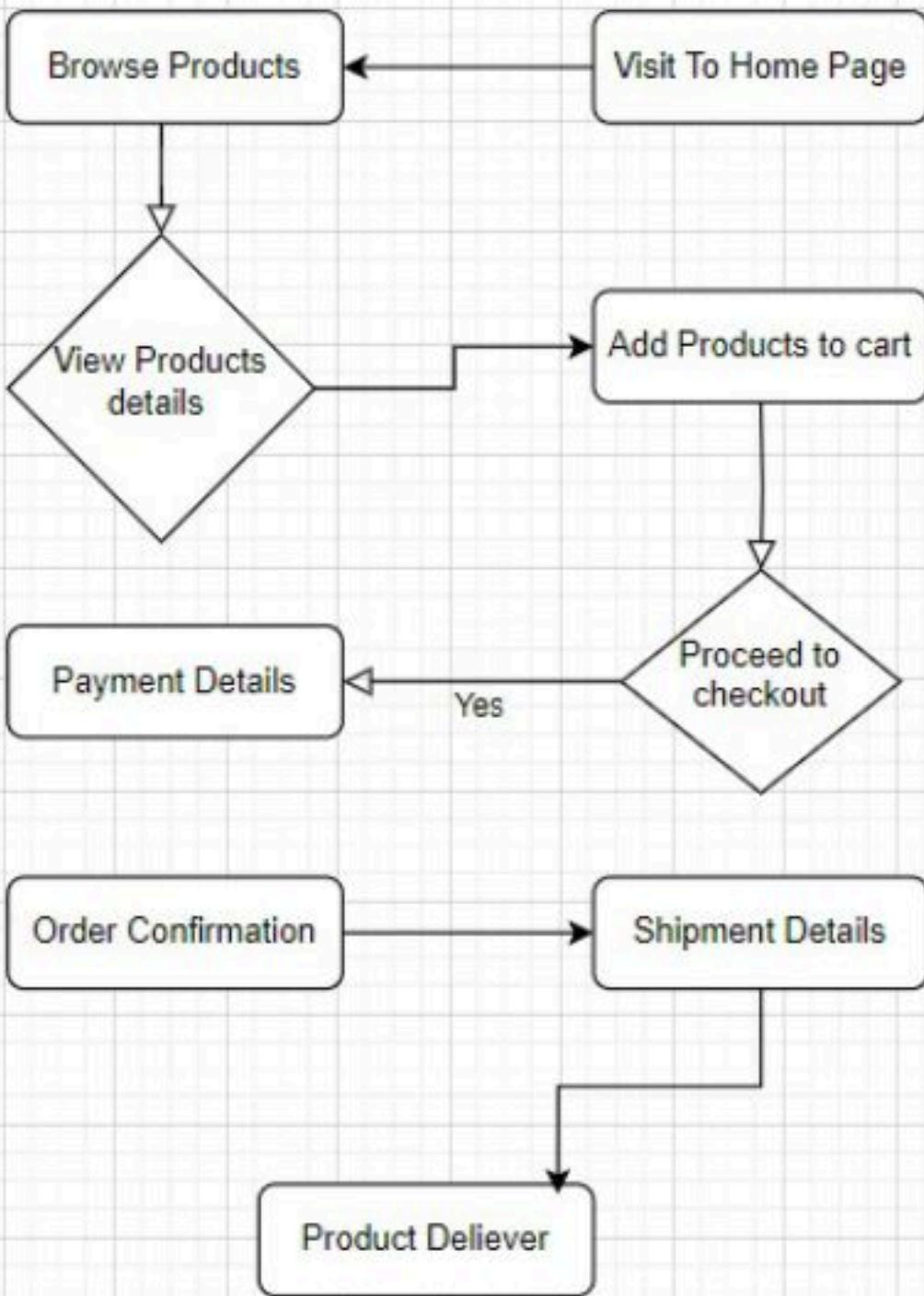
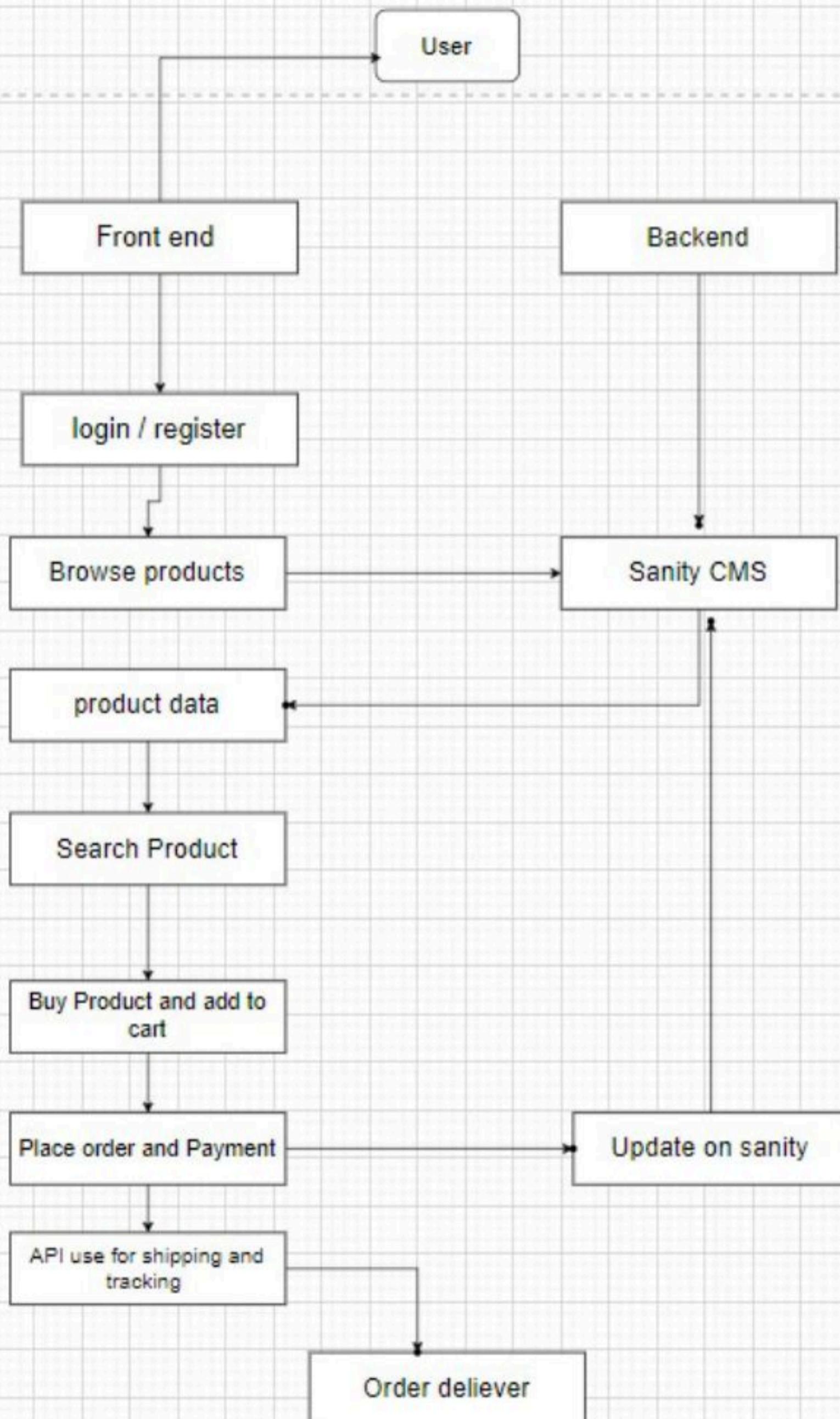


