

Команда Purple Dragons представляет проект Purple Wardrobe

# PURPLE WARDROBE

By Purple Dragons



г. Москва, АНО ДО «НОЦ МГТУ им. Н.Э. Баумана»

Участники: Назарова Полина Алексеевна

Фролова Ксения Николаевна

Руководитель: Секисов Давыд Михайлович

## Содержание

|  |    |
|--|----|
| 1. Презентация команды.....                        | 2  |
| 2. Социальный опрос.....                           | 2  |
| 3. Исследование.....                               | 6  |
| 4. Концепция.....                                  | 6  |
| 5. Этапы разработки проекта.....                   | 7  |
| 6. Анализ аналогов.....                            | 8  |
| 7. Механическая часть                              |    |
| 6.1. Система конвейерных модулей.....              | 9  |
| 6.2. Система транспортировки вещей.....            | 10 |
| 6.3. Система подачи вещей.....                     | 11 |
| 6.4. Система складывания вещей типа “бабочка”..... | 11 |
| 6.5. Элементы управления и питания.....            | 12 |
| 6.6. Моделирование компонентов.....                | 13 |
| 8. Программная часть                               |    |
| 7.1. Telegram bot.....                             | 13 |
| 7.2. Приложение.....                               | 14 |
| 7.3. Чтение штрихкодов.....                        | 14 |
| 7.4. Голосовой помощник и интеграция с Алисой..... | 15 |
| 7.5. Рекомендательная система.....                 | 15 |
| 7.6. Модуль Purp_Ward.....                         | 17 |
| 7.7. Данные в системе.....                         | 18 |
| 7.8. ТСР-сервер.....                               | 18 |
| 9. Социальное взаимодействие.....                  | 19 |
| 10. Экономическая составляющая.....                | 20 |
| 11. Перспективы развития.....                      | 22 |
| 12. Развитие по сравнению с прошлым годом.....     | 23 |

## Презентация команды

Наша команда называется “Purple Dragons”. Мы из Москвы. Команда состоит из двух человек:

- Полина - программист
- Ксюша - механик и схемотехник
- Давыд - самый лучший в мире куратор :-)



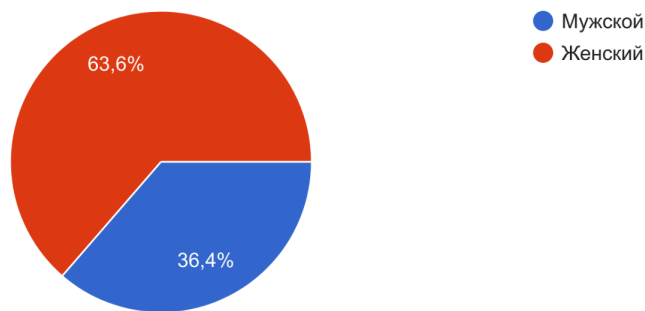
Рисунок 1 - Фотография команды с МКОР.

## Социальный опрос

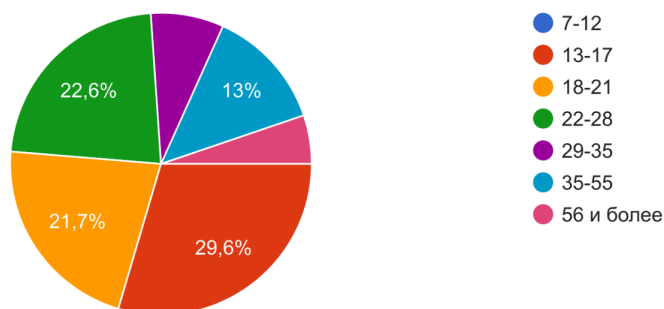
Мы провели опрос в котором поучаствовало 114 человек: 63% женщин и 37% мужчин. Мы опросили абсолютно разную аудиторию. Мы хотели исследовать проблему затраты большого количества времени на сбор образа и последствия этих затрат.

По результатам нашего опроса использование нашего шкафа сократит время на подбор образа 60 процентам опрошенных на 63 процентов, сократит время на сортировку вещей после стирки 42 процентам опрошенных. Наш проект гораздо облегчит жизнь людям с нарушениями опорно-двигательной системы, которые составили 9 процентов опрошенных, а информационные системы нашего шкафа отлично подойдут респондентам, которые активно пользуются онлайн календарями и телеграммом, которых 58% и 95% от опрошенных соответственно. Также убережет от дискомфорта 41% респондентов и от опозданий - 40%.

