Project Summary:

The proposed title I have for this project is atomic household because its intended user is the nuclear family ha ha. So this project will allow the main user(parent) to log in, create tasks for their kids, and depending on if that task is complete or not add a reward in the form of an allowance, or punish the kid in the form of taking money out of their allowance, the parent can also view and see what tasks have been completed, and they can add as many children as they want. The kid will be able to sign in view what tasks thy have been assigned and upon completion they can say its complete and view their allowance. Both the parent and the child will be able to send each other messages and they can reply to each other and also view any previous messages. The problem I trying to solve is forgetting when your parents give you a task this way you can just look and see what you have to do for the day, and on the parents side sometimes when they are at work and need meat or something taken out of the freezer they can just send their child the reminder and it can get done in time for whatever they are planning. I am going to be user a file to serialize everything and some arraylist to store and add data to those files.

Data Design:

My programs data is mainly focus on storying user input and also allowing the user to change some input, for example when a task is assigned, I want to store that, then when the task is completed, I want to change it to reflect the completion.

I am using array lists to store and add to that data

There will be some aspects of the data that need to be persistent like what tasks are assigned to who, if they completed it or not, the amount of allowance they have, and all this data will be serialized.

Each class will use some sort of aggregation to implement certain features, for example the parent class will need to use the assign method from the task class and the Kid class will also need to use the view task method from the task class.

UI Design:

Currently my project is console based and it has a “menu” screen that takes the users to different functions depending on their choice. The user can either log in create or a new account both of which will ultimately lead to them being able to have all the functionalities available, or they can see what the app is about and what they can do with it.

Graphical user interface, application, Teams

Description automatically generated

Algorithm:

For the User class the user will have to sign up or log in, I will be using a username and password and having their data types be strings, and I will save that information in an arraylist and serialize it so that the information can be saved and the user can log back in and have their progress saved, from there they will be directed to either the parents page or the kids page. From the parents page the parent is able add elements to the kid arraylist because they might have more than one child and that child’s information will be saved in an arraylist, most of my data types are going to be strings because when the parent is assigning a task they need to name the task etc, the parent can add an allowance to their child but its not required and the allowance data type is double to account for any cents they want to give, and everything the parent done is saved into an arraylist for consistency. The kid will be able to complete tasks, view their allowance, and mark their tasks as complete, when they mark their tasks as complete it will append the empty completion status so that the parent can see which tasks have been completed and which ones haven’t. The task class itself will have the methods of adding and viewing tasks, as well as changing completions status all of which will be strings. The tasks themselves will also be stored in an arraylist so that either party can see a list of what they have to do/what is done. As for the allowance class this has 3 basic methods, they all take double as their data type, the first one is to add an allowance, the second is to take from an allowance, and the third one shows what is in the allowance balance. The final class I have is the message class and both the parent and Kid have an instance of this class so that they can send messages to one another, view the messages, and look at all message history. For this specific class I serialized it separately, and store all messages sent from either party.