Introduction

The objective of this project was to create a software application with a navigational menu and four distinct use cases.  The application was to be designed according to modern practices, employing Agile Methods, Scrum, and Unified Modeling Language (UML) diagrams to aid the refine the development of the project.  A conceptual foundation for the application was established through user stories, scenarios, a product backlog, and use case diagrams.  Class diagrams and class, responsibility, collaborator (CRC) cards helped construct a framework for the data within the app.  The build of the app was further matured through sequence, state, activity, and component/deployment diagrams.  The application itself was built according to Model-View-Controller (MVC) architecture in a series of sprints with each sprint seeing the stand-alone completion of a use case.

We chose to build a Pool Access System (PAS) as our project application.   The app was envisioned as a tool that swimming pool staff could use to manage and monitor access to a pool electronically.  The fundamental purpose of the app is to allow the creation of pool members in a system that can check them in and out.  The system can also retain pertinent data on swimmers (allergies, weak swimmers, etc.) to improve pool safety.  In addition, the app allows management to draw meaningful statistics from data on pool visitation.  The app is also intended to provide access privileges to higher management so that sensitive data is maintained by appropriate personnel.

To accomplish this, the PAS app utilizes four use cases: Manage Staff, Find Swimmer, View Pool, View Reports.  Manage staff supports the creation, deletion, and modification of pool staff registered to the system by those with management privileges.  Find Swimmer enables staff to search for a swimmer within the system to check the swimmer in or view a swimmer profile in detail.  Management may register swimmers new to the pool as well.  View Pool supplies a running list of those currently checked into the pool so that staff may easily locate swimmers leaving the pool for easy check-out.  Finally, View Reports offers a tool for management to calculate statistic based off pool usage within a specified period.  Ideally, management will be able to identify peak and low periods so that they might know how to better allocate personnel and advertise to swimmers.

The group felt the progression of the project through the semester was very fluid and successful.  One of the key challenges of creating the application was to place limitations on what the app should achieve in a way that was reasonable given the timeline of semester and the constraint of four use cases.  As we developed the app through course, we continued to debate the priorities of the app and adjust the application as necessary.  Ultimately, there were several features we chose to leave out even though they may have added value to swimmers and staff in an electronic pool system (guest passes, profile flags, historical storage of comments/flags, for example).  Deciding which features were most essential fundamental proved to be an on-going trial

Another challenge the group faced was to complete all the collaborative project deliverables by their deadlines with the added difficulty of working remotely.  Substantial time zone differences also came into play as one of the teammates resided in Russia while the other three were US-based.  To this end, we used WhatsApp, a free text message application to communicate amongst one another.  The group completed assignments as early on as possible and strategically divided the work load between members.  Communicating openly and often with one another mitigated the constraints of working remotely.  With solid communication, having a third person in a separate time zone was beneficial as it enabled the group to work on assignments 24 hours a day.

One final challenge of the project was developing code asynchronously between team members in a way that would not override the work of others.  We used GitHub to pull and push modifications we made to the app and made an effort to communicate which classes or user interfaces were being worked on by which members. There was a small learning curve in understanding how to commit and merge branches.