МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-ХОРАЗМИЙ

ФАКУЛЬТЕТ «КОМПЬЮТЕРНЫЙ ИНЖИНИРИНГ»
КАФЕДРА ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

У Т В Е Р Ж Д А Ю
Зав. кафедрой
_____
«____» _____ 2017 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему

«Создание системы отправки коротких сообщении СМС на основе YII2»

Выполнил: Сайтов А.Ж.

Научный руководитель: к.т.н. Арзымбетов Т.З.

НУКУС - 2017 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

ГЛАВА 1. ОБЗОРНАЯ ЧАСТЬ

1.1 Обзор известных СМС сервисов и СМС платформ.

1.2 Обзор функциональных возможностей фреймворка YII2.

ГЛАВА 2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1 Установка скрипта и настройка модуля отправки СМС на основе YII2.

2.2 Вопросы внедрения и настройка скрипта отправки СМС на примере корпоративного сайта ООО “INFOSYSTEM NUKUS” .

ГЛАВА 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1 Реализация и настройка модуля автоматической рассылки СМС заказчикам компании.

3.2 Применение СМС системы для маркетинга и рекламы компании.

ЗАКЛЮЧЕНИЕ

ЛИТЕРАТУРА

ПРИЛОЖЕНИЕ

ВВЕДЕНИЕ

Актуальность темы: В последнее время SMS-коммуникация привлекает все большее внимание специалистов в связи с различными тенденциями, определяющими развитие современного коммуникативного пространства. Одной из них является увеличение числа информационных каналов и расширение потоков информации, появление интерактивных СМИ. Сейчас активно развиваются персональные средства коммуникации, такие как компьютер, смартфоны и мобильные телефоны, интерес к ним со стороны пользователей неуклонно растет. Не вдаваясь в мельчайшие подробности и в преимущества данной системы , можно привести лишь самые основные которые обезопасят и облегчат нашу повседневную жизнь:

-Аутентификация номера пользователя посредством смс;

-Рассылка СМС в минимальное время большому количеству пользователей мобильных устройств;

- Оплата и получение денежных средств с помощью СМС;

СМС услуги развиваются быстрыми темпами во всем мире и в том числе в нашей стране. Если в 2000 году было отправлено только 17 млд. СМС сообщений, то в 2001 это число составило 250 млд., а в 2004 уже 500 млд. Особенно популярны СМС сообщения в Азии (кроме Японии), Европе, Австралии, и Новой Зеландии.

В мире число людей, использующих смс более чем в два раза превышает число людей, использующих электронную почту. Согласно CTIA, более 28.8 млд. текстовых сообщений отправляется каждый месяц (June 2007, www.ctia.org).

Филиппины известны как «смс-столица мира». Каждый абонент отправляет здесь около 10 СМС в день, тогда как на долю абонента в Великобритании приходится не более 3 СМС в день. В среднем на Филиппинах отправляется 400 млн. текстовых сообщений в день, что составляет примерно 142 млд. в год.

Второе место по отправке СМС занимает Китай (18 млд. СМС отправлено в 2001 г.), что позволило компании China Mobile стать второй компанией в мире по доходности за 2007г.

СМС популярны и в Индии, где многие компании пользуются ими для продвижения своих услуг, информационной поддержки различных мероприятий. С помощью СМС можно заказать авиабилеты или воспользоваться банковским сервисом.

После Азии по СМС популярности лидирует, безусловно, Европа. В 2003, в среднем, 16 млрд. сообщений отправлялось здесь каждый месяц.

Абоненты в Испании отправляют около 50 сообщений в месяц. В Италии, Германии и Великобритании эта цифра составляет около 35–40 СМС в месяц. По данным ассоциации Mobile Data Association (MDA), объединяющей статистику всех без исключения операторов, в 2009 году ежедневно жители Соединенного Королевства обменивались в среднем 265 млн. текстовых сообщений и 1,6 млн. ММС. Всего за год эфир Великобритании пронзили 96,8 млрд. СМС и 600 млн мобильных картинок. Любопытный факт, но во Франции СМС менее популярны (в среднем 20 СМС в месяц на каждого пользователя).

Источник: http://www.ictdata.org/2015/12/philippines-tops-for-sms-usage-in-2014.html

Целью выпускной квалификационной работы является: изученения ряда вопросов и проблем и особенностей создания системы отправки коротких сообщении смс на основе Фреймворка Yii2 . Для достижения поставленной цели необходимо решить следующие задачи:

- анализ известных СМС сервисов и СМС платформ в мировой практике и в Узбекистане;

- изучение функциональных возможностей фреймворка YII2;

- установка скриптов и настройка модулей отправки СМС на основе YII2 и его применение;

- изучение вопросов внедрения и настроек скрипта отправки СМС на примере корпоративного сайта ООО “INFOSYSTEM NUKUS” ;

-частичная реализация системы отправки коротких сообщении СМС на примере корпоративного сайта ООО “INFOSYSTEM NUKUS”;

Материалы в выпускной квалификационной работе располагается в той последовательности, в которой происходило его изучение и обработка.

Данная работа состоит из трёх глав. В первой главе рассматриваются общие теоретические вопросы понятия СМС систем и функциональность работы Фреймворка YII2 . Приводится обзор современных СМС сервисов в мире и в Узбекистане. Также приведен обзор возможностей и преимуществ Фреймворка YES IT IS 2.

Во второй главе изучены вопросы скриптов и настроек модуля отправки СМС на основе YII2 и вопросы внедрения системы отправки СМС на примере корпоративного сайта ООО “INFOSYSTEM NUKUS” .

В третьей главе рассматривается реализация и настройка модуля автоматической рассылки СМС заказчикам компании и разрабатывается система СМС уведомлений о предстоящих и выполненных работах сотрудников.

Также будут рассмотрены применения СМС системы для маркетинга и рекламы компании.

В заключении приводятся результаты, достигнутые в ходе квалификационной работы.

ГЛАВА 1. ОБЗОРНАЯ ЧАСТЬ

В этой главе рассмотрим технологию и историю развития службы коротких сообщении (SMS),а также рассмотрим его разновидности и тарификацию в нашей стране .В подглавах сделаем обзор известных СМС сервисов и СМС платформ и функциональные возможности фреймворка YII2.

1.1 Обзор известных СМС сервисов и СМС платформ.

SMS (СМС — кириллицей) (МФА: [ɛsɛmˈɛs], сокр. от англ. *Short Message Service* — «служба коротких сообщений») — технология, позволяющая осуществлять приём и передачу коротких текстовых сообщений с помощью сотового телефона. входит в стандарты сотовой связи.

Технология

- SMS, как правило, доставляются в течение не более 10 секунд. Отправитель может получать уведомление о доставке сообщения.
- Можно отправить сообщение на выключенный или находящийся вне зоны действия сети телефон. Как только адресат появится в сети, он получит сообщение. Если отправитель получает уведомления о доставке, то таким образом можно зафиксировать момент появления в сети получателя.
- Можно отправить сообщение абоненту, который в данный момент занят разговором.
- С помощью расширенного варианта SMS, называемого EMS, можно отправлять и получать мелодии звонков, пиктограммы и многое другое.
- Особенность технологии такова, что передача SMS почти никак не нагружает сотовую сеть.

Технология SMS поддерживается основными сотовыми сетями (GSM, NMT, D-AMPS, CDMA, UMTS).

Также SMS на телефоны можно отправлять из Интернета и из других сетей (пейджинговых, Фидонет, x.25 и др.) используя специальные программы, универсальные SMS-формы, а также непосредственно SMS-шлюзы мобильных операторов.

История

SMS была создана как составная часть стандарта GSM Phase 1. Впервые идея осуществления сервиса возникла в 1984 г. и была затем реализована группой инженеров, среди которых были: Фридхельм Хиллебранд (Deutsche Telekom), Бернар Жильбер (PTT), Финн Тросбю (Telenor), Кевин Холли (Cellnet), Иэн Харрис (Vodafone) к 1989 году. Впервые система рассылки коротких сообщений была опробована в декабре 1992 года в Великобритании для передачи текста с персонального компьютера на мобильный телефон в сети GSM компании Vodafone, спустя год после появления стандарта GSM на европейском рынке. И содержание первого SMS-сообщения было, в переводе на русский язык, «Счастливого Рождества!»

Длина Сообщении

Текст может состоять из алфавитно-цифровых символов. Максимальный размер сообщения в стандарте GSM — 140 байт (1120 бит). Таким образом, при использовании 7-битной кодировки (латинский алфавит и цифры) можно отправлять сообщения длиной до 160 символов. При использовании 8-битной кодировки (немецкий, французский язык) можно отправлять сообщения длиной до 140 символов. Для поддержки других национальных алфавитов (китайского, арабского, русского и др.) используется 2-байтовая (16-битная) кодировка UCS-2. Таким образом, SMS, написанное кириллицей, не может превышать 70 знаков. Существуют[3] и поддерживаются частью телефонов и восьмибитные кодировки кириллицы — так называемая локальная российская кодировка KOI8-R и Windows-1251. Но при использовании таких кодировок возникают проблемы с совместимостью: как телефон отправителя, так и телефон получателя

сообщения должны быть заранее настроены на «сокращённый набор символов», при такой настройке невозможна отправка сообщений с использованием других алфавитов, кроме кириллицы и латиницы. К тому же поддержка кодировки телефонами несовершенна: в случае переключения телефона на UCS-2 (Юникод) сообщения, сохранённые в восьмибитной кодировке, могут быть испорчены так, что не восстанавливаются даже при обратном переключении. Поэтому даже телефоны, поддерживающие восьмибитную кириллическую кодировку, по умолчанию (то есть без изменения настройки) используют UCS-2.

Таблица-Максимальное количество символов в сегментированном виде

Сегментов	Кол-во символов	
	Кириллица	Латиница
1	70	160
2	134	306
3	201	459
4	268	612

Источник: https://ru.wikipedia.org/wiki/SMS

В стандарте также предусмотрена возможность отправлять сегментированные сообщения. В таких сообщениях в заголовке пользовательских данных (UDH) помещается информация о номере сегмента сообщения и общем количестве сегментов. Максимальная длина сегмента при этом уменьшается за счет этого заголовка. Как правило, каждый сегмент тарифицируется как отдельное сообщение. Сегментирование поддерживают почти все современные телефоны, но часто в телефонах вводится ограничение на количество сегментов в сообщении. Телефон, который не поддерживает сегментирование, отображает каждый сегмент как отдельное сообщение.

Некоторые абоненты сотовых сетей предпочитают писать SMS на родном языке, используя латинские буквы (см.: транслит). Первоначально

это было обусловлено отсутствием поддержки кириллицы и других национальных алфавитов телефонными аппаратами; а с широким распространением русифицированных телефонов — привычкой, а также тем, что на латинице можно писать более длинные SMS (160 вместо 70 символов на кириллице) за те же деньги. Например: *Ura! Ya napisal pro SMS v Diplomnoy*. При этом фактическая экономия меньше, чем 160/70, так как те буквы кириллицы, у которых отсутствуют аналоги в латинице, приходится заменять буквосочетаниями не менее чем из двух букв.

В англоязычных странах для экономии символов в SMS часто используют аббревиатуры, пропуски гласных, а также обозначают слова и слоги схожими по звучанию цифрами и буквами. Например, *«C u l8r»* вместо *«See you later»*. Со временем у пользователей сложился целый определённый SMS-язык, имеющий варианты у носителей разных национальных языков и письменностей.

Тарификация SMS в Узбекистане

В Узбекистане, как правило, оплачивается отправка SMS , входящие SMS бесплатны. Стоимость отправки и получения SMS в роуминге зависит от роумингового соглашения и может меняться в зависимости от сети пребывания абонента. В последние годы почти все операторы мобильной связи ввели разную стоимость отправки SMS: внутри домашнего региона и на все внутрисетевые страны, за пределы домашнего региона внутри страны, за рубеж.

Отправка сообщений через официальные сайты сотовых операторов, как правило, бесплатна.

С 2013 года многие сотовые операторы добавляют пакет смс-сообщений в тарифные планы с абонентской платой дополнительно к звонкам и интернет-трафику. Размер таких пакетов может варьироваться от оператора к оператору и составлять от 50 сообщений в пакете в месяц до 2000-3000 сообщений в более дорогих и топовых тарифах.

Кроме того у операторы предоставляют возможность приобрести пакеты смс в не зависимости от основного тарифа.

