

بوابة طالب الكترونية لمركز اللغات والترجمة بجامعة إقليم سبا

**Electronic student portal for the Languages and
Translation Center at Saba University**

21230156

تسنيم لطف الله الفراصي

21210187

أمة الله علي البعداني

20210255

سمية مجاهد الحشيبري

BACHELOR OF COMPUTER INFORMATION SYSTEMS

Supervised by Dr. Abdulaziz Ahmed Thawaba

Proposal Submitted to the Department of Computer Science,

Faculty of Information Technology & Computer Science, University of Saba Region

In Partial Fulfillment of the Requirements for the Degree of Bachelor of

Computer Science



UNIVERSITY OF SABA REGION

FACULTY OF INFORMATION TECHNOLOGY & COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

2024

بوابة طالب الكترونية لمركز اللغات والترجمة بجامعة إقليم سبا

**Electronic student portal for the Languages and
Translation Center at Saba University**

By

21230156 تسنيم لطف الله الفراصي

21210187 أمة الله علي البعداني

20210255 سمية مجاهد الحشيبري

Thesis Submitted to the Department of Computer Science,

Faculty of Information Technology & Computer Science, University of Saba Region

In Partial Fulfillment of the Requirements for the Degree of Bachelor of

Computer Science

September, 2024

بوابة طالب الكترونية لمركز اللغات والترجمة بجامعة إقليم سبا

**Electronic student portal for the Languages and
Translation Center at Saba University**

By

21230156 تسنيم لطف الله الفراصي

21210187 أمة الله علي البعداني

20210255 سمية مجاهد الحشبيري

**Project Report Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of
Computer Science**

At

the Department of Computer Science,

Faculty of Information Technology & Computer Science,

University of Saba Region

September 2024

Signature of Author: _____

Committee

Signature & Date

Dr. _____

Dr. _____

Dr. _____

إقرار

نقر بـأن هذا المشروع هو ثمرة جهودنا وبحثنا الشخصي. كما نقر بـأن جميع المعلومات والبيانات الواردة في هذا المشروع هي صحيحة ولم يسلم من قبل طرف آخر لأي جامعة، ونقر أياً ما بأننا لم ننقل أي معلومات أو بيانات من مصادر أخرى دون الإشارة إلى مصدرها.

إهـاء

للعائـلة أولاً،

لأنـ الطريق ما هـان إـلا بـهم،
وـما كانـ الوـصول إـلا ثـمرة جـهـدـهم.

ثـمـ لـكـلـ مـنـ كـانـ لـنـاـ عـونـاـ فـيـ الطـرـيقـ،
لـدـكـاتـرـتـناـ الـأـفـاضـلـ،
لـرـفـاقـنـاـ الـذـينـ سـانـدـنـاـ حـتـىـ آـخـرـ لـحـظـةـ
شـكـرـاـ لـأـنـكـمـ كـنـتـمـ جـزـءـاـ ثـرـيـاـ مـنـ الرـحـلـةـ.

ثـمـ أـنـ تـلـكـ الـبـذـرـاتـ الـيـوـمـ أـصـبـحـنـ حـصـادـاـ ..ـ هـذـاـ إـلـنـجـازـ لـكـمـ.

شكر خاص

نقدم بخالص الشكر والعرفان لدكتورنا الفاضل، ومشرف المشروع:

الدكتور عبدالعزيز ثوابة

لما قدمه لنا طيلة الفترة السابقة من جهود وتوجيهات ونصائح تستحق الشكر والثناء،

ثم للدكتور الفاضل:

إبراهيم الحوري

مدير مركز اللغات والترجمة لجامعة إقليم سبأ

لوقته المبذول في سبيل نجاح المشروع

شكراً لا تفي حكم.

جدول المحتويات

1	مقدمة	1
2	خلفية المشروع project background	1.1
3	مشكلة المشروع Problem Statement	1.2
3	الأهداف Objective	1.3
3	نطاق المشروع Project Scope	1.4
3	النتائج المتوقعة من المشروع Project Results	1.5
3	أهمية المشروع Project Significance	1.6
4	دراسة الجدوى الاقتصادية	1.7
4	الجدوى السوقية	1.7.1
6	الجدوى التقنية	1.7.2
6	الجدوى المالية	1.7.3
8	مخطط المشروع Project Outline	1.8
5	الأنظمة السابقة	2
6	المقدمة	2.1
6	دراسة الأنظمة الحالية System Study Existing	2.2
6	دراسة الأنظمة الحديثة study system of modern	2.3
6	تطبيق معهد ASA (ASA، 2024)	2.4
7	الخدمات التي يقدمها التطبيق:	2.4.1
8	طلب خدمة البحث عن الكتب في مكتبة معهد ASA جامعة القاهرة الجديدة	2.4.2
	عبر التطبيق	8
8	تطبيق UAEU الاماراتي (UAEU، 2024)	2.5
9	عملية تسجيل الدخول	2.5.1
10	الخدمات التي يقدمها التطبيق	2.5.2
12	تطبيق مركز النخبة (النخبة، 2024)	2.6
12	عملية تسجيل الدخول	2.6.1
13	الخدمات التي يقدمها التطبيق:	2.6.2
14	المقارنة (Comparison)	2.7
15	ملخص الفصل	2.8

16	التحليل والتصميم.....	3
17	منهجية المشروع.....	3.1
17	ما هي منهجية ? prototype	3.1.1
17	كيف بدأت؟.....	3.1.2
18	مراحل المنهجية.....	3.2
18	مرحلة جمع المتطلبات.....	3.2.1
18	مرحلة إعداد تصميم سريع.....	3.2.2
19	مرحلة تطوير النموذج الأولي.....	3.2.3
19	مرحلة التقييم + الاختبار والتحسين.....	3.2.4
19	مرحلة تنفيذ المنتج	3.2.5
19	تحليل وتصميم متطلبات النظام.....	3.3
20	لغة التمذجة الموحدة (UML)	3.4
20	مخطط ERD	3.5
21	مخطط (Use Case Diagram)	3.5.1
23	مخطط سير العمليات (Activity Diagram)	3.5.2
25	مخطط التتابع او مخطط التسلسل (Sequence Diagram)	3.5.3
27	مخطط التعاون (Collaboration Diagram)	3.5.4
27	مخطط الفئة (Class Diagram)	3.5.5
28	مخطط الحالة (State Chart Diagram)	3.5.6
29	الخلاصة	3.6
28	التنفيذ	4
29	مقدمة	4.1
29	نظام إدارة شؤون طلاب المركز	4.2
31	صفحة تسجيل طالب	4.2.1
36	تطبيق الطالب	4.3
36	صفحة تسجيل الدخول	4.3.1
43	طلب HTTP ومعالجة الاستجابة	4.3.2
45	إدارة خدمات التطبيق	4.3.3
46	المكتبة الإلكترونية	4.3.4

48	الخلاصة	4.4
53	الخاتمة	5
54	تقييم مدى تحقق الأهداف المحددة للمشروع	5.1
54	اقتراحات مستقبلية	5.2
56	المراجع	6

قائمة الأشكال

شكل 2.1 تطبيق ASA (2024)	7
شكل 2.2 مكتبة التطبيق	8
شكل 2.3 واجهة تطبيق UAEU الإماراتي	9
شكل 2.4 تسجيل الدخول لتطبيق UAEU الاماراتي	9
شكل 2.5 نافذة الخدمات	11
شكل 2.6 شعار التطبيق	12
شكل 2.7 نافذة تسجيل الدخول لتطبيق النخبة	12
شكل 2.8 نافذة الخدمات التي يقدمها التطبيق	13
شكل 2.9 نافذة تمكن الطالب من معرفة نتائج البحث	14
شكل 3.1 مراحل منهجية الـ prototype	18
شكل 3.2 : مخطط ERD	21
شكل 3.3: عملية قيام الطالب بتسجيل الدخول إلى النظام	22
شكل 3.4: عملية قيام الموظف بتسجيل الدخول لنظام	22
شكل 3.5: مخطط الأنشطة لتطبيق الطالب	23
شكل 3.6: مخطط الأنشطة لعملية تسجيل طالب عبر النظام	24
شكل 3.7: مخطط التسلسل لتطبيق الطالب	25
شكل 3.8: مخطط التسلسل لنظام	26
شكل 3.9: مخطط التعاون لتطبيق الطالب	27
شكل 3.10: مخطط class diagram	28
شكل 3.11: مخطط حالة لنظام	29
شكل 4.1: واجهة تسجيل طالب جديد من خلال النظام	29
شكل 4.2: تابع واجهة تسجيل طالب جديد	30
شكل 4.3: عرض خبر تم نشره من لوحة النظام	30
شكل 4.4: واجهة الدفع	31
شكل 4.5	31
شكل 4.6	32

شكل 4.7 كود جلب المستويات حسب البرنامج.....	32
شكل 4.8: دالة إرسال بيانات الطالب	33
شكل 4.9: كود التعامل مع حقول الإدخال.....	34
شكل 4.10 بعض واجهات التطبيق.....	36
شكل 4.11 جزء من كود تسجيل الدخول	37
شكل 4.12: دالة التتبیه على الخروج من التطبيق.....	38
شكل 4.13: كود تصميم نموذج تسجيل الدخول.....	39
شكل 4.14: تكملة لكود واجهة تسجيل الدخول.....	40
شكل 4.15: الكود الخاص بحقول الإدخال والقابل لإعادة الاستخدام	41
شكل 4.16: كود التحقق من نوع محتوى حقل الإدخال.....	42
شكل 4.17: كود التتحقق في الواجهة	43
شكل 4.18 كود معالجة الـ response	43
شكل 4.19: كود ارسال بيانات المستخدم للخادم للتحقق من صحتها.....	44
شكل 4.20: استخدام الفايربيس والـ shared preference بالتطبيق.....	45
شكل 4.21: بناء صف من المستويات	46
شكل 4.22: تابع كود المستويات.....	48

قائمة الجداول

6.....	جدول 1-1 التكاليف
7.....	جدول 2-1 الإيرادات.....
7.....	جدول 3-1 مقارنة بين النظام اليدوي والآلي
8.....	جدول 4-1 الجدوى الاقتصادية.....
14	جدول 2-2 مقارنة بين الأنظمة الحديثة والنظام الحالى.....

الفصل الأول

1 مقدمة

1.1 خلفية المشروع | project background

في ظل الموجة الرقمية والتطور التقني المتتسارع أصبح استخدام التطبيقات جزءاً لا يتجزأ من حياتنا اليومية، حيث تساعد العديد من التطبيقات في تسهيل الكثير من المهام التي قد تواجه الفرد في يومه، فنجد تطبيقات المعاملات المالية وتطبيقات المتابعة المدرسية وموقع بوابة الطالب وتطبيقات التواصل الاجتماعي التي بدورها تضم مجموعة من المجتمعات التي تهم الفرد وتسهل عليه الكثير.

وفي مشروعنا حاولنا إيجاد مؤسسة أو معهد ما يتبنى فكرة هذا التحول الرقمي التي بدأت المؤسسات بالخارج تعتمده كوسيلة لتسهيل مهامها وتحفيض الضغط عليها وعلى عمالها أو طلابها، وكانت فكرة المشروع مقدمة إلى مركز اللغات والترجمة بجامعة إقليم سبا، حيث عملنا على برمجة وتصميم تطبيق طالب بالإضافة لنظام كامل للتحكم به، يمكن التطبيق الطالب من الوصول لمعلومات الدورة المسجل فيها كما يزوده بأحدث الأخبار و الفعاليات التي ينظمها المركز، مما يتيح له البقاء على اطلاع دائم على جميع الأنشطة والفرص المتاحة. يوفر مكتبة من الكتب الإنجليزية المقسمة حسب مستويات الطلاب، من المبتدئ إلى المتقدم. يمكن للطالب من خلال التطبيق الاطلاع على نتائجه ودرجاته في الاختبارات، يرسل التطبيق إشعارات للطالب حول أي مستجدات أو معلومات مهمة من المركز.

نسعي في مشروعنا بالمؤسسات التعليمية إلى مواكبة التقدم التقني الذي اكتسح العالم في الفترة الأخيرة وذلك من خلال تبني فكرة التحول الرقمي وأتمته المهام التي يمكن أتمتها وجعلها تتم الكترونياً.

ففيما يتعلق بالتطبيقات الطلابية فإنها تتضمن تنظيم الجداول، وتوزيع الفصول، والسجلات، وتوزيع الطلاب ونظام القبول والتسجيل، ونظام الحضور والغياب، وتصحيح الاختبارات والأنشطة المدرسية، ونظام متابعة الانتقالات وكذلك نظام الامتحانات، والنتائج والتقارير الدورية لدرجات الطلاب. يضاف إلى ذلك، أن الإدارة الإلكترونية تعزز التواصل بين المؤسسة التعليمية والمنزل، من خلال الاتصال عبر الشبكة، والحصول على كافة المعلومات المتصلة التعليمي بالتحصيل الدراسي وتبادل المعلومات إلكترونياً، وتحقيق مستوى أعلى من التواصل بين المعلمين والطلاب. وذلك كلّه يتم عن طريق إدارة، وتخزين، ومعالجة كافة البيانات والمعلومات الخاصة بالطلاب (أمونة، 2009)

وهناك تطبيقات مصادر التعلم والمكتبات التي تتضمن توفير قاعدة بيانات للبحث عن الكتب، والمصادر، والمراجع، وتنظيم الاستعارة والزيارة. (نجم، 2009)

1.2 مشكلة المشروع | Problem Statement

في حين ازداد الاعتماد على التكنولوجيا أصبح من الضروري مواكبة التطورات وتقديم خدمات تعليمية تتناسب مع احتياجات الطلاب، وفي مركز اللغات والترجمة بجامعة إقليم سبأ يواجه الطالب صعوبة في الاستعلام عن النتائج والقاعات وعدم توفر مكتبة الكترونية تدعم الطالب في تطوير مهارة القراءة وتطوير مفرداته في اللغة الانجليزية، ومن هنا جاءت فكرة تطبيق بوابة الطالب التي تسمح له بالوصول لأخبار المركز والوصول أيضاً لمكتبة انجليزية مقسمة حسب المستوى، كما تسهل عليه استعراض الجداول والنتائج وغيرها مما يؤدي إلى تسهيل عملية التعلم وتحسين تجربة الطالب بشكل عام.

1.3 الأهداف | Objective

نهدف في مشروعنا لما يأتي:

1. تسهيل وصول الطالب إلى بياناته في الدورة.
2. تسهيل الوصول إلى النتائج، مع تقييد الوصول لنتيجة الدورة التي لم يقم الطالب بتصفيه رسومها بعد.
3. مواكبة التطور التقني والتحول الرقمي في جعل كافة التعاملات بين الإدارة والعميل تتم عبر الهاتف.

1.4 نطاق المشروع | Project Scope

تحسين خدمات مركز اللغات والترجمة في جامعة إقليم سبأ-مأرب لتسهيل حصول الطالب إلى خدمات المركز من نتائج ومكتبة إلكترونية وجداول.

1.5 النتائج المتوقعة من المشروع | Project Results

الخروج بحل تقني يعمل على تحسين وصول خدمات المركز للطلاب بطريقة أسهل وأيسر، وذلك من خلال تصميم وبرمجة تطبيق طالب يضم مجموعة من الوظائف التي يستطيع الطالب الوصول لها عن بعد كالنتائج والجداول والمكتبة الالكترونية.

1.6 أهمية المشروع | Project Significance

أصبحت الهواتف الذكية منتشرة بين جميع أطياف المجتمع وخاصة الطلاب، وذلك نظراً لتطور تكنولوجياتها، وتعدد خدماتها التي يمكن أن تقدمها في مجال التعليم والتعلم. فالهاتف الذكي تتسم بقدرتها العالية على الوصول إلى الأفراد في أي مكان وأي وقت، وأثبتت جدواها وحاجة المجتمعات الضرورية لها، والمساهمة في التغلب على ما يعنيه التعليم التقليدي من مشكلات. (الدهشان، 2013)

وهنا يعمل نظامنا على تسهيل عمليات الوصول لخدمات المركز رقمياً دون الحاجة للحضور الشخصي إلى إدارة المركز أو إلى التواصل الهاتفي الذي بدوره قد يشكل مصدر تشتت لموظفي المركز نظراً لمعاناة المركز الأساسية من قلة عدد الموظفين، حيث تم استخدام أحدث التقنيات في محاولة لأتمنة عملية وصول هذه الخدمات للطلاب رقمياً عبر شاشة الجوال، مما يسهل عليهم رحلة التعلم ويوفر عليهم أعبائها.

1.7 دراسة الجدوى الاقتصادية

تهدف هذه الدراسة إلى تقييم جدوى مشروع تطوير تطبيق ونظام بوابة الطالب لجامعة أقليم سباء، مع التركيز على جانب تسهيل وصول الطلاب إلى بياناتهم في دورات اللغات والترجمة. سيتم تحليل جوانب المشروع المختلفة، بما في ذلك دراسة السوق، والتكاليف المتوقعة.

