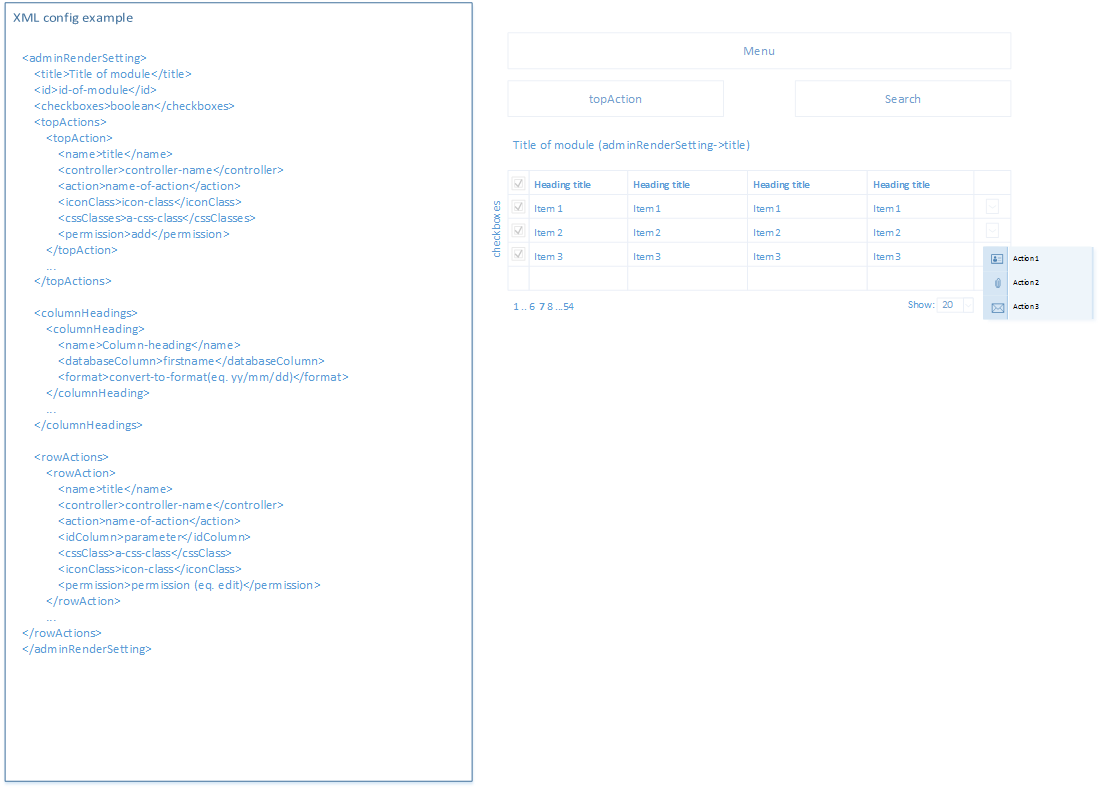
# HR admin portal

## RenderAdminView

The RenderAdminView component generates all admin views based on table layout content. It’s connection between **User role permission**, **Module config** file and required **data** from database. This component will cover two modules in the *Phase 1b*, **Employee module** and **User module**.

### Schema



### Config file

**﻿**<adminRenderSetting>  
 <title>**Title of module**</title>  
 <id>**id-of-module**</id>  
 <checkboxes>**boolean**</checkboxes>  
 <topActions>  
 <topAction>  
 <name>**title**</name>  
 <controller>**controller-name**</controller>  
 <action>**name-of-action**</action>  
 <iconClass>**icon-class**</iconClass>  
 <cssClasses>**a-css-class**</cssClasses>

<idAttr>**aId** </idAttr>  
 <permission>**add**</permission>  
 </topAction>  
 **...** </topActions>  
  
 <columnHeadings>  
 <columnHeading>  
 <name>**Column-heading**</name>  
 <databaseColumn>**firstname**</databaseColumn>  
 <format>**convert-to-format(eq. yy/mm/dd)**</format>  
 </columnHeading>  
 **...** </columnHeadings>  
  
 <rowActions>  
 <rowAction>  
 <name>**title**</name>  
 <controller>**controller-name**</controller>  
 <action>**name-of-action**</action>  
 <idColumn>**parameter**</idColumn>  
 <cssClass>**a-css-class**</cssClass>  
 <iconClass>**icon-class**</iconClass>  
 <permission>**permission (eq. edit)**</permission>  
 </rowAction>  
 **...** </rowActions>  
</adminRenderSetting>

### Top actions

The arw component generates the **Top action** element depending on the **module config** file.

<ul class=**"three\_up tiles top-options"**>  
 **[foreach adminRenderSetting->topActions as ta]  
 [if {user\_permission[id-of-module][ta->permission]}]** <li>  
 <a href=**"{controller}/{action}"** class=**"{a-css-class}"** id=**"{idAttr}"**>  
 <i class=**"{iconClass}"**></i>  
 **{title}** </a>  
 </li>  
 **[endif]  
 [endforeach]**</ul>

### Table Heading

<thead>  
 <tr>  
 **[if {adminRenderSetting->checkboxes}]** <th><input type=**"checkbox"** id=**"toggle\_all"**/></th>  
 **[endif]  
 [foreach {adminRenderSetting->columnHeadings}]** <th>**{title}**</th>  
 **[endforeach]** <th>**&nbsp;**</th>  
 </tr>  
</thead>

### Table

<tbody>  
 **[foreach {db\_records as record}]** <tr>  
 **[if {adminRenderSetting->checkboxes}]** <td>  
 <input type=**"checkbox"** name=**"row\_id[]"** value=**"{record[id]}"**/>  
 </td>  
 **[endif]  
 [foreach {adminRenderSetting->columnHeadings as column}]** <td>

[if {column->format}]

**[format[{column->format}](**

**record[{column->databaseColumn}]**

**)**

**[else]**

**record[{column->databaseColumn}]**

**[endif]**

</td>  
 **[endforeach]** <td>**[rowAction, see below]**</td>  
 </tr>  
 **[endforeach]**</tbody>

### Row actions

Displaying of row actions is dependent on user role permission and on setting of current module.

<tbody>  
 **[foreach {db\_records as record}]** <tr>  
 **..** <td>  
 <div class=**"line-sub-menu relative"**>  
 <a class=**"arrow"**><i class=**"icon-down-dir"**></i></a>  
 <ul>  
 **[foreach {adminRenderSetting->rowActions as ra}]  
 [if {user\_permission[id-of-module][ra->permission]}]** <li>  
 <a   
 data-id=**"{record[ra->idColumn]}"** class=**"{ra->cssClass}"** href=**"{ra->controller}/{ra->action}/{record[ra->idColumn]}"** >  
 <i class=**"{action->iconClass}"**></i>   
 **{action->name}** </a>  
 </li>  
 **[endif]  
 [endforeach]** </ul>  
 </div>  
 </td>  
 </tr>  
 **[endforeach]**</tbody>

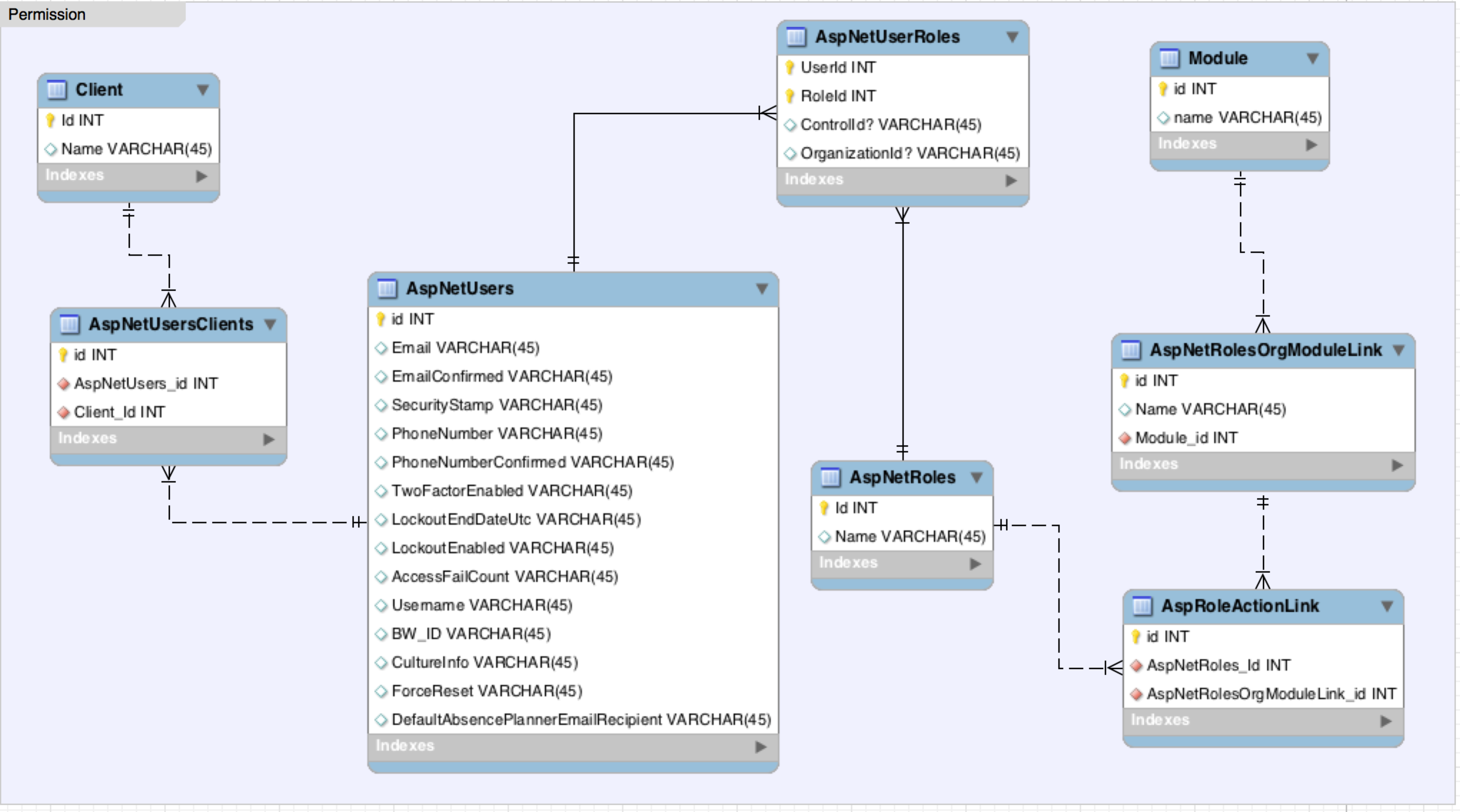
### Permission

Permission setting needs to be passed from the user role, where all permissions are held. Probably the easier way how we can manage user role and permission setting is creating/generating internal module by RAV. You can find more information in the section **User permission**

## User permission

Complete database schema is described below. The one of the most important parts is connection between Client and User. Depending on this setting, the raw will receive data entries from database. For example: if the user has a connection with client1, in the Employee module will see records just for client1. But if he has connection for client1 and client2, he will see records for both of them.

