# INTRODUCTION

## INTRODUCTION

### System Overview

Computers have wide applications and their uses extend to diverse area. This has made the various sectors to computerize their various activities with the most sophisticated software available in the most comfortable way. There are plenty of ways to help seniors and grow a successful business. Errand services can be a success, as long as you can bring your skills to their homes or assist older people who can no longer walk or drive in some way. **‘SENIOR CARE SERVICES’** help seniors transition from their homes to assisted living facilities, from packing and transporting, to setting up or storing their furniture and possessions. Many older folks would like to keep living on their own, but need a little assistance with daily chores, bill paying, grocery shopping, etc. With just a little outside help and a daily visit from a friendly caregiver, many can maintain their independence. Families are also happy to find a responsible, reliable person in whom they can trust.

The system consists of mainly three modules Admin, Senior citizens and Service providers. Service providers may be working or non-working people. They can use their free time to provide such a social service to the old people according to their needs. „SENIOR CARE SERVICES‟ is based on this scenario. To manage the activities of senior citizens and service providers, there exists an admin module. Admin can check whether the details provided by the senior citizens and the service providers (for example, age, identity proofs uploaded) are genuine or not. Admin can give approval if it is genuine only. Approved service providers and senior citizens can only get into the system and request and provide services. In order to ensure the quality of services provided by the service providers there exists a rating for the services. The proposed system uses PYTHON technology with DJANGO framework and MYSQL as backend. We can register and add resources to the site for helping the old people by mentioning price, location etc. It helps the old people to search the services based on location. Overall the proposed system will help the old people by

providing what they needed. Also, the service providers can use their free time for a social service by using this site.

### Objectives of System

* + - People can use their free time as social service to the senior citizen according to their needs.
    - Provide easy and flexible management of services.
    - Provide user friendly interface
    - Less expensive
    - Overall, the system helps the old people by providing them what they needs.

# SYSTEM ANALYSIS

## SYSTEM ANALYSIS

System analysis is the process of reduction of the entire system by studying the various operations performed and the relationship with the system and the requirement of its successor. It is the most essential part of project development. System will be analyzed in terms of their objectives, and the input, processes and output required achieving these objectives. This aims to identify the boundaries of the system, the subsystems and the interfaces between subsystems. A system can be defined as an orderly grouping of independent component linked together according to a plan to achieve a specific objective.

System analysis is the term used to describe the process of collecting and analysing facts in respect of existing operation of the situation prevailing so that an effective computerized system may be designed and implemented of proved feasible. To analyze a system means to study the system in detail. The analyst has to understand the functioning and concept of the system, before design the appropriate computer based system that will meet all the requirements of the existing system. The system analyst has to carry out a customary approach to use the computer for problem solving.

During the system analysis phase our system is analysed. SENIOR CARE SERVICES consist of admin, senior citizen, and service provider. Admin has the complete control of the system. There is no proper existing system. For overcome the difficulties of the existing system we develop this web application called SENIOR CARE SERVICES. The proposed system called SENIOR CARE SERVICES is an efficient, easy to manage, user friendly website. The existing system is mainly focused on the senior citizen who need just a little outside help.

### Existing System

Now a days help for the old people are done by some organizations so the old people can‟t access the service because the service organization may be not in the current location or not near locations. Also, there is no proper system for helping seniors. For example, transition from their homes to assisted living facilities, from packing and transporting to setting up or storing their furniture and possessions. Many older folks would like to keep living on their own. They lose their freedom and independency in old age homes. New generation is willing to do services but they don‟t know how to do the services in their busy schedule. Currently there is no proper existing system & no common platform exist for such services together to the senior citizen.

### Drawbacks

* + - Existing system is in a scattered form that is, we have to depend different websites for different services.
    - There is no way to ensure the quality of services provided by the service providers in the existing system.
    - There is no location based search of services.
    - Existing system doesn‟t show any service status of service providers.

### Requirement Analysis

Requirement analysis is the first technical step in the software engineering process. It is at this point that a general statement of software scope is refined into a concrete specification that becomes a foundation for all the software engineering activities that follow. Analysis must focus on the informational, functional and behavioral domains of a problem. To better understand what is required, models are created, the problem is partitioned, and representation that depict the essence of requirements and later, implementation detail, are developed.

Requirements are refined and analyzed to assess their clarity, completeness and consistency. Data, functional and behavioural requirements are identified by eliciting information from the user. Software requirements analysis work products must be reviewed for clarity, completeness and consistency.

SENIOR CARE SERVICES is a user friendly and also an easy to use website for the senior citizen to request and receive services online without causing much delay. The basic concepts of registration, service details, payment, feedback, rating etc. should be maintained by this web site. The main goal is the interaction between the senior citizen and service providers. This system is technically feasible. The input can be done through input boxes which are both interactive and user friendly.

#### Functional Requirements

* + - User interface should be simple and aesthetically pleasing
    - Details about services, requests, senior citizen, service providers should be stored in database.
    - There should be an interactive platform for requests and services.
    - Administrator area for approving senior citizen and service providers, reply to feedbacks and also view of other details.

#### User requirements

On the basis of the requirement survey conducted among different senior citizen and service providers, I reached the conclusion that it will be more beneficial to all users.