Бесплатная отправка SMS через Интернет

В настоящее время возможно относительно анонимно и бесплатно отправлять сообщения через официальные сайты операторов сотовой связи. Под анонимностью в данном случае понимается то, что информация об отправителе неизвестна получателю. Веб-сайт, с помощью которого отправлено сообщение, может записать IP-адрес и многие другие данные о компьютере отправителя. Есть также множество сайтов-сервисов, на которых пользователю нужно ввести лишь номер мобильного телефона, а программа сама перенаправит посетителя на сайт нужного сотового оператора.

SMS и Мобильные Телефоны

Большинство современных мобильных телефонов всех стандартов позволяют использовать SMS в полном объёме.

Для того чтобы телефон мог отправлять SMS, необходимо указать номер SMS-центра (SMSC) оператора мобильной связи. В подавляющем большинстве случаев этот номер уже записан на SIM-карте и настраивать его вручную не нужно.

Входящие SMS сохраняются в списке входящих сообщений, где они могут быть просмотрены. Некоторые модели телефонов хранят этот список на SIM-карте, и потому имеют ограничения на количество хранящихся сообщений (несколько десятков). Современные модели хранят список сообщений в памяти телефона, и количество сообщений ограничено только объёмом памяти телефона. Отправленные сообщения сохраняются в списке отправленных сообщений, также существуют отдельные списки для неотправленных сообщений и для черновиков.

В смартфонах и коммуникаторах SMS иногда хранятся в общем почтовом ящике, там же, где сообщения электронной почты и MMS.

Для набора SMS на мобильном телефоне, как правило, используется цифровая клавиатура телефона. Набор осуществляется либо путём

последовательных нажатий для выбора нужной буквы, либо при помощи какой-либо системы предиктивного набора, типа T9 или iTAP. Некоторые модели телефонов имеют алфавитно-цифровую клавиатуру, что существенно облегчает набор. В смартфонах и коммуникаторах также может использоваться экранная клавиатура.

Flash-SMS

lash-SMS — это SMS-сообщение, сразу отображаемое при получении на экране телефона, в разных моделях телефонов по-разному. Обычно Flash-SMS не сохраняются в памяти телефона или на SIM-карте, однако на некоторых телефонах возможно сохранение.

В стандарте GSM Flash-SMS принадлежит к Class 0.

Не все сети GSM и не все мобильные телефоны поддерживают Flash-SMS.

Некоторые телефоны поддерживают приём «мигающих сообщений» (англ. Blinking Messages), в том числе Flash-SMS. Мигающим может быть как всё сообщение, так и его часть.

Для передачи Flash-SMS сообщения в соответствии со стандартом GSM используются следующие Data Coding Scheme (DCS) :

- 0x10 SMS GSM 7-bit default alphabet flash-сообщение
- 0x18 SMS UCS2(16 bit) flash-сообщение

Для латиницы возможно также использование DCS = 0xF0 (GSM 7 bit)

Современный бизнес невозможно представить без активного использования электронных писем и смс-сообщений. Используя массовые email и смс-рассылки, можно легко оповещать клиентов о новых предложениях, акциях, распродажах. Индивидуальные сообщения обычно используются для оповещения клиентов о проведении банковских операций или о ходе выполнения заказа интернет-магазином.

SMS-шлюз — интерфейс, который позволяет отправлять и получать SMS-сообщения без использования мобильного телефона. При этом SMS-сообщения преобразуются в сообщения электронной почты или HTTP-

запросы и обратно. Сообщение, отправляемое через подобный шлюз, может быть бесплатным для отправителя; однако возможны технические ограничения, такие, как ограничение числа отправляемых с одного компьютера сообщений в сутки.

Часто на сайтах мобильных операторов размещается форма бесплатной отправки смс на номера своих абонентов. При этом стоит ограничение на количество отправляемых сообщений с одного компьютера, и смс приходит обычно с номера сервисной службы оператора.

Выбирая рассылочный сервис, прежде всего необходимо убедиться, что предоставляемый функционал соответствует вашему проекту. Причём здесь имеются в виду не только возможности по отправке сообщений. Не менее важное значение имеет удобство загрузки в систему ваших рассылочных баз, возможности встроенного редактора сообщений, полнота предоставляемых статистических отчетов и т. д.

Важное значение имеет ценовая политика сервиса. Необходимо убедиться, что в системе имеется тарифный план, подходящий вам для запуска вашего проекта, и что линейка тарифных планов соответствует вашим планам развития бизнеса.

Для многих проектов важно не только иметь возможность управлять рассылками через веб-интерфейс, но и интегрировать рассылочный функционал в свои интернет-проекты. В этом случае необходимо убедиться, что предоставляемый API позволяет решить ваши задачи и хорошо документирован. Также стоит иметь в виду, что многие сервисы предлагают готовые модули для интеграции с наиболее популярными CMS.

Поскольку вы будете загружать в рассылочный сервис свою клиентскую базу, важно убедиться, что выбранный сервис уделяет должное внимание безопасности работы: для доступа используется протокол https или VPN-соединение, в работе сервиса используется ПО известных поставщиков, сервера располагаются в надёжных дата-центрах и в штате имеется квалифицированный персонал.

Наиболее известные разновидности: WEB2SMS, SMS2EMAIL, SMS2Skype.

Web2SMS: Кроме служб, доступных через веб-интерфейс, некоторые компании предоставляют службы, доступные посредством приложений, или даже наборы функций API для работы с SMS.

SMS2Email: Эта разновидность SMS-шлюзов позволяет посылать электронную почту через мобильный телефон. Большинство провайдеров мобильной связи в США предоставляют эту возможность. В Великобритании служба M-Mail, предоставляемая компанией Connectotel, позволяет посылать электронные сообщения даже из сетей провайдеров, которые прямо не поддерживают эту услугу.

Существуют шлюзы, позволяющие посылать сообщения SMS на обычные стационарные телефоны. В этом случае SMS автоматически преобразуется в голосовое сообщение. Иногда адресату предоставляется возможность отправить голосовую почту на мобильный телефон, с которого пришло SMS. Существуют и так называемые корпоративные SMS-шлюзы, позволяющие посредством SMS получать доступ к корпоративным приложениям или базам данных.

Приведём несколько конкретных примеров сервисов по рассылке смс-сообщений:

- Сервис СМС-рассылок компании Intis.SMS может с успехом использоваться для решения различных задач: для миллионных рассылок федеральных компаний с выборками по географии, времени, полу, возрасту, для сверхскоростных уведомлений о банковских транзакциях, для организации смс-сервиса в крупных интернет-проектах, для рассылки персональных уведомлений о спецпредложениях магазинов и служб такси. Разработчики сервиса особое внимание уделяют качеству и надежности: сервера располагаются в лучших дата-центрах Москвы, Санкт-Петербурга и Германии, имеющих быстрые и широкие каналы связи до СМС шлюзов операторов связи. Значительное внимание уделяется вопросам

конфиденциальности и защите информации. Клиенты могут работать через безопасное VPN-соединение с использованием протокола SSL с 256-битным шифрованием канала, все данные касающиеся работы шифруются и резервируются. Для интеграции системы с вашими интернет-проектами предоставляется удобный и хорошо документированный API. Для интеграции с 1С-Битрикс имеется готовый модуль.

- Сервис СМС-рассылок SemySMS обеспечивает возможность проводить рекламные смс-рассылки и оповещения о различных акциях и скидках. Управление сервисом возможно как через веб-интерфейс, так и через API. Особенностью сервиса является то, что сообщения отправляются не с смс-шлюзов, а с ваших собственных Android-телефонов, на которые устанавливается необходимое для этого приложение. Поэтому оплату за отправленные СМС вы производите не сервису SemySMS, а непосредственно оператору мобильной связи по ценам соответствующим тарифному плану вашей сим-карты. Соответственно на неограниченных тарифах вы получаете фозможность за фиксированную ежемесячную плату отправлять произвольное количество смс-сообщений. Начинать пользоваться сервисом можно сразу же после регистрации в системе и установки приложения на ваш телефон. Сервис обеспечивает удобную работу со списками контактов (импорт из файлов Excel, объединение в группы и т. д.), отслеживание статистики и статуса отправляемых смс-сообщений, информативные отчёты по смс-рассылкам. Имеется модуль для интеграции в интернет-проекты на 1С-Битрикс.
- Сервис рассылок SendPulse — единая платформа для email-рассылок, СМС, push-уведомлений и транзакционных писем. Сервис email-рассылок предоставляет всё необходимое для эффективного email-маркетинга: удобную загрузку адресных баз, функциональный конструктор шаблонов писем, умную персонализацию писем, возможности сплит-тестирования, просмотра подробной статистики каждой кампании. Имеется большая коллекция готовых шаблонов писем для различных сфер бизнеса.

Предлагается услуга email маркетинга «Под ключ», в рамках которой вы ставите цели и задачи, а остальное делает команда SendPulse. Сервис предоставляет также возможность усилить ваш email-маркетинг дополнительными каналами коммуникации — push-уведомлениями и смс. Push-уведомления являются новым каналом и несомненно заинтересуют подписчиков. Их можно быстро и легко настроить для вашего сайта. Смс-рассылки можно интегрировать в онлайн-проекты с помощью хорошо документированного API. Имеются готовые модули для интеграции с CRM, CMS, SaaS-платформами, и поддержка протоколов SMPP, HTTP/HTTPS, email2sms. Сервис рассылок SendPulse насчитывает более 200 тысяч пользователей.

- SMSintel — это интеллектуальный сервис массовой рассылки смс в Москве и регионах России. Уникальность сервиса заключается в наличии готовых решений для маркетологов (SMS-Купон, смс-поздравления и уведомления, смс-отзывы, смс-опросы) и технических специалистов (смс-уведомления об e-mail, обработчик писем для пересылки их содержания или темы по смс). Сервис также характеризует удобная работа с клиентскими базами. Имеется возможность загрузки контактов из файлов в форматах MS Excel, MS Word, OpenOffice, CSV и т. д. и автоматическое удаление дубликатов. Предоставляются готовые модули для интеграции со сторонними платформами: 1C-Bitrix, LPGenerator.ru, AmoCRM, UmiCMS и др. Предоставляется детальная статистика о проведённых рассылках, выгружается в MS Excel. Можно пользоваться таймером отправки и указывать время начала и окончания рассылки. Сервис отличает высокая скорость отправки сообщений: более 6000 сообщений в минуту и привлекательные цены на основные и дополнительные услуги. Услугами SMSintel пользуются более 25000 клиентов.

В Узбекистане также существуют несколько таких сервисов. Один из этих сервисов центр UZINFOCOM, специализирующийся на оказании различным компаниям и организациям содействия во внедрении и

интеграции информационных систем и ресурсов, обеспечении их информационной безопасности, развитии интерактивных услуг в начале 2011 года запустил собственную SMS-платформу для организации SMS-услуг и сервисов.

Проект SMS-Gateway позволяет предоставлять услуги и сервисы посредством SMS-сообщений.

Схема взаимодействия SMS-платформы выглядит следующим образом: «Мобильные клиенты» --> Мобильные операторы» --> «SMS-платформа» --> «Сервисы и приложения».

Источник http://smsg.uz

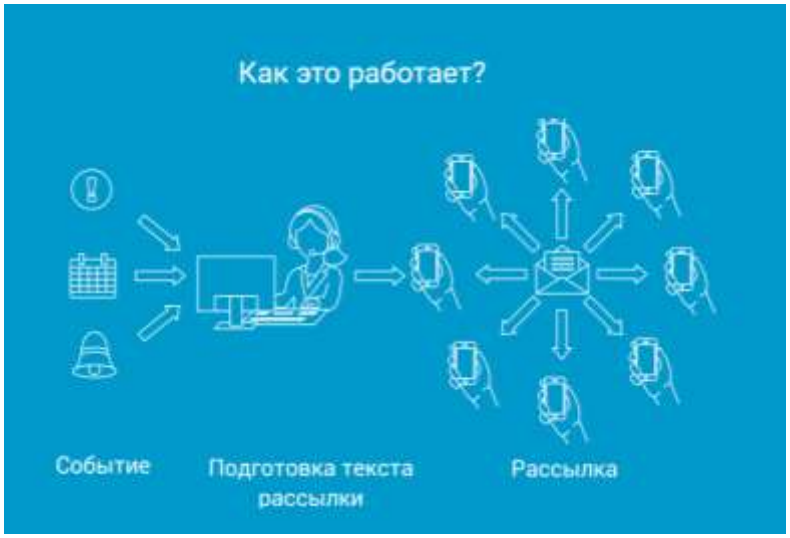
Эта SMS-платформа имеет целый набор преимуществ. К примеру, это возможность интеграции с системой единой авторизации ID.UZ, постоянная доступность и работоспособность платформы, взаимодействие со всеми мобильными операторами Узбекистана и многое другое.

