

Факультет программной инженерии и компьютерной техники

Архитектура программных систем

Лабораторная работа №1: Алкулятор в терминах UML

Преподаватель: Перл Иван Андреевич

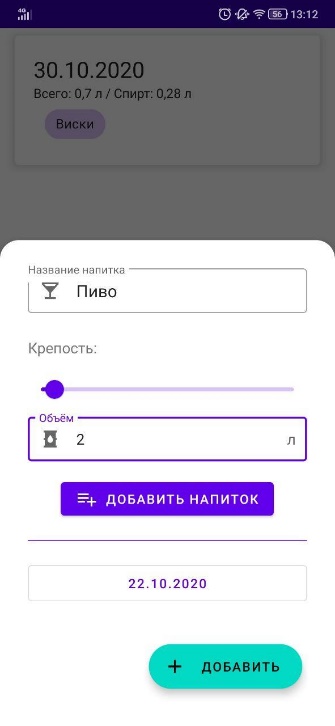
Выполнил: Кульбако Артемий Юрьевич Р33112

# Задание

Выбрать любую реально существующую систему и описать её в терминах UML. Желательно чтобы система была не полностью информационной, но опиралась на информационную систему как показано в примере на лекции (Point of sale). Необходимо описать границы системы на разных уровнях, а также описать сценарии использования для нескольких Акторов.

# Описание системы

## Цель системы

Алкулятор – приложение для контроля выпитого алкоголя, с возможностями:

* Смотреть список выпитого у друзей
* Рассчитывать время отрезвления и вывода спирта из крови
* Просмотра глобальной статистики по выпитому пользователем за всё время использования

В первую очередь нацелено на молодую аудиторию, прожигающую свою молодость на Думской и Рубенштейна. Пока что существует лишь в виде прототипа для Android 7.0 и выше.

