

INTRODUCTION

19 August 2025 14:15

UNIVERSITY STUDENT MANAGEMENT SYSTEM

KEY FEATURES:

- Student management with CRUD operations
- Automatic grade conversion
- Email notifications & Adobe forms
- Failed student tracking
- Statistics & reporting
- Many more...

OBJECTS:

- Tables: ZSTUDENT_T, ZFSTUDENT_T, ZBHVRULES
- Data Elements: ZSTUDENTID_DE, ZSTUDENTNAME_DE, etc.
- Search Help: ZSHSTUDENTBHV
- Adobe Form: ZUNI_GRADE_SYSTEM_AFORM

STRUCTURE:

- Main: ZUNI_GRADE_SYSTEM.abap
- Classes: ZUNI_GRADE_SYSTEM_CLASS.abap
- Data: ZUNI_GRADE_SYSTEM_DATA.abap
- Forms: ZUNI_GRADE_SYSTEM_FORMS.abap
- Modules: ZUNI_GRADE_SYSTEM_MODULES.abap

KEY METHODS:

- lcl_grade_converter=>convertscoretograde
- lcl_age_calculator=>calculate_age
- lcl_set_cell_color=>set_cell_color

MAIN FUNCTIONS:

- ADD - Add new student
- UPD - Update existing student
- SEARCH - Search by ID or name
- SENDMAIL - Email notifications
- SALV/OOALV - Display ALV views
- STATS - Show statistics

SCREENS:

- 0001: Main - SALV pop-up
- 0002: OO ALV view

T-CODE:

- ZHSTUGS - for main screen (0001)
- ZHSTUGSTM - for main table maintenance

DB-SCREEN-ADOBEFORM-ALV-VIEWS

19 August 2025 14:15

STUDENT TABLE

Transparent Table 2STUDENT_T Active

Short Description STUDENT TABLE

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes

Field	Key	Init...	Data element	Data Type	Length	Decim...	Short Description
MANDT	✓	✓	MANDT	CLNT	3		0 Client
STUDENTID	✓	✓	2STUDENTID_DE	INT1	3		0 STUDENT ID
STUDENTNAME		✓	2STUDENTNAME_DE	CHAR	30		0 STUDENT NAME
STUDENTLNAME		✓	2STUDENTLNAME_DE	CHAR	20		0 STUDENT LASTNAME
STUDENTGEN		✓	2STUDENTGEN_DE	CHAR	1		0 STUDENT GENDER
STUDENTBDATE		✓	2STUDENTBDATE_DE	DATS	8		0 STUDENT BIRTH DATE
STUDENTAGE		✓	2STUDENTAGE_DE	CHAR	2		0 STUDENT AGE
STUDENTMAIL		✓	2STUDENTMAIL_DE	CHAR	30		0 STUDENT MAIL
STUDENTSCORE		✓	2STUDENT_SCORE_DE	DEC	6		3 STUDENT SCORE
STUDENTGRADE		✓	2STUDENT_GRADE_DE	CHAR	2		0 STUDENT GRADE
STUDENTBHV		✓	2STUDENTBHV_DE	CHAR	10		0 STUDENT BEHAVIOUR

FAILED STUDENT TABLE

Transparent Table 2FSTUDENT_T Active

Short Description STUDENTS WHO FAILED CLASS

Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields Indexes

Field	Key	Init...	Data element	Data Type	Length	Decim...	Short Description
STUDENTID	✓	✓	2STUDENTID_DE	INT1	3		0 STUDENT ID
STUDENTNAME		✓	2STUDENTNAME_DE	CHAR	30		0 STUDENT NAME
STUDENTLNAME		✓	2STUDENTLNAME_DE	CHAR	20		0 STUDENT LASTNAME
STUDENTSCORE		✓	2STUDENT_SCORE_DE	DEC	6		3 STUDENT SCORE

MAIN SCREEN

UNIVERSITY GRADE SYSTEM

STUDENT INFO

NAME :
LAST NAME :
MAIL AD. :
BIRTH DATE :
SCORE :
BEHAVIOUR :
GENDER : ☒ M ☐ F

UPDATE STUDENT
STUDENT ID :

SEARCH STUDENT

NAME :
LAST NAME :
STUDENT ID :

FILTER SCORES
 -

ADOBE FORM

HST

STUDENT REPORT

ID
NAME
LAST NAME
GENDER
BIRTH DATE
AGE
MAIL ADDRESS
SCORE
BEHAVIOUR

??

DATE
TIME

1234567890

OOALV VIEW

ID	NAME	LNAME	GEN	BIRTH DATE	AGE	MAIL ADDRESS	SCORE	GRADE	BHV
1	HASAN SELIM	TOK	M	20.07.2005	20	HSELIMT@GMAIL.COM	99,999	AA	GREAT
2	AYSE	ASLAN	F	15.06.2004	21	AYSEASLAN@GMAIL.COM	78,200	BC	GOOD
5	AHMET	KOC	M	20.03.2000	25	AHMETKOC@HOTMAIL.COM	77,777	BC	BAD
3	MUSTAFA	AK	M	12.08.2003	21	MUSTAFAAK@GMAIL.COM	63,000	CC	AVERAGE
6	MEHMET	TURK	M	15.06.2004	21	MEHMETURK@GMAIL.COM	48,000	CD	AVERAGE
4	ZEYNEP	NAZ	F	14.07.2004	21	ZEYNEPN@HOTMAIL.COM	32,777	FF	POOR

SALV VIEW

STUDENT SALV VIEW

ID	NAME	LNAME	GEN	BIRTH DATE	AGE	MAIL ADDRESS	SCORE	GRADE	BHV
1	HASAN SELIM	TOK	M	20.07.2005	20	HSELIMT@GMAIL.COM	99,999	AA	GREAT
2	AYSE	ASLAN	F	15.06.2004	21	AYSEASLAN@GMAIL.COM	78,200	BC	GOOD
5	AHMET	KOC	M	20.03.2000	25	AHMETKOC@HOTMAIL.COM	77,777	BC	BAD
3	MUSTAFA	AK	M	12.08.2003	21	MUSTAFAAK@GMAIL.COM	63,000	CC	AVERAGE
6	MEHMET	TURK	M	15.06.2004	21	MEHMETURK@GMAIL.COM	48,000	CD	AVERAGE
4	ZEYNEP	NAZ	F	14.07.2004	21	ZEYNEPN@HOTMAIL.COM	32,777	FF	POOR

