ОГБПОУ «ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Методические рекомендации по выполнению учебной-практической работы №1 учебной практики

УП 03. Моделирование и анализ предметной области

Тема: «Изучение предметной области»

Выполнил:

студент 601 группы

Буянков И.А.

Проверил:

преподаватель

Михайлов М.А.

г. Томск – 2022 г

Вариант 4 – Реализуйте информационную систему «Библиотека». Обязательные компоненты: название книги, автор, жанр книги, место нахождения книги, раздел, нахождение книги на руки.

**Описание предметной области**

**Библиотека** – это информационная система, которая служит учетом книг для библиотекаря, что позволяет вести учет об информации об идентификации книге, месте хранения книги (для оперативного доступа к печатному изданию), учет выдачи книги посетителям заведения.

**Виды деятельности в заведении разделяются:**

1. Ведение актуальной информации по книге
2. Организация хранения книги и учет хранения
3. Выдача книг на руки посетителям

**Для данной системы характерны следующие актеры**

|  |  |
| --- | --- |
| Актер | Описание |
| Библиотекарь | Администратор системы, устанавливающий контроль в системе учета информации о книгах. |
| Читатель | Пользователь системы, который был зарегистрирован как лицо, взявшее книгу на руки |
| БД | Хранилище данных, в которое заносится информация о книгах и читателях |

**Система предусматривает следующие возможности:**

1. Система располагает каталогом книг, в удобном виде для просмотра читателя. В каталоге представлена главная информация для читателя: название книги, автор, жанр книги, место нахождения книги (на руках\в библиотеке), раздел (место книги в библиотеке), фото (при наличии). К нему может обратиться читатель и администратор.
2. Администратор управляет информацией и взаимодействует с БД через панель администратора.
3. Администратор может добавлять, изменять, удалять книги и информацию о них.
4. Администратор ведет учет выдачи книг на руки. Т.е. вносит данные читателя, который пользуется услугами библиотеки

**Для данной системы можно выделить следующие прецеденты:**

|  |  |
| --- | --- |
| Добавление\изменение\удаление книги | Функция администратора. Позволяет администратору управлять информационным потоком о книгах |
| Предоставление информации о книгах в каталоге | Запускается читателем, для ознакомления с информаций о книге и месте ее нахождения |
| Выдача книги на руки\ возврат книги в библиотеку | Установка для книги статуса «на руках» и учет читателя, который взял книгу, а также возврат книги в библиотеку |

**Описание функций системы**

1. Система дает возможности добавления новой книги в системе и вести учет по ее геолокации, а также изменять эту информацию или удалять книгу.
2. Для читателя информация предоставляется в удобном формате в виде каталога книг. В каталоге будет выделена главная информация о книгах, доступной для читателя (исключая лицо, которое взяло книгу на руки), а также инструкция о взаимодействии с учреждением «Библиотека»
3. Все обращения к общей базе данных проходят через систему для того, чтобы избежать повторяющихся книг или неполной информации, как в хранилище, так и в самом каталоге.
4. Существует гибкий контроль в сфере выдачи книг на руки, во избежание не точной информации о геолокации книги в связи нахождением\не нахождением книги в библиотеке.
5. Система не является складом или печатным предприятием, в его функционал входит установление связи книги и читателя. Т.е. книга уже существует как печатное издание и присутствует в БД, что позволяет предоставить читателю информацию об интересующей его книге

**Построение диаграммы вариантов использования (UML)**

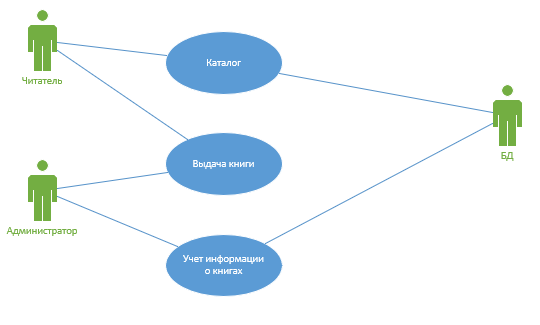


Рис.1 - диаграммы вариантов использования

**Диаграмма классов**

Пример выдачи на руки существующей в системе книги

Класс «Книга»

|  |  |
| --- | --- |
| Параметр | Значение |
| Комментарий | Класс администратора, который ведет учет книги |
| Атрибуты | id = PositiveIntegerField("ID")  title = CharField("Название")  genre = CharField("Жанр")  authors = ForeignKey(Author)  location = ForeignKey(Location)  image = ImageField("Изображение)  existence = BooleanField("Наличие")  reader = ForeignKey(Reader)  date = DateField("Дата взятия книги") |
| Операции | create() – создать книги  edit() – редактировать данные о книге |

Класс «Читатель»

|  |  |
| --- | --- |
| Параметр | Значение |
| Комментарий | Класс читатель, который взял книгу «на руки» |
| Атрибуты | id = PositiveIntegerField("ID")  first\_name = CharField("Имя)  last\_name = CharField("Фамилия")  email = models.EmailField() |
| Операции | create() – создать читателя  edit() – редактировать данные о читателе |