### DB schema



### Options

**Module Employees**

* index (access into a module)
* file\_upload
* send\_email
* open\_lifestyle\_event
* employee\_details
* edit\_employee
* attachments
* attachments\_add
* attachments\_edit
* attachments\_delete
* logs
* logs\_add\_contact
* logs\_delete
* life\_events
* life\_events\_edit
* life\_events\_add
* life\_events\_rollback
* user\_role
* set\_benefit
* activate\_user
* deactivate\_user
* reset\_password
* dependants\_details
* dependants\_details\_edit

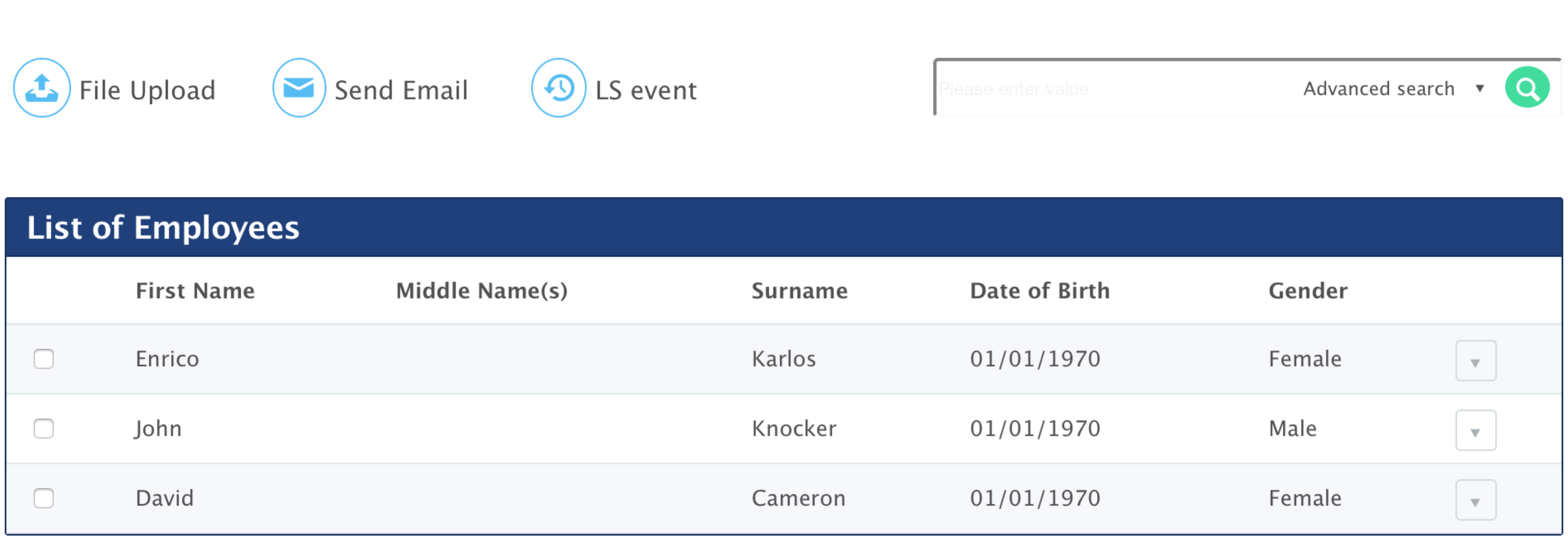
**Module Users**

* index
* new user
* user\_role
* reset\_password
* deputy\_setting
* line\_manager

