

Задание на разработку проекта для тестирования.

1. Разработать ТЗ (требования).
2. Создать учебный проект (программу) по заданию;
3. Создать Чек-лист для тестирования всей функциональности программы;
4. Автоматизировать тестирование;
5. Выложить проект на github. Сделать описание проекта в файле README.md

архитектор/гоки

тестировщик

разработчик интерфейса

разработчик баз данных



1. Необходимо разработать техническое задание (или требования в более свободной форме)

Вам необходимо составить и согласовать с Заказчиком Техническое задание на проект.

Согласовать макет дизайна и требования к интерфейсу.

Согласовать язык и инструменты разработки.

Согласовать UML диаграммы и модель базы данных (для этого нужно их спроектировать).

При необходимости согласовать дополнительные требования.

Вместе, согласовываем с преподаем,
поддержку тоже — по ролям



2. Создать учебный проект (программу)

2.1 Описание создаваемого проекта:

Проект должен состоять из следующих частей:

- Окно Авторизации/Регистрации;
- Основное окно профиля Пользователя;
- Переходы (Меню):
 - * Настройки (Изменение Имени Пользователя, Язык, Тема (Тёмная/Светлая), Изменение пароля);
 - * Доступные авиаперелеты;
 - * Избранное;
 - * «Выход» из текущего сеанса/Смена пользователя;
- Окно списка доступных авиаперелетов;
- Окно детализированной информации об авиаперелете;
- Окно «Избранных» авиаперелетов.
- Возможность купить билет.

реализация

разработка

документирование



2.2 Окно Авторизации/Регистрации.

В проекте должна быть отработана возможность входа Пользователя по его Логину и Паролю.

Если у Пользователя нет Логина и Пароля, должна быть предусмотрена возможность регистрации Нового Пользователя.

Данные Пользователя (Логин и Пароль) должны храниться в Базе Данных.

При корректных значениях Логина и Пароля, происходит переход в Основное окно Пользователя (Профиль).

При успешно совершенном входе Пользователя, при повторном Перезапуске (Закрытии/Открытии) программы повторная авторизация не требуется. Происходит переход сразу в окно профиля.

Нужно организовать соответствие загружаемого контента Пользователя (Посты, Тема, Язык) и персональных данных (Имя Пользователя, Статус, Аватарка) для каждого конкретного индивидуального Логина Пользователя



2.3 Основное окно профиля Пользователя



описание дизайна + логика взаимодействия

2.3.1 Основное окно профиля

реализация дизайна

Основное окно профиля должно содержать:

- Аватарку (Картинку/Фото Пользователя);
- Поле «Имя» Пользователя;
- Поле «Фамилия» Пользователя;
- Поле «Статус» Пользователя;
- Поле для ввода «Статуса»;
- Кнопка «Изменить статус»;
- Меню с Настройками;
- Возможность выхода из Сеанса/Смены пользователя;
- Кнопки перехода на экран «Доступные авиаперелеты» и «Избранное»;
- Таблица (Список) с Постами (организовать возможность просмотра более подробного описания поста).

дизайн - все вместе
либо  + 

Поле «Статус» может содержать и отображать любое текстовое значение, например как метка (Label). Чтобы изменить это текстовое значение «Статуса», нужно организовать отдельное поле для ввода текста и кнопку «Изменить статус».

Дополнительно можно настроить Локализацию (язык отображения). Пусть на выбор будет два языка: Русский и Английский.



Также можно настроить (выбрать) тему отображения программы: Светлую или Тёмную.

2.3.2 Посты/Объявления

Посты (Объявления) должны содержать информацию туристической направленности (любую, по усмотрению разработчика), например:

- достопримечательности города;
- интересные места;
- информацию об отеле;
- акции и скидки партнеров;
- информацию об авиакомпании;
- и др.

содержание

реализация механики

экр. диз. окна + логика

Посты должны подгружаться из внутренней Структуры или Класса. Можно организовать как некий массив из 10 – 15 возможных Постов.

Для конкретного Пользователя нужно отображать 5-6 определенных Постов из общего массива.

Должен быть организован вертикальный скроллинг для прокрутки и просмотра всех постов;

Посты (Объявления) должны быть расположены один над другим вертикально и должны содержать:

- 1) Автор (или название сайта откуда пришло данное Объявление/Пост);
- 2) Картинку поста;
- 3) Описание поста;



- 4) Количество просмотров;
- 5) Кнопочку «Подробнее», при нажатии на которую открывается Новое Окно с детальным описанием поста. Есть возможность вернуться обратно к общему списку постов.

2.4. Окно просмотра доступных авиабилетов по авиаперелетам

2.4.1 Первое окно (Окно списка доступных авиаперелетов)

В проекте должна быть возможность загрузки данных из формата JSON.

Нужно организовать отображение списка актуальных доступных авиаперелетов.

Данные по каждому авиаперелету должны быть загружены и расположены в ячейки в «табличном виде», ячейки отображаются одна над другой, последовательно.

Нужно организовать вертикальный скроллинг элементов.

Каждая ячейка списка должна содержать:

1. Город отправления.
2. Город прибытия.
3. Дата отправления.
4. Дата возвращения.
5. Цена в рублях.

внешний вид

содержание

механика



6. Иконка «Лайк» (имеет два состояния и означает лайкнул ли пользователь данный перелет, если «Да» – перелет добавляется в «Избранное»).

Кроме того, сверху над списком доступных авиаперелетов необходимо организовать некий Хедер-заголовок, где должно отображаться название «Доступные авиаперелеты», возможность фильтрации списка перелетов по начальному и конечному городу отдельно, кнопку «Обновить», дату и время последнего обновления данных.

2.4.2 Второе окно (Окно детализированной информации об авиаперелете)

Второе окно (экран) – это детализация перелета.

Второе окно открывается при выборе одной из ячеек первого (предыдущего) окна. Должна быть возможность вернуться к первому окну назад.

Второе окно содержит данные по выбранному перелету (город отправления, город прибытия, дата отправления, дата возвращения, цена в рублях), кнопка «Лайк», кнопка «Купить билет».

Кнопка лайк имеет два состояния:

- 1) Перелет нравится
- 2) Перелет не нравится



2.4.3 Третье окно (Избранное)

Третье окно содержит **список Избранных авиаперелетов**.

Перелеты добавляются в «Избранное» при нажатии (активации) кнопочки «Лайк» у перелёта (**заполненное сердечко**), и удаляются из «Избранного» при деактивации «Лайк» (**незаполненное сердечко**) при повторном нажатии на кнопку «Лайк».

При удалении Перелета из списка Избранных перелётов, **необходимо оповестить об этом Пользователя и запросить подтверждение удаления из «Избранного»**. **При добавлении** в «Избранного» по активации «Лайк», **запрос** на подтверждение **не производить**; **можно вывести оповещение**/сообщение о том, что «Перелет был успешно добавлен в Избранное».

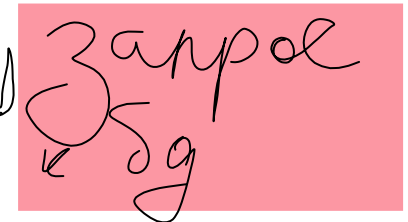
Добавление Перелёта в «Избранное» **должно использовать механизм записи** **необходимой информации в Базу Данных**. Соответственно, для каждого Пользователя «Избранное» будет своё.

Переход в данное окно происходит по нажатию кнопки «Избранное», расположенной в меню, или в специально отведенных кнопках программы).



2.4.3 Примечание:

1. Дизайн произвольный, на усмотрение разработчика (согласовывается с преподавателем).
2. Список доступных авиаперелетов запрашивается один раз при старте программы. Может быть обновлен (актуализирован) на текущую дату, для этого нужно организовать обновление данных и даты последней актуальной загрузки (дата + время) по нажатию кнопки «Обновить». При невозможности загрузить актуальные данные (например, при отсутствии Интернет-соединения) выводить соответствующее сообщение, список доступных авиаперелетов заполнять полем «Данные не доступны».
3. Пока происходит загрузка списка, должен быть отображен индикатор загрузки.
4. Если пользователь лайкнул перелет и вернулся в список перелетов, то в списке у «лайкнутого» перелета должна «гореть» иконка «Лайк». При этом «лайкнутый» перелет должен добавляться в список с «Избранными» перелетами (или просто «Избранное»). *до конца сессии, пока работает приложение можно реализовать через промежуточное сохр. (прим.)* \Leftarrow *ака в БД таблица с полем "избранное"*
5. Если пользователь лайкнул перелет, вышел на первый экран и ещё раз зашел на второй экран, то кнопка-иконка «Лайк» должна быть в состоянии «Перелет нравится» (заполненное сердечко) и наоборот.
6. Если на каком-то из экранов «Лайк» переводится в состояние «Перелет не нравится» (незаполненное сердечко), то это



состояние одинаково отображается для другого экрана (Окна) и Перелет удаляется из «Избранного».

3. **Создать Чек-лист для тестирования всей функциональности программы**

Необходимо составить Чек-лист (желательно с помощью ресурса Sitechco.ru) для проверки функциональности программы. Рекомендовано расширять список проверок в Чек-лист с каждой новой доработкой по каждому модулю.

4. **Необходимо автоматизировать тестирование, используя Unit-тестирование, а также при необходимости дополнительные инструменты тестирования.**

5. **Выложить проект на github. Сделать описание проекта в файле README.md**

Вести git обязательно !!!

