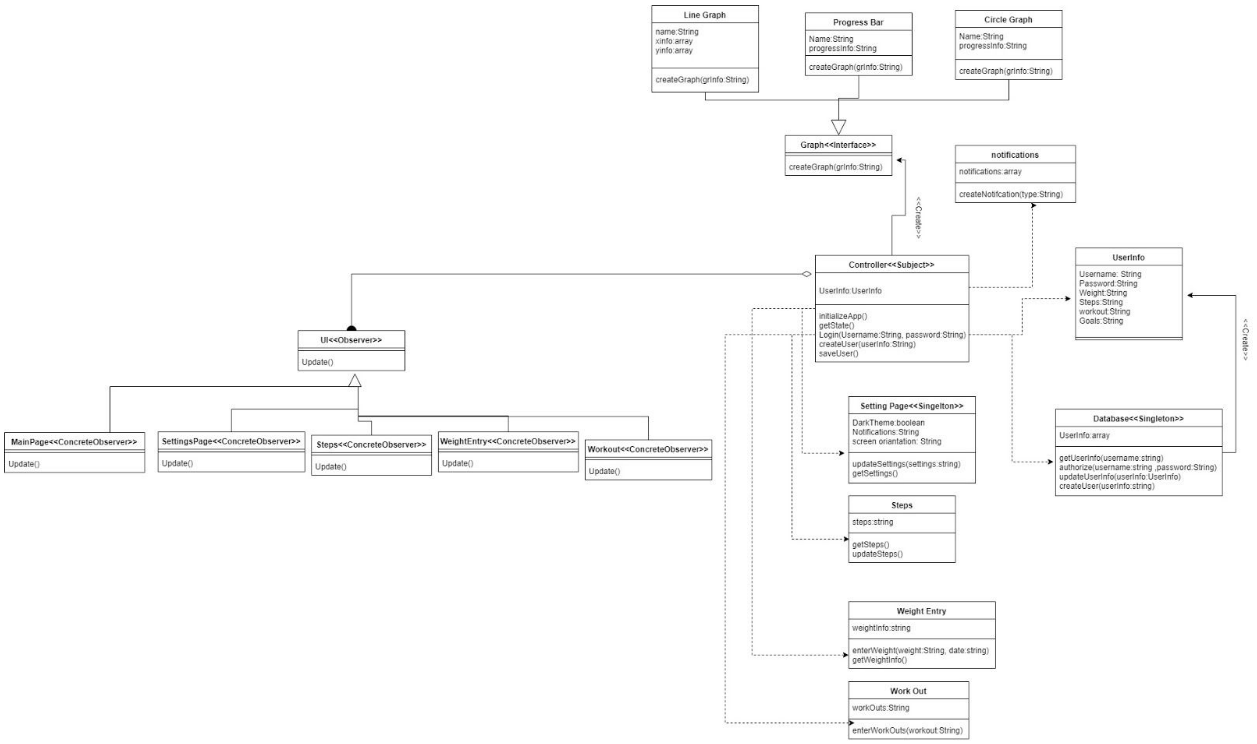
Detailed Design Class and Patterns Diagram

Design Class Diagram


Line Graph
name:String
xinfo:array
yinfo:array
createGraph(grInfo:String)
Progress Bar
Name:String
progressInfo:String
createGraph(grInfo:String)
Circle Graph
Name:String
progressInfo:String
createGraph(grInfo:String)
Graph<<Interface>>
createGraph(grInfo:String)
notifications
notifications:array
createNotifcation(type:String)
<<Create>>
Controller<<Subject>>
UserInfo:UserInfo
initializeApp()
getState()
Login(Username:String, password:String)
createUser(userInfo:String)
saveUser()
UserInfo
Username: String
Password:String
Weight:String
Steps:String
workout:String
Goals:String
<<Create>>
UI<<Observer>>
Update()
MainPage<<ConcreteObserver>>
Update()
SettingsPage<<ConcreteObserver>>
Update()
Steps<<ConcreteObserver>>
Update()
WeightEntry<<ConcreteObserver>>
Update()
Workout<<ConcreteObserver>>
Update()
Setting Page<<Singelton>>
DarkTheme:boolean
Notifications:String
screen oriantation: String
updateSettings(settings:string)
getSettings()
Database<<Singleton>>
UserInfo:array
getUserInfo(username:string)
authorize(username:string ,password:String)
updateUserInfo(userInfo:UserInfo)
createUser(userInfo:string)
Steps
steps:string
getSteps()
updateSteps()
Weight Entry
weightInfo:string
enterWeight(weight:String, date:string)
getWeightInfo()
Work Out
workOuts:String
enterWorkOuts(workout:String)

The above is the design class diagram with the design patterns within our app. We chose to include a single graph that represents the whole system. This better represents the interactions that occur within the system.

Design Patterns Present:
Observer
Singleton
Factory

Description of Diagram Classes

Controller<<Subject>>:
The controller is the system that observes and controls the entire state of the system. This is the only system that can generate notifications, create graphs, interacts with the user database, update settings, workout,steps,weight. The controller is also the subject of the observer design pattern.

UI<<Observer>>: The UI is an observer that has 5 concrete observers.Each observer represents a page present within the app. The controller provides state information for each page.

Graph<<Factory>>: Graph is factory design pattern meant to generate different graphs for the user.
LineGraph: creates a line graph
ProgressBar: generates progress bar towards goal
CirleGraph: Generates progress bar in circle

Notifications: Creates app notifications when called by controller.
Database<<Singleton>>: The database is a singleton design pattern. It can authorize users and sends user info objects to controller. It is the only class that can create new instances of user info. The database also stores all user info.
UserInfo: A class that holds all user private data. The controller can interact with instances of user info.
SettingsPage<<Singleton>>: The settings page holds all configuration information for the app that can be changed by the user or changes during runtime. The controller is the only object that gets and updates settings.
Steps:Holds and updates steps. Controller is only object that interacts with steps.
Weight:Holds and updates weight. Controller is only object that interacts with weight.
Workout:Holds and updates workouts. Controller is only object that interacts with workout.