1.     Introduction  
  
1.1 Description  
  
This document denotes the software requirement specification (SRS) for the project tentatively titled Sharktopus. Shartktopus is a web-based data management software system built specifically to address the needs of the Long Beach State Marine Biology graduate research team, often referred to as Shark Lab, in an effort to improve the efficiency and robustness of the data collection, retention, and recall in their research.  
  
1.2    Background  
  
The focal point of the Shark Lab research team, consisting of Dr. Lowe and his graduate students, is to collect data for marine life through acoustic telemetry, the science of collecting information on movement patterns through sound waves.  This is accomplished by placing transmitter tags in fish, which relay their position to a receiver as they come within detection proximity.  Another source of data comes from temperature logs about the environment in which the detections occurred.  
  
Together these input sources total millions of entries through which the research team must look, and they must correlate the data with each other by eye in order to extract relevant information about a tagged fish.  That is, transmitter tags must be linked to fish, which must be also be linked to temperature data to gain any insight into their movement behavior.  
  
To further confound the tediousness of this process, some detections of transmitter tags used by research teams other than Shark Lab are included with their data and must be filtered out.  In addition, the Shark Lab team employs a method of acquiring data which allows them obtain information quicker than they would normally be allowed.  Stationary “dummy” transmitter tags are placed around receivers to track movement patterns of real transmitter tags because transmitter information is collected frequently whereas receiver data can only be collected approximately every six months.  These tags generate many more useless entries to sift through.  
  
The gathered data is also distributed amongst the different team members and must be coordinated amongst themselves.  
  
The goal of Sharktopus is to address and alleviate these issues by providing a centralized repository of all sources of information that will be linked together by the system and that will also allow queries matching specified criteria.

1.5     Intended Audience and Reading Suggestions  
The intended readers for this document are:

1. Developers, who will read this in order to understand what must be implemented.  By doing so they will be able to create the project based on the client’s needs.

2. Testers, who will use the document in order to formulate any problems that may exist within the project.  
    3. Users of Sharktopus, who will read this in order to understand the software system’s capability.  
    4. Manual writers, who will use this document to understand the feature set.  
  
Reading suggestion:  
For the majority of this document, the term “data” will refer to receiver/transmitter data, environmental data, temperature logs, and any other relevant research data.  
  
  
1.6     Project Scope

This SRS covers descriptions of the functionalities and general features of Sharktopus.  The primary goal of Sharktopus is to store and retrieve acoustic telemetry data for marine biologists.  The data will be organized and presented in such a way as to provide easy access to information the biologists may find relevant.

2. Overall Description

2.1     Product Perspective

The current method of data entry used by Shark Lab is to store data in different files across multiple programs.  There is no single program to manage all data and display them in one convenient view with an option to view entries matching specific criteria.  Sharktopus aims to resolve these issues.  While Sharktopus is not considered a component of a larger product, it still requires a web server with features detailed in section 2.4 to function.

2.3     User Classes and Characteristics  
  
Each user will have access to Sharktopus via the Internet.  
  
**Super Administrator:**  
Super Administrators will be able to change software security settings, as well as grant access to the system as necessary to new users.  Super Administrators will be capable of modifying accessibility permissions for users and administrators.  They are also able to upload and update entries in the database, and they have all the functionalities of an administrator.  In addition, they are able to delete database information.   
  
**Administrator:**  
Administrators will be able to change software security settings, as well as grant access to the system as necessary to new users.  Administrators will be capable of modifying accessibility permissions for users.  They are also able to upload and update entries in the database, and they have all the functionalities of a user.  
  
**User:**  
Unless specifically granted privileges to do so by the administrator, a registered user will not be able to read or write data from or to the database.  Once given permission, a user is able to upload files to and update entries in the database as well as run queries over data and download the results.  Users will be able to take a file in a specific format and upload it to the system, which will automatically update the database.  Some users may only be given read-only permissions if the administrator classifies them as “guest users.”

2.5     User Documentation

Sharktopus users will be supplied with various tools to help gain an understanding of the software and its features, including:

1. An online help section that will assist new users with the database upload and download process.
2. An FAQ page in order to receive some basic tips about the application.  The FAQ format will follow typical web FAQ style where commonly asked questions will be listed with their respective answer.
3. Tooltips on various fields will be available both to help a user learn the system as it is being used.
4. Electronic manual pages to serve as a reference for the features available.