Рисунок - пример интерфейса

## Термины

**Пиршество** – запись, инкапсулирующая информацию о выпитых **напитках** в конкретную дату.

**Напиток** – алкогольный напиток (название, крепость, объём).

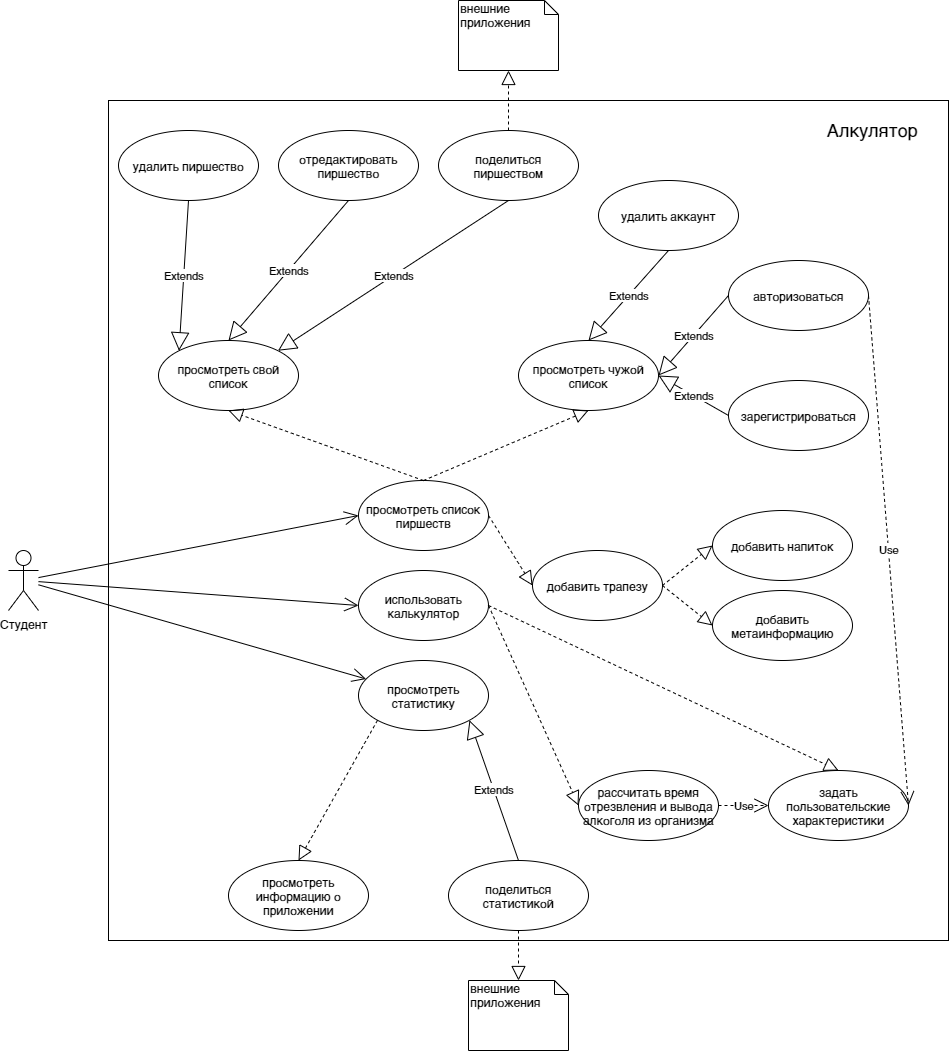
## Функциональные требования

* Система должна отображать список пиршеств пользователя в виде списка на стартовом экране.
* Свои пиршества можно редактировать, удалять, делится ими.
* Система должна отображать список пиршеств друзей пользователя, если он зарегистрирован.
* Должна быть реализована регистрация, авторизация, удаление аккаунта(!).
* Должен присутствовать калькулятор отрезвления (информация о весе/поле пользователя должна быть сохранена).
* Просмотр глобальной статистики, статистики за заданный период, возможность делится ей.

## Нефункциональные требования

* Хранение данных о зарегистрированных пользователях в SQL бд на сервере, для незарегистрированных только в локальной бд устройства.
* Мобильный приложения должны быть реализованы в соответствии с гайдлайнами ос и на нативном уровне (Android SDK – Kotlin, iOS SDK – Swift).
* В качестве минимальных версий ос использовать те, которые позволят реализовать функционал на 100% без гонки за актуальной версий.
* Возможность авторизации через OAuth.2.0.
* Возможность делится статистикой реализовать через Intent-ы для Android и App Extensions для iOS.
* Передача данных между бэкендом и фронтом через json.
* Легковесный бэкенд на Ktor или Node.js.

## Диаграмма UseCase



## Прецеденты использования

|  |  |
| --- | --- |
| Имя прецедента: | Просмотреть список своих пиршеств |
| ID: | 1 |
| Описание: | Просмотр списка своих пиршеств в виде списка, с большой кнопкой добавления нового пиршества. |
| Акторы: | Пользователь |
| Основной поток: | Актор запустил приложение. |
| Альтернативный поток: | Список пиршеств пуст: вывести информацию об этом и предложение с добавлением нового. |

|  |  |
| --- | --- |
| Имя прецедента: | Удалить пиршество |
| ID: | 2 |
| Описание: | Удаление пиршества из списка своих пиршеств. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: |  |

|  |  |
| --- | --- |
| Имя прецедента: | Редактировать пиршество |
| ID: | 3 |
| Описание: | Редактировать список напитков и метаинформацию о пиршестве. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: |  |

|  |  |
| --- | --- |
| Имя прецедента: | Поделится пиршеством |
| ID: | 4 |
| Описание: | Поделить информацией о пиршестве через встроенные механизмы ос. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: | При запрете ос на шаринг информации из приложения, предложить дать разрешение на это. |

|  |  |
| --- | --- |
| Имя прецедента: | Просмотреть список чужих пиршеств |
| ID: | 5 |
| Описание: | Просмотреть список пиршеств друзей главного актора. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: | Если пользователь не авторизирован, предложить войти или зарегистрироваться. |

|  |  |
| --- | --- |
| Имя прецедента: | Авторизироваться |
| ID: | 6 |
| Описание: | Авторизация через почту-пароль или сторонние сервисы. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 после перехода на экран с чужими пиршествами |
| Альтернативный поток: |  |

