

# 04 Ideas

---

- Github - PDF (<https://github.com/FrankBevr/100-Ideas/blob/main/practice/04/04-Ideas.pdf>).
- Github - Code (<https://github.com/FrankBevr/100-Ideas/blob/main/practice/04/diabetes/lib.rs>).

# Overview

---

- 04 Ideas
  - Overview
  - Intro
  - Think
    - 100 Words
    - 10 Sentences
    - Rate & Select
  - Sketch
    - Idea
    - Reuquirements
    - Stories
    - Diagrams
  - Paint
  - Outro

# Intro

---

I'm Frank

## 100 Words

Words				
Kindergarten	Box	Toys	Plüsstier	Garden
Dirt	Football	Language	Car	Computer
School	Homework	Reading	Math	Mandala
Teacher	Grades	Horses	Dog	Cat
Middle Shool	Beer	Class Leader	Inline Skating	Facebook
Parties	Organisation	Rock	Festivals	High School
Coffee Automat	Chemistry	Biotechnology	Food	Mopped
Carneval	Bus	Requirements	River	Apprenticeship
Shoes	Insoles	Insurance	Crafting	Leather
PVC	Ziehen	Drilling	Hammer	Tools
University	Renting	Professors	Exams	Furtunity
House	Clubs	Traveling	Trade Fairs	Developing
Unity	Hololens	Blender	Jira	Davinci
Christmas Party	5G	Kids	Museums	City
Construction	Web	Udemy	Figma	Zero To Mastery
Motivation	Fiverr	Emails	Linux	Frameworks
CSS	Api	ThreeJS	WebGL	gIsI
Blockchain	Solidity	Ethereum	Polkadot	Moonbeam
ink	rust	daos	amsterdam	slovenia
open source	minting	swapping	nfts	

## 10 Sentences

Word	Name	Phrase	Rating
Biotechnology	Diabetes	Reward your effort	8
Plüshtier	Plusho	Your Ai Assients	6
amsterdam	Infm	Share, collect, summarize	5
Mandala	Geni	Digital Assets Generation	4
Parties	Puto	Your web3 party space	3
Furtunity	DaoHous0	Your Dao, Your House	2
glsI	Shadero	Generative Art	2
Emails	Maila	Enhance your communication	1
glsI	Shadero	Generative Art	2

## Rate & Select

**Selected:** Diabetes - Reward your effort



# Sketch

---

## Idea

Problem: People take less care about themselves. Public pays for it.

Idea: diabetes.xyz

Solution: Create a reward mechanism which incentivizes Diabetes people.

## Requirements

Functional Requirements:

- The App must allow to post their achievements
- The App must allow to display a leaderboard
- The Smart Contract must pay token
- The Smart Contract must post achievements
- The Smart Contract must be able to validate achievements

Non Functional Requirements:

- The Smart Contract must be able to the public
- The Smart Contract must be legally confirm

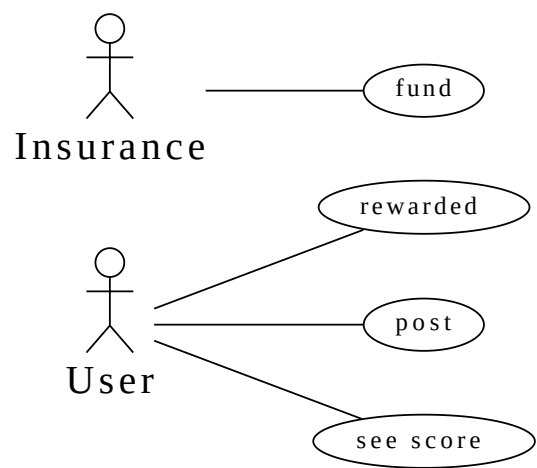
## Stories

User Story

- As a User I want to get rewarded for my diabetes treatment
- As a User I want to post my achievements
- As a User I want to see my score Insurance Story
- As a Insurance Company I want to be able to invest