MAIN

19 August 2025 14:17

REPORT zunigradesystem.

```
INCLUDE <icon>.
INCLUDE zuni_grade_system_data.
INCLUDE zuni_grade_system_class.
INCLUDE zuni_grade_system_modules.
INCLUDE zuni_grade_system_forms.
```

```
START-OF-SELECTION.
  PERFORM initialize_data.
  sy-title = 'UNI GRADE SYSTEM'.
  CALL SCREEN 0001.
```

```
AT USER-COMMAND.
  CASE sy-ucomm.
    WHEN 'BACK'.
      LEAVE LIST-PROCESSING.
      LEAVE TO SCREEN 0001.

    WHEN 'EXIT'.
      LEAVE PROGRAM.
  ENDCASE.
```

DATA

19 August 2025 14:17

```
TYPES: BEGIN OF ty_cell_color,
        studentid      TYPE zstudentid_de,
        studentname     TYPE zstudentname_de,
        studentlname    TYPE zstudentlname_de,
        studentgen      TYPE zstudentgen_de,
        studentbdate    TYPE zstudentbdate_de,
        studentage      TYPE zstudentage_de,
        studentmail     TYPE zstudentmail_de,
        studentscore    TYPE zstudent_score_de,
        studentgrade    TYPE zstudent_grade_de,
        studentbhv      TYPE zstudentbhv_de,
        row_color       TYPE lvc_t_scol, " cell color information table
    END OF ty_cell_color.

DATA: gv_studentID      TYPE zstudentid_de,
      gv_studentName    TYPE zstudentname_de,
      gv_studentLName   TYPE zstudentlname_de,
      gv_studentGender  TYPE zstudentgen_de,
      gv_mrad           TYPE xfeld,
      gv_frad           TYPE xfeld,
      gv_studentBDate   TYPE zstudentbdate_de,
      gv_studentAge     TYPE zstudentage_de,
      gv_studentMail    TYPE zstudentmail_de,
      gv_studentScore   TYPE zstudent_score_de,
      gv_studentGrade   TYPE zstudent_grade_de,
      gv_studentBhv     TYPE zstudentbhv_de,
      gs_student_t      TYPE zstudent_t,
      gt_student_t      TYPE TABLE OF zstudent_t,
      gs_failed_t       TYPE zfstudent_t,
      gt_failed_t       TYPE TABLE OF zfstudent_t,
      gs_ostudent_t    TYPE zstudent_t,
      gt_ostudent_t    TYPE TABLE OF zstudent_t,
      go_salv_students  TYPE REF TO cl_salv_table,
      gs_cell_color     TYPE ty_cell_color, " global structure for cell
color
      gt_cell_color     TYPE TABLE OF ty_cell_color, " global table for
cell color
      go_oalv_students  TYPE REF TO cl_gui_alv_grid,
      go_cont           TYPE REF TO cl_gui_custom_container,
      ls_layout         TYPE lvc_s_layo, " ALV layout settings
      lt_sort           TYPE lvc_t_sort, " ALV sorting rules table
      lt_fieldcatalog   TYPE lvc_s_fcat, " single ALV column definition
      ls_fieldcatalog   TYPE lvc_s_fcat, " work area for field catalog
      lt_selected_rows  TYPE lvc_t_row, " table of selected row numbers
      lv_next_student_id TYPE zstudentid_de,
      lv_highest_score  TYPE zstudent_score_de VALUE 0,
      lv_new_age        TYPE zstudentage_de,
      ls_max_grade_student TYPE zstudent_t,
      p_name            TYPE zstudentname_de,
      p_lname           TYPE zstudentlname_de,
      p_bdate           TYPE zstudentbdate_de,
      p_score           TYPE zstudent_score_de,
      p_mail            TYPE zstudentmail_de,
      p_male            TYPE c,
      p_female          TYPE c,
      P_bhv             TYPE zstudentbhv_de,
      p_sname           TYPE zstudentname_de,
      p_slname          TYPE zstudentlname_de,
      p_id              TYPE zstudentid_de,
      p_uid             TYPE zstudentid_de,
      p_score1          TYPE i,
```

```

        p_score2          TYPE i,
        recipients        TYPE somlrecil OCCURS 0 WITH HEADER LINE, " who
gets mail
* grows as needed and creates work area with the same name
        object_hd         TYPE solistil OCCURS 0 WITH HEADER LINE, " mail
text
        doc_chng          TYPE sodocchg1, " mail properties
        mail_msg           TYPE c LENGTH 100,
        lv_mail_adress     TYPE c LENGTH 30,
        gs_outparams       TYPE sfpoutputparams,
        gv_name            TYPE fpname,
        gv_funcname        TYPE funcname,
        gs_docparams       TYPE sfpdocparams,
        gs_formoutput      TYPE fpformoutput.

