**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э.Баумана)**

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_Робототехника и комплексная автоматизация\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_Компьютерные системы автоматизации производств\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_Разработка подсистемы протоколирования работы мобильного приложения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент РК9-18(М) **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Прохорова А.И.\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Урусов А.В.**

(Подпись, дата) (И.О.Фамилия)

*2017 г.*

*Государственное образовательное учреждение высшего профессионального образования*

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

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_ г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине Моделирование дискретных производственных процессов

Разработка модуля управления шаговым двигателем1.

(Тема курсового проекта)

Студент Прохорова А.И. РК9-18(М)

(Фамилия, инициалы, индекс группы)

График выполнения проекта: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***1. Техническое задание***

1.1.Используя интегрированную среду разработки Android Studio разработать приложение с 3 экранами(Activity).

1.2. Изучить СУБД «ClickHouse»

1.3. Разработать библиотеку APPVisor

2. Оформление курсового проекта

2.1. Расчетно-пояснительная записка на 24 листах формата А4.

Дата выдачи задания «15» сентября 2015 г.

**Руководитель курсового проекта** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Урусов

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.И. Прохорова

(Подпись, дата) (И.О.Фамилия)

Примечание:

1. Задание оформляется в двух экземплярах; один выдаётся студенту, второй хранится на кафедре.

Содержание

[1. Введение 4](#_Toc476976481)

[2. Предпроектное исследование 5](#_Toc476976482)

[2.1 AppVisor 5](#_Toc476976483)

[2.2 Платформа 5](#_Toc476976484)

[Среда разработки и язык программирования 6](#_Toc476976485)

[2.3 База данных 7](#_Toc476976486)

[3. Формирование технического задания 8](#_Toc476976487)

[3.1 Введение 8](#_Toc476976488)

[3.2 Общие сведения 8](#_Toc476976489)

[3.3 Назначение разработки 8](#_Toc476976490)

[4.1 Структура библиотеки AppVisor 11](#_Toc476976491)

[4.1.1 Класс DataBaseHelper.java. 11](#_Toc476976492)

[4.1.2 Класс AppVisor.java 11](#_Toc476976493)

[Сложности возникшие на этапе разработки. 12](#_Toc476976494)

[Диаграмма тестового приложения 14](#_Toc476976495)

[Рабочий этап проектирования 15](#_Toc476976496)

# Введение

В эру цифровых технологий необходимо уметь собирать и анализировать данные. APPVisor – технология, которая позволит отслеживать работу клиента с приложением и собирать данные о наиболее часто используемых функциях, а также, благодаря детальному анализу полученных данных, выявить и исключить проблемы, возникающие в процессе работы.

APPVisor предполагается создавать по аналогии с технологией «Вебвизор». Вебвизор - технология, которая позволяет пользователям Яндекс.Метрики проанализировать поведение посетителей на своем сайте. Владелец сайта может воспроизвести действия посетителей в формате видео и узнать, что они делают на каждой странице, как осуществляют навигацию, передвигают курсор мыши, по каким ссылкам переходят.

Детальный анализ поведения посетителей помогает выявить проблемы в навигации, логике и комфортном пользовании, а в результате — повысить конверсию сайта.

В эру смартфонов и планшетов интересно использование вебвизора в другом формате, а именно применительно к мобильным приложениям. Решением данной проблемы является создание android-библиотеки, которая будет связана с базой данных ClickHouse. Полностью данная система и будет представлять из себя AppVisor.

## Предпроектное исследование

## AppVisor

Технология Аппвизор должна позволить разработчикам, использующим ее библиотеку, проанализировать поведение пользователей в своем мобильном приложении. Владелец приложения сможет собирать статистику по его использованию его, а именно: какие элементы его приложения наиболее активно используется, какая страница наиболее интересна пользователю, какие функции наиболее популярны, как часто происходят какие-либо события. Все действия будут считываться и записываться в базу.

Детальный анализ поведения пользователей помогает выявить проблемы в навигации, логике и удобстве использования приложения.

В первую очередь следует обратить внимание на поведение пользователей на ключевых страницах приложения. Это страницы, на которых пользователь осуществляет основные действия или принимает важное решение. К таким страницам, например, относятся:

* Страницы входа.
* Страница регистрации.
* Страница с формой поиска (товаров, подбора по параметрам).
* Страница оплаты, если таковая имеется.

Также особое внимание следует уделить **процессам**: процесс регистрации или оформления заказа. С помощью APPVisor можно узнать, например, на каком шаге регистрации приложение теряет пользователей и прозрачна ли навигация при оформлении заказа.

## 2.2 Платформа

В качестве платформы для APPVisor была выбрана операционная система Android. Т.к. она является наиболее популярной на современном рынке.

Android — операционная система для смартфонов, интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google, телевизоров и других устройств. В 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Изначально разрабатывалась компанией Android, Inc., которую затем купила Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (OHA), который сейчас занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать библиотеки и компоненты приложений, написанные на Си и других языках.

## Среда разработки и язык программирования

В качестве среды разработки была выбрана Android Studio, т.к. это официальное средство разработки Android приложений, и оно находится в свободном доступе.

Android Studio — это интегрированная среда разработки (IDE), основанная на программном обеспечении IntelliJ IDEA, для работы с платформой Android.

В качестве языка программирования был выбран язык Java. Данный язык программирования наиболее подходящий для создания приложений. Основным его достоинством является то, что программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Данным способом достигается полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве

## 2.3 База данных

Для хранения данных была выбрана база данных от компании Яндекс – ClickHouse.

ClickHouse - это столбцовая СУБД (система управления базами данных) для OLAP (online обработки аналитических запросов).

ClickHouse была разработана в Яндексе для решения задач Яндекс.Метрики - второй по величине системы web-аналитики в мире. ClickHouse позволяет выполнять аналитические запросы по обновляемым данным в режиме реального времени. Система линейно масштабируемая и способна работать с триллионами записей и петабайтами данных.

ClickHouse используется в ряде проектов внутри Яндекса, например, в open-source проекте Яндекс.Танк для хранения данных о телеметрии, Яндекс.Маркете для мониторинга здоровья сервиса и т.д. Кроме того, ClickHouse применяется и во внешних проектах, например, для анализа метаданных о событиях в LHCb эксперименте в CERN (порядка миллиарда событий и 1000 параметров для каждого события) и как хранилище в проекте Тинькофф Банка.

1. Формирование технического задания
   1. Введение
   2. Общие сведения

Основание для разработки: задание на курсовой проект

Заказчик: Кафедра «Компьютерные системы автоматизации производства».

Разработчик: студент кафедры «Компьютерные системы автоматизации производства» Прохорова А.И.

Наименование темы разработки: Разработка подсистемы протоколирования работы мобильного приложения.

* 1. Назначение разработки

Разработать библиотеку AppVisor, которая должна отслеживать нажатия на произвольные места на экране, кнопки, текстовые поля и записывать данные о нажатии в базу данных, для последующего анализа.

* 1. Требования к программе или программному изделию:
     1. Требования к функциональным характеристикам:

Система должна реализовывать:

* Сбор статистики
* Отображение на экране результатов
* Возможность выборки результатов по заданным временным рамкам
  + 1. Требования к надежности:

Основное требование к надежности направлено на поддержание в исправном и работоспособном состоянии ЭВМ, на которой происходит использование программного продукта

* + 1. Условия эксплуатации:
* Эксплуатация должна производиться на оборудовании, отвечающем требованиями к составу и параметрам технических средств, и с применением программных средств, отвечающим требованиям к программной совместимости
* Аппаратные средства должны эксплуатироваться в помещениях с выделенной розеточной электросетью 220В ±10%, 50 Гц с защитным заземлением
  + 1. Требования к составу и параметрам технических средств

Программный продукт должен работать:

1. На компьютерах со следующими характеристиками:

* объем ОЗУ не менее 2 Гб
* объем жесткого диска не менее 50 Гб
* микропроцессор с тактовой частотой не менее 2ГГц

