**Требования**

**Функциональные:**

1. Создание аккаунта
2. Просмотр личный кабинета
3. Просмотр расписания
4. Выборка моих курсов
5. Покупка курсов
6. Уведомления
7. Чаты с преподавателями, менторами, администрацией
8. Общие настройки приложение: звуки, уведомления и т.п.
9. Просмотр и скачивание сертификатов за пройденные курсы
10. Курсы разделены на текущие и прошедшие
11. Название текущего курса
12. Каталог с кнопками: лк, мои курсы, мое портфолио, услуги, настройки
13. В расписании отображались дедлайны и другие учебные мероприятия
14. Успеваемость в виде таблицы по выполненным практикам

**Нефункциональные:**

1. Система должна загружать приложение менее чем за 5 секунд даже при одновременном использовании 1000 пользователями.
2. Система должна выдавать не более 5 ошибочных запросов .
3. Система должна иметь возможность увеличивать или уменьшать масштаб по мере необходимости.
4. Система должна быть доступна в любое время.
5. Система должна быть проста в обслуживании и обновлении.
6. Система должна работать на мобильных устройствах (Android версия).
7. Доступ к аккаунтам должен быть только у самих пользователей.
8. Система должна быть простой в использовании и понятной.
9. Система не должна противоречить закону о персональных данных.

**User story**

**Ученик**

Я, как ученик, хочу видеть свое расписание учебных мероприятий по текущему курсу.

Я, как ученик, хотел бы в приложении иметь возможность купить курс.

Я, как ученик, хотел бы иметь возможность связываться со своими преподавателями через приложение.

Я, как ученик, хотел бы следить за своей успеваемостью.

Я, как ученик, хотел бы иметь доступ к материалам всех моих курсов: текущих и пройденных.

Я, как ученик, хотел бы иметь возможность просматривать и скачивать сертификаты по своим завершенным курсам.

**Преподаватель**

Я, как преподаватель, хотел бы иметь возможность связываться с учениками своего курса.

Я, как преподаватель, хотел бы смотреть общую статистику успеваемости учеников курса.

Я, как преподаватель, хотел бы иметь возможность оценивать практические задания учеников курса.

Я, как преподаватель, хотел бы иметь доступ к просмотру успеваемости конкретных учеников курса.

Я, как преподаватель, хотел бы вносить изменения в расписание учеников своего курса.

**Менеджер**

**Use Case**

Use Case:

|  |  |
| --- | --- |
| Название | Авторизация |
| Описание | Ученик может авторизироваться в приложении, используя онлайн-систему авторизации. |
| Акторы | Ученик |
| Предусловия | Пользователь первый раз запустил приложение. |
| Основной поток событий | 1. Клиент выбирает город отправления и прибытия.  2. Клиент выбирает дату и время вылета.  3. Система отображает список доступных рейсов.  4. Клиент выбирает нужный ему рейс.  5. Система запрашивает у клиента информацию о пассажирах.  6. Клиент вводит информацию о пассажирах.  7. Система отображает информацию о стоимости билетов.  8. Клиент подтверждает бронирование и оплачивает билеты. |
| Альтернативный поток событий | 1. На шаге 3 система не отображает список доступных рейсов.  2. Клиент изменяет параметры поиска рейсов.  3. Система отображает новый список доступных рейсов.  4. Клиент продолжает процесс бронирования. |
| Постусловия | Клиент успешно забронировал билет на выбранный рейс и получил подтверждение бронирования. |
| Расширенные атрибуты | 1. Если клиент вводит неверные данные, система сообщает об ошибке и предлагает исправить их.  2. Клиент может выбрать места в самолете.  3. Система принимает оплату за билеты. |
| Диаграмма Use Case | Пример use case диаграммы «бронирование рейса» |
| Рекомендации по реализации | 1. Реализовать удобный и интуитивно понятный интерфейс для ввода данных учеником.  2. Отобразить доступные рейсы в удобном для клиента виде (например, сортировать по цене или дате вылета) |