**Error! Unknown document property name.**

**Error! Unknown document property name.**

High Level Design & Low Level Design

The purpose of this document is to provide with a template for documenting both HLD & LLD

**Document Control :**

| **Project Revision History** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| 19-12-2022 | 0.1 | 19-12-2022\_LINUX17NOVB1\_SPRINT-1\_B | Initial Draft on Introduction and Designing | | | |  | |
| 20-12-2022 | 0.2 | 19-12-2022\_LINUX17NOVB1\_SPRINT-1\_B | Modification on Designing and system Architecture | | | |  | |
| 21-12-2022 | 0.3 | 19-12-2022\_LINUX17NOVB1\_SPRINT-1\_B | System design , Mapping and Reference | | | |  | |

[**1. Introduction**](#_heading=h.3znysh7) 5

[1.1. Intended Audience](#_heading=h.2et92p0) 5

[1.2. Acronyms/Abbreviations](#_heading=h.tyjcwt) 5

[1.3. Project Purpose](#_heading=h.3dy6vkm) 5

[1.4. Key Project Objectives](#_heading=h.1t3h5sf) 5

[1.5. Project Scope and Limitation](#_heading=h.4d34og8) 6

[1.5.1. In Scope](#_heading=h.2s8eyo1) 6

[1.5.2. Out of scope](#_heading=h.17dp8vu) 6

[1.6. Functional Overview](#_heading=h.3rdcrjn) 6

[1.7. Assumptions, Dependencies & Constraints](#_heading=h.26in1rg) 6

[1.8. Risks](#_heading=h.lnxbz9) 7

[**2. Design Overview**](#_heading=h.35nkun2) 7

[2.1. Design Objectives](#_heading=h.1ksv4uv) 7

[2.1.1. Recommended Architecture](#_heading=h.44sinio) 7

[2.2. Architectural Strategies](#_heading=h.2jxsxqh) 8

[2.2.1. Design Alternative](#_heading=h.z337ya) 8

[2.2.2. Reuse of Existing Common Services/Utilities](#_heading=h.3j2qqm3) 8

[2.2.3. Creation of New Common Services/Utilities](#_heading=h.1y810tw) 8

[2.2.4. User Interface Paradigms](#_heading=h.4i7ojhp) 8

[2.2.5. System Interface Paradigms](#_heading=h.2xcytpi) 8

[2.2.6. Error Detection / Exceptional Handling](#_heading=h.1ci93xb) 8

[2.2.7. Memory Management](#_heading=h.3whwml4) 8

[2.2.8. Performance](#_heading=h.2bn6wsx) 8

[2.2.9. Security](#_heading=h.qsh70q) 9

[2.2.10. Concurrency and Synchronization](#_heading=h.3as4poj) 9

[2.2.11. Housekeeping and Maintenance](#_heading=h.1pxezwc) 9

[**3. System Architecture**](#_heading=h.49x2ik5) 10

[3.1. System Architecture Diagram. (Not Necessary)](#_heading=h.2p2csry) 10

[3.2. System Use-Cases](#_heading=h.147n2zr) 11

[3.3. Subsystem Architecture](#_heading=h.3o7alnk) 11

[3.4. System Interfaces](#_heading=h.23ckvvd) 11

[3.4.1. Internal Interfaces](#_heading=h.ihv636) 11

[3.4.2. External Interfaces](#_heading=h.32hioqz) 12

[**4. Detailed System Design**](#_heading=h.1hmsyys) 12

[4.1. Key Entities](#_heading=h.41mghml) 13

[4.2. Detailed-Level Database Design 13](#_heading=h.2grqrue)

[4.2.1. Data Mapping Information](#_heading=h.vx1227) 14

[4.2.2. Data Conversion](#_heading=h.3fwokq0) 14

[4.3. Archival and retention requirements](#_heading=h.1v1yuxt) 14

[4.4. Disaster and Failure Recovery](#_heading=h.4f1mdlm) 15

[4.5. Business Process workflow](#_heading=h.2u6wntf) 15

[4.6. Business Process Modeling and Management (as applicable)](#_heading=h.19c6y18) 15

