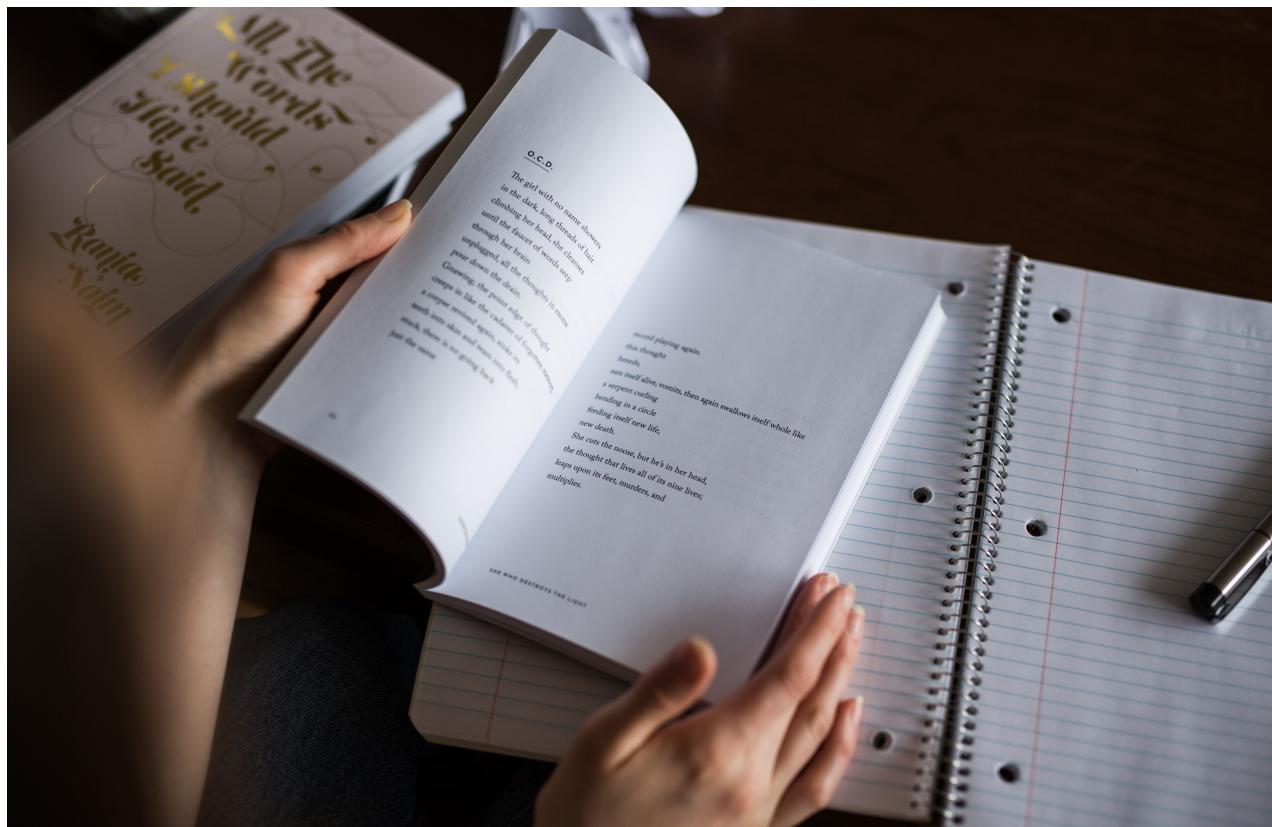




FEASIBILITY STUDY



MEMBERS:

Bindu	20114024	CSN-254
Deepanshu	20114030	
Deepanshu Meena	20114031	
Priya Singh	20114077	
Sonali Gupta	20114094	
Youwan Soni	20114107	

A statement of the task to be undertaken:

We would like to develop application software that will provide all artists a platform for expressing their art, particularly paintings and their variants. Not only they will have their art displayed to a wide audience but also there will be various competition, which could be free or requires a participation fee, and there will be various incentives for an artist to participate in these competitions.