3.2     System Requirements

Nick

3.2.2 Requirement:  System must allow the creation of new accounts.

3.2.2.1 Use Case: Create Account  
Comments: A potential user must have an account before he or she can gain access to product features. Once the account has been created, the newly created user will be able to log in, but access to the system will be restricted by the default user permissions until an administrator modifies them.  
Main Actor: User  
Entry Condition: Must be no other existing account with the same name.  
Exit Condition: User has successfully created a new user account.  
Flow of Events:

1. User navigates to the create account page.

2. User fills out required information in the provided interface.

3. User confirms the new account is to be created.

4. User is presented with a confirmation screen.

Exceptions:

1. Invalid values in account creation interface.

1.1 User is alerted of the errors in the form.

1.2 User must provide valid information to continue the account creation process.

2.  User cancels account creation.

2.1 No user account is created.

2.2 User is informed of the cancellation of the operation.

Nick

3.2.9 Requirement: The system must allow an administrator to delete the account of a user.

3.2.9.1 Use Case: Delete User  
Main Actor: Administrator  
Entry Condition: The system must require a user with administrator privileges. User to be deleted must be present in the system.  
Exit Condition:User to be deleted is no longer present in the system and must not allow the deleted user to log on to the system.  
Flow of Events:  
    1. Administrator navigates to an administrative panel.  
    2. Administrator selects the option to delete a user.  
    3. Administrator selects the user to delete.  
    3. Administrator confirms the delete is to be completed.  
    4. The user selected is deleted from the system.  
    5. The administrator is informed of the completion of the operation.  
Exceptions:  
    1. Administrator cancels the delete operation

1.1 The Administrator is informed of the cancellation of the operation.

Nick

3.2.4 Requirement:  System will allow user to log in to access software features.

3.2.4.1 Use Case: User logs in  
Comments: The user will log in with a username and password.  
Main Actor: User  
Entry Condition: User must have an account.  
Exit Condition: User is logged in and able to access the system.  
Flow of Events:

1. User navigates to Log In View.

2. User fills out log-in form with user credentials.

3. User clicks log-in button.

4. User is logged in and is directed to that user’s account Page.

Exceptions:

1. The user enters incorrect credentials.

1.1 User is notified that either the password or user name was incorrect and is asked to resubmit log-in credentials.

Nick

3.2.5 Requirement:  System will allow a user to end session by logging out of the product.

3.2.5.1 Use Case: Log Out  
Comments: Logout button accessed from global header links.  
Main Actor: User  
Entry Condition: User must be logged in.  
Exit Condition: User has successfully ended the session with the product.  
Flow of Events:

1. User selects the option to log out.

2. User is logged out of the session.

Exceptions: None.

Edward

3.2.1 Requirement: System will allow user to views a list of data updates or changes.

3.2.1.1 Use Case: View Log of Data Updates  
Comment: A user can view a list of all changes to the data that may include accompanying comments by those who made the changes.  The update messages will appear in groups of approximately 20.  A user will see the most recent messages first by default but can view changes made up to a year prior.  
Main Actor: User  
Entry Conditions: User is logged in.  
Exit Conditions: User views the log of data updates.  
Flow of Events:

1. User navigates to Update Log View.

2. User views the most recent updates made during the day by default.

3. User can choose to view changes made over different time intervals (week/month/ etc.) up to a year prior.

4. Can opt to arrange changes by user?

Exceptions: None.

3.2.6 Requirement: System will allow users to query the database.

3.2.6.1 Use Case**:** Making a query through provided search fields.  
Comments: Items to be searched will include all receiver/transmitter data, environmental data, temperature logs, and any other relevant data.  The results of the query will be displayed in tabulated form, and an option to display all research data will also be provided.  
Main Actor: User  
Entry Conditions: User is logged in and is granted permission by an administrator.  
Exit Conditions: The results of the query are displayed.  
Flow of Events:

1. User inputs parameters to be searched in input fields if searching for particular entries.

2. User sends search request.  
3. The results are shown in a table.

Exceptions: None.   
  
3.2.7 Requirement: System will allow query results to be downloaded.

3.2.7.1 Use Case: Downloading  
Comments: The user downloads the results of a query into CSV format.  
Entry Conditions: The results of a query are displayed.  
Exit Conditions: A file containing the results is downloaded.  
Flow of Events:  
    1. User makes a query.  
    2. User opts to download results.  
    3. User is prompted for a file name and download location.  
    4. User enters the file name and selects the location.  
    5. A file containing the values of the query are downloaded to the user’s system to the location specified, with the file name specified, in CSV format.  
Exceptions:  
    1. User cancels the download operation.  
        1.1 The file is not downloaded to the user’s computer.  
  
Eric C.

3.2.8 Requirement: System will allow files to be uploaded into the database.

3.2.8.1 Use Case: Uploading  
Comments: The user will upload a file in CSV format to add to the database of research data.  In addition to appending the contents of the file(s) data to the database, a copy of the file itself will be kept. Data to be uploaded must be in the form of a table that has been set to allow direct insertion and updating, or in the form of a previously agreed upon format such as the format output by VUE software.  The system will automatically add entries of the data to the appropriate tables in the database.  
Entry Conditions: User is logged in and is granted permission by an administrator.  
Exit Conditions: Upload is complete.  
Flow of Events:  
    1. The user navigates to the file upload page.  
    2. The user selects the type of data to be uploaded.  
    3. The user selects the file to be uploaded.  
    4. The information within the file is transferred to the server and added to the database.  
    5. The user is notified of the operation’s completion.  
Exceptions: None.  
    1. The user cancels the upload operation.  
        1.1 The user is notified of the cancellation.  
    2. The user selects a file that does not match the format of the selected type of data.  
        2.1 The user is informed that the file is not valid for upload for the data type selected.  
        2.2 The user is prompted to upload a new file, restarting the Flow of Events.