Ваш пол  
110 ответов

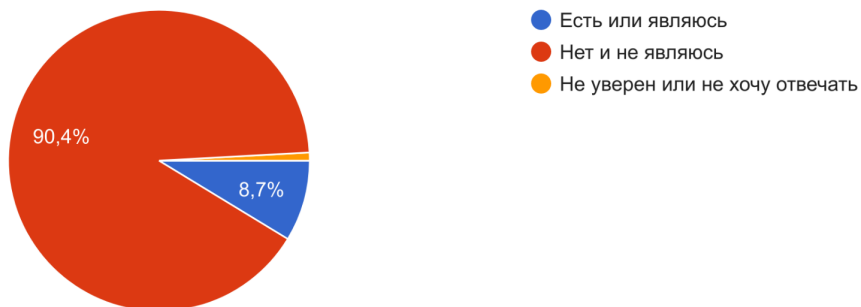


Ваш возраст (лет)  
115 ответов



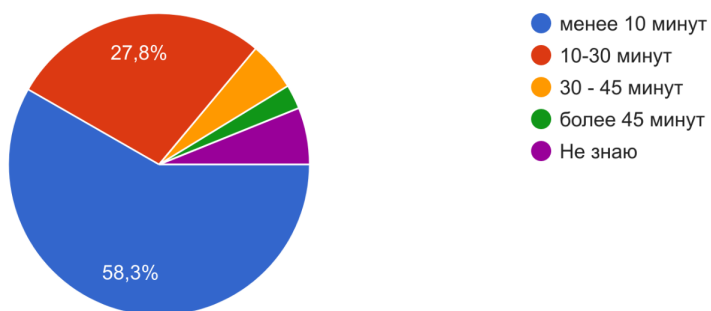
Являетесь ли вы или члены вашей семьи\* людьми с нарушением опорно-двигательного аппарата?

115 ответов



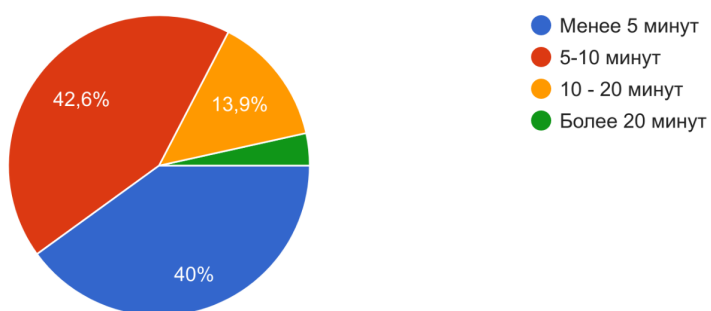
Сколько времени вы тратите на сортировку одежды после стирки в шкафу?

115 ответов



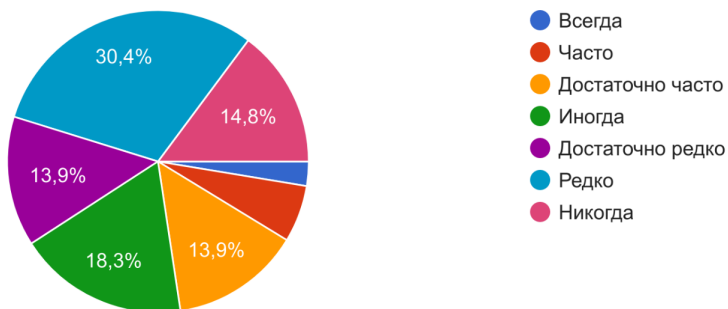
Сколько вы тратите времени на выбор образа (преимущественно утром)?

115 ответов



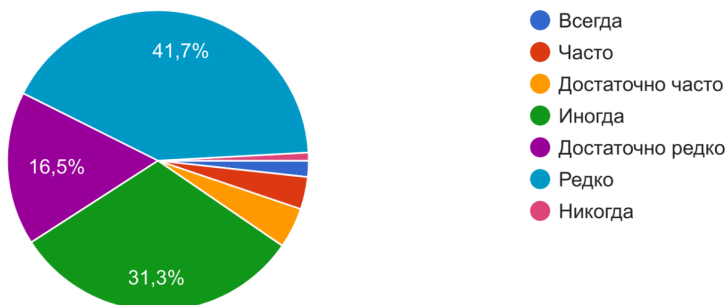
### Как часто вы опаздываете на учебу/работу?

115 ответов



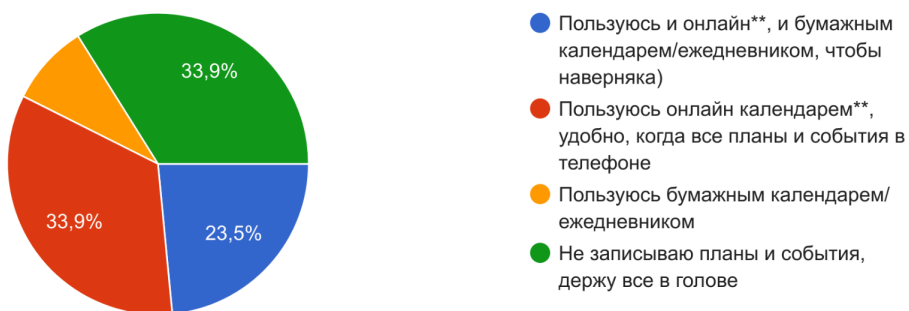
### Как часто вы промахиваетесь с выбором одежды по погоде?

115 ответов



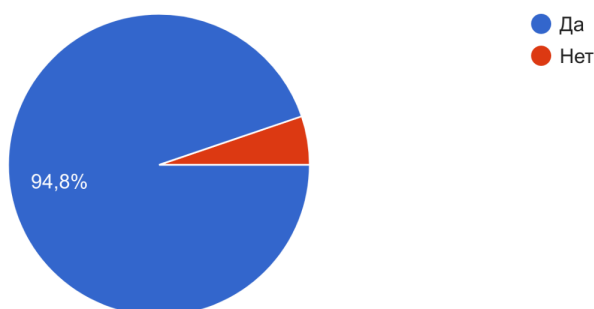
### Пользуетесь ли вы календарем?

115 ответов



Пользуетесь ли вы Telegram?

115 ответов



## Исследование

При выборе темы для проекта мы решили обратиться к нашим повседневным бытовым проблемам и пришли к выводу, что тратим слишком много времени на подбор одежды. Ведь нужно учесть множество факторов: запланированные на день мероприятия, погоду на улице, гармоничность образа. Мы провели социальный опрос, и оказалось, что данная проблема актуальна для всех возрастов и людей с разными возможностями здоровья.

## Концепция

Наше решение - это умный шкаф Purple Wardrobe. Система, исходя из данных о запланированных Вами мероприятиях сама подберёт Вам образ. Вам не придется тратить время, чтобы отыскать нужную вещь в шкафу, система транспортировки вещей шкафа сама сложит на полку выдачи необходимые предметы гардероба. Вам достаточно выбрать образ с помощью Телеграмм бота или голосового помощника.

Нет более важного ресурса, чем время. Purple Wardrobe поможет сэкономить его абсолютно каждому. Также он особенно поможет маломобильным и пожилым людям, которым зачастую практически невозможно достать что-либо, например, с высокой полки.

## Этапы разработки проекта

|   |               |
|---|---------------|
| Создание концепции  | 20.03 - 25.03 |
| Моделирование системы/ Разработка приложения  | 25.03 - 04.04 |
| Закупка составных компонентов/ Разработка базы данных   | 04.04 - 06.04 |
| Сборка корпуса шкафа/ Разработка считывателя штрихкодов                                       | 07.04 - 19.04 |
| Разработка и сборка системы конвейерных полок/ Разработка модуля <code>purp_ward</code>       | 10.04 - 14.04 |
| Разработка и сборка системы транспортировки вещей/ Разработка Telegram bot                    | 15.04 - 24.04 |
| Разработка и сборка системы подачи вещей/<br>Разработка рекомендательной системы              | 16.05 - 06.06 |
| Разработка и сборка системы складывания вещей типа “бабочка”/ Разработка голосового помощника | 06.06 - 22.06 |
| Написание программного кода Arduino/<br>Разработка TCP-сервера                                | 07.06 - 19.06 |
| Отлаживание всех систем и их доработка  | 19.06 - 22.06 |
| Тестирование системы  | 23.06 - 25.06 |



## Анализ аналогов

В ходе изучения аналогов мы обнаружили такие версии “умного шкафа”:

- Паровой шкаф LG Styler. Данная разработка служит для сушки и отпаривания вещей.



Рисунок 2 - Паровой шкаф LG Styler.

- Умный гардероб Laundroid. Роботизированный шкаф способен с помощью искусственного интеллекта рассортировывать сложенные вещи по полкам.



Рисунок 3 - Умный гардероб Laundroid.

Наше решение можно назвать инновационным, потому что Purple Wardrobe это не просто сортировщик или отпариватель вещей, это персональный стилист, скоростной помощник и функциональное пространство для хранения вещей.

### **Механическая часть**

#### **Система конвейерных модулей**

В шкафу реализована система конвейерных полок. В качестве конвейерного полотна выступает эластичная резинка, её расположение в виде двух полос обеспечивает циркуляцию воздуха внутри шкафа, позволяет осям не прокручиваться, полотно в середине не провисает. Для движения конвейерной ленты используются моторы-редукторы.

## Система транспортировки вещей

Система транспортировки вещей в нашем шкафу похожа на систему внутри вендингового аппарата. Движение ездящей полки осуществляется по двум осям: X и Y. Для движения используются шаговые моторы. Крутящий момент передаётся с помощью винтовой передачи. В начале работы над проектом мы столкнулись с несколькими проблемами: сбор комплекта одежды занимал очень много времени, т. к. движение полки было слишком медленным, а также при движении по горизонтальной оси возникали сильные колебания полки. Первую проблему мы решили путём замены шпилек с метрической однозаходной резьбой 8 мм и шагом 1,25 мм на шпильки с трапецеидальной четырёхзаходной резьбой 8 мм с шагом 2 мм. А от колебаний полки мы избавились, повысив жёсткость опорных конструкций.

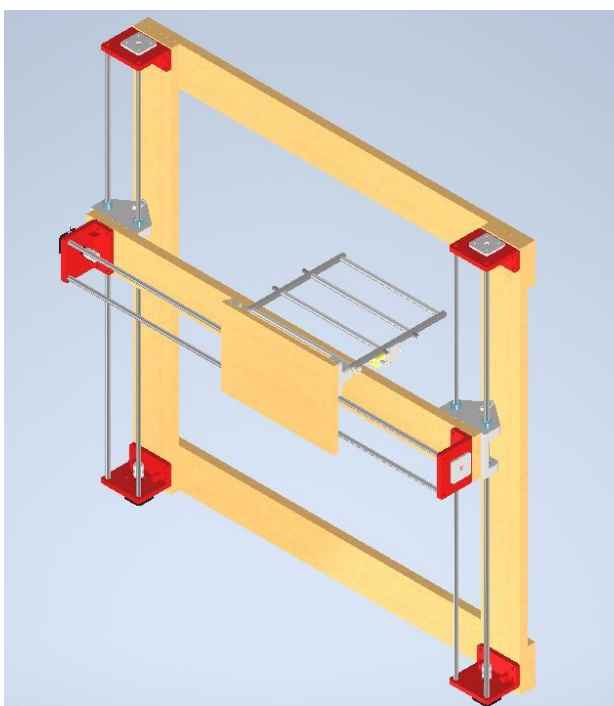


Рисунок 4 - 3D-модель системы транспортировки вещей.

