



**SOFTWARE REQUIREMENTS SPECIFICATION (SRS) FOR**  
**320 Green Team Project**

**Version 1.0**  
**October 22, 2015**

**Prepared for:**  
**Sunderland/Leverett, MA Health Inspector**

**Prepared by:**  
**Ather Akhtar, Andrew Chang, Peter Marathas, Andrew Marchetti,**  
**Michael Markman, Eric Maryea, Shawn Sowersby, and Alex Sullivan.**  
**University of Massachusetts**  
**Amherst, MA 01003**

## 320 GREEN TEAM PROJECT: HEALTH-E TABLE OF CONTENTS

Section	Page
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 PURPOSE . . . . .	1
1.2 DOCUMENT CONVENTIONS . . . . .	1
1.3 PRODUCT SCOPE . . . . .	1
1.3.1 Sub Projects . . . . .	1
1.4 INTENDED AUDIENCE . . . . .	2
1.5 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS . . . . .	3
<b>2 OVERALL DESCRIPTION</b>	<b>4</b>
2.1 PRODUCT PERSPECTIVE . . . . .	4
2.2 PRODUCT FUNCTIONS . . . . .	4
2.2.1 Data Entry (Form) . . . . .	4
2.2.2 Browsing & Search . . . . .	4
2.2.3 Delayed Upload . . . . .	5
2.3 USER CHARACTERISTICS . . . . .	5
2.4 OPERATING ENVIRONMENT . . . . .	5
2.5 CONSTRAINTS . . . . .	5
2.5.1 Data Entry (Form) . . . . .	5
2.5.2 Browsing & Search . . . . .	5
2.5.3 Delayed Upload . . . . .	5
2.6 USER DOCUMENTATION . . . . .	5
2.7 ASSUMPTIONS & DEPENDENCIES . . . . .	6
2.7.1 Data Entry (Form) . . . . .	6
2.7.2 Browsing & Search . . . . .	6
2.7.3 Delayed Upload . . . . .	6
<b>3 EXTERNAL INTERFACE REQUIREMENTS</b>	<b>7</b>
3.1 USER INTERFACES . . . . .	7
3.2 HARDWARE INTERFACES . . . . .	7
3.3 SOFTWARE INTERFACES . . . . .	7
3.3.1 Operating System . . . . .	7
3.3.2 Web Server . . . . .	7
3.3.3 Database . . . . .	7
3.4 COMMUNICATION INTERFACES . . . . .	7
3.4.1 Form Interface . . . . .	7
3.4.2 Web Communication . . . . .	7
3.4.3 USB Communication . . . . .	7

<b>4</b>	<b>NONFUNCTIONAL REQUIREMENTS</b>	<b>8</b>
4.1	PERFORMANCE REQUIREMENTS . . . . .	8
4.2	SAFETY REQUIREMENTS . . . . .	8
4.3	SECURITY REQUIREMENTS . . . . .	8
4.4	SOFTWARE QUALITY ATTRIBUTES . . . . .	8
4.5	BUSINESS ROLES . . . . .	8
<b>5</b>	<b>FUNCTIONAL REQUIREMENTS</b>	<b>9</b>
5.1	DATA ENTRY (FORMS) . . . . .	9
5.1.1	<i>Use Case 1</i> . . . . .	9
5.1.2	<i>Use Case 2</i> . . . . .	9
5.1.3	<i>Use Case 3</i> . . . . .	10
5.2	BROWSING & SEARCH . . . . .	11
5.2.1	<i>Use Case 1</i> . . . . .	11
5.3	DELAYED UPLOAD . . . . .	12
5.3.1	<i>Use Case 1</i> . . . . .	12
5.3.2	<i>Use Case 2</i> . . . . .	13
<b>6</b>	<b>TEST PLAN</b>	<b>15</b>
6.1	OBJECTIVES . . . . .	15
6.2	TESTING STRATEGY . . . . .	15
6.3	SCOPE . . . . .	15
6.4	FEATURES TO BE TESTED . . . . .	15
6.5	APPROACH . . . . .	16
6.6	PASS/FAIL CRITERIA . . . . .	16
6.7	SCHEDULE . . . . .	16
<b>7</b>	<b>FUTURE EXTENSIONS</b>	<b>17</b>
<b>8</b>	<b>APPENDIX A. Example Screens</b>	<b>18</b>
<b>9</b>	<b>APPENDIX B. [insert name here]</b>	<b>19</b>

