

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

02.03.03 Математическое обеспечение и администрирование информационных систем

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

**БАКАЛАВРСКАЯ РАБОТА**

на тему Организация массовых рассылок с использованием CRM-систем на примере Тольяттинского государственного университета

Студент	Н.Ю. Баунина	_____
Руководитель	О.В. Аникина	_____
Консультант по аннотации	Н.В. Яценко	_____

**Допустить к защите**  
Заведующий кафедрой, к.тех.н, доцент, А.В. Очеповский \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 2017 г.

Тольятти 2017

## АННОТАЦИЯ

Тема данной квалификационной работы: «Организация массовых рассылок с использованием CRM-систем на примере Тольяттинского государственного университета»

Целью данной работы является разработка и интеграция сервиса для массовых рассылок с использованием CRM-системы. Разработанный в ходе работы сервис может применяться, например, в сфере дистанционного образования с целью оповещения важной информацией групп студентов, абитуриентов или преподавателей путём отправки на электронную почту текстового сообщения, в котором будет указана вся необходимая информация, что значительно сокращает время, необходимое на личное оповещение каждого человека.

Актуальность темы бакалаврской работы обусловлена необходимостью создания сервиса для массовых рассылок с использованием CRM-систем.

Объектом работы является сервис для массовых рассылок с использованием CRM-систем, используемый ТГУ. Предметом - организация массовых рассылок с использованием CRM-систем на примере Тольяттинского государственного университета, с помощью языка программирования PHP.

Для достижения поставленной цели в работе были решены следующие задачи:

- анализ состояния вопроса;
- постановка задач для реализации массовых рассылок;
- разработка и интеграция сервиса для рассылок.

Выпускная квалификационная работа содержит 19 рисунков, 2 таблицы, и список литературы, включающий в себя 20 источников, объем работы 43 стр.

Работа состоит из введения, трех глав, заключения и списка использованной литературы. Разработанный в ходе выпускной квалификационной работы сервис для отправки сообщений, может быть использован для автоматизации процесса массовых рассылок в Тольяттинском государственном университете.

## ANNOTATION

The topic of this qualification work: "Organization of mass mailings using CRM-systems on the example of Togliatti State University"

The purpose of this work is the development and integration of the service for mass mailings using the CRM-system. The service developed in the course of the work can be applied, for example, in the field of distance education in order to inform important groups of students, applicants or teachers by sending an e-mail to a text message, which will indicate all the necessary information, which significantly reduces the time required for personal notification each person.

The relevance of the topic of bachelor's work is conditioned by the need to create a service for mass mailings using CRM-systems.

The object of work is a service for mass mailings using CRM-systems used by TLTSU. The subject of work is the organization of mass mailings using CRM-systems on the example of Togliatti State University, using the programming language PHP. To achieve this goal, the following tasks were accomplished:

- analysis of the status of the issue;
- setting tasks for implementing mass mailings;
- Development and integration of the service for mailings.

The work consists of an introduction, three chapters, conclusion and a list of used literature. Developed during the final qualifying work, the service for sending messages can be used to automate the process of mass mailings in Togliatti State University.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
Глава 1 АНАЛИЗ ОБЪЕКТА АВТОМАТИЗАЦИИ.....	6
1.1 Характеристика предприятия .....	6
1.2 Техничко-экономическая характеристика предметной области.....	8
1.3 Сравнительный обзор сервисов по email-рассылкам.....	12
Глава 2 ПОСТАНОВКА ЗАДАЧ ПО ОРГАНИЗАЦИИ МАССОВЫХ РАССЫЛОК С ИСПОЛЬЗОВАНИЕМ CRM-СИСТЕМ.....	16
2.1 Эксплуатационное назначение программы.....	16
2.2 Выбор среды реализации .....	17
2.3 Создание контекстной панели .....	22
Глава 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МАССОВЫХ РАССЫЛОК.....	26
3.1 Выбор средств реализации.....	26
3.2 Реализация сервиса по организации массовых рассылок.....	31
ЗАКЛЮЧЕНИЕ .....	41
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	42
ПРИЛОЖЕНИЕ А.....	44

## ВВЕДЕНИЕ

В последние годы наблюдается устойчивая тенденция к внедрению в деятельность российских учебных заведений CRM-систем, что бы оповещение студентов и преподавателей происходило в более простой и удобной форме.

Современные CRM-системы позволяют расширить влияние и открывают возможные пути взаимодействия с клиентами. На сегодняшний день они предлагают функции, предоставляемые социальными сетями, мобильными приложениями, порталы самообслуживания и многое другое. CRM-системы дают возможность для интеграции внутренних ресурсов компании с «внешним» маркетинговым инструментом, это помогает лучше понять и попытаться удовлетворить потребности клиентов.

Современное учебное заведение должно сочетать в себе свои традиции и новинки IT-технологий. Для вуза CRM-системы создаются для автоматизации стратегической разработки процесса, чтобы сохранять информацию о потребителях и отдельную историю о каждом, оптимизировать маркетинговую деятельность, ускорить общую работу вуза. Организация рассылок является ключевой функцией в CRM-системе, но автоматизация массовых рассылок может включать массу проблем, связанных с использованием самой системы, одной из таких проблем является то, что все рассылки производятся однотипно и невозможно поменять текст сообщения без технических знаний в области работы с кодом.

Благодаря массовым рассылкам увеличивается процент посещения мероприятий, учебные программы, предлагаемые ВУЗом, быстрее найдут отклик при целенаправленной рассылке и многое другое. На данный момент в ТГУ отсутствует функция, которая организовывала бы такие рассылки.

Таким образом, актуальность темы дипломной работы обусловлена необходимостью разработать сервис для необходимых ВУЗу рассылок, который будет интегрировать с данной CRM-системой.

Объект исследования - сервис для создания массовых рассылок с использованием CRM-системы.

Предмет исследования - организация массовых рассылок с использованием CRM-систем на примере Тольяттинского государственного университета.

Целью данной выпускной квалификационной работы является разработка сервиса для email-рассылок. Для достижения поставленной цели в работе решались следующие задачи:

- анализ объекта автоматизации
- постановка задач для реализации массовых рассылок
- разработка и интеграция сервиса для рассылок на основе проанализированных данных

Работа состоит из введения, трех глав, заключения, списка использованной литературы и приложения.

Во введении описывается актуальность проводимого исследования, формулируется цель, и ставятся задачи, которые необходимо решить для достижения цели.

Первая глава посвящена анализу объекта автоматизации.

Вторая глава посвящена постановке задач для поставленной цели.

В третьей главе описана реализация разработанного сервиса для рассылок.

В заключении сформулированы основные выводы, которые были сделаны в процессе проведения бакалаврской работы и описаны результаты практической реализации ВКР.

# Глава 1 АНАЛИЗ ОБЪЕКТА АВТОМАТИЗАЦИИ

## 1.1 Характеристика предприятия

Чтобы создать единое рабочее пространство для сотрудников, в 2015 году Тольяттинский государственный университет внедрил Корпоративный портал Битрикс24. Одна из главных задач была автоматизировать работу приема студентов дистанционного образования, проекта «Росдистант». В корпоративном портале работают только сотрудники университета, студенты университета - контакты в CRM-системе, а абитуриенты – лиды.

«1С-Битрикс» - лидирующая компания на российском рынке, которая занимается разработкой систем управления для веб-проектов и корпоративных порталов. Продукты компании «1С-Битрикс» пользуются успехом у партнёров во многих городах России и странах бывшего СССР. Более 140 000 клиентов используют продукты «1С-Битрикс» для управления своими проектами. «1С-Битрикс» - предприятие, созданное фирмой «1С» и компанией «Битрикс» для возможности развития веб-направлений, также разрабатывать новые интернет-решения.

Компания занимается продажей профессиональных систем для управления веб-проектами и корпоративными порталами: «1С-Битрикс: Управление сайтом», «Битрикс24» коробочной версии и облачного сервиса Битрикс24, а также даёт возможность развиваться этим продуктам и строит партнерские сети для того, чтобы их распространять, внедрять и поддерживать.

Программные продукты «1С-Битрикс» - профессиональная система для управления веб-проектами «1С-Битрикс: Управление сайтом» и «Битрикс24» - полный комплект инструментов для организации совместной работы компании (облачный сервис и коробочная версия).

Битрикс24.CRM имеет облачную версию и коробочную, Тольяттинский государственный университет выбрал для организации своей работы корпоративный портал коробочной версии.

Коробочная версия Битрикс24 (серверное решение, standalone) – это пакет с программным обеспечением, который клиент разворачивает самостоятельно на своём серверном оборудовании, имеет поддержку и администрирование у заказчика. К работе с ней можно приступить сразу после того, как пользователь установит её на сервер. Для успешной организации работы система имеет все необходимые инструменты. Данная версия предназначена для пользователей, которые предпочитают хранить информацию на собственном сервере. Заказчик приобретает лицензию и право на использование программы, а что требуется компании для полноценной работы, разрабатывает самостоятельно. На рисунке 1.1 представлена стартовая страница Битрикс24.

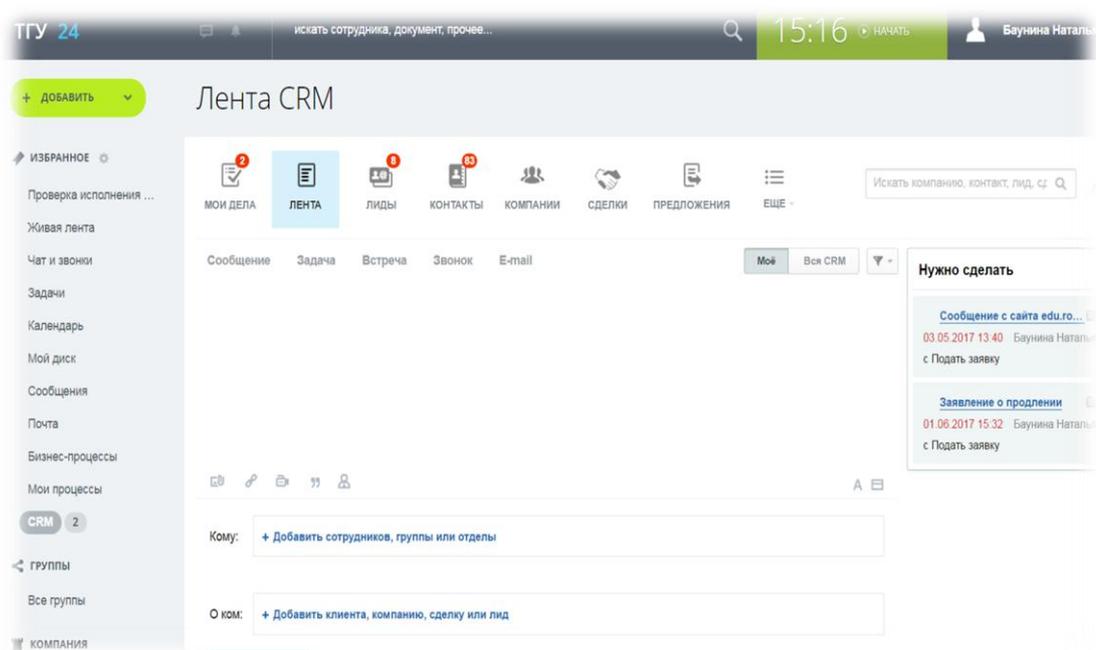


Рисунок 1.1 – Стартовая страница Битрикс24

В данной версии спектр допустимых настроек функционала намного шире, чем в облачной версии, так как пользователю доступна административная панель и сервер находится в его распоряжении. Безопасность работы полностью обеспечивает и контролирует клиент. Битрикс24 коробочной версии имеет больше возможностей для интеграции, пользователь может использовать более функциональное API, так как имеет доступ к базе данных [1].

