Примерная тема: **«Подсистема управления рассылкой и учёта писем»**.

# АНАЛИЗ ТРЕБОВАНИЙ

## **Описание процесса отправки сообщений**

Обмен сообщениями является неотъемлемой частью любого взаимодействия. Если важна скорость взаимодействия, чаще всего стороны используют мессенджеры. Электронная почта же из-за своей специфики обязывает вкладывать достаточно смысла в каждое сообщение. Более того, электронная почта является важным звеном в регистрации и защите аккаунтов.

Процесс отправки сообщения разрабатываемым сервисом заключается в выборе подходящего сервиса длТея отправки сообщений, рассылке сообщений, проверке того, доставлено сообщение или нет.

### ***Основные понятия***

Дадим определения основным понятиям необходимым в дальнейшей работе.

*Сообщение –* определенная информация, которую необходимо передать.

*Отправитель* – лицо, отправляющее сообщение.

*Получатель* – лицо, которому адресовано отправляемое сообщение.

*Отправленное письмо –* письмо, отправленное получателем.

*Доставленное письмо* – отправленное письмо, дошедшее до получателя.

*Сервис для отправки сообщений* – стороннее ПО, предоставляющее функционал для рассылки сообщений. Чаще всего использование таких сервисов бесплатно до определенного объема трафика.

*WEB-интерфейс* – веб-страница или несколько веб-страниц, позволяющие взаимодействовать с нужным сайтом, сервисом.

*API (Application programming interface)* – набор функций, описывающих способы взаимодействия с ПО.

### ***Процесс отправки сообщения***

Изучим процесс отправки сообщения (рис. 1.2).

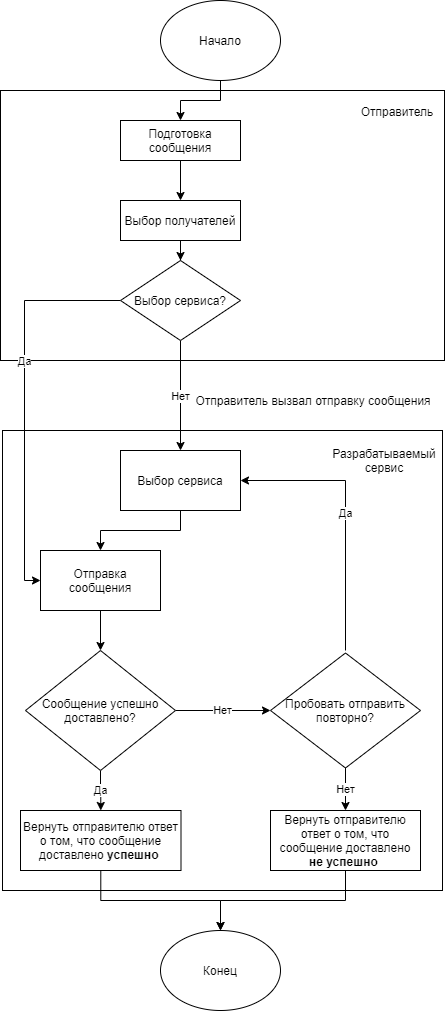


Рис. 1.1. Процесс отправки сообщения

В данном процессе принимает участие только отправитель. Причем отправка сообщения происходит асинхронно, т.е. отправитель вовлечен в процесс только до вызова отправки.

На первом шаге отправитель составляет сообщение, может прикрепить к нему какие-либо файлы, выбирает получателя или список получателей, при желании может выбрать предпочтительный сервис отправки и вызывает отправку сообщения.

Сервис пытается отправить сообщение получателю(-ям) посредством выбранного отправителем сервиса или сервисом, выбранным по умолчанию для адреса(-ов) выбранных получателя(-ей). Затем сервис проверяет, доставлено ли сообщение всем получателям. Если не доставлено, сервис отправляет сообщение заново в соответствии с выбранным алгоритмом (пробует тот же сервис определенное число раз или отправляет с помощью других сервисов).

На заключительном этапе сервис возвращает ответ отправителю – успешно ли доставлено сообщение (если необходимо, выводит, каким сервисом доставить не получилось, а какой сервис успешно доставил).

* + 1. ***Перечень проблем***

Выделим основные, часто встречающиеся проблемы, связанные с процессом отправки сообщений.

К техническим проблемам отправки сообщения можно отнести следующие:

* На стороне сервера отправителя (указаны неверные реквизиты, сервер не настроен);
* На стороне сервера получателя (письмо считается спамом; письмо – не спам, но всё равно было отклонено, несуществующий адрес, почтовый ящик получателя переполнен).

Как уже отмечалось выше, сервисы для отправки сообщений предоставляют свой функционал бесплатно до достижения определенного объема трафика (например, определенное количество отправленных сообщений в месяц). Таким образом, для использования некоторых сервисов придется приобретать платный тариф.

## **Обзор программ-аналогов**

В ходе обзора аналогов учитывались следующие функции:

1. Отправка сообщения;
2. Выбор сервиса для отправки;
3. Повторная отправка недоставленного сообщения;
4. Повышенная отказоустойчивость сервиса.

Были рассмотрены следующие сервисы-аналоги:

1. Amazon Simple Email Service;
2. SendGrid;
3. Tin-cat

Amazon Simple Email Service (SES) – это экономичный, гибкий и масштабируемый сервис электронной почты, с помощью которого разработчики могут отправлять электронные письма из любого приложения. Вы можете быстро настроить Amazon SES и выбрать несколько вариантов использования электронной почты, включая отправку транзакций, маркетинговых писем или выполнение массовой рассылки. Amazon SES включает различные возможности развертывания IP-адресов и аутентификации по электронной почте, которые позволяют повысить эффективность доставки и защитить репутацию отправителя, а также предоставляет аналитику, с помощью которой можно проанализировать эффективность каждого отправленного письма. Используйте Amazon SES для безопасной отправки электронной почты по всему миру.

Данный сервис обладает целым рядом преимуществ:

1. Быстрая интеграция;
2. Эффективная отправка сообщений;
3. Оптимизация доставки;
4. Безопасное масштабирование.

Данный сервис широко применяется для отправки мгновенных сообщений в ответ на действия пользователя, например, для подтверждения регистрации или восстановления пароля, а также хорошо подходит для массовых рассылок.

Amazon SES является достаточно недорогим способом отправки сообщений. Цены на рассылки – 10 центов за 1000 писем. При этом учитывается объем исходящего трафика. Бесплатно только первый гигабайт.

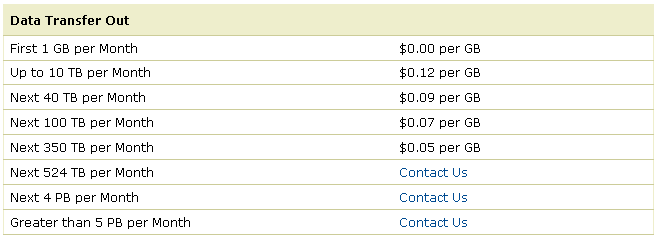


Рис. 1.2.1. Стоимость траффика Amazon SES

Данный сервис не лишен минусов:

1. Относительно низкая скорость отправки сообщений;
2. Ненадежность (немалое число отзывов о том, как Amazon SES отказывал в отправке сообщений).

Sendgrid запущен в 2017 году. Заменяет пользователям индивидуальные почтовые серверы, анализирует репутацию почтовых рассылок, не требует дополнительных затрат для масштабирования инфраструктуры. Стоимость использования сервиса международного уровня начинается с 20$/месяц.

Сервис Сендгрид предлагает каждому пользователю стать партнёром:

* Партнёрам агентства предоставляются инструменты для управления почтовыми программами клиентов с одной платфомы;
* Партнёры по рынку, после согласования с администрацией, добавляют собственные инструменты и продают их более чем 60-тысячной «горячей» аудитории;
* OEM-парнёры могут рассылать письма без подписи «via sendgrid.net»;
* Партнёры-посредники занимаются перепродажей услуг Сендгрида своим пользователям, экономя собственные ресурсы.

В бесплатной версии сервиса пользователю предоставляют 40 тыс. бесплатных электронных писем на 30 дней. По истечению месяца, можно использовать бесплатную версию сервиса с ограничением 100 писем в день.

Итоговая стоимость услуг зависит от выбранного плана и количества отправленных писем.

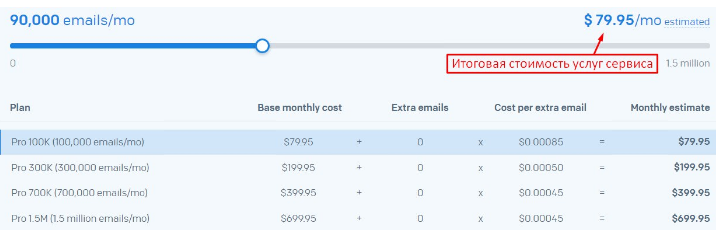


Рис. 1.2.2. Тарифные планы Sendgrid

Для увеличения производительности можно приобрести дополнительные IP-адреса за 30$/мес., эта функция доступна и будет полезной только на плане PRO.

Отправлять сообщения можно сразу после подтверждения почты и интеграции с Сендгрид через SMTP или API. У SMTP больше функций, но его сложнее настроить. API рекомендуется большинству пользователей сервиса благодаря простоте кодирования.

Для интеграции через API нужно его сгенерировать, для каждого приложения или сервиса нужен отдельный API.

Важным недостатком Sendgrid является значительная стоимость его использования. Также отмечаются проблемы с нотификацией – SendGrid просто не запрашивает ваш URL в некоторых случаях; некачественная поддержка. Отмечается частые обновления в API, что иногда приводит к проблемам с обратной совместимостью.

Tin-cat email queue – система очереди для отправки сообщений. При попытке отправить сообщение, сообщение отправляется в очередь. При этом каждую минуту система проверяет наличие сообщений в очереди и отправляет их.

Tin-cat позволяет регулировать частоту проверки очереди, а также количество сообщений, отправляемых за один период (за одну проверку).

Данная система является эффективным примером асинхронной отправки сообщений.

Недостатком системы является ее надежность. Если сообщений не будет доставлено, будет осуществлена попытка отправить его тем же способом, что не является эффективным способом.

Рассмотренные программные комплексы, разработанные сторонними компаниями, не решают все поставленные задачи. Данные ПО позволяют лишь частично контролировать отправку сообщений. Также важным моментом является высокая цена использования некоторых сервисов (таких как Sendgrid), ненадежность API. Важно: если сообщение не удается отправить, данные сервисы не предпринимают попыток отправить его снова, возможно другим способом, что сказывается на стабильности рассылки.

Из вышеперечисленного можно сделать вывод о необходимости разработки программного продукта, который должен решать все поставленные задачи.

## **Функциональное моделирование программного комплекса**

Назначение разработки: разрабатываемый программный комплекс является частью системы финансового учета. Внедрение такой подсистемы направлено на решение проблем, связанных с отправкой сообщений пользователям (сообщение не доставлено, сообщение было принято за спам), а также на упрощение работы с сообщениями в целом.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Техническое задание включает в себя требования, необходимые для реализации программного продукта.

Название программного комплекса «Подсистема управления рассылкой и учёта писем на примере сервиса управленческого учёта ПланФакт».

## **Основание для разработки**

Основанием для разработки подсистемы управления рассылкой и учёта писем на примере сервиса управленческого учёта ПланФакт является задание на дипломную работу, выданное доцентом Трубаковым А.О. на основании приказа по Брянскому государственному техническому университету № 428-3 от 27 мая 2020 г.

## **Назначение разработки**

Разрабатываемый программный комплекс должен выполнять следующие задачи:

* рассылка сообщений выбранным адресам с помощью выбранного сервиса доставки;
* обеспечение надежности доставки;
* обратная связь после доставки сообщения.

## **Требование к программному комплексу**

### ***Требования к функциональным характеристикам***

Реализуемый программный комплекс должен включать в себя нижеупомянутые функциональные особенности:

1. Авторизация пользователей.

Роли:

* Оператор – может управлять рассылкой сообщений, просматривать соответствующую историю сообщений;
* Администратор – имеет все возможности оператора, но также может создавать аккаунты для новых операторов, просматривать список операторов и управлять доступом к сервису у существующих операторов.

1. После входа в Web-сервис пользователю предоставляются следующие возможности:
   * отправлять сообщения на любые почтовые ящики;

* выбирать сервис для отправки сообщения из доступных на данный момент;
  + организовывать отправку сообщений по расписанию;
  + просматривать историю отправленных сообщений.

1. Разрабатываемый комплекс должен обладать следующими функциями:

* гарантия отправки сообщения при наличии Интернет-соединения (если не вышло отправить через один сервис, необходимо повторить попытку через другой. Повторять, пока сообщение не будет отправлено или пока не будут испробованы все доступные сервисы);
* возможность отправлять сообщения посредством минимум трех разных сервисов для отправки сообщений;
* наличие API, с помощью которого ее можно будет встраивать в различные сервисы, применяющие рассылку сообщений;
* подсистема должна давать возможность прикреплять файлы к сообщению;
* возможность хранить историю сообщений, т.е. хранить определенное число ранее отправленных сообщений и давать доступ к просмотру их состояния;
* наличие обратной связи - подсистема должна оповещать пользователя об удачное доставке или о причине неудачной отправки сообщения.

1. Разрабатываемый комплекс предоставляет асинхронную отправку сообщений.
   * Когда оператор или администратор отправляет сообщение, сообщение отправляется с очередь на удаленном сервере. Система периодически проверяет эту очередь и инициирует отправку сообщений (передает сообщения соответствующим почтовым сервисам).

* **Получение данных.** Система получает на вход тему, тело сообщения, прикрепляемый файл сообщения, список адресов, на которые нужно отправить сообщения, а также сервис, посредством которого необходимо доставить сообщение.
* **Выходные данные.** Система предоставляет пользователю историю сообщений со всей необходимой информацией, в том числе со статусами доставки.
* **Отправка данных.** Система обрабатывает полученные от пользователя данные и передает их указанному сервису для отправки.

### ***Требования к интерфейсу***

Реализуемый программный комплекс должен включать в себя нижеупомянутые функциональные особенности:

* интерфейс для работы с подсистемой (админка) – web-страница, включает в себя список со всей информацией об отправленных сообщениях, предоставляет возможность вручную отправить сообщение через выбранный сервис, настраивать почтовые шлюзы;
* необходимо обеспечить совместимость с основными браузерами (последние версии Chrome, Firefox, Safari, Opera; IE начиная с 10 версии). Верстка должна быть адаптивной и рассчитана на минимальное разрешение экрана 1280×720;
* стиль страниц должен совпадать со стилем всего сервиса. Цветовая гамма приложения не должна быть излишне яркой, но сочетания цветов должны быть контрастными. Дизайн должен быть простым и понятным, нужно избегать непонятных иконок. Списки, с помощью которых также может быть реализовано меню, не должны превышать 5-9 элементов;
* если время загрузки страницы составляет более 5с., необходимо обеспечить пользователю уверенность в том, что процесс действительно происходит при помощи индикатора выполнения процесса;
* на длинных страницах необходимо применять ссылки, возвращающие пользователя в верхнюю часть страницы.

## **Условия эксплуатации**

### ***Требования к квалификации и численности персонала***

Минимальное количество персонала, требуемого для работы программного комплекса, должно составлять не менее 1 человека: администратор, управляющий рассылкой.

### ***Климатические условия эксплуатации***

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

### ***Требования к составу и параметрам технических средств***

Подсистема функционирует в составе уже имеющегося сервиса и не предъявляет дополнительных требований к составу и параметру технических средств. Подсистема должна работать на тех же самых устройствах, что и вся система учета.

Сервер приложения:

* наличие веб-сервера IIS.
* процессор: 3,1 ГГц и выше;
* ОЗУ: 8Гб;
* SSD диск.

Сервер баз данных:

* процессор: 3,1 ГГц и выше;
* ОЗУ: 8Гб и выше;
* SSD диск.

### ***Требования к информационной и программной совместимости***

В перечень программного обеспечения, работающего на стороне сервера входит следующее ПО:

* лицензионная версия операционной системы «Microsoft Windows Server 2019»;
* базы данных «Microsoft SQL Server 2019»;
* web-браузер Google Chrome.
* антивирусное программное обеспечение.

Дополнительные требования к защите программного обеспечения и информации не предъявляются.

## **Этапы разработки**

### ***Стадии разработки***

Разработка подсистемы состоит из трех этапов:

1. разработка технического задания;
2. проектирование программного комплекса;
3. внедрение разработанного сервиса.

### ***Этапы разработки***

Стадия проектирования программного комплекса включает в себя следующие этапы:

1. разработка программного комплекса;
2. написание программной документации;
3. тестирование сервиса.

### ***Содержание работ по этапам***

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;
2. определение и уточнение требований к техническим средствам;
3. определение требований к программе;
4. определение стадий, этапов и сроков разработки программы;
5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.