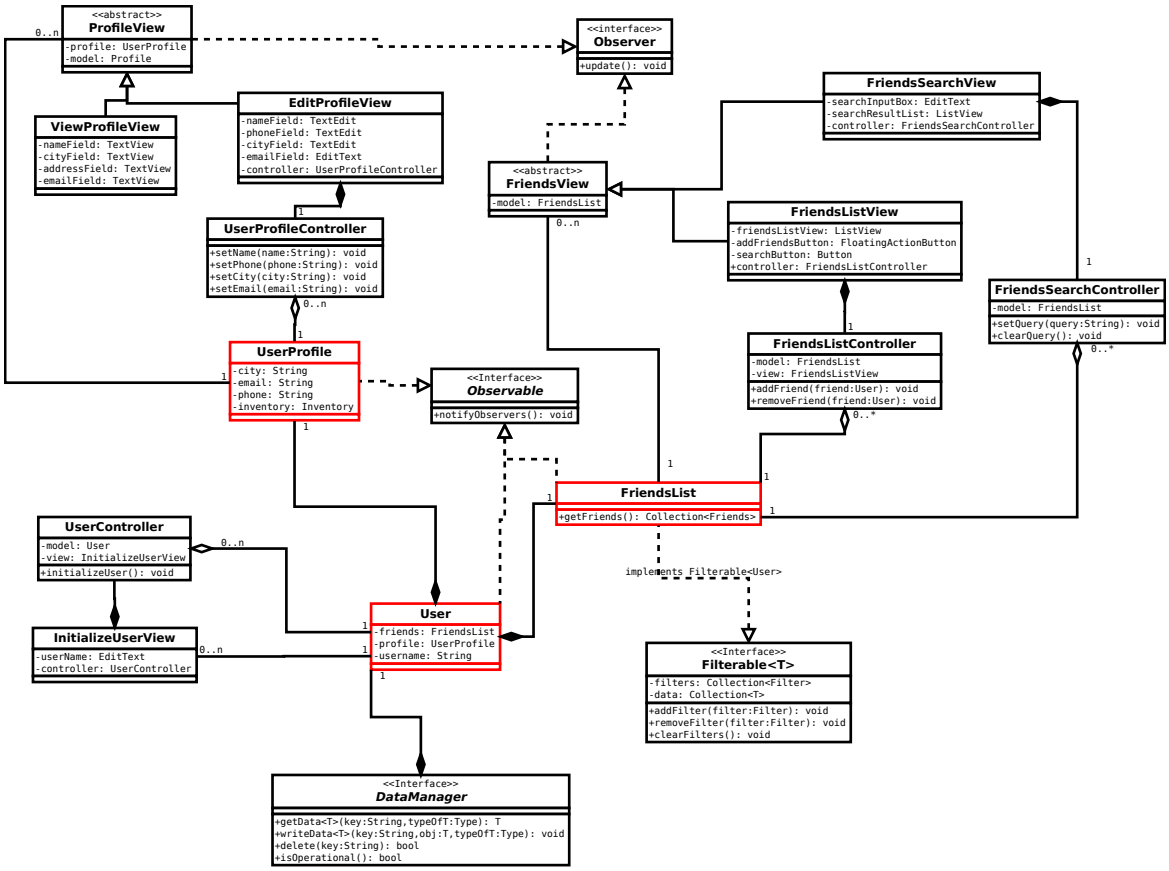
<<abstract>>
ProfileView
-profile: UserProfile
-model: Profile
<<interface>>
Observer
+update(): void
ViewProfileView
-nameField: TextView
-cityField: TextView
-addressField: TextView
-emailField: TextView
EditProfileView
-nameField: TextEdit
-phoneField: TextEdit
-cityField: TextEdit
-emailField: EditText
-controller: UserProfileController
FriendsSearchView
-searchInputBox: EditText
-searchResultList: ListView
-controller: FriendsSearchController
<<abstract>>
FriendsView
-model: FriendsList
FriendsListView
-friendsListView: ListView
-addFriendsButton: FloatingActionButton
-searchButton: Button
+controller: FriendsListController
UserProfileController
+setName(name:String): void
+setPhone(phone:String): void
+setCity(city:String): void
+setEmail(email:String): void
FriendsSearchController
-model: FriendsList
+setQuery(query:String): void
+clearQuery(): void
UserProfile
-city: String
-email: String
-phone: String
-inventory: Inventory
<<Interface>>
Observable
+notifyObservers(): void
FriendsListController
-model: FriendsList
-view: FriendsListView
+addFriend(friend:User): void
+removeFriend(friend:User): void
FriendsList
+getFriends(): Collection<Friends>
UserController
-model: User
-view: InitializeUserView
+initializeUser(): void
implements Filterable<User>
User
-friends: FriendsList
-profile: UserProfile
-username: String
InitializeUserView
-userName: EditText
-controller: UserController
<<Interface>>
Filterable<T>
-filters: Collection<Filter>
-data: Collection<T>
+addFilter(filter:Filter): void
+removeFilter(filter:Filter): void
+clearFilters(): void
<<Interface>>
DataManager
+getData<T>(key:String,typeOfT:Type): T
+writeData<T>(key:String,obj:T,typeOfT:Type): void
+delete(key:String): bool
+isOperational(): bool