

جدول المحتويات

1	مقدمة.....	3
1.1	تعريف المشروع.....	3
1.2	تعريف المشكلة.....	4
1.3	الأهداف والغايات.....	5
1.3.1	الهدف العام:.....	5
1.3.2	الأهداف التفصيلية:.....	5
1.3.3	الرؤية المستقبلية والأهداف الإضافية:.....	6
1.4	نطاق المشروع.....	6
1.4.1	الفئة المستهدفة:.....	6
1.4.2	الخدمات الأساسية:.....	7
1.4.3	المزايا التقنية:.....	8
2	خلفية المشروع والمبررات.....	10
2.1	خلفية المشكلة وتحليلها.....	10
2.1.1	نظرة عامة على النقل الجماعي بين المدن.....	10
2.1.2	تطور النقل الجماعي.....	10
2.1.3	مراحل تطور النقل الجماعي:.....	11
2.1.4	تحديات النقل الجماعي التقليدي.....	11
2.1.5	النمو في الطلب على النقل الجماعي.....	12
2.1.6	أهمية التحول الرقمي.....	13
2.1.7	تأثير غياب الرقمنة.....	14
2.1.8	الحاجة إلى الحل الرقمي.....	14
2.2	أصحاب المصلحة.....	16
2.2.1	تحديد أصحاب المصلحة.....	16
2.3	نمذجة مشروع BUS E-Ticket Platform.....	17
2.3.1	كيف يطبق مشروع BUS E-Ticket نموذج SaaS؟.....	17
2.3.2	كيف يطبق مشروع BUS E-Ticket نموذج B2B2C؟.....	18

20.....	2.4 مشاريع مشابهة.....
20.....	2.4.1 المشاريع المشابهة:
21.....	2.4.2 جدول مقارنة المشاريع المشابهة.....
22.....	2.4.3 استنتاج.....
22.....	2.5 مبررات المشروع.....
22.....	2.5.1 الحاجة إلى التحول الرقمي في قطاع النقل الجماعي.....
22.....	2.5.2 دعم التكامل بين مزودي الخدمة
22.....	2.5.3 تحسين تجربة العملاء.....
22.....	2.5.4 القيمة المضافة.....
22.....	2.5.5 الفرصة الاقتصادية والتوسع المستقبلي.....
23.....	3 التصميم والتحليل.....
23.....	3.1 تحليل المتطلبات.....
23.....	3.1.1 المتطلبات الوظيفية.....
24.....	3.1.2 المتطلبات غير الوظيفية.....
26.....	3.1.3 مخطط الحالة.....
28.....	3.2 معمارية النظام.....
28.....	3.2.1 معمارية البرمجيات.....
30.....	3.2.2 معمارية هندسة البرمجيات المستخدمة في منصة BUS E-TICKET Platform.....
31.....	3.3 تصاميم النظام.....
31.....	3.3.1 UML Class Diagram (مخطط الكلاسات):.....
33.....	3.3.2 Package Diagram (مخطط الحزم):.....
34.....	3.3.3 Schema Diagram (مخطط قاعدة البيانات):.....
35.....	4 المنهجية.....
35.....	4.1 SQL Database.....
35.....	4.2 الواجهة الخلفية Back-End.....
35.....	4.2.1 اللغات وأطر العمل.....
36.....	4.2.2 Entity Framework Core (EF Core):.....
36.....	4.2.3 Repository Pattern :.....

36.....	Unit of Work Pattern4.2.4
36.....	4.2.5 خدمات الهوية والمصادقة (Authentication & Authorization)
37.....	4.2.6 طبقة الأعمال (Business Logic Layer)
37.....	4.2.7 كلاسات الخدمات (Service Classes)
37.....	4.2.8 الواجهات (Interfaces)
37.....	4.2.9 طبقة Presenter Layer
38.....	4.3 الواجهة الامامية Front-End
39.....	5 التطبيق
39.....	6 النتائج والتوصيات
39.....	7 اضافات

1 مقدمة

منذ فجر التاريخ، سعى الإنسان إلى تقليل الزمن اللازم للتنقل بين الأماكن، مما أدى إلى تطوير وسائل نقل متقدمة بدءًا من استخدام الحيوانات وحتى السيارات الحديثة. ومع زيادة الاعتماد على السيارات الشخصية، ظهرت تحديات متعلقة بارتفاع التكاليف وأسعار الوقود، مما دفع الدول إلى تعزيز أنظمة النقل الجماعي كحل مستدام وفعال، فظهرت خدمات النقل الجماعي العامة. وفي العقد الأخير، أصبحت الرقمنة عنصرًا حيويًا في تطوير خدمات النقل العام، حيث قامت العديد من الدول والشركات ببنّي أنظمة تقنية لإدارة العمليات وتوفير خدمات الاستعلام وشراء التذاكر عن بُعد لما تضيفه الرقمنة من تعزيز للاقتصاد والتنمية[1]. مشروعنا يأتي استجابة لهذه التحديات، حيث يهدف إلى تقديم حل تقني مبتكر لتحسين تجربة النقل الجماعي ما بين المدن من خلال تقديم نظام رقمي متكامل يخدم الشركات والعملاء على حد سواء.

1.1 تعريف المشروع

مشروع BUS E-TICKET Platform هو منصة تقديم خدمة سحابية متكاملة تستهدف العملاء ومقدمي خدمات النقل الجماعي بين المدن. تعمل المنصة على توفير حلول رقمية متقدمة لدعم مزودي الخدمة الذين يفتقرون إلى البنية الرقمية[2]. كما تقدم واجهة مخصصة للعملاء تتيح لهم اختيار الرحلات المناسبة، حجز التذاكر، وإدارة الحجوزات بسهولة وكفاءة[3]. (سيكون هناك تعريف خاص لمن هم العملاء وموفرين الخدمة في قسم الفئة المستهدفة من قسم نطاق المشروع)

1.2 تعريف المشكلة

قطاع النقل الجماعي بين المدن يواجه العديد من التحديات التي تعيق نموه وكفاءته، أبرزها غياب البنية الرقمية لدى العديد من الشركات والمشغلين الأفراد، ما يؤدي إلى ضعف الكفاءة وصعوبة الوصول للعملاء. تواجه الشركات الناشئة تحديات كبيرة في التوسع بسبب ارتفاع التكاليف وضعف الكفاءة التشغيلية، بينما تفتقر الشركات الصغيرة والمشغلون إلى المعرفة والخبرة اللازمة للتحويل الرقمي. حتى عند إدراك أهمية الرقمنة، قد لا تستطيع بعض الشركات تحمل التكاليف المرتفعة لبناء أنظمة رقمية متكاملة. إضافة إلى ذلك، يفتقر القطاع إلى التكامل بين الشركات، ما يقلل من الخيارات المتاحة للعملاء ويجعل عمليات الحجز والدفع معقدة وغير مريحة. كما أن غياب الحلول الرقمية يؤدي إلى ضعف إدارة الرحلات، المدفوعات، وبيانات العملاء. منصة Bus E-Ticket تقدم حلاً رقمياً شاملاً لمعالجة هذه التحديات، حيث توفر واجهة موحدة تتيح للعملاء البحث عن الرحلات، الحجز، وإدارة التذاكر بسهولة، بينما تقدم لمزودي الخدمة أدوات متقدمة لإدارة العمليات والمدفوعات بكفاءة، وتقارير شفافة لتحسين الأداء. المنصة تتيح للشركات التوسع بسهولة بفضل بنيتها الرقمية الجاهزة، ما يساهم في تقليل التكاليف التشغيلية وتحسين تجربة المستخدم بشكل ملحوظ.

المشكلة	الحل المقدم من Bus E-Ticket
افتقار مزودي الخدمة إلى البنية الرقمية	توفر المنصة بنية تحتية رقمية جاهزة يمكن لمزودي الخدمة استخدامها بسهولة دون الحاجة لمعرفة تقنية متقدمة.
ظهور الشركات الناشئة	تقدم المنصة حلول SaaS منخفضة التكلفة، مما يساعد الشركات الناشئة على استخدام التكنولوجيا بكفاءة وتوسيع نطاقها.
شركات لا تعلم كيف تبدأ	توفر المنصة واجهة سهلة الاستخدام ودليل شامل يساعد الشركات الصغيرة والمشغلين الأفراد على بدء التحويل الرقمي.
شركات لا تستطيع تحمل التكلفة	تعتمد المنصة نموذج SaaS بمرونة دفع شهرية أو سنوية، مما يجعل التكلفة منخفضة مقارنة ببناء أنظمة مستقلة.
غياب التكامل بين مزودي الخدمة	توفر المنصة واجهة موحدة تجمع بين مختلف مقدمي الخدمة، مما يتيح للعملاء الاطلاع على جميع الخيارات المتاحة بسهولة.
تعقيد عمليات الحجز والدفع	تقدم المنصة نظام حجز ودفع إلكتروني يسهل استخدامه، ما يتيح تجربة حجز سلسلة وأمنة للعملاء.
غياب البنية التحتية لرقمنة العمليات	توفر المنصة أدوات لإدارة الرحلات، المدفوعات، والبيانات، ما يحسن الكفاءة ويوفر تقارير دقيقة لمزودي الخدمة

1.3 الأهداف والغايات

1.3.1 الهدف العام:

تطوير منصة رقمية شاملة ومتكاملة لتقديم حلول تقنية مبتكرة تخدم عملاء النقل الجماعي بين المدن ومزودي الخدمة، مع تحقيق تجربة استخدام مريحة ودعم رقمنة العمليات التشغيلية بأسلوب مستدام وفعال.

1.3.2 الأهداف التفصيلية:

1.3.2.1 تحقيق الرقمنة:

- إنشاء بنية تحتية رقمية لمزودي الخدمة الذين يفتقرون إلى الأنظمة الرقمية.
- توفير أدوات لإدارة البيانات، المدفوعات، والحجوزات.
- توحيد عرض الرحلات من قسم موفري الخدمة إلى قسم العملاء.

1.3.2.2 تعزيز التكامل:

- دمج خدمات مزودي الخدمة في منصة موحدة.
- عرض جميع الرحلات المتاحة في واجهة موحدة لتوفير خيارات متعددة للعملاء.

1.3.2.3 تحسين تجربة العملاء:

- تسهيل عملية البحث عن الرحلات باستخدام أدوات فلاتر دقيقة وسهلة.
- تمكين العملاء من إجراء الحجوزات والدفع الإلكتروني بشكل آمن.
- إتاحة الوصول إلى التذاكر والفواتير الإلكترونية والاستعلام عن الرحلات السابقة.
- تقديم تجربة تسجيل دخول مبسطة وسهلة الاستخدام.

1.3.2.4 دعم مزودي الخدمة:

- تقديم لوحة تحكم متكاملة لإدارة المركبات، السائقين، والرحلات.
- تزويد مزودي الخدمة بتقارير احترافية لتحسين الكفاءة التشغيلية.

1.3.2.5 زيادة المبيعات:

- تمكين مزودي الخدمة من الوصول إلى قاعدة عملاء أوسع.
- إضافة ميزات مثل الحملات التسويقية وإنشاء كوبونات خصم.

1.3.2.6 تعزيز الأمان:

- تطبيق خوارزميات تشفير متقدمة لحماية بيانات العملاء.
- تعزيز مصداقية العمليات من خلال إجراءات مصادقة قوية.
- تحسين الأمان على مستوى الكود والتطبيق باستخدام أحدث التقنيات.

1.3.2.7 ضمان التوسع المستقبلي:

- تصميم المنصة بمعمارية مرنة تتيح التوسع وإضافة خدمات جديدة.
- ضمان استيعاب المزيد من المستخدمين دون التأثير على الأداء.

1.3.2.8 تحقيق إدارة المنصة

- توفير حسابات لمدراء المنصة
- حوكمة تسجيل موفري الخدمة عبر بروتوكولات محددة وبعد موافقة ومراجعة المدراء
- الوصول لمعلومات العملاء بشكل محدود ومحافظ من قبل موفري الخدمة لزيادة جودة وكفاءة الإدارة

1.3.3 الرؤية المستقبلية والأهداف الإضافية:

- استهداف مشغلي الأفراد بسياسات مناسبة لدعمهم.
- تحقيق تكامل نظام العملاء مع مزودي الخدمة الخارجيين الذين يمتلكون بنية تحتية رقمية ولكن يفتقرون للتكامل مع منصة موحدة.
- تضمين للوحة التحكم واجهة مدعومة بأنظمة ذكاء الأعمال (BI) او ربط .
- ضمان استدامة المنصة من خلال سياسات ربح لتغطي التكاليف التشغيلية.
- الانتقال الى المعماريات البرمجية الأكثر حداثة وتعقيداً مثل Microservice لزيادة الأداء والتجريد بين مكونات الأنظمة.

1.4 نطاق المشروع

1.4.1 الفئة المستهدفة:

1.4.1.1 موفري الخدمة

في هذا الإصدار، سيركز مشروع Bus E-Ticket على دعم الشركات الخاصة التي تقدم خدمات النقل الجماعي بين المدن، والتي تفتقر إلى البنية التحتية الرقمية أو التكامل مع أي منصة تعرض خدماتها. أما المشغلون الأفراد أو الشركات التي تمتلك بنية تحتية رقمية خارج المنصة، فتم تضمينهم ضمن الأهداف المستقبلية، كما هو موضح في قسم الأهداف. لذلك، لا تُعد هذه الفئات ضمن الفئة المستهدفة للإصدار الأول من المشروع.

