Table Of Contents

Abstract

The members of the development team in web development must work from several locations. In such situations, collaborating and coordinating at the same time is extremely challenging. To get around these issues, website developers and software engineers employ social media software engineering tools like Github and Source Forge. Real-time editing is well supported by the Integrated development environment (IDE). Our project here comprises not only UI / UX design of the site for all team members at once, but also UI development process using HTML, CSS, SASS, LESS, and JS web development languages as a solution to these challenges that developers face while constructing web sites as a team. Makes it simple by providing room. Syntax Highlighting, Code Complement, Brace Matching, Indentation, team chat, and other features are included in the "Codinoc IDE," which is being developed as an open-source cloud software. Anyone using any operating system, including Windows, Linux, Android, Mac, or IOS, can utilize this IDE. In this document, we'll provide you a quick introduction of the Codinoc IDE development process, including the tools and techniques you'll need.

Keywords: Integrated development environment, Open-Source, Cloud

First Chapter

Introduction

A web-based integrated development platform is referred to as a cloud IDE. Codinoc is a Cloud Based Integrated Development Environment for Web Based Designers.

What exactly is an IDE? Most programmers utilize an Integrated Development Environment (IDE) to create computer software or mobile apps (IDE). An integrated development environment (IDE) is a tool that allows you to write, run, and debug software all in one place.

Everything in this century is connected to the internet and no one can live without it. The majority of the Internet consists of websites and pages. Static and dynamic websites are two types of websites. Dynamic sites have a front end and a rear end, but static sites only have a front end. So, how do you create websites / pages? For that, we will need to use specialized software. Programming languages are first and editors second. There are many languages for web design and development, including JS, PHP, Ruby, HTML (Hypertext Markup Language), CSS, and more. In web application development, anyone can use any text editor, code editor, or integrated development environment (IDE). However, code editors such as Sublime Text, Visual Studio Code, Brackets, JetBrains WebStorm, JetBrains PHP Storm, and several other IDEs also provide a number of useful and powerful features, such as Syntax Highlighting, Code Complement, Indentation, Brace Matching. and so on. Creating a website, on the other hand, can be a daunting task. If you build a website with a team, the problems in the development process will be less. You will also need to use additional technologies such as GitHub and GitLab. Therefore, as a result of this project, we are introducing the "Codonic IDE" to help you easily accomplish your group projects. Because Codonic is an IDE cloud-based program, any team can work with it remotely on any operating system, including Windows, Linux, Mac, iOS, and Android. Only for creating web-based applications such as our Codinoc IDE, web pages / sites, mobile applications and desktop applications that are hosted on the cloud. Finally, our cloud-based IDE allows you to successfully integrate UI development processes into a single application.

In this paper we will provides a brief overview of the development process, tools and techniques required in the Codinoc IDE development process. Compared to the majority of existing web IDEs, the Codinoc IDE provides a new approach to web design, development, and teamwork. The remainder of the paper is structured as follows: The second section provides an overview of the Existing systems and Diagrams. Sections 3 and4 describe the specific development phases and tools of the Codinoc IDE. Last section brings the paper to a conclusion.

Problem Definition

When a software is created by several people together rather than by one person, it can be done very quickly. That is because the development process of the relevant software can be shared with each other. This is an easy task in a company. This is because in a typical software development company, all the computers are interconnected so it can be done easily. But suppose an ordinary team, not a company, does this. There, each other 's computers are not directly connected. Here you have to share the project related to the software created by the team.

Take a look at this example. Imagine that a team is developing a mobile app. Here are the basics: UI Design Process and App Development Process. For the UI design process, you can use online software such as Figma and Lunacy. But think about the app development process. How is it done? The project should be shared with each other.

Software such as Visual Studio provides Team support for this problem. But can everyone use it? Suppose I am a Linux User, another Mac User, another Windows User. Simply put, this is our problem.

Project Objectives

The main objective of our project is to connect a team of web designers in one place and allow web-based applications to be designed together.

Scope of Project

This cloud IDE gives a user-friendly and this will support for languages such as HTML, CSS, Java Script, SASS, SCSS and LESS. It contains a complete package including a Project Templates, Team Collaboration and Chat, Built-In Live Preview, Built-In Storage per Team and Full Featured Text Editor. IDE integrates project files, which you work on and includes full source file version control of source files and it will contain source file auto save option.

Chapter Summery

Codinoc is a Cloud Based Integrated Development Environment for Web Based Designers. In web application development, anyone can use any text editor, code editor, or integrated development environment (IDE).

If build a website with a team, the problems in the development process will be less. So, for creating web-based applications such as our Codinoc IDE, web pages / sites, mobile applications and desktop applications that are hosted on the cloud. As, our cloud-based IDE allows you to successfully integrate UI development processes into a single application. This will provide a brief overview of the development process, tools and techniques required in the Codinoc IDE development process. Compared to the majority of existing web IDEs, the Codinoc IDE provides a new approach to web design, development, and teamwork. IDE integrates project files, which you work on and includes full source file version control of source files and it will contain source file auto save option. The main objective of our project is to connect a team of web designers in one place and allow web-based applications to be designed together.

Second Chapter

Requirement Gathering Techniques

Conduct a brainstorming session:

We first discussed each other's ideas on what an IDE is. We then proceeded to gather information about the IDE from books on the Internet.

Study analogous systems:

Here we have studied the existing text editors such as sublime text, Vim, notepad, ace editor etc. under several sections. We have primarily explored the available features, the technology used, the RAM usage, the language in which they are accessed, and the language used to create them.

Examine suggestions and problem reports:

What we did with this was identify the vulnerabilities in the current hot editors currently in use. We then prepared a report for each of them and then made suggestions to them.

Talk with supporters:

Prior to creating Codinoc, we consulted with people with experience in the field.

Attend Conduct collaborative workshops:

Join workshops organized by the university or other organization affiliated with this department to gain knowledge about this. They also exchanged pdfs, videos, etc. and books written on the subject and shared knowledge.

Existing System

Today we can see many text editors like Vim, Sublime Text, Notepad etc.

They are designed for individual coding that considers the Internet to be a non-essential component. But there are no text editors that can do all the coding at once when creating team software. This is because they are not directly connected to the Internet.

Also, nowadays it is possible to create code as a team in visual studio, but it has problems when using different operating system such as windows, mac, Linux.