#### Project requirements

On the basis of the requirements demanded by the user the following project requirements are found out: A web application cum website is more helpful for the society

1. Having a website for search and request different service details.
2. Having a dynamic platform which is more beneficial to the aged people who need just a little outside help and also for the service providers.

### Feasibility Study

Feasibility is defined as the practical extent to which a project can be performed successfully. The objective of feasibility study is to establish the reasons for developing the software that is acceptable to the user, adaptable to changes and conformable to the established standards. The proposed System is feasible. Various type of feasibility that is commonly considered includes:

1. Economic Feasibility
2. Operational Feasibility
3. Technical Feasibility

### Economic Feasibility

Economic feasibility determines whether the proposed system is capable of generating financial gains for the organization. It involves cost incurred on the software development, cost of performing feasibility study, estimated cost of hardware and software and so on.

The proposed “SENIOR CARE SERVICE” is developed as a website needs less memory space for executing a web application. The machines running Windows operating system, Django, and MySQL server, all these are free of cost, which is very useful for senior citizen who need services. Senior citizen requires no special training for operating this web site. Also, the service providers can use it easily. Thus the proposed website cum Web app “SENIOR CARE SERVICES” meets the user requirements, required resources in efficient way and within minimum cost &time. By taking all these points into consideration, our proposed system is said to be less expensive and hence an economically feasible one.

### Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology which are required to accomplish the user requirements in the software within the allocated time and budget.

The proposed website “SENIOR CARE SERVICES” is effective and requires minimum of hardware and software at less cost and within more efficiency. The languages used so simply like Python and Django server. In the proposed system, data can be easily stored and manage using data management system software. The reports and results for various queries can be generated easily. Technical needs of the system include python as front-end and My SQL backend. Also, Wamp server software gives suitable environment for testing PYTHON & MYSQL on the local computer. All this software is freely available; hence the proposed system is technically feasible.

### Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and requirements. This feasibility is dependent on human resources and involves visualizing whether the software will operate after it is developed and be operative once it is launched. It also analyses whether users can adapt to new software.

The proposed “SENIOR CARE SERVICES” that develop here have much improved operational features, for that we used web application. It is an interactive website, easy to access from everywhere. It has very user-friendly interface so that there will not require any need for the training for its users. Do not need thorough on knowledge on the web application; user must only the basic net surfing knowledge. The proposed website provides lot of features through which senior citizen are able to register site and to see service details provided by the service provider and request the service etc. Hence it is easy to use, we can say that our proposed system is an operationally feasible one.

### Proposed System

SENIOR CARE SERVICES is a website which overcomes all the drawbacks of the existing system and provides a common and efficient platform for the senior citizens. In the proposed system, we can add the required services and resources to the site for helping the senior citizens. When we add our services and resource, we can also specify the location that helps the old people to find the resource near their location. The new generation those who are willing to provide services can join as a service provider.

#### The main areas covered in the Senior Care Services are

1. Web Side view
2. Admin Panel
3. Senior Citizen Login
4. Service Provider Login
5. Admin Activity

### Advantages

* + - Services can be search and request easily.
    - Proposed system is user friendly.
    - Reduced complexity
    - Service providers can use their free time as social service to the senior citizen according to their needs.
    - The proposed system is less time consuming.
    - Ensure the quality of services providing.
    - Effective interaction between Senior citizen, Service providers and Admin.

# SYSTEM SPECIFICATION

## SYSTEM SPECIFICATIONS

### Hardware Specification

Processor : Intel core i3 and above

Memory : Minimum 1 GB RAM

Hard Disk Drive : 100 GB

Keyboard : QWERTY

Components : Scroll Mouse

Display : 14inch Colour Monitor

### Software Specification

Operating System : Windows7 and above

Front End : Django Framework for PYTHON HTML, JavaScript, CSS

Browser : Any One

IDE : Visual Studio Code

Back End : MYSQL

Documentation : Microsoft Word 2007

# SYSTEM DESIGN

## SYSTEM DESIGN

The most creative and challenging phase of the system life cycle is the system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing candidate system. In system design we move from the logical to physical aspect of the life cycle. Design is the phase that indicates the final system. It is the solution, the translation of requirements in ways of meeting them. Firstly logical design was done where the outputs, inputs, databases and the procedures were formulated in a manner that meet the project requirements. After logical design physical construction of the system is done.

### Fundamental Design Concepts of System

The first step is to determine how the output is to be produced and in which format. The input data and master files have to be designed at the next step and finally the impact of the candidate system on the user and organization are documented and evaluated by the management. After identifying the problem and the limitations of the existing system, a detailed design of the proposed system is conducted. Free flow personal interview and reference to previous prepared manually were the only methods taken to collect necessary information. At present, the all organizations are on the path of computerization process.

Design is the phase that indicates the final system. It is the solution, the translation of requirements into ways of meeting them. In this phase the following elements were designed namely dataflow, data stores, processes, procedures. Firstly logical design was done where the outputs, inputs, databases and the procedures were formulated in a manner that meet the project requirements. After logical design physical construction of the system is done.

### Input Design

In the input design, user oriented inputs are converted in to a computer based format. The major approach to input design is the menu and the prompt design. In each alternative, the user option is predefined. The user has to input only the minimum data

required, which also helps in avoiding the errors that the user may make. The goal of input design is to make entry easy, logical and free from errors.

