Functional Specification

Version: 1.0

ABC Organization
Electronic Survey Application

Copyright and Trademarks

Summary of Changes

Version	Description	Created by	Date
1.0	Created final version	Christopher Muggridge	November 19, 2018

Document Conventions

The following typefaces are used throughout this guide:

- The '`Courier New`' typeface is used for directory objects and attributes, file names and command line code.
- The 'Times New Roman' typeface is used for cause and effect descriptions.
- '*Italics*' are used for emphasis and for cross references.
- This **bold** typeface is used to represent information you should type in at the keyboard.

Table of Contents

1 Purpose

This document purpose is to describe the functionality of the Electronic Survey Application.

1.1 Audience

This document is intended for <developer> managers, solution architects and developers.

1.2 Assumptions

This document assumes that:

- *The surveys and all associated management and reporting features will be hosted by <developer> on a secure server housed in their state-of-the-art datacenter.*
- *The application database will be hosted by <developer> on a secure server housed in their state-of-the-art datacenter and will be accessed directly by all versions of the application including browser and mobile versions.*
- *Host requirements for the application will include the following:*
    - *PHP*
    - *MySQL*
- *The core application will be developed in PHP with a MySQL database and will be optimized for the following browsers:*
    - *Firefox (v. 17+)*
    - *Chrome (v. 25+)*
    - *Internet Explorer (v. 9+)*
    - *Safari*
- *The application will be developed using a CSS framework which will allow the user interface to dynamically adjust to suit the device being used to access it.*
- *Mobile access to surveys will be made available through the use of an installable app which will utilize the same browser based version specifically optimized for each device. The app will be responsible only for launching the mobile device's browser and pointing to the website automatically.*
- *Mobile launch apps will be required for the following devices:*
    - *Android Tablet*
    - *Android Phone*
    - *iPad*
    - *iPhone*
- *Audio files used in the application will be in MP3 format.*
- *Audio files will be played using technology compatible with both desktop and mobile browsers*

2 Module Description

The high-level architecture for the system is shown in Figure 2-0.

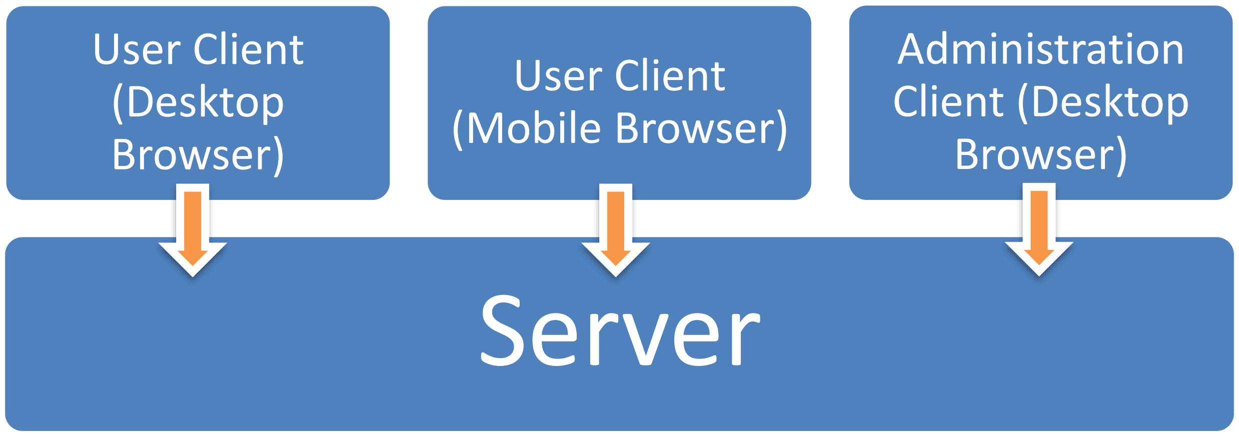
User Client (Desktop Browser)
User Client (Mobile Browser)
Administration Client (Desktop Browser)
Server

Figure 2-0 High-Level Architecture

2.1 Server Module

Purpose

The purpose of this module is to provide a centralized place where information for the system can be stored, manipulated, and accessed.

Rational

This module is created to centralize and encapsulate all data storage and retrieval duties on the system. This includes user profiles, survey questions, and survey results. It also provides some services, such as authentication, network communication and search.

High Level Server Design

The server module is broken down into lower-level modules, as shown in Figure 2-1.

As is obvious, all communications from the client come through the communications module. The provided interface of the high-level server component and the server communications module are thus identical. Two modules in the server handle administrator and user functions, respectively. Each of these modules will be described in detail in following sections.

