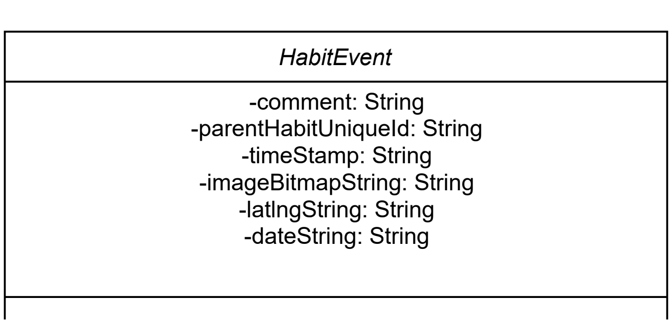


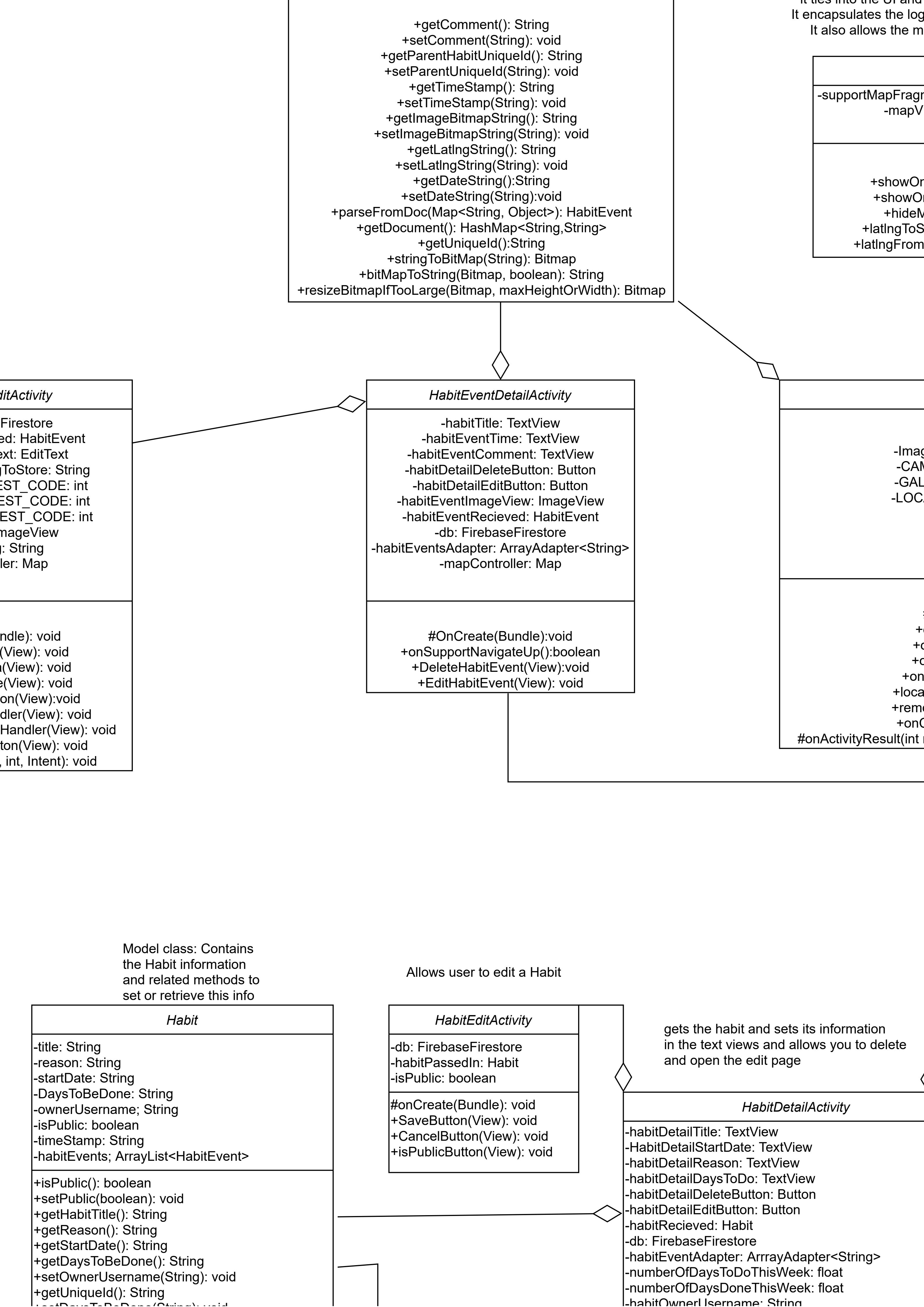
The Map class acts as a contr
it ties into the UI and

