4/27/2022

Team 1

**Process Report**

ScavengeRUs is a hosting solution that allows third party organizations to create a customized scavenger hunt for a targeted audience. Our team and two other teams were assigned tier 1 out of 7 for this semester. Tier 1 is a standard hunt for QR codes centered on a certain geographic place. Each hunt participant receives an email with an access code and a link to the hunt's homepage. Because it is browser-based, players may use whatever internet-connected device they desire. A task can be completed by scanning a QR Code. The player is given a predetermined list of tasks, each with a completion indication next to it. Players are allowed to change their display name, which is set to a random number by default. The access code tracks their progress on the server, so they may utilize more than one device during the hunt. When the hunt is over, the player receives a completion email. The ScavengeRUs project was divided up into four iterations which all consisted of team-based activates and a presentation. Iteration 0 was the planning portion. Iteration 1 was an opportunity for teams to learn to work together as a group through the ScavengeRUs project. Iteration 2 was the application design and the review of the code. Iteration 3 is the last iteration, and it is the delivery of the final product of tier 1.

Iteration 0 our team made the product identification and definition document (PID), the use cases, the Trello board, did the teams time tracking, and time meetups. For iteration 1 our team completed the QR Scanner and started the bootstrap. The QR scanner class uses a library called ZXing. One of our team members implemented the scanner by researching the ZXing barcode reader and coding it in visual studios. The biggest challenge for this iteration 1 was that a lot more time and research was put in because of how new the project is which led to small amounts of progress. The biggest struggle was the server being implemented because AWS was not working the way our team thought it would. For iteration 2, our team decided to step away from AWS servers after learning that AWS did not support .NET 6. In turn, our team decided to use Azure. It was simpler than AWS because it was easier to understand and had simple documentation to help deploy an ASP.net project. Once the server was deployed, the bootstrap was also implemented to the created server. The team struggled with database implementation and timing due to other classes and assignments around Spring Break. For iteration 3, the database was successfully implemented and classes were made for database population of entities. Bootstrap was completed and navigation bars were added to allow quick and easy navigation throughout the entire website. During this iteration, there were no issues except that more time would allowed us to manipulate the database and actually create tables and populate them through the website GUI. Our team worked hard and diligently together on the project. We could not implement everything we wanted, however, everyone did the best they could for the time the team members had.