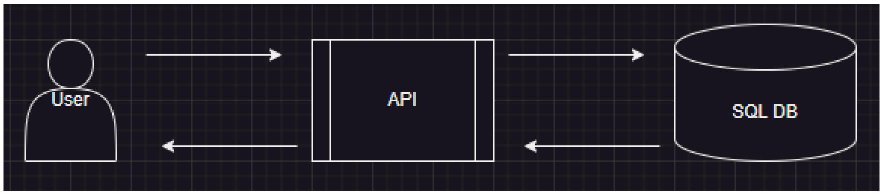
# PASTA worksheet

| Stages | Sneaker company |
|---|---|
| **I. Define business and security objectives** | Company goals are to protect user PII/SPII. We must consider the industry standard, PCI DSS when evaluating this app. |
| **II. Define the technical scope** | Company will use APIs, PKI: AES/RSA encryption, SHA-256, and SQL<br>After reviewing the technology Company is using, we will first start with identifying threats to their SQL database. Verifying form/input validation, input sanitization, and any prepared statements is a fairly quick process, but will reveal high risk coding practices that need to be addressed. |
| **III. Decompose application** | Each user will communicate with the API as they click on products and shop, and the API will be communicating to the SQL database to fetch data. The API sends information to the browser, but this is a simplified diagram.<br> |
| **IV. Threat analysis** | Internal threats could include misconfigured APIs & databases or disgruntled employees.  External threats Company could face are SQL injections, malware, DoS/DDoS and social engineered employees |
| **V. Vulnerability analysis** | Company using SQL for their DB leaves them exposed to query code being run through user inputs, known as SQL injections. There are possible weaknesses within the DB itself if hashing isn't being applied to user's sensitive data like passwords. There's also no information found on firewalls which leaves Company exposed to a slew of network attacks. |
| **VI. Attack modeling** | Threat actors can infiltrate this application through the app itself |

| | |
|---|---|
| | or through the users when they access it. SQL injections would be using the input fields on the app forms to run query code. The other option would be a session hijack, involving the act of stealing a user's session ID/token to impersonate them. If the threat actor could time when an admin or someone with elevated permissions will log in, then the threat actor could pretend they're an admin and do serious damage from within. |
| **VII. Risk analysis and impact** | The first security control recommended to be implemented is encrypting/hashing user passwords/cc numbers on the SQL DB. Second, following the AAA framework and implementing OAuth or another verification software would ensure buyers and sellers that log into the system are verified users. Third, to prevent internal threats, using least privileges and separation of duties will ensure nobody has too much access in the system and only has enough access to complete their day-to-day tasks. Lastly, it's time to be absolutely positive that every button and link and form on the app is coded correctly to protect against SQL injections and other malicious activity that can be uploaded. |