

**Курсов проект**

**по**

**Системи с бази от данни**

Изработил: Проверил:

Михаил Бисеров Георгиев /доц. Гео Кунев /

III курс, 1а група, Факултет: Фита, Катедра: СИТ, Специалност: СИТ Факултетен номер: 19621820

Дисциплина: Системи с бази от данни.

Задание УД

Да се проектира и реализира база от данни за **БАНКА**, която да съхранява следната информация:

* Клиент – име, ЕГН, адрес, телефон:
* Служител – име, позиция, телефон:
* Сметка – номер, клиент, вид валута, лихва, наличност:
* Транзакция - вид, клиент, служител, сметка, сума.

Правила:

* Всеки клиент може да има повече от една сметка
* Всеки клиент може да прави транзакции от различен вид

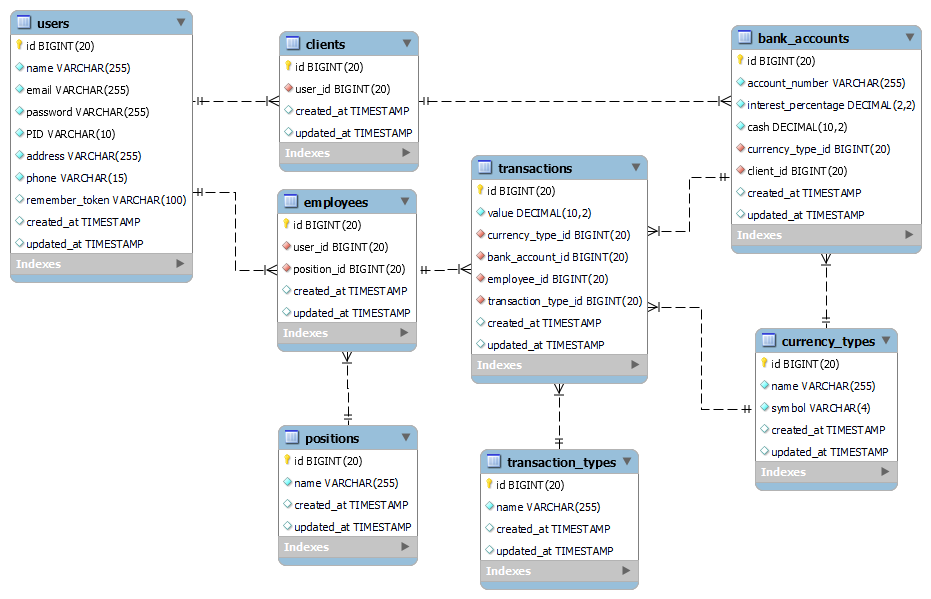
Базата от данни трябва да е **нормализирана** и да позволява:

1. **Въвеждане и корекции** на данни.
2. **Търсене** на сметки по: клиент, номер, ЕГН.
3. **Справки** за:
   * сметки на клиент;
   * транзакции на клиент, подредени по дата;
   * транзакции за служител;
   * транзакции за период.

Обяснителната записка към проекта трябва да съдържа:

* Задание
* Модели ( Oracle Data Moduler ) - 6 седмица
* Примерни данни -10 седмица
* SQL команди – DLL,DML
* Резултати от изпълнението на заявките -12 седмица

Релационен модел:



КОД НА ПРОГРАМАТА

*Създаване на таблици:*

|  |  |
| --- | --- |
| SET @OLD\_UNIQUE\_CHECKS = @@UNIQUE\_CHECKS, UNIQUE\_CHECKS = 0;  SET @OLD\_FOREIGN\_KEY\_CHECKS = @@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS = 0; | |
| -- | Schema uni-db |
| CREATE SCHEMA IF NOT EXISTS `uni\_db` DEFAULT CHARACTER SET utf8;  USE `uni\_db`; | |
| -- | Table `bank\_accounts` |
| DROP TABLE IF EXISTS `bank\_accounts`; CREATE TABLE IF NOT EXISTS `bank\_accounts` (  `id` INT NOT NULL AUTO\_INCREMENT,  `account\_number` VARCHAR(256) NOT NULL,  `interest\_percentage` DECIMAL(5, 2) NULL DEFAULT 0,  `cash` DECIMAL(12, 2) NULL DEFAULT 0,  `currency\_type\_id` INT NOT NULL,  `client\_id` INT NOT NULL,  `created\_at` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY (`id`),  UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE,  INDEX `bank\_accounts\_clinet\_id\_FK\_idx` (`client\_id` ASC) VISIBLE, INDEX `bank\_accounts\_currency\_type\_id\_FK\_idx` (`currency\_type\_id` ASC)  VISIBLE,  CONSTRAINT `bank\_accounts\_clinet\_id\_FK` FOREIGN KEY (`client\_id`) REFER ENCES `clients` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  CONSTRAINT `bank\_accounts\_currency\_type\_id\_FK` FOREIGN KEY (`currency\_t ype\_id`) REFERENCES `currency\_types` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  ) ENGINE = InnoDB; | |
| -- | Table `clients` |
| DROP TABLE IF EXISTS `clients`; CREATE TABLE IF NOT EXISTS `clients` (  `id` INT NOT NULL AUTO\_INCREMENT,  `name` VARCHAR(256) NOT NULL,  `PID` VARCHAR(10) NOT NULL,  `address` VARCHAR(256) NOT NULL,  `phone` VARCHAR(15) NOT NULL, PRIMARY KEY (`id`),  UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE  ) ENGINE = InnoDB; | |
| -- | Table `currency\_types` |
|  |

