

A Model Checking - Verify Web Application

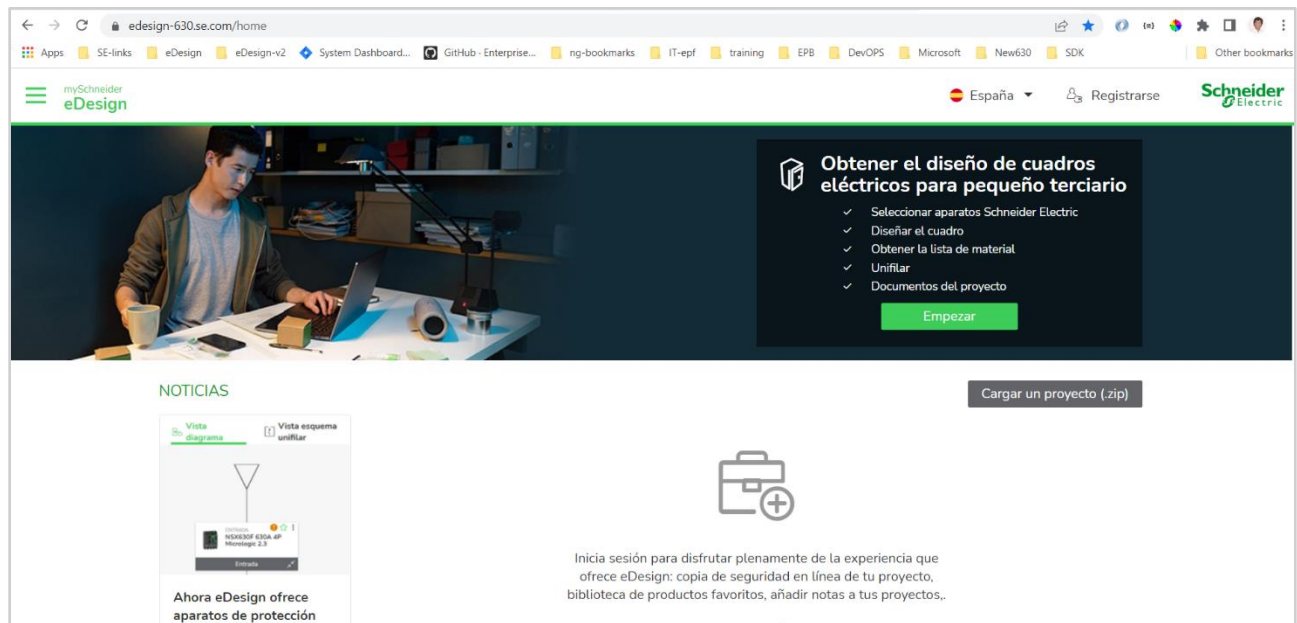
Application Name: eDesign(new630) – small building

Introduction

Web Applications are becoming more and more widespread and efficient, then an increase of their reliability is now strongly required. Hence methods to support design and automatically perform validation of a Web Application (WA) could be helpful. In this I document I am presenting eDesign(new630) Application verification.

New630 is a switchboard configuration web application that allows users to define, configure and quote electrical switchboards up to 630A. It is built for electricians, and it targets Small Tertiary installations. This application is available only for EU countries. Hence we don't have English language view.