In addition, a separate communication medium such as WhatsApp or telegram should be used to exchange information between team members during code creation. This is because existing text editors do not include the chat feature.

Use Case Diagram

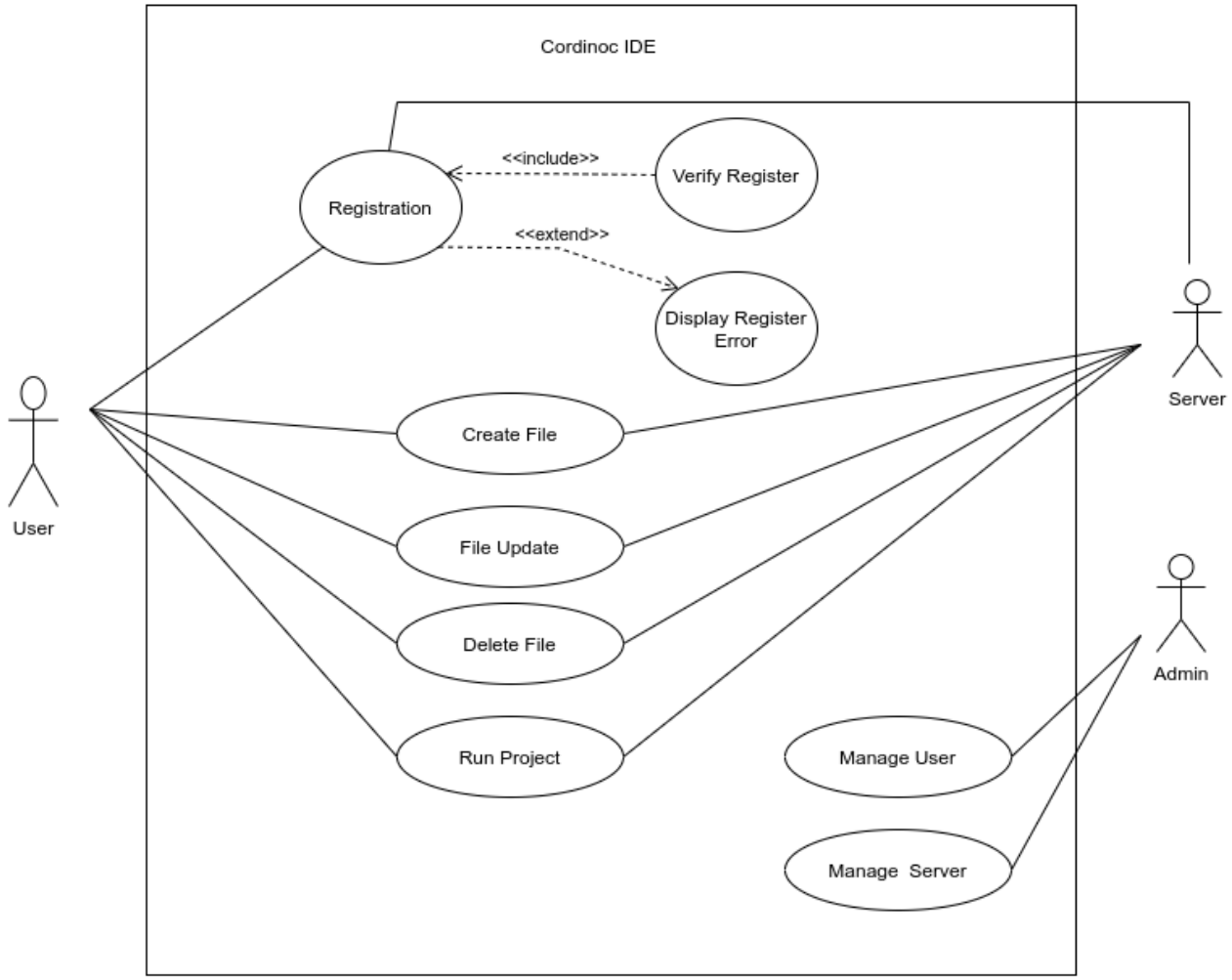


Figure 2.1

ER Diagram

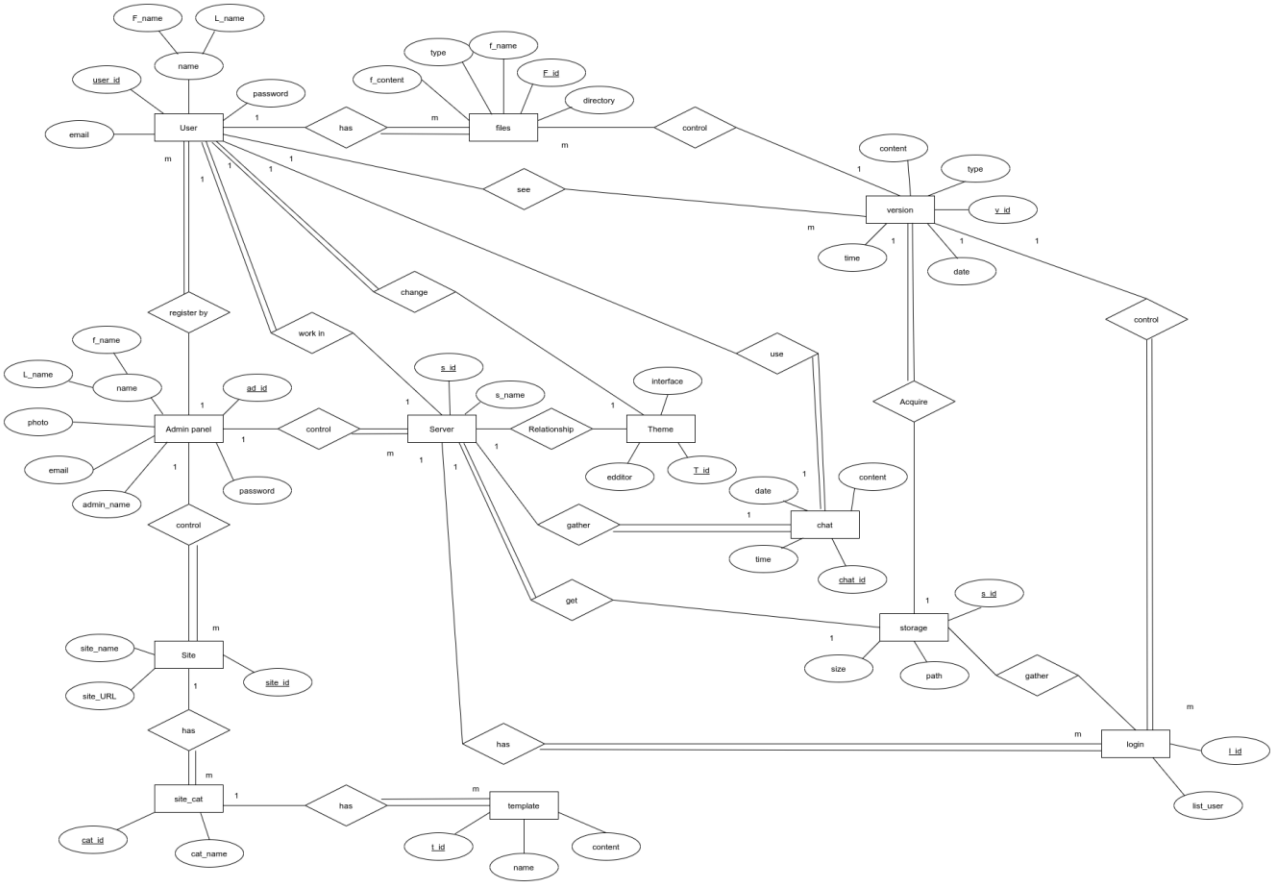


Figure 2.2

Class Diagram

