

Appeon Mobile

Workshop

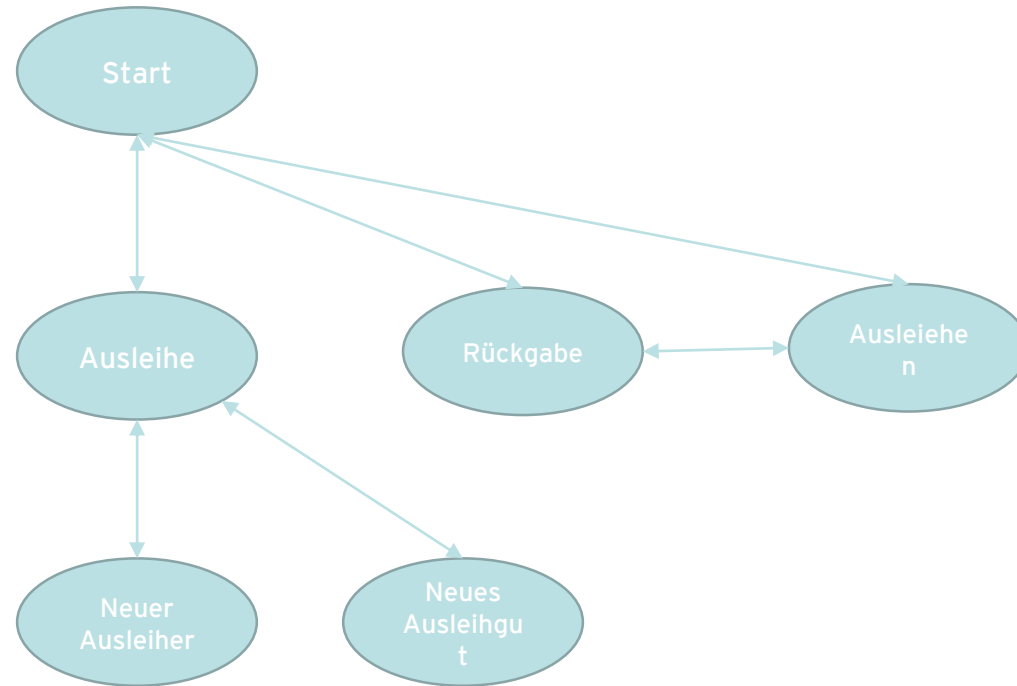
33. PBUGG Meeting

- Die App
- Das Framework

- Kleine Verwaltung von verliehenen Gegenständen.
 - Erfassen und Ändern von Ausleihern
 - Erfassen und Ändern von Gegenständen
 - Erfassen von Verleihungen
 - Erfassen von Rückgaben

33. PBUGG Meeting

- Use cases



33. PBUGG Meeting

	Use Case „Erste Ausleihe“
U	Startet App
S	Startbildschirm
U	Drück „Verleihen
S	Zeigt „Verleih Dialog“ Auswahlfelder (Dropdown) sind leer
U	Drückt „Neuer Ausleiher“
S	Zeigt „neuer Ausleiher Dialog“
U	Erfasst Ausleiher Daten Drück „Speichern und zurück“ Alternativ: Drückt „Abbrechen und zurück“
S	Zeigt „Verleih Dialog“ mit neuem Ausleiher
U	Drückt „Neuer Ausleihgegenstand“

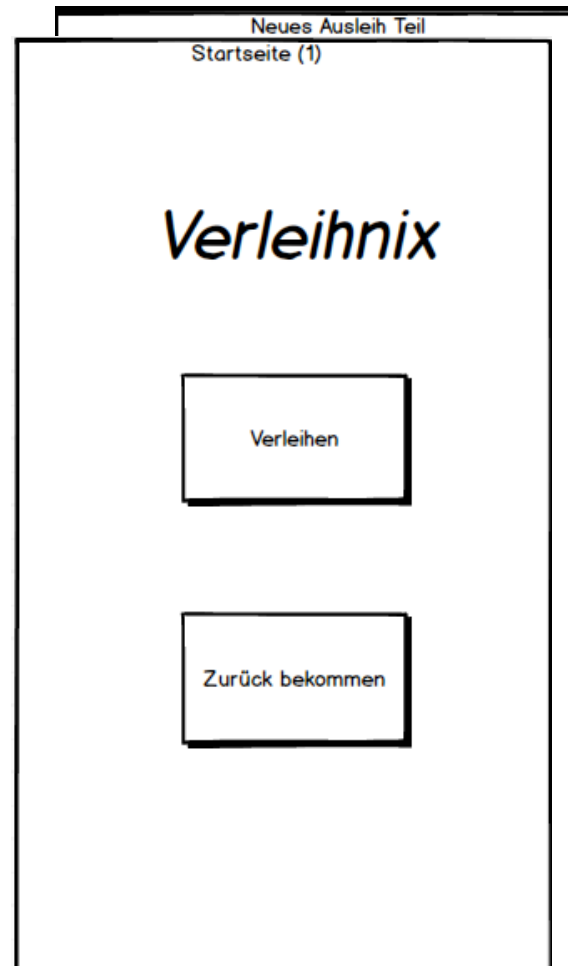
S	Zeigt „Neuer Ausleihgegenstand Dialog“
U	Erfasst Ausleihgegenstand Daten Drück „Speichern und zurück“ Alternativ: Drückt „Abbrechen und zurück“
S	Zeigt „Verleih Dialog“ mit neuem Ausleihgegenstand
S	Trägt Tagesdatum als Ausleihtag vor
U	Ändert ggf. das Datum
U	Ausleiher unterschreibt Drückt „Speichern“ Alternativ: Drückt „Abbrechen“
S	Speichert, säubert die Seite und geht zur Startseite Alternativ: säubert die Seite und geht zur Startseite

33. PBUGG Meeting

	Use Case „ n te Ausleihe“
U	Startet App
S	Startbildschirm
U	Drück „Verleihen
S	Zeigt „Verleih Dialog“ Auswahlfelder (Dropdown) haben Daten
U	Wählt Ausleiher
U	Wählt Ausleihgegenstand
S	Trägt Tagesdatum als Ausleihtag vor
U	Ändert ggf. das Datum

U	Ausleiher unterschreibt Drückt „Speichern“ Alternativ: Drückt „Abbrechen“
S	Speichert, säubert die Seite und geht zur Startseite Alternativ: säubert die Seite und geht zur Startseite

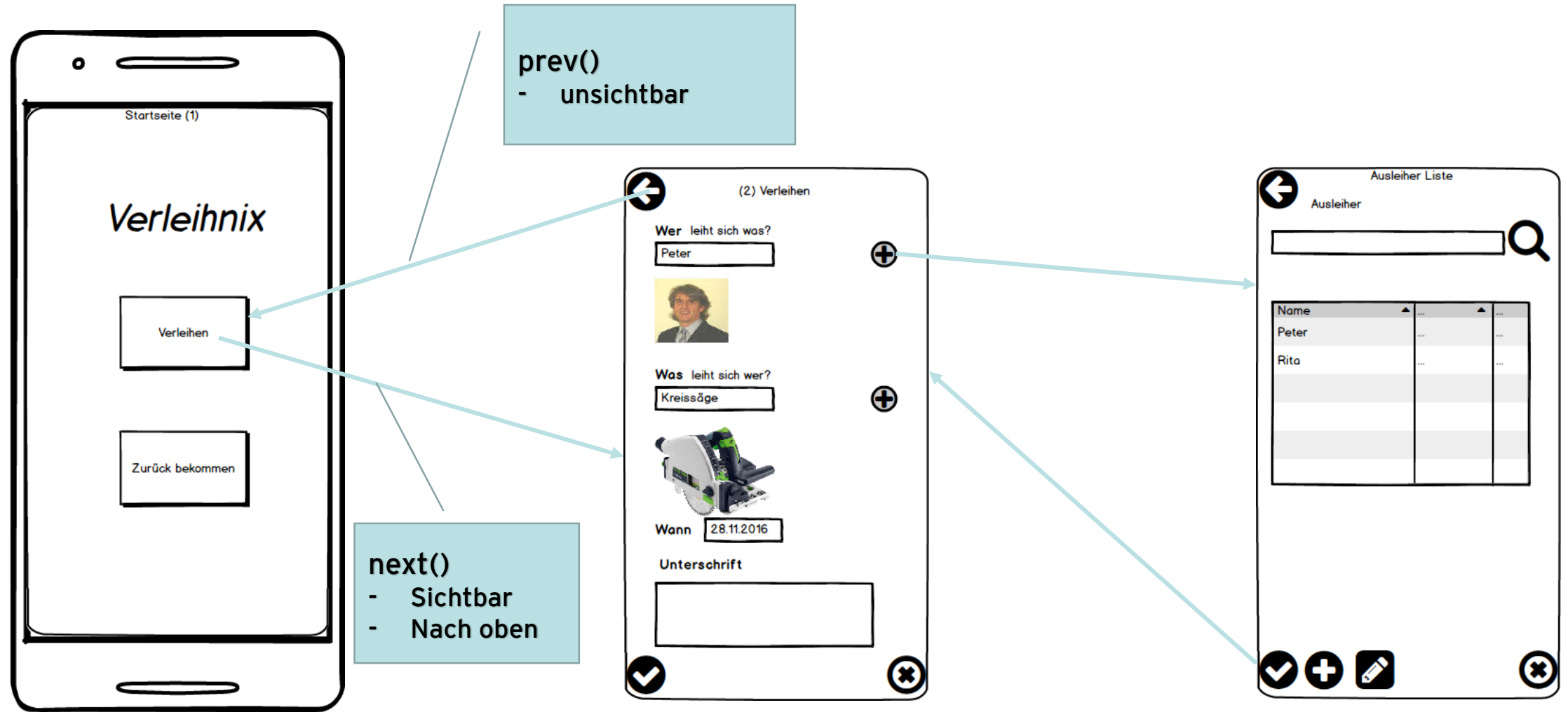
33. PBUGG Meeting



Jede Karte „weiß“ welche Karten sie mit welchen Parametern aufrufen muss.

Das „sagt“ sie der Steuerung und die sorgt dafür, dass die Karte nach oben kommt.

33. PBUGG Meeting



- Die Steuerung „hält“ den Kartenstapel und registriert jede Karte
 - uo_m_base ivno_baseobject[]
 - of_register (...)
- Die Steuerung führt eine Historientabelle für Vorwärts/ Rückwärts

33. PBUGG Meeting

- Kartenstapel
- Alle Karten ausbreiten
- Reihenfolge durch Pfeile
- Datenaustausch an die Pfeile

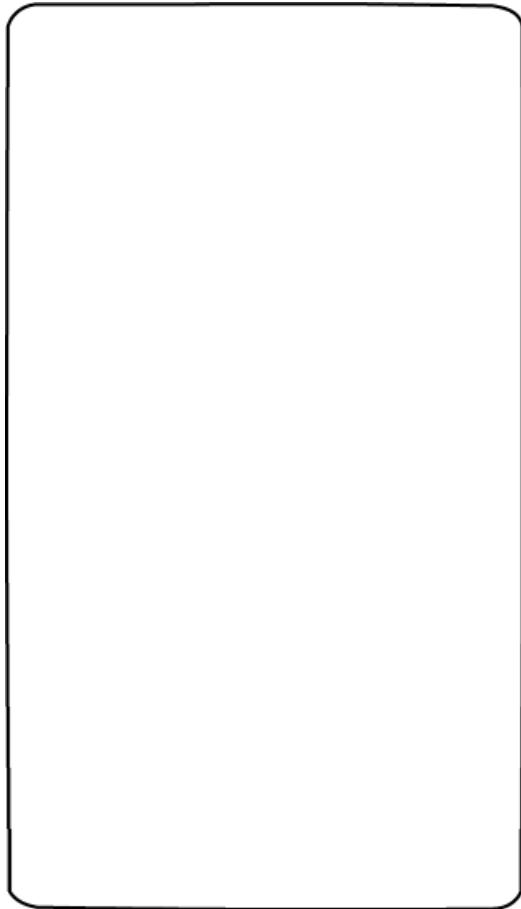
- Mobile App als Single Document Interface (SDI)
- „Masken“ sind alle auf dem Window
 - Wie Spielkarten auf dem Stapel
- Die aktuelle Karte ist immer oben
- Die Interaktion bestimmt, welche Karte als nächstes nach oben kommt.
- Bei der Interaktion werden Informationen zwischen den Karten ausgetauscht

- UserObject
- DataWindow
- Navigation
 - Nächste
 - Vorherige
 - Andere

33. PBUGG Meeting

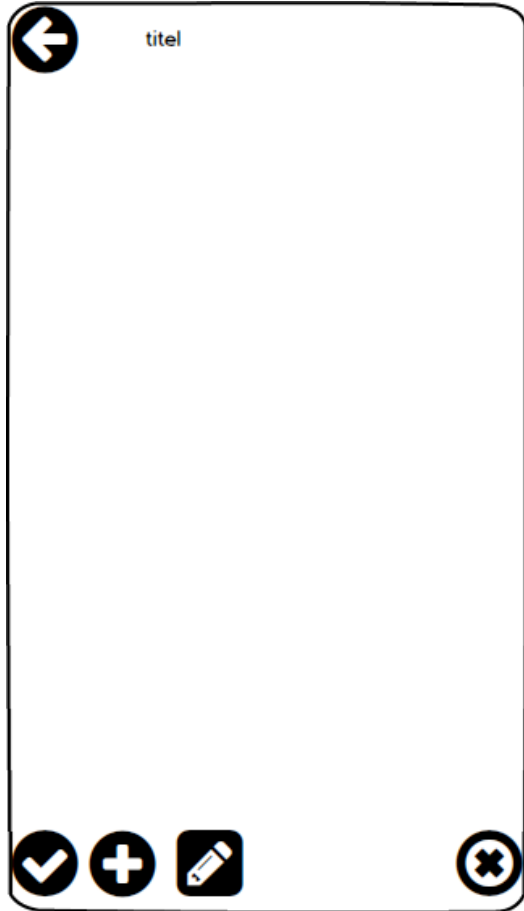
- uo_m_base
- uo_m_buttonbar
- uo_m_single
- uo_m_masterdetail
- uo_m_searchlist

33. PBUGG Meeting



- ue_insertrow (any aa_arg[20])
- ue_retrieve (any aa_arg[20])
- ue_start()
- ue_refreshdddw(any aa_arg[20])
.. aus retrieve

33. PBUGG Meeting

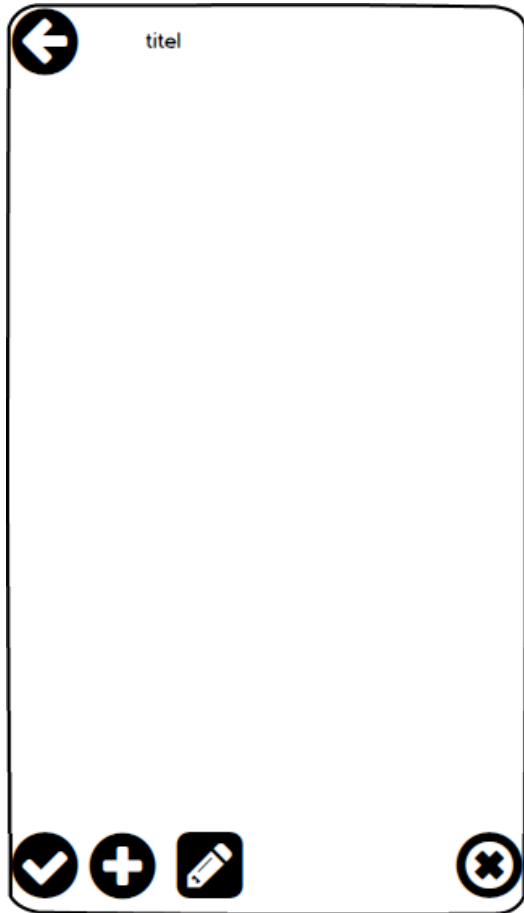


Alle Karten haben (mehr oder weniger) die gleichen Aktionen.

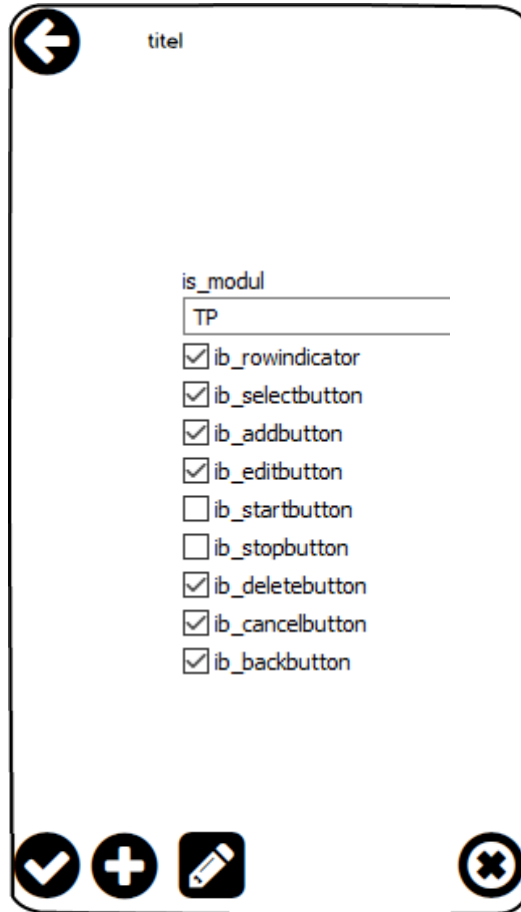
- Zurück
- Auswahl
- Zufügen
- Editieren
- Löschen
- Abbrechen

