For our work topic we were assigned to research on web applications, so we opted to take a look on the types of existing web applications, the technologies behind them, and how using interaction works in comparison with regular web pages.

To start off, a web application (or web “app”) is a webpage (or even a group of webpages) accessed through the HTTP that allows the user to experience and application through the web. In order to do so there must me a remote web server acting as a cloud as well as local executable elements (however these applications do not need to be installed) and require an internet connection. Web applications can be ran in any device that supports web browsers and so in most cases the information accessible in the web app can be attainable through any device anywhere, which is a big advantage.

On the other hand, since connection to the internet is required, the information contained within web apps faces several security risks as the content can be exposed. It is up to the app developer to take proper measures to ensure the integrity of the data. Traditionally, web applications used mostly the keyboard and the mouse as input sources, however, with the development of more advanced recent technology, the most predominant input source nowadays is the touch screen, which has also encouraged the use of other file and media formats.

Much like any other applications, the back-end of Web applications can be written in varied programming languages such as C++, C, Python etc.

In the front-end, nowadays, the technologies have evolved to make it so the user doesn’t need to keep waiting for the information. “The classic web application model works like this: Most user actions in the interface trigger an HTTP request back to a web server. The server does some processing — retrieving data, crunching numbers, talking to various legacy systems — and then returns an HTML page to the client.” (Garrett, 2005) But, according to Helder António Almeida, “Nowadays, traditional web client’s have become obsolete and there is a concern into making these into *Rich Internet Applications*” (António, Brandão, Doutora, Ramada, & Pimenta, 2008)

So, previously, Web applications would depend on the server to do most of the work for them. Nowadays there are concepts like Ajax that make this problem go away with the use of an Engine that will process basic data to minimize server dependency. Software like JavaScript, HTML and CSS help in this regard.

The web content is increasingly generated dynamically, contrary to what was verified in the early days of the Web when virtually all content consisted of static HTML or image files.

Nowadays, using the Web as execution platform, we come across a huge variety of web products, from simple collections of static HTML pages to full distributed applications.

Web applications contain pages with partially or entirely indeterminate content and the final content is determined only when the visitor requires a page from the web server. Due to the change of content of a request according to the actions of the user these pages are called dynamics. However a simple page containing just some text and a visitor counter or the current date can be considered a web application.

Any traditional desktop application, such as a calculator, that has a web interface is a web application, as well as sites that minimum degree of behavior, logic and state.

For a web application to have a favorable user experience it must have a level of interaction that is both fast and efficient so the user can use it as smoothly as possible with minimum down times.

Web developers try to make an intuitive experience with their apps through Technologies like AJAX, that function like in-between engine between the web server and the user interface of the web application. This way, a web application can be a fully interactive experience instead of being just a traditional use of HTML which only features static pages, with simplified ways of interaction like simple text boxes and links, that are limited by the request-response paradigm, which needs a refresh of the page’s contents to give feedback to the user.

With AJAX the request-response process was made invisible, as well as updating the contents of a page in real time, with no need of reloads.