وصف المشروع

يتضمن المشروع تطوير تطبيق للهواتف الذكية ونظام ويب متكامل لبوابة الطالب، حيث يمكن للطلاب من خلاله:

- الوصول إلى بيانات الدورة وموعدها.
- متابعة نتائج الدورات.
- الوصول لمكتبة الكترونية مقسمة إلى مستويات.

ما هي المشكلة التي يحلها؟

- صعوبة وصول الطلاب إلى المعلومات المتعلقة بدورات اللغة والترجمة.
- عدم توفر خدمات إلكترونية تسهل التواصل بين الطالب والإدارة.
- زيادة العبء على الموظفين بسبب الطلب المتزايد على الخدمات.

1.7.1 الجدوى السوقية

العملاء المستهدفون: طلاب الجامعة المسجلين في دورات اللغة والترجمة. وموظفي المركز.

حجم السوق: عدد الطلاب المستخدمين للتطبيق والموظفين المستخدمين للنظام

الاتجاهات السوقية: هناك اتجاه عالمي نحو التحول الرقمي وتقديم الخدمات بمختلف أشكالها عبر الإنترنت.

المنافسون: لا يوجد منافسون مباشرون داخل الجامعة، ولكن هناك بعض التطبيقات الخارجية التي تقدم خدمات مشابهة.

الميزة التنافسية: تخصيص التطبيق لاحتياجات طلاب مركز اللغات والترجمة، وتوفير ميزات إضافية مثل المكتبة الإلكترونية.

استراتيجية التسويق: الترويج للتطبيق داخل الجامعة عبر الملصقات والإعلانات، واستخدام وسائل التواصل الاجتماعي للتسويق العالمي

تحليل SOWT

نقاط القوة:

1. تخصيص التطبيق لاحتياجات طلاب مركز اللغات والترجمة بإقليم سبأ.
2. تجربة استخدام واضحة وسلسة.
3. توفير ميزات فريدة مثل الوصول لمكتبة الكترونية.

نقاط الضعف:

1. الاعتماد على استقرار نظام المركز.
2. الحاجة إلى تحديثات مستمرة لمواكبة التطوير.

الفرص: زيادة اقبال الطلاب للتسجيل في الدورات الإنجليزية. زيادة كفاءة المركز، تقليل الأخطاء الإدارية، تحقيق التحول الرقمي.

التهديدات المنافسة: التغيرات التكنولوجية السريعة.

الخطة التسويقية المقترحة:

استراتيجية الترويج:

1. الترويج داخل الجامعة: الملصقات، الإعلانات في النشرات الجامعية.
2. التسويق عبر وسائل التواصل الاجتماعي: إنشاء صفحات خاصة بالتطبيق والتفاعل مع الطلاب.
3. عروض خاصة لجذب المستخدمين: تخفيضات على الدورات، مسابقات، مكافآت.

1.7.2 الجدوى التقنية

التقنيات المطلوبة: تطوير تطبيقات الهواتف الذكية (Android و iOS)، قواعد بيانات لتخزين المعلومات، واجهات برمجة تطبيقات للتكامل مع أنظمة الجامعة.

المهارات التقنية: يحتاج المشروع إلى فريق عمل يمتلك مهارات في البرمجة، تصميم قواعد البيانات، وتصميم واجهات المستخدم.

المخاطر التقنية: أخطاء برمجية، ضعف الأمان، عدم التوافق مع الأجهزة المختلفة.

1.7.3 الجدوى المالية

التكاليف: تكاليف التطوير، الاستضافة، التسويق، الصيانة.

الإيرادات: يمكن تحقيق الإيرادات من خلال فرض رسوم اشتراك سنوية على الطلاب، أو من خلال بيع مساحات إعلانية داخل التطبيق، أو من خلال تقديم خدمات إضافية مدفوعة.

التكاليف

جدول 1-1 التكاليف

البند	التكلفة	ملاحظات
صيانة التطبيق	\$100	يشمل صيانه التطبيق، تحديثه واصلاح اي اعطال
الاستضافة	\$7.99	يشمل استضافه النظام على سيرفر متخصص
التسويق	\$100	يشمل تكلفة التسويق للتطبيق، الاعلانات

الإيرادات

جدول 2-1 الإيرادات

الملاحظات	التكلفة	البند
يشمل رسوم الاشتراك التي يدفعها الطالب للوصول إلى جميع الخدمات (100 طالب)	\$500	رسوم الاشتراك

جدول 31- مقارنة بين النظام اليدوي والآلي

النظام الآلي (بوابة الطالب الإلكتروني)	النظام اليدوي (الحالي)	الميزة
سهولة الوصول عبر الهاتف الذكي في أي وقت ومكان	وصول محدود فقط خلال ساعات العمل في المركز	الوصول إلى الخدمات
تواصل مباشر مع الإدارة عبر التطبيق	تواصل مباشر من خلال زيارة المركز شخصياً	التواصل مع الإدارة
توفير الوقت والجهد	اهدر لوقت والجهد سبب الاعتماد على العمل اليدوي	الكافأة
عرض النتائج بسهولة وسرعة مع تقييد الوصول للنتائج غير المدفوعة	عرض النتائج من خلال زيارة المركز شخصياً	النتائج
وصول مباشر إلى مكتبة الكترونية مقسمه إلى مستويات	لاتتوفر بالمركز	المكتبة الإلكترونية
توفير التكلفة على المركز من خلال تقليل الاعتماد على الموظفين	تكلفة عالية على المركز	التكلفة

جدول 4-1 الجدوی الاقتصادية

النظام اليدوي	شهر	سنہ	خمس سنوات	النظام الالكتروني	شهر	سنہ	خمس سنوات
رواتب الموظفين	800\$	9,600\$	48,000\$	رواتب الموظفين	800\$	9,600\$	48,000\$
تكلفة المواد الورق_الحبر	40\$	448\$	2,400 \$	HP Notebook Laptop, 15.6" HD Touchscreen, Intel Core i3-1115G4 Processor, 32GB RAM	40\$	480\$	448\$
الإجمالي			50,848\$				448\$
تكلفة النظام اليدوي بعد خمس سنوات = 50,848							

تكلفة النظام اليدوي بعد خمس سنوات = 50,848

القرار: يتم الاعتماد على النظام نظراً لتخفيض التكلفة وزيادة الكفاءة، ويقلل من الأخطاء البشرية.

1.8 مخطط المشروع | Project Outline

في نهاية هذا الفصل يتكون هذا المشروع من عدة فصول مرتبة كالتالي:

الفصل الثاني: دراسة الأنظمة السابقة ومقارنة بينها وبين النظام الحالي لمركز اللغات

الفصل الثالث: نظرة عن المنهجية المتبعة في المشروع مع تحليل وتصميم المشروع من خلال مخططات use و ERD و case

الفصل الرابع: تنفيذ المشروع برمجياً وشرح بعض أجزاءه المهمة.

الفصل الخامس: الخاتمة وتقييم مدى تحقيق الأهداف التي تم تحديدها في الفصل الأول من التوثيق.

الفصل الثاني

2 الأنظمة السابقة

2.1 المقدمة

في هذا الفصل سنقوم بدراسة النظام الحالي لمركز اللغات والترجمة لجامعة إقليم سبا، وسيتم مقارنته بأنظمة حديثة مشابهة لفكرة المشروع وذلك بغرض مقارنتهما معًا ومعرفة نقاط القوة والضعف في كلاً من النظائرتين لتبني نقاط القوة والاحتفاظ بها في مشروعنا، وتجنب نقاط الضعف، لنصل في الأخير لنظام حديث يدعم احتياجات الطالب في مركز اللغات.

2.2 دراسة الأنظمة الحالية | System Study Existing

النظام الحالي لمركز اللغات والترجمة في جامعة إقليم سبا يعتمد على الطرق التقليدية في تقديم خدماتهم للطلاب مثل: الاتصال الهاتفي أو الزيارة الشخصية.

يمكن أن يكون هذا النهج غير فعال في كثير من الأحيان وقد يسبب ضغطًا على موظفي المركز، أضف إلى ذلك عدم توفير مكتبة طلابية للكتب الإنجليزية لدعم الطالب في رفع مستوى اللغة، والتي تعد من أكثر العناصر الأساسية التي يفتقر إليها طالب اللغات في مركز الجامعة، علاوة على ذلك اعتماد المركز على الطرق التقليدية: مثل الإعلان عن النتائج والجداول على لوحات الإعلانات وما إلى ذلك.

2.3 دراسة الأنظمة الحديثة | study system of modern

سنقوم في هذا القسم بدراسة 3 تطبيقات مشابهة لفكرة مشروعنا نوردهم كما يلي:

2.4 تطبيق معهد ASA (2024)

هو تطبيق مصمم خصيصاً لطلاب معهد ASA. يوفر التطبيق للطلاب مجموعة من الخدمات، منها: عرض أخبار المعهد وفعالياته بحيث يمكن للطلاب الاطلاع على آخر الأخبار والفعاليات التي ينظمها المعهد. أيضاً عرض نتائج الطلاب بحيث يمكن للطلاب الاطلاع على نتائجهم في جميع الدورات التي يلتحقون بها. وعرض رقم القاعة بحيث يمكن للطلاب معرفة رقم القاعة التي يتواجدون فيها. وإرسال إشعارات للطلاب عند توفر نتائج جديدة أو عند تغيير رقم القاعة. خدمات أخرى: تقديم خدمات أخرى مثل التسجيل في الدورات، ودفع الرسوم، ومتابعة تقدمهم في الدورات، كل ما عليك فعله للاستفادة من تطبيق ASA هو تحميل التطبيق من متجر جوجل للهواتف التي تعمل بنظام اندرويد ومن متجر آبل لهواتف آيفون، ثم تسجيل الدخول للتطبيق، ثم طلب الخدمة المراده وفق تعليمات الطلب الذي تقدمت به.

2.4.1 الخدمات التي يقدمها التطبيق:

عرض الجدول الدراسي: يمكن للطلاب الاطلاع على جداولهم الدراسية، بما في ذلك الأيام والوقت والمكان

لكل محاضرة

عرض النتائج: يمكن للطلاب الاطلاع على نتائج اختباراتهم وواجباتهم.

تحميل المواد الدراسية: يمكن للطلاب تحميل المواد الدراسية، مثل المحاضرات والعرض التقديمية والكتب الإلكترونية.

البحث عن الكتب في المكتبة: يمكن للطلاب البحث عن الكتب في مكتبة المعهد عبر التطبيق، انظر الشكل (2.1) والذي يظهر محتويات الرئيسية



شكل 2.1 تطبيق ASA (2024)، ASA

2.4.2 طلب خدمة البحث عن الكتب في مكتبة معهد ASA جامعة القاهرة الجديدة عبر التطبيق

لتقديم طلب خدمة البحث عن الكتب في مكتبة معهد ASA جامعة القاهرة الجديدة عبر التطبيق، اتبع الخطوات التالية:

افتح التطبيق وقم بتسجيل الدخول انتقل إلى قسم "الخدمات" ، اختر "البحث عن الكتب في المكتبة" ، انظر الشكل(2.2). أدخل اسم الكتاب أو المؤلف أو موضوع الكتاب في حقل البحث. ستظهر لك قائمة بالكتب التي تتطابق مع معايير البحث الخاصة بك. يمكنك تصفية النتائج حسب المؤلف أو الموضوع أو سنة النشر. اضغط على اسم الكتاب الذي تريد معرفة المزيد عنه. ستظهر لك معلومات عن الكتاب، مثل المؤلف، والناشر، وسنة النشر، والوصف. إذا كان الكتاب متاحًا، يمكنك حجز نسخة منه.



شكل 2.2 مكتبة التطبيق

2.5 تطبيق UAEU الاماراتي (2024)

يُعد تطبيق جامعة الإمارات العربية المتحدة أداة قيمة للطلاب وأعضاء هيئة التدريس والموظفين في الجامعة. يوفر التطبيق وصولاً للعديد من الخدمات التي يتم توفيرها عن طريقه



شكل 2.3 واجهة تطبيق UAEU الإماراتي

2.5.1 عملية تسجيل الدخول

أدخل اسم المستخدم وكلمة المرور الخاصة بك في الحقول المخصصة كما بالشكل (2.3). إذا نسيت كلمة المرور ، يمكنك إعادة تعديتها عن طريق الضغط على "نسيت كلمة المرور؟".



شكل 2.4 تسجيل الدخول لتطبيق UAEU الإماراتي

2.5.2 الخدمات التي يقدمها التطبيق

عرض الجدول الدراسي: يمكنك عرض جدولك الدراسي الحالي والمستقبل، بما في ذلك الوقت والمكان والوصف لكل دورة.

تسجيل الدورات الدراسية: يمكنك إضافة وإزالة الدورات من عربة التسوق الخاصة بك وتسجيلها بسهولة.

عرض الدرجات: يمكنك عرض درجاتك الحالية والماضية، بما في ذلك متوسط درجاتك وتعليقات أعضاء هيئة التدريس.

دفع الرسوم الدراسية: يمكنك دفع رسومك الدراسية عبر الإنترنت باستخدام بطاقة الائتمان أو بطاقة الخصم.

المكتبة: البحث عن الكتب والمجلات: يمكنك البحث عن الكتب والمجلات في مكتبة الجامعة باستخدام الكلمات الرئيسية أو المؤلف أو العنوان.

طلب الكتب: يمكنك طلب الكتب والمواد الدراسية الأخرى من مكتبة الجامعة. عرض حساب المكتبة: يمكنك عرض حساب المكتبة الخاص بك، بما في ذلك الكتب المستعارة وتاريخ استحقاقها.



شكل 2.5 نافذة الخدمات

2.6 تطبيق مركز النخبة (النخبة، 2024)

يمكن لكافة الأهالي متابعة تطور أبنائهم وتحصيلهم العلمي في المدرسة عبر التطبيق، كما يمكن للزوار الاطلاع على معلومات المدرسة ورسالتها وأهدافها، والمعلومات الخاصة بالتواصل مع إدارة المدرسة.



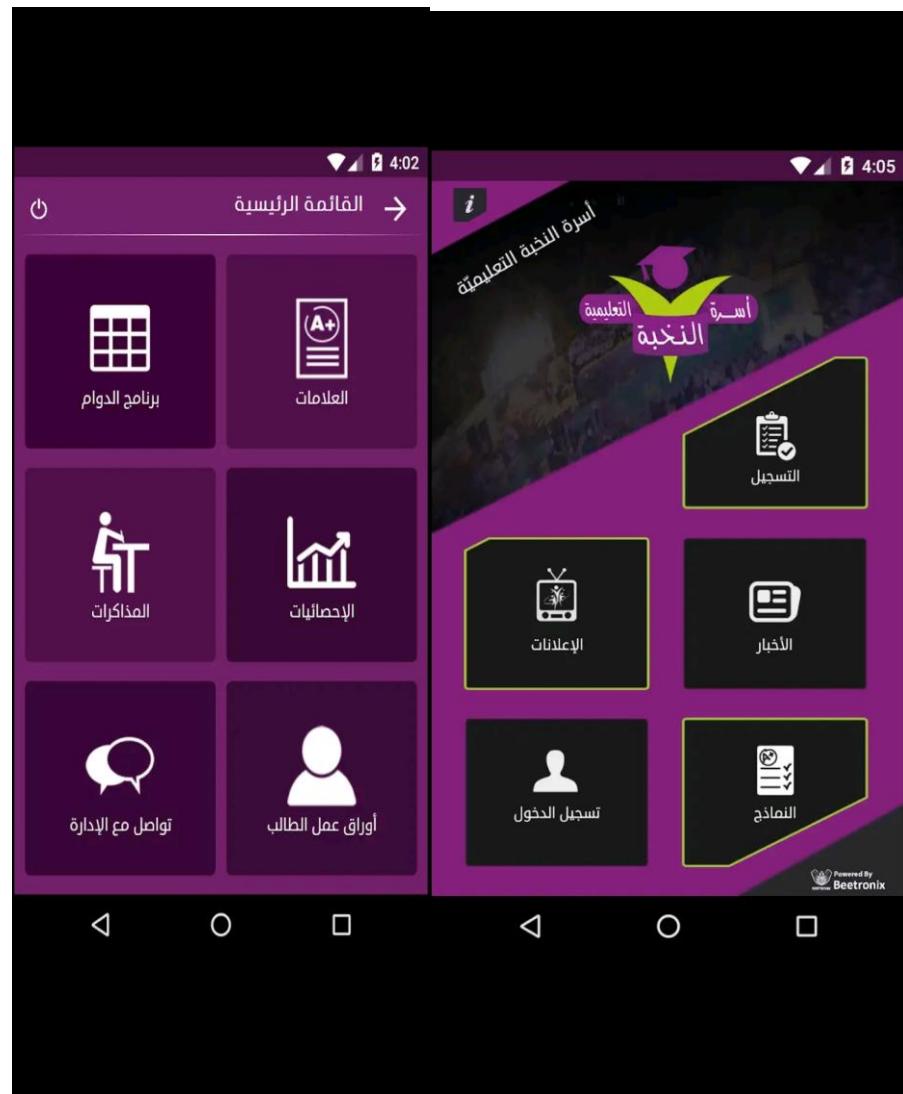
شكل 2.6 شعار التطبيق

2.6.1 عملية تسجيل الدخول

عملية تسجيل الدخول تعد بسيطة حيث تتطلب اسم المستخدم وكلمة المرور كما في الشكل (2.7).