К основным возможностям платформы можно отнести:

- Отправка и прием СМС-сообщений от всех мобильных операторов РУз;
- Реализации подключения через шифрованный протокол https;
- Быстрая скорость работы и своевременная доставка СМС-сообщений до абонентов (100 СМС-сообщений за 5 секунд).

И еще один сервис смс-рассылок в Узбекистане это Bedana.uz. Это сервис, созданный на базе платформы SMS-Gateway, позволяющий организовывать различные SMS-рассылки по собственной базе. Bedana

может быть полезна как в компаниях малого и среднего бизнеса, так и в больших учреждениях и корпорациях, заинтересованных в организации SMS-уведомлений своих сотрудников и/или клиентов посредством SMS.



Источник:http://Bedana.uz

Возможности сервиса

Bedana предоставляет своим пользователям специальный персональный кабинет, где можно собрать все необходимые контакты и сгруппировать их по необходимым критериям (начальники, подчиненные, клиенты и т.д.). При необходимости распространения какой-либо информации, подготовленный текст рассылается при помощи сервиса выбранным пользователям и группам.

В итоге, в один миг ваше сообщение может быть расчирикано неограниченному количеству мобильных абонентов.

Скорость рассылаемых SMS-сообщений в рамках сервиса – 20 в секунду!

Где Bedana может быть полезна?

Например, в магазин поступила новая коллекция товара определенной категории – заходите в персональный кабинет и рассылаете новость об этом группе контактов, которую вы заранее отсортировали по интересам.

Если же вы руководитель большой компании и вам необходимо срочно провести собрание с начальниками отделов, а их не менее 20ти. Не нужно тратить время секретаря на обзванивание каждого из них, нужно лишь выбрать группу контактов «начальники отделов» и сообщение о собрании расчирикается именно им. Именно такой подход будет реализован в главе три.

1.2 Обзор функциональных возможностей фреймворка YII2.

В наше время стало легко создавать сайты благодаря появлению фреймворков. Фремворк это - программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Для данной дипломной работы был выбран фреймворк YII2, поскольку он сейчас наиболее современный в плане дизайна и по функциональности, об этом фреймворке будет рассказано в этой части данной работы.

Yii – это высокопроизводительный компонентный PHP фреймворк, предназначенный для быстрой разработки современных веб приложений. Слово Yii (произносится как Йи [ji:]) в китайском языке означает «простой и эволюционирующий». Также Yii может расшифровываться как акроним Yes It Is!

Yii – это универсальный фреймворк и может быть задействован во всех типах веб приложений. Благодаря его компонентной структуре и отличной поддержке кэширования, фреймворк особенно подходит для разработки таких крупных проектов как порталы, форумы, CMS, магазины или RESTful-приложения.

Сравнение Yii с другими фреймворками

Как и многие другие PHP фреймворки, для организации кода Yii использует архитектурный паттерн MVC (Model-View-Controller).

Yii придерживается философии простого и элегантного кода не пытаясь усложнять дизайн только ради следования каким-либо шаблонам проектирования.

Yii является full-stack фреймворком и включает в себя проверенные и хорошо зарекомендовавшие себя возможности, такие как ActiveRecord для реляционных и NoSQL баз данных, поддержку REST API, многоуровневое кэширование и другие.

Yii отлично расширяем. Вы можете настроить или заменить практически любую часть основного кода. Используя архитектуру расширений легко делиться кодом или использовать код сообщества.

Одна из главных целей Yii – производительность.

Yii — не проект одного человека. Он поддерживается и развивается сильной командой и большим сообществом разработчиков, которые ей помогают. Авторы фреймворка следят за тенденциями веб разработки и развитием других проектов. Наиболее подходящие возможности и лучшие практики регулярно внедряются в фреймворк в виде простых и элегантных интерфейсов.

Структура приложения

Yii приложения организованы согласно шаблону проектирования модель-представление-контроллер (MVC).Модели представляют собой данные, бизнес логику и бизнес правила; представления отвечают за отображение информации, в том числе и на основе данных, полученных из моделей; контроллеры принимают входные данные от пользователя и преобразовывают их в понятный для моделей формат и команды, а также отвечают за отображение нужного представления.

Кроме MVC, Yii приложения также имеют следующие сущности:

входные скрипты: это PHP скрипты, которые доступны напрямую конечному пользователю приложения. Они ответственны за запуск и обработку входящего запроса;

приложения: это глобально доступные объекты, которые осуществляют корректную работу различных компонентов приложения и их координацию для обработки запроса;

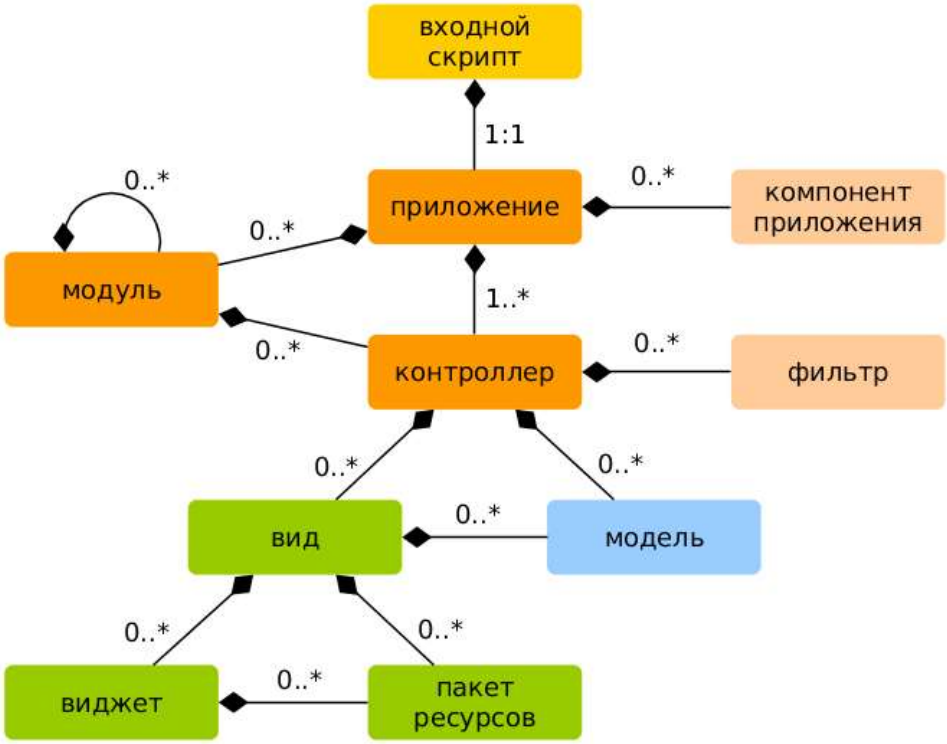
компоненты приложения: это объекты, зарегистрированные в приложении и предоставляющие различные возможности для обработки текущего запроса;

модули: это самодостаточные пакеты, которые включают в себя полностью все средства для MVC. Приложение может быть организованно с помощью нескольких модулей;

фильтры: это код, который должен быть выполнен до и после обработки запроса контроллерами;

виджеты: это объекты, которые могут включать в себя представления. Они могут содержать различную логику и быть использованы в различных представлениях.

Ниже на диаграмме представлена структурная схема приложения:



Источник: ***https://yiiframework.com.ua/ru/doc/guide/2/structure-overview/***

Обработка запросов

Все запросы, обрабатываемые Yii приложением, проходят подобный путь.

Пользователь создает запрос ко входному скрипту web/index.php.

Входной скрипт загружает конфигурацию и создает экземпляр приложения для обработки запроса.

Приложение определяет запрошенный маршрут при помощи компонента request.

Приложение создает экземпляр контроллера для обработки запроса.

Контроллер создает экземпляр действия и выполняет фильтры для этого действия.

При неудачном выполнении любого фильтра, действие не выполняется.

При успешном выполнении всех фильтров, выполняется действие.

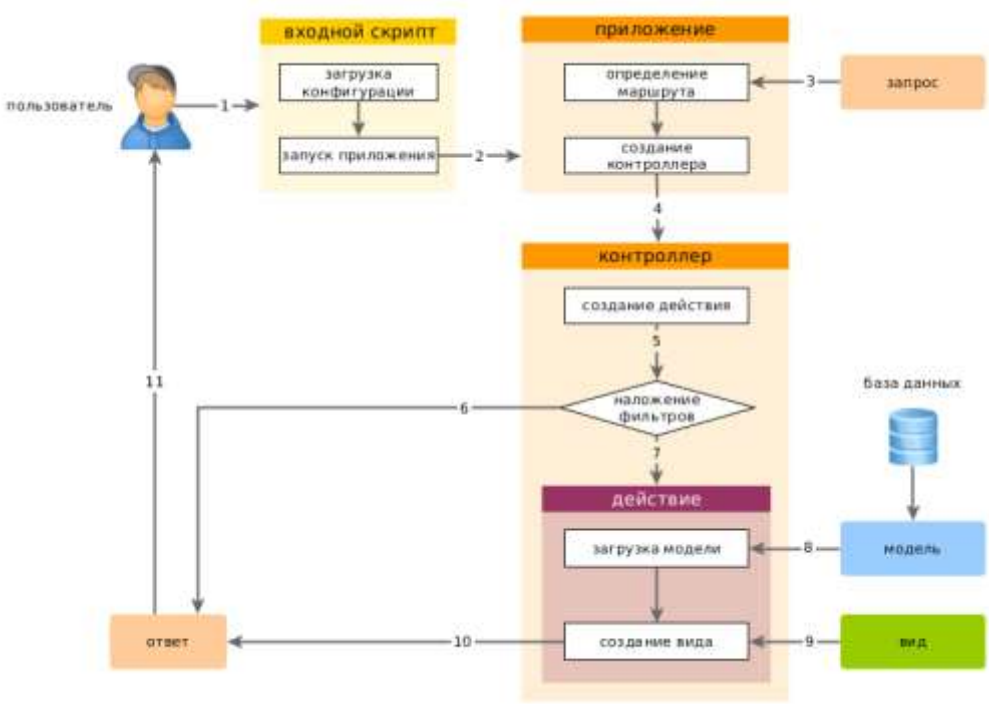
Действие загружает модель данных, возможно, из базы данных.

Действие рендерит представление и передает ему модель данных.

Результат рендеринга передается в компонент приложения response.

Компонент response посылает готовые данные пользователю.

Ниже представлена диаграмма обработки запроса приложением.



Источник: https://yiiframework.com.ua/ru/doc/guide/2/structure-overview/

Компоненты — это главные строительные блоки приложений основанных на Yii. Компоненты наследуются от класса yii\base\Component или его наследников. Три главные возможности, которые компоненты предоставляют для других классов:

- Свойства.
- События.
- Поведения.

Как по отдельности, так и вместе, эти возможности делают классы Yii более простыми в настройке и использовании. Например, пользовательские компоненты, включающие в себя виджет выбора даты, могут быть использованы в представлении для генерации интерактивных элементов выбора даты:

```
use yii\jui\DatePicker;

echo DatePicker::widget([
    'language' => 'ru',
```

```
    'name'  => 'country',
    'clientOptions' => [
        'dateFormat' => 'yy-mm-dd',
    ],
]);
```

Свойства виджета легко доступны для записи потому, что его класс унаследован от класса yii\base\Component.

Компоненты — очень мощный инструмент. Но в то же время они немного тяжелее обычных объектов, потому что на поддержку событий и поведений тратится дополнительные память и процессорное время. Если ваши компоненты не нуждаются в этих двух возможностях, вам стоит унаследовать их от yii\base\Object, а не от yii\base\Component. Поступив так, вы сделаете ваши компоненты такими же эффективными, как и обычные PHP объекты, но с поддержкой свойств.

При наследовании ваших классов от yii\base\Component или yii\base\Object, рекомендуется следовать некоторым соглашениям:

- Если вы переопределяете конструктор, то добавьте последним аргументом параметр $config и затем передайте его в конструктор предка.
- Всегда вызывайте конструктор предка в конце вашего переопределенного конструктора.
- Если вы переопределяете метод yii\base\Object::init(), убедитесь, что вы вызываете родительскую реализацию этого метода в начале вашего метода init().

