## Novus

Icon

Description automatically generated

# “THE BID”

### **Details and Summarizes USER REQUIREMENT Risk ANALYSIS FOR NOVUS Project decision and plan Project costing usability evaluation of mock-ups**

##### ***VS, IE, KD, ER, AL, AK, MZ, AD***

##### ***Group 3***

##### ***20/10/2020***

**Document details and history**

|  |  |  |
| --- | --- | --- |
| Document Title | Confidentiality Level/Status | Copyright |
| The Bid | Final Version | © 2020  VS, IE, KD, ER, AL, AK, MZ, AD |

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Authors | Changes |
| 20/10/20 | Initial draft | VS, IE, KD, ER, AL, AK, MZ | N/A |
| 09/11/20 | Add Capability Requirements | VS, IE, AL | Capability Requirements |
| 09/11/20 | Added Risks | AL, ER, KD | Risk Analysis |
| 10/11/20 | Added Constraints | VS, IE | Constraint Requirements |
| 12/11/20 | Updates to the Requirements | VS, IE, AL | Capability requirements |
| 14/11/20 | Added Risk Planning and monitoring | AL, ER, KD | Risk Analysis |
| 15/11/20 | Added Project Costing | MZ | Project Costing |

**introduction**

**Purpose**

The document describes the requirements for the social photo sharing system as requested by HWU. This system is intended to be used by members of the general population to share and interact with photos of objects and animals. The user will be able to post their own photos and edit them whilst also interacting with other users’ posted photos creating a social community based around the sharing of these photos.

This document is created for the benefit of Heriot Watt University and The Developers.

**Scope**

The system will be both compatible as a mobile application and as a web application to ensure full compatibility across all devices. The system will use a variety of frameworks and technologies to ensure both the full functionality of the system and the creating of a easy to use interface for the user.

The databases will hold:

1. Information about the user’s profile (eg. username, password, email, etc.).
2. The user’s uploaded photos and their relevant meta-data (eg. GPS data).
3. Each photo’s “community” aspects (eg. comments, points, etc).
4. Information about each user’s usage of the app (eg. time spent on app, most searched terms, etc.)

The general purpose of the system will be for users to share and create communities around pictures of objects and animals. Users should have the ability to express themselves creatively on the system and connect with other users to create a social network.

Furthermore, user data will be sent back to administrators where many facets of it can be studied and analyzed (eg. time spent on app, user interests, user age, photo point rankings, etc.)

**Overview**

This document will outline a general description of the system but more importantly will outline in detail specific functional and non-functional requirements of the system developed by The Developer for Heriot Watt University.

**general description**

**Product Perspective**

The product is being developed for Heriot Watt and will have similar capabilities to social networking applications such as Instagram which focus on the sharing of photos. The product is designed with the emphasis on sharing photos of animals by the use of individuals.

**General Capabilities**

The main functional requirements are:

* Allow users to upload photos
* Allow moderators to modify or remove content
* Allow users to interact with another user’s photo
* Analytical data is collected and sent to administrators

In essence, the product would allow users to post their own photos and interact with other users photos. Our vision for the product is for it to become a lively and vibrant social networking platform based upon the sharing of animal photos.

*Allow users to upload photos*

The product should allow users to upload photos to their own personal page. The uploaded photo can be taken from within the app or from outside. Furthermore, users should have the capacity to edit these uploaded from within the application.

*Allow moderators to modify or remove content*

Moderators of the product should be allowed to remove content as seen fit. Certain content is prohibited from the product (such as photos of people) and moderators should have the capability to remove said content.

*Allow users to interact with another user’s photo*

Users will have the capacity to interact with another user’s photo. This may be through the comments system or through the points system. In addition, a user will be able to see the results of their interaction through global displays such as competitions or leaderboards.

*Analytical data is collected and sent to administrators*

Analytical data such as number of hours spent on the product, most popular tags, number of photos posted, number of points, etc. will be collected and sent to an administrative body for data analysis.

**General Constraints**

* The product must be compatible on PC and mobile
* Product interface must be user friendly (ie. Easy to understand and appealing)
* User data must adhere to GDPR standards
* The product must be well optimized and run fast on all devices
* The systems database capacity must be scalable
* Normal user permissions and power will be different to a moderator’s
* Product to be complete by the first of May 2021

**User Characteristics**

The large majority of users are expected to be regular individuals. The products is meant to appeal to a wide audience of individuals who are passionate about animals and who intend to use the product as a means of social networking with others who feel the same way.

