**Chapter 1: Introduction**

* 1. **Introduction of system**

Nowadays, people have been conscious about their life & ecosystem. So, they would prefer for single child only. Some parents would not get child due to some issues. So, they adopt child from orphanages.

It would run with funds and NGO’s / INGO’s. They would bear all expenditures. For this, a web application (Orphanage Management System) would be developed. It keeps records of all incomes and expenses for transparency. Here, sources and amounts of incomes & expenses, proofs of expenses, number of children with their details like age, name, height, color, gender, etc. would be seen by world. Similarly, number of adopted children and details of adopted family like address, phone number, etc. would be kept.

* 1. **Background of system**

It’s a totally new project. Here, users could sign in with login form. If they are new, they could register their account with registration form. After that they could see the detailed information about the Orphanages, Orphanage Management System.

If they wish to adopt child, they could see the details of the children such as age, name, color, height, weight, etc. According to their choices and comfort, they could adopt number of children. And for that parents should give their full details to the organization like contact numbers, addresses, etc. Similarly, users could easily see all the information.

* 1. **Justification**

It has been difficult to believe on people on any aspects. So, it is far away to donate funds blindly anywhere. Similarly, no one likes to adopt child haphazardly. Similarly, some people are frauds and they adopt child for their personal benefits like they compel them to work inside the home or on any other places like hotels. Not only could that government unable to get data about the number of adopted children and number of age group of orphans.

Therefore, in order to solve all such problems, this project is going to be come to the practical.

**Chapter 2: Scope**

Orphanage Management System is a platform which provides better services to the entire world. Here, we could see number of different age group of orphans with their all details and could easily adopt them for their bright future. We could also see the number of adopted children and their parents as well as we could also donate some funds to Orphanage Management System.

**2.1 Aims**

1. To build a website where people could adopt orphans as per their wishes and helps nation to produce skilled human resources.
2. To eradicate discrimination between rich & poor, gender, cast, color, etc. by providing a platform where every people are free to accept any orphans keeping discrimination aside.
3. To provide happiness and opportunity to be parents to them who suffer from infertility. Similarly, orphans also get someone to call their family.

**2.2 Objectives**

1. To keep all records of children and charity in a scientific and systematic manner.
2. To keep also detailed information about parents that have been adopted children till now and number of adopted children.
3. To release transparent budgets, funds & expenses with legal proof yearly.
4. To keep records of those who have given funds to Orphanage Management System either that amount or assets are small or large like people, NGO’s, etc.
5. For testing the project.

**2.3 Features**

1. **Users registration and login to the system**

New users could register to the system through registration form & are notified that they have registered successfully and then could login through login form where username and password are checked in database for validity.

1. **Users could see details of children and can adopt children as per their wishes.**
2. **Admin could add number of children**

Admin could easily make changes to the system.

1. **Users could easily make fund and donate to Orphanage Management System funds form.**

**2.4 Overview**

Users could not register to the system of ages below 20 and more than 60. Users could not have more than one account on same email addresses and contact numbers.

**Chapter 3: Development Methodology**

A software development methodology is a framework which used to structure, plan, and control the processes of developing an information system. There are different types of methodologies for the development of system project such as Joint Application Development (JAD), Rapid Application Development (RAD), Scrum, Spiral, Systems Development Life Cycle (SDLC), and Waterfall.

**3.1 Methodology Used**

For this project (Orphanage Management System), waterfall model is used because it is simple and easy to understand as well as use. It is also easy to manage due to rigidity of the model. In this model, each phase are analyzed deeply and given equal priority. The steps applied in the waterfall model are as follows:

* **Requirement analysis:**

In this phase, all possible requirements of the system to be developed are captured, analyzed deeply and documented in a requirement specification document.

* **System design:**

Here, the requirement specifications from first phase are studied and thus, system design is prepared.

* **Implementation:**

With inputs from designed system, firstly, system is developed in small programs i.e. units and then are integrated in the next phase.

* **Testing:**

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

* **Deployment:**

When the functional and non-functional testing is completed then the product is deployed in the customer environment or released into the market.

* **Maintenance:**

It’s a truth fact that there are some issues which come up in the client environment. So, in order to fix those issues, patches are released, maintenance is done. It delivers all these changes in the customer environment.



Figure 1: Waterfall model

**3.2 Design Pattern**

In software engineering, design pattern is a general repeatable solution to a commonly occurring problem in software design. It is not a finished design that can be transformed directly into code. It’s a description or template for how to solve a problem that can be used in many different situations.

For this project (Orphanage Management System), MVC (Model View Controller) design pattern is used because it separates domain/application/business (whatever you prefer) logic from the rest of the user interface. It does this by separating the application into three parts: the model, the view, and the controller.

