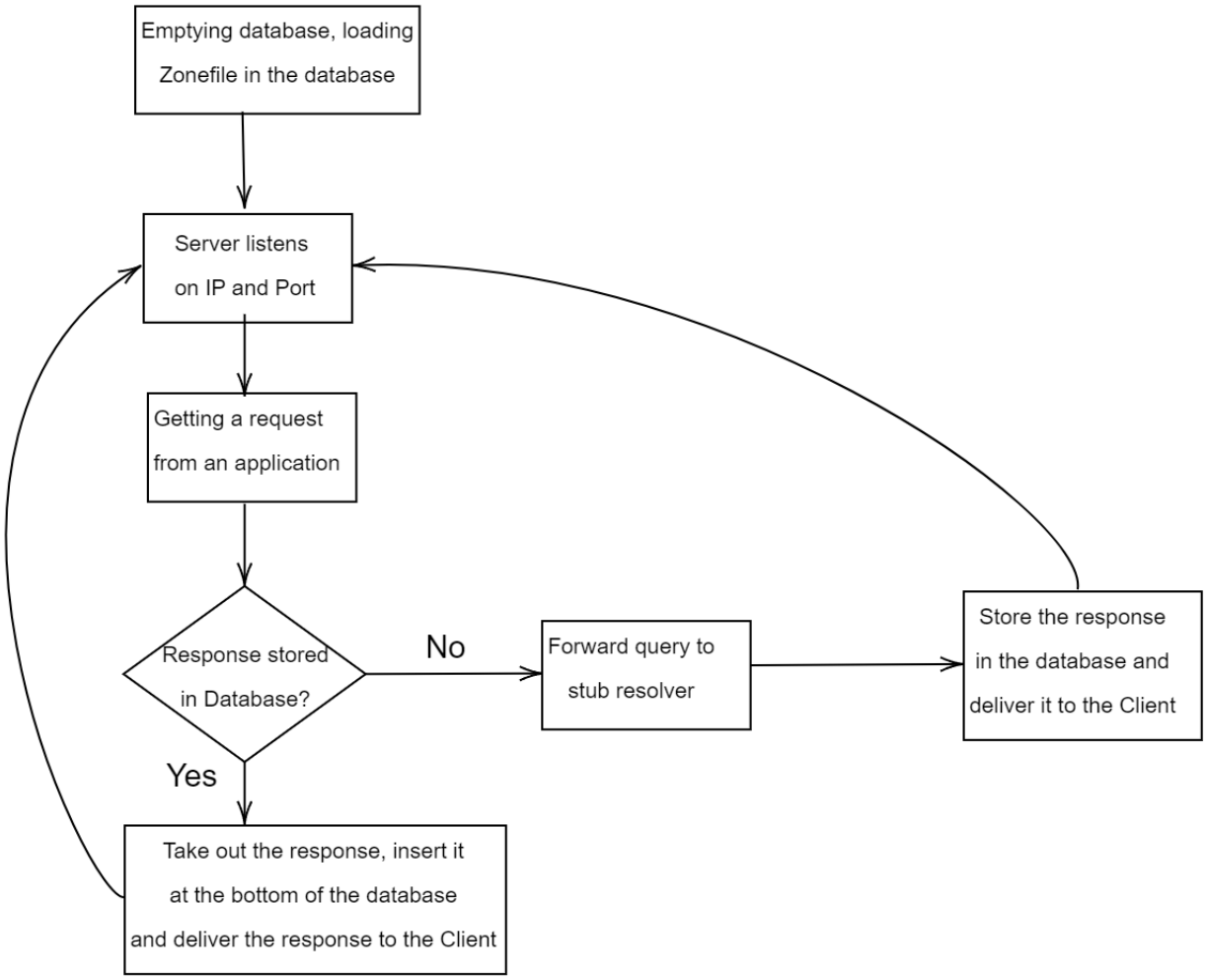
Documentation of the DNS Server

In this documentation is everything somebody needs to know about how the implemented DNS Server works.

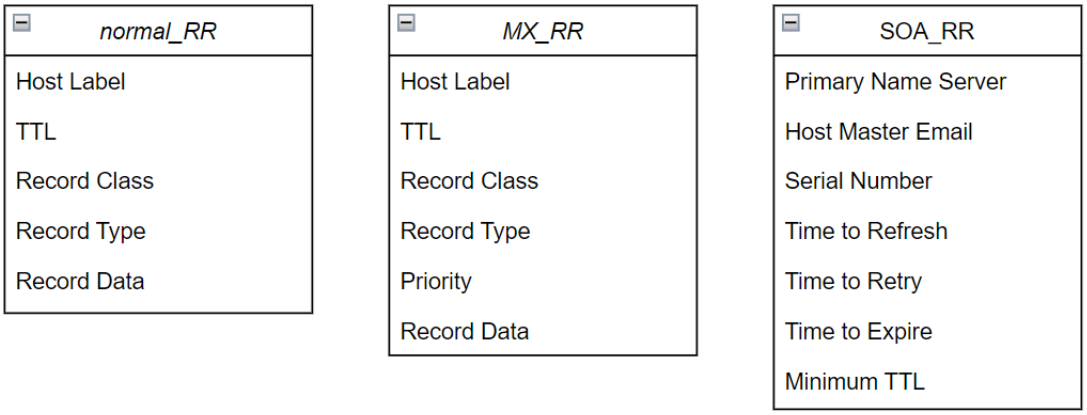
First, let's go over the superficial functioning of the DNS server. In the following flowchart it is simply described which steps are performed of the DNS server to understand the simple process.

Flow Chart of how the DNS Server works



In the first step the database is being created and emptied and the entries of the Zonefile are being loaded up into the database. Our database has different sections for different entries. The main difference is here at first that our Cache has an own section with many subsections and the zone of our DNS server has also its own section with subsections. Another meaningful difference is, that each entry has a type for which they are getting stored in the right database subsection table. These types are also called resource record types which declares what kind of data the entry contains. In the following database scheme we are able to see in which points they differ.

Database scheme of the RRs



Both sections, the Cache section and the Zone section have the same tables for their entries. In the normal_RR table there are stored more than one type of resource records because they all share the same characteristic. To this table the following RRs are belonging to: A, AAAA, CNAME, PTR, TXT, NS. The other 2 tables have other characteristics because of the difference of MX and SOA RRs, so they needed special tables for themselves.

After setting up the database, our DNS server will begin to listen for incoming packets. These are getting captured if an application sends to the defined IP address and Port of the DNS Server a DNS query.
After capturing a DNS query packet the first thing that happens is, checking if the answer of the captured query packet is already stored in our database. If that's the case, our DNS Server will apply the load-balancing method. What it does is basically to take out the most top entry and inserts it at the bottom of the table, so that not only the one and same entry will be delivered to a domain query which has more than one IP address. Because mostly, a website with a high occupancy has more than 1 IP address to distribute this occupancy on these for a better performance. And with our load-balancing method we are contributing to this. After the load-balancing method the response packet is getting delivered to the Client and our DNS server will wait for new queries.
In the case of that the entry is not in the database, the query will be forwarded to the stub resolver, which is in this case an other DNS server which provides us with entries if our database does not contain them itself. Our cache then stores the delivered response packet of the stub resolver and forwards it to the client back. And like before, after delivering the response packet to the client our server will start again listening for new incoming queries.

The advantage of having a cache and zone database is that these entries can be delivered much faster than forwarding the query packet to the stub resolver and waiting for an answer. It is also more reliable because it is possible that the stub resolver could be shut down and having entries stored in the database can still be used to reply to some queries.