

Постановка задачи приложения

1. Проблема

В настоящее время в университетах нет приложений, предназначенных для студентов. Студенты хотят участвовать в университетских курсах и видеть объявления на этих курсах, но эта информация размещена на нескольких веб-сайтах, которые трудно найти. Они хотят найти сокурсников с одинаковыми интересами и поделиться своим мнением о курсах и материалах курса, однако пока это невозможно. Они также хотят обсудить экзаменационные вопросы и найти подходящее место, где проходит экзамен, без поиска в Google.

2. Сценарии

Сарвар, студент, изучает информатику в ТМУО. Он изучает бизнес-администрирование в качестве второстепенного предмета и уже привык посещать курсы в корпусе компьютерных наук. Однако курсы делового администрирования расположены в лекционном зале в другом здании в центре Ташкента. Он никогда раньше не бывал в другом здании, поэтому не знает, как найти лекционные залы для своего второстепенного предмета. Он просматривает курсы в каталоге курсов университетского приложения и находит курс “Основы делового администрирования” со временем прохождения курса и расположением лекционного зала на карте.

Во время посещения курса он вступает в контакт с сокурсниками, которые также посещают курс, и читает их комментарии. Ему нравится один комментарий “Отличные упражнения” Лолы, которая также изучает информатику. По фотографии Лолы он помнит, что они познакомились неделю назад у кофейного автомата. Он просит дружбы с Лолой (она может помочь ему сдать выпускной) и добавляет новый комментарий к экзаменационным вопросам с предыдущих экзаменов. Пока он просматривает, Лола получает уведомление о запросе в друзья и принимает его. Сарвар, в свою очередь, получает уведомление о том, что Лола приняла его запрос и теперь просматривает все курсы, которые посещает Лола. Он находит еще один интересный курс “Учет затрат”, который хочет посетить, и сохраняет его в своем списке курсов.

3. Требования

В проекте должны быть учтены следующие функциональные требования (FR) и нефункциональные требования (NFR).

FR1: Поиск доступных курсов: Студент может просмотреть все курсы текущего семестра по своему основному и второстепенному предмету. Он может присоединиться к курсу, который сохраняет его в своем списке курсов. Он также может отказаться от курса.

FR2: Проверьте сведения о курсе: Студент может просмотреть подробную информацию о курсе, такую как время прохождения курса, расположение лекционного зала на карте и других участников курса, включая их имя и фотографию.

FR3: Обновить профиль: Студент может обновить настройки своего профиля и фотографию профиля. Он также может изменить настройки уведомлений.

FR4: Добавление комментариев: Студент может добавлять комментарии к курсу и, таким образом, начинать обсуждение. Другие могут поставить лайк этому комментарию и написать последующие комментарии.

FR5: Запрос на дружбу: Студент может запросить дружбу с другим студентом, который затем получит уведомление о запросе. Второй студент может принять и отклонить дружбу, о чем оба уведомляют первого студента.

FR6: Просмотр курсов друзей: Студент может просматривать курсы своих друзей.

FR7: Просмотр объявлений: Студент может просматривать объявления о курсах и комментировать / ставить им лайки.

FR8: Публиковать обновления на временной шкале: Учащийся может публиковать обновления на своей временной шкале. Друзья получают уведомления об обновлениях и могут комментировать их и ставить лайки. Некоторые обновления публикуются автоматически, например, сохранение курса в списке курсов или комментирование курса.

FR9: Смотрите календарь курсов: Студент может просматривать все курсы в календаре.

NFR1: Удобство использования: Приложение должно быть интуитивно понятным в использовании, а пользовательский интерфейс должен быть простым для понимания. Все взаимодействия должны быть завершены менее чем за три клика.

NFR2: Соответствие руководящим принципам: Дизайн приложения должен соответствовать рекомендациям по удобству использования для выбранной операционной системы.

NFR3: Целевая платформа: Приложение должно быть разработано на Python.

NFR4: Серверная система: Клиент предоставляет серверную систему с несколькими сервисами, которые необходимо использовать в приложении.

Дополнительные ограничения:

- Система контроля версий должна быть git.
- Документация по исходному коду должна быть в формате HTML.

4. Целевая среда

Приложение должно быть продемонстрировано на Python.

5. Документация

- Документ с анализом требований (RAD - Requirements Analysis Document)
- Проектный документ системы (SDD - System Design Document)
- Исходный код под контролем версий, включая документацию по исходному коду

6. Критерии Приемлемости Клиента

Приложение должно демонстрировать, по крайней мере, следующую функциональность: оно показывает список курсов, к которым может присоединиться студент. Студент может видеть участников курса с их фотографиями и может запросить дружбу с другими студентами. Приложение взаимодействует с серверной системой, предоставленной заказчиком, и соответствует требованиям к удобству использования.