## **1.2 Техничко-экономическая характеристика предметной области**

Электронная почта является старейшим сервисом в сети Интернет.

Массовой рассылкой называется автоматическая рассылка, которая отсылается по электронной почте. Необходимость массовых рассылок возникает при необходимости разослать много одинаковых электронных писем с уведомлениями о важной информации. Огромный плюс e-mail рассылки в том, что рассылка писем очень быстро осуществляется. В течение нескольких часов можно подготовить текст для уведомления и разослать сообщения. Пользователей реагируют на полученные сообщения уже через несколько минут после начала рассылки. Рассылку можно контролировать и вести отчёты по ней.

ВУЗ, предлагающий свои услуги клиентам, количество которых более 15, нуждается в массовой рассылке по электронной почте. Вовремя поступившее сообщение об акциях или специальных предложениях, мотивируют клиента быстрее оплатить обучение или подать заявление на обучение в данный период времени, положительно влияют на лояльность к ВУЗу, улучшают имидж университета. Массовые рассылки дают возможность проводить маркетинговые опросы, которые помогают выяснить заинтересованность абитуриентов в предлагаемых учебных программах.

Предметной областью бакалаврской работы является организация массовых рассылок с использованием CRM-систем.

Рассмотрим Битрикс24.CRM с точки зрения решения данной задачи.

CRM называют системой управления взаимоотношениями с клиентами (CRM, CRM-система, сокращение от англ. Customer Relationship Management), по-другому CRM-систему можно назвать прикладным программным обеспечением для организации, в том числе ВУЗа, которое предназначено для автоматизации стратегии взаимодействия с клиентами, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

CRM устанавливается как один из компонентов общей корпоративной системы Битрикс24.

В настоящее время существует достаточно большое количество программных товаров, в наименовании или описании которых встречается аббревиатура CRM. Перечень функций основной массы исследований:

1. Создание базы данных по клиентам. От стандартного "Справочника контрагентов", который есть во многих системах, эта база будет отличаться большим числом реквизитов;

2. Возможность ведения контактов с клиентами. В более "продвинутых" программах данная функция хорошо проработана. Фактически, такого рода системы позволяют отказаться от бумажных ежедневников и планировщиков и ликвидировать риск потери контактных данных человеческого фактора;

3. Возможность ведения сделки. Обычно подразумевается бизнес-процесс – «поиск клиентов - коммерческие предложения - этапы уговаривания 1 - этап уговаривания N - этап согласования 1 - этап согласования N - заключение договора - оплата - поставка - установка». Количество этапов зависит от исходного продукта и настраивается под конкретный бизнес. По числу потенциальных клиентов, находящихся на каждом этапе, строится «воронка продаж» - чем дальше этап, тем меньше клиентов;

4. Организация рассылок с помощью CRM-систем;

5. Возможность построения выборок по клиентской базе. На клиентов можно поставить фильтр по всем параметрам клиентской базы. Позже выборка будет использоваться для подготовки отчетов или для организации рассылки.

Рассмотрим подробнее организацию рассылок с помощью CRM-системы:

Существует несколько вариантов работы с почтой с помощью CRM-систем:

Собственный почтовый клиент в CRM.

CRM-система имеет собственный почтовый клиент. Нужно дописать настройки для переадресации из корпоративного почтового ящика и получится полноценный почтовый сервис внутри системы CRM, который имеет функции:

- Получение любых писем, которые отправляются в почтовый ящик;
- Отправка писем;
- Сортировка и поиск по адресу или тексту. писем в почтовом ящике;
- Создание папок для разновидностей писем и хранение их в удобном виде.

Таким образом, получается два варианта работы с email-сообщениями: чтение писем и отправка писем из карточки лида/клиента. Такая система позволяет:

Автоматически прикреплять письмо от известного контакта к карточке лида/клиента.

На основе нового письма-запроса в системе можно создать новый лид.

Если лид уже существует или его удобнее создать вручную, адрес может быть скопирован и добавлен в карточку лида. После этого все новые письма с этого email также будут автоматически прикрепляться к указанному лиду.

Преимущества:

Простота настройки, большой перечень готовых решений для автоматизации работы с email, возможность быстрой настройки и другое.

Недостатки:

Работа с почтой вручную. В CRM приходят письма, никак не связанные с клиентами. Отправка письма не из карточки клиента, а с помощью почтового модуля, может повысить вероятность потери важной информации и возникновения различных накладок из-за человеческого фактора.

Внешний почтовый ящик и отправление писем из CRM.

CRM-система имеет почтовый сервис, но нет рабочего почтового интерфейса. К системе прикреплен корпоративный почтовый ящик, CRM получает с него почту так же, как это делают почтовые клиенты. Но при этом происходит отбор:

Когда письмо приходит с известного системе адресата, оно автоматически будет добавлено в карточку, и ответственное сможет прочитать переписку, а также отправить из карточки ответы на письма.

Когда электронный адрес системе неизвестен, он игнорируется, письмо остаётся в почтовом ящике непрочитанным до того момента, когда пользователь сам просмотрит почту.

Преимущества:

В этом варианте работы с почтой, переписка будет вестись только через карточку клиента.

Недостатки:

Вне системы CRM почту можно просматривать только вручную, если будет выявлен новый запрос, лид для него нужно будет создать вручную. После того, как лид появится в системе с прикрепленным к нему email-адресом, письма от данного контрагента будут попадать напрямую в CRM-систему.

Внешний почтовый ящик и отправка писем с помощью почтового клиента.

Будущий основной почтовый ящик необходимо зарегистрировать в CRM-системе, чтобы вся работа велась через него. Входящие сообщения будут копироваться из почты в карточки клиентов, а отправка исходящих сообщений составит немного больше труда.

Нажатие ответственным лицом кнопки «отправить письмо» перекинет его во внешний почтовый ящик, который был прикреплен к системе.

Исходящее письмо будет отправлено из обычного почтового интерфейса. Скрытая копия этого письма придёт в специальный почтовый ящик, который нужно будет заранее зарегистрировать в CRM-системе специально для такой исходящей почты. Позже система увидит во входящей почте письмо с этим служебным адресом, проанализирует, на какие ещё адреса было отправлено письмо, и прикрепит его к нужной карточке в качестве исходящего в переписке. Архивный почтовый ящик можно создать несколькими способами (рисунок 1.2).

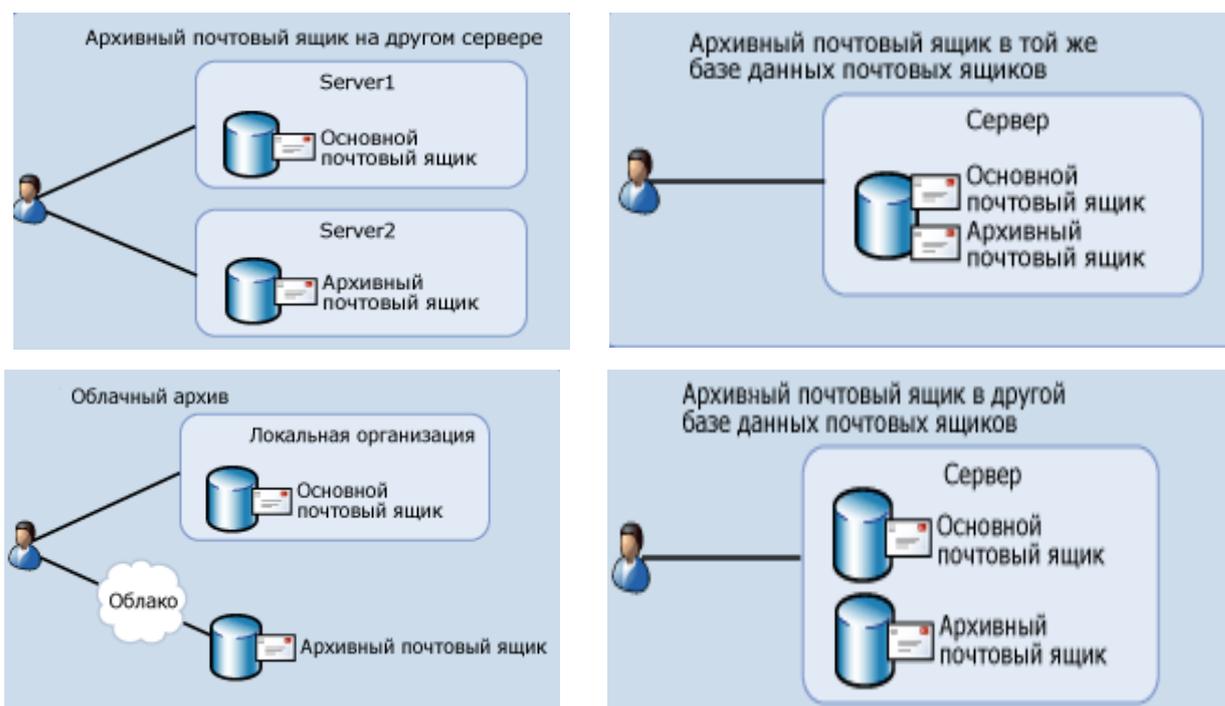


Рисунок 1.2 - Варианты создания архивного почтового ящика

Преимущества:

Переписка и прикрепление сообщений к карточке клиента осуществляется автоматически получение, отсутствуют «лишние модули».

Недостатки:

Обязательное создание «скрытых копий» для почты клиентов, без таких копий письмо для CRM будет утеряно и нельзя будет воспользоваться никакими шаблонами.

Был проведён анализ вариантов работы с почтой в CRM-системе [2].

В разрабатываемом сервисе будет применяться 3 вариант работы с почтой.

### 1.3 Сравнительный обзор сервисов по email-рассылкам

Существует много почтовых сервисов, предназначенных для работы с клиентской базой, самые популярные из них.

Таблица 1.1 – Сравнение email-сервисов

Email-сервис	Интеграция с Битрикс24	Возможности		Недостатки
UniSender	Есть	быстрый перенос контактных данных	перенос данных	Функционал не позволяет делать

Email-сервис	Интеграция с Битрикс24	Возможности	Недостатки
		клиентов подписать клиентов на уже имеющуюся рассылку прямо из интерфейса системы создавать новый список отправки сообщений	рассылки регулярно по расписанию
SendPulse	Нет	контакты, компании и сделки экспортируются в адресную книгу SendPulse позволяет создавать автоматические триггерные рассылки, рекламные рассылки и новостные дайджесты	Отсутствует автоматическая рассылка новых постов из RSS.
Pechkin-mail	Нет	базовые возможности	Отсутствие шаблонов

Рассмотренные сервисы обладают необходимым минимумом функционалом, которого достаточно для: импорта данных, создания писем, отслеживания результатов рассылки, но не подходят для интеграции с корпоративным порталом ТГУ, так как платные.

После проведенного обзора, были собраны главные плюсы и минусы этих сервисов, которые могут помочь для реализации поставленной цели.

