**1. Risk Analysis:**

*1.1 Background*

Risk identification is the first stage of risk management process. It looks to identify potential sources of risk along with the probability of these risks occurring. The second stage is to calculate the impact of these risks. The third and final stage is the remedial action as to how to avoid this risk from happening.

These pre-defined risk types provide a good base that helps to identify the risk and ensures that a certain process is followed in order to deal with the mentioned risks.

After identifying and categorizing a risk, it is documented.

1. *Staff Risks*

Risks that are related to the members of the Group involved in the project.

1. *Technical Risks*

Risks that are related to the software and hardware aspect/limitations of the group project.

1. *Accessibility Risks*

Risks that are linked to the availability of the code for the various group members.

1. *Organizational Risks*

Risks that emerge from the organizational environment including the commerce side of things where the software is made and where its used.

1. *Estimation Risks*

Risks that are derived from management of time, i.e., the parts of the project and the project itself to be completed on time.

1. *Security Risks*

Risks that are related to the security of the code and the threats of it being attacked by viruses, along with the theft of the software/code.

*1.2. Documentation:*

Low Priority

Medium Priority

High Priority

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk ID | Risk  Type | Risk | Probability | Impact | Remedial Action |
| RID1 | Staff | Lack of group arrangement | Moderate | Tolerable | Any meetings should be scheduled in advance AND with every group member’s say taken in account. Roles and tasks should be clear to anyone who has been assigned one. |
| RID2 | Technical | Code unclear to some | High | Tolerable | Any piece of code written down should have a reasonable amount of comments written to avoid confusion from within the group members. |
| RID3 | Staff | Poor communication | High | Serious | Group members should be notified whenever there is any change in plans or objectives regarding the project. |
| RID4 | Technical | Unexpected outcomes and errors | High | Serious | Functions should be implemented and tested in singles to allow for easier tracking of error outcomes. |
| RID5 | Technical | Software limitations | Moderate | Serious | Code should be written in a way such that it allows for future expansion for extra features. |
| RID6 | Accessibility | Lack of accessibility | Low | Tolerable | Anything involving this project should be posted in a place accessible and possibly updateable by all group members of the group, possibly GitHub. |
| RID7 | Technical | Loss of content/information | Moderate | Tolerable | Keep all files organized and double-check before deleting work. |
| RID8 | Technical | Most of functional and non-functional requirements are not met. | Low | Serious | Team members can keep track of all the requirements as they go ahead with the project. One of the members can create a checklist of all the requirements that have been met. |
| RID9 | Staff | Inconsistent costumers | Moderate | Serious | Have regular meetings with the customer, thereby we get a clear assurance of what is to be implemented.  Providing them with prototype and progress at every stage of the project. |
| RID10 | Organizational | Financial problems | Moderate | Serious | Project costing should be done realistically and cleverly. Once the budget is fixed, it should not be exceeded. |
| RID11 | Staff/  Technical | Team members may not have sufficient knowledge to do certain aspects of the project. | High | Serious | Assign tasks to the members based on their strengths and team members who are familiar with the topics should help the rest overcome obstacles. |
| RID12 | Technical | Server failure | Moderate | Serious | If the server fails for any reason whatsoever, there is a high risk of losing valuable data. We can avoid that by using incremental data backups, which work on backing up data only when data is added or modified and adds that to the previous backup. |
| RID13 | Technical | Ill-fitted photos posted | Moderate | Serious | Constant monitoring to make sure any unsuitable photos do not exist on the feed, consequently giving out warnings and possibly temporary account limitations to users who belong to that criteria. |
| RID14 | Estimation | The project targets not met by a certain time | Moderate | Tolerable | We should set realistic timings as for when goals are to be met. We should observe our previous behavior to learn from any mistakes or bad decisions made, and possibly prioritize our timings better for more future productivity relative to the project. |
| RID15 | Security | DDOS attacks | Low | Serious | DDOS attacks can’t be predicted, but we can reduce the chance of it happening by having someone monitor the firewall for any suspicious activity. |
| RID16 | Staff | Team lacks motivation | Moderate | Tolerable | The team members should constantly motivate each other. |
| R1D17 | Organizational | Illegal documentation | Low | Serious | Team members should access only legal documents. |
| R1D18 | Technical | User experience is poor and User interface is not aesthetically pleasing. | Moderate | Tolerable | Assign UI/UX to the designers in the group. Polls may be put up to see which interface Staff prefer more. |
| R1D19 | Organizational | Unable to raise extra capital if cost exceeds budget. | Low | Serious | The group members must try to raise money for the project via various sources such as private/government banks, individual asset lenders. |
| R1D20 | Organizational | Copyright/Patent issues. | Low | Moderate | Sometimes when the final product is done and dusted, there are issues with the copyright/patent where someone has already made a similar product, or the name of our project is in conflict with others projects. |
| R1D22 | Staff | Team members can have problems with their workload | High | Moderate | Group members should let others know if they are having problems with their workloads as this can cause failure to produce work. Hence work needs to be split amongst members equally for productivity. |

