

# Система управління клієнтською базою як SaaS на прикладі страхового брокера

Курсова робота  
Студент: Василенко Андрій Миколайович



# Мета

Метою курсової роботи є дослідження та реалізація системи управління клієнтською базою за моделлю Software-as-a-Service (SaaS) для потреб брокерської компанії у сфері страхування, а також створення висновків та перспектив розвитку такої системи.

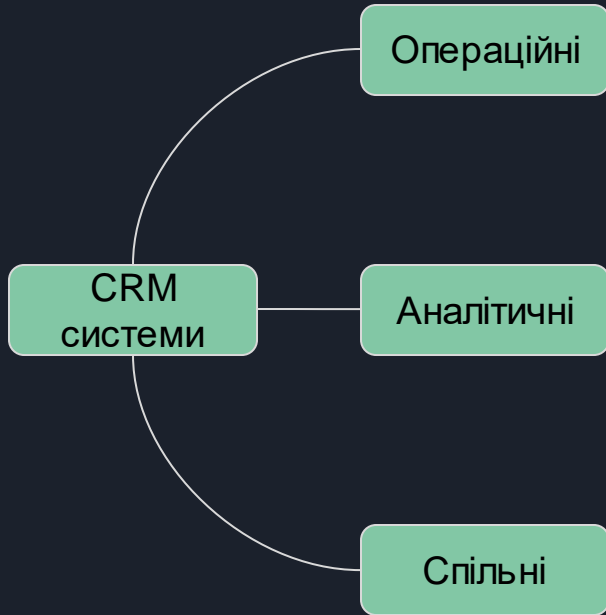
# Короткий огляд CRM систем

Управління відносинами з клієнтами (англ. Customer relationship management (CRM), укр. сі-ар-ем) — поняття, що охоплює концепції, котрі використовуються компаніями для управління взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

Існують три основні типи CRM систем:



# Функції CRM систем за типами



Більшість CRM систем. До їх функцій відносять ведення реєстру трафіку (дзвінки, листи, заявки на сайті), зберігання в базі даних інформації про клієнтів, 3. автоматизацію документообігу, відображення поточного статусу угод, нагадування про заплановані події (дзвінки, листи, зустрічі);

Допомагають простежити закономірності в продажах. Мета аналітичних CRM - аналіз інформації по клієнтах та результатах продажів з метою створення більш ефективної стратегії роботи.

Або колабораційні. Такі CRM системи налагоджують комунікації з клієнтами для збору зворотного зв'язку. Інформація, отримана з їх допомогою, допомагає скорегувати асортимент товарів, цінову політику, а також процес обслуговування покупців.



# Специфіка вимог CRM системи для компаній у сфері страхування

## 1. Особливості меню:

окрім традиційних розділів меню, таких як Клієнт чи Договори, необхідність мати додатково розділи меню для об'єктів страхування та процесу врегулювання страхових подій.

## 2. Специфічна термінологія:

форми договорів повинні містити поля для обов'язкових атрибутів саме страхових договорів, наприклад “франшиза”, “страхова премія”, “бонус-малус”, “страхувальник та застрахований”

## 3. Договори нефіксованого строку:

система має підтримувати договори на декілька днів (туристичне страхування), квартал, півроку або рік чи більше.

## 4. Договір може включати багато об'єктів:

один договір з одним клієнтом може мати безліч об'єктів страхування, як наприклад страхування автомобілів компанії таксі або страхування співробітників великої корпорації.

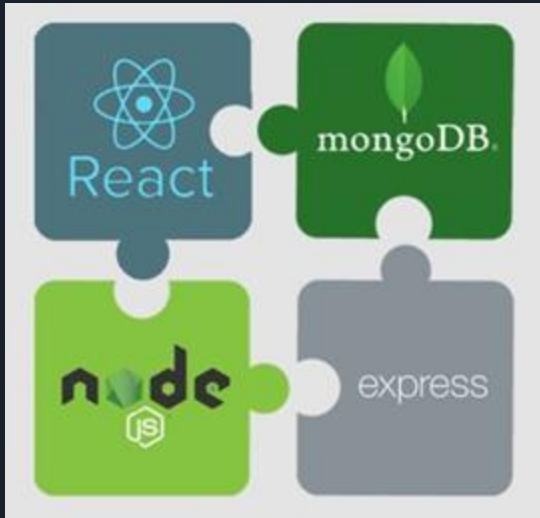


# Вибір моделі SaaS: “за” та “проти”

SaaS (англ. Software-as-a-Service - програмне забезпечення як послуга) - модель поширення програм споживачам, при якій постачальник розробляє веб-програму, розміщує її й керує нею (самостійно або через третіх осіб) з метою використання її замовниками через інтернет.

