**1** ОБЗОР ЛИТЕРАТУРЫ

**1.1** Обзор аналогов

На сегодняшний день существует множество приложений, которые обладают сложной структурой. Поддержание работоспособности таких приложений является одной из самых важных и сложных задач для компаний. В связи с этим, возросла актуальность появления helpdesk сервисов. Современный рынок широко представлен web-приложениями, созданными для оказания услуг обратной связи с клиентами, основной целевой аудитории, удовлетворенность которой показывает, насколько важен и актуален продукт компании.

**1.1.1** Uservoice

UserVoice HelpDesk — инструмент поддержки для отслеживания и реагирования на проблемы клиентов. В рамках UserVoice каждая компания может легко оптимизировать процессы, чтобы помочь сотрудникам обеспечить наилучший ход обслуживания клиентов.

Основной особенностью данного продукта является то, что он использует игровую механику для повышения эффективности службы поддержки. Разработчики helpdesk-сервиса UserVoice решили использовать игровую механику для того, чтобы сделать работу сотрудников поддержки более интересной, а их клиентов — более удовлетворенными. В игровой вселенной UserVoice сотрудники службы поддержки соревнуются друг с другом, зарабатывая очки (см. рис. 1.1).

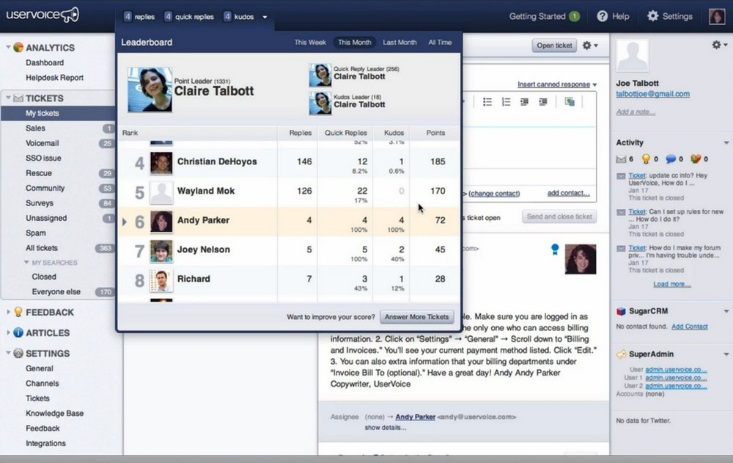


Рисунок 1.1 –­ Экран рейтинга Uservoice

За каждый ответ клиенту сотрудник получает бонус, причем если он ответил быстро (в течении одного часа) - очки утраиваются. А если клиент доволен ответом, он может подарить сотруднику виртуальный приз — звезду, которая стоит много очков.

Также данный сервис обладает ранжированием проблем клиентов. Клиент может увидеть все жалобы и предложения по поводу продукта и проголосовать за них. Самые актуальные будут показаны сверху, что облегчает разработчиках устранять наиболее важные проблемы (см. рис. 1.2).