Furthermore, we do expect a small presence of organizations within animal industries to use the application. This may be for advertising purposes or for the betterment of public relations.

**Operational Environment**

The product will be compatible on all internet browsers, Android and IOS.

Flutter will be used to create the app. Flutter is a SDK by Google that allows for the development of apps which are compatible with multiple platforms using the same codebase.

Firebase will also be primarily used as the database for the product. This is because it offers interoperability with flutter and will allow us to store and use all relevant data in a comprehensive and simple manner.

**Capability requirements**

**High Priority**

**Medium Priority**

**Low Priority**

**F-UR1 User Interaction**

F-UR1-1 Comment Section

Every photo has its own comments section

F-UR1-2 Comment Manipulation

Users can post/delete and edit comments on a photo comment section

F-UR1-3 Point Section

Users can “score” the photo using the point system

F-UR1-4 Channel Creation

Users can create channels which act as the user’s hub for all their photos.

F-UR1-5 Photo Search

Ability to search for photos based on photo tags and other relevant information.

F-UR1-6 User Following

Users can follow other users which means they are more likely to see their photos in their feed

F-UR1-7 Photo GPS Search

Users can search for photos based off the location the photo was taken

**F-UR2 User Analytics**

F-UR2-1 Leaderboard of Photos

There is a leaderboard of all top-rated photos.

F-UR2-2 Summary Analytics Report

A report is generated which supplies administrator with the current state of the platform

**F-UR3 Photo Manipulation**

F-UR3-1 Photo Upload

Users can upload photos to the platform

F-UR3-2 Photo Tagging

Users can tag photos to categorize them.

F-UR3-3 Photo Editing

Users can make changes and edits to their photos in the app.

F-UR3-4 In-app Camera

Users can take photos from inside the app.

F-UR3-5 Location Tagging

Users can tag the photos they have taken with its relevant GPS metadata.

**F-UR4 Photo contest**

F-UR4-1 Create contest

Users can create general or theme-based contests

F-UR4-2 Contest settings

Creator of the contest can edit contest settings based in their preference

F-UR4-3 Join contest

Users can participate in different contests

F-UR4-4 Leave contest

Users can leave contests

F-UR4-5 Moderate the contest

Creator of the contest can moderate the contest posted in the contest

**F-UR5 Leaderboard**

F-UR5-1 Data display

Data from different contests should be displayed

F-UR5-2 Trending

Trending contests should be displayed.

F-UR5-3 Player username

Username of the players with the points should be displayed.

**F-UR6 Group**

F-UR6-1 Create group

User can create a general or theme-based group

F-UR6-2 Join group

Users can join different groups

F-UR6-3 Leave group

Users can leave groups

F-UR6-3 Group settings

Creator of the group can edit the settings of the group

F-UR6-3 Moderation the group

Creator of the group can moderate the content of the group

**F-UR7 Registration**

F-UR7-1.1 Register with Email

User shall be able to sign up using an email address and password.

F-UR7-1.2 Verify Email

User receives email upon registering.

F-UR7-2.1 User Login to account

User should be able to sign in as a recurring user with their email and password.

F-UR7-2.2 User Logout of account

User should be able to logout of their account.

F-UR7-3 Secure login credentials

User details should be stored securely in the system.

F-UR7-4 Deletion of Account by User

User profiles will be removed from the system once deleted by the user.

F-UR7-5 User adds profile details upon account creation

User should be able to add their profile details when creating their account.

**F-UR8 Profile**

F-UR8-1 Profile Details

User can edit profile details such as the display name, dob, profile picture and description.

F-UR8-2 Archive Posts

Posted deleted by the user will remain in a user archive for them to refer to.

F-UR8-3 Follower/Following Count

following and follower count can be displayed on a profile.

F-UR8-4 User Posts

All post associated to a profile can be displayed when a user views another or their own profile.

F-UR8-5 Profile Searching

Users could search for other users on the platform.

**F-UR9 Administration**

F-UR9-1 Administration support

Administration tools should be available to manage accounts and data.

F-UR9-2 Delete Account

Admins should be able to delete accounts of users.

F-UR9-3 Edit Account

Admins should be able to edit a given users account.

F-UR9-4 Maintain system stability

Admins should be able to handle bug reports or issues while the system is active.

