* 1. **المقدمة :**

مع النمو السكاني غير المنضبط ، تصعب طرق التنقل من مكان الي اخر . الزيادة في عدد السكان تسببت في نمو كبير في حركة المرور في كل طريق من طرقات مدينة الخرطوم . هذا بدوره يساهم في إهدار الوقود الثمين والوقت الذي يؤدي إلى نفاد الصبر وإحباط المواطنين.

الازدحام المروري هومشكلة شائعة نشأت بسبب الزيادة في عدد المركبات على الطرق. من أجل التعامل مع هذه المشكلة ، اقترح الباحثون العديد من الحلول. واحد من هذه النماذج المستخدمة حاليا هي نظام اشارة المرور الذي يعمل حاليا في جميع انحاء السودان .

سبب الازدحام في حركة المرور يعتمد على الكثير من العوامل مثل وقت الذروة ، سوء الاحوال الجوية ، أوأحداث غير متوقعة مثل الحوادث والمناسبات الخاصة أوالأنشطة الثقافية .

عندما نجد انفسنا داخل ازدحام ما نحاول جاهدين بالتخلص من ذلك الازدحام ولكن غياب المعلومة عنا وعن شرطة المرور يجعلنا نتخذ خيارات تأزم من عملية الازدحام ، هنا يأتي دور التقنيات الحديثة في حل مثل هذه المشاكل المرورية .

* 1. **مشكلة البحث :**

إن مشكلة الزحمة في الطرقات هي مشكلة تعاني منها اغلب المدن الكبيرة التي تحتوي علي تعداد سكاني عالي ومنها مدينتنا الخرطوم التي تعاني في هذه الأيام من زحمة مرورية عالية تصعب علي شرطة المرور حلها .وعلي عكسنا تماما تستخدم المدن الكبري الأخري التقنية الحديثة في حل هذا الاختناق المروري والتسهيل علي رجل المرور لحل نقاط الازدحام.

وتتمثل المشاكل في الاتي :-

* + 1. المشاكل التي تواجه شرطة ادارة المرور :

1. عدم معرفة نقاط الزحمة.
2. عدم معرفة ساعات الذروة.
   * 1. **المشاكل التي تواجه هيئة الطرق والجسور :**
3. **عدم المعرفة الدقيقة لإنشاء الطرق المستقبلية.**
4. **عدم معرفة المدن التي تتطلب طرق جديدة نسبة لازدياد التعداد السكاني.**
   * 1. المشاكل التي تواجه المواطن :
5. عدم معرفة الطرق التي من المتوقع ان تكون بها زحمة.
   1. أهمية البحث :

تتمثل أهمية البحث في توفير كل المعلومات المتعلقة بحركة السير العامة وتوفير البيانات اللازمة لمنع حدوث الاختناقات المرورية وتقليل الزحمة .

* 1. أهداف البحث :

يمكن أن نجمل أهداف البحث في المحاور التالية :-

1. **مساعدة إدارة المرور في النقاط التالية :**
2. توجيه رجل المرور الي المناطق الأكثر ازدحاماً.
3. تحديد ساعات الذروة في كل طريق.
4. معرفة التغير في ساعة الذروة علي مدار السنة.
5. **مساعدة هيئة الطرق والجسور في النقاط التالية :**
6. تحديد الطرق التي تحتاج الي توسعة حسب حركة المرور.
7. تحديد المناطق التي ازداد عدد سكنها لإضافة طرق جديدة.
8. مساعدة المواطن في النقاط التالية :
9. معرفة الطرق المزدحمة في الوقت الحالي.
10. اقتراح انسب الأوقات للقيام برحلته.
    1. منهجية البحث :

يستخدم الباحث المنهج الوصفي التحليلي التطبيقي في حل مشكلة البحث.

* 1. **أدوات البحث :**

الأدوات المستخدمة في هذا البحث تشمل الاتي :

* ASP.Net MVC – C#
* Restfull
* Sql server
* Flutter - Dart
* Open CV – Python
* Qt Frame – C++
* Visual Paradigm uml modeling tool
  1. **الدراسات السابقة :**
     1. **حركة المرور التلقائية باستخدام معالجة الصور:**

**المؤلف :**

Al Hussain Akoum – Email: [Hussain\_alkoum@hotmail.com](mailto:Hussain_alkoum@hotmail.com)

August 11, 2017 -

**الخلاصة** :

الازدحام المروري المتكرر عند التقاطعات الرئيسية يستدعي وجود نظام إدارة فعال. تقترح الورقة تنفيذ وحدة تحكم مرور ذكية باستخدام معالجة الصور في الوقت الفعلي. يتم تحليل تسلسل الكاميرا باستخدام خوارزميات مختلفة للكشف عن الحواف وطرق عد الكائنات. في السابق استخدموا طريقة المطابقة مما يعني أنه سيتم تثبيت الكاميرا مع إشارة المرور. سوف يلتقط تسلسل الصورة. لتعيين صورة طريق فارغ كصورة مرجعية ، تتم مطابقة الصور الملتقطة بالتسلسل باستخدام مطابقة الصور ؛ لكن في ورقي ، استخدمنا طريقة التصفية ، التي قامت بتصفية الصورة وإطلاق جميع النفايات وعرض السيارات فقط ، وبعد ذلك أظهر عدد السيارات في الصورة جيدًا. هذه الورقة عبارة عن برنامج يلتقط صورة أو مقطع فيديو. تم تخصيصه لاستخدامه في المستقبل للتحكم في إشارة المرور من خلال إعطاء كل إشارة وقتًا كافيًا ، اعتمادًا على عدد السيارات في كل اتجاه.

**الأهداف :**

في الحياة العصرية ، علينا أن نواجه العديد من المشاكل ، أحدها أن الازدحام المروري يصبح أكثر خطورة يومًا بعد يوم. نتيجة للزيادة في حركة مرور المركبات ، ظهرت العديد من المشاكل ، على سبيل المثال ، حوادث المرور والازدحام المروري وما إلى ذلك. كان الازدحام المروري مشكلة صعبة للغاية. نتيجة لذلك ، اهتم العديد من المحققين بـ ITS (نظام النقل الذكي) مثل التنبؤ بتدفق حركة المرور على أساس مراقبة حركة المرور عند تقاطعات المرور لاكتشاف الاختناقات. تظل هذه المهمة تمثل تحديًا لأنظمة رؤية الكمبيوتر. تم تنفيذ العديد من الأساليب لهذه المهمة على مدى عقود عديدة.

**الطرق و الأدوات :**

سنناقش حول التحكم الذكي في حركة المرور باستخدام معالجة الصور لحساب عدد المركبات. يعد اكتشاف المركبات وحسابها مهمين في حساب الازدحام المروري على الطرق السريعة. الهدف الرئيسي من الكشف عن السيارات والعد في ورقة مرور فيديو أو صورة هو تطوير منهجية للكشف التلقائي عن المركبات وحسابها على الطرق السريعة. لا تستخدم طريقتنا الخلفية ، بل تستخدم مرشحًا نكتشف السيارات ونعدها ، وتلتقط مقطع فيديو أو صورة وتقوم ببعض المعالجة لإعطاء عدد السيارات في النهاية.

1. **كشف المركبة بالفيديو:**

تم تطوير العديد من التقنيات في معالجة الفيديو خلال العقود الأربعة إلى الخمسة الماضية. أحدهما هو تقنية المطابقة ، فهو يأخذ الصورة السابقة والصورة الحالية ثم يطرح بين هاتين الصورتين ووفقًا للاختلاف فإنه يحصل على نسبة الازدحام. لكننا نستخدم الآن تقنية التصفية التي يمكن أن تعطي نتائج دقيقة تصل إلى 90٪.

1. **كشف المركبة بالصورة :**

هناك طريقة أخرى مستخدمة في ورقي وهي استخدام معالجة الصور. معالجة الصور هي معالجة الصور باستخدام العمليات الحسابية باستخدام أي شكل من أشكال معالجة الإشارات التي يكون الإدخال فيها صورة. قد يكون ناتج معالجة الصورة إما صورة أو مجموعة من الخصائص أو المعلمات المتعلقة بالصورة. تُستخدم معالجة الصور لاكتشاف الكائن ولكن في هذه الورقة خاصةً لاكتشاف المركبات.

* + 1. **نظام التحكم في حركة المرور المعتمد على الكثافة باستخدام معالجة الصور:**

**المؤلف :**

Uthara E. Prakash ,AthiraThankappan , Vishnupriya K. T. , Arun A. Balakrishnan.

November 2018 -

**الخلاصة** :

في هذا البحث ، تم تقديم نظام جديد للتحكم في حركة المرور في الوقت الفعلي يمكنه بسهولة التحكم في حركة المرور باستخدام تقنيات معالجة الصور. في هذه الطريقة ، يتم استخدام كاميرا الويب في كل مرحلة من مراحل إشارة المرور من أجل التقاط صور للطرق التي لا بد أن تحدث فيها حركة المرور. يتم حساب عدد المركبات في هذه الصور باستخدام أدوات معالجة الصور في Matlab ويتم تخصيص أوقات مختلفة وفقًا للعدد جنبًا إلى جنب مع الإشارة الخضراء للمركبات لتمريرها. في النموذج الأولي المقترح ، يتم تمثيل الإشارات الخضراء والحمراء باستخدام مصابيح LED ويتم تمثيل مؤقت التناقص للإشارة الخضراء بعرض سبعة أجزاء.

**المشكلة :**

مع النمو السكاني غير المنضبط ، أصبح السفر مهمة محمومة حقًا في عالم اليوم. أدت الزيادة في عدد المسافرين إلى نمو كبير في حركة المرور في كل زاوية وركن من أركان المدينة. وهذا بدوره يساهم في إهدار الوقود الثمين والوقت مما يؤدي إلى نفاد صبر الناس وإحباطهم. يعد الازدحام المروري مشكلة شائعة نشأت بسبب زيادة عدد المركبات على الطريق. من أجل التعامل مع هذه المشكلة.

**الطرق و الأدوات :**

تم تنفيذ النظام المقترح في ماتلاب بهدف تقليل حركة المرور على أساس الكثافة. يتم النظر في أربع خطوات رئيسية للنظام: أ) الحصول على الصورة ب) تحويل RGB إلى التدرج الرمادي ج) تحسين الصورة و د) العمليات المورفولوجية. يتم تثبيت كاميرا واستخدامها لالتقاط الفيديو على الطريق السريع. يتم تسجيل الفيديو بشكل مستمر في إطارات متتالية وتتم مقارنة كل إطار بالصورة الأولية الملتقطة. تم اكتشاف العدد الإجمالي للسيارات الموجودة في الفيديو باستخدام خوارزميات معالجة الصور. إذا تجاوز العدد الإجمالي للسيارات حدًا محددًا مسبقًا ، يتم عرض حالة حركة المرور الكثيفة كرسالة.

**الخوارزمية :**

1. ابدأ البرنامج.

2. التقاط صورة للطريق الفارغ عن طريق الاتصال وحدة الكاميرا كمرجع.

3. التقاط صورة مع المركبات.

4. يتم تحويل الصور من RGB إلى اللون الرمادي.

5. تم إيجاد قيمة حدية باستخدام مبدأ أوتسو.

6. أوجد الفرق بين الإطارات باستخدام العتبة.

7. أضف ضوضاء Gaussian إلى إخراج الفرق.

8. قم بتطبيق مرشح Weiner عليه لتصفية النقط.

9. تحويل إلى صورة ثنائية.

10. ملء الثقوب للنقاط.

11. افتح جميع النقاط التي تزيد مساحتها عن 2000.

12. تحديد عدد السيارات.

13. عرض صورة الإخراج.

14. تم العثور على عدد المركبات وعرضها.

15. حسب عدد المركبات ، الضوء الأخضر هو مخصصة لتوقيتات مختلفة لكل عدد معروض بقطعة سبعة.

**أعمال مستقبلية :**

لا تؤخذ الظروف الجوية في الاعتبار والتي قد تؤثر على جودة الصورة عندما تصبح ضبابية أو في ظل هطول أمطار غزيرة. يمكن إجراء المزيد من التطورات على النظام المقترح للتحقق من تحديد هوية المركبات التي تمر عبر دائرة النظام والتي يمكن أن تساعد في مراقبة حركة المرور.

# مقدمة عن الرؤية بالحاسوب (Computer Vision) :

ظلت رؤية الكمبيوتر موجودة منذ أكثر من 50 عامًا ، ولكن في الآونة الأخيرة ، نشهد زيادة كبيرة في الاهتمام بكيفية "رؤية" الآلات وكيفية استخدام رؤية الكمبيوتر لإنشاء منتجات للمستهلكين والشركات. بعض الأمثلة على هذه التطبيقات هي: Amazon Go و Google Lens و Autonomous Vehicles و Face Recognition.

العامل الرئيسي وراء كل هذا هو رؤية الكمبيوتر. بعبارات أبسط ، تمثل Computer Vision مجالا واسعا من الذكاء الاصطناعي الذي يعلم الآلات أن ترى. هدفها هو استخراج المعنى من البكسلات.

من وجهة نظر العلوم البيولوجية ، تتمثل أهدافها في التوصل إلى نماذج حسابية للنظام البصري البشري. من وجهة النظر الهندسية ، تهدف رؤية الكمبيوتر إلى بناء أنظمة مستقلة يمكنها أداء بعض المهام التي يمكن أن يؤديها النظام البصري البشري (بل وتجاوزها في كثير من الحالات).

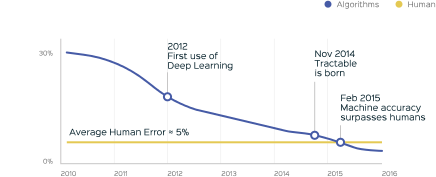
* 1. **لمحة تاريخية :**

في صيف العام 1966 ، بدأ سيمور بابيرت ومارفين مينسكي من مجموعة MIT Artificial Intelligence مشروعًا بعنوان مشروع الرؤية الصيفية. كان الهدف من المشروع هو بناء نظام يمكنه تحليل المشهد وتحديد الكائنات في المشهد. لذا فإن المجال الواسع المحير المتمثل في رؤية الكمبيوتر الذي لا يزال الباحثون وعمالقة التقنية يحاولون فك تشفيره ، كان يُعتقد أولاً أنه بسيط بما فيه الكفاية لمشروع صيفي لمرحلة ما قبل التخرج من قبل نفس الأشخاص الذين كانوا رواد في مجال الذكاء الاصطناعي.

في سبعينيات القرن الماضي ، أخذ ديفيد مار ، وهو عالم الأعصاب في معهد ماساتشوستس للتكنولوجيا ، الذي أخذ أفكارًا من دراسات المخيخ والحصين ، اللبنات الأساسية لرؤية الكمبيوتر الحديثة ، وبالتالي يعرف باسم والد رؤية الكمبيوتر الحديثة. تتوج غالبية أفكاره في الكتاب الذي يحمل عنوان VISION.

* 1. **الرؤية العميقة (Deep Vision) :**

لقد بدأ التعلم العميق (Deep Learning) منذ عام 2012. التعلم العميق هو مجموعة فرعية من التعلم الآلي (machine learning) حيث تتعلم الشبكات العصبية الاصطناعية ، الخوارزميات المستوحاة من الدماغ البشري ، من كميات كبيرة من البيانات. تشغيل أنظمة التوصية ، وتحديد الأصدقاء في الصور ووضع علامات عليهم ، وترجمة صوتك إلى نص ، وترجمة النص إلى لغات مختلفة ، وقد حوّل التعلم العميق رؤية الكمبيوتر إلى أداء أكثر تفوقاً.

****

**معدل الخطأ في تصنيف الصور مع مرور الوقت ، انخفاض حاد بعد إدخال التعلم العميق**

* 1. **استخدامات رؤية الحاسوب :**
* الهواتف الذكية: أكواد QR ، والتصوير الحسابي (Android Lens Blur ، iPhone Portrait Mode) ، إنشاء بانوراما (Google Photo Spheres) ، اكتشاف الوجه ، اكتشاف التعبير (الابتسامة) ، فلاتر Snapchat (تتبع الوجه) ، Google Lens ، Night Sight (Pixel).
* الويب: البحث عن الصور ، صور Google (التعرف على الوجوه ، التعرف على الأشياء ، التعرف على المشهد ، تحديد الموقع الجغرافي من الرؤية) ، Facebook (شرح الصورة) ، خرائط Google للتصوير الجوي (خياطة الصورة) ، YouTube (تصنيف المحتوى).
* VR / AR: التتبع الخارجي (HTC VIVE) ، التتبع الداخلي (التعريب والتخطيط المتزامنان ، HoloLens) ، انسداد الكائنات (تقدير العمق الكثيف).
* التصوير الطبي: إعادة بناء CAT / التصوير بالرنين المغناطيسي ، والتشخيص بمساعدة ، وعلم الأمراض التلقائي ، و connectomics ، والجراحة الموجهة بالذكاء الاصطناعي (AI-guided surgery).
* الوسائط: التأثيرات المرئية للفيلم ، والتلفزيون (إعادة الإعمار) ، وإعادة الرياضة الافتراضية (إعادة الإعمار) ، والتعديلات التلقائية المستندة إلى دلالات (إعادة الإعمار ، والاعتراف).
* التأمين: أتمتة المطالبات ، تحليل الأضرار ، فحص الممتلكات.
  1. **التحديات :**
* حتى بعد نشر قدر كبير من العمل ، لا يتم حل رؤية الكمبيوتر. إنه يعمل فقط في ظل بعض القيود. أحد الأسباب الرئيسية لهذه الصعوبة هو أن النظام البصري البشري هو ببساطة جيد جدًا للعديد من المهام مثل التعرف على الوجوه. يمكن للإنسان التعرف على الوجوه تحت جميع أنواع الاختلافات في الإضاءة ، وجهة النظر ، التعبير ، وما إلى ذلك الذي يعاني منه الكمبيوتر في مثل هذه الحالات.
* الخصوصية والأخلاق - أثناء استخدام المراقبة ، تشتمل أحدث قطاعات التأمين على تعديل الأقساط والسياسات من خلال مراقبة سلوك القيادة ، لكن من ناحية أخرى ، تشكل أنظمة المراقبة التي تعمل بالطاقة Vision مخاطر كبيرة على الخصوصية والأخلاق. كمثال ، نرى الصين ، باستخدام التعرف على الوجه لتتبع الأقليات العرقية. في الآونة الأخيرة ، أصبحت سان فرانسيسكو أول مدينة أمريكية تمنع استخدام التعرف على الوجه من قبل حكومتها.
* عدم وجود تفسير - لا تزال الخوارزميات الحديثة القائمة على الشبكة العصبية صندوقًا أسودًا إلى حد ما. لذلك عندما يصنف نموذج صورة كسيارة ، فإننا لا نعرف في الواقع سبب قيامها بذلك. تعد "قابلية التفسير" شرطا أساسيا في العديد من المجالات مثل التأمين والقيادة الذاتية التي تفتقد حاليا في هذه الخوارزميات.
* Deep Fakes -. باستخدام تقنيات التعليم العميق ، يمكن الآن لأي شخص لديه بيانات GPU وبيانات تدريب قوية إنشاء صورة مزيفة أو مقاطع فيديو يمكن تصديقها باستخدام DeepFakes. هذه المشكلة خطيرة للغاية ، حيث يعمل البنتاغون ، من خلال وكالة مشاريع الأبحاث المتقدمة للدفاع (DARPA) ، مع العديد من أكبر المؤسسات البحثية في البلاد لمعالجة DeepFakes.
* هجمات الخصوم (Adversarial attacks) - أمثلة الخصم هي مدخلات لنماذج التعلم الآلي التي صممها المهاجم عمدا لتتسبب في ارتكاب أخطاء ؛ مثل الأوهام البصرية للآلات.
  1. **مستقبل رؤية الكمبيوتر :**

وفقًا لتقرير ، بلغت قيمة سوق رؤية الكمبيوتر 2.37 مليار دولار أمريكي في عام 2017 ، ومن المتوقع أن تصل إلى 25.32 مليار دولار أمريكي بحلول عام 2023 ، بمعدل سنوي مركب نسبته 47.54٪.