#### Input Forms are

* + - Login form
    - Senior citizen registration form
    - Service provider registration form
    - Service search form
    - Service request form
    - Request manage form
    - Adding service details form
    - Profile update form
    - Feedback form
    - Rating and payment form

### Output Design

In the output design, the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printers and the screen. Computer‟s output is the most important and direct source of information to the user, efficient, logical, output design should improve the systems relations with the user and help in decision-making.

As the outputs are the most important source of information to the user, better design should improve the system‟s relation and also should help in decision-making. The output device‟s capability, print capability, response time requirements etc. should also be considered form design elaborates the way output is presented and layout available for capturing information.

#### Output Forms are

* + - View service details form
    - View feedback form
    - View requests form
    - View ratings form
    - View payment form
    - View request status form

#### See Appendix A for screen shots

* 1. **Database Design**

In designing a database application you must set up not only the program‟s routines for maximum performance, but you must pay attention also to the physical layout of the data storage.

A good data base design does the following:

* Provides minimum search times when locating specific records.
* Stores the data in the most efficient manner possible to keep the database from growing large.
* Make data updates as easy as possible.
* Is flexible enough to allow inclusion of new functions.

A database typically has two components: the file holding the physical database and the database management system (DBMS) software that applications use to access data. Here we use tables for store details for the Senior Care Services. They are

* + senior\_register
  + provider\_register
  + login
  + request
  + services
  + rating
  + payment
  + message
  + feedback

### Normalization

Normalization technique is the main tool used for the designing of the tables in a database. At the beginning we will have only one table with all the required fields as a unit. In further decompositions we will get different levels of normalized tables with minimum redundancy and interdependency.

**First normal form (1NF)** lays the groundwork for an organized database design:

* Ensure that the each table has a primary key: minimal set of attributes which can uniquely identify a record.
* Eliminate repeating groups by defining key and non-key attributes appropriately.
* Atomicity: each attribute must contain as single value, not a set of values.

All the tables in the SENIOR CARE SERVICES have satisfied all these features so all the tables are in the first normal form.

**Second normal form (2NF)** is a normal form used in [database normalization](http://en.wikipedia.org/wiki/Database_normalization). Second normal form (2NF) is the second step in normalizing a database. A table that is in [first normal form](http://en.wikipedia.org/wiki/First_normal_form) (1NF) must meet additional criteria if it is to qualify for second normal form.

Specifically, a table is in 2NF [if and only if](http://en.wikipedia.org/wiki/If_and_only_if) it is in 1NF and no non-prime attribute is dependent on any proper [subset](http://en.wikipedia.org/wiki/Subset) of any [candidate key](http://en.wikipedia.org/wiki/Candidate_key) of the table. A non-prime attribute of a table is an attribute that is not a part of any candidate key of the table.

All the tables in the proposed system have satisfied all these features, So all the tables are in the second normal form.

#### See Appendix B for table design

* 1. **Data Flow Diagram**

The Data Flow Diagram (DFD) is a tool in the top-down approach. It moves from general requirements, illustrating the processes, transition and storage of data in the

system. In a DFD, processes are first identified and then the data flow between the processes are isolated and derived. Thus processes are the focal points of a DFD.

***Data flow diagram symbols:-***

**Source or Destination of data**

**Data Flow**

**Process**

**Storage**

**Steps to Construct Data Flow Diagrams**

*Four steps are commonly used to construct a DFD*

* Process should be named and numbered for easy reference. Each name should be representative of the process.
* The destination of flow is from top to bottom and from left to right.
* When a process is exploded in to lower level details they are numbered.
* The names of data stores, sources and destinations are written in capital letters.

#### See Appendix C for Data Flow Diagrams

* 1. **Unified Modelling Language Design**

The Unified Modelling Language (UML) is used to specify, visualize, modify, construct and document the artefacts of an object-oriented software intensive system under development. UML offers a standard way to visualize systems architectural blue prints.

### Use Case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

***The symbols are:***

Use case

Association

Actor

### Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Initial node

Final node



Decision node

Control Flow Activity

Fork Join

Fork is a black bar with one flow going into it and several leaving it. This denotes the beginning of parallel activity. Join is a black bar with several flows entering it and one leaving it. All flows going into the join must reach it before processing may continue. This denotes the end of parallel processing.

### Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence.

Messages written on horizontal arrows and the message name written above them, it displays interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn‟t have to wait for a response.

#### See Appendix IV for UML Diagrams

* 1. **Code Design**

The coding step transfers a detailed design of representation of software into a programming language realization. All the software engineering steps that have been presented up to this step are directed towards a final objective. Coding is a process that transforms design into a programming language. The translation process begins when a compiler accepts the source code as input and produces machine dependent object code as output. Compiler output is further translated into machine code. Language characteristics have an impact on the quality and efficiency of translation. Building the application refers to the coding step in software engineering process. I do this project by using PYTHON & HTML as front end language. Since it is a web based site the requirement of database is there and we are using MySQL Server as back end language to access and perform operations. Here JavaScript is also used as scripting languages. Also use WAMP as the web server for running the project locally.

### Python

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don‟t need to declare the type of variable because it is a dynamic typed language. For example, x=10, here x can be anything such as String, int etc.

