A Project Retrospective

1. What Went Right
   1. MySQL database was something we were all familiar with using so there was no learning curve
   2. Using PHP to access the database made the design very simple
   3. Our original plan was thorough, and we debated potential issues at length which helped us avoid extra bugs
   4. Using an agile lifecycle helped us be flexible to the uncertain situation we all found ourselves in due to the COVID-19 Pandemic
2. What Went Wrong
   1. COVID-19 Pandemic interrupted our plans significantly as we had to make the transition to fully remote development
   2. We started designing the schedule generation algorithm too late in the development process
   3. We kept focusing on details of issues that didn’t need to be a priority rather than solving bigger picture problems
   4. While an agile lifecycle was very helpful for our development process, we were slow to adopt it
3. Project Timeline
   1. We decided on an agile lifecycle with roughly two-week sprints and we believe this was a good and balanced choice
   2. Before we began our official scrum sprints, we took a week to try to map out a basic design plan and get in touch with our stakeholders
   3. After we felt we had a good place to start, we began our sprints
   4. Sprint 1: We accomplished most of the GUI design and got positive feedback from our stakeholders
   5. Sprint 2: We implemented some basic features of the GUI and the administration login on the front end and started designing the full database structure on the back end
   6. Sprint 3: This sprint happened to be during spring break, but we made a big breakthrough with the course catalog uploader, so we didn’t fall too far behind
   7. Sprint 4: Once Lindenwood University went to a fully online status, things slowed down quite a bit and we had to rethink our design to optimize it for mobile platforms
   8. Sprint 5: This is where we finally designed and implemented our schedule generation algorithm, which was successful but ultimately far too late in the development process
   9. We used all our slippage time to the full extent and still had some story points left over due to a lack of time and, once again, the COVID-19 Pandemic
   10. Finally, we delivered the application remotely

A screenshot of a social media post

Description automatically generated

A picture containing building

Description automatically generated

1. What We Would Have Done Differently
   1. We think our risk assessment was fair but in hindsight we needed to attempt a less ambitious design in order to complete the application in the time we were given
   2. Communication was key for completing this project but after moving to a fully remote development process we were less consistent with our group discussions
   3. As stated earlier, we should have designed (but not necessarily implemented) the schedule generation algorithm much sooner