**الجمهورية العربية السورية**

**الجامعة السورية الخاصة**

**كلية الهندسة المعلوماتية**

**مشروع تخرج 1**

**نظام حجز التذاكر الشامل**

**Total Ticket**

**إعداد الطلاب:**

**هدى السعيد تسنيم صعب كندة العبدالله**

**إشراف المهندس:**

**شادي بليدي**

**2023**



Syrian Arab Republic

Syrian Private University

Faculty of Informatics Engineering

Graduation Project 1

**Comprehensive Booking system**

**Total Ticket**

**Prepared by**

Huda Alsaeed , Kinda Alabdullah, Tasneem saab

**Supervised by**

ENG. Shadi Blidi

**2023**

**شهادة مشرف:**

أشهد أن إعداد هذا المشروع بعنوان "نظام حجز المواصلات الشامل "، من إعداد هدى السعيد كندة العبد الله تسنيم صعب تحت إشرافي في قسم هندسة البرمجيات ونظم المعلومات / كلية هندسة الحاسوب والمعلوماتية والاتصالات وهو جزء من متطلبات نيل الإجازة في هندسة البرمجيات ونظم المعلومات

**الاسم: …………..........**

**التاريخ:**

**التوقيع:**

**شكر وتقدير Acknowledgements**

أتوجه بالشكر لكل من درسني أو ساعد في تدريسي من دكاترة الجامعة ، كما أتقدم بالشكر والتقدير للمهندس المشرف المهندس شادي بليدي على هذا البحث، والشكر موجه أيضا لإدارة الجامعة لحسن توفيرهم الخدمات للطلاب وتسهيلها ومساعدتهم في كل الأمور التي من شأنها أن تمنحهم فضاءً مريحا للدراسة وطلب العلم في نظام وأمان، كما أتوجه بالشكر لأعضاء اللجنة الأفاضل الدكتور أكرم مسوح والدكتورة ثراء أصلان ،وسعادة عميد الكلية الدكتور مهيب النقري، وفقهم الله لكل ما هو خير .

**إهداء Dedication**

إلى والدتي الغالية ووالدي الحبيب اللذان لم يتوقفا أبداً عن تشجيعي لتحقيق حلمي.

إلى أسرتي الجميلة.

إلى أصدقائي وزملائي في الجامعة السورية الخاصة،

إلى كل من شجعني أو مد لي يد العون لإنجاح هذا البحث

إلى بلدي سوريا، على أمل ان يكون هذا البحث خطوة صغيرة نحو مزيد من التقدم والتحضر لبلدنا العظيم.

# **الملخص**

تعتبر أنظمة حجز التذاكر من الأنظمة التي تشكل صلب مجتمعات المعرفة الحديثة، كون النقل عنصر أساسي ومحوري في جميع جوانب الحياة اليومية. وهي النموذج الأمثل الذي يوضح استخدام التكنولوجيا الحديثة في تحسين حياة الناس، وخاصة أن عملية الحجز والدفع التقليدية في العالم عموماً وفي سوريا خصوصاً تشكل عبأً غير ضرورياً ، فالعملية التي تتم يدوياً تتطلب إجراءات وطوابير انتظار طويلة وتفتقر إلى السرعة والكفاءة، لذلك تم استخدام جميع التقنيات المتاحة في هذا البحث لتوفير نظام يطبق مفهوم الحجز الإلكتروني المتكامل في مختلف شركات النقل.

حيث يمكن لنظام المسافرين من عملية الحجز وإتمام الدفع من خلال نظام المدفوعات الإلكترونية ، بكل سهولة من خلال رقم يتم إنشاؤه من عملية الحجز يستخدمه مستخدمو النظام للدفع ، كما يوفر إمكانية التطور مع النظام من خلال إضافة تقييمات للرحلة من قبل مستخدمي النظام.

كل هذه الخدمات تسعى إلى التطور أنظمة الحجز الإلكتروني لتلبي الاحتياجات في سوريا من خلال الاستفادة من تقنيات البرمجيات والأدوات المتوفرة لدينا، وقد تم استخدام التكنولوجيا التي تناسب النظام حيث تم تصميم الواجهات من خلال react frameworks ، والبرمجة الخلفية باستخدام Fast API ، بحيث يكون النظام قابل للتطوير وأجزاءه قابلة لإعادة الاستخدام.

# **Abstract**

Ticketing systems are among the systems that form the core of modern knowledge societies, as transportation is an essential and pivotal element in all aspects of daily life. It is the ideal model that illustrates the use of modern technology in improving people's lives, especially since the traditional reservation and payment process in the world in general and in Syria in particular constitutes an unnecessary burden, as the process that takes place manually requires procedures and long waiting queues and lacks speed and efficiency, so all available technologies were used In this research to provide a system that applies the concept of integrated electronic reservation in various transport companies.

Where the travelers system can process the reservation and complete the payment through the electronic payment system, easily through a number generated from the reservation process used by the users of the system to pay, and also provides the possibility of development with the system by adding ratings for the trip by the users of the system.

All these services seek to develop electronic reservation systems to meet the needs in Syria by taking advantage of the software technologies and tools available to us, and the technology that suits the system has been used as the interfaces are designed through react frameworks, and back-end programming using Fast API, so that the system is scalable Its parts are reusable.

# **جدول المحتويات:**

Contents

[**الملخص** 5](#_Toc136040089)

[**Abstract** 6](#_Toc136040090)

[**جدول المحتويات:** 7](#_Toc136040091)

[**الفصل الأول** 10](#_Toc136040092)

[**مقدمة** 10](#_Toc136040093)

[**مقدمة عن المشروع:** 11](#_Toc136040094)

[**المشكلة:** 12](#_Toc136040095)

[**هدف البحث:** 12](#_Toc136040096)

[**الفصل الثاني** 13](#_Toc136040097)

[**موضوع البحث والدراسات المرجعية** 13](#_Toc136040098)

[**المفاهيم الأساسية:** 14](#_Toc136040099)

[**نظرية النظم:** 14](#_Toc136040100)

[**التعريف بنظم حجز التذاكر (بالإنجليزية: Computer Reservation System أو CRS)‏:** 15](#_Toc136040101)

[**واجهة برمجة التطبيقات Application Programming Interface API :** 15](#_Toc136040102)

[**الدفع الإلكتروني** 15](#_Toc136040103)

[**الدراسة المرجعية:** 16](#_Toc136040104)

[**أنماط نظم حجز التذاكر:** 16](#_Toc136040105)

[**النظام المقترح:** 18](#_Toc136040106)

[**أقسام النظام :** 19](#_Toc136040107)

[**3- قسم إدارة النظام:** 19](#_Toc136040110)

[**الفصل الثالث** 20](#_Toc136040111)

[**الدراسة التحليلة للنظام المقترح** 20](#_Toc136040112)

[**مقدمة:** 21](#_Toc136040113)

[**أولاً: دراسة الجدوى الاقتصادية:** 21](#_Toc136040114)

[**ثانياً: مخطط المشروع:** 23](#_Toc136040115)

[**ثالثاً: وثيقة المتطلبات:** 24](#_Toc136040116)

[**نمذجة المتلطلبات :** 25](#_Toc136040117)

[**مخطط حالات الاستخدام: use case Diagram** 26](#_Toc136040118)

[المخططات التفصيلية لحالات الاستخدام 28](#_Toc136040119)

[**مصفوفة تتبع النظام RTM:** 48](#_Toc136040120)

[**معمارية النظام:** 50](#_Toc136040121)

[**الفصل الرابع** 51](#_Toc136040122)

[**الدراسة التصميمية للنظام المقترح** 51](#_Toc136040123)

[**مخطط الصفوف:** 52](#_Toc136040124)

[**مخطط sequence للعملية الأساسية في النظام** 53](#_Toc136040125)

[**واجهات النظام:** 54](#_Toc136040126)

[**الفصل الخامس** 62](#_Toc136040127)

[**التطبيق العملي** 62](#_Toc136040128)

[**إعداد مخدم النظام API Server :** 63](#_Toc136040129)

[**المتطلبات والحزم البرمجية الاساسية المطلوبة لبرمجة المخدم:** 63](#_Toc136040130)

[**خطوات برمجة مخدم الويب الخاص بالتطبيق:** 63](#_Toc136040131)

[**ثانياً: إعداد بيئة العمل:** 64](#_Toc136040132)

[**ثالثا: تعريف النماذج Models التي نحتاجها في النظام :** 67](#_Toc136040133)

[**رابعاً تجهيز نماذج الاتصال مع قاعدة البيانات لاستخدامها ضمن توابع ال Models APIs:** 68](#_Toc136040134)

[**خامساً: توابع APIs / controllers** 69](#_Toc136040135)

[**الفصل السادس** 71](#_Toc136040136)

[**تحليل النتائج:** 72](#_Toc136040138)

[**الآفاق المستقبية:** 72](#_Toc136040139)

[**الفصل السابع | الخاتمة** 7](#_Toc136040136)4

# **الفصل الأول**

# **مقدمة**

## **مقدمة عن المشروع:**

تعتبر المواصلات العامة من أهم البنى التحتية التي تعكس تطور البلد وتقدمه، وكلما كانت هذه الخدمة مقدمة بشكل أفضل، بما يحقق راحة المسافرين، كلما كان ذلك مساعداً في الاستثمار الأمثل للنواحي الاقتصادية والسياحية لهذا القطاع.

ولعل أبرز الأساليب التي تساعد المسافرين في تحقيق مستوى عالي من الراحة الجسدية والنفسية، هي القدرة على حجز رحلاتهم بشكل مسبق، وخاصة عبر الأساليب الرقمية، بحيث يتمكن المسافر من خلال هاتفه المحمول أو حاسبه الشخصي من استعراض الرحلات من مكان ما إلى وجهة ما واختيار الأنسب من حيث التكاليف والزمن والخدمات المقدمة.

ومن هنا برزت أهمية منصات حجز التذاكر الإلكترونية، والتي حققت صلة وصل سريعة ما بين شركات النقل وعملائها بما يحقق ذلك من فائدة جمة للطرفين، فمن جهة تتيح للشركة من استعراض خدماتها على أوسع نطاق وبأسهل طريقة، مما يحقق لها مكاسب مادية إضافية، ومن جهة ثانية تتيح للمستخدمين تجربة أفضل في اختيار رحلاتهم بما يحقق غاياتهم.

## **المشكلة:**

تكمن المشكلة الأساسية التي يسعى النظام إلى حلها، هي بإيجاد معمارية معينة تلائم مختلف أنواع وسائط النقل، والتي هي على بساطتها لا يمكن أن تكون بلون واحد، ناهيك عن المعضلة المتعلقة بآلية العمل اللوجستية الخاصة بكل شركة، كأن يكون هناك رحلات مجدولة، ورحلات غير مجدولة ( استثنائية) وغير ذلك سيعرض في الأقسام اللاحقة.

## **هدف البحث:**

يهدف البحث إلى التوصل إلى البنى التحليلية والتصميمية لنظام حجز تذاكر يكون شامل، لمختلف شركات النقل والتي تخدم المسافر أياً كانت احتياجاته من وسيلة النقل، وتحقيقه من خلال واجهات تخاطبية تحسن من تجربة المستخدم، ومن خلال برمجة مخدمات API تؤمن قاعدة للتطوير النظام ليتماشى مع مختلف أنواع الأجهزة والنظم.

# **الفصل الثاني**

# **موضوع البحث والدراسات المرجعية**

## **المفاهيم الأساسية:**

### **نظرية النظم:**

تأتي نظرية النظم في إطار النظريات الحديثة التي تقوم على أساس نقد النظريات السابقة سواء التقليدية أو السلوكية لأن كل منهما ركز على أحد متغيري التنظيم (العمل والإنسان) باعتبار أن التنظيم نظام مقفل، بينما يرى للتنظيم في نظرية النظم إلى أنه نظام مفتوح يتفاعل مع البيئة المحيطة به وذلك ضمانا لاستمرارية التنظيم.

إن دراسة أي تنظيم لابد أن تكون من منطق النظم، بمعنى تحليل المتغيرات وتأثيراتها المتبادلة. فالنظم البشرية تحوي عددا كبيرا من المتغيرات المرتبطة ببعضها، وبالتالي فنظرية النظم نقلت منهج التحليل إلى مستوى أعلى مما كان عليه في النظرية الكلاسيكية والنظرية السلوكية، فهي تتصدى لتساؤلات لم تتصدى النظريتين السابقتين.

تقوم هذه النظرية على أجزاء يتكون منها النظام لها علاقة وثيقة ببعضها البعض. هذه الأجزاء هي:

1. إن الجزء الأساسي في النظام هو الفرد (قائدا أو منفذا) وبصفة أساسية التركيب السيكولوجي أو هيكل -الشخصية الذي يحضره معه في المنظمة. لذا فمن أهم الأمور التي تعالجها النظرية حوافر الفرد واتجاهاته وافتراضاته عن الناس والعاملين.
2. إن الجزء الأساسي الثاني في النظام هو الترتيب الرسمي للعمل أو الهيكل التنظيمي وما يتبعه من المناصب
3. إن الجزء الأساسي الثالث في النظام هو التنظيم غير الرسمي وبصفة خاصة أنماط العلاقات بين المجموعات وأنماط تفاعلهم مع بعضهم وعملية تكييف التوقعات المتبادلة.
4. الجزء الأساسي الرابع في النظام هو تكنولوجيا العمل ومتطلباتها الرسمية. فالآلات والعمليات يجب تصميمها بحيث تتمشى مع التركيب السيكولوجي والفسيولوجي للبشر.

### **التعريف بنظم حجز التذاكر (بالإنجليزية: Computer Reservation System أو CRS)‏:**

نظام الحجز الإلكتروني هو نظام محوسب يستخدم لتخزين والاستعلام والقيام بعمليات الحجز للسفر. استخدمته في البداية شركات الطيران، ثم امتد فيما بعد لتستخدمه وكالات السفر. تعرف نظم الحجز الإلكتروني الكبيرة التي تبيع تذاكر السفر لعدة شركات طيران باسم نظم التوزيع العالمي (بالإنجليزية: Global Distribution System أو GDS)‏. منحت شركات الطيران صلاحية الحجز الإلكتروني لشركات التوزيع العالمي المتخصصة، التي تتيح للعملاء عملية الحجز عبر الإنترنت. وتتيح نظم التوزيع العالمي الحديثة حجز غرف الفنادق وتأجير السيارات بالإضافة إلى تذاكر السفر، وتذاكر القطارات في بعض الدول.

### **واجهة برمجة التطبيقات Application Programming Interface API :**

هي واجهةً برمجية تسمح لتطبيقان بالتفاعل مع بعضهما دون أي تدخل من المستخدم. فهي عبارة عن مجموعة من الوظائف والاجراءات يتمْ تعريف رمز البرنامج الذي يمكن الوصول اليه او تنفيذه مما يعطي اختصار حركة البيانات بين بعضهما. وتقدم ميزة الى العميل من خلال نقل بيانات معينة ضمن الخادم دون الحاجة الى معرفة كيفية تنفيذها. ( سيتم تفصيلها بشكل أوسع في القسم العملي من تنفيذ النظام) .

### **الدفع الإلكتروني**

بدأ ظهور مفهوم الدفع الإلكتروني وحاجته الملحة بعد ظهور التجارة الإلكترونية. تعتبر عملية الدفع الإلكتروني التي تًعرف online payment عملية مالية متكاملة من النظم والبرامج يتم تقديمها لتسهيل الإجراءات المالية عبر الإنترنت. إن الدفع عبر الإنترنت هو عبارة عن صرف إلكتروني للعملة لشراء السلع أو الخدمات.

تعمل هذه العملية وفقًا لقوانين تضمن سرية وخصوصية إجراءات البيع والشراء. تحدث بشكل إلكتروني عن طريق الانترنت، وذلك من خلال استخدام نظام دفع آمن و معلومات مشفرة مطبوعة بطاقتك وتفاصيل سرية مختلفة. تختلف الشركات والمواقع المهتمة في هذا المجال باختلاف نظرتها المالية.

