



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

О Т Ч Е Т
по лабораторной работе № 2

Название: Создание прототипа ИС на базе MongoDB

Дисциплина: Распределенные базы данных

Студент

ИУ6-14М
(Группа)

(Подпись, дата)

А.И. Дегтярев
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

М.М. Фомин
(И.О. Фамилия)

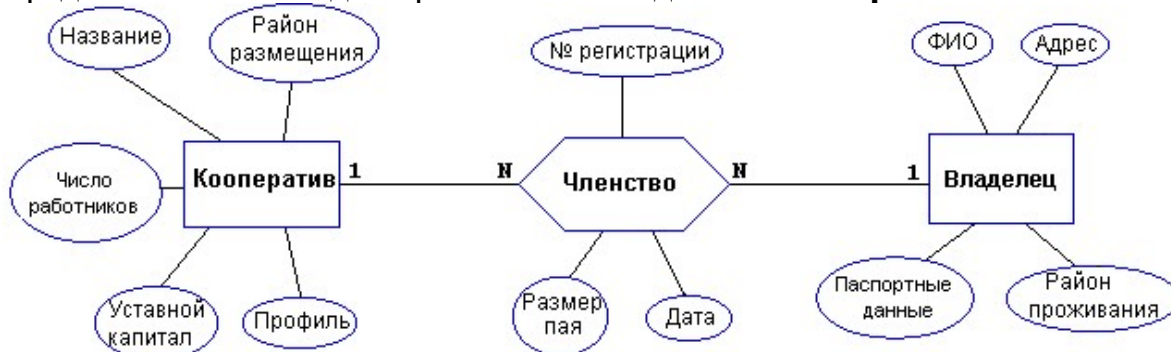
Москва, 2023

Задание

В ходе лабораторной работы требуется разработать:

- схемы бизнес-процессов в предметной области;
- схему взаимодействия сущностей;
- тексты скриптов для создания и заполнения коллекций;
- тексты скриптов для поиска и выдачи данных по базовым сущностям;
- скриншоты интерфейсов и заполненных коллекций.

Предметная область для практических заданий: **Кооператив**



Цель работы: научиться создавать БД в MongoDB.

1 Схемы бизнес-процессов в предметной области

В данной лабораторной работе были рассмотрены бизнес-процессы регистрации кооператива в ИС и передачи пая.

Регистрацию кооператива предлагается делать на основе документа решения о создании кооператива, после подтверждения факта внесения регистрации в ЕГРЮЛ, так как в нем представлена информация как о составе учредителей, так и юридическая информация о кооперативе. На рисунке 1.1 представлена верхнеуровневая декомпозиция регистрации кооператива в ИС в виде IDEF0 диаграммы.

По ГК РФ владельцы могут передать свой пай или его часть другим владельцам кооператива если иное не установлено уставом. Передача пая гражданину, не являющемуся членом кооператива допускается лишь с согласия кооператива. На рисунке 1.2 представлена верхнеуровневая декомпозиция передачи пая в виде IDEF0 диаграммы.

