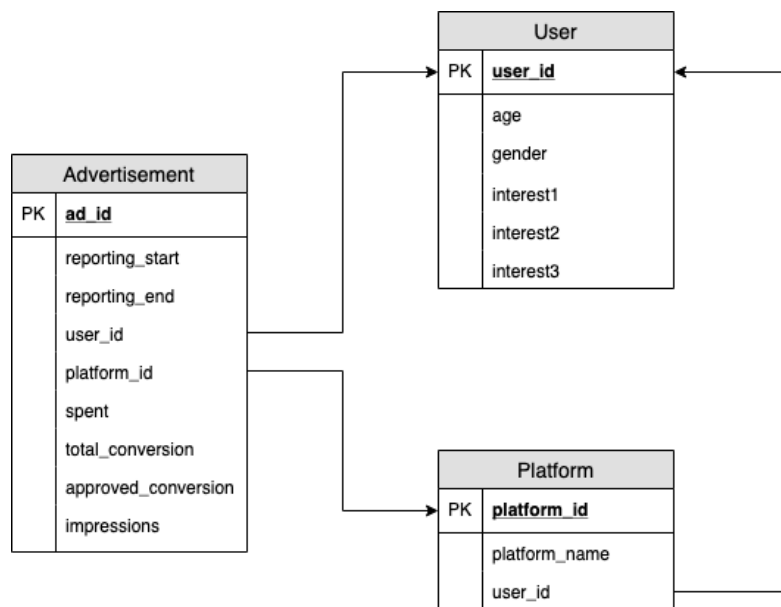


# Bring the bidding model to production

**1. Keep data organised.** To bring the model to production, our model should be able to efficiently process a massive amount of noisy data from many different sources. Therefore, the most important thing is to keep data organised and structured, so it is convenient for data retrieval and data modelling. For example, since we have campaigns with several platforms (Facebook, Google, Youtube), we need to aggregate the information from different sources and store them with a uniform data structure. As we can see in the figure below, I designed the most straightforward database schema to store our user's profile and campaign performance. Besides, we can create a parser program to transfer the raw data from the web into a standard format and store them in the database atomically. Furthermore, we can also compare the ROI of the campaign from different platforms and assign more credits to the best performance platform. Our bidding model may not suit for all platforms, so we can apply different bidding model (or same model but different hyper-parameters) which optimised specifically for each platform.

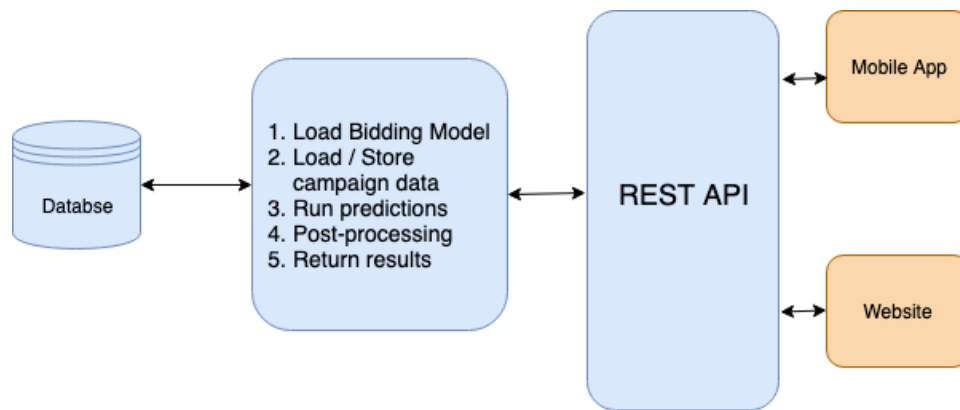


## 2. Identify our needs.

I introduced two approaches in the report to build a good bidding model; one is to use regression models to predict the ROI (return on investment) based on the user's profile; another approach is to use a classification model to classify our target customers. For the regression model, we can use the loss value to measure our model's performance. For the classification model, we can use the accuracy rate and prediction precision rate as the metrics, because we only want to spend money on our target customers.

## 3. Build an End-to-End system.

To improve our bidding performance, we need to build an automated real-time bidding system. As we can see in the figure below, I designed a toy structure of the system. For example, Database can store the historical campaign data and machine learning models. We have a build pack to load models/data and make predictions. Then we can return our prediction results to the website/mobile\_app through the REST API. Finally, the website/



mobile\_app can perform some actions based on the predictions. The advantage of this system is that it requires minimal human intervention and fast-response, which save us a lot of time and human resources. The disadvantage is that the system may make obvious mistakes with the unseen data. Therefore, we still need a smart human to monitor the decisions made by the system to avoid a huge loss.