```

MODULE

19 August 2025 14:17

```
MODULE status_0001 OUTPUT.  
  SET PF-STATUS '0001'.  
  SET TITLEBAR 'SALV VIEW'.  
ENDMODULE.
```

```
MODULE user_command_0001 INPUT.  
  CASE sy-ucomm.  
    WHEN 'ADD'.  
      IF p_name IS NOT INITIAL AND p_score >= 0 AND p_score <= 100.  
        PERFORM process_student_data.  
        PERFORM handle_failed_students.  
        PERFORM calculate_next_student_id.  
        PERFORM display_added_student_details.  
        LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0001.  
      CLEAR: p_name, p_lname, p_bdate, p_score, p_mail, p_bhv, gv_mrad, g  
v_frad.  
    ELSE.  
      MESSAGE 'FILL REQUIRED FIELDS' TYPE 'E'.  
    ENDIF.  
  
    WHEN 'UPD'.  
      IF p_uid IS NOT INITIAL.  
        PERFORM update_student_by_id.  
        CLEAR: p_uid, p_name, p_lname, p_bdate, p_score, p_mail, p_bhv, gv_  
mrad, gv_frad.  
      ELSE.  
        MESSAGE 'ENTER STUDENT ID' TYPE 'E'.  
      ENDIF.  
  
    WHEN 'SEARCH'.  
      IF p_id IS NOT INITIAL.  
        PERFORM search_student_by_id.  
        LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0001.  
      CLEAR: p_id, p_sname, p_slname.  
    ELSEIF p_sname IS NOT INITIAL AND p_slname IS NOT INITIAL.  
      PERFORM search_student_by_name.  
      LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0001.  
    ELSE.  
      MESSAGE 'FILL REQUIRED FIELDS' TYPE 'E'.  
    ENDIF.  
  
    WHEN 'SENDMAIL'.  
      IF p_id IS NOT INITIAL.  
        PERFORM send_mail_to_student.  
      ELSE.  
        PERFORM send_mail_to_students.  
      ENDIF.  
      CLEAR: p_id.  
  
    WHEN 'FORM'.  
      IF p_id IS NOT INITIAL.  
        PERFORM prepare_form.  
        PERFORM create_form.  
        CLEAR p_id.  
      ELSE.  
        MESSAGE 'FILL REQUIRED FIELDS' TYPE 'E'.  
      ENDIF.  
  
    WHEN 'FILTER'.  
      IF p_score1 IS NOT INITIAL AND p_score2 IS NOT INITIAL AND p_score1 =
```

```

> 0 AND p_score2 <= 100 AND p_score1 < p_score2.
    PERFORM list_students_by_score.
    LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0001.
ELSE.
    MESSAGE 'ENTER VALID VALUES' TYPE 'E'.
ENDIF.
CLEAR: p_score1, p_score2.

WHEN 'DEL'.
    PERFORM clear_all_data_with_popup.

WHEN 'STATS'.
    PERFORM display_statistics.
    LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0001.

WHEN 'SSTUDENTS'.
    PERFORM display_top_student.
    LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0001.

WHEN 'FSTUDENTS'.
    PERFORM display_failed_students.
    LEAVE TO LIST-PROCESSING AND RETURN TO SCREEN 0001.

WHEN 'SALV'.
    PERFORM set_cell_color.
    PERFORM create_student_salv.

WHEN 'OOALV'.
    PERFORM create_student_oalv.
    CALL SCREEN 0002.

WHEN 'EXIT'.
    IF go_oalv_students IS BOUND.
        CALL METHOD go_oalv_students->free.
        CLEAR go_oalv_students.
    ENDIF.
    IF go_cont IS BOUND.
        CALL METHOD go_cont->free.
        CLEAR go_cont.
    ENDIF.
    LEAVE PROGRAM.

ENDCASE.
ENDMODULE.

MODULE status_0002 OUTPUT.
    SET PF-STATUS '0002'.
    SET TITLEBAR 'OO ALV VIEW'.
ENDMODULE.

MODULE user_command_0002 INPUT.
    CASE sy-ucomm.
        WHEN 'SAVE'.
            PERFORM save_changes_from_bhv.
        WHEN 'BACK'.
            IF go_oalv_students IS BOUND.
                CALL METHOD go_oalv_students->free.
                CLEAR go_oalv_students.
            ENDIF.
            IF go_cont IS BOUND.
                CALL METHOD go_cont->free.
                CLEAR go_cont.
            ENDIF.
            LEAVE TO SCREEN 0001.
    ENDCASE.

```

```
WHEN 'EXIT'.
  IF go_oalv_students IS BOUND.
    CALL METHOD go_oalv_students->free.
    CLEAR go_oalv_students.
  ENDIF.
  IF go_cont IS BOUND.
    CALL METHOD go_cont->free.
    CLEAR go_cont.
  ENDIF.
  LEAVE PROGRAM.

ENDCASE.
ENDMODULE.
```


CLASS-METHOD

19 August 2025 14:18

```
CLASS lcl_grade_converter DEFINITION.  
  PUBLIC SECTION.  
    CLASS-METHODS:  
      convertscoretograde  
        IMPORTING  
          iv_score          TYPE zstudent_score_de  
        RETURNING  
          VALUE(rv_grade) TYPE zstudent_grade_de.  
ENDCLASS.
```

```
CLASS lcl_grade_converter IMPLEMENTATION.  
  METHOD convertscoretograde.  
    IF iv_score >= 95.  
      rv_grade = 'AA'.  
    ELSEIF iv_score >= 90.  
      rv_grade = 'AB'.  
    ELSEIF iv_score >= 85.  
      rv_grade = 'BB'.  
    ELSEIF iv_score >= 75.  
      rv_grade = 'BC'.  
    ELSEIF iv_score >= 55.  
      rv_grade = 'CC'.  
    ELSEIF iv_score >= 45.  
      rv_grade = 'CD'.  
    ELSEIF iv_score >= 35.  
      rv_grade = 'DD'.  
    ELSE.  
      rv_grade = 'FF'.  
    ENDIF.  
  ENDMETHOD.  
ENDCLASS.
```

```
CLASS lcl_age_calculator DEFINITION.  
  PUBLIC SECTION.  
    CLASS-METHODS:  
      calculate_age  
        IMPORTING  
          iv_bdate          TYPE zstudentbdate_de  
        RETURNING  
          VALUE(rv_age) TYPE zstudentage_de.  
ENDCLASS.
```

```
CLASS lcl_age_calculator IMPLEMENTATION.  
  METHOD calculate_age.  
    rv_age = sy-datum(4) - iv_bdate(4).  
  
    IF sy-datum+4(4) < iv_bdate+4(4).  
      rv_age = rv_age - 1.  
    ENDIF.  
  ENDMETHOD.  
ENDCLASS.
```

```
CLASS lcl_set_cell_color DEFINITION.  
  PUBLIC SECTION.  
    CLASS-METHODS:  
      set_cell_color  
        IMPORTING  
          iv_grade          TYPE zstudent_grade_de  
        RETURNING  
          VALUE(rv_color) TYPE lvc_s_scol.
```

```

ENDCLASS.

CLASS lcl_set_cell_color IMPLEMENTATION.
METHOD set_cell_color.
    CLEAR rv_color.
    rv_color-fname = 'STUDENTGRADE'.
    rv_color-color-int = '1'.

    CASE iv_grade.
        WHEN 'FF'.
            rv_color-color-col = col_negative.
        WHEN 'DD' OR 'CD'.
            rv_color-color-col = col_total.
        WHEN 'CC' OR 'BC' OR 'BB'.
            rv_color-color-col = col_key.
        WHEN 'AB' OR 'AA'.
            rv_color-color-col = col_positive.
    ENDCASE.
ENDMETHOD.
ENDCLASS.