يشهد العالم تحولًا رقميًا عميقًا ، خاصةً الهند التي لا تظهر عليها أي علامات على التباطؤ. متوسط استهلاك البيانات الشهري لـ Jio وحده هو 10.8 جيجابايت. وفقًا لهذا التقرير ، كل دقيقة:

* شاهد المستخدمين 4،146،600 مقطع فيديو على YouTube.
* ينشر مستخدمو Instagram 46،740 صورة.
* يشارك مستخدمو Snapchat 527،760 صورة.

التي تعطي جميعها مجموعة كبيرة من الفرص لرؤية الكمبيوتر لتجد أنماطًا وتفهمها.

حتى مع كل التطورات الرائعة ، يحتاج الذكاء الاصطناعى ومجال رؤية الكمبيوتر على وجه التحديد إلى معالجة المشاكل المرتبطة به حاليًا مثل التحيز ، وخطر عدم الوعي وعدم وجود تفسير. لمعالجة مثل هذه المشكلات ، بدأت شركات مثل Ping An في اتخاذ خطوات صغيرة ، باستخدام Symbolic AI ، وهو شكل مبكر من الذكاء الاصطناعي ، في خوارزميات الذكاء الاصطناعى الحديثة لإعطائها القدرة لتوضيح قرارها ، ولكن لا يزال هناك الكثير للقيام به.

**3-1 تحليل النظام :**

يتكون النظام من مجموعة من الكاميرات الموزعة على تقاطعات الشوارع (في بداية و نهاية كل شارع), حيث يقوم برنامج الرؤية الحاسوبية بعمل معالجات على الفيديو و من ثم اكتشاف و تتبع المركبات المارة و إتجاها.

بعد ذلك يقوم البرنامج بارسال معلومات عدد المركبات التي مرت خلال ذلك الطريق الى الخادم في فترة دورية (كل 30 ثانية مثلاَ) ليتم تحديث قاعدة البيانات واستخدام تلك المعلومات لتحقيق اهداف المشروع و عرضها للمستخدمين من خلال تطبيق هاتف (Android).

**3-1-1 المخرجات :**

1. **مخرجات خوارزمية الرؤية بالحاسوب :**

* عدد المركبات المارة في خطي السير خلال فترة التحديث الزمنية المحددة (up and down Counter).

1. مخرجات النظام (تطبيق المستخدم) :

تختلف مخرجات النظام بحسب صلاحيات المستخدم, ما يلي مخرجات النظام لكل نوع من أنواع المستخدمين :

1. مخرجات مشتركة لكل المستخدمين :

* لائحة الطرق المسجلة في النظام (التي بها كاميرات مراقبة) و معلومات نسب ازدحامها.
* خريطة لعرض تمثيل نسب الازدحام على الطرق.
* عرض ساعات الذروة لكل طريق.

1. إدارة و شرطة المرور :

* لائحة أكثر تفصيلاَ للطرق المسجلة تحتوي على الأرقام الفعلية للمركبات على الطريق.
* تقرير زمني للذروة حسب الساعة أو أيام الأسبوع أو الشهور مرتبة حسب الأكثر إزدحاماَ.

1. هيئة الطرق و الجسور :

* يقوم التطبيق بعرض معلومات المدن مصنفة حسب الأكثر إزدحاما و بالتالي تحتاج الى تحسين البنية التحتية لتلك المدينة.

**3-1-2 المعالجات :**

1. **خوارزمية معالجات برنامج الرؤية بالحاسوب :**
2. تخزين الإعدادات الاولية.
3. تشغيل الكاميرا واستقبال بث الفيديو.
4. حلقه التشغيل (لكل صورة في الفيديو) :

* قطع منطقة الأهمية (Region Of Interest) , و هي الجزء من الصورة الذي يتم تطبيق خوارزمية التعرف عليه فعلياَ و يتم إهمال باقي الصورة. فمثلاَ قد يكون الطريق في منتصف الصورة و يكون ممر المشاة على الجانبين.
* المعالجه الاوليه للصورة (blur) حيث ساعد ذلك في تقليل الضوضاء.
* التحويل في نظام اللون الرمادي (gray scale).
* تطبيق خوارزميه استخراج الخلفية (Background Subtractor).
* ايجاد الحدود (contours) للاشكال المستخرجة وتخزينها في مصفوفة.
* بناء مصفوفة (newVehicles[]) لتخزين معلومات المركبات باستخدام مصفوفة الاشكال بشرط ان يكون الطول والعرض للشكل في مدى محدد وأن لا يوجد شكل داخل شكل اخر. تتكون عناصر هذه المصفوفة من كائنات لفئة "مركبة" تحتوي على المتغيرات التالية:
  + الحالة (state), تستخدم في خوارزمية التتبع لمعرفة حالة التتبع للمركبة, الحالات لهذا المتغير هي : {matched, unmatched, new}.
  + الاسم, حيث يتم توليد اسم عشوائياَ من ثلاثة حروف لتسهيل عملية التطوير و تصحيح الأخطاء (debugging).
  + متغيرات الاحداثيات (x, y, w, h) , لتحديد مكان و طول و عرض المركبة.
  + centerX, centerY, لتحديد نقطة مركز المركبة.
  + الاتجاه (direction), لتحديد اتجاه حركة المركبة {up, down}.
  + مصفوفة لتخزين أماكن المركبة علي مرور الزمن (pointsHistory).
  + رقم يحدد قيمة المركبة (contains), تكون القيمة الافتراضية 1 و تستخدم هذه القيمة في حساب عداد المركبات.
  + متغير يحدد ما إذا كانت هذه المركبة قد غيرت إتجاهها (directionChanged).
* خوارزمية التتبع (tracking) :
  + اجعل حالة كل المركبات في مصفوفة المركبات (Vehicles[]) تساوي "unmatched".
  + لكل مركبة في مصفوفة المركبات المكتشفة (newVehicles[]):  
    إذا كانت مصفوفة المركبات فارغة: قم بإضافة المركبة الحالية الى مصفوفة المركبات و ضع الحالة = "new".  
    و إلا :
    - ضع المتغير (closest) يساوي اقرب مركبة غير مكتشفة (unmatched) للمركبة الحالية.
    - إذا كانت (closest) قريبة من المركبة الحالية قي حدود (proximity threshold) محدد:
      * قم بنسخ قيم متغيرات (closest) للمركبة الحالية و غير قيمة الحالة الى "matched".
      * قم بتحديث متغيرات المركبة الحالية (pointsHistory and direction).
      * قم باستبدال المركبة الحالية بالمركبة (closest) في مصفوفة المركبات (Vehicles).

و إلا :

قم بإضافة المركبة الحالية الى مصفوفة المركبات و ضع الحالة = "new".

* التعامل مع المركبة الغير معرفة (unmatched):
  + إذا كانت المركبة قريبة من حدود منطقة الرؤية في الصورة (roi) :فيتم اضافة قيمتها للعداد حسب اتجاه سير المركبة. ثم حذفها من المصفوفة.  
    و إلا فيتم حذفها فقط. و تجاهل قيمتها (يحدث ذلك عند حدوث ضوضاء في الفيديو أو اي حركة غريبة لا تشبه سلوك المركبات الطبيعي).
* قم برسم مربعات حول المركبات المكتشفة و طباعة معلوماتها على الشاشة.
* قم بارسال معلومات العداد الى الخادم بعد انقضاء فترة التحديث. و تصفير العداد في حالة ارسال المعلومات بنجاح. (تتضمن هذه المعلومات : رقم التعريف بالكاميرا و عدادين خطي السير في كلا الإتجاهين).

1. معالجات الخادم و قاعدة البيانات :

* إدارة المستخدمين و تشمل عمليات الإضافة و الحذف و التعديل.
* إدارة الكاميرات.
* إدارة الطرق و المدن.
* تخزين مخرجات كل برامج الرؤية الطرفية في قاعدة البيانات.
* معالجة طلبات تطبيق المستخدم.

1. معالجات تطبيق المستخدم :

* يقوم التطبيق بعمليات الدخول و التسجيل لمستخدمي النظام.
  + 1. **المدخلات :**

1. معلومات الإعدادات الأولية لرؤية الحاسوب , يتم ادخال هذه المعلومات عند بداية تشغيل البرنامج لأول مرة و تشمل :

* رقم جهاز الكاميرا (camera device id) , في حالة اذا كان الحاسوب متصلا بأكثر من كاميرا واحدة.
* الرقم التعريفي لموقع الكاميرا (camera client id) , حيث يتم ارسال هذا الرقم للخادم ليتمكن من معرفة مصدر المعلومات.
* تحديد أبعاد منطقة الأهمية (roi).
* فترة التحديث, الفترة الدورية لارسال معلومات خوارزمية التعرف و تحديث الخادم.
* موقع الخادم (server url)

1. تسجيل الفيديو من الكاميرا أو فيديوهات مسجلة مسبقاً في مرحلة التطوير والاختبار.
2. تسجيل معلومات مستخدمي النظام.
3. ادخال معلومات النظام (المدن و الطرق و الكاميرات).
   1. **تصميم النظام :**
      1. **مكونات النظام :**

يتكون النظام من ناحية معمارية الى ثلاث اجزاء هي :

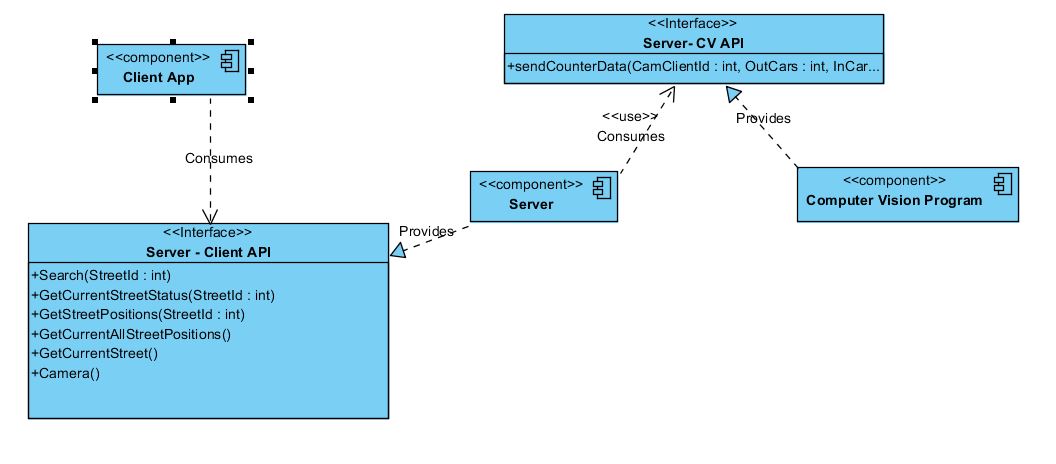
* + الخادم المركزي.
  + برامج الرؤية بالحاسوب.
  + تطبيق المستخدم.

كذلك من ناحية الأشخاص الذين يتعاملون مع النظام (actors) الى :

* + مشرف النظام (System Admin).
  + المستخدمين.
    1. **تصميم المخططات :**

1. مخطط المكونات (Components Diagram) :

صورة (3-1) : مخطط المكونات و واجهات اتصالها



1. مخطط التتابع التسلسلي (Sequence Diagram) :

صورة (3-2) : مخطط التتابع التسلسلي



* + 1. **تصميم الخوارزمية :**

شكل (3-3) : مخطط خوارزمية الرؤية

(حلقة 1) قطع منطقة الأهمية

التحويل في نظام اللون الرمادي

تطبيق خوارزميه استخراج الخلفية

المعالجه الاوليه للصورة (blur)

تشغيل الكاميرا واستقبال بث الفيديو

تخزين الإعدادات الاولية

ايجاد الحدود (contours) للاشكال المستخرجة وتخزينها في مصفوفة

إذا كانت مصفوفة المركبات فارغة

(حلقة2) لكل مركبة في مصفوفة المركبات المكتشفة (newVehicles[])

اجعل حالة كل المركبات في مصفوفة المركبات (Vehicles[]) تساوي "unmatched".

بناء مصفوفة (newVehicles[]) لتخزين معلومات المركبات باستخدام مصفوفة الاشكال

(نعم) -< قم بإضافة المركبة الحالية الى مصفوفة المركبات و ضع الحالة = "new"

(لا)-< قم بإضافة المركبة الحالية الى مصفوفة المركبات و ضع الحالة = "new"

قم باستبدال المركبة الحالية بالمركبة (closest) في مصفوفة المركبات (Vehicles).

قم بتحديث متغيرات المركبة الحالية (pointsHistory and direction).

(نعم)-< قم بنسخ قيم متغيرات (closest) للمركبة الحالية و غير قيمة الحالة الى "matched".

إذا كانت (closest) قريبة من المركبة الحالية قي حدود (proximity threshold) محدد

ضع المتغير (closest) يساوي اقرب مركبة غير مكتشفة (unmatched) للمركبة الحالية.

إذا تمت معالجة جميع عناصر مصفوفة المركبات المكتشفة

(لا)-< إذهب الى بداية (الحلقة 2)

فيتم حذفها فقط. و تجاهل قيمتها

(نعم)-< فيتم اضافة قيمتها للعداد حسب اتجاه سير المركبة. ثم حذفها من المصفوفة

(نعم) -< إذا كانت المركبة قريبة من حدود منطقة الرؤية في الصورة (roi)

قم برسم مربعات حول المركبات المكتشفة و طباعة معلوماتها على الشاشة

هل تم الضغط على زر الخروج "q" ؟

إذا تم الإرسال بنجاح

(نعم)-< قم بتصفير العداد في حالة ارسال المعلومات بنجاح

قم بارسال معلومات العداد الى الخادم بعد انقضاء فترة التحديث

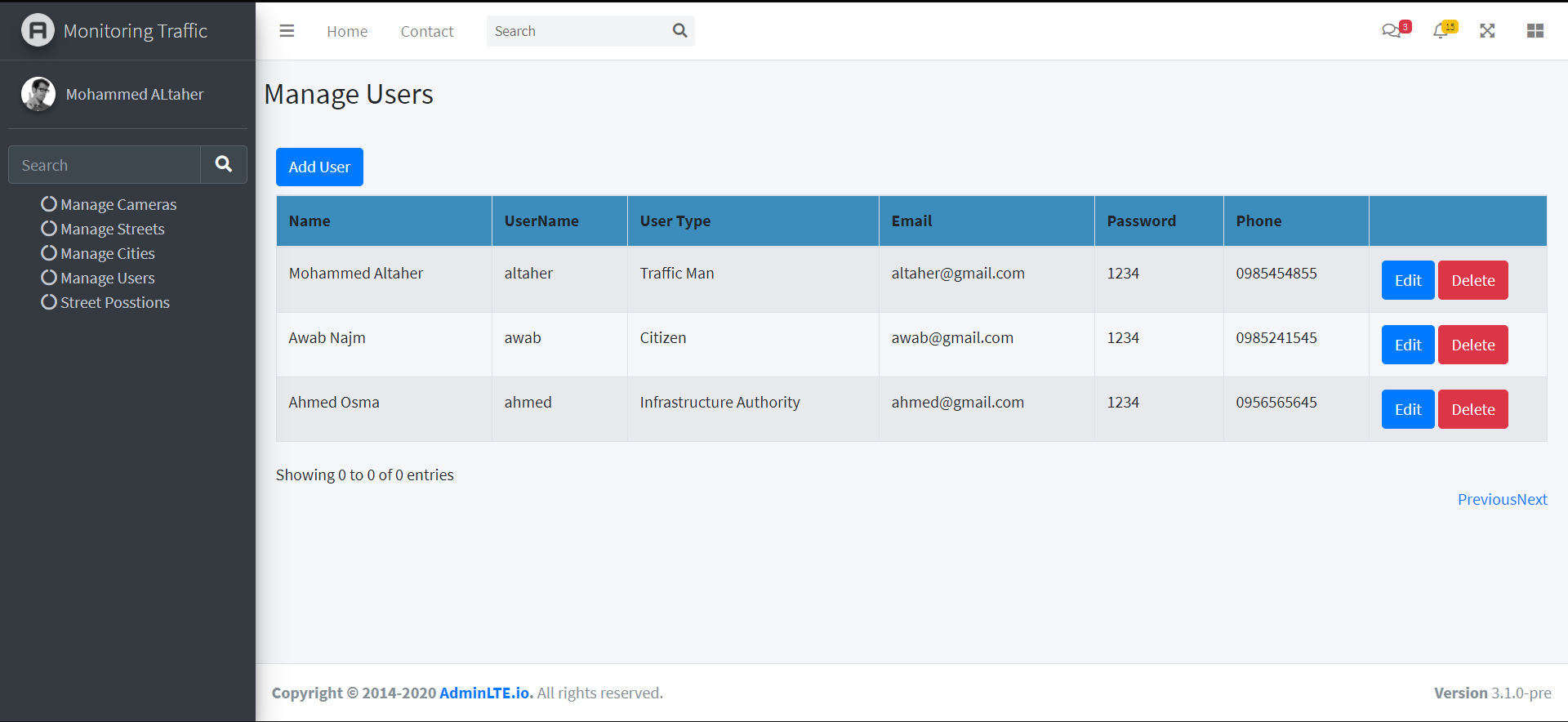
إذهب إلى بداية (الحلقة 1)

* 1. **التنفيذ :**

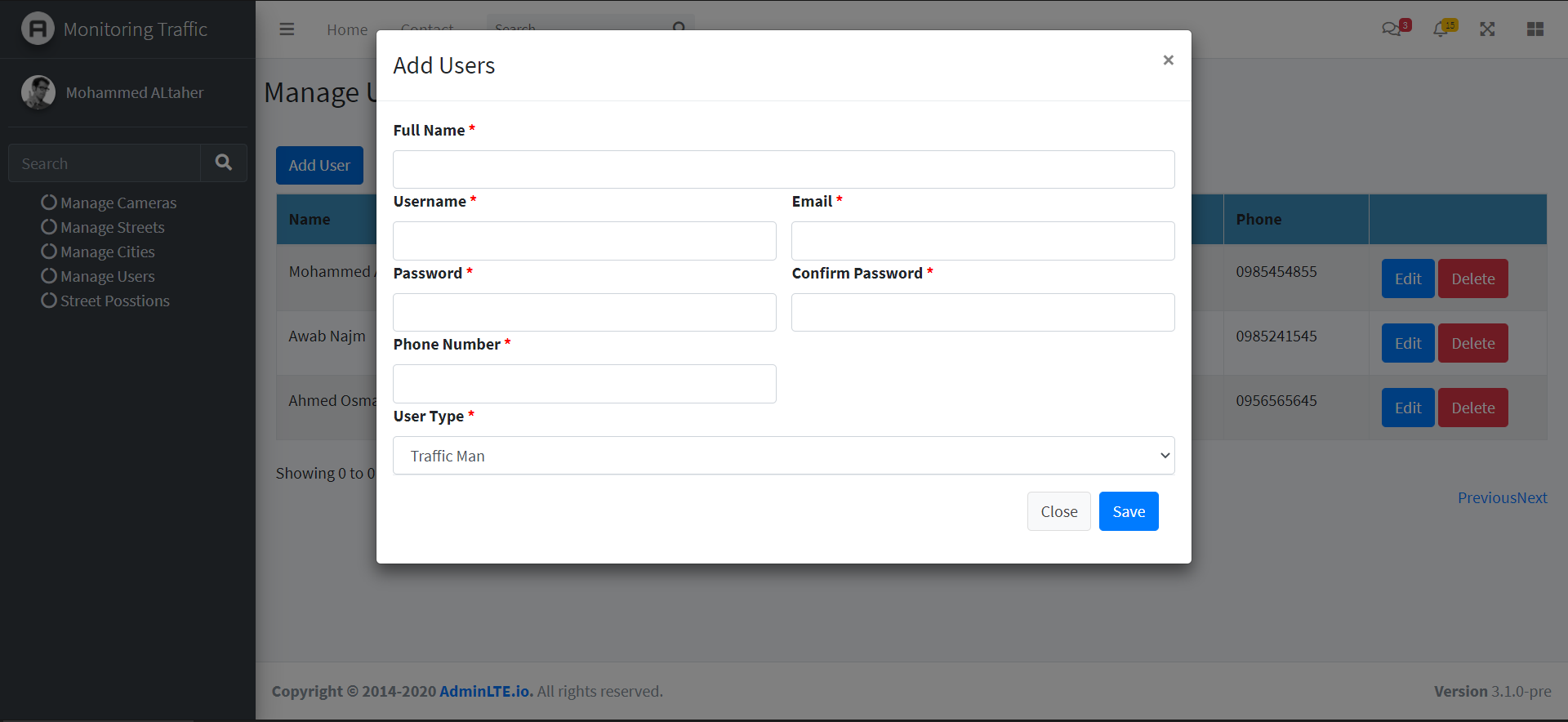
4-1-1 أولا يقوم مشرف النظام (System Admin) بإدخال معلومات النظام الأساسية باستخدام لوحة التحكم على الخادم, و تتكون من :

