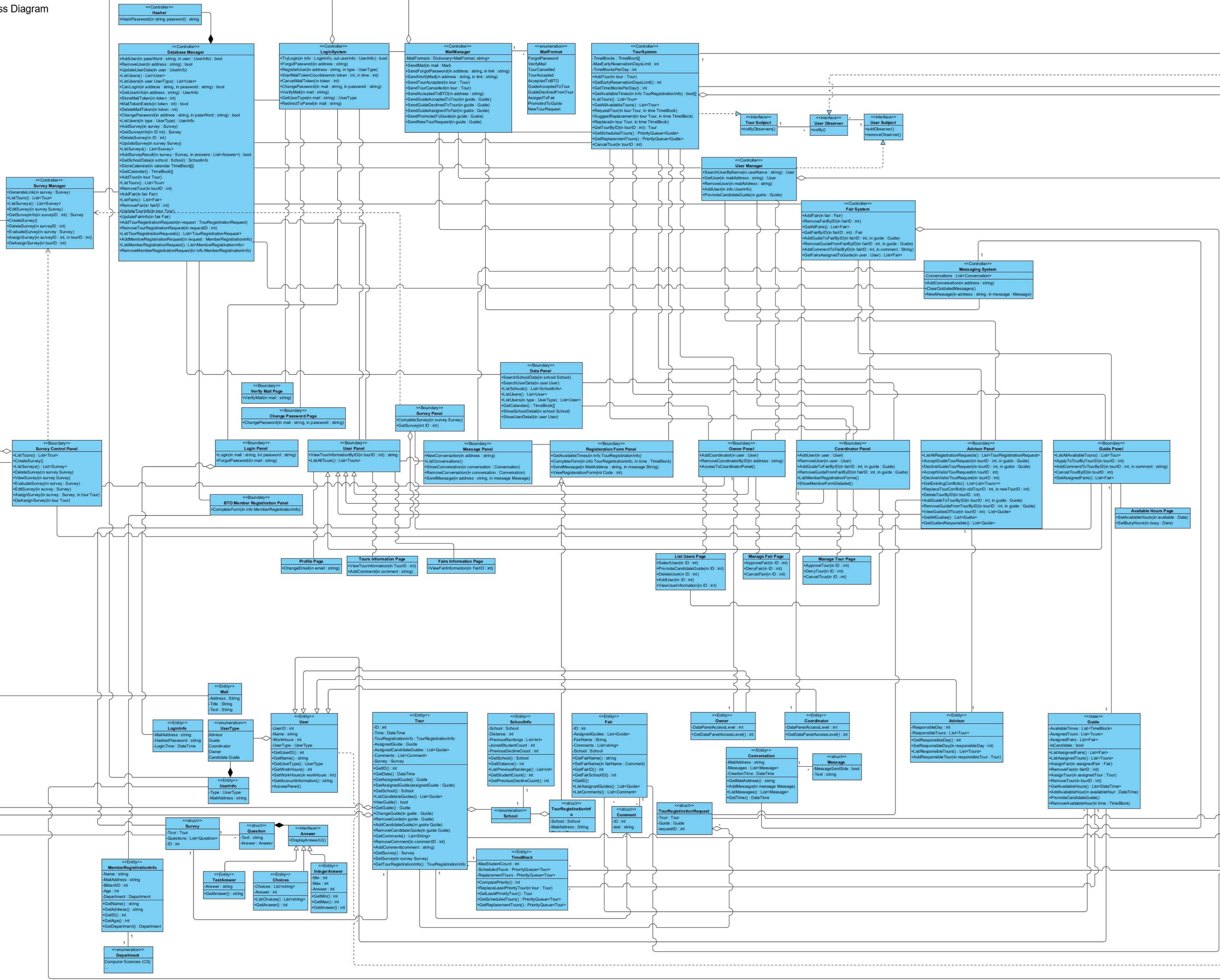
CS319 Term Project

Deliverable 4 - 1st Iteration

CS319 Section 1
Sami Bora Akoğuz 22202184
Ege Ertem 22202433
Can Kütükoğlu 22202619
Kerem Cindaruk 22201907
Ertuğrul Malkoç 22102737
11 December 2024

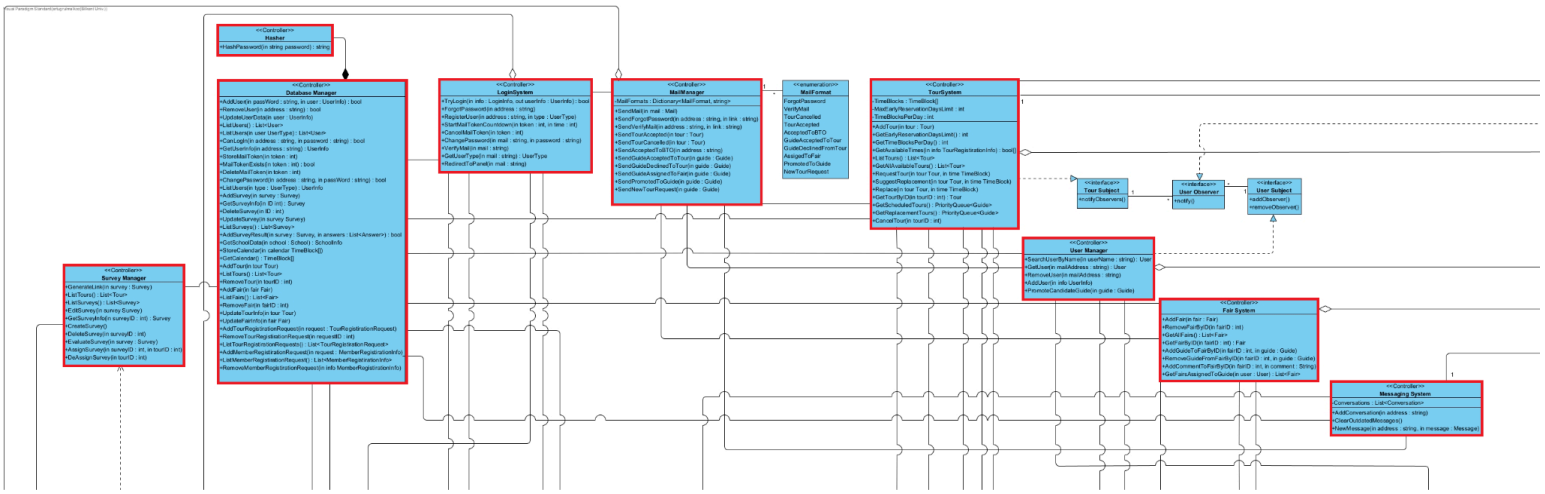
1.Class Diagram



2. Design Patterns

2.1 Singleton Pattern

For our web application, we have chosen to use the singleton design pattern for our controller classes in order to reduce system complexity and increase our control on the system. By keeping only one instance for our controller classes, we not only ensure consistency and reduce error possibilities, but we also reduce the system requirements for our application. This approach allows us to maintain centralized control over key systems such as Tour Scheduling. The example classes where the singleton pattern is used are: Survey Manager, Database Manager, Login System, Mail Manager, Tour System, User Manager, Fair System and Messaging System. These classes all control certain events and manage different objects throughout the runtime of our application. Making them a singleton ensures that the systems they control are reliable and robust.



2.2 Observer Method Pattern

In our web application, we have chosen to implement an observer design pattern in order to manage many instances of a class and update/notify them in a controlled manner. This pattern allows us to easily manage high amounts of User classes and notify them when it is needed. This ensures that dynamic/real-time updates are conveyed to these classes accordingly. We have decided to split our Subject interface into two parts as the Add/Remove functionality belongs to the User Manager controller class while the Notify functionality is in the TourSystem controller class' responsibility. This separation was needed because we wanted the User classes to be notified when there is an available tour in their allocated time zones but the managing users and managing tours are separated into two controller classes. User classes inherit the User Observer interface in order to inherit the needed notification functionality. The classes that use this pattern are: TourSystem (inherits Tour Subject interface), User Manager (inherits the User Subject interface) and User (inherits the User Observer interface).