# **Разработка базового расписания**

### **1) Прототипы экранных форм**

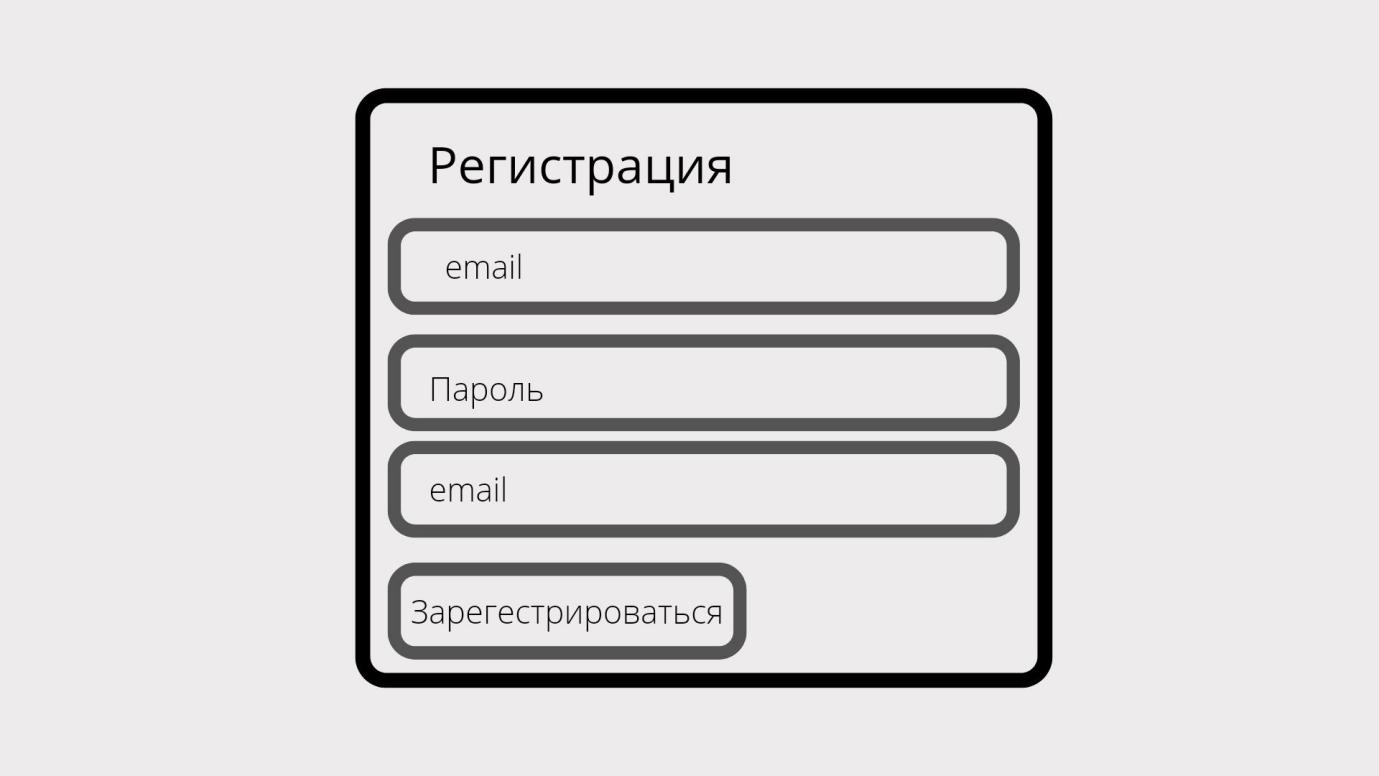
* Прототип формы авторицации

Начальное окно, окно авторизации, которое отображается при входе в приложение, служит стартовым экраном откуда пользователь может авторизоваться ввести свою почту в строку “EMAIL”, пароль в срок “ПАРОЛЬ” и нажать кнопку авторизоваться. Также возможно зарегистрироваться нажав кнопку регистрации.



