

**Task Management Documentation**  
**Group Magical Chess**

**Created by**  
**Narumedsr Pitayachamrat 6731328921**  
**Nititorn Rattanadech 6731329521**

**2110215 Programming Methodology**  
**Semester 2 Year 2024**

# Task Management

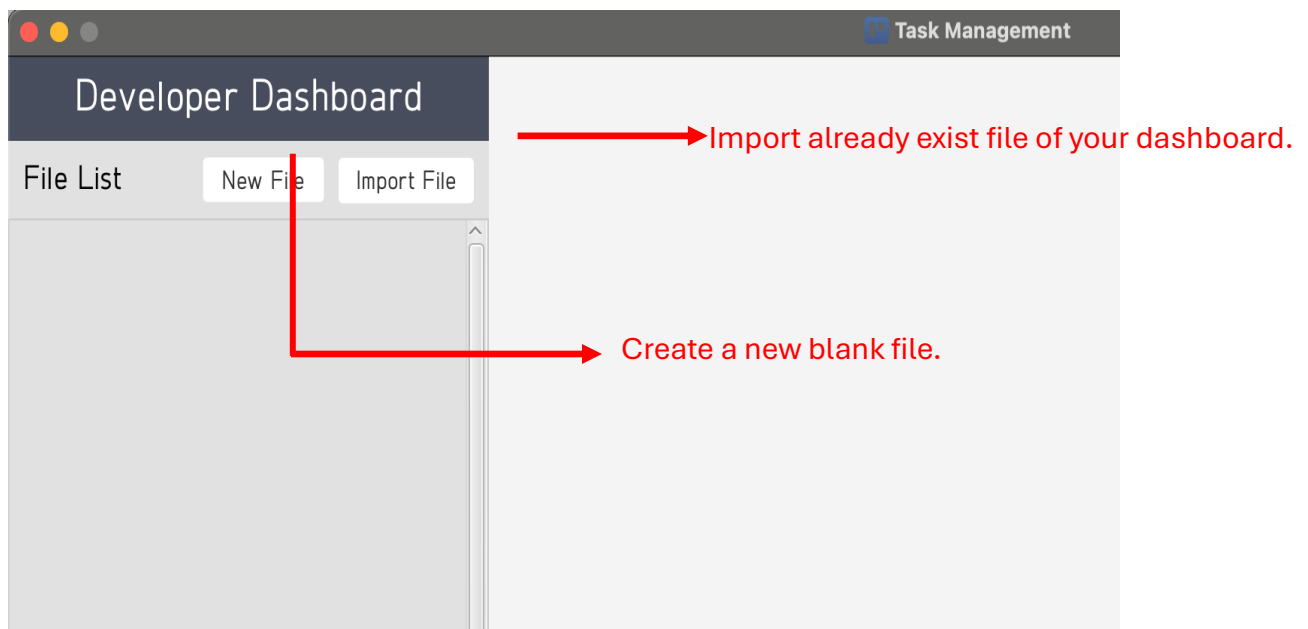
## Introduction

TaskManagement is an app for managing team and personal work schedules effectively. It can be used by a large group of individuals by clearly defining each person's responsibilities. It can also be applied in everyday management such as tracking your weight training program and your studying schedules.

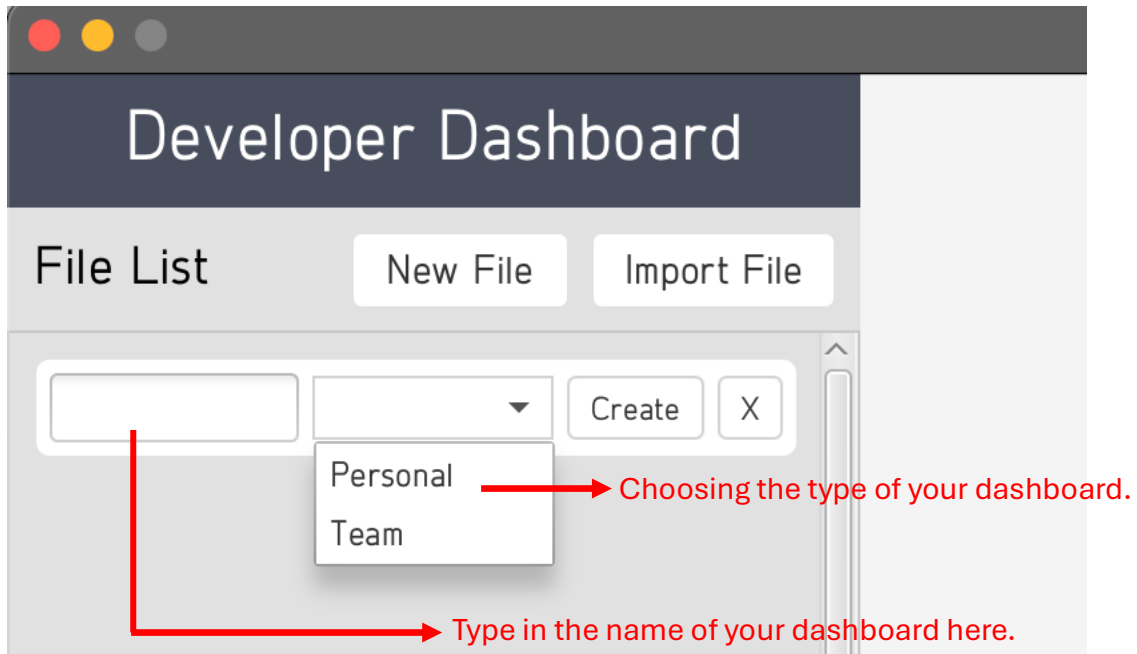
## Instruction

### 1. Create your first dashboard

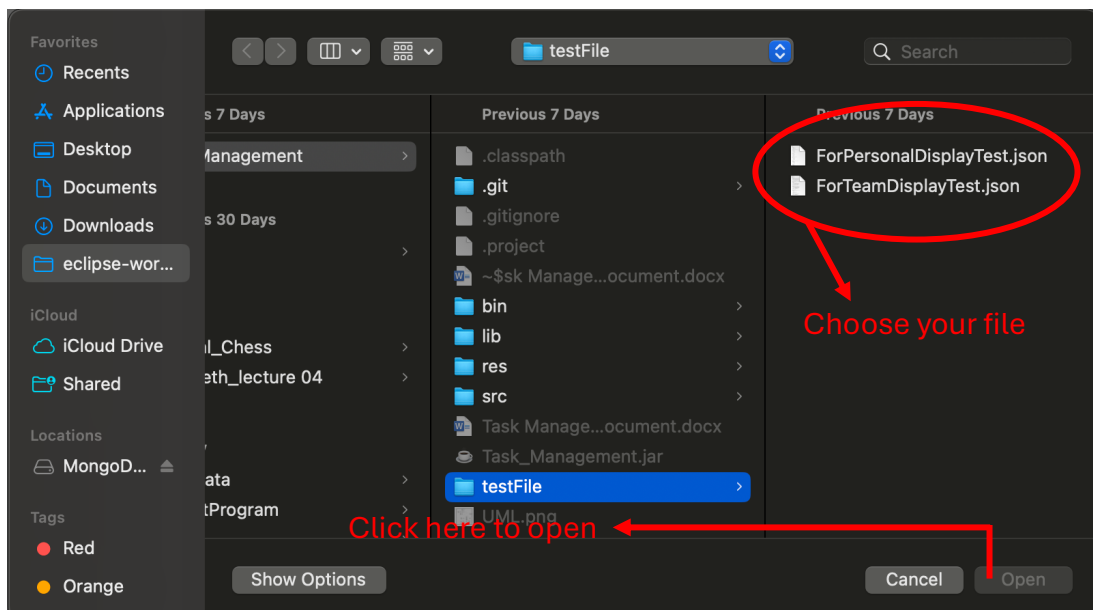
- First you can choose between create a new file or import JSON file to create a dashboard



- If you click create a new file, you must choose between Personal or Team dashboard and include the name of your dashboard in the box.



- If you click to import an already exist file, you must choose the JSON file that you want then click open.



- Example of an already finish personal card

Personal

Manage Label X

Study	Review Math ... X	Physics Practice X	Group Study X	Create Flashc... X	Summarize His... X
	Date 2025-05-01 Checklist <input checked="" type="checkbox"/> 0/2 Description Focus on calculus and algebra topics	Date 2025-05-01 Checklist <input checked="" type="checkbox"/> 0/3 Description Solve 10 physics questions	Date 2025-05-02 Checklist <input checked="" type="checkbox"/> 0/0 Description Review midterm content with classmates	Date 2025-05-02 Checklist <input checked="" type="checkbox"/> 1/2 Description Make cards for memorization-heavy subjects	Date 2025-05-03 Checklist <input checked="" type="checkbox"/> 0/3 Description Create bullet summaries for key events
	Organize Note... X	Prepare Revisi... X	Print Study M... X	Back Up Files X	Declutter Stud... X
	Date 2025-05-01 Checklist <input checked="" type="checkbox"/> 1/2 Description Sort and rename digital files	Date 2025-05-02 Checklist <input checked="" type="checkbox"/> 2/3 Description Create a daily study plan	Date 2025-05-02 Checklist <input checked="" type="checkbox"/> 0/0 Description Print notes, papers, and schedules	Date 2025-05-03 Checklist <input checked="" type="checkbox"/> 1/2 Description Save all exam material to cloud	Date 2025-05-03 Checklist <input checked="" type="checkbox"/> 1/3 Description Clean and organize workspace
	Sleep Routine X	Eat Healthy M... X	Stretching X	Walk Breaks X	Mental Health ... X
Work	Date 2025-05-01 Checklist <input checked="" type="checkbox"/> 0/2 Description Fix bedtime to 10 PM for better focus	Date 2025-05-02 Checklist <input checked="" type="checkbox"/> 1/2 Description Focus on fruits, veggies, and hydration	Date 2025-05-03 Checklist <input checked="" type="checkbox"/> 1/3 Description Stretch every morning before studying	Date 2025-05-04 Checklist <input checked="" type="checkbox"/> 0/0 Description Take a 10-minute walk every hour	Date 2025-05-04 Checklist <input checked="" type="checkbox"/> 0/2 Description Relax and stay balanced during exams
Health					

