

PALLIATIVE CARE PLATFORM

A MINI PROJECT REPORT

Submitted by

SIJI JOSE (TKM20MCA2037)

to

The APJ Abdul Kalam Technological University

In partial fulfillment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS



**Thangal Kunju Musaliar College of Engineering
Kerala**

MARCH 2022

DECLARATION

We undersigned hereby declare that the mini project report “ PALLIATIVE CARE PLATFORM ” , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of NATHEERA BEEVI M. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place : Karicode
Date : 05/03/2022

SIJI JOSE

Thangal Kunju Musaliar College of Engineering
Dept. of Computer Applications



C E R T I F I C A T E

This is to certify that, this report entitled ***“PALLIATIVE CARE PLATFORM”*** is a bonafide record submitted by **SIJI JOSE (TKM20MCA2037)** to the **APJ Abdul Kalam Technological University** in partial fulfillment of the requirements for the award of the Degree of **Master of Computer Applications** is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor

Mini Project Coordinator

Head of the Department

ACKNOWLEDGEMENT

First and foremost we thank GOD almighty and our parents for the success of this project. We owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of our project.

We extremely grateful to **Dr. Nadera Beevi S**, Head of the Department, Dept of Computer Applications, for providing us with best facilities.

We would like to thank our project guide **Natheera Beevi M**, Department of Computer Applications, who motivated us throughout the project

We profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout our course of study.

We owe our thanks to my friends and all others who have directly or indirectly helped us in the successful completion of this project.

SIJI JOSE

Abstract

“Palliative Care Platform ” aim to provide care such as home nursing services, ambulatory services, equipment and so on . Palliative care is specialized medical care for people living with a serious illness. This type of care is focused on providing relief from the symptoms and stress of the illness. The goal is to improve quality of life for both the patient and the family. Palliative care is provided by a specially-trained team of doctors, nurses and other specialists who work together with a patient’s other doctors to provide an extra layer of support. Palliative care is based on the needs of the patient, not on the patient’s prognosis. It is appropriate at any age and at any stage in a serious illness, and it can be provided along with curative treatment The services of the service providers will be listed and published. They have a special account and will publish it. The user can filter the service on the basis of location and service type. For example, when searching for services such as home nursing care, the companies that provide it are listed. User can subscribe and use the services according to their needs and also user can review the service.

.

Coxtexts

1	Introduction	1
1.1	Objective	1
2	Methodology	
2.1	Existing System.....	2
2.2	Proposed System.....	2
2.3	Front-End and Back-End Technologies	3
2.3.1	HTML	3
2.3.2	CSS	4
2.3.3	Python	4
2.4	Framework — Django.....	6
3	Result Aed Discussion	
3.1	Testing Methods	8
3.2	Test Plan	10

Chapter 1

Introduction

“Palliative Care Platform” is a web based application. It provide various kinds of services such as home nursing services, ambulatory services, equipment and so on. Various service providers register to this platform to provide above mentioned services. The services registered by the service providers will be listed only after admin approves. The approval will be depend on various documents corresponding to their service. Thenthe user can book the service on the basis of their need and make payment to ensure service. User can make reviews related to their service so that it will help the upcoming users to get an overall idea about the services provided by various service providers.

1.1 Objective

The goal of palliative care is to relieve suffering and provide the best possible quality of life for patients and their families. Symptoms may include pain, depression, shortness of breath, fatigue, constipation, nausea, loss of appetite, difficulty sleeping, and anxiety. The team will help you gain the strength to carry on with daily life. In short, palliative care will help improve your quality of life. And recent studies, including one published in the New England Journal of Medicine, have shown that patients with a serious illness who received palliative care lived longer than those who did not receive this care.

The key Features are:

- Improves Quality Of Life
- Relieves Suffering From Symptoms and Stress
- Helps You Match Treatment Options to Your Goals
- Works Together with Your Other Doctors

Chapter 2

Methodology

2.1 Existing System

- Palliative care currently uses a manual system for the management and maintenances of critical information.
- The current system requires numerous paper forms, with data stores spread throughout the Palliative care infrastructure.
- Often information is incomplete or does not follow management standards.
- In the case of existing system, we depend on various different agencies to get home nurses for various health care services.
- We contact one of the agencies and they provide home nurses.
- In this case we actually don't know the character of the person that they provide.
- They may or may not be a good person.
- We only get to know that after their services.
- If all these were not going well then it will cause money and time loss, etc.

2.2 Proposed System

- The Palliative Care System is designed for any Palliative Care organization to replace their existing manual paper-based system.
- The new system is to manage the services provided by Palliative Care units.
- Home nursing, ambulance service and equipment providing service, these services are to be provided in an efficient, cost-effective manner.
- Provide location-based service filtering option.
- Palliative care filter provides an easy and validated mechanism to retrieve relevant literature with the goal of reducing the time and resources currently required for such tasks.

2.3 Front-End and Back-End Technologies

2.3.1 HTML

To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (from Hyper Text Markup Language). HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Publish online documents with headings, text, tables, lists, photos, etc.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents

HTML was originally developed by Tim Berners-Lee while at CERN, and popularized by the Mosaic browser developed at NCSA. During the course of the 1990s it has blossomed with the explosive growth of the Web. During this time, HTML has been extended in a number of ways. The Web depends on Web page authors and vendors sharing the same conventions for HTML. This has motivated joint work on specifications for HTML. It is a platform independent language that can be used on any platform such as Windows, Linux, Macintosh, and so on. To display a document in web it is essential to mark-up the different elements (headings, paragraphs, tables, and so on) of the document with the HTML tags. To view a mark-up document, user has to open the document in a browser. A browser understands and interprets the HTML tags, identifies the structure of the document (which part are which) and makes decision about presentation (how the parts look) of the document.

2.3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) `text/css` is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

2.3.3 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python 2.0 was released

on 16 October 2000 with any major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2 to 3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or pascal. In 1999, Guido van Rossum defined his goals for python:

- an easy and intuitive language just as powerful as those of the major competitor
- open source, so anyone can contribute to its development
- Code that is as understandable as plain English

Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. Some of the key advantages of learning Python:

- Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language: Python is a great language for the beginner level programmers and supports the development of a wide rangrnditemizee of applications from simple text processing to WWW browsers to games.

Following are important characteristics of Python Programming:

- It supports functional and structured programming methods as well as OOP

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

2.4 Framework

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support. Django helps you write software that is:

Complete

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation.

Versatile

Django can be (and has been) used to build almost any type of website from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!

Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

Secure

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

Scalable

Django uses a component-based "shared-nothing" architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (eg. Instagram and Disqus, to name just two).

Maintainable

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

Portable

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

Chapter 3

Result and Discussion

Testing is the major quality measures employed during the software development. After the coding phase, computer programs available are executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during the previous phase. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

- Testing is a process of executing a program with the intention of finding an error.
- A good test case is one that has a highest probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

Our objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time and effort. Testing demonstrate that software functions appear to be working according to specification, that performance requirements appears to have been met. Data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But there is one thing that testing cannot do: Testing cannot show the absence of defects it can only show that software defects as present.

3.1 Testing Methods

There are different types of testing methods available.

- **Unit Testing**

In this testing we test each module individually and integrate the overall system. Unit testing focuses verification efforts on the smaller unit of software

design in the module. This is also known as “module” testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In these testing steps each module is found to work satisfactory as regarding to the expected output from the module. There are some validation³⁰ checks for verifying the data input given by the user. It is very easy to find errors and debug the system.

In this project, after coding each module have been individually tested to determine whether they are coded correctly so that they satisfy the requirements in the specifications and execute effectively as individual units was tested and run individually.

• **Integration Testing**

Data can be lost across an interface. One module can have an adverse effect on the other sub functions when combined may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The need for integrated test is to find the overall system performance. According to this project, using the integrated test plan prepared in the design phase of the system developed as a guide, the integration test was carried out. All the errors found in the system were corrected for the next testing steps.

• **Validation Testing**

At the culmination of black box testing, software is completely assembled as a package, interface errors have been uncovered and corrected and final series of software test, validation test begins. Validation testing can be defined in many ways but a simple definition is the validation succeeds when the soft-ware functions in a manner that can be reasonably accepted by the customer.

After validation test have been conducted one of the two possible conditions exists.

- The function or performance characteristics confirmed to the specification and are accepted.

- A deviation from specification is uncovered and a deficiency list is created.

- **Dry run (testing)**

A dry run (or a practice run) is a testing process where the effects of a possible failure are intentionally mitigated. Dry run testing is a static test and should be performed by the developer³¹ to mitigate the effects of a failure of the product meaning before the end user gets the product and discovers it doesn't do what it says it will. In dry run testing, no hardware is used, but it is assumed that the programmer who is testing the code is aware of what each line of code is supposed to do and gives him or her opportunity to make corrections to the code before it becomes an issue for the actual software. Basically, a dry run test consists of programmers manually reading their code line by line to find errors and fix them.

