**RISK:**

* **Application Design Out Of Scope:**
  + If our design is out scope for our allotted time then we would have to redesign our project that is more manageable for this class. This risk will be minimized by the time we allotted for testing, and also the time we set aside for implementation is long enough for a redesign if needed.
* **Application Feature Failure**:
  + If we want to implement a feature of the original design of the application but are having trouble implementing that feature in the time we have, then we could just not implement that feature. This risk is caused by our general lack of experience with working at such an in depth level in app design, so as we learn more about it we can realize what is feasible in the time we have for this application. This risk is minimized by our long amount of time for implementing the applications code and features.
* **Collaboration Of Application Feature Failure:**
  + After we design and code the features, database, and server for this application, there is a risk that combining all these facets could cause a failure. This risk is applicable since this will be the first time our group implements an application like this. This risk is minimized by our long testing period.
* **Database Failure**:
  + If there are problems with strictly our database implementation we could seek help from one of the many qualified professors to help us with this, or also we could ask our TA for this class. This risk is minimized by the long period of time allotted for database implementation and also the long testing period that is planned to be at the end of the implementation period.
* **Server Failure**:
  + Similar to a database failure, a shortcoming with the server can be prevented with seeking additional help by either our TA or a professor who could be of assistance. This risk is minimized by the long period of time allotted to implement the server and also the long period of time allotted for testing.
* **Demo Failure**:
  + The possibility of our demo going poorly can easily be avoided by pre-recording our demo. This can only be done if we plan to have time to record the video, which means finishing the project earlier and recording the presentation we would have to practice beforehand while testing.

**V Model:**

1. **Requirement Analysis:**
   1. Decide what the functionality of the project and what we want our product to do.
2. **System Design:**
   1. Decide how to best to implement the features we want and draw up what our product will do and how it will do it.
3. **Program Design:**
   1. Decide how to implement and code the project, including classes and how to implement each feature.
4. **Code:**
   1. Implement the design that was previously drawn up.
5. **Unit & Integration Testing:**
   1. During and after coding each feature test it concurrently with the rest of the system.
6. **System Testing:**
   1. After testing each feature by itself and with the system, ensure the design follows the plan.
7. **Acceptance Testing:**
   1. Test finished project with what our original product design would be, trying to bridge any gaps in the two designs.
8. **Operation & Maintenance**
   1. Collaborate and test product under stressful conditions and maintain operation through testing.