F-UR3-5 Moderate the system

Admins should be able to moderate content as well as users on the system.

**Constraint requirements**

**High Priority**

**Medium Priority**

**Low Priority**

**NF-UR1 Hardware**

NF-UR1-1 Environment

The system should be hosted on an internet-connected web server.

NF-UR1-2 System Optimization

The system will be optimized to run smoothly on various devices using faster/slower hardware.

NF-UR1-3 Storage Scalability

The systems database capacity must be scalable in order to account for increasing users and data on the platform.

**NF-UR2 Software**

NF-UR2-1 Viewports

There interface of the system must have different viewports based on different devices such as mobile, tablet and desktop.

NF-UR2-2 Browser Support

The website for the system must run across all widely supported web browsers.

NF-UR2-3 Disability Support

The system could provide functionality to improve accessibility for disabled users (deaf, colorblind, etc).

**NF-UR3 Security**

NF-UR3-1 Appropriate view of data

Users must be able to access and edit their own data and not someone else’s.

NF-UR3-2 Credentials Stored Securely

Users credentials stored securely in the system

NF-UR3-3 Security techniques

The system must incorporate standard techniques to prevent code injection which could break the system.

NF-UR3-4 Validate Environmentalists

Environmentalists registering to the system can be validated by an administrator before they can start posting.

NF-UR3-5 Privacy Options

Users on the system could have the option to hide their content from people that aren’t following their account with their knowledge.

NF-UR3-6 Privacy Options

Users on the system could have the option to hide their content from people that aren’t following their account with their knowledge.

NF-UR3-7 Criteria for password

Users registering to the system must follow a set criteria for their passwords, which could be mix between numbers, letters, symbols.

**NF-UR4 Time**

NF-UR4-1 Deployment Deadline

The system must be ready for use by the 1st April 2021.

**NF-UR5 Performance**

NF-UR5-1 Responsiveness

The system could respond to user actions within 2 seconds.

NF-UR5-2 Large active user base

The system could handle over 100 users at a given moment.

NF-UR5-2 Fast read/write database speeds

The systems database could have fast reading or writing speeds for data

**NF-UR6 Data**

NF-UR6-1 Storage of data

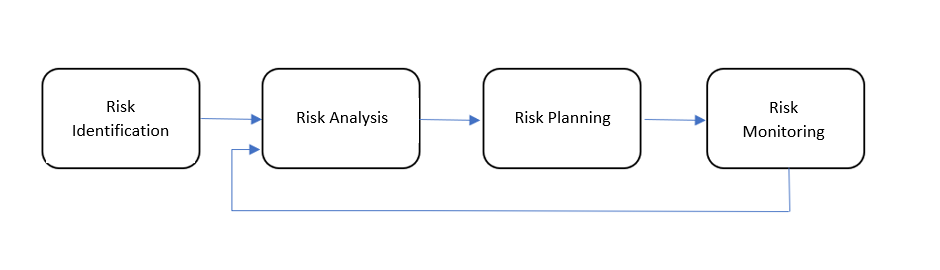
All user data on the system must be stored on firebase which would have a limited size.

**risk analysis**

**1. Risk Management Approach**

*1.1 Background*

A risk management plan is generated to minimize the impact of any potential risks that may occur during the project. The risk management strategy can be

broken down into four key stages that take place sequentially.

1. *Risk Identification*

At this stage, we identify and list out the potential risks as product risks, project risks and business risks.

1. *Product risks:*

These risks have an impact on the quality and performance of the project.

1. *Project risks:*

These risks have an impact on project resources and deadlines.

1. *Business risks*

These risks have an impact on the company creating or acquiring the project.

1. *Risk Analysis*

At this stage, we access the list of risks and analyze them. We figure out the likelihood, the consequences and effect of each risk.

1. *Risk Planning*

At this stage, we draw up a plan to prevent and mitigate the impact of the risks.

1. *Risk Monitoring*

At this stage, we monitor the risks as we go ahead with our project. Then, we add on to our risk analysis as more information about the risks is acquired.

**2. Risk Introduction**

*2.1 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 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.

*2.2. 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 cannot 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 | Tolerable | 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 conflicts with other projects. |
| R1D22 | Staff | Team members can have problems with their workload | High | Tolerable | 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**

***3.1* *Background***

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

*3.2 Risk Strategies*

  There are three types of risk strategies we must 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.

