RRD-**R**oi **R**aviv **D**or- Software Engineer Company- Best in market

HLD

Kanban Project

Milestone 3#

Main purpose and description:

Kanban Board Project- This system is designed to give an answer for managing tasks for projects and long term assignments. A user will be able to create new tasks, divide the tasks to columns, depending on their status, modify their information and status, and that will generate the ability to observe the whole picture, all the tasks and their current status.

This document contains the information, requirements, specifications and general description of the software.

The document will serve as a term sheet on which the project will be based, according to which the development process will be advanced.

Terminology:

Kanban-

Kanban (Japanese 看板, signboard or billboard) is a lean method to manage and improve work across human systems. It represents a board of tasks, and divides the tasks by tagged columns. A user can login to the review the existing tasks, manage them and create new tasks.

Board-

A board is a graphical way to describe a collection of columns, which each of them contains a collection of tasks. The board binds all the parts together.

Column-

Represent a collection of tasks that are bounded together with a joined attribute. In this case the attribute each collection will represent is the status which the tasks are in. The columns are defaulted for all boards: Backlog, In Progress, Done. New Columns can be added, and exist columns can be removed. In addition, user can change the order of columns in his board.

Task-

A task is a described mission that the user needs to accomplish. The task will contain all the information that is needed, so the user will be able to review his tasks and manage his work correctly. The information that will be restored:

- Dates: Creation date and due date
- Title and description

Each task will be assigned to the user that created it, and will be his responsibility. User can add new task to the leftmost column on board, and each task can be promoted from one column to the next one in current presentation order, Except for the last column.

A task can be changed by the user, unless it arrives to the final column in board. Task can be sorted by date, and filtered by its text fields.

Login-

A process in which the user enters his email address and password, and the system allows him to use it. Each email will be unique and will represent the user's profile in the system. The password will be used to validate that the person entering the email and password is authentic.

Password-

A personal token, string of letters and digits that the user will use, and that will allow the system to verify the identity of the logging in user.

Actors:

User-

Each member of the team that is using the system. Staff members from all types of roles.

Manager-

(Proposed feature)

A manager will be a high level user. He will be able to assign users to tasks, and create joined boards for group projects. The manager will be able to monitor the group's progress and the performance of the group's members.

Client-

The clients are companies or organizations with the need to manage tasks and monitor their performance. The BGU software engineer students- is an example for such group.

System's Behavior Description:

Users and login:

Functional-

- Each user is be able to sign up or login the system using his email address and password
- Each email is unique in the system
- The login succeeds if and only if the email and the password matches the user's email and password.

Non-functional-

- Each password matches the semantics requirements:
  - length of 4 to 20 characters
  - must include at least one capital character, one small character and a number

System logic:

Functional-

- Each board contains a list of columns, in each one a list of tasks
- The user is be able to review the board and it's columns
- The user is able to create new tasks for the leftmost column
- The user is able to modify tasks that are in specific columns
- The user is able to promote task from each column to the next one in line, until the last one
- The user can limit the maximum number of tasks in each column

Non functional-

- The board will have 3 columns, by default, named:
  - Backlog
  - In Progress
  - Done
- The user can add, remove and change presentation order for columns.
- Each task will be created with the following information:
  - The creation date
  - Title, max of 50 characters and not empty
  - Description, max of 300 characters, optional

- A board will support adding new tasks to its leftmost column only
- Only tasks that are not in the rightmost can be changed by the user
- All the task's data can be changed except for the creation time

Proposed features:

- The users are created by the hierarchy of the group
- A manager has the ability to create new boards, and add permit access for the users that are relevant to each board
- Each user is able to see the tasks of the entire group working on the same project
- Each user is responsible of his own tasks
- The manager can transfer the responsibility of tasks between the people in the group.