1. На мобильных устройствах со следующими характеристиками:

* объем ОЗУ не менее 2 Гб
* Частота процессора не менее 2ГГц
  + 1. Требования к информационной и программной совместимости:
* Система компьютера должна работать под управлением ОС Linux Ubuntu не более старой, чем Precise (12.04) архитектура x86\_64 с поддержкой набора инструкций SSE 4.2.
* Версия Android мобильного устройства не ниже 4.2.2, возможность подключения к Wi-fi сетям.
  + 1. Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

* + 1. Требования к транспортированию и хранению

Требования к транспортированию и хранению не предъявляются.

* 1. Требования к программной документации

Требования к программной документации не предъявляются.

* 1. Стадии и этапы разработки

Плановый срок начала разработки – 10 октября 2015г.

Плановый срок окончания разработки – 19 декабря 2015г.

Этапы разработки:

3.7 Порядок контроля и приемки

Контроль и приемка работоспособности системы автоматизированной сборки, тестирования и развертывания должны осуществляться в процессе проверки функциональности (апробирования) системы путем многократных тестов в соответствии с требованиями к функциональным характеристикам системы.

1. Технический этап проектирования

## Структура библиотеки AppVisor

### Класс DataBaseHelper.java.

DataBaseHelper отвечает за работу с базой данных. Подключение к базе данных, запись события.

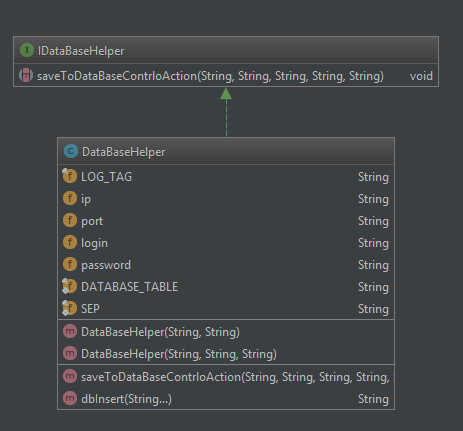
Поле **LOG\_TAG** используется для отладки приложения. В нем хранится имя класса. Log.d(LOG\_TAG, “Сообщение выводится в консоль”).

Поле **ip** – ip-адрес базы данных, к которой подключается библиотека AppVisor для записи действий пользователя.

Поле **port** – порт, к которому подключается библиотека AppVisor. По умолчанию, база данных ClickHouse использует порт 8123.

Поля **login** и **password** используются для создания защищенного соединения.

Поле **DATABASE\_TABLE** – название таблицы в которую будет производится запись действий пользователей.



### Класс AppVisor.java

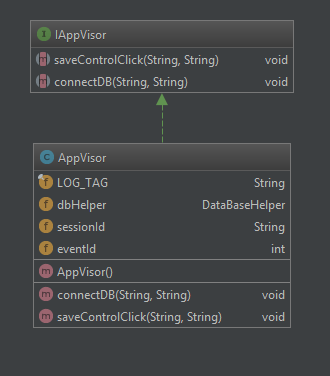
AppVisor отлавливает события нажатия на любой объект экрана (кроме пустых мест) и направляет событие в класс DataBaseHelper, для его сохранения.

Поле **LOG\_TAG** используется для отладки приложения. В нем хранится имя класса. Log.d(LOG\_TAG, “Сообщение выводится в консоль”).

В поле **dbHelper** хранится объект класса DataBaseHelper, который отвечает за работу с базой данных. (Подключение к базе данных, запись события)

Поле **sessionId** – это универсальный номер клиента(сессии), который записывается в базу данных, для разделения логов разных пользователей.

Поле **eventId** – номер события, начинающийся с 0 при создании объекта AppVisor, который итерируется при записи события в базу данных.



