Challenges and Lessons learned:

- **Serializability**:
  - Serializability has been useful to send data to and from our database storage but we ran into an issue whereby serializability is recursive and all subclasses/objects of a class must also be serializable in order to send the object back and forth. We learned about serializable objects in Java and the required contents of serializable object output streams

- **Databases**:
  - We ran into a challenge where the database code would not compile due to a missing package in the code which required a significant amount of debugging research due to the lack of error messages

- **Code Design**:
  - A challenge we faced was inherent within our chosen game design of storing the game state in a HashMap. The map is setup as <Location, Piece> and initially the only key values that exist in the map are the 12 playing pieces at the start of the game. A new map value of <Location, Piece> was added into the map upon the first instance of a piece moving to that specific location, and the value was set to null after the piece has moved. Some of our playing piece classes did not check for map key values not existing in the Map which led to sporadic and unpredictable bugs in our code based entirely upon the specific game board state. We learned the value of tests and debuggers

- **Time Conflicts:**
  - Our team faced challenges of being able to all meet at the same time to work on the project, we had to communicate remotely and work on compartmentalized code in order to make everything work together

- **GUI Programming**:
  - We learned how to use a Java CardContainer to create a menu system that allows for quick transitions between pages.
  - We learned that taking time to design the flow of the user interface can help identify places where the system needs to be modified or refactored.
  - We learned how to create an event handling system to allow the GUI to respond to events on the server.

- **Migration from single instance to distributed system**
  - Only one of our team members had ever worked with a distributed system so it was hard to work together to find bugs that went along with it while trying to implement it. It was very tedious to do overall, but not too challenging. We

learned more about output streams and when to use different types. For instance we were trying to use file output streams at first, when we actually should have been using byte array output streams.

- **Code compartmentalization and debugging**
  - Unable to effectively debug/work on code with compartmentalization working on different code sections