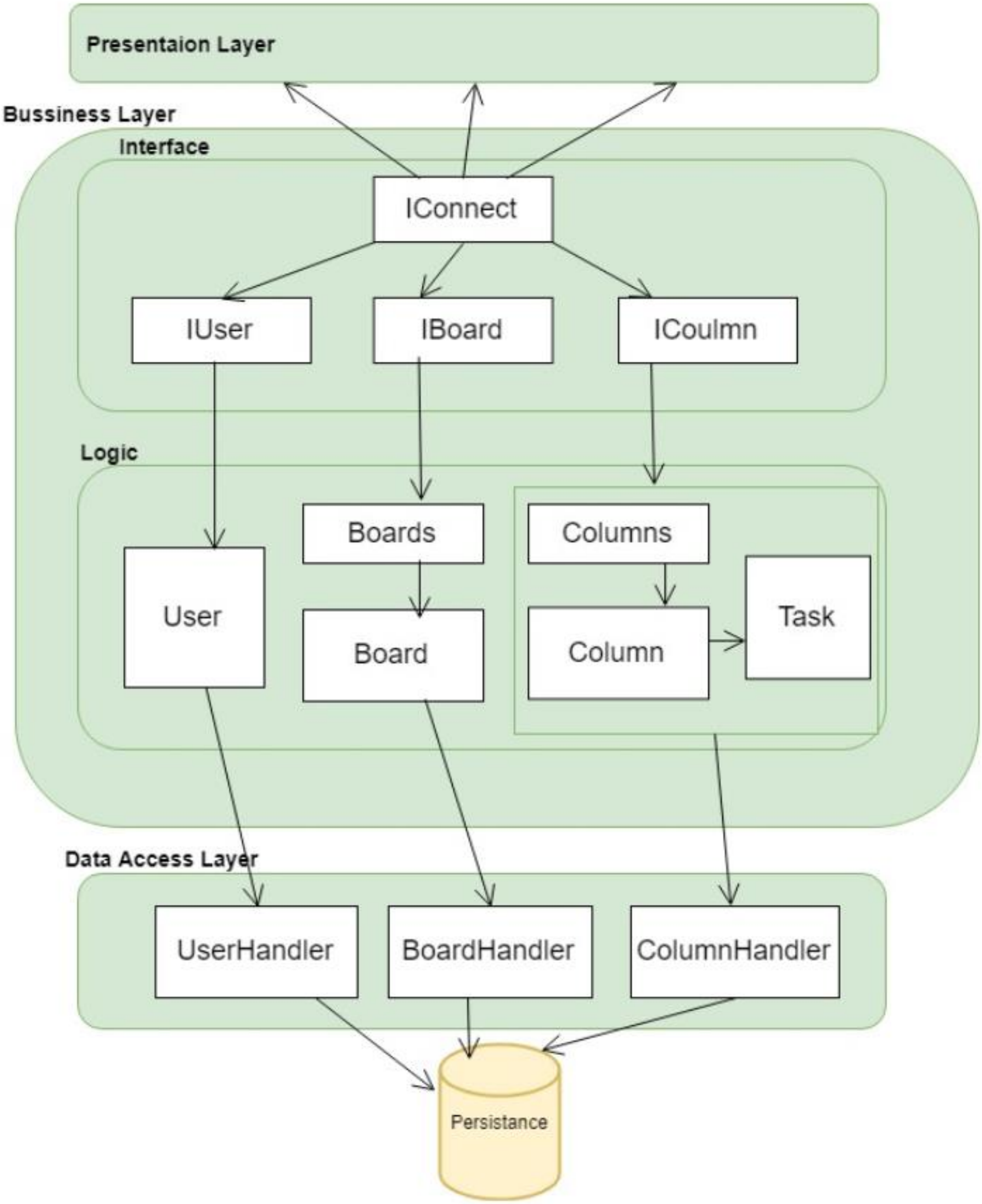
General non-functional requirements:

Non-functional-

- The system's data: users, boards, columns and tasks, will be stored and preserved.
- The system will log all actions that occur during runtime, the faults and the abnormal events (exceptions), to monitor the system, warn of user errors, and to allow to improve system performance