```

FORM

19 August 2025 14:18

```
FORM initialize_data.
  PERFORM load_student_data.
  PERFORM load_failed_student_data.
  PERFORM calculate_next_student_id.
  PERFORM update_age_of_all_students.
ENDFORM.

FORM load_student_data.
  SELECT * FROM zstudent_t
  INTO CORRESPONDING FIELDS OF TABLE @gt_student_t.
ENDFORM.

FORM load_failed_student_data.
  SELECT * FROM zfstudent_t
  INTO CORRESPONDING FIELDS OF TABLE @gt_failed_t.
ENDFORM.

FORM process_student_data.
  PERFORM prepare_student_record.
  PERFORM save_student_record.
ENDFORM.

FORM update_age_of_all_students.
  SELECT * FROM zstudent_t
  INTO TABLE gt_student_t.

  LOOP AT gt_student_t INTO gs_student_t.
    lv_new_age = lcl_age_calculator=>calculate_age( gs_student_t-
studentbdate ).
    IF lv_new_age NE gs_student_t-studentage.
      gs_student_t-studentage = lv_new_age.

      UPDATE zstudent_t SET studentage = lv_new_age
      WHERE studentid = gs_student_t-studentid.
    ENDIF.
  ENDLOOP.

  COMMIT WORK.
ENDFORM.

FORM prepare_student_record.
  CLEAR gs_student_t.

  gs_student_t-studentid = lv_next_student_id.
  gs_student_t-studentname = p_name.
  gs_student_t-studentlname = p_lname.
  gs_student_t-studentbdate = p_bdate.
  gs_student_t-studentmail = p_mail.
  gs_student_t-studentscore = p_score.
  gs_student_t-studentgrade = lcl_grade_converter=>
convertscoretgrade( p_score ).
  gs_student_t-studentage = lcl_age_calculator=>calculate_age( p_bdate ).
  gs_student_t-studentbhv = p_bhv.

  IF gv_mrad EQ 'X'.
    gs_student_t-studentgen = 'M'.
  ELSEIF gv_frad EQ 'X'.
    gs_student_t-studentgen = 'F'.
  ENDIF.
```

```

IF p_bdate <= sy-datum AND p_mail CS '@' AND p_mail CS '.'.
  APPEND gs_student_t TO gt_student_t.
ELSE.
  MESSAGE 'MAKE SURE TO ENTER VALID VALUES' TYPE 'E'.
ENDIF.
ENDFORM.

FORM calculate_next_student_id.
  SELECT MAX( studentID ) FROM zstudent_t INTO @lv_next_student_id.

  IF sy-subrc NE 0.
    lv_next_student_id = 1.
  ELSE.
    lv_next_student_id = lv_next_student_id + 1.
  ENDIF.
ENDFORM.

FORM save_student_record.
  INSERT zstudent_t FROM @gs_student_t.

  IF sy-subrc = 0.
    COMMIT WORK.
    MESSAGE 'STUDENT ADDED' TYPE 'S'.
  ELSE.
    MESSAGE 'ERROR ADDING STUDENT' TYPE 'E'.
  ENDIF.
ENDFORM.

FORM display_added_student_details.
  SET PF-STATUS '0001'.

  WRITE: /1 '@0V@', 'NEW STUDENT'.
  WRITE: /3 '├ STUDENT ID:      ', gs_student_t-
studentid COLOR COL_HEADING.
  WRITE: /3 '├ NAME:          ', gs_student_t-
studentname COLOR COL_NORMAL.
  WRITE: /3 '├ LASTNAME:       ', gs_student_t-
studentlname COLOR COL_NORMAL.
  WRITE: /3 '├ BIRTH DATE:     ', gs_student_t-
studentbdate COLOR COL_NORMAL.
  WRITE: /3 '├ AGE:           ', gs_student_t-
studentage COLOR COL_NORMAL.
  WRITE: /3 '├ MAIL:          ', gs_student_t-
studentmail COLOR COL_NORMAL.
  WRITE: /3 '├ GENDER:         ', gs_student_t-
studentgen COLOR COL_NORMAL.
  WRITE: /3 '├ SCORE:          ', gs_student_t-
studentscore COLOR COL_NORMAL.
  WRITE: /3 '├ GRADE:          ', gs_student_t-
studentgrade COLOR COL_NORMAL.
  WRITE: /3 '├ BEHAVIOUR:       ', gs_student_t-
studentbhv COLOR COL_NORMAL.
  WRITE: /.
ENDFORM.

FORM handle_failed_students.
  IF gs_student_t-studentgrade = 'FF'.
    PERFORM process_failed_student.
  ENDIF.
ENDFORM.

FORM process_failed_student.
  IF gs_student_t-studentscore BETWEEN 34 AND 35.