1.4.1.2 العملاء

العملاء هم الأفراد المستفيدون من خدمات مزودي النقل الجماعي العام بين المدن. يمكن تصنيفهم إلى فئات متعددة بناءً على معايير مختلفة، مثل:

- الجنس: ذكور وإناث.

- الفئة الوظيفية: موظفون وطلاب.
- الغرض من الاستخدام: سياح وزوار أو مقيمون محليون.

هذا التنوع في تصنيفات العملاء يعكس الحاجة إلى تقديم خدمات مرنة وشاملة تلبي احتياجات مختلف الفئات.

1.4.2 الخدمات الأساسية:

1.4.2.1 خدمات المنصة لمزودي الخدمة:

- التسجيل الرسمي والممنهج: تقديم طلب تسجيل يتم مراجعته والموافقة عليه قبل إنشاء الحساب.
- إدارة معلومات الحساب: تعديل بيانات الحساب مثل الشعار، الاسم، كلمة المرور، وغيرها.
- لوحة تحكم سهلة الاستخدام: واجهة مرنة لإدارة العمليات اليومية.
- إدارة الرحلات: إنشاء وتعديل الرحلات وجدولتها بسهولة.
- إدارة الحجوزات: مراقبة ومعالجة الحجوزات المقدمة من العملاء.
- الوصول إلى بيانات المسافرين والعملاء: استعراض بيانات الحجوزات وأسماء العملاء المرتبطين بها.
- إحصائيات عامة: عرض ملخصات تحليلية لأنشطة الشركة مثل المبيعات والحجوزات.

1.4.2.2 خدمات المنصة للعملاء:

- تسجيل حساب جديد: إنشاء حساب شخصي للاستفادة من خدمات المنصة.
- إدارة معلومات الحساب: تحديث البيانات الشخصية مثل الاسم، البريد الإلكتروني، وكلمة المرور.
- البحث عن الرحلات: استعلام عن الرحلات بناءً على المعايير المحددة.
- حجز الرحلات: إمكانية الحجز سواء للحساب الشخصي أو لصالح عملاء آخرين غير مسجلين.
- الوصول إلى مواقع الانطلاق والوصول: تحديد المواقع على الخريطة أو عبر الإرشادات.
- تتبع الحجز والرحلة: مراقبة حالة الحجز وتفاصيل الرحلة.
- استعلام عن الرحلات السابقة والمدفوعات: الوصول إلى سجل الرحلات والمدفوعات السابقة.
- الوصول إلى الفواتير والتذاكر: تحميل الفواتير والتذاكر الإلكترونية.

1.4.2.3 خدمات المنصة للمدراء:

- إدارة جميع الحسابات: التحكم الكامل في حسابات مزودي الخدمة والعملاء.
- استعلام بيانات العملاء غير المسجلين: الوصول إلى معلومات العملاء غير المسجلين المرتبطة بالحجوزات.
- إحصائيات شاملة لأنشطة المنصة: عرض تقارير عامة تشمل أداء المنصة بالكامل.
- إدارة طلبات التسجيل: مراجعة واعتماد طلبات تسجيل مزودي الخدمة الجدد.
- استعلام شامل عن الرحلات والحجوزات: الاطلاع على جميع الرحلات المقدمة من مزودي الخدمة وجميع حجوزات العملاء.

1.4.2.4 النطاق الجغرافي:

ستركز منصة Bus E-Ticket في إصدارها الأول على السوق السعودي نظراً للاحتياج الكبير لهذا النوع من الخدمات الرقمية والتحول الرقمي في النقل الجماعي. مع ذلك، تم تصميم المنصة لتكون عالمية في المستقبل وللتوسع التدريجي إلى أسواق أخرى بناءً على النجاح المحلي.

1.4.3 المزايا التقنية:

1.4.3.1 التجريد والتكامل

تتميز المنصة بهيكلية تقنية تعتمد على التجريد العالي بين قسم العملاء وقسم مزودي الخدمة.

- قسم مزودي الخدمة: يعمل كنظام مستقل يتيح لمزودي الخدمة إدارة أنشطتهم مثل إدارة الرحلات، الحجوزات، والمعلومات التشغيلية بسهولة وكفاءة.
- قسم العملاء: مصمم بشكل منفصل تمامًا وتتجمع خدمات موفري الخدمة (الرحلات... الخ) كلها من قسم مزودي الخدمة وتعرض على العملاء.
- هذا التجريد يضمن استقلالية ومرونة كل قسم مع إمكانية التكامل السلس بينهما.

1.4.3.2 الأمان

تبنى المنصة أحدث التقنيات لتعزيز الأمان من خلال:

- تصميم كود آمن باستخدام مبادئ البرمجة الكائنية (OOP) لضمان بنية كودية قوية.
- تشفير البيانات بتطبيق خوارزميات Hash لتشفير كلمات المرور وحماية البيانات الشخصية.
- استعمال مكتبات حديثة في تعزيز المصادقة مثل Identity on ASP.NET Core

ملاحظة : سيتم تناول دور SOLID و OOP في تعزيز الأمان في قسم المنهجية.

1.4.3.3 قابلية التوسع

تم تصميم المنصة لتكون قابلة للتوسع بسهولة مع الإصدارات المستقبلية.

- بناء الواجهة الخلفية بمعمارية حديثة تستند إلى مفاهيم مثل Design Patterns و SOLID Principles.
- الانتقال من المعمارية التقليدية 3-Tier Architecture إلى N-Layered Architecture، مما يتيح تقسيم النظام إلى طبقات أكثر تفاعلية وديناميكية.
- تعزيز القابلية للتوسع باستخدام مبادئ Clean Architecture لضمان سهولة إضافة الميزات المستقبلية دون التأثير على النظام الأساسي.
- استعمال معمارية السيرفر والعمي. هذا الإصدار تم التركيز على طبقة السيرفر بشكل منفصل وربطه مع واجهة العميل موقع الكتروني مدعوم بتقنية ال PWA ليعمل كتطبيق للجوالات.

ملاحظة: سيتم شرح المعماريات المستخدمة وتقنية ال PWA بشكل تفصيلي في قسم المنهجية.

1.4.3.4 // الكفاءة

تم تصميم المنصة لضمان أعلى مستويات الكفاءة في الأداء من حيث:

- تسريع الاستجابة: تحسين استعلامات قواعد البيانات باستخدام T-SQL واستغلال تقنيات متقدمة مثل Entity Framework لتقليل زمن الوصول (Latency).
- واجهة مستخدم فعالة: بناء واجهة المستخدم باستخدام تقنيات مثل SPA (Single Page Application) لضمان تجربة سلسة وسريعة. أداء عالي: اعتماد بنية خلفية تدعم التوازي في العمليات وتعزز الكفاءة باستخدام .NET Core.
- اختبارات أداء شاملة: ضمان استقرار النظام وسرعته من خلال إجراء اختبارات أداء مستمرة وتطبيق استراتيجيات.
- أداء متقدم على الجوال: استعمال تقنية ال PWA لتحسين تجربة المستخدم للمنصة على الجوال.

ملاحظة: سيتم مناقشة تقنيات تعزيز الكفاءة وأدوات الاختبار المستخدمة بمزيد من التفصيل في قسم المنهجية.

1.4.3.5 حدود النطاق:

في المرحلة الأولى، سيركز المشروع على استهداف موفري الخدمة الذين يفتقرون إلى البنية الرقمية لتلبية احتياجاتهم من خلال توفير حلول رقمية شاملة. أما الشركات التي تمتلك بنية تحتية رقمية جزئية أو كاملة فسيتم إدراجها ضمن الإصدارات المستقبلية للمشروع، مع خطط لتطوير حلول تكاملية تتيح دمج أنظمتها مع المنصة لتحسين الكفاءة وتعزيز التعاون.

2 خلفية المشروع والمبررات

2.1 خلفية المشكلة وتحليلها

2.1.1 نظرة عامة على النقل الجماعي بين المدن

النقل الجماعي بين المدن يُعدّ نظامًا حيويًا يهدف إلى تلبية احتياجات التنقل للمسافات الطويلة بين المناطق المختلفة. يشمل هذا النظام في الغالب الحافلات الجماعية التي تُشغلها شركات عامة أو خاصة، حيث يتم تصميم خطوط النقل لتغطية المسافات بين المدن الكبرى والمناطق الريفية [4].

تتمثل أهمية النقل الجماعي بين المدن في تقليل الاعتماد على السيارات الشخصية، مما يساعد في تقليل الازدحام المروري وانبعاثات الغازات الضارة بالبيئة. بالإضافة إلى ذلك، يُوفر النقل الجماعي وسيلة ميسورة التكلفة مقارنة بتكاليف امتلاك السيارات أو استئجارها، مما يجعله الخيار المثالي للأفراد من ذوي الدخل المتوسط أو المحدود [4].

يُستخدم النقل الجماعي بين المدن من قبل شرائح متنوعة من السكان، مثل العمال، الطلاب، المسافرين لأغراض العمل، والسياح. ويساهم في تسهيل الوصول إلى فرص العمل، التعليم، والخدمات الأساسية، خاصةً في المناطق التي قد تفتقر إلى وسائل نقل بديلة [4].

من الناحية التشغيلية، يتميز هذا النظام بالمرونة في إدارة الجداول الزمنية والخطوط لتلبية الطلب المتغير، كما يعتمد على البنية التحتية مثل المحطات المركزية ونقاط التوقف الموزعة في المواقع الإستراتيجية. يساهم هذا في تعزيز كفاءة العمليات وضمان سهولة الوصول للمستخدمين [4].

2.1.2 تطور النقل الجماعي

2.1.2.1 كيف بدأ النقل الجماعي بين المدن؟

النقل الجماعي بين المدن بدأ باستخدام الحيوانات والعربات البدائية كوسيلة للتنقل لمسافات طويلة. مع الثورة الصناعية في القرن التاسع عشر، ظهرت الحافلات التي تعمل بالفحم والبخار، مما أدى إلى تحسين كفاءة وراحة النقل الجماعي بين المدن [5]. لاحقاً، أصبحت الحافلات التي تعمل بمحركات الاحتراق الداخلي أساساً لتطوير النقل الجماعي، حيث ساهمت في زيادة سرعة وكفاءة النقل [4].

2.1.3 مراحل تطور النقل الجماعي:

بدأ النقل الجماعي باستخدام العربات التي تجرها الحيوانات لنقل الركاب والبضائع كمرحلة أولى [5]. ومع الثورة الصناعية، شهد القطاع استخدام الحافلات التي تعمل بالفحم والبخار، مما ساهم في تقليل أوقات السفر وزيادة الاعتمادية [4]. ومع تزايد الطلب، ظهرت شبكات نقل منظمة بجدول زمنية ومسارات متكاملة، إلى جانب إدخال مفهوم "النقل التبادلي" الذي يربط المدن والقرى بشكل فعال [6]. لاحقاً، ومع الثورة التكنولوجية، تم إدخال أنظمة الحجز الإلكتروني والتطبيقات الذكية وأنظمة تتبع GPS، مما أدى إلى تحسين الخدمة وتقليل فترات الانتظار [5][7]. وأسهمت هذه التقنيات في تسهيل عمليات الحجز وتتبع المسارات، ما عزز تجربة السفر [7]. بالإضافة إلى استخدام البيانات الضخمة والذكاء الاصطناعي لتحسين تخطيط وتشغيل الحافلات عبر التنبؤ بعدد الركاب وتوزيع الموارد بشكل أكثر كفاءة [4][7]

المرحلة	الوصف
المرحلة الأولى - النقل التقليدي	بدأت باستخدام العربات التي تجرها الحيوانات لنقل الركاب والبضائع.
المرحلة الثانية - الثورة الصناعية	استخدام الحافلات التي تعمل بالفحم والبخار، ما قلل وقت السفر وزاد الاعتمادية.
المرحلة الثالثة - التنظيم والتكامل	ظهرت شبكات نقل بجدول زمنية ومسارات متكاملة وربطت المدن والقرى بالنقل التبادلي.
المرحلة الرابعة - الثورة التكنولوجية	تم إدخال أنظمة الحجز الإلكتروني، التطبيقات الذكية، وتتبع GPS لتحسين الخدمة.
دور التكنولوجيا	الحجز الإلكتروني والتطبيقات: سهلت الحجز وتتبع المسارات والبيانات الضخمة والذكاء الاصطناعي: حسنت تخطيط وتشغيل الحافلات.

2.1.4 تحديات النقل الجماعي التقليدي

2.1.4.1 اعتماد الأنظمة اليدوية في الحجز النقل الجماعي التقليدي

يعتمد بشكل كبير على الأنظمة اليدوية لإدارة عمليات الحجز، مما يؤدي إلى مشاكل تتعلق بالبطء، الأخطاء البشرية، وصعوبة إدارة البيانات بشكل فعال. الأنظمة اليدوية تعيق قدرة الشركات على التعامل مع العدد المتزايد من الركاب وتقديم تجربة سلسلة للعملاء. استخدام أنظمة الحجز الإلكتروني وحلول البرمجيات كخدمة (SaaS) يمكن أن يساهم في تحسين الكفاءة وتقليل التكاليف التشغيلية [2][1].

2.1.4.2 صعوبة إدارة العمليات الداخلية للشركات

إدارة النشاط الداخلي في شركات النقل الجماعي التقليدية غالباً ما تكون معقدة بسبب غياب الأنظمة الرقمية التي تساهم في تتبع الأداء وتحسين الإنتاجية. غياب حلول تكامل العمليات يجعل إدارة الموارد البشرية والجدول الزمنية أكثر تعقيداً. يمكن لتطبيقات SaaS وأنظمة الإدارة الرقمية أن توفر أدوات لتحسين إدارة الموارد البشرية والعمليات التشغيلية، مما يؤدي إلى تقليل الوقت والجهد المبذول [3][2].

