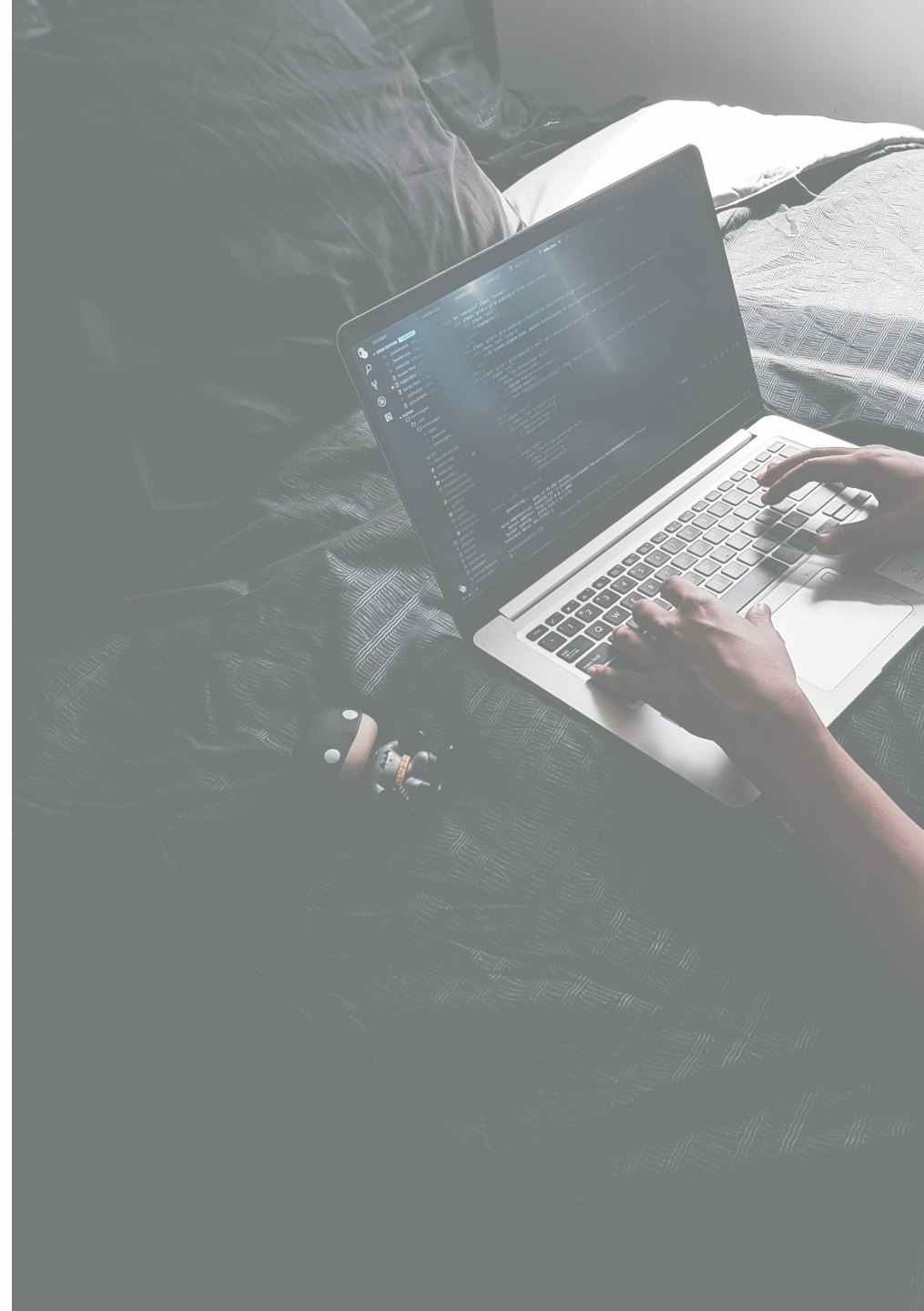




ЗАИНТЕРЕСОВАННЫЕ ЛИЦА И УЧАСТНИКИ ПРОЕКТА



НА ЭТОМ УРОКЕ

- Классифицируем типы и роли заинтересованных лиц.
- Разберёмся с классами и ролями пользователей в проекте.
- Поговорим о том, как выявлять заинтересованных лиц.
- Изучим подход к сбору требований через варианты использования (use case).