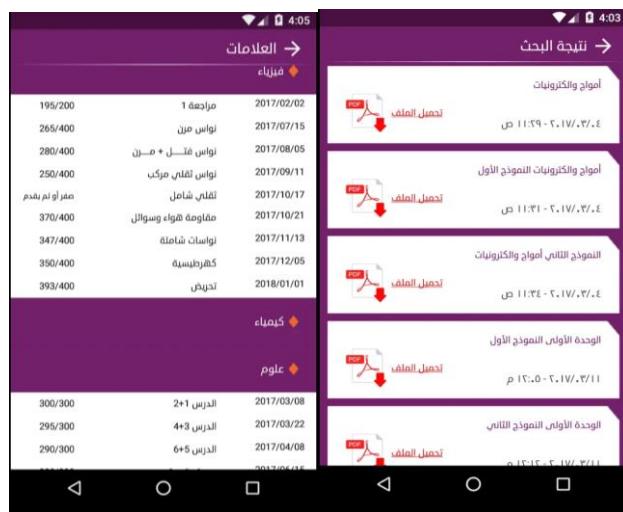
شكل 2.7 نافذة تسجيل الدخول لتطبيق النخبة



شكل 2.8 نافذة الخدمات التي يقدمها التطبيق

2.6.2 الخدمات التي يقدمها التطبيق:

1. يمكن لولي أمر الطالب أو الطالبة رؤية برنامج دوامه بشكل مباشر.
2. يتيح التطبيق معرفة كافة الامتحانات القادمة وتفاصيلها.
3. يمكن لولي الأمر معرفة كل نتائج الامتحانات من خلال التطبيق.
4. إمكانية التواصل مع الإدارة عبر ميزة المحادثة.
5. يعرض التطبيق كذلك احصائيات عامة تخص الطالب.
6. معلومات التواصل مع المدرسة ورسالتها وأهدافها.



شكل 2.9 نافذة تمكن الطالب من معرفة نتائج البحث.

2.7 المقارنة (Comparison)

تطبيق ASA المصري وتطبيق مركز النخبة وتطبيق UAEU الاماراتي في هذا الجزء من هذا الفصل سنقوم بمقارنة ما بين التطبيقات لكي اوضح سهولة استخدام والميزات. أيضا هذا البيانات لا يتشابهان ولكن سوف نأخذ بعض مميزات لتطبيقنا الذي سنقوم بعملة.

جدول 1-2 مقارنة بين الأنظمة الحديثة والنظام الحالي.

النظام الحالي	تطبيق لمركز النخبة	تطبيق UAEU الاماراتي	تطبيق ASA المصري	التطبيق	الميزات
✓	✓	✓	✓		سهولة الاستخدام
✗	✓	✗	✓		سهولة الوصول للخدمات
✗	✗	✓	✓		توفير مكتبة رقمية
✗	✓	✗	✓		رضي الطلاب
✗	✓	✓	✓		خدمة عرض النتائج
✗	✓	✓	✓		ميزة التواصل مع الادارة
✗	✓	✓	✓		توفر تطبيق

2.8 ملخص الفصل

تحدثنا في هذا الفصل عن المشاريع السابقة أو أنظمة مشابهة نوعاً ما لنظامنا الذي قمنا بتصميمه وتم ادراج صور توضح هذه الأنظمة وفكرتها ومقارنة بسيطة بين الأنظمة.

الفصل الثالث

3 التحليل والتصميم

3.1 منهجية المشروع

اخترنا في مشروعنا منهجية prototype أو ما تسمى بمنهجية النماذج الأولية للعمل عليه، وقد كان اختيارنا لها بسبب جهاناً بتفاصيل النظام الذي بدأنا العمل عليه منذ البداية، ولنفهم تفاصيله تماماً لكي لا يكون عملنا على المشروع مبني على جهل بتفاصيل النظام الأساسي للمركز.

3.1.1 ما هي منهجية prototype ؟

"في مجال تصميم البرمجيات، يُعد النموذج الأولي عبارة عن تصميم أولي للمنتج يمكن أن يتفاعل معه المستخدمون. تكون النماذج الأولية غير مبرمجة بعد، لكن يمكن استعراضها من خلال أدوات معينة للتتقل بين الشاشات ومحاكاة تجربة المستخدم النهائي. تسمح النماذج الأولية بتوفير الوقت والمال في مرحلة التطوير، لأنها تساعد على اكتشاف المشاكل التي تعيق سهولة الاستخدام في مرحلة مبكرة وتحسين التصميم قبل مرحلة بناء المنتج." (شريجي، 2024)

3.1.2 كيف بدأت؟

"في عام 2001، اجتمع ثلاثة أصدقاء في إدنبرغ Edinburgh ليشاركونا اقتراحاتهم عن مشروع يعملون عليه معاً، وقد أعدَّ كلُّ منهم مجموعة أفكار ليصوتووا عليها ويختاروا الفكرة الأفضل، لكن واحداً منهم فقط كان قد أعدَّ نموذجاً أولياً لفكرته.

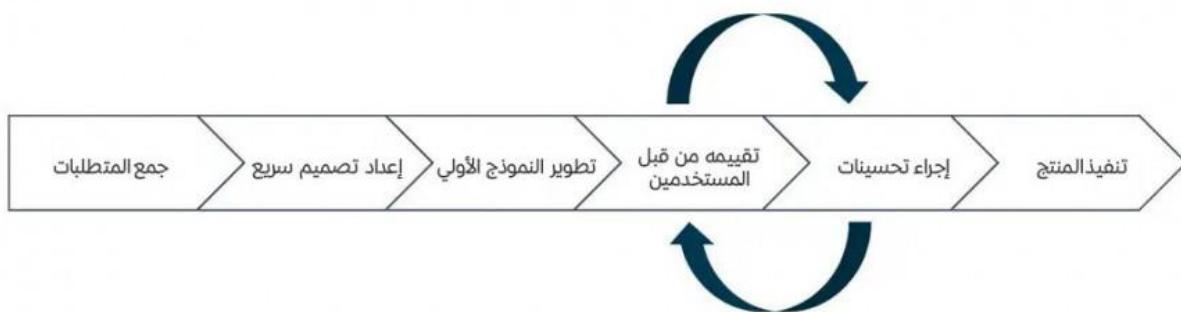
شارك غاريث ويليامز Gareth Williams مع أصدقائه ملف إكسيل كان قد أعده ليساعده على إيجاد رحلات طيران رخيصة، مما حمس الجميع مباشرةً لفكرة المتمثلة في إنشاء موقع ويب للبحث عن رحلات طيران؛ وفي عام 2002 صدرت النسخة الأولى من موقع سكاي سكانر Skyscanner لحجوزات الطيران.

لا تقتصر أهمية إعداد النماذج الأولية prototypes على إقناع المستثمرين وشركاء المشروع بالفكرة، بل تستخدَّم هذه النماذج أيضًا في مرحلة التطوير للمساعدة على التخطيط وتقييم المنتج واختباره قبل إنفاق الكثير من المال والوقت." (شريجي، 2024)

3.2 مراحل المنهجية

تمر منهجية الـ prototype بعدة مراحل كغيرها من المنهجيات، إلا أنها تقوم بتصميم عدة نماذج أولية قبل تنفيذ العمل حتى تصل للنموذج النهائي والمثالي ليتم تنفيذه على أرض الواقع، كما يظهر في الشكل(8).

مراحل إعداد النماذج الأولية



شكل 3.1 مراحل منهجية الـ prototype

الآن لنقوم بتفصيل وشرح كل مرحلة من المراحل:

3.2.1 مرحلة جمع المتطلبات

حيث قمنا خلالها باستخدام عدة طرق لجمع المتطلبات الوظيفية وغير الوظيفية للمشروع، كان من ضمنها أن نسقنا لاجتماعات (مقابلات) مباشرة مع إدارة المركز، وأيضاً قمنا باستخدام وسيلة أخرى لجمع المتطلبات وهي ما تسمى باللحظة أو Observation وفيها قمنا بطلب رؤية النظام الحالي الذي يعمل عليه المركز وتم بالفعل تشغيله والعمل عليه على مرأى ومسمع منا وقمنا بمتابعة طريقة العمل عليه وتسجيل الملاحظات والتي استخدمنا منها لاحقاً في مشروعنا.

3.2.2 مرحلة إعداد تصميم سريع

قمنا في هذه المرحلة برسم سketches أو مخططات بسيطة توضح النقاط الأساسية في المنتج الرقمي الذي نعمل عليه.

3.2.3 مرحلة تطوير النموذج الأولي

تم في هذه المرحلة تطوير نموذج أولي لكل من التطبيق والنظام، وذلك عن طريق Figma وهي أداة تساعد مصممي UI/UX على تصميم منتجاتهم رقمياً لإعطاء تصور أولي لكيف سيكون شكل المنتج، في هذه المرحلة تبدأ الأفكار تتنظم تدريجياً حيث تبدأ ترجمتها من مجرد أفكار مختلفة ومشتتة في الذهن إلى صورة منظمة لمنتج حقيقي.

3.2.4 مرحلة التقييم والاختبار والتحسين

من هنا تبدأ الحلقة التكرارية ما بين تقييم واختبار وتحسين ثم تقييم واختبار وتحسين وهكذا، حتى يتم الاتفاق على صورة المنتج النهائي.

3.2.5 مرحلة تنفيذ المنتج

يتم في هذه المرحلة تنفيذ المنتج واقعياً وذلك من خلال برمجته وتشغيله كمنتج نهائي قابل للاستخدام، حيث انتقلنا هنا من بناء واجهة المستخدم UI وتجربة المستخدم UX إلى مرحلة كتابة كود والتي تستهلك الوقت الأطول من بين بقية المراحل، وتستمر مراقبة المنتج بعد إطلاقه لضمان سهولة الاستخدام.

3.3 تحليل وتصميم متطلبات النظام

تنقسم متطلبات المشروع إلى:

متطلبات وظيفية:

1. توفير نظام تسجيل الدخول والخروج للطالب والـadmin.
2. توفير نظام لرفع الكتب وعرضها على المكتبة الالكترونية الخاصة بالطالب.
3. إتاحة خيار حظر حساب طالب عبر النظام الخاص بالمركز.
4. توفير طريقة للتواصل بين الطالب والمركز وتلقي الاشعارات حول المستجدات.

متطلبات غير وظيفية:

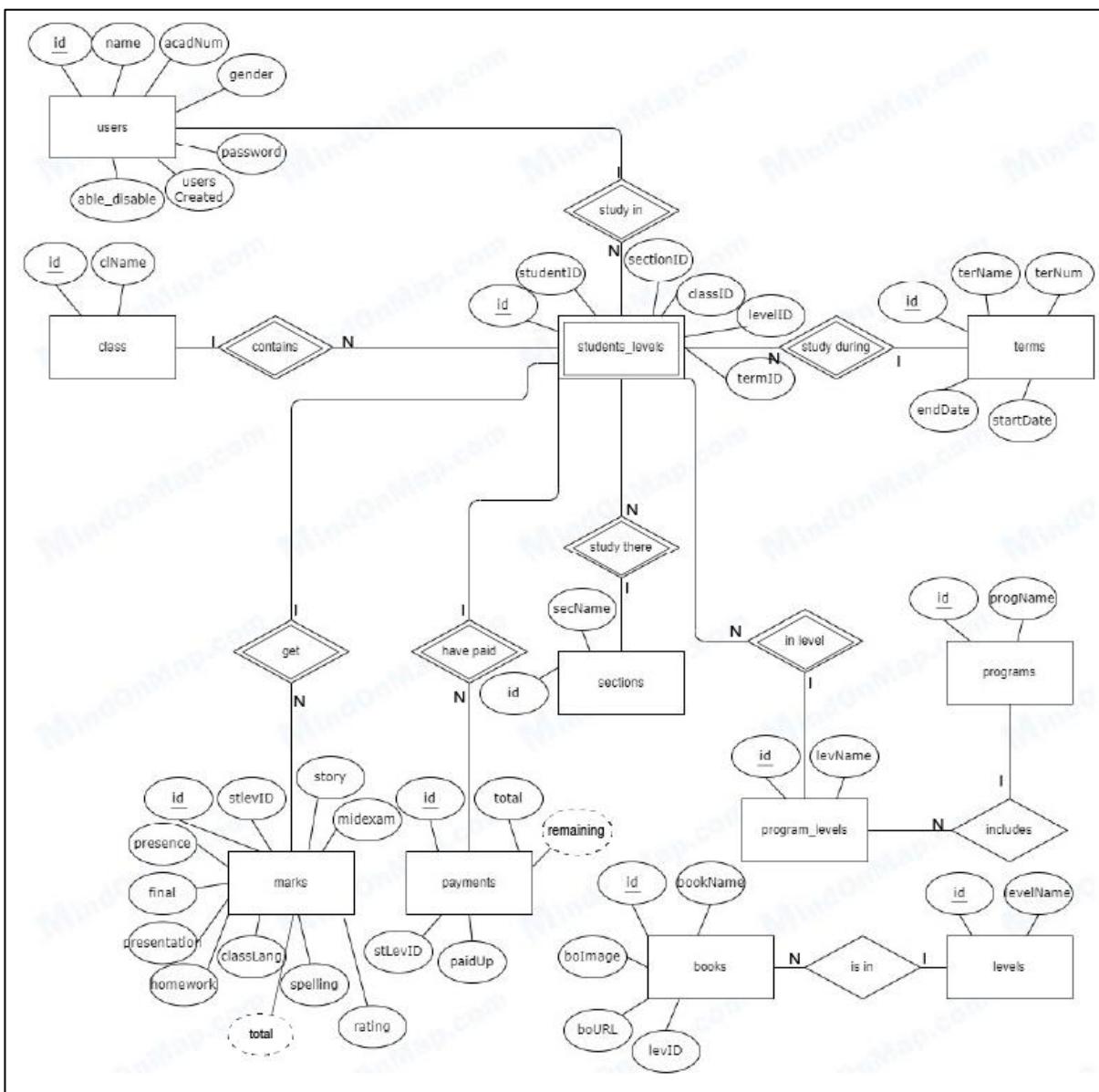
1. سرعة الإستجابة.
2. الأمان والحماية.
3. سهولة الاستخدام.
4. سهولة الصيانة.
5. مريح للعين

3.4 لغة التمذجة الموحدة (UML)

لغة التمذجة الموحدة هو اختصار UML وهي أداة لنمذجة الأنظمة والتطبيقات التي تعتمد على البرمجة الكائنية OOP تستخدم لعمل نموذج مبدئي للبرنامج سهل الفهم وتحديد هيكل وسلوك النظام بحيث تصبح سهلة ومقرؤة من قبل أي شخص وإن لم يكن مبرمج ، سنعرض في القاسم بعض المخططات التي توضح فكرة المشروع الخاص بنا من حيث العناصر أو خط سير العمليات الذي يقوم به.