# 1 INTRODUCTION

## 1.1 PURPOSE

Convert current, physical method of archiving Board of Health data from restaurant inspections, well water quality tests, and septic tank inspections to a new, digital method. This document provides a high level overview of the entire system then goes into detail on the tablet applications form entry, browsing and search, and delayed upload.

## 1.2 DOCUMENT CONVENTIONS

WRITE HERE

## 1.3 SCOPE

This project contains multiple sub-projects and systems.

This new digital method requires a database to hold different types of inspection forms (restaurant, water well, septic tank). The database will be searchable and sortable by keywords or established parameters thus improving accessibility and simplifying analysis. Storing this data digitally will allow the customer to better analyze patterns and create a more permanent location for the data to reside. It will act as an intuitive tool to help the inspector utilize and access data in the future. To increase usability of this database, a user interface must be created to provide access and retrieval of information. This application will focus on usability of the database and a later described portion will focus on portability, thus one can think of this application as residing in an office environment. The application should include a convenient and easy method to search, sort, and analyze data from the restaurant inspections, well water quality reports, and septic tank inspections. Septic tank and well water quality inspection data should also be viewable on a geographical map with the ability to show change over time. This will illuminate differences and trends in the data that may help explain and predict the cause of said trends. The cataloguing of restaurant health inspection data could be facilitated by a form on a portable application (i.e. mobile, tablet). This portable application should be designed to locally store the inputted data and convert it to a form readable by the database so it can add the report to the rest of the database when connected. Overall, creating and structuring this database for the customer and providing easy-to-use tools for accessing it, will assist him in his inspections and simplify the retrieval of key information the customer might be interested in at a later date.

### 1.3.1 Sub Projects

- Central Database
- Form Application for Tablet
- Application

*Central Database:*

A central database with consistent access to data will create the integrity of the entire system. The three distinct data entities identified were: Well water inspections, septic tank inspections, and restaurant inspections, which will all need to contain location to differentiate towns. The attributes of the models will correspond to the required fields that the customer must fill in on their forms during inspections. The data for the restaurant inspections shall contain all fields that can be found on the restaurant inspection form. The data for water inspections must contain all of the relevant fields, such as water quality, measurements, address of water well, flow rate, name of the company that drilled the well, and the permit and lot number.

Similarly, the data for the septic inspection will hold information such as the address, lot number, size, capacity, condition of the tank, date of last inspection, and quality of gallons pumped in and out. The type of data stored must be able to be changed in the event of new legislation.

The two features existing in the database include a basic database that provides basic queries to GET, POST, PUT, and DELETE records safely and securely. The other feature is to create and design a federated database.

#### *Form Application for a Tablet:*

This application will be focused on usability in attempt to make it as simple as possible to enter form data. The application will be able to store data locally on the device followed by having the ability to upload to the office-based application noted (i.e. desktop application or web -application). This app will replicate the forms that the customer requires for every inspection. For security reasons, the app will have a login screen for user authentication. This application will be able to either send the document to a printer or email it on the spot as well as being able to receive an e-signature. The application should allow the user to view contextual information based on data that is associated with the users current task. For example, when the user is going to inspect a restaurant, they should be shown previous inspection results.

The form application will have the following main areas that correspond with the functionality above. User authentication as stated. Data entry as stated where the user interface allows the inspector to flow through the inspection process dynamically. Data entry (GIS) is required in order to locate well, septic, and restaurant location. Form printing is another important point as a physical copy of a restaurant inspection will most likely be required to provide to the restaurant owner. Browsing and search is also a feature of the application which includes pulling up relevant data of a current restaurant. The final important feature is the ability for delayed upload. This will allow the tablet to flow data into the central database that communicates with the main application on the web.

