



الجمهورية العربية السورية

المعهد العالي للعلوم التطبيقية والتكنولوجيا

قسم النظم المعلوماتية – السنة الرابعة

شبكات ونظم تشغيل

العام الدراسي 2022/2023

مشروع السنة الرابعة

بناء تطبيق موبايل يساعد الباحثين عن عمل على إيجاد عمل مناسب لهم

ويساعد الشركات على إيجاد موظفين مناسبين

Job Portal Application

إعداد الطالب

علي حسان سعيد

إشراف المهندس

محمود الياس

24/8/2023

الإهداء

أهدي هذا العمل المتواضع إلى أهلي وأصدقائي الذين ساندوني في كل شيء.

كلمة شكر

أَتَقَدِّمُ بِخَالصِ الشُّكْرِ إِلَى كُلِّ مَنْ سَاهَمَ فِي إِنْجَازِ هَذَا الْعَمَلِ. وَأُخَصِّ بِالذِّكْرِ الْمُهَنْدِسَ الْمَشْرِفَ: مُحَمَّدُ الْيَاسَ
لِلْمُسَاعَدَةِ عَلَى إِنْجَازِ هَذَا الْعَمَلِ، وَلِتَوْجِيهَاتِهِ الْقِيَمَةَ وَنَصَائِحِهِ الْمُسْتَمِرَّةِ.

الخلاصة

يهدف هذا المشروع الى بناء تطبيق موبايل يساعد الباحثين عن عمل على إيجاد وظيفة مناسبة لهم، وكذلك يساعد الشركات على إيجاد الموظفين المناسبين للأعمال التي يطلبونها. هذا التطبيق موجه للاستخدام من قبل الباحثين عن عمل وممثلي الشركات، حيث يطلب ممثلو مهارات ومؤهلات يبحثون عنها تناسب الأعمال في شركاتهم، ويستعرض الباحثون عن عمل هذه الأعمال ويتقدمون بطلبات توظيف وطلبات مقابلة للحصول على عمل، بالمقابل يتلقى ممثلو الشركات هذه الطلبات ويستعرضون المتقدمين وملفاتهم الشخصية ومهاراتهم الموجودة على التطبيق من أجل تحديد مواعيد مقابلة للمتقدمين الذين يتم اختيارهم. أوردنا خلال هذا التقرير شرحاً مفصلاً عن الدراسة المرجعية، مروراً بالدراسة التحليلية التي تتضمن مخطط حالات الاستخدام ومتطلبات النظام الوظيفية وغير الوظيفية وصولاً إلى الدراسة التصميمية للنظام وآلية عمله والأدوات المستخدمة في بنائه.

Abstract

This project aims to build a mobile application that helps job seekers to find suitable employment opportunities, as well as assists companies in finding the right employees for their required jobs. This application is intended for use by job seekers and company representatives. Company representatives log into the application and post job listings that require specific skills and qualifications. Job seekers review these listings and submit employment applications and interview requests to get the job. In return, company representatives receive these requests and review the applicants' profiles and skills on the application to schedule interview dates for the selected candidates. In this report, we provide a detailed explanation of the theoretical study, followed by the analytical study, which includes a use case diagram, functional and non-functional system requirements, and the design study of the system, its working mechanism, and the tools used in its development.

الفهرس

1	الخلاصة
11	مقدمة عامة
12	مقدمة والتعريف بالمشروع
12	1. هدف المشروع
12	2. المتطلبات
12	1.2. المتطلبات الوظيفية
16	2.2. المتطلبات غير الوظيفية
16	3. حالات الاستخدام مع أرقام المتطلبات المتعلقة بكل منها
19	4. الخطّة الزمنية للمشروع
21	الدراسة المرجعية
21	1. Glassdoor
22	2. Monster
24	الدراسة التحليلية وتحليل المتطلبات
24	1. مقدمة
24	2. مخطط حالات الاستخدام
26	3. الوصف النصّي لحالات الاستخدام
26	1.3. وصف نصّي لحالة استخدام إنشاء حساب
27	2.3. وصف نصّي لحالة استخدام تسجيل دخول
29	3.3. وصف نصّي لحالة استخدام استعراض ملف شخصي
30	4.3. وصف نصّي لحالة استخدام استعراض الأعمال المطروحة على التطبيق
31	5.3. وصف نصّي لحالة استخدام البحث عن عمل معيّن بحسب الاسم
33	6.3. وصف نصّي لحالة استعراض تفاصيل عمل
34	7.3. وصف نصّي لحالة استخدام تقديم طلب توظيف
35	8.3. وصف نصّي لحالة إنشاء حساب
37	9.3. وصف نصّي لحالة استخدام تسجيل دخول
39	10.3. وصف نصّي لحالة استخدام استعراض جميع الأعمال المطروحة من قبل الشركة
40	11.3. وصف نصّي لحالة استخدام استعراض جميع المتقدمين على عمل معيّن
42	12.3. وصف نصّي لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين على عمل معيّن
43	13.3. وصف نصّي لحالة استخدام حجز مقابلة
44	4. مخططات التتالي للنظام (System Sequence Diagram) SSD
45	1.4. مخطط SSD لحالة استخدام إنشاء حساب
46	2.4. مخطط SSD لحالة استخدام تسجيل دخول
47	3.4. مخطط SSD لحالة استخدام استعراض الملف الشخصي
47	4.4. مخطط SSD لحالة استخدام استعراض الأعمال المطروحة على التطبيق

48	5.4. مخطط SSD لحالة استخدام البحث عن عمل معيّن بحسب الاسم.
49	6.4. مخطط SSD لحالة استخدام استعراض تفاصيل عمل.
49	7.4. مخطط SSD لحالة استخدام تقديم طلب توظيف.
50	8.4. مخطط SSD لحالة استخدام إنشاء حساب.
51	9.4. مخطط SSD لحالة استخدام تسجيل دخول.
52	10.4. مخطط SSD لحالة استخدام استعراض جميع المتقدمين على عمل معيّن.
53	11.4. مخطط SSD لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين على عمل معيّن.
53	12.4. مخطط SSD لحالة استخدام حجز مقابلة.
54	الدراسة التصميمية
54	1. الأدوات والبيئات المستخدمة
54	1.1. بيئة العمل Flutter
55	2.1. بيئة 6 ASP.NET Core
55	3.1. Restful API
56	4.1. SQL Server
57	5.1. Visual Studio Code
57	6.1. Visual Studio 2022
58	7.1. Android Studio
59	8.1. Swagger
59	9.1. Postman
60	10.1. ngrok
61	التنفيذ
61	1. طرف المخدم Back-end
64	1.1. بنية API على المخدم
70	2.1. اختبار ال API
70	2. قاعدة المعطيات
71	1.2. شرح الكيانات
72	3. واجهات المستخدم
72	1.3. بنية front-end
75	2.3. إدارة حالة الصفحة
77	3.3. المكتبات المستخدمة
78	الفصل السادس الاختبارات والواجهات
78	1. مقدمة
78	2. الواجهات
79	1.2. واجهات طالب الوظيفة
90	2.2. واجهات ممثل الشركة
98	3.2. الإشعارات في التطبيق

100 الخاتمة والأفاق المستقبلية
100 1. مقدّمة
100 2. المشاكل والتحدّيات
101 3. الأفاق المستقبلية
101 4. الخاتمة
102 المراجع

قائمة الأشكال

24	الشكل التوضيحي 1: كتلة حالات الاستخدام الخاصة بطالب الوظيفة.
25	الشكل التوضيحي 2: كتلة حالات الاستخدام الخاصة بممثل الشركة.
25	الشكل التوضيحي 3: كتلة حالات الاستخدام الخاصة بممثل شركة وطالب وظيفة معاً.
45	الشكل التوضيحي 4: مخطط SSD لحالة استخدام إنشاء حساب.
46	الشكل التوضيحي 5: مخطط SSD لحالة استخدام تسجيل دخول.
47	الشكل التوضيحي 6: مخطط SSD لحالة استخدام استعراض الملف الشخصي.
47	الشكل التوضيحي 7: مخطط SSD لحالة استخدام استعراض الأعمال المعروضة على التطبيق.
48	الشكل التوضيحي 8: مخطط SSD لحالة استخدام البحث عن عمل معيّن بحسب الاسم.
49	الشكل التوضيحي 9: مخطط SSD لحالة استخدام استعراض تفاصيل عمل.
49	الشكل التوضيحي 10: مخطط SSD لحالة استخدام تقديم طلب توظيف.
50	الشكل التوضيحي 11: مخطط SSD لحالة استخدام إنشاء حساب كممثل شركة.
51	الشكل التوضيحي 12: مخطط SSD لحالة استخدام تسجيل دخول كممثل شركة.
52	الشكل التوضيحي 13: مخطط SSD لحالة استخدام استعراض جميع المتقدمين على عمل معيّن.
53	الشكل التوضيحي 14: مخطط SSD لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين.
53	الشكل التوضيحي 15: مخطط SSD لحالة استخدام حجز مقابلة.
61	الشكل التوضيحي 16: Auto Mapper.
62	الشكل التوضيحي 17: المصادقة باستخدام token.
63	الشكل التوضيحي 18: الجداول التي تقدّمها الخزمة identity.
64	الشكل التوضيحي 19: بنية المخدم back-end.
65	الشكل التوضيحي 20: بنية الطبقة Models.
66	الشكل التوضيحي 21: بنية الطبقة IRepository.
67	الشكل التوضيحي 22: بنية الطبقة Repository.
67	الشكل التوضيحي 23: بنية الواجهة العمومية IRepository.
68	الشكل التوضيحي 24: بنية الطبقة UnitOfWork.
68	الشكل التوضيحي 25: بنية الواجهة IUnitOfWork.
69	الشكل التوضيحي 26: بنية الطبقة Controllers.
71	الشكل التوضيحي 27: تصميم قاعدة المعطيات.
72	الشكل التوضيحي 28: بنية front-end.
72	الشكل التوضيحي 29: بنية الطبقة data.
73	الشكل التوضيحي 30: بنية الطبقة logic.
73	الشكل التوضيحي 31: توابع الصف fetch.
74	الشكل التوضيحي 32: توابع الصف fromJson.
74	الشكل التوضيحي 33: بنية الطبقة Services.
75	الشكل التوضيحي 34: أهم ال widgets التي تم إنشاؤها.
79	الشكل التوضيحي 35: الواجهة الرئيسية في التطبيق.
80	الشكل التوضيحي 36: واجهة تسجيل دخول طالب وظيفة.
81	الشكل التوضيحي 37: تحديد صورة الملف الشخصي أثناء إنشاء الحساب.

81	الشكل التوضيحي 38: واجهة إنشاء حساب طالب وظيفة.
82	الشكل التوضيحي 39: تحديد الاختصاص الذي يعمل ضمنه طالب الوظيفة.
82	الشكل التوضيحي 40: التحقق من قوة كلمة المرور.
83	الشكل التوضيحي 41: الصفحة الرئيسية لطالب الوظيفة.
83	الشكل التوضيحي 42: الصفحة الرئيسية لطالب الوظيفة.
84	الشكل التوضيحي 43: الملف الشخصي لطالب الوظيفة.
84	الشكل التوضيحي 44: الملف الشخصي لطالب الوظيفة.
86	الشكل التوضيحي 45: فترة الأعمال وفقاً لخاصية معينة.
86	الشكل التوضيحي 46: البحث عن عمل بحسب الاسم.
87	الشكل التوضيحي 47: استعراض تفاصيل عمل من قبل طالب وظيفة.
87	الشكل التوضيحي 48: استعراض تفاصيل عمل من قبل طالب وظيفة.
88	الشكل التوضيحي 49: تقديم طلب توظيف.
88	الشكل التوضيحي 50: تقديم طلب توظيف.
89	الشكل التوضيحي 51: تعديل كلمة مرور حساب طالب وظيفة.
89	الشكل التوضيحي 52: تعديل المهارات التي يمتلكها طالب الوظيفة.
90	الشكل التوضيحي 53: استعراض لمحة عامة عن التطبيق.
91	الشكل التوضيحي 54: استعراض الأعمال المطروحة من قبل الشركة لإدارتها.
92	الشكل التوضيحي 55: استعراض تفاصيل عمل مطروح من قبل الشركة.
93	الشكل التوضيحي 56: استعراض المقابلات المحجوزة لعمل مطروح من قبل الشركة.
94	الشكل التوضيحي 57: استعراض الملف الشخصي لأحد المتقدمين.
94	الشكل التوضيحي 58: استعراض جميع المتقدمين على عمل مطروح من قبل الشركة.
95	الشكل التوضيحي 59: حجز مقابلة لطالب وظيفة.
95	الشكل التوضيحي 60: حجز مقابلة لطالب وظيفة.
96	الشكل التوضيحي 61: إضافة عمل جديد من قبل ممثل شركة.
97	الشكل التوضيحي 62: تحديد نوع الدوام المطلوب للعمل المضاف.
97	الشكل التوضيحي 63: تحديد الاختصاص الذي ينتمي إليه العمل المضاف.
99	الشكل التوضيحي 64: إستلام إشعار بالزمن الفعلي.
99	الشكل التوضيحي 65: إستلام إشعار عند تسجيل الدخول إلى التطبيق.

قائمة الجداول

جدول 1: الخطة الزمنية للمشروع.	20
جدول 2: وصف نصي لحالة استخدام تسجيل دخول كطالب وظيفة.	27
جدول 3: وصف نصي لحالة استخدام تسجيل دخول كطالب وظيفة.	28
جدول 4: وصف نصي لحالة استخدام استعراض الملف الشخصي.	30
جدول 5: وصف نصي لحالة استخدام استعراض جميع الأعمال.	31
جدول 6: وصف نصي لحالة استخدام بحث عن عمل معين بحسب الاسم.	32
جدول 7: وصف نصي لحالة استخدام استعراض تفاصيل عمل.	34
جدول 8: وصف نصي لحالة استخدام إضافة طلب توظيف.	35
جدول 9: وصف نصي لحالة استخدام إنشاء حساب ممثل شركة.	36
جدول 10: وصف نصي لحالة استخدام تسجيل دخول ممثل شركة.	38
جدول 11: وصف نصي لحالة استخدام استعراض جميع الأعمال المطروحة من قبل الشركة.	40
جدول 12: وصف نصي لحالة استخدام استعراض جميع المتقدمين على عمل معين.	41
جدول 13: وصف نصي لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين.	42
جدول 14: وصف نصي لحالة استخدام حجز مقابلة.	44

قائمة المصطلحات والاختصارات

الاختصار	English	عربي
	Authentication	مصادقة
	Authorization	التحويل وإعطاء الصلاحيات والسماحيات
API	Application Programming Interface	واجهة برمجة التطبيقات
APK	Android Package	حزمة اندرويد
DTO	Data Transfer Object	غرض نقل المعطيات
ERD	Entity Relationship Diagram	مخطط علائقي للكيانات
HTTP	Hyper Text Transfer Protocol	بروتوكول نقل النصوص الفائقة
IDE	Integrated Development Environment	بيئة تطوير متكاملة
iOS	iphone Operating System	نظام تشغيل آيفون
IP	Internet Protocol	بروتوكول الإنترنت
JWT	Json Web Token	رمز جيسون المشفر
LLVM	Low-Level Virtual Machine	آلة افتراضية منخفضة المستوى
MVC	Model-View-Controller	نموذج-عرض-متحكم
NDK	Native Development Kit	مجموعة أدوات التطوير الأصلية
OAS	OpenAPI Specification	توصيف واجهة برمجة تطبيقات مفتوحة
OEM	Original Equipment Manufacturer	مُصنّع للأدوات الأصلية
ORM	Object-Relational Mapping	الربط العلائقية للكائنات
RDBMS	Relational Database Management System	نظام إدارة قواعد المعطيات العلائقية
REST	Representational State Transfer	نقل الحالة التمثيلي
SQL	Structured Query Language	لغة الاستعلامات المهيكلية
SSAS	SQL Server Analysis Services	خدمات تحليل مخدّم لغة الاستعلامات المهيكلية
SSD	Sequence System Diagram	مخطط تنالي النظام
SSL	Secure Sockets Layer	طبقة مآخذ التوصيل الآمنة

خدمات تقارير مخدم لغة الاستعلامات المهيكلية	SQL Server Reporting Services	SSRS
واجهة المستخدم	User Interface	UI
معرّف الموارد الموحد	Uniform Resource Locator	URL
آلة افتراضية	Virtual Machine	VM

مقدمة عامة

تعد أتمتة عملية التقديم على الوظائف والأعمال بين الشركات والباحثين عن عمل أمراً ذا أهمية كبيرة في العصر الحديث. فالتكنولوجيا والتطور الرقمي قد غيرا بشكل كبير طريقة التواصل والتفاعل بين الشركات والباحثين عن عمل، وأصبحت الخدمات الإلكترونية والتطبيقات المحمولة توفر وسائل فعّالة ومبتكرة لتسهيل هذه العملية. تتيح أتمتة عملية التقديم على الوظائف للشركات إمكانية نشر الوظائف الشاغرة وتفاصيلها بسهولة وسرعة على منصات إلكترونية وتطبيقات محمولة مختلفة. يمكن للشركات تحديد المهارات والمؤهلات المطلوبة لكل وظيفة وتلقي طلبات التوظيف بسرعة كبيرة، مما يوفر الوقت والجهد في عملية الفرز والتواصل مع المتقدمين المحتملين. من جهة أخرى، يستفيد الباحثون عن عمل من تلك الأدوات والتطبيقات التي توفر لهم فرصة البحث والاطلاع على الوظائف المتاحة بسهولة ويسر. يمكن للباحثين عن عمل تقديم طلباتهم وعرض مهاراتهم وسيّرتهم الذاتية بسهولة، مما يزيد من فرصهم في الحصول على فرص عمل مناسبة. بالإضافة إلى ذلك، يمكن للباحثين عن عمل استخدام تلك الأدوات لتتبع حالة طلباتهم والتواصل مع صاحب العمل مباشرة، مما يسهل عليهم متابعة تطورات العملية التوظيفية. بشكل عام، يمثل توفر أدوات التقديم على الوظائف والأعمال أحد العناصر الرئيسية في تحسين عملية التوظيف وتوفير فرص عمل ملائمة للباحثين عن عمل. تسهم هذه الأدوات في توفير الوقت والجهد وتحسين كفاءة التواصل بين الشركات والباحثين عن عمل، مما يعود بالفائدة على الطرفين ويسهم في تحقيق التوافق الأمثل بين المهارات والمؤهلات المطلوبة واحتياجات سوق العمل الحالية.

الفصل الأول

مقدمة والتعريف بالمشروع

يقدم هذا الفصل الأهداف والمتطلبات والخطة الزمنية للمشروع.

1. هدف المشروع

يهدف هذا المشروع إلى بناء تطبيق موبايل يساعد الباحثين عن عمل على إيجاد وظيفة مناسبة لهم، وكذلك يساعد الشركات على إيجاد الموظفين المناسبين للأعمال التي يطلبونها.

2. المتطلبات

تقسم متطلبات المشروع إلى متطلبات وظيفية وغير وظيفية، نجدها فيما يلي.

1.2. المتطلبات الوظيفية

يستخدم التطبيق وفق دورين أساسيين هما طالب الوظيفة وممثل الشركة:

أولاً: يحق لطالب الوظيفة ما يلي:

1. الدخول إلى النظام كطالب وظيفة:

1.1 إنشاء حساب جديد إذا لم يكن يملك حساب من قبل.

2.1 تسجيل الدخول إلى النظام إذا كان يملك حساب من قبل.

2. إمكانية تعديل معلومات حسابه الشخصي:

1.2 تعديل المعلومات الشخصية (الاسم، العنوان، رقم الهاتف، عنوان البريد الإلكتروني، الخبرات، الشهادات

التي حصل عليها، ...إلخ).

2.2 تعديل كلمة المرور.

3.2 تعديل المهارات الشخصية

1.3.2 حذف مهارة.

2.3.2 إضافة مهارة.

3.3.2 تعديل معلومات مهارة.

3. استعراض ملفه الشخصي.

4. استعراض جميع الأعمال والوظائف المعروضة من قبل جميع الشركات:

1.4 إمكانية البحث عن عمل معين من خلال اسمه.

- 2.4 إمكانية فلترة الأعمال:
- 1.2.4 فلترة الأعمال حسب اختصاص معيّن (تكنولوجيا معلومات، محاسبة، ... إلخ).
- 2.2.4 فلترة الأعمال وفقاً لنوع الدوام المطلوب (دوام كامل، دوام جزئي، دوام عن بعد، ... إلخ).
- 3.4 عرض تفاصيل عمل معيّن.
5. التقديم على عمل معيّن:
- 1.5 إرسال طلب توظيف من أجل عمل معيّن.
- 2.5 إرسال طلب إجراء مقابلة.
- 3.5 إرسال إشعار للشركة المالكة للعمل لإعلامها بذلك.
6. إلغاء طلب تقديم على عمل:
- 1.6 حذف الإشعار المرسل إلى الشركة المالكة للعمل، الخاص بطلب التقديم الذي تم إلغاؤه.
- 2.6 حذف طلب التقديم.
7. استعراض جميع الأعمال التي قام بتقديم طلبات توظيف عليها.
- 1.7 استعراض تفاصيل عمل معيّن.
8. استعراض جميع المقابلات المحجوزة له.
- 1.8 البحث عن مقابلة بحسب اسم الشركة التي حجزت له المقابلة.
9. استعراض الملف الشخصي لشركات الموجودة على التطبيق والتي قامت بنشر أعمال على التطبيق.
10. تلقي الإشعارات:
- 1.10 تلقي إشعار عند حجز موعد مقابلة له، من قبل شركة ما بخصوص عمل معيّن.
- 2.10 تلقي إشعار عند تعديل موعد مقابلة له، من قبل شركة ما بخصوص عمل معيّن.
- 3.10 تلقي إشعار عند إلغاء موعد مقابلة له، من قبل شركة ما بخصوص عمل معيّن.
- 4.10 استعراض الإشعارات المستلمة.
- 5.10 إمكانية البحث ضمن الإشعارات عن إشعار معيّن بحسب اسم المرسل.
- 6.10 استعراض تفاصيل إشعار.
11. استعراض المتقدمين على عمل معيّن:
- 1.11 معرفة عدد المتقدمين على عمل معيّن.
- 2.11 استعراض الملفات الشخصية وحسابات المتقدمين على عمل معيّن لتحقيق الشفافية في التطبيق وتحفيز المتقدمين على تحسين مهاراتهم.
- 3.11 استعراض الملف الشخصي لأحد المتقدمين.
- 4.11 التواصل مع أحد المتقدمين من خلال رقم هاتف.
- 5.11 التواصل مع أحد المتقدمين عبر البريد الإلكتروني.
12. إمكانية التواصل مع شركة معيّنة:
- 1.12 إمكانية التواصل مع الشركة من خلال الاتصال عبر رقم هاتف.

- 2.12 إمكانية التواصل مع الشركة من خلال إرسال إيميل عبر عنوان بريد إلكتروني.
13. تسجيل الخروج من التطبيق.
- ثانياً: يحق لممثل الشركة ما يلي:
14. الدخول إلى النظام كممثل شركة:
- 1.14 إنشاء حساب جديد إذا لم يكن يملك حساب من قبل.
- 2.14 تسجيل الدخول إلى التطبيق إذا كان يملك حساب من قبل.
15. تعديل حساب الشركة:
- 1.15 إمكانية تعديل معلومات الحساب الشخصي للشركة (الاسم، العنوان، رقم الهاتف، عنوان البريد الإلكتروني، أنشطة الشركة، ... إلخ).
- 2.15 تعديل كلمة مرور حساب الشركة.
- 3.15 إدارة أعمال الشركة.
16. استعراض الملف الشخصي للشركة.
17. إضافة عمل جديد على التطبيق:
- 1.17 تحديد المعلومات الخاصة بالعمل (الاختصاص، الراتب، نوع الدوام المطلوب للعمل، ... إلخ).
18. استعراض جميع الأعمال المطروحة على التطبيق من قبل جميع الشركات:
- 1.18 إمكانية البحث عن عمل معين من خلال اسمه.
- 2.18 إمكانية فلتر الأعمال:
- 1.2.18 فلتر الأعمال حسب اختصاص معين (تكنولوجيا معلومات، محاسبة، ... إلخ).
- 2.2.18 فلتر الأعمال وفقاً لنوع الدوام المطلوب (عمل بدوام كامل، عمل بدوام جزئي، ... إلخ).
- 3.18 استعراض جميع المتقدمين على عمل معين:
- 1.3.18 معرفة عدد المتقدمين على عمل معين.
- 2.3.18 استعراض الملفات الشخصية وحسابات المتقدمين على هذا العمل.
- 1.2.3.18 استعراض الملف الشخصي لأحد المتقدمين.
- 1.1.2.3.18 إمكانية حجز مقابلة في حال كان العمل مطروح من قبل الشركة المالكة للحساب.
- 2.1.2.3.18 إمكانية إلغاء مقابلة في حال كان العمل مطروح من قبل الشركة المالكة للحساب ومحجوز له مقابلة مسبقاً.
- 4.18 استعراض الملف الشخصي للشركة الطارحة للعمل.
- 1.4.18 التواصل مع الشركة المالكة للعمل من خلال رقم هاتف.
- 2.4.18 التواصل مع الشركة المالكة للعمل من خلال بريد إلكتروني.

- 5.18 استعراض تفاصيل عمل.
19. استعراض جميع الأعمال المطروحة من قبل الشركة المالكة للحساب:
- 1.19 معرفة عدد المتقدمين على عمل مطروح من قبل الشركة.
- 2.19 استعراض الملفات الشخصية وحسابات المتقدمين على عمل مطروح من قبل الشركة.
- 1.2.19 استعراض الملف الشخصي لأحد المتقدمين.
- 1.1.2.19 إمكانية حجز مقابلة.
- 2.1.2.19 إمكانية إلغاء مقابلة في حال كانت محجوزة مسبقاً.
- 3.19 معرفة عدد المقابلات التي تم حجزها لعمل مطروح من قبل الشركة.
- 4.19 استعراض جميع المقابلات التي تم حجزها للمتقدمين على عمل مطروح من قبل الشركة:
- 1.4.19 إمكانية إلغاء مقابلة تم حجزها مسبقاً.
- 2.4.19 إمكانية تعديل موعد مقابلة.
- 3.4.19 إمكانية استعراض الملف الشخصي للمتقدم الذي تم حجز المقابلة له.
- 5.19 استعراض تفاصيل عمل معين.
20. إمكانية تعديل عمل مطروح من قبل الشركة:
- 1.20 إمكانية تعديل المعلومات الخاصة بالعمل (الاسم، الراتب، نوع الدوام المطلوب، ... إلخ).
- 2.20 تعديل المهارات المطلوبة للعمل:
- 1.2.20 إضافة مهارة جديدة للعمل.
- 2.2.20 حذف مهارة من المهارات المطلوبة للعمل.
- 3.2.20 تعديل معلومات مهارة مطلوبة للعمل.
21. إمكانية حذف عمل معين مطروح من قبل الشركة.
22. إمكانية إخفاء ظهور عمل مطروح من قبل الشركة عن جميع طالبي الوظيفة والشركات الأخرى:
- 1.22 إظهار العمل فقط للشركة المالكة للعمل.
23. تلقي الإشعارات:
- 1.23 تلقي الإشعارات من المتقدمين، عند إرسالهم لطلب التوظيف.
- 2.23 استعراض إشعارات الشركة.
24. التواصل مع متقدم سواء كان العمل مملوك من قبل الشركة أو من الشركات الأخرى:
- 1.24 التواصل مع المتقدم من خلال رقم هاتف.
- 2.24 التواصل مع المتقدم من خلال البريد الإلكتروني.
25. تسجيل الخروج من التطبيق.
26. إرسال إشعارات:

1.26 إرسال إشعار عند حجز مقابلة.

2.26 إرسال إشعار عند تعديل مقابلة.

3.26 إرسال إشعار عند إلغاء مقابلة.

ثالثاً: يحق لطالب الوظيفة وممثل الشركة:

27. استعراض جميع المستخدمين:

1.27. استعراض جميع الشركات الموجودة على التطبيق مع عددهم

1.1.27. استعراض الملف الشخصي لشركة.

2.27. استعراض جميع طالبي الوظيفة الذين يستخدمون التطبيق مع عددهم.

1.2.27. استعراض الملف الشخصي لطالب وظيفة.

28. استعراض لمحة عامة عن التطبيق.

2.2. المتطلبات غير الوظيفية

1. سماحية الاستخدام: لا يسمح للمستخدم إلا بالقيام بالعمليات المسموحة له بناءً على دوره في التطبيق.
2. أن يكون النظام سهل الاستخدام بالنسبة للمستخدمين العاديين (أي الأشخاص غير الخبراء)، وأن يكون التنقل سهل وسلس بين صفحات التطبيق.
3. استخدام ميزة إرسال الإشعارات بالزمن الحقيقي (أي لا يضطر المستخدمون إلى تحديث محتوى الصفحات لتلقي الإشعارات الجديدة).
4. استعمال إطار العمل Flutter لبناء التطبيق.
5. زمن الاستجابة: يجب أن يكون التطبيق ذو استجابة مقبولة عند كل تفاعل مع النظام.
6. الاستيعاب وقابلية التوسع: يجب أن يستوعب النظام 1000 مستخدم على الأقل، مع قابلية التوسع.
7. يجب أن يوفر النظام حماية خصوصية معلومات الزبائن.

3. حالات الاستخدام مع أرقام المتطلبات المتعلقة بكل منها

الفاعلون في النظام: طالب الوظيفة، ممثل الشركة.

نورد فيما يلي حالات الاستخدام مع أرقام المتطلبات التي تتعلق بها:

فيما يتعلق بطالب الوظيفة:

○ تسجيل دخول [2.1].

○ إنشاء حساب [1.1].