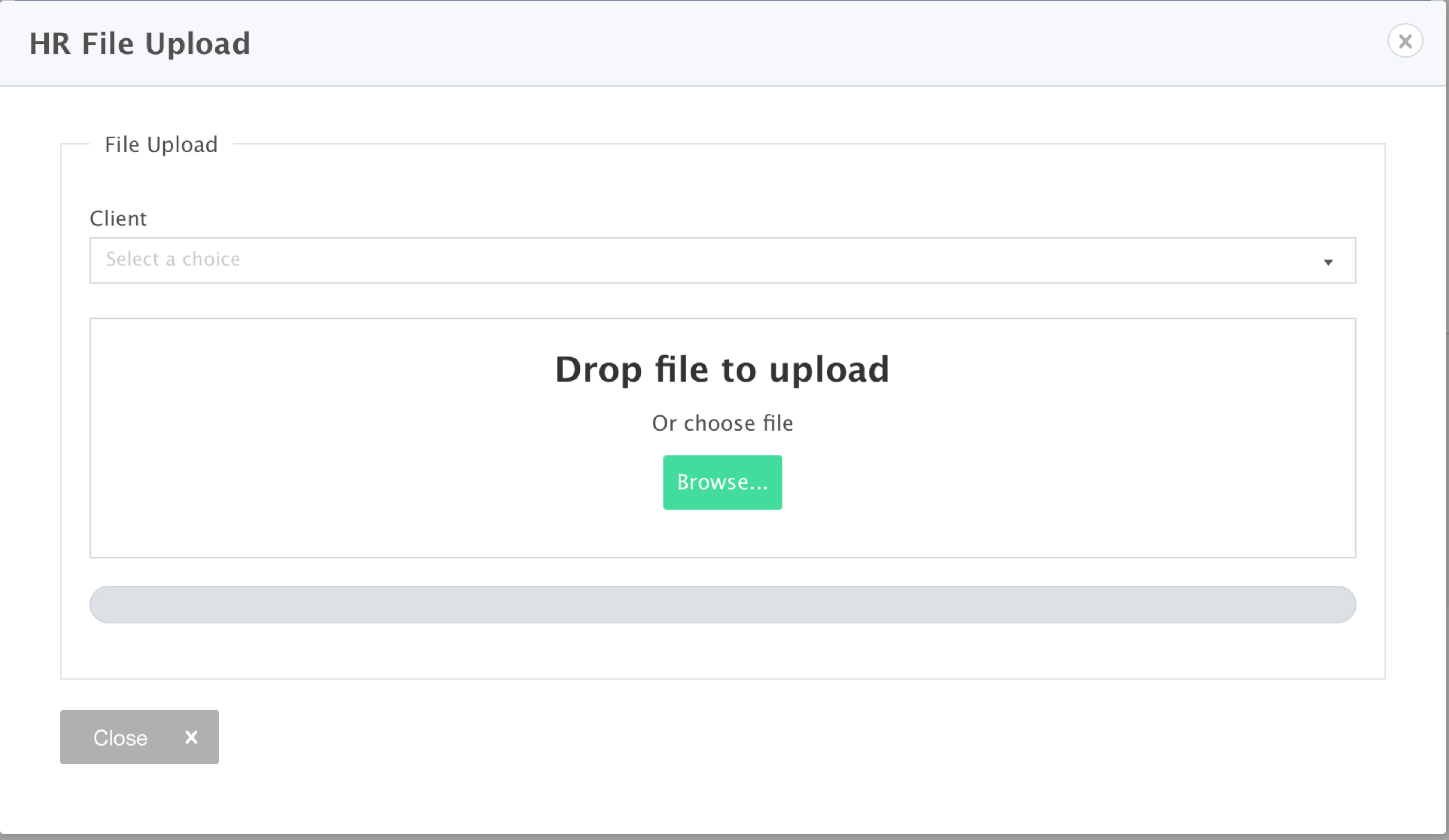
## UX screens

### Employee module

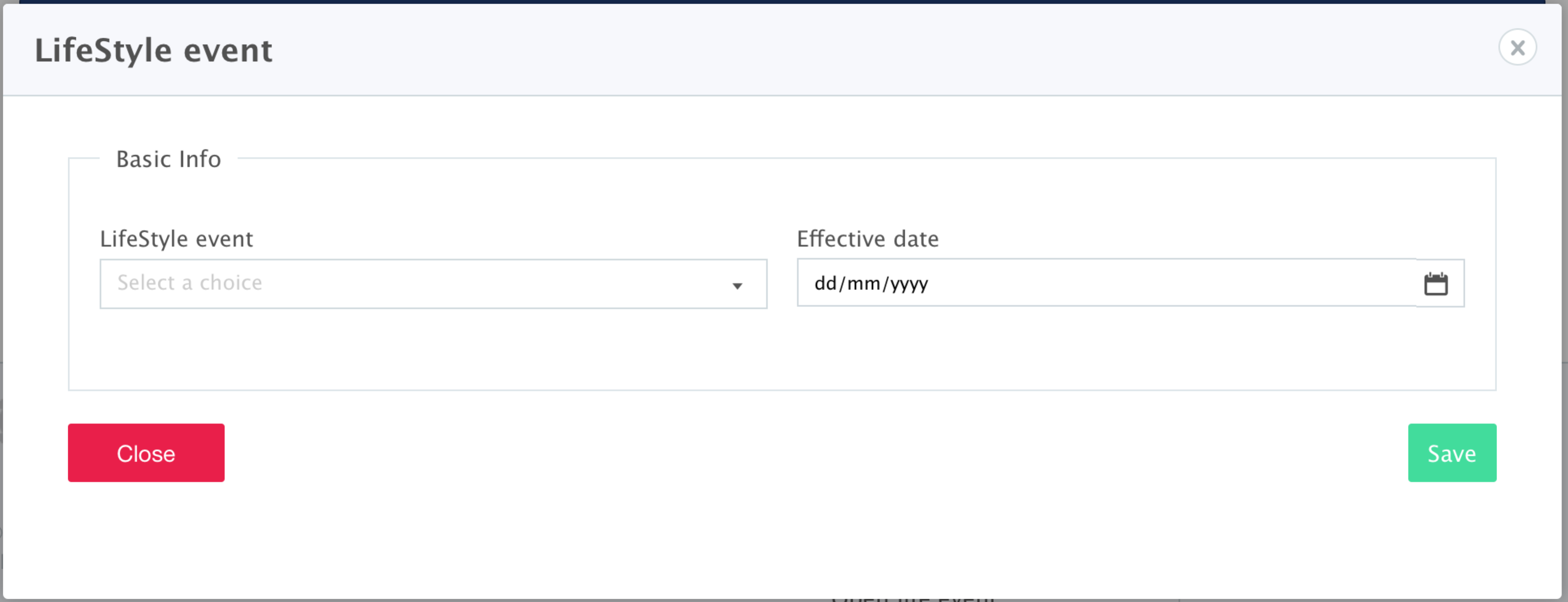
**Index**

****

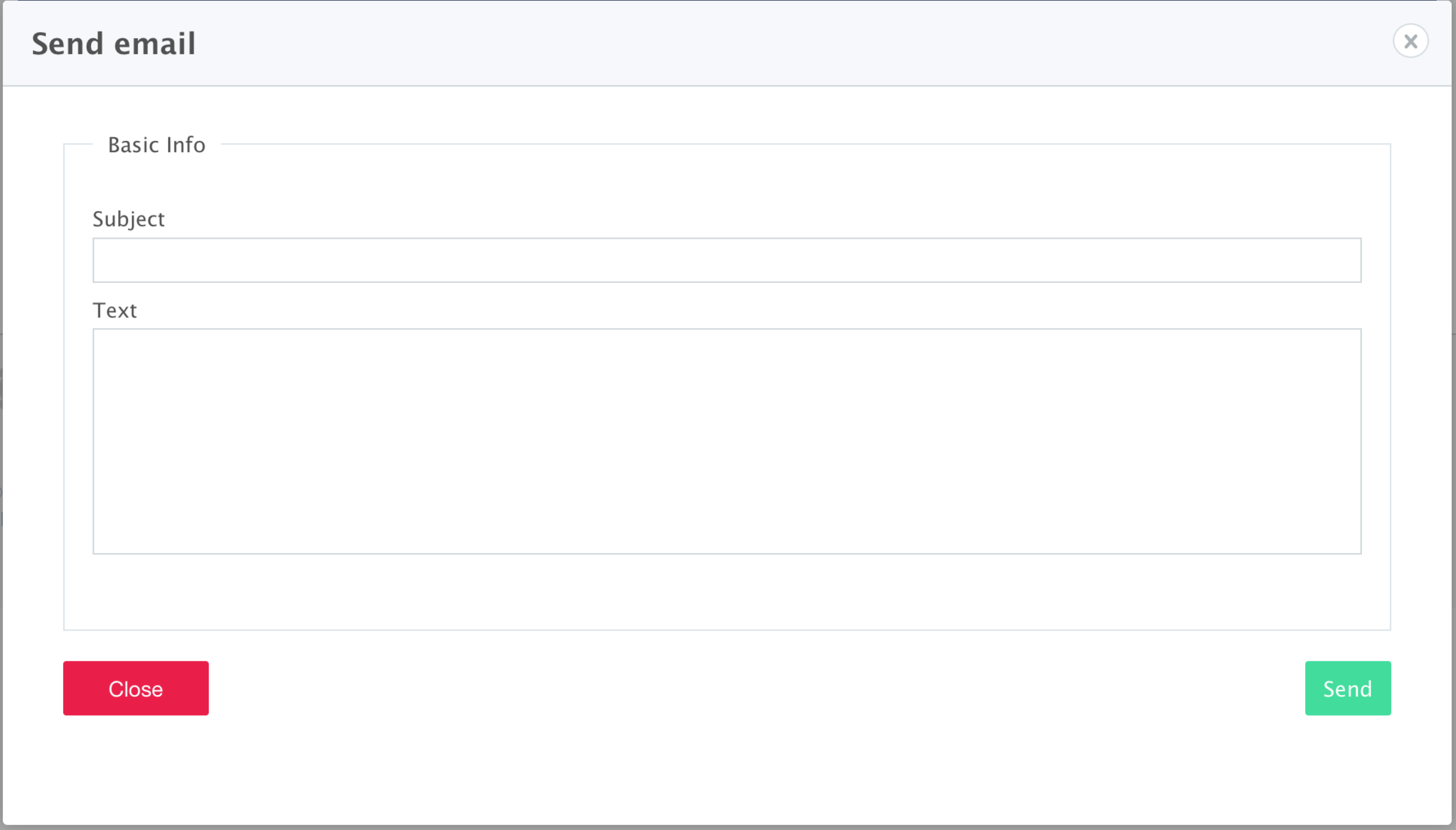
**File upload**

****

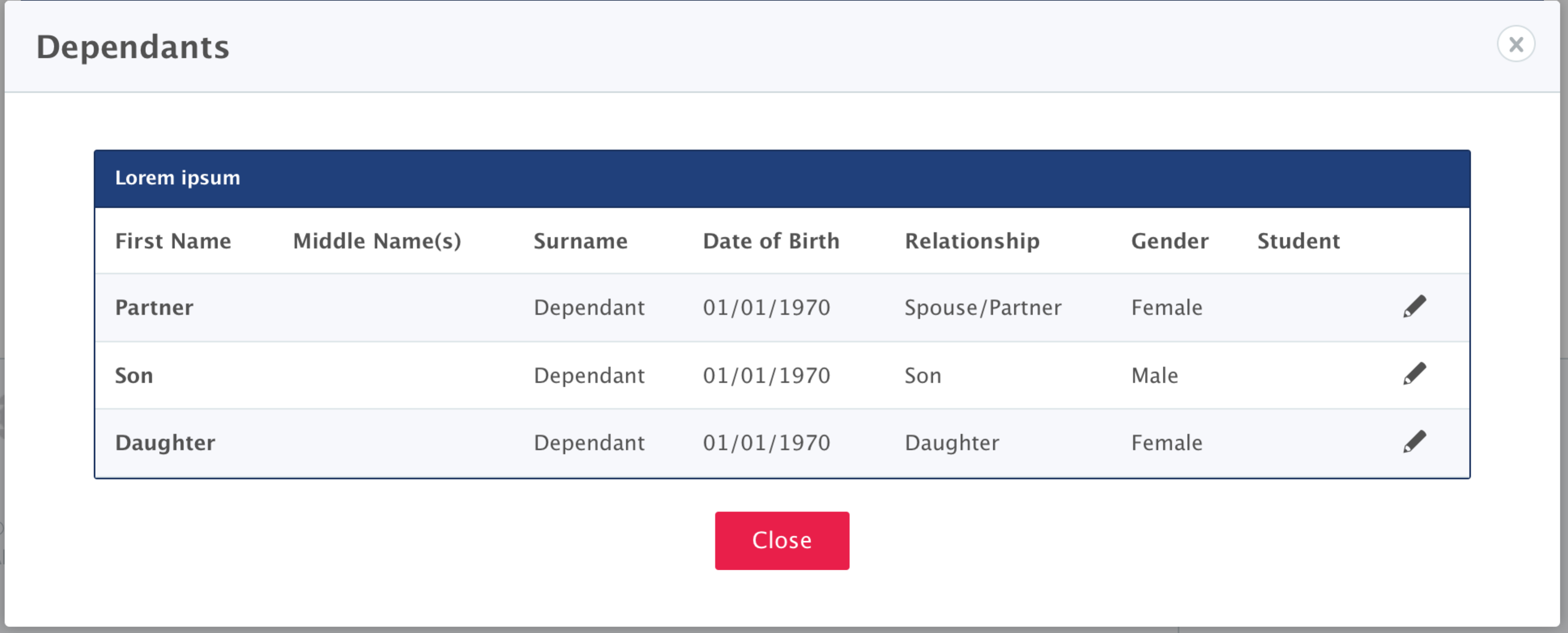
**Bulk Lifestyle events**

****

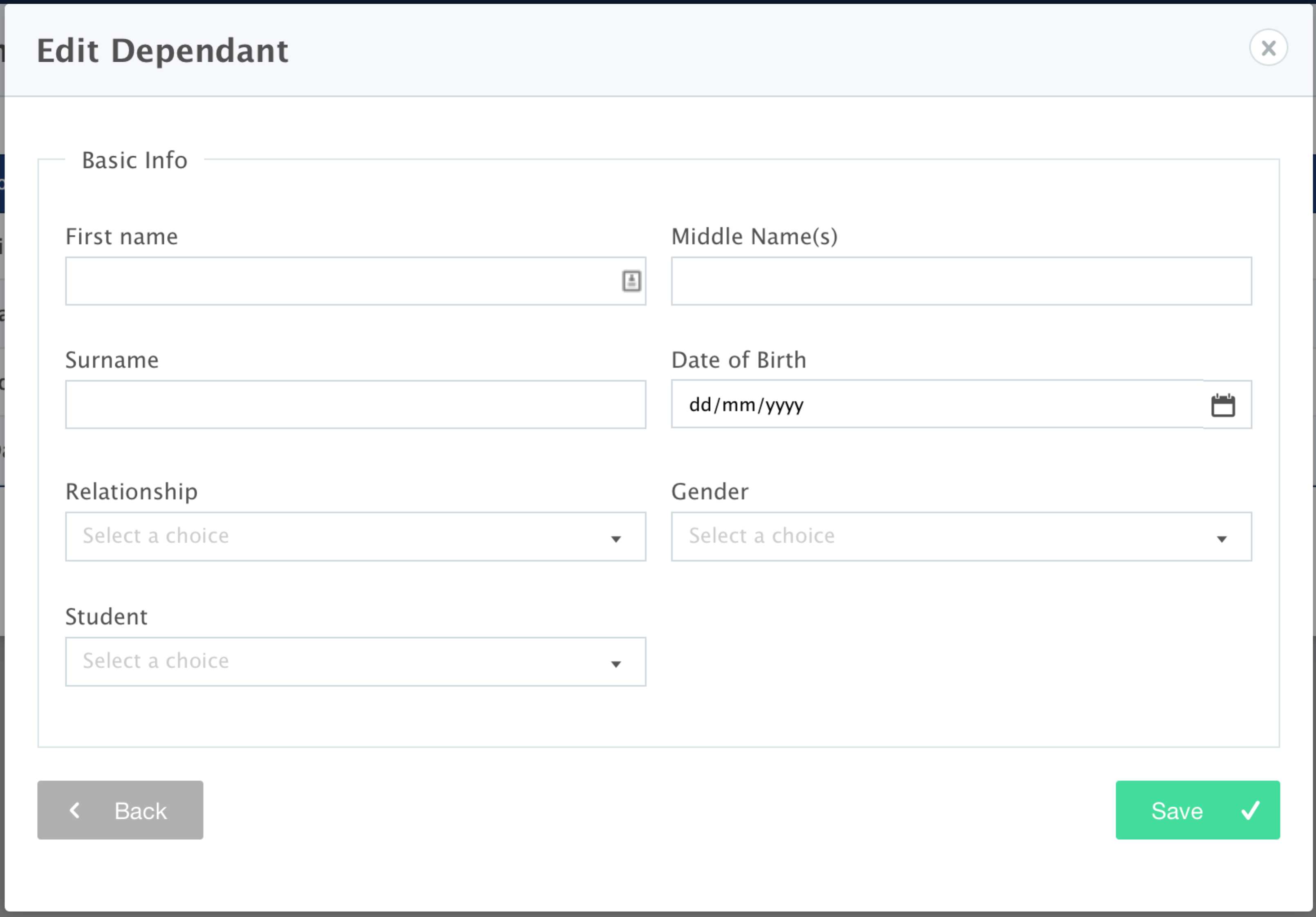
**Bulk email**

****

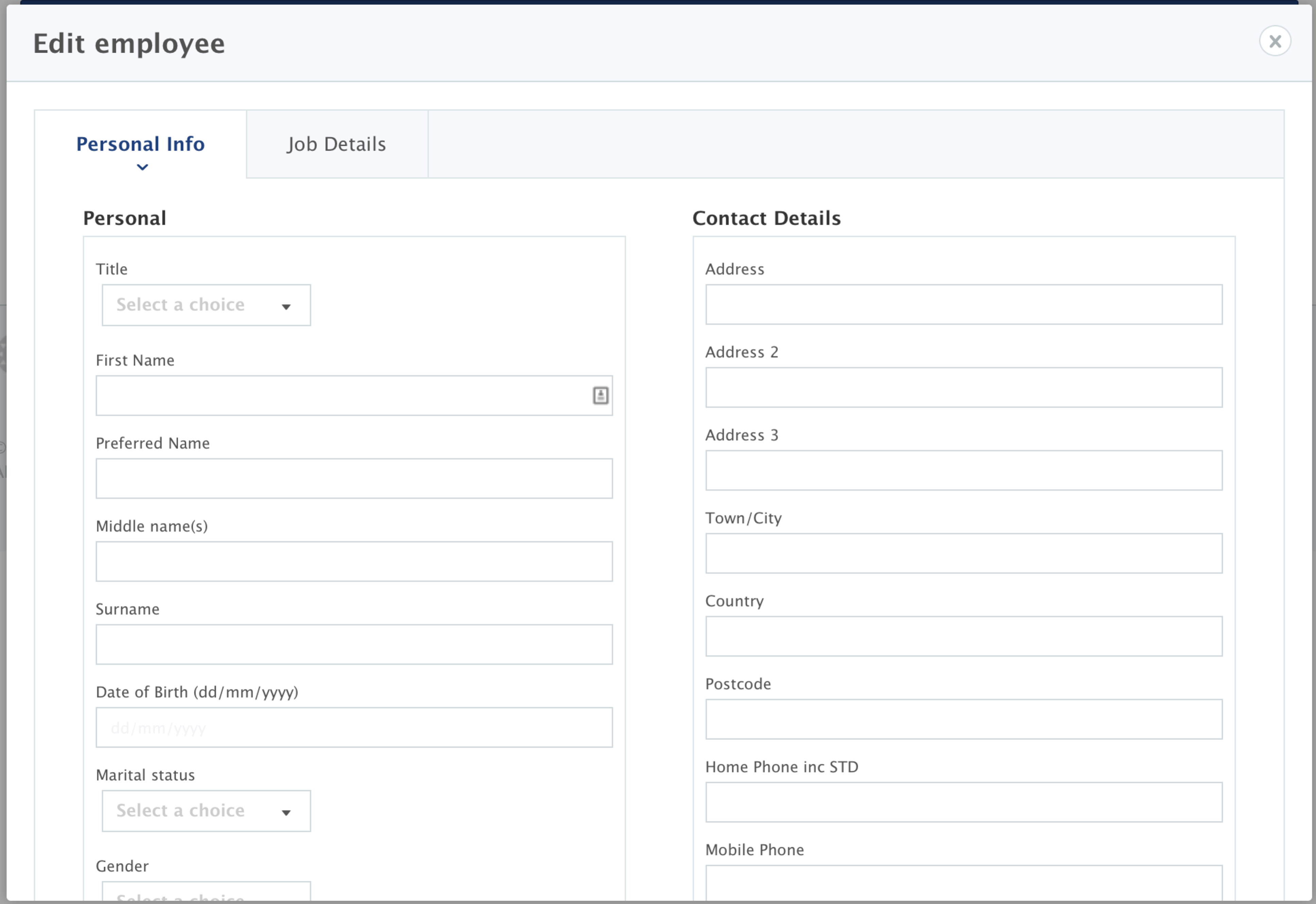
**Dependants details**

****

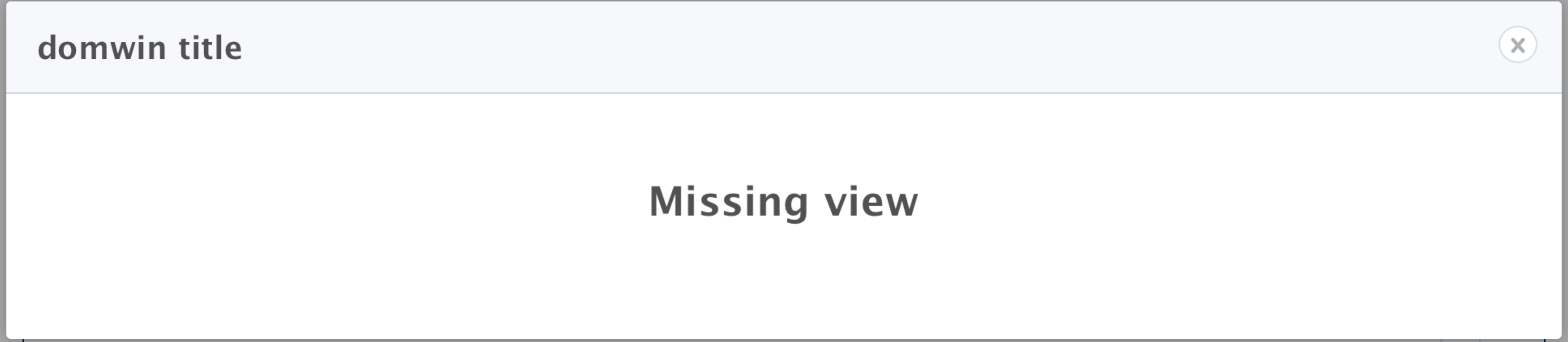
**Dependants details – edit**

****

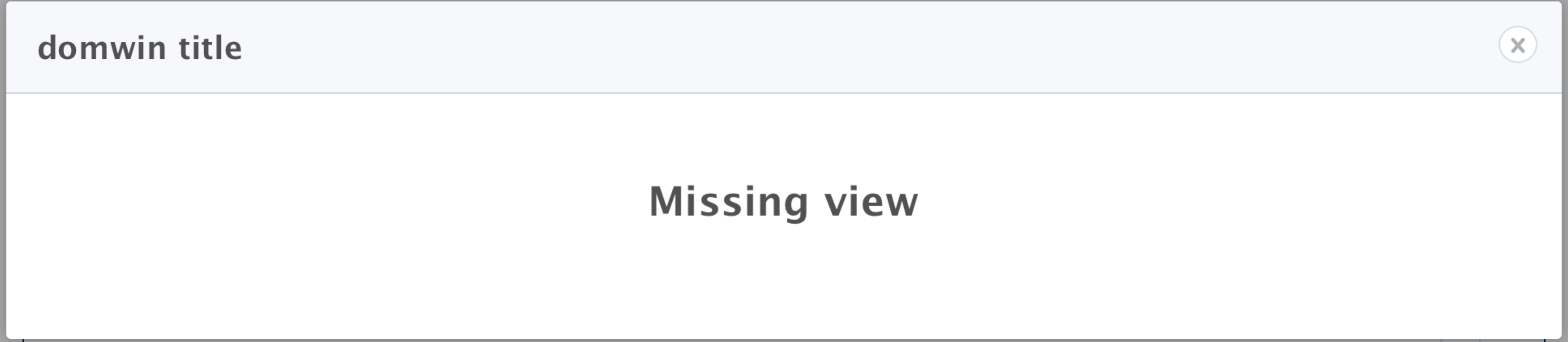
**Edit employee**

****

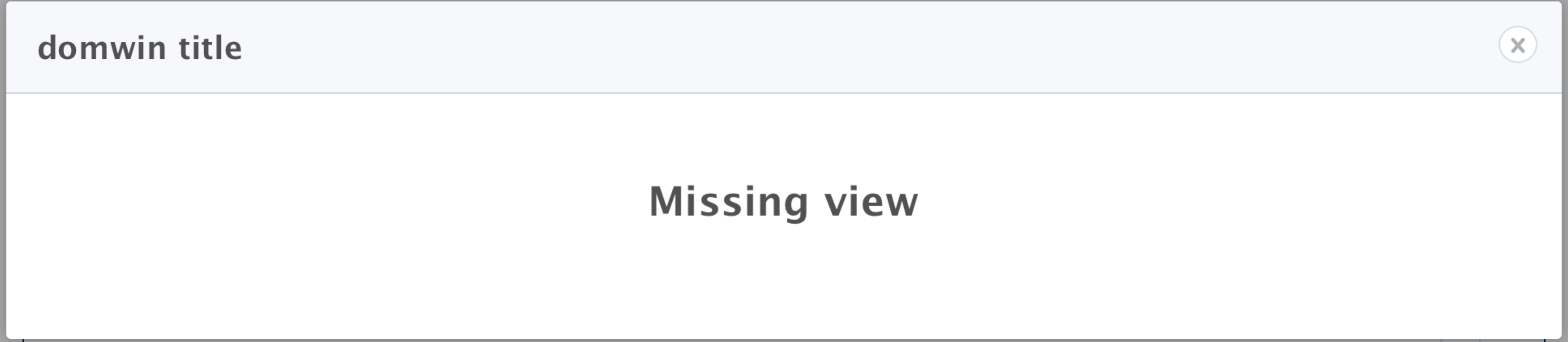
**Show employee**

****

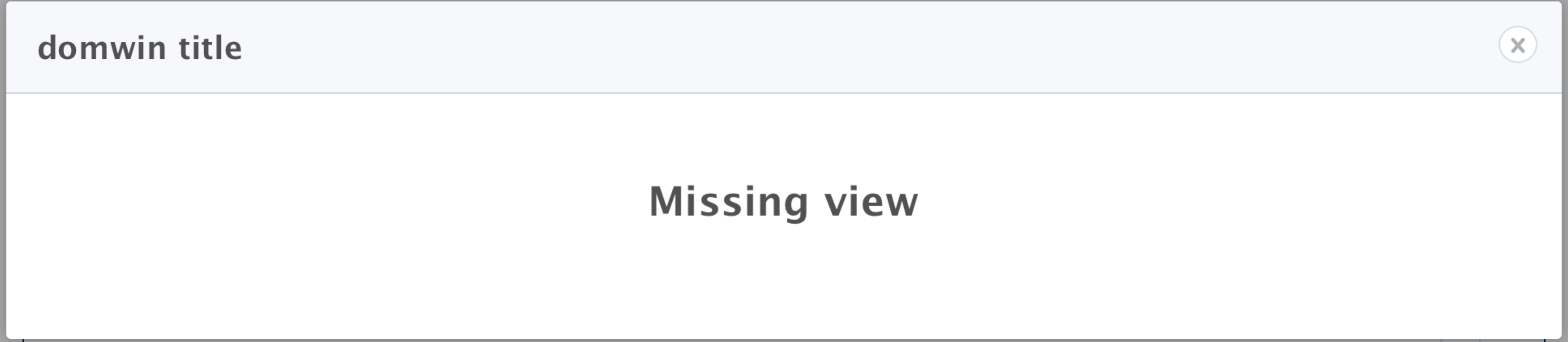
**Attachments**

****

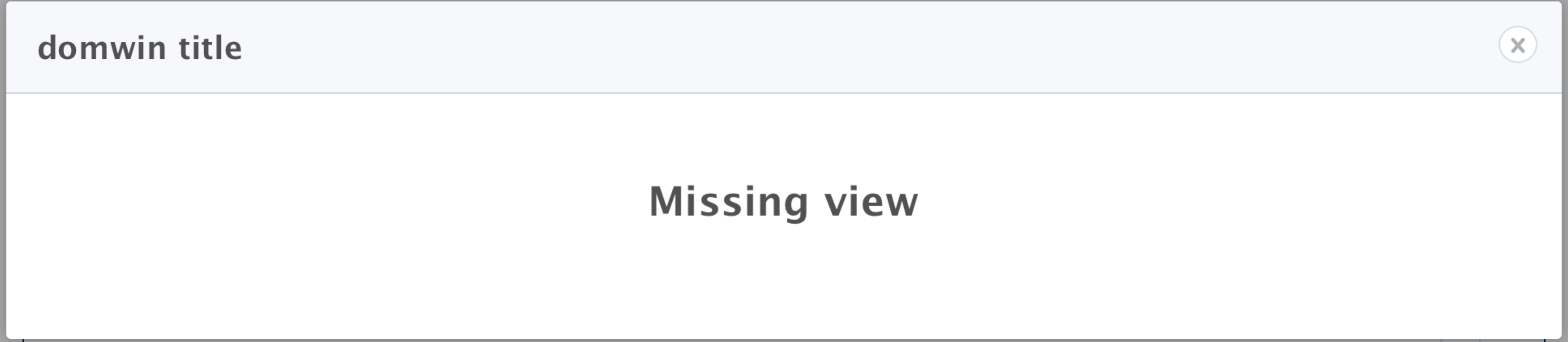
**Attachments – add**

****

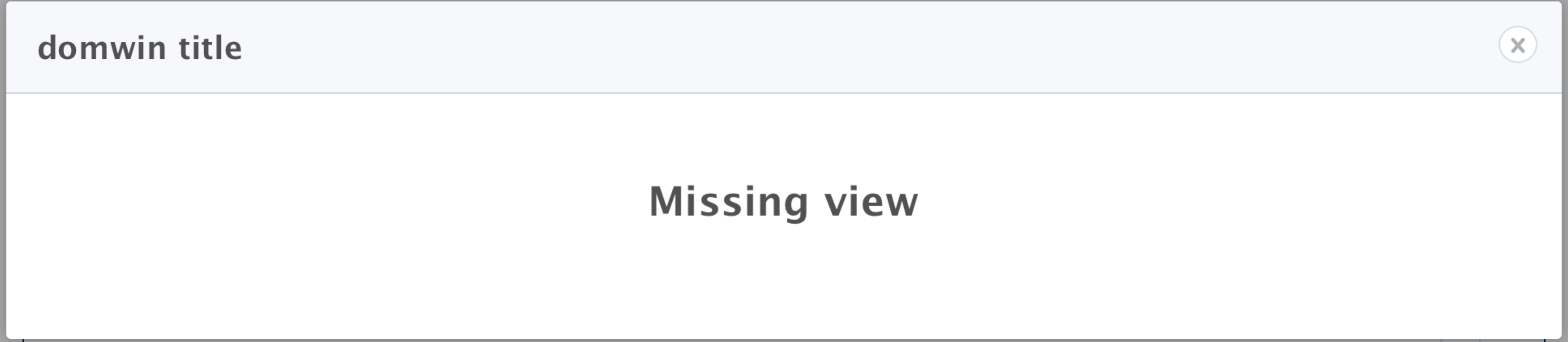
**Attachments – edit**

****

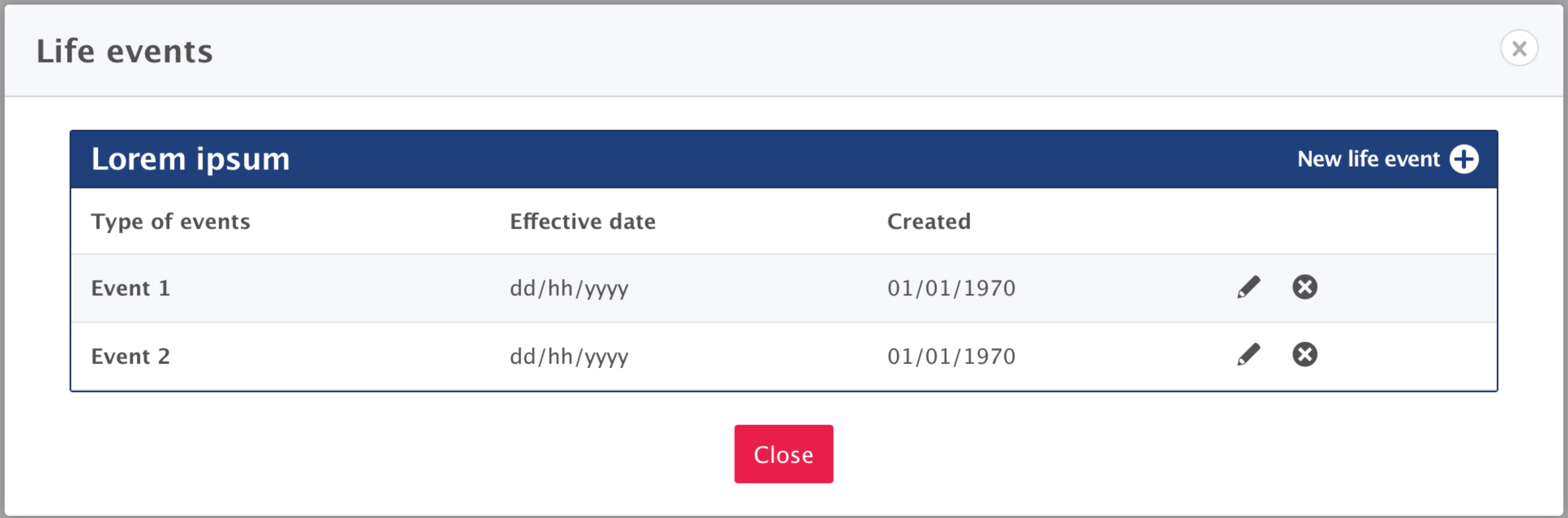