2. The user to be deleted is also an administrator.

2.1 The Administrator is informed that the delete operation cannot be performed.

Eric C.

3.2.10 Requirement: The system must allow users to insert or modify data that are manually to be stored and later recalled by the system.

3.2.10.1 Use Case: Manual Data Entry  
Main Actor: User  
Entry Condition: User must be logged in.  
Exit Condition: The data the user entered is stored in the system.  
Flow of Events:  
    1. The user navigates to an add or modify entry page.  
    2. The user chooses to add or modify an entry into system.  
    3. The user inputs the data to modify existing data.  
    4. The user confirms the choice to add data or modify existing data.  
    5. The user is informed of the success of the operation.  
Exceptions:   
    1. The user cancels the operation.

1.1 The user is informed of the cancellation, and the entered data is discarded.

2. The user inputs data that does not match the required format of the field of entry.

2.1 The user is informed of the error and which fields have errors.

    3. The user inputs insufficient data to record the entry.

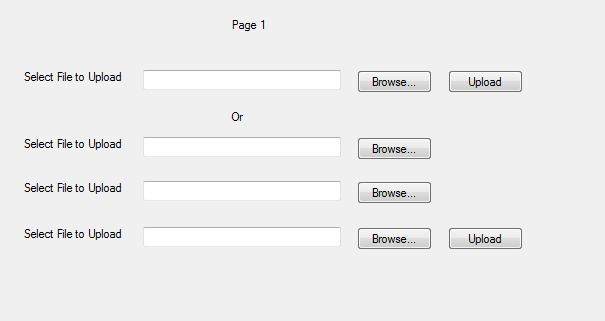
3.1 The user is informed of the error and which fields need to be filled to complete the operation.

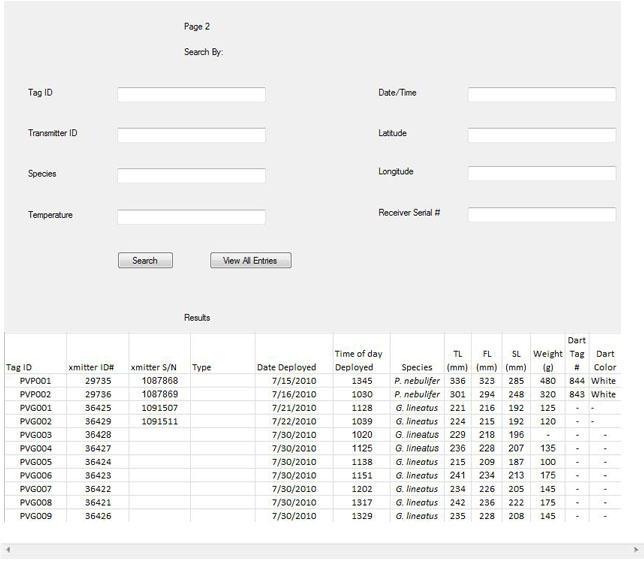
    4. The user inputs data that already exists.

4.1 The user is informed of the error and which fields need to be modified to qualify as a separate entry. Revert changes to previous state.

3.2.11 Requirement (optional): The system must allow location data (latitude and longitude) to be plotted on a map of the area on which the set of plotted data resides.

3.2.11.1 Use Case: Graphical Data Plotting  
Main Actor: User  
Entry Condition: User must be logged in and in a view that displays data.  
Exit Condition: User is shown a map with plotted points.  
Flow of Events:  
     1. User navigates to data view.  
     2. User selects data to plot.  
     3. User selects to plot the data.  
     4. Map with marked data points is shown to the user.  
Exceptions: None.





4.3     Communications Interfaces  
  
Communication functionality outside the system will be limited to email notifications regarding user registration and user permissions.  Within Sharktopus, there will be an optional comment to be attached to each update a user makes.  
  
  
  
5.3     Security Requirements  
User account information must be secure due to sensitive data such as email addresses, passwords, and the contents contained in the database. Users will be forced to provide account authentication to access their respective accounts. Confirmation emails will be sent to newly activated accounts.  
  
User profile information must be secure due to sensitive data such as email addresses and passwords and match participation. Users will use a secure log-in form in order to access his or her respective account. A user’s password must be at least 6 characters long and contain letters and numbers to render brute force attacks an unreasonable method to attempt to again unauthorized entry into the system.  Each user can only perform actions permissible for that user’s account as allowed by the account type and permissions granted by an administrator or super administrator.