Survey

Table of Contents

Peter Petkanic

I was member of four-member team. Together we did our best to complete our assignment - survey management system. I was responsible for XML databases creation as well as their XML schemas and also for design of our desktop application.

Survey

XML Databases

My task was to create XML files for surveys and their responses.

To save our surveys I used *survey.xml* file. In order for surveys to be unique I used sid as an attribute that is unique in whole file. Beside title we added description element for title and description representation that will be used in our web interface. Also, every survey has its questions set. Each question has qid attribute unique in its survey, question description and set of answers. For simplicity, we decided to use only two types of questions. Closed (only one possible answer) and multiple (0-n) answers. Type is saved in type attribute of question. Each answer has unique aid attribute in its question element. The reason why I used id for all entities isfor easier implementation of survey create and edit functions in our desktop app.

Reponses (filled surveys) are saved in *response.xml* file. Responses does not require unique id, since survey is anonymous. In this file are saved IDs of all selected elements in web interface. I save IDs inside element not as an attribute because IDs are not metadata in this case, they are raw data. XML tree is same as in *survey.xml*.

XML Schema

In order to validate XML databases I decided to use XML schema.

For validation of *survey.xml* files I used schema in *survey.xsd*. All elements in survey are required. Every set of questions has to have at least one question with at least two answers.

response.xsd is used to validate response.xml files. If response element is present it means that respondent has selected at least one answer. If question element is present it means that respondent has selected at least one answer in this question.

In order to prove that my schemas are valid I created few valid and invalid XML files.

Desktop Application

I tried to make our application structured as *survey.xml* and keep the tree of survey. When you enter application you can see surveys than questions than answers just like XML database.

I created *jPanelMain.java*, *jPanelSurvey.java* and *Main.java*. Application uses *SurveyEngine.java* that communicates with XML database. This class is initialized in main class and than handed down to each panel, that is dependant on its functions.

Main.java

This file is the main class of whole application. It loads XML file, validates it and prepare SurveyEngine for later use.

jPanelMain.java

This class represents first panel of application after startup. It lists all surveys in survey.xml file. From here we are able to create new survey or edit existing survey.

jPanelSurvey.java

This class represents survey create/edit panel. We can change title and description as well as edit or create questions. This class was very good template for team member that created *jPanelQuestion.java*.