Application URL : <https://edesign-630.se.com/home>



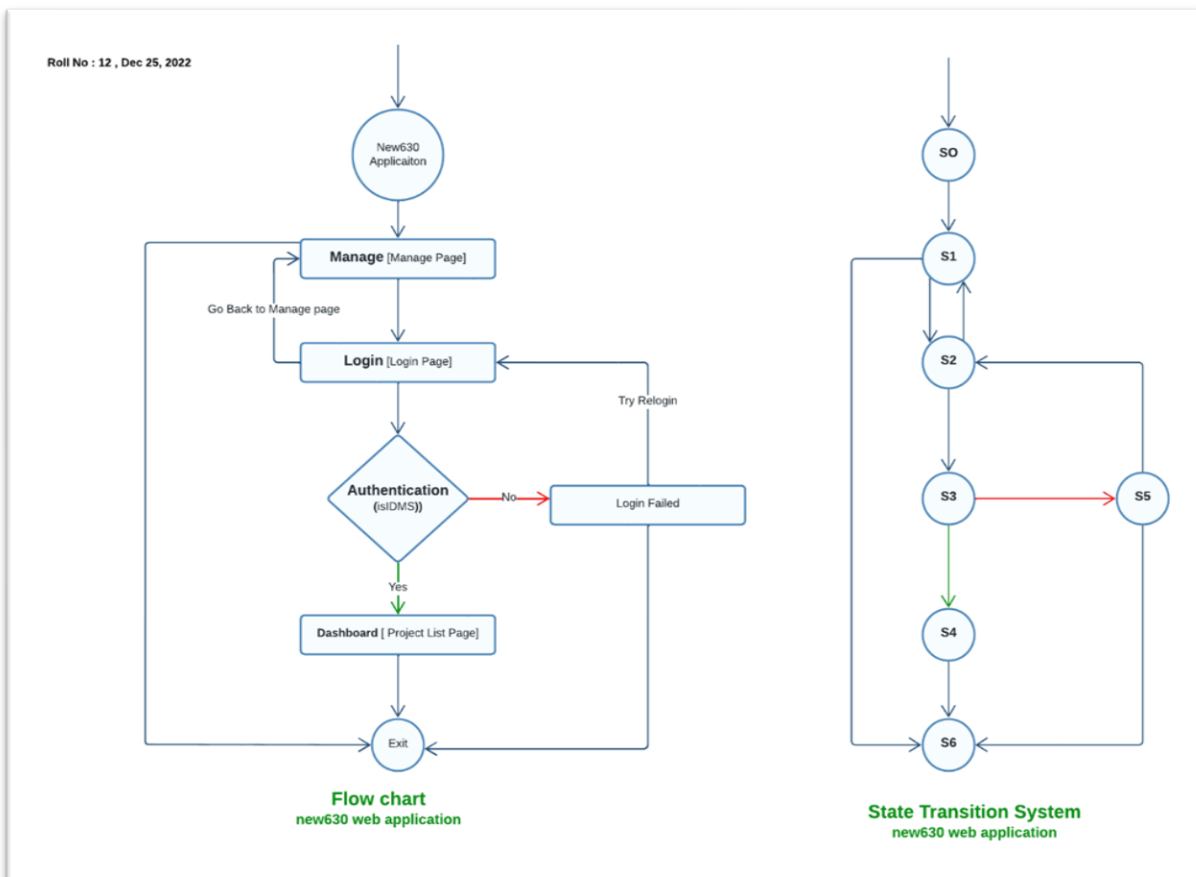
This application is used by electrician to design electric circuits by the needs of small builds. He can design switchboard, SLD, generate BOM. Also this application provides BOM, SLD download document features.

State Transition Details

In the proposed approach, the SMV model is derived from a web application navigation model which is drawn in Unified Modeling Language (UML – Flowchart in the below image). The model defined in this document is Kripke structure to model Each state represents a particular navigation step in the workflow of the online system.

The equivalent state model representation of the flowchart navigation of the Online **eDesign** is shown in the below image, where each state is,

1. S0: Application – idle state → **eDesignNew630**
2. S1: Homepage; welcome page. → **Homepage**
3. S2: Login; to be done by the user → **Login Page**
4. S3: Authentication; → **Authentication**; navigate to IDMS for authentication/user login.
5. S4: Dashboard with all the projects fetched from SPIM → Login Failed, if the login credentials are correct
6. S5: Login Failed, → **Dashboard**, if the login credentials are incorrect.
7. S6 : Exits the application → **Exit**. User can close the application doing no action or he can login and perform operation then exit.



I used Lucidchart is used to draw above diagram.

NuSMV VERIFICATION

To verify the system model, I am using the NuSMV (New Symbolic Model Verifier) tool. NuSMV is a model checker tool based on the SMV symbolic model checker.

Run the following commands:

1. Open the command prompt, if nuSMV is not added to system path, go to NuSMV folder, navigate to source code file, and run, else you can directly run the command from anywhere with correct .svm file.

