**3) Кратенько технически описать каждый из компонентов. То есть ОС терминала, язык приложения терминала, ОС сервера, язык сервера, фреймворки всякие, использованные технологии и прочее.**

Для ОС терминала будем использовать Linux, язык программирование – Python, т.к. на нем будет удобно работать с БД и со сканированием штрих-кодов на абонементе пользователя. Для разработки приложения можно использовать фреймворк Flask, чтобы создать веб-приложение, с помощью которого можно обеспечить доступ к базе данных и реализовать необходимые операции, такие как сканирование штрих-кода, внесение данных в базу данных и удаление записей. Для сканирования штрих-кодов можно использовать библиотеку ZBar, которая предоставляет возможность сканировать штрих-коды с помощью подключенного сканера. Для работы с базой данных можно использовать PostgreSQL, который обеспечивает удобный и гибкий способ работы с базами данных.

**UI и функционал терминала:** На терминале должны быть расположены элементы управления, которые позволят пользователям вводить свой уникальный ID или сканировать его, чтобы автоматически загрузить свою учетную запись. Также должны быть кнопки для подтверждения ввода и начала процесса аренды книги. Он будет состоять из следующих частей:

1) Компьютер: компьютер может использоваться для запуска программного обеспечения, управления периферийными устройствами, такими как сканеры штрих-кодов, и обеспечения связи с сервером базы данных.

2) Сенсорный экран: сенсорный экран может использоваться для удобного ввода данных и управления терминалом. Для этого можно использовать сенсорный дисплей, который поддерживает мультитач и может работать с различными видами жестов.

3) Штрих-код сканер: сканер штрих-кодов может использоваться для быстрого и точного чтения уникальных идентификаторов пользователей и книг как в маке.

4) Принтер: штука которая печатает чек со всякими данными, в том числе с номером очереди, если такая имеется на данный момент в библиотеке

5) Веб-приложение: веб-приложение может использоваться для реализации логики работы терминала, управления базой данных, авторизации пользователей и регистрации аренды книг.

6) База данных: база данных может использоваться для хранения информации о пользователях, книгах, арендных сроках, задолженностях и другой важной информации. Для этого можно использовать реляционные базы данных, такие как PostgreSQL или MySQL, или NoSQL базы данных, такие как MongoDB. Как уже выше сказано будем использоваться SQLAlchemy для питона.

7) В дополнение к питону можно использовать JS и html\css для web-клиента.

**Сервер**: ну также можно юзать питон. ОС также Linux (ubuntu server/CentOC).

Приложение для библиотекаря:

Будем использовать Python, так как он предоставляет множество библиотек для работы с базами данных и имеет простой синтаксис.

Для связи с общей базой данных можно использовать PostgreSQL, которая позволяет работать с разными СУБД через единый интерфейс.

Для создания пользовательского интерфейса приложения можно использовать фреймворк PyQt или Tkinter, которые предоставляют удобные инструменты для создания графических интерфейсов.

Для обработки оплаты через кассу можно использовать специализированные библиотеки для работы с фискальными регистраторами (библиотека fptrlib - библиотека на языке Python для работы с фискальными регистраторами через интерфейс ФПТР) или сами API кассовых аппаратов.

Само приложения должно состоять из следующих основных частей:

1) Библиотекарь может вводить id пользователь, чтобы проверить его задолженности

2) После проверки задолженностей библиотекарь может выбрать определенную книгу, которую выберет пользователь и ввести в БД информацию о том, что книга взята

3) Также библиотекарь может из этого приложения завершать аренду, отправлять на терминал оплаты сумму для списания и, если у пользователя будет оплата наличными, то произвести это через кассу.

**8) Чётко описать в чём проблема "того как есть сейчас" и нахрена мы вводим этот терминал, а также как это решить. То есть нам нужен текст о том:**

**1)В чём проблема не иметь терминал**

**2)Что изменится при вводе терминала**

**3)Пример проблемы, что есть сейчас без терминала**

**4)Список того, что мы затронем (билет, библиотекарь, местный сервер)**

**5) ну и накидать таски как для кликКапа - "сделать анализ", "купить серверное оборудование" и т.д**

1) Проблема не иметь терминала заключается в том, что все операции по выдаче книг и возврату книг происходят только через библиотекаря, что занимает много времени и увеличивает время ожидания для пользователей. Более того, возможна ошибка при вводе данных вручную, что может привести к неправильному списанию книг или задержке при их возврате. Также при большом потоке посетителей нужно больше библиотекарей, а, следовательно, больше трат на зарплаты и содержание кадров.

2) При вводе терминала будет происходить автоматическая проверка задолженностей у пользователей и резервирование книг в режиме реального времени. Пользователи смогут самостоятельно брать книги и сдавать их в любое время, без необходимости общения с библиотекарем. Это значительно сократит время ожидания для пользователей и увеличит эффективность работы библиотеки. Также имея собственную базу данных и веб-приложение можно будет реализовать проверку или доп. продление аренды через сайт.

3) Одной из проблем, которые есть сейчас без терминала, является задержка в выдаче книг из-за очереди на обслуживание у библиотекаря. Также возможны ошибки при вводе данных вручную, которые могут привести к неправильному списанию книг или задержке при их возврате.

4) При введении терминала нам необходимо будет затронуть следующие аспекты:

1. Разработка софта для терминала и сервера
2. Анализ и выбор подходящей аппаратной платформы для терминала и сервера
3. Приобретение необходимого оборудования и серверного оборудования
4. Проведение настройки и интеграции сервера и терминала
5. Обучение сотрудников работы с новым оборудованием
6. Закупить компьютеры для библиотекарей (можно ноутбуки)
7. В дальнешем-поддержание всей аппаратуры в рабочем состоянии

Таким образом, введение терминала позволит библиотеке повысить эффективность работы, снизить время ожидания пользователей и уменьшить вероятность ошибок при выдаче и возврате книг. Необходимыми затратами на реализацию проекта будут являться анализ, приобретение оборудования и настройка, но это будет оправдано в долгосрочной перспективе. Следовательно, инвесторы довольны, а мы покупаем Москвич 6 и гоняем по ночному Краснодару.