**Abstract**

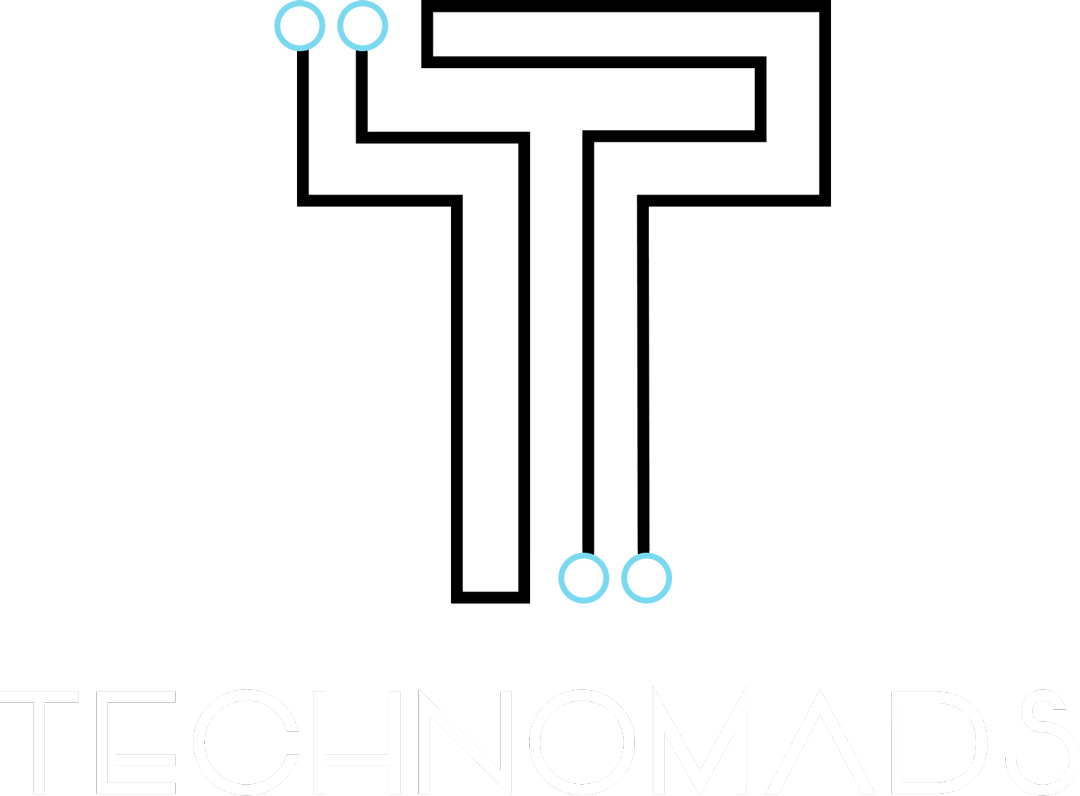
Our project provides a platform for intra society communication. It allows the users within a society to communicate. Users can chat, contact emergency services, file complaints, put up suggestions, vote on matters and search information. Users can also be a part of multiple societies at once. Data like suggestions, complains and voting results can be administered by the admin, while private data like chats are kept completely private to the users. Out project eliminates the need of multiple apps to manage and live in a society, providing a single platform for almost all tasks. Our project runs on multiple platforms so, the project can cover an even larger userbase.

**CHAPTER :- 1**

**Industry Introductions**

**1.1 Company Profile**

**Technomads Solution Pvt. Ltd.**



**Contact Information:-**

Phone:- [+91 63549 53278](tel:+91%2063549%2053278)

E-Mail:- [info@technomads.in](mailto:info@technomads.in)

Address:- 4080, Silver Business Point VIP Circle Utran Surat, Gujarat 394105

**What the Company Does?**

The company helps organizations streamline their efforts by providing customer-specific technical solutions. Their proven methodologies have helped their clients increase both internal efficiencies and revenue. They have helped businesses to build an online presence which opens the door to more sales and more growth of their business. They help their clients reach more customers with less money spend than the traditional marketing methods.

**What the Company has Achieved:**

Two 24 years old young entrepreneurs started their journey to build a great company after quitting their 9 to 5 job. TechNomads started their operations in July, 2019 in Surat and in a shortest span of time they have gained trust of local clients as well as international clients from Australia. They are building products in the healthcare industry and they believe the greatest products or solutions solve the simplest problems.

**Services provided by the Company:**

The company provides various services. Their work includes but is not limited to projects built using Blockchain, NodeJS, ReactJS, Flutter, and they also make Android App, iOS App, and apps for other platforms.

