**IV. Exemple d’application (système de gestion de parking avec reconnaissance).**

1. **Introduction**

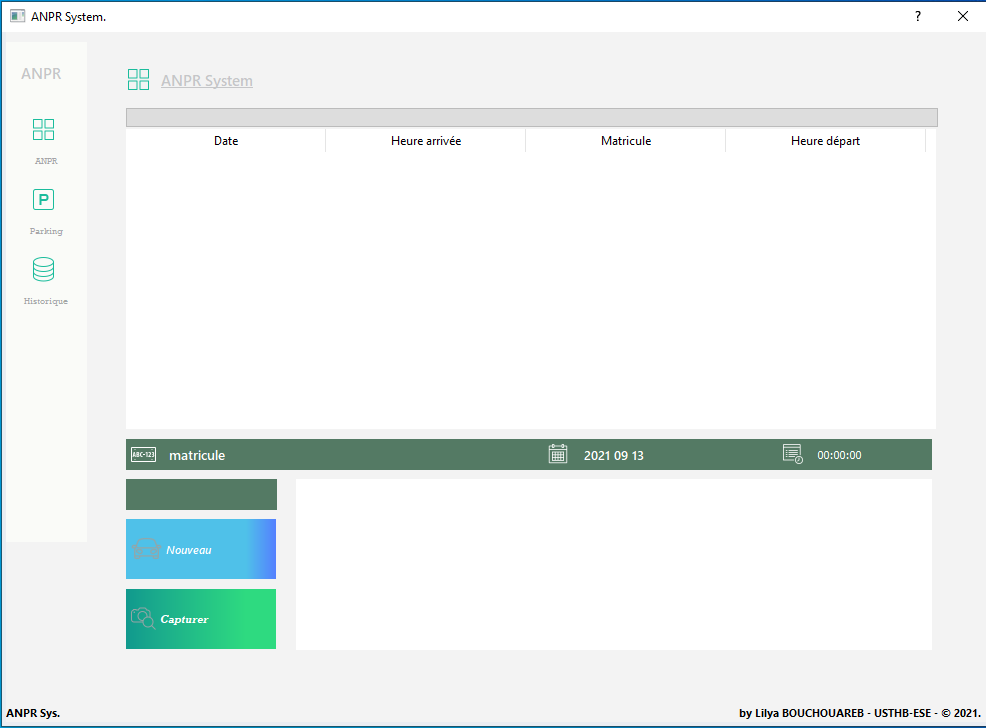
Afin d’illustrer les différents usages possible des systèmes de reconnaissances de matricule, un programme informatique a été développé pour l’occasion, il s’agit d’un logiciel nommé ***ANPR Sys.*** développé en Python (langage de programmation Fonctionnel/Orienté objet) pour

L’implémentation, et jumelé au code de reconnaissance décri plutôt via *Jupyter (*on parle de *Tensorflow* et des librairie OpenCv de traitemet d’image), ensuite une librairie en **C++** nommée **Qt** a été utilisée pour l’interface utilisateur graphique (GUI), ainsi que quelques lignes de codes en CSS (*Cascading Style* *Sheets*) ont servi au design du programme, et le langage ***Sql*** pour le stockage des données.

1. . **Présentation du programme**

Le programme se présente sous forme d’une interface interactive (Figure IV.1), où l'utilisateur peut sélectionné l’image d’une voiture (image prise préalablement via la camera et enregistrée automatiquement dans un fichier spécifique), une fois l’image prise on clique sur **nouveau** ce qui permet d’introduire l’image de la nouvelle voiture afin de la traiter et reconnaitre le matricule, une fois l’image introduite la fonction **Capturer** une fois cliquée permet de reconnaitre le matricule et de l’introduire dans la base de données et ainsi marquer **l’heure** d’arrivée, **la date** et **le matricule**.

Une fois ces données reçu le programme permet de savoir quelles sont les voitures présentent dans le parking (Menu : Parking), de marquer leur départ, et enfin d’avoir un aperçu complet du trafic du parking en termes de voitures passées par là (heures d’arrivée/départs, dates d’arrivée/départ et matricules).



