**RISK MANAGEMENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RISK** | **PROBABILITY** | **EFFECTS** | **AFFECTS** | **STRATEGY CATEGORY** | **STRATEGY TO BE EMPLOYED** |
| The software may crash due to information overload | High | Serious | Product | Underestimated workload | Ensure system can handle large amounts of information at once |
| The software may not be accepted by targeted population | Moderate | Catastrophic | Project and business | The software is underwhelming | Take customer feedback to improve overall system before launching |
| A competitive product may be implemented before | Moderate | Tolerable | Business | Choosing the correct market and audience | Improve marketing skills and provide what the audience needs |
| The database used may not process as many information as expected | Moderate | Tolerable | Product and project | Database performance | Investigate in buying a higher performance database that can handle extreme amounts of data |
| The scale of the project workload is underestimated | High | Tolerable | Project and product | Underestimated development | Investigate buying-in components |
| The timetable generated by code is not feasible to implement | Moderate | Serious | Product | Patching of software | Generate code that produces what the user is most comfortable using |
| The size of the system may be underestimated | high | Tolerable | Product | Underestimated size of software | Manage the components being used in order to maximize the space being used |
| The time required to develop the software may be underestimated | High | Serious | Product and project | Underestimated development time | Investigate the use of a program generator or buying-in components |