Presentation
View
LoginView
- svm: LoginViewModel
+ Login_Click(sender object, e RoutedEventArgs): void
+ Register_Click(sender object, e RoutedEventArgs): void
SlideTaskView
- slvm: SlideTaskViewModel
+ UpdateTaskView(user: UserModel, selectedTask: TaskModel)
+ Update_Click(sender object, e RoutedEventArgs): void
- Back_Click(sender object, e RoutedEventArgs): void
AddTaskView
- atvm: AddTaskViewModel
+ AddTaskView(user: UserModel)
+ Add_Click(sender object, e RoutedEventArgs): void
- Back_Click(sender object, e RoutedEventArgs): void
BoardView
- bvm: BoardViewModel
- user: UserModel
+ BoardView(u: UserModel)
- AddTask_Click(sender object, e RoutedEventArgs): void
- DeleteTask_Click(sender object, e RoutedEventArgs): void
- AddColumn_Click(sender object, e RoutedEventArgs): void
- UpdateTask_Click(sender object, e RoutedEventArgs): void
- ResetFilter_Click(sender object, e RoutedEventArgs): void
- Filter_Click(sender object, e RoutedEventArgs): void
- RemoveColumn_Click(sender object, e RoutedEventArgs): void
- MoveRight_Click(sender object, e RoutedEventArgs): void
- MoveColumnLeft_Click(sender object, e RoutedEventArgs): void
- AdvanceTask_Click(sender object, e RoutedEventArgs): void
- SortTasks_Click(sender object, e RoutedEventArgs): void
- Logout_Click(sender object, e RoutedEventArgs): void
- AssignTasks_Click(sender object, e RoutedEventArgs): void
AddColumnView
- acvm: AddColumnViewModel
+ AddColumnView(user: UserModel)
- Add_Click(sender object, e RoutedEventArgs): void
- Back_Click(sender object, e RoutedEventArgs): void
AssignTaskView
- atvm: AssignTaskViewModel
+ AssignTaskView(user: UserModel, selectedTask: TaskModel)
- AssignTask_Click(sender object, e RoutedEventArgs): void
- Back_Click(sender object, e RoutedEventArgs): void
ViewModel
AddTaskView
- Controller: BackendController
- user: UserModel
- Board: BoardModel
- _title: string
- _description: string
- _duedate: DateTime
- message: string
+ AddTask(title): UserModel
+ Add(): void
+ Back(): void
SlideTaskViewModel
- Controller: BackendController
- user: UserModel
- Board: BoardModel
- selectedTask: TaskModel
- _title: string
- _description: string
- _emailassignee: string
- _newduedate: DateTime
- _duedate: DateTime
- message: string
+ UpdateTaskViewModel(user: UserModel, selectedTask: TaskModel)
+ Update(): void
+ Back(): UserModel
LoginViewModel
- Controller: BackendController
- _email: string
- _password: string
- _nickname: string
- _emailhost: string
- message: string
+ Login(): UserModel
+ Register(): void
BoardViewModel
- Controller: BackendController
- user: UserModel
- Board: BoardModel
- _selectedColumn: ColumnModel
- _selectedTask: TaskModel
- _newsortmanyTask: bool
- _search: string
- _hast: Visibility
+ BoardViewModel(user: UserModel)
+ DeleteTask(): void
+ AdvanceTask(): void
+ RemoveColumn(): void
+ MoveRight(): void
+ MoveColumnLeft(): void
+ Logout(): void
+ SortTasks(): void
+ Filter(): void
+ ResetFilter(): void
+ Unsignedt(): bool
+ Unsid(): bool
NotifiableObject : ReadOnlyPropertyChanged
+ PropertyChanged: PropertyChangedEventHandler
+ RaisePropertyChanged(propertyname): void
AddColumnViewModel
- Controller: BackendController
- user: UserModel
- Board: BoardModel
- _Name: string
- _Ordinal: int
- message: string
+ AddColumnView(u: UserModel)
+ Add(): void
+ Back(): void
AssignTaskViewModel
- Controller: BackendController
- user: UserModel
- Board: BoardModel
- selectedTask: TaskModel
- _emailassignee: string
- message: string
+ AssignTaskViewModel(user: UserModel, selectedTask: TaskModel)
+ Assign(): void
+ Back(): UserModel
ViewModel
ColumnModel
- _id: int
- _Title: string
- _Description: string
- _emailassignee: string
- _Duedate: DateTime
- _CreatorTime: DateTime
- _Ismoke: bool
- _Textbackground: SolidColorBrush