- استعراض الملف الشخصي [3].
- استعراض جميع الأعمال [4].
- البحث عن عمل معيّن بحسب الاسم [1.4].
- فلترة الأعمال حسب إختصاص معيّن [1.2.4].
- فلترة الأعمال وفقاً لنوع الدوام المطلوب [2.2.4].
- استعراض جميع الأعمال [4].
- استعراض تفاصيل عمل [3.4 – 1.7 – 1.11].
- تعديل المعلومات الشخصية للحساب [1.2].
- تعديل كلمة المرور [2.2].
- حذف مهارة [1.3.2].
- إضافة مهارة [2.3.2].
- تعديل معلومات مهارة [3.3.2].
- تقديم طلب توظيف [1.5 – 2.5 – 3.5].
- إلغاء طلب توظيف [1.6 – 2.6].
- استعراض جميع الأعمال التي قام بالتقديم عليها [7].
- استعراض جميع المقابلات المحجوزة له [8].
- البحث عن مقابلة بحسب اسم الشركة [1.8].
- تلقي إشعارات [3.10 – 2.10 – 1.10].
- استعراض الإشعارات [4.10].
- البحث عن إشعار بحسب اسم الشركة المرسل [5.10].
- استعراض تفاصيل إشعار [6.10].
- استعراض الملفات الشخصية للمتقدمين على عمل معيّن [2.11].
- استعراض الملف الشخصي لأحد المتقدمين [3.11].
- التواصل مع متقدّم من خلال رقم هاتف [4.11].
- التواصل مع متقدّم من خلال بريد إلكتروني [5.11].
- استعراض الملف الشخصي للشركة المالكة للعمل [9].
- التواصل مع الشركة من خلال رقم هاتف [1.12].
- التواصل مع الشركة من خلال البريد الإلكتروني [2.12].
- تسجيل الخروج [13].

فيما يتعلق بممثل الشركة:

- إنشاء حساب [1.14].
- تسجيل الدخول [2.14].
- تعديل المعلومات الشخصية للحساب [1.15].
- تعديل كلمة المرور [2.15].
- إدارة أعمال الشركة [3.15].
- استعراض الملف الشخصي [16].
- إضافة عمل [1.17].
- البحث عن عمل من بين الأعمال المطروحة على التطبيق [1.18].
- فلترة الأعمال المعروضة على التطبيق بحسب الاختصاص [1.2.18].
- فلترة الأعمال المعروضة على التطبيق بحسب نوع الدوام المطلوب [2.2.18].
- استعراض الملف الشخصي لأحد المتقدمين [1.2.3.18 – 1.2.19].
- التواصل مع المتقدم عبر رقم هاتف [1.24].
- التواصل مع المتقدم عبر البريد الإلكتروني [2.24].
- استعراض الملف الشخصي لشركة أخرى [4.18].
- التواصل مع شركة عبر رقم هاتف [1.4.18].
- التواصل مع شركة عبر البريد الإلكتروني [2.4.18].
- استعراض جميع الأعمال المطروحة من قبل الشركة [19].
- استعراض جميع المتقدمين على عمل [2.3.18 – 2.19].
- حجز مقابلة [1.1.2.19 – 1.1.2.3.18].
- استعراض المقابلات المحجوزة للعمل [4.19].
- إلغاء مقابلة [2.1.2.19 – 2.1.2.3.18 – 1.4.19].
- تعديل مقابلة [2.4.19].
- استعراض الملف الشخصي لمتقدم تم حجز مقابلة له [3.4.19].
- استعراض تفاصيل عمل [5.18 – 5.19 – 1.19 – 3.19 – 1.3.18].
- تعديل معلومات عمل [1.20].
- إضافة مهارة [1.2.20].
- حذف مهارة [2.2.20].
- تعديل معلومات مهارة [3.2.20].
- حذف عمل [21].
- إخفاء ظهور عمل [22 – 1.22].

- تلقي إشعارات [1.23].
- استعراض الإشعارات [2.23].
- تسجيل الخروج [25].
- إرسال إشعار [26].

فيما يتعلق بمستخدم سواء كان طالب وظيفة أو ممثل شركة:

- استعراض جميع مستخدمي التطبيق [27].
- استعراض جميع ممثلي الشركات [1.27].
- استعراض جميع طالبي الوظيفة [2.27].
- استعراض الملف الشخصي لشركة [1.1.27].
- استعراض الملف الشخصي لطالب وظيفة [1.2.27].
- استعراض لمحة عامة عن التطبيق [28].

4. الخطة الزمنية للمشروع

المرحلة	عدد الأيام	يوم البداية	يوم النهاية
تثبيت الأدوات اللازمة للمشروع، بالإضافة إلى الدراسة المرجعية.	3	2023/7/10	2023/7/12
تصميم وإنشاء قاعدة المعطيات على SQL Server	4	2023/7/13	2023/7/16
تصميم وإنشاء ال API الخاصة بالتطبيق	8	2023/7/17	2023/7/24
تعلم بيئة عمل Flutter	4	2023/7/25	2023/7/28
تصميم واجهات التطبيق وربطها مع ال API	10	2023/7/29	2023/8/7

2023/8/13	2023/8/8	6	إضافة بعض ال features للتطبيق
2023/8/15	2023/8/14	2	اختبار التطبيق والتأكد من عمله بشكل جيد، والتأكد من ترابط الصفحات والانتقال السلس فيما بينها
			إعداد وكتابة التقرير

جدول 1: الخطة الزمنية للمشروع.

الفصل الثاني

الدراسة المرجعية

يعرض هذا الفصل أمثلة عن بعض المنصات العالمية التي تقدّم خدمات بوابة العمل

1. Glassdoor

منصة إلكترونية تمثّل بوابة وظائف شهيرة تقدم مجموعة شاملة من الميزات لكل من الباحثين عن عمل وأصحاب العمل. فيما يلي بعض التفاصيل حول الوظائف التي تقدّمها هذه المنصة:

- تقدّم للباحثين عن عمل:
 - البحث عن الوظائف: يسمح Glassdoor للباحثين عن عمل بالبحث عن فرص عمل بناءً على معايير مختلفة مثل الموقع أو المسمى الوظيفي أو الشركة أو الكلمات الرئيسية. يمكن للمستخدمين تحسين نتائج البحث للعثور على قوائم الوظائف الأكثر صلة.
 - مراجعات الشركة: إحدى الميزات البارزة لـ Glassdoor هي مجموعتها الواسعة من المراجعات وتقييمات الشركة. يمكن للباحثين عن عمل قراءة المراجعات والتقييمات المقدمة من الموظفين الحاليين والسابقين، مما يمنحهم نظرة ثاقبة لثقافة الشركة، وبيئة العمل، والإدارة، وغيرها.
 - معلومات الراتب: يوفر Glassdoor معلومات الراتب لمختلف الوظائف. حيث يمكن للباحثين عن عمل الوصول إلى بيانات الرواتب، بما في ذلك متوسط الرواتب ونطاقات الرواتب والمكافآت والمزايا، مما يساعدهم في تقييم عروض العمل.
 - رؤية المقابلة: يقدم Glassdoor رؤى للمقابلات حيث يمكن للباحثين عن عمل العثور على معلومات حول عملية المقابلة والأسئلة المطروحة وخبرات المقابلة التي شاركها الآخرون الذين أجروا مقابلات مع الشركة.
 - ملفات تعريف الشركة: يوفر Glassdoor ملفات تعريف مفصلة للشركات، بما في ذلك معلومات حول تاريخ الشركة وقيادتها والموظفين الرئيسيين. يمكن للباحثين عن عمل الحصول على فهم أفضل لخلفية الشركة قبل التقديم.
- تقدّم لأصحاب العمل:
 - إعلانات الوظائف: يسمح Glassdoor لأصحاب العمل بنشر فرص العمل على منصتهم. يمكن لأصحاب العمل تقديم تفاصيل حول الوظيفة والمتطلبات والمؤهلات المطلوبة.
 - مشاركة المرشح: يمكن لأصحاب العمل التواصل مع الباحثين عن عمل من خلال منصة Glassdoor. وهذا يشمل الرد على استفسارات المرشحين، وجدولة المقابلات، وتقديم تحديثات بشأن عملية التوظيف.

- العلامة التجارية لصاحب العمل: يوفر Glassdoor أدوات لأصحاب العمل لعرض ثقافة الشركة وقيمها ومزاياها لجذب المرشحين المحتملين. يمكن لأصحاب العمل مشاركة الصور ومقاطع الفيديو وتحديثات الشركة لتعزيز جهود العلامة التجارية لأصحاب العمل. [12]

2. Monster

منصة إلكترونية تمثل بوابة وظائف شهيرة تقدم مجموعة شاملة من الميزات لكل من الباحثين عن عمل وأصحاب العمل. فيما يلي بعض التفاصيل حول الوظائف التي تقدمها هذه المنصة:

- تقدم للباحثين عن عمل:
 - البحث عن الوظائف: توفر Monster قاعدة بيانات واسعة لقوائم الوظائف من مختلف الصناعات والمواقع في جميع أنحاء العالم. يمكن للباحثين عن عمل البحث عن وظائف بناءً على معايير مثل المسمى الوظيفي والموقع والصناعة والكلمات الرئيسية. تساعد عوامل تصفية البحث المتقدم في تحسين النتائج للعثور على الفرص الأكثر صلة.
 - استئناف الإنشاء والتحميل: يمكن للباحثين عن عمل إنشاء ملفات تعريف احترافية على Monster وتحميل سيرهم الذاتية. توفر المنصة أدوات لبناء السير الذاتية وتنسيقها، مما يسهل على المرشحين عرض مهاراتهم وخبراتهم.
 - تنبيهات الوظائف: تقدم Monster خدمات التنبيه الوظيفي، حيث يمكن للباحثين عن عمل إعداد تنبيهات مخصصة بناءً على تفضيلاتهم. يتلقون إشعارات حول قوائم الوظائف الجديدة التي تتوافق مع معاييرهم، مما يضمن بقائهم على اطلاع بالفرص ذات الصلة.
 - إدارة التطبيقات: يمكن للباحثين عن عمل التقدم للوظائف مباشرة من خلال منصة Monster. يمكنهم تتبع حالة طلباتهم وإدارة طلبات الوظائف المتعددة وتلقي الإخطارات المتعلقة بتحديثات التطبيق وطلبات المقابلة.
 - الموارد المهنية: توفر Monster مجموعة من الموارد والنصائح المهنية لمساعدة الباحثين عن عمل في البحث عن عمل. يتضمن ذلك نصائح حول كتابة السيرة الذاتية، والتحضير للمقابلة، والتطوير الوظيفي، ورؤى الصناعة لمساعدة المرشحين على تحسين فرصهم في الحصول على الوظائف المطلوبة.
- تقدم لأصحاب العمل:
 - إعلانات الوظائف: يمكن لأصحاب العمل نشر فرص عمل على منصة Monster، والوصول إلى مجموعة كبيرة من المرشحين المحتملين. يمكنهم تقديم أوصاف وظيفية شاملة ومتطلبات وتفاصيل أخرى ذات صلة لجذب المتقدمين المناسبين.

- البحث عن المرشحين: تقدم Monster أدوات بحث قوية وعوامل تصفية تسمح لأصحاب العمل بالبحث عن المرشحين بناءً على معايير محددة. يمكن لأصحاب العمل تصفح السير الذاتية وعرض ملفات تعريف المرشحين ووضع قائمة مختصرة بالمرشحين المحتملين لمزيد من التقييم.
- الوصول إلى قاعدة البيانات: يمكن لأصحاب العمل الوصول إلى قاعدة بيانات السيرة الذاتية الشاملة الخاصة بشركة Monster، حيث يمكنهم البحث عن المرشحين الذين قاموا بتحميل سيرهم الذاتية على المنصة. تتيح هذه الميزة لأصحاب العمل العثور بشكل استباقي على المرشحين المحتملين والتواصل معهم.
- إدارة التطبيق: يمكن لأصحاب العمل إدارة عملية التوظيف بأكملها من خلال منصة Monster. يمكنهم مراجعة الطلبات والتواصل مع المرشحين وجدولة المقابلات وتتبع تقدم كل مرشح.
- العلامة التجارية لأصحاب العمل: تقدم Monster العلامات التجارية لأصحاب العمل لمساعدة الشركات على عرض علامتها التجارية وثقافتها وقيمها. يمكن لأصحاب العمل إنشاء ملفات تعريف الشركة ذات العلامات التجارية، وتبسيط الضوء على شهادات الموظفين، ومشاركة المحتوى ذي الصلة لجذب أفضل المواهب. [13]

3. الوصف النصّي لحالات الاستخدام

سنقوم في هذا القسم بسرد نصّي لحالات الاستخدام الأهمّ بالنظام.

أولاً: طالب الوظيفة:

1.3. وصف نصّي لحالة استخدام إنشاء حساب.

هويّة الحالة

اسم حالة الاستخدام **Title**: إنشاء حساب.

النمط **Type**: أساسيّة موسّعة.

ملخص summary: في هذه الحالة يقوم طالب الوظيفة بإنشاء حساب له على التطبيق كطالب وظيفة، ليتمكن من الحصول على الصلاحيات التي يتمتّع بها طالب الوظيفة.

الفاعلون Actors: طالب الوظيفة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/2. تاريخ التعديل **Date of update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلّبات المتعلقة **Cross reference**: [1.1].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: طالب الوظيفة وصل إلى صفحة التسجيل من خلال التطبيق.

السيناريو الأساسي الناجح **Main success scenario**:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يريد طالب الوظيفة إنشاء حساب له على التطبيق.	
2. يطلب النظام من المستخدم إدخال المعلومات اللازمة لإنشاء الحساب (إيميل، كلمة مرور، معلومات شخصية أخرى، إلخ).	
3. يقوم المستخدم بإدخال المعلومات المطلوبة.	
4. يتحقق النظام من المدخلات (غير فارغة).	

5. يتحقق النظام من عدم وجود إيميل آخر بنفس العنوان.	
6. يتحقق النظام من قوة كلمة المرور.	
7. يقوم النظام بإنشاء الحساب، ويقوم بإظهار رسالة نجاح.	
8. يطلب النظام من المستخدم تسجيل الدخول.	

جدول 2: وصف نصّي لحالة استخدام تسجيل دخول كطالب وظيفة.

الظروف اللاحقة postconditions: يصبح المستخدم معرّفًا في النظام كطالب وظيفة.

مسارات الخطأ:

E1- إدخال إيميل موجود مسبقاً.

يبدأ هذا المسار عند النقطة 5 من المسار الأساسي.

4. يطلب النظام من المستخدم إدخال إيميل آخر.

E2- إدخال كلمة مرور غير صالحة.

يبدأ هذا المسار عند النقطة 6 من المسار الأساسي.

7. يطلب النظام من المستخدم إدخال كلمة مرور صالحة (أقوى).

E3- إدخال قيم حقول فارغة.

يبدأ هذا المسار عند النقطة 3 من المسار الأساسي.

4. يطلب النظام من المستخدم إدخال معلومات صالحة (قيم غير فارغة).

2.3. وصف نصّي لحالة استخدام تسجيل دخول.

هوية الحالة

اسم حالة الاستخدام **Title**: تسجيل دخول.

النمط **Type**: أساسية موسّعة.

ملخص summary: في هذه الحالة يطلب المستخدم تسجيل الدخول إلى النظام كطالب وظيفة، باستخدام إيميل وكلمة مرور سجل بهما مسبقاً.

الفاعلون Actors: طالب الوظيفة (مبادر).

تاريخ الإنشاء Creation date: 2023/8/2 **تاريخ التعديل Date of Update:** -

النسخة version: 1 **الشخص المكلف person in charge:** علي سعيد.

المتطلبات المتعلقة Cross reference: [2.1].

سير الأحداث Flow of events

الظروف المسبقة preconditions: يجب أن يكون المستخدم أنشأ حساب على التطبيق كطالب وظيفة.

Main success scenario: السيناريو الأساسي الناجح

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب طالب الوظيفة تسجيل الدخول إلى النظام كطالب وظيفة.	
2. يطلب النظام من المستخدم (طالب الوظيفة) إدخال الإيميل وكلمة المرور.	
3. يقوم طالب الوظيفة بإدخال المعلومات المطلوبة.	
4. يتحقق النظام من أن الإيميل موجود فعلاً.	
5. يتحقق النظام من كون الإيميل يعود لمستخدم دوره هو طالب وظيفة.	
6. يتحقق النظام من تطابق الإيميل وكلمة المرور.	
7. يظهر النظام رسالة بنجاح عملية تسجيل الدخول.	
8. يسمح النظام لطالب الوظيفة بتسجيل الدخول إلى التطبيق.	

جدول 3: وصف نصي لحالة استخدام تسجيل دخول كطالب وظيفة.

الظروف اللاحقة postconditions: طالب الوظيفة دخل إلى التطبيق، وأصبح يتمتع بكامل الصلاحيات المسموحة له.

E1- إدخال إيميل غير موجود.

يبدأ هذا المسار عند النقطة 4 من المسار الأساسي.

5. يظهر النظام للمستخدم رسالة خطأ بفشل عملية تسجيل الدخول.

6. يطلب النظام من المستخدم إدخال إيميل موجود.

E2- إدخال إيميل موجود على التطبيق ولكن يعود لشركة.

يبدأ هذا المسار عند النقطة 5 من المسار الأساسي.

6. يظهر النظام للمستخدم رسالة خطأ بفشل عملية تسجيل الدخول.

7. يطلب النظام من المستخدم إدخال إيميل آخر.

E3- عدم تطابق كلمة المرور مع الإيميل.

يبدأ هذا المسار عند النقطة 6 من المسار الأساسي.

7. يظهر النظام للمستخدم رسالة خطأ بفشل عملية تسجيل الدخول.

8. يطلب النظام من المستخدم إعادة إدخال كلمة المرور.

3.3. وصف نصي لحالة استخدام استعراض ملف شخصي.

هوية الحالة

اسم حالة الاستخدام **Title**: استعراض الملف الشخصي.

النمط **Type**: أساسية موسعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (طالب الوظيفة) استعراض ملفه الشخصي لرؤية معلوماته الشخصية.

الفاعلون **Actors**: طالب الوظيفة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/4 تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [3].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: أن يكون المستخدم (طالب الوظيفة) سجل دخول إلى النظام.

السيناريو الأساسي الناجح **Main success scenario**:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب المستخدم من النظام استعراض ملفه الشخصي	
	2. يقوم النظام بالحصول على الملف الشخصي للمستخدم.
	2. يعرض النظام الملف الشخصي للمستخدم.

جدول 4: وصف نصي لحالة استخدام استعراض الملف الشخصي.

الظروف اللاحقة **postconditions**: عُرض الملف الشخصي للمستخدم.

مسارات الخطأ:

لا يوجد.

4.3. وصف نصي لحالة استخدام استعراض الأعمال المطروحة على التطبيق.

هوية الحالة

اسم حالة الاستخدام **Title**: استعراض الأعمال المطروحة على التطبيق.

النمط **Type**: أساسية موسعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (طالب الوظيفة) من النظام استعراض جميع الأعمال المطروحة على التطبيق من قبل جميع الشركات.

الفاعلون **Actors**: طالب الوظيفة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/3 . تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة [4]:Cross reference

سير الأحداث Flow of events

الظروف المسبقة **preconditions**: أن يكون المستخدم (طالب الوظيفة) سجل دخول إلى النظام.

السيناريو الأساسي الناجح :Main success scenario

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب المستخدم (طالب الوظيفة) من النظام استعراض جميع الأعمال المطروحة على التطبيق.	
	2. يقوم النظام بالحصول على جميع الأعمال المطروحة على التطبيق.
	2. يقوم النظام باستعراض جميع الأعمال المطروحة على التطبيق.

جدول 5: وصف نصي لحالة استخدام استعراض جميع الأعمال.

الظروف اللاحقة **postconditions**: تم عرض جميع الأعمال المطروحة على التطبيق للمستخدم.

مسارات الخطأ:

لا يوجد.

المسارات البديلة:

A1- لا يوجد أعمال مطروحة على التطبيق

يبدأ هذا المسار عند النقطة 3 من المسار الأساسي.

يقوم النظام بإظهار رسالة، يُعلم فيها المستخدم بعدم وجود أعمال على التطبيق.

5.3. وصف نصي لحالة استخدام البحث عن عمل معين بحسب الاسم.

هوية الحالة

اسم حالة الاستخدام **Title**: البحث عن عمل بحسب الاسم.

النمط **Type**: أساسية موسعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (طالب الوظيفة) من النظام البحث عن عمل معين من خلال اسم العمل.

الفاعلون **Actors**: طالب الوظيفة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/3 تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [1.4].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: أن يكون المستخدم (طالب الوظيفة) سجل دخول إلى التطبيق وطلب استعراض جميع الأعمال المطروحة على التطبيق، أو طلب استعراض جميع الأعمال التي قدم عليها.

السيناريو الأساسي الناجح **Main success scenario**:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يقوم المستخدم (طالب الوظيفة) بطلب البحث عن عمل من خلال اسم العمل.	
2. يطلب النظام من المستخدم إدخال اسم العمل.	
3. يبدأ المستخدم بإدخال الحروف الأولى من اسم العمل.	
4. يقوم النظام بعرض الأعمال التي تحتوي أسماؤها الحروف التي أدخلها المستخدم.	
5. يقوم المستخدم بإدخال اسم العمل كاملاً.	
6. يقوم النظام بعرض العمل الموافق للاسم الذي أدخله المستخدم.	

جدول 6: وصف نصي لحالة استخدام بحث عن عمل معين بحسب الاسم.

الظروف اللاحقة **postconditions**: يظهر العمل المطلوب للمستخدم.

مسارات الخطأ:

لا يوجد.

مسارات بديلة:

A1- لا يوجد أعمال أسمائها تحتوي الأحرف التي أدخلها المستخدم.

يبدأ هذا المسار من النقطة 3 من المسار الأساسي.

لا يقوم النظام بإظهار شيء.

A2- لا يوجد عمل موافق للاسم الذي أدخله المستخدم.

يبدأ هذا المسار من النقطة 5 من المسار الأساسي.

يقوم النظام بإظهار رسالة، يُعلم فيها المستخدم بعدم وجود عمل بالاسم المطلوب.

6.3. وصف نصّي لحالة استعراض تفاصيل عمل.

هوية الحالة

اسم حالة الاستخدام **Title**: استعراض تفاصيل عمل.

النمط **Type**: أساسيّة موسّعة.

ملخص Summary: في هذه الحالة يطلب المستخدم (طالب الوظيفة) من النظام استعراض تفاصيل عمل معيّن يختاره.

الفاعلون **Actors**: طالب الوظيفة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/3 تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [3.4 – 1.7].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: أن يكون طالب الوظيفة سجّل دخول إلى النظام، وقام باستعراض جميع الأعمال المعروضة على التطبيق.

السيناريو الأساسي الناجح :Main success scenario

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب المستخدم (طالب الوظيفة) استعراض تفاصيل عمل معين يختاره.	
	2. يقوم النظام بالحصول على جميع تفاصيل العمل.
	3. يقوم النظام بعرض تفاصيل هذا العمل.

جدول 7: وصف نصي لحالة استخدام استعراض تفاصيل عمل.

الظروف اللاحقة postconditions: العمل عُرضت تفاصيله.

مسارات الخطأ:

لا يوجد.

مسارات بديلة:

لا يوجد.

7.3. وصف نصي لحالة استخدام تقديم طلب توظيف.

هوية الحالة

اسم حالة الاستخدام **Title**: تقديم طلب توظيف.

النمط **Type**: أساسية موسعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (طالب الوظيفة) من النظام تقديم طلب توظيف على عمل معين.

الفاعلون **Actors**: طالب الوظيفة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/3 تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [1.5 – 2.5 – 3.5].

سير الأحداث Flow of events

الظروف المسبقة **preconditions**: أن يكون طالب الوظيفة سجّل دخول إلى النظام، وقام باستعراض تفاصيل عمل معيّن، ولا يوجد طلب توظيف مسبق من هذا المستخدم على هذا العمل.

السيناريو الأساسي الناجح: Main success scenario:

الفاعلون	النظام
2. تبدأ هذه الحالة عندما يطلب المستخدم (طالب الوظيفة) تقديم طلب توظيف على عمل معيّن قام باستعراض تفاصيله.	
3. يقوم النظام بإضافة طلب التوظيف.	
4. يقوم النظام بالحصول على معرف الشركة المالكة للعمل.	
5. يقوم النظام بإرسال إشعار إلى الشركة الطارحة للعمل يعلمها بوجود متقدّم جديد على العمل المطروح.	

جدول 8: وصف نصّي لحالة استخدام إضافة طلب توظيف.

الظروف اللاحقة **postconditions**: تمّ إضافة طلب التوظيف، وإرسال إشعار للشركة الطارحة.

مسارات الخطأ:

لا يوجد.

مسارات بديلة:

A1- طالب الوظيفة قام بتقديم طلب توظيف مسبقاً على العمل.

يبدأ هذا المسار من النقطة 1 من المسار الأساسي.

لا يحق للمستخدم (طالب الوظيفة) إطلاق حالة الاستخدام هذه.

ثانياً: ممثل الشركة:

8.3. وصف نصّي لحالة إنشاء حساب

هويّة الحالة

اسم حالة الاستخدام **Title**: إنشاء حساب.

النمط **Type**: أساسية موسعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (ممثل الشركة) من النظام إنشاء حساب له كممثل شركة، ليتمكن من الحصول على الصلاحيات التي يتمتع بها ممثل الشركة.

الفاعلون **Actors**: ممثل الشركة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/3 تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [1.14].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: ممثل الشركة وصل إلى صفحة التسجيل من خلال التطبيق.

السيناريو الأساسي الناجح **Main success scenario**:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يريد ممثل الشركة إنشاء حساب له على التطبيق.	
2. يطلب النظام من المستخدم إدخال المعلومات اللازمة لإنشاء الحساب (إيميل، كلمة مرور، معلومات شخصية أخرى، ... إلخ).	
3. يقوم المستخدم بإدخال المعلومات المطلوبة.	
4. يتحقق النظام من المدخلات (غير فارغة).	
5. يتحقق النظام من عدم وجود إيميل آخر بنفس العنوان.	
6. يتحقق النظام من قوة كلمة المرور.	
7. يقوم النظام بإنشاء الحساب، وإظهار رسالة نجاح.	
8. يطلب النظام من المستخدم تسجيل الدخول.	

جدول 9: وصف نصي لحالة استخدام إنشاء حساب ممثل شركة.

الظروف اللاحقة **postconditions**: يصبح المستخدم معرّفاً في النظام كممثل شركة.

مسارات الخطأ:

E1- إدخال إيميل موجود مسبقاً.

يبدأ هذا المسار عند النقطة 5 من المسار الأساسي.

4. يطلب النظام من المستخدم إدخال إيميل آخر.

E2- إدخال كلمة مرور غير صالحة.

يبدأ هذا المسار عند النقطة 6 من المسار الأساسي.

4. يطلب النظام من المستخدم إدخال كلمة مرور صالحة (أقوى).

E3- إدخال قيم حقول فارغة.

يبدأ هذا المسار عند النقطة 4 من المسار الأساسي.

4. يطلب النظام من المستخدم إدخال معلومات صالحة (قيم غير فارغة).

9.3. وصف نصّي لحالة استخدام تسجيل دخول.

هويّة الحالة

اسم حالة الاستخدام **Title**: تسجيل دخول.

النمط **Type**: أساسيّة موسّعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم تسجيل الدخول إلى النظام كطالب وظيفة، باستخدام إيميل وكلمة مرور سجّل بهما مسبقاً.

الفاعلون **Actors**: ممثل الشركة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/2 تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلّبات المتعلقة **Cross reference**: [2.14].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: يجب أن يكون المستخدم أنشأ حساب على التطبيق كممثل شركة.

السيناريو الأساسي الناجح Main success scenario:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب ممثل الشركة تسجيل الدخول إلى النظام كممثل شركة.	
2. يطلب النظام من المستخدم (ممثل الشركة) إدخال الإيميل وكلمة المرور.	
3. يقوم ممثل الشركة بإدخال المعلومات المطلوبة.	
4. يتحقق النظام من أن الإيميل موجود فعلاً.	
5. يتحقق النظام من كون الإيميل يعود لمستخدم دوره هو ممثل شركة.	
6. يتحقق النظام من تطابق الإيميل وكلمة المرور.	
7. يظهر النظام رسالة بنجاح عملية تسجيل الدخول.	
8. يسمح النظام لممثل الشركة بتسجيل الدخول إلى التطبيق.	

جدول 10: وصف نصي لحالة استخدام تسجيل دخول ممثل شركة.

الظروف اللاحقة **postconditions**: ممثل الشركة دخل إلى التطبيق، وأصبح يتمتع بكامل الصلاحيات المسموحة له.

مسارات الخطأ:

E1- إدخال إيميل غير موجود.

يبدأ هذا المسار عند النقطة 4 من المسار الأساسي.

5. يظهر النظام للمستخدم رسالة خطأ بفشل عملية تسجيل الدخول.

6. يطلب النظام من المستخدم إدخال إيميل موجود.

E2- إدخال إيميل موجود على التطبيق ولكن يعود لطالب وظيفة.

يبدأ هذا المسار عند النقطة 5 من المسار الأساسي.

6. يظهر النظام للمستخدم رسالة خطأ بفشل عملية تسجيل الدخول.

7. يطلب النظام من المستخدم إدخال إيميل آخر.

E3- عدم تطابق كلمة المرور مع الإيميل.

يبدأ هذا المسار عند النقطة 6 من المسار الأساسي.

7. يظهر النظام للمستخدم رسالة خطأ بفشل عملية تسجيل الدخول.

8. يطلب النظام من المستخدم إعادة إدخال كلمة المرور.

10.3. وصف نصي حالة استخدام استعراض جميع الأعمال المطروحة من قبل الشركة.

هوية الحالة

اسم حالة الاستخدام **Title**: استعراض جميع الأعمال المطروحة من قبل الشركة.

النمط **Type**: أساسية موسعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (ممثل الشركة) من النظام استعراض جميع الأعمال المطروحة على التطبيق من قبل الشركة المالكة للحساب.

الفاعلون **Actors**: ممثل الشركة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/5 . تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [19]

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: أن يكون المستخدم (ممثل الشركة) سجل دخول إلى النظام، وطلب إدارة أعمال الشركة.

