Background information

We have some parking lots where users can stay without any prior registration. Those so called short term parkers have to pay for the duration of their stay. When they drive out of the parking lot, they pay at a credit/debit card terminal. Per law they have to receive a (signed) receipt immediately. Thus we display a receipt next to the terminal, in the form of a QR code. The QR code will lead the user to a webpage where he can download his signed receipt. Since we assume users will not scan / photograph the QR code, the website will have an option to search for a receipt if the user enters the following information: the date of payment, license plate and the last 4 digits of the card-number. This way, if someone forgot to scan his receipt QR code, he can always search for it online if he knows when he stayed at the parking lot.

Your task

Your task is to provide the search function described above. Attached you find some example data, how the individual log entries are sent in our backend. It is a json entry per line. You cannot parse the whole file as one json object, you have to parse every individual line. It contains different messages but most importantly **payment messages**. **Parse the data** and **store it** somewhere. You should build a **user interface** where the user can enter the **license plate**, the **date of payment** and the **last 4 digits of the card number**. If a matching entry is found, **display some information** about it. Otherwise an error message would be nice.

The data format

Each message consists of `["topic", {... payload}]`.

Payment messages

Payment messages are sent when a payment was requested and when a payment was completed. The license plate can be found in **`payload.om_payload`**. All information about the payment is stored in **`payload.payment_payload`**. The card number field in **`payload.payment_payload.extra.card_number`** has a funky format, try to find out how to decode it. The time of payment can be found in **`payload.timestamp`** and it is stored as a unix timestamp.

Form of the submission

Choose any programming language you like. The user interface can also be of any kind (Native UI, Web UI, Terminal, Voice-controlled, ...). You can invest as much or as little time as you want (If it's done in 5 min. and it works it may be more worth than a whole gamut solution that took 2 months). You should write your code the way you would write it anyways (if you are overly detailed you may be slow, if your code is a mess, it's not easy to maintain).