1. تسجيل بيانات المستخدمين (حسابات إدارة المرور و هيئة الطرق) :
   * + - الاسم الكامل
       - اسم المستخدم
       - كلمة السر
       - رقم الهاتف
       - نوع الحساب
       - الايميل

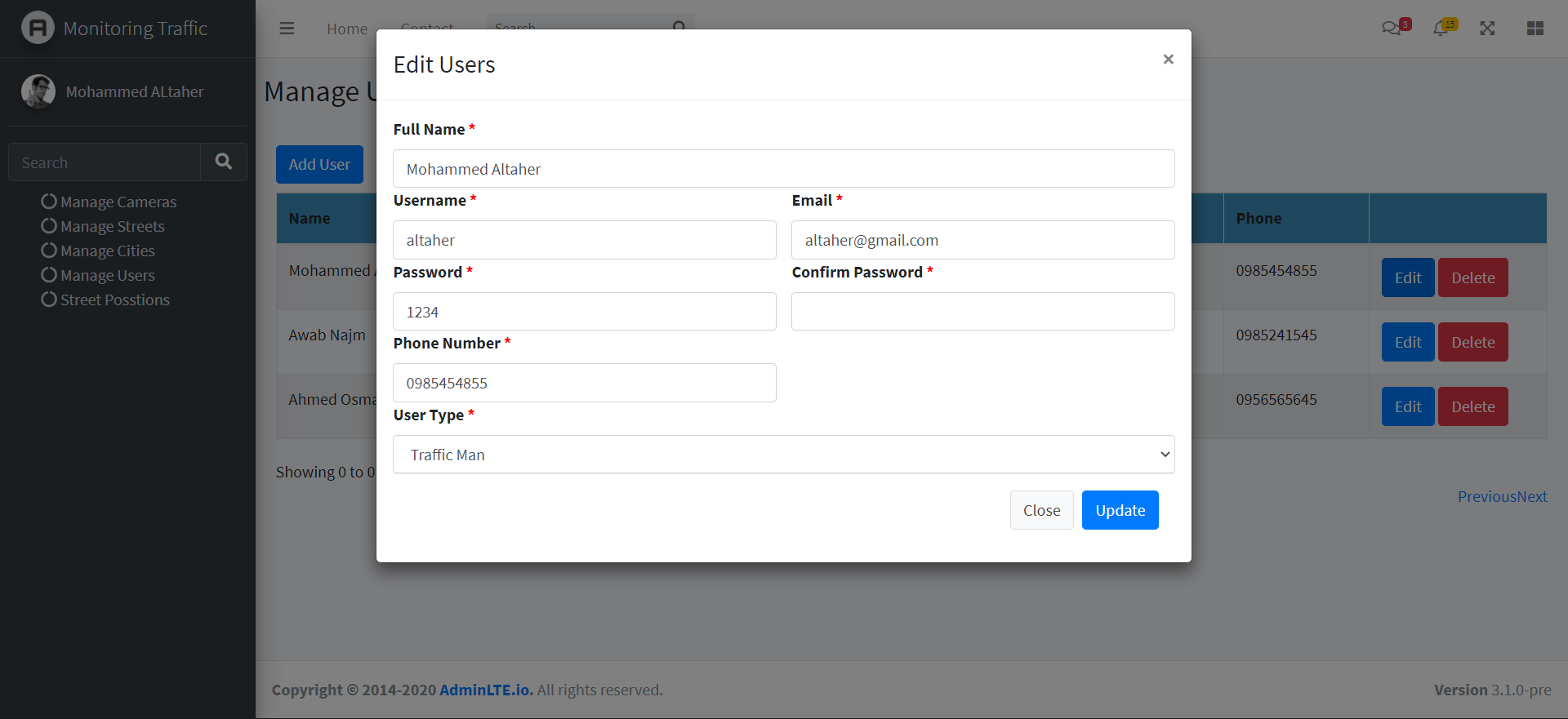
الصورة (4-1) : إدارة المستخدمين



الصورة (4-2) : إدارة المستخدمين – إضافة مستخدم



الصورة (4-3) : إدارة المستخدمين – تعديل بيانات مستخدم

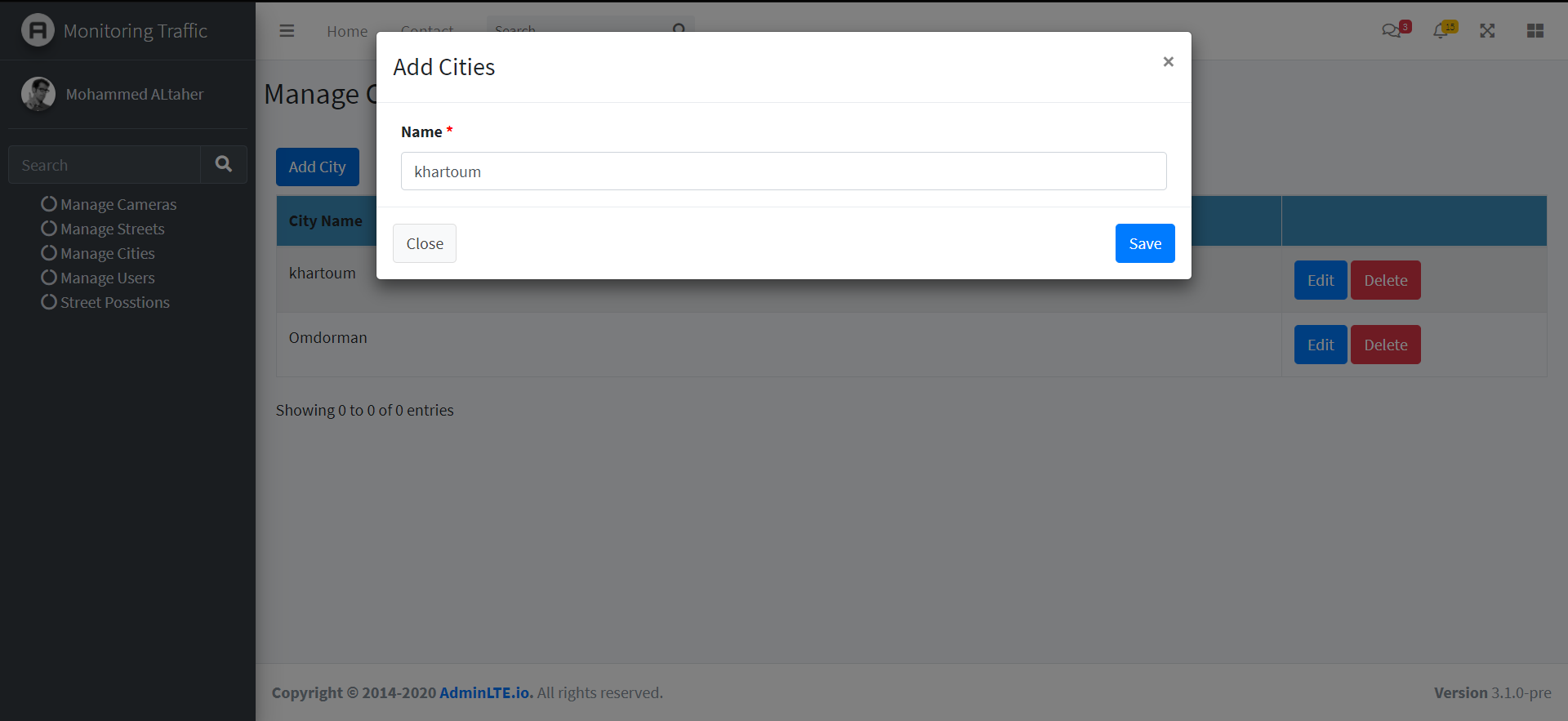


1. إدخال معلومات المدن :
   * + - يتم إدخال اسم المدينة

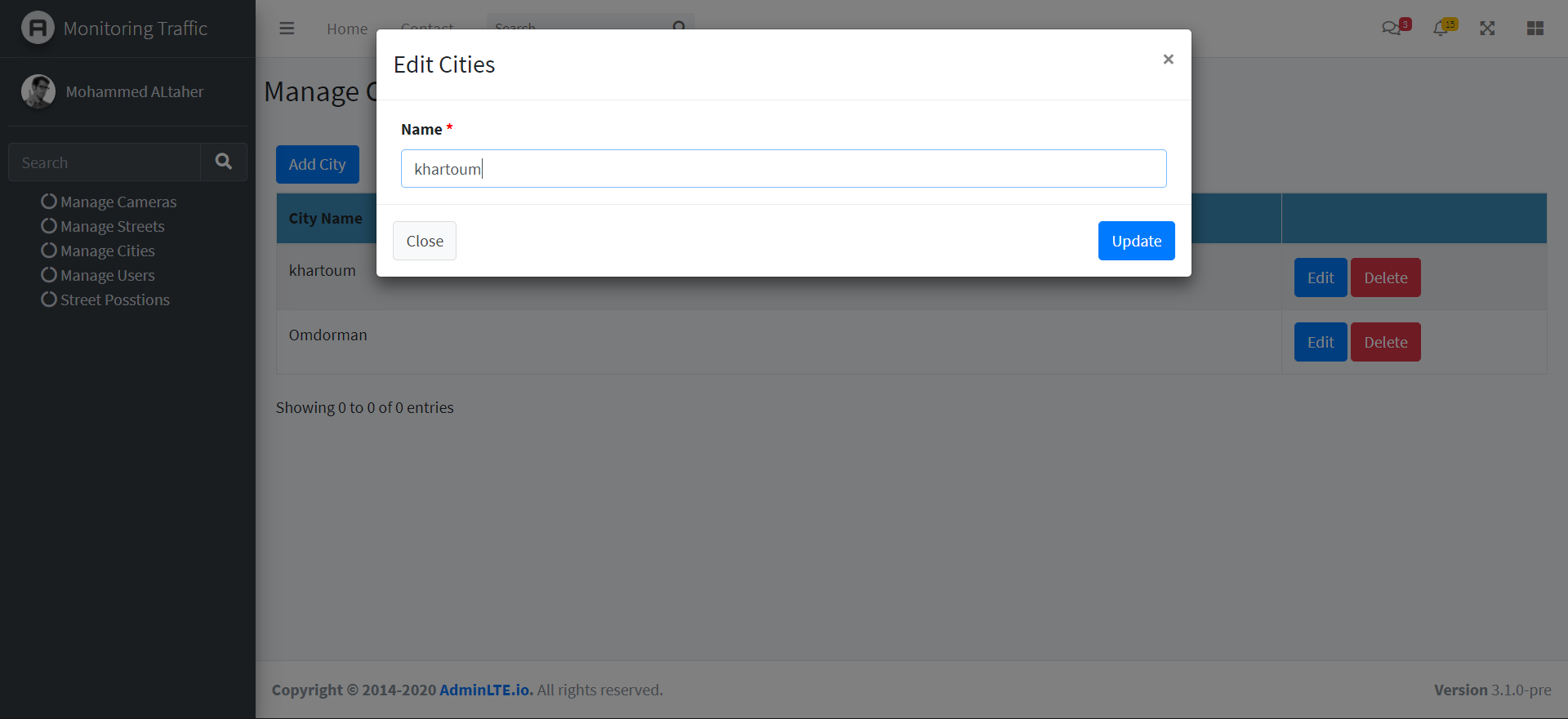
الصورة (4-4) : إدارة المدن



الصورة (4-5) : إدارة المدن - إضافة مدينة

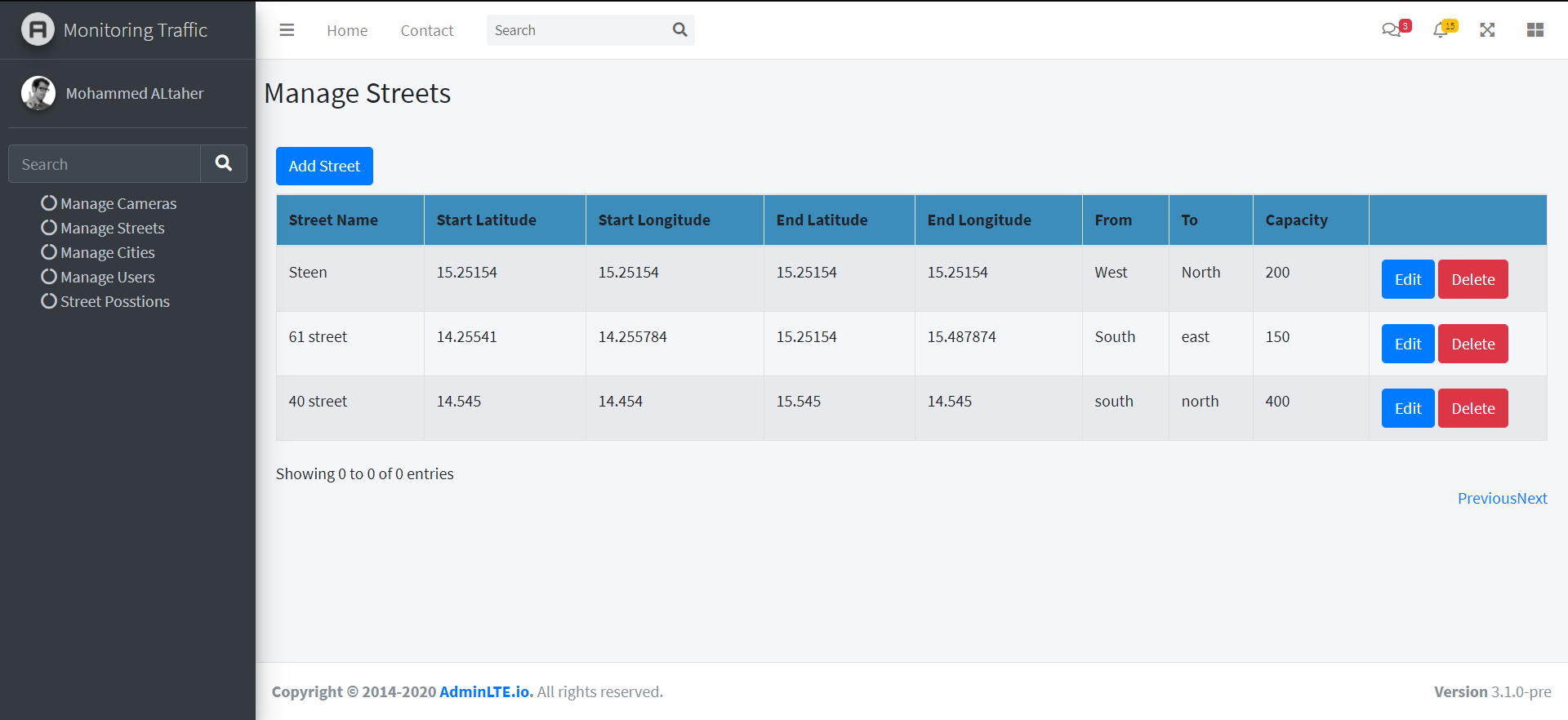


الصورة (4-6) : إدارة المدن – تعديل مدينة

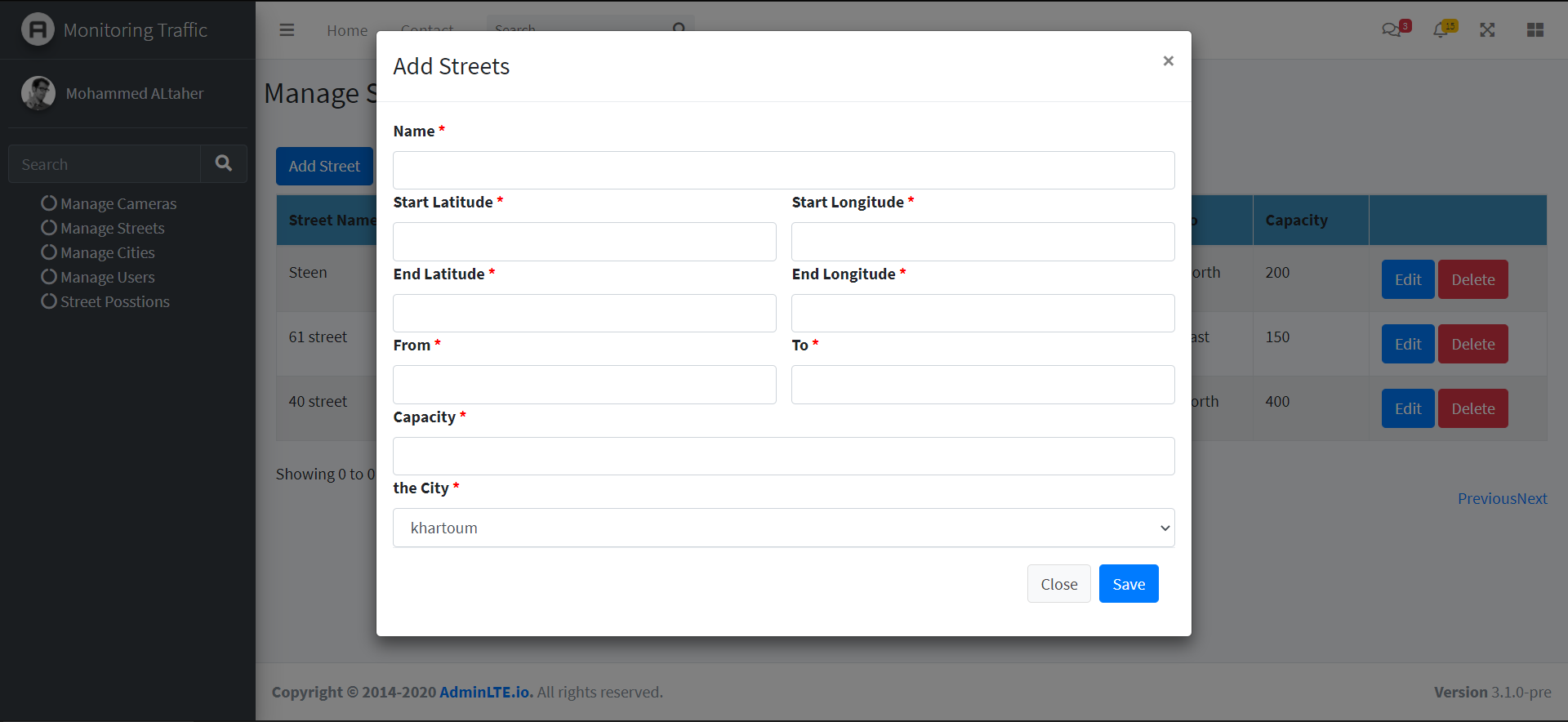


1. إدخال الطرق التي بها كاميرات :
   * + - اسم الطريق
       - بداية خط العرض
       - نهاية خط العرض
       - بداية خط الطول
       - نهاية خط الطول
       - من (اي اتجاه)
       - الى (اي اتجاه)
       - السعة
       - المدينة

