1 DEVELOPMENT

1.1 PAGE-CONTAINER STRUCTURE

LOGO	NAVIGATION MENU	LOGIN

SUB-NAVIGATION MENU (BY CATEGORY)

I-FRAME (ALL LINKS TARGET)

Figure 1.1 DISPLAY LAYOUT

1.2 NAVIGATION MENU
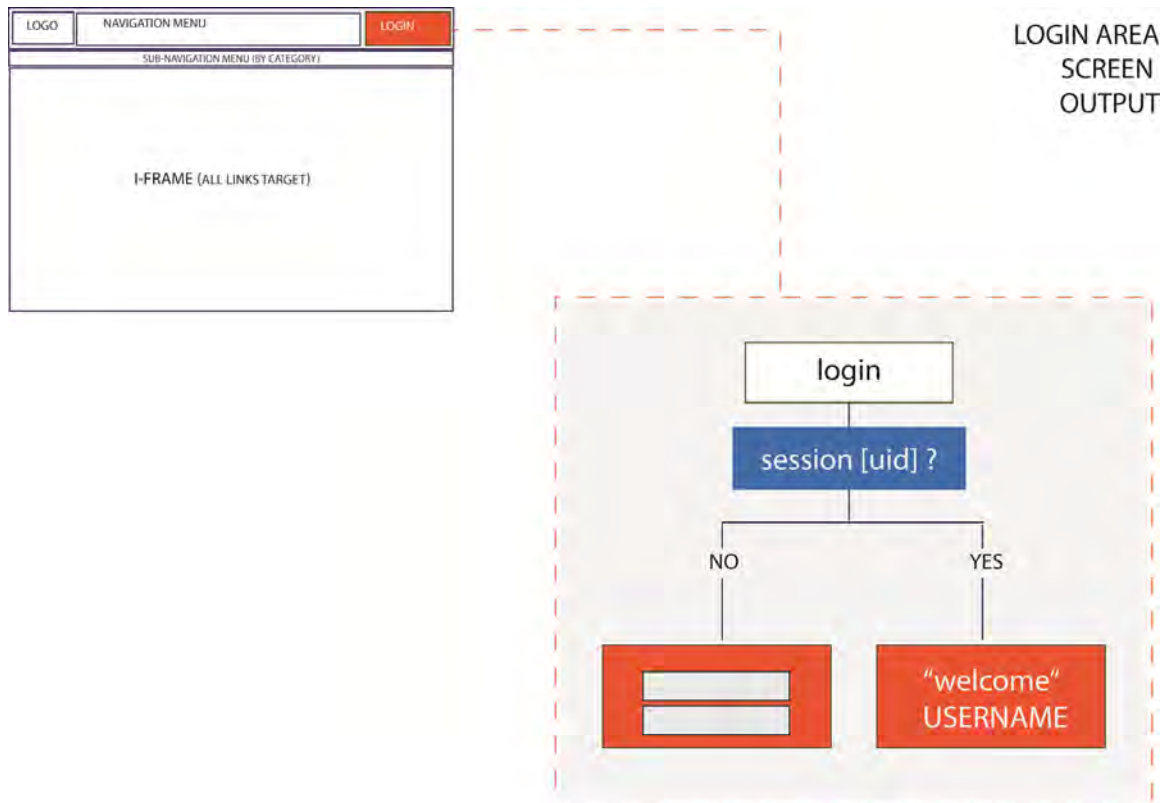
Figure 1.2 LOGIN ALTERNATIVE VIEWS

1.3   USER AUTHENTICATION

USER
AUTHENTICATION
PROCESS

HTML form   PHP   SERVER

username   LOGIN   FIND PASSWORD

password   OUTPUT   HASH   NOT FOUND   USERS
                                           password

                      if /else

HTML div
                                           PASSWORD FOUND
welcome   FIND USER NAME + USERID
"user-name"   OUTPUT   USER-NAME + USERID
                                           START SESSION [USERID]   SESSION [UID]
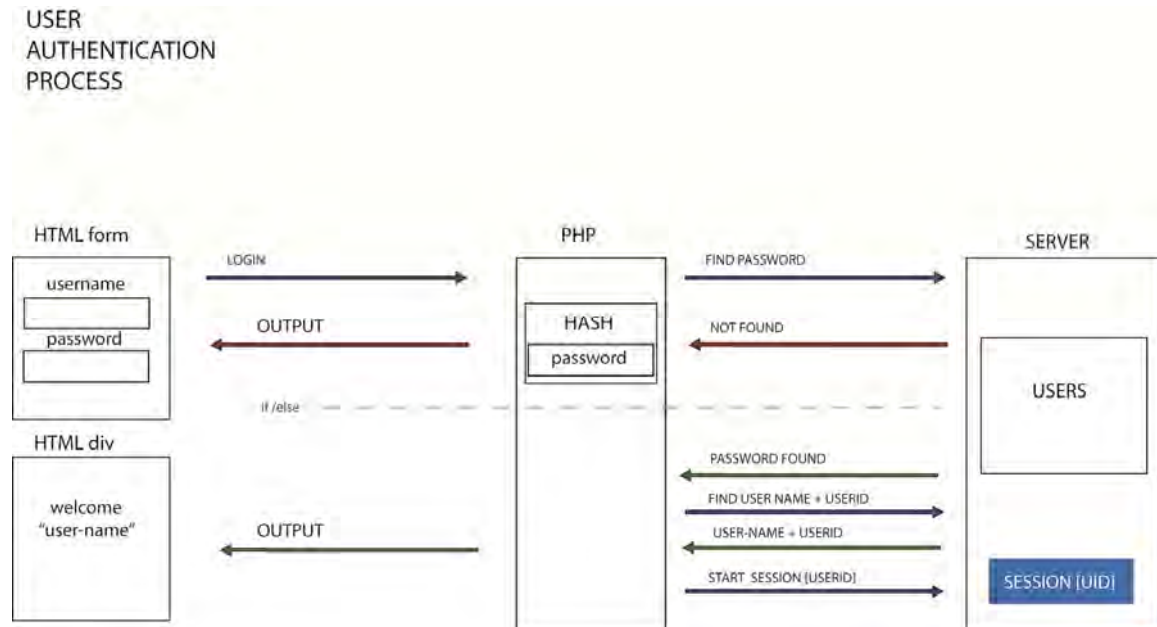
Figure 1.3 USER AUTHENTICATION PROCESS