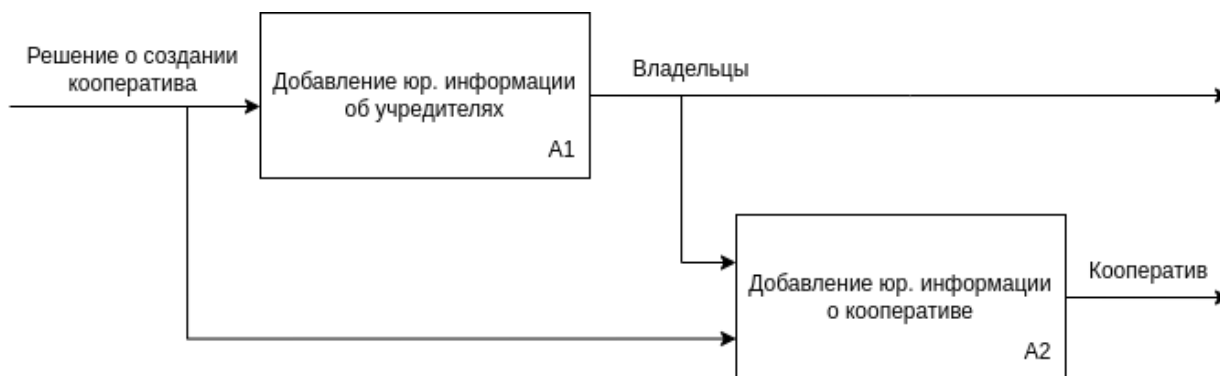


Рисунок 1.1 – Декомпозиция процесса регистрации кооператива в ИС

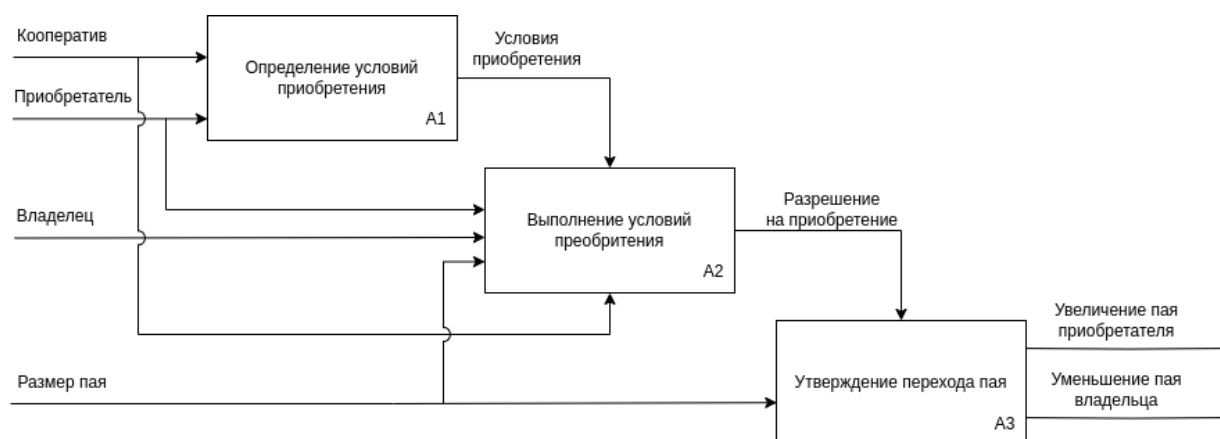


Рисунок 1.2 – Декомпозиция процесса передачи пая

2 Схема взаимодействия сущностей

Для реализации рассмотренных процессов следует выделить следующие сущности: Владелец, Кооператив и Членство. Схема их взаимодействия соответствует представленной в задании.

Для хранения информации об этих сущностях были созданы 3 коллекции, каждая соответствует своей сущности. Документы коллекции Членства связывают документы Кооператив и Владелец.

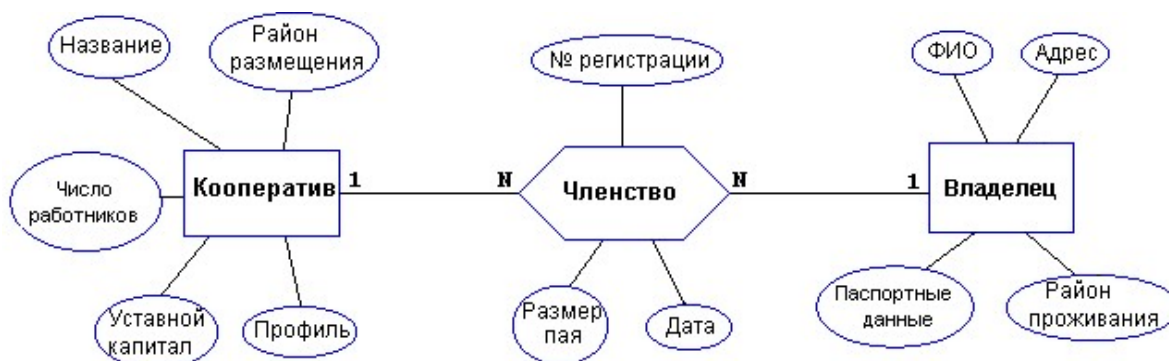


Рисунок 2.1 – Схема взаимодействия сущностей

3 Скрипты для создания и заполнения коллекций

На листинге 3.1 представлен скрипт заполнения коллекций тестовыми данными в соответствии с таблицей 3.1. В коллекции Членство было создано 2 индекса для полей с внешними ключами для увеличения скорости запросов по поиску владельцев кооператива.

Таблица 3.1 – Размеры паев владельцев кооперативов в тестовых данных

	Иванов	Петухов	Кузнецов
Мир овощей	101	–	–
Мир пуговиц	102	202	–
Мир ручек	103	203	303

Листинг 3.1 – Заполнение коллекций тестовыми данными

```
from settings import DATABASE_URI
from pymongo import MongoClient

db = MongoClient(host=DATABASE_URI).get_default_database()

o1, o2, o3 = db.owners.insert_many([
    {
        "name": "Иванов Иван Иванович",
        "address": "Январская улица, д 1",
        "residence": "Россия",
        "passport": "1111222233334444",
    }, {
        "name": "Петухов Петр Петрович",
        "address": "Январская улица, д 1",
        "residence": "Россия",
        "passport": "1111222233334444",
    }, {
        "name": "Кузнецов Кирилл Кириллович",
        "address": "Январская улица, д 1",
        "residence": "Россия",
        "passport": "1111222233334444",
    }
]).inserted_ids

c1, c2, c3 = db.cooperatives.insert_many([
    {
        "name": "Мир овощей",
        "residence": "Россия",
        "workers_number": 11,
        "capital_amount": 101,
```

```

        "profile": "Продукты",
    }, {
        "name": "Мир пуговиц",
        "residence": "Россия",
        "workers_number": 22,
        "capital_amount": 304,
        "profile": "Галантерея",
    }, {
        "name": "Мир ручек",
        "residence": "Россия",
        "workers_number": 33,
        "capital_amount": 609,
        "profile": "Канцелярия",
    }]).inserted_ids

today = "2024-01-01"

db.memberships.insert_many([
    {
        "number": 1,
        "owner_id": o1, "cooperative_id": c1,
        "amount": 101,
        "date": today,
    }, {
        "number": 2,
        "owner_id": o1, "cooperative_id": c2,
        "amount": 102,
        "date": today,
    }, {
        "number": 3,
        "owner_id": o2, "cooperative_id": c2,
        "amount": 202,
        "date": today,
    }, {
        "number": 4,
        "owner_id": o1, "cooperative_id": c3,
        "amount": 103,
        "date": today,
    }, {
        "number": 5,
        "owner_id": o2, "cooperative_id": c3,
        "amount": 203,
        "date": today,
    }, {
        "number": 6,
        "owner_id": o3, "cooperative_id": c3,
        "amount": 303,
        "date": today,
    }
])

