Core idea:
Intuitively separate type definitions from type declarations

Definition of MyType

```
// Definition

typedef struct {
    int count;
    char *array;
} MyType;

// other definition syntax
struct MyType {
    int count;
    char *array;
};
```

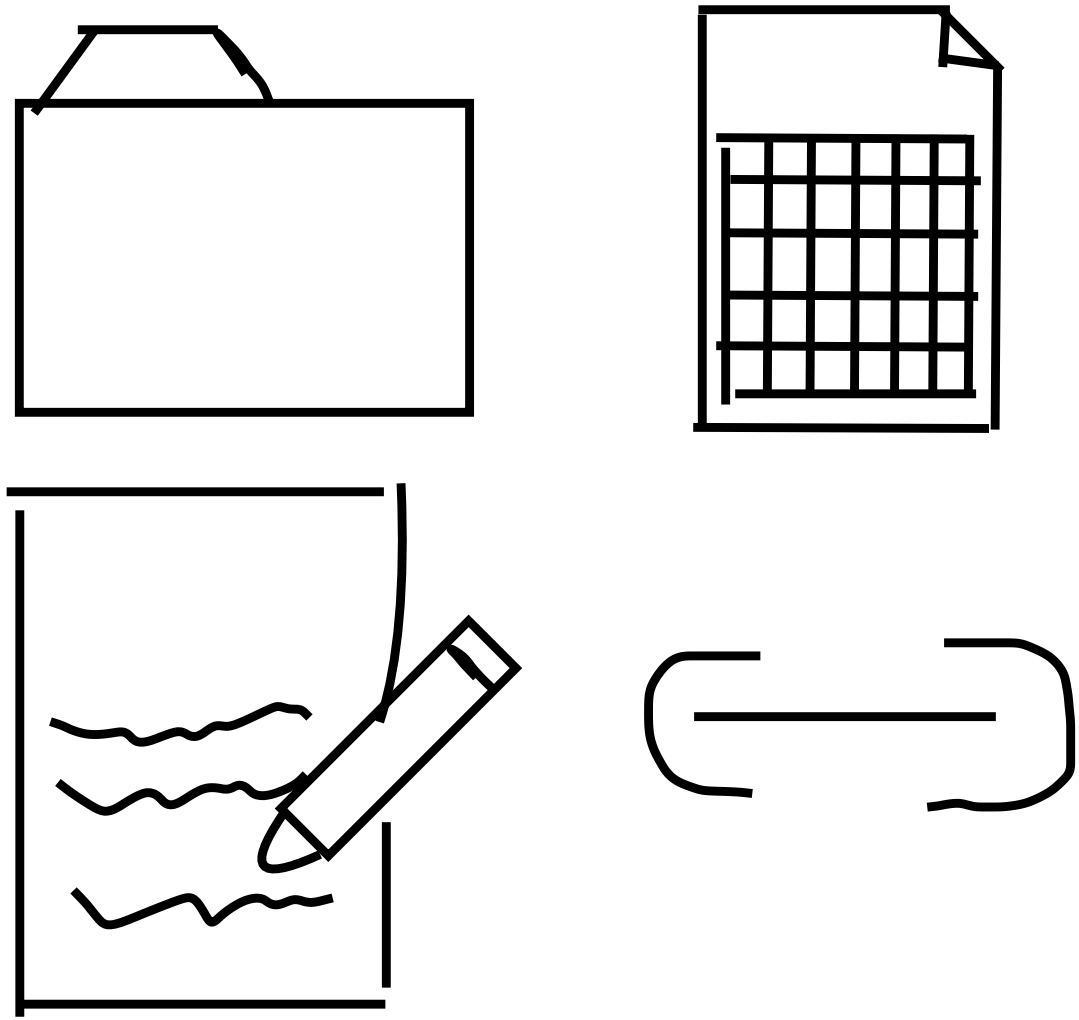
Declaration of MyType

```
// Declaration

struct SomeTypedef {
    MyType declaration_of_mytype; // declaration occurs here
    int count;
};

// Anonymous Declaration
// a declaration with an inline definition
struct SomeDefinition {
    struct {
        int count;
        int *array;
    } declaration_of_anon_type;
};
```