- **User Acceptance testing**

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for a user acceptance by constantly keeping in touch with prospective system at the time of developing and making changes wherever required. This is done with regard to the following points:

- Output screen design
- Input screen design
- Menu driven system

3.2 Test Plan

A test plan is a systematic approach to test a system. The plan typically contains a detailed understanding of what the eventual workflow will be. Normally testing of any large system will be in two parts.

- The functional verification and validation against the requirement specification
- Performance evaluation against the indicated requirements

Testing activity is involved right from the beginning of the project. At the very first stage of testing, the goals and objectives are set. This simplifies the limits or borders of testing process.

Before testing, the tester should plan what kind of data he is giving for test. Give data inputs as functional, boundary, stress, performance, usability values etc.

Characteristics of a Good Test:

- Tests are likely to catch bugs
- No redundancy
- Not too simple or too complex

Test Cases

A specific set of steps and data along with expected results of a particular test objective. A test case should only test one limited subset of a feature or functionality. Test cases documents for each functionality/testing area will be written, reviewed and maintained separately in excel sheets. In system testing, test data should cover the possible values of each parameter based on the requirements.

Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check our error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if error test cases are not yet written, it is OK for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any, is expected to trigger errors.

Implementation

Implementation is the process of having the system personnel check out and put new equipment to use, train the users to use the new system and construct any file that are needed to see it. The final and impartment phases in the system life cycle are the implementation of the new system. System implementation refers to the steps necessary to install a new system to put into operation. The implementation has different meaning, ranging from the conversion of a basic application to complete replacement of computer system. Implementation includes all these activities that take place to convert from old system to new one. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The methods of implementation and time scale adopted are found out initially. The system is tested properly and at the same time the users are trained in the new procedure.³³ Proper implementation

is essential to provide a reliable system to meet organizational requirements. Successful implementations may not guarantee improvement in the organization involves the following things:

- Careful planning
- Investigation of the system and constraint
- Design the methods to achieve the change over
- Train the staff in the changed phase
- Evaluation of change over method Implementation methods

There are several methods for handling the implementation and consequent conversation from the old to new automated system. The most secure for this conversation is to run the old and new system in parallel. This method offers high security but the cost for maintaining the two systems in parallel is very high. Another method is direct cut over the existing system to automated system. The chance may take place within a week or within a day.

Implementation Phase: It includes a description of all activities that most occur to implement the new system and put into operation. It consists of the following

steps:

- List all files required for the implementation
- Identify all data required to build new files during the implementation
- List all new document and procedure that go to the new system

Testing Phases and Objectives

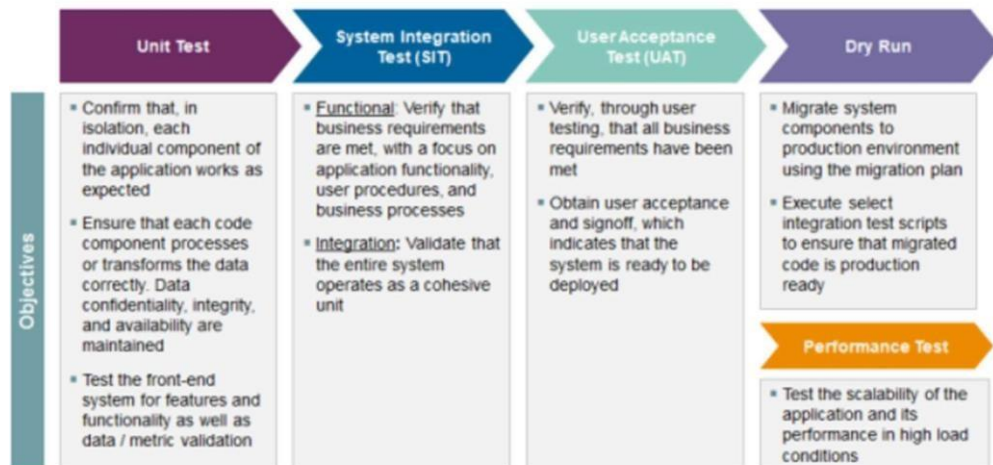


Figure 4.1: Testing Phases and Objectives