

Заказчик:	НИУ ВШЭ, Международная лаборатория интеллектуальных систем и структурного анализа, Паринов Андрей Андреевич
Название проекта:	Клиент-серверное iOS приложение Ассистента Студента
Исполнители	Зубарева Наталия Дмитриевна, Мостачев Андрей Олегович, Поволоцкий Виктор Александрович, Сальникова Алиса Дмитриевна

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. Вводы системы.

Ввод представляет собой текстовые данные, вносимые пользователем, а также его взаимодействие с графическим пользовательским интерфейсом.

Эти данные, а также жестовое взаимодействие пользователя с программой происходит посредством интерфейса мобильного приложения на платформе IOS. Интерфейс разработан на основе комбинации фреймворков UIKit и SwiftUI и содержит стандартные элементы ввода: изменяемые текстовые поля, выпадающие списки, чекбоксы и тому подобное.

Текстовые данные, которые пользователь вводит:

1. данные для авторизации - логин и пароль в едином личном кабинете НИУ ВШЭ - вводятся через изменяемое текстовое поле;
2. личные данные в аккаунте - имя пользователя, о себе, факультет, курс, группа, контакты - электронная почта, номер телефона, ники в соцсетях - вводятся через изменяемые текстовые поля, выпадающие списки;
3. учебные интересы - текстовые описания дисциплин, технологий, фреймворков - вводятся через выпадающий список.

2. Выводы системы.

Система выводит данные в интерфейсе в виде текста и минимальной численной инфографики. Вывод содержит информацию о пользователе, информацию о проектах, а также информацию о действиях в системе: каждое действие в программе должно визуальнo оповещать пользователя о статусе успешности своего окончания, например изменением данных, сообщением об успехе или о неудаче, сообщением об ожидании.

3. Функции системы.

Данные авторизации:

При вводе логина и пароля должна становиться доступной для нажатия кнопка авторизации, неактивная, если хотя бы одно из полей не заполнено.

При вводе неверных данных авторизации должно выводиться сообщение о неудаче авторизации, пользователь должен оставаться на том же экране ввода данных авторизации.

При вводе верных данных авторизации должна происходить успешная авторизация, пользователь должен перенаправляться на экран списка проектов.

Персональные данные пользователя:

При вводе данных в поля личной информации (или их изменении) должна становиться доступной для нажатия кнопка сохранения данных, после ее нажатия новые данные должны отображаться в личном кабинете пользователя.

Действия пользователя в приложении:

При совершении пользователем ввода посредством взаимодействия с какими-либо элементами интерфейса состояние интерфейса должно меняться и отображать получение информации от пользователя (например, при отметке проекта как выбранного, он должен после этого отображаться как выбранный).

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ:

Требования к мобильному приложению:

Функции, которые критически важно реализовать:

1. Авторизовать пользователя в приложении через ЕЛК ВШЭ по данным
 - a. логина,
 - b. пароля.
2. Позволять студенту просматривать список проектов.
3. Позволять студентам просматривать более подробную информацию о любом из проектов в списке:
 - a. название проекта
 - b. описание проекта,
 - c. тип проекта (исследовательский/программный)
 - d. технологические требования,
 - e. руководитель проекта,
 - f. количество студентов на проект,
 - g. сроки выполнения проекта,
 - h. сроки записи на проект,
 - i. форма подачи заявки на проект.
4. Позволять студенту вводить данные своего email и список научных интересов в виде текста.
5. Позволять студенту выбрать проект из списка для выполнения.
6. Позволять студенту отменить сделанный выбор проекта.
7. Предоставлять рекомендации студенту в зависимости от соответствия описания проектов интересам студента, определенного рекомендательной системой.
8. Хранить список проектов.

Функции, которые следует реализовать:

1. Позволять студенту создать описание собственной инициативной темы проекта.
2. Позволять информационным системам ВШЭ получать список проектов в виде текста.

Функции, которые можно реализовать:

1. Предоставлять студенту возможность добавить его дополнительную личную информацию в личном кабинете в текстовом виде, например
 - a. имя,
 - b. о себе,
 - c. факультет,
 - d. группу,
 - e. курс,
 - f. адрес электронной почты,
 - g. номер телефона,
 - h. никнейм в социальной сети.
2. Предоставлять студенту дополнительную текстовую информацию о статусе проекта:

- a. сколько человек его выбрали,
 - b. кто выбрал проект,
 - c. стадии выполнения проектов,
 - d. статус прохождения этих стадий.
3. Передавать решение студента о выборе проекта в информационную систему ВШЭ.

Функции, которые не следует реализовать:

1. Позволять студенту отмечать несколько проектов как заинтересовавшие его.
2. Позволять студенту просматривать список заинтересовавших его проектов.
3. Позволять студенту участвовать в переписке с лицами, связанными с проектом.

Требования к серверной части:

1. обеспечивать регистрацию и авторизацию пользователя,
2. обеспечивать CRUD-взаимодействие с данными о пользователе и его предпочтениях,
3. обеспечивать CRUD-взаимодействие с проектами и их данных,
4. формирование и выдача рекомендаций проектов для пользователя,
5. Обеспечивать подачу заявок на проект.

НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Требования к производительности

Система в условиях даже самой высокой допустимой нагрузки не должна находиться в состоянии ожидания получения данных с сервера дольше 180 секунд.

Система должна поддерживать максимально 40000 консолидированных запросов в сутки, не менее 200 запросов одновременно.

Требования к надежности

Система должна работать корректно при любом вводе пользователя, не должна завершать свою работу аварийно, в случае ошибки должна завершаться в соответствии с предусмотренным протоколом, уведомляя пользователя (graceful exit).

В условиях отсутствия интернет-соединения система не должна показывать пользователю никакую информацию о пользователе или проектах, но должна информировать его о необходимости интернет-подключения для работы приложения.

Система должна быть способна обслужить до 40000 консолидированных запросов в течение суток и до 200 консолидированных запросов в единицу времени, с временем ожидания не более 180 секунд.

Требования к программному окружению, требования к переносимости

Система должна работать на устройствах с лицензионным программным обеспечением IOS 13 или более поздней версии, но не обязана работать ни на каких других платформах. Мобильное приложение должно занимать не более 250 Мб памяти устройства и использовать не более 2 Гб оперативной памяти.

Требования к возможности обслуживания

Для обеспечения возможности обслуживания система должна быть спроектирована в соответствии с основными принципами ООП и мобильной разработки (более подробно описаны в блоке “Организационные требования” данного документа).

Организационные требования.

Приложение должно быть разработано с использованием языков, фреймворков и библиотек, указанных в разделе “Ограничения проектирования” данного документа. При разработке должны быть использованы принципы ООП (декомпозиция, инкапсуляция, полиморфизм), мобильное приложение должно быть разработано в соответствии с архитектурой MVVM (model-view-view-model) для обеспечения качества программного кода, его надежности и возможности обслуживания.

При разработке должна использоваться система контроля версий Git в любой предпочитаемой разработчиками реализации.

Для организации работы команды должны использоваться практики методологии Agile с спринтами по 2 недели, на собраниях по завершении каждого из которых команда должна предоставлять оговоренные в рамках планирования спринта требуемые в срок результаты, а также озвучивать прогресс по процессам, которые пока не подлежат немедленной сдаче. Также команда должна использовать любой предпочитаемый инструмент организации и мониторинга командной разработки. Разработка должна завершиться к 25.03.2023.

Для обеспечения дальнейшей поддержки разработки и возможности обслуживания программный код должен сопровождаться документирующими комментариями. Должна быть разработана программная документация в рамках документации, требуемой на дисциплине “Командный проект” направления Программная инженерия.

Внешние требования.

Система должна быть совместима с цифровыми системами НИУ ВШЭ в рамках авторизации пользователей через личный кабинет. Желательна разработка базы данных и программного интерфейса для получения данных системы о проектах со стороны цифрового блока НИУ ВШЭ.

Система должна функционировать в рамках законодательства Российской Федерации. В особенности, она не должна допускать утечек персональных данных пользователей.

ОГРАНИЧЕНИЯ ПРОЕКТИРОВАНИЯ

Программа реализуется для платформы IOS на языке Swift с использованием фреймворков UIKit и SwiftUI.

База данных системы должна быть разработана на базе PostgreSQL.

Система рекомендаций должна быть разработана на языке Python с использованием алгоритмов кластеризации.

Приложение 1. АНАЛИЗ КОНКУРЕНТОВ ПРИЛОЖЕНИЯ ЦИФРОВОЙ АССИСТЕНТ СТУДЕНТА

	Таблицы для выбора проектных работ на ФКН		
		ЛМС ВШЭ	HSE App X
Мобильное приложение	Нет	Нет	Есть*
Авторизация через личный кабинет ВШЭ	Нет	Есть	Есть*
Личная информация студента	Нет	Есть	Есть*
Личный кабинет студента	Нет	Есть	Есть*
Общая информация обо всех проектах	Есть*	Есть	Нет
Подробная информация о статусе проекта	Есть*	Нет	Нет
Возможность выбирать проект	Нет +-	Есть*	Нет
Возможность отменять выбор проекта	Нет +-	Нет	Нет
Возможность создавать свой проект	Нет +-	Есть*	Нет
Заполнение своих учебных интересов	Нет	Нет	Нет
Рекомендации от системы	Нет	Нет	Нет
Централизованное хранение всех проектов	Нет	Есть*	Нет
Интегрированность в цифровые системы ВШЭ	Нет	Есть	Есть*

Примечание: знаком +- отмечены функции, которые реализованы с использованием сторонних ассоциированных систем. Знаком * отмечена реализация функции, являющаяся предпочтением заказчика и образцом для разработки приложения Цифровой ассистент студента.

Заказчик

Личная подпись



/Паринов А.А

Ответственный по проекту

Личная подпись



/Зубарева Н.Д.

Дата: 27.11.2022