roller it helps in the separation of concerns for all things related to maps.
contains utility methods for latitude longitude string conversion

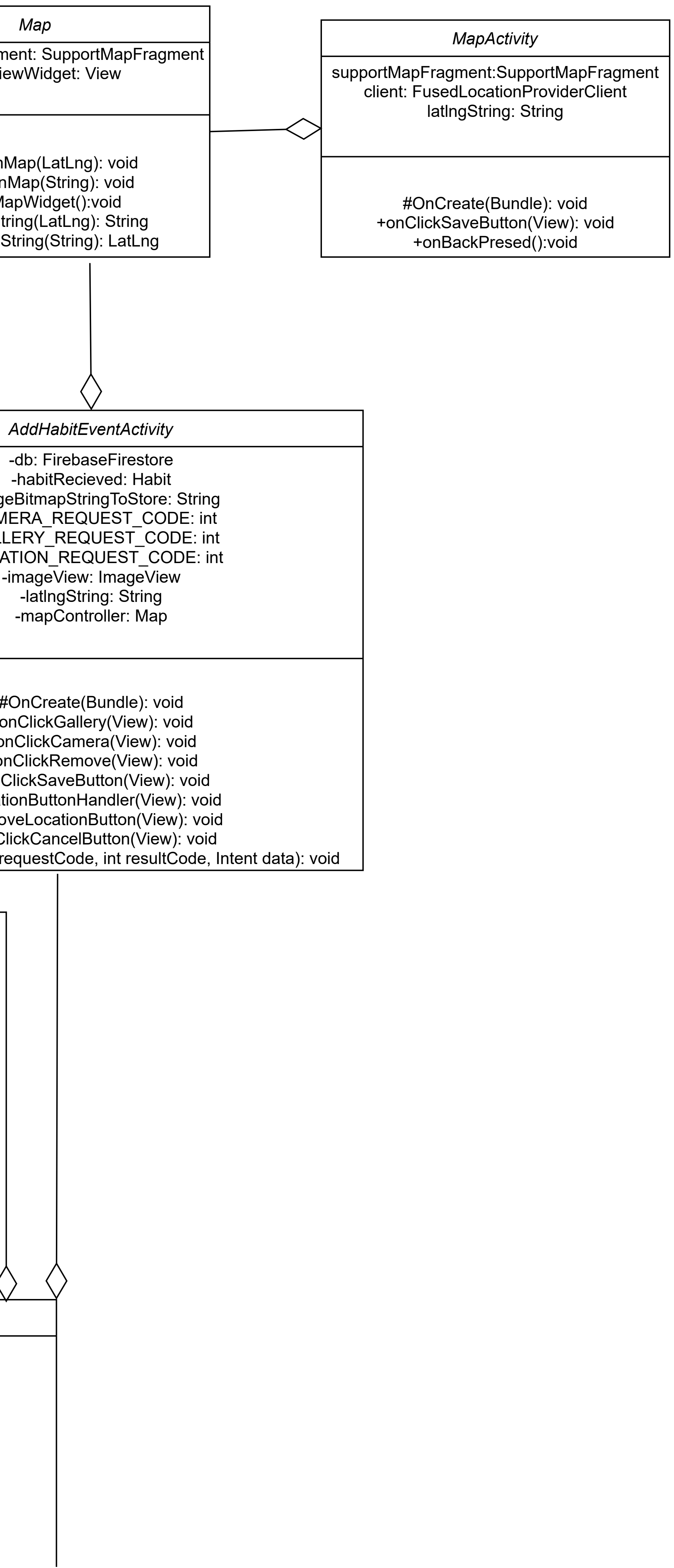
HabitEventEd

