GUI.Menu

|  |
| --- |
| <<abstract>>  Menu |
| + getContr(): InterfaceController  + show(anchorPane: AnchorPane): void  + hide(anchorPane: AnchorPane): void  + errorMessage(label: Label, text: String): void  + successMessage(label: Label, text: String): void  + update(): void |

|  |
| --- |
| MenuLoggedIn |
| - contr: InterfaceControllerMenuLoggedIn  - messageLoggedIn: Label  - myCalendarsTab, sharedWithMeTab, historyTab, adminSettingsTab, settingsTab, saveButton, logOutButton: Button  - myCalendarsMenu, sharedWithMeMenu, historyMenu, adminSettingsMenu, settingsMenu: AnchorPane  - newUsername: TextField  - newPassword, confirmPassword: PasswordField  - messageChangeUsername, messageChangePassword, messageConfirmPassword: Label  - usersList: ListView<AnchorPane>  - histroyTable: TableView<EntryHistory>  - historyDate, historyTime, historyUser, historyAction: TableColumn<EntryHistory, String>  - myCalendarsTable: TableView<EntryCalendar>  - myCalendarsName, myCalendarsDescription: TableColumn<EntryCalendar, String>  - sharedWityhMeTable: TableView<EntryCalendar>  - sharedWithMeName, sharedWithMeDescription, sharedWithMeAccess: TableColumn<EntryCalendar, String>  - popupShadowLeft, popupShadowRight, popup: AnchorPane  - calendarName: TextField  - calendarDescription: TextArea  - newCalendar: boolean |
| + getContr(): InterfaceController  + update(): void. + selectCalendarsTab(): void  + selectCalendarstab(): void + selectTab(): void  + selectHistoryTab(): void + selectAdminTab(): void  + selectSettingsTab(): void + logOut(): void  + save(): void + refreshSettingTab(): void  + changeUsername(): void + changePassword(): void  + deleteMyAccount(): void + ban(): void  + unban(): void + deleteAccount(): void  + refreshHistoryTab() :void + refreshMyCalendarsTab(): void  + openCalendar(): void + editCalendarPreperties(): void  + shareCalendar(): void + deleteCalendar(): void  + refreshSharedWithMeTab(): void + removeMyself(): void  + newCalendar(): void + confirmCalendarNameDescription: void  + showPopup(newCalendar: boolean): void + exitPopup(): void  - configureTables(): void. - refreshSaveButton(): void  - showTabsBasedOnUser(): void  - toggleTab(selected: Button, selectedMenu: AnchorPane): void  - refreshAdminSettingTab: void |

|  |
| --- |
| MenuLoggedOut |
| - contr: InterfaceControllerMenuLoggedOut  - subMenu1, subMenu2: AnchorPane  - confirm: Button  - menuMessage: Label  - username: TextField  - password: PasswordField |
| + getContr(): InterfaceController  + update(): void  + getSubMenu1(): void  + getSubMenu2(menuType: String): void  + getSubMenuSignIn(): void  + getSubMenuSignUp(): void  + confirm(): void  + exit(): void |

|  |
| --- |
| <<Interface>>  InterfaceControllerMenuLoggedIn |
| + logOut(): void  + save(): void  + changeUsername(newUsername: String, messageChanegUser: Label, messageLoggedIn: Label): void  + changePassword(newPassword: String, messageChangePassword: Label): void  + deleteMyAccount(confirmPassword: String, messageConfirmedPassword: Label): void  + buildAdminSettingsList(usersList: ListView<anchorPane>): void  + ban(usersList: ListView<AnchorPane>): void  + unban(usersList: ListView<AnchorPane>): void  + deleteAccount(usersList: ListView<AnchorPane>): void  + buildHistoryTable(historyTable: TableView<EntryHistory>): void  + buildMyCalendarsTabel(myCalendarsTable: TableView<EntrtCalendar>): void  + buildSharedWithMeTable(SharedWithMeTable: TableView<EntryCalendar>): void  + shareCalendar(myCalendarsTable: TableView<EntryCalendar>): void  + deleteCalendar(myCalendarsTabel: TableView<EntryCalendar>): void  + removeMyself(sharedWithMeTable: TableView<EntryCalendar>): void  + newCalendar(name: String, description: String): void  + editProperties(myCalendarsTable: TableView<EntryCalendar>): void  + setCalendarNameDescriptionText(myCalendarsTabel: TableView<EntryCalendar>, calendarName: TextField, calendarDescription: TextArea) |

|  |
| --- |
| <<Interface>>  InterfaceController |
| + getCals(): CalendarManager  + getPlans(): PlanManager  + getLabs(): LabelManager  + getlogs(): LoggingManager  + getUsers(): UserManager  + getHists(): HistoryManager  + getS(): InterfaceSaver  + update(): void  + setMenu(menu: Menu): void  + setData(stage: Stage, cals: CalendarManager, plans: PlanManager, labs: LabelManager, logs: LoggingManager, users: UserManager, hists: HistoryManager, s: InterfaceSaver): void |