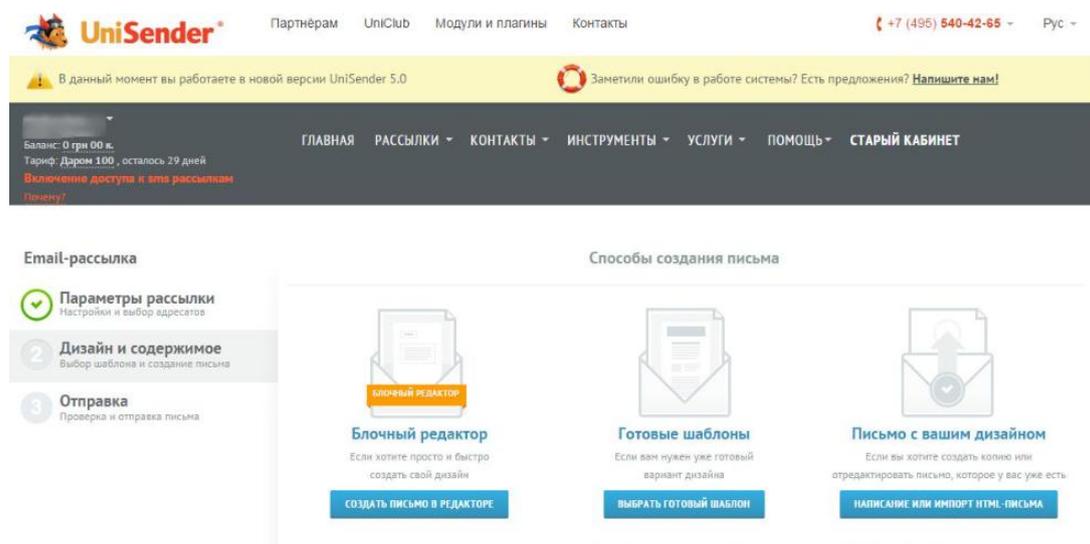


Рисунок 1.3 – Интерфейс UniSender

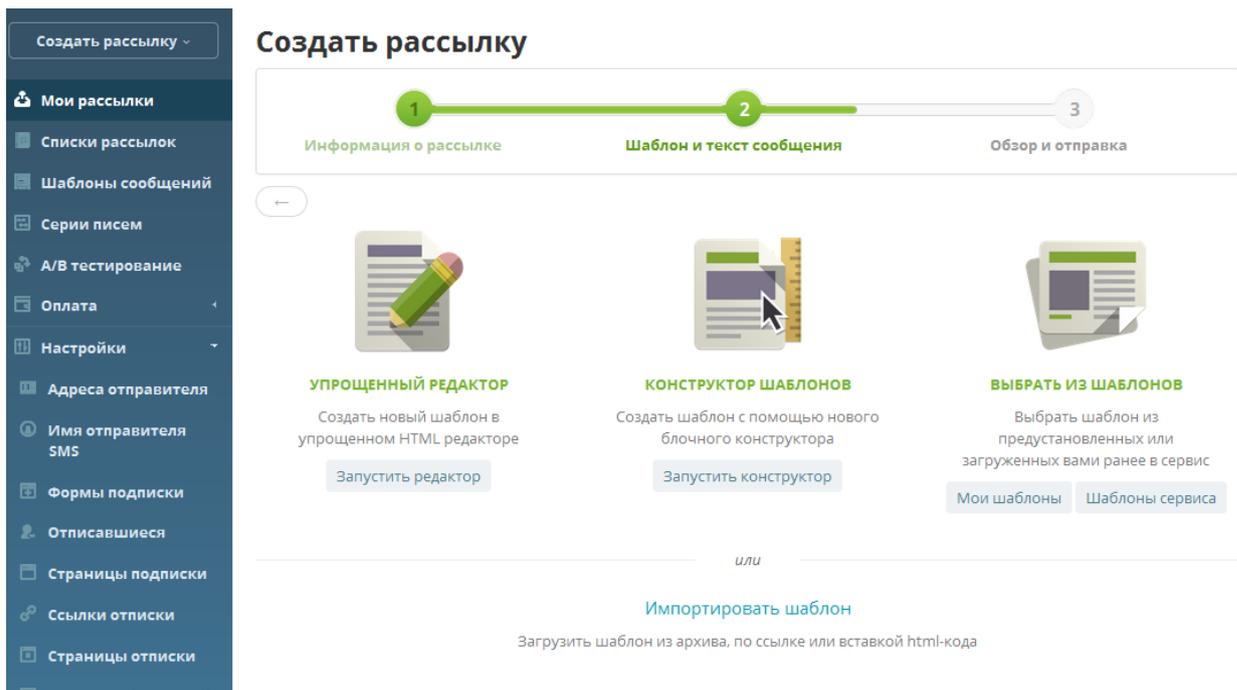


Рисунок 1.4 – Интерфейс SendPulse

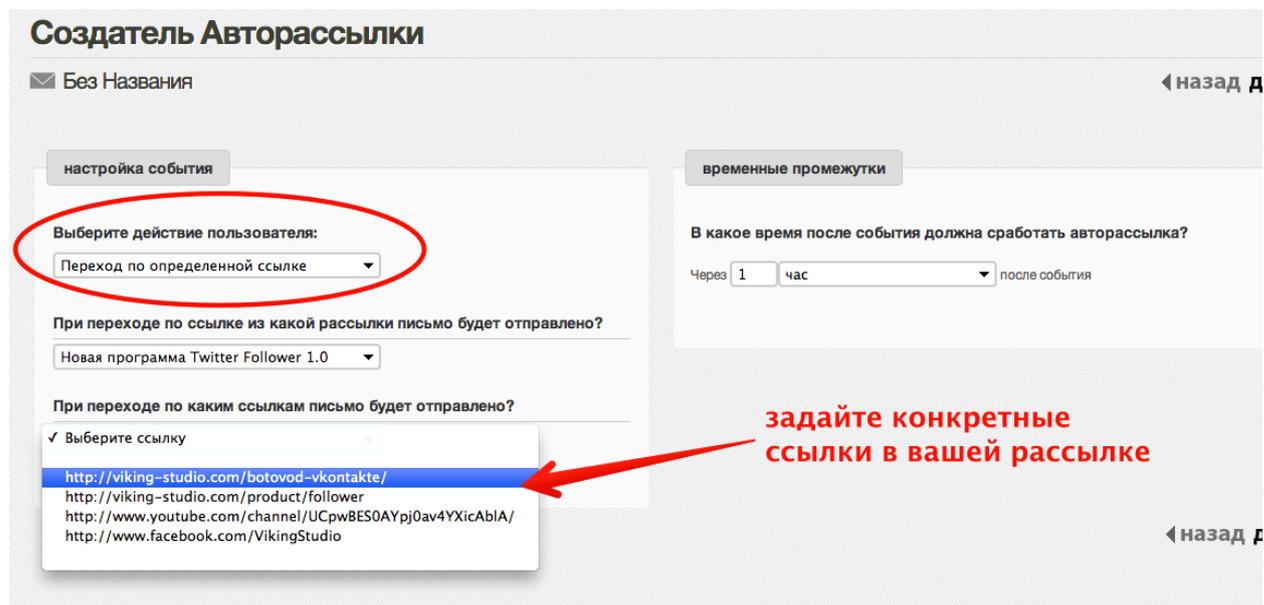


Рисунок 1.5 – Интерфейс SendPulse

Рассмотренные сервисы обладают необходимым минимумом функционалом, которого достаточно для: импорта данных, создания писем, отслеживания результатов рассылки, но не подходят для интеграции с корпоративным порталом ТГУ, так как платные.

После проведённого обзора, были собраны главные плюсы и минусы этих сервисов, которые могут помочь для реализации поставленной цели.

### **Выводы по главе 1**

В главе был проведен анализ объекта предприятия и рассмотрена его характеристика, были проведен анализ email-сервисов, в результате которого были выявлены главные минусы и плюсы сервисов, которые помогли для реализации поставленной цели.

## **Глава 2 ПОСТАНОВКА ЗАДАЧ ПО ОРГАНИЗАЦИИ МАССОВЫХ РАССЫЛОК С ИСПОЛЬЗОВАНИЕМ CRM- СИСТЕМ**

### **2.1 Эксплуатационное назначение программы**

Программа должна эксплуатироваться в учебных заведениях и на предприятиях организаций. Программа автоматически отправляет сообщение по определенному шаблону каждому из пользователей, без участия кого-либо. Конечными пользователями программы могут являться как сотрудники и учащиеся (полное право доступа к информации), так и лица, не имеющие отношения к данному предприятию или учебному заведению (частичное предоставление информации).

Устойчивый функционал программы должен быть обеспечен выполнением организационно-технических мероприятий:

- организация бесперебойного питания технических средств;
- выполнение рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- выполнение требований ГОСТ Р 51583-2000. Защита информации.

Порядок создания автоматизированных систем в защищенном исполнении.

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации – ГОСТ 15150-69.

В Системе должен обеспечиваться надлежащий уровень защиты информации, который будет соответствовать закону о защите персональной информации и программного комплекса в целом от несанкционированного доступа – «Об информации, информатизации и о защите информации» РФ N 149-ФЗ от 27.07.2006.

Для работы с программой, со стороны персонала, требуются не менее двух штатных единиц - системный программист и конечный пользователь программы – оператор, которые будут следить за работоспособностью технических средств, организовывать установку и поддержание операционной системы и самой программы.

## **2.2 Выбор среды реализации**

Для начала нужно определиться с параметрами, которые влияют на выбор языка программирования. Самый главный параметр – направление.

Прикладное или системное направление — это программирование систем на низких уровнях, очень многое будет зависеть от операционной системы. Разрабатываться может операционная система, драйвера, простые и сложные программы. Необходимо обязательно продумать для какой ОС будет разрабатываться программное обеспечение. Основные языки, использующиеся для прикладного программирования – это C, Java, Python, Assembler, C++, Perl.

Второе направление - это web-программирование. Под данным направлением имеется ввиду разработка блогов, конструкторов, сайтов, систем управления контентом и многое другое, другими словами всего, что связано с интернетом. Зависимости от операционной системы почти нет, но существует зависимость, которая связана с какими-либо программами или браузерами. Можно разрабатывать код, который будет выполняться на всех ОС, но разработан будет только на одной. Основные языки для web-программирования – PHP, JS, Java, Python, Ruby.

Так же необходимо учитывать следующие параметры:

- Гибкость языка программирования
- Целевую платформу
- Производительность.
- Время для исполнения проекта.
- Поддержку и сообщество.

1. Гибкость языка программирования.

Для определения гибкости языка нужно проанализировать насколько легко будут добавляться к существующей программе новые возможности, добавление новых функций, посмотреть, как будет работать существующая библиотека, если добавить ей новую функциональность и другое. Прежде чем принять решение, нужно точно знать, как была спроектирована программа и какой функционал ещё не задействован.

## 2. Целевая платформа.

Целевая платформа является очень важным фактором, так как именно на ней будет работать будущая программа. Для примера можно рассмотреть языки программирования C и Java. Для написания программы на C и работы на ОС Windows и Linux, нужны будут два разных исполняемых файла компилятор для платформы. В случае написании программы на языке Java, для выполнения программы на любом компьютере, на котором заранее будет установленная виртуальная Java-машина. будет достаточно сгенерированного байт-кода. Для Web-сайтов можно применить аналогичный аргумент. Сайт должен выглядеть и работать одинаково во всех браузерах, если использовать теги CSS3 и HTML5 без проверки совместимости с браузерами, то работа приведет к разному поведению и отражению сайта в разных браузерах.

## 3. Производительность.

Предел производительности, на которую влияет использованный язык для разработки программы, имеет каждая платформа и программа. Существует немало способов для сравнения скорости работы в одинаковой среде программ, написанных на разных языках. Пользователи используют различные эталонные тесты, хоть и их результаты не являются конкретной оценкой производительности для языка программирования.

Производительность языка нужно учитывать, если целевая среда не предлагает широкой масштабируемости, как например, для разработки приложений для мобильных устройств.

