|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Факультет международных образовательных программ» (ФМОП)

«Информатика и системы управления» (ИУ)

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии» (ИУ7)

**Отчет по производственной практике**

Студент: Чаушев Александър Красимиров

Группа: ИУ7-46Б

Тип практики: Производственная практика

Название предприятия

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ МГТУ им. Баумана \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_**Куров A.В.**\_\_\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2020 г.*

Содержание

[Введение](#_gjdgxs) **3**

1. [Знакомство с проектом.](#_30j0zll) **4**

1.1 [Описание проекта](#_1fob9te) 4

1.2 [Структура, состав и основные компоненты систем мониторинга](#_3znysh7) 5

1. [Разработка мобильного приложения](#_2et92p0) **7**

[2.1 Выбор Архитектуры](#_tyjcwt) 7

[2.2 Сущность приложения](#_3dy6vkm) 8

[2.3 Алгоритм падения](#_1t3h5sf) 9

1. [Заключение](#_2s8eyo1) **11**
2. [Список используемых ресурсов](#_17dp8vu) **12**

# Введение

Существует ряд сфер деятельности, в которых человек серьезно рискует своим здоровьем. Работодатели заинтересованы в снижении количества несчастных случаев, поэтому необходимо организовать постоянный мониторинг за различными показателями сотрудника: его координатами, температурами тела и окружающей среды, частотой пульса, положением тела и другими.

В рамках практики, была разработана система мониторинга в целях предотвращения несчастных случаев или же ситуаций, в которых человек попадает в беду.

**Цель практики** - развитие практических навыков и умений, а также формирование компетенций, обучающихся в процессе выполнения определенных видов работ, связанных с будущей профессиональной деятельностью.

Задачи практики:

1. получение студентом первоначального профессионального опыта, опыта работы в трудовом коллективе;
2. проверка профессиональной готовности к самостоятельной трудовой деятельности;
3. формирование умений самостоятельного обучения новым методам исследования.

Индивидуальное задание на практику:

1. Разработать мобильное приложение которое обеспечивает безопасность сотрудника, собирает его жизненные параметры и параметры окружающей среды, и отправляет их на сервер. Добавить функцию обнаружения падения, которая определяет падения и, при необходимости, вызывает службы экстренной помощи. Спроектировать интуитивно-понятный интерфейс.

# Знакомство с проектом.

## Описание проекта

Проект обеспечивает безопасность работников производственных компаний на рабочем месте. Постоянный мониторинг на опасных для здоровья человека предприятиях может помочь как и человеку, так и компании. Трудовой кодекс РФ предусматривает выплаты по производственным травмам, обязывает часть платить и работодателям.

Несчастные случаи и травмы на рабочем месте очень дорого обходятся компаниям и их сотрудникам. Исходя из этого, многие работодатели заинтересованы в системе мониторинга различных показателей сотрудника, таких как GPS-локация, значение температуры, частота пульса и другие. Проект реализует комплексную систему, которая будет состоять из следующих частей:

* + центральный сервер - отвечает за хранение и обработки данных;
  + панель мониторинга службой безопасности - отвечает за вывод последних данных по каждому сенсору и оповещениях о тревоге;
  + носимое устройство - мобильный телефон Samsung A71, выданный сотруднику, к которому непосредственно подключаются датчики (по Bluetooth и проводам), задачей которого является передача показателей сенсоров на центральный сервер.

## Структура, состав и основные компоненты систем мониторинга

Первое звено, мобильное устройство и датчики, которые будут отправлять свои показатели на центральный сервер. В рамках практики было написано мобильное приложение на языке Java, в функционал которого входило отслеживание локации человека через встроенные в телефон Android сервисы GPS и отправку их на сервер. Отправка происходила по стеку TCP/IP по протоколу HTTP. Конечно, можно было отправлять данные сразу в базу данных, тем более многие устройства, такие как сенсоры, мобильные телефоны и другие могут отправлять данные по схеме подписчик-издатель по протоколу MQTT, но было принято решение проводить анализ этих данных на сервере и только после анализа их сохранять. Таким образом был устранен ряд возможных ошибок, которые не может обработать БД, но может серверное приложение.