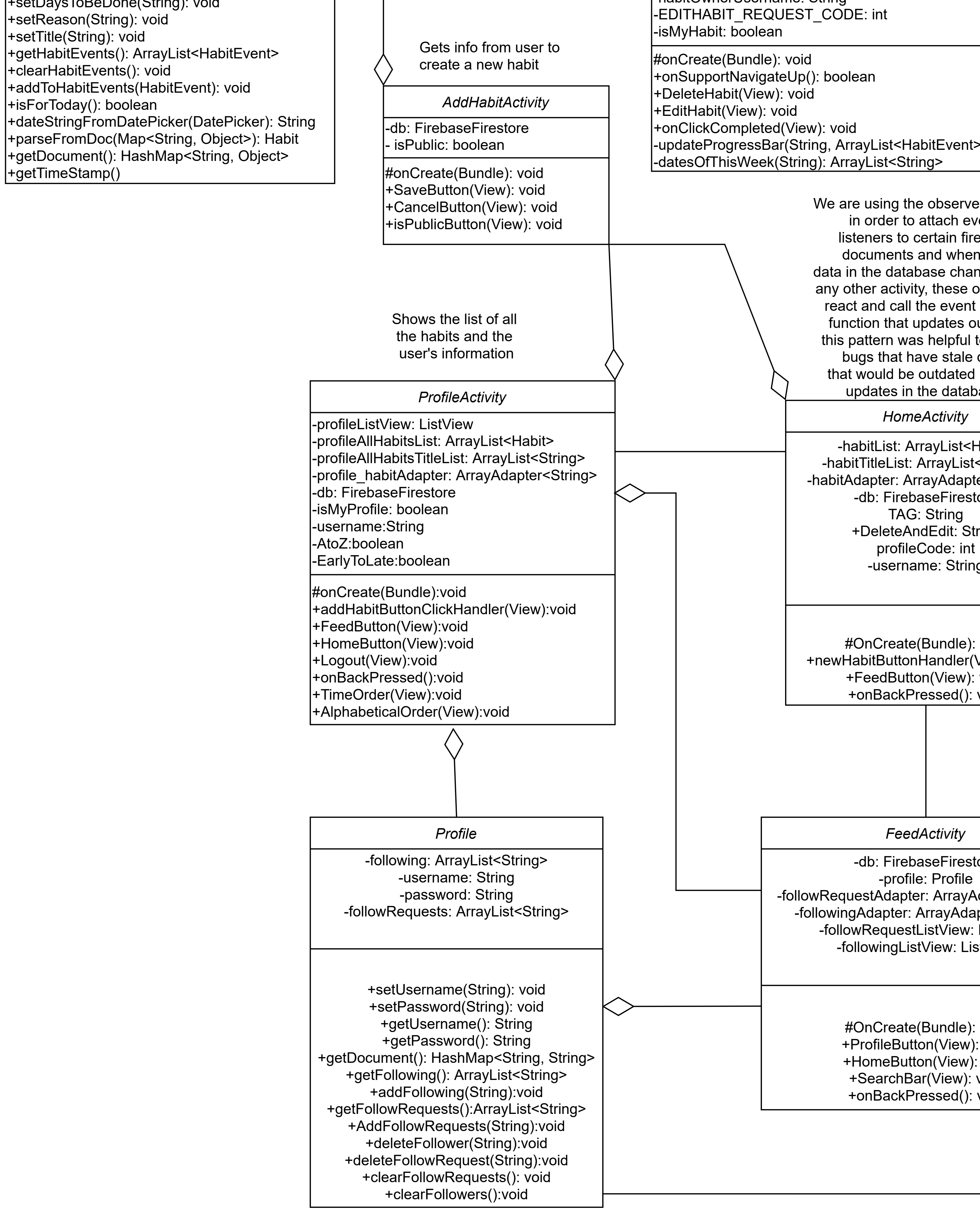
-db: Firebase
-habitEventRecieve
-commentEditTe
-imageBitmapString
-CAMERA_REQUE
-GALLERY_REQUE
-LOCATION_REQU
-imageView: Ir
-latlngString
-mapControl