#### *Application (Usable, Connected to Database):*

An application will be created for viewing all past records stored in the database. The application should include three different sections for restaurant, water and septic. The application will need to communicate with the central databases containing all information of past record. The site will provide an interface for querying data, including searching by location or violation. The application will contain user authentication that, once validated for an administrator, will allow for the submission of new records. Without login credentials, the information is read-only. The site will use the accepted regulation form as a base to record data from a new restaurant, water, or septic inspection and then send this new data to the relevant database. Additional functionality including mapping is mentioned next.

Features described have included user authentication, data entry for eligible users, reporting and search as well as mapping. The reporting and search will include the ability to retrieve a general summary of 1 of 3 sections (restaurants, wells, septic tanks) that will be able to be downloaded and printed for use in a physical report. The application will also include data mapping. This map will contain points of location of wells, septic tanks, and restaurants which can then be queried and visualized.

## **1.4 INTENDED AUDIENCE**

The intended audience is for health inspectors who want to use a digital alternative to record and store information as an alternative to physical methods. The primary audience will be the health inspector of the town of Leverett and Sunderland Massachusetts.

## 1.5 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

The products name is Health.e standing for Health electronics as it is designed to digitize information on health inspections, well water and septic tanks.

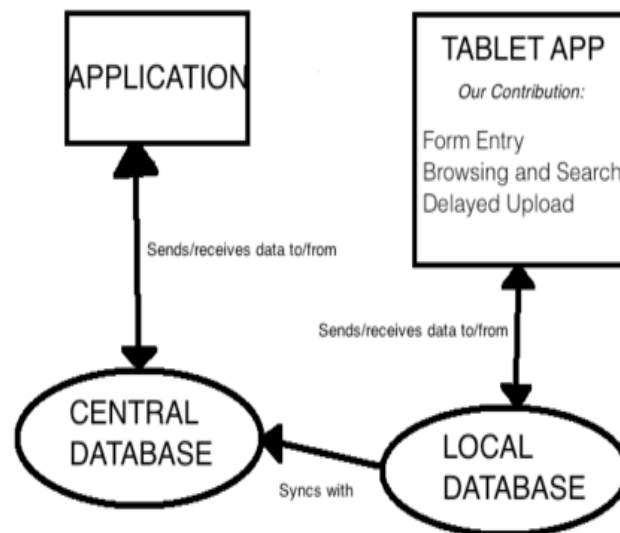
Term or Acronym	Definition
Alpha test	Limited release(s) to selected, outside testers
Beta test	Limited release(s) to cooperating customers wanting early access to developing systems
Final test	aka, Acceptance test, release of full functionality to customer for approval
DFD	Data Flow Diagram
SDD	Software Design Document, aka SDS, Software Design Specification
SRS	Software Requirements Specification
SSRS	System and Software Requirements Specification

## 2 OVERALL DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE

This product, in all of its glory, will be replacing the primary user's current pen and paper system for documenting inspection forms. Rather than filling out paper forms on site and handing the inspectee a carbon copy of the form, the tablet app will allow the inspector to fill out a digitalized version of the same form, and print said form from a wireless printer on site for the inspectee to obtain. Also, rather than storing paper files in a filing cabinet that may never be looked at again, the health inspector may, with our product, review previous forms with ease by searching for them with our comprehensive browsing and search functionality. This way, files may be found in a matter of seconds. Another aspect of this product will be the delayed upload in which forms that have been filled out on the tablet and stored locally will be synced with the central database so that they may be integrated with the main application. This eliminates the need for the health inspector to "file" each form after completing it.

