CARE AND SERVICE

PROJECT THESIS SUBMITTED

TO

AWH ENGINEERING COLLEGE KUTTIKKATTOOR, CALICUT - 8
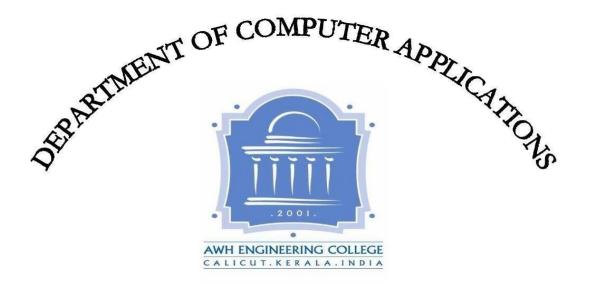
IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE AWARD OF THE

DEGREE OF

Master of Computer Applications

BY PRAVEENA V S

DEPARTMENT OF COMPUTER APPLICATIONS

AWH ENGINEERING COLLEGE KUTTIKKATTOOR CALICUT

FEBRUARY 2022

AWH ENGINEERING COLLEGE

CALICUT

CERTIFICATE

This is to certify that this thesis entitled "CARE

AND SERVICE" submitted herewith is an authentic record of this

by PRAVEENA V S (AWH20MCA-2010) under our guidance

In partial fulfilment of the requirements for the award of Master of

Computer Applications from APJ ABDUL KALAM TECHNOLOGICAL

UNIVERSITY during the academic year 2021.

Mrs. Sruti Sudevan   **Mrs. Keerthi Shankar K**

Assistant. Professor   Assistant. Professor

Head of the Department   Project Guide

External Examiner   **Internal Examiner**

ACKNOWLEDGEMENT

ABSTRACT

Palliative care is an approach that improves the quality of life of patients and their families who are facing problem associated with life threatening problems. In this web application I introduce a system which is helpful for pain and palliative care units. Pain and palliative can register for providing their services. They can enter all services providing by them. The public who need support under this category can register and place a request. The units can accept the request according to their pin code. The public can also donate to the palliative unit.

CONTENTS

INTRODUCTION

1. INTRODUCTION

Palliative care an approach that improves the quality of life of patients and their families who are facing problems associated with life threatening illness. Palliative care uses a team approach to support patients and their caregivers. It offers a support system to help patients live as actively as possible until death. Palliative care is explicitly recognized under the human right to health. It should be provided through person- centered and integrated health services that pay special attention to the specific needs and preferences of individuals. In this web application I introduce a system which is helpful for pain and palliative care units. Pain and palliative can register for providing their services. They can enter all services providing by them. The public who need support under this category can register and place a request. The units can accept the request according to their location. The public can also donate to the palliative unit.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 Existing system

Palliative care has lots of services. But no many people know about this. There is no way for them to do service or apply service through online today.

2.2 Proposed system

Care and service help people to inform their needs directly to the palliative care and they can also donate items directly. They also can make request for their needs to the palliative care.

2.3 Module description

This project has 3 modules:

ADMIN

- View registered users

- View services

- View donations

- Accept/ reject donations

- Approve palliative units

PALLIATIVE CARE UNITS

- View request

- Accept / reject request

- Add/ delete services

- View feedback

PUBLIC

- Post donations

- Post request

- Post feedback

- View request/donation status

2.4 Sprint

Sprint 1

Module	Task	Pending task of any	Hours for Completion	Expected date of Completion	Actual date of Completion	Reason for deviation
Admin	Registration	-	2 hrs	31/12/2021	31/12/2021	-
	Login	-	1 hrs	01/01/2022	01/01/2022	-
	Homepage	-	3 hrs	01/01/2022	01/01/2022	-
	View Registered Users	-	2 hrs	26/11/2021	26/11/2021	-
	View Services	-	2 hrs	23/12/2021	23/12/2021	-
	View Donations	-	2 hrs	26/11/2021	26/11/2021	-
	Accept/Reject Donations	-	2 hrs	10/12/2021	10/12/2021	-
	Approve Palliative Units	-	2 hrs	17/12/2021	17/12/2021	-
	Approve Public	-	2 hrs	23/12/2021	23/12/2021	-