**Their Social media presence:**

Website:- https://technomads.in

LinkedIn:- https://www.linkedin.com/company/tech-nomads/mycompany

Twitter:- https://twitter.com/technomadss

Facebook:- https://www.facebook.com/technomadss

**CHAPTER: - 2**

**Introduction of project**

**2.1 Introduction of Problem**

**2.1.1 Problem statement**

* Maintenance fees. Payments-related concerns include mode of payment, timely submission of fees, etc. So we provide the perfect solution for all the above matters for instance payments mode includes online payment, UPI along with cash notes. Also, our system will provide a timely reminder to everyone for submission of maintenance fees through text message.
* Voting for various events is important in urban areas. In our system, we have included a voting system for various events.
* In residential areas, there is a lack of communication between people because of a daily busy life. The solution is for society members can communicate with each other through the chats module.
* Society members can complain about day-to-day issues in residential areas. As well as members can suggest new ideas to improve residential area facilities.
* Members can see the notice of various events, meetings, etc. which can be added by the admin.

**2.1.2 Scope of proposed system:**

* Our project is a smart and technology-forward security solution that helps you manage gated community security needs like entry and exit of visitors, household help, and online delivery personnel. It also helps in managing gated community needs like clubhouse and amenities management, complaint management, accounting and billing, and more.
* So to solve this problem there are many Security management services that are recognized and ensure the protection of resources and practices of an organization, person, or any group.
* Occupants can utilize the our application to speak with neighbours, examine matters of the local area and society on inward conversation gatherings, manage guests (visitors, conveyances, taxis, and so on), maintain participation record, and compensation installments for day by day help, find services, take care of society maintenance bills, among much else.
* In our project user can add location based emergency alerts.
* People who cannot help will not be bothered.
* On-Device database for Offline access.
* Paid plan for larger societies.

**2.2 Environment Description**

**2.2.1 Hardware and Software Requirements:**

**Minimum**

**Windows**

* Hardware
  + Processor: Dual code, 1GHz
  + RAM: 4GB
  + Storage: 2GB free
  + Display: 720p or higher
* Software
  + Windows 7 or higher

**macOS**

* Hardware
  + Processor: Dual code, 1.8GHz
  + RAM: 4GB
  + Storage: 2GB free
  + Display: 720p or higher
* Software
  + macOS El Capitan (macOS 10.11) or higher

**Linux**

* Hardware
  + Processor: Dual code, 1GHz
  + RAM: 4GB
  + Storage: 2GB free
  + Display: 720p or higher
* Software
  + Debian based Linux distribution

**Android**

* Hardware
  + Processor: 1GHz
  + RAM: 1GB
  + Storage: 2GB free
  + Display: 720p or higher
* Software
  + Android Jelly Bean (4.1) or higher

**iOS**

* Hardware
  + Storage: 2GB free
* Software
  + iOS 9 or higher

**Recommended**

**Windows**

* Hardware
  + Processor: Dual code, 2.3GHz
  + RAM: 8GB
  + Storage: 2GB free
  + Display: 1080p or higher
* Software
  + Windows 10 or higher

**macOS**

* Hardware
  + Processor: Dual code, 1.8GHz
  + RAM: 8GB
  + Storage: 2GB free
  + Display: 1080p or higher
* Software
  + macOS Monterey (macOS 12) or higher

**Linux**

* Hardware
  + Processor: Dual code
  + RAM: 8GB
  + Storage: 2GB free
  + Display: 1080p or higher
* Software
  + Debian based Linux distribution

**Android**

* Hardware
  + Processor: modern multicore processor
  + RAM: 4GB
  + Storage: 2GB free
  + Display: 1080p or higher
* Software
  + Android KitKat (4.4) or higher

**iOS**

* Hardware
  + Processor: Apple A9 or higher
  + RAM: 2GB
  + Storage: 2GB free
* Software
  + iOS 14 or higher

**2.1.2 Technology Used**

**Flutter:**

****

* Flutter is a framework developed by Google that allows developers to build native cross-platform apps with just one codebase.
* Traditionally, a developer would have to build for all their platforms separately but, using flutter, as discussed above, the developer can have just a single codebase to worry about.
* This also helps with other important factors such as improving data consistency and reducing data redundancy.
* Besides its compilation tools, Flutter acts as a framework by providing a collection of UI building blocks (widgets) such as tabs, dropdowns, buttons, and many more.