1.4   MAIN MENU
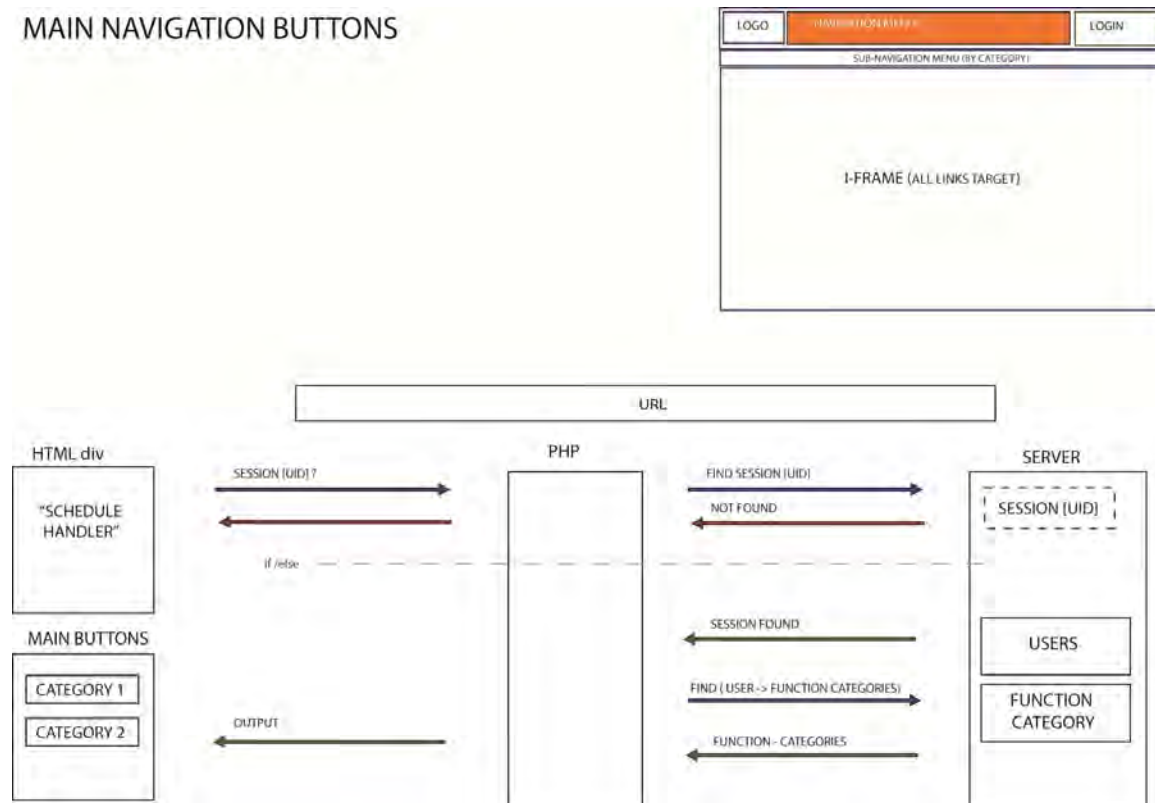

Figure 1.4 MAIN-NAVIGATION BUTTONS DESIGN
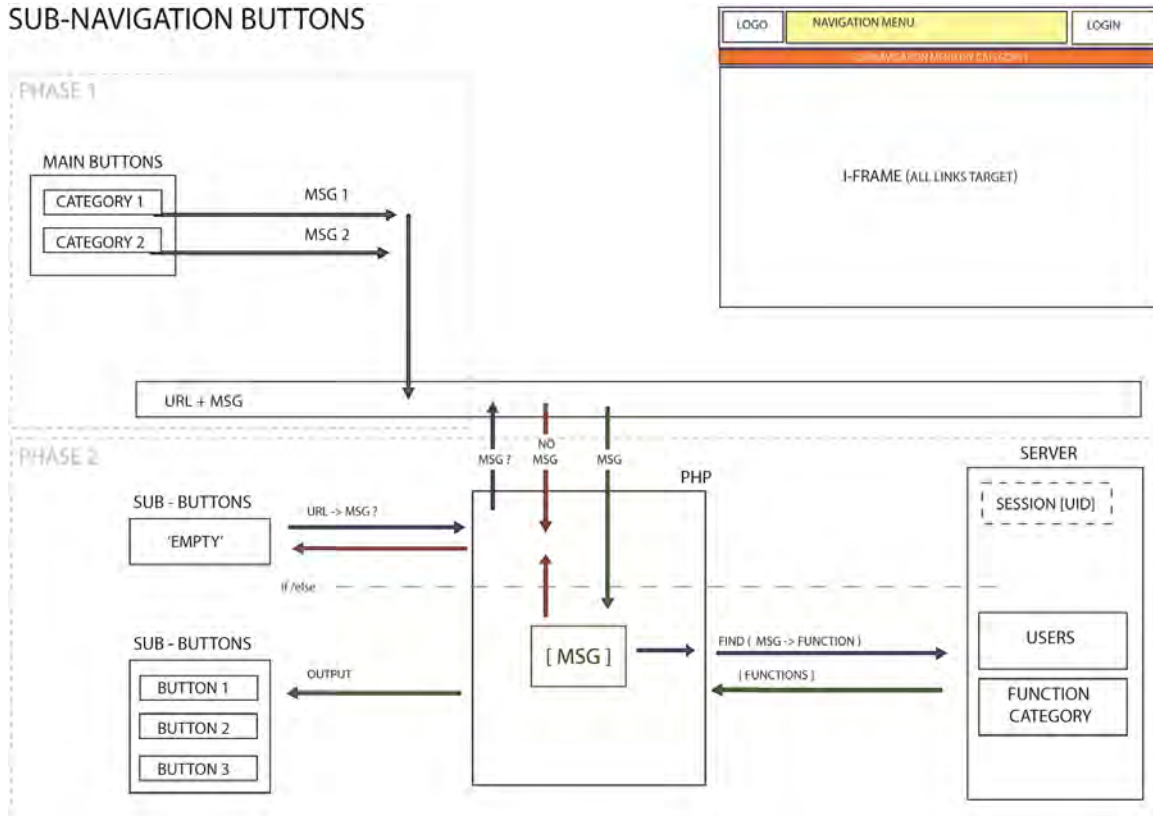
1.5 SUB-NAVIGATION MENU