Sprint 2

Module	Task	Pending task of any	Hours for Completion	Expected date of Completion	Actual date of Completion	Reason for deviation
Palliative Care Units	Registration	-	2 hrs	31/12/2021	31/12/2021	-
	Login	-	1 hrs	01/01/2022	01/01/2022	-
	Homepage	-	3 hrs	01/01/2022	01/01/2022	-
	View Request	-	2 hrs	26/11/2021	26/11/2021	-
	Accept/reject Request	-	2 hrs	23/12/2021	23/12/2021	-
	Add/Delete Services	-	2 hrs	26/11/2021	26/11/2021	-
	View Feedback	-	1 hrs	24/12/2021	24/12/2021	-

Sprint 3

Module	Task	Pending task of any	Hours for Completion	Expected date of Completion	Actual date of Completion	Reason for deviation
Public	Registration	-	2 hrs	31/12/2021	31/12/2021	-
	Login	-	1 hrs	01/01/2022	01/01/2022	-
	Homepage	-	3 hrs	01/01/2022	01/01/2022	-
	Post Request	-	2 hrs	26/11/2021	26/11/2021	-
	Post donations	-	2 hrs	23/12/2021	23/12/2021	-
	View Services and status	-	2 hrs	26/11/2021	26/11/2021	-
	Post Feedback	-	1 hrs	24/12/2021	24/12/2021	-

FEASIBILITY STUDY

3. FEASIBILITY STUDY

An analysis of the ability to complete a project successfully, taking into account legal, economic, technological, scheduling, and other factors is considered as feasibility study. Rather than just diving into a project and hoping for the best, feasibility study allows project managers to investigate the possible negative and positive outcomes of a project before investigate the possible outcomes of a project before investing too much time and money.

3.1 Economical Feasibility

The economic analysis is done to determine the benefits and savings that are expected from candidate system and compare them with costs. Thus, coming to a conclusion that weather the system is economically feasible or not. This system is cost effective as well as time effective, thereby making it economically feasible. This study presents tangible and intangible benefits from the project by comparing the developments and operational costs. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility.

3.2 Technical Feasibility

The technical requirements for the system are economic and it does not use additional software. That is whether the system can be implemented using the existing technologies or not. This application is develop using python, whose development kit are easily available and free of cost, thus making our system technically feasible.

3.3 Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and there for it will accept broad audience.

3.4 Behavioral Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and there for it will accept broad audience.

3.5 Software Feasibility

Even though this application is developed in very high software environment, it is also supported by many other environments with minimum changes. The system is fully feasible to be executed on any kind of operating systems and browsers.

3.6 Hardware Feasibility

Software can be developed with the existing resources. But the existing resources may or may not be used to produce hardware. If no hardware is newly bought for project, then software is said to achieve hardware feasibility. The system is hardware wise feasible because it needed absolutely no new hardware.

SOFTWARE ENGINEERING

PARADIGM

4. SOFTWARE ENGINEERING PARADIGM

The software engineering paradigm which is also referred to as a software process model or Software Development Life Cycle (SDLC) model is the development strategy that encompasses the process, methods and tools. SDLC describes the period of time that starts with the software system being conceptualized.

4.1 Agile model

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. At the end of the iteration, a working product is displayed to the customer and important stakeholders.

In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

4.2 Scrum

Scrum is an agile framework for managing knowledge work, with an emphasis on software development. It is designed for teams of three to nine members, who break their work into actions that can be completed within time boxed iterations, called "sprints", no longer than one month and most commonly two weeks, then track progress and re-plan in 15-minute stand-up meetings, called daily scrums.

Scrum is an iterative and incremental framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines involved.

SYSTEM REQUIREMENT SPECIFICATION

5. SYSTEM REQUIREMENTS SPECIFICATION

5.1 Software Requirements

    Operating System   : Windows 10 or above

Backend   : HTML

Frontend   : PYTHON

Database   : MySQL

Framework   : Django

IDE   : PyCharm

5.2 Hardware Requirements

A device with internet connection

Processor   : Intel core or above

RAM   : Minimum 2GB (recommended 8GB)

Monitor   : 14" LCD or Above

Hard disk   : Minimum 128 GB

Key Board   : 108 Keys

Mouse   : Any type of mouse

SYSTEM DESIGN

6. SYSTEM DESIGN

System design is the first in the development phase for many engineered product or system. It may define the process of applying various techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

6.1 Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data.

In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system. The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must: Determine the relationships between the different data elements and superimpose a logical structure upon the data on the basis of these relationships.

Normalization

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e., repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this normal forms or rules for structuring relation are used.

Insertion anomaly**:** Inability to add data to the database due to absence of other data.

Deletion anomaly**:** Unintended loss of data due to deletion of other data.