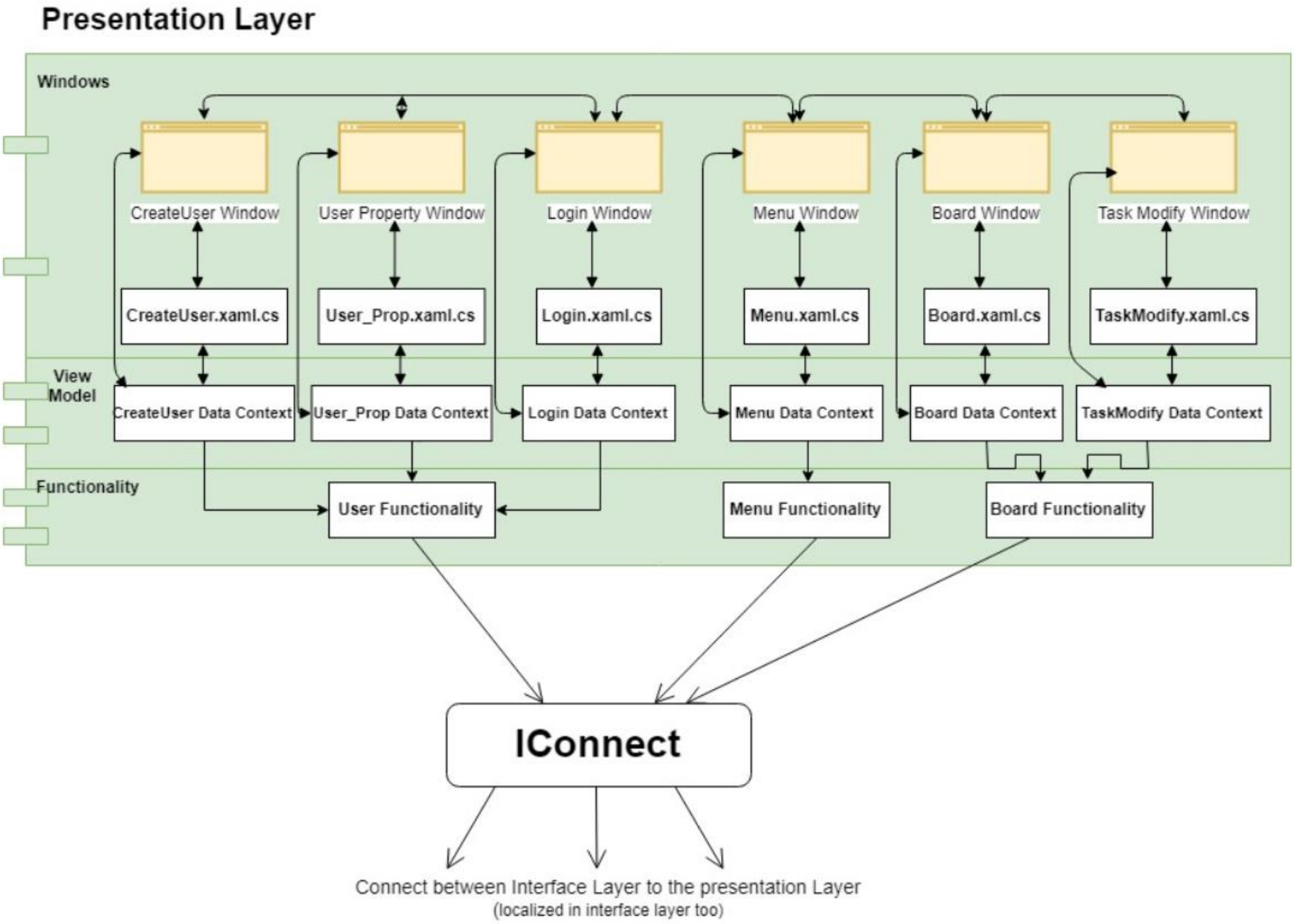
High level entities and software design:

The system will contain the following Tiers and main entities (objects):