**Dart:**

****

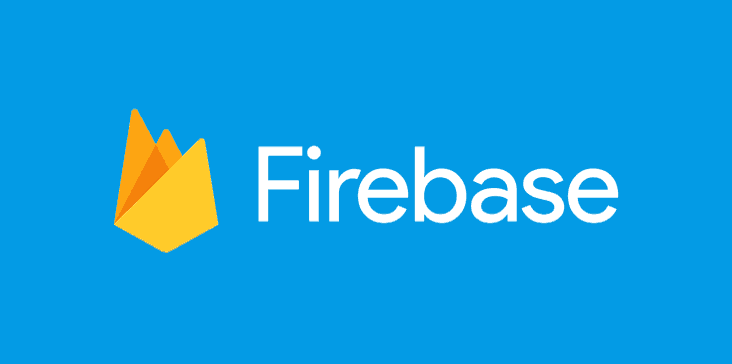
* Dart is the open-source programming language developed by Google. It is meant for both server side as well as the user side development.
* Dart is **Object-oriented language** and is quite like that of [**Java Programming**](https://www.geeksforgeeks.org/java/).
* Dart is extensively use to create single-page websites and web-applications. Best **example** of dart application is **Gmail**.
* Dart works as a base language for the framework we used which is Flutter

**GitHub:**

****

* Any team of developers working on a single project need a way to control the source code between then and a platform to store all the code while in development. This service to us is provided by GitHub
* GitHub is a web-based interface that uses Git, the open-source version control software that lets multiple people make separate changes to web pages at the same time.
* GitHub allows multiple developers to work on a single project at the same time, reduces the risk of duplicative or conflicting work.
* It also helps decrease production time.
* With GitHub, one can build code, track changes, and innovate solutions to problems that might arise during the development process.

**Firebase:**



* Firebase is a service developed by google that provides multiple facilities such as database, authentication, real-time databases among other services
* This service is primarily free to use but, only up to a paint since, after a certain amount of frequency, it starts to bill the developer.

**CHAPTER: - 3**

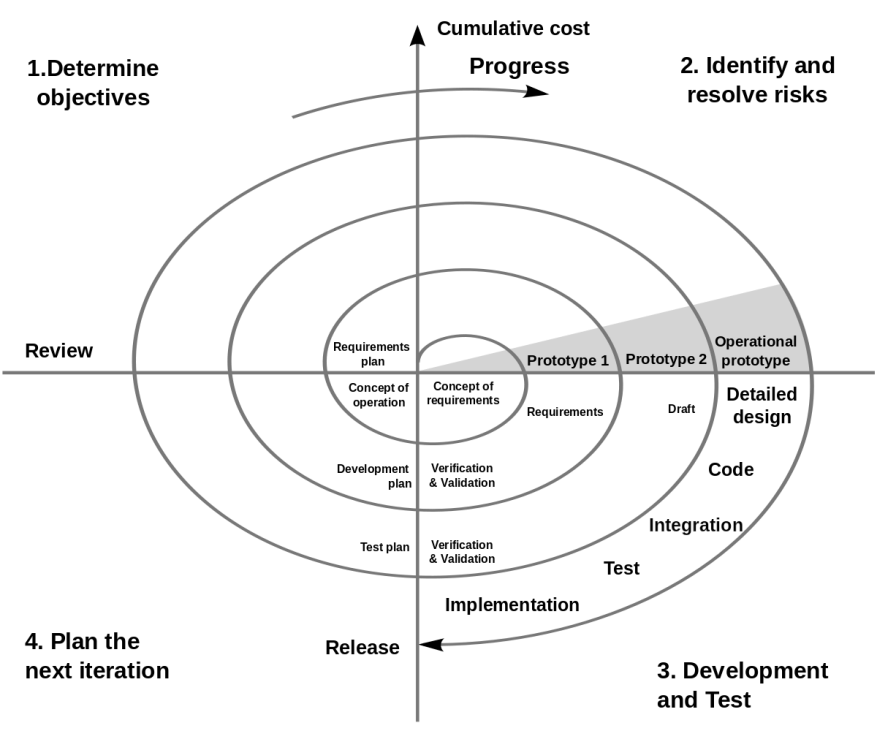
**The Whole Industrial Process And Problem Study**

**3.1 System Planning**

**3.1.1 Project Life Cycle Model**

# Spiral Model

Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using the spiral model.



The different phases of the spiral model are as follows:

* Determine Objectives
* Identify and Resolve risks
* Development and Test
* Plan the next Iteration

**Determine Objectives:**

* Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
* The objectives and alternatives of the project are determined and are documented.

**Identify and Resolve risks:**

* During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy.
* At the end of this quadrant, the Prototype is built for the best possible solution.