The form entry, browsing and search, and delayed upload functionalities are all subprojects for the entire tablet app, which is, in and of itself, a subproject of the main project. Our contribution to the main project is depicted below:



### 2.2 PRODUCT FUNCTIONS

#### 2.2.1 Data Entry (Form)

1. The user chooses one of the following inspection forms to fill in: restaurant, septic tank, or water well.
2. The tablet then displays the chosen form to the user with blank, fillable forms.

#### 2.2.2 Browsing & Search

1. The user types in a location of the desired restaurant, septic tank, or water well.
2. The tablet displays forms from its database relevant to the search requirements.

### **2.2.3 Delayed Upload**

1. The tablet synchronizes with the external database and uploads either specific or all form entries from the tablet to the external database once a connection has been established.

## **2.3 USER CHARACTERISTICS**

The form entry, browsing and search, and delayed upload parts of the tablet will primarily be used by health inspectors. They will have full access to all functional aspects listed in section 2.2. Unregistered guests will have limited, read-only access to the main application of the project while only the registered health inspectors may access the tablet application.

## **2.4 OPERATING ENVIROMENT**

The form entry, browsing and search, and delayed upload aspects of this project will all exist on the same platform: a tablet. The tablet should have a small amount of memory allocated to the app so that data forms may be stored locally until uploaded to the central database.

## **2.5 CONSTRAINTS**

### **2.5.1 Data Entry (Form)**

The form entry itself is primarily based on inspection forms created by the state and used by the town. The entry fields on each form entry in the tablet has to match the required sections on the current inspection forms. Also, there will be a finite amount of form entries the user could generate with the tablet since the tablet will have a limited amount of memory space. With recent tablet device improvements, the chances of filling up the memory on the tablet is low. With a good design of the entire project, modern day memory capacities will suffice to hold the data required.

### **2.5.2 Browsing & Search**

Browsing and searching features are constrained by the design of the database itself. With a good design and structure, these features will be user intuitive and minimal effort will be needed by the user to use these features. Browsing and searching the form entries is constrained by the number of form entries in the database on the tablet.

### **2.5.3 Delayed Upload**

The delayed upload is constrained by the health inspector's access to WiFi or some form of manual docking technology. Either method may be used in order for the local memory on the tablet to sync with the central database, but one must exist.

## **2.6 USER DOCUMENTATION**

There will be a GitHub Wiki that will help guide the user through all of the basic functionality of the tablet app, and in general, the application as a whole. The Wiki will be user friendly, easily comprehensible, and will touch on all of the important features of the app that the primary user should be familiar with.



## **2.7 ASSUMPTIONS & DEPENDENCIES**

### **2.7.1 Data Entry (Form)**

Our implementation of the form entry on the tablet app depends on how the central database is constructed. Each field that the database contains for each type of form must be represented in the UI of the tablet app so that the health inspector may enter all of the fields correctly. The assumption here being that any changes made to the physical forms will be updated accordingly in the database, and thus updated in the digital forms as well.

### **2.7.2 Browsing & Search**

Our implementation of browsing and search on the tablet app relies partly on the implementation of the database since fields that will ultimately be searched for within the tablet app must be connected to the database in some form. However, the only main assumption for this aspect of the project is that the files being searched for still exist on the tablet. There is a possibility that files that have been removed from the tablet may be searched for, but such a case will be handled by a file not found message.

### **2.7.3 Delayed Upload**

We are assuming that the health inspector will have WiFi access in order for the tablet app to upload data from its local database to the central database via a syncing process. In the case that this assumption is incorrect and the user does not have WiFi access, a manual docking process will have to occur between the tablet and the health inspector's computer on which the main application exists. This assumes that the health inspector has the proper cable to connect the tablet to said computer, but this is a reasonable assumption since such a cable typically is included with the purchase of a tablet since it serves the dual purpose of a charger. The delayed upload also depends on the construction of the database and its fields as well as the application itself since it must sync with said application.