ЗАИНТЕРЕСОВАННЫЕ ЛИЦА

Лица или организации, имеющие права, долю, требования или интересы относительно системы или её свойств, удовлетворяющих их потребностям и ожиданиям (ISO/IEC/IEEE 15288:2015).

ТИПЫ ЗАИНТЕРЕСОВАННЫХ ЛИЦ

- Приобретающая сторона.
- Заказчик.
- Разработчик.
- Поставщик.
- Пользователь.
- Производитель.
- Сопровождающая сторона.
- Ликвидатор.
- Аккредитор / Инспектор.
- Регулирующий орган.
- Остальные.

ПРОЕКТНАЯ КОМАНДА

Code File Edit Selection View Go Debug Terminal Window Help

```
APIBitcoin-ticker/index.js -- Web-dev2019
index.js index.html

app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html');
});

app.post('/', (req, res) => {
  const crypto = req.body.crypto;
  19 const fiat = req.body.fiat;

  const baseUrl = "https://api2.bitcoinaverage.com/indices/global/ticker/";
  const finalURL = baseUrl + crypto + fiat;
  request(finalURL, (error, response, body) => {
    const data = JSON.parse(body);
    const price = data.last;

    const currentDate = data.display_timestamp;
    res.write("<p>The current date is " + currentDate + "</p>");
    res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");
    res.send("<h1>The current price of bitcoin is " + price + "</h1>");
  });
});
```


ПРОЕКТНАЯ КОМАНДА

Это коллектив специалистов, объединённых для достижения общих целей и решения поставленных перед ними задач в течение жизненного цикла проекта.

Каждый включённый в команду специалист обладает специфическими компетенциями и выполняет определённые функции.

ЦЕЛИ ПРОЕКТНОЙ КОМАНДЫ

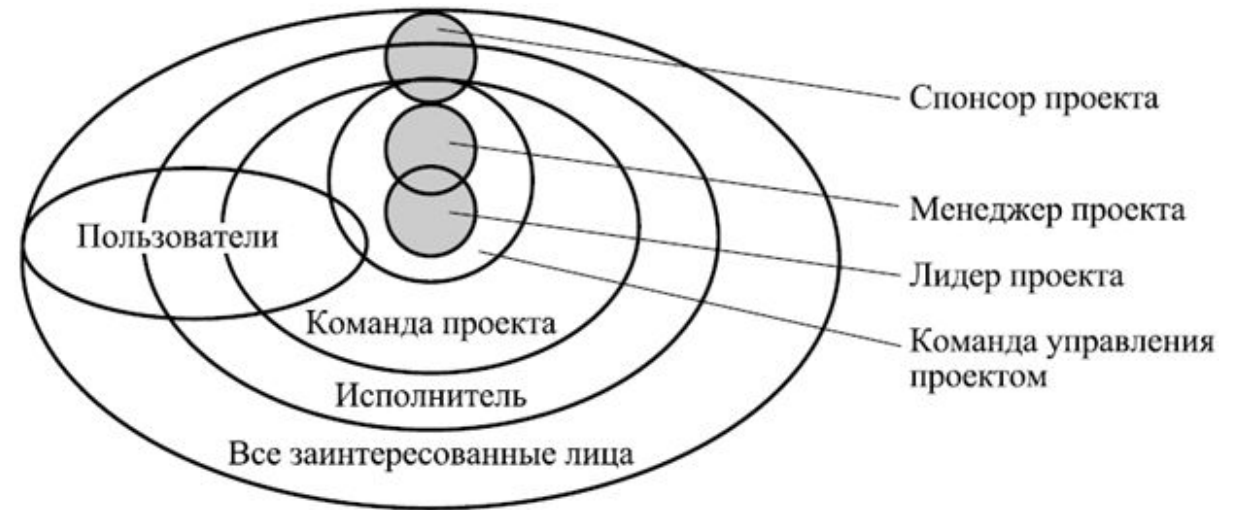
Идентифицировать заинтересованные стороны, определить их требования и ожидания, до возможной степени управлять их влиянием на проект для обеспечения его успешности.



ВЗАИМООТНОШЕНИЯ МЕЖДУ ЗАИНТЕРЕСОВАННЫМИ ЛИЦАМИ ПРОЕКТА



- Спонсор проекта.
- Менеджер проекта.
- Лидер проекта.
- Заказчик.
- Пользователь.
- Организация-исполнитель.
- Команда проекта.
- Команда управления проектом.
- Источники влияния.