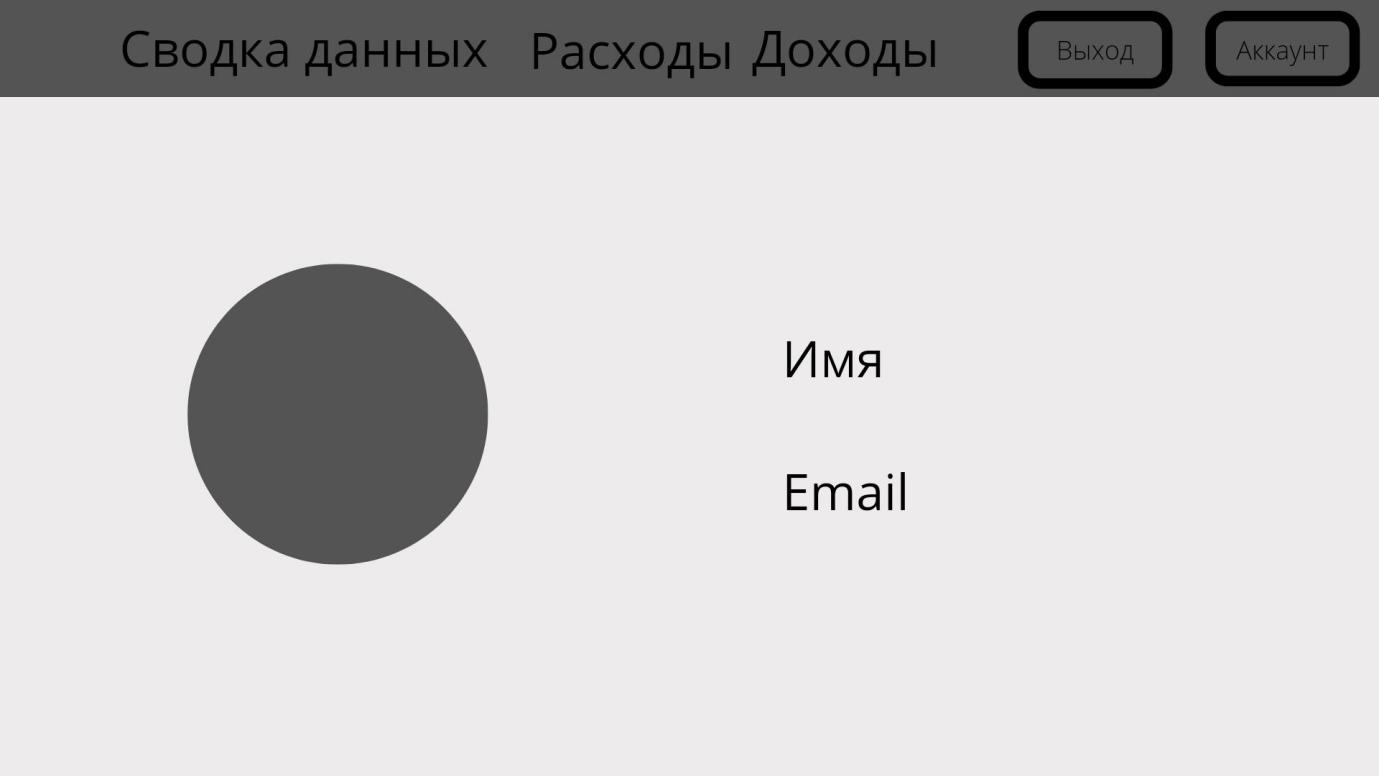
* Прототип формы регистрации

Форма регистрации запускается при нажатии кнопки регистрации в форме авторизации, форма регистрации содержит в себе поле “EMAIL” куда нужно ввести адрес электронной почты, поле “Логин” и поле пароль, кнопка регистрация отвечает за проверку правильно введенных данных , а также за саму регистрацию.