Пример:

```
<?php

namespace yii\components\MyClass;

use yii\base\Object;

class MyClass extends Object
{
```

```
    public $prop1;
    public $prop2;

    public function __construct($param1, $param2, $config = [])
    {
        // ... инициализация происходит перед тем, как будет применена конфигур
ация.

        parent::__construct($config);
    }

    public function init()
    {
        parent::init();

        // ... инициализация происходит после того, как была применена конфигура
ция.
    }
}
```

Способ инициализации через вызов Yii::createObject() выглядит более сложным. Но в то же время он более мощный из-за того, что он реализован на самом верху контейнера внедрения зависимостей.

Жизненный цикл объектов класса yii\base\Object содержит следующие этапы:

1. Предварительная инициализация в конструкторе. Здесь вы можете установить значения свойств по умолчанию.
2. Конфигурация объекта с помощью $config. Во время конфигурации могут быть перезаписаны значения свойств по умолчанию, установленные в конструкторе.
3. Конфигурация после инициализации в методе init(). Вы можете переопределить этот метод, для проверки готовности объекта и нормализации свойств.
4. Вызов методов объекта.

Первые три шага всегда выполняются из конструктора объекта. Это значит, что если вы получите экземпляр объекта, он уже будет проинициализирован и готов к работе.

Свойства

В PHP, переменные-члены класса называются *свойства*. Эти переменные являются частью объявления класса и используются для хранения состояния объектов этого класса (т.е. именно этим отличается один экземпляр класса от другого). На практике вам часто придётся производить чтение и запись свойств особым образом. Например, вам может понадобиться обрезать строку при её записи в поле label. Для этого вы *можете* использовать следующий код:

$object->label = trim($label);

Недостатком приведённого выше кода является то, что вам придется вызывать функцию trim() во всех местах, где вы присваиваете значение полю label. Если в будущем понадобится производить еще какие-либо действие, например преобразовать первую букву в верхний регистр, вам придётся изменить каждый участок кода, где производится присваивание значения полю label. Повторение кода приводит к ошибкам и его необходимо избегать всеми силами.

Что бы решить эту проблему, в Yii был добавлен базовый класс yii\base\Object который реализует работу со свойствами через геттеры и сеттеры. Если вашему классу нужна такая возможность, необходимо унаследовать его от yii\base\Object или его потомка.

Почти все внутренние классы Yii наследуются от yii\base\Object или его потомков. Это значит, что всякий раз, когда вы встречаете геттер или сеттер в классах фреймворка, вы можете обращаться к нему как к свойству.

Геттер — это метод, чьё название начинается со слова get. Имя сеттера начинается со слова set. Часть названия после get или set определяет имя свойства. Например, геттер getLabel() и/или сеттер setLabel() определяют свойство label, как показано в коде ниже:

namespace app\components;

use yii\base\Object;

```
class Foo extends Object
{
   private $_label;

   public function getLabel()
   {
      return $this->_label;
   }

   public function setLabel($value)
   {
      $this->_label = trim($value);
   }
}
```

В коде выше геттер и сеттер реализуют свойство label, значение которого хранится в private свойстве _label.

Свойства, определенные с помощью геттеров и сеттеров, можно использовать как обычные свойства класса. Главное отличие в том, что когда происходит чтение такого свойства, вызывается соответствующий геттер, при присвоении значения такому свойству запускается соответствующий сеттер. Например:

```
      // Идентично вызову $label = $object->getLabel();
$label = $object->label;

// Идентично вызову $object->setLabel('abc');
$object->label = 'abc';
```

Свойство, для которого объявлен только геттер без сеттера, может использоваться только для чтения. Попытка присвоить ему значение вызовет InvalidCallException. Точно так же, свойство для которого объявлен только сеттер без геттера может использоваться только для записи. Попытка получить его значение так же вызовет исключение. Свойства, предназначенные только для чтения, встречаются не часто.

При определении свойств класса при помощи геттеров и сеттеров нужно помнить о некоторых правилах и ограничениях:

- Имена таких свойств регистронезависимы. Таким образом, $object->label и $object->Label — одно и то же. Это обусловлено тем, что имена методов в PHP регистронезависимы.
- Если имя такого свойства уже используется переменной-членом класса, то последнее будет иметь более высокий приоритет. Например, если в классе Foo объявлено свойство label, то при вызове $object->label = 'abc' будет напрямую изменено значение свойства label. А метод setLabel() не будет вызван.
- Свойства, объявленные таким образом, не поддерживают модификаторы видимости. Это значит, что объявление геттера или сеттера как public, protected или private никак не скажется на области видимости свойства.
- Свойства могут быть объявлены только с помощью не статичных геттеров и/или сеттеров. Статичные методы не будут обрабатываться подобным образом.
- Обычный вызов property_exists() не работает для магических свойств. Для них необходимо использовать canGetProperty() или canSetProperty().

Возвращаясь к проблеме необходимости вызова функции trim() во всех местах, где присваивается значение свойству label, описанной в начале этого руководства, функцию trim() теперь необходимо вызывать только один раз — в методе setLabel(). При возникновении нового требования о возведение первой буквы в верхний регистр, можно быстро поправить метод setLabel() не затрагивая остальной код. Эта правка будет распространяться на все присвоения значения свойству label.

События

События - это механизм, внедряющий элементы собственного кода в существующий код в определенные моменты его исполнения. К событию можно присоединить собственный код, который будет выполняться автоматически при срабатывании события. Например, объект, отвечающий за

почту, может инициировать событие messageSent при успешной отправке сообщения. При этом если нужно отслеживать успешно отправленные сообщения, достаточно присоединить соответствующий код к событию messageSent.

Для работы с событиями Yii использует базовый класс yii\base\Component. Если класс должен инициировать события, его нужно унаследовать от yii\base\Component или потомка этого класса.

Подытоживая рассмотрим самые основные функциональные возможности фреймворка YII2:

- Высокая производительность относительно других фреймворков, написанных на PHP
- Парадигма Модель-представление-контроллер
- Интерфейсы DAO и ActiveRecord для работы с базами данных (PDO)
- Поддержка интернационализации
- Кэширование страниц и отдельных фрагментов
- Перехват и обработка ошибок
- Ввод и валидация форм
- Аутентификация и авторизация (RBAC и ACL)
- Использование AJAX и интеграция с jQuery. Со второй версии добавлена поддержка Bootstrap,
- Генерация базового PHP-кода для CRUD-операций (скаффолдинг)
- Поддержка тем оформления для их лёгкой смены
- Возможность подключения сторонних библиотек
- Миграции базы данных
- Автоматическое тестирование
- Поддержка REST (добавлена со второй версии)

ГЛАВА 2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1 Установка скрипта и настройка модуля отправки СМС на основе YII2.

В этой главе будут рассмотрены теоретические вопросы скриптов и модулей, а также будет короткое ознакомление с корпоративным сайтом “INFO SYSTEM NUKUS” куда будет внедрено система отправки коротких сообщении.

И так что такое скрипт? Скрипт - (язык сценариев англ. *scripting language*) — высокоуровневый язык сценариев (англ. *script*) — кратких описаний действий, выполняемых системой. Разница между программами и сценариями довольно размыта. Сценарий — это программа, имеющая дело с готовыми программными компонентами. В более узком смысле под скриптовым языком может пониматься специализированный язык для расширения возможностей командной оболочки или текстового редактора и средств администрирования операционных систем.

Классификация скриптов.

Языки программирования вообще и сценарные языки в частности могут быть классифицированы множеством различных способов.

В плане быстродействия скриптовые языки можно разделить на языки динамического разбора (sh, COMMAND.COM) и предварительно компилируемые (Perl). Языки динамического разбора считывают инструкции из файла программы минимально требующимися блоками, и исполняют эти блоки, не читая дальнейший код. Предкомпилируемые языки транслируют всю программу в байт-код и затем исполняют его. Некоторые скриптовые языки имеют возможность компиляции программы «на лету» в машинный код (т. н. JIT-компиляция).

По применению языки можно грубо разделить на три типа:

- командно-сценарные;

- прикладные сценарные;
- универсальные сценарные.

Командно-сценарные языки.

Появились ещё в 1960-х годах для управления заданиями в операционных системах. Из языков того времени наиболее известен JCL для OS/360. В этот класс входят языки пакетной обработки (англ. batch language) и языки командных оболочек, например sh, csh для Unix. Эти языки чаще всего используются в пакетном режиме обработки.

- JCL
- sh
- bash
- csh
- ksh
- Pilot]
- REXX
- AppleScript
- COMMAND.COM и cmd.exe
- VB Script
- PowerShell

Например, язык AppleScript операционной системы MacOS имеет редактор Script Editor, который позволяет записывать действия по мере их выполнения пользователем в системе в файл сценария (текстовый файл) и оформлять в виде исполняемой программы. Такой подход позволяет составлять простейшие сценарии непрограммирующим пользователем.

Прикладные сценарные языки.

Сценарные языки этого типа начали появляться в 1980-е годы, когда на промышленных персональных компьютерах стало возможным интерактивное общение с ОС. В клиент-серверной архитектуре такие языки работали в клиентской части программного обеспечения.

- AutoLISP
- ECMAScript и его диалекты (JScript, JavaScript, ActionScript)
- Emacs Lisp
- ERM

- Game Maker Language
- LotusScript[7]
- MQL4 script
- UnrealScript
- VBA

Универсальные сценарные языки.

Этот тип сценарных языков наиболее известен (особенно в применении к веб-программированию). Языки этого типа стали возникать с 1990-х годов.

- Tcl (Tool command language)
- Lua
- Perl
- PHP
- Python
- REBOL
- Ruby

Следует заметить, что многие языки этой категории имеют более широкое применение, чем в качестве просто языков сценариев.

Плагины и скрипты

Для написания пользовательских расширений могут использоваться как скрипты (в терминологии некоторых программ «макросы»), так и плагины (независимые модули, написанные на компилируемых языках; в некоторых программах они могут называться «утилитами», «экспортёрами», «драйверами»).

Скриптовый язык удобен в следующих случаях:

Если нужно обеспечить программируемость без риска дестабилизировать систему. Так как, в отличие от плагинов, скрипты интерпретируются, а не компилируются, неправильно написанный скрипт выведет диагностическое сообщение, а не приведёт к системному краху;

Если важен выразительный код. Во-первых, чем сложнее система, тем больше кода приходится писать «потому, что это нужно». Во-вторых, в скриптовом языке может быть совсем другая концепция программирования, чем в основной программе — например, игра может быть монолитным

однопоточным приложением, в то время как управляющие персонажами скрипты выполняются параллельно или как сопрограммы. В-третьих, скриптовый язык имеет собственный проблемно-ориентированный набор команд, и одна строка скрипта может делать то же, что несколько десятков строк на традиционном языке. Как следствие, на скриптовом языке может писать программист очень низкой квалификации — например, геймдизайнер своими руками, не полагаясь на программистов, может корректировать правила игры;

Если требуется кроссплатформенность. Хорошим примером является JavaScript — его исполняют браузеры под самыми разными ОС.

У плагинов же есть три важных преимущества.

Готовые программы, оттранслированные в машинный код, выполняются значительно быстрее скриптов, которые интерпретируются из исходного кода динамически при каждом исполнении. Поэтому скриптовые языки не применяются для написания программ, требующих оптимальности и быстроты исполнения. Но из-за простоты они часто применяются для написания небольших, одноразовых («проблемных») программ.

Полный доступ к любому аппаратному обеспечению или ресурсу ОС (в скриптовом языке для этого должен существовать специальный API, написанный на компилируемом языке). Плагины, работающие с аппаратным обеспечением, традиционно называют драйверами.

Если предполагается интенсивный обмен данными между основной программой и пользовательским расширением, для плагина его обеспечить проще.

А теперь разберемся что такое модуль? Модули - это законченные программные блоки, состоящие из моделей, представлений, контроллеров и других вспомогательных компонентов. При установке модулей в приложение, конечный пользователь получает доступ к их контроллерам. По этой причине модули часто рассматриваются как миниатюрные

приложения. В отличии от приложений, модули нельзя развертывать отдельно. Модули должны находиться внутри приложений.

