Meeting #1 – 11am 10/4/24

Attendance: Kate, Jimmy, Shane, Muhammad

Meeting Agenda:

* Discuss GitHub and user environment. Divide up userstories assignment due Monday.

Talking points for next Meeting:

* How we want to divide up implementation.

Meeting notes by Kate

Open-Door policy on issues

* Issues day of? Come forward and ask for help
* No judgement on issues
* Communication!

Group Work Deadline

* Have things done by Friday midnight, so that we have buffer of weekend for straggling items to be pushed

GitHub

* Jimmy put in safegaurd for pushes to main
* Being careful pushing to main
* Do not use Force Merge without talking with group about why it’s needed
* Branch Structure:
  + Feature/”your-name”
    - Where we will be working individually
  + Develop
    - Where we push our features to
    - This is where we review as a team
  + Release
    - Our working version
  + Main
    - Finished “live” version
    - Will be what we push to for our final submission
    - What we turn in for each sprint
* Running into issues or confused on Git commands? Put is in the discord. Jimmy is good at Git!
* Creating a branch: tutorial in GitHub commands channel
  + FOLLOW JIMMY’S INSTRUCTIONS TO GET YOUR WORKING VERSION

Environment

* Java, JDK 22 (this will auto implement when connecting to our repository)
* Checkout commands Jimmy provided for help with commandline Git
* Note: update PowerShell to version 7 if on Windows
* Update IntelliJ if needed

Sprint 1: User Stories Due Monday midnight – Dividing work by color

\*\* Create user stories that encapsulate highlighted area

\*\* minimum of 5 user stories, but do as many as needed

\*\* submit to discord in one file, to be compiled in one file once everyone has submitted

Kate

Jimmy

Shane

Muhammad

Colton

Class Identifiable by a unique name

Add requires a name. should fail if name is invalid or duplicated

Delete requires a name or selection

Rename requires a name or selection. Should fail if name is invalid or duplicated

Relationship Identifiable by a unique source and destination

Add requires a "source" and "destination" (currently no types)

Delete requires an identification to remove

Fields/Methods

Add Field requires a class name and a unique name; should fail if name is invalid or duplicated or if class name does not exist

Add Method Requires a class name, method name, and list of parameter types and names; fails if method name and parameter types are identical to another existing in the class, class/method/parameter names or types are invalid

Remove Field requires a class name and a unique name; should fail if name is invalid or duplicated or if class name does not exist

Remove Method Requires a class name and method name. If overloaded, give user a choice of methods with this name. Fails if class or method names do not exist

Rename Field requires a class name and a unique name; should fail if name is invalid or duplicated or if class name does not exist or if new name is duplicate

Rename Method Requires class name and old method name, prompt for one or all methods with that name (only if overloaded), then new method name. Fails if class or method does not exist or if new name is the same as one with the same parameter list

Add Parameter Requires class name and method name, prompt for one or all methods with that name (only if overloaded), then a new parameter name and type. Fails if class or method name does not exist, or if new parameter matches an existing one

Remove Parameter Requires class name, method name (choice if overloaded), and parameter name. Fails if class, method, or parameter do not exist

Change Parameter Replaces an existing parameter (or all parameters) with a new list of formal parameters. Fails if a method does not exist.

Save/Load Functionality related to saving and loading state

Save Should save to a JSON or YAML format

Load Should load from a JSON or YAML format

Interface Should be a CLI application initially

List Classes There should be a command that lists all of the classes and contents of the classes in some nice way

List Class There should be a command that lists the contents of a specified class in some nice way

List Relationships There should be a command that lists all of the relationships that exist between classes in some nice way

Help There should be a command that lists how to use the application without the user leaving the program

Exit There should be some way to exit the application