Figure 1.5 SUB-NAVIGATION BUTTONS DESIGN

1.6   IFRAME - PAGES CONTAINER
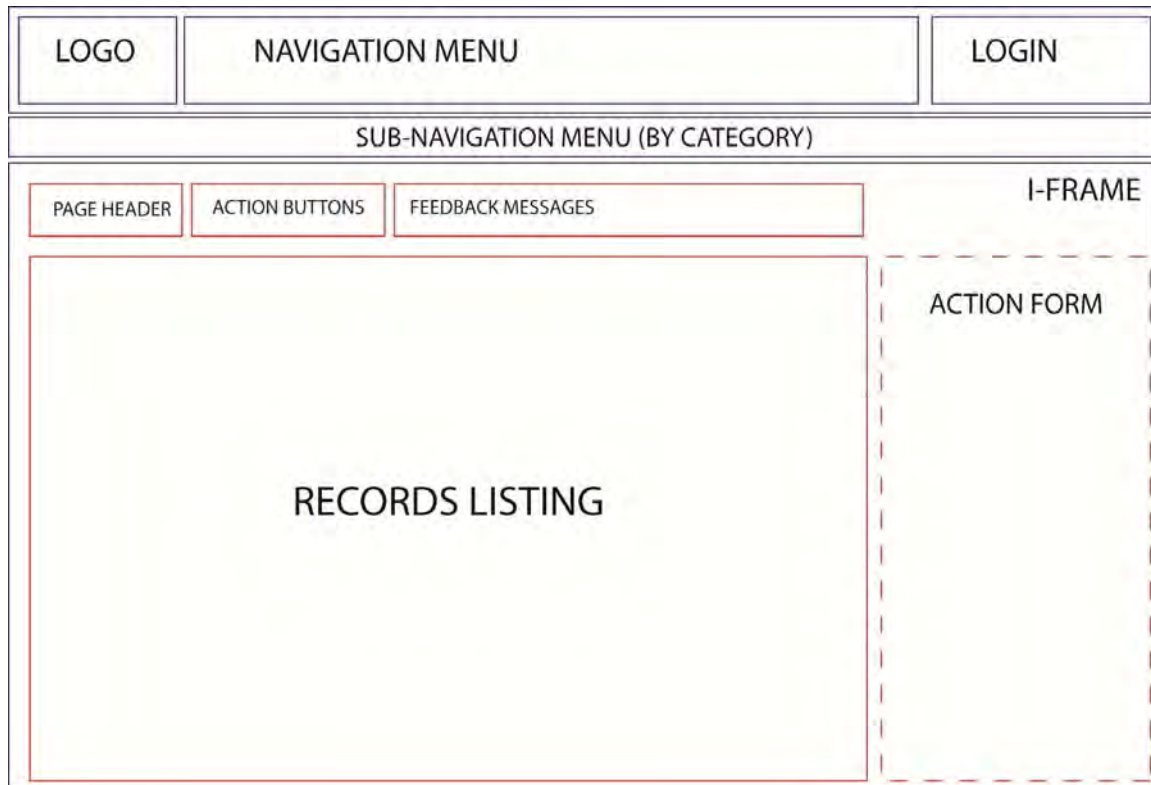

Figure 1.6 IFRAME CONTAINER

1.7   PAGE STRUCTURE

Figure 1.7 LOADED PAGES LAYOUT

1.8 PAGES GENERATION AND INTERACTION

1.8.1 ORIGINAL PAGES AND DERIVATIVES

|	html page	main navigation	sub-navigation
INTERFACE AUTO GENERATED	LIST RECORDS	FACILITIES	associations, campuses, class type, course, degree titles, instructions, semesters, study domain, time shifts, weekdays, years
		SYSTEM	available functions, function categories, name aliases, privileges, sql statements, system activity, table permissions, user logs, user roles, users
INTERFACE MANUAL SHARE INTERFACE	COURSES & INSTRUCTORS	SCHEDULE	course-instructor
	MASTER SCHEDULE	SCHEDULE	master schedule
	DAY SCHEDULE	SCHEDULE	day schedule
INTERFACE MANUAL	ESTIMATES	SCHEDULE	estimations
	BACKUP	SYSTEM	backup
	html page	table category	database table