[4.7. Business Logic](#_heading=h.3tbugp1) 15

[4.8. Variables](#_heading=h.28h4qwu) 15

[4.9. Activity / Class Diagrams (as applicable)](#_heading=h.nmf14n) 15

[4.10. Data Migration](#_heading=h.37m2jsg) 15

[4.10.1. Architectural Representation](#_heading=h.1mrcu09) 15

[4.10.2. Architectural Goals and Constraints](#_heading=h.46r0co2) 15

[4.10.3. Logical View](#_heading=h.2lwamvv) 15

[4.10.4. Architecturally Significant Design Packages](#_heading=h.111kx3o) 15

[4.10.5. Data model](#_heading=h.3l18frh) 15

[4.10.6. Deployment View](#_heading=h.1egqt2p) 15

[**5. Environment Description**](#_heading=h.3ygebqi) 15

[5.1. Time Zone Support](#_heading=h.2dlolyb) 16

[5.2. Language Support](#_heading=h.sqyw64) 16

[5.3. User Desktop Requirements](#_heading=h.3cqmetx) 16

[5.4. Server-Side Requirements](#_heading=h.1rvwp1q) 16

[5.4.1. Deployment Considerations](#_heading=h.4bvk7pj) 16

[5.4.2. Application Server Disk Space](#_heading=h.2r0uhxc) 16

[5.4.3. Database Server Disk Space](#_heading=h.1664s55) 16

[5.4.4. Integration Requirements](#_heading=h.3q5sasy) 16

[5.4.5. Jobs](#_heading=h.25b2l0r) 16

[5.4.6. Network](#_heading=h.kgcv8k) 16

[5.4.7. Others](#_heading=h.34g0dwd) 16

[5.5. Configuration](#_heading=h.1jlao46) 16

[5.5.1. Operating System](#_heading=h.43ky6rz) 17

[5.5.2. Database](#_heading=h.2iq8gzs) 17

[5.5.3. Network](#_heading=h.xvir7l) 17

[5.5.4. Desktop](#_heading=h.3hv69ve) 17

[**6. References**](#_heading=h.1x0gk37) 17

[**7. Appendix**](#_heading=h.4h042r0) 17

# 

# Introduction

DND stands for Do Not Disturb. DND is an initiative by TRAI (Telecom Regulatory Authority of India) to save telecom consumers from unsolicited calls/SMS. If a user registers his/her mobile number, then it would be illegal for telemarketers to send any sort of unsolicited SMS or calls on those numbers.

A Do Not Disturb feature allows you to choose when and how you want to be updated with incoming calls or messages. With this feature, it’s easier to mute all calls and mobile devices .

If you’re busy and don’t want to be disturbed by calls , you can activate the do not disturb feature to shut off all, or part, of the notification and alerts that would normally come to you.

## Intended Audience

| User | Anyone with a sim & mobile phone can activate or use this service. |
| --- | --- |

## Acronyms/Abbreviations

| DND | Do Not Disturb |
| --- | --- |
| TRAI | Telecom Regulatory Authority of India |

## Project Purpose

The main aim of this project is that subscribers can block/stop unwanted calls. The subscriber can activate the DND service either at global level or at selective level. In case of global, the subscriber will not receive any call, while in case of selective, he will not receive any call from the selected number or the selected groups. Once the DND service is activated, the subscriber would be informed via a confirmation message. There will be provision for deactivating the DND service, once deactivated; the subscriber will receive all the incoming calls.

## Key Project Objectives

The key objectives are-

* To provide users or subscribers with the DND service to avoid any unsolicited calls,or messages.
* To provide users accessibility to set DND services globally as well as for selective numbers or groups.
* To inform the subscriber about DND activation via a confirmation message.
* After deactivating , the user can receive all the calls.