C:\Users\Sandeep>**NuSMV -int**

2. NuSMV > read_model -I edesign_model_verification.smv

3. NuSMV > flatten_hierarchy

4. NuSMV > encode_variables

5. NuSMV > build_model

Below is the NuSMV program for the verification.

```
MODULE main
VAR
    state: {s0, s1, s2, s3, s4, s5, s6};
    eDesignNew630: boolean;
    Homepage:boolean;
    LoginPage: boolean;
    Authentication: boolean;
    LoginFailed: boolean;
    ProjectDashboardPage: boolean;
    Exit: boolean;

ASSIGN
    init(state):= s0;
    init(Hompage):=TRUE;
    init(LoginPage):=FALSE;
    init(Authentication):=FALSE;
    init(eDesignNew630):=FALSE;
    init(LoginFailed):=FALSE;
    init(ProjectDashboardPage):=FALSE;
    init(Exit):=FALSE;

next(state):=case
    state=s0: {s1};
    state=s1: {s2,s6};
    state=s2: {s1,s3};
    state=s3: {s4,s5};
```

```
        state=s4: {s6};
        state=s5: {s2,s6};
        state=s6: {s6};
            TRUE: state;
esac;

next(eDesignNew630):=case
    next(state)=s0: TRUE;
    TRUE:FALSE;
esac;

next(Homepage):=case
    next(state)=s1: TRUE;
    TRUE:FALSE;
esac;

next(LoginPage):=case
    next(state)=s2: TRUE;
    TRUE:FALSE;
esac;

next(Authentication):=case
    next(state)= s3:TRUE;
    TRUE:FALSE;
esac;

next(LoginFailed):=case
    next(state)=s5:TRUE;
    TRUE:FALSE;
esac;

next(ProjectDashboardPage):=case
    next(state)=s4:TRUE;
    TRUE:FALSE;
esac;

next(Exit):=case
    next(state)=s6:TRUE;
    TRUE:FALSE;
esac;
```

```
D:\code\nuSMV_assignment>NuSMV -int
*** This is NuSMV 2.6.0 (compiled on Wed Oct 14 15:37:51 2015)
*** Enabled addons are: compass
*** For more information on NuSMV see <http://nusmv.fbk.eu>
*** or email to <nusmv-users@list.fbk.eu>.
*** Please report bugs to <Please report bugs to <nusmv-users@fbk.eu>>

*** Copyright (c) 2010-2014, Fondazione Bruno Kessler

*** This version of NuSMV is linked to the CUDD library version 2.4.1
*** Copyright (c) 1995-2004, Regents of the University of Colorado

*** This version of NuSMV is linked to the MiniSat SAT solver.
*** See http://minisat.se/MiniSat.html
*** Copyright (c) 2003-2006, Niklas Een, Niklas Sorensson
*** Copyright (c) 2007-2010, Niklas Sorensson

NuSMV > read_model -i edesign_model_verification.smv
NuSMV > flatten_hierarchy
NuSMV > encode_variables
NuSMV > build_model
NuSMV > how_vars
unknown command 'how_vars'
NuSMV > show_vars
Number of Input Variables: 0
Number of State Variables: 8
  1: {s0, s1, s2, s3, s4, s5, s6}
  7: boolean
Number of Frozen Variables: 0
  state : {s0, s1, s2, s3, s4, s5, s6}

eDesignNew630 : boolean

Homepage : boolean

LoginPage : boolean

Authentication : boolean

LoginFailed : boolean

ProjectDashboardPage : boolean

Exit : boolean

Number of bits: 10 (0 frozen, 0 input, 10 state)
NuSMV >
```

```
NuSMV > check_ctlspec -p "AG (EF(Hompage -> LoginPage))"
-- specification AG (EF (Hompage -> LoginPage)) is true
NuSMV > check_ctlspec -p "AG (EF(LoginPage -> Homepage))"
-- specification AG (EF (LoginPage -> Homepage)) is true
NuSMV > check_ctlspec -p "AG (EF(ProjectDashboardPage -> Homepage))"
-- specification AG (EF (ProjectDashboardPage -> Homepage)) is true
NuSMV > check_ctlspec -p "AG (AX(ProjectDashboardPage -> Homepage))"
-- specification AG (AX (ProjectDashboardPage -> Homepage)) is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
-> State: 1.1 <-
  state = s0
  eDesignNew630 = FALSE
  Homepage = TRUE
  LoginPage = FALSE
  Authentication = FALSE
  LoginFailed = FALSE
  ProjectDashboardPage = FALSE
  Exit = FALSE
-> State: 1.2 <-
  state = s1
-> State: 1.3 <-
  state = s2
  Homepage = FALSE
  LoginPage = TRUE
-> State: 1.4 <-
  state = s3
  LoginPage = FALSE
  Authentication = TRUE
-> State: 1.5 <-
  state = s4
  Authentication = FALSE
  ProjectDashboardPage = TRUE
NuSMV >
```

```
NuSMV > check_ltlspec -p "G F(LoginPage -> X Authentication)"
-- specification G ( F (LoginPage -> X Authentication)) is true
NuSMV > check_ltlspec -p "G F(LoginPage -> X ProjectDashboardPage)"
-- specification G ( F (LoginPage -> X ProjectDashboardPage)) is true
NuSMV > check_ltlspec -p "G F(LoginPage -> F ProjectDashboardPage)"
-- specification G ( F (LoginPage -> F ProjectDashboardPage)) is true
NuSMV >
```

Conclusion

I have tested the happy path of application home page, Login, Dashboard and Exit the application.

Git Repo : https://github.com/MTech2022/smv_assignment