```

```

        PERFORM upgrade_student_grade.
        MESSAGE 'STUDENT'S GRADE UPGRADED TO DD WITH THE SCORE OF 35' TYPE 'I'
    .
    ELSE.
        PERFORM add_to_failed_students.
        MESSAGE 'STUDENT HAS FAILED THE CLASS AND ADDED TO FAILED LIST' TYPE 'I'
    .
    ENDIF.
ENDFORM.

FORM upgrade_student_grade.
    gs_student_t-studentgrade = 'DD'.
    gs_student_t-studentscore = 35.

    UPDATE zstudent_t SET studentgrade = @gs_student_t-studentgrade,
                        studentscore = @gs_student_t-studentscore
                        WHERE studentid = @gs_student_t-studentid.

    IF sy-subrc = 0.
        COMMIT WORK.
    ENDIF.
ENDFORM.

FORM add_to_failed_students.
    CLEAR gs_failed_t.
    MOVE-CORRESPONDING gs_student_t TO gs_failed_t.

    INSERT zfstudent_t FROM @gs_failed_t.
    IF sy-subrc = 0.
        COMMIT WORK.
    ENDIF.
ENDFORM.

FORM update_student_by_id.
    SELECT SINGLE * FROM zstudent_t
        INTO @gs_student_t
        WHERE studentid EQ @p_uid.

    IF sy-subrc = 0.
        IF p_name IS NOT INITIAL.
            gs_student_t-studentname = p_name.
        ENDIF.

        IF p_lname IS NOT INITIAL.
            gs_student_t-studentlname = p_lname.
        ENDIF.

        IF p_bdate IS NOT INITIAL AND p_bdate <= sy-datum.
            gs_student_t-studentbdate = p_bdate.
            gs_student_t-studentage = lcl_age_calculator=>
calculate_age( p_bdate ).
        ENDIF.

        IF p_score IS NOT INITIAL AND p_score >= 0 AND p_score <= 100.
            gs_student_t-studentscore = p_score.
            gs_student_t-studentgrade = lcl_grade_converter=>
convertscoretograde( p_score ).
            gs_failed_t-studentscore = p_score.

            PERFORM update_failed_student_table.
        ENDIF.

        IF p_mail IS NOT INITIAL AND p_mail CS '@' AND p_mail CS '.'.
            gs_student_t-studentmail = p_mail.
        ENDIF.

```

```

IF gv_mrad EQ 'X'.
    gs_student_t-studentgen = 'M'.
ELSEIF gv_frad EQ 'X'.
    gs_student_t-studentgen = 'F'.
ENDIF.

IF p_bhv IS NOT INITIAL.
    gs_student_t-studentbhv = p_bhv.
ENDIF.

UPDATE zstudent_t FROM gs_student_t.

IF sy-subrc = 0.
    COMMIT WORK.
    MESSAGE 'STUDENT UPDATED SUCCESSFULLY' TYPE 'S'.

    WRITE: /1 '@08@', 'SEARCHED STUDENT'.
    WRITE: /3 '├─ STUDENT ID:      ', gs_student_t-
studentid COLOR COL_HEADING.
    WRITE: /3 '├─ NAME:          ', gs_student_t-
studentname COLOR COL_NORMAL.
    WRITE: /3 '├─ LASTNAME:      ', gs_student_t-
studentlname COLOR COL_NORMAL.
    WRITE: /3 '├─ BIRTH DATE:    ', gs_student_t-
studentbdate COLOR COL_NORMAL.
    WRITE: /3 '├─ AGE:          ', gs_student_t-
studentage COLOR COL_NORMAL.
    WRITE: /3 '├─ MAIL:         ', gs_student_t-
studentmail COLOR COL_NORMAL.
    WRITE: /3 '├─ GENDER:       ', gs_student_t-
studentgen COLOR COL_NORMAL.
    WRITE: /3 '├─ SCORE:        ', gs_student_t-
studentscore COLOR COL_NORMAL.
    WRITE: /3 '├─ GRADE:        ', gs_student_t-
studentgrade COLOR COL_NORMAL.
    WRITE: /3 '├─ BEHAVIOUR:    ', gs_student_t-
studentbhv COLOR COL_NORMAL.
    WRITE: /.
ELSE.
    ROLLBACK WORK.
    MESSAGE 'ERROR UPDATING STUDENT' TYPE 'E'.
ENDIF.
ELSE.
    MESSAGE 'STUDENT NOT FOUND FOR UPDATE' TYPE 'E'.
ENDIF.
ENDFORM.

FORM update_failed_student_table.
    PERFORM handle_failed_students.

IF gs_student_t-studentgrade NE 'FF'.
    SELECT SINGLE * FROM zfstudent_t
        WHERE studentid = @p_uid
        INTO @gs_failed_t.

IF sy-subrc = 0.
    DELETE FROM zfstudent_t WHERE studentid = @p_uid.
    COMMIT WORK.
ENDIF.
ENDIF.
ENDFORM.

