**SYSTEM DESIGN**

**CHAPTER 5**

**SYSTEM DESIGN**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.

**5.1 Logical Design**

Design for WebApps encompasses technical and non-technical activities. The look and feel of content is developed as part of graphic design; the aesthetic layout of the user interface is created as part of interface design; and the technical structure of the WebApp is modeled as part of architectural and navigational design.

This argues that a Web engineer must design an interface so that it answers three primary questions for the end-user:

* + 1. Where am I? – The interface should provide an indication of the WebApp has been accessed and inform the user of her location in the content.
    2. What can I do now? – The interface should always help the user understand his current options- what functions are available, what links are live, what content is relevant.
    3. Where have I been; where am I going? – The interface must facilitate navigation. Hence it must provide a ―map‖ of where the user has been and what paths may be taken to move elsewhere in the WebApp.

**5.2 Design Goal**

The following are the design goals that are applicable to virtually every WebApp regardlessof application domain, size, or complexity.

1. Simplicity
2. Consistency
3. Identity
4. Visual appeal
5. Compatibility

Design leads to a model that contains the appropriate mix of aesthetics, content, and

technology. The mix will vary depending upon the nature of the WebApp, and as a consequence the design activities that are emphasized will also vary.

**5.3 The Activity of the Design Process**

1. Interface design-describes the structure and organization of the user interface. Includes a representation of screen layout, a definition of the modes of interaction, and a description ofnavigation mechanisms. Interface Control mechanisms- to implement navigation options, the designer selects form one of a number of interaction mechanism.
   1. Navigation menus
   2. Graphic icons
   3. Graphic images

Interface Design work flow- the work flow begins with the identification of user, task, and environmental requirements. Once user tasks have been identified, user scenarios are created and analyzed to define a set of interface objects and actions.

1. Aesthetic design-also called graphic design, describes the ―look and feel‖ of the WebApp. Includes color schemes, geometric layout. Text size, font and placement, the use of graphics, and related aesthetic decisions.
2. Content design-defines the layout, structure, and outline for all content that is presented as part of the WebApp. Establishes the relationships between content objects.
3. Navigation design-represents the navigational flow between contents objects and for all WebApp functions.
   1. Content architecture, focuses on the manner in which content objects and structured for

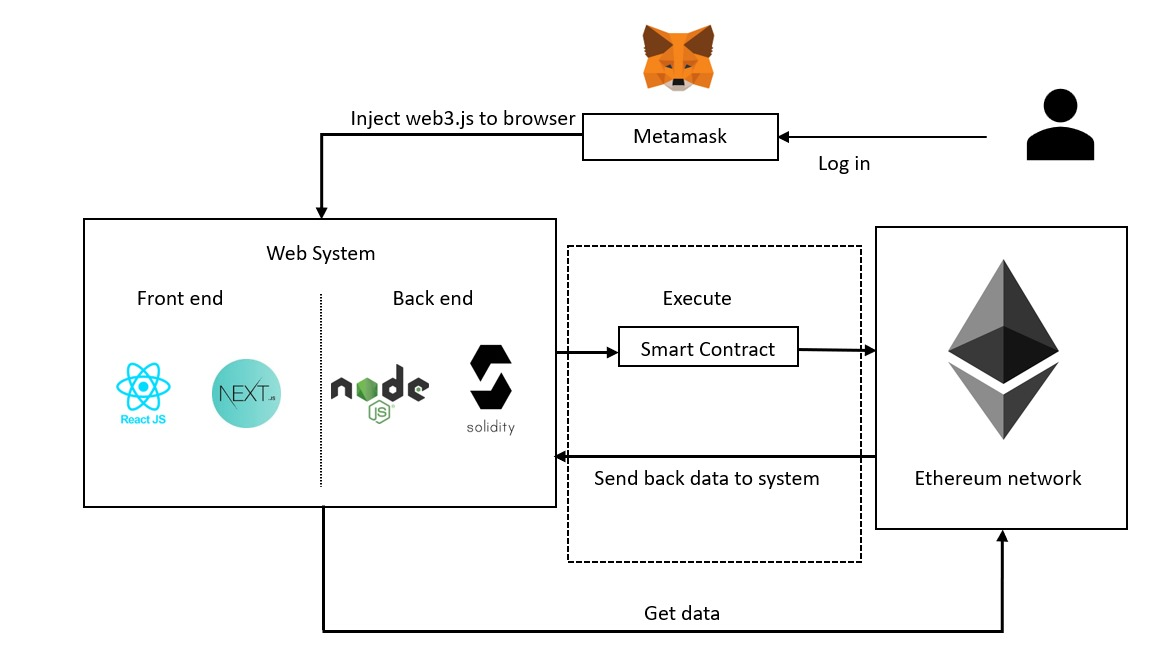
presentation and navigation.

b. WebApp architecture, addresses the manner in which the application is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. WebApp architecture is defined within the context of the development environment in which the application is to be implemented.

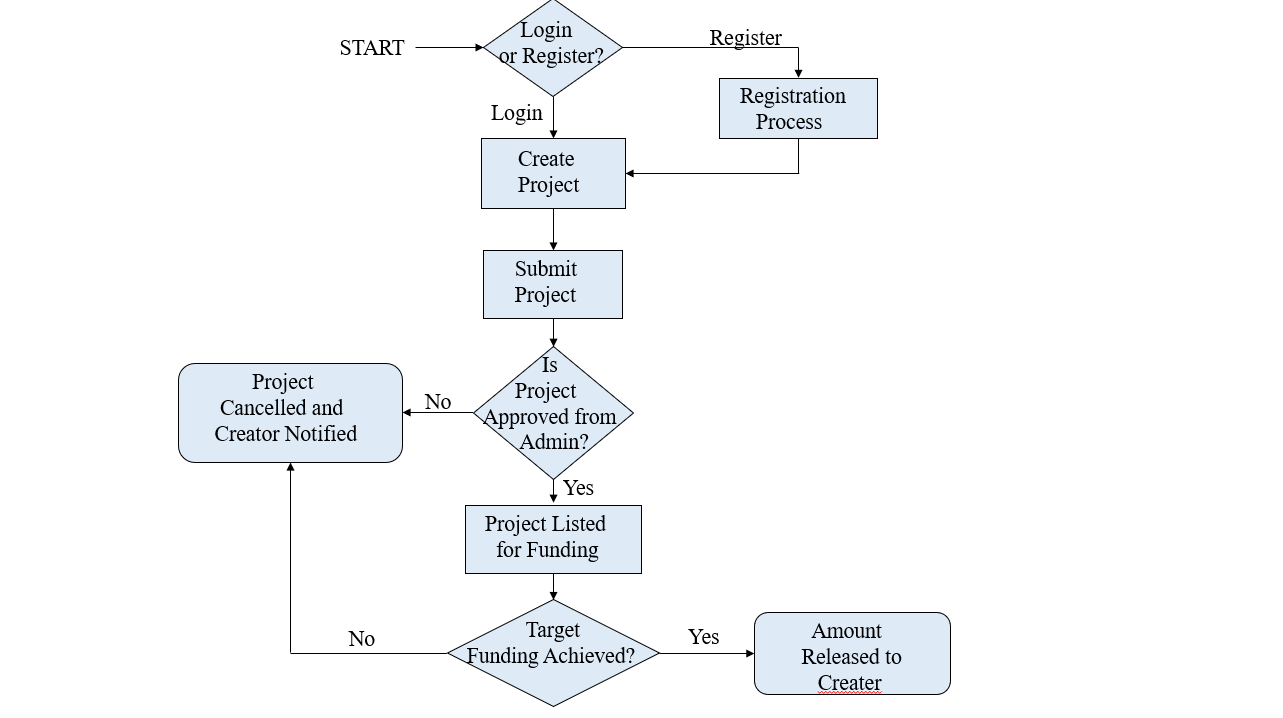
**5.4 Architectural Design**

System architecture is a conceptual model that defines the structure and behavior of the

system. It comprises of the system components and the relationships describing how they work

together to implement the overall system

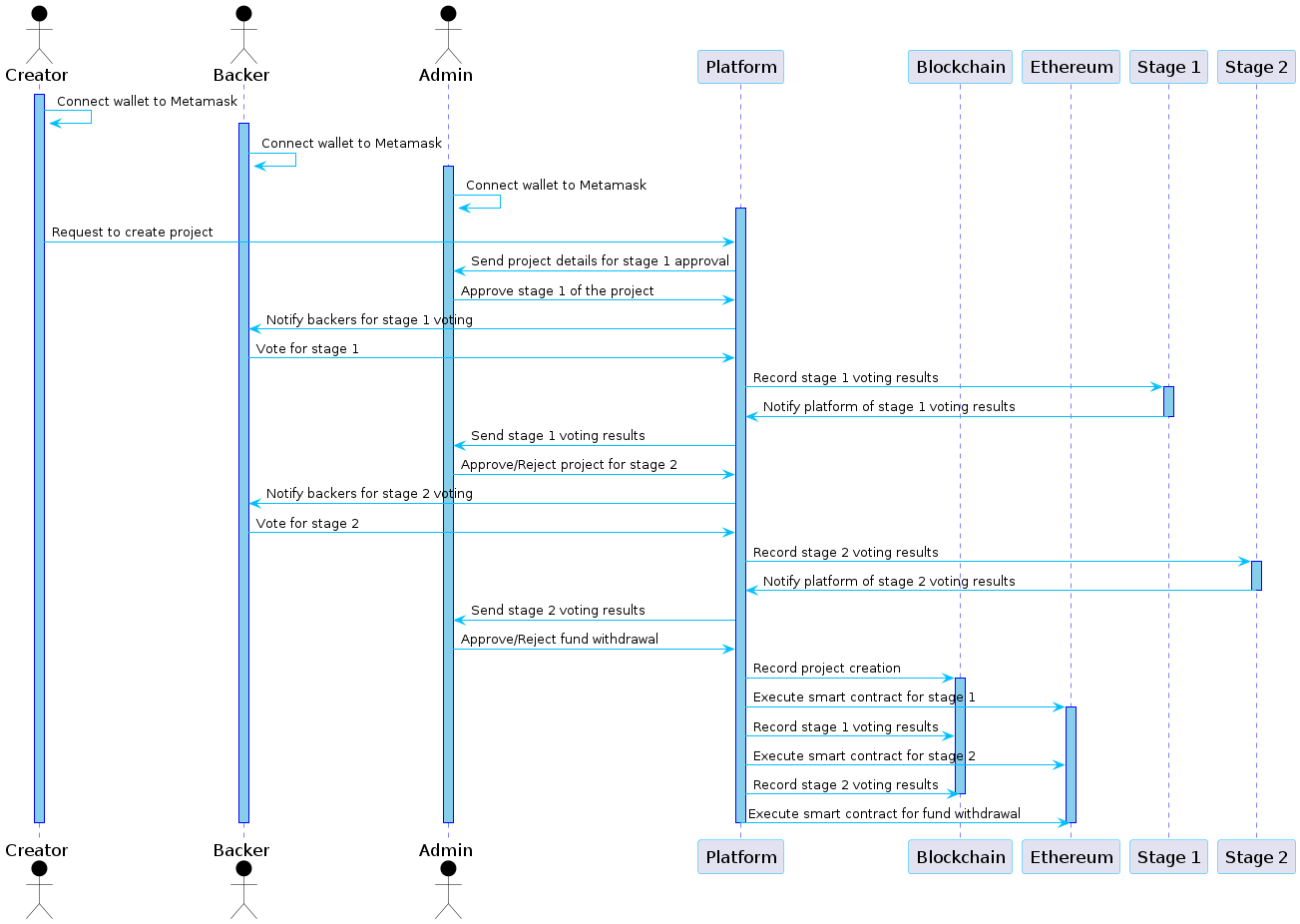
**Fig 5.1 System Architecture**

**5.5 Flowchart**

**Fig 5.2 Flowchart**

For the crowdfunding project utilizing blockchain and Ethereum, the flowchart provides a comprehensive depiction of the operational process. It begins with the creator initializing the crowdfunding process by creating a project on the platform. Upon submission, the admin reviews the project details and grants approval for the initial funding stage. Backers, upon accessing the platform, are presented with the approved projects and can contribute funds to those they support. The platform diligently records all contributions and updates the project's funding status accordingly. Subsequently, the admin evaluates the project's progress and determines whether it advances to the second funding stage. If endorsed, backers are prompted to vote on the project's viability for continued funding. The platform records these voting outcomes and forwards them to the admin for review. Upon admin approval, the project enters the fund withdrawal phase, where smart contracts executed on the Ethereum network facilitate the transfer of funds. All transactions and project activities are securely recorded on the blockchain, ensuring transparency and accountability throughout the crowdfunding process.

**5.6 Sequence Diagram**



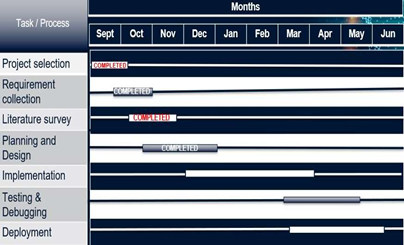
**Fig 5.3 Sequence diagram**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

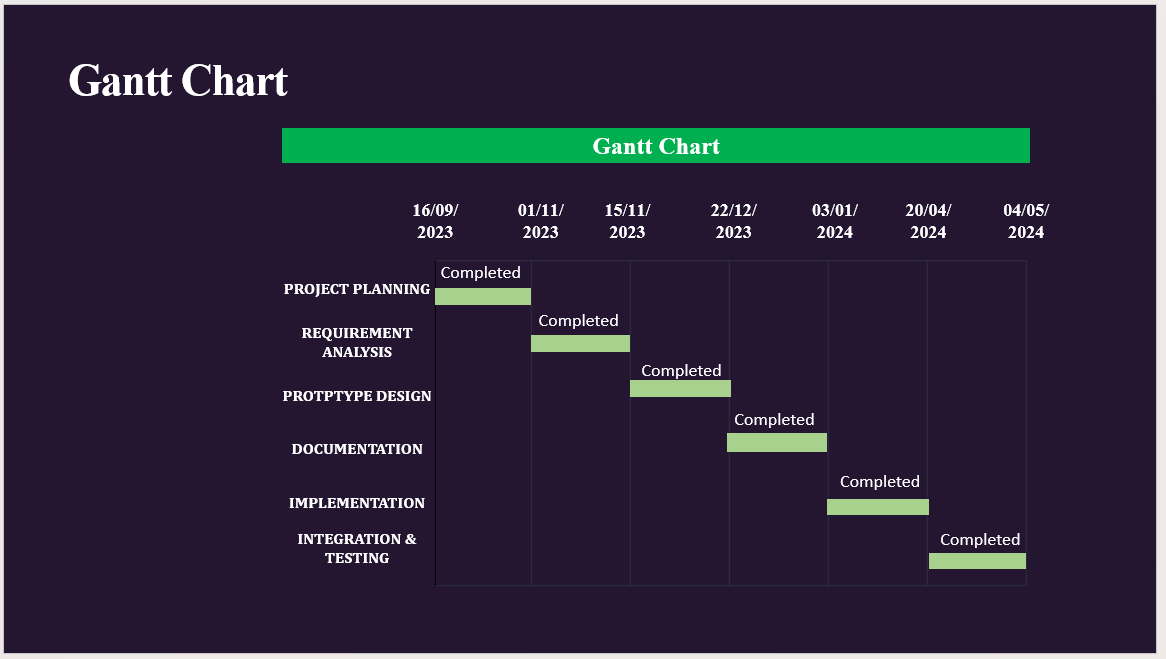
A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

**5.7 Gannt Chart**

The Gantt chart has been prepared for 1,2 phase of the project. The single date tasks have been depicted as milestones and are shown along the horizontal line. Similarly, the tasks which take multiple days to accomplish are shown in the vertical line. The Gantt chart has been shown below:

**5.7.1 Gannt Chart For Phase-1**

**Fig 5.4 Gannt Chart Phase-1**

**5.7.2 Gannt Chart For Phase-2**

**Fig 5.5 Gannt Chart Phase-2**

**5.8 Life Cycle Model**

The agile model is applied for the software development process in our project. Agile means relating to or denoting a method of project management, used especially for software development, that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans. Phases involved in our project:

1. Recognizing hardware and software requirements
2. Developing and working on architecture diagram, class diagram, use case design
3. developing the software required for execution and working on GUI
4. Testing with various media file and checking if model classifies them correctly

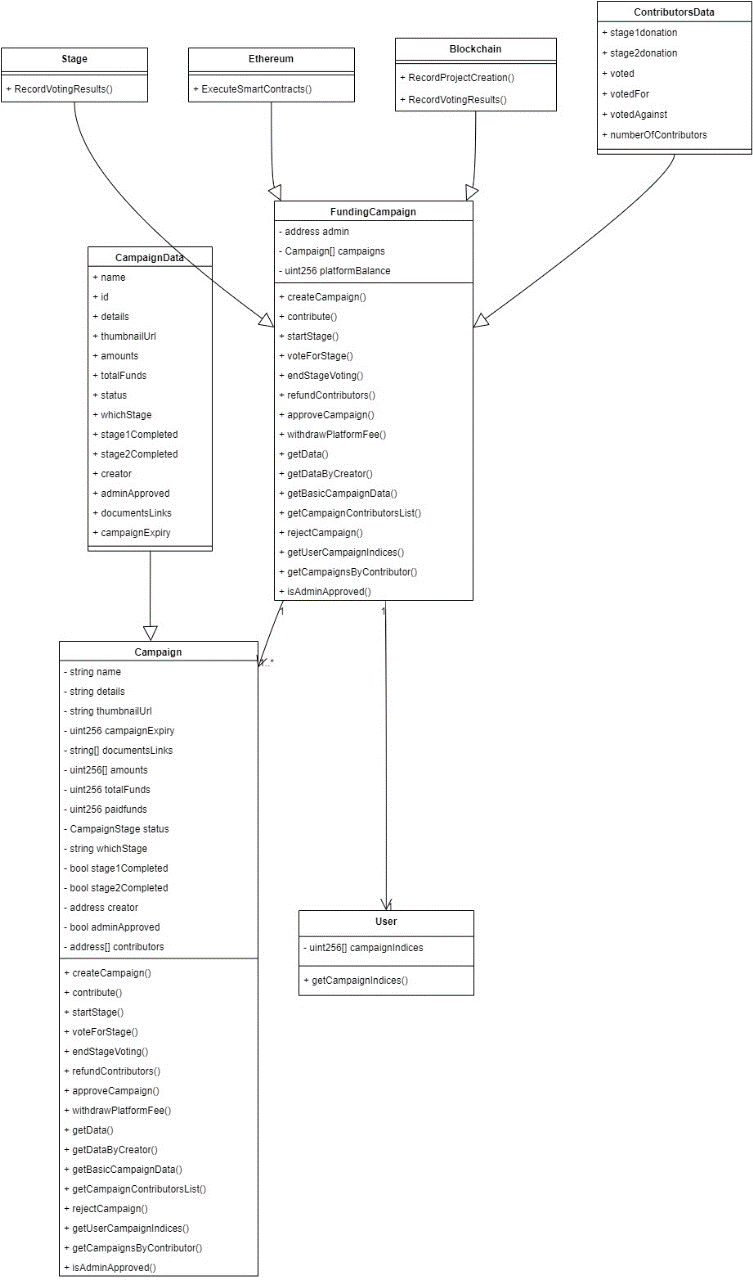
5. Deployment on Windows OS and using VS Code.

6.Review and test of whether accuracy desired has been achieved. Repetitively, this

process iterates to improve accuracy of classification.



**Fig 5.6 Agile Model**

** 5.9 Class Design**

**Fig 5.7 Class Design**

In software engineering, a class diagram in the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) [(UML)](https://en.wikipedia.org/wiki/Unified_Modeling_Language) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

**Purpose of Class Diagram**

* + - 1. Shows static structure of classifiers in a system
      2. Diagram provides basic notation for other structure diagrams prescribed byUML.
      3. Helpful for developers and other team members too
      4. Business Analysts can use class diagrams to model systems from business perspective.

**5.10 System Design Summary**

This chapter shows the detailed design of the architecture, components, modules, interfaces, and data for the proposed system to satisfy specified requirements. It shows various standard UML diagrams that are needed to design the system. It provides a visualization of how the data will flow among various components of the system.