Working at the King Faisal Specialist Hospital and Research Centre gave me the opportunity to work on big systems, developing big applications for a large facility. The hospital is considered one of the biggest hospitals in the Middle East, with a capacity for over 1,200 beds, 20,000 patients, and 11,000 employees. Additionally, the hospital is recognized as a national referral center for organ transplantation, genetic diseases, cardiac surgery, and other specialties. During my time at the hospital I worked on many projects for different platforms, which I believe have prepared me well enough from the software engineering technical perspective.

The biggest project that I worked on in the hospital was Sehaty, or “My Health,” which provides the entire health record for the patient, and the data can be viewed in either English or Arabic. The application includes various aspects of patient care such as appointments, immunization records, lab tests, radiology screenings, vital signs, and more. Sehaty also provides the patient with the ability to cancel or postpone appointments, review statements of visits, and many more actions. It is easy for the patients to switch between current medical records or prior records to which they have been granted access.

Sehaty is divided into three parts: the back end, web application, and mobile application. The back-end section connects to the hospital’s medical system (Cerner), which is then provide all the needed data to the patients via the web or mobile applications. The back-end section also performs all received actions from the patient through the web or mobile applications. The web application is built using JavaServer Faces (JSF) technology and allows patients to view records via a web-based connection. While I was involved in the development of all aspects of Sehaty, I primarily focused on the mobile application. The mobile application, currently offered only for iOS systems, is built with Objective-C and mirrors features of the web application, but it also offers mobile-specific functions such as accessing the calendar and notify the patient through the device notification. Patients can download the mobile application from the app store; the application had over 4,000 downloads in its first month of availability. [Link]

Another application that is integrated within Sehaty is Lifestyle, a health tracking application consisting of two parts, the client and the back end. I was the main developer for both the client and the back end. The client is the source of the data, which is implemented in Sehaty's iOS application and reads all the patient records from Apple's Health app for the iPhone. It then sends records to the back end for storage, this process is completed manually based on patient action or by the background fetching feature if it enabled by the patient. Back end for lifestyle is set up behind Sehaty's back end, so any communication for this feature has to go through Sehaty's back end. Because it uses the same authentication as Sehaty, this feature improves patient security, integrity, and the user experience. Lifestyle's back end handles most of the logic, as well as what to insert, update, retrieve, and delete. It also communicates with Sehaty's back end through a Simple Object Access Protocol (SOAP) web service.

In addition to the applications for Sehaty, I have worked on various JSF applications. These include Default Location, Supply Chain Customer Service, and Supplies Scan. Default Location is an application concerned with management of the location of PCs connected to the hospital's medical system (Cerner). It narrows visible patients to current locations of a certain PC, in addition to many functions such as managing a specific PC or managing all PCs for a certain location. The application also has in-app Access Control List (ACL), which allows the administrators to provide access and privileges to various users. Additionally, the application has an auditing feature that records all actions taking place inside the application itself.

With no interaction from the Supply Chain’s agent, the Supply Chain Customer Service application presents data from multiple sources based on the customer selection. The application works by receiving a request number through a query string passed by the customer service application before it presents a view with all the required data in it. If the request number was wrong or has no data, the application presents an option for the agent to enter the request number manually.

Another JSF application I worked with was Supplies Scan. Supplies Scan is an extension for the medical store application, which handles unregistered items by connecting the item's barcode with an item code in the Enterprise Resource Planning (ERP) system and then adds more details about this item. The application provides users with a view to scan the product, and then the application checks to see if the item exists. The application also provides users with a view to manage all scanned and registered items.

Finally, I also worked on development of MyShare Drive, an application that has been installed in over 1,000 PCs. It is a collection of Batch, VB Script, and registry files packaged inside of an .exe file that installs, configures, maintains, and monitors a WebDAV client (WebDrive) to connect to Alfresco (ECM). Different events trigger the script for this application, the first of which occurs upon installation where the script configures the client for the first time using login credentials through the Single Sign-On (SSO) feature from Alfresco (if applicable) or through basic authentication. In addition to the configuration, this event also sets up other events, including the startup event and the schedule event through the Task Scheduler. Both of these events check the integrity of the connection, and if they find an error, they run a fixing script to repair the connection. If failed, the script flags the connection and send an email to notify administrators with details and the error code. The startup script also attempts to fix a flagged connection, as the cause may be resolved after a system reboot.

In conclusion, along with my technical experience in many areas of software development. I would like to support my experience with a scientific knowledge. In order to improve myself, my skills, my future projects, and to reach my goal of becoming a successful Software Architect.