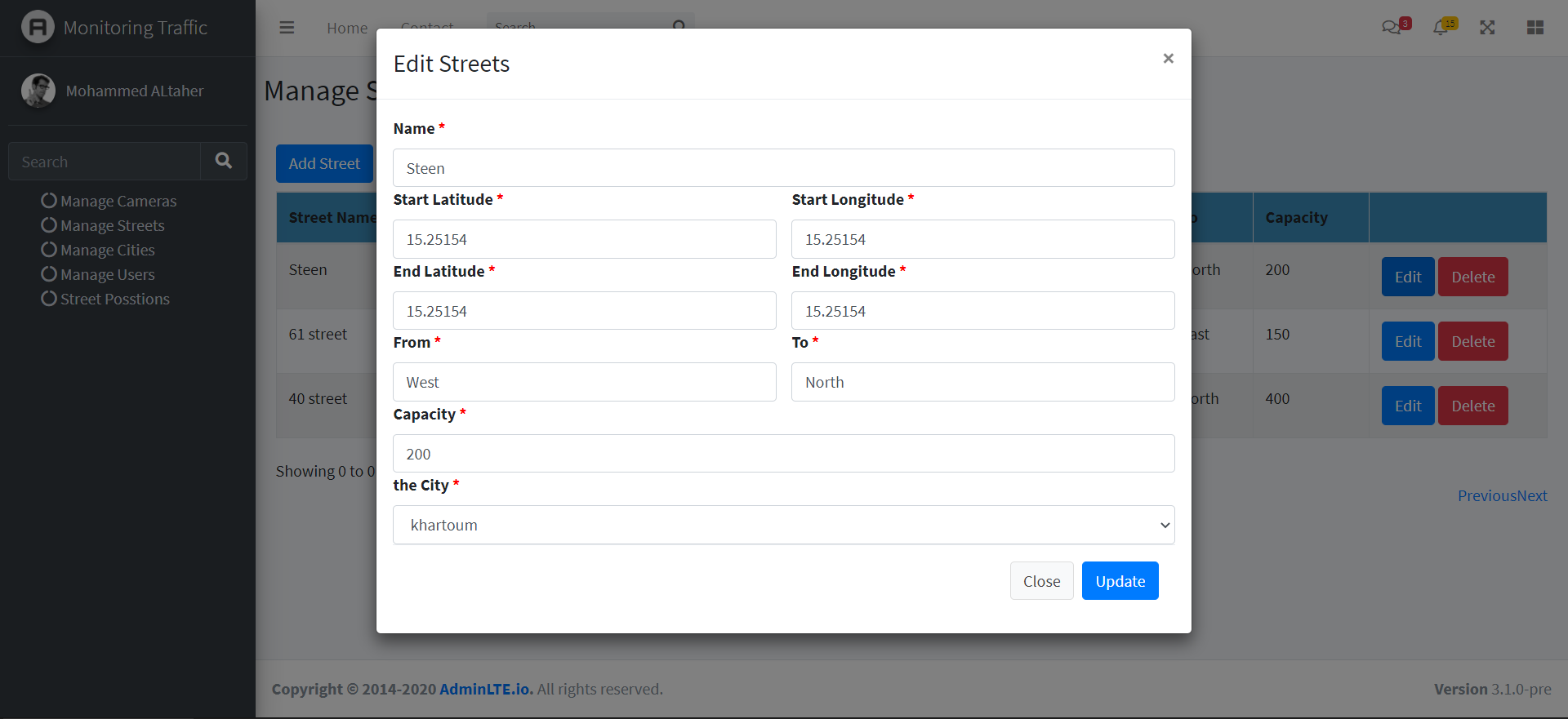
الصورة (4-7) : إدارة الطرق



الصورة (4-8) : إدارة الطرق - اضافة طريق

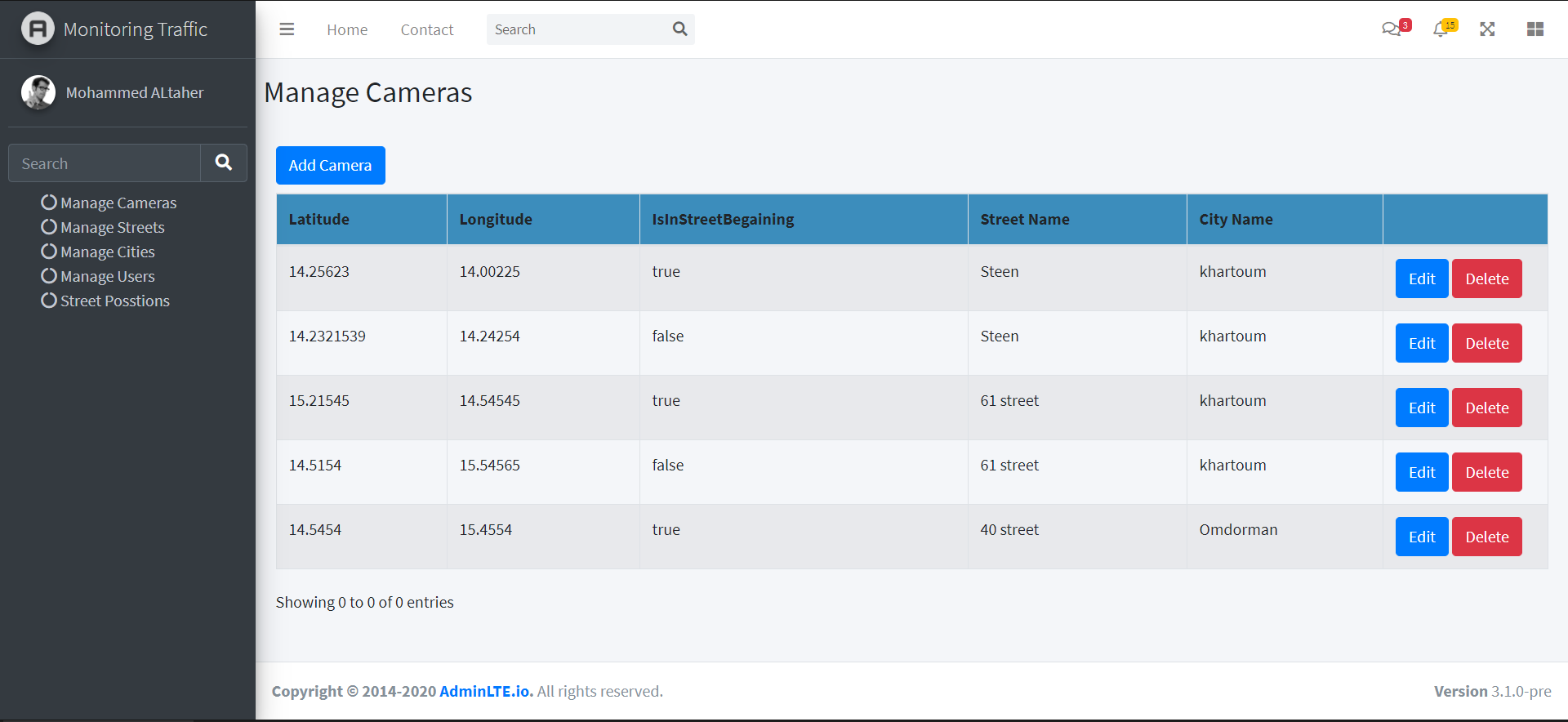


الصورة (4-9) : إدارة الطرق - تعديل طريق

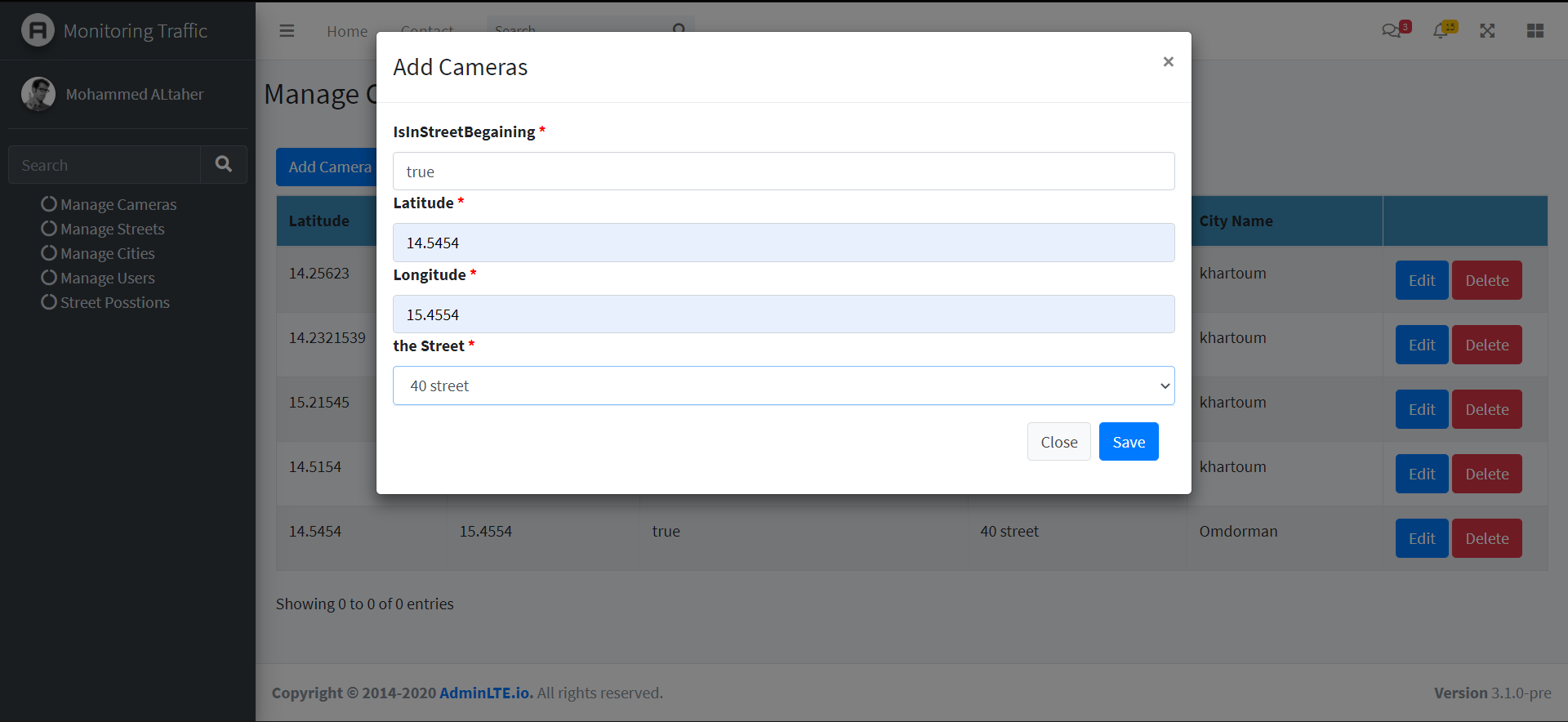


1. معلومات الكاميرات التي تم تركيبها :
   * + - هل الكاميرا في بداية الطريق (قيمة ثنائية)
       - خط العرض
       - خط الطول
       - اسم الطريق

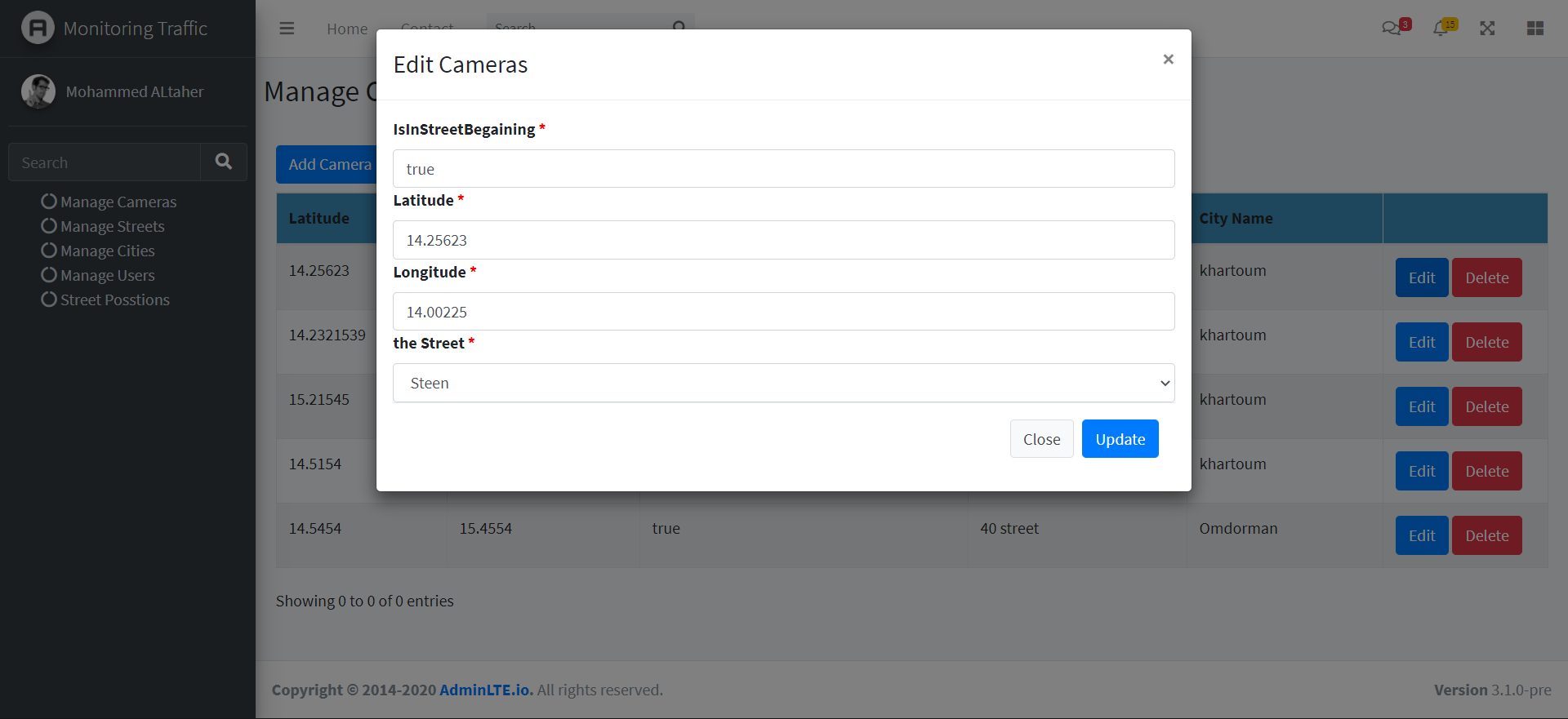
الصورة (4-10) : إدارة الكاميرات



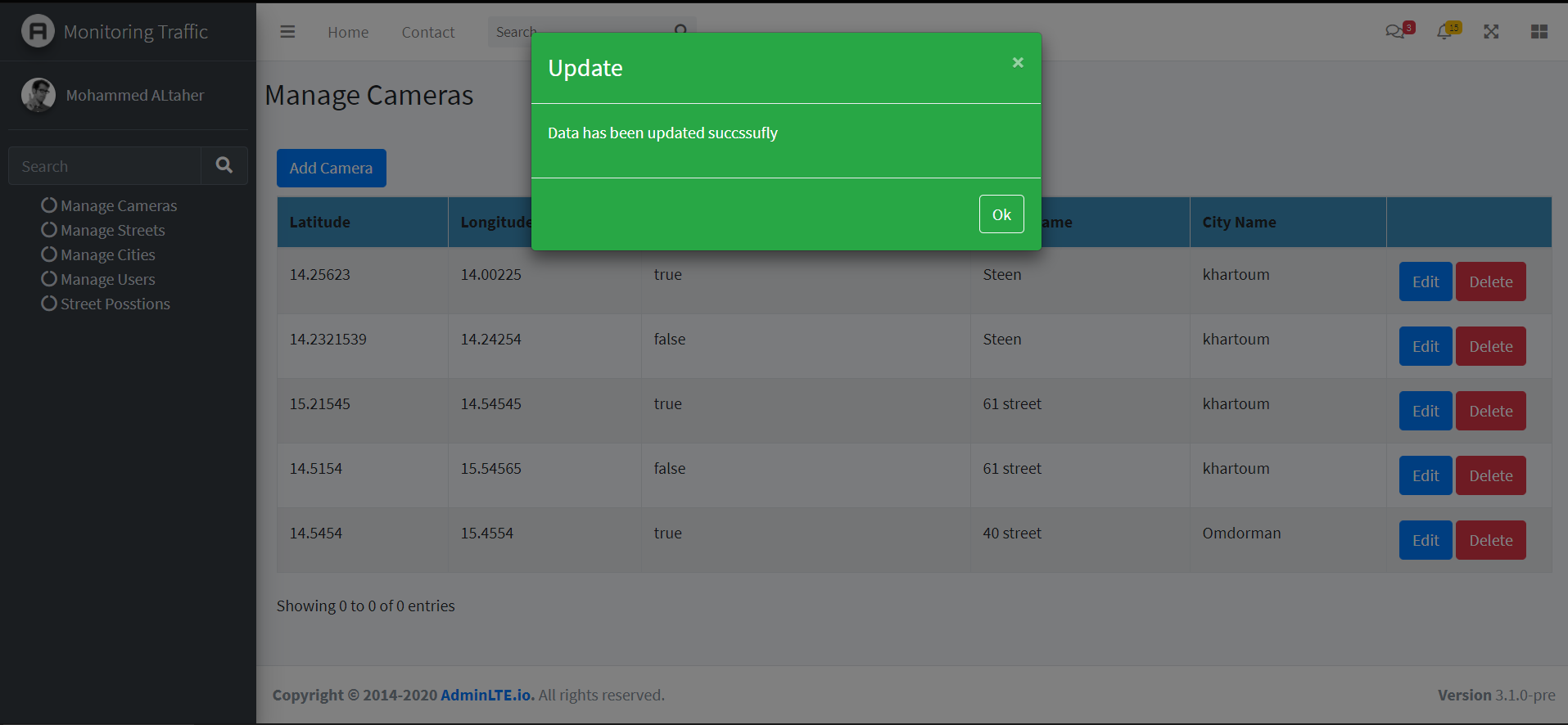
الصورة (4-11) : إدارة الكاميرات – إضافة كاميرا



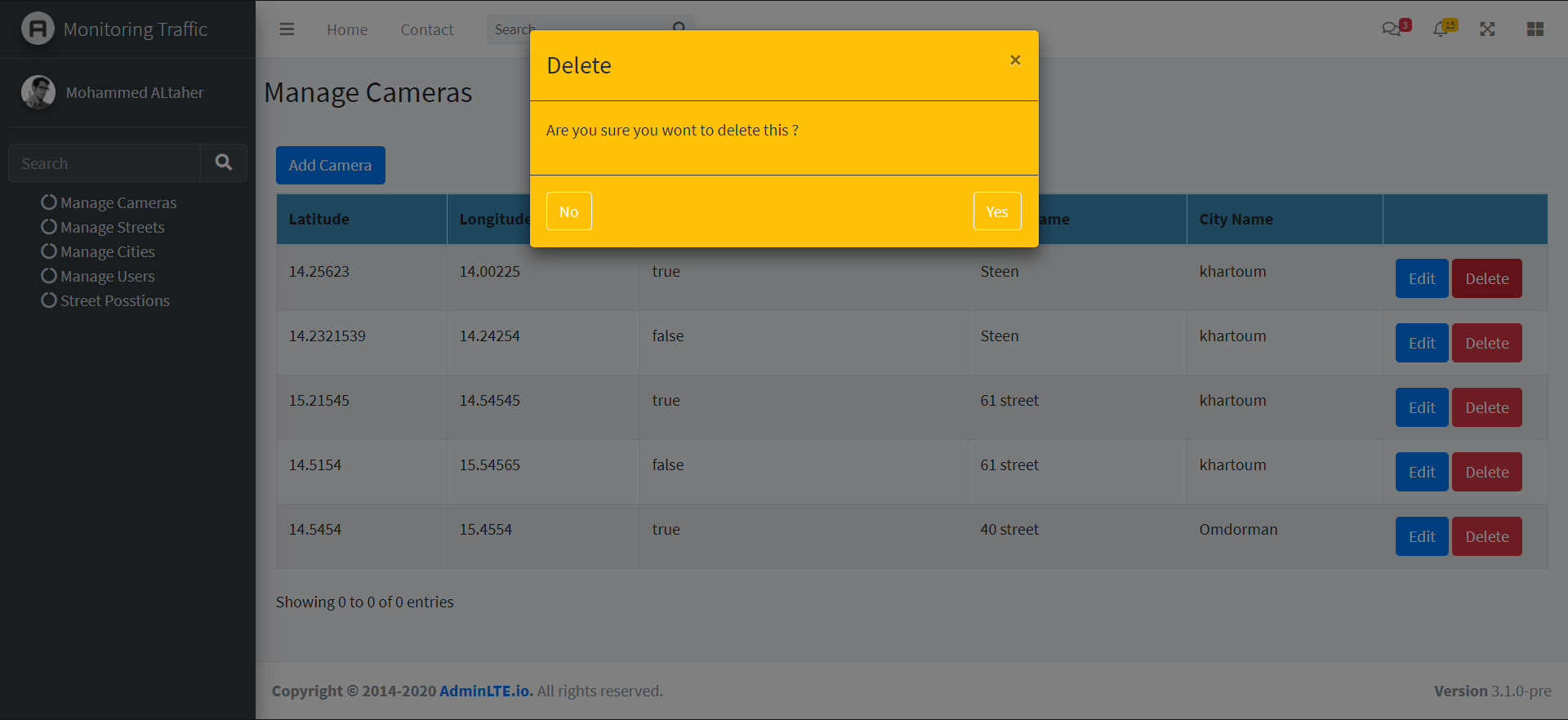
الصورة (4-12) : إدارة الكاميرات – تعديل كاميرا