### **Система подачи вещей**

Сложенная вещь через окошко попадает на вилообразную полку вертикального конвейера, далее вещь спускается на вилообразную полку, расположенную в шкафу, с конвейерной лентой, которая “перехватывает” вещь. Движение конвейера осуществляется за счёт мотора-редуктора. Для определения положения полок вертикального конвейера используется энкодер. Геркон, вынесенный на небольшой выступ, используется для калибровки системы подачи вещей.

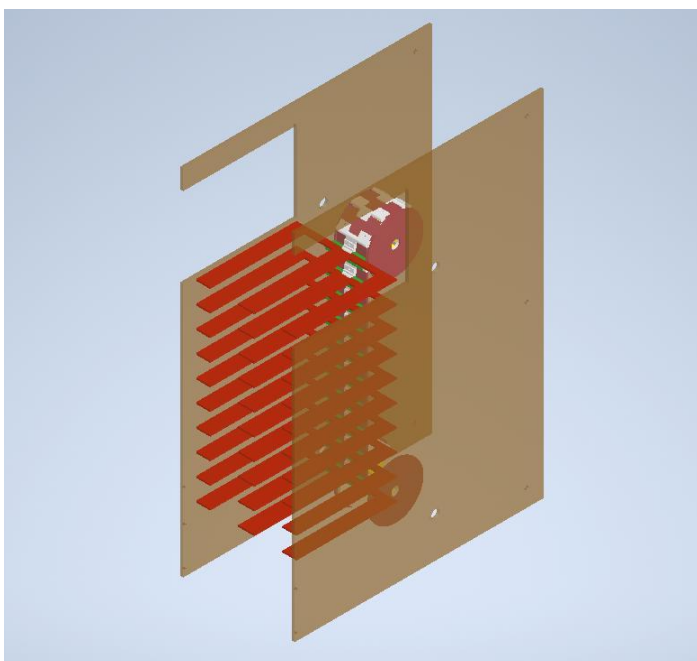


Рисунок 5 - 3D-модель вертикального конвейера системы подачи вещей.

### **Система складывания вещей типа “бабочка”**

Система “бабочка” состоит из двух одинаковых вилообразных модулей, на который крепится небольшой конвейер. Чтобы сбросить вещь, модули поворачиваются на  $90^\circ$  с помощью сервопривода. После попадания с полки на вертикальном конвейере на вилообразную полку вещь перемещается на одну из полок типа “бабочка”, которая сбрасывает вещь на конвейерную полку подачи, то есть полку, с которой система транспортировки может забрать вещь. Далее вещь при необходимости перемещается на вторую

полку типа “бабочка”, которая используется для складывания вещей в стопку.

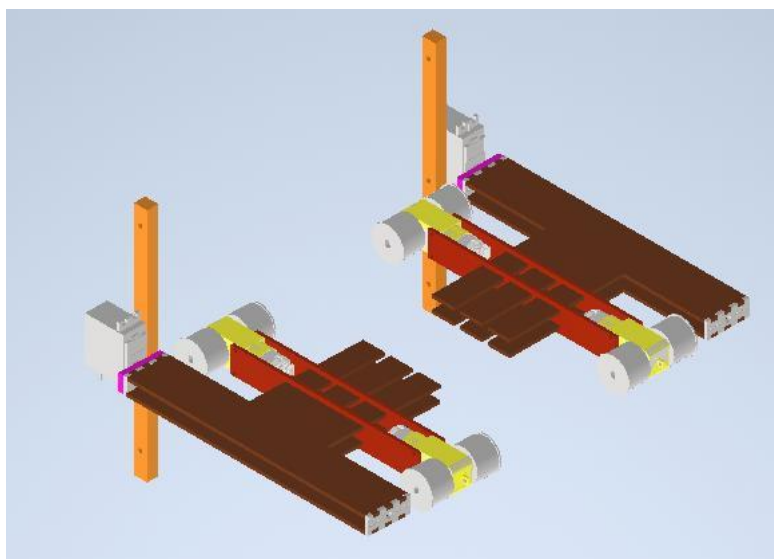


Рисунок 6 - 3D-модель системы складывания вещей типа “бабочка”.

### **Элементы управления и питания**

Для управления механической частью в умном шкафу используются две платы Arduino Mega и одна плата Arduino Uno. Одна из плат Arduino Mega с CNC шилдом используется для управления шаговыми моторами системы транспортировки, моторами-редукторами на конвейерных потоках и управления одной полкой системы “бабочка”, эта плата связана с двумя другими с помощью протокола I2C и получает оттуда все необходимые команды. Другая плата Arduino Mega с Ethernet шилдом используется для обработки показаний энкодера, геркона и управления второй полкой системы “бабочка”. Плата Arduino Uno с Ethernet шилдом служит для связи с сервером посредством TCP-соединения и управления светодиодной лентой. Вся система запрограммирована на языке C++ в среде Arduino IDE. В качестве источника питания используется блок питания на 12 В и 5 В.

