

# W13-011

# Project Report

Max Chung

A00743631

COMP 8045 (9 Credits)

# Table of Contents

Abstract.....	6
Introduction.....	6
Company Background.....	6
Project Background.....	7
Rationale for Custom Solution.....	7
Project Goals.....	7
Project Deliverables.....	11
Functional Requirements.....	11
Admin Backend.....	12
User Section.....	15
Development Details.....	18
Methodology.....	18
Gather Requirements.....	18
Create Proposal.....	18
Do Visual Design.....	18
Perform usability testing with mock-ups.....	18
Create Database Schema.....	19
Setup Development Environment.....	19
Create Page Templates.....	19
Enter Initial Data.....	19
Hook up Backend.....	19
Testing and Bug Fixes.....	19
Do Manual Browser Compatibility Checks.....	19
Deploy on Live Site.....	19
Create Project Report.....	20
Obtain Final Client Signoff.....	20
Project Schedule.....	20
Entity Relationship Diagram.....	24
Database Tables with Sample Data.....	27
Message Sequence Chart.....	27
Creating a new section for a published questionnaire (FR3).....	28
Creating a new section for an unpublished questionnaire (FR3).....	29

Editing an existing section for a published questionnaire (FR3)..... 30

Editing an existing section for an unpublished questionnaire (FR3) ..... 31

Creating a new category for a published/unpublished questionnaire (FR2)..... 32

Editing an existing category for a published questionnaire (FR2)..... 33

Editing an existing category for an unpublished questionnaire (FR2) ..... 34

Deleting a section for a published questionnaire (FR3)..... 35

Deleting a section for an unpublished questionnaire (FR3)..... 36

Rearranging sections for a published questionnaire (FR5) ..... 37

Rearranging sections for an unpublished questionnaire (FR5)..... 38

Editing questionnaire overview for a published questionnaire (FR9, FR10) ..... 39

Editing questionnaire overview for an unpublished questionnaire (FR9, FR10) ..... 40

Implementation Implications..... 41

Use of new technologies..... 42

    Responsive Layout & Twitter Bootstrap..... 42

    Clicking vs. Tapping..... 45

CSS3 Modals and Transitions ..... 48

MeekroDB ..... 50

Use of APIs..... 50

Questionnaire Revision Control..... 50

AJAX Form Submission ..... 51

Future Enhancements..... 51

    Conditional Branching of Sections..... 51

    Support for Browsers with JavaScript Disabled ..... 51

    AJAX Form Submission per Question ..... 51

    Colour Coded Categories..... 52

    Improved Authentication System ..... 52

    Remove Questionnaire Categories..... 52

    Purge Old Answer Sets and Questionnaire Revisions..... 52

    View Old Questionnaire Revisions..... 52

    Preview Questionnaire..... 52

    Support for Multiple Questionnaires..... 52

Development Environment Network Topology..... 53

Technologies ..... 53

Innovation..... 54

    Usability Testing ..... 54

    Questionnaire Revision Control..... 56

    Custom Formatted PDF Creation ..... 56

    HTML and CSS Styled Emails..... 58

Conclusion ..... 60

Appendix A– Design Process & Mock-ups ..... 62

    Questionnaire Mock-up Version 1 ..... 62

    Questionnaire Results Mock-up Version 1..... 63

    Questionnaire Mock-up Version 2..... 68

    Questionnaire Mock-up Version 3..... 69

    Questionnaire Results Mock-up Version 3..... 72

    Questionnaire Results Mock-up Version 4.....77

    Questionnaire Results Mock-up Version 5 ..... 82

    Questionnaire Mock-up Mobile Version..... 86

Appendix B – Database Tables with Data ..... 98

Appendix C – User Manual..... 98

    Web Server and Database Installation ..... 98

    Website and Database Configuration..... 99

    Customizing the Questionnaire Overview..... 100

    Adding Categories to the Questionnaire..... 100

    Adding Sections and Questions to the Questionnaire..... 100

    Editing Questionnaire Categories..... 100

    Editing Questionnaire Sections and Questions ..... 100

Appendix D – Usability Testing..... 101

    Desktop Usability Test ..... 101

        Summary of Changes..... 101

        Where will you click to name the project?..... 102

        Where will you click to get more information about a question?..... 104

        Where will you click to answer a question?..... 106

        Where will you click to save the questionnaire to finish later? ..... 108

        What do you expect will happen when the “Next Section” button is clicked? ..... 110

        Where will you click to correct this error?..... 112

What do you expect will happen when each item in the actions sidebar is clicked?..... 114

Which form fields do you think are required?..... 115

How can the input be corrected so the form will submit?..... 117

Mobile Usability Test ..... 119

    Summary of Changes..... 121

    Where will you tap to name the project?..... 121

    Where will you tap to get more information about a question?.....123

    Where will you tap to answer a question?.....124

    Where will you tap to save the questionnaire to finish later? .....125

    What do you expect will happen when the “Next Section” button is tapped? .....126

    Where will you tap to correct this error?.....128

    You want to save your results. Where will you tap? .....129

    What do you expect will happen when one of the menu options is selected?..... 131

    Which form fields do you think are required?.....132

    How can the input be corrected so the form will submit?.....133

Appendix E – Site Map .....134

Appendix F – Test Cases & Results of Testing.....136

    Admin Section.....136

    User Section .....139

Appendix F – Writing and Reading from the Database .....142

## Abstract

Project managers and students can benefit from online decision-making tools as it can assist them in choosing appropriate plans of action based on certain criteria. The current off-the-shelf survey solutions are not suitable choices for a decision-making tool because they only accept responses and do not supply feedback. The current available tools also are inflexible or difficult to use as they are difficult to customize, do not support question categorization, and question weighting, which are needed in order to determine applicable methodologies or are difficult to customize. This project will create a customizable decision-making tool that addresses these needs.

## Introduction

Community Solutions needs a custom made online decision-making tool to help evaluators determine the evaluation approach they should use based on answering Yes or No questions on questions regarding background factors such as the type of program, evaluator role, and type of participants.

A questionnaire similar to the format used by the ARECCI Ethics Screening Tool needs to be created with customizable sections, questions, and recommendations based on question responses as the ARECCI Ethics Screening Tool requires modifying the HTML page mark-up in order to customize the questionnaire. The custom solution aims to go beyond the ARECCI Ethics Screening Tool's capabilities by supporting questionnaire versioning so that previous questionnaires and questionnaire responses remain accessible as new versions are created.

The ARECCI Ethics Screening Tool can be accessed from <http://www.aihealthsolutions.ca/arecci/screening>.

This project further enhances my expertise in my chosen specialization because it allows me to apply my skills in entity-relationship (EER) modeling, data normalization, and creating schemas from EER models. This project also allows me to do wireframes, visual design, templates, AJAX form submissions, and API calls, which is vital in web development.

## Company Background

Community Solutions is a small consulting firm that aims to help organizations improve their delivery of program and services. This company uses a community-based and participatory approach for program evaluation to:

- Increase stakeholder involvement by including management and staff in the design and implementation phase of projects.
- Improve program and service sustainability by training and mentoring staff to explain what evaluation is and how it benefits them.

- Ensure the correct implementation of results and recommendations by assisting clients and following-up on all the projects.

## Project Background

Community Solutions needs a customizable questionnaire, similar to the format of the ARECCI Ethics Screening Tool to be used by program evaluators, both new and experienced, and in training new evaluators by showing them the factors used to determine suitable evaluation approaches. These factors include the context, the type of program, the type of participants, and the program outcomes.

The questionnaire recommends evaluation approaches based on the responses to Yes or No questions.

The client hopes for the tool to be incorporated in the curriculum for some graduate courses on evaluation.

## Rationale for Custom Solution

Although there are many existing online questionnaire creation tools such as SurveyMonkey and QuestionPro, none of them satisfy the client's requirements. One of the critical pieces of the tool is the ability to calculate and display results to the user after they have completed the questionnaire. In addition, the client wants the ability to easily modify the question set and adjust aspects like question weighting in order to have fine grain control of how the results are calculated and shown to the user.

Tools like SurveyMonkey and QuestionPro were designed to collect information from the user and to aggregate these for the survey creator and does not meet the client's requirements for creating a tool that display results directly to the user.

Other sites like HelloQuizzy and OkCupid offer tools that ask questions to generate simple results term "quizzes", but they are used for entertainment purposes (e.g. a quiz for "What Harry Potter character are you?"). The tone of these sites is not suitable for the client's questionnaire, nor do they offer custom branding for the client.

The client also feels it is important for the user to be able to save the results for further references. Additional functionality like exporting the results as a PDF, sharing of results via email needs to be available.

## Project Goals

The goal of this project is to create a custom usability tested questionnaire creation solution for Community Solutions that follows the design and layout of the ARECCI Ethics Screening Tool.

The following images are screen captures taken from the above website and are used as a reference when designing and creating the questionnaire layout. These reference images show the basic set of functionalities that needs to be included in the custom solution although the layout may differ in the implemented solution. Such functionality include the option to get more information about a question, save the questionnaire link for future access, email the questionnaire results, print the questionnaire results, and get a recommendation upon completion of the questionnaire.

Figure 1 shows the legend used in the ARECCI Ethics Screening Tool. All of the functions shown in this legend was included in the custom solution.

The functionality to get more information about a question as shown in Figure 2 must be included in the custom solution because it can provide a more detailed explanation of the question and perhaps explain any technical terms to avoid misunderstandings, which can potentially affect the suggested methodologies and give the user an incorrect suggestion.

The functionality to save the questionnaire to complete later is included in the custom solution as the questionnaire may be too long to complete in one session and the user wants to save the questionnaire responses before leaving.





The functionality to calculate and show relevant methodologies as shown in Figure 3 is included in the custom solution because the sole purpose of the decision-making tool is to make decisions.

The functionality to email the questionnaire results is included in the custom solution for convenience so the user can quickly email the results for safekeeping. This is also useful for mobile users who may want to review the results on their desktop computer.

The functionality to print the questionnaire results is included in the custom solution so the recommended methodologies and questionnaire responses will be neatly formatted on the page for future reference.

Since the client has the ability to modify the questionnaire and users must be able to reference their previous results, the questionnaire system also needs to have version control keep track of changes. I have implemented a custom database schema so older questionnaire versions will continue to work and be accessible as newer questionnaires are created.

## Legend

-  Provides more information about a question (if available).
-  Save the current state of the tool and retrieve a URL for bookmarking.  
*Note: This is not the same as selecting File > Save As in your browser. Please use this save function instead.*
-  E-mail the results of the tool.
-  Print the results of the tool.

**Figure 1 - Legend shown on the reference website**

 1. Is there an explicit requirement for review of this project by a Research Ethics Board as part of its funding arrangements?

---

*This item refers to projects where the funder requires ethics review by a REB. Examples of such funding agencies are: the Canadian Institutes of Health Research, the Canadian Health Services Research Foundation, the Natural Sciences and Engineering Research Council of Canada, the Social Sciences and Humanities Research Council, and Alberta Innovates – Health Solutions. Projects funded by these agencies are typically (but not always) considered research and all are required to undergo REB ethics review.*

Yes

No

**Figure 2 - Question with question details shown on the reference website**

This project should be submitted to a  
Research Ethics Board.

**Figure 3 - Suggestion based on questionnaire responses shown on the reference website**

People are also using smart phones and tablets to access the Internet, so the questionnaire needs to be easily used on devices other than a standard desktop computer. The user-facing side of the application was designed and built to be “responsive” so that the application will adjust to fit in and be easy to use on different screen dimensions.

Visual design mock-ups for the mobile and desktop versions of the web application were created based on reference images from the ARECCI Ethics Screening Tool and uploaded into Usabilla, an online service that allows testers to provide usability feedback based on screenshots. This process allowed for the incorporation of user feedback as well as ironing out of any usability issues before the web application is developed. More details about this process are found in the methodology section.

## Project Deliverables

- Project proposal
- Visual mock-ups of admin and public sections (see Appendix A– Design Process & Mock-ups)
- Database schema (see Appendix B – Database Tables with Data)
- Static templates (contained in zipped code package)
  - Admin section page templates
  - Public section page templates
  - Email template
  - PDF template
- Electronic questionnaire and admin site (contained in zipped code package)
- Test Plans & Detailed Test Cases (see Appendix F – Test Cases & Results of Testing)
- User documentations (see Appendix C – User Manual)
- Usability Studies Report (see Appendix D – Usability Testing)

## Functional Requirements

The scope of this practicum is to create a questionnaire that supports the following features:

### Admin Section

- Secure admin with a login (FR1)
- Support for creating and editing multiple questionnaires
  - Create/edit questionnaire categories (FR2)
  - Create/edit or remove questionnaire sections (FR3)
  - Create/edit or remove questions from the questionnaire (FR4)
  - Reorder questionnaire sections (by drag and drop using jQuery sortable) (FR5)
  - Reorder questionnaire questions (by drag and drop using jQuery sortable) (FR6)
  - Customize question category (Color coded) (FR7)
  - Provide explanation for each question (FR8)
  - Rename the questionnaire (FR9)
  - Change the overview text (FR10)
- Support for publishing questionnaire changes (FR11)

**Public Section**

- View and complete questionnaire (FR12)
- Load saved questionnaire answers (FR13)
- Show an explanation of the question with a click of a button (FR14)
- Change the yes or no value of each question (FR15)
- Add optional notes at the end of each section (FR16)
- Validate all questionnaire responses before continuing to the next section (FR17)
- Automatically save form contents on section completion (FR18)
- Display questionnaire recommendations on questionnaire completion (FR19)
- Generate a permalink to revisit a questionnaire (FR20)
- Print questionnaire questions and results to a PDF (custom formatted with FPDF) (FR21)
- Email summary of questionnaire results with permalink to actual results (formatted and styled with HTML and inline CSS) (FR22)

These requirements are described in more details below.

**Admin Backend**

Title:	Logging into the questionnaire backend (FR1)
Description:	As an administrator, I need the ability to log into the backend so I can make edits to the questionnaire.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ An authentication dialog will appear for an unauthenticated user upon visiting one of the backend pages.</li> <li>■ If the user credentials are valid, the user will be redirected to the admin landing page and the auth-attempts variable will be reset to 0.</li> <li>■ If the user cancels authentication, an authentication required message will appear.</li> <li>■ If the user enters invalid user credentials, the authentication dialog will appear again and the auth-attempts variable will increment by 1.</li> <li>■ If the user enters invalid user credentials more than five times, a forbidden message will appear and the user will not be able to authenticate again for an hour.</li> </ul>

Title:	Renaming the questionnaire and changing the overview text (FR9, FR10)
Description:	As an administrator, I need the ability to rename the questionnaire and change the tool overview text.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ The user will be able to enter the questionnaire title and tool overview text on the admin landing page.</li> <li>■ Upon clicking on the Save Overview Changes button, a new questionnaire revision with the changed questionnaire title and tool overview will be created.</li> </ul>

	<ul style="list-style-type: none"> <li>■ The new questionnaire title will show in the page header.</li> <li>■ An unpublished changes message with a Publish Changes button will appear when the questionnaire title or the tool overview is changed.</li> </ul>
--	---

Title:	Creating categories (FR2, FR7)
Description:	As an administrator, I need the ability to create categories.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking the Add Category button will take the user to the Add Category page with a blank form to create a new category.</li> <li>■ On the Add Category page, the user will be able to enter the category details</li> <li>■ The Save Changes button on the Add Category page will create the new category.</li> <li>■ An unpublished changes message with a Publish Changes button will appear when the new category is created.</li> <li>■ The name of the new category will appear under the Questionnaire Categories section.</li> </ul>

Title:	Creating sections (FR3, FR4, FR6, FR7, FR8)
Description:	As an administrator, I need the ability to create sections.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking the Add Section button will take the user to the Add Section page with a blank form to create a new section.</li> <li>■ On the Add Section page, the user will be able to enter the section and question details.</li> <li>■ The category drop down menu will show the categories the question can be assigned to.</li> <li>■ Questions can be added by clicking the Add Question button.</li> <li>■ Questions can be removed by clicking the X button at the top right corner of a question.</li> <li>■ Questions can be rearranged by dragging them.</li> <li>■ The Save Changes button on the Add Section page will create a new questionnaire revision with the new section.</li> <li>■ An unpublished changes message with a Publish Changes button will appear when the new section is created.</li> <li>■ The name of the new section will appear under the Questionnaire Sections section.</li> </ul>

Title:	Editing sections (FR3, FR4, FR6, FR7, FR8)
Description:	As an administrator, I need the ability to edit sections.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking the section name under the Questionnaire Sections section will take the user to the Edit Section page with a form populated with the section details.</li> <li>■ On the Edit Section page, the user will be able to edit the section and</li> </ul>

	<p>question details.</p> <ul style="list-style-type: none"> <li>■ The category drop down menu will default to the category the question is assigned to.</li> <li>■ Questions can be added by clicking the Add Question button.</li> <li>■ Questions can be removed by clicking the X button at the top right corner of a question.</li> <li>■ Questions can be rearranged by dragging them.</li> <li>■ If the questionnaire is published, the Save Changes button on the Edit Section page will create a new questionnaire revision with the new section details else the current section is overwritten with the changes.</li> <li>■ An unpublished changes message with a Publish Changes button will appear when the section is edited.</li> <li>■ The name of the edited section will appear under the Questionnaire Sections section.</li> </ul>
--	---

Title:	Reordering sections (FR5)
Description:	As an administrator, I need the ability to reorder sections.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking and dragging the grey box surrounding each section name under the Questionnaire Sections section will allow the section to be rearranged.</li> <li>■ If the sections have been rearranged, clicking the Save Changes button will create a new questionnaire revision with the new section order else nothing happens.</li> <li>■ An unpublished changes message with a Publish Changes button will appear when the sections are rearranged.</li> <li>■ The new section order will appear under the Questionnaire Sections section.</li> </ul>

Title:	Deleting sections (FR3)
Description:	As an administrator, I need the ability to delete sections.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking the X button to the right of each section will allow the section to be deleted.</li> <li>■ If any section has been deleted, clicking the Save Changes button will create a new questionnaire revision with the deleted sections removed from the section order else nothing happens.</li> <li>■ An unpublished changes message with a Publish Changes button will appear when a section has been deleted.</li> <li>■ The new section order will appear under the Questionnaire Sections section.</li> </ul>

Title:	Publishing questionnaires (FR11)
Description:	As an administrator, I need the ability to publish questionnaires so the questionnaire changes are visible to the public.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking the Publish Changes button beside the unpublished changes message will cause a confirmation dialog to appear.</li> <li>■ If the user selects Yes in the confirmation dialog, the is_published flag for the current unpublished questionnaire will be set to 1 else, nothing happens.</li> <li>■ When the questionnaire is published, the unpublished changes message will disappear.</li> <li>■ The published questionnaire changes will be visible on the questionnaire page.</li> </ul>

Title:	Logging out of the questionnaire backend (FR1)
Description:	As an administrator, I need the ability to log out of the backend so unauthorized users cannot access it.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking the Log Out button in the page header will cause the user's session to be invalidated.</li> <li>■ Logged out users will be redirected to the questionnaire page.</li> <li>■ Revisiting one of the backend pages after logging out will cause an authentication dialog to appear.</li> </ul>

### User Section

Title:	Filling out a questionnaire (FR12, FR15, FR17, FR18, FR20)
Description:	As a user, I need the ability to view and complete the latest published questionnaire so I can get a project recommendation.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Visiting the questionnaire page will display the latest published questionnaire.</li> <li>■ A unique 13 character ID will be appended to the page URL if a valid one does not exist.</li> <li>■ Form fields will validate onblur or onclick.</li> <li>■ The next section will appear below the current one when the Next Section button is clicked if all required form fields have been filled out.</li> <li>■ If a required form field has not been filled out, it will be highlighted and scrolled to.</li> <li>■ Form submission will be cancelled if there are any validation errors.</li> <li>■ The error highlighting will disappear once the question is answered.</li> <li>■ If all form fields have been filled out and the Finish button has been clicked, the user will be redirected to the questionnaire results page.</li> </ul>

Title:	Getting more information about a question (FR14)
Description:	As a user, I need the ability to view a detailed explanation of a question to avoid misunderstandings that can potentially affect my responses.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Clicking on the more information button while in mobile phone mode will pop up a modal with the question details.</li> <li>■ Clicking on the more information button while in tablet or desktop mode will pop up a popover with the question details.</li> <li>■ Clicking on the more information button while JavaScript is disabled will pop up a CSS3 modal with the question details.</li> <li>■ The modal will close when the user either clicks the close button or clicks outside the modal.</li> <li>■ The popover will close when the user either clicks outside the popover or clicks on the more information button for another question.</li> </ul>

Title:	Adding optional section notes (FR16)
Description:	As a user, I need the ability to optionally add notes at the end of each section to remind myself the reasons as to why I made a certain choice for a question.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ The contents of each notes textbox will be saved when either the Next Section or Finish buttons are clicked.</li> <li>■ The saved notes will be loaded into their respective notes textbox when the questionnaire is revisited.</li> </ul>

Title:	Saving a questionnaire/questionnaire responses to a PDF (FR21)
Description:	As a user, I need the ability to save the questionnaire/questionnaire responses so I can complete/reference it offline.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ A PDF version of the questionnaire will be created when the print button is clicked on the questionnaire page.</li> <li>■ A PDF version of the questionnaire responses will be created when the print button is clicked on the questionnaire results page.</li> <li>■ When revisiting an old questionnaire, a PDF version of the old questionnaire will be created instead of the current one.</li> <li>■ The Ctrl+P and Command+P keyboard shortcuts will trigger the PDF generation.</li> </ul>

Title:	Revisiting a questionnaire (FR12, FR13, FR15, FR17, FR18)
Description:	As a user, I need the ability to revisit questionnaire so I can review my responses or continue filling out the questionnaire.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Visiting the questionnaire page with a valid 13-character answer set ID will display the saved questionnaire responses.</li> <li>■ If the answer set ID is invalid, the user will be redirected to the questionnaire page with a newly generated answer set ID.</li> <li>■ If the questionnaire is complete, all questionnaire sections will be</li> </ul>

	<p>unhidden.</p> <ul style="list-style-type: none"> <li>■ If the questionnaire is incomplete, the page will scroll to the first unanswered question or incomplete section.</li> <li>■ Clicking on the Next Section or Finish buttons will allow the questionnaire to be validated and submitted as usual.</li> </ul>
--	--

Title:	Viewing questionnaire results (FR19)
Description:	As a user, I need the ability to view a completed questionnaire’s results so I can see which project methodology I should choose.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ Visiting the questionnaire results page with a valid answer set ID that corresponds to a completed questionnaire will display the recommended methodologies for the submitted responses along with the submitted responses.</li> <li>■ If the answer set ID is invalid or it corresponds to an incomplete questionnaire, the user will be redirected to the questionnaire page with the supplied answer set ID or with a new answer set ID if the one supplied is invalid.</li> <li>■ The questionnaire results page will show the methodology descriptions of the highest scoring categories.</li> </ul>

Title:	Emailing questionnaire results (FR22)
Description:	As a user, I need the ability to email the questionnaire results for safekeeping.
Acceptance Criteria:	<ul style="list-style-type: none"> <li>■ If a required form field has not been filled out, it will be highlighted.</li> <li>■ If an email form field contains an invalid email address, it will be highlighted.</li> <li>■ Form submission will be cancelled if there are any validation errors.</li> <li>■ The error highlighting will disappear once a required field has been filled out or a valid email address has been provided.</li> <li>■ If all form fields have been filled out and the Send button is clicked, the form will be submitted.</li> <li>■ A success message will appear if the email has been sent successfully.</li> <li>■ An error message will appear if the email could not be sent (e.g. API error).</li> <li>■ The recipient will receive an HTML formatted email with the recommended methodologies and questionnaire responses.</li> </ul>

# Development Details

## Methodology

This project consisted of the following phases:

- Gather requirements
- Create proposal
- Do visual design
- Perform usability testing with mockups
- Create database schema
- Setup development environment
- Create page templates
- Enter initial data
- Hookup backend
- Testing and bug fixes
- Do manual browser compatibility checks with beta site
- Deploy on live site
- Create project report
- Obtain final client signoff

### Gather Requirements

The client was asked to clarify the project's purpose, intended users, and requirements.

### Create Proposal

A project proposal describing the project's goals, scope, and methodology was created and submitted to the project committee.

### Do Visual Design

High-fidelity mock-ups for the desktop and phone screen sizes are created and are then modified based on client feedback. See Appendix A– Design Process & Mock-ups for the wireframe mock-ups.

### Perform usability testing with mock-ups

Three sets of tests were created in the usability test service Usabilla (<https://usabilla.com/>) using the mock-ups generated in the previous step: the Desktop test, the Tablet test and the Smartphone test. In each test, participants were presented with a screenshot and asked a question that mimics a task a questionnaire taker will perform on the web app.

Two types of questions were used: “click-based” and “answer-based”. A typical click-based question would be “Where will you click to name the project?”. Participants answered the question by clicking on the screenshot. Other questions asked participants to provide comments and feedback on the screenshot. A sample question of this type is “What do you expect will happen when the “Next Section” button is clicked?”

Using the feedback, the design of the web app was adjusted slightly before I moved into implementation.

### **Create Database Schema**

A database schema was created to allow for version control so older questionnaire versions will continue to work and be accessible as newer questionnaires are created.

### **Setup Development Environment**

MAMP 2.1.2, which bundles Apache HTTP Server 2.2.23, PHP 5.4.10, MySQL 5.5.29, and phpMyAdmin 3.5.5 was installed.

### **Create Page Templates**

Twitter Bootstrap (<http://getbootstrap.com>) is a front-end framework that allows for rapid development of mobile friendly website and web applications. The base Twitter Bootstrap 2.3.2 website template was modified based on the wireframe mock-ups created earlier. The desktop responsive design was created first followed by the tablet (desktop 800x600 wireframe), and finally the phone. The administrator backend page template is also created at this point.

### **Enter Initial Data**

See Appendix B – Database Tables with Data for the database schema with sample data.

### **Hook up Backend**

The static webpage were modified to include PHP tags to fetch sample data from the MySQL database. The MeekroDB library (<http://www.meekro.com/>) was used to provide secure access to MySQL data using PHP. The use of this library ensures the web application is safe from SQL injections and free of database related security issues. The questionnaire page loads the latest questionnaire version and the administrator backend allows the administrator to modify the questionnaire using the web interface.

### **Testing and Bug Fixes**

The backend was checked to ensure the web interface displays the correct questionnaire data and saves the data in the correct format. The questionnaire page is checked to ensure that the latest questionnaire is displayed unless a valid saved questionnaire ID is provided as a URL parameter. The questionnaire results page is checked to ensure that the questionnaire question scores are calculated correctly and the correct methodologies are recommended.

Functional test cases for the admin and user portions can be found in Appendix F – Test Cases & Results of Testing.

### **Do Manual Browser Compatibility Checks**

The website frontend layout was tweaked to display correctly on Internet Explorer 8, 9, and 10, Firefox 3 and 19, Safari 4 and 6, and Chrome 2 and 28 using PHP browser user agent detection. Unsupported browsers such as Internet Explorer 6 and 7, Firefox 2, and Safari 3 are shown a browser upgrade message.

### **Deploy on Live Site**

The final website code will be uploaded and deployed on the production web server likely through SSH, FTP, or CPanel. The databases will be created through SSH or phpMyAdmin.

**Create Project Report**

A project report detailing the project's background, design, innovative aspects, and future enhancements is created and submitted to the project committee.

**Obtain Final Client Signoff**

The client signoff stating that the project work was done according to the project's requirements and delivered.

**Project Schedule**

The following table shows each of the project's tasks with the actual hours included.

<b>Task</b>	<b>Description</b>	<b>Est. Hours</b>	<b>Actual Hours</b>
<b>Project Management</b>			
Gather requirements	Requirements from client	5	3
	Show client proposal		
	Clarify questions		
Client communications	Get client feedback regarding project deliverables	20	10
	Make changes based on feedback		
Create proposal	Outline requirements	30	40
	Describe purpose of project		
	Steps involved to achieve goal		
	Describe how success is measured		
<b>Design</b>			
Wireframes	Wireframe for admin section and user facing section	35	65
Visual design	Determine fonts, layout, and colour scheme for admin and user sections	35	50
Database schema	Design ER diagram and schema for MySQL database to store:	10	8
	Questionnaire data as designed by the admin		
	User submitted questionnaire		

## answers

**Development**

Environment setup	Install Apache web server, PHP, and MySQL	3	1
Create schema	Create database tables based on ER diagram	2	2
Create templates	Create HTML and CSS framework for admin and user sections	50	30
Backend hookup	Connect the front-end UI with the database to make the tool functional	30	55

**Usability Testing**

Initial mock-ups testing	Obtain user feedback early in the process by testing conceptual designs using Usabilla, then make changes to address issues and retest in an iterative fashion for several rounds.	40	15
Final testing of actual tool	Obtain user feedback on the final product. If necessary, make tweaks to resolve additional usability problems that remain.	50	N/A
Usability Report	Create report summarizing usability findings and feedback from initial usability tests	5	8

**Automated Testing**

Enter initial data	Enter dummy data into the database	3	3
Test Environment Setup	Set up Windows 7 and Mac OS X 10.8 virtual machines in VMware and install Selenium Remote Control on the host machine	10	N/A
Test Case Creation	Create test cases to test the functionality of the admin and user-facing sections of the web site  Create scripts in Selenium IDE for the test cases.  Revise test cases for later tests	50	N/A

**Manual Testing**

General Testing	Perform manual testing to ensure the major features are implemented and working.	15	20
Browser compatibility	Ensure the admin section works for the latest version of Firefox  Ensuring user sections works across the following browsers:  Windows  Latest versions of Firefox and Chrome, Internet Explorer 7, 8, 9, and 10  Mac OS X  Latest version of Firefox, Chrome, and Safari	20	10
Email client compatibility	Ensure emails sent can be viewed correctly for the following mailboxes:  Outlook, Thunderbird, Gmail, Hotmail, and Yahoo	10	5
<b>Bug fixes</b>	Fix bugs from automated testing, general testing, browser compatibility tests, and email client compatibility tests.	20	10
<b>Live site deployment</b>	Upload final site onto actual hosting server  Do final tests to ensure site functions correctly	5	N/A
<b>User documentations</b>	Create documentation to show how to setup and use the admin sections	15	5
<b>Final Report</b>	Summarize all work done on the project and reflect on possible improvements	30	150
<b>Total Hours.</b>		<b>453</b>	<b>580</b>

The project schedule as shown in Figure 4 shows the order of each of the phases mentioned above.

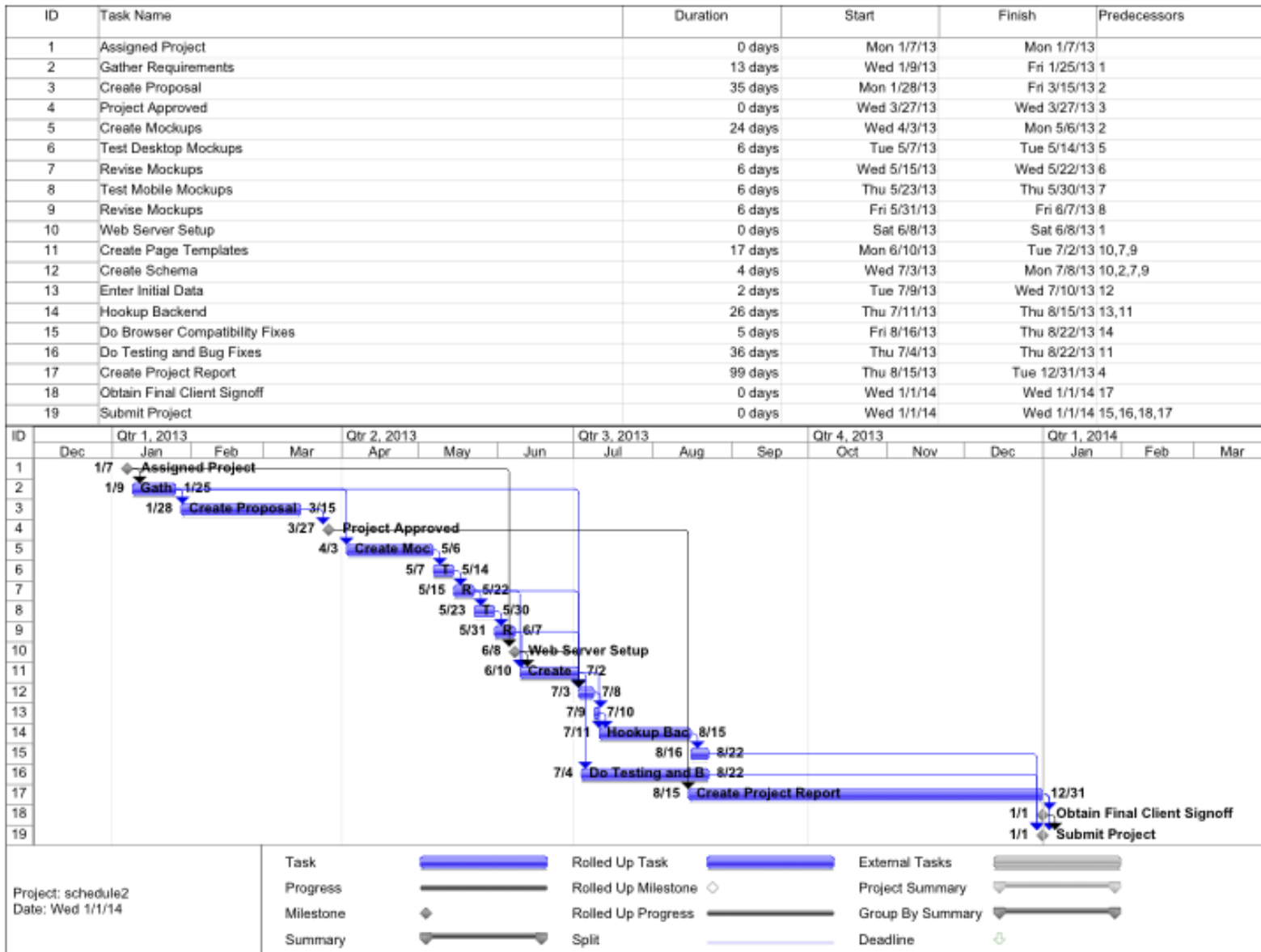


Figure 4 – Gantt chart of the project schedule

## Entity Relationship Diagram

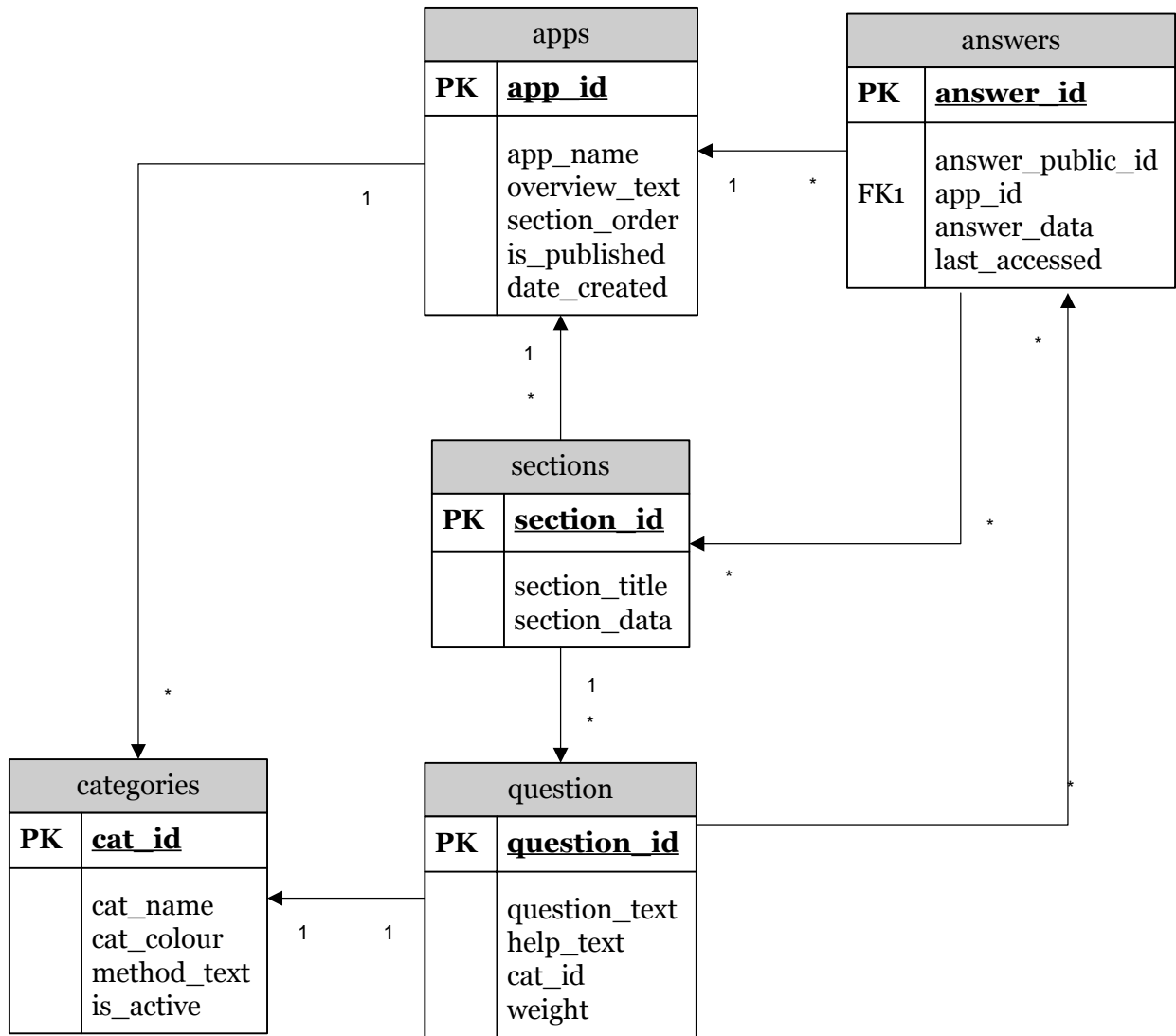


Figure 5 – The decision\_tool database entity relationship diagram

### apps

app_id	int(20) PK AUTO_INCREMENT
app_name	tinytext
overview_text	text
section_order	text (JSON array storing section ids)
is_published	tinyint(1)
date_created	timestamp

**sections**

section_id	int(20) PK AUTO_INCREMENT
section_title	tinytext
section_data	longtext (Two dimensional JSON array storing the question descriptions, question information texts, question category ids, and question weights separated by question id)

**categories**

cat_id	int(20) PK AUTO_INCREMENT
cat_name	tinytext
cat_colour	tinytext
method_text	text
is_active	tinyint(1)

**answers**

answer_id	int(20) PK AUTO_INCREMENT
answer_public_id	char(13) (13 character id generated by uniqid)
app_id	int(20) FK references app_id in apps
answer_data	longtext (Three dimensional JSON array storing the project name, and questions and notes separated by section id)
last_accessed	timestamp

Each questionnaire is represented as an app. Each record in the apps table stores the questionnaire name, the overview text (a description of the questionnaire), and the section order (an array of section ids). The `is_published` flag determines if the questionnaire is viewable to the public or only through the admin backend. The `date_created` timestamp will be used in the future for version tracking.

A questionnaire consists of multiple questions that are grouped into sections. Each record in the sections table stores the section title and the section data (an array of question text, information text, category ids, and question weights). As an array, `section_data` stores the question order. The question text contains the question, the help text contains a through explanation of a question when a user clicks the information icon beside a question, the category id references the category id in the category table that the question is assigned to, the question weight is the question weighting that is used if a question's response is deemed more important than another question's response.

As mentioned above, questions are assigned categories. These categories are stored in the categories table. This table contains the category id, which is referenced in the sections table, the category (methodology) name, and the method (methodology) text. The `cat_colour` column will be used in the future to colour code questions for easier understanding of how the results are calculated. The `is_active` flag is used to determine which categories to clone when a new questionnaire revision is created. Only active categories are cloned.

When a user fills out the questionnaire, the questionnaire responses are saved in the answers table. This table contains the answer public id (the 13 character id in the questionnaire results link), the app id (a foreign key that references the `app_id` in the apps table), and the answer data (a JSON array storing the project name, and questions and notes separated by section id). The answer data is overwritten every time the Next Section button or Finish button is clicked and the form passes validation. The `last_accessed` timestamp will be used in the future to determine which unused answer sets should be purged.

## **Business Rules**

- The users can only view the latest published questionnaire except when a valid answer set id is specified.
- The administrator can only view and edit the latest published or unpublished questionnaire through the backend.
- Old questionnaires must remain accessible when a valid answer set id is specified.
- Users can only view the questionnaire results page when the questionnaire is complete.
- An app must have a section order (an array of section ids).
- An app must have at least one section.
- A section must have at least one question.
- A section can contain an unlimited number of questions.
- A section must have a title.

- A question must be part of a section.
- A question must have a question description.
- A question must have an assigned weight (default is 1).
- A category must have a name.
- A category must have method text.
- Answer sets are only referenced by the `answer_public_id`.

## **Database Tables with Sample Data**

Please refer to Appendix B – Database Tables with Data.

## **Message Sequence Chart**

The following figures describes the series of steps the decision tool backend has to perform to add, edit, remove, or rearrange sections, create or edit categories, and edit the questionnaire overview. These steps allow the backend to keep older, published versions of the questionnaire intact and accessible as newer versions of the questionnaire are created.

