**Введение**

1. Наименование продукта

Продукт называется “Каталог репозиториев студентов”.

2. Назначение и область применения

Продукт создается для облегчения учебного процесса в вузе. Студент сможет найти в списке свой курс и, указав нужный предмет, просматривать повторно лекции в виде презентаций и записей преподавателей. А преподаватели смогут быстро довести информацию до студентов или, к примеру, найти список студентов.

В каталоге будет 2 вида репозиториев, а именно:

1. Учебный. Где будет храниться: список студентов, предметы, лекции.
2. Индивидуальный проект. Здесь можем открыть и посмотреть имя автора работы, группу, используемые языки и библиотеки, область применения проекта. И желающие смогут присоединиться к автору для помощи в разработки проекта.

3. Выбор методологии

В разработке проекта мы будем применять методологию Scrum. Основной акцент использования данной методология делается на качественном контроле процесса разработки. Для представления хода выполнения проекта будут использованы, так называемые, Sprint’ы (это ограниченный по времени процесс, на протяжении которого производится работа и управление продуктом). На выходе каждого Sprint’а должна получаться новая функционирующая версия продукта.

**Общее описание**

1. Функции продукта

Данный продукт должен предоставлять пользователям интерфейс для управления разными Git-репозиториями, как личными, так и учебными на большое количество участников.

Интерфейс должен быть представлен как в виде мобильного приложения под ОС Android, так и web-сайтом.

2. Характеристики пользователя

Виды пользователей и их возможности:

1. Администратор - имеет право назначить пользователя преподавателем с правами, модерировать существующие репозитории.
2. Преподаватель - имеет право создавать и управлять учебными каталогами/репозиториями, добавлять в них студентов, создавать студентам репозитории в своих учебных, создавать и управлять репозиториями для личных проектов, подавать запросы на участие в чужих учебных репозиториях.
3. Студент - имеет право подавать запросы на участие и управлять своей веткой в учебных репозиториях, создавать и управлять репозиториями для личных проектов, подать заявку на статус преподавателя.

Все пользователи имеют право на поиск по тегам и фильтрацию результатов в репозиториях, открытых для них.

3. Ограничения

В системе не могут работать неавторизованные пользователи. Система обязана работать для определенных Git-репозиториев, количество и наименование которых будет определено в ходе проектирования ( их общее количество не должно быть меньше 2).

4. Допущения и зависимости

**Детальные требования**

1. Требования к внешним интерфейсам

Во всех внешних интерфейсах должны быть доступны все функциональные требования для каждого из видов пользователей.

Дизайн согласовывается с Заказчиком и производятся корректировки по мере надобности. Положение основных элементов для разных видов интерфейсов представлено в дополнительном документе.

2. Функциональные требования

Общие возможности пользователей:

1. Авторизация. Происходит через аккаунт системы Google как для сайта, так и для приложения для всех типов пользователей.
2. Редактирование анкеты, включающей в себя поля про личные данные пользователя, его статистику и список репозиториев как частных, так и учебных.
3. Просмотр анкетных данных других пользователей.
4. Просмотр репозиториев, к которым есть доступ.Также доступны поиск, статистика, фильтрация и сортировка.
5. Просмотр новостей, которые представляют из себя ленту событий, связанных с репозиториями пользователя.
6. Создание репозиториев для личных проектов.
7. Создание репозиториев в учебных проектах, если пользователь состоит в нем как участник.

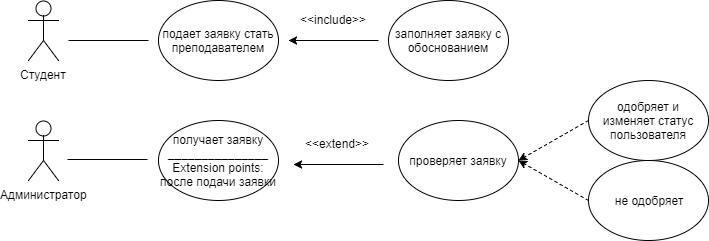
Возможности преподавателя:

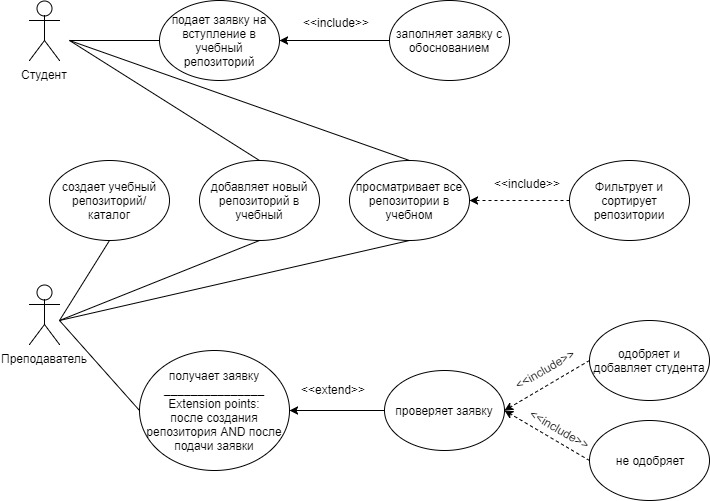
1. Создавать учебные репозитории.
2. Добавлять студентов в свои учебные репозитории, создавать им ветки.

Возможности администратора:

1. Менять статусы пользователей.
2. Модерировать пользователей.

На рисунках ниже в UML-диаграмме прецедентов наглядно представлено развитие событий для перехода пользователя из статуса студента в статус преподавателя и действий с учебным репозиторием.





4. Нефункциональные требования

Требования к квалификации:

Команда разработчиков должна состоять минимум из 4 человек, включающих в себя специалиста по web-разработке, специалиста по разработке мобильных приложений, технического писателя, тестировщика и менеджера проекта.

Требования по отчетности в течение процесса разработки:

Личные встречи с заказчиком с демонстрацией текущей версии продукта минимум раз в две недели.

Письменные отчеты о ходе процесса на электронную почту минимум раз в неделю.

**Программные продукты и технологии**

**Порядок контроля и приемки продукта**

**Требования к документированию**