1.2 Anatomy of a Software Application

Now you should probably have a general idea of what software is and how it can help in our day-to-day lives. Let’s take a deeper look into the tiers you will learn about as a FullStack developer. We will look into what functionalities they have, and what technologies and tools that you’ll be working with.

Front End / Client

This is the layer most users will be directly interacting with. The front end can be developed in several ways. You can have a web-based front end, which is in the form of a website or a web application. If we take our bank as an example, this is what the banking assistant will be interfacing with to retrieve all the information necessary to provide you a service. This will mostly be a desktop application, which means it is a separate application installed in that desktop that enables them to connect to the back end. Desktop applications are normally developed using programming languages such as C++, C#, Java, etc. They are similar to a game that you install on your computer, and instead of fighting enemies, the banking assistant retrieves and updates information. The ATM you use to withdraw or deposit cash can be also seen as part of the front end.

Another example of a front-end application is the mobile or web application you use to check your balance or to make an online fund transfer. Unlike with the banking assistant, you can directly interact with the front end in this situation. To develop these kinds of applications there are a plethora of languages to choose from. For example, if it is a web page/application, it can be developed using HTML, CSS, JavaScript, Node, etc. If it is a mobile app, technologies such as Ionic, Swift, or Kotlin can be used.

Back End / Server

The back end or the server is the backbone of the system. It is responsible for handling and catering to all needs of the user. Even though the user interacts with the front end, the backend is where all the work happens. You can think of this as the kitchen of a restaurant. The front end is the waiters, whom you interact with and inform what you want to have. Then the waiter goes and informs the kitchen, which is analogous to the backend. This is where the chefs cut vegetables, cook them and arrange them on a plate with some decorations to be served. When the order is ready, the waiter will pick up the dish and deliver it to you.

This is a very powerful concept we use in software engineering, called abstraction. As the user, you do not need to know how your request is catered or what additional functionalities are needed to fulfill it. The back end takes care of all these tasks, just like the chefs. For example, from the banking system we discussed, imagine that you want to withdraw some money from your account. What you see is you enter the card, enter your PIN and withdraw the amount required. In order to perform these tasks, the backend should verify your identity, access a central database to check your balance, authorize the transaction and update your current balance. All these actions are performed by the server in abstraction. Programming languages such as

PHP, Ruby, Python, and Go are used to build such systems. This is also known as server-side programming.

Database

For the bank to provide you services, it should have access to information about you. For example, they would need to perform a security check on you to make sure you are the holder of a certain account. When you register with the bank you provide your details by filling out a form and this information gets stored on a database at the time of your account creation. A database can hold various kinds of data. It can hold string data such as your name and address, integer and float data such as date of birth, account balance and telephone number, and even image data such as your signature or photo.

There are several types of databases. The most commonly used database type is called relational databases, which you will learn more about them in future lessons. These types of databases comprise a set of tables. We call a single row of a table, a record. The backend will access the database and retrieve the relevant information using queries. A query is a command or a request to the database to retrieve a set of data points that satisfies a set of rules. Queries can also be used to update a database. One of the most popular database management systems (DBMS) is MySQL. There are also other DBMSs such as PostgreSQL, Cassandra, SQLite, and MongoDB. This was only a basic introduction to the technologies that are available for a full stack developer and there are many more to learn. We hope reading about all these interesting technologies got you excited, and hope to see you in the future lessons where we go into detail about these areas.