السيناريو الأساسي الناجح :Main success scenario

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب المستخدم (ممثل الشركة) إدارة أعمال الشركة.	
2. يقوم النظام بالحصول على الأعمال المطروحة من قبل الشركة.	
3. يقوم النظام باستعراض جميع الأعمال المطروحة من قبل الشركة على التطبيق.	

جدول 11: وصف نصي لحالة استخدام استعراض جميع الأعمال المطروحة من قبل الشركة.

الظروف اللاحقة postconditions: جميع الأعمال المطروحة من قبل الشركة عُرضت على التطبيق.

مسارات الخطأ:

لا يوجد.

المسارات البديلة:

A1- لا يوجد أعمال مطروحة من قبل الشركة على التطبيق

يبدأ هذا المسار عند النقطة 2 من المسار الأساسي.

يقوم النظام بإظهار رسالة، يُعلم فيها المستخدم بعدم وجود أعمال للشركة على التطبيق.

11.3. وصف نصي لحالة استخدام استعراض جميع المتقدمين على عمل معين.

هوية الحالة

اسم حالة الاستخدام **Title:** استعراض جميع المتقدمين على عمل معين.

النمط **Type:** أساسية موسعة.

ملخص **summary:** في هذه الحالة يطلب المستخدم (ممثل الشركة) من النظام استعراض جميع المتقدمين على عمل معين.

الفاعلون **Actors:** ممثل الشركة (مبادر).

تاريخ الإنشاء **Creation date:** 2023/8/5 . تاريخ التعديل **Date of Update:** -

النسخة **version:** 1 الشخص المكلف **person in charge:** علي سعيد.

المتطلبات المتعلقة Cross reference: [1.3.18 – 2.3.18 – 1.19 – 2.19].

سير الأحداث Flow of events

الظروف المسبقة preconditions: أن يكون المستخدم (مثل الشركة) سجّل دخول إلى النظام، وطلب إدارة أعمال الشركة وقام باستعراض تفاصيل عمل معيّن، أو قام باستعراض جميع الأعمال المطروحة على التطبيق من قبل جميع الشركات وقام باستعراض تفاصيل عمل معيّن.

السيناريو الأساسي الناجح Main success scenario:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب المستخدم (مثل الشركة) استعراض جميع المتقدمين على عمل معيّن قام بعرض تفاصيله.	
2. يقوم النظام بالحصول على جميع المتقدمين على العمل المطلوب.	
3. يقوم النظام باستعراض جميع المتقدمين على هذا العمل.	

جدول 12: وصف نصّي لحالة استخدام استعراض جميع المتقدمين على عمل معيّن.

الظروف اللاحقة postconditions: عُرض على التطبيق جميع المتقدمين على هذا العمل.

مسارات الخطأ:

لا يوجد.

المسارات البديلة:

A1- لا يوجد متقدمين على هذا العمل

يبدأ هذا المسار عند النقطة 2 من المسار الأساسي.

يقوم النظام بإظهار رسالة، يُعلم فيها المستخدم بعدم وجود متقدمين على هذا العمل.

12.3. وصف نصّي لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين على عمل معيّن.

هوية الحالة

اسم حالة الاستخدام **Title**: استعراض الملف الشخصي لأحد المتقدمين.

النمط **Type**: أساسيّة موسّعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (ممثل الشركة) من النظام استعراض الملف الشخصي لأحد المتقدمين على عمل معيّن، سواء أكان هذا العمل مطروح من قبل الشركة ام لا.

الفاعلون **Actors**: ممثل الشركة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/5 . تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [1.2.19 – 1.2.3.18].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: أن يكون المستخدم (ممثل الشركة) سجّل دخول إلى النظام، وطلب استعراض جميع المتقدمين على عمل معيّن.

السيناريو الأساسي الناجح **Main success scenario**:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب المستخدم (ممثل الشركة) استعراض الملف الشخصي لأحد المتقدمين على عمل معيّن.	
2. يقوم النظام بالحصول على الملف الشخصي للمتقدّم المطلوب.	
3. يقوم النظام باستعراض الملف الشخصي للتطبيق.	

جدول 13: وصف نصّي لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين.

الظروف اللاحقة **postconditions**: عُرض الملف الشخصي للمتقدّم المطلوب على التطبيق.

مسارات الخطأ:

لا يوجد.

13.3. وصف نصّي حالة استخدام حجز مقابلة.

هوية الحالة

اسم حالة الاستخدام **Title**: حجز مقابلة.

النمط **Type**: أساسيّة موسّعة.

ملخص **summary**: في هذه الحالة يطلب المستخدم (ممثل الشركة) من النظام حجز موعد مقابلة لأحد المتقدمين.

الفاعلون **Actors**: ممثل الشركة (مبادر).

تاريخ الإنشاء **Creation date**: 2023/8/5 . تاريخ التعديل **Date of Update**: -

النسخة **version**: 1 الشخص المكلف **person in charge**: علي سعيد.

المتطلبات المتعلقة **Cross reference**: [1.1.2.3.18 - 1.1.2.19].

سير الأحداث **Flow of events**

الظروف المسبقة **preconditions**: أن يكون المستخدم (ممثل الشركة) سجّل دخول إلى النظام، وطلب استعراض جميع المتقدمين على عمل معيّن وقام باستعراض الملف الشخصي لأحد المتقدمين، ولم يتم حجز مقابلة لهذا المتقدم مسبقاً.

السيناريو الأساسي الناجح **Main success scenario**:

الفاعلون	النظام
1. تبدأ هذه الحالة عندما يطلب المستخدم (ممثل الشركة) حجز مقابلة لأحد المستخدمين.	
2. يتحقق النظام من كون الشركة هي المالكة للعمل المطلوب ويحق لها حجز مقابلات لهذا العمل	
3. يطلب النظام من المستخدم (ممثل الشركة) إدخال المعلومات المطلوبة عنوان وتاريخ المقابلة.	
4. يقوم ممثل الشركة بإدخال المعلومات المطلوبة.	
5. يقوم النظام بحجز المقابلة.	

6. يقوم النظام بإرسال اشعار للمتقدّم يُعلمه بحجز مقابلة له من قبل الشركة.	
7. يقوم النظام بإظهار رسالة نجاح حجز المقابلة.	

جدول 14: وصف نصّي لحالة استخدام حجز مقابلة.

الظروف اللاحقة postconditions: حُجزت المقابلة وأُظهرت رسالة بذلك.

مسارات الخطأ:

E1- يقوم ممثل الشركة بإدخال قيم فارغة، عند إدخال معلومات المقابلة.

يبدأ هذا المسار عند النقطة 5 من المسار الأساسي.

يقوم النظام بإظهار رسالة خطأ يُعلم فيها المستخدم (ممثل الشركة) بعدم إدخال قيم فارغة.

مسارات بديلة:

A1- الشركة غير مالكة للعمل المطلوب حجز مقابلة له.

يبدأ هذا المسار عند النقطة 1 من المسار الأساسي.

لا يحق للمستخدم (ممثل الشركة) إطلاق حالة الاستخدام هذه.

A2- الشركة قامت بحجز مقابلة لهذا المتقدّم مسبقاً.

يبدأ هذا المسار عند النقطة 1 من المسار الأساسي.

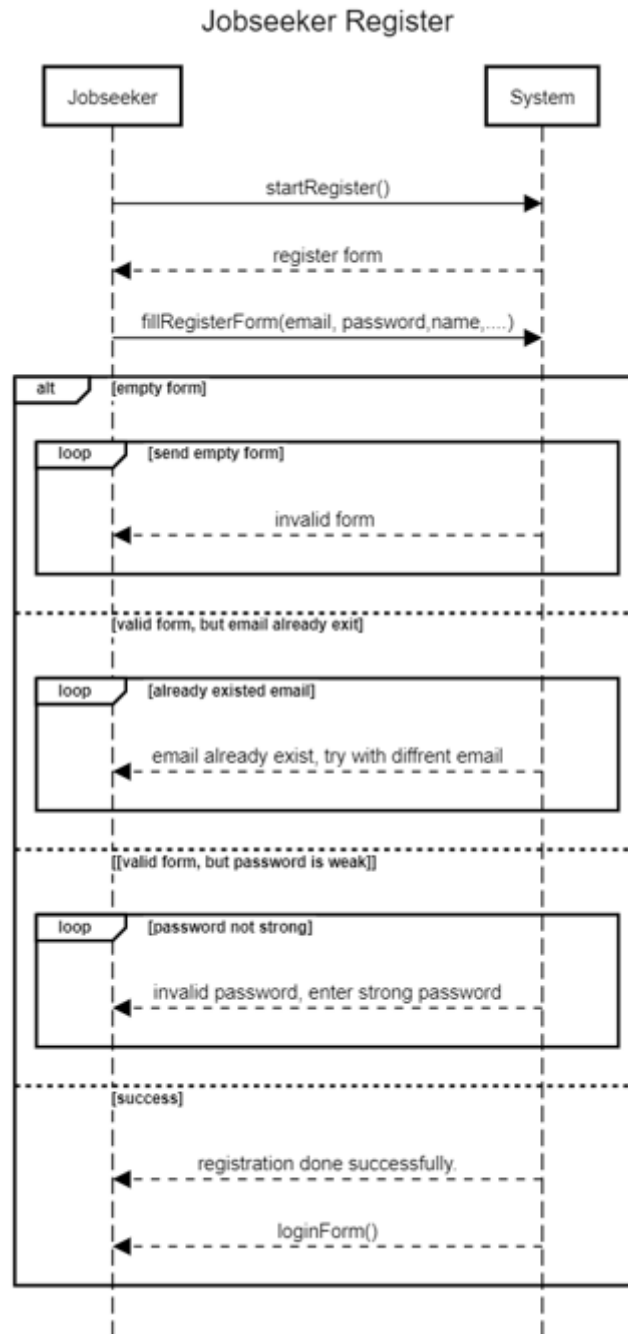
لا يحق للمستخدم (ممثل الشركة) إطلاق حالة الاستخدام هذه.

4. مخططات التتالي للنظام SSD (System Sequence Diagram)

سنقوم في هذا الفصل بعرض مخططات تتالي النظام لحالات الاستخدام التي تمّ عرضها في الوصف النصّي لحالات الاستخدام في الفصل الماضي.

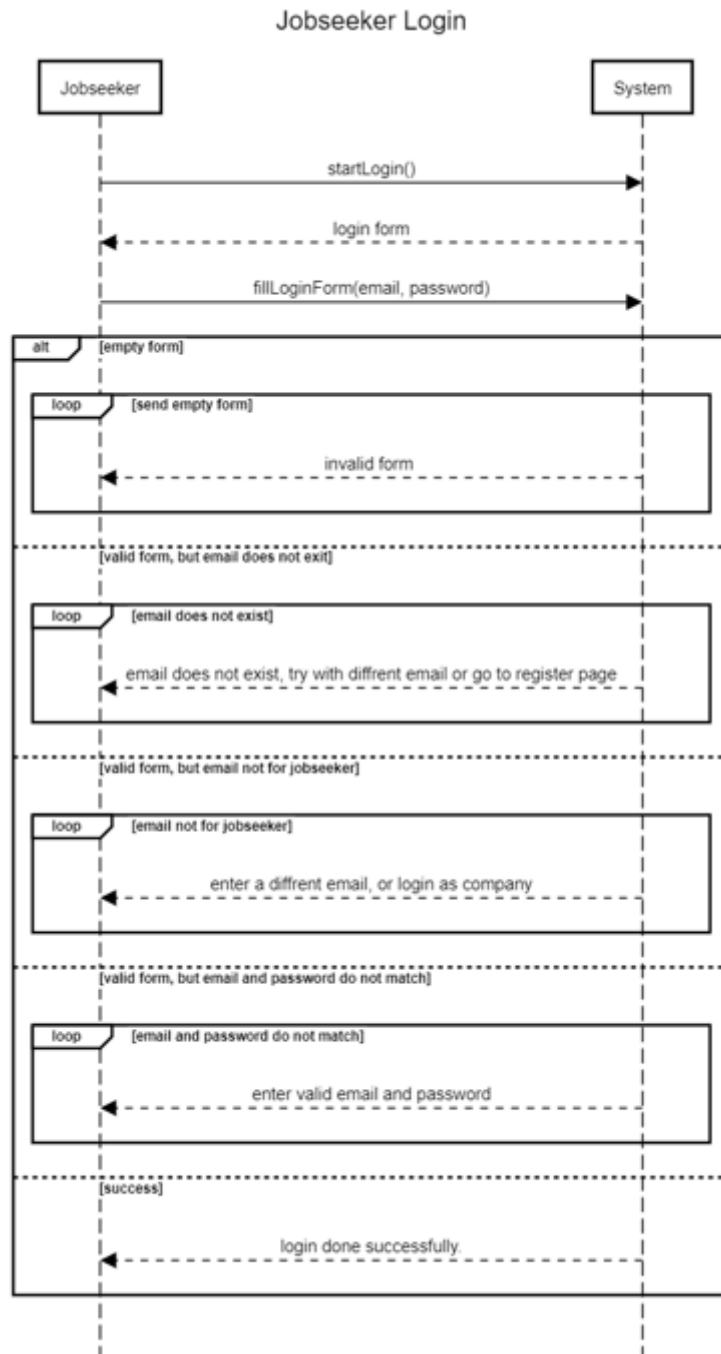
اولاً: طالب الوظيفة.

1.4. مخطط SSD لحالة استخدام إنشاء حساب.



الشكل التوضيحي 4: مخطط SSD لحالة استخدام إنشاء حساب.

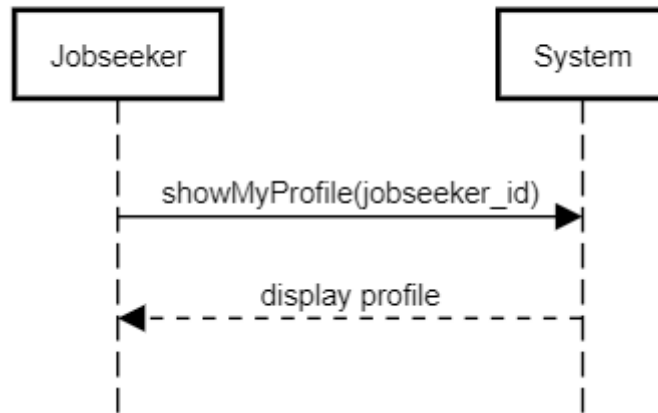
2.4. مخطط SSD لحالة استخدام تسجيل دخول.



الشكل التوضيحي 5: مخطط SSD لحالة استخدام تسجيل دخول.

3.4. مخطط SSD لحالة استخدام استعراض الملف الشخصي.

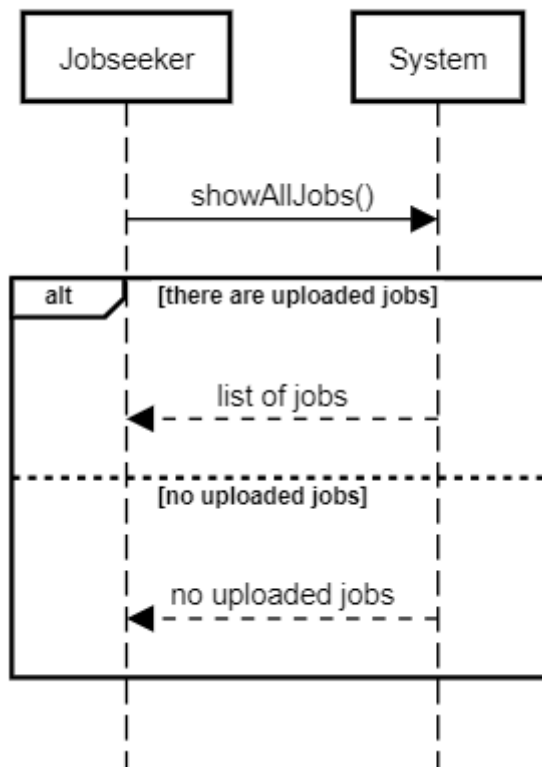
display jobseeker profile



الشكل التوضيحي 6: مخطط SSD لحالة استخدام استعراض الملف الشخصي.

4.4. مخطط SSD لحالة استخدام استعراض الأعمال المطروحة على التطبيق.

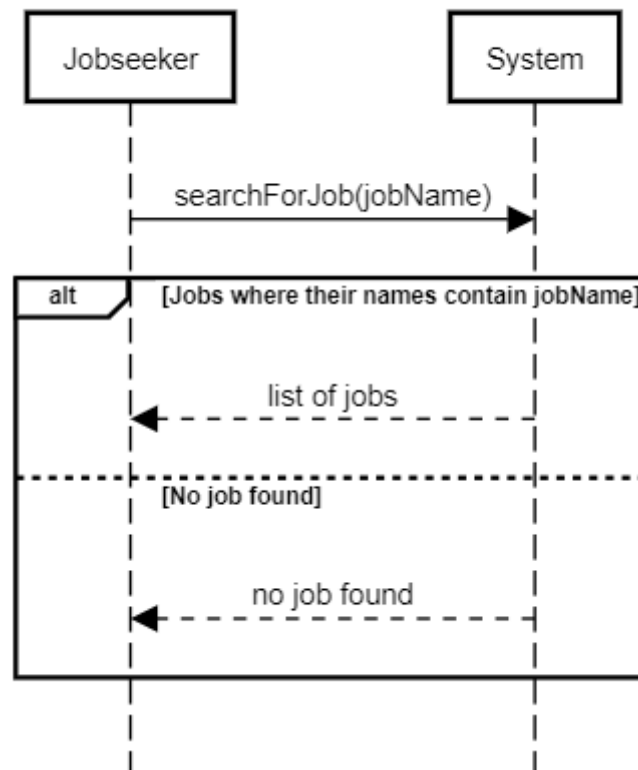
Display All Jobs



الشكل التوضيحي 7: مخطط SSD لحالة استخدام استعراض الأعمال المعروضة على التطبيق.

5.4. مخطط SSD لحالة استخدام البحث عن عمل معين بحسب الاسم.

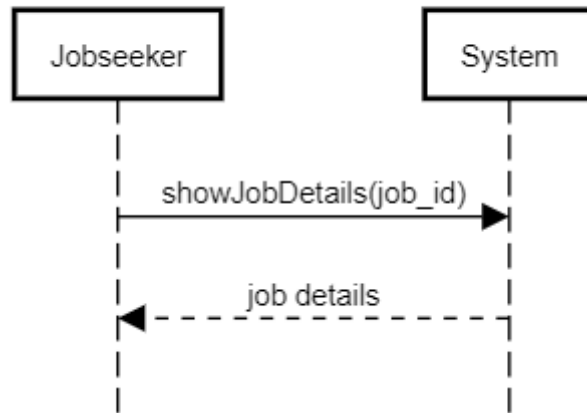
Search For Job



الشكل التوضيحي 8: مخطط SSD لحالة استخدام البحث عن عمل معين بحسب الاسم.

6.4. مخطط SSD لحالة استخدام استعراض تفاصيل عمل.

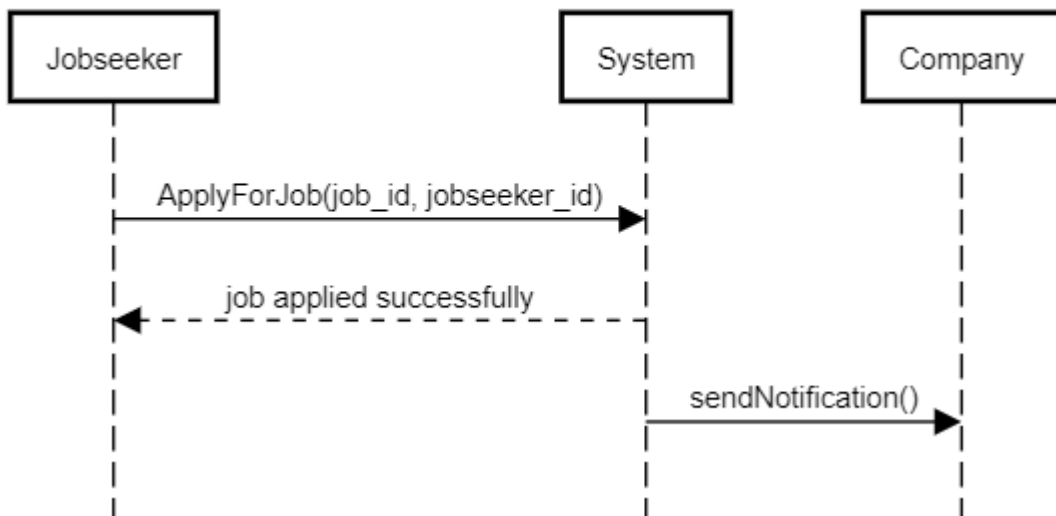
Show Job Details



الشكل التوضيحي 9: مخطط SSD لحالة استخدام استعراض تفاصيل عمل.

7.4. مخطط SSD لحالة استخدام تقديم طلب توظيف.

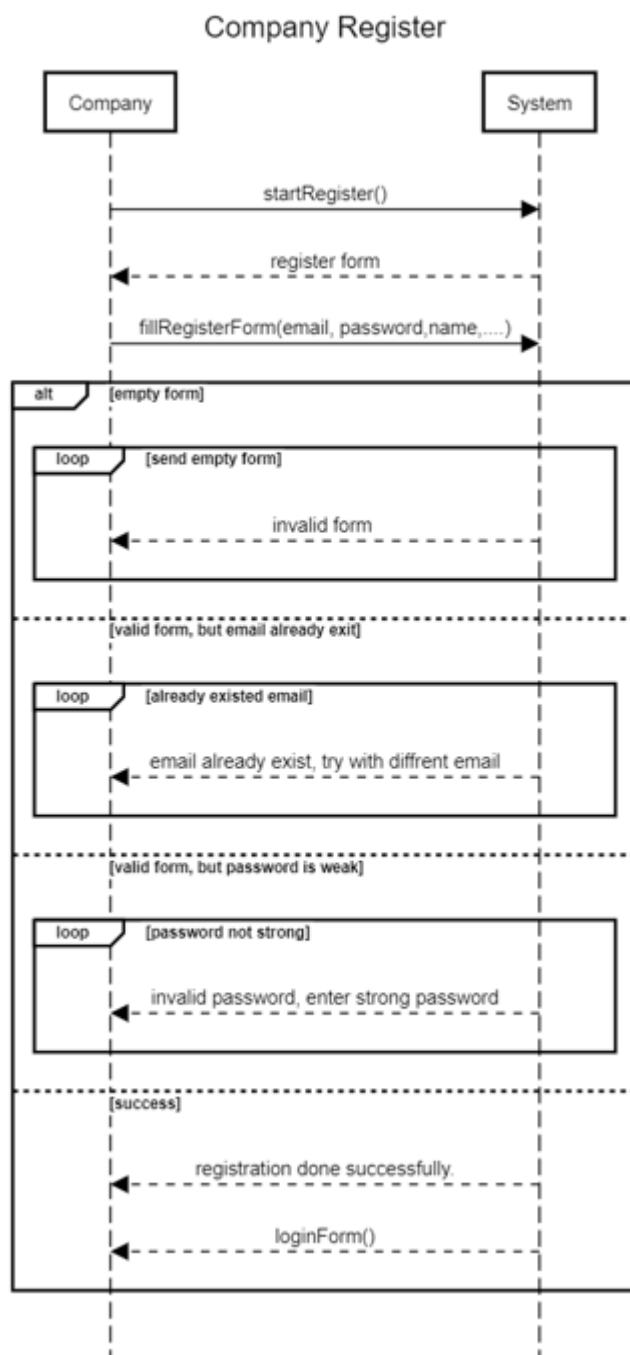
Apply For Job



الشكل التوضيحي 10: مخطط SSD لحالة استخدام تقديم طلب توظيف.

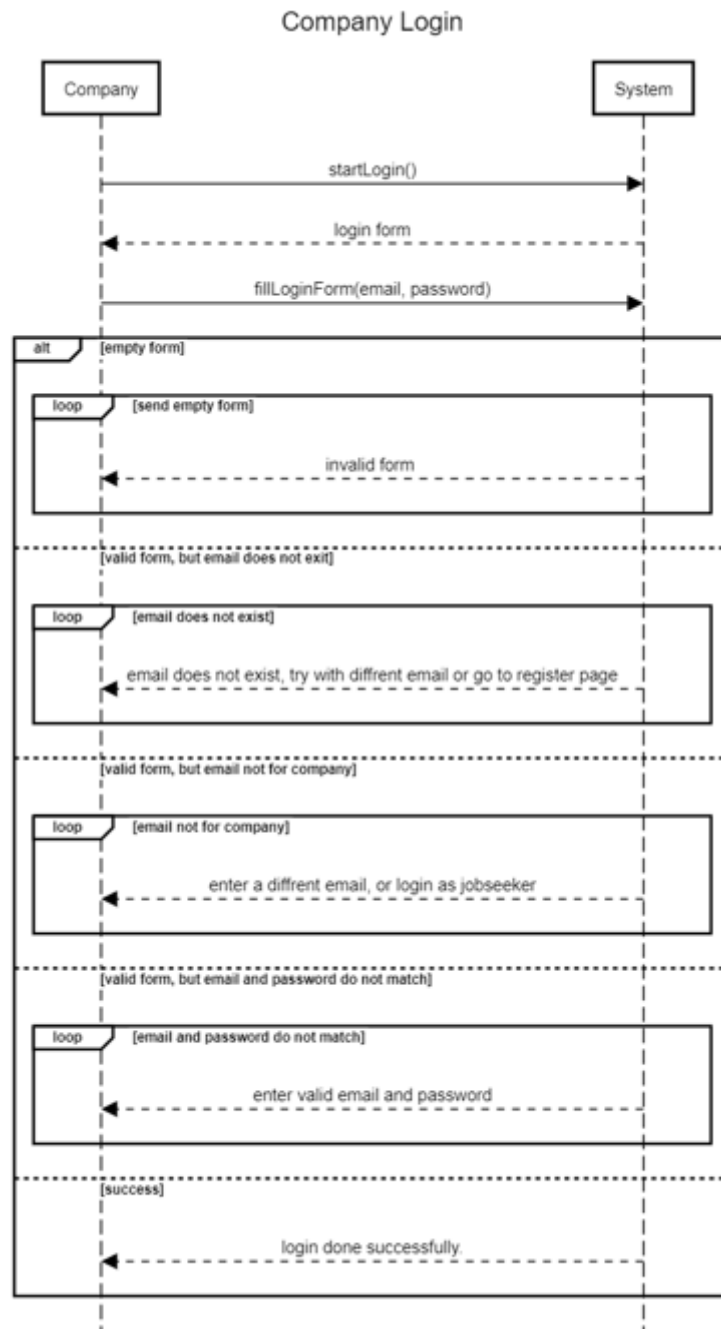
ثانياً: فيما يتعلق بممثل الشركة:

8.4. مخطط SSD لحالة استخدام إنشاء حساب



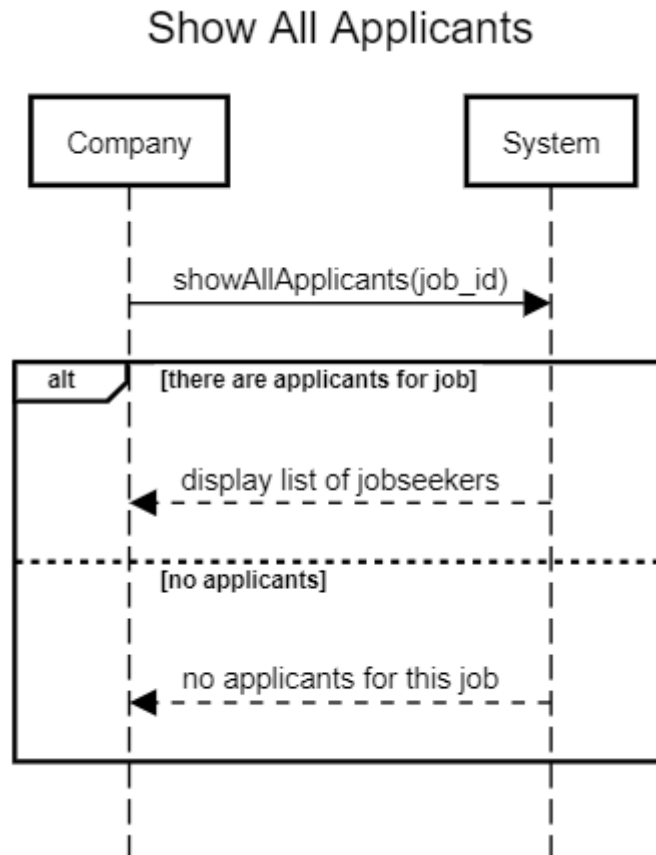
الشكل التوضيحي 11: مخطط SSD لحالة استخدام إنشاء حساب كممثل شركة.

9.4. مخطط SSD لحالة استخدام تسجيل دخول



الشكل التوضيحي 12: مخطط SSD لحالة استخدام تسجيل دخول كممثل شركة.

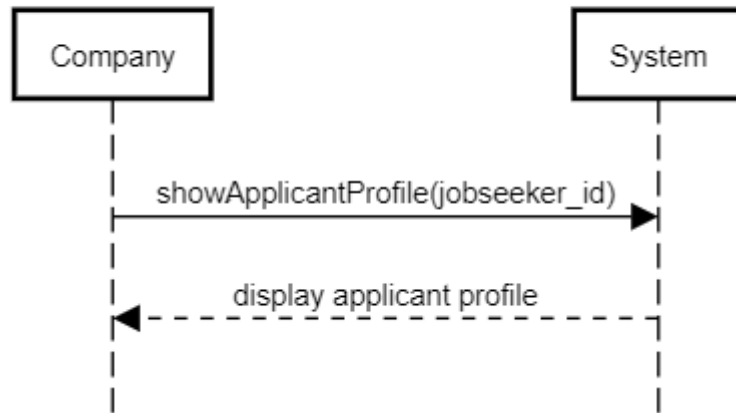
10.4. مخطط SSD لحالة استخدام استعراض جميع المتقدمين على عمل معين



الشكل التوضيحي 13: مخطط SSD لحالة استخدام استعراض جميع المتقدمين على عمل معين.