```

```
FORM search_student_by_id.  
SET PF-STATUS '0001'.
```

```
SELECT SINGLE * FROM zstudent_t  
INTO @gs_student_t  
WHERE studentid EQ @p_id.
```

```
IF sy-subrc = 0.  
WRITE: /1 '@08@', 'SEARCHED STUDENT'.  
WRITE: /3 '├ STUDENT ID:      ', gs_student_t-  
studentid COLOR COL_HEADING.  
WRITE: /3 '├ NAME:          ', gs_student_t-  
studentname COLOR COL_NORMAL.  
WRITE: /3 '├ LASTNAME:      ', gs_student_t-  
studentlname COLOR COL_NORMAL.  
WRITE: /3 '├ BIRTH DATE:    ', gs_student_t-  
studentbdate COLOR COL_NORMAL.  
WRITE: /3 '├ AGE:          ', gs_student_t-  
studentage COLOR COL_NORMAL.  
WRITE: /3 '├ MAIL:         ', gs_student_t-  
studentmail COLOR COL_NORMAL.  
WRITE: /3 '├ GENDER:       ', gs_student_t-  
studentgen COLOR COL_NORMAL.  
WRITE: /3 '├ SCORE:        ', gs_student_t-  
studentscore COLOR COL_NORMAL.  
WRITE: /3 '├ GRADE:        ', gs_student_t-  
studentgrade COLOR COL_NORMAL.  
WRITE: /3 '├ BEHAVIOUR:    ', gs_student_t-  
studentbhv COLOR COL_NORMAL.  
WRITE: /.  
ELSE.  
WRITE: /1 '@0A@', 'STUDENT NOT FOUND'.  
ENDIF.  
ENDFORM.
```

```
FORM search_student_by_name.  
SET PF-STATUS '0001'.
```

```
SELECT SINGLE * FROM zstudent_t  
INTO @gs_student_t  
WHERE studentname EQ @p_sname AND studentlname EQ @p_slname.
```

```
IF sy-subrc = 0.  
WRITE: /1 '@08@', 'SEARCHED STUDENT'.  
WRITE: /3 '├ STUDENT ID:      ', gs_student_t-  
studentid COLOR COL_HEADING.  
WRITE: /3 '├ NAME:          ', gs_student_t-  
studentname COLOR COL_NORMAL.  
WRITE: /3 '├ LASTNAME:      ', gs_student_t-  
studentlname COLOR COL_NORMAL.  
WRITE: /3 '├ BIRTH DATE:    ', gs_student_t-  
studentbdate COLOR COL_NORMAL.  
WRITE: /3 '├ AGE:          ', gs_student_t-  
studentage COLOR COL_NORMAL.  
WRITE: /3 '├ MAIL:         ', gs_student_t-  
studentmail COLOR COL_NORMAL.  
WRITE: /3 '├ GENDER:       ', gs_student_t-  
studentgen COLOR COL_NORMAL.  
WRITE: /3 '├ SCORE:        ', gs_student_t-  
studentscore COLOR COL_NORMAL.  
WRITE: /3 '├ GRADE:        ', gs_student_t-  
studentgrade COLOR COL_NORMAL.
```

```

        WRITE: /3 'L BEHAVIOUR:      ', gs_student_t-
studentbhv COLOR COL_NORMAL.
        WRITE: /.
    ELSE.
        WRITE: /1 '@0A@', 'STUDENT NOT FOUND'.
    ENDIF.
ENDFORM.

FORM send_mail_to_student.
    SELECT SINGLE * FROM zstudent_t
        INTO @gs_student_t
        WHERE studentid EQ @p_id.

    CLEAR: recipients, object_hd.
    REFRESH: recipients, object_hd.

    lv_mail_adress = gs_student_t-studentmail.

    IF gs_student_t-studentgrade NE 'FF'.
        mail_msg = 'CONGRATS YOU SUCCESSFULLY PASSED THE CLASS'.
    ELSE.
        mail_msg = 'WE ARE SORRY TO INFORM YOU THAT YOU FAILED THE CLASS'.
    ENDIF.

    doc_chng-obj_descr = 'UNIVERSITY CLASS REPORT'.

    object_hd-line = | { gs_student_t-studentname } { gs_student_t-
studentlname } |.
    APPEND object_hd.

    object_hd-line = mail_msg.
    APPEND object_hd.

    object_hd-line = | YOUR SCORE IS: { gs_student_t-studentscore } |.
    APPEND object_hd.

    recipients-rec_type = 'U'.
    recipients-receiver = lv_mail_adress.
    APPEND recipients.

    CALL FUNCTION 'SO_NEW_DOCUMENT_SEND_API1'
        EXPORTING
            document_data      = doc_chng
            document_type      = 'RAW'
            put_in_outbox      = 'X'
            commit_work        = 'X'
        TABLES
            object_content     = object_hd
            receivers          = recipients
        EXCEPTIONS
            too_many_receivers = 1
            document_not_sent  = 2
            OTHERS              = 99.
    IF sy-subrc = 0.
        MESSAGE 'MAIL HAS BEEN SENT' TYPE 'S'.
    ELSE.
        MESSAGE 'MAIL SENDING HAS FAILED' TYPE 'E'.
    ENDIF.
ENDFORM.

FORM send_mail_to_students.
    DATA: lt_students TYPE TABLE OF zstudent_t.

    SELECT * FROM zstudent_t
    INTO CORRESPONDING FIELDS OF TABLE @lt_students.

```



```

LOOP AT lt_students INTO gs_student_t.
  CLEAR: recipients, object_hd.
  REFRESH: recipients, object_hd.

  lv_mail_adress = gs_student_t-studentmail.

  IF gs_student_t-studentgrade NE 'FF'.
    mail_msg = 'CONGRATS YOU SUCCESSFULLY PASSED THE CLASS'.
  ELSE.
    mail_msg = 'WE ARE SORRY TO INFORM YOU THAT YOU FAILED THE CLASS'.
  ENDIF.

  doc_chng-obj_descr = 'UNIVERSITY CLASS REPORT'.

  object_hd-line = | { gs_student_t-studentname } { gs_student_t-
studentlname } |.
  APPEND object_hd.

  object_hd-line = mail_msg.
  APPEND object_hd.

  object_hd-line = | YOUR SCORE IS: { gs_student_t-studentscore } |.
  APPEND object_hd.

  recipients-rec_type = 'U'.
  recipients-receiver = lv_mail_adress.
  APPEND recipients.

  CALL FUNCTION 'SO_NEW_DOCUMENT_SEND_API1'
    EXPORTING
      document_data      = doc_chng
      document_type       = 'RAW'
      put_in_outbox       = 'X'
      commit_work         = ''
    TABLES
      object_content      = object_hd
      receivers            = recipients
    EXCEPTIONS
      too_many_receivers = 1
      document_not_sent  = 2
      OTHERS              = 99.
  ENDLOOP.
  COMMIT WORK.
  IF sy-subrc = 0.
    MESSAGE 'MAILS HAVE BEEN SENT' TYPE 'S'.
  ELSE.
    MESSAGE 'MAIL SENDING HAS FAILED' TYPE 'E'.
  ENDIF.
ENDFORM.

FORM prepare_form.
  SELECT SINGLE * FROM zstudent_t
    INTO CORRESPONDING FIELDS OF @gs_student_t
    WHERE studentid = @p_id.
ENDFORM.

FORM create_form.
  gv_name = 'ZUNI_GRADE_SYSTEM_AFORM'.

  CALL FUNCTION 'FP_FUNCTION_MODULE_NAME'
    EXPORTING
      i_name      = gv_name
    IMPORTING
      e_funcname = gv_funcname.