## Project Scope and Limitation

* Our project aims for easy management of distraction from unknown calls or from promotional type calls.
* The limitation of our project is that users cannot opt for two services at once before starting a new service. Users have to discontinue from the previous service.

### 

### 1.5.1. In Scope

It may help the users from not getting distracted via these calls or messages.

### 1.5.2. Out of scope

* It may ask the user first if he/she is a registered user or not before providing any DND service for that particular number.
* Easy to use by the user and understood by the user.
* Increases productivity.

## 

## Functional Overview

* Our project helps the user from not receiving or blocking any kind of unwanted calls.
* Users can choose any type of service-global or selective .
* He/she can only activate any one service at a time.
* If there is a need to change the level of DND service, then it is mandatory to first deactivate the currently active DND service and then try activation with a new level.

## Assumptions, Dependencies & Constraints

* Related software or hardware
* Operating systems
* End-user characteristics
* Possible and/or probable changes in functionality]
* With the use of C language and Linux OS we can create this DND project.
* This project will help the subscribers in blocking the unwanted calls.
* If a user wants to subscribe for any other service in the same type then he/she has to deactivate from the previous service first and then can use any other service.

## Risks

* Users must be a registered user before opting for any kind of service.
* Also we have to keep check on the number of services a user can take
* Users cannot take two services on any level.

# Design Overview

* The project is designed by dividing the program into multiple c files i.e dnd.c , test.c
* These files contain multiple functions & declarations that are used for different cases.These functions are called based on the user input and his/her requirements.
* In the dnd.c file we have created multiple functions for different use cases.

## Design Objective

Following are the operations that are performed in this file.

* Each case performs different operations
* Registration of user
* Selection of level
* Global or Selective

### Recommended Architecture

In our project DND Implementation subscriber can block/stop unwanted calls. The subscriber can activate the DND service either at global level or at selective level. In case of global, the subscriber will not receive any call, while in case of selective, he will not receive any call from the selected number or the selected groups. Once the DND service is activated, the subscriber would be informed via a confirmation message. There will be provision for deactivating the DND service, once deactivated; the subscriber will receive all the incoming calls.

## Architectural Strategies

The Architectural Strategy used in this project is used for applying DND (Do Not Disturb) to stop receiving unwanted or promotional calls at Global Level or Selective Level.

### 2.2.1. Design Alternative

Not Applicable

### 2.2.2. Reuse of Existing Common Services/Utilities

In our project we are calling the menu after every selection of choices.

### 2.2.3. Creation of New Common Services/Utilities

In our project we are creating a menu function for users so that users can search files according to his/her preferences.

### 2.2.4. User Interface Paradigms

This service helps the subscribers to easily visualize the levels and then according to their requirements they can choose the available service.

### 2.2.5. System Interface Paradigms

Good design is good business.If the system has interactive interface and it satisfies the user requirements then its a plus point for developers .This DND project can provide the subscribers with the appropriate services.

### 2.2.6. Error Detection / Exceptional Handling

These elements are the main factors for cleanliness or to provide quality code.

* Maintainability- Allows us to easily fix bugs without the fear of breaking current functionality.
* Extensibility-Allows us to easily add to our code base, implementing new or changed requirements without breaking existing functionality.
* Readability-Allows us to easily read code and discover its purpose without spending too much time digging.

### 2.2.7. Memory Management

In our project we have used file handling for managing memory and usage of functions which helps in code reusability.

### 2.2.8. Performance

* User can activate DND services
* User can adopt global or selective DND services
* After DND is activated it is provided with the message “DND Activated”.
* Users can also change DND services.

### 2.2.9. Security

* In our project unregistered users can not take the DND service.
* DND deactivates unnecessary calls/sms.