11.4. مخطط SSD لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين على عمل معين

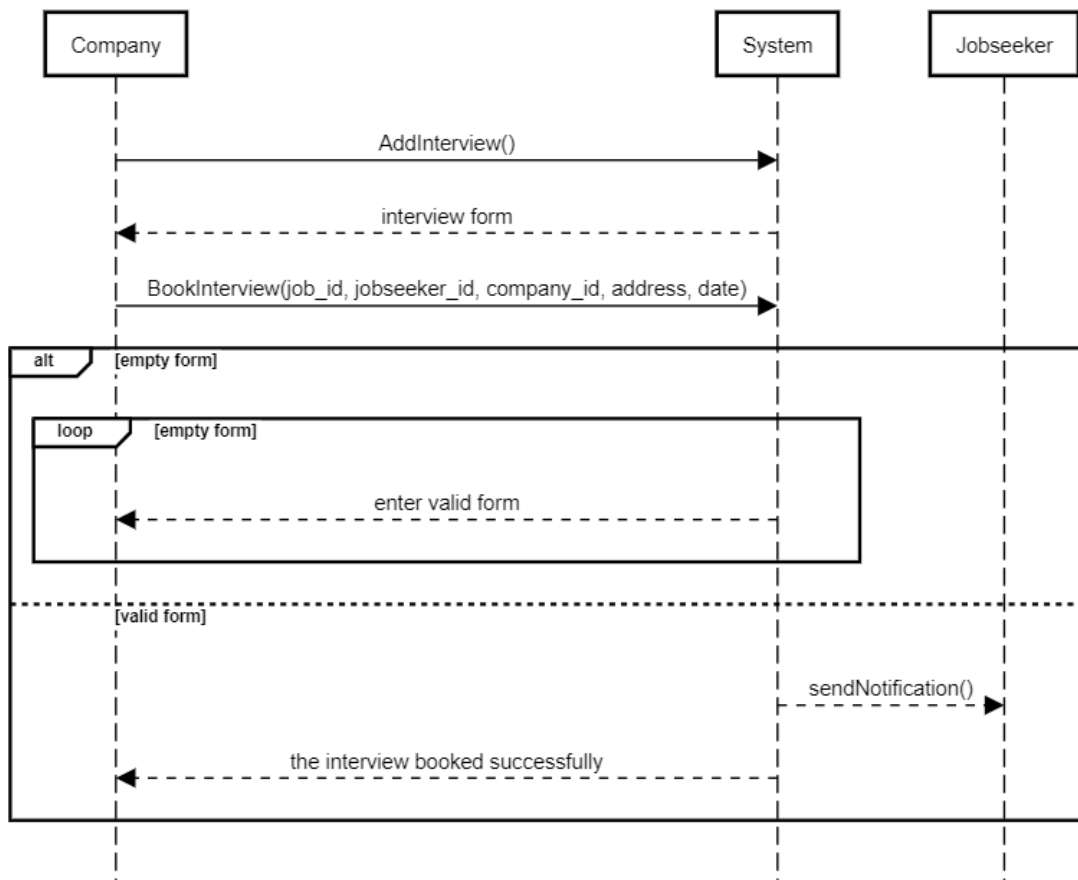
Show Applicant Profile



الشكل التوضيحي 14: مخطط SSD لحالة استخدام استعراض الملف الشخصي لأحد المتقدمين.

12.4. مخطط SSD لحالة استخدام حجز مقابلة

Book Interview



الشكل التوضيحي 15: مخطط SSD لحالة استخدام حجز مقابلة.

الفصل الرابع

الدراسة التصميمية

يقدم هذا الفصل لمحة عن الأدوات والبيئات المستخدمة.

1. الأدوات والبيئات المستخدمة

اعتمدنا في بناء التطبيق على مجموعة من المنصات والأدوات ضمن مفهوم FrontEnd – BackEnd، تتضمن ما يلي:

1.1. بيئة العمل Flutter

فلاتر (Flutter) هي منصة تطوير متعددة الأغراض تم تطويرها بواسطة Google لبناء تطبيقات محمولة عالية الأداء. تستخدم فلاتر لغة البرمجة دارت (Dart) كلغة رئيسية لتطوير التطبيقات. Dart هي لغة برمجة غرضة التوجه تم تطويرها أيضاً بواسطة Google وتستخدم كلغة رئيسية في تطوير تطبيقات فلاتر. تتميز دارت بسهولة التعلم والاستخدام وبأنها بُنيت على قاعدة مستقرة من المفاهيم البرمجية المألوفة.

- فلاتر تمكن المطورين من بناء تطبيقات محمولة عالية الجودة وسهلة التشغيل على مختلف المنصات مثل Android و iOS و web و desktop.
- تتميز فلاتر بتوفير واجهة مستخدم سريعة الاستجابة وجميلة المظهر باستخدام تقنية محرك الرسومات الخاص بها.
- توفر فلاتر أدوات قوية لإنشاء واجهات مستخدم مخصصة وتفاعلية ومتناسقة عبر المنصات المختلفة.
- يتم تطوير فلاتر بلغة دارت ويتم تشغيلها على محرك دارت.

ما يميز Flutter هو نهجها الفريد لبناء واجهات المستخدم. بدلاً من الاعتماد على طرق عرض الويب أو أدوات OEM الأصلية، يستخدم Flutter محرك العرض عالي الأداء الخاص به لإنشاء كل مكون عرض. يتيح ذلك للمطورين إنشاء تطبيقات تتوافق مع أداء التطبيقات الأصلية. محرك Flutter مكتوب بلغة C++ / C وتم تجميعه باستخدام NDK من Android على أجهزة Android و LLVM على أجهزة iOS. تتمثل إحدى الميزات البارزة لمنصة Flutter في دعمها لإعادة التحميل السريع (Hot-Reload) أثناء التطوير. تعمل هذه الميزة على تحسين دورة التطوير بشكل كبير من خلال السماح للمطورين بإجراء تغييرات على التعليمات البرمجية الخاصة بهم ومشاهدة النتائج على الفور دون فقدان الحالة الحالية للتطبيق. يحقق ذلك عن طريق إدخال كود مصدر محدث في جهاز Dart الظاهري قيد التشغيل (VM) دون تغيير البنية الداخلية للتطبيق، مع الحفاظ على جميع الانتقالات والإجراءات. بشكل عام، توفر منصة Flutter للمطورين إطار عمل قوياً لبناء تطبيقات جوال عالية الأداء عبر الأنظمة الأساسية، مما يجعلها خياراً جذاباً للمطورين الذين يتطلعون إلى إنشاء تطبيقات جذابة ومرئية عبر منصات متعددة. [1], [2], [3]

2.1. بيئة ASP.NET Core 6

ASP.NET Core 6 هو إطار عمل مفتوح المصدر قوي ومتعدد الاستخدامات لبناء تطبيقات الويب الحديثة. يعد الإصدار Core 6 من أحدث إصدارات إطار عمل ASP.NET Core من Microsoft ويأتي مع العديد من التحسينات والميزات الجديدة.

ASP.NET Core 6 هو إطار عمل عبر الأنظمة الأساسية يسمح للمطورين بإنشاء تطبيقات وخدمات الويب التي يمكن تشغيلها على أنظمة تشغيل Windows و macOS و Linux. يوفر نموذج برمجة موحدًا لبناء واجهات برمجة تطبيقات الويب وتطبيقات الويب وتطبيقات الوقت الفعلي والخدمات المصغرة.

يوفر ASP.NET Core 6 أداءً محسّنًا وقابلية للتوسع وأمانًا أكبر مقارنة بسابقتها. يستفيد من أحدث تقنيات تطوير الويب وأفضل الممارسات. يدعم إطار العمل العديد من أساليب التطوير، بما في ذلك MVC (Model-View-Controller) لبناء تطبيقات الويب و Web API لبناء خدمات الويب RESTful. يتضمن ASP.NET Core 6 ميزات مثل Razor Pages لبناء صفحات ويب بسيطة، و SignalR للتواصل في الوقت الفعلي، و Blazor لبناء واجهات مستخدم ويب تفاعلية باستخدام C# بدلاً من JavaScript [4].

3.1. Restful API

يعني مصطلح API وجود مجموعة من القواعد والبروتوكولات التي تمكن التطبيقات المختلفة من التواصل والتفاعل مع بعضها البعض. تعمل APIs كجسر للاتصال بين تطبيقين مختلفين، حيث يطلب التطبيق الأول من التطبيق الثاني بعض البيانات أو الخدمات ويحصل عليها من خلال طلب API المناسب. يأتي مفهوم RESTful API كنمط معماري يستخدم في تصميم الخدمات التي يمكن الوصول إليها واستهلاكها عبر منصات وبيئات مختلفة. يستخدم RESTful API بروتوكول HTTP للتواصل ويعتمد على تعيين نقاط النهاية (Endpoints)، تمثل وظائف محددة لعملية تجارية أو تطبيق معين. تتميز RESTful API ببساطتها وقابليتها للتطوير والتكيف. وتتم معالجة طلبات RESTful API باستخدام أفعال http القياسية مثل:

- Get: يستخدم الطلب من هذا النوع للحصول على بيانات.
- Post: يستخدم الطلب من هذا النوع لإرسال بيانات جديدة.
- Put: يستخدم الطلب من هذا النوع لتعديل معلومات موجودة.
- Delete: يستخدم الطلب من هذا النوع لحذف معلومات موجودة.
- Options: يستخدم الطلب من هذا النوع للاستفسار عن معلومة معينة.

تبادل RESTful APIs البيانات باستخدام تنسيق رسائل موحد، حيث بدأت بوصف نصي غير رسمي وتطورت لاحقاً لاستخدام JSON (JavaScript Object Notation) كتتنسيق قياسي. JSON هو تنسيق نصي يمكن معالجته وفهمه بسهولة من قِبل الأجهزة المختلفة عبر الشبكات والمنصات المختلفة. باختصار، يمثل RESTful API نمجاً معمارياً مبسطاً لتصميم خدمات الويب التي يمكن الوصول إليها بسهولة واستهلاكها عبر الإنترنت. يعتمد على استخدام بروتوكول HTTP وتنسيق رسائل موحد مثل JSON لتحقيق التواصل الفعال بين التطبيقات المختلفة. [5]

4.1 SQL Server

SQL Server هو نظام إدارة قواعد المعطيات العلائقية RDBMS المطور من قبل شركة Microsoft. يعتبر SQL Server واحداً من أنظمة إدارة قواعد المعطيات الأكثر شهرة واستخداماً في العالم. يعتمد SQL Server على لغة SQL لإدارة وتخزين البيانات، ويوفر مجموعة واسعة من الأدوات والخدمات لإدارة قواعد البيانات، بدءاً من إنشاء قواعد بيانات جديدة وتعريف الجداول والعلاقات، وصولاً إلى استعلام وتحليل البيانات المخزنة. يوفر SQL Server ميزات قوية لإدارة البيانات وضمان الأمان والاستقرار. تشمل هذه الميزات:

- قدرة مقاومة الأعطال والاسترداد: يتيح SQL Server إمكانية حماية البيانات من الأعطال وضمان استردادها في حالة حدوث خلل في النظام أو فشل الأجهزة.
- أمان البيانات: يوفر SQL Server آليات أمنية قوية لحماية البيانات، بما في ذلك تحقق الهوية والوصول المبني على الأدوار والتشفير والتوثيق.
- توفر النسخ الاحتياطي والاستعادة: يمكن لـ SQL Server إنشاء نسخ احتياطية لقواعد البيانات واستعادتها في حالة فقد البيانات أو حدوث خلل.
- قدرة على التوسع: يمكن لـ SQL Server التعامل مع مجموعة متنوعة من أحجام البيانات وحملات العمل، ويوفر أدوات لتوزيع البيانات وتحقيق الأداء المناسب.
- تحليل البيانات والاستعلام: يوفر SQL Server أدوات وخدمات متقدمة لاستعلام وتحليل البيانات، مثل تقارير SSRS وتحليلات البيانات SSAS.
- قوي وقابل للتطوير: تم تصميم SQL Server للتعامل مع قواعد البيانات الكبيرة والكميات الكبيرة من المعاملات. يوفر ميزات قوية مثل الفهرسة والتقسيم وتحسين الاستعلام التي تعمل على تحسين الأداء وقابلية التوسع.
- ملائم للمطورين: يوفر SQL Server مجموعة شاملة من أدوات التطوير والميزات للمطورين. وهو يدعم لغات برمجة متعددة (مثل T-SQL و dot net و Java) ويوفر واجهات برمجة تطبيقات وأطر عمل قوية لتطوير التطبيقات. يعد SQL Server Management Studio (SSMS) أداة شائعة توفر واجهة رسومية غنية لإدارة قواعد البيانات ومهام التطوير.

- فعال من حيث التكلفة: يوفر SQL Server إصدارات متنوعة لتلبية احتياجات الأعمال المختلفة، بدءًا من التطبيقات صغيرة الحجم إلى عمليات النشر الكبيرة للمؤسسات. يتيح توفر الإصدارات المختلفة للمؤسسات اختيار حل فعال من حيث التكلفة يتوافق مع متطلباتهم.
 - المجتمع والدعم: يحتوي SQL Server على مجتمع كبير ونشط من المستخدمين والمطورين. توفر Microsoft وثائق وبرامج تعليمية وموارد شاملة لدعم المطورين والمسؤولين. بالإضافة إلى ذلك، تقدم Microsoft الدعم الفني والتحديثات المنتظمة لمعالجة الثغرات الأمنية وتحسين وظائف قاعدة البيانات.
- بشكل عام، يوفر SQL Server نظامًا أساسيًا قويًا وآمنًا وغنيًا بالميزات لإدارة البيانات وتحليلها. إن تكامله مع Microsoft ecosystem، وقابلية التوسع، واحتوائه أدوات المطورين الشاملة تجعله خيارًا شائعًا للمؤسسات والأفراد. يستخدم SQL Server على نطاق واسع في مجموعة متنوعة من التطبيقات والصناعات، بما في ذلك الأعمال التجارية والمؤسسات والتطبيقات الويب والتطبيقات الحكومية. يمكن تثبيته على أنظمة التشغيل Windows ويتكامل بشكل وثيق مع تقنيات Microsoft الأخرى مثل dot NET Framework و Azure. [6]

5.1 Visual Sudio Code

محرر نصوص مجاني مفتوح المصدر مطروح من قبل شركة Microsoft صغير الحجم نسبيًا ولكنه قوي يعمل على سطح المكتب ومتاح لأنظمة Windows و macOS و Linux. يحتوي مجموعة كبيرة من الإضافات المفيدة التي يمكن تنزيلها. يأتي مع دعم مدمج لـ JavaScript و TypeScript و Node.js ويحتوي على نظام بيئي غني من الإضافات للغات وأوقات التشغيل (run-times) الأخرى مثل (.NET, Go, PHP, Python, Java, C#, C++) [7].

6.1 Visual Studio 2022

Visual studio 2022 هو بيئة تطوير متكاملة (IDE)، تقدّمها Microsoft لتطوير البرمجيات. يقدّم مجموعة واسعة من الأدوات والميزات لمساعدة المطورين في إنشاء تطبيقات لمختلف المنصات، بما في ذلك الويب، سطح المكتب، الهواتف المحمولة، الحوسبة السحابية وغيرها.

يعتبر Visual Studio 2022 محيط تطوير متكامل يوفر العديد من الميزات التالية:

1. تحسين الأداء: تم تحسين أداء Visual Studio 2022 ليكون أسرع في بدء التشغيل، تحرير التعليمات البرمجية، تصحيح الأخطاء وتقليل استهلاك الذاكرة.

2. تعزيز التعاون: يتضمن الإصدار الجديد ميزات تعاون محسنة تسمح للعديد من المطورين بالعمل على نفس الرماز في وقت واحد. تتضمن هذه الميزات Live Share للتحرير التعاوني في الوقت الفعلي وتصحيح الأخطاء ومراجعة التعليمات البرمجية.

3. واجهة مستخدم حديثة وتحسينات في الإنتاجية: يقدم Visual Studio 2022 واجهة مستخدم محدثة، مما يجعل من السهل التنقل والعمل مع البيئة. كما يشمل تحسينات في الإنتاجية مثل إكمال الرماز المدعوم بالذكاء الاصطناعي، وتحسينات في إمكانية البحث، واقتراحات IntelliCode المحسنة.

4. تطوير متعدد المنصات: يدعم Visual Studio 2022 التطوير متعدد المنصات مع أدوات محسنة لبناء تطبيقات لنظام التشغيل Windows والويب والهواتف المحمولة (بما في ذلك Android و iOS) ومنصات الحوسبة السحابية مثل Azure. توفر البيئة قوالب المشروع وأدوات التصحيح وإمكانيات النشر لمختلف المنصات. [8]

7.1 Android Studio

أندرويد ستوديو (Android Studio) هو بيئة تطوير متكاملة (IDE) مصممة خصيصًا لتطوير تطبيقات الأندرويد. إنها البيئة الرسمية لتطوير تطبيقات الأندرويد وتقدمها شركة غوغل.

بعض الميزات والوظائف الرئيسية لأندرويد ستوديو:

1. تحرير الرماز: يوفر أندرويد ستوديو محرر رماز قوي يتضمن ميزات مثل إكمال الرماز وتمييز الصيغة وتنسيق الرماز وأدوات إعادة التنظيم. يدعم عدة لغات برمجة بما في ذلك Java, Kotlin.

2. محرر التخطيط: يتيح محرر التخطيط للمطورين تصميم واجهة المستخدم (UI) لتطبيقات الأندرويد بشكل بصري. يوفر واجهة سحب وإفلات لترتيب مكونات واجهة المستخدم ويدعم معاينة التخطيطات على أحجام وتوجيهات الشاشة المختلفة.

3. تحليل APK: يتضمن أندرويد ستوديو أداة تحليل APK تسمح للمطورين بفحص محتويات ملفات APK (حزمة أندرويد). تقدم معلومات حول حجم مكونات التطبيق المختلفة وتحدد استخدام الموارد وتساعد في تحسين حجم APK.

4. تصحيح الأخطاء: يوفر أندرويد ستوديو مصححًا قويًا لتصحيح الأخطاء في تطبيقات الأندرويد. يسمح للمطورين بتعيين نقاط التوقف والتنقل في الشفرة وفحص المتغيرات وتحليل سلوك التطبيق أثناء التنفيذ. كما يوفر أدوات لتحسين الأداء لتحديد وتحسين نقاط ضعف الأداء في التطبيق.

5. اختبار المحاكى والجهاز: يتضمن أندرويد ستوديو محاكي أندرويد يسمح للمطورين باختبار تطبيقاتهم على أجهزة أندرويد افتراضية بتكوينات مختلفة. كما يدعم الاتصال بأجهزة أندرويد فعلية للتجربة وتصحيح الأخطاء.

6. إدارة البناء والاعتماديات: يتكامل أندرويد ستوديو مع نظام بناء Gradle، الذي يقوم بإدارة الاعتماديات في المشروع. [9]

8.1 Swagger

هو إطار عمل مفتوح المصدر يساعد المطورين على تصميم وبناء وتوثيق واستهلاك خدمات الويب. يوفر مجموعة من الأدوات والمواصفات التي تبسط عملية إنشاء وصيانة واجهات برمجة التطبيقات. يستخدم Swagger في جوهره تنسيقًا يمكن قراءته آليًا يسمى (OAS) OpenAPI Specification لوصف بنية وسلوك API. تحدد وثيقة OAS نقاط النهاية المتاحة وتنسيقات الطلب / الاستجابة ومعلومات الإدخال ومتطلبات المصادقة والتفاصيل الأساسية الأخرى. الفوائد الرئيسية لاستخدام Swagger هي:

- توثيق واجهة برمجة التطبيقات: ينشئ Swagger وثائق تفاعلية وسهلة الاستخدام لواجهات برمجة التطبيقات الخاصة بنا تلقائيًا. تتضمن هذه الوثائق تفاصيل حول نقاط نهاية API، وتنسيقات الطلب / الاستجابة، والمعلومات المدعومة، بل وتسمح للمستخدمين باختبار استدعاءات API مباشرة من الوثائق.
- تصميم واجهة برمجة التطبيقات: يوفر Swagger نهج التصميم أولاً لبناء واجهات برمجة التطبيقات. يمكنك تحديد هيكل API الخاص بك والمنطق باستخدام Swagger Editor أو أدوات أخرى متوافقة. هذا يساعد في تخطيط وتصوير API قبل كتابة أي رمز، مما يؤدي إلى عملية تطوير أكثر كفاءة. [10]

9.1 Postman

عبارة عن نظام أساسي تعاوني شائع يبسط عملية تصميم واجهات برمجة التطبيقات واختبارها وتوثيقها. يوفر واجهة سهلة الاستخدام ومجموعة من الميزات القوية التي تجعل العمل مع واجهات برمجة التطبيقات أكثر كفاءة وإنتاجية. الميزات والفوائد الرئيسية ل Postman هي:

- اختبار API : يسمح Postman للمطورين بإنشاء وإرسال طلبات HTTP إلى نقاط نهاية API وتلقي الردود. هذا يجعل من السهل اختبار وظائف وسلوك واجهات برمجة التطبيقات أثناء التطوير. يمكن للمطورين أيضًا إعداد الاختبارات والتأكدات الآلية لضمان عمل واجهة برمجة التطبيقات كما هو متوقع.
- تجميع الطلبات: يتيح لنا Postman تنظيم طلبات API في مجموعات، مما يسهل إدارتها ومشاركتها مع أعضاء الفريق. يمكن تنظيم المجموعات بشكل هرمي وتضمين طلبات متعددة بنقاط نهاية ورؤوس headers ومعاملات نصية مختلفة.
- متغيرات البيئة: يسمح لنا Postman بتحديد متغيرات البيئة التي يمكن استخدامها لتكوين الطلبات ديناميكيًا. يكون هذا مفيدًا عند العمل مع بيئات مختلفة مثل التطوير أو التدريج أو الإنتاج. يمكن تبديل المتغيرات بسهولة، مما يسهل اختبار واجهات برمجة التطبيقات في إعدادات مختلفة.
- التوثيق والتعاون: يوفر Postman ميزة تسمى "Postman Documentation" تقوم تلقائيًا بإنشاء وثائق API تفاعلية بناءً على الطلبات والردود في المجموعة التي تم إنشاؤها. يمكن مشاركة هذه الوثائق مع أصحاب المصلحة ويمكن أن تشمل الأوصاف والأمثلة وإرشادات الاستخدام.

- مراقبة API: يوفر Postman ميزة مراقبة API التي تسمح لنا بمجدولة الاختبارات لواجهات برمجة التطبيقات الخاصة بنا على فترات منتظمة. يساعد هذا في ضمان أن تكون واجهات برمجة التطبيقات الخاصة بنا متاحة دائمًا وتعمل كما هو متوقع. [11]

10.1 ngrok

أداة تقنية تستخدم لإنشاء اتصالات آمنة وعامة بين الخوادم المحلية والإنترنت. يعمل Ngrok كخادم وكيل يسمح لنا بإعادة توجيه حركة المرور من خادمنا المحلي إلى عنوان عام يمكن الوصول إليه عبر الإنترنت. يعد Ngrok مفيداً بشكل خاص عندما نحتاج إلى اختبار تطبيقات الويب المحلية أو إجراء اختبار في بيئة تطوير محلية ونرغب في الوصول إليها من أي مكان على الإنترنت. بدلاً من الاعتماد على شبكة محلية أو نشر مباشر على الإنترنت، يمكننا في هذه الحالة استخدام Ngrok لإنشاء رابط يمكن الوصول إليه بشكل عام لعرض تطبيقنا المحلي على أجهزة أخرى أو مشاركته مع الآخرين لجمع ملاحظاتهم وتقييماتهم. يوفر Ngrok أيضاً ميزات أمان قوية، ويؤمن اتصالاتنا عبر بروتوكول HTTPS ويقدم شهادات SSL لتشفير البيانات المرسلة والمستلمة عبر الشبكة. باستخدام Ngrok، يمكننا بسهولة التغلب على الحواجز التقنية وحواجز الشبكة التي قد تعيق الوصول إلى التطبيق المحلي الخاص بنا عبر الإنترنت. حيث إنه أداة قوية ومفيدة للمطورين وفرق التطوير لتسهيل عملية تطوير واختبار تطبيقاتهم. [14]

الفصل الخامس

التنفيذ

يقدم هذا الفصل خلاصة عن آلية إنجاز المشروع.

1. طرف المخدم Back-end

تم بناء Restful API على المخدم باستخدام ASP.NET Core 6، لتقوم بتلقي طلبات الزبون -front end وتقوم بمعالجتها، والإستجابة لها، وتم بناء هذه ال API بالاستفادة من مجموعة من الحزم packages التي تقدمها ASP.NET Core 6 وأهم هذه الحزم والمكتبات:

- AutoMapper.Extensions.Microsoft.DependencyInjection: تستخدم هذه

الحزمة للربط بين غرض وغرض من صف آخر. حيث تقوم تلقائياً بمقابلة قيمة كل واصفة من الغرض الأول إلى الواصفة من الغرض الثاني التي تحمل نفس الاسم، إذا كان الغرضان لهما واصفات بأسماء مختلفة. يمكن أن نقوم بتحديد الواصفة من الغرض الأول التي سننقل قيمتها إلى الواصفة في الغرض الثاني والتي تحمل اسم مختلفاً.

```
// from , to
CreateMap<Person, PersonDto>();
CreateMap<JobSeeker, JobSeekerDto>()
    .ForMember(dest => dest.SpecificationName, opt => opt.MapFrom(src => src.Specification.Name))
    .ForMember(dest => dest.Address, opt => opt.MapFrom(src => src.IdNavigation.Address))
    .ForMember(dest => dest.Name, opt => opt.MapFrom(src => src.IdNavigation.Name))
    .ForMember(dest => dest.Email, opt => opt.MapFrom(src => src.IdNavigation.Email))
    .ForMember(dest => dest.Image, opt => opt.MapFrom(src => src.IdNavigation.Image))
    .ForMember(dest => dest.Phone, opt => opt.MapFrom(src => src.IdNavigation.Phone));

CreateMap<PersonJobSeekerDto, Cv>();
CreateMap<Notification, NotificationDto>();
CreateMap<BookInterviewDto, Interview>();
CreateMap<Interview, InterviewDto>();
CreateMap<PersonDto, Person>().ForMember(src => src.Image, opt => opt.Ignore());
CreateMap<PersonJobSeekerDto, Person>()
    .ForMember(src => src.Image, opt => opt.Ignore());
CreateMap<PersonCompanyDto, Person>()
    .ForMember(src => src.Image, opt => opt.Ignore());

CreateMap<SkillDto, Skill>()
    .ForMember(dest => dest.Name, opt => opt.MapFrom(src => src.Name.ToUpper()));
```

الشكل التوضيحي 16: AutoMapper.

- Microsoft.AspNetCore.Authentication.JwtBearer: هذه الحزمة تسمح بإمكانية

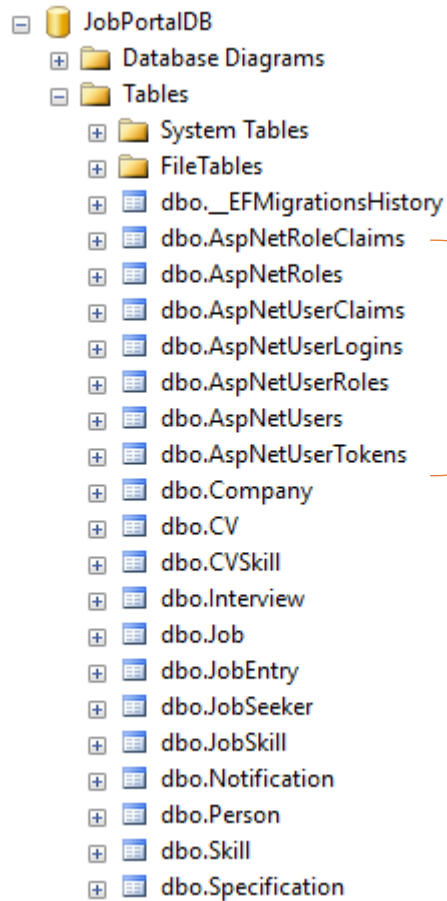
المصادقة باستخدام Jwt حيث عندما يسجل المستخدم الدخول بنجاح، يقوم المخدم بإرسال token مميزة له، وعندما يقوم المستخدم بطلب عمليات أخرى من المخدم تحتاج عملية مصادقة، لا داعي أن يقوم المستخدم

بإجراء تسجيل دخول مرة أخرى، وإنما يكفي أن يرسل ال token التي أرسلها له المخدم ضمن طلبات http المرسلة إلى المخدم، وعندها تتم عملية المصادقة بنجاح، ويقوم المخدم بتنفيذ العمليات المطلوبة.

```
final url = Uri.parse('$myUrl/api/CVs/AddSkill/${identifier.id}');
final response = await http.post(
  url,
  headers: <String, String>{
    'Content-Type': 'application/json; charset=UTF-8',
    HttpHeaders.authorizationHeader: "Bearer ${identifier.token}"
  },
  body: jsonEncode(<String, String>{
    'name': _nameController.text,
    'practicePeriod': _periodController.text,
  }),
);
```

الشكل التوضيحي 17: المصادقة باستخدام token.

- Microsoft.AspNetCore.Identity.EntityFrameworkCore: تقدّم هذه الحزمة مجموعة من الصفوف والواجهات المنجزة لتحقيق عملية ال identity ضمن ASP.NET Core، هذه الصفوف والواجهات تسمح بإمكانية إدارة المستخدمين، المصادقة، التخويل وإعطاء الصلاحيات، وتحديد أدوار المستخدمين. حيث أن هذه الصفوف تتم إضافتها إلى جداول قاعدة المعطيات الخاصة بنا من خلال إجراء عملية Migration.