- _BorderColor: SolidColorBrush
+ TaskModel(controller: BackendController, email: string, id: int, title: string, description: string, emailassignee: string, duedate: DateTime, creatortime: DateTime, ordinal: int)
+ TaskModel(controller: BackendController, email: string, id: int, title: string, description: string, emailassignee: string, duedate: DateTime, creatortime: DateTime)
+ UpdateduDate: DateTime, title: string, description: string): void
0..n
ColumnModel
- Email: string
- Tasks: ObservableCollection<TaskModel>
- _name: string
- _ord: int
+ ColumnModel(controller: BackendController, name: string, bord: int, basis List<TaskModel>, ordinal: int, email: string)
+ ColumnModel(controller: BackendController, name: string, bord: int, basis List<TaskModel>, email: string)
+ ColumnModel(controller: BackendController, column: ColumnModel, email: string)
+ Add(task: TaskModel): void
+ DeleteTaskcontrolerTask(): TaskModel: void
+ UpdateTask(id: int, duedate DateTime, title string, description string): void
+ MoveLeft(): void
+ MoveRight(): void
+ ResetFilter(): void
+ Filterswarning_word(): void
+ SortTasks(): void
0..*
BoardModel
- Email: string
- Columns: ObservableCollection<ColumnModel>
- _hostemail: string
+ BoardModel(controller: BackendController, columns: ObservableCollection<string>, emailhost: string, bord: string, email: string)
+ BoardModel(controller: BackendController, user: UserModel)
+ AdvanceTask(user: UserModel, selectedTask: TaskModel): void
+ UpdateTask(title: duedate DateTime, title string, description string): void
+ AssignTaskAssignee(string: text, TaskModel): void
+ DeleteTask(user: UserModel, selectedTask: TaskModel): void
+ AddTask(user: UserModel, duedate DateTime, title string, description string): void
+ MoveColumnRight(columnOrdinal: int): void
+ MoveColumnLeft(columnOrdinal: int): void
+ RemoveColumn(columnordinal: int): void
+ AddColumn(user: UserModel, name string, ordinal: int): void
+ ResetFilter(): void
+ Filterswarning_word: string): void
+ SortTasks(): void
UserModel
- _email: string
- _nickname: string
+ UserModel(controller: BackendController, email: string, nickname: string)
+ GetBoard(): BoardModel
NotifiableModelObject
- Controller: BackendController
NotifiableModelObject(controller: BackendController)
הערה חשובה הקשרים
האדומים אומרים שגל מה
שנמצא בתקיה מכיר את
המחלקה שמתאבר אליה
BackendController
- Service: IService
+ BackendController(service: IService)
+ BackendController()
+ Register(email: string, password: string, nickname: string): void
+ Register(email: string, password: string, nickname: string, hostemail: string): void
+ Login(email: string, password: string): UserModel
+ GetBoard(email: string): BoardModel
+ GetColumns(email: string, ordinal: int): ColumnModel
+ AddColumn(user: UserModel, name: string, ordinal: int): ColumnModel
+ RemoveColumn(email: string, columnordinal: int): void
+ MoveColumnRight(email: string, ColumnOrdinal: int): void
+ MoveColumnLeft(email: string, ColumnOrdinal: int): void
+ ChangeName(email: string, ColumnOrdinal: int, value: string): void
+ ChangeNickname(email: string, ColumnOrdinal: int, value: string): void
+ AdvanceTask(user: UserModel, duedate DateTime, title string, description string): TaskModel
+ DeleteTask(user: UserModel, selectedTask: TaskModel): void
+ UpdateTask(selectedTask: TaskModel, duedate DateTime, title string, description string): void
+ AssignTask(email: string, columnordinal: int, tasksid: int, emailAssignee: string): void
+ AdvanceTask(user: UserModel, selectedTask: TaskModel): void
Service Layer
<<Interface>>
IService
+ LoadGlobal(): Response
+ DeleteData(): Response
+ Register(email: string, password: string, nickname: string): Response