3.5 مخطط ERD

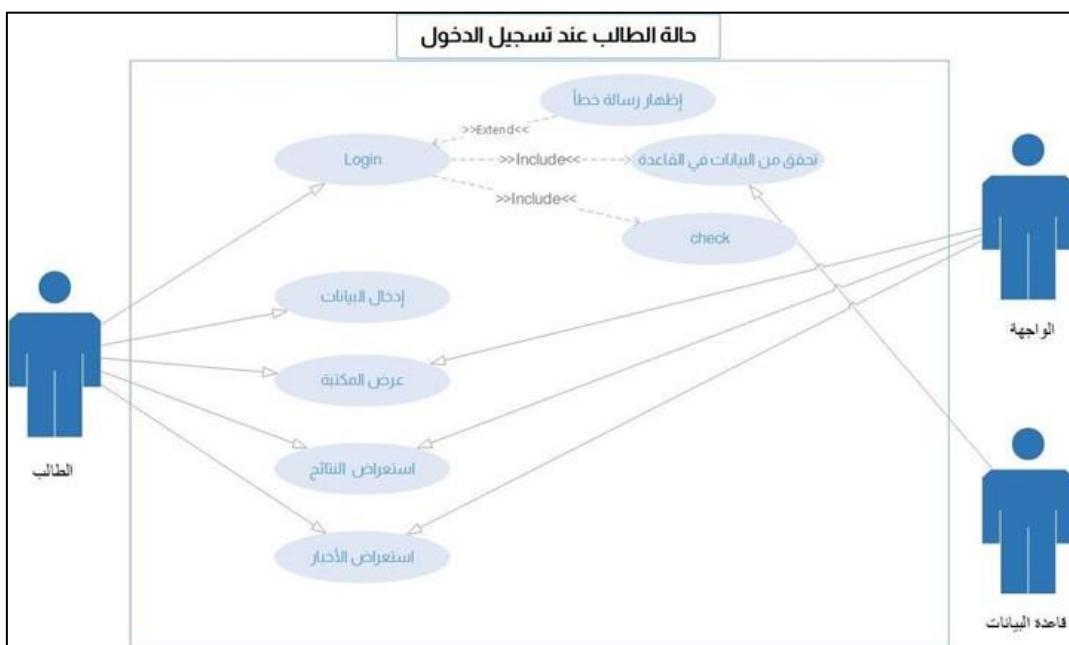
يعرض هذا النوع من المخططات تصوّراً لكيف تبدو قاعدة البيانات مع كل العلاقات التي تربط بين الجداول وبعضها، انظر الشكل (3.2)



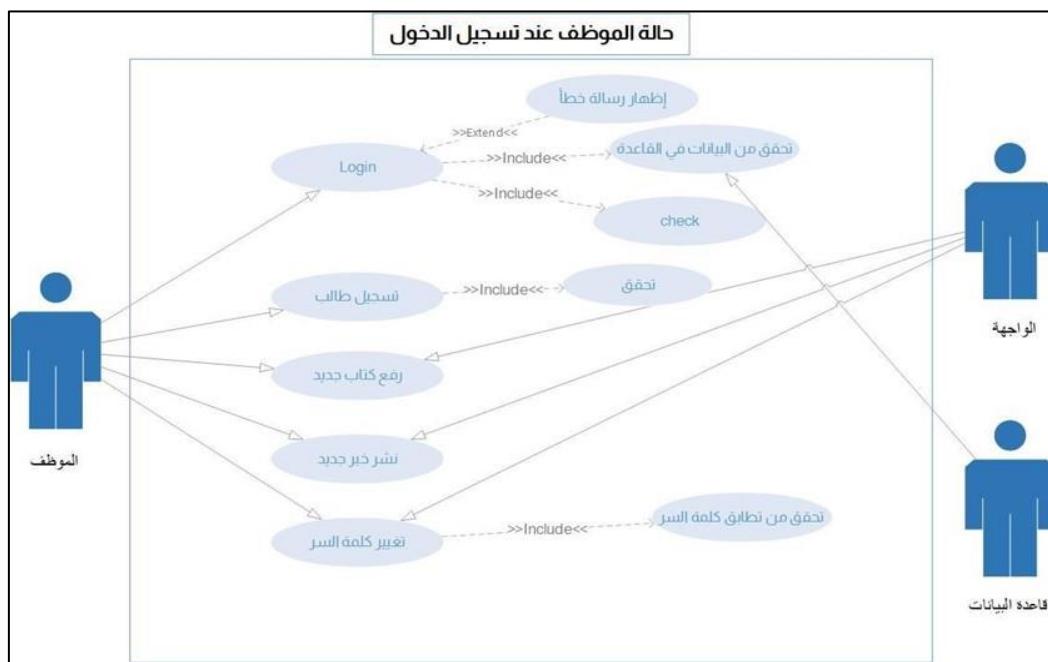
شكل 3.2 : مخطط 3.2 : مخطط

(Use Case Diagram) 3.5.1 مخطط

في هذا المخطط يوضح عملية قيام الطالب بتسجيل الدخول إلى النظام، وإدخال بياناته الشخصية والتعليمية، والتحقق من صحة المعلومات وعرضها



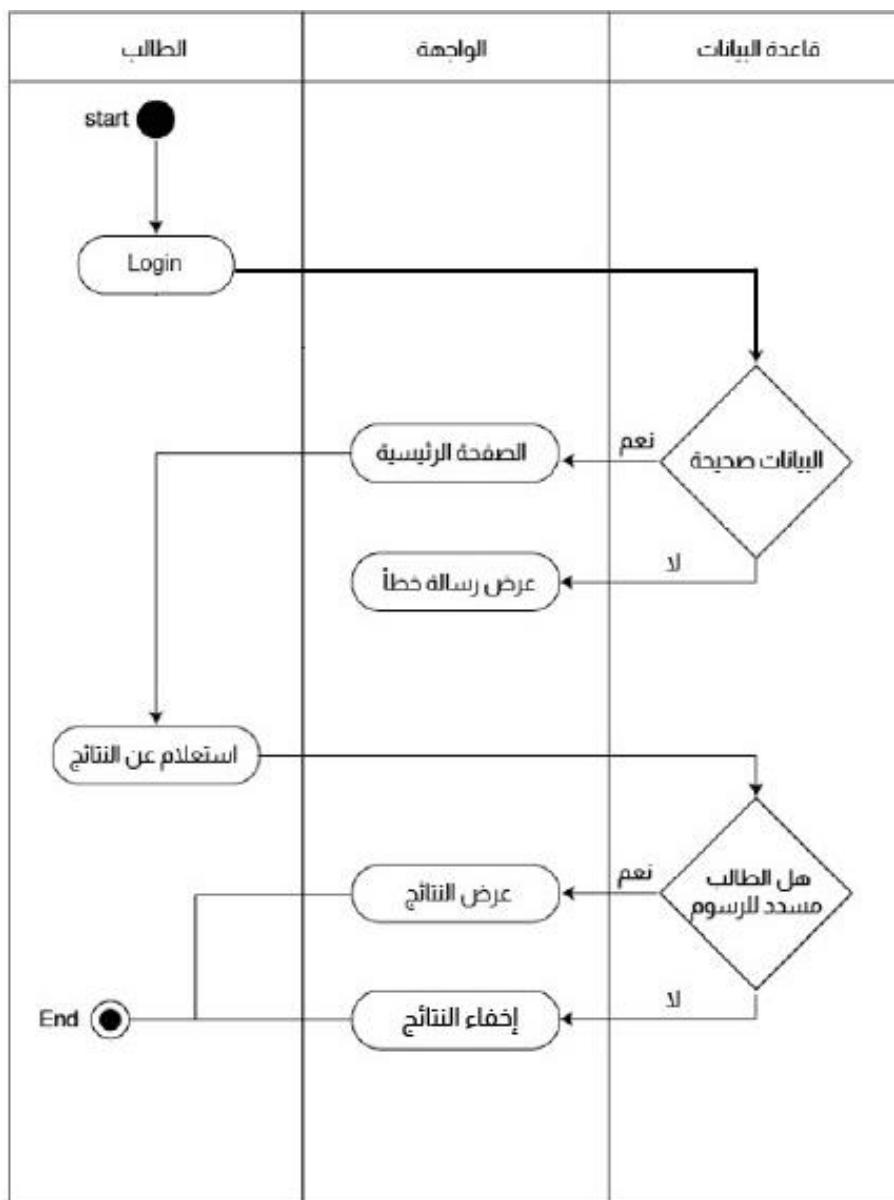
شكل 3.3: عملية قيام الطالب بتسجيل الدخول إلى النظام،



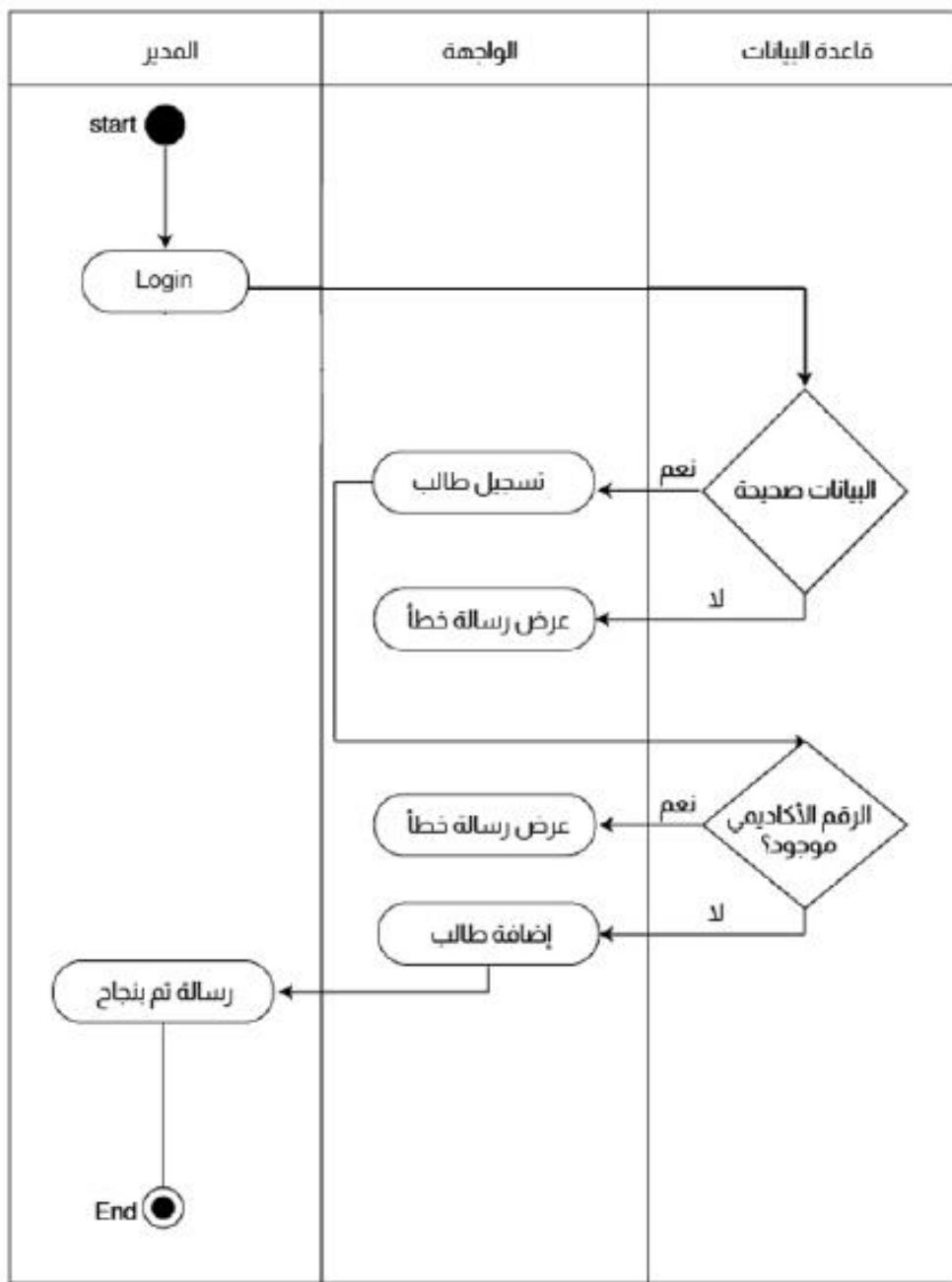
شكل 3.4: عملية قيام الموظف بتسجيل الدخول لنظام

3.5.2 مخطط سير العمليات (Activity Diagram) (Activity Diagram)

يُستخدم مخطط سير العمليات لتمثيل سلسلة من الخطوات التي يتم اتخاذها لإكمال مهمة أو عملية ما. يُظهر المخطط كيف تنتقل المهمة من خطوة إلى أخرى، ويُحدد من المسؤول عن كل خطوة، ويُوضح ما يحدث في كل خطوة.



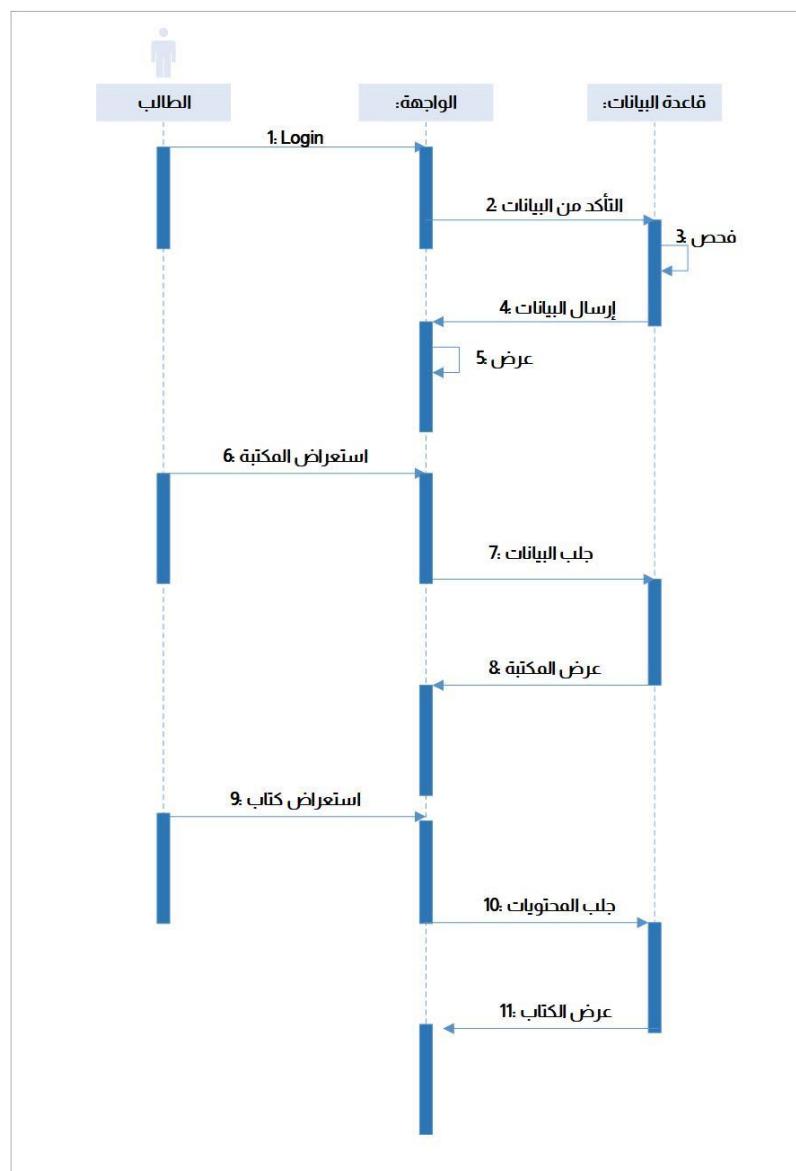
شكل 3.5: مخطط الأنشطة لتطبيق الطالب.



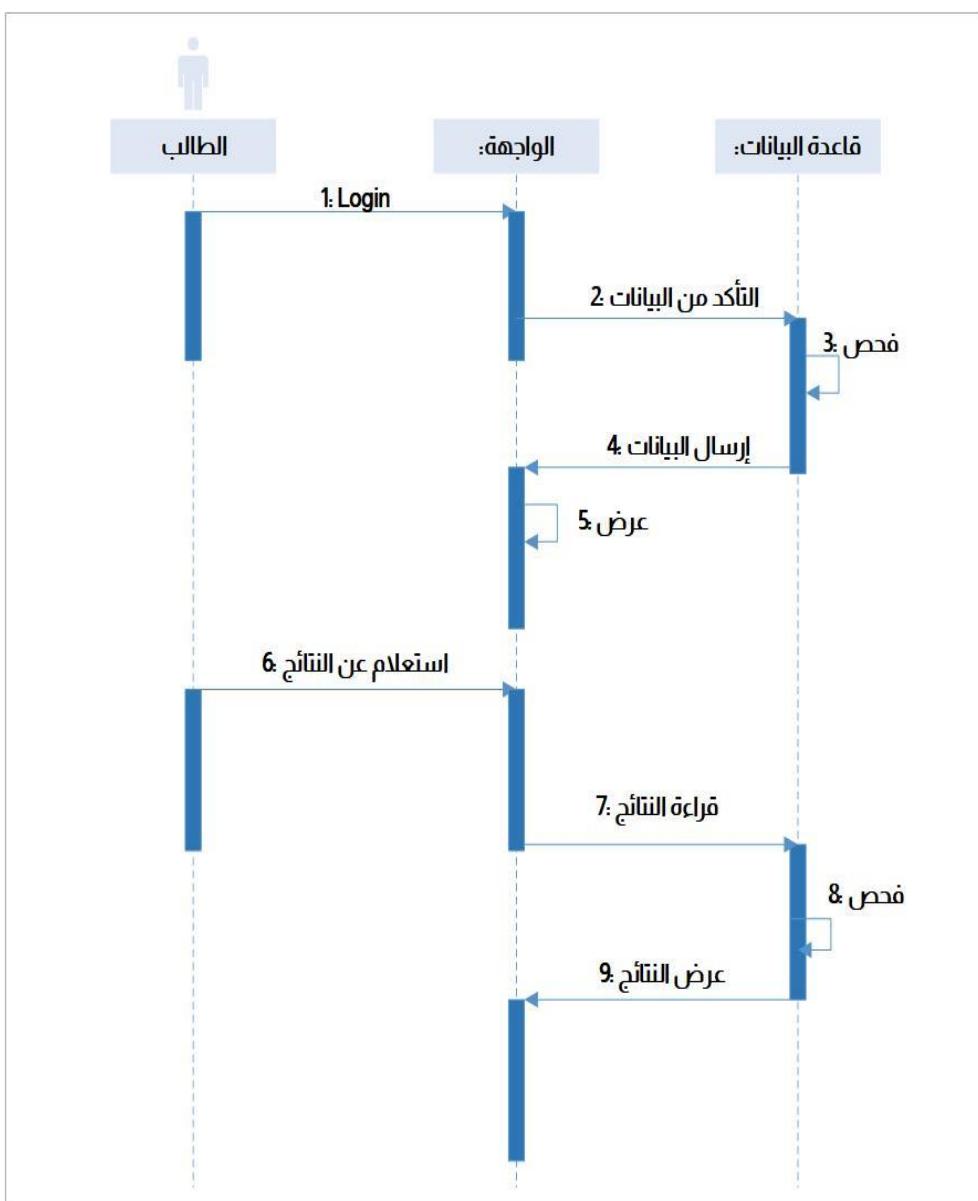
شكل 3.6: مخطط الأنشطة لعملية تسجيل طالب عبر النظام

3.5.3 مخطط التتابع او مخطط التسلسل (Sequence Diagram)

يظهر هذا المخطط Sequence Diagram تسلسل العمليات التي تحدث عند تسجيل الدخول إلى نظام إدارة خدمات بوابة الطالب لمركز اللغات والترجمة في الجامعة، قد تشمل الاصدارات اتصال الطالب بنظام بوابة الطالب، وتحقق النظم من هوية الطالب، وعرض الأخبار وغيرها.