1. **Risk Planning:**

2.1 Background

  Once the risks have been identified, we must find the appropriate risk management methods.

2.2 Risk Strategies

  There are three types of risk strategies we have to follow, and they are: -

1. Avoidance:

The avoidance strategy reduces the probability of occurrence of the risks.

2. Contingence:

The contingency strategy is assuming the worst case and developing a plan to deal with it.

3. Minimization:

The minimization strategy is intended to reduce the impact of the risks.

* 1. Documentation

**Risk (RID1): Lack of group arrangement**

**1. Avoidance:**Any meetings should be scheduled in advance AND with every group member’s say taken in account. Roles and tasks should be clear to anyone who has been assigned one.

**2. Contingence:**In case the group lacks total order and organization, a meeting should be conducted for the sole purpose of discussing this matter, to try to work around this obstacle.

**3. Minimization:**Group members should always update each other on edits they add to the project. Communication should be constant.

**Risk (RID2): Code unclear to some**

**1. Avoidance:**Any piece of code written down should have a reasonable amount of comments written to avoid confusion from within the group members.

**2. Contingence:**In case code is vague to group member(s), a meeting should be conducted to make sure all coding doubts are cleared and fully explained.

**3. Minimization:**Code comments should be updated frequently, especially when new code is added.

**Risk (RID3): Poor communication**

**1. Avoidance:**Group members should be notified whenever there is any change in plans or objectives regarding the project.

**2. Contingence:**Nothing too complicated here, if communication is weak, a meeting should scheduled to discuss the importance of communication in this project and how strongly it affects it.

**3. Minimization:**Meetings should be routinely scheduled.

**Risk (RID4): Unexpected outcomes and errors**

**1. Avoidance:**Functions should be implemented and tested in singles to allow for easier tracking of error outcomes.

**2. Contingence:**In case we encounter errors deeper into the app, we should trace the error in order to know its exact source. However, we hope that this is not the case in the future.

**3. Minimization:**Make sure every function works as intended and testing it

immediately after it is done.

**Risk (RID5): Software limitations**

**1. Avoidance:**Code should be written in a way such that it allows for future expansion for extra features.

**2. Contingence:**In case we encounter a wall midst coding, we should try our best to modify certain aspects of our whole code to allow for more functions to work.

**3. Minimization:**Always code in a way that allows for countless features to be implemented in the future.

**Risk (RID6): Lack of accessibility**

**1. Avoidance:**Anything involving this project should be posted in a place accessible and possibly updateable by all group members of the group, possibly GitHub.

**2. Contingence:**Any member that encounters problems when trying to access the project files should immediately report to the group so that he/she can access them.

**3. Minimization:**Always push commits for any changes or modifications made to the project files and keep it updated.

**Risk (RID7): Loss of content/information**

**1. Avoidance:**

Team members must keep all files organized and double-check before deleting work.

**2. Contingence:**

In case of loss of work, the team members must meet up, and the person with the most dated version of the project must provide the rest with all the content and then discuss what must be redone. If a significant portion of the work is lost, the liaison must contact the line manager.

**3. Minimization:**

To minimize the impact of the risk, the team members are asked to save their work frequently and upload it to GitHub.

**Risk (RID8): Most of functional and non-functional** **requirements are not met.**

**1. Avoidance:**

Team members can keep track of all the requirements as they go ahead with the project. One of the members can create a checklist of all the requirements that have been met.

**2. Contingence:**

Significant deviations from the requirements require the team members to schedule a meeting with the customer and make them aware of the significant deviations from the requirements. The customer will lay down the necessary steps that are required to solve the problem. Team members should make sure they stick to the new requirements set by the customer.

**3. Minimization:**

To minimize the impact of this risk, the team members must set up a weekly meeting to discuss all the requirements they were able to meet and the ones that were not possible. If any member faces difficulty in achieving a said requirement, the work can be delegated to someone else or split between a few members to ease the load.