### 2.2.10. Concurrency and Synchronization

Not Applicable

### 2.2.11. Housekeeping and Maintenance

Not Applicable

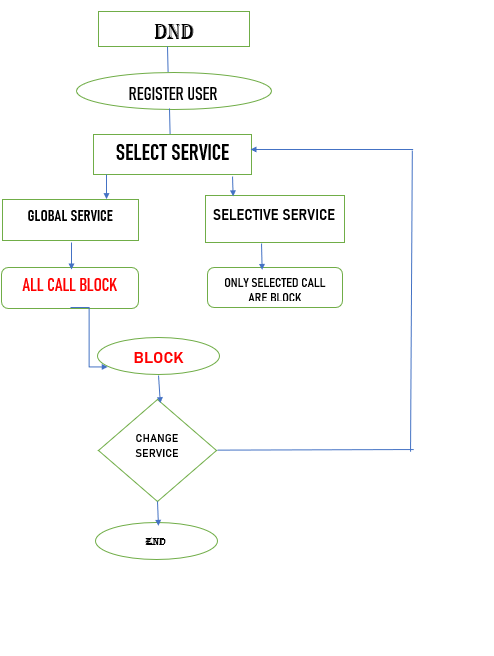
# System Architecture

The Register user is provided with two options Global and Selective service if the user wishes to go with Global in that case all the calls are blocked . In selective it is upto to the user what option he chooses to call, SMS .

If the user wants to block calls only then only calls will be blocked and if any other user wants to block SMS then SMS will be blocked according to user choice.

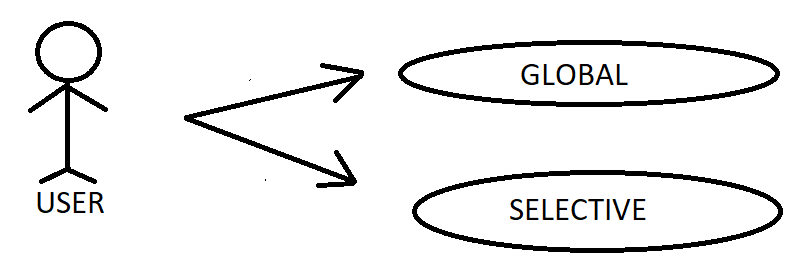
After DND is activated if the user wants to call or SMS the blocked user he gets the message DND activated .

## System Architecture Diagram

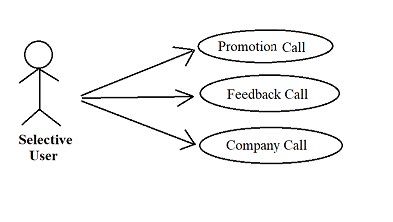
**

## 

## System Use-Cases



## Subsystem Architecture



## System Interfaces

- The User interface depends upon the user.