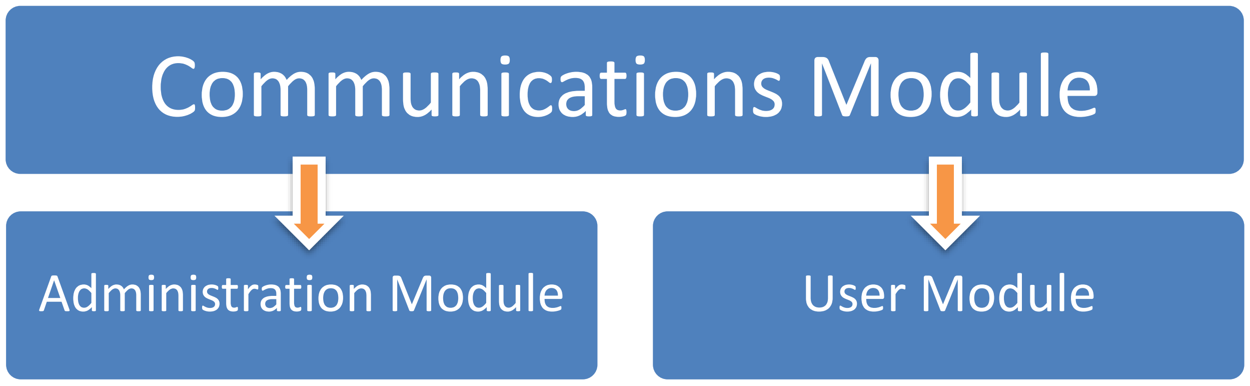

Figure 2-1 High-Level Server Architecture

Required Interface
This module has no required interface.

Provided Interface
The provided interface of this module is the union of the provided interfaces of the following sub-modules:

- Communications

For more detail, please see these modules' descriptions.

2.2 Communications Module

Purpose
The purpose of this module is to provide communication services between the clients of the system (both administrative and user clients) and the server. This module represents the part of the communications link that resides on the server.

Rational
This module is created to centralize and encapsulate network communication duties between clients and the server.

Required Interface
This module's required interface is the union of the provided interfaces of the following components:

- Administrative Module
- User Module

Provided Interface

The provided interface of this module is the same as its required interface, except that the interface is made available for network calls. As such, it cannot be represented by procedure calls, as this would unnecessarily constrain the underlying representation to be Remote Procedure Call (RPC) oriented.

2.3 Administration Module

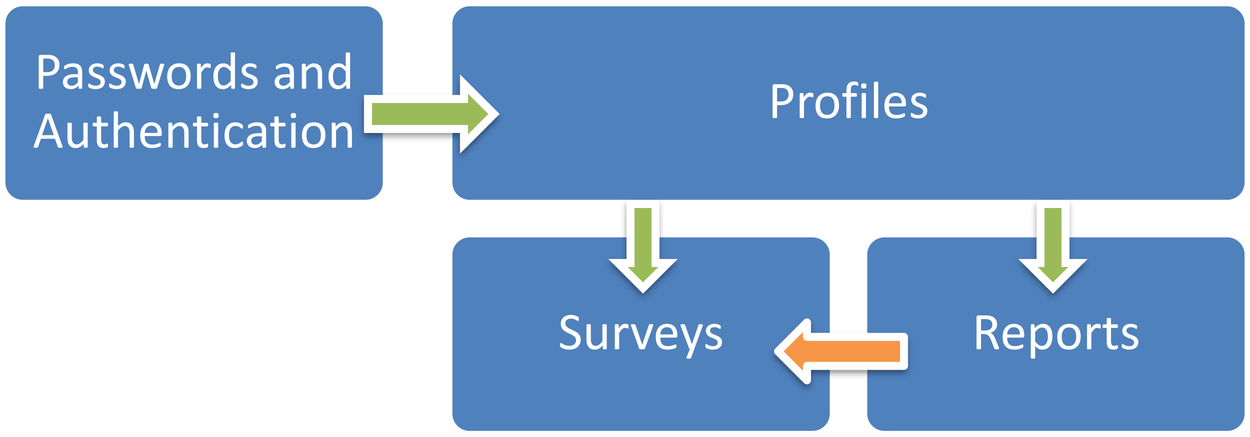

Figure 2-3 Architecture of the Administration Module

Purpose

The purpose of this module is to authenticate administrators, provide data reports, and manage surveys to be interacted with via the User Interface.

Rational

This module is created to centralize services related to administrators.