```

```
db.memberships.create_index("owner_id")
db.memberships.create_index("cooperative_id")
```

4 Скрипты для поиска и выдачи данных по базовым сущностям

Лабораторная работа была выполнена на языке Python 3.9.18 в виде простого веб сервера на [Flask](#) и библиотеки валидации веб форм [wtforms](#). Работа с MongoDB происходит с помощью пакета pymongo.

На листинге 4.1 представлен скрипт создания Flask приложения с обработчиками следующих путей:

- / — выводит список созданных кооперативов в виде таблицы;
- /cooperatives/register — форма регистрации кооператива позволяет выполнить соответствующий бизнес-процесс;
- /cooperatives/<cooperative_id>/memberships — выводит список членств определенного кооператива;
- /memberships/<membership_id>/transfer — форма передачи пая позволяет выполнить соответствующий бизнес-процесс;
- /cooperatives/search — поиск кооператива по имени;
- /owners — выводит список зарегистрированных в ИС владельцев в виде таблицы;
- /owners/create — форма добавления нового владельца;
- /owners/search — поиск владельца по ФИО.

На листинге 4.2 представлено описание форм, которые отвечают за валидацию данных в теле POST-запроса и сохранение этих данных в базу:

- CreateOwnerForm — создание нового владельца;
- RegisterCooperativeForm — регистрация нового кооператива, форма создает документы в коллекции кооперативов и членства. Это происходит в одной форме, так как кооператив не может быть без членов;

- TransferMembershipForm — передача пая, форма создает 2 документа в коллекции членства: с уменьшенным размером пая владельца и новым размером пая получателя.

Листинг 4.1 – Обработчики Flask приложения, app.py

```
from bson import ObjectId
from flask import Flask, abort, redirect, render_template, request
from flask_wtf.csrf import CSRFProtect

from db import db
from forms import CreateOwnerForm, RegisterCooperativeForm, \
    TransferMembershipForm

app = Flask(__name__)
app.config.from_pyfile("settings.py")
csrf = CSRFProtect(app)

@app.route("/", methods=["GET"])
def index():
    cooperatives = list(db.cooperatives.find(limit=25, sort=[('name', 1)]))
    return render_template("pages/index.html", cooperatives=cooperatives)

@app.route("/cooperatives/register", methods=["GET", "POST"])
def register_cooperative():
    form = RegisterCooperativeForm()
    if form.validate_on_submit():
        form.save()
        return redirect('/', code=302)

    return render_template("pages/cooperatives/register.html", form=form)

@app.route("/cooperatives/<cooperative_id>/memberships", methods=["GET"])
def view_cooperative_memberships(cooperative_id):
    try:
        cooperative_id = ObjectId(cooperative_id)
    except:
        return abort(400)

    cooperative = db.cooperatives.find_one({"_id": cooperative_id})
    if not cooperative:
        return abort(404)
```

```

all_memberships = list(db.memberships.find({
    "cooperative_id": cooperative_id
}, sort=[("number", -1)]))

owners_set = {m.get("owner_id") for m in all_memberships}
owners_filter = {"_id": {"$in": list(owners_set)}}
search = request.args.get("q", "")
if search:
    owners_filter["name"] = {
        "$regex": search,
        "$options": "i",
    }

owners = {o["_id"]: o for o in db.owners.find(owners_filter)}
showed_owners = set()

memberships = []
for member in all_memberships:
    owner_id = member.get('owner_id')
    member["can_transfer"] = (
        owner_id not in showed_owners and
        member.get("amount", 0) > 0
    )
    member["owner"] = owners.get(owner_id)
    showed_owners.add(owner_id)
    if member["owner"]:
        memberships.append(member)

return render_template(
    "pages/cooperatives/memberships.html",
    cooperative=cooperative,
    memberships=memberships,
    search=search
)

@app.route("/memberships/<membership_id>/transfer", methods=["GET",
"POST"])
def transfer_membership(membership_id):
    try:
        membership_id = ObjectId(membership_id)
    except:
        return abort(400)

    membership = db.memberships.find_one({"_id": membership_id})
    if not membership:
        return abort(404)

```



```

        membership["owner"] = db.owners.find_one({"_id":
membership.get("owner_id")})
        membership["cooperative"] = db.cooperatives.find_one({"_id":
membership.get("cooperative_id")})

        form = TransferMembershipForm(membership)
        if form.validate_on_submit():
            form.save()
            return
    redirect(f"/cooperatives/{membership['cooperative_id']}/memberships",
code=302)

    return render_template("pages/memberships/transfer.html", form=form)

@app.route("/cooperatives/search", methods=["GET"])
def search_cooperative():
    filter = {}
    search = request.args.get("q", "")
    if search:
        filter["name"] = {
            "$regex": search,
            "$options": "i",
        }
    cooperatives = list(db.cooperatives.find(filter, limit=25,
sort=[('name', 1)]))

    return render_template("pages/cooperatives/search.html",
cooperatives=cooperatives, search=search)

@app.route("/owners", methods=["GET"])
def list_owners():
    owners = list(db.owners.find(limit=25, sort=[('name', 1)]))
    return render_template("pages/owners/index.html", owners=owners)

@app.route("/owners/create", methods=["GET", "POST"])
def create_owner():
    form = CreateOwnerForm()
    if form.validate_on_submit():
        form.save()
        return redirect("/owners", code=302)

    return render_template("pages/owners/create.html", form=form)

