

**СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“**



Учебен Проект

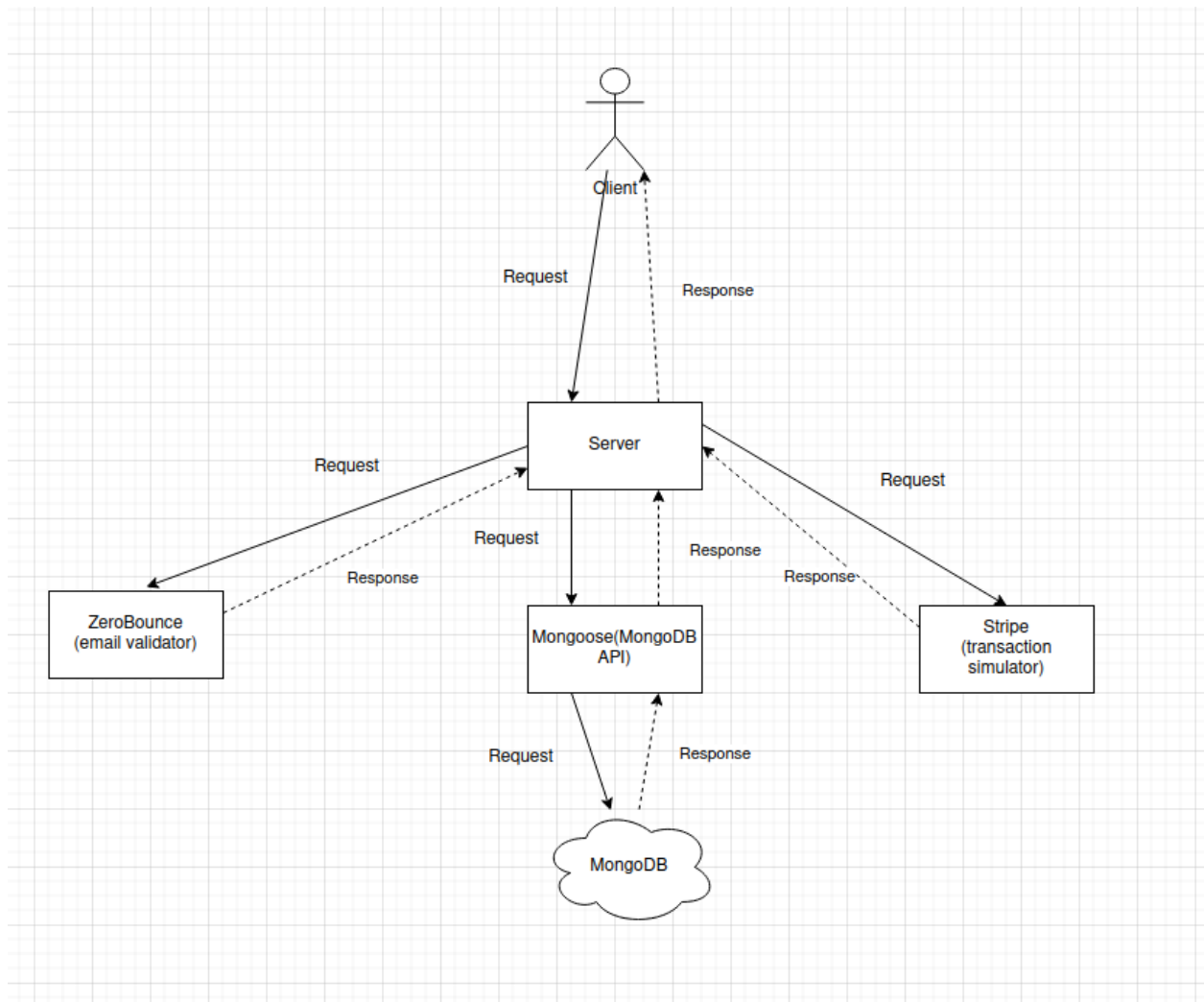
по Системи за е-бизнес

На тема: “Автокъща с опция за лизинг”

