|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Description | Evaluation | Likelihood | Impact | Responsibility | Response | Control Measure |
| SQL Injection | User could drop tables, schema, or cause other issues. | Low | High | Developer | Identify block of code causing issue via code analysis tools and fix it. | Use SonarQube to identify SQL injection risks before committing any code to the repository. |
| Missing non-nullable fields | User could create a ticket without filling in all the mandatory fields. | High | Low | Developer | Identify missing information through error in console log and add user-interface restrictions for non-nullable fields or add appropriate error handling code. | Coordinate the front-end code and back-end code releases so that error handling code (front-end and back-end) are in sync. |
| Deleting completed tickets | User could delete a ticket after the technician has worked on it and added a solution. | High | Medium | Developer/User | Identify cases where preserving history is vital and add update/delete user interface restrictions for such cases. | Limiting user functions. |
| Corrupted production database | The back-end server could crash if the production database is not healthy. | Low | High | Developer/AWS | Inform the right people.  Try to recover as much data as possible. | Follow AWS best practices and setup database backups in case of issues. |
| Back-end server crashes in production | The back-end server could crash in production due to unhandled exceptions. | Low | High | Developer | Where possible use a process monitor to bounce the back-end server if the server goes down. | Follow AWS best practices when setting up the back-end server.  Exhaustive testing in development environment. |
| Incompatible releases | The front-end and back-end releases could be mismatched due to continuous delivery pipeline. | Med | Med | Dev Ops | Rollback to a compatible version or coordinate front-end and back-end releases so that happen at the same time. | Use the industry standard of only pushing code to the main branch on the days of lowest traffic (e.g. Sundays).  Push front-end code and back-end code to the main branch at the same time if changes happen on both sides. |