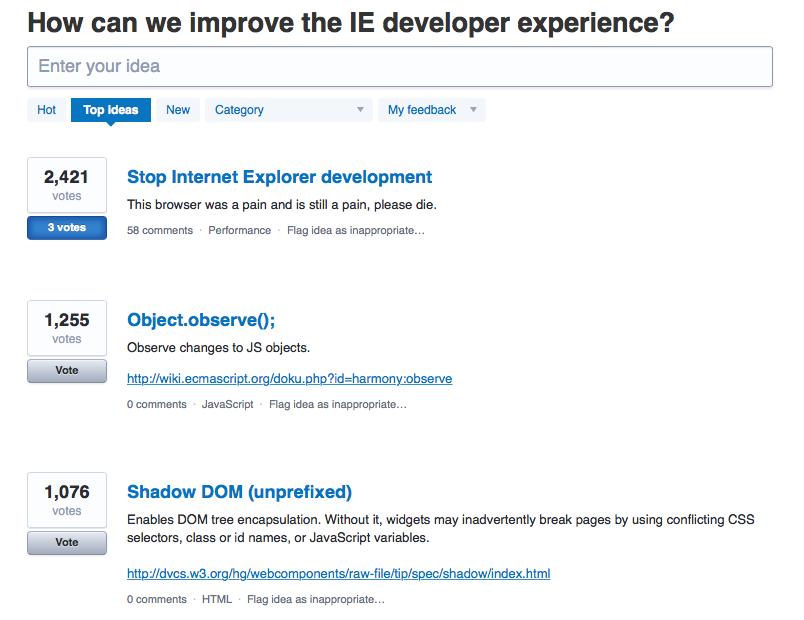


Рисунок 1.2 –­ Экран ранжирования проблем Uservoice

Uservoice представляет собой web-приложение, но также имеются приложение для мобильных устройств под управлением операционной системы Android и IOS. Одним из главных недостатков является то, что данный сервис не является бесплатным, а всего лишь предоставляет бесплатный пробный период на 14 дней. Также отсутствует мультиязычность приложения, что сильно влияет на рынки, где люди не используют английский язык.

**1.1.2** Омнидеск

Омнидеск также, как и UserVoice, является сервисом обратной связи, который предоставляет многоканальную службу поддержки. Данный сервис позволяет покрывать многие популярные и актуальные каналы связи с клиентами: Twitter, Facebook, e-mail и обратную связь. Особенностью данного сервиса является то, что контакты пользователя с различных каналов связи собираются и обрабатываются в аккаунте пользователя на Омнидеск, что позволяет сотрудникам службы поддержки оказывать более качественную и своевременную помощь клиентам (см. рис. 1.3).

