Card Management System Documentation

Project for Spring 5 Course

Пламена Петкова, ФН: 62123

Ради Сергеев, ФН: 62124

# Съдържание

[Описание на проекта](#_745nrp4beera)

[Архитектура](#_ii4lsrqskl3x)

[Използвани технологии](#_y9qu083e0hvf)

[Frontend](#_29mst8n6lt0)

[Backend](#_vwq2trg74fv3)

[Случаи на употреба и роли в системата](#_493jyx2btc3f)

[Роли:](#_erpie7uhneg2)

[Случаи на упортеба:](#_gwkcehy9rtre)

[Data Model](#_9xwb5ucrkg4r)

[ER Diagram](#_atdy98ukdydu)

[Описание](#_bvmnprpqos9q)

[Потребители](#_2mz1yuo9e11s)

[Заявки за промяна на права](#_h68abdbblv7m)

[Карти](#_pwritjpwuk13)

[Заявки](#_hszc47z9pq95)

[Симетрични Ключове](#_a077ohlaa9rv)

[Бин Рейнджове](#_bg5bmc5koqjx)

[Бин](#_7bxhaxh5s7f1)

[Front End](#_85b9t7anrem)

[Back End](#_rr7u6a40lqza)

[API Resource(Backend)](#_p6lpf03091gy)

[Бъдещо Развитие](#_q9mhzud1o4br)

# Описание на проекта

Card Management System е специализирана система за управление на банкови карти, предназначена за вътрешно ползване.

Платформата предоставя възможност за създаване и изтриване на бин рейнджове, бинове и ключове. Възможност за създаване на заявки за производство на банкови карти, както и управление на вече създадени банкови карти.Платформата също предоставя възможност за управление на потребителите, които използват системата.

# Архитектура

Приложението е с MVC архитектура.

# Използвани технологии

### Frontend

Thymeleaf за визуализация.

Thymeleaf extras за филтриране визуализацията на база роля на потребителя.

Bootstrap CSS за по-представителни странички.

### Backend

MySQL база данни.

Tomcat сървър.

Spring MVC за обработка на заявки.

Spring Data JPA за достъп и управление на базата данни.

Spring Security за управление на ролите и достъпа до ресурси базиран на ролите.

# Случаи на употреба и роли в системата

### Роли:

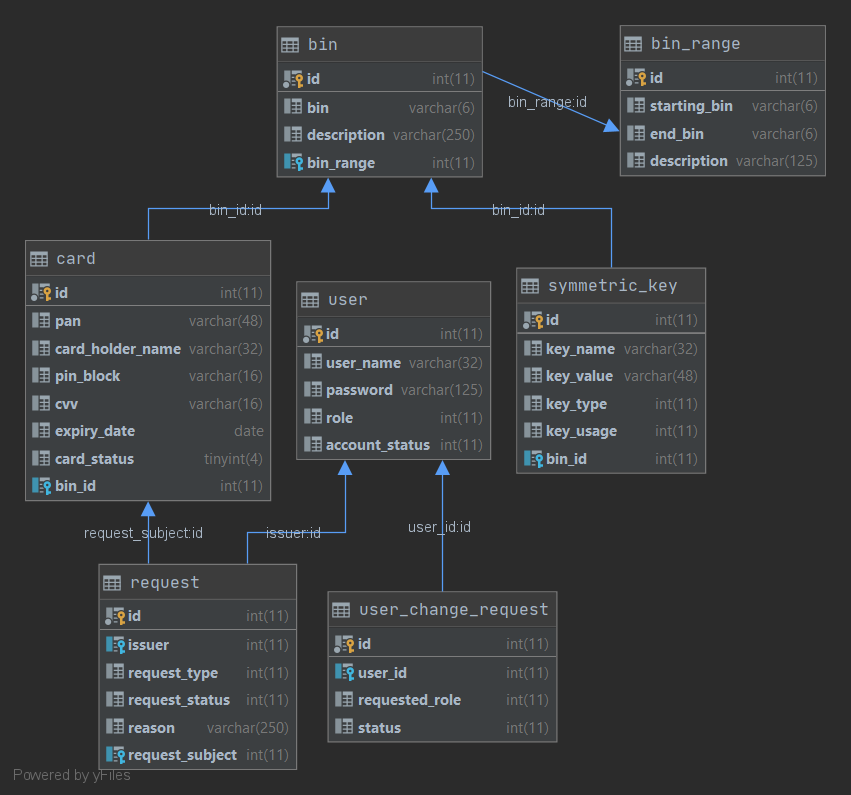
* Анонимен Потребител - няма достъп до системата, може да да направи заявка за регистрация.
* Регистриран потребител - потребител с одобрена заявка за регистрация или регистриран от Администратор.
* Администратор - одобрява заявки за регистрация, създава потребители, може да променя и изтрива потребителски данни.
* Card Production Потребител - създава заявки за нови карти и заявки за деактивация.
* Management Потребите - управялва статуси на карти, зарежда нови ключове, управлява бин-ове и бин рейндж-ове.
* Audit Потребител - разглежда заявки по карти.
* Authorization Потребител - проверява валидността на карта и нейният пин/cvv