### Creating a new section for a published questionnaire (FR3)

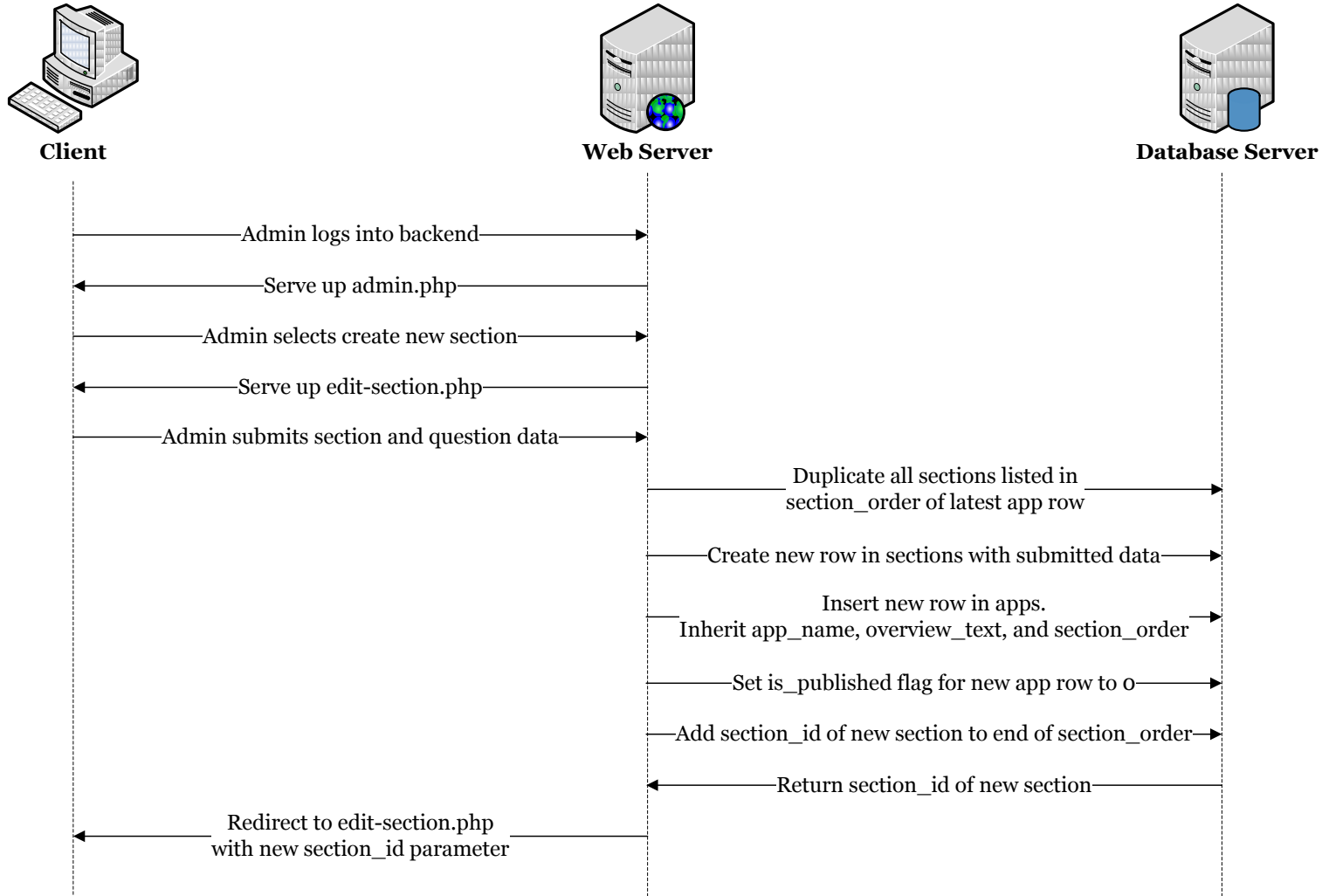


Figure 6 – Message sequence for creating a new section for a published questionnaire

### Creating a new section for an unpublished questionnaire (FR3)

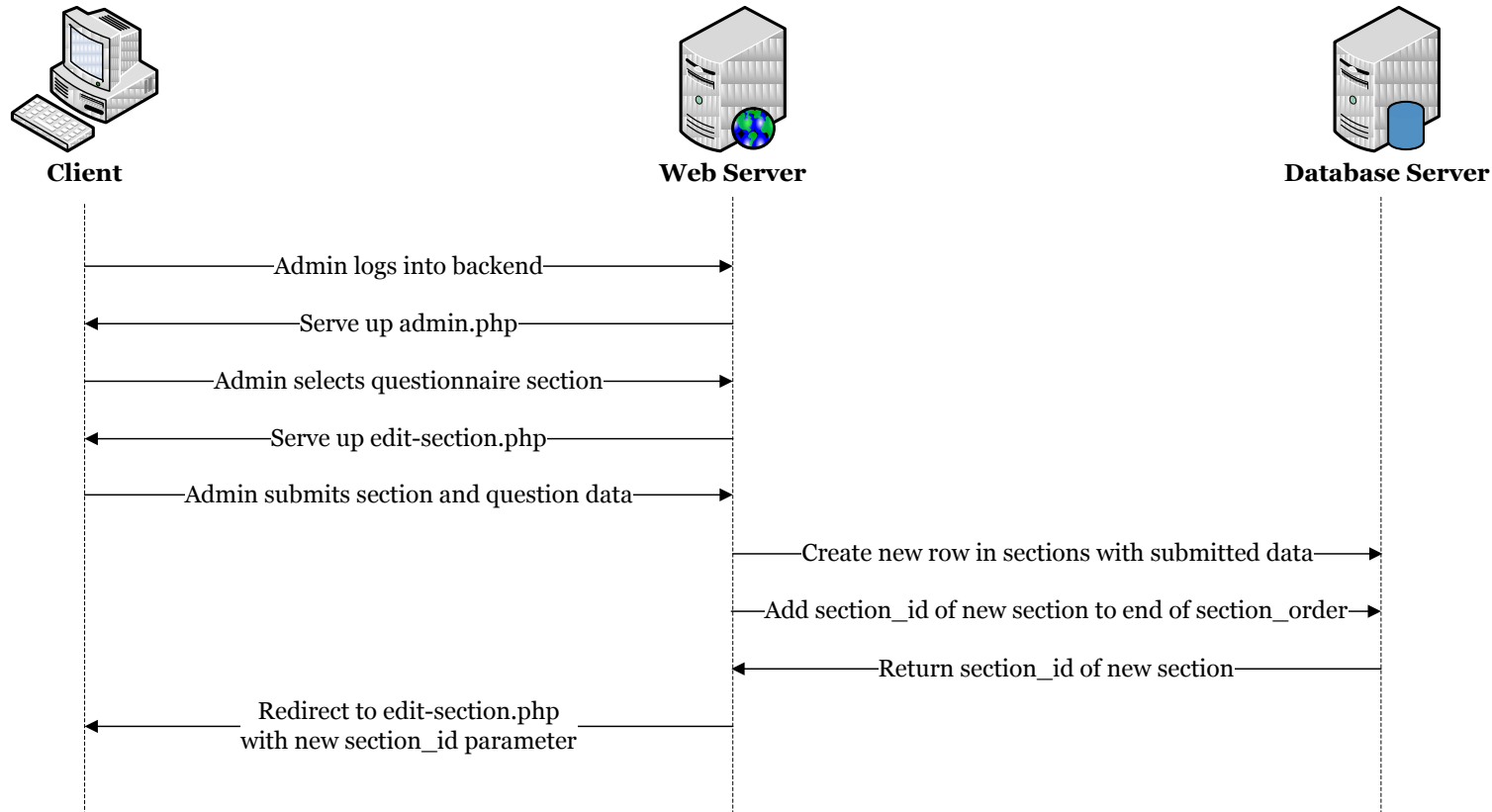
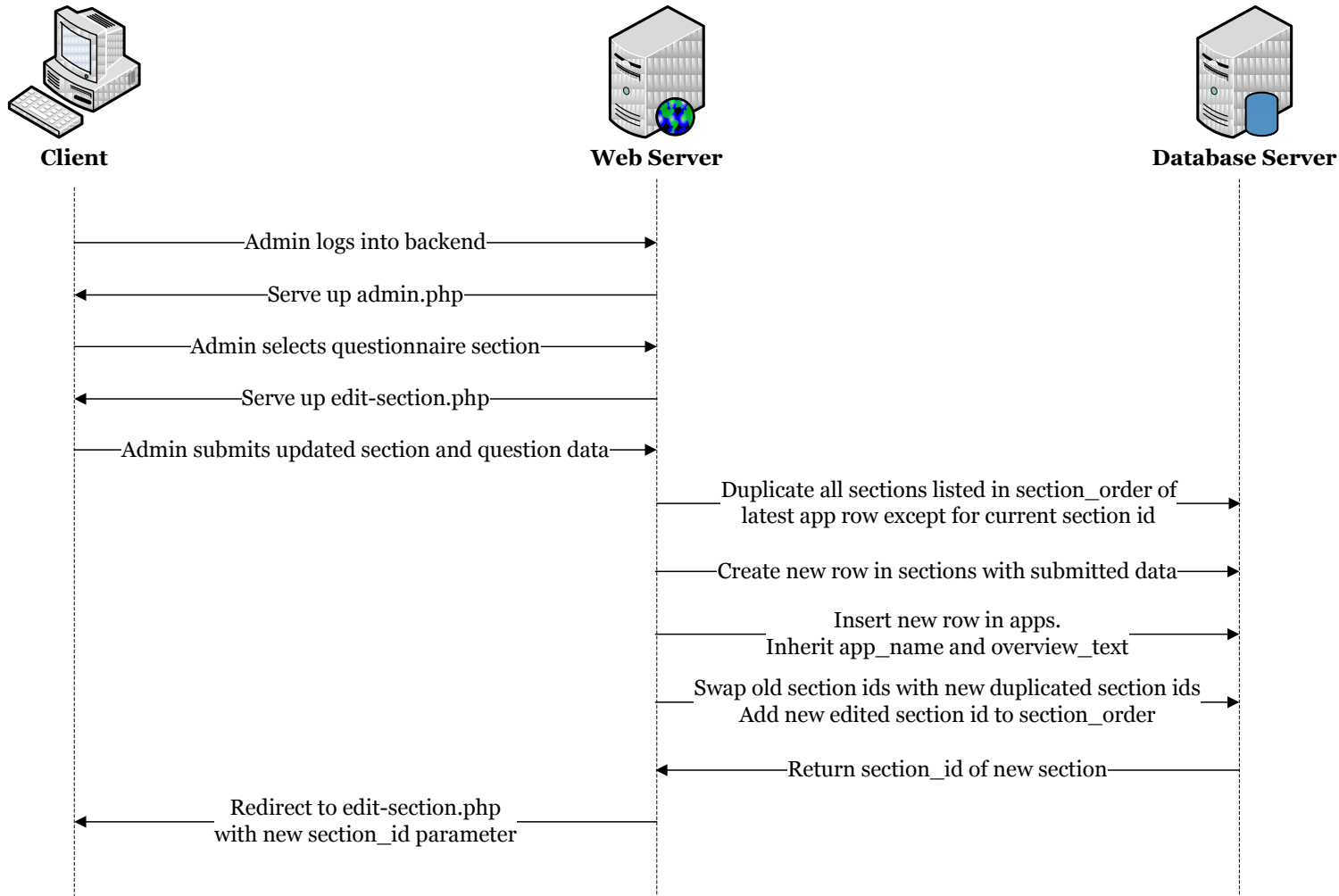


Figure 7 – Message sequence for creating a new section for an unpublished questionnaire

**Editing an existing section for a published questionnaire (FR3)**



**Figure 8 – Message sequence for editing an existing section of a published questionnaire**

### Editing an existing section for an unpublished questionnaire (FR3)

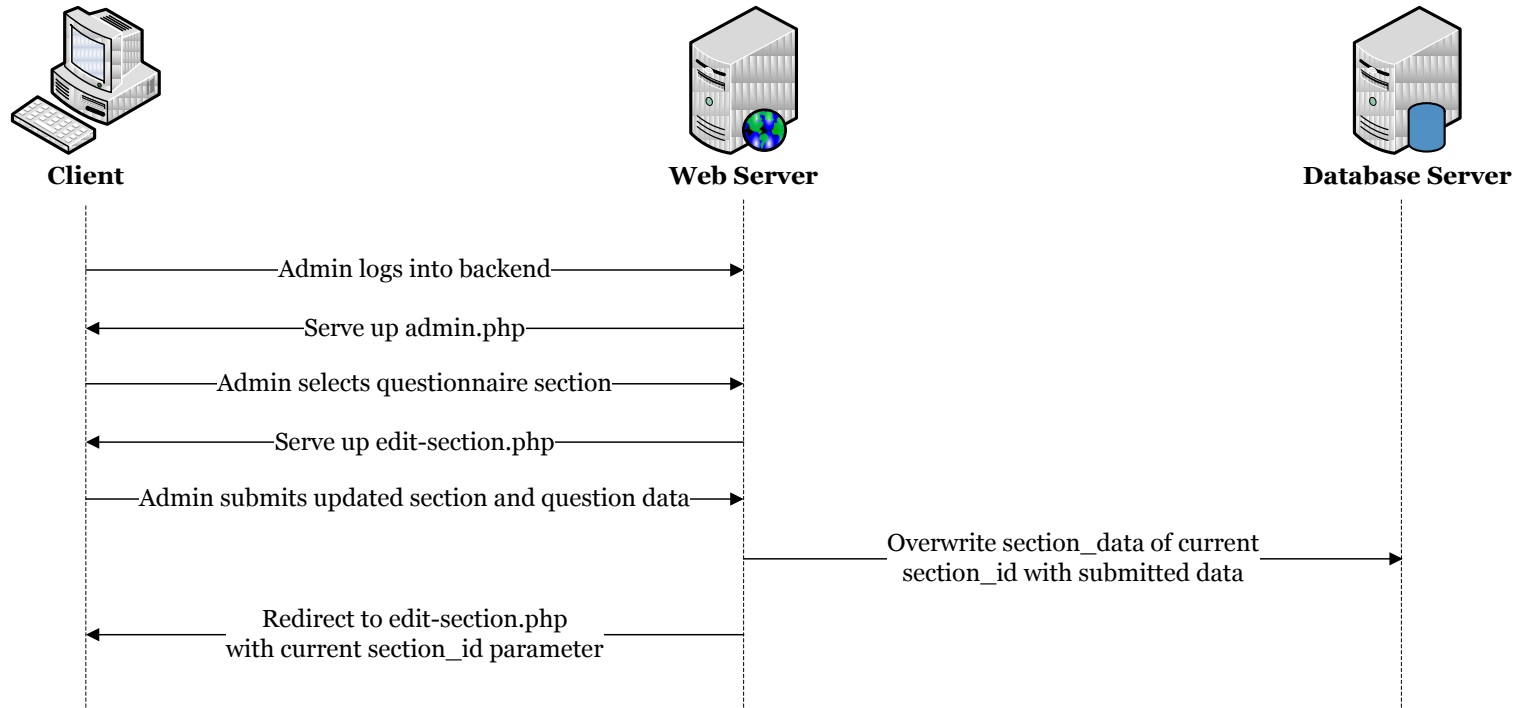
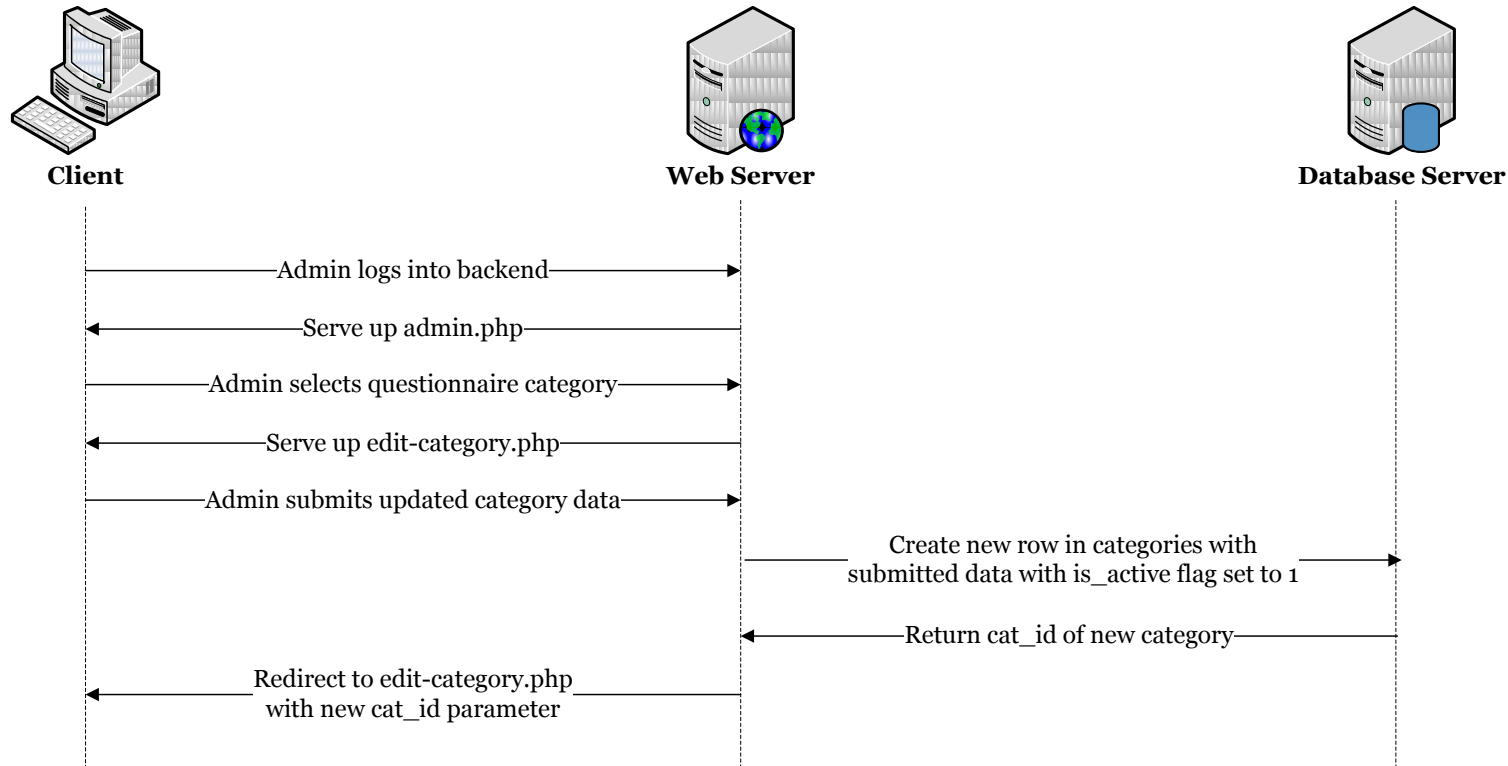


Figure 9 – Message sequence for editing an existing section for an unpublished questionnaire

**Creating a new category for a published/unpublished questionnaire (FR2)**



**Figure 10 – Message sequence for creating a new category for a published/unpublished questionnaire**

### Editing an existing category for a published questionnaire (FR2)

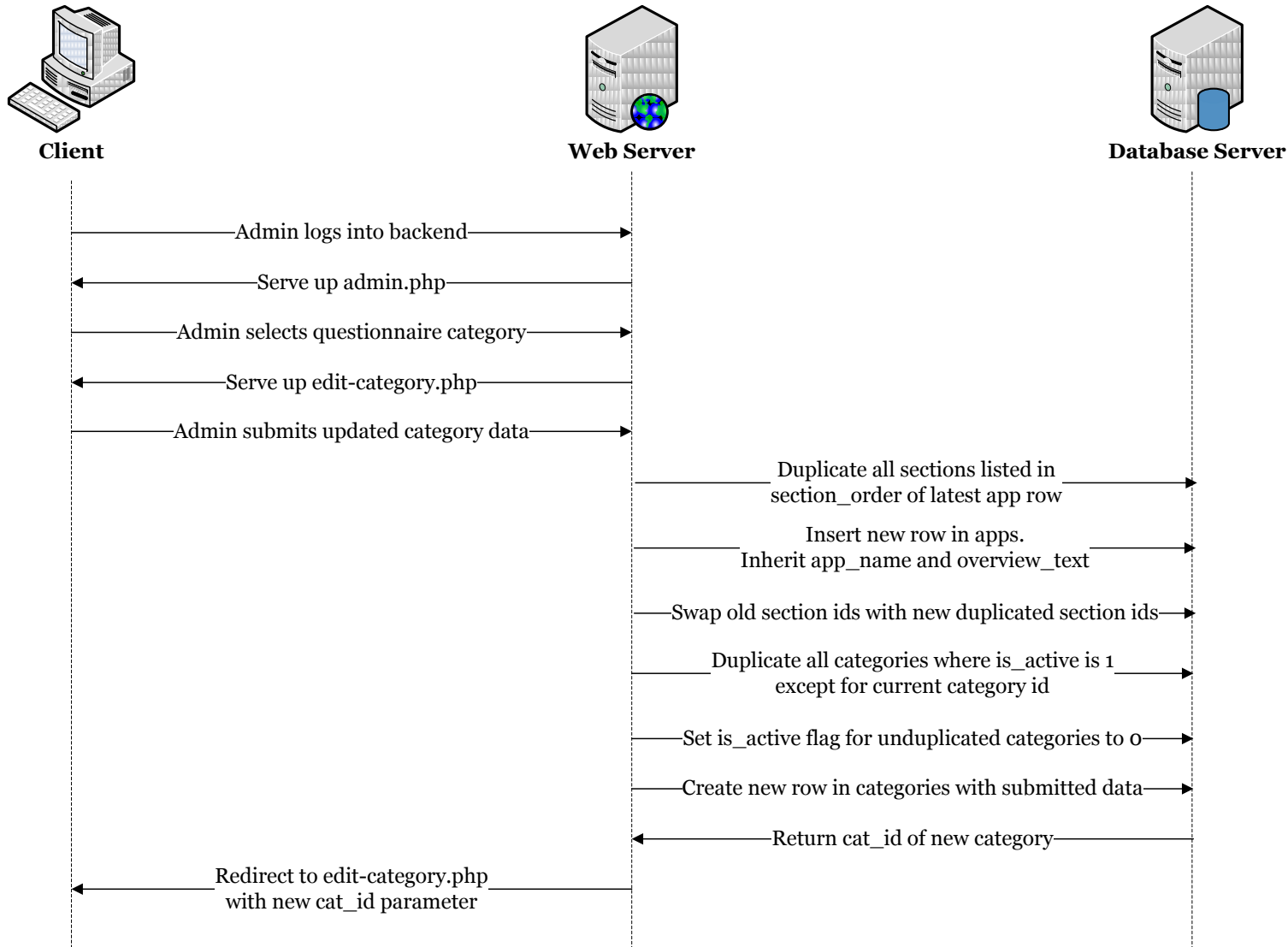
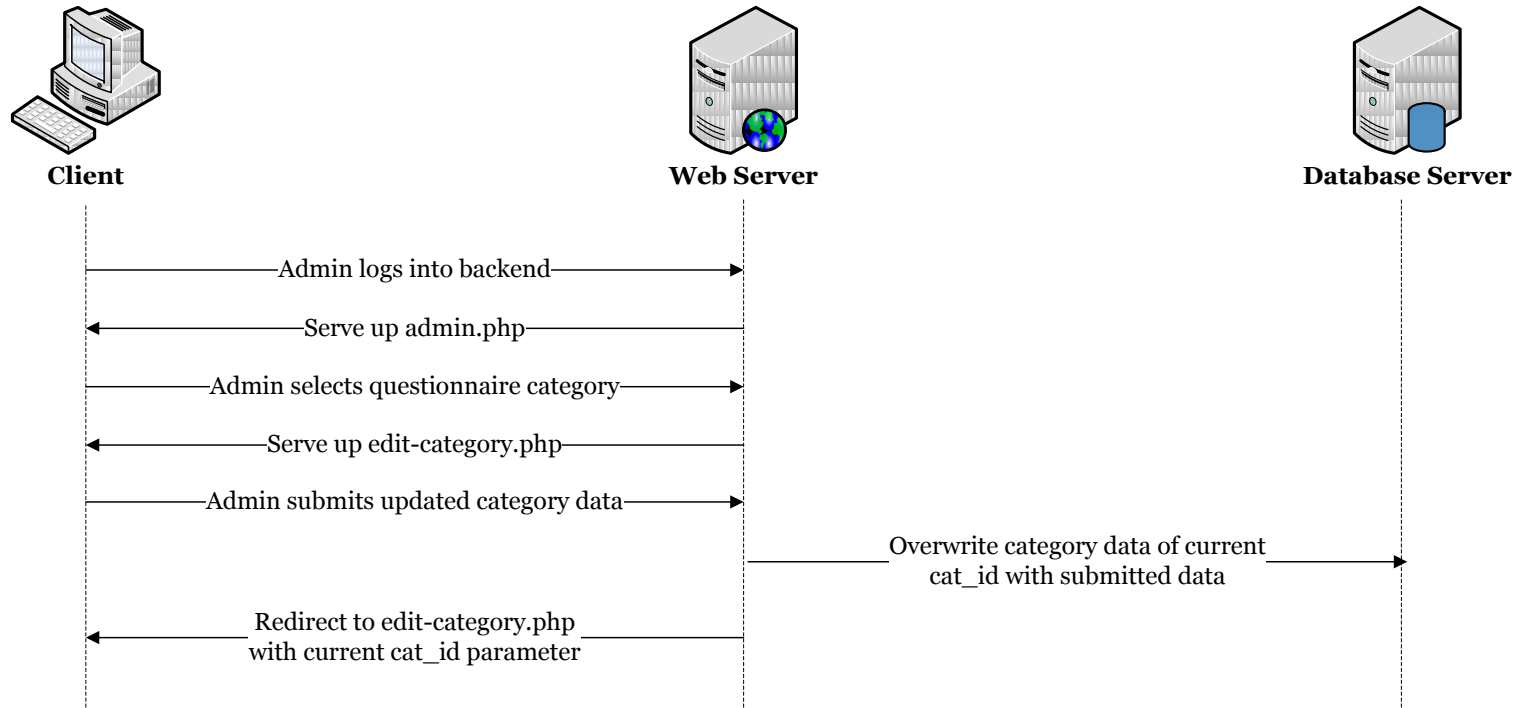


Figure 11 – Message sequence for editing an existing category for a published questionnaire

**Editing an existing category for an unpublished questionnaire (FR2)**



**Figure 12 – Message sequence for editing an existing category for an unpublished questionnaire**

### Deleting a section for a published questionnaire (FR3)

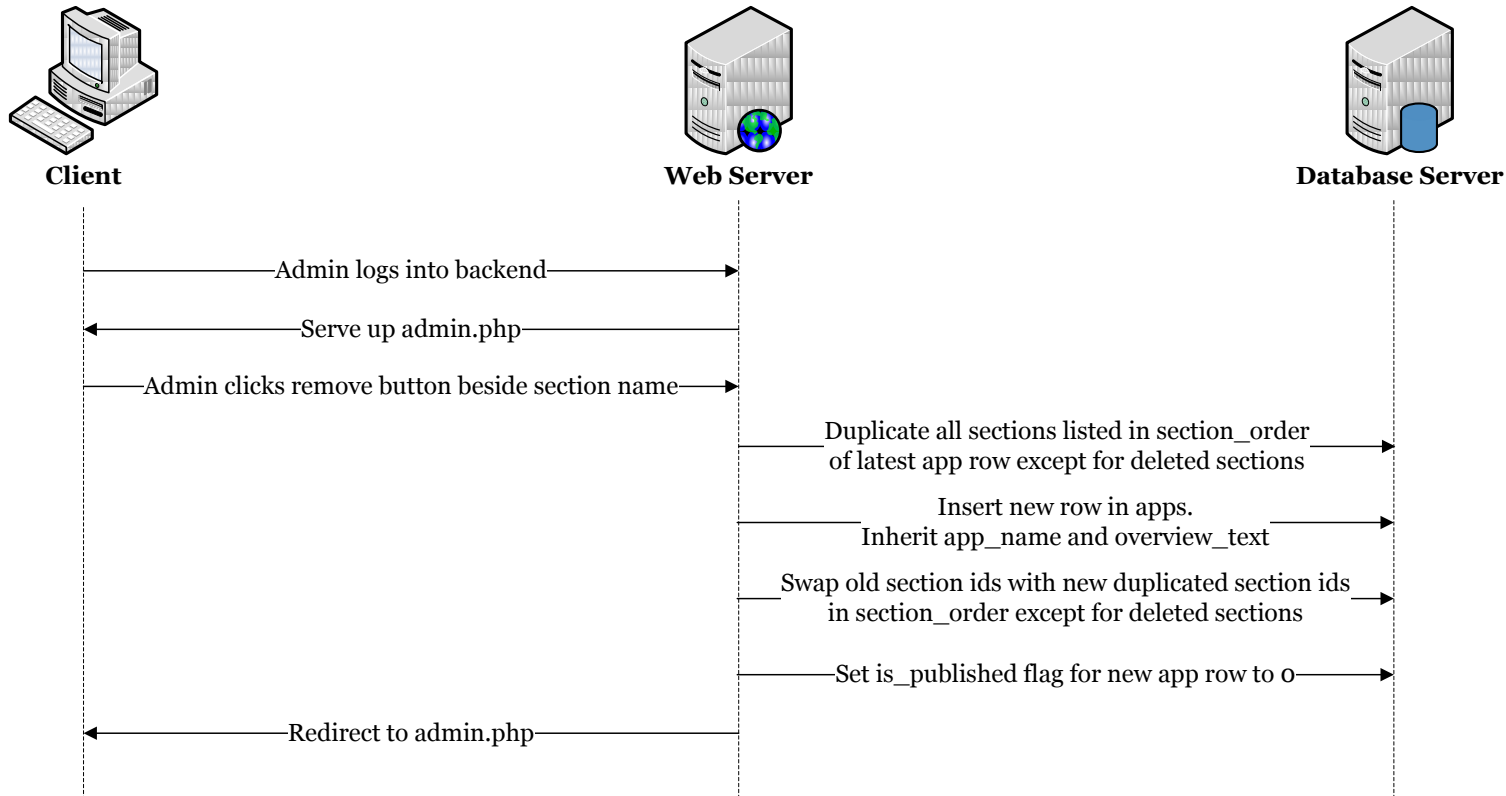
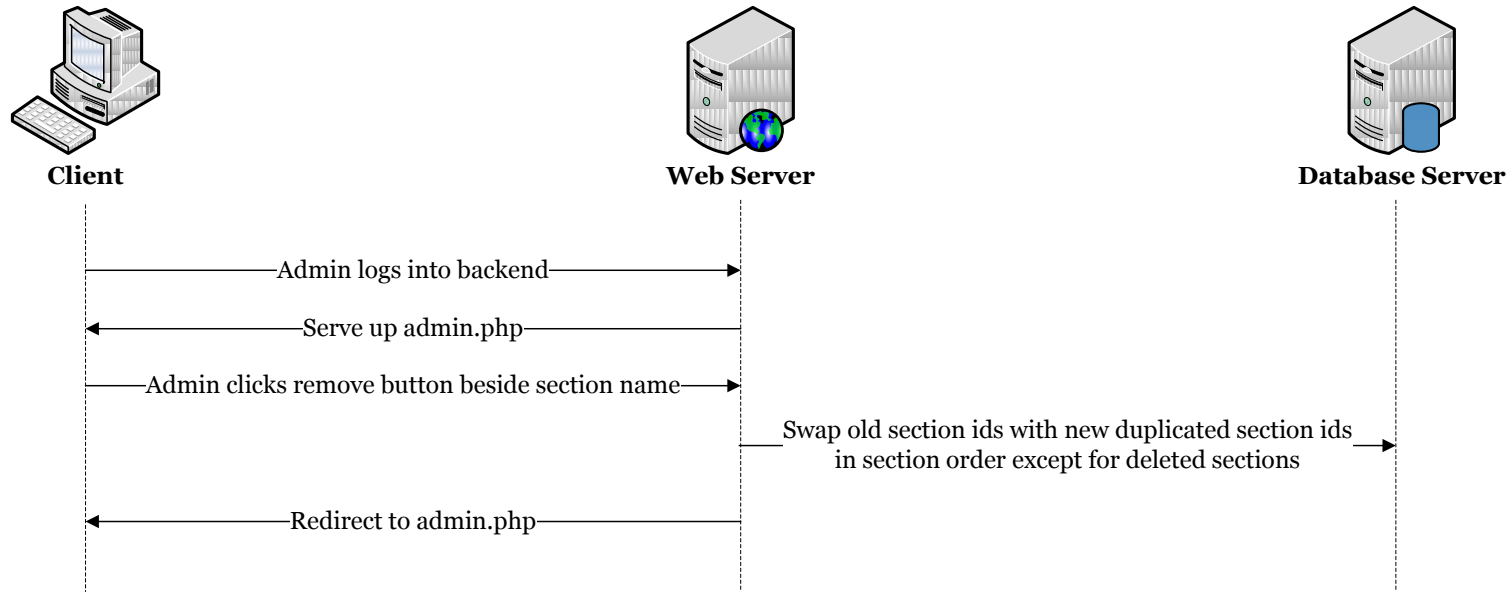


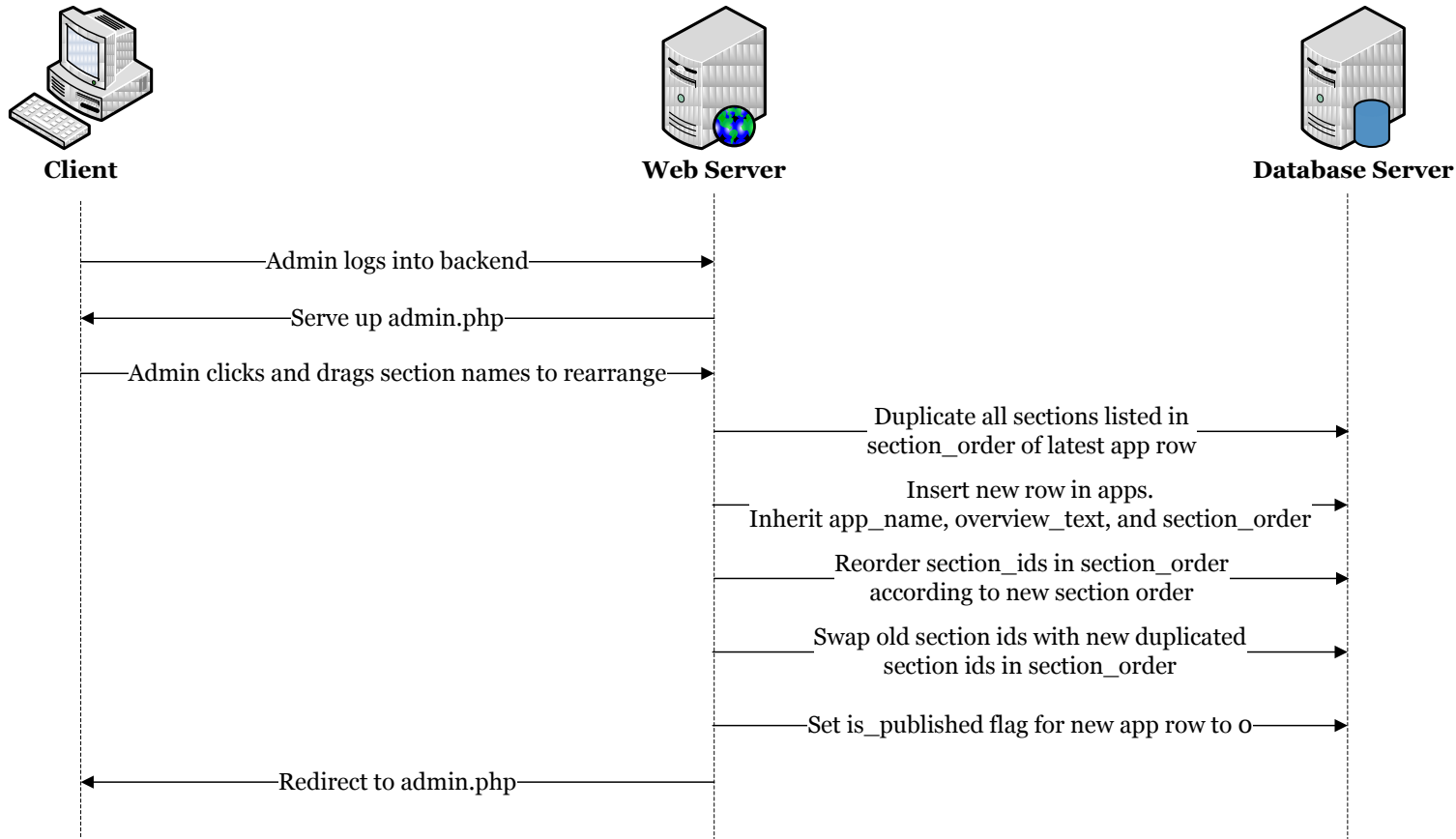
Figure 13 – Message sequence for deleting a section for a published questionnaire

**Deleting a section for an unpublished questionnaire (FR3)**



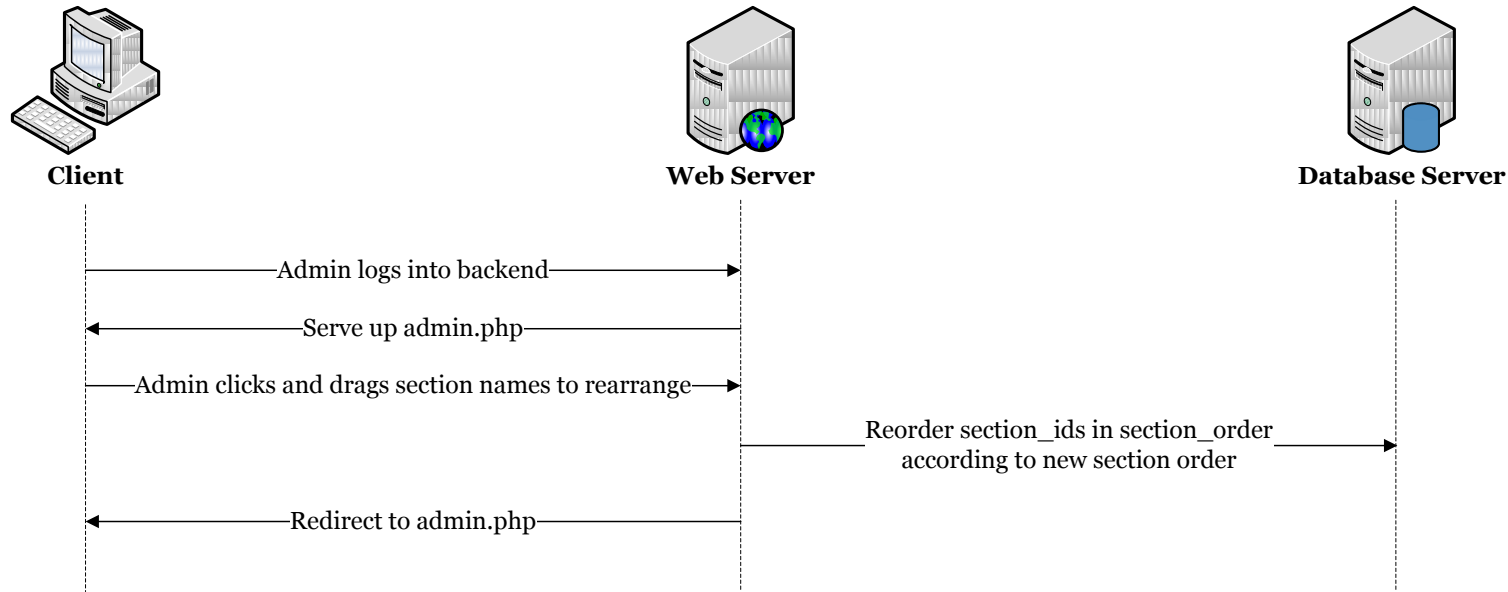
**Figure 14 – Message sequence for deleting a section for an unpublished questionnaire**

**Rearranging sections for a published questionnaire (FR5)**



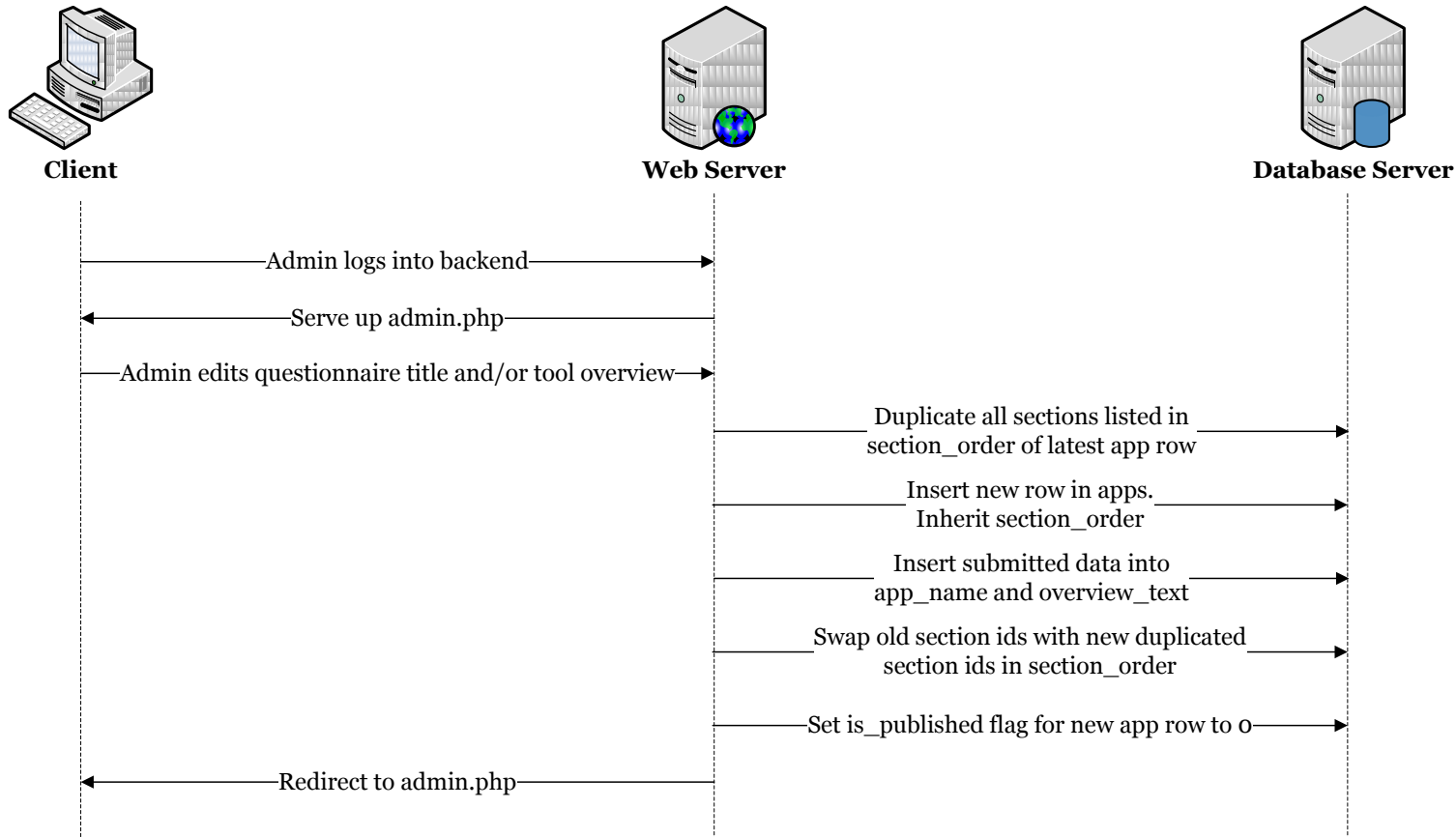
**Figure 15 – Message sequence for rearranging sections for a published questionnaire**

**Rearranging sections for an unpublished questionnaire (FR5)**



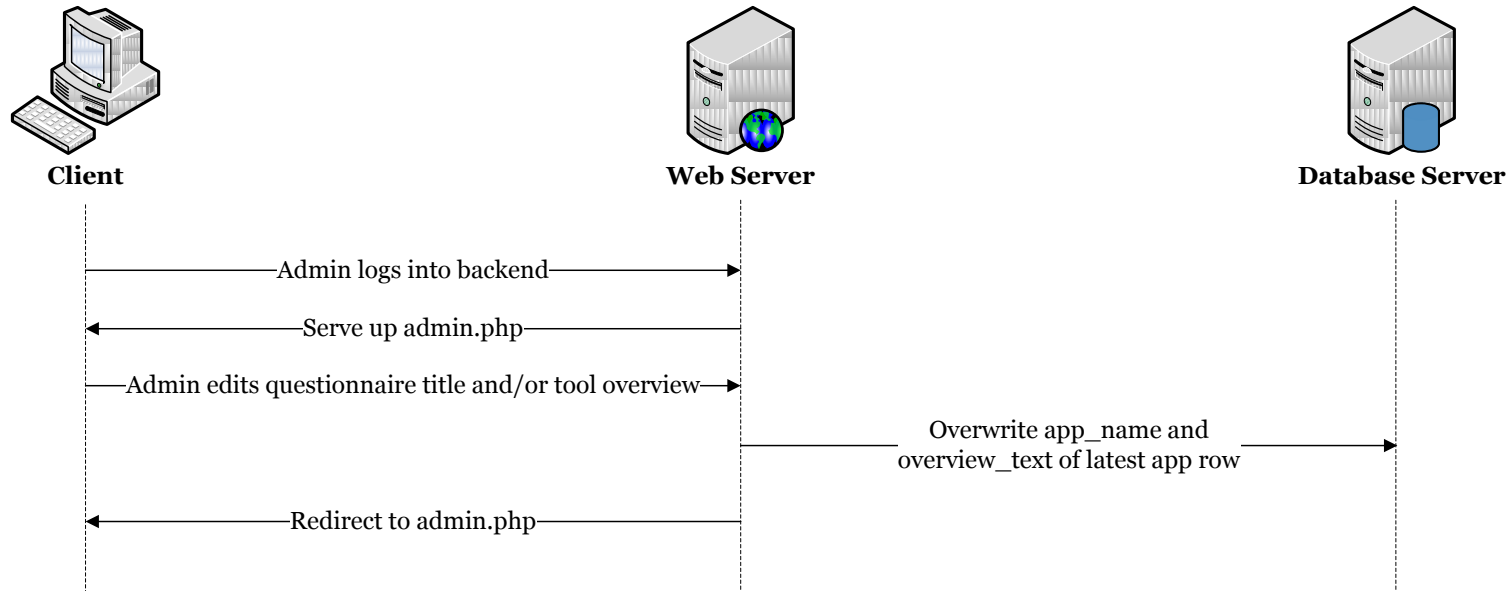
**Figure 16 - Message sequence for rearranging sections for an unpublished questionnaire**

**Editing questionnaire overview for a published questionnaire (FR9, FR10)**



**Figure 17 – Message sequence for editing questionnaire overview for a published questionnaire**

**Editing questionnaire overview for an unpublished questionnaire (FR9, FR10)**



**Figure 18 – Message sequence for editing questionnaire overview for an unpublished questionnaire**

## User Manual

Please refer to Appendix C – User Manual.