|  |  |
| --- | --- |
| DROP TABLE IF EXISTS `currency\_types`; CREATE TABLE IF NOT EXISTS `currency\_types` (  `id` INT NOT NULL AUTO\_INCREMENT,  `name` VARCHAR(256) NOT NULL,  `symbol` VARCHAR(4) NOT NULL, PRIMARY KEY (`id`),  UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE  ) ENGINE = InnoDB; | |
| -- | Table `employees` |
| DROP TABLE IF EXISTS `employees`; CREATE TABLE IF NOT EXISTS `employees` (  `id` INT NOT NULL AUTO\_INCREMENT,  `name` VARCHAR(256) NOT NULL,  `phone` VARCHAR(15) NOT NULL,  `position\_id` INT NOT NULL, PRIMARY KEY (`id`),  UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE,  INDEX `employees\_position\_id\_FK\_idx` (`position\_id` ASC) VISIBLE, CONSTRAINT `employees\_position\_id\_FK` FOREIGN KEY (`position\_id`) REFER  ENCES `positions` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  ) ENGINE = InnoDB; | |
| -- | Table `positions` |
| DROP TABLE IF EXISTS `positions`; CREATE TABLE IF NOT EXISTS `positions` (  `id` INT NOT NULL AUTO\_INCREMENT,  `name` VARCHAR(256) NOT NULL, PRIMARY KEY (`id`),  UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE  ) ENGINE = InnoDB; | |
| -- | Table `transaction\_types` |
| DROP TABLE IF EXISTS `transaction\_types`; CREATE TABLE IF NOT EXISTS `transaction\_types` (  `id` INT NOT NULL AUTO\_INCREMENT,  `name` VARCHAR(256) NOT NULL, PRIMARY KEY (`id`),  UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE  ) ENGINE = InnoDB; | |
| -- | Table `transactions` |
| DROP TABLE IF EXISTS `transactions`; CREATE TABLE IF NOT EXISTS `transactions` (  `id` INT NOT NULL AUTO\_INCREMENT,  `value` DECIMAL(12, 2) NOT NULL,  `employee\_id` INT NOT NULL,  `bank\_account\_id` INT NOT NULL,  `transaction\_type\_id` INT NOT NULL,  `created\_at` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY (`id`), | |

UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE,

INDEX `transactions\_employee\_id\_FK\_idx` (`employee\_id` ASC) VISIBLE, INDEX `transactions\_bank\_account\_id\_FK\_idx` (`bank\_account\_id` ASC) VIS

IBLE,

INDEX `transactions\_transaction\_type\_id\_FK\_idx` (`transaction\_type\_id` ASC) VISIBLE,

CONSTRAINT `transactions\_employee\_id\_FK` FOREIGN KEY (`employee\_id`) RE FERENCES `employees` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `transactions\_bank\_account\_id\_FK` FOREIGN KEY (`bank\_account

\_id`) REFERENCES `bank\_accounts` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `transactions\_transaction\_type\_id\_FK` FOREIGN KEY (`transact ion\_type\_id`) REFERENCES `transaction\_types` (`id`) ON DELETE NO ACTION O N UPDATE NO ACTION

) ENGINE = InnoDB;

SET FOREIGN\_KEY\_CHECKS = @OLD\_FOREIGN\_KEY\_CHECKS; SET UNIQUE\_CHECKS = @OLD\_UNIQUE\_CHECKS;

*Въвеждане на данни:*

INSERT INTO currency\_types(`name`, `symbol`) VALUES ('Bulgarian lev', 'лв.'),

('Euro', '€'),

('United States dollar', '$'), ('Japanese yen', '¥'),

('Pound sterling', '£');

INSERT INTO positions(`name`) VALUES ('Financial Analyst'),

('Personal Financial Adviser'), ('Relationship Manager'), ('Accountant'),

('Auditor'), ('Branch Manager'), ('Loan Officer'),

('Bank Teller'),

('Treasurer');

INSERT INTO `transaction\_types` (`name`) VALUES ('ATM'),

('Charge'),

('Check'),

('Deposit'),

('Online'),

('POS'),

('Transfer'),

('Withdrawal');

INSERT INTO `employees` (`name`, `phone`, `position\_id`) VALUES ('Kira', '(811) 603-3553 ', 2),

('Richie', '1-254-758-9289 ', 2),

('Aron', '(981) 125-7840 ', 4),

('Einar', '543-454-6131 x3', 2),

('Susie', '(409) 342-3282 ', 2),

('Jude', '(941) 466-7445', 8),

('Myron', '495-661-2319 x8', 4),

('Buddy', '268.920.8951 x3', 6),

('Estelle', '014.887.9639', 2),

('Rozella', '855-772-9774', 6),

('Zechariah', '091.850.0465', 6),

('Waldo', '956-785-6519', 8),

('Magnus', '042.479.7215 x4', 1),

('Adelle', '1-373-867-7694 ', 4),

('Caden', '(223) 881-3367 ', 6),

('Abigale', '(186) 295-1926 ', 1),

('Dalton', '1-048-613-7598', 1),

('Marlee', '996.514.1756', 1),

('Golda', '969.495.4681 x1', 2),

('Pearline', '(891) 587-8054', 5);

INSERT INTO `clients` (`name`, `PID`, `address`, `phone`) VALUES

('Zachariah','5571331562',"711 O'Reilly Isle",'(595) 972-3473 '), ('Brice','8843493664','764 Carol Brook','446.806.1226'), ('Wilmer','5193941889','226 Runte Forge','084.215.7845'), ('Aida','4332184394','455 Emmitt Alley','(593) 039-2198 '), ('Jamel','6309316599','54932 Murray Hollow','983.518.5074 x9'), ('Maximillian','5567833368','359 Raegan Glen','1-784-435-3870 '), ('Beth','2370949112','04141 Medhurst Vista','026.729.9501 x3'), ('Murl','3091585099','82008 Zieme Point','1-306-517-7342'), ('Kelly','1016692048','0512 Hayes Drives','1-970-052-3206'), ('Michaela','3910427894','3200 Bednar Road','1-240-482-0781 ');

INSERT INTO `bank\_accounts` (`account\_number`,`interest\_percentage`,`cash

`,`currency\_type\_id`,`client\_id`,`created\_at`) VALUES

('6809731976','0.9','5451720.00',4,2,'2020-11-01 09:13:25'),

('1676134712','0.00','349400.00',4,10,'2020-11-01 11:27:35'),

('6125776066','5.00','8231553.00',4,6,'2020-11-01 00:02:09'),

('5647485326','6.00','6327175.00',5,1,'2020-10-31 22:13:41'),

('3897456539','0.01','4949436.00',5,10,'2020-11-01 09:31:44'),

('1967620069','0.20','6026268.00',3,2,'2020-11-01 07:52:49'),

('2501141516','0.90','6650693.00',1,8,'2020-10-31 19:26:52'),

('7739475419','0.65','5660381.00',3,7,'2020-11-01 11:53:21'),

('1427876257','0.34','6795946.00',2,9,'2020-11-01 12:34:07'),

('5494223126','2.25','3406630.00',3,5,'2020-10-31 15:10:29');

