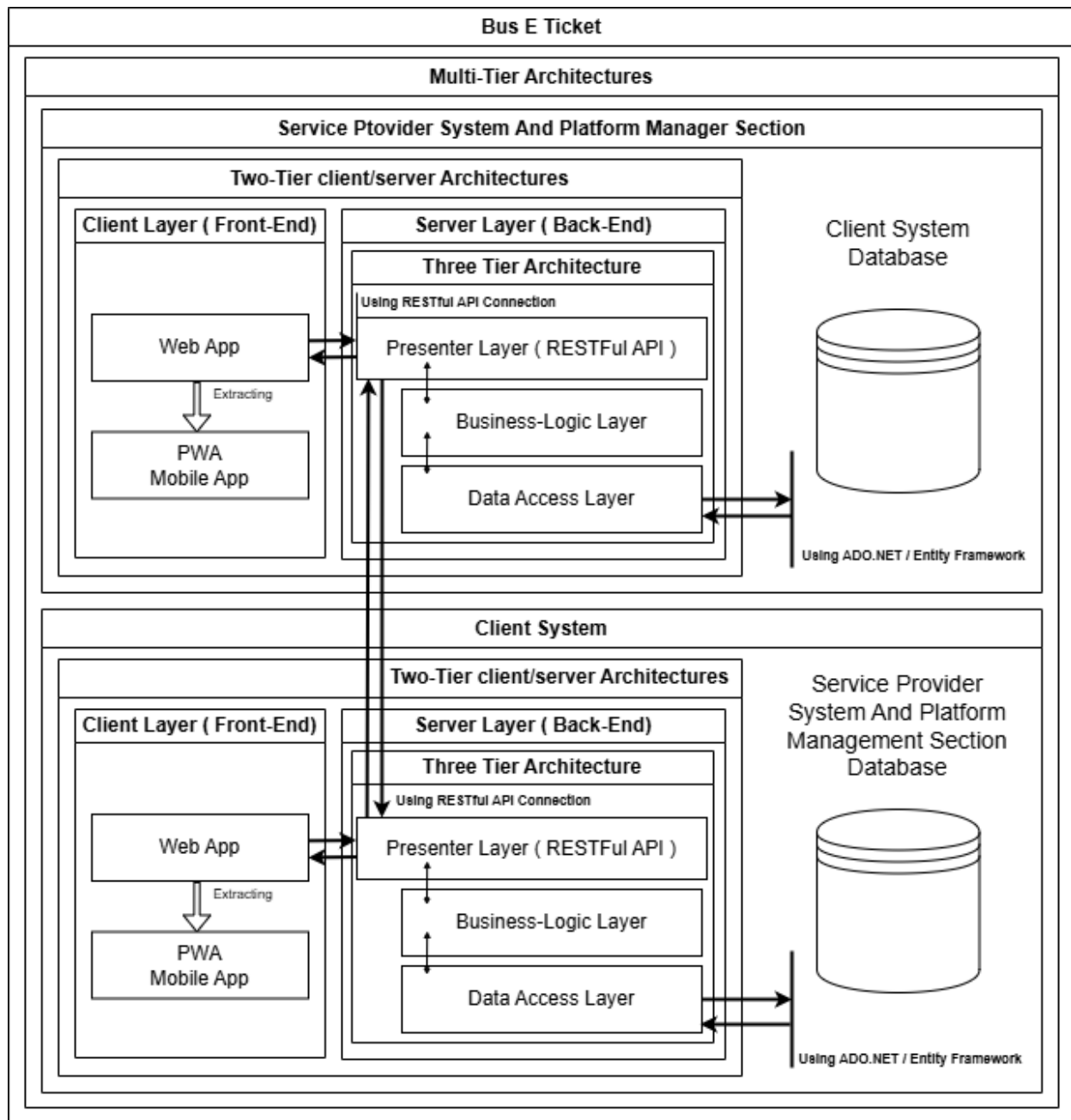


## Bus E-Ticket platform architecture design



# معماريّة نظام تذاكر الحافلات الإلكترونيّة (Bus E Ticket)

## 1- مقدمة عن النظام

النظام مقسم بشكل رئيسي إلى قسمين:

- نظام العملاء (Client System) يُستخدم من قبل العملاء (المستخدمين) لحجز التذاكر.
- نظام مزودي الخدمة وإدارة المنصة (Service Provider System and Platform Manager Section) يُستخدم من قبل الشركات المشغلة للحافلات والمديرين لإدارة الرحلات والمستخدمين. لتنظيم هذا النظام وضمان عمله بكفاءة عالية، تم استخدام معمارية متعددة الطبقات (Multi-Tier Architecture) التي تجعل من السهل إدارة كل جزء من النظام وصيانته.