KEY WORKFLOW



DESIGN STRUCTURE ARCHITECTURE



Planning THE TECHNICAL FOUNDATION

TECHNICAL REQUIREMENTS

A user friendly design of website with responsiveness for all the devices

-Frontend → where user can select the Requirements product according to their choice and place an order.


Sanity cms handle the backend of my website and manage all

sanity cms → products details and records as Backend
Sanity act as a database in my e-commerce website. It handle products through Schemas

Third party API provide data for front-end and It also

Third party → provide shipping, tracking and payment gateways for backend services

DESIGN STRUCTURE ARCHITECTURE



PLAN API REQUIREMENTS

Endpoint	Methods	Description	Parameters	Response Example
/api/product	GET	fetch all the product.	None	{ "id":1, "name": "shoes", "p
/api/product/:id	GET	fetch a single product	id (path)	{ "id":1, "name": "Nike" }
/api/product	POST	Add new item	name, price, category (body)	{ "id":2, success: true }
/api/product/:id	PUT	Update a feed	id (path), name, price (body)	{ "success":true }
/api/product/:id	DELETE	Delete a feed	id (path)	{ "success":true }

SANITY SCHEMA

```
export default defineType =>
```

```
  name: 'product',
```

```
  type: 'document',
```

```
  title: 'Product',
```

```
  fields: [
```

```
    {
```

```
      name: 'id',
```

```
      type: 'string',
```

```
      title: 'ProductId',
```

```
    },
```

```
    { name: 'name',
```

```
      type: 'string',
```

```
      title: 'ProductName',
```

```
  },
```

```
  { name: 'slug',
```

```
    type: 'slug',
```

```
    title: 'Slug',
```

```
    description: 'URL-friendly',
```

```
  Options: {
```

```
    source: 'name', maxLength: 200
```

```
  }, Validation (rule) =>
```

```
    Rule.required().error('Slug is required for product')
```

```
},
```

5
name: 'price'

type: 'number'

title: 'Product Price'

6,

name: 'rating',

type: 'number',

title: 'Rating',

description: 'Average rating of product,'

7,

{

name: 'tags',

type: 'array',

title: 'Tags'

of: [{type: 'string'}],

options: {

layout: 'tags',

},

description: 'Add tags such as "Clothes",

8

)