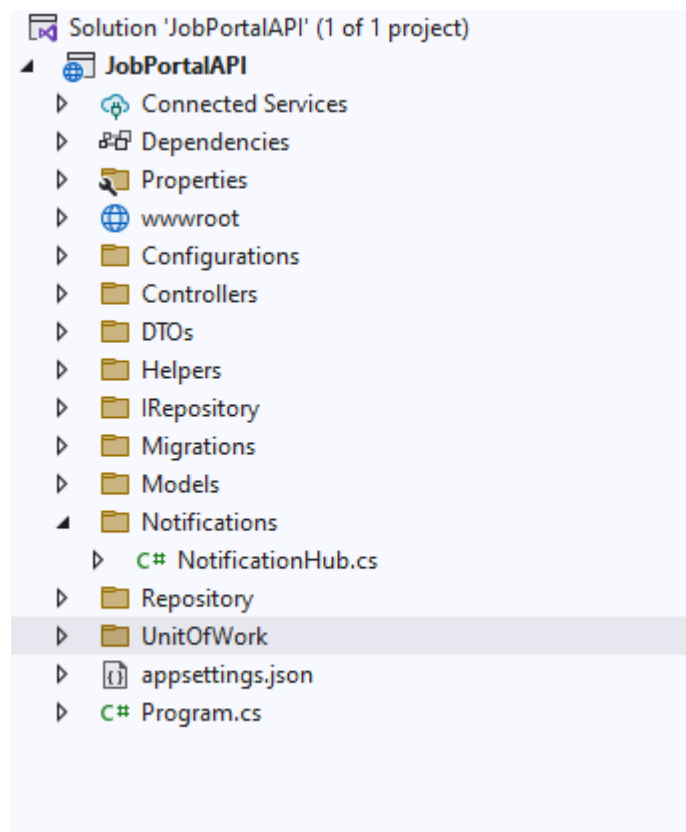
الشكل التوضيحي 18: الجداول التي تقدمها الحزمة identity.

- **SignalR:Microsoft.AspNetCore.SignalR** هي مكتبة تواصل بالزمن الحقيقي الفعلي، تتيح للمخدّم إمكانية تنجيز تطبيقات تعمل بالزمن الحقيقي الفعلي، تم الاستفادة من هذه المكتبة لإضافة ميزة إرسال الإشعارات بالزمن الحقيقي الفعلي بين المستخدمين.
- **Entity Framework Core:Microsoft.EntityFrameworkCore** توفر هذه الحزمة مكتبة Entity Framework Core، وهي عبارة عن إطار عمل ORM للعمل مع قواعد المعطيات. يسمح للمطورين بالتعامل مع قواعد المعطيات باستخدام البرمجة غرضية التوجه.
- **Scaffold:Microsoft.EntityFrameworkCore.Design** توفر هذه الحزمة أدوات مثل Scaffold و Migration. يتم استخدامها عادةً أثناء عملية البناء، لإنشاء قاعدة المعطيات أو الرماز وذلك بحسب النموذج المعتمد.
- نموذج database first: في هذا النموذج يتم أولاً إنشاء قاعدة المعطيات، ثم تحويلها إلى رماز غرض التوجه باستخدام عملية scaffold.

- نموذج code first: في هذا النموذج يتم أولاً إنشاء رماز غرضي التوجه ثم تحويله إلى قاعدة معطيات من خلال عملية migration.
- Microsoft.EntityFrameworkCore.SqlServer: توفر هذه الحزمة مقدّم قاعدة معطيات SQL Server ل Entity Framework Core والذي يسمح ل Entity Framework Core بالتعامل مع قواعد معطيات Microsoft SQL Server.
- Microsoft.AspNetCore.Mvc.Razor.RuntimeCompilation: تقدّم هذه الحزمة إمكانية التعديل على المشروع، دون الحاجة لإيقاف المشروع وإعادة تشغيله مما يوفر الكثير من الوقت.

1.1. بنية API على المخدم

تم بناء API على عدة طبقات تتخاطب مع بعضها البعض لتقوم بالوظائف المطلوبة من ال API بشكل صحيح، وهذه الطبقات هي:



الشكل التوضيحي 19: بنية المخدم back-end.

- طبقة Models: تتضمن هذه الطبقة مجموعة من الصفوف وتمثّل كيانات قاعدة المعطيات، التي نهم بتخزينها، وإجراء عمليات عليها.



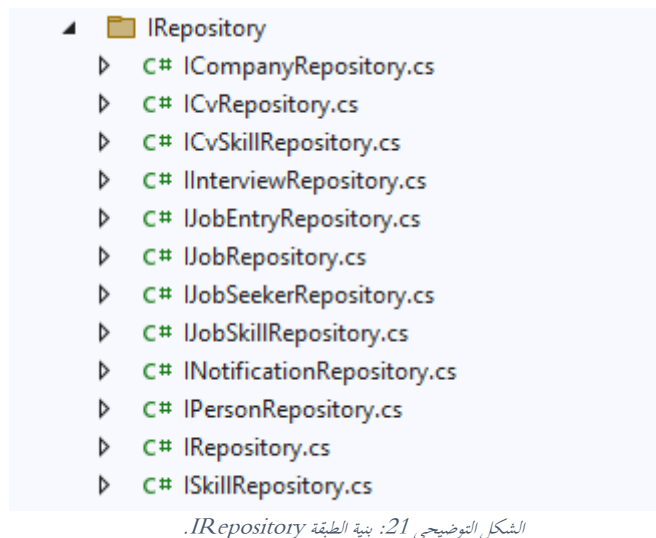
الشكل التوضيحي 20: بنية الطبقة Models.

- طبقة Repositories: وهذه الطبقة تغلف الطبقة Models، وتم إضافة هذه الطبقة لتحقيق

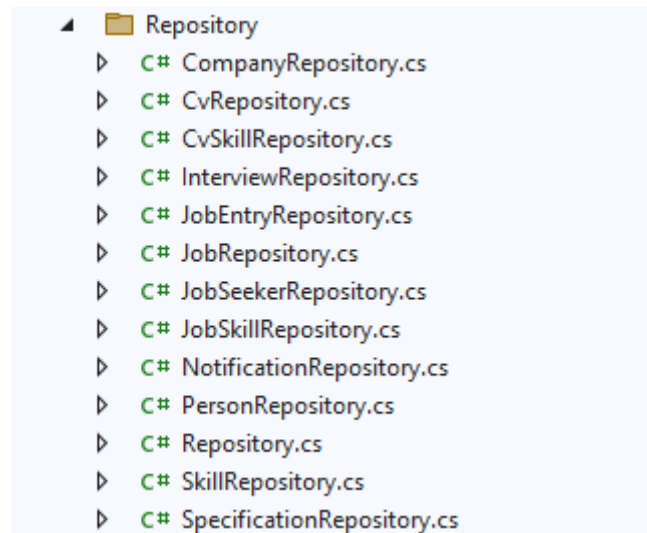
:Repository Design Pattern

Repository Design Pattern: نمط تصميم شائع الاستخدام في تطوير البرمجيات لتوفير طريقة متسقة ومبسطة للوصول إلى البيانات من مصادر البيانات المختلفة، مثل قواعد البيانات أو واجهات برمجة التطبيقات الخارجية. يساعد في تغليف منطق الوصول إلى البيانات ويوفر فصلاً بين طبقة الوصول إلى البيانات وطبقة منطق الأعمال للتطبيق. الفكرة الرئيسية من هذا النمط هي إنشاء طبقة تجريدية (المستودع) تعمل كواجهة تشبه المجموعة للاستعلام عن البيانات ومعالجتها. يخفي المستودع تفاصيل مصدر البيانات الأساسي ويوفر مجموعة من الأساليب الموحدة لأداء عمليات CRUD الشائعة (إنشاء، قراءة، تحديث، حذف).

حيث تم إضافة واجهة من أجل كل صف في طبقة Models تحتوي مجموعة من العمليات التي تطبق على هذا ال model، بالإضافة إلى صف يقوم بتنفيذ هذه الواجهة، ولكوننا نتعامل مع قاعدة معطيات وهناك



مجموعة من العمليات التي تطبق على جميع ال models تم تنفيذ واجهة عمومية تحتوي العمليات الأساسية التي تطبق على جميع ال models وبالطبع تم إنشاء صف لتنفيذ هذه الواجهة.



الشكل التوضيحي 22: بنية الطبقة Repository.

```
namespace JobPortalAPI.IRepository
{
    13 references
    public interface IRepository<T> where T : class
    {
        // Finding Objects
        64 references
        T Get(int id);
        46 references
        IEnumerable<T> GetAll();
        1 reference
        IEnumerable<T> Find(Expression<Func<T, bool>> predicate);

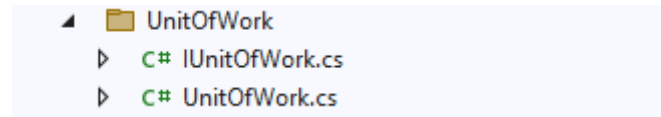
        // Adding Objects
        27 references
        void Add(T entity);
        1 reference
        void AddRange(IEnumerable<T> entities);

        18 references
        void Update(T entity);

        // Removing Objects
        22 references
        void Remove(T entity);
        1 reference
        void RemoveRange(IEnumerable<T> entities);
    }
}
```

الشكل التوضيحي 23: بنية الواجهة العمومية IRepository.

- طبقة UnitOfWork: وهي أيضاً نمط تصميمي شائع الاستخدام، يستخدم لإدارة المعاملات وضمان الإنساق عند العمل مع مستودعات متعددة، يوفر طريقة لتجميع العمليات ذات الصلة في مكان واحدة. الفكرة الأساسية من هذا النمط هي التعامل مع عمليات قاعدة المعطيات المتعددة كمعاملة منطقية واحدة.



الشكل التوضيحي 24: بنية الطبقة UnitOfWork.

```
public interface IUnitOfWork : IDisposable
{
    10 references
    public ICompanyRepository CompanyRepo { get; }
    10 references
    public ICvRepository CvRepo { get; }
    14 references
    public ICvSkillRepository CvSkillRepo { get; }
    14 references
    public IInterviewRepository InterviewRepo { get; }
    22 references
    public IJobRepository JobRepo { get; }
    11 references
    public IJobEntryRepository JobEntryRepo { get; }
    13 references
    public IJobSeekerRepository JobSeekerRepo { get; }
    15 references
    public IJobSkillRepository JobSkillRepo { get; }

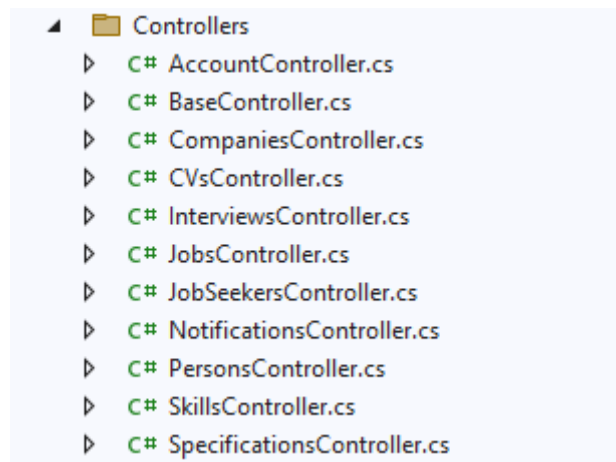
    17 references
    public INotificationRepository NotificationRepo { get; }
    40 references
    public IPersonRepository PersonRepo { get; }
    24 references
    public ISkillRepository SkillRepo { get; }

    22 references
    public ISpecificationRepository SpecificationRepo { get; }
    61 references
    public int SaveChanges();
}
```

الشكل التوضيحي 25: بنية الواجهة IUnitOfWork.

- طبقة Controllers: تقوم هذه الطبقة بالعديد من المهام:
 - استدعاء منطق العمل المناسب: بمجرد أن يحدد ال controller الإجراء المطلوب تنفيذه، فإنه يستدعي الأساليب أو الخدمات المقابلة في طبقة منطق الأعمال (Models) لمعالجة الطلب وتنفيذ العمليات الضرورية مثل استرداد البيانات أو التحقق من صحتها أو التحديثات.

- تحضير الاستجابة: بعد أن يكمل منطق الأعمال (Models) عملياته، يقوم المراقب بإعداد الاستجابة لإرسالها مرة أخرى إلى العميل. قد يتضمن ذلك تنسيق البيانات، واختيار قالب عرض مناسب، وتعبئته بالمعلومات الضرورية.
- معالجة الاستثناءات والأخطاء: يتعامل ال controller أيضًا مع أي استثناءات أو أخطاء قد تحدث أثناء معالجة الطلب. قد يقوم بتسجيل الأخطاء وتقديم رسائل الخطأ المناسبة واتخاذ الإجراءات التصحيحية، مثل عرض صفحات الخطأ أو إعادة توجيه المستخدم.



الشكل التوضيحي 26: بنية الطبقة Controllers.

● طبقة DTOs:

- وهي نمط تصميمي شائع في تطوير البرامج لنقل البيانات بين طبقات أو مكونات مختلفة للتطبيق. تُستخدم DTOs لتغليف ونقل البيانات بتنسيق منظم، مما يسهل تبادل المعلومات بين أجزاء مختلفة من النظام.
- الغرض الأساسي من DTOs هو توفير تمثيل معياري جيد التحديد للبيانات التي يمكن تسلسلها بسهولة ونقلها عبر شبكة أو تمريرها بين طبقات مختلفة من التطبيق. عادةً ما تحتوي DTOs فقط على حقول البيانات الضرورية ولا تتضمن السلوك أو منطق الأعمال.
- فيما يلي بعض خصائص وفوائد استخدام DTOs:
 - تغليف البيانات: تقوم DTOs بتغليف البيانات التي يجب نقلها، مما يوفر تمثيلًا منظمًا يمكن فهمه بسهولة من قبل كل من المرسل والمستقبل.

- تحويل البيانات: تسمح DTOs بتحويل البيانات عن طريق تعيين حقول كيان أو كائن إلى بنية مختلفة أكثر ملاءمة لحالة أو طبقة استخدام معينة.
- الأمان والخصوصية: يمكن لـ DTOs المساعدة في فرض الأمان والخصوصية عن طريق إخفاء حقول البيانات الحساسة عند نقل المعلومات بين المكونات أو الطبقات المختلفة.

- يتم الربط بين DTOs و Models باستخدام AutoMapper.
- تم إضافة قسم خاص بال NotificationHub والذي يسمح بإرسال الإشعارات بين المستخدمين بالزمن الحقيقي الفعلي، وهو قائم على فكرة إنشاء socket بين إجرائيتين. تعرّف socket من خلال أربعة عناصر IP Source, Port Sourc, IP Destination, Port Destination، ويتم فتح stream للقراءة والكتابة بين الإجرائيتين، وفي حالتنا عندما تريد إجرائية المستخدم الأول إرسال اشعار لإجرائية المستخدم الثاني تكتب على ال Stream باستخدام NotificationHub وتقرأ إجرائية المستخدم الثاني هذا الإشعار.

2.1. اختبار ال API

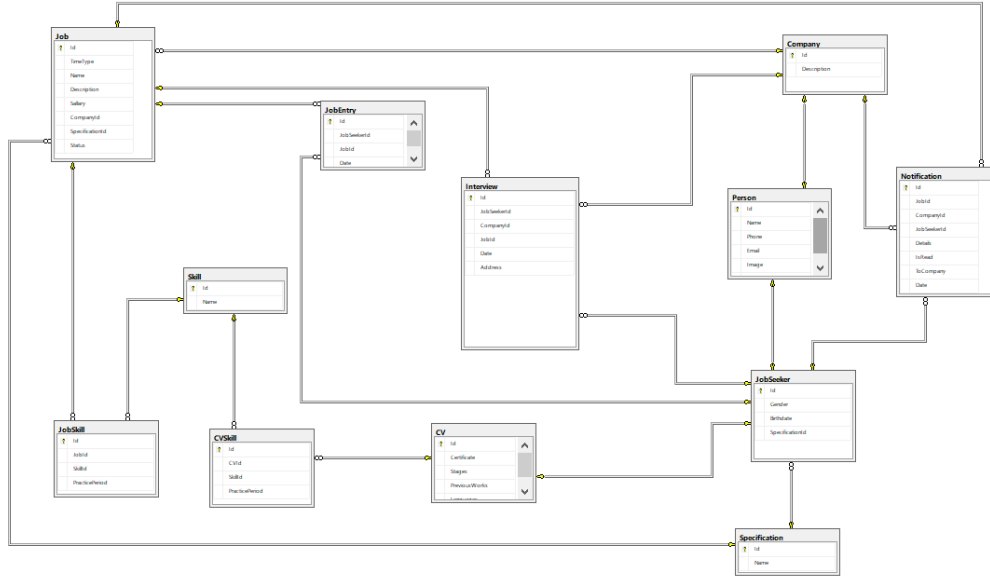
تم اختبار الواجهة API التي أنجزناها باستخدام أداتين مهمتين:

- Swagger.
- Postman.

حيث تم اختبار ال endpoints الموجودة ضمن ال API من خلال هاتين الأداتين من خلال اختبار إرسال طرود Http بمختلف الأفعال القياسية لبروتوكول http لهذه ال endpoints التي تتلقى هذه الطلبات وتقوم بمعالجتها وإعادة النتيجة إلى الطرف المرسل.

2. قاعدة المعطيات

قاعدة المعطيات هي مجموعة منظمة من البيانات التي يمكن الوصول إليها وإدارتها بسهولة. تم إنشاء قاعدة المعطيات الخاصة بالمشروع باستخدام SQL Server. نورد فيما يلي مخطط ERD يوضح تصميم قاعدة المعطيات:



الشكل التوضيحي 27: تصميم قاعدة المعطيات.

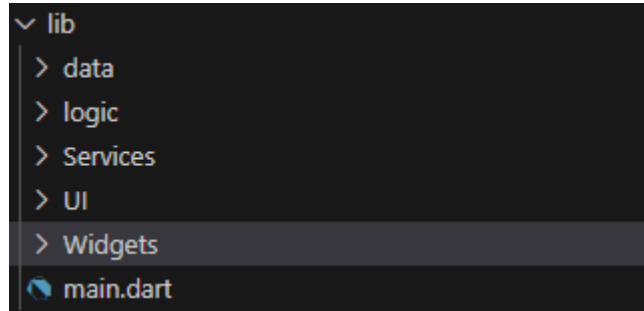
1.2. شرح الكيانات

- **Person:** النظام موجه للاستخدام من قبل دورين وهما طالب وظيفة وممثل شركة، ونظراً لوجود واصفات مشتركة بين طالب الوظيفة وممثل الشركة تم إنشاء الكيان **Person** ليحتوي هذه الوصفات المشتركة، ويكون كل كيان من نمط **Person** هو إما ممثل شركة أو طالب وظيفة.
- **JobSeeker:** يمثل طالب الوظيفة.
- **Company:** يمثل ممثل الشركة.
- **CV:** كل طالب وظيفة له **CV** واحد.
- **CVSkill:** كل **CV** يحتوي على عدة مهارات.
- **Job:** ممثل الشركة يقوم بإضافة عمل جديد على النظام.
- **JobSkill:** كل عمل يتطلب مجموعة من المهارات.
- **Specification:** طالب الوظيفة يعمل ضمن اختصاص معين، كذلك الأمر للعمل المطروح فهو ينتمي إلى اختصاص معين.
- **JobEntry:** يقدم طالب الوظيفة طلب توظيف على عمل معين.
- **Interview:** يحجز ممثل الشركة موعداً لمقابلة طالب الوظيفة من أجل عمل معين.
- **Notification:** يتم إرسال إشعار من طالب وظيفة إلى ممثل شركة عندما يقوم بتقديم طلب توظيف، كما يتم إرسال إشعار من ممثل شركة إلى طالب وظيفة عند حجز/تعديل/إلغاء مقابلة.

3. واجهات المستخدم

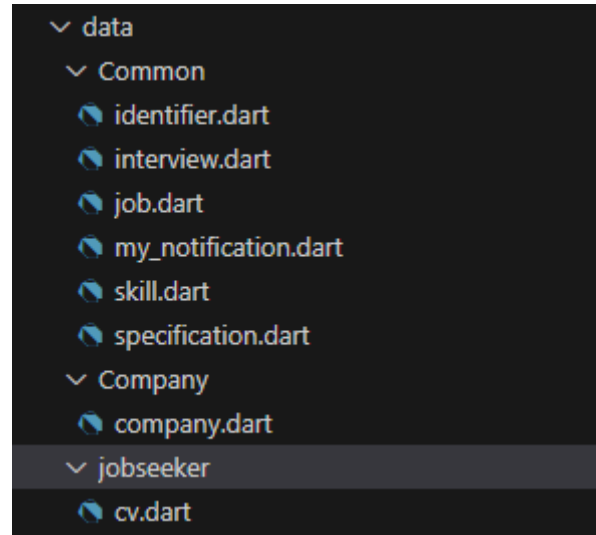
1.3. بنية front-end

تم بناء واجهات المستخدم باستخدام بيئة عمل Flutter، حيث قمنا ضمن هذه البيئة ببناء التطبيق الذي يحتوي واجهات المستخدم، تم تقسيم المشروع إلى عدة طبقات بسيطة وهي كالتالي:



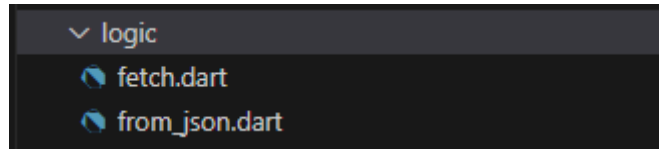
الشكل التوضيحي 28: بنية front-end.

- طبقة data: تتضمن مجموعة من الصفوف التي تعرف الأغراض التي نتمتع بالتعامل معها ضمن التطبيق، أي الأغراض التي نرسلها ونستقبلها من وإلى API.



الشكل التوضيحي 29: بنية الطبقة data.

- طبقة logic: تتضمن منطق التعامل مع ال API وهي قسمين:



الشكل التوضيحي 30: بنية الطبقة *logic*.

- **fetch**: وتتضمن جميع طلبات http من نوع Get التي من خلالها نحصل على المعلومات من ال API.

```
class fetch {  
  static Future<CV> fetchCV() async { ...  
  static Future<Company> fetchCompany() async { ...  
  static Future<Company> fetchCompanyById(int id) async { ...  
  static Future<List<Job>> getAllJobs() async { ...  
  static Future<Job> getJobById(int id) async { ...  
  static Future<List<Job>> getAllJobsByCompanyId() async { ...  
  static Future<List<Job>> getAllJobsByJobseekerId() async { ...  
  static Future<int> getNumberOfJobseekerSync(int id) async { ...  
  static Future<int> getNumberOfInterviewsByJobId(int id) async { ...  
  static Future<int> getNumberOfInterviewsByJobseekerId(int id) async { ...  
  static Future<int> getNumberOfCompanyNotifications(int id) async { ...  
}
```

الشكل التوضيحي 31: توابع الصف *fetch*.

- `from_json`: تتضمن جميع عمليات التحويل بين json والصفوف الموجودة ضمن طبقة `data`.

```
class fromJson {
> static CV toCv(Map<String, dynamic> json) { ...
> static Company toCompany(Map<String, dynamic> json) { ...
> static Job toJob(Map<String, dynamic> json) { ...
> static MyNotification toMyNotification(Map<String, dynamic> json) { ...
> static Interview toInterview(Map<String, dynamic> json) { ...
> static Identifier toIdentifier(Map<String, dynamic> json) { ...
> static String toErrors(Map<String, dynamic> json) { ...
}
```

الشكل التوضيحي 32: توابع الصف `fromJson`.

- طبقة `Services`: تتضمن مجموعة من الخدمات التي تقدّم على مستوى التطبيق:

```

└─ Services
   └─ Common
      ├── new_notification_manager.dart
      ├── signalR.dart
      ├── global_methods.dart
      └── global_variables.dart

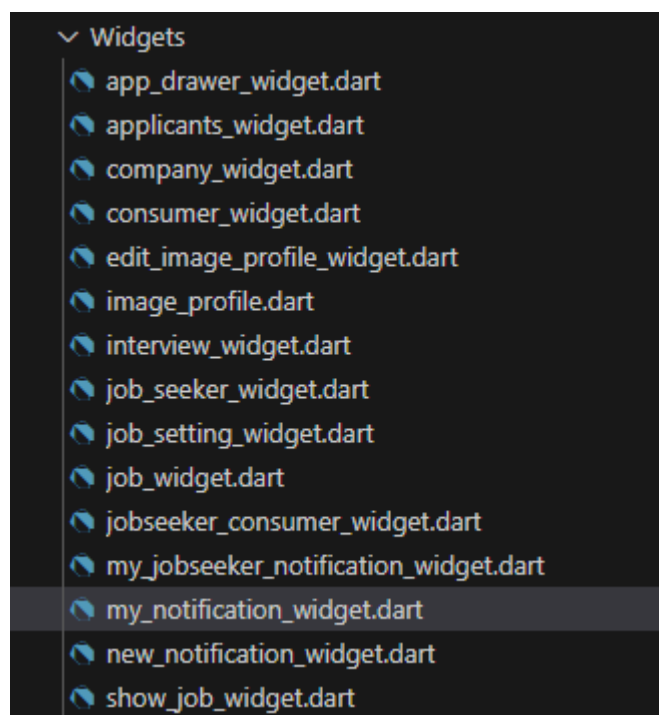
```

الشكل التوضيحي 33: بنية الطبقة `Services`.

- تتضمن مجموعة من التوابع المستخدمة على مستوى التطبيق، مثل تابع استلام الأخطاء القادمة من المخدم وعرضها ضمن شكل معيّن.
- تتضمن مجموعة من المتحولات المستخدمة على مستوى التطبيق، أهمها المتحول `identifier` الذي يحتوي `id` المستخدم الذي قام بتسجيل دخول بالإضافة إلى `token` التي أرسلها المخدم إلى المستخدم بعد تسجيل الدخول أول مرة بنجاح.
- خدمة `signalR` التي تم استخدامها لاستلام إشعار في الزمن الحقيقي الفعلي.
- خدمة إدارة وصول إشعار جديد، والتي تستخدم لتحديد ما إذا كان هناك إشعارات جديدة قد وصلت إلى التطبيق وتحديد عددها، وإعلام الصفحات التي نتم بعرض وصول إشعار جديد ضمنها.
- والذي يتيح لنا عملية الإدارة هذه هي الحزمة `provider` والتي سنأتي على ذكرها لاحقاً.
- طبقة `UI`: وهي الطبقة الأهم ضمن التطبيق، والتي تحتوي جميع الواجهات المعروضة للمستخدم، يتم بناء واجهات التطبيق على شكل مكّس `stack` من الصفحات، حيث نبدأ بعرض صفحة البداية ولانتقال إلى الصفحات الأخرى نستخدم الـ `Navigator` الذي يتيح لنا العمليات التالية:

- `push`: تسمح هذه التعليمة بإضافة صفحة جديدة إلى ال `stack` دون حذف أيّاً من الصفحات السابقة.
- `pushReplacement`: تسمح هذه التعليمة بتبديل الصفحة الحالية (أعلى صفحة) ضمن ال `stack` بصفحة جديدة.
- `pop`: تسمح هذه التعليمة بحذف الصفحة الحالية من ال `stack` والعودة إلى الصفحة السابقة ضمن ال `stack`.

كل صفحة من هذه الصفحات تُبنى على شكل شجرة من العناصر `widgets`، حيث يتم استدعاء عنصر أب `main widget` وضمن هذا العنصر نستدعي عناصر أخرى حتى نقوم بتشكيل الصفحة المطلوبة. كل `widget` يكون لها شكل معيّن وقيم تنسيق افتراضية، ويمكن تعديل تنسيق كل `widget` من خلال تعليمات تشبه إلى حد كبير تعليمات لغة `css`. كما يمكن أن نقوم بتشكيل `widgets` خاصة بنا لاستخدامها التطبيق، حيث قمنا بإنشاء `widgets` عامة تستخدم على مستوى التطبيق، بالإضافة إلى مجموعة `widgets` خاصة بكل صفحة.



الشكل التوضيحي 34: أهم ال `widgets` التي تم إنشاؤها.

2.3. إدارة حالة الصفحة

من أهم المواضيع التي يجب معالجتها ضمن بيئة عمل `Flutter` هي إدارة حالة الصفحة، والمقصود بإدارة حالة الصفحة هو تغيير محتوى الصفحة التي يتم عرضها بناءً على تغيّر الدخل المعتمدة عليه هذه الصفحة، حيث يتم بناء الصفحة بناءً

على دخل أولي ومن ثم يتفاعل المستخدم مع الصفحة بعمليات غالباً ما تؤدي إلى تغيير الدخل الذي بُنيت على أساسه الصفحة. تقدّم بيئة عمل Flutter العديد من الطرق لإدارة حالة الصفحة وأهمها:

▪ **Provider Package:** تسمح هذه الحزمة لنا بإمكانية جعل صفحات أو أجزاء من الصفحات ضمن التطبيق تَتم بحدث معيّن. عند تغيير هذا الحدث يتغيّر محتوى الصفحة أو أجزاء منها التي تَتم بهذا الحدث، تقوم هذه الفكرة على وجود provider و consumer ال provider هو المعني بإحداث التغييرات وإعلام جميع ال consumers التي تنصت على هذا ال provider لتقوم بتغيير محتوى الصفحة. تتم عملية إعلام ال consumers من خلال التابع notifyListeners() الذي يُعلم جميع ال consumers بوجود تغيير قد حدث لذلك يقوم هؤلاء ال consumers بإعادة بناء الصفحات بناءً على دخل جديد. يمكن أن يكون ال provider يُعلم ال consumers بالتعديلات التي تحدث على مستوى التطبيق ككل أو التي تحدث ضمن عدة صفحات معينة أو فقط ضمن صفحة واحدة وهذا يعود إلى الحالة المدروسة وقرار مطور التطبيق.

الحالة المدروسة في تطبيقنا:

تم استخدام الحزمة signalR لتلقّي الإشعارات التي يرسلها المخدم إلى المستخدمين، هذه الإشارة تُعلم إجرائية المستخدم أن لديه إشعاراً جديداً وقد حدث لدينا تغيير. يتطلب هذا التغيير تغيير محتوى الصفحة التي يستخدمها المستخدم حالياً. لكي نُعلم هذه الصفحة بوجود تغيير عند استلام إشعار جديد تم استخدام Provider Package حيث يتم إضافة Consumers إلى الصفحات التي نَتم بعرض الإشعار الجديد ضمنها، عندما تصل إشارة بوجود إشعار جديد تستلمه إجرائية signalR الخاصة بطرف الزبون، يقوم ال provider بإعلام جميع ال consumers بالتغيير الذي حدث ليقوموا بدورهم بعرض رسالة تتضمن وجود إشعار جديد للمستخدم.

