NORTHERN ARIZONA UNIVERSITY

Hayden Aupperle, Gage Cottrell, Peter Huettl, Garrison Smith

Drop the Code

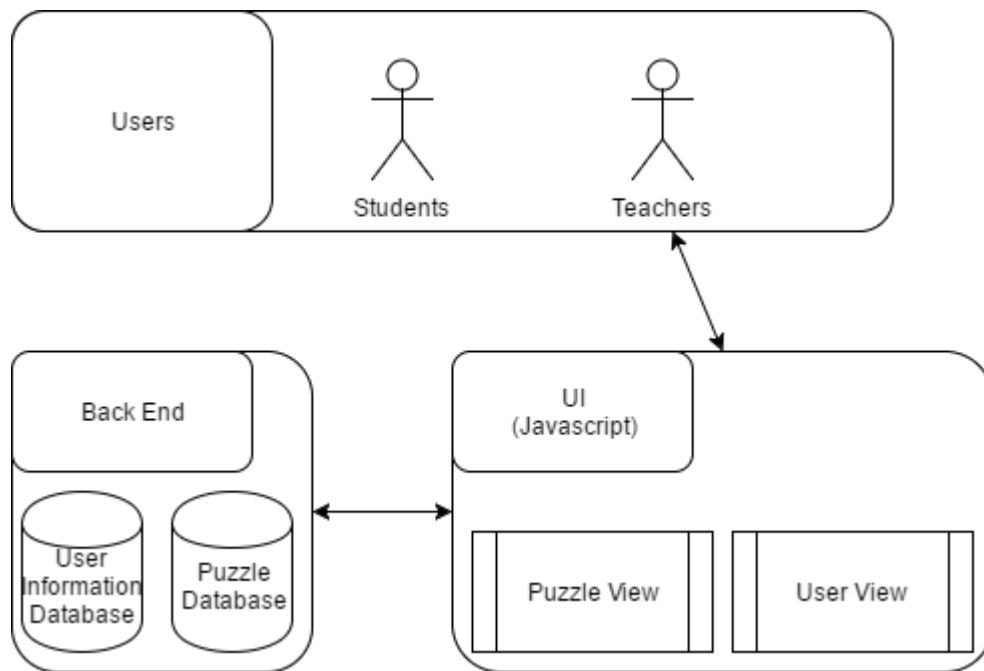https://github.com/petetetete/cs386-project

D5 – Internal Design

CS 386 – Software Engineering

Spring 2017

Marco Gerosa

High-level Architecture:

Our high level architecture diagram outlines the three major components our system operates on and with. The first of which are the *Users*, who can be either students or teachers, and they interact directly with the second major component, the *UI*. The *UI* acts as both a view and controller to the data that is stored on the *Back End*, the third high-level component of our software.
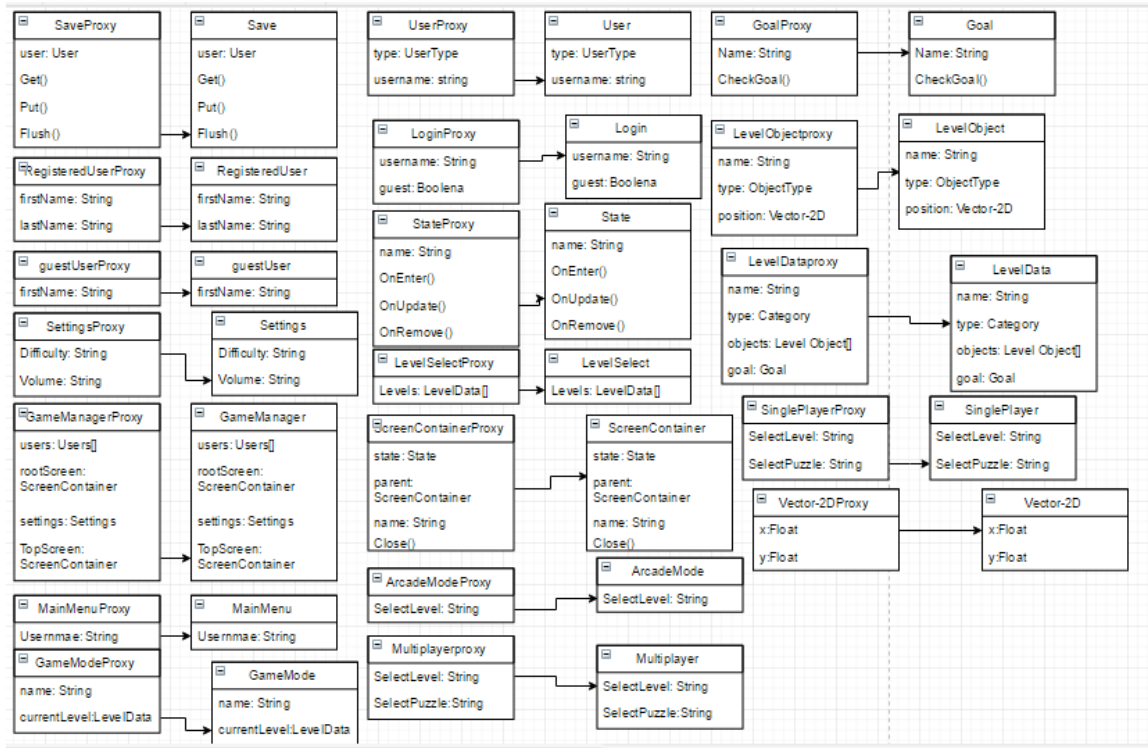
Class Diagram:
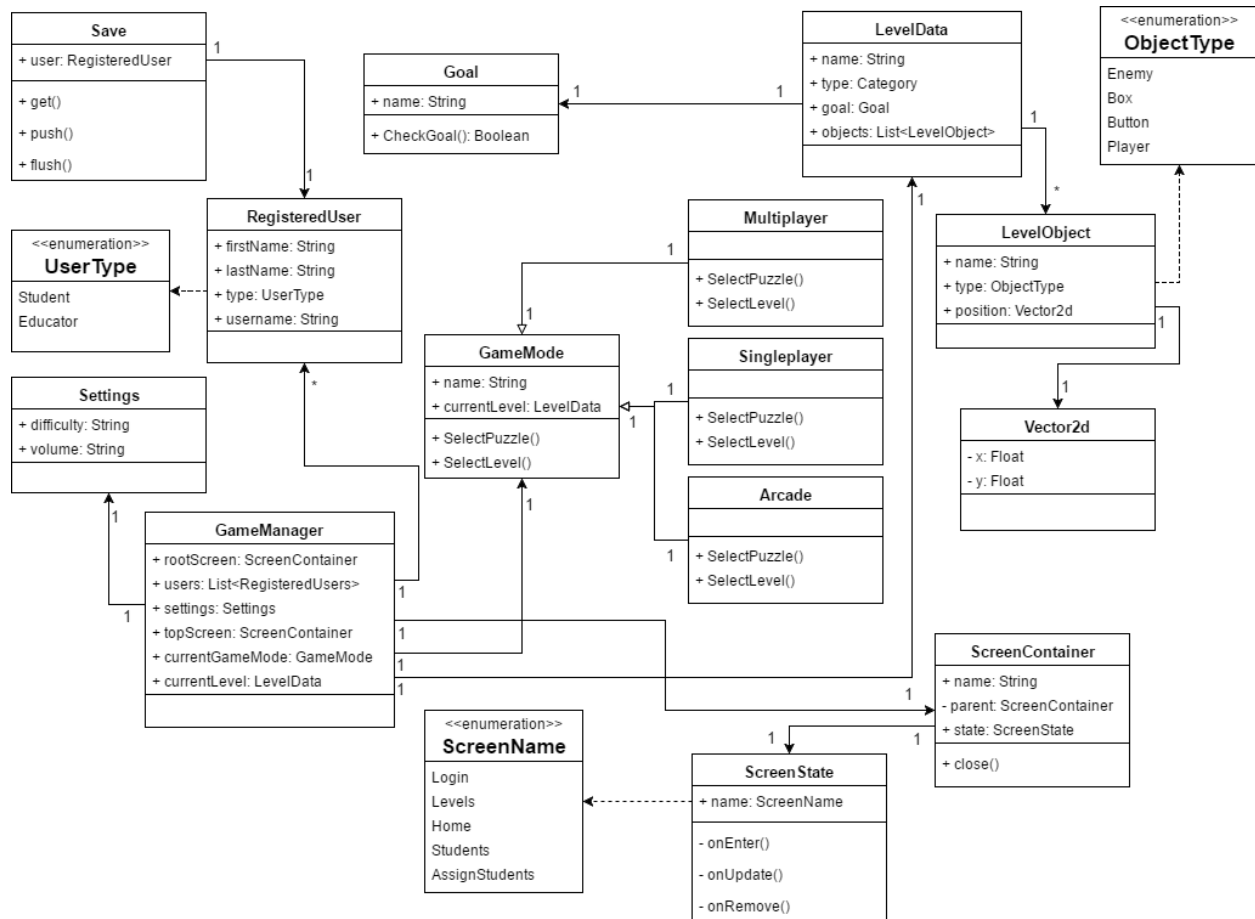
We added visibilities to all member variables and methods and added information to the relational arrows to denote the kind of relationship. Aside from that, we also cleaned up unnecessary classes and improved the attributes on the classes that do exist. Now the relationships are more clear and accurate to our system.

Sequence Diagram:
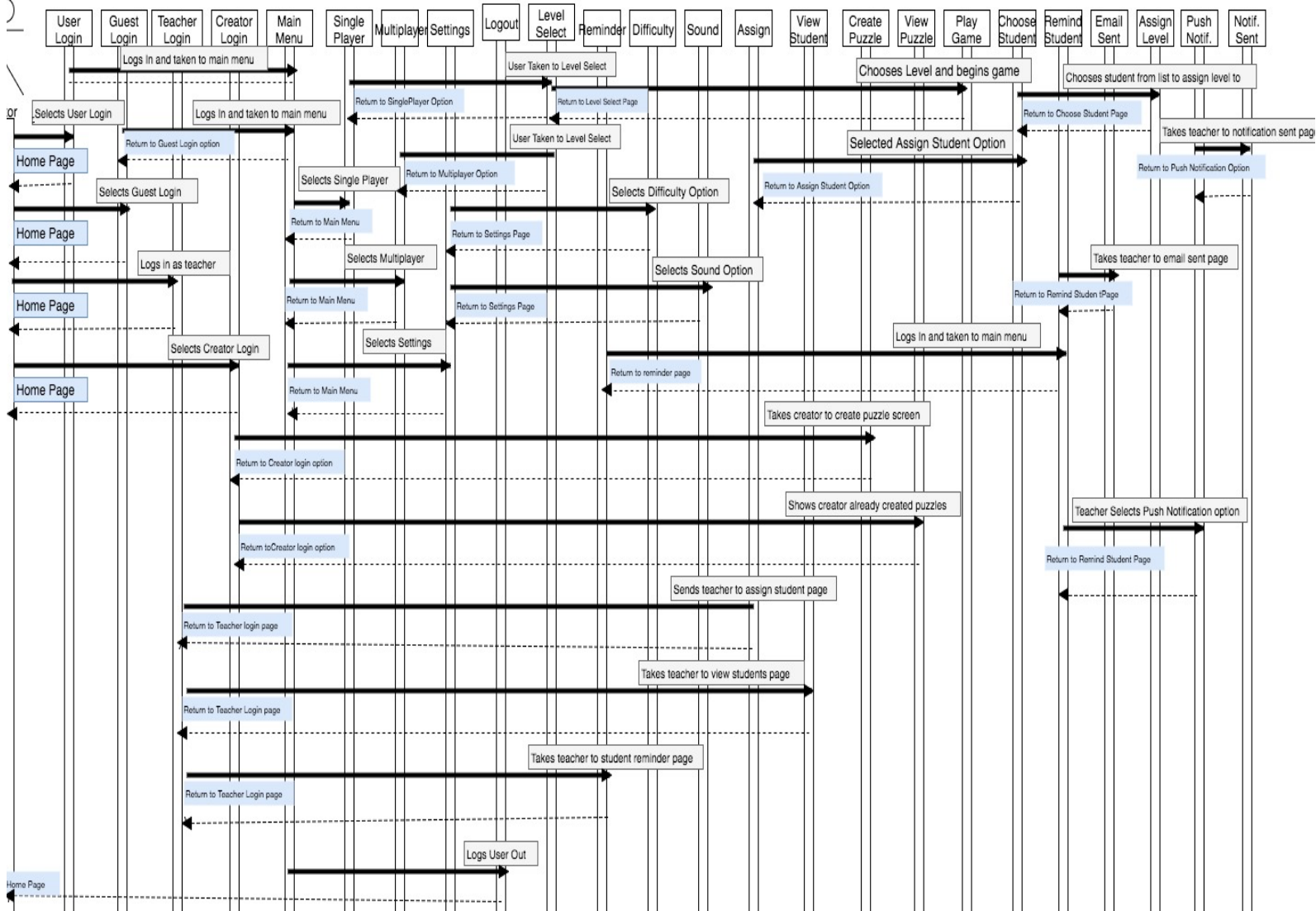
The sequence diagram displays how all of the different menus interact with each other and where you can go from each menu. For example, you can navigate from the main page to the single player menu, multiplayer menu, and the settings page, which you can navigate back to the main menu from all of these menus too. There are several different menus to see in our application and this diagram shows how you can get to each of those menus by following the path to that menu based off which arrows lead to it. Furthermore, the diagram shows how you can navigate back to other menus from the current menu you are at.

Class Diagram Outlining Design Pattern:

The Outlined Design pattern for the class diagram is similar to the class diagram except that it uses a proxy class which is a copy of the original class that links together to make an outlined design pattern diagram.

Group Participation:

Peter – Create initial documents and template the deliverable and implemented the revised class diagram.

Hayden – Implemented sequence diagram.

Garrison – Implemented the class diagram using design patterns/used the proxy diagram.

Gage – Implemented the high level architecture diagram.