Переваги	Недоліки
Вартість	Ризики безпеки
Обслуговування	Залежність від швидкості інтернету
Оновлення програмного забезпечення	Значно менший контроль за системою.
Час розгортання і продуктивність	
Доступність робочого місця	

# Технології: MERN stack



MongoDB - нереляційна база даних

Express.js - фреймворк для створення веб-додатків. У нього вбудований функціонал для вирішення багатьох проблем, зокрема, для налагодження маршрутизації.

React.js - бібліотека для створення веб-інтерфейсу

Node.js - платформа для створення серверної частини

+ безліч допоміжних бібліотек

# Реалізація: створення схеми для MongoDB

```
server > customers > customerModel.js > _
1  const mongoose = require('mongoose');
2  const Schema = mongoose.Schema;
3
4  const schema = new Schema({
5    name: { type: String, default: "" },
6    type: { type: String, default: "" },
7    address: { type: String, default: "" },
8    phone: { type: String, default: "" },
9    email: { type: String, default: "" },
10   dateOfBirth: { type: Date },
11   inn: { type: Number, default: 0 },
12   documentType: { type: String },
13   documentNumber: { type: String },
14   documentIssuer: { type: String },
15   documentIssueDate: { type: Number },
16   documentExpiryDate: { type: Number },
17   gender: { type: String },
18   comments: { type: String, default: "" },
19   citizenship: { type: String, default: "" },
20   bankName: { type: String, default: "" },
21   edrpou: { type: Number, default: "" },
22   accountNumber: { type: String, default: "" },
23   mfo: { type: Number },
24   director: { type: String, default: "" },
25   manager: { type: String, default: "" },
26   customerPhotos: { type: Array, default: [] },
27   objects: { type: Array, default: [] },
28   contracts: { type: Array, default: [] },
29   settlements: { type: Array, default: [] },
30   createdAt: { type: Number, default: Date.now() },
31 });
32 schema.set('toJSON', { virtuals: true });
33 module.exports = mongoose.model('customers', schema);
```


Схема зберігання даних в MongoDB формується у вигляді JSON об'єкту.

Приклад реалізації схеми можна побачити на малюнку зліва. Ця схема була створена для зберігання даних по клієнтам.



# Реалізація: створення серверної частини (server.js)

```
server > JS server.js > ...
1  const express = require('express');
2  const cors = require('cors');
3  const bodyParser = require('body-parser');
4  const app = express();
5  const mongoose = require('mongoose');
6  require('dotenv').config()
7
8  mongoose.Promise = global.Promise;
9  mongoose.connect(process.env.MONGODB_URI,
10     {
11       useCreateIndex: true, useUnifiedTopology: true, useNewUrlParser: true
12     }).then(
13     () => { console.warn('DB connected') },
14     err => { console.log('Error in DB connection', err) }
15   );
16
17   app.use(cors());
18
19   app.use(bodyParser.urlencoded({ extended: false }));
20   app.use(bodyParser.json());
21
22   // start server
23   const PORT = process.env.NODE_ENV === 'production' ? (process.env.PORT || 80) : 5000;
24   app.listen(PORT, () => {
25     console.log(`Server listening on port ${PORT}`);
26   });
```



## Реалізація: створення серверної частини (реалізація функції API)

```
router.post("/addcustomer", passport.authenticate("jwt", { session: false }), (req, res) => {
  Customer.findOne({ inn: req.body.inn, dateOfBirth: req.body.dateOfBirth })
    .then((customer) => {
      if (customer) {
        console.warn("tried to add customer but found existing one", customer);
        return res.json(customer);
      } else {
        var newCustomer = new Customer({ ...req.body });
        newCustomer.save().then((cust) => {
          return res.json(cust);
        });
      }
    })
    .catch((err) => res.status(500).send("Whoops, customer add operation failed"));
});
```