▪ **StateFulWidget:** تتيح لنا بيئة عمل Flutter بإمكانية جعل أي صفحة ترث من الصف StateFulWidget مما يعطي هذه الصفحة إمكانية إنشاء حالات، عندي كل حالة يتم إعادة بناء الصفحة بناءً على الدخل الذي تحدده هذه الحالة. حيث تتم هذه العملية من خلال التابعين اللذين يقدّمهما الصف StateFulWidget وهما:

❏ initState(): يحدّد الحالة البدائية للصفحة، والتي على أساسها يتم بناء الصفحة أول مرة.

❏ setState(): عند إجراء عمليات من قبل المستخدم تؤدي إلى تغيير الدخل الذي بُنيت عليه

الصفحة، يتم طلب تابع setState() مع تحديد الدخل الجديد، ليتم إعلام بيئة عمل Flutter

أن تقوم بإعادة بناء الصفحة وفقاً للدخل الجديد.

3.3. المكتبات المستخدمة

- تم استخدام مجموعة كبيرة من الحزم والمكتبات ضمن طرف front-end نذكر فيما يلي أهم هذه المكتبات:
- **http: ^1.1.0**: توفر هذه الحزمة مجموعة من الصفوف والوظائف لإجراء طلبات HTTP ومعالجة الاستجابات. مما يسمح لتطبيق Flutter بالتواصل مع خوادم الويب وواجهات برمجة التطبيقات.
 - **url_launcher: ^6.1.12**: توفر هذه الحزمة طريقة لتشغيل عناوين URL وفتح تطبيقات أخرى على جهاز المستخدم. مما يسمح لنا بفتح صفحات الويب وإجراء المكالمات الهاتفية وإرسال رسائل البريد الإلكتروني وغيرها. تم استخدام هذه الحزمة ضمن تطبيقنا لتوفير آليات اتصال بين المستخدمين على التطبيق.
 - **curved_navigation_bar: ^1.0.3**: توفر هذه الحزمة أداة شريط تنقل منحنى قابل للتخصيص لتطبيق Flutter. يسمح لنا بإنشاء شريط تنقل بحواف منحنية ورسوم متحركة سلسة. تم استخدام هذه الحزمة ضمن تطبيقنا لإنشاء شريط انتقال سلس يحقق الانتقال السهل والسلس بين صفحات التطبيق.
 - **image_cropper: ^5.0.0**: توفر هذه الحزمة وظائف لاقتصاص الصور في تطبيق Flutter. مما يسمح للمستخدمين بتحديد صورة ثم قصها إلى الحجم المطلوب أو تحديد نسبة العرض إلى الارتفاع.
 - **image_picker: ^1.0.1**: توفر هذه الحزمة وظائف لاختيار الصور أو مقاطع الفيديو من معرض الجهاز أو التقاطها باستخدام الكاميرا. مما ييسر عملية العمل مع الصور ومقاطع الفيديو في تطبيق Flutter الخاص بنا. تم استخدام هذه الحزمة ضمن تطبيقنا لتمكين المستخدمين من اختيار صورة من المعرض أو باستخدام الكاميرا في حالات إنشاء حساب أو تعديل صورة الملف الشخصي.
 - **signalr_netcore: ^0.1.7+2-nullsafety.0**: تسمح هذه الحزمة لتطبيق Flutter الخاص بنا بالاتصال بخادم SignalR. SignalR عبارة عن إطار عمل للاتصال عبر الويب في الوقت الفعلي يتيح الاتصال ثنائي الاتجاه بين الخادم والعملاء. تم استخدام هذه الحزمة ضمن تطبيقنا لتمكين المستخدمين من تلقي الإشعارات المرسلة اليه من المخدم عند حدث معين بالزمن الفعلي الحقيقي.
 - **flutter_image_compress: ^2.0.4**: توفر هذه الحزمة وظائف لضغط الصور في تطبيق Flutter الخاص بنا. مما يسمح لنا بتقليل حجم ملف الصورة دون خسارة كبيرة في الجودة. تم استخدام هذه الحزمة ضمن تطبيقنا من أجل ضغط الصور التي نقوم بإرسالها للمخدم لتسريع العمليات التي تحتاج نقل صور بين المستخدم والمخدم.
 - **flutter_pw_validator: ^1.6.0**: توفر هذه الحزمة وظيفة التحقق من صحة كلمة المرور لتطبيق Flutter الخاص بنا. وتسمح لنا بفرض متطلبات قوة كلمة المرور، مثل الحد الأدنى للطول ووجود أحرف كبيرة / صغيرة وأرقام وأحرف خاصة.
 - **provider: ^6.0.5**: هذه الحزمة هي حل إدارة الحالة لتطبيقات Flutter. يساعد في إدارة حالة التطبيق ومشاركتها بين أجزاء مختلفة من التطبيق، مما يسهل إنشاء تطبيقات Flutter التفاعلية والقابلة

للتطوير. تم استخدام هذه الحزمة عند إستلام إشعارات من المخدم لنعلم الصفحات التي تهتم بعرض الاشعار
بأنه يوجد إشعار جديد، وعلى هذه الصفحات تعديل حالتها لعرض هذا الإشعار.

الفصل السادس

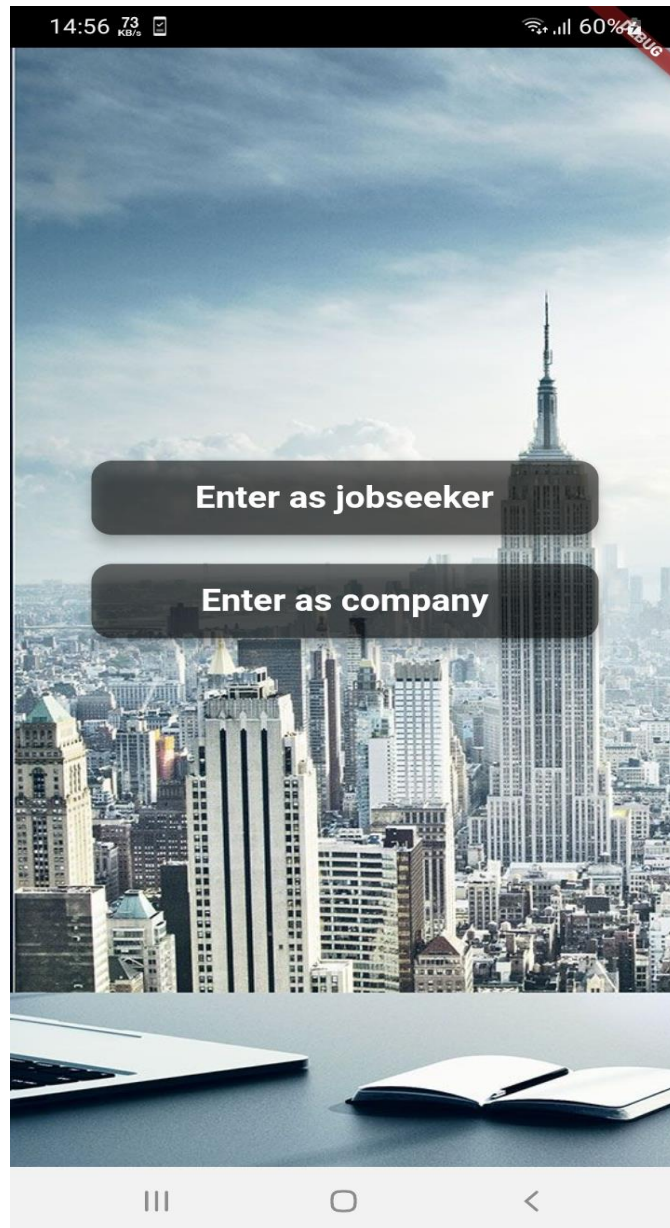
الاختبارات والواجهات

1. مقدمة

سنقوم في هذا الفصل باستعراض أهم واجهات النظام، مع شرح مبسط عن هذه الواجهات وما يمكن للمستخدم أن
يقوم به في كل واجهة.

2. الواجهات

كما ذكرنا سابقاً التطبيق موجه للاستخدام من قبل ممثل شركة وطالب وظيفة، لذلك عند فتح التطبيق يتم سؤال المستخدم عن دوره في النظام.



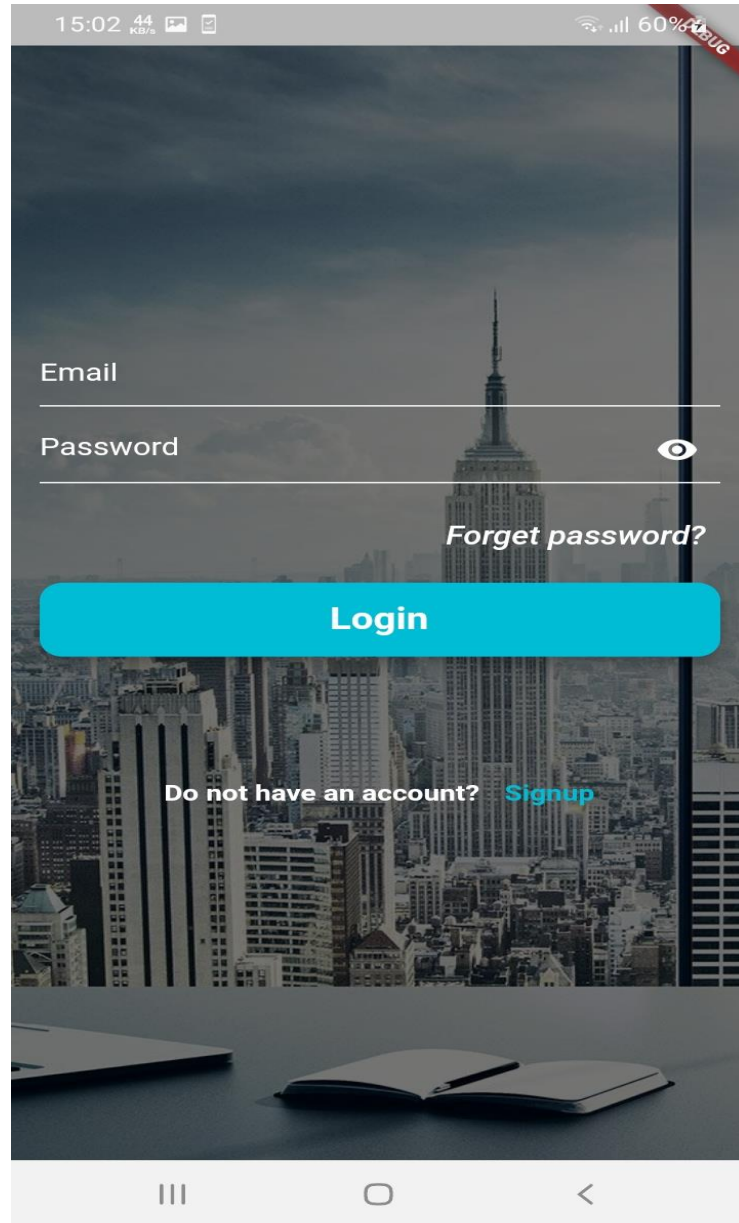
الشكل التوضيحي 35: الواجهة الرئيسية في التطبيق.

ما تقوم به هذه الواجهة هو فقط سؤال المستخدم ما إذا كان يرغب بالدخول كطالب وظيفة او ممثل شركة، بالإضافة إلى خلفية هي عبارة عن صورة متحركة.

1.2. واجهات طالب الوظيفة

نستعرض فيما يلي أهم الواجهات التي سيتعامل معها المستخدم عند الدخول إلى التطبيق كطالب وظيفة.

1.1.2. واجهة تسجيل الدخول كطالب وظيفة



الشكل التوضيحي 36: واجهة تسجيل دخول طالب وظيفة.

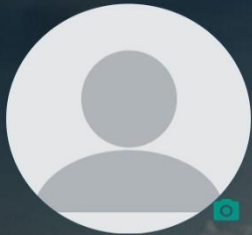
في هذه الواجهة يطلب النظام من المستخدم تسجيل الدخول في حال كان يملك حساب، أو أن يقوم بإنشاء حساب جديد في حال لم يكن يملك حساب سابقاً.

2.1.2. واجهة إنشاء حساب طالب وظيفة

في هذه الواجهة يطلب النظام من المستخدم أن يقوم بإدخال المعلومات اللازمة لإنشاء حساب جديد.

15:07 28 KB/s 61% Aug

Register Your Infomation



Email :

0/50

Password :

0/25


- At least 8 character
- 1 Uppercase letter
- 1 Lowercase letter
- 1 Numeric character
- 1 Special character

User name :

الشكل التوضيحي 38: واجهة إنشاء حساب طالب وظيفة.

15:10 38 KB/s 61% Aug

Register Your Infomation



Email :



0/50

Password :

0/25

- At least 8 character
- 1 Uppercase letter

Choose your profile image

 camera  gallery

الشكل التوضيحي 37: تحديد صورة الملف الشخصي أثناء إنشاء الحساب.

نلاحظ في هذه الواجهة (الشكل التوضيحي 37) أن المستخدم لديه إمكانية إضافة صورة ملف شخصي باستخدام الكاميرا أو باستخدام المعرض (الذي يحتوي الصور المخزنة على ذاكرة الهاتف المحمول)، حيث يطلب منه النظام تحديد مصدر الصورة عند النقر على أيقونة الكاميرا بجانب حقل إدخال الصورة.

الشكل التوضيحي 40: التحقق من قوة كلمة المرور.

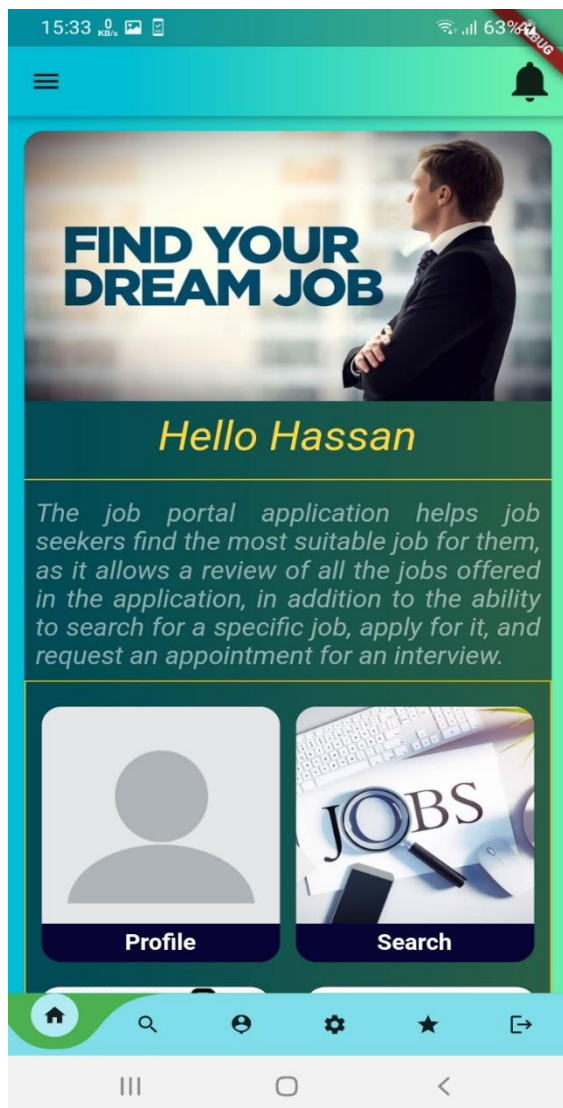
الشكل التوضيحي 39: تحديد الاختصاص الذي يعمل ضمنه طالب الوظيفة.

كما أن النظام يقوم بالتحقق من قوة كلمة المرور التي يدخلها المستخدم، وذلك من أجل التأكد من أن المستخدم أدخل كلمة سر قوية وصعبة الإختراق، وأيضاً يمكن للمستخدم إخفاء أو إظهار كلمة المرور بحسب رغبته. عند إنشاء الحساب يطلب النظام من المستخدم أن يقوم بإدخال اسم الاختصاص الذي يعمل ضمنه، لذلك عندما يريد المستخدم إدخال قيمة هذا الحقل، يُظهر له النظام جميع الإختصاصات الموجودة على التطبيق، مع إمكانية البحث عن اختصاص معين،

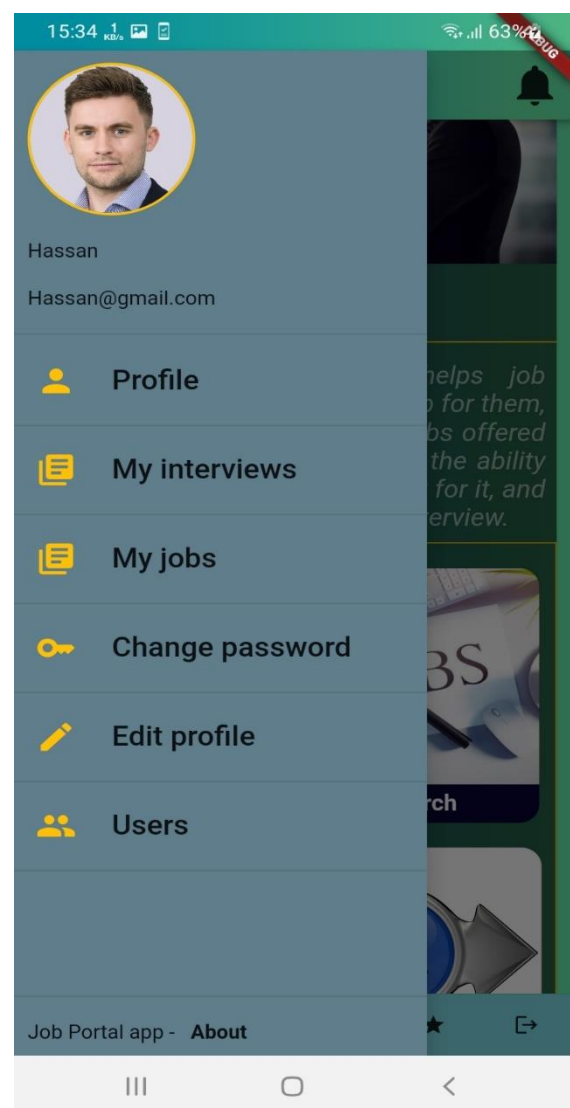
وفي حال عدم العثور على الاختصاص المطلوب، يمكن للمستخدم إضافته إلى النظام، حيث يقوم النظام بالتحقق من أن هذا الاختصاص غير موجود مسبقاً، ويقوم بإضافته إلى قاعدة المعطيات، وإلا يظهر النظام رسالة للمستخدم يُعلمه بها أن الإختصاص الذي يريد إضافته موجود مسبقاً.

3.1.2. الصفحة الرئيسية لطالب الوظيفة

نلاحظ أن هذه الواجهة تحتوي على شريط انتقال سفلي، Dashboard بسيطة و Sidebar جميعها تسمح لطالب الوظيفة بالتنقل بين أهم صفحات التطبيق، وكذلك تسمح هذه الواجهة لطالب الوظيفة باستعراض الإشعارات المرسلة إليها، وسنأتي على ذكر الإشعارات لاحقاً.



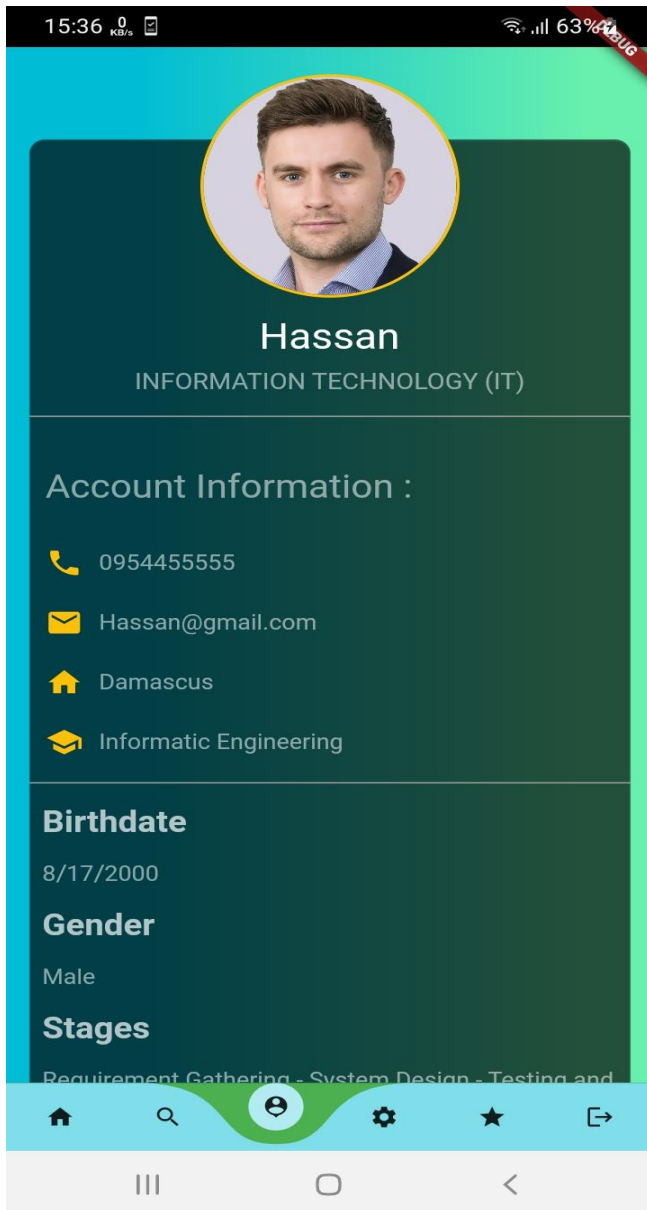
الشكل التوضيحي 42: الصفحة الرئيسية لطالب الوظيفة.



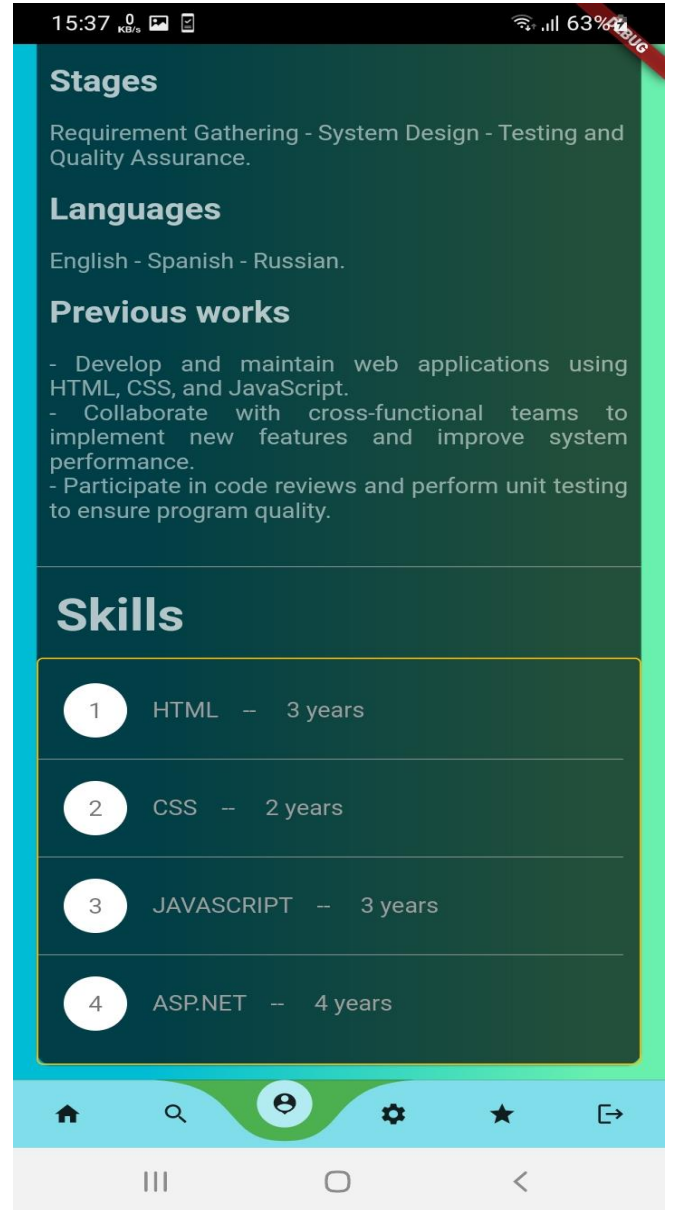
الشكل التوضيحي 41: الصفحة الرئيسية لطالب الوظيفة.

4.1.2. استعراض الملف الشخصي

تسمح هذه الواجهة لطالب الوظيفة باستعراض ملفه الشخصي الذي قام بإنشاءه مسبقاً.



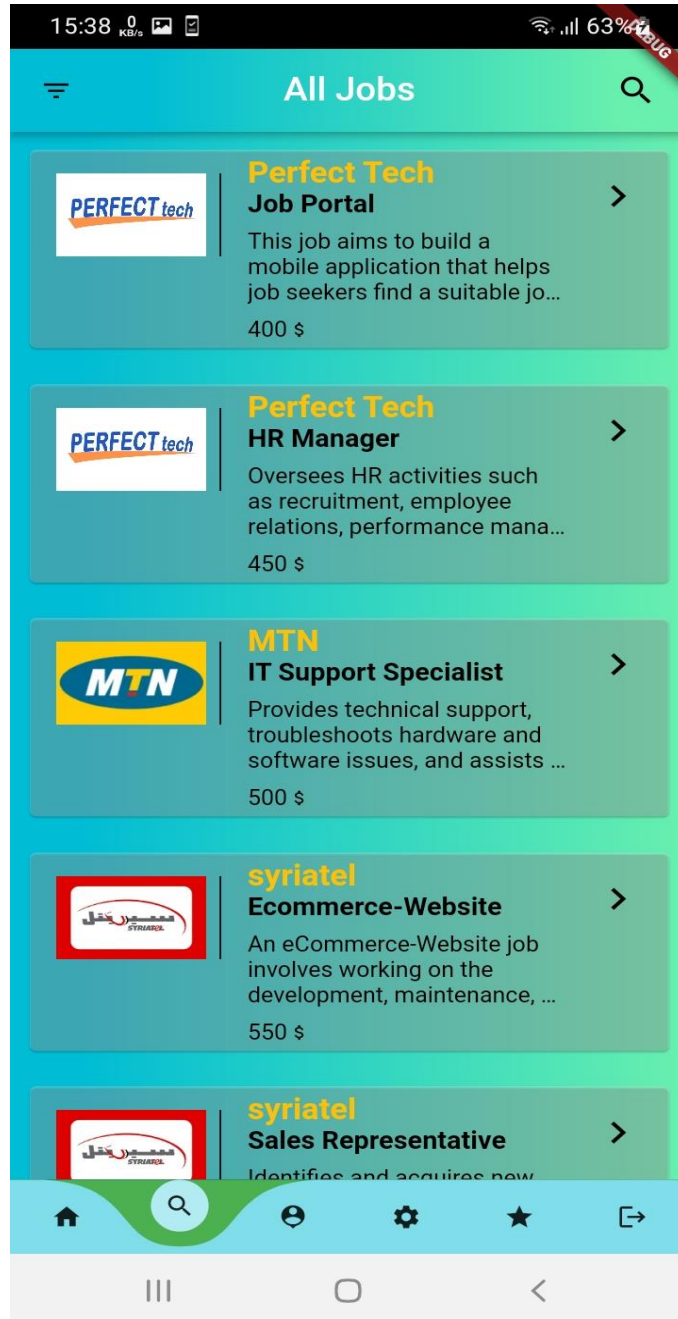
الشكل التوضيحي 44: الملف الشخصي لطالب الوظيفة.



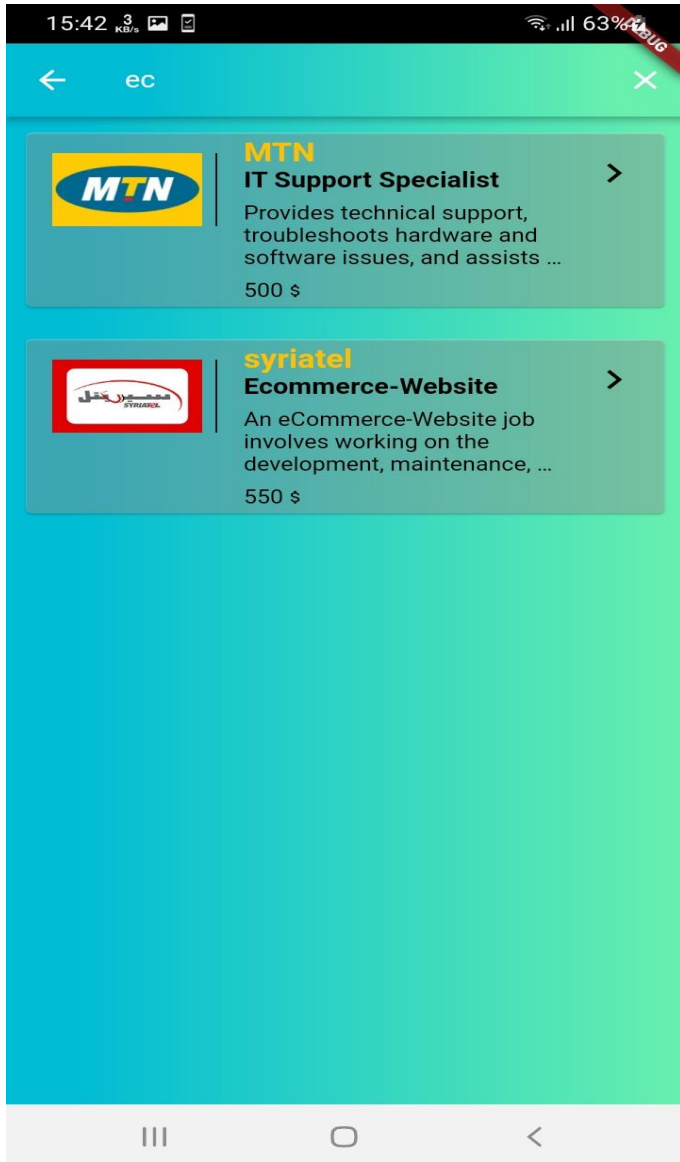
الشكل التوضيحي 43: الملف الشخصي لطالب الوظيفة.