КЛАССЫ ПОЛЬЗОВАТЕЛЕЙ

Code File Edit Selection View Go Debug Terminal Window Help

APItoBitcoin-Ticker/index.js — Web-dev2019

index.js index.html

app.get('/', (req, res) => {
 res.sendFile(__dirname + '/index.html');
});

app.post('/', (req, res) => {
 const crypto = req.body.crypto;
 19 const fiat = req.body.fiat;

const baseURL = "https://api2.bitcoinaverage.com/indices/global/ticker/";
const finalURL = baseURL + crypto + fiat;
request(finalURL, (error, response, body) => {
 const data = JSON.parse(body);
 const price = data.last;

const currentDate = data.display_timestamp;
res.write("<p>The current date is " + currentDate + "</p>");
res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");
res.send("<h1>The current price of bitcoin is " + price + "</h1>");
res.send();
});

};



ПОЛЬЗОВАТЕЛЬ

лицо или организация, которая использует действующую систему для выполнения конкретной функции.

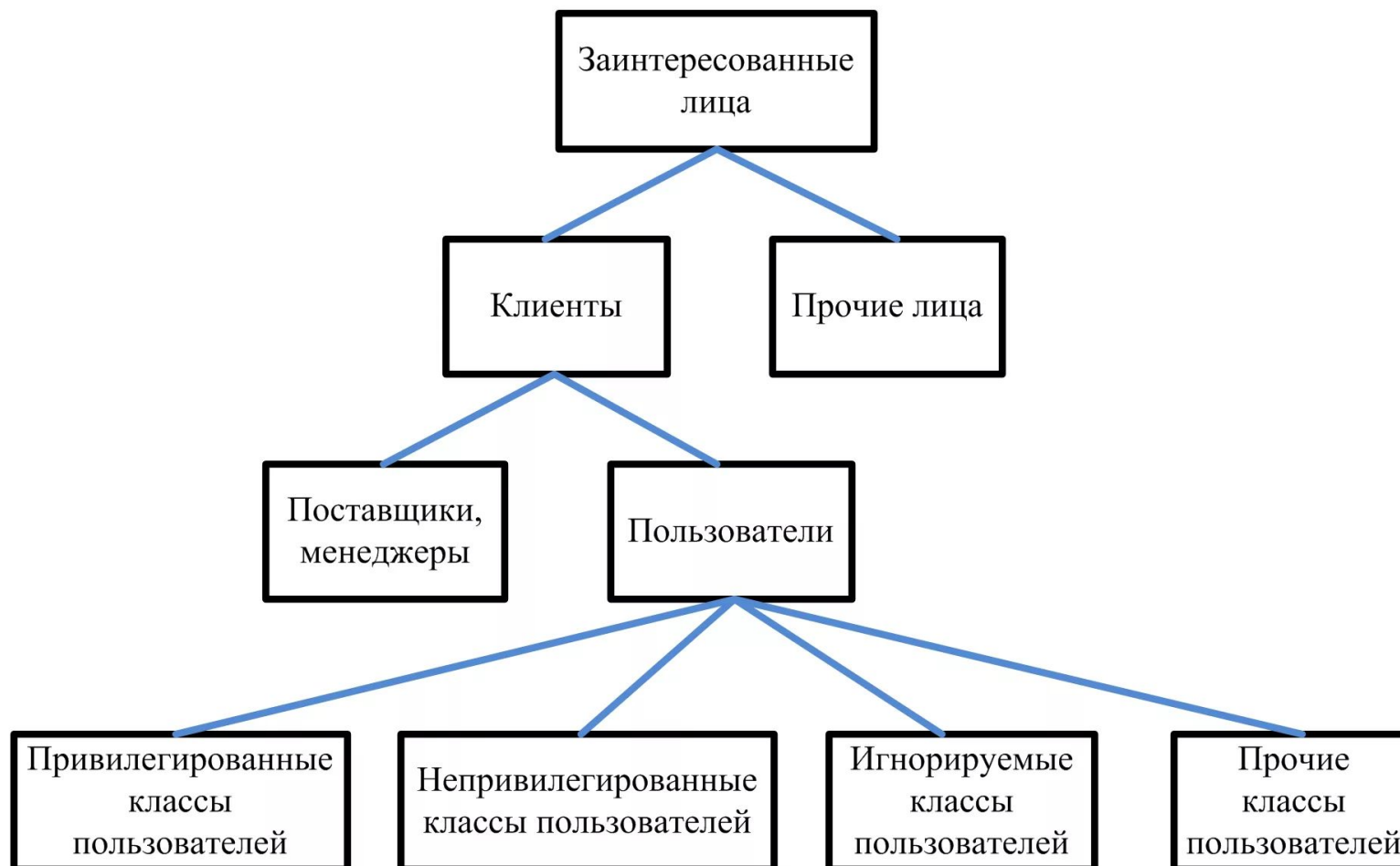


ПРИЗНАКИ КЛАССОВ ПОЛЬЗОВАТЕЛЕЙ ПРОДУКТА

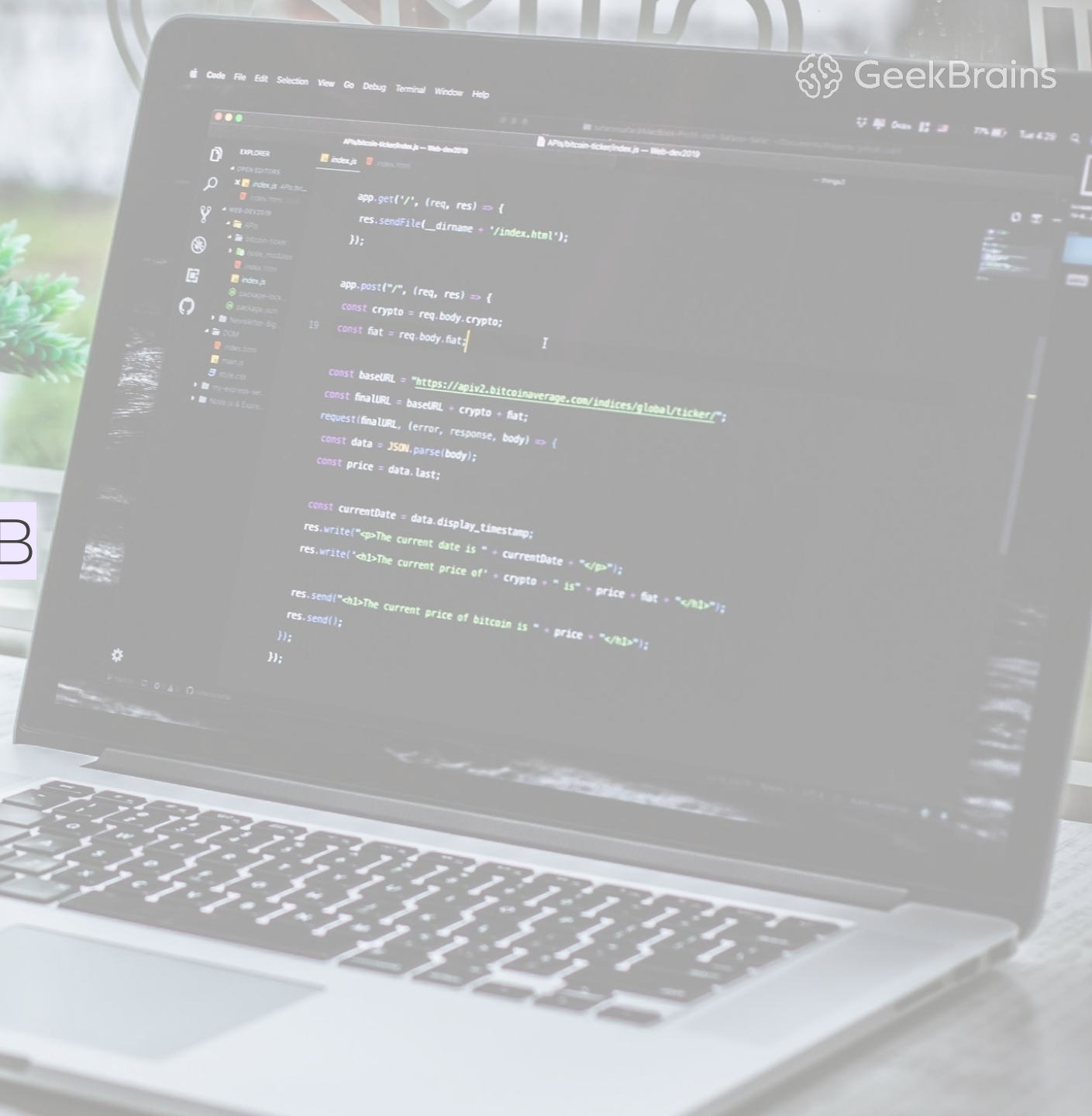
- Частота использования продукта.
- Опыт в предметной области и опыт работы с компьютерными системами.
- Требуемая им функциональность.
- Задачи, которые им приходится выполнять.
- Права доступа к системе. Например, «обычный пользователь», «гость» или «администратор».