Изготвил: Алкан Мустафа, 72018

1. Архитектура

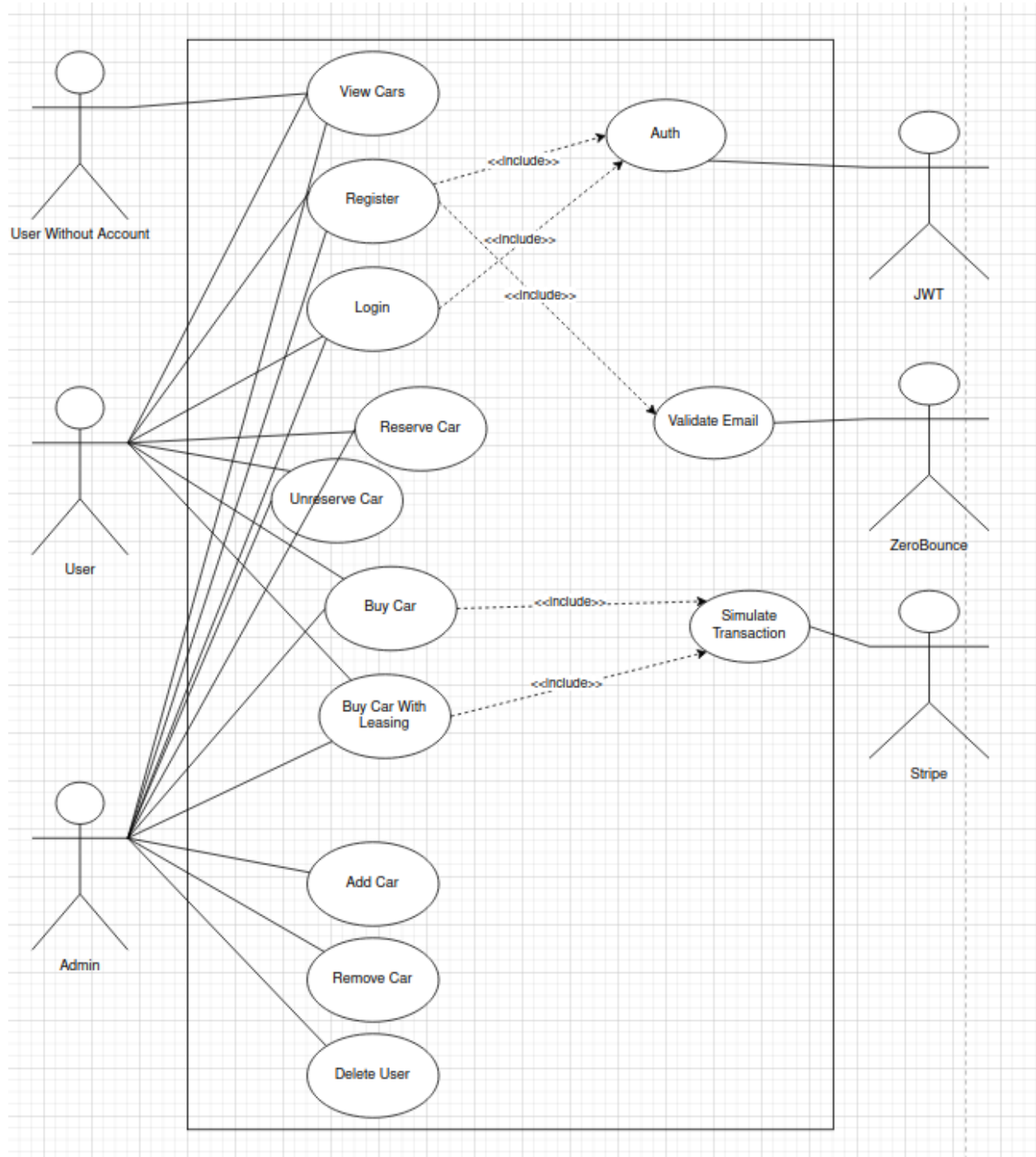
Системата се състои от 2 компонента: Сървър и Бази Данни. Клиентската част не е имплементирана. Съответно клиентът взаимодейства със системата, чрез заявки към сървъра и сървърът след взаимодействие с базата данни връща резултат на клиента.



2. Технологии

За сървър частта е използван Typescript и nodejs, а за база данни е използвано MongoDB.

3. Use Case Diagram



Представя функционалностите, които предоставя системата и услугите на външните системи, които се използват.

4. Описание на функционалности

а. Нерегистрирани Потребители -

Нерегистрираните потребители могат, само да разглеждат наличните за продан автомобили, без правото да резервират или да закупуват автомобил.

б. Регистрирани Потребители - Регистрираните

потребители могат да разглеждат наличните за продан автомобили, да резервират някой автомобил и да махнат резервация на някой автомобил, който по рано са резервирали, да купят автомобил както плащайки пълната цена наведнъж, така и с възможност за изплащане с първоначална вноска(20%,30%,40%,50%) и брой месеци(12,24,48,60).

с. Административни Потребители -

Административните потребители разполагат с функционалностите на регистрираните потребители и освен това имат право да добавят коли за продан. Да махат коли(само ако тя НЕ е резервирана от клиент), да махат потребители(само ако потребителят НЯМА лизинг за плащане). Също така административните потребители могат да видят всички лизингови покупки(които все още не са изплатени) на даден потребител.

d. Предлагами Автомобили - Предлагащите автомобили могат да бъдат Дизелови, Бензинови, Газови, Бензин/Газ, Хибрид, Електрически и могат да бъдат с Автоматична или Механична трансмисия.

e. Предлагащият Лизинг - Предлагащият лизинг е следният: Потребителят плаща първоначална вноска в размер 20%, 30%, 40% или 50% от цената на автомобила и останалата сума може да се раздели на брой месечни вноски 12 месеца, 24 месеца, 48 месеца или 60 месеца.