|  |  |
| --- | --- |
| Имя прецедента: | Удалить аккаунт |
| ID: | 7 |
| Описание: | Удаление аккаунта пользователя. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 5 |
| Альтернативный поток: |  |

|  |  |
| --- | --- |
| Имя прецедента: | Зарегистрироваться |
| ID: | 8 |
| Описание: | Регистрация внутри приложения. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 после перехода на экран с чужими пиршествами |
| Альтернативный поток: | При ошибке регистрации предложить попробовать ещё раз чуть позже. |

|  |  |
| --- | --- |
| Имя прецедента: | Добавить трапезу |
| ID: | 9 |
| Описание: | Добавление трапезы к списку трапез пользователя. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: | Отмена добавления трапезы возвращает к списку. |

|  |  |
| --- | --- |
| Имя прецедента: | Добавить напиток |
| ID: | 10 |
| Описание: | Добавление напитка к трапезе. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 9 |
| Альтернативный поток: | Отмена добавления напитка возвращает к списку. |

|  |  |
| --- | --- |
| Имя прецедента: | Добавить метаинформацию |
| ID: | 11 |
| Описание: | Добавление/изменение метаинформации о трапезе. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: | Отмена изменения метаинформации возвращает к списку. |

|  |  |
| --- | --- |
| Имя прецедента: | Просмотреть глобальную статистику |
| ID: | 12 |
| Описание: | Просмотреть пользовательскую статистику по выпитому за всё время. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: | Если трапезы отсутствуют, вывести пустой график и подсказку об этом. |

|  |  |
| --- | --- |
| Имя прецедента: | Просмотреть информацию о приложение |
| ID: | 13 |
| Описание: | Просмотреть информацию о приложении (версия, авторы и т.д.). |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 12 |
| Альтернативный поток: |  |

|  |  |
| --- | --- |
| Имя прецедента: | Поделится статистикой |
| ID: | 14 |
| Описание: | Поделить информацией о пиршестве через встроенные механизмы ос. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 12 |
| Альтернативный поток: | При запрете ос на шаринг информации из приложения, предложить дать разрешение на это. |

|  |  |
| --- | --- |
| Имя прецедента: | Использовать калькулятор |
| ID: | 15 |
| Описание: | Показать экран с вкладками для сохранение пользовательских характеристик и калькулятором. |
| Акторы: | Пользователь. |
| Основной поток: | Из пр. 1 |
| Альтернативный поток: |  |

|  |  |
| --- | --- |
| Имя прецедента: | Рассчитать время отрезвления и вывода алкоголя из организма |
| ID: | 16 |
| Описание: | Вывести время вывода алкоголя из организма, рассчитанное на основе выпитого сейчас и пользовательских характеристиках. |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 15 |
| Альтернативный поток: | Если пользовательские характеристики не сохранены, предложить задать их. |

|  |  |
| --- | --- |
| Имя прецедента: | Задать пользовательские характеристики |
| ID: | 17 |
| Описание: | Ввести и сохранить пользовательские характеристики, необходимые для расчётов (вес, пол). |
| Акторы: | Пользователь |
| Основной поток: | Из пр. 15 |
| Альтернативный поток: | Сигнализировать об ошибках ввода. |

# Вывод

Перезалив с добавлением вывода

Выполнив данную лабораторную работу, я улучшил свои навыки составления UML-диаграмм и прецедентов использования для информационных систем (и в теории, не только для информационных).

UseCase-диаграммы (разновидность UML-диаграмм) – отличный способ представления взаимодействия некий сущности (актора) с некой системой. Таким образов можно удобно и понятно описать, как то или иное действие вызывает другое действие и от чего оно может быть зависимо. Средства UML позволяют обобщать несколько похожий действий разных акторов к одному, что уменьшает диаграмму, а также задать область действия каждой системы на одной диаграмме, благодаря чему на общем файле можно удобно наблюдать полный цикл жизни системы.

К тому же, из UseCase можно выйти на составление прецедентов использования, что уже будет полезно для разработчиков будущей системы, так как содержит подробную информацию по действию.