# Сложности возникшие на этапе разработки.

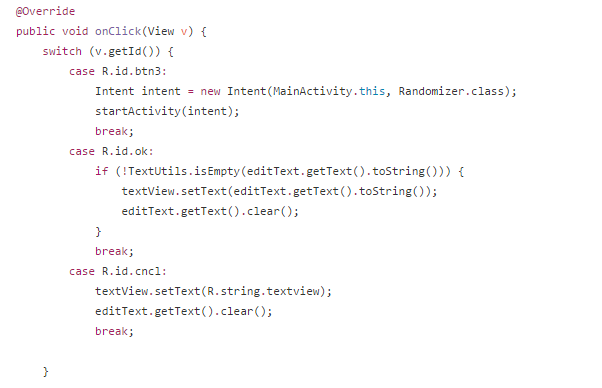
При изучении работы интерфейса View.OnClickListener, было выяснено, что невозможно переопределить так.

Существует три способа добавить слушателя на кнопку.

1. В главном Activity реализовать интерфейс View.onClickListener

public class MainActivity extends AppCompatActivity implements View.OnClickListener

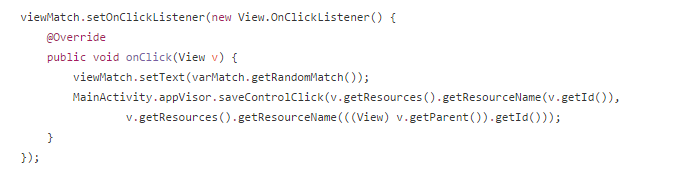
В реализации пишем



И тогда для кнопки, которую необходимо задействовать пишем

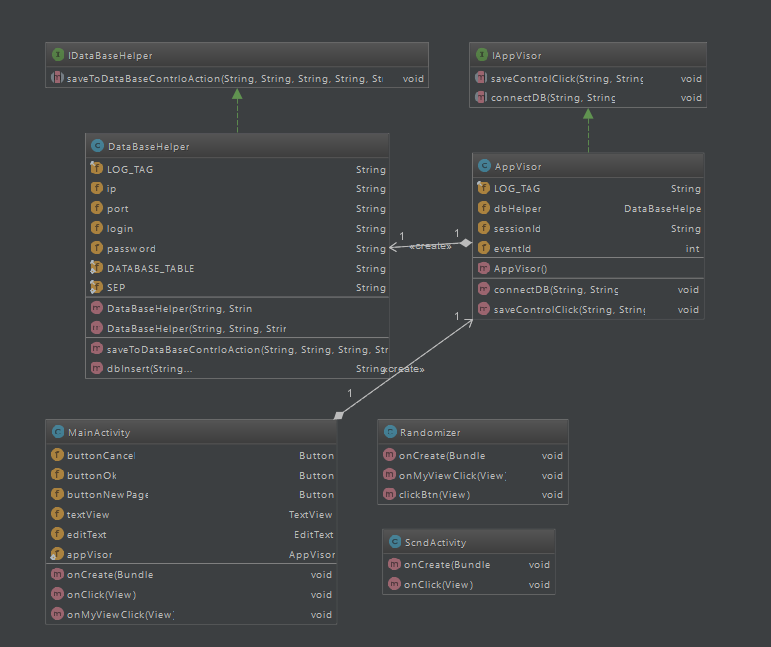
buttonCancel.setOnClickListener(this);

1. Второй способ — это использование анонимного класса при назначении кнопке «слушателя»



1. Третий способ - это создание своего класса, обрабатывающего нажатие на кнопку. При реализации такого способа, программисту, разрабатывающему приложение, понадобится заменить все обработчики события нажатия на кнопку. Например, все View.OnCliclListener необходимо заменить на AppVisor.OnClickListener.

# Диаграмма тестового приложения



# Рабочий этап проектирования

На рабочем этапе проектирования было разработано приложение с 3 экранами(Activity), с возможностью переключения между ними. Экраны содержат различные элементы графического интерфейса. Состояния элементов сохраняются при переключении между экранами. Также разработана библиотека AppVisor, которая должна отслеживает нажатия на кнопки, текстовые поля и другие элементы и записывает данные о нажатии в базу данных, для последующего анализа.