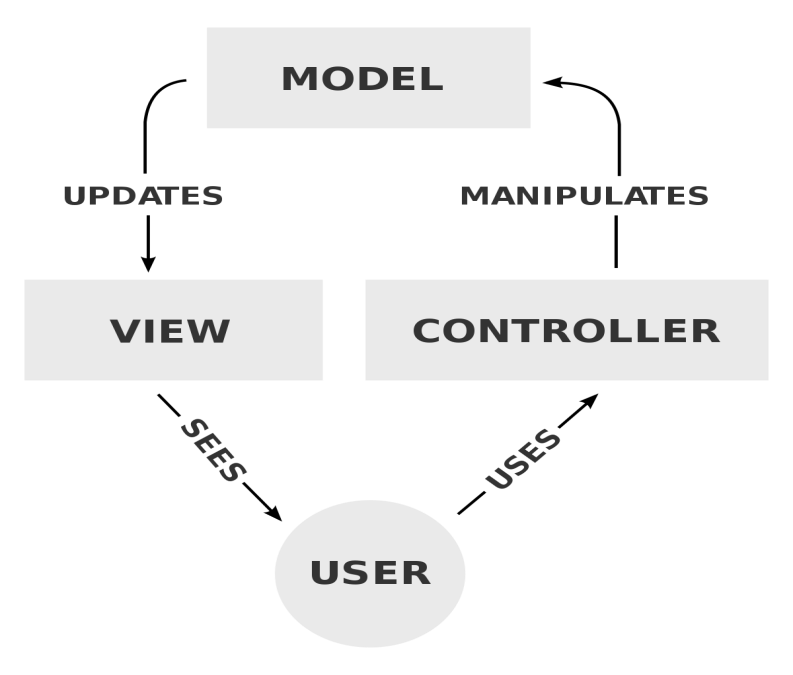


Fig 2: MVC design pattern

* **Model**

It handles all the data related to logic that users work with. It represents data that is transfer between view and controller.

* **View**

The view handles all UI logic of application.

* **Controller**

Controller acts as transitional between model and view to process incoming requests and logic.

**3.3 System Architecture**

It is a reaction to the conceptual and practical difficulties of depiction and the outline or design of complex systems. It allocates required functionality to equipment’s and programming segments. It is essential since high-level choices influence a task for quite a while. Poor decisions will negatively affect its encouraging down. While developing, there would be use of MVC design patter that stands for the model, view, and controller.

It is three-tier architecture that it is a client-server design in which the functional procedure of reasoning, Data get to the devices, data storing and UI are delivered and kept up as self-sufficient modules on separate stages.

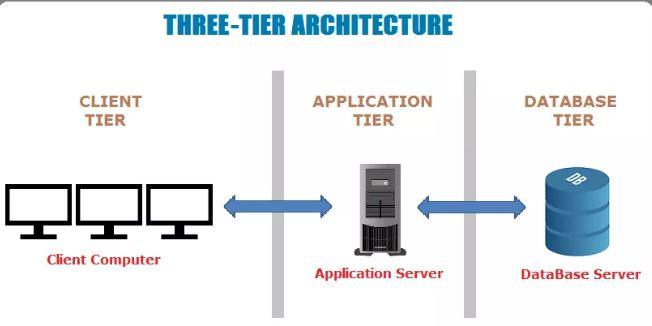


Figure 3: Three-tier architecture

**Chapter 4: Scheduling**

**4.1 WBS (Work Breakdown System)**

It is a process of dividing complex projects to the simpler and manageable tasks. Generally, project managers use this method for simplifying the project execution. In WBS, larger tasks are broken down to manageable portions of work. These portions could be easily supervised and estimated. WBS is not restricted to a specific field when it comes to application. This methodology can be suitable for any type of projects.



Figure 4: WBS structure

**4.2 Milestone**

It is also a process of dividing complex projects to the simpler and manageable tasks within the specific point of times. Project milestones are specific points along a project timeline. They are short term goals that lead to the overall project objective. These milestones act as a major project progress points and indicate your project progress. If milestones are achieved along the way, then only could be moved in the right direction to achieve overall project objectives.

