

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

Вычислительная техника

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Профиль Вычислительные машины, комплексы, системы и сети

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА БАКАЛАВРА**

на тему

**Система управления PUSH-уведомлениями для сайта кафедры
образовательной организации**

Студентка _____ Денежкина Вероника Михайловна
(подпись, дата)

Руководитель _____ Пащенко Д. В.
(подпись, дата)

Нормоконтролёр _____ Кучин А.В.
(подпись, дата)

Работа допущена к защите (протокол заседания кафедры от
_____ 25.05.2018г. № 38-09-12)

Заведующий кафедрой _____ Пащенко Д.В.
(подпись)

Работа защищена с отметкой _____ (протокол заседания ГАК от _____ № _____)

Секретарь ГАК _____ Гурин Е. И.
(подпись)

Пенза, 2018

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

Вычислительная техника

«Утверждаю»
Заведующий кафедрой
Д.В. Пащенко
«__» _____ 2018г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студентка Денежкина Вероника Михайловна гр. 14ВВ1 факультета вычислительной техники.
2. Тема работы: «Система управления PUSH-уведомлениями сайта кафедры образовательной организации».

Тема утверждена приказом ПГУ № 398а/о от "03" мая 2018 г.

3. Руководитель работы д.т.н., профессор каф. ВТ ПГУ Пащенко Д. В.
4. Задание на работу (назначение разработки, исходные данные и т.п.).

Назначение разработки – технология предназначена для оповещения пользователей сайта и кафедр ПГУ о новых новостях. Разработать технологию PUSH-уведомлений, которая работает при помощи специального сервера сообщений, который передает сообщения определённым пользователям, которые дали положительный ответ на получение уведомлений.

5. Перечень подлежащих разработке вопросов.

Требуется разработать серверную и клиентскую часть программы. Серверная часть в свою очередь должна присылать пользователю

уведомления. А клиентская часть заключается в подписке на данные уведомления. Так же необходимо обеспечить взаимодействие с сервером сообщений.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	10.05.18	
2.	Выбор ПО, языка программирования и сервера	10	18.05.18	
3.	Проектирование PUSH-технологии для сайта	20	20.05.18	
4.	Разработка серверной части	10	21.05.18	
5.	Разработка клиентской части	10	24.05.18	
6.	Отладка и тестирования технологии	10	29.05.18	
7.	Оформление выпускной работы	30	06.06.16	

Дата выдачи задания "5" февраля 2018 г.

Руководитель бакалаврской работы _____ Пашенко Д. В.
(подпись, дата)

Задание к исполнению приняла студентка _____ Денежкина В. М.
(подпись, дата)

Работу к защите допустить
Декан факультета _____ Фионова Л.Р.
(подпись, дата)

РЕФЕРАТ

Бакалаврская работа содержит 65 листа, 16 рисунков, 3 приложения.

PUSH-УВЕДОМЛЕНИЯ, PUSH-УВЕДОМЛЕНИЯ ДЛЯ САЙТА УНИВЕРСИТЕТА, РАЗРАБОТКА PUSH-УВЕДОМЛЕНИЙ, СИСТЕМА УПРАВЛЕНИЯ PUSH-УВЕДОМЛЕНИЯМИ САЙТА КАФЕДРЫ ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ.

Цель работы – разработать технологию PUSH-уведомлений для оповещения всех пользователей на сайте о новостях.

Объектом разработки является программный интерфейс для сервера и клиента. Между этими двумя объектами существует сервер сообщений, который помогает доставлять сообщения с сервера всех клиентам.

В результате проведенной работы была разработана технология, удовлетворяющая поставленной задаче. А именно, разработан интерфейс пользователя, который представляет собой подписку на уведомления и запрос подписки на уведомления. А серверная часть и сервер сообщений взаимодействуют между собой, чтобы передавать сообщения пользователю. Сообщения передаются спомощью уникального идентификатора пользователя(UID), который дается каждому пользователю, кто подписался на обновления на сайте. Хранится UID на сервере.

Данную технологию можно внедрить на сайт университета с целью того, чтобы пользователи не пропускали новых новостей на сайте.

					ПГУ 1.090301.05.001 ПЗ					
Изм.	Лист	№ докум.	Подпись	Дата	Система управления PUSH-уведомлениями сайта кафедры образовательной организации. Пояснительная записка			Лит.	Лист	Листов
Разраб.		Денежкина В.М.								
Провер.		Пащенко Д. В.							4	65
Н. Контр		Кучин А.В.						ФВТ, каф. ВТ гр. 14ВВ1		
.										
Утверд.		Пащенко Д.В.								

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	10
1.1 Постановка задачи	10
1.2 Актуальность использования PUSH-технологий	11
2 ИСПОЛЬЗУЕМЫЕ ЯЗЫКИ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	20
2.1 Требования к разрабатываемой PUSH-технологии	20
2.2 Язык JavaScript.....	21
3 ПРОЕКТИРОВАНИЕ PUSH-ТЕХНОЛОГИИ ДЛЯ САЙТА КАФЕДРЫ.	23
3.1 Проектирование структуры системы.....	23
3.2 Выбор сервера для обмена сообщениями	25
3.3 Проектирование интерфейса клиентской части	26
3.4 Проектирование серверной части	27
4 ОПИСАНИЕ РАЗРАБОТКИ ПРОГРАММЫ	30
4.1 Описание разработки клиентской части приложения	30
4.2 Описание серверной части приложения.....	33
5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	35
6 ОТЛАДКА И ТЕСТИРОВАНИЕ.....	38
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	40
ПРИЛОЖЕНИЕ А ЛИСТИНГИ ПРОГРАММЫ.....	41
Приложение А.1 Файл firebase-messaging-sw.js.....	41
Приложение А.2 Файл firebase-subscribe.js	42
ПРИЛОЖЕНИЕ В UML ДИАГРАММЫ.....	48
Приложение В.1 Диаграмма взаимодействия	48
Приложение В.2 Диаграмма последовательности	49
Приложение В.3 Диаграмма вариантов использования.....	50

ВВЕДЕНИЕ

Потребность в получении известий из других местностей и стран восходит к глубокой древности и первоначально удовлетворялась посредством гонцов, которые приносили сообщения, как устные, так и письменные или облечённые в символическую форму. С развитием человеческой цивилизации происходили изменение и развитие способов, средств и форм почтовой связи.

Доисторические люди использовали для передачи важной информации голос, что способствовало появлению членораздельной речи. Однако устная передача вестей была несовершенна, так как человеческий голос слышен лишь на близком расстоянии. Для усиления передаваемого звука тысячелетия назад стали использовать полые стволы деревьев и позднее барабаны (ок. 6 тыс. лет до н. э.).

Следующим видом связи в истории человечества стали гонцы — вначале пешие, позднее конные. В древних государствах Месопотамии, Египта, Греции, Персии, Китая, Римской империи существовала хорошо налаженная государственная почтовая связь: письменные сообщения пересылались с пешими и конными гонцами по принципу эстафеты.

С зарождением письменности информация стала передаваться в письменном виде, что положило начало почтовой связи. Вначале такая связь была эпизодической. С возникновением рабовладельческих государств на Древнем Востоке, правители которых нуждались в постоянной информации о положении в собственной стране и на подвластных им территориях, почтовая связь стала приобретать упорядоченный характер.

В Греции почтовая система была достаточно хорошо налажена в виде сухопутной и морской почтовой связи, но она не могла значительно развиваться из-за множества воюющих между собой городов-государств. Правительства для передачи сообщений имели в своем распоряжении, как правило, пеших посланцев. Они назывались гемеродромами. Гонцы-бегуны преодолевали за час расстояние в 55 стадий (около 10 км) и за один рейс — 400—500 стадий.

В VII веке в мусульманском халифате возникла хорошо организованная регулярная почтовая служба, услугами которой в ограниченных размерах могли пользоваться и частные лица. Существовавшие к тому времени почтовые связи стали ещё более развиты. Почтовые курьеры имели особые знаки отличия, чтобы их можно было распознать издали.

В XVI—XVII веках во Франции, Швеции, Англии и других странах возникла централизованная королевская почта. Идея почтовой регалии, то есть исключительного права правительства содержать в пределах государственной территории почтовые учреждения, впервые выдвинута была в конце XVI века, а в XVII веке стала проводиться в жизнь.

В XIX веке коренной переворот в почтовом деле был вызван распространением железных дорог и пароходства. Появление в начале XIX века паровоза и парохода, а в начале XX века самолёта значительно увеличило скорость пересылки почтовых отправок. Почтовая связь стала общегосударственной и начала обслуживать всё население.

С изобретением телеграфа (1832), телефона (1876) и радио (1895) почтовая связь не утратила своей важной роли средства общения миллионов людей.

В настоящее время люди получают новости в интернете или из телевизора. В интернете большое развитие с каждым годом получают PUSH-уведомления, который оповещают пользователя о новых новостях на сайте.

В данной работе мы подробнее рассмотрим push-уведомления, выберем язык, на котором будем разрабатывать их и внедрим на сайт ПГУ.

					ПГУ 1.090301.05.001 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Постановка задачи

Требуется разработать систему управления PUSH-уведомлениями сайта ПГУ. Система должна предусматривать способ уведомления всех пользователей, которые посещают сайт, новыми новостями.

Обязательные требования к системе:

- Пользователи должны иметь выбор, разрешать PUSH-уведомления или нет.
- Возможность включить и отключить уведомления.
- Интерфейс системы на сайте должен быть прост для понимания пользователями.

Операционная система – MS Windows 10;

Язык программирования – JavaScript.

1.2 Актуальность использования PUSH-технологий

В первую очередь необходимо подробнее разобраться, что же такое PUSH-уведомления в целом и выявить их актуальность в современном мире.

PUSH-уведомления – это краткие всплывающие уведомления, которые появляются на экране мобильного телефона или обычного компьютера и сообщают о важных событиях и обновлениях. Любой гаджет, подключенный к интернету, способен принимать объявления такого рода. Основной целью PUSH-уведомлений является информирование пользователей об обновлениях веб-сайтов или приложений, добавлении нового контента, либо о каких-либо других новостях. Пользователь же в свою очередь либо отвергает, либо принимает данные.

Обычный пользователь может подписаться на различные темы, информацию от сервис-провайдера, и каждый раз, когда новое обновление формируется на сервере, это обновление доставляется на компьютер пользователя. Противоположностью PUSH-технологии является технология Pull, где запрос инициирует клиентское программное обеспечение.

PULL-технологии приобрели известность благодаря продукту PointCast, популярному в 1990-е годы. Сеть PointCast занималась доставкой новостей и данных фондового рынка, содержала агрегатор с собственным форматом, отдалённо напоминавшим телевидение, с текстом и рисунками, вместо видео. Влияние СМИ было значительным, так что Netscape и Microsoft в разгар браузерной войны решили включить эту технологию в свои браузеры NetscapeNavigator и InternetExplorer соответственно. Однако в большинстве случаев пользователи имели низкую скорость подключения, поэтому популярность сервиса была низкой, а позже сошла на нет, вытесненная PUSH-технологией RSS в начале 2000-х годов.

Изм.	Лист	№ докум.	Подпись	Дата

Уведомления бывают:

- Браузерные — через PUSHAPI или через дополнение
- Мобильные — через мобильное приложение
- На PC через приложение, независимо от браузера

Изначально PUSH-уведомления появились именно как способ взаимодействия мобильных приложений с пользователями мобильных устройств, однако в последнее время очень популярной стала также технология браузерных PUSH-уведомлений или WEBPUSH.

Если говорить о браузерных PUSH-уведомлениях - то это сообщения от сайтов, которые появляются в виде всплывающих окон на рабочем столе пользователя, если последний дал разрешение на их получение. При клике на такое уведомление пользователь будет перенаправлен на определенную страницу сайта, от которого пришло уведомление.

Вот как выглядит приглашение подписаться на уведомление от сайта.

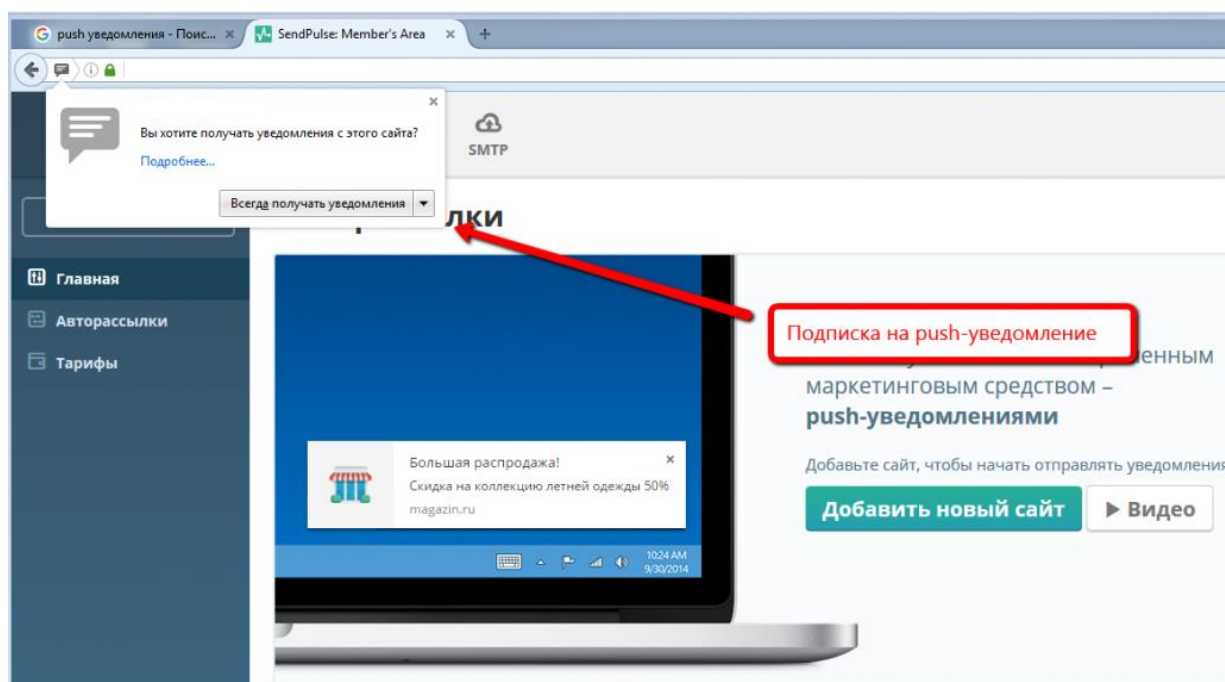


Рисунок 1. Пример подписки на PUSH-уведомления

Для того, чтобы было проще понять схему работы браузерных PUSH-уведомлений, можно взглянуть на эту схему.

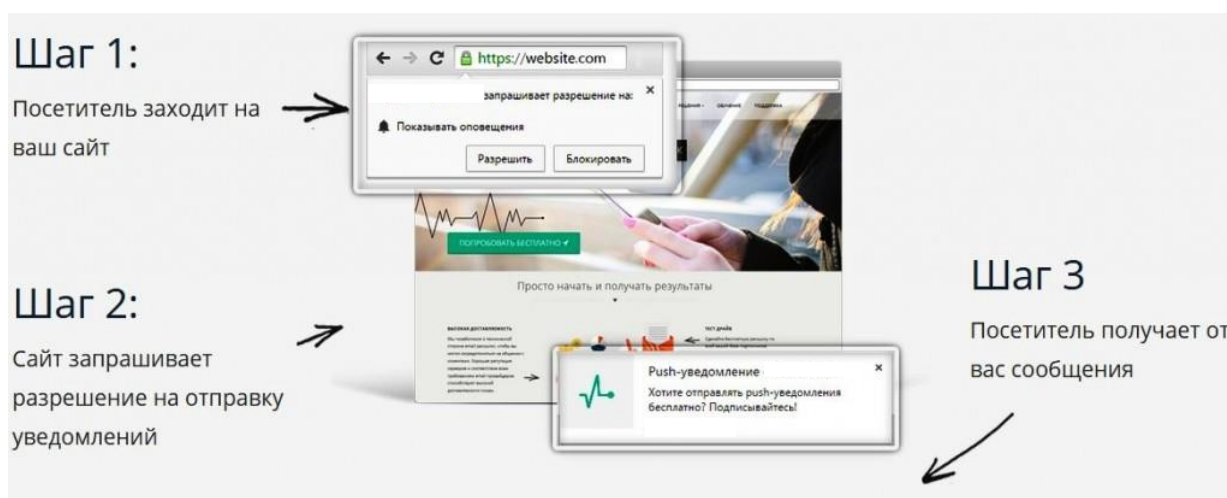


Рисунок 2. Схема работы

У PUSH-уведомлений есть свои преимущества:

1. Легко подписаться. Чтобы подписаться на PUSH-уведомления достаточно просто нажать на одну кнопку в всплывающем окне браузера, и подписка осуществляется.
2. Высокий процент доставки и переходов. По некоторым оценкам процент доставки PUSH-уведомлений составляет до 90%, а процент переходов на сайты по PUSH-уведомлениям достигает до 50%.
3. Просто подключить. Для того, чтобы подключить отправку PUSH-уведомлений на вашем сайте, достаточно установить на него всего несколько строчек кода.
4. Удобно отправлять. Вы можете настроить автоматические серии или отправку по RSS.

WEB PUSH уведомления могут быть использованы Интернет маркетологом в разных целях. Вот только некоторые из вариантов:

- Продажи
- Новости/новинки

- Напоминания
- Поздравления
- События

Для работы PUSH-уведомлений используется четыре компонента:

- Сервер PUSH-уведомлений (у каждой современной ОС есть свой сервер, у iOS это APNS, у Android — FCM (FirebaseCloudMessaging), у Windows 10/Windows 10 Mobile — WNS).
- Сервер автора приложения, который посылает уведомления серверу PUSH-уведомлений.
- Постоянно работающая служба в операционной системе устройства, общающаяся с сервером PUSH-уведомлений.
- Приложение, поддерживающее PUSH-уведомления.

Предварительно разработчик приложения регистрирует свой сервер на сервере уведомлений ОС.

Разберем плюсы и минусы PUSH-уведомлений:

Плюсы

1. Технические

- На уведомления легко подписаться новому пользователю сайта. Всё, что ему нужно, это просто нажать 1-2 кнопки.
- Просто установить и легко настроить на самом сайте. Вам нужно лишь установить код и сделать настройку в 5 шагов.

- Выгодно. Стоимость одного контакта с клиентом намного дешевле, чем SMS или E-Mail рассылка.

2. Маркетинговые

- Полная автоматизация. Настроив раз в дальнейшем все будет работать на автопилоте.
- Новая технология. Люди еще не насытились этим решением, поэтому относительно желанно подписываются и читают Ваши сообщения.
- Высокий OpenRate (процент открываемости/прочтений). Открываемость PUSH-уведомления доходит до 90% (против 30-35% в E-Mail-маркетинге)
- Сегментация. Вы можете показывать разные уведомления отличающимся целевым аудитория, таким образом, попадать в потребность клиента максимально точно.

Минусы

1. Технические

- Скрипт на сайте. Дополнительный код может незначительно в худшую сторону повлиять на скорость загрузки.
- Низкая доставляемость (около 65%). Связано это с техническими моментами: обновление операционной системы или банальная очистка кэша.
- Довольно сложная отписка. Существуют даже специальные статьи с подробными инструкциями на тему “Как отписаться от push уведомлений”. Естественно, это негативно влияет на лояльность пользователей.

- Двойное подтверждение. Для сайтов на HTTP, разрешение на подписку будет всплывать 2 раза. Причем, именно на втором разе очень большое количество отказов.
- Старые браузеры. Если у Вашего посетителя окажется старый браузер, есть большая вероятность, что он просто не увидит форму заявки. В случае новых браузеров такой проблемы точно не будет.

2. Маркетинговые

- Страх вируса. Так как инструмент не привычный, многие рассматривают это всплывающее окно как спам или попытку навредить Вашему сайту.
- Сегментация. Даже с учётом того, что в сервисах есть возможность разделения целевой аудитории, нужно отдавать отчёт, что она не является максимальной.
- Средний Click-Rate (процент переходов по уведомлению). Открываемость PUSH-уведомления высокая, но кликабельность такая же как у тех же e-mail-рассылок (2-5%).
- Интеграция. Работа этих уведомлений редко когда встраивается в общую цепочку маркетинговых инструментов, поэтому этот инструмент полностью самостоятелен и не хочет дружить с другими.
- Короткий формат. Очень малое количество информации Вы можете донести через PUSH-уведомления, только несколько строк текста, одну маленькую картинку и ссылку на сайт.

Уведомления могут содержать различные поля, такие как: кнопки ответа, изображение, числовое значение для конструктора сообщения, звук и другие. (Рисунок 4)



Рисунок 4. Пример PUSH-уведомления

Почему стоит использовать браузерные PUSH-уведомления?

1. Это новый канал. Он начал развиваться с 2015 года
2. Удобные для пользователей (сообщения высвечиваются в виде ленты, есть режим «не беспокоить»)
3. Высокий процент подписки (по результатам тестирования в 30 раз превышают E-Mail)
4. Защищенность от спама
5. Читаемость 90% (за счет доставки уведомлений прямо на рабочий стол)

Кому полезны новые возможности браузерных push-уведомлений:

1. Интернет магазинам

В первую очередь PUSH-уведомления будут полезны и эффективны в рассылке транзакционных сообщений. В отличие от смс сообщений, push уведомления бесплатны, что поможет вам значительно минимизировать затраты. Еще один плюс – PUSH приходят мгновенно и с их помощью удобно сообщать о состоянии заказа, брошенных корзинах и новых услугах.

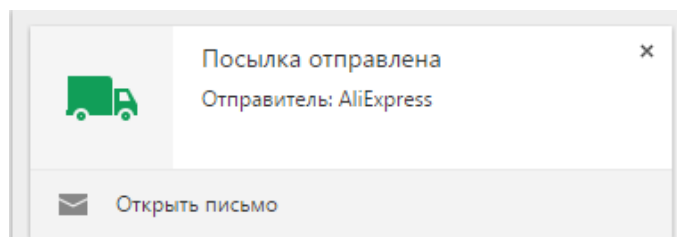


Рисунок 5. Пример PUSH-уведомления для интернет-магазина

2. Онлайн презентациям и вебинарам

PUSH-сообщения – это лучший способ сообщить аудитории о предстоящем событии. Вы можете начать отправлять уведомления за месяц до начала презентации либо вебинара поэтапно рассказывая подписчикам что нового и полезного они извлекут для себя из программы.

3. В промо кампаниях

Главные цели каждой промо кампании – привлечь потенциальных клиентов и разослать информацию о продуктах, услугах и скидках.

Почему стоит воспользоваться PUSH-уведомлениями?

- Быстрее преобразуют посетителей сайта в подписчиков (по результатам исследований минимальный показатель подписок составляет 2%, средний- 5%, а высокий – 28%)
- Коэффициент переходов по PUSH-сообщениям – 10-20%
- При нажатии на уведомление пользователь сразу перенаправляется на страницу сайта
- Уменьшает шаги пользователя на встречу к покупке.

4. Биржевым приложениям, службам прогноза погоды.

У этих сервисов есть необходимость рассылать важные напоминания, статистику, цифры несколько раз в день. С помощью PUSH-уведомлений это делать удобнее и эффективнее, потому что сообщения приходят пользователям мгновенно.

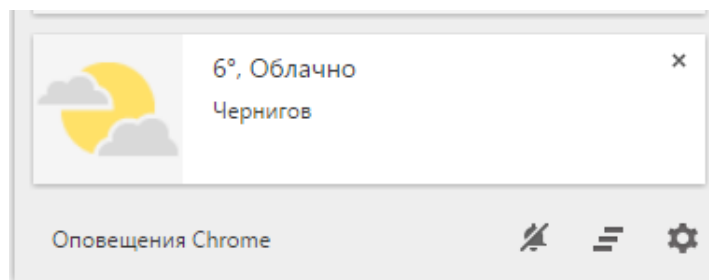


Рисунок 6. Пример PUSH-уведомления для прогноза погоды

5. Новостным порталам

Они также используют PUSH чтобы распространять актуальную и свежую информацию. Среди таких пользователей: The New York Times, Apple Insider, CNN, The Next Web.

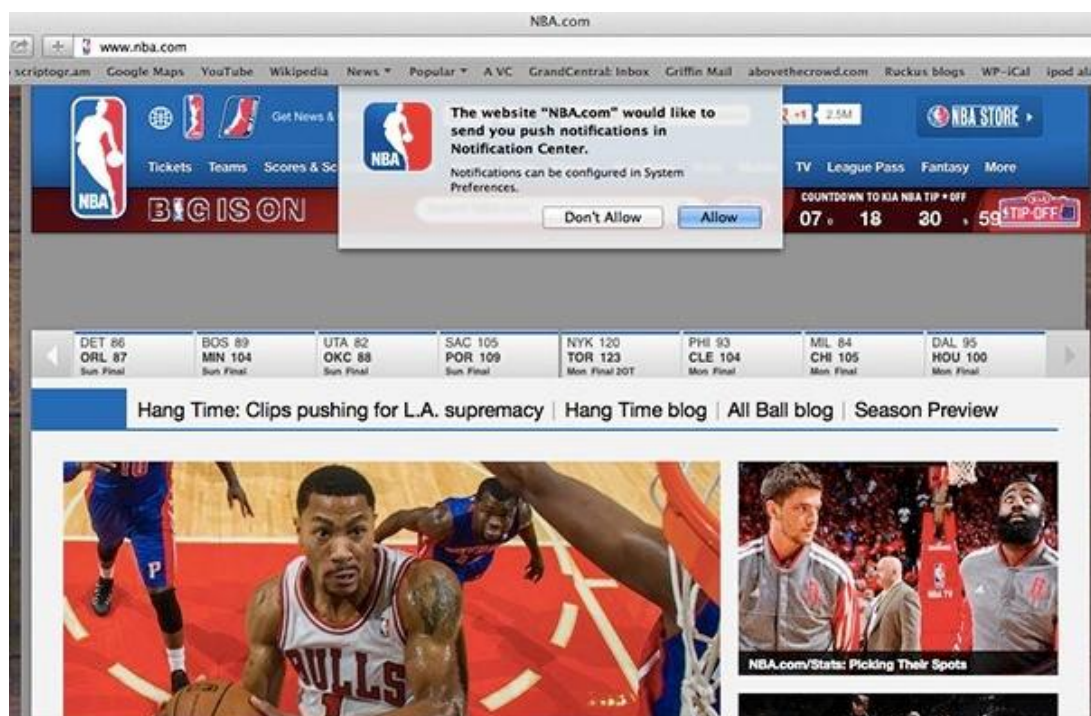


Рисунок 7. Пример PUSH-уведомления для сайта новостей

Исходя из выше сказанного, можем сделать вывод, что PUSH-уведомления являются очень потенциальной развивающейся технологией в интернете и PUSH-уведомления могут отлично дополнить функциональность вашего сайта.

2 ИСПОЛЬЗУЕМЫЕ ЯЗЫКИ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

2.1 Требования к разрабатываемой PUSH-технологии

- Простота подписки. Для выражения согласия на получение уведомлений пользователю достаточно поставить одну галочку.
- Хорошая просматриваемость уведомлений. Объявление, которое появляется поверх работающих приложений, прочитывается практически в 100% случаев. Количество последующих переходов по имеющимся в нем ссылкам также гораздо выше, чем в случае с рекламными баннерами.
- Легко установить сервисы на свой сайт. Процедура установки модуля PUSH-рассылок элементарна и требует только введения кода.
- Отсутствие спама. Пользователь, который будет получать слишком много уведомлений нецелевого характера, может без труда аннулировать подписку и избавить себя от потока ненужной информации. В отличие от PUSH-технологии, на электронную почту или номер сотовой связи могут направлять сообщения многочисленные службы по рассылкам спама, для которых не составляет проблем добыть подобные данные.
- Релевантность информации. Получателями распространяемой информации являются только те пользователи, которые подтвердили свою заинтересованность в ней. Все эти факторы способствуют тому, что PUSH-уведомления стремительно распространяются как уникальный рекламный инструмент. Их оперативность и актуальность дает возможность пользователям быть в курсе всех обновлений.

2.2 Язык JavaScript

JavaScript изначально создавался для того, чтобы сделать WEB-странички «живыми». Программы на этом языке называются скриптами. В браузере они подключаются напрямую к HTML и, как только загружается страничка – тут же выполняются.

Программы на JavaScript— обычный текст. Они не требуют какой-то специальной подготовки.

JavaScript—это язык программирования, который служит для управления сценариями просмотра WEB-страниц. Главная особенность данного языка заключается в том, что при его использовании есть возможность изменять свойства среды отображения при просмотре WEB-сайта и при этом не будет происходить перезагрузка WEB-страниц. Также есть возможность создать новое окно отображения либо вывести различные уведомления. JavaScript представляет собой объектно-ориентированный язык программирования, но так как в нем используется прототипирование (быстрая реализация базовой функциональности для анализа работы системы), поэтому есть ряд особенностей, связанных с непосредственным исполнением кода языка по сравнению с традиционными объектно – ориентированными языками программирования. Помимо этого, язык JavaScript обладает перечнем свойств, которые присущи другим функциональным языкам, а именно: функции как объекты первого класса, объекты как списки, анонимные функции.

Хотя язык JavaScript по синтаксису похож на язык Си, он все же имеет ряд отличий:

1. используются объекты со способностью самоанализа;
2. функции как объекты первого класса;

3. автоматическое приведение типов;
4. анонимные функции.

В качестве недостатков языка JavaScript можно выделить следующие:

1. нет возможности регулировать области;
2. отсутствие интерфейса как такового;
3. использование стандартных интерфейсов доступа к Web-серверам и базам данных.

					ПГУ 1.090301.05.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

3 ПРОЕКТИРОВАНИЕ PUSH-ТЕХНОЛОГИИ ДЛЯ САЙТА КАФЕДРЫ

3.1 Проектирование структуры системы

Как мы выше рассмотрели, мы с уверенностью можем сказать, что PUSH-уведомления являются очень нужной технологией на сайте. Сайт ПГУ не обладает такой технологией. Для ее разработки необходимо определить, как это будет реализовано на стороне клиента и сервера. Определить модули. Разработать интерфейс, чтобы он был максимально понятен для всех пользователей, которые посещают сайт. Возможность подписаться на обновления, а также запретить присылать уведомления. PUSH-уведомления будут оповещать пользователя о новых новостях, которые появляются на сайте. Это облегчает работу, а также не заставит пользователя заходить по несколько раз на сайт, чтобы узнать, появились новые новости или нет. Данная технология будет оповещать пользователя сразу, вне зависимости от того, находитесь вы сейчас на сайте или нет.

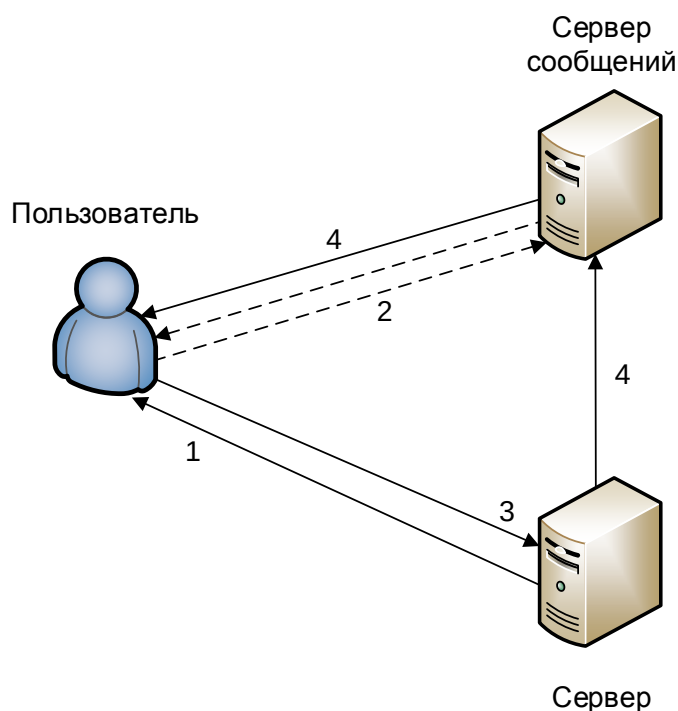


Рисунок 8. Схема взаимодействия

Изм.	Лист	№ докум.	Подпись	Дата

Для начала рассмотрим, как эта вся технология устроена и работает. Схема взаимодействия у нас показана на рисунке 8. У нас имеются 3 составляющие данного процесса:

- Сервер
- Сервер сообщений
- Пользователь

Первым делом пользователь входит на сайт (1). После того пользователю будет предложен выбор подписаться на обновления или нет. Мы рассмотрим случай положительного ответа. В случае положительного ответа пользователь регистрируется и получает уникальный UID (2). Далее клиент отправляет полученный UID на сервер и сервер привязывает пользователя к конкретному устройству (3). Когда у нас появляется новое сообщение, которое нужно доставить пользователю, сервер отправляет данное сообщение клиенту через сервер сообщений, при этом используя полученный уникальный UID (4).

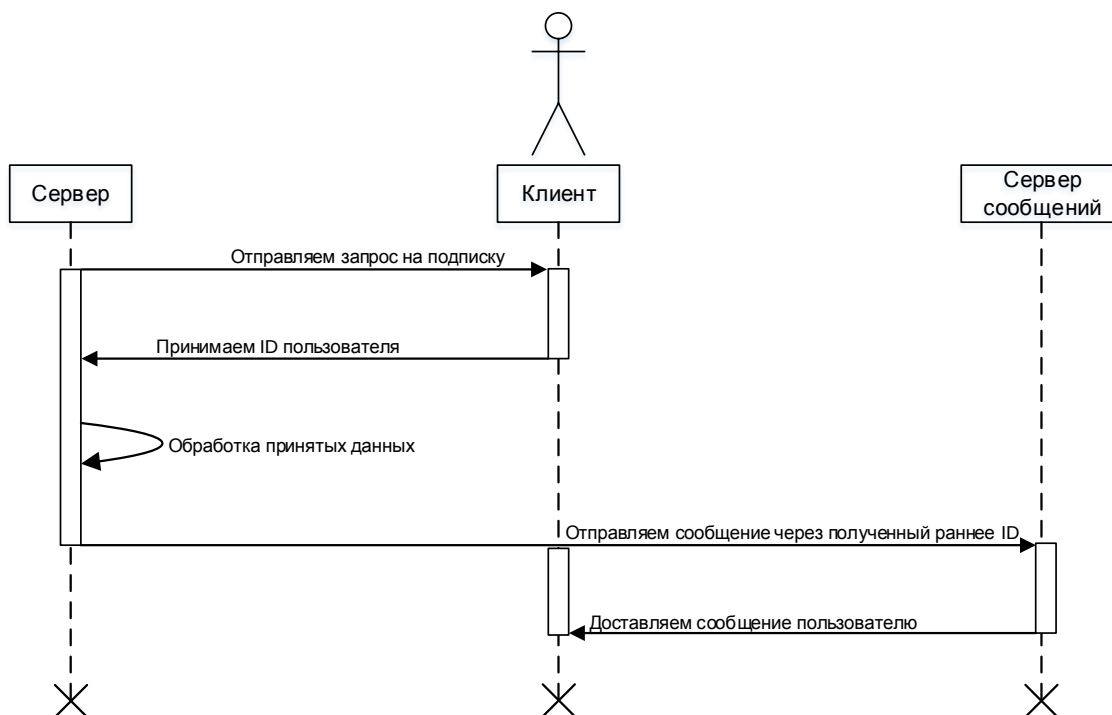


Рисунок 9. Диаграмма последовательности

На рисунке 9 представлена диаграмма последовательности взаимодействия между собой клиента, сервера и сервера сообщений.

3.2 Выбор сервера для обмена сообщениями

В качестве сервера для обмена сообщениями мы будем использовать FirebaseCloudMessaging(FCM). FCM – служба, которая обрабатывает отправку, маршрутизацию и организацию очередей сообщений между серверными приложениями и мобильными клиентскими приложениями. Позволяет надежно доставлять сообщения без каких-либо затрат. FCM является преемником GoogleCloudMessaging (GCM), и она построена на GooglePlayServices.

Работает служба по следующему принципу:

Когда сообщение, которое необходимо доставить до пользователя отправляется с сервера на сервер соединений FCM. Данный сервер FCM, в свою очередь, посылает сообщение на устройство, когда будет запущен браузер. Так как сайт не всегда может быть открыт на стороне пользователя, сервер FCM сохраняет сообщения, чтобы в дальнейшем доставить их пользователю, когда он зайдет на сайт.

FCM использует следующие параметры для идентификации сервера и сайта пользователя. И также использует данные параметры для передачи сообщений:

- Идентификатор отправителя - уникальное числовое значение.
- Ключ API – удостоверение сайта для получения сообщений от FCM.
- Токен регистрации – разрешает данному устройству получение сообщений от FCM.

3.3 Проектирование интерфейса клиентской части

В общем смысле интерфейс — совокупность средств, методов и правил взаимодействия (управления, контроля и т.д.) между элементами системы. При разработке интерфейса клиентской части необходимо обратить внимание на тот факт, что основными элементами управления являются подписка на новости или отказ от них.

При разработке необходимо обратить внимание на некоторые моменты. Во-первых, необходимо разработать такой JavaScript, который будет у каждого пользователя, который зашел на сайт спрашивать разрешение, чтобы показывать уведомление. Пользователь должен быть в праве, как принять подписку, а также и отказаться от нее. Это является очень важным моментом. Объясняется это тем, что очень неправильно будет подписывать на новые новости каждого пользователя, зашедшего на сайт, без возможности дальнейшего отказа от уведомлений.

Во-вторых, если пользователь подписался на уведомления необходимо, чтобы его уникальный идентификатор (UID) находился на сервере и на сервере сообщений. На сервере тем самым мы идентифицируем пользователя, которому необходимо отправлять сообщения, и привязываем данный UID к уникальному идентификатору устройства (ID). В случае появления нового сообщения на сервере он передает его на сервер сообщений с помощью уникального идентификатора пользователя UID, который уже находится на сервере сообщений. А сервер сообщений в свою очередь уже отправляет конкретному пользователю или группе пользователей.

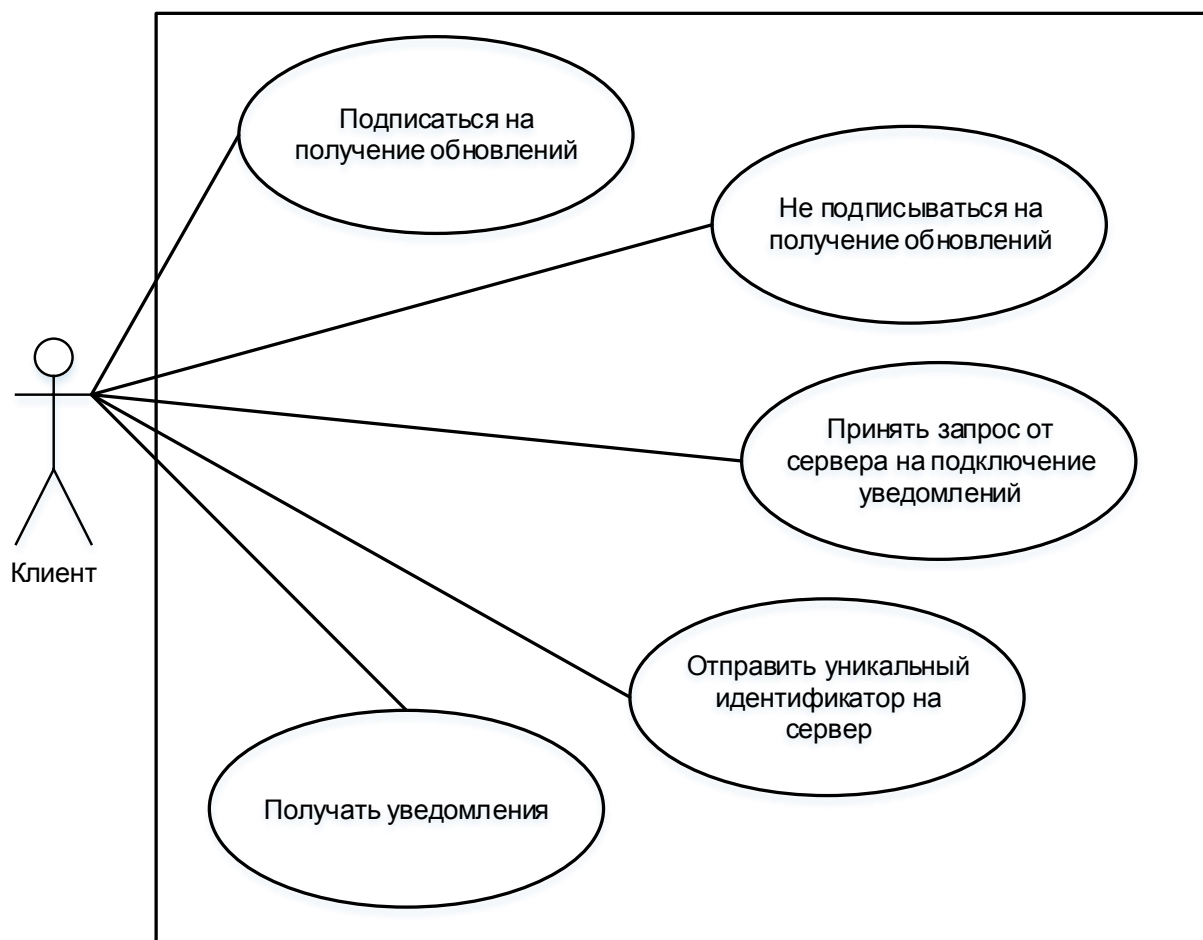


Рисунок 10. Диаграмма вариантов использования для клиента

На диаграмме показаны основные взаимодействия клиента с PUSH-уведомлениями.

3.4 Проектирование серверной части

Если посмотреть на особенности при проектировании интерфейса клиентской части, то может показаться, что это все очень сложно. Но на серверной части все еще сложнее. Рассмотрим подробнее.

Во-первых, самым важным является запоминание уникальных идентификаторов устройства(ID). Потому что мы должны знать, кому необходимо отправить уведомления. Каждый идентификатор у нас будет храниться в базе данных.

Каждый идентификатор устройства(ID) необходимо привязывать к уникальному идентификатору пользователя(UID). Потому что стоит помнить, что с различного устройства у нас могут находиться в сети различные пользователи.

Так как сайт ПГУ является многопользовательским сайтом, следовательно, много пользователей, которые желают получать рассылку. Из-за этого будут возникать очереди и необходимо будет создавать дополнительные модули, которые будут ускорять процесс отправки путем распараллеливания процессов.

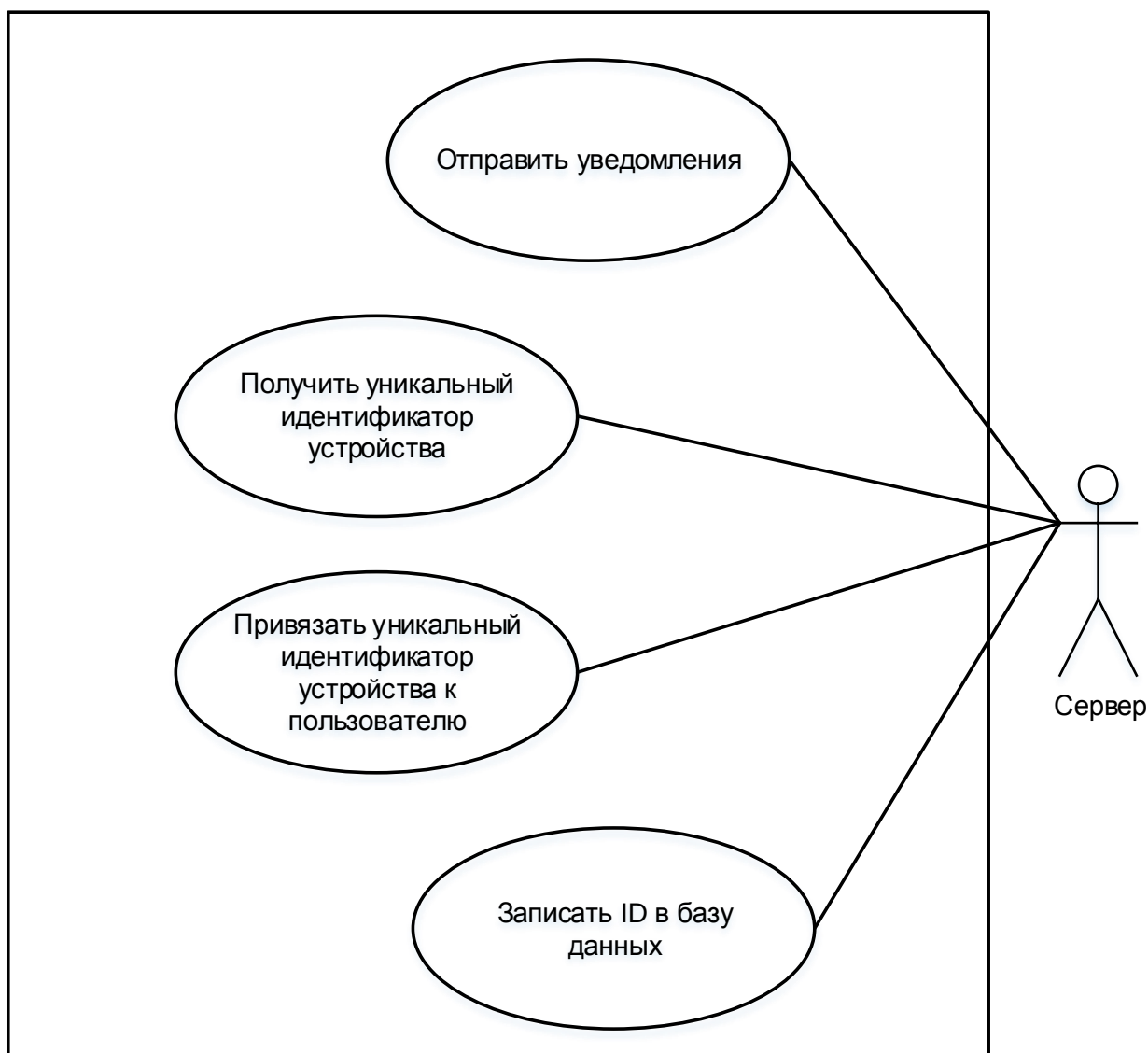


Рисунок 11. Диаграмма вариантов использования для сервера

На диаграмме показаны основные функции, которые выполняются сервером.

Сервер имеет такие функции:

- Записывает UID пользователя в базу данных
- Передает данный UID на сервер сообщений, чтобы сервер сообщений знал, кому передавать сообщение в дальнейшем
- Получает уникальный UID
- Привязывает UID к ID устройства
- Отправляет нужное уведомление на сервер сообщений

					ПГУ 1.090301.05.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

4 ОПИСАНИЕ РАЗРАБОТКИ ПРОГРАММЫ

4.1 Описание разработки клиентской части приложения

Как мы ранее говорили, что будем использовать сервер FirebaseCloudMessaging для передачи сообщений. Для этого первым делом нам будет необходимо зарегистрироваться и создать проект, который у нас будет отвечать за уведомления всех пользователей, находящихся на сайте нашего университета.

Этапы создания:

1. Необходимо зайти на официальный сайт FCM.
2. Зарегистрироваться.
3. После регистрации на главном экране видим «Createnewproject» и нажимаем.
4. При создании проекта необходимо будет указать название проекта и страну.
5. После создания проекта необходимо перейти в «Настройки проекта».
6. Находим вкладку «CloudMessaging» и переходим на нее.
7. Здесь нам необходимо будет сохранить SERVERKEYи SENDERID, который будет использоваться для получения сообщений на стороне клиентов.

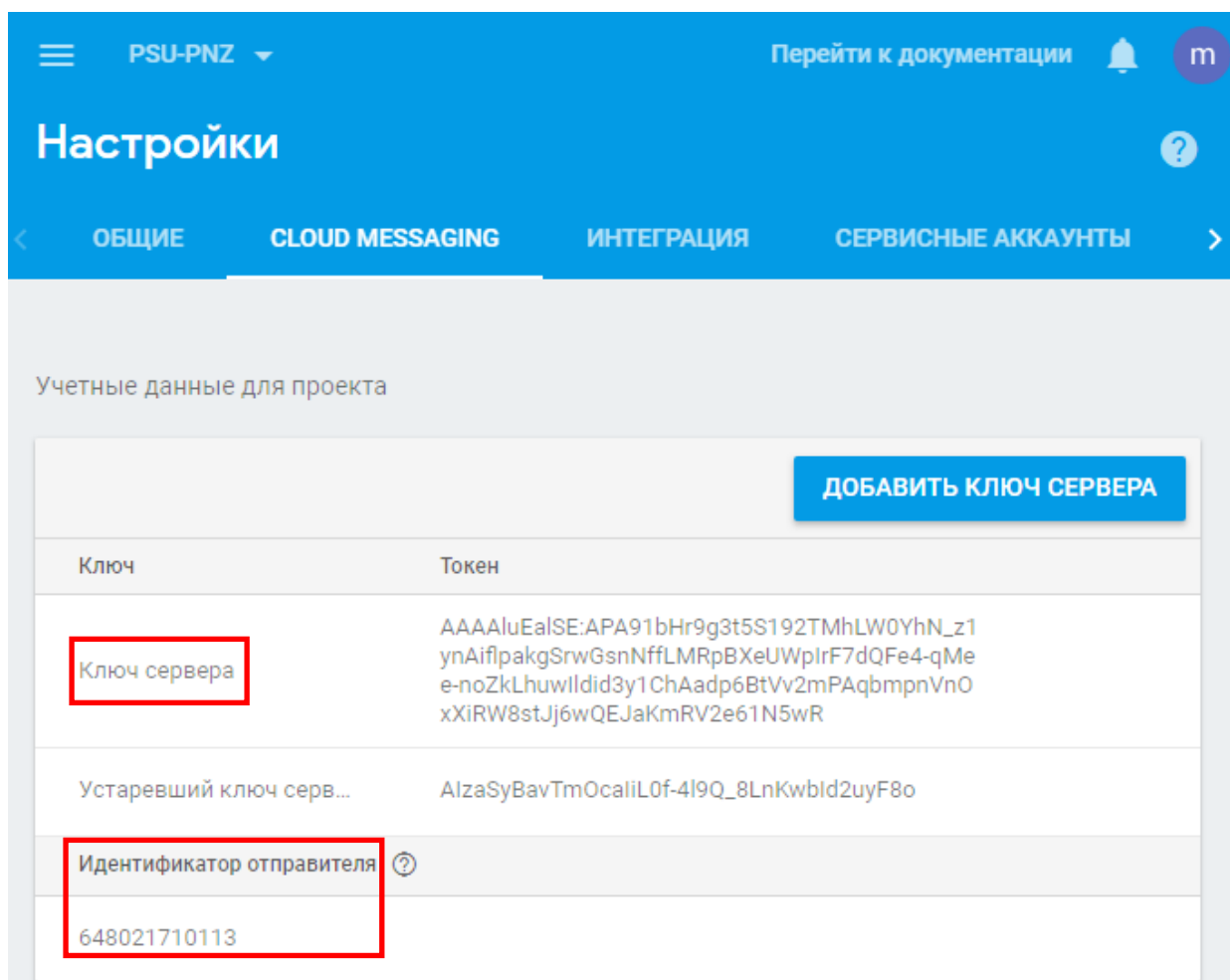


Рисунок 12. Сайт сервера сообщений FCM

После всех выполненных этапов мы приступаем к написанию кода, чтобы уведомления показывались на сайте. Создаем файл `firebase-messaging-sw.js` (Приложение А), отвечающий за показ уведомлений. Файл должен называться именно таким образом, потому что мы используем специальный сервер для передачи сообщений и путь к данному файлу прописан в библиотеке Firebase. Данный проект должен находиться в корневой папке сайта.

Теперь перейдем к самому главному. Так как мы сказали, что необходимо опрашивать всех пользователей, хотят ли они получать новые новости, а не сразу подписывать всех, то необходимо будет создать такой скрипт. Файл будет называться `firebase_subscribe.js` (Приложение А). Данный

файл и будет являться скриптом подписки. Для того, чтобы объявить вызов данного скрипта необходимо в коде самого сайта между тегами **HEAD** вставить данный код:

```
<script type="text/javascript" src="//www.gstatic.com/firebasejs/3.6.8/firebase.js"></script><script type="text/javascript" src="/firebase_subscribe.js"></script>
```

Добавим на страницу сайта кнопку для подписки на уведомления, для этого необходимо между заголовками **BODY** вставить следующий текст:

```
<button type="button" id="subscribe">Подписаться на уведомления</button>
```

Чтобы кнопка удовлетворяла дизайну сайта необходимо её изменить с помощью каскадных таблиц стиля (CSS). Кнопка нужна, чтобы пользователь, в случае отказа от показа новых уведомлений захотел их получать, то она нажимает на кнопку и перед ним появляется уведомления, которое показано на рисунке 13.

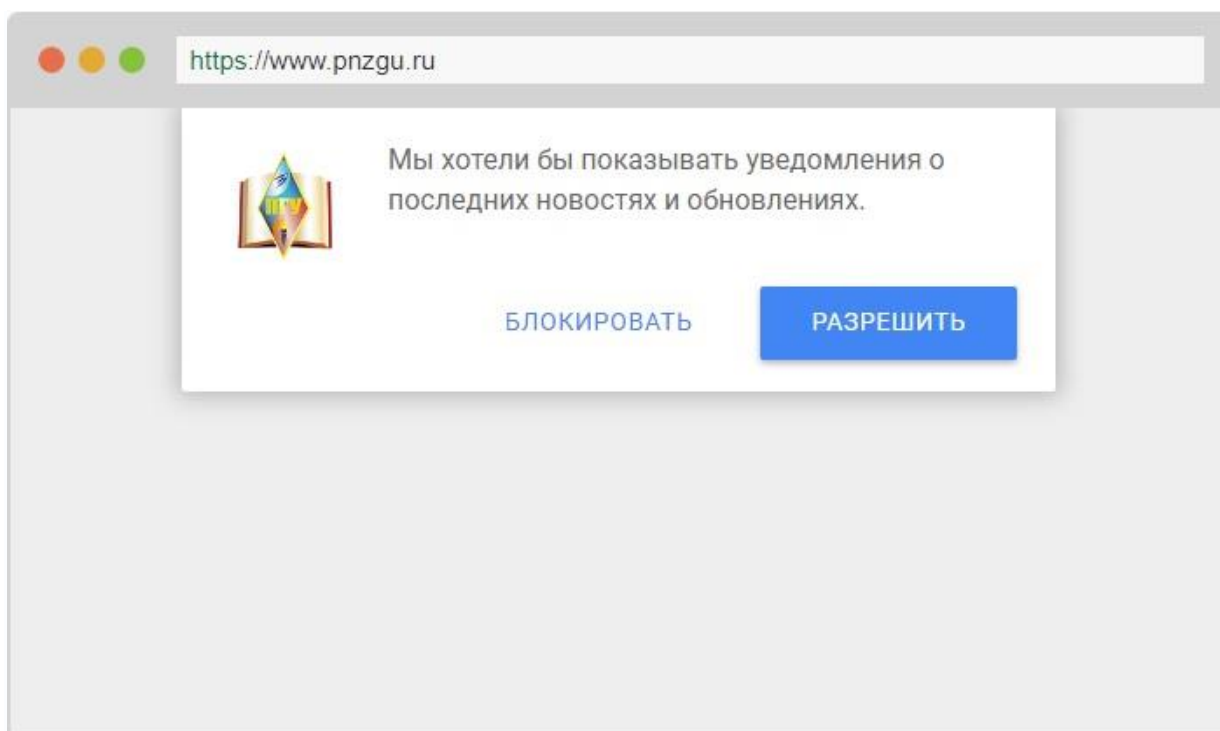


Рисунок 13. Пример оповещения для подписки на уведомления

Также это уведомление увидит каждый новый пользователь, который зашел на сайт.

4.2 Описание серверной части приложения

Серверная часть в нашем случае предназначена для отправки PUSH-уведомлений. Разберем на небольшом примере, как у нас выглядит уведомления:

```
POST/fcm/sendHTTP/1.1
Host: fcm.googleapis.com
Authorization: key=YOUR-SERVER-KEY
Content-Type: application/json
{
  "notification": {
    "title": "ПГУ",
    "body": "Спасибо за подписку!",
    "icon": "https://www.pnzgu.ru/images/logo_pnzgu.png",
    "click_action": "https://www.pnzgu.ru/"
  },
  "to": "YOUR-TOKEN-ID"
}
```

В данном примере представлен пример обычного уведомления, без всего лишнего. Уведомление представлено на рисунке 14. Тело самого уведомления, то есть что у нас будет содержать наш конструктор, будет находиться внутри объекта «notification». Внутри этого тега мы можем добавить много различных свойств объекта. Эти свойства можно найти в специальной документации для PUSH-уведомлений. Разберём по порядку, что означает каждое свойство в нашем примере:

1. Большими буквами у нас выделены поля YOUR_SERVER_KEY и YOUR_TOKEN_ID. Первый параметр означает ServerKey, который мы получили при регистрации на FCM. А второй параметр —

это уникальный идентификатор устройства конкретного пользователя, которому предаём сообщение.

2. Поле «notification» — это поле отвечает за параметры, которые будет содержать уведомление.
3. Поле «title» — поле отвечает за заголовок уведомления. Указывается как параметр конструктора.
4. Поле «body» — поле содержит основное тело (текст) уведомления. Указывается как параметр конструктора.
5. Поле «icon» — поле, содержащее URL изображения, которое должно быть изображено в уведомлении. Указывается как параметр конструктора.
6. Поле «click_action» — поле, содержащее URL адрес страницы, на которую перейдет пользователь после нажатия.
7. Поле «to» — поле, которое содержит ID устройств получателей уведомления.

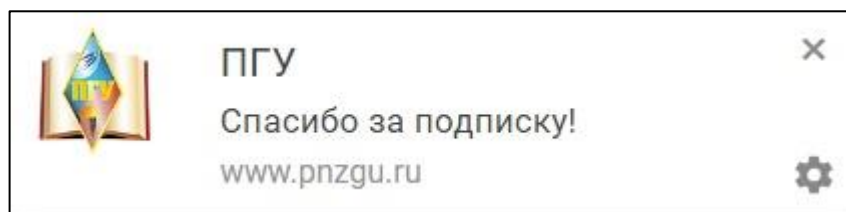


Рисунок 14. Пример PUSH-уведомления

На рисунке 14 представлен простой пример PUSH-уведомления. Данные уведомления могут иметь различную структуру: с несколькими вариантами, с картинками, анимированные и т.д.

5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В этом пункте мы более подробно разберем, как же у нас функционирует наша система PUSH-технологий и сделаем небольшое описание.

Когда пользователь зайдет на сайт ПГУ, то через несколько секунд он увидит следующее сообщение:

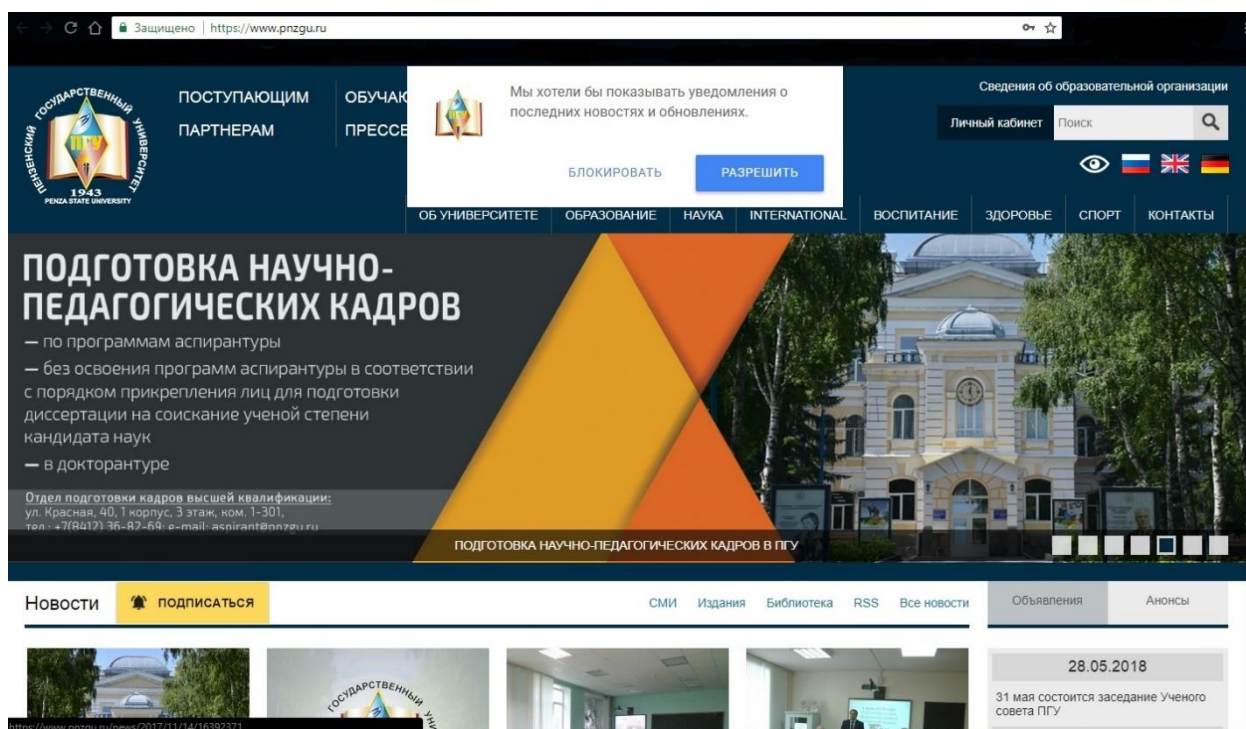


Рисунок 15. Запрос подписки на уведомления

Каждому пользователю, который зайдет на сайт, будет предложено подписаться на обновления сайта ПГУ. У пользователя есть два выбора: «Подписаться» или «Блокировать». Сначала рассмотрим вариант, когда пользователь нажмет на кнопку «Разрешить» получать уведомления от сайта. Сразу после подтверждения пользователь в правом нижнем углу увидит первое уведомление (рисунок 16).

Изм.	Лист	№ докум.	Подпись	Дата

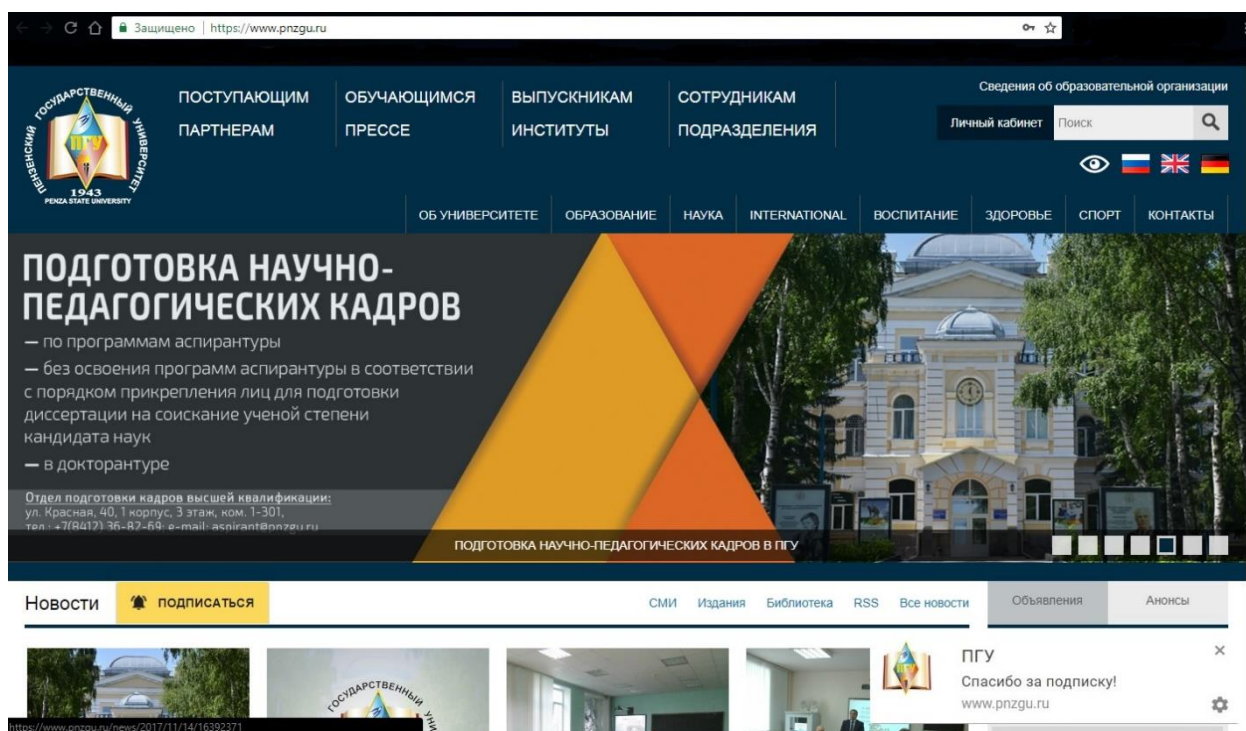


Рисунок 16. Получение первого уведомления

Если мы нажмем на данное уведомление, то попадем снова на сайт ПГУ. После этого пользователь будет получать уведомления о всех новых новостях на сайте.

Теперь рассмотрим пример, когда пользователь сначала нажал «Блокировать» уведомления, но потом передумал и решил подписаться. Чтобы не искать в настройках браузера, как же включить уведомления, на сайте будет находиться кнопка яркая и заметная кнопка. Кнопка «Подписаться» находится рядом с заголовком «Новости», на рисунке 15 и 16 можем ее увидеть. Если мы ее нажмем, то снова у пользователя появляется всплывающее окно сверху, как на рисунке 15.

В качестве сервера для обмена сообщениями мы использовали Firebase Cloud Messaging (FCM). FCM – служба, которая обрабатывает отправку, маршрутизацию и организацию очередей сообщений между серверными приложениями и мобильными клиентскими приложениями. Позволяет надежно доставлять сообщения без каких-либо

затрат. FCM является преемником GoogleCloudMessaging (GCM), и она построена на GooglePlayServices.

Работает служба по следующему принципу:

Когда сообщение, которое необходимо доставить до пользователя отправляется с сервера на сервер соединений FCM. Данный сервер FCM, в свою очередь, посылает сообщение на устройство, когда будет запущен браузер. Так как сайт не всегда может быть открыт на стороне пользователя, сервер FCM сохраняет сообщения, чтобы в дальнейшем доставить их пользователю, когда он зайдет на сайт.

Сервер сообщений выполняет функции:

- Отправляет сообщение пользователю
- Принимает UID пользователя, которому необходимо отправить
- Регистрирует пользователя на сервере, присваивая ему UID

6 ОТЛАДКА И ТЕСТИРОВАНИЕ

Отладка и тестирование написанного кода для скрипта проводилась на стороннем сайте, чтобы не прекращать работу основного сайта и не испортить его функционирование. В процессе отладки были выявлены недочеты в работе уведомлений. Но в процессе разработки все было устранено и технология уведомлений готова к работе.

					ПГУ 1.090301.05.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы была разработана технология PUSH-уведомлений, которая уведомляет о новых новостях на сайте учебного заведения. Разработанная технология может быть внедрена на сайт Пензенского Государственного Университета. Также технология предлагает каждому пользователю выбор, подписываться или не подписываться на новости.

В дальнейшем технологию можно развить и внедрить PUSH-уведомления о новостях сайта на мобильные устройства.

					ПГУ 1.090301.05.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		39

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. PUSH-уведомления: что это такое и как использовать для сайта.

<https://in-scale.ru/blog/push-uvedomleniya-cto-eto.html>

2. Технология PUSH

https://ru.wikipedia.org/wiki/Технология_push

3. Язык JavaScript – Современный учебник JavaScript

<https://learn.javascript.ru/js>

4. Firebase Cloud Messaging

<https://firebase.google.com/docs/cloud-messaging/>

					ПГУ 1.090301.05.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

ПРИЛОЖЕНИЕ А ЛИСТИНГИ ПРОГРАММЫ

Приложение А.1 Файл firebase-messaging-sw.js

```
importScripts('https://www.gstatic.com/firebasejs/3.7.2/firebase
.js');
importScripts('firebase-subscribe.js');

firebase.initializeApp({
messagingSenderId: '..' //SenderIDнаFCM
});

const messaging = firebase.messaging();

messaging.setBackgroundMessageHandler(function(payload) {
console.log('Обработка фонового сообщения', payload);

// Copy data object to get parameters in the click handler
payload.data.data = JSON.parse(JSON.stringify(payload.data));

    return self.registration.showNotification(payload.data.title,
payload.data);
});

self.addEventListener('notificationclick', function(event) {
const target = event.notification.data.click_action || '/';
event.notification.close();

event.waitUntil(clients.matchAll({
    type: 'window',
includeUncontrolled: true
}).then(function(clientList) {
    // clientList always is empty?!
    for (vari = 0; i<clientList.length; i++) {
var client = clientList[i];
        if (client.url === target && 'focus' in client) {
            return client.focus();
        }
    }

    return clients.openWindow(target);
})));
});
```

Приложение А.2 Файл firebase-subscribe.js

```
firebase.initializeApp({
  messagingSenderId: '..' //SenderIDнаFCM
});

var bt_register = $('#register');
var bt_delete = $('#delete');
var token = $('#token');
var form = $('#notification');
var message_id = $('#message_id');
var message_row = $('#message_row');

var info = $('#info');
var info_message = $('#info-message');

var alert = $('#alert');
var alert_message = $('#alert-message');

var input_body = $('#body');
var timerId = setInterval(setNotificationDemoBody, 10000);

//Тело уведомления для отправки
function setNotificationBody() {
  ...
}

setNotificationBody();
resetUI();

if (window.location.protocol === 'https:' &&
    'Notification' in window &&
    'serviceWorker' in navigator &&
    'localStorage' in window &&
    'fetch' in window &&
    'postMessage' in window
) {
  var messaging = firebase.messaging();

  if (Notification.permission === 'granted') {
    getToken();
  }

  // Проверяем подписку
  bt_register.on('click', function() {
    getToken();
  });
}
```

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.05.001 ПЗ

Лист

42

```

bt_delete.on('click', function() {
    // Delete Instance ID token.
    messaging.getToken()
    .then(function(currentToken) {
        messaging.deleteToken(currentToken)
        .then(function() {
            console.log('Token deleted');
            setTokenSentToServer(false);
            // Once token is deleted update UI.
            resetUI();
        })
        .catch(function(error) {
            showError('Unable to delete token', error);
        });
    })
    .catch(function(error) {
        showError('Error retrieving Instance ID token', error);
    });
});

form.on('submit', function(event) {
    event.preventDefault();

    var notification = {};
    form.find('input').each(function () {
        var input = $(this);
        notification[input.attr('name')] = input.val();
    });

    sendNotification(notification);
});

// handle catch the notification on current page
messaging.onMessage(function(payload) {
    console.log('Message received', payload);
    info.show();
    info_message
        .text('Новые новости на сайте! Заходи!');
};

//показна мобильных устройствах
navigator.serviceWorker.register('/serviceworker/firebase-
messaging-sw.js');
Notification.requestPermission(function(permission) {
    if (permission === 'granted') {
        navigator.serviceWorker.ready.then(function(registration) {
            // Copy data object to get parameters in the
            click handler
            payload.data.data = JSON.parse(JSON.stringify(payload.data));

```

```

registration.showNotification(payload.data.title, payload.data);
    }).catch(function(error) {
        // registration failed :(
showError('ServiceWorker registration failed', error);
    });
    }
    });
});

messaging.onTokenRefresh(function() {
messaging.getToken()
.then(function(refreshedToken) {
console.log('Token refreshed');
        // Send Instance ID token to app server.
sendTokenToServer(refreshedToken);
updateUIForPushEnabled(refreshedToken);
    })
    .catch(function(error) {
showError('Unable to retrieve refreshed token', error);
    });
});

} else {
    if (window.location.protocol !== 'https:') {
showError('Is not from HTTPS');
    } else if (!('Notification' in window)) {
showError('Notification not supported');
    } else if (!('serviceWorker' in navigator)) {
showError('ServiceWorker not supported');
    } else if (!('localStorage' in window)) {
showError('LocalStorage not supported');
    } else if (!('fetch' in window)) {
showError('fetch not supported');
    } else if (!('postMessage' in window)) {
showError('postMessage not supported');
    }
}

console.warn('This browser does not support desktop
notification.');
```

```

console.log('Is HTTPS', window.location.protocol === 'https:');
console.log('Support Notification', 'Notification' in window);
console.log('Support ServiceWorker', 'serviceWorker' in
navigator);
console.log('Support LocalStorage', 'localStorage' in window);
console.log('Support fetch', 'fetch' in window);
console.log('Support postMessage', 'postMessage' in window);

updateUIForPushPermissionRequired();
}

```

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.05.001 ПЗ

Лист

44

```

function getToken() {
messaging.requestPermission()
.then(function() {
messaging.getToken()
.then(function(currentToken) {

                if (currentToken) {
sendTokenToServer(currentToken);
updateUIForPushEnabled(currentToken);
                } else {
showError('No Instance ID token available. Request permission to
generate one');
updateUIForPushPermissionRequired();
setTokenSentToServer(false);
                }
            })
.catch(function(error) {
showError('An error occurred while retrieving token', error);
updateUIForPushPermissionRequired();
setTokenSentToServer(false);
            });
        })
.catch(function(error) {
showError('Unable to get permission to notify', error);
        });
}

function sendNotification(notification) {
var key = '..'; //КлючнаFCM

console.log('Send notification', notification);

info.hide();
message_row.hide();

messaging.getToken()
.then(function(currentToken) {
    fetch('https://fcm.googleapis.com/fcm/send', {
        method: 'POST',
        headers: {
            'Authorization': 'key=' + key,
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({
            // Firebase loses 'image' from the
notification.

```

```

        data: notification,
        to: currentToken
    })
    }).then(function(response) {
        return response.json();
    }).then(function(json) {
console.log('Response', json);

        if (json.success === 1) {
message_row.show();
message_id.text(json.results[0].message_id);
        } else {
message_row.hide();
message_id.text(json.results[0].error);
        }
    }).catch(function(error) {
showError(error);
    });
    })
    .catch(function(error) {
showError('Error retrieving Instance ID token', error);
    });
}

function sendTokenToServer(currentToken) {
    if (!isTokenSentToServer(currentToken)) {
console.log('Sending token to server...');
        // send current token to server
        //$.post(url, {token: currentToken});
setTokenSentToServer(currentToken);
    } else {
console.log('Token already sent to server so won\'t send it
again unless it changes');
    }
}

function isTokenSentToServer(currentToken) {
    return
window.localStorage.getItem('sentFirebaseMessagingToken') ===
currentToken;
}

function setTokenSentToServer(currentToken) {
    if (currentToken) {
window.localStorage.setItem('sentFirebaseMessagingToken',
currentToken);
    } else {
window.localStorage.removeItem('sentFirebaseMessagingToken');
    }
}

```

```

function updateUIForPushEnabled(currentToken) {
    console.log(currentToken);
    token.text(currentToken);
    bt_register.hide();
    bt_delete.show();
    form.show();
}

function resetUI() {
    token.text('');
    bt_register.show();
    bt_delete.hide();
    form.hide();
    message_row.hide();
    info.hide();
}

function updateUIForPushPermissionRequired() {
    bt_register.attr('disabled', 'disabled');
    resetUI();
}

function showError(error, error_data) {
    if (typeof error_data !== "undefined") {
        console.error(error, error_data);
    } else {
        console.error(error);
    }
}

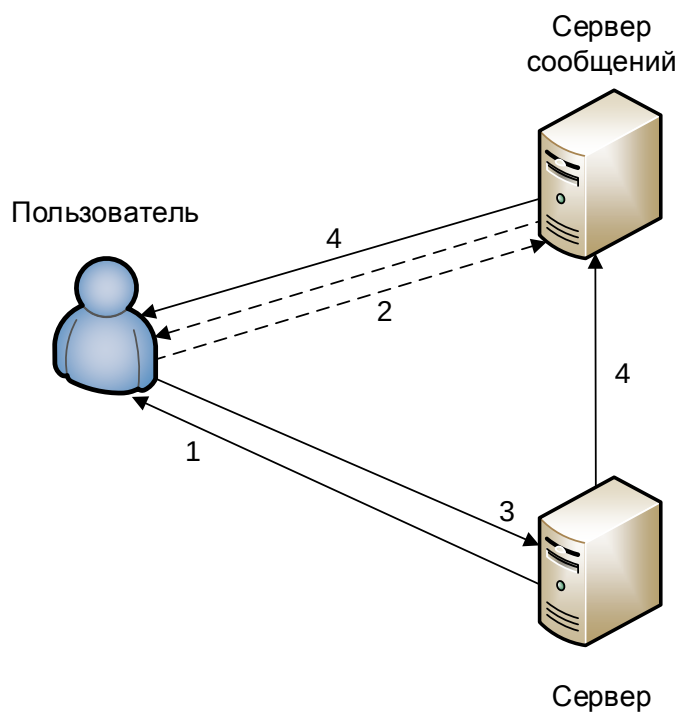
alert.show();
alert_message.html(error);
}

```

Изм.	Лист	№ докум.	Подпись	Дата

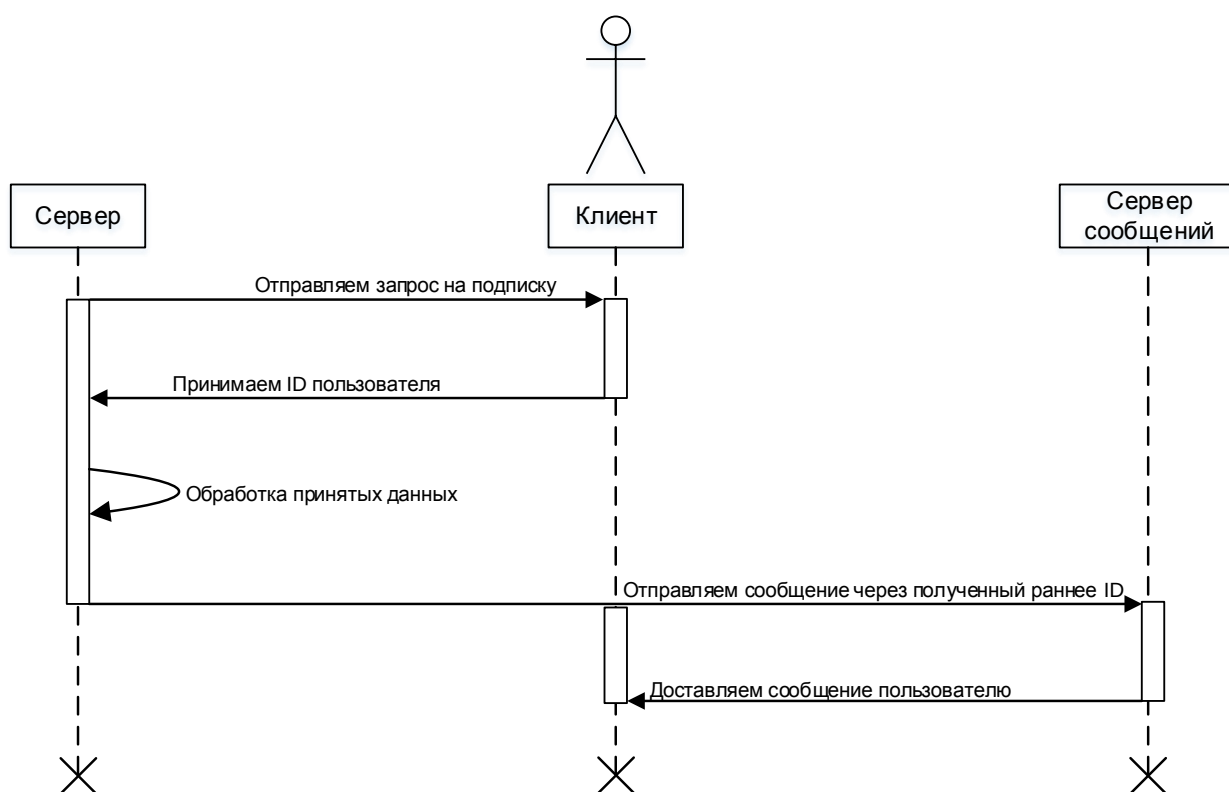
ПРИЛОЖЕНИЕ В UML ДИАГРАММЫ

Приложение В.1 Диаграмма взаимодействия



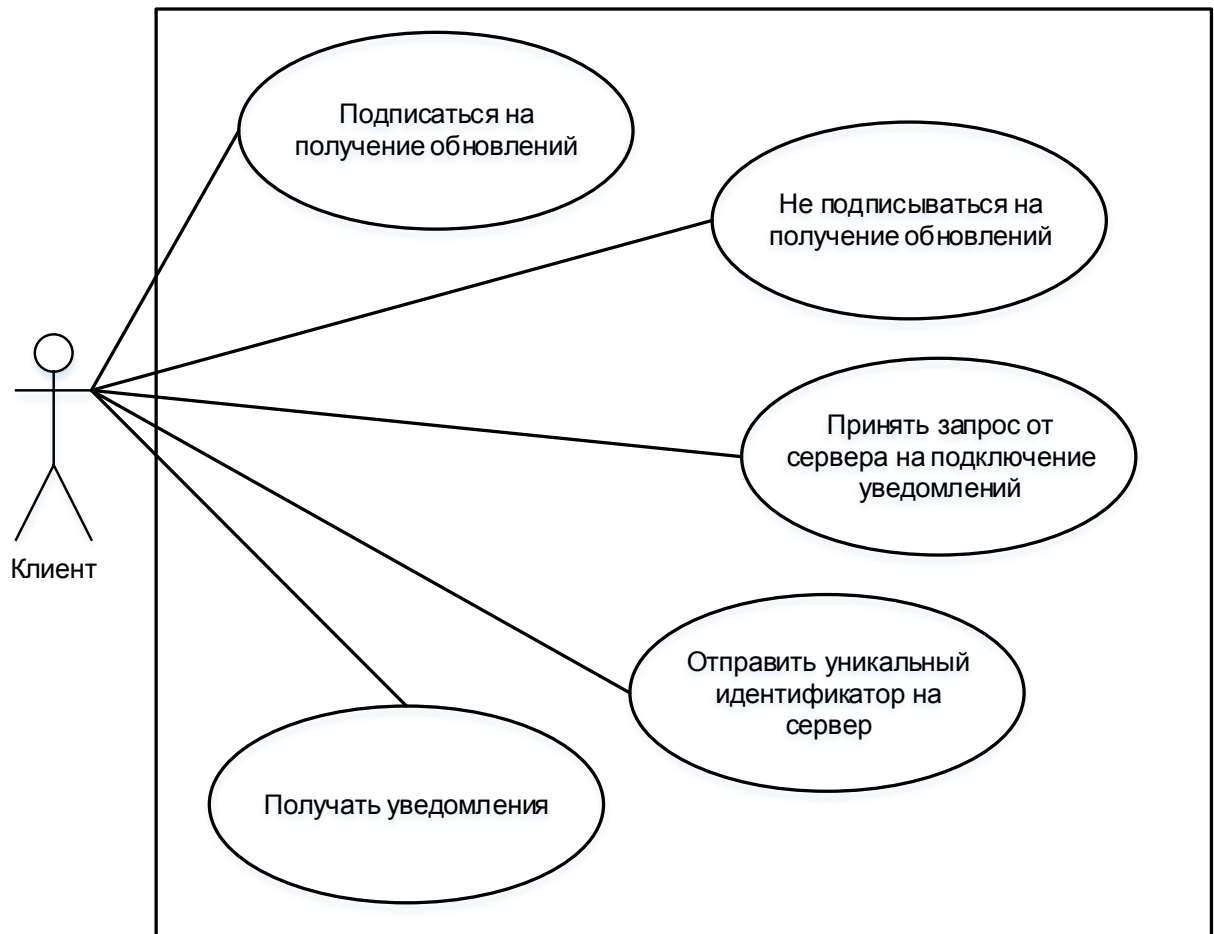
Приложение В.2

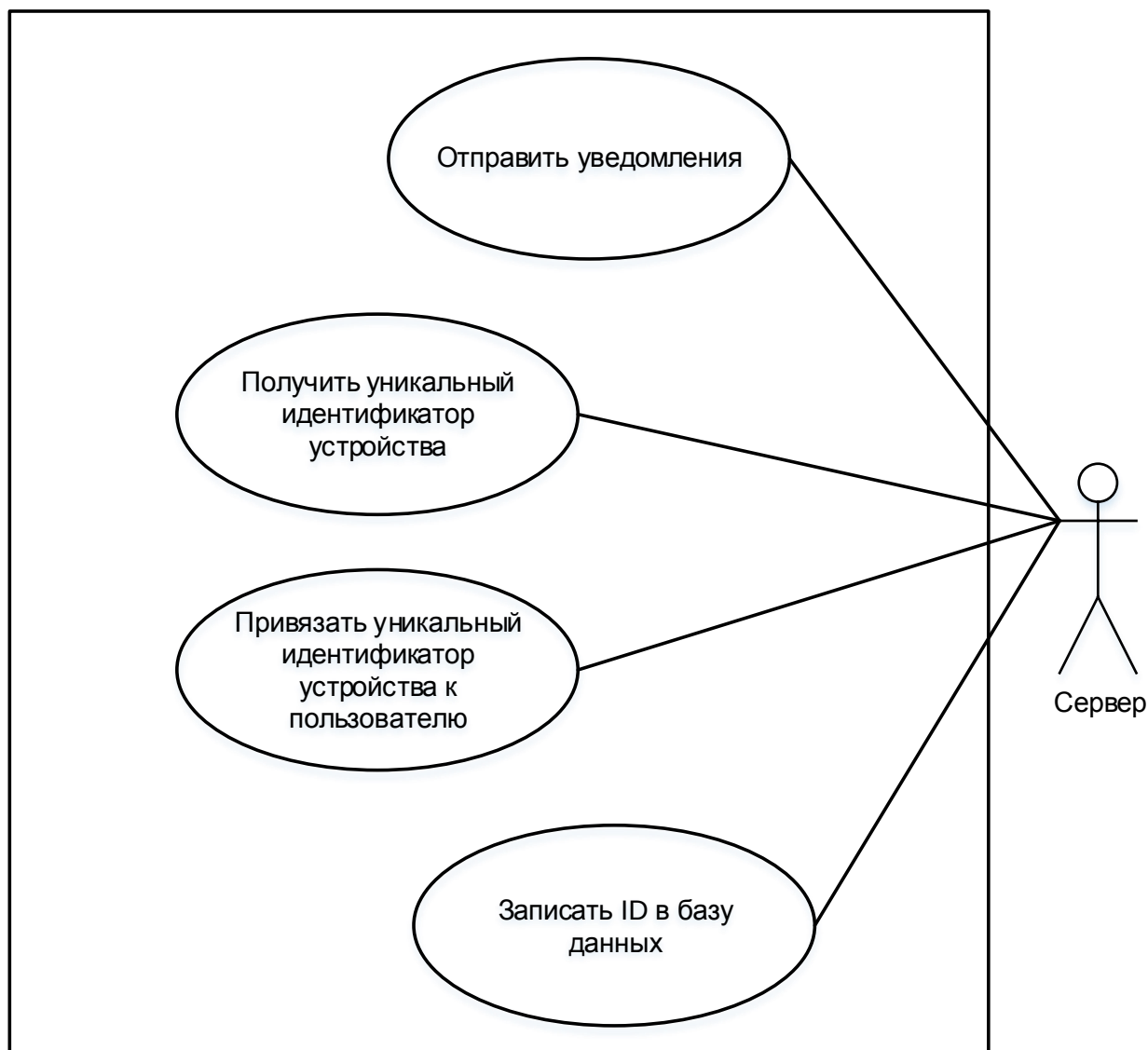
Диаграмма последовательности



Приложение В.3

Диаграмма вариантов использования





ПРИЛОЖЕНИЕ С
ПРЕЗЕНТАЦИЯ

					ПГУ 1.090301.05.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		52

ТЕМА РАБОТЫ:
«Система управления PUSH-
уведомлениями сайта кафедры
образовательной организации»

Автор работы: студентка группы 14BV1 Денежкина В.М.

Научный руководитель: д.т.н., профессор каф. ВТ ПГУ Пашенко Д. В.



Изм.	Лист	№ докум.	Подпись	Дата

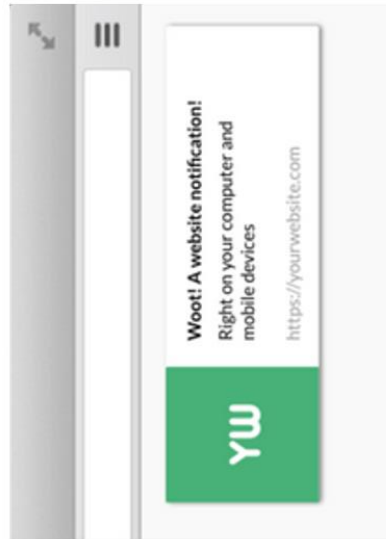
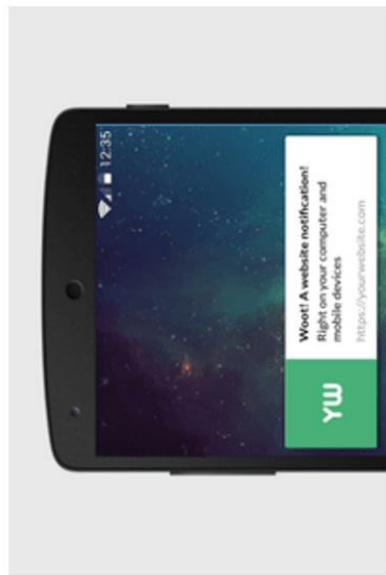
PUSH-уведомления

- PUSH-уведомления – это краткие всплывающие уведомления, которые появляются и сообщают о важных событиях и обновлениях. Любой гаджет, подключенный к интернету, способен принимать объявления такого рода. Основной целью PUSH-уведомлений является информирование пользователей об обновлениях веб-сайтов или приложений, добавлении нового контента, либо о каких-либо других новостях. Пользователь же в свою очередь либо отвергает, либо принимает данные.

Изм.	Лист	№ докум.	Подпись	Дата

Классификация PUSH-уведомлений

- Браузерные
- Мобильные



PUSH-сервисы



35000T.p.

letReach

MOST POPULAR

\$ 15 /mo
50,000 subscribers

Standard

- Hosted Subdomain
- Mobile Push Notifications
- Chrome & Mozilla Support
- Safari Support
- Multiple Opt-Ins
- Realtime Customizable
- Auto Geo Segmentation

Show More

\$ 280 /mo
50,000 subscribers

Features of Standard plan plus:

- Option to remove branding
- Upto 5 Domains
- API Calls: Subscribers X 45
- Downloadable Notification Performance
- Downloadable Subscriber Analytics
- Marketing Automation
- A/B Testing of Notifications

\$ 550 /mo
50,000 subscribers

Features of Pro plan plus:

- Option to remove/customize branding
- Upto 7 Domains
- Add upto 3 Sub-users
- API Calls: Subscribers X 60
- Re-targeting Module
- eCommerce Module

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ПГУ 1.090301.05.001 ПЗ

Цель работы – разработка технологии PUSH-уведомлений для оповещения всех пользователей на сайте о новых новостях

Задачи:

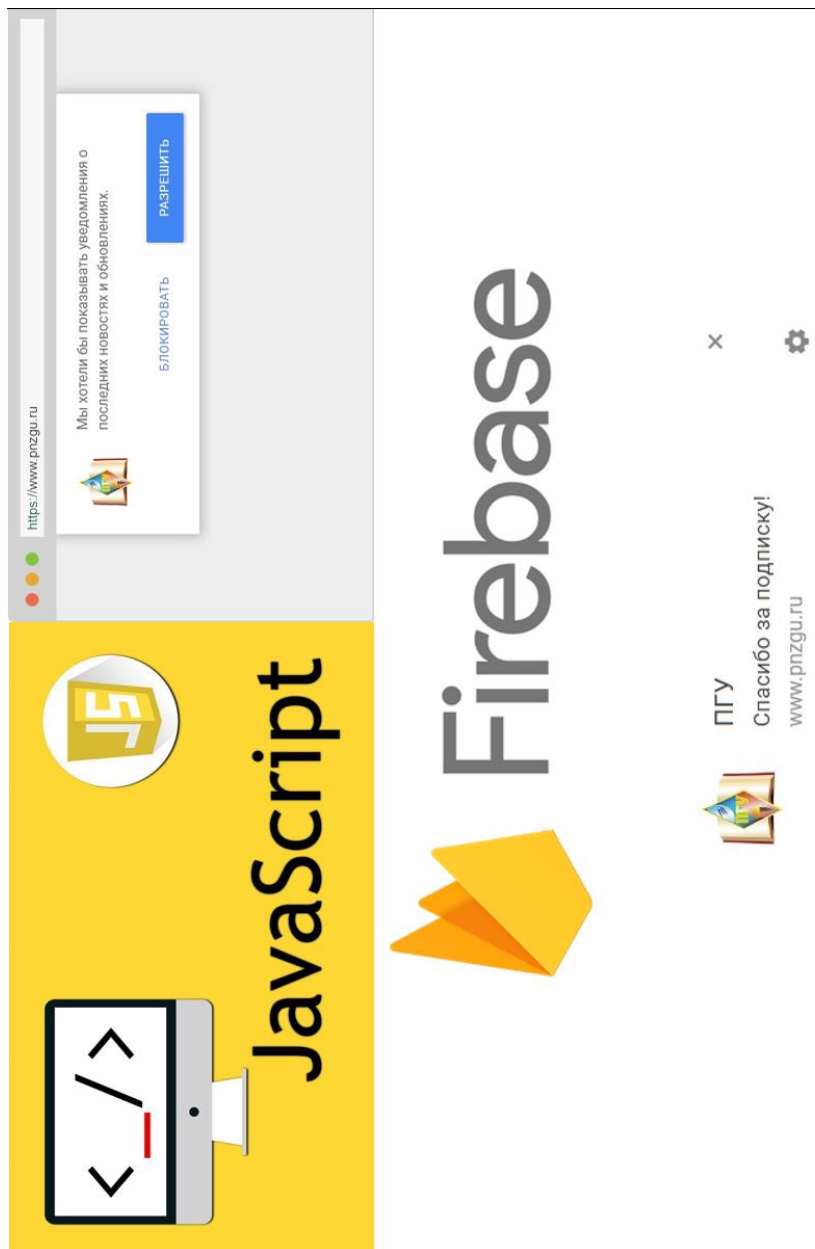
- Создание системы, обладающей понятным интерфейсом для всех пользователей
- Разработать технологию, чтобы в дальнейшем система могла быть внедрена на сайте университета или на сайтах кафедр университета

Изм.	Лист	№ докум.	Подпись	Дата

Подзадачи:

- Выбрать язык программирования
- Выбрать сервер сообщений
- Спроектировать и разработать клиентскую и серверную часть технологии

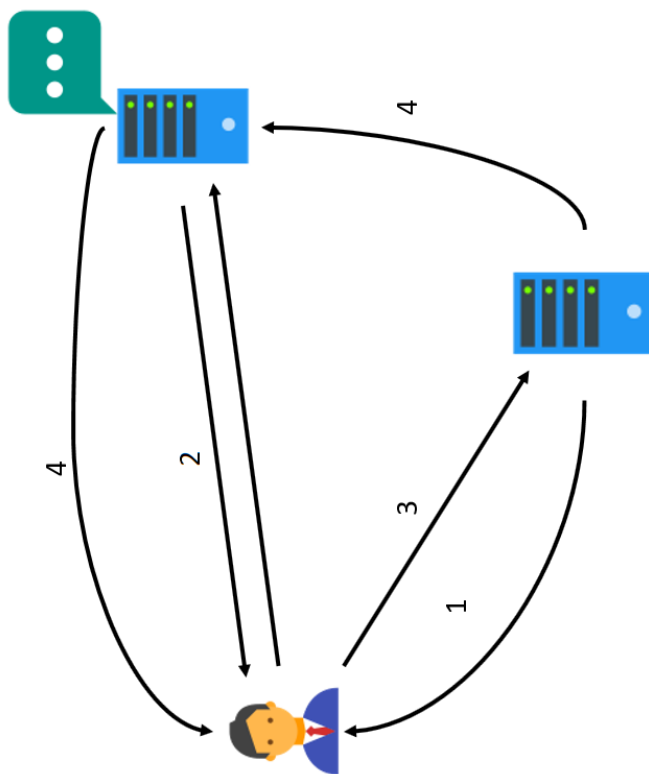
Изм.	Лист	№ докум.	Подпись	Дата



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.05.001 ПЗ

Схема взаимодействия

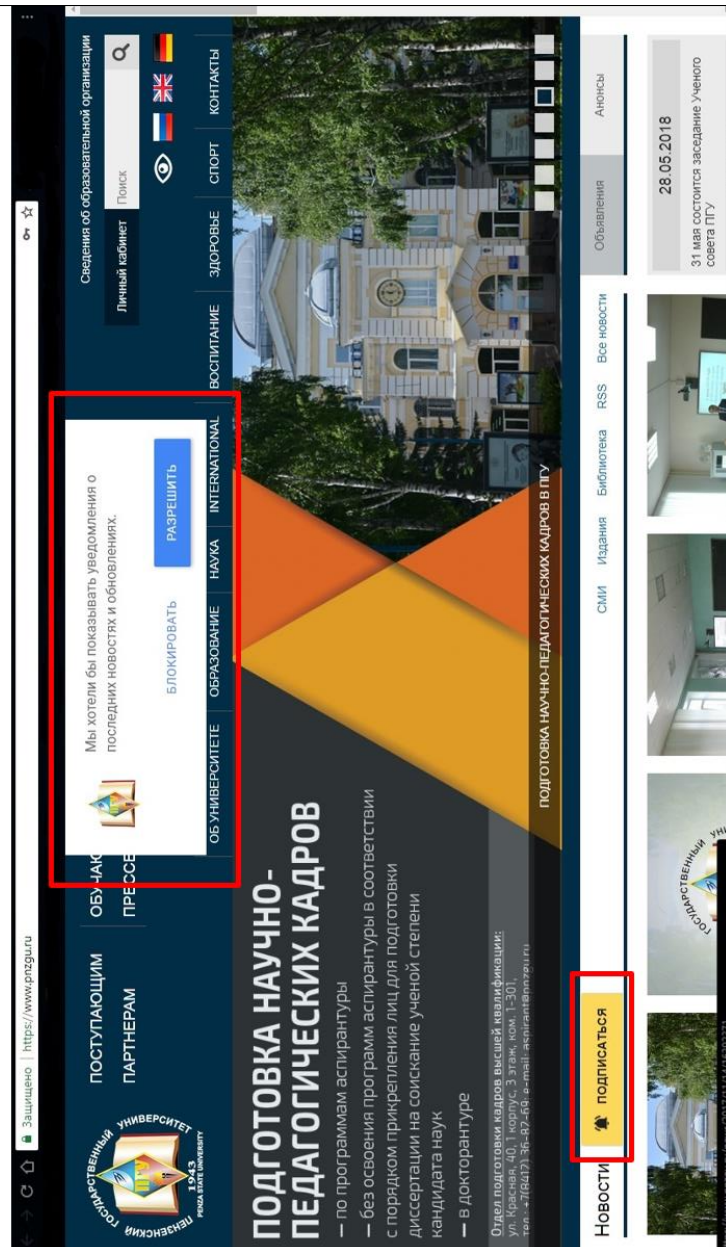


Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.05.001 ПЗ

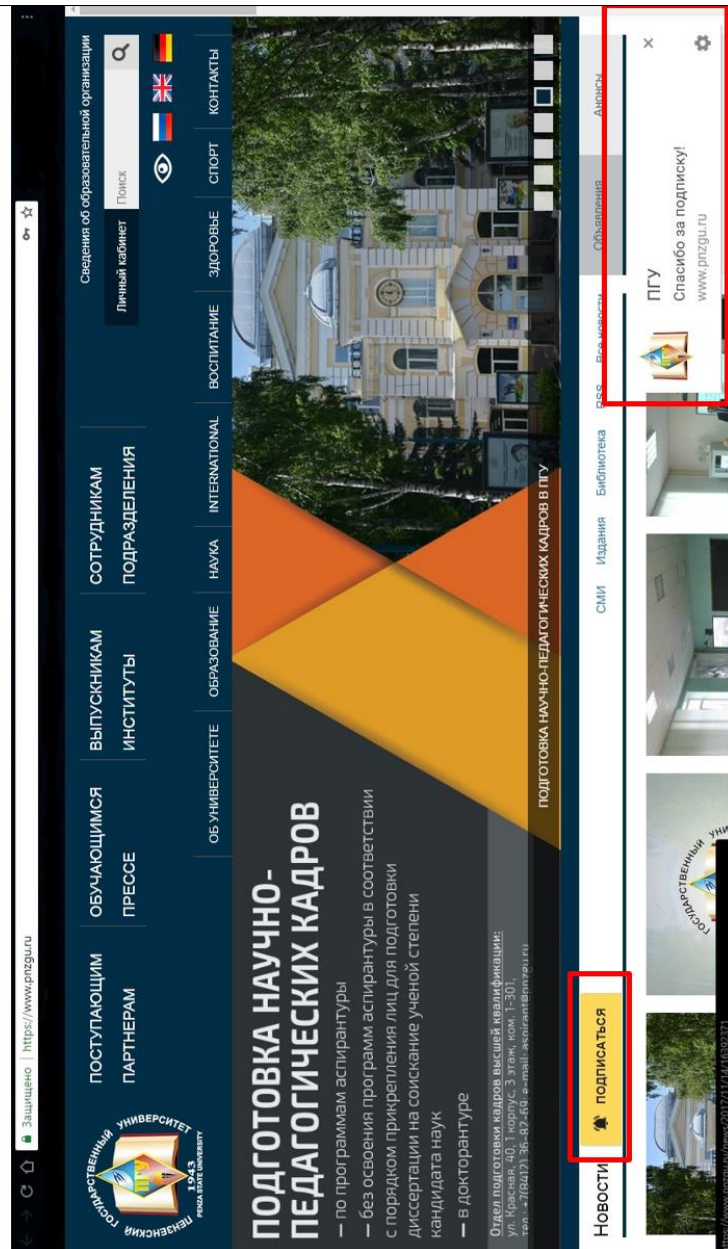
Лист

60



Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ПГУ 1.090301.05.001 ПЗ



Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ПГУ 1.090301.05.001 ПЗ

Спасибо за внимание!

					ПГУ 1.090301.05.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		63