Также в мобильное устройство помимо сенсора местоположения входила возможность отправить сигнал SOS, как отдельная реализованная функция. Сигнал SOS отправлялся, когда срабатывал сигнал падения.

О сигнале падения стоит сказать, что это еще одна возможность приложения. Алгоритм определения падения строился на встроенном акселерометре устройства.

Далее идет сам Центральный сервер, он может быть, как один, так и кластер серверов, в зависимости от количества объектов, за которыми идет мониторинг. Была использована технология Microsoft Asp Net Core WebAPI, приложение на ОС Linux и в качествен HTTP Серверов использован nginx( проксирующий сервера) и Kestrel для связи с кодовой базов.

Следующая часть проекта — это Базы данных, также можно иметь по одной для записей показателей и для хранения индикаторов сенсоров/девайсов так и кластер, если проект является большим и имеет высокую нагрузку. В качестве баз данных были использованы PostgreSQL- это объектно-реляционная БД. Очень удобная база данных, и удобство состоит в поддержке нетривиальных объектов, как например геолокация. Вторая база данных была для хранения показателей сенсоров, при том, что данные отсылались каждые 5 секунд, нужно было их быстро записывать, а поскольку реляционные БД используют индексирование, то процесс записи занимает огромное время при большом кол-ве уже записанных данных. Базы данных временных рядов отличный выход, и они адаптированы под быструю запись, мы использовали InfluxDb. Также политика хранения, это еще одна функция этих баз данных, устаревшие показатели удаляются и поэтому база данных не забивается.

Центр мониторинга, та часть, которая потребляется командой специалистов, благодаря ей можно визуализировать все показатели и действовать по анализу этих самых показателей. Было написано веб-приложение для мониторинга с использованием популярного фреймворка Angular на языке TypeScript, также использовали библиотеку Leaflet для добавления карт. На этом описание системы заканчивается.

# Разработка мобильного приложения

## 2.1 Выбор Архитектуры

Системная архитектура приложения определяет, как взаимодействуют элементы приложения. Приложение, взаимодействующее с пользователем, состоит из трех элементов: презентационного сервиса, прикладной логики и сервиса данных. Главная задача презентационного сервиса, или пользовательского интерфейса, — взаимодействие с пользователем. Бизнес-правила (или прикладная логика) — это набор алгоритмов, реализующих нужные вычисления и контролирующих поток управления в приложении. Сервисы данных управляют информацией, отвечают за ее сохранение и обеспечивают функциональность, необходимую для обработки данных.

Код, описывающий логику View, жёстко связан с пользовательским интерфейсом, так как он напрямую взаимодействует с элементами на экране. Это прямолинейный подход. Он применяется только для очень простых интерфейсов. Когда логика становится более сложной, поддержка такого UI может стать проблемой.

Проблемы заключается в том, что построение UI таким способом нарушает принцип единственной ответственности. Например, если хотим поменять тип пользовательского элемента, который используется для отображения данных, то изменения не должны повлиять на логику. Однако поскольку логика так тесно связана с элементами управления, ее тоже придется менять.

Цель MV\* - отделить UI-код (View) от кода логики (Presenter, Controller, ViewModel и т. д.) и кода обработки данных (Model). Это позволяет развивать каждую из частей по отдельности. Например, можно изменить внешний вид и стиль приложения, не затрагивая логику и данные.