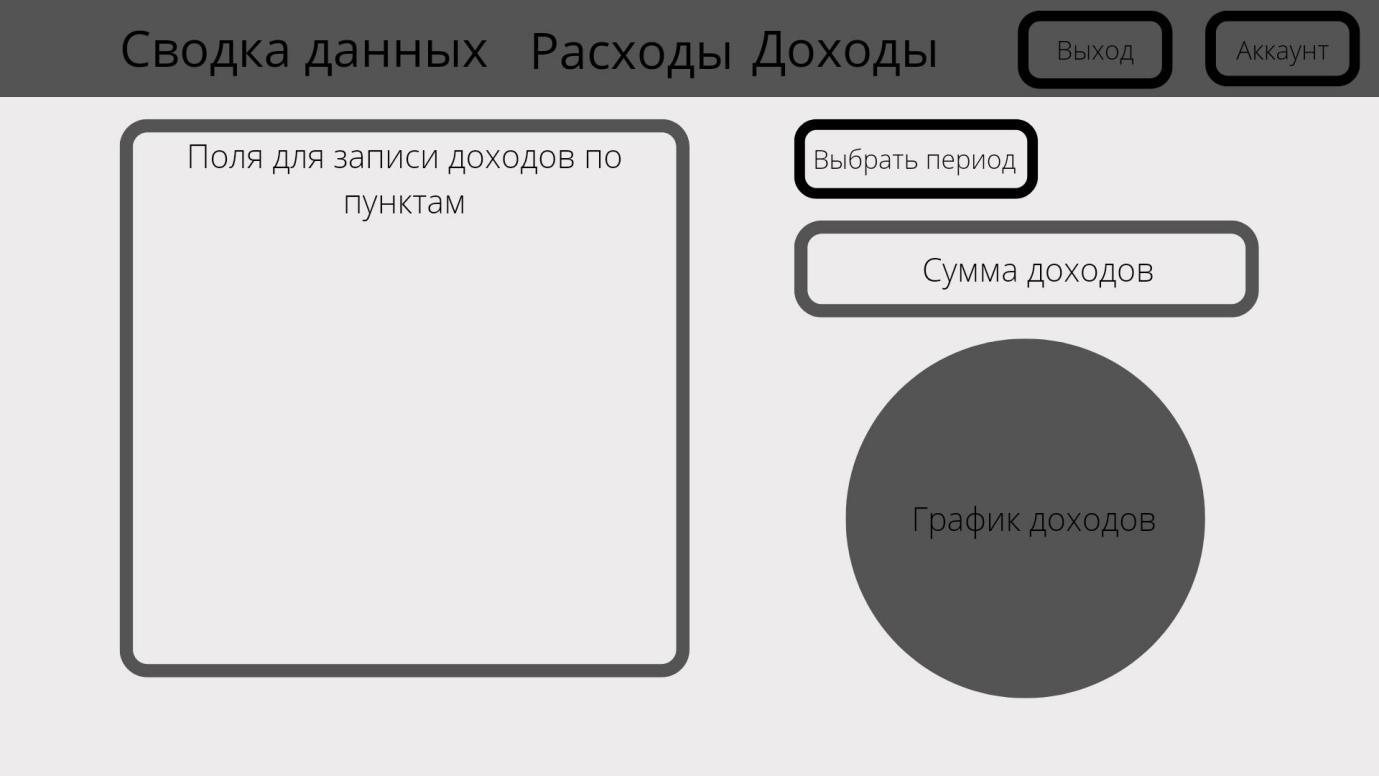
* Прототип страницы пользователя

Страница пользователя, в котором пользователь может изменить имя Email или добавить или изменить фото пользователя.



