So for People who wants to add a Height and Weight class, follow this 4 steps :

1 - You need to modify some data in your database, go to phpmyadmin in your database, and simply launch the following sql request :

ALTER TABLE `ow\_base\_question` CHANGE `presentation` `presentation`
ENUM('text','textarea','select','date','location','checkbox','multicheckbox','radio','url','password','age','birthdate','range','high','weight') CHARACTER SET utf8 COLLATE
utf8\_general\_ci NOT NULL

2 - You need to declare the two new class in :

ow\_system\_plugins\base\bol\question\_service.php

Line 82 aftrer const Password add like this :
const QUESTION\_PRESENTATION\_PASSWORD = 'password';
const QUESTION\_PRESENTATION\_HIGH = 'high';
const QUESTION\_PRESENTATION\_WEIGHT = 'weight';

Then Line 175 add 2 lines like this ;
self::QUESTION\_PRESENTATION\_PASSWORD => self::QUESTION\_VALUE\_TYPE\_TEXT,
self::QUESTION\_PRESENTATION\_HIGH => self::QUESTION\_VALUE\_TYPE\_TEXT,
self::QUESTION\_PRESENTATION\_WEIGHT => self::QUESTION\_VALUE\_TYPE\_TEXT

Then Line 346 add :
case self::QUESTION\_PRESENTATION\_HIGH :
                    $class = new HighField($fieldName);
                    break;
case self::QUESTION\_PRESENTATION\_WEIGHT :
                     $class = new WeightField($fieldName);
                    break;

Do the same line 498 :
case self::QUESTION\_PRESENTATION\_HIGH :
                    $class = new HighField($fieldName);
                    break;
case self::QUESTION\_PRESENTATION\_WEIGHT :
                     $class = new WeightField($fieldName);
                    break;

Save the file !

3 -  You need to declare also the two new class in :

ow\_system\_plugins\base\bol\user\_service.php

Line 1822 :
case BOL\_QuestionService::QUESTION\_PRESENTATION\_WEIGHT:
case BOL\_QuestionService::QUESTION\_PRESENTATION\_HIGH:
case BOL\_QuestionService::QUESTION\_PRESENTATION\_URL:

Save and closed your file !

4 - Eventually you have to make the new class :
open ow\_core\form\_element.php

Go to the line 832 and insert :
class HighField extends FormElement
{

    /\*\*
     \* Constructor.
     \*
     \* @param string $name
     \*/
    public function \_\_construct( $name )
    {
        parent::\_\_construct($name);

        $this->addAttribute('type', 'text');
    }

    public function renderInput( $params = null )
    {
        parent::renderInput($params);
      $questionData = BOL\_QuestionService::getInstance()->getQuestionData(array( OW::getUser()->getId() ), array( 'height'));
         $value =trim($questionData[OW::getUser()->getId()]['height']);

          $highOptionsString = UTIL\_HtmlTag::generateTag('option', array('value' => '< 4"7 - 140 cm'), true, '< 4"7 - 140 cm  ');

        for ( $i = 140; $i <= 200; $i++ )
        {

            $cm = $i.' cm ';
         $footInt = intval($i/30.48);
         $footIntCm = $footInt\*30.48;
         $InchCm = $i-$footIntCm;
         $Inch = intval($InchCm/2.54);

         $feet = $footInt.'\'\' '.$Inch.'\''.' - '. $i.' cm';
         $attrs['value'] = trim($feet);
            if($attrs['value']==$value)
               $highOptionsString .= UTIL\_HtmlTag::generateTag('option', array('value' => $attrs['value'],'selected' => 'selected'), true, $attrs['value']);
            else
               $highOptionsString .= UTIL\_HtmlTag::generateTag('option', $attrs, true, $attrs['value']);

        }

      $highOptionsString .= UTIL\_HtmlTag::generateTag('option', array('value' => '> 6"6 200 cm'), true, '6"6 200 cm   ');

        $attributes = array();
        $attributes['name'] = $this->attributes['name'];
        $attributes['id'] = $this->attributes['id'];
        $attributes['type'] = 'hidden';
        {
            $result = '<div class="' . $this->getAttribute('name') . '">

                            <div class="ow\_inline owm\_inline">' . UTIL\_HtmlTag::generateTag('select', $this->attributes, true, $highOptionsString) . '</div>
                        </div>';
        }

        return $result;
    }
}

class WeightField extends FormElement
{

    /\*\*
     \* Constructor.
     \*
     \* @param string $name
     \*/
    public function \_\_construct( $name )
    {
        parent::\_\_construct($name);

        $this->addAttribute('type', 'text');
    }

    public function renderInput( $params = null )
    {
        parent::renderInput($params);
      $questionData = BOL\_QuestionService::getInstance()->getQuestionData(array( OW::getUser()->getId() ), array( 'weight'));
         $value =trim($questionData[OW::getUser()->getId()]['weight']);

          $highOptionsString = UTIL\_HtmlTag::generateTag('option', array('value' => '< 99,2 lbs - 45 kg'), true, '< 99,2 lbs - 45 kg   ');

        for ( $i = 45; $i <= 200; $i++ )
        {

           $lbs = round($i\*2.2045,2);

         $weight = $lbs.' lbs - '. $i.' kg';
         $attrs['value'] = trim($weight);
            if($attrs['value']==$value)
               $highOptionsString .= UTIL\_HtmlTag::generateTag('option', array('value' => $attrs['value'],'selected' => 'selected'), true, $attrs['value']);
            else
               $highOptionsString .= UTIL\_HtmlTag::generateTag('option', $attrs, true, $attrs['value']);

        }

      $highOptionsString .= UTIL\_HtmlTag::generateTag('option', array('value' => '> 441 lbs - 200 kg'), true, '> 441 lbs - 200 kg');

        $attributes = array();
        $attributes['name'] = $this->attributes['name'];
        $attributes['id'] = $this->attributes['id'];
        $attributes['type'] = 'hidden';
        {
            $result = '<div class="' . $this->getAttribute('name') . '">

                            <div class="ow\_inline owm\_inline">' . UTIL\_HtmlTag::generateTag('select', $this->attributes, true, $highOptionsString) . '</div>
                        </div>';
        }

        return $result;
    }
}

Save your file and closed It !!

That's set !! Now you have to new class on admin user question, create a question size, just select high class
Create a question weight, select weight class !

Finish !!