5.1.2. استعراض جميع الأعمال المطروحة على التطبيق من قبل جميع الشركات

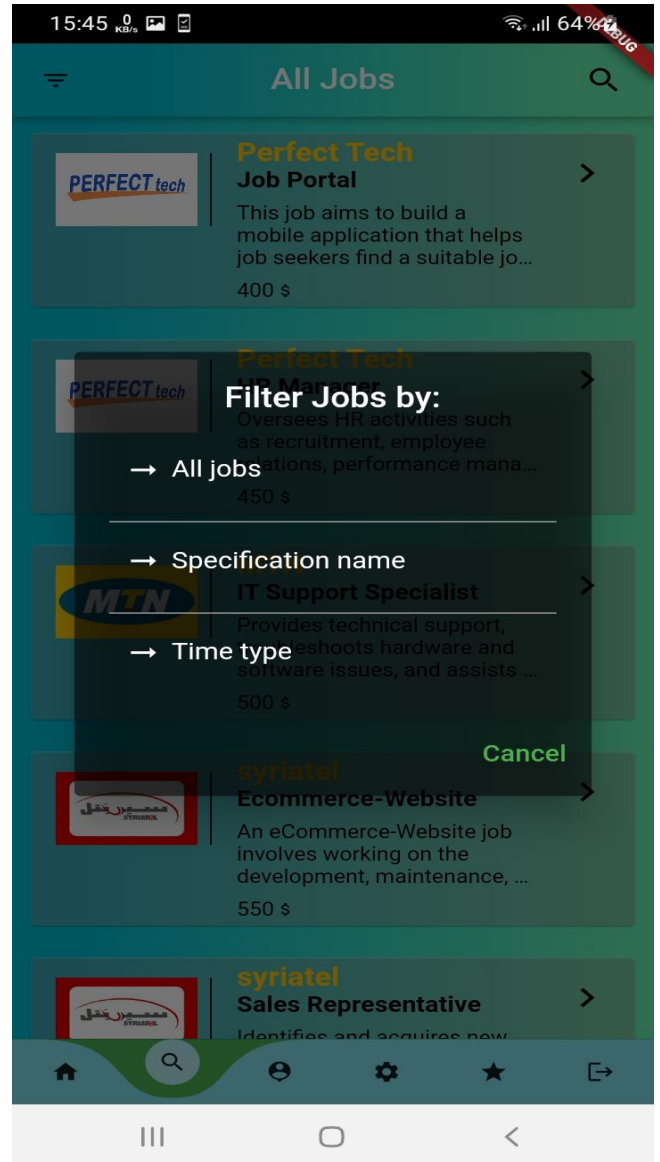
ضمن هذه الواجهة يستعرض طالب الوظيفة جميع الأعمال المطروحة على التطبيق من قبل جميع الشركات مع إمكانية البحث عن عمل معين من خلال الاسم أو إجراء عملية فلتره بحسب بعض خصائص العمل.



1.5.1.2. البحث عن عمل معين بحسب الاسم أو فلترة الأعمال وفقاً لخاصية معينة.



الشكل التوضيحي 46: البحث عن عمل بحسب الاسم.

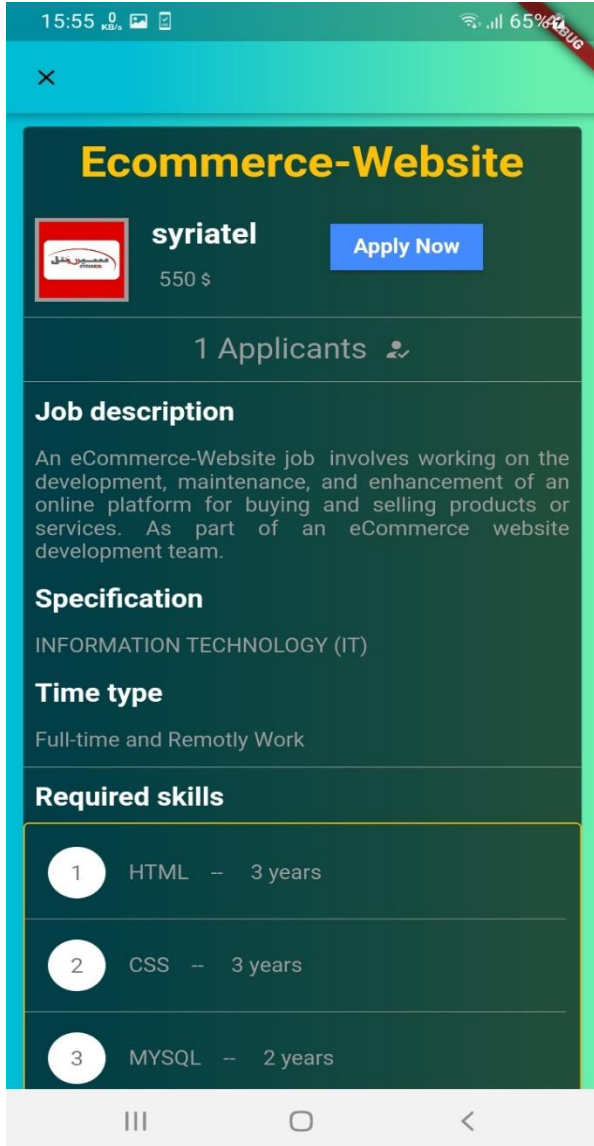


الشكل التوضيحي 45: فلترة الأعمال وفقاً لخاصية معينة.

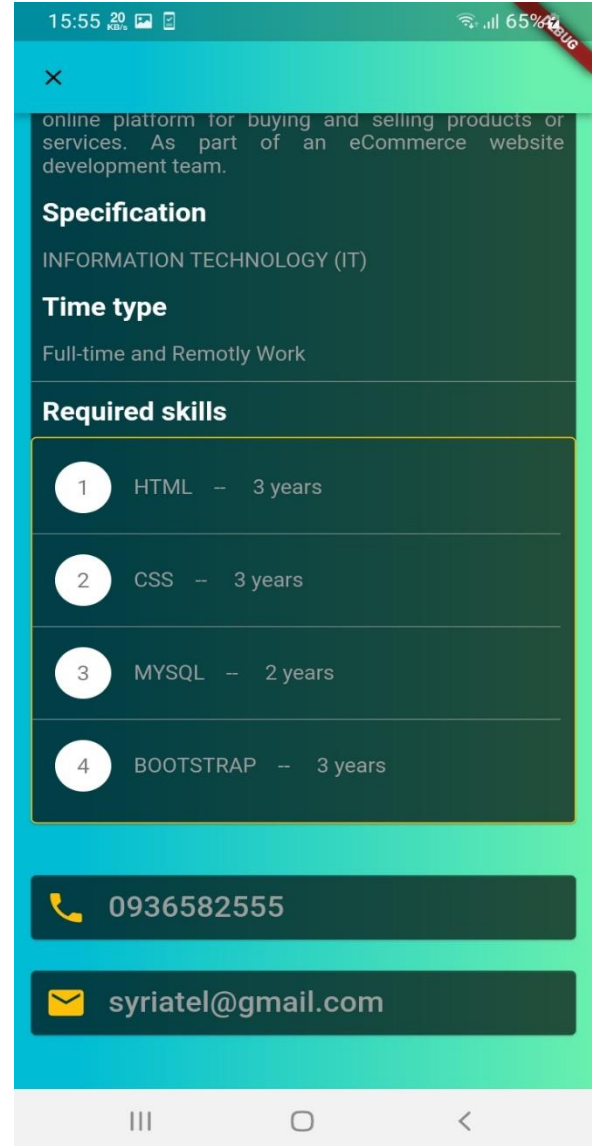
6.1.2. استعراض تفاصيل عمل

- هذه الواجهة تستعرض تفاصيل عمل معين، حيث يظهر لطالب الوظيفة المعلومات الخاصة بالعمل مثل الراتب والاسم والشركة الطارحة ووصف عن العمل والمهارات المطلوبة للعمل، بالإضافة إلى عدد المتقدمين على العمل مع إمكانية استعراض جميع المتقدمين على العمل، واستعراض الملف الشخصي لأحد المتقدمين بالإضافة إلى استعراض الملف الشخصي للشركة الطارحة وذلك من خلال الضغط على صورة الشركة بالإضافة إلى إمكانية

التواصل مع الشركة من خلال رقم هاتف الشركة أو من خلال إيميل الشركة وذلك من خلال الضغط على الإيميل أو الرقم الموجودين أسفل تفاصيل العمل.



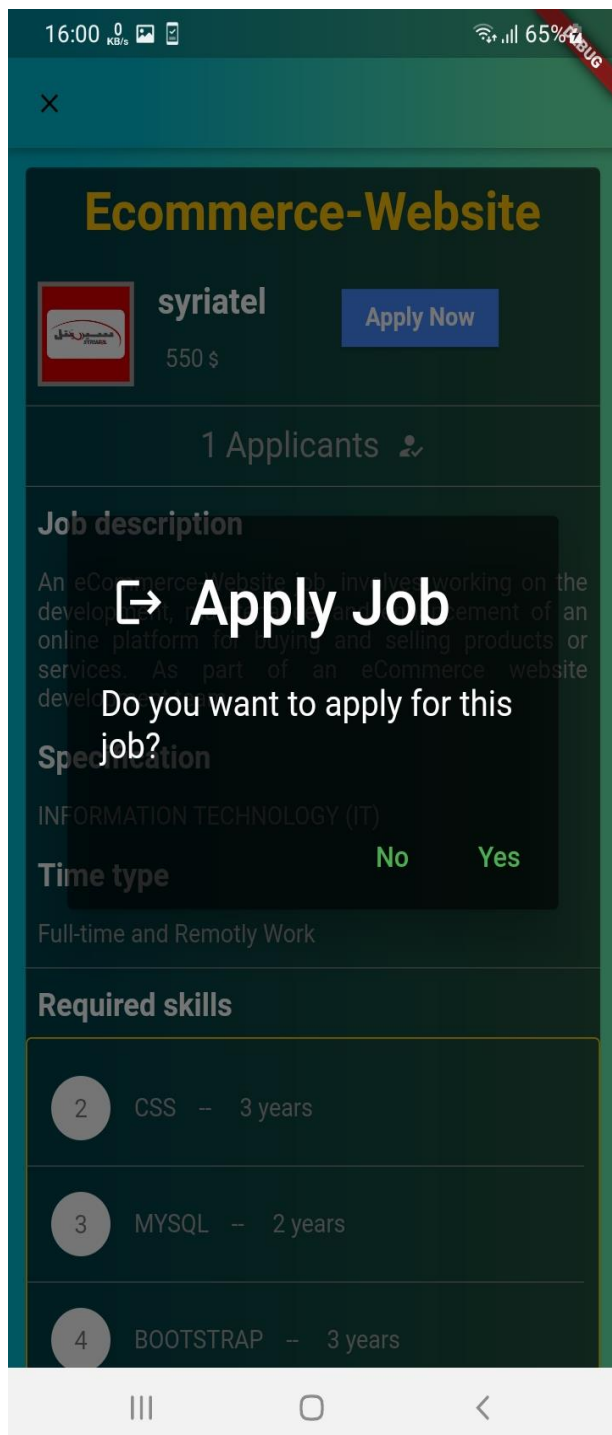
الشكل التوضيحي 48: استعراض تفاصيل عمل من قبل طالب وظيفة.



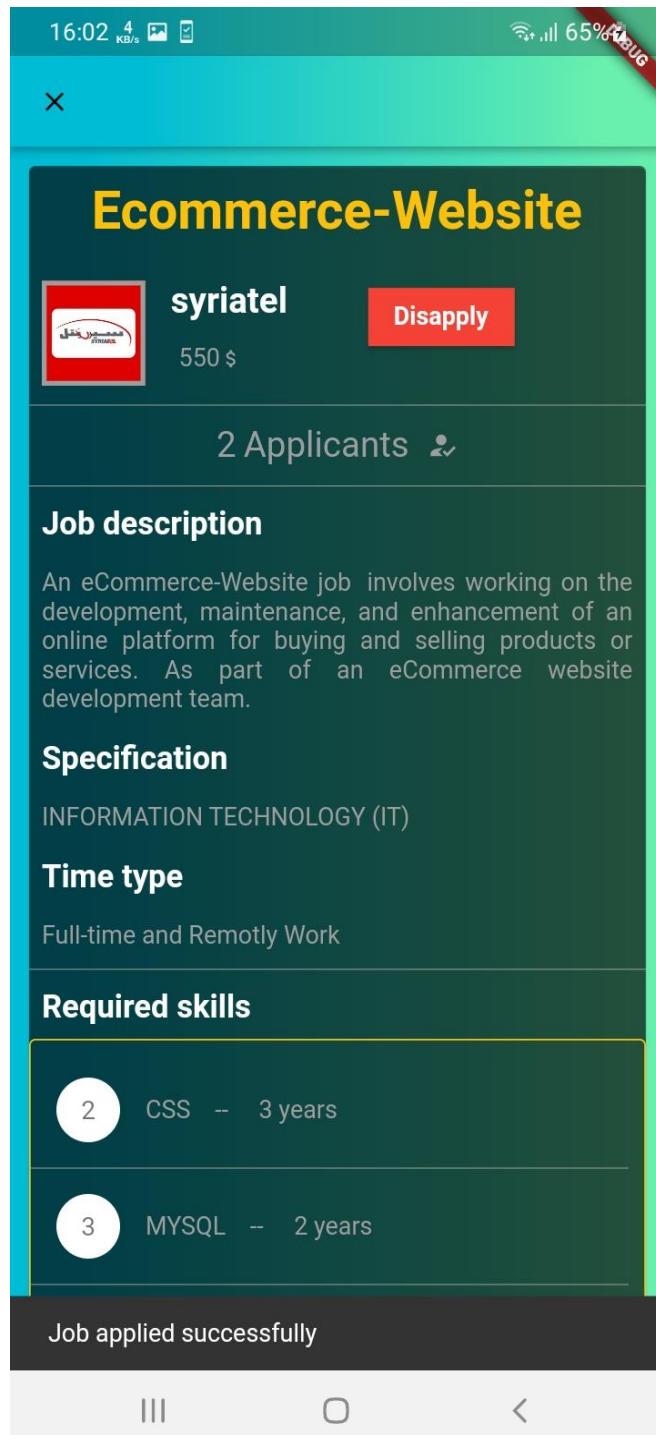
الشكل التوضيحي 47: استعراض تفاصيل عمل من قبل طالب وظيفة.

• التقديم على العمل

يمكن لطالب الوظيفة أن يقوم بتقديم طلب توظيف على العمل الذي استعرض تفاصيله وسيقوم النظام تلقائياً بإرسال إشعار إلى الشركة الطارحة للعمل يُعلمها بوجود طلب توظيف جديد، كما إنه يمكن لطالب الوظيفة أن يقوم بإلغاء الطلب إذا أراد ذلك.



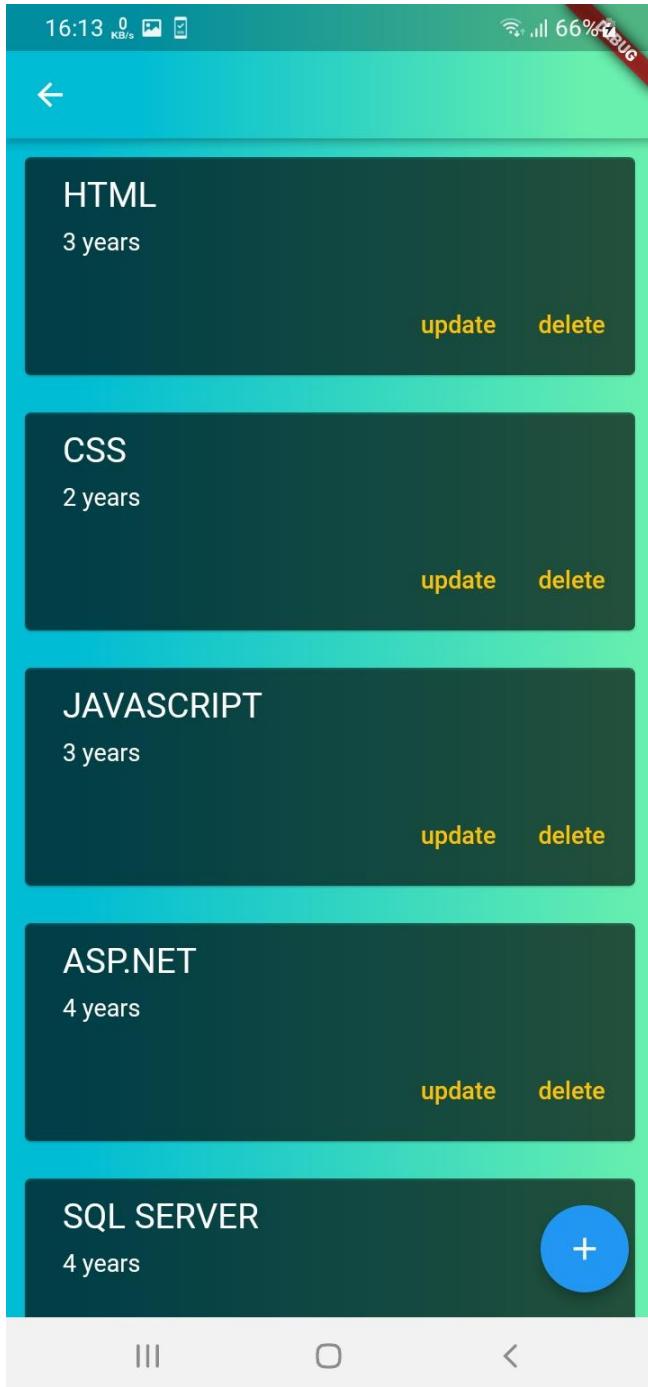
الشكل التوضيحي 50: تقديم طلب توظيف.



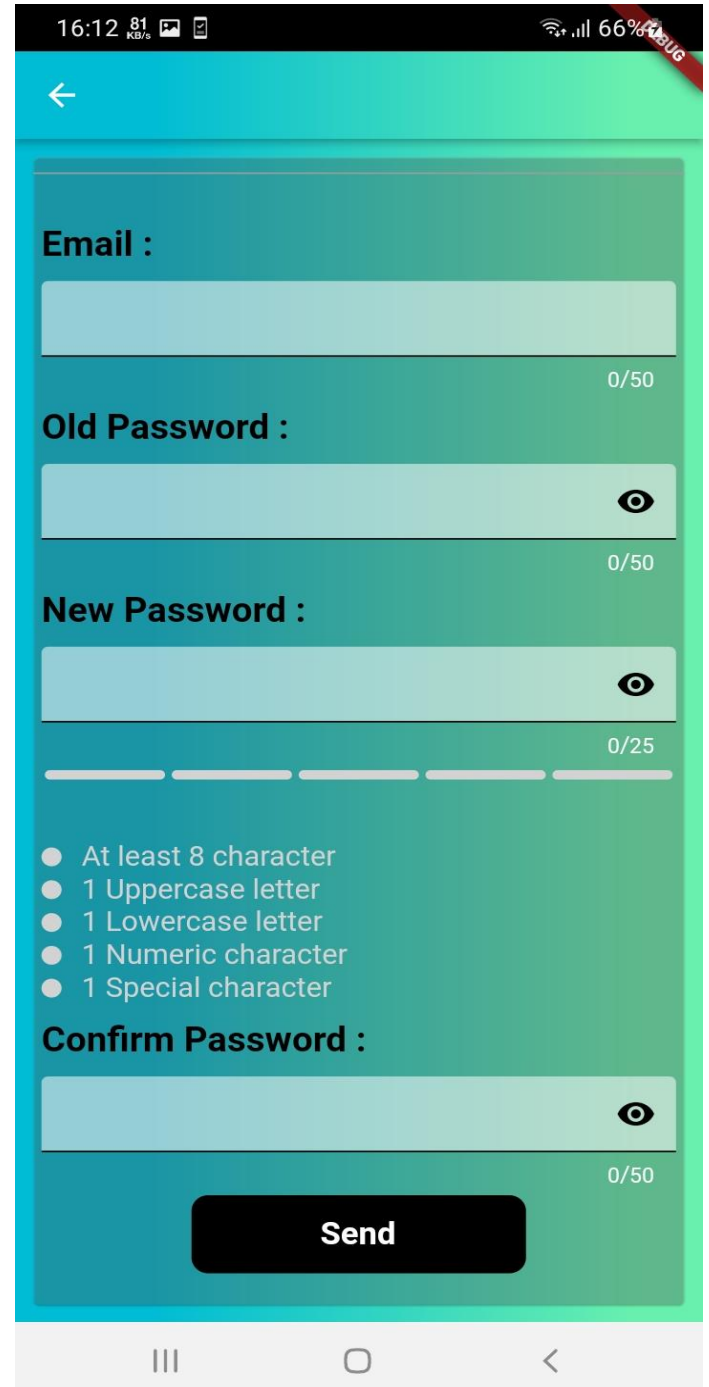
الشكل التوضيحي 49: تقديم طلب توظيف.

7.1.2. تعديل كلمة المرور وتعديل المهارات التي يمتلكها طالب الوظيفة

يمكن لطالب الوظيفة أن يقوم بتعديل معلوماته الشخصية ومهاراته وكلمة المرور الخاصة بالحساب الذي قام بإنشائه، نستعرض فيما يلي الواجهات الخاصة بتعديل كلمة المرور وتعديل المهارات التي يمتلكها.



الشكل التوضيحي 52: تعديل المهارات التي يمتلكها طالب الوظيفة.

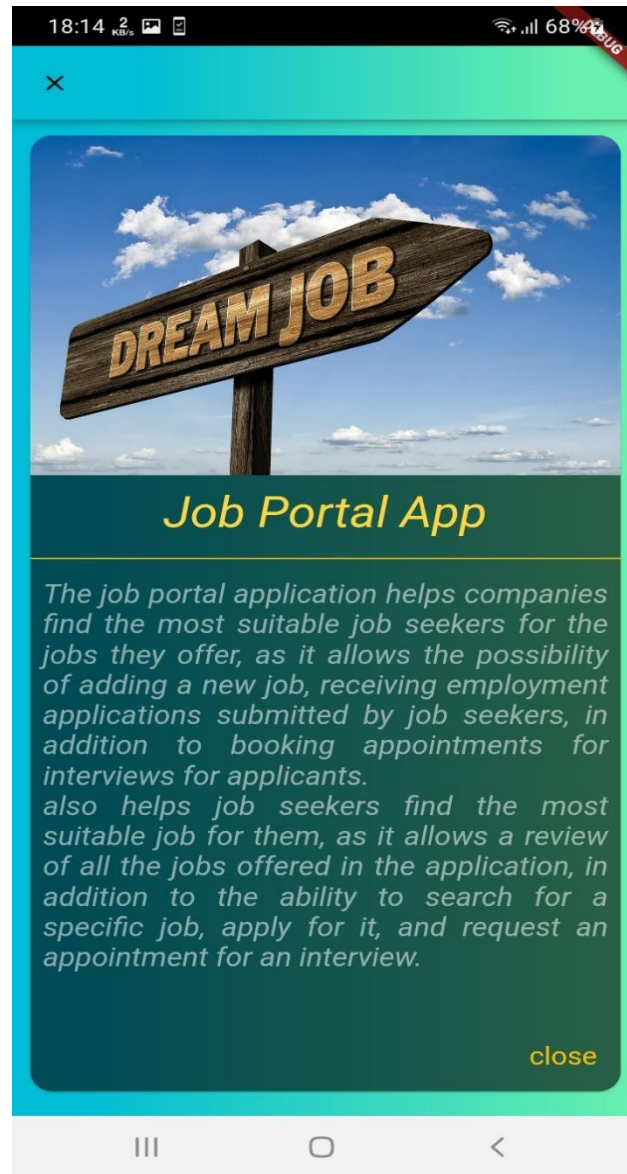


الشكل التوضيحي 51: تعديل كلمة مرور حساب طالب وظيفة.

2.2. واجهات ممثل الشركة

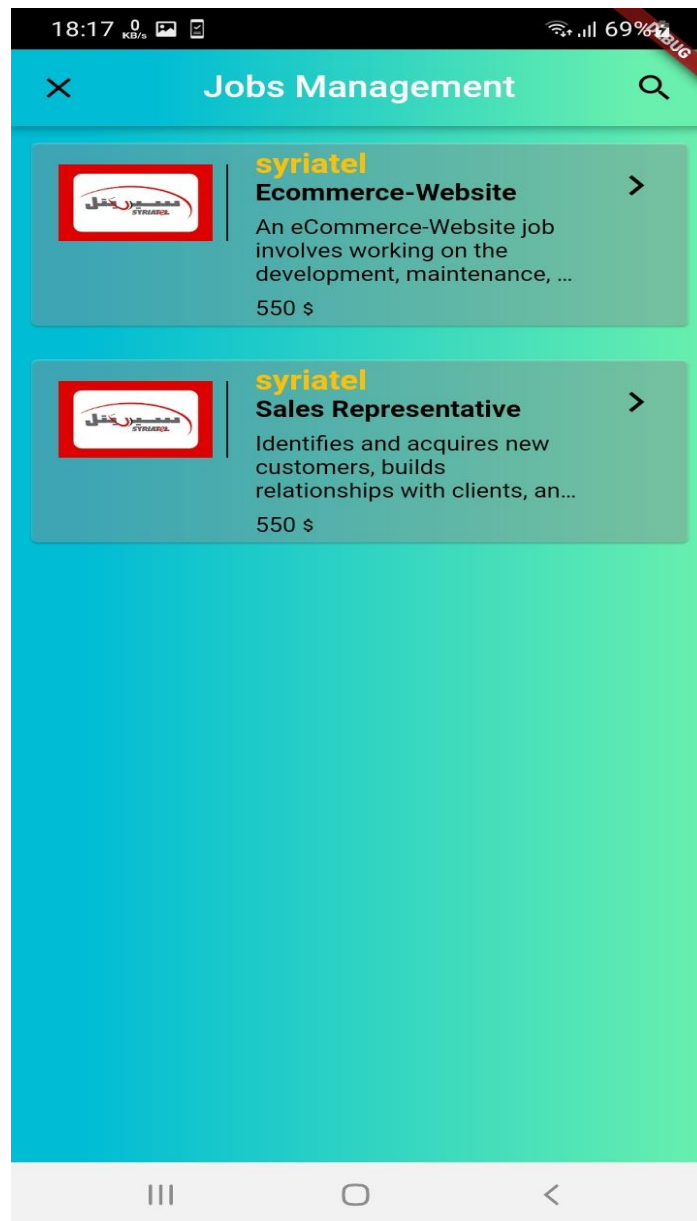
نستعرض فيما يلي أهم الواجهات التي سيتعامل معها المستخدم عند الدخول إلى التطبيق كممثل شركة، ضمن الواجهات الخاصة بممثل الشركة يوجد بعض الواجهات التي تقدّم نفس الوظائف التي استعرضناها في قسم واجهات طالب الوظيفة ولذلك لن نقوم بتكرارها، مثل الواجهات الخاصة بتسجيل الدخول وإنشاء الحساب واستعراض جميع الأعمال المطروحة على التطبيق مع إمكانية البحث والفلتر وتعديل كلمة المرور والعديد من الصفحات الأخرى.

1.2.2. استعراض لمحة عامة حول التطبيق



الشكل التوضيحي 53: استعراض لمحة عامة عن التطبيق.

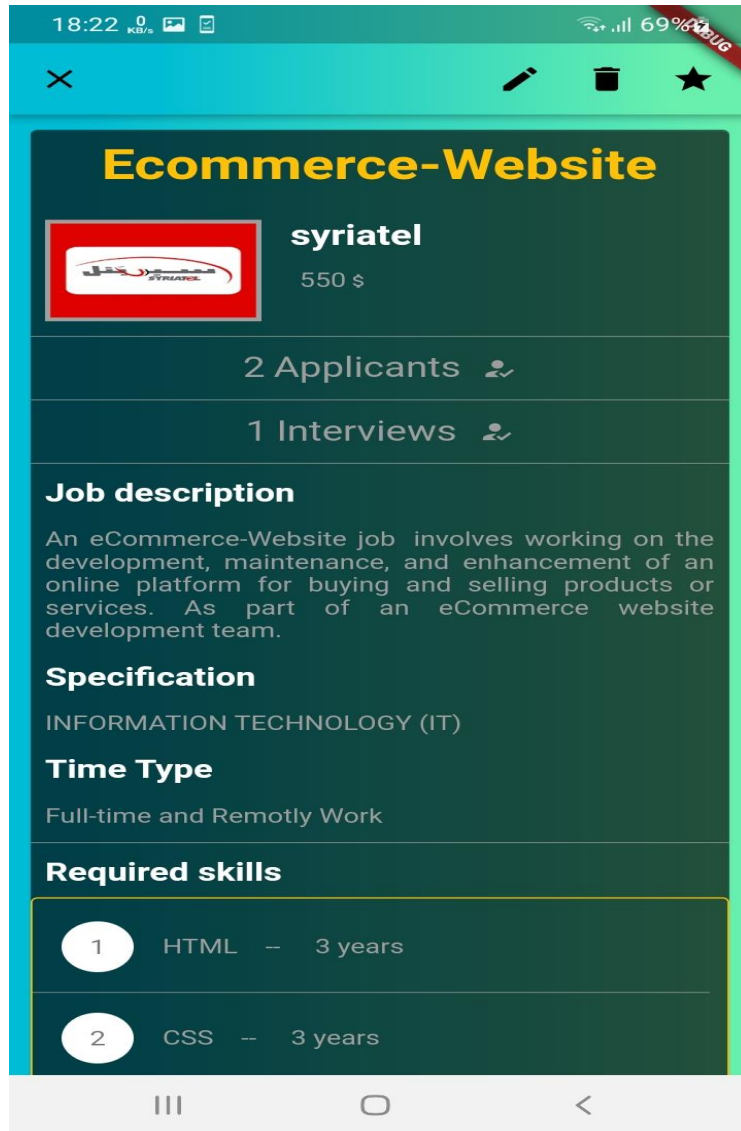
2.2.2. استعراض الأعمال المطروحة من قبل الشركة لإدارتها



الشكل التوضيحي 54: استعراض الأعمال المطروحة من قبل الشركة لإدارتها.