Figure 1.8 MAIN PAGES OF THE SYSTEM

1.9   THE PAGE DESIGN
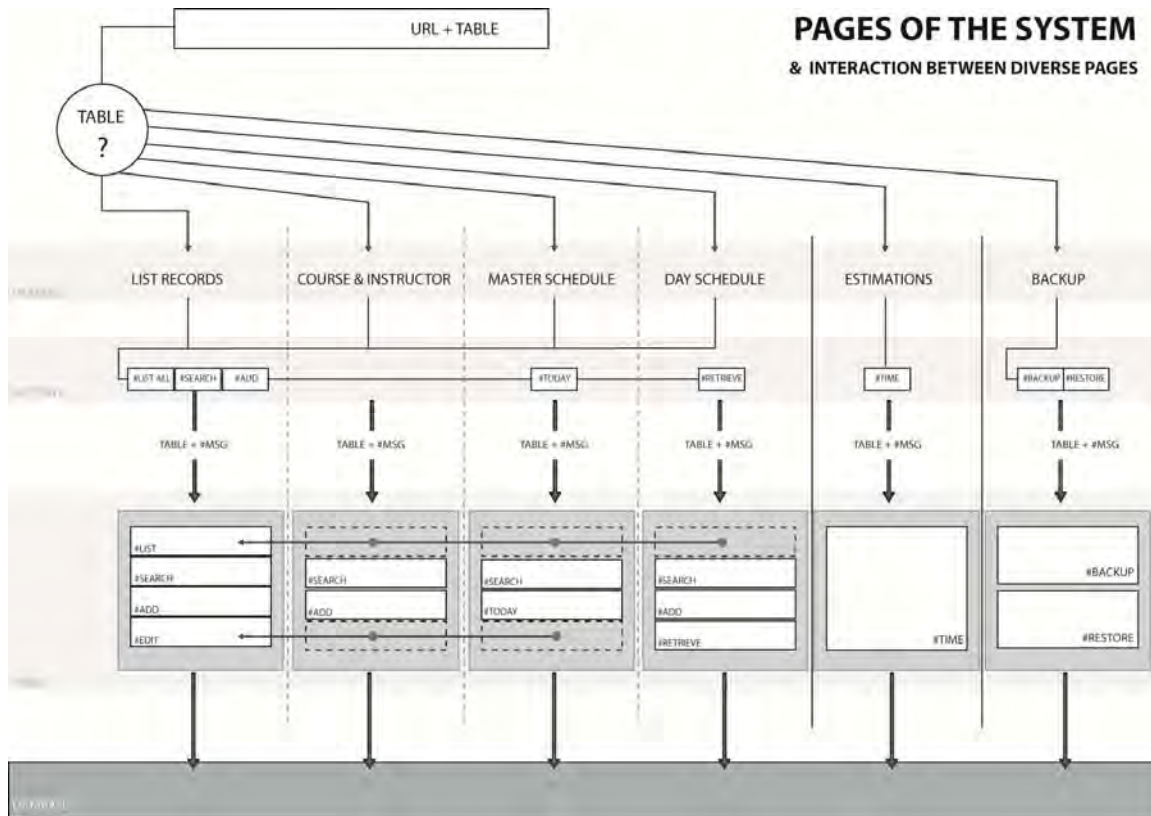
Figure 1.9 REAL-TIME PAGE DESIGN

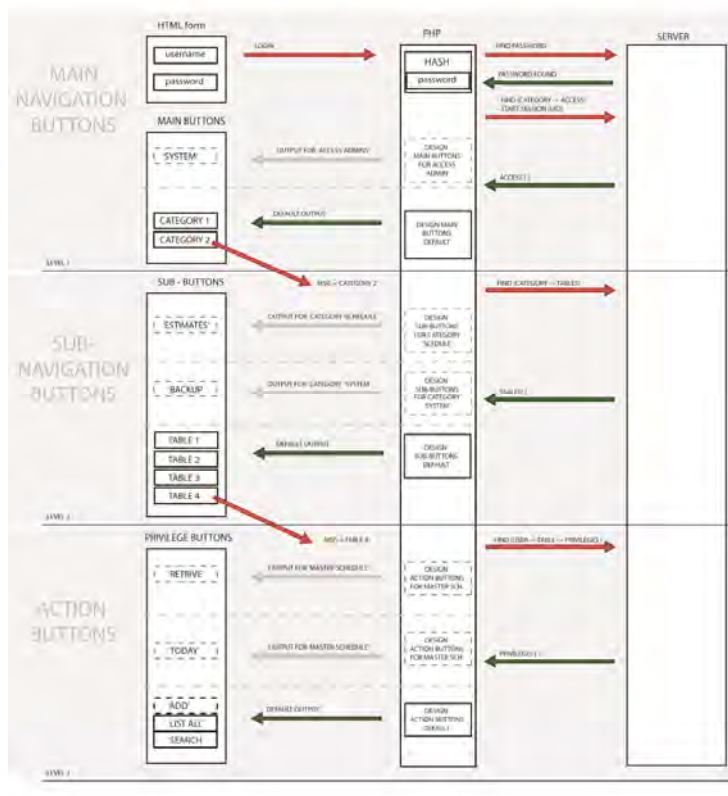# The sequence of system's buttons design

Figure 1.10 BUTTONS DESIGN PROCESS
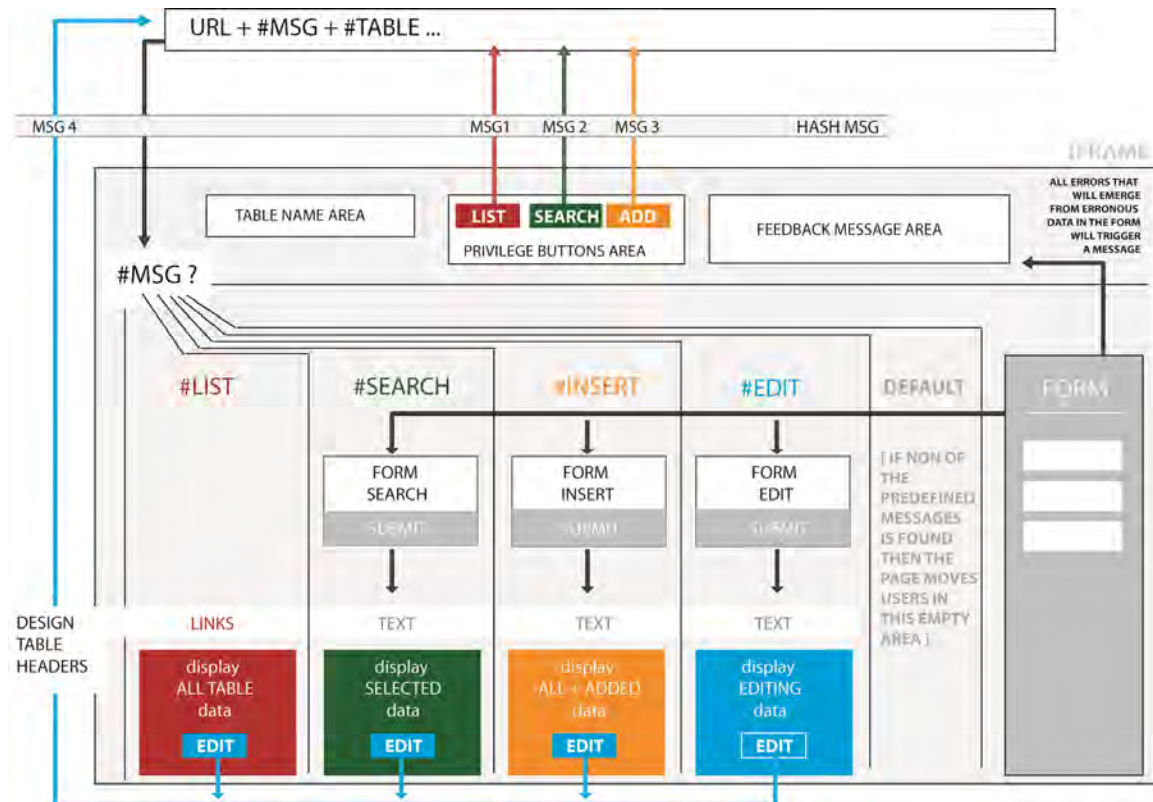
1.10 'ACTION' BUTTONS