Создание модулей

Модуль помещается в директорию, которая называется базовым путем модуля. Так же как и в директории приложения, в этой директории существуют поддиректории controllers, models, views и другие, в которых размещаются контроллеры, модели, представления и другие элементы. В следующем примере показано примерное содержимое модуля:

forum/

Module.php файл класса модуля

controllers/ содержит файлы классов контроллеров

DefaultController.php файл класса контроллера по умолчанию

models/ содержит файлы классов моделей

views/ содержит файлы представлений контроллеров и шаблонов

layouts/ содержит файлы представлений шаблонов

default/ содержит файлы представления контроллера DefaultController

index.php файл основного представления

Классы модулей

Каждый модуль объявляется с помощью уникального класса, который наследуется от yii\base\Module. Этот класс должен быть помещен в корне базового пути модуля и поддерживать автозагрузку. Во время доступа к модулю будет создан один экземпляр соответствующего класса модуля. Как и экземпляры приложения, экземпляры модулей нужны, чтобы код модулей мог получить общий доступ к данным и компонентам.

Приведем пример того, как может выглядеть класс модуля:

```
namespace app\modules\forum;

class Module extends \yii\base\Module
{
   public function init()
   {
      parent::init();

      $this->params['foo'] = 'bar';
      // ... остальной инициализирующий код ...
   }
}
```

Если метод init() стал слишком громоздким из-за кода, который задает свойства модуля, эти свойства можно сохранить в виде конфигурации, а затем загрузить в методе init() следующим образом:

```
public function init()
{
   parent::init();
   // инициализация модуля с помощью конфигурации, загруженной из conf
ig.php
   \Yii::configure($this, require(__DIR__ . '/config.php'));
}
```

При этом в конфигурационном файле config.php может быть код следующего вида, аналогичныйконфигурации приложения:

```
<?php
return [
   'components' => [
      // список конфигураций компонентов
   ],
   'params' => [
      // список параметров
   ],
];
```

Контроллеры в модулях

При создании контроллеров модуля принято помещать классы контроллеров в подпространство controllers пространства имён класса модуля. Это также подразумевает, что файлы классов контроллеров должны располагаться в директории controllers базового пути модуля. Например,

чтобы описать контроллер post в модуле forum из предыдущего примера, класс контроллера объявляется следующим образом:

```
namespace app\modules\forum\controllers;

use yii\web\Controller;

class PostController extends Controller
{
    // ...
}
```

Изменить пространство имен классов контроллеров можно задав свойство yii\base\Module::controllerNamespace. Если какие-либо контроллеры выпадают из этого пространства имен, доступ к ним можно осуществить, настроив свойство yii\base\Module::controllerMap, аналогично тому, как это делается в приложении.

Представления в модулях

Представления модуля также следует поместить в поддиректорию views базового пути модуля. Виды, которые рендерит контроллер модуля, должны располагаться в директории views/ControllerID, где ControllerID соответствует идентификатору контроллера. Например, если контроллер реализуется классом PostController, представления следует разместить в поддиректории views/post базового пути модуля.

В модуле можно задать шаблон, который будет использоваться для рендеринга всех представлений контроллерами модуля. По умолчанию шаблон помещается в директорию views/layouts, а свойство yii\base\Module::layout должно указывать на имя этого шаблона. Если не задать свойство layout, модуль будет использовать шаблон, заданный в приложении.

Консольные команды в модулях

Модуль также может объявлять команды, которые будут доступны через консоль.

Для того, чтобы команда стала доступна, надо изменить свойство yii\base\Module::controllerNamespace для консольного режима так, чтобы оно содержало пространство имён ваших команд.

Этого можно добиться проверяя класс экземпляра приложения Yii в методе init модуля:

```
        public function init()
{
   parent::init();
   if (Yii::$app instanceof \yii\console\Application) {
      $this->controllerNamespace = 'app\modules\forum\commands';
   }
}
```

Ваши команды будут доступны из командной строки как:

yii <module_id>/<command>/<sub_command>

Использование модулей

Чтобы задействовать модуль в приложении, достаточно включить его в свойство modules в конфигурации приложения. Следующий код в конфигурации приложения задействует модуль forum:

```
        [
   'modules' => [
      'forum' => [
         'class' => 'app\modules\forum\Module',
         // ... другие настройки модуля ...
      ],
   ],
]
```

Свойству modules присваивается массив, содержащий конфигурацию модуля. Каждый ключ массива представляет собой идентификатор модуля, который однозначно определяет модуль среди других модулей приложения, а соответствующий массив - это конфигурация для создания модуля.

Маршруты

Как маршруты приложения используются для обращения к контроллерам приложения, маршруты модуля используются, чтобы

обращаться к контроллерам этого модуля. Маршрут контроллера в модуле должен начинаться с идентификатора модуля, за которым следуют идентификатор контроллера и идентификатор действия. Например, если в приложении задействован модуль forum, то маршрут forum/post/index соответствуетдействию index контроллера post этого модуля.

Если маршрут состоит только из идентификатора модуля, то контроллер и действие определяются исходя из свойства yii\base\Module::defaultRoute, которое по умолчанию равно default. Таким образом, маршрут forum соответствует контроллеру default модуля forum.

Получение доступа к модулям

Зачастую внутри модуля может потребоваться доступ к экземпляру класса модуля, через который получаются идентификатор модуля, его параметры, компоненты, и т. п. Это можно сделать с помощью следующей конструкции:

$module = MyModuleClass::getInstance();

где MyModuleClass соответствует имени класса модуля, доступ к которому нужно получить. Метод getInstance() возвращает запрошенный в данный момент экземпляр класса модуля. Если модуль не запрошен, метод вернет null. Учтите, что обычно экземпляры класса модуля вручную не создаются, так как созданный вручную экземпляр будет отличаться от экземпляра, созданного Yii в качестве ответа на запрос.

При разработке модуля нельзя исходить из предположения, что модулю будет назначен конкретный идентификатор. Это связано с тем, что идентификатор, назначаемый модулю при использовании в приложении или в другом модуле, может быть выбран совершенно произвольно. Чтобы получить идентификатор модуля, нужно вначале выбрать экземпляр модуля, как это описано выше, а затем получить доступ к идентификатору через свойство $module->id.

Доступ к экземпляру модуля можно получить следующими способами:

```
// получение дочернего модуля с идентификатором "forum"
$module = \Yii::$app->getModule('forum');

// получение модуля, к которому принадлежит запрошенный в настоящее время контроллер
$module = \Yii::$app->controller->module;
```

Первый подход годится только если известен идентификатор модуля, а второй подход наиболее полезен, если известно, какой контроллер запрошен.

Имея экземпляр модуля можно получить доступ к параметрам и компонентам, зарегистрированным в модуле. Например,

```
$maxPostCount = $module->params['maxPostCount'];
```

Предзагрузка модулей

Может потребоваться запускать некоторые модули при каждом запросе. Модуль debug - один из таких модулей. Для этого список идентификаторов таких модулей необходимо указать в свойстве bootstrap приложения.

Например, следующая конфигурация приложения обеспечивает загрузку модуля debug при каждом запросе:

```
[
    'bootstrap' => [
        'debug',
    ],

    'modules' => [
        'debug' => 'yii\debug\Module',
    ],
]
```

Вложенные модули

Модули могут вкладываться друг в друга без ограничений по глубине. Иными словами, в модуле содержится модуль, в который входит еще один модуль, и т. д. Первый модуль называется родительским, остальные - дочерними. Дочерние модули объявляются в свойстве modules родительских модулей. Например,

```
namespace app\modules\forum;

class Module extends \yii\base\Module
{
   public function init()
   {
      parent::init();

      $this->modules = [
         'admin' => [
            // здесь имеет смысл использовать более лаконичное пространство имен
            'class' => 'app\modules\forum\modules\admin\Module',
         ],
      ];
   }
}
```

Маршрут к контроллеру вложенного модуля должен содержать идентификаторы всех его предков. Например, маршрут forum/admin/dashboard/index соответствует
действию index контроллера dashboardмодуля admin, который в свою очередь является дочерним модулем модуля forum.

Метод getModule() возвращает только те дочерние модули, которые принадлежат родительскому модулю непосредственно. В свойстве yii\base\Application::loadedModules содержится список загруженных модулей, в том числе прямых и косвенных потомков, с индексированием по имени класса.

Модули лучше всего подходят для крупных приложений, функционал которых можно разделить на несколько групп, в каждой из которых функции тесно связаны между собой. Каждая группа функций может разрабатываться в виде модуля, над которым работает один разработчик или одна команда.

Модули - это хороший способ повторно использовать код на уровне групп функций. В виде модулей можно реализовать такую функциональность, как управление пользователями или управление комментариями, а затем использовать эти модули в будущих разработках.

Для установки скрипта и модуля в систему были рассмотрены в пример модули отправки СМС с сайта https://github.com/istt/yii2-sms-module/blob/master/SmsModule. Php:

Источник: https://github.com/istt/yii2-sms-module/blob/master/SmsModule.php

В итоге был реализован скрипт отправки коротких сообщении СМС который можно увидеть в “Приложении.Листинг 1.”.

2.2 Вопросы внедрения и настройка скрипта отправки СМС на примере корпоративного сайта ООО “INFO SYSTEM NUKUS” .

В этой части моей дипломной работы хотелось бы рассказать непосредственно о самом сайте и о вопросах внедрения системы СМС.

Для реализации системы отправки СМС был разработан корпоративный сайт ООО “INFOSYSTEM NUKUS”, рассмотрим администраторскую страницу сайта, где можем подробно ознакомится с самим сайтом и в каких целях будут рассылаться сообщения.

В главной странице панели меню можно увидеть такие вкладки как:

- Начало работы;
- Компания;

– Работа;
– Заявки;
– Данные о клиенте;
– Документы;
– Расходы;
– Сотрудники;
– Склад;
– Справочники;
– Отчеты;

(По соображениям конфиденциальности некоторые элементы сайта были скрыты).

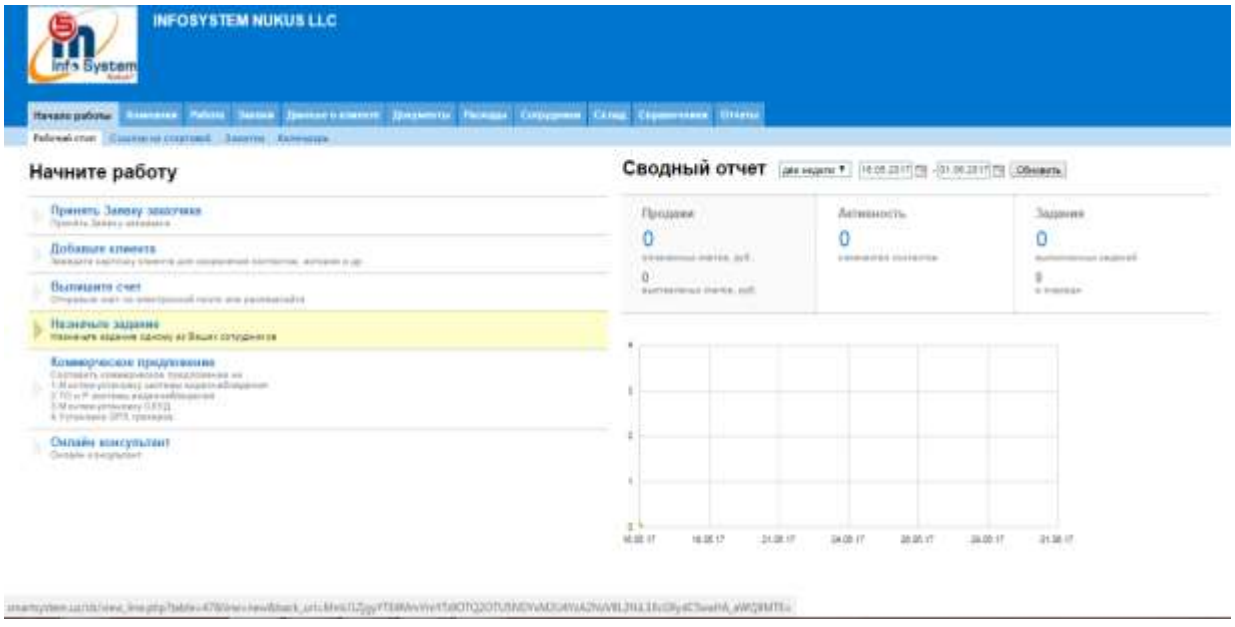


Рис 2.2.1 Начало работы

В этой вкладке существуют возможности принятии заявок, добавление клиента, выставление счета, назначение задания, разные коммерческие предложения и онлайн консультация.