# Реалізація: створення серверної частини (бібліотека для аутентифікації passport.js)

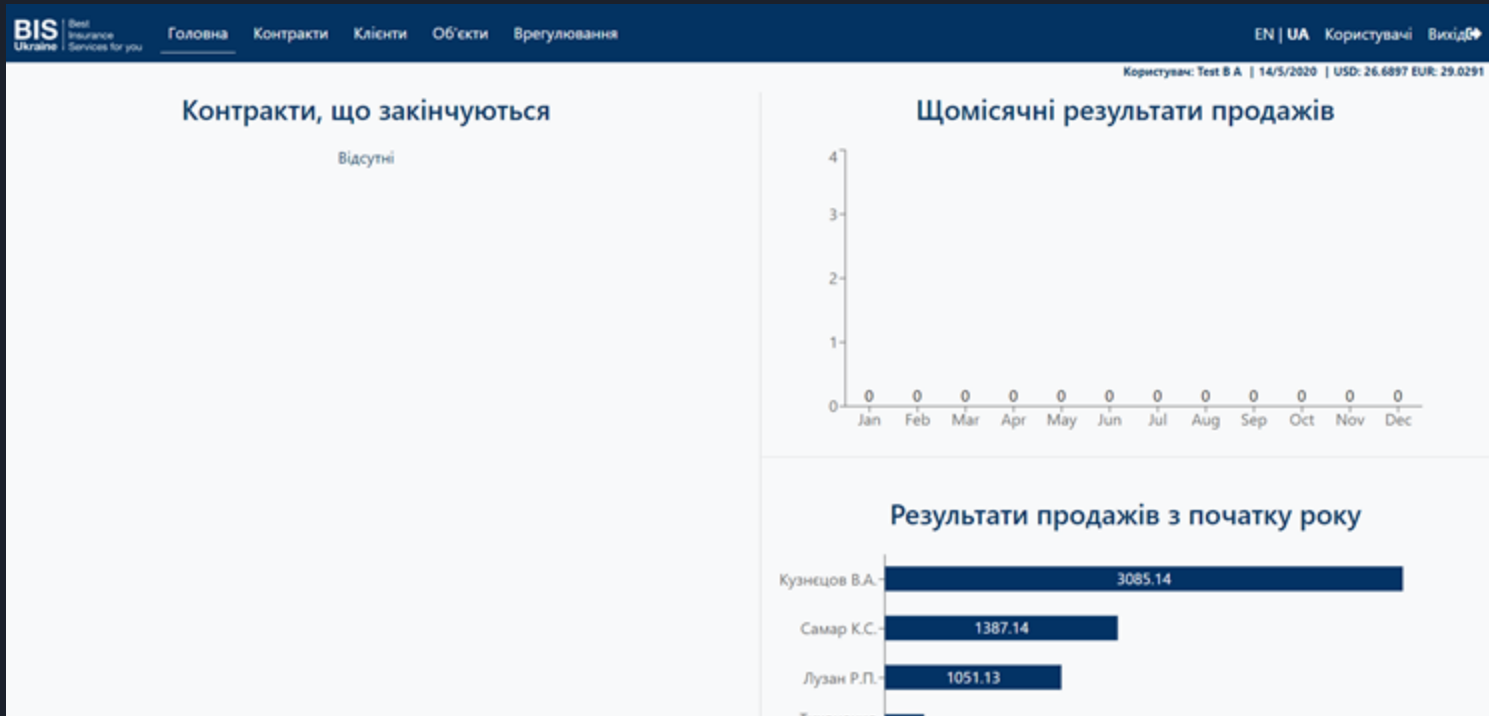
```
server > JS passport.js > <unknown> > module.exports
1  const JWTStrategy = require('passport-jwt').Strategy;
2  const ExtractJWT = require('passport-jwt').ExtractJwt;
3  const mongoose = require('mongoose');
4  const User = mongoose.model('User');
5  require('dotenv').config();
6
7  const opts = {};
8
9  opts.jwtFromRequest = ExtractJWT.fromAuthHeaderAsBearerToken();
10 opts.secretOrKey = process.env.SECRET;
11
12 module.exports = passport => {
13   passport.use(new JWTStrategy(opts, (jwt_payload, done) => {
14     User.findById(jwt_payload.id)
15       .then(user => {
16         if(user) {
17           return done(null, user);
18         }
19         return done(null, false);
20       })
21       .catch(err => console.error(err));
22   }));
23 }
```

