Production Calculator ^ Responsible for validating data, as well as making the proper calls to the calculator module in the proper order **Questions** Answers How will items be • At the moment, every item Summary fetched given an item INPUTS: • The action of pressing the submit button has an ID; the ID is then identifier? What will used to search for the item <<store>> <<view/controller>> Stores shouldn't interact A store will contain the the identifier look like? <<module>> Calculator in an object containing info <<module>> Loaded Factory The item identifier **FactoryEditor** Output The item count required per time unit currently-loaded factory; about every item in the User submits valid IRPTU request Valid IRPTU Submission the loaded factory can't be The ID is a string LoadedFactory:Factory + SaveChanges Info about the item is fetched stored in the view because + AddIRPTU(itemID:string,units:uint) + AddIRPTU(itemID:string,units:uint) + ClearFactory Loaded factory is updated + RemoveIRPTU(itemID:string,units:uint) the data will get wiped • I like the idea of having a A store will track user I'm thinking about Visual output of loaded factory is refreshed store contain the having a factory be Controllers and views will be super coupled information, including save currently-loaded factory "loaded"; there could • An empty factory could be slots for factories be a store that acts as loaded in by default There will be a menu in the a model for the loaded The store would be used <<store>> INPUTS: • The action of pressing the submit button whether the current user is **User Data** <<module>> <<component/controller>> factory from a save slot signed in or not SaveSlotsEditor **Validators** The item identifier User submits invalid IRPTU request invalid IRPTU Submission The item count required per time unit A user can edit a factory, + SaveSlots:Object<<key,Factory>> but will be unable to save + AddIRPTU(itemID:string,units:uint) + SignedIn:bool Bad inputs could be avoided The identifier doesn't correspond to an item their changes unless + Username:string by placing all the valid How will a message indicating inputs in a dropdown/search they're signed in a bad input be displayed to the Visual output of loaded factory is refreshed The stores will act as the user? Will I set it up so that I would need to some model, and the components bad inputs are impossible? pre-determined list of valid item identifiers will act as the view and The API and calculator A username and password are submitted logic are model territory, • The user's data can be The user's data is fetched from the database but could still be seperated loaded in on sign-in FPC Glossary The user's data is loaded into some store When will the user's data The load in will not overwite into their own files and User Successfully Signs In Valid sign-in The state tracker of the entire site is updated to initially be loaded in? Will the TimeUnits the currently loaded factory. • Items Required Per Time Unit (IRPTU) the signed-in status called by the models load-in overwrite the existing This is because an options The unit that represents the number of items required per unit of time; the specific item, box will appear with the The menu to load in a factory from the user's data as well as the type of time unit being used (minutes, seconds, etc.), are both implied by user's save slots, and they the context of the system and the usage of the unit type can then be loaded at will All the valid item identifier The menu to save a factory to a save slot is inputs can be placed in a The menus/buttons related to signing out are A series of one or more outputs and inputs; always ends in a user-required output The ability to load in a dropdown menu that gets What features will be enabled factory from the user's data filtered down via a search The data representation of a production chain; contains output data and input data, as well as the The ability to save a factory data used to link the two to a save slot The user's data will get • The ability to sign out Can be represented using one or more production chains The action of pressing the load button while When a user signs in, a certain save slot is selected The data representation of a factory; contains a series of production chain objects User Loads In Factory The user will have access to become available Factory Loaded ————— • The store/property containing the loaded factory What does the user all the same features as a is overwritten experience for a logged out The demand of particular items required directly by the user signed-in user, but won't be user look like? able to save their changes to Signed-In User The visuals relating to the loaded factory are the loaded factory The demand of particular items required in the process of filling user demand Used to visually represent the data pertaining to an item in a production chain This is a feature that's not Can a signed-out user still required in my mind, so I save changes to a factory? likely won't put much effort The action of pressing the save button while a certain factory save slot is selected I imagine I'd use cookies or User Saves Loaded Factory Factory Saved something of the like The factory associated with the save slot is overwritten Signed-In Ùser I think I'm just gonna bounce A factory will be loaded in The save slot menu is updated to reflect these How will I keep track of all the How will I integrate the Will a factory always be from requirement to Like the API, this is also model by default, and will only be calculator logic into the use cases I've covered and requirement and come up with territory; at-least I know that overwritten by other the cases I still need? use cases till I feel generally the logic shouldn't be placed factories, or cleared confident about the system in the components The action of pressing the save button while a factory is loaded Could be a sort of visual User Saves Changes To Factory Factory Changes Saved What will IRPTU output look chart connecting the nodes; I think these requests could seems more difficult Will I have an extra layer just be added to the stores; I How dumb/smart will my My components could make The save slot associated with the loaded factory will be overwritten by the loaded factory seperate from the model for calls directly to the models, Could be a more structured, components be? could seperate them out into seeing as they're partial controllers? table-like design my API requests? their own files and import them into the stores for more The save slot menu is updated to reflect these breathability, though Will be filled in along with What will crafters output look the nodes already on the User Clears Factory ...Observation... Can use a pre-determined A factory can have multiple My current calculator module ...Thought... The stores, in this case the How will validation of list of valid IDs for the item different input and outputs; the produces an object whose production output be carried The views shouldn't have to resulting chains from each keys are the IDs of the item Can use typescript models, shouldn't know about know about the models; this input can exist completely types used somewhere in the eachother; I'm not enforcing should work because the Schema? production chain, and whose seperately, or they can overlap decoupling through methods controller part of each at one or more nodes; for values are objects containing or interfaces, so this is the component can just pull the example, two inputs requiring info about each item type. least I can do required values out of the metal plates somewhere down stores and expose them to the This info includes the amount the line will overlap at the • I think "production chain" is views for binding metal plates node; the chain of each item required to be a good name; the fact that I should come up with an resulting from one output is a produced per second, or the the production chain is an official name for the output of object in the context of the tree structure; URPS. The URPS is actually the production calculator split among two different code will just be implied; I values, though. One value is can still refer to it as the the amount of the item "production chain object", required as output for the though user; the other value is the amount of the item required to My intention is that users make other items. will be able to model their whole factory over the What is the context of the This info also includes an object containing a list of all therefore, the website the items who require each should be more of a item to be built themselves; modeling tool as opposed to each entry in the list has the ID a quick mockup tool, like of the requiring object, as well what other websites are as the portion of the parent object's URPS they require The user should have more How does the context of the fine-grain control over the website affect the system; for example, they should be able to edit every node directly, as opposed to Since the user will be able to How will the removal requests edit every node directly, the removal requests will have to factor in the products that rely on the product being The time unit a given production chain is using should stored in the object I don't wanna store too How will time units be much in this object, but the alternative is representing the data in an entirely • The only way to ensure the data stays with the object is by passing it along with the What relationship do crafters have to time units? I don't know enough to come up with alternatives The current implementation actually makes it easier to add extra data to the object, as well as store in it a Alternatives to storing everything in one data The stores will have to know structure and passing it alot about the object if I want the methods in the modules to just take certain parts of the object; for example, I just want the methods in the modules to

only take the parts of the