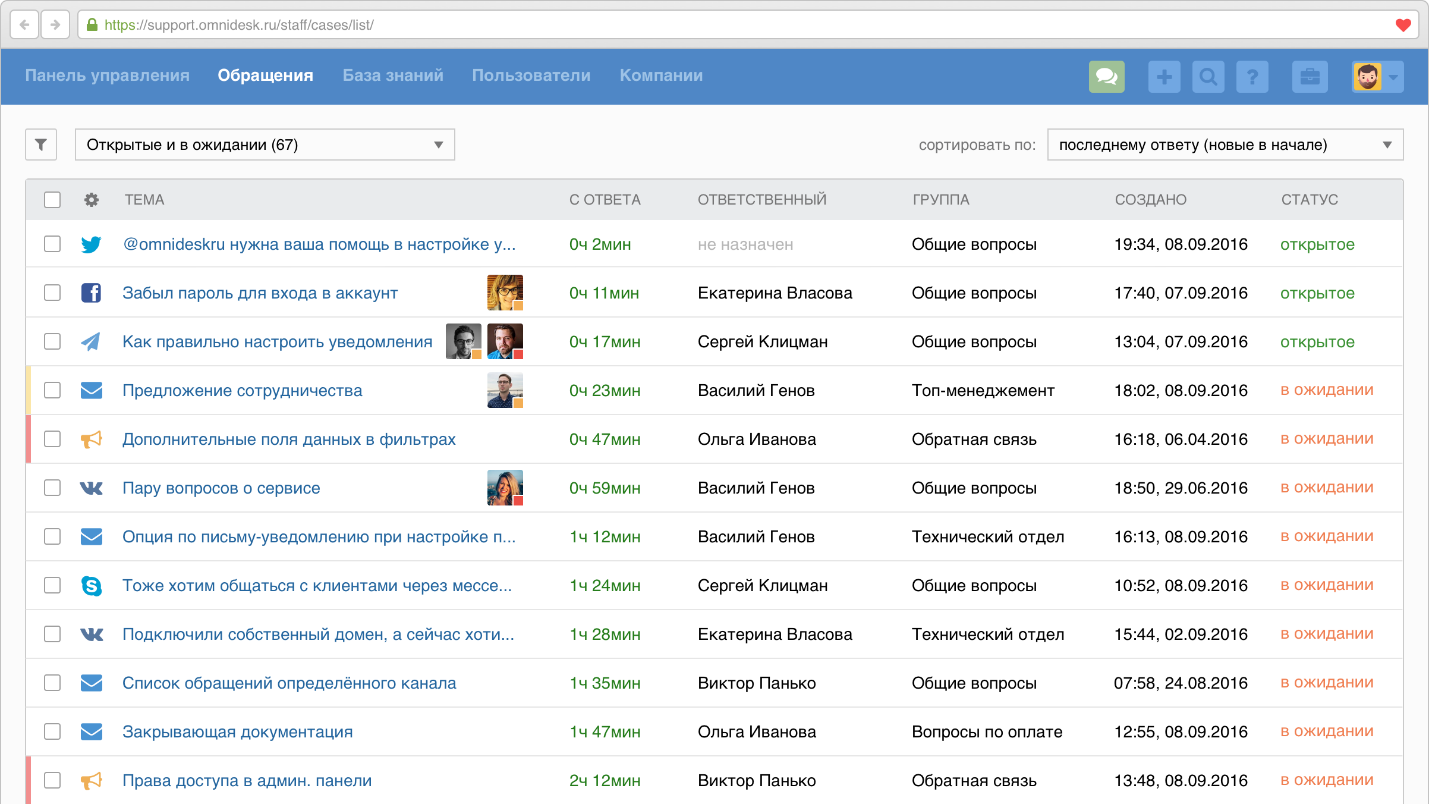


Рисунок 1.3 –­ Экран аккаунта пользователя Омнидеск

Как отмечалось выше, у сервиса Омнидеск присутствует интеграция со многими каналами связи, что также является преимуществом перед сервисом UserVoice. Онлайн система для поддержки клиентов Омнидеск реализована с помощью JIRA — сервисом, которым пользуются практически все компании, имеющие отношение к разработке. Удобство интеграции заключается в том, что она двухсторонняя: сотрудники поддержки работают с JIRA-проблемами (issues) из Омнидеска, а разработчики имеют доступ к обращениям прямо из JIRA. Также есть возможность отправки уведомлений из одного сервиса в другой.

В отличии от Uservoice, в сервисе Омнидеск отсутствует ранжирование проблем, а также игровая механика для людей, ответственных за обратную связь с клиентами. Однако стоит заметить, что в данном сервисе присутствует подобие системы оценок обслуживания клиентов (см. рис. 1.3).

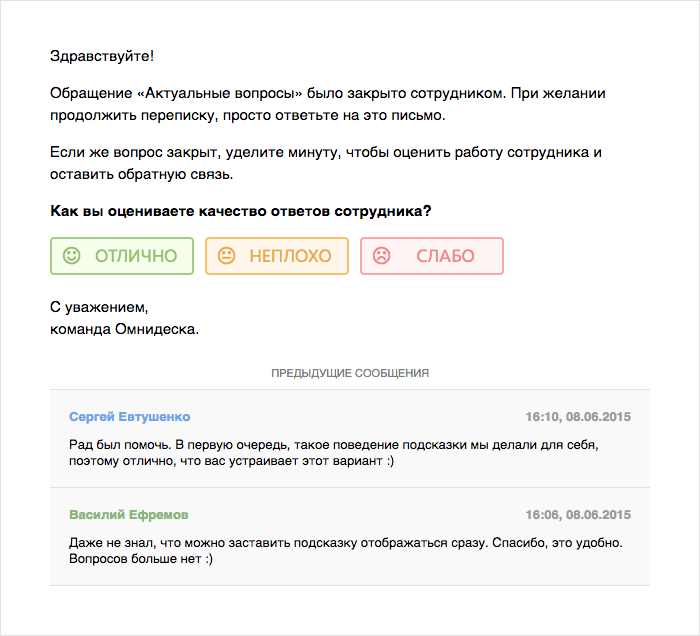


Рисунок 1.4 –­ Экран обслуживания пользователя Омнидеск

Сервис Омнидеск, как и Uservoice, является web-приложением, но не имеется приложений под мобильные операционные системы. Несмотря на интеграцию со многими известными сервисами, данный продукт ориентирован только на русскоязычную аудиторию, что сильно мешает продвижению продукта на другие рынки.

**1.2** Обзор используемых технологий

В данном разделе будут рассмотрены технологии, на которых в дальнейшем будет реализован дипломный проект.