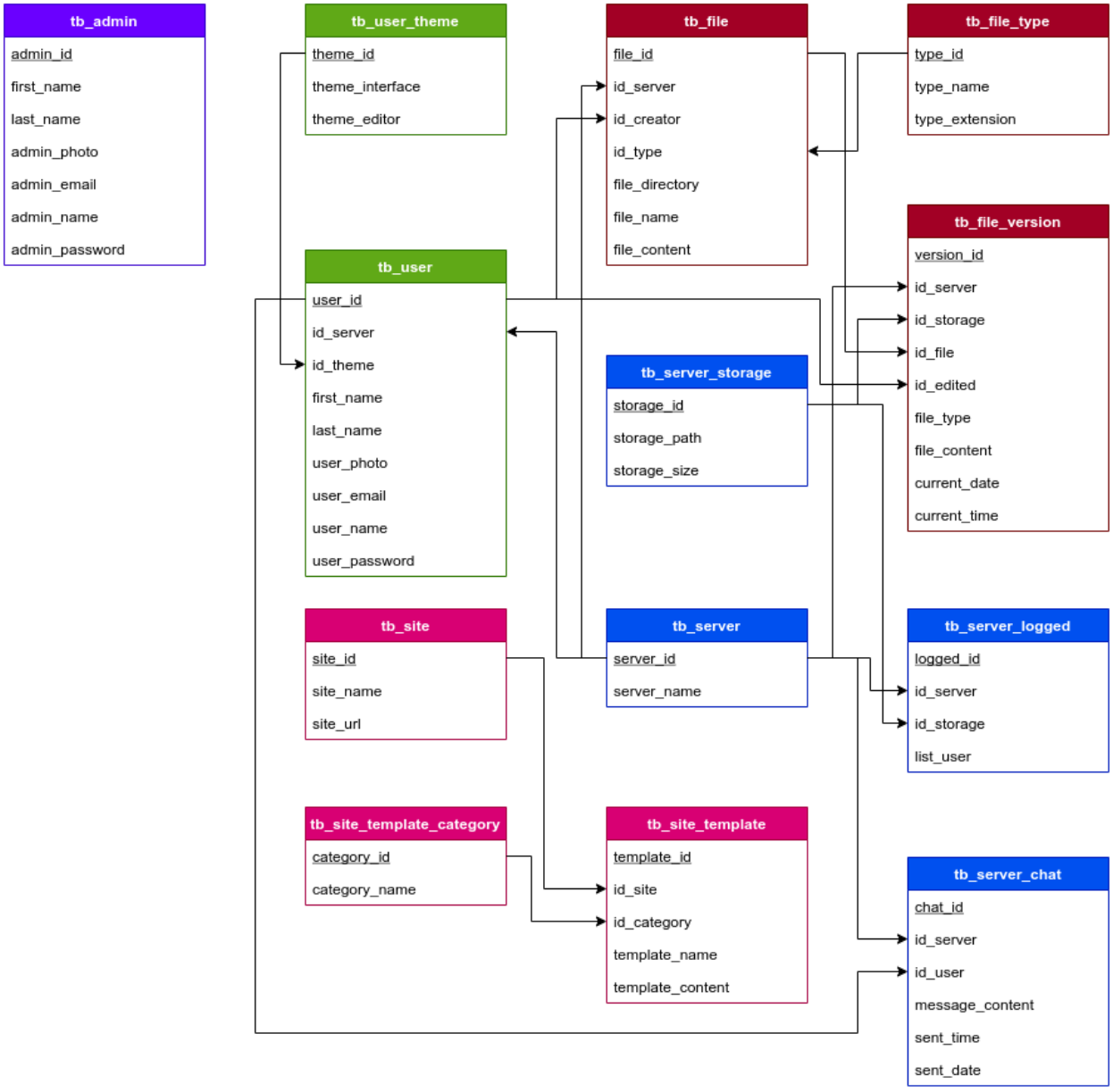


Figure 2.3

High Level Architecture Diagram

The High-level architecture diagram depicts how the Codinoc IDE's main design of the system development is carried out. The overall system is depicted in the high-level architectural diagram. A high-level architecture analysis also allows for the identification of the major components of IDEs and its interfaces.