## Implementation Implications

The website front end is designed and tested to run on Internet Explorer 8 or later, Firefox 3 or later, Safari 4 or later, and Chrome 2 or later with JavaScript enabled. The responsive website layout is designed to be used comfortably on screen resolutions of 320×480 pixels and above.

The questionnaire is currently parsed on page load and validated on submit using JavaScript. Although the JavaScript logic used to scroll to the first unanswered question/section on page load and the validation function used to highlight unanswered questions on submit needs to be refactored, the functions are good enough for now for a low traffic website and can be improved if the need arises.

To keep things simple, the questionnaire currently does not support the conditional branching of sections. In other words, all sections of the questionnaire need to be completed sequentially rather than selectively depending on the responses to the questions in previous section. If the needed later, the database schema allows nesting sections as the section order is stored as a JSON array.

The questionnaire's backend is currently secured by HTTP authentication with PHP. Although it is rudimentary, it serves its purpose for now. It is more flexible and secure than using an .htaccess and .htpasswd file because individual files rather than directories can be secured, login attempts can be limited, login data is encrypted and protected against replay attacks, and error messages can be customized.

The HTTP authentication dialog is shown in Figure 19. This authentication mechanism is almost identical to Apache's as HTTP authentication headers are resent by PHP until the user either successfully authenticates or the failed authentication limit is reached. If the user clicks the cancel button, they will receive an authentication required message and if the authentication limit is reached, they will keep receiving a forbidden message until the ban is lifted.

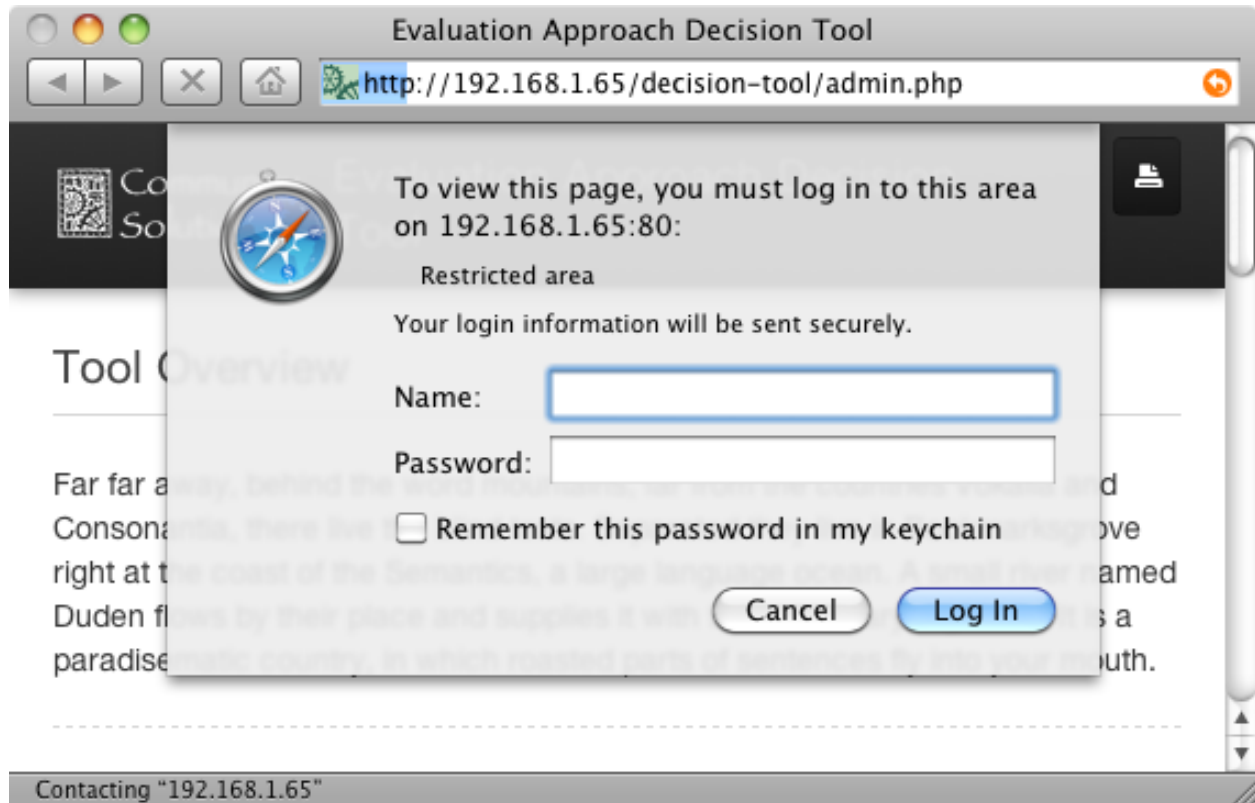


Figure 19 – HTTP authentication with PHP

## Use of new technologies

### Responsive Layout & Twitter Bootstrap

Since the front-end needs to be mobile-friendly, the new CSS @media queries were used extensively to create a “responsive” layout that adjusts the look and feel of the page depending on the width of the screen. @media queries acts as if statements. The styles within the query get applied if and only if the conditions specified at the beginning of the query is met. This allows us to improve the user experience by altering the website layout to better suit the user’s viewing device. In the example below, in desktop mode, buttons are set to regular size while in tablet and mobile phone mode, buttons are set to span the width of the screen.

```
/* Desktop large */
@media (min-width: 1200px) {
  .question .btn-group .btn-resize {
    padding: 4px 12px;
    font-size: 14px;
  }
}
/* Tablet */
@media (max-width: 767px) {
  .modal .modal-footer .btn-resize {
    display: block;
    width: 100%;

    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;

    padding: 11px 0;
    margin-left: 0;
    font-size: 17.5px;
  }

  .question .btn-group .btn-resize {
    padding: 11px 19px;
    font-size: 17.5px;
  }
}
```

**Figure 20 – A code snippet from docs.css showing how responsive layout styles are applied**

Twitter Bootstrap (<http://getbootstrap.com/>) was used as a method of speeding up the front-end development process. A key strength of Bootstrap is its mobile-friendly “responsive” scaffolding. The built-in grid system has already been set up to accommodate screen layout for desktop (layout width 1200+px), tablet (layout width 768px – 1200px) and phone (layout width less than 480px). This makes it much easier to tweak the layout to meet our mock-up design.

Bootstrap as a framework has also been tested across common browsers. This means there is significantly less work involved in browser testing as majority of browser related issues have already been address by the framework.

A downside of using this framework is that the web app currently has the look and feel of a bare-bone “Bootstrap” application. In the future, this can easily be skinned to suit the client’s branding requirements. Numerous pre-built “themes” are available for bootstrap on different marketplaces (e.g. <https://wrapbootstrap.com/>). This can be applied relatively quickly if needed.

The Bootstrap framework also includes some dynamic elements such as modals, popovers, tooltips, and accordions that are used throughout the website. I took advantage of it for displaying question explanations as popovers on tablet and desktop layouts (Figure 21), and used full-screen modals on phone layouts due to screen real-estate constraints (Figure 22).

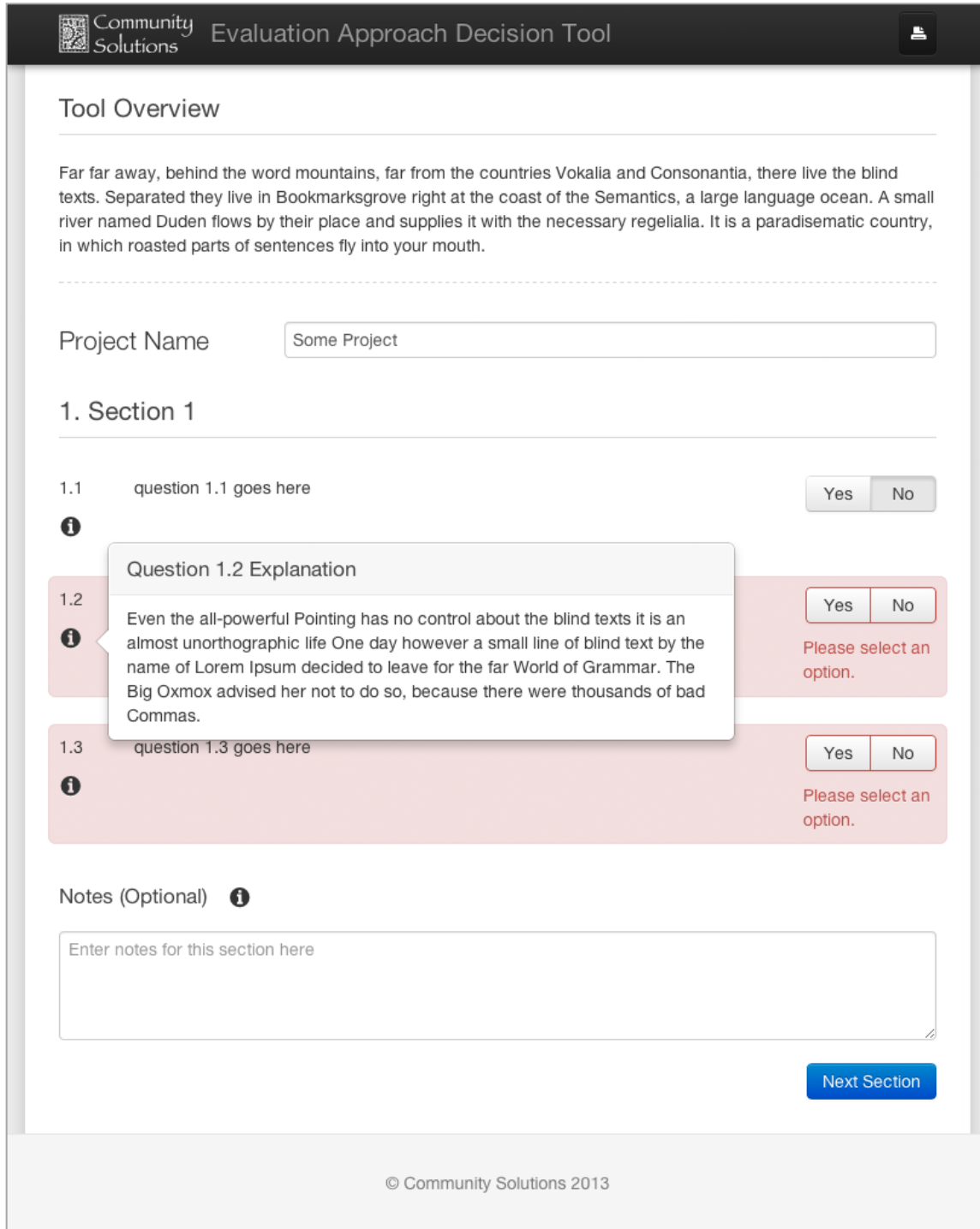


Figure 21 – Question information popover in phone to tablet size

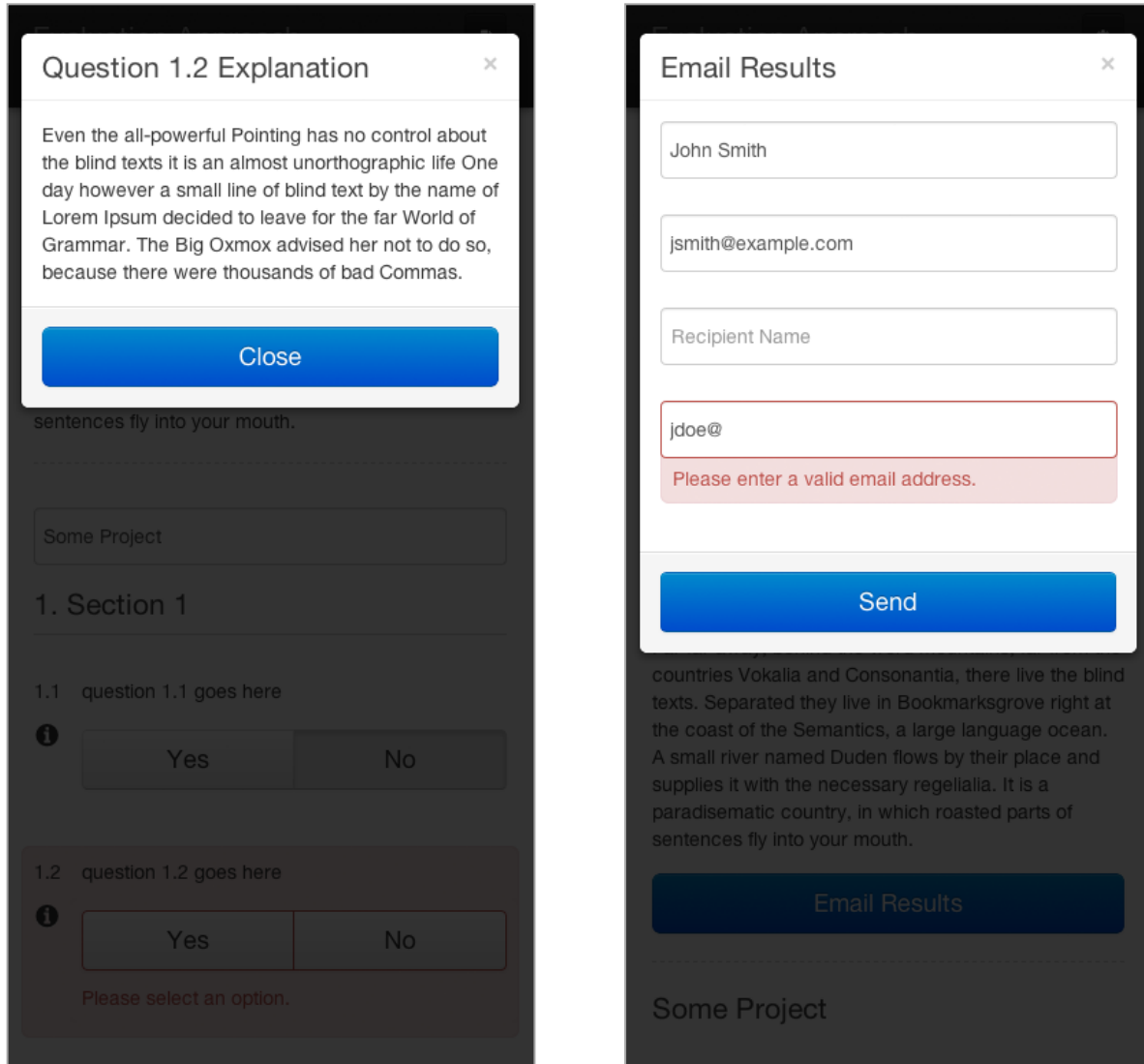


Figure 22 – Question information and email results modals in mobile phone size

### Clicking vs. Tapping

In addition to smaller screen real estate, users interact with smart phones and tablets by tapping using their fingers as oppose to clicking with the mouse pointer. Fingers are much less precise as compared to the mouse pointer when it comes to selecting items on the screen. Because of this, “Yes” “No” buttons are used instead of radio buttons on the tablet and phone layout (Figure 23, Figure 24). This specificity of user controls depending on the user’s device is only possible with the use of CSS @media queries.

## Evaluation Approach Decision Tool

### Tool Overview

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regellialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.

Some Project

### 1. Section 1

1.1 question 1.1 goes here

**i**

1.2 question 1.2 goes here

**i**

Please select an option.

1.3 question 1.3 goes here

**i**

Please select an option.

Notes (Optional) **i**

Enter notes for this section here

**Next Section**

**Figure 23 – Questionnaire page in mobile phone size**

Community Solutions Evaluation Approach Decision Tool

### Tool Overview

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelliala. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.

Project Name

### 1. Section 1

1.1 question 1.1 goes here

1.2 question 1.2 goes here    
Please select an option.

1.3 question 1.3 goes here    
Please select an option.

Notes (Optional)

© Community Solutions 2013

Figure 24 – Questionnaire in tablet size

## CSS3 Modals and Transitions

Experimental CSS3 modals are implemented for cases in which the Twitter Bootstrap modal does not appear due to JavaScript being disabled. These modals are used show the question information, the questionnaire results link, the email results form, and the give feedback form.

CSS3 animations are implemented to make transitions such as the showing and hiding of the CSS3 modals and of the form error boxes smooth.

This feature has been tested on Internet Explorer 10 and 11, Firefox 19, Safari 4 and 6, and Chrome 28.

The following code snippet shows how no-JavaScript styling is applied. There are different CSS styles declared for radio buttons with the no-js class and radio buttons without the no-js class. If JavaScript is disabled, all the radio buttons will have the no-js class styling. However, if JavaScript is enabled, jQuery removes the no-js class from each of the radio buttons when the page loads so all the radio buttons are moved off the screen.

```
<style>
  .question .no-js .radio-btn-lbl input {
    position: static;
    left: auto;
  }
  .question .radio-btn-lbl input {
    position: absolute;
    left: -99999999px;
  }
</style>
<script>
  /* Show styled "radio buttons" if JavaScript is enabled */
  if($("#main-contain .question .btn-group").hasClass("no-js")) {
    $("#main-contain .question .btn-group").removeClass("no-js");
  }
</script>
```

**Figure 25 – A code snippet from index.php showing how no-js styling is applied**

The following code snippet shows how the web application switches between using modals, popovers, and CSS3 modals depending on the screen width and if JavaScript is enabled.

JQuery and CSS3 modals for each question (Figure 26) that has information text are generated on page load and are hidden by default.

If JavaScript is enabled, when an information link is clicked (an onClick event), the JavaScript function show\_info() is run. The modal ID that is stored in the link's data-tag is passed to show\_info and if the screen width is under 767 pixels (mobile phone and tablet screens), popovers are disabled and the modal the ID that is passed in as a parameter is shown. If the screen width is above 767 pixels (desktop screens), popovers are enabled and a popover with the information stored in the data-content attribute of the information link is shown.

If JavaScript is disabled, when an information link is clicked, the CSS3 modal ID contained in the href attribute of the information link will open the CSS3 modal.

```

<script>
  <?php <!-- Question Modal -->
  if ( !empty($question['help']) ) { ?>
  <div id="<?php echo $q_id; ?>-info" class="modal hide fade"
  tabindex="-1" role="dialog" aria-labelledby="<?php echo $q_id;
  ?>-modal-label" aria-hidden="true">
    <div class="modal-header">
      <button type="button" class="close" data-
      dismiss="modal" aria-hidden="true">×</button>
      <h3 id="<?php echo $q_id; ?>-modal-label">Question
      <?php echo $i+1, '.', $j; ?> Explanation</h3>
    </div>
    <div class="modal-body">
      <?php echo $question['help']; ?>
    </div>
    <div class="modal-footer">
      <button class="btn btn-large btn-block btn-primary"
      data-dismiss="modal" aria-
      hidden="true">Close</button>
    </div>
  </div>
  <?php } ?>
</script>
<?php <!-- A CSS3 Modal -->
if ( !empty($question['help']) ) { ?>
  <div id="<?php echo $q_id; ?>-InfoCSS3Modal" class="css3Modal">
  <a href="#close" class="css3ModalBg"> </a>
  <div>
    <div class="modal" tabindex="-1" role="dialog"
    aria-labelledby="<?php echo $q_id; ?>-info-css3-
    modal-label" aria-hidden="true">
      <div class="modal-header">
        <a href="#close" class="close" data-
        dismiss="modal" aria-
        hidden="true">×</a>
        <h3 id="<?php echo $q_id; ?>-info-css3-
        modal-label">Question <?php echo $i+1,
        '.', $j; ?> Explanation</h3>
      </div>
      <div class="modal-body">
        <?php echo $question['help']; ?>
      </div>
      <div class="modal-footer">
        <a href="#close" class="btn btn-primary
        btn-resize" data-dismiss="modal" aria-
        hidden="true">Close</a>
      </div>
    </div>
  </div>
  </div>
  </div>
  <?php } ?>

```

**Figure 26 – A code snippet from index.php showing the JQuery and CSS3 modals**

```

<script>
    function show_info(modal_id) {
        // Show modal for phones and tablets
        if ($(window).width() < 767) {
            $(modal_id).modal('show');
            $(".show-info").popover('disable');
        } else {
            $(".show-info").popover('enable');
        }
    }
</script>

<!-- An information button shown beside a question -->
<a href="#"<?php echo $q_id; ?>-InfoCSS3Modal" data-tag="#"<?php echo $q_id;
?>-info" onClick="show_info($(this).attr('data-tag'))" role="button"
class="btn btn-link show-info" title="Question <?php echo $i+1; ?>.<?php
echo $j; ?> Explanation" data-content="<?php echo $question['help']; ?>"
data-html="true"><i class="icon-info-sign"></i></a>

```

**Figure 27 – A code snippet from index.php showing how to switch between modals, popovers, and CSS3 modals**

## MeekroDB

Even though PHP has native functions and packages for interfacing with MySQL databases, the MeekroDB library (<http://www.meekro.com/>) was used to ensure the web app would be secured and not subjected to SQL injections. This also abstracts the database connections into a separate object making it easy to quickly query the database with the minimum lines of code.

## Use of APIs

Mandrill (<http://mandrillapp.com>) is an email infrastructure service created by MailChimp for applications and websites to send and track transactional (non-bulk) emails such as email address confirmations, password resets, and purchase receipts.

The Mandrill API (Application Programming Interface) is used to simplify the creation of styled emails with the use of templates and customize them with dynamic variables (merge tags). The Mandrill API also simplifies things by handling all the validation. Mandrill has been whitelisted by common email servers, which means the formatted emails will always arrive in the user's inbox and not be marked as spam.

Mandrill also keeps of a record of when the email was read and the number of times it has been read. This will be useful for the client if they need to analyze the effectiveness of emailed results.

## Questionnaire Revision Control

The custom questionnaire revision control system allows for administrators to make and publish changes to the questionnaire while still allowing users to view, print, and modify the responses to previous questionnaire versions if they supply a valid saved questionnaire ID. Users who do

not supply a valid saved questionnaire ID will be shown the latest published version of the questionnaire.

## AJAX Form Submission

AJAX (Asynchronous JavaScript and XML) is implemented for all JavaScript-enabled forms to allow for form submissions without a page refresh. AJAX is currently used to write questionnaire responses to the database and redirect the user to the questionnaire results page when the “Next Section” or “Finish” buttons are clicked, email questionnaire results, and save user feedback.

The success message returned by the PHP file determines the next action such as whether to unhide the next section of the questionnaire or to take the user to the questionnaire results page. As shown in the PHP code snippet below, if the submitted form contains the key value pair finish-btn and finish, the questionnaire\_id session variable will be printed. This printed value will be read by the questionnaire page and the user will be redirected to the questionnaire results page.

```
if(isset($_POST['finish-btn']) && $_POST['finish-btn']=='finish'){  
    echo $_SESSION['questionnaire_id'];  
}
```

**Figure 28** – A code snippet from questionnaire-submit.php showing how to check if the finished button was clicked

## Future Enhancements

Due to the project’s time constraints, some features were not implemented to decrease the project’s complexity.

### Conditional Branching of Sections

Currently, the questionnaire’s sections are displayed sequentially and all questions must be answered to reach the questionnaire results page. This is unlike the ARECCI Ethics Screening Tool where the section subsequently shown on completion of a section depends on the responses in the previous sections.

### Support for Browsers with JavaScript Disabled

The questionnaire and questionnaire results page both have special styling for when JavaScript is disabled. However, there is currently no logic to handle questionnaire submission, emailing questionnaire results, and submitting feedback when JavaScript is disabled.

### AJAX Form Submission per Question

The questionnaire currently only sends an AJAX form submission on completion of a section of the questionnaire. This can be improved by doing an AJAX form submission whenever an input box, text area, or radio button has been changed so users can stop filling out the questionnaire and bookmark the page to return later at any time without losing any changes.

## **Colour Coded Categories**

Colours are currently assigned to each category to help users understand how the resulting questionnaire category scores are calculated, which can then lead them to understand why the evaluation approaches are recommended on the questionnaire results page. However, the category colours are not currently shown on the questionnaire and questionnaire results pages.

## **Improved Authentication System**

The authentication system can also be improved if the password hashes are stored in the database rather than having the passwords stored in plain text in the authentication script, if privileges are limited to certain accounts such as having a read only account, and if there was an actual login page rather than just sending a WWW-Authenticate header.

## **Remove Questionnaire Categories**

To keep things simple, the function to remove questionnaire categories has not been implemented since it would require scanning each section's questions to unassign it from a category that is to be deleted and then notifying the user with a list of questions with no category assigned. Allowing categories to be removed would be beneficial in the long run if lots of categories are to be created.

## **Purge Old Answer Sets and Questionnaire Revisions**

Allowing an administrator to purge old answer sets and then purge questionnaire revisions that have no answer sets associated with them can potentially save disk space in the long run if the web app has heavy usage. The last accessed column in the answers table can be used to filter old answer sets to determine which should be deleted.

## **View Old Questionnaire Revisions**

Currently, only the latest version of the questionnaire can be viewed from the admin page. This can be improved by creating a questionnaire history page where the administrator can browse through older revisions and “revert” to an older revision if necessary. Reverting in this case means duplicating the questionnaire revision to allow for editing as published revisions cannot be edited.

## **Preview Questionnaire**

Currently, the questionnaire cannot be tested until it is published. It would be helpful to administrators to be able to test if the questionnaire “flows” correctly between sections before it is shown to the public. A “is\_preview” flag can be added to the answers table and any answer sets that that flag set can be deleted when visiting any of the backend pages.

## **Support for Multiple Questionnaires**

Currently, the system only allows for one active questionnaire. This can be improved by allowing the administrators to create multiple questionnaires and allow the user to select which questionnaire they want to complete.

## Development Environment Network Topology

The development network as shown in Figure 29 consists of four machines, three of which are virtual machines in bridged networking mode. The host machine runs MAMP (web server) and VMware and the web application is tested from the web browsers running on each of the virtual machines.

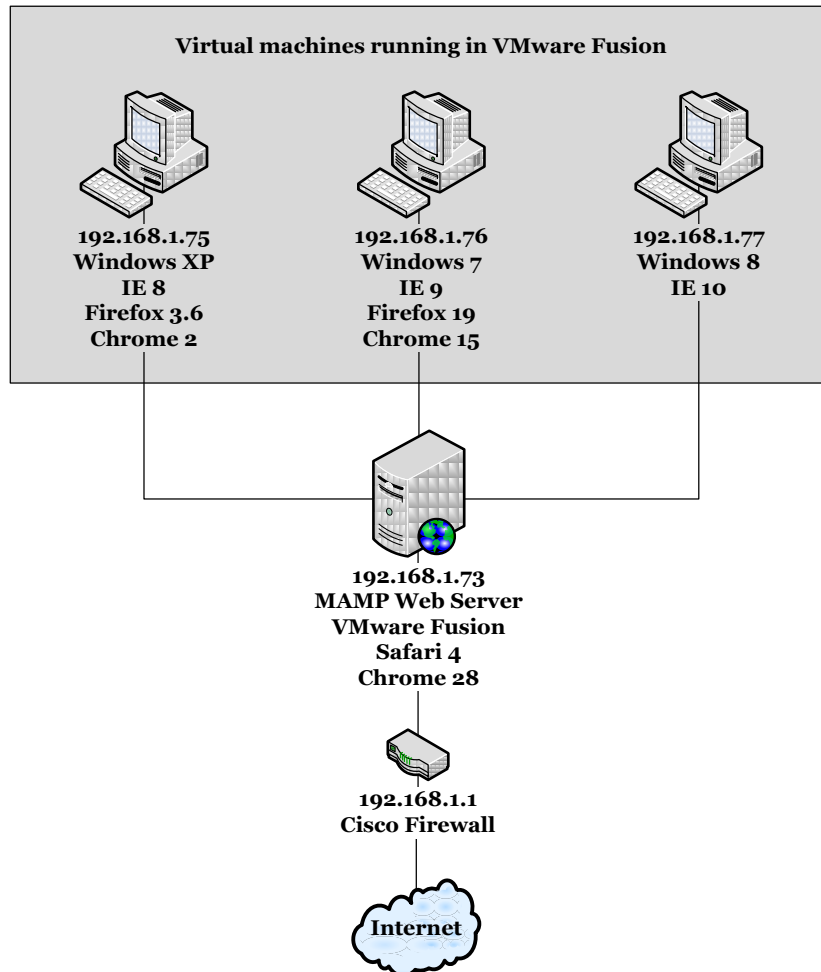


Figure 29 – Development environment network topology

## Technologies

The questionnaire and admin site is coded in HTML and CSS. JavaScript (jQuery) is used for form validation, dynamically showing each section of the questionnaire, and reordering/deleting questionnaire sections and questions. FPDF is used to generate a PDF version of the questionnaire. Email styling is handled using HTML and CSS and sent using the Mandrill API. The questionnaire data is stored in a MySQL database and the data is written to the database and retrieved using PHP. Everything is hosted on an Apache web server.

Usability studies are conducted using Usabilla.

## Innovation

In this project, I learned how to perform usability studies, do revision control, and make API calls.

## Usability Testing

In the traditional waterfall development approach, users do not have a chance of give feedback to the developers about a product until the very end when it is difficult and costly to make changes. This results in a product that does not satisfy a user's needs and will likely not be used hence being a waste of time and resources.

Instead of taking the waterfall approach, we should use the agile methodology, where developers can receive feedback from users early and make changes to the software iteratively. It's been found that instead of showing a fully built product to users, conceptual design mock-ups are often sufficient for the users to evaluate whether the product will meet their needs. A mock-up is a rough sketch of how a user interface will look and outlines the workflow. Changes can then be quickly made to the mock-ups according to user feedback and then be tested again. This iterative process results in a product that users will be more satisfied with.

To test the mock-ups, we used Usabilla. In Usabilla, we uploaded screenshots of the mock-ups and paired each screenshot with a series of questions. Users are then asked to click on the part of the mock-up that they think satisfies the question. For example, we asked ten users where they will click to get more information regarding a question. Users can also provide comments on individual screens. After multiple users have completed the mock-up testing, a heat map can be generated to show where users clicked to answer the questions to discover problematic areas within the tool. Usabilla also measures the time it took users to answer each question, and the total amount of time to complete all the questions. We then make changes in the implemented design based on the feedback. The results of the usability studies can be found in Appendix D – Usability Testing.

One of the problems that are noted in the usability study is that 60% of the mobile phone mock-up testers did not know where to tap to save the questionnaire to finish later. The implemented version of the website incorporated this feedback by replacing the bookmark page button with a print to PDF button as the bookmark function does not work in some browsers and the button is redundant and adding a message when the Next Section button is clicked to inform the user the questionnaire responses have been saved.



**Figure 30 – Heat map showing where mobile phone mock-up testers clicked**

Another problem noted that is noted in the usability study is that 20% of desktop mock-up testers were unsure what some of the items in the actions sidebar means. The implemented version of the website incorporated this feedback by making the labels beside each icon more

descriptive and by rewording the labels to begin with a verb. For example, “Permalink” has been renamed to “Save Results Link”, “Email” has been renamed to “Email Results”, and “Feedback” has been renamed to “Give Feedback”. Other suggestions made by testers include removing the heading inside each accordion section as it is redundant and making the “Save to PDF” section as a button rather than an expandable section. The feedback made by the testers can be found in Appendix D – Usability Testing.

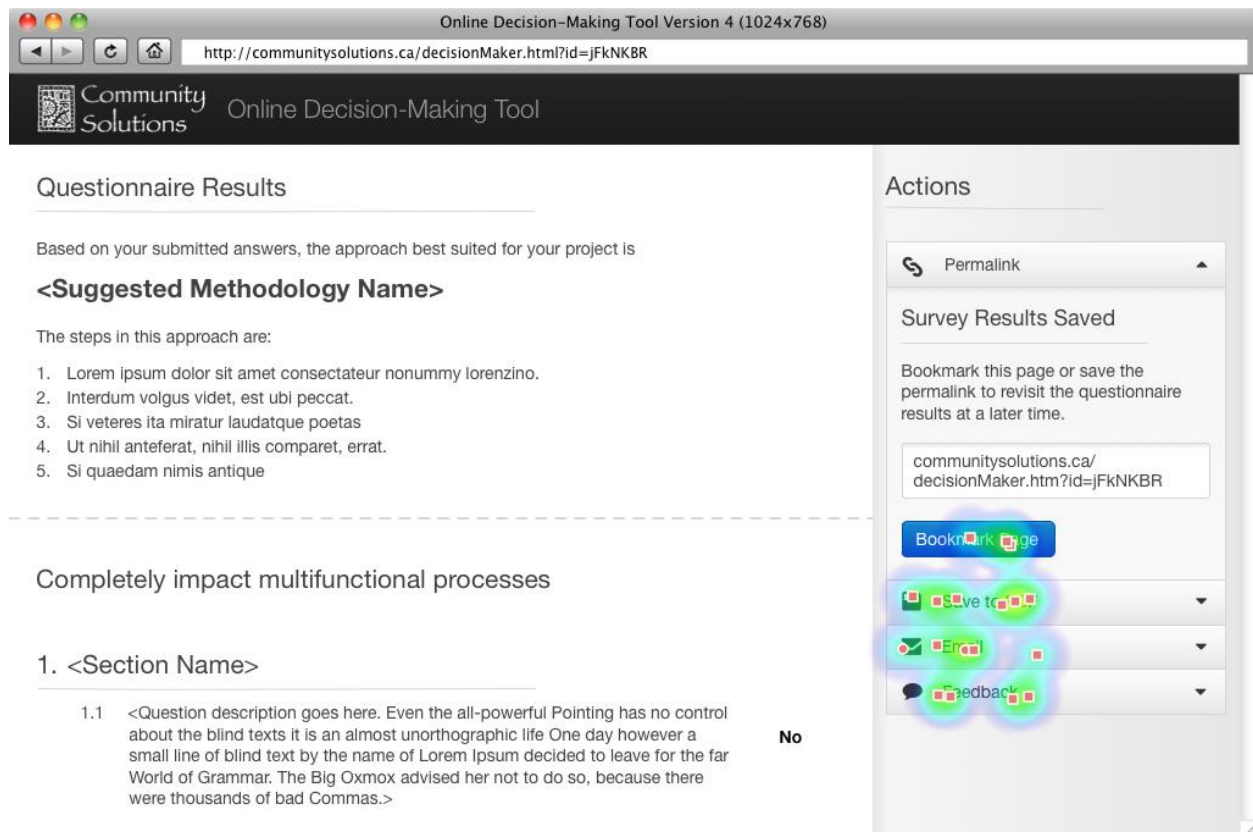


Figure 31 – Heat map showing where desktop mock-up testers clicked

## Questionnaire Revision Control

Since the survey tool will allow the admin to freely modify the questionnaire questions, sections, categories, and results weighting scheme, questionnaire revision control must be implemented so that when a questionnaire is revised, the previously completed questionnaires will remain correctly stored in the database and always be available for viewing.

## Custom Formatted PDF Creation

After completing the questionnaire, users can request a PDF version of the completed questionnaire along with the recommendations. To make things look tidy, the PDF of the questionnaire will not look like a print out from a web page. Instead, these PDF documents are generated by a PHP script and the layout can be data can be custom formatted for printing. This is achieved using the FPDF PHP library. A PDF preview of some sample questionnaire results is shown in Figure 32.

Some Project Questionnaire Results

Tuesday, September 3, 2013



## Evaluation Approach Decision Tool

### Questionnaire Results

---

Based on your submitted answers, the approaches best suited for your project are:

#### Category 1

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.

#### Category 2

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.

---

### Some Project

#### 1. Section 1

---

1.1	question 1.1 goes here	No
1.2	question 1.2 goes here	No
1.3	question 1.3 goes here	Yes
Notes	Some notes.	

#### 2. Section 2

---

2.1	question 2.1 goes here	Yes
-----	------------------------	-----

<http://192.168.1.65/decision-tool/results.php?id=51d50975d5b3e>

Page 1/2

**Figure 32 – Custom formatted PDF of the questionnaire results**

## **HTML and CSS Styled Emails**

Emails requested by the user at completion of the questionnaire containing the questionnaire recommendations and questionnaire permalink will be styled with HTML and CSS to make the email organized and visually appealing. Emails are sent using the Mandrill API. Mandrill allows us to track the emails that have been sent and allow us to monitor bounce rates and the number of times an email has been read. An email screenshot of some sample questionnaire results is shown in Figure 33.

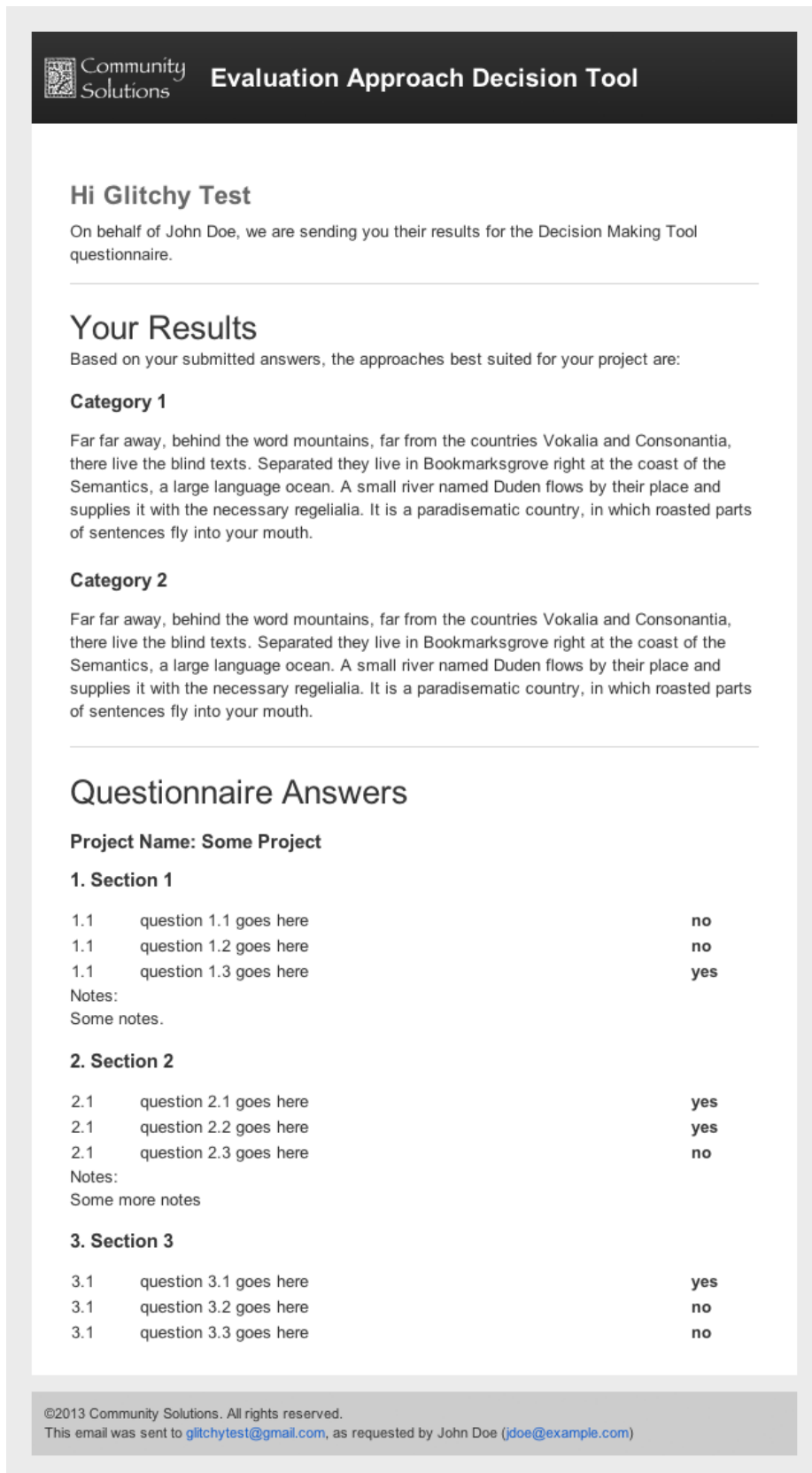


Figure 33 – An email sent by Mandrill containing the questionnaire results

## Conclusion

From the completion of this project, I created a decision-making tool similar to the ARECCI Ethics Screening Tool that allows users to receive project advice upon completion of the questionnaire.

While creating this web application, I learned many new skills that can be applied for future web development projects. Such skills include doing visual design, performing click-through usability testing, creating responsive website layouts, making API calls, handling JSON data, submitting forms via AKAX, adding CSS3 transitions, and authenticating users over HTTP with PHP. I have also improved my skills in designing database schemas, writing SQL queries, generating PDF documents with FPDF, writing JavaScript logic, creating CSS rules, and documenting functional requirements.

In the visual design phase, I experimented with using Omnigraffle to create mock-ups of the various screen sizes. I discovered there are Omnigraffle stencils available for Twitter Bootstrap that can be imported and used. Use of these stencils greatly speed up the process as opposed to designing everything from scratch in a more traditional software tool like Adobe Photoshop.

This was also my first time using the Twitter Bootstrap framework for front-end web development and dealing with responsive web design. Version 2.3 of Twitter Bootstrap received a lot of positive feedback from various online communities. So I decided to test it out for this project. I was pleased that this framework met my expectations. The Bootstrap grid system has been worked out from design (with Omnigraffle stencils) to development side. The functionality to switch between different layouts based on the browser width was built-in. This made it very easy for me to decide what the layout should be like and how I should implement it using the framework. Numerous tutorials online also helped when it comes to troubleshooting issues and for me to learn more about responsive design.

Another new skill I learnt is how to perform click-through usability testing using mock-ups. The Usabilla testing service has guides on how to create a usability test. I followed these to create my test questions. Then I invited users online to give feedback on my designs. Because the front-end of the web application is a simple online survey, the design was very simple. There was not a lot to be tested and the users for the most part had no difficulties using the tool. So there was not much to analyze and change. However, this was still a good learning experience for me in designing and performing the usability tests.

