Assignment No. 2

Aim:
Project problem statement feasibility assessment using NP-Hard, NP-Complete or satifability issues using modern algebra and/or relevant mathematical models.

Problem Statement:
LiFi is a data transmission technology using visible light communication which provides faster data transmission. This project is developed to manage the data transmission across various devices. The main modules in this project are API, file selection, file/data transfer, file receiving, starting or terminating transmission devices, closing of user software.

0.1 Mathematical Model:

User Module:
set U = U1, U2
U1 = User name, Password
U2 = File submission/selection

System Module:
set S = S1, S2, S3, S4, S5, S6
S1 = Accept file
S2 = Load the file for transfer
S3 = Initiate the sending and receiving hardware devices
S4 = Start data transmission
S5 = Accept file at receiver side
S6 = Terminate connection

0.2 Common factors of Feasibility Study:

The feasibility study is major factor which contributes to analysis of system. In earlier stages of software development, it is necessary to check whether system is feasible or not. Detail study was carried out to check work ability of proposed system, so feasibility study is system proposal regarding to its work ability, impact on organization, ability to meet user requirements and effective use of resources. Thus when application processes it normally goes through a feasibility study and risk analysis. Feasibility of project is checked using various categories like.
1. Technical Feasibility.
2. Economical Feasibility.
3. Operational Feasibility.
4. Time Feasibility.

0.2.1 Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Hardware Configuration	Software Configuration
Main Processor – Pentium IV processor 2.6 GHz.	Operating System - Windows 7/8/10 or Linux.
RAM - 128 MB.	Programming Languages – Java, C and Assembly.
Monitor - 15'6" color.	File Manager – Linux/Windows file manager.
Hard Disk - 20 GB.	Tools – IntelliJ IDEA, Eclipse, CodeBlocks.
Cache Memory – 512 MB.	
Arduino, MSP430 G2 microcontrollers	
LED, Photodiode, BreadBoard	

Table 1: Technical Feasibility

0.2.2 Economical Feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Cost Estimation:

There are two types of costs incurred when using a virtual operating system approach:
1) The costs of writing the utilities: This is a one-time cost, since these utilities are independent of any real operating system. The program development costs for the utilities will be similar to those for any other software system designed for a specific machine, since the virtual operating system utilities are designed for the virtual machine.
2) The costs of implementing the virtual machine: These are incurred once for each different host operating system within the organization. It is important to note that this is the only cost in moving all personnel and software to the new computing environment.

0.2.3 Operational Feasibility:

Operational feasibility reviews the willingness to support the proposed system. This is probably the most difficult of the feasibility to gauge. Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of the system development.

0.2.4 Time Feasibility:

Similar to economic feasibility, a rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required time frame.

SR	Scheduling Strategy	Period
1	Planning	July - August
2	Requirement Gathering	August - October
3	Designing	October - November
4	Coding	November - March
5	Testing	March - April

Table 2: Time Feasibility

0.3 NP-hard, NP-Complete Analysis:

When solving problems we have to decide the difficulty level of our problem. There are three types of classes provided for that. These are as follows:
1. NP Class
2. NP-hard Class
3. NP-Complete Class

0.3.1 NP Class:

Informally the class P is the class of decision problems solvable by some algorithm within a number of steps bounded by some fixed polynomial in the length of the input. Turing was not concerned with the efficiency of his machines, but rather his concern was whether they can simulate arbitrary algorithms given sufficient time. However it turns out Turing machines can generally simulate more efficient computer models (for example machines equipped with many tapes or an unbounded random access memory) by at most squaring or cubing the computation time. Thus P is a robust class and has equivalent definitions over a large class of computer models. Here we follow standard practice and define the class P in terms of Turing machines.

0.3.2 NP-Hard Class

A problem is NP-hard if solving it in polynomial time would make it possible to solve all problems in class NP in polynomial time. Some NP-hard problems are also in NP (these are called "NP-complete"), some are not. If you could reduce an NP problem to an NP-hard problem and then solve it in polynomial time, you could solve all NP problems. Also, there are decision problems in NP-hard but are not NP-complete, such as the infamous halting problem.

0.3.3 NP-Complete Class

A decision problem L is NP-complete if it is in the set of NP problems so that any given so- lotion to the decision problem can be verified in polynomial time, and also in the set of NP-hard problems so that any NP problem can be converted into L by a transformation of the inputs in polynomial time. The complexity class NP-complete is the set of problems that are the hardest problems in NP, in the sense that they are the ones most likely not to be in P. If you can and a way to solve an NP-complete problem quickly, then you can use that algorithm to solve all NP problems quickly.

0.4 Related to proposed system:

In our proposed architecture we are going to show in this section the relevance of NP Hard and NP Completeness. The proposed system comes under the NP Complete category and it can be achieved in finite time with the consideration and definition of NP Hard problem, a problem in NP Hard can not be possibly achieved, so it is clear that our our proposed system is NP Complete

NP Hard problem can not be optimally solved in an efficient way. The proposed system is designed with NP Complete category hence we can solve the problems in efficient manner.

0.5 Conclusion:

The project supports NP Complete class problem as in our proposed system users can transfer the data across various devices and also provides control over individual hardware. Hence this project is under NP Complete class.