Figure 1.11 BUTTONS DETERMINE PAGE DESIGN

1.11 FORM DESIGNERS

In this section we will examine the way that the script manages to design the required forms that mediate to the system's operation.
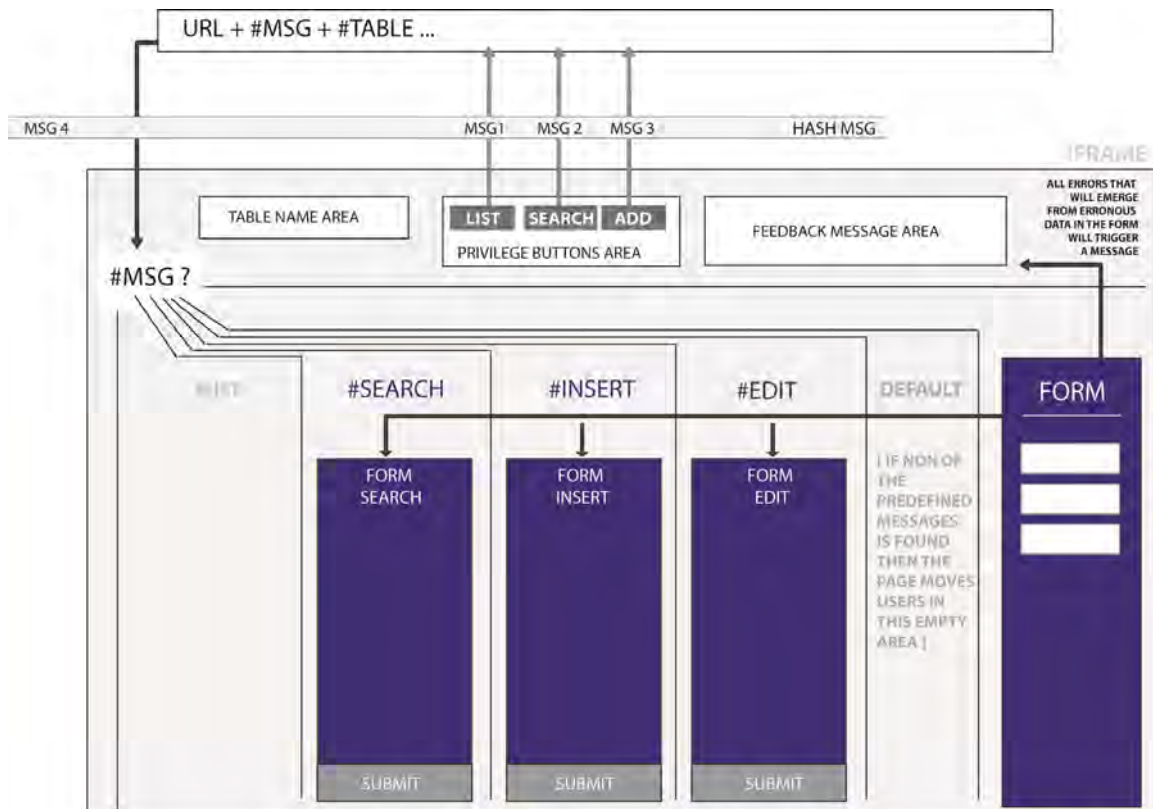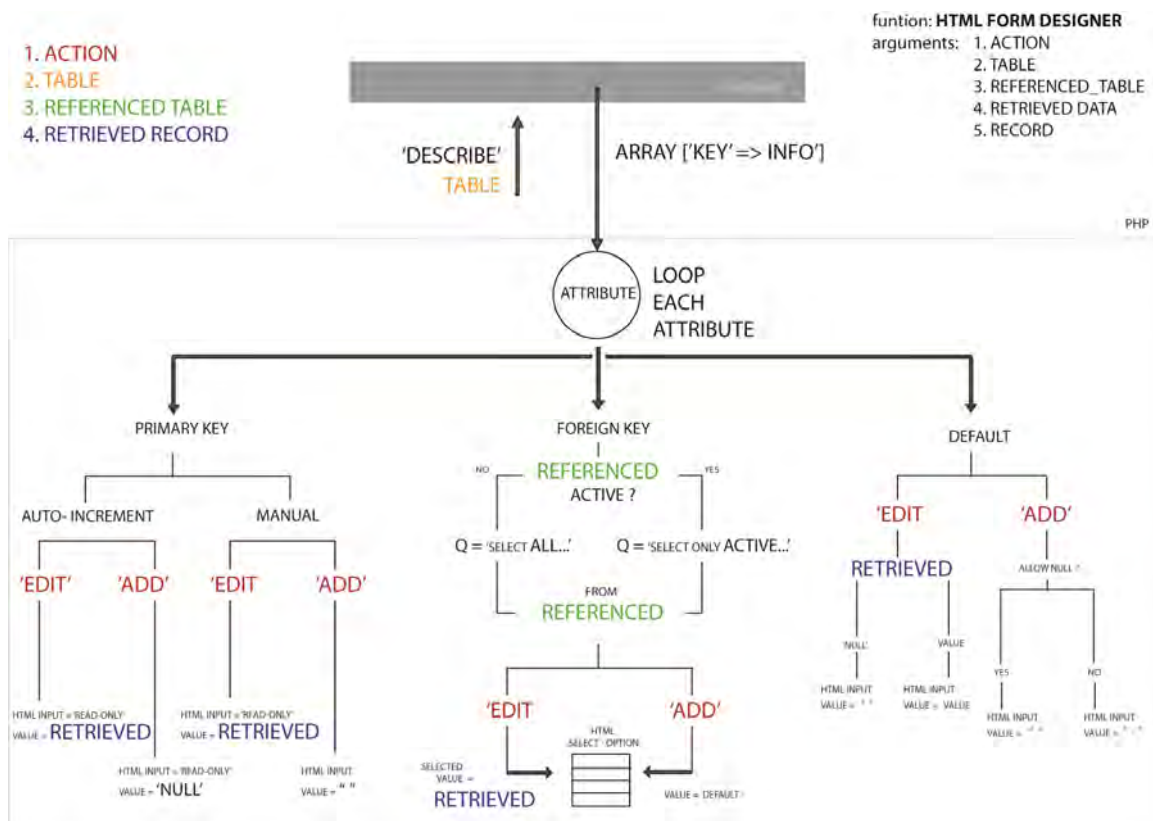
1.11.1 'HTML FORM DESIGNER'

Figure 1.13 FORM INPUTS DESIGN PROCESS
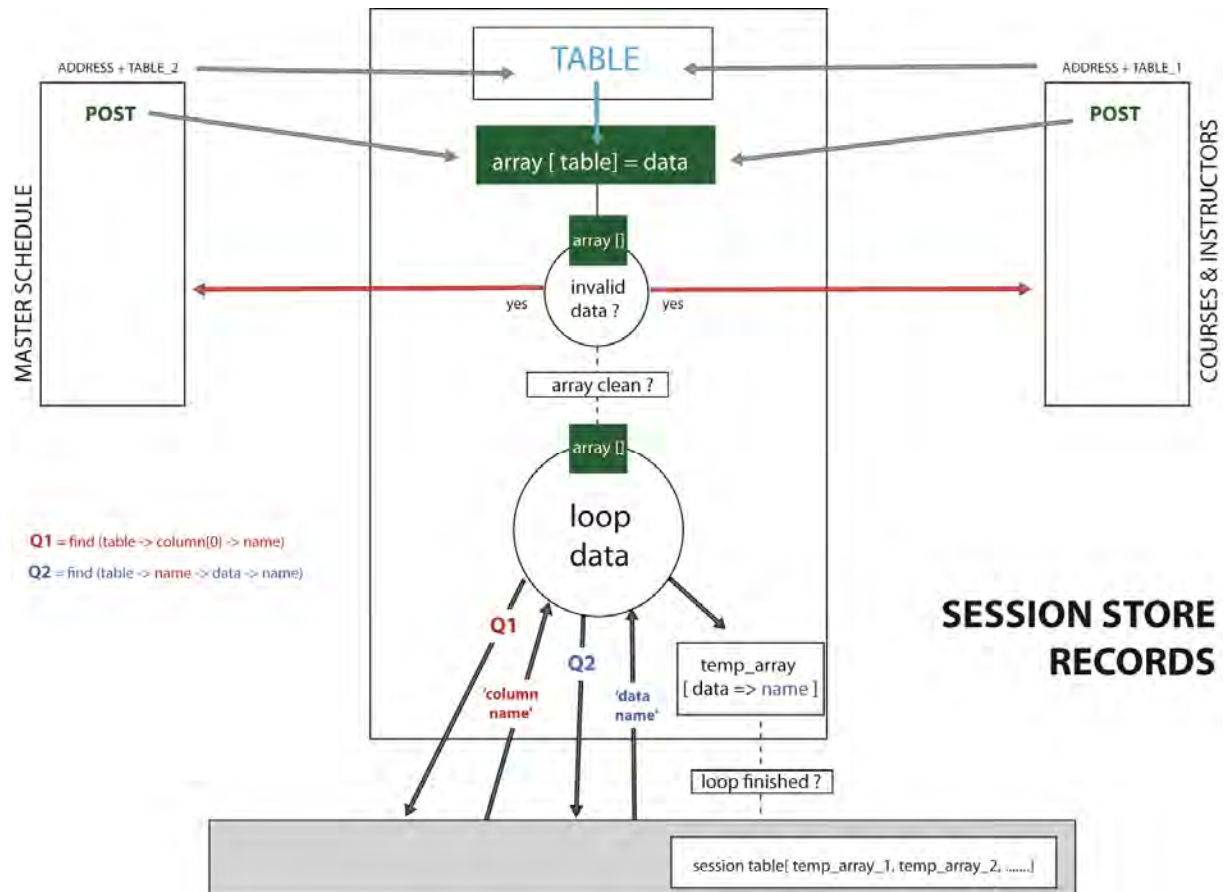
13