**1.2.1** Выбор технологий клиентской части приложения

Клиентская часть web-приложений существует для создания пользовательского интерфейса, а также для формирования запросов к серверу и обработки ответов от него. На сегодняшний день web-приложения получили большую популярность и активно вытесняют десктопные приложения. С развитием web-приложений появилось множество концепций и технологий по созданию клиентской части такого приложения. Сегодня большой спрос и популярность получил язык JavaScript. Существует множество фреймворков данного языка программирования, которые имеют свои достоинства и недостатки для создания приложений. В данном разделе мы рассмотрим наиболее популярные: AngularJS, ReactJS, BackboneJS.

AngularJS является популярным фреймворком, который широко используется для создания и поддержания сложных веб-приложений. Популярность AngularJS огромна и компании, использующие его, столь же разнообразны: Domino’s Pizza, Ryanair, ITunes Connect, PayPal, Google Checkout. AngularJS имеет открытый исходный код при поддержке Google. AngularJS позиционируется как расширение HTML для создания сложных веб-приложений.

AngularJS — MVC-фреймворк. Он имеет двусторонний дата-биндинг между моделями и представлениями (view). Эта привязка данных позволяет проводить автоматическое обновление с обеих сторон всякий раз, когда происходит изменение данных. Это позволяет создавать многократно используемые компоненты представления, что обеспечивает легкий обмен данными между серверной и клиентской частью.

Данный фреймворк можно использовать при построении сложного web-приложения, нуждающемся в едином модульном фреймворке. Однако, данный фреймворк имеет ряд недостатков. Самый главные недостатки являются в сложности освоения данного фреймворка и отсутствии совместимости между первой и второй версиями. Но в тоже время, огромное количество документации делает AngularJS одним из самым используемых фреймворков в мире.

ReactJS является топовым JavaScript проектом этого года. ReactJS имеет открытый исходный код и развивается в основном с помощью Facebook при участии других крупных технологических компаний. React описывает себя как JavaScript библиотека для создания пользовательских интерфейсов.

В известном паттерне Model-View-Controller React ближе всего к пользователю. Он отвечает за представление данных, получение и обработку ввода пользователя. React — это всего лишь View приложения. React построен на парадигме реактивного программирования. Этот декларативный подход предлагает описывать данные в виде набора утверждений или формул. Изменение одного из параметров ведёт за собой автоматический пересчёт всех зависимостей. ReactJS — это отличный goto-фреймворк для создания простых веб-приложений.

Однако, данный фреймворк имеет свои недостатки. Высокий порог вхождения и отсутствие широкого и разнообразного количества документации, отсутствие множества выработанных практик делают менее привлекательным данный фреймворк.

Backbone — это простой фреймворк, который вписывается в один JavaScript файл. Backbone был разработан Джереми Ашкенасом с помощью CoffeeScript. Backbone особенно популярен среди команд, которые ищут простую структуру для небольших веб-приложений, без применения больших фремворков как Angular или Ember.

Работая с Backbone, данные представляются как модели, которые могут быть созданы, проверены, удалены, и сохранены на сервере. Всякий раз, когда в интерфейсе изменяется атрибуты модели, модель вызывает событие «change» и все представления, которые отображают состояние модели, могут быть уведомлены об изменении атрибутов модели, с тем чтобы они могли отреагировать соответствующим образом — например, перерисовать себя с учетом новых данных. В готовом приложении на Backbone отсутствует необходимость писать код, ищущий элемент с определенным id в DOM и обновлять HTML вручную. При изменении модели представление просто обновит себя самостоятельно.

Backbone не предоставляет структуры. Это всего лишь набор простых инструментов. Для создания структуры необходимо заполнить много пустых мест. Конечно, многие из этих мест заполняются сторонними плагинами, но это значит, что необходимо принять много решений при их выборе. Виды в Backbone напрямую манипулируют DOM, поэтому их сложно тестировать и сложнее повторно использовать.

Выполнив детальный разбор наиболее востребованных технологий клиентской части приложения, а также оценив порог вхождения и наличие обширной и понятной документации, можно увидеть, что для данного дипломного проекта наиболее актуальным и удобным вариантом является фреймворк AngularJS.