#### Features in Python

There are many features in Python, some of which are discussed below –

#### Easy to code:

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer- friendly language.

#### Free and Open Source:

Python language is freely available at official website and you can download it. Since, it is open source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

#### Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

#### GUI Programming Support:

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python.PyQt5 is the most popular option for creating graphical apps with Python.

#### High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

#### Extensible feature:

Python is an Extensible language. We can write some python code into c or c++ language and also we can compile that code in c/c++ language.

#### Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

#### Python is Integrated language:

Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

#### Interpreted Language:

Python is an Interpreted Language. Because python code is executed line by line at a time. Like other language c, c++, java etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

#### Large Standard Library:

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

#### Dynamically Typed Language:

Python is dynamically-typed language. That means the type (for example- int, double, long etc.) for a variable is decided at run time not in advance. Because of this feature we don‟t need to specify the type of variable.

### Django

Django is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) based [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [web framework,](https://en.wikipedia.org/wiki/Web_framework) which follows the model-template-view (MTV) [architectural pattern.](https://en.wikipedia.org/wiki/Architectural_pattern_(computer_science)) Django's primary goal is to ease the creation of complex, database driven websites. The framework emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability) and pluggability of components, less code, low coupling, rapid development, and the principle of [don't repeat yourself.](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself) Python is used throughout, even for settings files and data models.

Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create%2C_read%2C_update_and_delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Type_introspection) and configured via admin models.

Despite having its own nomenclature, such as naming the callable objects generating the [HTTP](https://en.wikipedia.org/wiki/HTTP) responses views, the core Django framework can be seen as an [MVC](https://en.wikipedia.org/wiki/Model-view-controller) architecture. It consists of an [object-relational mapper](https://en.wikipedia.org/wiki/Object-relational_mapping) (ORM) that mediates between [data models](https://en.wikipedia.org/wiki/Data_modeling) (defined as Python classes) and a [relational database](https://en.wikipedia.org/wiki/Relational_database) (Model), a system for processing HTTP requests with a [web templating system](https://en.wikipedia.org/wiki/Web_template_system) (View), and a [regular-expression](https://en.wikipedia.org/wiki/Regular_expression)-based [URL](https://en.wikipedia.org/wiki/Uniform_Resource_Locator) dispatcher (Controller).

Also included in the core framework are:

* A lightweight and standalone web server for development and testing.
* A form serialization and validation system that can translate between [HTML](https://en.wikipedia.org/wiki/HTML) forms and values suitable for storage in the database.
* A template system that utilizes the concept of [inheritance](https://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming)) borrowed from object-oriented programming.
* A [caching](https://en.wikipedia.org/wiki/Web_cache) framework that can use any of several cache methods
* Support for [middleware](https://en.wikipedia.org/wiki/Middleware) classes that can intervene at various stages of request processing and carry out custom functions
* An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals
* An [internationalization](https://en.wikipedia.org/wiki/Internationalization_and_localization) system, including translations of Django's own components into a variety of languages
* A [serialization](https://en.wikipedia.org/wiki/Serialization) system that can produce and read [XML](https://en.wikipedia.org/wiki/XML) and/or [JSON](https://en.wikipedia.org/wiki/JSON) representations of Django model instances
* A system for extending the capabilities of the template engine
* An interface to Python's built-[in unit test](https://en.wikipedia.org/wiki/Unit_test) framework
* Django REST framework is a powerful and flexible toolkit for building Web APIs

### Java Script

Java script is a scripting language that can be used to create client side scripts and server side scripts. Client side scripts are executed in the browser while server side scripts are executed on a server. That is JavaScript is an object-based scripting language for developing client based and server based internet applications. We can insert JavaScript statements directly into an HTML page. When the page is displayed in the browser, the JavaScript statements are interpreted and executed by the browser. JavaScript statements can recognize and respond to user events such as mouse clicks or system generated events and so on. So you can change the content and position of the elements on the page dynamically, in response to user interaction.

When the client requests an HTML page that includes a client side Script, the server forwards the full content of the HTML document- the JavaScript statements and the HTML content. When the browser receives the document, it executes the HTML and JavaScript statements without any interaction with the server while both client-side JavaScript and server-side JavaScript have the same core language; each also has additional features relevant to the environment. That is, client-side JavaScript includes predefined objects that can be used only in the browser. Server-side JavaScript contain predefined objects that can be used in server-side application.

### Hyper Text Mark Up Language

An HTML file is a text file containing small markup tags. These tags tell the web browser how to display the page. An HTML file must have an htm or html file extension. An HTML file can be created by using a simple text editor. HTML documents are text files made up of HTML elements e.g.: <html>, <body>. HTML elements are defined using HTML tags.

HTML tags are used to markup HTML elements. The two characters surround HTML tags <and>. The surrounding characters are called angle brackets. HTML tags normally come in pairs like <b> and </b>. The first tag in a pair is the start tag: the second tag is the end tag. The text between the start and the end tag is the element content. HTML tags are not case sensitive; <b> means the same as <B>.

### MySQL

Relational database systems are the most important database systems used in the software industry today. One of the most outstanding systems is MySQL.