شكل 3.7: مخطط التسلسل لتطبيق الطالب

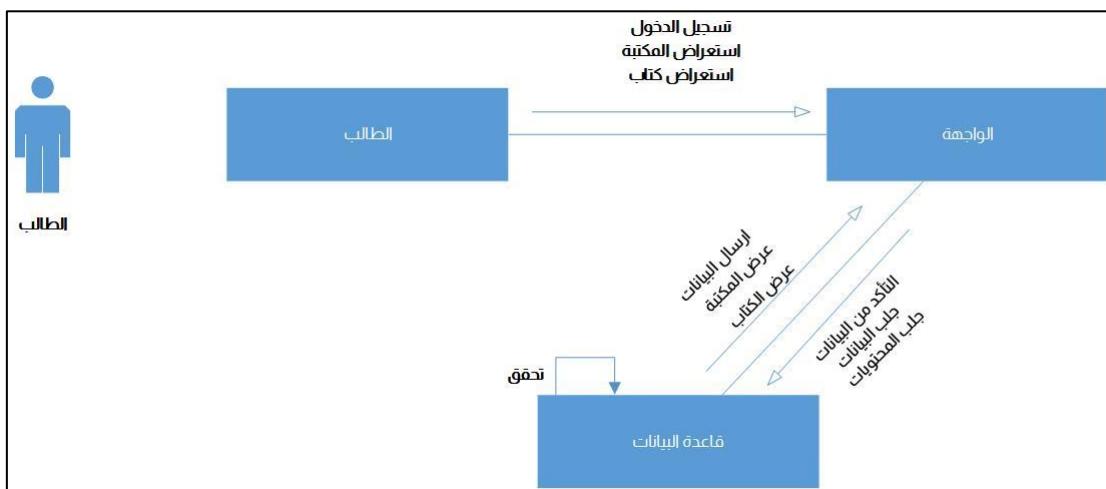


شكل 3.8: مخطط التسلسل للنظام

3.5.4 مخطط التعاون (Collaboration Diagram)

يعد مخطط التعاون شكل آخر من اشكال المخطط الديناميكي والتي يؤكد على تنظيم الكائنات المشاركة في التفاعل.

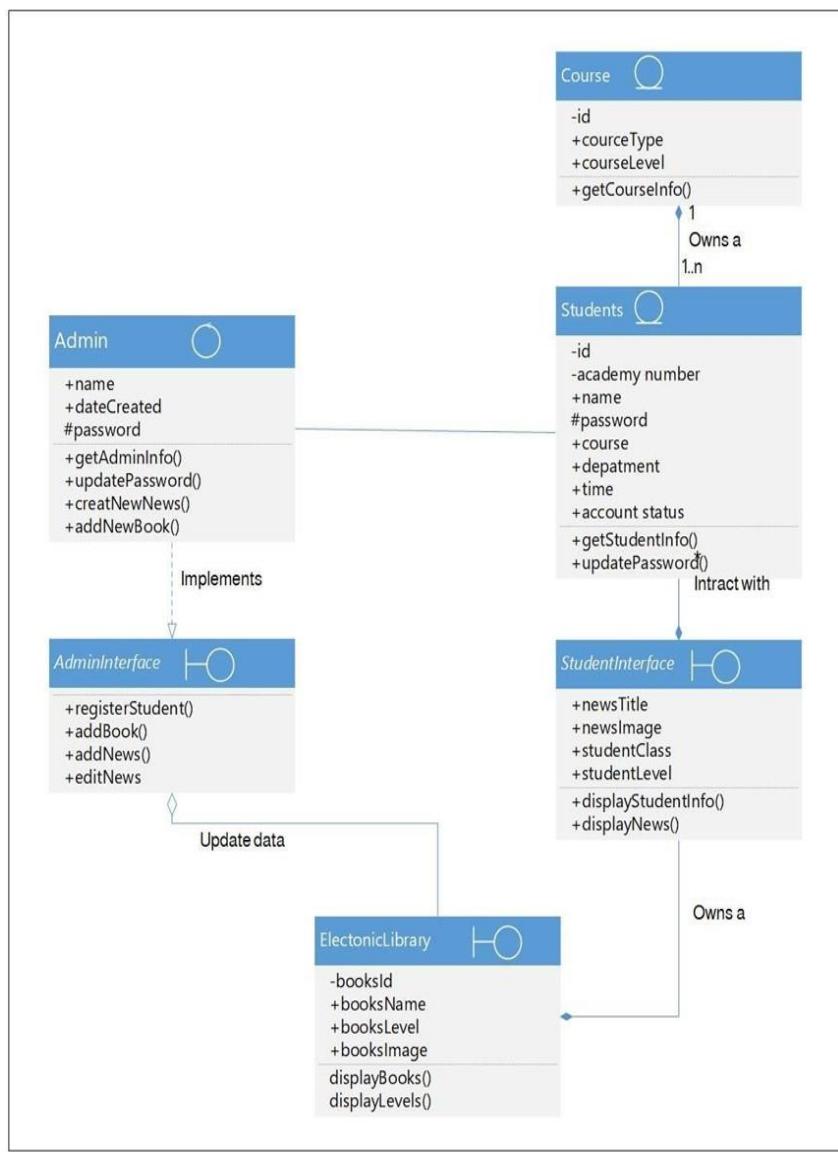
يُظهر هذا المخطط Collaboration Diagram تفاعلات الفئات المختلفة في نظام إدارة خدمات بوابة الطالب لمركز اللغات والترجمة في الجامعة.



شكل 9.9: مخطط التعاون لتطبيق الطالب

3.5.5 مخطط الفئة (Class Diagram)

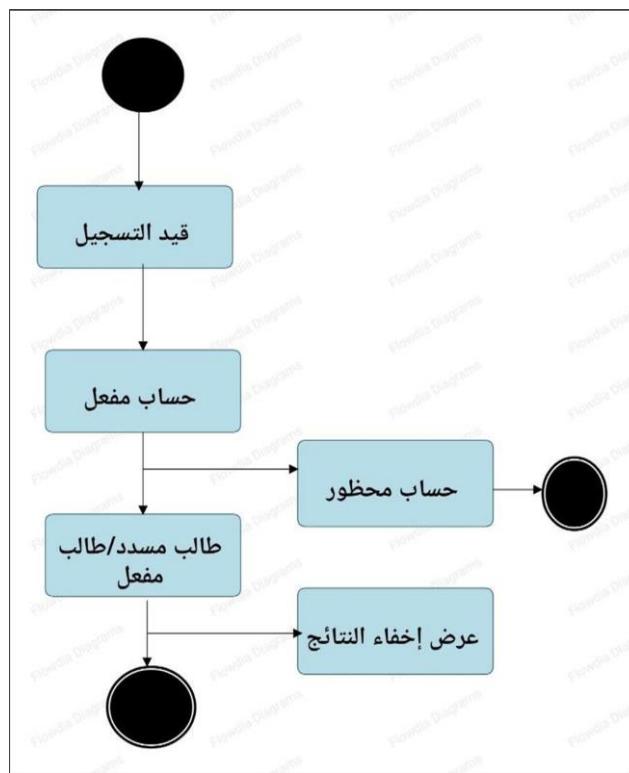
مخطط الفئة (Class Diagram) هو أحد أهم أنواع مخططات UML ويستخدم لتصميم وتوثيق البنية الأساسية لأنظمة البرمجيات الكائنات (Object-Oriented).



شكل 3.10: مخطط class diagram

3.5.6 مخطط الحالة (State Chart Diagram)

يُستخدم مخطط الحالة لتمثيل سلوك كائن ما بمرور الوقت. يُظهر المخطط كيف يتغير سلوك الكائن استجابةً للأحداث الخارجية



شكل 3.11: مخطط حالة للنظام

3.6 الخلاصة

في هذا الفصل قمنا بشرح المنهجية المتبعة في المشروع وهي الـ prototyping ووضمنا مراحلها بالقصص ثم انتقلنا إلى تحليل وتصميم المشروع وقمنا بعرض عدة مخططات منها ERD والخاص بقاعدة البيانات بالإضافة إلى Use case ومخطط التسلسل ومخطط التعاون ومخطط الفئة وأخيراً مخطط الحالة.

الفصل الرابع

4 التنفيذ

4.1 مقدمة

قمنا في مشروعنا باستخدام React js للنظام ، وFlutter للتطبيق، وفي هذا الفصل سنقوم باستعراض بعض الأكواد الخاصة بكلًا من النظام والتطبيق وشرحها وتوضيحها، كما نوضح مالمكاتب والpackages التي تم استخدامها في المشروع وما عملها والفائدة منها، وسنبدأ بدايةً مع التطبيق وواجهاته ثم يليه النظام.

4.2 نظام إدارة شؤون طلاب المركز

قمنا في النظام الخاص بالمركز باستخدام react js والتي تميز بكونها Single Page Application (SPA) بمعنى أن التطبيق يعمل كصفحة ويب واحدة يتم تحميلها مرة واحدة فقط، وبعد التحميل الأولي يتم تحديث المحتوى ديناميكياً على الصفحة نفسها دون الحاجة إلى إعادة تحميل كامل للصفحة من الخادم.

فوائد React SPA

- أداء أفضل : حيث يقلل من وقت التحميل وإعادة التحميل الكامل للصفحات.
- تجربة مستخدم سلسة : تبديل سريع بين المحتوى والصفحات دون انقطاع.
- تحميل البيانات الجديدة فقط، مما يجعل التفاعل أسرع وأكثر استجابة.

The screenshot shows the 'Student Registration' page of a web application. The main form contains fields for 'Name', 'Academic Number', 'Gender', 'Program', 'Session', and a 'Register' button. Below the form are search and filter buttons ('Search', 'Search by Student', 'Filter'). To the right, a sidebar lists various administrative actions: 'New Student', 'New Course', 'Add Book', 'Edit Record', 'New Program', 'Degree Levels', 'Last Name', 'Activate Windows', and 'Logout'. The sidebar also includes a 'Logout' link at the bottom.

شكل 4.1: واجهة تسجيل طالب جديد من خلال النظام

درازت المدون

العنوان	العنوان	العنوان
العنوان	العنوان	العنوان
ادخل العنوان	ادخل العنوان	ادخل العنوان
ادخل العنوان	ادخل العنوان	ادخل العنوان
تسجيل		

ابحث عن الطالب

ملتقى

#	نسبة/مقدم	تاريخ الإنشاء	الجنس	اسم الطالب	الرقم الريادي
X	نشيط	2024-03-27 13:31:43	أنثى	لسميم لطف الله الفراصي	2024111111
X	نشيط	2024-04-04 20:49:06	ذكر	عمر لطف الله الفراصي	2024112211
G	م Freed	2024-04-09 17:23:32	أنثى	زياب الفراصي	20241515
X	نشيط	2024-04-20 22:45:29	أنثى	رهام الفراصي	2023111111
X	نشيط	2024-06-25 17:09:12	أنثى	سمينة عبدالماهر الفناطي	2024151478

Activate Windows
Go to Settings to activate Windows.

الصفحة السابقة | الصفحة التالية

شكل 4.2: تابع واجهة تسجيل طالب جديد



شكل 4.3: عرض خير تم نشره من لوحة النظام

النرم
اسم الدورة
اسم الطالب
المبلغ المدفوع
المبلغ المدفوع
المبلغ المدفوع
اسم الدورة
اسم الترم
اسم الطالب

#	اسم الدورة	اسم الترم	اسم الطالب
١	letracy(A)	الثلاثون	رباب الفراصي
٢	letracy(B)	الواحد والثلاثين	نسرين لطف الله الفراصي
٣	letracy(C)	الواحد والثلاثين	نسرين لطف الله الفراصي

شكل 4.4: واجهة الدفع

4.2.1 صفحة تسجيل طالب

اسم الطالب
اسم الترم
اسم البرنامج
اسم الدورة
اسم الشعبة
اسم الشعبة

#	اسم الدورة	اسم البرنامج	اسم الترم	اسم الطالب
١	letracy(A)	اعتبادي	الثلاثون	رباب الفراصي
٢	letracy(A)	اعتبادي	الثلاثون	نسرين لطف الله الفراصي
٣	letracy(B)	اعتبادي	الواحد والثلاثين	نسرين لطف الله الفراصي

شكل 4.5

قمنا في صفحة تسجيل طالب بإضافة عدة حقول مثل: اسم الطالب، الرقم الأكاديمي، البرنامج، الدورة، الترم -انظر الشكل(4.5)- ثم قمنا بالتعامل معها في الباك اند عن طريق لغة PHP ، حيث كانت عملية الربط بين الواجهة والبرمجة الخلفية تتم كالتالي:



```
const fetchLevelsByProgram = (programId) => {
  axios.get(`http://localhost/languagecentersystem/getLevels.php?program_id=${programId}`)
    .then(response => {
      setFilteredLevels(response.data);
    })
    .catch(error => {
      console.error('حدث خطأ أثناء جلب البيانات', error);
    });
};
```

شكل 4.6

ال코드 في الشكل(4.6) يوضح طريقة جلب المستويات الخاصة بكل برنامج، حيث يجب على مدير النظام ان يحدد نوع البرنامج أولاً ثم ينتقل للحقل الخاص باسم الدورة حيث أن لكل برنامج مستويات خاصة به مختلفة بالعدد والمسمي، وكما يظهر في الكود فقد قمنا بعملية get من الخادم ومررنا له رقم الـid الخاص بالبرنامج الذي قام مدير النظام باختيارة ليتم جلب كافة المستويات المرتبطة به عن طريق المفتاح الأجنبي في جدول المستويات بقاعدة البيانات.