- It depend upon the choice made by user according to his/her requirements

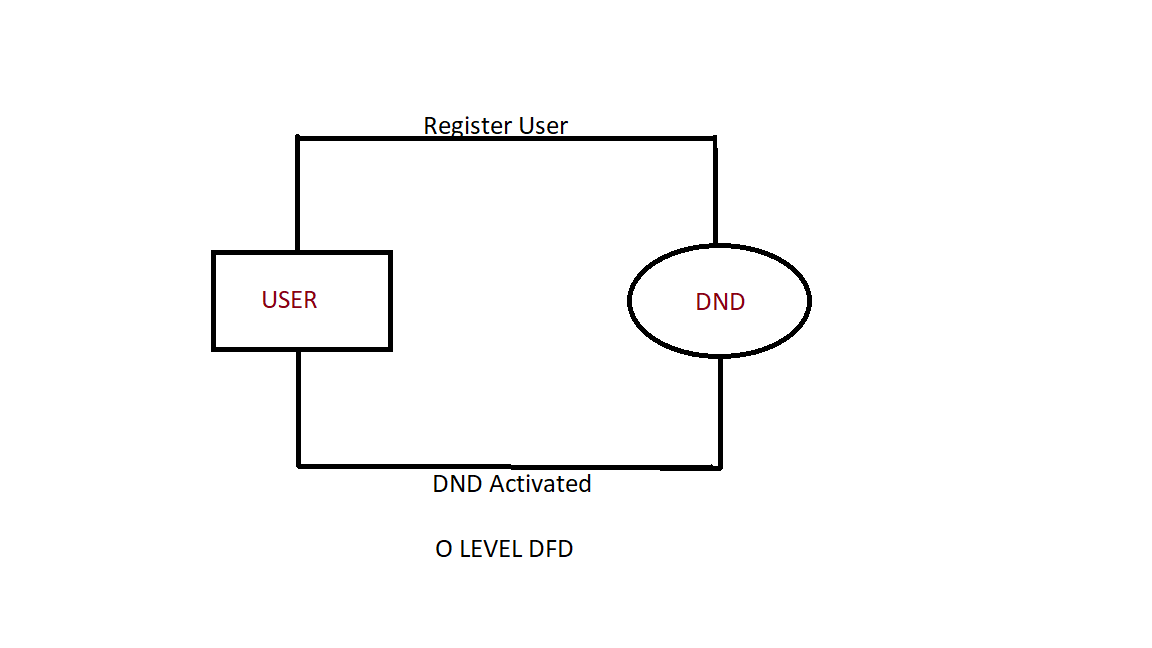
.

**3.4.1. Internal Interfaces**

As an Internal Interface we are using Ubuntu Linux distribution. It is an operating system that is made up of a collection of software based on Linux kernel or you can say distribution contains the Linux kernel and supporting libraries and software.

**3.4.2. External Interfaces**

User will give input via mobile phone.

****

# Detailed System Design

* Dealing with a constant stream of notifications is annoying . DND Mode helps us handle things ,to block unnecessary calls like operator calls,loan related calls not only calls but SMS also we can block the unnecessary message that we no longer need.
* DND stands for do not disturb, DND is an initiative by TRAI (Telecom Regulatory Authority of India) to save telecom consumers from unsolicited calls/SMS. If a user registers his/her mobile number, then it would be illegal for telemarketers to send any sort of unsolicited SMS or calls on that number.
* The project is designed by dividing the program into two files i.e dnd.c and txt.c.
* The register user will be provided to choose from two choices

1.Global

2. Selective

## Key Entities

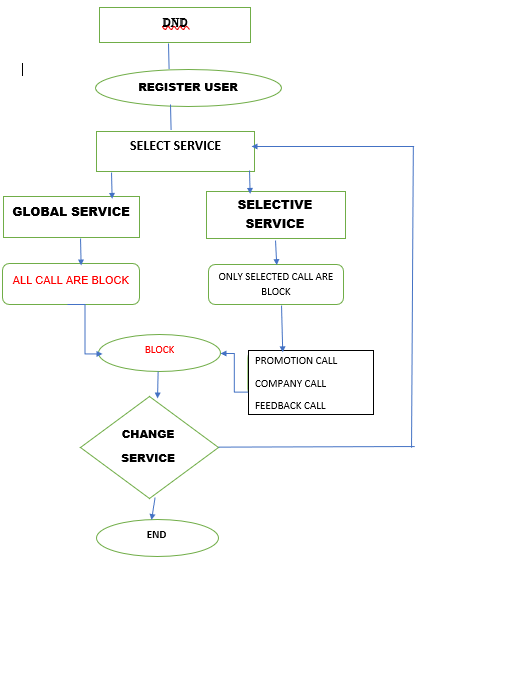
**GLOBAL USER()**

* USER NAME
* USER PHONENUMBER

**SELECTIVE USER()**

* USER NAME
* USER PHONE NUMBER
* BLOCK PROMOTION CALLS
* BLOCK FEEDBACK CALLS
* BLOCK COMPANY CALLS

## Detailed-Level Database Design



### 4.2.1. Data Mapping Information

**4.2.1.1. Registration()**

At first the user has to register his or herself using his/her name and phone number.