**Конечная диаграмма классов**

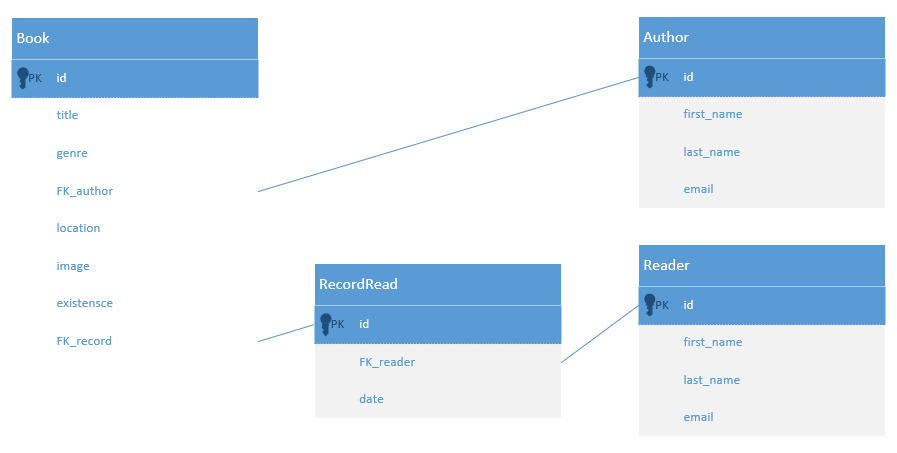


Рис.2 – диаграмма классов

**Диаграмма деятельности**

Пример формирования каталога - списка книг

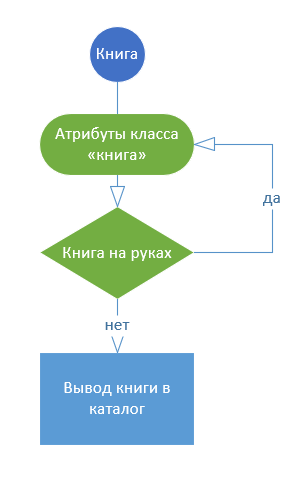


Рис.3 – диаграмма деятельности «каталог»

**Диаграмма последовательности**

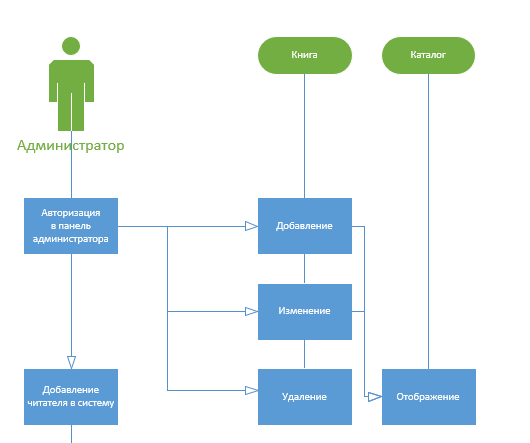


Рис.4 – диаграмма последовательности «администратор»

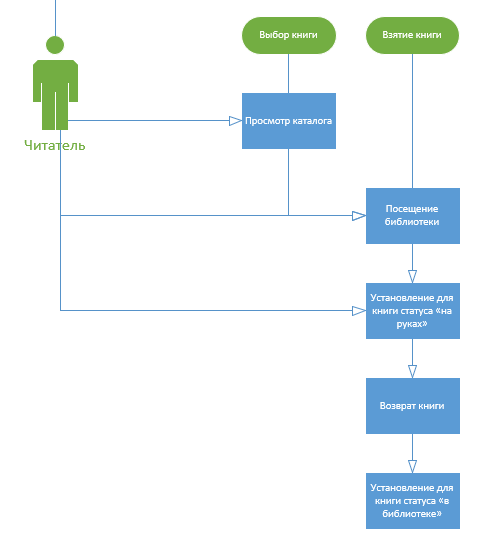


Рис.5 – диаграмма последовательности «читатель»

**Диаграмма состояния**

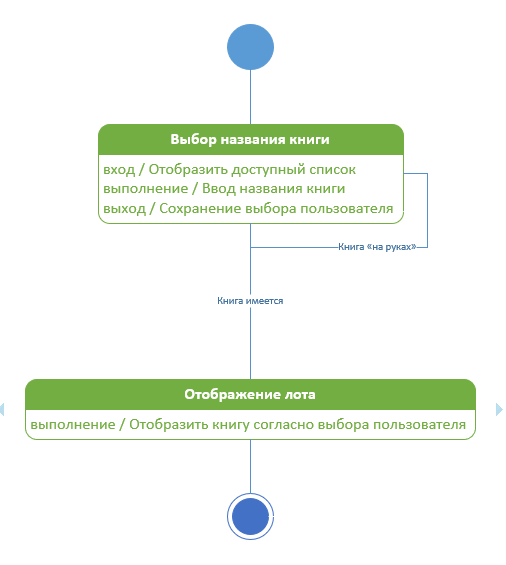
****

Рис.6 – диаграмма состояния «книга»