Update anomaly**:** Data inconsistency resulting from data redundancy and partial update.

Normal Forms**:** These are the rules for structuring relations that eliminate anomalies.

1. First Normal Form (1NF)

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. This mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

2. Second Normal Form (2NF)

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

- Primary key is a not a composite primary key
- Every non key attribute is fully functionally dependent on full set of primary keys.

3. Third normal Form(3NF)

A relation is said to be in third normal form if there exist no transitive dependencies.

Transitive dependency**:** If two non-key attributes depend on each other as well on the primary key then they are said to be transitively dependent. the above normalization principle was applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

6.2 TABLES

Login

FIELD	DATA TYPE	DESCRIPTION
Login_id	INT	PRIMARY KEY
Username	VARCHAR (50)	NOT NULL
User_type	VARCHAR (20)	NOT NULL
User_id	INT (20)	NOT NULL

Palliative registration

FIELD	DATA TYPE	DESCRIPTION
Unit_id	INT (20)	PRIMARY KEY
Address	VARCHAR (100)	NOT NULL
Pincode	Int (6)	NOT NULL
Email_id	VARCHAR (50)	NOT NULL
Mob_1	INT (11)	NOT NULL
Mob_2	INT(11)	NOT NULL
Username	VARCHAR(50)	NOT NULL
Password	VARCHAR(50)	NOT NULL
Confirm_Password	VARCHAR (50)	NOT NULL
Status	VARCHAR(50)	NOT NULL

Public Registration

FIELD	DATA TYPE	DESCRIPTION
Public_id	INT (20)	PRIMARY KEY
Name	VARCHAR(50)	NOT NULL
Address	VARCHAR (100)	NOT NULL
Email_id	VARCHAR (50)	NOT NULL
Mobile_number	INT (11)	NOT NULL
Username	VARCHAR(50)	NOT NULL
Password	VARCHAR (50)	NOT NULL
Confirm_password	VARCHAR(50)	NOT NULL
Status	VARCHAR(50)	NOT NULL

Request

FIELD	DATA TYPE	DESCRIPTION
Request_id	INT (20)	PRIMARY KEY
Public_id	INT(20)	FOREIGN KEY
Unit_id	INT(20)	FOREIGN KEY
Name_patient	VARCHAR(50)	NOT NULL
Type_disease	VARCHAR(50)	NOT NULL
Service_request	VARCHAR(50)	NOT NULL
Equipment_request	VARCHAR(50)	NOT NULL

Other-diaease	VARCHAR(50)	NOT NULL
Other-service	VARCHAR(50)	NOT NULL
Status	VARCHAR(50)	NOT NULL
Date	DATE	NOT NULL
Time	TIME	NOT NULL

Donation

FIELD	DATA TYPE	DESCRIPTION
Donation_id	INT (20)	PRIMARY KEY
Public_id	INT (20)	FOREIGN KEY
Unit_id	INT (20)	FOREIGN KEY
Donation	VARCHAR (20)	NOT NULL
Value	INT(20)	NOT NULL
Date	DATE	NOT NULL
Time	TIME	NOT NULL

Doctor Service

FIELD	DATA TYPE	DESCRIPTION
doctor_id	INT (20)	PRIMARY KEY
Unit_id	INT (20)	FOREIGN KEY
Public_id	INT(20)	FOREIGN KEY

| Name | VARCHAR(50) | NOT NULL |
| Department | VARCHAR (50) | NOT NULL |

Feedback

FIELD	DATA TYPE	DESCRIPTION
feedback_id	INT (20)	PRIMARY KEY
Unit_id	INT (20)	FOREIGN KEY
Public_id	INT(20)	FOREIGN KEY
Feedback	VARCHAR (50)	NOT NULL
Date	DATE	NOT NULL
Time	TIME	NOT NULL

Availability Services

FIELD	DATA TYPE	DESCRIPTION
Service_id	INT (20)	PRIMARY KEY
Unit_id	INT (20)	FOREIGN KEY
Item	VARCHAR(50)	NOT NULL
Existing_no	INT (50)	NOT NULL
Remaining	INT(50)	NOT NULL
Doctor_id	INT(10)	FOREIGN KEY

6.3 UML Designs

The unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artefacts of the software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object-oriented software and software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

6.4 Use case diagram

6.5 Scenario

Admin:

- Can approve palliative units.
- Can view services.
- Can view registered users.
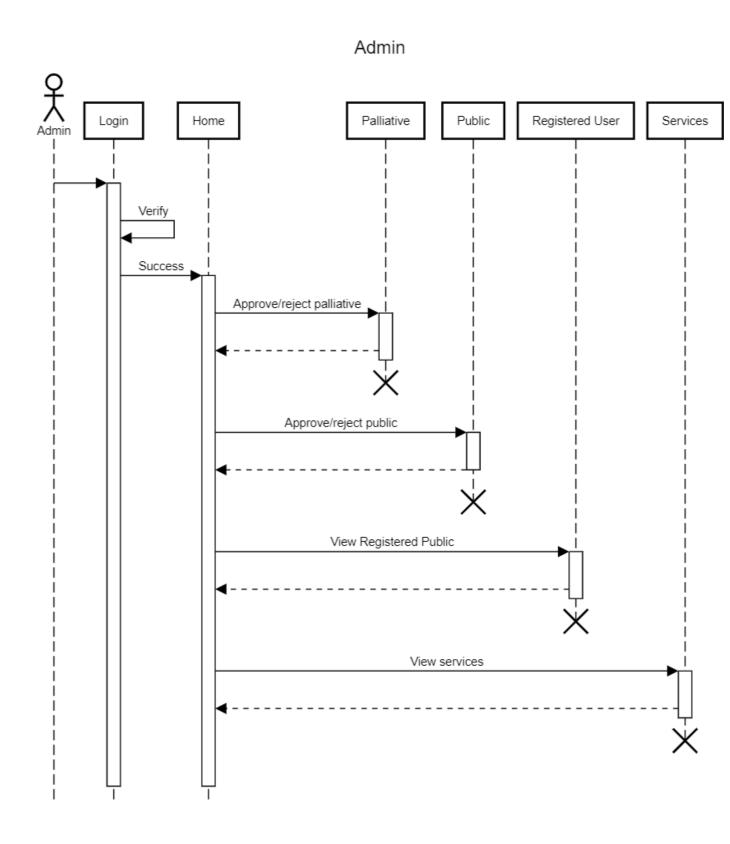
Palliative Units:

- Can view the requests and approve or reject requests.
- Can view donations and approve or reject donations.
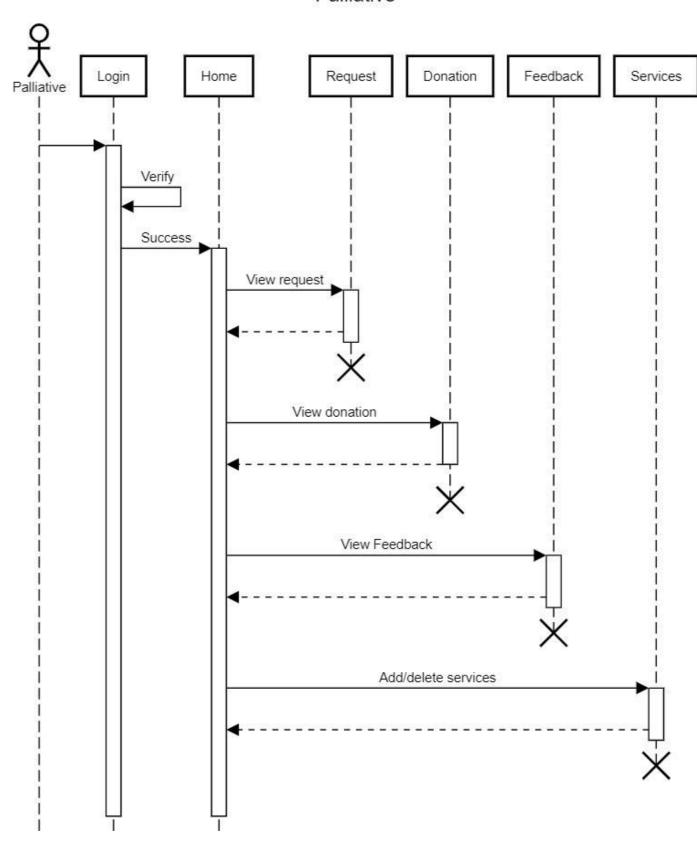- Can view feedback.
- Can add or delete services.