```

```

@app.route("/owners/search", methods=["GET"])
def search_owner():
    filter = {}
    search = request.args.get("q", "")
    if search:
        filter["name"] = {
            "$regex": search,
            "$options": "i",
        }
    owners = list(db.owners.find(filter, limit=25, sort=[('name', 1)]))

    return render_template("pages/owners/search.html", owners=owners,
search=search)

```

Листинг 4.1 – Обработчики Flask приложения, app.py

```

import datetime
from typing import Optional
from bson import ObjectId
from flask_wtf import FlaskForm
from wtforms import BooleanField, StringField, IntegerField, FormField, \
    FieldList, Form, DateField, ValidationError
from wtforms.widgets import HiddenInput
from wtforms.validators import DataRequired, NumberRange

from db import db
from widgets import CooperativeMembershipListWidget, OwnerForeignKeyWidget

def date2str(date: Optional[datetime.datetime]):
    if date:
        return date.strftime("%Y-%m-%d")

class CreateOwnerForm(FlaskForm):
    name = StringField('ФИО', validators=[DataRequired()])
    residence = StringField('Район проживания')
    address = StringField('Адрес')
    passport = StringField('Паспортные данные')

    def save(self):
        data = {
            name: f.data
            for name, f in self._fields.items()
            if f is not None and not isinstance(f.widget, HiddenInput)
        }

        return db.owners.insert_one(data)

```

```

class RegisterCooperativeForm(FlaskForm):
    class MembershipForm(Form):
        number = IntegerField('Номер регистрации',
validators=[DataRequired(), NumberRange(min=1)])
        owner_id = StringField('Владелец', validators=[DataRequired()],
widget=OwnerForeignKeyWidget())
        amount = IntegerField('Размер пая', validators=[DataRequired(),
NumberRange(min=1)])
        date = DateField('Дата', validators=[DataRequired()],
default=datetime.datetime.today)

        name = StringField('Название', validators=[DataRequired()])
        residence = StringField('Район размещения', default="Россия")
        workers_number = IntegerField('Количество работников (человек)',
validators=[NumberRange(min=1)], default=10)
        profile = StringField('Профиль', default="Разное")
        memberships = FieldList(FormField(MembershipForm), 'Членство',
min_entries=1, widget=CooperativeMembershipListWidget())

    def save(self):
        cooperative_id = db.cooperatives.insert_one({
            'name': self.name.data,
            'profile': self.profile.data,
            'workers_number': self.workers_number.data,
            'residence': self.residence.data,
            'capital_amount': sum(member['amount'] for member in
self.memberships.data),
        }).inserted_id

        db.memberships.insert_many([
            {
                'number': member['number'],
                'owner_id': ObjectId(member['owner_id']),
                'amount': member['amount'],
                'date': date2str(member['date']),
                'cooperative_id': cooperative_id,
            }
            for member in self.memberships.data
        ])

class TransferMembershipForm(FlaskForm):
    buyer_id = StringField('Получатель пая', validators=[DataRequired()],
widget=OwnerForeignKeyWidget())
    cooperative_approve = BooleanField('Согласие кооператива было
получено?')
    amount = IntegerField('Размер пая', validators=[DataRequired()],

```

```

NumberRange(min=1]])
    number = IntegerField('Номер регистрации', validators=[DataRequired(),
NumberRange(min=1]])
    date = DateField('Дата', validators=[DataRequired()],
default=datetime.datetime.today)

    def __init__(self, membership, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.membership = membership
        self.cooperative_id = self.membership.get("cooperative_id")
        self.buyer_membership = None

    def validate_number(self, field):
        number = field.data
        last_membership = db.memberships.find_one(
            {
                "cooperative_id": self.cooperative_id,
            },
            sort=[('number', -1)],
        )

        if last_membership and last_membership.get("number") >= number:
            raise ValidationError(f"Номер должен быть больше
{last_membership.get('number')}")

    def validate_buyer_id(self, field):
        buyer_id = ObjectId(field.data)
        if buyer_id == self.membership.get("owner_id"):
            raise ValidationError(f"Владелец и получатель должны
отличаться")

        self.buyer_membership = db.memberships.find_one(
            {
                "owner_id": buyer_id,
                "cooperative_id": self.cooperative_id,
            },
            sort=[('number', -1)],
        )
        if self.buyer_membership and self.buyer_membership.get("amount", 0)
<= 0:
            self.buyer_membership = None

    def validate_amount(self, field):
        owner_amount = self.membership["amount"]
        if field.data > owner_amount:
            raise ValidationError(f"Владелец не может передать пай, больше
чем {owner_amount}")