ИЕРАРХИЯ ЗАИНТЕРЕСОВАННЫХ ЛИЦ И ПОЛЬЗОВАТЕЛЕЙ



ВЫЯВЛЕНИЕ КЛАССОВ ПОЛЬЗОВАТЕЛЕЙ



ВЫЯВЛЕНИЕ КЛАССОВ ПОЛЬЗОВАТЕЛЕЙ

- Кто пользователи системы?
- Кто заказчик (экономический покупатель) системы?
- На кого ещё окажут влияние результаты работы системы?
- Кто будет оценивать и принимать систему, когда она будет представлена и развёрнута?
- Существуют ли другие внутренние или внешние пользователи системы, чьи потребности необходимо учесть?
- Кто будет заниматься сопровождением новой системы?
- Не забыли ли мы кого-нибудь?

СБОР ТРЕБОВАНИЙ

ПОДХОД С ПРИМЕНЕНИЕМ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Code File Edit Selection View Go Debug Terminal Window Help

APIS/bitcoin-ticker/index.js — Web-dev2019

index.js index.html

EXPLORER

OPEN EDITORS

index.js APIS/bit...

index.html

WEB-DEV2019

APIS

bitcoin-ticker

index-modules

index.html

index.js

package-lock...

package.json

newsletter-sip

2020

index.html

main.js

live.css

my-express-ser...

node.js & Expre...

19

const baseURL = "https://api2.bitcoinaverage.com/indices/global/ticker/";

const finalURL = baseURL + crypto + fiat;

request(finalURL, (error, response, body) => {

const data = JSON.parse(body);

const price = data.last;

const currentDate = data.display_timestamp;

res.write("<p>The current date is " + currentDate + "</p>");

res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");

res.send("<h1>The current price of bitcoin is " + price + "</h1>");



ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ



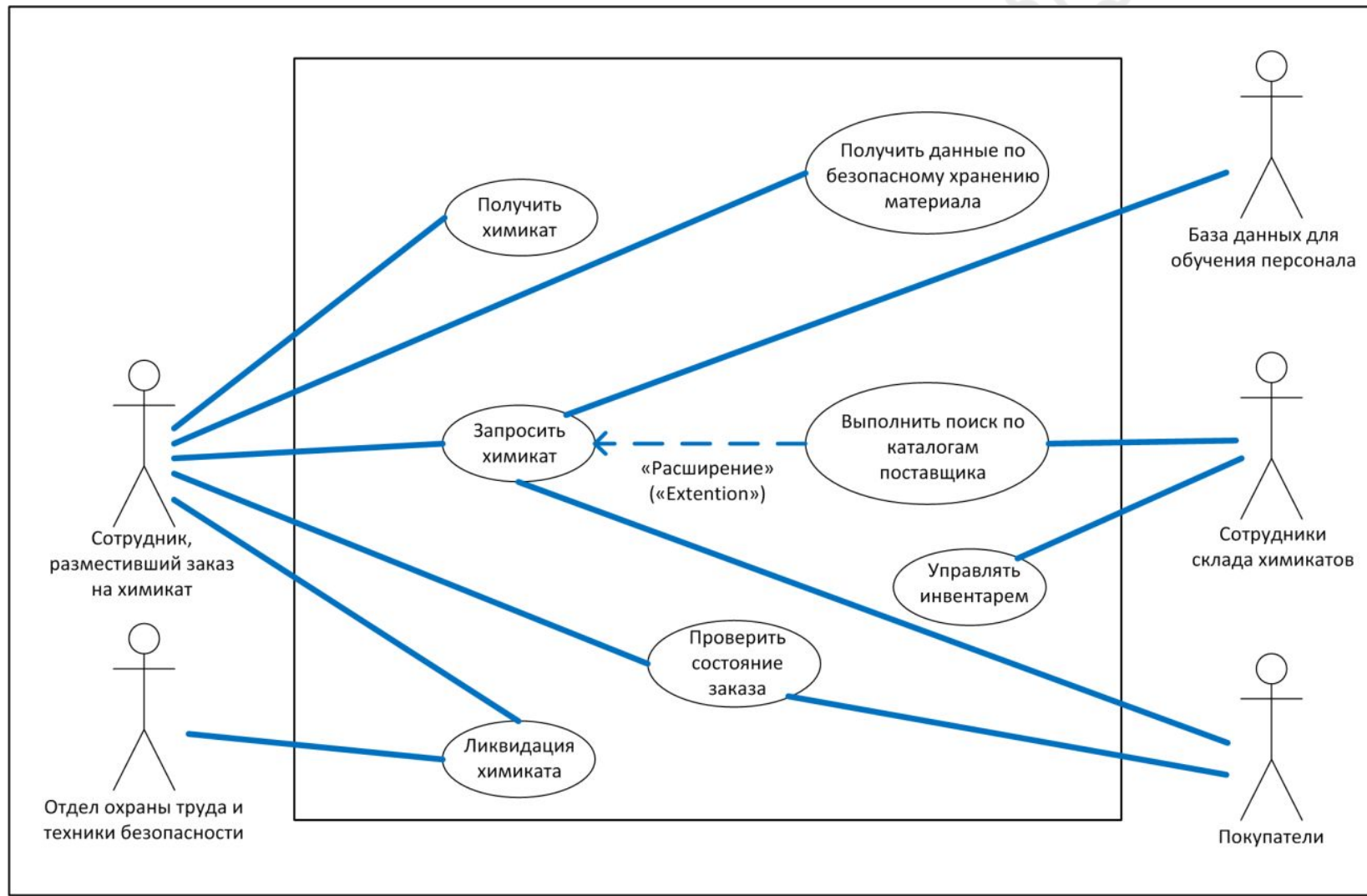
Вариант использования (use case) продукта описывает последовательность взаимодействия системы и внешнего действующего лица.