THE CUSTOMER FOR WHOM THE WORK WILL BE DONE

Arthub is intended to target two types of customers, the one who creates art and the other who appreciates it.

As a customer who creates art, they want a platform for expressing their art among those who appreciate it and some sort of competition to happen between brilliant artists all over the world. They also want a way to find people belonging to their own category of art style and get to know their mindset and art experiences. As a customer who appreciates art, they want a simple platform where they can view the great artist insights of people around the globe. The paintings which will come for competitions will be up for display at for the commons and whoever likes it can buy it.

PRELIMINARY REQUIREMENTS ANALYSIS:

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Preliminary requirements and expectations that users have from the Arthub are:

- Web Interface
- ❖ Public Side
 - An option for the user to sign in and login to the system.
 - An option for the user to upload a document on the system.
 - An option for the to download the document from the system using the generated key at the time of uploading.
- ❖ Admin Side
 - Admin can block some users temporarily if there is suspicion of them being involved in activities that can DOS the server.
 - Admin can also delete a user's account if it seems necessary.
 - Admin can give contact details of a user if need.
- *A database where we store the user's details along with the information about which file is accessible for which user.*

PROBABLE TECHNICAL REQUIREMENTS:

For this project, most probably tech stack will include android studios which is an IDE and will be helpful in managing both the frontend and the backend. MySQL will be used for database management. For the customer, it will just be an app in the play store which can be downloaded and would require only an internet connection.

SUGGESTED DELIVERABLES:

Deliverables for the project include source code, design document, SRS document, test document, user's manual, etc.

Source Code- Source code is a complete structured and modeled program files of software written in high-level languages or assembly languages.

- **Design document** - Design documentation is a collection of documents and resources that covers all aspects of your product design. This Documentation includes information about users, product features, and project deadlines; all essential implementation details; and design decisions that our team have agreed on.
- **SRS document**- A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil all

stakeholders (business, users) needs.

➤ **Test documentation - Testing**

documentation involves the documentation of artifacts that should be developed before or during the testing of Software.

Documentation for software testing helps in estimating the testing effort required, test coverage, requirement tracking/tracing, etc.

➤ **User Manual- The User Manual**

contains all essential information for the user to make full use of the information system. This manual includes a description of the system functions and capabilities, contingencies and alternate modes of operation, and step-by-step procedures for system access and use.

PROCESS TO BE FOLLOWED:

The primary reasons for choosing an agile based philosophy are the following:

- The Agile model works best when we are able to include customer representatives on the team. Though the end-users are not defined in our case, we will still try our best to reach out to the potential users who will benefit from our project and make efforts to document their requirements efficiently. The potential users will be able to get an idea of the updated software after each phase and iteration and will be able to provide feedback.
- Depending on the different stages of the market research and the feedback gathered through surveys and questionnaires, the requirements of our potential customers can change. An agile based model will be able to address these changing requirements, whereas a model like iterative waterfall locks the requirements in the initial phases itself. The product managers will be responsible for remaining in touch with the customers and enhancing the user-developer interaction and communication.
- The Agile philosophy will help in delivering incremental versions of the project. This is required because we are valuing the customer and development team feedback not only for detecting the errors committed in the previous stages, but also for improving the quality of the software and the user experience.
- Agile Philosophy will bring another benefit to us because of the nature of our project. Our project has multiple features like

login-signup, file upload-download, result declaration, buying an art, giveaways and managing the sponsors. We will also be looking at the possibility of incorporating a bidding system. Using the Agile method, we would be able to arrange these features in a sequential and logical manner and implement the simpler features in the early phases and more complex features in the later phases.

- After implementing one feature, the product would be deployed on servers. Then it would be made available to the general public and some selected potential users and beta customers for providing necessary feedback. Along with this we will release progress reports, prepare user interface presentations and conduct surveys for gathering more information about user experience and feedback.
- Once we have the feedback, the agile philosophy will help us in addressing the changing requirements. Also owing to the incremental nature of the project development and implementation, we would be able to deploy the product in usable form with limited functionalities. This will ensure that we have a usable end product with some of the implemented features in the given timeframe, instead of only having documentation and reports, which might be the case in an iterative waterfall setup.
- The Agile philosophy will also help in establishing a better communication between team members allowing them to discuss and plan the upcoming phases. This will increase the overall efficiency and enhance productivity.

- The Agile philosophy usually deploys pair programming in which two programmers work together on a feature and one does the coding part while the other reviews the code as it is typed in. This decreases the chances of bugs in the code and also increases code productivity..

OUTLINE PLAN:

Firstly the project proposal is created and a thorough feasibility study regarding user requirements, cost, timeline, and resources is carried out. After this, SRS documentation is prepared in order to be used by users and various stakeholders. And finally team members with the help of SRS document prepare the design of the software. The next coding phase begins where work is divided wisely among team members to develop the frontend and backend of the project and then integrate it to working software. Risk Analysis, Testing, and Deployment are the next important steps undertaken in the project.

Milestones:

Phase-1: Project Proposal, Feasibility Study, SRS document, and Design Documents are prepared.

Phase-2: Coding and Implementation of Software

Stage I - Development of backend modules that will take user code as input and will output variables' state at every line of code.

Stage II - Development of frontend in which backend modules are used to visualize the variable states in an animated manner.

Stage III - Monitoring of various flaws or presence of bugs in the code.

Phase-3: Testing and Deployment of software

VISIBILITY PLAN:

It is absolutely essential for the development team to be in constant touch with the customers in order to clarify any doubts and conflicts. There can be a case where a new unforeseen challenge arises which is difficult to solve using the standard software engineering principles. In this case, the feedback and input of the client are necessary to make the right and optimum decision. Not only during requirement gathering but communication is also needed and essential through all the stages of the development lifecycle. Hence, it is necessary to establish a channel or medium which can be employed easily by both the customers and developers for effective communication. Any inaccuracy during communication can lead to misunderstanding and inappropriate work which can add to the cost and effort of the project severely. Our team will ensure maximum efforts to increase the visibility of the project and establish effective communication with the end-users. One difficulty that we face here is that even though the requirements of the customers can be defined based on the general problems faced by customers using similar platforms of art competitions and buying in the market, we might not be able to gather feedback after implementing an iteration of the project. This is because our project has been proposed by observing the general difficulties faced by many users and from personal experience. There are no defined end-users for the project. We will have to employ various visibility methods to keep our clients updated on the

in various stages of the software development lifecycle.

RISK ANALYSIS:

As with our project, it is not entirely risk-free. We have identified three major risks discussed below:

➤ **Resources:** Resource risks involve the team's technologies for their use. Due to costs and other external constraints, the team may not be able to obtain the needed or best resources to complete parts of the system. For example, one of the most versatile resources used in our project is the database system. If it gets corrupted at any point in time, we may lose all the data associated with different users.

To minimize it we may use multiple databases instead of one database to store data. In addition to this, we are a small group of 6 students and it may happen that some of them may not be able to contribute due to different reasons(as not everyone is tech-oriented or they may not have full knowledge of the technologies used), so we may lack in human resources and it might be difficult to deliver the product in the given timeframe.

➤ **Time:** Our course demands that the project must be completed within one academic semester and any extensions are not possible. This introduces the risk that the system may not be completed with the full functionality the client wants within the given

timeframe of a semester. In this case, there is another risk of delayed implementation of the full system if the client chooses to wait until the system is fully functional. This can be minimised by taking a buffer time.

- **Functionality:** Functionality risks are associated with how the system works. This includes the interface that may not be user-friendly and the clients may not like them. The greater risk is that the system may not function the way the client wants it to do. It can be minimised by omitting such functions which are not of so much use to the user as decreased functionalities automatically decrease the risks associated with them. It may also happen that the user's requirements may change throughout the project. So depending on the situation, we may have to implement the changes within the given timeframe. Effective communication through surveys and user feedback can help to understand the user requirements more clearly