*3.3 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 be 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 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:**If 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 cannot 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.

**Risk (RID19): Unable to raise extra capital if cost exceeds budget.**

**1. Avoidance:**

Team members should create a financial structure beforehand including the project specification cost that are software costs, third party communication costs, etc.

**2. Contingence:**

The group members must try to raise money for the project via various sources such as private/government banks, individual asset lenders.

**3. Minimization:**

Regular costing should be done, that is every week/month. A buffer amount should be kept on hold for these kinds of situations.

**Risk (RID20): Copyright/Patent issues.**

**1. Avoidance:**

Use caution if it is like an already existing project. Look for the license or permissions before you use anything that is not yours.

**2. Contingence:**

A court case can be done which includes identifying the owner, identify the rights needed, contact the owner, and negotiate whether payment is required and hence getting your agreement in writing.

**3. Minimization:**

To reduce issues like this, the team should be in talks with the owner of the rights, if it is not acquired by the team. They should try to gain rights to it in the initial stages to avoid the project from getting dismissed in court.

**Risk (RID20): Team member’s inability to understand a spoken language.**

**1. Avoidance:**

At the start of employing a team, the members should be questioned about their preferred/known languages. It is better for the team’s communication if all team members can understand each other verbally.

**2. Contingence:**

The group members can easily communicate with each other in native English language using third party apps for translation verbally and text.

**3. Minimization:**

This can be minimized using external software or by communicating with a fellow team member who speaks a common language and can act as a translator when needed.

**Risk (RID21): Team members can have problems with their workload**

**1. Avoidance:**

Group members should let others know if they are having problems with their workloads as this can cause failure to produce work.

**2. Contingence:**

Work needs to be split amongst members equally for productivity. If one department is being overwhelmed with work, the team should hire extra members for that department to reduce workload individually.

**3. Minimization:**

Group members should not work on any task on their own, it should always be done in pairs, in this way even if one person doesn’t approach the group leader, his task partner may raise this topic.

1. **Risk monitoring**

*4.1 Background*

Risk monitoring is a process where each risk is assessed and its track is monitored, i.e., whether the risk probability has increased or decreased, or whether the impact has a change. Risk monitoring is a process which goes on until the end of the project.

*4.2 Documentation*

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

Team should always be wary of their state of organization. Whenever anyone

observes any unorderly behavior, a meeting should be conducted to discuss it

to avoid it.

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

Team should be careful enough to note down any necessary explanations for any

code written. In case code is still not understood by any team member(s), they

should immediately contact the team and ask for a more practical explanation.

* **Risk (RID3): Poor communication**

Roles assigned to a member or two of the group to monitor and observe the

communicated behavior concurrent between the group’s members. In case

communication behavior seems insufficient, they should conduct a meeting to

discuss it.

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

Team members should always have their precautions during every stage of this   
 project. This is to minimize the effects of any potential negative outcomes and

errors.

* **Risk (RID5): Software limitations**

The group should prioritize this matter as it is very relevant to the project.

Software options and choices should be discussed in detail to avoid

any irregularities midst implementation.

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

The group should make sure everything they are posting and/or working with

gets uploaded for the other members to be able to access. GitHub is the

current solution for that.

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

The team must always look at what is being added and what is being deleted. Whatever work is going to be deleted must be double-checked. After the work is complete people can push their work into their GitHub repository. The team leader can have a final look before they merge it.

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

The team keeps track of all the functional and non-functional requirements that are met as they go forward with their project.

* **Risk (RID9): Inconsistent customers**

Regular meetings with the customers. We must pay attention to the complaints and demands of the customer. Keep the customer aware of all the possible work that will be done. In that way, the customer will tell us what they like and dislike beforehand.

* **Risk (RID10): Financial problems**

The team must ensure the financial expenditures does not exceed the set budget. Any financial commitments that are going to be made must be negotiated with the line manager and other team members; the payment must be carried out only after the approval of the line manager.

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

The team must be made aware of people’s strength and capabilities. Any member having a hard time understanding or completing given work must as for assistance.

* **Risk (RID12): Server Failure**

Implementation of incremental data backup as soon as possible is

recommended to minimize any negative effects a server failure can

have on the progress of this project.

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

Setting relevant role(s) for team members should be enough to eliminate any

photos in the wrong category, if any.

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

Keeping contact with our line manager should be a set priority. This will

help us know our position and place. This will also let us know if we need to

make changes before it is too late.