Действующее лицо (actor) — это человек, другая система ПО или аппаратное устройство, взаимодействующее с системой для достижения некой цели (Cockburn, 2001).

Роль пользователя — роль, которую члены одного или нескольких классов пользователей могут выполнять по отношению к системе (Constantine & Lockwood, 1999).




ПРИМЕР ДИАГРАММЫ USE CASE





ЭЛЕМЕНТЫ ОПИСАНИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



- Уникальный идентификатор.
- Имя.
- Краткое текстовое описание.
- Список предварительных условий
- Выходные условия.
- Пронумерованный список действий.

НАПРАВЛЕНИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Code File Edit Selection View Go Debug Terminal Window Help

APIS/bitcoin-ticker/index.js — Web-dev2019

index.js index.html

```
app.get('/', (req, res) => {  
  res.sendFile(__dirname + '/index.html');  
});
```

```
app.post('/', (req, res) => {  
  const crypto = req.body.crypto;  
  const fiat = req.body.fiat;
```

```
  const baseURL = "https://api2.bitcoinaverage.com/indices/global/ticker/";  
  const finalURL = baseURL + crypto + fiat;  
  request(finalURL, (error, response, body) => {  
    const data = JSON.parse(body);  
    const price = data.last;
```

```
    const currentDate = data.display_timestamp;  
    res.write("<p>The current date is " + currentDate + "</p>");  
    res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");  
    res.send("<h1>The current price of bitcoin is " + price + "</h1>");  
  });  
});
```

```
const baseURL = "https://api2.bitcoinaverage.com/indices/global/ticker/";  
const finalURL = baseURL + crypto + fiat;  
request(finalURL, (error, response, body) => {  
  const data = JSON.parse(body);  
  const price = data.last;
```

```
  const currentDate = data.display_timestamp;  
  res.write("<p>The current date is " + currentDate + "</p>");  
  res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");  
  res.send("<h1>The current price of bitcoin is " + price + "</h1>");  
});  
});
```

```
const currentDate = data.display_timestamp;  
res.write("<p>The current date is " + currentDate + "</p>");  
res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");  
res.send("<h1>The current price of bitcoin is " + price + "</h1>");  
});  
});
```

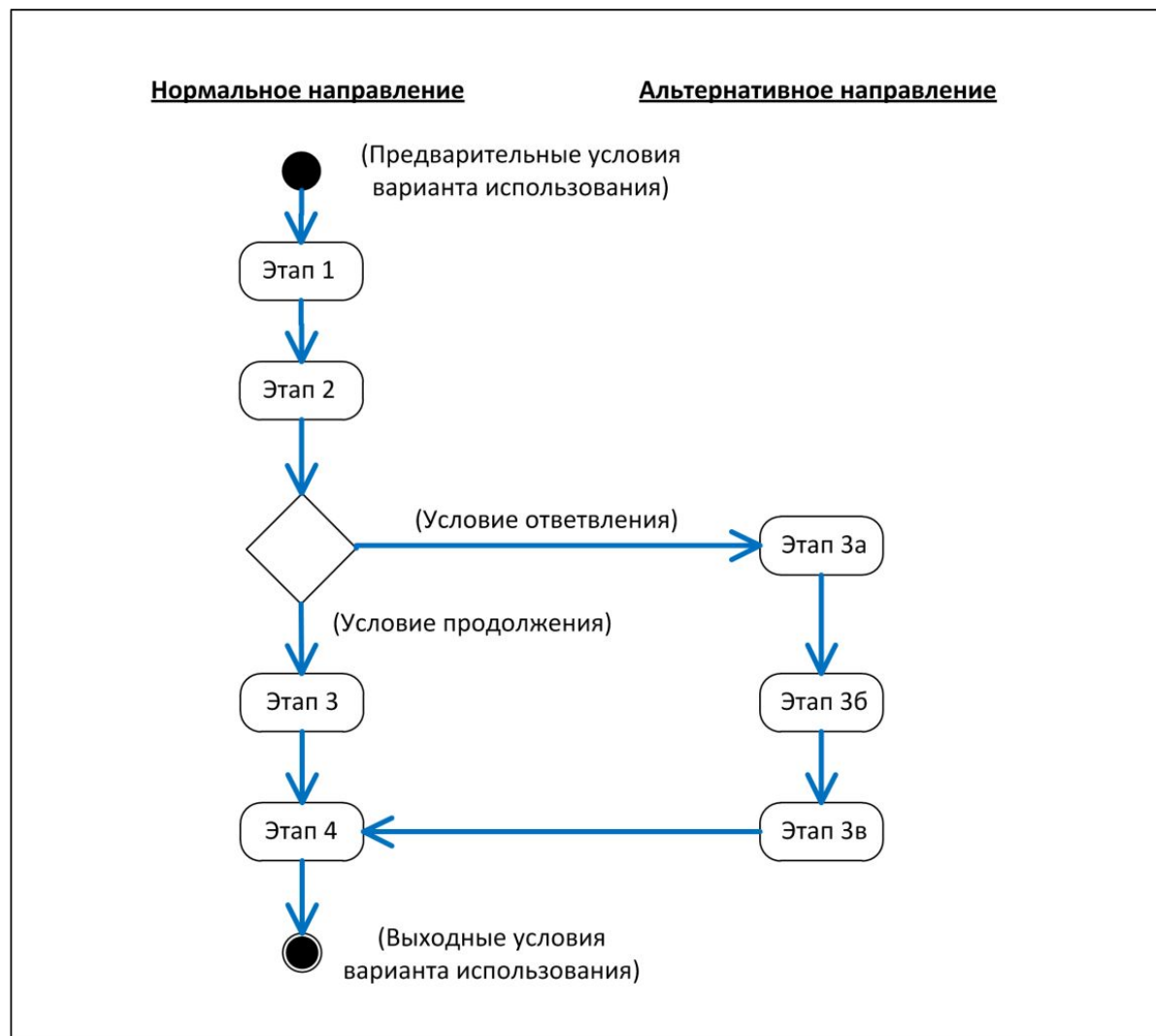
```
const currentDate = data.display_timestamp;  
res.write("<p>The current date is " + currentDate + "</p>");  
res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");  
res.send("<h1>The current price of bitcoin is " + price + "</h1>");  
});  
});
```

```
const currentDate = data.display_timestamp;  
res.write("<p>The current date is " + currentDate + "</p>");  
res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");  
res.send("<h1>The current price of bitcoin is " + price + "</h1>");  
});  
});
```

```
const currentDate = data.display_timestamp;  
res.write("<p>The current date is " + currentDate + "</p>");  
res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");  
res.send("<h1>The current price of bitcoin is " + price + "</h1>");  
});  
});
```

```
const currentDate = data.display_timestamp;  
res.write("<p>The current date is " + currentDate + "</p>");  
res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");  
res.send("<h1>The current price of bitcoin is " + price + "</h1>");  
});  
});
```

НОРМАЛЬНОЕ И АЛЬТЕРНАТИВНОЕ НАПРАВЛЕНИЯ





ИСКЛЮЧЕНИЯ




Исключения (exceptions) — это условия, препятствующие успешному завершению задания.