I spent a lot of time implementing the feature for emailing results to users. Mandrill was my first time using a public API. Like everything else, there were lots of documentation and tutorials available on the use of this email API. I learnt how to pass specific parameters, such as email address, names, report details and email template, into a properly formatted JSON API request. I also look at how to handle responses sent back by the Mandrill API.

Another component to the email results feature is AJAX validation and submission for web form. I discovered jQuery, the JavaScript library I was using, includes AJAX calls. I was able to hook up a form validation plug-in to jQuery to make it work.

The end result is a simple web application that has a questionnaire on the front-end that is user-friendly across desktop, tablet and mobile usage that allows them to take the questionnaire, get results and the ability to save the results and review it at a later time. There is also an admin back-end that allows the client to customize the questionnaire questions and how results are calculated.

The current application is still very simplistic. In the future, it can be improved by adding the following features: supporting section branching logic, allowing for multiple active questionnaires, improving the backend authentication system, permitting old answer sets and questionnaire revisions to be purged, and enabling older questionnaires to be viewed.

## Appendix A– Design Process & Mock-ups

Several high-fidelity wireframes revisions of various screen resolutions were created for both the questionnaire and questionnaire results pages and submitted to the client for feedback.

### Questionnaire Mock-up Version 1

This mock-up version shows the ribbon bar (the grey bar underneath the header) that is implemented in the tablet version of the web app. The bookmark button in the ribbon bar got changed to a print to PDF button in the implemented version as some web browsers such as Chrome and Safari forbid JavaScript from activating the bookmark page feature due to security concerns. The indentation applied to each section's questions got removed to save space. The radio button layout is used in the no-JavaScript web app styling as the Twitter Bootstrap "radio buttons" do not function without JavaScript.

Online Decision-Making Tool Version 1 (800x600)

http://communitysolutions.ca/decisionMaker.html?id=jFkNKBR

Community Solutions Evaluation Approach Decision Tool

Progress 0% Complete Results Link communitysolutions.ca/decisionMaker.htm?id=jFkN

### Tool Overview

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary r egelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.

---

Project Name

### 1. <Section Name>

	Yes	No
1.1  <Question description goes here. Even the all-powerful Pointing has no contr ol about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were	<input type="radio"/>	<input type="radio"/>

Figure 34 – Mock-up version 1 of the questionnaire page

The error highlighting got toned down in the web app as the red was too bright. The radio buttons in the no-JavaScript version of the web app has a red box around them for more emphasis, but not too much. The “Please select an option” error message got moved below the radio buttons to save space and is no longer a tooltip as tooltips require JavaScript.

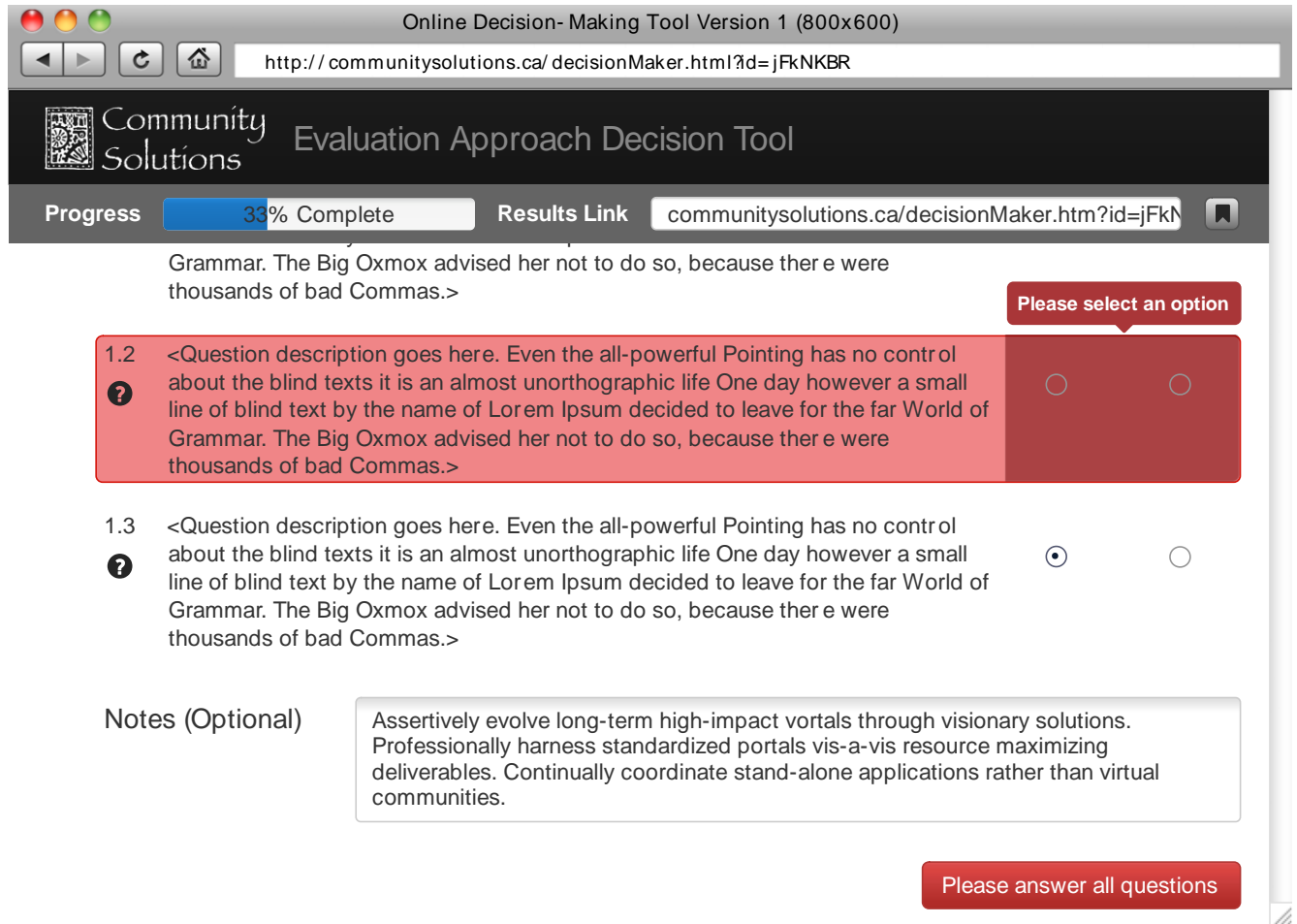
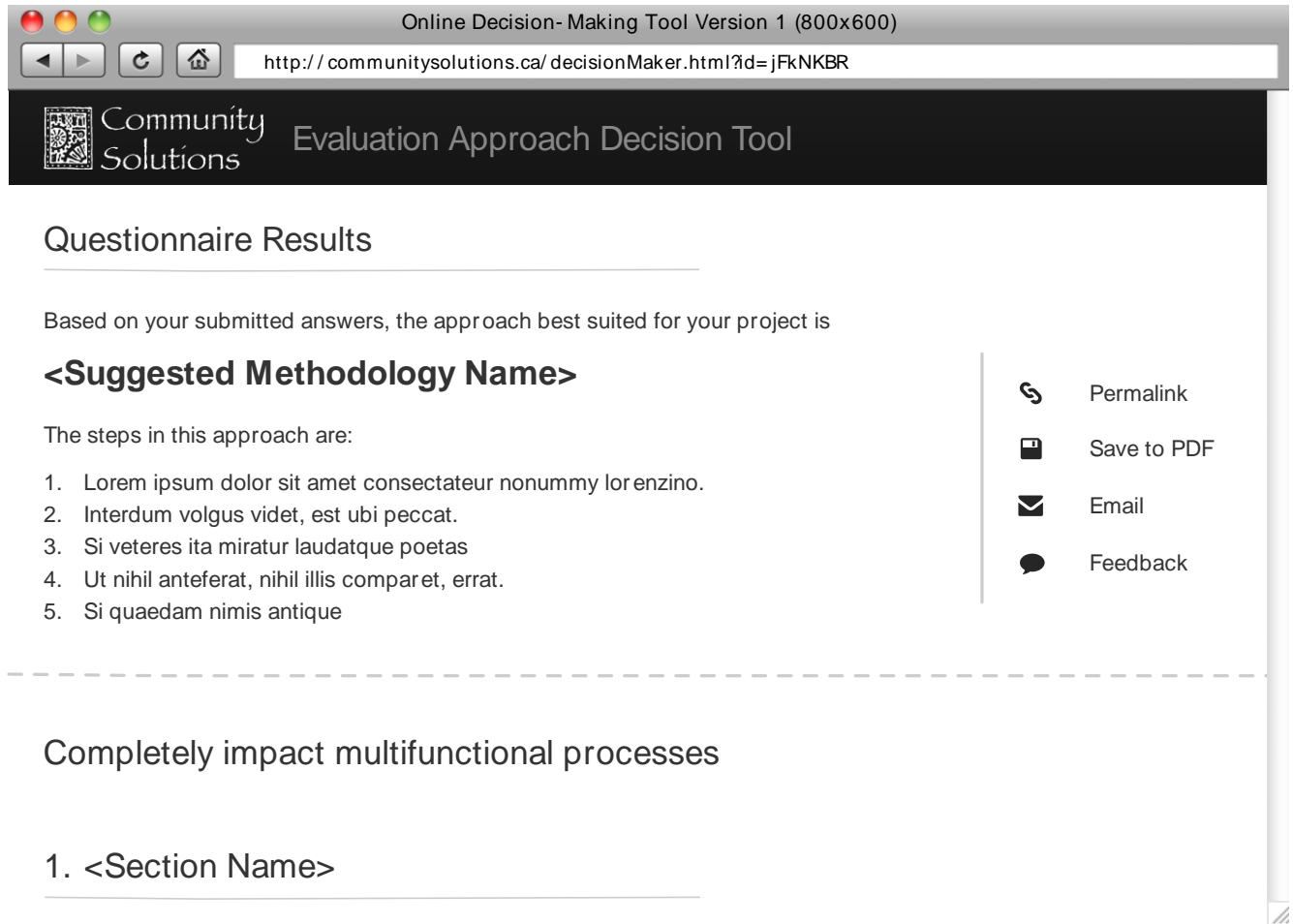


Figure 35 – Question error highlighting on the questionnaire page

## Questionnaire Results Mock-up Version 1

The actions box (the list of items to the far right of the questionnaire results heading) is implemented in the tablet version of the web app. The label by each icon has been reworded to begin with a verb and to be more descriptive. For example, “Permalink” has been renamed to “Save Results Link”, “Email” has been renamed to “Email Results”, and “Feedback” has been renamed to “Give Feedback”.



**Figure 36 – Mock-up version 1 of the questionnaire results page**

The indentation applied to each section’s questions got removed to save space. Also, the alignment of the notes heading is changed to align with the question numbering and the alignment of the notes text is changed to align with the question description text.

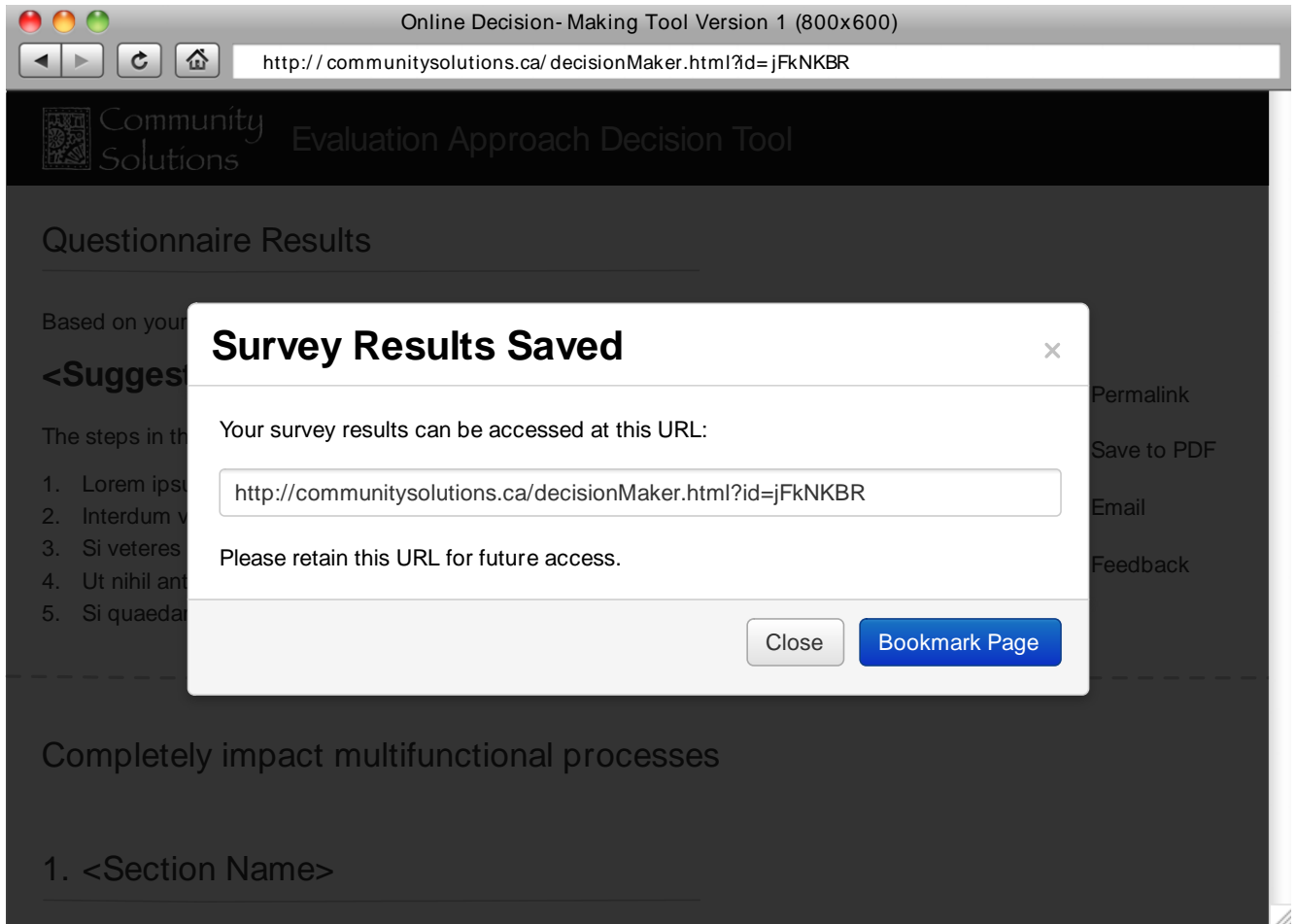


Figure 37 – Results link modal on the questionnaire results page

The required fields in the modal form are marked with an asterisk so there is no confusion as to which fields are required and which are not. The error message is moved below the text field so as to not overlap with the text field text. The “Sender Email” label has been renamed to “Your Email” for consistency.

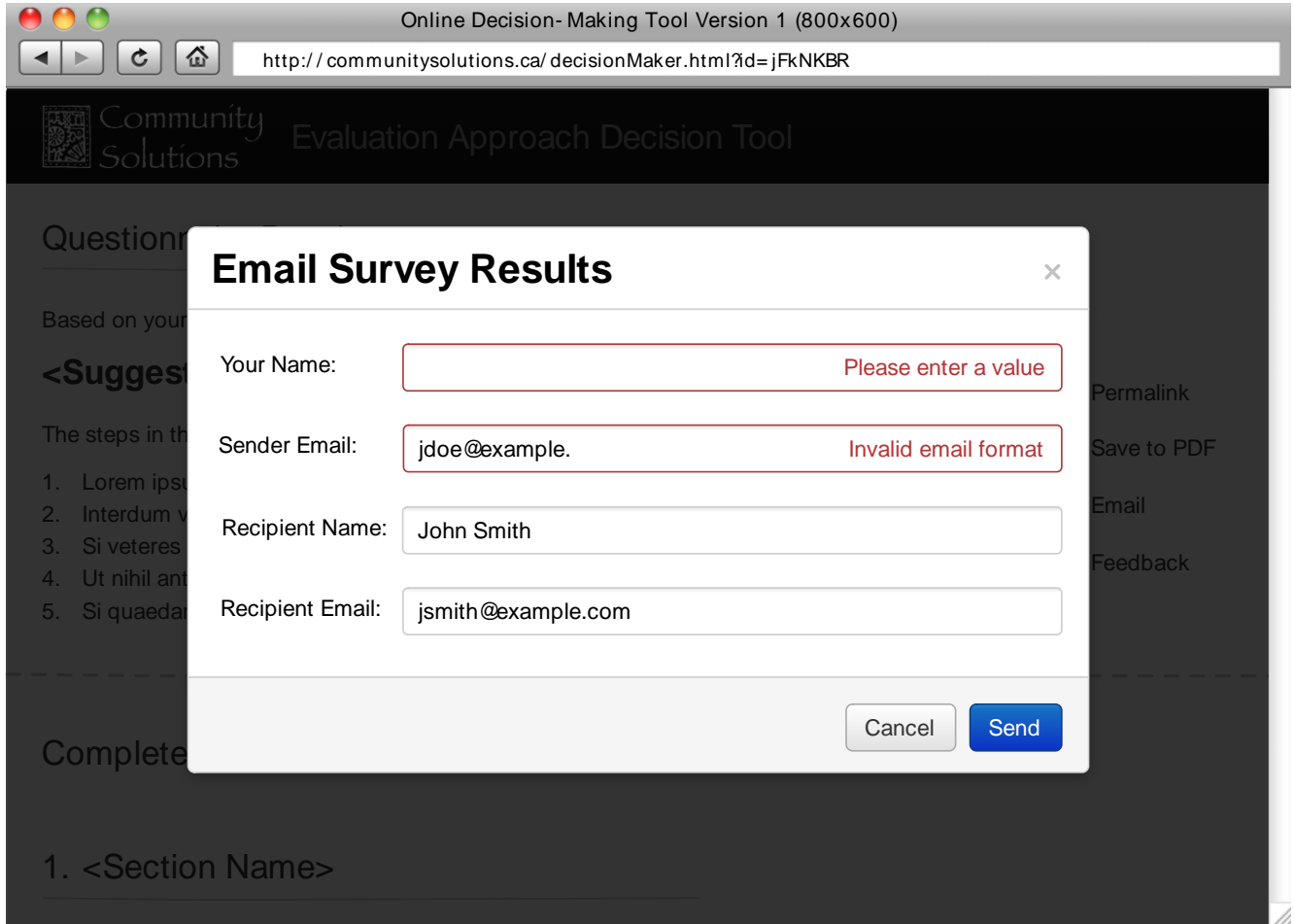


Figure 38 – Email results modal on the questionnaire results page

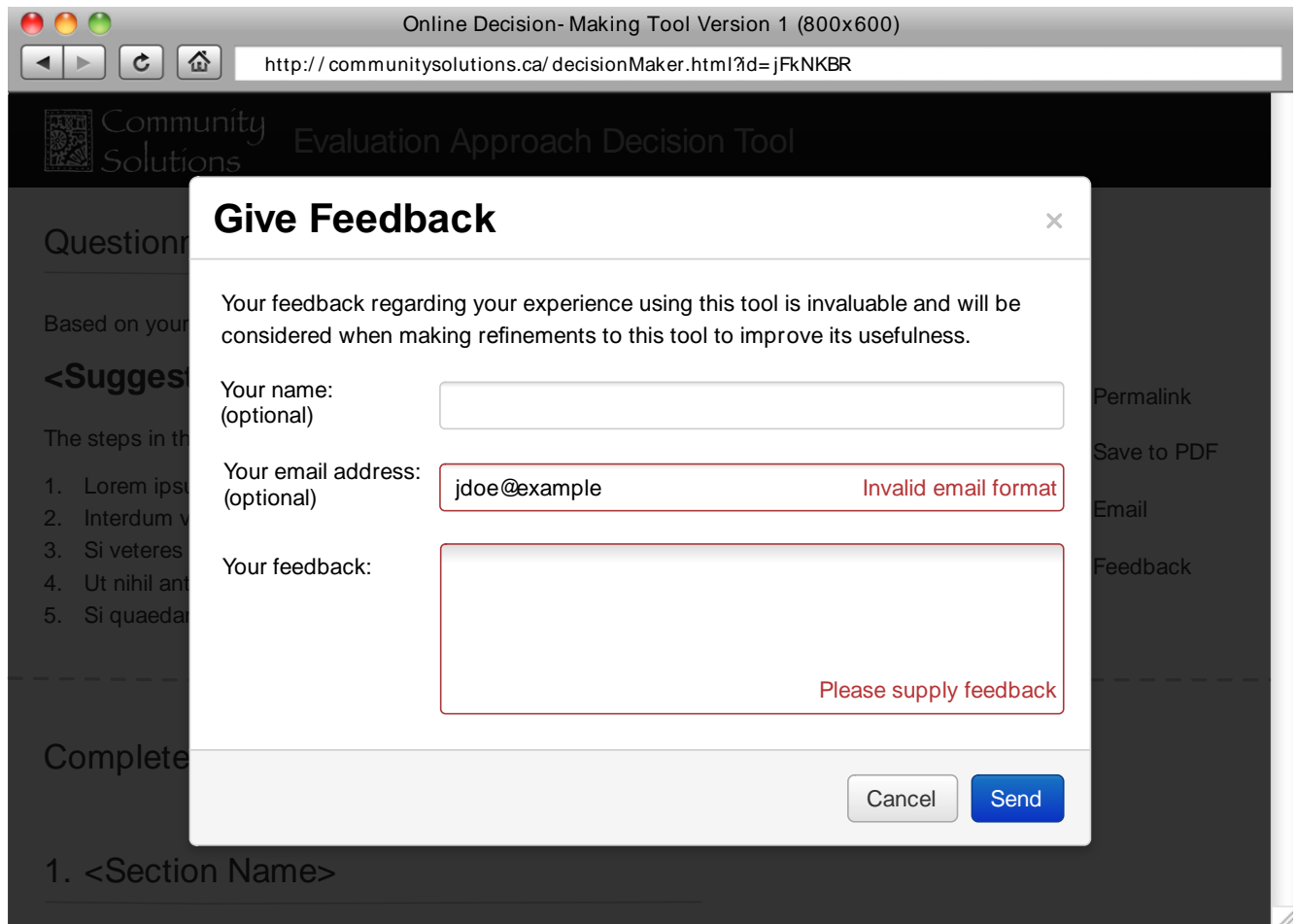
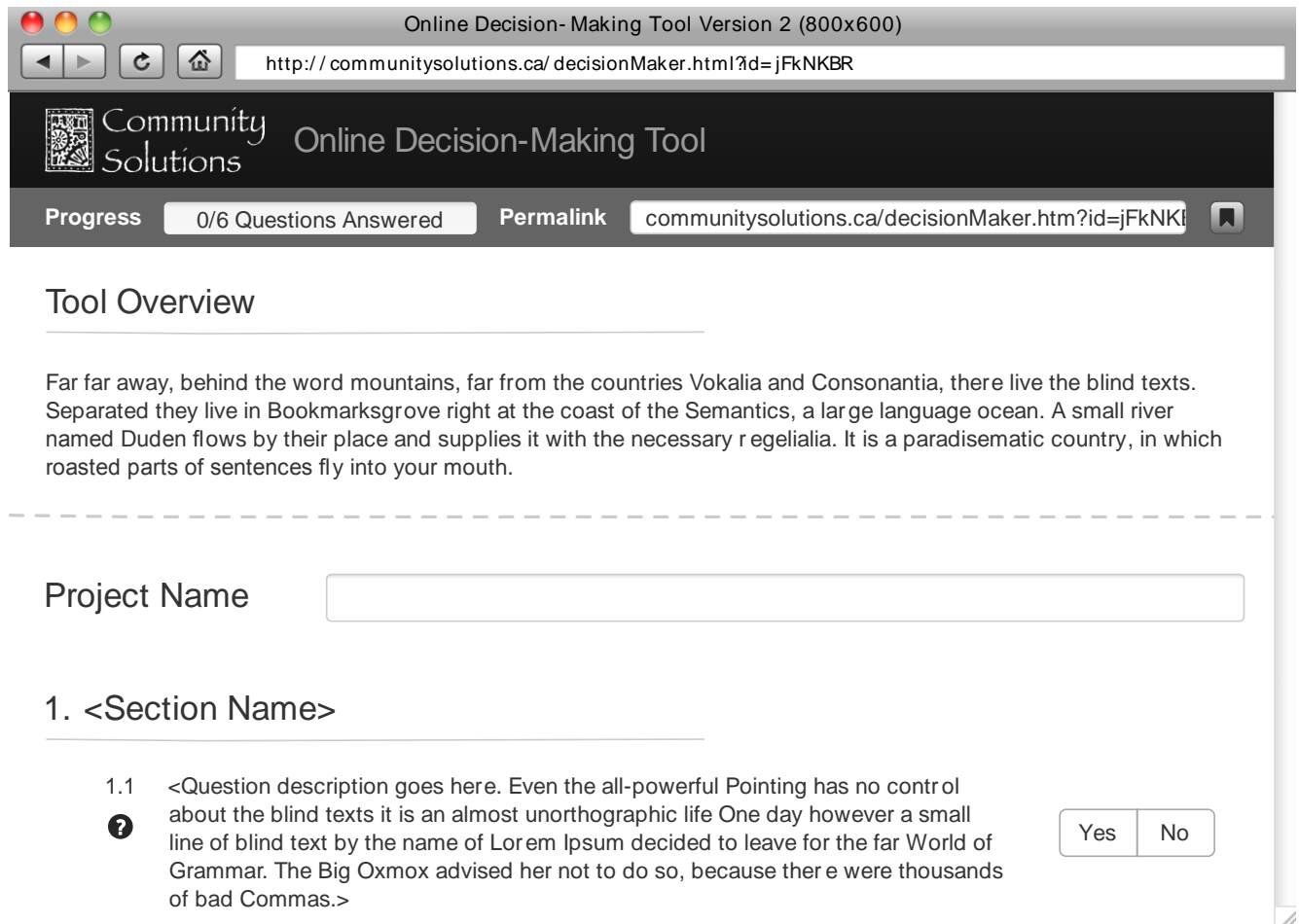


Figure 39 – Give feedback modal on the questionnaire results page

## Questionnaire Mock-up Version 2

This mock-up version shows the Twitter Bootstrap “radio buttons” in place of the standard radio buttons.



**Figure 40 – Mock-up version 2 of the questionnaire page**

This mock-up shows what it would look like if the information text for a question is shown right below the question when the information icon below the question number is clicked just like in the ARECCI Ethics Screening Tool, which this web app is based off of. This design was scrapped in favour of modals and popovers as having too much question information text shown will cause too unnecessary screen clutter and require the user to scroll too much on mobile devices. This design would have been used in the no-JavaScript version of the web app, but CSS3 modals proved to be a better, more innovative choice.

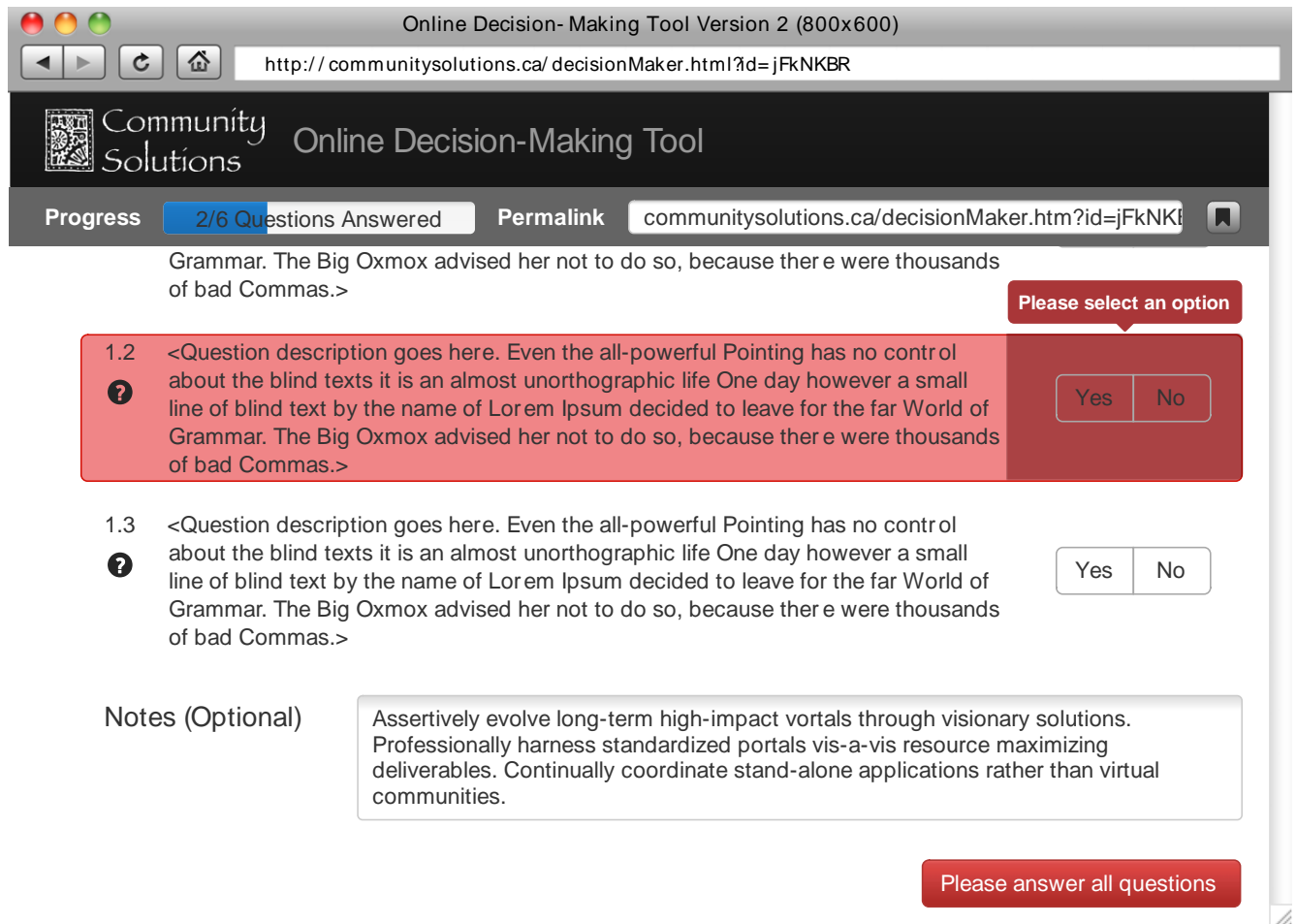


Figure 41 – Question error highlighting on the questionnaire page

### Questionnaire Mock-up Version 3

This mock-up shows what it would look like if the information text for a question is shown as a popover when the information icon below the question number is clicked. Popovers are used in the tablet and desktop wide screen versions of the web app. Also shown in this mock-up is a sidebar. The ribbonbar will turn into a sidebar when switching between the tablet and desktop wide screen versions of the web app.

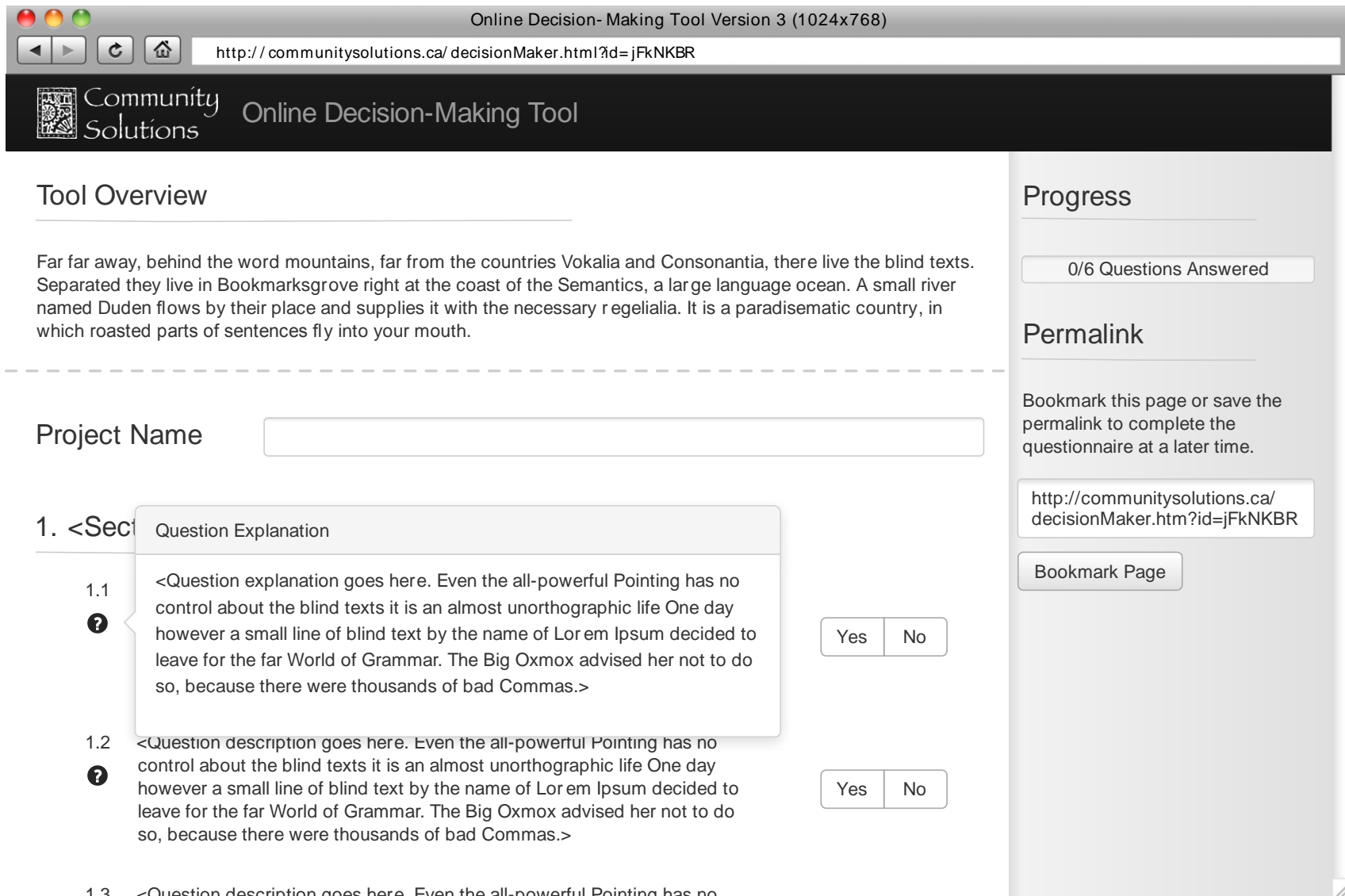


Figure 42 – Information popover on the questionnaire page

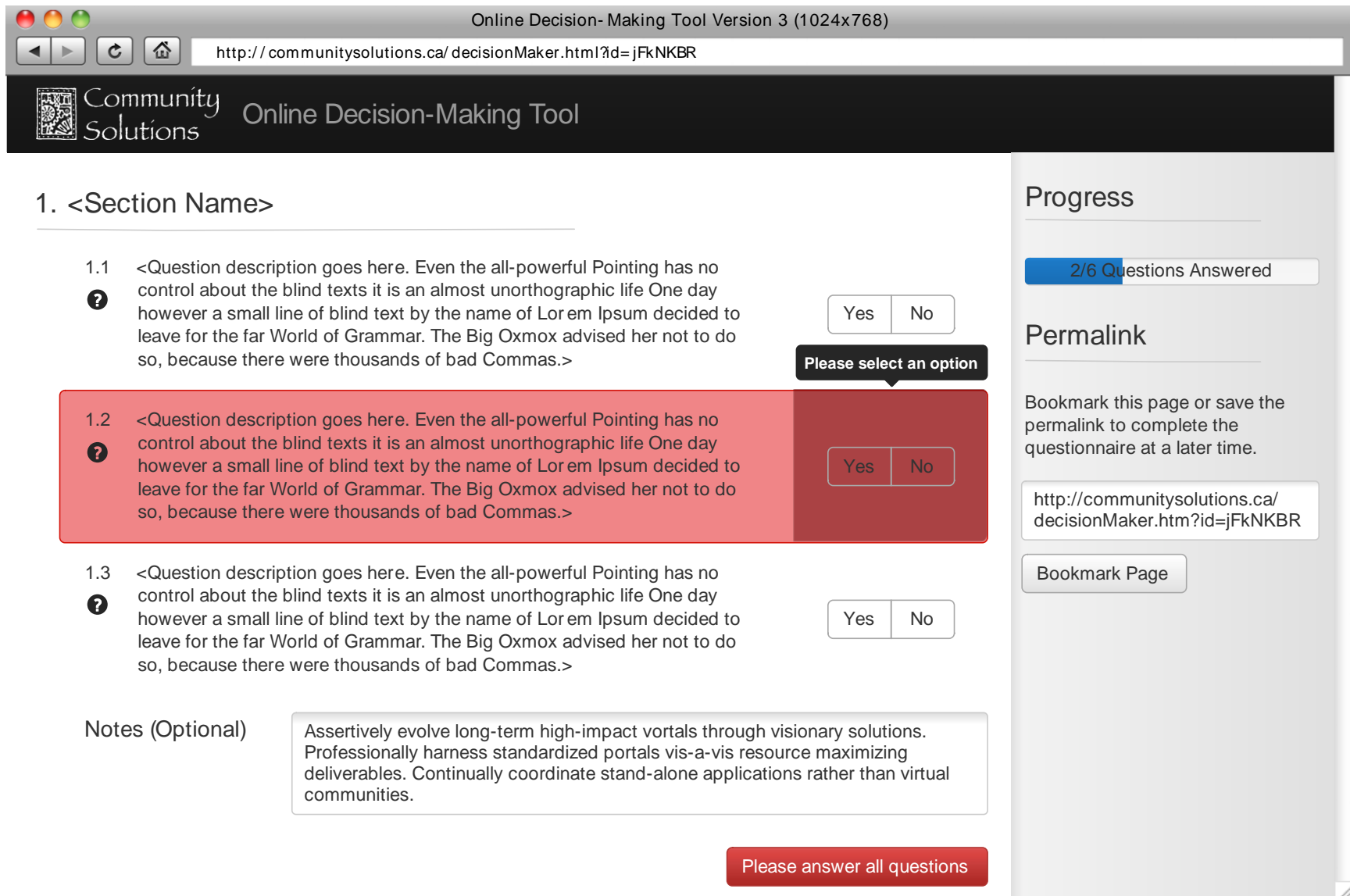


Figure 43 – Question error highlighting on the questionnaire page

## Questionnaire Results Mock-up Version 3

This mock-up version shows the sidebar in the questionnaire results page. This version is scrapped because having both the actions box and the sidebar is a poor use of screen estate.

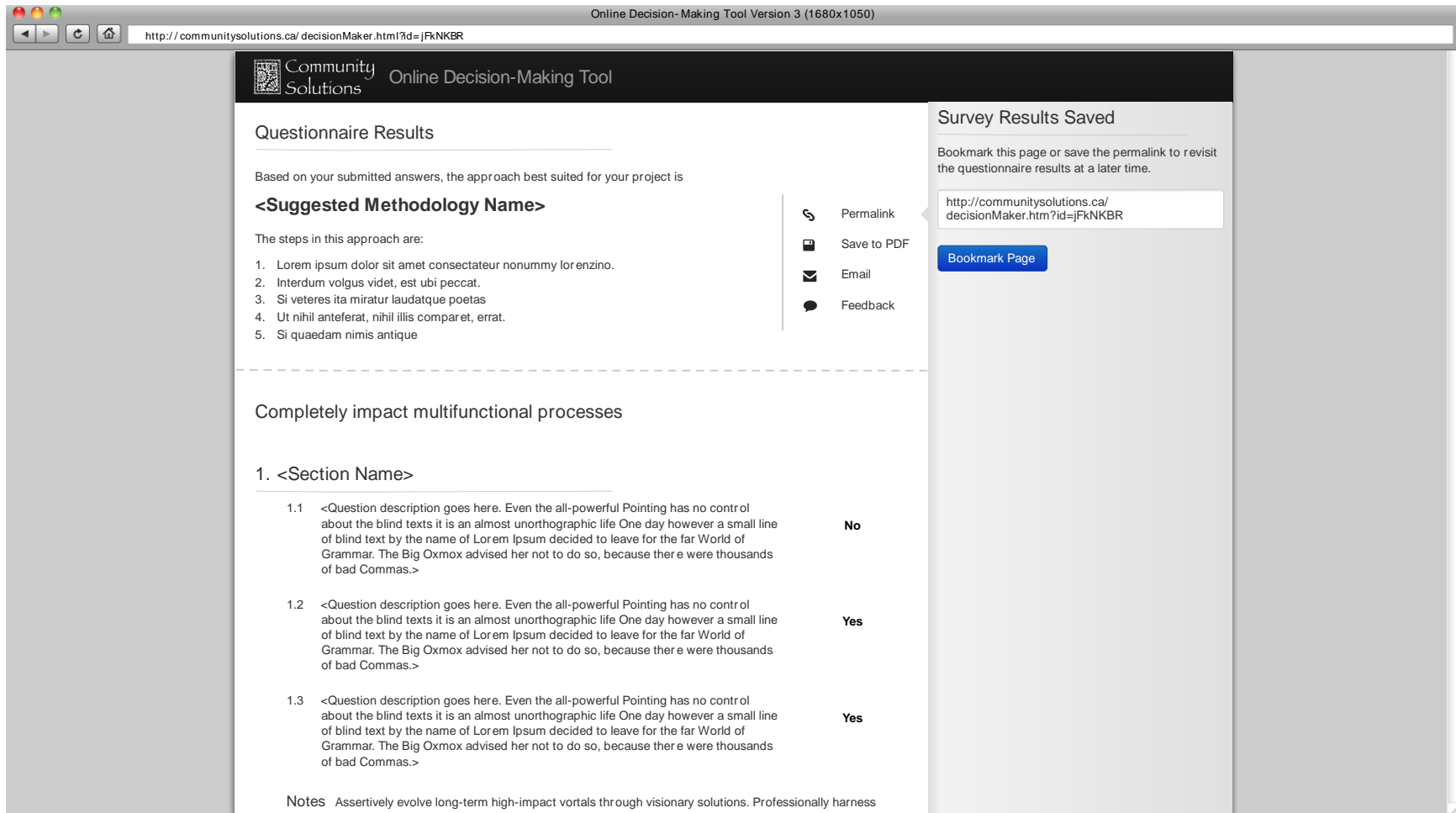


Figure 44 – Results link form on the questionnaire results page

Shown in this mock-up is the email survey results form. In the implemented version, the text field labels are positioned above the text field to save space. Required fields are also marked with the word required to prevent confusion as to which fields are required and which are optional.

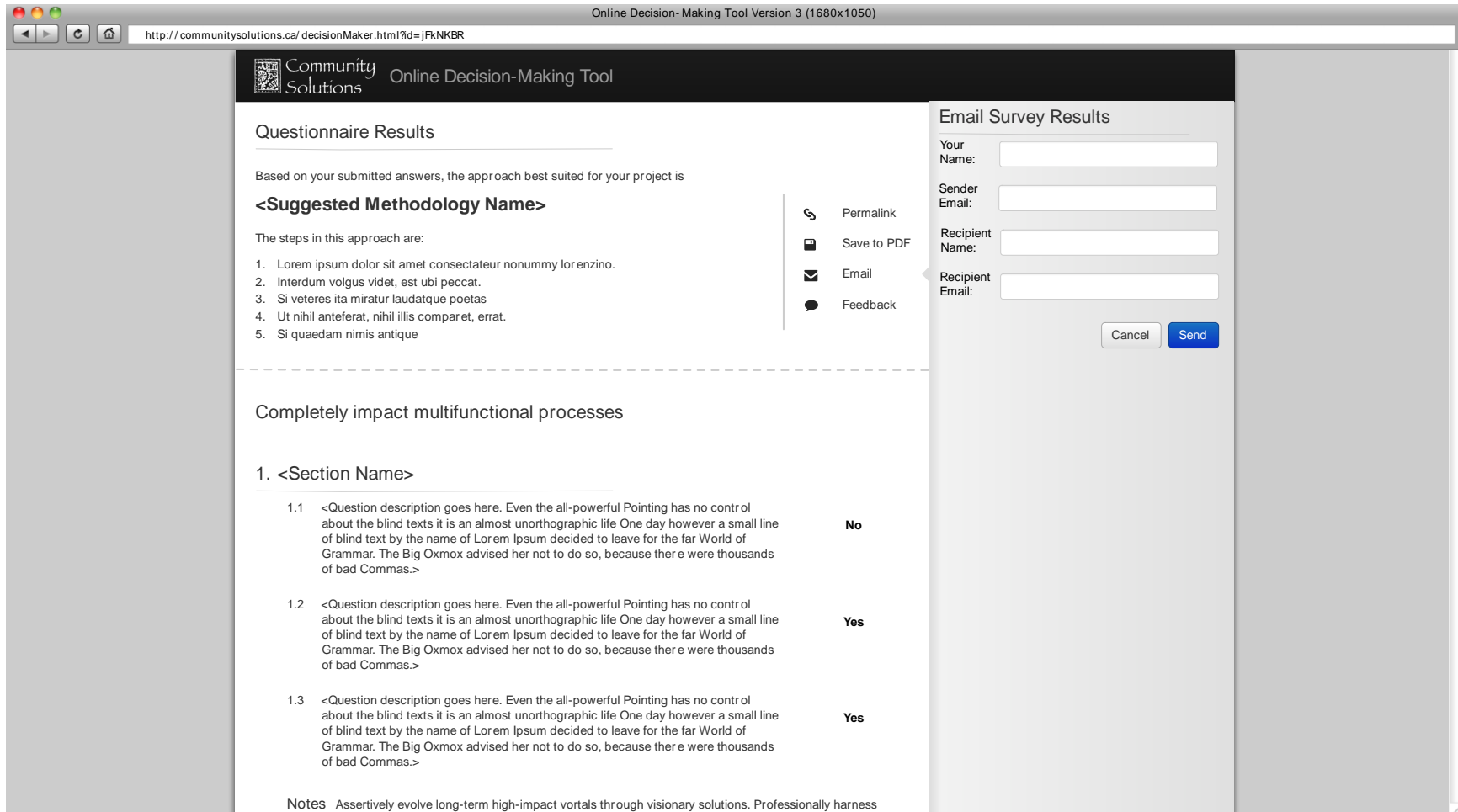


Figure 45 – Email results form on the questionnaire results page

Shown in this mock-up is the form error highlighting style that is used for all forms in the implemented web app.

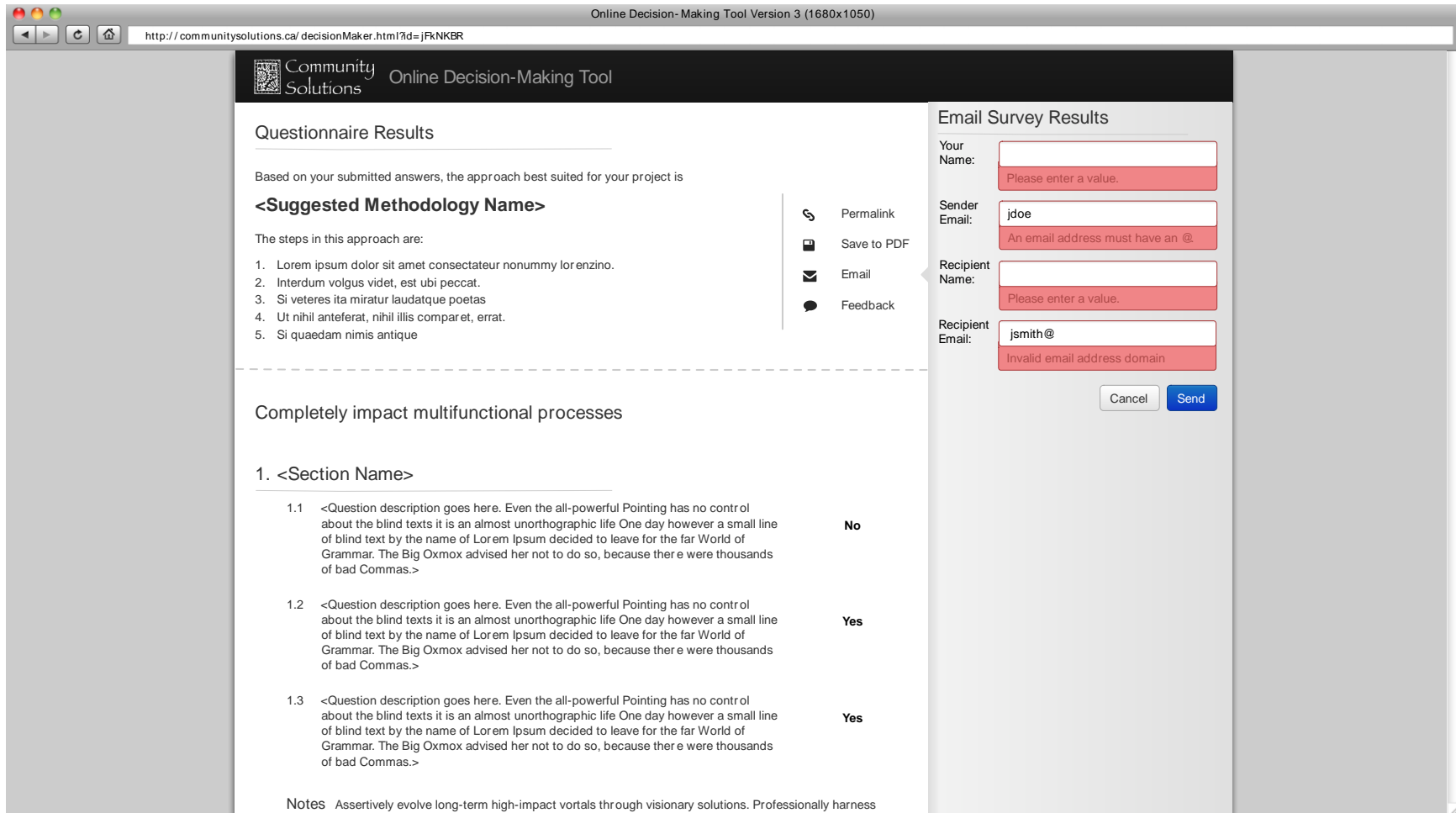


Figure 46 – Email results form validation on the questionnaire results page

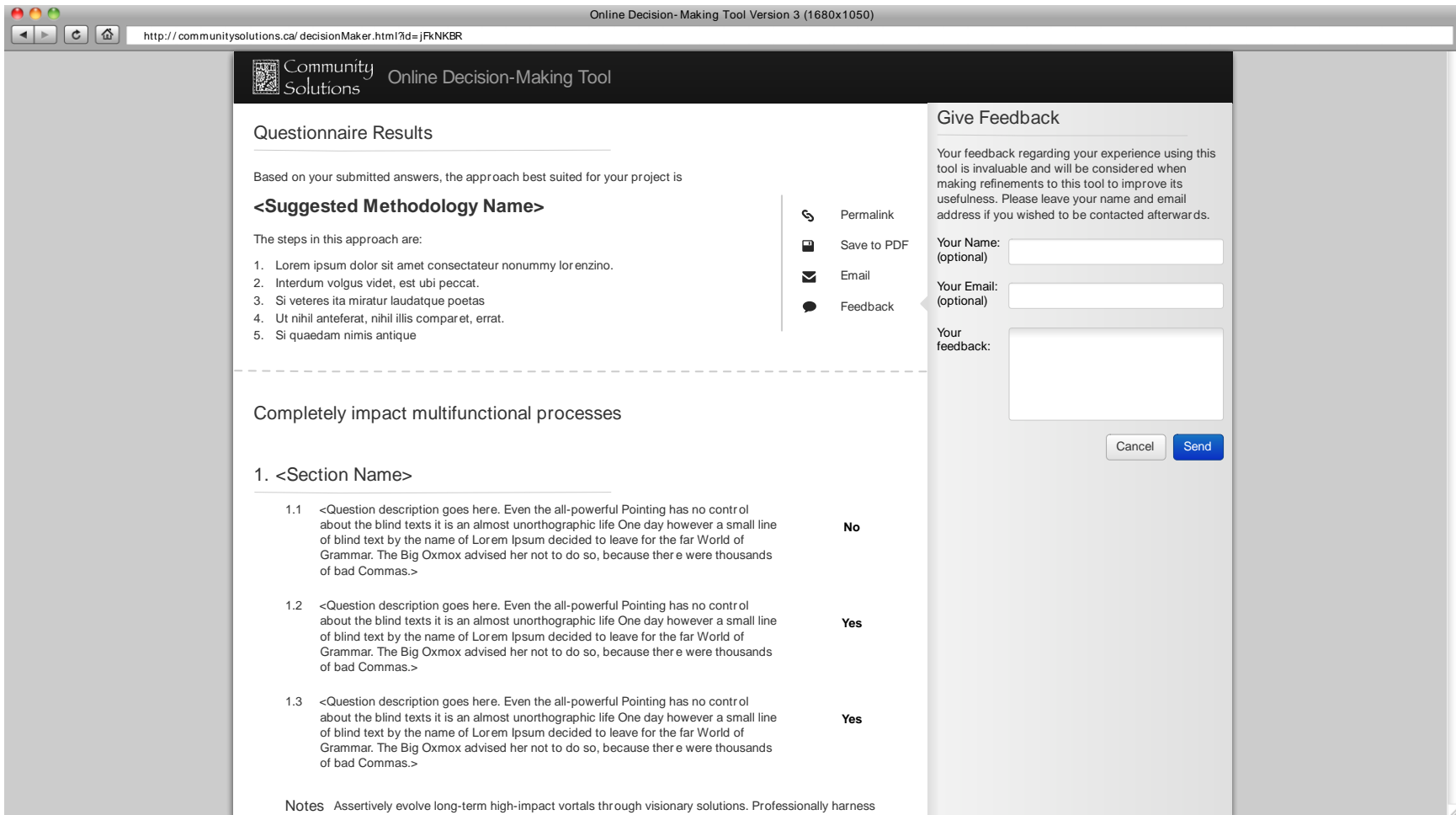


Figure 47 – Give feedback form on the questionnaire results page

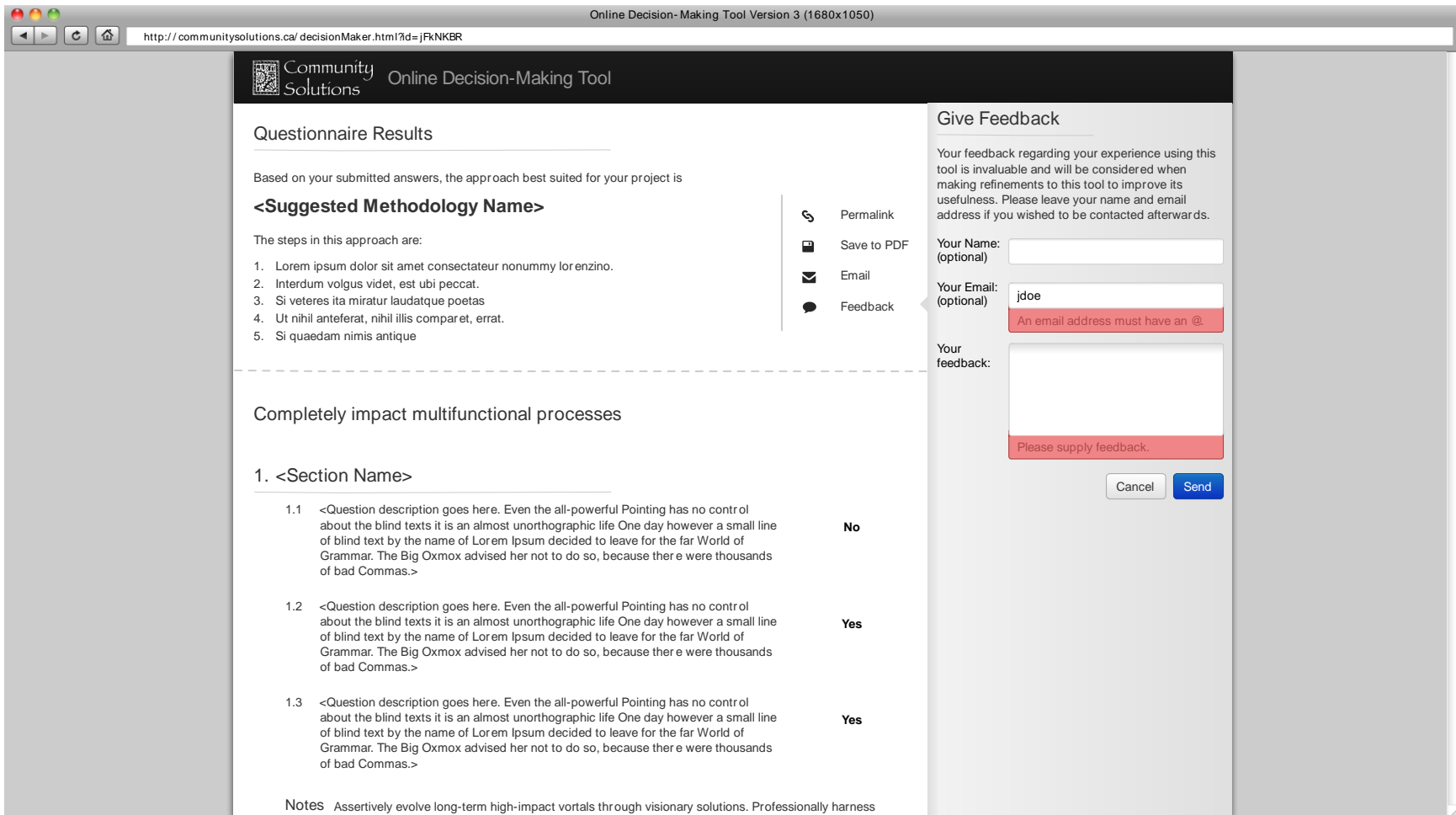


Figure 48 – Give feedback form validation on the questionnaire results page

## Questionnaire Results Mock-up Version 4

This version of the mock-up shows a variation of the sidebar that avoids the problems illustrated in version 3 of the questionnaire results mock-up namely hiding the actions box and moving the text field labels above the text field to save space.

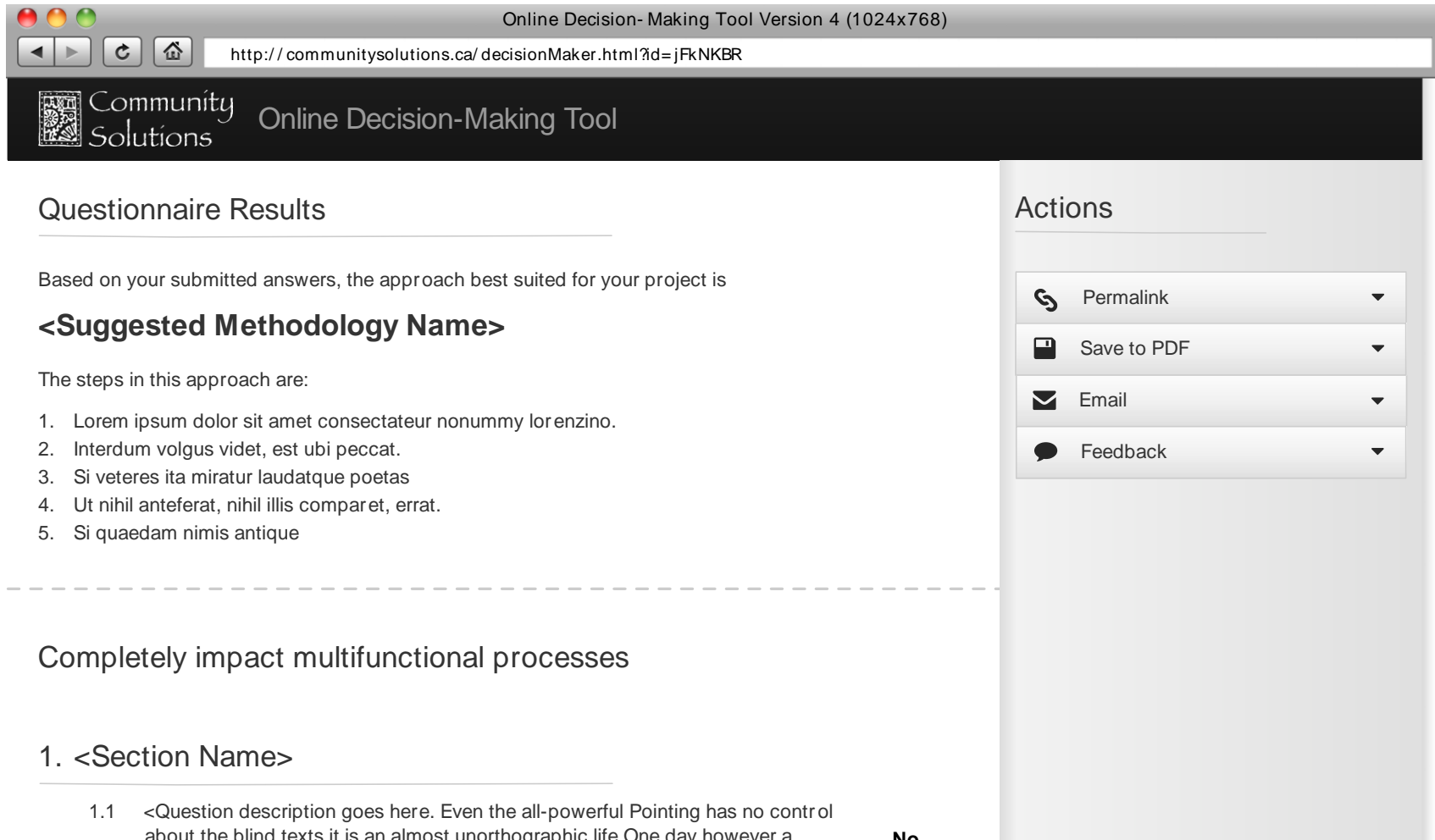


Figure 49 – Mock-up version 4 of the questionnaire results page

The triangle icon beside each accordion (the element with expandable and collapsible sections under the actions heading) heading will change when a section is expanded or collapsed as a visual cue. The heading inside each accordion section is removed in the implemented web app as it is redundant.

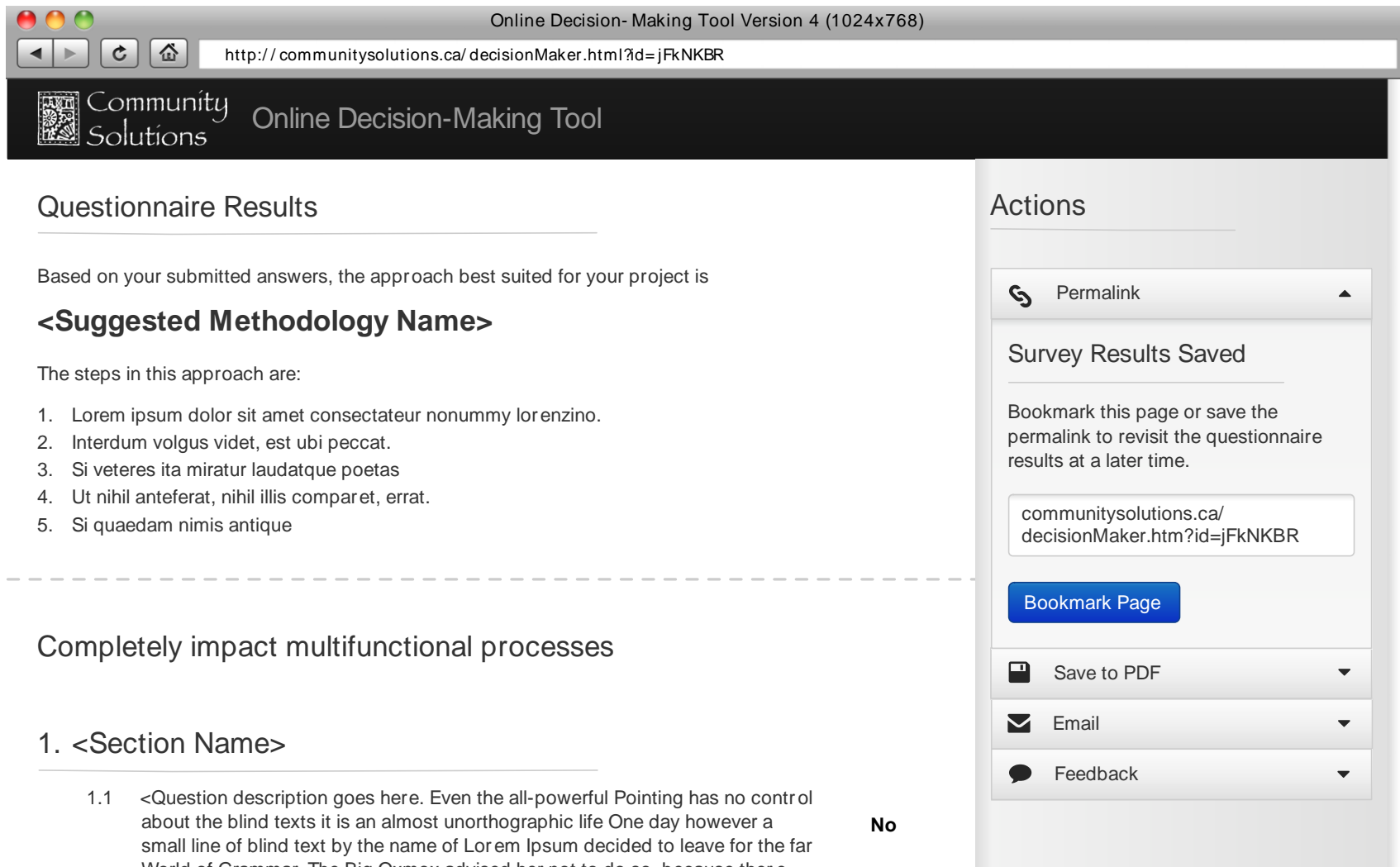


Figure 50 – Results link form on the questionnaire results page

The “Save to PDF” accordion heading has special logic applied in the implemented web app so clicking on the heading will not expand the section. The triangle icon is also hidden for this section as a visual cue that this section is not expandable. If JavaScript is not enabled, the accordion will “flatten” and all sections will be expanded. There will be a link in the “Save to PDF” section to generate the questionnaire results PDF.

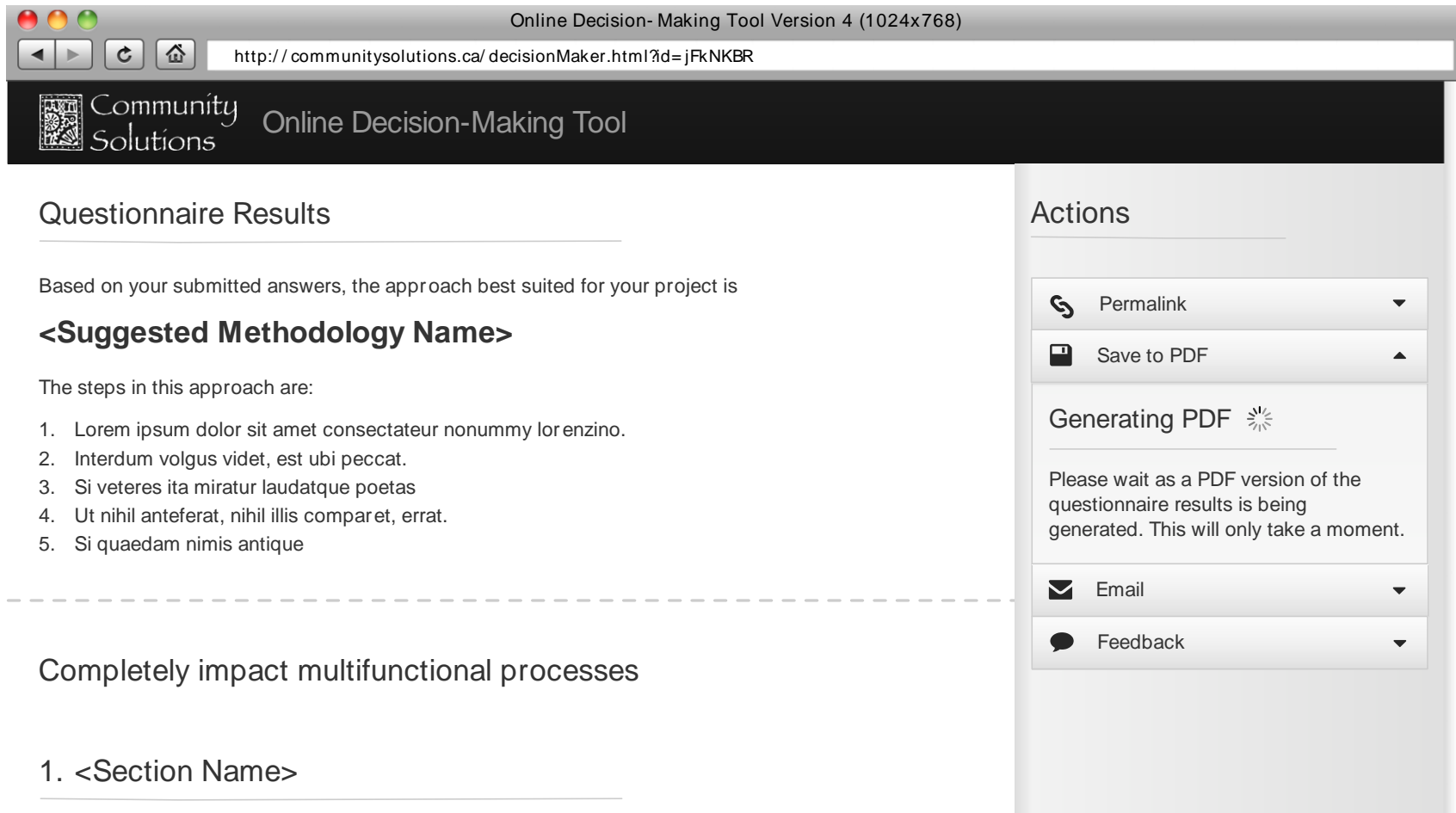


Figure 51 – Save to PDF message on the questionnaire results page

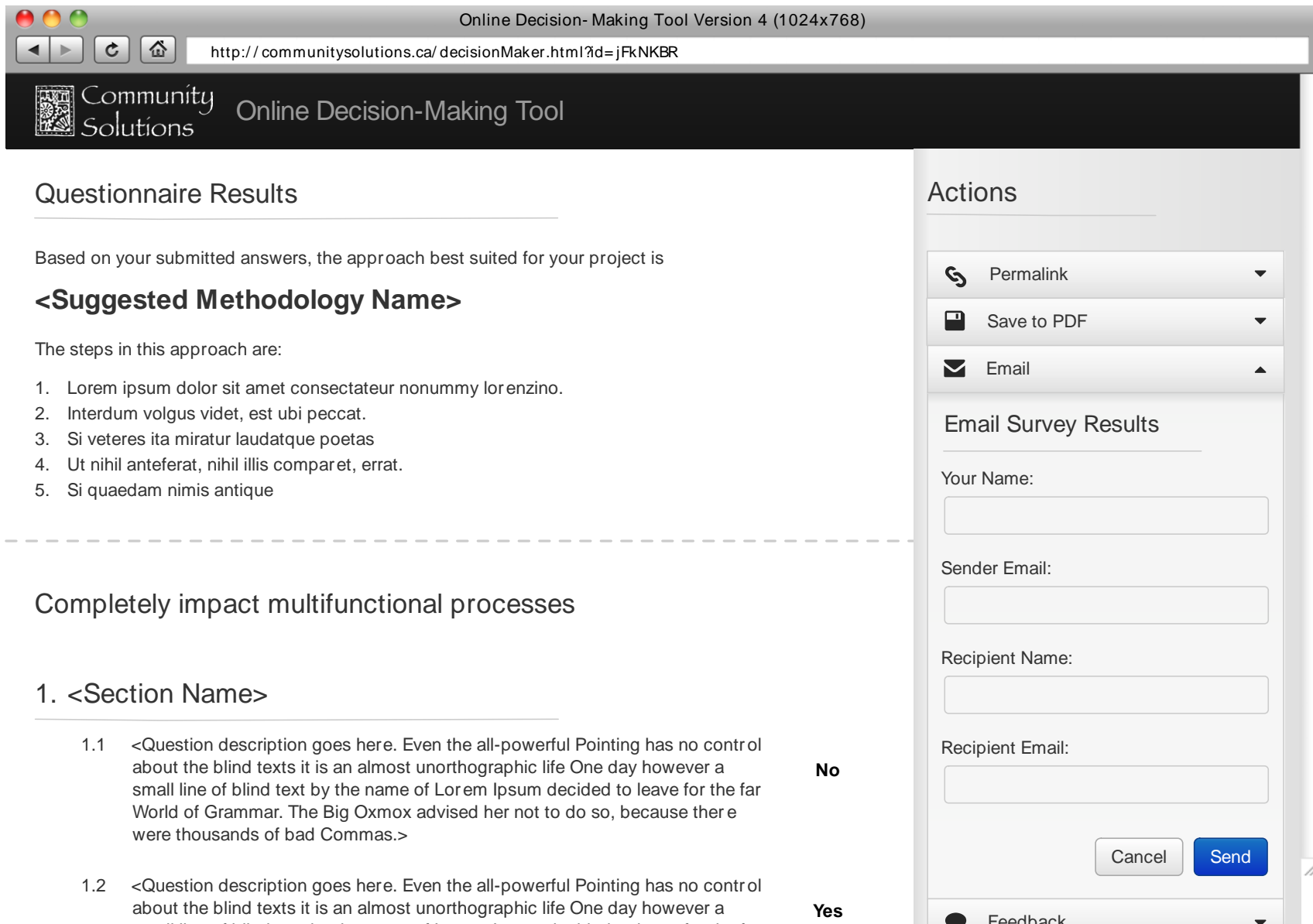


Figure 52 – Email results form on the questionnaire results page

If there is enough vertical screen estate to show the whole accordion, the contents of the sidebar will be scrollable.

## Questionnaire Results

Based on your submitted answers, the approach best suited for your project is

### <Suggested Methodology Name>

The steps in this approach are:

1. Lorem ipsum dolor sit amet consectetur nonummy loremipsum.
2. Interdum volgus videt, est ubi peccat.
3. Si veteres ita miratur laudatque poetas
4. Ut nihil anteferat, nihil illis comparet, errat.
5. Si quaedam nimis antique

---

## Completely impact multifunctional processes

### 1. <Section Name>

- 1.1 <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>
- 1.2 <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there

No

Yes

The image shows a feedback form overlay on the right side of the questionnaire results page. The form is titled "Feedback" and contains the following elements:

- A header "Give Feedback" with a horizontal line below it.
- A paragraph of text: "Your feedback regarding your experience using this tool is invaluable and will be considered when making refinements to this tool to improve its usefulness. Please leave your name and email address if you wished to be contacted afterwards."
- A label "Your Name (optional):" followed by a text input field.
- A label "Your Email (optional):" followed by a text input field.
- A label "Your Feedback:" followed by a large text area for input.
- At the bottom right, there are two buttons: "Cancel" (light gray) and "Send" (blue).

Figure 53 – Give feedback form on the questionnaire results page

## Questionnaire Results Mock-up Version 5

This version of the mock-up shows another variation of the sidebar shows form contents inside a popover. This version of the mock-up illustrates many flaws with this design and was therefore scrapped very quickly.

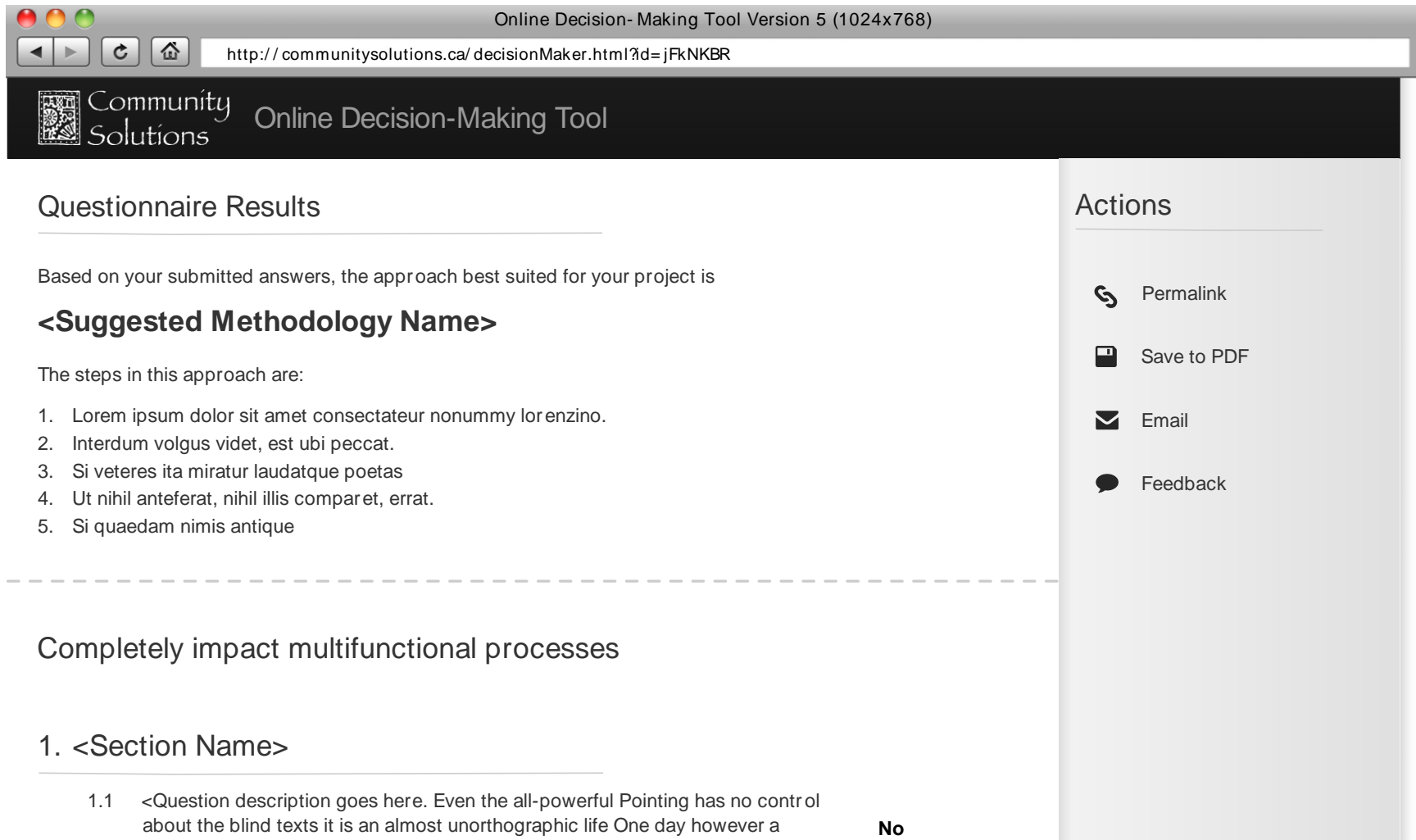


Figure 54 – Mock-up version 5 of the questionnaire results page

In the English language, people read from left to right. Clicking on the action link on the right and then having the form appear on the left is awkward.

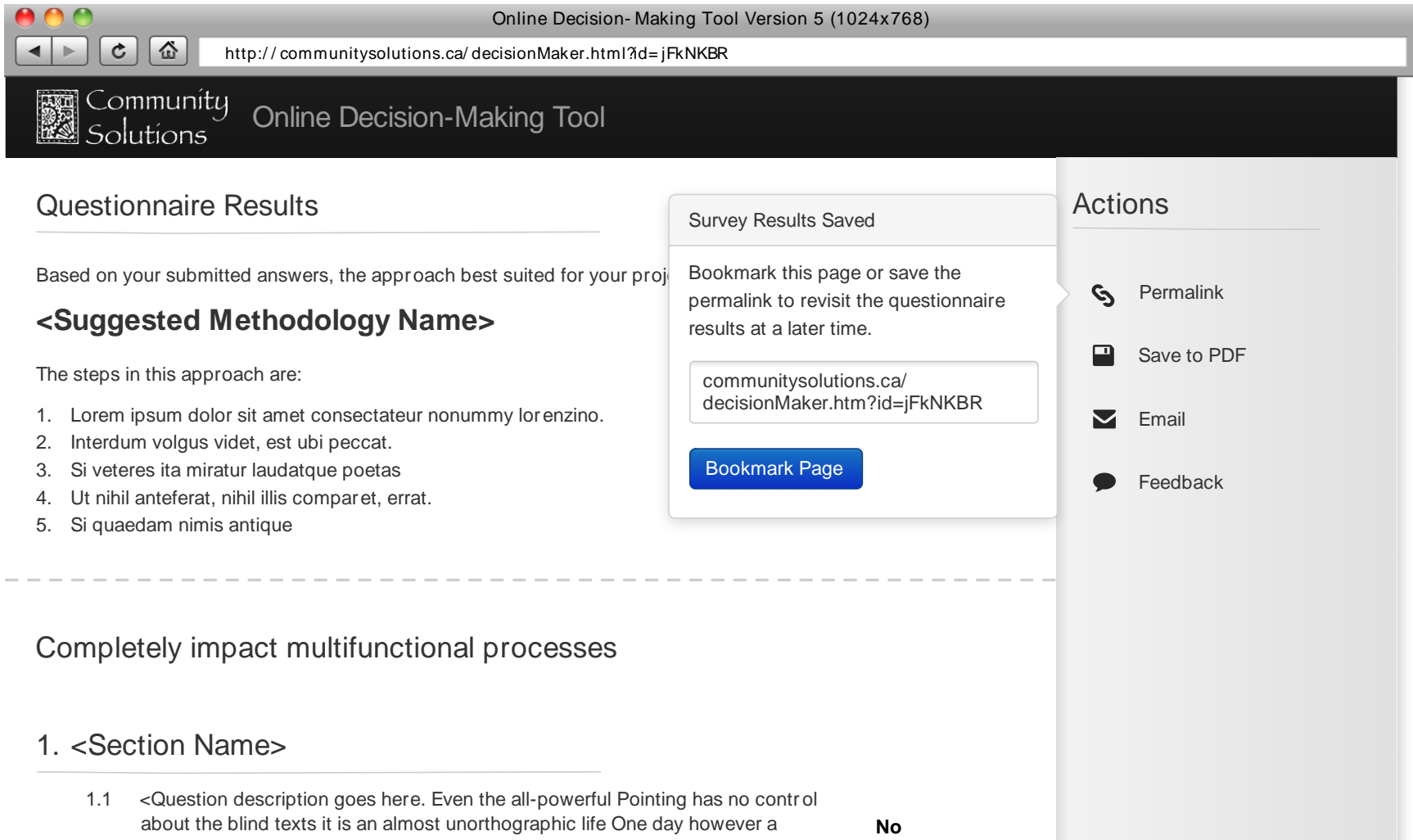


Figure 55 – Results link form on the questionnaire results page

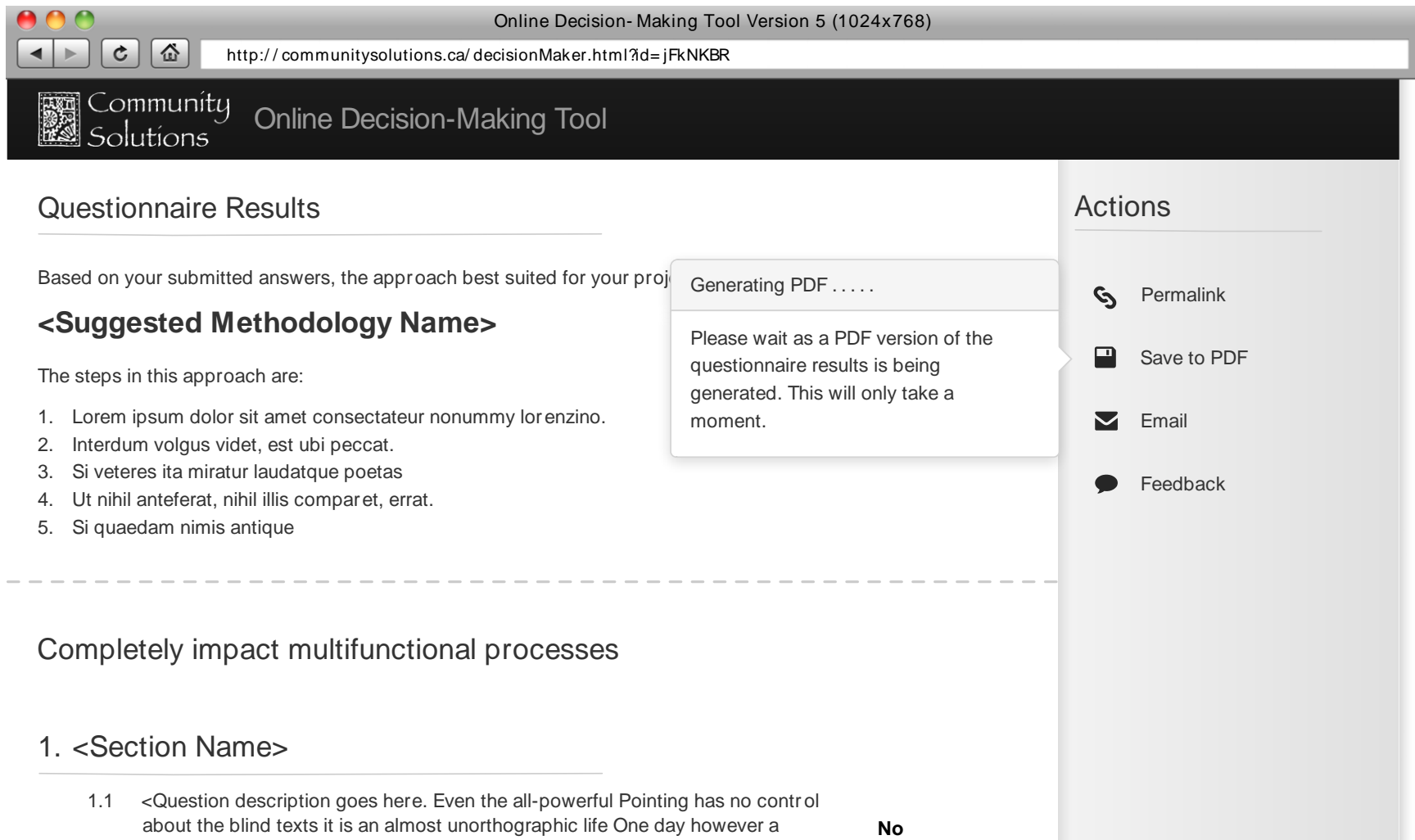


Figure 56 – Save to PDF message on the questionnaire results page

Users may panic if they misclick outside of the popover and the popover closes thinking that their entered data is lost.