**Figure IV. 1 ANPR Sys. Interface utilisateur**

Le programme se décompose en cinq principales parties (Figure IV.2) :

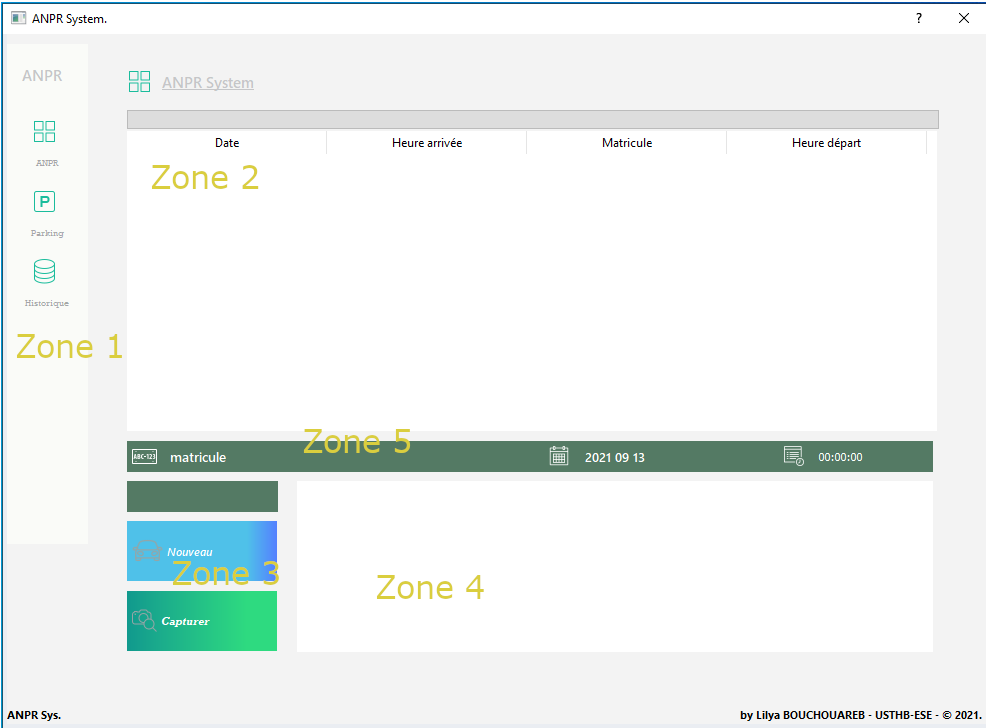
* **La partie menu (zone 1)**, qui comme son nom l'indique, permet de naviguer entre les différentes partie des logiciel
  + - **ANPR Sys** pour la détection des matricules et l’introduction de la voiture dans la base de données du système.
    - **Parking** pour la visualisation en liste des voitures actuellement garées (places occupées), et le marquage du départ d’une voitures.
    - **Historique,** une fois la voiture marquée comme partie, elle est affichée avec le reste des voitures passée par le parking, avec le traçage des horaires d’arrivée et de départ.
* **La partie Affichage en liste (zone 2)**; elle représente la partie dans laquelle sont

Affichées les données relatives aux horaires arrivée/départ ainsi que les matricules, cette zone se répète quelques soit la partie du logiciel où l’on se trouve, les données qu’elle affiche varient.

* **La partie** **fonctions (zone 3)**, il s’agit de la partie qui permet de sélectionner l’action à faire selon l’endroit du système ou l’en se trouve (**ANPR Sys, Parking, Historique)**.
* **La zone d’affichage des images et matricules (zone 4)**, elle concerne

L’affichage des données images de la voiture arrivant au parking, et d’afficher l’image du matricule une fois **capturée** par la reconnaissance.

* **La zone d’indication des évènement (Zone 5)**, une fois le matricule capturée cette zone permet d’afficher à l’utilisateur les donnée horaire et matricule actuellement capturées.