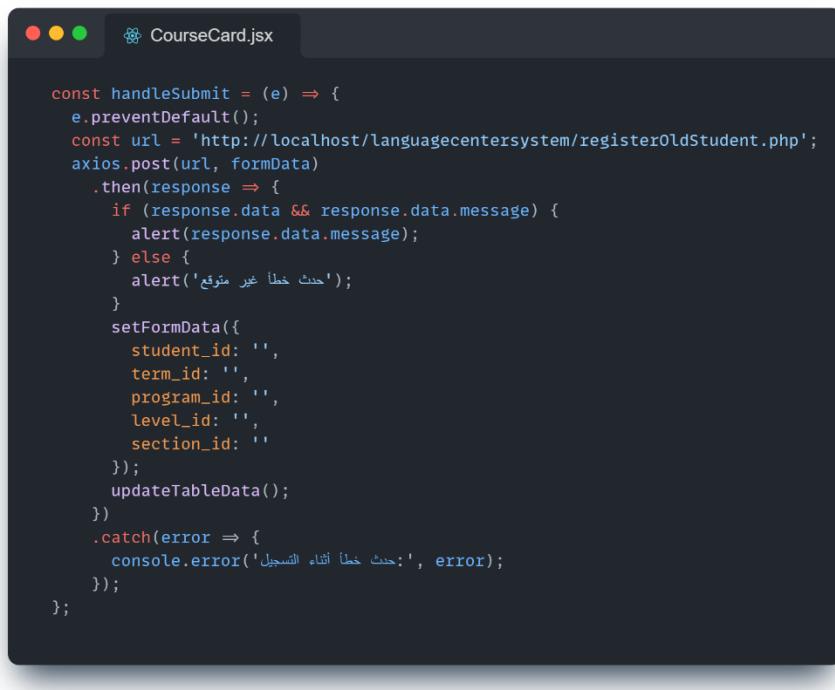
```
<?php
include "./connect.php";

$program_id = isset($_GET['program_id']) ? $_GET['program_id'] : '';
if ($program_id) {
    $stmt = $con->prepare("SELECT prlevelid, levelsName FROM program_levels WHERE program_id = ?");
    $stmt->execute([$program_id]);
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($rows);
}
?>
```

شكل 4.7 كود جلب المستويات حسب البرنامج

كما يظهر في الشكل(4.7) فقد قمنا في البداية باستدعاء ملف الاتصال بقاعدة البيانات والذي قمنا بإنشاؤه مسبقاً ضمن ملف منفصل وقمنا بتسميته connect.php ، ثم عرفنا متغير جديد \$program_id والذي يستخدم \$_GET للحصول على قيمة program_id -والذي سبق وأن مررناه من الواجهة- ، إذا لم يكن هناك program_id ، يتم تعين قيمة فارغة.(")

بعد ذلك قمنا بإضافة شرط يقوم بفحص program_id إذا كان قد تم تمريره فيتم تنفيذ الاستعلام بجلب كافة المستويات المرتبطة بذات البرنامج المختار.



```
const handleSubmit = (e) => {
  e.preventDefault();
  const url = 'http://localhost/languagecentersystem/registerOldStudent.php';
  axios.post(url, formData)
    .then(response => {
      if (response.data && response.data.message) {
        alert(response.data.message);
      } else {
        alert('حدث خطأ غير متوقع');
      }
      setFormData({
        student_id: '',
        term_id: '',
        program_id: '',
        level_id: '',
        section_id: ''
      });
      updateTableData();
    })
    .catch(error => {
      console.error('حدث خطأ أثناء التسجيل:', error);
    });
};
```

شكل 4.8: دالة إرسال بيانات الطالب

بالعودة إلى الواجهة يجب أن نوضح طريقة تفاعل الصفحة مع المستخدم عند إنهاءه تعبئة الحقول والنقر على زر التسجيل، كما يظهر في الكود بالشكل (4.8) قمنا بإنشاء دالة handleSubmit تستخدم للتعامل مع إرسال نموذج البيانات (form) ، وتعتمد الدالة على مكتبة axios لإرسال طلب POST إلى خادم ويب.

e.preventDefault()

تُستخدم هذه الدالة لمنع الإجراء الافتراضي بإعادة تحميل الصفحة عند عملية إرسال النموذج، مما يسمح بإرسال النموذج دون إعادة تحميل الصفحة.

`const url = 'http://localhost/languagecentersystem/registerOldStudent.php'`

يتم تحديد عنوان URL الذي سيتم إرسال طلب POST إليه.

`axios.post(url, formData)`

يتم استخدام مكتبة axios لإرسال طلب POST إلى الخادم،(formData هو الكائن الذي يحتوي على البيانات التي سيتم إرسالها إلى الخادم –سيتم التطرق إليه لاحقاً).

.then(response => { ... })

عند نجاح طلب POST ، يتم تنفيذ الكود داخل الدالة then ، الكود داخل then يتعامل مع الاستجابة من الخادم.

setFormData({ ... });

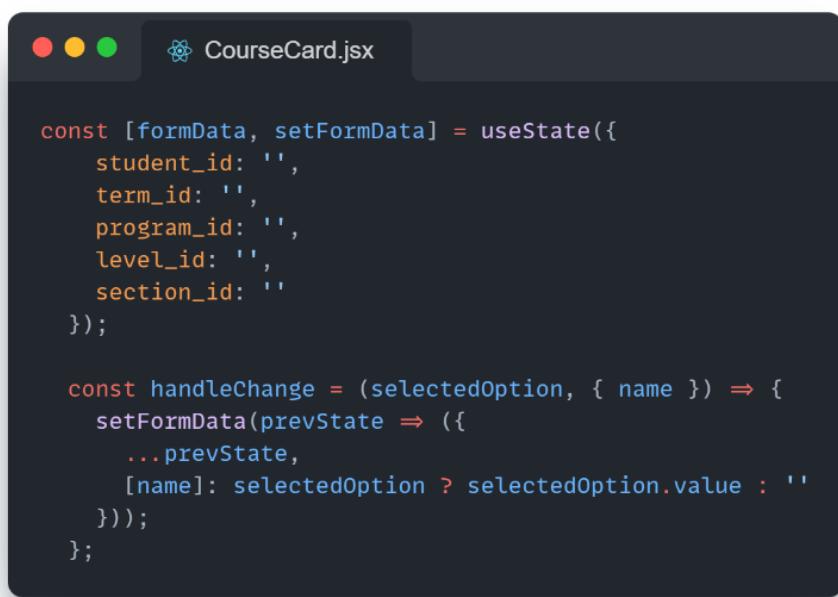
يتم استخدام هذه الدالة لإعادة تعيين حالة formData إلى قيمة فارغة، أي تفريغ حقول الإدخال في واجهة المستخدم.

updateTableData();

يتم استدعاء هذه الدالة لتحديث بيانات الجدول في الواجهة، ربما لإظهار التغييرات الجديدة بعد تسجيل الطالب.

.catch(error => { ... }):

إذا حدث خطأ أثناء إرسال الطلب، يتم تنفيذ الكود داخل catch. ، الكود يعرض الخطأ ضمن(console.error) ، مما يساعدنا في تصحيح الأخطاء.



```
CourseCard.jsx

const [formData, setFormData] = useState({
  student_id: '',
  term_id: '',
  program_id: '',
  level_id: '',
  section_id: ''
});

const handleChange = (selectedOption, { name }) => {
  setFormData(prevState => ({
    ...prevState,
    [name]: selectedOption ? selectedOption.value : ''
  }));
};
```

شكل 4.9: كود التعامل مع حقول الإدخال

يوضح الكود في الشكل (4.9) طريقة التعامل مع حالة النموذج (form state) في React ويستخدم الدالة handleChange لتحديث هذه الحالة بناءً على تغييرات المستخدم. لنوضح الآن أجزاء الكود بشكل أكثر تفصيلاً:

useState هو هوك في React يُستخدم لإدارة الحالة (state) في المكونات الوظيفية (functional components). يحتوي على القيم الحالية لبيانات النموذج. formData هو متغير حالة (state variable) يحتوي على القيم الحالية لبيانات النموذج.

setFormData هو دالة تُستخدم لتحديث قيمة formData.

في الحالة الأولية يتم تعين قيم افتراضية فارغة لكل من section_id, student_id, term_id, program_id, level_id، حيث انهم أسماء لسمة controlId التي تميز كل form group عن الآخر.

أما فيما يخص دالة handleChange فُستخدم لتحديث حالة النموذج عند حدوث تغييرات في الحقول المختلفة، مثلاً في النموذج الذي يحتوي على حقل اختيار اسم الطالب (student_id)، عندما يختار المستخدم طالباً سيتم استدعاء الدالة handleChange مع الخيار المحدد واسم الحقل، وستقوم الدالة بتحديث حالة النموذج لتحتوي على القيمة الجديدة للطالب المختار مع الاحتفاظ بالقيم الأخرى دون تغيير، هذا يسمح للنموذج بالتفاعل بشكل ديناميكي مع إدخالات المستخدم وتحديث الحالة بناءً على تغييرات الحقول، الآن لنشرح الكود بشكل أكثر تفصيلاً:

selectedOption

يحتوي على القيمة التي تم اختيارها في الحقل (يتمثل الخيار الذي اختاره المستخدم)

{ name }

يُمثل اسم الحقل الذي تغير (مثلاً "student_id")

setFormData

يقوم بتحديث حالة النموذج.

prevState

يُمثل الحالة السابقة للنموذج.

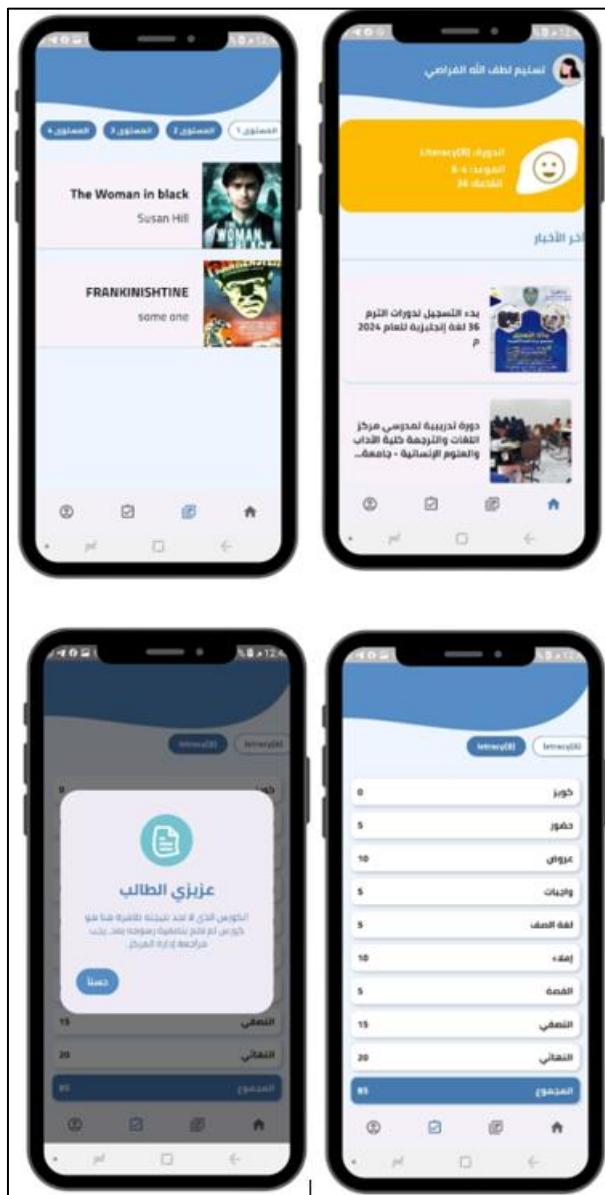
...prevState

يُنسخ جميع القيم الحالية للحالة السابقة.

[name]

يُستخدم لتحديث القيمة الخاصة بالحقل الذي تغير حيث يتم تعين القيمة إلى selectedOption.value إذا كانت متوفرة، أو إلى سلسلة فارغة ("") إذا لم تكن متوفرة، أي أنه متغير يتم تمرير قيمته لاحقاً.

4.3 تطبيق الطالب



شكل 4.10 بعض واجهات التطبيق

4.3.1 صفحة تسجيل الدخول

سنقوم بتقسيم الكود الرئيسي لعدة أقسام لنعطي كل نقطة حقها بالشرح والتفصيل، انظر الشكل (4.11)

```
class Login extends StatelessWidget{
  const Login({Key?key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    Get.put(LoginControllerImp());
    final size = MediaQuery.of(context).size; // الحصول على حجم الشاشة

    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          image: DecorationImage(
            image: AssetImage(AppImageAsset.background), // تحويل السلسلة الى AssetImage
            fit: BoxFit.cover,
          ),
        ),
        child: WillPopScope(
          onWillPop: () async => alertExitApp(), // تمرير دالة بدلاً من استدعائنا
          child: GetBuilder<LoginControllerImp>(
            builder: (controller)=>
            HandlingDataRequest(
              statusRequest: controller.statusRequest,
            )
          ),
        ),
      ),
    );
  }
}
```

شكل 4.11 جزء من كود تسجيل الدخول

كما يظهر في الشكل(4.11) فقد تم تعريف الكلاس Login ك StatelessWidget، مما يعني أن واجهة المستخدم في هذه الفئة لا تتغير بمجرد بنائها.

هو الفئة التي تدير حالة واجهة تسجيل الدخول.

عندما نقوم بإنشاء مثيل جديد، نحن نقوم بإنشاء كائن جديد من هذه الفئة، الكائن الجديد سيكون المسؤول عن إدارة البيانات والسلوكيات الخاصة بواجهة تسجيل الدخول

Scaffold هي ويدجيت التخطيط الرئيسي في Flutter حيث يوفر بنية أو هيكل أساسي للواجهة.

يحتوي على خلفية باستخدام Container، حيث يتم وضع صورة الخلفية باستخدام AssetImage و DecorationImage

WillPopScope تُستخدم لاستدعاء دالة تقوم بعمل معين عند النقر على زر الرجوع على الهاتف المحمول، وهنا قمنا باستدعاء الدالة alertExitApp والتي تقوم بعرض رسالة للمستخدم قبل إغلاق التطبيق، نستعرض كود الدالة في الشكل (4.12)،



```

alertExitApp(){
  Get.defaultDialog(
    title: "alert",
    middleText: "Do you want to get out from this app?",
    actions: [
      ElevatedButton(
        onPressed: (){
          exit(0);
        },
        child: const Text("Yes")),
      ElevatedButton(
        onPressed: (){
          Get.back();
        },
        child: const Text("cancel")),
    ],
  );
  return Future.value((true));
}

```

شكل 4.12: دالة التبیه على الخروج من التطبيق

هذه الدالة لا تأخذ أي معاملات وتعيد كائن Future<bool> ، حيث يتم استخدام Future<bool> لضمان التعامل الصحيح مع العمليات غير المتزامنة ، ولفهم فائدته بشكل أفضل، دعونا نتعمق في مفهوم Future ولماذا هو مهم في هذا السياق.

Dart في Future 4.3.1.1

Future هو نوع يُستخدم للتعامل مع العمليات غير المتزامنة في Dart ، العملية غير المتزامنة هي عملية قد تستغرق بعض الوقت للانتهاء (مثل انتظار استجابة من خادم)، ولكن بدلاً من تجميد التطبيق أثناء الانتظار، يُسمح للتطبيق بالاستمرار في تنفيذ التعليمات البرمجية الأخرى، أي أنه على عكس العمليات المتزامنة لا يقوم

بإعطاء كل عملية وقتاً معيناً تقوم بتنفيذ مهمتها فيه ثم الانتقال للمهمة التي تليها وهكذا، إنما تتم كل العمليات بشكل متزامن مع بعضها مما يؤدي إلى رفع أداء التطبيق وتحسينه.

4.3.1.2 لماذا نستخدم `Future<bool>` في `alertExitApp` ؟

في `WillPopScope` الدالة `onWillPop` تأخذ دالة تُعيد `Future<bool>` هذا يسمح لـ `WillPopScope` بانتظار نتيجة هذه الدالة قبل اتخاذ قرار بشأن السماح للمستخدم بالعودة (المغادرة) أو البقاء في الصفحة.

لند الآن لكود الدالة `alertExitApp` ونشرح ما تبقى:

عرض حوار التبيه باستخدام : `Get.defaultDialog` وهو جزء من حزمة `GetX` لعرض مربعات حوار تتبّعه بسيطة.

المعاملات المستخدمة في `Get.defaultDialog` هي:

"`title` عنوان الحوار ، في هذه الحالة هو"

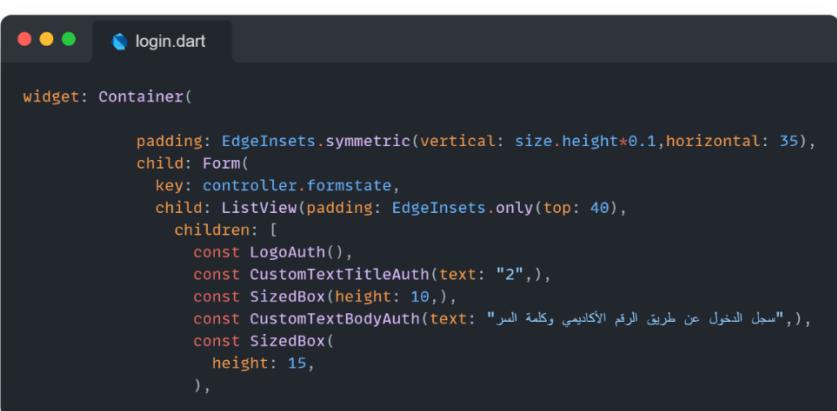
"`Do you want to get out from this app ، وهو.`" `middleText` النص الرئيسي في مربع الحوار

قائمة من الأزرار التي تمثل الإجراءات التي يمكن للمستخدم اتخاذها.