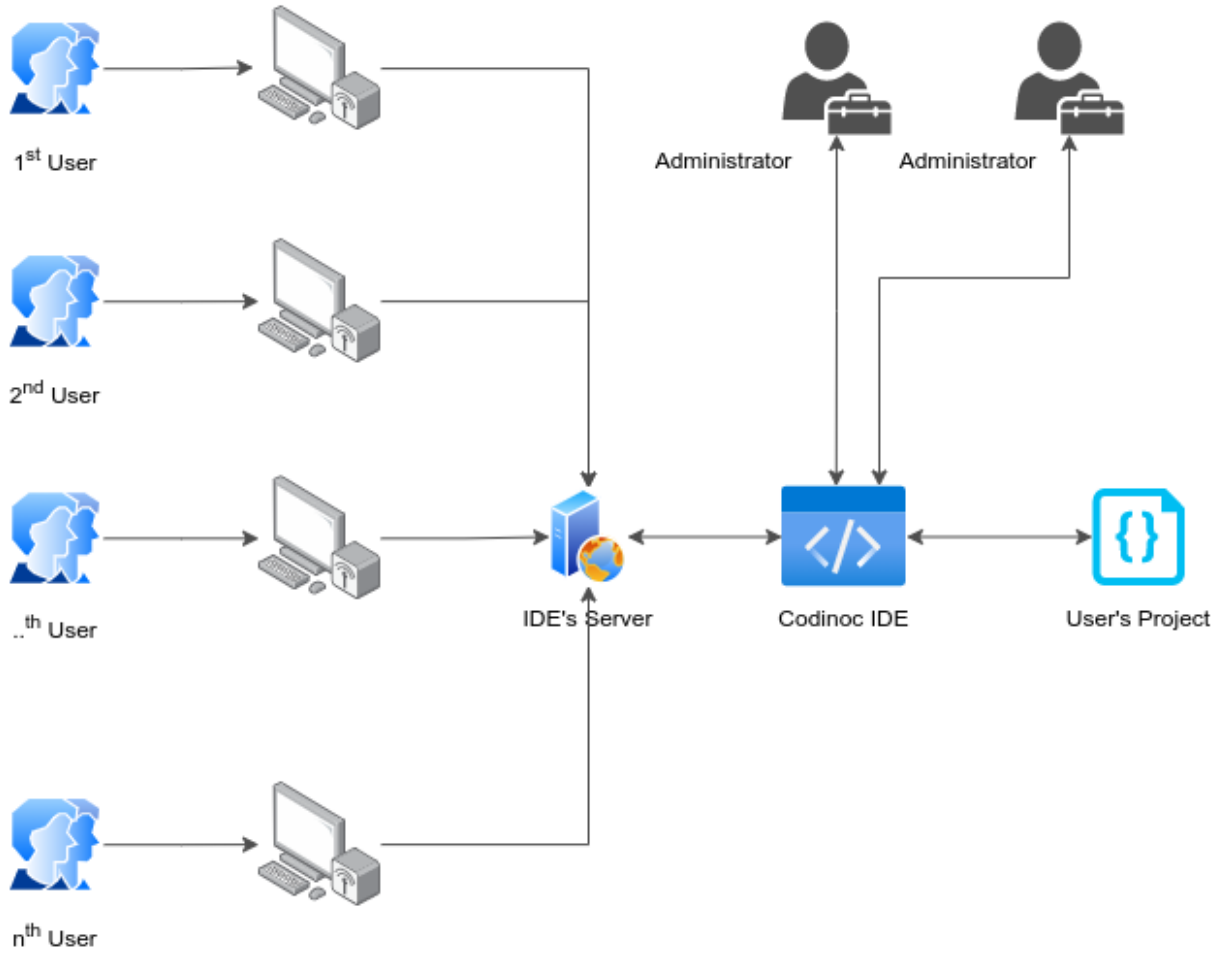


Figure 2.4

Chapter Summery

Conduct a brainstorming session, Study analogous systems, examine suggestions and problem reports, talk with supporters, attend Conduct collaborative workshops are used to collect requirement.

Having to use a separate medium for chatting, and having text editors to write code as a team, but having difficulty using different operating systems have been identified as major weaknesses in existing text editors.

When designing the IDE, several visual cues were created primarily to make the process easier to understand. These are ER Diagrams, Class Diagrams, User Case Diagrams and High-Level Diagram. As a result, it reveals more effective concepts, focuses more on the important tools of the project, and makes it easier for others to understand. The ER format shown here is a visual tool that helps to represent high-level data formats. The ER diagram also shows the relationship between the entity set stored in the DBMS database. The class diagram visualizes how the IDE process works and how it works for team members as well as those who interested in Codinoc project.

Third Chapter

Functional Requirements

Key Features of the IDE

- HTML, CSS and Java Script Support
- Project Templates
- Team Collaboration and Chat
- Built-In Live Preview
- Built-In Storage per Team
- User Interface and Editor Color Schemes
- File and Folder Management
- Source File Auto Save
- Source File Version Control
- Full Featured Text Editor
    - Cut, Copy and Paste
    - Automatic Indent and Out-dent
    - Handles huge documents (four million lines seems to be the limit!)
    - Fully Customizable key bindings including Vim and Emacs modes
    - Search and replace
    - Highlight matching parentheses
    - Toggle between soft tabs and real tabs
    - Displays hidden characters
    - Drag and drop text using the mouse
    - Line wrapping
    - Code folding
    - Multiple cursors and selections
    - Cut, Copy and Paste functionality
    - Live Syntax Checker

User Accounts

- User accounts are password protected and user can change it by themselves.
- Password reminders and resets are handled by the user email

Data Integrity

- At once a week the system sends reminder email
- Reminding notification can be sent in case of any coding exceeding the time limit.

Security

- If the user wants, he can show his code to others, at that time public has read-only access via Codinoc.

- Here the team with the user has the ability to change, read, update, or delete their code. Accessed members will also be able to change the theme.

Non-Functional Requirements

Usability

User will interact with IDE to achieve required goals effectively and efficiently. Can work as team have option to chat with the team as well. Easy to use.

Legal or Regulatory Requirements

- Data privacy & collection
- Copyright requirements
- Data security measures

These Legal requirements we need to basically focus on while we are developing IDE.

Reliability

Before we deploy our cloud IDE, we make sure that are thoroughly tested against a variety of real-world scenarios. This helps to ensure that they are reliable and will meet customer expectations.

User can access from any device at any time from any location. There will be no interruptions or downtime. User connection is secure.

User will be able to perform the tasks need to complete the work.

Performance

As a performance improving feature, we decide to limit 10 users for a team. It will help to improve the quality of the product.

Hardware Requirements

There is no official minimum for the hardware requirements that are needed to host the IDE it will depend on the situation. To make the best estimation of the hardware specifications for the web and database server where following questions should need to be fulfill:

- Number of users(teams) per day/month
- Maximum number of Teams
- Projects per teams at once
- Size and complexity of the projects
- Number of templates
- Size of the database

These are only those questions that should be considered first of all to make a more precise estimation regarding the hardware requirements that are needed for hosting and manage the Codinoc IDE.