+ Register(email: string, password: string, nickname: string, emailhost: string): Response
+ Login(email: string, password: string): Response<User>
+ Logout(email: string): Response
+ GetBoard(email: string): Response<Board>
+ LimitColumnTasks(email: string, int columnOrdinal, int limit): Response
+ AssignTask(email: string, int columnOrdinal, int taskid, int emailassignee): Response
+ AddTask(email: string, int columnOrdinal, int taskid, int emailassignee, string emailhost): Response
+ DeleteTask(email: string, int columnOrdinal, int taskid): Response
+ UpdateTask(email: string, int columnOrdinal, int taskid, int duedate, string title, string description): Response
+ AddTask(email: string, int columnOrdinal, int taskid, int duedate, string title, string description): Response
+ DeleteTask(email: string, int columnOrdinal, int taskid): Response
+ UpdateTaskDueDate(email: string, int columnOrdinal, int taskid, int duedate): Response
+ UpdateTaskTitle(email: string, int columnOrdinal, int taskid, string title): Response
+ UpdateTaskDescription(email: string, int columnOrdinal, int taskid, string description): Response
+ AdvanceTask(email: string, int columnOrdinal, int taskid): Response
+ UpdateTaskDescription(email: string, int columnOrdinal, int taskid, int duedate, string title, string description): Response
+ AddTask(email: string, int columnOrdinal, int taskid, int duedate, DateTime, active bool): Response
+ UpdateTaskTitle(email: string, int columnOrdinal, int taskid, int title, string, active bool): Response
+ UpdateTaskDescription(email: string, int columnOrdinal, int taskid, int, description: string, active bool): Response
+ AdvanceTask(email: string, int columnOrdinal, int taskid, int, active bool): Response
+ GetColumn(email: string, int columnOrdinal, int taskid): Response
+ GetColumn(email: string, columnOrdinal, int, active bool): Response<Column>
+ GetColumn(email: string, columnOrdinal, int, active bool): Response<Column>
+ ChangeColumnName(email: string, columnOrdinal: int, newname: string): Response
+ RemoveColumn(email: string, columnOrdinal: int): Response
+ AddColumn(email: string, columnOrdinal: int, name: string): Response
+ MoveColumnRight(email: string, columnOrdinal: int): Response<Column>
+ MoveColumnLeft(email: string, columnOrdinal: int): Response<Column>
Response
+ ErrorMessage: string
+ ErrorOccured: bool
+ Responsecmy: string
ResponseT<T>
+ Value T:
+ Responsecmy: string
+ Response(value: T)
+ Response(value: T,msg:)
Objects
Column
+ Tasks: ObservableCollection<Task>
+ Name: string
+ Limit: int
+ ColumnTasks: ReadOnlyCollection<Task>(name: string, bord: int)
1
0..*
Task
- id: int
+ CreatorTime: DateTime
+ Title: string
+ Description: string
+ DueDate: DateTime
+ emailassignee: string
+ Task(id: int, creatortime: DateTime, title: string, description:string, emailassignee: string)
1
0..*
Board
+ ColumnsNames: ReadOnly<Collection<string>
+ emailowner: string
+ Board(columnsNames: ReadOnlyCollection<string>, emailCreator: string)
User
+ Email: string
+ Nickname: string
+ Usersmail:string,nickname: string)
Service
- bs: BoardService
- us: UserService
+ BoardControl()
+ LoadGlobal(): Response
+ DeleteData(): Response
+ Register(email: string, password: string, nickname: string): Response
+ Register(email: string, password: string, nickname: string, emailhost: string): Response
+ Login(email: string, password: string): Response<User>
+ Logout(email: string): Response
+ GetBoard(email: string): Response
+ LimitColumnTasks(email: string, columnOrdinal, int limit): Response
+ AssignTask(email: string, columnOrdinal, int taskid, int emailassignee, string active bool): Response
+ AddTask(email: string, title string, description string, duedate DateTime, active bool): Task
+ DeleteTask(email: string, columnOrdinal, int taskid, int, active bool): void