```

```

def validate(self, extra_validators=None) -> bool:
    valid = super().validate(extra_validators)

    if not self.buyer_membership and not self.cooperative_approve.data:
        self.cooperative_approve.errors.append(
            "Передача пая гражданину, не являющемуся членом
кооператива "
            "допускается лишь с согласия кооператива"
        )
        valid = False

    return valid

def save(self):
    buyer_amount = self.amount.data
    if self.buyer_membership:
        buyer_amount += self.buyer_membership.get('amount', 0)

    db.memberships.insert_many([
        {
            'number': self.number.data,
            'owner_id': ObjectId(self.buyer_id.data),
            'amount': buyer_amount,
            'date': date2str(self.date.data),
            'cooperative_id': self.cooperative_id,
        }, {
            'number': self.number.data,
            'owner_id': self.membership['owner_id'],
            'amount': self.membership['amount'] - self.amount.data,
            'date': date2str(self.date.data),
            'cooperative_id': self.cooperative_id,
        }
    ])

```

5 Скриншоты интерфейсов и заполненных коллекций

Интерфейс пользователя доступен по соответствующим путям в браузере. Для улучшения внешнего вида была использована библиотека [Simple.css](#).

На рисунках 5.1-5.3 представлены шаги создания нового владельца, на 5.4-5.8 – шаги регистрации кооператива, на 5.9-5.11 – шаги передачи пая владельца другому владельцу. На листинге 5.1 представлены заполненные коллекции после указанных шагов.