#OnCreate(Bu
+onClickGallery
+onClickCamera
+onClickRemove
+onClickSaveButt
+locationButtonHan
+removeLocationButton
+onClickCancelBut
#onActivityResult(int,


It ties into the UI and
It encapsulates the log
It also allows the m
+getComment(): String
+setComment(String): void
+getParentHabitUniqueId(): String
+setParentUniqueId(String): void
+getTimeStamp(): String
+setTimeStamp(String): void
+getImageBitmapString(): String
+setImageBitmapString(String): void
+getLatlngString(): String
+setLatlngString(String): void
+getDateString():String
+setDateString(String):void
+parseFromDoc(Map<String, Object>): HabitEvent
+getDocument(): HashMap<String,String>
+getUniqueId():String
+stringToBitMap(String): Bitmap
+bitMapToString(Bitmap, boolean): String
+resizeBitmapIfTooLarge(Bitmap, maxHeightOrWidth): Bitmap
-supportMapFragm
-mapV
+showOn
+showOn
+hideM
+latlngToS
+latlngFrom
ditActivity
Firestore
ed: HabitEvent
ext: EditText
gToStore: String
EST_CODE: int
EST_CODE: int
EST_CODE: int
mageView
g: String
ler: Map
ndle): void
(View): void
n(View): void
e(View): void
on(View):void
dler(View): void
Handler(View): void
ton(View): void
, int, Intent): void
HabitEventDetailActivity
-habitTitle: TextView
-habitEventTime: TextView
-habitEventComment: TextView
-habitDetailDeleteButton: Button
-habitDetailEditButton: Button
-habitEventImageView: ImageView
-habitEventRecieved: HabitEvent
-db: FirebaseFirestore
-habitEventsAdapter: ArrayAdapter<String>
-mapController: Map
#OnCreate(Bundle):void
+onSupportNavigateUp():boolean
+DeleteHabitEvent(View):void
+EditHabitEvent(View): void
-Imag
-CAM
-GAL
-LOCA
+c
+c
+c
+on
+loca
+remo
+onC
#onActivityResult(int r
Model class: Contains
the Habit information
and related methods to
set or retrieve this info
Habit
-title: String
-reason: String
-startDate: String
-DaysToBeDone: String
-ownerUsername; String
-isPublic: boolean
-timeStamp: String
-habitEvents; ArrayList<HabitEvent>
+isPublic(): boolean
+setPublic(boolean): void
+getHabitTitle(): String
+getReason(): String
+getStartDate(): String
+getDaysToBeDone(): String
+setOwnerUsername(String): void
+getUniqueId(): String
+setDaysToBeDone(String): void
Allows user to edit a Habit
HabitEditActivity
-db: FirebaseFirestore
-habitPassedIn: Habit
-isPublic: boolean
#onCreate(Bundle): void
+SaveButton(View): void
+CancelButton(View): void
+isPublicButton(View): void
gets the habit and sets its information
in the text views and allows you to delete
and open the edit page
HabitDetailActivity
-habitDetailTitle: TextView
-HabitDetailStartDate: TextView
-habitDetailReason: TextView
-habitDetailDaysToDo: TextView
-habitDetailDeleteButton: Button
-habitDetailEditButton: Button
-habitRecieved: Habit
-db: FirebaseFirestore
-habitEventAdapter: ArrrayAdapter<String>
-numberOfDaysToDoThisWeek: float
-numberOfDaysDoneThisWeek: float
-habitOwnerUsername: String

contains utility methods for latitude longitude string conversion.
ic of the map showing the location on the map and animating it.
ap to be hidden the map from view when there is no location.