## **3 EXTERNAL INTERFACE REQUIREMENTS**

### **3.1 USER INTERFACES**

The user interface for the system will be an app based application on a tablet. The user will enter information to pass the user authentication. The user will then be able to select which forms to access. The user will be able to enter form data in the app on the replicated forms for the inspection. The user will also be able to send the document to a printer or email it. The user will also be able to accept e-signatures directly inside the app. The app will also have the ability for the user to browse and search for information on the app. The user will be able to select for a delayed upload as well.

### **3.2 HARDWARE INTERFACES**

The hardware interfaces of the system will be run on an android device. An internet connection to allow the software interfaces to connect to the internet to sync with files in the database.

### **3.3 SOFTWARE INTERFACES**

#### **3.3.1 Operating System**

The software is being designed to run on Android OS.

#### **3.3.2 Web Server**

The software will be able to sync with a database

#### **3.3.3 Database**

The software will use a federated database to store all the data.

### **3.4 COMMUNICATION INTERFACES**

#### **3.4.1 Form Interface**

The application will provide electronic forms that the user needs.

#### **3.4.2 Web Communication**

The application will be able to wirelessly sync with the database.

#### **3.4.3 USB Communication**

The application will be able to sync through a usb cable with the database.

## **4 NONFUNCTIONAL REQUIREMENTS**

### **4.1 PERFORMANCE REQUIREMENTS**

For the tablet application, the user expects to experience appropriate speed performance. The focus here is browsing and searching. Getting information from the database should never operate at a speed that will keep the user waiting.

### **4.2 SAFETY REQUIREMENTS**

When it comes to the tablet applications ability to do delayed uploading, safety becomes a factor. Its essential that the data entered on site with the tablet is properly indexed and saved. The tablet must make it back to the users desktop without any loss of this data.

### **4.3 SECURITY REQUIREMENTS**

Its vital that data entry is allowed only to authorized users. User authentication should take care of this, and only let administrators actually enter information into the database. However, we shouldn't have worry about browsing and search, as anyone can use the feature because the information is public.

### **4.4 SOFTWARE QUALITY ATTRIBUTES**

The biggest goal here is that the tablet application is extremely user friendly. From what we know, the client who is going to be using the application is not tech-savvy. Between the user interface and functionality, the highlight must be that the user can easily learn how to use the application effectively.

### **4.5 BUSINESS ROLES**

Any user with access to the tablet application can do browsing and searching for information on the wells, septic systems and restaurants. All of this information is public. However, only a registered administrator is allowed to access the form entry and delayed upload features. They are the sole owners of the ability to add new information.

## 5 FUNCTIONAL REQUIREMENTS

### 5.1 DATA ENTRY (FORMS)

#### 5.1.1 *Use Case 1*

*Title:* **Fill out septic tank inspection form**

*Primary Actor:* User

*Secondary Actor:* System

*Trigger:* User selects septic inspection form.

*Precondition:* User successfully logged in and accessed the form entry user interface for the septic tank.

*Postcondition:* Data is stored to be submitted to a central database

*Main Scenario:*

1. System presents modifiable septic tank information form.
2. User fills out each required field of the form.
3. User submits the form on tablet.
4. System determines that each required and non-required field entered is valid.
5. System creates an accessible local copy.
6. System displays a response when successfully entering form.
7. User safely closes out of process (data is preserved).

*Exception Scenario:*

1. User forgets to enter information in 'required field'.
  - (a) System notifies user of empty fields and does not submit.
2. User enters an invalid form entry.
  - (a) System displays invalid form entry and does not submit.
3. System shuts down during data submission.
  - (a) System can show user whether or not data was submitted successfully.

#### 5.1.2 *Use Case 2*

*Title:* **Fill out well water inspection form**

*Primary Actor:* User

*Secondary Actor:* System

*Trigger:* User selects well water inspection form.

*Precondition:* Successfully logged in and accessed the form entry user interface for water well inspection.

