Project Documentation

Construction Company Management System (CCMS)

The four limbs of Object Oriented Programming:

Our project implements the main aspects of Oop in the following manner:

* Encapsulation:

*Encapsulation* is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse. Encapsulation is as a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside the wrapper. Access to the code and data inside the wrapper is tightly controlled through a well-defined interface. We can observe this feature in CCMS project, consider different classes in this project like builder class, retailer class, customer class etc. all these classes keep their member variables, member functions and data to themselves. The code and data inside each of these classes cannot be arbitrarily accessed by other classes.

* Abstraction:

An essential element of object-oriented programming is *abstraction.* We manage complexity of this management system through abstraction. We have implemented the feature in the following way, there are many users to this system like builders, customers, retailers etc. all of these users need not worry about the working of the application and the intricate details in them, they only have to know about the GUI interface and how to work with it. This level of abstraction is managed through the use of hierarchical classifications. This allows us to layer the semantics of complex systems, breaking them into more manageable pieces. From the users view, the application is a single thing. Once if we see into that application it consist of several classes which are then again interrelated and also a complex database management system. We manage all these with the help of abstraction.

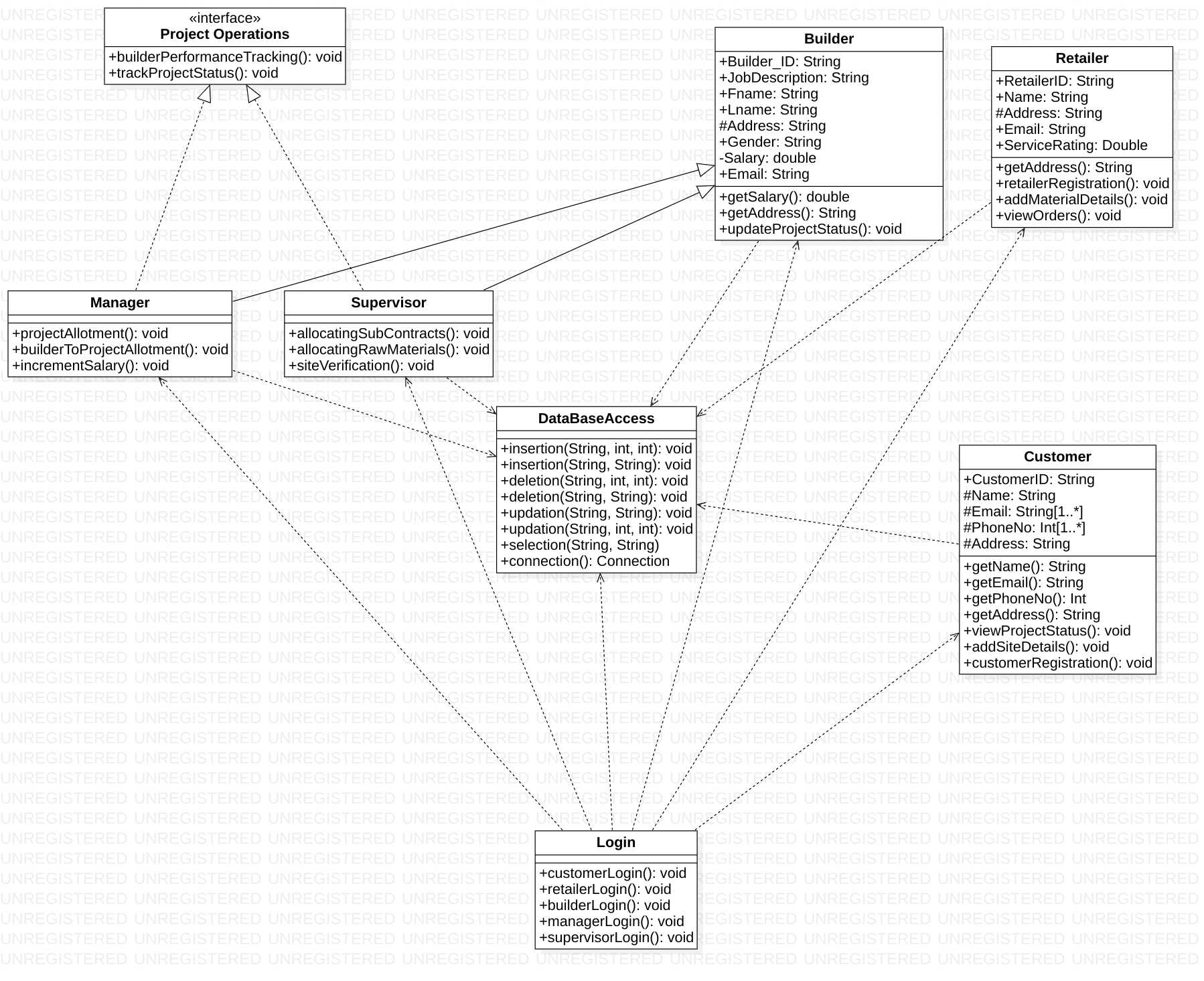
* Inheritance:

*Inheritance* is the process by which one object acquires the properties of another object. In this project all the employees in the company are called builders and some of these builders could hold different posts such as a manager, a supervisor or a accepting authority. All these builders with different posts inherit the common features of a normal builder and with addition to it they have their respective features and functionalities.

* Polymorphism:

*Polymorphism* is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation. In this CCMS, there is an interface “ProjectOperations” which has two functionalities namely “builderPerformanceTracking” the function that tracks builders performance and “trackProjectStatus” the function that tracks the status of a project. These functions are implemented in one way in case of the manager and in other way in case of supervisor. In case of manager he tracks the performance of builder and he rewards them with bonus and in case of supervisor he keeps track of the builder and helps him achieve the given work.

Class diagram



Group Members:

1) C V Aswarth Narayana (19016)

2) Kotla Jaideep Reddy (19067)

3) Boggavarapu Nikhil (19063)

4) M Vishwanath (19032)