**Структурная схема базы данных**

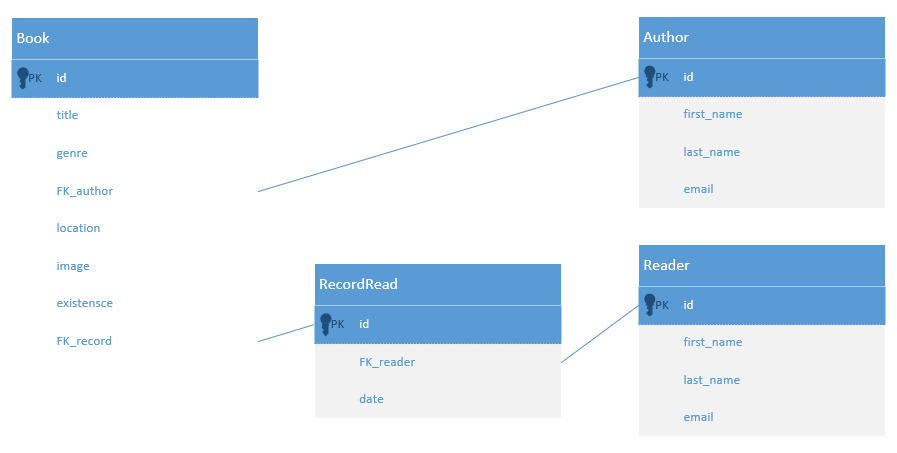


Рис.7 – структурная схема БД

Таблица 1- Книга

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| id | int | Идентификатор книги в системе |
| title | Text-15 | Название книги |
| genre | Text-15 | Жанр книги |
| FK\_author | int | Внешний идентификатор автора |
| location | Text-50 | Место хранения в библиотеке |
| image | image | Фото книги |
| existence | Bool | Наличие книги |
| FK\_record | int | Внешний идентификатор записи |

Таблица 2- Запись книги на руки

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| id | int | Идентификатор записи в системе |
| FK\_reader | int | Внешний идентификатор читателя, взявшего книгу (если статус на руках) |
| date | Date | Дата взятия книги |

Таблица 3- Автор

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| id | int | Идентификатор автора в системе |
| first\_name | Text-15 | Имя |
| last\_name | Text-15 | Фамилия |
| email | Email | Почта |

Таблица 4- Читатель

|  |  |  |
| --- | --- | --- |
| Название поля | Тип данных | Описание |
| id | int | Идентификатор читателя в системе |
| first\_name | Text-15 | Имя |
| last\_name | Text-15 | Фамилия |
| email | Email | Почта |

**Разработка блок-схемы «предоставление информации о книге»**

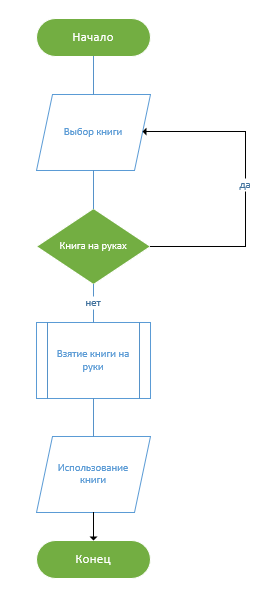


Рис.8 – блок-схема «книга-каталог»

**Вывод:**

В процессе выполнения данной расчётно-графической работы была смоделирована система «Библиотека». Были разработаны прецеденты, диаграмма вариантов использования, диаграммы деятельности, диаграмма классов, диаграммы состояний, диаграмма последовательности системы, структурную схему базы данных и блок-схемы.

**Контрольные вопросы:**

1. **Что такое диаграмма деятельности**
   1. Диаграмма деятельности представляет, по существу, обычную блок-схему. На ней показываются деятельности – шаги в выполнении процесса, изображаемые в виде прямоугольников с сопряженными дугами горизонтальными сторонами и переходы между ними, показываемые стрелками. Предусмотрена возможность ветвления, изображаемая в виде ромба. На этих диаграммах можно показать распараллеливание процесса на подпроцессы и слияние подпроцессов. Для обозначения этих действий используются жирные горизонтальные или вертикальные линии. Все элементы могут быть поименованы
2. **Зачем нужны диаграммы состояния**
   1. Диаграммы состояний предназначены для представления жизненного цикла объекта в виде конечного автомата. Каждое состояние – это период жизни объекта, когда он удовлетворяет определенным условиям. Некоторое событие может привести переходу объекта в другое состояние. При переходе может выполняться действие, предписанное данному переходу.
3. **Что такое стереотипы классов**
   1. классов. Стереотип класса – это элемент расширения словаря UML, который обозначает отличительные особенности в использовании класса. Стереотип имеет название, которое задается в виде текстовой строки. При изображении класса на диаграмме стереотип показывается в верхней части класса в двойных угловых скобках. Стереотип используется для обозначения классов-сущностей