**Development and Test:**

* During the third quadrant, the identified features are developed and verified through testing.
* At the end of the third quadrant, the next version of the software is available.

**Plan the next Iteration:**

* In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.
* After evaluation if customer wants to add more feature in the software then the process of iteration is continued.

**Why we used spiral model?**

* For development of large scale / high risk projects.
* When cost and risk evolution is important.
* Users are unsure of their needs.
* Requirements are complex.

**3.1.3 Expected Modules**

1. **Welcome page:**

* A welcome page is **one or more web pages or modal overlays that appear the first time you open an app**.
* The welcome page is the first page the user comes across when they first install the application.

1. **Login page:**

* The login page authenticates the user.
* It can do so via multiple ways including the traditional id/password method or even other options like continue with Google, Apple Facebook, and many more for added convenience.

1. **Home page:**

* The home page is the first page that is displayed after starting the app (provided that the user is signed in)**.**
* The user can navigate most other pages from the home page.

1. **Setting:**

* The user may change the setting relating to their account or the app from the settings page.
* The settings page shall also be used for personalization purposes.

1. **Emergency page:**

* The user can use the emergency page to contact or communicate to the first responders and/or emergency contacts of a person.
* The emergency page is helpful to the user in emergency problems relating to but not limited to medical help, electrical help, vehicle help, fire assistance.

1. **Profile page:**

* In this module, the user can edit and view their profile and personal details.
* Changes to the details can also be made from here.

1. **Complaint:**

* In this module, users can post their complaints related to society.
* Later, the admin staff of the society can access this information and act upon it.

1. **Search page:**

* From the search module, the user can search for any information they need to like flat number or vehicle number etc. of other members of the society.

1. **Voting page:**

* Knowing the people’s opinion is extremely important in a democratic society.
* However important, voting can be a chore.
* This module provides the users with the convenience of voting on issues from the devices.
* The admin can make polls and, let the users make their votes

1. **service page:**

* This module connects the user with different types of services like plumbers, electricians, mechanics etc.

1. **Suggestions:**

* Knowing what the people want in a society is important for the admins.
* Such data can help the admins improve the society and tailor it to their member's needs.
* This module is the digital version of a physical suggestion box.
* From this module, the user can send suggestions the admin of their respective societies.

**3.1.4 Feasibility Study**

**1. Technical feasibility:**

* Flutter allows us to build for multiple platforms at once.
* Due to this, we do not have to maintain separate codebases for each platform.
* Hence, we conclude that it is technically feasible to work on the project.

**2. Operational Feasibility:**

* The company (Technomads Solutions) is a software development company.
* The company works on project in Flutter among other languages and frameworks.
* The Company also works on such large-scale projects for benefit of the society.
* Hence, we conclude that the project is operationally feasible and fits well on the portfolio of the company.

**3. Economic Feasibility:**

* The project depends heavily on Firebase by Google.
* Firebase provides the project with amenities like authentication, database, and so on.
* Firebase has a free tier where; the developer can use it for free as long as they don’t pass the threshold.
* We would not be crossing the threshold.
* Hence, we conclude that it is economically feasible to work on the project**.**

**3.1.5 Limitation of Project**

Although the team has been hard at work on the project, some limitations in every piece of code ever written is inevitable and, improvement in such areas is always a possibility.

Here are some of the limitations from our project:

Our app is one that requires an active internet connection. Without this, most if not all functionalities of the app stop working. Some of the issues created when no internet is provided are…

1. It is not possible to authenticate the user
2. It is not possible to chat with on-another
3. It is not possible to search information on anyone
4. It is not possible to issue complains
5. It is not possible to vote
6. It is not possible to contact emergency services
7. It is not possible to modify one’s profile

All out limitations are relating to the lack of an active internet connection. This is because out service is reliant on Firebase for most of these tasks. Firebase being an online service, cannot be accessed when the app is offline.

In latter versions of out project, we shall add an offline database to work against these limitations in a limited capacity.