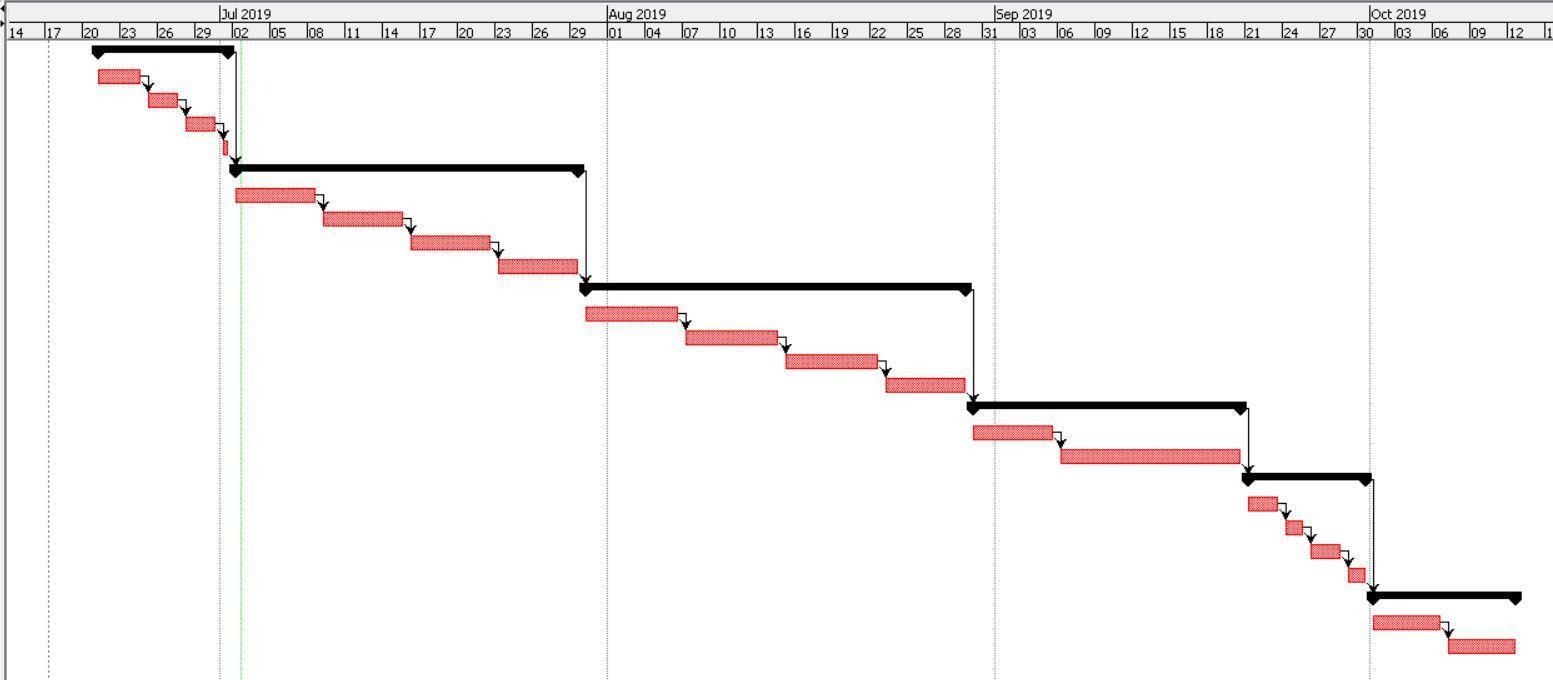
|  |  |  |
| --- | --- | --- |
| Milestones | Date | Days |
| **Project Management**  Risk Management  WBS  Configuration Management  Proposal Submission | June 21- July 1, 2019  June 20- June 24, 2019  June 25- June 27, 2019  June 28- June 30, 2019  July 1, 2019 | 11 days  4 days  3 days  3 days  1 days |
| **Analysis**  Requirements analysis  Use Case  Architecture (Initial Class Diagram)  Analysis Specification | July 2- July 29, 2019  July 2- July 7, 2019  July 8- July 14, 2019  July 15- July 21, 2019  July 22- July 29, 2019 | 28 days  7 days  7 days  7 days  7 days |
| **Design**  Structural Diagram  Behavioral Diagram  UI Design  Database Design (ER, Data Dictionary) | July 30 – Aug 29, 2019  July 30 – Aug 6, 2019  Aug 7 – Aug 14, 2019  Aug 15 – Aug 22, 2019  Aug 23 – Aug 29, 2019 | 31 days  8 days  8 days  8 days  7 days |
| **Implementation**  Building Database  Coding | Aug 30 – Sept 20, 2019  Aug 30 – Sept 5, 2019  Sept 6 – Sept 20, 2019 | 22 days  7 days  15 days |
| **Testing**  Unit Testing  Integration Testing  Black Box Testing  White Box Testing | Sept 21 – Sept 30, 2019  Sept 21 – Sept 23, 2019  Sept 24 – Sept 25, 2019  Sept 26 – Sept 28, 2019  Sept 29 – Sept 30, 2019 | 10 days  3 days  2 days  3 days  2 days |
| **Deployment**  User Training  Final Report | Oct 1 – Oct 12, 2019  Oct 1 – Oct 6, 2019  Oct 7 – Oct 12, 2019 | 12 days  6 days  6 days |