## 4. Время исполнения проекта.

Время исполнения – это время, которое учитывается для разработки рабочей версии программы, т.е. версии, которая будет готова для работы, выполняющей предусмотренные функции в производственных условиях. При расчете такого времени необходимо будет учитывать логику управления, и логику представления. От размера кода зависит время исполнения работы.

#### 5. Поддержка и сообщество.

Как и хорошая программа, язык программирования, должен опираться на твердую поддержку сообщества. Язык с активным форумом должен быть популярнее замечательного языка, помощь по которому трудно найти.

Поддержка сообщества – это форумы, вики-сайты, учебные руководства и, самое важное, дополнительные библиотеки, которые помогают развить навыки языка.

Сервис будет разрабатываться в уже существующем API, на языке программирования PHP.

API - это интерфейс прикладного программирования, содержит в себе готовые классы, функции, процедуры, структуры, которые предоставляет сервис или операционная система, для возможности использования во внешних программных продуктах.

Язык программирования PHP, на сегодняшний день, является наиболее распространенным языком для Web-программирования. С помощью этого языка разработано большинство сайтов и веб-сервисов в интернете. Он применяется более чем на 80% сайтов, среди которых такие сервисы, как facebook.com, vk.com, baidu.com и другие. Простота языка позволяет быстро и легко создавать сайты и порталы различных сложностей.

Преимущества языка программирования PHP:

- Для наиболее распространенных операционных систем (Windows, MacOS, Linux) имеются версии пакетов разработки на PHP
- PHP может работать в связке с веб-серверами, например, Apache, Nginx, IIS

- язык работает с системами баз данных (MySQL, Oracle, PostgreSQL, MSSQL, MongoDB и другие)

- Распространены хостинговые услуги и их недорогая цена. Очень часто, хостинговые компании размещают веб-сайты на PHP на веб-серверах Nginx или Apache, которые работают на одной из операционных систем семейства Linux. И операционные системы, и веб-серверы, на базе Linux бесплатны, что снижает общую стоимость использования хостинга

- Данный язык программирования продолжает развиваться, выходят новые версии с новыми функциями.

Недостатки:

- Не подходит для создания системных компонентов или десктопных приложений;

- Имеет слабые средства для работы с исключениями;

- Глобальные параметры конфигурации влияют на базовый синтаксис языка, что затрудняет настройку сервера и разворачивание приложений;

- объекты передаются по значению, это смущает многих программистов, которые привыкли к передаче объектов по ссылке.

- веб-приложения, написанные на PHP, зачастую имеют проблемы с безопасностью.

После проведенного анализа языка программирования PHP [7], был сделан вывод, что с точки зрения решения данной задачи этим средством можно реализовать поставленную цель.

## **2.7 Требования к составу выполняемых функций**

Для создания функционала будущей программы были поставлены следующие задачи:

- шаблоны-заготовки с готовым текстом ответа. Отправка писем-шаблонов из CRM должна быть автоматизированной. Такие письма можно привязать к любому событию. Это может быть оповещение клиентов о

предоставляемых скидках, может быть связано с переходом конкретного клиента из одной категории другой, уведомление о получении документов, напоминание о заказанной услуге и другое. Благодаря такой рассылки абитуриент или студент своевременно получит важную информацию;

- автоответ, например, если студент сделал запрос в нерабочее время, работник не сможет моментально среагировать на его сообщение, поэтому CRM-система может отправить автоответ, что его письмо было получено, но, к сожалению, связаться с ним не могут, так рабочее время закончилось, позже с ним обязательно свяжутся. Событие, которое вызывает автоответ, является внутренним для университета, например, если в CRM-системе сменится ответственный куратор по контакту, то абитуриенту придёт оповещение об этом. Автоматически может заносить в отдельную группу студентов, у которых имеется задолженность, им придёт автоматическое уведомление системы о сумме, которую следует погасить и о необходимых сроках;

- рассылки по базе клиентов;
- рассылка для нескольких условий.

#### Администрирование.

В разрабатываемом сервисе будет два типа доступа:

Доступ для всех. Доступ будет настроен для всех сотрудников, например, на уровне чтения или редактирования. Чаще всего публичный доступ настраивают ко входящим письмам от новых заказчиков, которые будут попадать в корпоративный почтовый ящик. Каждый пользователь сможет зайти и прочитать уведомления в данном почтовом ящике, просмотреть запросы от потенциальных заказчиков, забрать себе в работу лид. Этот подход является частью системы работы с лидами по принципу – «кто увидел, тот и забрал».

Доступ только для ответственных лиц. Входящую почту от клиентов, с которыми уже состоялось общение, будут видеть лица, ответственные за работу с ними. Часто ответственными лицами назначают персонального куратора, который ведёт работу с контактом и его непосредственного руководителя, который контролирует работу сотрудника. Остальной персонал эту почту

прочитать не смогут. Такое ограничение устанавливают в компаниях, где требуется строгое разграничение для работы с определёнными клиентами.

Чтобы посмотреть переписку или написать ответ куратору не требуется вход в почтовый ящик. К карточке клиента в CRM будет прикреплена необходимая переписка, где без каких-либо затруднений сотрудник сможет прочитать сообщения и написать ответ.

### 2.3 Создание контекстной панели

Прежде чем приступить к разработке необходимого сервера, нужно иметь представление, как будет выглядеть его контекстная панель и какие будет иметь функции.

Таблица 2.1 - Контекстная панель разрабатываемого приложения

<b>Кнопка</b>	<b>Описание</b>
Список	Переход на страницу со списком рассылок
Добавить	Создание новых рассылок
Удалить	Удаление редактируемых рассылок
Проверить	Переход на страницу для проверки шаблона автоматической генерации выпусков

Таблица 2.2 - Закладка «Рассылка»

<b>Поле</b>	<b>Описание</b>
Активна	Показывает активность рассылки. Если рассылка активна, можно подписаться на нее
Выводить в списке публичных рассылок	Показывает, разрешен или запрещен показ ссылки для подписки на данную рассылку на публичном разделе сайта
Показывать рассылку	Сайт рассылки

<b>Поле</b>	<b>Описание</b>
на сайте	
Название	Название рассылки
Сортировка	Показывает объём сортировки списка рассылки. Чем больше это значение, тем выше располагается рассылка в списке
Символьный код	Символьный код рассылки.
Описание для подписчиков (до 1000 симв.)	Описание рассылки, отображаемое посетителям сайта.
Автоматическая рассылка	Возможность создания автоматической рассылки. Рассылка будет отправляться по расписанию.

Таблица 2.3 - Закладка «Автоматическая генерация»

<b>Поле</b>	<b>Описание</b>
<b>Расписание</b>	
Время последней генерации выпуска (DD.MM.YYYY HH:MI:SS)	Дата и время последней генерации сообщений автоматической рассылки. Когда создаётся новая автоматическая рассылка, в поле будет подставлена дата и время создания рассылки. Значение поля используется для задания даты, начиная с которой сообщения автоматической рассылки будут генерироваться и рассылаться.
Дни месяца (например 1,10-20,25)	Список чисел месяца, по которым будет происходить автоматическая генерация выпусков рассылки. Возможно указание как отдельных дат через запятую (например. 1,8,10), так и диапазона дат (например: 19-25, 27-30).
Дни недели	Дни недели для выпуска рассылки.
Время	Время, когда будет автоматически генерироваться выпуск

<b>Поле</b>	<b>Описание</b>
(например 8:00,17:00)	рассылки. Возможно указание списка значений (например, 9:00, 14:00, 21:00).
<b>Шаблон для автоматической генерации выпусков</b>	
Доступные шаблоны	Позволяет выбрать шаблоны из доступных, на основании которого будут автоматически генерироваться выпуски данной рассылки.  Если шаблоны будут отсутствовать, пользователь увидит сообщение об отсутствие шаблонов для автоматической генерации выпуска.
<b>Поля письма</b>	
От кого	Необходимо написать адрес отправителя, от которого будут приходить рассылки.  По умолчанию будет стоять адрес, заданный в настройках модуля Рассылки.

Вкладка доступна только при создании или редактировании автоматической рассылки.

Выпуск рассылки отправляется:

- подписчикам на определённые рассылки;
- пользователям определённых групп прав;
- на произвольные адреса.

Таблица 2.4 - Кнопки управления для параметров рассылки

<b>Кнопка</b>	<b>Описание</b>
Сохранить	Позволяет сохранить параметры рассылки и перейти к отображению списков рассылок.
Применить	Позволяет сохранить параметры рассылки.
Отменить	Отмена изменений, внесённых в параметры рассылки.

В таблицах описано примерное представление о функционале контекстного меню будущего модуля.

### **Выводы по 2 главе**

Во данной главе были поставлены и рассмотрены главные задачи для разработки требуемого сервиса, было выбрано средство реализации и составлена структура контекстной панели.

# Глава 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МАССОВЫХ РАССЫЛОК

## 3.1 Выбор средств реализации

Для написания и редактирования кода была выбрана интегрированная среда разработки PhpStorm, построенная на основе платформы IntelliJ IDEA. Сравнивая простые редакторы, например, Notepad++, Sublime Text, Text Mate, Coda 2, Espresso, у PhpStorm функционал удобнее и быстрее.

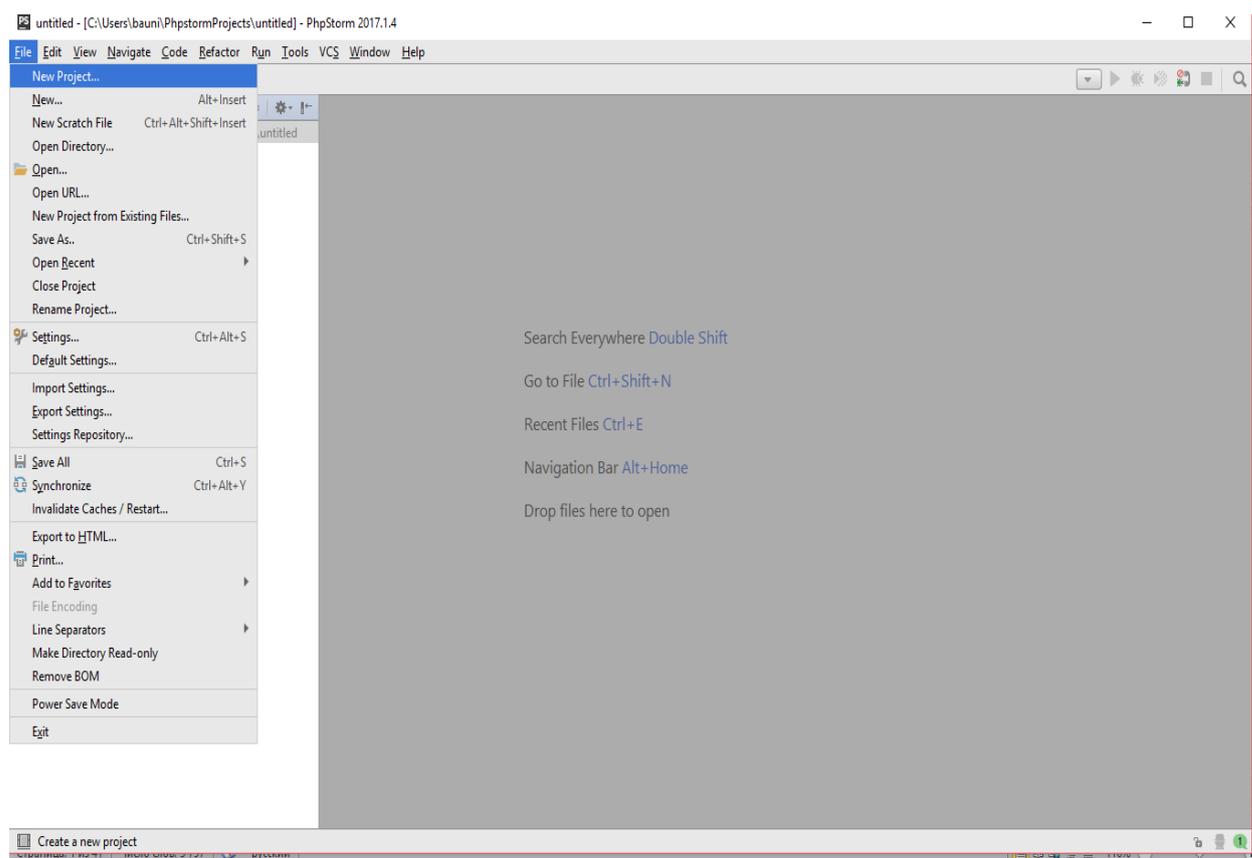


Рисунок 3.1 – Начало работы в PhpStorm

Рассылка должна быть, реализована на web-сервере, для этого необходимо установить web-сервер на локальную машину. У большинства серверов функциональные возможности практически одинаковы. Для установки был выбран Xampp. Благодаря удобному расположению каталогов, сразу видно, что динамические и статические данные разделены по разным папкам, что упрощает синхронизацию данных между различными копиями комплекса и создаёт экономию места при резервном копировании.