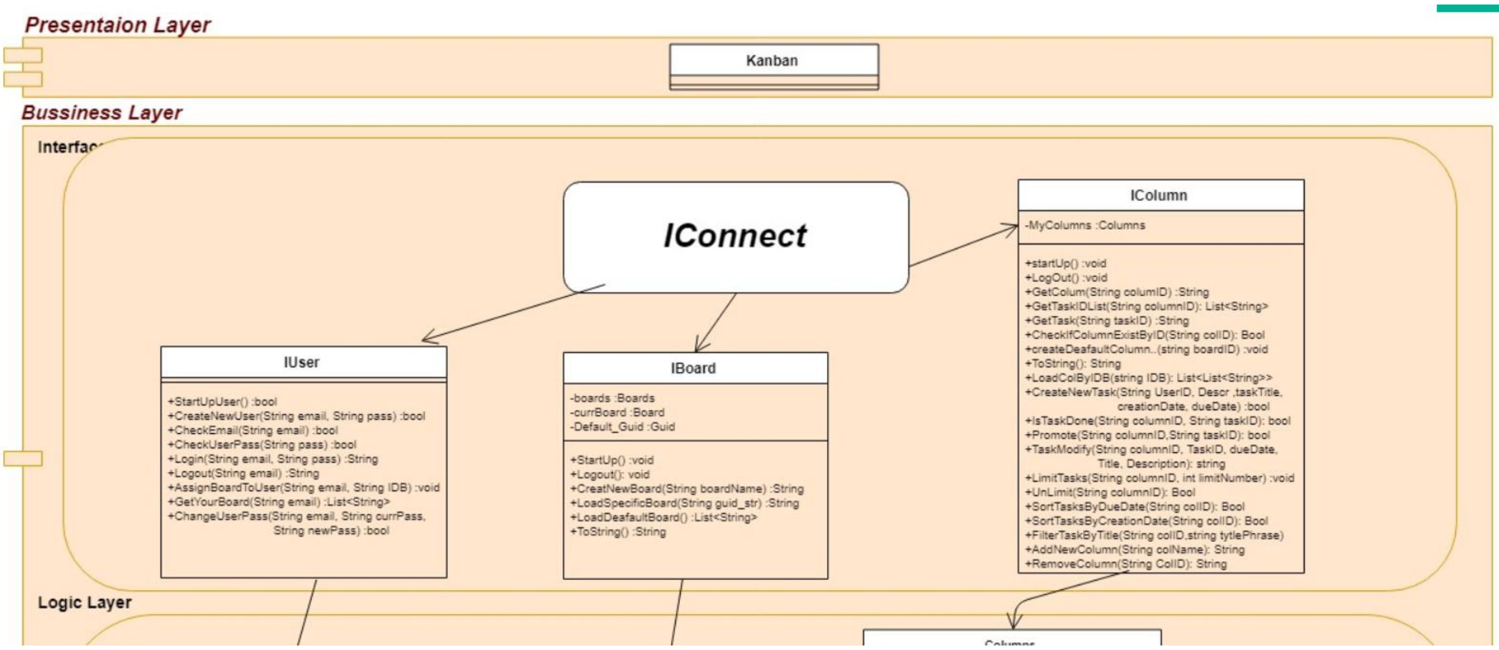
The presentation layer will contain the following Tiers and main entities (objects):