INSERT INTO `transactions` (`value`,`employee\_id`,`bank\_account\_id`,`tran saction\_type\_id`,`created\_at`)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VALUES ('68.00', | | 14, 5, 8, '2020-11-01 00:49:01'), | | | |
| ('105.00', | 14, | 4, | 1, | '2020-11-01 | 10:47:53'), |
| ('562.00', | 20, | 3, | 3, | '2020-11-01 | 01:29:42'), |
| ('437.00', | 10, | 7, | 5, | '2020-11-01 | 04:11:04'), |
| ('717.00', | 19, | 6, | 5, | '2020-10-31 | 14:49:45'), |
| ('330.00', | 18, | 2, | 7, | '2020-10-31 | 17:26:45'), |
| ('823.00', | 16, | 3, | 5, | '2020-10-31 | 18:44:02'), |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ('616.00', | 14, | 7, | 4, | '2020-11-01 | 03:55:59'), |
| ('299.00', | 10, | 5, | 6, | '2020-10-31 | 22:38:21'), |
| ('416.00', | 18, | 7, | 8, | '2020-11-01 | 12:25:09'), |
| ('388.00', | 16, | 8, | 3, | '2020-11-01 | 08:30:56'), |
| ('865.00', | 13, | 8, | 4, | '2020-10-31 | 16:37:29'), |
| ('519.00', | 12, | 6, | 1, | '2020-10-31 | 14:27:55'), |
| ('286.00', | 2, | 4, | 5, | '2020-11-01 00:20:36'), | |
| ('779.00', | 6, | 1, | 1, | '2020-11-01 07:45:55'), | |
| ('345.00', | 18, 9, 5, '2020-11-01 13:08:42'), | | | | |
| ('182.00', | 11, 1, 2, '2020-10-31 21:28:56'), | | | | |
| ('304.00', | 5, | 6, | 5, | '2020-11-01 | 06:30:39'), |
| ('608.00', | 9, | 4, | 3, | '2020-11-01 | 02:37:12'), |
| ('129.00', | 9, | 2, | 6, | '2020-11-01 | 11:39:43'), |
| ('976.00', | 12, 9, 8, '2020-10-31 15:29:51'), | | | | |
| ('61.00', 10, | | 7, | 6, | '2020-11-01 | 00:49:14'), |
| ('374.00', 6, | | 5, | 8, | '2020-10-31 | 21:49:55'), |
| ('417.00', 9, | | 1, | 6, | '2020-11-01 | 06:39:08'), |
| ('999.00', | 16, 7, 5, '2020-11-01 11:14:26'), | | | | |
| ('347.00', | 6, | 4, | 7, | '2020-10-31 | 18:07:19'), |
| ('345.00', | 5, | 2, | 2, | '2020-11-01 | 05:26:54'), |
| ('993.00', | 8, | 8, | 8, | '2020-10-31 | 22:33:26'), |
| ('482.00', | 4, | 8, | 5, | '2020-10-31 | 18:18:46'), |
| ('707.00', | 5, | 1, | 1, | '2020-10-31 | 13:37:41'), |
| ('592.00', | 6, | 8, | 3, | '2020-11-01 | 00:53:52'), |
| ('495.00', | 6, | 3, | 3, | '2020-11-01 | 08:20:30'), |
| ('901.00', | 1, | 2, | 1, | '2020-10-31 | 15:07:57'), |
| ('488.00', | 11, | 4, | 5, | '2020-11-01 | 05:01:19'), |
| ('696.00', | 18, | 2, | 7, | '2020-11-01 | 02:38:29'), |
| ('832.00', | 17, | 4, | 3, | '2020-10-31 | 15:19:13'), |
| ('803.00', | 11, | 1, | 5, | '2020-11-01 | 04:34:38'), |
| ('420.00', | 7, | 7, | 7, | '2020-10-31 | 15:22:25'), |
| ('188.00', | 1, | 8, | 1, | '2020-11-01 | 03:58:50'), |
| ('116.00', | 20, 6, 1, '2020-11-01 12:09:24'), | | | | |
| ('610.00', | 4, | 7, | 6, | '2020-11-01 08:34:48'), | |
| ('546.00', | 6, | 9, | 7, | '2020-11-01 08:06:13'), | |
| ('235.00', | 11, | 6, | 5, | '2020-11-01 01:36:25'), | |
| ('124.00', | 11, | 3, | 8, | '2020-11-01 01:07:14'), | |
| ('641.00', | 16, | 9, | 2, | '2020-10-31 13:50:24'), | |
| ('638.00', | 9, | 8, | 1, | '2020-10-31 16:37:26'), | |
| ('736.00', | 9, | 2, | 8, | '2020-10-31 20:59:14'), | |
| ('40.00', 9, 10, | | | 7, | '2020-10-31 | 20:51:02'), |

('164.00', 13, 8, 4, '2020-11-01 06:51:43'),

('776.00', 16, 8, 1, '2020-10-31 17:09:50');

***Търсене на сметки по: номер***

SELECT bank\_accounts.account\_number, clients.name, bank\_accounts.interest\_percentage, CASE

WHEN currency\_types.id = 1 THEN CONCAT(bank\_accounts.cash, currency\_t ypes.symbol)

ELSE CONCAT(currency\_types.symbol, bank\_accounts.cash) END as cash,

bank\_accounts.created\_at FROM bank\_accounts

INNER JOIN clients ON clients.id = bank\_accounts.client\_id

INNER JOIN currency\_types ON currency\_types.id = bank\_accounts.currency

\_type\_id

WHERE bank\_accounts.account\_number = '5647485326';



***Търсене на сметки по: ЕГН***

SELECT bank\_accounts.account\_number, clients.name, bank\_accounts.interest\_percentage, CASE

WHEN currency\_types.id = 1 THEN CONCAT(bank\_accounts.cash, currency\_t ypes.symbol)

ELSE CONCAT(currency\_types.symbol, bank\_accounts.cash) END as cash,

bank\_accounts.created\_at FROM bank\_accounts

INNER JOIN clients ON clients.id = bank\_accounts.client\_id

INNER JOIN currency\_types ON currency\_types.id = bank\_accounts.currency

\_type\_id

WHERE clients.PID = '5571331562';