الصورة (4-13) : إدارة الكاميرات – رسالة تعديل الكاميرا بنجاح

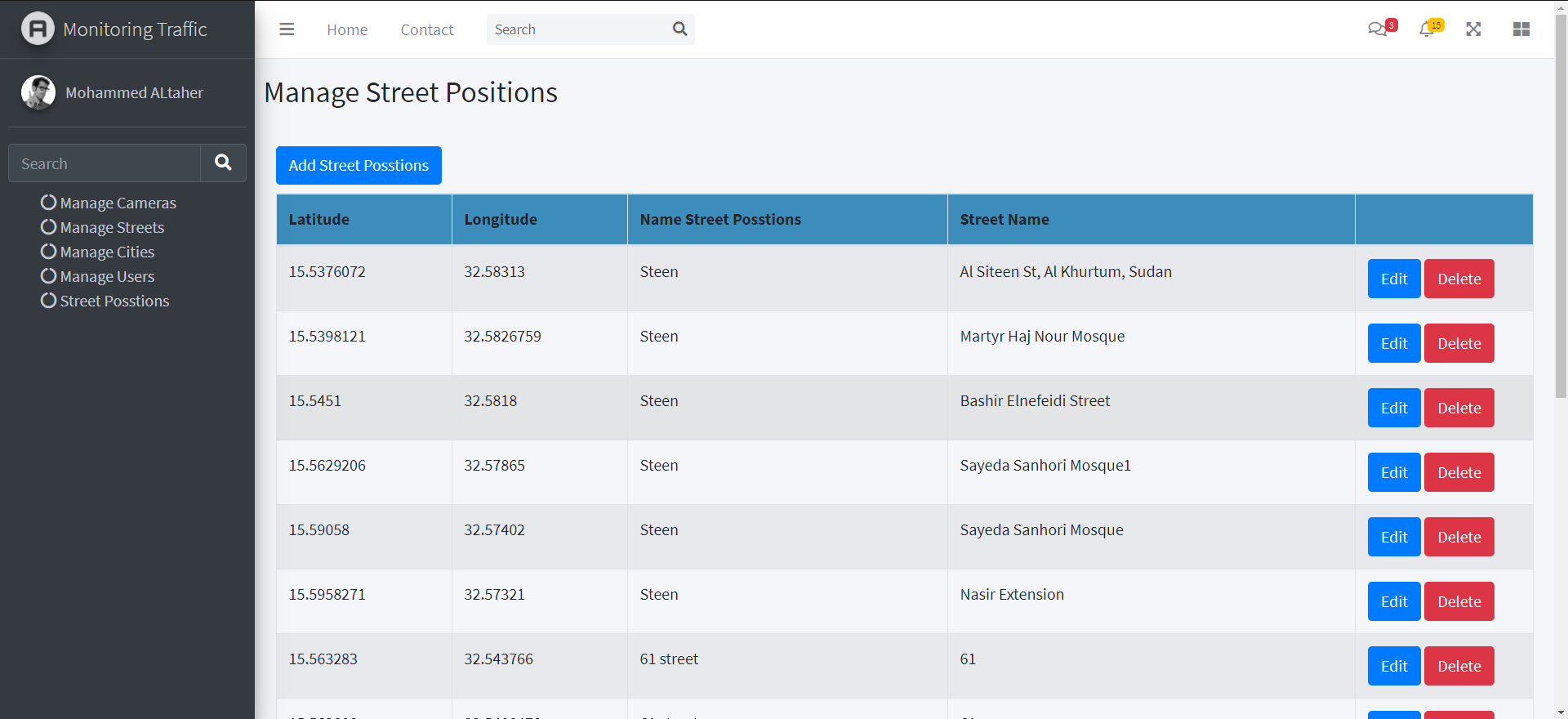


الصورة (4-14) : إدارة الكاميرات – رسالة تحذير حذف الكاميرا

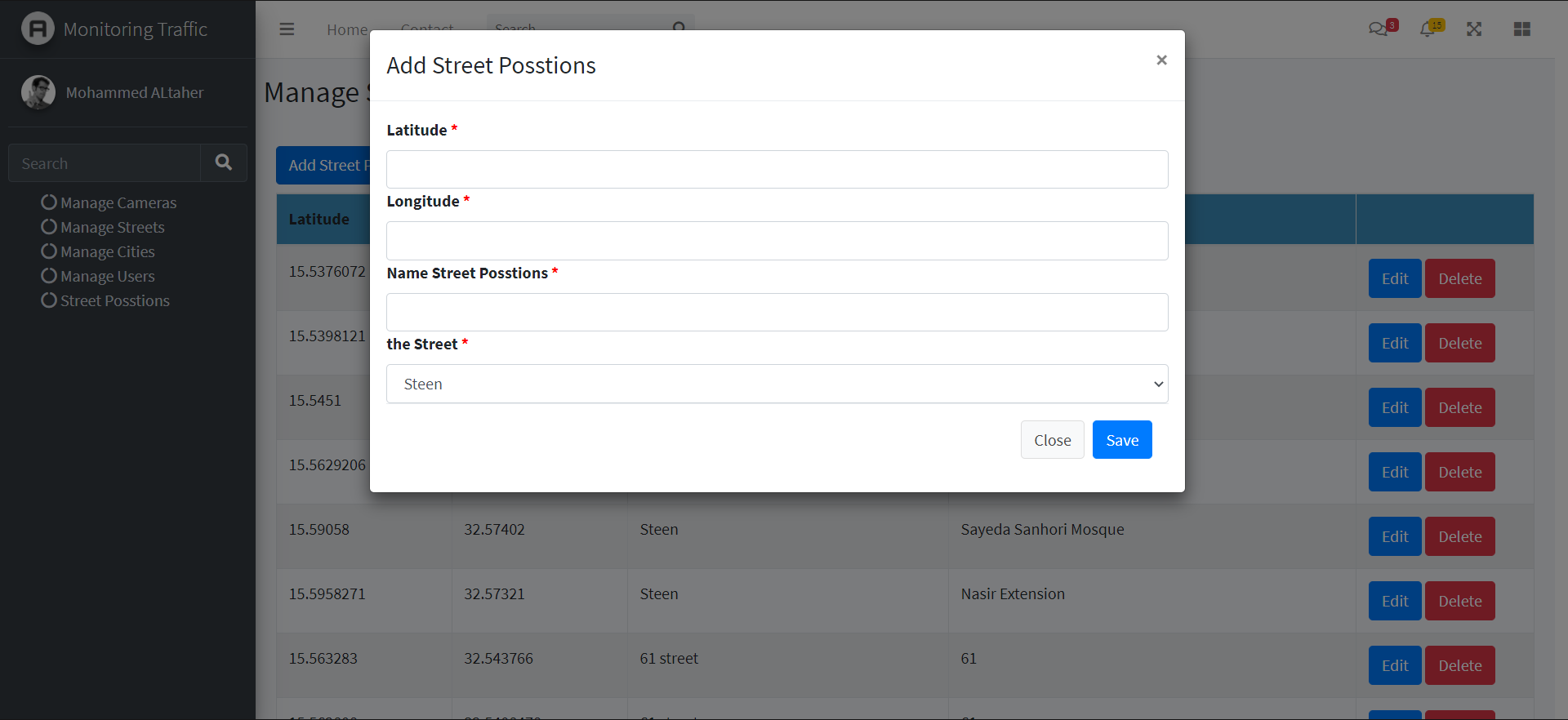


1. إدخال مواقع على الطرقات لاستخدامها لتمثيل الإزدحام على الطرق :
   * + - خط الطول
       - خط العرض
       - اسم الموقع
       - اسم الطريق

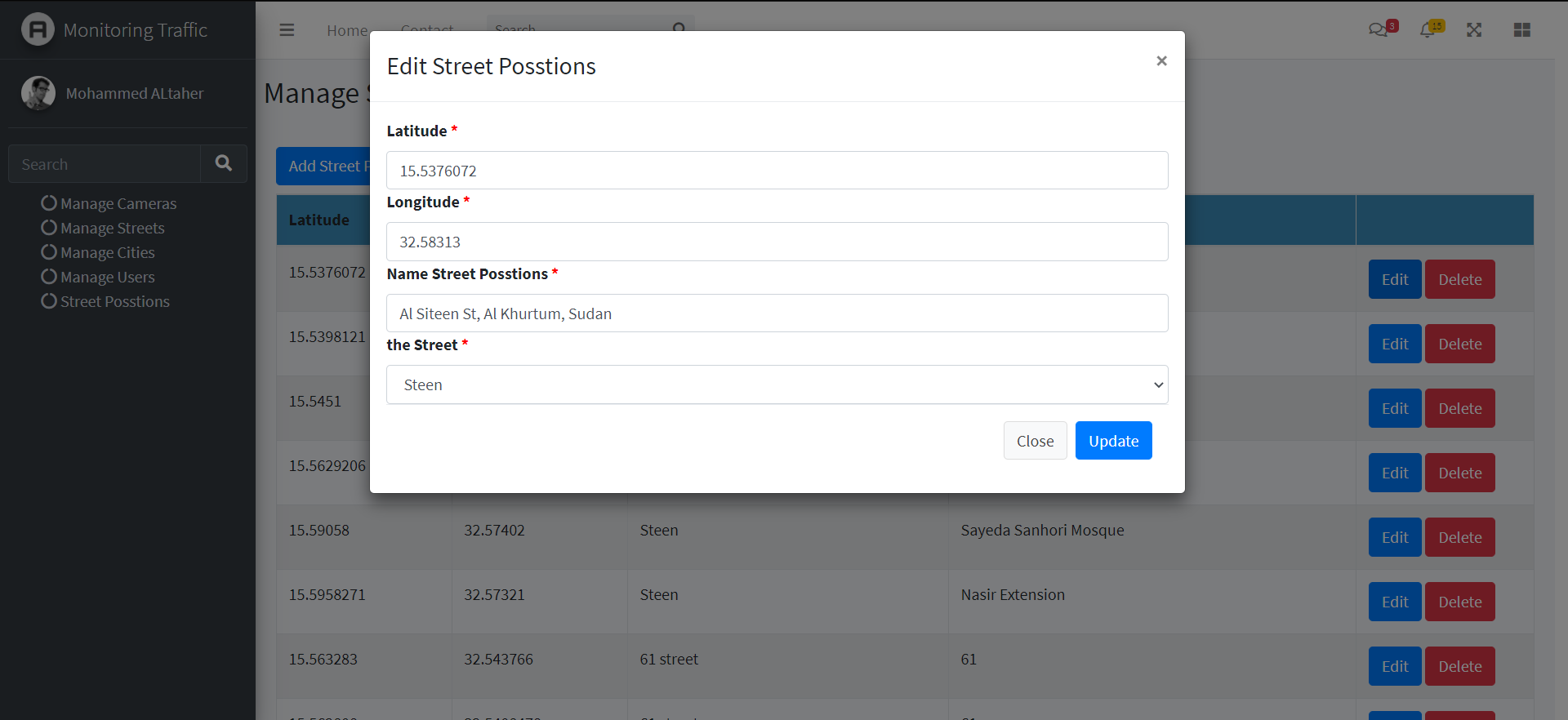
الصورة (4-15) : إدارة مواقع الطرق



الصورة (4-16) : إدارة مواقع الطرق – إضافة موقع

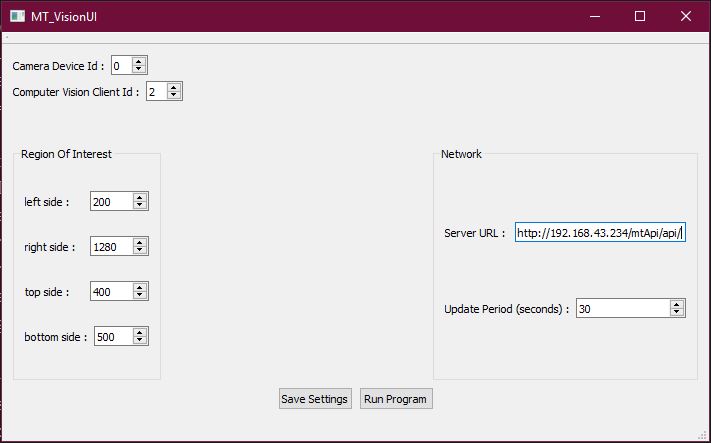


الصورة (4-17) : إدارة مواقع الطرق – تعديل موقع



* + 1. يتم إعداد و تشغيل برنامج الرؤية بالحاسوب :
       - رقم جهاز الكاميرا (camera device id) , في حالة اذا كان الحاسوب متصلا بأكثر من كاميرا واحدة.
       - الرقم التعريفي لموقع الكاميرا (camera client id) , حيث يتم ارسال هذا الرقم للخادم ليتمكن من معرفة مصدر المعلومات.
       - تحديد أبعاد منطقة الأهمية (roi).
       - فترة التحديث, الفترة الدورية لارسال معلومات خوارزمية التعرف و تحديث الخادم.
       - موقع الخادم (server url).

الصورة (4-18) : برنامج الرؤية – شاشة الإعدادات

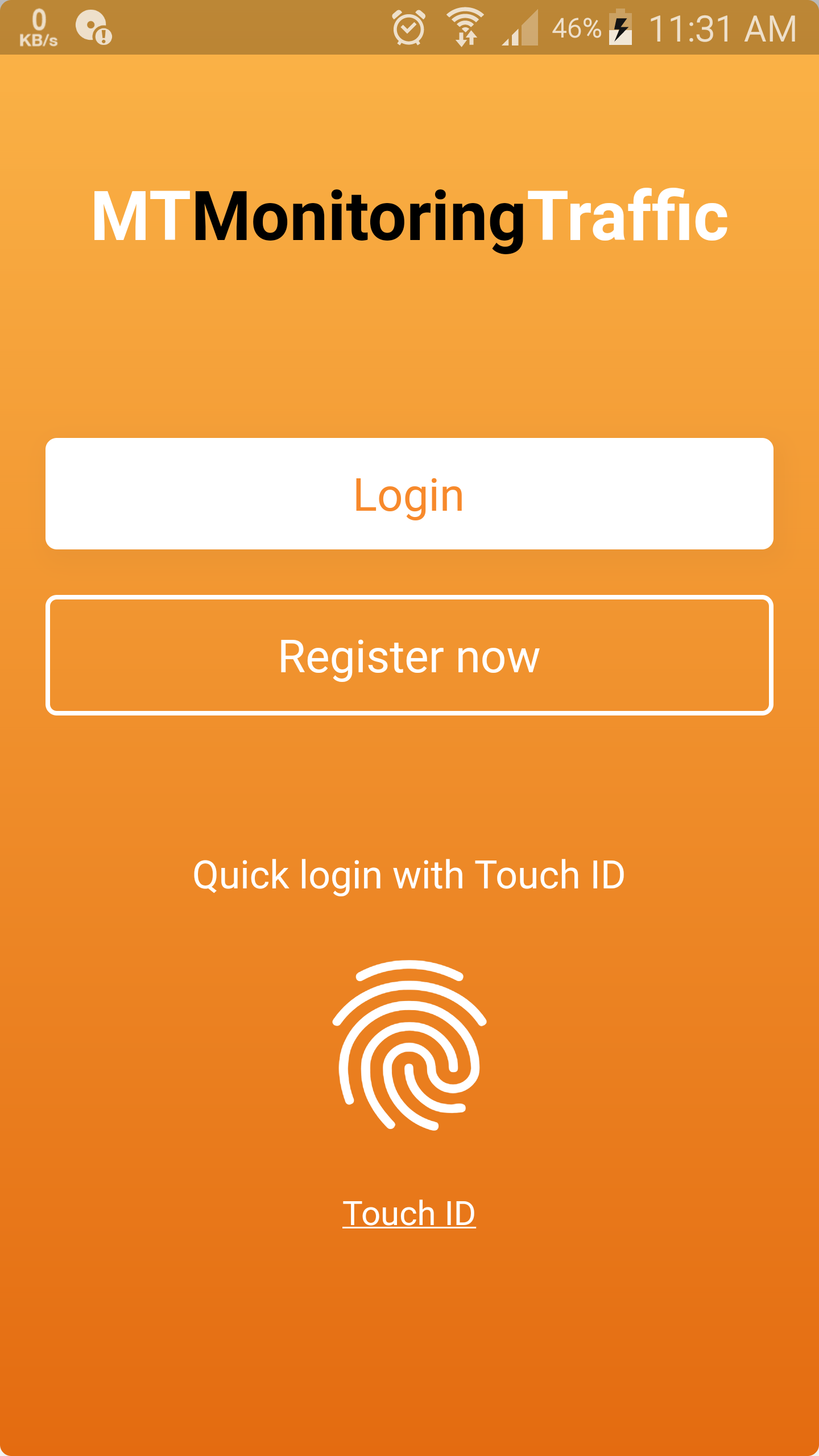


الصورة (4-19) : برنامج الرؤية – نافذة التنفيذ الرئيسية (معالجة الفيديو)



* + 1. و أخيراَ يقوم المستخدم بتشغيل التطبيق على الهاتف :

