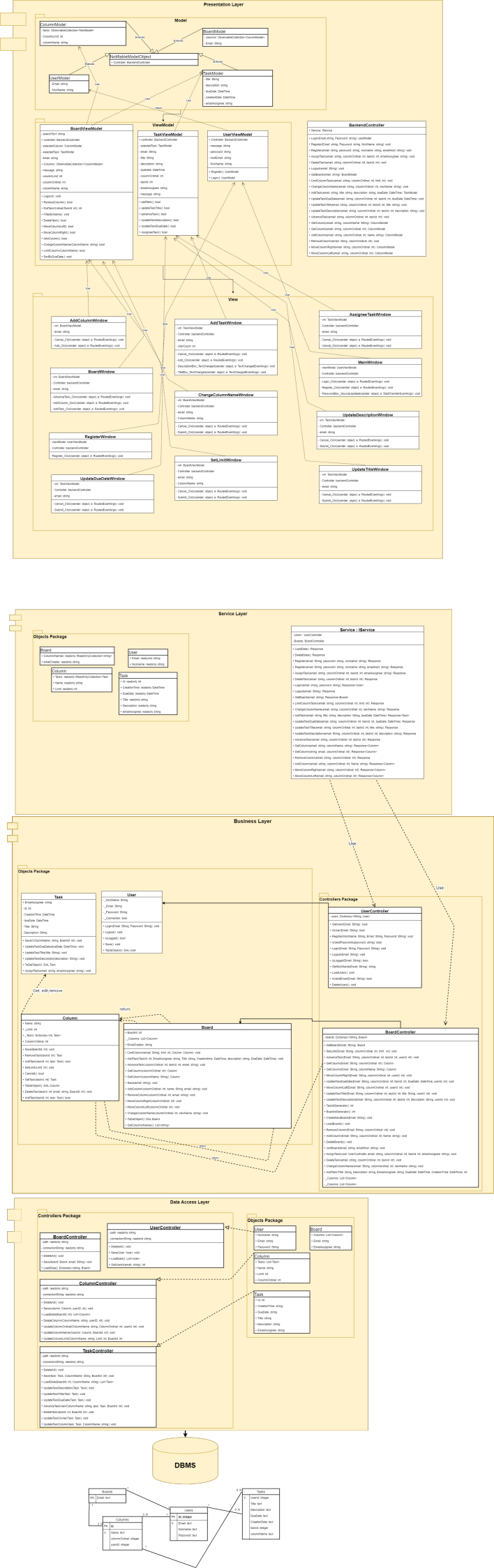
Presentation Layer
Model
ColumnModel
- data: ObservableCollection<TaskModel>
- ColumnLimit: int
- columnName: string
BoardModel
- columns: ObservableCollection<ColumnModel>
- Email: string
NotifiableModelObject
+ Controller: BackendController
UserModel
- Email: string
- NickName: string
TaskModel
- title: string
- description: string
- dueDate: DateTime
- creationDate: DateTime
- emailAssignee: string
ViewModel
BoardViewModel
- searchText: string
- controller: BackendController
- selectedColumn: ColumnModel
- selectedTask: TaskModel
- email: string
- Columns: ObservableCollection<ColumnModel>
- message: string
- usersLoad: int
- columnOrdinal: int
- columnName: string
+ Logout(): void
+ RemoveColumn(): void
+ ShiftBackOrder(int basid, int): int
+ FilterBy(name): void
+ DeleteTask(): bool
+ MoveColumnLeft(): bool
+ MoveColumnRight(): bool
+ AddColumn(): bool
+ ChangeColumnNameColumnName: string): bool
+ LimitColumnColumnName: bool
+ SortByDueDate(): void
TaskViewModel
+ controller: BackendController
- selectedTask: TaskModel
- email: string
- title: String
- description: string
- dueDate: DateTime
- columnOrdinal: int
- taskid: int
- emailAssignee: string
- message: string
+ addTask(): bool
+ updateTaskTitle(): bool
+ advanceTask(): bool
+ UpdateTaskDescription(): bool
+ UpdateTaskDueDate(): bool
+ AssignerTask(): bool
UserViewModel
+ Controller: BackendController
- message: string
- password: string
- hostEmail: string
- NickName: string
+ Register(): UserModel
+ Login(): UserModel
BackendController
+ Service: IService
+ LoginEmail:string, Password :string): userModel
+ RegisterEmail: string, Password :string, NickName :string): void
+ RegisterEmail: string, password: string, nickname: string, emailHost: string): void
+ AssignTaskUsrEmail: string, columnOrdinal: int, taskid: int, emailAssignee :string): void
+ DeleteTaskUsrEmail: string, columnOrdinal: int, taskid: int): void
+ LogoutEmail: String): void
+ GetBoardEmail: string): BoardModel
+ LimitColumnTasksUsrEmail: string, columnOrdinal: int, limit: int): void
+ ChangeColumnNameUsrEmail: string, columnOrdinal: int, newName: string): void
+ AddTaskUsrEmail: title: string, description: string, dueDate: DateTime): TaskModel
+ UpdateTaskDueDateUsrEmail: string, columnOrdinal: int, taskid: int, dueDate: DateTime): void
+ UpdateTaskTitleUsrEmail: string, columnOrdinal: int, taskid: int, title: string): void
+ UpdateTaskDescriptionUsrEmail: string, columnOrdinal: int, taskid: int, description :string): void
+ AdvanceTaskUsrEmail: string, columnOrdinal: int, taskid: int): void
+ GetColumnsUsrEmail: string, columnName :String): ColumnModel
+ GetColumnsUsrEmail: string, columnOrdinal: int): ColumnModel
+ AddColumnUsrEmail: string, columnOrdinal: int, name: string): ColumnModel
+ RemoveColumnUsrEmail: string, columnOrdinal: int): void
+ MoveColumnRightUsrEmail: string, columnOrdinal: int): ColumnModel
+ MoveColumnLeftUsrEmail: string, columnOrdinal: int): ColumnModel
View
AddColumnWindow
- vm: BoardViewModel
- email: string
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Add_Click(sender: object, e: RoutedEventArgs): void
AddTaskWindow
- vm: TaskViewModel
- Controller: backendController
- email: string
- charCount: int