***Справка за сметки на клиент:***

SELECT bank\_accounts.account\_number, bank\_accounts.interest\_percentage, CASE

WHEN currency\_types.id = 1 THEN CONCAT(bank\_accounts.cash, currency\_t ypes.symbol)

ELSE CONCAT(currency\_types.symbol, bank\_accounts.cash) END as cash,

bank\_accounts.created\_at FROM bank\_accounts

INNER JOIN clients ON clients.id = bank\_accounts.client\_id

INNER JOIN currency\_types ON currency\_types.id = bank\_accounts.currency

\_type\_id

WHERE clients.name LIKE '%Michaela%';



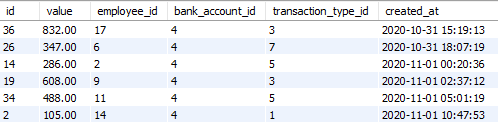
***Справка за транзакции на клиент, подредени по дата:***

SELECT transactions.\* FROM transactions

INNER JOIN bank\_accounts ON transactions.bank\_account\_id = bank\_account s.id

INNER JOIN clients ON clients.id = bank\_accounts.client\_id WHERE clients.name LIKE '%Zachariah%'

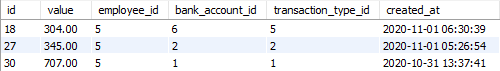
ORDER BY transactions.created\_at;



***Справка за транзакции за служител:***

SELECT transactions.\* FROM transactions

INNER JOIN employees ON employees.id = transactions.employee\_id WHERE employees.name LIKE '%Susie%';



***Справка за транзакции за период:***

SELECT clients.name AS client\_name, employees.name AS employee\_name, bank\_accounts.account\_number, transactions.value, transactions.created\_at

FROM transactions

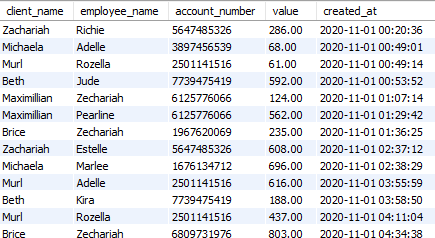
INNER JOIN bank\_accounts ON transactions.bank\_account\_id = bank\_account s.id

INNER JOIN clients ON clients.id = bank\_accounts.client\_id INNER JOIN employees ON employees.id = transactions.employee\_id

WHERE transactions.created\_at BETWEEN '2020-11-1 00:00:00' AND '2020-11-1

05:00:00'

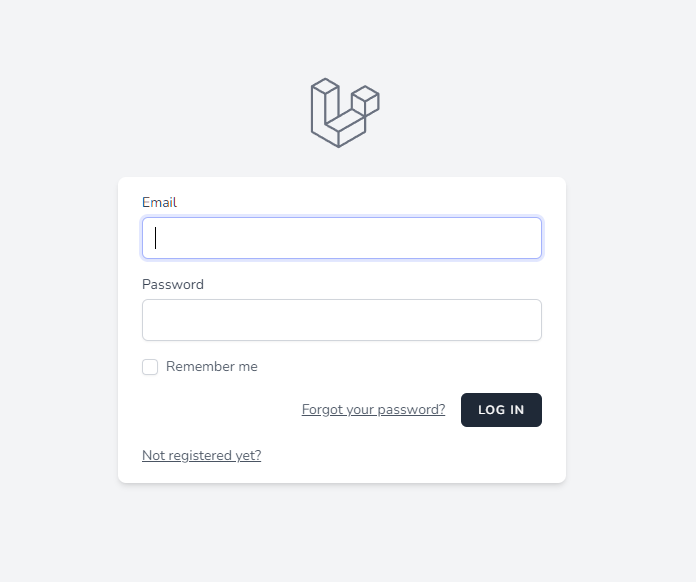
ORDER BY transactions.created\_at;



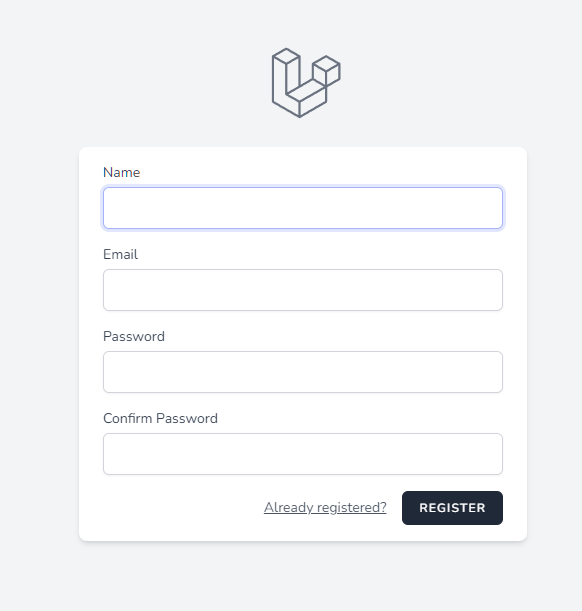
Използвани технологии

* Laravel – back-end
* Vue 3 – front-end
* MySQL – база от данни
* MySQL Workbench – модел и администриране на базата

Ръководство за потребителя

 Началната страница на проекта представлява логин форма.