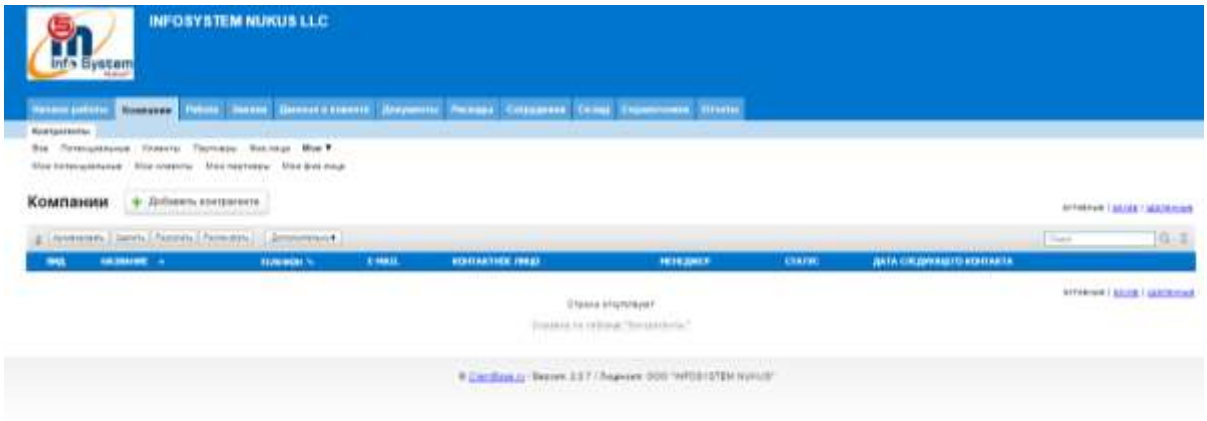


Рис 2.2.2 Компания

Во вкладке “компании” можно посмотреть все компании добавить новую компанию или удалить по разным причинам.

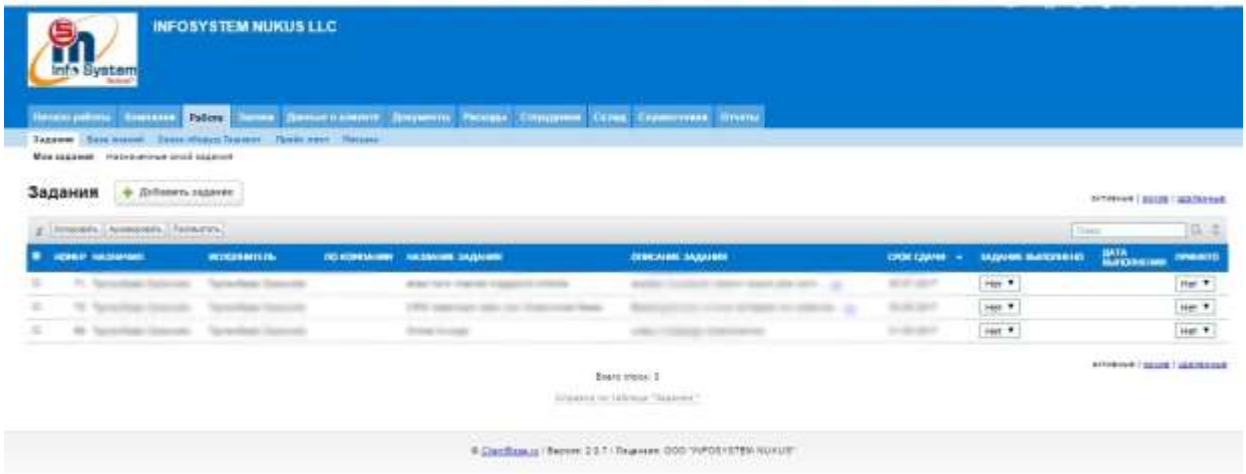


Рис 2.2.3 Работа

На этой вкладке можно добавить новое задание, или посмотреть список задании и заказов, назначить новое задание сотрудникам . Для этого пункта посредством внедрения CMC сервиса будет выполнено работа по автоматизации и контролю работ в главе три настоящей работы.

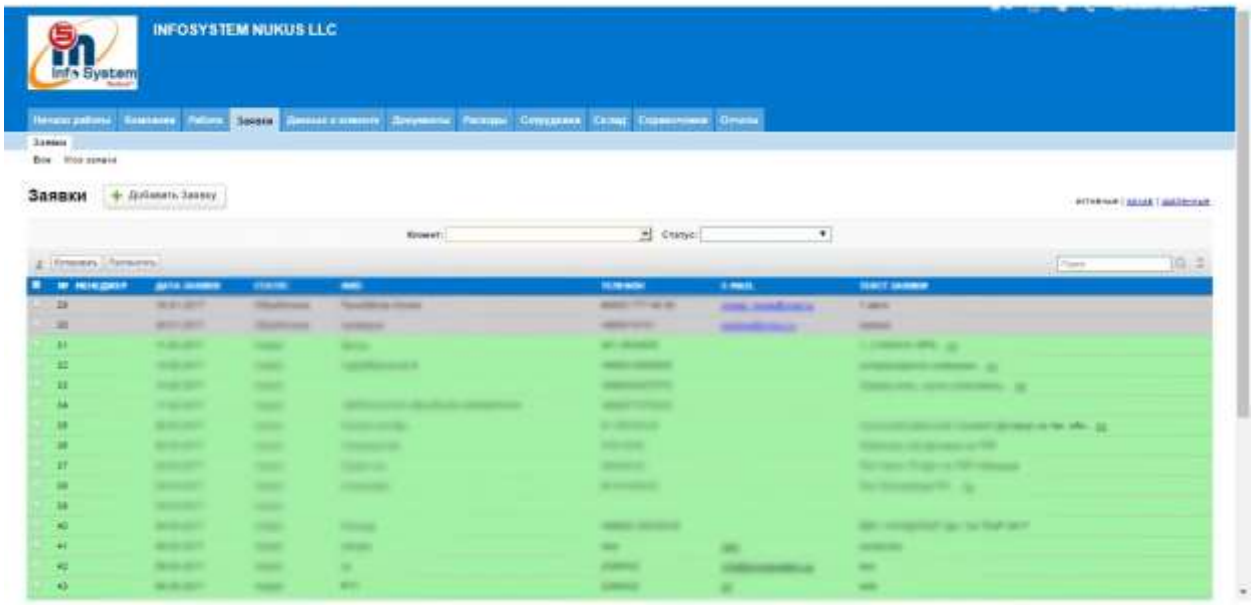


Рис 2.2.4 Заявки

Тут хранятся список заявок клиентов и их данные,есть возможность добавления новой заявки. Это самый ключевой момент работы нашей системы , так как по этим данным можно рассылать СМС клиентам или некоторые данные сотрудникам, например: адрес заказчика или номер телефона. К этой теме мы с вами еще вернемся в главе три.

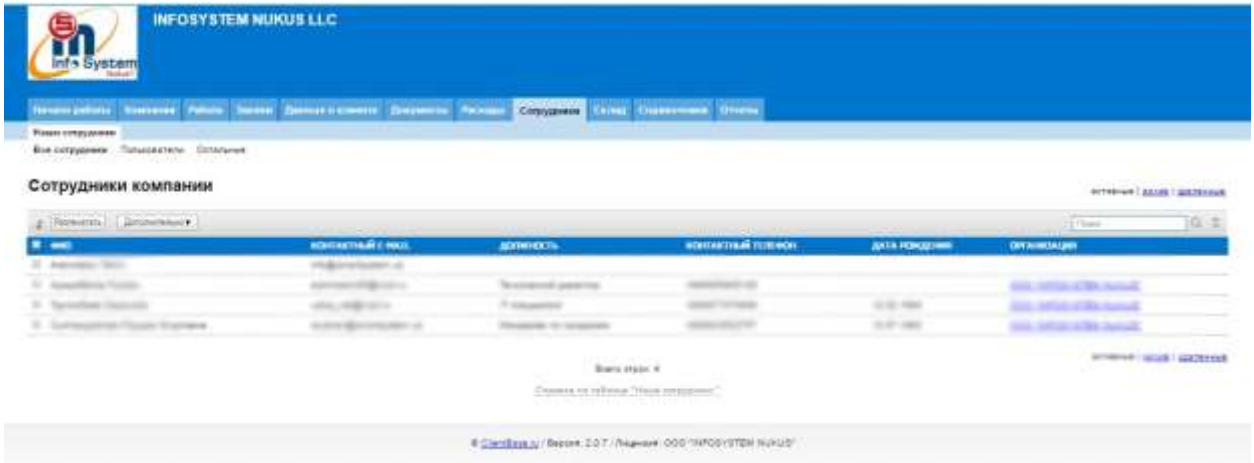


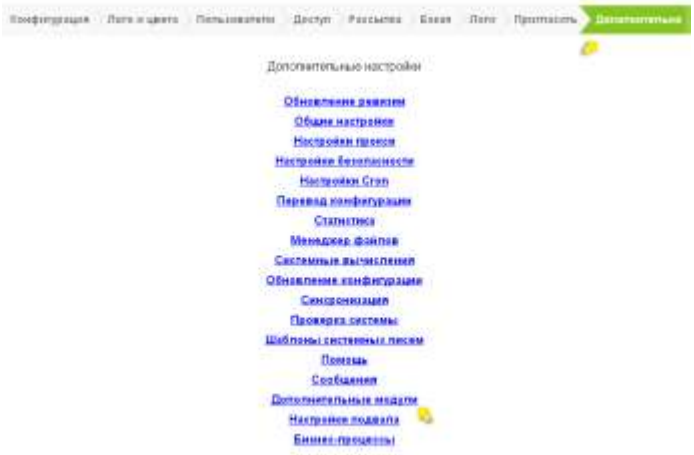
Рис 2.2.5 Сотрудники компании

И самая основная вкладка “сотрудники компании”, как видите тут хранятся данные о сотрудниках. В основном система будет отправлять СМС сотрудникам и изначально в системе будет хранится их контактные номера.

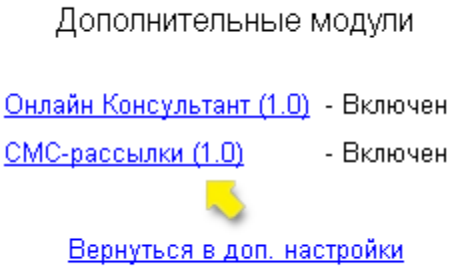
ГЛАВА 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1 Реализация и настройка модуля автоматической рассылки СМС заказчикам компании.

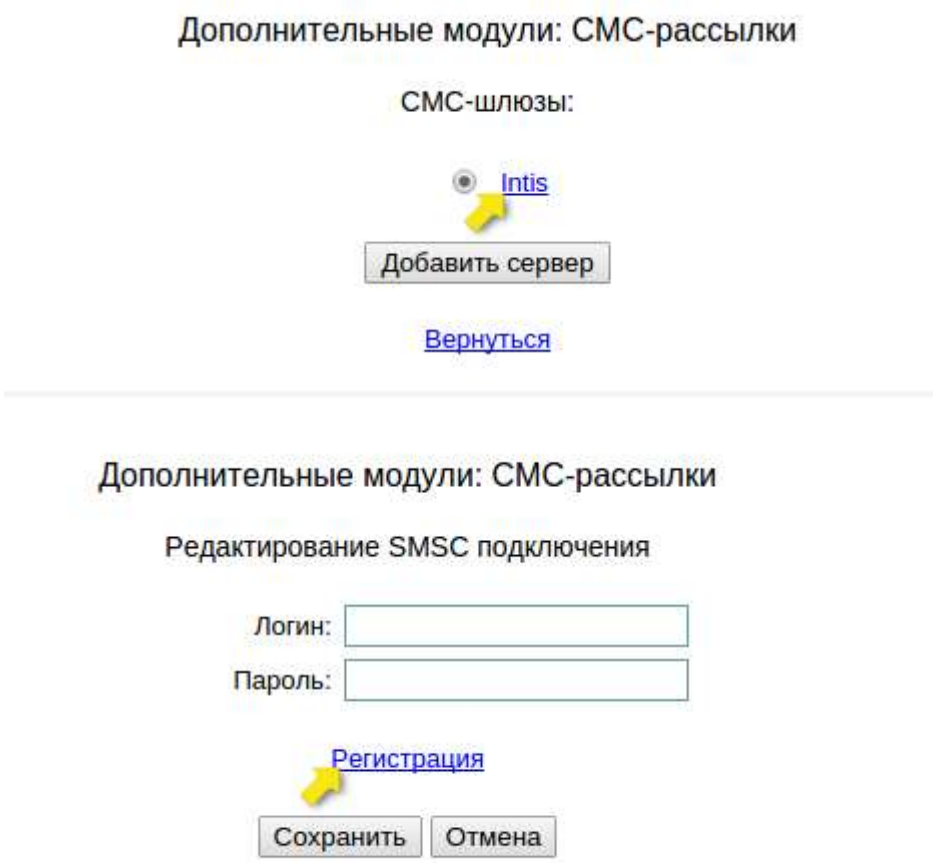
Для того чтобы произвести рассылку по СМС, необходимо для начала активировать СМС-модуль. Для этого перейдем в "Настройки" — "Дополнительно" и выберем "Дополнительные модули".