* **Risk (RID15): DDOS attacks**

Set a role to a group member to keep watch on the firewall, in case anything

looks slightly suspicious, it should be instantly reported to both the group

members and the line manager.

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

Members can look out for the general demeanor of its members. Make sure everyone and everything is moving smoothly.

* **Risk (RID17): Illegal documentation**

Always make sure the documents that are used by the members are not illegitimate.

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

The members in charge of the design aspect must take get an opinion from the people to check if the user experience is good and accept constructive criticism.

* **Risk (RID19): Unable to raise extra capital if cost exceeds budget.**

The team members responsible for the financials of the project should keep a regular check on the projects financial records and statements and make changes to the financial structure if costs change.

* **Risk (RID20): Copyright/Patent Issues**

The team members should be cautious of not using anything like an existing project or an idea. If necessary, they should seek for a license or a permission to use anything that is not yours.

* **Risk (RID21): Team Member’s inability to understand a spoken language**

The team members should keep in touch with their colleagues often to ask if they can understand the plan of the project and if they are comfortable in the working environment.

* **Risk (RID22): Team Members can have problems with their workload**

The team leaders should not overload their team members. They should always be mutual when assigning over time work, i.e., not forcing it on the members. Team members should be in regular touch with each other on how they are performing and if they are comfortable in the working environment. The team members should always approach their leader if they think they are being overworked.

**5. Definitions**

*5.1 Risk Probability*

|  |  |  |  |
| --- | --- | --- | --- |
| Probability | Matrix Value | Rate of Probability | Description |
| Low | 0.25 | <25% | More likely not to occur |
| Moderate | 0.50 | 25-50% | May or may not occur |
| High | 0.75 | >50% | Expected to occur |

*5.2 Risk Impact*

|  |  |  |
| --- | --- | --- |
| Impact | Matrix Value | Description |
| Low | 0.25 | Very little impact on project |
| Tolerable | 0.50 | Manageable impact on project |
| Serious | 0.75 | Severe impact on project |

*5.3 Risk Probability and Impact Matrix*

|  |  |  |  |
| --- | --- | --- | --- |
| Probability |  |  |  |
| High | 0.1875 | 0.375 | 0.5625 |
| Moderate | 0.125 | 0.25 | 0.375 |
| Low | 0.0625 | 0.125 | 0.1875 |
|  | Low | Tolerable | Serious |
|  | **Impact** |  |  |

|  |  |
| --- | --- |
| Risk Priority | Color |
| High |  |
| Moderate |  |
| Low |  |

**project decision and plan**

**Choice of Framework**

We decided to go with an existing development kit for the project, it ensures we have access to documentation as we develop the system, and less time will be wasted testing code.

The flutter development kit was decided on due to its ability to support all platforms from just one codebase as well as its rich feature set. It’s a new developing tool for us but we are more than prepared to learn. Flutter uses dart which is very similar to java syntax which makes it easier to pick up since we have worked on various java related projects. To handle the backend, we would use firebase which provides us with the systems database as well as a hosting service.

**Multi-Platform Application**

The system is developed to be a multi-platform application (Browser, Windows, IOS, Android). This is based on the specification provided to us. The user interface will primarily be designed using flutter and backend handled by firebase.

**Version Control**

GitHub will be used to control and manage updates to the system code as well as all documentation. This will avoid files getting over written and maintain backup versions whenever needed.

**Communication Tools**

To maintain communication with the team, we will be using Microsoft Teams as a conferencing tool for us to discuss the project.

**Intended Audience**

Based on the specification, the system needs to be targeted towards people of all ages and cultures, that are passionate about site seeing and animals. The core functionality of the application is for users to build profiles, sharing their photographs amongst a community with common interests. It will also be a place for educational outreach and organizing public meetups, where people can learn more about the area, they live in.

With that, the design of the system must be appealing towards the younger demographic as well as maintain professional designs that appeal an adult userbase.

**Risk Management**

Possible risks that could arise during the project have been addressed by the team members. A proper risk management has been documented in order provide context on what they are, and what avoidance strategies we have in place.

**Project Costing**

This is mainly theoretical, since we don’t have any funding at our disposal, so all software we use is open source. We will provide a project costing file which document assumptions for each member’s average wage and will assume working hours on a weekly basis. Also included would be assumed costs for the devices needed, for the development of the system.