## Моделирование компонентов

При разработке конструкции в среде Autodesk Inventor были полностью смоделированы система транспортировки, подачи вещей и система складывания вещей типа “бабочка”. Часть деталей для сборки вырезаны из трёхмиллиметровой фанерной доски на станке для лазерной резки, а часть напечатана на 3D-принтере из PLA пластика, в виду его экологичности.

## Программная часть

### Telegram bot

Для создания бота использовался [BotFather](#). Мы добавили картинку, команды и описание. Добавили функционал, с помощью которого сможет управлять шкафом с помощью смартфона!

Бот написан на python библиотеки с открытым кодом-[telebot](#). Взаимодействие с телеграмом происходит с помощью api. API (Application Programming Interface) - конструкции языка программирования, позволяющие разработчику строить сложную функциональность с меньшими усилиями. Они "скрывают" более сложный код от программиста, обеспечивая простоту использования. Проще говоря, api-переводчик с языка программиста на язык приложения. В нашем случае Telegram представляет своё собственное API для более удобной реализации тех, или иных задач, связанных с автоматизацией всевозможных процессов.

Функции telegram bot:

1. Посмотреть список всех вещей
  - a. С помощью библиотеки [csv](#) считываем всю базу данных, далее обрабатывается и отправляется
2. Достать одну вещь из гардероба
  - a. С помощью нашего модуля `purp_ward` вызывается функция, которая меняет `clothes.csv`(нашу единую базу данных) и `choose.txt` (файл, который анализируется tsp-сервером)

### 3. Собрать образ

- a. С помощью нашего модуля `purp_ward` вызывается функция, которая возвращает `pd.DataFrame` с отсортированными вещами (выше самые рекомендованные)

### 4. Добавить и изменить изображение

- a. Реализовано с помощью кнопок в Telegram

### 5. Посмотреть следующее событие и его стиль

- a. С помощью нашего модуля `purp_ward` вызывается функция, которая с помощью [API Google Calendar](#) парсит и идентифицирует стиль следующего события по ключевым словам

b.

## Приложение

Приложение написано на языке программирования C++, использованный последним стандартом данного языка - C++20. При его написании мы использовали фреймворк [QT](#), позволяющий реализовывать интерфейс программы для всех известных платформ (Windows, Linux, MacOS). У самой программы на данный момент есть 6 функций: положить вещи, зарегистрировать вещи, просмотреть список всех вещей, достать одну вещь из гардероба и собрать образ из вещей. Также реализован переход между окнами с помощью ООП, что делает использование приложения очень удобным. Есть окно основного меню, окно для считывания штрихкода, окна для отображения информационной карточки и списка всех вещей с возможностью взятия определенной, также есть окно регистрации.

## Чтение штрихкодов

Это скрипт на языке программирования Python, интегрированный в приложение, позволяющий с помощью обычной камеры считывать штрих код. Сначала мы переводим каждый кадр в черно-белое изображение, а затем мы используем функцию `threshold`, которая возвращает изображение, в котором

все пиксели, которые темнее (меньше) 135 заменены на 0, а все, которые ярче (больше) 127, — на 255. Таким образом мы четко выделяем контуры и черные полосы штрихкода. Далее с помощью библиотеки [pyzbar](#) мы считываем штрихкод с высокой точностью. Если 4 кадра подряд считывается одинаковый штрихкод (цифры одинаковые), то мы считаем, что мы его правильно распознали и записываем. Это работает быстро, с потоковым видео, и точно.

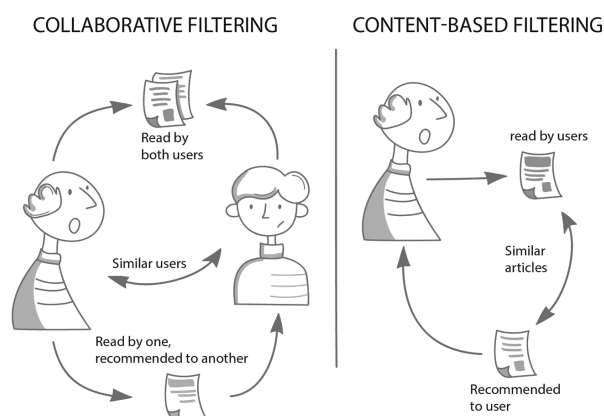
## Голосовой помощник и интеграция с Алисой

Мы смогли научить «Алису» новым вещам, используя Python и веб-фреймворк Flask, это называется навыком. Как вообще работают навыки? Если говорить простыми словами, то «Алиса» и наш сервер обмениваются файлами в формате JSON, где содержится необходимая информация. Причем на сервер приходит не просто распознанный текст, но уже подготовленный для обработки запрос. Наш навык позволяет пользователю голосом выбрать вещь, которую шкаф выдаст или спросить про следующее событие в его календаре.

## Рекомендательная система

Мы взяли за основу рекомендательную систему типа CONTENT-BASED FILTERING, чтобы не зависеть от других пользователей, но поскольку сложно выбирать “похожие” вещи, ведь это будет крайне субъективно, поэтому мы разработали свой алгоритм подбора одежды. У нас особый метод определения стиля вещи.