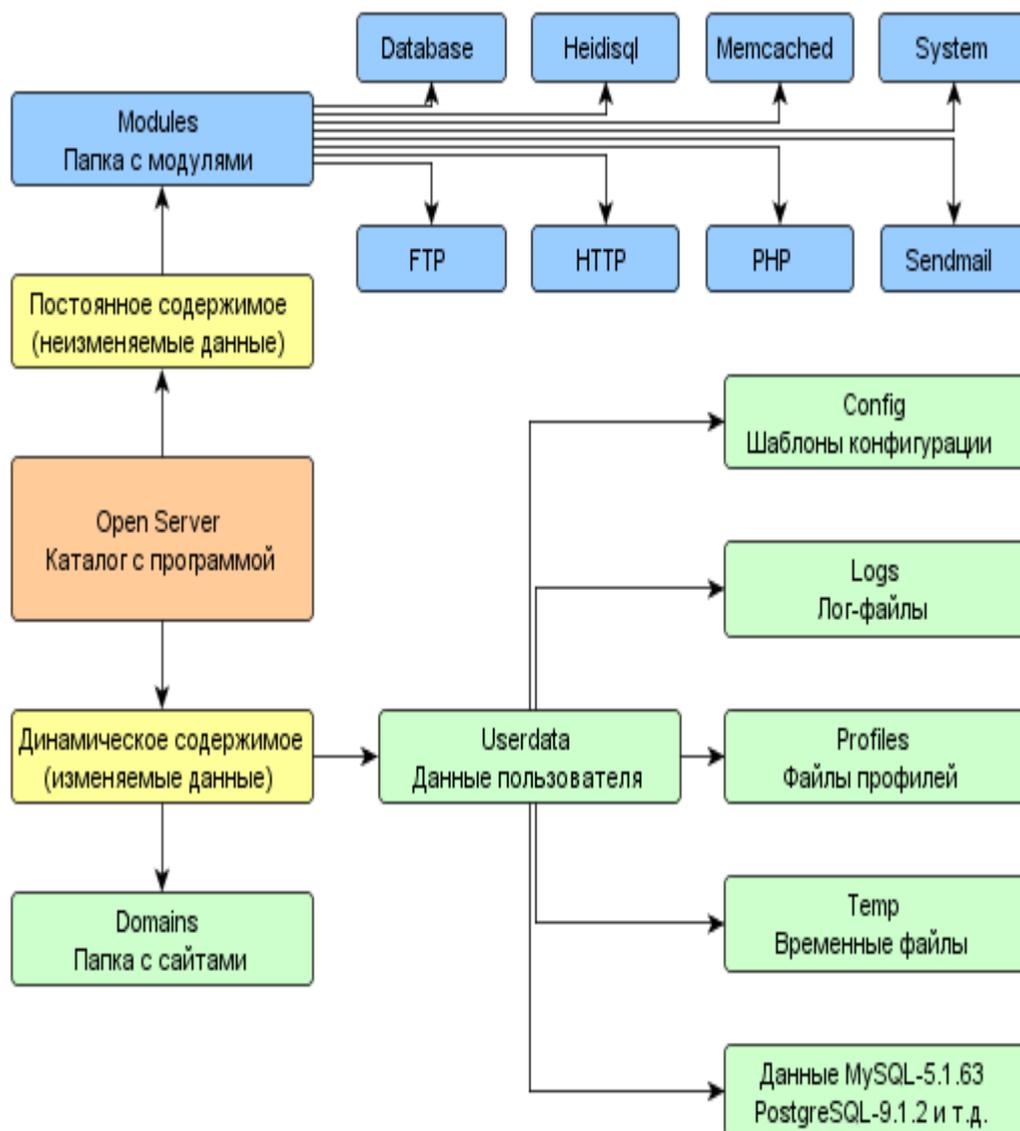


Рисунок 3.2 – Архитектура хатрр

### Установка сервера.

Для установки выбранного сервера на локальную машину требуется прочтение мануалов, которые представлены на официальном сайте разработчика. Нужно обратиться к системным требованиям и убедиться, что локальная машина, на которую было принято установить данный сервер, удовлетворяем им.

Системные требования:

Необходимый минимум системных ресурсов: 200 Мб RAM и 1 Гб на HDD;

Windows (32-bit или 64-bit): Windows 10 / Windows 8 / Windows 7 / Windows Server 2008 / Windows Vista / Windows XP SP3;

Должен быть установлен набор библиотек Microsoft Visual C++ 2005-2008-2010 Redistributable Package x86;

После изучения системных требований, сделан вывод, что используемый компьютер подходит для поставленной цели. Выбранное программное обеспечение после обновления сервиса станет портативным комплексом, который не будет требовать установку. Для того чтобы начать процесс настройки, необходимо будет разместить дистрибутив на локальном компьютере.

После того, как исполняемый файл хampp-win32-5.6.30-1-VC11-installer.exe будет запущен от имени администратора, программа автоматически сворачивается в трей, через которую можно зайти в панель управления локальной машиной и подключить или отключить необходимые составные модули. Изначально наш сервер находится в выключенном состоянии.

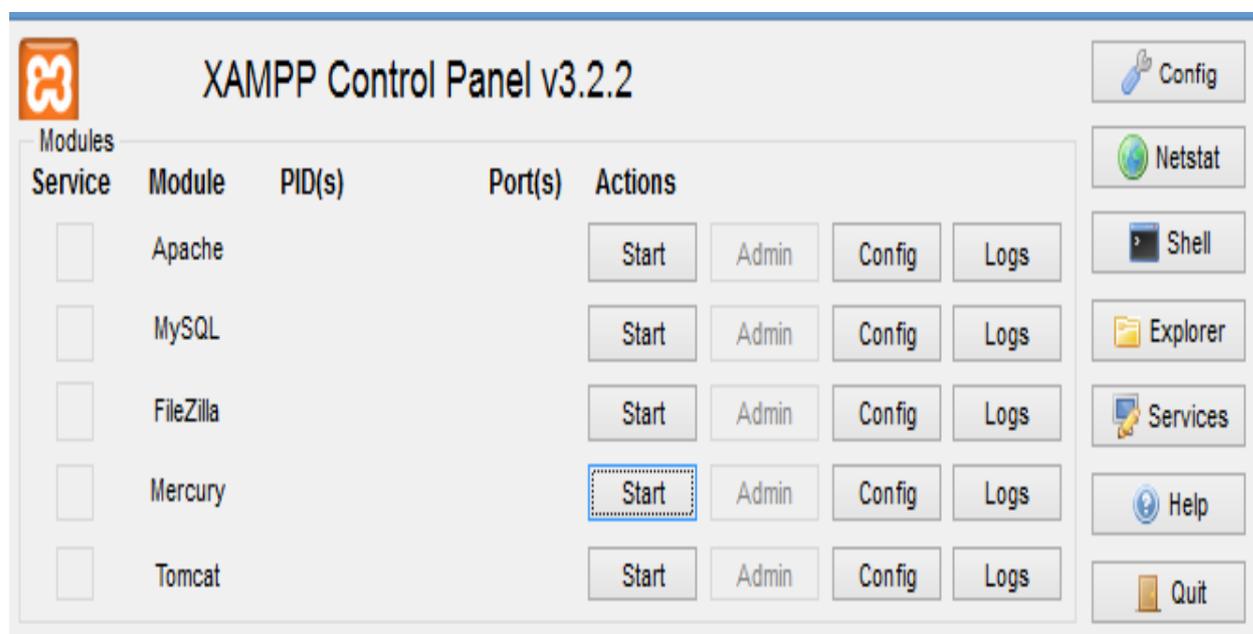


Рисунок 3.3 – Панель управления локальной системы

Перед запуском требуется зайти в раздел настроек, чтобы посмотреть, удовлетворяют ли стандартные настройки поставленным необходимым требованиям. В окне настроек будет находиться подразделы: основные, сервер,

модули, меню, кодировки, FTP сервер, почта, закладки, автозагрузка, разное, планировщик заданий, алиасы, домены.

В разделе «основные» языком интерфейса приложения для удобства можно выбрать язык «Russian» или другой предлагаемый. В следующей вкладке настроек находят настройки для сервера, а именно выбор удовлетворяющего IP-адреса, настройка портов для избегания ошибок при запуске. Следующая вкладка – настройка модуля, очень важная для функционала сервера.

В этих настройках выбираются версии, представленных модулей для использования на данном сервере. Во вкладке «HTTP» необходимо выбрать «Apache-2.4» и поставить галочку «вести лог запросов». Во вкладке PHP указать версию фреймворка Yii, удовлетворяющую поставленным условиям – 2.0.2. Официальный сайт разработчиков Yii 2, утверждает, что данный фреймворк работает на версии PHP от 5.4, значит, нужно выбрать версию выше 5.4 или 5.4 . Далее следует сделать выбор между предложенными СУБД MySQL / MariaDB или PostgreSQL.

Другие вкладки в меню настроек не несут функциональной значимости для разрабатываемого сервиса. После выбранных настроек следует запустить сервер, чтобы убедиться, что проблемы с запуском отсутствуют, если в трее загорелся зеленый флажок — значит сервер запущен успешно.

Следующий шаг для реализации главной цели бакалаврской работы - установка и настройка Yii 2.

#### Установка фреймворка

Фреймворк - программное обеспечение, которое помогает в разработке проектов, служит для них «каркасом».

Фреймворк привязывается к языкам программирования, на которых он написан, в данном случае: php-framework. Обычно он создаётся одним или несколькими разработчиками для того, чтобы в будущем использовать для своих целей. Некоторые фреймворки выкладывают в общий доступ.

Пример php-фреймворка — известный Yii. Фреймворк существенно облегчает разработку сайта по сравнению с тем, если все делать его только на php. В фреймворках имеется ряд готовых решений, поэтому требуется только научиться работать с ними.

```
namespace app\controllers;

use Yii;
use yii\base\Model;
use yii\db\Exception;
use yii\db\Expression;
use yii\web\Response;
use yii\filters\ContentNegotiator;
use yii\rest\ActiveController;
use yii\helpers\Url;
use yii\web\ServerErrorHttpException;
```

Рисунок 3.4 – Фрагмент программного кода php с использованием yii

Сборка Yii 2 имеет два варианта, basic и advanced. Basic – это минимальная сборка, а advanced продвинутая сборка, которая включает в себя готовые решения некоторых задач. Еще одно преимущество продвинутой сборки, это разделение пользовательской части (frontend) и модуля управления (backend), именно это различие сборок сподвигло к установке именно продвинутого варианта фреймворка.