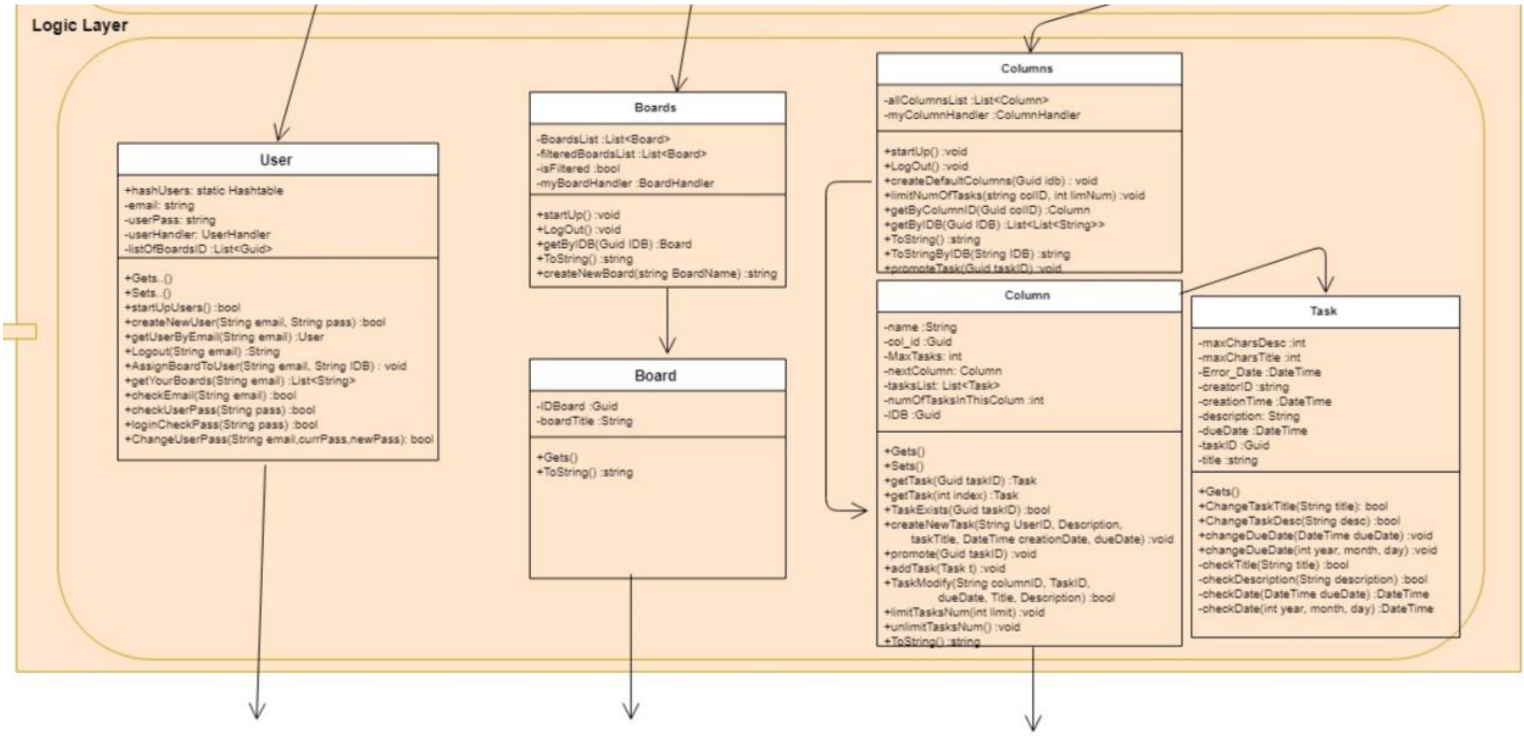
'IConnect' is placed in interface Layer, and is a link between the layers.