Всего у нас пять стилей: деловой, спортивный, парадный, ежедневный и, так называемый, молодежный. А стиль каждой вещи - это 5 чисел - проценты того,



насколько эта вещь принадлежит каждому стилю. Таким образом вещь может менять стили в процессе эксплуатации. При регистрации указывается изначальный стиль и это

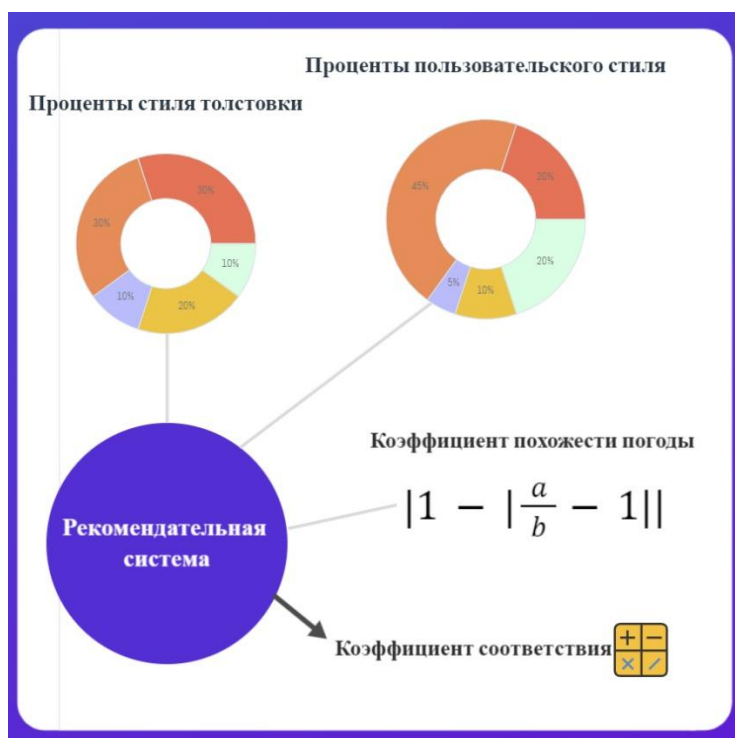


приравнивается к 1 взятию вещи. Далее после каждого взятия будет пересчет стиля. Он считается как процент взятия вещи в этом стиле от общего кол-ва взятий.

При ранжировании подходящих под следующее события вещей у нас есть 2 фактора. Первый фактор - соответствие пользовательскому стилю. Это опять же 5 чисел - проценты того, насколько часто пользователь берет тот или иной

стиль. Допустим, пользователь почти постоянно ходит в ежедневном стиле, тогда в пользовательском стиле ежедневный стиль, очевидно, будет доминировать над остальными и наоборот.

Второй фактор - погода. Мы пока не рассчитываем на то, что плащи и зонты будут храниться в нашем шкафу, так что дождь нам отслеживать не надо. Мы берем, так называемый,



коэффициент похожести по формуле:  $|1 - |\frac{a}{b} - 1||$ . Таким образом, чем этот коэффициент ближе к единице, тем числа “ближе”, то есть тем температура более подходящая для носки этой вещи. Идеальная температура для носки вещи - среднее арифметическое между максимальной и минимальной температурой. За счет такой формулы мы выберем вещь даже тогда, когда, вроде бы, подходящих вещей нет.

Финальный коэффициент соответствия рассчитывается для каждого стиля каждой вещи, как произведения процента стиля в вещи, процента этого стиля в пользовательском и коэффициент похожести температуры. Чем больше

коэффициент соответствия, тем больше данная вещь подходит сейчас. Далее мы сортируем список вещей от максимального к минимальному по стилю следующего события. То есть выше будут находиться наиболее рекомендованные вещи.

## **Модуль Purp\_Ward**

Для проекта мы реализовали собственную локальную библиотеку(модуль) на языке программирования Python. Это эффективное решение по оптимизации кода и сокращению его количества. Всего реализовано 14 функций:

Вспомогательные:

1. `is_part_in_list` - является ли слово или несколько слов частью строки
2. `identify` - определение стиля события по ключевым словам
3. `chunks` - разделение строки `s` на небольшие строки длиной `n` символов

И основные:

1. `choose_thing` - возвращает следующую вещь при выборе вещи
2. `get_index_from_shelf` - возвращает индекс вещи в базе данных по номеру полки, на которой она лежит
3. `get_name_from_index` - возвращает “имя” вещи по ее индексу в базе данных
4. `get_shtr_from_index` - возвращает цифры штрихкода вещи по ее индексу в базе данных
5. `take_thing` - функция, которая изменяет данные в файлах и передает информацию шкафу о том, какую вещь надо выдать пользователю, зная индекс вещи в базе данных или то, на какой полке она лежит
6. `read_style` - позволяет преобразовать строку с индексами в список упорядоченных процентов
7. `similarity_coeff` - функция, которая считает коэффициент похожести чисел `a` и `b`

8. `get_temp` - с помощью [API OWM](#) считывает и возвращает температуру в Москве сейчас в градусах Цельсия
9. `recommend` - функция, которая возвращает “таблицу” типа [pd.DataFrame](#) с отсортированными вещами, где чем “выше” вещь, тем более она рекомендована
10. `list_of_things` - тоже возвращает “таблицу” типа [pd.DataFrame](#), но уже просто со списком вещей, без сортировки
11. `style_of_next_event`-функция, которая считывает следующее события из Google Calendar и возвращает его стиль

### **Данные в системе**

Наша база данных - это файл csv, содержащий соответствующие колонки: штрихкод, “имя” вещи, ее тип, проценты стилей, размер, минимальную и максимальную температуру, номер полки, где она лежит и общее количество взятий. Это централизованная база данных, то есть ее использует как приложение, так и Telegram bot и голосовой помощник. Еще у нас есть файл `choose.txt`, с помощью которого мы передаем номера полок, с которых надо забрать вещи, и `style.txt`, с помощью которого приложение “узнает” стиль следующего события.