**Attachments – delete**

****

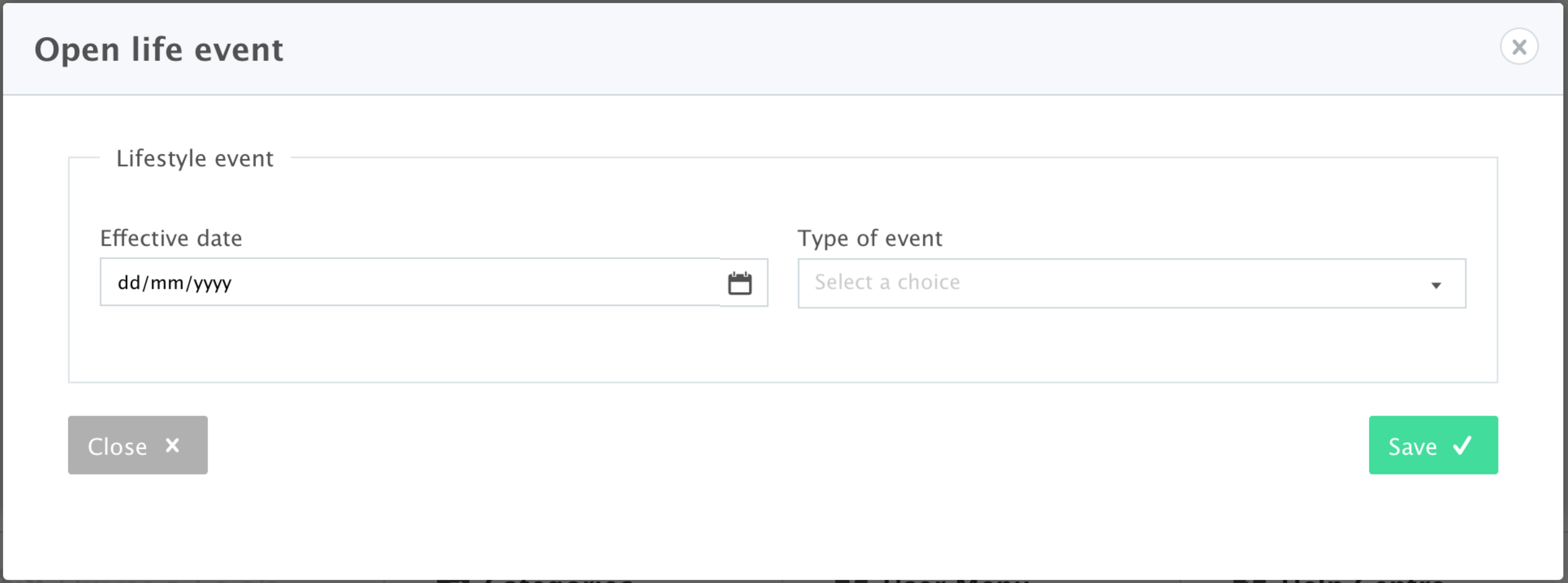
**Logs**

****

**Life events**

****

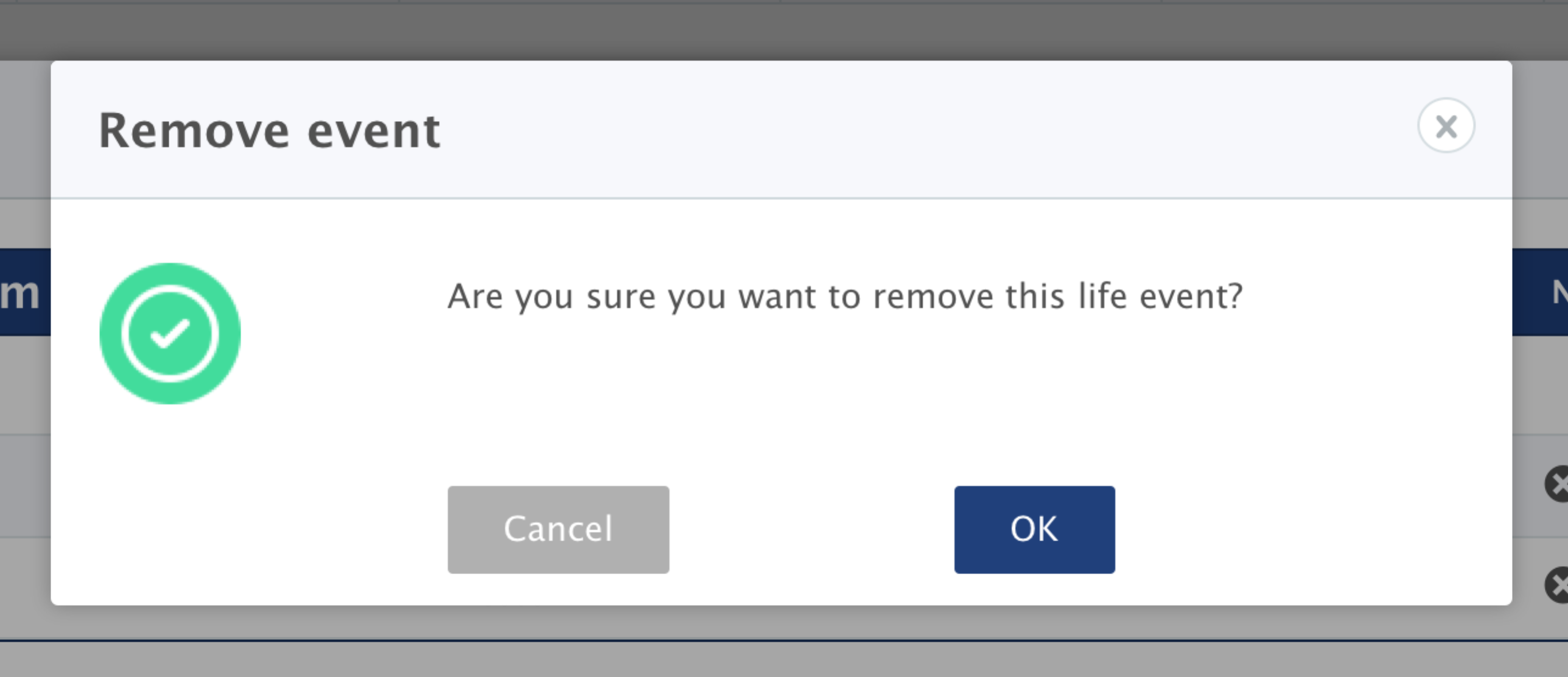
**Life events – open**

****

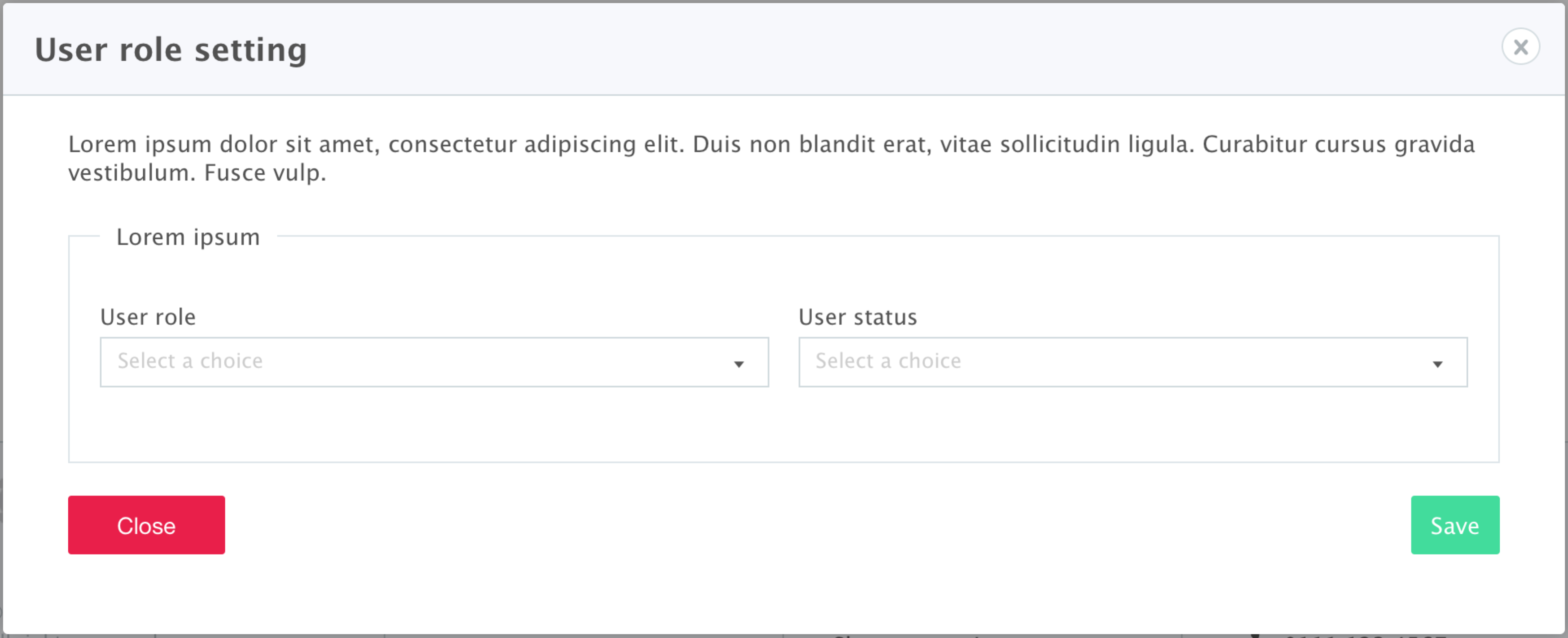