### Случаи на упортеба

|  |  |  |
| --- | --- | --- |
| Случаи на употреба | | |
| **No.** | **Описание** | **Потребител** |
| **1.1** | Вход | Регистриран потребител |
| **1.2** | Регистрация | Анонимент Потребител |
| **1.3** | Одобряване на заявка за регистрация | Администратор |
| **1.4** | Управление на потребители | Администратор |
| **1.5** | Управление на заявки за карти | Card Production Потребител |
| **1.6** | Управление на карти | Management Потребител |
| **1.7** | Управление на ключове | Management Потребител |
| **1.8** | Управление на бин рейнджове | Management Потребител |
| **1.9** | Управление на бинове | Management Потребител |
| **1.10** | Преглед на заявки по карти | Audit Потрбител |
| **1.11** | Авторизация на транзакции | Authorization System Потребител |

# Data Model

## ER Diagram



## Описание

### Потребители

|  |  |  |
| --- | --- | --- |
| **Име** | **Описание** | **Тип** |
| id | Еднозначен идентификатор | Int auto increment |
| user\_name | Потребителско име | varchar(32) |
| password | Хеширана парола | varchar(125) |
| role | Потребителск роля | int |
| account\_status | Статус на потребител | int |

### Заявки за промяна на права

|  |  |  |
| --- | --- | --- |
| **Име** | **Описание** | **Тип** |
| id | Еднозначен идентификатор | Int auto increment |
| user\_id | Референция към потребител | int |
| requested\_role | Нова Роля | int |
| status | Статус на заявката | int |

### Карти

|  |  |  |
| --- | --- | --- |
| **Име** | **Описание** | **Тип** |
| id | Еднозначен идентификатор | Int auto increment |
| pan | Криптиран пан | varchar(48) |
| card\_holder\_name | Име на Картодържателя | varchar(32) |
| pin\_block | Криптиран пин блок в iso 0 формат | varchar(16) |
| cvv | Криптирано cvv | varchar(16) |
| expiry\_date | Крайн дата на валидност | date |
| card\_status | Статус на картата | int |
| bin\_id | Референция към bin | int |

### Заявки

|  |  |  |
| --- | --- | --- |
| **Име** | **Описание** | **Тип** |
| id | Еднозначен идентификатор | Int auto increment |
| issuer | Потребител създал заявката  Референция към потребител | int |
| request\_type | Вид Заявка | int |
| request\_status | Статус на заявката | int |
| reason | Причина за заявката | varchar(125) |
| request\_subject | Карта свързана със заявката  Референция към Карти | int |

### Симетрични Ключове

|  |  |  |
| --- | --- | --- |
| **Име** | **Описание** | **Тип** |
| id | Еднозначен идентификатор | Int auto increment |
| key\_name | Имен на Ключа | varchar(32) |
| key\_value | Стойност на ключа | varchar(48) |
| key\_type | Тип на ключа | int |
| key\_usage | Предназначение на ключа | int |
| bin\_id | Референция към бин | int |

### Бин Рейнджове

|  |  |  |
| --- | --- | --- |
| **Име** | **Описание** | **Тип** |
| id | Еднозначен идентификатор | Int auto increment |
| starting\_bin | Начален бин | varchar(6) |
| end\_bin | Последен бин | varchar(6) |
| description | Описание | varchar(125) |

### Бин

|  |  |  |
| --- | --- | --- |
| **Име** | **Описание** | **Тип** |
| id | Еднозначен идентификатор | Int auto increment |
| bin | Стойност на бина | varchar(6) |
| description | Описание | varchar(125) |