The important aspects of SQL Server are:

* + MySQL is easy to use.
  + Embedded database library.
  + Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) open source web application software stack (and other "[AMP](https://en.wikipedia.org/wiki/List_of_AMP_packages)" stacks). LAMP is an acronym for "[Linux](https://en.wikipedia.org/wiki/Linux), [Apache,](https://en.wikipedia.org/wiki/Apache_HTTP_Server) MySQL, [Perl](https://en.wikipedia.org/wiki/Perl)/[PHP](https://en.wikipedia.org/wiki/PHP)/[Python.](https://en.wikipedia.org/wiki/Python_(programming_language))" [Free-](https://en.wikipedia.org/wiki/Free_software) [software](https://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL.

On all platforms except Windows, MySQL ships with no [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](https://en.wikipedia.org/wiki/Command_line) tools, or install [MySQL workbench](http://dev.mysql.com/downloads/workbench/) via a separate download. Many third party GUI tools are also available.

# SYSTEM TESTING

## SYSTEM TESTING

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system

* + - Unit testing
    - Integration testing
    - Validation testing
    - Output testing
    - User acceptance testing.

### Unit Testing

It focuses effort on the smallest unit of software designs in the module. This is also known as “Module Testing”. To check whether each module in the software is proper and it gives desired outputs to the given inputs. All validations and conditions are tested in the module in the first unit. Control paths are tested to ensure the information properly flows into, and out of the program unit under test. Boundary conditions are tested to ensure that they operate at boundaries.

The proposed system SENIOR CARE SERVICES has 3 different modules and hence each module is coded and tested differently. In order to provide strong security or administrator view sessions are added in the code. So testing has to be done for each activity of admin.

The other users of SENIOR CARE SERVICES are Senior citizen and Service provider. Testing has to done while each person login in to the site so that no one can use site with different username and password.

The main objective of the site is to provide an interactive space for the users and hence each of the functions added in the platform has to be tested and need to check whether the database is linked correctly.

### Integration Testing

The modules are integrated to form complete software package. It addresses the issues associated with given problem of verification and program construction. Test that part of the system at some level work together correctly the purpose of integration is to verify functional, performance and reliability requirements placed on major design items. This testing is conducted on the basis of modules. The integration testing is performed to detect design errors by focusing on testing the interconnection between modules. The objective is to take the unit tested modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complete the isolation. Here We Integrate the SENIOR CARE SERVICES system into a complete website cum web application.

### System Testing

When a system is developed, it is hoped that it perform properly. In practice, however some errors always occur .The main purpose of testing an information system is to find the errors and correct them. A successful test is one, which finds an error.

The main objectives of system testing are:

* + - * To ensure during operating the system will perform as per specification.
      * To make sure that the system meets user‟s requirements during operation.
      * To verify that the controls incorporated in the system function as intended.

### Acceptance Testing

This is the key factor for the success of any system. User acceptance is done by constantly keeping in touch with prospective system at the time of development and making changes whenever required. This is done with regard to the input and output screen designs.

#### The Different Test Cases are

* Guarantees that all independent parts within a module have been exercised at least once. In SENIOR CARE SERVICES, the independent parts of the system are tested and it fulfills the acceptance of the user requirements.
* Exercise all logical design on their true/false side
* Exercise all loops at their boundaries and within their operational bounds
* Each module was tested and modules were linked and integration was carried out

#### Levels of Testing

Client needs - Acceptance testing Requirements - System testing Design - Integration testing

Code - Unit testing

#### Testing For the Proposed System

Each module in the proposed system undergone unit testing that evaluated each module of the system. When all the forms provided exactly correct output each of them were integrated to form a particular module. Necessary validation including numeric validation, string validation, e-mail validation and mandatory field validation were done to ensure reliability of output. Thus our project covered all the major phases of testing proves to be a system that functions according to user requirements.

# SYSTEM IMPLEMENTATION AND

**MAINTANANCE**

## SYSTEM IMPLEMENTATION AND MAINTANANCE

### Implementation

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

#### Implementation Plan

The implementation plan includes a description of all the activities that must occur to implement the new system and to put into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps:

* + - * List all files required for implementation.
      * Identify all data required to build new files during the implementation.
      * List all new documents and procedures that go into the new system.

The implementation plan should anticipate possible problems and must be able to deal with them. The usual problems may be missing documents; mixed data formats between current and files, errors in data translation, missing data etc. So the Proposed system satisfies all the implementation plan properly.

### Maintenance

Software maintenance is the process of modifying a software system or component after its delivery in order to correct faults, improve the performance and the other attributes, or to adapt to the changed environment. Maintenance covers a wide range of activities including correcting the coding and design errors, updating the documentation and test data, and upgrading the user support. There is an aging process that calls for periodic maintenance

of hardware software. Maintenance is always necessary to keep the software usable and useful. Hardware also requires periodic maintenance activities can be classified into:

* Corrective Maintenance
* Perceptive Maintenance
* Adaptive Maintenance
  + - **Correction:-**Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects.
    - **Adaptation:-**Over time, the original environment (e.g. CPU, Operating System, Business Rules, External Product Characteristics) for which was developed was likely to change. Adaptive maintenance results in the modification to the software to accommodate changes to its external environment.
    - **Enhancement:-**As the software is used, the customer/user will recognize additional functions that will provide benefit. Perfective maintenance extends the software beyond its original functional requirements.
    - **Prevention:-**Computer software deteriorates due to change, and because of this, preventive maintenance, often called software reengineering, must be conducted to enable preventive maintenance makes changes to computer programs. So that they can be more easily corrected, adapted and enhanced.

