Техническая документация ПО Рыболовный оракул

Общее описание

1 База данных

2 Сбор данных

3 Анализ данных

Общее описание

Программный продукт предназначен для сбора, хранения и обработки информации о рыбалках с последующим анализом и прогнозированием планируемых рыбалок на основе собранной статистики.

Разработка делиться на несколько этапов.

1 этап. Подготовка документации

2 этап. Разработка каркаса для сбора информации и внесения её в БД

3 этап. Подготовка аналитической части

…

Для реализации проекта выбран фреймворк python 3+django 3

имя базы h9d1v8

пользователь h7dhr

пароль h6iHjLc4

1 этап.

На этапе подготовки документации разрабатывается концепция программного продукта, его структура и логика работы основных частей. Часть сбора и хранения информации и аналитической части.

Сбор и хранения информации

Для хранения информации используется база данных состоящая из следующих таблиц:

Блок результатов

1. Fish – Содержит в себе описание видов рыб, за основу принят справочник https://gdekluet.ru/directory/fish/

Таблица содержит поле name\_of\_fish тип поля CharField

class Fish(models.Model):

name\_of\_fish = models.CharField(max\_length=20)

def \_\_str\_\_(self):

return self.name\_of\_fish

2. fishing\_result — Содержит информацию о результате рыбалки, принимает в себя несколько записей о пойманой рыбе на одной рыбалке

содержит поля:

fish - выбор из таблицы Fish, поле должно содержать id записи в таблице Fish для дальнейшего подбора значения и фото. Тип поля PositiveIntegerField

number\_of\_fish — количество экземпляров. Тип поля PositiveIntegerField

fish\_weight — общий вес улова выбранной породы. Тип поля DecimalField

class Fishing\_Result(models.Model):

fish = models.ForeignKey(Fish, no\_delete=models.CASCADE)

number\_of\_fish = models.IntegerField(default=0)

fish\_weight = models.IntegerField(default=0)

3. Fish\_Trophy — Содержит информацию трофейных уловах. Содержит поля:

ID fish - выбор из таблицы Fish

fish\_trophy\_weight — вес трофея. Тип поля DecimalField

fish\_trophy\_photo — фото трофея. Тип поля ImageField

class Fish\_Trophy(models.Model):

fish = models.ForeignKey(Fish, no\_delete=models.CASCADE)

fish\_trophy\_weight = models.DecimalField(default=0)

fish\_trophy\_photo = models.ImageField()

Блок погоды

1 Weather\_Phenomena — Содержит информацию о погодных явлениях.

Допустимые значения:

- Туман

- Роса

- Дождь

- Ливень

- Гроза

- Снег

Содержит поля:

weather\_phenomena\_name — Содержит указанные выше погодные явления. Тип поля CharField

class Weather\_Phenomena(models.Model):

weather\_phenomena\_name = models.CharField(max\_length=10)

def \_\_str\_\_(self):

return self.weather\_phenomena\_name

2 Overcast – Содержит информацию о степени облачности. Допустимые значения:

- Ясно

- Малооблачно

- Переменная облачность

- Облачно

- Сплошные

Содержит поля:

overcast\_name — Содержит указанные выше степени облачности. Тип поля CharField

class Overcast(models.Model):

overcast\_name = models.CharField(max\_length=10)

def \_\_str\_\_(self):

return self.overcast\_name

3 Weather — Содержит сводные сведения о погоде опираясь на ручной ввод либо архив погоды по rp5.ru в выбранном населенном пункте.

Содержит поля:

ID Overcast — Выбор из таблицы Overcast

weather\_locality — Указывается название ближайшего населенного пункта, возможно из кладр. Тип поля CharField

ID weather\_phenomena — Выбор погодных явлений из таблицы Weather\_Phenomena, необходимо предусмотреть возможность выбора нескольких вариантов. Тип поля ForeignKey

weather\_temperature — Температура воздуха, ручной ввод либо выбор из архива погоды rp5.ru. Тип поля IntegerField

pressure – Атмосферное давление, ручной ввод либо выбор из архива погоды rp5.ru. Тип поля PositiveIntegerField

direction\_wind — Направление ветра, ручной ввод либо выбор из архива погоды rp5.ru.

Принимает значения

- СЗ - север-запад

- С - север

- СВ - север-восток

- В - восток

- ЮВ - юго-восток

- Ю - юг

- ЮЗ - юго-запад

- З - запад

Тип поля Заполнение из списка. Тип поля CharField

wind\_speed — Скорость ветра, ручной ввод либо выбор из архива погоды rp5.ru. Тип поля DecimalField.

lunar\_day — День по лунному календарю. Ручной ввод (возможен расчет). Тип поля PositiveIntegerField

class Weather(models.Model):

overcast\_name = models.CharField(max\_length=10)

def \_\_str\_\_(self):

return self.overcast\_name

Взаимосвязь с таблицами

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Название таблицы  (name=models.ForeignKey(название таблицы, ...) | Способ связи | Таблица Связи | Примечание |
| 1 | Fishing\_Result | ForeignKey | Fish | В таблице Fishing\_Result в разных рыбалках используются одни и те же записи из таблицы Fish |
| 2 | Fish\_Trophy | ForeignKey | Fish | В таблице Fish\_Trophy в разных рыбалках используются одни и те же записи из таблицы Fish |
| 3 | Weather | ForeignKey | Overcast | В таблице Weather в разных рыбалках используются одни и те же записи из таблицы Overcast |
| 4 | Weather | ForeignKey | Weather\_Phenomena | В таблице Weather в разных рыбалках используются одни и те же записи из таблицы Weather\_Phenomena |
| 5 | Water |  | District | В таблице Water  записи привязываются к только к одной из записей таблицы District |
| 6 | Place | ForreignKey | Water | Несколько записей талбицы Place может быть привязвно к 1 водоему |
| 7 | Bottom\_Map |  | Place | Запись в таблице Bottom\_Map привязана к одной записе в таблице Place |
| 8 | Point |  | Bottom\_Map | Несколько записей таблицы Point привязаны к одной записи таблицы Bottom\_Map |
| 9 | Point | ForeignKey | Priming | Несколько записей таблицы Point могут быть привязаны к одной записи таблицы Priming |
| 10 | Fishing\_Point |  | Place | Запись таблицы Fishing\_Point может быть привязана только к одной записи таблицы Place |
| 11 | Fishing |  | Place | Запись таблицы Fishing привязана к к записи в таблице Place, но записи таблицы Place могут использоваться в разных записях таблицы Fishing |
| 12 | Fishing |  | Weather | Запись в таблице Fishing привязана к записи в таблице Weather |
| 13 |  |  |  |  |