После установки продвинутого шаблона было получено несколько возможностей, которые помогут при разработке программы:

- разделенные области пользователей и администраторов;
- встроенная интеграция Twitter Bootstrap;
- автоматическая генерация моделей, контроллеров и отображений.

После установки фреймворка и запуска сервера, можно посмотреть результат, зайдя по адресу, который был выбран в настройках сервера, и указать порт. Зайдя по нужному адресу, откроется стартовая страница.

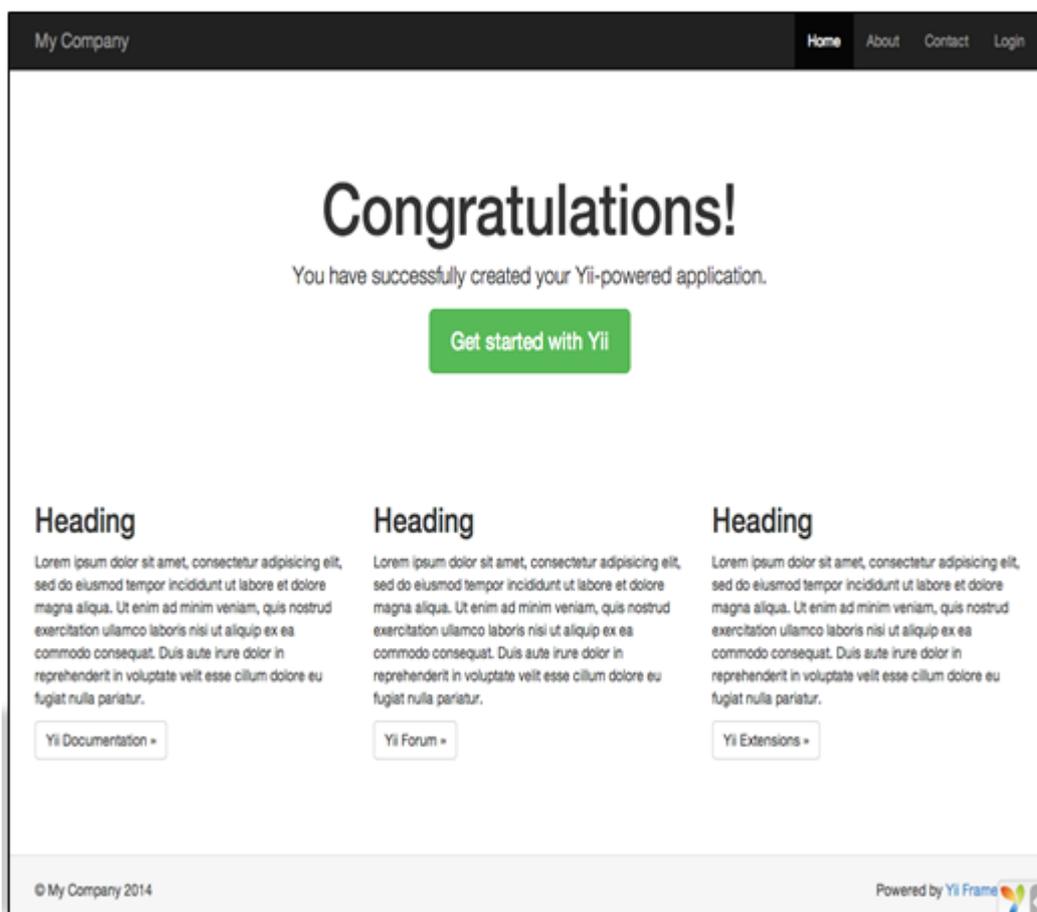


Рисунок 3.5 – Вступительная страница

MCV система, входящая в Yii framework включает в себя модели, контроллеры и отображения. Для создания модели используется встроенный в Yii сервис, Gii. Gii сервис является web-ориентированным генератором кода для приложений на Yii. Он подходит для быстрого создания моделей, форм, модулей, CRUD и других. Но для создания моделей, необходимо подключить исходный фреймворк к базе данных. Для этого в папке config, в файле bd.php вводятся данные подключения к имеющейся базе. После успешного подключения фреймворка к базе данных, нужно создать базу данных через СУБД. После выбора активной базы данных внутри фреймворка можно приступить к созданию модели, через Gii генератор.

### 3.2 Реализация сервиса по организации массовых рассылок

Для представления статистической структуры модели сервиса приведена часть диаграммы классов api-контроллера (рис. 3.6).



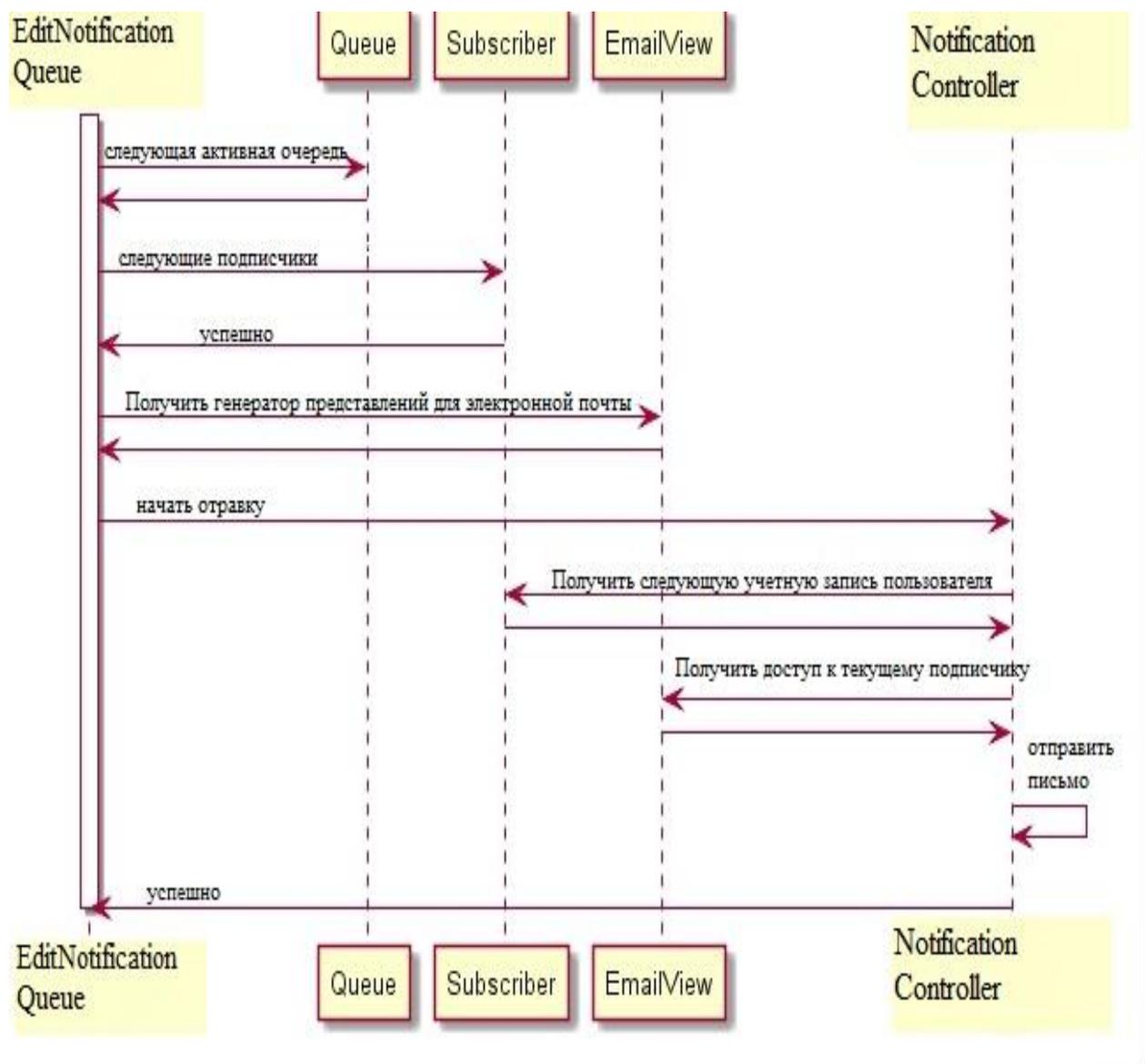


Рисунок 3.8 – Диаграмма состояния работы модуля

Существует класс NotificationController, который отвечает за отправку писем и занесение данных об этом в статистику.

NotificationController не выбирает подписчиков, по которым рассылаться сообщения, или каким алгоритмом генерировать html письма. Это передается как параметры из скрипта Editnotificationqueue, так же он управляет и контролирует процессом рассылок.

Editnotificationqueue запускается периодически по крону или по команде ИЗ КОНСОЛИ

Editnotificationqueuee получает из базы данных письмо, которое нужно разослать, условия рассылки, например, какие шаблоны подключать и список подписчиков, которым письмо уйдет

Все полученные данные передаются в NotificationController.

Editnotificationqueuee отвечает за корректность завершения работы и сохранения данных.

Если во время вызова окажется, что имеется возможность разослать больше сообщений, чем было выбрано из базы данных, то Editnotificationqueuee повторит все операции, пробуя отправить дополнительные письма.

В данном разделе будут описаны классы, и функции, которые были написаны в ходе разработки массовых email-рассылок.

Рассмотрим список всех основных функций, которые мы будем использовать в нашем проекте, для непосредственного вызова формы для рассылки писем:

- MailAReminderToPayTheDebt
- MailToDebtPaymentThisMonth
- MailDebtMore10000
- MailAcademicLeave
- MailExpelledStudent
- ActionAddnewmail
- Addaneventtoqueueelocal;
- Addanotificationqueuee;
- Editnotificationqueuee.

Первые пять функций делают выборки из базы по необходимым условиям. Также они записывают необходимые данные в базу для последующей рассылки. Для вызова всех этих функций, в нашем проекте применяется ActionAddnewmail.

```

public function actionAddnewmail() {
    $this->MailAcademicLeave ();
    $this->MailAReminderToPayTheDebt ();
    $this->MailDebtMore10000 ();
    $this->MailExpelledStudent ();
    $this->MailToDebtPaymentThisMonth ();
}

```

Рисунок 3.9 – Фрагмент кода разрабатываемого сервиса

Функция `Addaneventtoqueuelocal` записывает в базу необходимые данные для отправки писем.

```

protected function addaneventtoqueuelocal($login, $type, $options) {
    $params = Yii::$app->getRequest()->getBodyParams();
    $this->logApi( func: 'abiturient/AddAnEventToQueue', $params);

    if(is_null($login) or is_null($type) or is_null($options)) return array('error'=>true, 'message'=>'Отсутствует один из параметров.');
```

```

    else {
        $ev = NotificationQualifier::find()->where(['CODE'=>$type])->One();
        $eventqr = EventQualifier::find()->where(['id'=>$ev['id_eq']])->one();
        if(!$eventqr) return array('error'=>true, 'message'=>'Указанное событие не найдено.');
```

```

        $eventq = new EventQueue();
        $eventq->login = $login;
        $eventq->id_event = $ev['id_eq'];
        $eventq->addition_of_events = date( format: 'Y-m-d H:i:s', timestamp: time() + 4 * 60 * 60);
        $eventq->processed = 'N';
        $eventq->options = $options;

        if($eventq->save()) return true;
        else return array('error'=>true, 'message'=>'Ошибка при сохранении в базу.');
```

```

    }
}

```

Рисунок 3.10 – Программный код функции `Addaneventtoqueuelocal`

Функция `Addnotificationqueue` обрабатывает данные и добавляет записи в очередь рассылок.

```

public function actionAddnotificationqueue()
{
    $eq = EventQueue::find()->where(['processed' => 'N'])->All();

    foreach ($eq as $eqs) {
        $nq = NotificationQualifier::find()->where(['id_eq' => $eqs['id_event']])->All();
        for ($i = 0; $i < count($nq); $i++) {
            $noa = new NotificationOfAll();
            $noa->login = $eqs['login'];
            $noa->id_notification_qualifier = $nq[$i]['id'];
            $noa->extra_options = $eqs['options'];
            $noa->id_eq = $eqs['id'];
            $noa->save();

            $eqs->processed = 'Y';
            $eqs->save();
        }
    }
}

```

Рисунок 3.11 – Программный код функции Addnotificationqueue

Функция Editnotificationqueue непосредственно выполняет рассылку, код функции приведён в приложении А.

Так же, рассмотрим список всех основных функций, которые мы будем использовать в нашем проекте, для sql выборки из базы, на основании которой нужно делать рассылки:

### MailToDebtPaymentThisMonth;

```

//Отправка писем студентам, которым не хватает денег для оплаты текущего месяца обучения, за исключением должников
protected function MailToDebtPaymentThisMonth(){
    $sql = "select distinct fsfio, studid, login, oplata from( select alls.fsfio, alls.studid, case when alls.login is null then alls.gall
se alls.login end as login, alls.oplata from ( " .
        "select stud.fsfio, stud.studid, stud.login as login, stud.gal_login as gall, nach.nach, paid.paid, nach.nach + paid.paid as dolg, " .
        "pr.price - (nach.nach + paid.paid) as oplata, pr.price from gal_public.v_stud_do stud " .
        "left join (select nach.fnrec, sum(nach.ss) as nach from gal_public.mv_stud_debt_nach nach whereaid.ss) as paid from gal_public.mv_stud_debt_paid " .
        "group by nach.fnrec) paid on paid.fnrec = stud.studid " . dat < to_date('01.' . date('m.Y') . ',' . 'dd.mm.YYYY') " .
        "left join ( glue: select paid.fnrec, sum(p
        "left join (select s.studid, p.fprice as price " .
        "from gal_public.v_stud_do s, GAL_WORK.PRICES p, GAL_WORK.KLPRICE kl, gal_work.pricetobasecurr pbc " .
        "left join gal_work.catalogs kat on pbc.fcrank = kat.fnrec " .
        "where p.FCKLPRICE = kl.FNREC " .
        "and p.fnrec = pbc.FCPPRICE " .
        "and pbc.FCBASECURR != '8000000000000000' and s.FCCURR = pbc.FCBASECURR " .
        "and (kat.fnrec=s.FCPAIDCATEGORY or (kat.FNREC is null and s.FCPAIDCATEGORY is null) " .
        "or (s.FCPAIDCATEGORY = '8000000000000000' and kat.FNREC is null) " .
        "and gal_work.to_atldate(sysdate) BETWEEN kl.FDBEG and kl.f dend) pr on pr.studid = stud.studid) alls " .
        "where alls.dolg > 0 and alls.dolg < alls.price);

    $conn = Yii::$app->galdb;
    $command = $conn->createCommand($sql);
    $results = $command->queryAll();

    foreach($results as $all) {
        if(is_null($all['LOGIN'])) continue;
        $json = json_encode(array('srok' => '17.12.2016', 'pay' => $all['OPLATA']));
        $this->AddaneventtoqueueLocal($all['LOGIN'], type: 'regular_pay', $json);
    }
}

```

Рисунок 3.12 – Программный код функции MailToDebtPaymentThisMonth

### MailAReminderToPayTheDebt;

```

//Напоминание о задолженности по оплате обучения. Отправляется после 17 числа за текущий месяц
protected function MailReminderToPayTheDebt(){
    $sql = "select distinct fsfio, studid, login, oplata from( select alls.fsfio, alls.studid, case when alls.login is null then alls.gall else alls.
login end as login, alls.oplata from ( " .
    "select stud.fsfio, stud.studid, stud.login as login, stud.gal_login as gall, nach.nach, paid.paid, nach.nach + paid.paid as dolg, " .
    "pr.price - (nach.nach + paid.paid) as oplata, pr.price from gal_public.v_stud_do stud " .
    "left join (select nach.fnrec, sum(nach.ss) as nach from gal_public.mv_stud_debt_nach nach where dat < to_date('17,' . date( format: 'm.Y') . ',' , 'dd.mm.YYYY') " .
    "group by nach.fnrec) nach on nach.fnrec = stud.studid " .
    "left join (select paid.fnrec, sum(paid.ss) as paid from gal_public.mv_stud_debt_paid paid " .
    "group by paid.fnrec) paid on paid.fnrec = stud.studid " .
    "left join (select s.studid, p.fprice as price " .
    "from gal_public.v_stud_do s, GAL_WORK.PRICES p, GAL_WORK.KLPRICE kl, gal_work.pricetobasecurr pbc " .
    "left join gal_work.catalogs kat on pbc.fcrank = kat.fnrec " .
    "where p.FCKLPRICE = kl.FNREC " .
    "and p.fnrec = pbc.FCPRISE " .
    "and pbc.FCBASECURR != '8000000000000000' and s.FCCURR = pbc.FCBASECURR " .
    "and (kat.fnrec=s.FCPAIDCATEGORY or (kat.FNREC is null and s.FCPAIDCATEGORY is null) " .
    "or (s.FCPAIDCATEGORY = '8000000000000000' and kat.FNREC is null) ) " .
    "and gal_work.to_atldate(sysdate) BETWEEN kl.FDBEG and kl.fvend) pr on pr.studid = stud.studid) alls " .
    "where alls.dolg > 0 and alls.dolg < alls.price)";

    $conn = Yii::$app->galdb;
    $command = $conn->createCommand($sql);
    $results = $command->queryAll();

    foreach($results as $all) {
        if(is_null($all['LOGIN'])) continue;
        $json = json_encode(array('pay' => $all['OPLATA']));
        $this->addaneventtoqueue($all['LOGIN'], type: 'dolg', $json);
    }
}

```

Рисунок 3.13 – Программный код функции MailReminderToPayTheDebt

## MailDebtMore10000;

```

//Отправка писем студентам, у которых долг более 10000
protected function MailDebtMore10000(){
    $sql = "select distinct fsfio, studid, login, dolg from ( select alls.fsfio, alls.studid, " .
    "case when alls.login is null then alls.gall else alls.login end as login, " .
    "alls.dolg from ( select stud.fsfio, stud.studid, stud.login as login, stud.gal_login as gall, nach.nach, paid.paid, nach.nach + paid.paid as dolg, " .
    "pr.price - (nach.nach + paid.paid) as oplata, pr.price from gal_public.v_stud_do stud " .
    "left join (select nach.fnrec, sum(nach.ss) as nach from gal_public.mv_stud_debt_nach nach where dat < to_date('02,' . date( format: 'm.Y') . ',' , 'dd.mm.YYYY') " .
    "group by nach.fnrec) nach on nach.fnrec = stud.studid " .
    "left join (select paid.fnrec, sum(paid.ss) as paid from gal_public.mv_stud_debt_paid paid " .
    "group by paid.fnrec) paid on paid.fnrec = stud.studid " .
    "left join (select s.studid, p.fprice as price " .
    "from gal_public.v_stud_do s, GAL_WORK.PRICES p, GAL_WORK.KLPRICE kl, gal_work.pricetobasecurr pbc " .
    "left join gal_work.catalogs kat on pbc.fcrank = kat.fnrec " .
    "where p.FCKLPRICE = kl.FNREC and p.fnrec = pbc.FCPRISE and pbc.FCBASECURR != '8000000000000000' and s.FCCURR = pbc.FCBASECURR " .
    "and (kat.fnrec=s.FCPAIDCATEGORY or (kat.FNREC is null and s.FCPAIDCATEGORY is null) " .
    "or (s.FCPAIDCATEGORY = '8000000000000000' and kat.FNREC is null) ) " .
    "and gal_work.to_atldate(sysdate) BETWEEN kl.FDBEG and kl.fvend) pr on pr.studid = stud.studid) alls " .
    "where alls.dolg < 0 and alls.dolg < -10000)";

    $conn = Yii::$app->galdb;
    $command = $conn->createCommand($sql);
    $results = $command->queryAll();

    foreach($results as $all) {
        if(is_null($all['LOGIN'])) continue;
        $json = json_encode(array('dolg' => $all['DOLG'], 'srok'=>'15.06.2017', 'otch'=>'15.12.2017'));
        $this->addaneventtoqueue($all['LOGIN'], type: 'dolg_10000', $json);
    }
}

```

Рисунок 3.14 – Программный код функции MailDebtMore10000

## MailAcademicLeave;

```

//Отправка писем студентам, перед выходом из академ. отпуска
protected function MailAcademicLeave(){
    $sql = "select studid, login, dateand from ( " .
        "select stud.studid, s.fcpersons, max(tb.ord_date), stud.login, gal_work.to_oradate2(acat.fplanyearend) as dateand " .
        "from gal_public.v_stud_do stud " .
        "left join gal_work.u_student s on s.fnrec = stud.studid " .
        "left join gal_public.tb_app tb on tb.person_id = s.fcpersons " .
        "left join gal_work.communications com on com.fperson = s.fcpersons " .
        "left join gal_work.vacations acat on acat.fcprikaz = tb.rpd_id and acat.fperson = s.fcpersons " .
        "where (s.fdisdate <> 0 or s.fdisdate is not null) " .
        "and tb.rpd_name like '%'. iconv( in_charset: 'utf8', out_charset: 'cp1251', str: 'назем') .'% ' and tb.rpd_name like '%'. " .
        "iconv( in_charset: 'utf8', out_charset: 'cp1251', str: 'отмыск') .'% ' and tb.rpd_name not like '%'. iconv( in_charset: 'utf8', out_charset: 'cp1251', str: 'Выход') .'% ' " .
        "and com.fcomtype = '800000000000034C' " .
        "and com.femail is not null " .
        "and acat.fplanyearend is not null " .
        "group by stud.studid, s.fpic, tb.rpd_name, com.femail, stud.login, tb.rpd_id, s.fcpersons, acat.fplanyearend) " .
        "where add_months(to_date(dateand),-1) = sysdate";

    $conn = Yii::$app->galdb;
    $command = $conn->createCommand($sql);
    $results = $command->queryAll();

    foreach($results as $all) {
        if(is_null($all['LOGIN'])) continue;
        $json = json_encode(array('srok' => $all['DATEAND']));
        $this->AddaneventtoqueueLocal($all['LOGIN'], type: 'academ_otp',$json);
    }
}

```

Рисунок 3.15 – Программный код функции MailAcademicLeave  
MailExpelledStudent.

```

//Отправка писем отчисленным студентам отчисленным 4 года назад
protected function MailExpelledStudent(){
    $sql = "select stud.login, gal_work.to_oradate2(s.fdisdate) as dates from gal_public.v_stud_do stud " .
        "left join gal_work.u_student s on s.fnrec = stud.studid " .
        "left join gal_public.tb_app tb on tb.person_id = s.fnrec " .
        "where (s.fdisdate <> 0 and s.fdisdate is not null) " .
        "and tb.rpd_name like '%'. iconv( in_charset: 'utf8', out_charset: 'cp1251', str: 'Отчислен') . '% ' " .
        "and add_months(gal_work.to_oradate2(s.fdisdate),-12) = sysdate";

    $conn = Yii::$app->galdb;
    $command = $conn->createCommand($sql);
    $results = $command->queryAll();

    foreach($results as $all) {
        if(is_null($all['LOGIN'])) continue;
        $json = json_encode(array('srok' => $all['DATES']));
        $this->AddaneventtoqueueLocal($all['LOGIN'], type: 'recovery',$json);
    }
}

```

Рисунок 3.16 – Программный код функции MailExpelledStudent

Все вышеперечисленные функции выполняют выборки из базы данных по необходимым условиям, которые задаёт администратор системы.