Dafür gibt es ein DataWindow Objekt, mit all diesen Buttons.
Die Button-Icons werden aus einer Parameter-Tabelle geladen
Die Buttons sind über Instanzvariablen ein-/ausschaltbar

33. PBUGG Meeting



```
case "b_back" //, "t_title"  
    this.event ue_backclicked( )  
case "b_select"  
    this.event ue_selectclicked( )  
case "b_add"  
    this.event ue_addclicked( )  
case "b_edit"  
    this.event ue_editclicked( )  
case "b_delete"  
    this.event ue_deleteclicked( )  
case "b_cancel"  
    this.event ue_cancelclicked( )  
case "b_menu"  
    this.event ue_popupmenu( )
```



Constructor

```
this.settransobject( sqlca )
```

```
this.retrieve( is_modul )
```

is_modul ist eine Instanzvariable, die unter Properties gesetzt wird (z.B. "TP"). Mit diesem Wert werden aus der Property Tabelle die Icons geladen

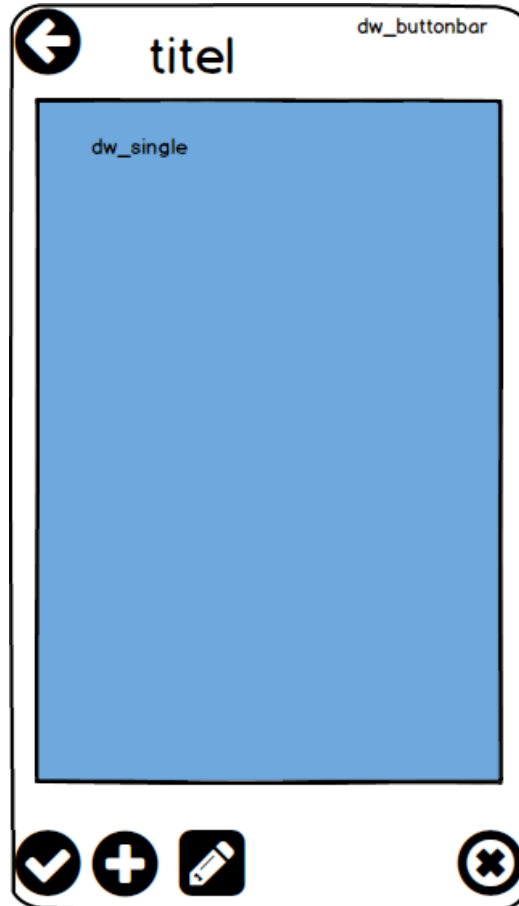
```
this.of_showtitle( this.title )
```

Setzt den Titel aus den Properties

```
this.of_showbutton( )
```

Zeigt die Buttons, die unter Properties ausgewählt wurden

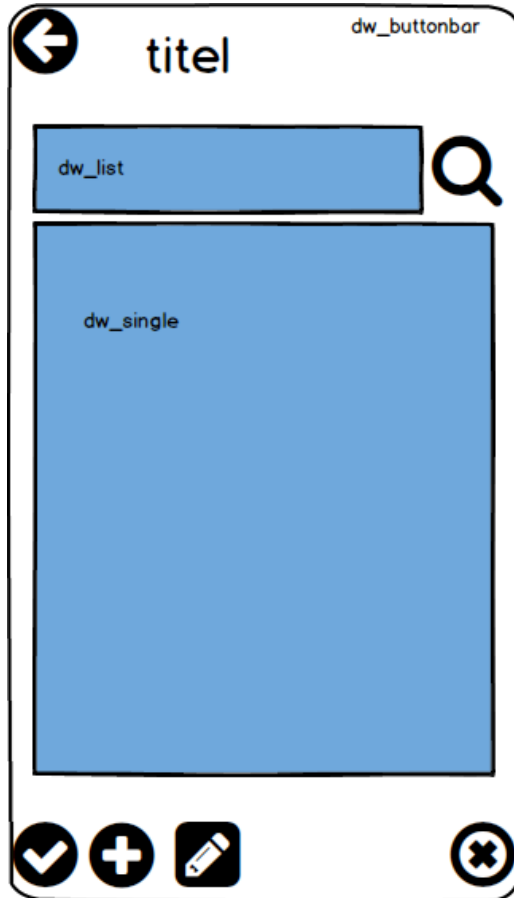
33. PBUGG Meeting



Enthält

- dw_buttonbar aus uo_n_buttonbar
- dw_single aus uo_basedw (datawindow)
 - Keine besonderen Events