**Risk (RID9): Inconsistent costumers**

**1. Avoidance/Minimization**

Have regular meetings with the customer; thereby, we get a clear assurance of what is to be implemented. Providing them with prototype and progress at every stage of the project. Let the customer aware of all potential obligations. In this way, the client knows what they want and what they do not.

**2. Contingence:**

If the customer fails to be consistent with their requirements, the team members must contact the project manager. And they must be made aware of the irreconcilable behavior of the customer.

**Risk (RID10): Financial problems**

**1. Avoidance:**

Project costing should be done realistically and cleverly. Once the budget is fixed, it should not be exceeded.

**2. Contingence:**

Once Financial problems assize, a contingency plan must be submitted to the management.

**3. Minimization:**

Members of the team must be made aware of the significant repercussions of the budget being exceeded. Only after the acknowledgement of all the team members must financial decisions be carried out.

**Risk (RID11): Team members may not have sufficient knowledge to do certain aspects of the project.**

**1. Avoidance/Minimization:**

Assign tasks to the members based on their strengths and team members who are familiar with the topics should help the rest overcome obstacles. Make sure the team members have sufficient knowledge and understanding of the concept. If they face difficulties, members can provide them with videos or links that will help them understand the concepts better.

**2. Contingence:**

If the team member still faces difficulties with the delegated components of the project, their parts will be taken up by members with greater understanding.

**Risk (RID12): Server failure**

**1. Avoidance:**If the server fails for any reason whatsoever, there is a high risk of losing valuable data. We can avoid that by using incremental data backups, which work on backing up data only when data is added or modified and adds that to the previous backup.

**2. Contingence:**In the unfortunate case that this happens, we can use the last-updated back up we have for the project files.

**3.** **Minimization:**Using an incremental backup system will minimize the effects of any data lost due to server failure.

**Risk (RID13): Ill-fitted photos posted**

**1. Avoidance:**Constant monitoring to make sure any unsuitable photos do not exist on the feed, consequently giving out warnings and possibly temporary account limitations to users who belong to that criteria.

**2. Contingence:**As long as there is constant monitoring, ill fitted photos posted are bound to get removed eventually.

**3. Minimization: Code an intelligent bot to report any suspicious photos.**

**Risk (RID14): The project targets not met by a certain time**

**1. Avoidance:**We should set realistic timings as for when goals are to be met. We should observe our previous behavior to learn from any mistakes or bad decisions made, and possibly prioritize our timings better for more future productivity relative to the project.

**2. Contingence:**In case that happens, we need to contact our line manager and seek advice. We would also put more focus onto this project specifically and free up more time for it.

**3. Minimization:**Always make sure we are on track by staying in contact with our line manager.

**Risk (RID15): DDOS attacks**

**1. Avoidance:**DDOS attacks can’t be predicted, but we can reduce the chance of it happening by having someone monitor the firewall for any suspicious activity.

**2. Contingence:** In the unfortunate event of a DDOS attack, we should first contact our line manager and standby until we get proper instructions.

**3. Minimization:**This could be minimized by having someone constantly monitor the firewall to make sure there is not any suspicious activity in incoming traffic.

**Risk (RID16): Team lacks motivation**

**1. Avoidance:**

The team members should always motivate each other. The jobs can be allocated long in advance to keep the participants from being burnt out.

**2. Contingence:**

Members of the team lack motivation or feel overworked; they should approach and share their thoughts with the line manager/.

**3. Minimization:**

To minimize the impact of this risk, team members must maintain a healthy work environment and continuously motivate, encourage, and help each other.

**Risk (RID17): Illegal documentation**

**1. Avoidance:**

Team members should access only legal documents.

**2. Contingence:**

Immediate meeting with the project manager and make them conscious of the illicit paperwork.

**3. Minimization:**

Stray free of all unlawful records.

**Risk (RID18): User experience is poor, and the User interface is not aesthetically pleasing.**

**1. Avoidance:**

Assign UI/UX to the designers in the group. Polls may be put up to see which interface Staff prefer more.

**2. Contingence:**

Schedule a meeting with the customer and inquire about the basic design requirements that must be fulfilled.

**3. Minimization:**

To minimize the impact of this risk, proper research must be done before implementing a particular design. Design specifications must be based on the needs of the customer and what pleases the common eye. The latest design trends may be implemented for a better user experience.