```

```

gs_outparams-nodialog = 'X'.
gs_outparams-preview = 'X'.

CALL FUNCTION 'FP_JOB_OPEN'
  CHANGING
    ie_outputparams = gs_outparams.

CALL FUNCTION gv_funcname
  EXPORTING
    /lbcdwb/docparams = gs_docparams
    iv_studentid      = gs_student_t-studentid
    iv_studentname     = gs_student_t-studentname
    iv_studentlname    = gs_student_t-studentlname
    iv_studentgen      = gs_student_t-studentgen
    iv_studentbdate    = gs_student_t-studentbdate
    iv_studentage      = gs_student_t-studentage
    iv_studentmail     = gs_student_t-studentmail
    iv_studentscore    = gs_student_t-studentscore
    iv_studentgrade    = gs_student_t-studentgrade
    iv_studentbhv      = gs_student_t-studentbhv
    iv_cdate           = sy-datum
    iv_ctime           = sy-uzeit
  IMPORTING
    /lbcdwb/formoutput = gs_formoutput.

CALL FUNCTION 'FP_JOB_CLOSE'.
ENDFORM.

FORM list_students_by_score.
  SET PF-STATUS '0001'.

  SELECT * FROM zstudent_t
    INTO CORRESPONDING FIELDS OF TABLE @gt_student_t
    WHERE studentscore >= @p_score1 AND studentscore <= @p_score2.

  WRITE: /1 '@08@', 'STUDENTS IN RANGE'.

  IF lines( gt_student_t ) > 0.
    LOOP AT gt_student_t INTO gs_student_t.
      WRITE: /3 '|', gs_student_t-studentname, gs_student_t-
studentlname, gs_student_t-studentscore.
    ENDLOOP.
  ELSE.
    WRITE: /3 '|', '@0A@', ' NO DATA FOUND'.
  ENDIF.
  WRITE: /.
ENDFORM.

FORM clear_all_data_with_popup.
  DATA lv_answer TYPE c.

  CALL FUNCTION 'POPUP_TO_CONFIRM'
    EXPORTING
      titlebar          = 'CONFIRM DATA DELETION'
      text_question     = 'YOU WANT TO DELETE ALL STUDENT DATA?'
      text_button_1     = 'YES'
      text_button_2     = 'NO'
      default_button    = '2'
      display_cancel_button = space
    IMPORTING
      answer            = lv_answer.

  IF lv_answer = '1'.
    DELETE FROM zstudent_t.

```

```

DELETE FROM zfstudent_t.
COMMIT WORK.

lv_next_student_id = 1.
CLEAR: gt_student_t, gt_failed_t.
MESSAGE 'ALL STUDENTS HAVE BEEN DELETED' TYPE 'S'.
ENDIF.
ENDFORM.

FORM display_statistics.
SET PF-STATUS '0001'.
DATA: lv_total TYPE i,
      lv_avg TYPE p DECIMALS 3,
      lv_passed TYPE i,
      lv_failed TYPE i.

SELECT COUNT(*) FROM zstudent_t INTO @lv_total.
SELECT AVG( studentscore ) FROM zstudent_t INTO @lv_avg.
SELECT COUNT(*) FROM zstudent_t INTO @lv_passed WHERE studentscore >= 35.
lv_failed = lv_total - lv_passed.

WRITE: /1 '@17@', 'STATS'.
WRITE: /3 '| STUDENTS: ', lv_total.
WRITE: /3 '| PASSED: ', lv_passed.
WRITE: /3 '| FAILED: ', lv_failed.
WRITE: /3 '| AVG SCORE: ', lv_avg.
WRITE: /.
ENDFORM.

FORM find_top_student.
CLEAR: ls_max_grade_student, lv_highest_score.

SELECT MAX( studentscore ) FROM zstudent_t INTO @lv_highest_score.
IF sy-subrc = 0 AND lv_highest_score IS NOT INITIAL.
SELECT SINGLE * FROM zstudent_t
INTO @ls_max_grade_student
WHERE studentscore EQ @lv_highest_score.
ENDIF.
ENDFORM.

FORM display_top_student.
SET PF-STATUS '0001'.

PERFORM find_top_student.

IF ls_max_grade_student IS NOT INITIAL.
WRITE: /1 '@10@', 'TOP STUDENT'.
WRITE: /3 '| FULLNAME: ', ls_max_grade_student-studentname.
WRITE: /3 '| NAME: ', ls_max_grade_student-studentlname.
WRITE: /3 '| ID: ', ls_max_grade_student-studentid.
WRITE: /3 '| SCORE: ', ls_max_grade_student-studentscore.
WRITE: /.
ELSE.
WRITE: /1 '| ', '@0A@', ' NO DATA FOUND'.
WRITE: /.
ENDIF.
ENDFORM.

FORM display_failed_students.
SET PF-STATUS '0001'.

DATA lv_count TYPE i.