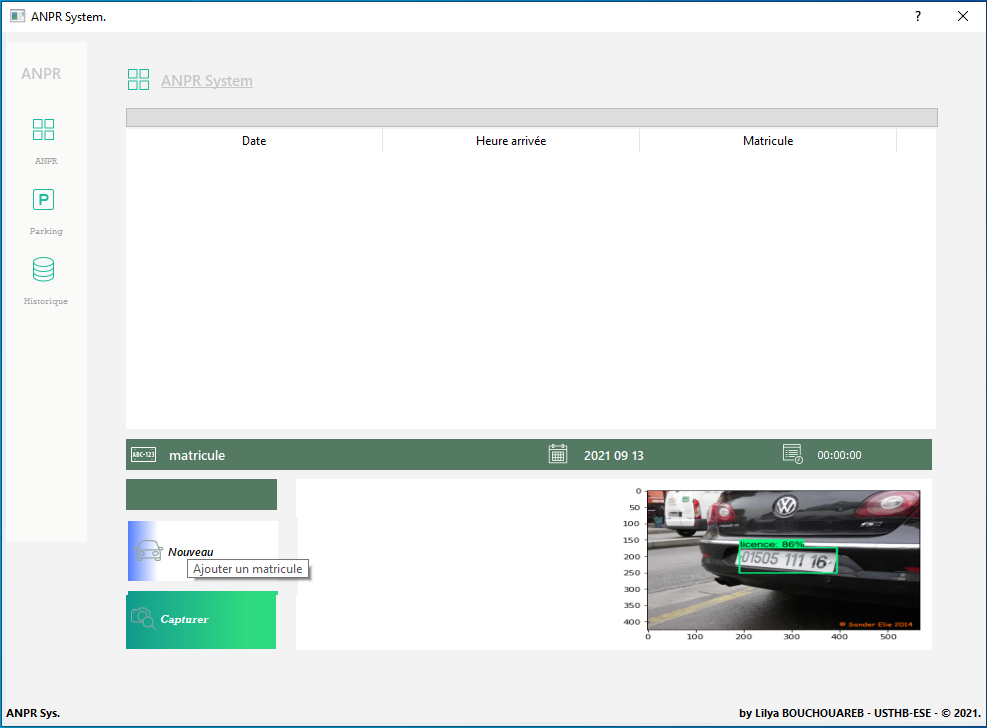


**Figure IV. 2 Différentes parties du programme ANPR Sys.**

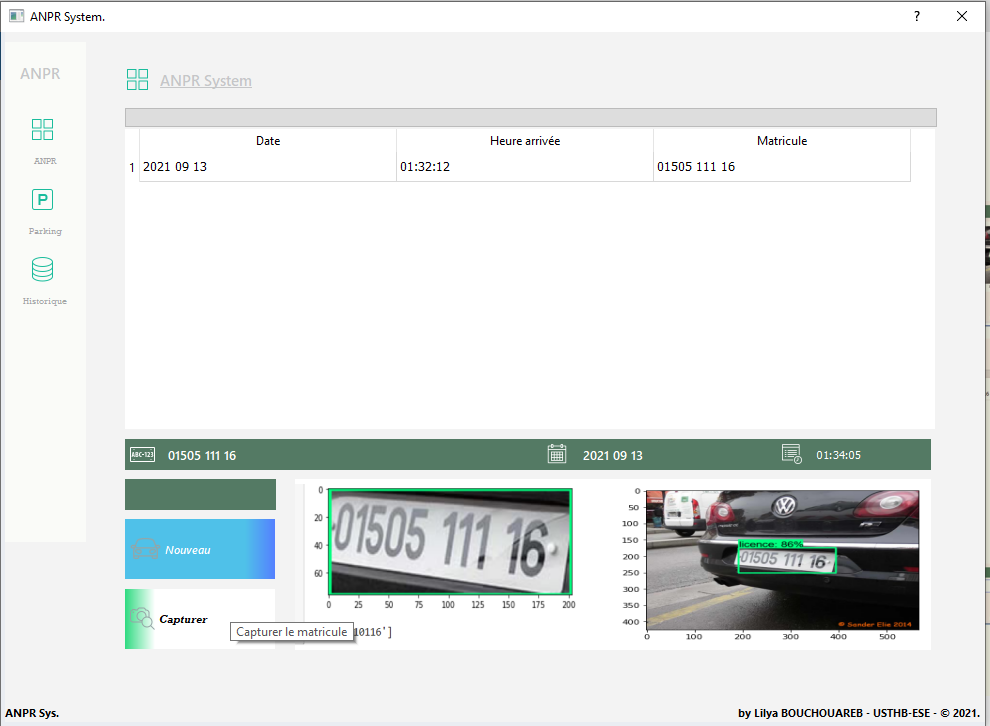
1. **Capture du matricule et stockage des données**

Une fois dans la page d’accueil (**APR Sys.**) une fois l’image prise on clique sur **nouveau** ce qui permet d’introduire l’image de la nouvelle voiture afin de la traiter et reconnaitre le matricule (Figure IV.3), ainsi l’image introduite la fonction **Capturer** cliquée permet de reconnaitre le matricule et de l’introduire dans la base de données et ainsi marquer **l’heure** d’arrivée, **la date** et **le matricule** (Figure IV.4).

Ce qui mène à introduire la voiture dans la base de donnée et d’afficher ces horaires d’arrivée das la zone d’affichage (Zone 2).



**Figure IV. 3 Introduction de l’image à traiter, bouton *Nouveau***

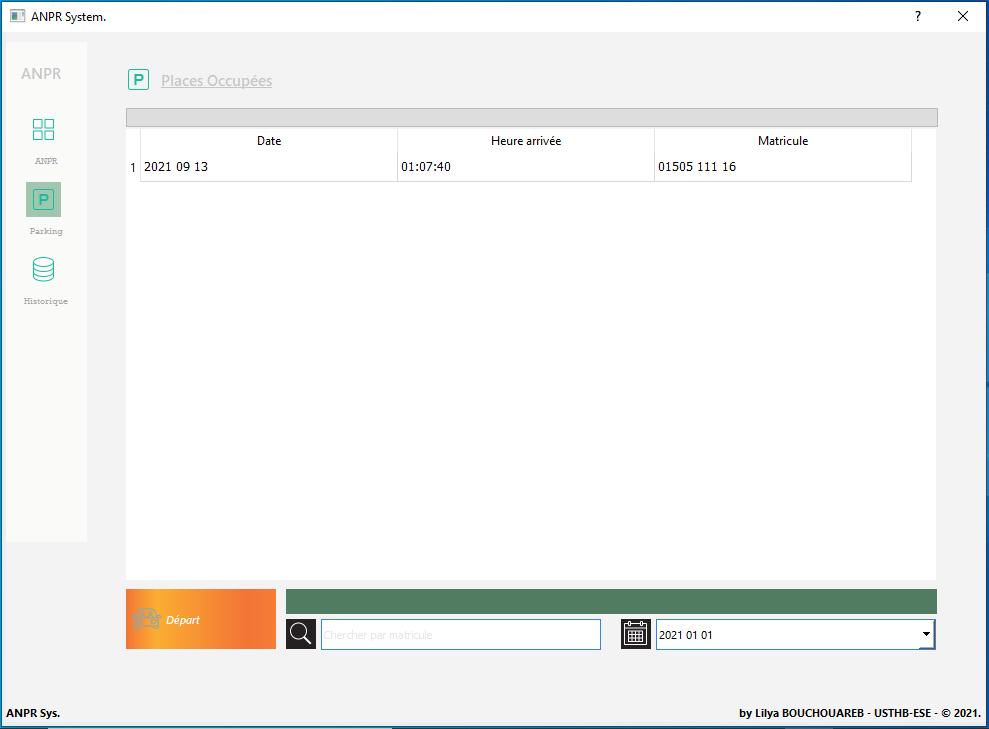
****

**Figure IV. 3 Reconnaissance et Capture du matricule, et stockage des donnée**

**4. Traitement des données et visualisation du trafic**

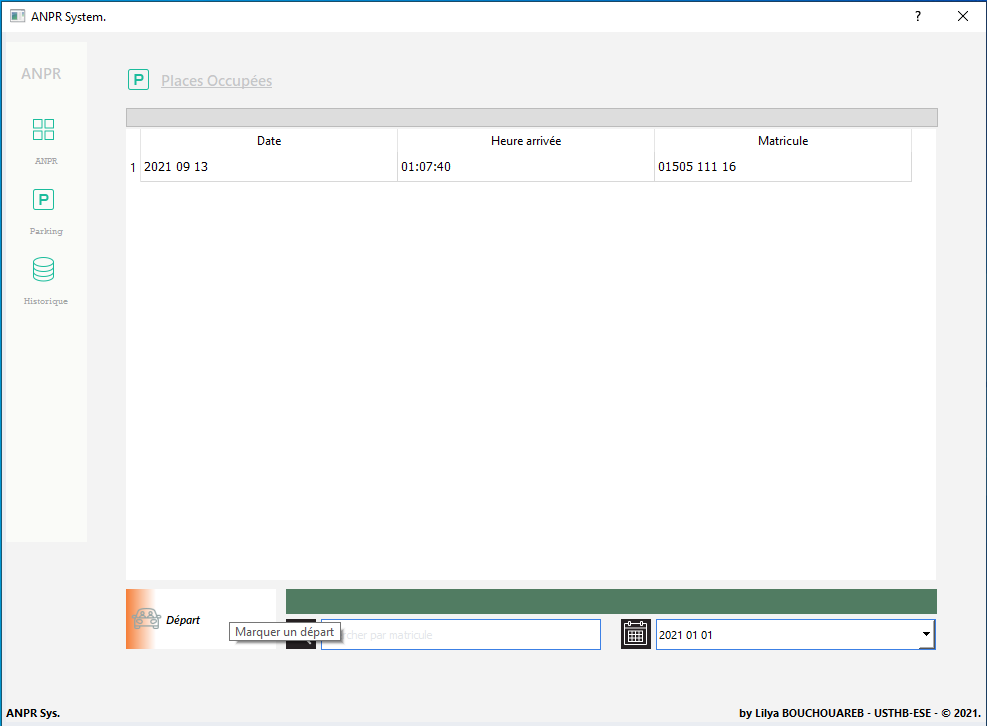
Une fois que nous avons procédé à la capture et l’enregistrement des données de la voiture, on retrouve sa trace via le **menu : Parking** (Figure IV.5), qui permet de :

* Visualiser dans la zone d’affichage les voiture actuellement garées.
* Rechercher une voiture stationnée par **date** ou **matricule** en cas de besoin.
* Marquer les départ d’une voiture (Figure IV.6).



**Figure IV. 5 Menu Parking.**

Le marquage du départ d’une voiture (Figure IV.6), permet de marquer une voiture comme ayant quittée le parking, de ce fait elle sera répertoriée et archivée dans la base de données comme voiture ayant stationnée un lapse de temps donné, que l’on peut récupérer dans **menu : Historique**.

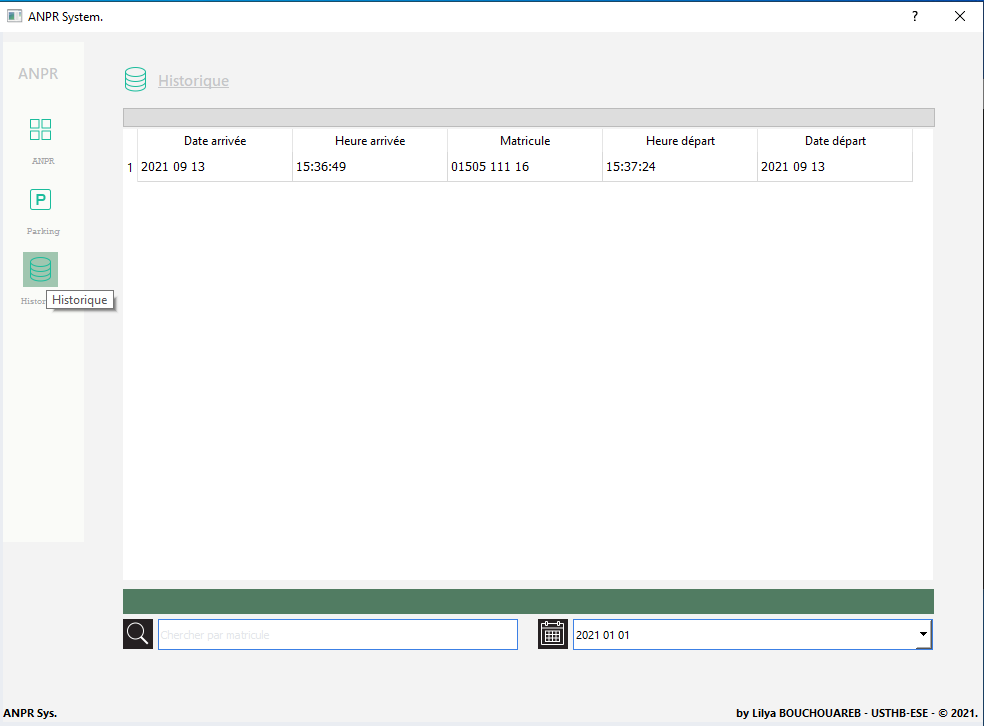


**Figure IV. 6 Marquage du départ d’une voiture.**

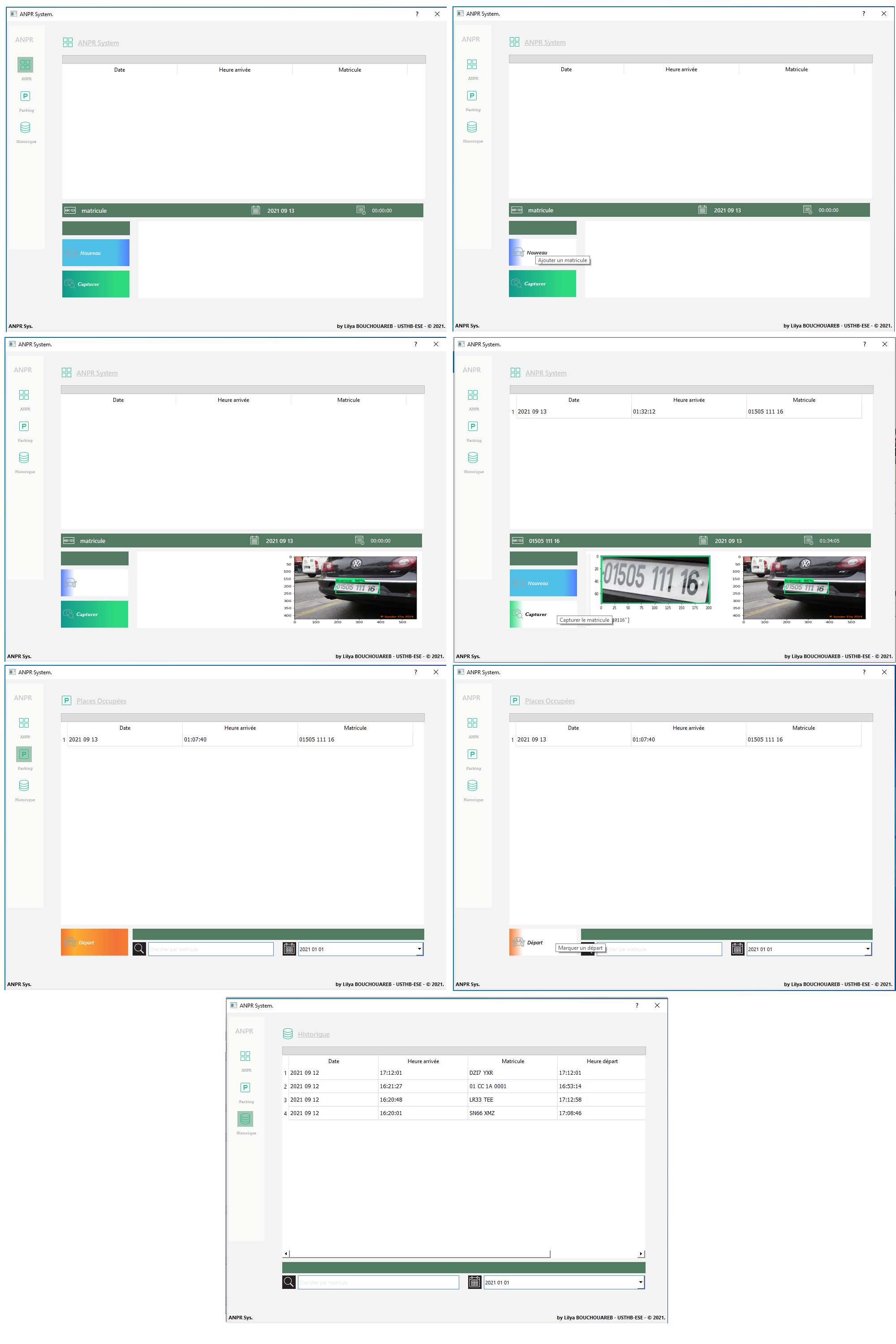
**5. Historique du trafic**

Pour chaque voiture passée par le parking, le programme permet de savoir l’historique des passages via **menu : Historique** (Figure IV.7), qui permet de :

* Visualiser dans la zone d’affichage les voitures passées par le parking, répertoriées par dates et heures d’arrivée/départ.
* Rechercher une voiture par **date** ou **matricule** en cas de besoin.

****

**Figure IV. 7 Menu : Historique.**



**Figure IV. 7 Mosaïque des différents partie/sections du programme.**

**5. Conclusion**

Le logiciel est conçu pour être open source et offrir une illustration d’un éxemple d’utilisatio pratique de la reconnaissance de matricules (ANPR) ou autre projets relatifs, il se présente comme support ou référence, son code source sera intégré à une plateforme de partage de logiciel et de codes sources et autres types de projets (github).