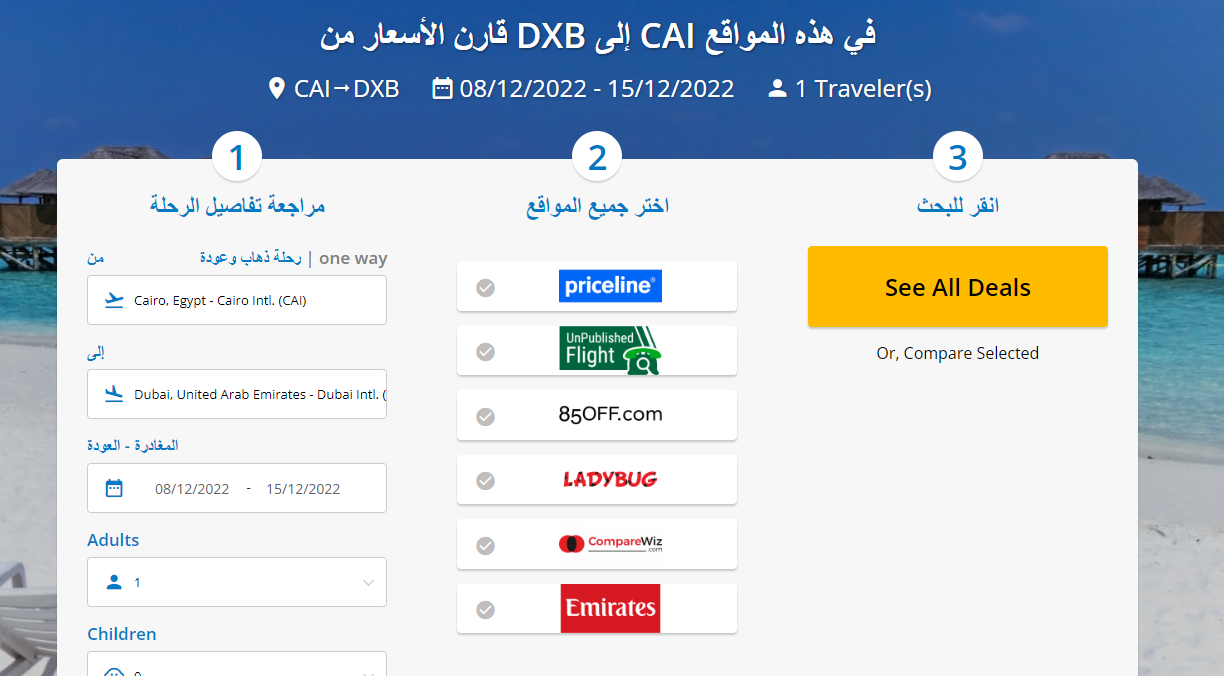
## **الدراسة المرجعية:**

### **أنماط نظم حجز التذاكر:**

* النظم المستقلة: وهي النظم الحاسوبية التي تعمل بشكل منفصل عن شركات المواصلات، في لا تقوم بعملية الحجز، وانما باستعراض الرحلات والتكلفة والزمن، أي انها وسيط إعلامي للرحلات، تعتمد على API من كل شركة، وتسمح للمستخدم باستعراض هذه الرحلات وتفاصيل متعلقة فيها، من ابرز هذه المواقع: save70 :

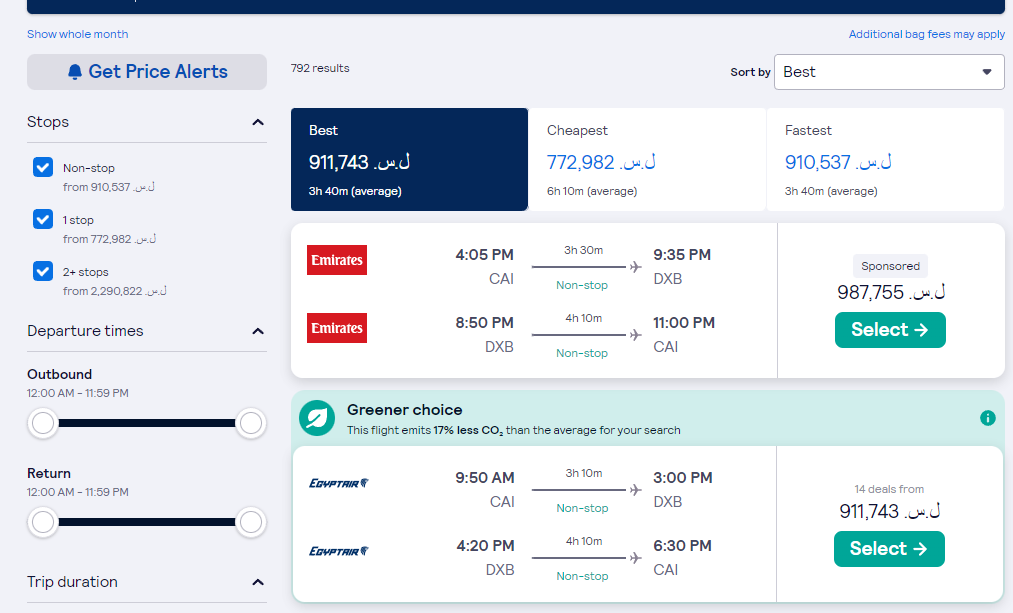


النتائج: لم يجد الموقع رحلات إلى دمشق في الوقت الراهن، وعند تغيير البحث (من القاهرة إلى دبي) يقدم الموقع روابط لصفحات التي تقدم الحجز وفقاً لأفضل سعر:

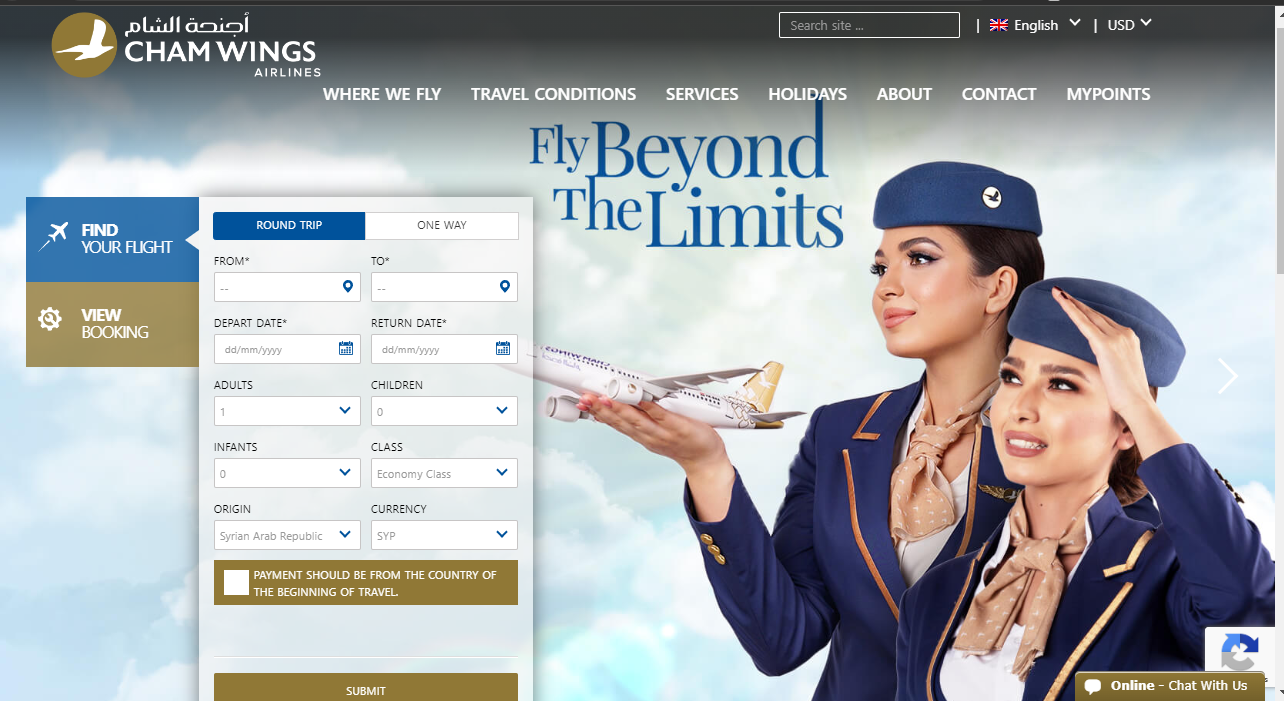


Skyscanner

وهو موقع عالمي للاستعلام عن رحلات الطيران، وحجوزات الفنادق وتأجير السيارات.



النظم التابعة: وهي النظم أو المواقع التي ترتبط بشركات طيران معينة، وفيها يتم إضافة الرحلات من قبل موظفي الشركة، ويمكن للمستخدم من خلالها الاستعلام عن الرحلات والحجز والدفع الإلكتروني للرحلة. كما في الموقع الخاص لشركة أجنحة الشام السورية.



## **النظام المقترح:**

وفي هذا المشروع سنقوم بتصميم نظام حجز التذاكر أقرب إلى النمط الثاني، بحيث يكون منصة خاصة بحجز التذاكر وفقاً لشركات محددة، تتفاعل مع النظام عن طريق إدارة قسم خاص بالشركة يمكن من خلاله (إضافة، تعديل، إدارة ) جداول الرحلات، وفي المقابل يمكن للمستخدم ( العميل) التفاعل مع النظام بشكل مستقل، بهدف إتمام عملية الحجز بشكل كامل دون الحاجة للتواصل مع الشركة. وسنقوم بتصميم النظام على أساس العمل مع الشركات المحلية في سوريا، ( القدموس، زريق، السورية للطيران) وسنعمل على إضافة خدمات الدفع الإلكتروني المتاحة حالياً في سوريا، ك Syriatel Cash أو الحوالات البنكية.

## **أقسام النظام :**

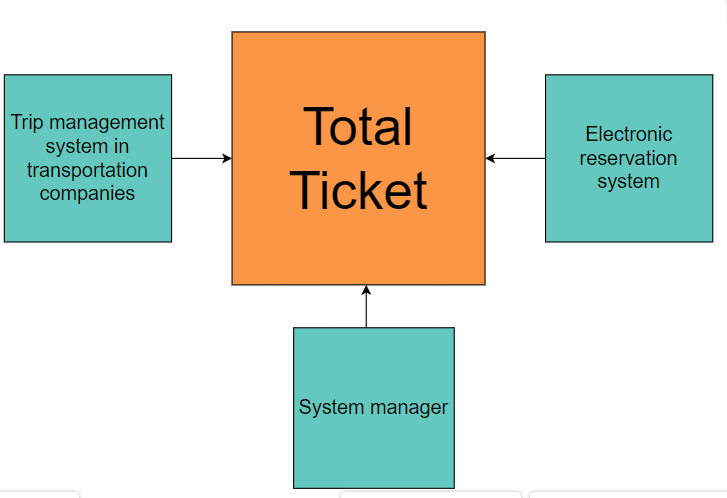
1. **نظام إدارة الرحلات في شركات النقل**

يتفاعل مع هذا القسم موظفو شركات النقل عموماً، بحيث يتمكن الموظف في هذا القسم من إضافة رحلة (مزودة بمكان الانطلاق، الوجهة، توقيت الانطلاق، توقيت الوصول المتوقع، عدد الركاب المتاحين، سعر التذكرة) وغيره من المعلومات الخاصة بالرحلة، كما يمكنهم استثمار النظام من خلال (استعراض الرحلات وأجراء التعديلات عند الضرورة، استعراض أسماء المسافرين وبياناتهم. أجراء استعلامات خاصة "معرفة عدد الحجوزات المتاحة، أجمالي أرباح الرحلة، التوقيت الأكثر ازدحاماً، كما يمكنهم إضافة إعلانات عن الشركة، تدرج في الصفحة الخاصة للشركة.

1. **نظام الحجز الإلكتروني**

يمكن للمستخدم الوصول إلى هذا النظام، بهدف استعراض الرحلات الممكنة من مكان محدد إلى مكان آخر، وفق إطار زمني معين، ومقارنة الأسعار، وعلى هذا الأساس يقوم بعملية الحجز الإلكتروني للرحلة، كما يمكنه من استعراض تقارير شخصية (عدد الرحلات التي قام بها، المبالغ التي دفعها)

## **3- قسم إدارة النظام:**

تتركز مهمة هذا القسم حالياً في عملية أنشاء حسابات خاصة للشركات، وقد يكون له دور بارز في الآفاق المستقبلية في حال أردنا استثمار النظام.

# **الفصل الثالث**

# **الدراسة التحليلة للنظام المقترح**

**مقدمة:**

سنقوم في هذا الفصل باستعراض المخططات الأساسية التي تم اعتمادها في المراحل الأولى من تصميم النظام، بما في ذلك الطرق والأساليب المتبعة في تحليل متطلبات النظام والوثيقة المقترحة وفقاً للبنود الأساسية، بالإضافة إلى استعراض دراسة الجدوى الاقتصادية، ووضع مصفوفة تتبع المتطلبات RTM، وفي النهاية نحدد مخطط معمارية النظام مع شرح موجز.

**أولاً: دراسة الجدوى الاقتصادية:**

نقصد بدراسة الجدوى الاقتصادية مجموعة الاختبارات والتقـديرات التي يتم إعدادها بنية الحكم على صلاحية المشروع الاستثماري المقتـرح، أو القرار الاستثماري و ذلك على ضـوء توقعـات التكـاليف و الفوائد المباشرة وغير المباشرة، وذلك طوال العمر الافتراضي للمشروع.

ويمكن القول بأنه يتم في هذه الدراسة تحليل وقياس القدرة المالية للمشروع، واحتمالية استمراره بنجاح، بما في ذلك جميع العوامل التي يمكن أن تؤثر عليه، مثل: العوامل الاقتصادية، والتكنولوجية، والقانونية.

ومن جهة أخرى، فإن دراسة الجدوى الاقتصادية تستخدم لتحديد النتائج الإيجابية والسلبية المحتملة للمشروع قبل استثماره، والتأكيد على المشاكل المحتملة، وتحديد إذا كانت فكرة المشروع فكرة جيدة أم لا بعد النظر إلى جميع العوامل الهامة.

وقبل الخوض في دراسة الجدوى الاقتصادية للنظام، سنستعرض التكلفة التي تفرض على شركات النقل التي لا تعتمد على الأتمتة في عملية حجز التذاكر، فهي تحتاج إلى ما يقارب اذا افترضنا أن سعر التذكرة الواحدة ما يقارب 0.03 دولار امريكي او ما يعادل 500 ليرة سورية فإنها يومياً بحاجة إلى 50000 ليرة سوريا ثمن تذاكر ل 100 شخص، أي تقريباً كل ثلاث رحلات، وبالتالي مليون ونصف ليرة سورية شهرياً تكاليف طباعة هذه التذاكر، والتي ترتفع وفقاً لأسعار الصرف، في حين ان تكلفة استضافة خاصة بالنظام، مع تكاليف بنائه والتي تدفع مرة واحدة ستوفر ما يقارب ثلثي هذا المبلغ على المدى المتوسط فقط.

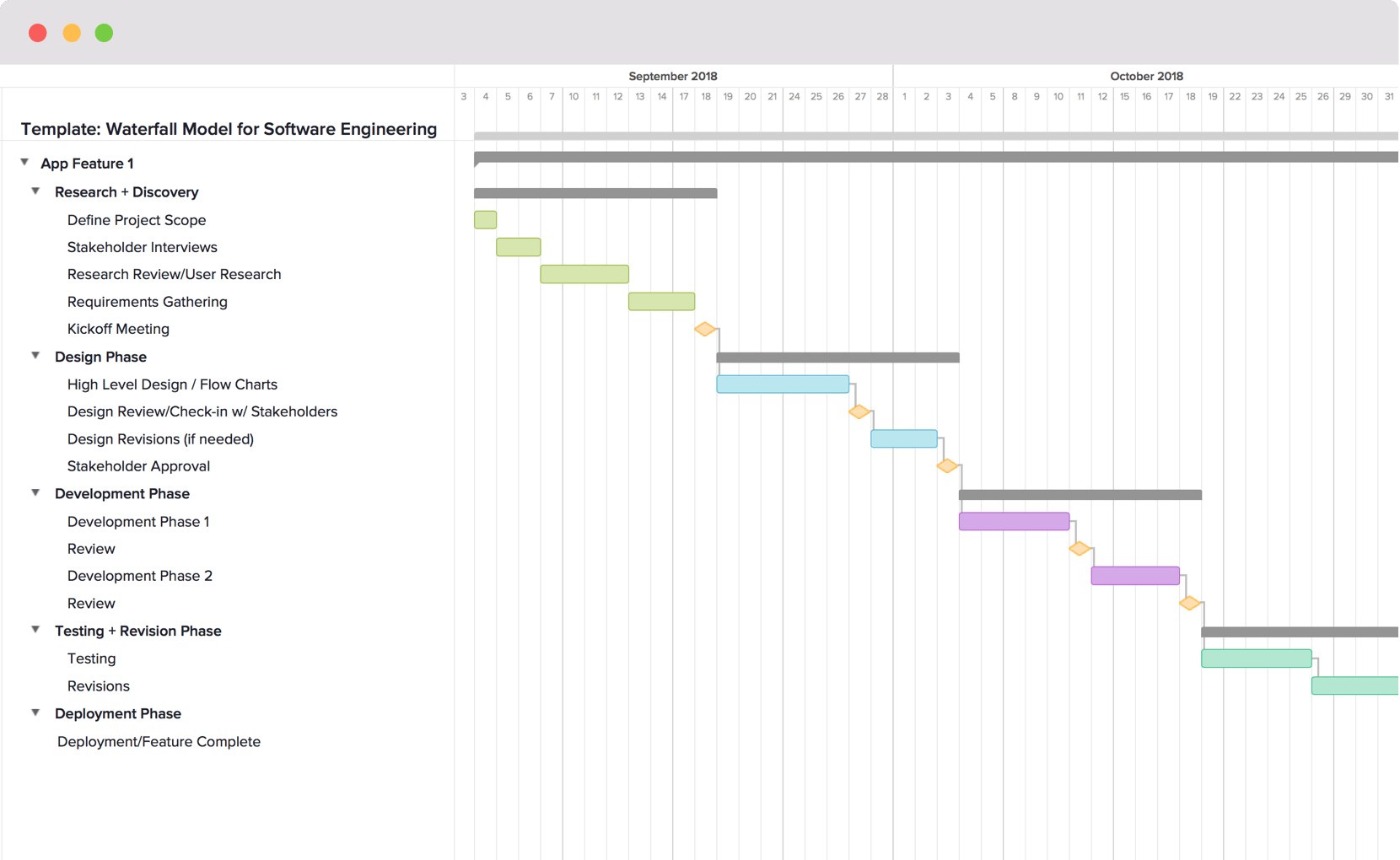
وللخوض أكثر عن التكاليف المحتملة لتنفيذ النظام بشكل احترافي مع تكلفة تشغيله كالتالي:

|  |  |  |
| --- | --- | --- |
| الغرض | التكلفة | النمط |
| تكلفة بناء النظام | 1000$ | مرة واحدة تدفع لفريق المهندسين |
| تكلفة صيانة النظام | 100$ | تدفع سنوياً |
| تكلفة استضافة النظام | 100$ | تدفع سنوياً |
| تكلفة حجز الدومين | 10~ 20 $ | تدفع سنوياً. |
| التكلفة النهائية | 1000$ او ما يعادل 8 مليون ليرة سورية لمرة واحدة | 220$ تدفع سنوياً أو ما يعادل مليوني ليرة سورية |

وبالتالي خلال السنة الأولى يكلف المشروع تقريباً 10 مليون ليرة سورية، في حين ان تكلفة الأوراق المطلوبة في النموذج القديم تكلف أكثر من 15 مليون ليرة، بجدوى اقتصادية تصل إلى 5 مليون في السنة الأولى، وترتفع إلى 12 مليون في السنة الثانية وما بعدها.

## **ثانياً: مخطط المشروع:**

لقد تم اعتماد منهجية النموذج الشلالي في عملية تطوير النظام، وذلك كون النظام يتمتع بمتطلبات شبه ثابتة من جهة، كما انه يعتبر من المشاريع المتوسطة والتي يمكن التعويل على النماذج التحضير المسبق في هندسة البرمجيات،



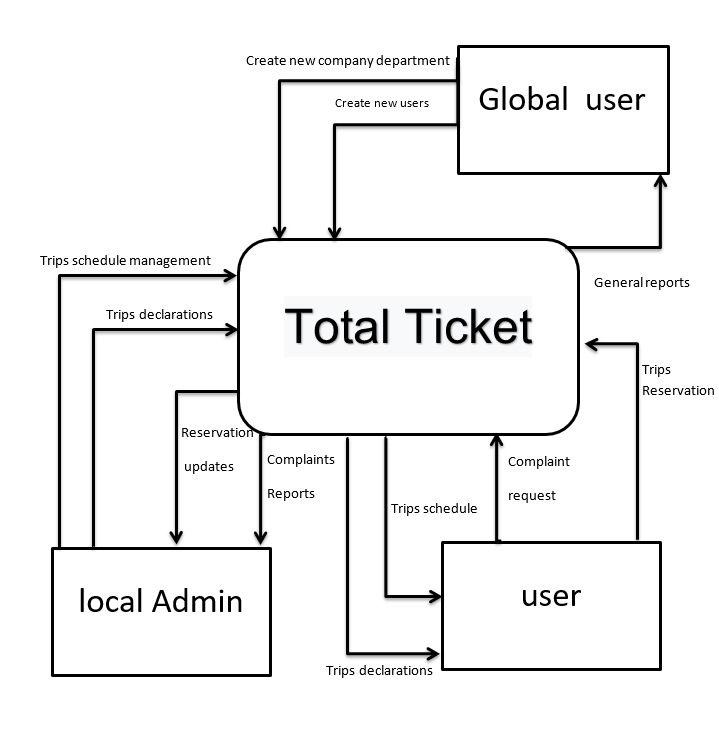
وفر نموذج الشلال نهج منظم؛ النموذج نفسه تقدم خطيا من خلال مراحل منفصلة، يسهل فهمها وتفسيرها، وبالتالي من السهل أن نفهم. كما يوفر المعالم التي يسهل التعرف عليها في عملية التنمية. ولعله لهذا السبب نموذج الشلال يستخدم كنموذج بداية لنموذج التنمية في العديد من النصوص هندسة البرمجيات والدورات.

**ثالثاً: وثيقة المتطلبات:**

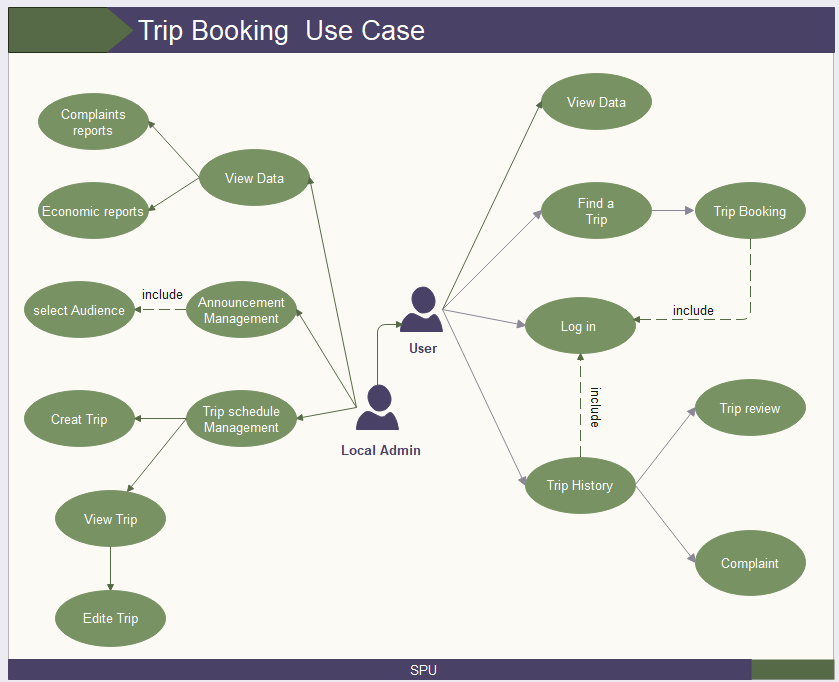
|  |  |  |  |
| --- | --- | --- | --- |
| رمز المتطلب | نوع المتطلب | الفاعل actor | الشرح |
| R1 | وظيفي | المدير المحلي للشركة | يجب ان يكون المدير المحلي للشركة قادر على إدارة أسطول الشركة من السيارات او الطائرات او الباصات، وذلك من خلال اتاحة ( إضافة – تعديل – حذف- استعراض الأسطول\* |
| R2 | وظيفي | المدير المحلي للشركة | يجب أن يكون المدير المحلي قادر على إدارة جدول الرحلات الخاصة بالشركة (إضافة، تعديل، استعلام) |
| R3 | وظيفي | المدير المحلي للشركة | يجب ان يكون المدير المحلي قادر على استعراض طلبات الحجز والتأكد من سلامة عملية الحجز (التحقق من الدفع وصحة معلومات العملاء) |
| R4 | وظيفي | المدير المحلي للشركة | يجب أن يكون المدير المحلي قادر على استعلام عن تقارير خاصة (على مستوى الشركة) كعدد الحجوزات في رحلة معينة، ورأي العملاء فيها |
| R5 | وظيفي | المدير المحلي للشركة | يجب أن يكون المدير المحلي قادر على تلقي شكاوي العملاء، وتحويلها إلى القسم المختص في الشركة. |
| R6 | وظيفي | العميل | يجب على المستخدم أن يكون قادر على الاستعلام عن رحلات من خلال عدد من المدخلات (مكان الانطلاق والوجهة، الزمن .. الخ) |
| R7 | وظيفي | العميل | يجب على المستخدم أن يكون قادر على الحجز على الرحلة وإتمام عملية الدفع. |
| R8 | النظام | وظيفي | يجب على النظام إرسال إشعارات حول التغيرات التي من المحتمل ان تطرأ على الرحلة. |
| R9 | العميل | وظيفي | يجب على النظام أن يسمح للعميل بإرسال شكوى بعد القيام برحلة ما |
| R10 | النظام | غير وظيفي  **الاداء** | أن التطبيق يجب أن يكون متاحاً على الإنترنت عبر تطبيقات الويب. |
| R11 | النظام | غير وظيفي **الاداء** | يجب أن يعمل النظام بأفضل وقت استجابة. |
| R12 | النظام | غير وظيفي **الحماية** | يجب على النظام تشفير جميع المعلومات الخاصة بالعملاء في قاعدة البيانات |
| R13 | النظام | غير وظيفي **الاتاحة** | الحماية من الهجمات التي تهدد الإتاحية من قبل المستخدمين ( ضبط عدد الاستعلامات من قبل المستخدم وخاصة التي تتطلب عمليات حساب معقدة، كالبحث عن الرحلات الأقرب من مكان معين، أو عمليات ترتيب المستخدم. |
| R14 | النظام | غير وظيفي **قابلية الاستخدام** | - يجب ان تصمم الواجهات بأبسط شكل، والتركيز على الوظيفة التي يؤديها بالنسبة للمستخدم |

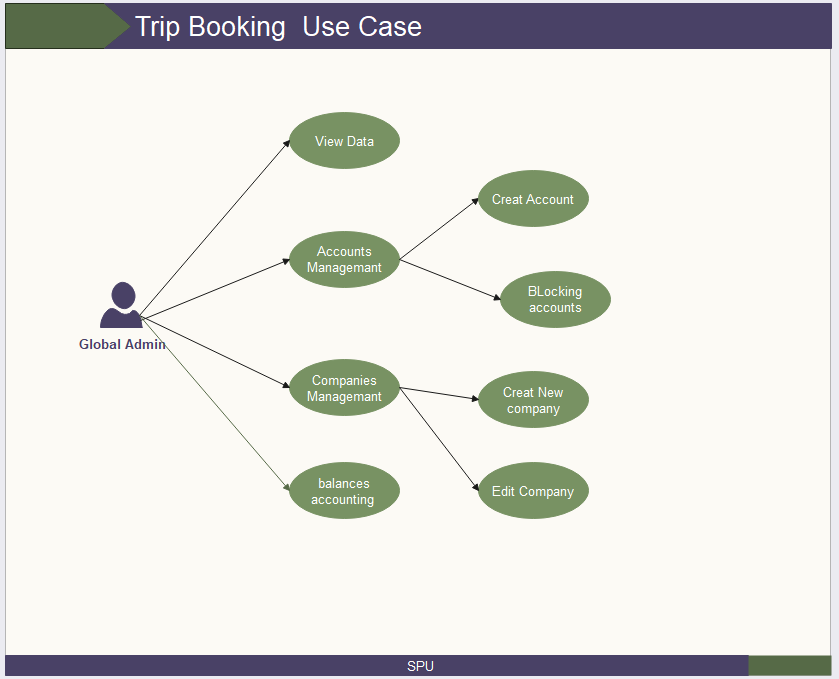
**نمذجة المتطلبات :**

يوضح مخطط context diagram التالي كامل النظام كعملية واحدة، ويبين الاطراف المدخلة ونتائجها.



## **مخطط حالات الاستخدام: use case Diagram**





|  |  |
| --- | --- |
| Use Case: log in | |
| 1 | **ID** |
| The user log in to the system | **Brief description** |
| Global Admin  Local Admin  user | **Primary Actors** |
| - | **Secondary actor** |
| - | **Preconditions** |
| 1. The user goes the website 2. The user Clicks on log in button 3. The user Enters his/her user mail and password 4. The system searches in the database for such user information   If matched system will redirect the user to the main page that prepared with his rules  Else:  The system will ask the user to retype his information | **Main flow** |
| The system will generate a user’s token and store it on Database and user’s cookies | **Post condition:** |
| Alt 1:   1. The use goes the website 2. The user Clicks on find a trip 3. The user search for a trip 4. The user clicks on reserving trip 5. If the user does not logged in, the system will redirect him to the log in page.   Alt 2:   1. The use goes the website 2. The user Clicks on log in button 3. The user Clicks on forget password button 4. The user Enters his/her user e-mail   If matched system will send the user message with code to reset his password   1. The user enter the code and reset his password | **Alternative flow:** |

## المخططات التفصيلية لحالات الاستخدام

|  |  |
| --- | --- |
| Use Case: Find a Trip | |
| 2 | **ID** |
| The system finds some trips based on user search criteria and displays them to the user | **Brief description** |
| User or  anonymous | **Primary Actors** |
| - | **Secondary actor** |
| - | **Preconditions** |
| 1. The use case start when the user select “Trip browsing ” 2. The system ask the user for search criteria 3. The user enters the requested criteria 4. The system search for trips that match the user’s criteria 5. If the system finds some matching trips then:    1. The system displays the trip’s card contains (company name, get-way hour, delivery hour get-way address, destination address, and price)    2. The system provides a booking link for each trip’s card 6. Else the system tell the user there is no matching trips, and asking to change search criteria | **Main flow** |
| - | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: Trip booking | |
| 3 | **ID** |
| The user can book on trip online | **Brief description** |
| User  Local admin | **Primary Actors** |
| - | **Secondary actor** |
| 1. The user logged in 2. The user had enough balance in his account | **Preconditions** |
| 1. the user case starts when the user search for a trip and select “Trip booking” on trip’s card. 2. The user Enter number for seats 3. The systems calculate the price depends on the seats number, and show it to user automatically 4. After that the user press process button 5. Before the system send the request will display dropped window with the price and the Trip information and ask the user to final conferment 6. If the user click ok, the system check the user has enough balance for booking, then if the process complete the system show message “booking complete” 7. Else the system show the user the Error massage | **Main flow** |
| 1. The user is registered in the trip 2. Available seats on the trip decreased by the number of seats that the user book | **Post condition:** |
| 1. The costumer goes the company office 2. .the local admin itself register the user on the trip | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: Trip History | |
| 4 | **ID** |
| The system show to user list of his previous Trips | **Brief description** |
| User | **Primary Actors** |
| - | **Secondary actor** |
| The user has logged in | **Preconditions** |
| 1. The use case start when the user clicks on trip history 2. The system display all previous trip as a list 3. For each trip: the system shows two buttons:    1. Trip review button ( to estimate the trip)    2. Complaint button ( to submit a complaint ) | **Main flow** |
| - | **Post condition:** |
|  | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: Trip review | |
| 4.1 | **ID** |
| The user can review trips that he has booked | **Brief description** |
| User | **Primary Actors** |
| - | **Secondary actor** |
| * The user has logged in * The user makes at least one trip * The user selects this trip from history | **Preconditions** |
| The use case starts when the user clicks on Trip review  The system proved the user with:   1. Star scale to estimate the trip 2. Text box to write a short review | **Main flow** |
| The trip score changes | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: Trip Complaint | |
| 4.2 | **ID** |
| The system allows to user to submit a complaint for each trip he had booked | **Brief description** |
| User | **Primary Actors** |
| - | **Secondary actor** |
| * The user has logged in * The user makes at least one trip * The user selects this trip from history | **Preconditions** |
| The use case starts when the user clicks on submit complaint  The system proved the user with: three text boxes:  ( title , body, mail \* optional if communication is needed ) | **Main flow** |
| An complaint has send to company’s local admin | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: show data | |
| 5 | **ID** |
| The system display Announcement cards | **Brief description** |
| User  Local admin | **Primary Actors** |
| - | **Secondary actor** |
| * The user has logged in * The user is from the Announcement audience | **Preconditions** |
| The user goes to the main page | **Main flow** |
| None | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: Announcement Management | |
| 6 | **ID** |
| The local admin can review old declarations and send new one | **Brief description** |
| Local Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as local admin | **Preconditions** |
| 1. The use case starts when the local admin clicks on Announcement Management button 2. The system shows all historical permits and send a new Announcement button   . | **Main flow** |
| None | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: send declaration | |
| 6.1 | **ID** |
| The local admin can send an announcement for specific of general audience | **Brief description** |
| Local Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as local admin | **Preconditions** |
| 1. The use case starts when the local admin clicks on send declaration button 2. The system provides local admin with two text boxes   (header, body)   1. After the local admin write the message, and press on Done the system ask him to select the audience 2. After selecting the audience, the local admin can publish the announcement | **Main flow** |
| An declaration message will sent to audience | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: Trip schedule Management | |
| 7 | **ID** |
| The local admin can schedule trips, create and edited | **Brief description** |
| Local Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as local admin | **Preconditions** |
| 1. The use case starts when the local admin clicks on trip schedule management 2. The system shows list of planned trips 3. For each planned trip system shows edit button 4. The system display add new trip button | **Main flow** |
| None | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: edit trip | |
| 7.1 | **ID** |
| The local admin can edit some details for a trip | **Brief description** |
| Local Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as local admin  The user selects a trip from planned trip list | **Preconditions** |
| 1. The use case starts when the local admin clicks on edit trip on the list 2. The system display form for the details that could be edit 3. The system asks the local admin to confirm the new details 4. The system send automatically declaration for effected user | **Main flow** |
| The system sends an automatically declaration for effected customer | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: create a trip | |
| 7.2 | **ID** |
| The local admin can add new trip | **Brief description** |
| Local Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as local admin | **Preconditions** |
| 1. The use case start when the local admin click on add trip button 2. The system ask the local admin about trip criteria 3. The local admin enter the planned trip criteria 4. The local admin click on add button 5. The system shows the local admin prompt window of the trip information 6. The local admin confirm that the information is well right 7. The system add the new trip | **Main flow** |
| New trip will be added to the database and be available for booking | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: show complaint reports | |
| 8 | **ID** |
| The local admin can review complaints from users | **Brief description** |
| Local Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as local admin | **Preconditions** |
| 1. The use case start when the local admin click complaints report button 2. The system displays list of complaint 3. For each unread it item, the system shows it in different color 4. the system provide print form for items | **Main flow** |
| - | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: show Economic reports | |
| 9 | **ID** |
| The local admin can review some Economic reports | **Brief description** |
| Local Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as local admin | **Preconditions** |
| 1. The use case start when the local admin click Economic report button 2. The system shoes details about trips and its profitability | **Main flow** |
| - | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: account management | |
| 10 | **ID** |
| The Global admin manage all types of accounts (create , blocks ) | **Brief description** |
| Global Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as global admin | **Preconditions** |
| 1. The use case starts when the global admin click account management 2. The system show three buttons ( global admin list , local admin list, user list ) | **Main flow** |
|  | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: create a new account | |
| 10.1 | **ID** |
| The Global admin can create a new account | **Brief description** |
| Global Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as global admin | **Preconditions** |
| 1. The use case starts when the global admin click create a new account 2. The system asks global admin about account criteria 3. The global admin add account information and click add 4. The system shows confirmation window 5. The global admin confirm | **Main flow** |
|  | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: company management | |
| 11 | **ID** |
| The Global admin can manage all companies on the system | **Brief description** |
| Global Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as global admin | **Preconditions** |
| The use case starts when the user click on company management button on the dashboard  The system shows list of all working companies  The global admin can edit/ banned any company  the admin can add new company | **Main flow** |
|  | **Post condition:** |
| None | **Alternative flow:** |

|  |  |
| --- | --- |
| Use Case: balances accounting | |
| 12 | **ID** |
| The Global can control balances accounting | **Brief description** |
| Global Admin | **Primary Actors** |
| - | **Secondary actor** |
| The user logged in as global admin | **Preconditions** |
| The use case starts when the user clicks on balances section  The system provides the global admin with adding balance form  The global adding enters the account details  The system updates the new balance for the account | **Main flow** |
| The system sends a declaration to effected user | **Post condition:** |
| None | **Alternative flow:** |

**شجرة الواجهات: ) الصور مرفقة في القسم التالي)**

**الآدمن:**

**المسافر :**

## **مصفوفة تتبع النظام RTM:**

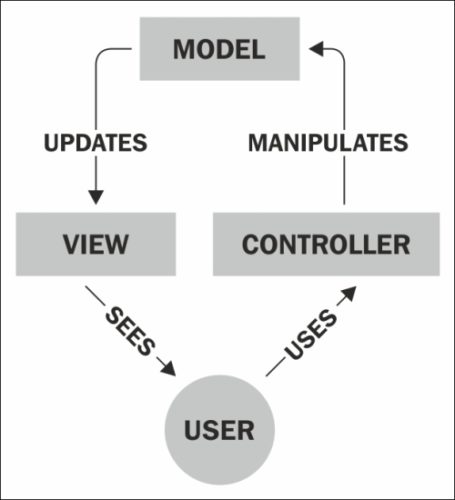
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Requirements Traceability Matrix (RTM) | | | | | |  |  | |  | |  |  | |  |  | |  | |  | |  |
| **ID** | | **Title** | **Associated Dependency** | **SRS Section** | **High Level Design** | **Detailed Design** | | | **Coding** | | **Component Test** | | | **Integration Test** | | | **System Test** | | **Acceptance Test** | |
| R1 | | يجب ان يكون المدير المحلي للشركة قادر على إدارة اسطول الشركة من السيارات او الطائرات او الباصات، وذلك من خلال اتاحة ( إضافة – تعديل – حدف- استعراض الاسطول\* |  | [**Study1**](file:///D:\Dataware\SPU_Total.docx) | [**Sec**](file:///D:\Dataware\Files\sec.png) | **USE case 11** | | |  | |  | | |  | | | [**Test1**](file:///D:\Dataware\Files\test%201%20.side) | |  | |
| R1-2 | | يجب على النظام الا يسمح للادمن المحلي من إضافة رحلة في تاريخ سابق | **R2** | [**Study1**](file:///D:\Dataware\SPU_Total.docx) | [**Sec**](file:///D:\Dataware\Files\sec.png) | **USE vase 7.2** | | |  | |  | | |  | | | [**Test 2**](file:///D:\Dataware\Files\Test2.side) | |  | |
| R2 | | يجب ان يكون المدير المحلي قادر على إدارة جدول الرحلات الخاصة بالشركة (إضافة، تعديل، استعلام) | **R1** | [**Study1**](file:///D:\Dataware\SPU_Total.docx) |  | **USE case 6** | | |  | |  | | |  | | | [**Test 2**](file:///D:\Dataware\Files\Test2.side) | |  | |
| R3 | | يجب ان يكون المدير المحلي قادر على استعراض طلبات الحجز والتأكد من سلامة عملية الحجز (التحقق من الدفع وصحة معلومات العملاء) | **R2 , R7** |  | [**Sec**](file:///D:\Dataware\Files\sec.png) |  | | |  | |  | | |  | | | [**Test 3**](file:///D:\Dataware\Files\Test3.side) | |  | |
| R4 | | يجب ان يكون المدير المحلي قادر على استعلام عن تقارير خاصة (على مستوى الشركة) كعدد الحجوزات في رحلة معينة، ورأي العملاء فيها | **R2,R3,R7** | [Study3](http://tfsmain/sites/DHS/Product%20Development/DHS01-PD-SoftwareRequirementsSpecification.docx) | [**Sec**](file:///D:\Dataware\Files\sec.png) |  | | |  | |  | | |  | | | [**Test 3**](file:///D:\Dataware\Files\Test3.side) | |  | |
| R5 | | يجب ان يكون المدير المحلي قادر على تلقي شكاوي العملاء، وتحويلها إلى القسم المختص في الشركة. |  | [Study3](http://tfsmain/sites/DHS/Product%20Development/DHS01-PD-SoftwareRequirementsSpecification.docx) | [**Sec**](file:///D:\Dataware\Files\sec.png) |  | | |  | |  | | |  | | |  | |  | |
| R6 | | يجب على المستخدم ان يكون قادر على الاستعلام عن رحلات من خلال عدد من المدخلات (مكان الانطلاق والوجهة، الزمن .. الخ) |  |  | [**Sec**](file:///D:\Dataware\Files\sec.png) |  | | |  | |  | | |  | | | [**Test4**](file:///D:\Dataware\Files\Test4.side) | |  | |
| R7 | | يجب على المستخدم ان يكون قادر على الحجز على الرحلة وإتمام عملية الدفع. | **R6** |  |  |  | | |  | |  | | |  | | | [**Test4**](file:///D:\Dataware\Files\Test4.side) | |  | |
| R8 | | يجب على النظام ارسال إشعارات حول تغيرات التي من المحتمل ان تطرأ على الرحلة. | **R6,R7** |  | [**Sec**](file:///D:\Dataware\Files\sec.png) |  | | |  | |  | | |  | | |  | |  | |
| R9 | | يجب على النظام ان يسمح للعميل بارسال شكوى بعد القيام برحلة ما | **R6, R7,R8** |  | [**Sec**](file:///D:\Dataware\Files\sec.png) |  | | |  | |  | | |  | | |  | |  | |

## **معمارية النظام:**

تم الاعتماد على معمارية MVC في بناء الهيكلية الخاصة بالنظام، هو اختصار لModel-View- Controller وهو نمط يستخدم لفصل مكونات المشروع المنطقية عن بعضها وهو من افضل النظم لإنشاء موقع قابل للتوسع.

يساعد نمط MVC على فصل الأكواد البرمجية وتقسيمها إلى أجزاء مستقلة، مما يسهل عملية الصيانة والتطوير والتحكم بالتطبيقات بشكل فعال. بالإضافة إلى ذلك، يمكن لفرق التطوير المختلفة العمل على أجزاء مختلفة من التطبيق بشكل مستقل دون التأثير على باقي الأجزاء.

ويتيح استخدام نمط MVC أيضًا توسيع وتطوير التطبيقات بسهولة، حيث يمكن إعادة استخدام الأكواد البرمجية المكتوبة في أجزاء محددة من التطبيق في أجزاء أخرى، مما يساعد على تسريع عملية التطوير والإنتاجية.

 MVC نمط من انماط التصميم Design Pattern تعتبر أنماط التصميم (Design Patterns) من أهم مفاهيم هندسة البرمجيات، وهي نماذج وصفية لحل مشاكل تصميم البرامج. تم اقتباس معظم هذه الأنماط من الخبرة السابقة للمبرمجين في حل مشاكل مشابهة.

**مكونات MVC**

**View** هو عبارة عن الواجهة التي يتعامل معها المستخدم في التطبيق مثل القوائم و حقول البحث و السلايدات وغيرها.

**Model** يتعامل مع البيانات التي يتعامل معه المستخدم ويدخلها الى قاعدة البيانات او يخرج بيانات معينة منها. إذا فدوره مقتصر على التعامل مع قواعد البيانات.

**Controller** هو الوسيط بين view و controller فهو يعالج العمليات المنطقية والطلبات القادمة (requests) وكذلك التعامل مع البيانات القادمة من قواعد البيانات لإخراجها على view في صورة ملائمة.

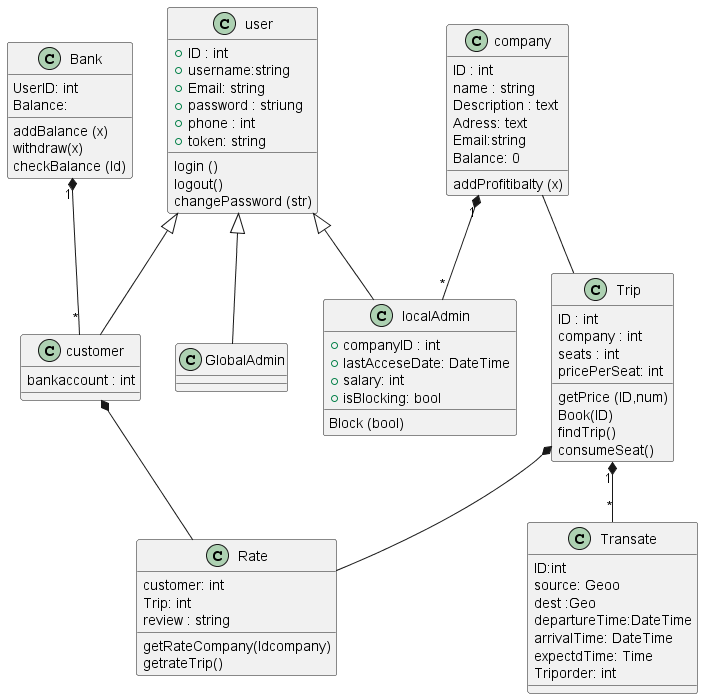
# 

# **الفصل الرابع**

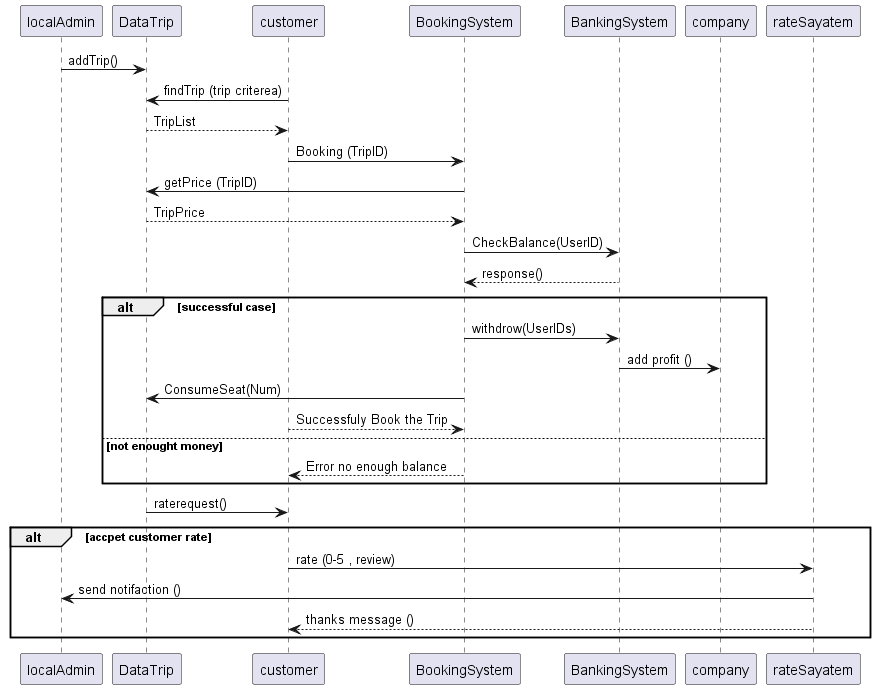
# **الدراسة التصميمية للنظام المقترح**

نستعرض في هذا القسم المخططات التصميمية للنظام، بالإضافة إلى عرض واجهات المستخدم الخاصة بالنظام:

## **مخطط الصفوف:**

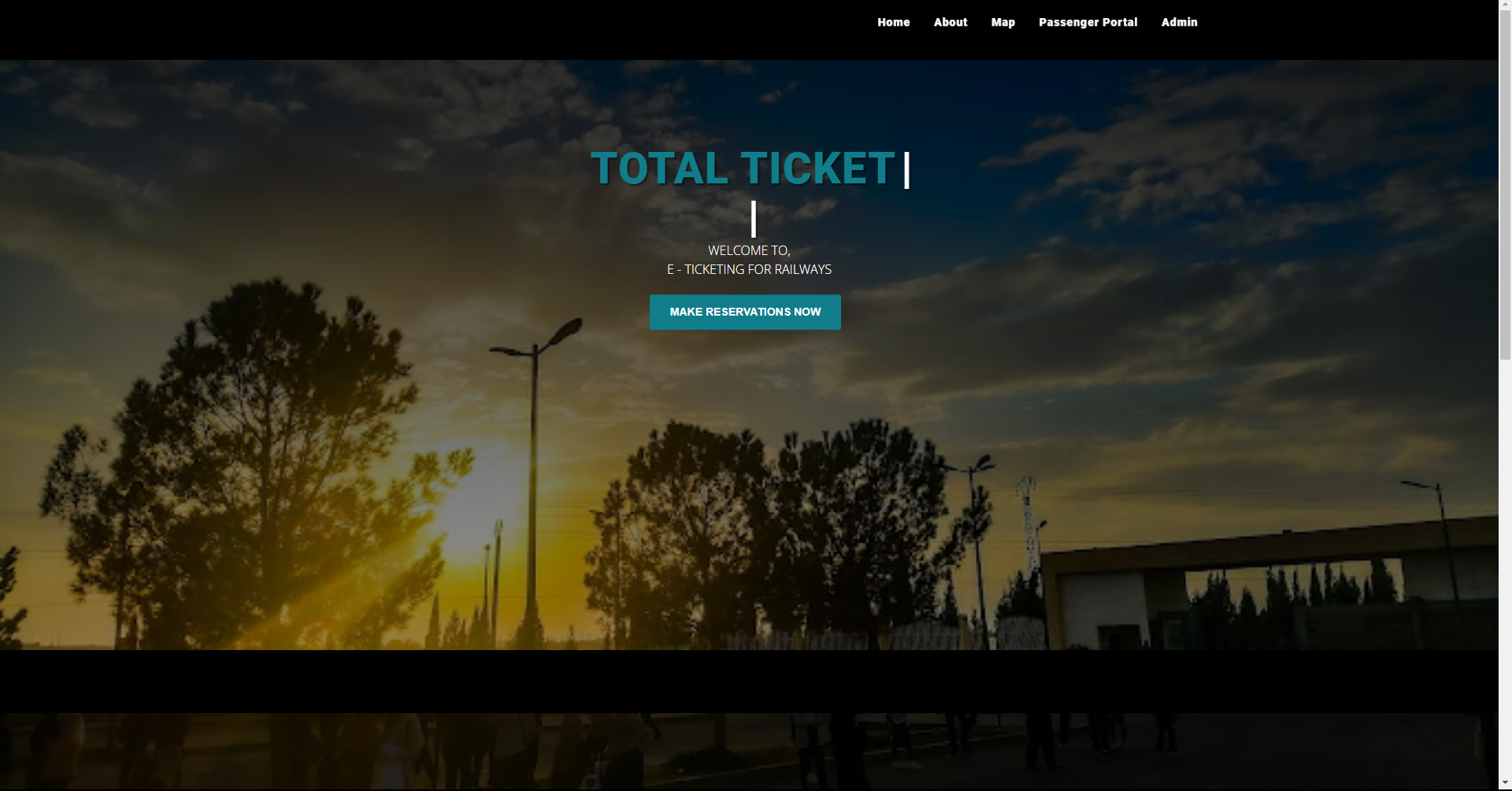


## **مخطط sequence للعملية الأساسية في النظام**

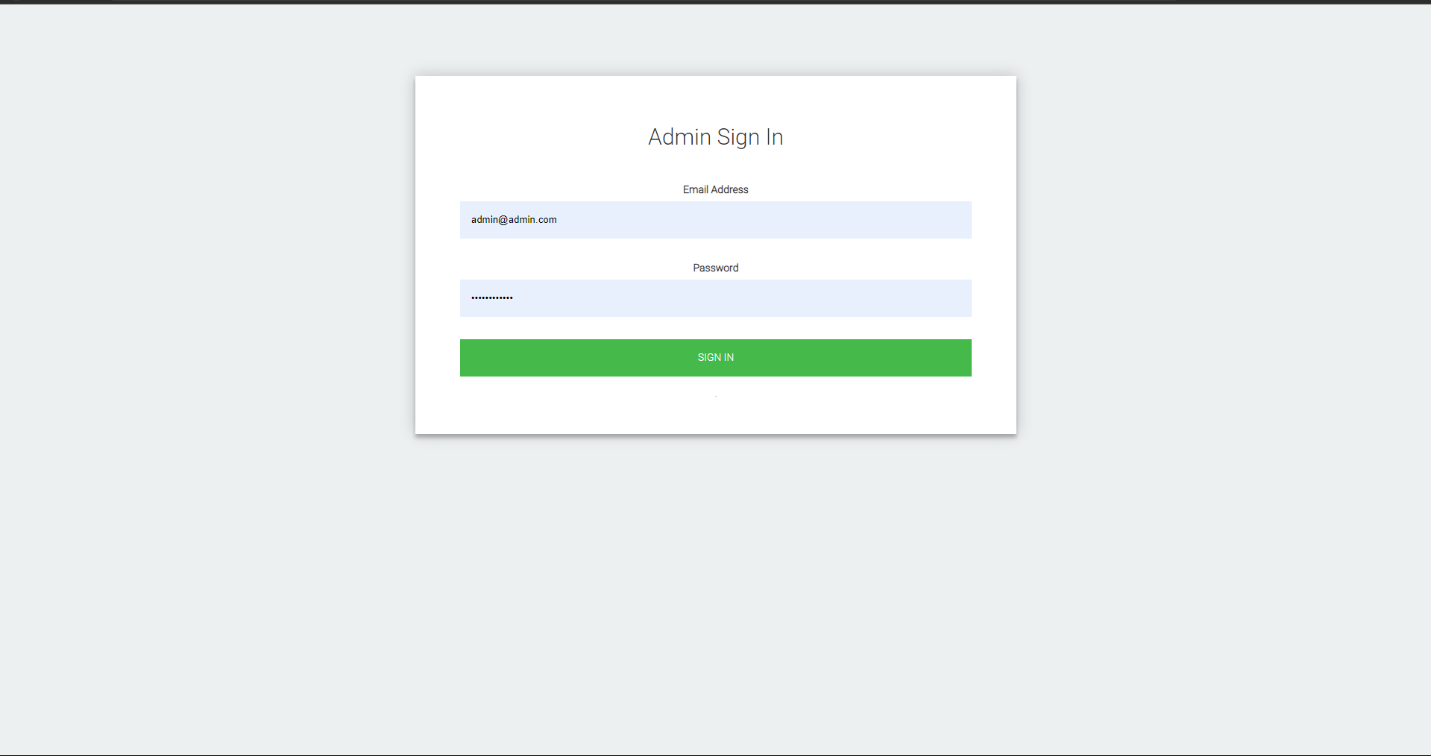


## **واجهات النظام:**

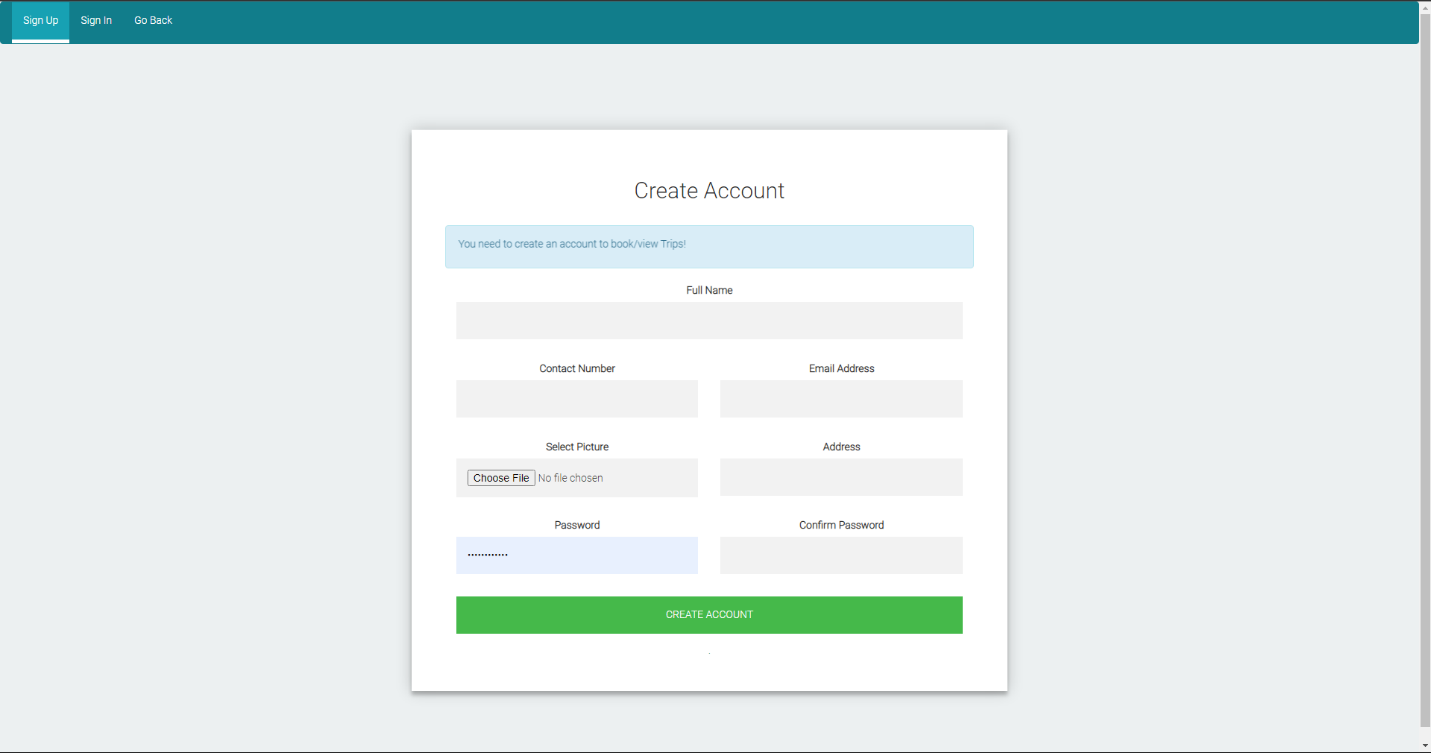
1. **واجهة المستخدم العام:**



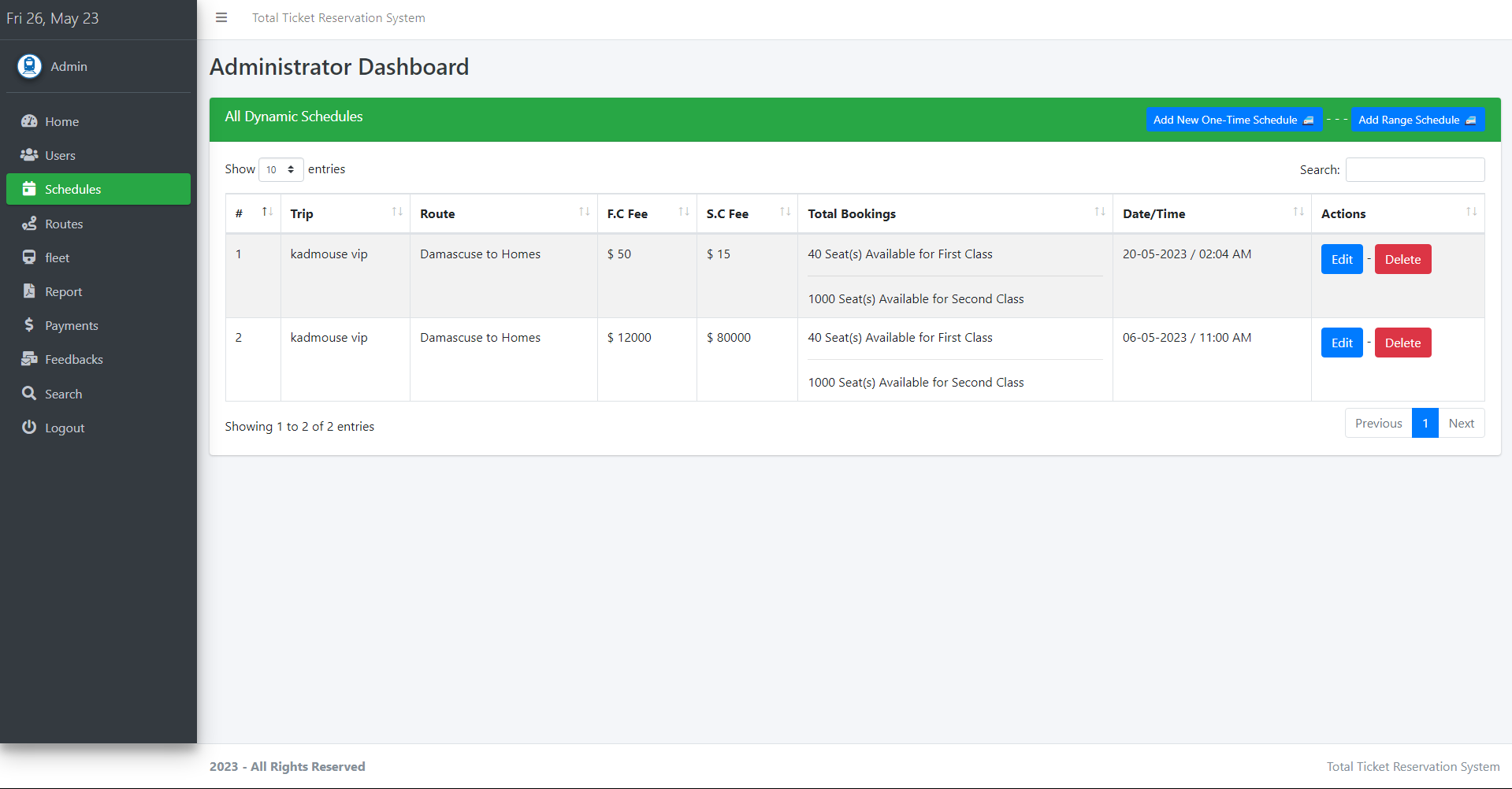
1. **واجهة تسجيل الدخول:**



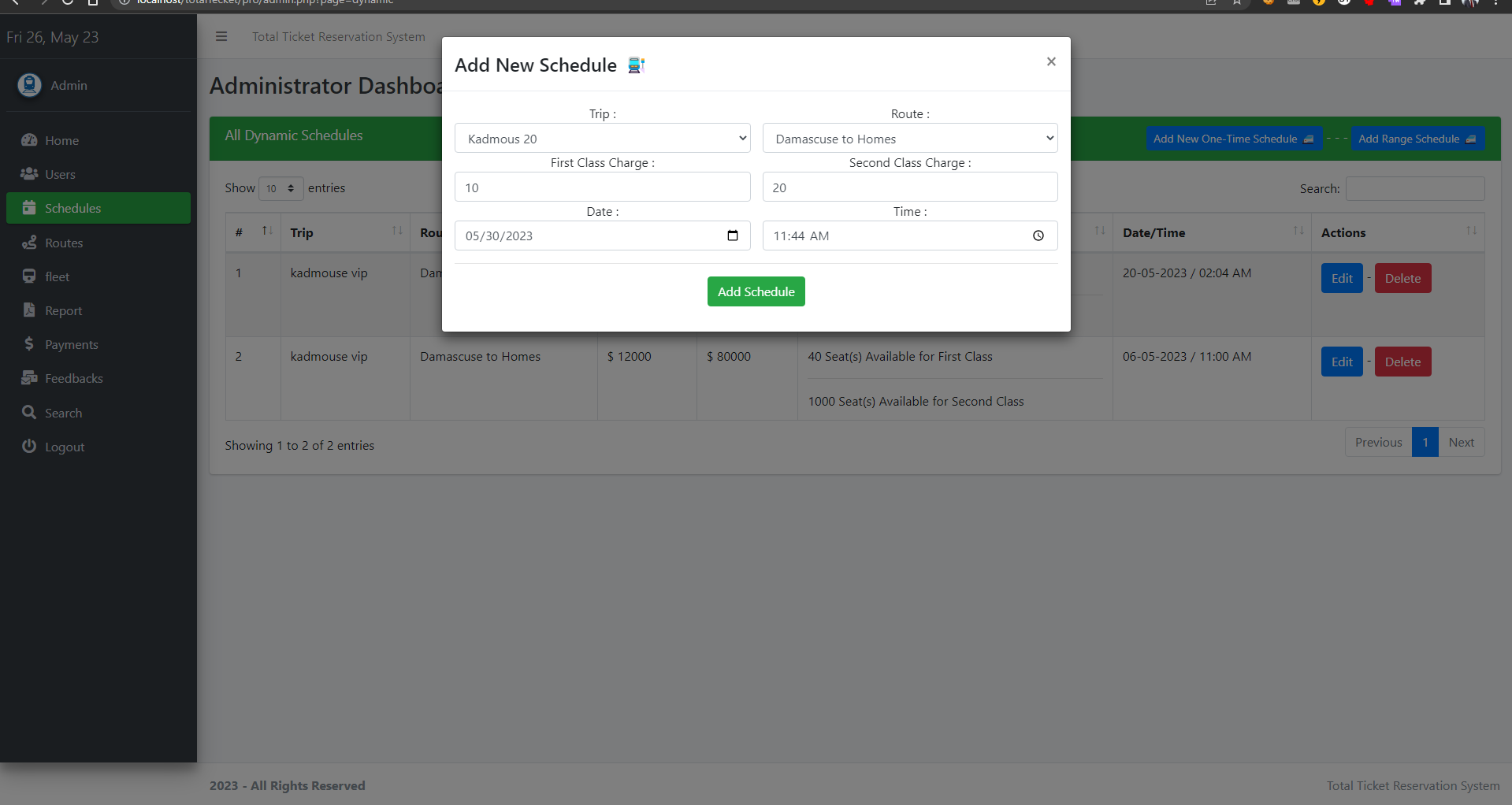
1. **واجهة تسجيل مستخدم جديد:**



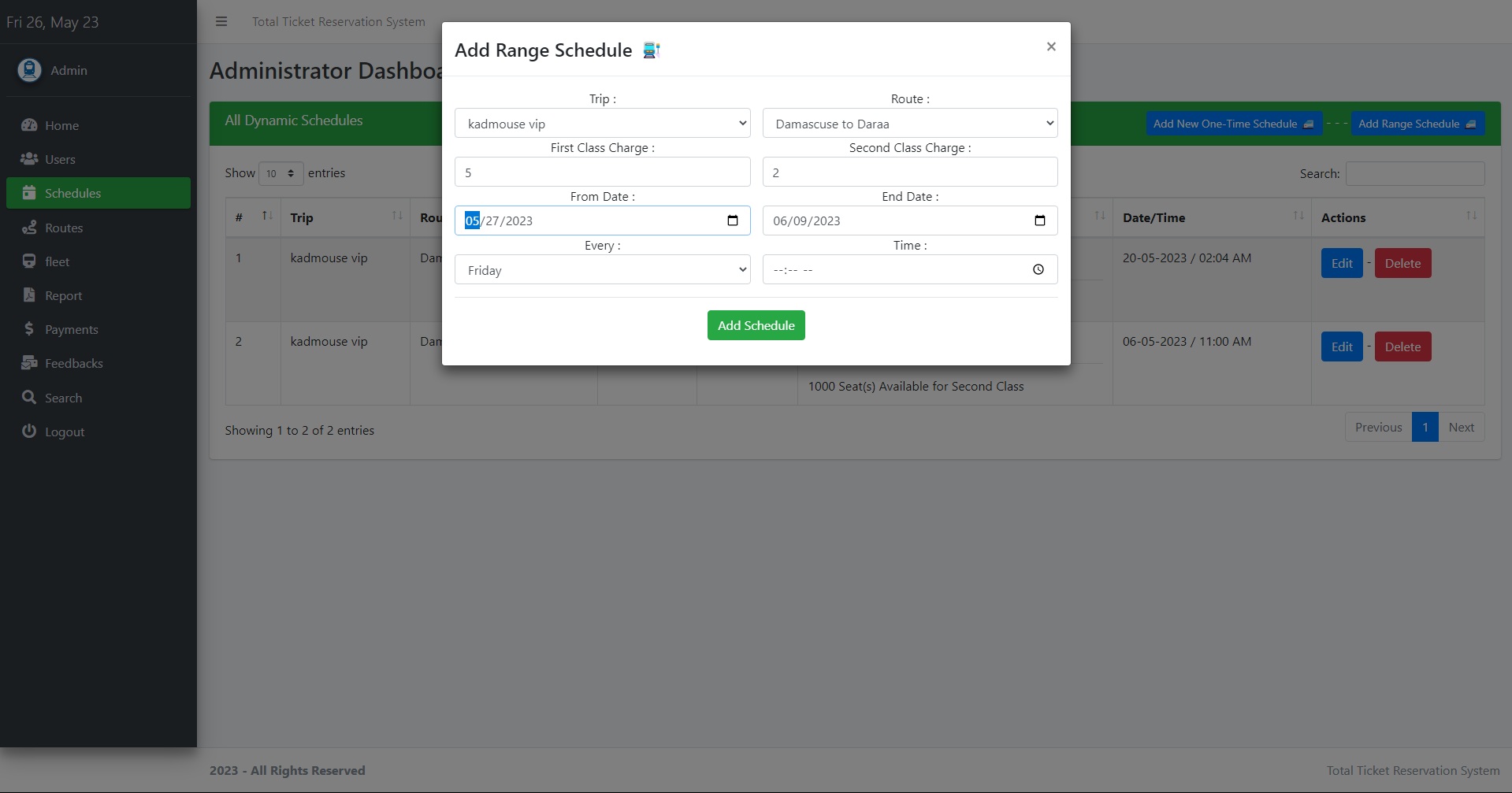
1. **إدارة الرحلات: والتي يمكن من خلالها إضافة – حذف – تعديل رحلة**



1. إضافة رحلة في وقت استثنائي ( لمرة واحدة ) :

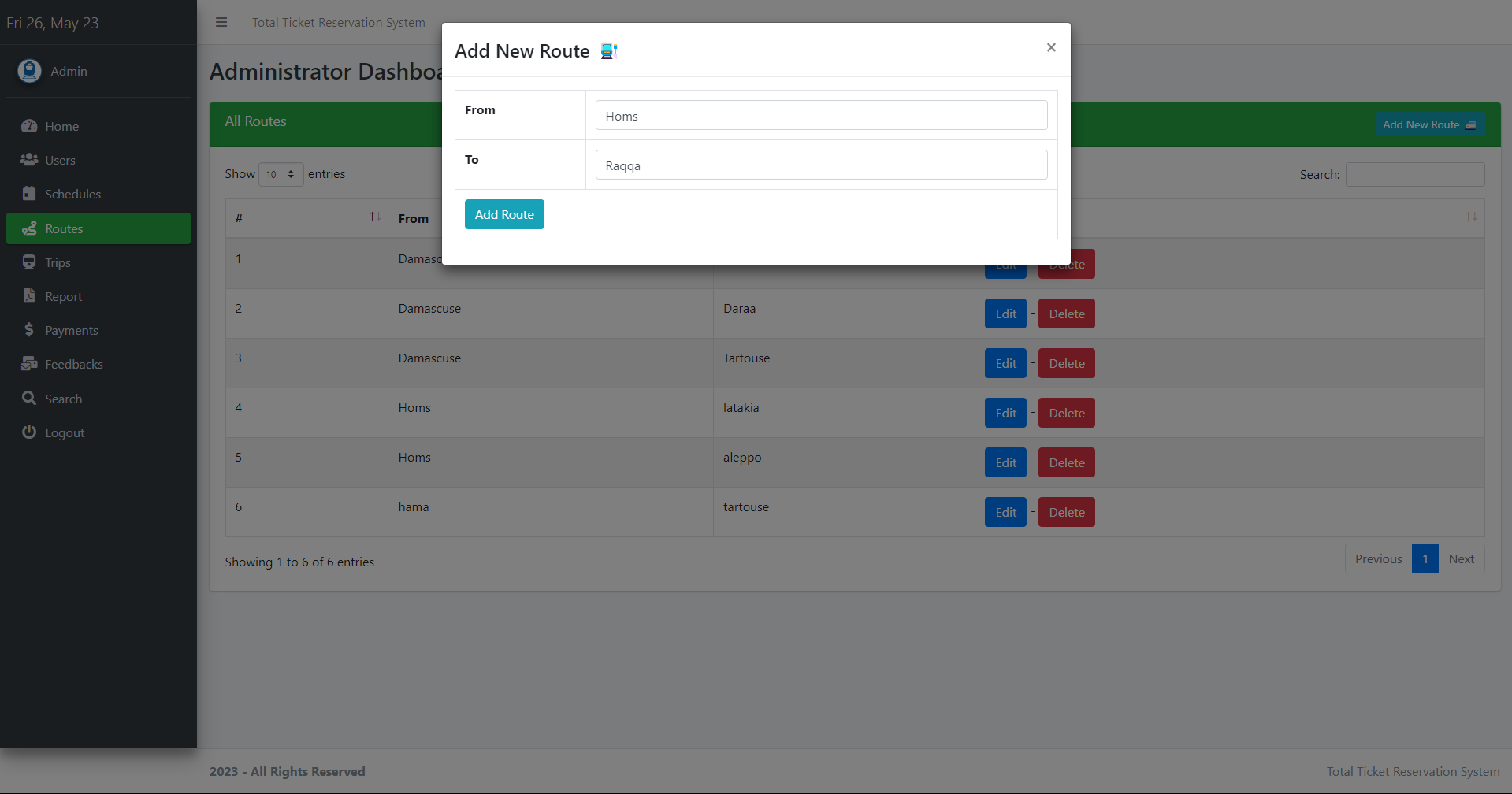


1. إضافة رحلة مجدولة ( كل أسبوع مثلاً ) :

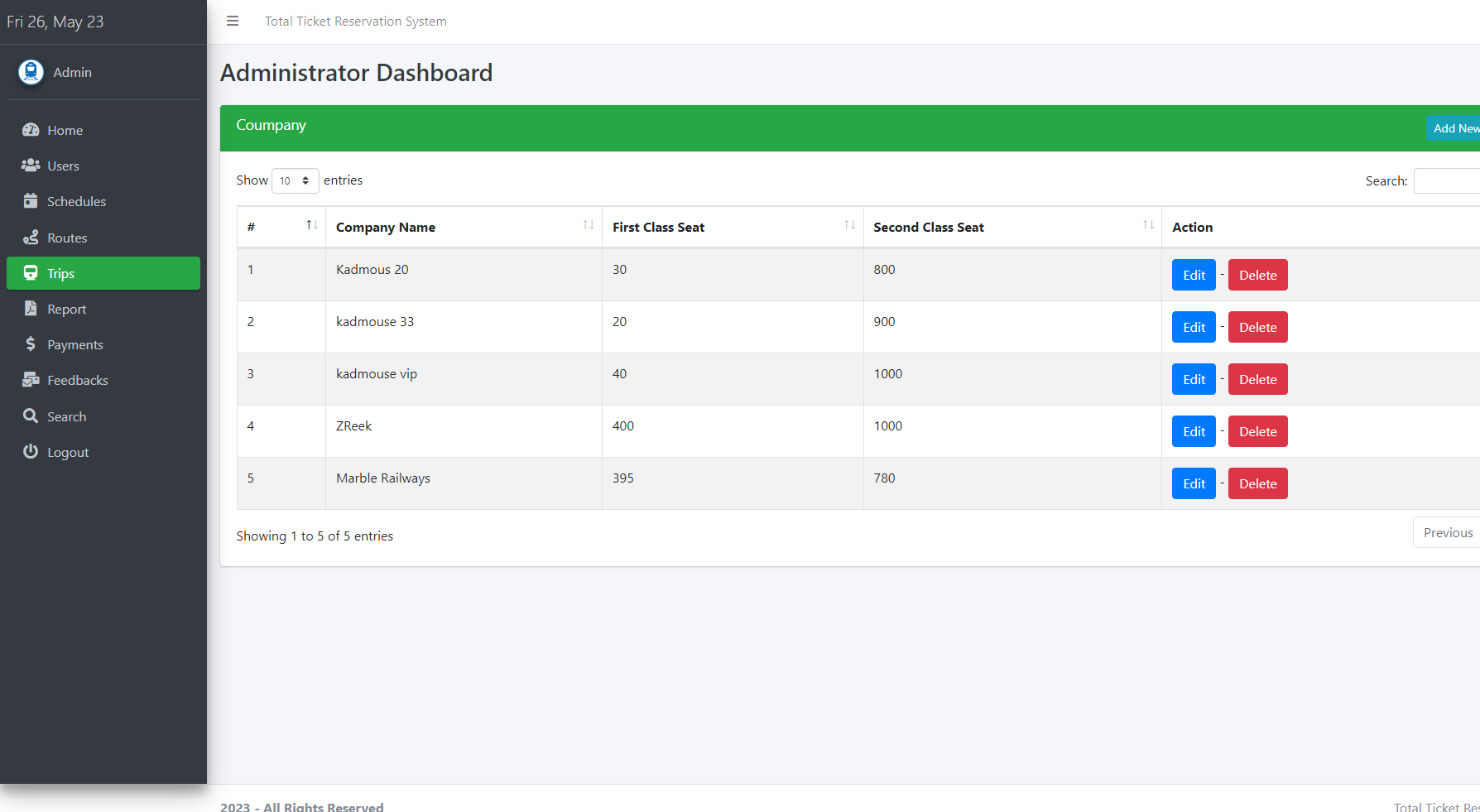


1. إدارة خطوط السيرة ( من نقطة إلى نقطة) ( لاستخدامها عندما يكون للنقلة أكثر من موقف)

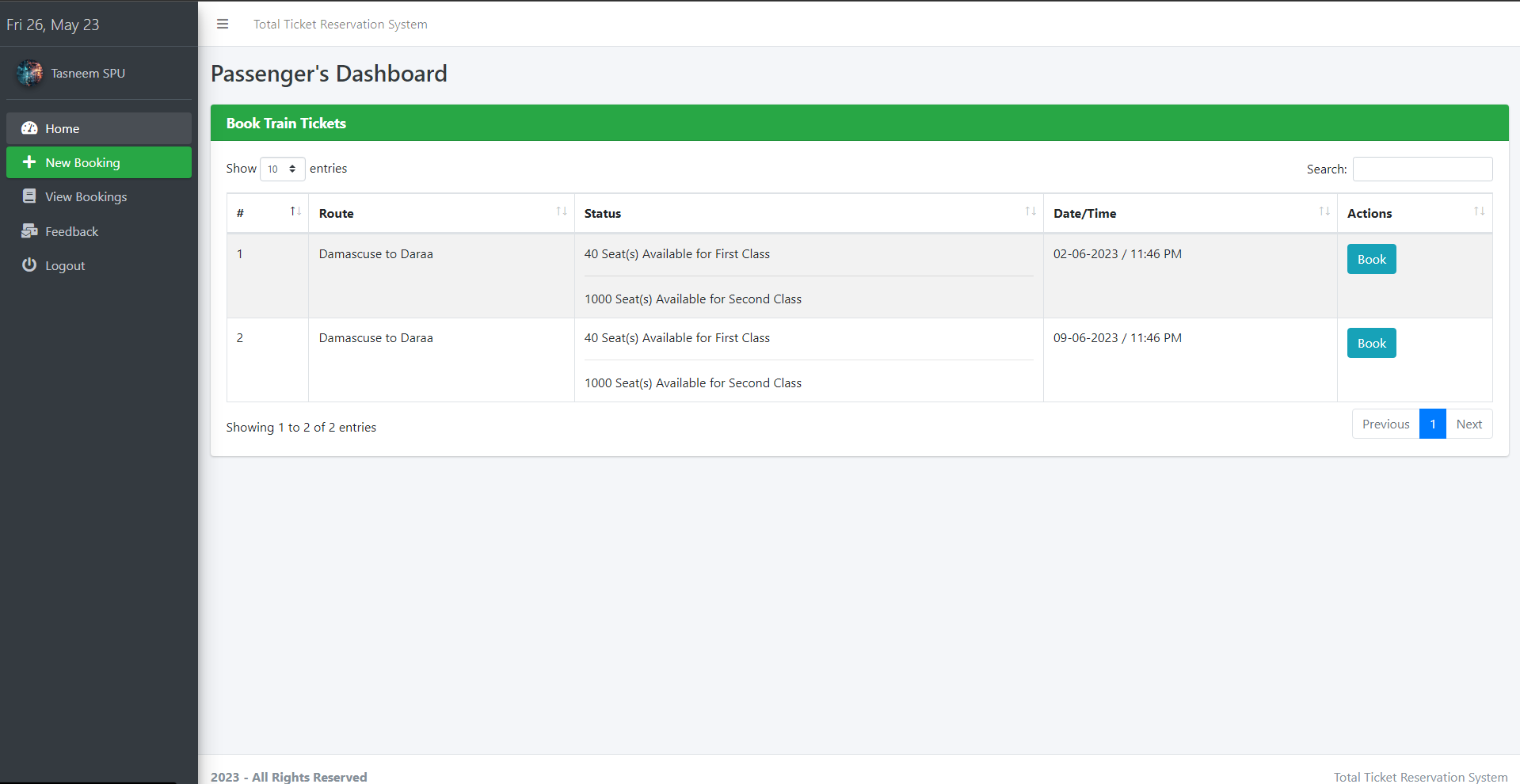
 .



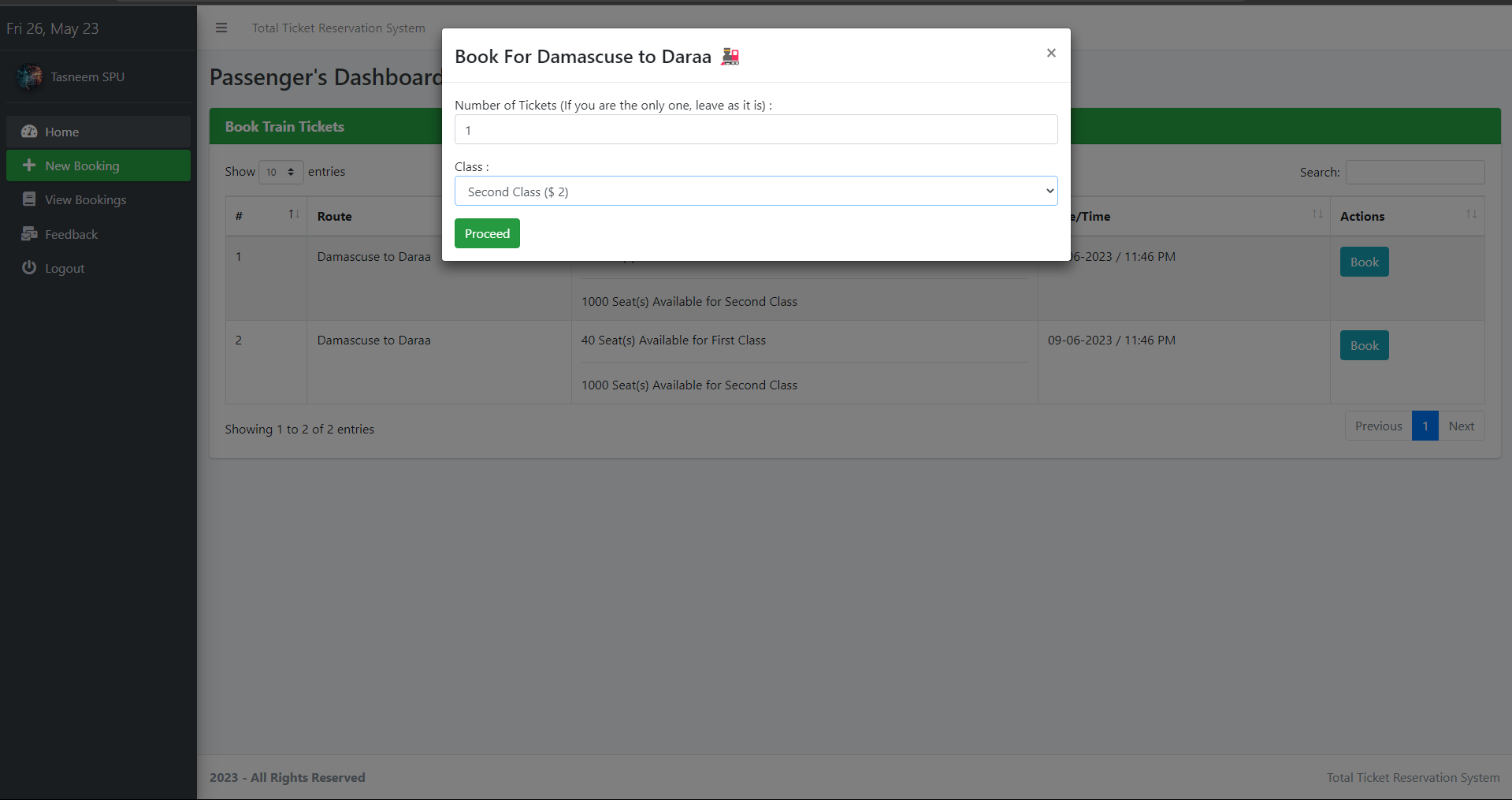
1. إدارة الاسطول: fleet ( إضافة باصات، نوع العربة هو من يحدد نوع الرحلة)



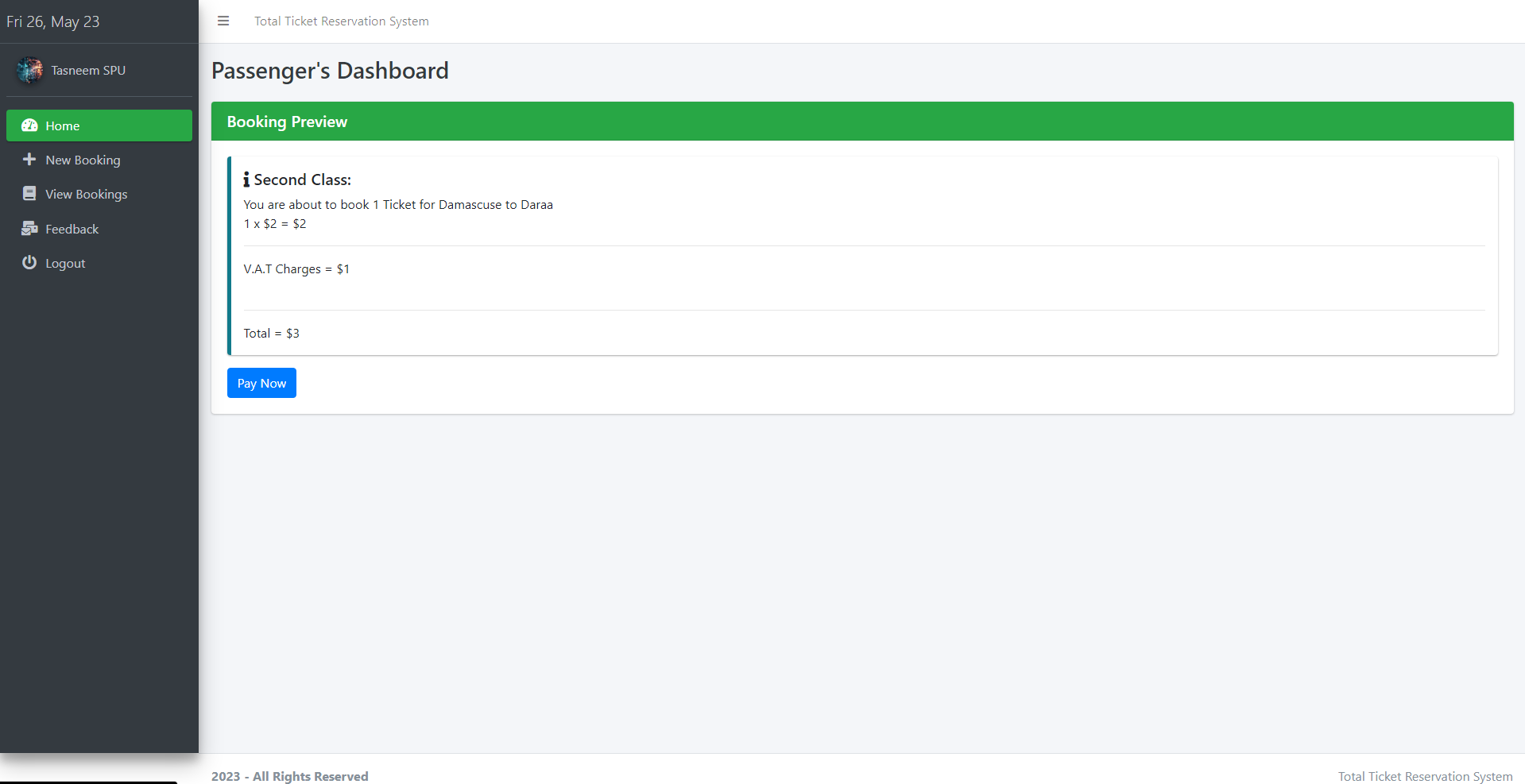
1. واجهة المستخدم في البحث عن رحلة:



1. عندما يضغط على رز book : ، ثم يحدد عدد المقاعد ونوع الرحلة ( في الدرجة الأولى او الاقتصادية)