- Example of an already finish team card

Dev Company Workflow

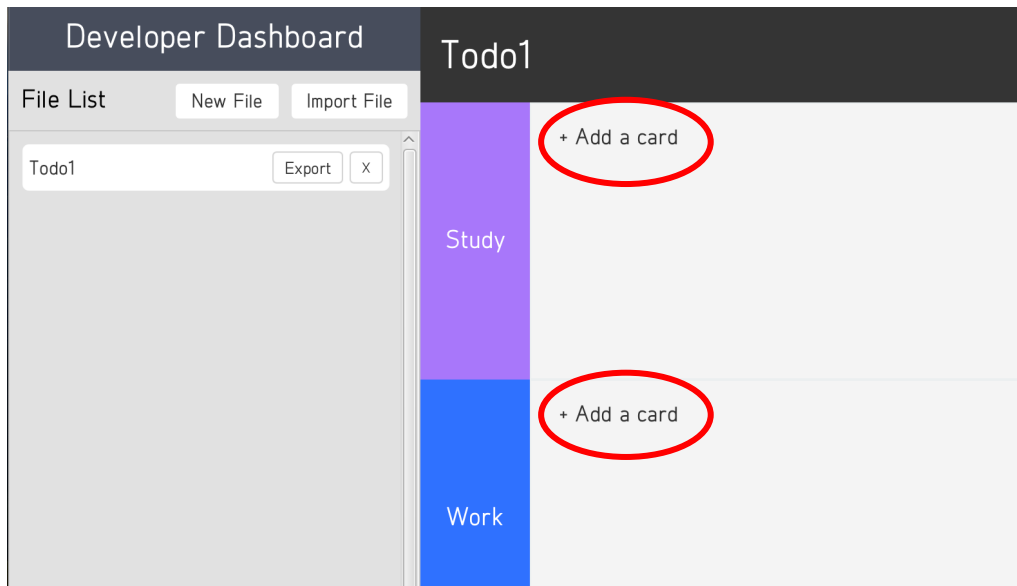
Manage Member X

Planning X	Development X	Testing X	Deployment X
Sprint Kickoff X Date 2025-06-01 3 2/4	Auth Module X Date 2025-06-09 3 1/6	Unit Testing X Date 2025-06-17 2 1/3	CI/CD Pipeline Se... X Date 2025-06-24 3 1/4
Requirements Gat... X Date 2025-06-02 3 1/5	User Profile UI X Date 2025-06-10 3 1/4	Integration Testing X Date 2025-06-18 3 1/4	Staging Deployment X Date 2025-06-25 3 1/3
Architecture Desi... X Date 2025-06-03 3 1/5	Payment Integration X Date 2025-06-11 3 1/7	Load Testing X Date 2025-06-19 2 1/4	User Acceptance ... X Date 2025-06-26 3 0/4
Tech Stack Selec... X Date 2025-06-04 3 1/5	Search Feature X Date 2025-06-12 4 1/6	Security Testing X Date 2025-06-20 3 0/3	Hotfixes in Staging X Date 2025-06-27 3 0/4
Sprint Backlog Cr... X Date 2025-06-05 2 2/3	Notifications Syst... X Date 2025-06-13 3 1/6	Usability Testing X Date 2025-06-21 2 0/5	Final Approval X Date 2025-06-28 2 0/3
Task Estimation X Date 2025-06-06 2 1/3	Data Export X Date 2025-06-14 3 1/6	Bug Triage X Date 2025-06-22 3 1/3	Production Release X Date 2025-06-29 3 0/4
Risk Assessment X Date 2025-06-07 2 1/4	Performance Opti... X Date 2025-06-15 3 1/7	Regression Testing X Date 2025-06-23 2 0/4	
Stakeholder Revi... X Date 2025-06-08 2 1/4	Code Review X Date 2025-06-16 3 1/4		

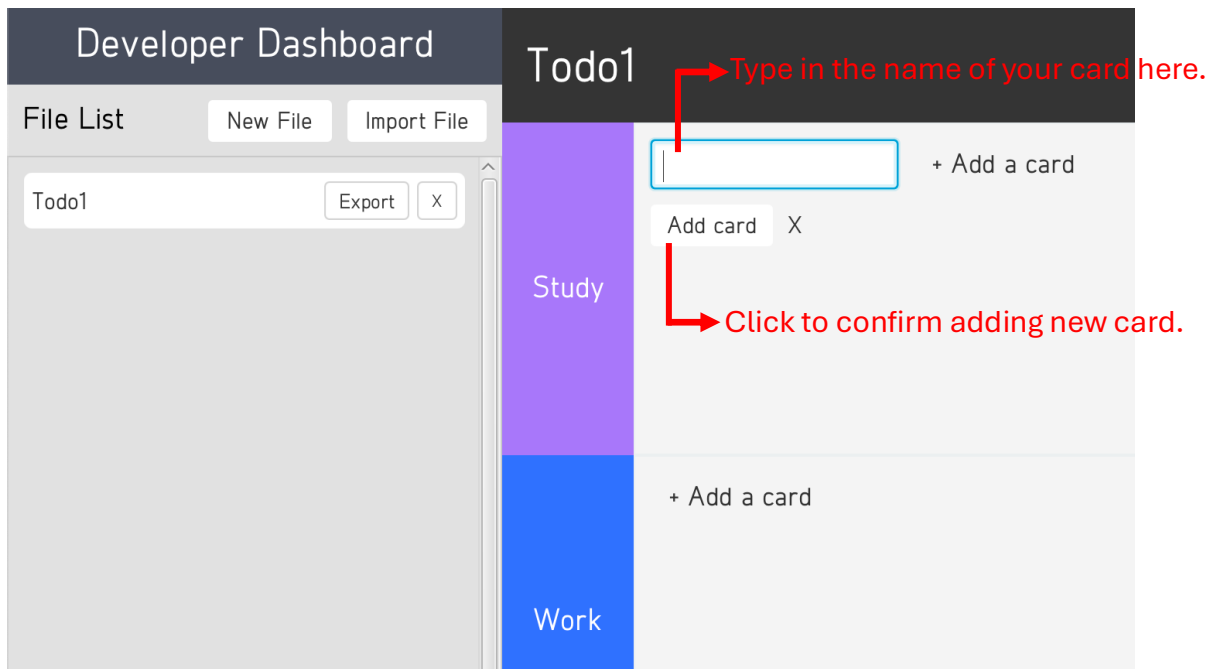
## 2. Creating your first card

### 2.1 Personal card

- Once create a personal dashboard, you can choose what category you want your card to belong to by clicking “+Add a card” button on that category.

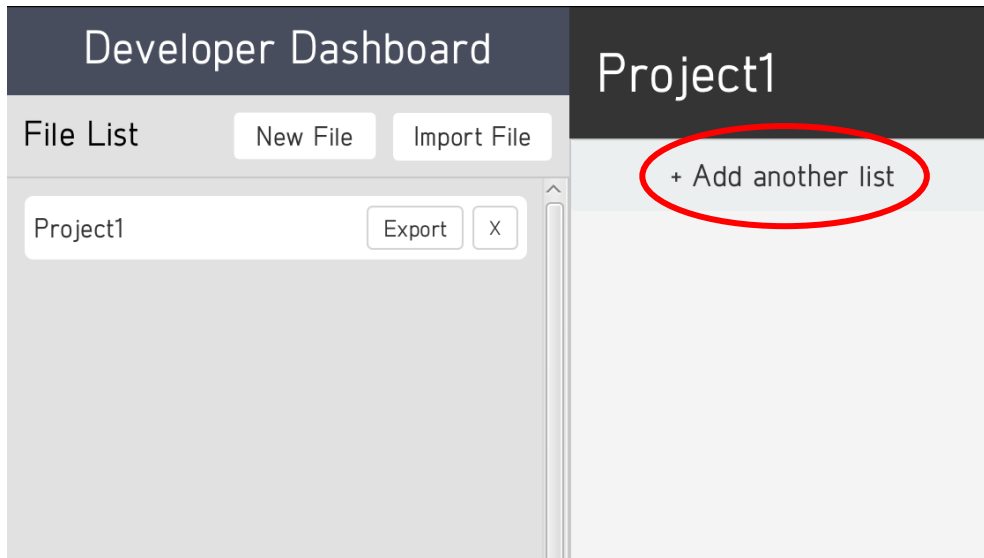


- After clicking “Add a card” button, type in the name of your card in the box. Then press “Add card” button to confirm adding a card.

