Class Diagram - Advertising

UI **Business Logic** **Persistence**

AddAdvertisingController

- advertisingFacade: AdvertisingFacade
- userFacade: UserFacade
- walletFacade: WalletFacade
- selectedImage : Image
- selectedFile : File

- initialize(): void
- openNavbar(): void
- closeNavbar(): void
- handleButtonNavbar(open: boolean): void
- goToProfile(): void
- goToMyAdvertisings(): void
- removeImage(): void
- handleButton(hasFile: boolean): void
- addImage(): void
- handleCreateAdvertising(): void
- openOrderBox(duration: int): void
- openPaymentBox(): void
- pay(): void

advertisingFacade

OtherUserAdComponent

- advertising: Advertising

- initialize(): void
- loadAdvertising(advertising: Advertising, refreshNews: Runnable): void
- openLink(): void

advertisingFacade

UserAdvertisingsController

- advertisingFacade: AdvertisingFacade
- userFacade: UserFacade

- initialize(): void
- goToProfile(): void
- goToAddAdvertising(): void
- openNavbar(): void
- closeNavbar(): void
- handleButtonNavbar(open: boolean): void
- loadAdvertisings(): void
- createAdDetails(advertising: Advertising): VBox
- openNewADPopup(): void

advertisingFacade

UserAdComponent

- advertisingFacade: AdvertisingFacade
- advertising: Advertising

- initialize(): void
- handleAdBox(): void
- handleConfirmDeleteBox(): void
- loadAdvertising(advertising: Advertising, refreshAds: Runnable): void
- handleDeleteAdvertising(): void

<<singleton>> AdvertisingFacade

- advertisingFacade: AdvertisingFacade
- currentAdvertising: Advertising

+ getInstance(): AdvertisingFacade
+ getCurrentAdvertising(): Advertising
+ createAdvertising(text: String, link: String, image: File, user: User, duration: int): Boolean
+ deleteAdvertising(advertisingId: int): Boolean
+ getAdvertisingByUser(user: User): List<Advertising>

currentAdvertising

1

Advertising

- text: String
- link: String
- user: User
- id: int
- image: byte[]
- begindate: LocalDate
- enddate: LocalDate

+ Advertising()
+ Advertising(user: User, begindate: LocalDate, enddate: LocalDate, text: String, link: String, image: byte[])
+ getId(): int
+ setId(id: int): void
+ getUser(): User
+ setUser(user: User): void
+ getBegindate(): LocalDate
+ setBegindate(begindate: LocalDate): void
+ getEnddate(): LocalDate
+ setEnddate(enddate: LocalDate): void
+ getText(): String
+ setText(text: String): void
+ getLink(): String
+ setLink(link: String): void
+ getImage(): byte[]
+ setImage(image: byte[]): void

<> <<singleton>> AbstractDAOFactory

- factory: AbstractDAOFactory

+ getFactory(): AbstractDAOFactory
+ getAdvertisingDAO(): AdvertisingDAO

returns

<> <<singleton>> AdvertisingDAO

- advertisingDAO: AdvertisingDAO

+ getAdvertisingDAO(): AdvertisingDAO
+ createAdvertising(text: String, link: String, image: File, user: User, duration: int): Boolean
+ deleteAdvertising(advertisingId: int): Boolean
+ getAdvertisingByUser(user: User): List<Advertising>

PostgreSQLDAOFactory

+ getConnection(): Connection

Create

PostgreSQLAdvertisingDAO

Create

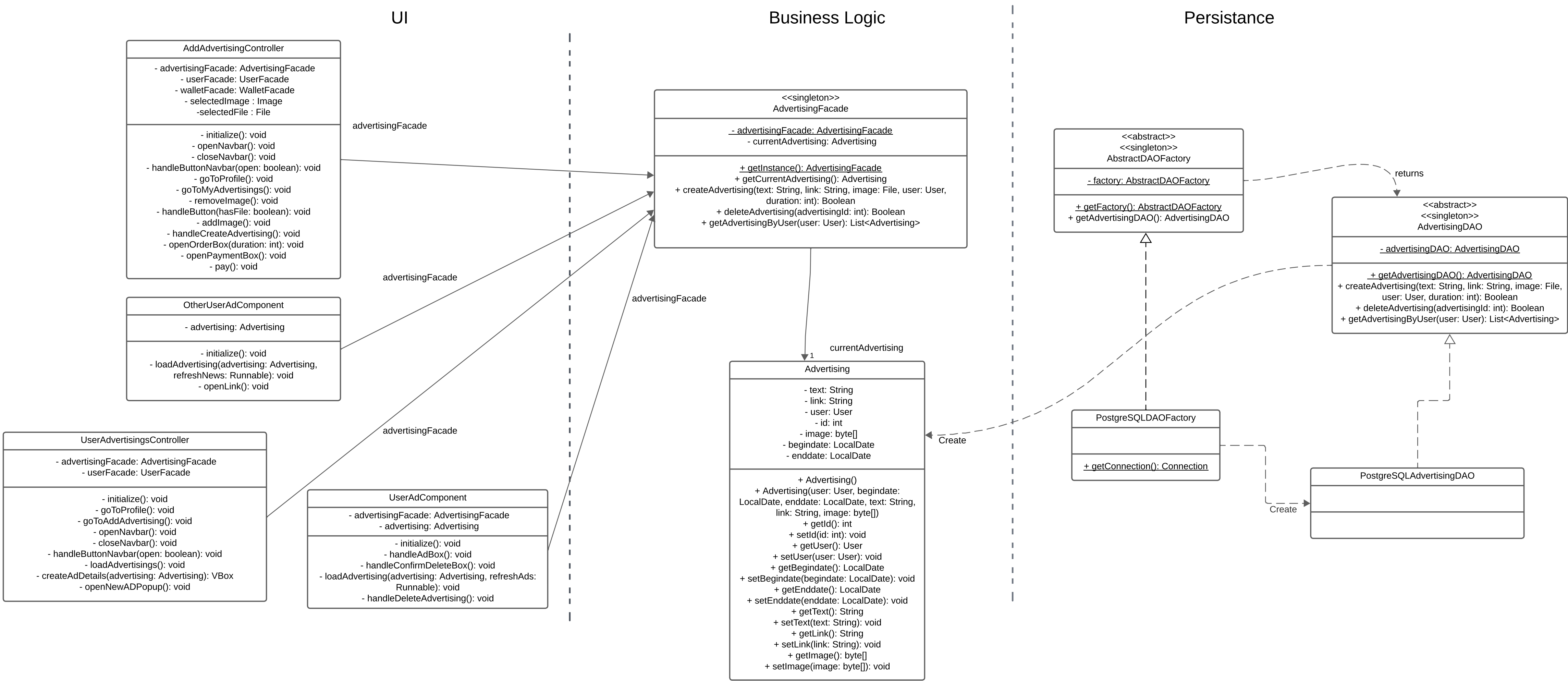# Sequence Diagram
user clicks on advertisings (for his own ads)