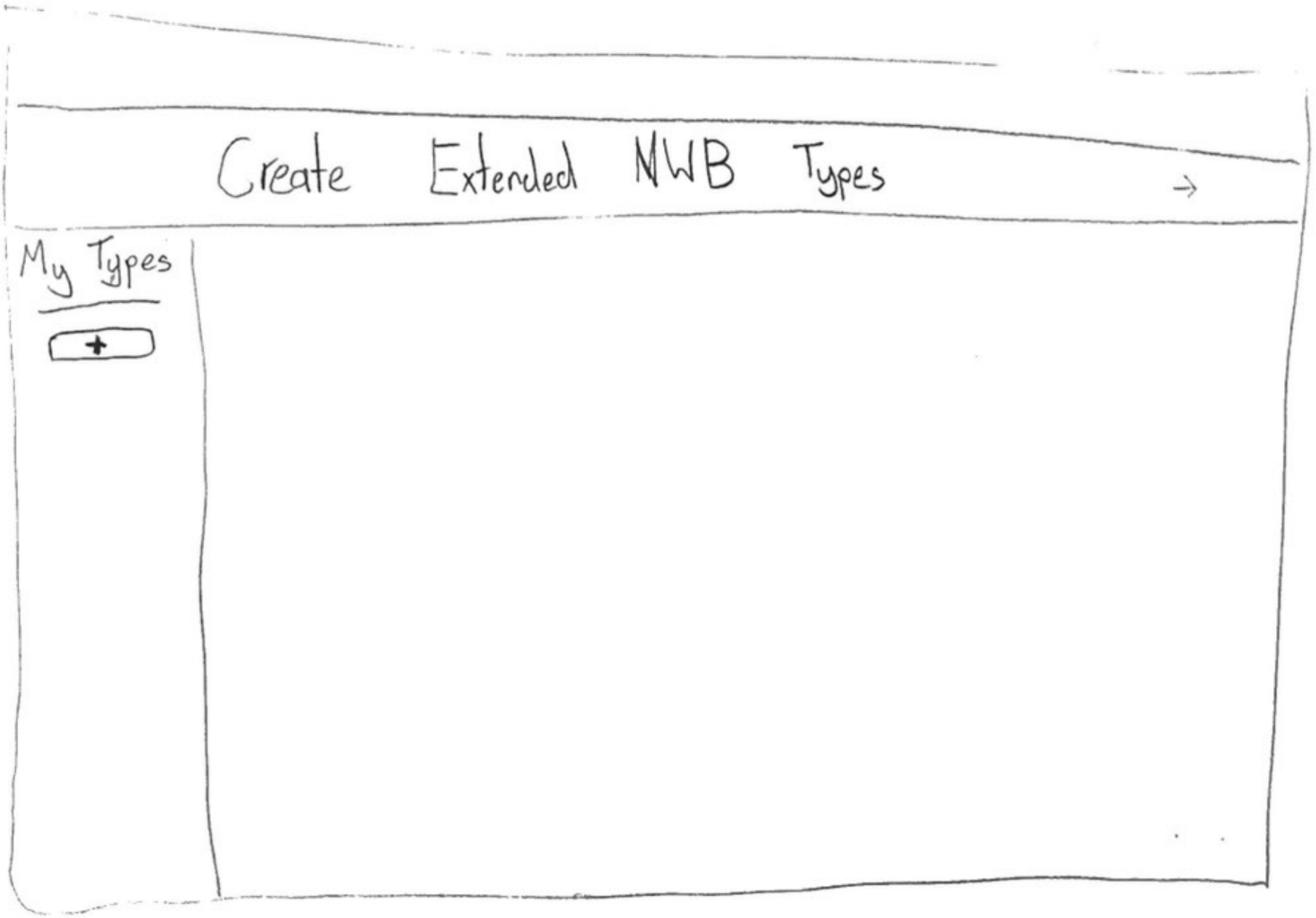
Quick terminology

Kind refers to dataset, group, attribute or link

In the GUI kinds use icons instead of full name.
Hover over an icon it's full name.

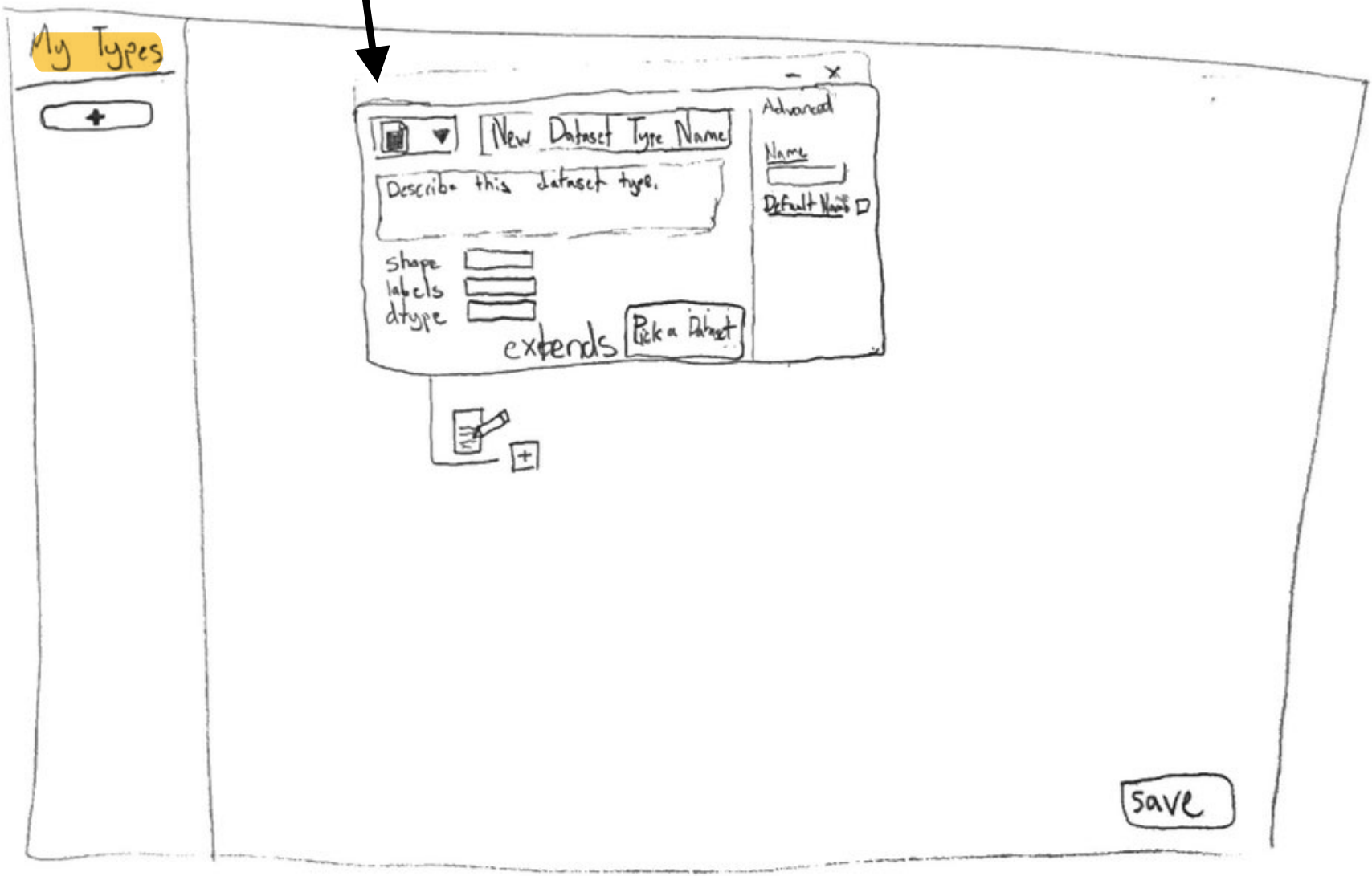


Simplified Landing Page
Note: Only one active button (the add type button)



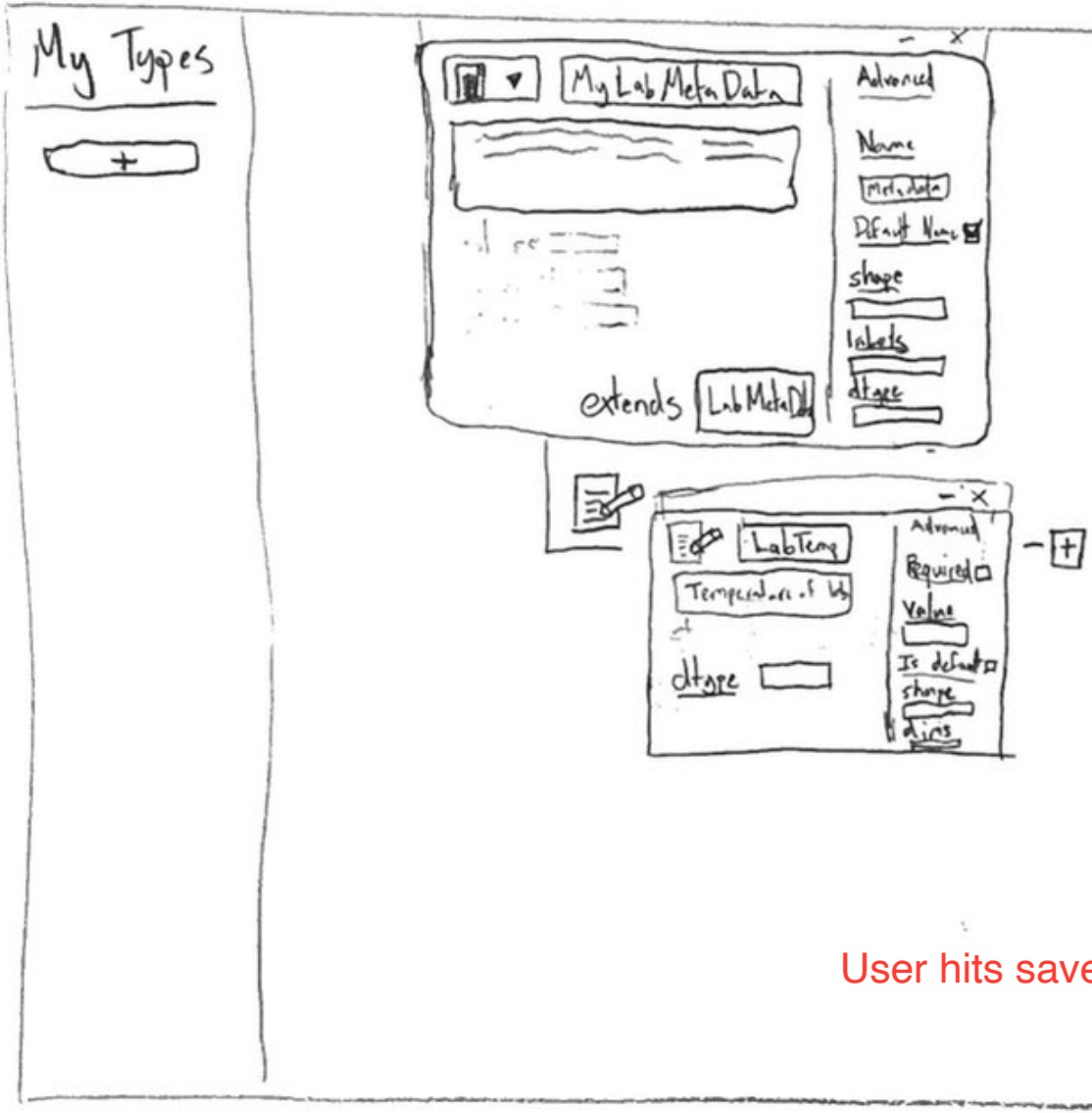
After pressing add,
user hits the top left icon dropdown to select dataset.

App presents user with
DatasetTypeDef options



User fills in DatasetTypeDef options
And hits add note (attribute) to create an attribute declaration.

The type of the attribute declaration is implicit (anonymous).



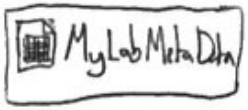
User adds more attributes
with plus button.
The full shape of the dataset
typedef is always visible.

User hits save when ready for validation

Save

Create NWB Extended Types

My Types

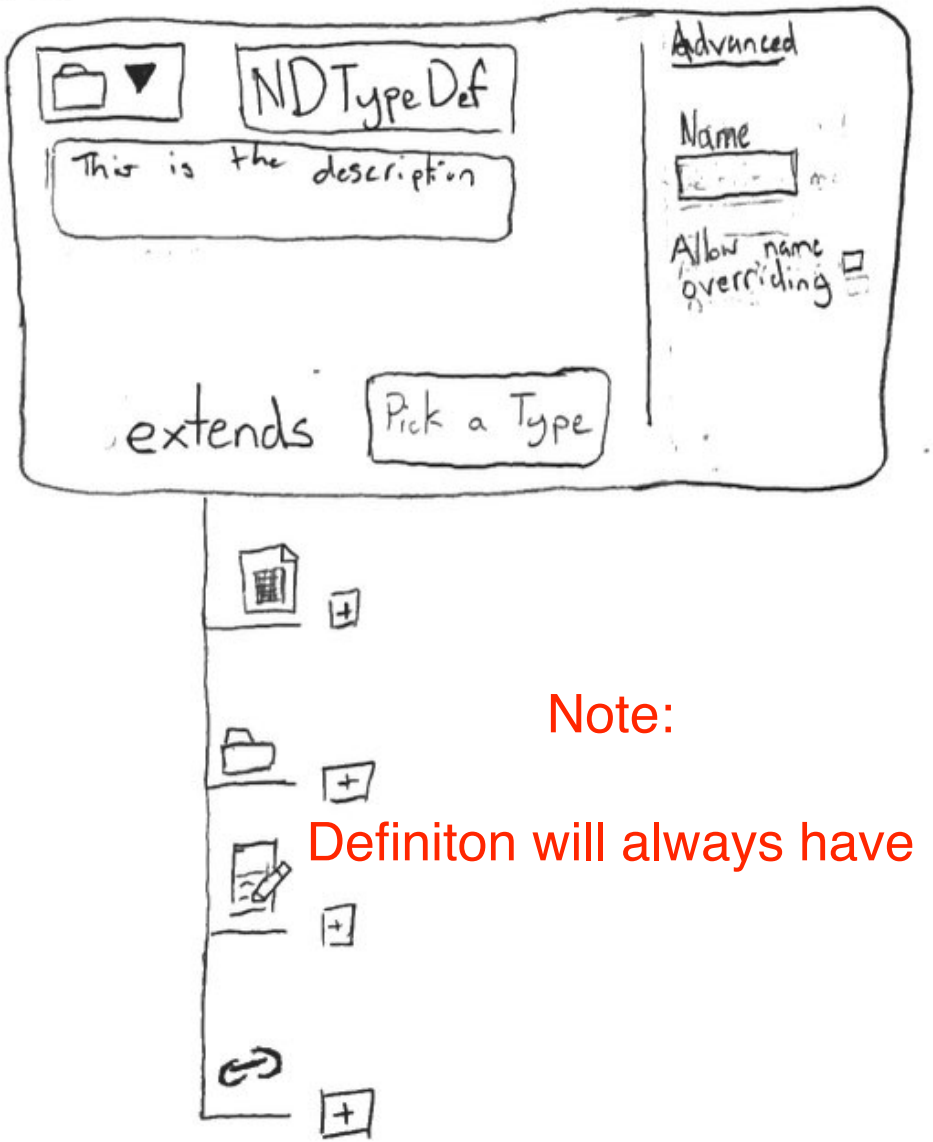


User can now click on a type to re-edit,

Or use arrow to move on and define their namespace

<typedef-builder>
<Group-Typedef-builder>
</type def-builder>

Group typedef

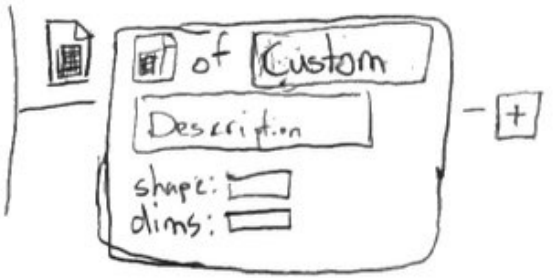


Note:

Definiton will always have

<anon-dataset-dec>

Anonymous type dataset declaration



or

< Named type dataset declaration