**Life events – edit**

**?**

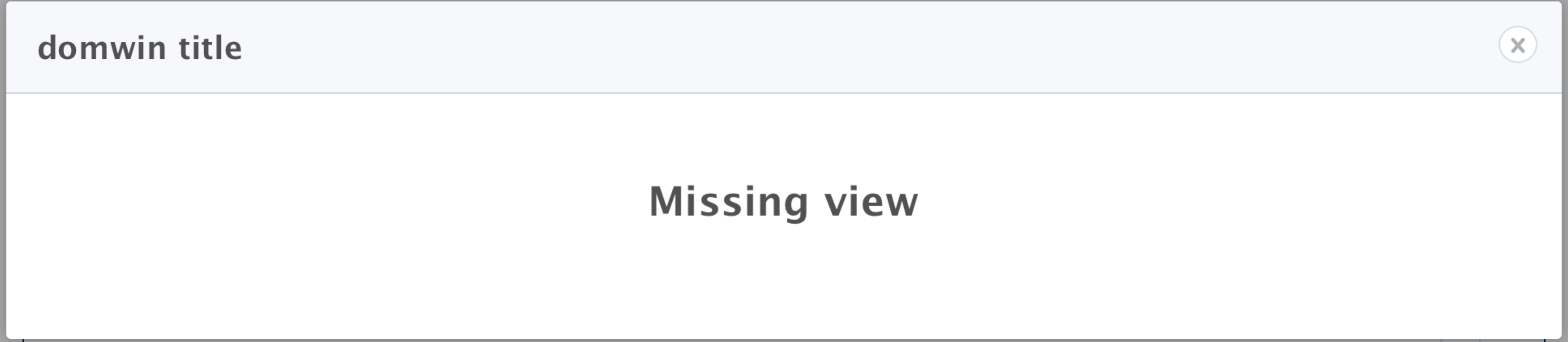
**Life events - rollback**

****

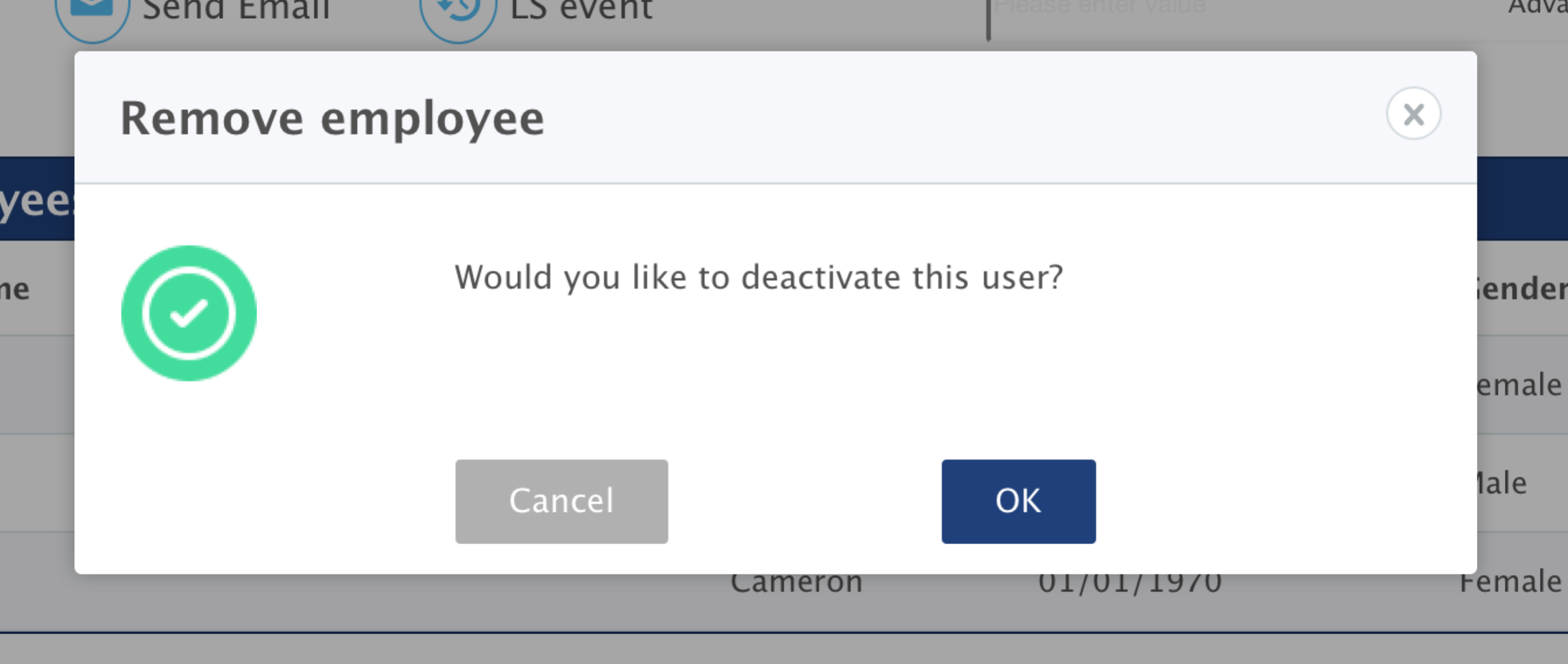
**Permission**

****

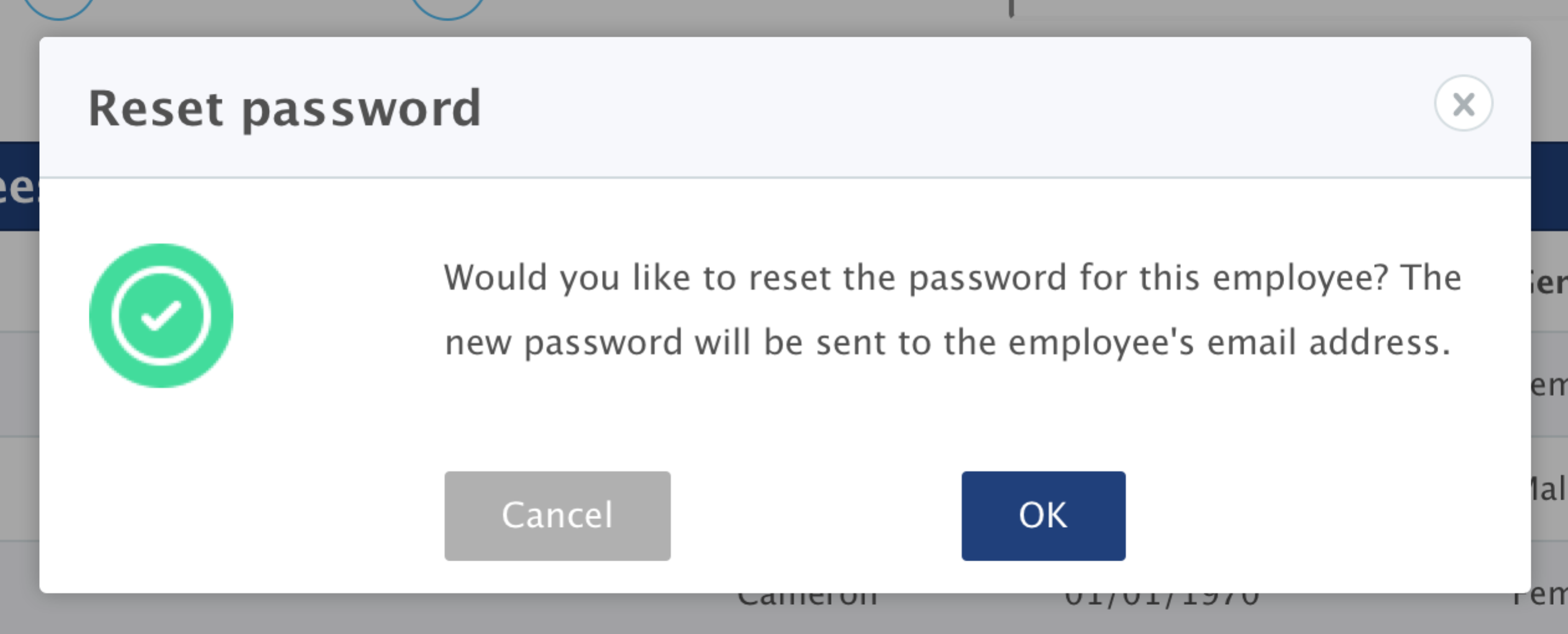
**Set benefits**

****

**Deactivate/Activate user**

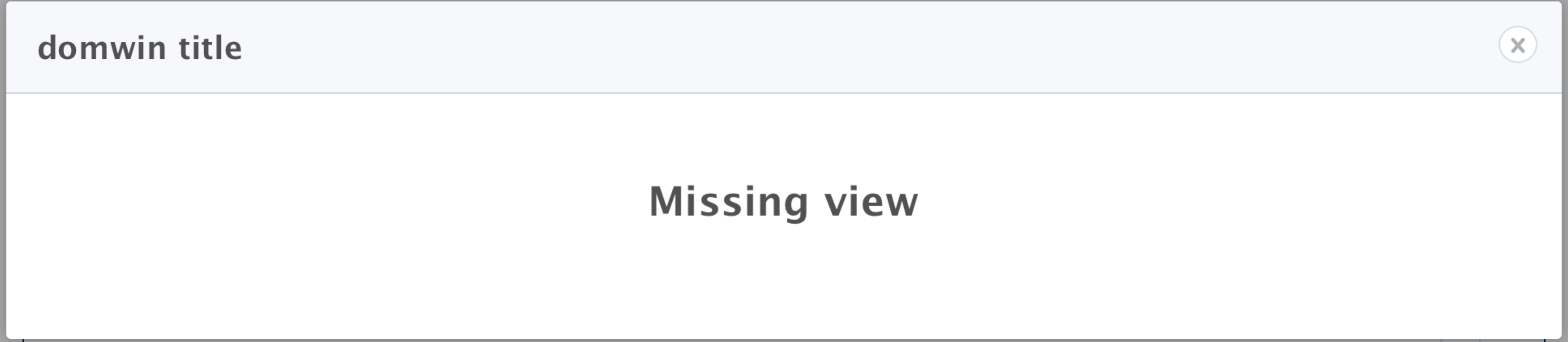