33. PBUGG Meeting



Funktion

- Searchbox für suche innerhalb des dw_single

Enthält

- dw_buttonbar aus uo_n_buttonbar
- dw_list
- dw_single aus uo_basedw (datawindow)
 - Keine besonderen Events

- w_main
 - Diese Fenster ist quasi der „Spieltisch“, auf den alle Karten gelegt werden, die am „Spiel“ teilnehmen sollen.
 - *Es enthält zur Zeit keine wichtige Funktionalität mehr und fliegt ggf. wieder aus dem Framework raus.*

- `u_cst_m_steuerung`
 - `of_register`

- `(uo_m_base anvo_object, String as_objectname)`

Alle „Karten“ die am „Spiel“ teilnehmen, müssen registriert werden.

Das Object ist die Karte selbst und liegt auf dem „Spieltisch“ (dem Window)

Der objectname wird aus praktischen Gründen als Konstante in der Steuerung deklariert.

- **u_cst_m_steuerung**
 - **of_shownext**
 - (string as_object, any aa_arg[20], boolean ab_retrieve)
 - as_object ist die „Karte“, die nach oben soll. Hier nimmt man am besten die Konstante mit der das Objekt registriert wurde
 - aa_any kann retrieval Argumete für ein Retrieve enthalten
 - ab_retrieve steuert, ob ein Retrieve gemacht werden soll.
 - **Varianten**
 - of_shownext (string as_object, any aa_arg[20])
 - » return this.of_shownext(as_object, aa_arg[],true)
 - of_shownext (string as_object)
 - » return this.of_shownext(as_object, la_arg, False)

- `u_cst_m_steuerung`
 - `of_getlast()` return `uo_m_base`
 - Holt die letzte Karte vom Stapel
 - OHNE sichtbar machen
 - NUR zum Bearbeiten (Werte lesen oder setzen)

- Aufruf einer „Karte“ ohne Retrieve

```
gnv_app.inv_steuerung.of_shownext( gnv_app.inv_steuerung.CS_VERLEIHEN)
```

- Aufruf einer Karte mit Retrieve

```
CASE "p_suche_gegenstand"
```

```
ll_ausleiher_id = this.getItemNumber(row,"ausleiher_id")
```

```
la_args[1] = ll_ausleiher_id
```

```
gnv_app.inv_steuerung.of_shownext (
```

```
    gnv_app.inv_steuerung.cs_gegenstandsliste_ausleiher , la_args )
```

- Aufruf einer Karte mit zu füllenden dddw
 - Aufruf of_shownext mit arg[]
 - Codieren des ue_refreshdddw

```
long ll_rc,&  
    ll_ausleiher_id  
datawindowchildldwc_dwc
```

```
ll_ausleiher_id = long(aa_arg[1])  
ll_rc = dw_single.getchild( "ausleiher_id", ldwc_dwc )  
ldwc_dwc.settransobject( SQLCA)  
ldwc_dwc.retrieve(ll_ausleiher_id)
```

- Setzen von Werten in der aufrufenden Karte
 - Holen der letzten Karte

```
Invo_obj = gnv_app.inv_steuerung.of_getlast( )      // Letzte Karte holen
choose case classname(Invo_obj)                    // Kartentyp bestimmen
  case "iuo_verleihen,,                            // beim richtigen Kartentyp
    Invo_verleihen = Invo_obj                      // ggf. Klasse zueisen
    ll_row = Invo_verleihen.dw_single.getrow()      // aktuelles Row holen
    if ll_row > 0 then                              // Wert setzen
      Invo_verleihen.dw_single.setitem( ll_row, "ausleiher_id",
                                         dw_list.getItemNumber(ll_listrow, "id") )
      Invo_verleihen.event ue_refreshdddw(la_arg)
    end if
  ...
```

- `n_cst_m_appmanager`
 - Viele nützliche Funktionen (wächst ständig ..)
 - `of_connect`
 - `of_connect_server`
 - `of_phonecall`
 - `of_photo`
 - `of_encrypt`
- Globale Variable `gnv_app`