+ UpdateTaskDueDate(email: string, columnOrdinal, int taskid, int, duedate, DateTime, active
+ UpdateTaskTitle(email: string, columnOrdinal, int taskid, int title, string, active bool): void
+ UpdateTaskDescription(email: string, columnOrdinal, int taskid, int, description: string, active
+ AdvanceTask(email: string, columnOrdinal, int taskid, int, active bool): void
+ GetColumn(email: string, columnName: string, active bool): Column
+ GetColumn(email: string, columnOrdinal, int, active bool): Column
+ ChangeColumnName(email: string, columnOrdinal, int, newname: string, active bool): void
+ LimitColumnTasks(email: string, columnOrdinal, int, limit, int, active bool): void
+ RemoveColumn(email: string, columnOrdinal, int, active bool): void
+ AddColumn(email: string, columnOrdinal, int, Name: string, active bool): Column
+ MoveColumnRight(email: string, columnOrdinal, int, active bool): Column
+ MoveColumnLeft(email: string, columnOrdinal, int, active bool): Column
+ AddBoard: email: string)
+ AddUserToBoard(email: string, password: string, nickname: string, emailhost: string): void
EventService
- c: EventControl
+ LoadGlobal(): Response
+ DeleteData(): Response
+ GetBoard(email: string, active bool): Response<Board>
+ LimitColumnTasks(email: string, columnOrdinal, int, limit int, active bool): Response
+ AssignTask(email: string, columnOrdinal, int taskid, int, emailassignee, string, active bool): Response
+ AddTask(email: string, title string, description: string, duedate DateTime, active bool): Response<Task>
+ DeleteTask(email: string, columnOrdinal, int taskid, int, active bool): Response
+ UpdateTaskDueDate(email: string, columnOrdinal, int taskid, int, duedate, DateTime, active bool): Response
+ UpdateTaskTitle(email: string, columnOrdinal, int taskid, int title, string, active bool): Response
+ UpdateTaskDescription(email: string, columnOrdinal, int taskid, int, description: string, active bool): Response
+ AdvanceTask(email: string, columnOrdinal, int taskid, int, active bool): Response
+ GetColumn(email: string, columnOrdinal, int, active bool): Response<Column>
+ ChangeColumnName(email: string, columnOrdinal, int, newname: string, active bool): Response
+ RemoveColumn(email: string, columnOrdinal, int, active bool): Response
+ AddColumn(email: string, columnOrdinal, int, name: string, active bool): Response<Column>
+ MoveColumnRight(email: string, columnOrdinal, int, active bool): Response<Column>
+ MoveColumnLeft(email: string, columnOrdinal, int, active bool): Response<Column>
+ AddBoard(email: string): Response
+ AddUserToBoard(email: string, password: string, nickname: string, emailhost: string): void
UserService
- c: UserControl
+ UserService()
+ LoadData(): Response
+ DeleteData(): Response
+ Register(email: string, password: string, nickname: string): Response
+ Register(email: string, password: string, nickname: string, emailhost: string): Response
+ Login(email: string, password: string): Response<User>
+ Logout(email: string): Response
+ Unsid(email: string): bool
Business Layer
EventControl
- boards: Dictionary<int,Board>
- columns: List<Column>
- boards_current_id: int
- columns_current_id: int
- tasks_current_id: int
+ BoardControl()
+ LoadData(): void
+ DeleteData(): void
+ GetBoard(email: string, active bool): Event
+ AssignTask(email: string, columnOrdinal, int taskid, int, emailassignee, string, active bool)
+ AddTask(email: string title string, description string, duedate DateTime, active bool): Task
+ DeleteTask(email: string, columnOrdinal, int taskid, int, active bool): void
+ UpdateTaskDueDate(email: string, columnOrdinal, int taskid, int, duedate, DateTime, active
+ UpdateTaskTitle(email: string, columnOrdinal, int taskid, int title, string, active bool): void