2.1.4.3 غياب التكامل بين الشركات

عدم وجود تكامل بين شركات النقل الجماعي يؤدي إلى تحديات في ربط الرحلات والخدمات، مما يؤثر سلباً على تجربة الركاب ويقلل من كفاءة النظام [7][6].

2.1.4.4 صعوبة الوصول إلى العملاء

النقل الجماعي التقليدي غالبًا ما يواجه تحديات في جذب العملاء وتلبية احتياجاتهم بسبب غياب استخدام التكنولوجيا الحديثة لتحليل البيانات وفهم سلوكيات المستخدمين. الأنظمة التقليدية تعتمد على الأساليب القديمة للتسويق وخدمة العملاء، مما يحد من قدرتها على جذب شرائح جديدة من الركاب. استخدام البيانات الضخمة والتطبيقات الذكية لتحليل سلوكيات العملاء يمكن أن يحسن من إمكانية الوصول إلى العملاء ويوفر خدمات مخصصة تلبي احتياجاتهم [4][1].

2.1.5 النمو في الطلب على النقل الجماعي

في سياق التركيز على الجغرافية السعودية، شهدت المملكة العربية السعودية نموًا ملحوظًا في الطلب على النقل الجماعي.

شهدت المملكة العربية السعودية زيادة كبيرة في عدد مستخدمي حافلات النقل العام. وفقًا للهيئة العامة للإحصاء، بلغ عدد الركاب الذين تم نقلهم عبر حافلات النقل العام داخل وبين المدن خلال عام 2022 حوالي 43.5 مليون راكب، مما يمثل ارتفاعًا بنسبة 233.9% مقارنة بعام 2021 [8]. هذا النمو يعكس الجهود المبذولة لتعزيز وتطوير خدمات النقل العام في المملكة، بما يتماشى مع رؤية السعودية 2030.

2.1.5.1 تدشين شركات جديدة لتلبية الطلب المتزايد

في أكتوبر 2023، دشّن معالي وزير النقل والخدمات اللوجستية، المهندس صالح بن ناصر الجاسر، مشروعًا لخدمات النقل بالحافلات يربط أكثر من 200 مدينة ومحافظة في المملكة. شمل هذا المشروع ثلاث شركات رئيسية: "درب وطن"، "نورث وست باص"، و"سات" [9].

- **درب وطن:** حصلت على الامتياز الشمالي لتشغيل النقل بين المدن بالحافلات، وتقدم 124 رحلة يومية تربط 75 مدينة ومحافظة.
- **نورث وست باص:** تركز على الامتياز الشمالي الغربي، حيث توفر 190 رحلة يومية تربط أكثر من 70 مدينة ومحافظة.
- **سات:** تغطي المنطقة الجنوبية، وتشغل 178 رحلة يومية عبر 27 مسارًا.

2.1.5.2 تحديات التكامل بين الشركات

على الرغم من هذه التطورات، تواجه الشركات تحديات كبيرة تتعلق بالتكامل. حيث يمتلك كل منها منصة إلكترونية مستقلة ولا تتكامل خدماتها مع الأخرى، مما يؤدي إلى تكرار الخدمات وعدم الاستفادة المثلى من الموارد. كما أن بعض الشركات لا تزال تعتمد على أنظمة تسجيل يدوية، مما يحد من الكفاءة التشغيلية ويزيد من احتمالية الأخطاء البشرية. هذا النقص في التكامل والتنسيق بين الشركات يمثل تحديًا كبيرًا لتحقيق تجربة نقل جماعي أكثر تكاملًا واستدامة [9].

2.1.5.3 شركات نقل جماعي أخرى

بالإضافة إلى الشركات المذكورة، هناك العديد من شركات النقل الجماعي الأخرى التي تقدم خدماتها في مناطق مختلفة من المملكة وبعض هذه الشركات:

- شركة حافل: تقدم خدمات النقل التعليمي ورحلات الحج والعمرة، وتتمتع بحضور قوي في مكة المكرمة ومدن أخرى. ومع ذلك، تفتقر إلى التكامل مع المشغلين الآخرين، مما يؤدي إلى عدم وجود نظام نقل متكامل يسهل التنقل بين مختلف المناطق والشركات. [10]
- شركة تاسكو (TASEKO): تعمل في مجال النقل البري وتقدم خدماتها في مناطق متعددة، لكنها تعتمد بشكل كبير على الأنظمة التقليدية وتفتقر إلى الرقمنة، مما يؤثر على كفاءة العمليات وجودة الخدمة. [10]
- شركة بولت (BOLT): تقدم خدمات النقل في مدن مثل الرياض، جدة، ومكة، إلا أنها تعمل بشكل منفصل عن الشركات الأخرى، مما يحد من تكامل خدمات النقل على مستوى المملكة. [10].

والعديد من الشركات الأخرى والمؤسسات التي لا تملك حتى أدنى وجود رقمي على الإنترنت، فيتعذر الوصول إليها إلا من خلال الزيارات الميدانية وإجراء بحث أكثر دقة. ولكن الأكيد هو وجود تلك المؤسسات والشركات التي تعتبر ضمن الفئة المستهدفة لتطوير خدمات النقل الجماعي

2.1.6 أهمية التحول الرقمي

2.1.6.1 لماذا يحتاج قطاع النقل الجماعي إلى التحول الرقمي؟

يعد التحول الرقمي أمراً أساسياً في قطاع النقل الجماعي لتحسين كفاءة الخدمات وزيادة رضا العملاء. يمكن للتكنولوجيا أن توفر حلولاً تساهم في:

- تعزيز التكامل: من خلال ربط المشغلين المختلفين وتحسين تجربة العملاء. توفير بيانات فورية: مثل الجداول الزمنية والمسارات، مما يسهل على الركاب التخطيط لرحلاتهم.
- تقليل التكاليف التشغيلية: عبر أتمتة العمليات واستخدام تقنيات مثل الذكاء الاصطناعي لتحليل البيانات واتخاذ قرارات أفضل. تشير الأبحاث إلى أن رقمنة أنظمة النقل تؤدي إلى زيادة الكفاءة التشغيلية وتعزيز رضا العملاء، حيث تتيح التطبيقات الذكية للركاب حجز الرحلات وتتبع الحافلات بسهولة [1]، [2].

2.1.6.2 أمثلة من دول أخرى حققت نجاحاً في التحول الرقمي

- تركيا - تطبيقات EnUcuz O-Bilet: تطبيقات مثل EnUcuz O-Bilet في تركيا توفر منصات رقمية تمكن العملاء من مقارنة خيارات النقل المختلفة، حجز التذاكر بسهولة، والوصول إلى جداول الحافلات بين المدن. هذه المنصات ساعدت على تعزيز التكامل بين الشركات المختلفة وجعلت عملية النقل أكثر كفاءة وسلاسة [11].
- عالمي - تطبيق Booking: تطبيق Booking.com يُستخدم عالمياً لتوفير حلول شاملة لحجز تذاكر السفر والتنقل. يتيح التطبيق تكاملاً بين وسائل النقل المختلفة مثل الحافلات، القطارات، والطيران، مما يجعل تخطيط الرحلات أكثر سهولة للعملاء. هذه النماذج تثبت أهمية الرقمنة في تحسين تجربة العملاء [12].
- ألمانيا والنمسا وسويسرا - نظام Verkehrverbund: في هذه الدول، أدى التكامل الرقمي إلى إنشاء شبكات نقل جماعي مترابطة تشمل الحافلات والقطارات. النظام يستخدم تطبيقات ذكية تسهل الحجز والتخطيط، مما زاد من كفاءة النقل العام ورفع عدد المستخدمين [6].

2.1.7 تأثير غياب الرقمنة

2.1.7.1 التأثير السلبي على العملاء

غياب الرقمنة في قطاع النقل الجماعي يؤدي إلى تعقيد تجربة العملاء بشكل ملحوظ. من أبرز التحديات التي يواجهها العملاء:

- تعقيد الحجز: الحجز اليدوي يتطلب زيارة المكاتب أو الاعتماد على مكالمات هاتفية، مما يزيد من الوقت والجهد المبذول للحصول على تذكرة سفر. تشير الدراسات إلى أن غياب المنصات الرقمية يُعد أحد أبرز العقبات أمام تحسين تجربة العملاء [4].
- عدم توفر معلومات فورية: غياب الأنظمة الرقمية يجعل من الصعب على العملاء معرفة الجداول الزمنية، توفر الرحلات، أو تغييرات اللحظة الأخيرة، مما يؤدي إلى تجربة غير مرضية. تطبيقات مثل O-Bilet في تركيا توفر مثالاً واضحاً على كيفية تجاوز هذه المشكلة من خلال تقديم خدمات متكاملة ومعلومات فورية عبر الإنترنت [11].

2.1.7.2 التأثير السلبي على مزودي الخدمة

مزودو الخدمة يواجهون تحديات كبيرة نتيجة غياب الرقمنة:

خسارة العملاء: العملاء يفضلون الخدمات الرقمية التي تقدم تجربة سلسة وسريعة، مما يدفعهم إلى البحث عن بدائل رقمية حتى لو كانت خارج السوق المحلية. في ألمانيا وسويسرا، أدى التكامل الرقمي إلى تحسين تجربة العملاء بشكل كبير وزيادة ولائهم للخدمات المحلية [6].

صعوبة إدارة العمليات: الأنظمة اليدوية تعيق إدارة الموارد مثل الأسطول، الجداول، والموظفين بكفاءة، وتزيد من الأخطاء البشرية. تقنيات مثل حلول البرمجيات كخدمة (SaaS) أثبتت أنها فعالة في تقليل هذه المشكلات، كما أشار إليها [3].

2.1.8 الحاجة إلى الحل الرقمي

2.1.8.1 لماذا الحل الرقمي مثل "Bus E-Ticket Platform" ضروري الآن؟

يهدف مشروع BUS E-TICKET Platform إلى سد الفجوة بين مقدمي خدمات النقل الذين يفتقرون إلى البنية الرقمية والمستخدمين الذين يعانون من التعقيدات المرتبطة بالحجوزات اليدوية. أهمية الحل الرقمي "Bus E-Ticket"

2.1.8.2 حل مشاكل مزودي الخدمة:

المشروع يعالج تحدي عدم التكامل الرقمي من خلال توفير البنية التحتية الرقمية للشركات التي تفتقر إلى التقنيات اللازمة للتحويل الرقمي. باستخدام نموذج SaaS، تتمكن الشركات من تحسين كفاءتها وإنتاجيتها عبر لوحة تحكم رقمية توفر أدوات لإدارة الرحلات، السائقين، المركبات، والتقارير التشغيلية. بالإضافة إلى ذلك، تدعم المنصة أنظمة الدفع الإلكتروني، مما يساهم في تقليل الأخطاء البشرية وتحسين العمليات التشغيلية (3).

2.1.8.3 التكامل الرقمي:

توفر المنصة حلاً فريداً للتكامل الرقمي من خلال جمع جميع خدمات الرحلات من مقدمي الخدمة في قسم مركزي مخصص داخل المنصة، مما يعزز من سهولة وصول العملاء إلى الخيارات المختلفة. في الإصدارات المستقبلية، تهدف المنصة إلى توفير جاهزية رقمية عالية للتكامل مع شركات أخرى تعمل خارج إطار المنصة، مما يعزز من شمولية الخدمات وتوسيع نطاقها. تبسيط تجربة المستخدم: يتيح المشروع للمستخدمين حجز الرحلات بسهولة من خلال تطبيق أو موقع إلكتروني، مما يلغي الحاجة إلى زيارة مكاتب الشركات.

تشمل الميزات المقدمة:

- البحث المتقدم

يتيح للمستخدمين تصفية الرحلات بناءً على المدينة، السعر، أو الفئة. الحجز الإلكتروني والدفع عن بُعد: يضمن ذلك راحة وسرعة في الإجراءات، مما يجعل التجربة أكثر سلاسة (11).

- إدارة الحجز

يوفر للعملاء إمكانية تعديل أو إلغاء الحجز بسهولة مع استرداد الأموال حسب السياسات.

- تعزيز التنافسية للشركات الصغيرة والمتوسطة

يدعم المشروع الشركات الصغيرة والمتوسطة التي تفتقر إلى القدرات الرقمية من خلال منحها إمكانية الوصول إلى أدوات متقدمة لتوسيع أعمالها. يتم ذلك عبر حملات تسويقية داخل المنصة، عروض ترويجية، وبرامج ولاء لجذب المزيد من العملاء وزيادة الحصة السوقية.

- تحسين رضا العملاء

باستخدام تقنيات مثل الإشعارات الفورية وتوفير الوصول إلى البيانات الفورية حول الرحلات والتوافر، تضمن المنصة تحسين تجربة العملاء، مما يؤدي إلى رفع نسبة الولاء وزيادة استخدام النقل الجماعي.