1.11.2 BUILD FILTERS IN A FORM (search/insert)

TABLE

ADDRESS + TABLE_2

ADDRESS + TABLE_1

MASTER SCHEDULE

COURSES & INSTRUCTORS

POST

POST

array [table] = data

array []

invalid data ?

yes

yes

array clean ?

array []

loop data

Q1 = find (table -> column(0) -> name)

Q2 = find (table -> name -> data -> name)

Q1

Q2

'column name'

'data name'

temp_array [data => name]

SESSION STORE RECORDS

loop finished ?

session table[temp_array_1, temp_array_2, ...]

Figure 1.14 MECHANISM THAT 'COURSE-INSTRUCTOR' & 'MASTER SCEDULE' USE TO STORE DATA TEMPORARILY BEFORE SAVED

1.12 VALIDATE DATA
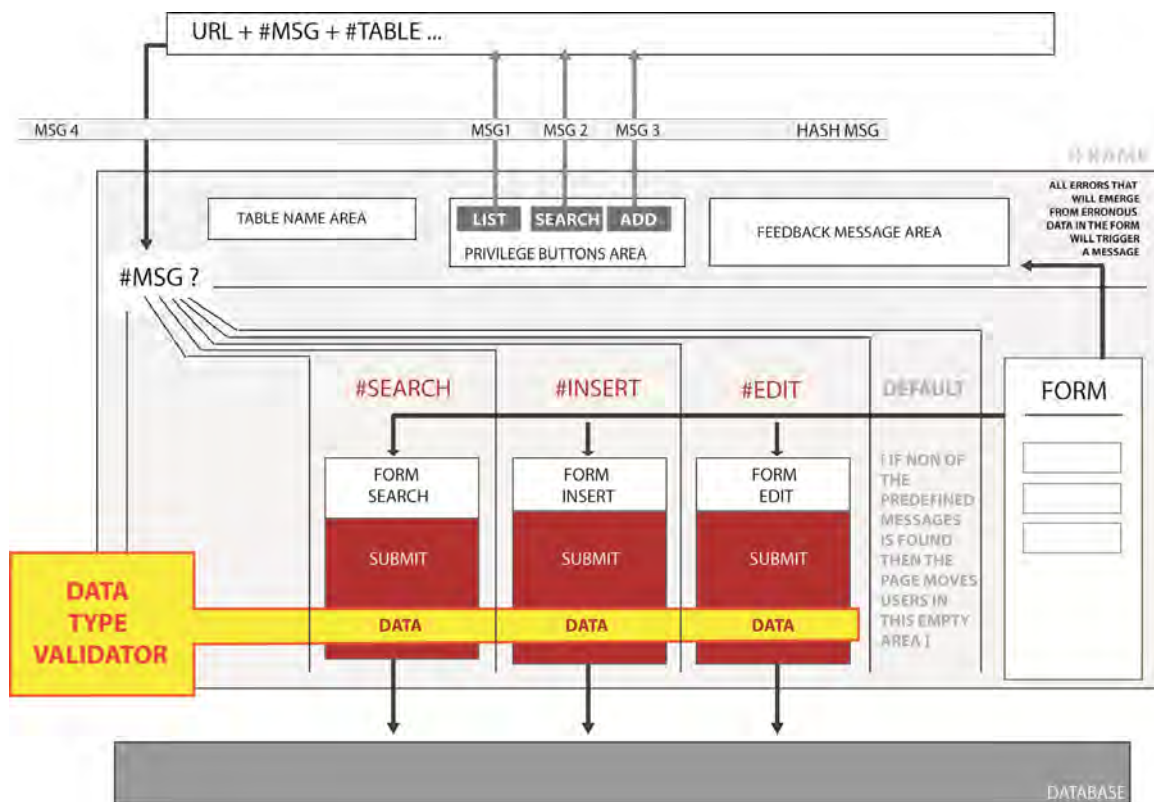
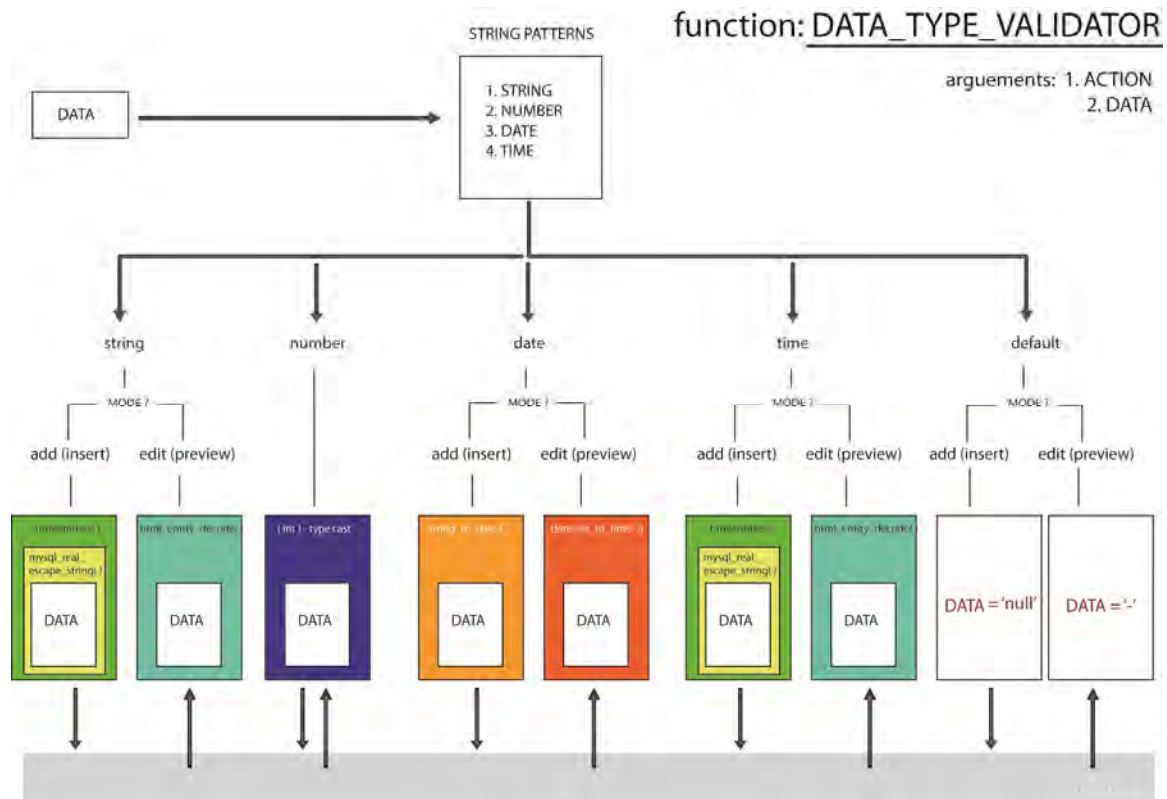Figure 1.15 SUBMITING DATA