The higher requirements, the higher hardware specifications of the web and database servers are can be expected as below.

Minimum Requirements

- Memory: 4GB RAM
- Processor: 1.9 gigahertz (GHz) x86- or x64-bit dual core processor

Recommended Requirements

- Memory: 6GB RAM or more
- Processor: 3.3 gigahertz (GHz) or faster 64-bit dual core processor

Running model-driven apps on less than the recommended requirements can result in poor performance. In addition, you can get satisfactory performance by running a system that uses a different hardware configuration than the one published here. B. Systems with modern quad-core processors, lower clock speeds and more RAM.

Chapter Summery

When considering Functional Requirement, we have considered key features of IDE, user accounts, data integrity and security.

Also, as a Non-Functional Requirement, our focus was on usability, legal or regulatory requirements, reliability and performance.

Number of users (teams) per day / month, Maximum number of Teams, Projects per teams at once, Size and complexity of the projects, Number of templates, Size of the database Regarding the hardware requirements for hosting and managing this CODINOC IDE We wanted to get a more accurate estimate.

4GB RAM and a 1.9 gigahertz (GHz) x86- or x64-bit dual core processor will suffice, but 6GB RAM or more and a 3.3 gigahertz (GHz) or faster 64-bit dual core processor are recommended.

Fourth Chapter

Development Methodology

Language sites such as HTML, CSS, and JS require three simple steps to write software, mainly for web design:

- Edit the code
- Compile the code
- Run the code

You will need a text editor and a compiler for this. For this, basically give a feature bonus in the cloud IDE for work as a team.

As indicated IDE there are distinct functions: for design modification: IDE design and IDE development. This can be accomplished using the Software Development Lifecycle (SDLC).

The Software Development Lifecycle (SDLC) is a well-structured phase stream that enables you to produce advanced software rapidly and easily for a range of purposes.

SDLC Methodology mainly focuses at the following software development phases:

- Analyze requirements
- Planning
- Design Software Model
- Software development
- Testing
- Maintenance

The waterfall model, the spiral model, the Agile model, and the V-shaped model are examples of software development life cycles (SDLC). The Agile model is the one that is most commonly utilized for IDE development.

Of these, the Agile model was primarily used for Codinoc IDE development as it is an agile software development process that, when properly implemented, allows the team to drastically improve the quality of the software in each release. Also, team members are able to adapt quickly. Furthermore, the Agile model is more appropriate because the Codinoc IDE project's UI DEVELOPMENT and DATABASE DEVELOPMENT processes occur at the same time.

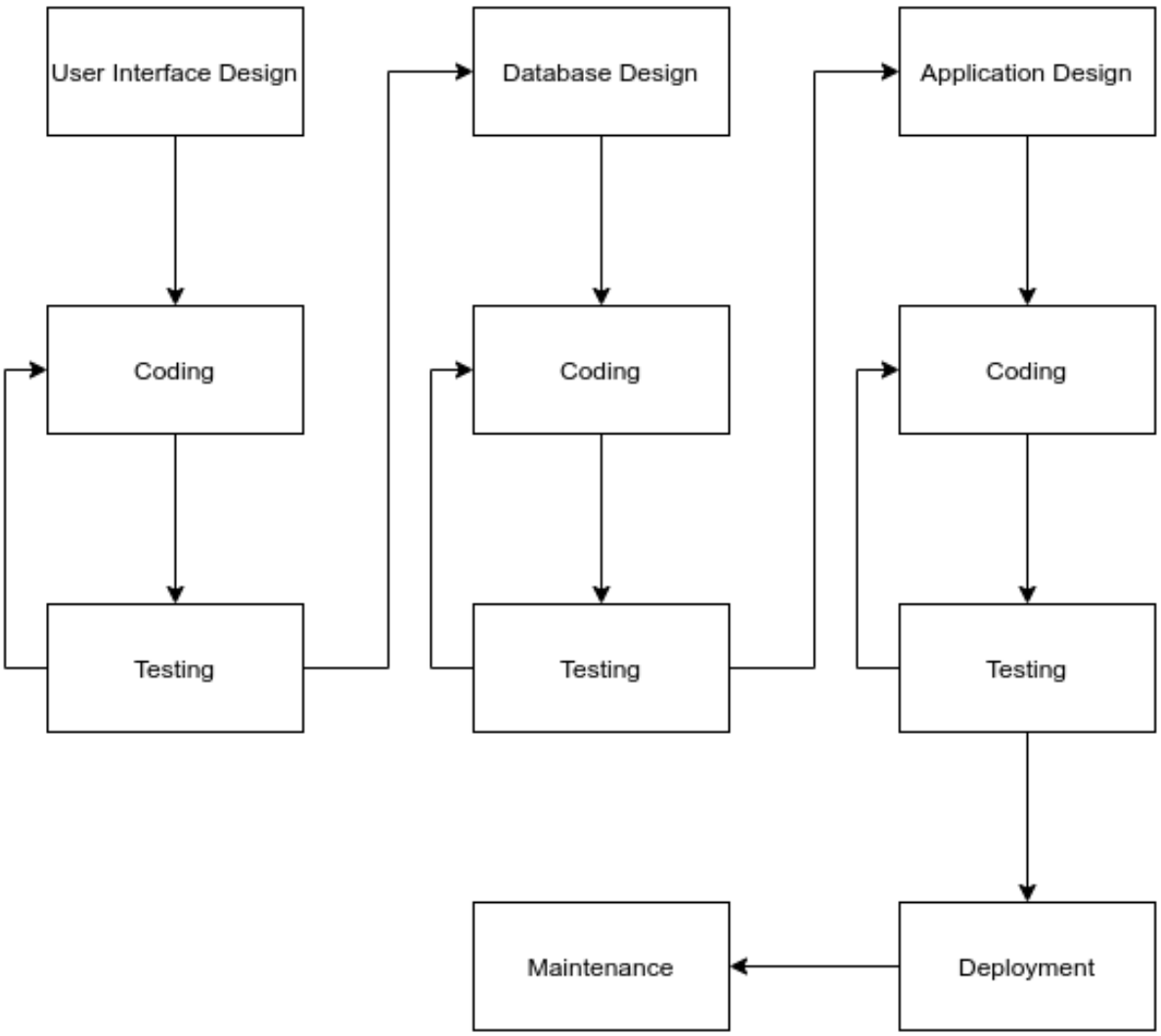


Figure 4.1

Supervised project monitoring using the Agile model is primarily done by JetBrains YouTrack and GitHub.

What made us choose JETBRAINS YOUTRACK?

Simplicity of use of JetBrains YouTrack software. A project management application that helps customize processes and generate outstanding results. Tracking projects and tasks, having a knowledge base, using fast boards, planning Sprint and release issues, working with report and dashboard and having a knowledge base, being able to share and work with the dashboard, being able to use the free version.

Programming Language and Tools

The Codinoc IDE was built on Windows 10/11 and the Fedora Workstation Development Platform, and it was hosted on the Fedora Server Platform. The decision to host on Fedora Server was made for the following reasons:

- Fedora is incredibly fast. Most boots are completed in few seconds, and some are completed in much less time
- Excellent Graphics
- Updates are carried out automatically
- Cloud host
- Flexibility at the entrepreneur level

We approved MariaDB when we created a database system, which is a key component of the IDE. The Team members have chosen MariaDB Because Mint has previously been worked on it and every member can understands it and new features to Maria DB can be easily added.

Figma and DbSchema are the primary software tools used in IDE design. We've already designed the user interface using Figma.

IDE interface is going to build with libraries such as ACE Code Editor and Font Awesome, and theme icons are used.

Here, Go lang is used to create the backend of the IDE, Javascript is used to develop the front end, HTML is used for IDE Theme design and also CSS and SCSS are used for theme styling.

Go language was chosen as the appropriate language for the development of this project for the following reasons.

- It is a simple and fast language.
- The Go lang originally uses a clear syntax structure throughout the encoding, making code manipulation flexible and simple.
- It comes with a text-based approach, making it easy for team members to Study Go in just a few days and save you much time understanding its coding.
- OOP's does not necessarily require a "class" in Go because Go has provided "structs" instead of classes, so the number of properties and methods can be defined using Structs.
- Integral Concurrent Model.
- Trash collector.
- Static Language.

Admin LTE Bootstrap is used to design the Administrator dashboard of the IDE and also in this IDE, The Git plug-in is used to version control the source code.

Third Party Libraries

ACE Editor

Ace editor is designed using JavaScript. Because of its ease with which it can be embedded in any web page and JavaScript application, we decided to use it as a 3rd party library to create Codinoc.

- Displays hidden characters
- Drag and drop text using the mouse
- Line wrapping
- Code folding
- Handles huge documents
- Automatic indent and outdent
- An optional command lines
- Syntax highlighting

These became our favorite features when choosing ace editor.

Ace is also a community project. And while there are no open sources here, it is very simple and friendly for all types of projects. Such factors also motivated us to choose this.

Font Awesome

It's designed to allow the user to place text icons anywhere using a style prefix and icon name, and to take icons' features and make them appear naturally with text. In choosing this it became our reason for interest.

Font Awesome icons automatically inherit CSS size and color so it blends in with the text line wherever it is placed. So, when we use those icons, we do not need to work on them unnecessarily.

Considering the reference to additional design options.

While the basic method of pointing an icon goes straight ahead with simplicity and optimism, we found that it provides a support-level design for things like icon sizing, icon sorting and listing, as well as rotating and translating icons. We also recommend this for CODINOC design, as it allows us to create new designs using our own CSS code, using icons of our choice.

Chapter Summery

After considering the above, the Agile model was primarily used for the development of the Codinoc IDE. Also, project monitoring using Agile mode is primarily done by JetBrains YouTrack and GitHub.

Codinoc builds on Windows 10/11 and the Fedora Workstation Development Platform. It is hosted on the Fedora Server Platform. We also have to create our own database system on MariaDB.

Figma has already begun creating the user interface for Codinoc. dB Schema is also one of the basic software tools that we are going to use.

IDE interface is going to build with libraries such as ACE Code Editor and Font Awesome, and theme icons are used.

It also uses the basic GO language, java script, HTML, CSS and SCSS.

References

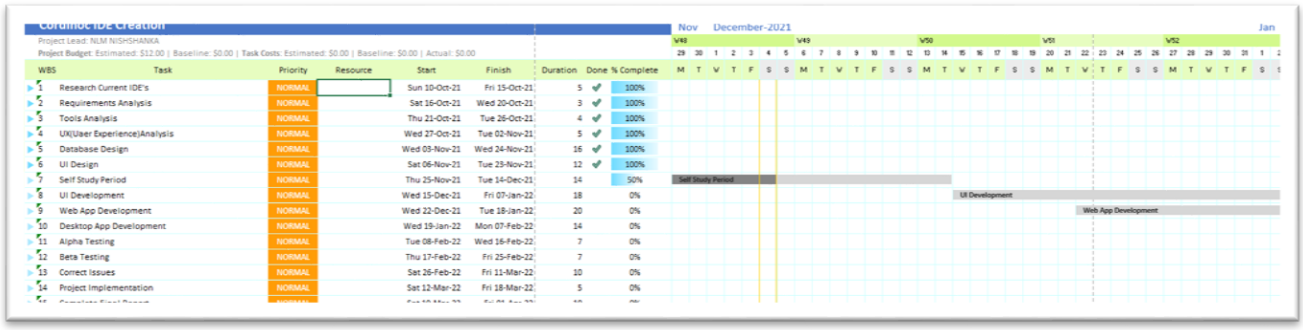
- Mbp.ms.gov. 2022. [online] Available at: <https://www.mbp.ms.gov/Documents/PMP%20NextGen%20Software.pdf >[Accessed 1 January 2022].
- Sites.nationalacademies.org. 2022. [online] Available at: <https://sites.nationalacademies.org/cs/groups/pgasite/documents/webpage/pga_179129.pdf> [Accessed 1 January 2022].
- Ijser.org. 2022. [online] Available at: <https://www.ijser.org/researchpaper/Different-Requirements-Gathering-Techniques-and-Issues.pdf> [Accessed 1 January 2022].
- AgiraTechnologies. 2021. 7 Reasons To Choose Golang for Development | Hire Golang Developers. [online] Available at::https://www.agiratech.com/7-reasons-why-golang-is-the-best-programming-language-for-developer [Accessed 31 December 2021].
- 2022. [online] Available at: https://tekeye.uk/programming/how-to-create-an-ide [Accessed 1 January 2022].

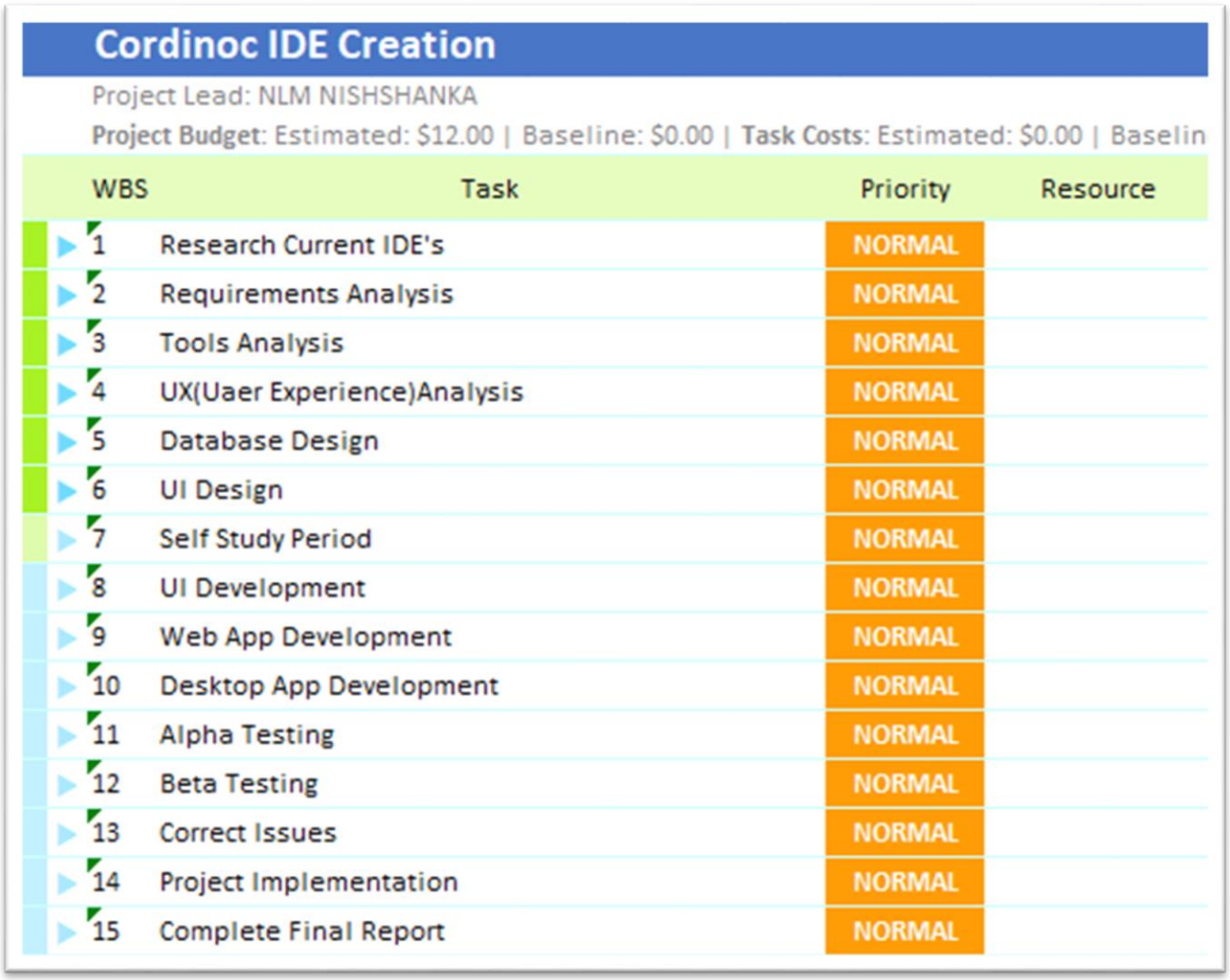
Team Plan and Responsibility Matrix

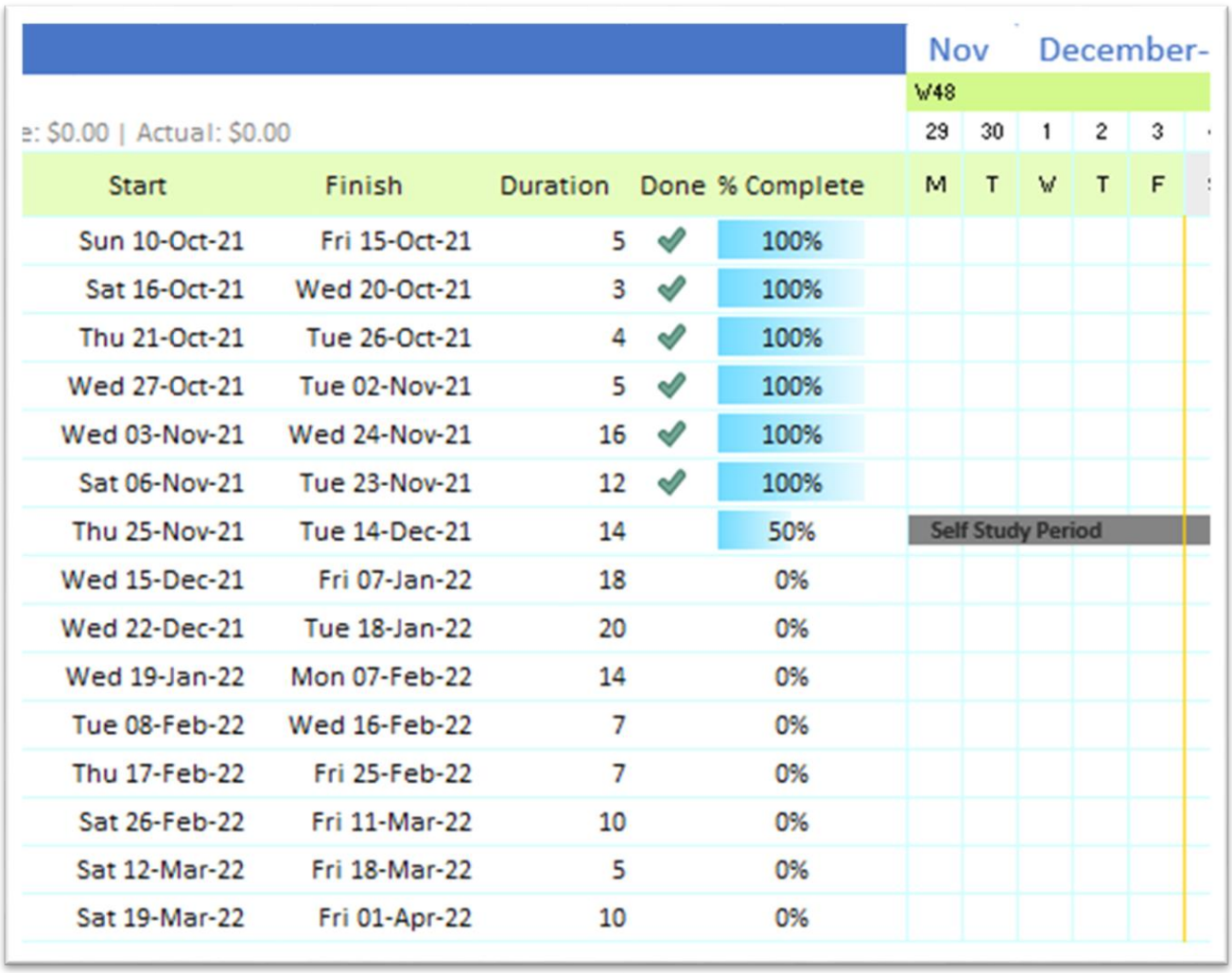
Member Responsibilities

Name	Index	Responsibilities
L.M. Nishshanka	10749846	• Back End Programming in GO Language • And other parts according to Group Leader Role
M.P.M. Abeyrathne	10749852	• Back End Programming in GO Language • And other parts according to Programming Leader Role
E.C.N. Nandasiri	10749855	• User Experience Analysis • Design User Interface using Figma • Develop User Interface using HTML, CSS and JS • Test Project • And other parts according to testing leader Role
A.B. Navodya	10749851	• Create and Maintenance Linux Server • Plan and Create Database using MariaDB • Maintenance Database • And other parts according to technical and quality leader Role
E.L.P. Prasandika	10749850	• User Experience Analysis • Design User Interface using Figma • Develop User Interface using HTML, CSS and JS • And other parts according to planning leader Role

Gantt Chart


Cordinoc IDE Creation
Project Lead: NLM NISHSHANKA
Project Budget: Estimated: $12.00 | Baseline: $0.00 | Task Costs: Estimated: $0.00 | Baseline: $0.00 | Actual: $0.00
Nov
December-2021
Jan
WBS
Task
Priority
Resource
Start
Finish
Duration
Done
% Complete
1 Research Current IDE's NORMAL Sun 10-Oct-21 Fri 15-Oct-21 5 100%
2 Requirements Analysis NORMAL Sat 16-Oct-21 Wed 20-Oct-21 3 100%
3 Tools Analysis NORMAL Thu 21-Oct-21 Tue 26-Oct-21 4 100%
4 UX(Uaer Experience)Analysis NORMAL Wed 27-Oct-21 Tue 02-Nov-21 5 100%
5 Database Design NORMAL Wed 03-Nov-21 Wed 24-Nov-21 16 100%
6 UI Design NORMAL Sat 06-Nov-21 Tue 23-Nov-21 12 100%
7 Self Study Period NORMAL Thu 25-Nov-21 Tue 14-Dec-21 14 50%
8 UI Development NORMAL Wed 15-Dec-21 Fri 07-Jan-22 18 0%
9 Web App Development NORMAL Wed 22-Dec-21 Tue 18-Jan-22 20 0%
10 Desktop App Development NORMAL Wed 19-Jan-22 Mon 07-Feb-22 14 0%
11 Alpha Testing NORMAL Tue 08-Feb-22 Wed 16-Feb-22 7 0%
12 Beta Testing NORMAL Thu 17-Feb-22 Fri 25-Feb-22 7 0%
13 Correct Issues NORMAL Sat 26-Feb-22 Fri 11-Mar-22 10 0%
14 Project Implementation NORMAL Sat 12-Mar-22 Fri 18-Mar-22 5 0%
Self Study Period
UI Development
Web App Development


Cordinoc IDE Creation
Project Lead: NLM NISHSHANKA
Project Budget: Estimated: $12.00 | Baseline: $0.00 | Task Costs: Estimated: $0.00 | Baselin
WBS
Task
Priority
Resource
1 Research Current IDE's NORMAL
2 Requirements Analysis NORMAL
3 Tools Analysis NORMAL
4 UX(Uaer Experience)Analysis NORMAL
5 Database Design NORMAL
6 UI Design NORMAL
7 Self Study Period NORMAL
8 UI Development NORMAL
9 Web App Development NORMAL
10 Desktop App Development NORMAL
11 Alpha Testing NORMAL
12 Beta Testing NORMAL
13 Correct Issues NORMAL
14 Project Implementation NORMAL
15 Complete Final Report NORMAL


Nov
December-
W48
e: $0.00 | Actual: $0.00
29
30
1
2
3
Start
Finish
Duration
Done
% Complete
M
T
W
T
F
Sun 10-Oct-21
Fri 15-Oct-21
5
100%
Sat 16-Oct-21
Wed 20-Oct-21
3
100%
Thu 21-Oct-21
Tue 26-Oct-21
4
100%
Wed 27-Oct-21
Tue 02-Nov-21
5
100%
Wed 03-Nov-21
Wed 24-Nov-21
16
100%
Sat 06-Nov-21
Tue 23-Nov-21
12
100%
Thu 25-Nov-21
Tue 14-Dec-21
14
50%
Self Study Period
Wed 15-Dec-21
Fri 07-Jan-22
18
0%
Wed 22-Dec-21
Tue 18-Jan-22
20
0%
Wed 19-Jan-22
Mon 07-Feb-22
14
0%
Tue 08-Feb-22
Wed 16-Feb-22
7
0%
Thu 17-Feb-22
Fri 25-Feb-22
7
0%
Sat 26-Feb-22
Fri 11-Mar-22
10
0%
Sat 12-Mar-22
Fri 18-Mar-22
5
0%
Sat 19-Mar-22
Fri 01-Apr-22
10
0%