Smart Parking System
(SPS)

Abstract

With the increase in population density in the Kingdom of Saudi Arabia especially in Riyadh city which is considered the capital of Saudi and the largest Arab cities in term of area also the fastest expanding cities in the world.

With all these reasons, the crowding increases, especially in the morning and evening peak times.
One of the problems that facing the city is the improper parking which consider the most common causes of traffic congestion and disruption not only in Riyadh but in the whole Kingdom. Therefore, this behavior is considered a traffic violation in the Saudi Traffic Law, It is called an “Illegal parking violation”.

Our goal in this project is to serve The General Saudi Department of Traffic from a side and the civil from the other side by building a system that contains 2 main services:

1- chatbot that can answer all the questions, receive the inquiries and objections that are being asked by the civilian.
2- Detect all parking issues.

For more explanation, this project is based primarily on two deep learning applicants

1- Natural Language Processing NLP.
   The concept is about analyzing the text that sent by the end-user (Input) then the model will find the best answer to replay within an active human-based conversation (Output).

2- Image processing.
   Capturing the image frames from videos and do object detection to determine:
   - Occupied parking.
   - Available parking.
   - Disabled parking.
   - Parked in wrong space.
   - One in two parking.

Pre-processing:

1-Capture frames from the video that taken by Drone
   For each frame
   Path to png
   Gray scale
   Edge detection

2- draw boundaries using mouse click around the car using cv2.setMoudeCallBack()
3-find the object boundaries coordinates
4- apply those coordinates on the parking space
5-seprate each object from the frame to treat as our dataset
6- set the image size to 72*72
7- Do the image generator using the augmentation to balance the data

Create wight to detect all the cars photo that has been taken by the Drone because we used transfer learning and all the models detect the objects and capture it as a cell phone!

Split Data to training and testing:
1-We have 2 classes (empty [0], occupied[1])
2- Flatten
3-Loading the model
4-make and fit the model

Models:

Pre-trained models such as :

CNN:

- Xception
- Vgg16
- Resent50

OpenCV:

- MOG
- KNN

YOLOv5

Part I : Natural language Processing – Chatbot

Data Collection

The dataset that we used comes from:

1. Surveys we spread to collect as much as we can of sentences and words that related to our project.
2. Web scraping of articles that covers the 3 basic scenarios that meets our project expectations:

- Inquiry about a violation.
- Reporting a violation such as: having a lonely child in the car, My car has been scrached.

- Wrong parking caused line traffic.

Pre-Processing

- Remove number and punctuation.
- Create stop word dictionary “prepositions and pronouns” and then remove them all from the dataset.
- Replace the word “وراي” by the word “خلفي”.
- Normalize the following from Camel tool:
  - All "أ، إ ، آ" with “ ا ”
  - All “ ه ” to “ ة ”
  - All “ ى ” to “ ي ”
- Stemming from Farasah library
- Part of speech (P.O.S) using Camel tool `model CAMeL-Lab/bert-base-arabic-camelbert-mix-pos-glf`
  لقيت سيارتي مصدومه the tool could figure out that the word لقيت is verb

Models

First Model:
LDA (Latent Dirichlet Allocation)
LDA is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.
We apply the model and then print all the results. however, it wasn’t that good cause there was a coherence and this method isn’t applicable on Arabic.

```
[ ]  lda.print_topics()

    [(0,
      '0.217*"مليح"*0.024 + "نايم"*0.024 + "زجاج"*0.024 + "مفتل"*0.024 + "ناس"*0.028 + "وحد"*0.043 + "أطفال"*0.051 + "طفل"*0.101 + "مخالفة"*0.108 + "سيارة"'),
     (1,
      '0.175*"أطفال"*0.024 + "خلف"*0.026 + "متوقف"*0.029 + "موقف"*0.045 + "مخالفة"*0.052 + "وجد"*0.064 + "طريق"*0.074 + "تشخص"*0.086 + "مفتل"*0.134 + "سيارة"'),
     (2,
      '0.118*"أشخاص"*0.032 + "ناسي"*0.032 + "محتجز"*0.032 + "طفل"*0.032 + "أطفال"*0.048 + "مخالفة"*0.071 + "أعرف"*0.087 + "سيارة"*0.087 + "نوع"*0.095 + "مخالف"'),
     (3,
      '0.218*"سبب"*0.026 + "مطق"*0.026 + "مخالفة"*0.027 + "محجوز"*0.031 + "مراي"*0.035 + "انكسر"*0.035 + "طريق"*0.035 + "تشخص"*0.044 + "مفتل"*0.093 + "سيارة"'),
     (4,
      '0.188*"مفتل"*0.028 + "هرب"*0.033 + "حك"*0.033 + "وجد"*0.040 + "واحد"*0.042 + "هي"*0.051 + "صدم"*0.061 + "واقف"*0.070 + "أحد"*0.070 + "سيارة"'),
     (5,
      '0.087*"طفل"*0.037 + "مخالفة"*0.042 + "مخالف"*0.046 + "محجوزين"*0.046 + "وجد"*0.046 + "داخل"*0.051 + "مطق"*0.055 + "أطفال"*0.078 + "مركب"*0.082 + "سيارة"'),
     (6,
      '0.229*"حك"*0.026 + "هرب"*0.028 + "وجد"*0.029 + "خلف"*0.030 + "أطفال"*0.035 + "وجد"*0.035 + "طفل"*0.038 + "أحد"*0.044 + "تشخص"*0.061 + "سيارة"')]
```

Second Model:
NMF (Non-negative Matrix Factorization)
First apply TF_IDF factorize, fits it to NMF, then print the result.

```
[44] for i in range(0,len(topics_NMF)):
       print(topics_NMF[i])
```

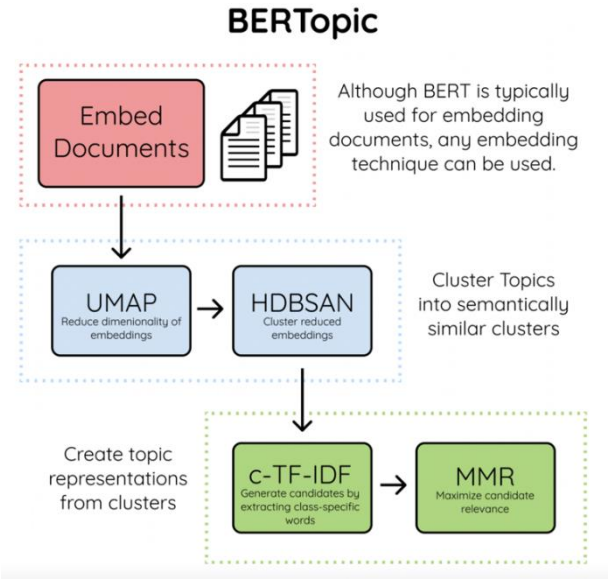
```
['اطفال', 'خط', 'مركب', 'اولاد', 'أحد', 'سيارة', 'وجد', 'شخص', 'طريق', 'مقفل']
['جديد', 'استفسار', 'عدد', 'تفصيل', 'مجموع', 'أظهر', 'مرصوده', 'كم', 'مبلغ', 'مخالفة']
['اهل', 'ناسيين', 'نايم', 'نائم', 'وجد', 'مغلق', 'محجوز', 'وحد', 'سيارة', 'طفل']
['غلط', 'خلف', 'واحد', 'سيارة', 'هرب', 'هي', 'واقف', 'صدم', 'حك', 'أحد']
['وجد', 'نائم', 'محجوزين', 'داخل', 'مغلق', 'وحد', 'مركب', 'سيارة', 'محتجز', 'أطفال']
['غلط', 'شكل', 'خاطئ', 'زجاج', 'موقف', 'كسر', 'متوقف', 'سيارة', 'خلف', 'شخص']
['خدش', 'طافي', 'صدمة', 'عدد', 'كم', 'عند', 'ماسبب', 'أعرف', 'نوع', 'مخالف']
```

Third Model (The chosen model):
BERT_for_Arabic_Topic_Modeling_ACLing2021

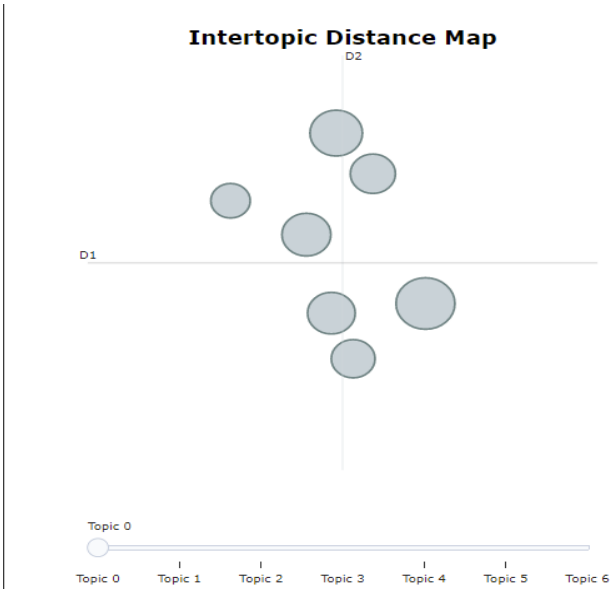
First, we have to know how the BERTopic modeling technique works? it uses transformers (BERT embeddings) and class-based TF-IDF (Term Frequency — Inverse Document Frequency) to create dense clusters. It also allows you to easily interpret and visualize the topics generated.
BerTopic algorithm contains 3 stages:

1. **Embed the textual data (documents)**
   In this step, the algorithm extracts document embeddings with BERT, or it can use any other embedding technique.
2. **Cluster Documents**
   It uses UMAP to reduce the dimensionality of embeddings and the HDBSCAN technique to cluster reduced embeddings and create clusters of semantically similar documents.
3. **Create a topic representation**
   The last step is to extract and reduce topics with class-based TF-IDF

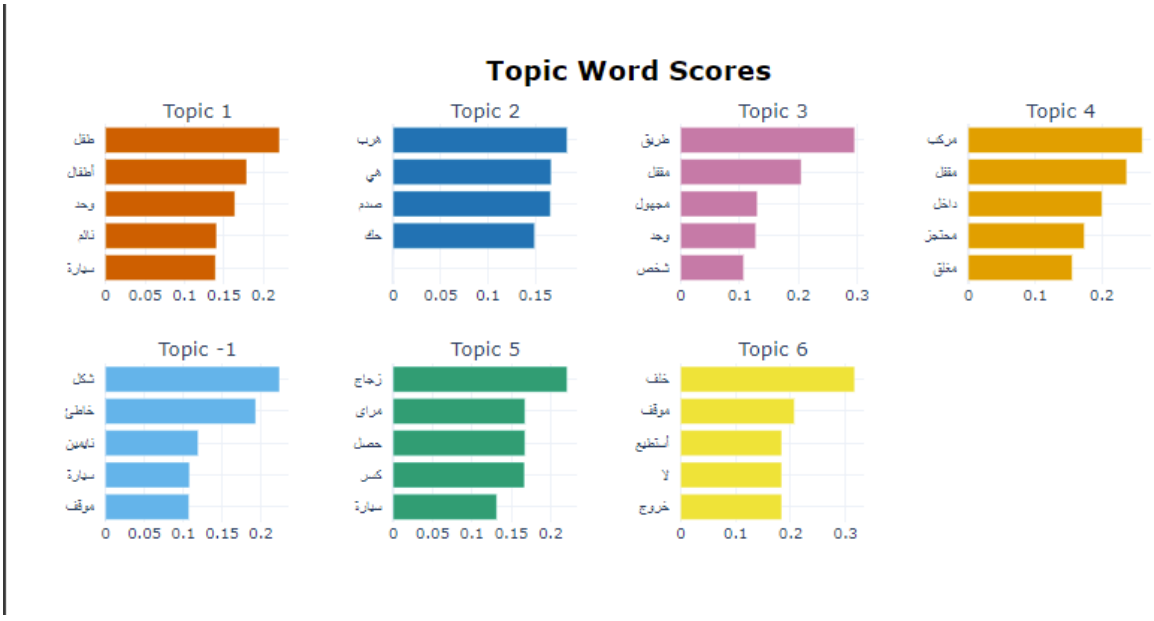
The figure below represents how the model will be processed:



Overthrow, In our project the BERTopic Pre-Trained Arabic Language Model is embedded, and compare its results against LDA (Latent Dirichlet Allocation) and NMF (Non-negative Matrix Factorization) techniques.
We used Normalized Pointwise Mutual Information (NPMI) measure to evaluate the results of topic modeling techniques. The overall results generated by BERTopic showed better results compared to NMF and LDA.
The big plus for this model that it supports the different Arabic dialects especially the Gulf countries dialects, for example the word "لقيت" is conver to "وجد" and so on on many local words.



This figure above represents the intertopic distance map, in this project the group agreed on choosing seven as a fixed number for the number of clusters (topics).



- The figure above represents the BarH plot, the plot shows the frequency of word in each topic,the plot shows seven diffrent topics, every subplot shows the most words occured within the topic.

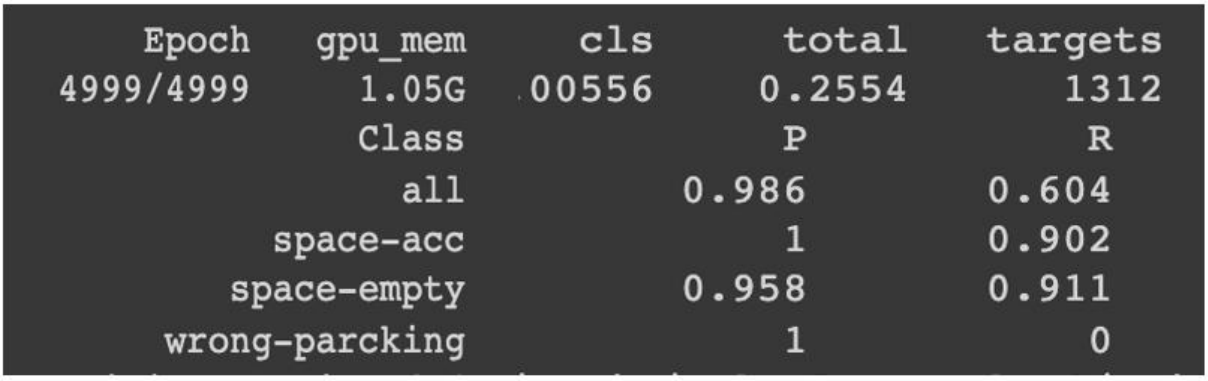
- The (-1 topic) represent topic has no relation with the other topics (outlier), this topic occured due to that data collected from users by google form, which some of them did not filled the form correctly.

Part II: Deep learning – Detect all parking issues

FIRST APPROACH: Object detection with (Drone)

- The Dataset: the Dataset has been collected from captured frames of videos with more than 12 thousand images with different angels of cam
- The Annotation: in this approach Roboflow application was used for Annotation labeles as follows:
    - Empty-space
    - Occupied-space
    - Wrong-parking

- The used model: in this approach the used model is YOLOv5 (last version of yolo) which is has been released within the last 6 months, the model got trained on yolov5x.pt(wight).
- The output of model: the output of the model shows Number of spaces of the occupied parking spaces and available space using object detection to achieve this goal.



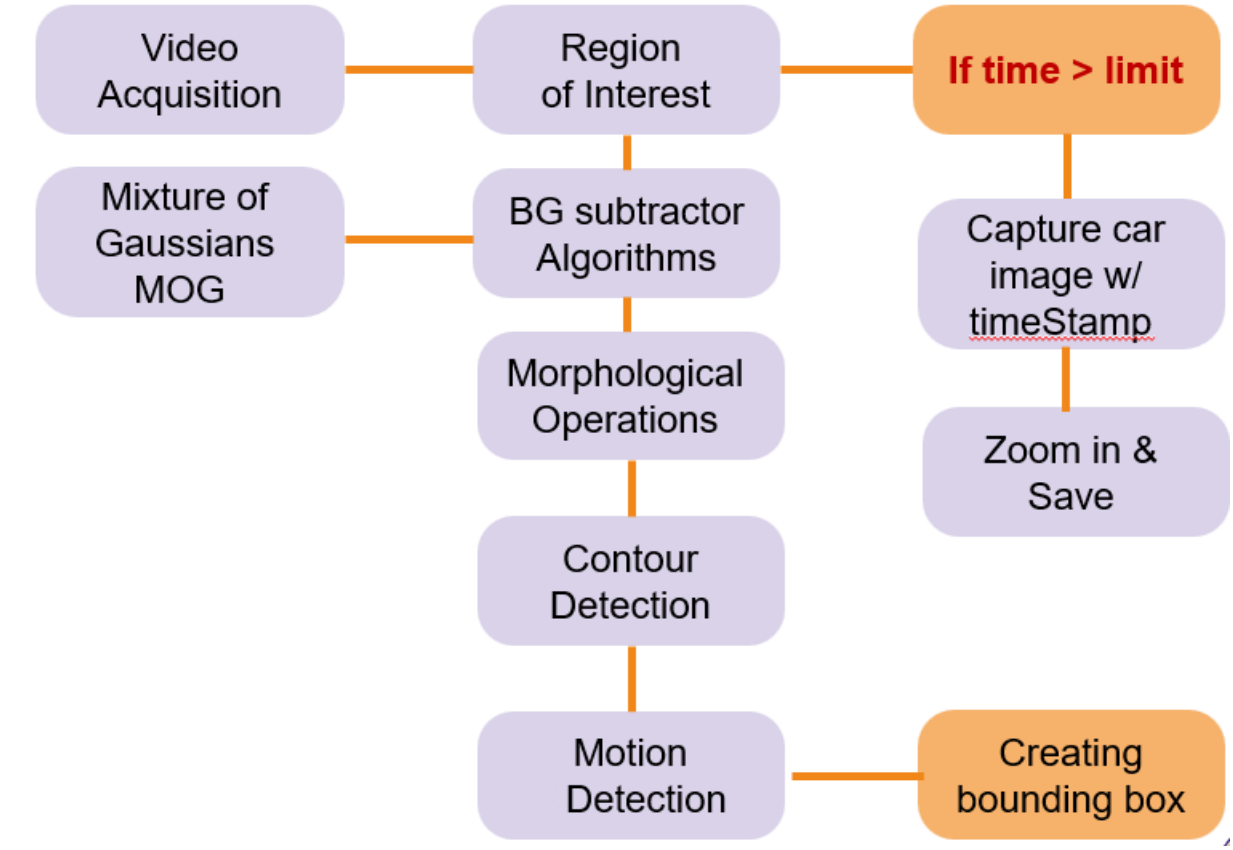
Epoch	gpu_mem	cls	total	targets
4999/4999	1.05G	.00556	0.2554	1312
	Class		P	R
	all		0.986	0.604
	space-acc		1	0.902
	space-empty		0.958	0.911
	wrong-parcking		1	0

Second APPROACH: Real time detection and object tracking

- The Dataset: the Dataset has been collected from captured frames of videos, Since we don't have dataset of double parking, we took a tracking object as an alternative approach.
  - The used model: in this approach the used model is resnet50 with algorithms includes MOG and KNN which implemented with OpenCV.
  - The output of model: the output of the model shows Number of spaces of the occupied parking spaces and available space using object detection to achieve this goal, also the model shows the wrong parking's in different types, also the model calculate the time for every detected car.

  The figure below represents how the model will be processed:

Parking Violation Workflow



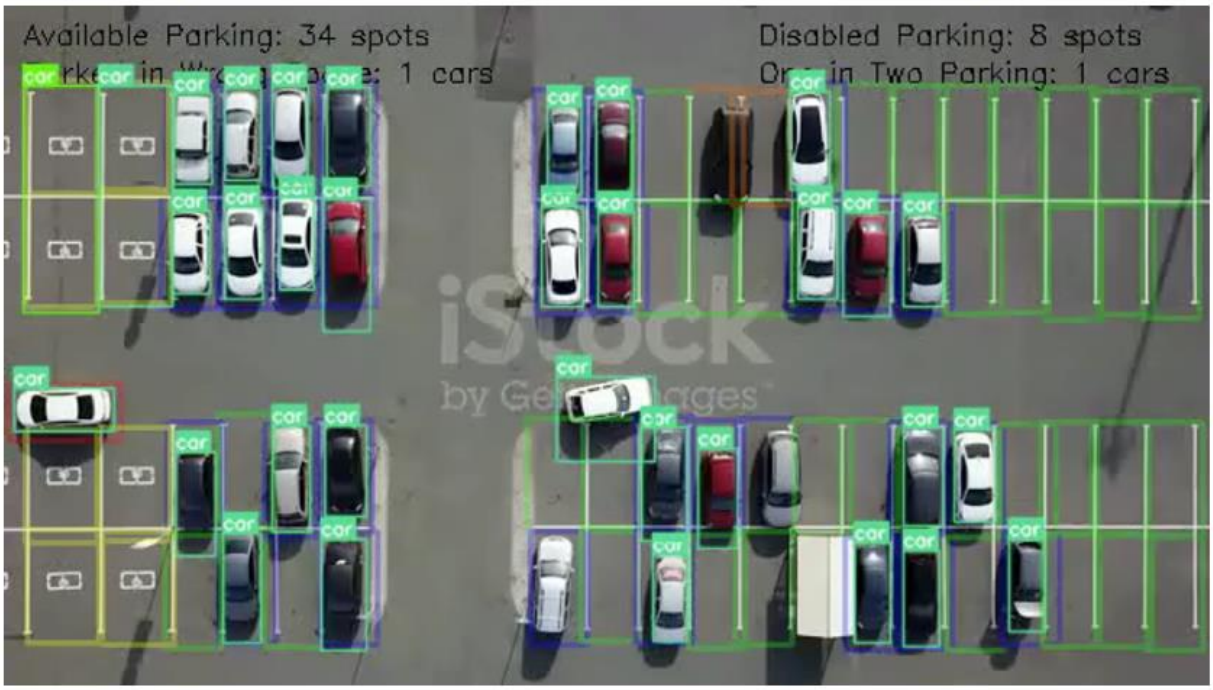
Third APPROACH: Combining two models. Combining two models Resnet50 and YOLOv5, due to the fact that yolov5 works better with 1500 image for each class.

- The Dataset: The Dataset has been collected from captured frames of videos with more than 12 thousand images with different angels of cam

  - The used model: in this approach the used model is YOLOv5 since it is the best model for detection and tracking object and Resnet50 is the best in classification we made combination of two of these models which enhanced the accuracy and speed up the model in processing.
  - The output of model: the output of the model shows Number of spaces of the occupied parking spaces and available space using object detection to achieve this goal, also the model shows the wrong parking's in different types, also the model calculate the time for every detected car.

  The figure below represents how the model will be processed:

Combine 2 Models (RESNET50 and YOLOv5)

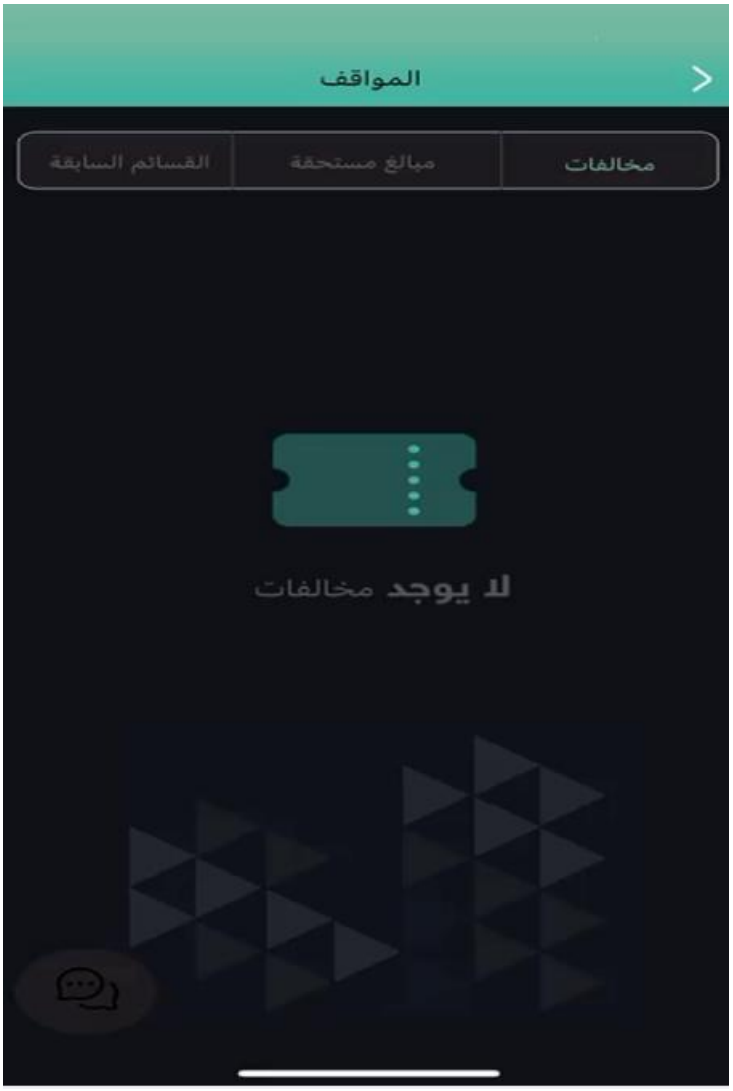


Part III: Graphical user interface (GUI) – Twakklna prototype user interface

The Graphical user interface:

- The tools: in this section we going to mention the used tools in this project
  1- Figma in designing the graphical user interface
  2- Bravo studio in creating the design into code and connect it database
  3- Ari table in creating real time Database

  The pages model: this prototype include different pages as follows
  - the main page of this prototype (with no fines)
  - the page of chatbot user interface
  - the page of previous fines
  - the page fines


المواقف
مخالفات
مبالغ مستحقة
القسائم السابقة
لا يوجد مخالفات