Worst of all, popovers have no no-JavaScript fallback mode, they do not have a function that resizes them vertically if there is not enough vertical screen estate to fit the whole popover, and they do not have a function to reposition itself relative to the screen if the screen resizes horizontally.

## Questionnaire Results

Based on your submitted answers, the approach best suited for your project is:

### <Suggested Methodology Name>

The steps in this approach are:

1. Lorem ipsum dolor sit amet consectetur nonummy loremzino.
2. Interdum volgus videt, est ubi peccat.
3. Si veteres ita miratur laudatque poetas
4. Ut nihil anteferat, nihil illis comparet, errat.
5. Si quaedam nimis antique

---

## Completely impact multifunctional processes

### 1. <Section Name>

- 1.1 <Question description goes here. Even the all-powerful Pointillism about the blind texts it is an almost unorthographic life One day a small line of blind text by the name of Lorem Ipsum decided to enter the World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>
- 1.2 <Question description goes here. Even the all-powerful Pointillism about the blind texts it is an almost unorthographic life One day a small line of blind text by the name of Lorem Ipsum decided to enter the World of Grammar. The Big Oxmox advised her not to do so, because there

Survey Results Saved

Your feedback regarding your experience using this tool is invaluable and will be considered when making refinements to this tool to improve its usefulness. Please leave your name and email address if you wished to be contacted afterwards.

Your Name (optional):

Your Email (optional):

Your Feedback:

Email

Feedback

Figure 57 – Give feedback form on the questionnaire results page

## Questionnaire Mock-up Mobile Version

This mock-up illustrates the mobile phone version of the web app. The bookmark button in the header got changed to a print to PDF button in the implemented version as some web browsers such as Chrome and Safari forbid JavaScript from activating the bookmark page feature due to security concerns.

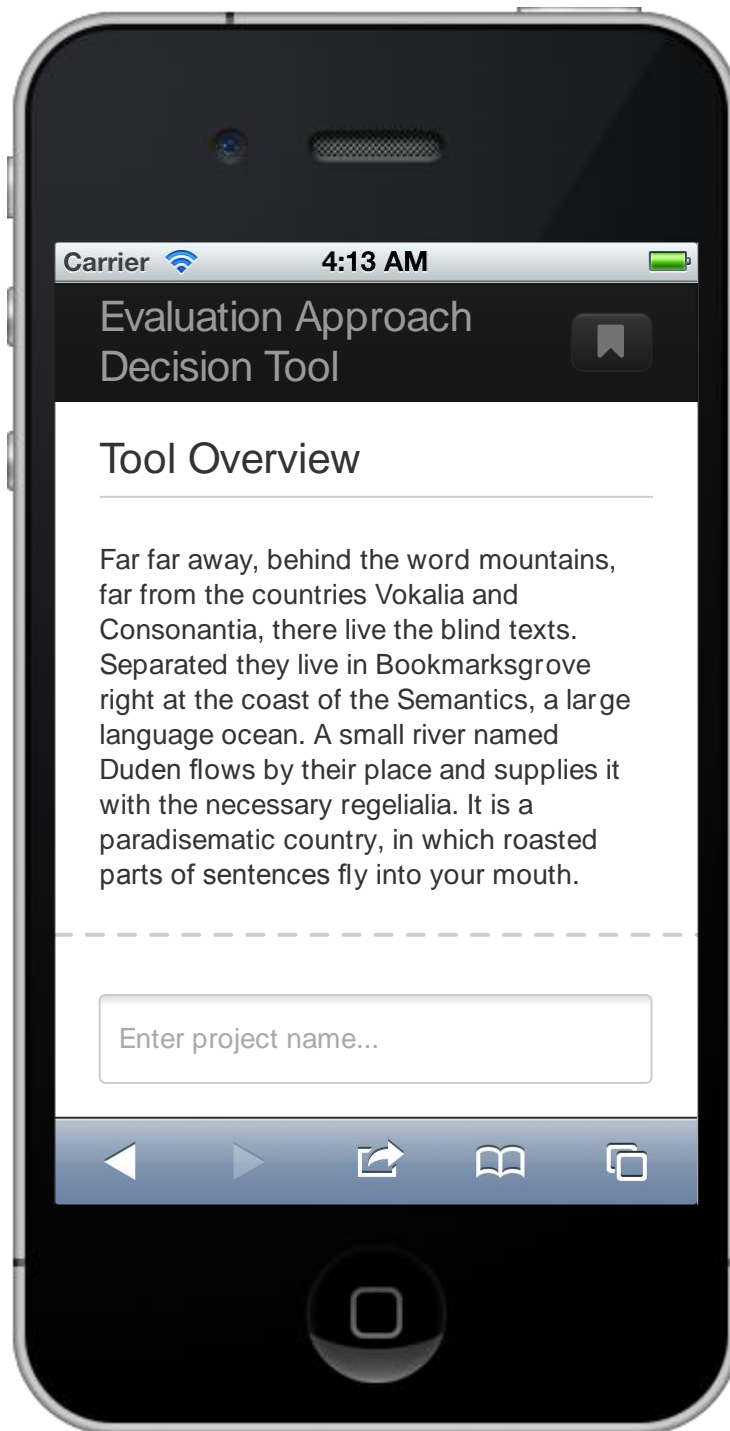


Figure 58 – Mobile version of the questionnaire page

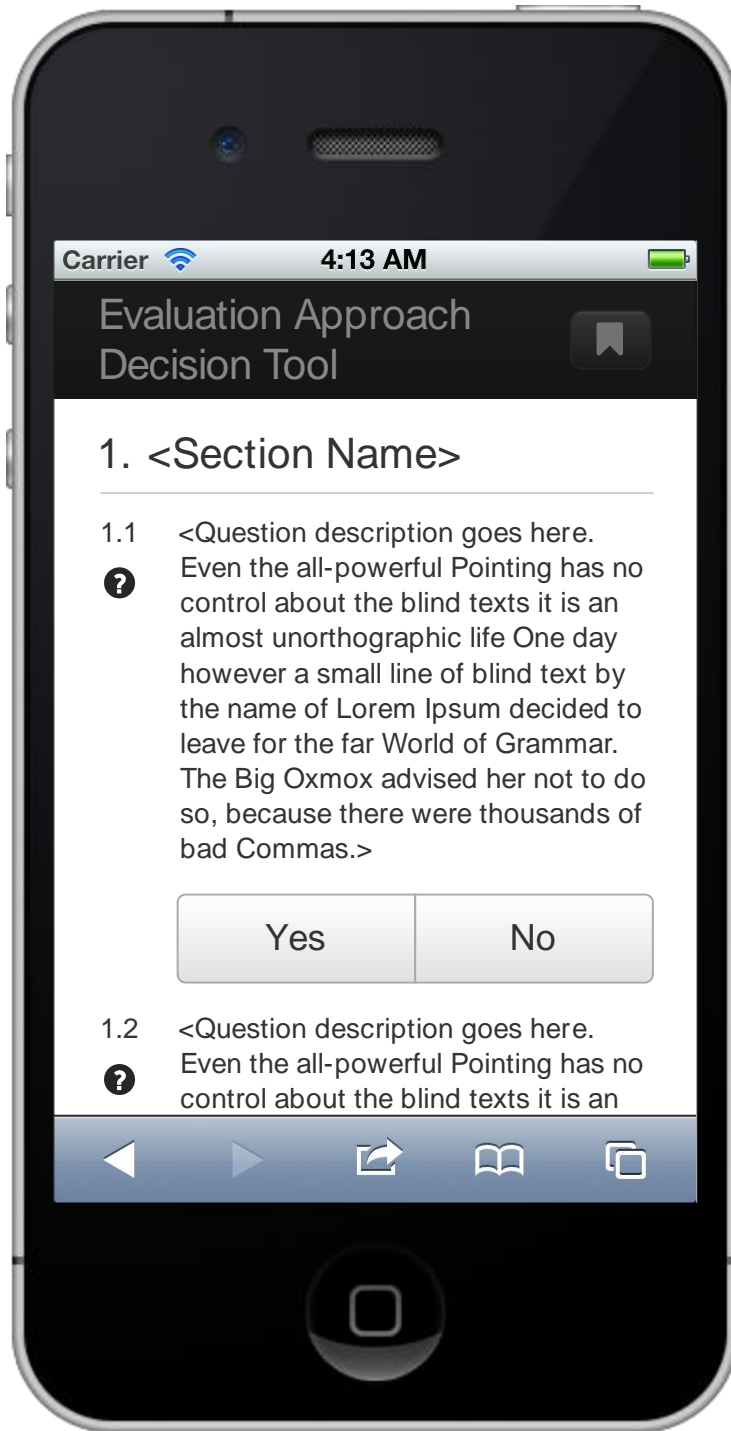


Figure 59 – Mobile version of the questionnaire page

The error highlighting got toned down in the web app as the red was too bright. The error highlighting box styling was changed so it surrounds the whole question just like in the desktop version.

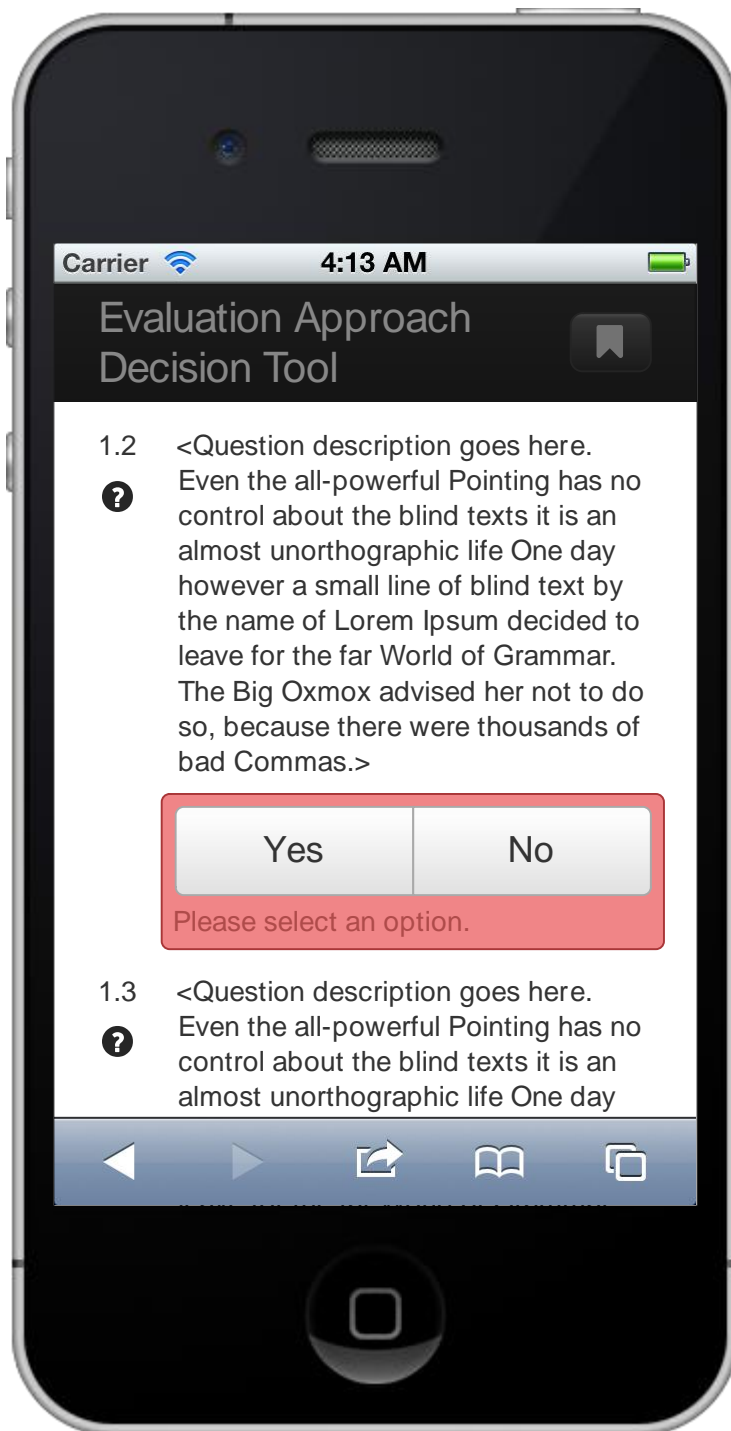


Figure 60 – Question error highlighting on the questionnaire page

JavaScript logic is added to resize the height of the modal if there is not enough vertical screen estate to show the whole modal.

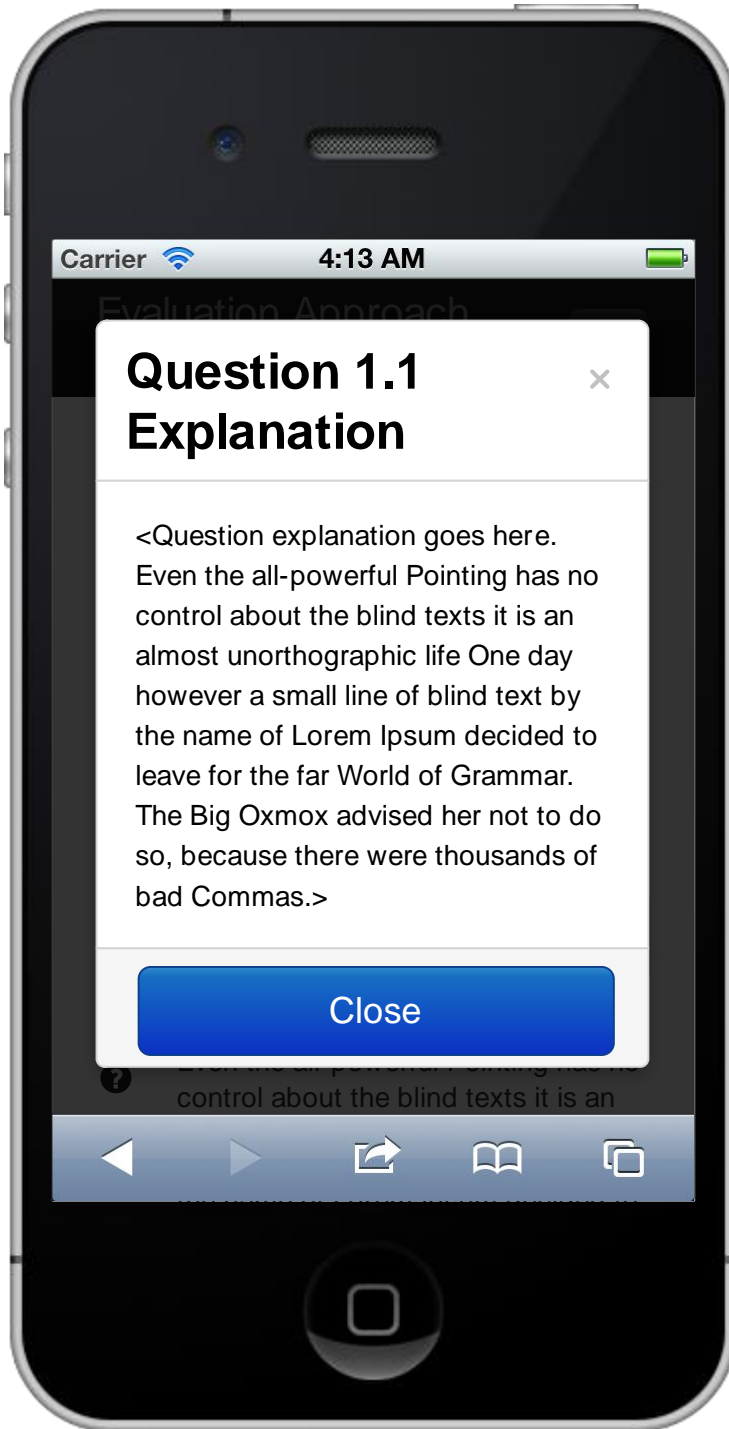


Figure 61 – Information modal on the questionnaire page

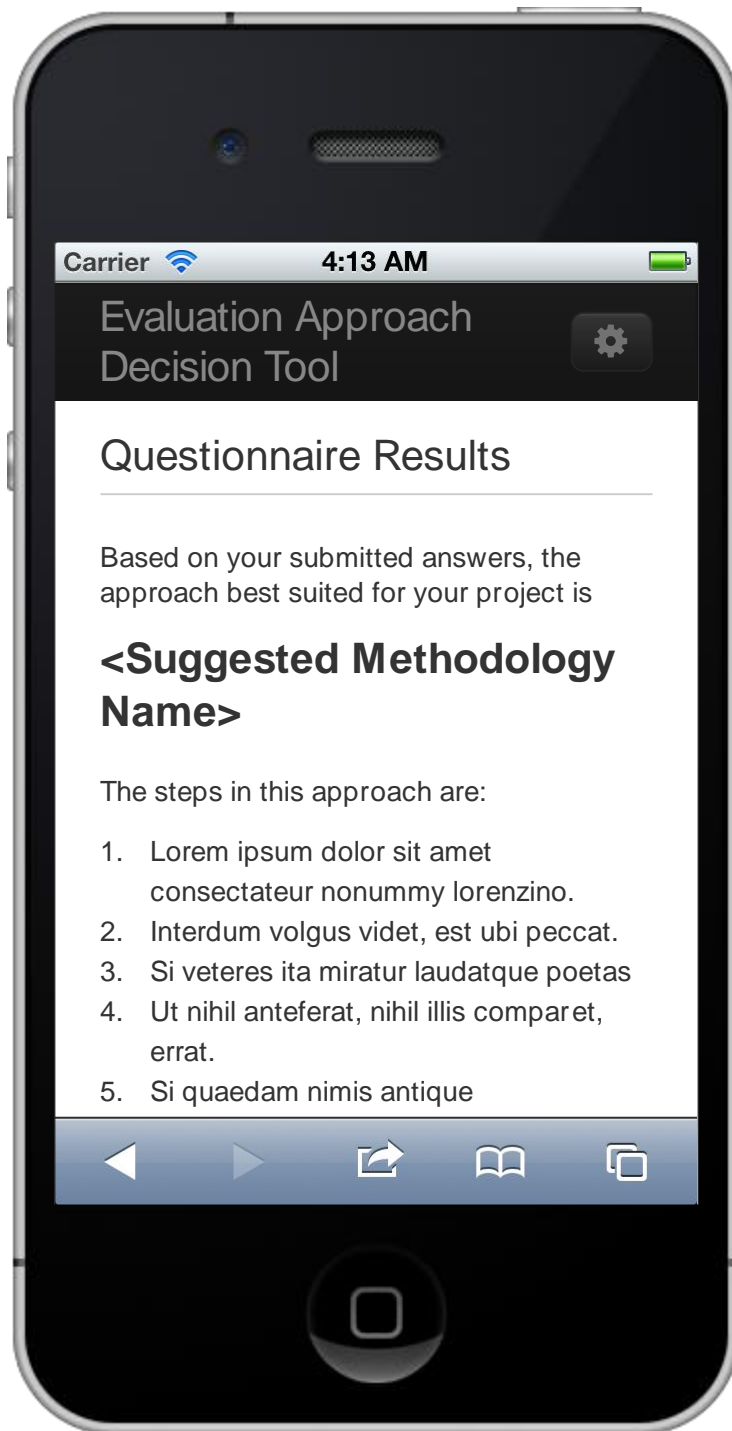


Figure 62 – Mobile version of the questionnaire results page

The email results button is added for convenience on phone and tablet sized screens in case users do not see the actions button (the gear in the top right corner).

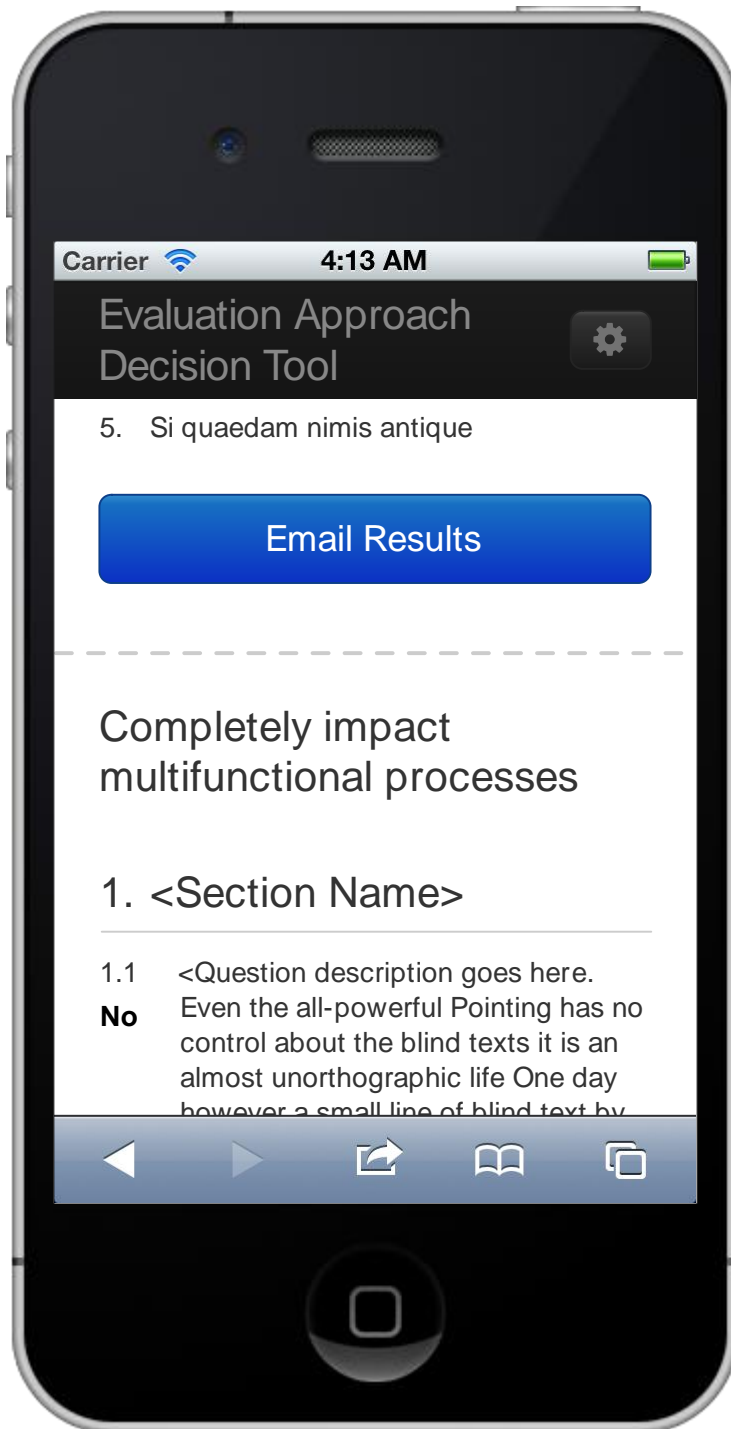


Figure 63 – Mobile version of the questionnaire results page

The “Yes” and “No” responses are moved to the left underneath the question number on phone sized screens to save space.

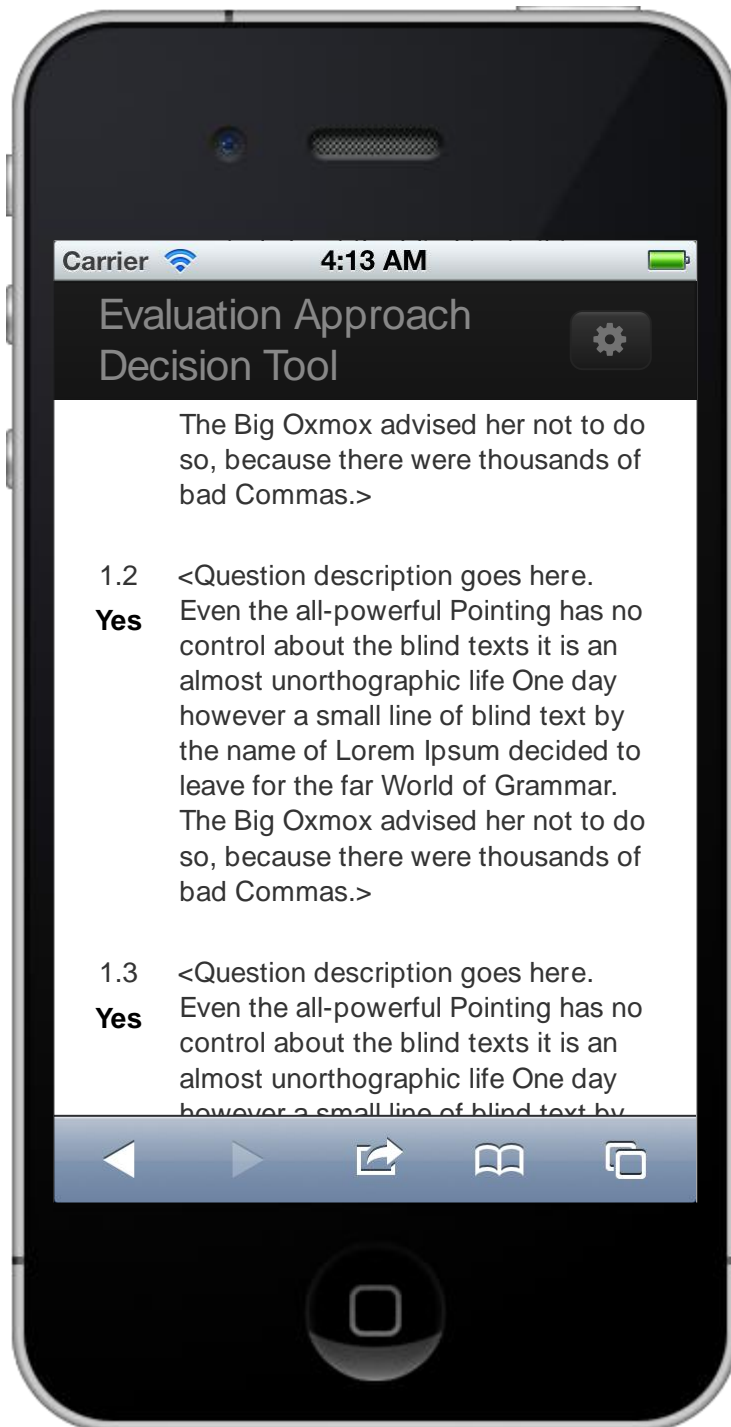


Figure 64 – Mobile version of the questionnaire results page

The notes heading is changed to move above the notes text to avoid the two from overlapping if the screen becomes too narrow.

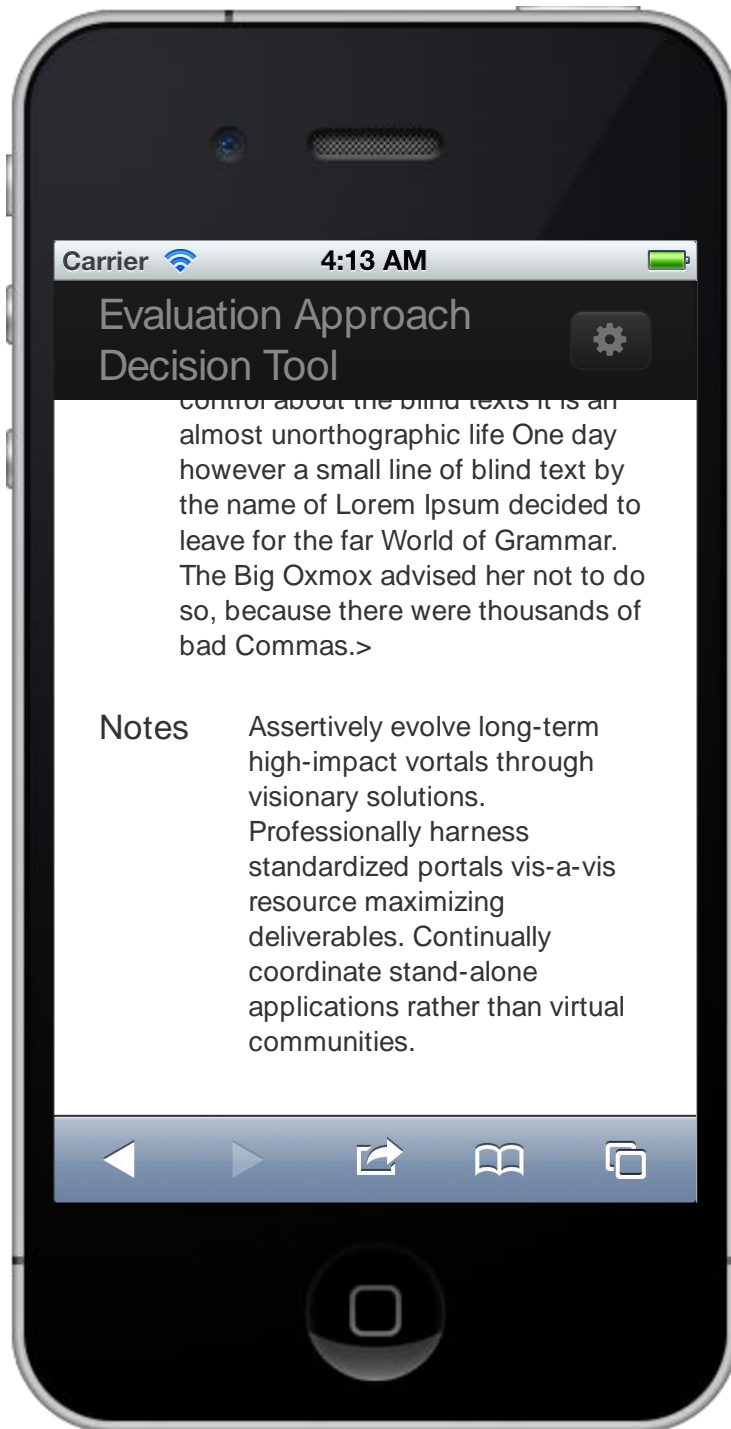


Figure 65 – Mobile version of the questionnaire results page

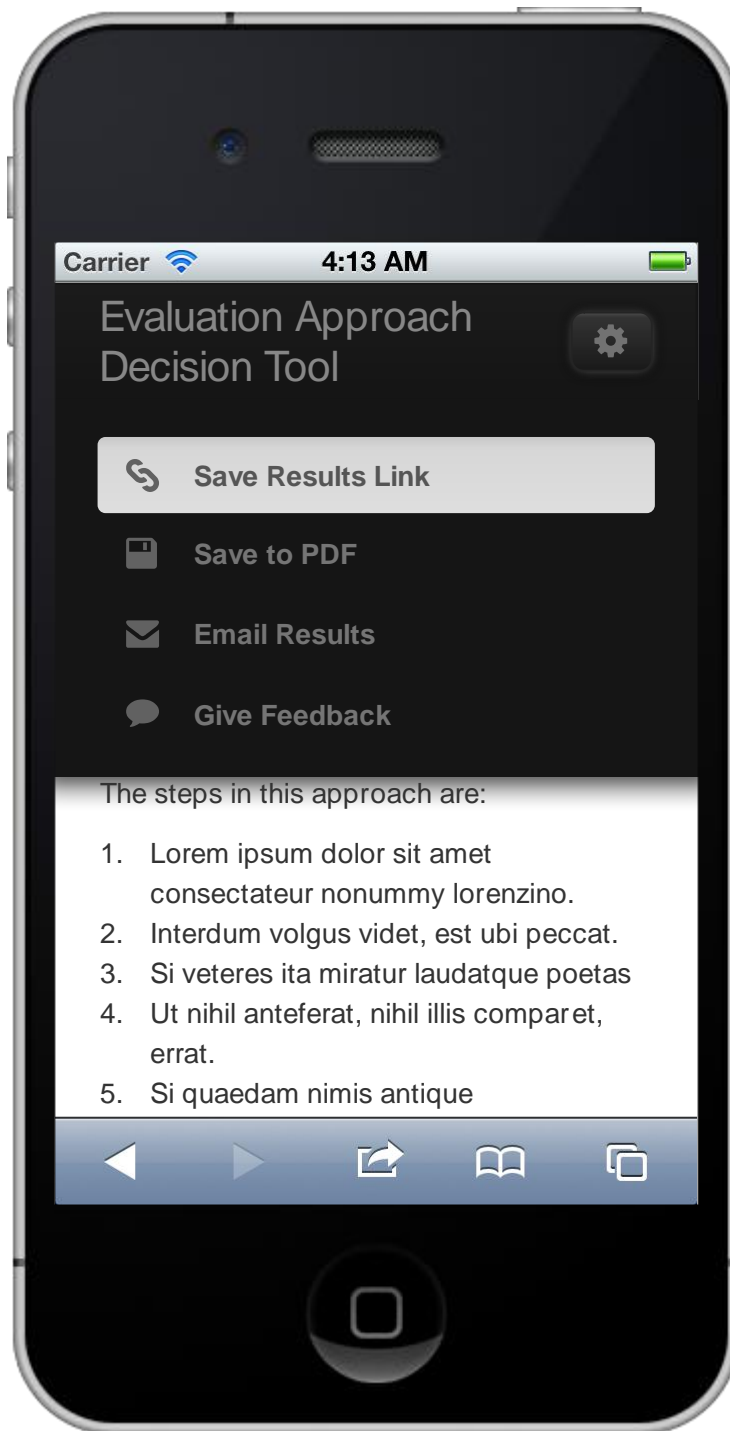


Figure 66 – Actions menu on the questionnaire results page

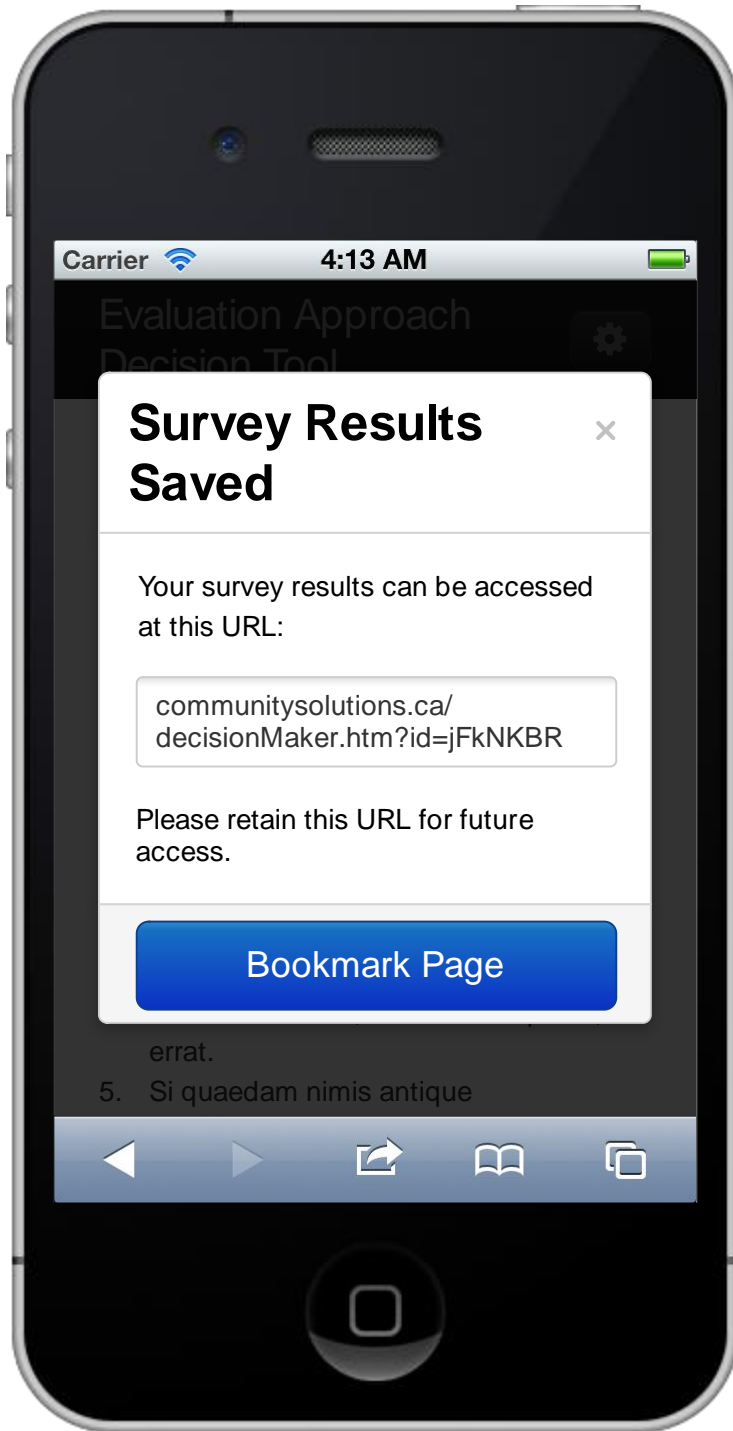


Figure 67 – Results link form on the questionnaire results page

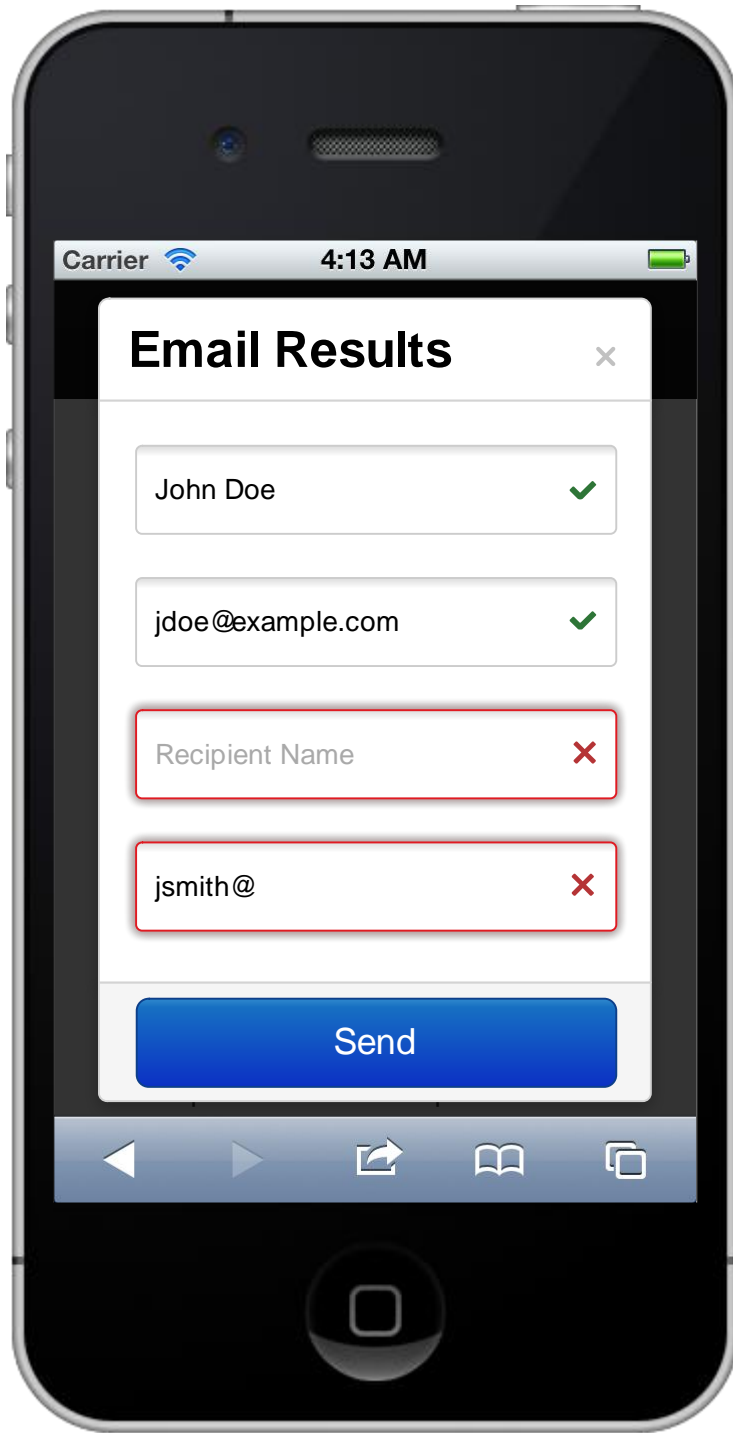


Figure 68 – Email results form validation on the questionnaire results page

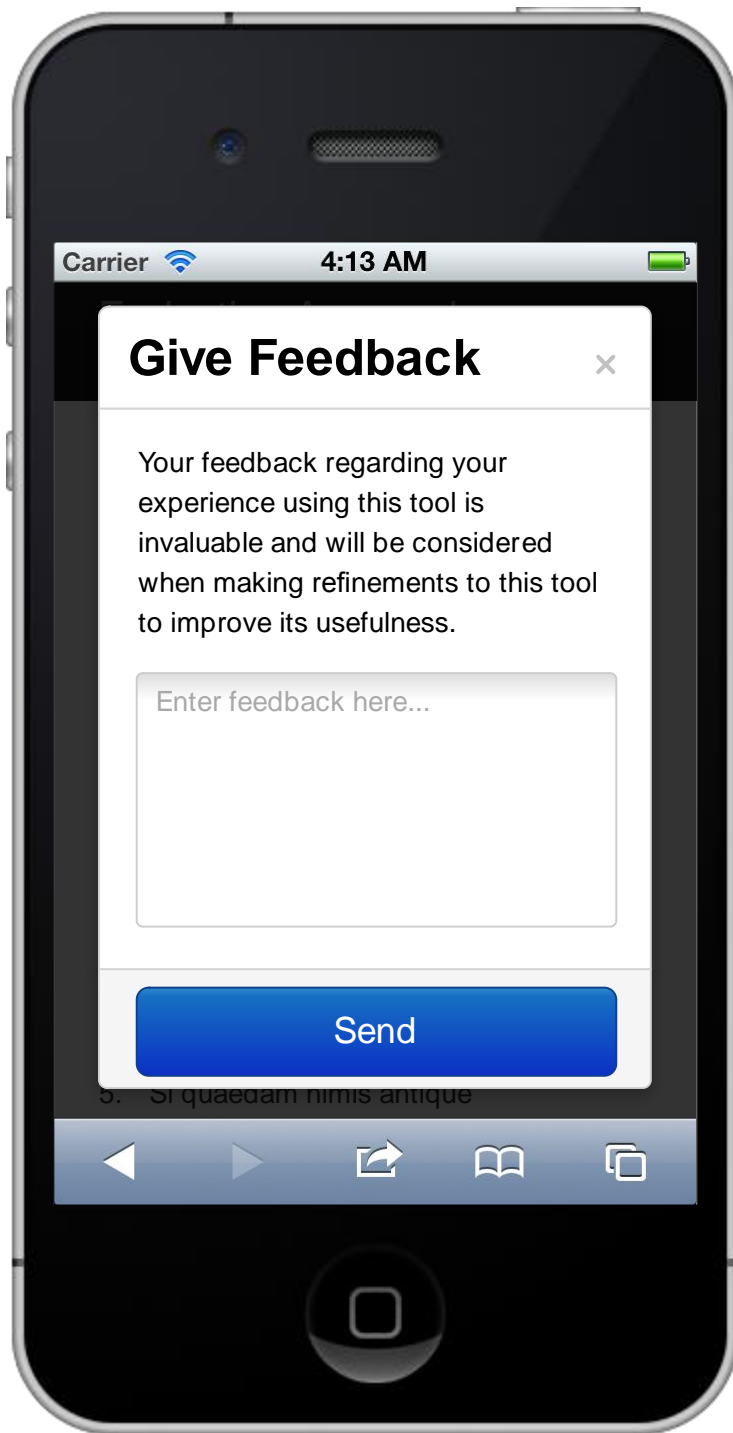


Figure 69 – Give feedback form on the questionnaire results page

## Appendix B – Database Tables with Data

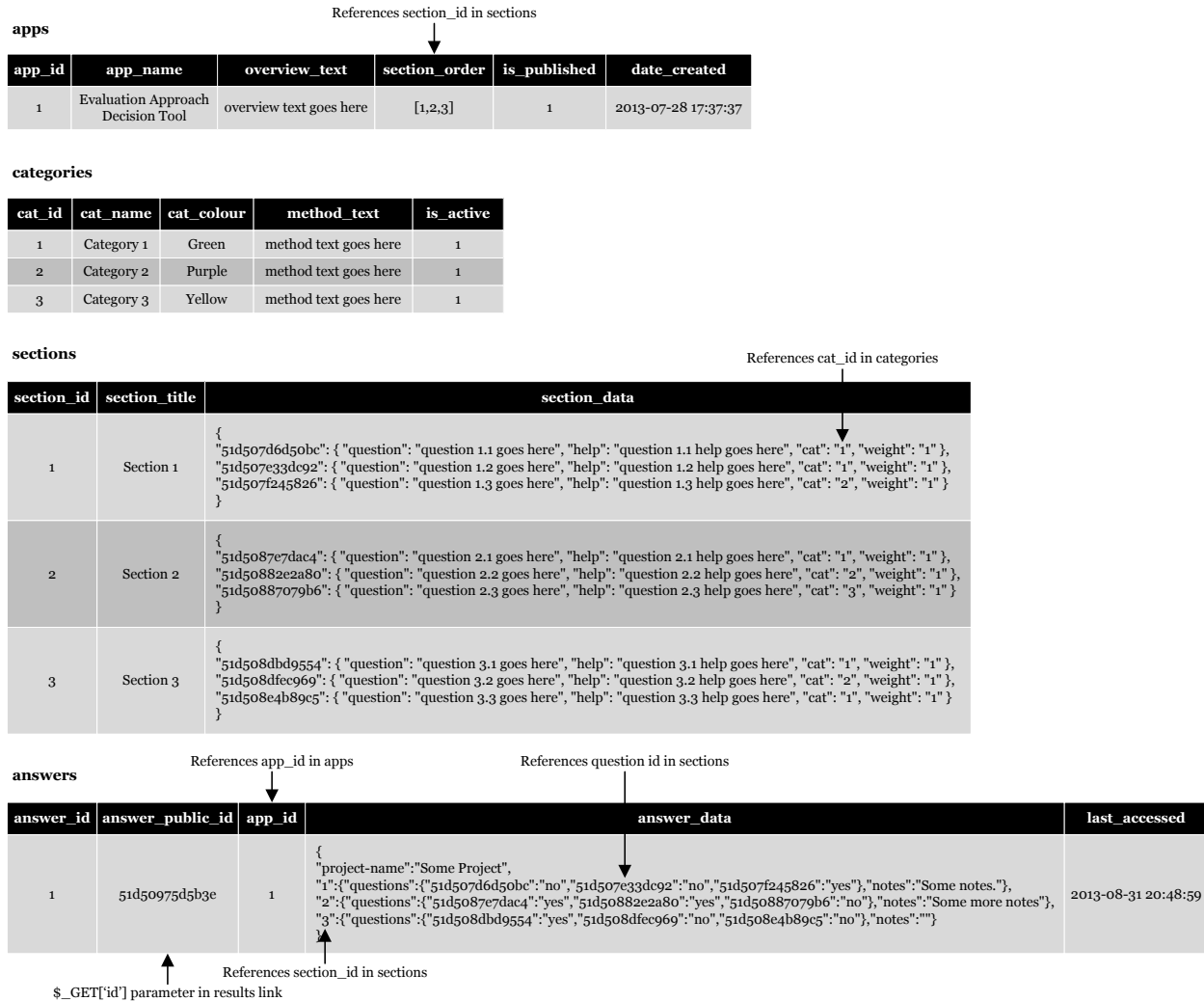


Figure 70 – Decision tool database schema with sample data

## Appendix C – User Manual

### Web Server and Database Installation

1. Download the latest version of MAMP from “www.mamp.info”
2. Install the downloaded package to your current drive.
3. Run “MAMP.app” in “/Applications/MAMP” or “MAMP PRO.app” in “/Applications/MAMP PRO”.