## 2- تقسيم النظام إلى طبقتين رئيسيتين (Two-Tier Architecture)

يتكون النظام من طبقتين رئيسيتين، يتم استخدامهما في كل من نظام العملاء ونظام مزودي الخدمة. هذا التقسيم يُبسط عملية التواصل ويُحسن الأمان والمرونة:

- طبقة العميل (Client Layer): هي الواجهة الأمامية للنظام، حيث يتفاعل المستخدم مع الموقع أو التطبيق مباشرة.
- طبقة الخادم (Server Layer): هي المسؤولة عن معالجة البيانات وإدارة جميع العمليات الداخلية في النظام

## 3- طبقة العميل (Client Layer)

تشمل هذه الطبقة تطبيقات الويب وتطبيقات الجوال التقدمية (PWA) ، وهي التطبيقات التي يتفاعل معها المستخدم بشكل مباشر. يتم الاتصال بين هذه التطبيقات والخادم عبر واجهة (RESTful API) لنقل البيانات بين الواجهة الأمامية والخلفية بشكل آمن وفَعّال. تطبيق الويب (Web App) يسمح للمستخدمين بالوصول إلى النظام من خلال المتصفح، حيث يمكنهم حجز التذاكر واستعراض التفاصيل. تطبيق الجوال التقدمي (PWA Mobile App) هو تطبيق يشبه تطبيقات الجوال الأصلية، لكنه يعمل عبر المتصفح ويتيح تجربة شبيهة بتطبيقات الموبايل.

## 4- طبقة الخادم (Server Layer)

تحتوي هذه الطبقة على ثلاث طبقات فرعية وهيكلية ثلاثية الطبقات (Three-Tier Architecture) تساهم في تنظيم المهام داخل النظام. هذه الطبقات هي:

### 4.1 طبقة العرض (Presenter Layer)

هذه الطبقة هي واجهة برمجية API تتبع معمارية RESTful ، وهي مسؤولة عن تلقي الطلبات من طبقة العميل (تطبيقات الويب والجوال) وإعادة إرسال البيانات المطلوبة. بعبارة أخرى، هي الجسر بين التطبيقات الأمامية وقلب النظام، حيث يتم التأكد من أن الطلبات مُعالجة بشكل صحيح قبل الانتقال إلى الطبقات الأخرى.

#### 4.2- طبقة منطق الأعمال (Business-Logic Layer)

تعتبر هذه الطبقة هي العقل المدبر للنظام، حيث تتضمن جميع قواعد العمل وتقوم بتنفيذ العمليات الرئيسية، مثل التحقق من توّفر التذاكر وحساب التكلفة. طبقة منطق الأعمال هي التي تُحدد الإجراءات التي ينبغي اتخاذها لكل طلب يأتي من طبقة العرض، مما يضمن تنفيذ القواعد المحددة لكل عملية.

#### 4.3- طبقة الوصول للبيانات (Data Access Layer)

هذه الطبقة هي المسؤولة عن الوصول إلى قاعدة البيانات وإدارة البيانات. حيث يتم استرجاع المعلومات الضرورية من قواعد البيانات أو تحديثها عند الحاجة. يتم استخدام أدوات مثل ADO.NET أو Entity Framework للقيام بهذه العمليات، مما يتيح تنفيذ الاستعلامات بأمان وكفاءة.

#### 5- قاعدة البيانات (Database)

يوجد قاعدتا بيانات منفصلتان لتخزين بيانات نظام العملاء ونظام مزودي الخدمة بشكل مستقل. يتم ذلك لتحقيق الأمان وفصل البيانات حسب كل نظام. يتم في قاعدة البيانات تخزين معلومات مثل بيانات الحجز، تفاصيل المستخدمين، وجدول الرحلات. فوائد فصل قواعد البيانات:

- زيادة الأمان: فصل البيانات يمنع وصول العملاء إلى معلومات خاصة بمزودي الخدمة.
- سهولة الإدارة: فصل البيانات يجعل من السهل إدارة البيانات وحمايتها.

#### 6- التكامل بين النظامين عبر RESTful API

يتم التكامل بين نظام العملاء ونظام مزودي الخدمة عبر واجهة برمجية RESTful API ، والتي تعمل كقناة تواصل آمنة بين النظامين. هذا التكامل يسمح بتبادل البيانات بين النظامين بشكل منظم وفعال، حيث يتم إرسال واستقبال البيانات عبر بروتوكول HTTP لضمان سرعة الاتصال واستقراره.

مزايا RESTful API في المشروع:

- الأمان: يتيح التحقق من الطلبات قبل الوصول إلى البيانات، مما يزيد من أمان النظام.

- سهولة التكامل: يجعل من السهل توسيع النظام في المستقبل أو إضافة أنظمة أخرى.

#### 7- مزايا المعمارية المستخدمة في النظام

استخدام معمارية متعددة الطبقات وتطبيق فصل نظام العميل عن نظام مزود الخدمة يقدم العديد من المزايا التي تجعل هذا النظام مناسبًا للتوسع، بما في ذلك:

الأمان: النظام مصمم بحيث تكون البيانات الخاصة بكل طرف محمية ويتم التحكم في الوصول إليها.

المرونة: كل طبقة مستقلة عن الأخرى، مما يجعل من السهل تحديث أي جزء من النظام دون التأثير على الأجزاء الأخرى.

قابلية التوسع: النظام مرن بما يكفي ليتم إضافة ميزات جديدة، مثل تحسين الواجهة الأمامية، أو تغيير قاعدة البيانات.

إمكانية الصيانة: يمكن لمطوري النظام صيانة أو تعديل أي طبقة بسهولة دون إيقاف النظام بأكمله.