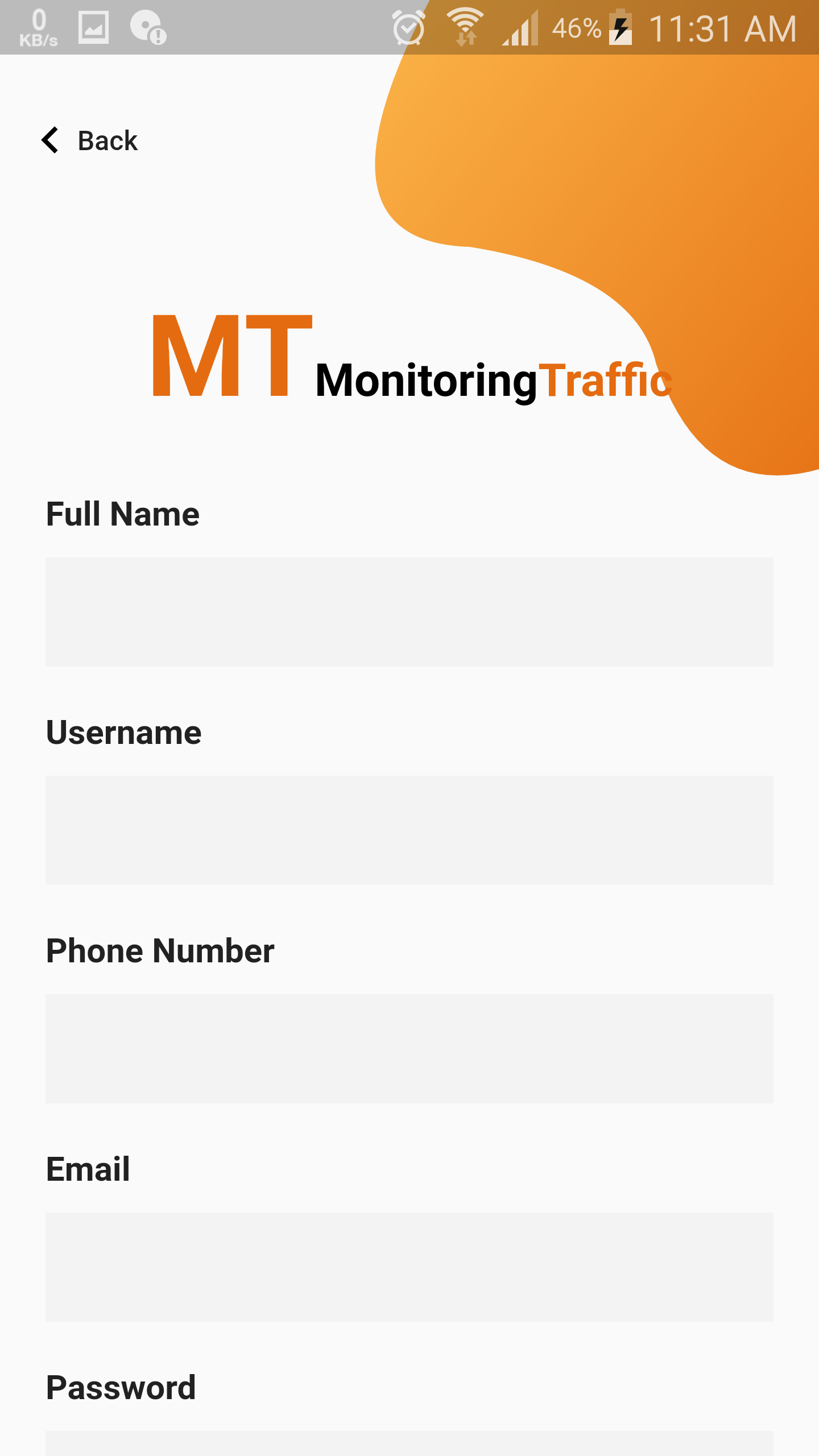
الصورة (4-20) : تطبيق المستخدم – شاشة الرئيسية



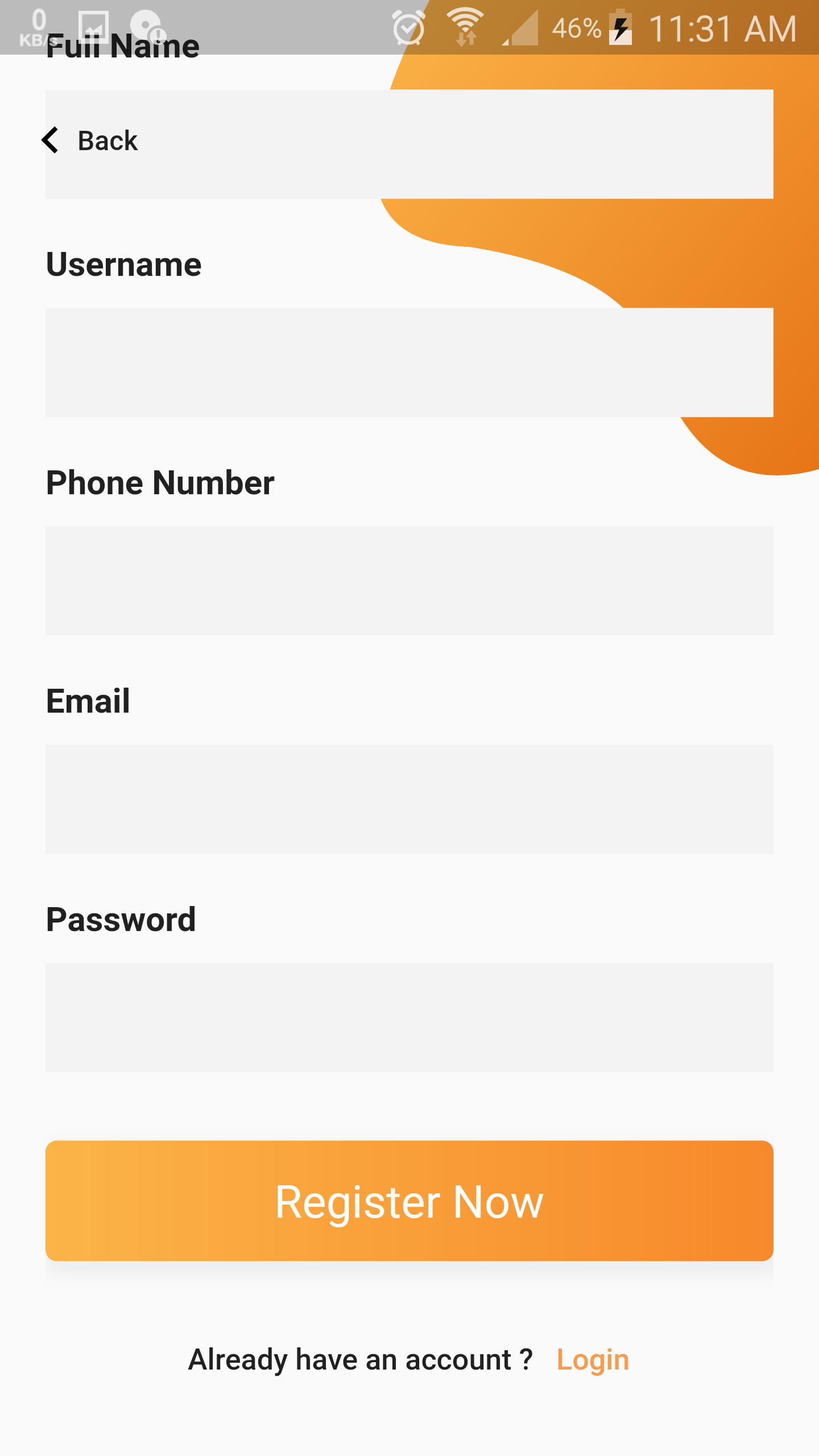
ثم بناءَ على نوع المستخدم تكون العملية التالية مختلفة :

* + - إذا كان المستخدم مستخدماَ عادياَ إذا يقوم بتسجيل حساب جديد :

الصورة (4-21) : تطبيق المستخدم – تسجيل حساب جديد (الجزء الأول)

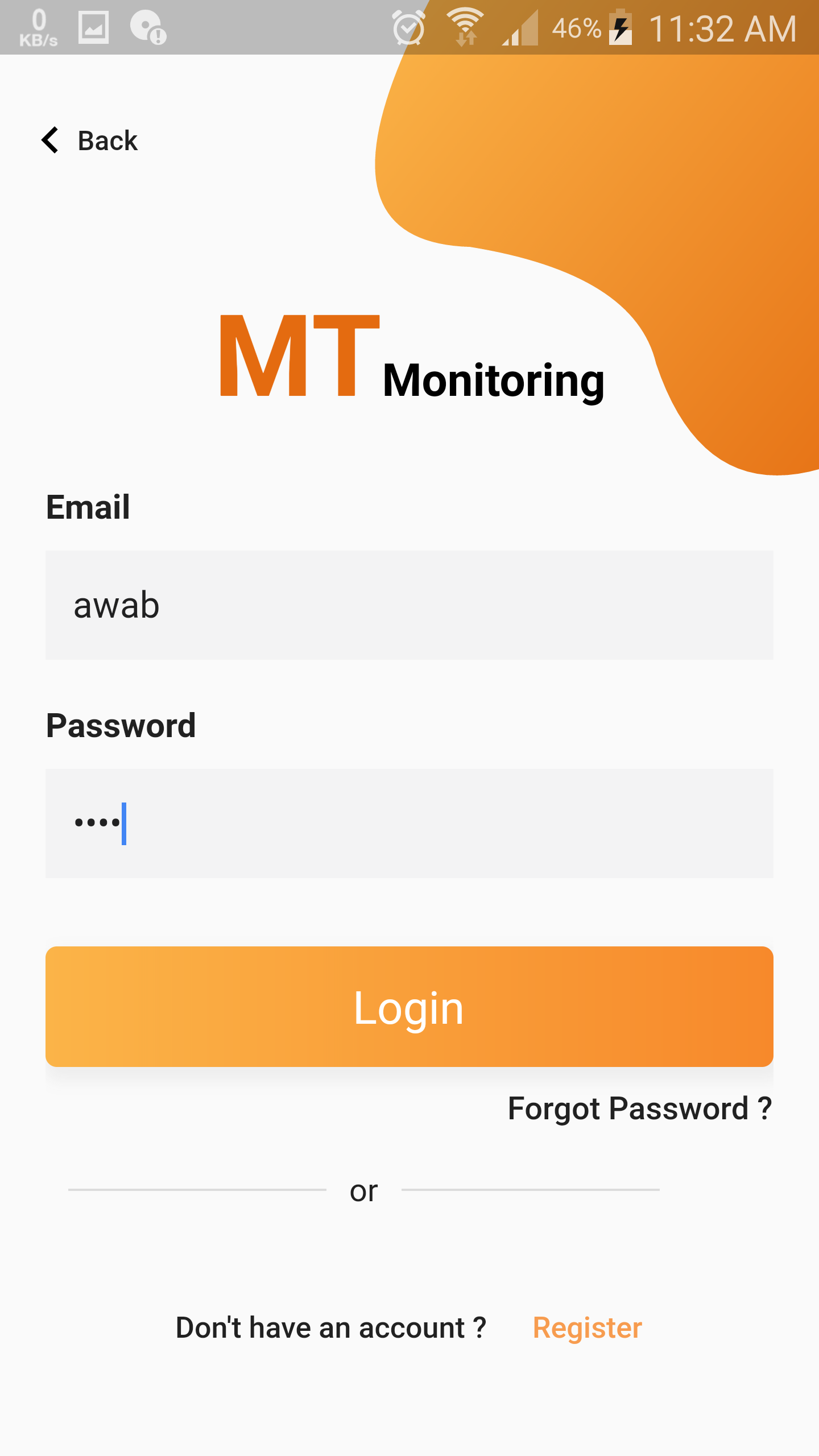


الصورة (4-22) : تطبيق المستخدم – تسجيل حساب جديد (الجزء الثاني)



* + - إذا كان مستخدما تابعاَ لإدارة المرور أو هيئة الطرق فإنه يقوم بتسجيل الدخول مباشرة :

الصورة (4-23) : تطبيق المستخدم – تسجيل الدخول

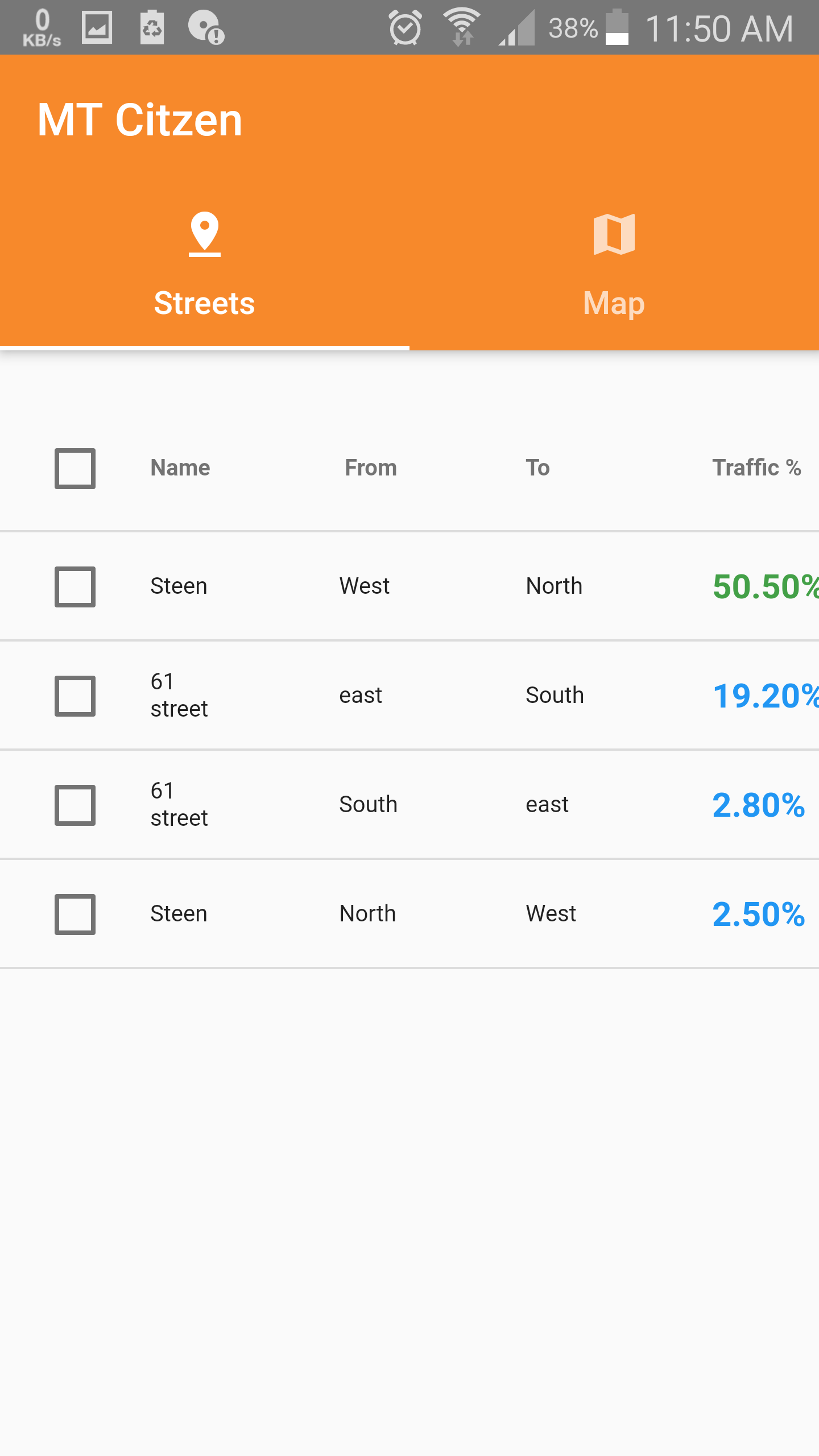


الصورة (4-24) : تطبيق المستخدم – رسالة خطأ في الإدخال



* الصلاحيات المشتركة لكل المستخدمين :
  + عرض لائحة معلومات الطرق.

الصورة (4-25) : تطبيق المستخدم – لائحة الطرق



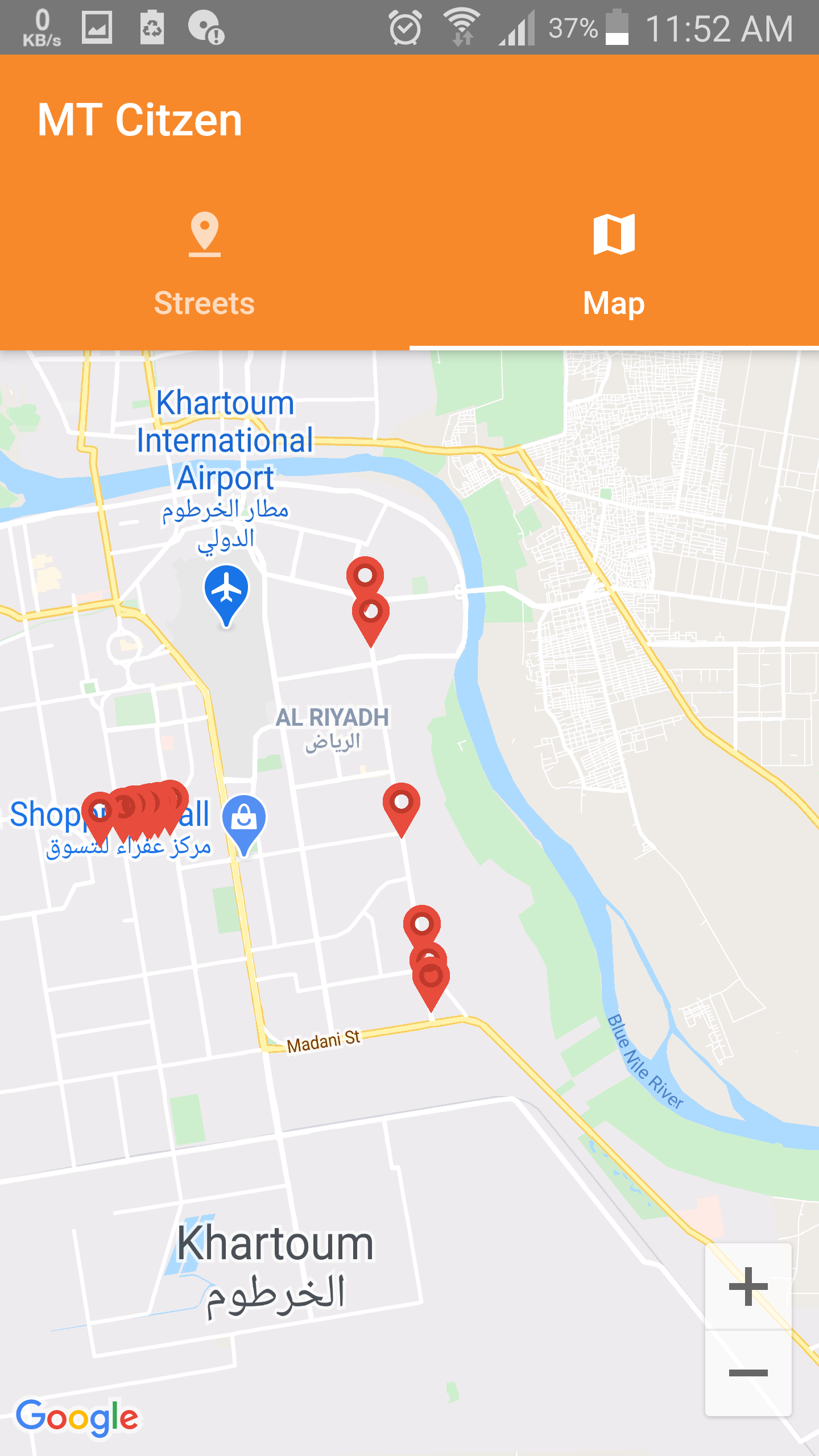
* + عرض الخريطة التي تظهر طريقا محدداَ.