+ UpdateTaskDescription(email: string, columnOrdinal, int taskid, int, description: string, active
+ AdvanceTask(email: string, columnOrdinal, int taskid, int, active bool): void
+ GetColumn(email: string, columnName: string, active bool): Column
+ GetColumn(email: string, columnOrdinal, int, active bool): Column
+ ChangeColumnName(email: string, columnOrdinal, int, newname: string, active bool): void
+ LimitColumnTasks(email: string, columnOrdinal, int, limit, int, active bool): void
+ RemoveColumn(email: string, columnOrdinal, int, active bool): void
+ AddColumn(email: string, columnOrdinal, int, Name: string, active bool): Column
+ MoveColumnRight(email: string, columnOrdinal, int, active bool): Column
+ MoveColumnLeft(email: string, columnOrdinal, int, active bool): Column
+ AddBoard: email: string)
+ AddUserToBoard(email: string, password: string, nickname: string, emailhost: string): void
UserControl
- users: Dictionary<string,User>
- activeuser: User
- users_id: int
- boards_id: int
+ UserControl()
+ LoadData(): void
+ DeleteData(): void
+ Register(email: string, password: string, nickname: string, factory: IFactory): void
+ Login(email: string, password: string): User
+ Logout(): void
+ Unsid(email: string): bool
+ user_boardidinuserid: string): int
0..n
BoardPackage
Task
- id: int
- CreatorTime: DateTime
- Title: string
- Description: string
- DueDate: DateTime
- _columnid: int
+ Task(id: int, creatortime: DateTime, title: string, description: string, duedate DateTime, columnid: int)
+ edittitle(title: string): void
+ editdescription(description: string): void
+ edituedate(duedate: DateTime): void
+ Import(Jsn:Data Task): void
+ Save (action: string): void
+ toJsn() : Data Task
+ update-id: int, attributename: string, attributeValue: string): void
+ update-id: int, attributename: string, attributeValue: int): void
+ update-id: int, attributename: string, attributeValue: DateTime): void
1..n
0..n
Column
- _id: int
- _name: string
- _tasks: List<Task>
- _bord: int
- _ordinal: int
- _email: string
- _boardid: int
+ Column(id: int, name: string, bord: int, ordinal: int, email: string, boardid: int)
+ Column()
+ ChangeColumnname(name:string): void
+ updateordinal(ord: int): void
+ updateBord(ord: int, title: string): void
+ updatedescription(ord: int, description: string): void
+ updateduedate(ord: int, duedate: DateTime): void
+ addtask(task: Task): void
+ RemoveTask(taskid: int): Task
+ AssignTask(taskid: int, emailassignee: string): void
+ DeleteTask(email: string, taskid: int): void
+ ahasbeen(id: int): void
+ isfull () bool
+ Import (Jsn :Data Column): void
+ Save (action: string): void
+ toJsn() : Data Column
+ update-id: int, attributename: string, attributeValue: string): void
+ update-id: int, attributename: string, attributeValue: int): void
Board
- _id: int
- _emailhost: string
- _columns: List<Column>
+ Board(id: int, email: string, column_current_id: int)
+ Board(columns: List<Column>)
+ AssignTask(email: string, columnOrdinal, int taskid, int, emailassignee: string): void
+ AddTask(id: int title, string, description:string, duedate DateTime): Task
+ DeleteTask(email: string, columnOrdinal, int taskid, int): void
+ UpdateTaskDueDate(columnOrdinal, int taskid, int duedate, DateTime): void
+ UpdateTaskTitle(columnOrdinal, int taskid, int title, string): void
+ AdvanceTask(columnOrdinal, int taskid, int): void
+ UpdateTaskDescription(columnOrdinal, int taskid, int, description: string): void
+ LimitColumnTasks(columnOrdinal, int, limit, int): void
+ ChangeColumnname(email: string, columnOrdinal, int, newname: string): void
+ GetColumn(email: string, columnName: string): Column
+ GetColumn(email: string, columnOrdinal, int): Column
+ RemoveColumn(columnOrdinal: int): void