**3.6 Data Dictionary**

**Society Admin Staff Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | size | constraint0 | constraint1 | constraint2 | constraint3 |
| Society ID | Text string | 13 | Primary Key | Not Null | Unique | check Like("SID%") |
| Society Name | Text string | 600 |  | Not Null |  |  |
| Name | Text string | 300 |  |  |  |  |
| Country | Text string | 300 |  | Not Null |  | check In(…List of Countries...) |
| State | Text string | 300 |  | Not Null |  | check In(…List of States...) |
| City | Text string | 300 |  | Not Null |  |  |
| Address | Text string | 1800 |  | Not Null | Unique |  |
| Society SPOC | Text string | 600 |  | Not Null | Unique |  |
| Society E-Mail ID | Text string | 600 |  | Not Null | Unique | check Like("%@%.%") |
| Society Phone Number | Integer | 15 |  | Not Null | Unique |  |

**Society Member Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | size | constraint0 | constraint1 | constraint2 | constraint3 | constraint4 |
| ResidentID | Text String | 20 | Primary Key | Not Null | Unique | check Like("SaHi-B-103-01") |  |
| Name | Text String | 300 |  | Not Null |  |  |  |
| Phone Number | Integer | 15 |  |  |  |  |  |
| email | Text String | 600 |  | Not Null | Unique | check Like("%@%.%") |  |
| Society | Text String | 100 |  | Not Null |  | check In(…List of Societies...) |  |
| flat no | Integer | 5 |  | Not Null |  |  |  |
| building name | Text String | 100 |  | Not Null |  | check In(…List of Buildings...) |  |

**Society Design**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | size | constraint0 | constraint1 | constraint2 | constraint3 | constraint4 | constraint5 |
| Name of Structure | Text String | 100 | Primary Key | Not Null | Unique |  |  |  |
| Number of stories | Integer | 4 |  | Not Null |  |  |  |  |
| Homes per story | Integer | 2 |  |  |  | 999 | 10 | 9990 |
| Homes | Array(String) | 10000 |  | Not Null | Unique |  |  |  |

**Voting List Table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | size | constraint0 | constraint1 | constraint2 | constraint3 | constraint4 | constraint5 |
| PollID | Text String | 150 | Primary Key | Not Null | Unique | check Like(SocietyID + PollNumber) |  |  |
| SocietyID | Text String | 100 |  | Not Null |  | check In(…List of Socites...) |  |  |
| Title | Text String | 500 |  | Not Null | Unique |  |  |  |
| Fileds | Array(Text String) | 100 |  | Not Null | Unique |  |  |  |

**Voting Data Table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | size | constraint0 | constraint1 | constraint2 | constraint3 | constraint4 | constraint5 |
| PollID | Text String | 150 | Primary Key | Not Null | Unique | check Like(SocietyID + PollNumber) |  |  |
| Votes | Array(Integer) | 100 |  | Not Null |  |  |  |  |
| Voted users List | Array(Text String) | 100000 |  |  |  |  |  |  |

**Society List Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | size | constraint0 | constraint1 | constraint2 | constraint3 | constraint4 |
| Society ID | Text string | 13 | Primary Key | Not Null | Unique | check Like("SID%") |  |
| Society Name | Text string | 100 |  | Not Null |  |  |  |
| Country | Text string | 300 |  | Not Null |  | check In(…List of Countries...) |  |
| State | Text string | 300 |  | Not Null |  | check In(…List of States...) |  |
| City | Text string | 300 |  | Not Null |  |  |  |
| Address | Text string | 1000 |  | Not Null | Unique |  |  |
| Society SPOC | Text string | 600 |  | Not Null | Unique |  |  |
| Society E-Mail ID | Text string | 600 |  | Not Null | Unique | check Like("%@%.%") |  |
| Society Phone Number | Integer | 15 |  | Not Null | Unique |  | Foreign Key - AdminStaffTable |

**CHAPTER: - 6**

**Conclusion**

Our project is easy to use and, caters to the needs of each society. It provides a single platform for the society to function, eliminating the multiple services a society might need to function smoothly. With features like user-suggestions and user-complains, our project also helps the societies management comity to do their best by knowing what their members want. The project also provides the non-comity members of the society with additional features such as chat and emergency contact adding to the quality of life and safety of the society.

Society Admin Staff Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | size | constraint0 | constraint1 | constraint2 | constraint3 |
| Society ID | Text string | 13 | Primary Key | Not Null | Unique | check Like("SID%") |
| Society Name | Text string | 600 |  | Not Null |  |  |
| Name | Text string | 300 |  |  |  |  |
| Country | Text string | 300 |  | Not Null |  | check In(…List of Countries...) |
| State | Text string | 300 |  | Not Null |  | check In(…List of States...) |
| City | Text string | 300 |  | Not Null |  |  |
| Address | Text string | 1800 |  | Not Null | Unique |  |
| Society SPOC | Text string | 600 |  | Not Null | Unique |  |
| Society E-Mail ID | Text string | 600 |  | Not Null | Unique | check Like("%@%.%") |
| Society Phone Number | Integer | 15 |  | Not Null | Unique |  |