الصورة (4-26) : تطبيق المستخدم – خريطة لإعرض تمثيل طريق واحد



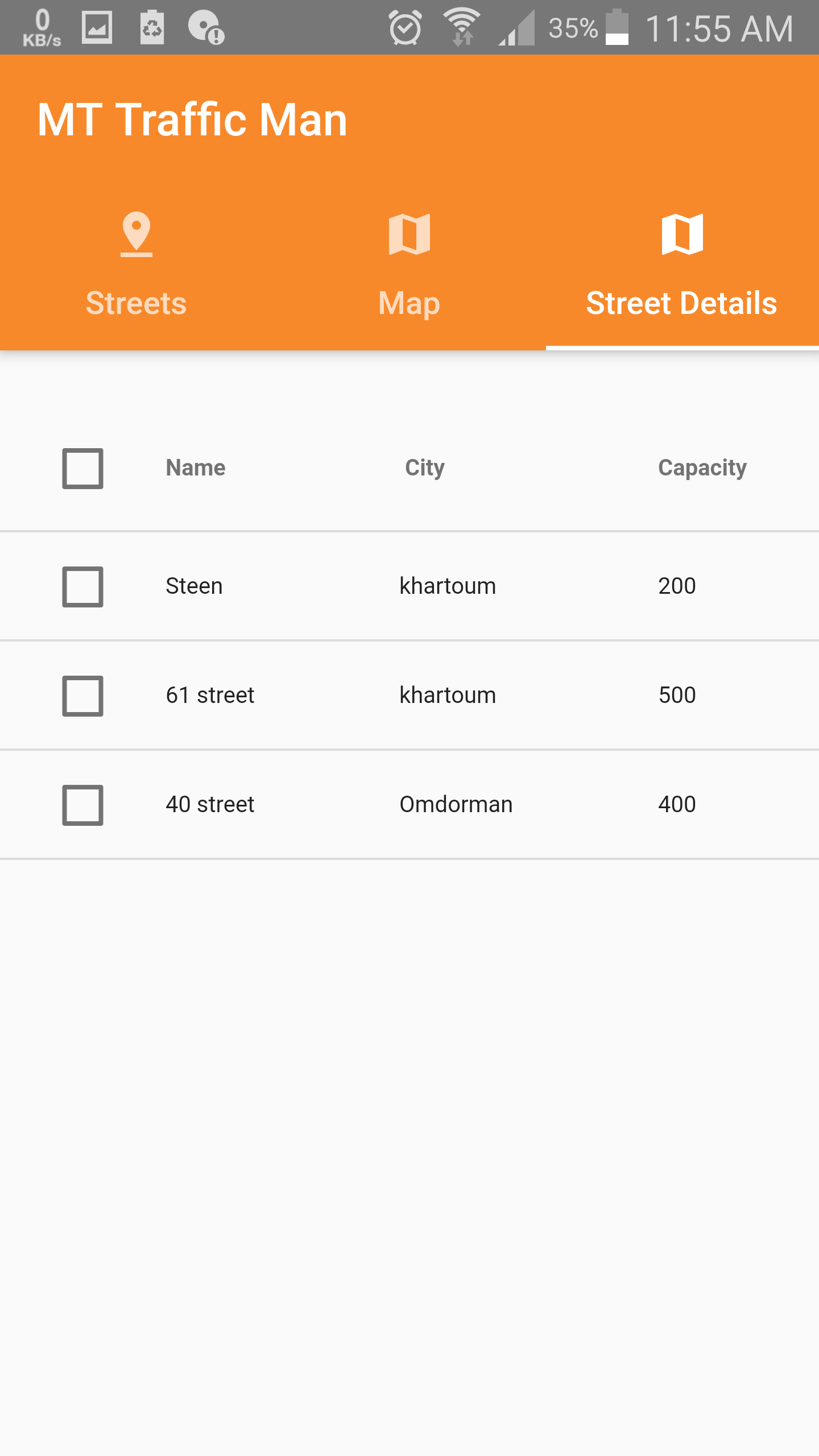
* + عرض خريطة الكاملة لكل الطرق المسجلة في النظام.

الصورة (4-27) : تطبيق المستخدم – الخريطة الكاملة



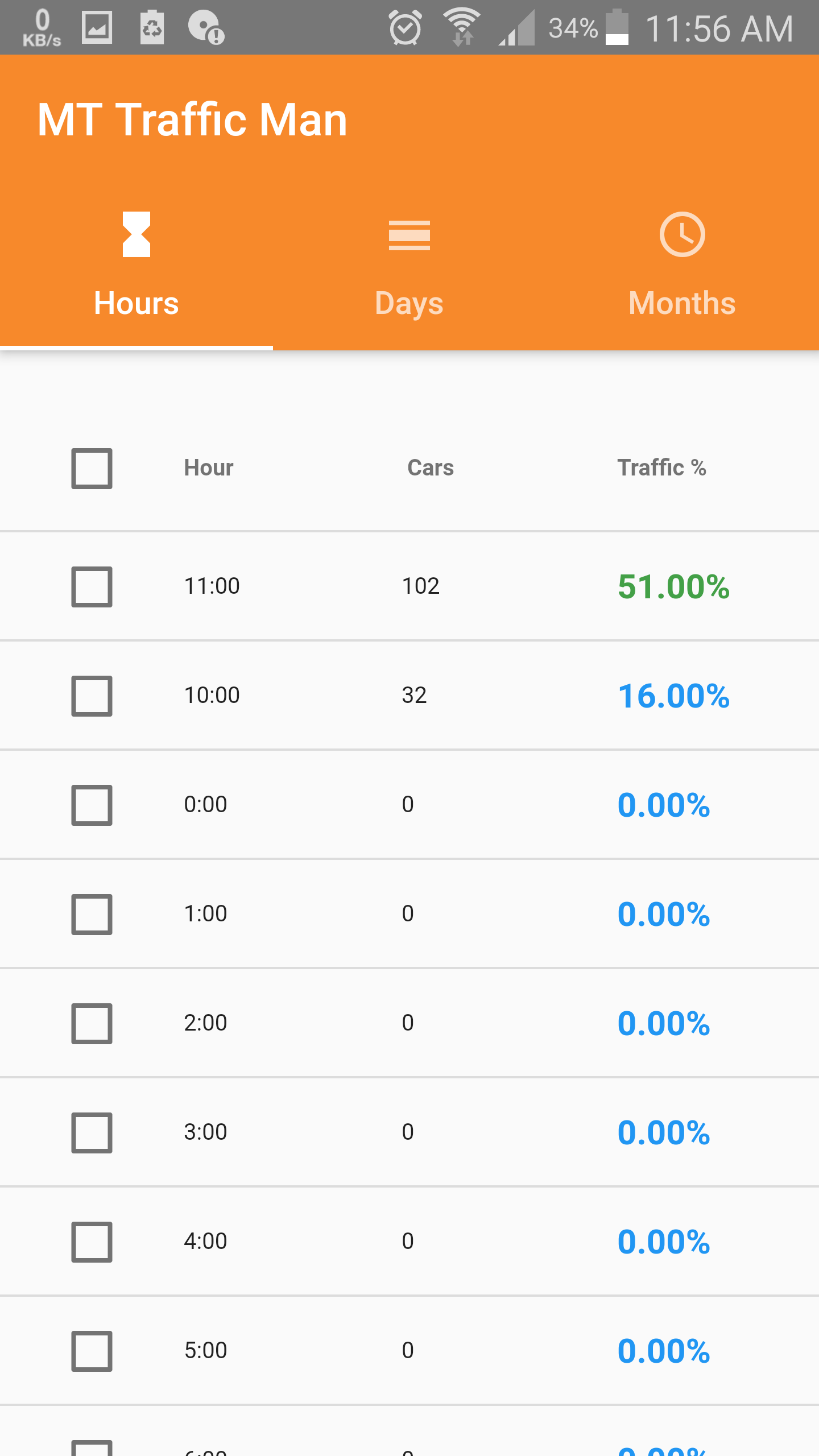
* صلاحيات إدارة و شرطة المرور :
  + لائحة أكثر تفصيلاَ للطرق المسجلة تحتوي على الأرقام الفعلية للمركبات على الطريق.

الصورة (4-28) : تطبيق المستخدم – تفاصيل الطرق

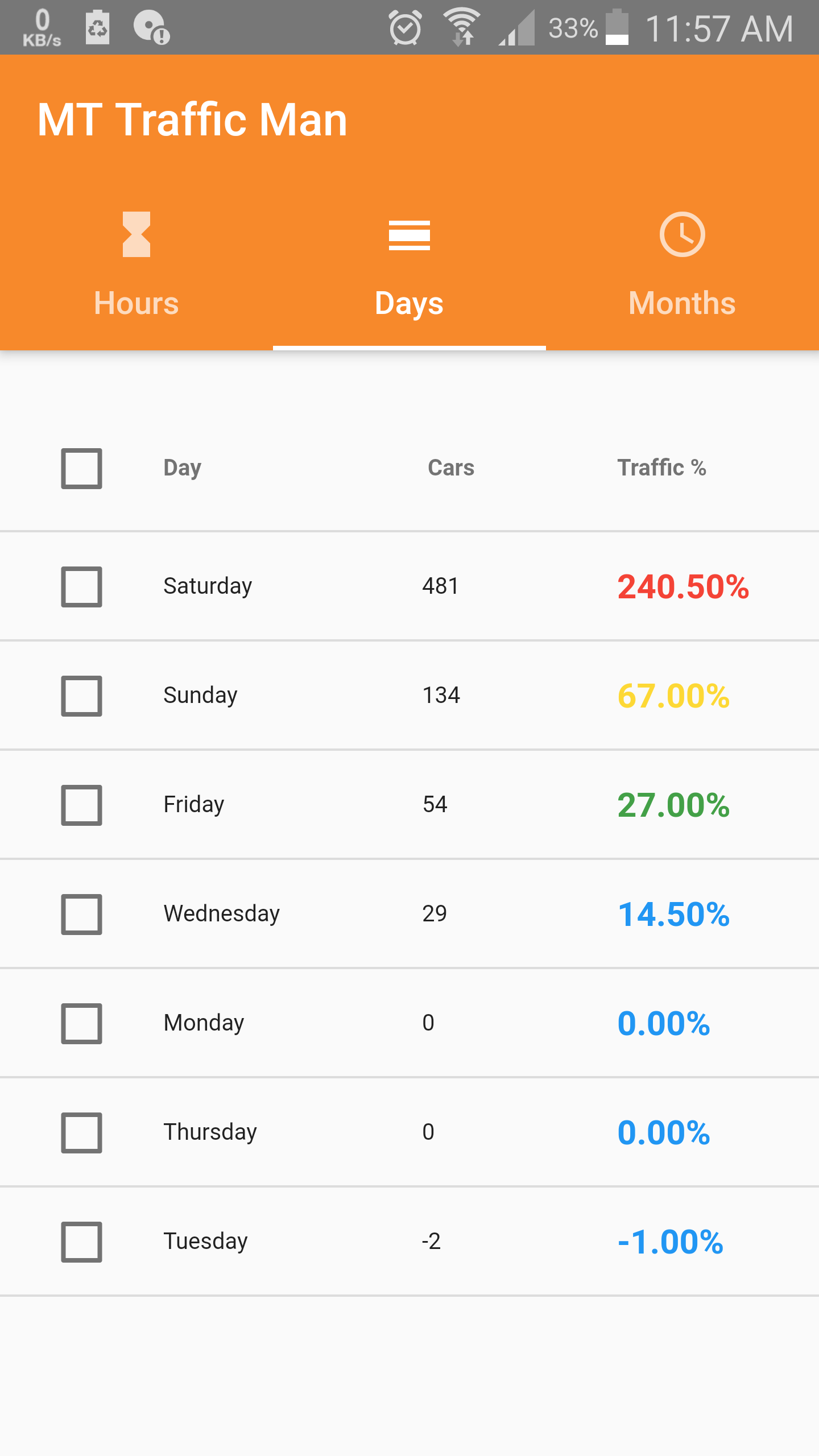


* + تقرير زمني للذروة حسب الساعة أو أيام الأسبوع أو الشهور مرتبة حسب الأكثر إزدحاماَ.

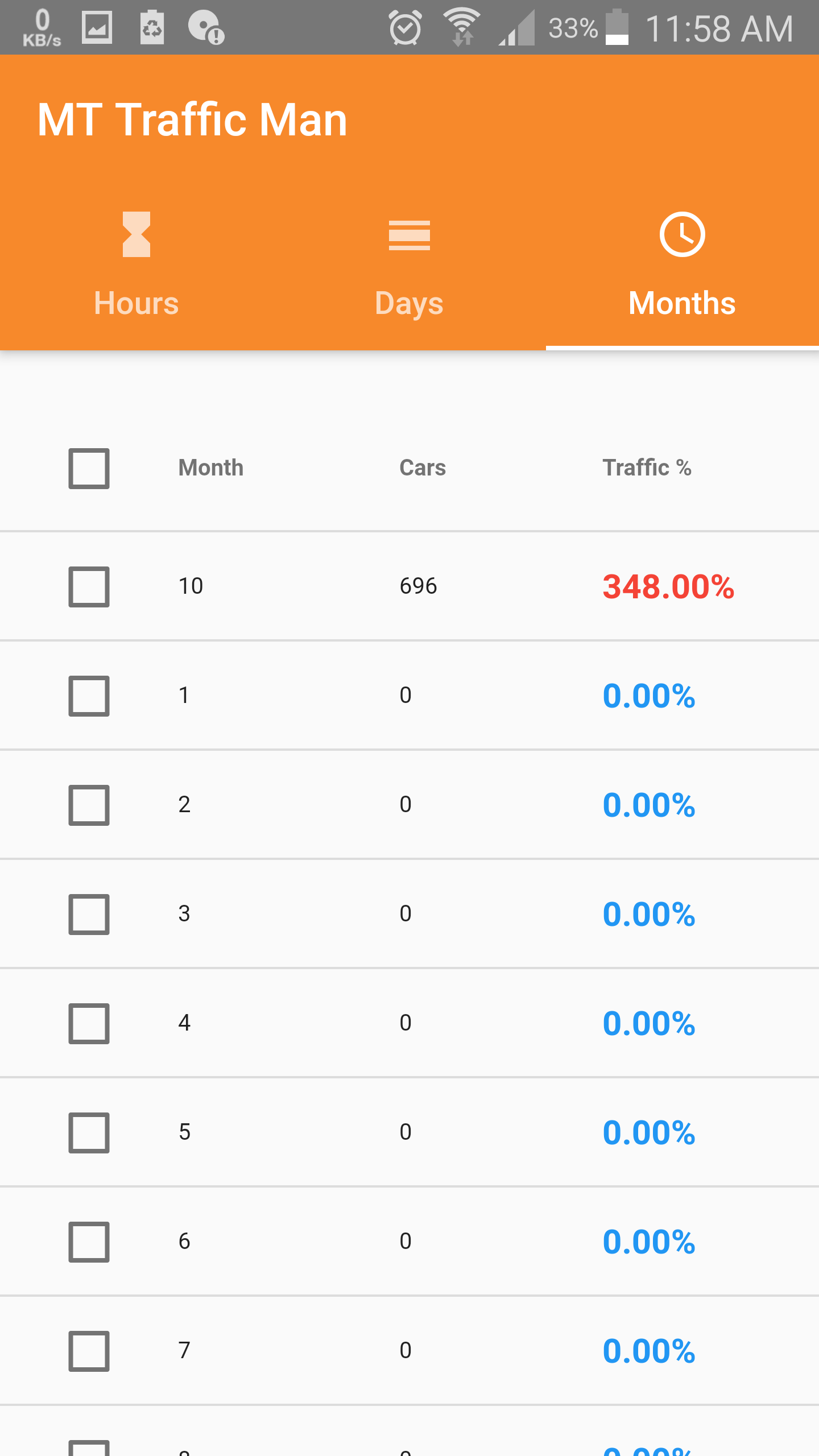
الصورة (4-29) : تطبيق المستخدم – ساعات الذروة بالترتيب لطريق محدد



الصورة (4-30) : تطبيق المستخدم – الأيام الأكثر إزدحاماَ

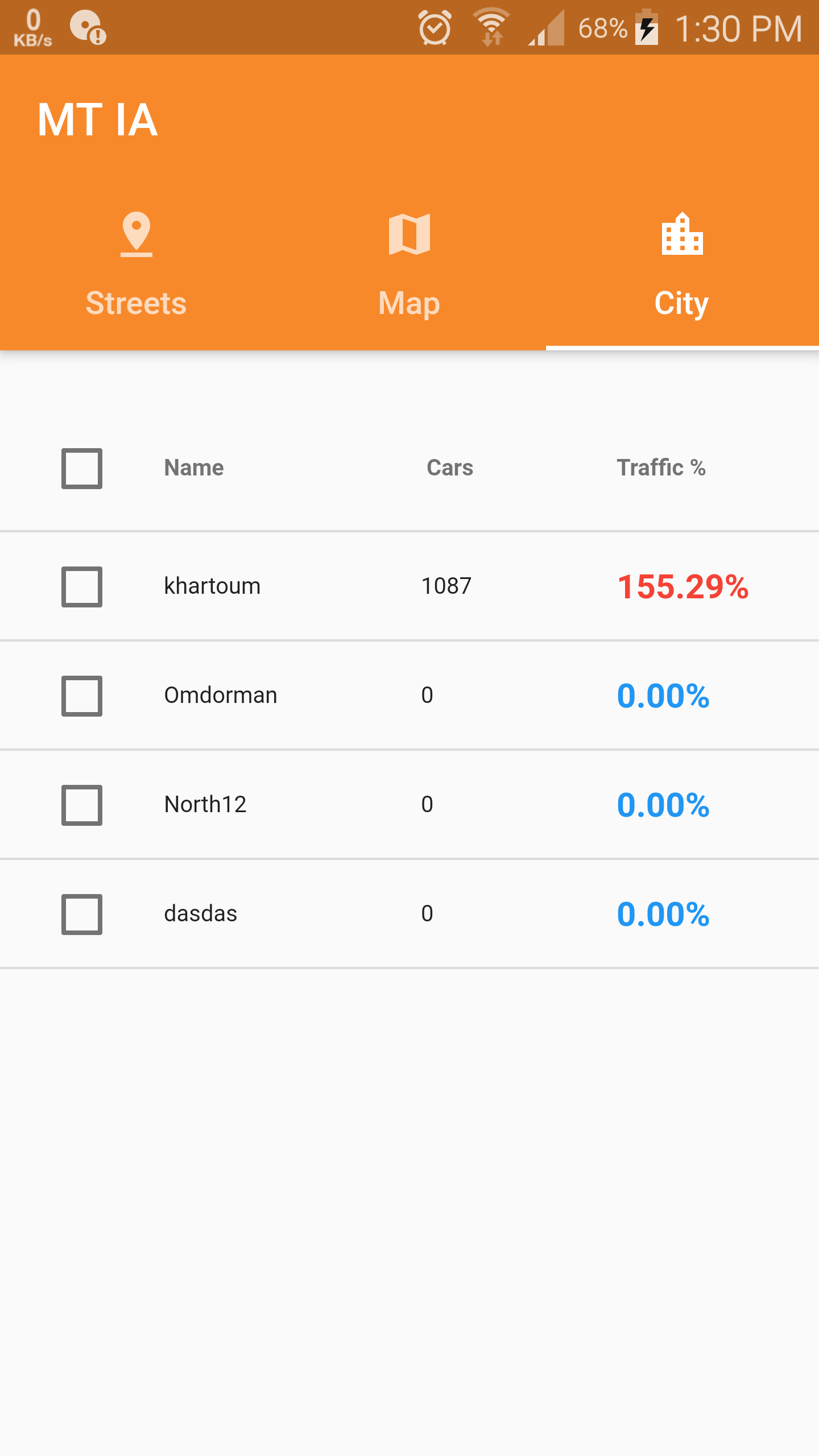


الصورة (4-31) : تطبيق المستخدم – ترتيب الشهور الأكثر إزدحاماَ



* صلاحيات هيئة الطرق و الجسور :
  + عرض معلومات المدن مصنفة حسب الأكثر إزدحاماَ.

الصورة (4-32) : تطبيق المستخدم – ترتيب المدن الأكثر إزدحاماَ



**4-2 التقييم :**

توصلنا الى خوارزمية رؤية حاسوبية ذات كفاءة عالية نسبياَ في الظروف الجوية الطبيعية حيث كانت نسبة الخطأ تساوي 12%. و كان الأداء كافيا لتحديد المناطق و الطرق المزدحمة بما يحقق أهداف المشروع.

* + 1. **كيفية حساب الخطأ :**

تعمل الخوارزمية على اكتشاف و تتبع المركبات في الفيديو ثم حساب المركبات المارة في كلا اتجاهي الحركة, و بالتالي تعتمد دقة و كفاءة الخوارزمية على مقدرتها على عد المركبات بدون أخطاء زيادة أو نقصان.

**4-2-2 نتائج الإختبار :**

تم اختبار الخوارزمية باستخدام تسجيلات فيديو مسبقة لحركة المرور على شارع محدد.

جدول (4-1) يوضح نتائج اختبار الخوارزمية:

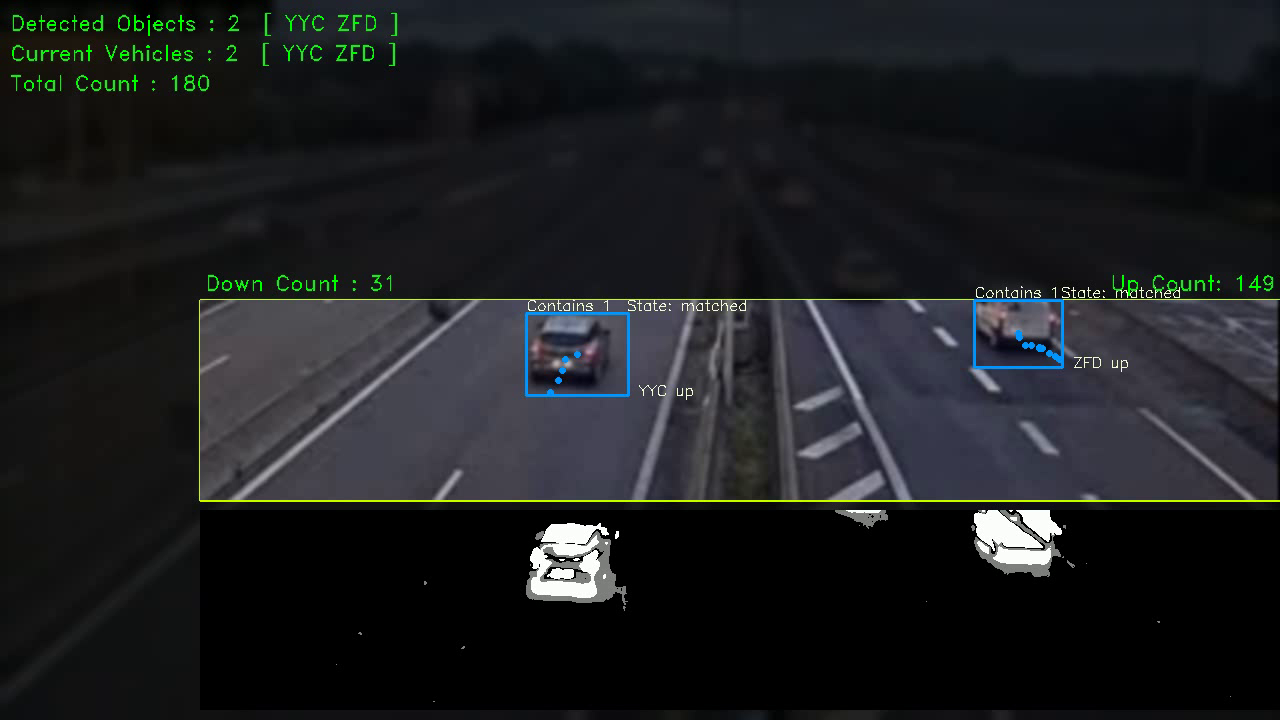
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| رقم الإختبار | طول الفيديو | جودة الفيديو | نتيجة الجانب الأيمن | نتيجة الجانب الأيسر | النتيجة الحقيقية (يمين) | النتيجة الحقيقية (شمال) | النتيجة النهائية |
| 1 | 53 ثانية | جيدة | 82 | 68 | 80 | 52 | 132 من أصل 150  False negatives : 18  False positives : 0 |
| 2 | 3 دقائق و 30 ثانية | سيئة و معتمة (بداية الصباح) | 150 | 31 | 70 | 0 | 181 من أصل 70  False negatives : 0  False positives : 111 |

من الملاحظ أن العوامل المؤثرة على دقة الخوارزمية هي:

* جودة الفيديو عموما حيث تؤثر الضوضاء على مقدرة الخوارزمية على اكتشاف المركبة.
* الإضاءة سواء كانت من الشمس أو المركبة نفسها.
* زاوية تركيب الكاميرا : من الأفضل أن تكون الكامير في منتصف الطريق بحيث لا تغطي مركبة رؤية مركبة أخرى (bird-eye view). و ظهر ذلك في الإختبار الأول.

صورة (4-33) لنتيجة الإختبار الأول :

صورة (4-34) لنتيجة الإختبار الثاتي :



**5-1 النتائج :**

توصلنا من خلال هذا البحث للنتائج التالية :

1. تصميم خوارزمية رؤية بالحاسوب ذات كفاءة عالية في إكتشاف و تتبع المركبات.
2. تطوير نظام متكامل لرصد حركة المرور و تخزين معلوماتها في قاعدة بيانات لتصبح مرجعاَ لصناع القرار في المستقبل.
3. تحقيق أهداف المشروع و التي يمكن أن نجملها في إتاحة المعلومات بسرعة و سهولة لكل مستخدمي النظام.

**5-2 التوصيات :**

1. نوصي باستخدام كاميرات مراقبة ذات جودة عالية و مراعاة وضعها بزاوية عمودية في منتصف الطريق بحيث تكون جميع المركبات و اضحة.
2. العمل على تطوير خوارزمية الرؤية و جعلها أقل عرضة للتأثر بالضوضاء و جودة الصورة.
3. و في النهاية نوصي بأن الجانب البشري من النظام هو المسؤول من تقليل مشاكل الإزدحام سواء كان شرطي المرور أو سائقي المركبات فالحاسوب في النهاية هو أداة لتسهيل اتخاذ القرار.

**5-3 الدراسات المستقبلية في هذا المجال :**

1. دراسة تدمج بين الرؤية بالحاسوب و تعلم الألة و ذلك للاستفادة من مقدرة طرق التعلم العميق على اكتشاف انواع المركبات و قراءة أرقام ألواحها.
2. دراسة في تعدين البيانات و ذلك بالإستفادة من المعلومات التي ستتراكم في قاعدة بيانات هذا النظام في حال تشغيله لسنوات.

**5-4 الخاتمة :**

الحمد لله تعالى الذي وفقنا في تقديم هذا البحث، وها هي القطرات الأخيرة في هذا المشوار ، وقد كان البحث يتكلم عن (رصد حركة المرور بإستخدام الرؤية بالحاسوب)، وقد بذلنا كل الجهد والبذل لكي يخرج هذا البحث في هذا الشكل.

ونرجو من الله أن تكون رحلة ممتعة وشيقة، وكذلك نرجو أن تكون قد أرتقت بدرجات العقل و الفكر، حيث لم يكن هذا الجهد بالجهد اليسير، ونحن لا ندعى الكمال فإن الكمال لله عز وجل فقط، ونحن قد قدمنا كل الجهد لهذا البحث، فإن وفقنا فمن الله عز وجل وإن أخفقنا فمن أنفسنا، وكفانا نحن شرف المحاولة، واخيراً نرجو أن يكون هذا البحث قد نال إعجابكم.

وصل اللهم وسلم وبارك تسليما كثيراً على معلمنا الأول وحبيبنا سيدنا محمد عليه أفضل الصلاة والسلام.

* 1. **مصادر الدراسات السابقة :**

[1] [Automatic Traffic Using Image Processing](https://www.scirp.org/journal/paperinformation.aspx?paperid=78431#:~:text=This%20method%20uses%20image%20processing,or%20controlling%20the%20traffic%20signal.&text=number%20of%20hardware.-,This%20method%20uses%20image%20processing%20techniques%20to%20count)

Al Hussain Akoum Email: Hussain\_alkoum@hotmail.com. August 11, 2017.

[2] [Density Based Traffic Control System Using Image Processing](https://www.researchgate.net/publication/329235731_Density_Based_Traffic_Control_System_Using_Image_Processing)

Uthara E. Prakash ,AthiraThankappan , Vishnupriya K. T. , Arun A. Balakrishnan. November 2018.

* 1. **الإنترنت :**

[3] <https://www.scirp.org/> for research. 11/18/2019 -7:40 AM

[4] <https://www.researchgate.net/> , research. 11/18/2019 8:12 AM

[5] [Live Traffic Videos Playlist](https://www.youtube.com/watch?v=0iKFgEPOk20&list=PLi_shezBLcn1aIw90v4AgMP7b3yX_e1lR) from youtube. 7/23/2020.

[6] <https://www.tutorialspoint.com/python/python_multithreading.htm>. python multithreading help. 9/24/2020.

* 1. **الملاحق :**

1. **ملحق (1) شيفرة برنامج خوارزمية تتبع المركبات (Traffic Counter.py):**

import numpy as np

import cv2

import random

import string

import re

from datetime import datetime

import threading

import requests

import sys

# - Networking

SERVER\_URL = "http://192.168.43.234/mtApi/api/"

ClientCamID = 1

client = requests.session()

# - Recognotion Vars

videoFileName = "test 1.mp4"

cameraDevice = 1

roiWidth = (200, 1280)

roiHeight = (400, 500)

updatePeriod = 10 # seconds to update the server

updating = False

# ------- Analasys Variables :

Vehicles = []

newVehicles = []

totalCarsCount = 0

upCarsCount = 0

downCarsCount = 0

# ------

class myThread (threading.Thread):

def \_init\_(self, up, down):

threading.Thread.\_init\_(self)

self.up = up

self.down = down

def run(self):

global upCarsCount

global downCarsCount

global totalCarsCount

data = {"CameraID": random.choice(

[1, 2]), "InCount": self.up, "OutCount": self.down}

response = client.post(SERVER\_URL + "/Camera",

data=data, cookies=client.cookies)

if response.status\_code == 200:

print(

'Post Successfully : {up:'+str(self.up)+', down:'+str(self.down)+'}')

# reset counters

upCarsCount = 0

downCarsCount = 0

totalCarsCount = 0

else:

print("Network Error : " + str(response.status\_code))

#-------------------------------- Network

def abs(a):

return a if a >= 0 else a\*-1

def randomString(stringLength=3):

letters = string.ascii\_uppercase

return ''.join(random.choice(letters) for i in range(stringLength))

class Vehicle:

def \_init\_(self):

self.x = 0

self.y = 0

self.w = 0

self.h = 0

self.centerX = 0

self.centerY = 0

# new:just fonund matched:matched to previous frame unmatched:....

self.state = "new"

self.direction = 'up' # or down

self.name = randomString()

self.pointsHistory = [] # centroid points history

# as a blob by default it contains 1 vehicle this could become more incase of two vehicles becoming one big blob

self.contains = 1

self.directionChanged = False

def distToOther(self, other):

return abs(self.centerX - other.centerX) + abs(self.centerY - other.centerY) + ((abs(self.w - other.w) + abs(self.h - other.h))/2)

# recives a Vehicle object and a list of Vechicles

# and returns the closest object from ls to v, based on v.centerX and v.centerY [center points]

def getClosest(v, ls, stat):

closest = ls[0] # It's OKAY! ls is never empty here!

for ov in ls:

if ov.state == stat and closest.state != stat:

closest = ov

if v.distToOther(ov) < v.distToOther(closest) and ov.state == stat:

closest = ov

return closest

def countClosest(v, ls, stat, thresh):

sum = 0

for ov in ls:

if ov.state == stat and v != ov and v.distToOther(ov) < thresh:

sum += 1

return sum

def getNamesOfVehicles(ls):

str = ""

for v in ls:

str += v.name + " "

return str

def setDirection(v):

dirChangeThr = 5

global totalCarsCount

global downCarsCount

global upCarsCount

sum = 0

for i in range(0, len(v.pointsHistory)-1, 1):

sum += v.pointsHistory[i][1] - v.pointsHistory[i+1][1]

newDir = "up" if sum >= 0 else "down"

if v.direction != newDir:

if v.direction == "up" and v.y < roiHeight[0]+dirChangeThr:

# hmm, for now total++ and change the name

totalCarsCount += 1

upCarsCount += 1

v.name = randomString()

v.directionChanged = True

else:

if v.direction == "down" and v.y + v.h > roiHeight[1]-dirChangeThr:

totalCarsCount += 1

downCarsCount += 1

v.name = randomString()

v.directionChanged = True

v.direction = newDir

def isCloseToEdge(v, thresh=25):

if v.y < roiHeight[0] + thresh or v.y + v.h > roiHeight[1] - thresh:

return True

else:

return False

cap = cv2.VideoCapture(videoFileName)

fourcc = cv2.VideoWriter\_fourcc(\*'XVID')

out = cv2.VideoWriter('last run.mp4', fourcc, 25.0, (1280, 720))

ret, frame = cap.read()

dark = frame

fgbg = cv2.createBackgroundSubtractorMOG2()

while ret:

#gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# fills the region with black color [0,0,0]

dark[0:720, 0:1280] = ([0, 0, 0])

# Blurred version of the frame

blurred = cv2.GaussianBlur(frame, (35, 35), 0)

# Mixed effect of blurr and tinted black

darkBlurred = cv2.addWeighted(blurred, 0.3, dark, 0.7, 0)

cv2.rectangle(darkBlurred, (roiWidth[0], roiHeight[0]), (

roiWidth[1], roiHeight[1]), (0, 255, 200), 2) # drawing a green Rect

roi = frame[roiHeight[0]:roiHeight[1], roiWidth[0]:roiWidth[1]

] # Region Of Interest

darkBlurred[roiHeight[0]:roiHeight[1], roiWidth[0]:roiWidth[1]] = roi

carMask = fgbg.apply(roi) # Masking The Movement in ROI

bgrMask = cv2.cvtColor(carMask, cv2.COLOR\_GRAY2BGR)

darkBlurred[roiHeight[1]+10:roiHeight[1]+10 +

(roiHeight[1]-roiHeight[0]), roiWidth[0]:roiWidth[1]] = bgrMask

# ----- Blob Detection -------

contours, \_ = cv2.findContours(

carMask, cv2.RETR\_LIST, cv2.CHAIN\_APPROX\_SIMPLE)

newVehicles.clear()

# ------- Create Vehicles

for c in contours:

rect = cv2.boundingRect(c)

x, y, w, h = rect

if w < 50 or h < 50:

continue # too small, probably a Pedestrian

if w > 450 or h > 120:

# too big, probably the first frame [which is just black then suddenly an image appears]

continue

# --------------------

x += roiWidth[0]

y += roiHeight[0]

# ---- Initial Info

obj = Vehicle()

obj.x = x

obj.y = y

obj.w = w

obj.h = h

obj.centerX = obj.x + (obj.w/2)

obj.centerY = obj.y + (obj.h/2)

if obj.centerY <= ((roiHeight[0]+roiHeight[1])/2):

obj.direction = "down"

else:

obj.direction = "up"

# Append

newVehicles.append(obj)

# Clear blobs inside other blobs

for v1 in reversed(newVehicles):

for v2 in reversed(newVehicles):

if v2.y > v1.y and v2.y + v2.h < v1.y + v1.h:

newVehicles.remove(v2)

# compare newVehicles with Vehicles, this is the tracking code

trackingThreshold = 35

# but first...

for v in Vehicles:

v.state = "unmatched" # so that i can tell which vehicle have gone out of frame

for v in newVehicles:

if len(Vehicles) < 1:

v.pointsHistory.append((v.centerX, v.centerY))

v.state = "new"

Vehicles.append(v)

else:

closest = getClosest(v, Vehicles, "unmatched")

if v.distToOther(closest) < trackingThreshold:

# same same!

v.state = "matched"

v.name = closest.name

v.direction = closest.direction

v.directionChanged = closest.directionChanged

v.pointsHistory = closest.pointsHistory

v.pointsHistory.append((v.centerX, v.centerY))

v.contains = closest.contains

if len(v.pointsHistory) > 10:

v.pointsHistory.pop(0)

Vehicles.remove(closest)

setDirection(v)

Vehicles.append(v)

else:

# could be a new one... [a new vehicle coming from the sides of the frame]

# hmmm.....

v.pointsHistory.append((v.centerX, v.centerY))

v.state = "new"

Vehicles.append(v)

intersectionThreshold = 130

i = 0

while i < len(Vehicles):

v = Vehicles[i]

if v.state == "unmatched":

# intersection or seperation ???

closest = getClosest(v, Vehicles, "new")

newCount = 0

for vv in Vehicles:

if vv.state == "new":

newCount += 1

if v.distToOther(closest) < intersectionThreshold and closest.state == "new":

if newCount == 2: # seperation

if v.directionChanged == True and isCloseToEdge(v, 25):

totalCarsCount -= 1

if v.direction == "up":

upCarsCount -= 1

else:

downCarsCount -= 1

# intersection, however there should be 2 close "unmatched"

if newCount == 1 and countClosest(v, Vehicles, "unmatched", intersectionThreshold) == 2:

if closest.directionChanged != True and v.directionChanged == True:

closest.directionChanged = True

#closest.contains += v.contains

closest.contains = 2

pass

else:

if isCloseToEdge(v, 10): # out of frame

totalCarsCount += v.contains

if v.direction == "up":

upCarsCount += v.contains

else:

downCarsCount += v.contains

Vehicles.remove(v)

i -= 1

i += 1

# ------ Draw

for v in Vehicles:

cv2.rectangle(darkBlurred, (v.x, v.y),

(v.x+v.w, v.y+v.h), (255, 150, 0), 2)

cv2.putText(darkBlurred, v.name + " " + v.direction,

(v.x+v.w+10, v.y+v.h), 0, 0.5, (210, 255, 240))

cv2.putText(darkBlurred, "Contains " + str(v.contains),

(v.x, v.y-3), 0, 0.5, (210, 255, 240))

cv2.putText(darkBlurred, " State: " + v.state,

(v.x+v.w-10, v.y-3), 0, 0.5, (210, 255, 240))

# draw history points

for pt in v.pointsHistory:

cv2.circle(darkBlurred, (int(pt[0]), int(

pt[1])), 2, (255, 150, 0), 2)

currSec = int(datetime.now().strftime('%S'))

if currSec % updatePeriod == 0: # Every 20 seconds

if updating != True:

print('Posting... (' + str(upCarsCount) +

', ' + str(downCarsCount) + ')')

#thread = myThread(upCarsCount, downCarsCount)

# thread.start()

updating = True

else:

updating = False

cv2.putText(darkBlurred, 'press q to exit..',

(600, 20), 2, 0.7, (0, 255, 0))

cv2.putText(darkBlurred, 'Detected Objects : ' + str(len(newVehicles)) +

" [ " + getNamesOfVehicles(newVehicles) + "]", (10, 30), 2, 0.7, (0, 255, 0))

cv2.putText(darkBlurred, 'Current Vehicles : ' + str(len(Vehicles)) +

" [ " + getNamesOfVehicles(Vehicles) + "]", (10, 60), 2, 0.7, (0, 255, 0))

cv2.putText(darkBlurred, 'Total Count : ' +

str(totalCarsCount), (10, 90), 2, 0.7, (0, 255, 0))

cv2.putText(darkBlurred, 'Down Count : ' + str(downCarsCount),

(roiWidth[0]+5, roiHeight[0]-10), 2, 0.7, (0, 255, 0))

cv2.putText(darkBlurred, 'Up Count: ' + str(upCarsCount),

(roiWidth[1]-170, roiHeight[0]-10), 2, 0.7, (0, 255, 0))

out.write(darkBlurred)

cv2.imshow('Stream feed', darkBlurred)

ret, frame = cap.read()

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

out.release()

cv2.destroyAllWindows()

1. **ملحق (2) شيفرة برنامج مساعد (compress.py):**

لتقطيع الفيديو و كتابته مرة أخرى بدون صوت لتقليل الحجم:

import numpy as np

import cv2

cap = cv2.VideoCapture('test 2.mp4')

fourcc = cv2.VideoWriter\_fourcc(\*'XVID')

dim = (1280,720)

out = cv2.VideoWriter('compressed.mp4', fourcc, 30, dim)

timer = 0

ret, frame = cap.read()

while ret:

timer += 1

if timer >= 0:

#cv2.putText(frame, 'Recording...', (10,25), 0, 1, (0,0,255))

out.write(frame)

cv2.imshow('Compressing...', frame)

ret, frame = cap.read()

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

out.release()

cv2.destroyAllWindows()