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Add_Click(sender: object, e: RoutedEventArgs): void
- DescriptionBox_TextChanged(sender: object, e: TextChangedEventArgs): void
- TitleBox_TextChanged(sender: object, e: TextChangedEventArgs): void
AssigneeTaskWindow
- vm: TaskViewModel
- Controller: backendController
- email: string
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Cancel_Click(sender: object, e: RoutedEventArgs): void
MainWindow
- viewModel: UserViewModel
- Controller: backendController
- Login_Click(sender: object, e: RoutedEventArgs): void
- Register_Click(sender: object, e: RoutedEventArgs): void
- PasswordBox_SourceUpdated(sender: object, e: DataTransferEventArgs): void
BoardWindow
- vm: BoardViewModel
- Controller: backendController
- email: string
- AdvanceTask_Click(sender: object, e: RoutedEventArgs): void
- AddColumn_Click(sender: object, e: RoutedEventArgs): void
- AddTask_Click(sender: object, e: RoutedEventArgs): void
ChangeColumnNameWindow
- vm: BoardViewModel
- Controller: backendController
- email: string
- ColumnName: string
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Submit_Click(sender: object, e: RoutedEventArgs): void
UpdateDescriptionWindow
- vm: TaskViewModel
- Controller: backendController
- email: string
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Submit_Click(sender: object, e: RoutedEventArgs): void
RegisterWindow
- viewModel: UserViewModel
- Controller: backendController
- Register_Click(sender: object, e: RoutedEventArgs): void
SetLimitWindow
- vm: BoardViewModel
- Controller: backendController
- email: string
- ColumnName: string
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Submit_Click(sender: object, e: RoutedEventArgs): void
UpdateDueDateWindow
- vm: TaskViewModel
- Controller: backendController
- email: string
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Submit_Click(sender: object, e: RoutedEventArgs): void
UpdateTitleWindow
- vm: TaskViewModel
- Controller: backendController
- email: string
- Cancel_Click(sender: object, e: RoutedEventArgs): void
- Submit_Click(sender: object, e: RoutedEventArgs): void
Service Layer
Objects Package
Board
+ ColumnNames: readonly IReadOnlyCollection<string>
+ emailOwner: readonly string
Column
+ Tasks: readonly IReadOnlyCollection<Task
+ Name: readonly string
+ Limit: readonly int
User
+ Email: readonly string
+ NickName: readonly string
Task
+ id: readonly int
+ CreationTime: readonly DateTime
+ DueDate: readonly DateTime
+ Title: readonly string
+ Description: readonly string
+ emailAssignee: readonly string
Service : IService
- Users : UserController
- Boards: BoardController
+ LoadData() : Response
+ DeleteData() : Response
+ Register(email: String, password: string, nickname: string): Response
+ Register(email: String, password: string, nickname: string, emailHost: string): Response
+ AssignTaskUsrEmail: string, columnOrdinal: int, taskid: int, emailAssignee :string): Response
+ DeleteTaskUsrEmail: string, columnOrdinal: int, taskid: int): Response
+ Login(email: string, password: string): Response<User>
+ LogoutEmail: String): Response
+ GetBoardEmail: string): Response<Board>
+ LimitColumnTasksUsrEmail: string, columnOrdinal: int, limit: int): Response
+ ChangeColumnNameUsrEmail: string, columnOrdinal: int, newName: string): Response
