

Авторизация:

На сервере хранится таблица игроков.

На телефоне есть экран с кнопками «авторизация» и «регистрация»

При регистрации появляется просьба ввести ник,е-mail,пароль, ~~номер и CVC-код банковской карты, номер паспорта и адрес прописки.~~

При авторизации появляется просьба ввести ник и пароль

На экранах авторизации/регистрации будет кнопка «Войти». При нажатии на эту кнопку на сервер отправляется наследник класса Query класс экземпляр класса Player с полями:

1. Boolean isNew
2. String nickname
3. String e-mail
4. String password **(ОБЕСПЕЧИТЬ ШИФРОВАНИЕ)**

У класса же Query будут присутствовать поля:

1. Integer connectionKey
2. String operation
3. String source

Поле connectionKey = null

Поле operation = Auth,

source = null (отвечает за пользователя,осуществившего запрос) остальные очевидно.

В случае регистрации на сервере проходит проверка, занят ли данный никнейм. Сервер отправляет на телефон экземпляр класса Answer с полями:

1. Boolean isOk
2. String operation
3. String description
4. Integer connectionKey.

,если никнейм занят то в operation указывается тип операции (authorisation), в description указывается описание проблемы (Nickname Lololoshka is already used by another user) а isOk=false. Иначе isOk=True, operation=authorisation, description=Authorisation is finished well.

В случае,если после проверки на сервере оказалось,что никнейм не занят, то в таблицу БД записывается новый аккаунт и отправляется экземпляр класса Answer с сгенерированным ключом connectionKey(чтобы мошенник не мог связаться с нашим сервером напрямую и от лица любого человека написать любому). По получении данного ответа в телефоне в локальную БД мгновенной авторизации по клику(как в яндекс.почте,почте mail.ru) добавляется аккаунт,что пытался зарегистрироваться. На телефоне происходит переход в личный кабинет игрока. На данном этапе он будет представлять из себя просто кнопку гаечки в левом верхнем углу при нажатии на которую выпадет единственная опция:выйти из аккаунта. По нажатию на эту кнопку отправляется класс Query с текущим connectionKey и с operation=Exit. При поступлении на сервер любого запроса (кроме запросов авторизации и регистрации) сервер проверяет на соответствие ключ connectionKey для данного пользователя. Если ключ соответствует,то действие совершается с положительным Answer. Иначе ничего не выполняется,а Answer отрицателен. В случае,если Answer положителен,на сервере сбрасывается connectionKey, а в таблице пользователей для быстрого доступа на телефоне данный пользователь пропадает.

Дедлайн: 25 октября