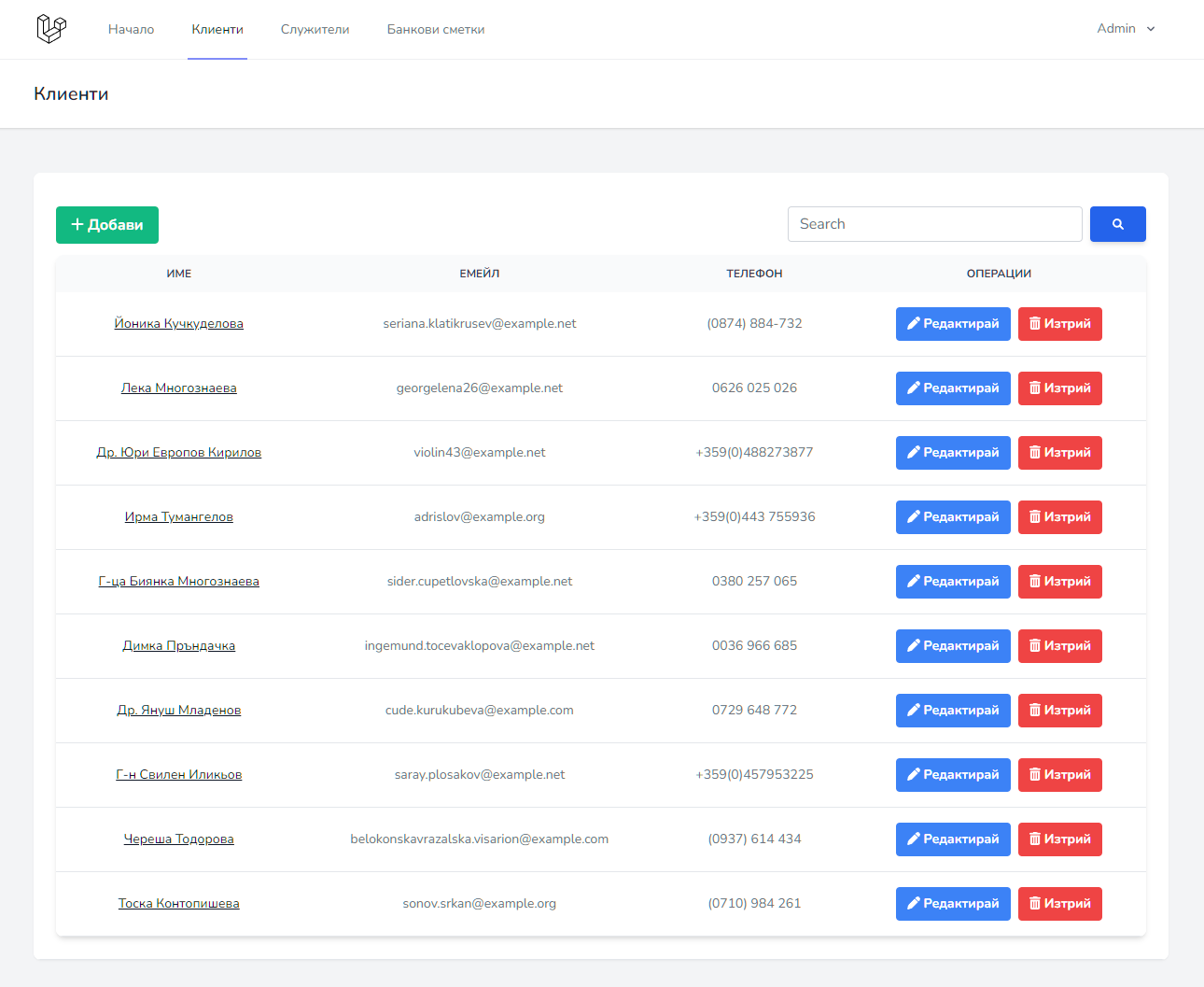
В случай че потребителят няма регистрация, той може да се регистрира натискайки *Not registered yet?*, което му отваря формата за регистрация.



Картина, която съдържа маса

Описанието е генерирано автоматичноПри успешно влизане в системата се показва начален екран със съобщение.

При натискането на таб *Клиенти* се визуализират всичките клиенти в системата, с възможност за търсене по всички колони, добавяне, редактиране и изтриване.



Картина, която съдържа текст

Описанието е генерирано автоматичноПри натискане на бутона Добави се отваря форма за добавяне на клиент, като има както front-end, така и back-end валидация на всички полета, дали са попълнени с коректните типове данни.

Картина, която съдържа текст

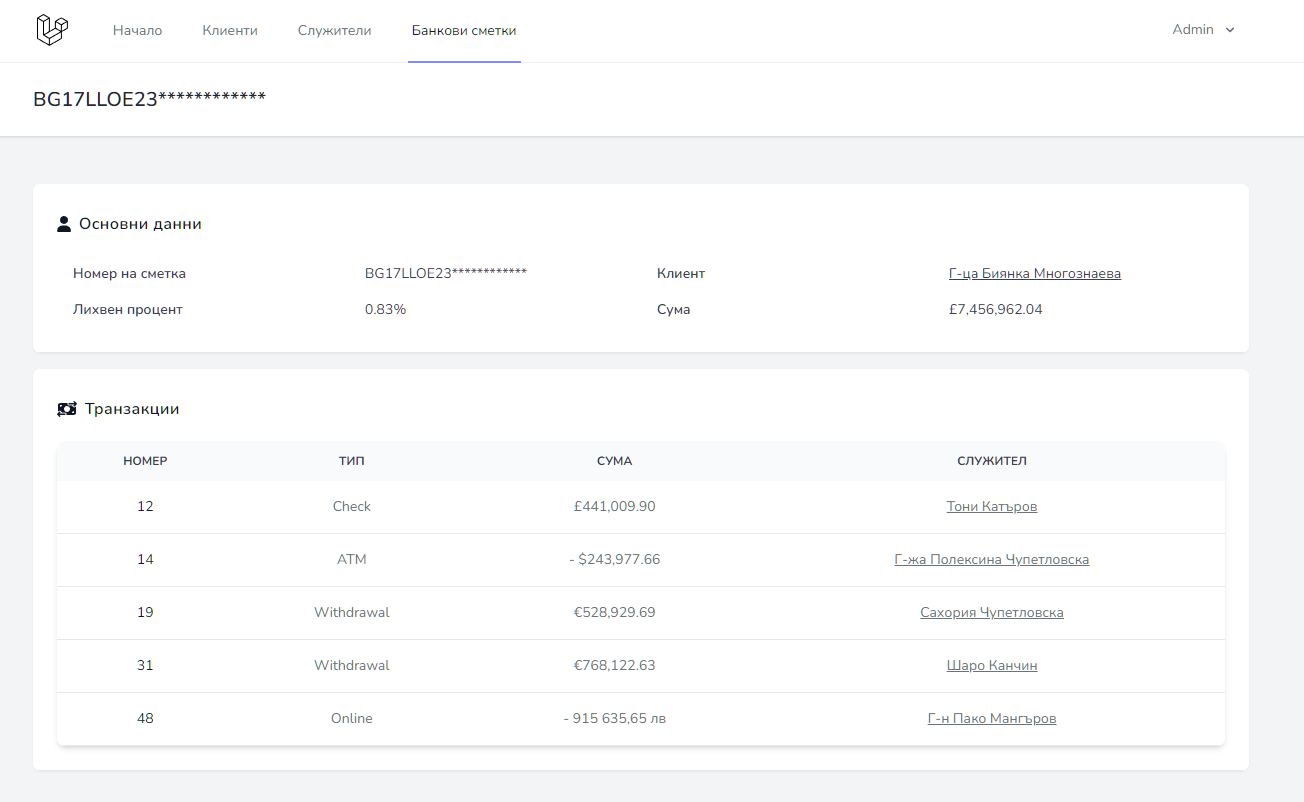
Описанието е генерирано автоматичноПри натискане на бутона Редактирай се отваря същата форма с попълнени данни за съответния клиент.