+ AddTaskUsrEmail: string, title: string, description: String, dueDate: DateTime): Response<Task>
+ UpdateTaskDueDateUsrEmail: string, columnOrdinal: int, taskid: int, dueDate: DateTime): Response
+ UpdateTaskTitleUsrEmail: string, columnOrdinal: int, taskid: int, title: string): Response
+ UpdateTaskDescriptionUsrEmail: string, columnOrdinal: int, taskid: int, description: string): Response
+ AdvanceTaskUsrEmail: string, columnOrdinal: int, taskid: int): Response
+ GetColumnsUsrEmail: string, columnName :string): Response<Column>
+ GetColumnsUsrEmail: string, columnOrdinal: int): Response<Column>
+ RemoveColumnUsrEmail: string, columnOrdinal: int): Response
+ AddColumnUsrEmail: string, columnOrdinal: int, Name :string): Response<Column>
+ MoveColumnRightUsrEmail: string, columnOrdinal: int): Response<Column>
+ MoveColumnLeftUsrEmail: string, columnOrdinal: int): Response<Column>
Business Layer
Use
Use
Objects Package
Task
+ EmailAssignee: string
+ id: int
- CreationTime: DateTime
- dueDate: DateTime
- Title: String
- Description: String
+ SaveUCColumnName: string, Boardid: int): void
+ UpdateTaskDueDateStatus(dueDateTime): void
+ UpdateTaskTitleTitle: String): void
+ UpdateTaskDescriptionDescription: String): void
+ TaskObject(): DAL.Task
+ AssignTaskUsrEmail: string, emailAssignee :string): void
User
- _nickname: String
- _Email: String
- _Password: String
- _Connection: bool
+ LoginEmail: String, Password: String): void
+ Logout(): void
+ IsLogged(): bool
+ Save(): void
+ TaskObject(): DAL.User
Controllers Package
UserController
- users: Dictionary<String, User>
+ GetUserEmail: String): User
+ IsUserEmail: String): bool
+ RegisterNickname: String, Email: String, Password: String): void
+ IsValidPasswordpassword: string): bool
+ LoginEmail: String, Password: String): void
+ LogoutEmail: String): void
+ IsLoggedEmail: String): bool
+ GetNickNameEmail: String): string
+ LoadUsers(): void
+ IsValidEmailEmail: String): bool
+ DeleteUsers(): void
Get, edit,remove
return
Column
+ Name: string
- _Limit: int
- _Tasks: Dictionary<int, Task>
- ColumnOrdinal: int
+ SaveBoardid: int): void
+ RemoveTaskbasid: int): Task
+ AddTaskbasid: int, task: Task): void
+ SetLimitLimit: int): void
+ Cancel(): bool
+ GetTaskbasid: int): Task
+ TaskObject(): DAL.Column
- DeleteTaskbasid: int, email: string, Boardid: int): void
+ AddTaskbasid: int, task: Task): bool
Board
+ Boardid: int
_Columns : List<Column>
+ EmailOwner: string
+ LimitColumnLimit: String, limit: int, Column: Column): void
+ AddTaskTasktid: int, EmailAssignee: string, Title: string, CreationTime: DateTime, description: string, DueDate: DateTime): void
+ AdvanceTaskcolumnOrdinal: int, taskid: int, email: string): void
+ GetColumnscolumnOrdinal: int): Column
+ GetColumnscolumnName: String): Column
+ Save(email: string): void
+ AddColumncolumnOrdinal: int, name: String, email: string): void
+ RemoveColumncolumnOrdinal: int, email: string): void
+ MoveColumnRightcolumnOrdinal: int): void
+ MoveColumnLeftcolumnOrdinal: int): void
+ ChangeColumnNamecolumnOrdinal: int, newName: string): void
+ TaskObject(): DAL.Board
+ GetColumnsNames(): List<string>
BoardController
- boards: Dictionary<String, Board>
+ GetBoardOrdinal: String): Board
+ SetLimitEmail: String, columnOrdinal: int, limit: int): void