### **TCP-сервер**

Такое большое количество информации между устройствами необходимо как-то передавать, усложняет ситуацию еще и то, что все устройства имеют разную платформу, операционную систему и др. Серверу необходимо передавать данные между `arduino`, `raspberry` и `macbook`.

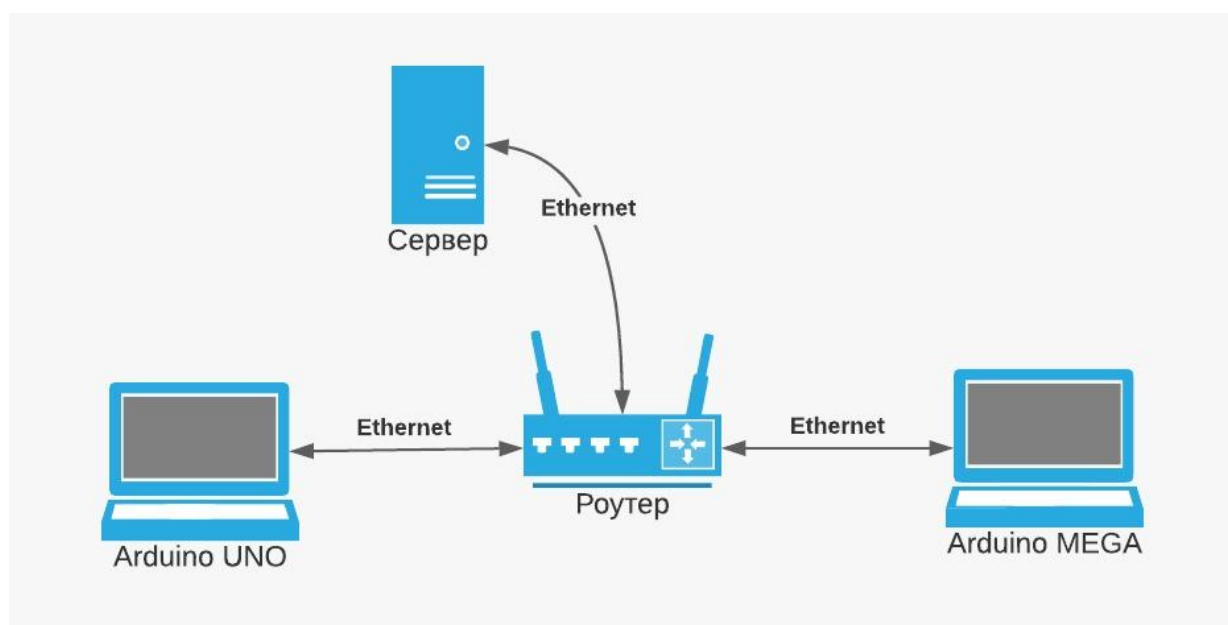
Изначально мы хотели использовать Bluetooth, но отказались от этой идеи из-за сложность настройки на разных устройствах, затем планировали настроить Wi-Fi сеть, чтобы в беспроводном режиме все устройства могли “общаться” эта идея нам понравилась больше, но возникла проблема того, что

на крупных мероприятиях Wi-Fi легко “теряется” в виду большого количества других сетей, поэтому было принято решение использовать соединение по Ethernet кабелю.

Для этого у нас в системе есть роутер с встроенным DHCP сервером и маршрутизатор для увеличения количества возможных подключаемых устройств.

Было принято решение поставить сервер на raspberry. Сервер написан на языке python, используя библиотеки socket и multiprocessing. Первая нужна для возможности создания сети устройств на нижнем уровне сети. Вторая была добавлена чтобы на каждое устройство выделить отдельный поток, а также чтобы сервер обрабатывал каждое устройство параллельно другим. Была разработана система специальных команд для общения устройств с сервером. Это было сделано для безопасности подключения устройств и для удобства работы с устройствами.

Еще одной немало важной частью является устойчивость системы, а именно сервера по обмену данными, он должен уметь правильно реагировать на всевозможные команды и данные и в случае ошибки отключать только “сломавшегося” клиента. Эта особенность была успешно реализована



посредством библиотеки multiprocessing, при экстренном завершении работы одного из процессов другие продолжают работать в штатном режиме.

### **Социальное взаимодействие**

Исходя из данных нашего социального опроса, каждый ? человек тратит на подбор одежды более 5 минут. С Purple Wardrobe эти каждые ? человека смогут экономить время, потому что сбор комплекта с нашим шкафом занимает менее 5 минут. Это сэкономленное время человек может потратить на что угодно: увлечения, отдых или уделить хотя бы немного больше времени семье, ведь в нынешних реалиях, где время, зачастую - самый недостающий ресурс, люди забывают даже о таких важных вещах.