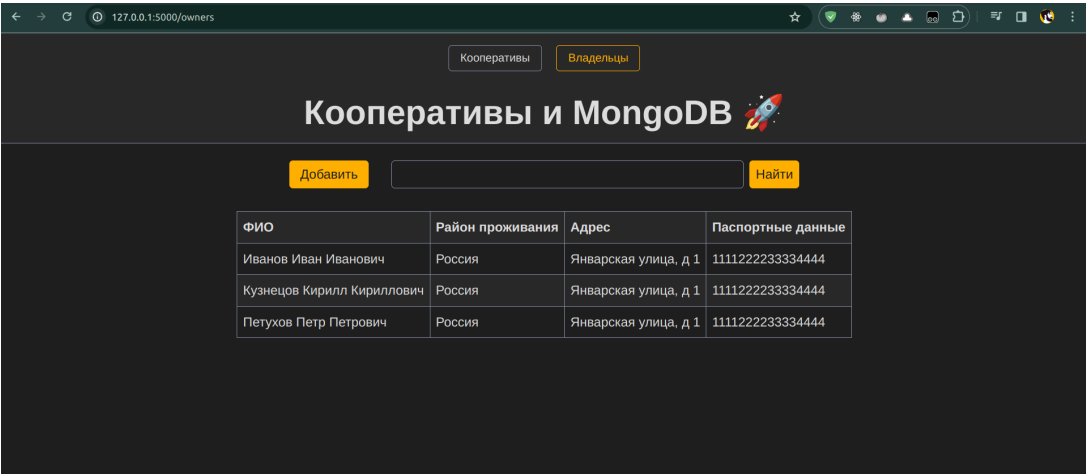


Рисунок 5.1 – Список владельцев с тестовыми данными

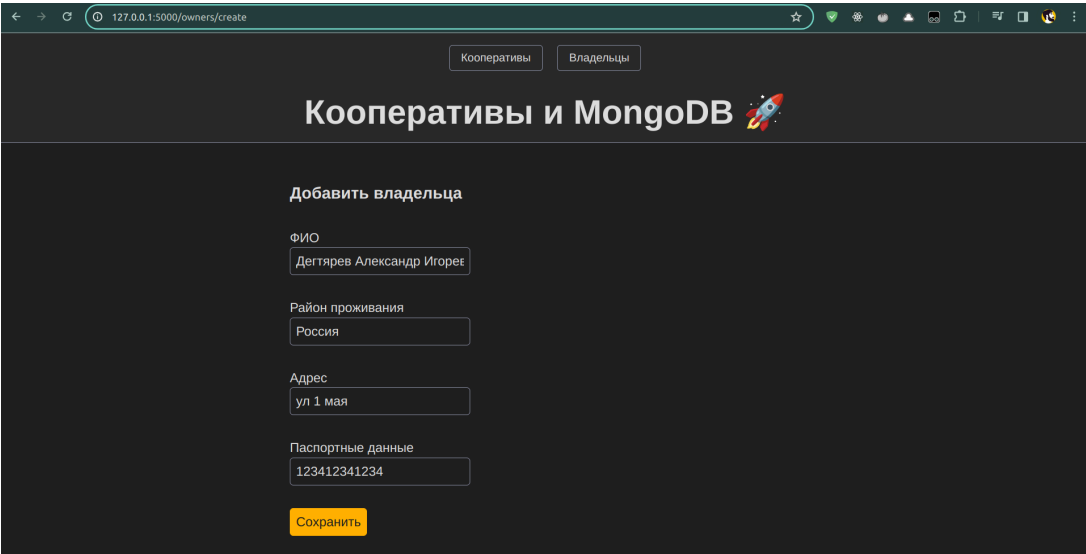


Рисунок 5.2 – Добавление владельца

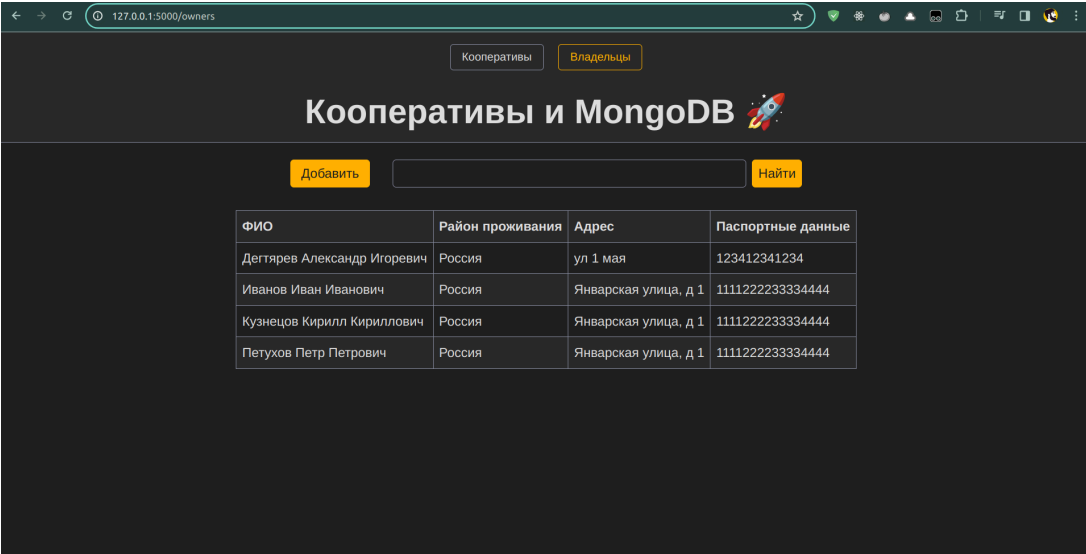


Рисунок 5.3 – Результат добавления

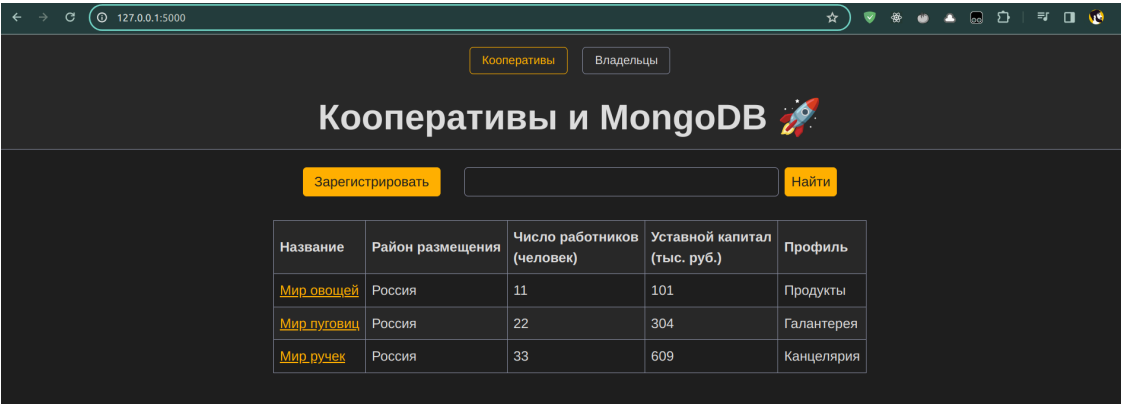


Рисунок 5.4 – Список кооперативов с тестовыми данными

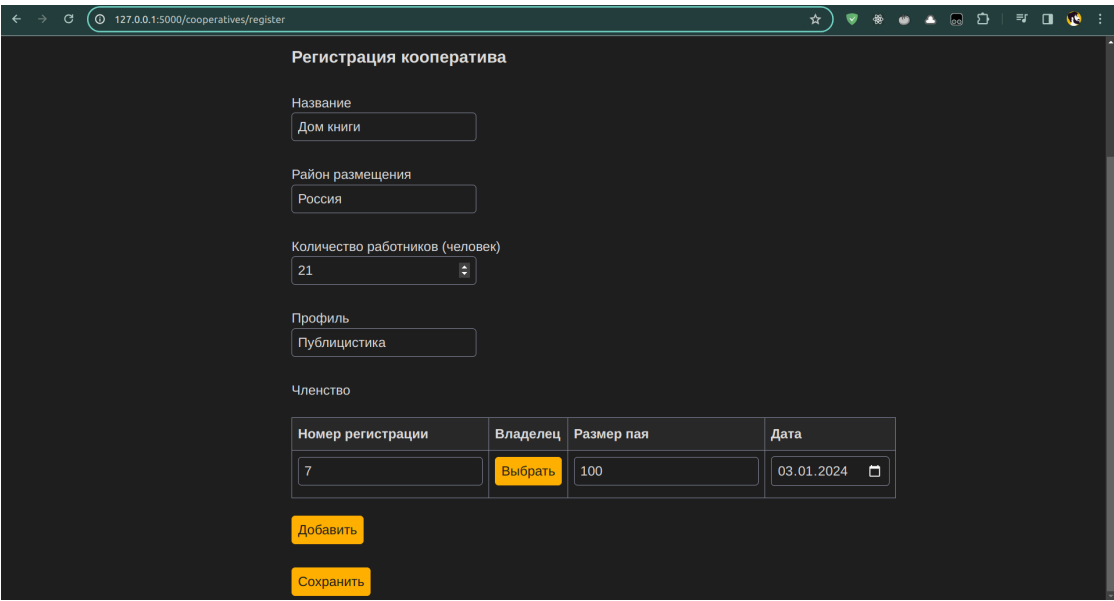


Рисунок 5.5 – Заполнение юр. информации о кооперативе

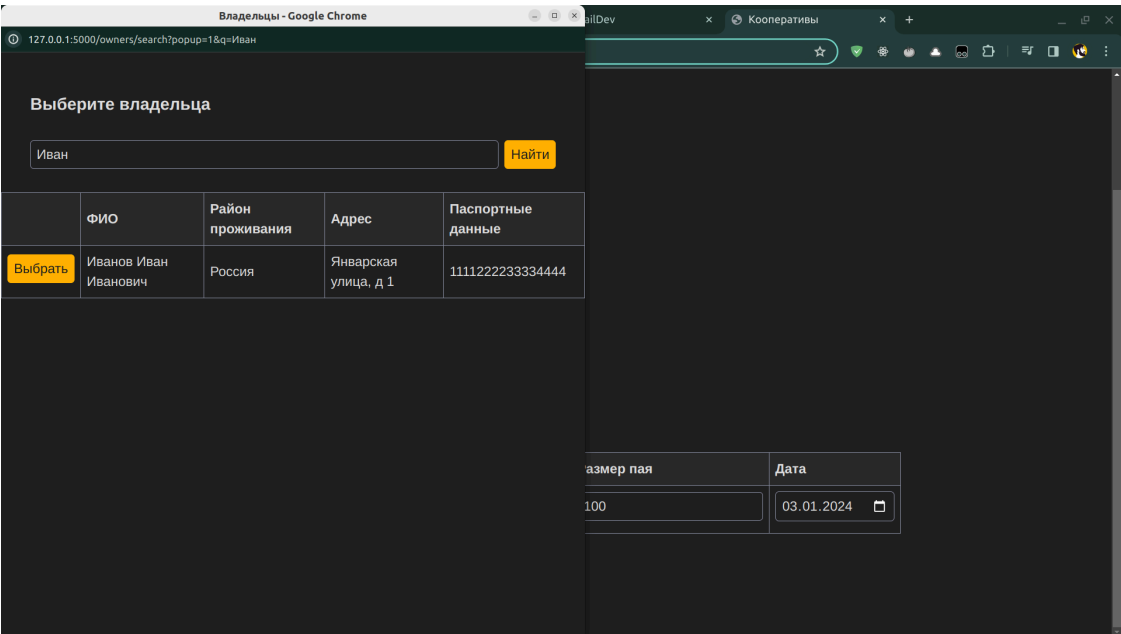


Рисунок 5.6 – Поиск и выбор владельца в ИС

127.0.0.1:5000/cooperatives/register

Количество работников (человек)
21

Профиль
Публицистика

Членство

Номер регистрации	Владелец	Размер пая	Дата
7	Иванов Иван Иванович Изменить	100	03.01.2024
8	Дегтярев Александр Игоревич Изменить	300	03.01.2024

Добавить

Сохранить

Рисунок 5.7 – Заполнение информации о членах кооператива

127.0.0.1:5000

Кооперативы Владелец

Кооперативы и MongoDB

Зарегистрировать

Найти

Название	Район размещения	Число работников (человек)	Уставной капитал (тыс. руб.)	Профиль
Дом книги	Россия	21	400	Публицистика