+ AdvanceTaskEmail: String, columnOrdinal: int, taskid: int, usersid: int): void
+ GetColumnsOrdinal: String, columnOrdinal: int): Column
+ GetColumnsEmail: String, columnName: String): Column
+ MoveColumnRightEmail: String, columnOrdinal: int, usersid: int): void
+ UpdateTaskDueDateEmail: String, columnOrdinal: int, taskid: int, DueDate: DateTime, usersid: int): void
+ MoveColumnLeftEmail: String, columnOrdinal: int, usersid: int): void
+ UpdateTaskTitleEmail: String, columnOrdinal: int, taskid: int, title: String, usersid: int): void
+ UpdateTaskDescriptionEmail: String, columnOrdinal: int, taskid: int, description: String, usersid: int): void
+ TaskIdGenerator(): int
+ BoardIdGenerator(): int
+ CreateNewBoard(Email: String): void
+ LoadBoards(): void
+ RemoveColumnEmail: String, columnOrdinal: int): void
+ AddColumnOrdinal: String, columnOrdinal: int, Name: String): void
+ DeleteBoard(): void
+ JoinBoardEmail: string, emailHost: string): void
+ AssignTaskUser: UserController, email: string, columnOrdinal: int, taskid: int, emailAssignee: string): void
+ DeleteTaskUsrEmail: string, columnOrdinal: int, taskid: int): void
+ ChangeColumnNameUsrEmail: String, columnOrdinal: int, newName: string): void
+ AddTaskTitle: string, Description: string, EmailAssignee: string, DueDate: DateTime, CreationTime: DateTime): int
_Columns: List<Column>
_Columns: List<Column>
Data Access Layer
Controllers Package
BoardController
- path: readonly string
- connectionString: readonly string
+ DeleteAll(): void
+ Save(board: Board, email: String): void
+ LoadData(): Dictionary<string, Board>
UserController
- path: readonly string
- connectionString: readonly string
+ DeleteAll(): void
+ Save(User User): void
+ LoadData(): List<User>
+ GetUser(email: string): int
ColumnController
- path: readonly string
- connectionString: readonly string
+ DeleteAll(): void
+ Save(column: Column, userID: int): void
+ LoadDataBoardId: int): List<Column>
+ DeleteColumnColumnName: string, userID: int): void
+ UpdateColumnOrdinalColumnName: string, ColumnOrdinal: int, usersid: int): void
+ UpdateColumnNameColumn: Column, Boardid: int): void
+ UpdateColumnLimitColumnName: string, Limit: int, Boardid: int
TaskController
- path: readonly string
- connectionString: readonly string
+ DeleteAll(): void
+ Save(task: Task, ColumnNameid: string, Boardid: int): void
+ LoadDataBoardId: int, ColumnName: string): List<Task>
+ UpdateTaskDescriptionTask: Task): void
+ UpdateTaskTitleTitle: Task): void
+ UpdateTaskDueDateTask: Task): void
+ AdvanceTaskcolumnColumnName: string, task: Task, Boardid: int): void
+ DeleteTaskcolumnid: int, Boardid: int): void
+ UpdateTaskOwnerTask: Task): void
+ UpdateTaskColumnUser: Task, ColumnName: string): void
Objects Package
User
+ Nickname: string
+ Email: string
+ Password: String
Board
+ Columns: List<Column>
+ Email: string
+ EmailAssignee: string
Column
- Tasks: List<Task>
+ Name: string
+ Limit: int
+ ColumnOrdinal: int
Task
+ id: int
+ CreationTime: string
+ DueDate: string
+ Title: string
+ Description: string
+ EmailAssignee: string
DBMS
Boards
PK Email text
Columns
PK Id
Name text
columnOrdinal integer
userID integer
Users
PK Id, integer
Email text
Nickname text
Password text
Tasks
Id Userid integer
Title text
Description text
DueDate text
CreationDate text
taskid integer
columnname text