Разрабатываемый сервис для email-рассылок был успешно интегрирован с CRM-системой, рассылка осуществляется успешно, с использованием требуемых шаблонов.

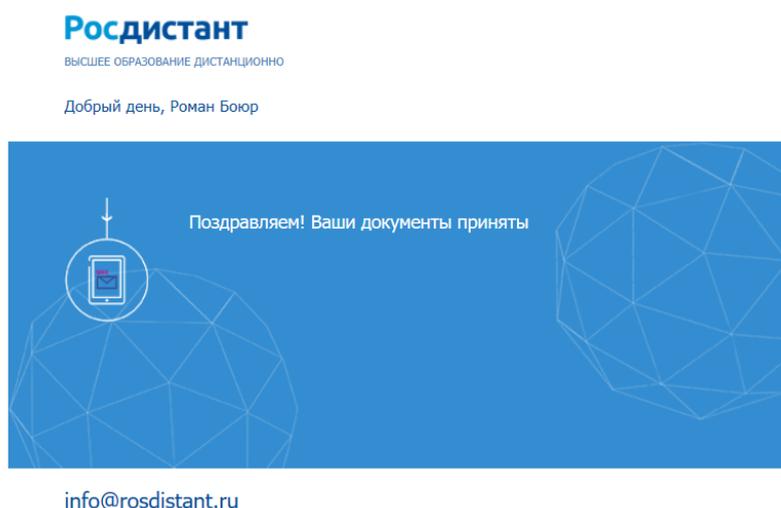


Рисунок 3.17 – Полученное сообщение

Расчет экономии за счет увеличения производительности труда пользователя.

При экономии  $i$ - вида с применением программы пользователь экономит  $\Delta T_i$ , часов, повышение производительности труда  $P_i$  (в %) определяется по формуле:

$$P_i = \left( \frac{\Delta T_j}{F_j - \Delta T_j} \right) \times 100$$

$F_j$  - время, которое планировалось пользователем для выполнения работы  $j$ -вида до внедрения программы (часы).

Таблица 3.1- Таблица работы пользователя

№ п/п	Вид работы	До автоматизации, Мин $F_j$	Экономия времени, мин $\Delta T$	Повышение производительности $P_i$ (%)
1	Ввод текста	40	20	100
2	Отправка отдельного сообщения каждому адресату из нужной группы	30	15	100

№ п/п	Вид работы	До автоматизации, Мин Fj	Экономия времени, мин $\Delta T$	Повышение производительности Pi (%)
3	Анализ и выборка данных	44	10	300

В результате расчетов экономической эффективности и внедрения средства автоматизации становится видно, что это выгодно, потому что задачи начинают выполняться быстрее, а времени тратится меньше.

### **Выводы по 3 главе**

В последней главе была описана реализация поставленных задач и её результат. Была рассчитана экономическая эффективность внедрения данного средства автоматизации.

## ЗАКЛЮЧЕНИЕ

Качество и оперативность предоставляемых услуг – залог успеха любой компании. Одним из способов повышения удобства и эффективности предоставления услуг является создание сервиса для массовых рассылок.

Настоящая выпускная квалификационная работа посвящена теме разработки сервиса для организации массовых рассылок.

Выпускная квалификационная работа выполнена на основе материалов преддипломной практики, проведенной в ФГБОУ ВО «Тольяттинский государственный университет», которое является одним из ведущих страховщиков Поволжья.

В ходе решения обозначенной задачи и проведенного исследования получены следующие результаты:

- произведен анализ предметной области и разработана концептуальная модель email-сервиса;
- Разработан и интегрирован сервис для email-рассылок на базе Тольяттинского Государственного Университета

Разработанный сервис прошёл успешное тестирование в ФГБОУ ВО «Тольяттинский государственный университет».

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

### *Учебники и учебные пособия*

1. Олейник, П.П. Корпоративные информационные системы: учеб. пособие / П. П. Олейник. – СПб.: Питер, 2012. – 176 с.
2. Корнеев, И. К. Информационные технологии : учебное пособие / И. К. Корнеев./ - М.: Проспект, 2007.- 224 с.
3. Скляр, Д. PHP. Рецепты программирования / Д. Скляр, А. Трахтенберг. – 3-е изд. – СПб.: Изд-во «Питер», 2015. – 784с.
4. Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд. – СПб.: Изд-во «БХВ-Петербург», 2015. – 766 с.
5. Куроуз Д. Компьютерные сети. Многоуровневая архитектура Интернета: 2-е издание / Д. Куроуз, К. Росс./ – СПб.: Питер, 2004. – 765с.
6. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста /Р. Мартин./ – СПб.: Питер, 2010. - 464с.: ил.
7. Станек У. Windows Server 2008. Справочник администратора / У. Станек./ – М.: Русская Редакция, 2009 - 688с.: ил.
8. Интернет как инструмент библиографического поиска / И. С. Галеева. - Санкт-Петербург : Профессия, 2007. - 247 с. : ил. - (Библиотека). - Библиогр.: с. 182-201. - Прил.: с. 202-246.

### *Электронные ресурсы*

9. Информационный сайт системы Битрикс24 [Электронный ресурс]. – Режим доступа: <https://www.bitrix24.ru>
10. Информационный сайт системы 1С-Битрикс [Электронный ресурс]. – Режим доступа: <https://www.1c-bitrix.ru>
11. Корпоративный портал ТГУ [Электронный ресурс]. – Режим доступа: <http://portal.tltsu.ru>/Информационный сайт с публикациями email-сервисов [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru>

12. Руководство по PHP [Электронный ресурс]: PHP: Руководство по PHP – Manual / группа документирования PHP. – Электрон. дан. – [2016]. – Режим доступа: <http://php.net/manual/ru/index.php>

13. Справочное руководство по MySQL [Электронный ресурс]: MySQL Manual. – Электрон. дан. – URL: <http://www.mysql.ru/docs/man/>

14. Сайт Тольяттинского государственного университета [Электронный ресурс] // Тольяттинский государственный университет: История Тольяттинского государственного университета. – URL: <http://www.tltsu.ru> (1.06.2016)

15. Рудинский И. Д. Технология проектирования автоматизированных систем обработки информации и управления [Электронный ресурс] : учеб. пособие / И. Д. Рудинский. - Москва : Горячая линия - Телеком, 2011. - 304 с. - ISBN 978-5-9912-0148-3.

*Литература на иностранном языке*

16. David Skla. Learning PHP: A Gentle Introduction to the Web's Most Popular Language / David Skla. - O'Reilly, 2016. – 416с.

17. Masoud Kalali, Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON / Masoud Kalali – United Kingdom, Birmingham,: Packt Publishing, LTD, 2013. – 107 p.

18. Hudson O. Getting started with IntelliJ IDEA // O. Hudson, Birmingham: Packt Publishing, 2013. – 114p.

19. Spring Data / M. Pollack, O. Gierke, T. Risberg and other, - California: O'Reilly Media. - 1st edition. - 316 с.

20. Lockhart, J. Modern PHP. New Features and Good Practices / J. Lockhart. – 1st edition, O'Reilly Media, 2015.

## ПРИЛОЖЕНИЕ А

```
public function actionEditnotificationqueue()
{
    $noa = NotificationOfAll::find()->all();

    foreach($noa as $key) {
        $UtsCont = UtsCrmContact::find()-
>where(['UF_CRM_1435666743'=>$key['login']])->One();
        $Cont = CrmContact::find()->where(['ID'=>$UtsCont['VALUE_ID']])->one();

        $nq = NotificationQualifier::find()-
>where(['id'=>$key['id_notification_qualifier']])->one();
        $mh = new MailHelper();
        $sh = new SMSHelper();

        if($nq['type'] == 'mail'){
            if(is_null($key['extra_options']) or $key['extra_options'] == ''){
                $mess = $nq['template'];
            }
            elseif(json_decode($key['extra_options'],true)) {
                $arr = json_decode($key['extra_options'],true);
                if(gettype($arr) != 'array') {echo gettype($arr); continue;}
                $paramsto = array_values($arr);
                $paramsin = json_decode($nq['params'],true);
                $mess = preg_replace($paramsin,$paramsto,$nq['template']);
            }
            if(isset($mess)) {
                $results = $mh->SendingmailCRM($UtsCont['VALUE_ID'],'Сообщение
от Росдистант',$mess);
                $this->Addactotocontact($UtsCont->VALUE_ID,'Сообщение от
Росдистант',$mess);
            }
        }
        elseif($nq['type'] == 'sms') {
            if (is_null($key['extra_options']) or $key['extra_options'] == '') {
                $mess = $nq['template'];
            }
            elseif (json_decode($key['extra_options'], true)) {
                $arr = json_decode($key['extra_options'], true);
                if(gettype($arr) != 'array') {echo gettype($arr); continue;}
                $paramsto = array_values($arr);
                array_unshift($paramsto, $Cont['NAME'], $Cont['SECOND_NAME']);
                $paramsin = json_decode($nq['params'], true);
                $mess = preg_replace($paramsin, $paramsto, $nq['template']);
            }
        }
        $cphone = $UtsCont->contact->getMulti()
->where(['=', 'ENTITY_ID', 'CONTACT'])
->andWhere(['=', 'TYPE_ID', 'PHONE'])
->andWhere(['=', 'VALUE_TYPE', 'MOBILE'])
->one();
        if ($cphone) {
            $phone = $cphone->VALUE;
            if (isset($mess)) {
                $results = $sh->Sendnotification('devino', 'ROSDISTANT',
$mess, $phone);
                $this->Addrecordinhist($UtsCont->VALUE_ID, $mess,
'Отправлена смс');
            }
        }
    }
    elseif($nq['type'] == 'lk'){
        //отправляем сообщение на стену в ЛК
        $arr = json_decode($key['extra_options'], true);
        if(is_null($arr)) {
```

```

        $paramsto = array();
        $paramsto[0] = $Cont['NAME'];
        $paramsto[1] = $Cont['SECOND_NAME'];
        $paramsin = json_decode($nq['params'], true);
    }
    else {
        $paramsto = array_values($arr);
        $paramsin = json_decode($nq['params'], true);
        array_unshift($paramsto, $Cont['NAME'], $Cont['SECOND_NAME']);
    }
    $mess = preg_replace($paramsin, $paramsto, $nq['template']);

    if(isset($mess)) {
        $sch = curl_init('https://lk.rosdistant.ru/json_api/user/' .
$key['login'] . '/comment/');
        curl_setopt($sch, CURLOPT_POST, 1);
        curl_setopt($sch, CURLOPT_POSTFIELDS, array('LOGIN' =>
$key['login'],
        'COMMENT' => $mess));
        curl_setopt($sch, CURLOPT_RETURNTRANSFER, 1);
        $results = curl_exec($sch);
        curl_close($sch);
        $this->Addrecordinhist($UtsCont->VALUE_ID, $mess, 'Отправлена
запись в ЛК');
    }
    if(isset($results)) {
        $key->delete();
    }
}
}
}

```