**4.2.1. 2.**  **Activate DND()**

After registration, user will be provided with two services i.e(Global and Selective)

**a.Global**- In Global service all the calls will be blocked.

**b.Selective**- In Selective service users will be given three choices( Promotional call, Feedback call,Company Call) so that users can choose the service according to their choices.

**4.2.1.3. Dial()**

It is used to check DND service is activated or not on a registered number .If the number is registered then the user will be provided with the message the DND is activated on this number.

**4.2.1.4**. **Change DND service()**

If the user is already a registered user then he/she has to deactivate the previous activated service after that he/she can change the service.

**4.2.1.5 .Display()**

* + - It will show the name of the user, service of the user(global or selective) phone number of the user and DND status.
    - Deactivate DND()
    - Users can deactivate the DND service.

**4.2.1.6. Deactivate DND()**

All The services taken by user can be deactivated and user will get the thank

message for taking DND survive.

### 4.2.2. Data Conversion

Not Applicable

## Archival and retention requirements

Not Applicable

## Disaster and Failure Recovery

Not Applicable

## Business Process workflow

Not Applicable

## Business Process Modeling and Management (as applicable)

Not Applicable

## Business Logic

Not Applicable

## Variables

Not Applicable

## Activity / Class Diagrams (as applicable)

Not Applicable

## Data Migration

Not Applicable

### 4.10.1. Architectural Representation

Not Applicable

### 4.10.2.Architectural Goals and Constraints

Not Applicable

### 4.10.3. Logical View

### Not Applicable

### 4.10.4.Architecturally Significant Design Packages

Not Applicable

### 4.10.5.Data model

Not Applicable

### 4.10.6. Deployment View

Not Applicable

# Environment Description

Not Applicable

## Time Zone Support

Not Applicable

## Language Support

C Language is used in this project.

## User Desktop Requirements

Linux , C

## Server-Side Requirements

Not Applicable

### 5.4.1.Deployment Considerations

Not Applicable

### 5.4.2.Application Server Disk Space

Not Applicable

### 5.4.3.Database Server Disk Space

Not Applicable

### 5.4.4.Integration Requirements

Not Applicable

### 5.4.5.Jobs

Not Applicable

### 5.4.6.Network

Not Applicable

### 5.4.7 . Others

Not Applicable

## Configuration

For making this project we need windows version above 7 and Ubuntu Linux

### 5.5.1.Operating System

* Windows 7 and above
* Linux

### 5.5.2.Database

Not Applicable

### 5.5.3. Network

Not Applicable

### 5.5.4.Desktop

Linux & Windows 7

# References

* <https://codesteps.com/2014/06/13/c-programming-dealing-with-files-part-3/>
* <https://youtu.be/1lNL31q4T5I>
* <https://support.google.com/android/answer/9069335?hl=en>
* [https://www.mitel.com/features-benefits/do-not-disturb#:~:text=A%20do%20not](https://www.mitel.com/features-benefits/do-not-disturb#:~:text=A%20do%20not%20disturb%20feature,devices%20and%20chat%2Fcollaboration%20tools)

[%20disturb%20feature,devices%20and%20chat%2Fcollaboration%20tools](https://www.mitel.com/features-benefits/do-not-disturb#:~:text=A%20do%20not%20disturb%20feature,devices%20and%20chat%2Fcollaboration%20tools).

# Appendix

* C Functions
* Linux commands

**Change Log**

| **QMS Template Version Control (Maintained by QA)** | | | | | |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Date** | **Version** | **Author** | | **Description** | |
| 28-May-2015 | 1.0 | QA Team | | Initial Version | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |