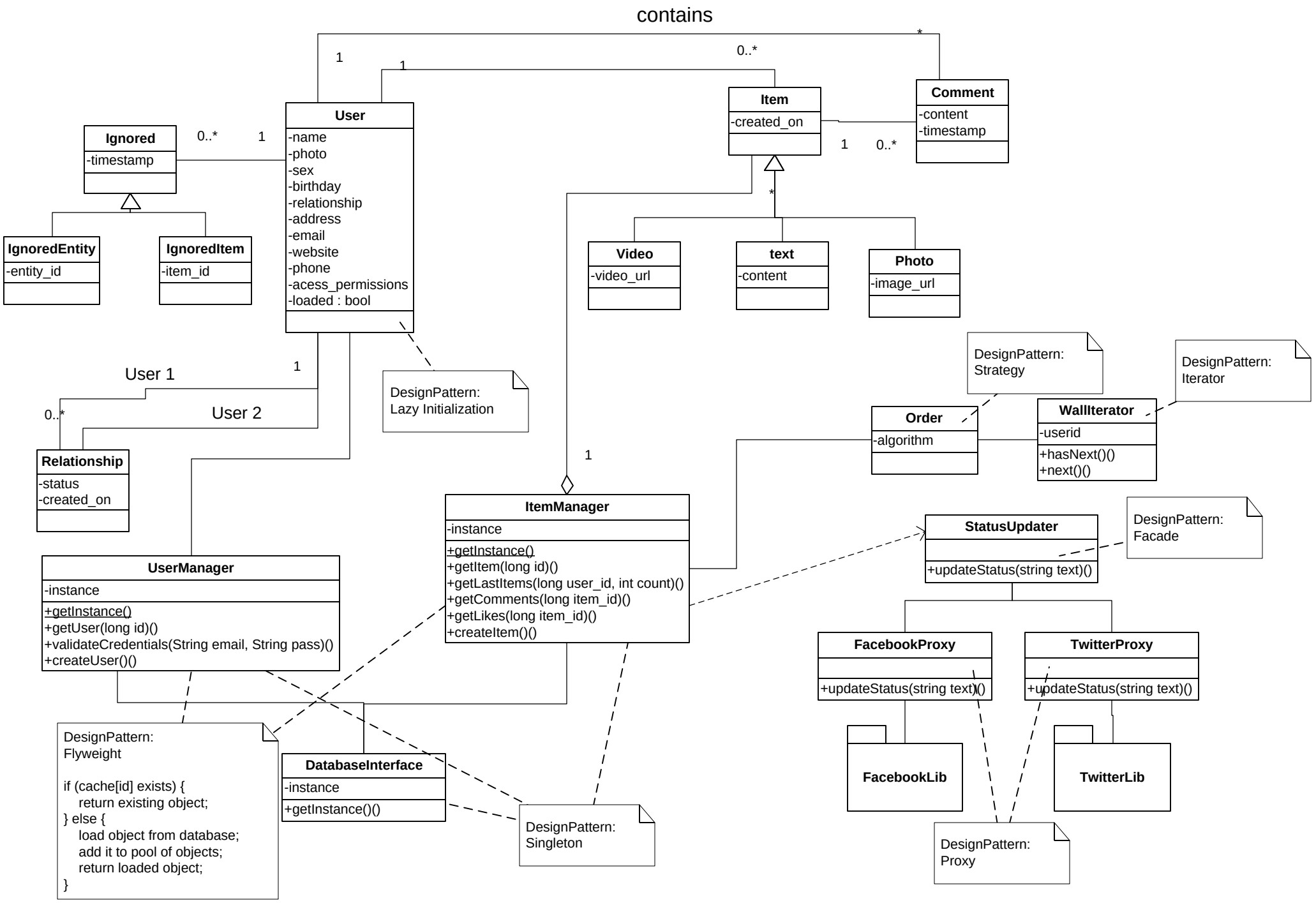
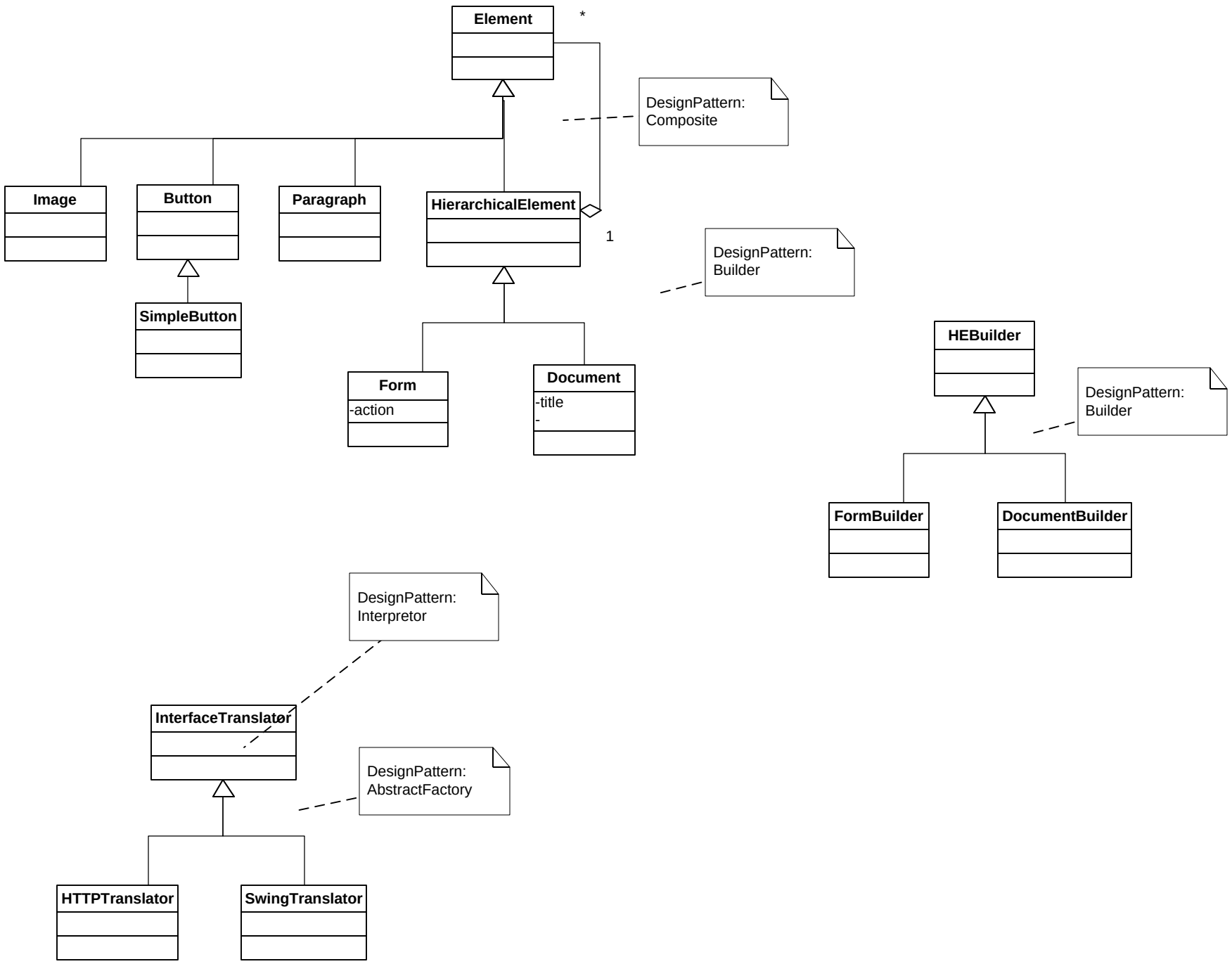

contains
1
1
0..*
*
Comment
-content
-timestamp
Item
-created_on
1
0..*
User
-name
-photo
-sex
-birthday
-relationship
-address
-email
-website
-phone
-acess_permissions
-loaded : bool
Ignored
-timestamp
0..*
1
IgnoredEntity
-entity_id
IgnoredItem
-item_id
*
Video
-video_url
text
-content
Photo
-image_url
User 1
1
0..*
User 2
Relationship
-status
-created_on
DesignPattern:
Lazy Initialization
DesignPattern:
Strategy
DesignPattern:
Iterator
Order
-algorithm
WallIterator
-userid
+hasNext()()
+next()()
1
ItemManager
-instance
+getInstance()
+getItem(long id)()
+getLastItems(long user_id, int count)()
+getComments(long item_id)()
+getLikes(long item_id)()
+createItem()()
StatusUpdater
+updateStatus(string text)()
DesignPattern:
Facade
UserManager
-instance
+getInstance()
+getUser(long id)()
+validateCredentials(String email, String pass)()
+createUser()()
FacebookProxy
+updateStatus(string text)()
TwitterProxy
+updateStatus(string text)()
DesignPattern:
Flyweight
if (cache[id] exists) {
return existing object;
} else {
load object from database;
add it to pool of objects;
return loaded object;
}
DatabaseInterface
-instance
+getInstance()()
DesignPattern:
Singleton
FacebookLib
TwitterLib
DesignPattern:
Proxy


Element
*
DesignPattern:
Composite
Image
Button
Paragraph
HierarchicalElement
1
DesignPattern:
Builder
SimpleButton
Form
-action
Document
-title
-
HEBuilder
DesignPattern:
Builder
FormBuilder
DocumentBuilder
DesignPattern:
Interpretor
InterfaceTranslator
DesignPattern:
AbstractFactory
HTTPTranslator
SwingTranslator