+ AddColumn(email: string, columnOrdinal, int, name: string, ordinal, int): Column
+ MoveColumnRight(email: string, columnOrdinal, int): Column
+ MoveColumnLeft(email: string, columnOrdinal, int): Column
+ MoveColumnLeft(columnOrdinal: int): void
+ MoveColumnRight(columnOrdinal: int): void
UserPackage
<<Interface>>
PersonalObject
<T extends DataObject>
+ Tvalues(): T
+ Import(Jsn : T): void
+ Save(Jsn: string): void
<<Interface>>
IUser
+ login(+ email: string, password: string): void
+ login(+ logged_in: bool, hostemail: string): void
+ login(+ bool loggedin, hostemail: string): void
+ Logout(): void
UserPackage
<<Interface>>
Factory
+ Create(search id: string,nickname, string,password): IUser
+ login(+ boardid: int)
AssignerFactory
- boardid: int
- email: string
+ AssignerFactory(email: string, boardid: int)
HostFactory
- boardid: int
- email: string
+ HostFactory(email: string, boardid: int)
HostUser
- _id: int
- _email: string
- _nickname: string
- password: string
- hast: bool
- _boardid: int
- logged_in: bool
+ Host(boardid: int, nickname: string, password: string, boardid: int)
+ Login(password: string): void
+ Logout(): void
+ Data(): Data User
+ Import(Jsn: Data User): void
+ Save(action: string): void
+ logged_in(): bool
+ Unsid(hostemail: string): bool
+ addPlaceJsonUsers(): bool
AssignerUser
- _id: int
- _email: string
- _nickname: string
- password: string
- hast: bool
- _boardid: int
- logged_in: bool
+ Host(boardid: int, nickname: string, password: string, boardid: int)
+ Login(password: string): void
+ Logout(): void
+ Data(): Data User
+ Import(Jsn: Data User): void
+ Save(action: string): void
+ logged_in(): bool
+ Unsid(hostemail: string): bool
+ addPlaceJsonUsers(): bool
Data Access Layer
Lib0104
UserControl
- UserTableName: string
+ SelectAllUsers(): List<User>
# ConvertReaderToObject(reader: SQLiteDataReader): DataObject
+ Insert (user: User): void
TaskControl
- TaskTableName: string
+ SelectAllTasks(): List<Task>
# ConvertReaderToObject(reader: SQLiteDataReader): DataObject
+ Insert (task: Task): void
BoardControl
- BoardTableName: string
+ SelectAllBoards(): List<Board>
# ConvertReaderToObject(reader: SQLiteDataReader): DataObject
+ Insert (board: Board): void
ColumnControl
- ColumnTableName: string
+ SelectAllColumns(): List<Column>
# ConvertReaderToObject(reader: SQLiteDataReader): DataObject
+ Insert (column: Column): void
SqlObjectmodel
+ DCUserHasName: string
#_controller: DaControl
+ id: int
DaControl abstract
_connectionString: string
_tableName: string
+ DaControl(dbFilePath: string)
+ Update(id: int, attributeName: string, attributeValue: string): void
+ Update(id: int, attributeName: string, attributeValue: int): void
+ Update(id: int, attributeName: string, attributeValue: DateTime): void
# Select(): List<DaObject>
# ConvertReaderToObject(reader: SQLiteDataReader): DaObject
+ Delete(ZTCCSu: DaObject): void
Objects
Task
_creatorTime: DateTime
+ CreatedTime: string
_Title: string
+ Title: string
_description: string
+ Description: string
_duEDate: DateTime
+ DuDate: string
_columnID: int
+ ColumnID: string
+ Task(id: int, creatortime: DateTime, title: string, description: string, duEDate: DateTime, columnID: int)
User
_email: string
+ Email: string
_nickname: string
+ Nickname: string
_password: string
+ Password: string
_hast: bool
+ Hast: string
_boardid: int
+ Boardid: string
+ User(email: string, nickname: string, password: string, hast: bool, boardid: int)
Board
_emailhost: string
+ Emailhost: string
+ Boardid: int, emailhost: string)
Column
_name: string
+ Name: string
_limit: string
+ Limit: string
_ordinal: int
+ Ordinal: string
_bsgbid: int
+ Bsgbid: int
+ Column(id: int, name: string, limit: int, ordinal: int, email: string, boardid: int)