Model: Main

«View» EssentialGUI
- cFacade: CFacade
- +changeGUI()
- +loadApp()

«Bean» BEvent
- -eventID: Integer
- -eventOrganizer: String
- -eventOrganizerID: Integer
- -name: String
- -province: String
- -city: String
- -address: String
- -musicGenre: String
- -time: String
- -date: String
- -image: File
- +getEventID(): Integer
- +getEventOrganizer(): String
- +getEventOrganizerID(): Integer
- +getEventName(): String
- +getEventProvince(): String
- +getEventCity(): String
- +getEventMusicGenre(): String
- +getEventTime(): String
- +getEventDate(): String
- +getEventImage(): File
- +getEventID(): Integer
- +setEventOrganizer(orgName: String)
- +setEventOrganizerID(orgID: Integer)
- +setEventName(name: String)
- +setEventProvince(province: String)
- +setEventCity(city: String)
- +setEventAddress(address: String)
- +setEventMusic(music: String)
- +setEventTime(time: String)
- +setEventDate(date: String)
- +setEventImage(img: File)

«View» ParticipateToEvent
- +getEventInfo(): BEvent
- +checkAvailableEvents()
- +confirmParticipation()

«Session» LoggedUser
- -username: String
- -password: String
- -userID: Integer
- -inputStream: SecureObjectInputStream
- -outputStream: ObjectOutputStream
- -userType: UserType
- -notifAll
- +getUserID(): Integer
- +getUsername(): String
- +getOutputStream(): ObjectOutputStream
- +getUserType(): UserType

«View» Login
- +checkLogin()

«Bean» BUserData
- -username: String
- -password: String
- -city: String
- +setUsername(username: String)
- +getUsername(): String
- +setPassword(password: String)
- +getPassword(): String
- +setCity(city: String)
- +getCity(): String

Facade
- +loginUser(user: BUserData)
- +participateToEvent(event: BEvent)
- +removeEventParticipation(event: BEvent)

«Control» LoginControl
- -LoggedUser: User
- +checkLogin(user: BUserData)

«Model» User
- -username: String
- -password: String
- -city: String
- +setUsername(username: String)
- +getUsername(): String
- +getCity(): String
- +setPassword(password: String)
- +getPassword(): String

«DAO» UserDAO
- +checkLoginInfo()
- +getLoggedUser()

«Control» NotificationControl
- +sendNotification(type: String, clientID: Integer, notProps: NotificationProperties, eventID: Integer, cityData: CityData, usrType: String): void
- +addNotification(notifiedIDs: List<Integer>, type: NotificationTypes, eventID: Integer)

«DAO» NotificationDAOJDBC
- +addNotification(notifiedIDs: List<Integer>, type: NotificationTypes, eventID: Integer)

NotificationDAO
- +addNotification(notifiedIDs: List<Integer>, type: NotificationTypes, eventID: Integer)

«DAO» NotificationDAOCSV
- +addNotification(notifiedIDs: List<Integer>, type: NotificationTypes, eventID: Integer)

«Model» Event
- -eventID: Integer
- -eventOrganizer: String
- -eventOrganizerID: Integer
- -name: String
- -province: String
- -city: String
- -address: String
- -musicGenre: String
- -time: String
- -date: String
- -image: File
- +getEventID(): Integer
- +getEventOrganizer(): String
- +getEventOrganizerID(): Integer
- +getEventName(): String
- +getEventProvince(): String
- +getEventCity(): String
- +getEventMusicGenre(): String
- +getEventTime(): String
- +getEventDate(): String
- +getEventImage(): File
- +getEventID(): Integer
- +setEventOrganizer(orgName: String)
- +setEventOrganizerID(orgID: Integer)
- +setEventName(name: String)
- +setEventProvince(province: String)
- +setEventCity(city: String)
- +setEventAddress(address: String)
- +setEventMusic(music: String)
- +setEventTime(time: String)
- +setEventDate(date: String)
- +setEventImage(img: File)

«Control» ManageEventControl
- +retrieveMyEvents(userType: UserType, city: String): List<BEvent>
- +participateToEvent(eventBean: BEvent)
- +removeEventParticipation(eventBean: BEvent)

«DAO» EventDAO
- +retrieveMyEvents(userID: Integer): List<Event>
- +joinUserToEvent(event: Event)
- +removeUserToEvent(event: Event)
- +checkPreviousParticipation(eventID: Integer)

«Model» NotificationProperties
- -notifierID: Integer
- -notificationID: Integer
- +setNotification(ID: Integer): void
- +setNotifierID(ID: Integer): void
- +getNotificationID(): Integer
- +getNotifierID(): Integer

«Model» CityData
- -city: String
- -newCity: String
- +setCity(city: String): void
- +setNewCity(newCity: String): void
- +getCity(): String
- +getNewCity(): String

NotificationFactory
- +createNotification()

«Model» ServerNotification
- -notificationType: NotificationTypes
- -city: String
- -newCity: String
- -clientID: Integer
- -eventID: Integer
- -userType: UserType

«Model» LocalNotification
- -notificationType: NotificationTypes
- -notifiedID: Integer
- -notifiedID: Integer
- -notifierID: Integer
- -notificationID

Notification
- +getNotificationType(): String
- +getNotifierID(): Integer
- +getNotifiedID(): Integer
- +getEventID(): Integer
- +getNewCity(): String
- +getUserType(): String
- +setNotificationType(type: String): void
- +setNotifierID(ID: String): void
- +setEventID(ID: Integer): void
- +setNotificationID(ID: Integer): void
- +setNotifiedID(ID: Integer): void
- +setNewCity(City: String): void
- +setUserType(type: String): void

SingletonDBSession
- +getConnection()
- +closeConnection()
- +getInstance()

Serializable

Relationship labels: +extend, +use, +call, +create, +realize, +extends, +1, +0..*

«metaClass»
UMLClassifier

«stereotype»
Realization

«stereotype»
Specification

«metaClass»
UMLAbstraction

«stereotype»
Derive

+computation: String

«stereotype»
Refine

«stereotype»
Trace

«metaClass»
UMLArtifact

«stereotype»
File

«stereotype»
Document

«stereotype»
Executable

«stereotype»
Library

«stereotype»
Script

«stereotype»
Source

«metaClass»
UMLClass

«stereotype»
Auxiliary

«stereotype»
Focus

«stereotype»
Metaclass

«stereotype»
ImplementationClass

«stereotype»
Type

«stereotype»
Utility

«metaClass»
UMLComponent

«stereotype»
Entity

«stereotype»
Implementation

«stereotype»
Process

«stereotype»
Service

«stereotype»
Subsystem

«stereotype»
Destroy

«metaClass»
UMLBehavioralFeature

«stereotype»
Create

«metaClass»
UMLDependency

«stereotype»
Call

«stereotype»
Instantiate

«stereotype»
Responsibility

«stereotype»
Send

«metaClass»
UMLPackage

«stereotype»
Framework

«stereotype»
ModelLibrary

UMLStandardProfile::Robustness Stereotypes

«metaClass»
UMLClass

«stereotype»
control

«stereotype»
entity

«stereotype»
boundary

«metaClass» **UMLComponent**	←	«stereotype» **BuildComponent**

«metaClass» **UMLModel**	←	«stereotype» **Metamodel**
	←	«stereotype» **SystemModel**