# CONCLUSION

## CONCLUSION

Senior care services is not only a website, it is a web application that provides benefits in the society. The system will help the old people by providing what they needs. Also, the service providers can use their free time for a social service by using this site. All the knowledge I gained is fully applied in the design of the mentioned system. All the suggestions forwarded in the software proposal have been completed. This system is developed in such a way that the modules developed in the future can be linked easily to the system, without affecting the existing system, since it provides a hierarchical structure.

Top down programming approach has been adopted while developing the project; each task is divided separate modules. Hence the modification and enhancement can be easily made without affecting any other part of the program. This system has been developed to satisfy the user needs. The entire system is user friendly. The performance of the system is provided efficiently. The system was tested with all possible test data and was found to have an affective planning of the functions or processes with a high degree of accuracy and user friendliness.

# SCOPE FOR FUTURE ENHANCEMENT

## FUTURE ENHANCEMENTS

The SENIOR CARE SERVICES is a socially relevant application for the senior citizen. The system is developed in Django framework, HTML, MySQL and JS which makes the system more reliable and compatible with the other environments. The application proves better extensibility and flexibility for future enhancements. Any further requirement application is possible with the features guaranteed. The design of this software is in such a way that the addition of any new module if necessary is possible without affecting the integrity of the present system.

For the future we can implement facilities like

* + - * Free online consultation via chat within site, which assist the users when having a slew of questions in order to request a service or not.
      * Service providers search other providers and can make a group and do service activities together.
      * Track the live location of service provider on the requested service time.
      * Add more organizations which provide services like Pakal veedu.
      * Add notification features for informing Senior citizen.

# APPENDIX

**Appendix B: Database Design**

**Table: login**

**primary key: login\_id**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| login\_id | Int(11) | Login id |
| userid | Int(11) | User id |
| usertype | Varchar(20) | Type of user |
| username | Varchar(20) | User name |
| password | Varchar(20) | password |

**Table: senior\_register Primary key: senior\_id**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| senior\_id | Int(11) | Senior citizen id |
| name | Varchar(20) | Senior citizen name |
| age | Int(11) | Age of senior citizen |
| gender | Varchar(20) | Gender of senior citizen |
| aadhar\_id | Int(20) | Aadhar id of senior citizen |
| uploadphoto | Varchar(500) | Senior citizen photo |
| email | Varchar(20) | Senior citizen email |
| phonenumber | Varchar(20) | Senior citizen phone number |
| housename | Varchar(30) | Senior citizen House name |
| streetname | Varchar(30) | Senior citizen street name |
| city | Varchar(20) | Senior citizen city |
| state | Varchar(20) | Senior citizen state |
| pincode | int(20) | Senior citizen pincode |
| password | Varchar(20) | Password |
| status | Varchar(20) | Approval status |

**Table: provider\_register Primary key: provider\_id**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| provider\_id | Int(11) | Service provider id |
| Name | Varchar(20) | Service provider name |
| Email | Varchar(20) | Service provider email |
| gender | Varchar(20) | Service provider gender |
| phonenumber | Varchar(20) | Service provider Phone number |
| aadhar\_id | Varchar(20) | Aadhar id of Service provider |
| uploadphoto | Varchar(500) | Photo of service provider |
| housename | Varchar(20) | Service provider House name |
| streetname | Varchar(20) | Service provider street name |
| City | Varchar(20) | Service provider city |
| State | Varchar(20) | Service provider state |
| pincode | Int(11) | Service provider pincode |
| designation | Varchar(20) | Service provider designation |
| designation\_proof\_id | Int(11) | Designation proof id |
| designation\_proof\_type | Varchar(10) | Designation proof type |
| service1 | Varchar(20) | Providing service1 |
| service2 | Varchar(20) | Providing service2 |
| service3 | Varchar(20) | Providing service3 |
| service4 | Varchar(20) | Providing service4 |

|  |  |  |
| --- | --- | --- |
| uploadproof | Varchar(500) | Designation proof |
| Password | Varchar(20) | Service provider password |
| status | Varchar(20) | Approval status |

**Table: request**

**Primary key: request\_id**

**Foreign key: service\_id, senior\_id, provider\_id**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| request\_id | Int(11) | Request id |
| description | Varchar(100) | Request description |
| senior\_id | Varchar(11) | Requesting senior citizen id |
| wanted\_date | Varchar(35) | Wanted date |
| requested\_date | Varchar(35) | Requesting date |
| phone | Varchar(10) | Phone number |
| service\_id | Varchar(11) | Service id |
| provider\_id | Varchar(11) | Service provider id |
| Status | Varchar(10) | Request status |

**Table: service**

**Primary key: service\_id**

**Foreign key: senior\_id, provider\_id, request\_id**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| service\_id | Int(11) | Service id |
| provider\_id | Varchar(11) | Service provider id |
| service\_name | Varchar(100) | Service name |
| experience | Varchar(100) | Experience year |
| availability | Varchar(100) | Availability |
| cno | Varchar(50) | Contact number |
| amount | Varchar(60) | Amount |
| location | Varchar(30) | Location |
| Status | Varchar(20) | Availability status |