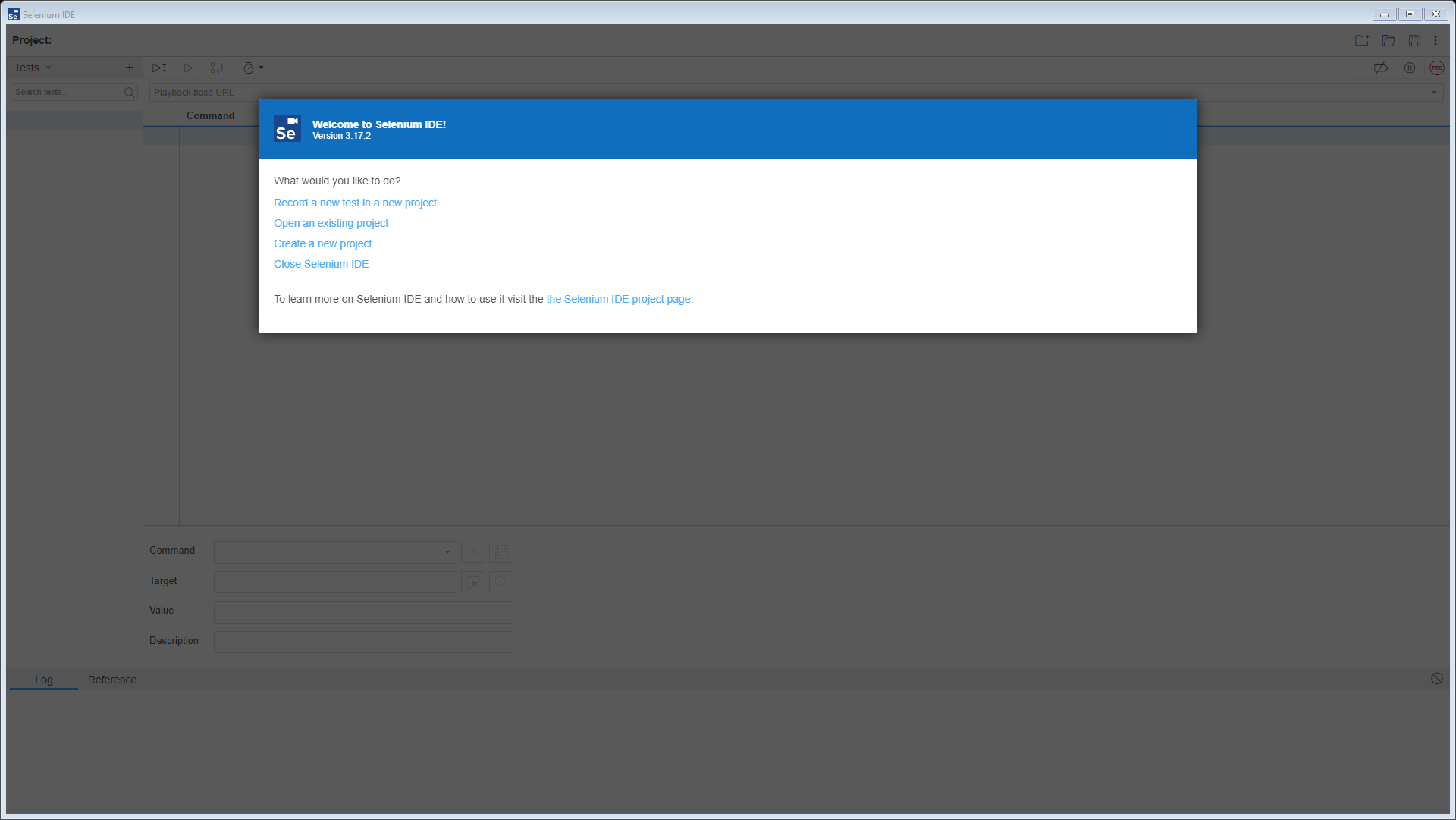
1. اظهار الفاتورة:



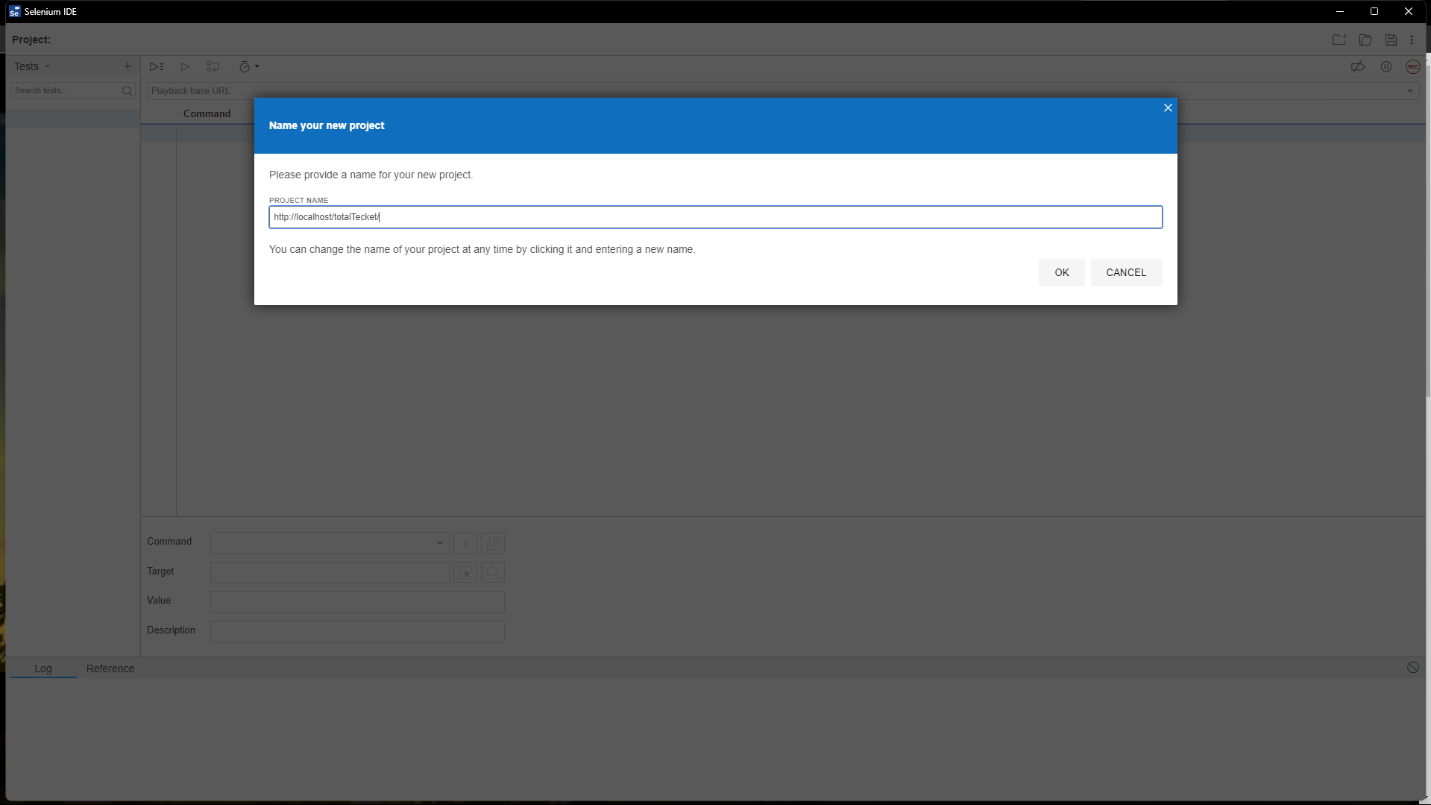
# **الفصل السادس**

# **اختبار النظام**

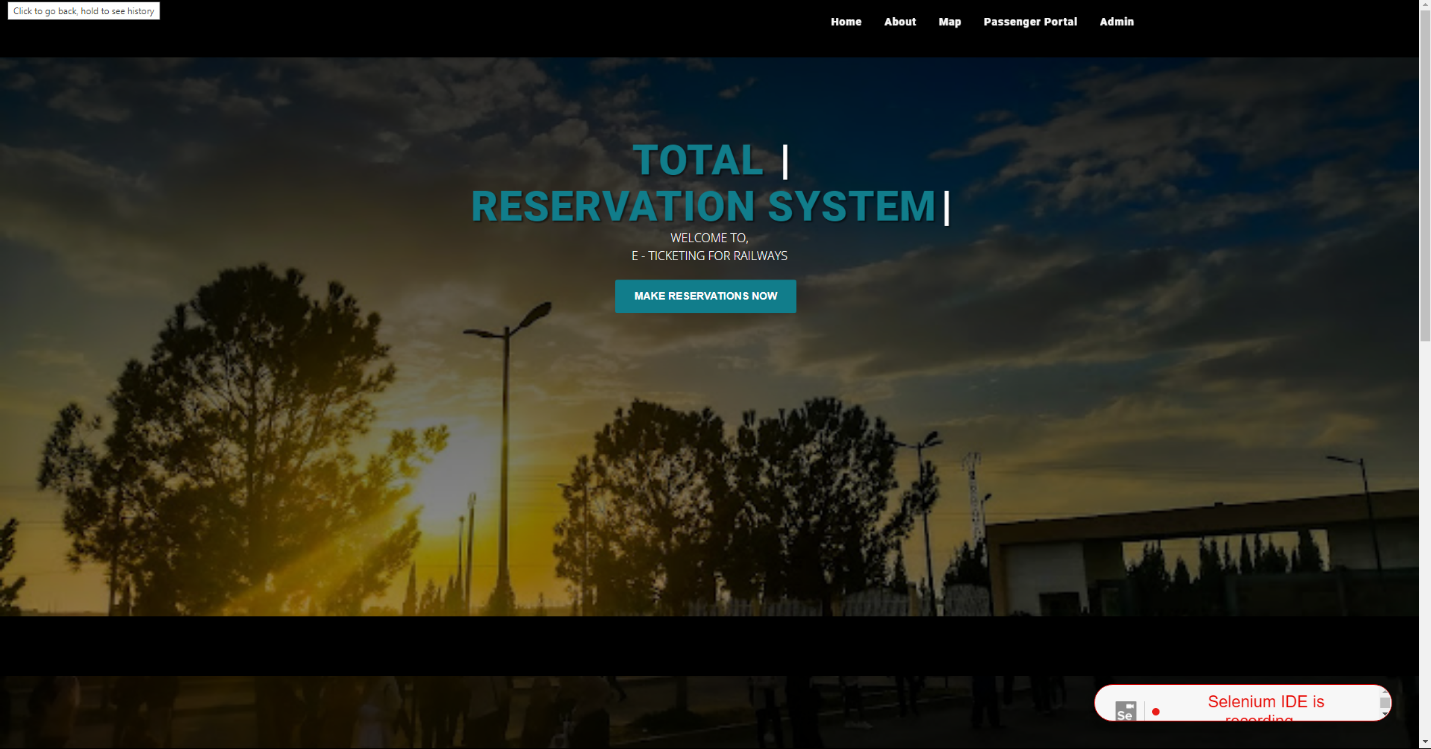
لقد تم اجراء اختبار للنظام من خلال إضافة Selenium IDE والتي تسمح لنا بتسجيل الخطوات المتبعة في اختبار أي نظام يعمل على الويب، والتي تفيد غالبا إقرار عمل أو فشل النظام.



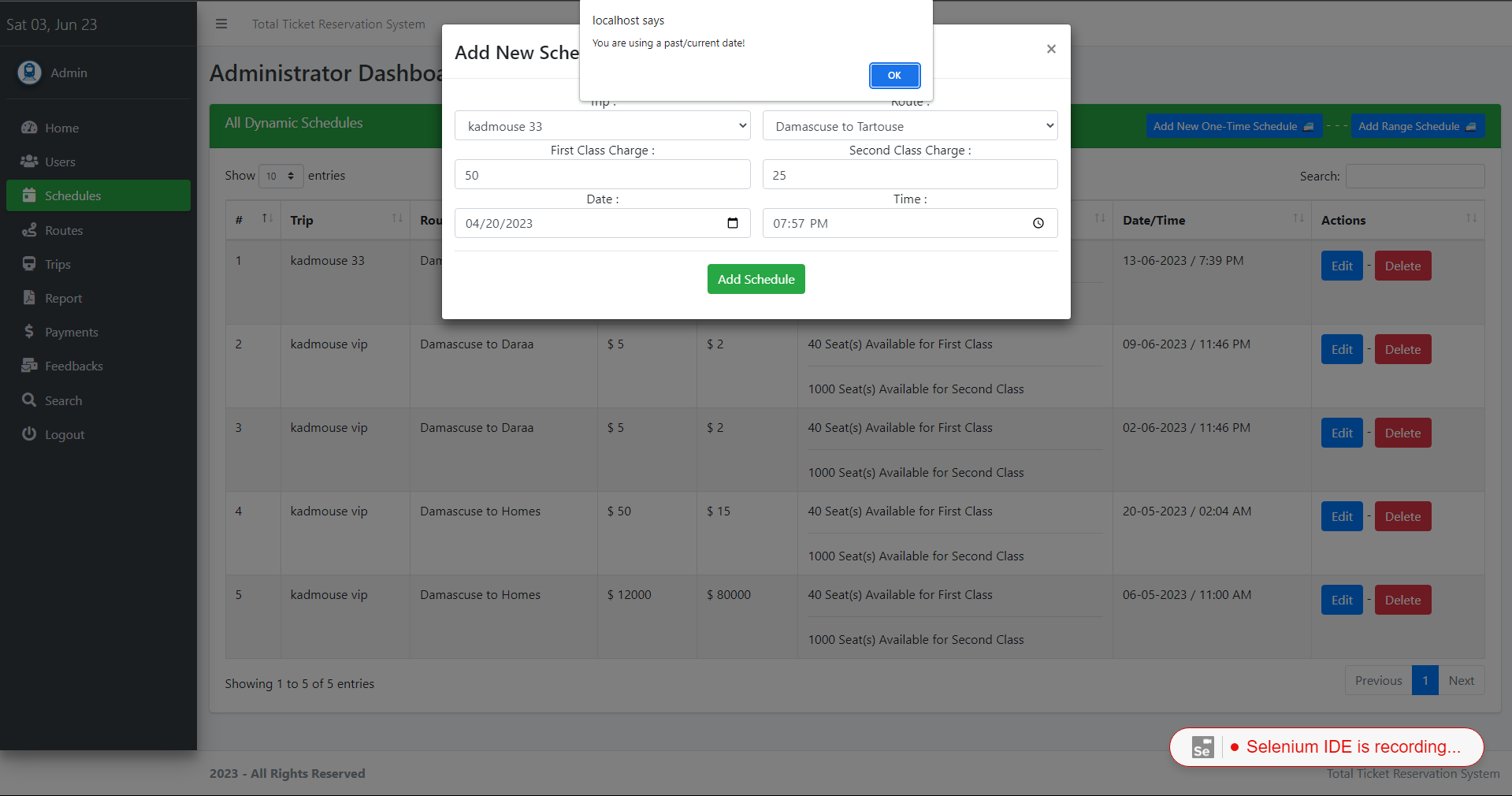
قمنا بعد ذلك بإضافة عنوان المشروع:



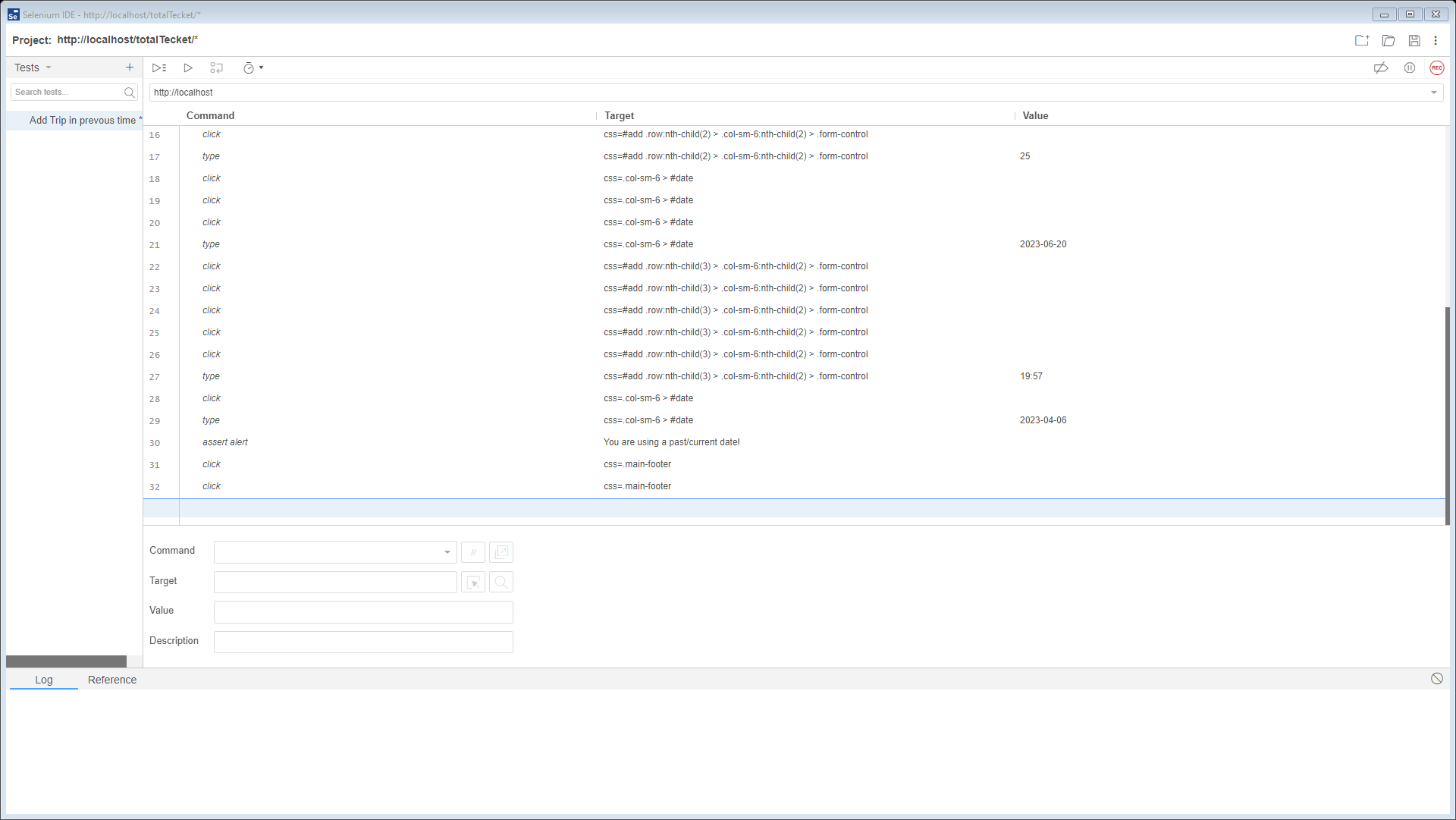
ليبدأ بتسجيل ومراقبة أداء وسلوك النظام:



فعلى سبيل المثال عند اختبار إضافة رحلة في موعد سابق تمكن النظام من التعرف على المشكلة وإظهار رسالة خطأ تبين عدم إمكانية إضافة رحلة في وقت مضى:



نتيجة الخطوات على برمجية الاختبار:



# **الفصل الخامس**

# **التطبيق العملي**

## **إعداد مخدم النظام API Server :**

تم تصميم المخدم اعتماداً على إطار عملPython Fast API | ، الذي يقدم بيئة مثالية للتصميم مخدم ويب يتعامل مع تطبيقات مختلفة، بالإضافة إلى إتاحة أدوات خاصة بالتعامل مع قواعد البيانات الغير علائقية مثل mongoDB Motor الذي يسمح لنا بتنفيذ تعليمات الاستعلام بلغة بايثون.