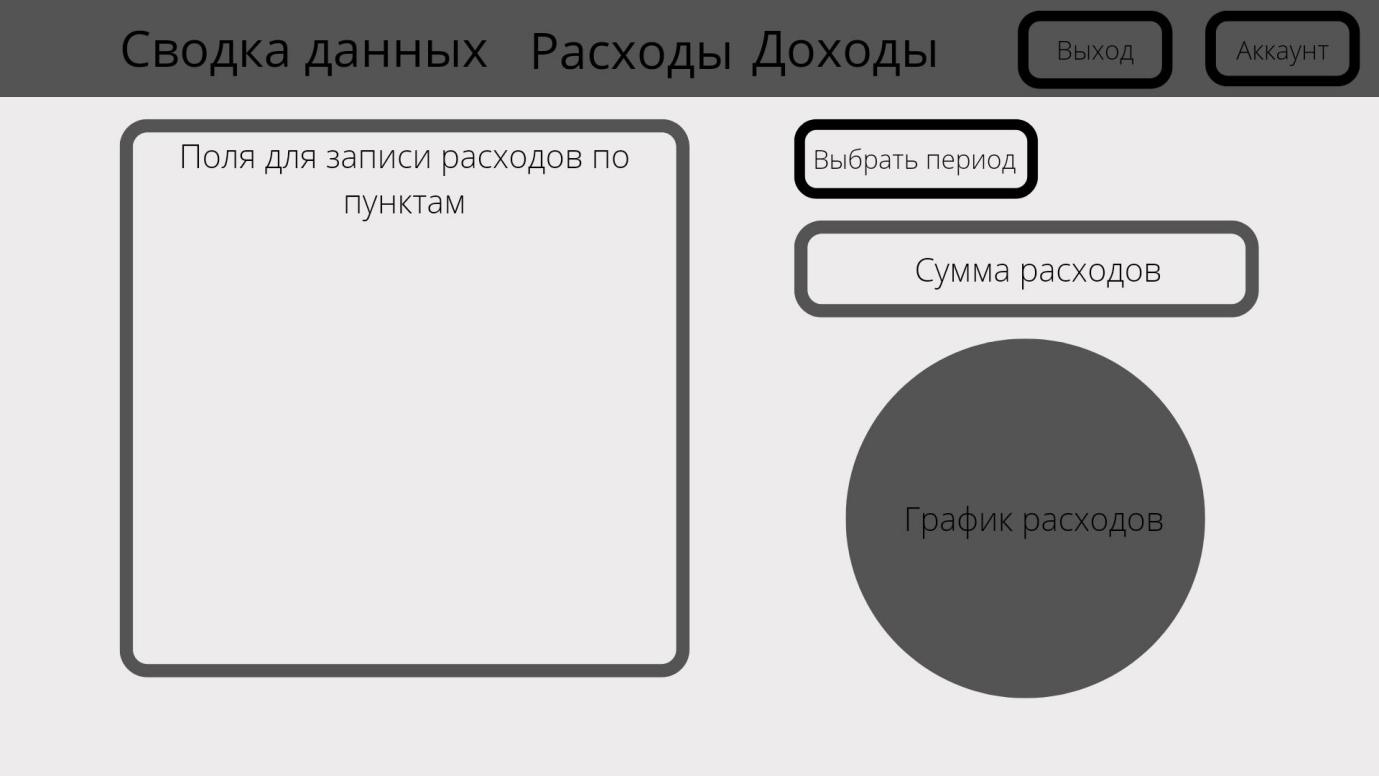
* Прототип интерфеса страницы доходов

Страница доходов пользователя, в котором можно добавить доходы по определённой категории, переключиться на расходы нажав на кнопку расходы, выйти из системы нажав на кнопку “Выход”.При внесение доходов появляться диаграмма доходов по каждой категории.



