

## АНАЛИЗ АЛГОРИТМОВ ДЛЯ РАЗРАБОТКИ CRM-СИСТЕМЫ

Целью настоящей работы является анализ алгоритмов для задачи разработки CRM-системы управления взаимоотношениями с клиентами (от англ. CustomerRelationshipManagement) с поддержкой принятия решений DSS. Система управления взаимоотношениями с клиентами должна облегчать привлечение новых клиентов, сбор и анализ информации о взаимодействии с клиентом, настройку бизнес-процессов и последующий анализ результатов. Система поддержки принятия решений (СППР) (от англ. DecisionSupportSystem, DSS) – компьютерная автоматизированная система, целью которой является помощь растущему количеству пользователей смартфонов и мобильных приложений для поддержания отношений с потенциальными заёмщиками, математической оценке рисков, сбора и анализа клиентских трат.

Основные компоненты CRM-систем – база данных с клиентами, серверное приложение, работающее с базой данных и пользовательский интерфейс. Архитектура серверного приложения должна иметь гибкость, устойчивость к сбоям и защиту от несанкционированного доступа к информации. В связи с этим систему лучше устроить по примеру клиент-серверной архитектуры, в которой сервер ответственен за сбор, предобработку и сохранение данных, а клиент – за работу с интерфейсом и визуализацией.

Для клиентского приложения выбор веб-архитектуры разумен по следующим причинам:

- Веб-приложение снимает с пользователя обязанности администратора приложения, а именно: его установку, обновление, управление аппаратными средствами компьютера и т.д. Поскольку ориентировочные клиенты – маркетологи и менеджеры, в рабочие обязанности которых не входит системное администрирование.

- При желании в архитектуру клиент-сервер можно добавить мобильное клиентское приложение, однако это нецелесообразно. Большую часть времени с системой будут работать в определённые рабочие часы. Поэтому клиентское приложение, работающее на мобильных устройствах, будет использоваться крайне редко.

- Выбор веб-приложения упрощает разработку, так как отпадает необходимость поддерживать кроссплатформенность, обратную совместимость и синхронизацию данных между клиентами.

Пользователи, как с ролью клиента, так и администратора, работают с системой через пользовательский интерфейс. Для упрощения доступа к данным, различные операции сгруппированы по общему назначению, а информация о клиентах и их данных предоставляется в виде таблицы. Большая часть функциональности, доступная клиенту, также доступна и администратору. Однако некоторые действия, такие как подключение рекламной интеграции, подключение магазина приложений, осуществляются только клиентом. Любые операции по отчётности, анализу рисков и назначению платежей же, наоборот, доступны только клиенту.

Клиентское приложение связывается с сервером через асинхронные запросы, чтобы не вынуждать пользователя ждать полной загрузки данных. Основная задача клиентского приложения – быть простым, понятным в использовании и удобным. Основные операции производятся быстро – в 5-6 кликов, не заставляя пользователя ждать загрузки страницы.

Для того, чтобы максимизировать удобство использования интерфейса и минимизировать количество кликов до основных элементов, главная страница приложения спроектирована таким образом, чтобы содержать в себе элементы, отображающие основную информацию по клиенту. Переходы между страницами осуществляются при помощи бокового меню, содержащего в себе ссылки на основные страницы приложения.

Таким образом, спроектированный интерфейс уменьшает среднее время пребывания пользователя в системе, при этом не перегружая визуальное пространство элементами управления.

В сфере мобильного маркетинга, данные, получаемые и агрегируемой системой, имеют большое количество полей и свойств, а также имеют формат, делающий невозможным использование одной базы данных для всего массива данных.

В докладе проводится анализ и выбор алгоритмов архитектуры серверного и клиентских приложений, архитектуры данных и систем управления базами данных, архитектуры баз данных и дополнительных баз данных.

Архитектура всей системы состоит из следующих компонентов: интерфейсов администратора и клиента, веб-сервера, веб-приложения, реляционной базы данных без секционирования и с секционированием таблиц, колоночной базы данных.

Все подсистемы способны поддерживаться и модифицироваться самостоятельно. Взаимодействие подсистем происходит через сетевые интерфейсы.