**Risk Analysis**

**High Chance- \_\_\_**

**Medium Chance- \_\_\_**

**Low Chance- \_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **Hazard** | **risk** | **Safety measure** | **Alternative** |
| University server going down. | Losing current progress/ reduction in productivity. | Constant use of GitHub, have alternative work to do in case. | While servers down try and do other aspects of system. |
| Input errors from user. | Don’t want Customers inputting information that may damage the system. | Triggers and constraints in database to reduce risk. | Use of dropdown and rollers to reduce user error. |
| Numerous commits at the same time. | Overlapping code where code is edited in some versions but not in others so version control messy. | Use of git pull to get the most up to date version of the system before committing files. | Work on same computer for certain aspects (API). |
| Member leaving. | Work load increased for remaining group. | Use of sprints and clear targets being reached to show improvement and predictability for the future sprints. | Frequent meetings to see progress. |
| Security Breach. | Risk in sensitive information being accessible through system flaws. | -Hashing passwords  - (salting passwords)  -API tokens |  |
| API key usage | Risk in API keys being readily accessible to skilled users | Keeping the API keys out of our visible code |  |
|  |  | Restrict accessibility from other functionality by role |  |