**Table: rating Primary key: rate\_id**

**Foreign key: senior\_id, provider\_id, request\_id**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| rate\_id | Int(11) | Rating id |
| Title | Varchar(35) | Title |
| comments | Varchar(200) | Comments |
| service\_id | Varchar(11) | Service id |
| Senior\_id | Varchar(11) | Senior citizen id |

**Table: feedback Primary key: feed\_id Foreign key: uid**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| feed\_id | Int(11) | Feedback id |
| Uid | Varchar(100) | User id |
| Usertype | Varchar(100) | User type |
| feed\_date | Date | Feedback date |
| Title | Varchar(50) | Title for feedback |
| Feedback | Varchar(100) | Feedback |
| Reply | Int(50) | Feedback reply |

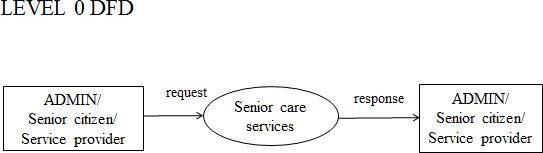
**Table: payment**

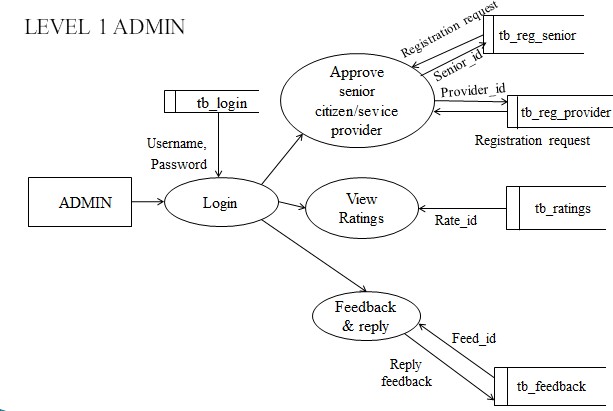
**Primary key: Payment\_id**

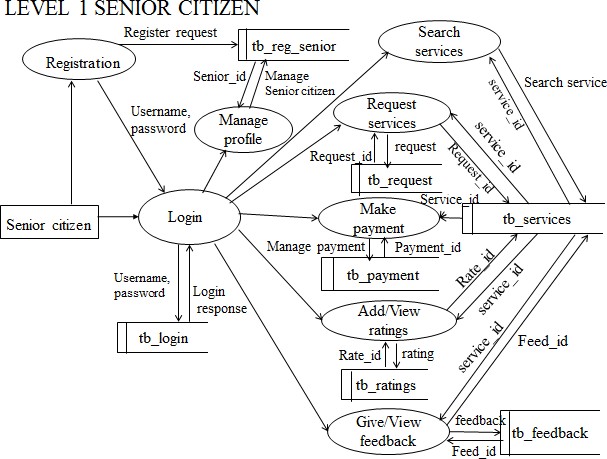
**Foreign key: Senior\_id, Provider\_id, Service\_id**

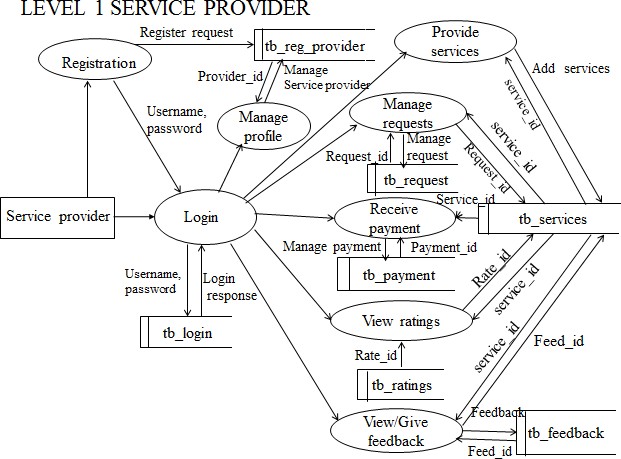
|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| payment\_id | Int(11) | Payment id |
| senior\_id | Varchar(50) | Senior citizen id |
| amount | Varchar(50) | amount |
| provider\_id | Varchar(50) | Amount receiving service provider id |
| requestid | Int(10) | Request id |
| pdate | Date | Payment date |
| service\_id | Varchar(50) | Service id |

