Постановка задачи приложения

1. Проблема

На данный момент все ваши денежные средства затянуты во всемирную паутину и, к сожалению, с упрощением вашей работы с вашими финансами, приходит и большая опасность потерять их из-за кибер-мошенников. В текущее время существует огромное количество вредоносных программных обеспечений и методов для получения ваших честно заработанных денег какому-то воришке с интернета.

1. Сценарии

Сценарий 1

Артем, студент, живет простой студенческой жизнью. В один день с ним связывается неизвестный человек и говорит, что является сотрудником банка, после нескольких минут разговора сотрудник под видом профилактики или проверкой безопасности просит Артема произнести код, который пришел ему по СМС. Артем не почувствовав подвоха произносит ранее упомянутый код, и через некоторое время после этого телефонного разговора с баланса банковской карты Артема пропадают все деньги.

И в этой ситуации наша система может помочь вам уберечь ваши деньги. Если с вами связывается реальный работник службы безопасности банка он попросит вас пройти алгоритм верификации личности, это очень просто

Вы открываете ваш аутентификатор.

Работник зачитывает вам первый элемент кода.

Вы второй.

Он третий.

Вы четвертый.

Ну вы поняли.

И после ряда этих простых действий вы можете быть с точностью уверенны что с вами связался работник банка, а не какой-то воришка-психолог.

Сценарий 2

В большинстве случаев для входа в ваш личный кабинет необходимо ввести Логин и Пароль. И, по сути, все ваши деньги, которые вы заработали защищены паролем из комбинации букв и цифр длинной в 8-10 элементов.

И на данный момент существует огромное количество вредоносных программ чтобы расшифровать ваш код за несколько минут при помощью ключевых слов или вирусов, который записывают нажатые вами клавиши.

Вот решение для данной проблемы

После ввода логина и пароля на вашем экране появляется поле для ввода шестизначного пароля, а наше программное обеспечение генерирует случайный пароль, который может состоять из 6 цифр, латинских букв, или их комбинации.

Из них мы должны составить шестизначную комбинацию, чтобы узнать количество возможных комбинаций мы 36 возводим в 6 степень и получаем: Два миллиарда 176 миллионов 782 тысячи 336 возможных комбинаций. Что звучит весьма впечатляюще. И вот мы решили проблему с доступом к нашим банковским учетным записям.

1. Требования

В проекте должны быть учтены следующие функциональные требования (FR) и нефункциональные требования (NFR).

FR1: Приложение должно демонстрировать, по крайней мере, следующую функциональность: Генерация одноразовых кодов состоящий из определенного списка символов

NFR1: Удобство использования: Приложение должно быть интуитивно понятным в использовании, а пользовательский интерфейс должен быть простым для понимания. Все взаимодействия должны быть завершены менее чем за три клика.

NFR2: Соответствие руководящим принципам: Дизайн приложения должен соответствовать рекомендациям по удобству использования для выбранной операционной системы.

NFR3: Целевая платформа: Приложение должно быть разработано на Python.

Дополнительные ограничения:

* Система контроля версий должна быть git.
* Документация по исходному коду должна быть в формате HTML.

1. Целевая среда

Приложение должно быть продемонстрировано на Python

1. Документация

* Документ с анализом требований (RAD - Requirements Analysis Document)
* Проектный документ системы (SDD - System Design Document)
* Исходный код под контролем версий, включая документацию по исходному коду.