# Diagrams

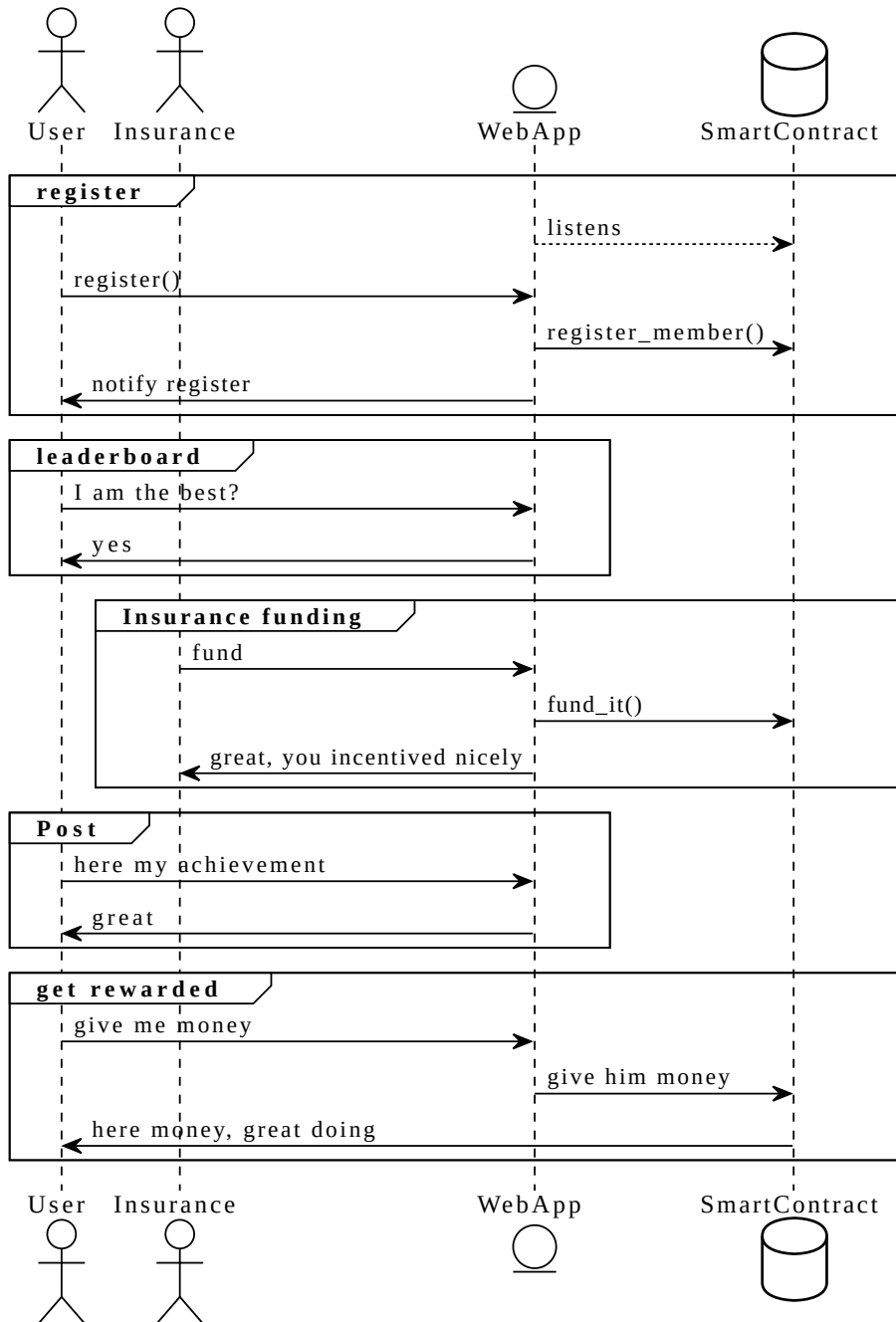
## Use Case



## Class Diagram

Ⓒ Diabetes
◦members: Member[] ◦achievements: [MemberId, Achievement] ◦leader: Member
◦rewarded(_member: Member) ◦post(_achievement: Achievement) ◦get_leader(): Member ◦fund()

# Sequence Diagram





```
#![cfg_attr(not(feature = "std"), no_std, no_main)]

#[ink::contract]
mod diabetes {
    use ink::prelude::string::String;

    #[ink(storage)]
    pub struct Diabetes {
        value: bool,
        member: AccountId,
        leader: AccountId,
        achievements: String,
    }

    impl Diabetes {
        #[ink(constructor)]
        pub fn new() -> Self {
            Self {
                value: true,
                member: AccountId::from([0xff;32]),
                leader: AccountId::from([0xff;32]),
                achievements: String::from("")
            }
        }

        #[ink(message)]
        pub fn flip(&mut self) {
            self.value = !self.value;
        }

        #[ink(message)]
        pub fn get(&self) -> bool {
            self.value
        }

        #[ink(message)]
        pub fn rewarded(&self) {}

        #[ink(message)]
        pub fn post_achievements(&mut self) {
            self.achievements = String::from("Great Certifacte from the Fitness Studi
        }

        #[ink(message)]
        pub fn get_achievements(&self) -> String {
            self.achievements.clone()
        }

        #[ink(message)]
```

```
    pub fn get_leader(&self) -> AccountId {
        self.leader
    }

    #[ink(message)]
    pub fn fund(&self) {}
}
}
```

## Outro

---

- Communication : Think, Sketch, Paint
- Adaptive: Ink, Pallets, Blockchain - It's Rust
- Potential: XXX Parachain, XX Pallet Contract, XX Dapps