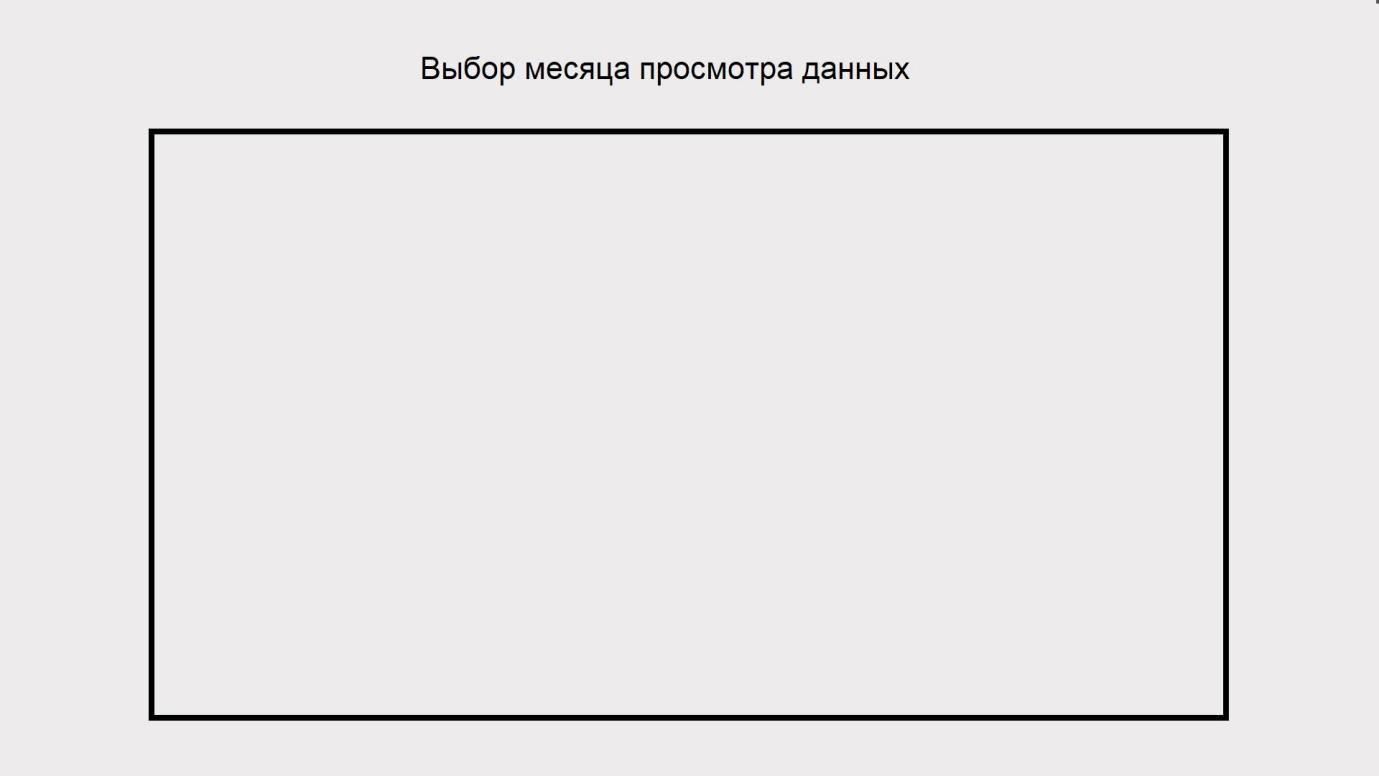
* Прототип интерфейса страницы расходов

Страница расходов пользователя, в котором можно добавить расходы по определённой категории, переключиться на доходы нажав на кнопку доходы, выйти из системы нажав на кнопку “Выход”.При внесение расходов появляться диаграмма расходов по каждой категории.