5. Схема на базата от данни

```
export const CarSchema = new Schema({
  id: Schema.Types.ObjectId,
  VIN: {
    type: Schema.Types.String,
    required: true,
    unique: true
  },
  brand: {
    type: Schema.Types.String,
    required: true
  },
  model: {
    type: Schema.Types.String,
    required: true
  },
  year: {
    type: Schema.Types.Number,
    required: true
  },
  engine: {
    type: Schema.Types.String,
    required: true,
    enum: ['Diesel', 'Petrol', 'LPG', 'Petrol/LPG', 'Hybrid', 'Electric']
  },
  transmission: {
    type: Schema.Types.String,
    required: true,
    enum: ['Manual', 'Automatic']
  },
  kilometers: {
    type: Schema.Types.Number,
    required: true
  },
  price: {
    type: Schema.Types.Number,
    required: true
  },
  state: {
    type: Schema.Types.String,
    default: 'For Sale',
    enum: ['For Sale', 'Reserved', 'Sold']
  }
})
```

```
export const UserSchema = new Schema({
  id: Schema.Types.ObjectId,
  username: {
    type: Schema.Types.String,
    required: true,
    unique: true
  },
  email: {
    type: Schema.Types.String,
    required: true,
    unique: true
  },
  password: {
    type: Schema.Types.String,
    required: true
  },
  firstName: {
    type: Schema.Types.String,
    required: true
  },
  lastName: {
    type: Schema.Types.String,
    required: true
  },
  debitCardNumber: {
    type: Schema.Types.String,
    required: true
  },
  cardExpMonth: {
    type: Schema.Types.String,
    required: true
  },
  cardExpYear: {
    type: Schema.Types.String,
    required: true
  },
  cardCVC: {
    type: Schema.Types.String,
    required: true
  },
},
```

```
boughtCarVINS: {  
    type: [String],  
    default: [],  
    required: false  
},  
reservedCarVINS: {  
    type: [String],  
    default: [],  
    required: false  
},  
isAdmin: {  
    type: Schema.Types.Boolean,  
    default: false  
}  
})
```



```

export const LeaseSchema = new Schema({
  id: Schema.Types.ObjectId,
  VIN: {
    type: Schema.Types.String,
    required: true,
    unique: true
  },
  buyerUsername: {
    type: Schema.Types.String,
    required: true,
  },
  percentFirstPayment: {
    type: Schema.Types.Number,
    required: true,
    default: 20
  },
  paymentCnt: {
    type: Schema.Types.Number,
    required: true,
    default: 12
  },
  leasePayment: {
    type: Schema.Types.Number,
    required: true
  },
  paymentCntRemaining: {
    type: Schema.Types.Number,
    required: true
  }
})

```

6. Използвани Външни Системи

За Базата Данни се използва Cloud Mongo DB, който се достъпва чрез mongoose като се предостави и API Key-а на профил.

За валидация на email се използва ZeroBounce, предоставя се API Key-а на профил и чрез api fetch с въведен email се прави валидация(всяка валидация на email изразходва по 1 кредит от профила и евентуално ще е нужно заплащане, при продължително ползване на услугата). Проверява се както

формата(дали email формата е валиден) така и дали реално съществува такъв email.

За симулиране на транзакция се използва Stripe, предоставя се API Key-а на профил и чрез stripe библиотеката се симулира транзакция. Транзакция се симулира при купуване на кола на пълна цена(прехвърля се цялата сума на колата) и при закупуване на кола на лизинг(първо се прехвърля сумата на първоначалната вноска и на всяко следващо плащане се прехвърля сумата на вноската). След имплементация на front end(client side) е възможно да се интегрира и възможност за валидиране на дебитна карта(Stripe забранява правенето на заявки с raw data на дебитна карта, трябва данните на дебитната карта да бъдат токенизирани, това става чрез библиотеката stripe js, тя приема данните директно от client side-а, затова на този етап не може услугата за валидация на дебитна карта да бъде използвана).

7. Насоки за стартиране на приложението

Клониране на проекта -> влизане в папката сървър на проекта
-> npm install -> npm start -> сървърът е пуснат -> може endpoint-ите да бъдат тествани чрез заявки с postman

8. Връзка към актуалната версия на проекта(branch main)

Линк: <https://github.com/alkan369/Car-Dealership>