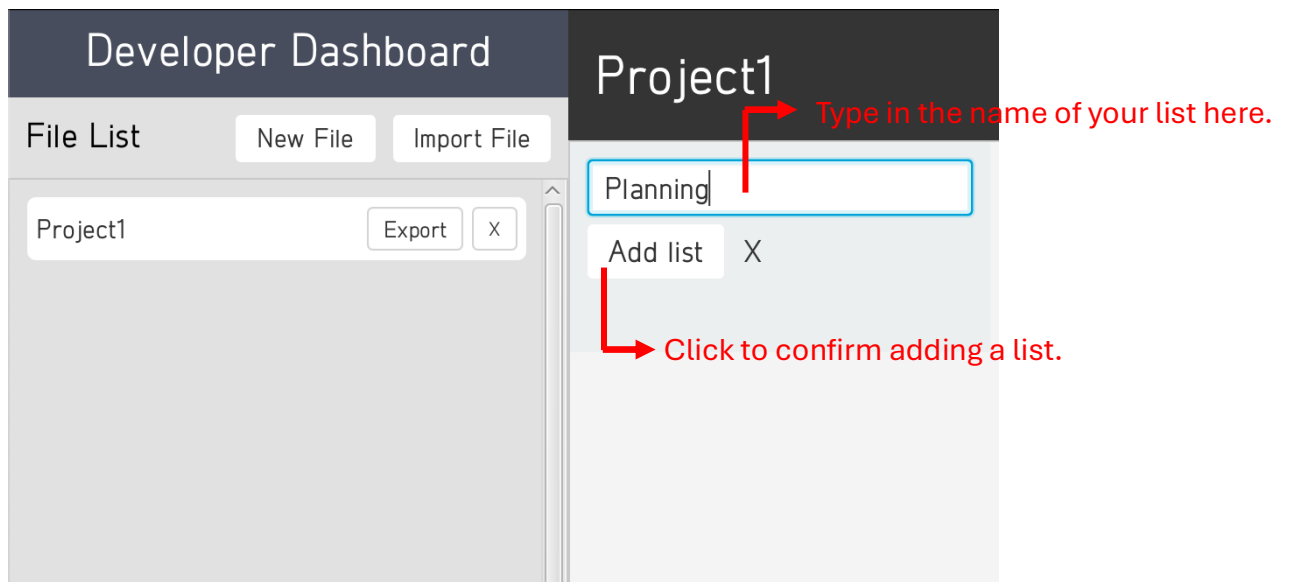


## 2.2 Team card

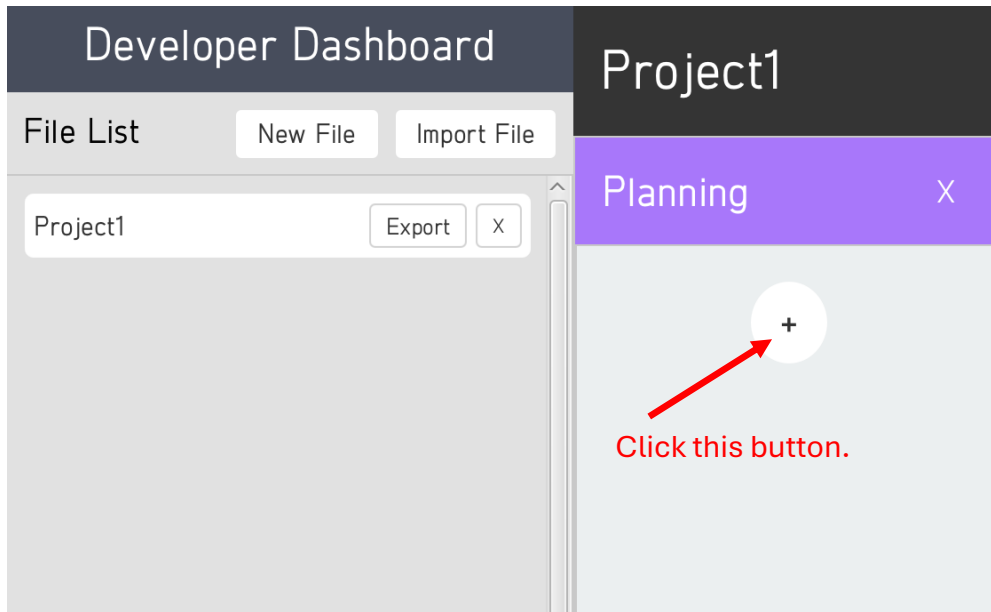
- First create a new list by clicking “Add another list” button.



- Then you must enter your list name in the box and click “Add list” to confirm adding a list.



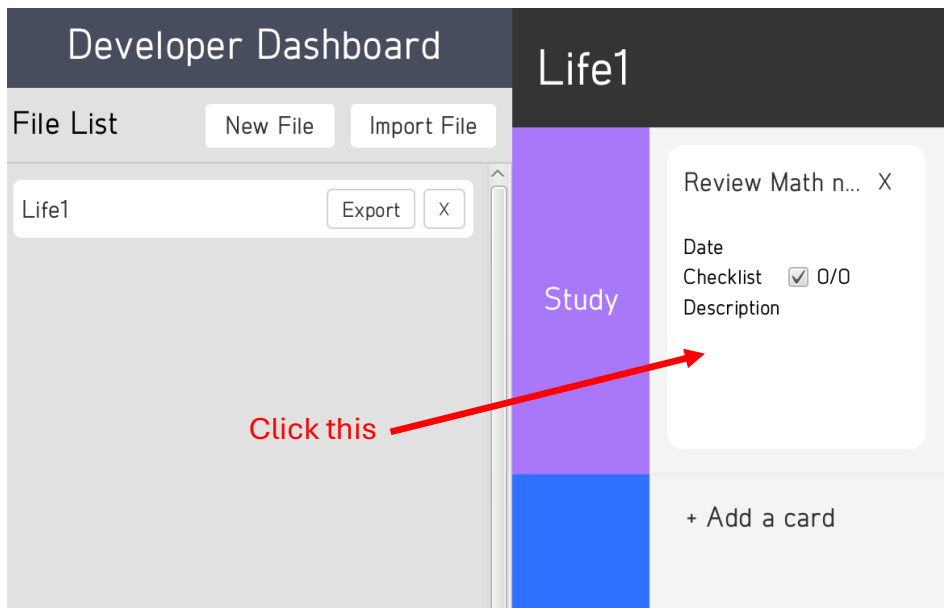
- Then click this button to create a card. Then you can create a team card by using the same method to create a personal card as state in 2.1



### 3. Adjusting your card and dashboard detail

#### 3.1 Personal card

- Click on the card to access the card.



- This is all the things that you can adjusting to suit your use.

**Review Math note**

Label  Edit

Date  Edit

Description  Edit

CheckList  0 % Edit

Red arrows point from the 'Edit' buttons to labels: 'Edit Label', 'Edit Date', 'Edit Description', and 'Edit Checklist'.

- Choose the date to set your deadline for this task. Then click save to confirm.

**Review Math note**

Label  Edit

Date  Save Cancel

Description  Edit

CheckList  0 % Edit

A date picker calendar is open, showing the month of May 2025. The date 10 is highlighted with a blue border. A red circle is drawn around the calendar. Red arrows point from the 'Save' button to 'Click here to save.' and from the date 10 to 'Choose your deadline here.'



- Write the description in this highlight box then press save to confirm.

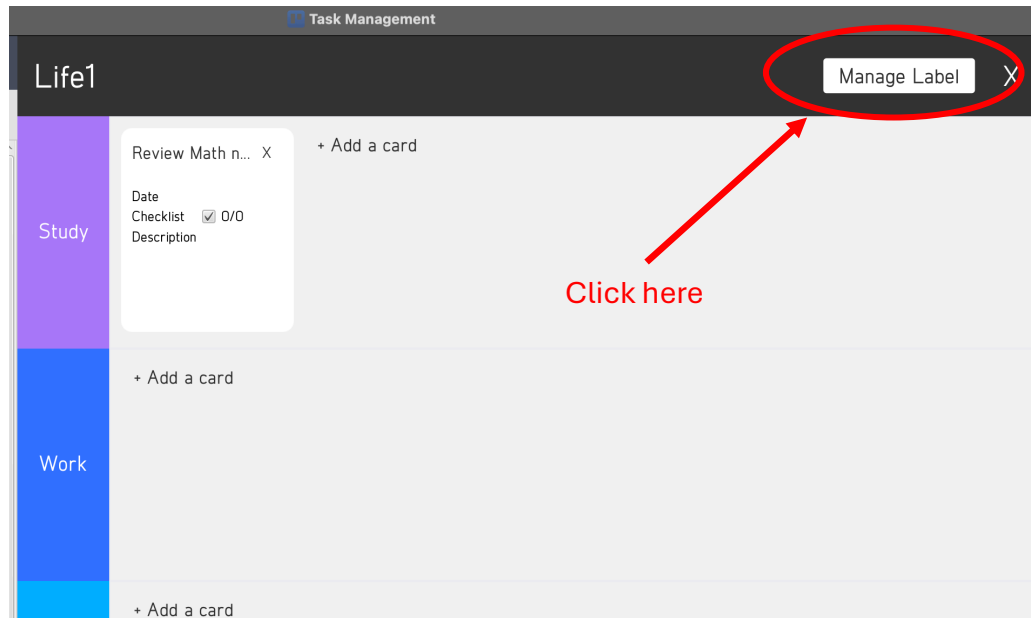
The screenshot shows a web form titled "Review Math note". It contains several input fields and buttons. The "Description" field is a large text area, which is highlighted with a blue border. A red arrow points from the text "Click here to save." to the "Save" button located to the right of the "Description" field. Another red arrow points from the text "Write your description here." to the "Description" text area itself. Other fields include "Label", "Date" (with a calendar icon), "CheckList" (a progress bar showing 0%), and "Edit" buttons. "Save" and "Cancel" buttons are also present near the "Date" and "Description" fields.

- Create a check list by write it in this box then click save to confirm

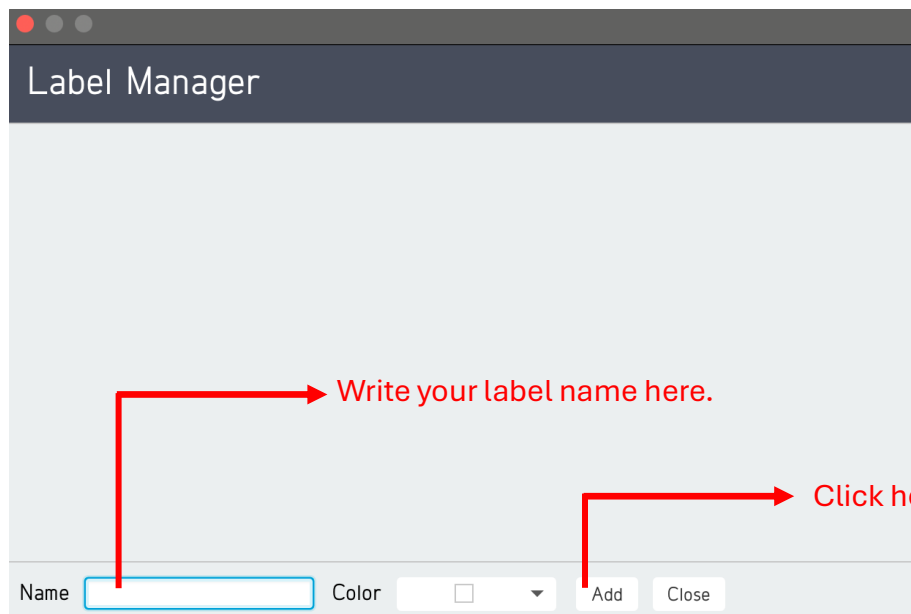
This screenshot shows the same "Review Math note" form, but with annotations for the "CheckList" section. A red arrow points from the text "Write your checklist here." to a small text input box located below the "CheckList" progress bar. Another red arrow points from the text "Click here to add" to the "Add" button, which is positioned to the right of the small text input box. The "CheckList" progress bar still shows 0%. The "Description" field and other buttons remain visible in the background.