+setDaysToBeDone(String): void
+setReason(String): void
+setTitle(String): void
+getHabitEvents(): ArrayList<HabitEvent>
+clearHabitEvents(): void
+addToHabitEvents(HabitEvent): void
+isForToday(): boolean
+dateStringFromDatePicker(DatePicker): String
+parseFromDoc(Map<String, Object>): Habit
+getDocument(): HashMap<String, Object>
+getTimeStamp()

-habitOwnerUsername: String
-EDITHABIT_REQUEST_CODE: int
-isMyHabit: boolean

#onCreate(Bundle): void
+onSupportNavigateUp(): boolean
+DeleteHabit(View): void
+EditHabit(View): void
+onClickCompleted(View): void
-updateProgressBar(String, ArrayList<HabitEvent>
-datesOfThisWeek(String): ArrayList<String>

Gets info from user to create a new habit

AddHabitActivity

-db: FirebaseFirestore
- isPublic: boolean

#onCreate(Bundle): void
+SaveButton(View): void
+CancelButton(View): void
+isPublicButton(View): void

We are using the observe in order to attach eve listeners to certain fire documents and when data in the database chan any other activity, these o react and call the event function that updates ou this pattern was helpful t bugs that have stale c that would be outdated updates in the databa

Shows the list of all the habits and the user's information

ProfileActivity

-profileListView: ListView
-profileAllHabitsList: ArrayList<Habit>
-profileAllHabitsTitleList: ArrayList<String>
-profile_habitAdapter: ArrayAdapter<String>
-db: FirebaseFirestore
-isMyProfile: boolean
-username:String
-AtoZ:boolean
-EarlyToLate:boolean

#onCreate(Bundle):void
+addHabitButtonClickHandler(View):void
+FeedButton(View):void
+HomeButton(View):void
+Logout(View):void
+onBackPressed():void
+TimeOrder(View):void
+AlphabeticalOrder(View):void

HomeActivity

-habitList: ArrayList<H
-habitTitleList: ArrayList<
-habitAdapter: ArrayAdapte
-db: FirebaseFirest
TAG: String
+DeleteAndEdit: Str
profileCode: int
-username: String

#OnCreate(Bundle):
+newHabitButtonHandler(V
+FeedButton(View):
+onBackPressed():

Profile

-following: ArrayList<String>
-username: String
-password: String
-followRequests: ArrayList<String>