# Реалізація: компонент веб-інтерфейсу

























```
class Header extends Component {
  constructor(props) {
    super();
    this.state = {
      current: "home",
    };
  }

  render() {
    return (
      <Navbar style={navBarStyle}>
        <Navbar.Brand style={{ width: 145, padding: 0 }}>
          <Link to="/home">
            
          </Link>
        </Navbar.Brand>
        <Nav className="mr-auto">
          <Link className={"nav-link " + (current === "/home" ? "active" : "")} to="/home">
            <Trans i18nKey="home">Home</Trans>
          </Link>
        </Nav>
      </Navbar>
    );
  }
}
```

# Результат: загальний вигляд веб-інтерфейсу



# Результат: таблиця договорів




BIS Best Insurance Services for you							
Головна <u>Контракти</u> Клієнти Об'єкти Врегулювання							
ДОДАТИ НОВИЙ		ЕКСПОРТ В EXCEL					
Інструменти	Тип	Страхова Компанія	Номер Договору	Дата початку	Дата закінчення		
▶   	ОСЦПВ	АТ Страхова компанія "ІНГО"	0173810	15/02/2020	14/02/2021	Скурато	
▶   	ОСЦПВ	АТ Страхова компанія "ІНГО"	1149031	07/02/2020	06/02/2021	Белов М	
▶   	ОСЦПВ	АТ Страхова компанія "ІНГО"	2АО/492	08/02/2020	07/02/2021	ТОВ "НІ	
▶   	ДСЦПВ	АТ Страхова компанія "ІНГО"	3310303	04/02/2020	03/02/2021	Дучук В	
▶   	ОСЦПВ	АТ Страхова компанія "ІНГО"	4190544	04/02/2020	03/02/2021	Дучук В	
▶   	ОСЦПВ	ПрАТ "СК "ПЗУ Україна"	5AM.935	24/01/2020	23/01/2021	ТОВ "ТР	
▶   	ОСЦПВ	ПрАТ "АСК "ІНГО Україна"	6158176	08/04/2020	07/04/2021	Чайка І	
▶   	ДСЦПВ	АТ Страхова компанія "ІНГО"	7310307	28/01/2020	27/01/2021	Дмитре	

# Результат: розгорнуті дані договору


**BIS** Best Insurance Services for you Ukraine

Головна Контракти Клієнти Об'єкти Врегулювання

**ДОДАТИ НОВИЙ** **ЕКСПОРТ В EXCEL**

Інструменти	Тип	Страхова Компанія	Номер Договору	Дата початку	Дата закінчення	Клієнт
  	ОСЦПВ	АТ Страхова компанія "ІНГО"	0173810	15/02/2020	14/02/2021	Скुरатовський Олекса

**Об'єкти**

Інструмен...	Дата об'єк...	Власник	Документ	Серія і но...	Орган що ...	Дата видачі	Бренд
	29/04/2020	Кириченк...	технічний ...	CXI 832172	3247	12/02/2019	Opel

Врегулювання

Клієнти

Документи






















Previous Page **1** of 1 5 rows ▾

# Результат: дані об'єктів

**BIS** Best Insurance Services for you  
Україна

Головна   Контракти   Клієнти   Об'єкти   Врегулювання

**+** ДОДАТИ НОВИЙ   **+** ЕКСПОРТ В EXCEL

Інструменти	Дата об'єкту	Власник	Документ	Се...	Op
▶   	29/04/2020	Белов Михайло В'ячеславович	технічний паспорт	CX...	654
▶   	29/04/2020	Кириченко Олена Миколаївна	технічний паспорт	CX...	324
▶   	15/04/2020	Радіонов Павло Павлович	технічний паспорт	CX...	TC
▶   	15/04/2020	ТОВ "НІГІНА"	технічний паспорт	CX...	800
▶   	11/04/2020	ТОВ "ТРАНСАВТО+"	технічний паспорт	CX...	TC
▶   	08/04/2020	Саєнко Віктор Іванович	технічний паспорт	A...	УВ
▶   	08/04/2020	Влаа Kpictiou Jovocny Icar Gotic	технічний паспорт	CX...	TC





# Практична значимість

За результатами роботи, ми отримали першу версію CRM системи, котра відповідає базовим вимогам користувачів. Ця система ще має великий потенціал для поліпшення та розвитку, але можливість комерційного використання застосування підтвердилось реальним випадком такого використання.

ТОВ "БІ АЙ ЕС Україна" (<https://bisukraine.com>) - брокерська компанія у сфері страхування, котра почала використовувати дану CRM систему.

# Перспективи подальшого розвитку