- Label

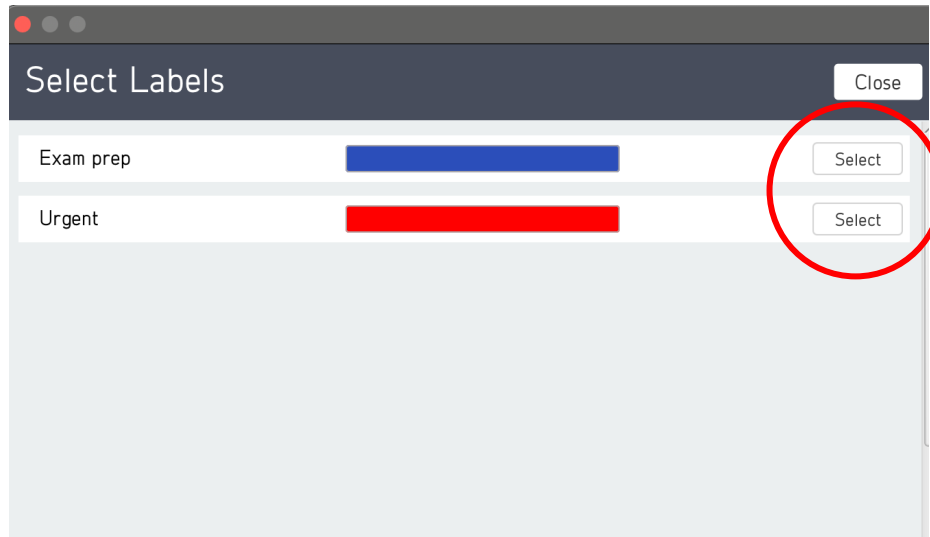
- Create your Label by clicking “Manage Label” button.



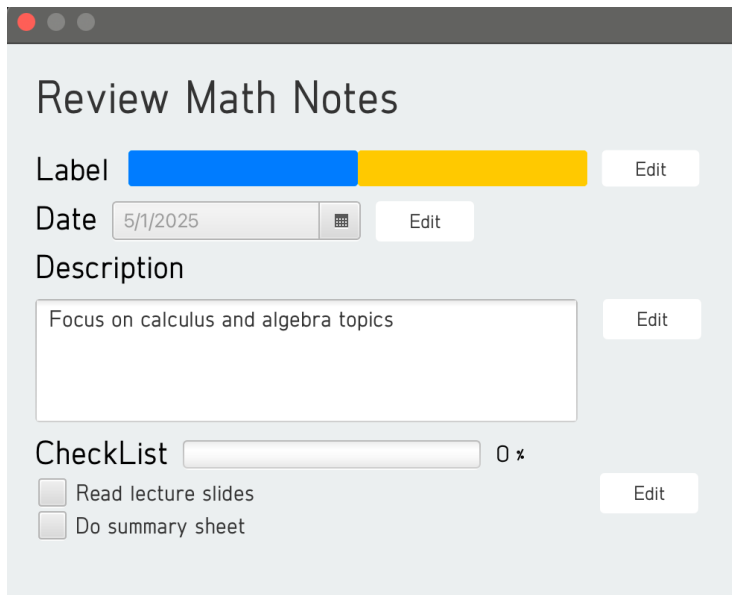
- Write your Label name in the box and choose the color for this label then click add to save.



- Then go back to your card and select the label to display it on your card.

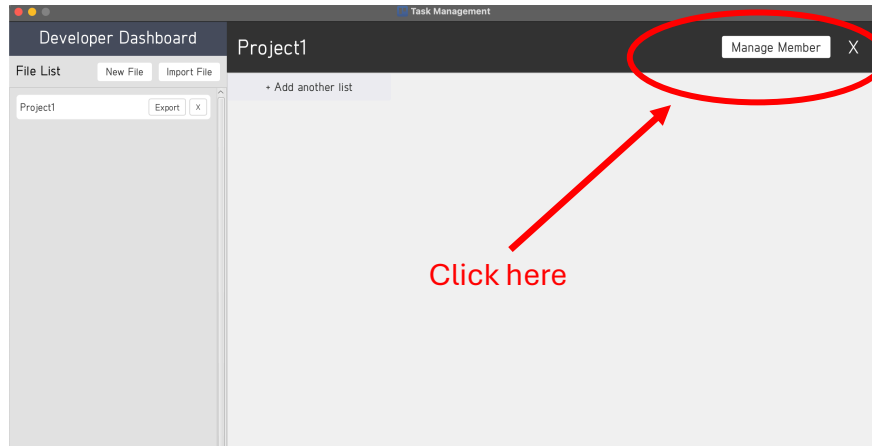


- Example of an already finish card.

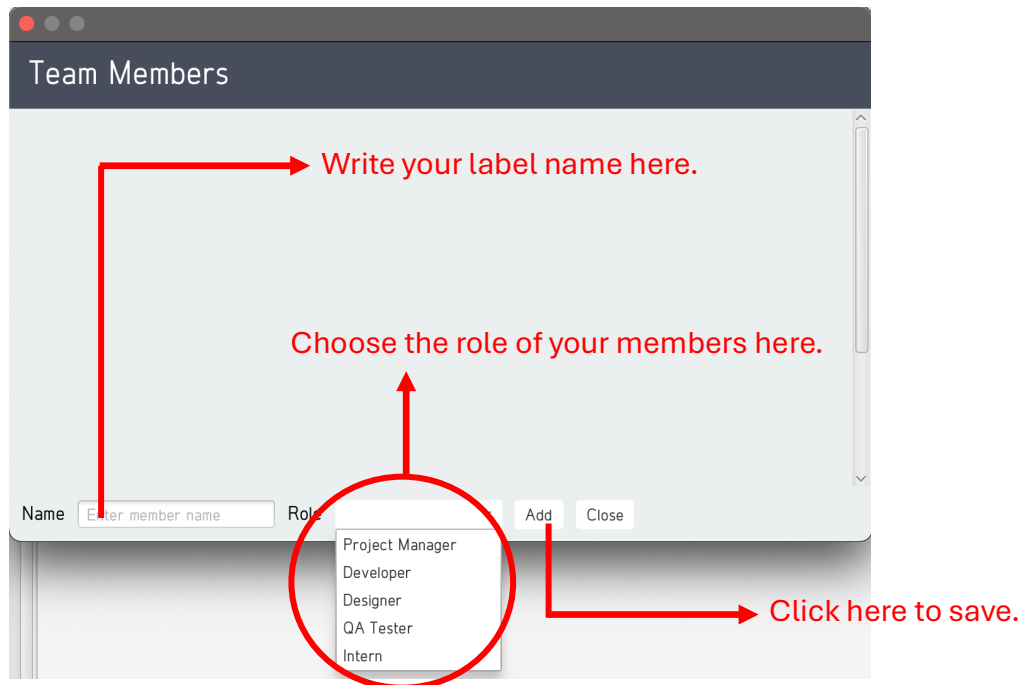


### 3.2 Team card

- Date, Description and Checklist are using the same method to add and edit as a personal card with all state in 3.1
- Member
  - Create your member by clicking “Manage member” button.



- Write your member name in the box, then click add to save.



- Then go back to your card and select the label to display it on your card.

Name	Role	Action
Alice Johnson	Role Project Manager	Deselect
Bob Smith	Role Developer	Deselect
Henry Zhao	Role Designer	Select
Carol Nguyen	Role Developer	Select
Dave Patel	Role QA Tester	Select
Emma Zhang	Role Developer	Select
Frank Liu	Role Intern	Select
Grace Kim	Role Intern	Deselect

- Example of an already finished card

## Sprint Kickoff

**Date** 6/1/2025 Edit

**Description**

Align team on sprint goals, scope, and timelines Edit

**CheckList** 50 %

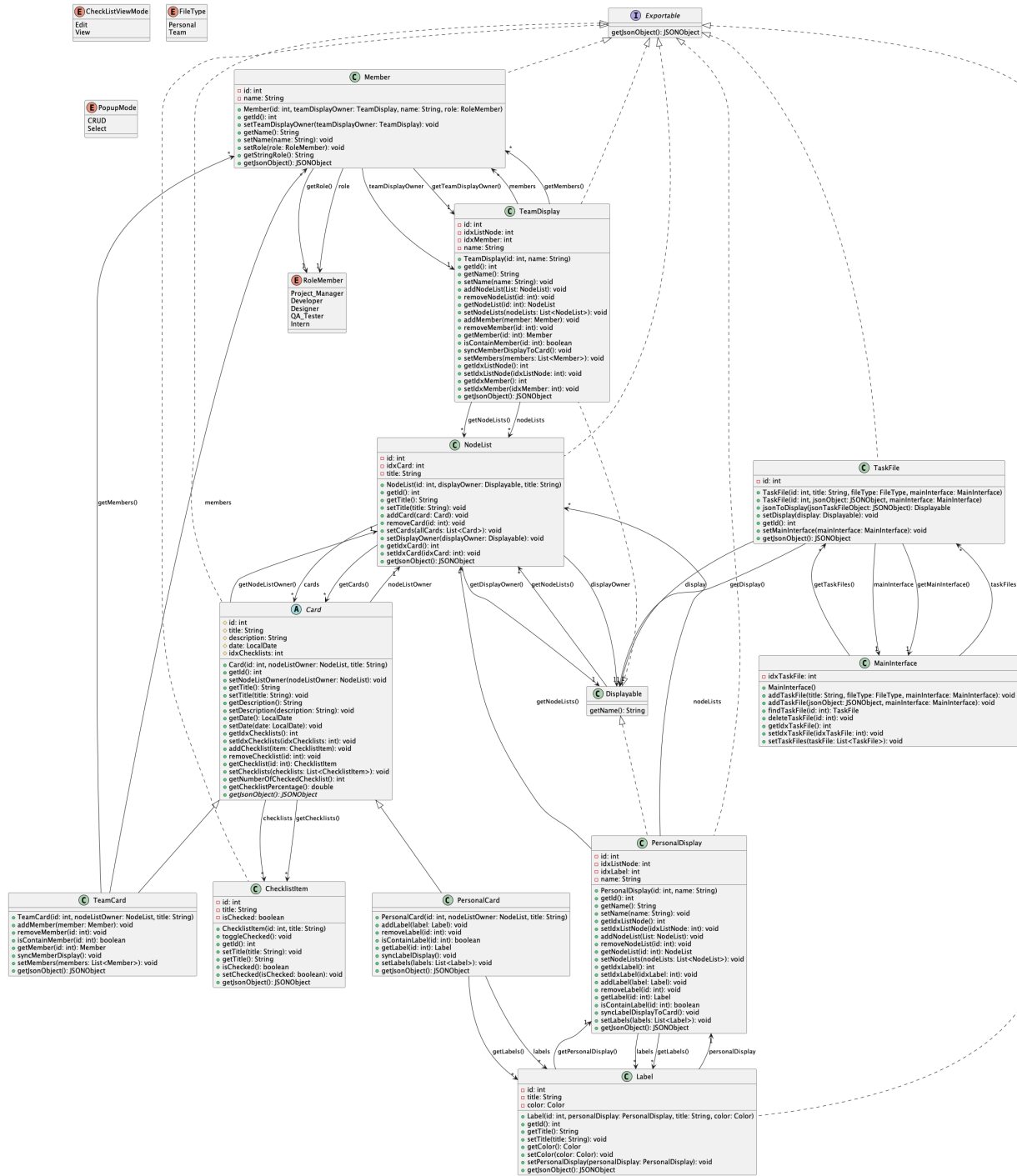
- ☒ Draft agenda Edit
- ☒ Book conference room
- ☐ Invite stakeholders
- ☐ Share pre-reads

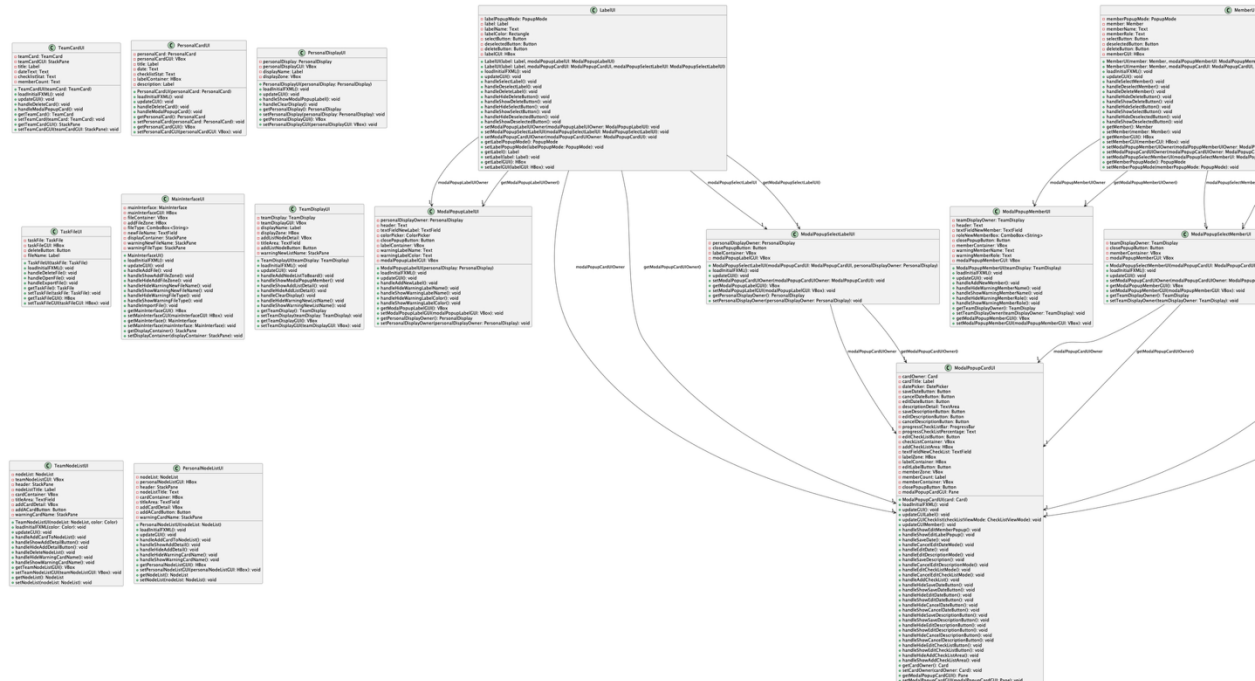
**Members** 3 members

- Alice Johnson Role Project Manager Edit
- Grace Kim Role Intern
- Bob Smith Role Developer

# Class diagram

## Package Models, Interfaces and Enums





## Implementation detail

This project uses the org.json library, which is embedded in the runnable JAR file

## 1.Package models

### 1.1 class MainInterface

### 1.1.1 Fields

- int idxTaskFile	id for new taskfile
- List<TaskFile> taskFiles	keep all Taskfile in the application

### 1.1.2 Methods

+ MainInterface()	Set idxTaskFile to 0 and create a new ArrayList for taskFiles
+ void addTaskFile(String title,FileType fileType, MainInterface mainInterface)	Add a new TaskFile to taskFiles using title, fileType, and mainInterface, then increment idxTaskFile by 1
+ void addTaskFile(JSONObject jsonObject, MainInterface mainInterface)	Add a new TaskFile to taskFiles using jsonObject and mainInterface, then increment idxTaskFile by 1
+ TaskFile findTaskFile(int id)	Return the TaskFile that has the same ID as the argument.

+ void deleteTaskFile(int id)	Delete the TaskFile that has the same ID as the argument.
getters and setters for all variables	

## 1.2 class TaskFile implements Exportable

### 1.2.1 Fields

- int id	Unique ID of this TaskFile.
- MainInterface mainInterface	MainInterface associated with this TaskFile.
- Displayable display	Display representation of this TaskFile.

### 1.2.2 Methods

+ TaskFile(int id, String title, FileType fileType, MainInterface mainInterface)	Initialize id, mainInterface, and display based on fileType
+ TaskFile(int id, JSONObject jsonObject, MainInterface mainInterface)	Initialize id, mainInterface, and display using jsonObject
+ Displayable jsonToDisplay(JSONObject jsonTaskFileObject)	Return a Displayable created from jsonTaskFileObject
+ JSONObject getJsonObject()	Return the JSONObject representation of this TaskFile
getters and setters for all variables except id's setter	

## 1.3 class PersonalDisplay implements Displayable,Exportable

### 1.3.1 Fields

- int id	Unique ID of this PersonalDisplay
- int idxListNode	ID for the new NodeList
- int idxLabel	ID for the new Label
- String name	Name of this PersonalDisplay
- List<NodeList> nodeLists	All NodeList elements of this PersonalDisplay
- List<Label> labels	All Label elements of this PersonalDisplay

### 1.3.2 Methods

+ PersonalDisplay(int id, String name)	Initialize id and name, set idxListNode to 0, and create a new ArrayList for nodeLists
+ void addNodeList(NodeList List)	Add a List to nodeLists



+ void removeNodeList(int id)	Remove the NodeList that has the same id as the argument
+ NodeList getNodeList(int id)	Return the NodeList that has the same id as the argument
+ void addLabel(Label label)	Add a Label to labels
+ void removeLabel(int id)	Remove the Label that has the same id as the argument
+ Label getLabel(int id)	Return the Label that has the same id as the argument
+ boolean isContainLabel(int id)	Check if labels contain a Label with the same id as the argument
+ void syncLabelDisplayToCard()	Call syncLabelDisplay for every card
+ JSONObject getJsonObject()	Return the JSONObject representation of this PersonalDisplay
getters and setters for all variables except id's setter	

## 1.4 class TeamDisplay implements Displayable,Exportable

### 1.4.1 Fields

- int id	Unique ID of this TeamDisplay
- int idxListNode	ID for the new NodeList
- int idxMember	ID for the new Member
- String name	Name of this TeamDisplay
- List<NodeList> nodeLists	All NodeList elements of this TeamDisplay
- List<Member> members	All Member elements of this TeamDisplay

### 1.4.2 Methods

+ TeamDisplay(int id, String name)	Initialize id and name, set idxListNode to 0, and create a new ArrayList for nodeLists
+ void addNodeList(NodeList List)	Add a List to nodeLists and set idxNodeList equal to the id of the List + 1
+ void removeNodeList(int id)	Remove the NodeList that has the same id as the argument
+ NodeList getNodeList(int id)	Return the NodeList that has the same id as the argument
+ void addMember(Member member)	Add a Member to members and set idxMember equal to the id of the Member + 1
+ void removeMember(int id)	Remove the Member that has the same id as the argument

+ Member getMember(int id)	Return the NodeList that has the same id as the argument
+ JSONObject getJsonObject()	Return the JSONObject representation of this TeamDisplay
getters and setters for all variables except id's setter	

## 1.5 class NodeList implements Exportable

### 1.5.1 Fields

# int id	Unique ID of this NodeList
# Displayable displayOwner	Displayable that contains this NodeList
# int idxCard	ID for the new Card
# String title	Title of this NodeList
# List<Card> cards	All Card elements of this NodeList

### 1.5.2 Methods

+ NodeList(int id, Displayable displayOwner, String title)	Initialize id, displayOwner, and title. Set idxCard to 0 and create a new ArrayList for cards
+ void addCard(Card card)	Add a Card to cards
+ void removeCard(int id)	Remove the Card that has the same id as the argument
+ JSONObject getJsonObject()	Return the JSONObject representation of this NodeList
getters and setters for all variables except id's setter	

## 1.6 abstract class Card implements Exportable

### 1.6.1 Fields

- int id	Unique ID of this Card
- NodeList nodeListOwner	NodeList that contains this Card
- String title	Title of this Card
- String description	Description of this Card
- LocalDate date	Due date of this Card
- int idxCheckLists	ID for the new CheckListItem
- List<CheckListItem> checkLists	All CheckListItem elements in this Card

### 1.6.2 Methods

+ Card(int id,NodeList nodeListOwner, String title)	Initialize id, nodeListOwner, and title. Set description to an empty string, date to null, idxChecklists to 0, and create a new ArrayList for checkLists
+ void addChecklist(ChecklistItem item)	Add a CheckListItem to checkLists
+ void removeChecklist(int id)	Remove the CheckListItem that has the same id as the argument
+ CheckListItem getChecklist(int id)	Return the CheckListItem that has the same id as the argument
+ int getNumberOfCheckedChecklist()	Return the number of CheckListItem elements that are checked (complete)
+ double getChecklistPercentage()	Return the percentage of CheckListItem elements that are checked (complete)
+ abstract JSONObject getJsonObject()	Return the JSONObject representation of this Card
getters and setters for all variables except id's setter	

## 1.7 class PersonalCard extends Card

### 1.7.1 Fields

- List<Label> labels	All Label elements in this PersonalCard
----------------------	---

### 1.7.2 Methods

+ PersonalCard(int id,NodeList nodeListOwner, String title)	Initialize id, nodeListOwner, and title to Card. then create a new ArrayList for labels
+ void addLabel(Label label)	Add a label to labels
+ void removeLabel(int id)	Remove the Label that has the same id as the argument
+ boolean isContainLabel(int id)	Check if labels contain a Label with the same id as the argument
+ Label getLabel(int id)	Return the Label that has the same id as the argument
+ void syncLabelDisplay()	Remove any Label whose id is not contained in the labels of the PersonalDisplay associated with this PersonalCard
+ JSONObject getJsonObject()	Return the JSONObject representation of this PersonalCard
getters and setters for all variables	

## 1.8 class TeamCard extends Card

### 1.8.1 Fields

- List<Member> members	All Member elements in this TeamCard
------------------------	--------------------------------------

### 1.8.2 Methods

+ TeamCard(int id,NodeList nodeListOwner, String title)	Initialize id, nodeListOwner, and title to Card. then create a new ArrayList for members
+ void addMember(Member member)	Add a member to members
+ void removeMember(int id)	Remove the Member that has the same id as the argument
+ boolean isContainMember(int id)	Check if members contain a Member with the same id as the argument
+ Member getMember(int id)	Return the Member that has the same id as the argument
+ void syncMemberDisplay()	Remove any Member whose id is not contained in the members of the TeamDisplay associated with this TeamCard
+ JSONObject getJsonObject()	Return the JSONObject representation of this TeamCard
getters and setters for all variables	

## 1.9 class CheckListItem implements Exportable

### 1.9.1 Fields

- int id	Unique ID of this Member
- String title	Title of this CheckListItem
- boolean isChecked	Indicates the checked status of this CheckListItem

### 1.9.2 Methods

+ ChecklistItem(int id,String title)	Initialize id, and title. Then set isChecked to false
+ void toggleChecked()	Toggles the checked status of this CheckListItem
+ JSONObject getJsonObject()	Return the JSONObject representation of this CheckListItem
getters and setters for all variables except id's getter	

## 1.10 class Member implements Exportable

### 1.10.1 Fields

- int id	Unique ID of this Member
- TeamDisplay teamDisplayOwner	TeamDisplay that contains this Member
- String name	Name of this Member
- RoleMember role	Role of this Member

### 1.10.2 Methods

+ Member(int id, TeamDisplay teamDisplayOwner, String name, RoleMember role)	Initialize id, teamDisplayOwner, name and role
+ String getStringRole()	Returns the role as String
+ JSONObject getJsonObject()	Return the JSONObject representation of this Member
getters and setters for all variables except id's getter	

## 1.11 class Label implements Exportable

### 1.11.1 Fields

- int id	Unique ID of this Label
- PersonalDisplay personalDisplayOwner	PersonalDisplay that contains this Label
- String title	Title of this Label
- Color color	Color of this Label

### 1.11.2 Methods

+ Label(int id, PersonalDisplay personalDisplayOwner, String title, Color color)	Initialize id, personalDisplayOwner, title and color
+ JSONObject getJsonObject()	Return the JSONObject representation of this Label
getters and setters for all variables except id's getter	

## 2.Package enums

### 2.1 enum FileType

Personal	use for TaskFile that contain PersonalDisplay
Display	use for TaskFile that contain TeamDisplay

## 2.2 enum CheckListViewMode

Edit	Edit the checklist in ModalPopupCard
View	View the checklist in ModalPopupCard

## 2.3 enum PopupMode

CRUD	Popup used for Create, Read, Update, and Delete operations
Select	Popup used for selection purposes

## 2.4 enum RoleMember

Project_Manager	Project Manager role of Member
Developer	Developer role of Member
Designer	Designer role of Member
QA_Tester	QA Tester role of Member
Intern	Intern role of Member

## 3.Package interfaces

### 3.1 interface Displayable

#### 3.1.1 Methods

+ String getName()	return name of this Displayable
+ List<NodeList> getNodeLists	return all NodeList of this Displayable

### 3.2 interface Exportable

#### 3.2.1 Methods

+ JSONObject getJsonObject()	Return the JSONObject representation of this Exportable
------------------------------	---

## 4.Package controllers

### 4.1 class MainInterfaceUI

#### 4.1.1 Fields

- MainInterface mainInterface	Model of this class
- @FXML HBox mainInterfaceGUI	Root UI of mainInterface
- @FXML VBox fileContainer	Container for TaskFile
- @FXML HBox addFileZone	Area to create a new TaskFile
- @FXML ComboBox<String> fileType	Selector for the fileType of the new TaskFile
- @FXML TextField newFileName	Input field for the name of the new TaskFile
- @FXML StackPane displayContainer	Area to show DisplayableUI
- @FXML StackPane warningNewFileName	Warning when the new file name is empty
- @FXML StackPane warningFileType	Warning when the fileType is not selected

#### 4.1.2 Methods

+ MainInterfaceUI()	Initialize a new MainInterface and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder and set up the UI
+ void updateGUI()	Load all current TaskFile elements and open the DisplayableUI of the TaskFile with ID Main.taskFileOpening
+ void handleAddFile()	Add a new TaskFile to MainInterface using the values from fileType and newFileName, and show warningNewFileName or warningFileType for validation errors
+ void handleShowAddFileZone()	Show the addFileZone (by setting visible and managed to true)
+ void handleHideAddFileZone()	Hide the addFileZone and clear the values in newFileName and fileType
+ void handleHideWarningNewFileName()	Hide warningNewFileName
+ void handleShowWarningNewFileName()	Show warningNewFileName
+ void handleHideWarningFileType()	Hide warningFileType
+ void handleShowWarningFileType()	Show warningFileType
+ void handleImportFile()	Add a new TaskFile by importing a JSON file
getters and setters for mainInterface, mainInterfaceGUI and displayContainer	

### 4.2 class TaskFileUI

### 4.2.1 Fields

- TaskFile taskFile	Model of this class
- @FXML HBox taskFileGUI	Root UI of taskFile
- @FXML Button deleteButton	Button for delete this TaskFile from MainInterface
- @FXML Label fileName	Show name of this TaskFile

### 4.2.2 Methods

- TaskFileUI(TaskFile taskFile)	Set taskFile using given taskFile and load the initial FXML layout
- void loadInitialFXML()	Load the FXML file from the resources folder and set up the UI
- void handleDeleteFile()	Delete this TaskFile from MainInterface and updateGUI of MainInterfaceUI
- void handleOpenFile()	Open this TaskFile in the DisplayContainer
- void handleExportFile()	Export this TaskFile using FileChooser and the getJsonObject method from the TaskFile model
- getters and setters for taskFile and taskFileGUI	

## 4.3 class PersonalDisplayUI

### 4.3.1 Fields

- PersonalDisplay personalDisplay	Model of this class
- @FXML VBox personalDisplayGUI	Root UI of personalDisplay
- @FXML Label displayName	Show name of this PersonalDisplay
- @FXML VBox displayZone	Area used to show all NodeList elements in this PersonalDisplay

### 4.3.2 Methods

- PersonalDisplayUI(PersonalDisplay personalDisplay)	Set personalDisplay using given personalDisplay and load the initial FXML layout
- void loadInitialFXML()	Load the FXML file from the resources folder, set up the UI and add 3 NodeList Study, Work and Health to personalDisplay
- void updateGUI()	Display 3 NodeListUI components, each with a different color
- void handleShowModalPopupLabel()	Show the ModalPopupLabelUI of this PersonalDisplay



- void handleClearDisplay()	Set Main.taskFieldOpening to -1 and clear DisplayContainer children
- getters and setters for personalDisplay and personalDisplayGUI	

## 4.4 class TeamDisplayUI

### 4.4.1 Fields

- TeamDisplay teamDisplay	Model of this class
- @FXML VBox teamDisplayGUI	Root UI of teamDisplay
- @FXML Label displayName	Show name of this teamDisplay
- @FXML HBox displayZone	Area used to show all NodeList elements in this TeamDisplay
- @FXML VBox addListNodeDetail	Area used to create new NodeList
- @FXML TextField titleArea	Input field for the name of the new NodeList
- @FXML Button addListNodeButton	Button for show addListNodeDetail
- @FXML StackPane warningNewListName	Warning when the new NodeList name is empty

### 4.4.2 Methods

+ TeamDisplayUI(TeamDisplay teamDisplay)	Set teamDisplay using given teamDisplay and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder set up the UI to show name of teamDisplay
+ void updateGUI()	Load all current NodeList elements in teamDisplay
+ void handleAddNodeListToBoard()	Add a new NodeList to teamDisplay using the values from titleArea, and show warningNewListName for validation errors
+ void handleShowModalPopupMember()	Show the ModalPopupMemberUI of this teamDisplay
+ void handleShowAddListDetail()	Show AddListNodeDetail and hide addListNodeButton
+ void handleHideAddListDetail()	Hide AddListNodeDetail, show addListNodeButton, and reset titleArea
+ void handleClearDisplay()	Set Main.taskFieldOpening to -1 and clear DisplayContainer children
+ void handleHideWarningNewListName()	Hide warningNewListName (by setting visible and managed to false)

+ void handleShowWarningNewListName()	Show warningNewListName
getters and setters for teamDisplay and teamDisplayGUI	

## 4.5 class PersonalNodeListUI

### 4.5.1 Fields

- NodeList nodeList	Model of this class
- @FXML HBox nodeListGUI	Root UI of nodeList
- @FXML StackPane header	StackPane that contain nodeListTitle
- @FXML Text nodeListTitle	Show name of nodeList
- @FXML HBox cardContainer	Area used to show all Card elements in this nodeList
- @FXML TextField titleArea	Input field for the name of the new Card
- @FXML VBox addCardDetail	Area used to create new Card
- @FXML Button addACardButton	Button for show addCardDetail
- @FXML StackPane warningCardName	Warning when the new Card name is empty

### 4.5.2 Methods

+ PersonalNodeListUI(NodeList nodeList)	Set nodeList using given nodeList and load the initial FXML layout
+ void loadInitialFXML()	Set UI using nodeList and load the initial FXML layout
+ void updateGUI()	Load all current PersonalCard elements in nodeList
+ void handleAddCardToNodeList()	Add a new Card to nodeList using the values from titleArea, and show warningNewCardName for validation errors and updateGUI
+ void handleShowAddDetailButton()	Show addCardDetail (by setting visible and managed to true) and hide addACardButton
+ void handleHideAddDetailButton()	Hide addCardDetail and show addACardButton
+ void handleHideWarningCardName()	Hide warningCardName
+ void handleShowWarningCardName()	Show warningCardName
getters and setters for personalNodeList and personalNodeListGUI	

## 4.6 class TeamNodeListUI

#### 4.6.1 Fields

- NodeList nodeList	Model of this class
- @FXML VBox nodeListGUI	Root UI of nodeList
- @FXML StackPane header	StackPane that contain nodeListTitle and deleteButton
- @FXML Label nodeListTitle	Show name of nodeList
- @FXML VBox cardContainer	Area used to show all Card elements in this nodeList
- @FXML TextField titleArea	Input field for the name of the new Card
- @FXML VBox addCardDetail	Area used to create new Card
- @FXML Button addACardButton	Button for show addCardDetail
- @FXML StackPane warningCardName	Warning when the new Card name is empty

#### 4.6.2 Methods

+ TeamNodeListUI(NodeList nodeList, Color color)	Set nodeList using given nodeList and load the initial FXML layout
+ void loadInitialFXML(Color color)	Set UI using given color and nodeList then load the initial FXML layout
+ void updateGUI()	Load all current TeamCard elements in nodeList
+ void handleAddCardToNodeList()	Add a new Card to nodeList using the values from titleArea, and show warningNewCardName for validation errors and updateGUI
+ void handleShowAddDetailButton()	Show addCardDetail and hide addACardButton
+ void handleHideAddDetailButton()	Hide addCardDetail and show addACardButton
+ void handleDeleteNodeList()	Delete nodeList from displayOwner and updateGUI of MainInterfaceUI
+ void handleHideWarningCardName()	Hide warningCardName
+ void handleShowWarningCardName()	Show warningCardName
getters and setters for teamNodeList and teamNodeListGUI	

### 4.7 class PersonalCardUI

#### 4.7.1 Fields

- PersonalCard personalCard	Model of this class
- @FXML VBox personalCardGUI	Root UI of personalCard

- @FXML Label title	Show title of personalCard
- @FXML Text date	Show date of personalCard
- @FXML Text checklistStat	Show number of checked items out of the total checklist items, e.g. 2/3
- @FXML HBox labelContainer	Area used to show all Card elements in this nodeList
- @FXML Label description	Show description of personalCard

#### 4.7.2 Methods

+ PersonalCardUI(PersonalCard personalCard)	Set personalCard using given personalCard and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder and call updateGUI
+ void updateGUI()	Update the UI for title, date, checklistStat, description and add Rectangle for each label in personalCard to the labelContainer
+ void handleDeleteCard()	Delete personalCard from nodeListOwner and updateGUI of MainInterfaceUI
+ void handleModalPopupCard()	Show the ModalPopupCardUI of this personalCard
getters and setters for personalCard and personalCardGUI	

### 4.8 class TeamCardUI

#### 4.8.1 Fields

- TeamCard teamCard	Model of this class
- @FXML StackPane teamCardGUI	Root UI of teamCard
- @FXML Label title	Show title of personalCard
- @FXML Text dateText	Show date of personalCard
- @FXML Text checklistStat	Show number of checked items out of the total checklist items, e.g. 2/3
- @FXML Text memberCount	Show number of member of personal

#### 4.8.2 Methods

+ TeamCardUI(TeamCard teamCard)	Set teamCard using given teamCard and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder and call updateGUI

+ void updateGUI()	Update the UI for title, date, checklistStat, and memberCount
+ void handleDeleteCard()	Delete teamCard from nodeListOwner and updateGUI of MainInterfaceUI
+ void handleModalPopupCard()	Show the ModalPopupCardUI of this teamCard
getters and setters for teamCard and teamCardGUI	

## 4.9 class MemberUI

### 4.9.1 Fields

- ModalPopupMemberUI modalPopupMemberUIOwner	modalPopupMemberUI of this MemberUI (use when MemberPopupMode is CRUD)
- ModalPopupSelectMemberUI modalPopupSelectMemberUI	modalPopupSelectMemberUI of this MemberUI (use when MemberPopupMode is Select)
- ModalPopupCardUI modalPopupCardUIOwner	modalPopupCardUI of this MemberUI (use when MemberPopupMode is Select)
- PopupMode memberPopupMode	PopupMode(enum) of this MemberUI
- Member member	Model of this class
- @FXML Text memberName	Show name of member
- @FXML Text memberRole	Show role of member
- @FXML Button selectButton	Button for add member to teamCard of modalPopupCardUIOwner
- @FXML Button deselectedButton	Button for remove member from teamCard of modalPopupCardUIOwner
- @FXML Button deleteButton	Button for delete member to teamCard of modalPopupCardUIOwner
- @FXML HBox memberGUI	Root UI of member

### 4.9.2 Methods

+ MemberUI(Member member, ModalPopupMemberUI modalPopupMemberUI)	Set member and modalPopupMemberUI using given argument and setMemberPopupMode to CRUD then load the initial FXML layout
+ MemberUI(Member member, ModalPopupCardUI modalPopupCardUI, ModalPopupSelectMemberUI modalPopupSelectMemberUI)	Set member,modalPopupCardUI and modalPopupSelectMember using given argument and setMemberPopupMode to Select then load the initial FXML layout

+ void loadInitialFXML()	Load the FXML file from the resources folder and call updateGUI
+ void updateGUI()	Update the UI for memberName and memberRole, and show the delete button when memberPopupMode is CRUD, or show the select and deselect buttons when memberPopupMode is Select
+ void handleSelectMember()	Add member to teamCard of modalPopupcardUIOwner and updateGUI objects that use this data
+ void handleDeselectMember()	Remove member from teamCard of modalPopupCardUIOwner and updateGUI objects that use this data
+ void handleDeleteMember()	Delete member to teamCard of modalPopupCardUIOwner and updateGUI objects that use this data
+ void handleHideDeleteButton()	Hide deleteButton
+ void handleShowDeleteButton()	Show deleteButton
+ void handleHideSelectButton()	Hide selectButton
+ void handleShowSelectButton()	Show selectButton
+ void handleHideDeselectedButton()	Hide deselectButton
+ void handleShowDeselectedButton()	Show deselectButton
getters and setters for member, memberGUI, modalPopupMemberUIOwner, modalPopupCardUIOwner, modalPopupSelectMemberUI and memberPopupMode	

## 4.10 class LabelUI

### 4.10.1 Fields

- ModalPopupLabelUI modalPopupLabelUIOwner	modalPopupLabelUI of this LabelUI (use when LabelPopupMode is CRUD)
- ModalPopupSelectLabelUI modalPopupSelectLabelUI	modalPopupSelectLabelUI of this LabelUI (use when MemberPopupMode is Select)
- ModalPopupCardUI modalPopupCardUIOwner	modalPopupCardUI of this LabelUI (use when LabelPopupMode is Select)
- PopupMode labelPopupMode	PopupMode(enum) of this LabelUI
- Label label	Model of this class
- @FXML Text labelName	Show name of label
- @FXML Rectangle labelColor	Show color of label

- @FXML Button selectButton	Button for add label to personalCard of modalPopupCardUIOwner
- @FXML Button deselectedButton	Button for remove label from personalCard of modalPopupCardUIOwner
- @FXML Button deleteButton	Button for delete label to personalCard of modalPopupCardUIOwner
- @FXML HBox labelGUI	Root UI of label

#### 4.10.2 Methods

+ LabelUI(Label label, ModalPopupLabelUI)	Set label and modalPopupLabelUI using given argument and set MemberPopupMode to CRUD then load the initial FXML layout
+ LabelUI(Label label, ModalPopupCardUI modalPopupCardUI)	Set label,modalPopupCardUI and modalPopupSelectLabel using given argument and set LabelPopupMode to Select then load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder and call updateGUI
+ void updateGUI()	Update the UI for labelName and labelColor, and show the delete button when labelPopupMode is CRUD, or show the select and deselect buttons when labelPopupMode is Select
+ void handleSelectLabel()	Add label to personalCard of modalPopupCardUIOwner and updateGUI objects that use this data
+ void handleDeselectLabel()	Remove label from personalCard of modalPopupCardUIOwner and updateGUI objects that use this data
+ void handleDeleteLabel()	Delete label to personal of modalPopupCardUIOwner and updateGUI objects that use this data
+ void handleHideDeleteButton()	Hide deleteButton
+ void handleShowDeleteButton()	Show deleteButton
+ void handleHideSelectButton()	Hide selectButton
+ void handleShowSelectButton()	Show selectButton
+ void handleHideDeselectedButton()	Hide deselectButton
+ void handleShowDeselectedButton()	Show deselectButton
getters and setters for label, labelGUI,modalPopupLabelUIOwner, modalPopupCardUIOwner,	

modalPopupSelectLabelUI and labelPopupMode	
---	--

## 4.11 class ModalPopupCardUI

### 4.11.1 Fields

- Card cardOwner	Card model of this ModalPopupCardUI
- @FXML Label cardTitle	Title of cardOwner
- @FXML DatePicker datePicker	Show or edit date of cardOwner
- @FXML Button saveDateButton	Button to change cardOwner's date to the value selected in the datePicker
- @FXML Button cancelDateButton	Button to cancel edit mode in the datePicker and reset it to show the current date of the cardOwner
- @FXML Button editDateButton	Button to enable the datePicker, allowing the user to change its value
- @FXML TextArea descriptionDetail	Show and edit description of cardOwner
- @FXML Button saveDescriptionButton	Button to change cardOwner's description to the text in the descriptionDetail
- @FXML Button editDescriptionButton	Button to cancel edit mode in descriptionDetail and reset it to show the current description of the cardOwner
- @FXML Button cancelDescriptionButton	Button to enable descriptionDetail, allowing the user to change its value
- @FXML ProgressBar progressCheckListBar	Show progress of cardOwner's checklist in progress bar
- @FXML Text progressCheckListPercentage	Show progress of cardOwner's checklist in percentage
- @FXML Button editCheckListButton	Button to enable checkListContainer, by show addCheckListArea and allow the user to add and delete checklist of cardOwner
- @FXML VBox checkListContainer	Area used to show all CheckList elements in this cardOwner
- @FXML HBox addCheckListArea	Area used to create new CheckList
- @FXML TextField textFieldNewCheckList	Input field for name of the new CheckList
- @FXML HBox labelZone	Area used to show label information
- @FXML HBox labelContainer	Area used to show all CheckList elements in this cardOwner
- @FXML Button editLabelButton	Button to show ModalPopupSelectLabelUI of this cardOwner
- @FXML VBox memberZone	Area used to show member information



- @FXML Label memberCount	Show number of member in cardOwner
- @FXML VBox memberContainer	Area used to show all Member elements in this cardOwner
- @FXML Button closePopupButton	Button for close this UI
- @FXML Pane modalPopupCardGUI	Root UI of this class

#### 4.11.2 Methods

- ModalPopupCardUI(Card card)	Set cardOwner using given card and load the initial FXML layout
- void loadInitialFXML()	Load the FXML file from the resources folder, set up UI especially - personalCard show label - teamCard show member and call updateGUI
- void updateGUI()	Update the UI for cardTitle, datePicker, descriptionDetail, and call updateGUICheckList in view mode, updateGUIMember and updateGUILabel
- void updateGUILabel()	If cardOwner is personalCard add label to labelContainer
- void updateGUIChecklist (CheckListViewMode checkListViewMode)	Add CheckBox using field from checklist in cardOwner and add to checkListContainer and show closeButton only when checkListViewMode is Edit then update checkListPercentage
- void updateGUIMember()	If cardOwner is teamCard add member to memberContainer
- void handleShowEditMemberPopup()	If cardOwner is teamCard then show the ModalPopupSelectMemberUI of this cardOwner
- void handleShowEditLabelPopup()	If cardOwner is personalCard then show the ModalPopupSelectLabelUI of this cardOwner
- void handleSaveDate()	Set cardOwner's date using value from datePicker and updateGUI objects that use this data then call - handleHideSaveDescriptionButton - handleHideSaveDateButton - handleShowEditDescriptionButton
- void handleCancelEditDateMode()	Set editable of descriptionDetail to false then updateGUI and call - handleHideSaveDescriptionButton

	<ul style="list-style-type: none"> <li>- handleHideSaveDateButton</li> <li>- handleShowEditDescriptionButton</li> </ul>
- void handleEditDate()	Disable datePicker and cell <ul style="list-style-type: none"> <li>- handleHideEditDateButton</li> <li>- handleShowSaveDateButton</li> <li>- handleShowCancelDateButton</li> </ul>
- void handleEditDescriptionMode()	Set editable of descriptionDetail to true and call <ul style="list-style-type: none"> <li>- handleHideEditDescriptionButton</li> <li>- handleShowSaveDescriptionButton</li> <li>- handleShowCancelDescriptionButton</li> </ul>
- void handleSaveDescription()	set cardOwner's description to descriptionDetail's text, set editable of descriptionDetail to true, updateGUI then call <ul style="list-style-type: none"> <li>- handleHideSaveDescriptionButton</li> <li>- handleHideCancelDescriptionButton</li> <li>- handleShowEditDescriptionButton</li> </ul>
- void handleCancelEditDescriptionMode()	set editable of descriptionDetail to false, updateGUI then call <ul style="list-style-type: none"> <li>- handleHideSaveDescriptionButton</li> <li>- handleHideCancelDescriptionButton</li> <li>- handleShowEditDescriptionButton</li> </ul>
- void handleEditCheckListMode()	updateGUICheckList in Edit mode and call handleHideEditCheckListButton and handleShowAddCheckListArea
- void handleCancelEditCheckListMode()	updateGUICheckList in Edit mode and call handleShowEditCheckListButton and handleHideAddCheckListArea
- void handleAddCheckList()	add new ChecklistItem using field from textFieldNewCheckList to cardOwner and updateGUI then call <ul style="list-style-type: none"> <li>- handleShowEditCheckListButton</li> <li>- handleHideAddCheckListArea</li> </ul>
- void handleHideSaveDateButton()	Hide saveDateButton
- void handleShowSaveDateButton()	Show saveDateButton
- void handleHideEditDateButton()	Hide editDateButton
- void handleShowEditDateButton()	Show editDateButton
- void handleHideCancelDateButton()	Hide cancelDateButton
- void handleShowCancelDateButton()	Show cancelDateButton
- void handleHideSaveDescriptionButton()	Hide saveDescriptionButton
- void handleShowSaveDescriptionButton()	Show saveDescriptionButton
- void handleHideEditDescriptionButton()	Hide editDescriptionButton

- void handleShowEditDescriptionButton()	Show editDescriptionButton
- void handleHideCancelDescriptionButton()	Hide cancelDescriptionButton
- void handleShowCancelDescriptionButton()	Show cancelDescriptionButton
- void handleHideEditCheckListButton()	Hide editCheckListButton
- void handleShowEditCheckListButton()	Show editCheckListButton
- void handleHideAddCheckListArea()	Hide addCheckListArea
- void handleShowAddCheckListArea()	Show addCheckListArea
getters and setters for cardOwner and setModalPopupCardGUI	

## 4.12 class ModalPopupMemberUI

### 4.12.1 Fields

- TeamDisplay teamDisplayOwner	TeamDisplay model of this ModalPopupMemberUI
- @FXML Text header	Show word "Select Members"
- @FXML TextField textFieldNewMember	Input field for the name of the new Member
- @FXML ComboBox<String> roleNewMemberBox	Selector for the role of the new Member
- @FXML Button closePopupButton	Button for close this UI
- @FXML VBox memberContainer	Area used to show all Member elements in teamDisplayOwner
- @FXML Text warningMemberName	Warning when new member name is empty
- @FXML Text warningMemberRole	Warning when the role is not selected
- @FXML VBox modalPopupMemberGUI	Root UI of this class

### 4.12.2 Methods

+ ModalPopupMemberUI(TeamDisplay teamDisplay)	Set teamDisplayOwner using given teamDisplay and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder, set up UI and call updateGUI
+ void updateGUI()	Show all current Member elements of teamDisplayOwner in memberContainer
+ void handleAddNewMember()	Add a new Member to teamDisplayOwner using the values from textFieldNewMember, roleNewMemberBox, and show warningMemberName or warningMemberRole for validation errors and updateGUI
+ void handleHideWarningMemberName()	Hide warningMemberName
+ void handleShowWarningMemberName()	Show warningMemberName
+ void handleHideWarningMemberRole()	Hide warningMemberRole
+ void handleShowWarningMemberRole()	Show warningMemberRole

getters and setters for teamDisplayOwner and modalPopupMemberGUI	
--	--

## 4.13 class ModalPopupLabelUI

### 4.13.1 Fields

- PersonalDisplay personalDisplayOwner	PersonalDisplay model of this ModalPopupLabelUI
- @FXML Text header	Show word "Select Labels"
- @FXML TextField textFieldNewLabel	Input field for the name of the new Label
- @FXML ColorPicker colorPicker	Selector for the color of the new Label
- @FXML Button closePopupButton	Button for close this UI
- @FXML VBox labelContainer	Area used to show all Label elements in personalDisplayOwner
- @FXML Text warningLabelName	Warning when the new label name is empty
- @FXML Text warningLabelRole	Warning when the role is not selected
- @FXML VBox modalPopupLabelGUI	Root UI of this class

### 4.13.2 Methods

+ ModalPopupLabelUI(PersonalDisplay personalDisplay)	Set personalDisplayOwner using given personalDisplay and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder, set up UI and call updateGUI
+ void updateGUI()	Show all current Label elements of personalDisplayOwner in labelContainer
+ void handleAddNewLabel()	Add a new Label to personalDisplayOwner using the values from textFieldNewLabel, roleNewLabelBox, and show warningLabelName or warningLabel Role for validation errors and updateGUI
+ void handleHideWarningLabelName()	Hide warningLabelName
+ void handleShowWarningLabelName()	Show warningLabelName
+ void handleHideWarningLabelColor()	Hide warningLabelRole
+ void handleShowWarningLabelColor()	Show warningLabelRole
getters and setters for personalDisplayOwner and modalPopupLabelGUI	

## 4.14 class ModalPopupSelectMemberUI

#### 4.14.1 Fields

- ModalPopupCardUI modalPopupCardUIOwner	modalPopupCardUI of this ModalPopupSelectMemberUI
- TeamDisplay teamDisplayOwner	TeamDisplay model of this ModalPopupSelectMemberUI
- @FXML Button closePopupButton	Button for close this UI
- @FXML VBox memberContainer	Area used to show all Member elements in teamDisplayOwner
- @FXML VBox modalPopupMemberGUI	Root UI of this class

#### 4.14.2 Methods

+ ModalPopupSelectMemberUI (ModalPopupCardUI modalPopupCardUI, TeamDisplay teamDisplayOwner)	Set teamDisplayOwner and modalPopupCardUI using given argument and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder, set up UI and call updateGUI
+ void updateGUI()	Show all current Member elements of teamDisplayOwner in memberContainer
getters and setters for teamDisplayOwner , modalPopupMemberGUI and modalPopupCardUIOwner	

### 4.15 class ModalPopupSelectLabelUI

#### 4.15.1 Fields

- ModalPopupCardUI modalPopupCardUIOwner	modalPopupCardUI of this ModalPopupSelectLabelUI
- PersonalDisplay personalDisplayOwner	PersonalDisplay model of this ModalPopupSelectLabelUI
- @FXML Button closePopupButton	Button for close this UI
- @FXML VBox labelContainer	Area used to show all Label elements in personalDisplayOwner
- @FXML VBox modalPopupLabelGUI	Root UI of this class

#### 4.15.2 Methods

+ ModalPopupSelectLabelUI (ModalPopupCardUI modalPopupCardUI, PersonalDisplay personalDisplayOwner)	Set personalDisplayOwner and modalPopupCardUI using given argument and load the initial FXML layout
+ void loadInitialFXML()	Load the FXML file from the resources folder, set up UI and call updateGUI

+ void updateGUI()	Show all current Label elements of personalDisplayOwner in labelContainer
getters and setters for personalDisplayOwner , modalPopupLabelGUI and modalPopupCardUIOwner	

## 5.Package app

### 5.1 class Main

#### 5.1.1 Fields

- MainInterfaceUI mainInterfaceUI	MainInterfaceUI of this application
- Stage primaryStage	PrimaryStage of this application
- int taskFieldOpening	Id of the current TaskFile that is displayed in displayContainer, initialized to -1

#### 5.1.2 Methods

+ void start(Stage primaryStage)	set primaryStage, mainInterfaceUI and load Font EkkamaiNew-Bold. Then setup primaryStage and show primaryStage
+ void main(String[] args)	launch(args)