2.1.8.4 الفجوة بين التقدم التكنولوجي الحالي ومتطلبات النقل الجماعي

رغم التقدم التكنولوجي الكبير، لا تزال هناك فجوة بين مستوى الرقمنة المطلوب وتطبيقها في قطاع النقل الجماعي:

- غياب التكامل: العديد من المشغلين يعملون بشكل منفصل بدون أنظمة متكاملة تسهل تجربة العملاء. تجربة ألمانيا والنمسا مع نظام Verkehrsverbund توضح كيف أن التكامل الرقمي بين شركات النقل يؤدي إلى تحسين كبير في الكفاءة والراحة [6]
- نقص الاستثمار في التكنولوجيا: قلة الاستثمار في أنظمة الحجز الإلكتروني وتحليل البيانات تمنع مزودي الخدمة من الاستفادة الكاملة من التقدم التكنولوجي. تقرير المنتدى الاقتصادي العالمي أشار إلى أن البنية التحتية الرقمية تعد حجر الزاوية للتحويل الرقمي الناجح [1]

2.2 أصحاب المصلحة

فهم متطلبات المشروع بوضوح يساعد تحديد أصحاب المصلحة على جمع المتطلبات بشكل دقيق لضمان أن المنتج النهائي يلبي احتياجاتهم. بدون هذا التحديد، قد تُهدر الموارد على ميزات غير ضرورية ويساهم في إدارة التوقعات فيمكن من خلال إشراك أصحاب المصلحة تحديد توقعاتهم وإدارة المخاطر المتعلقة بالتغيرات أو عدم التطابق بين أهداف المشروع واحتياجاتهم.[13]

تعزيز التعاون يساهم تحديد الأطراف المختلفة في خلق بيئة تعاونية بين الفرق الفنية، الإدارية، ومزودي الخدمة، مما يساعد على حل المشكلات بشكل أسرع. زيادة فرص نجاح المشروع من خلال التواصل المستمر مع أصحاب المصلحة أثناء دورة حياة المشروع، يمكن اكتشاف التحديات المحتملة والتكيف معها مبكرًا، مما يقلل من فرص الفشل.[13]

تقييم الأداء والمخرجات يساعد تحديد أصحاب المصلحة على وضع معايير واضحة لتقييم النجاح من خلال تحديد المقاييس والنتائج المتوقعة لكل جهة مستفيدة.[13]

2.2.1 تحديد أصحاب المصلحة

- عملاء المنصة (End Users): وصف تفصيلي لفئات العملاء وكيف يستفيدون من المشروع.
- مزودو الخدمة (Service Providers): دورهم في تقديم الرحلات وكيف يدعم المشروع عملياتهم.
- إدارة المشروع والمنصة (Platform Management): فريق العمل التقني والإداري الذي يدير النظام.
- الجهات التنظيمية والحكومية: دورها في وضع المعايير وضمان الامتثال والسياسات.
- المستثمرون والشركاء الاستراتيجيون: دعمهم المالي والتقني.
- مزودو التقنيات (Technical Providers): شركات البرمجيات الموفرة للاستضافة.

جدول تحديد أصحاب المصلحة باختصار. Tablo 2.1

الأهمية للمشروع	الدور	الفئة
زيادة عدد المستخدمين ونجاح المشروع	استخدام المنصة لإجراء الحجوزات	العملاء
ضمان وجود الرحلات على المنصة	تقديم الرحلات وإدارة العمليات	مزودو الخدمة
الحفاظ على الأداء والجودة	تطوير وصيانة النظام	إدارة المنصة

تعزيز الموثوقية والقانونية	ضمان الامتثال للقوانين	الجهات الحكومية
توفير الموارد المالية والتقنية	تمويل المشروع	المستثمرون
تعزيز الكفاءة التشغيلية	تطوير الأنظمة ودعمها	مقدمو التقنيات

2.3 نموذج مشروع BUS E-Ticket Platform

مشروع "BUS E-Ticket Platform" هو حل رقمي مبتكر يهدف إلى رقمنة قطاع النقل الجماعي بين المدن من خلال تقديم منصة إلكترونية متكاملة لحجز التذاكر وإدارة الرحلات. يعتمد المشروع على نماذج أعمال تقنية حديثة نموذج تقديم البرمجيات (البرمجيات كخدمة SaaS) ونموذج الاعمال (شركة إلى شركة إلى مستهلك B2B2C) لتقديم خدماته لمزودي النقل والعملاء النهائيين على حد سواء.

2.3.1 كيف يطبق مشروع BUS E-Ticket نموذج SaaS؟

2.3.1.1 تعريف نموذج تقديم البرمجيات SAAS

نموذج البرمجيات كخدمة (SaaS) هو طريقة لتقديم التطبيقات البرمجية عبر الإنترنت بدلاً من تثبيتها على أجهزة المستخدمين. يتيح هذا النموذج للمستخدمين الوصول إلى البرامج عبر المتصفح، مما يتيح لهم استخدام الوظائف المتاحة دون الحاجة إلى شراء أو إدارة البنية التحتية التقنية.

يقدم "BUS E-Ticket Platform" كحل SaaS، أي كخدمة برمجية قائمة على السحابة يمكن لمزودي خدمات النقل الجماعي الاشتراك بها لاستخدام الأدوات الرقمية التي توفرها المنصة دون الحاجة إلى تطوير بنية تحتية خاصة بهم.

2.3.1.2 آلية عمل نموذج SaaS ونطاق التطبيق

لكون هذا الإصدار الأول من مشروع BUS E-TICKET Platform فلن يتم تطبيق نموذج تقديم البرمجيات SAAS بشكل كامل.

2.3.1.2.1 نطاق التطبيق:

2.3.1.2.1.1 الوصول عبر الإنترنت

يستطيع مزودو الخدمة إدارة الرحلات والأنشطة عبر لوحة تحكم متاحة عبر الإنترنت وإدارة جميع الأنشطة وحسابات الدفع والمدفوعات والحجوزات عن بعد بالإضافة إلى تمكن المدراء من نفس الشيء إدارة المنصة كاملاً من الحسابات لحكومة تسجيل موفري الخدمة..الخ.

2.3.1.2.1.2 التحديثات

- سهولة تحديث النظام دون الحاجة إلى توقف الخدمة أو إجراء تغييرات من طرف مزودي الخدمة او المدراء

2.3.1.2.2 خارج نطاق التطبيق (ما لن يتم تطبيقه في الإصدار الأول):

2.3.1.2.2.1 غياب سياسات الربح

ستعمل المنصة بشكل مجاني في هذه المرحلة، مع خطط مستقبلية لتوفير باقات مدفوعة أو فرض رسوم على خدمات متقدمة.

2.3.1.2.2.2 عدم تفعيل خطط التوسع للحسابات

لن يتم تخصيص موارد مختلفة لحسابات الشركات الكبرى، حيث ستحصل جميع حسابات مزودي الخدمة على نفس الإمكانيات.

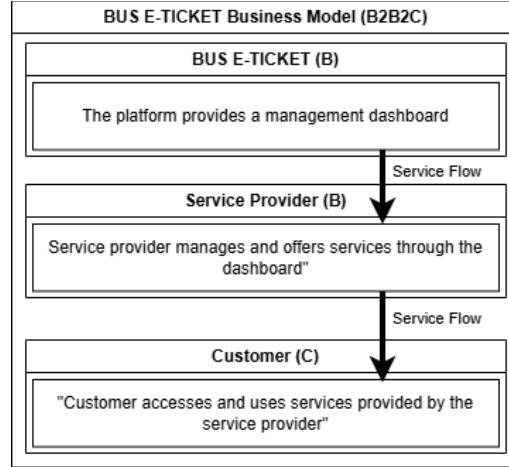
2.3.1.2.2.3 السوق المحلي فقط

سيركز الإصدار الأول على السوق المحلي، مع خطط مستقبلية للتوسع إقليميًا أو دوليًا.

2.3.1.3 فوائد نموذج SaaS/مشروع BUS E-Ticket

- تقليل التكاليف التشغيلية: يوفر SaaS مزايا لمزودي الخدمة دون الحاجة إلى استثمار كبير في البنية التحتية.
- التكامل السريع: يمكن لمزودي الخدمة البدء في استخدام المنصة فورًا بعد التسجيل.
- التحديثات والأمان: يحصل المستخدمون على تحديثات أمنية وتطويرات دورية دون الحاجة إلى إدارة البنية التحتية.
- سهولة التوسع: يمكن لمزودي الخدمة توسيع نطاق أعمالهم بسهولة من خلال المنصة.
- واجهة مستخدم سهلة: توفر المنصة واجهة بسيطة وسهلة الاستخدام لإدارة الرحلات والحجوزات بكفاءة.

2.3.2 كيف يطبق مشروع BUS E-Ticket نموذج B2B2C؟



2.3.2.1 تعريف نموذج الأعمال B2B2C

B2B2C (شركة إلى شركة إلى مستهلك) هو نموذج أعمال يوضح هيكلية الخدمات أو هيكلية تدفق الخدمات بين الأطراف الموفرة والمستهلكة والوسيلة ويجمع بين نموذجين تقليديين:

- B2B (شركة إلى شركة): حيث تقدم شركة ما خدمات أو منتجات لشركة أخرى.
- B2C (شركة إلى مستهلك): حيث تقدم الشركة خدمات أو منتجات مباشرة للمستهلكين الأفراد.

في B2B2C، تلعب الشركة (B) دور الوسيط بين شركة أخرى (B) والمستهلك النهائي (C). تعمل الشركة الوسيطة على توفير منصة أو قناة تتيح للشركة الأولى الوصول إلى المستهلكين وبيع خدماتها أو منتجاتها بشكل مباشر.

2.3.2.2 تطبيق نموذج B2B2C على BUS E-TICKET

في مشروع BUS E-Ticket Platform، يتم تطبيق نموذج B2B2C من خلال دور المنصة كوسيط رقمي يربط بين مزودي خدمات النقل الجماعي (B) والمستهلكين النهائيين أو المسافرين (C).

2.3.2.2.1 كيف يعمل النموذج في المنصة:

B2B (شركة إلى شركة): تقوم منصة BUS E-Ticket بتوفير أدوات تقنية لمزودي خدمات النقل (الشركات). يتيح ذلك للشركات إدراج رحلاتها وإدارتها على المنصة.

B2C (شركة إلى مستهلك): يتمكن العملاء (المستهلكون) من البحث عن الرحلات، حجز التذاكر، والدفع إلكترونياً مباشرة عبر المنصة.

B2B2C (التكامل بين الطرفين): تعمل المنصة كوسيط، حيث تعرض خدمات مزودي النقل بشكل مباشر للمستهلكين النهائيين. يحصل العملاء على خيارات متعددة من مختلف شركات النقل الجماعي في واجهة واحدة، مما يسمح لهم بمقارنة الأسعار والرحلات بسهولة.

مثال عملي: منصة BUS E-Ticket Platform توفر لوحة التحكم لشركة نقل (B) فتُدرج عن طريقها رحلاتها. المستهلك (C) يدخل إلى المنصة، ويختار رحلته من بين العروض المتاحة من شركات نقل متعددة، ثم يحجز التذكرة مباشرة عبر المنصة. BUS E-Ticket تعالج الدفع وترسل تفاصيل الحجز للشركة والعميل.

2.3.2.2.2 الفوائد نموذج B2B2C

لشركات النقل (B): الوصول إلى قاعدة عملاء أكبر دون الحاجة إلى إنشاء منصة مستقلة. إدارة الرحلات بسهولة عبر لوحة تحكم موحدة. للمستهلكين (C): سهولة الوصول إلى خيارات متنوعة من مقدمي الخدمة. تجربة حجز سلسة ومتكاملة توفر الوقت والجهد.

2.3.2.2.3 كيف يعزز B2B2C نجاح المشروع:

- يوفر نموذج B2B2C فرصة لشركات النقل للتواصل مع العملاء مباشرة عبر منصة واحدة، مما يعزز التكامل والتنافسية.
- يساهم في رفع معدلات الحجز وزيادة الإيرادات لمزودي الخدمة من خلال تقديم تجربة رقمية متطورة وسلسة للمستهلك.

2.3.2.3 نطاق تطبيق نموذج الأعمال B2B2C على BUS E-TICKET

2.3.2.3.1 نطاق التطبيق (ما سيتم تطبيقه):

2.3.2.3.1.1 واجهة موحدة لعرض الرحلات

سيتمكن مزودو خدمات النقل الجماعي من إدراج رحلاتهم وتحديد أسعار التذاكر مباشرة عبر المنصة. يمكن للعملاء النهائيين البحث، الحجز، والدفع من خلال واجهة موحدة تعرض جميع خيارات الرحلات من مختلف مزودي الخدمة.

2.3.2.3.1.2 حجز مباشر من العملاء (C)

- تتيح المنصة للعملاء النهائيين (المسافرين) حجز التذاكر بشكل مباشر بعد اختيار الرحلة المناسبة من بين مقدمي خدمات النقل المدرجين على المنصة.
- يتم إشعار مزودي الخدمة فور تأكيد الحجز، مما يضمن عملية سلسة وفعالة.

2.3.2.3.1.3 الدفع الإلكتروني

سيتم دمج بوابات دفع إلكتروني تتيح للعملاء الدفع المباشر عبر الإنترنت، مما يسهل المعاملات ويقلل من العمليات النقدية التقليدية.

2.3.2.3.1.4 الربط بين الشركات والعملاء

المنصة تعمل كوسيط رقمي يربط بين شركات النقل الجماعي (B) والعملاء النهائيين (C)، مما يخلق بيئة متكاملة توفر خدمات متعددة.

2.3.2.3.2 خارج نطاق التطبيق (ما لن يتم تطبيقه في الإصدار الأول):

2.3.2.3.2.1 عدم تخصيص خدمات فردية مباشرة للعملاء (B2C فقط)

لن يتم تقديم خدمات مخصصة للأفراد خارج إطار شركات النقل. جميع الرحلات ستكون عبر مزودي خدمة مسجلين فقط.

2.3.2.3.2.2 عدم دعم حجوزات الشركات الكبرى (B2B فقط):

لن يشمل الإصدار الأول خدمات مخصصة لشركات أو مؤسسات ترغب في حجز رحلات جماعية. يقتصر الحجز على الأفراد فقط (C).

2.3.2.3.2.3 غياب التكامل مع منصات النقل الخارجية:

لن يتضمن النظام في هذه المرحلة تكاملاً مع منصات أخرى أو شركات نقل خارجية تعمل بأنظمة مستقلة.

2.3.2.3.2.4 عدم توفير حلول مخصصة للرحلات الحكومية (G2B2C)

لن يتم تخصيص خدمات لرحلات حكومية أو مؤسسية، مع تركيز المنصة على القطاع الخاص ومزودي النقل الجماعي التجاري.

2.3.2.3.2.5 إدارة الحجز والتذاكر

لن يتمكن العملاء في الإصدار الأول من تعديل الحجوزات، أو إلغاؤها عبر المنصة.

2.4 مشاريع مشابهة

2.4.1 المشاريع المشابهة:

2.4.1.1 درب وطن

درب وطن هي شركة سعودية متخصصة في تقديم خدمات النقل الجماعي بالحافلات، وتركز على تلبية احتياجات النقل بين المدن باستخدام نموذج B2C، إلا أن النظام يفتقر إلى التكامل مع شركات نقل أخرى، مما يحد من إمكانية تقديم خدمات موحدة للعملاء. [9]

2.4.1.2 نورث وست باص

نورث وست باص تُعد من الشركات السعودية العاملة في مجال النقل الجماعي بين المدن. تعتمد الشركة على نموذج B2C، لكنها تفتقر إلى التكامل الرقمي، مما يضعف من كفاءة إدارة الرحلات إلكترونياً ويجعل عملية الحجز تقليدية. [9]

o-bilet 2.4.1.3

o-bilet هو تطبيق تركي متقدم يوفر خدمات حجز التذاكر للنقل الجماعي بين المدن، ويتيح للمستخدمين مقارنة الرحلات المتاحة من عدة شركات نقل. يتميز التطبيق بتكامل عالٍ مع مقدمي الخدمة ويدعم تطبيقات الهواتف المحمولة لزيادة سهولة الاستخدام [11].

2.4.2 جدول مقارنة المشاريع المشابهة

Tablo 2.2. BUS E-TICKET لـ BUS E-TICKET مقارنة المشاريع المشابهة

المعيار	BUS E-TICKET Platform	درب وطن	نورث وست باص	o-bilet
النموذج التجاري	B2B2C	B2C	B2C	B2C
واجهة المستخدم (UI)	ممتازة	مقبولة	مقبولة	جيدة
الجمهور المستهدف	موفرين الخدمة والعملاء	العملاء	العملاء	العملاء
تجربة المستخدم (UX)	عالية	ضعيفة	ضعيفة	متوسطة
التكامل (Integration)	تكامل عالي	تفتقر للتكامل	تفتقر للتكامل	تكامل عالي
الدفع الإلكتروني	متوفر	غير متوفر	متوفر	متوفر
تطبيقات الهاتف (Mobile App)	PWA	لا يوجد	لا يوجد	تطبيق جوال
اللغات	إنجليزية	إنجليزية + محلية	إنجليزية + محلية	إنجليزية
وجود باصات خاصة	لا يوجد	يوجد	يوجد	لا يوجد

2.4.3 استنتاج

2.5 مبررات المشروع

2.5.1 الحاجة إلى التحول الرقمي في قطاع النقل الجماعي

- يواجه قطاع النقل الجماعي بين المدن تحديات تشغيلية كبيرة بسبب غياب البنية التحتية الرقمية. استخدام الأنظمة اليدوية في إدارة الحجز والعمليات التشغيلية يؤدي إلى أخطاء بشرية، انخفاض الكفاءة، وصعوبة إدارة البيانات بفعالية.
- مع ازدياد الطلب على خدمات النقل الجماعي، أصبح التحول الرقمي ضرورة لتلبية احتياجات العملاء ومزودي الخدمة على حد سواء.

2.5.2 دعم التكامل بين مزودي الخدمة

عدم وجود تكامل بين مزودي خدمات النقل الجماعي يجعل من الصعب على العملاء الوصول إلى خيارات مرنة وموحدة. المشروع يهدف إلى توفير منصة رقمية متكاملة تعمل على توحيد مزودي الخدمات في بيئة مركزية، مما يعزز من سهولة الوصول للمستخدمين ورفع الكفاءة التشغيلية للشركات.

2.5.3 تحسين تجربة العملاء

العملاء يواجهون تحديات كبيرة تتعلق بتعقيد عمليات الحجز التقليدية وعدم توافر المعلومات الفورية عن الرحلات. المشروع يوفر حلاً تقنياً يمكّن العملاء من حجز التذاكر إلكترونياً، تتبع الرحلات، وإدارة الحجوزات بسهولة، مما يحسن تجربة الاستخدام ويزيد من رضا العملاء.

2.5.4 القيمة المضافة

التي يقدمها المشروع يوفر المشروع أدوات متقدمة مثل تقارير الأداء، التكامل الرقمي، والدفع الإلكتروني، مما يساهم في تحسين كفاءة العمليات وتقليل التكاليف التشغيلية للشركات.

2.5.5 الفرصة الاقتصادية والتوسع المستقبلي

التوسع المستقبلي للمشروع يتماشى مع النمو العالمي لنماذج B2B2C & SaaS. من المتوقع أن يساهم المشروع في جذب استثمارات جديدة ودعم الاقتصاد الرقمي من خلال تقديم حلول مبتكرة وقابلة للتوسع [1]

3 التصميم والتحليل

3.1 تحليل المتطلبات

تم متابعة الإحصائيات الرسمية للأسواق المستهدفة، مع التركيز على الأسواق التي تفتقر إلى خدمات رقمية في مجال النقل الجماعي. كما تمت دراسة تطبيقات محلية وعالمية مشابهة من دول مختلفة لاستخلاص أفضل الممارسات.

- تحديد المتطلبات وأصحاب المصلحة: بدأت العملية بتحديد المتطلبات الأساسية للنظام ودراسة احتياجات العملاء ومزودي الخدمة.
- التخطيط والتصميم: تم تصميم الدياغرامات اللازمة مثل Use Case و UML Class و UML Package و ERD (Schema و Diagram) لتوثيق التخطيط وتوضيح هيكلية المشروع.

3.1.1 المتطلبات الوظيفية

3.1.1.1 الحساب

- المنصة تحوي ثلاث أنواع حسابات (المدراء, العملاء, موفري الخدمة)
- يستطيع العملاء الانضمام للمنصة والاستفادة من الخدمات عن طريق تسجيل حساب جديد او باستعمال التسجيل السريع عبر موقع التواصل الاجتماعي
- ينضم موفري الخدمة الى المنصة عن طريق تعبئة نموذج وارسال طلب عن طريق المنصة وعند الموافقة يتم انشاء حساب فعال
- يستطيع مدير المنصة فقط انشاء حسابات مدراء.
- يستطيع كل نوع من الحسابات إدارة معلومات حسابه ضمن النطاق المسموح
- يستطيع المدير إدارة الحسابات بشكل واسع

3.1.1.2 الرحلات

- يستطيع موفر الخدمة إدارة الرحلات من ارفاق رحلات جديدة, تعديل رحلات, وحتى حذف الرحلات.
- يستطيع مدير الاطلاع على جميع الرحلات
- يستطيع العملاء المسجلين والزوار (العملاء الغير منضمين للمنصة) الاستعلام عن الرحلات
- تظهر الرحلات محتوية معلومات موفر الخدمة

3.1.1.3 الحجوزات

- يستطيع موفر الخدمة استعلام عن الحجوزات والوصول لمعلومات العملاء الحاجزين بشكل محدود
- يستطيع العميل احجز رحلة لنفسه او لشخص آخر

- يستطيع المدير الوصول الى جميع الرحلات بشكل واسع
- يصل العميل للحجوزات عن طريق حسابه او عن طريقة صفحة الاستعلام على الحجوزات

3.1.1.4 الدفع الآمن

- يدعم المنصة الدعم الدفع بانواع مختلفة او على الأقل نوع واحد آمن ومشهور.

3.1.1.5 استعلام المدفوعات

- يمكن للعميل المسجل الوصول للمدفوعات السابقة
- يمكن لموفر الخدمة استعلام جميع المدفوعات داخل المنصة

3.1.1.6 الفواتير والتذاكر

- بعد نجاح الحجز يتم انشاء فاتورة وتذكير للعميل
- يستطيع العميل الوصول الى تذاكر وفواتير الرحلات السابقة
- يستطيع العميل الوصول الى الفواتير اللازمة
- تستطيع الجهات الحكومية ذات الصلة الوصول الى الفواتير

3.1.2 المتطلبات غير الوظيفية

3.1.2.1 الأداء

يجب أن تكون استجابة النظام سريعة بحيث لا يتجاوز زمن تحميل الصفحة 3 ثواني في الظروف العادية. الأداء. يتم تحسين استعلامات قاعدة البيانات باستخدام تقنيات مثل T-SQL لضمان سرعة الاستجابة.

3.1.2.2 القابلية للتوسع

يجب تصميم النظام بعمارة مرنة لتسهيل إضافة ميزات مستقبلية دون التأثير على الأداء. يجب ان يكون النظام كل خدماته مبنية في السيرفر لضمان الاستقلالية وتعدد التوسع الى منصات مختلفة غير الويب

3.1.2.3 الأمان

تشفير جميع البيانات الحساسة (مثل كلمات المرور والمدفوعات) باستخدام خوارزميات التشفير القوية (Hashing). دعم التسجيل الآمن باستخدام التحقق بخطوتين (FA2). تغيير كلمة المرور بخطوتين. تقييد وصول البيانات لموفر الخدمة من خلال سياسات الوصول المحدودة. التحقق من صحة المدخلات لمنع هجمات SQL Injection أو XSS. تأمين واجهة برمجة التطبيقات (APIs) باستخدام Tokens أو OAuth.

3.1.2.4 سهولة الاستخدام

تصميم واجهة المستخدم (UI) بشكل مبسط وتفاعلي باستخدام معايير تصميم حديثة مثل SPA. دعم تجربة المستخدم (UX) بمميزات مثل السرعة والوضوح في التنقل بين صفحات النظام. يجب أن تكون واجهة المستخدم متوافقة مع الأجهزة المختلفة (سطح المكتب،

الجوال، التابلت)، استخدام الإشعارات التفاعلية لتسهيل التفاعل مع النظام (تأكيد الحجز، إشعارات الدفع). يجب ان يدعم الموقع تقنية ال PWA لتعزيز تجربة المستخدم على الأجهزة المحمولة

3.1.2.5 التوافقية

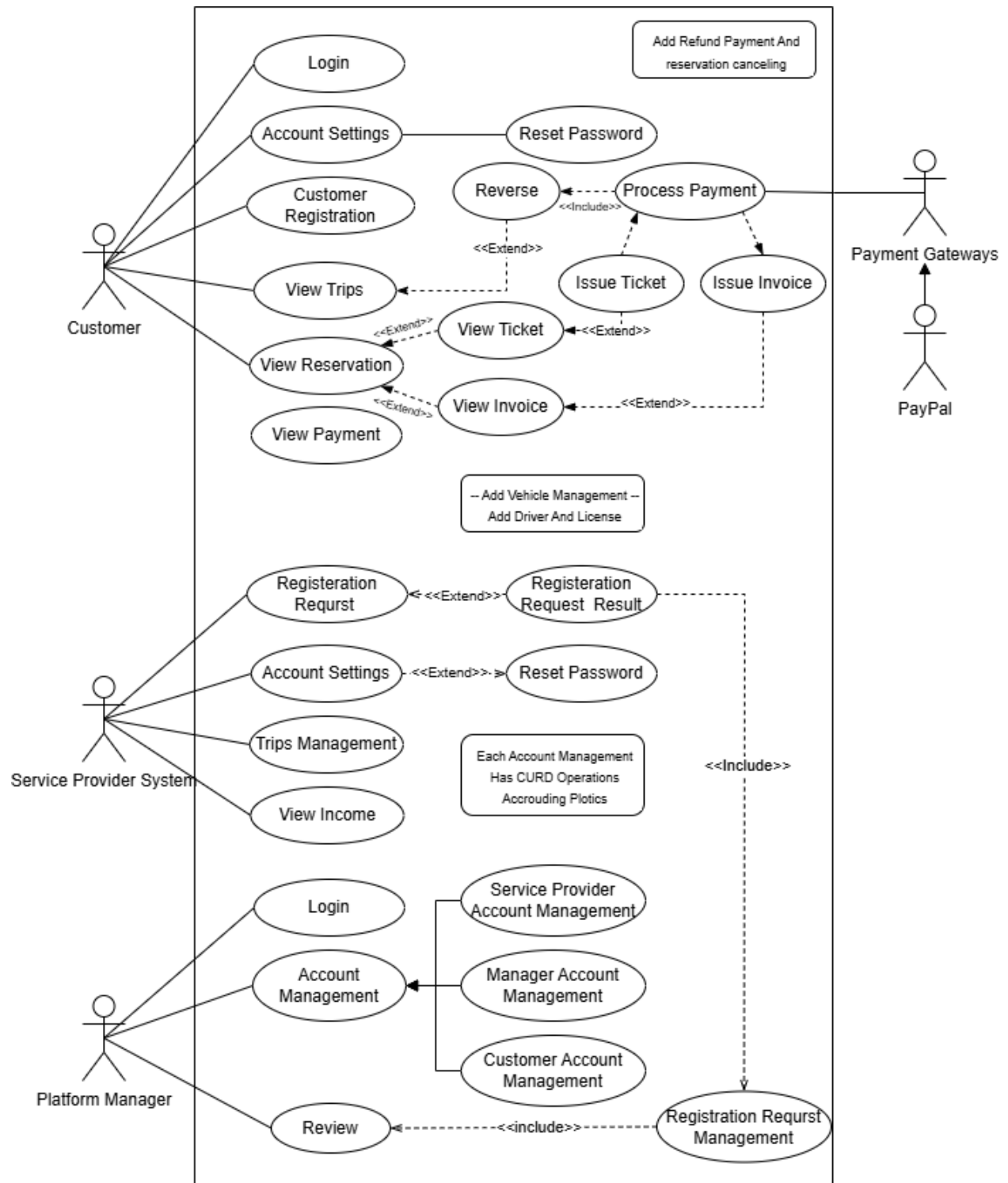
يجب أن يعمل النظام على جميع المتصفحات الحديثة (مثل Chrome، Edge، Firefox، Safari). دعم تعدد اللغات (الإنجليزية) لضمان انتشار أوسع.

3.1.2.6 الصيانة

تصميم الكود باستخدام SOLID Principles لضمان سهولة الصيانة. يجب أن تكون تقارير الأخطاء (Logs) واضحة ويتم تسجيلها وحفظها والوصول لها. لتساعد في تسريع عملية حل المشكلات. توفير تحديثات دورية لتحسين النظام وإضافة ميزات جديدة مع الحفاظ على الأداء.

3.1.2.7 القابلية للنقل

إمكانية نقل النظام إلى خوادم مختلفة دون الحاجة إلى تغييرات كبيرة في الكود. دعم تشغيل النظام على الخوادم السحابية مثل Azure أو AWS.

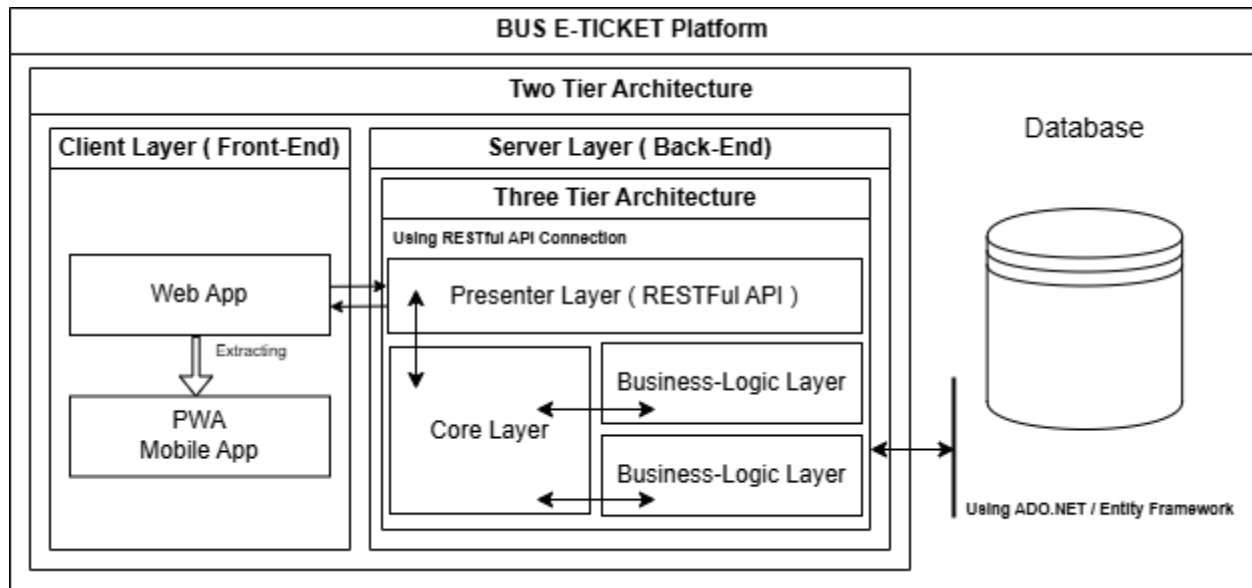


جدول سيناريوهات مخطط الحالة 3.1 Tablo

اسم الحالة	السيناريو	الممثل (Actor)
Login	يقوم المستخدم بتسجيل الدخول عن طريق إدخال بيانات الاعتماد.	العميل / موفر الخدمة / المدير
Account Settings	يضبط المستخدم إعدادات حسابه (تحديث البيانات).	العميل / موفر الخدمة
Reset Password	يطلب المستخدم إعادة ضبط كلمة المرور، ويتضمن هذا الإجراء الدخول إلى إعدادات الحساب.	العميل / موفر الخدمة
Customer Registration	يسجل العميل في النظام عبر تعبئة البيانات المطلوبة وتأكيد التسجيل.	العميل
View Trips	View يستعرض المستخدم الرحلات المتاحة، ويمتد هذا إلى حالة Reservation.	العميل
View Reservation	يعرض المستخدم تفاصيل الحجوزات المرتبطة بحسابه ويمتد ليشمل حالة View Ticket.	العميل
View Payment	يستعرض المستخدم تفاصيل المدفوعات السابقة المرتبطة بحسابه.	العميل
Process Payment	ينفذ المستخدم عملية الدفع ويتضمن هذا إصدار التذكرة من خلال حالة Issue Ticket.	العميل
Reverse	يعكس المستخدم عملية حجز معينة، إما بالإلغاء أو التعديل.	العميل
Issue Ticket	View يتم إصدار التذكرة للعميل بعد إتمام الدفع ويمتد إلى حالة Ticket.	النظام
Issue Invoice	View Invoice. يتم إصدار فاتورة بعد إتمام الدفع ويمتد إلى حالة	النظام
View Ticket	يستعرض المستخدم تفاصيل التذكرة بعد إصدارها	العميل
View Invoice	يستعرض المستخدم تفاصيل الفاتورة المرتبطة بحجزه	العميل
Registration Request	يقدم موفر الخدمة طلب تسجيل للانضمام إلى النظام، ويمتد إلى حالة Registration Request Result.	موفر الخدمة
Registration Request Result	يعرض النظام نتيجة طلب التسجيل المقدم من موفر الخدمة	النظام
Trips Management	يدير موفر الخدمة الرحلات من خلال إضافة أو تعديل أو حذف الرحلات.	موفر الخدمة

View Income	موفر الخدمة	يعرض موفر الخدمة الإيرادات المالية المرتبطة بحسابه
Account Management	المدير	Service Provider يدير المدير حسابات المستخدمين ويتضمن Account Management g Customer Account Management.
Review	المدير	يراجع المدير طلبات تسجيل موفري الخدمة ويتضمن حالة Registration Request Management.

3.2 معمارية النظام



3.2.1 معمارية البرمجيات

3.2.1.1 تعريف معمارية البرمجيات

معمارية البرمجيات تُعرّف بأنها الهيكل الأساسي الذي ينظم مكونات النظام البرمجي وعلاقاتها مع بعضها البعض، مع تحديد المبادئ التوجيهية لتصميم النظام وتطوره. تُعد المعمارية خطوة أساسية في دورة حياة تطوير البرمجيات، حيث تساعد على تقسيم النظام إلى وحدات قابلة للإدارة تسهل من فهمه وصيانته وتطويره. توفر معمارية البرمجيات مزايا عديدة مثل تعزيز الكفاءة، تسهيل التعاون بين فرق التطوير، وضمان مرونة النظام للتوسع المستقبلي. كما أنها تُساهم في تحسين الأداء والموثوقية وتقليل التكاليف التشغيلية عن طريق توفير بنية مُنظمة للنظام تدعم إعادة استخدام الكود وتطبيق معايير التصميم الجيدة [1].

3.2.1.2 أمثلة

3.2.1.2.1 Two-Tier Architecture (Client-Server)

تعتبر Two-Tier Architecture أحد أبسط أنواع المعماريات البرمجية، حيث تنقسم إلى طبقتين الطبقة الأولى الخادم والعميل

- (Client): تشمل واجهة المستخدم (User Interface) التي يتفاعل معها المستخدم، مثل تطبيقات سطح المكتب أو تطبيقات ويب بسيطة.
 - الطبقة الثانية (Server) تحتوي على قاعدة البيانات والخدمات التي تتعامل مع البيانات، حيث يرسل العميل طلباته إلى الخادم ويستقبل النتائج. تتميز هذه المعمارية بالبساطة وسهولة التنفيذ، لكنها تفتقر إلى المرونة وقابلية التوسع مقارنة بالمعماريات الأكثر تعقيدًا. مثال: تطبيق إدارة المكتبة البسيط حيث يرسل العميل استعلامًا مباشرًا إلى قاعدة البيانات لاسترجاع الكتب المتاحة.
- [1]

3.2.1.2.2 : Three-Tier Architecture

تُعد Three-Tier Architecture تطويرًا للمعمارية الثنائية، حيث تُقسم إلى ثلاث طبقات رئيسية:

- طبقة العرض (Presentation Layer): تشمل واجهة المستخدم (مثل تطبيق ويب أو تطبيق موبايل).
 - طبقة المنطق (Business Logic Layer): تحتوي على منطق الأعمال الذي يعالج البيانات ويُطبق القواعد الخاصة بالنظام.
 - طبقة البيانات (Data Layer): تحتوي على قاعدة البيانات التي تُخزن البيانات وتعالجها.
- هذه المعمارية تفصل بين واجهة المستخدم والمنطق وقاعدة البيانات، مما يُسهل الصيانة، يُحسّن الأداء، ويزيد من مرونة التطوير. مثال: أنظمة الحجز عبر الإنترنت حيث يُرسل المستخدم الطلب من خلال واجهة ويب، تقوم طبقة المنطق بمعالجة الطلب، ثم تُرجع البيانات المطلوبة من قاعدة البيانات. [18]

3.2.1.2.3 :N-Tier Architecture

تُعتبر N-Tier Architecture امتدادًا للمعمارية ثلاثية الطبقات (Three-Tier Architecture)، حيث يتم تقسيم النظام إلى عدة طبقات مستقلة، تتضمن طبقات فرعية مخصصة لوظائف معينة مثل الأمان، التكامل، وخدمات الويب. تدعم هذه المعمارية توزيع الخدمات، التوسع الأفقي، وتُسهل تطوير الأنظمة الكبيرة والمعقدة، مما يُحسن من صيانة النظام وإعادة استخدام مكوناته.

- طبقة العرض (Presentation Layer): واجهة المستخدم التي يتفاعل معها المستخدم، مثل تطبيق ويب أو تطبيق جوال.
- طبقة الخدمات (Service Layer): طبقة وسيطة تحتوي على واجهات RESTful API لتسهيل التكامل بين النظام ومكوناته.
- طبقة الأعمال (Business Logic Layer): معالجة البيانات وتطبيق العمليات، حيث يتم تنفيذ قواعد الأعمال ومعالجة منطق النظام.
- طبقة الوصول إلى البيانات (Data Access Layer - DAL): تُسهل الاتصال بقاعدة البيانات من خلال استعلامات البيانات أو الأطر مثل Entity Framework.
- طبقة النواة (CORE (Core Layer) : طبقة مركزية تحتوي على الكلاسات (Classes) والكيانات (Entities) المشتركة التي تستخدمها باقي الطبقات. تسهم في فصل التعريفات العامة والوظائف الأساسية لتعزيز التنظيم وإعادة الاستخدام، مع الالتزام بمبادئ OOP و SOLID.

وفقًا لـ Robert C. Martin في كتابه "Clean Code"، تعتمد بنية الطبقات على مبدأ تنظيم الكود لتحقيق أقصى درجات الوضوح والقابلية للصيانة، مع التأكيد على أهمية الطبقات المنفصلة مثل CORE لإعادة الاستخدام والحد من التبعية بين المكونات [19].

Microservices Architecture 3.2.1.2.4

تُقسم هذه المعمارية النظام إلى خدمات صغيرة مستقلة تُنفذ وظائف معينة، حيث تتواصل هذه الخدمات عبر واجهات برمجة التطبيقات (APIs). تُعد مرنة وتدعم التطوير المتوازي. مثال: أنظمة التجارة الإلكترونية مثل Amazon التي تعتمد على خدمات مصغرة لإدارة الطلبات، المخزون، والدفع. [3].

Serverless Architecture 3.2.1.2.5

تعتمد هذه المعمارية على خدمات سحابية تُنفذ التعليمات البرمجية عند الطلب، دون الحاجة لإدارة خوادم مادية. تُستخدم غالبًا لتطبيقات صغيرة أو وظائف مؤقتة. مثال: تطبيقات معالجة الصور أو وظائف الإشعارات السريعة في الأنظمة السحابية مثل [20]

3.2.2 معمارية هندسة البرمجيات المستخدمة في منصة BUS E-TICKET Platform

تم تصميم منصة BUS E-TICKET Platform باستخدام Two-Tier Architecture مقسمة إلى Client-Server Architecture، حيث يتفاعل العميل (Client) مع الخادم (Server) الذي يستخدم معمارية N-Tier Architecture لضمان الأداء العالي وتنظيم النظام. تقسيم المعمارية :

3.2.2.1 العميل (Client):

يمثل واجهة المستخدم (User Interface) التي يتفاعل معها المستخدمون النهائيون. يتم تنفيذها من خلال تطبيق ويب يعتمد على تقنية PWA (Progressive Web App) لتوفير تجربة سلسلة وسريعة على مختلف الأجهزة.