*Postcondition:* Data is stored to be submitted to a central database

*Main Scenario:*

1. System presents modifiable well water information form.
2. User fills out each *required* field of the form.

3. User submits the form on tablet.
4. System determines that each required and non-required field entered is valid.
5. System allocates memory and creates an accessible local copy.
6. System displays a response when successfully entering form.
7. User safely closes out of process (data is preserved).

*Exception Scenarios:*

1. User forgets to enter information in 'required field'.
  - (a) System notifies user of empty fields and does not submit.
  - (b) User presented with partially filled out form.
2. User enters an invalid form entry.
  - (a) System displays invalid form entry and does not submit.
  - (b) User presented with partially filled out form.
3. System shuts down during data submission.
  - (a) System shows user whether or not data was submitted successfully.

### **5.1.3 Use Case 3**

*Title:* **Fill out restaurant inspection form**

*Primary Actor:* User

*Secondary Actor:* System

*Trigger:* User selects restaurant inspection form to fill out.

*Precondition:* User has passed login authentication, and has chosen to submit data for a restaurant inspection.

*Postcondition:* Data is stored to be submitted to a central database

*Main Scenario:*

1. The page for restaurant inspection form is shown.
2. User records location and name of restaurant
3. User fills out fields in form to record data from inspection
4. User clicks save button.
5. Data is stored locally

*Exception Scenario:*

1. User forgets to enter information in 'required field'.
  - (a) System notifies user of empty fields and does not submit.
  - (b) User presented with partially filled out form.
2. User enters an invalid form entry.

- (a) System displays invalid form entry and does not submit.
- (b) User presented with partially filled out form.
- 3. System shuts down during data submission.
  - (a) System shows user whether or not data was submitted successfully.

## 5.2 BROWSING & SEARCH

### 5.2.1 Use Case 1

*Title:* **Search for prior inspection report**

*Primary Actor:* User

*Secondary Actor:* System

*Trigger:* User selects septic inspection form to fill out.

*Precondition:* User has logged in and is on the lookup report screen.

*Postcondition:* System displays selected prior report of a location to the user.

*Main Scenario:*

1. User enters an address or location title for the specified document
2. System queries external database with location and retrieves records with a matching location
3. System displays a list of retrieved records and their dates, sorted by date
4. User is given option to narrow results by inspection type (restaurant, water well, septic) or keyword
5. If user chooses narrowing option, remove from the list any reports void of the keyword or inspection type
6. User selects from list the desired report
7. System displays screen for relevant information from record

*Exception Scenario:*

1. System cannot connect to database.
  - (a) Screen displays connection error message and asks user to try again later.
  - (b) User selects 'Ok' button and system returns to search screen
2. The database returns no record for the given location.
  - (a) System displays "No matches found".
  - (b) User clicks on 'Ok' button to return to the lookup by location screen.
3. None of the records match the inputted keyword or inspection type
  - (a) Display message, "No reports match the specified keyword"
  - (b) User clicks 'Ok', return to step 4

## 5.3 DELAYED UPLOAD

### 5.3.1 Use Case 1

**Title:** User attempts to upload a specific piece of form data.

*Primary Actor:* System

*Secondary Actor:* User

*Trigger:* User selects septic inspection form to fill out.

*Precondition:* User has completed and saved some type of form data, and is associated with some online database. User is on the start page and has signed in successfully. Alternatively, User opts to send form data immediately after saving to device and begins at step 2.

*Postcondition:* Form data has not already been submitted to the database and is added successfully. If conditions are not correct for proper form data upload, data is not uploaded.

*Main Scenario:*

1. User navigates the form data on the application to find the data they desire to upload.
2. User indicates that they want to upload the form data.
3. System verifies that the form data is acceptable for submission to online database.
4. System verifies that internet connection is available.
5. System verifies that connection to associated database is available.
6. System verifies that given form data has not already been submitted to the database.
7. System asks the user for confirmation to send the form data.
8. User gives confirmation of intent to send the form data
9. System adds the new form data to the database using the internet connection available.
10. System notifies the user of successful upload of the data.

*Exception Scenario:*

1. Form data is not acceptable for upload.
  - (a) System notifies the user that the form data is not acceptable for upload. System specifies which fields in the form data are unacceptable and asks that they be fixed before submission.
  - (b) User is returned to the form's viewing page.
2. No internet connection is available.
  - (a) System notifies the user that internet connection is not available and to try again once they are within access range.
  - (b) User is returned to the form's viewing page.
3. Only internet connections that are available are not Wi-Fi connections.
  - (a) System asks the user to verify that they would like to send the connection on the mobile network.
  - (b) If user accepts, use case proceeds as normal at step 4.
  - (c) If user refuses, system notifies the user to try uploading again once they are within access range.

- (d) User is returned to the form's viewing page.
- 4. System is unable to connect with the database.
  - (a) System notifies the user that it is impossible to connect to the database and to verify that the database is online.
  - (b) User is returned to the form's viewing page.
- 5. Form data has already been submitted to the database.
  - (a) System notifies the user that the form has already been uploaded before.
  - (b) User is returned to the form's viewing page.
- 6. User declines the confirmation to send the data.
  - (a) Form data is not sent. User is returned to the form's viewing page.
- 7. No internet connection is available.
  - (a) System notifies the user that internet connection is not available and to try again once they are within access range.
  - (b) User is returned to the form's viewing page.

### 5.3.2 Use Case 2

**Title:** Batch upload all forms that are not already uploaded.

*Primary Actor:* System

*Secondary Actor:* User, Database

*Trigger:* User selects septic inspection form to fill out.

*Precondition:* User has completed and saved some type of form data, and the device is associated with some online database. User is on the start page and has signed in successfully. Alternatively, setting for automatic upload is on and System engages automatically, starting at step 2.

*Postcondition:* Form data that has not already been submitted to the database and is acceptable for upload is added successfully. If conditions are not correct for proper form data upload, data is not uploaded.

*Main Scenario:*

1. User indicates that they would like to perform a batch upload of form data.
2. System verifies that internet connection is available.
3. System verifies that connection to associated database is available.
4. System cross-references form data on the device and on the database to find which data is on the device and not in the database.
5. System displays a list of the eligible form data on the device and requests the user confirms the data.
6. User selects the data they wish to upload to the database and confirms to send those forms.
7. System checks each data that is desired to be uploaded is acceptable to be added to the database.
8. System sends acceptable form data to the database to be added.
9. System notifies the user of which form data was successfully sent to the database.



*Exception Scenario:*

1. No internet connection is available.
  - (a) System notifies the user that internet connection is not available and to try again once they are within access range.
  - (b) User is returned to the main viewing page.
2. Only internet connections that are available are not Wi-Fi connections.
  - (a) System asks the user to verify that they would like to send the connection on the mobile network.
  - (b) If user accepts, use case proceeds as normal at step 3.
  - (c) If user refuses, system notifies the user to try uploading again once they are within access range.
  - (d) User is returned to the main viewing page.
3. System is unable to connect with the database.
  - (a) System notifies the user that it is impossible to connect to the database and to verify that the database is online.
  - (b) User is returned to the main viewing page.
4. All form data on the device is also on the database.
  - (a) System notifies the user that there is no data to be upload to the database.
  - (b) User is returned to the main viewing page.
5. User declines to upload data.
  - (a) User is returned to the main page.
6. Some form data is not acceptable to be uploaded to the database
  - (a) Any data that is not acceptable is removed from the list of data to be uploaded. After batch upload is finished, a notification is sent with a list of any data unacceptable to be uploaded.
7. No internet connection is available.
  - (a) System notifies the user that internet connection is not available and to try again once they are within access range.
  - (b) User is returned to the main viewing page.
8. System is unable to connect with the database.
  - (a) System notifies the user that it is impossible to connect to the database and to verify that the database is online.
  - (b) User is returned to the main viewing page.

## 6 TEST PLAN

### 6.1 OBJECTIVES

The objective of the test plan is to ensure a high level of confidence in the correctness and usefulness of the project deliverables.