If you are running MAMP:

- 3.1 Click “Preferences...”.
- 3.2 In the “Ports” tab, ensure that the Apache port is set to 8888 and the MySQL port is set to 8889.

- 3.3 In the PHP tab, ensure that the PHP version is set to 5.4.10.
- 3.4 In the Apache tab, ensure that the document root is set to “/Applications/MAMP/htdocs”.
- 3.5 Click the “OK” button to save the changes.
- 3.6 Click the “Start Servers” button.

If you are running MAMP Pro:

- 3.1 Click “Server”.
- 3.2 In the “General” tab, ensure that the Apache port is set to 8888, the MySQL port is set to 8889, and the Apache/MySQL user is “www/mysql”.
- 3.3 In the “MySQL” tab, ensure that the “Allow local access only” checkbox is checked.
- 3.4 In the “PHP” tab, ensure that the PHP version is set to 5.4.10 and that the “Display” checkbox in the “Error handling” section is unchecked.
- 3.5 Click “Hosts”
- 3.6 Click the “+” button underneath the list of hosts.
- 3.7 On the right, in the “General” tab, specify a name for the entry under “Server Name”.
- 3.8 Set the port number to 80.
- 3.9 Set the disk location to “/Applications/MAMP/htdocs”.
- 3.10 In the “Advanced” tab, ensure that the “Indexes” checkbox in the “Directory” section is unchecked.
- 3.11 Click the “Apply” button at the bottom right of the window to save the changes.
- 3.12 Click the “Start” button in the toolbar.

### **Website and Database Configuration**

1. Extract the project files to “/Applications/MAMP/htdocs”.
2. Point your web browser to “localhost:8888/phpMyAdmin”.
3. Click the “Databases” tab.
4. Under “Create database” specify “decision\_tool” as the database name and ensure the database type is set to Collation.
5. Click the “Create” button to create the database.
6. Click the “Users” tab.
7. Click the “Add user” link.
8. In the “Login Information” section, specify “decision\_test” as the user name, “localhost” as the host, and “testing” as the password.
9. In the “Global privileges” section, check the select, insert, update, and delete checkboxes under data.
10. Click on “decision\_tool” in the left sidebar.
11. In the “Structure” tab, ensure that there are no tables in the database.
12. Click the “Import” tab.
13. Under the “File to Import” section, click the “Choose File” button.
14. In the file selection dialog, select “decision\_tool.sql” in “/Applications/MAMP/htdocs/decision-tool”.
15. Click the “Go” button to import the data.

### **Customizing the Questionnaire Overview**

1. Point your web browser to “localhost:8888/decision-tool/admin.php”.
2. Enter a name for the questionnaire under “Title”.
3. Enter the tool overview under “Tool Overview”. The tool overview is introductory text that appears above the questionnaire.

### **Adding Categories to the Questionnaire**

1. Point your web browser to “localhost:8888/decision-tool/admin.php”.
2. Click the “Add Category” button under the “Questionnaire Categories” section.
3. Enter a name for the category/methodology under “Category Name”.
4. Choose a colour for the category (Category colours are not currently shown in the questionnaire).
5. Enter the methodology description under “Method text”.
6. Click the “Save Changes” button to create the category.

### **Adding Sections and Questions to the Questionnaire**

1. Point your web browser to “localhost:8888/decision-tool/admin.php”.
2. Click the “Add Section” button under the “Questionnaire Sections” section.
3. Enter a title for the section under “Section Title”.
4. For each question, do the following:
  - 4.1 Assign a category/methodology for the question under “Category”.
  - 4.2 Enter a weighting for the question (when “Yes” is selected) under “Weight”.
  - 4.3 Enter the question description under “Question”.
  - 4.4 Optionally, enter the question information text under “Help Text”. The help text is shown when the information button beside each question is clicked.
5. Click on the “Add Question” button to add another question below.
6. Remove a question by clicking on the “X” button in the top right of each question.
7. Rearrange questions by clicking and dragging the grey boxes around each question.
8. Click the “Save Changes” button to create the section.

### **Editing Questionnaire Categories**

1. Point your web browser to “localhost:8888/decision-tool/admin.php”.
2. Change the category name by editing the title under “Category Name”.
3. Change the category colour by selecting another category colour under “Category Colour”.
4. Change the category’s methodology text by editing the methodology text under “Method Text”.

### **Editing Questionnaire Sections and Questions**

1. Point your web browser to “localhost:8888/decision-tool/admin.php”.
2. Click the title of the section that you want to edit under the “Questionnaire Sections” section.
3. Change the section title by editing the “Section Title” text field.
4. For each question you want to edit, do the following:
  - 4.1 Change the question’s assigned category by selecting another category under “Category”.
  - 4.2 Change the question’s weighting by editing the weight value under “Weight”.
  - 4.3 Change the question’s description by editing the description under “Question”.

- 4.4 Change the question’s information text by editing the information text under “Help Text”.
5. Click on the “Add Question” button to add another question below.
6. Remove a question by clicking on the “X” button in the top right of each question.
7. Rearrange questions by clicking and dragging the grey boxes around each question.
8. Click the “Save Changes” button to save the section changes.

## Appendix D – Usability Testing

### Desktop Usability Test

A desktop usability test was created using Usabilla and visitors of the graphic design section of 4chan and some of the client’s colleagues were asked to complete the usability test. Below are the questions and the associated heat maps. Changes to the design are made based on tester feedback.

The screenshot shows a Usabilla survey post on 4chan. The survey form includes a title, a description, a project name field, and a list of questions. The questions are:

1. I need some people to give feedback on some desktop website mockups. You can complete the usability test by clicking on the URL below:
2. Thank you for your help.

The URL provided is <http://participate.usabilla.com/83fecc6b072f0023608e588aa435e427ff6035ae>. Below the survey, there are several replies from anonymous users:

- >> **Anonymous** 05/06/13(Mon)16:45 UTC-7 No.111328 Replies: >>111624  
I need some people to give feedback on some desktop website mockups. You can complete the usability test by clicking on the URL below:  
<http://participate.usabilla.com/83fecc6b072f0023608e588aa435e427ff6035ae>  
Thank you for your help.
- >> **Anonymous** 05/07/13(Tue)22:41 UTC-7 No.111623  
answered  
hope that helps anon
- >> **Anonymous** 05/07/13(Tue)23:01 UTC-7 No.111624  
[>>111328 \(OP\)](#)  
Don't tell me what to do. /DeliveryBlackMan
- >> **Anonymous** 05/08/13(Wed)04:30 UTC-7 No.111669  
Done. Good luck with it all
- >> **Anonymous** 05/08/13(Wed)11:02 UTC-7 No.111757  
Submitted.

Figure 71 – Post on the graphic design 4chan board to get testers for a usability test

### Summary of Changes

Based on the usability test results and tester feedback, the following changes were incorporated into the implemented version of the web application.

- Added a save confirmation when the Next Section button is clicked.
- Replaced the bookmark page button with a save to PDF button.
- Reworded the actions sidebar labels to be more descriptive and to begin with a verb.

- Removed the heading inside each accordion section.
- Made the “Save to PDF” section as a button rather than an expandable section.

**Where will you click to name the project?**

It was clear to all the testers where they needed to click to specify a project name.

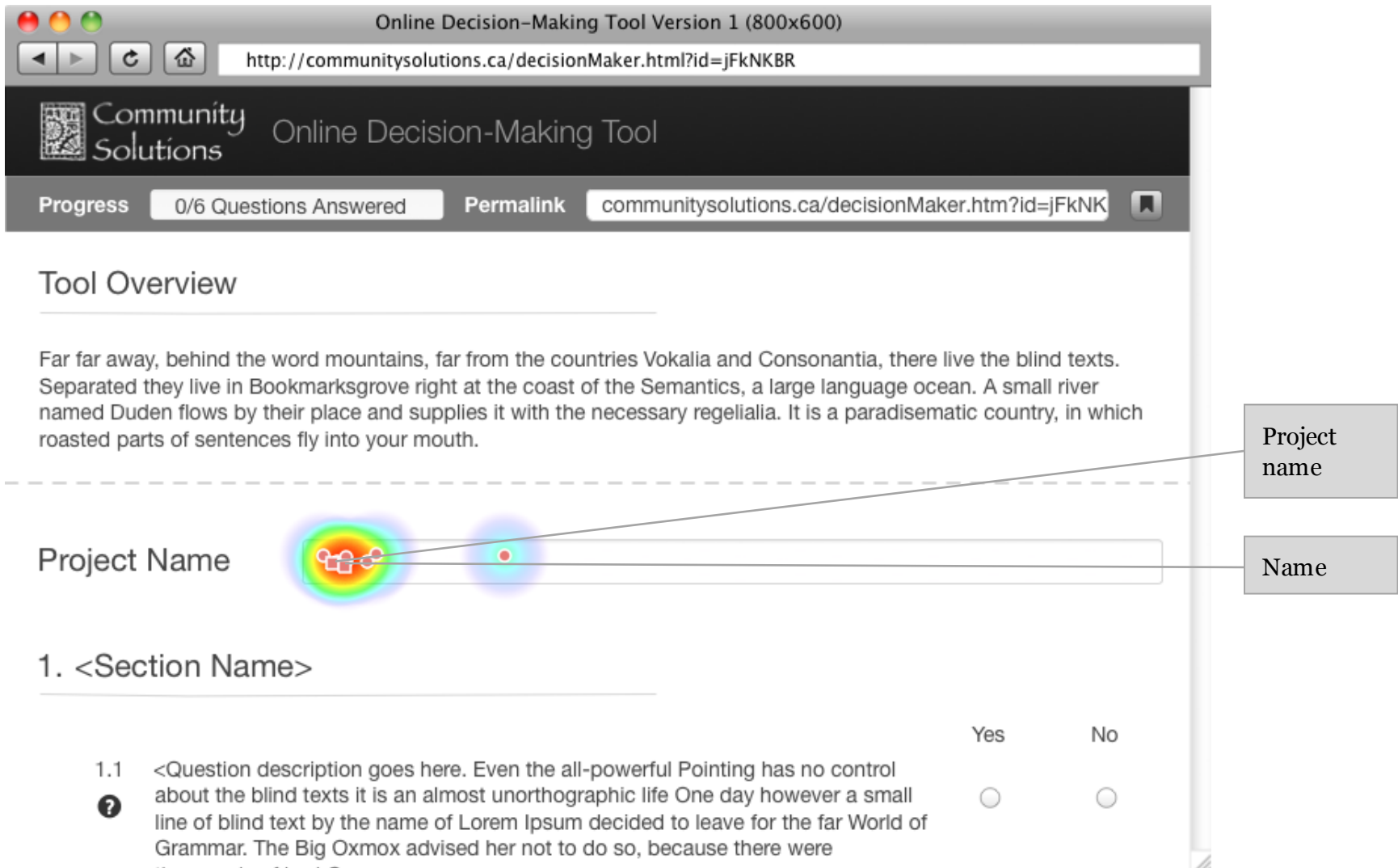


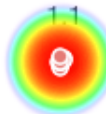
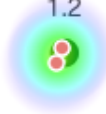

Figure 72 – Heat map showing where testers clicked when asked to specify a project name

**Where will you click to get more information about a question?**

It was clear to all the testers where they needed to click to get more information about a question.

Project Name

1. <Section Name>

	Yes	No
 1.1 <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>	<input type="radio"/>	<input type="radio"/>
 1.2 <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>	<input type="radio"/>	<input type="radio"/>
 1.3 <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>	<input type="radio"/>	<input type="radio"/>

Notes (Optional)

Next Section










Figure 73 – Heat map showing where testers clicked when asked to get more information about a question

**Where will you click to answer a question?**

It was clear to all the testers where they needed to click to answer a question.

Project Name

1. <Section Name>

- |  | Yes   | No  |
|--|---|---|
| 1.1  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.> |  |  |
| 1.2  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.> |  |  |
| 1.3  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.> |  |  |

Notes (Optional)

Next Section

Figure 74 – Heat map showing where testers dicked when asked to answer a question

**Where will you click to save the questionnaire to finish later?**

60% of the testers figured out that the page can be bookmarked to return to the questionnaire later. The other 40% of the testers did not know they can save the questionnaire and think they have to complete all of it in one session, which is okay too. Two of the testers mentioned that the function of the bookmark button is unclear.

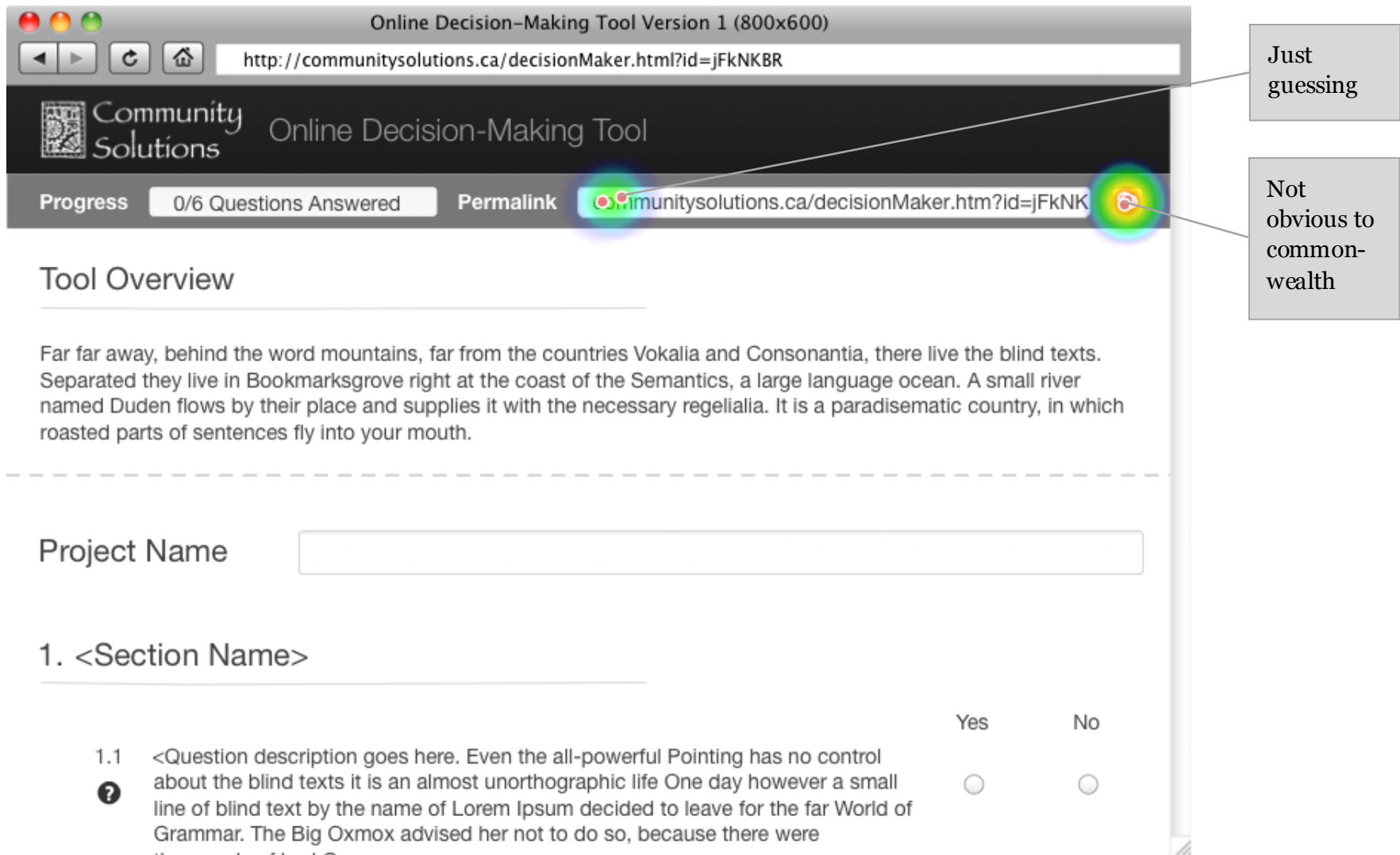





Figure 75 – Heat map showing where testers clicked when asked to save the questionnaire

**What do you expect will happen when the “Next Section” button is clicked?**

All of the testers figured out that clicking on the “Next Section” button will bring up the next set of questions. Only one tester had the expectation that the questionnaire responses will be saved in case of a session loss. This is improved by making a success message appear when the “Next Section” button is clicked to inform the user that the questionnaire is successfully saved.

Project Name

1. <Section Name>

- |  | Yes                   | No                    |
|--|-----------------------|-----------------------|
| 1.1  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.> | <input type="radio"/> | <input type="radio"/> |
| 1.2  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.> | <input type="radio"/> | <input type="radio"/> |
| 1.3  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.> | <input type="radio"/> | <input type="radio"/> |

Notes (Optional)



Page will load new questions or extend the length of the page with more content.

Page is saved in case of session loss and next set of questions appears

Section 2 with more questions will load.

This will open next section with new questions




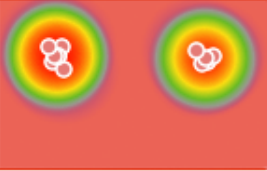

Figure 76 – Heat map showing where testers dicked when asked what the “Next Section” button does when clicked

**Where will you click to correct this error?**

All of the testers understood the error highlighting.

Project Name

### 1. <Section Name>

	Yes	No
1.1  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>	<input type="radio"/>	<input checked="" type="radio"/>
		
1.2  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>		
1.3  <Question description goes here. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas.>	<input checked="" type="radio"/>	<input type="radio"/>

Notes (Optional)

Figure 77 – Heat map showing where testers clicked when asked how to correct the submission error

**What do you expect will happen when each item in the actions sidebar is clicked?**

20% of the testers were unsure what some of the items in the actions sidebar means. This has been solved by making the labels beside each icon more descriptive and by rewording the labels to begin with a verb. For example, “Permalink” has been renamed to “Save Results Link”, “Email” has been renamed to “Email Results”, and “Feedback” has been renamed to “Give Feedback”. One tester suggested removing the heading inside each accordion section as it is redundant and making the “Save to PDF” section as a button.

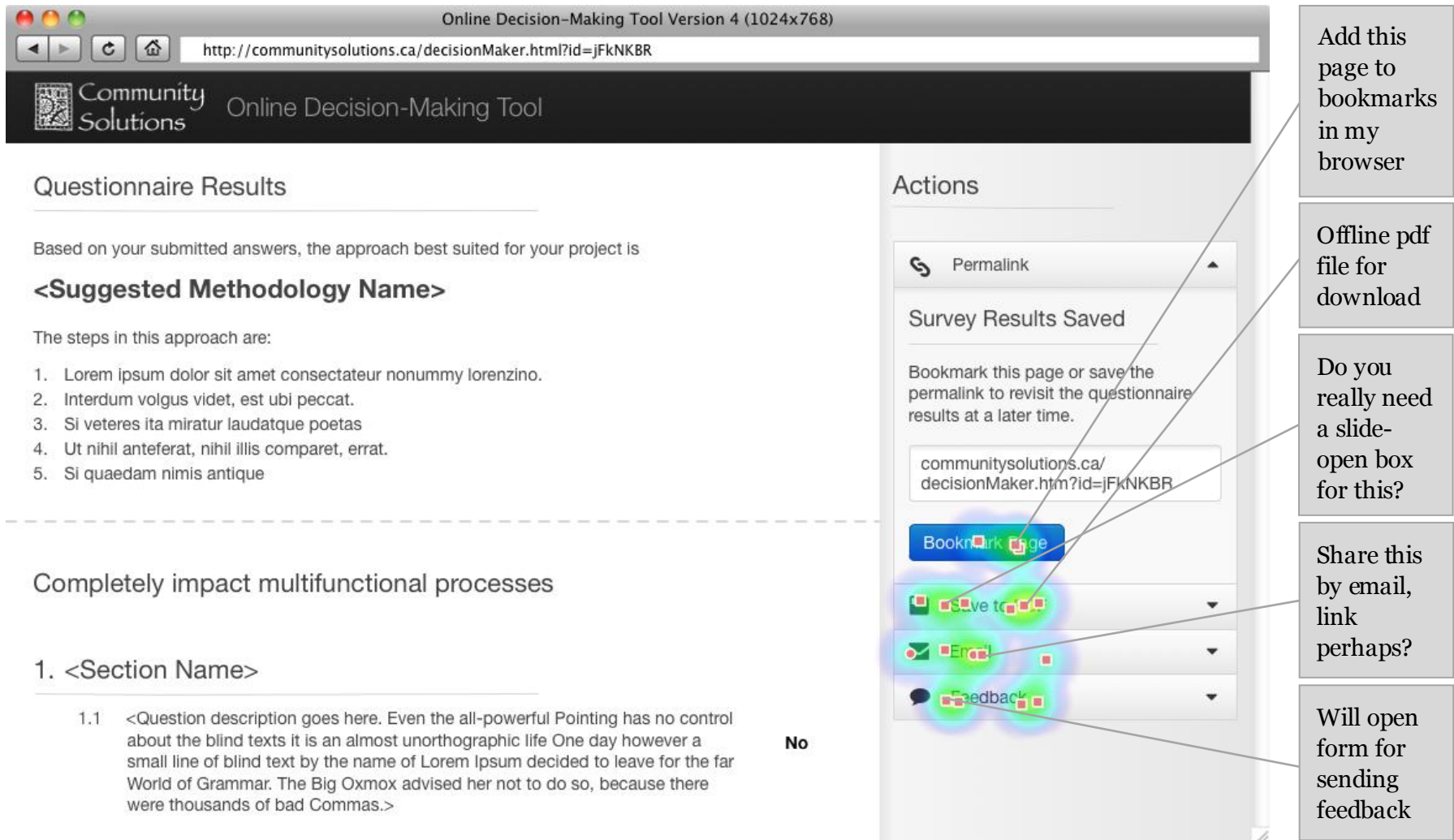


Figure 78 – Heat map showing where testers clicked when asked what each actions sidebar item does

**Which form fields do you think are required?**

All of the testers understood which fields are required.

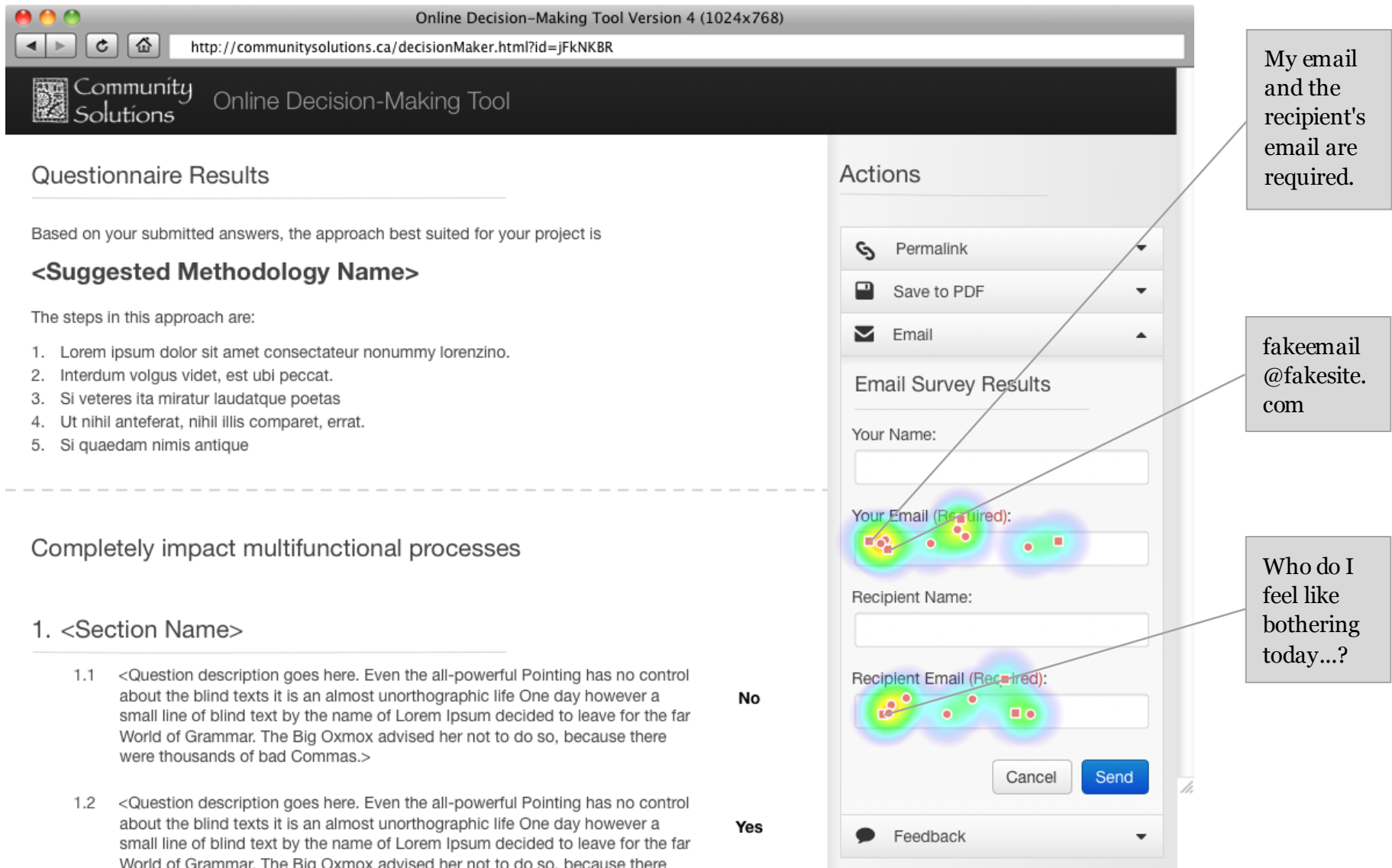


Figure 79 – Heat map showing where testers clicked when asked which form fields are required

**How can the input be corrected so the form will submit?**

All of the testers understood what they need to do to correct the error.

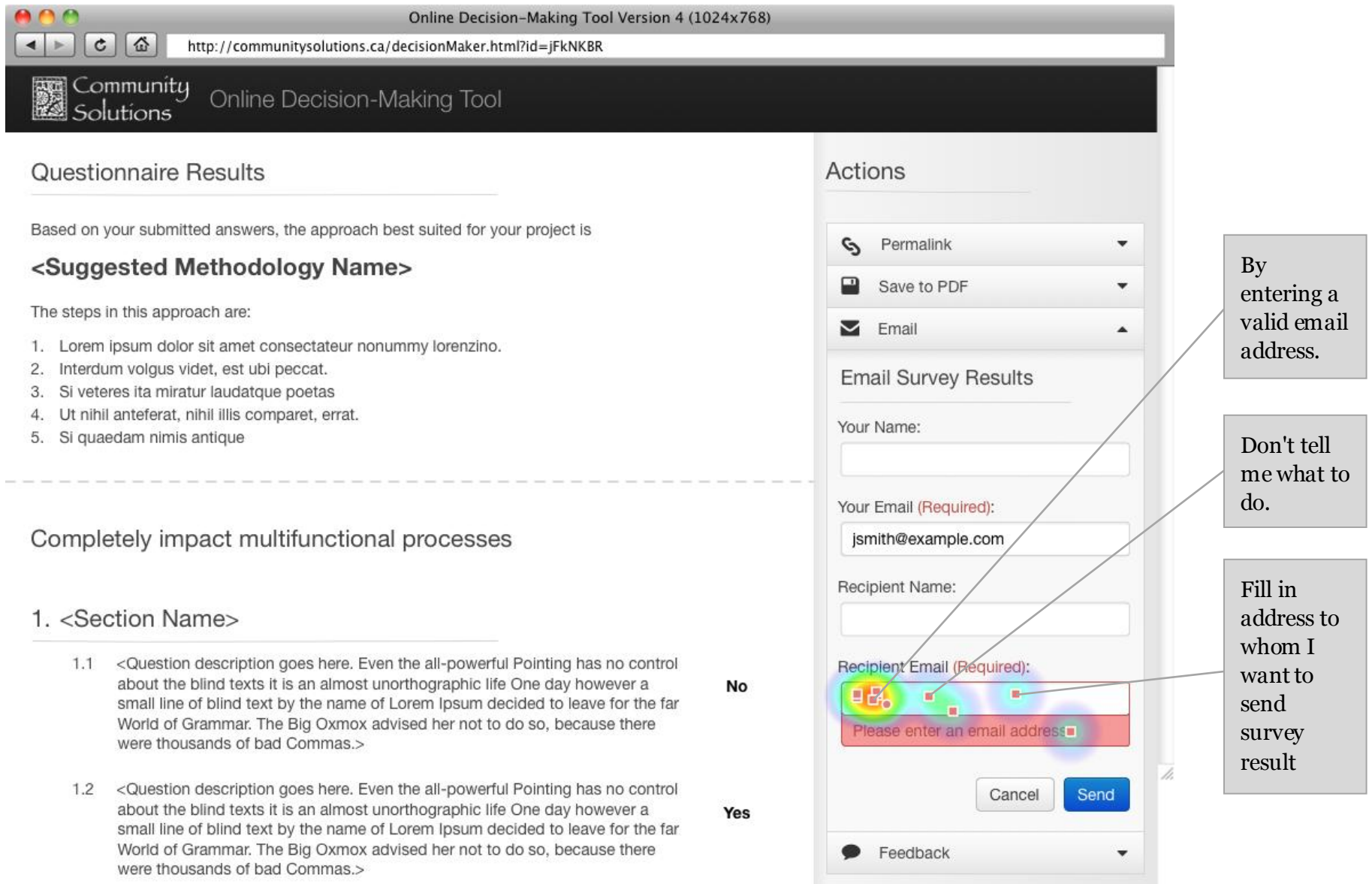



Figure 80 – Heat map showing where testers clicked when asked how to correct the form submission error

## **Mobile Usability Test**

A mobile usability test was created using Usabilla and visitors of the graphic design section of 4chan and some of the client's colleagues were asked to complete the usability test. Below are the questions and the associated heat maps. Changes to the design are made based on tester feedback.

File: [1369087239176.png](#) (130 KB, 365x713, 1.png)



**Need testers for a mobile usability test.** Anonymous 05/20/13(Mon)18:00 No.69983 Replies: >>71485

I need some people to give feedback on some mobile website mockups. You can complete the usability test by clicking on the URL below:

<http://participate.usabilla.com/ec9bd83754105ab4a7dcca9536935d7fb38856f>

Thank you for your help.

>>  **Anonymous** 05/23/13(Thu)12:55 No.71234

Is there anymore people willing to participate in this usability test? I need at least 5 people. This should only take about 10 minutes to complete.

>>  **Anonymous** 05/23/13(Thu)14:30 No.71280

I'll bite, gimme a sec.

>>  **Anonymous** 05/24/13(Fri)00:17 No.71452

done

>>  **Sorcerer** 05/24/13(Fri)01:59 No.71469

Pretty damn confused about what it...IS.  
But I completed it anyway.

>>  **Anonymous** 05/24/13(Fri)03:03 No.71485

[>>69983 \(OP\)](#)  
><http://participate.usabilla.com/ec9bd83754105ab4a7dcca9536935d7fb38856f>

Done. There were some glaring problems that I thoroughly noted.

Figure 81 – Post on the graphic design 4chan board to get testers for a usability test

**Summary of Changes**

Based on the usability test results and tester feedback, the following changes were incorporated into the implemented version of the web application.

- Added JavaScript validation onblur and onclick so users immediately spot unanswered questions.
- Added JavaScript validation on page load or when the Next Section or Finish buttons are clicked to automatically validate and scroll the first unanswered question or incomplete section.
- Added a save confirmation when the Next Section button is clicked.
- Replaced the bookmark page button with a save to PDF button.
- Added an Email Results button below the list of suggested methodologies.
- Added an asterisk or the word required beside required form fields.

**Where will you tap to name the project?**

80% of the testers understood where to click to specify a project name. For the other 20% of the testers, if they do not specify a project name, the project name text field will be scrolled to and highlighted.

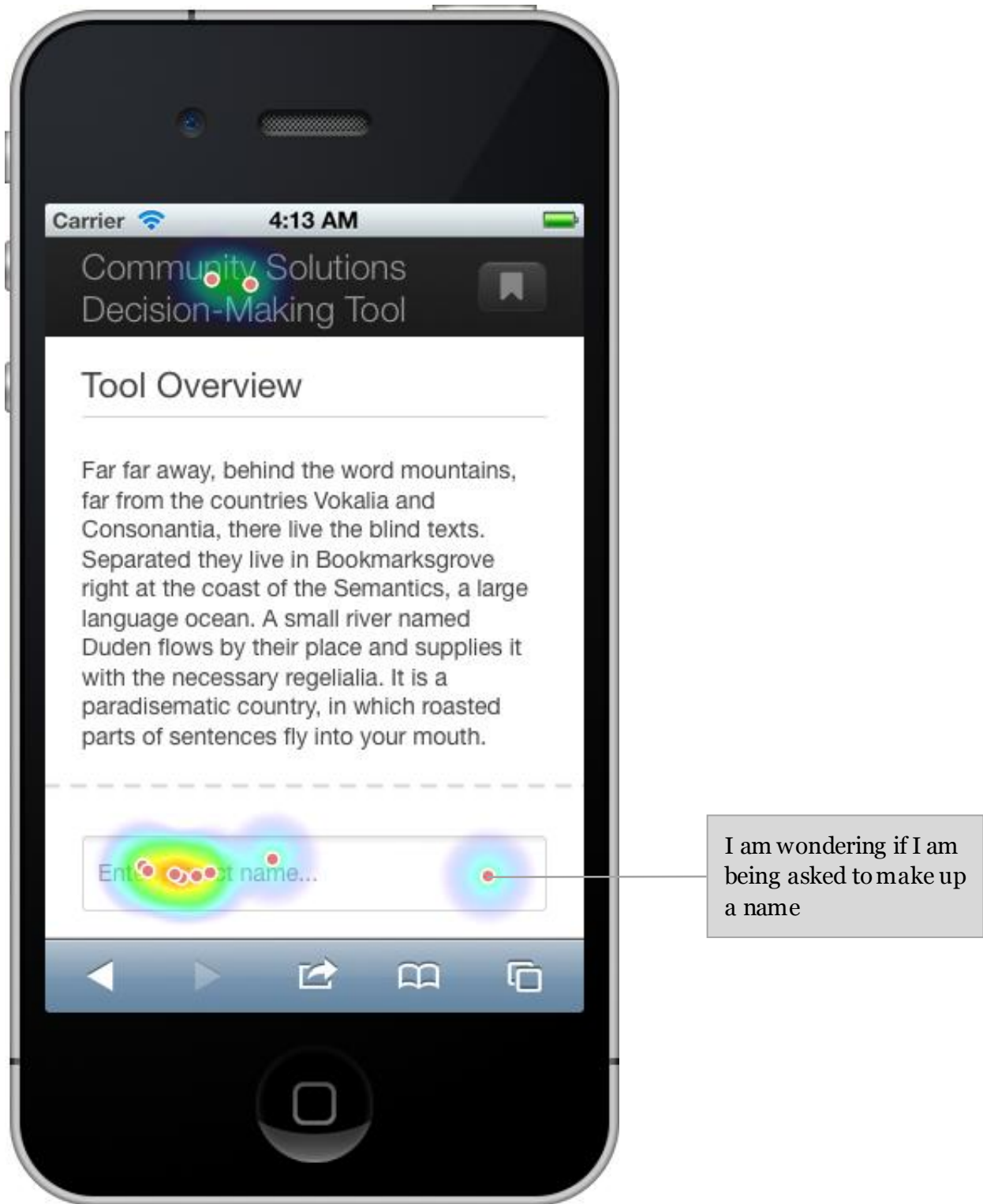


Figure 82 – Heat map showing where testers tapped when asked to specify a project name

**Where will you tap to get more information about a question?**

90% of the testers understood where to click to get more information about a question.



**Figure 83 – Heat map showing where testers tapped when asked to get more information about a question**

### Where will you tap to answer a question?

It was clear to all the testers where they needed to click to answer a question.

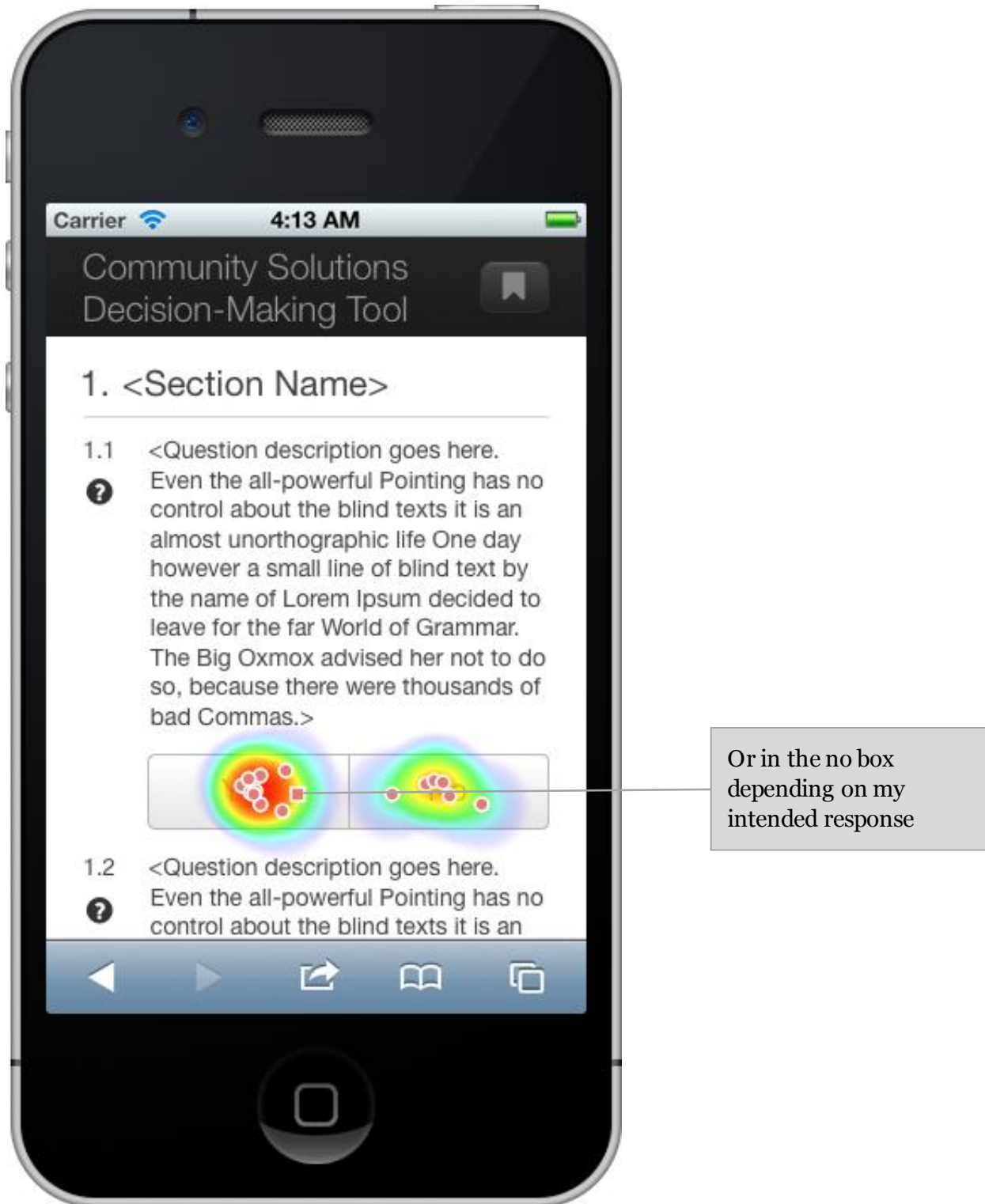


Figure 84 – Heat map showing where testers tapped when asked to answer a question

**Where will you tap to save the questionnaire to finish later?**

40% of the testers figured out that the page can be bookmarked to return to the questionnaire later. 40% of the testers did not know they can save the questionnaire and think they have to complete all of it in one session, which is okay too. The remaining 20% of the testers do not know where to click. The bookmark button has been replaced with a print to PDF button since the bookmark function does not work in some browsers and the button is redundant.

