1أنواع الاختبارات**

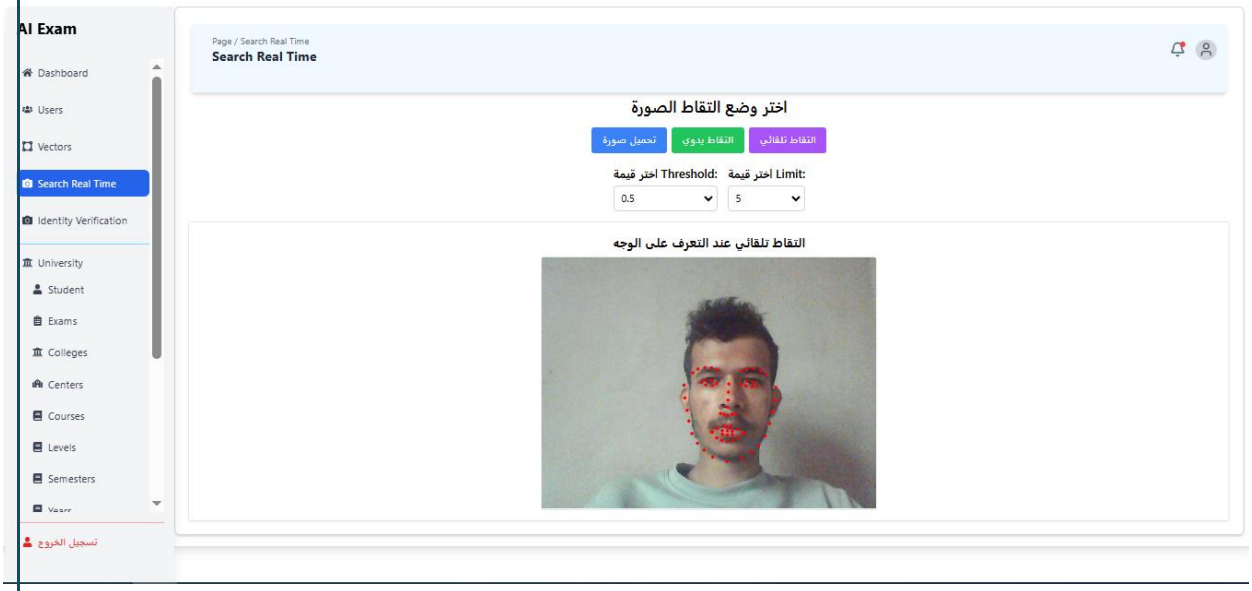
  - **اختبار الوحدة :(Unit Testing)** تم اختبار كل وحدة برمجية على حدة باستخدام PyTest للتأكد من أنها تعمل على النحو المنشود. تم كتابة اختبارات

للتحقق من صحة وظائف مثل تسجيل الدخول، ومعالجة الفيديو، والتفاعل مع قاعدة البيانات.

- **اختبار التكامل :(Integration Testing)** تم اختبار كيفية تفاعل الوحدات المختلفة مع بعضها البعض. على سبيل المثال، تم اختبار التفاعل بين واجهة المستخدم وقواعد البيانات للتأكد من أن البيانات تتدفق بشكل صحيح وأن جميع المكونات تعمل معًا بسلاسة.
- **اختبار النظام :(System Testing)** تم اختبار النظام بأكمله للتحقق من مطابقته للمتطلبات الوظيفية وغير الوظيفية. تم إجراء اختبارات للتحقق من الأداء والأمان وقابلية الاستخدام والموثوقية.
- **اختبار القبول :(Acceptance Testing)** تم اختبار النظام من قبل المستخدمين النهائيين (المراقبين والطلاب) في بيئة واقعية للتحقق من قابليته للاستخدام ورضاهم. تم جمع ملاحظات المستخدمين واستخدامها لتحسين النظام.

- **5.4.2 هنا يتم التقاط صوره للتحقق منها والمطابقة**

وبناء على قيمة Thresshold يتم تحديد دقة التعرف بواسطة خوارزمية face_recognition



- **5.4.3هنا يتم التحقق وإعطاء نتيجة المطابقة**

- **5.4.4 هنا يتم بدء المراقبة وارسال تنبيهات الفم**

بواسطة خوارزمية **Media Pipe js** تعمل في المتصفح بسرعة عالية واستجابة اعلى وموارد اقل تم التعرف على ملامح الوجة والراس وعلى ضوء التعرف يتم بدء المراقبة وبدء اكتشاف السيناريوهات مقدار فتح الفم بمقدار معين اذا تم تجاوزة يرسل اشعار وهكذا لبقية السيناريوهات.

- **5.4.5هنا أيضا اكتشاف التفات الي اليسار وإرسال تنبيه**

- **5.4.6 هنا يتم اكتشاف التفات لليمين وارسال تنبيه**

- **5.4.7 هنا يتم اكتشاف ميول الراس الي اسفل وارسال تنبيه**

- **5.4.8 هنا يتم ارسال تحذير عند تجاوز السيناريو والعدد المسموح للتنبيهات**

الفصل السادس

(النتائج والمقترحات)

6.1 مقدمة

يقدم هذا الفصل تحليلًا مفصلًا للنتائج التي تم الحصول عليها من خلال تنفيذ واختبار نظام مراقبة الامتحانات باستخدام الذكاء الاصطناعي. كما يتضمن مقترحات لتحسين النظام وتوسيع نطاقه في المستقبل. يهدف هذا الفصل إلى تقييم مدى تحقيق المشروع لأهدافه وتقديم رؤية مستقبلية لتطويره.

6.2النتائج الرئيسية

- **6.2.1 دقة التعرف على الوجه :**
  - حقق النظام دقة في التعرف على الوجه بلغت 70% في ظل الظروف المثالية (إضاءة جيدة، وجه واضح).
  - انخفضت الدقة إلى 52% في ظروف الإضاءة المنخفضة أو عندما كان هناك انسداد جزئي للوجه (مثل ارتداء النظارات)

  - يشير هذا إلى أن النظام فعال في التعرف على الطلاب في معظم الحالات، ولكن قد يتطلب تحسينًا في الظروف الصعبة.

- **6.2.2 كفاءة تحليل السلوك :**
  - تمكن النظام من اكتشاف محاولات الغش الشائعة (مثل النظر خارج الشاشة، التحدث) بدقة عالية.
  - كانت هناك بعض الحالات الإيجابية الخاطئة (False Positives) حيث تم اعتبار سلوكيات طبيعية على أنها مشبوهة، بمعدل.
  - يتطلب هذا تحسينًا في خوارزميات تحليل السلوك لتقليل الأخطاء الإيجابية الخاطئة وزيادة الدقة.

- **6.2.3 أداء النظام :**

- كان استهلاك النظام لموارد المعالج والذاكرة ضمن الحدود المقبولة، ولم يؤثر بشكل كبير على أداء الأجهزة.
- يشير هذا إلى أن النظام فعال من حيث الأداء ويمكن تشغيله على الأجهزة الموصى بها.

- **6.2.4 قابلية الاستخدام :**
  - أظهرت اختبارات المستخدم أن المراقبين وجدوا واجهة النظام سهلة الاستخدام وفعالة في مراقبة الطلاب.
  - تم الإبلاغ عن بعض الصعوبات الطفيفة في فهم بعض التنبيهات، مما يشير إلى الحاجة إلى تحسين وضوح التنبيهات.

6.3 المقترحات :

- **6.3.1 تحسين دقة التعرف على الوجه :**
  - استخدام نماذج أعمق وأكثر تعقيدًا للتعرف على الوجه.
  - تحسين عمل الخوارزميات، تتضمن صورًا في ظروف إضاءة مختلفة وزوايا وجه مختلفة.
  - استخدام تقنيات معالجة الصور لتحسين جودة الصور قبل معالجتها بواسطة نموذج التعرف على الوجه.
- **6.3.2 تحسين كفاءة تحليل السلوك :**
  - تحسين الية اكتشاف السلوكيات، تتضمن أمثلة متساوية على السلوكيات المشبوهة والسلوكيات الطبيعية.
  - إضافة المزيد من أنواع السلوكيات التي يمكن للنظام اكتشافها (مثل استخدام الهاتف، وجود شخص آخر في الغرفة).
- **6.3.3 تحسين قابلية الاستخدام :**
  - تحسين وضوح التنبيهات وإضافة المزيد من التفاصيل حول سبب إطلاق التنبيه.

- إضافة ميزات إضافية للمراقبين، مثل القدرة على تسجيل الفيديو يدويًا أو إضافة ملاحظات.
- تطوير واجهة مستخدم أكثر تخصيصًا، تسمح للمراقبين بتخصيص التنبيهات التي يتلقونها.

- **6.3.4 توسيع نطاق النظام :**

- دمج النظام مع أنظمة إدارة التعلم (LMS) لتوفير تجربة متكاملة.
- دعم أنواع مختلفة من الامتحانات (مثل الامتحانات الكتابية، الامتحانات الشفوية).
- تطوير تطبيق للهاتف المحمول للمراقبين، مما يسمح لهم بمراقبة الامتحانات من أي مكان.

6.4 خاتمة

حقق نظام مراقبة الامتحانات باستخدام الذكاء الاصطناعي نتائج واعدة في تحسين نزاهة الامتحانات وكفاءة المراقبة. ومع ذلك، هناك مجالات للتحسين والتوسع. تقدم المقترحات المذكورة في هذا الفصل خارطة طريق لتطوير النظام في المستقبل وجعله أداة أكثر قوة وفعالية.

الفصل السابع

(التوصيات)

7.1 مقدمة

يقدم هذا الفصل توصيات محددة بناءً على النتائج والمقترحات المذكورة في الفصل السابق. تهدف هذه التوصيات إلى توجيه الجهود المستقبلية لتطوير النظام وتحسينه، وضمان استخدامه الفعال في البيئات التعليمية.

7.2 التوصيات الخاصة بتطوير النظام

- **7.2.1 الأولوية العالية :**
  - **تحسين دقة التعرف على الوجه :**يجب أن يكون هذا على رأس الأولويات، خاصة في ظروف الإضاءة المنخفضة والانسداد الجزئي للوجه. يوصى بتجربة نماذج أعمق وتدريبها على مجموعة بيانات أكثر تنوعًا.
  - **تطوير تطبيق للهاتف المحمول للمراقبين :**يمكن تطوير تطبيق للهاتف المحمول في المستقبل، من خلال اقسام والقاعات التي يتم تحديدها له من قبل مشرف النظام بحيث يتم وصول اشعارات الغش لهاتف المراقب للتعامل معها واتخاذ الاجراء .
  - **تقليل الأخطاء الإيجابية الخاطئة في تحليل السلوك :**يجب العمل على تحسين خوارزميات تحليل السلوك لتقليل عدد المرات التي يتم فيها اعتبار السلوكيات الطبيعية على أنها مشبوهة. يوصى بتجربة نماذج تسلسلية اكثر كفائة.
  - **تطوير صلاحيات المستخدمين:** يجب أن يكون هناك أكثر من صلاحية في النظام مثل صلاحيات مراقبي التنبيهات حيث لك مراقب مركز وقاعة وفترات معينة.

- **7.2.2 الأولوية المتوسطة :**
  - **تحسين وضوح التنبيهات :**يجب جعل التنبيهات أكثر وضوحًا وتفصيلًا، بحيث يفهم المراقبون بسهولة سبب إطلاق التنبيه وما يجب عليهم فعله.
  - **إضافة ميزات إضافية للمراقبين :**يجب إضافة ميزات مثل القدرة على تسجيل الفيديو يدويًا وإضافة ملاحظات لتسهيل عمل المراقبين.
  - **تطوير واجهات لاضافة الاختبارات والتوزيعات:** بحيث انها تكون تضيف من خلال ملف اكسل (الملف الاختبارات الذي يتم ارساله من الكليه ويتم معالجته واضافته الى قاعده البيانات بشكل تلقائي، كذلك نفس الشي لتوزيع الطلاب .