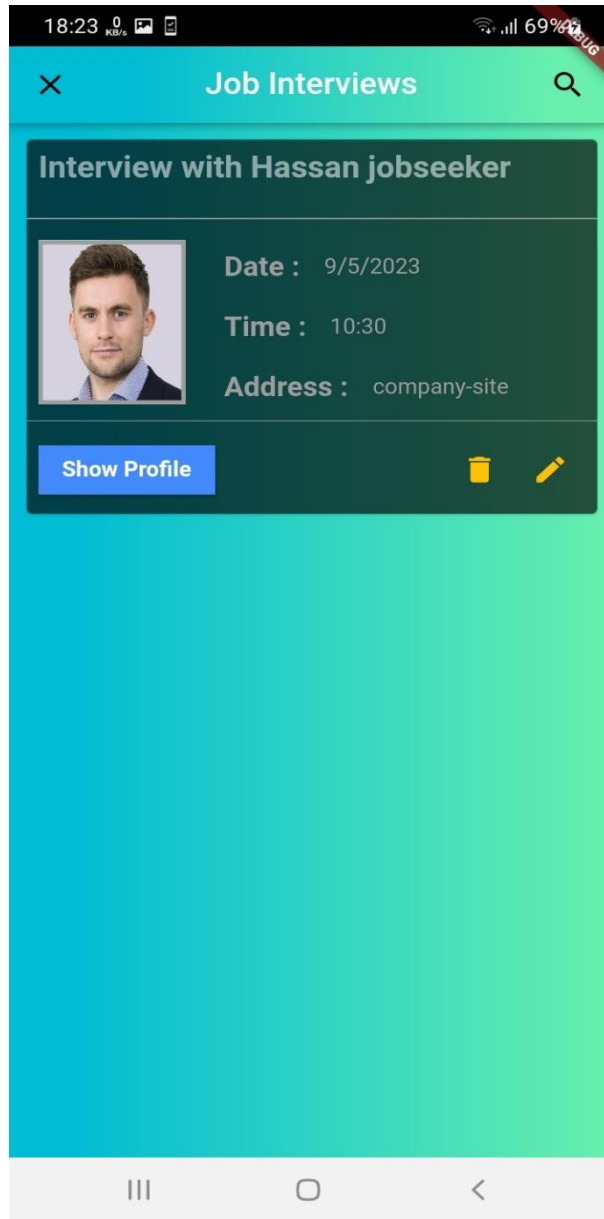
3.2.2. استعراض تفاصيل عمل مطروح من قبل الشركة

يتضمن استعراض تفاصيل عمل معيّن مطروح من قبل الشركة المالكة للحساب إظهار تفاصيل العمل بالإضافة إلى إظهار عدد المتقدمين على هذا العمل مع إمكانية استعراض ملفاتهم الشخصية أو البحث عن متقدّم معيّن، كما يتم إظهار عدد المقابلات المحجوزة لهذا العمل مع إمكانية إلغاء مقابلة أو تعديلها، بالإضافة إلى ذلك يمكن لممثّل الشركة حذف العمل من النظام، أو تعديل معلوماته حيث يمكن تعديل المهارات المطلوبة للعمل أو تعديل المعلومات الخاصة بالعمل التي تتضمن إمكانية إخفاء ظهور العمل عن باقي المستخدمين وفي هذه الحالة لا يظهر العمل إلا لممثّل الشركة.



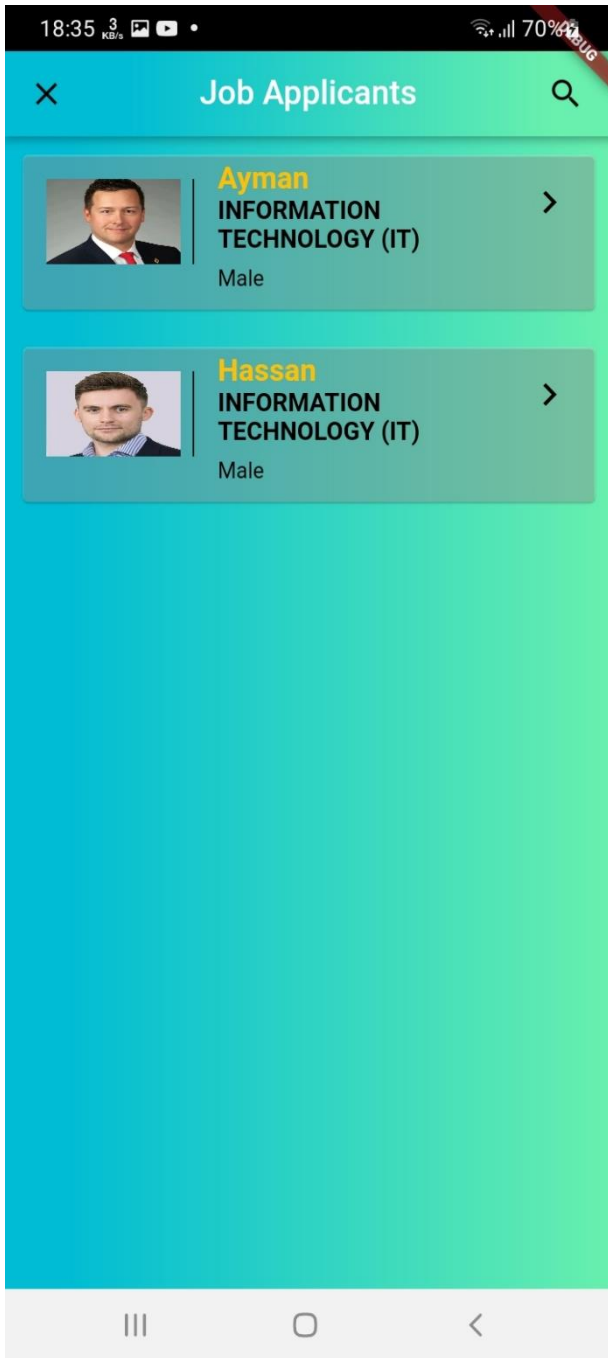
الشكل التوضيحي 55: استعراض تفاصيل عمل مطروح من قبل الشركة.

- استعراض جميع المقابلات المحجوزة لهذا العمل

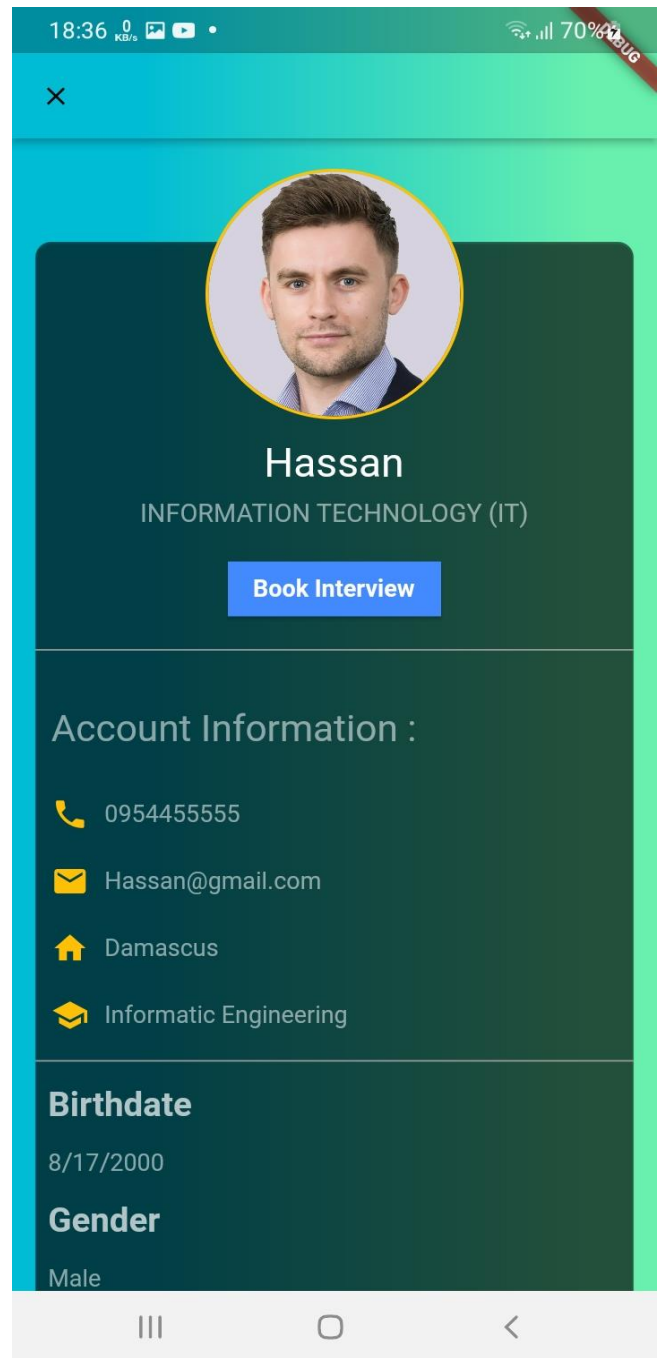


الشكل التوضيحي 56: استعراض المقابلات المحجوزة لعمل مطروح من قبل الشركة.

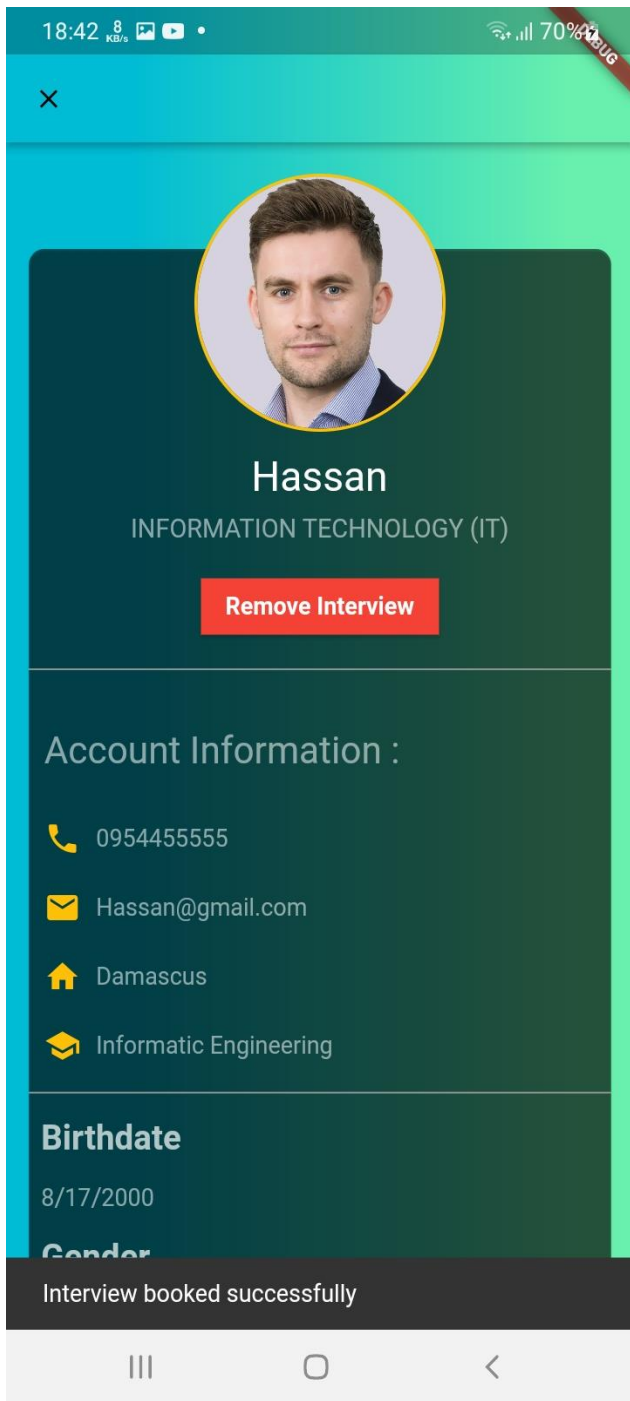
- استعراض جميع المتقدمين على هذا العمل



الشكل التوضيحي 58: استعراض جميع المتقدمين على عمل مطروح من قبل الشركة.




الشكل التوضيحي 57: استعراض الملف الشخصي لأحد المتقدمين.



18:42 8 KB/s 70% 4G


×





Hassan
INFORMATION TECHNOLOGY (IT)


Remove Interview

Account Information :

 0954455555

 Hassan@gmail.com

 Damascus

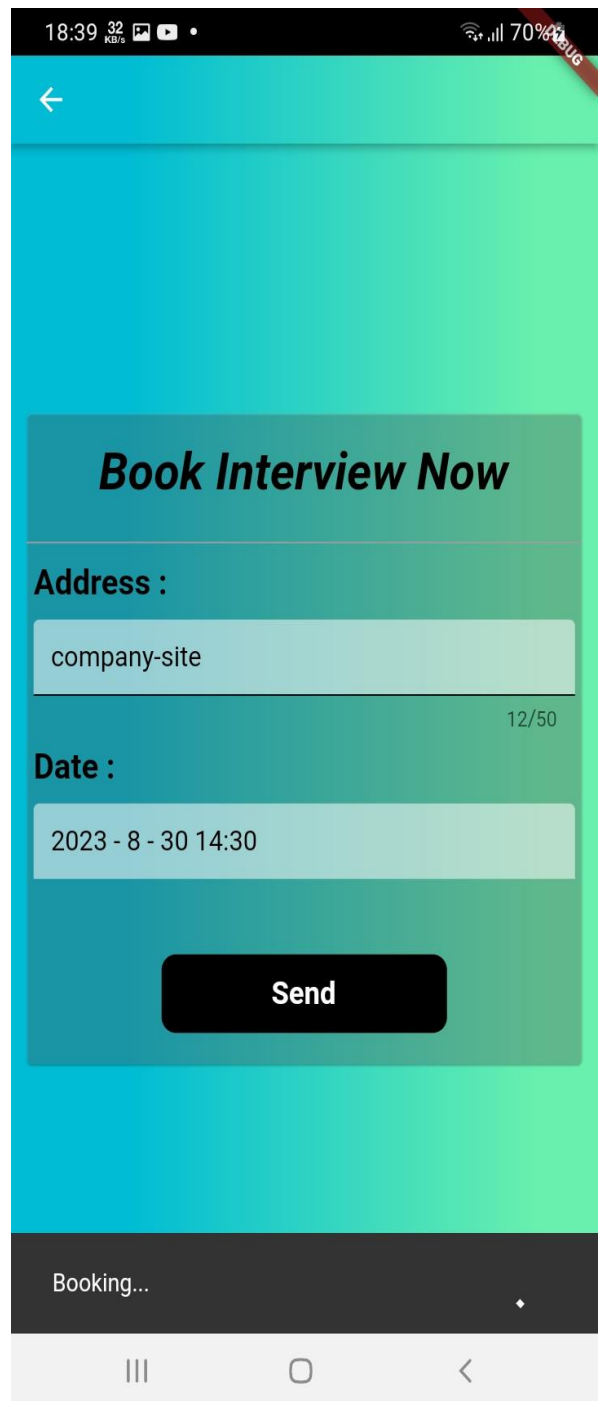
 Informatic Engineering

Birthdate
8/17/2000

Gender

Interview booked successfully

الشكل التوضيحي 60: حجز مقابلة لطالب وظيفة.



18:39 32 KB/s 70% 4G

←

Book Interview Now

Address :

company-site

12/50

Date :

2023 - 8 - 30 14:30

Send

Booking...

الشكل التوضيحي 59: حجز مقابلة لطالب وظيفة.

4.2.2. إضافة عمل جديد

Upload New Job

Job name :

0/50

Salary :

enter salary in dollar \$

0/50

Time type :

Specification name :

Availability :

Description :

الشكل التوضيحي 61: إضافة عمل جديد من قبل ممثل شركة.

- تحديد نوع الدوام المطلوب والاختصاص الذي ينتمي إليه العمل.

18:49 0 KB/s 71%

Salary :

enter salary in dollar \$

Time type :

Search for specifications....

Specification name :

- INFORMATION TECHNOLOGY (IT)
- GRAPHIC DESIGN
- SALES

Availability :

- HUMAN RESOURCES (HR)
- OPERATING SYSTEMS

Description :

Add Cancel

Upload

الشكل التوضيحي 63: تحديد الاختصاص الذي ينتمي إليه العمل المضاف.

18:49 1 KB/s 71%

Job name :

Salary :

enter salary in dollar \$

Time type :

→ Part-time and On-site Work

→ Part-time and Remotly Work

Specification name :

→ Full-time and On-site Work

→ Full-time and Remotly Work

Availability :

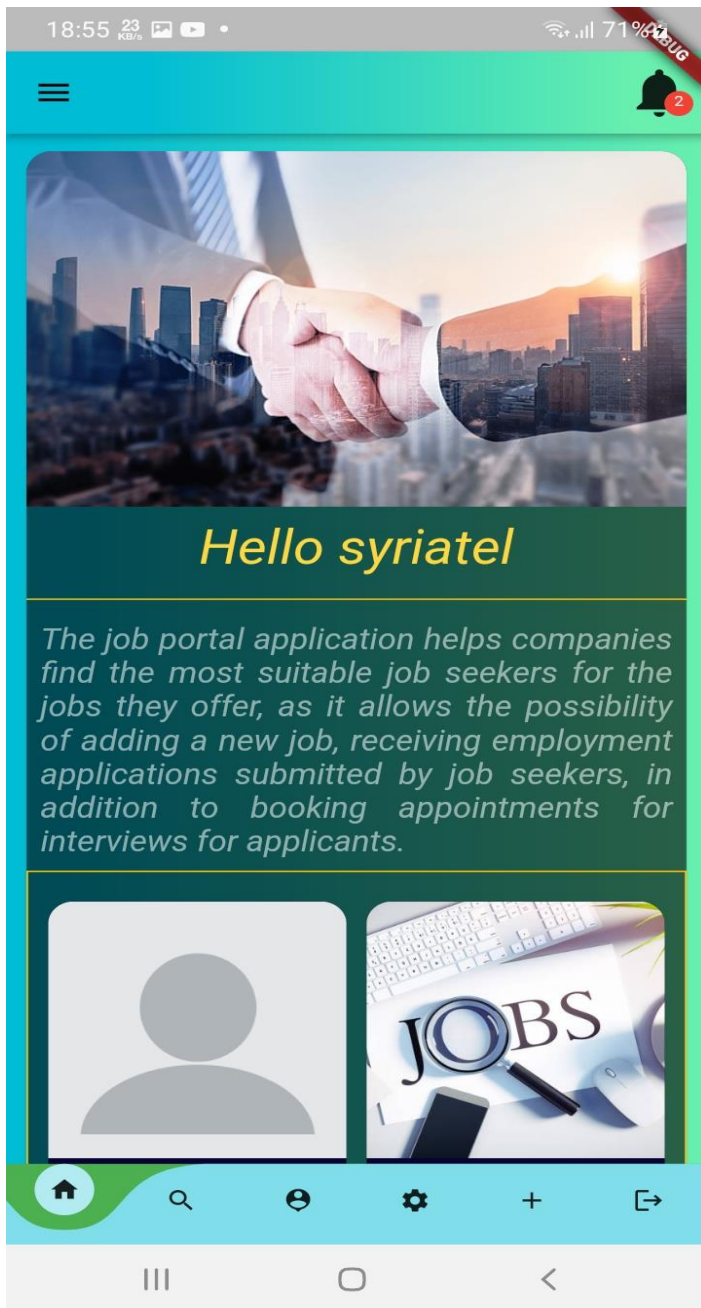
Cancel

Description :

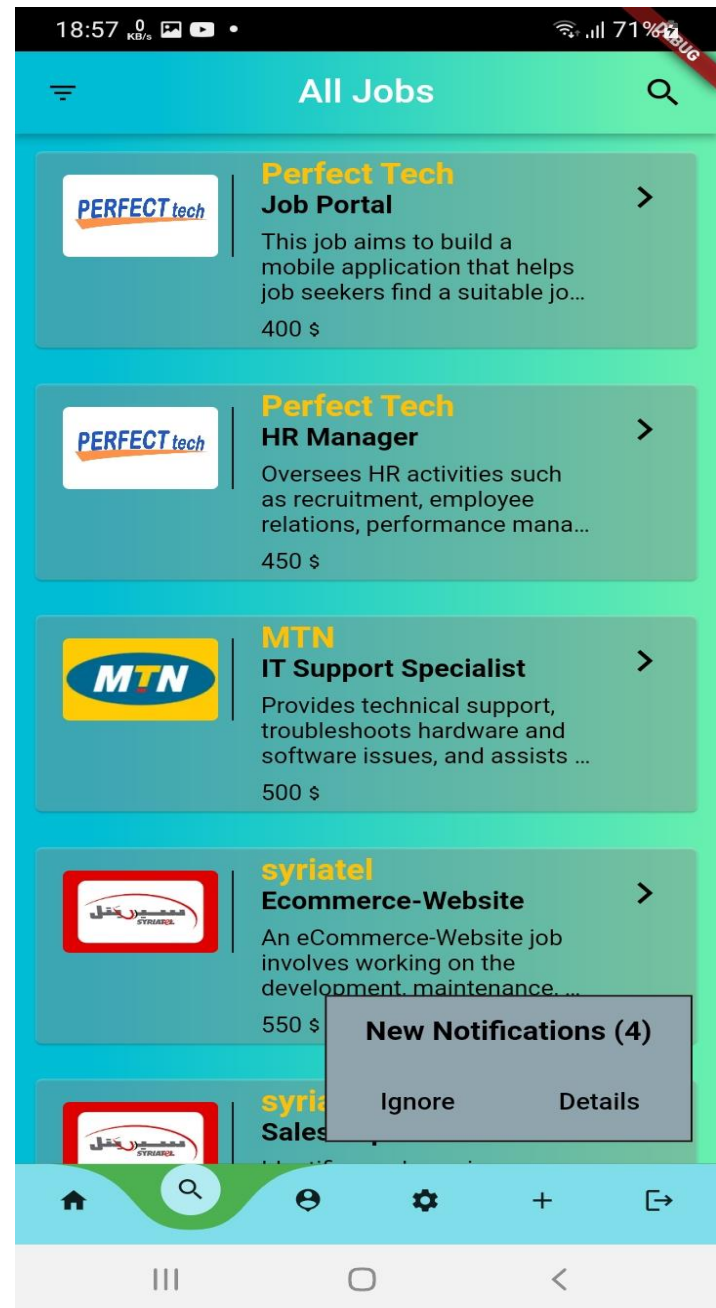
الشكل التوضيحي 62: تحديد نوع الدوام المطلوب للعمل المضاف.

3.2. الإشعارات في التطبيق

يقوم النظام بإرسال إشعارات للمستخدمين لإعلامهم بالأحداث المهمة، حيث يرسل إشعار لشركة عندما يقوم طالب وظيفة بتقديم طلب توظيف على عمل مطروح من قبل هذه الشركة، ويرسل إشعار لطالب وظيفة عندما يتم حجز أو تعديل أو إلغاء موعد مقابلة له. ويتلقى المستخدم هذا الإشعار مباشرة في حال كان يستخدم التطبيق أي يتلقى إشعار في الزمن الفعلي (real-time)، وأما إذا كان لا يستخدم التطبيق لحظة إرسال الإشعار له فإنه عندما يقوم بتسجيل دخول ستظهر له الإشعارات الجديدة ضمن الصفحة الرئيسية، وفي كلتا الحالتين يمكن للمستخدم أن يستعرض الإشعارات الجديدة ويبحث عن إشعار معين بحسب اسم المرسل.



الشكل التوضيحي 65: إستملا إشعار عند تسجيل الدخول إلى التطبيق.



الشكل التوضيحي 64: إستملا إشعار بالزمن الفعلي.

الفصل السابع

الخاتمة والأفاق المستقبلية

1. مقدّمة

في الحياة الواقعية، يواجه الأفراد والمنظمات دائماً التحديات والصعاب أثناء مواجهتهم للعمل وتنفيذ مشاريعهم. تلك التحديات قد تكون متعلقة بالبيئة التي يعملون فيها، مثل قيود الموارد والضغط الزمنية. قد تنشأ أيضاً مشاكل تقنية أو تنظيمية تعوق تقدم العمل وتؤثر على النتائج المرجوة. في هذا الفصل، سنتناول بعض العقبات والمشاكل التي واجهناها في بيئات العمل المختلفة. سنقدم نظرة عامة على هذه التحديات والحلول التي تم اتخاذها للتغلب عليها. بالإضافة إلى ذلك، سنناقش أيضاً الآفاق المستقبلية لمشروعنا وكيفية مواجهة التحديات المحتملة في المستقبل.

2. المشاكل والتحديات

- لكي يتم التواصل بين الـ `front-end` والمخدّم `back-end` يجب أن يتم إنشاء اتصال بين الطرفين، هذا الاتصال ينشئه الـ `front-end`، حيث يتنصّت المخدّم على بوابة معينة وينتظر ورود طلبات على هذه البوابة، لذلك يقوم الـ `front-end` بإنشاء اتصال مع المخدّم على البوابة التي يتنصّت عليها المخدّم. أي ما يلزمنا هو إنشاء اتصال مع المخدّم من خلال عنوان IP للمخدّم بالإضافة إلى البوابة التي يتنصّت عليها هذا المخدّم. وهذا ما كنا نقوم به من طرف الـ `front-end` حيث أنه في حالتنا عنوان المخدّم هو `localhost` والبوابة التي يتنصّت عليها هي `5158`. لذلك كنّا نطلب العنوان `"http://localhost:5158/"` والمشكلة التي واجهتنا هي عدم قدرة بيئة العمل `Flutter` على الاتصال مع المخدّم المحلي `localhost`. لذلك كان الحل لهذه المشكلة هو استخدام الأداة `ngrok` التي تؤمن عملية تغيير العنوان `"http://localhost:5158/"` إلى عنوان آخر عبر الإنترنت يمكن الاتصال به.

- عدد كبير من المكتبات يعمل في بيئة عمل `Flutter` ولكل منها إصدارات عديدة، وبالتالي مسألة اختيار مكتبة أو حزمة ضمن إصدار معين يتوافق مع إصدار `Flutter` المستخدم كانت تتطلب الكثير من الوقت.
- الحالات التي كانت تتطلب إرسال حجم كبير من المعطيات إلى المخدّم أو استقبال منه كمّ كبيرٍ من المعطيات كانت تستهلك الكثير من الوقت وتجعل أداء التطبيق متردّياً. لذلك كان الحل لهذه المشكلة القيام بعدّة إجراءات أهمها:

- ضغط الصور عند إرسالها إلى المخدّم مما يقلل من حجمها، وبالتالي يزيد سرعة استقبال هذه الصور عند طلبها من المخدّم.
- العمل على عرض المعلومات التي يتم الحصول عليها فور وصولها، وعدم انتظار استلام البيانات بالكامل، حتى لا يتأخر عرض محتوى الصفحة.

3. الآفاق المستقبلية

- إضافة نظام محادثات على التطبيق ليتمكن المستخدمون من التواصل مع بعضهم من خلال التطبيق، وبدون الحاجة لاستخدام موارد الجهاز الأخرى مثل الإيميل والمكالمات الهاتفية وغيرها.
- إضافة قسم خاص بتقديم دورات تدريبية للمتقدمين لتمكينهم من اكتساب خبرات جديدة.
- إضافة قسم خاص بتقديم توصيات واقتراحات لطلابين الوظيفة يقدم لهم أفضل الوظائف المناسبة لهم، وكذلك يقدم للشركات توصيات واقتراحات عن أفضل المتقدمين المناسبين للأعمال التي يطلبونها.

4. الخاتمة

تم إنجاز هذا المشروع بالكامل وتحقيق جميع المتطلبات الوظيفية وغير الوظيفية المطلوبة، حيث تم بناء تطبيق موبايل يساعد الباحثين عن عمل على استعراض الأعمال المطروحة من قبل الشركات الأخرى، والبحث عن عمل محدد وتقديم طلبات توظيف على الأعمال المطروحة، وتتلقى الشركات هذه الطلبات وتختار أفضل المتقدمين للأعمال التي تطرحها، وتقوم بحجز مواعيد مقابلات لهم. تم استخدام العديد من البيئات والأدوات البرمجية ضمن المشروع وأهمها Flutter و ASP.NET Core و SQL Server والعديد من الأدوات الأخرى. وتم بناء التطبيق بطريقة هندسية تعتمد على العديد من الأنماط التصميمية والمفاهيم المهمة، ولا سيما بنية API. واجهنا خلال المشروع العديد من المشاكل وقمنا بحلها بأفضل الوسائل الممكنة للحصول على منتج جيد يحقق المتطلبات المطلوبة.

المراجع

- [1] flutter. [Online]. Available: <https://flutter.dev/>.
- [2] dart. [Online]. Available: <https://dart.dev/>.
- [3] Wu, Wenhao. "React Native vs Flutter, Cross-platforms mobile application frameworks. (2018).
- [4] ASP.NET Core documentation. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/>.
- [5] Richardson, Leonard, Mike Amundsen, and Sam Ruby. *RESTful Web APIs: Services for a Changing World*. " O'Reilly Media, Inc.", 2013.
- [6] SQL Server documentation. [Online]. Available: <https://docs.microsoft.com/en-us/sql/sql-server>.
- [7] Visual Studio Code documentation. [Online]. Available: <https://code.visualstudio.com/docs>.
- [8] Visual Studio 2022. [Online]. Available: <https://learn.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2022>.
- [9] Android Studio. [Online]. Available: <https://developer.android.com/studio/intro>.
- [10] Swagger. [Online]. Available: <https://swagger.io/>.
- [11] Postman. [Online]. Available: <https://www.postman.com/>.
- [12] Glassdoor. [Online]. Available: <https://www.glassdoor.com/>.
- [13] Monster. [Online]. Available: <https://www.monster.com/>.
- [14] Ngrok. [Online]. Available: <https://ngrok.com/>.