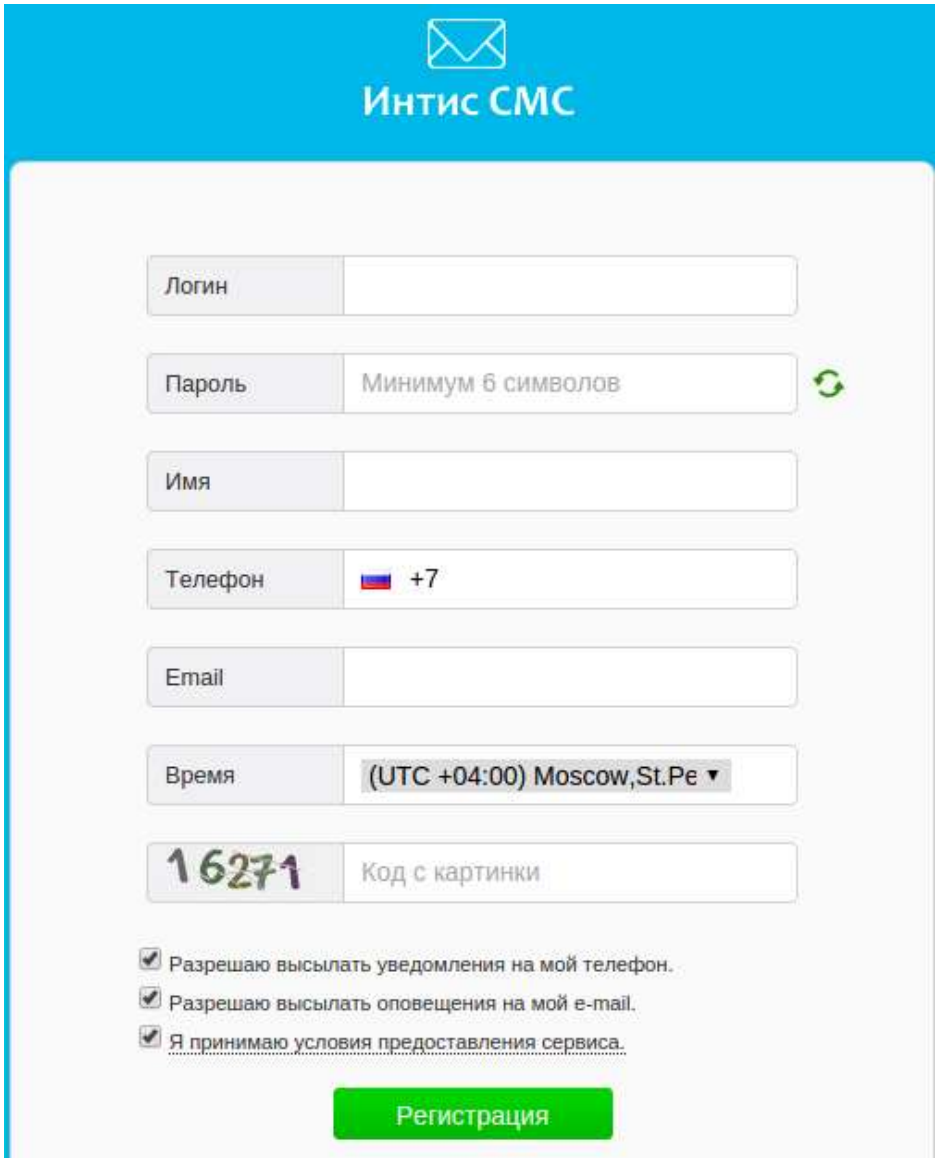
А затем в открывшемся окне выбирается модуль СМС.



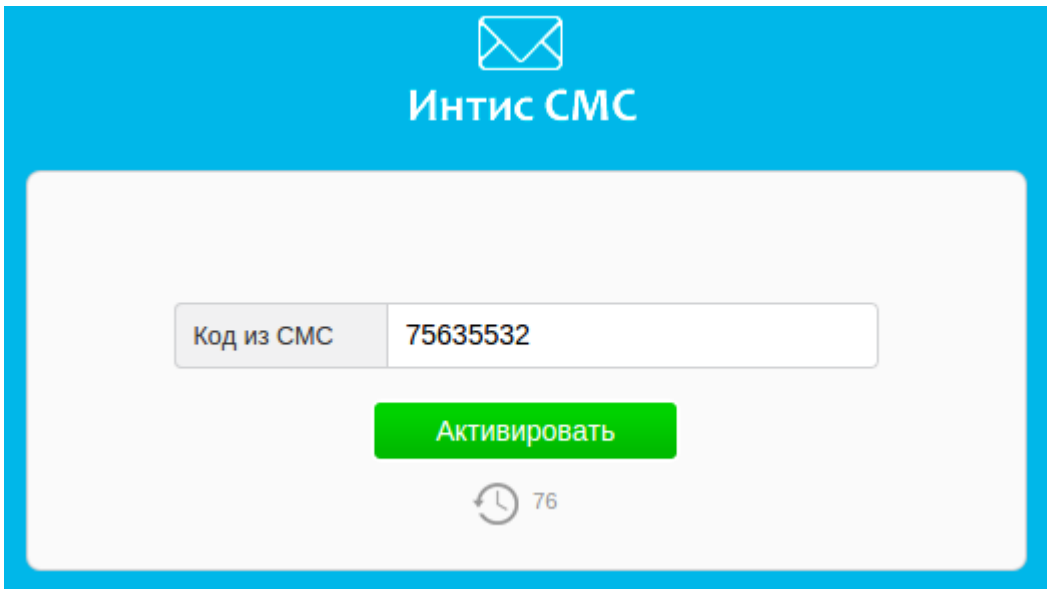
Здесь, нажав на уже имеющийся внутренний шлюз «Intis», необходимо будет первоначально зарегистрироваться на сайте поставщика смс-услуг. Для этого нужно перейти по ссылки «Регистрация»:



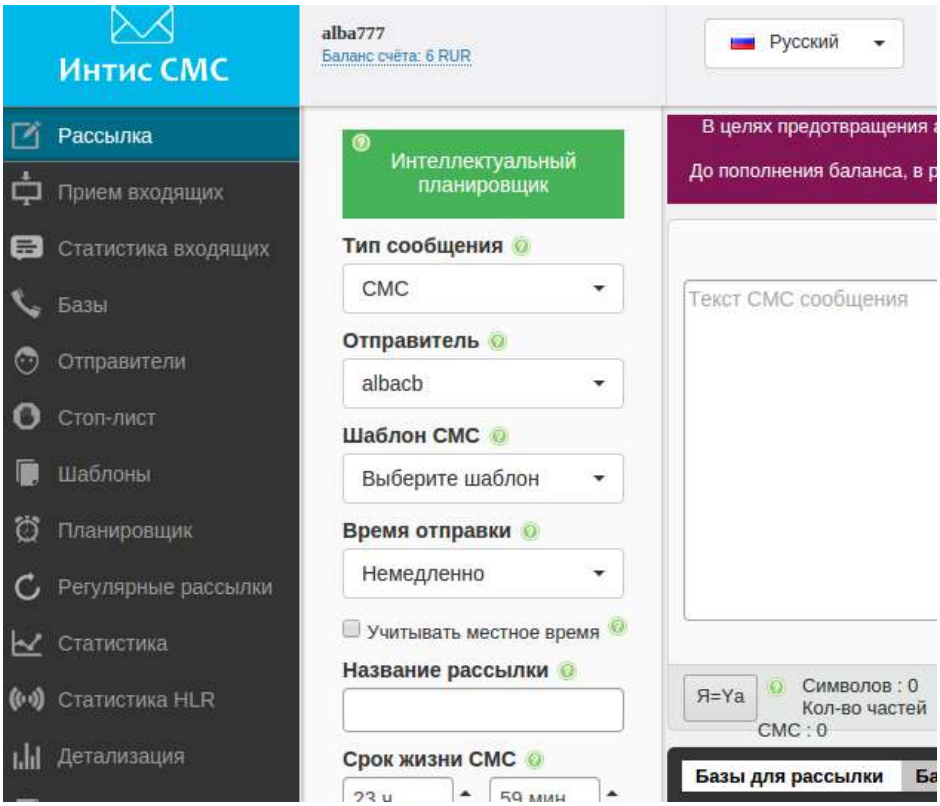
В открывшемся окне, нужно заполнить форму регистрации:



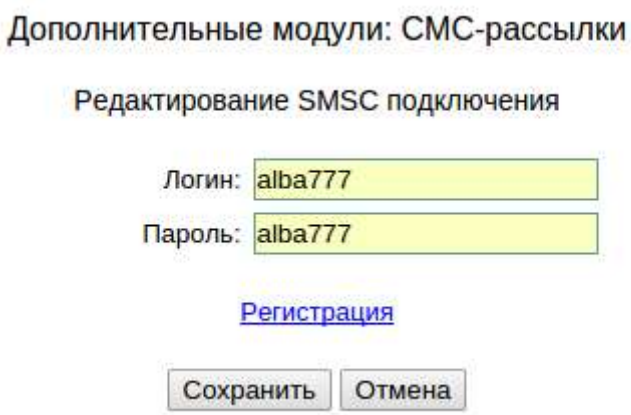
Заполнив регистрационную форму, на указанный контактный номер телефона должно прийти СМС с кодом, который необходимо будет ввести для успешного завершения и подтверждения регистрации:



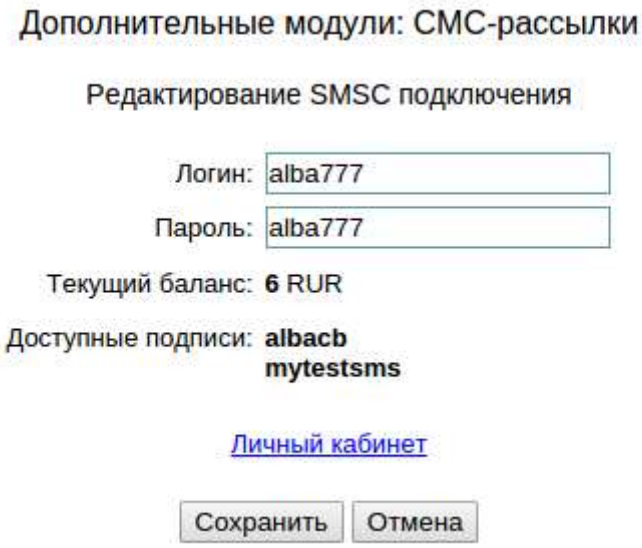
После завершения регистрации система автоматический переходит в личный кабинет. Здесь представлены все необходимые раздела для управления услугой СМС-рассылки.



Получив доступ от личного кабинета нужно вернуться к настройке СМС-шлюза «Intis», ввести регистрационные данные (логин и пароль) в соответствующие поля и сохранить.



Если данные были введены корректно,то сохранение пройдет успешно и шлюз «Intis» активируется. В настройках шлюза будут указаны следующие параметры:



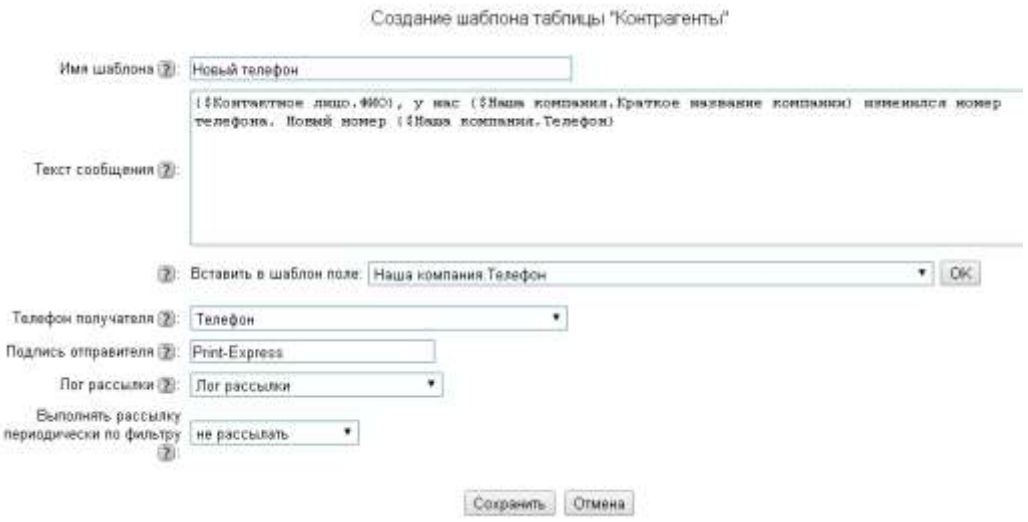
- Логин/пароль;

- Текущий баланс (пополняется через личный кабинет «Intis»);

- Доступные подписи.

