***Техническое задание на финальный проект студента***

***группы JV1901***

***(Сентябрь 2019 - Июнь 2020 гг)***

**Краткая информация о проекте**

Данная CRM (система управления взаимоотношениями с клиентами) называется “WeCRM”, в настоящем документе допускается именовать “система” или “приложение”. Эта система, заказчиком, которого является фитнес-клуб “Terra Sport” будет разрабатываться студентом “IT Academy” Сабыровым Айдином.

Список терминов и сокращений, которые могут присутствовать в данном документе:

|  |  |
| --- | --- |
| Термин или Сокращение | Полная форма |
| БД | База данных |
| CRM | Система управления взаимоотношениями с клиентами |
| “ADMIN” | Пользователь с правами администратора |
| “USER” | Пользователь данной системы(клиент) |

Кто заказчик?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название  и реквизиты | Миссия, цели | Продукция | Целевая аудитория | Кто их конкуренты? |
| Фитнес-клуб  “Terra sport”  г.Бишкек 4мкр  т. 0312900171 | Автоматизация администрирования.  Мониторинг посещений клиентов.  Ведение фин.  учета | “WeCRM”  Система управления взаимоотно -  шениями  с клиентами | Администрация и  Клиенты фитнес клуба | CRM система  “1C Фитнес” |

**Цель проекта**

|  |  |
| --- | --- |
| Цель: | Назначение создания |
| Автоматизация администрирования и мониторинга посещений клиентов, просмотр статистики посещений, увеличение продаж клубных карт. | Отказ от бумажного хранения данных о клиенте и его клубных карт.  Облегчение работы администраторам в поиске клиентов.  Автоматический учет посещений клиентов.  Просмотр финансового состояния за определенный период по определенным секциям тренировок. |

**Предмет разработки**

*Цель данной системы состоит во внедрении автоматизации администрирования, мониторинга и контроля посещений клиентов, просмотр статистики продаж клубных карт по направлениям.*

*Создание “WeCRM” должно обеспечить:*

* *Увеличение продаж клубных карт*
* *Усиление аналитики*
* *Сделает работу сотрудников прозрачной*
* *Снизит затраты на маркетинг*
* *Удобный поиск клиентов в базе и контроль их посещений администраторами*
* *Повысит лояльность клиентов*

**Требования к системе**

В бизнес-процессе “WeCRM” выделяются следующие участники:

Администратор – работник фитнес-клуба входящий в состав управления.

Пользователь – гражданин КР, желающий получать фитнес-услуги.

Функциональные требования:

* авторизация и аутентификация пользователей
* пользователи с ролью “ADMIN” должны получать список клиентов и информацию о его абонементах
* контролировать состояние абонементов и изменять количество посещений по мере визитов клиента
* блокировать и разблокировать аккаунт клиента, деактивировать абонемент независимо от наличия количества посещений.
* просмотр месячного дохода фитнес клуба, а также по определенным направлениям
* пользователи с ролью “USER” имеют: личный кабинет, где могут покупать абонементы и просматривать их состояния, вести дневник, оставляя заметки о последних результатах тренировок
* возможность клиента вступить в членство клуба для получения скидок на приобретение абонементов
* все гости приложения могут зарегистрироваться и пользоваться фитнес-калькулятором, базой упражнений для тренировок в тренажерном зале и в домашних условиях.

Требования к системе и в целом:

* аутентификация
* авторизация, возможность разграничения по ролям
* безопасность информации о клиентах
* слоистая архитектура приложения
* соблюдение таких принципов, как: SOLID, DRY, KISS
* завершения стадий и этапов разработки в заданные сроки
* при изменении структуры системы, согласовывать с заказчиком

В базе данных приложения должна хранится информация о направлениях тренировок, работниках фитнес клуба, пользователях и его ролей, кошельке пользователя, параметры пользователя для ведения личного дневника, информация абонементов пользователя и их посещений.

В БД будут следующие таблицы:

**Training\_group\_catregory,** где есть такие колонки как:

id (Long), name(String), created\_date(Timestamp)

**Training\_group,** где есть такие колонки как:

id (Long), training\_group\_category\_id(Long), name (varchar), employee\_id (Long), subscription\_price (BigInteger), training\_time(String), status(varchar)

**Promo\_codes,** где есть такие колонки как:

id (Long), name(String), discount\_percentages(Integer)

**Training\_group,** где есть такие колонки как:

id (Long), wallet\_from\_id(Long), wallet\_from\_id(Long), status(varchar),

currency(String)

**Employees,** где есть такие колонки как:

id (Long), user\_id (Long), position (varchar), status(varchar)

**Users,** где есть такие колонки как:

id (Long), email (varchar), password (varchar), phone\_number (varchar), name (varchar), surname (varchar), birth\_date (date), gender (varchar), is\_active (integer)

**Roles,** где есть такие колонки как:

id (Long), name (varchar)

**User\_roles,** где есть такие колонки как:

user\_id (Long), role\_id (Long)

**Person\_params,** где есть такие колонки как:

id (Long), user\_id (Long), height (float), weight (float), neck\_girth (float),

shoulder\_width (float), waste\_girth (float), hip\_girth (float), bicep\_girth (float),

chest\_girth (float), note\_girth (float)

**Subscriptions,** где есть такие колонки как:

id (Long), training\_group\_id (Long), session\_quantity (integer), code (varchar), discount\_percentages (integer), total\_amount(BigInteger), status (varchar), expiration\_date(Date), frozen\_until\_date(Date)

**Visits,** где есть такие колонки как:

id (Long), subscription\_id (Long), visit\_date (date)

**Visit\_history,** где есть такие колонки как:

id (Long), manager\_id (Long), visit\_id (Long), visit\_time(Timestamp),

session\_quantity\_after\_visit(Integer), subtracted\_sessions(Integer)

**Wallet,** где есть такие колонки как:

id (Long), user\_id (Long), requisite (varchar), balance (BigInteger), status (varchar)

**User\_confirmation\_codes,** где есть такие колонки как:

id (Long), code(String), user\_id(Long), is\_active(boolean)

**Auth\_logs,** где есть такие колонки как:

id (Long), user\_id(Long), status(String), is\_recovery(Integer)

**Стадии и этапы разработки**

1. Создать удаленную базу данных.

Срок выполнения до 2.06.2020

1. Собрать spring boot проект

Срок выполнения до 02.06.2020

1. Подумать о таблицах и внести их в базу данных

Срок выполнения до 02.06.2020

1. Создать Entity таблиц, сервисы и их имплементации, контроллеры в spring boot проекте

Срок выполнения до 02.06.2020

1. Написать бизнес-логику для регистрации / авторизации и аутентфикации

Срок выполнения до 02.06.2020

1. Написать бизнес-логику контроля посещений клиентов

Срок выполнения до 04.06.2020

**Требования к информационному и программному обеспечению:**

**Серверная часть**

Язык Программирования: Java

Версия языка: 1.8

База данных: PostgreSQL

Фреймворки: Spring Boot, Spring Security

**Клиентская часть**

Вертка: HTML, CSS

Язык программирования: Java Script

Фреймворки: ReactJS