Public:

- Can post request
- Can post feedback.
- Can post donations.
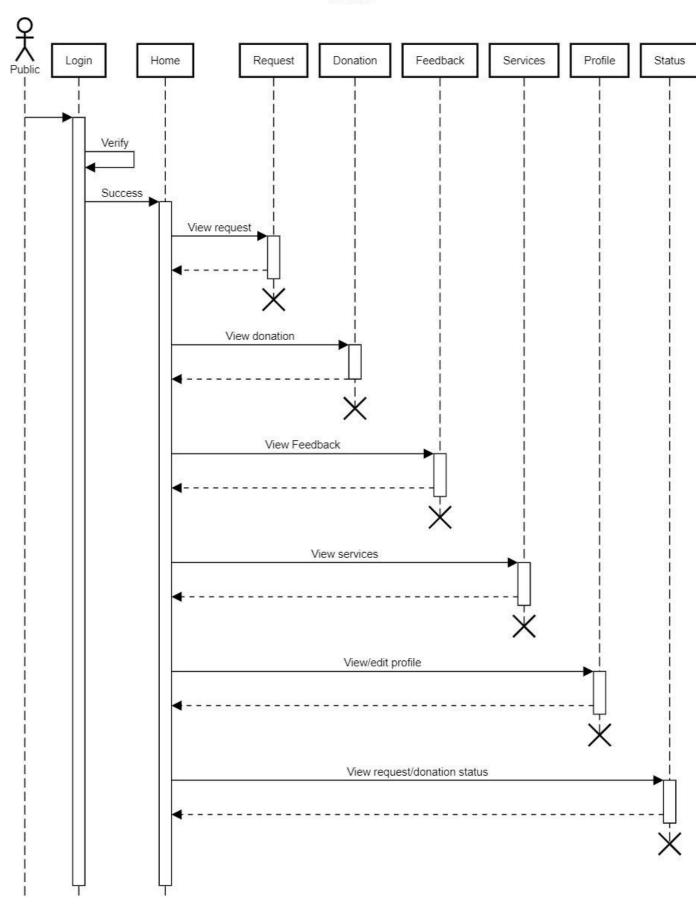- Can view request/ donation status.
- Can view services.

6.6 Sequence Diagram

Admin

Palliative

Public

SYSTEM DEVELOPMENT

7. SYSTEM DEVELOPMENT

System development is series of operations to manipulate data to produce output from computer system. The principal activities performed during the development phase can be divided into two major related sequences.

- External system development

- Internal system development

    The major external system activities are:

- Implementation

- Planning

- Equipment acquisition

- Installation

7.1 Coding

The purpose of code is to facilitate the identification and retrieval of items of information. A code is an ordered collection of symbols designed to provide unique identification of entity or an attribute. Code also show interrelationship among different items. Codes are used to identify, access, sort, matching records. The code ensures that only one value of code with a single meaning is applied to give entity or attribute as described in various ways.

PYTHON

Python is a widely used high-level programming language for general purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java.

Python is a multi-paradigm programming language. Object- oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Libraries

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

MySQL client

License GPL Platforms OS Independent Python versions Python 2.7 and 3.4+ PyPI. https://pypi.org/project/mysqlclient/ mysqlclient is a fork of MySQL-python. It adds Python 3 support and fixed many bugs. It is the MySQL library that is recommended by the Django documentation. **Django**

Django is a Python-based free and open-source web framework that follows the model template(MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

MYSQL Database

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications-which may run either on the same computer or on another computer across a network (including the Internet).Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. Structured Query Language is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements. SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SYSTEM TESTING AND IMPLEMENTATION

8. SYSTEM TESTING AND IMPLEMENTATION

Testing is the vital to the success of the system. It makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved in this project. It is the stage of implementation, which ensures that system works accurately and effectively before the live operation commences. It is a confirmation that all are correct and opportunity to show users that the system must be tested and show that the system will operate successfully and produce expected results under expected conditions. Software testing is a crucial element of software quality assurance and represents the unlimited review of specification, design and coding. Testing represents an interesting anomaly for the software. During the earlier definition and development phase, it was attempted to build the software from an abstract concept to implement.

Testing is a set of activity that can be planned in advance and conducted. Systematically, this is aimed at ensuring that the system works accurately and efficiently before live operations commences.

8.1 Types of Testing

Different types of testing are,

- Unit testing
- Integration testing
- System testing
- Validation testing
- User acceptance testing.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. All modules were tested and individually as soon as they were completed and were checked for their correct functionality. Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. This testing is carried out during programming stage itself. In this testing step each Module is found to be working satisfactorily as regard to the expected output from the module.

Integration Testing

Integration testing is a systematic technique for testing to overcome the errors associated within the interface. In this System all the modules such as login, registration, admin in web, user in web, and the final hardware and software are combined and then the entire program is tested as a whole. Thus, in the integration testing step all the errors in the implementation of the system are corrected. Data can be lost across an interface one module can have an adverse effect on others; subfunctions when combined may not produce the desired major functions integration testing is a systematic testing for constructing the program structure. The objective is to take unit tested modules and to combine them and test it as a whole.