```

```

WRITE: /1 '@17@', 'FAILED STUDENTS'.

SELECT * FROM zfstudent_t INTO TABLE @gt_failed_t.

IF lines( gt_failed_t ) > 0.
  LOOP AT gt_failed_t INTO gs_failed_t.
    WRITE: /3 |┌ { gs_failed_t-studentid } { gs_failed_t-
studentname } { gs_failed_t-studentlname } { gs_failed_t-studentscore }|.
  ENDLOOP.
ELSE.
  WRITE: /1 '└', '@0A@', ' NO DATA FOUND'.
ENDIF.
WRITE: /.
ENDFORM.

FORM create_student_salv.
  cl_salv_table=>factory(
    IMPORTING
      r_salv_table = go_salv_students
    CHANGING
      t_table      = gt_cell_color
  ).

DATA(lo_display) = go_salv_students->get_display_settings( ).
lo_display->set_list_header( value = 'STUDENT SALV VIEW' ).

DATA(lo_columns) = go_salv_students->get_columns( ).
lo_columns->set_color_column( 'ROW_COLOR' ).
lo_columns->set_optimize( value = 'X' ).

go_salv_students->set_screen_popup(
  EXPORTING
    start_column = 20
    end_column   = 90
    start_line   = 5
    end_line     = 15
  ).

DATA(lt_columns) = lo_columns->get( ).
LOOP AT lt_columns INTO DATA(ls_column).
  ls_column-r_column->set_alignment(
    value = if_salv_c_alignment=>centered
  ).
ENDLOOP.

DATA(lo_sorts) = go_salv_students->get_sorts( ).
lo_sorts->add_sort(
  EXPORTING
    columnname = 'STUDENTSCORE'
    sequence   = if_salv_c_sort=>sort_down
  ).

IF gt_cell_color IS NOT INITIAL.
  go_salv_students->display( ).
ELSE.
  MESSAGE 'ERROR DISPLAYING ALV SINCE ITS EMPTY' TYPE 'E'.
ENDIF.
ENDFORM.

FORM set_cell_color.
  SELECT * FROM zstudent_t
  INTO CORRESPONDING FIELDS OF TABLE @gt_cell_color.
  LOOP AT gt_cell_color INTO gs_cell_color.
  DATA: lt_color TYPE lvc_t_scol.
  CLEAR lt_color.

```

```

DATA(ls_color) = lcl_set_cell_color=>set_cell_color( gs_cell_color-
studentgrade ).

APPEND ls_color TO lt_color.
gs_cell_color-row_color = lt_color.
MODIFY gt_cell_color FROM gs_cell_color.
ENDLOOP.
ENDFORM.

FORM create_student_oalv.
CREATE OBJECT go_cont
EXPORTING
    container_name = 'CC_ALV'.

IF go_oalv_students IS NOT BOUND.
CREATE OBJECT go_oalv_students
EXPORTING
    i_parent = go_cont.
ENDIF.

PERFORM set_cell_color.

DATA: ls_layout          TYPE lvc_s_layo,
      lt_sort            TYPE lvc_t_sort,
      lt_fieldcatalog    TYPE lvc_t_fcat.

CALL FUNCTION 'LVC_FIELDCATALOG_MERGE'
EXPORTING
    i_structure_name = 'zstudent_t'
CHANGING
    ct_fieldcat      = lt_fieldcatalog.

READ TABLE lt_fieldcatalog INTO ls_fieldcatalog WITH KEY fieldname = 'STU
DENTBHV'.
IF sy-subrc = 0.
    ls_fieldcatalog-edit = 'X'.
    MODIFY lt_fieldcatalog FROM ls_fieldcatalog INDEX sy-tabix.
ENDIF.

LOOP AT lt_fieldcatalog INTO ls_fieldcatalog.
    ls_fieldcatalog-edit = 'X'.
    MODIFY lt_fieldcatalog FROM ls_fieldcatalog INDEX sy-tabix.
ENDLOOP.

LOOP AT lt_fieldcatalog ASSIGNING FIELD-SYMBOL(<fs_fcat>).
    <fs_fcat>-just = 'C'.
ENDLOOP.

ls_layout-ctab_fname = 'ROW_COLOR'.
ls_layout-cwidth_opt = 'X'.
ls_layout-col_opt = 'X'.

APPEND VALUE #( fieldname = 'STUDENTSCORE' down = 'X' ) TO lt_sort.

IF gt_cell_color IS NOT INITIAL.
CALL METHOD go_oalv_students->set_table_for_first_display
EXPORTING
    is_layout          = ls_layout
    i_structure_name = 'ZSTUDENT_T'
CHANGING
    it_outtab          = gt_cell_color
    it_fieldcatalog    = lt_fieldcatalog
    it_sort             = lt_sort.
ELSE.
    MESSAGE 'ERROR DISPLAYING ALV SINCE ITS EMPTY' TYPE 'E'.

```

```

ENDIF.
ENDFORM.

FORM save_changes_from_bhv.
  CALL METHOD go_oalv_students->check_changed_data.

  CALL METHOD go_oalv_students->get_selected_rows
    IMPORTING
      et_index_rows = lt_selected_rows.

  IF lines( lt_selected_rows ) > 0.
    READ TABLE gt_cell_color INTO gs_cell_color INDEX lt_selected_rows[ 1 ]
    -index.

    UPDATE zstudent_t SET studentbhv = @gs_cell_color-studentbhv,
                          studentname = @gs_cell_color-studentname,
                          studentlname = @gs_cell_color-studentlname,
                          studentbdate = @gs_cell_color-studentbdate,
                          studentmail = @gs_cell_color-studentmail,
                          studentscore = @gs_cell_color-studentscore
    WHERE studentid = @gs_cell_color-studentid.

    DATA(lv_studentg) = lcl_grade_converter=>
    convertscoretograde( iv_score = gs_cell_color-studentscore ).
    DATA(lv_studenta) = lcl_age_calculator=>
    calculate_age( iv_bdate = gs_cell_color-studentbdate ).

    UPDATE zstudent_t SET studentgrade = @lv_studentg,
                          studentage = @lv_studenta
    WHERE studentid = @gs_cell_color-studentid.

    IF sy-subrc = 0.
      COMMIT WORK.
      CALL METHOD go_oalv_students->refresh_table_display.
      MESSAGE 'DATA SAVED' TYPE 'S'.
    ELSE.
      MESSAGE 'ERROR SAVING DATA' TYPE 'E'.
    ENDIF.
  ELSE.
    MESSAGE 'SELECT A ROW' TYPE 'W'.
  ENDIF.
ENDFORM.

```