GUI.Controller

|  |
| --- |
| <<Interface>>  InterfaceControllerMenuLoggedOut |
| + logIn(username: TextField, password: PasswordField, menuMessage: Label): void  + newUser(username: TextField, password: PasswordField, menuMessage: Label): void  + exit(): void |

|  |
| --- |
| <<Interface>>  InterfaceController |
| + getCals(): CalendarManager  + getPlans: PlanManager  + getLabs: LabelManager  + getlogs: LoggingManager  + getUsers: UserManager  + getHists: HistoryManager  + getS(): InterfaceSaver  + update: void  + setMenu(menu: Menu): void  + setData(stage: Stage, cals: CalendarManager, plans: PlanManager, labs: LabelManager, logs: LoggingManager, users: UserManager, hists: HistoryManager, s: InterfaceSaver): void |

|  |
| --- |
| <<Abstract>>  Controller |
| - stage: Stage  - cals: CalendarManager  - plans: PlanManager  - labs: LabelManager  - logs: LoggingManager  - users: UserManager  - hists: HistoryManager  - s: InterfaceSaver |
| + getStage(): Stage  + getCals(): CalendarManager  + getPlans(): PlanManager  + getLabs(): LabelManager  + getLogs(): LoggingManager  + getUsers(): UserManager  + getHists(): HistoryManager  + getS(): InterfaceSaver  + setData(stage: Stage, cals: CalendarManager, plans: PlanManager, labs: LabelManager, logs: LoggingManager, users: UserManager, hists: HistoryManager, s: InterfaceSaver)  + *setMenu(menu: Menu): void*  + *update(): void*  + switchMenu(Stage: Stage, menuFileName: String): void |

|  |
| --- |
| <<Abstract>>  Controller |

|  |
| --- |
| <<Interface>>  InterfaceControllerMenuLoggedIn |

|  |
| --- |
| ControllerMenuLoggedIn |
| - menu: MenuLoggedIn  - mainTab: HandlerMainTab  - settings: HandlerSettings  - adminSettings: HandlerAdminSettings  - tables: HandlerTables  - calendars: HandlerCalendars |
| + getMenu(): MenuLoggedIn  + setMenu(menu: Menu): void  + update(): void  + logOut(): void  + save(): void  + changeUsername(newUsername: String, messageChanegUser: Label, messageLoggedIn: Label): void  + changePassword(newPassword: String, messageChangePassword: Label): void  + deleteMyAccount(confirmPassword: String, messageConfirmedPassword: Label): void  + buildAdminSettingsList(usersList: ListView<anchorPane>): void  + ban(usersList: ListView<AnchorPane>): void  + unban(usersList: ListView<AnchorPane>): void  + deleteAccount(usersList: ListView<AnchorPane>): void  + buildHistoryTable(historyTable: TableView<EntryHistory>): void  + buildMyCalendarsTabel(myCalendarsTable: TableView<EntrtCalendar>): void  + buildSharedWithMeTable(SharedWithMeTable: TableView<EntryCalendar>): void  + shareCalendar(myCalendarsTable: TableView<EntryCalendar>): void  + deleteCalendar(myCalendarsTabel: TableView<EntryCalendar>): void  + removeMyself(sharedWithMeTable: TableView<EntryCalendar>): void  + newCalendar(name: String, description: String): void  + editProperties(myCalendarsTable: TableView<EntryCalendar>): void  + setCalendarNameDescriptionText(myCalendarsTabel: TableView<EntryCalendar>, calendarName: TextField, calendarDescription: TextArea) |

|  |
| --- |
| <<abstract>>  Controller |

|  |
| --- |
| <<Interface>>  InterfaceControllerMenuLoggedOut |

|  |
| --- |
| ControllerMenuLoggedOut |
| - menu: MenuLoggedOut |
| + getMenu(): MenuLoggedOut  + setMenu(menu: Menu): void  + update(): void  + logIn(username: TextField, password: PasswordField, menuMessage: Label): void  + newUser(username: TextField, password: PasswordField, menuMessage: Label): void  + exit(): void |

|  |
| --- |
| HandlerMainTab |
| - cals: CalendarManager  - hists: HistoryManager  - labs: LabelManager  - users: UserManager  - plans: PlanManager  - logs: LoggingManager |
| + HandlerMainTab(cals: CalendarManager, hists l HistoryManager, labs: LabelManager, users: UserManager, plans: PlanManager, logs: LoggingManager)  + logOut(): void  + save(): void |

|  |
| --- |
| HandlerSettings |
| - users: UserManager  - logs: LoggingManager  - cals: CalendarManager  - hists: HistoryManager  - labs: LabelManager  - plans: PlanManager  - menu: Menu |
| + HandlerSetting(users: UserManager, logs: LoggingManager, hists: HistoryManager, menu: Menu, cals: CalendarManager, plans: PlanManager, labs: LabelManager)  + changeUsername(newUsername: String, messageChangeUsername: Label, messageLoggedIn: Label): boolean  + changePassword(newPassword: String, messageChangePassword: Label): boolean  + deleteMyAccount(confirmPassword: String, messageConfirmPassword: Label): boolean |