DATA TYPE VALIDATOR

Figure 1.16 VALIDATE SUBMITED DATA

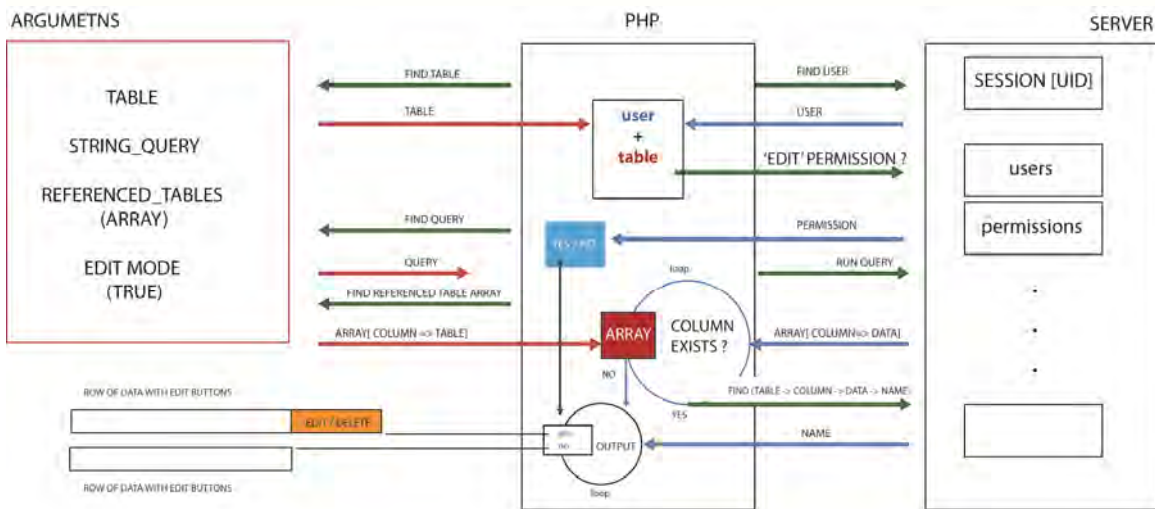## 1.13   LIST RECORDS AREA

Figure 1.17 DISPLAY DATA ON SCREEN PROCESS

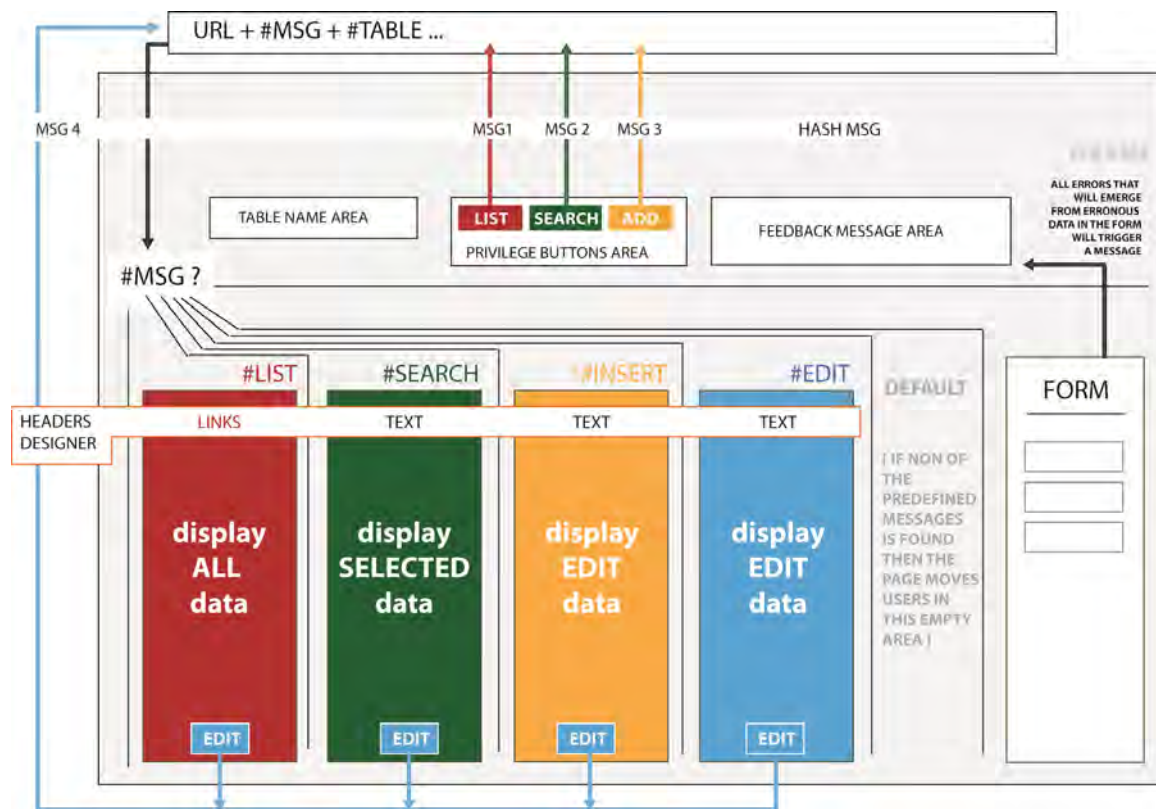# DISPLAY DATA FOR DIVERSE PAGES