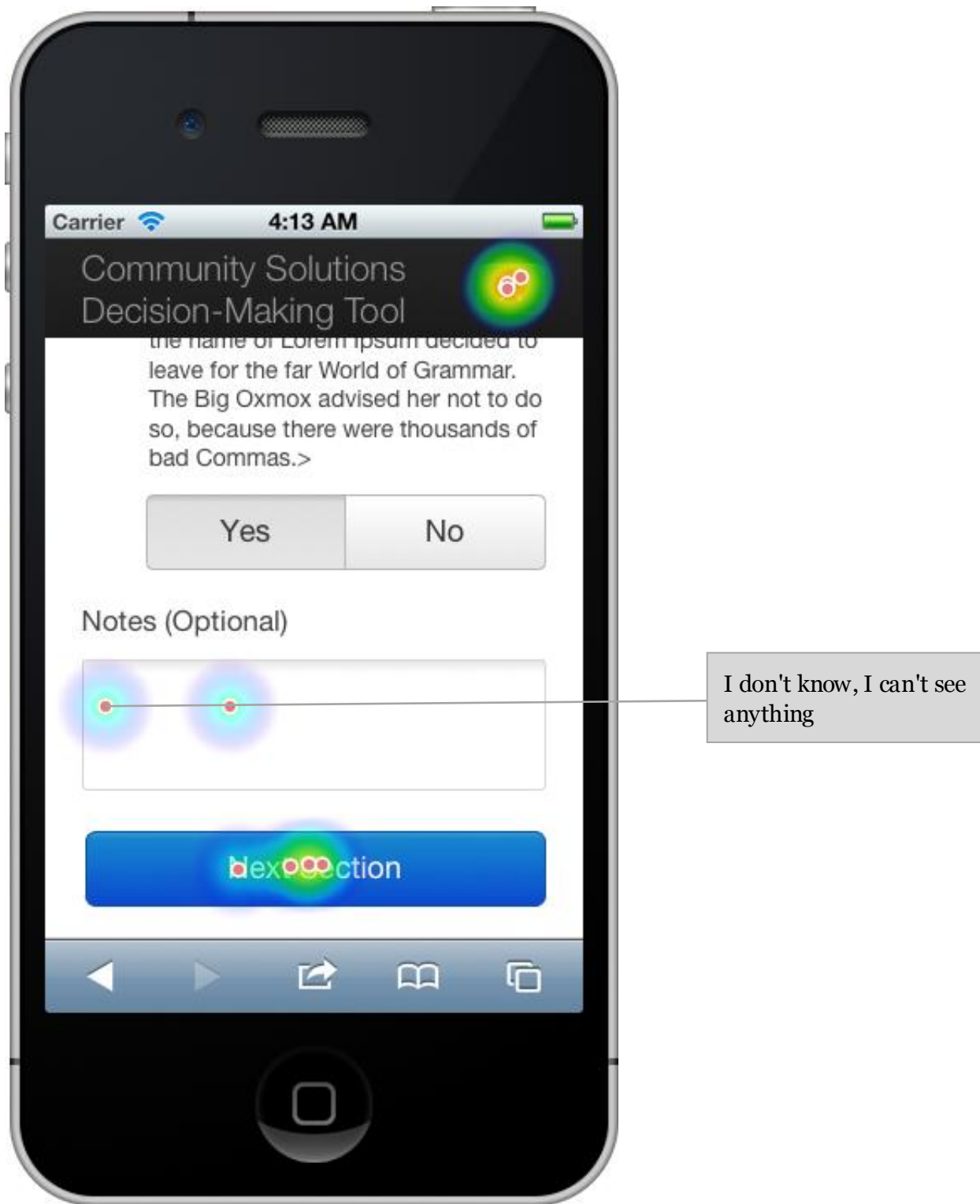


Figure 85 – Heat map showing where testers tapped when asked to save the questionnaire

**What do you expect will happen when the “Next Section” button is tapped?**

All of the testers figured out that clicking on the “Next Section” button will bring up the next set of questions. Only two testers had the expectation that the questionnaire responses will be saved

in case of a session loss. This is improved by making a success message appear when the “Next Section” button is clicked to inform the user that the questionnaire is successfully saved.

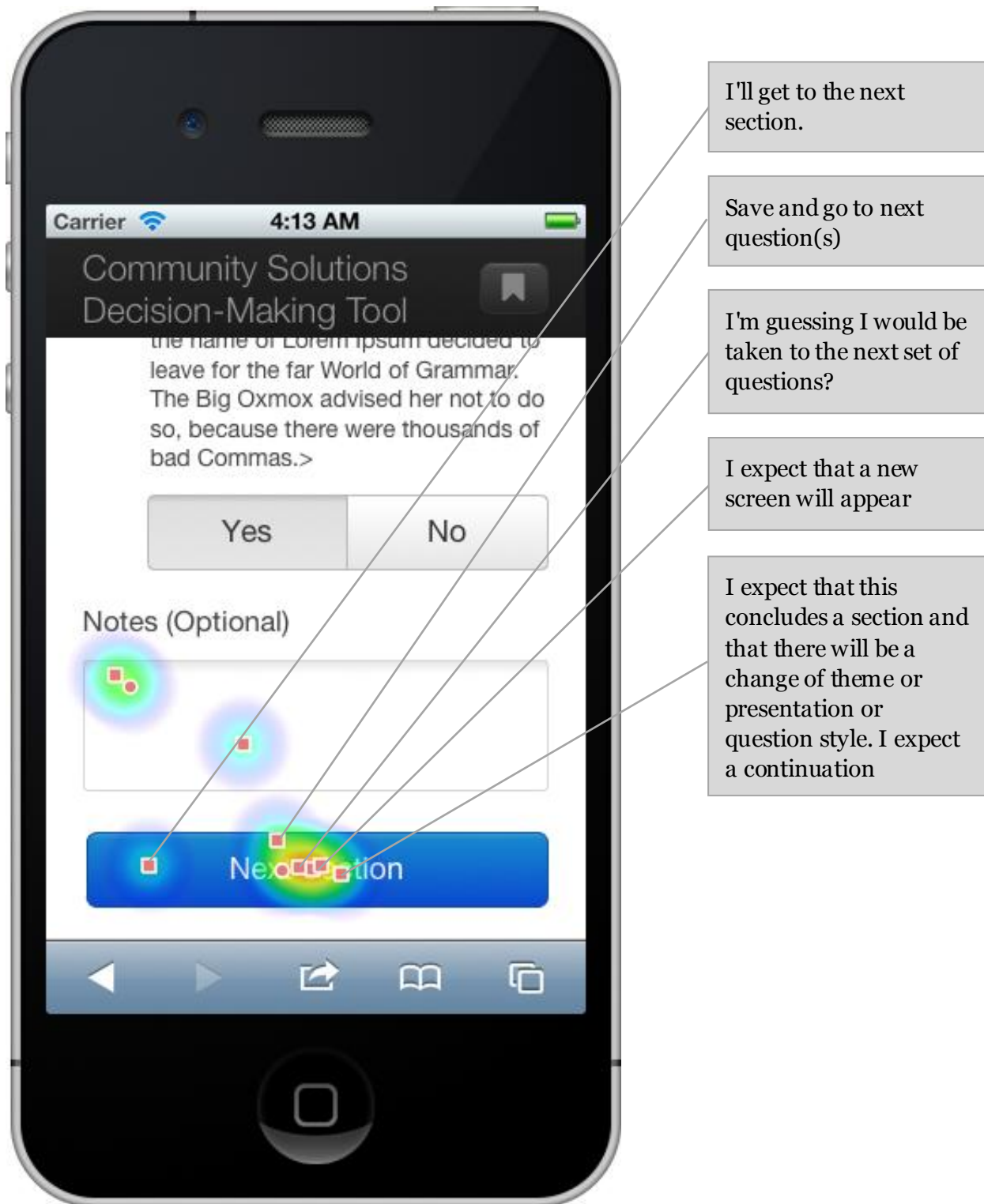


Figure 86 – Heat map showing where testers tapped when asked what the “Next Section” button does when tapped

**Where will you tap to correct this error?**

90% of the testers understood where they need to click to correct the error.



Figure 87 – Heat map showing where testers tapped when asked how to correct the submission error

**You want to save your results. Where will you tap?**

70% of the testers figured out that the page can be bookmarked to return to the questionnaire results later. 30% of the testers were confused about where to tap since the save option is not shown in this picture (The bookmark page and email results options are accessed from the collapsible menu). An Email Results button is added below the list of suggested methodologies so users can email their results in case they did not notice the actions button in the header.

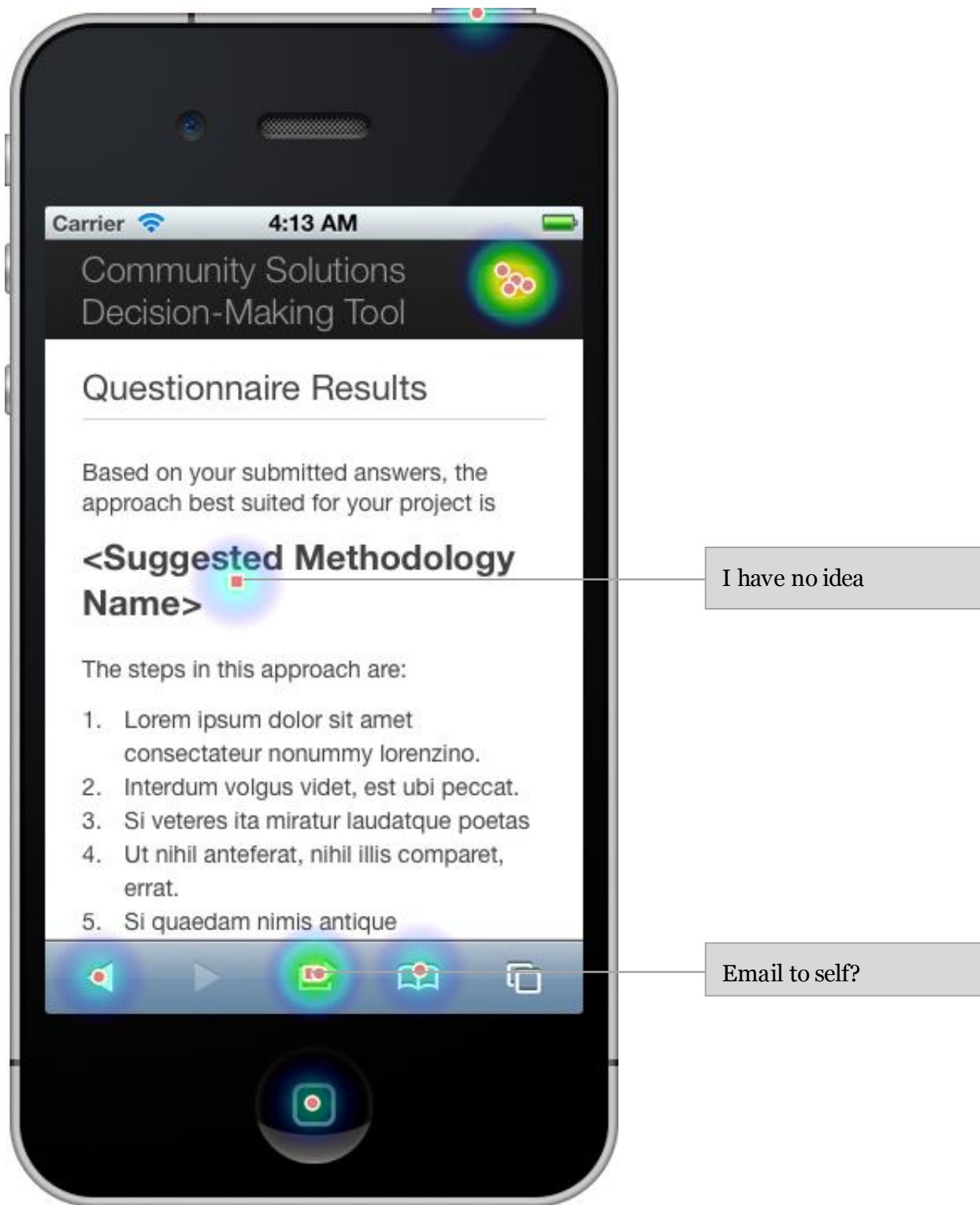
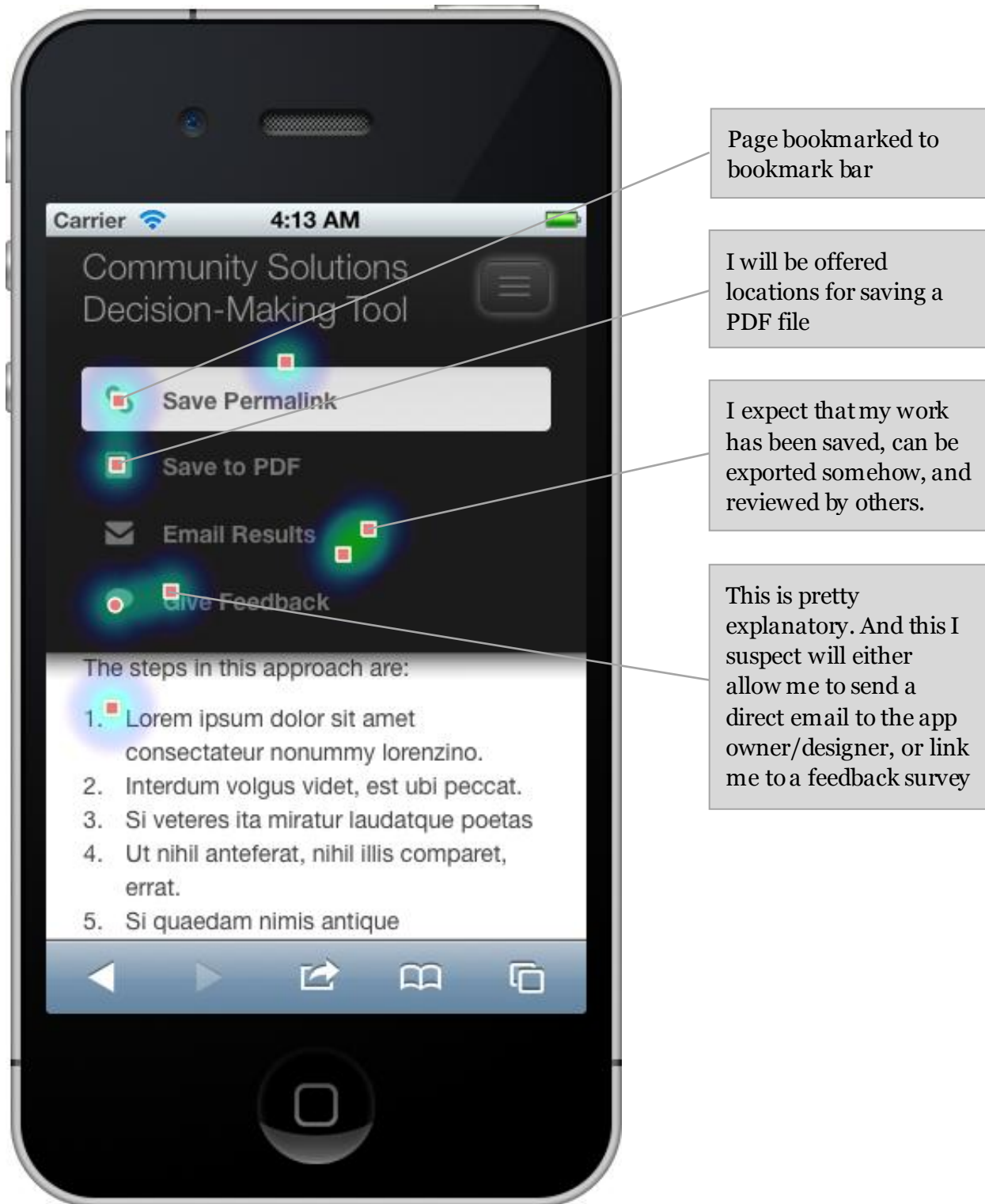


Figure 88 – Heat map showing where testers tapped when asked to save their questionnaire results

**What do you expect will happen when one of the menu options is selected?**

Most of the testers have a general idea of what each menu item does.



**Figure 89 – Heat map showing where testers tapped when asked what each menu option does**

**Which form fields do you think are required?**

Only 40% of testers realize that all the form fields are required. The form labels are modified to add an asterisk or the word required beside the required fields.

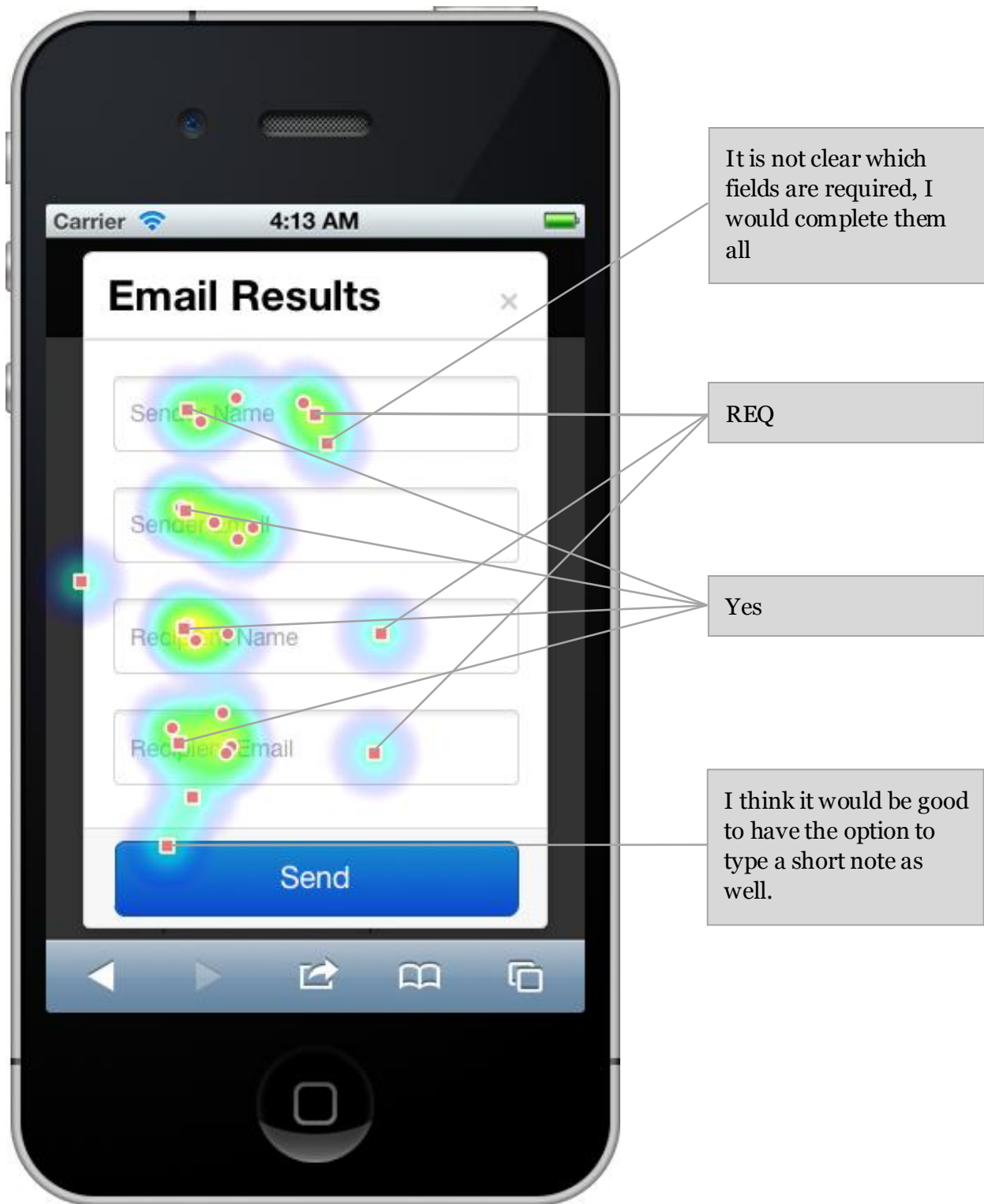
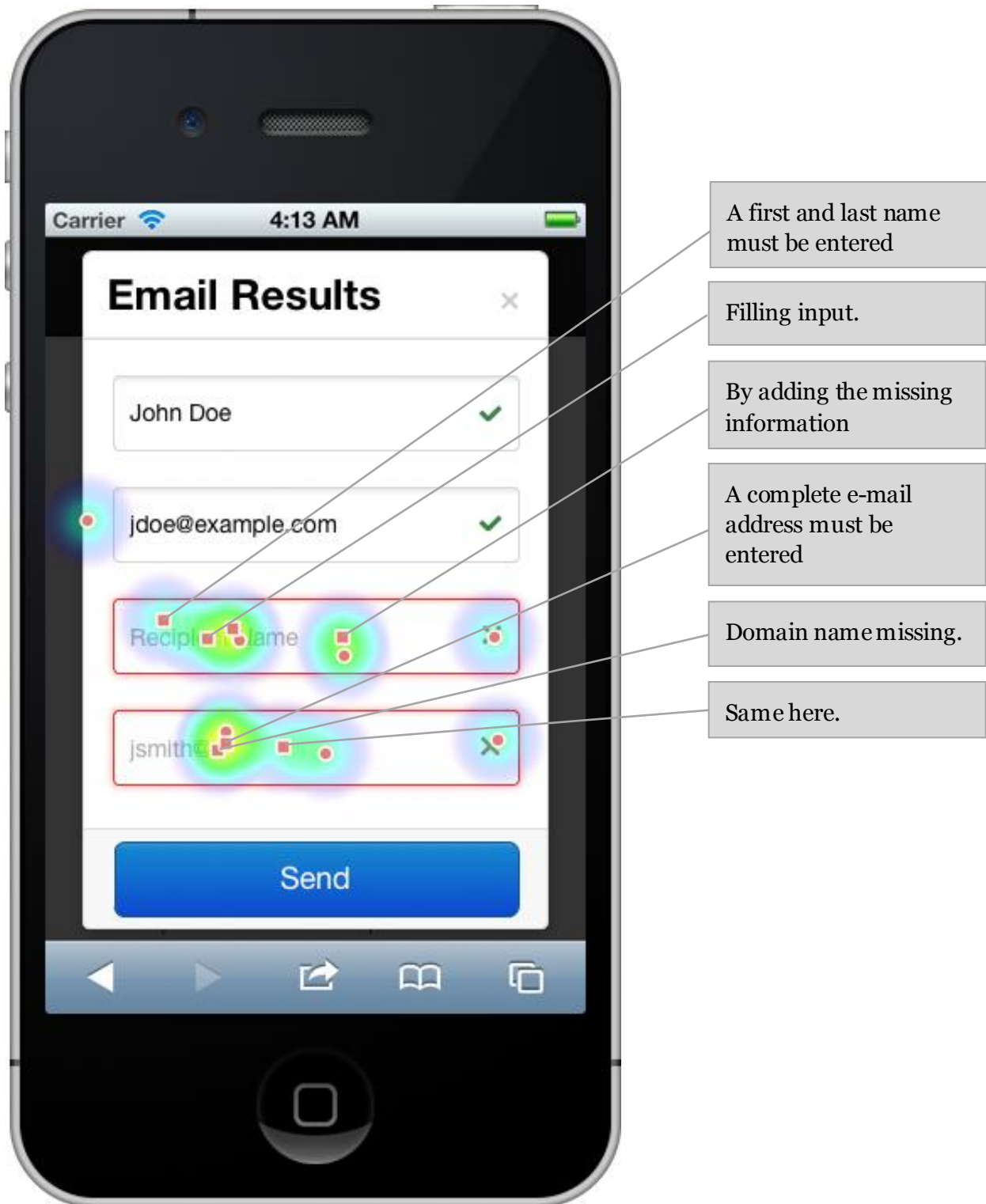


Figure 90 – Heat map showing where testers tapped when asked which form fields are required

**How can the input be corrected so the form will submit?**

All the testers figured out that the recipient name and email fields are required.



**Figure 91 – Heat map showing where testers tapped when asked how to correct the form submission error**

## Appendix E – Site Map

### Front End

index.php This is the page where one can complete the latest questionnaire.

results.php This is the page where one can view the questionnaire results.

### Back End

admin.php This is the questionnaire backend where one can modify the questionnaire overview, add, remove, or reorder questionnaire sections, add questionnaire categories, and publish questionnaire changes.

edit-section.php This is the questionnaire backend where one can create/modify the questionnaire sections to rename the section, add, remove, or reorder questions in the section, assign categories and weights to questions, and modify the question text and question information text.

edit-category.php This is the questionnaire backend where one can create/modify the questionnaire categories to rename the category, change the category colour, and change the category methodology text.

### Helper Files

functions.php This file contains all the functions required for the decision tool to work. This includes the functions needed to create/modify/reorder/delete questionnaire sections, create/modify questionnaire categories, gather the latest/latest published questionnaire data for display, save/load the questionnaire responses, and calculate which methodologies to show for a set of responses.

questionnaire-submit.php This file handles saving the questionnaire responses to the database using AJAX.

email\_results.php This file validates the email results form data, using AJAX, and sends the questionnaire results via the Mandrill API.

feedback-submit.php This file validates the give feedback form data, using AJAX, and writes the contents to feedback.txt.

print\_questionnaire.php This file handles the PDF generation for the latest published questionnaire version or an older one if a questionnaire ID is provided.

print_results.php	This file handles the PDF generation for the questionnaire results if a questionnaire ID for a completed questionnaire is provided.
auth.php	This file is included in all backend pages to handle user authentication and prevents brute force attacks by disallowing further authentication attempts after five tries.
unauth.php	This file is included in all backend pages to handle user deauthentication. The user is automatically redirected to index.php when logged out using AJAX.
meekrodb.2.2.class.php	This file is a PHP MySQL library used to handle SQL queries and prevent SQL injection attacks.

## Appendix F – Test Cases & Results of Testing

### Admin Section

Testing has been performed on Chrome 28, Safari 24, Safari 4, IE 8, IE 10, IE 11 and Firefox 19. All test cases passed on the above browser versions.

Test Case A1. The admin should be able to log into the backend.

Steps to complete	Expected Outcome
1. Visit the questionnaire backend. (http://192.168.1.73/decision-tool/admin.php)	<ul style="list-style-type: none"> <li>■ An authentication dialog should appear.</li> </ul>
2. Enter the user name and password into the authentication dialog.	<ul style="list-style-type: none"> <li>■ If the user credentials are valid, the user will be redirected to the admin landing page.</li> </ul>

Test Case A2. The admin should be able to rename the questionnaire and change the tool overview text.

Steps to complete	Expected Outcome
1. Enter the questionnaire title and the tool overview text. 2. Click the Save Overview Changes button.	<ul style="list-style-type: none"> <li>■ The admin landing page will refresh.</li> <li>■ The new questionnaire title will show in the page header.</li> <li>■ An unpublished changes message with a Publish Changes button will appear at the top of the admin landing page.</li> </ul>

Test Case A3. The admin should be able to create categories from the admin landing page.

Steps to complete	Expected Outcome
1. Click the Add Category button.	<ul style="list-style-type: none"> <li>■ The Add Category page will appear with a blank form to create a new category.</li> </ul>
2. Enter the category details and click the Save Changes button.	<ul style="list-style-type: none"> <li>■ The category data will be saved and the user will be redirected to the Edit Category page with the new category ID.</li> </ul>
3. Click on the questionnaire title in the page header.	<ul style="list-style-type: none"> <li>■ The user will return to the admin landing page.</li> <li>■ The name of the new category will appear under the Questionnaire Categories section.</li> <li>■ The category name will be a link to allow the user to edit that category.</li> </ul>

Test Case A4. The admin should be able to edit categories from the admin landing page.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Click on one of the category names under the “Questionnaire Categories” section.	<ul style="list-style-type: none"> <li>■ The user will be taken to the Edit Category page where the category’s saved details will be shown.</li> </ul>
2. Do any of the following: <ol style="list-style-type: none"> <li>i. Edit the category name.</li> <li>ii. Select a new category colour.</li> <li>iii. Edit the category’s method text.</li> </ol> Click the Save Changes button when finished.	<ul style="list-style-type: none"> <li>■ The category data will be saved and the user will be redirected to the Edit Category page.</li> <li>■ If the questionnaire is published, the Edit Category page will have a new category ID or else the category ID will remain the same.</li> </ul>
3. Click on the questionnaire title in the page header.	<ul style="list-style-type: none"> <li>■ The user will return to the admin landing page.</li> <li>■ The edited category name will appear under the Questionnaire Categories section.</li> <li>■ The category name will be a link to allow the user to edit that category.</li> <li>■ An unpublished changes message with a Publish Changes button will appear at the top of the admin landing page.</li> </ul>

Test Case A5. The admin should be able to create sections from the admin landing page.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Click the Add Section button.	<ul style="list-style-type: none"> <li>■ The Add Section page will appear with a blank form to create a new section.</li> </ul>
2. Enter the section details and add as many questions as needed by clicking on the Add Question button.	<ul style="list-style-type: none"> <li>■ A new blank question form will appear below the first question.</li> </ul>
3. Rearrange questions by clicking and dragging the grey box surrounding each question.	<ul style="list-style-type: none"> <li>■ The question form will detach from the page and become draggable so it can be rearranged.</li> </ul>
4. Click the X button at the top right corner of a question.	<ul style="list-style-type: none"> <li>■ The question will be removed.</li> </ul>
5. Click the Save Changes button.	<ul style="list-style-type: none"> <li>■ The section data will be saved and the user will be redirected to the Edit Section page with the new section ID.</li> </ul>
6. Click on the questionnaire title in the header.	<ul style="list-style-type: none"> <li>■ The user will return to the admin landing page.</li> <li>■ The newly created section name will appear under the Questionnaire Sections section.</li> </ul>

	<ul style="list-style-type: none"> <li>■ The section name will be a link to allow the user to edit that section.</li> <li>■ An unpublished changes message with a Publish Changes button will appear at the top of the admin landing page.</li> </ul>
--	---

Test Case A6. The admin should be able to edit sections from the admin landing page.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Click on one of the section names under the Questionnaire Sections section.	<ul style="list-style-type: none"> <li>■ The Edit Section page will appear where the section’s saved details will be shown.</li> </ul>
2. Do any of the following: <ol style="list-style-type: none"> <li>i. Edit the section title.</li> <li>ii. Select a category for a question.</li> <li>iii. Edit the question’s weight.</li> <li>iv. Edit the question description.</li> <li>v. Edit the question’s help text.</li> <li>vi. Delete a question.</li> <li>vii. Reorder questions.</li> </ol> Click the Save Changes button when finished.	<ul style="list-style-type: none"> <li>■ The section data will be saved and the user will be redirected to the Edit Section page.</li> <li>■ If the questionnaire is published, the Edit Section page will have a new section ID or else the section ID will remain the same.</li> </ul>
3. Click on the questionnaire title in the header.	<ul style="list-style-type: none"> <li>■ The user will return to the admin landing page.</li> <li>■ The edited section name will appear under the Questionnaire Sections section.</li> <li>■ The section name will be a link to allow the user to edit that section.</li> <li>■ An unpublished changes message with a Publish Changes button will appear at the top of the admin landing page.</li> </ul>

Test Case A7. The admin should be able to reorder sections from the admin landing page.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Rearrange sections by clicking and dragging the grey box surrounding each section name under the Questionnaire Sections section.	<ul style="list-style-type: none"> <li>■ The section name will detach from the page and become draggable so they can be rearranged.</li> </ul>
2. Click the Save Section Changes button.	<ul style="list-style-type: none"> <li>■ The user will return to the admin landing page.</li> <li>■ The new section order will be reflected under the Questionnaire Sections section.</li> <li>■ An unpublished changes message with a Publish Changes button will appear at the top of the admin landing page.</li> </ul>

Test Case A8. The admin should be able to delete sections from the admin landing page.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Click on the X button to the right of each section name under the Questionnaire Sections section.	<ul style="list-style-type: none"> <li>■ The deleted section will be removed from the Questionnaire Sections section.</li> </ul>
2. Click the Save Section Changes button.	<ul style="list-style-type: none"> <li>■ The user will return to the admin landing page.</li> <li>■ The deleted section name will no longer appear under the Questionnaire Sections section.</li> <li>■ An unpublished changes message with a Publish Changes button will appear at the top of the admin landing page.</li> </ul>

Test Case A9. The admin should be able to publish questionnaire changes from the admin landing page.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. If there are unpublished changes, there will be an unpublished changes message with a Publish Changes button at the top of the admin landing page. Click the Publish Changes button.	<ul style="list-style-type: none"> <li>■ A confirmation dialog to publish changes will appear.</li> </ul>
2. Click the OK button in the confirmation dialog.	<ul style="list-style-type: none"> <li>■ The admin landing page will refresh.</li> <li>■ The unpublished changes message will have disappeared.</li> <li>■ The questionnaire changes will now be reflected on the questionnaire page.</li> </ul>

Test Case A10. The admin should be able to log out of the backend.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Click the Log Out button in the page header.	<ul style="list-style-type: none"> <li>■ The user's session will be invalidated.</li> <li>■ The user will be redirected to the questionnaire page.</li> </ul>

## User Section

Testing has been performed on Chrome 28, Safari 24, Safari 6, IE 8, IE 10, IE 11 and Firefox 29. Due to inability to access tablets and smart phones for testing, Firefox's Responsive Design View was used to simulate mobile and tablet experiences.

All test cases passed on the above browser versions.

Test Case U1. The user should be able to view and complete the latest published questionnaire.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Visit the questionnaire page. ( <a href="http://192.168.1.73/decision-tool/index.php">http://192.168.1.73/decision-tool/index.php</a> )	<ul style="list-style-type: none"> <li>■ The user will be redirected to index.php with a newly generated questionnaire ID.</li> </ul>
2. Enter the project name.	<ul style="list-style-type: none"> <li>■ If the project name text field is left blank, it will be highlighted.</li> <li>■ The error highlighting will be removed once it is filled in.</li> </ul>
3. Answer each question in the section by clicking on the “Yes” or “No” radio buttons beside each question.	<ul style="list-style-type: none"> <li>■ The clicked radio button will be selected to reflect the choice.</li> <li>■ The previous questions will be checked to see if they have been answered.</li> <li>■ If a previous question has not been answered, the unanswered question will be highlighted.</li> <li>■ The error highlighting will disappear once the question is answered.</li> </ul>
4. Click on the Next Section or Finish button.	<ul style="list-style-type: none"> <li>■ The project name text field will be checked to see if it has been filled in and each question in the section will be checked to see if they have been answered.</li> <li>■ If the project name text field is blank or a question has been left unanswered, they will be highlighted and scrolled to.</li> <li>■ If there are any errors, the form submission will be cancelled. If not, the next section will be unhidden and the page will scroll to the next section.</li> <li>■ If the current section is the last section, the user will be redirected to the questionnaire results page.</li> </ul>

Test Case U2. The user should be able to revisit a questionnaire.

<b>Steps to complete</b>	<b>Expected Outcome</b>
1. Visit the questionnaire page with the unique 13 character questionnaire id. ( <a href="http://192.168.1.73/decision-tool/index.php?id=51d50975d5b3e">http://192.168.1.73/decision-tool/index.php?id=51d50975d5b3e</a> )	<ul style="list-style-type: none"> <li>■ If the questionnaire ID is valid, the saved questionnaire answer data and its associated questionnaire data will be loaded or else the user will be redirected to index.php with a newly generated questionnaire ID.</li> <li>■ If the questionnaire is unfinished, the page will scroll to the first unanswered question or section or else all sections will be</li> </ul>

	unhidden.
2. Continue filling out the questionnaire.	<ul style="list-style-type: none"> <li>■ The questionnaire will be validated and submitted as usual.</li> </ul>

Test Case U3. The user should be able to view the questionnaire results.

Steps to complete	Expected Outcome
1. Click the Finish button at the end of the questionnaire or visit the questionnaire results page with the unique 13 character questionnaire ID.	<ul style="list-style-type: none"> <li>■ If the questionnaire ID is valid and the questionnaire is fully complete, the methodology descriptions of the highest scoring categories will be shown or else the user will be redirected to index.php.</li> </ul>

Test Case U4. The user should be able to print the questionnaire questions and results to a PDF.

Steps to complete	Expected Outcome
1. Visit the questionnaire page and click the Print to PDF button in the header or in the sidebar. ( <a href="http://192.168.1.73/decision-tool/index.php">http://192.168.1.73/decision-tool/index.php</a> )	<ul style="list-style-type: none"> <li>■ A PDF file will be generated containing all the questionnaire questions and methodologies along with a scoring guide.</li> </ul>
2. Visit the questionnaire results page with the unique 13 character questionnaire ID and click the Print to PDF button in the collapsible header menu or in the sidebar.	<ul style="list-style-type: none"> <li>■ A PDF file will be generated containing the recommended methodologies along with all the questionnaire questions and responses.</li> </ul>

Test Case U5. The user should be able to email a summary of the questionnaire.

Steps to complete	Expected Outcome
1. Visit the questionnaire results page with the unique 13 character questionnaire ID and click the Email Results button in the collapsible header menu or in the sidebar.	<ul style="list-style-type: none"> <li>■ An email form will appear as a modal or as a form in the sidebar.</li> </ul>
2. Fill in all of the required form fields and click the Send button.	<ul style="list-style-type: none"> <li>■ If any of the required form fields are left blank or the email address field contains an email in the incorrect format, they will be highlighted and the form submission will be cancelled.</li> <li>■ If there are no validation errors, the form will submit and either the modal will close or a success message will appear above the sidebar form.</li> </ul>

## Appendix F – Writing and Reading from the Database

The following snippet of code shows how an administrator can see and change the questionnaire overview text.

The backend includes a file called functions.php. This file contains all the functions that are used by the decision-making tool. The two functions that will be used in this example are `get_display_app_info_for_admin()` and `app_info_update($new_app_name, $new_overview_text)`.

The function `get_display_app_info_for_admin` gets the most recent app info for display purposes while `app_info_update` updates the basic information of an app or creates new app depending on its publish status.

```
<?php
// contents of functions.php
function get_display_app_info_for_admin() {
    $app_row = get_latest_app_row();
    $section_order = json_decode($app_row['section_order'],1);
    $display_arr = array();
    $display_arr['app_id'] = $app_row['app_id'];
    $display_arr['app_name'] = $app_row['app_name'];
    $display_arr['overview_text'] = $app_row['overview_text'];
    $display_arr['sections'] = array();
    $display_arr['is_published'] = $app_row['is_published'];

    $count = count($section_order);
    for ($j=0; $j<$count; $j++) {
        $section = DB::queryFirstRow("select section_title from
sections where section_id = %i", $section_order[$j]);
        array_push($display_arr['sections'], array(
            'section_id' => $section_order[$j],
            'section_title' => $section['section_title']
        ));
    }
    return $display_arr;
}

function app_info_update($new_app_name, $new_overview_text) {
    $app_row = get_latest_app_row();
    if ($app_row['is_published']) {
        $old_section_order = json_decode($app_row['section_order'], 1);

        $new_section_order = array();
        $count = count($old_section_order);
        for ($i = 0; $i < $count; $i++) {
            array_push($new_section_order,
                clone_section($old_section_order[$i]));
        }
        $new_section_order_json = json_encode($new_section_order);
        DB::insert('apps', array(
            'app_id' => 0,
            'app_name' => $new_app_name,
            'overview_text' => $new_overview_text,
            'section_order' => $new_section_order_json,
            'is_published' => 0
        ));
    }
    else { // not published
        DB::update('apps', array(
            'app_name' => $new_app_name,
            'overview_text' => $new_overview_text
        ), "app_id=%i", $app_row['app_id']);
    }
}
?>
```

Figure 92 – A code snippet from functions.php

The backend has to first connect to the database with the supplied username and password using the meekrodb MySQL PHP library. Then the latest questionnaire data is pulled from the database using the `get_display_app_info_for_admin()` function from `functions.php`. The check to see if `save-overview-changes` is in the `$_POST` variable is used to see if the questionnaire overview form has been submitted. If it is, the `app_info_update` method from `functions.php` will be used to save to format and save the changes into the database.

```
<?php
// contents of admin.php
require_once 'meekrodb.2.2.class.php';
DB::$user = 'decision_test';
DB::$password = 'testing';
DB::$dbName = 'decision_tool';

include_once('functions.php');

if (! empty($_POST)) {
    // post save overview changes
    if (isset($_POST['save-overview-changes'])) {
        if (isset($_POST['appTitle']) && ($_POST['appTitle'] != '') &&
isset($_POST['appOverview'])) {
            app_info_update($_POST['appTitle'],
$_POST['appOverview']);
        }
    }
}

// now read DB data for display
$app_arr = get_display_app_info_for_admin();
?>
```

**Figure 93 – A code snippet from `admin.php`**

After the backend retrieves the latest questionnaire data for display, it needs to display it in a form. The questionnaire name, `$app_arr['app_name']`, is echoed into the title text field and the questionnaire overview text, `$app_arr['overview_text']`, is echoed into the tool overview text area. When the administrator clicks the form submit button, the name of the submit button, `save-overview-changes` is included in the `$_POST` array, which will then tell the web server to write the questionnaire overview changes to the database.

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- HTML portion of admin.php -->
</head>
<body>
  <form id="form-overview" action="#" method="post" class="form">
    <p class="required-txt">Fields marked <span class="form-
    required"><strong>*</strong></span> are required.</p>

    <div class="control-group">
      <label class="control-label notes-box"
      for="appTitle">Title <span class="form-
      required"><strong>*</strong></span></label>
      <div class="controls">
        <input type="text" name="appTitle" id="appTitle"
        class="span12" placeholder="" value="<?php echo
        $app_arr['app_name']; ?>" >
      </div>
    </div>

    <div class="control-group">
      <label class="control-label notes-box"
      for="appOverview">Tool Overview</label>
      <div class="controls">
        <textarea class="span12" name="appOverview"
        id="appOverview" rows="6" cols="100"
        placeholder="Enter the tool overview here"
        style="width: 100%; resize: vertical;"><?php echo
        $app_arr['overview_text']; ?></textarea>
      </div>
    </div>

    <div class="control-group">
      <div class="controls">
        <button type="submit" class="btn btn-primary"
        name="save-overview-changes" id="save-overview-
        changes">Save Overview Changes</button>
      </div>
    </div>
  </form>
</body>
</html>
```

Figure 94 – A code snippet from admin.php