Мир овощей	Россия	11	101	Продукты
Мир пуговиц	Россия	22	304	Галантерея
Мир ручек	Россия	33	609	Канцелярия

Рисунок 5.8 – Результат регистрации кооператива

127.0.0.1:5000/cooperatives/65958997cf6b78c9659626d1/memberships

Кооперативы Владелец

Кооперативы и MongoDB

Владельцы кооператива "Дом книги"

Найти

Номер регистрации	ФИО владельца	Размер пая	Дата	
8	Дегтярев Александр Игоревич	300	2024-01-03	Передать
7	Иванов Иван Иванович	100	2024-01-03	Передать

Рисунок 5.9 – Список членов кооператива “Дом книги”

Передача пая

Кооператив
Дом книги

Владелец
Дегтярев Александр Игоревич

Получатель пая
Иванов Иван Иванович Изменить

Согласие кооператива было получено?
☐

Размер пая
50

Номер регистрации
9

Дата
03.01.2024

Сохранить

Рисунок 5.10 – Интерфейс передачи пая

Кооперативы Владельцы

Кооперативы и MongoDB

Владельцы кооператива "Дом книги"

Найти

Номер регистрации	ФИО владельца	Размер пая	Дата	
9	Иванов Иван Иванович	150	2024-01-03	Передать
9	Дегтярев Александр Игоревич	250	2024-01-03	Передать
8	Дегтярев Александр Игоревич	300	2024-01-03	
7	Иванов Иван Иванович	100	2024-01-03	