إذا نقر المستخدم على "Yes" يتم إغلاق التطبيق عبر `exit(0)`

إذا نقر المستخدم على "cancel" يتم إغلاق مربع الحوار فقط باستخدام `(Get.back()`

بالعودة لكود واجهة تسجيل الدخول، انظر الشكل (4.13) الذي يحتوي جزء من بقية كود الواجهة



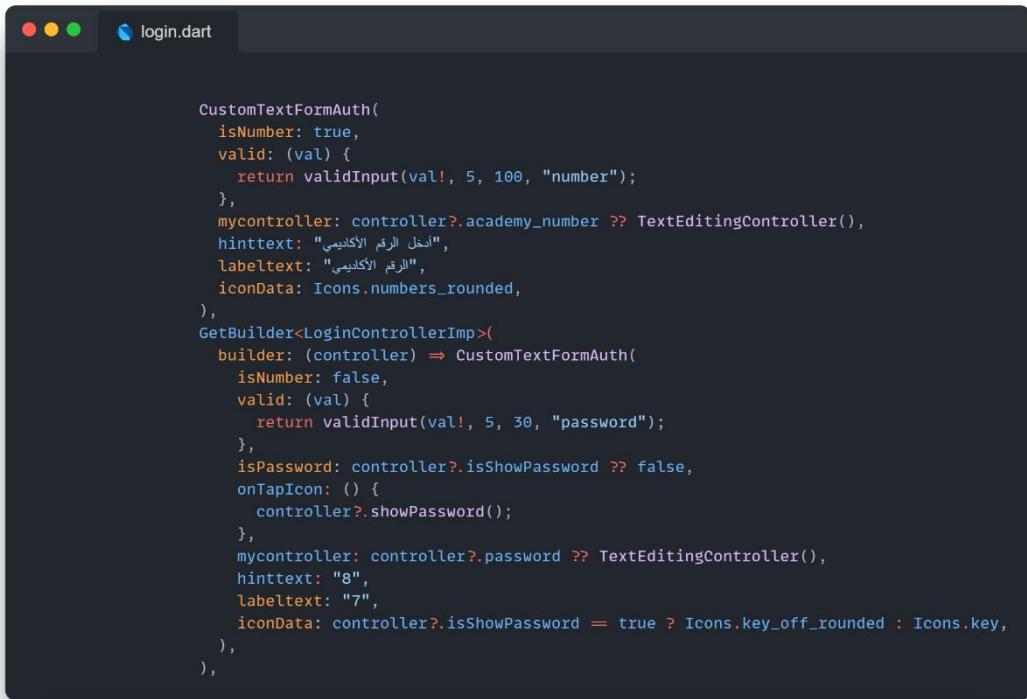
```
login.dart
widget: Container(
  padding: EdgeInsets.symmetric(vertical: size.height*0.1, horizontal: 35),
  child: Form(
    key: controller.formstate,
    child: ListView(padding: EdgeInsets.only(top: 40),
      children: [
        const LogoAuth(),
        const CustomTextTitleAuth(text: "2",),
        const SizedBox(height: 10,),
        const CustomTextBodyAuth(text: "سجل الدخول عن طريق الرقم الأكاديمي وكلمة المرور"),
        const SizedBox(
          height: 15,
        ),
      ],
    ),
  ),
)
```

شكل 4.13: كود تصميم نموذج تسجيل الدخول

في هذا الجزء من الكود قمنا ببرمجة وتصميم نموذج تسجيل الدخول وذلك باستخدام الويديجيت Form ، وحيث أن الـ form يأخذ child واحد فقط فقد قمنا باستخدام ListView وقمنا باستدعاء عدة widgets أنشأناها بأنفسنا في مسار مختلف وتحتوي تنسيقات مختلفة.

نلاحظ أننا قمنا بتمرير رقم "2" إليها، حيث أن الرقم "2" يشير إلى نص معين في كلاس يدعى MyTranslation ، حيث يقوم هذا الكلاس باستخدام حزمة GetX في Flutter لإدارة تعدد اللغات في التطبيق، وسنستعرض الكود الخاص به لاحقاً، أما الآن فنكمel مع باقي أجزاء الكود المعروض. ويدجت **SizedBox** يُستخدم لإضافة مساحة فارغة بين العناصر.

ويجت **CustomTextBodyAuth** مخصص لعرض نص وصفي مع النص "سجل الدخول عن طريق الرقم الأكاديمي وكلمة السر".



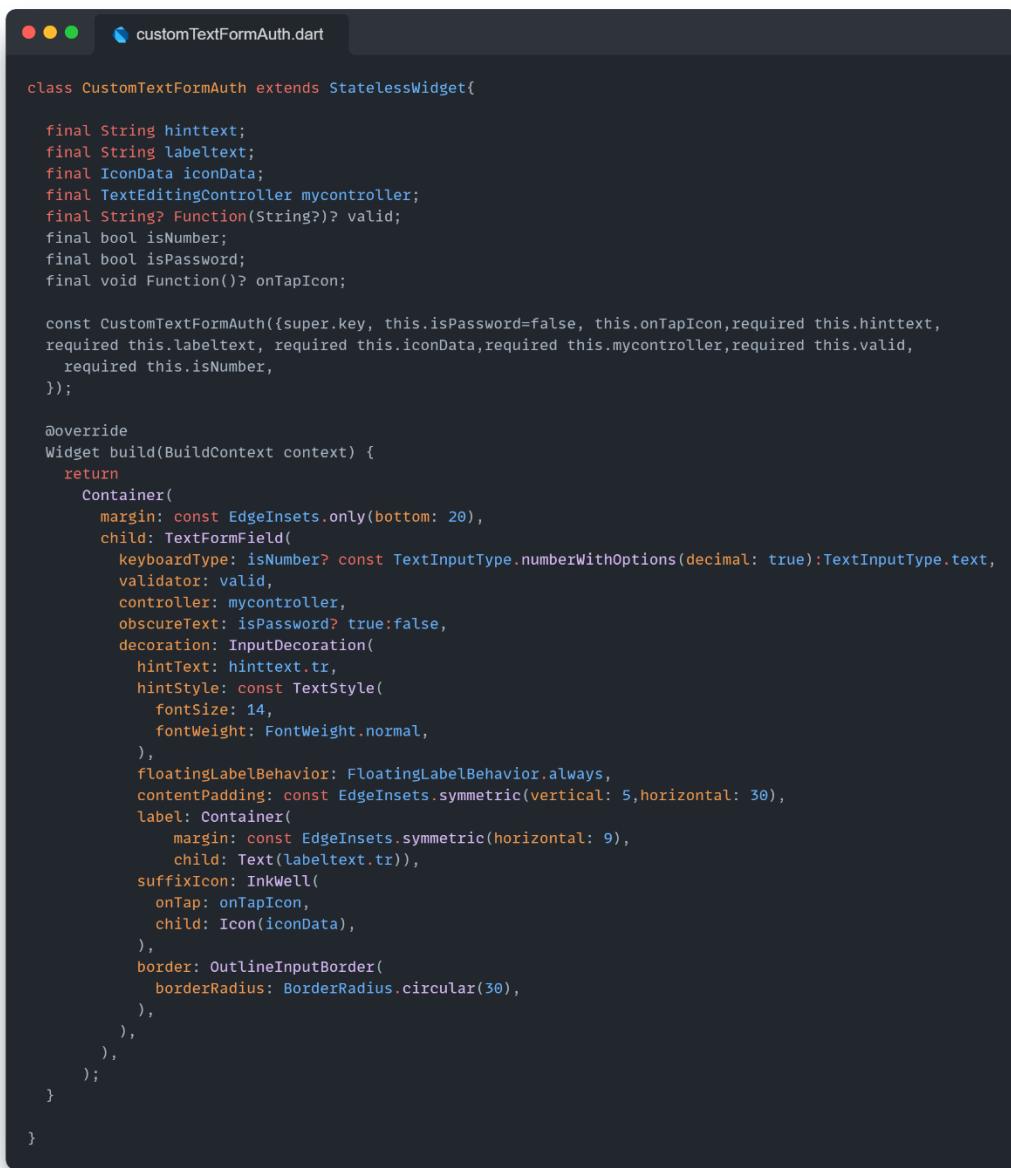
```
CustomTextFormAuth(
    isNumber: true,
    valid: (val) {
        return validInput(val!, 5, 100, "number");
    },
    mycontroller: controller?.academy_number ?? TextEditingController(),
    hinttext: "أدخل الرقم الأكاديمي",
    labeltext: "الرقم الأكاديمي",
    iconData: Icons.numbers_rounded,
),
GetBuilder<LoginControllerImp>(
    builder: (controller) => CustomTextFormAuth(
        isNumber: false,
        valid: (val) {
            return validInput(val!, 5, 30, "password");
        },
        isPassword: controller?.isShowPassword ?? false,
        onTapIcon: () {
            controller?.showPassword();
        },
        mycontroller: controller?.password ?? TextEditingController(),
        hinttext: "8",
        labeltext: "7",
        iconData: controller?.isShowPassword == true ? Icons.key_off_rounded : Icons.key,
    ),
),
```

شكل 4.14: تكملة للكود واجهة تسجيل الدخول

بالنظر إلى الشكل (4.14) ، فـ **CustomTextFormAuth** هو ويدجت مخصص لإنشاء حقل إدخال يتضمن مجموعة من الخصائص لتحديد نوع المدخلات والتحقق من صحتها والتحكم بها، وقد قمنا بإنشاؤه أيضاً ضمن

مسار آخر ثم تمت عملية استدعاءه هنا، لاحظ أن عملية فصل بعض أجزاء الواجهة تجعل الكود مفهوماً ومنظماً وقابلًا للتعديل سواءً من قبل المبرمج نفسه أو من قبل أي مطور بعده.

لنستعرض الآن كود الكلاس CustomTextFormAuth في الشكل (4.15)



```
customTextFormAuth.dart

class CustomTextFormAuth extends StatelessWidget{

    final String hinttext;
    final String labeltext;
    final IconData iconData;
    final TextEditingController mycontroller;
    final String? Function(String?)? valid;
    final bool isNumber;
    final bool isPassword;
    final void Function()? onTapIcon;

    const CustomTextFormAuth({super.key, this.isPassword=false, this.onTapIcon,required this.hinttext,
    required this.labeltext, required this.iconData,required this.mycontroller,required this.valid,
    required this.isNumber,
    });

    @override
    Widget build(BuildContext context) {
        return
            Container(
                margin: const EdgeInsets.only(bottom: 20),
                child: TextFormField(
                    keyboardType: isNumber? const TextInputType.numberWithOptions(decimal: true):TextInputType.text,
                    validator: valid,
                    controller: mycontroller,
                    obscureText: isPassword? true:false,
                    decoration: InputDecoration(
                        hintText: hinttext.tr,
                        hintStyle: const TextStyle(
                            fontSize: 14,
                            fontWeight: FontWeight.normal,
                        ),
                        floatingLabelBehavior: FloatingLabelBehavior.always,
                        contentPadding: const EdgeInsets.symmetric(vertical: 5,horizontal: 30),
                        label: Container(
                            margin: const EdgeInsets.symmetric(horizontal: 9),
                            child: Text(labeltext.tr)),
                        suffixIcon: InkWell(
                            onTap: onTapIcon,
                            child: Icon(iconData),
                        ),
                        border: OutlineInputBorder(
                            borderRadius: BorderRadius.circular(30),
                        ),
                    ),
                ),
            );
    }
}
```

شكل 4.15: الكود الخاص بحقول الإدخال والقابل لإعادة الاستخدام

كما يظهر في الشكل (4.15) فإن دالة البناء في الكلاس تتطلب عدة متغيرات أساسية، منها المطلوبة عند الاستدعاء، ومنها التي يمكن تحديتها ويمكن تركها كما هي وقت الاستدعاء، وذلك حسب حدث معين مثل

حدث النقر كما في حالتنا هذه والخاصة باخفاء أو إظهار كلمة السر عند قيام المستخدم بكتابتها في الحقل الخاص بها.

يحدد إذا كان الحقل يقبل إدخال الأرقام فقط. وهنا تم تعينه إلى `true` مما يعني أن الحقل سيقبل الأرقام فقط.

Valid دالة للتحقق من صحة الإدخال. هنا يتم استدعاء دالة `validInput` التي تتطلب قيمة الإدخال وطول الحد الأدنى والأقصى ونوع الإدخال والذي قد يكون رقمًا `number` فیتم تنفيذ الكود في الشكل(4.16)

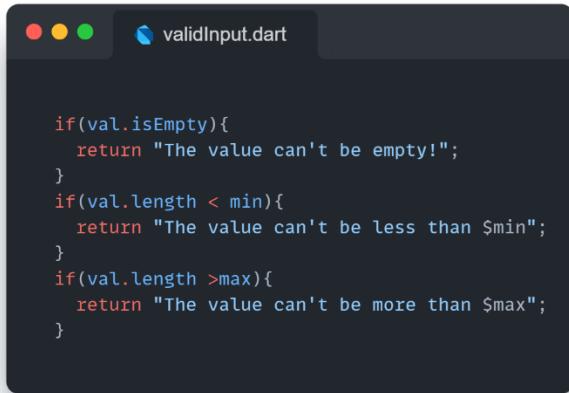


```
validInput.dart

if(type == "number"){
    if(!GetUtils.isNum(val)){
        return "not valid academy number";
    }
}
```

شكل 4.16: كود التحقق من نوع محتوى حقل الإدخال

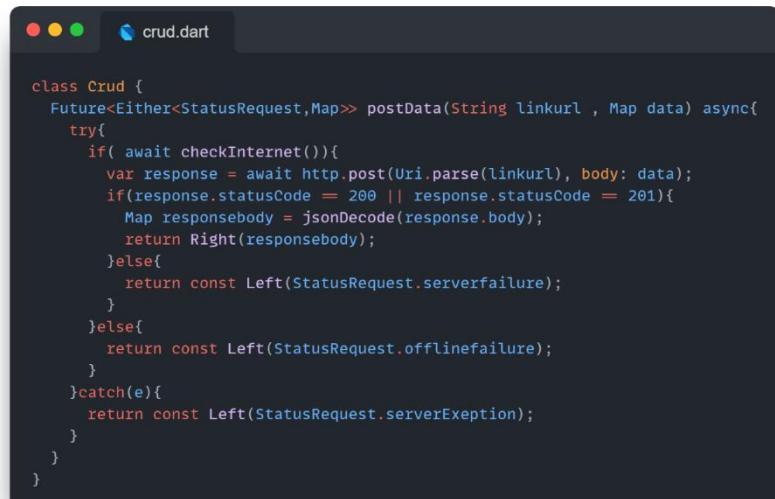
ثم يتم التحقق من أقل وأكبر عدد يمكن إدخاله للحقل عن طريق المعاملات `min` و `max` كما في الشكل(4.17)



```
if(val.isEmpty){
    return "The value can't be empty!";
}
if(val.length < min){
    return "The value can't be less than $min";
}
if(val.length > max){
    return "The value can't be more than $max";
}
```

شكل 4.17: كود التحقق في الواجهة

4.3.2 طلب HTTP ومعالجة الاستجابة



```
class Crud {
  Future<Either<StatusRequest, Map>> postData(String linkurl , Map data) async{
    try{
      if( await checkInternet()){
        var response = await http.post(Uri.parse(linkurl), body: data);
        if(response.statusCode == 200 || response.statusCode == 201){
          Map responsebody = jsonDecode(response.body);
          return Right(responsebody);
        }else{
          return const Left(StatusRequest.serverfailure);
        }
      }else{
        return const Left(StatusRequest.offlinefailure);
      }
    }catch(e){
      return const Left(StatusRequest.serverException);
    }
  }
}
```

شكل 4.18 كود معالجة الـ response

يبين الشكل (4.18) كود التعامل مع الخادم من خلال ارسال طلب http post وهو عبارة عن دالة قمنا بتسميتها postData ومررنا إليها معاملين هما linkurl من النوع String و data من النوع Map .. حيث ان الأول هو رابط الـ api بمساره الكامل الى الملف الذي يتم التعامل فيه مع جدول قاعدة البيانات المراد التعامل مع

بياناته ، أما الثاني فهي البيانات التي نريد إرسالها للخادم، قمنا بإضافة شرط التحقق من الاتصال عن طريق بالإضافة للتحقق من نجاح العملية حيث إذا كان رمز الحالة غير 200 أو 201، يتم إرجاع checkInternet خطأ يشير إلى فشل الخادم باستخدام .Left(StatusRequest.serverfailure)

في الكود الآتي سنوضح كيف قمنا بعملية استدعاء الدالة postData من الكلاس Crud وكيف قمنا باستخدامه.



```
class LoginData {
    Crud crud;
    LoginData(this.crud);
    postData(String academy_number ,String users_password) async {
        var response = await crud.postData(AppLink.login, {
            "academy_number" : academy_number ,
            "users_password" : users_password
        });
        return response.fold((l) => l, (r) => r);
    }
}
```

شكل 4.19: كود ارسال بيانات المستخدم للخادم للتحقق من صحتها

بالنظر إلى الكود نجده واضح وسهل حيث بالبداية قمنا بإنشاء كائن من Crud ثم عرفنا دالة البناء الخاصة بالكلas ومررنا لها الكائن crud، ثم عرفنا دالة postData والتي تقوم بإرسال بيانات تسجيل الدخول (رقم الأكاديمية وكلمة المرور) إلى الخادم للتحقق من صحة المعلومات، بداخل الدالة قمنا باستدعاء الدالة postData من خلال الكائن crud ومررنا لها الرابط والبيانات الازمة وقمنا ب تخزين النتيجة ضمن متغير response ، ثم الكود في آخر سطر يفحص قيمة ال response :

إذا كانت العملية فاشلة أي Left، سيتم إرجاع القيمة الفاشلة.

إذا كانت العملية ناجحة أي Right، سيتم إرجاع القيمة الناجحة.

4.3.3 إدارة خدمات التطبيق

من الخدمات التي قمنا باستخدامها في التطبيق هو ال Shared preference وال Firebase ، حيث استخدمنا ال Firebase في إرسال الاشعارات لمستخدمين التطبيق (الطلاب) عن طريق واجهة النظام، أما ال Shared preference فلتخزين بعض المعلومات المهمة عن المستخدم والتي تظل مخزنة حتى يقوم بتسجيل الخروج من التطبيق.



```
class MyServices extends GetxService{
    late SharedPreferences sharedpreferences;

    Future<MyServices> init() async{
        await Firebase.initializeApp();
        sharedpreferences= await SharedPreferences.getInstance();
        return this;
    }

    initialServices() async{
        await Get.putAsync(() => MyServices().init());
    }
}
```

شكل 4.20: استخدام الفاييربيس وال Shared preference بالتطبيق

ال코드 الظاهر في الشكل(4.20) يقوم بإعداد خدمة مخصصة لإدارة خدمات التطبيق مثل Firebase و SharedPreferences ، حيث قمنا في البداية بإنشاء كلاس وقمنا بسميه MyServices وجعلناه يرث من GetxServices حيث أنه جزء من حزمة GetX يُستخدم لتعريف خدمات تحتاج إلى التهيئة مرة واحدة فقط وتظل مخزنة محليًا طالما كان التطبيق قيد التشغيل.

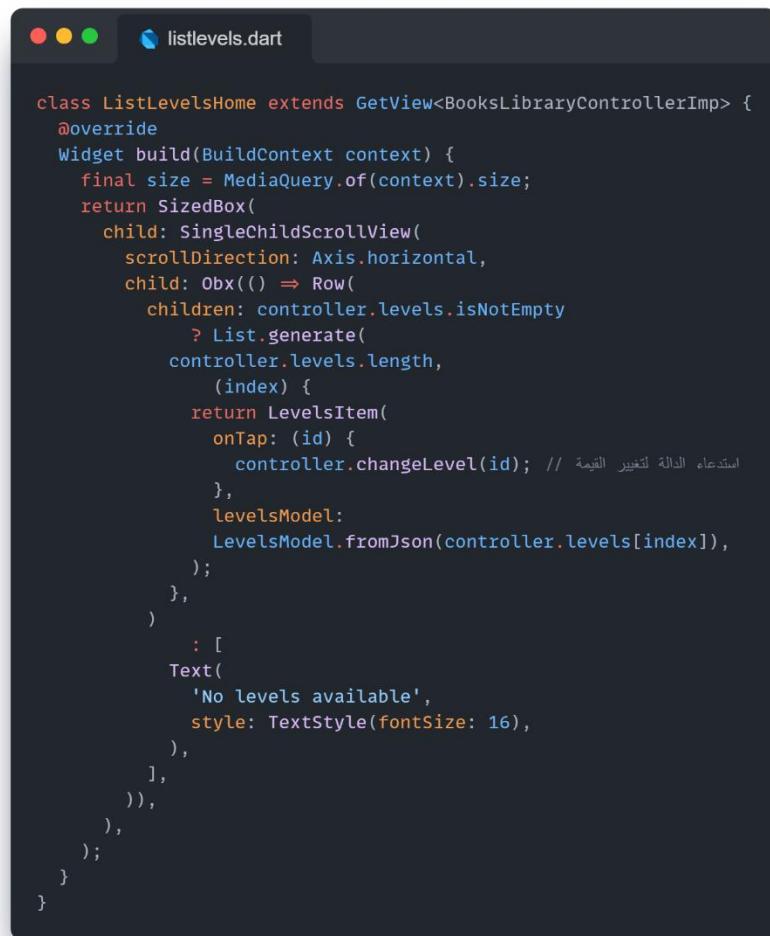
GetxService 4.3.3.1

"هذه الفئة تشبه GetxController، وتشارك معها نفس دورة الحياة ()onInit() و ()onReady() و ()onClose()" لكن ليس لديها "منطق" بداخلها. تقوم فقط بإخطار نظام GetX Dependency Injection بأن هذا الفرع لا

يمكن إزالته من الذاكرة، لذلك فهي مفيدة للغاية لحفظ على "الخدمات" الخاصة بك قابلة للوصول دائمًا ونشطة مع `(Get.find().()` (العثماني، 2023).

4.3.4 المكتبة الإلكترونية

قمنا ببرمجة صفحة منفصلة للمكتبة تحتوي على عدة مستويات مرقمة تصاعديًا حيث كلما كان المستوى أقل كلما كانت لغة الكتاب أو القصة أسهل لطلاب اللغات، وكلما كان أكبر كلما زادت الصعوبة.



```
class ListLevelsHome extends GetView<BooksLibraryControllerImp> {
  @override
  Widget build(BuildContext context) {
    final size = MediaQuery.of(context).size;
    return SizedBox(
      child: SingleChildScrollView(
        scrollDirection: Axis.horizontal,
        child: Obx(() => Row(
          children: controller.levels.isNotEmpty
            ? List.generate(
                controller.levels.length,
                (index) {
                  return LevelsItem(
                    onTap: (id) {
                      controller.changeLevel(id); // استدعاء الدالة لتعديل القيمة
                    },
                    levelsModel:
                    LevelsModel.fromJson(controller.levels[index]),
                  );
                },
            )
            : [
              Text(
                'No levels available',
                style: TextStyle(fontSize: 16),
              ),
            ],
        )),
    );
  }
}
```

شكل 4.21: بناء صف من المستويات

كما هو موضح في الشكل(4.21) فقد قمنا بإنشاء كلاس ListLevelsHome وهو ممتد من `<GetView>` حيث انه عبارة عن ويدجت من حزمةGetX BooksLibraryControllerImp يوفر وسيلة سهلة للوصول إلى الـ Controller إذ أننا لن نحتاج لكتابه الأمر التالي:

```
final controller = Get.find<BooksLibraryControllerImp>();
```

في أي مكان داخل الـ ويدجت لأن `GetView` يقوم بذلك تلقائياً، مما يجعل الكود أصغر وأقل تعقيداً.

وبالعودة للكود وتحديداً في جزئية توليد الـ `List` نلاحظ أننا قمنا بإضافة شرط حيث يتم فحص المتغير `levels` والذي هو من النوع `List`، إذا كان غير فارغ فيتم تمرير طول القائمة ذاتها ثم طباعة عناصرها على التوالي، إذ أن `LevelItem` هو ويدجت قمنا بإنشاؤه بشكل منفصل ويتم تمريره للدالة `onTap` التي تأخذ معرف المستوى (ID) كمعامل.

يتم استدعاء هذه الدالة لتغيير المستوى الحالي عند الضغط على عنصر المستوى.

يقوم بتحويل بيانات JSON الخاصة بالمستوى الحالي إلى كائن `LevelsModel`.

انظر الكود في الشكل(4.22) والذي يوضح كيف تم تمرير رقم المستوى إلى الكلاس السابق في الشكل(4.21)

```
class LevelsItem extends StatelessWidget {
    final LevelsModel levelsModel;
    final Function(String) onTap;

    const LevelsItem({Key? key, required this.levelsModel, required this.onTap})
        : super(key: key);

    @override
    Widget build(BuildContext context) {
        final controller = Get.find<BooksLibraryControllerImp>();
        return Obx(() => Container(
            padding: EdgeInsets.symmetric(horizontal: 10, vertical: 5),
            decoration: BoxDecoration(
                color: controller.levelid.value == levelsModel.levelsId
                    ? AppColor.white
                    : AppColor.secondColor,
                border: Border.all(
                    color: controller.levelid.value == levelsModel.levelsId
                        ? AppColor.secondColor
                        : Colors.transparent,
                ),
                borderRadius: BorderRadius.circular(20),
            ),
            margin: EdgeInsets.only(top: 10, right: 10),
            child: InkWell(
                onTap: () => onTap(levelsModel.levelsId!),
                child: Text(
                    "${levelsModel.levelsNameAr}",
                    style: TextStyle(
                        fontSize: 12,
                        color: controller.levelid.value == levelsModel.levelsId
                            ? AppColor.secondColor
                            : AppColor.white,
                        fontWeight: FontWeight.bold,
                        fontFamily: "Cairo",
                    ),
                ),
            ),
        )));
    }
}
```

شكل 4.22: تابع كود المستويات

4.4 الخلاصة

عرضنا في هذا الفصل بعض الأجزاء من النظام والتطبيق وشرحناها بشكل مفصل وحاولنا قدر الإمكان أن تكون الأكواد المقدمة في هذا الفصل متنوعة ومن أجزاء مختلفة في المشروع.

الفصل الخامس

5 الخاتمة

قمنا في مشروعنا ببناء بوابة الطالب لمركز اللغات والترجمة، حيث بدأناه من مركز اللغات والترجمة في جامعة إقليم سباء، واتبعنا خلال فترة العمل عليه منهجية prototype حيث قمنا في بداية الأمر بعمل تصاميم عرضناها على مدير المركز الدكتور إبراهيم الحوري وقام بالتعديل عليها وإضافة اللازم وحذف ما هو غير ضروري، عقدنا العديد من اللقاءات مع إدارة المركز والتي كنا نخرج منها بأفكار ومتطلبات وتحسينات جديدة، ثم انتقلنا إلى مرحلة التنفيذ وبنينا النظام والتطبيق بالكامل حسب النموذج الذي قمنا بتقادمه سابقاً لإدارة المركز فكان العمل على المشروع منظماً بشكل جيد، استخدمنا في تنفيذ التطبيق إطار عمل Flutter وقمنا ببرمجة البالك اند عن طريق PHP ، كما استخدمنا React js لواجهات النظام، حرصنا على اختيار تقنيات حديثة ذات أداء وكفاءة عالية مما يحسن من أداء المشروع ويحقق الأهداف المنشودة.

5.1 تقييم مدى تحقق الأهداف المحددة للمشروع

قمنا في بداية العمل على المشروع بتحديد ثلاثة أهداف لتحقيقها، وهي:

الهدف الأول: تسهيل وصول الطالب إلى بياناته في الدورة: قمنا بتحقيقه بالفعل من خلال تصميم card يعرض بيانات الطالب والدورة المسجل فيها.

الهدف الثاني: تسهيل الوصول إلى النتائج، مع تقييد الوصول لنتيجة الدورة التي لم يقم الطالب بتصفيتها رسومها بعد: وقد تم تحقيق الهدف الثاني من خلال برمجة screen لعرض نتائج الطالب في الدورات التي أنهى دراستها واختبارها.

الهدف الثالث: مواكبة التطور التقني والتحول الرقمي في جعل كافة التعاملات بين الإدارة والعميل تتم عبر الهاتف: حيث قمنا بإضافة العديد من الميزات التي تتيح للطالب الوصول لمختلف خدمات المركز عبر شاشة جواله وذلك من خلال توفير شاشة للأخبار وربط التطبيق بنظام إشعارات فايبربيس بالإضافة لإتاحة الفرصة للطالب للاطلاع على نتائجه وتصفّح مكتبة إلكترونية مقسمة حسب المستوى من مبتدئ إلى متقدم، بالإضافة إلى خدمة "اتصل بنا" والتي تتم عبر البريد الإلكتروني الخاص بالمركز.

5.2 اقتراحات مستقبلية

في نهاية هذا المشروع ما نعتبره اليوم نهاية قد يكون بداية جديدة لمن سيأتي بعدها فما يبدو مكملاً اليوم قد يكون مجرد بداية ومنصة لانطلاق أفكار جديدة وتجدid مستمر، وقد سبق وعرضنا الميزات التي عملنا عليها

ضمن المشروع، والتي يمكن الزيادة عليها وتطويرها مستقبلاً كإضافة نظام حضور وغياب خاص للطلاب بحيث يتم إنذارهم بمراجعة المركز بعد عدد معين من الغياب تجنبًا للفصل من الدورة، بالإضافة إلى تزويد المكتبة الإلكترونية بمكتبة صوتيات، كما يمكن توسيع التطبيق أكثر ليشمل لقاءات مباشرة وتعليم online للطلبة خارج حدود مأرب أو اليمن

وفي النهاية نحمد الله على فضله وكرمه أن أتم علينا هذا المشروع الذي نُعده خلاصة جهد وتعب لأربع أعوام مضت، وقد كانت أعواماً لا تخلو من التحديات والصعوبات، لكنها مرت بحمد الله بكل قلقها وكثيرها وتعباها، وكما كانت أيضًا مليئة بالتعلم والنمو والنجاح. نسأل الله أن يكون هذا المشروع بذرة خير لنا ولمن يأتي بعدها، وأن يسهم في تقديم الفائدة للمجتمع ويساعد في تحسين خدمات المعاهد التعليمية الخاصة باللغات وغيرها – مستقبلاً . ونوجه بالشكر الجزيل لكل من ساهم ودعمنا في مسيرتنا التعليمية، من دكاترة وأهل ورفاق، على دعمهم وتشجيعهم الدائم.

نأسال الله أن يوفقا في المستقبل لتحقيق المزيد من النجاحات والمساهمة في تطوير التعليم الرقمي في وطننا الحبيب، وأن يكون هذا المشروع نواة لابتكارات جديدة تخدم العملية التعليمية وتساهم في بناء جيل متعلم ومثقف قادر على مواكبة التطورات التقنية الحديثة.

والله ولي التوفيق.

6 المراجع

1. أبو أمنة. (2009). واقع إدارة الموارد البشرية إلكترونياً في الجامعات الفلسطينية النظامية - قطاع غزة، رسالة ماجستير في إدارة الأعمال غير منشورة، الجامعة الإسلامية، كلية الدراسات العليا، غزة، فلسطين.
 2. نجم. (2009). الإدارة والمعرفة الإلكترونية. دار البازوري العلمية للنشر والتوزيع، عمان، الأردن.
 3. جمال علي الدهشان. (2013). استخدام الهاتف المحمول في التعليم بين التأييد والرفض،. ورقة عمل مقدمة إلى الندوة العلمية الثانية "نظم التعليم العالي في عصر التنافسية"، بكلية التربية، جامعة كفر الشيخ، مصر.
- /ASA: <https://www.cu.edu.eg> .ASA .4 تم الاسترداد من موقع (2024 ,7 15)
4. UAEU .5 .تم الاسترداد من تطبيق UAEU الإمارati: (2024 ,7 15) <https://www.uaeu.ac.ae/ar>
 6. صادق العتمي. (Dec, 2023 12). GetX .: حزمة قوية ل Flutter . تم الاسترداد من medium: <https://medium.com/@sadiqalotmy00/getx%D8%AD%D8%B2%D9%85%D8%A9-%D9%82%D9%88%D9%8A%D8%A9-%D9%84-flutter-4a2228416acb>
 7. كندة شرجي. (30 مايو, 2024). النماذج الأولية prototypes: كيف تبنيها وما أهميتها . تم الاسترداد من أكاديمية حسوب: <https://academy.hsoub.com/design/general/%D8%A7%D9%84%D9%86%D9%85%D8%A7%D8%AC%D8%A7%D9%84%D8%A3%D9%88%D9%84%D9%8A%D8%A9-prototypes>

%D9%83%D9%8A%D9%81-%D8%AA%D8%A8%D9%86%D9%8A%D9%87%D8%A7-%D9%88%D9%85%D8%A7-%D8%A3%D9%87%D9%85%D9%8A%D8%AA%D9%87%D8%A7

8. كندة شريحي. (30 مايو, 2024). النماذج الأولية *prototypes*: كيف تبنيها وما أهميتها . تم الاسترداد من أكاديمية حسوب:
<https://academy.hsoub.com/design/general/%D8%A7%D9%84%D9%86%D9%85%D8%A7%D8%B0%D8%AC-%D8%A7%D9%84%D8%A3%D9%88%D9%84%D9%8A%D8%A9-prototypes-%D9%83%D9%8A%D9%81-%D8%AA%D8%A8%D9%86%D9%8A%D9%87%D8%A7-%D9%88%D9%85%D8%A7-%D8%A3%D9%87%D9%85%D9%8A%D8%AA%D9%87%D8%A7>

9. مركز النخبة. (15 , 7 , 2024). تم الاسترداد من تطبيق مركز النخبة:

hl=&<https://play.google.com/store/apps/details?id=com.beetronix.nokhba>

ar