Figure 1.18 DESIGN THE LISTING OF RECORDS
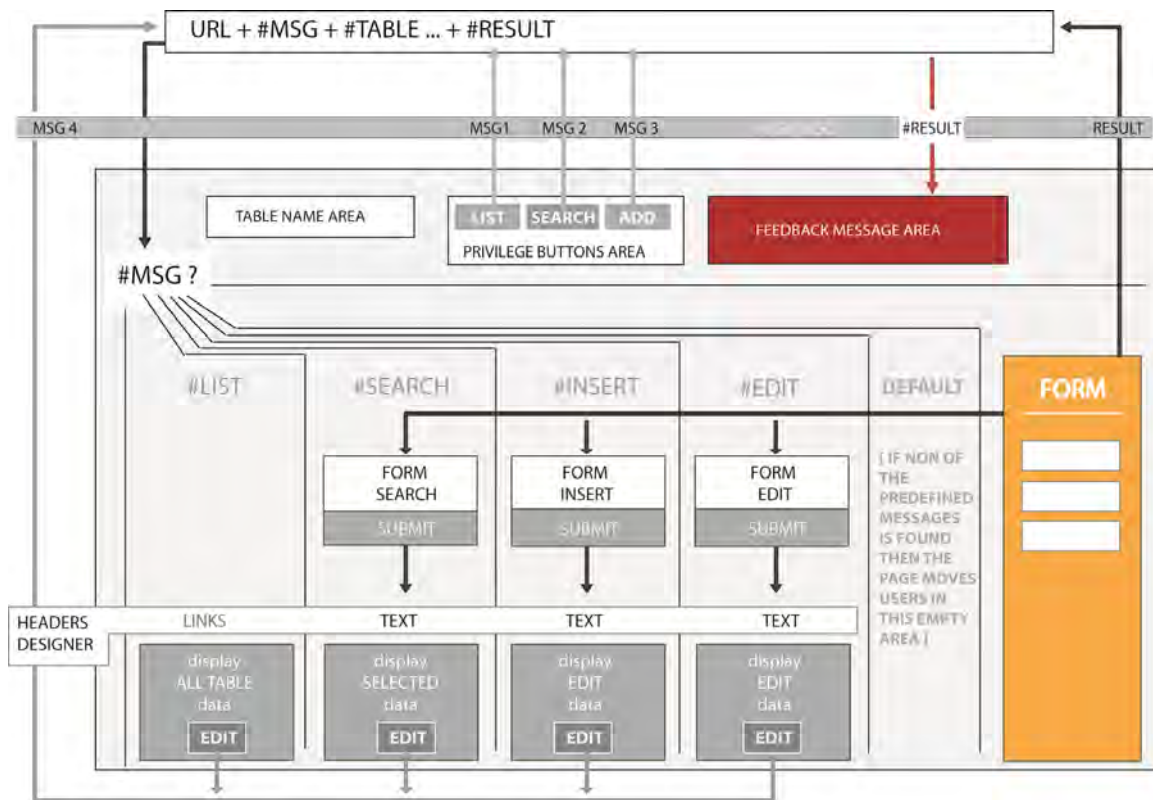
1.14 FEEDBACK MESSAGES
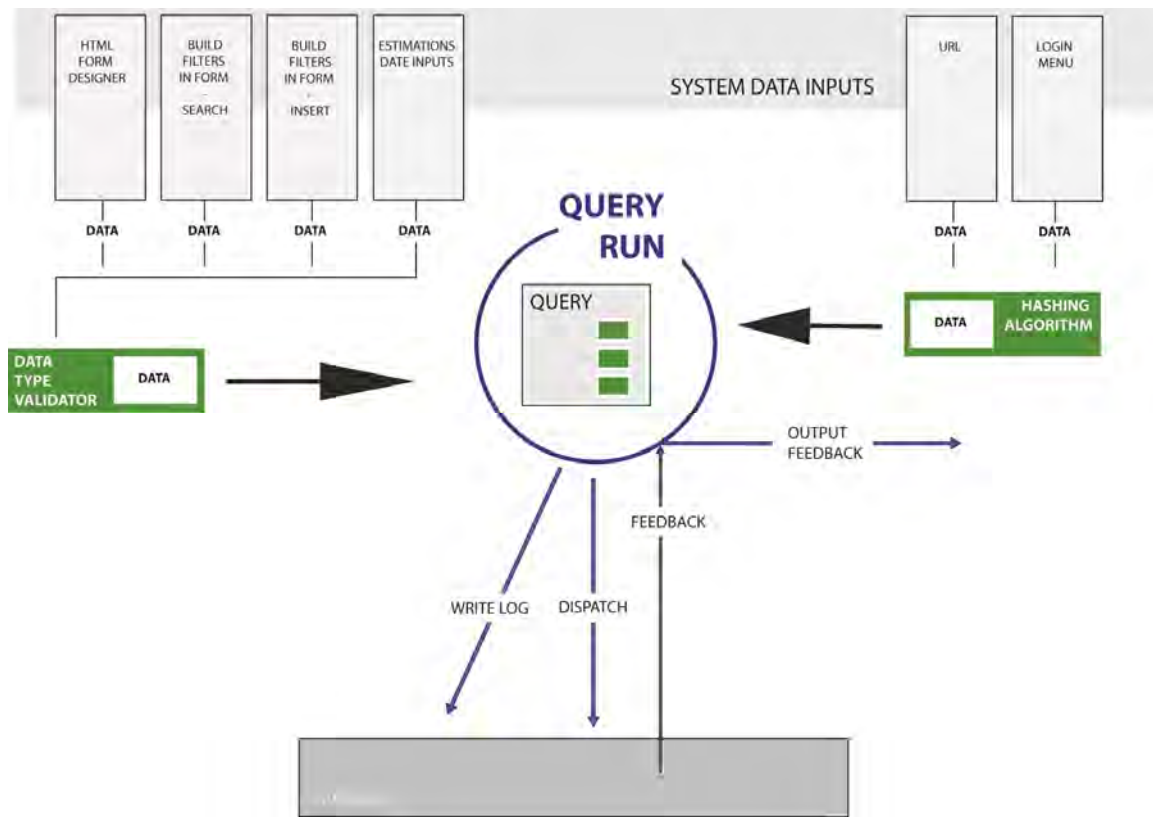

Figure 1.19 FORM DATA SUBMISSION GENERATE MESSAGES

DISPATCH QUERIES DELEGATE

Figure 1.20 'QUERY RUN' ASSIGNMENTS