|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Risk Statement | Response strategy | Objectives | Likelihood | Impact | Risk Level |
| GitHub | Any source code pushed to GitHub could potentially contain information that hackers would find useful when trying to a maliciously alter the project. The source files could potentially contain hard-coded login credentials which could allow for data leaks. | Use stronger passwords and usernames than just “admin” or “root” and keep them regularly updated. | Reduce the likelihood of hacking and data leaks. | Medium | High | Medium-to-High |
| Java code credentials | Hard coded credentials and passwords could be stolen if the code gets accessed by others. | All the credentials and passwords should be inputted by the user rather than being hard coded. | Reduce the risk of hacking and data leaks. | High | High | High |
| Bad quality of code | Bad quality of code brought in by bad practices could cause additional errors within the system while also making it harder to understand which would slow down the progress of work being done. | Make sure that the code is well commented explaining its functionality and utilise design principles such as SOLID to avoid bad code practices. | Reduce the complexity of code and make it more manageable to work with. | Medium | Low | Low-to-Medium |
| Time management | Mismanaged time could impact the quality of code written. | Make sure to not spend too much time on documentation and leave enough time for writing the code and testing it. | Improve time management. | Medium | Low | Low-to-Medium |
| Ambitious planning | Planning too many features at the start of the project could lead to not having enough time to make the system work. | Follow the minimum viable product requirements until they are met, only after additional features should be made. | Make sure that the project is finished on time. | Medium | Medium | Medium |
| Hardware | Hardware failure could lead to loss of work that has been done. | Make sure to backup and upload new features of the project on GitHub frequently. | Reduce the risk for loss of data. | Low | Medium | Low-to-Medium |
| Scope variation | Adding too many new tasks during the development of the project could shift the scope so much that the time required for finishing the project goes beyond the planned estimation. | Stick to the initial plan created for the project and try to minimise additional changes in the middle of the project at least until minimum viable product has been created. | Reduce the risk of running out of time. | Low | Low | Low |

**Risk Matrix**

**Potential Risks**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | GitHub | Java code credentials | Bad quality of code | Time management | Ambitious planning | Hardware | Scope variation |  |
| **Likelihood** |  | Severe | Major | Minor | Minor | Minor | Moderate | Not Significant |  |
| Should occur under normal circumstances | **Certain** | High | High | Medium | Medium | Medium | High | Medium |  |
| May occur sometimes. | **Likely** | High | High | Medium | Low | Medium | High | Low |  |
| Could occur. | **Possible** | High | Medium | Low | Low | Low | Medium | Low |  |
| Possible but shouldn’t occur under normal circumstances. | **Unlikely** | Medium | Low | Low | Low | Low | Medium | Low |  |
| Is possible but extremely unlikely to occur. | **Rare** | Low | Low | Low | Low | Low | Low | Low |  |