### 6.2 TESTING STRATEGY

The strategy for testing the softwares data entry, browsing and search, and delayed update features is a combination of automatic and manual tests. The components deemed untestable by user interaction will be tested automatically by a set of custom scripts. The components that involve user interaction will be tested manually by testers and developers.

### 6.3 SCOPE

Testing for this section will be conducted throughout the development life cycle of the product.

### 6.4 FEATURES TO BE TESTED

Test cases will cover all the features of the software covered in the SRS. It will include the exception cases in the use cases so as to ensure that the product behaves in the anticipated manner.

TEST ID	SRS Document Reference	Features to be Tested (Test Description)
001	3.1a	Validate the data entry form feature for septic tanks. Tests cases will ensure that the user is able to input and upload the form in the expected manner. Test cases will also test for exceptions: 1) User misses a required entry, 2) User inputs invalid entry 3) System shutdown during data upload.
002	3.1b	Validate the data entry form feature for water well. Tests cases will ensure that the user is able to input and upload the form in the expected manner. Test cases will also test for exceptions: 1) User misses a required entry, 2) User inputs invalid entry 3) System shutdown during data upload.
003	3.1c	Validate the data entry form feature for restaurant inspections. Tests cases will ensure that the user is able to input and upload the form in the expected manner. Test cases will also test for exceptions: 1) User misses a required entry, 2) User inputs invalid entry 3) System shutdown during data upload.
004	3.2	Validate the browsing and search functions. Test cases will include scenarios for software not able to connect to database and search query yielding no results.
005	3.3a	Validate the delayed upload feature for specific form selected by user. Test cases will include tests for exceptional scenarios covered in SRS to ensure that the software behaves in an expected manner.
006	3.3b	Validate the batch delayed upload feature. Test cases will include tests for exceptional scenarios covered in SRS to ensure that the software behaves in an expected manner.

## 6.5 APPROACH

<b>Component Testing</b>	Individuals will test the components they are working on during the software development cycle.
<b>Integration Testing</b>	A detailed set of tests cases will be created and then executed. These test cases will cover all the requirements detailed in the SRS.
<b>Interface Testing</b>	To ensure that all aspects of the display are rendered correctly, the external interfaces regarding data entry forms and browsing and searching will be tested manually by using the user interface to perform all product tasks.
<b>Security Testing</b>	Security of the software will be tested by making attempts to access the data without proper authentication.
<b>Performance Testing</b>	Test users will interact with the system to ensure that the system and its search and browsing utility respond in a reasonable time.

## 6.6 PASS/FAIL CRITERIA

<b>Suspension Criteria</b>	Test case execution will be suspended in the event of discovery of a critical failure that affects the value of the tests.
<b>Resumption Criteria</b>	Once the developer thinks the problem causing the suspension has been resolved, test case may resume.
<b>Approval Criteria</b>	The results for each test case will be considered approved if the results meet the expected results description in the test case.

## 6.7 SCHEDULE

<b>Task</b>	<b>Week Performed</b>
Develop test cases	6
Develop scripts for automated testing	4
Execute tests	8
Report defects	6
Complete test report	8

## 7 FUTURE EXTENSIONS

- User Voice Interface
- Expand location
- Smart watch application/feature
- Additional OS Support

Software is currently localized to two specific towns. The application can be extended to cover more towns and locations in the future. Search and browse functions in the software currently supports keypad and mouse/finger interface. Future extension can include a user voice interface as well. Software can be extended to possibly include full support as a smartwatch application. Software currently supports Android OS. It can support other operating systems in the future.

## 8 APPENDIX A. Example Screens

*Include copies of specifications, mockups, prototypes, etc. supplied or derived from the customer. Appendices are labeled A, B, ... n. Reference each appendix as appropriate in the text of the document.*  
[ insert appendix A here ]

## 9 APPENDIX B. [insert name here]

[ insert appendix B here ]