## **المتطلبات والحزم البرمجية الاساسية المطلوبة لبرمجة المخدم:**

* **Fast API Python Framework v: 0.65.1**
* **UVICORN** is an ASGI [[1]](#footnote-1)web server implementation for Python.**== 0.14.0**
* **MOTOR** is a full-featured MongoDB driver for Python ==2.4.0

كما سيتم التطرق إلى الحزم البرمجية الداعمة في السياق التي وردت فيه.

## **خطوات برمجة مخدم الويب الخاص بالتطبيق:**

**أولاً: تنصيب الأدوات:**

في البداية يجب أن نقوم مسبقاً بتنصيب الأدوات التالية:

* Vs. code IDE : <https://code.visualstudio.com/download>
* MongoDB (Compass , and MongoDB Shell, MongoDB Community Server): <https://www.mongodb.com/try/download/community>
* Python Programming Language : <https://www.python.org/downloads/>
* Anaconda Package Manager:

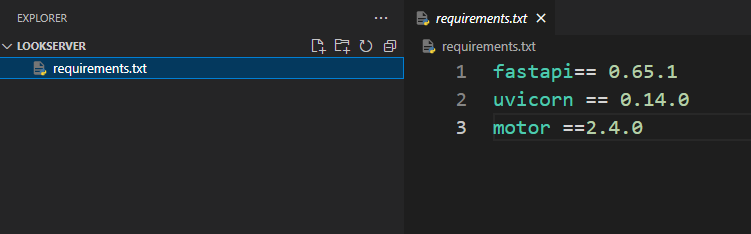
<https://www.anaconda.com/products/distribution>

## **ثانياً: إعداد بيئة العمل:**

**بعد ان** نقوم بإنشاء مجلد خاص بملفات السيرفر باسم Look-Server وندخل إلى المجلد عن طريق vs code.

في البداية سنقوم بإضافة ملف نصي خاص بالمتطلبات التي نريدها باسم requirements.txt

لتسيل عملية تنصيب المتطلبات خاصة بعد رفع المخدم على سيرفر الاستضافة.



كما سنقوم بإنشاء ثلاثة ملفات اساسية:

* Main.py :
* Models.py:
* DBconection.py

وبعد ذلك نقوم بالمرحلة الأولى من العمل وهو إعداد بيئة افتراضية خاصة بالمخدم من خلال anaconda

وذلك من خلال الأمر التالي:

تسمح لنا بيئة العمل بتجنب الأخطاء الناتجة عن تضارب المخدمات المختلفة على الجهاز من جهة، وسهولة نقل النظام.

بعد ذلك نقوم بتفعيل بيئة العمل من خلال الأمر التالي:

**وتنصيب المتطلبات الموجودة في ملف requirements.txt ننصب حزمة إضافية خاصة ب بايثون pip**

**تنصيب الحزم البرمجية المساعدة:**

* **للتعامل مع المعطيات الجغرافية:**

كما سنقوم بتنصيب مكتبة خاصة بدعم التعامل مع المعطيات الجغرافية ضمن بايثون

* **لأغراض الحماية والتوثيق security and authentication :**

سنقوم بتنصيب عدة برمجيات خاصة بالتوثيق والتي تقدم أغراضاً وطرائق object and methods للتحقق، تشفير كلمات السر، إنشاء مفاتيح تشفير:

حيث سيتم استخدام خوارزمية Bcrypt لتشفير كلمات السر الخاصة بالمستخدمين

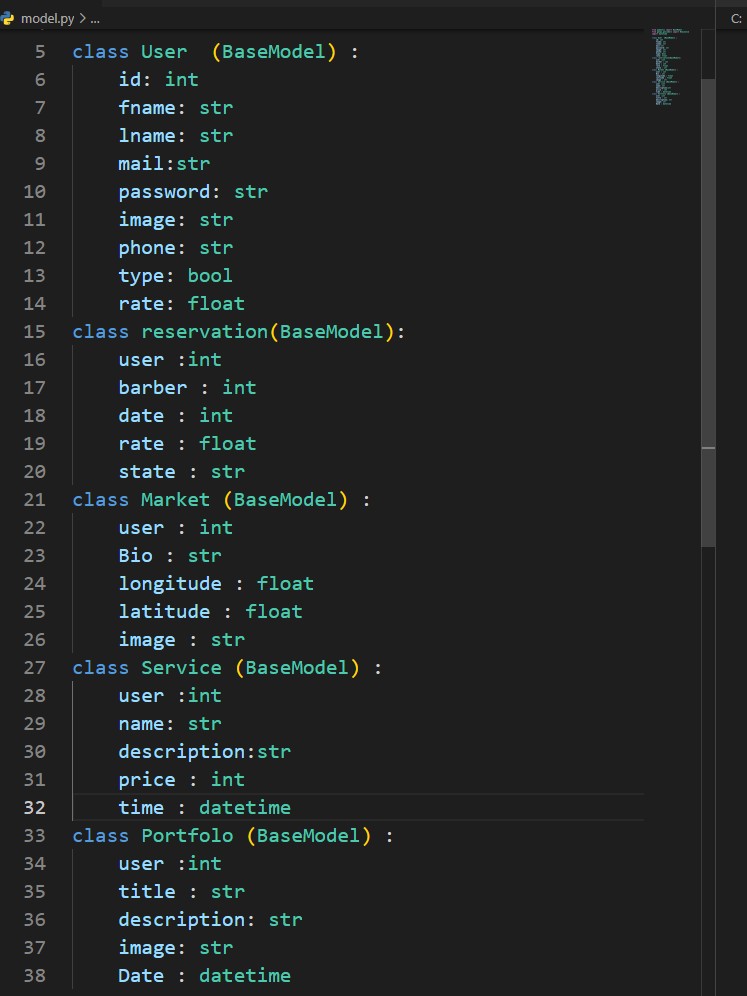
Provide and cryptography tools and Authentication Method

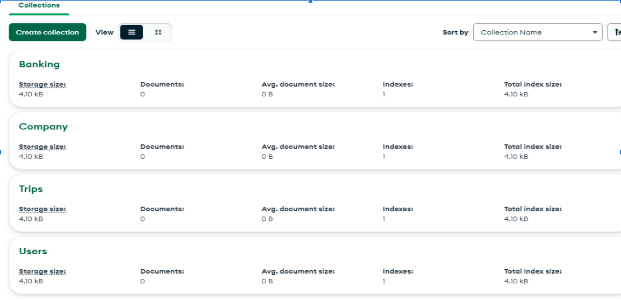
Provide Bcrypt algorithm for cryptography and Authentication Method

* **لأغراض التعامل مع طلبات الويب ودعم التعامل مع JSON :**

وبذلك نكون قد انتهينا من تنصيب البرمجيات الاساسية للعمل وننطلق إلى المرحلة الثالثة:

## **ثالثا: تعريف النماذج Models التي نحتاجها في النظام :**

باعتبار اننا نعمل على قاعدة بيانات غير علائقية وتحديداً MongoDB فإن عملية تصميم قاعدة البرمجيات تتم بشكل مباشر عن طريق تعريف الكيانات بلغة البرمجة المستخدمة، بحيث يقوم محرك قاعدة البيانات في Database Driver بتصميم البنية للقاعدة وفقاً لبرمجة الصفوف، الامر مشابه لما تقوم به Entity Framework في .NET مع اختلاف انه العملية هنا تكون runtime بحيث يمكننا انشاء صفوف جديدة والتي هي في مانغودبي collections .



D

في الصورة السابقة توضح كيف تم تنجيز الصفوف )غير العلائقية( بلغة بايثون اعتمادا Base ً Model والذي يساعدنا في ارسال النماذج إلى قاعدة البيانات mongoDB

## **رابعاً تجهيز نماذج الاتصال مع قاعدة البيانات لاستخدامها ضمن توابع ال Models APIs:**

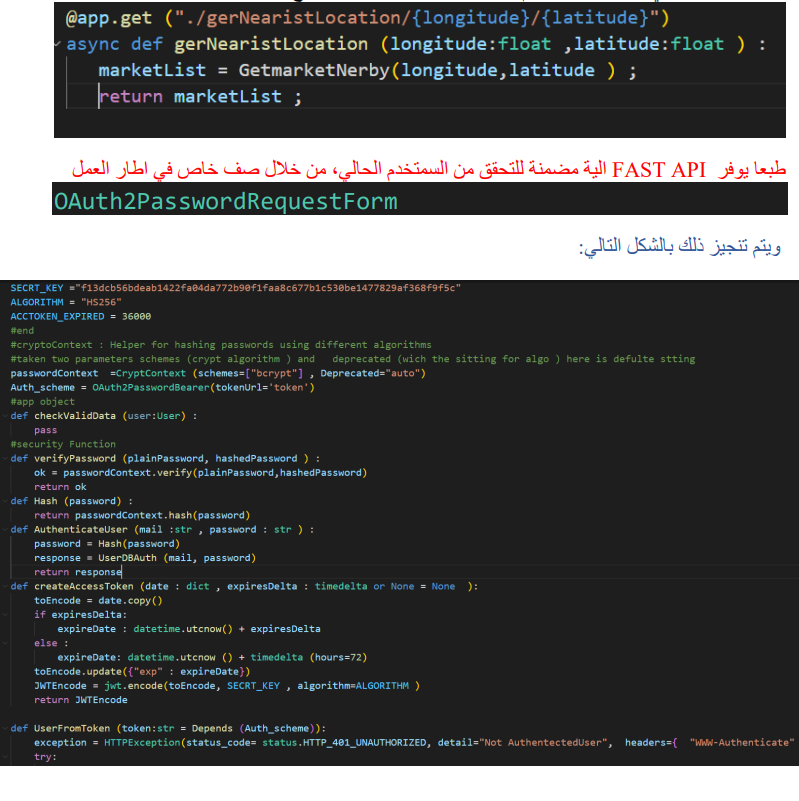
إن عملية التواصل مع قاعدة البيانات ستكون عبر الحزمة البرمجية كما تحدثنا سابقا Motor والتي تسمح لنا بكتابة تعليمات الاستعلام الخاصة ب mongoDB بلغة بايثون ذلك، حيث يتم استخدام هذه الحزمة، بعد تنزيلها كما تم التوضيح سابقا، يكفي أن نستدعيها بالشكل التالي:



## **خامساً: توابع APIs / controllers**



أو مثلاً ال API التالي للاستعلام عن أقرب محطة إلى x,y :

****

**.**

# **الفصل السادس**

# **النتائج والتوصيات**

## **تحليل النتائج:**

يمكن تلخيص النتائج التي تم الوصول إليها بالنقاط التالية:

* **نظم حجز التذاكر الشاملة هي من أساسيات الانتقال إلى مجتمعات المعرفة.**
* **نظم حجز التذاكر بحاجة ماسة لبنية إلكترونية متكاملة ( مثل خدمات الدفع الإلكتروني) ، وإلا أصبحت بلا معنى**
* **تطور التقنيات المترابط أتاح لنا تطبيق النظم بسهولة تامة، حيث أصبحت التوابع الجاهزة ( لتشفير البيانات، او الاستعلام عنها أمر في غاية من السلاسة).**
* **يجب أن يتم إضافة واجهات هاتف النقال ليكون النظام بفعاليته الكاملة.**
* **يجب على النظام ذو المهام ذات القيمة المرتفعة أن يكون بأبسط شكل بحيث يؤدي تلك المهام بأسرع وابسط شكل**
* **تلعب أدوات اختبار النظام، مثل selenium ide دوراً هاماً في عملية الاختبار، بما تحققه من توثيق وأتمتة عملية الاختبار.**

## **الآفاق المستقبلية:**

أحد أبرز الآفاق المستقبلية للنظام هو الاستعاضة عن الوثائق الورقية في الحجز، وتفعيل حلول إلكترونية، كالتحقق من خلال رمز QR ، وغيرها.

بالإضافة إلى تطوير واجهات هاتف نقال تتيح استهداف الشريحة الأكبر من الزبائن المستهدفين من نظام حجز التذاكر.

استخدام نظم الذكاء الاصطناعي ونظم ذكاء الأعمال في تعزيز استخدام الشركات للنظام.

# **الفصل السابع**

# **الخاتمة**

**الخاتمة**

تعتبر منصات حجز التذاكر من المنصات الأساسية في عصر مجتمعات المعرفة، كونها شكلت تكنولوجيا أساسية والتي حققت صلة وصل سريعة ما بين شركات النقل وعملائها بما يحقق ذلك من فائدة جمة للطرفين، فمن جهة تتيح للشركة من استعراض خدماتها على أوسع نطاق وبأسهل طريقة، مما يحقق لها مكاسب مادية إضافية، ومن جهة ثانية تتيح للمستخدمين تجربة أفضل في اختيار رحلاتهم بما يحقق غاياتهم.

وقد ارتكز هذا البحث على التكنولوجيا الحديثة المتاحة لتوفير نظام يطبق مفهوم الحجز الإلكتروني المتكامل في مختلف شركات النقل. حيث يمكن لنظام المسافرين من عملية الحجز وإتمام الدفع من خلال نظام المدفوعات الإلكترونية ، بكل سهولة من خلال رقم يتم إنشاؤه من عملية الحجز يستخدمه مستخدمو النظام للدفع ، كما يوفر إمكانية التطور مع النظام من خلال إضافة تقييمات للرحلة من قبل مستخدمي النظام.

كل هذه الخدمات تسعى إلى التطور أنظمة الحجز الإلكتروني لتلبي الاحتياجات في سوريا من خلال الاستفادة من تقنيات البرمجيات والأدوات المتوفرة لدينا، وقد تم استخدام التكنولوجيا التي تناسب النظام حيث تم تصميم الواجهات من خلال react frameworks ، والبرمجة الخلفية باستخدام Fast API ، بحيث يكون النظام قابل للتطوير وأجزاءه قابلة لإعادة الاستخدام.

.

## **الأدوات المستخدمة**

* **Fast API Python Framework v: 0.65.1**
* **UVICORN** is an ASGI [[2]](#footnote-2)web server implementation for Python.**== 0.14.0**
* **MOTOR** is a full-featured MongoDB driver for Python ==2.4.0
* Vs. code IDE : <https://code.visualstudio.com/download>
* MongoDB (Compass , and MongoDB Shell, MongoDB Community Server): <https://www.mongodb.com/try/download/community>
* Python Programming Language : <https://www.python.org/downloads/>
* Anaconda Package Manager:

<https://www.anaconda.com/products/distribution>

**المراجع:**

**uvicorn is an ASGI (async server gateway interface) compatible web server.**

**Implementation and Explanation:** [**https://www.topcoder.com/thrive/articles/bcrypt-algorithm**](https://www.topcoder.com/thrive/articles/bcrypt-algorithm)

**Fast API Documentation | www.fastapi.tiangolo.com**

**React . JS Documentation | www. legacy.reactjs.org/docs/getting-started.html**

**Mongo DB documentation |** [**https://www.mongodb.com/cloud/atlas/lp/try4?utm\_source=google&utm\_campaign=search\_gs\_pl\_evergreen\_atlas\_general\_prosp-brand\_gic-null\_ww-multi\_ps-all\_desktop\_eng\_lead&utm\_term=mongodb%20docs&utm\_medium=cpc\_paid\_search&utm\_ad=e&utm\_ad\_campaign\_id=7326400240&adgroup=150907573914&cq\_cmp=7326400240&gad=1&gclid=CjwKCAjwyeujBhA5EiwA5WD7\_QVdvD0-QXnA70n4tHGm9EhzfMSQYNjrFGlZ-a70cJAyECPHnqLLihoCUr4QAvD\_BwE**](https://www.mongodb.com/cloud/atlas/lp/try4?utm_source=google&utm_campaign=search_gs_pl_evergreen_atlas_general_prosp-brand_gic-null_ww-multi_ps-all_desktop_eng_lead&utm_term=mongodb%20docs&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=7326400240&adgroup=150907573914&cq_cmp=7326400240&gad=1&gclid=CjwKCAjwyeujBhA5EiwA5WD7_QVdvD0-QXnA70n4tHGm9EhzfMSQYNjrFGlZ-a70cJAyECPHnqLLihoCUr4QAvD_BwE)

**نظم حجز التذاكر والدفع الإلكتروني**

[**https://www.academia.edu/30931381/%D9%86%D8%B8%D8%A7%D9%85\_%D8%A7%D9%84%D8%AD%D8%AC%D8%B2\_%D9%88%D8%A7%D9%84%D8%AF%D9%81%D8%B9\_**](https://www.academia.edu/30931381/%D9%86%D8%B8%D8%A7%D9%85_%D8%A7%D9%84%D8%AD%D8%AC%D8%B2_%D9%88%D8%A7%D9%84%D8%AF%D9%81%D8%B9_)

**انتهى بعون الله**

1. uvicorn is an ASGI (async server gateway interface) compatible web server. [↑](#footnote-ref-1)
2. uvicorn is an ASGI (async server gateway interface) compatible web server. [↑](#footnote-ref-2)