**1.2.2** Выбор технологий для разработки серверной части приложения

Существует множество различных языков программирования и технологий для разработки серверной части приложения. В последнее время

широко распространены следующие технологии: фреймворк Spring, реализованный на языке Java, а также Node.js.

Spring Framework — универсальный фреймворк с открытым исходным кодом для Java-платформы. Spring имеет собственную MVC-платформу веб-приложений, которая не была первоначально запланирована. Разработчики Spring решили написать её как реакцию на то, что они восприняли как неудачность конструкции (тогда) популярного Apache Struts, а также других доступных веб-фреймворков. В частности, по их мнению, было недостаточным разделение между слоями представления и обработки запросов, а также между слоем обработки запросов и моделью. Spring MVC является фреймворком, ориентированным на запросы. В нем определены стратегические интерфейсы для всех функций современной запросно-ориентированной системы. Цель каждого интерфейса — быть простым и ясным, чтобы пользователям было легко его заново имплементировать, если они того пожелают. MVC прокладывает путь к более чистому front-end-коду. Все интерфейсы тесно связаны с Servlet API. Эта связь рассматривается некоторыми как неспособность разработчиков Spring предложить для веб-приложений абстракцию более высокого уровня. Однако эта связь оставляет особенности Servlet API доступными для разработчиков, облегчая все же работу с ним.

Spring предоставляет бо́льшую свободу Java-разработчикам в проектировании; кроме того, он предоставляет хорошо документированные и лёгкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба. особенности ядра Spring применимы в любом Java-приложении, и существует множество расширений и усовершенствований для построения веб-приложений на Java Enterprise платформе. По этим причинам Spring приобрёл большую популярность и признаётся разработчиками как стратегически важный фреймворк.

Node.js — программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS) и даже программировать микроконтроллеры (например, tessel и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

Однако данная платформа имеет ряд недостатков. Отсутствуют основные инструменты — существует множество альтернатив, достоинства и недостатки которых не ясны из документации, приходится лично пробовать и выбирать. Очень много заброшенных проектов. Для многих стандартных задач нет какого-то готового и законченного решения. Очень слабая интеграция между инструментами. Приходится при необходимости писать обвертки руками.

Исходя из сравнения вышеозначенных фреймворков видно, что фреймворк Spring наиболее целесообразен для данного дипломного проекта из-за наличия широкой документации и простоты написания серверной части приложений.

**1.2.3** Выбор СУБД

В настоящее время существует множество реляционных баз данных, которые имеют свои достоинства и недостатки, предназначены под различные платформы. В данном разделе рассмотрим наиболее актуальные и подходящие реляционные базы данных, основываясь на технологиях, выбранных для данного дипломного проекта.

SQLite является компактной встраиваемой реляционной базой данных. Слово «встраиваемый» (embedded) означает, что SQLite не использует парадигму клиент-сервер, а предоставляет библиотеку, с которой программа компонуется. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется. SQLite поддерживает динамическое типизирование данных. Возможные типы полей: INTEGER, REAL, TEXT, BLOB.

MySQL является свободной реляционной системой управления базами данных. MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц.

Для данного дипломного проекта можно увидеть, что реляционная база данных MySQL наиболее подходит для данного дипломного проекта, нежели SQLite. MySQL имеет преимущество в следующих компонентах:

* Дополнительные приложения позволяет довольно легко работать с

базами данных.

* MySQL поддерживает большинство функционала SQL.
* Большое количество функций, обеспечивающих безопасность,

которые поддерживается по умолчанию.

* MySQL легко работает с большими объемами данных и легко масштабируется.
* Упрощение стандартов позволяет MySQL значительно увеличить

производительность.

**1.3** Выводы

Данный дипломный проект будет реализован с помощью серверной и клиентской частей. Клиентская часть будет написана с использованием фреймворка AngularJS, реализованном на языке JavaScript. Серверная часть будет написана с использованием фреймворка Spring, реализованном на языке Java. В качестве СУБД была выбрана MySQL.