The project has been started on 16th June. Firstly, proposal could be completed on deadline 1st July i.e. within 11 days and then analysis could be preceded. After the completion of analysis portion i.e. 29th July (within 28 days of completion of proposal), system designing could be started and should be completed till 29th August (within 29 days of completion of analysis). Then coding parts should be done within 31 days of completion of designing parts and should be submitted on 20th September. Similarly, after final product, it could be tested and all bugs should be fixed within 22 days of coding portion and should be submitted on 30th September. Finally, the final document should be submitted within 0 days of completion of testing i.e. on 12th October.

**4.3 Gantt Chart**

It is a type of a bar chart that illustrates the project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. The width of horizontal bars in the graph shows duration of each activity. It illustrates the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements constitute the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here. Gantt charts are sometimes equated with bar charts.

Figure 5: Gantt Table

Figure 6: Gantt chart

**Chapter 5: Risks Management**

It is the process of identifying, measuring and controlling threats to an organization's investment and earnings. These threats, or risks, could stalk from a wide variety of sources, including financial hesitation, legal liabilities, plan management errors, accidents and natural disasters. IT security threats and data-related risks, and the risk management strategies to ease them, have become a top priority for digitized companies. As a result, a risk management plan increasingly includes companies' processes for identifying and controlling threats to its digital assets, including exclusive commercial data, a customer's personally identifiable information and intellectual property.

Likelihood and values are given below:

|  |  |
| --- | --- |
| **Likelihood** | **Values** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Consequences and values are given below:

|  |  |
| --- | --- |
| **Consequences** | **values** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

Some of the risks that can affect the project are as following:

**[Impact = Likelihood \* Consequence].**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risks** | **Likelihood** | **Consequences** | **Impact** | **Action needed** |
| Server failures | 2 | 3 | 6 | Data backup must be kept. |
| Error while coding | 3 | 4 | 12 | Coding must be done properly. |
| Low budgets | 2 | 4 | 8 | Sponsors must be arranged and managed. |
| Natural disasters | 2 | 4 | 8 | System backup must be kept. |
| Insufficient resources | 2 | 3 | 6 | All the required resources for the project must be collected. |
| Lots of bugs | 2 | 2 | 4 | Testing must be done. |
| Difficult to operate | 1 | 2 | 2 | System must be made user friendly. |

**Chapter 6: Configuration Management**

It is a system manufacturing process for establishing and maintaining the consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. There are three types configuration management i.e. change management, release management and version management. In change management, changes could be bringing in future. Similarly, how to release project in the market after completion of project is performed in release management. In version management, why are we using the configuration is explained.

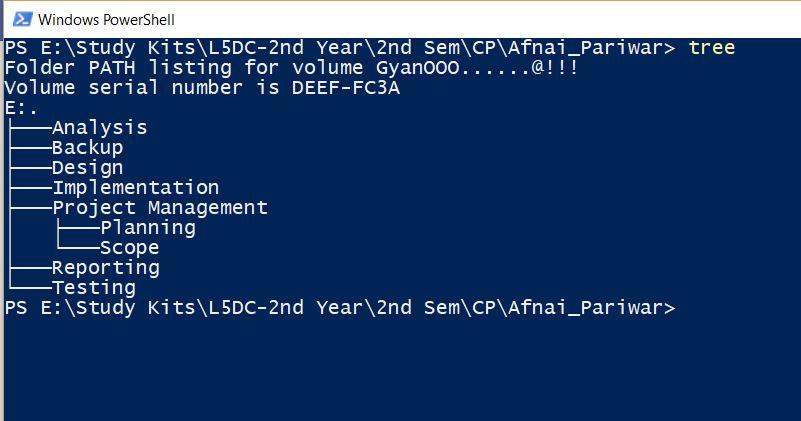


Figure 7: Command prompt screenshot

**Conclusion**

The created website (Orphanage Management System) would not be completed without the use of different website developing tools such as PHP, CSS, HTML, MYSQL, J Query, Java Script, bootstrap, etc. The web-based application is developed totally based on OOA (Object Oriented Analysis). Here, users could register through register form and then could login through login form. The web-based application would be very user friendly and reliable. Admin could easily add and make changes to the system.

**References**

* <https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>
* <https://searchitoperations.techtarget.com/definition/configuration-management-CM>
* <https://www.workbreakdownstructure.com/>
* <https://www.sitepoint.com/model-view-controller-mvc-architecture-rails/>