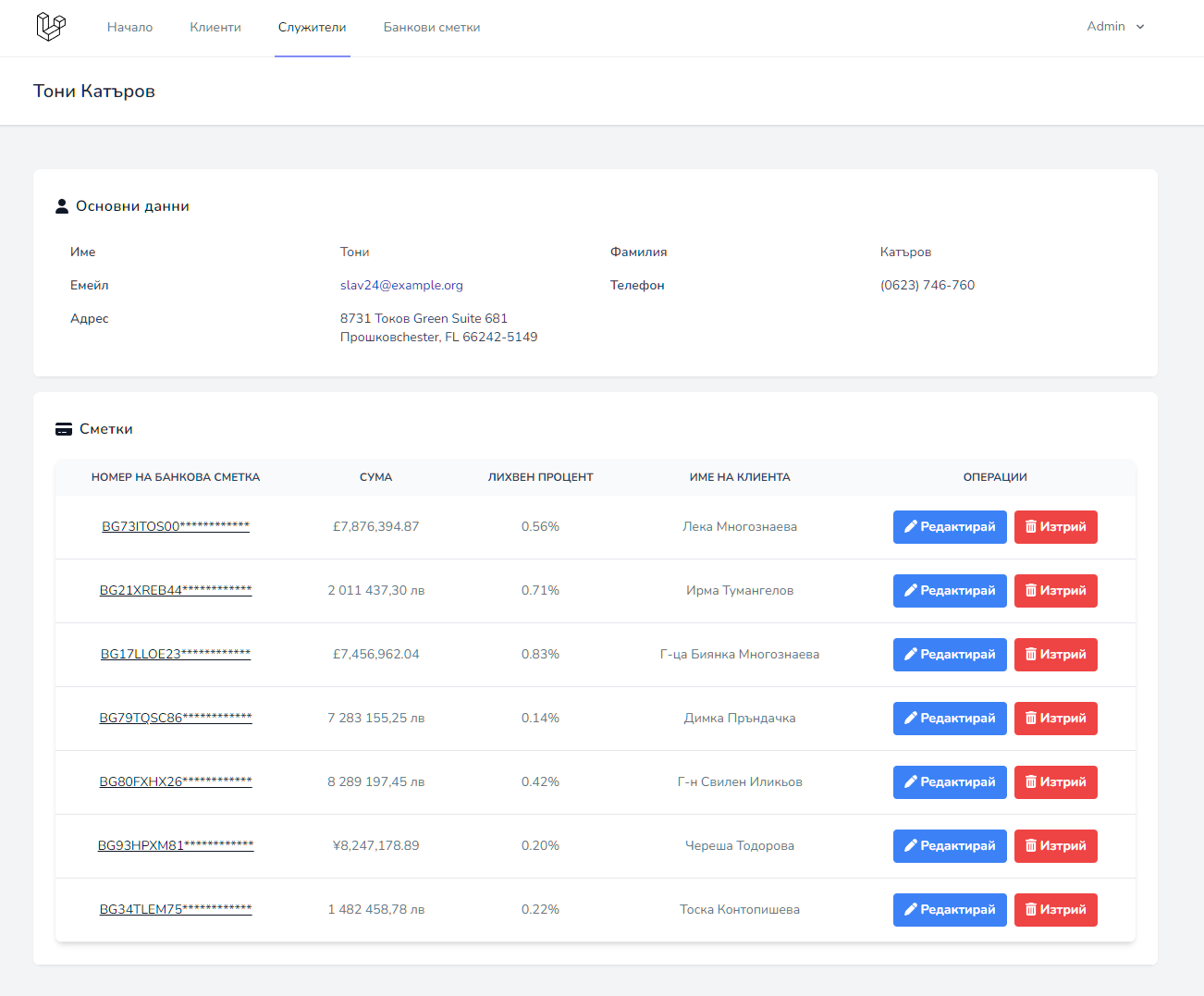
При натискане на бутона Изтрий се изтрива съответния потребител.

