Rebecca Rothschild

Isaac Schultz

Josh Bennet

**Class Diagram**

Our diagram contains several classes, some of which are for potential extra credit. We decided to create an abstract basic user class that four other user classes inherit from. These include a Registered User, Unregistered user, Business Admin, and a potential Moderator class for extra credit. Our Business class is composed of several properties that include instances of both the Hours and Attributes classes. One of our other opportunities for extra credit is including the ability for both businesses and users to reply to reviews, so we created a Reply class. Since several of the properties in Reply exist in the Review class, we created an abstract parent class called Posts to house the repeated properties and methods. We have included a Map class in our diagram to denote the already existing BingMap api library we will be using in this project. For our application, we have decided to use events to handle the changing our data. When a user inserts, deletes, or adds any data into the database, that will be handled in our QueryEngine class. This class takes care of all the interfacing between our application and database. We wanted to keep the GUI class as basic as possible, so it will be calling methods in the QueryEngine to populate/ update the GUI. We also have a Checkin class.