High-Level Module Design

The administration module is broken down into high-level modules, as shown in Figure 2-3.

Required Interface

This module has no required interface.

Provided Interface

The provided interface of this module is the union of the provided interfaces of the following sub-modules:

- Passwords and Authentication
- Profiles
- Surveys
- Reports

For more detail, please see these modules' descriptions.

2.3.1 Profiles Module

Purpose

The purpose of this module is to provide access to and a persistent data store for all administrator profiles in the system.

Rationale

This module is created to centralize and encapsulate data storage and retrieval duties and related to administrators.

Required Interface

This module has no required interface.

Provided Interface

```
putAdministratorProfile(AdministratorProfile);
```

Description:
Stores an administrator profile with a unique email address.

Parameters:
AdministratorProfile: Administrator profile to store in data store. If profile's email address is already stored, this profile will not be added.

```
AdministratorProfile
getAdministratorProfile(AdministratorID) throws
NoSuchAdministratorException;
```

Description:
Gets an administrator profile given the administrator's ID.

Parameters:
AdministratorID: The administrator ID of the profile to retrieve.

Returns:
The requested administrator profile.

Exceptions:
NoSuchAdministratorException: If the AdministratorID did not represent an existing profile.

```
AdministratorProfile[] getAllAdministratorProfiles();
```

Description:
Gets all administrator profiles.

Returns:
All administrator profiles.

```
AdministratorExists(AdministratorID);
```

Description:
Determines if an administrator profile exists in the data store.

Returns:
True if the administrator exists, false otherwise.

Parameters:
AdministratorID: The administrator ID of the profile to check.

```
UpdateAdministratorProfile(AdministratorProfile);
```

Description:
Updates an administrator profile.

Parameters:
AdministratorProfile: Administrator profile to store in data store. If profile's email address is already stored with another profile or required profile fields are left empty, this profile will not be added.

```
RemoveAdministratorProfile(AdministratorID) throws
NoSuchAdministratorException;
```

Description:
Removes an administrator profile from the system.

Parameters:
AdministratorID: The administrator ID of the profile to remove.

Exceptions:
NoSuchAdministratorException: If the AdministratorID did not represent an existing profile.

```
getNumProfiles();
```

Description:
Gets the number of administrator profiles stored on the system.

Returns:
An integer indicating the number of administrator profiles on the system.

2.3.2 Passwords and Authentication Module

Purpose

The purpose of this module is to provide an authentication service, allowing callers to determine whether a username/password combination is valid, and a change-password service, allowing users to change their passwords.

Rationale

This module is created to centralize and encapsulate password management and authentication services.

Required Interface

```
AdministratorExists(AdministratorID);
```

Provided Interface

```
SetPassword(AdministratorID, Password) throws
NoSuchAdministratorException;
```

Description:
Stores a password of the given administrator ID.

Parameters:
AdministratorID: Administration ID of the administrator whose password is being stored.
Password: The password for that administrator.

Exceptions:
NoSuchAdministratorException: If the AdministratorID did not represent an existing profile.

```
Logon(AdministratorID, Password) throws
NoSuchAdministratorException;
```

Description:

Determines whether a given username/password combination is valid or not. For security reasons, the incorrect portion of the combination (i.e. username or password) is not given.

Parameters:
AdministrationID: Administration ID of the administrator whose password is being checked.
Password: The presumed password for that administrator.

Returns:
True if the username/password combination is correct, false otherwise.

Exceptions:
NoSuchAdministratorException: If the username and/or password did not represent an existing profile.

```
Logoff(AdministratorID) throws
NoSuchAdministratorException,
AdministratorNotLoggedOnException;
```

Description:
Logs an administrator off the system.

Parameters:
AdministratorID: Administrator ID of the administrator who is logging off.

Returns:
True if the administrator exists, false otherwise.

Exceptions:
NoSuchAdministratorException: If the given administrator does not exist.
AdministratorNotLoggedOnException: If the given administrator is not logged on.

2.3.3 Surveys Module

Purpose

The purpose of this module is to provide a survey management service, allowing users to add, update, and archive surveys.

Rationale

This module is created to manage the surveys which will be completed by participants through the front end User Interface.

Required Interface

```
AdministratorExists(AdministratorID);
```

Provided Interface

```
PutSurvey(SurveyProfile, DynamicFields, AudioFile) throws
IncompleteDataException;
```

Description:
Stores a survey including identification, survey questions, survey answers, and MP3 audio files.