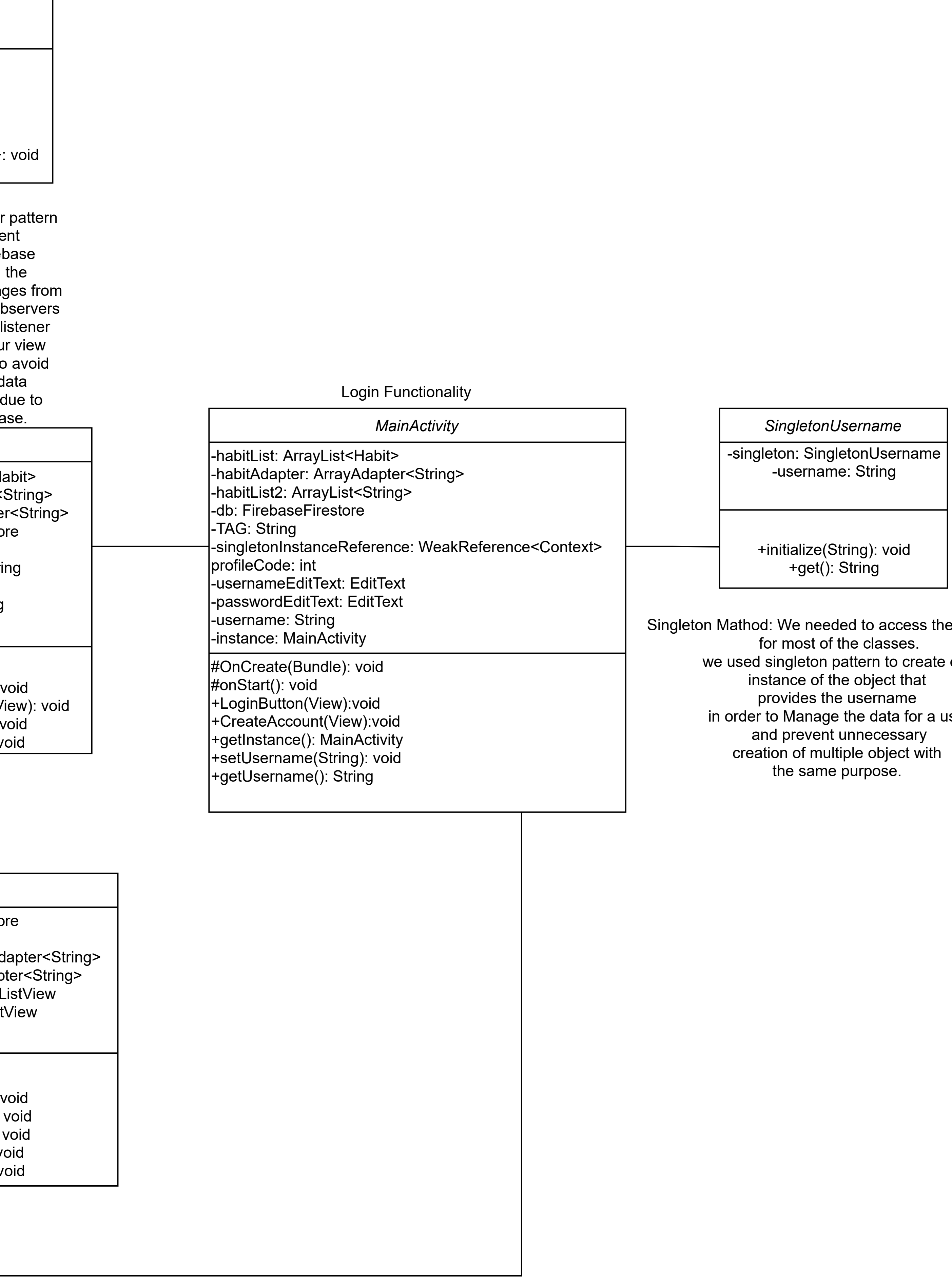
+setUsername(String): void
+setPassword(String): void
+getUsername(): String
+getPassword(): String
+getDocument(): HashMap<String, String>
+getFollowing(): ArrayList<String>
+addFollowing(String):void
+getFollowRequests():ArrayList<String>
+AddFollowRequests(String):void
+deleteFollower(String):void
+deleteFollowRequest(String):void
+clearFollowRequests(): void
+clearFollowers():void

FeedActivity

-db: FirebaseFirest
-profile: Profile
-followRequestAdapter: ArrayA
-followingAdapter: ArrayAda
-followRequestListView:
-followingListView: Lis

#OnCreate(Bundle):
+ProfileButton(View):
+HomeButton(View):
+SearchBar(View):
+onBackPressed():

Builder Pattern: Every time we use dialog boxes we use this to
allowing us to pass I the parameters that show
we can also use the callback functions to specify what t



r pattern
ent
ebase
the
nges from
bservers
listener
ur view
o avoid
data
due to
ase.

Singleton Mathod: We needed to access the username
for most of the classes.
we used singleton pattern to create one
instance of the object that
provides the username
in order to Manage the data for a user
and prevent unnecessary
creation of multiple object with
the same purpose.

make the dialog boxes and it allows the code to be organized
y what we want the user interface to look like
to do on success clicks, ok clicks, negative clicks, etc.