CHAPTER-4

SYSTEM DESIGN

System architecture of our project can be explained by using following diagram.

Figure 4.0 CPMS Architecture
Basically our system is divided into two parts

1]. Web Role

2]. Worker Role

Explanation of these roles is as follow.

1]. Web Role:

Web role basically used for creating user interface which can be viewed by user.

We are using MVC 3 architecture in web role.

So Web role is divided into three parts

a. View:

This part creates user interface for the client. User’s request can be capture from this view. After getting user request it invokes appropriate controller function which gives proper response to request.

b. Controller:

All requests for new view is first goes to controller which ultimately transfer control to respected view. All business logic is implemented in controller. In our system all cloud services for respected request is called from controller.

c. Model:

Model basically contains all class which are used to transfer data from controller to view.
2]. Worker Role

Worker role basically perform all business logic so that it can be used by many web role. It access database which are required in our project.

Web role knows to worker role by using its endpoint only by which we can access worker role.

We are deploying our project on Microsoft azure so it is very important to know what is azure and its architecture which is explain below.

We are going to use cloud for deployment of the services. For that purpose we are going to use Windows Azure. Therefore our system’s architecture will be the architecture of the Windows Azure. The uni4cloud provides the capability of deploying the application developed in multiple clouds so that the end user is not just stuck with only one cloud service provider [1].

Windows Azure Architecture

![Figure 4.1 Windows Azure Architecture]
4.1 Project Problem Statement Feasibility

In my research project, I am trying to develop platform for effect project management. So our problem statement is divided into two parts like selecting and providing best method for project management and implements it so that user can access it. I think that selecting best method for project management is most important aspect of our project. We are using and providing agile software development model. Agile provides iterative software development methodology [63]. I have selected following problem to show that our problem statement is as NP-complete.

Problem Statement:

Success Percentage: In agile development model, find success percentage of project for given project plan.

Q is an Complexity class problem if:

1) where the value of Q is involved in the Complexity class problem.

2) The reduction of the polynomial value is to Q

Is success percentage problem; we can prove that success percentage is NP-complete as explain below,

Solution for 1st requirement:

The budget and schedule of software project under development are also critical and the pressure of the same is impacting the software development lifecycle time and effort it is important to have a open source effort estimation model to measure each effort that is
made in the development of the software project [30]. The important point to focus is the collaboration of the software development team and the client [6].

![Diagram](image)

**Figure 4.2 Basic Phases In Agile SDLC**

Above diagram explain the basic phases in agile software development life cycle. Before starting actual practices of agile development cycle, software development team has to create pre-iteration planning for upcoming iteration development phases. Every iteration contain set of state like

1) Planning
2) Testing
3) Implementation
4) Feedback

Pre-iteration plan contain detail information about various iteration which will takes place during project development phase and various states mention above. We are **assuming that** if development team follow pre-iteration plan 100% effectively then project success rate will be 100%.

According to problem we have to find what will be percentage of success of selected project by observing pre-iteration plan, we cannot predict percentage of success only
From pre-iteration plan. To find percentage of success of project, we need know how efficiently pre-iteration plan has been followed. So we can verify success percentage of project when we have information about how effectively pre-iteration plan has been followed.

From above discussion it is clear that success percentage problem is in NP

**Solution for 2nd requirement:**

Following diagram explain iterative nature of agile development life cycle

![Agile Project Life Cycle Diagram](image)

Figure 4.3 Agile Project Life Cycle

From above diagram it is clear that in each iteration development team follow some state shown in figure [4.3]. If we consider these states as node (vertex) then for single iteration we can draw graph as below,
From above graph it is clear that iterative nature contain cycle and as there are many iteration during project development same graph (cycle) will repeat again and again. So we can consider single cycle for explanation. This problem is similar to Hamilton cycle problem. If problem can be reduce into Hamilton cycle problem then similar problem can reduce into success percentage problem [7].

Similarly 3-SAT problem can be reduce to success percentage problem and 3-SAT is NP problem so any NP problem can be reduce to success percentage problem ,it proves second condition from NP-complete`s definition.

From above discussion it is clear that success percentage is an NP-complete problem.
4.1.1 Relevant mathematical models.

The algorithm used is:

\[ K = \left( \frac{\text{LOC}}{(C \cdot t^{4/3})} \right) \cdot 3 \]

\( K \) is the total life cycle effort in working years, \( t \) is development and the \( C \) is the technology constant, combining the effect of using tools, languages, methodology and quality assurance (QA). Time in years. The value of technology constant varies from 610 to 57314. For easy, experienced projects technology constant is high. [82]. It is important to consider both the cost and the efforts taken by the developer of the system to design a software product most of the time the effort and the cost incurred in the development of the software project is neglected and hence it’s difficult to calculate the estimates which are critical factors governing the software industry today the COnstructive COst MOdel is the most important model to calculate these factors of the software engineering project the COnstructive COst Model 2 has the best value outcomes for estimating both the factors which influence the software engineering project. In the project the COnstructive COst Model 2 is widely used for the calculation of man hours and efforts taken by the individual to complete the software project the COnstructive COst Model 2 gives highly accurate results when input are fed to the system under development one of the key factors governing the system cost and the efforts is the resource involvement in the design and development of the software project also the amount of time spent by the governing bodies for the estimation of the effort taken by the staff who are involved in the designing and
development of the project. All the calculations done are exact and are computed using the COnstructive COst Model 2 of software engineering process. It's important to concentrate on the development of the product and keep the team members working in concatenation with the goals of the software project most of the factors influencing the efforts taken by the stakeholder involvement in the project development the lines of code designed by the developer of the software system should also be computed. The goals of the development is to measure the amount of hard work which was required while designing the software project. So it's important to measure the function point of the project by calculating various factors involved in determining the function point of the software project the cost and other complex features will help to perform more effectively in the software development its important to calculate each individual effort and the cost included in the project with the help of COnstructive COst Model 2 its various input parameters it will help in determining the relevance of the effort and the cost incurred in the development of the software product all the individuals should be distributed with adequate amount of work and the cost of development also be distributed its also important to determine the modularization cost of the system so that developers who are putting their efforts can easily be identified and the cost incurred in the design and development of the modules can also be computed all the constraints are important in the COnstructive COst Model 2.

Mathematical model for Functional Point Analysis:

Function point use to calculate estimation of project.

There are two method for estimation of project using functional point:

i) direct method
ii) derived method.

We are using deriver method for calculating estimation of product using function point analysis.

These component can be explain by using below figure

Figure 4.6 System Components

Explanation of these components is as follow,

**Internal Logical File (ILF)**

The Internal logical files are the files on which all the logical values related to the software project are stored all the data that is stroed is maintained from the external input files that access the data and feed it as the input the ILF.

The External interface file holds all the logically related data of the software project.
The data is maintained outside the application boundary and the logical interface is build to connect this data using the external interface that links to the logical values of the data that is valuable to the application.

In this the data is accessed from inside to outside of the application all the links to the logical data are from inside of the application to the internal interface logic of the system. The data that is flowing in the system will come from any user interface screen or any other software system can feed data as input to the system. This data will that in flowing into the system is either the business information or the control information

The external output files are basically used to add logical data this data actually flows from the output to the input so all the external interfaces can send data to the system and the system maintains a logical link of the external outputs. Information that passes from inside to outside of the application which updates or modifies the internal logical file.

The information that flows from the inside logic to the outside logic and which does not affect any logical file of the software system.
4.2 UML Diagrams

1. Use case Diagram

![Use case Diagram]

Figure 4.7 Use case Diagram
2. Activity Diagram

Figure 4.8 Activity Diagram
3. Sequence Diagram

Figure 4.9 Sequence Diagram
4. Timing Diagram

Figure 4.10 Timing Diagram