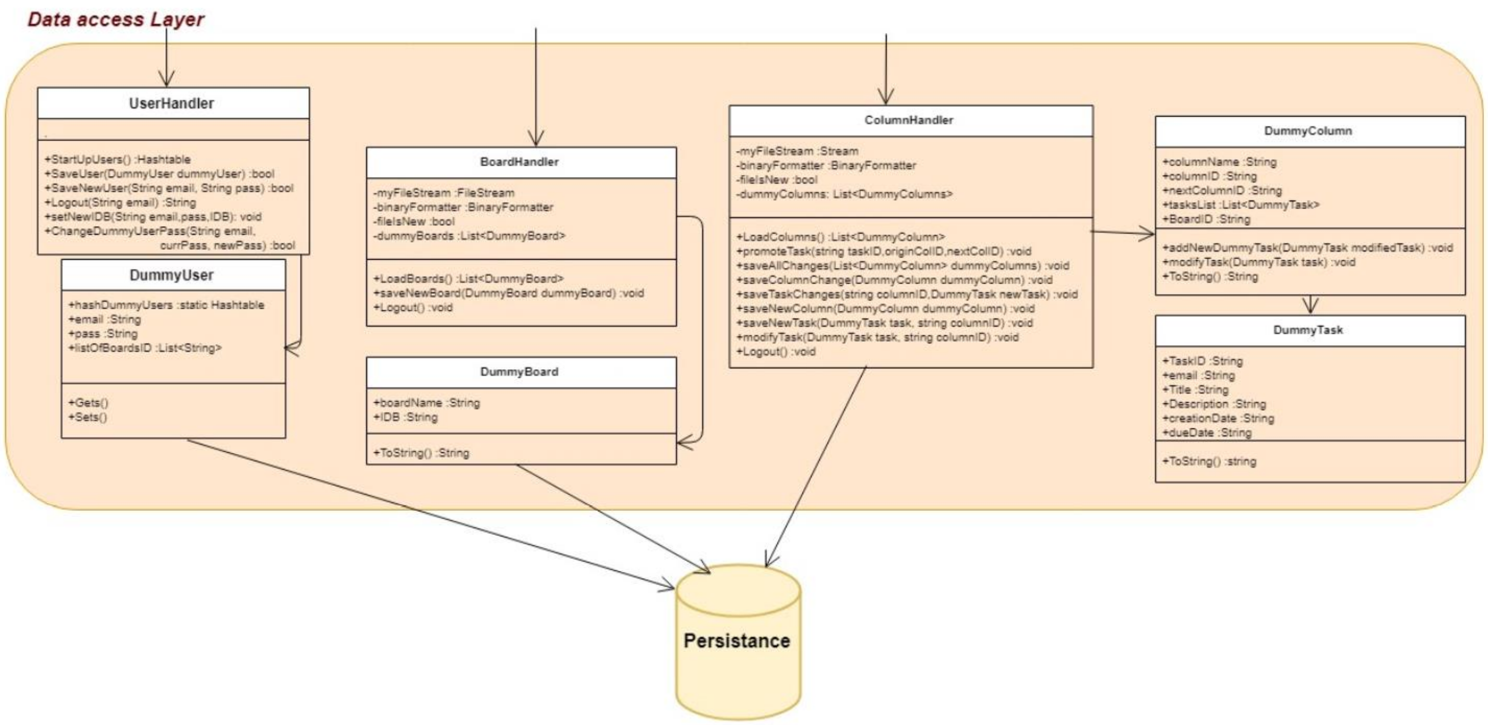


LLD:

We would like to focus in each layer, in order to elaborating and expanding on the classes and their functionality.


Presentaion Layer
Kanban
Bussiness Layer
Interfac
IConnect
IColumn
-MyColumns :Columns
+startUp() :void
+LogOut() :void
+GetColum(String columID) :String
+GetTaskIDList(String columnID): List<String>
+GetTask(String taskID) :String
+CheckIfColumnExistByID(String colID): Bool
+createDeafaultColumn..(string boardID) :void
+ToString(): String
+LoadColByIDB(string IDB): List<List<String>>
+CreateNewTask(String UserID, Descr ,taskTitle, creationDate, dueDate) :bool
+IsTaskDone(String columnID, String taskID): bool
+Promote(String columnID,String taskID): bool
+TaskModify(String columnID, TaskID, dueDate, Title, Description): string
+LimitTasks(String columnID, int limitNumber) :void
+UnLimit(String columnID): Bool
+SortTasksByDueDate(String colID): Bool
+SortTasksByCreationDate(String colID): Bool
+FilterTaskByTitle(String colID,string tytlePhrase)
+AddNewColumn(String colName): String
+RemoveColumn(String ColID): String
IUser
+StartUpUser() :bool
+CreateNewUser(String email, String pass) :bool
+CheckEmail(String email) :bool
+CheckUserPass(String pass) :bool
+Login(String email, String pass) :String
+Logout(String email) :String
+AssignBoardToUser(String email, String IDB) :void
+GetYourBoard(String email) :List<String>
+ChangeUserPass(String email, String currPass, String newPass) :bool
IBoard
-boards :Boards
-currBoard :Board
-Default_Guid :Guid
+StartUp() :void
+Logout(): void
+CreatNewBoard(String boardName) :String
+LoadSpecificBoard(String guid_str) :String
+LoadDeafaultBoard() :List<String>
+ToString() :String
Logic Layer
Columns