Была использована архитектура [MVVM](#_17dp8vu) (Model-View-ViewModel). Она добавляет возможности связывания данных (data binding). Data binding использует [рефлексию](#_17dp8vu) (механизм исследования данных о программе во время ее выполнения в Java; рефлексия позволяет исследовать информацию о полях, методах и конструкторах классов), чтобы связать View и ViewModel. Так же ViewModel, позволяет [Activity](#_17dp8vu) и фрагментам сохранять необходимые им объекты живыми при повороте экрана и упрощает тестирование кода так как в его объектах не хранятся референции на UI. Таким образом, можно использовать любую ViewModel для View, которая предоставляет нужные свойства. Приложение состоит из двух Activity и двух ViewModel. MainViewModel это главный ViewModel, в котором выполняется вся необходимая логика для работы приложения.

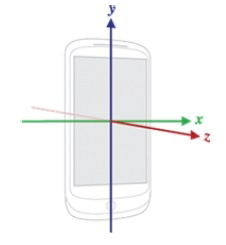
## 2.2 Сущность приложения

Мобильное приложение нацелено на обеспечение безопасность сотрудника, сбор его жизненных параметров и параметров, окружающей среды. Оно отслеживает показатели пользователя, и отправляет их на сервер.

Приложение имеет три основных экрана. Первый экран открывается после запуска приложение, дальше при удержании кнопки, пользователь подключается к серверу, и переходить на второй экран, где его данные уже отслеживаются и отправляются сервер каждые пять секунд в едином пост запросе, на сервер эти данные анализируются и отправляются в центр мониторинга. Если случится инцидент, приложение переедет на третий экран экстренного состояние, в которого можно отменить сигнал о помощи нажав на кнопку отменить. При падении осуществляется переход на экран падения.

## 2.3 Алгоритм падения

Был реализован алгоритм падения, в котором используется датчик акселерометра, встроенного в телефон. При падении рабочего, спасательная команда получит сообщение, которое включает GPS- координаты и информацию о пользователе. В случае падения, пользователь в течение 30 секунд может отменить тревогу, либо отправить сигнал о помощи.

Проблема в том, как обрабатывать данные, которые генерируются датчиком акселерометра и каков порог значение для создания оповещения. Акселерометр генерирует три значения ускорения в соответствии с осями, ad (значение ускорение) равна отношению силы (F) к массе (m), на которую влияет гравитационное ускорение (-g) . На основе этого уравнения, можно найти значения ax, ay, az, которые являются ускорением для каждой оси. значение g-force (G) является результатом деления ускорения ускорение силы тяжести. Это означает, что g-force (G) — это отношение величины ускорения к ускорению гравитации. Значение g-force будет примерно равно 1, когда смартфон лежит на столе, а находясь в состоянии движения значение g-force будет превышать 1. К тому времени, когда объект движется к земле, значение g-force будет приближаться ближе к числу 0. Причина использования G-Force есть, мы можем знать, движется ли объект к земле или нет.

# Заключение

В ходе практики по получению профессиональных умений и опыта в рамках выполнения индивидуального задания был создан программный продукт для мониторинга рабочих на предприятиях. В результате проведенной работы был изучен инструмент Android Studio . Получен опыт в разработке мобильных приложений и проектировании с использование паттерна MVVM.

За время прохождения практики было приобретено организационно-управленческие навыки, навыки составления и контролирования плана выполняемой работы, использования необходимых для выполнения работы ресурсов, оценивания результатов собственной работы. Был получен опыт работы в составе научно-исследовательского и производственного коллектива и решений задач профессиональной деятельности.

# Список используемых ресурсов

1. Android - Introduction to the Activities android [Электронный ресурс]. – Режим доступа: https://developer.android.com/guide/components/activities/intro-activities. – Дата доступа: 07.07.2020.
2. MVVM - полное понимание MVVM [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/338518/

Дата доступа: 15.07.2020.

1. Reflection - Рефлексия [Электронный ресурс]. – Режим доступа: https://javarush.ru/groups/posts/513-reflection-api-refleksija-temnaja-storona-java

Дата доступа: 16.07.2020.