*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

***«Московский государственный технический университет имени Н.Э. Баумана»   
(МГТУ им. Н.Э. Баумана)***

ОТЧЕТ

о прохождении непрерывной научно-производственной практики

на АО «ВПК «НПО машиностроения»

Исполнитель, студ. гр. АК5-121 Чуканцев Н.С.

2016 г.

Я, Чуканцев Н.С., во время обучения на Аэрокосмическом факультете МГТУ им. Н.Э. Баумана проходил непрерывную научно-производственную практику с декабря 2014 г. по настоящее время в отделе «системного, прикладного программно-математического и технического обеспечения».

За время прохождения практики было поставлено две задачи:

* изменение уже существующих и добавление новых возможностей в систему мгновенных сообщений клиента «Spark» и его сервера «Openfire»;
* Разработка репозитория инсталляционных пакетов.

Spark — свободная программа для мгновенного обмена сообщениями (мессенджер) в сети по протоколу Jabber. Программа представляет собой кроссплатформенное приложение с открытым исходным кодом, написанное на языке программирования Java. Openfire – это Jabber-сервер, так же написанный на языке программирования Java. Jabber - [открытый](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82), основанный на [XML](https://ru.wikipedia.org/wiki/XML), свободный для использования протокол для мгновенного обмена сообщениями и файлами. Модификация данных программных продуктов происходила в команде с моим одногруппником Мельниковым Никитой Михайловичем, который так же работал в этом отделе. Каждую неделю мы составляли отчет о проделанной работе для нашего руководителя.

Первой подзадачей была сборка вышеуказанных проектов – Spark и Openfire. Для этого было изучено средство сбора проектов для Java – Ant. Ant - [средство](https://ru.wikipedia.org/wiki/%D0%A3%D1%82%D0%B8%D0%BB%D0%B8%D1%82%D0%B0) для автоматизации процесса сборки программного продукта. Является платформонезависимым аналогом средства [make](https://ru.wikipedia.org/wiki/Make), где все команды записываются в [XML](https://ru.wikipedia.org/wiki/XML)-формате.

Следующей серией задач являлась более тонкая настройка мессенджера в нуждах АО «ВПК НПО «Машиностроения». Часть пользовательского интерфейса не имела перевода на русский язык. Этот недостаток был устранен добавлением переводов нелокализованных строк в файлах локализации. Поскольку не все возможности мессенджера необходимы сотрудникам АО «ВПК НПО «Машиностроения», некоторые из них были убраны с помощью модификации исходного кода.

Следующей была задача по переносу истории сообщений на серверную сторону. Данная задача затрагивала уже не только клиент Spark, но и сервер Openfire. Изначально, история сообщений хранилась у каждого пользователя на его рабочем компьютере. Если у пользователя было несколько компьютеров, то история сообщений на каждом их них отличалась и была не полной. В случае замены компьютера, история, хранящаяся на нем, терялась. Основная идея задачи заключалась в устранении данного недостатка путем переноса истории сообщений в базу данных сервера. Для этого были введены новые идентификаторы для системных сообщений. На сервере был добавлен обработчик системных сообщений и добавлена таблица в базу данных для хранения истории сообщений. Сообщения, отправляемые пользователями, не помечаются как системные и сохраняются в таблицу в базе данных. Для загрузки истории приложения формирует системное сообщение для запроса истории. Обработчик системных сообщений перехватывает такие запросы, формирует историю и отравляет клиенту для отображения пользователю.

Добавление идентификаторов сообщениям позволило расширить функционал программы: были введены специальные команды, доступные только администратору. Например, была добавлена специальная команда «/computerName». Если набрать такую команду в окне чата, то сервер ответит сообщением, содержащим название компьютера собеседника, за которым он работал. Чтобы данная возможность была доступна только администратору, была добавлена таблица в базу данных на сервере, хранящая роли у сотрудников – администратор и обычный пользователь.

После выполнения какой-либо задачи было необходимо пересобрать проект. Результатом сборки являлся файл с расширением «jar», который нельзя было без дополнительных средств установить на компьютере. Поэтому появилась необходимость в создании исполняемых файлов формата «exe» для их удобной установки. Для этого было изучено средство «Launch 4j». Оно предназначено для создания исполняемых файлов формата «exe» на основе «jar» файлов с указанием всех необходимых для работы программы библиотек. В результате получался «exe» файл, который можно установить на любой компьютер с ОС семейства Windows.

После выполнения всех поставленных задач были написаны руководства по внесению изменений в код программ Spark и Openfire.

Второй задачей являлась разработка репозитория инсталляционных пакетов. Основное назначение репозитория инсталляционных пакетов заключается в централизованном хранении инсталляционных пакетов и учете их версий. Пользователями данной программной системы могут быть работники предприятия АО "ВПК "НПО машиностроения".

В ходе работы были исследованы основные виды архитектур информационных систем с совместным использование файлов:

* Архитектура «файл-сервер»;
* Двухзвенная архитектура «клиент-сервер» в модификации «толстый клиент»;
* Двухзвенная архитектура «клиент-сервер» в модификации «тонкий клиент»;
* Трехзвенная архитектура «клиент-сервер».

Были рассмотрены основные достоинства и недостатки каждой из архитектур и на их основании была выбрана трехзвенная архитектура «клиент-сервер», как наиболее подходящая под поставленную задачу.

Система была реализована как веб-приложение на языке Java. В качестве СУБД был использован PostgreSQL. Такой подход позволил работать с системой без установки и настройки клиентского приложения на рабочие места, так как для работы с веб-приложением необходим только браузер. Так же, обновление такой системы будет происходить автоматически.

В ходе работы было произведено знакомство с Active Directory – служба каталогов корпорации Microsoft для операционных систем семейства Windows. После знакомства с ней, она была использована для реализации сквозной аутентификации в разработанном приложении.

В результате прохождения непрерывной научно-производственной практики на АО «ВПК НПО «Машиностроения», я решил поставленные передо мной задачи, ознакомился с новыми, неизвестными для меня технологиями и средствами, изучил основные архитектуры приложений, познакомился со службой каталогов Active Directory, познакомился с работой системных администраторов.