# Model Classes

User Class - The User class consists of 4 private variables, 1 private Object class, and 1 private array list(id, username, email, password, and tasks). Furthermore, it will have a constructor that will take 4 variables and initiate it with the class's variables. In addition to the variables, it will also initiate the Task arrayList. It will also have getters and setters for all variables including the arrayList.

Label Class - The label class contains only 2 private variables(name and color). The label class will create getters and setters for the variables. The getters will return the information while the setters will store the information. Furthermore, it will also contain a constructor method that will take in two strings and initialize it with the class's name and color.

Calendar Class - The calendar class consists of 1 private arrayList and 2 private String variables (currentMoth, and currentWeek). Furthermore, this class has 4 public methods which are getCurrentMonth, setCurrentMonth, getCurrentWeek, and setCurrentWeek. The getters will return the information while the setters will store the information.

Task class - This class will have 6 private variables, 2 int and 4 strings, and an arrayList (id, userId, title, description, dueDate, priority, and Label, respectively). This class will contain a constructor which will take 6 variables and initiate it to the class's variables. Additionally, it will initiate the Label arrayList. Finally, this method will have getters and setters for all the variables including the arrayList.

# Controller Classes

AuthController Class - This controller class will provide a private User arrayList. This class will have several methods which are loadUserFromCSV, signUp, saveUserToCSV, and a getUserById. The loadUserFromCSV method will take a String file, and will throw an IOException if the file is not found. Then, it will read the information from the file and make a new user and add it to the User arrayList, it will return nothing. The signUp method will take three strings (userName, email, and password), and will throw an IOException if the information is not found. This method is responsible for returning the user information if the user is not already registered. The saveUserToCsv will take in a User object and will throw an IOException if it is not found. This method will save the user information to the file. It will return nothing. The logInt method will have 2 Strings parameters (email and password). This method will

iterate with every available user in the User arrayList, which will return the user information if found. Otherwise it will return null.

CalendarController - This controller class will have only four methods which are getTaskByDate, getTaskByMonth, getTaskByWeek, and getTaskForm.The getTaskByDate will have a Date object as the parameter and will return void. The getTaskByMonth, which takes a String as a parameter (month) and will return void. The getTaskByWeek will take a String as a parameter (week) and will return void. The openTaskForm will take a Date object as the parameter and will return void.

LabelController - The labelController class will have 1 global Label arrayList. It will also consist of 6 methods which are loadLabelsFromCSV, findLabelByName, addLabel, deleteLabel, updateLabel, and getLabel.The loadLabelsFromCSV which have one string parameter (file) and throw an IOException if the file is not found. This method will load all content to the arraylist. The findLabelByName will take in a string parameter (name). This will return a label if found, otherwise it will return null. The addLabel will have a Label object parameter, and will add a Label object to the Label arrayList. The deleteLabel will take in a Label object. This method will delete a label object from the arrayList. The updateLabel method will take in a Label object and 2 Strings(newName and newColor). This method will update the content of the specific Label chosen by the user. The getLabel will return the labels.

TaskController - The taskController class will have a global Task arrayList. This class has a total of ten methods. The getTask method will take no parameters and will return Task. The loadtaskFromCSV will take 3 parameters (filename[string], user[User] arrayList, and allLabels[Label] arrayList ) and it will throw an IOException if the file is not found.This method will read every line and make a new task and insert all info to the new task before adding it to the Task arrayList.The addTask will take 2 parameters(User object and Task object) and throw an IOException. This method will add a task to the global task list and the user's task list before saving the task to the csv file. The saveTaskToCSV takes a Task parameter, and throws an IOException if not found. This will have all the information to save the task to the file. The editTask will have 3 parameters (User, Task taskId, Task updatedTaskData). This method will let the user edit the task. The deleteTask will have User and Task objects as their parameters. This method will let the user delete a task. The getTaskByPriority will have a User object and a String as its parameters. This method will create a new arraylist, search the task arrayList, and insert it to the new arraylist meant for labeled Tasks that are important. The getTaskByLabel will have a User and Task parameters and will return a Task. The getTaskByDate will have a User and String as its parameters and will return a Task. The getaAllTask will return all tasks in the Task arrayList.