3.2.2.2 السيرفر (Server):

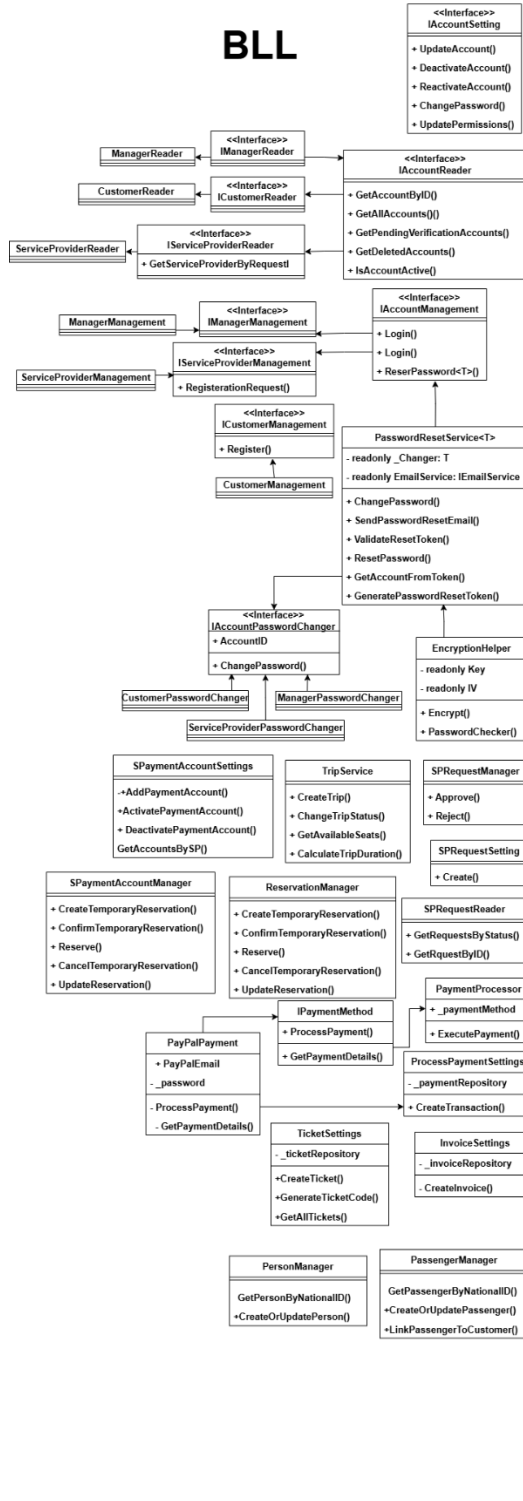
يستخدم السيرفر N-Tier Architecture لتقسيم النظام إلى أربع طبقات رئيسية لضمان استقلالية الطبقات وتنظيم المهام:

- طبقة الوصول إلى البيانات (Data Access Layer - DAL) وهي مسؤولة عن التعامل مع قاعدة البيانات، مثل تنفيذ الاستعلامات وحفظ البيانات واستردادها. تضمن أن الاتصال بالبيانات يتم بطريقة آمنة ومنظمة.
- طبقة النواة (Core Layer - CORE) وهي تحتوي على الكلاسات والكيانات المشتركة (Classes & Entities) التي تستخدمها باقي الطبقات. تساهم في فصل التعريفات والوظائف الأساسية المشتركة لتعزيز القابلية لإعادة الاستخدام والاستقلالية بين الطبقات.
- طبقة منطق الأعمال (Business Logic Layer - BLL) مسؤولة تُنفذ جميع القواعد والعمليات الخاصة بالنظام، مثل إدارة الحجوزات والرحلات. تعمل كوسيط بين البيانات والطبقة العليا لضمان تنفيذ منطق العمل المطلوب.
- طبقة العرض (Presentation Layer - PL) تقدم البيانات إلى المستخدم من خلال الواجهة الأمامية (Front-End). تستقبل الطلبات من العميل، تمررها إلى الطبقات السفلية لمعالجتها، ثم تعرض النتائج بطريقة سهلة الاستخدام.

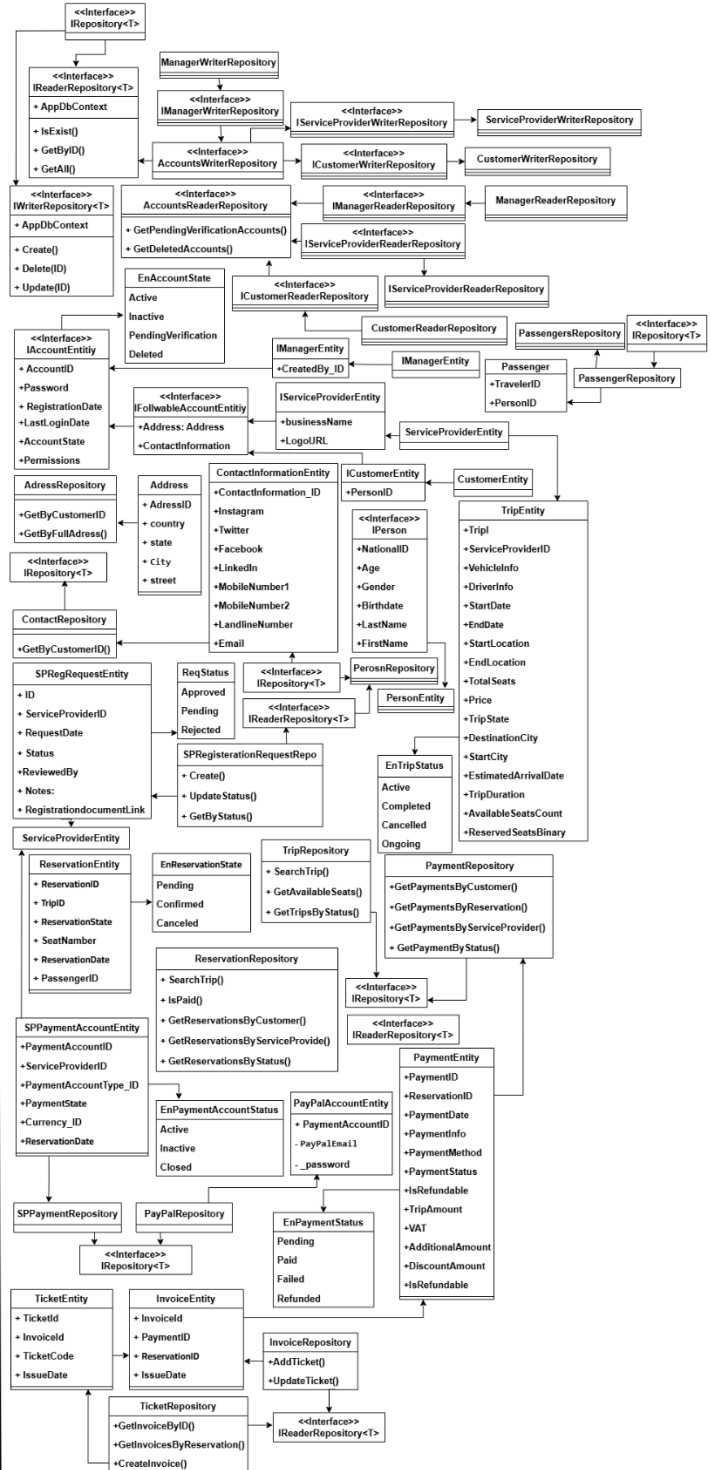
3.3 تصاميم النظام

3.3.1 UML Class Diagram (مخطط الكلاسات):

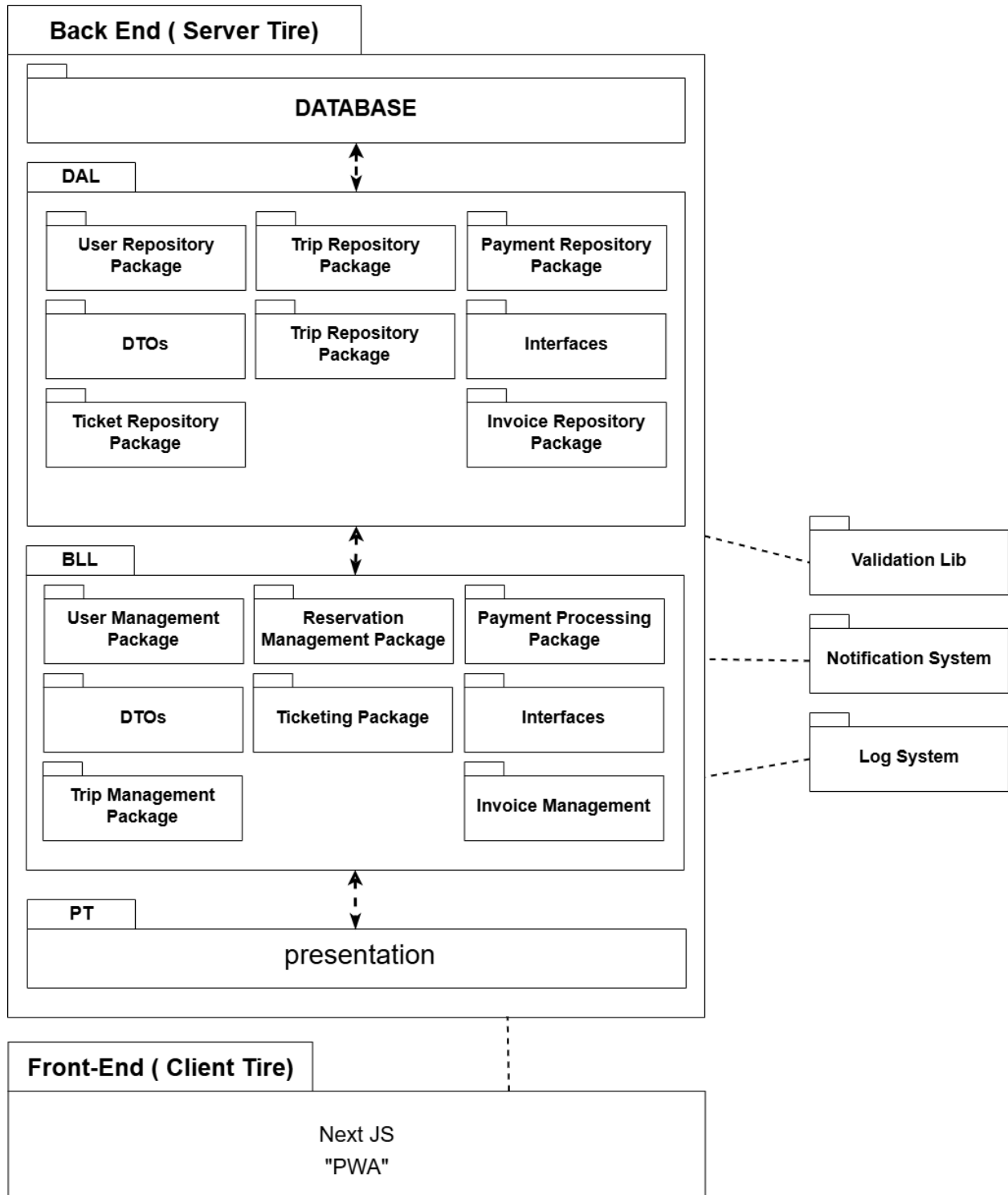
BLL



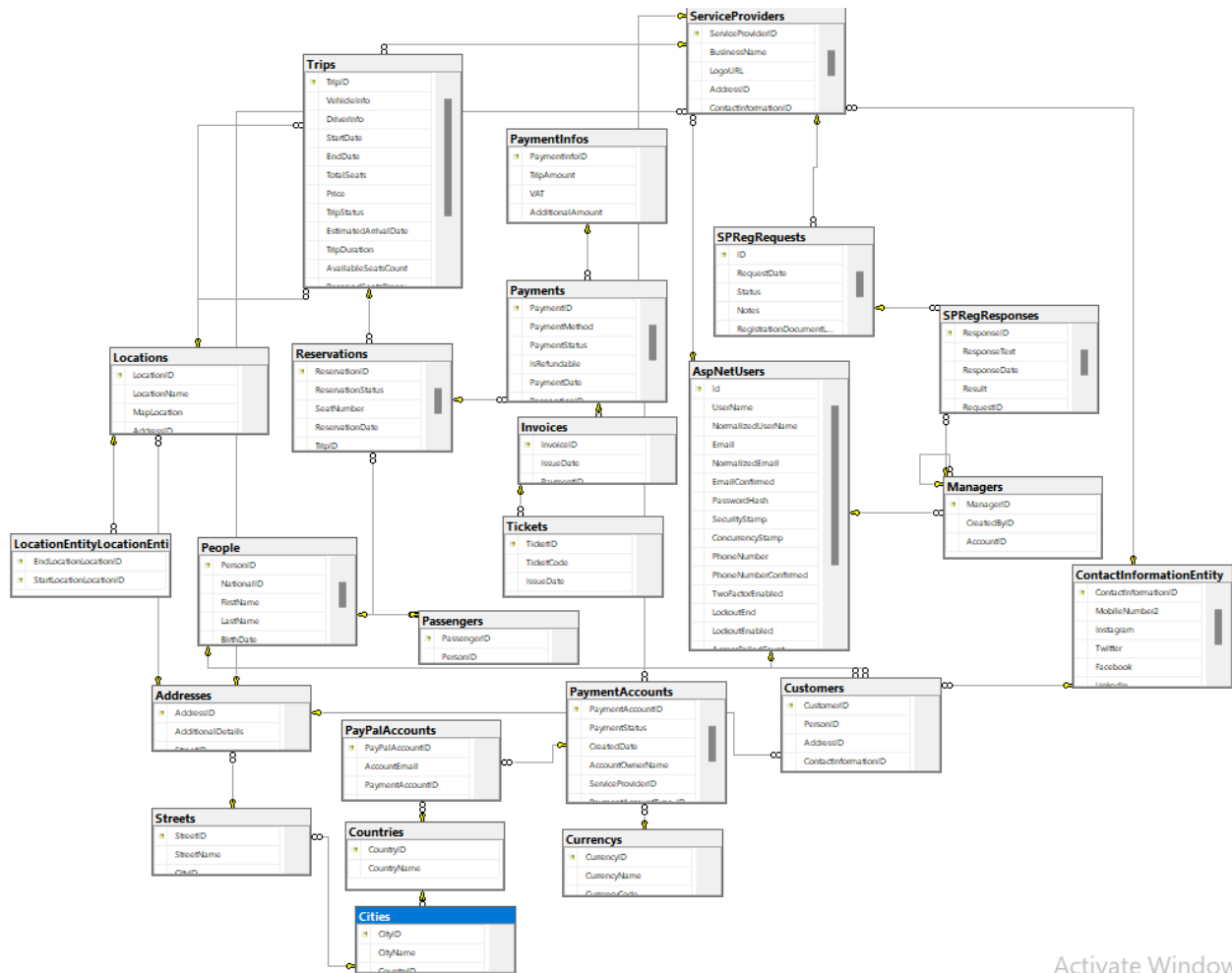
DAL



Package Diagram 3.3.2 (مخطط الحزم):



