|  |  |  |  |
| --- | --- | --- | --- |
| **PSEUDO-CODE** | | | |
| **Explanation & theory:**  In the original proposal we had two columns in the survey form (formsquestionnaire): JumpConditions and JumpTo.  In this proposal I have removed JumpConditions as it seem just to introduce redundancy.  In the form definition stage in the web interface I have created and extra step where the table with questions are presented row by row but without JumpTo data.  1. When the questions and their order is defined the survey form is created as a separate database table.  2. A new webform is opened. When it opens, the rows (questions) in the database table are tested one by one.  2.1 If it is a Free Text the questions are written out row by row in lines in the webform, and the attributes are in columns from the database table, and with no option to make a JumpTo from that question. All columns are presented as TEXT ONLY.  2.2 If it is a SC or MC the question is simply written out as it is as text. But in the column with options new “subrows” are created for each option. For each option, in the same row in the webform, there is an **edit box** where the user simply enters the number of the question the user should be guided to if that option is clicked. Each **edit box** has a default value of “0” indicating no “jump”.  3. After the user edits the needed jumps the completed form (db-table) should be saved into the database. The jumps for each question is stored as a semi-colon delimited “jumplist” with the question numbers to jump to as follows: “X;Y;0;0;0;”. The choice of the first two options will lead to a jump to Question X or Y. The last three options will not lead to any jumps.  4. After download to the mobile app, assuming that the whole survey is in one scrollable activity, an OnClickListener indicates if a jump-conditioned option is chosen. Then the app disables all questions between the current question up to the question that the jump leads to. | | | |
| **Data tables** | * MForm * MQuestions * MSurveyForms | | |
| **Fields:** | * MSurveyForms.JumpTo [String] | | |
| **Stringformat:** | MQuestions.Options[String] | “Option1;Option2;Option3” | |
|  | MSurveyForms.Jump[String] | “QuestionIDx1;QuesitonIDx2;NULL;” | |
| **Conditions:** | Jump conditions should only be enabled for single-choice type questions. | | |
| **Pseudo-code segment:** | 1. Preparation of survey-form in webinterface | | |
| **Pseudo-code:** | Click button Add Form  WebForm SurveyForm Design 1 open in browser  Enter SurveyForm Name  Enter SurveyForm Description  Add questions to survey  (Sort questions if needed)  OnClick button Next  Create Table MSurveyForm  Questions are stored in table MSurveyForm in the right order.  WebForm SurveyForm Design 2 open in browser  i==i+1  WHILE Table.MSurveyForm not EndofTable  Select Row (i) From MSurveyForm  IF MSurveyForm(i).QuestionType=”FreeText” THEN  POST MSurveyForm(i).QuestionId INTO TEXTBOX(TBQId) at COLUMN (1), LINE (i) in “WebForm Survey Design 2”  POST MSurveyForm(i).Question INTO TEXTBOX(TBQuestion) at COLUMN (2), LINE (i) in “WebForm Survey Design 2”  POST MSurveyForm(i).QuestionType INTO TEXTBOX(TBQuestionType) at COLUMN (3), LINE (i) in “WebForm Survey Design 2”  POST MSurveyForm(i).Options INTO TEXTBOX(TBOptions) at COLUMN (4), LINE (i) in “WebForm Survey Design 2”  POST “Not applicable” INTO TEXTBOX(TBCondition) at COLUMN (4), LINE (i) in “WebForm Survey Design 2”  POST “Not applicable” INTO TEXTBOX(TBJumpTo) at COLUMN (1), LINE (i) in “WebForm Survey Design 2”  ELSE  POST MSurveyForm(i).QuestionId INTO TEXTBOX(TBQId) at COLUMN (1), LINE (i) in “WebForm Survey Design 2”  POST MSurveyForm(i).Question INTO TEXTBOX(TBQuestion) at COLUMN (2), LINE (i) in “WebForm Survey Design 2”  POST MSurveyForm(i).QuestionType INTO TEXTBOX(TBQuestionType) at COLUMN (3), LINE (i) in “WebForm Survey Design 2”  POST MSurveyForm(i).Options INTO TEXTBOX(TBOptions) at COLUMN (4), LINE (i) in “WebForm Survey Design 2”  j==1  WHILE Table.MSurveyForm.Options not EndofString  optiontext==EXTRACT Table.MSurveyForm.Options(j)  *EXTRACT searches for the j:th option in the string of options. Options are separated by semicolons as follows Option1;Option2:Option3;…;OptionN.*  POST optiontext INTO TEXTBOX(TBOption) at COLUMN (4), LINE (i) in “WebForm Survey Design 2”  POST “*0*” INTO EDIT BOX(EBJump[i][j]) at COLUMN (1), LINE (i) in “WebForm Survey Design 2” *COMMENT: The edit box EBJump is 2-dimensional string-array with” i” corresponding to the table row and “j“ corresponding to the column.*  j==j+1  END WHILE  ENDIF  i==i+1  END WHILE  *COMMENT: Questions are lined up in the selected order in a “table” in the web browser. In the rightmost column, an edit box is created to enter the “jump” for each option if the question is an SC or MC question. If a FT question the editboxes are replaced with textboxes writing “Not applicable”.*  ***USER INPUT:***  ***The user can now for each option in all questions, enter where a response should lead to a “jump” to a specific question in the form. A question is referred to by its “order number”***  **Reading the input from user**  OnClick Button Save SurveyForm  *COMMENT: The edit boxes on the form has either a “0”, indicating there is no “jump”, or a number indicating where to “jump” for each option.*  i==i+1  WHILE Table.MSurveyForm not EndofTable  Select Row (i) From MSurveyForm  IF MSurveyForm(i).QuestionType=”FreeText” THEN  <DO NOTHING>  ELSE  j==1  WHILE (EBJump[i][j]) not EndofString  jumpString==jumpString+ EDIT BOX(EBJump[i][j]);  j==j+1  *COMMENT: The WHILE creates a semi-colon delimited string of jumps like the following for a three option jump conditions: “0,5,8;”  For the first option, the user will be automatically brought to the next question in the form. If the user responds with the second option the user will be referred to question #5 or if the third option is chosen, the user will be brought to question #8.*  END WHILE  UPDATE TABLE Table.MSurveyForm.Jump(i) WITH jumpString  ENDIF  i==i+1  END WHILE  *The MSurveyForm is now ready to upload* | |  |
| **Pseudo-code segment:** | 2. The mobile app interpretation of the logical jumps | |  |
| **Pseudo-code** | **Assumption:** The questions are downloaded and placed in one scrollable activity with all questions.  ***Example SC radio button with two choices on Question with order# 3.***  *Table-data:*  *a) Table.MSurveyForm.Option(i) = “Yes;No;”*  *b) Table.MSurveyForm.Jump(i)= “6,0;”*    *Input:*  *the user click “Yes” (option 1) on the SC radiobutton.*  view.OnClickListener inform that Option[1] (“Yes”) is chosen.  MobileApp extract Jump[1] in the string *“6,0;” (=”6”)*  MobileApp **disable** all questions with Order#>currentQuestion.Order# AND Order#<Question.Order#[6]  FOR i= currentQuestion.Order#+1 TO Order#<Question.Order#[6]  Disable(Question.Order#[i], SET FONT COLOR “GRAY”)  NEXT i | |  |
|  | | | |