**Appendix C: Data Flow Diagrams**





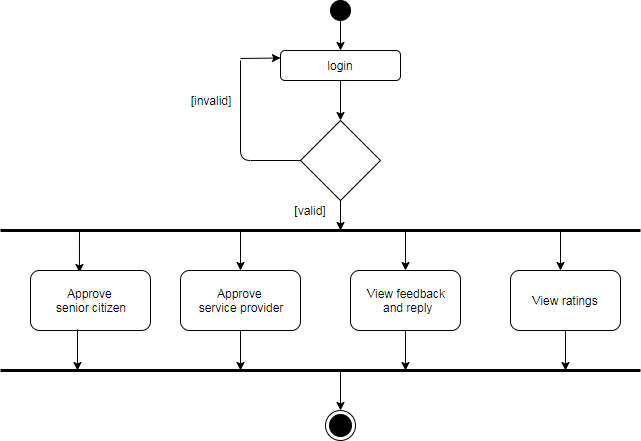




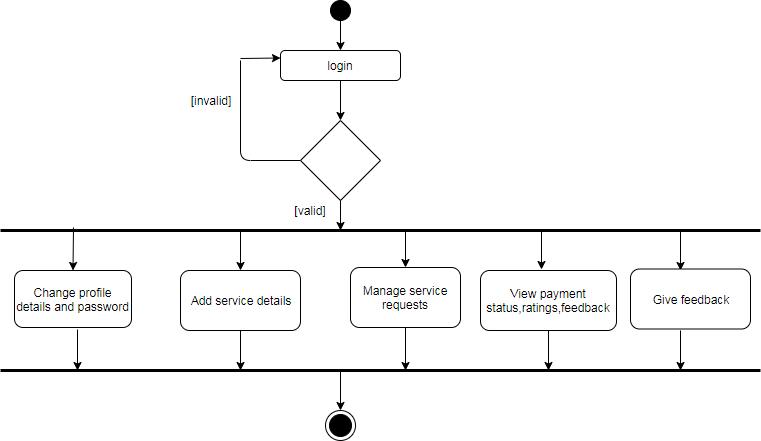
**Appendix D: UML Diagram**

**Activity Diagram**

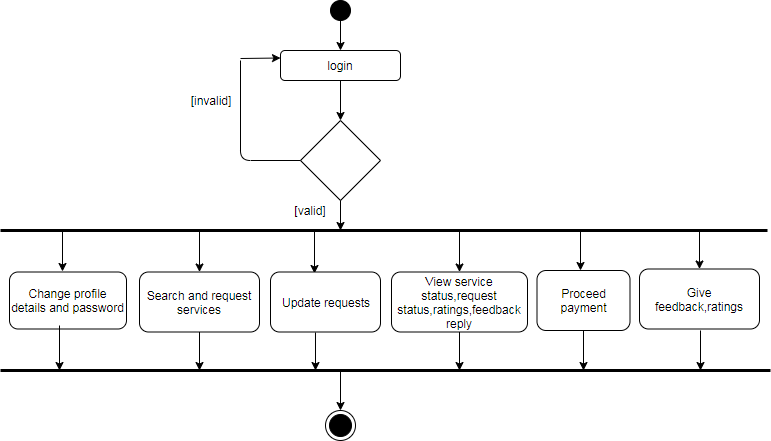
**ADMIN**



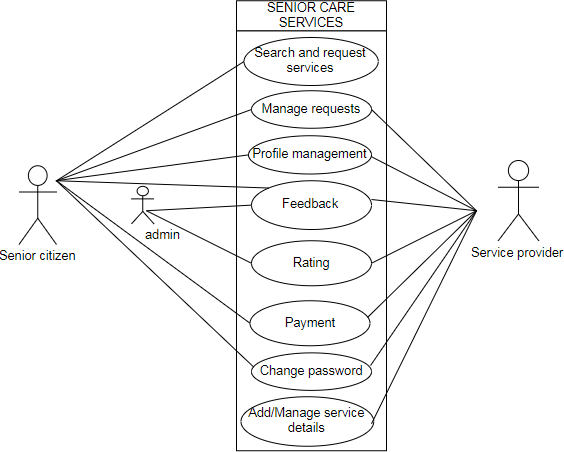
**SERVICE PROVIDER**

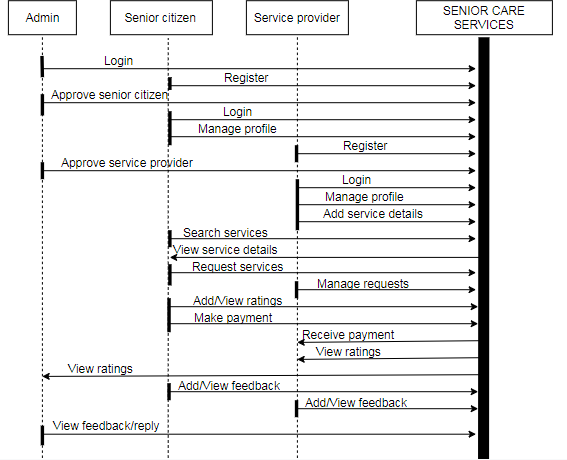


**SENIOR CITIZEN**



**Use case Diagram**





**Sequence Diagram**

# BIBLIOGRAPHY

## 10. BIBLIOGRAPHY

### Referred Books

1. Ian Somerville, Software Engineering, Pearson Education Asia, 6thEdition.
2. Awad, System Analysis and Design, Galgotia, 2ndEdition.
3. Elmasri, Navathe, Fundamentals of Database Systems, Pearson Education Asia, 6th Edition.

### Referred Websites

* + - [www.stackoverflow.com](http://www.stackoverflow.com/)
    - [www.python.org](http://www.python.org/)
    - [www.w3layouts.com](http://www.w3layouts.com/)
    - [www.codeproject.com](http://www.codeproject.com/)
    - [http://www.w3schools.com](http://www.w3schools.com/)
    - [http://www.djangoproject.com](http://www.djangoproject.com/)
    - [http://realpython.com](http://realpython.com/)