Для варианта использования «Запросить химикат» существует одно исключение — «Химиката нет в продаже».

Если в процессе сбора информации не указать, как обрабатывать исключение, то возможны два пути:

1. Разработчики предложат лучший, по их мнению, способ обработки исключений.
2. При генерации пользователем неверного условия произойдёт сбой системы, так как никто не предусмотрел такой ситуации.



ВЫЯВЛЕНИЕ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



- Определить действующие лица и бизнес-процессы, в которых каждое лицо участвует.
- Выразить бизнес-процессы в терминах определённых сценариев, обобщить сценарии в варианты использования и определить действующие лица для каждого варианта.
- Определить внешние события, на которые система должна реагировать, соотнести эти события с действующими лицами и определёнными вариантами использования.
- Определить вероятные варианты использования на основе функциональных требований.
- Если какие-либо требования невозможно проследить до какого-либо варианта использования, определить, нужны ли они.

ДОКУМЕНТИРОВАНИЕ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

| | |
|---------------------------------------|---|
| Идентификатор варианта использования: | Вариант использования 1 |
| Название варианта использования: | Запросить химикат |
| Действующее лицо | Сотрудник, разместивший заказ на химикат |
| Описание | Сотрудник, разместивший заказ на химикат, указывает в запросе необходимый химикат, вводя его название или номер идентификатора химиката или импортируя его структуру из соответствующего графического средства. Система выполняет запрос, предлагая новый или использованный контейнер с химикатом со склада или позволяя создать запрос на заказ у стороннего поставщика |
| Предварительные условия | <ol style="list-style-type: none"> 1. Личность пользователя аутентифицирована. 2. Пользователь имеет право запрашивать химикаты. 3. База данных ПО запасам химикатов в данный момент подключена |
| Выходные условия | <ol style="list-style-type: none"> 1. Запрос сохраняется в Chemical Tracking System. 2. Запрос отправлен по электронной почте на склад химикатов или поставщику |

ПРЕИМУЩЕСТВА И ОШИБКИ

Code File Edit Selection View Go Debug Terminal Window Help

APIBitcoin-ticker/index.js — Web-dev2019


index.js index.html

app.get('/', (req, res) => {
 res.sendFile(__dirname + '/index.html');
});

app.post('/', (req, res) => {
 const crypto = req.body.crypto;
 19 const fiat = req.body.fiat;

const baseURL = "https://api2.bitcoinaverage.com/indices/global/ticker/";
const finalURL = baseURL + crypto + fiat;
request(finalURL, (error, response, body) => {
 const data = JSON.parse(body);
 const price = data.last;

const currentDate = data.display_timestamp;
res.write("<p>The current date is " + currentDate + "</p>");
res.write("<h1>The current price of " + crypto + " is " + price + fiat + "</h1>");
res.send("<h1>The current price of bitcoin is " + price + "</h1>");
res.send();
});



ПРЕИМУЩЕСТВА ПРИМЕНЕНИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



Облегчают расстановку приоритетов:

- Варианты использования описывают один из основных бизнес-процессов, активизируемых системой.
 - Многие пользователи часто обращаются к ним.
 - Их запросил привилегированный класс пользователей.
 - Они предоставляют возможности, необходимые для соответствия требованиям.
 - Функции других систем зависят от их наличия.
- .



ОШИБКИ ПРИ СОСТАВЛЕНИИ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



- Слишком много вариантов использования.
- Очень сложные варианты использования.
- Включение пользовательского интерфейса в варианты использования.
- Включение определения данных в варианты использования.
- Варианты использования, которые непонятны пользователям.
- Новые бизнес-процессы.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ





ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Вспомним задание предыдущего урока.

Нам понадобится список заинтересованных лиц и бизнес-требований.

1. Классифицируйте типы и роли заинтересованных лиц. Выделите классы пользователей относительно списка требований.
2. Опишите основной вариант использования. Нарисуйте диаграмму вариантов использования. Выделите нормальное и альтернативное направление, найдите исключение.



ПРАКТИЧЕСКОЕ ЗАДАНИЕ *

1. Опишите ещё один основной вариант использования к предыдущему кейсу.
2. Выделите нормальное направление, два альтернативных направления и два исключения.
3. Выделите важнейший вариант использования.

+ заполнение шаблона!

ВАШИ ВОПРОСЫ