Теперь можно перейти к созданию шаблона рассылки для СМС. Для этого

нужно перейти в настройки таблицы, выбрать вкладку "Шаблоны СМС-рассылки" и нажать там "Добавить шаблон".



Принцип его создания такой же, как и у шаблонов рассылки по e-mail, однако здесь стоит учитывать, что при СМС-рассылке не будет отправляться html-код. А вместо поля, откуда будет браться e-mail получателя необходимо указать поле, откуда будет браться телефон получателя. Также есть дополнительная опция — это подпись отправителя. Подпись отправителя - это номер телефона или текст небольшой длины, отображаемый в телефоне получателя в поле "От кого". При рассылке через СМС-шлюз Intis в данном поле можно выбрать одну из зарегистрированных вами подписей (регистрация своей подписи осуществляется в личном кабинете на сайте компании «Intis»). При рассылке через сторонний шлюз — подпись в этом поле вводится вручную.

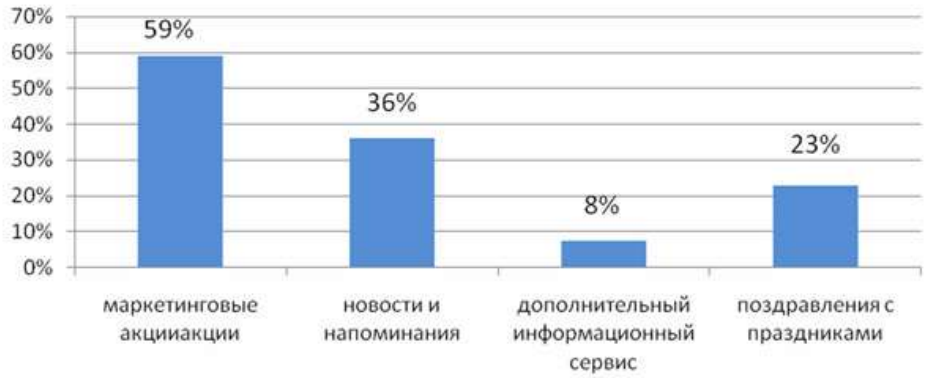
После создалуия шаблона СМС, можно перейти в таблицу, и нажав на кнопку "СМС" разослать его.

3.2 Применение СМС системы для маркетинга и рекламы компании.

SMS-рассылки - многофункциональный маркетинговый инструмент, и комплексный подход к его использованию значительно повышает эффективность. Проведенный смысловой анализ текстов сообщений, отправляемых компаниями, позволяет классифицировать SMS-рассылки по содержанию:

- маркетинговые акции
- новости и напоминания
- дополнительный информационный сервис (прогнозы погоды, курсы валют и т.д.)
- поздравления с праздниками (в том числе Дни рождения)

Соотношение текстов по смысловому содержанию

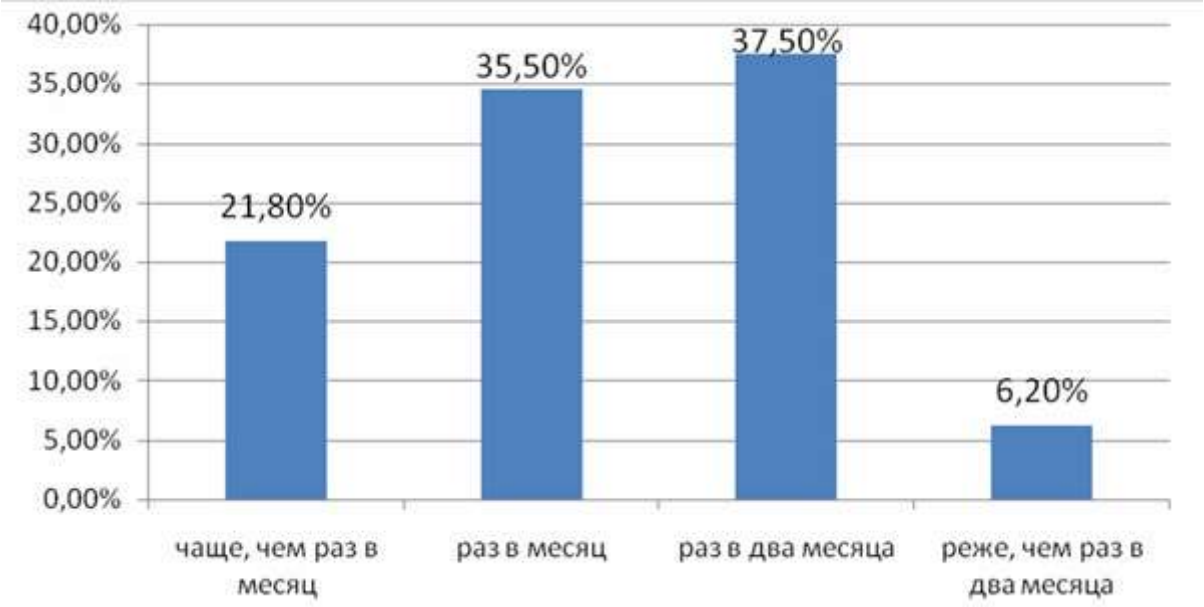


Источник:http://www.km.ru/tendentsii-ispolzovaniya-sms-rassylok-i-ikh-effektivnosti

Частота SMS-рассылок и объемы баз данных

Следующий немаловажный фактор в организации SMS-рассылок - это их частота. Коммуникация по SMS должна быть регулярной и, в то же время, неназойливой. Рекомендуется проводить рассылки примерно один раз в месяц, чтобы поддерживать баланс, однако при этом стоит учитывать и повод: насколько он может быть интересен абонентам.

Соотношение частоты рассылок



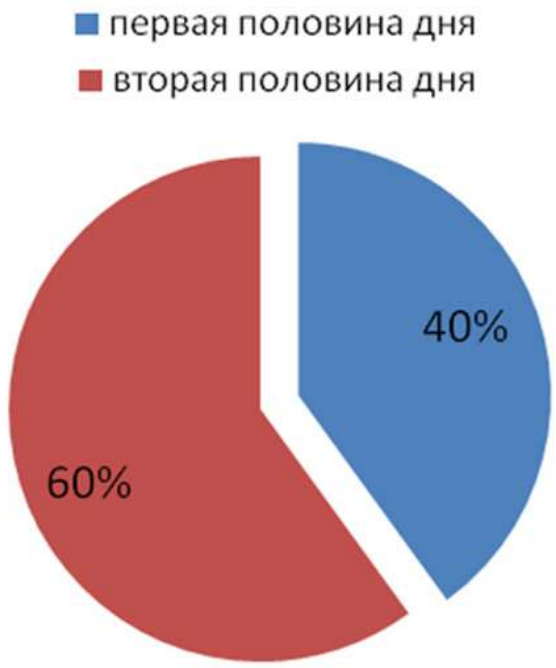
Источник:http://www.km.ru/tendentsii-ispolzovaniya-sms-rassylok-i-ikh-effektivnosti

Количество подписчиков у разных клиентов разное - от десятков и сотен до нескольких миллионов. Скажем, группа заказчиков с базами данных до 1000 номеров - самая многочисленная (~37%).

Время для запуска рассылки

SMS-рассылки - довольно личный канал коммуникации, поэтому необходимо учитывать многие факторы, в том числе время суток для отправки сообщений. Сотовые компании на 60% предпочитают запускать рассылки с 16:00 до 20:00, и 40%- с 11:00 до 16:00. Ни раньше, ни позже этого времени запускать рассылки не рекомендует никто.

Соотношение времени суток для запуска рассылки

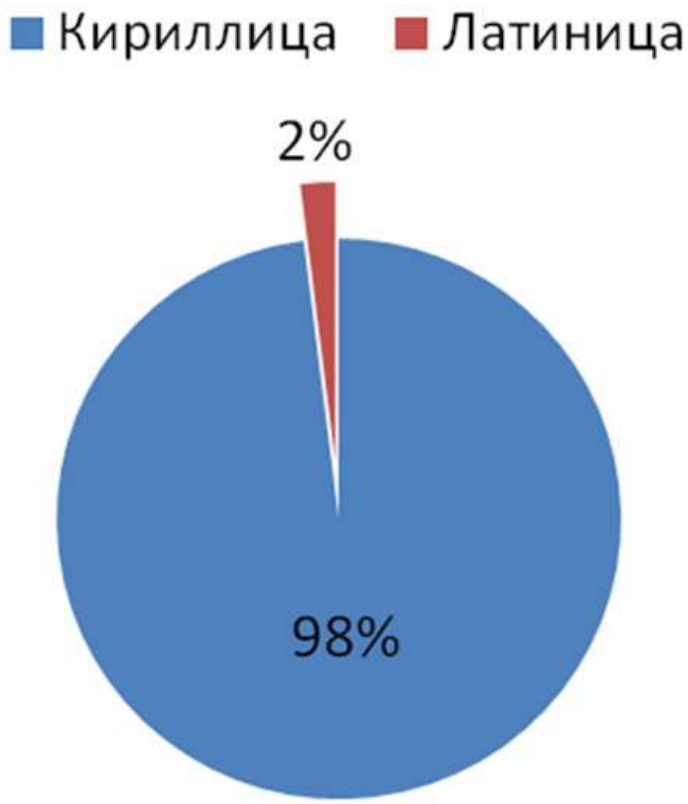


Источник:http://www.km.ru/tendentsii-ispolzovaniya-sms-rassylok-i-ikh-effektivnosti

Язык сообщения

Как было изложено в первой главе настоящей работы специфика SMS-рассылок заключается в том, что сообщения ограничены по объёму: 70 символов на русском языке, при превышении сообщения склеиваются в два и более. На латинице объём сообщения значительно увеличивается - до 160 символов. Однако прочтение текста на латинице может быть затруднительным, что, несомненно, влияет на отношение абонентов. Рекомендуется отправлять сообщения на кириллице, так как показывает практика в нашей стране 98% абонентов пишут и читают на кириллице.

Язык сообщения



Источник:http://www.km.ru/tendentsii-ispolzovaniya-sms-rassylok-i-ikh-effektivnosti

ЗАКЛЮЧЕНИЕ

Цель квалификационнй работы заключалась в изучении проблем и особенностей создания системы отправки коротких сообщении СМС на основе фреймворка YII2.

В ходе выполнения работы мною были решены следующие задачи:

- анализ известных СМС сервисов и СМС платформ в мировой практике и в Узбекистане;

- изучение функциональных возможностей фреймворка YII2;

- установка скриптов и настройка модулей отправки СМС на основе YII2 и его применение;

- изучение вопросов внедрения и настроек скрипта отправки СМС на примере корпоративного сайта ООО “INFOSYSTEM NUKUS” ;

-частичная реализация системы отправки коротких сообщении СМС на примере корпоративного сайта ООО “INFOSYSTEM NUKUS”;

На стадии разработки системы отправки коротких сообщении СМС на основе фреймворка YII2 существенную помощь оказали преподавательский состав кафедры информационных технологии Нукусского филиала ТУИТ имени Мухаммада Ал-Хоразмий. Программная реализация проектной работы выполнена на базе компьютерной и серверной техники научной лаборатории под руководством к.т.н Арзымбетова Т.З.

Система отправки коротких сообщении СМС на основе фреймворка YII2 была продемонстрирована специалистам с других кафедр и получила положительные отзывы.

Система имеет большой потенциал дальнейшего развития и сложных инфраструктур применения систем СМС и стандартов GSM. В настоящее время дорабатываются существующие СМС модули системы для дальнейшего развития и расширения системы.

Таким образом в ходе выполнения квалификационной работы все поставленные задачи были решены, цель квалификационной работы достигнута.

ЛИТЕРАТУРА