**4.1 Angular design.**

In Angular applications, modules are the basic building block. Each Angular application has a root module, named AppModule and provides the bootstrap mechanism that launches the application (Angular.io, 2020). Within the AppModule it contains child modules that are hierarchal in nature. While a small application many only contain the one module, larger applications, like the one developed, have a number of feature modules. A module groups together components, directives, pipes and services that are related so that they can be combined with other modules to create an application.

As well as having at least one module, every Angular application has at least one component. In the application this is AppComponent which is the root component and connects a component hierarchy with the page object model, or DOM. Each component defines a class that contains application data and logic and is associated with an HTML template that defines a view to be displayed (Angular.io, 2020). Shared functionality between components are written into services. Services are injected into components and a component can delegate tasks, such as fetching data from the server or validating user input to the services.

The organising of code into distinct functional modules with components and services helps to develop complex applications with reusable code, allowing applications are lean and efficient (Angular.io, 2020). This modularity can be seen throughout the designed application for example, the *profile module* uses a number of components such as *user-information* and *user-shifts* which each have their own class that contains the logic for that component and their own HTML template. These individual components are then combined together to give the *profile* page in the application. Furthermore, the majority of components make use of the created shared services. For example the *user-welcome-component* use the *getEmployeeDetails()* method that contains a HTTP GET request inside the *employee.service* to gather user information from the database and display it in the relative HTML template. A sequence diagram found at **FIGURE NUMBER** shows this process in more detail.