Unit testing

Unit testing is the initial testing phase. It consists of testing the smallest components or modules that make up the whole system. If done correctly and early enough, the development team is able to single out where the issue occurs and easily, without troubles integrate more individual code modules. Unit tests are usually automatic and should be run before one starts working on a feature for example. If he does not, the time and resources used to look for the bug that potentially could be somewhere in the system could outweigh unit testing significantly.

Typically, a unit test is made of three phases - Arrange, Act, Assert (known as AAA). In the Arrange phase, we set-up the test (e.g. we prepare and initialize all the variables). Later, in the Act phase we do the actual test. Its result is then checked in the Assert phase where the test either returns true (passes) or false (fails).

To unit test our code, we used the commonly used NUnit framework. Overall, unit tests were a significant part of our source code. They were covering each of the model class and they made sure the methods used in the Data Access Layer (DAL) were working flawlessly. During the development process, it was often hard keeping the tests up to date due to various changes in the database or the model layer. It must be admitted we did not run these tests as often as we should have, especially towards the end of the later sprints, where the deadline was approaching fast.   
Even though we did not find the time to test the entire DAL during the development, we made sure to add the tests afterwards, so that someone else can pick up the code and easily work with it.

Communication protocols

Protocols are sets of rules that must be abided when two devices communicate between each other. They are required, if would not be for their specific format, machines could not communicate. They are incredibly useful because they can authenticate and correct messages or detect errors. With some protocols - for example, the HTTPS (Hypertext transfer Protocol Secure), they serve the purpose of securing systems.

There are several protocols we directly or indirectly used in our project. The TCP (Transmission control Protocol), IP (Internet Protocol) & forementioned HTTPS are all protocols at the core of the whole internet and it would be extremely hard, even impossible to make a modern distributed system without them.

TCP is one of the most popular protocols, which is used for communicating across a network. It takes the senders message, divides it into smaller parts (packets) and sends them across the network. The receiver then reassembles the packets together and gets the message. It also makes sure that the receiver got all the packets, in the same sequence, without any duplications or damages.

Internet Protocol is usually used together with TCP. Essentially, it is an addressing protocol, for determining a good route based on the IP address the individual packets carry. There are two versions of the IP protocol - IPv4 and IPv6.

HTTPS is used for communication on the Internet as we, its users, know it. One is using the client browser and the other is sending the data from the Web server. The data is, as the name suggests, in hypertext format. This part is the same as for HTTP, but the advantage of HTTPS is that it is more secure. It sends the data in an encrypted format, so anyone in the network does not see in plain text your HTTP requests/responses. HTTPS as well as other network protocols has a specific port on which is listens to responses from the network.

We could have also easily come across the FTP (File Transport Protocol), which is used to send files between a client and web server (or generally between machines), or the SMTP (Simple mail transport Protocol) if we had implemented email-confirmation for the users of our system.

(Unit Testing - What is Its Importance in Software Testing?, 2020)  
 (Types of Network Protocols and Their Uses, n.d.)