Отдельно можно выделить пользу “умного” шкафа для людей с нарушениями опорно-двигательной системы, в том числе инвалидов и пожилых людей, ведь подбор одежды, её сортировка или даже достать что-либо с высокой полки - это практически непосильная задача для них.

Шкаф для людей с ограниченными возможностями плюс экономия самого ценного ресурса - времени

### **Экономическая составляющая**

На проект мы потратили 130 часов. Возьмем среднюю зарплату начинающего инженера/программиста - 600 рублей/час. Смета составила 35000 рублей. Таким образом финальная стоимость проекта -  $600 * 2 * 130 + 35000 = 191000$  рублей. Финальная стоимость продукта будет меньше, т.к. затраты на разработку будут “растянуты”.

Целевая аудитория Purple Wardrobe:

- 20-50 лет;

- Занятые люди, которые не хотят тратить время на одежду, но выглядеть стильно, в основном это мужчины;
- Доход более 100 000 рублей в месяц;
- Tech продвинутые пользователи;

Инвесторами данного проекта могут стать компании Smartsy и UltraTrade, занимающиеся производством “умной” мебели.

При разработке реального прототипа Purple Wardrobe структура затрат будет включать в себя затраты на:

- закупку оборудования
- закупку материалов и электрических компонентов
- изготовление специфических деталей
- разработку программного обеспечения
- сборку и отладку прототипов

Предлагая свою идею рынку мы могли получать доход от разовых сделок с клиентами, а также заключая контракты с производителями систем для “Умного дома”.

Ключевые ресурсы для работы над нашим прототипом это:

- бюджет для разработки стартового прототипа
- рабочий персонал из 4 человек
- рабочее помещение
- расходные материалы
- программное обеспечение

|  |  |  |  |  |
|--|--|--|--|--|
| <p><b>Партнёры</b></p> <p>Инвесторы: Smartsy, UiltraTrade</p>  | <p><b>Процессы</b></p> <p>Производство<br/>Продажа</p> | <p><b>Уникальное ценностное предложение</b></p> <p>Качественный скоростной помощник, не имеющий аналогов<br/>Может быть интегрирован в систему умного дома</p>   | <p><b>Отношения с клиентами</b></p> <p>Реклама в СМИ, поисковых сервисах<br/>Выступления на отраслевых площадках<br/>Сайт продукта</p> | <p><b>Потребители</b></p> <p>Целевая аудитория - люди от 35 до 60 лет с доходом выше среднего.</p> |
| <p><b>Структура затрат</b></p> <p>Закупка оборудования<br/>Закупка материалов и электрических компонентов<br/>Изготовление специфических деталей<br/>Разработка программного обеспечения</p> |  | <p><b>Каналы</b></p> <p>Прямые: сайт продукта.<br/>Непрямые: реализация в системе "умный дом"</p> <p><b>Потоки доходов</b></p> <p>Розничная торговля через собственные каналы сбыта<br/>Заключение контрактов с производителями систем "умный дом"</p> |  |  |

Таблица 1 - Бизнес-модель проекта Purple Wardrobe.

Для реализации нашей идеи мы выбрали стратегию Lean Startup, то есть избегая лишних затрат мы будем постоянно “обучаться”, понимая потребности клиентов, совершенствовать продукт, нивелировать путями продаж на рынке. Такая стратегия будет эффективна до формирования устойчивой репутации и рынка сбыта.

Шаги реализации:

1. Создать MVP (Minimal Viable Product (минимально жизнеспособный продукт) — тестовая версия товара, услуги или сервиса с минимальным набором функций (иногда даже одной), которая несет ценность для конечного потребителя)
  - a. Провести SWOT-анализ (анализ сильных и слабых сторон проекта, возможностей и угроз).
  - b. Создать карту пути пользователя (что делает пользователь при взаимодействии с продуктом). Это поможет определить требования к дизайну и интерфейсу.
  - c. Сделать промышленный дизайн, конструкторскую документацию.
  - d. Провести сборку и отладку прототипа.
  - e. Провести тестирование прототипа.
  - f. Создать документацию для серийного производства.
2. Произвести первую тестовую партию продукции.
3. Выбрать стратегию маркетинга.
4. Попробовать продать первую партию продукции.
5. Провести анализ продаж, отзывов пользователей.
6. Начать поиск инвесторов.

Данные шаги являются одним циклом реализации стратегии Lean Startup.

### **Перспективы развития**

В будущем мы планируем:

- Доработать дизайн шкафа;
- Добавить блок с вешалками;
- Встроить экран с приложением прямо в шкаф для взаимодействия с пользователем;
- Добавить умную камеру с виртуальной примерочной;

## **Развитие с прошлого года**

Как мы развились с прошлого года?

- Более сложная система;
- Использовали методы по обработке изображений на порядок сложнее прошлогодних, в том числе динамическую обработку кадров;
- Написали приложение на QT, до этого был tkinter, т.е. появился гораздо более широкий функционал;
- Использовали API;
- Теперь используется единая база данных в csv файле, а не большое количество txt-файлов;
- В данном проекте большое количество движущихся механических составляющих по сравнению с прошлым проектом.