Рисунок 5.11 – Результат передачи части пая

Листинг 5.1 – Состояние коллекций после указанных шагов

```
lab02> db.cooperatives.find()
[
  {
    _id: ObjectId("659587844dd874d5dac5ddfe"),
    name: 'Мир овощей',
    residence: 'Россия',
    workers_number: 11,
    capital_amount: 101,
    profile: 'Продукты'
```

```

    },
    {
      _id: ObjectId("659587844dd874d5dac5ddff"),
      name: 'Мир пуговиц',
      residence: 'Россия',
      workers_number: 22,
      capital_amount: 304,
      profile: 'Галантерея'
    },
    {
      _id: ObjectId("659587844dd874d5dac5de00"),
      name: 'Мир ручек',
      residence: 'Россия',
      workers_number: 33,
      capital_amount: 609,
      profile: 'Канцелярия'
    },
    {
      _id: ObjectId("65958997cf6b78c9659626d1"),
      name: 'Дом книги',
      residence: 'Россия',
      workers_number: 21,
      profile: 'Публицистика',
      capital_amount: 400
    }
  ]
lab02> db.owners.find()
[
  {
    _id: ObjectId("659587844dd874d5dac5ddfb"),
    name: 'Иванов Иван Иванович',
    address: 'Январская улица, д 1',
    residence: 'Россия',
    passport: '1111222233334444'
  },
  {
    _id: ObjectId("659587844dd874d5dac5ddfc"),
    name: 'Петухов Петр Петрович',
    address: 'Январская улица, д 1',
    residence: 'Россия',
    passport: '1111222233334444'
  },
  {
    _id: ObjectId("659587844dd874d5dac5ddfd"),
    name: 'Кузнецов Кирилл Кириллович',
    address: 'Январская улица, д 1',
    residence: 'Россия',
    passport: '1111222233334444'
  }
]

```

```

    },
    {
      _id: ObjectId("6595882dcf6b78c9659626c8"),
      name: 'Дегтярев Александр Игоревич',
      residence: 'Россия',
      address: 'ул 1 мая',
      passport: '123412341234'
    }
  ]
lab02> db.memberships.find()
[
  {
    _id: ObjectId("659587844dd874d5dac5de01"),
    number: 1,
    owner_id: ObjectId("659587844dd874d5dac5ddfb"),
    cooperative_id: ObjectId("659587844dd874d5dac5ddfe"),
    amount: 101,
    date: '2024-01-01'
  },
  {
    _id: ObjectId("659587844dd874d5dac5de02"),
    number: 2,
    owner_id: ObjectId("659587844dd874d5dac5ddfb"),
    cooperative_id: ObjectId("659587844dd874d5dac5ddff"),
    amount: 102,
    date: '2024-01-01'
  },
  {
    _id: ObjectId("659587844dd874d5dac5de03"),
    number: 3,
    owner_id: ObjectId("659587844dd874d5dac5ddfc"),
    cooperative_id: ObjectId("659587844dd874d5dac5ddff"),
    amount: 202,
    date: '2024-01-01'
  },
  {
    _id: ObjectId("659587844dd874d5dac5de04"),
    number: 4,
    owner_id: ObjectId("659587844dd874d5dac5ddfb"),
    cooperative_id: ObjectId("659587844dd874d5dac5de00"),
    amount: 103,
    date: '2024-01-01'
  },
  {
    _id: ObjectId("659587844dd874d5dac5de05"),
    number: 5,
    owner_id: ObjectId("659587844dd874d5dac5ddfc"),
    cooperative_id: ObjectId("659587844dd874d5dac5de00"),
  }
]

```

```

    amount: 203,
    date: '2024-01-01'
  },
  {
    _id: ObjectId("659587844dd874d5dac5de06"),
    number: 6,
    owner_id: ObjectId("659587844dd874d5dac5ddfd"),
    cooperative_id: ObjectId("659587844dd874d5dac5de00"),
    amount: 303,
    date: '2024-01-01'
  },
  {
    _id: ObjectId("65958997cf6b78c9659626d2"),
    number: 7,
    owner_id: ObjectId("659587844dd874d5dac5ddfb"),
    amount: 100,
    date: '2024-01-03',
    cooperative_id: ObjectId("65958997cf6b78c9659626d1")
  },
  {
    _id: ObjectId("65958997cf6b78c9659626d3"),
    number: 8,
    owner_id: ObjectId("6595882dcf6b78c9659626c8"),
    amount: 300,
    date: '2024-01-03',
    cooperative_id: ObjectId("65958997cf6b78c9659626d1")
  },
  {
    _id: ObjectId("65958ac1cf6b78c9659626db"),
    number: 9,
    owner_id: ObjectId("659587844dd874d5dac5ddfb"),
    amount: 150,
    date: '2024-01-03',
    cooperative_id: ObjectId("65958997cf6b78c9659626d1")
  },
  {
    _id: ObjectId("65958ac1cf6b78c9659626dc"),
    number: 9,
    owner_id: ObjectId("6595882dcf6b78c9659626c8"),
    amount: 250,
    date: '2024-01-03',
    cooperative_id: ObjectId("65958997cf6b78c9659626d1")
  }
]

```

Заключение

В рамках лабораторная был разработан веб сервер с пользовательским интерфейсом к ИС учета кооперативов и их владельцев с минимальным функционалом для реализации следующих бизнес-процессов: регистрация кооператива и передача пая. Хранение данных было реализовано в MongoDB, доступ к которой был осуществлен через пакет pymongo.