Parameters:
SurveyProfile: General information for survey including unique code, name, description, and notification email.
DynamicFields: Survey questions and answers created by the user. Number of questions and answers are not restricted to a minimum or maximum value.
AudioFile: Optional MP3 audio file of the corresponding question and answers. One audio file for each question, uploaded then assigned.

Exceptions:
IncompleteDataException: If required fields are left empty, this survey will not be added.

```
SurveyProfile getSurveyProfile(SurveyID) throws
NoSuchProfileException;
```

Description:
Gets a survey profile given the survey's ID.

Parameters:
SurveyID: The survey ID of the profile to retrieve.

Returns:
The requested survey profile.

Exceptions:
NoSuchSurveyException: If the SurveyID did not represent an existing profile.

```
SurveyProfile[] getAllSurveyProfiles();
```

Description:
Gets all survey profiles.

Returns:
All survey profiles.

```
SurveyExists(SurveyID);
```

Description:
Determines if a survey profile exists in the data store.

Returns:
True if the profile exists, false otherwise.

Parameters:
SurveyID: The survey ID of the profile to check.

```
UpdateSurvey(SurveyProfile, DynamicFields, AudioFile)
throws IncompleteDataException;
```

Description:
Updates survey.

Parameters:
SurveyProfile: General information for survey including name, description, and notification email.
DynamicFields: Survey questions and answers created by the user. Number of questions and answers are not restricted to a minimum or maximum value.
AudioFile: Optional MP3 audio file of the corresponding question and answers. One audio file for each question, uploaded then assigned.

Exceptions:
IncompleteDataException: If required fields are left empty, this survey will not be updated.

```
AudioUpload(AudioFile) throws FileUploadException;
IncorrectFileTypeException;
```

Description:
Uploads an MP3 audio file to server.

Parameters:
AudioFile: The audio file to be uploaded.

Exceptions:
FileUploadException: If the audio file was not able to be uploaded.
IncorrectFileTypeException: If the audio file is not of MP3 file type.

```
removeSurveyProfile(SurveyID) throws NoSuchSurveyException;
```

Description:
Removes a survey profile from the system.

Parameters:
SurveyID: The survey ID of the profile to remove.

Exceptions:
NoSuchSurveyException: If the SurveyID did not represent an existing profile.

```
getNumSurveys();
```

Description:
Gets the number of surveys stored on the system.

Returns:
An integer indicating the number of surveys on the system.

2.3.4 Reports Module

Purpose

The purpose of this module is to provide a reporting service, allowing users to retrieve and download survey results.

Rationale

This module is created to retrieve survey results which can be used for program accountability purposes.

Required Interface

```
userExists(UserID); surveyExists(SurveyID);
```

Provided Interface

```
SurveyProfile getSurveyResults(SurveyFilters) throws
IncompleteDataException;
```

Description:
Gets a list of survey results given the survey filters.

Parameters:
SurveyFilters: The filters provided to indicate which survey results to retrieve. Filters will include date of survey, date of submission, survey ID, survey status.

Returns:

All filtered survey results.

Exceptions:
IncompleteDataException: If the SurveyFilters did not represent any existing survey results.

```
GenerateSurveyReport(SurveyReportResults) throws
FailedCreationException; IncompleteDataException;
```

Description:
Creates an Excel version of the survey report results.

Parameters:
SurveyReportResults: Data returned by getSurveyResults.

Exceptions:
IncompleteDataException: If the SurveyReportResults did not represent any existing survey results.
FailedCreationException: If unable to create the Excel file.

2.4 Users Module

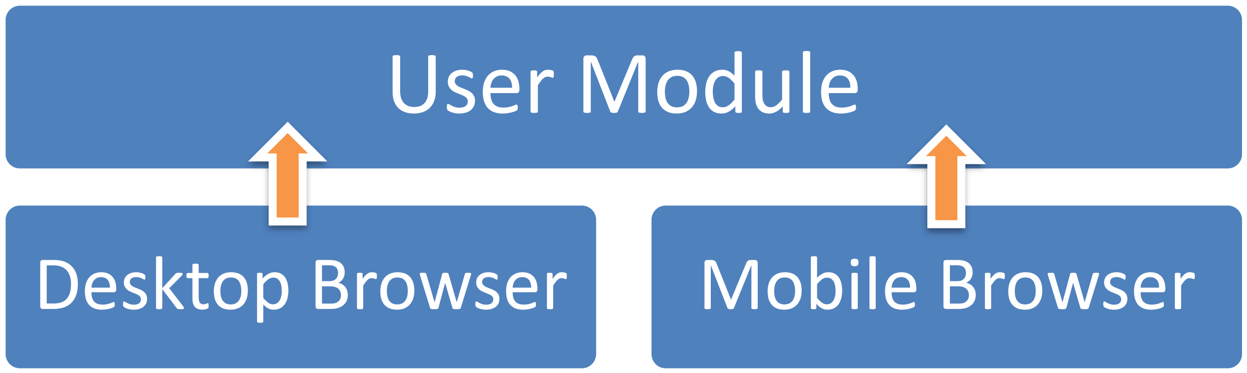

Figure 2-4 Architecture of the User Module

Purpose

The purpose of this module is for users to complete and submit surveys.

Rational

This module is created to enable users to access an online surveys, answer questions, and submit their responses to the data store.

High-Level Module Design

The user module is broken down into high-level modules, as shown in Figure 2-4.

Required Interface
This module has no required interface.

Provided Interface
The provided interface of this module is the union of the provided interfaces of the following sub-modules:

- User Module (Desktop Browser)
- User Module (Mobile Browser)

For more detail, please see these modules' descriptions.

2.4.1 User Module (Desktop Browser)

Purpose
The purpose of this module is to provide access to and a persistent data store for all active surveys and survey results in the system.

Rationale
This module is created to allow users to participate in surveys via their browser.

Required Interface
```
SurveyExists(SurveyID);
```

Provided Interface
```
SubmitSurvey(SurveyAnswers) throws IncompleteDataException;
```

Description:
Collects survey question responses from users and submits them to the data store.

Parameters:
SurveyAnswers: Answers submitted by user.

Exceptions:
IncompleteDataException: If required fields are left empty, this survey will not be submitted.

```
SurveyProfile getSurveyProfile(SurveyID) throws
NoSuchProfileException;
```

Description:
Gets a survey profile given the survey's ID.

Parameters:
SurveyID: The survey ID of the profile to retrieve.

Returns:
The requested survey profile presented as a form for completion.

Exceptions:
NoSuchSurveyException: If the SurveyID did not represent an existing profile.

```
SurveyExists(SurveyID);
```

Description:
Determines if a survey profile exists in the data store.

Returns:
True if the profile exists, false otherwise.

Parameters:
SurveyID: The survey ID of the profile to check.

```
AudioPlay(AudioFile) throws AudioPlayException;
```

Description:
Plays an MP3 audio file of question and answers when clicked on.

Parameters:
AudioFile: The audio file to be played.

Exceptions:
AudioPlayException: If the audio file was not able to be found or played.

3 Flows

3.1 Administrators

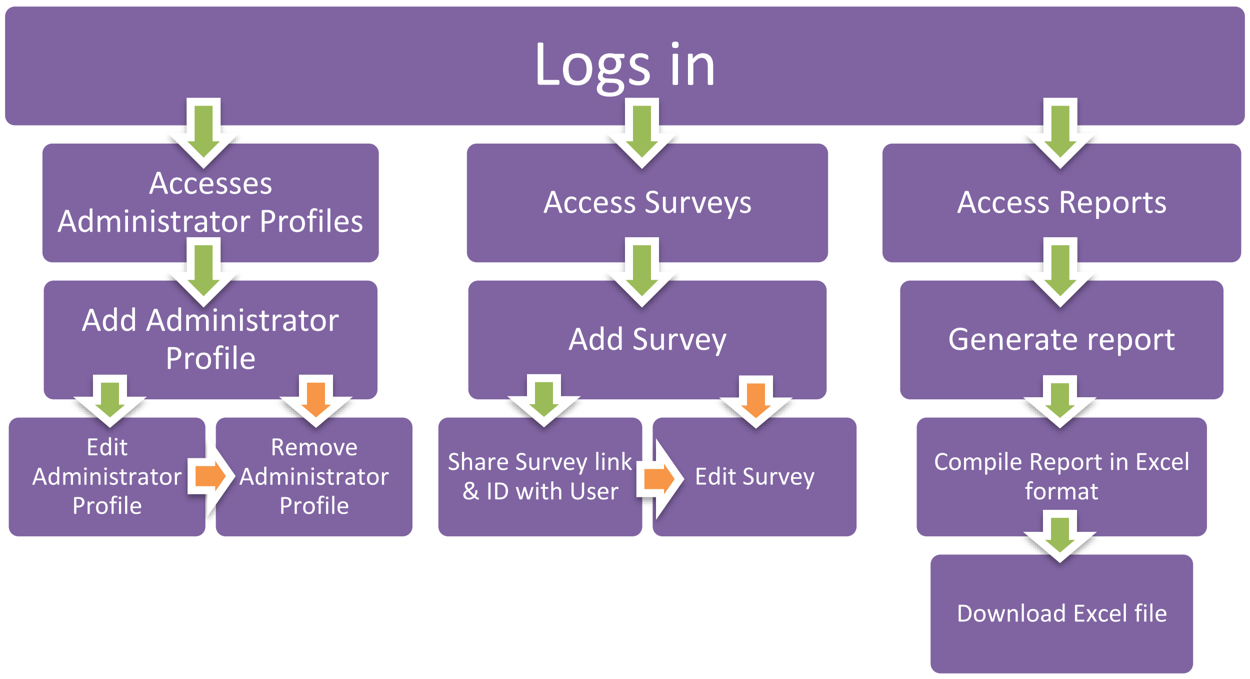
Logs in
Accesses Administrator Profiles
Add Administrator Profile
Edit Administrator Profile
Remove Administrator Profile
Access Surveys
Add Survey
Share Survey link & ID with User
Edit Survey
Access Reports
Generate report
Compile Report in Excel format
Download Excel file

Figure 3-1 Administrator Flowchart

3.2 Users

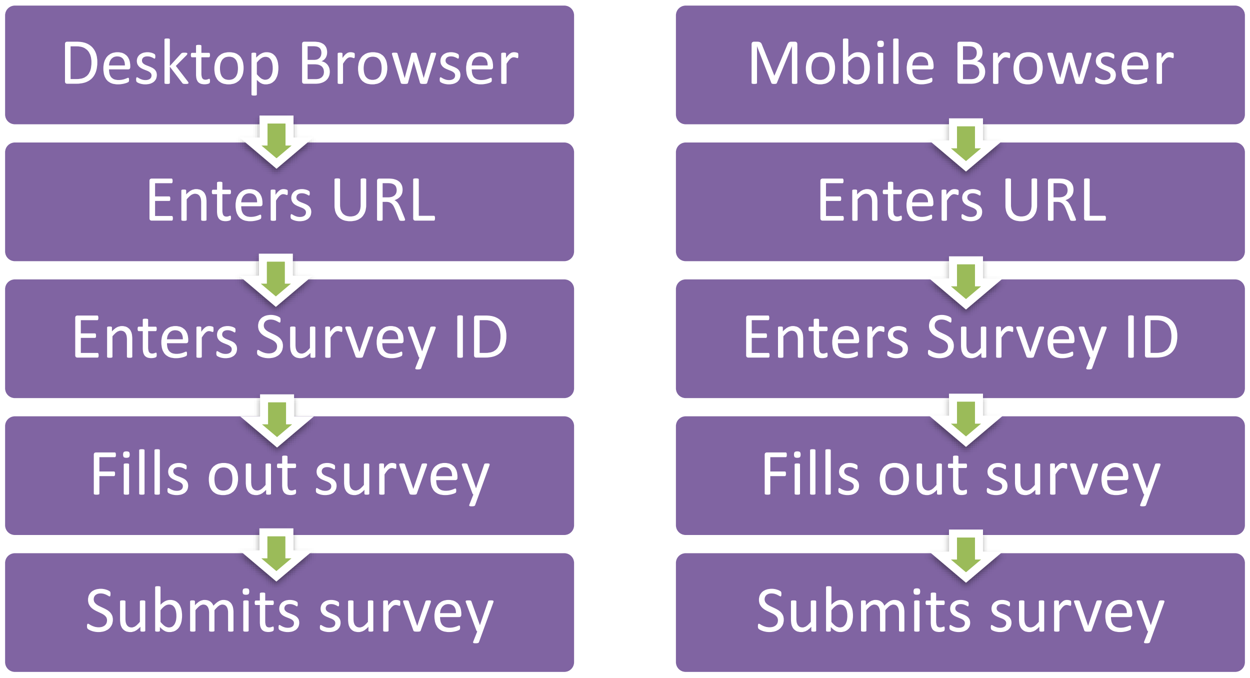

Figure 3-2 User Flowchart

4 Interfaces

The product will not interface with any third party products.

5 Internal Data Model

<Internal data model to be designed by development team>

6 Use Cases

6.1 Use Case 1: Administrator logs on

Main Flow

1. Administrator launches the application
2. System prompts Administrator for log-on details (email and password)
3. Administrator enters log-on details and submits them
4. System requests Administrator's log-on details from the sign on directory
5. System validates that Administrator's entered log-on details match those from the directory
6. System displays the main menu

Alternative Flow 4a: Incorrect log-on details

1. System tells Administrator their log-on has failed
2. Return to Main Flow step 2

6.2 Use Case 2: Administrator adds new administrator profile

Main Flow

1. Administrator enters profile information and submits form
2. System validates that all required fields have been filled in and are unique where required
3. System commits information to database

Alternative Flow 3a: Missing required fields

1. System tells Administrator missing fields are required
2. Return to Main Flow step 1

Alternative Flow 3b: Email address not unique

1. System tells Administrator email address is not unique
2. Return to Main Flow step 1

6.3 Use Case 3: Administrator updates administrator profile

Main Flow

1. Administrator clicks on a specific profile from a list

2. Profile details are presented in the administrator profile form
3. Administrator makes necessary changes and submits form
4. System validates that all required fields have been filled in and are unique where required
5. System commits information to database

Alternative Flow 4a: Missing required fields

1. System tells Administrator missing fields are required
2. Return to Main Flow step 3

Alternative Flow 4b: Email address not unique

1. System tells Administrator email address is not unique
2. Return to Main Flow step 3

6.4 Use Case 4: Administrator removes administrator profile

Main Flow

1. Administrator clicks on delete icon beside a specific profile from a list
2. System prompts Administrator for confirmation of profile deletion
3. Administrator approves action
4. System removes administrator profile from database

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Return to Main Flow step 1

6.5 Use Case 5: Administrator adds new survey

Main Flow

1. Administrator enters profile information and submits form
2. System validates that all required fields have been filled in and are unique where required
3. System commits information to database

Alternative Flow 3a: Missing required fields

1. System tells Administrator missing fields are required
2. Return to Main Flow step 1

Alternative Flow 3b: Email address not unique

1. System tells Administrator email address is not unique
2. Return to Main Flow step 1

6.6 Use Case 6: Administrator edits survey

Main Flow

1. Administrator clicks on a specific profile from a list
2. Profile details are presented in the administrator profile form
3. Administrator makes necessary changes and submits form
4. System validates that all required fields have been filled in and are unique where required
5. System commits information to database

Alternative Flow 4a: Missing required fields

1. System tells Administrator missing fields are required
2. Return to Main Flow step 3

Alternative Flow 4b: Email address not unique

1. System tells Administrator email address is not unique
2. Return to Main Flow step 3

6.7 Use Case 7: Administrator adds new question to survey

Main Flow

1. Administrator clicks on “add question” link on survey details page
2. A pop up form with a field for the question is presented
3. Administrator enters question and submits the form
4. Question is added to database, pop up form disappears, survey details page is updated with new question

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Pop up form disappears
3. Return to Main Flow step 1

Alternative Flow 4a: Question field is left empty

1. System tells Administrator missing fields are required
2. Return to Main Flow step 3

6.8 Use Case 8: Administrator edits survey question

Main Flow

1. Administrator clicks on question on survey details page
2. A pop up form with a field for the question is presented
3. Administrator edits question and submits the form
4. Question is updated in database, pop up form disappears, survey details page is updated with question changes

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Pop up form disappears
3. Return to Main Flow step 1

Alternative Flow 4a: Question field is left empty

1. System tells Administrator missing fields are required
2. Return to Main Flow step 3

6.9 Use Case 9: Administrator deletes survey question

Main Flow

1. Administrator clicks on delete icon beside question on survey details page
2. System prompts Administrator for confirmation of question deletion
3. Administrator approves action
4. System removes survey question and all associated answers from database, survey details page is updated with question changes

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Return to Main Flow step 1

6.10 Use Case 10: Administrator adds new answer to survey question

Main Flow

1. Administrator clicks on “add answer” link beside survey question

2. A pop up form with a field for the answer is presented
3. Administrator enters answer and submits the form
4. Answer is added to database, pop up form disappears, survey details page is updated with new answer

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Pop up form disappears
3. Return to Main Flow step 1

Alternative Flow 4a: Answer field is left empty

1. System tells Administrator missing fields are required
2. Return to Main Flow step 3

6.11 Use Case 11: Administrator edits survey question answer

Main Flow

1. Administrator clicks on answer on survey details page
2. A pop up form with a field for the answer is presented
3. Administrator edits answer and submits the form
4. Answer is updated in database, pop up form disappears, survey details page is updated with answer changes

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Pop up form disappears
3. Return to Main Flow step 1

Alternative Flow 4a: Answer field is left empty

1. System tells Administrator missing fields are required
2. Return to Main Flow step 3

6.12 Use Case 12: Administrator deletes survey question answer

Main Flow

1. Administrator clicks on delete icon beside answer on survey details page

2. System prompts Administrator for confirmation of answer deletion
3. Administrator approves action
4. System removes survey answer from database, survey details page is updated with answer changes

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Return to Main Flow step 1

6.13 Use Case 13: Administrator assigns mp3 audio file to survey question

Main Flow

1. Administrator clicks on audio icon beside a survey question
2. A pop up form with a file upload form is presented
3. Administrator locates file on their computer and submits form
4. System uploads mp3 file to server and associates file with question in database, pop up form disappears, survey details page is updated with audio file changes

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Return to Main Flow step 1

Alternative Flow 4a: No file is chosen

1. System tells Administrator a file must be chosen
2. Return to Main Flow step 3

Alternative Flow 4a: File is not in mp3 format

1. System tells Administrator file must be in mp3 format
2. Return to Main Flow step 3

Alternative Flow 4a: File cannot be uploaded

1. System tells Administrator file could not be uploaded
2. Return to Main Flow step 3

6.14 Use Case 14: Administrator deletes audio file

Main Flow

1. Administrator clicks on delete icon beside audio file on survey details page
2. System prompts Administrator for confirmation of audio file deletion
3. Administrator approves action
4. System removes audio file from server and association from database, survey details page is updated with audio file changes

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Return to Main Flow step 1

6.15 Use Case 15: Administrator disables survey

Main Flow

1. Administrator changes status to “disabled” on survey details page
2. System prompts Administrator for confirmation of status change
3. Administrator approves action
4. System updates survey status in database

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Return to Main Flow step 1

6.16 Use Case 16: Administrator enables survey

Main Flow

1. Administrator changes status to “enabled” on survey details page
2. System prompts Administrator for confirmation of status change
3. Administrator approves action
4. System updates survey status in database

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels action
2. Return to Main Flow step 1

6.17 Use Case 17: Administrator runs survey report

Main Flow

1. Administrator selects survey from list of surveys, enters optional survey result start and end dates, and submits query

2. System retrieves all survey result records which match criteria and prompts Administrator to create Excel document
3. Administrator approves action
4. System creates Excel document containing all submitted surveys with corresponding answers, prompts Administrator to download file
5. Administrator downloads file

Alternative Flow 2a: No search results found

1. System tells Administrator no results were returned
2. Return to Main Flow step 1

Alternative Flow 3a: Administrator cancels action

1. Administrator cancels Excel file creation
2. Return to Main Flow step 1

Alternative Flow 4a: Excel file cannot be created

1. System tells Administrator Excel report could not be created
2. Return to Main Flow step 1

6.18 Use Case 18: User accesses survey using desktop browser

Main Flow

1. User launches browser on their desktop computer and visits URL for survey page
2. System prompts User for ID of survey
3. User submits survey ID and System presents User with corresponding survey listing questions, corresponding answers in radio button format, and optional audio file icons beside each question

Alternative Flow 1a: Survey URL cannot be found

1. Browser notifies User that website cannot be loaded
2. Return to Main Flow step 1

Alternative Flow 3a: Survey ID cannot be found

1. System notifies User that survey ID cannot be located
2. Return to Main Flow step 1

Alternative Flow 3b: Survey ID in disabled status

1. System notifies User that survey is no longer available
2. Return to Main Flow step 1

6.19 Use Case 19: User accesses survey using mobile browser

Main Flow

1. User launches browser on their mobile device and visits URL for survey page
2. System prompts User for ID of survey
3. User submits survey ID and System presents User with corresponding survey listing questions, corresponding answers in radio button format, and optional audio file icons beside each question

Alternative Flow 1a: Survey URL cannot be found

1. Browser notifies User that website cannot be loaded
2. Return to Main Flow step 1

Alternative Flow 3a: Survey ID cannot be found

1. System notifies User that survey ID cannot be located
2. Return to Main Flow step 1

Alternative Flow 3b: Survey ID in disabled status

1. System notifies User that survey is no longer available
2. Return to Main Flow step 1

6.20 Use Case 20: User completes survey

Main Flow

1. User answers survey questions and submits survey
2. System saves survey response and thanks User for their cooperation

Alternative Flow 2a: Survey incomplete

1. System notifies User that not all questions have been answered
2. Return to Main Flow step 1

Alternative Flow 2b: Unable to save survey results

1. System notifies User that survey response could not be saved
2. Return to Main Flow step 1

6.21 Use Case 21: User listens to audio file

Main Flow

1. User clicks on audio file icon beside question
2. System plays audio file

Alternative Flow 2a: Audio file not found

1. System notifies User that audio file could not be found
2. Return to Main Flow step 1