****

**Reset password**

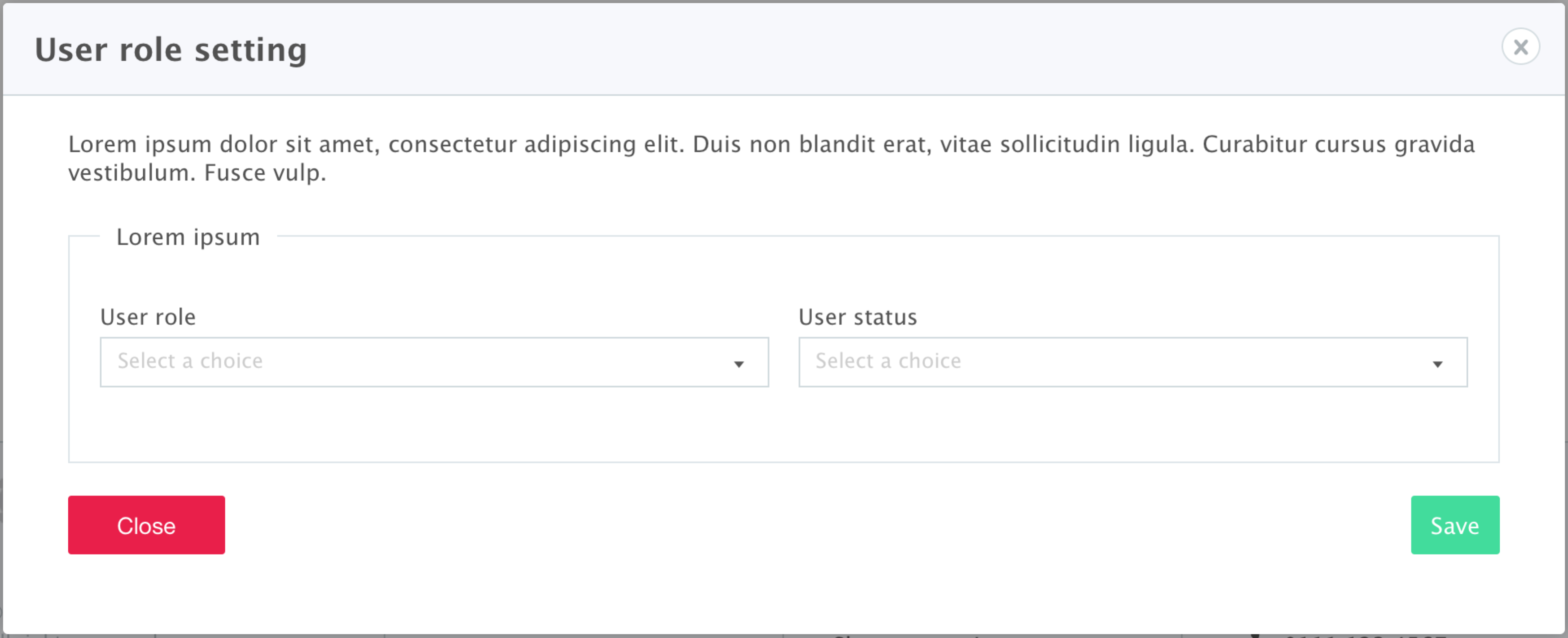


### User module

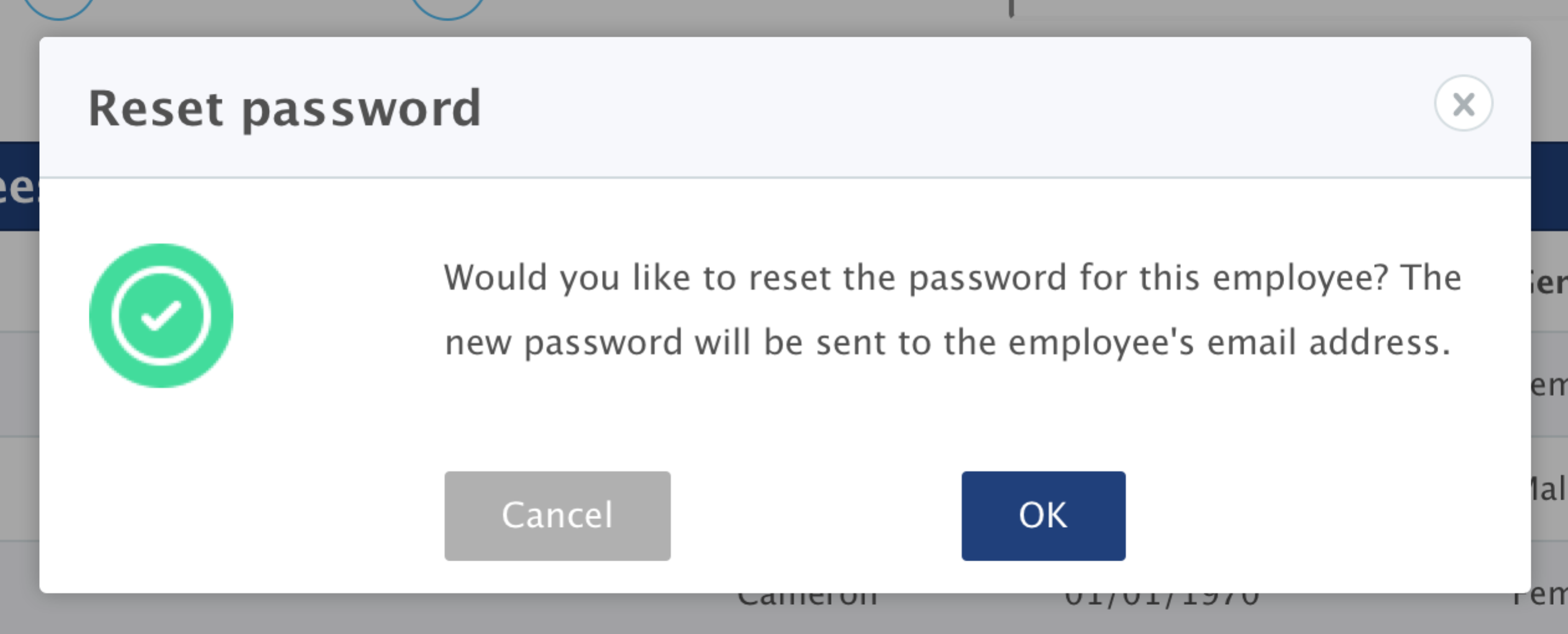
**New user**

****

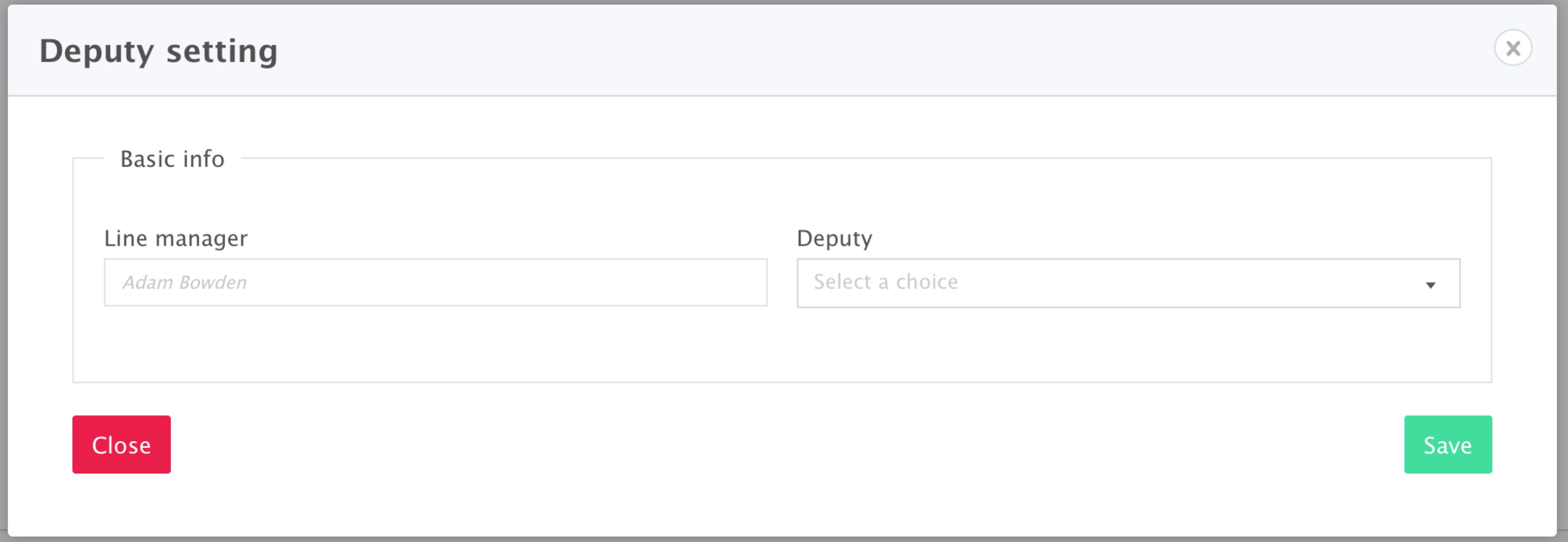
**User role**

****

**Reset password**



**Deputy setting**

****

**Line manager**