|  |
| --- |
| HandlerAdminSettings |
| - users: UserManager  - hists: HistoryManager  - logs: LoggingManager |
| + HandlerAdminSetting(users: UserManager, hists: HistoryManager, logs: LoggingManager)  + buildAdminSettingsList(usersList: ListView<AnchorPane>): void  + getUserSelection(usersList: ListView<AnchorPane>, ban: boolean)  + toggleBan(usersList: ListView<AnchorPane>, ban: boolean): void  + deleteAccount(usersList: ListView<AnchorPane>): void  - getUserSelection(usersList: ListView<AnchorPane>): List<String> |

|  |
| --- |
| HandlerCalendars |
| - cals: CalendarManager  - plans: PlanManager  - labs: LabelManager  - users: UserManager  - logs: LoggingManager  - hists: HistoryManager |
| + HandleCalendars(cals: CalendarManager, plans: PlanManager, labs: LabelManager, users: UserManager, logs: LoggingManager, hists: HistoryManager)  + shareCalendar(myCalendarsTable: TableView<EntryCalendar> ): void  + deleteCalendar(myCalendarsTable: TableView<EntryCalendar>): void  + removeMyself(sharedWithMeTable: TableView<EntryCalendars>) : void  + newCalendar(name: String, description: String): void  + editProperties(myCalendarsTabele: TableView<EntryCalendar>, name: String, descriptyion: String): void  + setCalendarNameDescriptionText(myCalendarsTable: TableView<EntryCalendars>, calendarName: TextField, calendarDescription: TextArea): void  - getSelectedCalendarEntry(calTable: TableView<EntryCalendar>): void |

|  |
| --- |
| HandlerTables |
| - users: UserManager  - logs: LoggingManager  - cals: CalendarManager  - hists: HistoryManager |
| + public HandlerTables(users: UserManager, logs: LoggingManager, cals: CalendarManager, hists: HistoryManager)  + buildMyCalendarsTable(myCalendarsTable: TableView<EntryCalendar>): void  + buildSharedWithMeTable(sharedWithMeTable: TableView<EntryCalendar>): void  + buildHistoryTable(historyTable: TableView<EntryHistory>): void |

GUI.ViewModel

|  |
| --- |
| <<Interface>>  BuilderHistoryTable |
| + buildDates(): void  + buildTimes(): void  + buildActions(): void  + buildHistoryEntries(): void  + getResult(): ObservableList<EntryHistory> |

|  |
| --- |
| <<Interface>>  BuilderUsersList |
| + buildUsernames(): void  + buildBanStatues(): void  + buildLabels(): void  + buildImafeViews(): void  + buildAnchorPanes(): void  + getResult(): List<AnchorPane> |

|  |
| --- |
| <<Interface>>  BuilderCalendarTable |
| + buildOwnedCalendarIDs(): void  + buildSharedCalendarIDs(): void  + buildCalendarEntries(): void  + getResult(): ObservableList<EntryCalendar> |

|  |
| --- |
| ConcreteBuilderCalendarTable |
| - calendarIDs: Map<String, List<String>>  - calendarEntries: List<EntryCalendar>  - users: UserManager  - logs: LoggingManager  - cals: CalendarManager |
| + ConcreteBuilderCalendarTable(users: UserManager, logs: LoggingManager, cals: CalendarManager)  + buildOwnedCalendarIDs(): void  + buildSharedCalendarIDs(): void  + buildCalendarEntries(): void  + getResult(): ObservableList<EntryCalendar>  - getCalendarEntries(calendarIDs: List<String>, access: String): List<EntryCalendar> |

|  |
| --- |
| ConcreteBuilderUsersList |
| - i: List<ImageView>  - l: List<Label>  - a: List<AnchorPane>  - usernames: List<Byte>  - banStatuses: List<Byte>  - users: UserManager |
| + ConcreteBuilderUsersList(users: UserManager)  + buildUsernames(): void  + buildBanStatues(): void  + buildLabels(): void  + buildImageViews(): void  + buildAnchorPanes(): void  + getResult(): List<AnchorPane> |

|  |
| --- |
| ConcreteBuilderHistoryTable |
| - dates, times, actions: List<String>  - entries: List<EntryHistory>  - hists: HistoryManager  - username: String |
| + ConcreteBuilderHistoryTable(hists: HistoryManager, username: String)  + buildDates(): void  + buildTimes(): void  + buildActions(): void  + buildHistoryEntries(): void  + getResult(): ObservableList<EntryHistory> |

|  |
| --- |
| DirectorCalendarTable |
| - builder: BuilderCalendarTable |
| + DirectorCalendarTable(builder: BuilderCalendarTable)  + make(type: String): ObservableList<EntryCalendar> |

|  |
| --- |
| EntryCalendar |
| - name: String  - description: String  - access: String  - id: String |
| + EntryCalendar(name: String, description: String, access: String, id: String)  + getName(): String  + getDescription(): String  + getAccess(): String  + getId(): String |

|  |
| --- |
| EntryHistory |
| - date: String  - time: String  - user: String  - action: String |
| + EntryHistory(date: String, time: String, user: String, action: String)  + getDate(): String  + getTime(): String  + getUser(): String  + getAction: String |