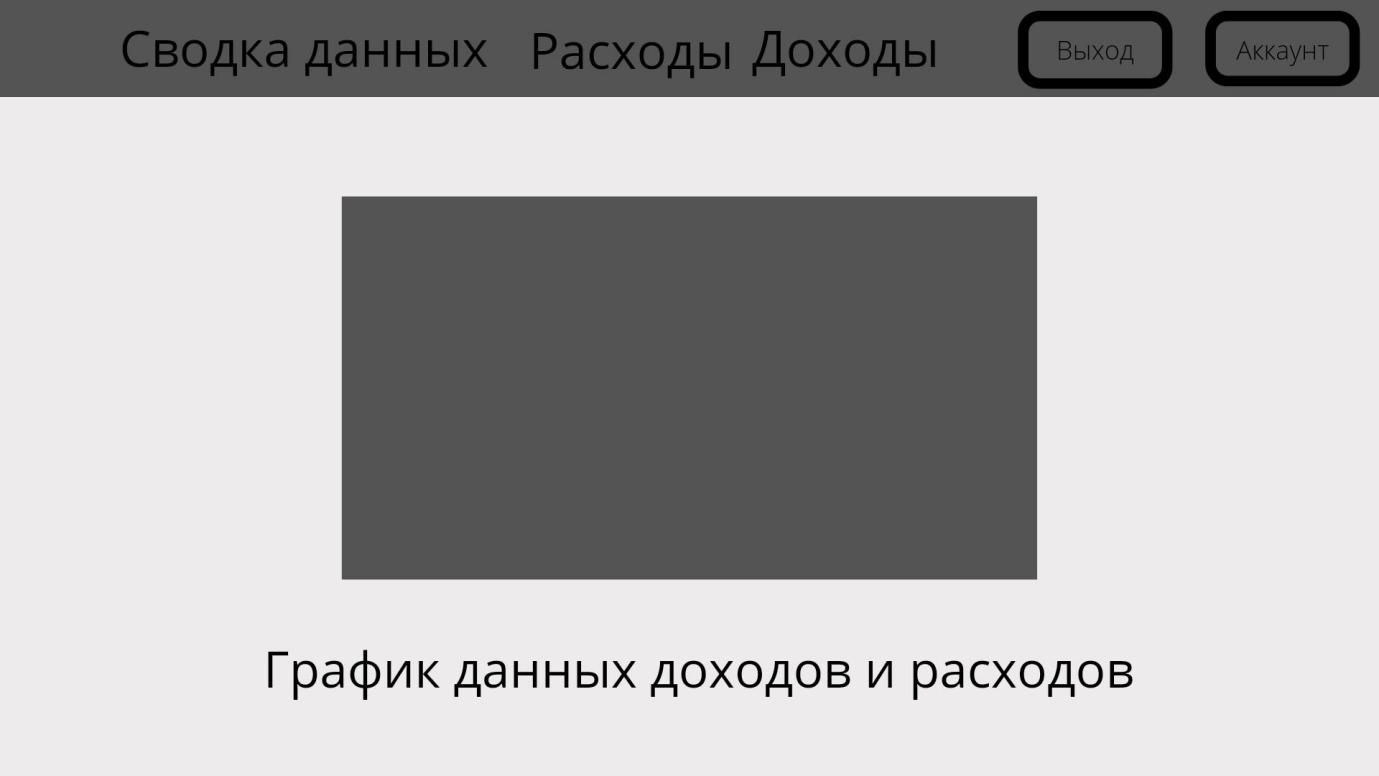


* Прототип окна выбора периода

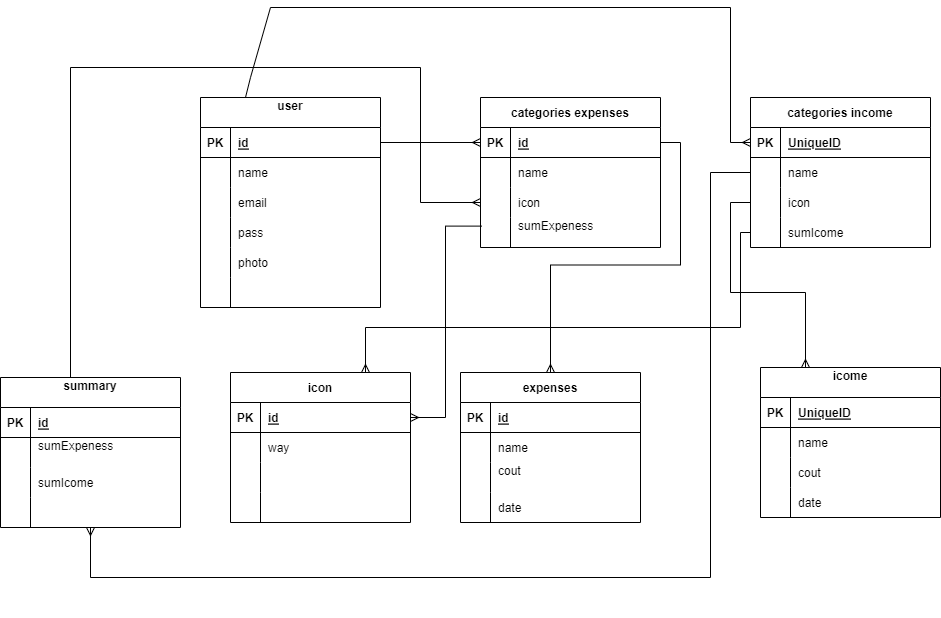
Окно выбора периода предназначено для выбора периода, за который будет подсчитаны доходы и расходы



### Прототип страницы сводка данных



### 2)Диаграмма сущностей



3. Разработка API системы

* Authorization

Функция предназначена для входа в систему и для проверки вводимых данных во избежание SQL- инъекции и прочие. Функция обращается к БД для проверки правильности ввода Email и пароля, если все данные верны, то функции разрешает доступ к системе. Срабатывание функции вызывают встроенные событие OnClick, которые вызываются по нажатию. Входные данные: string: Email, string: Пароль.

* Registration

Функция предназначена для регистрации пользователя в системе и для проверки вводимых данных во избежание SQL- инъекции и прочие. Функция обращается к БД для проверки если такой пользователь по Email, если нету то пользователь заноситься в БД, то функции разрешает доступ к системе. Срабатывание функции вызывают встроенные событие OnClick, которые вызываются по нажатию. Входные данные: string: Email, string: Пароль,string: Имя.

* Income

Функция предназначена для внесения доходов авторизованного пользователя по определенному разделу в БД. Срабатывание функции вызывают встроенные событие OnClick, которые вызываются по нажатию. Входные данные: int: income.

* Expenses

Функция предназначена для внесения расходов авторизованного пользователя по определенному разделу в БД. Срабатывание функции вызывают встроенные событие OnClick, которые вызываются по нажатию. Входные данные: int: expenses.

* chartIncome

Функция предназначена для отрисовки графика доходов. График имеет тип donut. Функция получает данные из БД по определенному пользователю. Входные данные: int: income.

