**Аналитическая записка.**

**Централизованный сервис для учёта книг библиотеки.**

**Целевая аудитория.**

Целевой аудиторией проекта являются работники библиотеки, а также те, кто ведёт персональный учёт книг.

**Назначение проекта**

Проект позволяет вести учёт книг, выдачи, людей и статистики с различных устройств (или ПО) и решает проблему синхронизации данных.

Сервис позволяет использовать меньшие ресурсы, так как направлен на ведение учёта, не предоставляя функционал электронной библиотеки для читателей (необходимость в электронной библиотеке может отсутствовать).

**Основной функционал**

* Возможность ведения учёта книг, людей и выдачи,
* Возможность просмотра статистики выдачи книг (по самым популярным книгам),
* Синхронизованная работа с различных устройств.

**Описание взаимодействия с потенциальным пользователем**

Пользователь может взаимодействовать с сервером с помощью графического интерфейса программы. На рисунке 1 представлена диаграмма прецедентов описывающая взаимодействие с потенциальным пользователем.

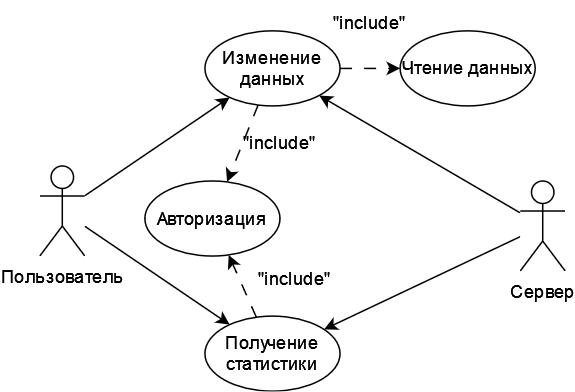


Рисунок 1 — Диаграмма прецедентов

**Возможные аналоги решения**

1. «1С:Библиотека»

Позволяет автоматизировать рабочие процессы библиотеки, в зависимости от её назначения, типа, состава фондов.

Особенности:

* Может быть автоматизирована деятельность библиотек разных назначений и типов;
* Обслуживание читателей;
* Виртуальный кабинет читателя.

1. «OPAC-Global»

Автоматизированная библиотечная информационная система, основанная на облачных технологиях.

Особенности:

* Централизованный сервер с доступом через браузер;
* Саморегистрация читателей.

1. «Либра»

Особенности:

* Поддерживает работу со считывателями штрих-кода;
* Возможность создания электронной библиотеки;

Представляет собой программный комплекс, предназначенный для ведения учёта каталога книг и периодических изданий, ведения картотеки читателей, регистрации выдачи и возврата литературы (абонемент), поиска по каталогу и многое другое.

**Предполагаемые к использованию технологии и модели**

* Микросервисы Spring Cloud для масштабируемости и отказоустойчивости
* Spring Security для реализации авторизованного доступа
* Кэширование с помощью Hazelcast для повышения производительности REST API.