Картина, която съдържа текст

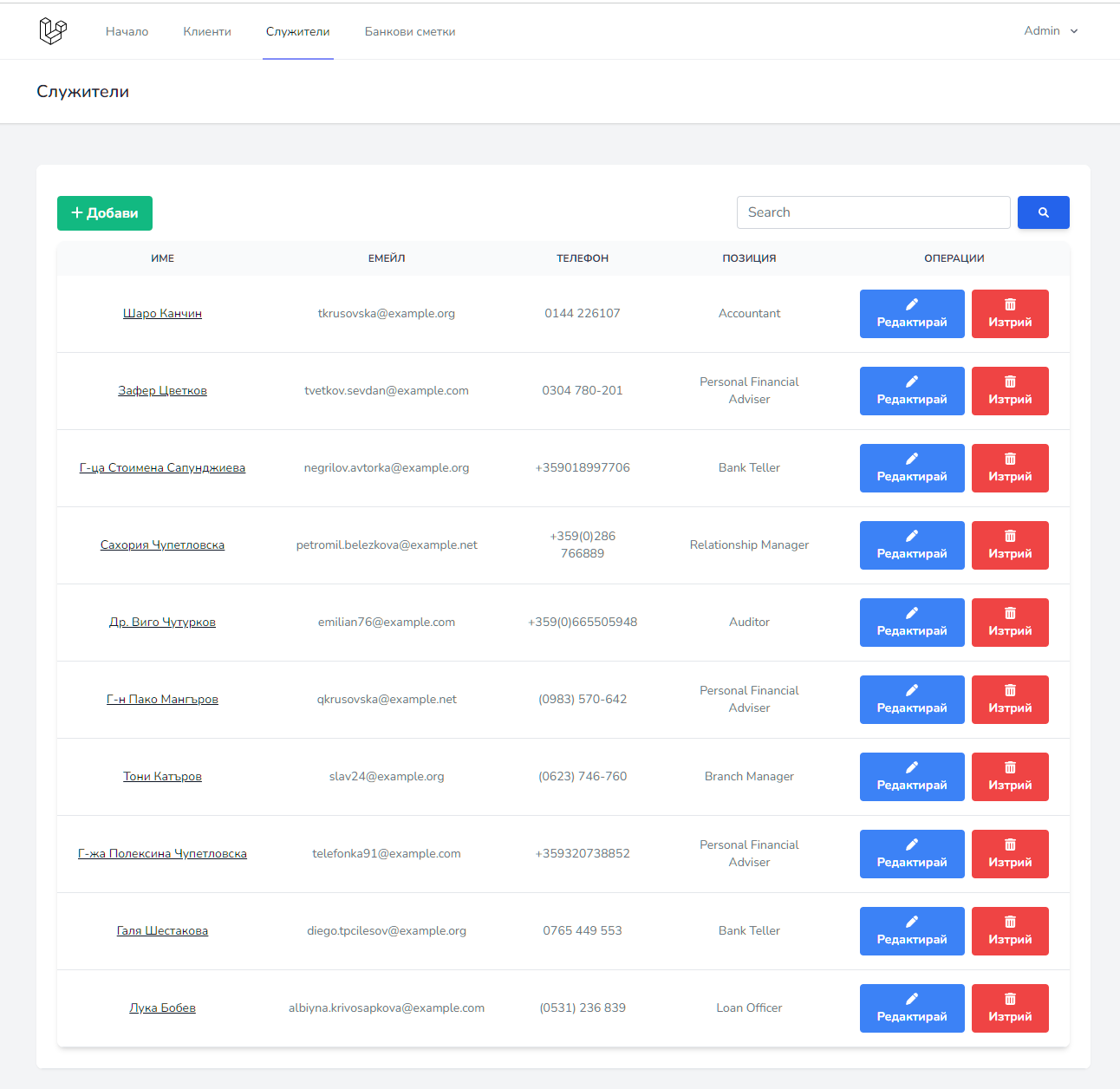
Описанието е генерирано автоматичноПри натискане на името на някой клиент се отваря екран с цялата информация за съответния клиент, както и с таблица за неговите банкови сметки. Те също могат да се редактират и изтриват от този екран.

При натискане на номера на банковата сметка се отваря екран с всички данни за нея, както и таблица със всички транзакции направени по нея.



При натискането на името на клиента, се връщаме обратно към страницата с информацията за него, а при натискането на името на служителя се отваря страница с информация за него и таблица с банковите сметки, в които е правил транзакции.

При натискане са номера на банковата сметка отново ни се показва страницата с нейната информация и транзакции.

При натискане на таб Служители ни се показва таблица с всички служители в системата, с възможност за търсене по всички колони, добавяне, редактиране и изтриване.

При натискане на бутона Добави се отваря форма за добавяне на служител, като има както front-end, така и back-end валидация на всички полета, дали са попълнени с коректните типове данни.