* chartExpenses

Функция предназначена для отрисовки графика расходов. График имеет тип donut. Функция получает данные из БД по определенному пользователю. Входные данные: int: expenses.

* chartSummary

Функция предназначена для отрисовки графика сравнения доходов и расходов. график имеет вид гистограммы. Функция получает данные из БД по определенному пользователю. Входные данные: int: expenses, int: income.

* Exit

Функция предназначена для выхода из системы и возвращения на экран авторизации.

* ChangingData

Функция предназначена для изменения данных о пользователе (имя, пароль, email или фото). Входные данные: string: Email, string: Пароль,string: Имя.

* Sassion

Функция предназначена для отсчета времени, которое пользователь авториззован, по истечению данного времени сессия прервется и будет необходимо снова авторизоваться.

**Иерархическая структура работ:**

1. Разработка технического задания

1.1. Сбор требований;

1.2. Определение стадий и этапов разработки

1.2.1. Определение стадий разработки;

1.2.2. Определение сроков разработки;

1.3. Общее описание

1.3.1. Назначение продукта;

1.3.2. Взаимодействие продукта;

1.3.3. Допущения и ограничения продукта;

1.3.4. Определение функций продукта;

2. Разработка приложения

2.1. Backend-разработка

2.1.1. Проектирование базы данных

2.1.1.1. Определение структуры базы данных;

2.1.1.2. Определение связей между сущностями;

2.1.1.3. Определение взаимодействия с базой данных;

2.1.2. Разработка API приложения

2.1.2.1. Authorization;

2.1.2.2. Registration;

2.1.2.3. Income;

2.1.2.4. Expenses;

2.1.2.5. chartIncome;

2.1.2.6. chartExpenses;

2.1.2.7. chartSummary;

2.1.2.8. Exit;

2.1.2.9. ChangingData;

2.1.2.10. Sassion;

2.1.3. Сетевое взаимодействие

2.1.3.1. Определение протокола взаимодействия;

2.1.3.2. Обеспечение защищённости соединения;

2.1.3.3. Определение местоположения хранения данных;

2.1.4. Взаимодействие с UI

2.1.4.1. Определение модели привязки данных;

2.1.4.2. Создание механизма обновления данных;

2.2. Frontend-разработка

2.2.1. Дизайн

2.2.1.1. Разработка макетов страниц

2.2.1.1.1. Разработка макета главной страницы;

2.2.1.1.2. Разработка макета страницы календаря;

2.2.1.1.3. Разработка макета страницы вызова;

2.2.1.1.4. Разработка макета страницы настроек;

2.2.1.1.5. Разработка макета страницы аккаунта;

2.2.1.1.6. Разработка макета страницы помощи;

2.2.1.1.7. Разработка страницы контактов;

2.2.1.2. Взаимодействие с backend

2.2.1.2.1. Определение модели привязки данных;

2.2.1.2.2. Настройка механизма обновления данных;

2.2.1.2.3. Определение динамического взаимодействия с данными;

2.2.1.3. Разработка общего стиль-кода приложения

2.2.1.3.1. Создание логотипа приложения;

2.2.1.3.2. Создание иконок элементов;

2.2.1.3.3. Определение стиля и размера шрифтов;

3. Приемо-сдаточные испытания

3.1. Подготовка и проведение демонстрации;

3.2. Проведение испытаний;

4. Размещение приложения

4.1. Аренда сервера;

4.2. Развёртывание приложения;

4.3. Размещение в магазине приложений;

5. Поддержка приложения

5.1. Мониторинг работоспособности;

5.2. Получение и обработка обратной связи;

5.3. Улучшение работы приложения

5.4. Добавление новой функциональности;

**Оценка времени выполнения проекта по методу PERT:**

Произведем вычисления для определения временных затрат на реализацию проекта. Для этого определим состав работ их оптимистичные, пессимистичные и средние трудозатраты в часах:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Работы | Количество | Оптимистичные трудозатраты | Пессимистичные трудозатраты | Наиболее вероятные трудозатраты |
| Создание сущностей | 7 | 2 | 8 | 3 |
| Создание макетов | 7 | 30 | 60 | 50 |
| Создание методов API | 10 | 40 | 70 | 55 |

Посчитаем средние трудозатраты по каждой работе:

Посчитаем среднеквадратичную оценку:

Посчитаем :

Посчитаем :

Оценка суммарной трудоёмкости проекта с вероятностью 95%: