# Workaholic Technical Guide

Contents

[Workaholic Technical Guide 1](#_Toc35953063)

[1 Introduction 1](#_Toc35953064)

[2 UML 2](#_Toc35953065)

[2.1 Use Case Diagrams 2](#_Toc35953066)

[2.2 Class Diagrams 3](#_Toc35953067)

[2.3 Sequence Diagrams 48](#_Toc35953068)

QRCode

JQM

Database objects

## 1 Introduction

This document is describing processes and architecture in place to provide functionality to the Workaholic application.

This guide uses UML (Universal Modelling Language) to document processes and structure to provided diagrams for discussion. This UML includes Use Case, Class and Sequence diagrams.

## 2 UML

### 2.1 Use Case Diagrams

In Workaholic there are two main use cases, the control individual members have over a team can be controlled by various properties on a team. The allow-edit team allows members of team to edit team data directly allow users to change name, description and configuration settings. The allow-others-invite property allows members to invite other users to a team. The property allow -add-task enables members to add tasks to a team and edit tasks they have made previously. The following diagrams represent the most (Figure 2) and (Figure 1) least restricted team member permissions.

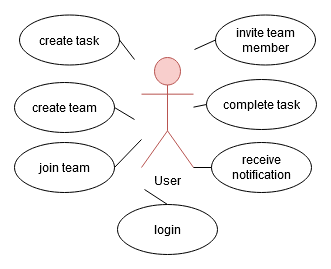


Figure 1 least restrictive team properties

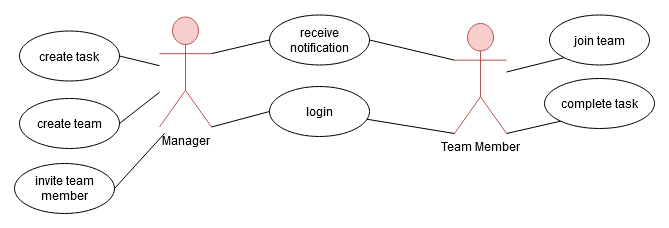


Figure 2 Most restrictive team properties

### 2.2 Class Diagrams

#### 2.2a Query Lists

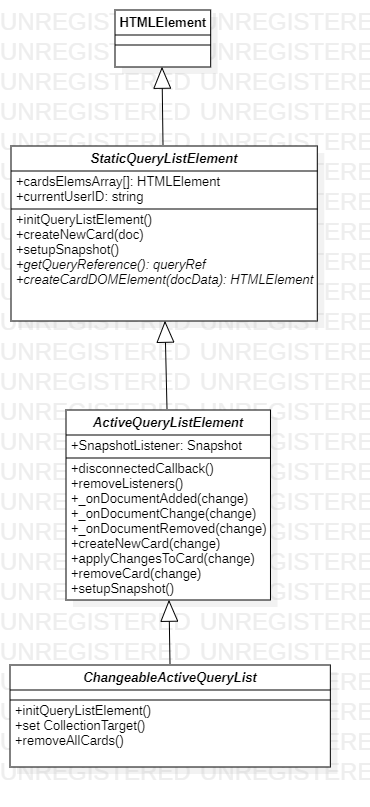


Figure 3 class diagram of query lists

StaticQueryListElement, ActiveQueryListElement and ChangeableActiveQueryList are all intended as abstract classes. The diagram above (Figure 1) shows the relationships of these classes and therefore how functionality is built up. The methods getQueryReference and createCardDOMElement must be fully defined in order to control what data is fetched and creation of list elements to display the data.

##### 2.2a.i StaticQueryListElement

Designed as an abstract this class requires additional definition in order to have a fully functioning element. This class defines how a collection of static documents can be fetched and presented in the DOM. This type of list will form a static collection of data at the point of connection to another element. The following tables document operations and attributes included in the StaticQueryListElement class.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [cardsElemsArray[]](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\98261c39128659303f451ddb742831bb.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\ed20779d2529b66a3d3b167e15b70b09.html)[] | Collection of cards the list is displaying |
| public | [currentUserID](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\9223a342d6747ca8916da0ce004e43ef.html) | string | The current users UUID |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [initQueryListElement()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\02d555fcec097da4890eda4768e33acd.html) | Setup the array that is storing card elements and execute the query to populate the list collection. |
| public | [createNewCard(doc)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\89d7481d3b6c5868db31fbe3645ba572.html) | Create card and add it to the collection and append to DOM. |
| public | [setupSnapshot()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\291658038ba520baf22535195f42cb59.html) | Most basic implementation simply executes on query promise success to generate a card for each returned document. |
| public | [getQueryReference(): queryRef](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\d0d3cc555a854bcdf0de4457cb1d91c2.html) | Extend this method to create a populated query. This is used by setupSnapshot to target specific collecitons of documents. |
| public | [createCardDOMElement(docData): HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\296166ea436addba8f318c577fdcc65a.html) | Extend this method to generate the HTMLElement card required for the particular list.(this should use a card base class) |

The create new card method (Figure 2) is used by any extension of this class to represent query data in the form of document cards. By assignation of the document being represented a card can display any information required by the interface.



Figure 4 Create new card

##### 2.2a.ii ActiveQueryListElement

This class builds on functionality of StaticQueryListElement in order to create an active list of documents. Proper extension of this class allows classes to manage display of a list of documents through the creation, update and removal of list elements. The following tables document operations and attributes included in the ActiveQueryListElement class.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [SnapshotListener](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\cf4b1842bcf99572f82f5f3fd5c79027.html) | Snapshot | Object that is listening to changes made to a collection. |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [disconnectedCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\52fbe2d520115318fd49a50d2b02a1d3.html) | Clean up data when removed from the DOM. |
| public | [removeListeners()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\dbaf5b516f41783214e8e1506f7b7b4b.html) | Remove any listeners on the current snapshot |
| public | [\_onDocumentAdded(change)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\7feeab2503d8a0274d4a9535f4b4df53.html) | Triggered by snapshot’s added call-back. New card is created, added to the DOM and spliced into the local array. |
| public | [\_onDocumentChange(change)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\3b41e4f27fd763efa3b6e440eb85c060.html) | call by snapshot modified call-back. Re-sets the cards document object. |
| public | [\_onDocumentRemoved(change)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\0d772862003790d55d2798e62fa0afec.html) | called by snapshot removed call-back |
| public | [createNewCard(change)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\a7f4fc1ba91daeaa0fd2cd8e6af2b267.html) | Works out where in the DOM this data should be represented. Triggers card creation and splices into the internal array at the position it was added to the DOM. |
| public | [applyChangesToCard(change)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\531848d52bf397dd2a6e7d6a6fb6e142.html) | Works out which card is representing this data and calls setAttributesFromDoc on it to mirror document properties with DOM attributes. |
| public | [removeCard(change)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\469cf45e2c7fec76fded0dbe9a5bab58.html) | Works out which card is representing the document that was removed and removed it from the DOM and internal list. |
| public | [setupSnapshot()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\09a27d99a575fb0b03f716b4e6675a17.html) | Creates the snapshot from the query and applies listeners to additions modifications and removals of any document in the query result. |

In order to respond to live changes in data the setupShapshot method is overridden (Figure 3). This provides additional functionality by appending listeners for changes to the query. This way any changes to documents or any additional data added after querying can be represented in the interface. The handlers call createNewCard, removeCard and applyChangesToCard to enact interface changes. 

Figure 5 setupSnapshot override

The createNewCard method is overridden by this class (Figure 4). As this list actively display of query contents this method will insert cards into the DOM and internal array at specific positions. This allows documents added to a collection to be reflected in the UI if they form part of a query result.



Figure 6 active create new cards method

If data is removed from a query by either changing a value or deleting a document so it no longer forms part of the query results this element needs to respond to this. For example when a task is completed it is no longer picked up by the query on the taskList. This class handles this via a remove change call-back calling the removeCard method (Figure 5). It is important to splice this object array to accurately change the correct card.



Figure 7 Remove Card Method

The final method called by the change handlers is applyChangesToCard (Figure 6). This method is responsible for triggering display changes to reflect changes to documents. By reassigning the document the card is the responsible for displaying this information change.



Figure 8 applyChangesToCard method

##### 2.2a.iii ChangeableActiveQueryList

This class utilities functionality defined by ActiveQueryListElement to create, update and remove list elements. Extensions provided by this class allow the target collection of the query to be changed continually through setting of the collection target displaying different collections of documents. The following tables document operations included in the ChangeableActiveQueryList class.

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [initQueryListElement()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\640c753db6b1080682856d10173ad1bc.html) | Overridden to obscured super to prevent query triggering before target is selected. Queries from this element are instead triggered through property changes. |
| public | [set collectionTarget()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\1cd5f9c859ca2456c209b2e93e25297d.html) | When target is changed all listeners and card elements are remove and a new card collection is formed. |
| public | [removeAllCards()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\5dfa42a9c75034a3c9d2d8c52d68baa5.html) | Remove any cards currently attached to this list |

Changing the data being displayed by the list elements query is done through the set collectionTarget operation (Figure 7). When setting this value the list is rebuilt with a new query to retrieve the data.



Figure 9 set collection target operation

#### 2.2b Document Cards

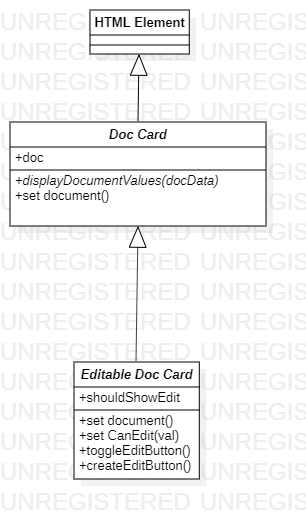


Figure 10 Relationships between Document cards

Document cards are abstract classes used to display document data. The class diagram (Figure 8) documents the relationship between document cards that builds up the functionality. Display updates are controlled setting of the document property. A concrete class formed from these classes needs definition for specific properties to be updated and how they are presented to the user.

##### 2.2b.i DocCard

The DocCard class is an abstract class designed to present document data. Display is updated by setting the document being viewed. In most implementations’ documents are set by an object derived from a StaticQuerylist but can be set in any way. When a document is set it triggers displayDocumentValues to update the display of data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [doc](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\0f0ffddacf11fdea8bf54471096f8d50.html) | Firebase document | firebase document that holds data about a document and the state of the current query result. |

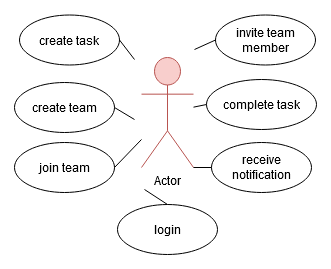
|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [displayDocumentValues(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\2e52ea717a00ef6374f6c6bf789d5bd0.html) | extend this method to display relevant document fields on individual document cards |
| public | [set document()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\44e3543e280530a791f045ae59970551.html) | Store the document in a local variable and trigger display of relevant document fields. |

##### 2.2b.ii EditableDocCard

This class builds on functionality by adding methods to control the display of an edit button. Two examples of classes that utilise functionality defined in both this and a normal DocCard (2.2bi) are utilised in this application these are the task (2.2c.iii) and team cards (2.2b.ii)\*\*\*. The following tables document the attributes and operations defined in the EditableDocCard class.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [shouldShowEdit](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\ce549b1611a7c7b09eaa9080abe9a7ec.html) | none | reflect value as HTML attribute and toggle the edit button |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [set document()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\6f0e59b38294cf45eba96bae2be26ad5.html) | override of set document to include checking whether to show the edit button |
| public | [set CanEdit(val)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\04c14c6b7cfd4eee6615b4b5f71b8cfc.html) | Set local variable to value. Trigger display or removal of edit button from document card. |



This class overrides the document setter (Figure 9) found in the parent class to add additional functionality to check whether the card should show an edit button. Tasks and teams can both be edited if the teams are set up in a way that enables it.



Figure 11 override document setter

Can Edit is used to trigger UI changes to display or remove the edit button on a card. This forces property and attribute reflection when changing this value. This is called by all extensions of this class by assignation of document changes.



Figure 12 can change operation

Toggle edit button is used to remove or add the button from the interface. Extended classes are expected to define their own createEditButton to allow for buttons to be created differently. This function is called when updating canEdit to make sure changes are up to date allow property modifications to be updated.



#### 2.2b Teams Page

The team page is a system responsible for displaying and updating a display of a user’s teams. A query is used to populate a list of skills that the currently logged in user has access to. This system is responsible for displaying team cards and providing UI to add and edit teams as well as invite new members. The class diagram (Figure 9) describes relationships to provide this functionality.

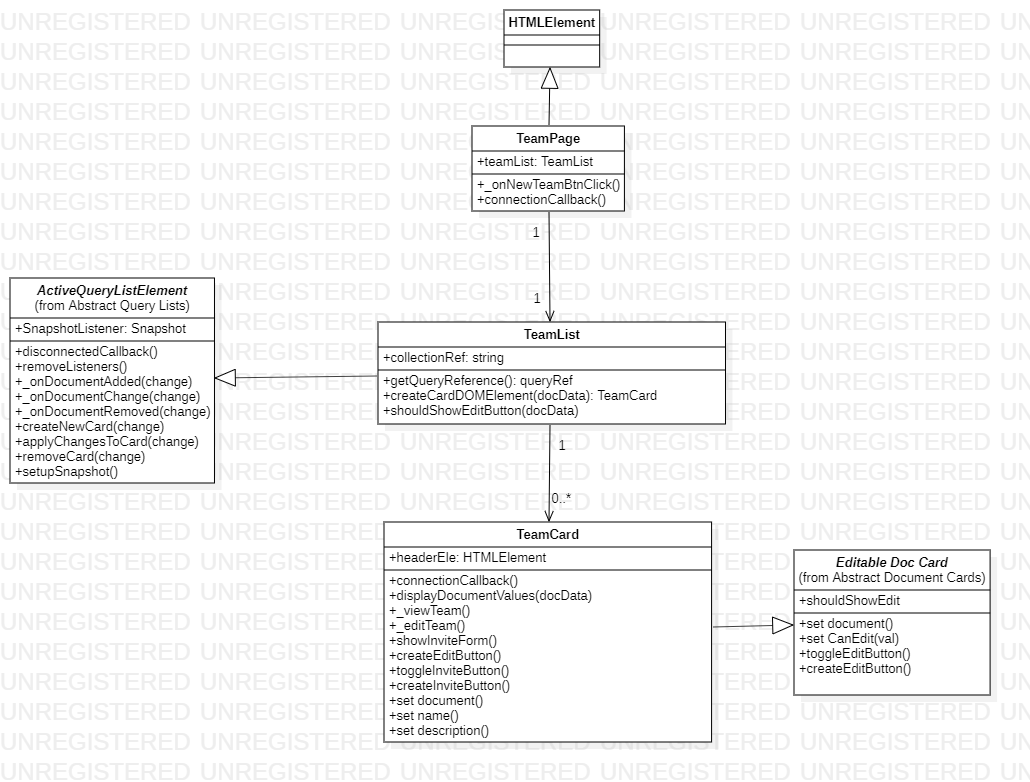


Figure 13 Team page class diagram

##### 2.2b.i TeamPage

The team page is the parent element for this system. It contains a TeamPageList to display up-to-date team information and creates interface control to create new teams. This page is added to the DOM after the user has logged in.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [teamList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\8e11709612d48a36f9eea51b0d6d20e8.html) | [TeamList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\ef58f0fc4bc925944f6e23ef4bc38178.html) | DOM object of the team list element |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [\_onNewTeamBtnClick()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\664fac298497d01fb350a4b20c6c2b6d.html) | On click trigger display of new document form and type of document that the form is attempting to create |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\53a2cb8086ac16c0c32394ec7d4c17e5.html) | When appended to the DOM create a new TeamList and button, apply click listeners to button. |

In order for a user to be able to make new teams this page has UI control to interface with the NewDocumentForm (2.2f.iv). This operation (Figure 11) is executes by clicking on the new team button and controls what document is being made and where to make it in the following way:



Figure 14 onNewTeamBtnClick button operation trigger

##### 2.2b.ii TeamList

The team list extends functionality defined by parent in order to maintaining a list of team cards. This element triggers a query when connected to the page, utilising the authentication object to form a query for team data. A user’s member ship of a team controls their access to data.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [collectionRef](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\0f3cac92f7e5de3a587a3c99a5b8e1ee.html) | string | String representation of the collection location being viewed |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [getQueryReference(): queryRef](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\842466bfdfde7328242c681896e69758.html) | Create a query reference of the teams a user has access to. |
| public | [createCardDOMElement(docData): TeamCard](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\192a1de6d4176f6f9741721fd5ed3483.html) | Create a new document card to represent a database object. |
| public | [shouldShowEditButton(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\1b2610a986f71f9e522b249382a2b77b.html) | Calculates whether an edit button should be shown on a card or not. |

ShouldShowEditButton (Figure 10) is executed on this element by child cards to calculate if the currently logged in user should have the ability to edit a team. Control of user’s ability to do this is done through allow-edit-team property on the team document. An owner of a team is always able to edit them.



Figure 15 shouldShowEditButton operation

The list of cards is populated through extension of the getQueryReference (Figure 13) method. This override performs a query against teams where the team members array contains the currently logged in user. A query against the whole team collection would fail due to enforcement of database rules.



Figure 16 Get Query Reference Override

##### 2.2b.iii TeamCard

The Team card is a fully defined version of an EditableDocCard. This class inheriets functionality to update properties when the document that this card is representing is changed. The displayDocumentValues operation is overridden to control what specific values are being displayed and control the display of both edit and invite buttons based on team properties.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [headerEle](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\c77e1a9eeb3b99bfca8897de2754b73d.html) | HTMLElement | Element to display the name of a team |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\8b2cc0ce051ec7c700036a5845a65074.html) | form basic HTML and apply click listeners to view, edit and invite buttons |
| public | [displayDocumentValues(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\6484bbb104f3a0238de41361034abb63.html) | Concrete implementation of abstract method, updates property on card that are relevant to team card display. |
| public | [\_viewTeam()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\565be29bc8488d24ffdb7deca6359f69.html) | Triggered by on click, call global method to navigate to team page loading the selected team. |
| public | [\_editTeam()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\91267e32437572a1fca8db5b1deda100.html) | Trigger display of change document form with the selected team card set as the target. |
| public | [showInviteForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\eafbd9e1d0703da3dc30780bc5952694.html) | control visibility of invite form, set the target team as the one selected. |
| public | [createEditButton()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\ba434fa2879714c4470218b97d7fecb0.html) | Concrete implementation of abstract method, knows where in the DOM to create an edit button for this card. |
| public | [toggleInviteButton()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\36bd47066dee7cd11bd203895fb81352.html) | Add or remove the invite button from the DOM depending on team properties. |
| public | [createInviteButton()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\78aefe11ab039651cc92e5d9e3975950.html) | Add the invite button to the DOM |
| public | [set document()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\dfadea412cc9b4145790912f60ea1e81.html) | Override the default method to also work out whether this card should be showing an invite button |
| public | [set name()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\38c8acb1029ea6afdde6f71253e7314f.html) | Display the name of the team document and set the property reflect to attribute |
| public | [set description()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\2ebfc518214be74ea7910c3520e92066.html) | Display the description against the team document card and reflect to attribute. |

Control over the display of properties is done through overriding the displayDocumetValues (Figure 14). This allows this class to define which properties are needed to display information. Display and attribute reflection is done through the setter methods of the individual properties.



Figure 17 displayDocumentValues override

Due to this cards relationship with EditableDocCard(2.2b.ii) setting the document value triggers control over the edit display. This utilises the teamList’s ShouldShowEditButton method (Figure 10) to work out if the edit button should be shown. This class adds definition for how an edit button is added to the DOM (Figure 15).



Figure 18 adding edit button to the DOM

Additionality this element has control over interface control to invite users. Invite button is added or removed from the DOM using toggleInviteButton (Figure 16). Control of user’s ability to do this is done through allow-others-invite property on the team document. An owner of a team is always able to perform edit and invite operations on it.

**

Figure 19 toggle invite button

#### 2.2c Tasks Page

Tasks page contains an example of uses of fully derived document cards and query lists. The parent element, tasks page, creates tasks list inside its inner html. This utilises templates that are pre-rendered before being included inside this list. The tasks list utilises functionality defined in changeable active query list and generalisations to keep information displayed up to date.

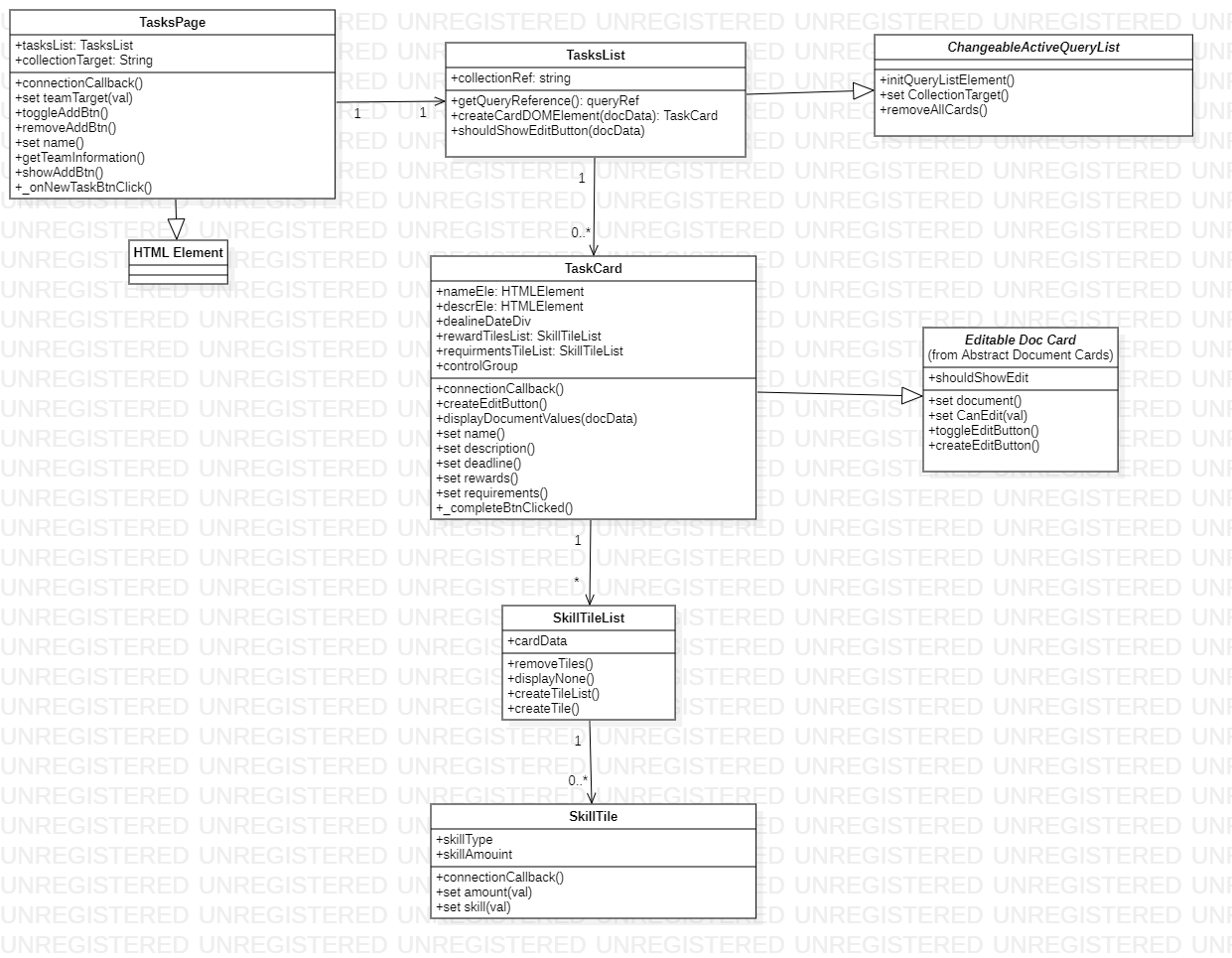


Figure 20 team page class diagram

##### 2.2c.i TasksPage

This list of tasks being display can be modified by changing the collectionTarget property. Doing so causes the tasks lists target collection property to change triggering functionality defined in the ChangeableActiveQueryList to modify which collection this list is viewing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [tasksList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\b74651e60242528354971c3e7ff0540f.html) | [TasksList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\f2ec34101463d7cb1bd618b3879ad8ac.html) | TasksList displaying currently selected teams tasks. |
| public | [collectionTarget](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\837ded7fad5e1e7f52a1bb63445f15ac.html) | String | String reference of collection path. This is the team the list is currently displaying. |

|  |  |  |
| --- | --- | --- |
| **Operations** | |  |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\ac8e4284663aa20a41cc5b4896135691.html) | Form the HTML for this page. Create a new tasks list with the currently viewed team. |
| public | [set teamTarget(val)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\d20ce1347611f93036166813beccc5c6.html) | Allow the page to respond to changes to the team being viewed. This allows the collection being displayed to be modified. |
| public | [toggleAddBtn()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\8c4a142f2ba3f8629469bb24e7c4f45e.html) | Add or remove the add task button from the DOM depending on current user privilege. |
| public | [removeAddBtn()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\11e0a48f7be90751bbdb2a0c7c18b0d4.html) | Remove the add button from the DOM if it is present |
| public | [set name()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\306774343f44771456fc62cb7331d5c5.html) | Update any changes to the team being viewed |
| public | [getTeamInformation()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\19db154a30e1f13522b37933a19c66c8.html) | Query the team document to display name and configure add task button. |
| public | [showAddBtn()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\7488aa278d839bb058d146bbab5c6f8e.html) | Add the add button to the DOM if it isn’t present |
| public | [\_onNewTaskBtnClick()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\362855a9a469aa88b177961a41dcf953.html) | UI control to trigger new document form display. Applies collection-target to query to make the new task create underneath the right team. |

Setting the team target property (Figure 18) is how to modify the target collection or the task list. This property change is reflected down to the task list to trigger a rebuild of list data. This method utilises a promise chain to make sure the asynchronous queries trigger consecutively.



Figure 21 set team target property

As part of changing the team being targeted this element is responsible for setting a global value for the current team being viewed and populating the team name to display the parent document (Figure 19). This operation must occur before the list creates cards as the list requires access to an up-to-date team document.



Figure 22 getTeamInformation

This class also contains a trigger for creating a new task document. The following (Figure 20) method is called to trigger display and set properties of the new document form (2.2f.iv).



Figure 23 new task button call-back

The following method (Figure 21) is used to work out whether a child card should show an edit button. Only owners of tasks and owners of the teams they belong to are able to edit tasks. Task owners can only edit tasks as long as the allow add task property is set on the team document. Changes to team properties will not modify edit buttons of currently rendered task cards.



Figure 24 work out whether a card should show the edit button

##### 2.2c.ii TasksList

The only fully extended form of the ChangeableActiveQueryList. This class builds on functionality by setting concrete implementations of the functions createCardDOMElement and shouldShowEditButton. This list is used to display task information of the currently viewed team.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [collectionRef](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\5910ae0dea296c5d5e6e09a4cf87bd68.html) | string | A string representation of the collection |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [getQueryReference(): queryRef](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\842466bfdfde7328242c681896e69758.html) | Create a query reference of the teams a user has access to. This function is triggered every time the collection target is set, rebuilding the list of documents. |
| public | [createCardDOMElement(docData): TeamCard](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\192a1de6d4176f6f9741721fd5ed3483.html) | Create new team card to represent the query result. |
| public | [shouldShowEditButton(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\1b2610a986f71f9e522b249382a2b77b.html) | Calculates whether an edit button should be shown on a card or not. |

Due to extension of ChangeableActiveQueryList (2.2a.iii) update of the collection target property triggers a rebuild of the list displayed. The following method is called every time this happens to build a list of tasks that are active and belong to the currently viewed team.



Figure 25 get query reference override

This list will contain editable cards. As a result, a definition is needed to work out if they should show their edit buttons (Figure 23). Edit buttons will only show if the person viewing the list is either the owner of the team or an owner of a task where allow add tasks is set on the parent team document.



Figure 26 Should task edit button show

##### 2.2c.iii TaskCard

Task card is used to display task data. A list of these elements are maintained by a the TasksList. Update of values is done through setting the document triggering the card to display information from the document. In cases that this occurs it caused by modifications to task documents belonging to the currently selected team.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [nameEle](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\3ee5a53773ae185763bcfd610a126bbf.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\ed20779d2529b66a3d3b167e15b70b09.html) | Element displaying task name |
| public | [descrEle](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\96c1ff04d1056d1616e9f0614c772876.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\04f945faf38fd53063907ccc53795150.html) | Element task description |
| public | [dealineDateDiv](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\61efe53a5d78da5aee3dc08ebf544112.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\04f945faf38fd53063907ccc53795150.html) | Element displaying the task deadline |
| public | [rewardTilesList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\a775a0177dfa32c6a76c717d87a3a053.html) | [SkillTileList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\608f555dfa3e22159496ad9f508b5c9b.html) | SkillTileList to display the rewards |
| public | [requirementsTileList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\5ca036b636caedf26f0a02caa1e6425f.html) | [SkillTileList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\608f555dfa3e22159496ad9f508b5c9b.html) | SkillTileList to display the requirements |
| public | [controlGroup](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\d829c53a94b3064bcf419b3ba7d6b711.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\04f945faf38fd53063907ccc53795150.html) | Group of controls that can be performed on the object this card represents. |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\7f5af99277070a227b793b09e5cf44e8.html) | Form basic HTML including a new edit button. Form references to elements that will be modified. |
| public | [createEditButton()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\cf637350ab86ed75b72283b1e9672aad.html) | Add a button to trigger the edit process |
| public | [displayDocumentValues(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\e5cc58cbb5735d340ff461bf897761a1.html) | Concrete implementation of abstract method, update any values a task card displays. |
| public | [createRewardTiles(reward)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\f18532f7b8e68bd033cb6c3cde93b4f1.html) | Create a set of tiles to represent the rewards available on a task to the user |
| public | [createRewardTile(type, amount)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\197bc67fd026b967244f3bbe5ddba529.html) | create a new reward tile, append it to the DOM and add it to the map reference on the card. |
| public | [set name()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\a9bc13134c9d51bf0c56ecc9fac22d52.html) | Display any changes to the name. Reflect as an attribute. |
| public | [set description()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\a9b06d2b6f2473d8b7438a38f757cf20.html) | Display any changes to the description. Reflect as an attribute. |
| public | [set deadline()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\5a874b317257cd4b58efea38c5d37a9d.html) | Display any changes to the task deadline. Sanitise data and reflect as an attribute. |
| public | [set rewards()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\9fde89a50e44a52f76fe15f0689d0780.html) | Display any changes to the rewards of a task via updating a skill card list element. Don't reflect this property as it requires parsing. |
| public | [set requirements()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\e00ac4e156bb1b0d5e79a66899ee9f63.html) | Display any changes to the requirements of a task via updating a skill card list element. Don't reflect this property as it requires parsing. |
| public | [\_completeBtnClicked()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\5ddb77a37145cde001835ea631e88879.html) | Call-back appended to complete task button. Triggers task completion process. |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\7f5af99277070a227b793b09e5cf44e8.html) | Form basic HTML including a new edit button. Form references to elements that will be modified. |
| public | [createEditButton()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\cf637350ab86ed75b72283b1e9672aad.html) | Add a button to trigger the edit process |

In order to display information on the card the displayDocumentValues (Figure 24) override sets relevant properties. In most cases they are directly on this element, rewards and requirements are then set on their respective tile lists.



Figure 27 displayDocumentValues

##### 2.2c.iv SkillTileList

List of skill tiles. Unlike implementation provided by ActiveQueryLists updates to values being displayed by this element trigger the element to clear its child elements before rebuilding tiles. The following tables define the operations and attributes present on the SkillTileList.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [cardData](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\c655771df24398c24195e85b3ef5927b.html) | map | Skill value map that this element is displaying. |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [removeTiles()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\289e6faabb8a519a6baffe10e6def992.html) | remove any elements within the list |
| public | [displayNone()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\9ea28b3eb4d2a6dc80fc6e89df5ea460.html) | create a display to show that this list is empty |
| public | [createTileList()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\d3b7b35d465c90152ecc9ede83713efe.html) | remove any existing tiles and create a new list |
| public | [createTile()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\74505ac5e6609ec693581caa9b464625.html) | Create a new skill tile |

##### 2.2c.v SkillTile

Single unit of a SkillTileList this element is used to display a single skill – amount mapping to show an image and a number. This class utilises the global method LookupImageURI to find the source of image that the skill represents.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [skillType](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\9f400f50071c581cae41ac8f2b1d0f66.html) | none | The type of skill the card is displaying |
| public | [skillAmouint](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\c7ffc7d12774ab02d5c1a332fc94a91a.html) | none | The value being displayed by this tile. |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\ee3d991715413f601ef5f5d2c57a07a9.html) | Construct HTML and create reference to image and value display. |
| public | [set amount(val)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\4ed28f61d53c6d92a3b7dc01d27fe2a2.html) | Controls what value is being displayed and mirrors properties to attributes |
| public | [set skill(val)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\09882f3462640dd9c5edc6158c91802b.html) | Controls what image is being displayed by the tile and mirrors the property to attributes. |

#### 2.2d Notification Page



Figure 28 notification page class diagram

The notification page is used to display notifications to the user. This system is used to notify of team members completing tasks, receive and accept team invites. The class diagram (Figure 11) describes the relationship between class to add this functionality. Both card and list elements utilise templates to populate HTML contents.

##### 2.2d.i NotificationPage

The simplest of the three-page custom elements. This class contains a list of notifications which is formed when the page is connected to the DOM. This class is used as the wrapper for any notification content. The following tables document the attributes and operations found on this class.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | |  |  |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [notificationList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\18ae0bd3976e9ef426548e97217a9c59.html) | [NotificationList](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\57eb2ef85ad5f729f56ddc0c1420f34b.html) | The DOM object displaying notifications |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\ad07be18d4e89c9f273a4ec1b9b576ff.html) | Create Notification list when the element is connected to the page |

##### 2.2d.ii NotificationList

Like all ActiveQueryLists this list is populated on connection with the page. This produces a list of notifications that the user has access too. Changes to any notifications trigger updates to any of the documents represented in the list.

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [getQueryReference(): queryRef](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\e2af6dffae7950056df3c7710d04dbb3.html) | Returns a reference of all notifications that are for the current user. |
| public | [createCardDOMElement(docData): NotificationCard](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\b62a56b950156697e2d6a5198b2be33c.html) | created DOM element to represent Notification data. This element depends on the notification type in the document the element it is representing. |

##### 2.2d.iii NotificationCard

Notification card is the concrete implementation of a DocumentCard. This is the only example in the application that inherits directly from this instead of the more derived EditableDocCard. The display of notifications is controlled by the isRead property hiding notifications that have been marked as read. Due to the nature of firebase queries a compound query producing notifications for a user that are marked as unread is not possible.

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\8dfcea0f5dacbd1ee2c44aa78254641f.html) | Forms basic HTML and created references to DOM elements needed to display data. |
| public | [displayDocumentValues(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\b48141266ed3a671d699f52c82d905e4.html) | Concrete implementation of abstract method. Update display of data relevant to this elements display. |
| public | [\_clickHandler()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\f0cc4593b88080283524ce333aa73342.html) | General click handler to allow generic notifications to be marked as read |
| public | [markAsRead()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\a4cd95c8417d0d7a858a2624b5d8c88a.html) | applies document changes to notification object to display as read. |
| public | [set Message(val)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\8065ab537e0b1e4ebefc7d0565239396.html) | Sets text being displayed to the value stored in the massage of the document. |
| public | [set IsRead(val)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Release%201\UML\Workaholic\html-docs\contents\e92a4c46214b39779b0c771477df2087.html) | Causes a class list update on the element to prevent it being rendered. |

##### 2.2d.vi InviteNotificationCard

This class extends NotificationCard functionality by adding buttons for the user to interact with. This is the way interface controls are added to allow a user to accept invites becoming a member of a team. A global method acceptInvite is used to commit changes to the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [teamDocLocation](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\fb65500fd5071f4d91eccc37d0a2eacb.html) | string |  |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\1f640a19c490c03ddf4d34a30a9e2d83.html) | Creates HTML and element variables that will be modified by connection callbacks. |
| public | [displayDocumentValues(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\a0ccfe1bf2f27576400cfe2c7c74181f.html) | Concrete implementation of abstract method. Adds the setting of team location to this object. |
| public | [set teamDocumentLocation(docData)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\f6b3a3bbac847de69671c6de80e5d5c2.html) | Set internal property and reflect to attribute |
| public | [\_clickAccept()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\a958096fe5f28ac10e8fac60cc082133.html) | Triggers notification accept process. This adds the current user to the members collection presant on the team document. |
| public | [\_clickDecline()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\34a064bb3ba0758003bfda0778d55576.html) | Mark the notification as read and trigger removal of current user from the pending-invites collection, revoking modification permissions. |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\1f640a19c490c03ddf4d34a30a9e2d83.html) | Creates HTML and element variables that will be modified by connection call-backs. |

#### 2.2e User Page



Figure 29 user page class diagram

The user page is responsible for displaying and updating a user’s skill levels and contains the users QRcode for inviting to a team. The diagram (Figure 12) describes the relationship between a UserPage and Experience bars that enable this functionality. The initial page is created using templates, but the experience bars themselves utilise modification of their inner HTML directly.

##### 2.2e.i UserPage

This user page is used as the parent container for the experience bars. It is also responsible for: triggering popup notifications when acquiring new skills, displaying a user’s QR code and creating new account objects.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [experienceBars](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\4e78ce79df90e2bc38d5501ae358a470.html) | map | Map of element to skill type. Collection o elements displaying experience values. |
| public | [isLoading](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\70fbc84c4aba67c37783560fb545cf66.html) | boolean | Has this element finished its initial query. Prevents new skills from triggering fanfare when adding them from the initial query. |
| public | [qrCodeWrapper](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\f4dcb06e8bed9ddd8b5f1562e3d23b17.html) | HTMLElement | wrapper containing the QR code and the text displaying the user code. |
| public | [userAccount](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\3470e694ecd414d567915a3ff1c70aea.html) | Document reference | Document reference of the user that is queried. |
| public | [snapshotListener](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\bc95681f2eda306e155697e98c1680f9.html) | snapshotListener | Listener responding to changes to the user document |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\5c425794b82731d47ff649c982b93563.html) | Build HTML, select elements and append listeners. |
| public | [setUserLogin()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\2d75c1cf2de422058c582fe91fd01a48.html) | Set the last logged in time on a user document. Will create a new user if there currently isn’t one for the person logged in. |
| public | [createQRCode()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\b56a38e7718866794c3a4c9f8c27b9f0.html) | Utilise qrcodejs to build a QR code to display on the interface. |
| public | [updateExperienceBars(skillLevels)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\004ae5297d4bb2b4532115e6be295944.html) | update experience bars with new values. Create new bars for new skills earned. |
| public | [\_ToggleInviteCode()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\b63904c6ecf4917dbf32c631ae288446.html) | display/hide the users invite code. |
| public | [disconnectedCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\5ba99ad2b3065fdaeca638c56fa91f2e.html) | remove any listeners still on the object |

##### 2.2e.ii ExperienceBar

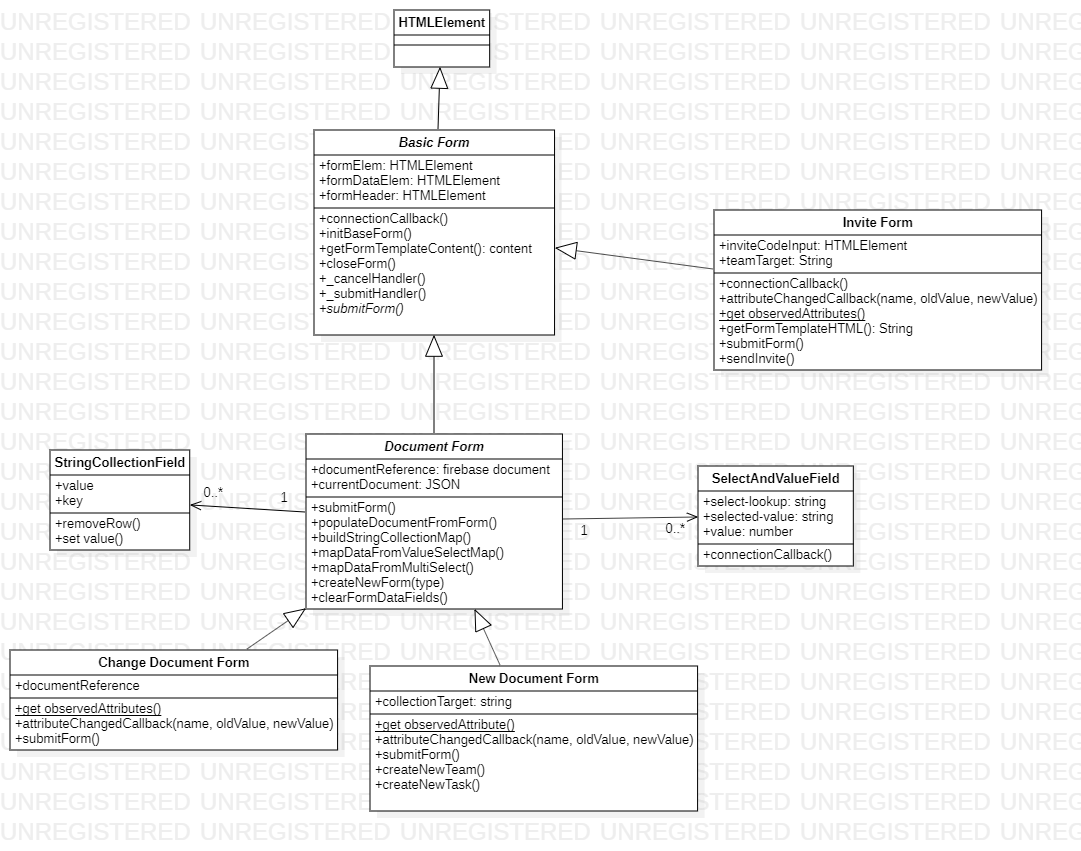
This class is responsible for displaying the skills and levels a user has. It is responsible for displaying how close a user is to a new level and triggering popup notifications of a skill level-up.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [progressBar](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\a0af318cb9afa48c2006f44c2433aa7e.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\6d961743dc5ff567586b50bbc1a4a10f.html) | Element used to reset progress into a level as a progress bar. |
| public | [skillLevelText](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\d9ae2aad03a8e1877bcd7cf41fe0e76c.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\6d961743dc5ff567586b50bbc1a4a10f.html) | Element displaying the current level as a number. |
| public | [experience](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\8af24ba1105cee8065456af6bd88302c.html) | number | The experience points value that this element is displaying. |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [checkForLevelUp(oldExperience, newExperience)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\ad2bea0f50aee9befe7eecc8dc66e715.html) | Check if changes to experience points result in a level up |
| public | [triggerFanFair()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\842cf1c6a02b565cc09f3d52f6862fd4.html) | Trigger popup notification to indicate a new level has been gained. |
| public | [setBarWidth()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\b4ff61eb21853076856a45b656e87cde.html) | Set the bar width to represent the progress into the current level. |
| public | [setTitleText()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\eb4cf4c711caece47c32353c13d8e961.html) | Set the level being shown to the user. |
| public | [set currentExperience(value)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\3f073a9136f8d1d6ff838abc4cab4a48.html) | Value being displayed in the amount of experience a user has in a particular skill. Reflect value to HTML attributes. |
| public | [set skillType(value)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\cc100ac7315906a2dc5bb7cc6a50e4eb.html) | Skill type that this element is representing. |

#### 2.2f Document Forms

Document forms are the main way for a user to interact with data allowing them to add and modify documents. These elements are created as autonomous custom elements. The classes Invite Form, Change Document Form and New Document Form are registered to the custom element registry and connected directly to the body element of the page. Display of these is controlled externally.



##### 2.2f.i Basic Form

This class is designed as an abstract class with no concrete definition for actions when the form is submitted. This class lays out fundamental functionality that is required by a form. Three different forms are added defined in the HTML triggering the base content of the forms to be populated on load. Basic forms have the base implementation of submit form to simply hide it.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [formElem](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\82fc38a28e09813c932eba5051df629e.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\04f945faf38fd53063907ccc53795150.html) | Element that contains the whole form |
| public | [formDataElem](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\f8d3532c5acaac817049ead0e36a68d9.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\04f945faf38fd53063907ccc53795150.html) | The form contents that will change depending on the type of object represented. |
| public | [formHeader](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\41febbf0c8c997936c725c8521270705.html) | [HTMLElement](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\04f945faf38fd53063907ccc53795150.html) | HTML element to display a header fort the form dialogue |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\3e744afc88c697572a3bb72dbcd6395a.html) | trigger form initialisation |
| public | [initBaseForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\07b7b64e00d330b7fb00a49d412dbe8d.html) | Trigger HTML generation and assign variables from the DOM relevant to the form. |
| public | [getFormTemplateContent(): content](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\a09640e8b5964f9058882f096da2df49.html) | Form a template content needed by the initial form. Allow implementation to define their own HTML, allowing for button text and contents to be different. |
| public | [closeForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\ab02f2ba0223ff007eb456072bd1b8cd.html) | Trigger form close by setting the hidden attribute |
| public | [\_cancelHandler()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\2f113b17eff65f12bf33c4e3d494ca80.html) | Applied to cancel button, triggers form close |
| public | [\_submitHandler()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\f78d927a4230b294ac44b7431c8afd9f.html) | Call the form submission method to submit changes and trigger queries |
| public | [submitForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\55665fcadb81bd25a1e74da3ce24d2f8.html) | implementations should define their own submission |

##### 2.2f.ii Document Form

This class is responsible for both displaying and modifying document data. Parsing uses JSON definitions of objects to define what fields should be displayed\*\* and define form inputs including validation properties. Fields not defined on this JSON can exist on a document but won’t be shown to the form interface and won’t be updated innately when submitting. All input element creation is done through utilising global form creation methods.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [documentReference](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\c7ef96a617cb4b17a76ed2943880896d.html) | firebase document ref | Document reference of the object that is being modified. |
| public | [currentDocument](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\265abd7caec5d36794d194011208d2e3.html) | JSON | Reference to JSON object representing the currently loaded object. This is built and modified to form database queries. |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [submitForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\ecba91c3b5bd233f974e8c48f8036d11.html) | Own Implementation of submission method. Uses data entered in the form to modify document data. |
| public | [populateDocumentFromForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\5c19c09700727fffe264cd0b00da028c.html) | Populates the new/changed object with any values entered in the form. Understands how to represent different input types as database objects. |
| public | [buildStringCollectionMap()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\7b20bcc06fd47bd17cf3fbf97394d776.html) | Convert a list of input fields into a collection map. This data is being used to make selection boxes so requires a key value mapping. |
| public | [mapDataFromValueSelectMap()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\f8154008f527a927bc3abf815eab7b37.html) | creates a key:value map from a SelectAndValueField any repeated keys will override previous ones. |
| public | [mapDataFromMultiSelect()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\02b256385556d919821a25bb89cb6217.html) | Map data from a multiselect field. Allows the database to represent select data in a more minimal way. |
| public | [createNewForm(type)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\3054b63d62d325ecfa08ce59921656b5.html) | Triggers creation of different form types. Forms a document from default values of the type and triggers form creation, populates form field values with type defaults. |
| public | [createFormFromExisting(type, document)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\09bd4904f016521bb7d8d79251c1ea83.html) | triggers form creation of specific type uses the document parsed in to populate the form field values. |
| public | [clearFormDataFields()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\17ae820be300774b95b3a15423a7fc99.html) | Clear any of the fields from the form data field set. |

In order to populate document objects the populateDocumentFromForm is used to populate fields (Figure 13) the contents of the switch statement have been omitted for brevity. This method targets specific element types ignoring any not defined by the switch statement.



Figure 30 form field to document parsing

For custom elements specific methods are called within the switch statement to parse the data. For example, the SelectAndValue field mapDataFromValueSelectMap is used (Figure 14). Changes made by a user to these fields are reflected to element attributes that this method reads.



Figure 31 data mapping of select and value field

##### 2.2f.iii Change Document Form

Document Form Type exclusively dedicated to creating new documents. This class utilises functionality defined in its parent class in order to parse and create input fields. Element attributes ‘document-target’ and ‘obj-type’ are used to define where to find the document and how to parse the data returned.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [documentReference](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\28bbeefafe89c1b841f7fbef01883874.html) | Firebase document reference | Document reference of the object that is being modified. |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [get observedAttributes()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\5cad81cd34a57221855bfd2b7bc1987d.html) | allows form to respond to changes to 'document-target' and 'obj-type' |
| public | [attributeChangedCallback(name, oldValue, newValue)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\121e77f80cfe8f32f4c984fd47a2c4a4.html) | modifying document-target triggers a get on the document and creates the form from the it. Requires obj-type to be set to control how the form interprets values found on the document. |
| public | [submitForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\2769ff32685e2158c2d3d1dc6b8ebb78.html) | apply any changes to data and close the form |

Submission of forms triggers parsing of the form data and a commit to the database of the new object (Figure 15). This method uses functionality defined by populateDocumentForm of the parent document form to parse form data into a JSON understood by firebase queries.



Figure 32 Change form submission

##### 2.2f.iv New Document Form

Create document form is used exclusively to add new documents to collections. The parsing and creation of input form is functionality is defined by its parent class. This form additionally understands how to commit new objects to the database. Additional creation logic can be defined with methods used to create specific documents.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [collectionTarget](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\1919ec6163e0d4620960e16b8eb9ffd4.html) | string | Location the new object should be created |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [get observedAttribute()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\0e450256654173cd330a497a7d91dc42.html) | Allows form to respond to changes to 'collection-target' and 'obj-type'. Changing this attribute triggers the form to rebuild. |
| public | [attributeChangedCallback(name, oldValue, newValue)](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\ee7e95e505a8d5f6c8ffdad36f5437dc.html) | Triggers changes to form display when changing obj-type. Assigned collection-target to the collection Target reference. |
| public | [submitForm()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\0dabca2afd4a40aedb5ca7ed8ed75f24.html) | Create the document in the way required by the type. |
| public | [createNewTeam()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\289e700dd1134bfcf4cfd727ac600d61.html) | Create a new team on the database set values from the form on this new team. Some values are set here that aren’t defined by the form. |
| public | [createNewTask()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\771edf4ea56d68adc89a2c7b9eacaa82.html) | Create a new task under the collection target, set the current user as the owner. Set any values from the form on the new task. Some values are set here that aren’t defined by the form. |

Properties that are not defined by the object type JSON therefore not editable in the interface are added before committing a new object to the database, for example:



Figure 33 create new team method

The owner field on a team document modify how permissions are calculated and the members collection is used to query the database for the teams a user has access to. Allowing these fields to be directly changed by a user has potential to put the data in a state where it is unusable.

##### 2.2f.v SelectAndValueField

This custom element is used to parcel a key value pair that is understood by the document form populateDocumentForm method (2.2f.ii). This type of form input can be seen when entering skill requirements on a task\*\*. This presents UI objects that allow a user to change values in the form of a number input and select field. Changes to both these are reflected to properties that are interpretable by the document form.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [select-lookup](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\51bab4ae3fe4fb94029aafb0f2e6cbd4.html) | string | string reference to the lookup this select field should use to display options |
| public | [selected-value](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\af9ced50f0d68850468d7129aeea005f.html) | string | chosen value from the select field |
| public | [value](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\04ad4fc10d969234af08f5c85064ff13.html) | number | Number that has been entered in the input box |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [connectionCallback()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\6064e837098d8a09e1d4bf631e0e621d.html) | creates a select field and number input wrapped in a. creates button to add and remove vales from the list. |

Listeners are added to the two form elements within to mirror properties up to the parent element (Figure 17). These are understood by the document form parser.



Figure 34 Mirroring input to parent properties

##### 2.2f.vi StringCollectionField

The string collection field is used to parcel information into a div that is interpretable by the document form populateDocumentForm method (2.2f.ii). Changes made to the input field are reflected into a property that is read by the document form. This can be seen in the \*\* to allow teams to add personalised skills.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | | | |
| **Visibility** | **Name** | **Type** | **Description** |
| public | [value](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\9d75979ba9506bd3f9517dd9c4191408.html) | string | the current value entered in the input field |
| public | [key](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\919a3a0d553772b9363bf3d04bce6e7a.html) | string | the key of the value as presented by JSON configuration |

|  |  |  |
| --- | --- | --- |
| **Operations** | | |
| **Visibility** | **Name** | **Description** |
| public | [removeRow()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\d239cd2a117d782b642f7cff7080c78f.html) | remove this field from the parent element |
| public | [set value()](file:///C:\Users\Jordan\Desktop\CI301%20FIXED\CI301\Planning%20Report\Submittables\New%20folder\html-docs\contents\e953446afa26828abd9246e7fbb07372.html) | reflect input value to element attribute |

### 2.3 Sequence Diagrams

This section presents sequence diagrams to aid in discussion of object interaction. Internal database opperations are controlled by Firebase by way of API calls made from the client.

#### Create Task/Team

Creating both tasks and teams is done through operations defined on the NewDocumentForm (2.2f.iv). Display and the type of document being created is defined by the respective pages. This interaction is described by the sequence diagram (Figure 32). This includes interactions with the Task/Team pages, NewDocumentForm and Firebase Database. Access permission checks are performed against a query to make sure it cannot return results a user doesn’t have permission to access. Input attributes are used to validate input on form input.

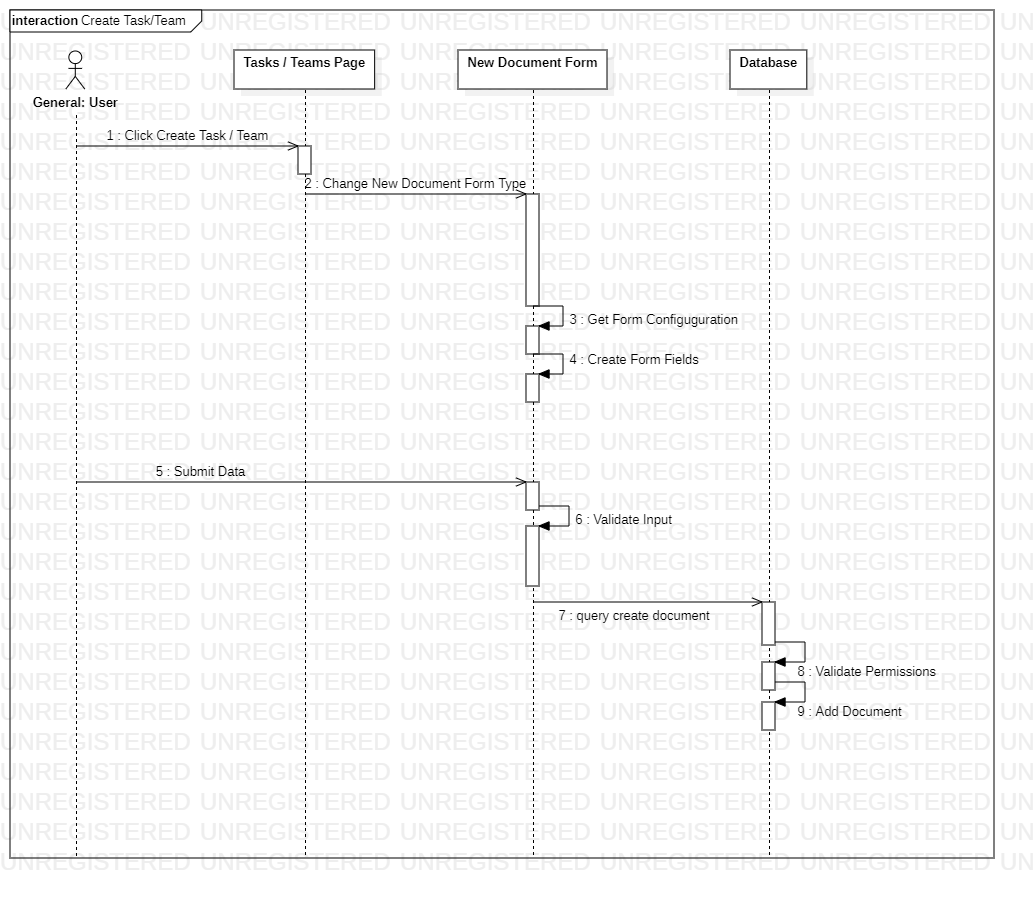


Figure 35 New Task/Team Document

#### Invite User

The invite user process involves two actors, the using inviting and the user being invited. The sequence diagram (Figure 33) documents the process of inviting a user to a team. Completing this process will result in a granting access to a teams data, including tasks. This process is current not reversable in the interface so once a user has been invited to a team they then cant be removed.

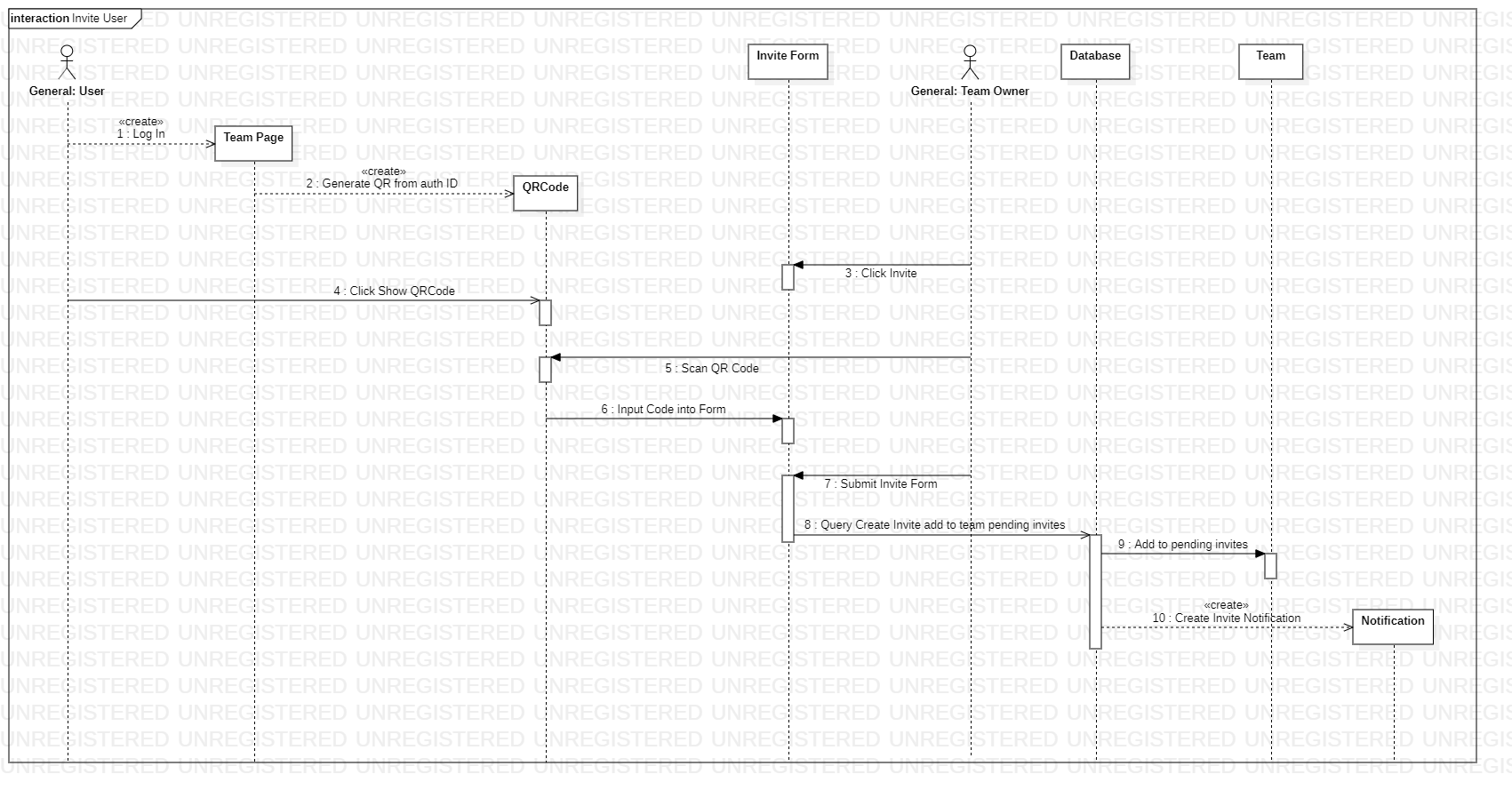


Figure 36 Invite user sequence diagram

#### Complete Task

The following diagram documents (Figure 34) the processes involved in completing tasks. This also includes how a user’s task list is updated as a result of completing a task. Any client with the same list of tasks will receive updates via this process.

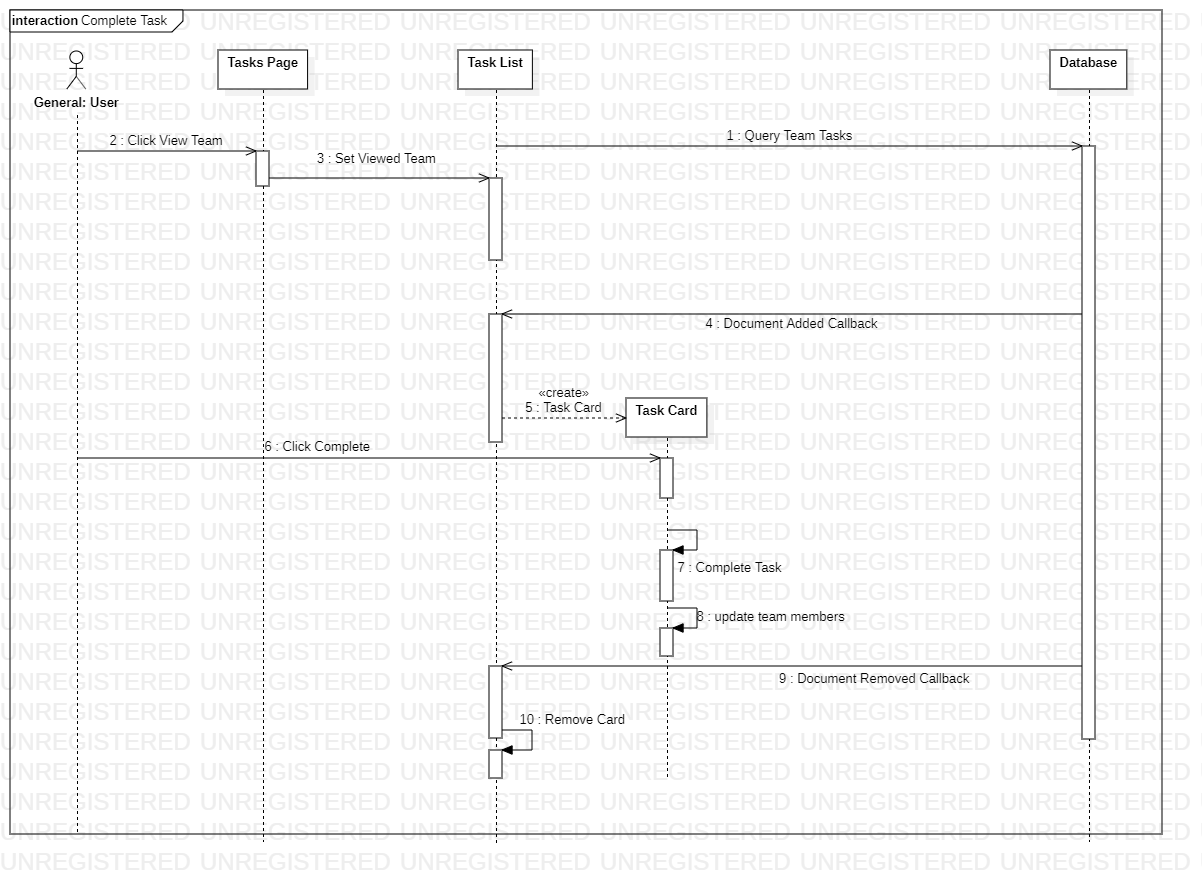


Figure 37 Complete Task

#### Active List Update

The following diagram () documents the process that occurs to update any of the active list elements. All three extensions of this class use this process to maintain an updated list of cards. The creation and callback of snapshot listeners is handled by the firebase API.

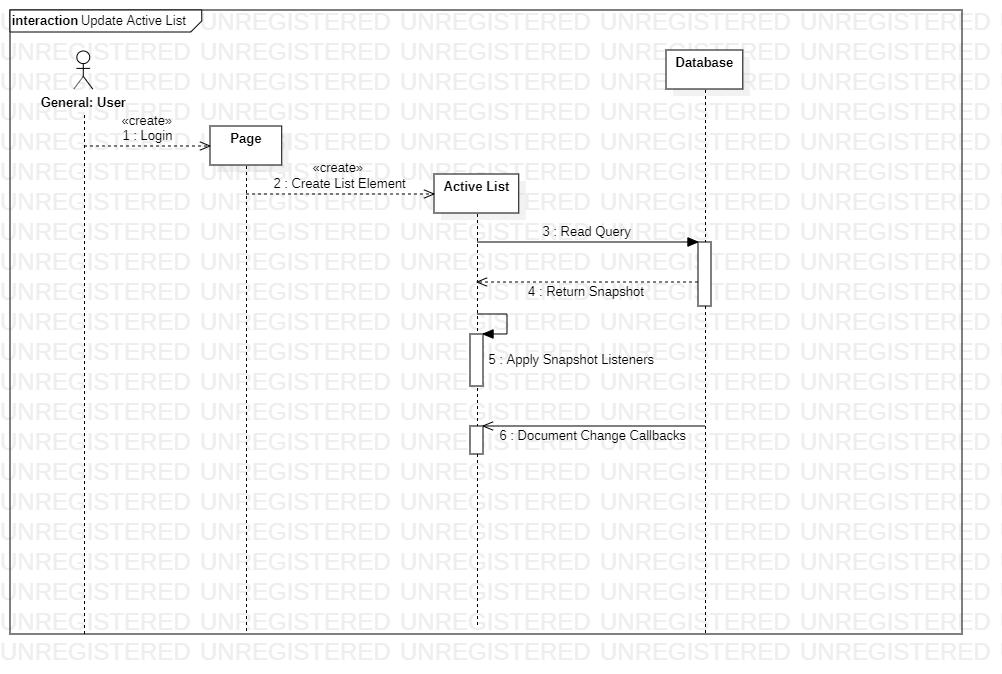


Figure 38 Active list Updates