Schema Diagram 3.3.3 (مخطط قاعدة البيانات):



Activate Window

4 المنهجية

تم بناء منصة Bus E-Ticket وفق أسس تقنية حديثة تضمن تقديم نظام رقمي متكامل، مرن، وقابل للتوسع بسهولة.

4.1 SQL Database

تم اختيار Microsoft SQL Server (MSSQL) كقاعدة بيانات رئيسية في المشروع نظراً لمزاياها القوية ودعمها الشامل من قبل Microsoft، مما يجعلها متوافقة تمامًا مع بيئة .NET. يدعم MSSQL عمليات إدارة البيانات بكفاءة عالية، ويتميز بتكامله السلس مع إطار العمل Entity Framework Core [26].

تتيح هذه البيئة البرمجية تنفيذ العمليات بشكل مرن وسريع عبر أدوات مثل EF Core وADO.NET، مما يساهم في تحسين أداء النظام وتقليل زمن الاستجابة. يُعد T-SQL (Transact-SQL)، اللغة المدمجة مع MSSQL، من أبرز مميزاته، حيث تُمكن المطورين من تنفيذ استعلامات معقدة وعمليات إدارة البيانات مثل Stored Procedures وFunctions، مما يدعم تحسين الأداء عبر Query Optimization [26]. بالإضافة إلى ذلك، تساهم ميزات الأمان المتقدمة مثل Data Encryption وRole-Based Access Control (RBAC) في حماية البيانات الحساسة وضمان أمان العمليات.

يعمل Entity Framework Core كطبقة ORM (Object Relational Mapper)، مما يتيح للمطورين العمل مع البيانات دون الحاجة إلى كتابة استعلامات SQL يدويًا. كما يدعم إنشاء الاستعلامات الديناميكية باستخدام LINQ وإدارة الكيانات بشكل مرن من خلال كلاس DbContext. مع القدرة على استخدام T-SQL لاستعلامات مخصصة عند الحاجة للأداء العالي، يُحقق هذا التكامل بين EF Core وMSSQL مرونة وكفاءة كبيرة في إدارة البيانات، مما يضمن تحسين أداء النظام واستدامته [23].

4.2 الواجهة الخلفية Back-End

تعتمد الواجهة الخلفية على مجموعة من التقنيات الحديثة في إطار عمل ASP.NET Core، ويتم استخدام المكتبات وأدوات متعددة لتحقيق الوظائف التالية:

4.2.1 اللغات وأطر العمل

- #C: لغة البرمجة الأساسية المستخدمة في المشروع.
- ASP.NET Core: إطار العمل المستخدم لبناء الواجهة الخلفية وتوفير خدمات RESTful API.

4.2.1.1.1 طبقة الوصول إلى البيانات (DAL - Data Access Layer)

هي المسؤولة عن التفاعل المباشر مع قواعد البيانات وإدارة جميع العمليات المتعلقة بتخزين البيانات واسترجاعها. تُعد هذه الطبقة ضرورية لضمان عزل عمليات البيانات عن منطق الأعمال والطبقة الغليا للعرض، مما يُعزز الاستقلالية، قابلية الصيانة، والكفاءة العامة للنظام. التقنيات والمفاهيم المستخدمة :

4.2.2 Entity Framework Core (EF Core):

- إطار عمل ORM (Object Relational Mapper) يُستخدم لتسهيل التفاعل مع قواعد البيانات العلائقية من خلال الكود البرمجي.
- يتيح استخدام LINQ لكتابة الاستعلامات بطريقة برمجية دون الحاجة إلى SQL يدوي، مما يقلل الأخطاء ويسهل صيانة النظام.
- يُسهل في فصل تفاصيل قاعدة البيانات عن الكود عبر توفير طبقة تجريدية للوصول إلى البيانات.
- يعتمد على ADO.NET كطبقة منخفضة المستوى لتنفيذ الاستعلامات عند الحاجة للأداء المباشر [23].

4.2.3 Repository Pattern:

- يُستخدم نمط Repository لفصل عمليات قاعدة البيانات عن منطق الأعمال.
- يوفر طبقة وسيطة تتيح عمليات CRUD (Create, Read, Update, Delete) على الكيانات بشكل موحد ومنظم.
- يُحقق مبدأ Separation of Concerns (SoC) حيث يتم عزل الاستعلامات الخاصة بالبيانات في طبقة منفصلة، مما يسهل اختبار الكود وتعديله.
- يُعزز من إعادة الاستخدام ويقلل التكرار في عمليات قاعدة البيانات [24].

4.2.4 Unit of Work Pattern:

- يُستخدم نمط Unit of Work لإدارة عدة عمليات على قاعدة البيانات كوحدة واحدة (Transaction).
- يسمح بتنسيق العمليات المتعددة مثل الإضافة، التحديث، والحذف قبل تنفيذ عملية الحفظ النهائية، مما يضمن التكامل والموثوقية (Consistency).
- يُعزز من مبدأ التحكم في البيانات وتحقيق مبدأ Atomicity في العمليات على مستوى النظام [25].

4.2.5 خدمات الهوية والمصادقة (Authentication & Authorization)

تُعد خدمات الهوية وإدارة المستخدمين جزءًا رئيسيًا في الأنظمة الحديثة لضمان حماية الوصول إلى الموارد. ASP.NET Core Identity: إطار عمل يُستخدم لإدارة عمليات المصادقة وتفويض الأدوار. يتميز بدعم خوارزميات التشفير المتقدمة وكفاءة إدارة الصلاحيات [22].

4.2.6 طبقة الأعمال (Business Logic Layer)

طبقة الأعمال (BLL) هي الطبقة المسؤولة عن تطبيق المنطق البرمجي الخاص بالنظام، حيث تعمل كجسر بين طبقة الوصول إلى البيانات (DAL) وطبقة العرض (Presentation Layer). تُعد هذه الطبقة ضرورية لضمان تنفيذ القواعد والوظائف الخاصة بالتطبيق بطريقة مُنظمة ومستقلة.

مكونات طبقة الأعمال:

4.2.7 كلاسات الخدمات (Service Classes):

تُعتبر الكلاسات مثل Service الوسيط الذي يقوم بربط الطبقة الغليا (الواجهة الأمامية) مع طبقة الوصول إلى البيانات (DAL). تحتوي هذه الكلاسات على منطق الأعمال (Business Logic) المُختص بتنفيذ العمليات مثل إدارة البيانات، معالجة المدخلات، وتطبيق قواعد العمل.

تتفاعل الكلاسات مع الـ Repositories أو الـ Unit of Work في DAL لاسترداد أو تحديث البيانات. [23][24]

4.2.8 الواجهات (Interfaces):

يتم تعريف الواجهات (Interfaces) لفصل التنفيذ عن التعريف، مما يُساهم في تعزيز الاستقلالية وتسهيل عملية اختبار الكود. يُتيح استخدام الواجهات تطبيق مبدأ Dependency Injection (DI) لفصل التبعية بين الطبقات وتحسين القابلية للصيانة.

4.2.9 طبقة Presenter Layer

طبقة Presenter Layer هي الوسيط بين طبقة الأعمال (BLL) وواجهة المستخدم (Front-End)، حيث تعمل على تنسيق البيانات المستخرجة من الخدمات وتجهيزها للعرض. تعتمد هذه الطبقة بشكل رئيسي على RESTful APIs المبنية باستخدام إطار عمل ASP.NET Core لتبادل البيانات بين الواجهة الخلفية والواجهة الأمامية. كما تُستخدم DTOs (Data Transfer Objects) لضمان تسليم البيانات المطلوبة فقط وتقليل حجم البيانات المنقولة، مما يحسن الأداء العام للنظام. تُساهم هذه الطبقة أيضًا في تطبيق مبدأ Separation of Concerns عبر فصل البيانات ومنطق التنسيق عن منطق العرض. من الناحية التقنية، تعتمد الطبقة على Dependency Injection (DI) لربط الكلاسات بالخدمات المختلفة، مما يُسهّل الصيانة واختبار الكود. [23][24]

إضافةً إلى ذلك، تتيح الطبقة تنسيق الاستجابات مثل تنسيقات JSON لعرض البيانات بشكل منظم ومتوافق مع الواجهات الأمامية. تكمن أهمية Presenter Layer في قدرتها على تحسين تجربة المستخدم باستخدام من خلال تقديم البيانات المُجهزة والجاهزة للعرض بسرعة وكفاءة، مع تعزيز قابلية إعادة الاستخدام عبر تطبيقات متعددة مثل تطبيقات الويب والجوال. [23][24]

4.3 الواجهة الامامية Front-End

الواجهة الأمامية تمثل الطبقة التي يتفاعل معها المستخدم النهائي، وتم تصميمها باستخدام مجموعة من التقنيات الحديثة لضمان تجربة مستخدم مرنة وسهلة الاستخدام، بالإضافة إلى دعم تطبيقات الويب التقدمية (PWA)

التقنيات المستخدمة:

- **Next.js:**
إطار عمل مبني على **React.js** يقدم ميزات متقدمة مثل **Server-Side Rendering (SSR)** و **Static Site Generation (SSG)**، مما يحسن أداء التطبيق وسرعة التحميل. يدعم **Next.js** تطوير تطبيقات قابلة للتوسع، بفضل إمكانياته في إدارة التوجيه الديناميكي وتحسين محركات البحث. [27] (SEO)
- **React.js:**
مكتبة مفتوحة المصدر تُستخدم لبناء واجهات مستخدم ديناميكية وسريعة. تعتمد على **JSX** لدمج كود **HTML** مع **JavaScript**، مما يسهل كتابة الكود وتطوير المكونات القابلة لإعادة الاستخدام. [28]
- **Redux:**
أداة لإدارة حالة التطبيق (**State Management**) تُستخدم لمزامنة البيانات بين مكونات الواجهة الأمامية. تساعد في تحسين الأداء عند التعامل مع التطبيقات الكبيرة ذات الحالات المتغيرة. [29]
- **TypeScript:**
لغة برمجية تُضيف نظام أنواع (**Typing System**) إلى **JavaScript**، مما يساهم في تقليل الأخطاء وزيادة استقرار الكود. تُوفر توثيقاً واضحاً وتعزز من قابلية الصيانة. [30]
- **Bootstrap:**
مكتبة **CSS** مفتوحة المصدر تُستخدم لتصميم واجهات متجاوبة وعصرية. تُوفر مكونات جاهزة مثل الأزرار والنماذج والقوائم، مما يقلل من وقت التطوير. [31]
- **HTML و CSS:**
CSS تُستخدم لتنسيق مظهر التطبيق، بما في ذلك الألوان والخطوط والتخطيطات. [32] **HTML** تعمل كأساس هيكلي للصفحات وتُحدد عناصر المحتوى. [32]
- **JSX:**
صيغة خاصة بـ **React.js** تُستخدم لكتابة **HTML** داخل كود **JavaScript**. تُسهل عملية تطوير المكونات وتحسين قراءتها. [28]
- **PWA (Progressive Web App):**
يُضيف التطبيق خصائص التطبيقات التقدمية مثل العمل دون اتصال (**Offline Mode**)، والإشعارات الفورية (**Push Notifications**)، والتكامل مع الأجهزة. يدعم **PWA** تحسين أداء التطبيقات على الهواتف الذكية، حيث يوفر تجربة مشابهة لتطبيقات الجوال

التقليدية. يتم تحقيق ذلك عبر تقنيات مثل Service Workers لتخزين البيانات محليًا و Web App Manifest لتوفير تجربة مستخدم أصلية.[33]

5 التطبيق

6 النتائج والتوصيات

7 اضافات