Картина, която съдържа текст

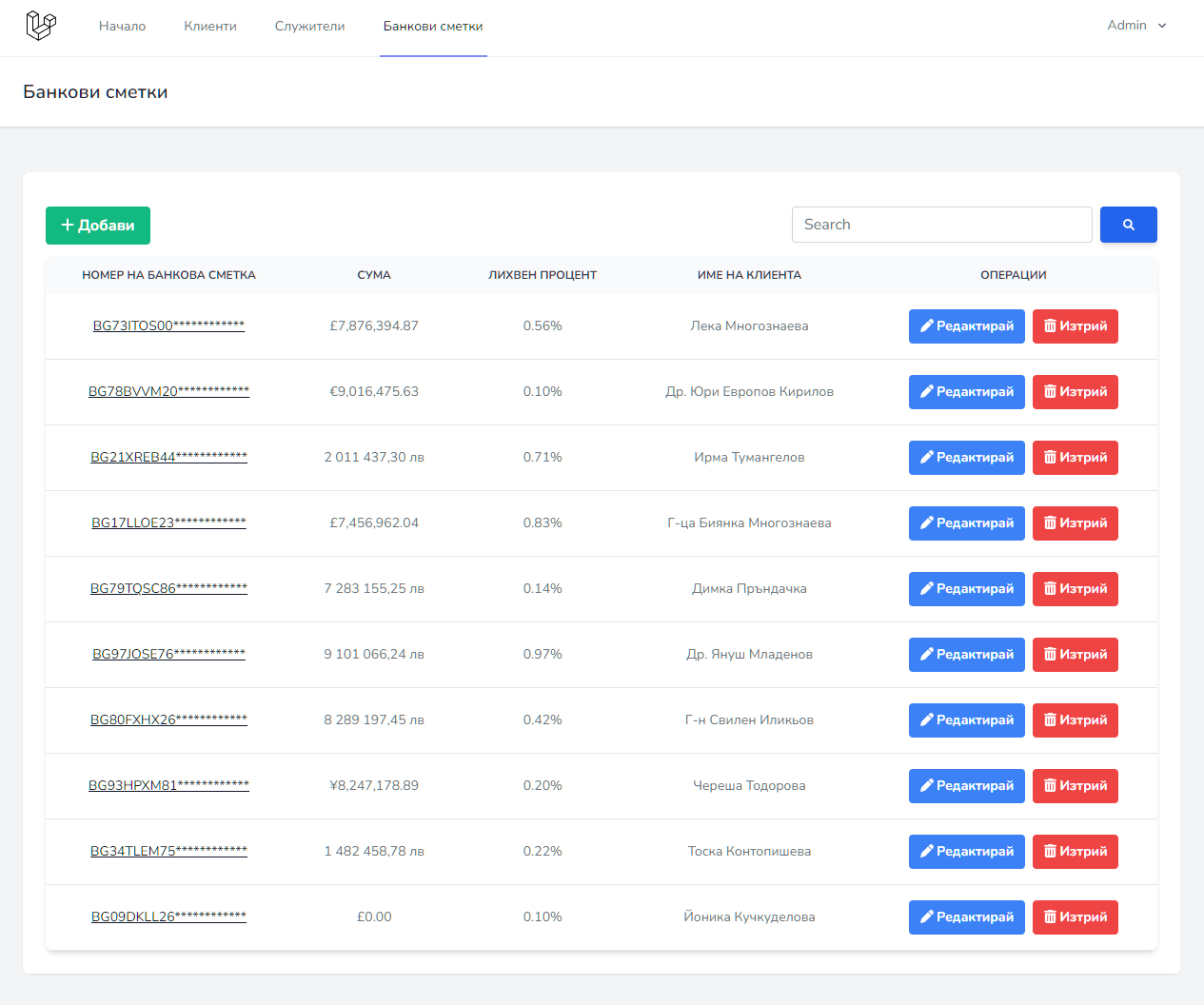
Описанието е генерирано автоматично

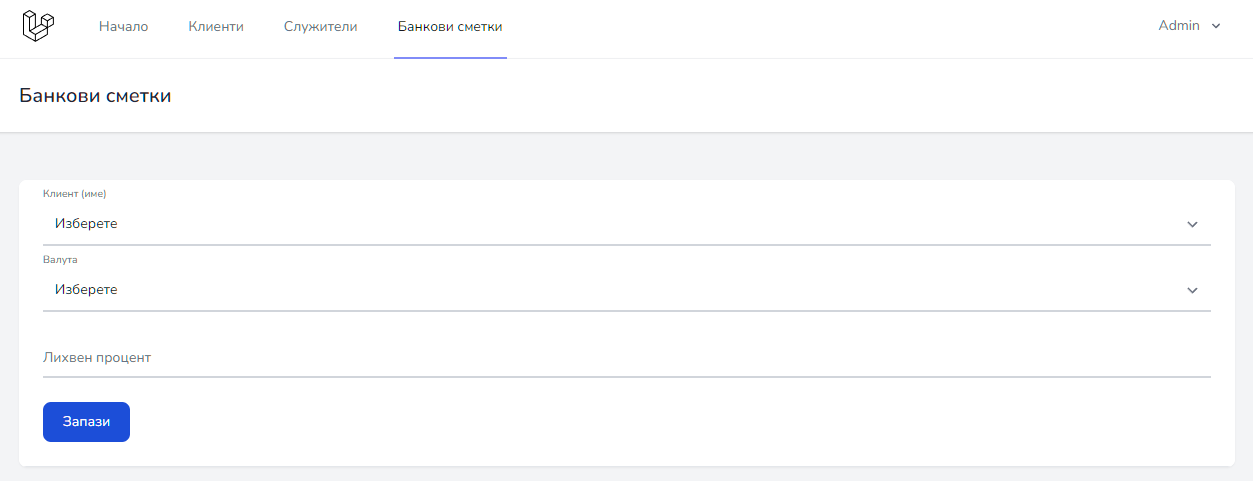
Картина, която съдържа текст

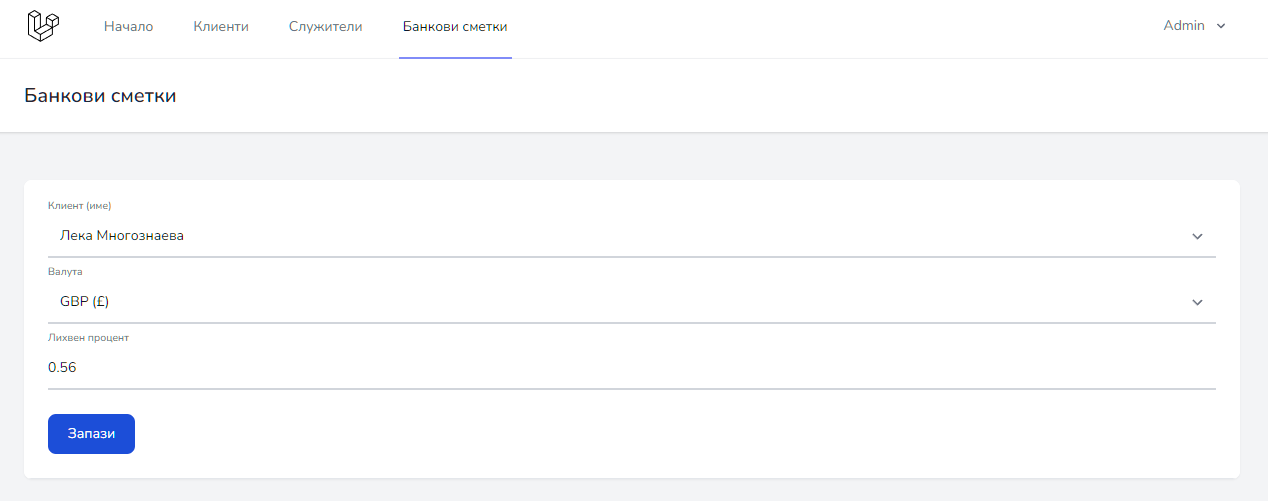
Описанието е генерирано автоматичноПри натискане на бутона Редактирай се отваря същата форма с вече попълнени данните на съответния служител.

При натискане на бутона Изтрий се изтрива съответния служител.

При натискане на името на служителя се показва страницата с данни за него и банковите сметки, в които е правил транзакции, която е спомената по-горе.

При натискането таб Банкови сметки се визуализира таблица с всички банкови сметки в системата с възможност за търсене по всички колони, добавяне, редактиране и изтриване.

При натискане на бутона Добави се отваря форма за добавяне на банкова сметка, като трябва да се избере вече съществуващ клиент, като има както front-end, така и back-end валидация на всички полета, дали са попълнени с коректните типове данни.

При натискане на бутона Редактирай се отваря същата форма с вече попълнени данните за дадената банкова сметка.

При натискане на бутона Изтрий се изтрива дадената банкова сметка.

При натискане на номера на банкова сметка се отваря страница с информация за нея и транзакциите по нея, която е спомената по-горе.

Линк към github repo: <https://github.com/mish0501/uni-db/tree/year-3>