- **7.2.3 الأولوية المنخفضة :**
  - **تطوير واجهة مستخدم أكثر تخصيصًا :**يمكن تخصيص الواجهة في المستقبل، ولكنها ليست ضرورية في المرحلة الحالية.

7.3التوصيات الخاصة بتطبيق النظام

- **7.3.1 التدريب :**
  - يجب توفير تدريب شامل للمراقبين على كيفية استخدام النظام وتفسير التنبيهات.
  - يجب توفير إرشادات واضحة للطلاب حول السلوكيات المسموح بها والمحظورة أثناء الامتحانات.
- **7.3.2 السياسات والإجراءات :**
  - يجب تطوير سياسات وإجراءات واضحة للتعامل مع حالات الغش التي يتم اكتشافها بواسطة النظام.
  - يجب مراجعة السياسات والإجراءات بانتظام وتحديثها حسب الحاجة.
- **7.3.3 البنية التحتية :**
  - يجب التأكد من أن البنية التحتية لتكنولوجيا المعلومات (الشبكات، الخوادم، الكاميرات) قادرة على دعم النظام.
  - يجب إجراء صيانة دورية للبنية التحتية لضمان الأداء الأمثل.

7.4 التوصيات الخاصة بالبحث المستقبلي

- **استكشاف تقنيات جديدة للتعرف على الوجه وتحليل السلوك :**يجب مواصلة البحث في أحدث التطورات في مجال الذكاء الاصطناعي لتطوير النظام.
- **دراسة تأثير النظام على سلوك الطلاب :**يجب إجراء دراسات لتقييم تأثير النظام على سلوك الطلاب أثناء الامتحانات.
- **تطوير نماذج مخصصة لأنواع مختلفة من الامتحانات :**يجب تطوير نماذج مخصصة لأنواع مختلفة من الامتحانات (مثل الامتحانات الكتابية، الامتحانات الشفوية).
- **يتم عمل تنبهات المراقبين في الجنه الامتحانات**

7.5 المراجع

 React:

- **نبذة عن المرجع :**مكتبة JavaScript قوية لبناء واجهات مستخدم تفاعلية وديناميكية لتطبيقات الويب.

- **الرابط:** https://react.dev/

 PostgreSQL Documentation:

- **نبذة عن المرجع:** الوثائق الرسمية والشاملة لنظام إدارة قواعد البيانات العلائقية مفتوح المصدر PostgreSQL. تعتبر المرجع الأساسي لفهم واستخدام هذا النظام.
- **الرابط:** https://www.postgresql.org/docs/

 Flask Quickstart:

- **نبذة عن المرجع:** صفحة البداية السريعة لإطار عمل الويب Flask في Python. تقدم خطوات عملية لإنشاء أول تطبيق ويب باستخدام Flask.
- **الرابط:** https://flask.palletsprojects.com/en/stable/quickstart/

 MediaPipe:

- **نبذة عن المرجع:** صفحة مشروع مكتبة MediaPipe على PyPI. توفر MediaPipe حلولًا جاهزة للعديد من مهام رؤية الكمبيوتر والتعرف على الحركة في الوسائط المتعددة.
- **الرابط:** https://pypi.org/project/mediapipe/

 إطار عمل Flask:

- **نبذة عن المرجع:** إطار عمل ويب مصغر ومرن مكتوب بلغة Python. يتيح للمطورين بناء تطبيقات ويب بسيطة ومعقدة بكفاءة وسرعة.

 FaceNet:

- **نبذة عن المرجع:** نموذج معماري طُوّر بواسطة Google Research ، يحول صور الوجوه إلى متجهات رقمية لتمثيل الخصائص الفريدة للوجه، مما يتيح مقارنة الوجوه بكفاءة.
- **المؤلفون:** Florian Schroff, Dmitry Kalenichenko, James Philbin
- **مكان النشر:** Computer Vision and Pattern Recognition (CVPR), 2015
- **الرابط:** https://arxiv.org/abs/1503.03832 .

7.6خاتمة

من خلال اتباع هذه التوصيات، يمكن للمؤسسات التعليمية تعزيز نزاهة الامتحانات وتحسين كفاءة المراقبة، مع ضمان تجربة عادلة وشفافة للطلاب.