**Mockups**

Prototypes for the system will be create using PowerPoint which will outline the layout of the system. This will then be used to collected subjective data on our design.

**Usability Testing**

Five Group members will perform usability tests, where we will use the results to come to conclusions that will help us improve the system during development.

**Team Roles**

Each member of the team will contribute to areas of the project bases on the skills and experience they have. We carried out sessions discussing our different capabilities, resulting in roles that each member is responsible for over the duration of the project.

(ROLES TO BE ADDED)

**project costing**

**1. Introduction**

*1.1 Purpose*

The purpose of this entire document is to list the costing of the entire Photo-Based Social Media Platform. The document contains the entire expenditure of the entire project and individual costs as well.

*1.2 Scope*

The scope of this project is to develop a photo-based social media platform.

The main goal of this application is to provide a safe platform for the people to share the pictures that they take of objects/animals and get feedback from others and gain points as well.

The platform even allows the people to have geolocation tagging on their pictures. This allows others on the platform to visit the places and click pictures of the objects or the animals on their own.

*1.3 Overview*

The document is to outline and summarize the list of areas where expenditure might occur. There will be a detailed breakdown of the cost provided in the document below. At the end of the document there will be a complete summarized table of the cost that will be provided to the client.

**2. Costing Tree**

Below is the breakdown tree of all the costs that will be detailed later in the final summarized table.

**Diagram

Description automatically generated**

**3. Cost Analysis**

This section provides the written descriptions of individual cost that is mentioned in the breakdown tree above.

*3.1 General Descriptions*

**Staff**

Salary of all the staff members who will be involved in this project.

**Project Manager**

They are supposed to oversee the entire project and macro manage everything.

**Lead Software Developer**

They are the same as a software developer, but they lead the entire software development team and review the code.

**Software Developer**

They are supposed to develop the application itself; they are also possibly the ones who will maintain the application in the future.

**Quality** **Assurance** **Staff**

They are the ones who will inspect, test, and provide feedback to help with the development.

*3.2 Equipment*

**Servers**

Costs of servers that will be used during the development process, these are required for the hosting.

**Development Kit**

Development Kit contains the development machines that will be used to develop and run the software. These are charged at 50% of a unit cost per year.

**Mobile Testing**

As the application is meant to run on mobile devices, a handset (android and iOS) is required for testing the application on the device. The development machines are used for desktop testing for cost saving purposes.

**Support & Maintenance**

Costs of the support and the maintenance that will be required after the development process.

**Support**

This will be the cost of keeping the system constantly updated and fixing errors for a period of 12 months.

It includes as mentioned earlier:

* + - Online Maintenance
    - Software Updates
    - Software Support

**4. Costing**

All the costs mentioned above are now calculated and shown in the tables below. Please note in case of conflict regarding extensions or other issues, the costs are subject to a 10% charge.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Staff Cost | Unit/Hours | No. of Staff | Cost Per Unit | Wage per Person | Total Cost |
| Project Manger | 150 | 1 | 62.16 | 9324 | 9324 |
| Lead Software Developer | 150 | 1 | 46.11 | 6916.5 | 6916.5 |
| Software Developer | 150 | 4 | 42.00 | 6300 | 25200 |
| Quality Assurance Staff | 50 | 2 | 7.25 | 362.5 | 725 |
|  |  |  |  | Total Cost | 42165.5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Equipment | Units/Hours | Cost Per Unit | Type | Total Cost |
| Server | 1 | 2000 | Rented | 2000 |
| Development Kit | 6 | 399 | Rented | 2394 |
| Mobile Testing | 2 | 500 | Rented | 1000 |
|  |  |  | Total Cost | 5394 |

|  |  |  |
| --- | --- | --- |
| Support | Length | Cost |
| Updates and Fixes | 12 months | Free\* |

|  |  |
| --- | --- |
| Total Cost of System | 47559.5 |
| *\*Included in the total cost* |  |

*Costing Refs*

Project Manager Cost: <https://www.indeed.com/career/project-manager/salaries>

Software Developer Cost: <https://www.indeed.com/career/software-engineer/salaries>

General Wages: <https://www.dol.gov/general/topic/wages>

*Development Kit*

<https://www.amazon.co.uk/Optiplex-i7-2600-Windows-Desktop->Computer/dp/B07YST68J3/ref=sr\_1\_2?dchild=1&keywords=desktop+i7&qid=1605036978&sr=8-2