**Behave! Documentation**

This implementation of behave is started by the runnable or the Main class. This main class holds the current Mode that the machine will be in (the userMode can be made with a login) and prompts for commands. This main file passes the commands to a handler which uses the userMode to execute commands from the correct user. The parent user can add/edit/delete children, tokens, and rewards. Both the tokens and rewards are based dependent on what mode the program is in (either positive or negative, which are denoted by the Identifier / type int. This can be expanded on later with more identifiers for more modes). Each parent can keep a list of children, where each child contains a list of rewards and tokens. The rewards are saved as a pair of two items, being Dicts. Tokens are saved in the same way. Each token holds info about that token, with the ability to be extended to have pictures added. Each token and reward is given a unique ID to be easily identified later or referenced from elsewhere. The modes are written as abstractions so that more can be added on and easily changed. Each user type holds an allowedModes list, so that more modes can be easily added to any existing or new user. The user class is also an abstraction, so that more Users can easily be added on later. All the info is saved in local memory as a list that is not removed with each addition or subtraction, so that list can easily be moved to a more permanent form of memory (such as updating to a database such as firebase using API calls). Each of the tokens are timestamped, and saved in a list; so they can be referenced later and organized by their dates to see trends.