- The system contains various components and they have to be combined and tested.
- The software is embedded into hardware and tested.
- The modules are combined and tested.
- The hardware part tested separately.

System Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it doesn't produce the required data in the specific format. The output displayed or generated by the system under consideration is tested by, asking the user about the format displayed. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. Hence the output testing doesn't result in any correction in the system.

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 user Acceptance by constantly keeping in touch with prospective system at the time of developing and making change wherever required. This is done with regard to the following points.

- Output Screen design.
- Input Screen design.
- Menu driven system.

8.2 Implementation

Implementation is the stage of project, when theoretical design is turned in to a working system. The most crucial stage is achieving a successful system and confidence that the new system will be work effectively. It involves careful planning, investigation of the manual system and to new system.

Implementation means converting a new or revised system design into an operational one. The implementation includes all those activities that take place to convert from the old system to the new one.

There are several activities involved while implementing a project:

- Careful planning.
- Investigating the current system and its constraints on implementation.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover procedure and evaluation of change over method.

Implementation is the final stage and it is an importance phase. The first task in implementation was the implementation planning, that is deciding on methods to be adopted. After the system was implemented successfully, training of the user was one of the most important subtasks of the developer. For this purpose, the user or system manual were prepared and handled over to the user to operate the developed system. So, change over plays a vital role, which checks the developed tool for the following requirements, and then only the user accepted the developed tool. The changeover took place only when the system had been proved to the satisfaction of the system analysis and other implementation activities have been completed.

SYSTEM MAINTENANCE

9. SYSTEM MAINTENANCE

Maintenance is making adaptation of the software for external changes (requirements changes or enhancements) and internal changes (fixing bugs). When changes are made during the maintenance phase all preceding steps of the model must be revisited.

There are three types of maintenance:

- Corrective (Fixing bugs/errors)

- Adaptive (Updates due to environment changes)

- Perfective (Enhancements, requirements changes)

Maintenance is enigma of the system development. The definition of the software maintenance can be given describing four activities that are undertaken after the program is released for use.

The maintenance activity occurs since it is unreasonable to assume that software testing will uncover all in a large system. The second activity that contributes the definition of maintenance occurs since rapid changes are encountered in every aspects of computing. The third activity involves recommendation for new capabilities, modification to the existing functions and general enhancements when the software is used. The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability.

FUTURE ENHANCEMENT

10. FUTURE ENHANCEMENT

Care and service are now just accepting the donations from public. In future the public who are willing can sponsor the patients who need help. The palliative can short lists the patients who need help and the public can help them.While requesting the publican choose options for equipment, service or both. The palliative can share their requirements with other units.

CONCLUSION

11. CONCLUSION

The project "CARE AND SERVICE" helps the patients who need help from the palliative care units. They can use web application. Palliative care has many services, but many people don't know about it. There is no way for them to do service or apply services online today. So this web application helps them to do through online.

With the involvement of technology everything is just a fingertip away. So, this application creates a way to make everything online instead of following an offline pattern. So, it's going to make the society a better place.

APPENDIX

12. APPENDIX

BIBLIOGRAPHY

13. BIBLIOGRAPHY

Journal Papers

[1] M. Mansourvar and N. Y. Mohd, "Web portal as a knowledge management system in the universities," World Academy of Science, Engineering and Technology, 2010.

[2] M. Mansourvar, Development of a Job Web Portal to Capture Industry's Needs, 2011.

[3] M. Mansourvar and N. B. M. Yasin, "Knowledge portal: a tool to capture university requirements," in Proc. 2011 International Conf. on Graphic and Image Processing, International Society for Optics and Photonics, October 2011.

Websites

[1] https://www.w3schools.com/python/

[2] https://docs.djangoproject.com/en/3.2/intro/tutorial01/

[3] https://pypi.org/project/mysqlclient/

[4] https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-sql- command.html

[5] https://www.geeksforgeeks.org/python-programming-language

Books

[1] Think Python: An Introduction to Software Design - Allen B. Downey
[2] MySQL in a Nutshell-Russell J. T. Dyer
[3] Mastering Django-Nigel George
[4] Wesley J. Chun, "Core Python Applications Programming", 3rd Edition, Pearson Education, 2016
[5] Charles Dieback, "Introduction to Computer Science using Python", Wiley, 2015