Logic Layer
User
+hashUsers: static Hashtable
-email: string
-userPass: string
-userHandler: UserHandler
-listOfBoardsID :List<Guid>
+Gets..()
+Sets..()
+startUpUsers() :bool
+createNewUser(String email, String pass) :bool
+getUserByEmail(String email) :User
+Logout(String email) :String
+AssignBoardToUser(String email, String IDB) : void
+getYourBoards(String email) :List<String>
+checkEmail(String email) :bool
+checkUserPass(String pass) :bool
+loginCheckPass(String pass) :bool
+ChangeUserPass(String email,currPass,newPass): bool
Boards
-BoardsList :List<Board>
-filteredBoardsList :List<Board>
-isFiltered :bool
-myBoardHandler :BoardHandler
+startUp() :void
+LogOut() :void
+getByIDB(Guid IDB) :Board
+ToString() :string
+createNewBoard(string BoardName) :string
Board
-IDBoard :Guid
-boardTitle :String
+Gets()
+ToString() :string
Columns
-allColumnsList :List<Column>
-myColumnHandler :ColumnHandler
+startUp() :void
+LogOut() :void
+createDefaultColumns(Guid idb) : void
+limitNumOfTasks(string colID, int limNum) :void
+getByColumnID(Guid colID) :Column
+getByIDB(Guid IDB) :List<List<String>>
+ToString() :string
+ToStringByIDB(String IDB) :string
+promoteTask(Guid taskID) :void
Column
-name :String
-col_id :Guid
-MaxTasks: int
-nextColumn: Column
-tasksList: List<Task>
-numOfTasksInThisColum :int
-IDB :Guid
+Gets()
+Sets()
+getTask(Guid taskID) :Task
+getTask(int index) :Task
+TaskExists(Guid taskID) :bool
+createNewTask(String UserID, Description, taskTitle, DateTime creationDate, dueDate) :void
+promote(Guid taskID) :void
+addTask(Task t) :void
+TaskModify(String columnID, TaskID, dueDate, Title, Description) :bool
+limitTasksNum(int limit) :void
+unlimitTasksNum() :void
+ToString() :string
Task
-maxCharsDesc :int
-maxCharsTitle :int
-Error_Date :DateTime
-creatorID :string
-creationTime :DateTime
-description: String
-dueDate :DateTime
-taskID :Guid
-title :string
+Gets()
+ChangeTaskTitle(String title): bool
+ChangeTaskDesc(String desc) :bool
+changeDueDate(DateTime dueDate) :void
+changeDueDate(int year, month, day) :void
-checkTitle(String title) :bool
-checkDescription(String description) :bool
-checkDate(DateTime dueDate) :DateTime
-checkDate(int year, month, day) :DateTime


Data access Layer
UserHandler
+StartUpUsers() :Hashtable
+SaveUser(DummyUser dummyUser) :bool
+SaveNewUser(String email, String pass) :bool
+Logout(String email) :String
+setNewIDB(String email,pass,IDB): void
+ChangeDummyUserPass(String email, currPass, newPass) :bool
DummyUser
+hashDummyUsers :static Hashtable
+email :String
+pass :String
+listOfBoardsID :List<String>
+Gets()
+Sets()
BoardHandler
-myFileStream :FileStream
-binaryFormatter :BinaryFormatter
-fileIsNew :bool
-dummyBoards :List<DummyBoard>
+LoadBoards() :List<DummyBoard>
+saveNewBoard(DummyBoard dummyBoard) :void
+Logout() :void
DummyBoard
+boardName :String
+IDB :String
+ToString() :String
ColumnHandler
-myFileStream :Stream
-binaryFormatter :BinaryFormatter
-fileIsNew :bool
-dummyColumns: List<DummyColumns>
+LoadColumns() :List<DummyColumn>
+promoteTask(string taskID,originColID,nextColID) :void
+saveAllChanges(List<DummyColumn> dummyColumns) :void
+saveColumnChange(DummyColumn dummyColumn) :void
+saveTaskChanges(string columnID,DummyTask newTask) :void
+saveNewColumn(DummyColumn dummyColumn) :void
+saveNewTask(DummyTask task, string columnID) :void
+modifyTask(DummyTask task, string columnID) :void
+Logout() :void
DummyColumn
+columnName :String
+columnID :String
+nextColumnID :String
+tasksList :List<DummyTask>
+BoardID :String
+addNewDummyTask(DummyTask modifiedTask) :void
+modifyTask(DummyTask task) :void
+ToString() :String
DummyTask
+TaskID :String
+email :String
+Title :String
+Description :String
+creationDate :String
+dueDate :String
+ToString() :string
Persistance