# Front End

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **Име на Изгледа** | **Описание** | **URI** |
| **1.Home** | Начална страница на приложението. | / |
| **2.Requests** | Показва всички заявки за производство на карти в системата. Има възможност за филтриране по статус на заявката. | */requests* |
| **3.User Registration** | Форма за регистрация, чрез която Анонимен потребител изпраща заявка за регистрация. Тя трябва да бъде одобрена от Администратор. | */register* |
| **4.Login** | Форма за вход на регистриран и активен потребител, като се въвеждат потребителско име и парола. | */login* |
| **5.Edit user details** | Регистриран потребител може да види информация за акаунта си и да заяви промяна на ролята си в системата. | */settings* |
| **6.Add request** | Форми за попълване на заявки за създаване на нови карти и отказ на такива. | */new-request-form, /revoke-request-form* |
| **7.Users** | Показва списък на всички потребители, регистрирани в системата. Администратор може да извършва CRUD операции. | */users* |
| **8.Cards** | Показва информация за всички карти в системата. Всяка карта може да се редактира. | */cards* |
| **9.Keys** | Показва списък на всички ключове. Може да се добави нов ключ или да се редактира съществуващ. | */key-management* |
| **10.Bins** | Показва всички бинове в системата. Дава възможност за създаване на нови бинове и нови Bin range, както и опция да се изтрият. | */bin-management* |

# 

# Back End

|  |  |  |  |
| --- | --- | --- | --- |
| API Resource(Backend) | | | |
| **No.** | **Метод** | **Описание** | **URI** |
| **1** | User | | |
| **1.1** | GET | Return a list of all users | /users |
| **1.2** | GET | Login Form | /users/login |
| **1.3** | POST | Authenticates User | /user/login |
| **1.4** | GET | Registration Form | /users/register |
| **1.5** | POST | Sends Pending User Request | /users/register |
| **1.6** | GET | Register form for administrator | /users/register-admin |
| **1.7** | POST | Registers new User and activates it | /users/register-admin |
| **1.8** | GET | Returns Information for current user | /users/settings |
| **1.9** | POST | Sends Request for change of permissions | /users/settings |
| **1.10** | GET | Admin view information for User | /users/{userId} |
| **1.11** | DELETE | Remove User | /users/{userId} |
| **1.12** | PUT | Update User | /users/{userId} |
| **2** | Card Management and Production | | |
| **2.1** | GET | Get List of All cards | /card-prod/cards |
| **2.2** | POST | Get Filtered List Of Cards | /card-prod/cards/filter |
| **2.3** | GET | Gets Details For Card | /card-pord/card/{id} |
| **2.4** | PUT | Change Card Status | /card-prod/card/{id}/status |
| **2.5** | PUT | Change Card Pin | /card-prod/card/{id}/update-pin |
| **2.6** | GET | New Card Form | /card-prod/request |
| **2.7** | POST | Send New Card Request | /card-prod/request |
| **2.8** | GET | Revoke Card Form | /card-prod/revoke |
| **2.9** | POST | Send Revoke Card Request | /card-prod/revoke |
| **3** | Requests | | |
| **3.1** | GET | Get List Of Requests | /requests |
| **3.2** | POST | Get Filtered Requests | /requests/filter |
| **3.3** | GET | Get Request Details | /requests/{id} |
| **3.4** | PUT | Change Request Status | /requests/{id} |
| **4** | Keys | | |
| **4.1** | GET | Returns List of Keys | /keys |
| **4.2** | GET | Add Key Form | /keys/new |
| **4.3** | POST | Adds New Key | /keys |
| **4.4** | GET | Returns Key Details | /keys/{id} |
| **4.5** | PUT | Update Key Details | /key/{id} |
| **4.6** | DELETE | Delete Key | /key/{id} |
| **5** | Bin Management | | |
| **5.1** | GET | Returns List of Bin Ranges | /bin-management |
| **5.2** | POST | Adds new Bin Range | /bin-management |
| **5.3** | DELETE | Removes Bin Range | /bin-management/{id} |
| **5.4** | GET | Add New Bin Range Form | /bin-management/new |
| **5.5** | GET | Returns Bins In Range | /bin-management/{rangeId}/bins |
| **5.6** | POST | Adds Bin to Bin Range | /bin-management/{rangeId}/bins |
| **5.7** | GET | Add new Bin Form | /bin-management/{rangeId}/bins |
| **5.8** | DELETE | Deletes Bin From Range | /bin-management/{rangeId}/bins/{binId} |
| **6** | Authorize | | |
| **6.1** | GET | Authorization Test Form | /authorize |
| **6.2** | GET | Authorize With Pin | /authorize/pin?panForPin={pan}&pin={pin} |
| **6.3** | GET | Authorize With Cvv | /authorize/cvv?panForCvv={pan}&cvv={cvv} |
| **7** | Miscellaneous | | |
| **7.1** | GET | Get Welcome Page | /greeting |

# Бъдещо Развитие

Добавяне на hardware security module за управление и работа с ключове.